Context-Aware Proactive Optimisation in Cellular Networks

Bo Ma

Department of Electronic and Electrical Engineering

University of Sheffield

Supervisors: Prof. Jie Zhang, Dr. Mauricio Álvarez

This thesis is submitted for the approval of the

Doctor of Philosophy

April 2021

This thesis is dedicated to my beloved family. Without their unconditional love, encouragement and support, I would not be the person I am today.

Acknowledgements

I am indebted to my supervisor Professor Jie Zhang. Prof. Jie offers me the opportunity for this research. His continued guidance, encouragement, and support enlighten and shape me to be an independent researcher. Prof. Jie and my second supervisor Dr. Mauricio Álvarez also point my direction in projects' planning and execution, I hope I made the best of them.

I must also acknowledge the financial support given by Ranplan Wireless Network Design Ltd, the EC H2020 Project Data Aware Wireless Network for Internet-of Everything (DAWN4IoE), and the UK Research and Innovation project Powering Urban Smart Mobility with Data Analytics (PUBLIC). They are equally important to allow me exchange knowledge and results in conference, journals, and staying abroad.

The works in this thesis are collaborated with many brilliant researchers. My sincere thanks also go to them, especially to Dr. Zitian Zhang, Prof. Bowei Yang, and Dr. Jiliang Zhang. Dr. Zitian Zhang and Dr. Jiliang Zhang selflessly devote their time and efforts to our weekly discussion for two years. Prof. Bowei Yang guides all my work for EC H2020 DAWN4IoE project while visiting Zhejiang University. Chapter 3 is a joint work with Dr. Zitian Zhang, Dr. Jiliang Zhang, Prof. Bowei Yang, and Prof. Jie Zhang. The work in Chapter 4 is collaborated with Dr. Jiliang Zhang, Dr. Zitian Zhang, Prof. Bowei Yang, Dr. Tao Hu, and Prof. Jie Zhang. Chapter 5 is a collaborated work with Prof. Bowei Yang, Dr. Yunpeng Zhu, and Prof. Jie Zhang. Besides, the State of Art in Chapter 2 partially comes from a collaborated survey work with Prof. Weisi Guo. My special thanks also go to him for his valuable suggestions in my early Ph.D stage.

My greatest thanks also go to my parents, who are always supporting me for chasing my dreams. I must thank my wife, Mrs Yue Weng. She offers me the greatest encouragement to

be brave and optimistic. The tireless support form family is always my spiritual anchor in this fast-changing and competitive environment.

I am very fortunate to work with the colleagues in the Wireless Group of University of Sheffield. I would like to specially thank Mr. Tao Hu and Mr. Songjiang Yang who are not only my friends and mentors but also the best men in my wedding.

Last but not the least, my sincere thanks go to all my colleagues, friends, and teachers who help and contribute to this thesis. Without your help, it is impossible to finish this work.

Abstract

5G and beyond networks are expected to meet the exponential traffic growth and fast-changing environments. Time-efficiency decides whether the 5G and beyond network optimisation can absorb the traffic growth and ensure a low latency simultaneously. These requirements bring challenges for operators to remain profitable while reducing operational expenditure. This thesis aims to improve the time efficiency by designing a more intelligent and user-oriented network-optimisation framework which is denoted as Context-Aware Proactive Optimisation (CAPO). This thesis quantifies the improved time-efficiency in three research lines: 1) aerial base stations (BSs) deployment and user association, 2) aerial BSs aided network off-loading, and 3) proactive load balancing. All of them share common characteristics of limited serving time and high computational complexity, so their performance becomes sensitive to the complexity. This thesis manages to keep the real-time complexity at a low level.

Firstly, Unmanned Aerial Vehicles (UAVs) are ideal carriers to substitute the terrestrial BSs and associate the ground users temporally. However, the UAV assisted BSs (UAV-BSs) deployment is a non-deterministic polynomial-time hard (NP-hard) problem that is difficult to be solved with time-efficiency. This thesis proposes a CAPO-based deployment strategy to solve this problem by reserving time-efficiency and energy-efficiency. The results indicate that problem-solving efficiency is improved at least ten times.

Secondly, fast deploying UAV-BSs will off-load the terrestrial overloaded BSs. Nevertheless, it still faces the time-efficiency problem because of jointly optimising multiple objectives, such as UAV-BSs' amount, locations, and allocating resource blocks. This thesis transforms the above joint optimisation problem into a combinatorial problem and uses a

CAPO-aided heuristic algorithm to solve it with both time-efficiency and robustness. In this result, under a time constraint, my design could finish 30% more optimisation compared with non-CAPO ones.

Lastly, the terrestrial nodes should own the ability to balance their overload to neighbouring idle cells. However, existing load balancing algorithms need more time to react to the intense traffic changes. This shortage leads to cold-start problems, which cause slower convergence speed and lower time-efficiency. This thesis employs CAPO to enable event detection from social networks and prepare the capacity to absorb an upcoming demand peak. The results indicated CAPO's ability to make the load balancing converge with eliminated overshoot.

In 5G and beyond networks, the optimisation will be proactive, service-oriented, and user-oriented. The CAPO approach presented in this thesis becomes an indispensable path to increase the quality of experience and reduce OPEX. This thesis's wider impact includes better cross-fertilising the academic fields of data analytics, mobile edge computing, artificial intelligence, and wireless communications, as well as informing the industry of the promising potentials in this area.

List of Publications

Published Journals

1. **B. Ma**, W. Guo and J. Zhang, "A Survey of Online Data-Driven Proactive 5G Network Optimisation Using Machine Learning," in IEEE Access, vol. 8, pp. 35606-35637, 2020, doi: 10.1109/ACCESS.2020.2975004.

2. **B. Ma**, B. Yang, Y. Zhu and J. Zhang, "Context-Aware Proactive 5G Load Balancing and Optimization for Urban Areas," in IEEE Access, vol. 8, pp. 8405-8417, 2020, doi: 10.1109/ACCESS.2020.2964562.

3. Hu, T.; Wang, Y.; **Ma, B.**; Zhang, J. Orbit Angular Momentum MIMO with Mode Selection for UAV-Assisted A2G Networks. Sensors 2020, 20, 2289.

Published Conference

1. **B. Ma**, B. Yang, Z. Zhang and J. Zhang, "Modelling Mobile Traffic Patterns Using A Generative Adversarial Neural Networks," NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, 2020

Submitted

1. **B. Ma**, J. Zhang, B. Yang, Z. Zhang, and J. Zhang, "Energy Efficient UAV-BS Deployment and User Association based on Machine Learning", submitted to IEEE Journal

on Selected Areas in Communications, UAV Communications in 5G and Beyond Networks

2. **B. Ma**, Z. Zhang, B. Yang, T. Hu, F. Li, Z. Zhao and J. Zhang, "Time-Efficient UAV-based Off-Loading by Few-Shot Learning Aided Simulated Annealing", submitted to IEEE Transactions on Network Science and Engineering

Table of contents

List of figures

List of tables

Nomenclature

Roman Symbols

a aerial-ground channel constant parameter

$a_{\mathrm{DL}}, b_{\mathrm{DL}}$ parameters for estimating cellular traffic based on number of Tweets [1]

B bandwidth of channels

b aerial-ground channel constant parameter

C minimum data rate requirement

c speed of light

C_i achievable data rate (in bits per second) of ground UE i

D_{diff} difference matrix

$D(\cdot)$ GAN's discriminator

d_{ij} the horizontal distance from UAV-BS j to the UE i

D_t the user distribution of a certain ground user set

$\mathbb{E}(\cdot)$ expected value

f frequency of of carrier signal

$G(\cdot)$ GAN's generator

g_{ij} BS j's channel pathloss to serve UE i

G_t G_r values of transmit and receive antenna gain

h height of UAV-BS

I number of UEs

i index of UE

$I_{\mathrm{UE},t}$ set of ground UEs

J number of UAV-BSs serving the region R

j index of UAV-BS

J_{UAV} set of UAV-BSs

k number of clusters in K-Means or number of neighbours in KNN

$\mathscr{K}$ urban region for load balancing

m number of move-in and move-out UEs from the region

n number of UEs moving in the region

N_i number of user coordinates

N_j amount of UAV-BSs for off-loading

n^o_{max} a maximum threshold for normal condition

$\mathscr{N}(\cdot)$ normal distribution

n_{ot} number of Tweets per period

N_{Test} number of testing ground user sets

$\mathscr{N}$ total number of Tweets

$n_y \times n_x$ grids divided in R

$O(\cdot)$ time complexity

o RoI cluster

$P(\cdot)$ probabilistic models

p_{ij} BS j's transmission power to serve UE i

$\mathrm{PL}(\mathbf{dB})$ the pathloss model in [2] in decibel

P^{LoS} probabilities of the transmission link in LoS state

$p_{\max}$ maximum transmission power from UAV-BS j to UE i

$p_{\min}$ minimum transmission power from UAV-BS j to UE i

P^{NLoS} probabilities of the transmission link in NLoS state

$P_R(\mathrm{dB})$ the received power in decibel

$P_T(\mathrm{dB})$ fixed transmission power for offloading

R a certain region

$\hat{r}_{DL}$ estimated Down-Link (DL) traffic load in cluster o in time interval t

$\mathbb{R}$ real number

$\mathbb{R}^{2\times 1}$ 2×1 real matrix

R_i average resource blocks prepared for each UE

R_I total allocated resource blocks for off-loading

r_{ij} distance between UE i and UAV-BS j

R^{θ} minimum requirement of resource blocks

$S_1, S_2, ..., S_m$ sets of UAV-BSs

S_{Jt} probabilistic model of UE distributions

t time interval

T, T^{cold} settings of temperatures in Simulated Annealing

TF target function

u_a a UE

$U(\cdot)$ uniform distribution

u_j load balancing BS association variable

$V(D,G)$ GAN's joint objective

$v_{\text{xmax}}, v_{\text{ymax}}$ experimental region limitation parameters

W size of off-line database

w_{ot} term frequency in Tweets

w' a large weight value in bipartite

x training data for GAN

(x_i, y_i) coordinate of each user

X_i, Y_i UE's position in grid

x_j, y_j grid location of UAV-BSs

$(x_I, y_I)^T$ a list of user coordinates

$\bar{x}_j, \bar{y}_j$ fixed positions of UAV-BSs

$(x_J, y_J)^T$ list of UAV-BSs' locations

z noise data

$\mathbb{Z}^+$ positive integer

Greek Symbols

$\alpha_1, \alpha_2, \alpha_3$ weights of diverse off-loading targets

δ_d grid size

δ_{ij} Boolean variable of BS-UE association relationship

Γ_{diff} difference degree

γ_{ij} SNR of UE i receiving the signal from UAV-BS j

γ_θ minimum required SNR

$\mu^{\mathbf{LoS}}$ the means of excessive loss caused by man-made structures for LoS

$\mu^{\mathbf{NLoS}}$ the means of excessive loss caused by man-made structures for NLoS

μ, σ log-normal distribution parameters

ρ_j load balancing channel association variable

σ_n^2 constant noise power

Φ OFDMA sub-channels

θ_{ij} elevation angle of the transmission link between UE i and UAV-BS j

ε radius setting in DBSCAN

Other Symbols

$\xrightarrow{u_x}$ the handover of u_x

Acronyms / Abbreviations

3*D* Three-dimensional

3*GPP* 3rd Generation Partnership Project

5*G* Fifth-generation cellular network

API Application Programming Interfaces

ARMA Auto-Regressive Moving Average

BBU Baseband Unit

BS Base Station

CA Context Aware

CAPO Context-Aware Proactive Optimisation

CSV Comma-Separated Values

D2D Device-to-Device

DBSCAN Density-Based Spatial Clustering of Applications with Noise

eNB Evolved Node B

GAN Generative Adversarial Networks

GP Gaussian Process

GPS Global Positioning System

IoT Internet of Things

KDE Kernel Density Estimation

KM Kuhn-Munkres algorithm

KNN K Nearest Neighbour

KPI Key Performance Indicators

LoS Line-of-Sight

LSTM Long Short Term Memory

MDT Minimisation of Drive Testing

MINLP Mixed Integer Non-Linear Programming

NARX Non-linear Auto-Regressive with exogenous model

$NP-hard$ Non-deterministic Polynomial-time hard

OFDMA orthogonal frequency division multiple access

OPEX Operational Expenditure

PSO Particle Swarm Optimisation

QoS Quality of Service

RNN Recurrent Neural Networks

RoI Region of Interest

SA Simulated Annealing

SNR Signal-to-Noise Ratio

SON Self-Organising Network

SVM support vector machine

UAV Unmanned Aerial Vehicles

UDUA UAV-BS deployment and user association

UE User Equipment

WiFi Wireless Fidelity

Chapter 1

Introduction

Overview

This chapter provides the background and motivation of this PhD thesis, followed by the research objectives and the thesis structure.

1.1 Background

The mobile network is the foundation of future global digital systems. Recent developments in fifth-generation (5G) and beyond 5G need to support three highly heterogeneous services, enhanced mobile broadband, ultra-reliable and low latency communications, and massive machine-type communications. In detail, these services need to support a 600x to 2500x capacity increase [3], sub 1ms round-trip latency[3], and 10,000 or more low-rate devices per cell site [4]. The dramatically increasing amount of devices, services, frequency bands, and capacity will bring a rising number of decision variables to be optimised so the operators need to face more challenges, such as higher complexity and worse time-efficiency of optimisation. However, the mobile network is expected to have low latency (better time-efficiency) which conflicts the reality. The direct solution is adding more computation ability, like faster chips, but this translates to a sharp rise in the operational expenditure (OPEX) ($\approx 60\times$ increase [5]), which is not desirable. Accordingly, designing a network optimisation framework aiming for reducing the computation time as well as the complexity becomes critical to maintain

operators' profit. Context-Aware Proactive Optimisation (CAPO) is designed for this goal. This section will provide this background in detail. Let's start from analysing the historical milestones of mobile network optimisation.

1.1.1 From Reactive (2G-4G) to Proactive Optimisation (5G-Beyond)

Radio resource management (RRM) and network deployment are the primary focus areas of the thesis. There are many underlying optimisation functions, including scheduling, mobile edge caching, backhaul optimisation, interference management, load balancing, and many aspects of coverage and capacity optimisation. From a historical perspective, RRM and network optimisation have moved from engineering expertise based (e.g., human-expertise driven manual configuration in the late 1990s) to reactive numerical optimisation (e.g., expertise-driven numerical functions with parameter inference in post-2010). With **increased complexity** and the need for real-time analytics that is personal to consumers, it now needs to evolve into big-data-driven proactive self-optimisation. The author will first briefly review the historical development of optimisation before diving into the enabling technologies for proactive optimisation.

Reactive Network Optimisation: In the early days of the 2G network, radio engineers monitored the network statistics and tuned the network to improve key performance indicators (KPI). Engineers used their field knowledge and previous experience (e.g., drive testing) to diagnose the origin of problems [6]. However, it took a long time for engineers to detect and diagnose the problem manually, and the network might need several hours from the occurrence of a problem to network recovery. For 3G optimisation, researchers and operators tried to reduce the human-machine interaction. For example, the 3GPP proposed Minimisation of Drive Testing (MDT) in [7] and designed Markov Decision Tree-based optimisation to maximise traffic offload in Wide-band Code Division Multiple Access [8]. Nevertheless, each optimisation algorithm still required frequent configuration by engineers and was not personal to individual consumers. It was more towards a service area (e.g., city council or a shopping mall) or a service genre (e.g., maximum rate or proportional

fair). Besides, in the automatic examples, the schedulers still required more than one hour to coverage (76 minutes in [9]).

Developments in Self / Proactive Optimisation : In the 4G period, the 3GPP stated the significance of implementing automatic optimisation and introduced Self-Organising Network (SON) in Release 8 [10]. In the past decade, a significant number of SON implementations have been developed to enable cell sites to self-optimise their coverage and capacity [11], energy savings [12], and load balance [13, 14]. The commonly used optimisation methods included reinforcement learning [12], Fuzzy controllers [11], regression tree [14]. One challenge with machine learning approaches is that the integration of data is typically low dimensional (e.g., channel estimation or QoS reporting). The contextual information is missing to personalise services as well as the forecasting capability to enable proactive optimisation. As such, typically advanced SON engines reacted over 10 minutes after a severe event [14, 15].

In the 5G, to meet the **fast-changing** demand in dense network deployment, the SON decision process has to converge to a satisfactory solution in a short time. The optimisation time is influenced by algorithm time complexity, computing ability, the time to trigger the algorithm, as well as the uncertainty of the decision's benefit (regret function). This time is vital because many optimisation algorithms are **time-sensitive** that the time-scale highly influences the Quality of Service (QoS) delivery. For example, in a Power Load Sharing research [15], the user dissatisfaction rate would increase by nearly 20% if the time to trigger the algorithm was delayed from one minute to one hour. That is because the network recovers after the degradation of performance. This outage period increases user dissatisfaction experience [15] and risks increased customer complaints and lower customer loyalty to their network [16]. As show in the 'Time Scale' column of Table 1.1, the optimisation time is developed to be closer to real-time and even proactive to save cost and improve loyalty. Approaching this purpose requires heterogeneous data sources to sense the fast-changing network context. On the one hand, current developments in cloud-fog-edge computing have laid the foundation to enable large-scale big-data analysis on cloud and small-scale streaming-data analysis on edge [17]. On the other hand, proactive optimisation in 5G and beyond 5G

offers low-latency and reliable communication services to transfer data. The widely-used machine learning algorithms aim to configure model-free optimisation for reducing real-time complexity [5, 18, 19], and a current trend for improving SON decision time horizon is triggering the algorithm in advance (e.g., proactive). Table 1.1 summarised the network optimisation the general view of developments from 2G to 5G.

Table 1.1 Developments of cellular network optimisation from passive to proactive, data-driven, and self-optimisation

Gen.	Background	Data	Source	Self / Proactive Optimisation Examples	Automation Methods	Time Scale	Summary
2G	Specialists monitor the network statistics to control KPIs and plan the route for drive test [6]	Network statistics and KPIs	BS and manual measurement	Alarm monitoring (intelligently set warning threshold)[6]	-Engineers use their filed knowledge to diagnose the origin of problem	Limited by the ability and knowledge of engineers from several minutes to hours	Passive optimisation by engineering expertise
3G	Manual trouble-shooting and optimisation through analysing the statistics and drive surveys[8]	Antenna tilt[8] Transmitter power UE measurement [9] Interference [9]	UE [8, 9] Macro BS[8] Micro BS[8]	-3GPP Minimisation of Drive Testing (MDT) [7] -Use MDT based SON to maximise offload [8] Automatic control of coverage [9]	-Decision tree[8] -Continuous optimisation iterations[9]	-Converge after 65 iterations and take 76 min to execute [9] -Take 23-162 optimisation steps [8]	Reactive numerical optimisation
4G	Manual network parameters managements associated with high cost and long delay [13]	Load [11, 13, 20, 21] Down tilt [11, 13] Handover parameters [14] Transmitter power [11, 21] QoS [23, 24]	eNB [11, 14] [13, 20, 21] UE measurement [14]	-3GPP SON Release 8 [10] -3GPP Mobile Data Applications Impacts (Release 11) [22] -3GPP Context Aware Service Delivery (Release 14) [25] -Coverage and capacity [11][23][24] -Load balancing [13, 20, 14, 26] -Interference management [21]	-Fuzzy controllers [11, 14, 20, 15] -Decision tree [13] -Polynomial regression [21][23] -Reinforcement learning [24] -Q learning [26]	-Converge in around 10 minutes [14] -Take around 500 episodes [11] -Take around 20 iterations [13] -Converge after 8 mins [15]	Self-optimisation
5G	Dense network deployment results in an immense amount of nodes [36] Manual and reactive optimisation face heavy burden [35]	Radio measurements [5, 18] Transmitter power [27] Mobility event [27] Load [28]	Femto cell [27] Small cells [28]	-3GPP service-based architecture and network slicing [29–31] -Femtocell coverage SON [27][34] -User Centric CoMP clustering[28] -Interference management [28] -Mobile edge caching and computing [32][33] -Sleeping cell detection [5]	Continuous optimisation iterations[27][32] Data analysis [35] Fuzzy controller [34] Polynomial regression [18] K-nearest neighbour [5] Decision tree [28]	Take 5-12 iterations [28] Proactive[32][33]	Big-data driven self-optimisation towards proactive

1.1.2 Enablers for Proactive Optimisation

The 3GPP Mobile Data Applications Impacts (Release 11) [22] mentioned that network optimisation will be boosted with time-efficiency if it can understand and forecast user behaviours and spatial-temporal traffic pattern through multi-source (heterogeneous) data. In that case, the proactive optimisation needs the status information of concrete entities in the social space (e.g., users) or virtual entities in the cyber space (e.g., software), such user-centric meta-information is called **context** [37, 38]. The context represents all the user information indicating spatial-temporal network traffic characteristics for a user-centric network, including geolocation and user behaviour. The methods to gain the context are named *context-aware* or *context-awareness*. This context-aware module automatically collects and

analyses data from different sources (e.g., online data and personal devices), then supplies context for adequately re-allocating communication resources [15].

Heterogeneous Data: The heterogeneous data includes sources from not only physical network side (e.g., network KPIs) but also the social side. **Online data** from the **Internet** has been a typical and major source. It can be divided into different types: social networks, video/photo sharing sites, online forums, product reviews/ratings, and wikis. These types inherently contain substantial hidden information about users and hold different data merits and pitfalls. The online data directly and tightly connects to users' intents, and therefore appropriate for transforming network optimisation to be proactive and user-centric [39]. Furthermore, the **social networks** have become the most popular ones which change users from content viewers to content creators and distributors. It not only owns plenty of shared information about public and individuals [15] but also supplies real-time details for forecasting spatial and temporal attributes of future events. The social network data consists of four data formats: geolocation, timeline, text, and photos/videos. The social network also records the social relationship/tie which benefits the estimation of the weights of Device-to-Device (D2D) links. For example, the forecasting of social ties in [40] [41] enable D2D in caching delivery with finding the most influential user [40] and sharing with friends [41]. In this thesis, the used heterogeneous data contains both social network data and the cellular network KPIs.

Context-Awareness: To alleviate the threat of optimisation complexity, context-aware (CA) network optimisation emerges to substitute traditional expertise-driven and numerical optimisation to transform the trade-off between performance and revenue in a fundamental way. The idea of context-awareness originated from computation systems that could react to users' changing context [42] in 1994. Then, it steps in mobile computation in 2000 [43] and wireless sensors network in 2008 [44]. This context-awareness allows efficient network optimisation to satisfy actual users' demands with the minimum cost of resources. In detail, it collects and analyses any information about network statuses, such as network traffic, user locations and behaviours. Then, the context is fed to optimisation-decision makers which adjust network behaviours to fit current demand. One 3GPP example is the video context-

aware scheduling optimisation [25]. It is based on collecting and analysing the user-side attention information to guide video caching. Users are supplied with contents meeting their current attention, so the video streaming transmission flexibly varies when the context changes. Otherwise, non-CA methods will waste resources in caching many not-hit videos or occupying extra backhaul for streaming transmission. Generally, the context-awareness includes but not limited to **network-aware, traffic-aware, user-aware, content-aware, data-aware, data-driven**, etc. Nevertheless, the value of context is not only limited to immediate decision-making, it can also be used to predict future network status. Real-time CA optimisation needs time for data collection, analytics, and computation, which results in a delay to deploy the computed strategies. Unfortunately, as the network is becoming more and more complex (heterogeneous), the increasing delay makes the optimisation's effects lag behind the arising of negative situations, which is not desired. **Forecasting** future context can solve this time-efficiency problem by using predictions to configure the network in advance, so the real-time heavy computation is not needed. This optimisation scheme is denoted as Context-Aware Proactive Optimisation (CAPO).

1.1.3 CAPO: Context-Aware Proactive Optimisation

CAPO aims to closely couple network-resource allocation dynamics with predicted context dynamics, allowing it to respond before a negative situation arises (e.g. signal outage or network congestion) and leads to the negative consumer experience. This definition is firstly proposed in [45] in 2016 for caching content that will be requested in future and further expanded to wide categories of network optimisation in [46] (e.g., load balancing and resource management). For instance, the 3GPP Release 15 (System Architecture for the 5G System) includes building users' **mobility** pattern for Access and Mobility Management Function to benefit the **priority** in radio resource management [29]. The changes in users' mobility lead to the spacial movements of communication demands, so the prediction of mobility enables adequate resources to be deployed to the destination in advance and wait for the demands' arriving. 50% time-efficiency and 1.3% user satisfaction rate are improved when predictive user mobility is provided for load balancing [15]. In summary, CAPO can

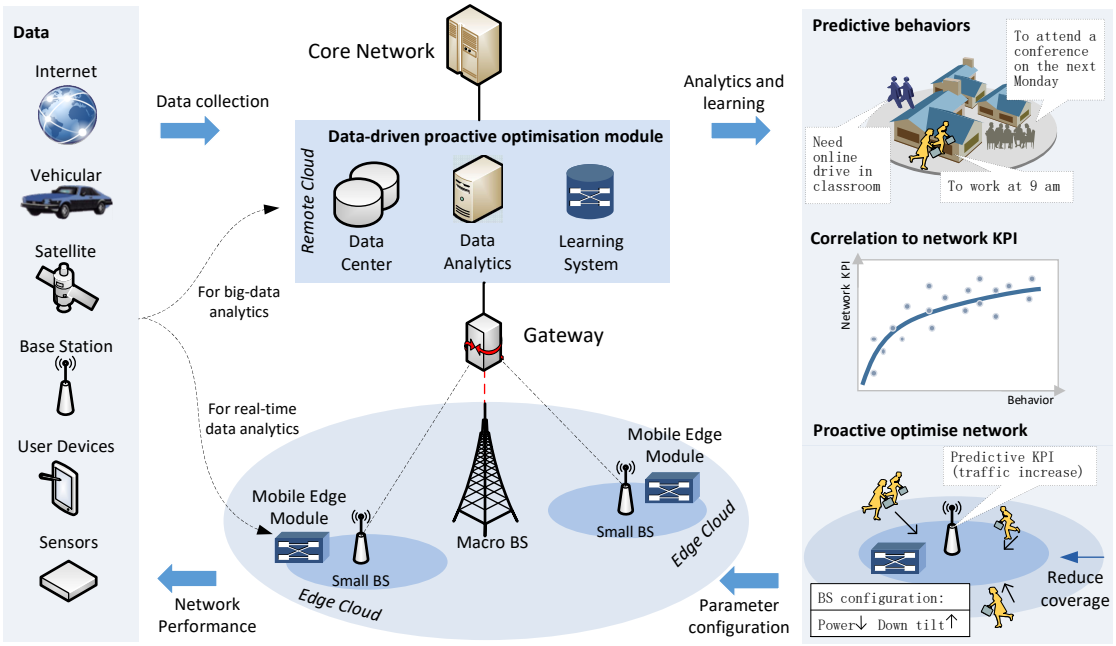


Fig. 1.1 The framework of CAPO.

decrease computation time as the predictions highlight the beneficial direction of optimising configurations. Moreover, it can increase profit-cost ratio because the resources are placed in advance to where they are needed.

1.1.4 The Framework of Context-Aware Proactive Optimisation

To make the enablers drive proactive optimisation, this thesis summarises a general CAPO framework in Fig. 1.1 that involves data acquisition, integration, and using forecasting to drive RRM and network optimisation. Complex computation relies on cloud and edge computing. Example frameworks of combining cloud-edge computing with big data analysis can be found in [47, 17, 48]. In these works, a **cloud plane** is responsible for global big-data with large-scale and long-term (cloud computing), while local data is processed by an **edge plane,** (edge computing) because of its small scale and short term. This thesis follows a similar idea, the remote cloud and the edge cloud work as the cloud and edge planes, respectively. In Fig. 1.1, the remote cloud (data-driven proactive optimisation module) is placed between the Gateway and the core network. An example of this implementation is

shown in [49]. It placed the caching module close to the small BS and between the core network and edge network. That will benefit the data storage, analysis and learning, but the precondition is that the latency should be acceptably low. The following parts illustrate the functions of the framework.

Data: This step includes data collection, cleaning, and storage. For a more detailed process, the work [17] further divides it into the organisation, representation, cleaning, reduction and integration. These aspects have been successfully implemented thanks to the support of edge computing. The data **collection** is a process of gathering information on variables of interests through multiple online and offline available sources, such as the Internet, vehicular, satellite, base station, user devices, government and business databases, and sensors. The combined data may vary in sparsity and resolution across urban and industrial areas [50]. For the Internet data, the providers provide Application Programming Interfaces (API) for the third parties to access the open data. Then, the raw data can be noisy and redundant, so it needs a **cleaning** process (e.g., sort, filtering) to be established in a systematic fashion and stored in the edge data centre or cloud data centre for further analytics. The tensor-based method is useful to analyse big data by focusing on typical features [51]. The extensible order tensors can represent unstructured and structured data. Typical applications of this method are in [52, 53]. These works illustrate how tensors work in cloud-edge computing. Big streaming data is also a challenge for real-time big data processing. High-order singular value decomposition is proved efficient to avoid redundancy (see examples in [54]).

Predict Consumer Behaviours: The network traffic fluctuates according to consumer behaviours. This step is building models to predict demand changes. The input can be the online data from the Internet, and the output is the probability of different demand levels across various behaviour contexts and slices. Understanding the posterior distribution of predictions will generate a spatial-temporal consumer demand distribution that helps predict the network KPI.

Correlate to Network KPI: The correlation models a path of mapping predictive behaviours to network KPIs, such as the network traffic. The polynomial regression (in the

figure) and statistics analysis (e.g., Pearson correlation) are two commonly used approaches. Therefore, the model input is the behaviour probability, and the output is the probability of KPI, such as the probability of high-load occurrence.

Proactive Network Optimisation: The predictive network KPIs are injected into this function. The optimisation algorithm needs to configure the parameters in advance for the upcoming condition changes to achieve targets. An example is proposed in [55] about optimising the network in a proactive and energy-efficient way. They presented a framework with implementing a big-data-aware intelligent platform between the core network and Baseband Unit (BBU) pool for analysing user behaviour and network patterns to output control strategies. Note that, these analytics and leanings are available to be carried out by both remote cloud plane and edge plane. The cloud computing can generate a general trend context of public behaviours, such as traffic of a city in rush hours, which suggests a macroscopic optimisation. At the same time, the edge computing processes personalised context with the help of edge data centre, edge tensors, edge data management and analysis. Finally, the network configuration in the physical space is decided according to both trend and individual context.

1.2 Motivations

The above potential of CAPO motivates the author's study, especially for the time-sensitive services, such as aerial BS aided cellular network. In this thesis, the author presents state-of-the-art methods using CAPO for addressing **three major issues**: aerial-BS deployment, aerial BS aided offloading, and proactive load balancing.

Aerial BSs Deployment: In the first place, recent developments indicate that the BSs are not limited to terrestrial, aerial BSs have attracted researching attention. The aerial BSs are carried by Unmanned Aerial Vehicles (UAVs) with **flexible** three-dimensional (3D) mobility to increase the probability of Line-of-Sight (LoS) connections. Moreover, they are able to **fast respond** to the needs of network recovery. However, such a transformation can fundamentally change the mechanism of the network optimisation. Not only the user

association but also the locations should be optimised. That becomes a **complex** NP-hard problem. And the serving **time is limited** because of battery constraints, so the optimisation should be not only energy-efficient but also time-efficient. Fortunately, these requirements can be satisfied by CAPO, but this CAPO-based UAV-BSs' deployment has not been well investigated.

Aerial BSs Aided Off-Loading: Secondly, the aerial BSs are also ideal performers for temporal traffic-offloading. They work by aiding the terrestrial BSs to cover hotspots such as social event fields, sports stadiums, and industrial areas with massive Internet of Things (IoT) devices. Compared with previous deployment optimisation, the UAV-BSs for offloading only require to serve part of the users previously served by the terrestrial BS. In other words, it is first to find the locations of hotspots and then help terrestrial BSs to handover the users in hotspots. Here, the key factor is to **fast** locate the hotspots. However, it is complex to answer the following three questions without human interaction: (1) how many UAV-BSs does the optimisation need to be dispatched? (2) how many resource blocks do the UAV-BSs occupy? (3) where are multiple UAV-BSs going to be deployed? To best benefit the profit/cost ratio, operators would like using fewer UAV-BSs with occupying less resource but offloading more users. It becomes a **complex** multi-objective optimisation problem that would be **time-consuming** to be solved. Thanks to the improvements in data analysis and machine learning, CAPO can build data-aware models based on historical data and reduce the repetitive computation in future. The challenges are designing the new CAPO framework and addressing low robustness while meeting a lack of data.

Proactive Load Balancing: Last but not the least, terrestrial BSs are also able to balance the load from full-loaded cells to their neighbouring idle cells, so if the neighbouring cells can digest the extra load, UAV-BSs are not necessary to be dispatched. In this condition, the involved terrestrial BSs will dynamically adjust the user association to balance their loads. This technique is denoted as load balancing. Nevertheless, the optimisation problem becomes **complicate** when there happens a popular event attracting many users. In this case, a large number of users gradually flood into the involved cell, which becomes fully-loaded soon. Then, the involved BS will continue emptying rooms again and again for the newcomers,

which is reactive to the traffic fluctuation. Such a reactive optimisation has a **time-lag** to converge to the optimised condition, which is not time-efficient. Here, if the social event is assumed to be known in advance, the location and time of the events can be alerted to the involved BS. In that case, this BS can proactively empty enough capacity to face the upcoming peak of demands. Then, the time-lag will be reduced. This becomes a new problem about how to couple social event detection with the load balancing. Novel CAPO-based framework and social event-detection techniques are needed.

1.3 Aim and Objectives

1.3.1 Aim

This thesis aims to push the above three barriers further by exploiting the applications of CAPO based on heterogeneous data analysis and machine learning. To accomplish it, this thesis follows three main research lines.

1.3.2 Objectives

The first research line is to investigate CAPO-based UAV-BS deployment algorithm. Specifically, the algorithm needs to provide reliable long-term coverage for ground users by multiple UAV-BSs with minimum energy consumption. Each user requires a basic downlink QoS. This is a joint UAV-BS deployment and user association problem with minimising the UAV-BSs' total transmission power. Moreover, such a joint problem has to be solved with time-efficiency. Detailed objectives are listed as follow:

1. To transform the joint optimisation problem into a Mixed Integer Non-Linear Programming (MINLP) problem and use a divide-conquer algorithm to divide it into two sub-problems, UAV deployment and user association, which can be conquered with optimum solutions, respectively.

2. To solve the two sub-problems by four methods (an exhaustive method, a random method, a heuristic method, and CAPO-based methods).

3. To compare the CAPO-based algorithm with other reviewed algorithms by the total transmission power of UAV-BSs and computation time.

The second research line is to design the CAPO-based network offloading scheme assisted by UAV-BSs. Diverse targets are required to be considered at the same time, they are 1) minimising dispatched UAV-BSs, 2) minimising occupied resource blocks, 3) maximising the offloaded users. The designed CAPO framework also needs to reduce the computation iterations to solve the multi-objective optimisation problem. At the same time, the data scarcity should be considered because it can have a negative impact on CAPO's performance. The detailed objectives are listed as follows:

1. To convert the multi-objective optimisation problem into a combinatorial problem.
2. To solve the combinatorial problem by Simulated Annealing.
3. To improve the time-efficiency of the traditional Simulated Annealing by the designed CAPO framework.
4. To address the performance degradation caused by data scarcity.

The third research line is to balance the load between cells under the circumstance of events. The event should be detected from social network data, and this context needs to be associated with load balancing. The expected condition is that the events are automatically alerted before their occurrence, and the involved cells are configured in advance to adapt the changes fast. To design a CAPO-based framework, three problems should be addressed: 1) How to detect traffic changes during events? 2) How to use the events' context in the load balancing? 3) How to find the best time to trigger the CAPO-based load balancing? Based on the above questions, this thesis lists the following objectives:

1. To design a data-analytic based module to alert the events' hotspots.
2. To design the CAPO-based framework consisting of event detection, fault correction, and load balancing.

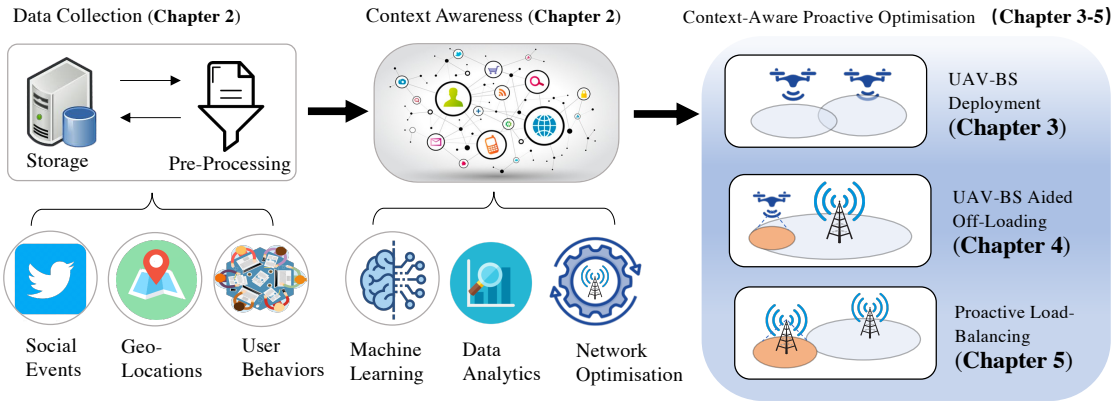


Fig. 1.2 Thesis structure and organisation of chapters.

3. To determine the best time to trigger the load-balancing settings. The best time should provide proper performance while reserving as much time for a prediction.

1.4 Structure of the Thesis

By following the above research lines, the author provides the organisations of chapters, as shown in Figure 1.2. Chapter II provides the state of the art of each research line. The author also gives learnt lessons to guide current and future researches. Techniques to gain context-awareness are summarised, including up-to-date machine learning, modelling geolocations, predictive user behaviours, and cellular traffic prediction. The author also reviews the solutions of widely-mentioned problems of CAPO: prediction errors and privacy problems. Chapter III follows the first research line and solves the problem of CAPO-based aerial BSs deployment. Chapter IV follows the second research line for improving the computation efficiency of Simulated Annealing based offloading schemes. Chapter V follows the third research line, and proactively balances the loads among cells during events. Chapter VII concludes this thesis and provides future research directions.

Chapter 2

State of the Art and Research Challenges

Overview

This chapter provides the state of the art. The author firstly reviews previous and current developments of each research line, then gives the up-to-date research challenges that require to be solved by the proposed CAPO framework in this thesis. Next, categorised context-aware methods are reviewed to support CAPO, including geo-locations and traffic predictions. Finally, all the mentioned machine learning tools, data analytics, and open challenges of CAPO (e.g., privacy and errors) are summarised to give a general view.

2.1 Review of UAV-BSs' Deployment

In 1996, the US Army Communications Electronics Command [56] published the plan to use UAVs as carriers of relays for battlefield broadcast system. It could be the first time to use UAVs to support wireless communications. This idea was transferred to cellular networks in 2001 by Google [57]. The UAVs were transformed to UAV-BSs to provide telecommunications services for terrestrial UEs. Because of their flying nature, the UAVs can provide LoS connections toward ground devices leading to an improved coverage and throughput performance. These benefits enable UAV-BSs as qualified candidates to support BS malfunction (e.g., during natural disasters). This new technique helped existing terrestrial infrastructure satisfy the transmission requirements of wireless users that attracted significant

researching attention. At this early stage, the deployment of the UAV-BSs highly relies on **human** interactions. For example, it is the engineer who decide the location for the UAV-BSs to visit or hover. Due to the demand of higher efficiency and better performance, the autonomous deployment became the new research challenge.

Automatically deploying the UAV-BSs requires an algorithm to determine the flying destinations and an ability to collect essential information. UAV-BSs are deployed targeting the quick-response service recovery after infrastructure damage or unexpected disasters. In 2008, Song proposed one of the first works for the **autonomous** UAV-BS deployment [58]. It was a geographical clustering method (e.g., K-means) to find the target locations of UAV-BSs. Also, the UAV-BSs were assumed to know users' distribution as well as their demands. However, along with the evolutionary demands for greener networks and a seamless coverage, the higher quality of optimising the UAV-BS networks is required in an urge way, which imposes enormous challenges on maximising the coverage, maximising the sum (data) rate of users, or minimising the power consumption.

Coverage optimisation influences the quality of service because some far-side users can also enjoy the service when the coverage is large enough. This gains highest importance when there emerges malfunction in the terrestrial BSs especially during disasters, like earthquakes. The coverage of UAV-BSs has been simplified as a function with the flying height and the channel path loss. For instance, A. Hourani in [59] derived the optimal flying altitude for a single UAV to yield the maximum **coverage range**. The path loss was given as fixed values. This work has opened the horizon of the coverage optimization and been followed by many works. Farooq [60] considered using multiple UAV-BSs to provide wider coverage. There UAVs cooperated in clusters and associated users in a greedy way that the users with better channel quality will be connected in priority. Sometimes, the goal of maximising the coverage can be altered to maximise the number of covered users. That increases the resource utilisation but sacrifices further users who has worse channels. An example was proposed in [61]. Thanks to these works, the coverage optimisation becomes implementable, but the available data rates to the users were not considered.

The **data rate** is derived by the Shannon–Hartley theorem in which the bits per second is related to the bandwidth and the signal to noise ratio (SNR). The bandwidth optimisation can be regarded as the optimisation of users' association to the UAV-BSs. For example, the UAV-BS serves the users with better channel quality first until reaching the maximum capacity. This is a common and simple user-association method which can be categorised as the greedy method. The user association is mutually influenced by UAV-BSs' 3D locations, the transmission power, and the channel quality (e.g., noise, interference, and LoS conditions). In that case, recent works commonly optimised the deployment of UAV-BSs to increase the sum of data rates. Alzenad et al. [62] presented a circle packing theory-based three-dimensional deployment algorithm to maximize the sum downlink throughput. Another similar work in [63] proposed a Dinkelbach-based method to maximize the **sum rate** of considered users. This paper positively took the time efficiency into consideration. Nevertheless, these solutions became limited when the number of UAV-BSs increases because multiple UAV-BSs can mutually influence each other. Modelling and solving the complicate effects among mobile UAV-BSs require a new method. Reinforcement learning is regarded as such a solution because the multi-layer neural networks own the potential modelling the complex inherent systematical influence. Liu et al. [64] and Hammouti et al. [65] employed the reinforcement learning as the core of the distributed UAV-BS development and user association framework. The results indicated that 4-6 UAV-BSs were guided to the proper destinations. However, limitations still exist. For instance, none of these studies considered the energy consumption while flying (propulsion) and transmitting signals. Such a shortage might cause battery waste and lifetime running out.

Energy management is a part of the green network for lower carbon emission. The energy consumption of UAV-BSs consists of the propulsion energy (power) and the (signal) transmission energy (power). In detail, the propulsion energy supports UAV-BSs' hovering (flying with fixed 3D location) and movement (flying with changed 3D location), which is mainly influenced by the UAV flying speed and acceleration. Other factors also affect the propulsion-related power, like air density, drag coefficient, wing area, but in literature these factors can be modelled as constant scalars that are not the optimising objects. While UAV-

BSs working in the hovering mode (zero speed), the power is influenced by the aircraft weight, air density, and rotor disc area, etc., so the propulsion power for hovering is approximately a constant value [66]. In contrast, the transmission power is related to the quality of circuitry, wireless systems, and signal processing. In different applications, the optimising target function changes. For example, during the malfunction of terrestrial BSs, UAV-BSs are expected to hover and substitute the terrestrial BSs. While collecting data from sensors, the UAVs need to move and visit several fixed location. Accordingly, the energy optimisation is categorised as trajectory optimisation and hovering optimization. The trajectory optimisation needs to consider both the propulsion power and the transmission power. And the hovering optimization considers only the transmission power but requires detailed user association.

The trajectory optimisation adjusts the moving path of UAVs. The UAV-BSs continuously observed their current statuses defined as channel conditions and gradually explored, and learned how to move to the optimal positions when they had been dispatched to the area of interest. This optimisation requires continuously updating its strategy in a time-varying environment, so the reinforcement learning is selected. Liu et al. [67] deployed UAV-BSs based on this method aiming for energy minimisation. The authors considered 4 UAV-BSs managed by a central controller but neglected their interference to each other. Hasini et al. [68] did a similar work and included interference in their system model. However, these works rely on frequently data exchange between UAV-BSs and the central controller. Such optimisation-after-dispatching introduced not only huge information exchange overhead but also extra propulsion energy consumption if the model is ill-trained. Off-line training is also needed and this thesis names it optimisation-before-dispatching.

The strategy of optimisation-before-dispatching (short as **before-dispatching**) determines the optimal deployment before UAV-BSs are dispatched, so it assumes quasi-static user distributions. If the environmental context is predicted correctly, such a strategy enables the proactive optimisation. The related works commonly divided the complex problem into UAV-BSs' 2D deployment, user association, and the hovering height optimisation, then solved them by the heuristic methods and the iterative optimisation. At the early stage, researches focus on optimising one UAV-BS. And the optimising objects gradually changed

Ref.	Target	Trajec-tory	Propulsion Power	Number of UAVs	Inter-ference	User Association	FDMA	Tran. Power	NLoS	Before-dispatching	Time Efficiency
[58]	Coverage	×	×	100+	×	×	×	×	×	✓	×
[59]	Coverage	×	×	1	-	×	×	×	✓	✓	×
[60]	Coverage	×	×	80	×	Greedy	×	×	×	×	×
[61]	Coverage	×	×	1	-	Greedy	×	30 dBm	✓	✓	×
[62]	Sum rate	×	×	1	-	Greedy	×	30 dBm	✓	✓	×
[63]	Sum rate	✓	×	1	-	×	×	-	×	×	✓
[64]	Sum rate	✓	×	4	×	Greedy	✓	20dBm	✓	✓	Complexity
[65]	Sum rate	×	×	6	✓	Greedy	×	10dBm	✓	✓	Complexity
[67]	Energy	×	✓	4	×	×	×	×	×	×	×
[68]	Energy	×	×	2	✓	Greedy	×	×	✓	×	×
[69]	Energy	✓	✓	1	-	×	×	27 dBm	✓	✓	Complexity
[70]	Energy	×	×	10	✓	Cell	✓	27 dBm	✓	✓	×
[71]	Energy	✓	✓	1	-	×	×	×	×	✓	×
[72]	Energy	✓	✓	1	×	×	×	Variable	×	✓	×
[73]	Energy	✓	✓	36	×	Cell	✓	Variable	✓	✓	×
[74]	Energy	×	✓	1	×	×	×	Variable	✓	✓	×
[75]	Energy	✓	×	5	✓	Greedy	✓	Variable	✓	✓	Complexity

Table 2.1 The summary of papers related to the deployment of UAV-BSs.

from a singe UAV-BS to multiple UAV-BSs. For example, a UAV serving in a cyclical way was designed in [69]. Hua et al. [72] jointly considered the UE association, UAV trajectory, and power allocation. With the assumption that UAV-BS served one user at any time, the work in [71] applied simulated annealing (SA) algorithm to solve the routing problem for one UAV. For multiple UAV-BSs, Mozaffari et al. [70] used iterative methods to solve the deployment problem and provided wireless service with minimising the power. Another example is in [76]. These works optimised not only the trajectory but also the hovering locations. However, the transmission power is regarded as a constant value (e.g., 27 dBm) or ignored. Such an assumption makes the optimisation problem easier but not generalises for wider applications. Zhang et al. [73, 77] regarded the transmission power as a variable and satisfied users with basic data rate. They provided a direct way to determine the 2D deployment by using a Gaussian mixture model to predict the future traffic distribution, then using clustering to divide users. The problem is that this solution can hardly be the optimum. By using iterative method, Mohammad et al. minimized the sum downlink transmission power [74], uplink transmission power [75], and total power consumption [78], respectively. More examples using iterative method can be found in [79]. Generally, the above works selectively considered minimising the transmission power and/or the propulsion power through optimising the hovering locations, the UAVs' trajectory, and inter interference, but neglected

another two important factors, user association and time efficiency. First, the user association allocates resources properly to all the users, not connects to nearby users, but shares resources to distantly connected users. Second, the fast UAV sending could hardly be guaranteed in many emergent scenarios. When that scenario comes, the deployment strategy is computed then the UAV-BSs are sent. It is required to immediately dispatch UAV-BSs with no need of complicate computation. These will be the new challenges.

Summary and Lessons Learnt:

- UAVs were firstly used as relays in broadcast communications. Then, their nature of flexible mobility expands the utility to the cellular network for emergency communication support. In 5G and beyond, the UAV-BSs have been regarded as necessary components to complete the structure of heterogeneous networks. The key challenge is to design an autonomous deployment/placement strategy for optimising coverage, sum rate, and energy.

- Many researches design the strategy aiming for maximising coverage or throughput but neglecting the reality that UAVs' battery capacities are very limited. To improve the lifetime of service, the energy-efficient deployment works are reviewed. There are two different modes of optimisation, one is calculating the optimisation after-dispatching, the other one is before-dispatching (proactive). The proactive one reduces the huge information exchange overhead between UAV-BSs and the central controller to improve energy-efficiency.

- The propulsion power is considered while optimising the trajectory. If the UAV-BSs work as hovering BSs, the transmission power becomes more important because the propulsion power is approximately a constant value.

- Interference exists when there are more than one UAV-BSs. Currently, there are two ways to build the system model with or without interference. First, the model without interference is named interference-limited method, which is used in complicate scenarios to reduce the complexity. This method assumes the interference-management methods are well deployed, like FDMA and beamforming. Only noise exists in the

system model. This is an ideal assumption aiming for simplicity, so it will be improved to noise-limited method. The works in this mode are [64][67][73]. Second, the noise-limited method considers both noise and interference. The data rate is gained based on SINR. Typical literature are [65] [68][70][75].

- Recent researches still face long computation time while facing complex scenarios. The **fast UAV sending** can hardly be guaranteed in many emergent scenarios. Further researches need to achieve not only energy-efficiency but also time-efficiency.

2.2 Review of UAV-BSs Aided Cellular Network Off-Loading

Let's start from the cellular network offloading. In 2009, H. Claussen and D. Calin from Bell Laboratories proposed the first related work of macro-cell offloading [80][81]. They selected small cells (e.g., femtocells) to offload indoor users from macro-cells and improve their radio channel conditions. Similarly, the load can also be transferred from macro-cells to WiFi connections. In 2010, B. Han et al. proposed two papers [82][83] to explain the necessity to offload 3G to WiFi. At the same time, K. Lee et al. [84] did a real-world off-loading experiment and achieved around 65% total cellular traffic offloaded and 55% reduction of battery energy usage. However, due to the explosive increase of user devices and the fast-changing spatial traffic, these **static nodes** (e.g, terrestrial BSs and WiFi nodes) become unable to achieve the ambition to maintain high performance and low cost at the same time.

In contrast, the UAV-BSs are flexible nodes to assist the terrestrial BSs to temporally offload the data traffic. In detail, traditional terrestrial BSs are deployed with satisfying peak traffic demands because it is fixed to its location and should satisfy the requirements of the most conditions. This setting becomes unnecessary during the low-traffic periods, which leads to low profit ratio and high operational cost. In that case, deploying **temporal UAV-BSs** for offloading the temporal peak traffic becomes a promising solution that the terrestrial BSs do not need to prepare the extra resources [4]. The UAV-BSs can also benefit the ground users with high-probability LoS links and timely communication assistance [85].

As stated in [86], the UAV-BSs aided traffic off-loading has become a necessary component to improve current cellular architecture in beyond 5G or even 6G.

In order to design an algorithm to automatically deployment the UAV-BSs for offloading, one research direction is to mathematically describe the problem as a mixed-integer non-convex problem. Mixed-integer means the some decision variables are constants, like the number of covered users should be a constant. Non-convex represents that the objective or the some of constraints are non-convex, so such an optimisation problem can have several local optimums. The 3D locations of UAVs, user association, and resource usage can be the decision variables. And throughput maximising, coverage maximising, or offload maximising can be selected as the objective respectively. Then, this problem can be decomposed to simpler sub-problems to search a solution for each sub-problem, and final solution is gained by iteratively solving the sub-problems. For example. in 2018, F. Cheng et al. [87] designed the **UAV-aided offloading** algorithm by transforming it into a mixed-integer non-convex problem. They set the objective as maximising throughput for serving cell-edge users and the decision variables as the UAV-BS's location. In the same year, J. Lyu et al. updated a related one-UAV work [88] by adding bandwidth allocation. Due to the problem's complexity, these researches chose not to consider the conditions with multi UAV-BSs. In the case with multiple UAV-BSs, the performance is mutually influenced by changing the locations of UAV-BSs, resource allocation (e.g., bandwidth), and selecting amount of UAV-BSs. Obviously, this joint optimisation problem is Non-deterministic Polynomial-time hard (NP-hard) and becomes more difficult to find solutions in polynomial time.

One solution for the above NP-hard problem is the heuristic algorithm. Such an algorithm is designed for combinatorial problems which aim at finding a group of finite set of decision variables that satisfy the given condition. For example, the locations and the user-association variables can be the finite set of decision variables, and the aim is finding an assignment of these variables to achieve both maximising the served users and minimising the resource used. Fortunately, the heuristic algorithm achieves this in a moderate computation time. This kind of algorithms include but not limited to Simulated Annealing, Evolutionary Algorithm, and Particle Swarm Optimisation. They commonly start with heuristically generating some

potential solutions. Then, these solutions are simulated and iteratively accepted if one of them has a better performance. When the maximum iteration is reached, the algorithm stops and outputs a final winner. The UAV-BSs can follow the winner's instructions and deploy. Examples can be found in the following works with diverse targets, such as maximising served users [89], maximising coverage [90, 2], or minimising number of drones [2, 91]. In detail, the work [91] optimised 3D UAV-BS development using the evolutionary algorithm. And the works [92, 93] were the examples applying particle swarm optimisation.

Nevertheless, these researches still worked on single-objective optimisation. In contrast, the **multi-objective** optimisation owns wider applications but increased complexity. For example, the network operators would like to use as less UAV-BSs as possible to associate more users. Meanwhile, the resources cost should be kept as low as possible. Doing this trade-off can offer a systematical increase of profit-cost ratio. Recent researches design a utility function to consider the multiple objectives. For example, the utility function is reverse to the power consumption and delay of communication, so maximising the utility function means to reduce the power and the delay. Yu et al. [94] achieved this by minimizing the weighted sum of the service delay and UAV energy consumption. Yang et al. [95] minimized the weighted-sum cost including the delay and energy consumption. Another similar example is done by Liu et al. [96]. It is interesting that the methods they used to maximise the utility is different, which include successive convex approximation [94], deep learning [95], and deep reinforcement learning [96]. And the trend can be observed that deep learning becomes popular (these methods are reviewed in Table 2.5.)

As we know that UAV-BSs are used for **fast response**, the complex computation is negative to this advantage. That means even though the optimisation problem is solved, if the computation time has to be strictly limited to a low level, such a computation delay can still make optimised network configurations expire. As a result, applying the out-of-date configurations generates optimisation overhead which negatively influences the operational cost. In the worse conditions, the high complexity generates a high delay to be deployed in time. Extra component is required to upgrade current heuristic solutions, especially for the time-efficiency.

Summary and Lessons Learnt:

- UAV-BS aided offloading is developed from the macro cell offloading. It becomes popular in recent years because using a flexible UAV-BS to temporally cover hotspots is attractive to reduce OPEX. The performance of UAV-aided traffic offloading can be mutually influenced by the the locations of UAV-BSs, the bandwidth allocation, and the amount of UAV-BSs. If all of these are optimised together, the optimisation problem becomes time-consuming to solve. This is not acceptable for time-sensitive UAV-based services.

- The heuristic methods sacrifice the optimal solutions for time-efficiency. Recent researches have addressed many single-objective optimisation problems for maximising served users, maximising coverage, or minimising number of drones. It is still challenging for optimising all of them due to the high computation complexity. In that case, further researches not only need to measure the performance but also compare the computation complexities to show competitiveness.

2.3 Review of Load Balancing in Cellular Networks

Load balancing had been a well-developed definition in computer network before the arising of cellular networks. In 1989, M. Baran et al. proposed an influential work to balance network overloads in distribution systems to reduce system power loss [97]. This idea was introduced to the cellular network in 1997 by Sajal K. Das et al. [98]. This work achieved borrowing channel assignment from a 'cold' cell (low-loaded) to a 'hot' cell. In general, the load balancing is required to cope with the imbalance distribution of users' demand [36]. Specifically, its goal is handovering the UEs at the edge of overlapping or adjacent cells from congested cell to idle cell through optimising handover offset values [99], thresholds [13], and the number of handovers [100].

Along with the development of self-organising networks in the late 2000s, the load balancing was also developed to take advantages of Fuzzy logic **controllers** for auto-tuning

handover margins [101, 20]. However, in 2012, J. Ruiz-Avilés et al. found that the controller-based methods might have a potential occurrence of oscillations, which may cause re-overload occurrence for target cells [101]. Another shortage was found by Aguilar-Garcia et al. in [15] in 2016 that the controller-based methods are reactive to random traffic spike. This causes a delay of the convergence and results in the limited ability in adapting the fast-changing load.

To solve the first oscillation-problem, two types of works were proposed using machine learning for deriving, predicting, and adjusting. (1) The cell load is modelled by machine learning for prediction. The machine learning methods are selected to forecast the cell load and decide the offsets according to their correlation. The offset could be adjusted automatically based on the cell load and the minimisation of packet loss ratio. For example, in [23], they used the polynomial **regression** to formulate the relationship and adjust the small cell offset value. However, the relationship between parameters can be complicated and related to a lot of parameters, which is challenging for the regression to deal with. (2) Reinforcement learning performs better than polynomial regression in complex scenarios. Q-learning is a **reinforcement learning** to solve the problem by learning the state-action table from training data. The state is indicated by the cell load, and the action represents the optimisation decisions (e.g., offset values or antenna down-tilt). With the employment of this model, the Reference Signal Received Power margin can be continuously adjusted according to the state-action table to maximise the user QoS. The works [26, 102, 103] followed this way to self-tune the cell margin or the antenna down-tilt. Beside the above methods, other methods also have been researched, such as game theory [104], reinforcement learning [105], and convex optimisation [106]. The above algorithms can relieve the pressure of oscillations and re-overload. Nevertheless, these researches still did not consider the modelling of users' mobility.

To take users' mobility into consideration, the context-aware module collected users' locations, then output the load difference between the loaded cell j_1 and its nearest idle cell j_2, this load difference is denoted as $\mathrm{LR}_{\mathrm{diff}}(j_1, j_2)$. $\mathrm{LR}_{\mathrm{diff}}(j_1, j_2)$ suggested the loaded cell j_1 to increase its speed to share the load to its nearest idle cell j_2. For example, the loaded cell j_1 reduced its transmission power by $k_j \times \Delta P_{\mathrm{TX}}(j_1)$. Such power reduction $\Delta P_{\mathrm{TX}}(j_1)$

was adjusted according to the controller's output. Besides, the algorithm could adjust the strength k_j to enhance or weaken the effect of load balancing algorithm according to the prediction of environmental changes. In that case, the cell load would be balanced according to users' mobility. Aguilar-Garcia et al. did this work in [15] and proposed a heterogeneous data-driven distribution-aware **indoor** load balancing study. Compared with the performance of Fuzzy controller-based optimisation, the context-aware optimisation could reduce at least 1.3% more user dissatisfaction rate, and nearly 50% convergence time.

As the above design is proposed for the indoor scenarios, further studies can improve this work by designing schemes for **outdoors**. In the outdoor scenario, real-time video analysis for locating users is expensive, Global Positioning System (GPS) data can be an alternative data source. Besides, Software-Defined Networking provides a chance to design different modes of optimisation to fit diverse services. For example, when the system receives an alert that a popular event is coming, the load balancing can automatically configure itself before the event. That can give a smooth transfer from low- to high- loaded, not to be over-loaded.

Summary and Lessons Learned:

- The summary of above proactive load balancing works, learning methods, required parameters, and pitfalls to avoid are proposed in Table 2.2.
- The cell offset needs to be determined according to the learnt correlation between the offset and the cell traffic subjecting to achieving the minimum packet loss. The polynomial regression and the Q-learning are the commonly chosen tools, where the cell load is an input, and it outputs the adjustment of the offset.
- One problem of using machine learning is that the regular conditions and the random components are mixed in the modelling, which causes a slow convergence in an anomalous condition (because the anomaly is not fully learnt). In that case, the **anomaly detection** needs to be implemented here to learn the anomalous traffic and alert the system to fit it.
- Currently, the context-aware module is not well-designed yet due to the **prediction errors** and the difficulties to quantify the cost of taking risks to optimise network

Proactive Load Balancing Works	- Determine RSRP margin based on number of active UEs to minimise the packet loss ratio [23] - Determine BS transmit power and RSRP margin based on CBR and OR to maximise reward of LB[24][107] - Determine RSRP margin based on cell load to maximise reward of LB [26][108][102] - Determine RSRP margin based on UEs' and BSs' distribution to maximise reward of LB [109] - Determine antenna down-tilt based on cell traffic [103]
Learning Methods	1) Polynomial regression [23] 2) Q-learning[24][26][109][102] 3) Unsupervised learning [28] 4) Neural network [108]
Required Parameters	1) Number of active UEs, cell load [23, 26, 108] [102, 103] 2) CBR, OR [24][107] 3) UE and BS distribution[109]
Online Data Analysis	1) Prediction of spatial-temporal network traffic [1] 2) Prediction of UEs' distribution [110] 3) Prediction of high-load [15]
Suggested Solutions	1) Cold-start controller =>Start with future network traffic prediction 2) Active all neighbour BSs for LB =>Active the hotspot-nearest BS for LB 3) Cellular data learning =>Heterogeneous data learning 4) React to a burst of traffic demand => Predict events and reduce the range of cell in advance

Table 2.2 A summary of proactive load balancing according to the focused parameters.

following the predictions. One method to avoid the potentially heavy cost is to use a parallel model to operate proactive optimisation while reserving the chance to be shifted back to traditional optimisation for minimising the risks. This method is reliable but does not quantify risk in a probabilistic framework. Another way is quantifying the cost based on the posterior distribution of predictions. It is a promising direction to address the concerns of overhead by the Gaussian Process or deep Gaussian process.

2.4 Review of Methods to Gain Context-Awareness

According to previous review of network optimisation methods, it is not difficult to find that improving the optimisations' efficiency requires users' context, such as geo-locations, predicting their behaviours, or forecasting cellular traffic. There are several tools have been used for this job, including data analytics and machine learning. To provide a general understanding of them, this section will classifies and reviews these methods.

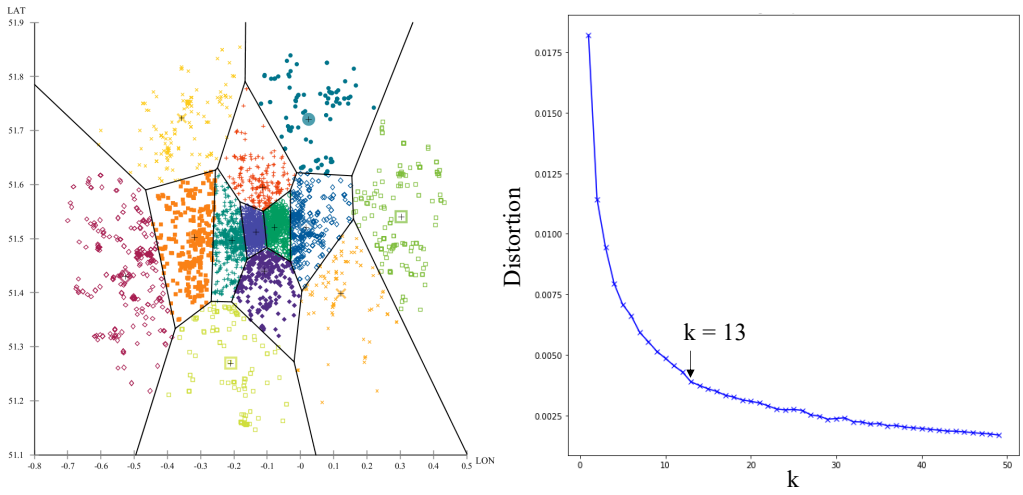
2.4.1 Modelling Geolocation

Geolocation represents the real-world measured locations of users, which offer spatial traffic distribution. It contains three components, observation time, moving objects, and geolocation records [111], which can be mined from data of GPS, Base Stations, and landmarks.

Popular Region

A popular region is a specific place with the potential to generate high communication traffic where a group of location records gather around a centre at a particular time. This region can be attractive all the time, such as commercial and tourists areas. In [1], the authors present the spatial correlations between 3G traffic and population density. This verifies the hypothesis that popular regions (with high population) have high probabilities of generating high demand. The network optimisation schemes should allocate resources in these areas to satisfy the imbalance traffic distribution, especially during events.

In the network optimisation, the prediction of the popular region represents the upcoming hotspots. It will benefit resource deployment [1], load balancing [110], and caching [112] by finding the place with high demand. The time requirement for hotspot prediction generally needs to be two hours in advance [1, 110] due to the achievable high accuracy (correlation$>$ 0.85 in [1]). In contrast, the geographic resolution requirements for hotspot prediction depend on the requirements of different network optimisations. For example, in [112], the predicted resolution of hotspot decided the flying height (332 m) of the flying BSs for proactive caching. Moreover, the work [110] achieved a load-resource matching with a 120-meter resolution.



(a) K-means result with 13 clusters (b) Elbow method to determine the number of clusters

Fig. 2.1 An example of using K-means.

The popular regions (hotspots) are usually modelled by clustering, such as k-means in [112]. It is to maximise similarity in the same group and guarantee that the assigned objects in different clusters are as different as possible.

K-Means based Spatial Model K-means is a simple but powerful method to divide data into groups. If the data have two dimensions, K-means will allocate centroids to each cluster and calculate the distance to each centroid, each data point will be allocated to its nearest centroid. So it is a centroid-based and distance-based method. In detail, researchers need to manually choose the number of clusters (k), then the algorithm groups coordinates (e.g., GPS data) according to k centroids cooperated with map information. In this thesis, the author uses K-means on a 2-dimension coordinates data and set number of clusters $k = 13$. The algorithm stops when centroids locations converge. The final result is shown in Fig. 2.1 (a). In this figure, the data are coloured according to their clusters and the region has been divided into 13 cells. If the distortion is defined as the sum distance from each point to its centroids, Fig. 2.1 (b) is generated when k is changed from 1 to 50. And it indicates that 13 is a proper k value because the distortion and k are all in a low level. This methods becomes useful when we want to deploy base stations. For example, the users in an urban area can be clustered into different groups to guide the location and height of UAV-BSs to cover them (see the

research in [112]). This method has been widely used due to its simplicity and effectiveness. However, the k has to be manually determined, and the cluster range is out of control. To determine the number of clusters and a cluster radius, D. Ashbrook and T. Starner [113] tried to use a variant of K-means which simulated radius regarding cluster numbers and picked the k at the convergence starting point. In fact, popular regions' ranges can vary a lot in both size and shape, which requires automatic range optimisation and methods to reduce computation cost. Besides, K-means cannot avoid the influence of noise data. In that case, another method named Density-Based Spatial Clustering of Applications with Noise (DBSCAN) emerged.

DBSCAN based Spatial Kernel The DBSCAN is a density-based algorithm that groups the points with many nearby neighbours and ignores the points lying along in low-density area as outliers (noise). This model requires no prior knowledge of clusters and no radius and results in fitting cluster shapes. Researchers choose only a minimum range and the minimum number of points in this range. Then a cluster with a minimum density is generated with arbitrary shape. A brief process of DBSCAN is presented in Fig. 2.2 explained as the following steps:

1. In the first step, each point has a range with a radius ε. In the ε radius circle, the core points in clusters own neighbours $\geq$ minPoints (which is 2 in this example). The edge points hold neighbours but $<$ minPoints. Also, the noise has no neighbour in the circle.

2. Then, the DBSCAN ignores the noise and clusters the data points into different groups (dash line circles).

3. Finally, this work calculates the centroid which is the mean of all the locations of points in the cluster. Moreover, the Voronoi diagram visualises the boundaries of different clusters.

For instance, Section 6.4.2 in this thesis applied DBSCAN on the geo-locations of GPS data for clustering. The work [114] also used this method to search popular regions considering the diversity of users and adaptive density. Popular regions, like the city of London, own denser small clusters indicating the high traffic demand. Even the spatial clustering

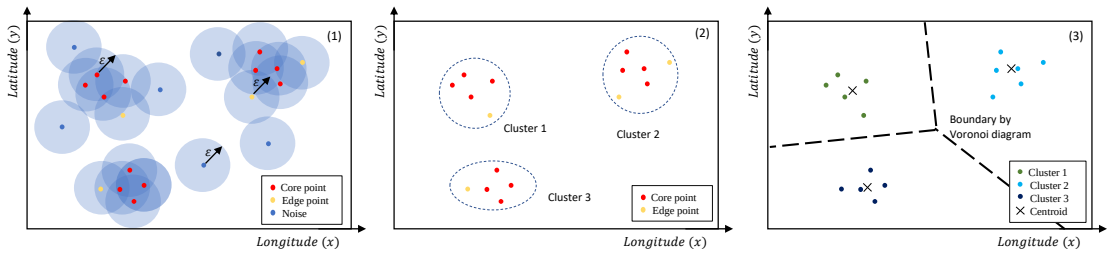


Fig. 2.2 The process of using DBSCAN to partition the region into clusters according Tweets density.

algorithms are widely applied to find popular areas, only clustering in one dimension was not enough. The temporal dimension is also needed for recognising sub-areas for the evolution of events. Therefore, researchers considered using a spatial-temporal model.

Spatial-Temporal Clustering Model Geolocation clustering has two main sub-categories, spatial clustering and temporal clustering, which make the objects gather regarding both dimensions (location and time). This thesis needs to consider the temporal dimension to find changing popular areas along with time. For example, the temporal dimension can be added to an extension of DBSCAN to take time changes into account to separate regions in both space and time. K. Tamura and T. Ichimura proposed an example work in [115] by analysing Twitter data. Based on this model, social events can be detected from the social networks, where users' sentiment can also be considered to locate the QoE blackspots.

Event-Detection based Model A place with an attractive event becomes popular in a particular period. Detecting events means to retrieve necessary information of a planned public occasion, such as schedule, topics, and attendance. Thanks to the online information, the occurrence of events can be automatically detected [116, 117]. Statistic method is chosen to forecast the regularity and the events. In detail, the city region can be partitioned into sub-areas by clustering. Then, in each sub-area, a geographical regularity estimation was executed, it was the usual condition of crowds moving pattern. Finally, the statistic method, such as box-plot, was chosen to find out the outliers. For example, in [117], R. Lee and K. Sumiya developed such an event detection algorithm to identify festival occurrence through

analysing the Twitter data. This thesis selects this method to alert the network about the upcoming traffic changes. The basic event-detection method can also be updated in future works. The extracted features of events can be incomplete while using only one data source. To improve in this aspect, H. Becker et al. proposed an approach [116] for identifying scheduled events from not only the social networks (e.g., Twitter, Flickr) but also media hosting site (e.g., YouTube). Another challenge is that the majority of online data is not geo-tagged, which limits the upper boundary of the detection precision. K. Watanabe et al. proposed a real-time local event detection system in [118] using both geo-tagged and non-tagged Tweets. The developments in event detection will help improve the used techniques in this thesis in the future works.

Summary of Findings and Lessons Learned

In summary, the main findings and lessons learned from the modelling of geolocation include:

- The popular regions indicate the hotspots distribution in a spatial traffic pattern. The proactive optimisation needs this context to decide the most profitable region for resource allocation and infrastructure deployment.
- These clustering methods, K-means and DBSCAN, are widely chosen in the popular region modelling. The K-means is easy to implement with low complexity but requires a manual selection of the cluster number k, leading to a degree of arbitrary parameterisation based on user bias/intuition. Some variants of K-means can mitigate this problem by re-simulating a series of k values, but still meet the negative influence caused by noisy samples. In that case, the DBSCAN based spatial kernel is selected to reduce the noise and highlight the high-density areas. The setting of minimum density determines the popular regions that can be found.

2.4.2 Predictive User Behaviour

The user behaviours become predictive when they are repetitive. The predictive models represent the **seasonality** in users' mobility because the demand components have periodic

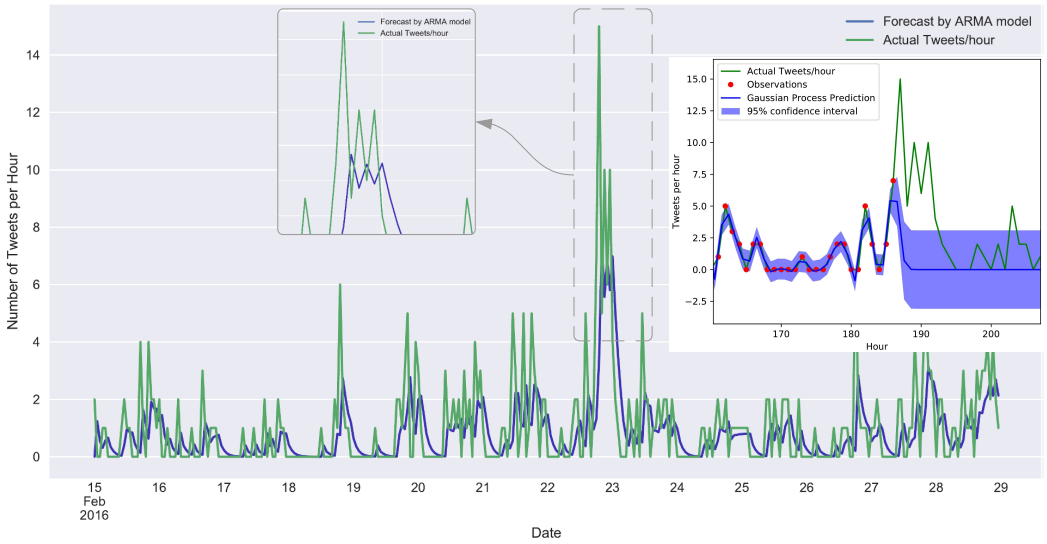


Fig. 2.3 An example of using ARMA and Gaussian Process to fit temporal Twitter traffic.

variations, such as working hours and daily transportation. It has a low possibility to change, which brings convenience for prediction. To track the habits in network usage, it is designed to forecast the regular spatial-temporal pattern according to historical data [119][110]. Nevertheless, the challenge appears when the network traffic does not follow the seasonality. In that condition, random components break the rules of the seasonality and cause prediction errors, which deserve research attention.

Anomaly Detection in User Behaviour

When the anomaly (irregularity) exists in user behaviour, where traffic-burst randomly occurs on the timeline, it is difficult to model such behaviour along with time. For example, Fig. 2.3 plots the number of geotagged Tweets per hour around a cinema in Leicester Square, London in two weeks. A movie world premiere caused an unusually high peak on 22/02/2016. It is difficult to predict such an event according to the regular traffic in previous days (15/02/2016-21/02/2016). To verify this view, the author uses an Auto-Regressive Moving Average model (ARMA) and a Gaussian Process (GP) model to forecast future confidence interval based on the data from 15/02/2016-21/02/2016. In the Gaussian Process model, the prediction of Tweets per hour y_* is based on the observations $\mathbf{y}$ before 18:00, 22/02/2016. Therefore,

the probability follows a Gaussian distribution $y_*|\mathbf{y} \sim \mathcal{N}(\bar{y}_*, \text{var}(y_*))$ [120], in which $\bar{y}_*$ is the mean indicating the best estimate of y_* and $\text{var}(y_*)$ is the variance representing the uncertainty. The 95% confidence interval is $\bar{y}_* \pm 1.96\sqrt{\text{var}(y_*)}$. This model helps understand seasonal characteristic in historical data. However, the burst on the event day does not obey the seasonal trend.

The key to solve is using online contents to detect the future popular event. The event is the dominating disturbance that enters early in the process of Tweeting. Therefore, the ARMAX (ARMA with exogenous terms) model can be used to track the irregular burst caused by the event. The events are regarded as unusual outliers in the regular traffic pattern. Therefore, it is needed to find irregular conditions in regularity. One can use machine learning, such as support vector machines (SVM), to classify the Tweets according to temporal-spatial-textual dimensions. Then the algorithm detects upcoming popular events as well as predicts the irregular behaviour of UEs. The paper [15] proposed a context-aware load balancing based on predicting an event in simulation, and [121] also studied that unexpected real-time road traffic prediction and control based the Tweets by waiting drivers.

Summary of Findings and Lessons Learned

- The non-periodic random components in the social behaviours cannot be predicted based on the training data in the regularity. In that case, the proactive optimisation has to be alerted with the newly emerging event's hotspots. This proactive action will increase the convergence ability of optimisation, but it also needs to deal with the prediction errors and the associated overhead. This thesis investigate this problem in Chapter V.

- The author summarises the reviewed papers in Table 2.3. This table classified the papers with the fields of used models, data types and amount. For example, the popular region prediction needs to satisfy the requirements of proactive load balancing with a minimum spatial granularity of 120 m and 2 hours ahead.

Table 2.3 References summary of online data type, amount, and analysing models

Context	Models	Data Type	Data Amount	Data-Driven Proactive Optimisation
Popular Region	-K-means [112][122][113] -DBSCAN [115][114] -Spatial-temporal Clustering [115] -Event-detection based model [116][117][118]	Social Network [112][1][115][122][114][116]	-2000 geo-tagged Flickr images [122] -86 million geo-tagged photos [114] -0.48 million Tweets [115] -21.6 million geo-tagged Tweets [117] -7135 geo-tagged Tweets and 2.45 million non-geo-tagged Tweets [118]	-Resource deployment: [1] hotspot prediction accuracy (correlation > 0.85), 2 hours ahead -Load balancing: [110] hotspot prediction accuracy (120 m), 2 hours ahead -Caching: [112] real-time hotspots clustering covered by flying BS at 332 m height

2.4.3 Cellular Traffic Prediction

Traditional cellular traffic prediction algorithms construct regression models for the one-step prediction in both of the core network [123] and the cell-level prediction [124]. However, these researches face bottlenecks to step further as the resolution is limited to cell-level. One of the solutions is analysing the high-resolution GPS data from heterogeneous datasets, then correlating it to cellular traffic. The work [1] has verified that the network traffic and the size of GPS records are both positively correlated to the number of involved network users. In that way, the GPS geo-tags could not only be used to predict flash crowds' needs but also offer operators suggestions about traffic forecasting for resource allocation [125]. This section follows the above milestones and reviews the developments of cellular traffic prediction.

Network-Level Traffic Prediction

The network-level traffic indicates the amount of exchanging information through the backbone network. Such data record the past traffic as a vector in the temporal dimension, which is the training data for neural networks. Then, the trained network forecasts the quantification of traffic at the next time stamp. The researches in [123, 126] followed this way by using a feedforward deep neural network or a Long Short Term Memory (LSTM) recurrent neural network. The results were satisfied in predictions, but they only provide the one-step prediction, which means that the network needs to be re-trained for multiple-step predictions. It may negatively impact the time for further optimisation. If the traffic seasonality and random spike can be decomposed in the training process, the multi-step traffic prediction can be transformed into a combination of seasonal prediction and adding external random information.

The Non-linear Auto-Regressive with exogenous model (NARX) makes predictions in this way and solve the one-step problem [127].

Cell-Level Traffic Prediction

This traffic includes both spatial (BS location) and temporal dimensions. The granularity of prediction is usually in hour-cell level to have a stable seasonality. In other words, based on the hourly data collected from the BSs, the cell-traffic will be modelled by statistic models or machine learning methods.

The **temporal** traffic consists of the trend, seasonality, and random components. In detail, the trend indicates the overall direction in which the traffic is developing or changing. The seasonality is that the traffic experiences regular and predictable changes which recur every calendar day or other periods. In that case, the cell-level traffic becomes predictable if the trend and seasonality are modelled, which can be easily implemented by ARMA model or exponential smoothing [128, 129]. However, these models only consider a constant time range, which is described by a 'window', so the long-term memory of all the training data is neglected. The LSTM is designed for solving this problem by feedback connections in Recurrent Neural Networks (RNN) to process the entire sequence of data with selectively remembering patterns. It has a forget gate to disable the meaningless information in recurrent states, such as the random fluctuations in the traffic pattern. One successful example is in [124], but this technique also has some limitations. One of them is that the knowledge learnt from one cell can not be shared with other cells, which is not intelligent with repeating training effort. It is a promising way to use meta learning to use the conclusions of other learning's results. The records of other learning methods will be stored and help current training in different cells for learning both temporal and spatial traffic.

The **spatial** traffic can be described by a probabilistic distribution whose parameters are adjusted for fitting the training data with minimum errors. The traditional method is to formulate using mathematical statistics, such as Zipf distribution [130]. This method finds relations between traffic and locations, and the significance of this relationship. For example, the work [131] designed an α-stable traffic model with parameter tuning for a city-wide

scale. The common shortage of this method is that it approximates the parameters without the optimisation like gradient descent so some important details will be ignored. With the development of machine learning, this shortage has been overcome by the neural networks (e.g., LSTM) which owns intelligent weights' fine-tuning methods like back-propagation [124]. Although the neural networks performed better in prediction accuracy, it lacks the ability for quantifying uncertainty as the mappings between layers governed by weights but not a stochastic process. The usage of Gaussian Process addresses this problem [124]. This non-parametric method trains its hyper-parameters to produce a posterior distribution of prediction with uncertainty quantified. Although the Gaussian Process may not surpass the performance of neural networks, it can quantify the risks via the posterior distribution. In that case, it becomes promising to use deep Gaussian process to couple the advantages of both deep learning and Gaussian Process [132].

Traffic Prediction Using Online Data

A common problem of the previously mentioned traffic prediction methods is that the behaviours are neglected, so it becomes difficult to explain and alert the **random** traffic **spike** caused by changed services. In that case, heterogeneous data become useful because they contain not only cellular information but also the individual-level geolocation and behaviours. In the general methods, the traffic is decomposed into the trend, seasonality, and random components. These random components may be the holiday traffic or the traffic spike during popular events. If such traffic is estimated, one will combine the quantification of the three components for the final prediction [133, 134].

This gap can be filled by estimating cellular traffic based on geo-tagged **social network** using machine learning, such as using linear regression [135][1]. Before training, it is required to select the interested dimensions first, such as cellular traffic and amount of Tweets. Then, the data will be fitted by regression models with minimising residuals like using least squares [135]. The work [135] found a strong correlation between Tweets and mobile traffic in a stadium even though the Twitter traffic takes only a small partition in the whole traffic. However, the correlation is not fixed when the temporal or spatial scale changes. The

strength of correlation increases along with the decrease of spatial resolution. It is a trade-off between better spatial resolution or higher correlation. The current method for this problem is re-calculating the correlation with different resolutions to pick an acceptable one [110].

Based on the positive correlation, network traffic becomes predictive using heterogeneous data. The regression methods shall provide the optimised parameters for the fitting correlations. One can formulate the model according to the parameters. For example, the work [1] predicted spatial-temporal traffic based on the estimation of correlation between 3G network load and Tweets using log-linear regression. This estimated **Down-Link traffic** load $\hat{r}_{\mathrm{DL}}$ in cluster o in time interval t can be described as

$$\hat{r}_{\mathrm{DL}}^{ot} = 10^{b_{\mathrm{DL}}}(\frac{n_{\mathrm{ot}}}{\tau})^{a_{\mathrm{DL}}}\tau \tag{2.1}$$

where $[a_{\mathrm{DL}} = 0.88kb/\mathrm{Tweet} \quad b_{\mathrm{DL}} = 2.37kbps]$ and τ is ratio between time interval and second (e.g, in this work $\tau = 3600s/\mathrm{hour}$). This formulation couples Tweets and cellular traffic but considers only general conditions. Sometimes, an anomalous traffic emerges without a holiday-like obvious signal. Therefore, an anomaly detection in traffic prediction emerges as another research direction.

Anomaly detection

The anomaly traffic does not follow the model of the trend or the seasonality because user behaviours become different during irregular conditions. The general method is to model the regular traffic first, then detect the **outliers** based on the modelled regularities. Finally, the outliers will be treated as a particular group with another model to fit the traffic. In that case, two machine learning models are needed for both regular and anomalous conditions, which are usually combined with a clustering and a regression. The clustering methods automatically distinguish the regular and anomalous conditions in the selected dimension. For example, using K-means on grouping the BSs with similar traffic will present the BSs with extremely high or low load [136, 137]. The extreme values are useful for proactive optimisations, such as load balancing for extremely high-load cells and BSs turn-off for extremely low-load cells. Another model for modelling anomalous traffic is usually undertaken by regression

Table 2.4 Summary of the literature about network traffic prediction.

Reference	Year	Spatial Scale Country: △ Province: □ City: *o*	Temporal Scale L: Multi-hour M: Hourly S: Minutes	Traffic Type Online Traffic: O Cellular Traffic: C	Prediction T: Temporal S: Spatial TS: Both	Decomposition	Model	Section	Suggested Solutions
[123]	2018	△	M	C	T	×	Deep Learning	2.4.3.1)	- Multi-step prediction
[126]	2017	△	S	C	T	×	LSTM		=>NARX model
[128]	2007	△	M	C	TS	✓	Exponential Smoothing	2.4.3.2)	- Long-term selective memory
[130]	2011	△	M	C	TS	×	Statistics		=> LSTM
[129]	2016	*o*	M	C	TS	✓	ARMA		- Knowledge share between BSs
[131]	2017	*o*	M	O	TS	×	α-stable		=> Meta learning
[124]	2018	*o*	M	C	TS	×	LSTM		- Uncertainty Quantification
[138]	2018	△	M	C	TS	×	Neural Network, GP		=> GP
[139]	2011	□	M	C	TS	×	Markov Chain	2.4.3.3)	- Social behaviours
[133]	2017	*o*	L	C	TS	✓	ARMA, Decision Tree		=> Heterogeneous data analytics
[1]	2016	*o*	S	O	TS	×	Statistic		- Estimation of random spike
[110]	2017	*o*	M	O	TS	×	Statistic		=> Anomaly detection
[135]	2017	*o*	S	O	TS	×	Regression		
[136]	2017	*o*	S	O	TS	×	K-Means, Neural Network	2.4.3.4)	- Avoid local optimum of gradient
[137]	2018	□	M	C	TS	×	K-Means, GP		descent
[127]	2018	□	M	O	TS	✓	K-Means, NARX		=> Neuro-evolution deep learning

methods, such as Gaussian Process, neural networks, or NARX model [127]. These methods performed well but faced a challenge in optimising the weights that they can not avoid the local optimum with using gradient descent. This is because the start of the gradient is randomly allocated in the global space. It is not controlled to walk iteratively to a close local optimum then converge. One of the solutions is applying evolution algorithm in parameters' optimisation. The scheme is inspired by biological evolution in which the generations will finish the procedures of reproduction, mutation, recombination, and selection. The mutation provides chances to jump out of the local optimum. In that case, it is promising to use neuro-evolution deep learning to better tune network weights.

Summary of Findings and Lessons Learned

In summary, the main findings and lessons learned from the traffic prediction are highlighted as the following items. The author also provides a summary Table 2.4 to compare the methods and suggest the solutions of current pitfalls.

- The network-level traffic prediction can be addressed by the **regression** methods. These methods can provide high-accuracy results in the **one-step** forecast but accumulating errors for multi-step prediction. The problem is that the network optimisation requires multi-step prediction to reduce redundant re-training efforts. The solution is using the NARX model regarding random spike as exogenous inputs and combine the multi-step

seasonality prediction with the exogenous information. In that way, it alleviates the influence of errors caused by random components.

- For the temporal dimension, the traffic is **decomposed** into the trend, the seasonality, and the random components. The first two items are predictable through using the ARMA or the exponential smoothing, but only part of the training data is used to deduce the prediction. Such a requirement about flexible long-term memory makes the LSTM a feasible choice. Its forget gate is trained to remember the meaningful items. Moreover, one of the future researches is to avoid the knowledge catastrophic forgetting between BSs by meta learning.

- In the spatial traffic prediction, traditional methods modelled it by mathematical **statistics** (e.g., Zipf distribution and α-stable). Compared with machine learning (e.g., neural networks), the traditional ones have no parameter optimisation (e.g., gradient descent). Instead, the parameters are determined using general approaches, such as maximum likelihood estimation which approximates the parameters without finding a path to the minimum gradient. The problem of current machine learning is that the predictions are generated without a quantification of the **uncertainty**. It causes difficulties for future decision makings to quantify the cost and profit considering potential errors. Such a problem is estimated to be solved by Gaussian Process or deep Gaussian process to produce predictions as posterior distributions (uncertainty).

- The random components in traffic are difficult to explain and predict due to the lack of user behaviours meta-information. This difficulty drives data analytics from cellular only to heterogeneous data (e.g., Twitter data). Current methods concentrate on using linear regression to formulate the positive correlation between cellular traffic and social network. Some statistic methods can quantify the strength of correlation but not formulate the model. Current gap of predicting the random traffic spike is that many anomalous conditions are unknown in advance (e.g., non-periodic events such as protests). It needs the **anomaly detection** to distinguish regular and anomalous conditions automatically. The general method is combined with a clustering method (for

distinguishing) and a regression model (for traffic modelling), such as a combination of K-means and neural networks. However, the weights selection in the neural network may meet the local optimum by the gradient descent. In that case, the neuro-evolution deep learning can jump out of the local optimum, which is a promising method to model traffic with fine-tuned weights in the long-term.

2.4.4 Overview of Data Analytics and Machine Learning

Previous sections have reviewed literature from two aspects: the network optimisations and the context-awareness. The commonly mentioned many machine learning methods, data analytics, and prediction error problems. This section will review these common methods, some of which have been used in this thesis. The author also expands the review to provide a general view of all the tools assisting CAPO.

Data Collection

Here, the author will mainly introduce how to collect online data from Internet. The online data consists of three major categories, social network (e.g., Twitter and Facebook), media hosting site (e.g., YouTube and Instagram), comments and reviews (e.g., e-commerce reviews and topic talk).

The traditional method for collection is using an API. The **API-collection** method is widely used as 'Search API' and 'Stream API'. It allows automatic data collection from service providers in an economical way. The search API and the stream API are all in this category. However, it still has some challenges, such as poor efficiency and difficulty in gathering historical data. Also, service providers could limit the collection.

Building **own datasets**, such as collecting from volunteers or cooperating with service providers, can avoid the above limitations. This method provides a nearly complete dataset and achieves a flexible setting of the environmental variables. A popular method is sending friend requests to other users with a statement that researchers are collecting the data for researches with privacy protection. The volunteers could feel free to accept or reject friend requests. For example, the researchers in [140] invited 19,484 users agreed to join

the experiment. Sometimes, incentives can improve the performance of collection that contributors who would be rewarded with payment according to their contribution[141, 142]. This method can be costly and requires ethical approvals, or one can use open datasets to reduce this cost.

The **open datasets** can reduce the cost in data-collection. Many organisations make available their datasets for transparency or research purposes, such as Kaggle datasets [143] and European Union Open Data. One typical example is the Italia Telecom operator dataset (see data from [144]). It is used to forecast cellular traffic pattern [145]. Some public projects will also open their data to other researchers. For example, an EC H2020 RISE Project DAWN4IoE has opened the datasets, such as cellular traffic data [146] and Twitter density [1, 147]. Researchers are required to choose their methods for collection and pre-process the raw data for further data analytics.

Machine Learning Techniques Overview

Scalable machine learning emerged in recent years as it gives the computer system an ability to learn user behaviour from online data. This work compares the most commonly used machine learning methods regarding proactive optimisation requirements in Table 2.5 and list the applications where these methods have been used. The complexity refers to the number of computation operations that should be performed to achieve the desired result. The training data and time indicate the required data amount and training efficiency. Then, the accuracy suggests the supposed performance that the algorithms generate. Finally, the evaluated levels (low/fair/high) is based on the previous literature cited after each name.

From Table 2.5, some common **characteristics** of the machine learning usage can be found. For example, the geo-location modelling usually requires unsupervised clustering methods, such as K-means and DBSCAN. This has the advantage of not requiring labels in a sophisticated problem setting and relies on topological features as a compressed representation of high dimensional attributes. However, the ill-defined nature of clustering means initial parameterisation is highly related to researcher bias or intuition. In contrast, social behaviours usually have limited categories, so supervised classification is commonly selected, such as

SVM and K Nearest Neighbour (KNN). For time series forecasting, regression and Markov methods are good at predicting network traffic. In network optimisation, more sophisticated methods are chosen, such as reinforcement learning and the neural network. However, many of the non-Bayesian methods face challenges of catastrophic forgetting and dealing with high-dimensional inputs. That requires more-advanced learning methods, such as deep Gaussian process, meta learning, deep reinforcement learning, and neuro-evolution deep learning. These approaches either provide quantitative uncertainty estimates, high-dimensional feature capture, and/or improved adaptation to the environment.

Methods for Addressing Prediction Errors

The machine learning approaches thus far produce prediction errors because of the **scarcity** of training data or the mismatch of prediction functions. The probability of error can be described by uncertainty in the predictions. In further optimisation, the overhead of the system will be incurred because of such uncertainty. This thesis draws lessons from other prediction and optimisation systems in other areas of science and engineering.

In prediction systems, such as climate science and structural mechanics, big data helps to inform the likelihood of outcomes of predictions that arise from dynamical systems. Probabilistic numerics translate input uncertainty into output uncertainty (e.g., probabilistic finite element). The uncertainty caused by data/estimation errors is required to monitor and control the computational overhead. In that way, the paper [162] provided an illustration of using the probabilistic numerics to describe the uncertainty with diagnosing error sources in computations. However, the Gaussian Process needs to be coupled with deep learning to face more complex tasks, so the deep Gaussian process emerges.

The deep Gaussian process acts as a deep neural network but with Gaussian Process governing the mappings between layers. It will give an empirical confidence interval to quantify the uncertainty. The higher uncertainty could mean a higher potential to cause overhead. In network optimisation, the forecasting associated with high potential of overhead could be discarded in the decision making. For example, the work [132] successfully learnt natural human motion by the Deep Gaussian Process even with scarce data. Besides the

uncertainty (overhead) quantification, a parallel system offers a useful structure to improve the robustness.

The parallel system owns a reliability-wise structure. It allows the system to function with any mechanism working. For example, if the network unexpectedly operates in a bias condition, it still has time to alter to a reactive optimisation. Such a method works as the parallel system as introduced in [163] to improve reliability.

Methods for Preserving Privacy and Data Utility

The privacy problem is critical in data analysis. The challenge is to gain high utility in data while ensuring confidentiality, integrity, and availability [164]. Besides, the network operators and the data providers should achieve not only encryption of all data but also a strict access control to avoid the unpleasant data collection, storage, and usage. In that case, the trade-off between data utility and privacy needs to be solved in three aspects.

Firstly, current users are usually unaware of the collection of personal data, which causes the anxiety about potential defraudation and hurt feelings. Appropriate notices and asking authorisation can relieve the anxiety during personal data collection. For example, in the Internet of Things, the users are notified about IoT privacy properties [165]. This 'right to know' alleviates anxiety and provides users with choice. An example is the current usage of Internet cookies (see cookies consent under the EU General Data Protection Regulation [166]), but it requires that consumers trust data storage and usage.

Secondly, the stored data must be provided with both privacy and authenticity. For this purpose, the encryption schemes will transform the data into a ciphertext with a symmetric-key mechanism to satisfy the two requirements (see Authenticated Encryption [167]).

Moreover, data protection not only needs to encrypt information but also protect them from attacks, which is a classification problem. For example, the classification of legal/illegal user can be achieved by utilising the radio channel information [168]. The work [169] used RNN to detect various attack variations, and the work [170] provides a panoramic survey of security in cyber-physical systems.

Finally, during the data usage, researchers need to protect sensitive latent information while reserving utility. Such a trade-off is studied in [171] by measuring data utility loss and latent-data privacy matrices. Another method to secure outsourced data analytics is by applying the homomorphic encryption [172].

Summary of Findings and Lessons Learned

In summary, the main findings and lessons learned from this sub-section include:

- Diverse Machine Learning tools are good at retrieving different contexts. Geo-location modelling is often an ill-defined unsupervised clustering challenge. In contrast, behaviour modelling is usually a supervised classification problem. Next, the traffic prediction can be addressed by the regression methods, including the polynomial regression, Gaussian Process. In the network optimisation, the parameters and principles become dynamic and numerous, so high-complexity methods (e.g., reinforcement learning and the neural networks) are becoming increasingly suitable.
- The prediction errors cause undesirable overhead in the proactive optimisation modules. It is necessary to quantify such overhead and also take into account other utility functions such as privacy and security.

Table 2.5 A summary of the reviewed machine learning methods with the usage position in Sections and a comparison of time complexity.

Machine Learning	Time Complexity	Description	Complexity	Training Data Amount	Training Time	Accuracy	Usage in Proactive Optimisation
K-means [148–150]	$O(KMNI)$	K: number of clusters M: dimension N: number observations I: number of iterations	Low	Low	High	Fair	Modelling Geo-location
DBSCAN [148, 149]	$O(Nlog(N))$	N: number of observations	Fair	Low	Fair	High	Modelling Geo-location
Support Vector Machine [150]	$O(N^2)$	N: number of observations	High	High	Fair	High	Modelling Social Behaviour Proactive Caching
Naive Bayes Classifier [151, 149]	$O(Nd)$	N: number of observations d: dimensionality of the features	Fair	High	Low	Low	Modelling Social Behaviour
K Nearest Neighbour [151, 149]	$O(Nkd)$	N: number of observations d: dimensionality of the features k: hyperparameter	High	High	Fair	Fair	Modelling Social Behaviour
Naive Bayes Neural Network [151, 149]	$O(nMPNe)$	n: input variables M: number hidden neurons P: number output values N: number of observations e: number of epochs	High	High	High	High	Modelling Social Behaviour
Linear Regression [150]	$O(NPI)$	N: number of observations P: number of variables I: number of epochs	Low	Fair	Fair	Fair	Modelling Social Behaviour Network Traffic Prediction Proactive Caching
Gaussian Process [152]	$O(N^3)$	N: number of observations	High	Low	Fair	High	Predictive User Behaviour Network Traffic Prediction
Deep Gaussian Process [132]	$O(N^3)$	N: number of observations	High	Low	High	High	Network Traffic Prediction
Long Short-Term Memory (LSTM) [153, 154]	$O(W)$ per time step	W: the total number of parameters in a standard LSTM network	High	Fair	-	High	Network Traffic Prediction
Polynomial Regression [150]	$O(NPI)$	N: number of observations P: number of variables I: number of epochs	Low	Fair	Fair	Fair	Proactive Load Balancing
Q-Learning [155, 156]	$O(S^2A)$	S: number of states A: number of actions	High	High	-	-	Proactive Load Balancing
Reinforcement Learning [155, 156]	$O(S^2)$	S: number of states	High	High	-	-	Proactive Caching
Neural Network [150, 155]	$O(nMPNe)$	n: input variables M: number hidden neurons P: number output values N: number of observations e: number of epochs	High	High	High	High	Proactive Caching
Echo States Networks [157]	$O(2TU/H)$	A recurrent neural network T/H: time of executions U: number of predictions	Fair	High	Fair	-	Proactive Caching
Convolutional Neural Network [158]	$O(\sum_{l=1}^{d} n_{l-1} s_l^2 n_l m_l^2)$	d: number of layers n: number of filters m: size of the output feature s: size of the filter	High	High	High	-	Proactive Caching
Transfer Learning [159]	$cO(n+m)$	c: time of calculating distance n,m: size of training set	Fair	-	Fair	Fair	Proactive Caching
Meta Learning [160]	$O(l+t/logt)$	l: length of program t: program halt time	High	High	Fair	High	Learn how to learn. Use conclusions of other learning.
Deep Reinforcement Learning [155, 156]	$O(S^2D)$	S: number of states D: deep learning complexity	High	High	High	High	Complex context perception Decision making
Neuro-evolution Deep Learning [161]	$O(n(1+logn)D)$	n: the problem size D: deep learning complexity	High	High	High	High	Avoid local optimum in using gradient descent

Chapter 3

Context-Aware Aerial Base Station Deployment

Overview

In recent years, unmanned aerial vehicle base stations (UAV-BSs) are widely viewed as a promising technique in the fifth-generation (5G) and beyond mobile networks to provide a reliable service for ground users. On the one hand, this joint UAV-BS deployment and user association (UDUA) problem has a high computational burden. On the other hand, it has to be solved with a restricted time. This work makes an early attempt to decrease the on-line computing time of UDUA problems. With the objective of minimising the system transmission power, this chapter formulates the joint UDUA problem into a mixed integer non-linear programming (MINLP) problem, which is challenging to solve directly, subject to the basic downlink QoS requirements of ground users. The author proposes a K nearest neighbour (KNN) based algorithm to calculate the proper UAV-BS deployment strategy for a new user distribution with the help of accumulated experiences in well-solved problems. This work evaluates the performance through extensive experiments. Numerical results show that the proposed UDUA mechanism can achieve near-optimal performance in terms of transmission power consumption and failure rate with enormously reduced computational time compared with existing UDUA approaches.

3.1 Introduction

Implying UAV-BSs poses some key challenges for both industry and academia. First, UAV-BSs are more flexible than terrestrial ones. As UAVs' positions and matching statuses to ground users have a significant impact on the system performance, intelligent UAV-BS deployment and user association strategy should be conducted according to practical user distributions. Second, energy-saving is critical for UAV systems due to the limited UAV on-board energy. An improvement in energy efficiency will directly increase the survival duration of the network, which is especially important in disaster recovery. Third, UAV-BSs are more expected to work in emergent scenarios with bursty traffic demands. Those UAV-BS deployment and user association strategies must be calculated in a time-efficient way.

The problem of UAV-BS deployment and user association (UDUA) has been widely investigated recently [62–65, 70–72, 69, 73–75]. Focusing on improving the system coverage or throughput, numerous UDUA methods have been proposed in [62–65]. However, none of them has considered the energy consumption of UAVs and user devices. With the objective of minimising the system power consumption, many energy-efficient UDUA strategies have also been leveraged for UAV-BS radio access networks [70–72, 69, 73–75]. Nevertheless, these UDUA methods need to conduct complex optimising algorithms to obtain the optimal or sub-optimal solution for each individual UDUA problem. Thus, a long on-line computing time will be taken before the UAVs are dispatched, and the fast network service can hardly be guaranteed through these strategies.

In this work, the author considers multiple UAV-BSs providing reliable coverage for ground users distributed in a certain area. Given the constraint of basic downlink quality of service (QoS) of each user, this chapter proposes a centralised joint UDUA mechanism to minimise the UAV-BSs' total transmission power. In order to decrease the on-line computing time, the proposed mechanism tries to acquire the proper UAV-BS deployment strategy related to a new user distribution with the help of accumulated experiences in well-solved problems based on machine learning, and then optimally associate the ground users to the UAV-BSs by solving an equivalent bipartite matching problem. The main contributions of this work are summarised as follows:

1. This chapter minimises the total transmission power of multiple UAV-BSs providing reliable coverage for ground users distributed in a certain area by jointly optimising the UAV-BS positions and the association of ground users. This work formulated the joint UDUA problem into a mixed integer non-linear programming (MINLP) problem subject to the basic downlink QoS requirements of ground users.

2. Since the user association can be performed when the UAV-BSs' positions are determined, this chapter decouples the joint UDUA problem into two sub-problems. One is the user association sub-problem, which acquires the optimal matching strategy between the UAV-BSs and the ground users for given UAV-BS positions. The other is the UAV-BS deployment sub-problem, which tries to find the best position combination of the UAV-BSs making the first sub-problem's solution minimal among all the possible ones.

3. This work proposes a centralised UDUA mechanism for the UDUA problem. In this mechanism, the user association sub-problem is transformed into an equivalent bipartite matching problem and solved with the Kuhn-Munkres (KM) method [173]. This method can address the bipartite matching (assignment) problem in polynomial time, which is also named Hungarian algorithm. For the UAV-BS deployment sub-problem, this chapter theoretically proves that adopting the best UAV-BS deployment strategy of a previous user distribution for each new user distribution will introduce a limited transmission power increase compared with the new user distribution's best strategy if the two user distributions have limited difference. Based on the proposed mathematical proven, this chapter defines the similarity level between two user distributions and K nearest neighbour (KNN) based algorithm to calculate the proper UAV-BS deployment strategy for a new user distribution with the help of accumulated experiences in well-solved problems.

4. This work evaluates the performance of the UDUA mechanism through extensive experiments. Numerical results show that the proposed mechanism can achieve a similar or even better performance in terms of transmission power consumption and

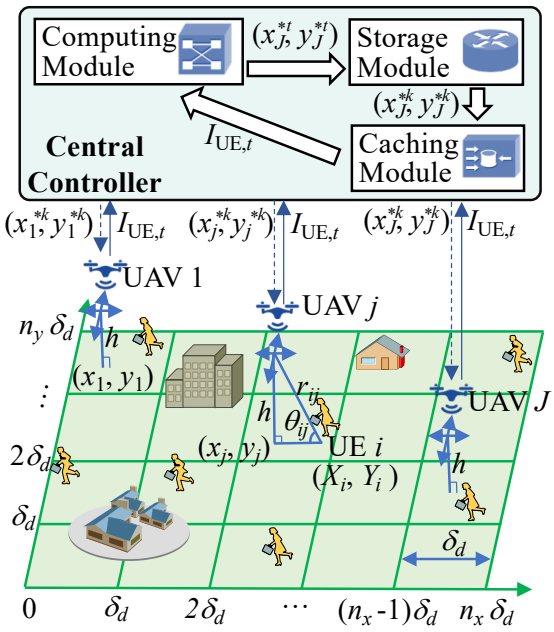


Fig. 3.1 The system model.

failure rate with an enormously reduced computational time compared with existing UDUA methods like Heuristic algorithms.

The rest of this work is organised as follows: In Section II, related works are reviewed. Section III provides the system model and optimisation problem formulation. In Section IV, the proposed UDUA mechanism is given. In Section V, this chapter provides extensive experiments about UDUA's performance estimation and comparison with other methods, and Section VI concludes the work.

3.2 System Model

The system model is illustrated in Fig. 3.1. This work considers a typical low-altitude UAV-BS radio access network (RAN) scenario where J UAV-BSs serve all the ground UEs over a certain region R, and these UAV-BSs are controlled by a central controller. This central controller is located in the local terrestrial base station which is equipped with the computing, caching, and storage modules. This work assumes that the UAV-BSs' backhaul uses different frequency bands compared with the fronthaul. Power-measurement sensors are enabled on

the UAV-BSs, the power consumption for communication system and propulsion system are collected and sent to the central controller. Since the downlink traffic is much higher than the uplink one in the usual multimedia communications [174], this work only focuses on the downlink transmission. The UAV-BSs use different frequency bands for the downlink, so the interference among UAVs is negligible [73]. The system model with interference will be used in the future work. Region R is further divided into $n_y \times n_x$ grids with the same size of $\delta_d \times \delta_d$. The author assumes that δ_d is small enough so that different ground UEs in the same grid have an equal distance from an arbitrary UAV-BS flying in the air [175]. Furthermore, this work assumes that the UAV-BS RAN works in time intervals, and the central controller has global information. At the beginning of every time interval, the central controller will first collect knowledge about UE distribution over the $n_y \times n_x$ grids and then calculate the optimal UDUA strategy in a centralised way. This work considers a quasi-static environment where the UE distribution is assumed to be fixed during an arbitrary time interval.

A UAV-BS possesses Φ orthogonal frequency division multiple access (OFDMA) sub-channels, each of which has a fixed bandwidth of B. During a certain time interval, a UAV-BS can construct a downlink transmission connection for one ground UE with every sub-channel, and a ground UE can be served by at most one UAV-BS. Taking advantage of proper spectrum management [176], this work assumes that the inter-UAV interference is well controlled and thus can be neglected. This work also assumes that ground UEs in the region R have the same external-interference condition with a constant noise power of σ_n^2 for analytical tractability. Being dispatched, the J UAV-BSs will hover in fixed positions with the same flight altitude of h. The author uses sets $I_{\mathrm{UE},t} = \{\mathrm{UE}_1, \mathrm{UE}_2, ..., \mathrm{UE}_I\}$ and $J_{\mathrm{UAV}} = \{\mathrm{UAV}_1, \mathrm{UAV}_2, ..., \mathrm{UAV}_J\}$ to represent the set of ground UEs located in the considered region at time interval t and the set of UAV-BSs, respectively. To guarantee that all the UEs can be served, this work assumes that $I \leq J\Phi$. This work uses X_i and Y_i to denote the ordinal numbers of UE i's ($\mathrm{UE}_i \in I_{\mathrm{UE},t}$) position grid in latitude direction and longitude direction, respectively. Taking into account the fact that UAV-BSs are generally utilised in scenarios like disaster aiding and crowd serving, this work guarantees the basic quality-of-service (QoS) for ground UEs with the minimum data rate requirement of C. For each UAV-BS $\mathrm{UAV}_j \in J_{\mathrm{UAV}}$, this work uses

variables x_j and y_j to denote the grid location of its ground projection, use Boolean variable δ_{ij} to denote its association relationship with UE i ($\delta_{ij} = 1$ if UE i is served by UAV-BS j, $\delta_{ij} = 0$ otherwise), and use variable p_{ij} to denote its transmission power to serve UE i if UAV-BS j and UE i are associated, respectively.

The air-to-ground pathloss describes the power reduction between a UE and its associated UAV-BS. It is combined with two main propagation groups, line-of-sight (LoS) and non-line-of-sight (NLoS), that depend on whether obstacles exist in the propagation path. For example, if the elevation angle between a UAV-BS and a UE approaches 90 degree, the probability of LoS condition is close to 1, which means the radio signal can hardly be obstructed. According to the paper [59], the pathloss in decibel can be modelled as $g^{\mathrm{LoS}}(\mathrm{dB}) = \mathrm{FSPL} + \mu^{\mathrm{LoS}}$, $g^{\mathrm{NLoS}}(\mathrm{dB}) = \mathrm{FSPL} + \mu^{\mathrm{NLoS}}$, where FSPL is the free space pathloss, and μ is the mean value of the excessive pathloss which is described by a Gaussian distribution. Considering an arbitrary pair of ground UE i and UAV-BS j during time period t, the average channel pathloss g_{ij} is calculated as the following equation [59]:

$$g_{ij}(\mathrm{dB}) = 20\log_{10}\left(\frac{4\pi f r_{ij}}{c}\right) + P_{ij}^{\mathrm{LoS}}\mu^{\mathrm{LoS}} + P_{ij}^{\mathrm{NLoS}}\mu^{\mathrm{NLoS}}, \tag{3.1}$$

where f, with the unit of Hz, is the frequency of of carrier signal, r_{ij} is the distance between UE i and UAV-BS j in three-dimensional space, c is the speed of light, P_{ij}^{LoS} and P_{ij}^{NLoS} are probabilities of the transmission link in LoS state or in NLoS state, μ^{LoS} and μ^{NLoS} are constants representing the means of excessive loss caused by man-made structures for LoS and NLoS connections, respectively. As the sub-channels used by UAV-BSs have a relatively narrow bandwidth and are adjacent in the frequency domain, this work approximately assumes that f is a constant for all the sub-channels.

Obviously, r_{ij} can be expressed as a function of variables x_j and y_j:

$$r_{ij} = \sqrt{(x_j\delta_d - X_i\delta_d)^2 + (y_j\delta_d - Y_i\delta_d)^2 + h^2}. \tag{3.2}$$

P_{ij}^{LoS} and P_{ij}^{NLoS} are determined by elevation angle of the transmission link between UE i and UAV-BS j, $\theta_{ij}(x_j, y_j)$, which is also a function about variables x_j and y_j [59]:

$$P_{ij}^{\mathrm{LoS}} = \frac{1}{1 + a\exp(-b(\frac{180}{\pi}\theta_{ij}(x_j, y_j) - a))}, \tag{3.3}$$

$$P_{ij}^{\mathrm{NLoS}} = 1 - P_{ij}^{\mathrm{LoS}}, \tag{3.4}$$

$$\theta_{ij}(x_j, y_j) = \arcsin(h/r_{ij}), \tag{3.5}$$

where a and b are constant parameters determined by the transmission environment.

Based on (3.1)-(3.5), this work further expresses g_{ij} with variables x_j and y_j as:

$$\mathrm{g}_{ij}(x_j, y_j) = 10^{0.1 g_{ij}(\mathrm{dB})}. \tag{3.6}$$

To ensure all the users in region R are well served, this work assumes that any UAV-BS $\forall j$ only hovers in R as long as $0 \leq x_j \leq n_x$ and $0 \leq y_j \leq n_y$. Like in the related works [64][67][73], this part builds the system model in a noise-limited way that the interference is assumed to be managed in the ideal condition, each UAV assigns a dedicated channel to the served users, so only SNR is considered. The interference-limited condition will be considered in the future work. According to Shannon theory, the achievable data rate (in bits per second) of ground UE i in set $I_{\mathrm{UE},t}$ is given by:

$$C_i = B \cdot \log_2 \left(1 + \frac{\sum\limits_{\mathrm{UAV}_j \in J_{\mathrm{UAV}}} \delta_{ij} p_{ij} / g_{ij}(x_j, y_j)}{\sigma_n^2} \right), \tag{3.7}$$

where $\sum\limits_{\mathrm{UAV}_j \in J_{\mathrm{UAV}}} \delta_{ij} p_{ij} / g_{ij}(x_j, y_j)$ is the received transmission power level at UE i. The achievable data rate of each ground UE depends not only on transmission power of its target UAV-BS but also on the UAV-BS's location which determines the expected channel pathloss of the UAV-UE transmission link. The UAV-BS deployment with UE association and transmission power control should be combined to reinforce system performance while guaranteeing the UEs' QoS requirements. As reviewed in Section 2.1, the total consumed power of the UAV-BS is combined with the propulsion power and the transmission power, in

which the propulsion power depends on the flying speed. In this experiment, the UAV-BSs are working in the hovering mode (no speed), so the propulsion power is assumed as constant which does not influence the total consumed power, so the optimisation target in this work only includes the transmission power. In detail, the problem is to jointly optimise variables x_j, y_j, δ_{ij}, and p_{ij} ($\mathrm{UE}_i \in I_{\mathrm{UE},t}$, $\mathrm{UAV}_j \in J_{\mathrm{UAV}}$), with the objective of minimizing the UAV-BSs' sum transmission power subject to the minimum data rate requirement of each ground UE. Mathematically, the optimisation problem can be formulated as follows:

(**P1:**)

$$\underset{x_j, y_j, \delta_{ij}, p_{ij}}{\arg\min} \ \{\sum_{ij}(\delta_{ij} \cdot p_{ij})\} \tag{3.8}$$

$$s.t \quad C1 : \delta_{ij} = \{0,1\}, \forall i, j, \tag{3.9}$$

$$C2 : \sum_{j} \delta_{ij} = 1, \ \forall i, \tag{3.10}$$

$$C3 : \sum_{i} \delta_{ij} \leq \Phi, \ \forall j, \tag{3.11}$$

$$C4 : p_{ij} \leq p_{\max}, \forall i, j, \tag{3.12}$$

$$C5 : C_i \geq C, \forall i, \tag{3.13}$$

$$\begin{aligned} C6 : 0 \leq x_j \leq n_x, \\ 0 \leq y_j \leq n_y, \\ x_j \in \mathbb{Z}^+, y_j \in \mathbb{Z}^+. \end{aligned} \tag{3.14}$$

The problem (3.8) is a joint optimisation problem for finding the optimal solutions for UAV-BS deployment (x_j, y_j), UAV-UE association (δ_{ij}), and power consumption (p_{ij}), and the aim is minimizing the overall transmission power. The constraint C1 (3.9) shows that δ_{ij}

is a binary to control the set-up of connections. The constraint C2 (3.10) ensures that any UE i is allowed to connect to only one UAV-BS at a time. Then, the constraint C3 (3.11) allocates the capacity threshold Φ to all UAV-BSs to confirm that the served UEs of the UAV-BS j should be limited. Next, the constraint C4 (3.12) ensures a maximum permitted-transmission power. Then, each UE i owns a connection with a minimum data rate requirement C, which is constrained in C5 (3.13). Last, the constraint C6 (3.14) limits the range of hovering.

3.3 Proposed UDUA Mechanism

Solving the optimisation problem P1 is challenging since it is a complex MINLP problem, which is NP-hard. From (3.8), it can be seen that the locations of UAV-BSs and the associating relationships between the UAV-BSs and the ground users both affect the system power consumption, and the user association can be performed when the UAV-BSs' locations are determined. An intuitive idea to address the optimisation problem is the exhaustive search, which compares the system power consumption values of optimal user associations corresponding to all the possible position combinations of the J UAV-BSs and finds the minimum. However, the exhaustive searching techniques are not practical for on-line computing when the scale of the problem gets large.

In this section, this work decouples the original optimisation problem into the user association sub-problem and the UAV-BS deployment sub-problem. This work transforms the first sub-problem into a bipartite matching problem and then tackle it using the Kuhn-Munkres method. For the second sub-problem, in order to decrease the on-line computing time, this work proposes a KNN based algorithm to obtain the proper UAV-BS deployment strategy related to each new user distribution with the help of optimal UAV-BS deployment strategies for a series of given user distributions, which are calculated off-line.

3.3.1 Decoupling P1

By dividing variables x_j, y_j, δ_{ij} and p_{ij} ($\mathrm{UE}_i \in I_{\mathrm{UE},t}$, $\mathrm{UAV}_j \in J_{\mathrm{UAV}}$) into two groups, the original optimisation problem of P1 can be decoupled into two sub-problems, one is the user

association sub-problem which acquires the optimal matching strategy between the UAV-BSs and the ground users for given UAV-BS positions. The other is the UAV-BS deployment sub-problem, which tries to find the best position combination of the J UAV-BSs making the first sub-problem's solution minimal among all the possible position combinations.

When positions of UAV-BSs are fixed ($x_j = \bar{x}_j, y_j = \bar{y}_j$, $\forall \text{UAV}_j \in J_{\text{UAV}}$), variables δ_{ij} ($\text{UE}_i \in I_{\text{UE},t}$, $\text{UAV}_j \in J_{\text{UAV}}$) will determine how the ground users are associated to the J UAV-BSs. The user association sub-problem can be formulated as follows:

(**P1-1:**)

$$\underset{\delta_{ij}, p_{ij}}{\arg\min} \{\sum_{ij}(\delta_{ij} \cdot p_{ij})\} \tag{3.15}$$

$$s.t \;\; C1 - C5, \tag{3.16}$$

$$\begin{aligned} g_{ij}(x_j, y_j) &= g_{ij}(\bar{x}_j, \bar{y}_j), \\ \forall \text{UAV}_j \in J_{\text{UAV}}, &\forall \text{UE}_i \in I_{\text{UE},t}, \end{aligned} \tag{3.17}$$

where constraints C1-C5 are defined in (3.9)-(3.13). For given ground user distribution $I_{\text{UE},t}$ and position combination of the J UAV-BSs, i.e., $(\bar{x}_1, ..., \bar{x}_J)$ and $(\bar{y}_1, ..., \bar{y}_J)$, the author defines the optimal value of **P1-1** as $f_{I_{\text{UE},t}}(\bar{x}_1, ..., \bar{x}_J, \bar{y}_1, ..., \bar{y}_J)$, where $f_{I_{\text{UE},t}}(x_1, ..., x_J, y_1, ..., y_J)$ can be seen as a function about the variables x_j and y_j ($\text{UAV}_j \in J_{\text{UAV}}$). Thus, the UAV-BS deployment sub-problem is formulated as:

(**P1-2:**)

$$\underset{x_j, y_j}{\arg\min} \; f_{I_{\text{UE},t}}(x_1, ..., x_J, y_1, ..., y_J) \tag{3.18}$$

$$s.t \;\; C6, \tag{3.19}$$

where the constraint C6 is defined by (3.14).

3.3.2 Solution for the User Association Sub-problem

When position of UAV-BS j is given as ($\bar{x}_j$, $\bar{y}_j$), the channel pathloss between UAV-BS j and ground user i has the certain value of $g_{ij}(\bar{x}_j, \bar{y}_j)$ according to (6). If user i is connected

to UAV-BS j, the transmission power p_{ij} should comply with the following inequality by jointly considering (3.7) and (3.13):

$$p_{ij} \geq g_{ij}(\bar{x}_j, \bar{y}_j)\sigma_n^2(2^{\frac{C}{B}} - 1) \tag{3.20}$$

This chapter uses the constant P_{ij} to present the minimal transmission power from the given UAV-BS j to the ground user i. If $P_{ij} > p_{\max}$, user i cannot be associated to UAV-BS j due to the maximum transmission power constraint (3.12). Otherwise, UAV-BS j will set its transmission power p_{ij} as P_{ij} to serve user i with the objective of decreasing the system power consumption.

The user association is to allocate UEs to UAV-BSs, it is an assignment problem. For example, the UAV-BSs are agents who need to perform tasks of associating the UEs. Several tasks can be allocated to one agent, but multiple agents are not allowed to do one task. Indeed, cost exists and varies according to the agent-task assignment. The power consumption is the cost that should be minimised while doing the agent-task assignment (user association). This assignment problem has an easier description by using the bipartite graph. After that, the well-developed KM algorithm [173] becomes available to optimally solve the problem.

The bipartite graph contains two disjoint sets of vertices, and a vertex connects to another vertex in the other group with an edge. This work represents the J UAV-BSs and the I ground users as two groups of vertexes shown in Fig. 3.2(a). For the vertex related to user i and the vertex related to UAV-BS j, they will have a link (edge) with weight P_{ij} as long as $P_{ij} \leq p_{\max}$, and can not connect to each other once $P_{ij} > p_{\max}$. Then, the user association sub-problem P1-1 is equivalent to the assignment problem for a bipartite graph, where the objective is to minimise the sum weight of the assigned links and the following principles should be satisfied:

1) The link between ground user i and UAV-BS j is assigned when and only when user i is served by UAV-BS j ($\delta_{ij} = 1$);

2) In accordance with C2 (3.10) that a ground user must be served by one UAV-BS in set J, the vertex related to any user will have and only have one assigned link to the vertexes related to the UAV-BSs;

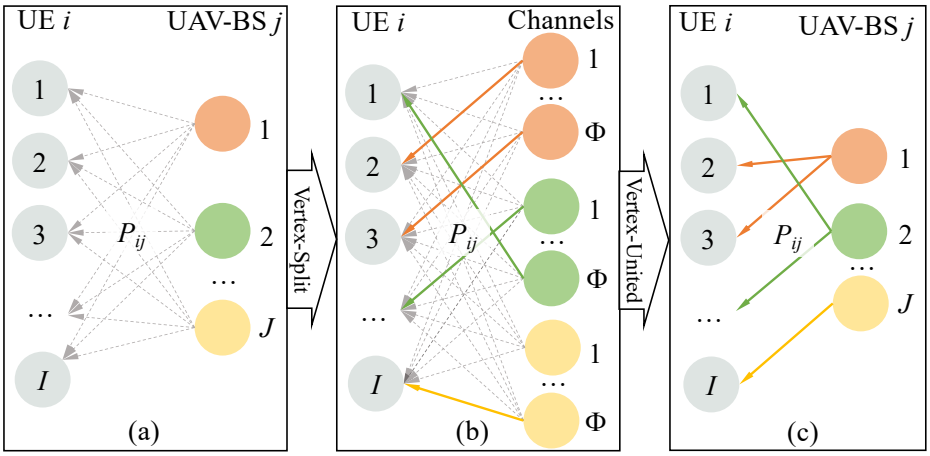


Fig. 3.2 Node-split KM algorithm to allocate UEs to UAV-BSs with the capacity threshold. (a) assignment problem description. (b) assignment problem solution with UE-UAV links assigned. (c) outcome of the algorithm, UEs are associated to the UAV-BSs.

3) In accordance with C3 (3.11) that a UAV-BS will at most serve Φ ground users due to its limited OFDMA sub-channels, not more than Φ assigned links can be connected to the vertex related to any UAV-BS in set J.

It should be noted that the traditional KM algorithm can not directly solve the assignment problem in Fig. 3.2(a) because it is designed to solve one-to-one assignment. In contrast, current problem is a many-to-one problem that many UEs (tasks) can be assigned to one UAV-BS (agent). Therefore, this work splits every vertex related to a UAV-BS in Fig. 3.2(a) into Φ vertexes as shown in Fig. 3.2(b). The links between a split UAV-BS vertex and the user vertexes have the same weight values as those between the original UAV-BS vertex and the user vertexes. Specifically, if user i is not associated to UAV-BS j in Fig. 3.2(a), the author sets links between the user vertex and the split UAV-BS vertexes in Fig. 3.2(b) with a large weight value of w' ($w' >> p_{\max}$) in a formal way. Thus, the assignment problem in Fig. 3.2(a) can further be transformed into the assignment problem in Fig. 3.2(b) with the same objective of minimising the sum weight of the assigned links. Different from Fig. 3.2(a), each split UAV-BS vertex in Fig. 3.2(b) can have one assigned link to the user vertexes at most.

The assignment problem in Fig. 3.2(b) is a typical minimum-weight one-to-one matching problem of a bipartite graph, which can be solved efficiently by the existing Kuhn-Munkres

algorithm [3.2]. It should be noted that the equivalent user association sub-problem will have no feasible solution when there is a w'-weight link in Fig. 3.2(b) being assigned by the Kuhn-Munkres algorithm.

Finally, as illustrated in Fig. 3.2(c), all the ground users possessing a assigned link to the split UAV-BS vertexes related to UAV-BS j will be associated to this UAV-BS. The optimal value of P1-1, $f_{I_{\mathrm{UE},t}}(\bar{x}_1,...,\bar{x}_J,\bar{y}_1,...,\bar{y}_J)$, can also be obtained by adding the weights of assigned links together if it has feasible solutions. Also, when P1-1 does not have feasible solutions for a certain UAV-BS deployment strategy, this work formally records $f_{I_{\mathrm{UE},t}}(x_1,...,x_J,y_1,...,y_J)$ as $I \times w'$.

3.3.3 Solution for the UAV-BS Deployment Sub-problem

Based on the solution of P1-1 for any given UAV-BS deployment strategy, this work uses the exhaustive searching method to test all the possible location combinations of the considered UAV-BSs and choose the best one that achieves minimum $f_{I_{\mathrm{UE},t}}(x_1,...,x_J,y_1,...,y_J)$ value. Nevertheless, this exhaustive searching method is not proper for on-line UDUA problems since the searching space augments exponentially as the UAV-BS number gets large. For $n_y \times n_x$ grids and J UAV-BSs considered, there are $(n_y \times n_x)^J$ possible UAV-BS deployment strategies in summary.

This work solves the UAV-BS deployment sub-problem by imitating the way of thinking used by humans. Inspired by the phenomenon that people tend to handle a new problem utilising the experiences and knowledge from previously solved ones, the author analyses whether the optimal UAV-BS deployment strategies of given ground user distributions can help to provide a proper UAV-BS deployment strategy for any newly considered ground user distribution.

Lemma 1: The author uses $I_{\mathrm{UE},1}$ to represent an arbitrary set of ground users, and use $(x_1,...,x_J,y_1,...,y_J)$ to represent a certain deployment strategy of the J UAV-BSs. For any ground user $u_a \notin I_{\mathrm{UE},1}$, this work uses $I_{\mathrm{UE},2}$ to represent $I_{\mathrm{UE},1} \cup \{u_a\}$. If the UAV-BS deployment strategy $(x_1,...,x_J,y_1,...,y_J)$ makes the user association sub-problems related to both $I_{\mathrm{UE},1}$ and $I_{\mathrm{UE},2}$ have feasible solutions, then for an arbitrary feasible user association

strategy of $I_{\mathrm{UE},1}$, u_a can be connected to a proper UAV-BS with available sub-channels by adjusting the connecting statuses of up to $J-1$ ground users in $I_{\mathrm{UE},1}$.

Proof: See Section 3.7.

This chapter further proves **Lemma 2**.

Lemma 2: For a given set of ground users, $I_{\mathrm{UE},1}$, this work uses $(x_1^{*1},...,x_J^{*1},y_1^{*1},...,y_J^{*1})$ to represent the optimal UAV-BS deployment strategy related to $I_{\mathrm{UE},1}$. Then for an arbitrary set of ground users, $I_{\mathrm{UE},2}$, where m new ground users are added to $I_{\mathrm{UE},1}$, if $(x_1^{*1},...,x_J^{*1},y_1^{*1},...,y_J^{*1})$ makes the user association sub-problem of $I_{\mathrm{UE},2}$ have feasible solutions, the following inequality is obtained:

$$\begin{aligned} &f_{I_{\mathrm{UE},2}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) \leq \\ &f_{I_{\mathrm{UE},1}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) + \\ &m[p_{\max}+(J-1)(p_{\max}-p_{\min})], \end{aligned} \tag{3.21}$$

where $f_{I_{\mathrm{UE},1}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1})$ and $f_{I_{\mathrm{UE},2}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1})$ are the optimal values of the user association sub-problems related to $I_{\mathrm{UE},1}$ and $I_{\mathrm{UE},2}$, respectively, when the UAV-BS deployment strategy is $(x_1^{*1},...,x_J^{*1},y_1^{*1},...,y_J^{*1})$, $p_{\min}$ is the transmission power needed by a UAV-BS to serve a ground user when the channel pathloss between them equals the minimal possible value.

Proof: See Section 3.8.

With **Lemma 1** and **Lemma 2**, **Proposition 1** can be proved.

Proposition 1: The author uses $(x_1^{*1},...,x_J^{*1},y_1^{*1},...,y_J^{*1})$ and $(x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2})$ to represent the optimal UAV-BS deployment strategies for two given sets of ground users, $I_{\mathrm{UE},1}$ and $I_{\mathrm{UE},2}$, respectively. If $I_{\mathrm{UE},2}$ can be generated by adding m ground users into or removing m ground users off $I_{\mathrm{UE},1}$, and $(x_1^{*1},...,x_J^{*1},y_1^{*1},...,y_J^{*1})$ and $(x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2})$ both make the user association sub-problems related to $I_{\mathrm{UE},1}$ or $I_{\mathrm{UE},2}$ have feasible solutions, then the following inequality is achieved:

$$
\begin{aligned}
&f_{I_{\mathrm{UE},2}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) \leq \\
&f_{I_{\mathrm{UE},2}}(x_1^{*2},\ ...,\ x_J^{*2},\ y_1^{*2},\ ...,\ y_J^{*2}) + \\
&mJ(p_{\max} - p_{\min}).
\end{aligned} \tag{3.22}
$$

Proof: See Section 3.9.

From **Proposition 1**, it can be concluded that, under certain conditions, adopting the optimal UAV-BS deployment strategy of a previous ground user set for a new ground user set will introduce limited extra transmission power consumption compared with this new user set's own optimal UAV-BS deployment strategy, if the new user set is achieved by adding some ground users into or removing some ground users off the previous set. Also, the limitation of the increased transmission power consumption is linearly correlated to the user number difference between the two ground user sets.

For a given ground user set, $I_{\mathrm{UE},1}$, when there are ground users moving inside the considered region R, the following **Lemma 3** can be proved.

Lemma 3: For a given set of ground users, $I_{\mathrm{UE},1}$, this work uses $(x_1^{*1},...,x_J^{*1},y_1^{*1},...,y_J^{*1})$ to represent the optimal UAV-BS deployment strategy related to $I_{\mathrm{UE},1}$. If $(x_1^{*1},...,x_J^{*1},y_1^{*1},...,y_J^{*1})$ makes the ground user set $I_{\mathrm{UE},2}$, where n ground users in $I_{\mathrm{UE},1}$ change their position grids, have feasible solutions for the corresponding user association problem, the following relationship is reached:

$$
\begin{aligned}
&f_{I_{\mathrm{UE},2}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) \leq \\
&f_{I_{\mathrm{UE},1}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) + \\
&nJ(p_{\max} - p_{\min}).
\end{aligned} \tag{3.23}
$$

Proof: See Section 3.10.

Based on **Lemma 3**, **Proposition 2** can be proved.

Proposition 2: This work uses $(x_1^{*1},...,x_J^{*1},y_1^{*1},...,y_J^{*1})$ and $(x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2})$ to represent the optimal UAV-BS deployment strategies for two given sets of ground users, $I_{\mathrm{UE},1}$ and $I_{\mathrm{UE},2}$, respectively. $I_{\mathrm{UE},2}$ is acquired by changing the position grids of n ground users in

$I_{\mathrm{UE},1}$. If $(x_1^{*1},...,x_J^{*1},y_1^{*1},...,y_J^{*1})$ and $(x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2})$ both make the user association sub-problems related to $I_{\mathrm{UE},1}$ or $I_{\mathrm{UE},2}$ have feasible solutions, then the following inequality is achieved:

$$\begin{aligned} &f_{I_{\mathrm{UE},2}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) \leq \\ &f_{I_{\mathrm{UE},2}}(x_1^{*2},\ ...,\ x_J^{*2},\ y_1^{*2},\ ...,\ y_J^{*2}) + \\ &2nJ(p_{\max} - p_{\min}). \end{aligned} \tag{3.24}$$

Proof: See Section 3.11.

Similar with **Proposition 1**, **Proposition 2** shows that under certain conditions, adopting the optimal UAV-BS deployment strategy of a previous ground user set for a new ground user set will introduce limited extra transmission power consumption compared with the new user set's own optimal UAV-BS deployment strategy, when the new user set can be achieved from the previous user set by moving some ground users inside the considered region R. Furthermore, the bound of increased transmission power consumption is proportional to the number of users moved.

Proposition 1 and **Proposition 2** imply that even though the optimal UAV-BS deployment strategy of a previous ground user set, $I_{\mathrm{UE},1}$, isn't the best UAV-BS deployment strategy of a new user set, $I_{\mathrm{UE},2}$, adopting this UAV-BS deployment strategy for $I_{\mathrm{UE},2}$ is likely to introduce very little extra power consumption compared with $I_{\mathrm{UE},2}$'s actual optimal UAV-BS deployment strategy if the two ground user sets are similar (m and n in (3.22) or (3.24) are small). From (3.22) and (3.24), it can be seen that each ground user moved inside the considered region seems to have a double effect on this extra power consumption than a user moved in or out.

Based on **Proposition 1** and **Proposition 2**, this work defines the difference degree between two ground user sets and propose a KNN [177] based algorithm to solve the UAV-BS deployment sub-problem as demonstrated in Fig. 3.3.

This work uses an $n_y \times n_x$ matrix D_t to represent the user distribution of a certain ground user set, $I_{\mathrm{UE},t}$. Each element $D_t(k_y,k_x)$ is an integer which records the number of ground

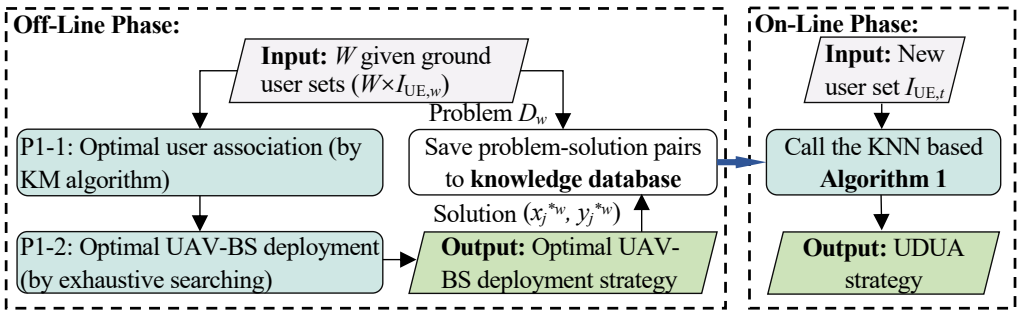


Fig. 3.3 Process description of the proposed algorithm.

users in $I_{\mathrm{UE},t}$ located in grid $(k_y,k_x), k_y = 1,...,n_y, k_x = 1,...,n_x$. For two ground user sets $I_{\mathrm{UE},1}$ and $I_{\mathrm{UE},2}$, this work defines their difference degree as follows.

Definition: For two arbitrary ground user sets $I_{\mathrm{UE},1}$ and $I_{\mathrm{UE},2}$, their difference matrix is obtained, D_{diff}, by operating the matrix subtraction between user distribution matrices related to the two user sets, D_1 and D_2, as shown in Fig. 3.4. The difference degree between $I_{\mathrm{UE},1}$ and $I_{\mathrm{UE},2}$ is defined as:

$$\Gamma_{\mathrm{diff}}(I_{\mathrm{UE},1}, I_{\mathrm{UE},2}) = m + 2n, \tag{3.25}$$

where m and n denote, compared to $I_{\mathrm{UE},1}$, the number of ground users in $I_{\mathrm{UE},2}$ moved in or moved out of the considered region and the number of ground users in $I_{\mathrm{UE},2}$ moved inside the considered region, respectively. m and n can be calculated through D_{diff}:

$$m = \left| \sum_{\substack{k_y=1,...,n_y,\\ k_x=1,...,n_x}} D_{\mathrm{diff}}(k_y,k_x) \right|, \tag{3.26}$$

$$n = \min\left\{ \sum_{\substack{k_y=1,...,n_y,\\ k_x=1,...,n_x,\\ D_{\mathrm{diff}}(k_y,k_x)>0}} |D_{\mathrm{diff}}(k_y,k_x)|, \sum_{\substack{k_y=1,...,n_y,\\ k_x=1,...,n_x,\\ D_{\mathrm{diff}}(k_y,k_x)<0}} |D_{\mathrm{diff}}(k_y,k_x)| \right\}. \tag{3.27}$$

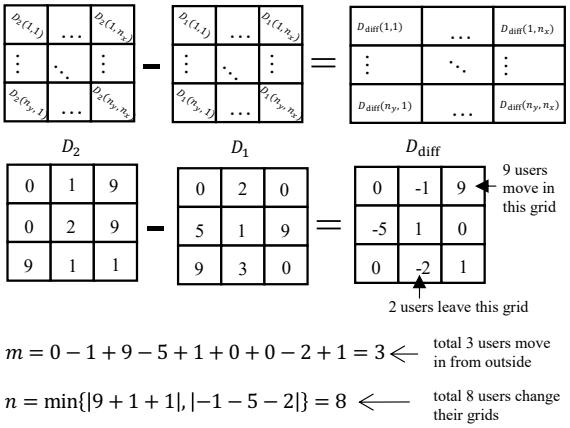


Fig. 3.4 Illustration of calculating the difference matrix D_{diff} and the parameters of difference degree, m and n.

The solution to the second sub-problem is demonstrated in Fig. 3.3. At the off-line phase, the proposed algorithm stores the optimal UAV-BS deployment strategies of W given ground user sets to construct a knowledge database in advance. This knowledge database can be viewed as an analogy to a human's experience, which this work uses to handle the new problems. For each ground user set $I_{\text{UE},w}$ in the knowledge database, this work uses matrix D_w to record the user distribution and get its optimal UAV-BS deployment strategy $(x_1^{*w}, ..., x_J^{*w}, y_1^{*w}, ..., y_J^{*w})$ by exhaustively comparing all the $(n_y \times n_x)^J$ possible UAV-BS deployment strategies. Notably, although preparing the knowledge database is relatively computing-resource consuming, this task can be accomplished before the UAV radio access network is set, and thus it will not influence the running time of each on-line UDUA problem. For each newly considered UDUA problem with ground user set $I_{\text{UE},t}$, the proposed UAV-BS deployment algorithm will first calculate $I_{\text{UE},t}$'s difference degree to each ground user set in the knowledge database. Then, the proposed algorithm will compare the optimal UAV-BS deployment strategies related to the k ground user sets in the knowledge database, which have the smallest difference degrees with $I_{\text{UE},t}$, and select the feasible one achieving the minimum transmission power consumption. The pseudo code of the proposed UAV-BS deployment algorithm's on-line phase is given in **Algorithm 1**.

Algorithm 1: KNN based UDUA algorithms in on-line phase

input : a new ground-user set $I_{\mathrm{UE},t}$, the knowledge database: given ground-user sets $W \times I_{\mathrm{UE},w}$ and associated UAV-BS deployment strategies $W \times (x_1^{*w}, ..., x_J^{*w}, y_1^{*w}, ..., y_J^{*w})$.

output : UAV-BS deployment $(x_1^{*k}, ..., x_J^{*k}, y_1^{*k}, ..., y_J^{*k})$ and user association strategy of $I_{\mathrm{UE},t}$.

generate the distribution matrix D_t according to $I_{\mathrm{UE},t}$;
for each user set matrix D_w in W **do**
    calculate m and n with D_t and D_w based on (3.26) and (3.27);
    generate the difference degree Γ_{diff} (3.25);
end
retrieve top k sets from the knowledge database $\{D_w^1, ..., D_w^k\}$ with minimum Γ_{diff};
retrieve k associated UAV-BS deployments $\{(x_1^1, ..., x_J^1, y_1^1, ..., y_J^1), ..., (x_1^k, ..., x_J^k, y_1^k, ..., y_J^k)\}$ from database;
for each k value **do**
    run the **Kuhn-Munkres** on $\{D_t, (x_1^k, ..., x_J^k, y_1^k, ..., y_J^k)\}$;
    if Kuhn-Munkres has a feasible solution **then**
        record the values of power consumption $f_{D_t}(x_1^k, ..., x_J^k, y_1^k, ..., y_J^k)$;
    else
        output : no feasible solution
    end
end
get the UAV-BS solution $(x_1^{*k}, ..., x_J^{*k}, y_1^{*k}, ..., y_J^{*k})$ with minimum power consumption;
output : $(x_1^{*k}, ..., x_J^{*k}, y_1^{*k}, ..., y_J^{*k})$

3.3.4 Computational Complexity of An On-line UDUA Problem

For an on-line UDUA problem with ground user set $I_{\mathrm{UE},t}$, constructing its user distribution matrix D_t has the complexity of $O(I)$, where I is the number of ground users; calculating the difference matrices and difference degrees between $I_{\mathrm{UE},t}$ and the W given ground user sets both have the complexity of $O(Wn_yn_x)$, where $n_y \times n_x$ are the total grid number of the considered region; finding the k ground user sets in the knowledge database possessing the smallest difference degrees with $I_{\mathrm{UE},t}$ has the complexity of $O(W)$. In line 9 of **Algorithm 1**, solving the user-association sub-problem for $I_{\mathrm{UE},t}$ with the UAV-BS deployment strategy related to each of the k selected ground user sets using the Kuhn-Munkres algorithm has the complexity of $O(I^4)$ [173]. Finally, choosing the feasible UAV-BS deployment strategy, which achieves the minimum transmission power consumption for $I_{\mathrm{UE},t}$ among the k candidate

ones has the complexity of $O(k)$. Thus, the overall computational complexity of an on-line UDUA problem is bounded by $O(Wn_yn_x + I^4k)$.

Notably, for a candidate UAV-BS deployment strategy and a considered ground user set $I_{\mathrm{UE},t}$, the channel power gain between each UAV-BS and each ground user can be acquired directly by reading a table that provides all the possible channel power gain values between a UAV-BS and a ground user when they are located in the rasterised region R. As a result, this work does not take the complexity of calculating these channel power gains into consideration in the complexity analysis.

3.4 Experimental Results

This work evaluates the performance of the proposed UDUA mechanism through extensive experiments. In this section, the experimental settings are first described. Then, this work tests how the two key hyper-parameters, i.e., the scale of the knowledge database, W, and the number of candidate UAV-BS deployment strategies, k, will influence the proposed mechanism's performance. This work also compares the UDUA mechanism with some baseline UDUA approaches under various network scenarios. Finally, experimental results about storage resources needed as well as the off-line and on-line computational time of the proposed mechanism with different hyper-parameter values will be provided. For reader's convenience, this work summarises the experimental parameters in Table 3.1

3.4.1 Experimental Parameters

In the experiments, the author considers a square stadium-size region that is 90 m $\times$ 90 m (a typical soccer field is 125m $\times$ 85m), and it is evenly divided into 9×9 grids ($n_y = n_x = 9$). Some urban squares are also designed in this size, such as London Trafalgar Square (110 m by 110 m), and Glasgow George Square (125 m by 90 m), where many users may distribute in this region requiring communication supports. The carrier frequency is 2 GHz as suggested by 3GPP release 15 [178] and a highly cited paper [59]. This frequency is selected because the

propagation performance satisfies both the requirements of the air-to-ground communication and the bandwidth allocation.

The sub-channel bandwidth is a constant parameter when one user is associated to one sub-channel. In the literature, their values varied from 0.015 MHz [179] to 0.18 MHz [88]. 3GPP [178] suggests the total bandwidth of a UAV-BS is 10 MHz, so this experiment sets the sub-channel bandwidth as 0.1 MHz, then the UAV-BS can serve up to 100 people. According to the paper [180], different services' minimum QoS requirements were Web browsing (minimum 0.03 Mb/s), multimedia on Web (minimum 0.028 Mb/s), video conferencing (minimum 0.064 Mb/s). Therefore, the minimum data rate in this experiment is set 10 times more than Web multimedia's requirements (0.28 MB/s). As the UAV-BSs are working in the hovering model, their hovering height needs to be fixed. According to 3GPP study of aerial vehicles [178], the height of urban UAV-BSs (micro-cell) is suggested to be chosen from $\{50, 100, 200, 300\}$ meters. 200 m is a commonly selected because the UAVs do not need to frequently ascend or descend to avoid buildings [59][70]. $p_{\max}$ influences the comparison of fail ratio among different algorithms, so it is manually selected according to the criteria that the better algorithm can achieve $p_{ij} \leq p_{\max}$ but the worse one will fail. For this purpose, the maximum permitted power for each user is set as 0.3877 mW for a better comparison.

According to the findings in [181] that people are usually distributed non-uniformly and tend to gather together in some hot spots due to certain social events, this work uses a log-normal distribution with parameters μ and σ to fit the number of ground users in each grid in the region R. μ and σ jointly determine the density of ground users in R, and σ denotes how non-uniformly the ground users are distributed. The author varies the value of μ in set $\{-1, -0.5, -0.4, -0.3, -0.2\}$ and vary the value of σ in set $\{0.2, 0.3, 0.4, 0.5, 1\}$. Though the 25 value combinations of μ and σ can not depict all the possible ground user distributions in the real world, they comprise lots of general radio access network scenarios where the density and the non-uniformity of ground users range widely.

For each of the 25 value combinations of μ and σ, this work randomly generates $W/25$ ground user sets to construct the knowledge database and use the exhaustive searching method to obtain their optimal UAV-BS deployment strategies. This work also randomly generates

Parameters	Description	Values
B	Sub-channel bandwidth	0.1 MHz [178]
C	Data-rate requirement	0.28 Mb/s [180]
c	Speed of Light	299792458 m/s
f	Frequency	2 GHz [178]
h	Height of UAV-BSs	200 m [70]
N_{Test}	Size of test user sets	10
$p_{\max}$	Maximum permitted power	0.3877 mW
W	Size of database	500
J	Amount of UAV-BSs	2
μ	Log-normal parameter	{-1, -0.5, -0.4, -0.3, -0.2} [181]
σ	Log-normal parameter	{0.2, 0.3, 0.4, 0.5, 1} [181]
σ_n^2	Noise power	-150 dBm [178]
(a,b)	Pathloss model parameters (urban)	(9.6117, 0.2782) [59]
(μ^{LoS},μ^{NLoS})	Mean of additive pathloss (urban)	(1,20) [59]

Table 3.1 Parameter values in experiment.

N_{Test} testing ground user sets related to every value combination of μ and σ to evaluate the proposed UDUA mechanism's performance. In order to demonstrate the efficiency of the UDUA mechanism, this work compares it with four kinds of algorithms. The first one is a combination of Exhaustive UAV-BS Deployment & Kuhn-Munkres User Association (ESUD-KMUA), this offers ground-truth results by searching all possible solutions. The second one is Simulated-Annealing UAV-BS Deployment & Greedy User Association (SAUD-GUA). Simulated Annealing is a heuristic method which sacrifices limited performance for reducing the time complexity [182]. And the Greedy method solves user association by connecting the users needing the lower transmission power first [75]. SAUD-GUA is a common mode in literature with combining the Greedy method with a feasible UAV-BS deployment strategy, so this work uses it to represent the performance in the related works. To be fair, the author also combines Simulated-Annealing UAV-BS Deployment & Kuhn-Munkres User Association (SAUD-KMUA) for the comparison. This will compare the widely used Greedy method with the proposed KMUA method. The final method uses the random method for the UAV-BS deployment, which randomly generates locations of UAV-BSs. The users are associated through the Greedy method, so this is named Random UAV-BS Deployment & Greedy User Association (RUD-GUA). It is no doubt that RUD-GUA has the lowest time-complexity but the worst performance.

If an approach does not find a feasible UDUA solution for a specific testing ground user set, this work will record one failure to this approach. The failure rate of a UDUA approach is calculated by the following equation:

$$\text{Failure rate} = \frac{N_{\text{Fail}}}{N_{\text{Test,Sum}}}, \tag{3.28}$$

where N_{Fail} is the failure number of the UDUA approach, $N_{\text{Test,Sum}}$ is the number of this approach's testing ground user sets.

Our experiments are executed on MATLAB 2018b by a MacBook Pro with a 1.4 GHz Intel Core i5 processor and 16 GB 2133 MHz LPDDR3 RAM. The values of major experimental parameters are summarised in **Table 1** according to 3GPP-LTE based radio access network systems [178].

3.4.2 Influence of Key Hyper-parameters on the Proposed UDUA Mechanism

As described in **Algorithm 1**, the proposed UDUA mechanism compares the optimal UAV-BS deployment strategies related to the k most similar ground user sets in the knowledge database with size W. As a result, both W and k will have an influence on the proposed mechanism's performance.

Fig. 3.5 presents the difference between the proposed UDUA mechanism and the ground-truth ESUD-KMUA approach in average system transmission power consumption (ΔP) over the total $25 \times N_{\text{Test}}$ testing ground user sets under various values of W and k. $\Delta P = 0$ means the optimum solution. From Fig. 3.5, it can be seen that as k rises from 1 to 10 and W varies from 10 to 500, the performance gap between the proposed mechanism and the ground-truth approach, which can achieve the theoretical optimal solutions for UDUA problems, decreases transparently from around 0.8×10^{-4} watts to lower than 0.2×10^{-4} watts. Moreover, for a certain value of W or k, increasing the value of the other hyper-parameter monotonously improves the proposed mechanism's performance in terms of system transmission power consumption. These observations can be explained as when the proposed

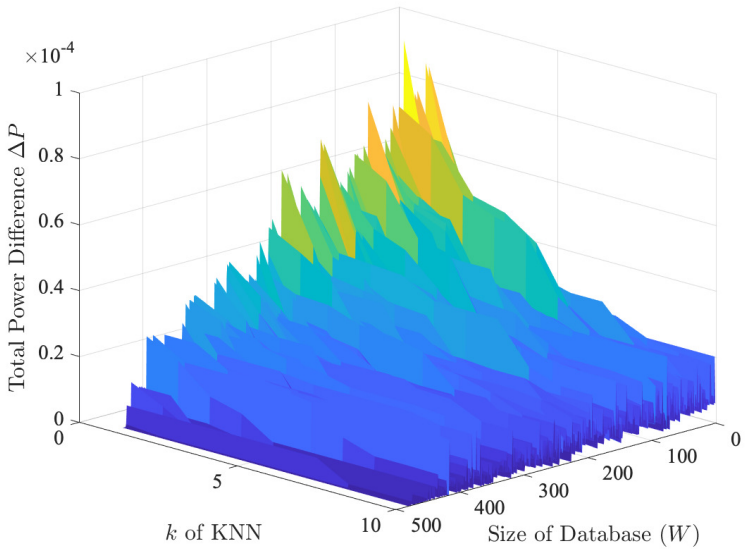


Fig. 3.5 Estimated performance of the proposed UDUA with two key hyper-parameters W and k.

UDUA mechanism possesses a larger knowledge database or considers more candidate UAV-BS deployment strategies for a new problem, it will have a higher probability to find the similar previous ground user sets and more chances to obtain the proper UDUA solution whose result approaches the optimal value according to **Proposition 1** and **Proposition 2**.

An interesting phenomenon in Fig. 3.5 is that when W exceeds 250 and k exceeds 6, further augments of W and k will lead to little performance improvement. This is a meaningful conclusion. It not only confirms the practicability of the proposed UDUA mechanism but provides guidance to the hyper-parameter selection as well.

3.4.3 System Transmission Power Consumption of the Proposed Mechanism and the Baseline Approaches

This subsection compares the system transmission power consumption achieved by the proposed UDUA mechanism and the baseline approaches. For the proposed mechanism, this work sets the values of W and k as 250 and 6, respectively, to balance the performance and computational complexity. For the SAUD approach, this work chooses the system

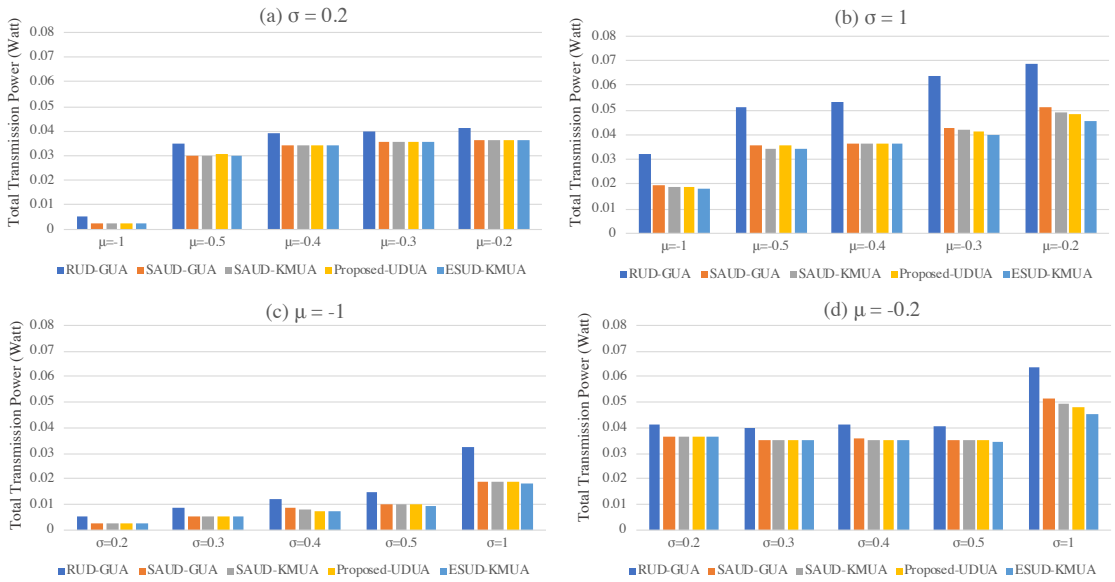


Fig. 3.6 Total transmission power comparison among RUD-GUA, SAUD-GUA, SAUD-KMUA, the proposed UDUA, and ESUD-KMUA.

transmission power consumption as the value of its evaluation function and set the annealing rate λ as 0.95. This work evaluates the system transmission power consumption and failure rates of the four considered approaches under different value combinations of μ and σ. Each result is averaged over N_{Test} testing ground user sets related to a specific network scenario with certain μ and σ.

Fig. 3.6 (a) and (b) show the system transmission power consumption achieved by the five approaches under different values of μ when $\sigma = 0.2$ or $\sigma = 1$. From Fig. 3.6, it shows that the system transmission power consumption of all the approaches increases as μ gets large. This is because for a larger μ, the considered region tends to have a higher ground user density, and thus the UAV-BSs will need more transmission power to serve the ground users. The RUD-GUA approach causes a very distinct rise in power consumption compared to the other approaches. This phenomenon reflects the importance of the UDUA problem addressed in this work since the UAVs might run out of energy quickly if their locations and associated ground users are not assigned properly. In terms of power consumption, the SAUD-GUA, the SAUD-KMUA, and the UDUA mechanism can achieve performance very close to the theoretically optimal value obtained by the ESUD-KMUA. The UDUA mechanism has the

smallest performance gap with the ESUD-KMUA under the most experimental scenarios. These numerical results comply with the fact that the proposed mechanism can find the optimal solution of the user association sub-problem to minimise the system transmission power consumption for any certain UAV-BS deployment strategy, and the proved propositions that adopting the optimal UAV-BS deployment strategy of a previous ground user set for any new user set will get the near optimal system performance if the two user sets are similar.

Fig. 3.6 (c) and (d) plot the system transmission power consumption versus the value of σ with $\mu = -1$ and $\mu = -0.2$. The power consumption achieved by the five approaches ascends as σ augments. This can be explained as, besides influencing the non-uniformity of ground user distribution, the increase of σ will also raise the user density. The performance of the RUD-GUA degrades sharply when σ gets large. This is because ground users tend to be distributed more non-uniformly in the region R for a larger σ, and the positions of UAV-BSs will have a more important effect on the system power consumption. Results in Fig. 3.6 indicate that the UDUA mechanism outperforms the SAUD-GUA approach and the SAUD-KMUA approach. This can also be owed to the optimal solution of the user association sub-problem and the UAV-BS deployment strategy selection based on similar UDUA problems.

3.4.4 Failure Rates of the Proposed Mechanism and the Baseline Approaches

Fig. 3.7 (a) and (b) demonstrate the failure rates of the proposed UDUA mechanism and the baseline approaches under various μ values with $\sigma = 0.2$ and $\sigma = 1$.The approaches' failure rates increase as μ grows. These results are consistent with the intuition that a high ground user density will reduce the probability of the fixed number of UAV-BSs to serve all the users successfully and thus lead to a large failure rate. The RUD-GUA's and the SAUD-GUA's failure rates ascend evidently when μ rises from -1 to -0.2. This is because the greedy algorithm based user association method in these two approaches can only find the local optimal user association strategies for the UAV-BSs. In RUD-GUA and SAUD-GUA,

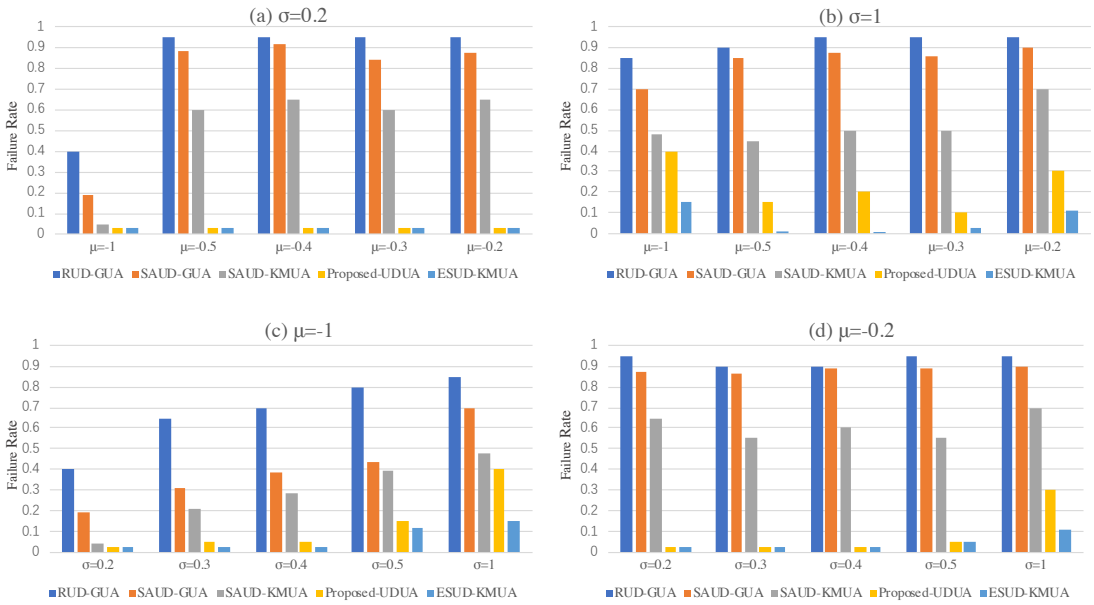


Fig. 3.7 Failure rates comparison among RUD-GUA, SAUD-GUA, SAUD-KMUA, the proposed UDUA, and ESUD-KMUA. The vertical axes indicate the rates varying from 0.03 to 1.

ground users may fail to connect to the proper UAV-BSs since these UAV-BSs are occupied by ground users with smaller pathloss. Thanks to the ability to obtain the optimal solution of the user association sub-problem, the SAUD-KMUA, the ESUD-KMUA, and the proposed mechanism hold very low failure rates even with a large user density. When $\mu = -0.2$, the UDUA mechanism can reduce the failure rates by about 78.5% and 72%, respectively, if it is compared with the RUD-GUA and the SAUD-GUA.

Fig. 3.7 (c) and (d) compare the five approaches' failure rates under different values of σ with $\mu = -1$ and $\mu = -0.2$. Similar to the results in Fig. 8, the RUD-GUA and SAUD-GUA always have much higher failure rates than the other three approaches, and their failure rates increase rapidly as the non-uniformity and density of user distribution go up. The proposed mechanism achieves very low failure rates under all the scenarios. When $\sigma = 1$ and $\mu = -0.2$, though the system sees a relatively high user density and non-uniformity of user distribution, the proposed mechanism only introduces a 19% increase in failure rate than the ESUD-KMUA approach, which can obtain the theoretically optimal system performance.

3.4.5 Analyses for Running Time and Storage Space Needed

This work also concerns about the running time and storage space needed for the proposed UDUA mechanism. Table II lists the average running time (ART) for on-line UDUA problems of the proposed mechanism, and the baseline approaches under different network scenarios. Specifically, this work tests the proposed mechanism's ART with various selections of hyper-parameters. From Table 3.2, it can be found that the RUD-GUA makes the fast decision, which only takes approximately 0.001 s since this approach always chooses a random UAV-BS deployment strategy directly and allocates the ground users to UAV-BSs with a low complexity greedy algorithm. ART of the SAUD-KMUA, the ESUD-KMUA, and the proposed mechanism increases when μ and σ gets large. This is because the computational complexity of the proposed bipartite matching theory-based solution for the user association sub-problem is positively correlative to the user amount in the considered region. For large values of W and k, the proposed mechanism needs more running time to search the knowledge database and compare the candidate UAV-BS deployment strategies. However, the proposed mechanism has a much shorter ART than the SAUD-GUA, the SAUD-KMUA, and the ESUD-GUA approaches. This can be owed to the machine learning technology adopted in the proposed mechanism that acquires the proper UAV-BS deployment strategy for each on-line UDUA problem with the help of accumulated experiences in well-solved problems. When $W = 250$ and $k = 6$, the proposed mechanism can reduce the ART by 99.747%, 99.947%, and 99.978%, respectively, if it is compared with the SAUD-GUA, the SAUD-KMUA, and the ESUD-KMUA.

For each given UDUA problem in the knowledge database, the off-line phase of the proposed mechanism uses the ESUD-KMUA approach to find its optimal UAV-BS deployment strategy and then records this UAV-BS deployment strategy as well as the related user distribution matrix. Table 3.3 demonstrates the off-line preparation time and storage space needed by the proposed mechanism with different scales of the knowledge database. It indicates that the off-line preparing time and storage space needed are proportional to the value of W. Even for a very large W ($W = 500$), the storage space of the proposed mechanism is quite small (less than 110 KB), and the off-line preparing time needed is acceptable (about 63848

	σ=0.2 μ=-1	σ=0.2 μ=-0.5	σ=0.2 μ=-0.2	σ=0.5 μ=-1	σ=0.5 μ=-0.5	σ=0.5 μ=-0.2	σ=1 μ=-1	σ=1 μ=-0.5	σ=1 μ=-0.2
UDUA-W50-k2	0.003211848	0.006429056	0.006714706	0.004350051	0.006319608	0.009314316	0.006537291	0.014682509	0.025597527
UDUA-W50-k6	0.008663324	0.01808706	0.018885926	0.012119843	0.017878434	0.026625138	0.020623513	0.041907311	0.076537955
UDUA-W50-k10	0.014162955	0.030063447	0.03029757	0.019775466	0.029408359	0.044636976	0.032717283	0.072679712	0.134733839
UDUA-W250-k2	0.003640654	0.006705876	0.007401147	0.004715499	0.006824409	0.009574027	0.007271657	0.014935287	0.02753117
UDUA-W250-k6	0.009172347	0.018761028	0.019195799	0.012210318	0.018813461	0.02739446	0.021023584	0.045029816	0.081580183
UDUA-W250-k10	0.014750536	0.030722237	0.031436288	0.020638099	0.030315332	0.045102449	0.034047478	0.074689281	0.137776225
UDUA-W450-k2	0.003952413	0.007004495	0.007879287	0.005043778	0.007354222	0.009918086	0.007718273	0.017382347	0.02929923
UDUA-W450-k6	0.009472272	0.01883904	0.019940736	0.012613304	0.019134735	0.027855519	0.02167515	0.045345232	0.077536795
UDUA-W450-k10	0.01502078	0.031230381	0.030512363	0.020661893	0.031901383	0.045198141	0.036161101	0.080373927	0.139565007
RUD-GUA	0.000156578	0.001174373	0.00140184	0.000432719	0.000953513	0.001325957	0.000751911	0.001318803	0.001767276
SAUD-GUA	1.780808131	9.376126432	11.48727022	4.644118532	10.22489695	14.05247276	10.0809072	15.50979606	22.81528959
SAUD-KMUA	10.89550716	34.44798555	61.62637463	11.73156727	39.62927126	69.93165012	24.4378083	82.07527361	139.8992538
ESUD-KMUA	22.3	82.9	176.5	33.6	131.2	183.5	62.6	183.6	286.7

Table 3.2 Average running time for on-line UDUA problems. The series of UDUA-W-k is the proposed algorithms with different W and k. For example, UDUA-W450-k10 represents the proposed UDUA algorithm with $W = 450$ and $k = 10$.

W	Storage (kB)	Off-line Time (s)
50	11	5946.27
100	22	9043.74
150	33	24347.96
200	44	31542.27
250	55	33776.64
300	66	40534.12
350	77	45792.39
400	88	58189.33
450	99	59654.86
500	110	63848.02

Table 3.3 The off-line preparing time and storage space of the proposed UDUA algorithm.

s). Moreover, as analysed before, although preparing the knowledge database is relatively computing-resource consuming, this task will be accomplished before the UAV radio access network is set.

3.5 Conclusion

To decrease the on-line computing time, this work has made an early attempt to introduce the machine learning technique to solve joint UDUA problems. With the objective of minimising the system transmission power, this work formulated the joint UDUA problem as an MINLP problem, decoupled it into the user association sub-problem and the UAV-BS deployment sub-problem, and proposed a centralised UDUA mechanism to solve them. For the UAV-BS

deployment sub-problem, the author theoretically proved that adopting the best UAV-BS deployment strategy of a previous user distribution for each new user distribution would introduce a slight transmission power increase compared with the new user distribution's best strategy if the two user distributions are similar. This work further defined the similarity level between two user distributions and proposed a KNN based algorithm to calculate the proper UAV-BS deployment strategy for a new user distribution with the help of accumulated experiences in well-solved problems. Numerical results showed that the proposed mechanism could achieve the system performance close to the theoretically optimal values obtained by the ESUD-KMUA approach in terms of transmission power consumption and failure rate for UDUA problems with enormously reduced computational time compared with existing UDUA approaches. Even for a large scale of the knowledge database and a large amount of candidate UAV-BS deployment strategies, the proposed mechanism could reduce the average running time by 99.747%, 99.947%, and 99.978%, respectively, in comparison with the SAUD-GUA, the SAUD-KMUA, and the ESUD-KMUA. Therefore, this work concludes that the proposed mechanism is promising to work in emergent scenarios with bursty traffic demands.

For future work, the author will test the proposed mechanism with ground user distributions generated in real networks. Furthermore, how to use the limited number of given ground user distributions to construct an efficient knowledge database for the most real UDUA problems will be another interesting research direction.

3.6 Proof of Lemma 1

When the UAV-BS deployment strategy $(x_1,...,x_J,y_1,...,y_J)$ makes the user association sub-problems related to $I_{\mathrm{UE},1}$ and $I_{\mathrm{UE},2} = I_{\mathrm{UE},1} \cup \{u_a\}$ have feasible solutions, there are UAV-BSs in set J_{UAV}, whose transmission power to ground user u_a will not be larger than $p_{\max}$ if u_a is matched to one of them, and this work uses set $S_1 \subseteq J_{\mathrm{UAV}}$ to record these UAV-BSs. For an arbitrary feasible user association strategy of $I_{\mathrm{UE},1}$, each ground user in $I_{\mathrm{UE},1}$ will be connected to one UAV-BS and the transmission power of the UAV-BS to serve this user will

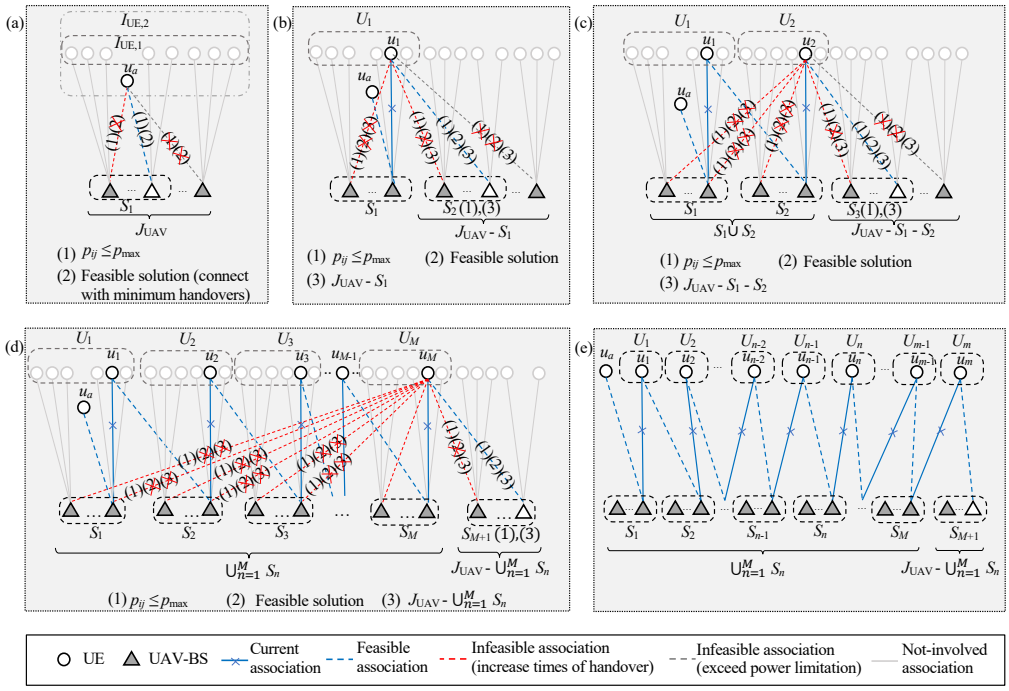


Fig. 3.8 Illustrations of Lemma 1.

not exceed $p_{\max}$. As demonstrated in Fig. 3.8, this work will discuss **Lemma 1** under the following possible conditions:

1. As shown in Fig. 3.8 (a), if there is at least one UAV-BS in set S_1 possessing available sub-channels in this feasible user association strategy of $I_{\mathrm{UE},1}$, then u_a can be associated to this UAV-BS and the connecting status of no ground user $I_{\mathrm{UE},1}$ will need to be changed. Under this condition, the conclusion of **Lemma 1** is achieved;

2. Otherwise, if all of the UAV-BSs in set S_1 are fully occupied by ground users in $I_{\mathrm{UE},1}$, u_a cannot directly be connected to a proper UAV-BS in J_{UAV} with spare sub-channel. This work constructs the set $U_1 \subseteq I_{\mathrm{UE},1}$ to represent the ground users that are associated to the UAV-BSs in S_1 in the considered feasible user association strategy of $I_{\mathrm{UE},1}$. Since $(x_1,...,x_J,y_1,...,y_J)$ makes the user association sub-problem related to $I_{\mathrm{UE},2} = I_{\mathrm{UE},1} \cup \{u_a\}$ have feasible solutions, there is at least one ground user $u_1 \in U_1$ being connected to a UAV-BS in set $J_{\mathrm{UAV}} - S_1$ in one feasible user association strategy of $I_{\mathrm{UE},2}$ when u_a is added to one UAV-BS in S_1. This work uses the set S_2 to represent

the UAV-BSs, whose transmission power to ground user u_1 does not exceed $p_{\max}$, in set $J_{\text{UAV}} - S_1$. Obviously, S_2 is not empty and $S_2 \cap S_1 = \phi$. As demonstrated in Fig. 3.8 (b), if there is at least one UAV-BS in set S_2 possessing available sub-channels in this feasible user association strategy of $I_{\text{UE},1}$, then ground user u_1 can be switched to this UAV-BS and u_a can be associated to the UAV-BS, which previously serves u_1 in S_1. Under this condition, the connecting status of one ground user in $I_{\text{UE},1}$ is adjusted and the conclusion of **Lemma 1** is achieved as the UAV-BS number must be larger than two to construct the sets S_1 and S_2;

3. Otherwise, if all of the UAV-BSs in set S_2 are also fully occupied by ground users in $I_{\text{UE},1}$, this work constructs the set $U_2 \subseteq I_{\text{UE},1}$ to represent the ground users that are associated to the UAV-BSs in S_2 in the considered feasible user association strategy of $I_{\text{UE},1}$. Because in a feasible user association strategy of $I_{\text{UE},2}$, u_a must be matched with a UAV-BS in $S_1 \cup S_2$ (in S_1, specifically), there is at least one ground user $u_2 \in U_1 \cup U_2$ being connected to a UAV-BS in set $J_{\text{UAV}} - S_1 - S_2$ in this user association strategy of $I_{\text{UE},1}$. Obviously, there is $u_2 \neq u_1$. This work uses the set S_3 to represent the UAV-BSs, whose transmission power to ground user u_2 does not exceed $p_{\max}$, in set $J_{\text{UAV}} - S_1 - S_2$. Also, S_3 is not empty and $S_3 \cap (S_1 \cup S_2) = \phi$. This work sets $M = 3$. As illustrated in Fig. 3.8 (c), if there is at least one UAV-BS in set S_3 possessing available sub-channels in this feasible user association strategy of $I_{\text{UE},1}$, the condition changes to 5);

4. Otherwise, if all of the UAV-BSs in set S_3 are fully occupied by ground users in $I_{\text{UE},1}$, then the set U_M is , find the ground user $u_M \in U_1 \cup U_2 ... \cup U_M$ that can be served by a UAV-BS in set $J_{\text{UAV}} - S_1 - S_2 - ... - S_M$ ($u_M \neq ... \neq u_2 \neq u_1$), construct the set S_{M+1}, and judge whether there are UAV-BSs in S_{M+1} possessing available sub-channels in this feasible user association strategy of $I_{\text{UE},1}$, using the similar process in 3). If there is at least one UAV-BS in set S_{M+1} possessing available sub-channels, the condition changes to 5). Otherwise, $M = M + 1$ which repeats the above process as depicted in Fig. 3.8 (d) until there is at least one UAV-BS in set S_{M+1} possessing available

sub-channels. Since S_1, S_2, ..., S_{M+1} are not empty and at least one UAV-BS in J_{UAV} possessing available sub-channels in this feasible user association strategy of $I_{\text{UE},1}$ ($I_{\text{UE},2}$ will not have feasible user association strategies otherwise), the S_{M+1} is under limited repeats and have $M+1 \leq J$;

5. As shown in Fig. 3.8 (e), for certain value of M, there is a ground user in set $U_1 \cup U_2 ... \cup U_M$ can be switched to a UAV-BS in S_{M+1} possessing available sub-channels. This work supposes this ground user belongs to set U_m and denotes it as u_m. Obviously, there is $m \leq M \leq J-1$. Then, u_a can be associated to a UAV-BS in S_1 by switching u_m to the UAV-BS in S_{M+1} possessing available sub-channels, switching u_{m-1} to the UAV-BS in S_m that previously serves u_m, ..., switching u_1 to the UAV-BS in S_2 that previously serves u_2, and adding u_a to the UAV-BS in S_1 that previously serves u_1. Under this condition, the connecting status of m ground users in $I_{\text{UE},1}$ is adjusted and the conclusion of **Lemma 1** is achieved as $m \leq J-1$.

Thus, the conclusion of **Lemma 1** can be achieved under all the conditions.

3.7 Proof of Lemma 2

For ground user set $I_{\text{UE},1}$ and its optimal UAV-BS deployment strategy $(x_1^{*1}, ..., x_J^{*1}, y_1^{*1}, ..., y_J^{*1})$, this work uses $\Delta^{*1} = (\delta_{ij}^{*1}, \text{UE}_i \in I_{\text{UE},1}, \text{UAV}_j \in J_{\text{UAV}})$ to represent the optimal solution of the related user association sub-problem. Obviously, Δ^{*1} is a feasible user association strategy of $I_{\text{UE},1}$.

Without loss of generality, this work denotes the m new ground users in $I_{\text{UE},2}$ as $u_{\text{new},1}$, $u_{\text{new},2}$, ..., and $u_{\text{new},m}$. Since $(x_1^{*1}, ..., x_J^{*1}, y_1^{*1}, ..., y_J^{*1})$ makes the user association sub-problem related to $I_{\text{UE},2}$ have feasible solutions, this UAV-BS deployment strategy will also make the user association sub-problems related to $I_{\text{UE},1} \cup \{u_{\text{new},1}\}$, $I_{\text{UE},1} \cup \{u_{\text{new},1}, u_{\text{new},2}\}$, and $I_{\text{UE},1} \cup \{u_{\text{new},1}, u_{\text{new},2}, ..., u_{\text{new},(m-1)}\}$ have feasible solutions. According to **Lemma 1**, $u_{\text{new},1}$ can be connected to a proper UAV-BS and find a feasible user association strategy of $I_{\text{UE},1} \cup \{u_{\text{new},1}\}$, $\Delta_{I_{\text{UE},1} \cup \{u_{\text{new},1}\}}$, from Δ^{*1} by adjusting the connecting statuses of up to $J-1$

previous ground users. Because the extra transmission power consumption caused by serving $u_{\text{new},1}$ or changing the associated UAV-BS of a previous ground user is $p_{\max}$ or $(p_{\max} - p_{\min})$, respectively, the following inequality is achieved:

$$\begin{aligned} &P(\Delta_{I_{\text{UE},1}\cup\{u_{\text{new},1}\}}) \leq P(\Delta^{*1}) + p_{\max} + \\ &(J-1)(p_{\max} - p_{\min}) \\ &= f_{I_{\text{UE},1}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) + p_{\max} + \\ &(J-1)(p_{\max} - p_{\min}), \end{aligned} \tag{3.29}$$

where $P(\Delta_{I_{\text{UE},1}\cup\{u_{\text{new},1}\}})$ and $P(\Delta^{*1})$ are the values of system transmission power consumption related to $\Delta_{I_{\text{UE},1}\cup\{u_{\text{new},1}\}}$ and Δ^{*1}, respectively.

Similarly, the following inequalities are also proved:

$$\begin{aligned} &P(\Delta_{I_{\text{UE},1}\cup\{u_{\text{new},1},u_{\text{new},2}\}}) \leq P(\Delta_{I_{\text{UE},1}\cup\{u_{\text{new},1}\}}) + \\ &p_{\max} + (J-1)(p_{\max} - p_{\min}) \\ &... \\ &P(\Delta_{I_{\text{UE},2}}) \leq P(\Delta_{I_{\text{UE},1}\cup\{u_{\text{new},1},u_{\text{new},2},...,u_{\text{new},(m-1)}\}}) + \\ &p_{\max} + (J-1)(p_{\max} - p_{\min}), \end{aligned} \tag{3.30}$$

where $\Delta_{I_{\text{UE},1}\cup\{u_{\text{new},1},u_{\text{new},2}\}}$, ..., $\Delta_{I_{\text{UE},1}\cup\{u_{\text{new},1},u_{\text{new},2},...,u_{\text{new},(m-1)}\}}$, and $\Delta_{I_{\text{UE},2}}$ are the feasible user association strategies of $I_{\text{UE},1}\cup\{u_{\text{new},1}, u_{\text{new},2}\}$, ..., $I_{\text{UE},1}\cup\{u_{\text{new},1}, u_{\text{new},2}, ..., u_{\text{new},(m-1)}\}$, and $I_{\text{UE},2}$, respectively.

Furthermore, there is $f_{I_{\text{UE},2}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) \leq P(\Delta_{I_{\text{UE},2}})$ due to the fact that $f_{I_{\text{UE},2}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1})$ is the optimal value of user association sub-problem related to $I_{\text{UE},2}$ when the UAV-BS deployment strategy is $(x_1^{*1},...,x_J^{*1},y_1^{*1},...,y_J^{*1})$. Thus, (3.21) is achieved through (3.29) and (3.30). **Lemma 2** is proved.

3.8 Proof of Proposition 1

When $I_{\text{UE},2}$ is obtained by adding m new ground users into $I_{\text{UE},1}$, this work denotes $I_{\text{UE},2}$ as $I_{\text{UE},1}\cup\{u_{\text{new},1}, u_{\text{new},2}, ..., u_{\text{new},m}\}$ without loss of generality. Since the transmission power of

an arbitrary UAV-BS in set J_{UAV} to serve a ground user is not less than $p_{\min}$, the following inequality is received:

$$\begin{aligned} &f_{I_{\mathrm{UE},2}}(x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2}) \geq \\ &mp_{\min} + f_{I_{\mathrm{UE},1}}(x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2}), \end{aligned} \tag{3.31}$$

where $f_{I_{\mathrm{UE},1}}(x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2})$ is the optimal value of user association sub-problem related to $I_{\mathrm{UE},1}$ when the UAV-BS deployment strategy is $x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2}$.

Since $(x_1^{*1},...,x_J^{*1},y_1^{*1},...,y_J^{*1})$ is the optimal UAV-BS deployment strategy for $I_{\mathrm{UE},1}$, the following inequality can be achieved:

$$\begin{aligned} &f_{I_{\mathrm{UE},1}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) \leq \\ &f_{I_{\mathrm{UE},1}}(x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2}). \end{aligned} \tag{3.32}$$

According to **Lemma 2**, there are:

$$\begin{aligned} &f_{I_{\mathrm{UE},2}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) \leq \\ &f_{I_{\mathrm{UE},1}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) + \\ &m[p_{\max} + (J-1)(p_{\max} - p_{\min})]. \end{aligned} \tag{3.33}$$

By jointly considering (3.31), (3.32), and (3.33), (3.22) can be got immediately.

When $I_{\mathrm{UE},2}$ is obtained by removing m ground users off $I_{\mathrm{UE},1}$, this work denotes $I_{\mathrm{UE},1}$ as $I_{\mathrm{UE},2} \cup \{u_{\mathrm{new},1}, u_{\mathrm{new},2}, ..., u_{\mathrm{new},m}\}$ without loss of generality. According to **Lemma 2**, there is:

$$\begin{aligned} &f_{I_{\mathrm{UE},1}}(x_1^{*2},\ ...,\ x_J^{*2},\ y_1^{*2},\ ...,\ y_J^{*2}) \leq \\ &f_{I_{\mathrm{UE},2}}(x_1^{*2},\ ...,\ x_J^{*2},\ y_1^{*2},\ ...,\ y_J^{*2}) + \\ &m[p_{\max} + (J-1)(p_{\max} - p_{\min})]. \end{aligned} \tag{3.34}$$

Furthermore, since the transmission power of an arbitrary UAV-BS in set J_{UAV} to serve a ground user is not less than $p_{\min}$, the following inequality is arrived:

$$\begin{aligned} &f_{I_{\mathrm{UE},1}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) \geq \\ &mp_{\min} + f_{I_{\mathrm{UE},2}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}). \end{aligned} \tag{3.35}$$

Using the inequality in (3.32) again and combining (3.34) with (3.35), (3.22) can be got immediately.

So when $I_{\mathrm{UE},2}$ is acquired by adding m ground users into or removing m ground users off $I_{\mathrm{UE},1}$, (3.22) can be satisfied. This work arrives at **Proposition 1**.

3.9 Proof of Lemma 3

When n ground users in $I_{\mathrm{UE},1}$ change their position grids in region R and generate $I_{\mathrm{UE},2}$, this work denotes $I_{\mathrm{UE},1}$ as $I_{\mathrm{UE,stable}} \cup \{u_{\mathrm{move},1}, u_{\mathrm{move},2}, ..., u_{\mathrm{move},n}\}$. $I_{\mathrm{UE,stable}}$ is the set of ground users in $I_{\mathrm{UE},1}$ remaining stable and $\{u_{\mathrm{move},1}, u_{\mathrm{move},2}, ..., u_{\mathrm{move},n}\}$ is the set of ground users who will move inside R. After the ground users in $\{u_{\mathrm{move},1}, u_{\mathrm{move},2}, ..., u_{\mathrm{move},n}\}$ have been allocated at their new positions, this work denotes $I_{\mathrm{UE},2}$ as $I_{\mathrm{UE,stable}} \cup \{u'_{\mathrm{move},1}, u'_{\mathrm{move},2}, ..., u'_{\mathrm{move},n}\}$. Since the transmission power of an arbitrary UAV-BS in set J_{UAV} to serve a ground user is not less than $p_{\min}$, the following inequality is received:

$$\begin{aligned} &f_{I_{\mathrm{UE},1}}(x_1^{*1}, ..., x_J^{*1}, y_1^{*1}, ..., y_J^{*1}) \geq \\ &np_{\min} + f_{I_{\mathrm{UE,stable}}}(x_1^{*1}, ..., x_J^{*1}, y_1^{*1}, ..., y_J^{*1}), \end{aligned} \tag{3.36}$$

where $f_{I_{\mathrm{UE,stable}}}(x_1^{*1}, ..., x_J^{*1}, y_1^{*1}, ..., y_J^{*1})$ is the optimal value of user association sub-problem related to $I_{\mathrm{UE,stable}}$ when the UAV-BS deployment strategy is $(x_1^{*1}, ..., x_J^{*1}, y_1^{*1}, ..., y_J^{*1})$.

Since $I_{\mathrm{UE},2}$ can be regarded as the ground user set obtained by adding the n ground users in $\{u'_{\mathrm{move},1}, u'_{\mathrm{move},2}, ..., u'_{\mathrm{move},n}\}$, the following inequality is achieved based on **Lemma 2**:

$$\begin{aligned} &f_{I_{\mathrm{UE},2}}(x_1^{*1}, ..., x_J^{*1}, y_1^{*1}, ..., y_J^{*1}) \leq \\ &f_{I_{\mathrm{UE,stable}}}(x_1^{*1}, ..., x_J^{*1}, y_1^{*1}, ..., y_J^{*1}) + \\ &n[p_{\max} + (J-1)(p_{\max} - p_{\min})]. \end{aligned} \tag{3.37}$$

Combining (3.36) with (3.37), (3.23) is achieved. Thus, **Lemma 3** is proved.

3.10 Proof of Proposition 2

When the UAV-BS deployment is fixed to $(x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2})$, by following Lemma 3, the following inequality is achieved,

$$\begin{aligned} &f_{I_{\mathrm{UE},1}}(x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2}) \leq \\ &f_{I_{\mathrm{UE},2}}(x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2}) + \\ &nJ(p_{\max}-p_{\min}). \end{aligned} \tag{3.38}$$

That is because transforming $I_{\mathrm{UE},1}$ to $I_{\mathrm{UE},2}$ is symmetrical to transforming $I_{\mathrm{UE},2}$ to $I_{\mathrm{UE},1}$. The number of moved UEs is same as n.

When the UAV-BSs' locations are changed, $f_{I_{\mathrm{UE},1}}(x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2})$ cannot be less than the optimum solution $f_{I_{\mathrm{UE},1}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1})$:

$$\begin{aligned} &f_{I_{\mathrm{UE},1}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) \leq \\ &f_{I_{\mathrm{UE},1}}(x_1^{*2},...,x_J^{*2},y_1^{*2},...,y_J^{*2}). \end{aligned} \tag{3.39}$$

According to (3.23), (3.38), and (3.39), the following inequality is derived:

$$\begin{aligned} &f_{I_{\mathrm{UE},2}}(x_1^{*1},\ ...,\ x_J^{*1},\ y_1^{*1},\ ...,\ y_J^{*1}) \leq \\ &f_{I_{\mathrm{UE},2}}(x_1^{*2},\ ...,\ x_J^{*2},\ y_1^{*2},\ ...,\ y_J^{*2}) + \\ &2nJ(p_{\max}-p_{\min}). \end{aligned} \tag{3.40}$$

The prove of **Proposition 2** is done.

Chapter 4

Context-Aware Mobile Traffic Pattern Modelling

Overview

Modelling cellular traffic pattern plays a critical role to efficiently understand the traffic distribution which will guide the resource allocation. Not only the traffic fluctuation of each cell but also the users' distribution inside the cells will indicate the location and size of hotspots. Although current methods can indicate the traffic fluctuation of each cell, it is still not enough as optimisation techniques require to know the traffic distribution inside cells. In this work, Neural Network is used to improve the resolution to intra-cell level by modelling the hotspots. Here, a similarity measurement is designed to quantify how two patterns are similar to each other. This work applied this measurement on the GPS geo-tags of users and found that in one of three periods (day, evening, and night), the hotspots distributions are more similar than the other periods. Then for each period, Generative Adversarial Networks (GAN) is used to train a generator for modelling the intra-cell hotspots distribution. Such a trained generator can also continuously generate convincing artificial-data to solve the few-shot data problem. The similarity measurement gives high similarity (above 0.8) between generated artificial-data and the real test data.

4.1 Introduction

There are currently two main ideas to model the high-resolution changes in traffic distribution. The first one concentrated on using a parametric specification of a probability distribution function[183], such as α-stable [131] and Zipf distribution [130]. This idea is not complicated in implementation. The data for modelling is the cellular traffic collected from base stations or the traffic in grids described by histograms. The parameters of statistic functions are tuned manually or using maximum likelihood [131] but still challenging to approximate many intractable probabilistic-computations [184]. The other idea of modelling traffic pattern is about using machine learning to train a generative network which learns the traffic distribution with parameter tuning by gradient descent [127]. Generative Adversarial Networks (GAN) is a typical application in this area. It executes an adversarial training process, which is a mini-max game between a discriminator and a generator. This method provides not only a pattern generator but also a discriminator to give judgement if the pattern becomes different. An example of the GAN-based pattern generation is in [145]. They modelled a city-scale spatial traffic distribution, but the resolution had reached the roof (minimum grid size $235m \times 235m$).

Nevertheless, these researches are facing the following challenges:

- Low resolution: the maximum spatial resolution is at a cell-level [131]. It is caused by the resolution of cellular data which is collected from base stations.
- Operator biased: Using the operator's own data can be biased to one operator. An operator-neutral data is required to reflect the traffic characteristic from all operators.
- Temporal dimension not considered: the spatial traffic changes along with time have not been taken into consideration. Accordingly, previous traffic pattern modelling methods can only be used as a reference for BS deployment but not the real-time network optimisation.

To improve in these aspects, GPS geo-tags based traffic pattern modelling emerges. In the previous work [1], the authors found that the GPS geo-tags in social networks are correlated to cellular traffic of different cells. And such a correlation has been modelled through

regression. Consequently, geo-tagged social network data can improve the resolution of the traffic pattern to higher than cell-level (to intra-cell level) based on the accurate geo-tags [1].

Here, the idea is to use geo-tagged social network data to train a reliable generator for modelling the real-world traffic distribution and expand data-set, so the low-resolution and operator biased problems are solved by using precise geo-tags. The temporal-dimension problem is solved by the designed toolchain that provides a convincing artificial-data generator for each temporal periods based on only a limited dataset. This design becomes useful in the conditions with few-shot data.

In summary, this work contributes to three aspects: The first and foremost, both spatial and temporal dimensions are considered to analyse the traffic distributions. A new similarity function is defined to pre-process the training data. Then, a generator is trained by an adversarial process (GAN) for modelling the traffic distributions of diverse intervals. The trained generator can continuously generate spatial-temporal hotspots with high-resolution. Finally, this work uses the designed similarity function to quantify the reliability of the generator. The results show that generated patterns have high similarity compared to the other four distributions.

4.2 Preliminary Knowledge

4.2.1 Generative Adversarial Networks

As stated in the introduction, the generator in GAN can be iteratively trained to model the regular hotspot distribution. This technology was first proposed by I. J. Goodfellow et al. [184] in 2014. The generator is trained to generate a fake pattern and maximise the probability to fool the discriminator with judging the fake one as a real pattern. After some iterations with convergence, the network not only models the actual pattern but also can re-generate reliable new patterns to expand the data amount.

The basic framework of GAN form [184] is given in Fig. 4.1. The two neural networks in GAN are trained simultaneously for the generator G to learn the data distribution while the discriminator D probabilistically judges whether the generated data (from G) belongs to

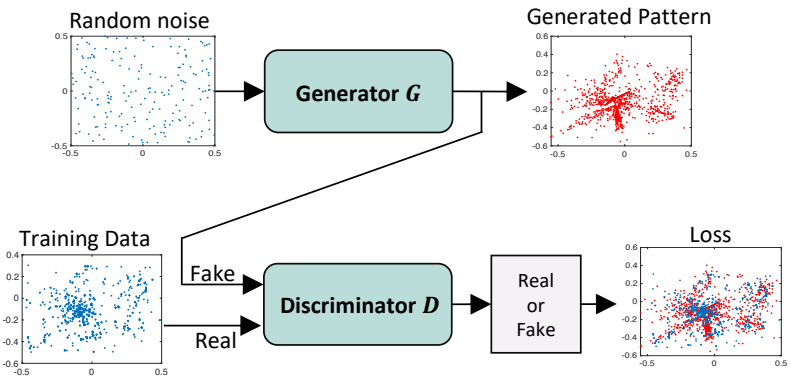


Fig. 4.1 The general framework of GAN.

the true data. It is a mini-max game between two players, G and D. The training process for D is maximising the probability of classifying the generated data to fake data. In contrast, G aims at maximising the probability of D making mistakes.

4.2.2 GPS Geo-Tags based Mobile Traffic Demand Estimation

The previous work of our group [1] quantified the linear relationship between 3G traffic and number of GPS geo-tags in the social network, Twitter, using log-linear regression. Even though geo-tagged Tweets occupy a small partition among all Tweets, the linear relationship still exists. The Twitter dataset comes from Twitter company, GNIP APIs. And the London cell information comes from a UK operator, Vodafone. The estimated Down-Link (DL) traffic $\hat{r}_{\mathrm{DL}}$ (kbps) in cluster o and time interval t is described by corresponding number of Tweets n_{ot} (Tweets/s):

$$\hat{r}_{\mathrm{DL}}^{ot} = 10^{b_{\mathrm{DL}}}\left(\frac{n_{\mathrm{ot}}}{\tau}\right)^{a_{\mathrm{DL}}} \tag{4.1}$$

where $[a_{\mathrm{DL}} = 0.88\mathrm{kb/Tweet} \quad b_{\mathrm{DL}} = 2.37\mathrm{kbps}]$ and τ is ratio between time interval and second (e.g, in this work $\tau = 3600\mathrm{s/hour}$). As a result, the real aggregate typical traffic in mobile cells can be accurately predicted for the next 2 hours using current geo-tagged Twitter activity level. The correlation is above 0.9. Besides, 71% to 79% of the variations in cellular traffic can be indicated by Twitter data. In that case, this thesis uses the geo-locations of Tweets as an indicator of mobile traffic. The spatial hotspots of Twitter are also cellular

hotspots in the cells. What needs to be done first is designing a similarity function and find in which period the Tweets distributions are similar.

4.3 Similarity Function

According to [183][127], social activities of human have periodicity, such as working, eating, or attending events. In that case, some particular places shall have users regularly arriving and leaving, such as home and metro station. The similarity means that users' GPS coordinates in different periods are close to each other. If two spatial traffic patterns are the same, the number of coordinates should be the same, and their latitude and longitude are also the same. Only a little difference can decrease the similarity, so the author chooses two factors (number and change of coordinates) to quantify the similarity.

The similarity describes the tendency of two patterns nearer to each other in the temporal dimension to be more similar than patterns further apart in time [185]. There are already some applications of the similarity measurement, such as Moran's I and random sample consensus [186]. These methods have been widely applied in gesture recognition with a fixed number of points. Nevertheless, there is still no proper method considering the similarity of the point clouds with a varying number of points. In this work, the author designs a suitable similarity function to quantify the similarity between patterns in the time dimension (geo-tagged coordinates point cloud).

The author denotes the two 2D point-clouds as A and B. The points in each cloud $a \in A$ and $b \in B$, and the number of coordinates are N_A and N_B. Euclidean distance matrix between a and each element in B is $||a,B||$, so the distance between a and its closest b is $\min||a,B||$. Then, the sum of minimum distance between A and B is $\sum_{a\in A} \min||a,B||$. This work presents a simple example to illustrate the relationship between the similarity and the above two factors in Fig.4.2. It is to quantify the similarity and compare which B is more similar to A. The difference between number of points (ΔN_1 and ΔN_2) and the sum of minimum distance (Δd_1 and Δd_2) are calculated in step 1 and step 2. In the final step, it is not difficult to find that

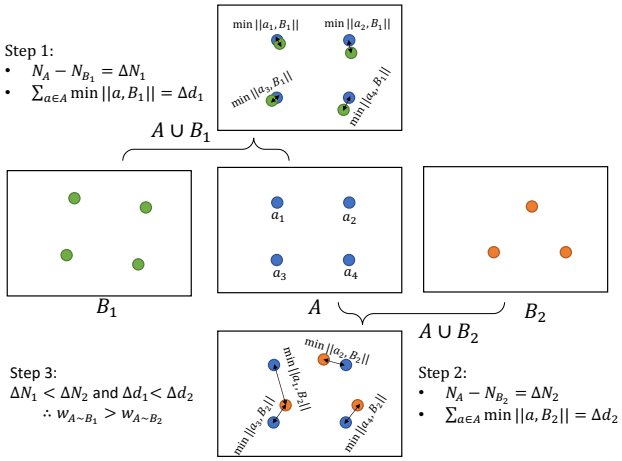


Fig. 4.2 An example of illustrating the process of calculating similarity.

the point cloud B_1 is more similar to A than B_2 ($w_{A\sim B_1} > w_{A\sim B_2}$) because $\Delta N_1 < \Delta N_2$ and $\Delta d_1 < \Delta d_2$ (step 3).

According to the above analysis, the similarity measurement of two point-clouds A and B is defined as:

$$w_{AB} = \frac{1}{1 + (k_1|N_A - N_B| + \varepsilon_1) * (k_2 \sum\limits_{a\in A} \min||a,B|| + \varepsilon_2)} \tag{4.2}$$

where w_{AB} is the weight of similarity. The two factors $\{\varepsilon_1, \varepsilon_2\}$ and $\{k_1, k_2\}$ control the significance of the number difference $|N_A - N_B|$ and minimum distance $\sum\limits_{a\in A} \min||a,B||$ to influence the similarity. And $\{\varepsilon_1, \varepsilon_2\}$ also help avoid the production to be zero. In the following data analytics, a temporally ordered sequence of point clouds $\mathscr{A} = \{A_1, A_2, ..., A_n\}$ require the measurement of a symmetry matrix of self similarity $W = [w_{ij}]_{n\times n}$, $0 < w_{ij} \leq 1$ and $w_{ij} = w_{ji}$, diagonal elements are one $(w_{ii} = 1)$. Therefore, the square symmetric similarity matrix W between pairs of pattern features is shown as follow:

$$W = \begin{bmatrix} 1 & w_{12} & w_{13} & \dots & w_{1n} \\ w_{21} & 1 & w_{23} & \dots & w_{2n} \\ w_{31} & w_{32} & 1 & \dots & w_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & w_{n3} & \dots & 1 \end{bmatrix} \tag{4.3}$$

where n is the number of time intervals in the data set.

4.4 Data Pre-Processing

4.4.1 Temporal Pre-Processing: High Similarity Period Selection

This work applied the designed similarity measurement on a captured London GPS records dataset from Twitter for verification. There are 0.6 million geo-tagged records for the Greater London and surrounding suburbs area from 15/02/2016 to 28/02/2016 (total two weeks and time resolution in seconds). The similarity matrix of two weeks is visualised in Fig.4.3. In detail, one point cloud includes the coordinates distributing on the map in 0.5 hours, so there is a total of 672 frames in two weeks. The quantification of similarity is visualised as a heat level that bright pixel represents high similarity. The temporal origin is located at the top-left corner from 00:00 AM on 15/02/2016. From this result, the findings are summarised as follow (the item numbers are also marked on Fig.4.3):

1. Fig. 4.3 shows alternately bright and dark rectangles which are caused by the socially in-active time of night (1:30 AM - 6:30 AM) and active time (7:00 AM - 11:00 PM).

2. Week-days and weekends are different. Weekends are more similar with other weekends and vice versa, which can be seen from the darker columns of weekends mapping to week-days. Therefore, it would be better to build two types of models.

3. The awake time of weekends is later than weekdays. For example, as shown in the similarity of weekends, the active time started at around 10:00 AM and ended at around

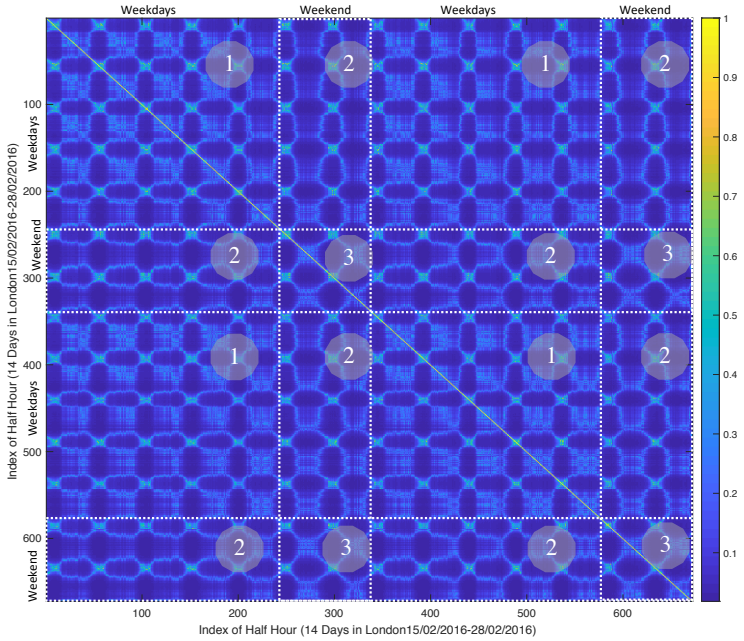


Fig. 4.3 The visualisation of similarity matrix of the GPS point cloud. The numbers indicate different kinds of similarities, weekdays' similarity (1), similarity between weekdays and weekends (2), weekends' similarity (3).

10:00 PM. It should be noted that this result does not mean consumers become more in-active during weekends. On the contrary, it verifies that user behaviours become more random (less similarity) during weekends than weekdays, so the high-similarity period shrinks.

The similarity in the same day can be further divided into three blocks. For example, in Fig. 4.4 with zoomed-in similarity matrix, three bright blocks are shown with indicating three periods in one day with high similarity, they are night (1:30 AM - 6:30 AM), day (7:00 AM – 15:30 PM), and evening (18:00 PM – 23:00 PM).

According to the above findings, this work chooses to use the data between 9:00 AM to 12:00 PM in the two weeks as the training data for modelling. Although the data in this period have an acceptable similarity, noise still exists and misdirects the modelling of hotspots resulting in slow convergence. In that case, a spacial pre-possessing procedure is required to filter the noise, which will be introduced in the next part.

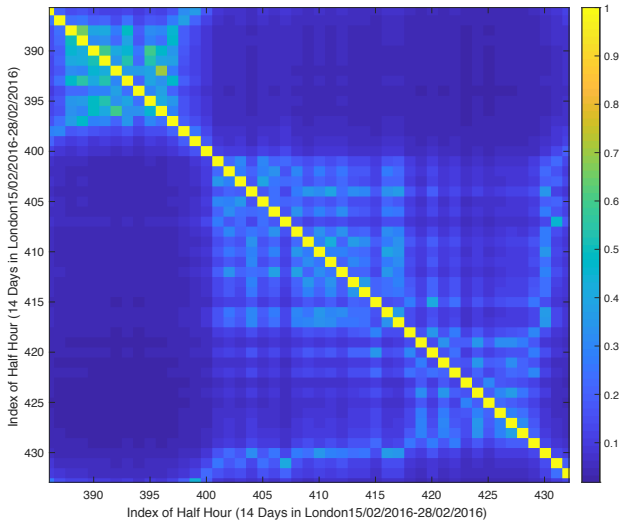


Fig. 4.4 One day of the similarity pattern with segmentation of night, day, and evening.

4.4.2 Spatial Pre-Processing: Noise Reduction by DBSCAN

To filter the noise, the author chooses a density-based unsupervised learning method, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) designed by M. Ester in [187]. This algorithm can automatically cluster the data in 'hotspots' (with many nearby neighbours) and regard the data with nearest neighbours far away as 'noise' so that the hotspots in the traffic distribution become the clusters generated by DBSCAN.

This work briefly illustrates the procedures of DBSCAN in Fig. 4.5 and explain as the following steps. Firstly, as shown in the left figure, a radius ε is visualised along with each coordinate (x, y) to provide a range (grey circles). A coordinate will be allocated to clusters (hotspots) when they have neighbouring coordinates $\geq$ minThreshold in its circle (which is 2 in this example). In contrast, the edge coordinates of clusters own neighbours but $<$ minThreshold.

Moreover, as the noise coordinates hold no neighbours, they can be picked and filtered out. Then, as highlighted by dash line circles in the middle figure, the algorithm has disregarded the noise coordinates and grouped the cluster coordinates into three clusters. These clusters (hotspots) not only suit the required density but also reduce the further influence of noise to

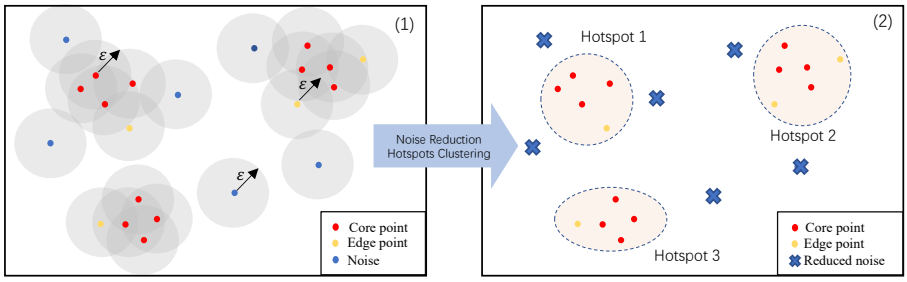


Fig. 4.5 The process of using DBSCAN to reduce the noise data and reserve the clusters (hotspots).

improve the performance. Finally, the cluster centroid can be produced by the mean of all the locations of points in the cluster.

This pre-processing procedure will provide the filtered data that are the cluster coordinates with a minimum density threshold. The noise data (with low density) has been ignored to provide a fast convergence of the modelling. Then, the author uses such a data set to train neural networks for modelling these clusters, which are the hotspots in traffic distribution.

4.5 GAN-based Traffic Pattern Modelling

The requirements are to model the spatial distribution and generate convincing artificial data to expand the limited dataset. In machine learning, generative networks own the ability to generate artificial data. Moreover, GAN owns a discriminator to quantify how convincing the generated patterns are, so it suits the requirements.

The application framework of GAN is presented in Fig. 4.6. This work gives a case study based on the GAN designed in [184]. In detail, the training data is denoted as x. The generator G needs to model the distribution p_g over data x using input noise variables $z \sim p_z(z)$ and outputs a distribution sample $G(z)$. The discriminator D regards x and $G(z)$ as inputs and produces single scalars $D(x)$ and $D(G(z))$. The discriminator D is trained to maximise $D(x)$ and $log(1-D(G(z)))$ that higher probability comes from true data x but not the generated data $G(z)$. In this work, G and D have two layers with 16 neurons in each layer and a Leaky ReLU activation function. Generally, D and G play the mini-max game to

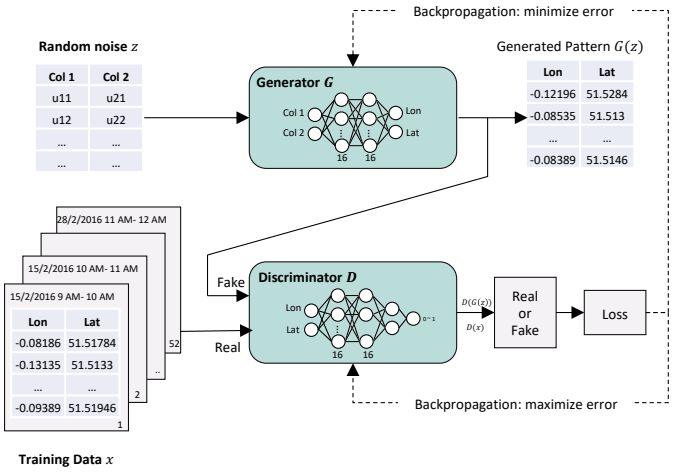


Fig. 4.6 The detail framework of GAN in this study.

mutually improve their abilities of two neural networks with the following joint objective $V(D,G)$ which designed by I. J. Goodfellow et al. in [184]:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_x(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]. \quad (4.4)$$

At the beginning of training, the noise z is fed to the generator with uniform distribution. With the comparison of real pattern x, D is trained first and can easily reject the pattern from the generator. Then, G is updated according to the loss with back-propagation. The training time in each iteration is the same for both G and D. The training process is to optimise the parameters of the two neural networks. G and D are trained iteratively (one at a time) for n_G and n_D sub-epochs until the convergence of loss functions.

The sigmoid cross-entropy in Tensorflow is used to calculate the loss. The logits (output of neural networks) is transformed by a sigmoid function to normalise the range to $[-1,1]$. Moreover, RMSprop optimiser is chosen in the training process to find a trade-off between performance and ability of convergence. The GAN converges until D is unable to identify the difference between the real distribution and the generated distribution ($p_g = p_x$).

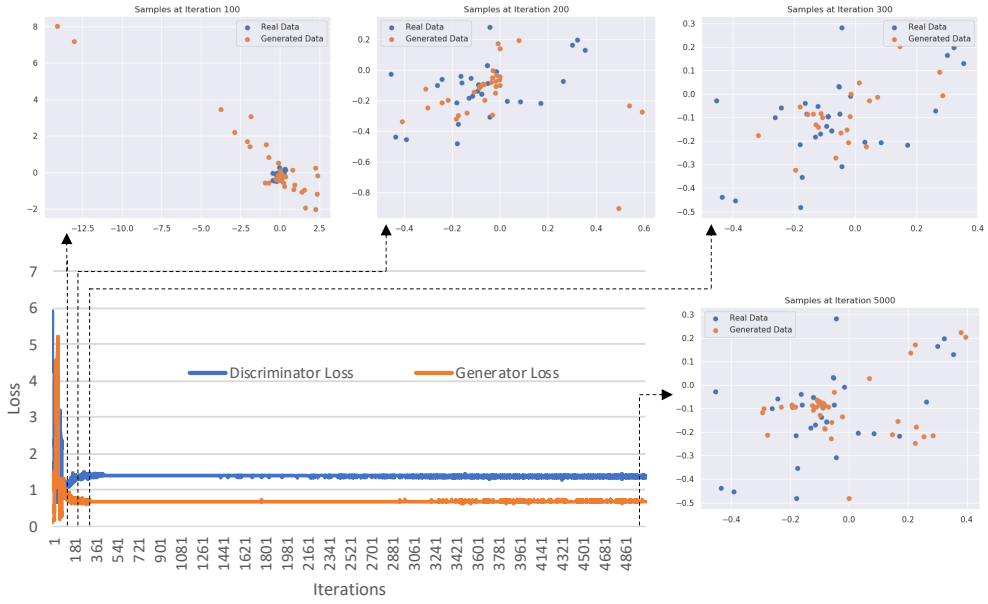


Fig. 4.7 Results of pattern generation.

4.6 Results

In this experiment, the author used MATLAB 2018b to calculate the similarity, Scikit-Learn 0.20.3 for DBSCAN, and TensorFlow 1.12.0 for training and testing GAN on Python 3.6.8. Natural logarithm is selected in the loss function (as well as objective function). As the training process converges when the discriminator can not distinguish the pattern is generated or real, the output of D is always $\frac{1}{2}$ for every input. According to Equation 4.4, $V(D,G)$ converges to $2\ln(\frac{1}{2}) \approx -1.39$. The line chart in Fig. 4.7 shows the changes of loss in each iteration, the blue line of D loss converges at around 1.39 because the minus is dismissed for convenience. Similarly, the generator loss (orange line) converges to $-\ln(\frac{1}{2}) \approx 0.69$.

4.6.1 Generated Tweets Locations

In Fig. 4.7, the dash-line arrows point to the patterns by comparing the generated data (orange points) with one of the real patches in the training data (blue points). Along with the convergence of the loss, the generated data is becoming acceptable after the iteration 300. The locations of real and generated coordinates are not exactly the same but indicating

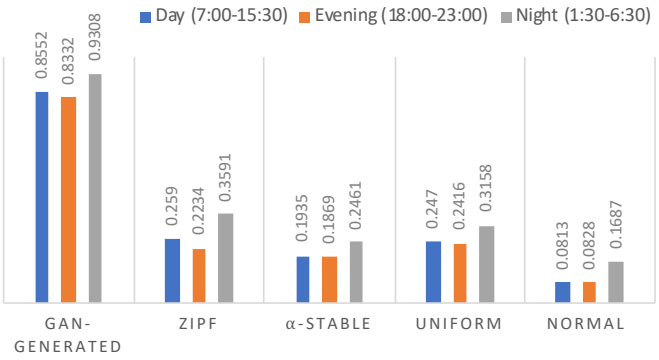


Fig. 4.8 Similarity comparison with the test data set.

the common distribution to reveal the regular hotspots. Also, the patterns for training are not constant in different days, so the variance will not be averaged to each day. Currently, the generator finishes the modelling of hotspots and can be used to expand the dataset with reliable loss.

Besides training and generating, the designed similarity measurement is also applied to the generated distribution to quantify how similar it is to the real distribution (test data). To give a reliable comparison of the performance, this work also did the same procedures on the other four probabilistic distributions. The similarity comparison is shown in Fig. 4.8. The author successively trained three generators for the three time period (day, evening, and night) during weekdays (Monday to Friday). GAN-generated patterns own the highest similarity (blue columns above 0.8) due to the similar number of coordinates and close locations. The author also provides the similarity between test data and the other four probabilistic distributions. The number of points is set to be same as the testing data. In that case, the difference between distribution becomes the only factor to influence the similarity. As shown in the result, the similarity measurements (orange and grey columns) are much lower (below 0.4) because of the diverse spatial distributions. The actual Tweets distribution contains not only sparsely distributed ones but also the hot spots. Zipf and α-stable distributions (codes [188]) gain a low performance because of lacking the ability to model sparse distribution. In contrast, uniform distribution is not good at modelling the hotspots. Note that, combinations of such probabilistic distributions will gain better performance of modelling but requires

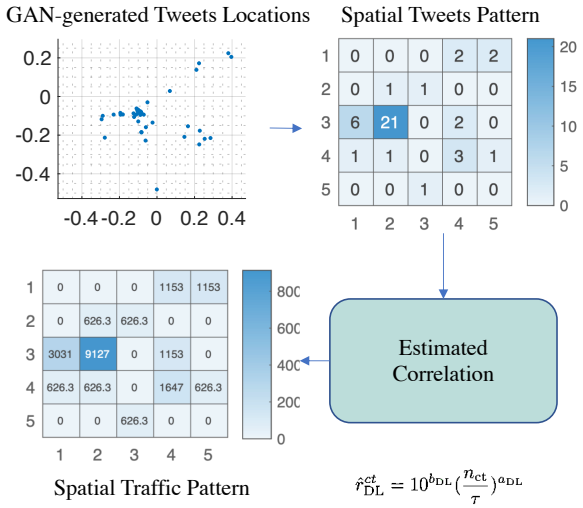


Fig. 4.9 Estimated cellular traffic pattern based on the generated locations of Tweets and their correlation to cellular traffic.

efforts for tuning the parameters. That reflects the convenience of using GAN from another aspect.

4.6.2 Mobile Traffic Pattern

The last step of this toolchain should be transforming the traffic of Tweets to the estimated traffic of cellular networks. Note that, the core function of such transformation is applying the estimation correlation. Here, Eq. 1 in preliminary knowledge is used. Thanks to the efforts of GAN, the generated data are coordinates so that the resolution of rasterisation is flexible. As shown in Fig. 4.9, the GAN-generated coordinates are rasterised into a spatial Tweets pattern (two-dimensional histogram). In each grid, the number of coordinates is counted and fed to the estimated correlation. Finally, the spatial traffic pattern is generated with estimated traffic in each grid. The size of the grid can be set flexibly. This work provides a $5 * 5$ example to visualise the process in a good manner, other sizes can also be selected, such as $10 * 10$ or $20 * 20$.

4.7 Conclusion

The data-driven cellular network optimisation needs to face the conditions with few-shot data. Due to this requirement, this work leverages the positive correlation between cellular network traffic and GPS records in social networks to model the intra-cell traffic distribution by neural networks for further few-shot learning studies.

This work firstly designed a new similarity measurement to determine the high-similarity time period of training data. Then, the non-hotspots data are filtered using density-based unsupervised learning (DBSCAN). Finally, this work trained a neural network as the generator to model the actual hotspot distribution through an adversarial process (GAN). Such a generator can be used to expand the limited dataset and continuously generate reliable traffic distribution.

In future, there are still some aspects requiring improvements. In the first place, the training process of GAN may meet non-convergence. The loss of discriminator will not converge, and the generator produces patterns with low similarity. This is a well-known training difficulty of GAN. It may be improved by selecting different optimiser.

Chapter 5

Context-Aware Network Off-Loading by UAVs

Overview

UAVs have become popular carriers of aerial BSs to cover temporal hotspots and off-load the transmission requirements of mobile users. Although the UAV-BSs can ignore the terrestrial obstacles, using them to fast deploy for off-loading still faces the *time-efficiency* problem of jointly optimising multi-objectives: such as UAV-BSs' amount, locations, and allocating resource blocks. This work transforms the above joint optimisation problem into a combinatorial problem and improves current Simulated Annealing (SA) solutions by CAPO with both time-efficiency and robustness. Results show that the CAPO-based frameworks enable a typical SA algorithm to finish 30% more work under a time limitation. The author also employ few-shot leaning to improve its robustness while facing the data scarcity.

5.1 Introduction

Compared with the deployment optimisation, these offloading works do not need the UAV-BSs to associate all the users in the region. In contrast, they only need to provide temporal coverage to the short-term hotspots, so the optimisation problem requires to be re-formulated. As reviewed, recent researches commonly faced time-efficiency problems because of opti-

mising multiple targets. CAPO is stated to solve this problem by taking the advantage of the seasonality of users' distribution. In that case, this section contributes in the following aspects:

1. This work converts the above multi-target joint optimisation problem into a combinatorial problem and uses a basic heuristic method, Simulated Annealing, to solve it without considering the optimisation complexity.

2. To improve the computation efficiency of Simulated Annealing, it is coordinated with the data-aware (CAPO) framework. In detail, probabilistic models (historical experience) are used to substitute traditional heuristic function to generate experience-based potential solutions. A classifier is trained based on the historical data to select which probabilistic model to use for helping current optimisation.

3. This work also manages to run the above design on few-shot data in which the classifier cannot be fully trained due to the lack of data to classify future conditions. This challenge causes mis-matching experience usage and further optimisation overhead. To overcome this challenge, this work uses generative neural networks to model the limited dataset and generate more artificial but convincing data to train the classifier. Then, the few-shot data problem is addressed that algorithm's robustness to face scarce data is improved.

4. Finally, this chapter uses real-world users' GPS locations to test the 1) basic, 2) data-aware, and 3) generative data-aware frameworks. The performance computation is indicated by computation iterations and optimisation overheads.

This chapter is organised as follows: Section 5.2 provides the aerial-ground system model and formulates the multi-UAV off-loading optimisation as a combinatorial problem. Section 5.3 illustrates three solutions: (1) traditional Simulated Annealing algorithm, (2) CAPO aided SA for time-efficiency, (3) few-shot learning aided SA for robustness. Section 5.4 lists the results of testing the three solutions with real-world user GPS locations. Finally, Section 5.5 concludes this chapter.

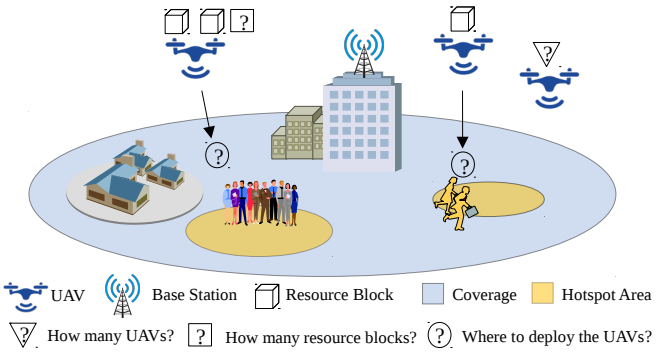


Fig. 5.1 The scenario of offloading by multiple UAV-BSs.

5.2 System Model

In this chapter, the network architecture follows the heterogeneous networks [4] that the overall coverage is served by a macro-cell, while hotspots' offloading is achieved by adding micro-cells (served by UAV-BSs in this work). The system model of deploying the UAV-BSs as well as their resources is visualised in Fig. 5.1. In the region served by a macro cell (blue region), the network becomes overload so hotspots emerge and shown as yellow regions. Some temporal mobile UAV-BSs are required to be dispatched to some locations for best offloading the macro BS. Different hotspots require diverse amount of resources blocks (white cubes). Generally, there are three parameters (number of UAVs, amount of resource blocks, and locations of UAVs) to be optimised. There is a central controller responsible for collecting information and computing algorithms. Air-to-ground downlinks are considered because this kind of traffic takes the majority. The terrestrial BSs use different frequency bands and the UAV-BS owns directional antennas, so beamforming is enabled. The interference is managed well among UAVs. The performance of UAV-BS has the highest priority to be optimised, so the handover management and interference management are assumed to be in ideal conditions.

The coordinate of each user is denoted as $(x_i, y_i)^T \in \mathbb{R}^{2\times 1}$, in which $|x_i| \leq v_{\text{xmax}}, v_{\text{xmax}} \in \mathbb{R}^+$ and $|y_i| \leq v_{\text{ymax}}, v_{\text{ymax}} \in \mathbb{R}^+$. v_{xmax} and v_{ymax} limits the range of the experimental region. And a list of user coordinates is $(x_I, y_I)^T \in \mathbb{R}^{2\times N_i}$, where N_i is the amount of users. Similarly, the list of UAV-BSs' locations is $(x_J, y_J)^T \in \mathbb{R}^{2\times N_j}$, so N_j is the amount of UAV-BSs. Each

UAV-BS's coordinate is described as $(x_j, y_j)^T \in \mathbb{R}^{2\times 1}$. It should also be deployed in the experimental region, so $|x_j| \leq v_{\text{xmax}}$ and $|y_j| \leq v_{\text{ymax}}$. Note that, the region in this work in continuous, which was discrete (grids) in the last chapter. This setting can increase the accuracy of all coordinates but lead to much higher computation complexity, so the exhaustive methods are not available here. Next, the horizontal distance from UAV-BS j to the UE i is the Euclidean norm d_{ij}:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{5.1}$$

This work uses the drone-ground channel model from [59]. The altitude of aerial BSs is denoted as h, so the elevation angle is:

$$\theta'_{ij} = \arctan(h/d_{ij}) \tag{5.2}$$

For simplicity, the altitude h is fixed for all UAV-BSs. The probability of LoS (P^{LoS}) is formulated as:

$$P^{\text{LoS}} = \frac{1}{1 + a\exp(-b(\frac{180}{\pi}\theta'_{ij} - a))} \tag{5.3}$$

$$P^{\text{NLoS}} = 1 - P^{\text{LoS}} \tag{5.4}$$

where a and b are two constant parameters determined by urban or rural area. In [59], urban environment has (9.6117, 0.2782) for a and b. As the radio signal first propagates in free space, then moves to the urban environment, the excessive pathloss caused by man-made structures is additive to the free-space pathloss and described by Gaussian distribution. Its mean is considered rather than the variance (random behaviour). $\mu^{\textbf{LoS}}$ and $\mu^{\textbf{NLoS}}$ are the means of excessive loss for LoS and Non-LoS (NLoS) connections which are (1, 20) for urban areas. Therefore, the pathloss model in [2] in decibel is:

$$\text{PL}(\textbf{dB}) = 20log(\frac{4\pi f r_{ij}}{c}) + P^{\text{LoS}}\mu^{\text{LoS}} + P^{\text{NLoS}}\mu^{\text{NLoS}} \tag{5.5}$$

$$r_{ij} = \sqrt{d_{ij}^2 + h^2} \tag{5.6}$$

where f is the frequency, c is the speed of light, and r_{ij} is the distance between user and the UAV-BS. Given a fixed transmission power $P_T(\mathrm{dB})$, the received power in decibel can be described as:

$$P_R(\mathrm{dB}) = P_T(\mathrm{dB}) - \mathrm{PL}(\mathrm{dB}) \tag{5.7}$$

Based on the above models, in this work γ_{ij} is denoted as the Signal-to-Noise Ratio (SNR) of UE i receiving the signal from UAV-BS j. The association between UE and BS is based on this parameter that γ_{ij} needs to satisfy the minimum requirement $\gamma_{ij} \geq \gamma_\theta$ to connect to UAV-BS. Here, γ_θ is a constant value. In that case, the user association runs in a greedy way that all users with better SNR will be connected in priority until the maximum capacity is reached or the rest UEs own $\gamma_{ij} < \gamma_\theta$. The total connected UEs of all UAV-BSs is denoted as N_i. And the equivalent average resource blocks prepared for each UE is R_i and it should be not less than a threshold R^θ to ensure the service quality. Therefore, total used resource blocks can be:

$$R = N_i \cdot R_i \geq N_i \cdot R^\theta \tag{5.8}$$

R can be regarded as the maximum capacity. Accordingly, N_i is influenced by three kinds of parameters: UAV-BSs' locations $\{x_J, y_J\}$, R, and N_j. Here, a function $f_i(\cdot)$ is used to describe computation of N_i through the system model:

$$N_i = f_i(\{x_J, y_J\}, R, N_j) \tag{5.9}$$

Generally, the optimisation target function (TF) is to achieve the minimum resource requirement of each UE and offload as many UEs as possible through adjusting the UAV-BSs' locations $\{x_J, y_J\}$, the allocated resource of each UAV-BSs (R), and the number of UAV-BS (N_j). Therefore, the target function is described as:

P1:

$$\underset{\{x_J,y_J\},R,N_j}{\text{maximise}}\ \text{TF} = \alpha_1 N_i - \alpha_2 R - \alpha_3 N_j \tag{5.10}$$

$$s.t. N_{\text{imin}} \leq N_i \leq N_{\text{imax}}, N_i \in \mathbb{Z}^+ \tag{5.11}$$

$$N_i = f_i(\{x_J, y_J\}, R, N_j), \{x_J, y_J\} \in \mathbb{R}^{N_j \times 2} \tag{5.12}$$

$$|x_J| \leq v_{\text{xmax}}, |y_J| \leq v_{\text{ymax}}, \{v_{\text{xmax}}, v_{\text{ymax}}\} \in \mathbb{R}^+ \tag{5.13}$$

$$N_i \cdot R^\theta \leq R \leq R_{\text{max}}, R \in \mathbb{Z}^+, R^\theta \in \mathbb{Z}^+ \tag{5.14}$$

$$N_{\text{jmin}} \leq N_j \leq N_{\text{jmax}}, N_j \in \mathbb{Z}^+ \tag{5.15}$$

In the above target function Eq. (5.10), x_J, y_J, R, N_j are optimised to maximise TF, and α_1 to α_3 are three positive weights to adjust diverse importance of N_i, R, and N_j. For example, when there is a short of UAV-BSs, minimising N_j becomes more important, so α_3 can be increased to force the target function to lean to the desired result. Moreover, $-\alpha_2$ and $-\alpha_3$ indicate that larger R and N_j can have negative influence to the target. Constraints (5.11), (5.12), and (5.13) clarify the range of N_i and its relation to x_J and y_J. N_{imin} decides the lowest offloaded users. In contrast, N_{imax} gives the maximum covered users. $\{v_{\text{xmax}}, v_{\text{ymax}}\}$ determines the terrestrial range for optimisation. Constraint (5.14) means that the resource blocks should satisfy the minimum requirement of each user. R_{max} indicates the maximum allowed resources blocks for offloading. And Constraint (5.15) limits the range of UAV-BSs' amounts. N_{jmin} is the minimum number of drones , and N_{jmax} is the maximum drones can be dispatched for this mission. Generally, it is an NP-hard combinatorial optimisation problem to find a grouping of discrete and finite parameters set $\{x_J, y_J, R, N_j\}$ satisfying the above conditions.

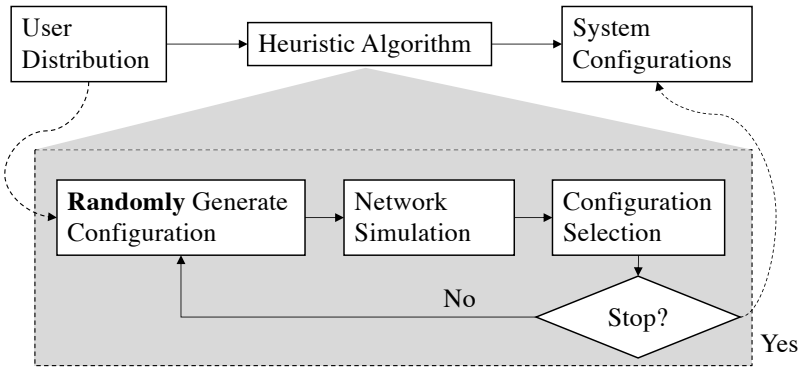


Fig. 5.2 General framework of Simulated Annealing algorithms to solve the combinatorial problems for off-loading.

5.3 Methodology

5.3.1 Simulated Annealing Algorithm

The above combinatorial problem is unable to be optimally solved by exhaustive searching in polynomial time. Alternatively, the heuristic methods can use finite iterations to output sub-optimal solutions. This thesis has reviewed these methods. Here, a general framework for the Simulated Annealing methods to solve problem P1 is given in Fig. 5.2.

As seen in the first row of the figure 5.2, there are three blocks, user distribution, heuristic algorithm, and the system configurations. The user distribution is the list of user coordinates $(x_I, y_I)^T$, which will be pushed as the input of the next block (named 'Heuristic Algorithm'). The sub-functions of this 'Heuristic Algorithm' has been expanded and shown in the box below (with grey background), and these sub-functions are 1) Randomly Generate Configuration, 2) Network Simulation, 3) Configuration Selection, and 4) Stop. The block named 'System Configurations' is the final output of this work flow. Details of each block are listed as following.

- User Distribution: The input of this framework. $(x_I, y_I)^T$ is collected and pushed to the next block.

- 1) Randomly Generates Configuration: this is the first sub-function which randomly generates a network configuration $\{x_J, y_J, R, N_j\}, \forall j$ (locations, resources, and num-

ber). These parameters are assumed to be independent and follow **uniform** distributions (denoted as $U(\cdot)$), so $x_J \sim U(0, v_{\text{xmax}}), y_J \sim U(0, v_{\text{ymax}})$, $R \sim U(0, R_{\max})$, $N_j \sim U(N_{\text{jmin}}, N_{\text{jmax}})$. At this stage, the uniform distributions are used because of adding **no bias**. This is common in using the heuristic algorithms. For example in Evolutionary Algorithm, it randomly creates an initial population according to the uniform distribution, and also like mutation with new genetic material in further generations. And in Simulated Annealing, it also generates random trails based on uniform distribution. The uniform distribution gives equal probability of all possible solutions to be tried.

- 2) Network Simulation: This is the second sub-function. The bundle of parameters $\{x_J, y_J, R, N_j\}, \forall j$ is received and tested here. This simulation runs with the system model and outputs the number of offloaded users (N_i). Note, the minimum resource requirement should be checked to ensure $R \geq N_i \cdot R^{\theta}$. If not, the above two blocks need to run again.

- 3) Configuration Selection: The above two steps will continue for several iterations, and better configurations are continuously accepted by this sub-function when they enable higher TF. For example, the fitness selection in Evolutionary Algorithm works in this way. Another example is in Simulated Annealing which stores the best configurations with lowering temperature.

- 4) Stop: Finally, a 'stop' mechanism in Fig. 5.2 is set up to terminate the heuristic process and output the accepted system configuration, such as setting maximum iterations, threshold of performance, or any other stopping criterion.

- System Configuration: This is the output of this framework. After running many loops, the accepted configuration will approach the optimum.

This algorithm has two advantages: 1) the 'Randomly Generate Configurations' block reserves potential to jump out of local optimums. 2) the 'Stop' block shuts down the process in limited iterations which offers time efficiency in computation. The results of testing this

framework on the problem P1 are given in Section 4.4.1. As expected, this multi-target problem can be addressed with sub-optimal solutions.

Nevertheless, there is still spaces to improve time efficiency. As the 'Randomly Generate Configurations' block assumes uniform distributions of parameters, it is for non-specific cases and causes mis-matching in specified cases. For example, the UAV-BSs should have high probability to hover above the clusters of users, such as tourist attractions or the transportation stations. And the low-density areas such as park, lakes, or river should not follow the same probability as the high density areas. In that case, the uniform distributions should be substituted with **specified models** to formulate the parameters' distribution.

5.3.2 Data-Aware Method

The direct way for specified models is fitting probabilistic models to the historical optimisation results. Many historical problems have been solved. And they may have found where the hotspots are and where the low-demand regions (e.g., parks and rivers) exist. These historical optimisation results become the source to build the probabilistic model for the substitution. This substitution will bring the following advantages: 1) computation resources are better allocated to highly-beneficial configurations (hotspots), 2) time-efficiency is further improved due to the reduced repetitive computation (in the low-demand regions). This improved framework relies on the historical data, so it is denoted as **data-aware** Simulated Annealing algorithms.

Analyzing Historical User Distribution

Data-Aware Simulated Annealing

Data-aware frameworks should own the ability to collect and store historical data, including user distributions and previously optimised configurations, then adaptively adjust the probabilistic models in the 'Randomly Generate Configuration'. It is established based on two pre-conditions: 1) the historical data can be collected. 2) the historical datasets are enough to be used to generate the probabilistic models. Here, these two conditions are assumed

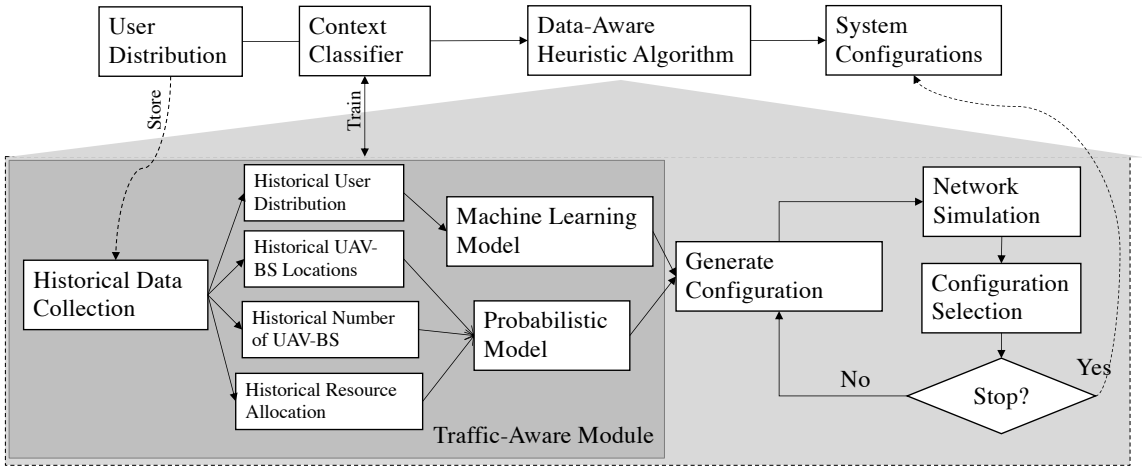


Fig. 5.3 The framework of data-awareness based Simulated Annealing algorithm. A new data-aware module is added to change way to heuristically generate potential configurations.

to be satisfied. Fig. 5.3 illustrates how the data-aware module is implemented. From the left side, the historical data is saved including historical user distribution $\{x_{It}, y_{It}\}$, previous UAV-BSs' locations $\{x_{Jt}, y_{Jt}\}$, previous number of UAV-BSs N_{it}, and previous resource allocation strategies R_t. Different probabilistic models can be used to describe the data, such as normal distribution, Zipf distribution, α-stable distribution, etc. This thesis denotes that S_{Jt} is the probabilistic model of $\{x_{Jt}, y_{Jt}\}$, $P(R_t)$ and $P(N_{jt})$ are the models of R_t and N_{jt} respectively.

As there are many models based on diverse temporal periods and geographical regions, a classifier is required to justify which model is most suitable for current condition. Therefore, the historical data is also used to train a classifier. The training data is labelled with different time stamps and region marks. When the current condition has been classified, the model of this class will be a representative model to generate potential configurations. Here, the K-Nearest-Neighbours (KNN) is selected for the classification. Indeed, it can be changed to other classifiers. The author selects KNN for simplicity here. The similarity between users' distributions is used to select the most similar neighbour, an example of this similarity function can be found in [189]. Next, these data-based probabilistic models **substitute** the previous uniform distributions $\{x_J, y_J\} \sim S_{Jt}$,$R_I \sim P(R_t)$,$N_j \sim P(N_{jt})$.

Further heuristic configuration generation is based on these new probabilistic models. Due to the seasonality of data, the common characteristics in historical data will repeat in future. This new framework is expected to save the heuristic iterations.

For measuring S_{Jt}, $P(R_t)$, and $P(N_{jt})$, a simple way is to use histogram, and another more complex way is using probabilistic functions to fit data. For example, S_{Jt} is denoted as the probabilistic model of historical distributions of UAV-BSs. A histogram divides region and counts number of coordinates in each grid then generates S_{Jt}. Moreover, describing the historical data as a parametrically specified probability distribution is also a choice, such as normal distribution. Here in this work, the histogram is used to generate S_{Jt}, and normal distributions are used for fitting R_t and N_{jt}: $P(R_t) \sim \mathcal{N}(\mu_R, \sigma_R^2)$, $P(N_{jt}) \sim \mathcal{N}(\mu_N, \sigma_N^2)$. This fitting is done by maximum likelihood and manual parameter selection. The following paragraphs give an example of applying the above data-aware framework on Simulated Annealing [190]:

1. Data-Aware Module: The algorithm firstly inputs settings of start- and stop- temperatures (T and T^{cold}) and the probabilistic models of S_{Jt}, $P(R_t)$, and $P(N_{jt})$.

2. Classifier: It takes current user distribution $\{x_I, y_I\}$ and historical user distributions $\{x_{It}, y_{It}\}$ as inputs. Then, it outputs the most suitable class ($t = t_1$) and selects S_{Jt_1}, $P(R_{t_1})$, and $P(N_{jt_1})$.

3. Generated Configuration: Then, it randomly samples the UAV-BS locations ($(x_J, y_J) \sim S_{Jt_1}$) , number of nodes ($N_j \sim P(N_{jt_1})$), and the amount of resource blocks ($R \sim P(R_{t_1})$). These probabilistic models are provided as examples, readers can change to other models better fitting the historical data.

4. Constraints Check: The generated parameter group ($\{x_J, y_J, R, N_j\}$) should subject to Constraints of P1.

5. Network Simulation: Then, the algorithm executes network system model simulation $N_i = f_i(x_J, y_J, R, N_j)$. After that, TF for this group of parameters is received. This step will be executed for several iterations until 'stop' command has been received.

6. Stop: The temperature T controls the time to stop the optimisation. It has a cold temperature T^{cold} to stop the process of finding global optimum when meeting $T \leq T^{cold}$. T has a high probability to decrease (annealing) when the new target function $\mathrm{TF}_{\mathrm{new}}$ is close but not larger than $\mathrm{TF}_{\mathrm{old}}$. Moreover, the annealing process is quicker with higher T. The algorithm accepts the parameter group with better meeting the target function, which is to maximise the TF in this study.

7. System Configuration: Finally, the algorithm produces the converging process of the optimised system configuration.

The results of testing this data-aware Simulated Annealing are provided in Section 4.4.2. It has been compared with non-data-aware conditions for quantifying the reduced computation iterations.

However, the data-aware algorithm takes the advantages of historical data but needs to face its shortages. The first one is the scarcity of data. For example, if the data does not contain historical parameters like $\{x_{Jt}, y_{Jt}, R_t, N_{Jt}\}$. In contrast, only historical user distributions $\{x_{It}, y_{It}\}$ are provided. How can the probabilistic models be generated? Secondly, collecting enough data to improve accuracy can be an ideal condition. If the data is limited to few shots, the classifier's training may not converge. How to reduce the optimisation overhead caused by reduced accuracy? Finally, manually choosing probabilistic models and tuning parameters are not as efficient as using neural network to model automatically. How to remove the manual involvement? Next part will manage to address these challenges.

5.3.3 Generative Data-Aware Simulated Annealing

According to the challenges stated in the last part, some new assumptions should be made as follows. The first one is that historical optimisation records become unknown. In that case, the system needs to use the traditional Simulated Annealing algorithms (in Section 4.3.1) to solve problem P1 for the historical optimisation records $\{x_{Jt}, y_{Jt}, R_t, N_{Jt}\}$. Secondly, the data is assumed to be few-shot. And only using these few samples will restrict the classifier's performance, so it might lead to a wrong probabilistic model. To overcome this restriction,

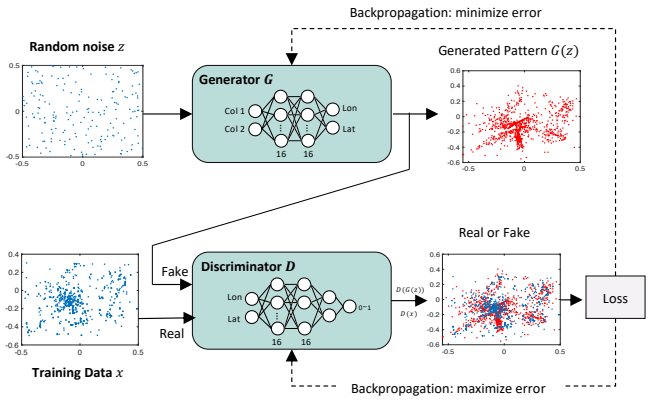


Fig. 5.4 Traffic modelling and generating based on GAN.

Generative Neural Networks (GAN) is denoted as a tool to deal with this few-shot data learning [191]. Also, using GAN for this job requires no manual selection and parameter tuning. In that case, this work employs GAN and the new framework is named generative data-aware Simulated Annealing. GAN is the key function in this section.

Generative Neural Network (GAN)

GAN was designed by I. Goodfellow et. al. [184] in 2014. It trains a generator and a discriminator simultaneously. The discriminator is able to find the difference between the real data and the generated (fake) data, while the generator can finally build a model to re-generate convincing data and make the discriminator regard this is the real data. The trained generator is useful when the data set is limited as it can generate any artificial data to be regarded as real ones. Such a technique already has applications in modelling city-scale 3G spatial traffic in [145]. This work also utilises this advantage of GAN to deal with the condition with few-shot data. One of my works [189] (see Appendix A) has done this GAN-based user coordinates modelling, which achieved training and generating based on few-shot data. Here, for the thesis's coherence, the author directly uses that design as a function. The following parts will briefly introduce the design of GAN, its performance evaluation and comparison are provided in Appendix A.

Traditional GAN is defined for the matrix. In this work, the author adjusts the GAN framework to model the distribution of coordinates (a list) as shown in Fig. 5.4, G is

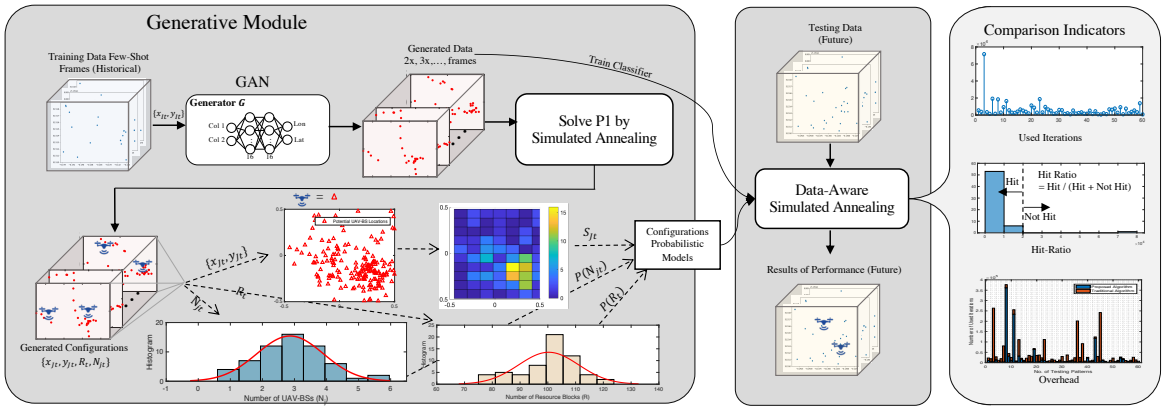


Fig. 5.5 The framework of generative data-aware Simulated Annealing as well as the performance indicators.

responsible for generating regular traffic pattern and D is for detecting the changes. The training process and target function is similar to [184]. Specifically, the training data x is denoted as the historical user coordinates $x = \{x_{It}, y_{It}\}$. Each frame of coordinates is marked by the label of time stamp. For example, the labelled data with 9 AM indicate that all the frames with this label are collected from 9:00 AM to 10:00 AM. Later, all the generated data will also be marked with the same label.

The Framework of Generative Data-Aware Simulated Annealing

This part proposes the framework of the generative data-aware framework in Fig. 5.5. This framework has two modules. The left one is the generative module, and the right one is the proposed data-aware framework in the last Section 4.3.2. It has visualised the inputs and outputs of each inner blocks. From the left side, the Generator G of GAN produces convincing artificial data to overcome the few-shot data problem. In this figure, the historical user distributions are used to train G after which previous few-shot samples can be expanded to any amount (e.g, 2x, 3x...). Thanks to GAN, the classifier has enough labelled data to be trained. Then, the traditional Simulated Annealing algorithm is executed based on the generated user distributions with solving the optimisation problem P1. Afterwards, the configurations are generated and ready for building the probabilistic models. Especially, the regions with high demand (hotspots) shall attract high probability of UAV-BSs' assistance,

Parameter	Value	Parameter	Value
h	200 m [178]	α_1	0.1
a	9.6117 [59]	α_2	0.1
b	0.2782 [59]	α_3	1
$\mu^{\mathbf{LoS}}$	1 [59]	N_{imin}	1
$\mu^{\mathbf{NLoS}}$	20 [59]	N_{imax}	50
f	2 GHz [178]	v_{xmax}	1770 m
c	299792458 m/s	v_{ymax}	1000 m
$P_T(\text{dB})$	20 dBm [192, 66]	N_{jmin}	2
R^{θ}	1	N_{jmax}	7

Table 5.1 Experimental Parameters

so the probabilistic distribution of the best locations for UAV is not uniform as assumed in the reviewed works. For example, in Fig. 5.5, the red triangles are used to describe the uneven density of UAV-BSs' optimal deployment. Next, a histogram is used to describe the density of the best places for deployments. Besides, R and N_j are also modelled by fitting the probabilistic functions.

The proposed data-aware Simulated Annealing is shown on the right side in Fig. 5.5. Real-time off-loading scheme can choose to work with the generative probabilistic models or not. If so, it becomes the *generative* data-aware algorithm. Otherwise, it is switched back to the previous data-aware algorithm with no ability to deal with scarce data. To compare these two frameworks, this work provides three indicators. The first one is comparing the used optimising iterations in gaining similar optimization results. The second is calculating a hit ratio which is the partition of tests whose computation iterations are indeed reduced by using the generative data-aware algorithm. The final indicator is overhead. Some tasks take anomaly large amount of iterations, this research needs to check if the anomaly is caused by misleading probabilistic models or the local optimum.

5.4 Design of Experiments

The experiments are based on realistic mobile user location records measured by GPS on smart phones. This data set is captured from geo-tags of a social networks, Twitter. There

are 0.6 million geotagged Tweets for the Greater London and suburbs area in two weeks. They have exact locations and users' identifications to indicate who used the network and where did it happen. Even though geotagged Tweets do not take the major part in all Tweets and its traffic occupies a small partition of the cellular traffic, the work [1] still has verified the distribution of geo-tagged Tweets is linearly correlated to consumers' and mobile traffic distribution. Therefore, this work uses this real-world geo-tagged dataset as the users' spatial-temporal distributions to check if the proposed frameworks can perform better in solving the Problem P1 and enable the time-efficient offloading of the actual hotspots. This experiment divides the data into two parts, one-week for historical (training) and another week for future (testing). Each data set (training or testing) includes 60 frames of geo-Tweets distribution from 8 AM to 8 PM in five week days (one hour of each frame). This work used MATLAB 2018b to simulate the UAV-BSs offloading, and TensorFlow 1.12.0 for training and testing GAN on Python 3.6.8. All the simulation with the above settings is executed on a MacBook Pro with a 1.4 GHz Intel Core i5 processor and 16 GB 2133 MHz LPDDR3 RAM.

The experimental parameters are listed in Table 5.1. These UAV-BSs are hovering over the hotspots, the hovering height is fixed in this experiment. Majority of the parameters are justified by 3GPP [178] and Al-Hourani's air-to-ground channel model [59].

- h: The hovering height h is fixed to 200 m. According to 3GPP study of aerial vehicles [178], the height of urban UAV-BSs (micro-cell) is suggested to be chosen from $\{50, 100, 200, 300\}$ meters. 200 m is a commonly selected because the UAVs do not need to frequently ascend or descend to avoid buildings [59][70]. Flexible height adjustment will be investigated in future work.

- $a, b, \mu^{\text{LoS}}, \mu^{\text{NLoS}}$: These parameters are the given constant parameters of using Al-Hourani's channel model [59]. The values are gained by fitting the urban air-to-ground channel model.

- f: The UAV-BS carrier frequency f is suggested as 2 GHz by 3GPP Release 15 [178]. Its air-to-ground performance has been verified, so 2 GHz is commonly selected as the carrier frequency.

- $P_T(\text{dB})$: The transmission power $P_T(\text{dB})$ is a fixed value, and it can be selected from $\{10, 20, 27, 30\}$ dBm as reviewed in Table 2.1. 20 dBm is selected because the higher transmission power (27 dBm, 30 dBm) requires larger batteries and bigger UAVs which increase the operational cost, and the lower transmission power (10 dBm) sacrifices the serving coverage. Accordingly, 20 dBm transmission power is used in [192, 66] and also this experiment.

- R^θ: The users require at least one resource block for communication, so R^θ is manually set to be 1 for the basic communication. That represents a 0.18 MHz wide and 0.5 ms long resource block [176]. The number of occupied resource blocks may vary according to the demand of users, larger R^θ represents that the users require more resources to support their basic communication.

- $\alpha_1, \alpha_2, \alpha_3$: These are designed parameters in this work. They are set as an example to compare the performance of the three algorithms. Optimising these three parameters belongs to the topic of multi-target optimisation which will be executed in future work.

- N_{imax}, N_{imin}: The UAV-BS can maximally serve 50 users (N_{imax}) and at least serve 1 user (N_{imin}). The bandwidth of UAV-BS is 10 MHz [178], and each resource block occupies 180 kHz bandwidth, so a UAV-BS can serve 55 users in the ideal condition. And this work sets $N_{\text{imax}} = 50$.

- $v_{\text{xmax}}, v_{\text{ymax}}, N_{\text{jmax}}$: The region sizes $v_{\text{xmax}}, v_{\text{ymax}}$ are set to describe the experimental range. And it is assumed that 7 UAVs are available now, so N_{jmax} is 7. And this is a multi-UAV optimisation, so at least 2 UAV-BSs are used.

In this work, the author has proposed three related methods to solve Problem (5.10). The first Simulated Annealing is the basic method, the data-aware Simulated Annealing is aiming for time-efficiency, and the generative data-aware Simulated Annealing is designed for robustness while facing the few-shot data. In this section, three experiments are proposed for evaluating their performance.

In this table, the following hypotheses have been made,

	Traditional	Data-Aware	Generative Data-Aware
No Data	✓	×	×
Time Efficiency	×	✓	✓
Robust to scarcity	✓	×	✓

Table 5.2 Comparison of the three solutions of Problem (5.10)

- All of these methods can solve Problem (5.10).
- The traditional method solves Problem (5.10) without the support of extra data but sacrificing the time-efficiency.
- The data-aware method can solve Problem (5.10) with time-efficiency relying on the historical data. Its performance is sensitive to the quality of data.
- The generative data-aware method can solve Problem (5.10) and becomes robust to the poor quality of data.

5.4.1 An Example of Using Simulated Annealing Algorithm

The Simulated Annealing is used to solve the problem P1 in a basic way. In this part, a user distribution is given, and the following hypotheses require verification.

- The traditional Simulated Annealing algorithm can solve Problem (5.10).
- This method does not need the help of historical data, so it is robust to the data scarcity.

Fig. 5.6 A-1 to A-4 illustrate the process of heuristically finding the solutions of an example test. In detail, the users' coordinates are shown as blue dots in A-4 in Fig. 5.6. The author denotes red triangles as the UAV-BSs, they are hovering with a fix height over the optimised locations. The process of optimising configurations is plotted as line chart in Fig. 5.6 A-1 to A-3 (horizontal axis is logarithmic scale). Simulated Annealing finished the optimisation in 10^5 iterations. At the beginning, the adjustment fluctuates intensely because the algorithm is searching the optimums. Then, the configuration becomes stable and converge finally. For example in A-2, after trying diverse number of UAV-BSs from 2 to 7, it finally decides

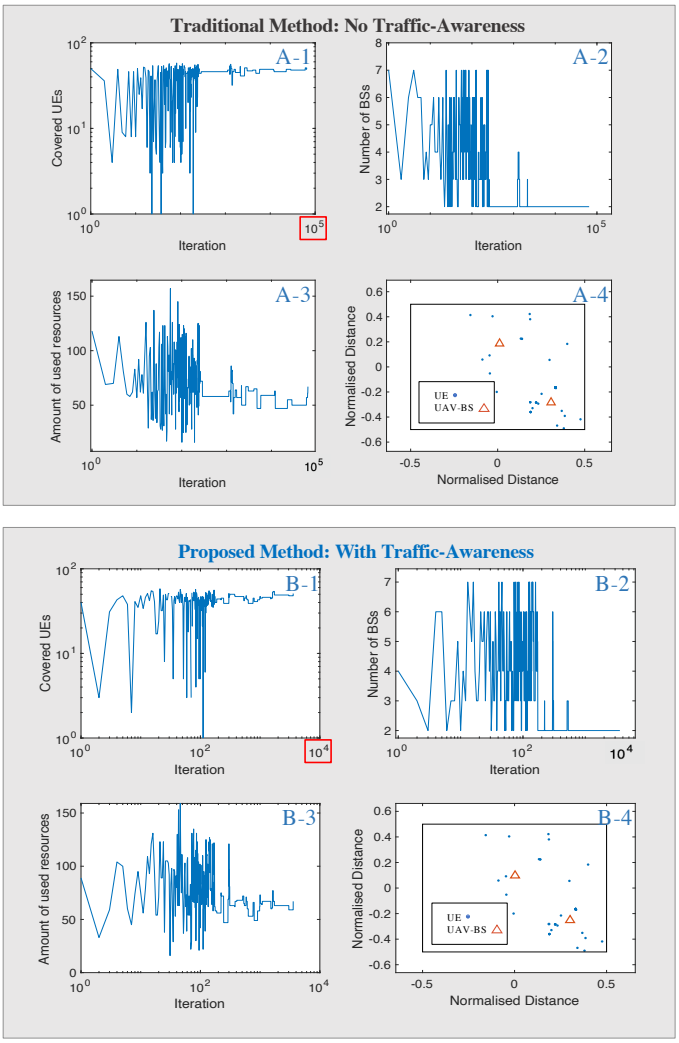


Fig. 5.6 The process of optimising UAV self-deployment and the final deployment. A: Traditional method. B: data-aware method.

a two-drone deployment. Two UAV-BSs are dispatched to around (0, 0) and (-0.3, -0.3) separately. The problem P-1 is sub-optimally solved in 10^5 iterations. Generally, Simulated Annealing is able to solve this multi-target problem. Then, it does not need the help of historical data to solve the problem but needs a large number of iterations.

5.4.2 Results of Data-Aware Simulated Annealing

In this experiment, the data-aware framework in Fig. 5.3 is applied on Simulated Annealing and the real-world dataset. The data-aware module is supplied with current user distribution and historical data. This experiment is designed to verify the following hypotheses.

- The data-aware Simulated Annealing can also solve Problem (5.10).
- Compared with the traditional Simulated Annealing, the new data-aware framework enables a reduction of computation iterations.

Firstly, this framework is applied on the same example user distribution as in previous part. Fig. 5.6 B-1 to B-4 present the process of optimisation. At a first look, the data-aware framework outputs similar result: two UAV-BSs are deployed to around (0, 0) and (-0.3, -0.3). What stands out is the difference of iteration-usage. The used iterations' orders-of-magnitude dramatically decreases from 10^5 to 10^4 through using the data-aware framework. In other words, the time-consumption of Simulated Annealing algorithm will be reduced around 90% due to data-awareness. It is promising, but testing with one example can still lead to an exception. In that case, this work expands the tests.

Fig. 5.7 presents the comparison between the non-data-aware (traditional) and the data-aware methods (proposed) from different aspects. Firstly, the two bar charts (A1 and A2) plot the original results of used iterations of each test in time sequence (from Monday to Friday). Blue bars are used by testing non-data-aware method, and green ones indicate the performance of proposed data-aware method. In a general view, the bars become much shorter by using the proposed method, especially for the the 9th, 17th, and 46th very high blue columns. The data-awareness indicates a dramatic improvement from the first look.

Next, this improvement is expected to be quantified. Here, a new indicator is defined, hit ratio, which describes how many tests can achieve computation reduction to be under a desired threshold (which is set to be 10^4). Higher hit ratio represents more profit by using the proposed method. In the plot B2 of Fig. 5.7, a stem plot of used iterations is given along with the sorted sequence. Lower values stand more forward. The difference is plotted as a

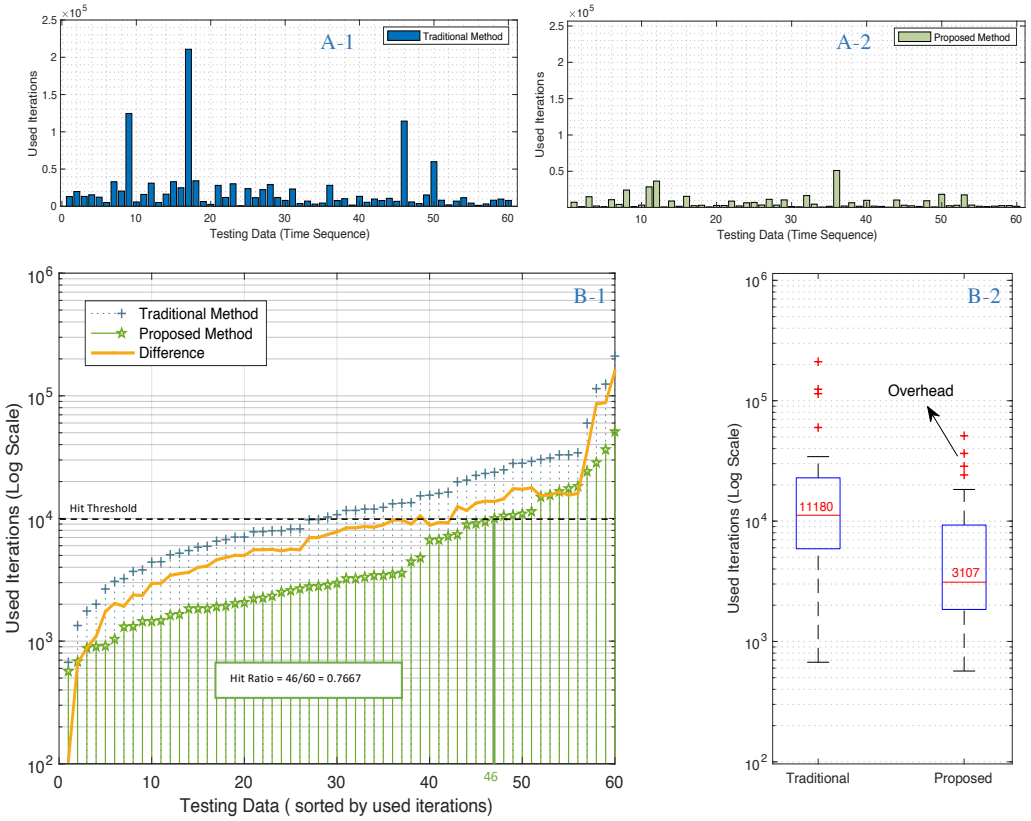


Fig. 5.7 The comparison of non-data-aware and the proposed data-aware Simulated Annealing methods. A1-A2: Bar chart (original result). B1: Sorted stem chart and difference. B2: Box plot.

yellow line chart. It shows the difference is positive and increasing. Each positive difference will be accumulated to the improvement. With setting the hit threshold, there are 46 tests achieving computation reduction to lower than 10^4, so the hit ratio is 76.67% (46/60). By contrast, non-data-aware method only has 28 tests satisfying the threshold (hit ratio 46.67%). In that case, if the optimisation's computation iterations are limited to 10^4, **the data-aware method can finish 30% more work**.

In addition, box plot of Fig. 5.7 B2 also verifies this view, the median decreased (from 11180 to 3107) as well the quartiles and maximum values. Some outliers are marked by red +. These unstable performance can cause overhead of optimisation. Here, two potential reasons are given: the first one is that the historical probabilistic models do not match these testing data (misleading probabilistic model). Another explanation may be the nature

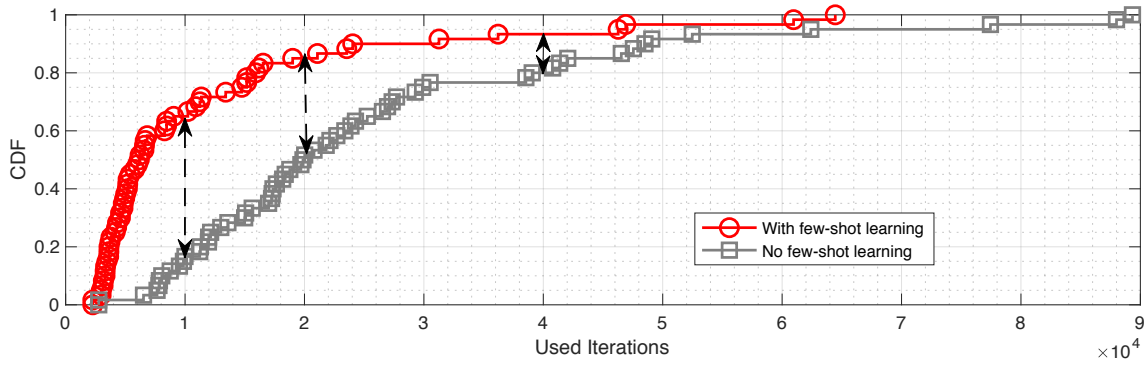


Fig. 5.8 Cumulative Distribution Function (CDF) of traditional data-aware algorithm without few-shot learning and proposed generative data-aware algorithm with few-shot learning.

of heuristic algorithms because optimisation can randomly select better or worse path to approach optimums, which may be caught by local optimums.

5.4.3 Results of Generative Data-Aware Simulated Annealing

In this experiment, the historical data amount is reduced from 60 frames to 15 frames. This setting results in a significant reduction of the historical data, and data scarcity shows up. This experiment aims for the following hypotheses.

- The data-aware methods are sensitive to the data-scarcity.
- Compared with the traditional data-aware method, the proposed generative data-aware method is more robust to the data scarcity.

This part compares the used iterations of each algorithm on the same testing dataset. A Cumulative Distribution Function (CDF) of the used iterations is plotted in Figure 5.8. The grey line marked with 'no few-shot learning' means that this is the result of traditional data-aware algorithm. The other red one is the result of the proposed generative data-aware algorithm. In detail, the average value of the grey line is 25946 (11180 in previous). While applying the 10^4 threshold, $< 20\%$ of tests can achieve this (76.67% in previous). Such a decrease of the computation efficiency is caused by the scarce data, so the traditional data-aware method is indeed sensitive to the data scarcity.

To improve this aspect, the generative data-awareness joins with GAN. After that as shown in Figure 5.8 (red line), the average value of red line drops to 11975 iterations. It represents a 53.85% reduction of computation iterations (compared with 25946). While applying the 10^4 threshold, around 65% of tests can achieve this, which is close to the whole-data condition (76.67%). Indeed, the data scarcity results in unavoidable performance degradation, but the generative data-aware algorithm is tested with the ability to alleviate the difficulties.

5.4.4 Discussion of Overhead

Even though the average time-efficiency is improved, there still exists some tests costing many iterations (e.g., more than 6×10^4 iterations in Figure 5.8). These tests could have optimisation overhead. Here, I give two hypothesis to explain what cause this.

- The generative module may provide mis-leading information.
- The Simulated Annealing owns inherent shortage that may be trapped in local optimum.

To find the answer, Fig. 5.9 gives a comparison of the overhead cases while using the generative module (right figure) or not (left figure). The upper and lower bounds (red lines) are three standard errors which limit the normal range of fluctuations. Then, if some values exceed the red lines, they are regarded as violations which are typical overhead tests, such as the violations (7, 13, 17, 25) in the left figure and the ones (1, 3, 6, 24, 26, 45) in the right figure. As shown in the left figure, when there is no generative module, the minimum violation is 62403. In contrast, this value nearly reaches the maximum value (64450) in the right figure. That is to say the generative module does not provide mis-leading information. In contrast, the generative module reduces the amplitudes of the violations. By contract, the violations still exist because of the nature of heuristic algorithm, caught by local optimum, just like the traditional method. The proposed method can not avoid such nature but still benefit in reducing its negative influence.

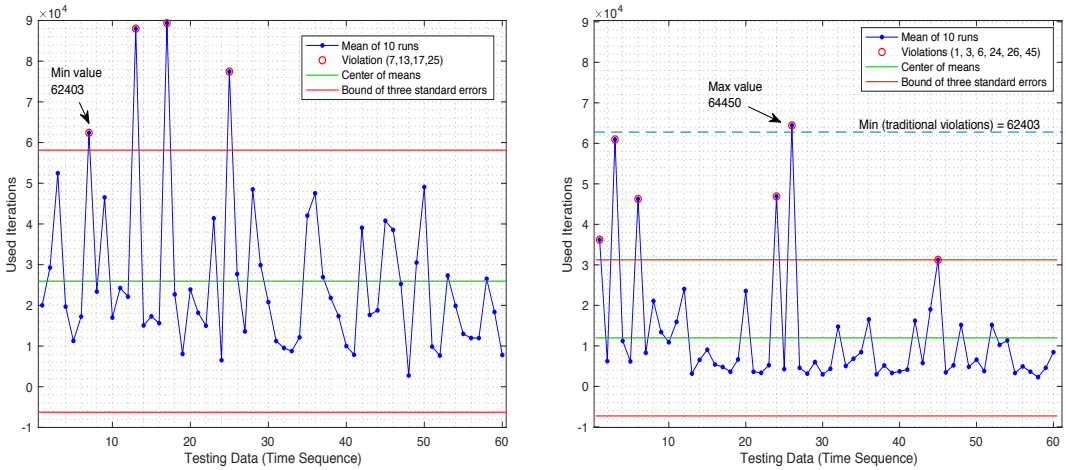


Fig. 5.9 Control graph of the used iterations of data-aware (left) and generative data-aware methods (right).

5.5 Conclusion

The usage of UAV-BS in network off-loading will complement structure in 5G and beyond by satisfying extra communication demand with low cost. It becomes a challenging part for solving the multi-target combinatorial problem. Traditional solutions concentrated on heuristic methods, but it is still a dilemma to decide to gain better performance or cost more computation iterations. This chapter aims to relieve the pressure of this problem by reducing computation iterations while remaining the performance. The author has designed two data-aware frameworks for this purpose. The first one utilised the probabilistic models of historical data to reduce the iterations of Simulated Annealing. The second one addressed the data-scarcity challenge by adding few-shot data learning (GAN). This work also designed experiments on real-world users' locations to test the frameworks. In the results, the data-aware method can finish 30% more work than traditional non-data-aware one. Then, the generative data-aware algorithm improves 53.85% time-efficiency while facing the scarce data.

Chapter 6

Context-Aware Proactive Load Balancing

Overview

The 5G load balancing aims to transfer the extra traffic from a high-load cell to its neighbouring idle cells. In recent literature, controller and machine learning algorithms are applied to assist the self-optimising and proactive schemes in drawing optimisation decisions. However, these algorithms lack the ability of forecasting upcoming anomaly conditions, especially during popular events. This shortage leads to cold-start problems because of reacting to the changes in the heterogeneous dense deployment. Notably, the hotspots corresponding with skew load distribution will result in low convergence speed. This chapter contributes to three aspects to address these problems. Firstly, urban event detection is proposed to forecast the changes in cellular hotspots based on social network data for enabling context-awareness. Secondly, a proactive 5G load balancing strategy is simulated considering the prediction of the skewed-distributed hotspots in urban areas. Finally, this work optimises this CAPO-based load balancing strategy by optimising the best activation time. This work represents one of the first works to couple the real-world urban event detection with proactive load balancing.

6.1 Introduction

Proactive load balancing is one of the optimisation methods, which can automatically pre-configure the cell margins for fast convergence when the environment changes. Nevertheless, traditional schemes have to satisfy not only the regular-time demands but also the peak-hour demands of users, which can cause network energy efficiency reduction [55]. To fill the gap, the CAPO-base load balancing will forecast high-resolution traffic demand and quantify the environmental context to drive decision making. In other words, the ultra-dense heterogeneous networks will have the intelligence to generate and utilise the context, such as traffic patterns and user behaviours, to help the network proactively deciding an energy-efficient optimisation scheme [55]. The context is often challenging to uncover, unfold over time, and it is difficult to collect personal data due to privacy concern. Therefore, public online data, such as online event calendars and social networks, are usually applied for research.

For this purpose, this work mainly **contributes** to three aspects. Firstly, the author uses a 3-stage data-analytic based on Twitter data to design the context-aware module for forecasting the changes in traffic hotspots during events in urban areas. Secondly, the prediction of hotspots is fed to a proactive load balancing strategy to automatically configure cell margins. Thirdly, this work optimises this proactive load balancing strategy through forecasting the earliest activation time with minimising the prediction errors.

The chapter is organised as follows: Section 6.2 introduces the framework of context-aware proactive load balancing for an urban area. Section 6.3 states a 3-stage data-analytics of generating spatial-temporal traffic pattern and executing event hotspots detection through using Twitter data. Then, Section 6.4 proposes a strategy of balancing load based on the production from the 3-stage data-analytics. This section also provides an urban-area simulation example of the context-aware proactive load balancing and its quantification optimisation. Finally, Section 6.5 concludes this chapter.

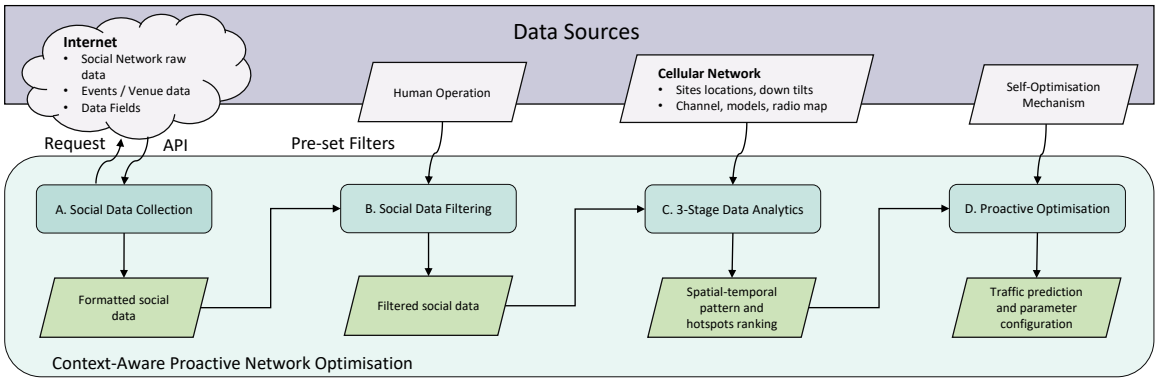


Fig. 6.1 The detailed framework of proposed algorithm.

6.2 The Framework of Context-Aware Proactive 5G Load Balancing for Urban Areas

The framework is visualised in Fig. 6.1 of integrating social-network data analytics into the proactive load balancing for gaining the desired context-awareness. This framework owns two functional blocks, data sources and context-aware proactive network optimisation. The optimisation block is divided into four minor functions: social data collection, social data filtering, 3-stage data analytics, and proactive optimisation. The first two functions (collection and filtering) are common in the related works, such as in the event-based network managements [193] and the context-aware load sharing [194]. The core contributions of this work are the functions of the 3-stage data analytics and proactive optimisation.

6.2.1 Social Data Collection

This function is designed to capture raw Tweets from online platforms and pre-process them to produce a formatted dataset.

1. Capture: The author captures 0.6 million geo-tagged Tweets for the Greater London and surrounding suburbs area for two weeks (time resolution in seconds). For protecting privacy, the users' identifications are not captured. Only time marks, geo-tags, and text are captured for this research. Besides, all the results presented in this work are checked carefully that none of them indicates personal information.

2. Parse: The raw data have many fields which need to be parsed into the required fields. In this work, the author selects the geolocations, time, and Tweets text as the fields to be analysed.

3. Format: The parsed data fields need to be formatted and stored as the proper file that can be fed to data-analytics programs. The Comma-Separated Values (CSV) format is chosen because it can be well supported by Python, Matlab, and Excel.

The formatted data are ready-to-use and contain the whole information in the required fields. To pre-process them for particular research objectives, a filter is required.

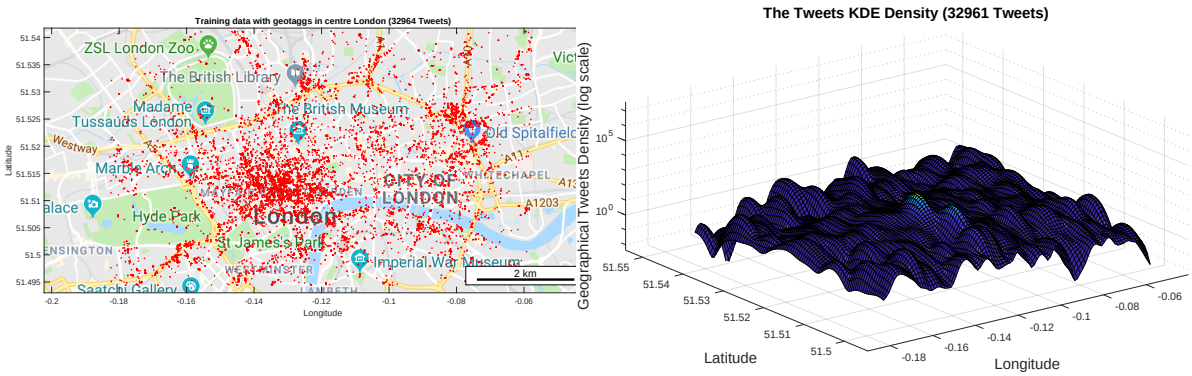
6.2.2 Social Data Filtering

The filter is designed by researchers to reduce irrelevant data from the dataset and further provide a numeric expression of the data in dimensions with interests.

1. Coordinates: This process is to filter the formatted data according to the region of interests. For example, the raw Tweets in this research come from the Greater London and surrounding suburbs, but this work is interested in the urban region. Therefore, the coordinates (bottom left: [51.494417, -0.182733], top right: [51.541160,-0.057710]) are selected to filtered the Tweets in the region for research. The Tweets locations in the first week are visualised on the map in Fig. 6.2a. It shows that the spatial distribution varies with different density.

2. Numeric: A numeric expression is to statistically count the density of data in different dimensions, such as using the histogram to describe the discrete Tweets density on the map. Fig 6.2b provides a continuous density map using Kernel Density Estimation (KDE). The density map shows varying density with ‘peak’ and ‘valley’. In detail, the ‘peak’ indicates a high-traffic region with the ‘valley’ as the boundary.

After generating the numeric expressions, the pre-process is finished. The data are ready to be fed to the 3-stage data-analytics.



(a) Geo-tagged Tweets plotted on map

(b) Tweets density described by KDE

Fig. 6.2 The Tweets locations (15/02/2016-21/02/2016) are plotted on map and their density is described by Kernel Density Estimation (KDE). The map has corner coordinates (bottom left: [51.494417, -0.182733], top right: [51.541160,-0.057710]).

6.2.3 3-Stage Data Analytics

This function analyses the Twitter data by machine learning and statistic methods. Stage 1 and stage 2 build a spatial-temporal traffic pattern. The final stage is detecting the hotspots changes in anomaly (events) based on the modelled traffic pattern. (see Section 5.3 for the detailed procedures)

1. Stage 1 - Spatial Traffic Pattern: As stated in [55], the traffic variations usually followed the regular behaviours of users day by day, so it became the access point for modelling traffic distribution and variation rules which will allow a better prediction of where diverse traffic volumes are requested by users. For example in Fig 6.2b, each ‘peak’ is a high-traffic region with the ‘valley’ as the boundary. It is naturally to partition the region into several traffic-based regions and model the traffic in different Region of Interest (RoI).

2. Stage 2 - Hotspots: In each RoI, the traffic distribution is also skewed. Some spots have high traffic demands, so they are named traffic hotspots. The histogram is an efficient way to find the hotspots in each RoI. However, the distribution of hotspots is not constant. The anomaly conditions also exist, such as during events.

3. Stage 3 -Anomaly detection: This stage is to detect the time and location of the irregularities (anomaly conditions) to provide a new ranking of the hotspots in the RoI with an event happening. The popular public events can influence the spatial traffic pattern with new temporal-spatial hotspots emerging [194] resulting in a dramatic rise of dropped calls [193].

Such forecasting of new hotspots can be an alert that the cells associated with the hotspots require optimisation. The cell margins are estimated to be proactively set according to the predicted cell load.

6.2.4 Proactive Optimisation

This function provides a framework of coupling the predicted hotspots with the network model to forecast network performance following the fuzzy rules of proactive load balancing. An optimisation of the proactive load balancing strategy is provided to estimate the best activation time. (see Section 5.4 for the detailed procedures and an example of implementation)

1. Irregularity Check: Anomaly detection may cause wrong alerts because of scarce data or other reasons. The network requires a mechanism for checking if the network indeed operates in the way of predictions. In that case, the system will continue monitoring the network performance and reserving the choice to switch back to traditional optimisation. When the check is passed, a network model is required to forecast network performance.

2. Network Model: This model simulates the network operation according to the context-awareness. It is needed to quantify the profit and the cost of future optimisation. Proactive decisions are made if the profit is acceptable.

3. Decision Maker: This function aims to associate the hotspots with cellular network optimisation. The cells adjust their margins to prepare for the upcoming traffic.

Finally, such a simulated performance provides a suggestion if the current parameter configuration is beneficial. The profit also depends on the activation time.

4. Activation Time: The proactive optimisation requires to decide the best time to be activated. It is better to start the optimisation earlier, but more errors will occur. Therefore, a design for balancing the trade-off is required.

The description of the framework ends here. The next section starts the detailed explanation of the 3-stage data-analytics for forecasting the events hotspots.

6.3 3-Stage Data-Analytics for Traffic Pattern and Event Hotspots Detection

On the one hand, the traffic pattern includes regular user demands distribution which owns natural convenience for prediction. On the other hand, the anomaly in traffic pattern (e.g., traffic burst during popular events) also exists so that the system needs to alert the changes in advance. This section proposes the process of generating traffic pattern and the forecasting of hotspots' changes.

6.3.1 Stage 1: Spatial Traffic Pattern

The RoIs are used to divide the urban area into small regions. Each region has a high-density 'peak' which represents the aggregation of demands in the urban area. To configure the RoIs, the author selects an unsupervised learning algorithm, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [187]. This clustering algorithm groups together the data at 'peaks' (with many nearby neighbours) and marks the data in 'valley' (with nearest neighbours far away) as noise.

The above data analytic is tested with the London Twitter data. This work denotes the 'peaks' as the areas with Tweets densities that are larger than the average density. The map is approximately a rectangle with ignoring the Earth curvature. In that case, the average density

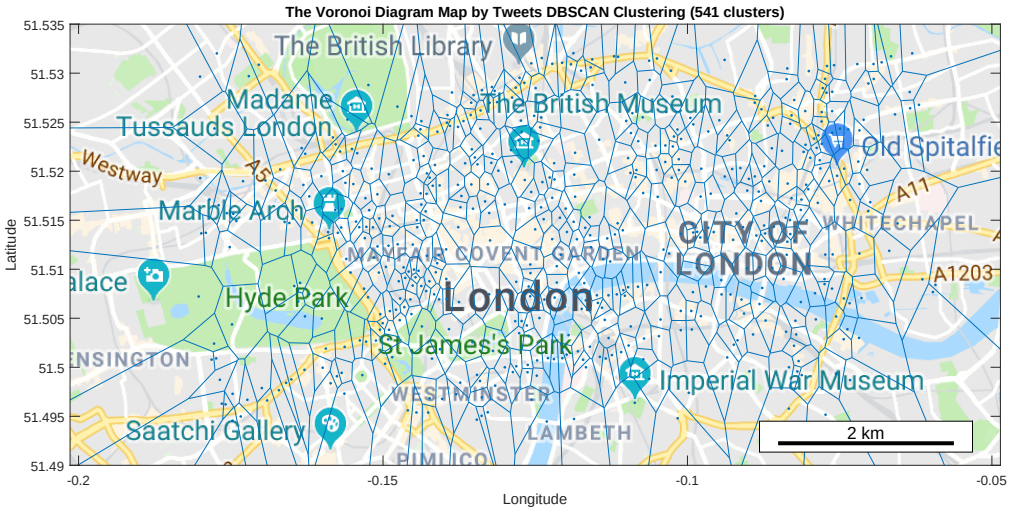


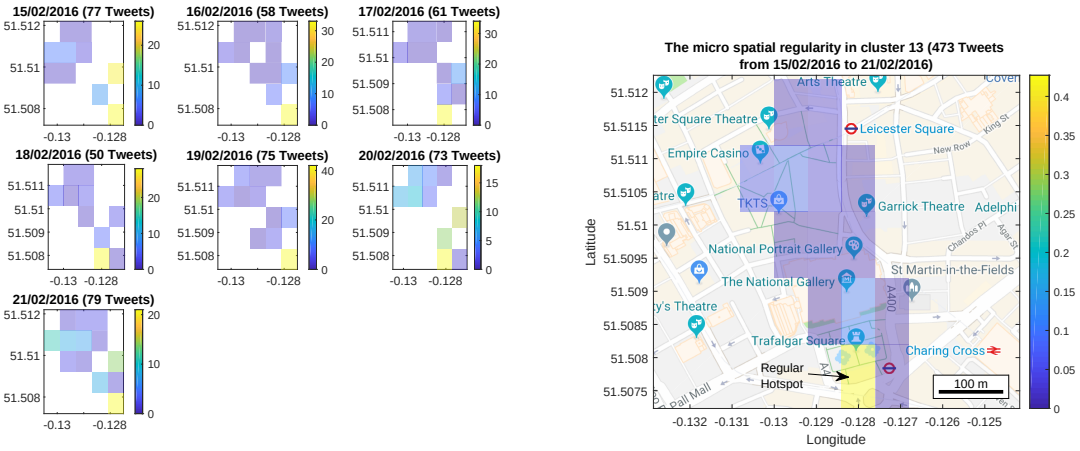
Fig. 6.3 The Tweets density-based clusters on map. Corners coordinates (bottom left: [51.494417, -0.182733], top right: [51.541160,-0.057710])

is the total number of Tweets over the map area, so the minPoints $= 5$ and $\varepsilon = 4.90 \times 10^{-4}$. Under this circumstance, the result of DBSCAN is shown in Fig 6.3. It divided the urban area into 541 clusters (RoIs) according to Tweets density. The dense cells usually located at commercial areas and tourist attractions, such as Westminster and the British Museum.

6.3.2 Stage 2: Hotspots

In each RoI, the Tweets have a skewed distribution which will generate some hotspots. The histogram counts Tweets in the pixels of each RoI and products a heatmap with highlighting the hotspots. For example, in the RoI consisting of the Leicester Square and Trafalgar Square (RoI 13 in the spatial traffic pattern as shown in Fig 6.4), the pixel of Trafalgar Square (yellow hotspot) is more crowded than its neighbours. The Fig. 6.4a displays the daily from 15/02/2016 to 21/02/2016. It shows that the hotspots are commonly distributed around a coordinate (-0.128, 51.508) in the seven days, and the hotspots point to Trafalgar Square. This phenomenon becomes more evident in the weekly hotspot pattern (Fig. 6.4b). In consequence, the pixel becomes hotspot if it is regularly hot in an extended period.

However, the hotspots will change along with the occurrence of social events because the users will aggregate at a new location. The network self-optimisation requires a long time



(a) The histogram of Tweets in one week (daily pattern). (b) The hotspots in one week (weekly pattern).

Fig. 6.4 The usual aggregation of users in each day generates some hotspots that users like to stay and use the network. For example, in the region of Trafalgar Square and Leicester Square, the first one attracts more people. It is indicated by the higher number of Tweets in the histogram.

to converge for balancing the changed distribution of load. To alleviate this problem, the event detection (or the network irregularity/anomaly detection) is required to pre-configure the network before the changes of the hotspots.

6.3.3 Stage 3: Network Anomaly Detection

There are two main steps of anomaly detection:

- Model the regularity (trend and seasonality) of the training data set.
- Detect the anomaly by finding the outliers in the modelled regularity.

In this work, the temporal traffic is used to model the regularity and detect the outliers. Fig. 6.5 illustrates the process. The detailed steps are shown as follow:

1. Firstly, the spatial traffic pattern has used the DBSCAN to divide the region into several RoIs. This part denotes the number of Tweets $\mathcal{N} \subset \mathbb{R}$ in the region $\mathcal{K} \subset \mathbb{R}^2$, then let $o \in \mathcal{K}$ be the clusters (RoI), so the number of Tweets in a RoI o in different time intervals $t \subset \mathbb{Z}+$ is n_{ot} (Tweets per hour).

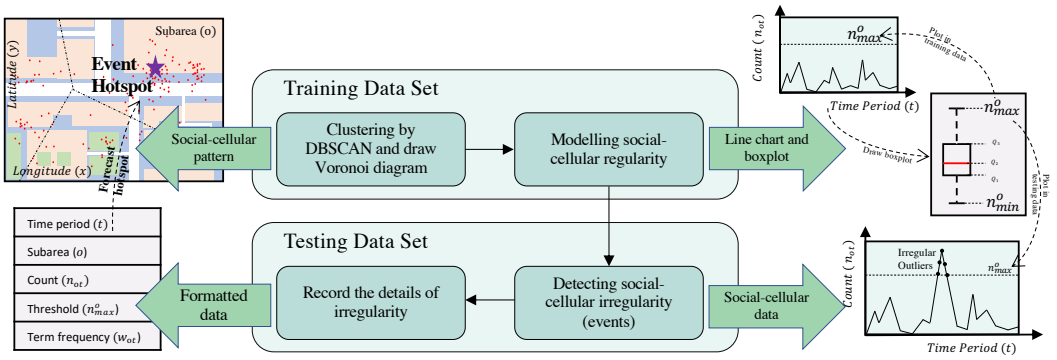


Fig. 6.5 The process of building regularity and detecting network anomaly (irregularity).

2. Then, for each RoI, a line chart of temporally changing traffic and its corresponding box-plot can be generated. Each box-plot offers a box of the majority and a maximum threshold (n^o_{max}). Such a threshold describes the regular traffic range that regular social network traffic is lower than it.

3. Finally, in the testing data set, if current Tweets per hour (n_{ot}) grows higher than the threshold (n^o_{max}). This hour will be regarded as the start of an event in this RoI. Then the algorithm automatically highlights the irregular outliers and records the details of these outliers, including the time period t, subarea (RoI) (o), count (n_{ot}), threshold (n^o_{max}), and term frequency (w_{ot}). The term frequency refers to the five most-appeared key works in Tweets.

The number of Tweets n_{ot} is an indicator of the network traffic. As indicated in [1], the estimated Down-Link (DL) traffic load $\hat{r}_{DL}$ in cluster o in time interval t can be described as

$$\hat{r}^{ot}_{DL} = 10^{b_{DL}}(\frac{n_{ot}}{\tau})^{a_{DL}}\tau \tag{6.1}$$

where $[a_{\mathrm{DL}} = 0.88\mathrm{kb/Tweet} \quad b_{\mathrm{DL}} = 2.37\mathrm{kb/s}]$ and τ is ratio between time interval and one second (e.g, in this work $\tau = 3600\mathrm{s/hour}$). Note that, $\hat{r}_{DL}$ is service-neutral that it represents the generated traffic from all services. According to this correlation, the burst of Tweets' traffic during an event represents an irregular increase of network traffic. In this chapter, n_{ot} is selected as the indicator of network traffic changes and the occurrence of events.

Event Topic	**Day**	**Start Time**	**End Time**	**Location**	**Cluster**	**Cluster centroid**	**Event Published**
Grimsby World Premiere	22/02/2016	17:00	21:00	Odeon Leicester Square, London	13	(51.50872, -0.12812)	18/02/2016
Curtain Up Exhibition	09/02-31/08/2016	10:00	17:45	Victoria and Victoria and Museum	128	(51.49669, -0.17228)	09/02/2016
Craft Beer Rising 2016	26-27/02/2016	12:00	00:30	Old Truman Brewery in Brick Lane	12	(51.52098, -0.07195)	23/02/2016
Stop Trident Demo	27/02/2016	12:00	-	Trafalgar Square, London	13	(51.50872, -0.12812)	08/12/2016

Table 6.1 The details of events form the online calendars. This work allocates the nearest cluster to each event location according to the distance to each centroid.

Grimsby World Premiere

Day	Hour	Cluster	Tweets/hour	Threshold	Top 5 terms, e.g., [('term 1', frequency), ('term 2', frequency). . .]
22	18	13	9	7	[('Square', 4), ('Leicester', 3), ('#Grimsby', 2), ('premiere', 2), ('#GrimsbyWorldPremiere', 2)]
22	19	13	16	7	[('Square', 7), ('Odeon', 7), ('Leicester', 6), ('de', 2), ('amp', 2)]
22	20	13	8	7	[('Leicester', 3), ('London', 3), ('#elvispresley', 2), ('Square', 2), ('Picadilly', 2)]
22	21	13	14	7	[('#London', 10), ('#GrimsbyWorldPremiere', 9), ('#redcarpet', 8), ('#igerspinoy', 5), ('#litratongpinoy', 5)]
22	23	13	13	7	[('#London', 10), ('#GrimsbyWorldPremiere', 10), ('#redcarpet', 10), ('#igerslondon', 8), ('#igerspinoy', 7)]

Curtain Up Exhibition

Day	Hour	Cluster	Tweets/hour	Threshold	Top 5 terms [('term 1', frequency), ('term 2', frequency). . .]
25	21	128	14	7	[('CURTAIN', 11), ('UP', 11), ('OF', 11), ('exhibited', 10), ('CELEBRATING', 9)]
25	22	128	12	7	[('Victoria', 20), ('Albert', 20), ('Museum', 20), ('U', 10), ('K', 10)]
27	16	128	10	7	[('Victoria', 9), ('Albert', 8), ('Museum', 8), ('museo', 2), ('manzana', 2)]

Craft Beer Rising 2016

Day	Hour	Cluster	Tweets/hour	Threshold	Top 5 terms [('term 1', frequency), ('term 2', frequency). . .]
26	13	12	17	14	[('Drinking', 8), ('Craft', 8), ('Beer', 8), ('Rising', 8), ('#photo', 4)]
26	14	12	18	14	[('Drinking', 11), ('Craft', 9), ('Beer', 8), ('Rising', 8), ('@trumanbrewery', 4)]
26	15	12	16	14	[('Drinking', 12), ('Craft', 9), ('Beer', 9), ('Rising', 9), ('Ale', 4)]
26	16	12	18	14	[('Drinking', 12), ('Beer', 12), ('Craft', 10), ('Rising', 10), ('@trumanbrewery', 3)]
26	18	12	17	14	[('Craft', 5), ('Beer', 5), ('Rising', 5), ('#photo', 4), ('Drinking', 4)]
26	19	12	20	14	[('Drinking', 10), ('@trumanbrewery', 8), ('Rising', 6), ('Craft', 5), ('Beer', 5)]
26	22	12	20	14	[('Drinking', 14), ('Craft', 12), ('Beer', 12), ('Rising', 12), ('#photo', 12)]
26	23	12	18	14	[('Drinking', 10), ('London', 7), ('Beer', 7), ('Craft', 6), ('Rising', 6)]
27	12	12	25	14	[('Drinking', 16), ('Beer', 15), ('Craft', 14), ('Rising', 14), ('London', 9)]
27	13	12	24	14	[('Beer', 18), ('Craft', 17), ('Rising', 16), ('Drinking', 14), ('London', 6)]
27	14	12	18	14	[('Drinking', 17), ('Craft', 11), ('Beer', 11), ('Rising', 11), ('@trumanbrewery', 5)]

Stop Trident Demo

Day	Hour	Cluster	Tweets/hour	Threshold	Top 5 terms [('term 1', frequency), ('term 2', frequency). . .]
27	14	13	8	7	[('Square', 4), ('National', 3), ('Gallery', 3), ('Trafalgar', 3), ('3', 2)]
27	15	13	9	7	[('Trafalgar', 5), ('Square', 5), ('London', 3), ('#stoptrident', 2), ("I'm", 1)]
27	16	13	14	7	[('Square', 6), ('Trafalgar', 4), ('think', 3), ("I'm", 2), ('London', 2)]

Table 6.2 The results of event (irregularity) detection

This work collected the advertisements of four events in London from the online event calendar as shown in Table 6.1. The information from online event calendars has been published at least four days before the events happen. This work match the published event location with the spatial traffic pattern to get which RoI the event belongs. Then, the 3-stage data-analytics method is applied to the London dataset to detect the network anomaly.

The results of anomaly detection are shown in Table 6.2. It displayed the irregular time when the Tweets per hour were higher than the cluster threshold. For example, the first event 'Grimsby World Premiere' caused network traffic irregularity from 18:00 until 23:00, and the traffic peaks arrived at 19:00 (16 Tweets/hour) and 21:00 (14 Tweets/hour). The third event 'Craft Beer Rising' event attracted high traffic demand from 13:00 to 23:00 on 26/02/2016.

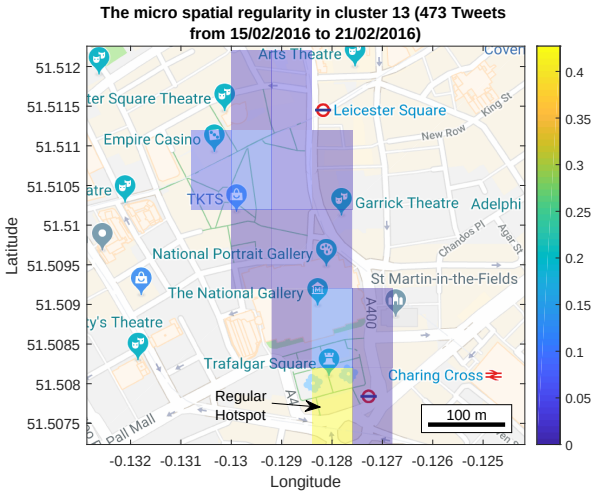
The author finds that the start time of Tweets irregularity does not precisely match the start time on the event calendar. It is because users have different arriving time and network usage behaviours. As the number of Tweets is positively correlated to cellular traffic, the Tweets irregularity becomes the alarm for upcoming high traffic demands. Furthermore, the detected most-frequently used words match the topic keywords in the event calendar.

According to the results of detection, this work can forecast the changes of hotspots by highlighting the event location as the new high-traffic region. For example, Fig. 6.6 (a) and (c) present the hotspots of regularity and on the 'irregular' day of the first event 'Grimsby World Premiere'. The traffic ranking of the pixels has changed and the hottest pixel altered from the Trafalgar Square to the Leicester Square. In this algorithm, once the outlier is detected, the event-detection algorithm automatically allocates the event pixel to be the first position in the hotspots ranking (as visualised in Fig. 6.6 (b)). This forecasting pattern alerts the network of upcoming high traffic demand at least one hour ahead. Besides, the traffic pattern changed back to regular conditions after the end of the 'World Premiere' (Fig. 6.6 (d)).

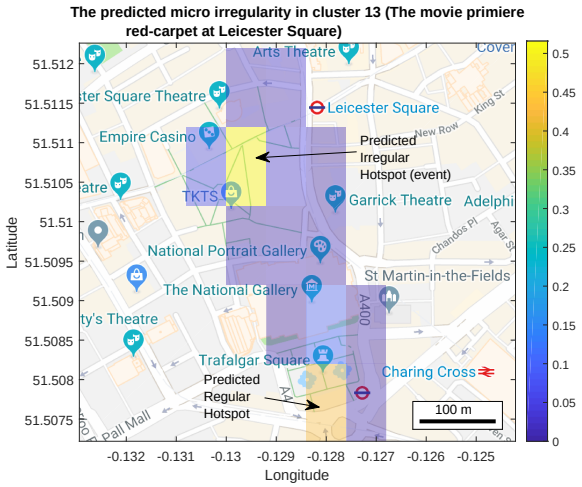
The 3-stage data-analytics also determine the relationship between cells and hotspots according to the hotspots locations and ranks. For example, the hotspots can be in the high-load cell or on the cell edge. This work needs the fuzzy rules to decide the strategy for each condition. Therefore, the author summarises the corresponding relations as follows:

- The cell is predicted to be high-load with event hotspot. (output 1)
- The cell is predicted to be the nearest neighbouring cell to the event hotspot. (output 1)
- The cell is predicted to be the neighbouring cell to the event hotspot but not the nearest one. (output 0)
- Other conditions. (output -1)

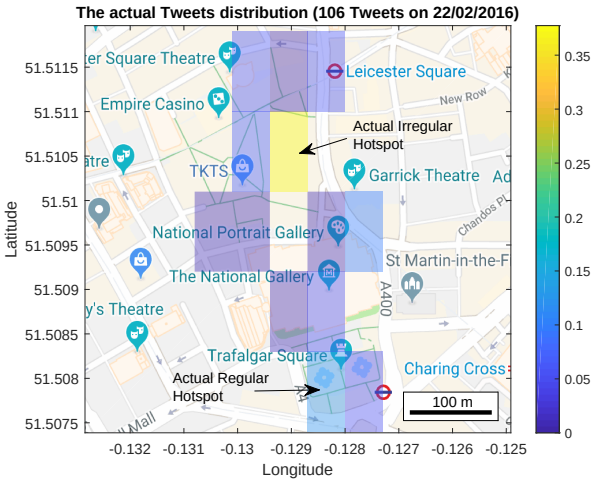
In the first and second conditions, the event hotspots influence the cells most so that they are the cells to be optimised (output 1). The output 0 indicates that the cells are still close to the event hotspot but not directly involved. Moreover, the output 0 denotes that the cells



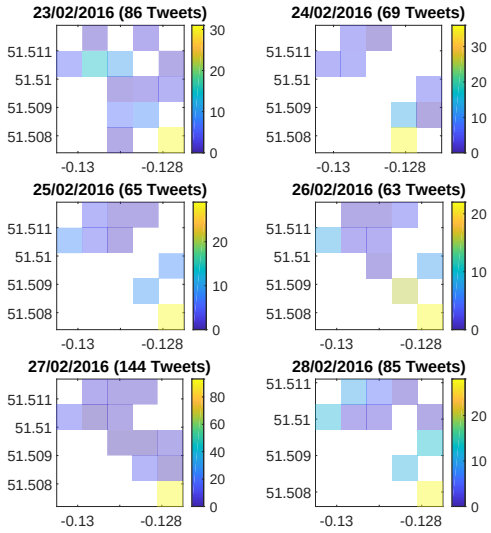
(a) The hotspots in the training data (15/02-21/02).



(b) Predicted hotspots locations and ranking before event.



(c) The actual hotspots on the event day (22/02/2016).



(d) The hotspots recover to regularity after the event (23/02-28/02).

Fig. 6.6 The actual and predicted traffic patterns before and after the event. The Tweets distribution in cluster 13 is transferred into a histogram that describes the Tweets occurrence in each pixel. The pixel with more Tweets is regarded as a hotspot. The regular hotspot is usually attractive for users. In contrast, the irregular hotspot brings a sudden burst of Tweets and disappears after the event.

become far from the event hotspot. The next section will provide the strategy of associating proactive optimisation focusing on the predicted hotspots.

6.4 Proactive Optimisation with Context-Awareness: Load Balancing Use Case

6.4.1 Optimisation Framework

The framework in Fig. 6.7 optimises network configurations based on the forecasting of hotspots from the 3-stage data-analytics. It consists of four major functions: irregularity check, network model, decision maker, and activation time. The following parts will introduce them in detail.

Irregularity Check

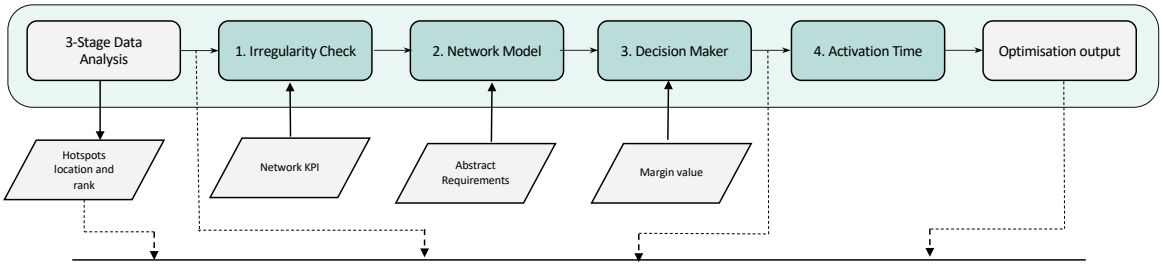
The irregularity check function takes the responsibility of checking and avoiding the following conditions:

- Errors of events detection.
- The network irregularity lasts for a short duration.
- The detected events do not come with the expected high traffic increase.

In that case, it needs two checking items. Firstly, the web-keywords check will compare the term frequency with the topic keywords on the event calendar. Secondly, for the involved BS itself, the cell traffic should be estimated and justified that the cell indeed has a load increase caused by the events. Accordingly, this function aims to minimise the probability of wrong alarms.

Network Model

The network model is required to forecast the network performance according to the forecasting of network context. As the context comes from real-world data-analytics, the network



The Relationships between Cell and Hotspots	Analyser Output	Decision Maker Output	Optimisation Detail (example)
If the cell is predicted to be with hotspot in or the nearest neighbouring cell	1	Apply new margin value to handover some active UEs between the two BSs	The high-load cell connects UE if >150 kb/s, handoff UE if <100 kb/s
If not, the cells are neighbours of a high-load cell	0	No variation is applied to current margin	No variation
Otherwise	-1	Cancel current margin value and return to original	Connect if > 50kb/s

Fig. 6.7 The framework of proactively making decisions of load balancing based on the urban-area anomaly detection and forecasting the hotspots' changes.

model should satisfy the abstract requirements of the actual network settings. For example, the filtered area in London is an urban area, so the small cell density (dense deployment) is 80 cells/km^2 [55]. There are several requirements to abstract the network model from the dataset:

- Urban scenario: Different scenarios decide the parameter setting of networks, such as the BS density and frequency band. The dataset in this work is from London, an urban area, so this work considers an ultra-dense heterogeneous network model (downlink) in [55] associated with orthogonal frequency division multiple access (OFDMA). The network consists J BSs and θ subcarriers, $j \in J$ $\theta_i \in \theta$, and the region served by all BSs is $\mathscr{K} \subset \mathbb{R}^2$, the cell area $o \in \mathscr{K}$.

- Time scale: The time scale varies in different scenarios from seconds to hours, so it is required to determine a proper scale to suit both 3-stage data-analytics and network optimisation. In the simulation, there does not exist an exact time scale to describe the fluctuation of traffic, so a discrete time normalisation is used. In this study, the time is normalised to integer units, and this work performs static simulation of network performance at different time snapshots $t \in \mathbb{Z}^+$ and compare the performance between the conditions with and without context-awareness. Therefore, the transmit power and channel gain of BS j to UE i in o using channel θ_i at time t are $P_j^{\theta_i}(o,t)$ and $g_j^{\theta_i}(o,t)$.

- Channel model: Such a mathematical representation describes the effect that wireless signals propagate through a wireless channel. In this study, a direct propagation path from BS to UE and a ray path reflected from the earth are considered as the channels. The author selects a two-ray model for this job because it considers both direct and reflection channels, and it owns computation simplicity. The channel gain is denoted as $g_j^{\theta_i}(o,t)$. MATLAB is equipped with the two-ray channel for simulation [195]. According to the settings by ITU [196], the height of BS transmitters is set as 20 m and the height of UE receivers is 1.5 m.

- UE association: The load balancing requires to change the UE association with different BSs for transferring the extra load to idle cells. In this work, the author brings channel association variable $\rho_j^{\theta_i}(o,t)$ and BS association variable $u_j^{\theta_i}(o,t)$ in the model, if the value is 1 which means the UE is associated with channel θ_i or BS j at time t. The power of noise is $\sigma^2(o,t)$.

Based on the above parameter settings, the author also denotes $P_{\bar{j}}^{\theta_i}(q,t)g_{\bar{j}}^{\theta_i}(q,t)$ as the interference from other BSs, so the signal-to-interference-plus-noise ratio (SINR) is:

$$\gamma_j^{\theta_i}(o,t) = \frac{P_j^{\theta_i}(o,t)g_j^{\theta_i}(o,t)\rho_j^{\theta_i}(o,t)}{\sum\limits_{j\in\mathscr{J},\bar{j}\neq j}\int_{\mathscr{K}} P_{\bar{j}}^{\theta_i}(q,t)g_{\bar{j}}^{\theta_i}(q,t)dq + \sigma^2(o,t)} \tag{6.2}$$

Then, according to Shannon's theory, the transmit rate on subscriber θ_i is $R_j^{\theta_i}(o,t) = B\log_2(1+\gamma_j^{\theta_i}(o,t))$, where B is the channel bandwidth. And the average data rate for all the UEs in BS j is

$$\bar{R}_j(o,t) = \frac{\sum\limits_{\theta_i=1}^{\theta} R_j^{\theta_i}(o,t)u_j^{\theta_i}(o,t)}{\sum\limits_{\theta_i=1}^{\theta} u_j^{\theta_i}(o,t)}, \forall j,o,t \tag{6.3}$$

The goal of load balancing is to handoff extra UEs (with worse data rate) to their nearest neighbour BS. In other words, this process is changing current BS association variable

$u_j^{\theta_i}(o,t) = 1 \rightarrow 0$ and the nearest neighbour BS association $u_j^{\theta_i}(q,t) = 0 \rightarrow 1$. In that case, the $u_j^{\theta_i}(o,t)$ value determination is the next aim.

As the edge UEs naturally have higher interference (lower SINR) and occupy channels, they are the objects to be unloaded in the predicted high-load cell (BS j). To do this, this work considers setting SINR margin $Margin(j,\bar{j})$ to preserve the UEs in BS j with better SINR when $\gamma_j^{\theta_i}(o,t) > Margin(j,\bar{j})$, and handoff the UEs with worse performance to the nearest neighbour BS $\bar{j}$ if $\gamma_j^{\theta_i}(o,t) < Margin(\bar{j},j)$. Moreover, the margins should be subjected to $Margin(j,\bar{j}) > Margin(\bar{j},j)$ for avoiding ping-pong effect.

The above network model can simulate the network performance but require to decide the strategy of load balancing according to the context-awareness. The next part introduces this process.

Decision Maker

This function follows fuzzy rules which are also used in [194]. According to the 3-stage data-analytics, the decision maker will decide the load balancing strategy (see Fig. 6.7 for the table of fuzzy rules):

- 1: Apply new margin value to handover some active UEs from the high-load cell to the idle cell nearest to the event hotspot.
- 0: No variation will be applied to the current margin.
- -1: Cancel the current margin.

The simulation based on the above strategy results in the average UE data rate $\bar{R}_i(x,t)$ with and without load balancing, and visualises the advantages when proactively trigger the optimisation (at t_1). Moreover, the passive load balancing starts at t_2, $t_1 < t_2$ because the proactive algorithm predicts the peak of traffic while the passive algorithm reacts to it. Therefore, it is important to optimise the strategy to determine an activation time t_1 with maximum profit.

Table 6.3 Simulation Parameters

Parameter Name	Value
Simulation Environment	Urban Microcell [196]
BS Transmit Power	41 dBm [196]
Frequency Carrier	5 GHz [28]
Channel Bandwidth	5 MHz [196]
Thermal Noise Level	-174 dBm/Hz [196]
UE Antenna Gain	0 dBi [196]
Small Cell Antenna Gain	17 dBi [196]
Small Cell Density Medium Deployment	40 cells/km^2 [28]
UE Density Hotspot High Load	12000 UEs/km^2 [28]
UE Density Hotspot Medium Load	10000 UEs/km^2 [28]
UE Density Hotspot Low Load	6000 UEs/km^2 [28]
UE Density Non-Hotspot	800 UEs/km^2 [28]
Event Hotspot Area Radius	100 m
Event Hotspot UE Number	18(10% Low) - 377 (High)
Regular Hotspot Area Radius	50 m
Regular Hotspot UE Number	78 (Medium Load)
Scattered UE Number	157 (Non-Hotspot)
UE Start/End Increasing Time	10-70
Time of Proactive Optimization Starts*	30
Time of Passive Optimization Starts*	60
Handoff Threshold*	50kb/s

Activation Time

The activation time of proactive optimisation influences the profit improvement because the forecasting will have more errors if the optimisation needs to be activated earlier. However, if the network is not configured earlier for the upcoming traffic changes, it will need some time to recover and have a poor performance period like in [194]. To balance the trade-off between the prediction errors and the poor performance period, a mechanism needs to be designed for particular scenarios. This chapter provides an example in the next part.

6.4.2 An Example of Proactive Load Balancing in the London Urban Scenario

This experiment follows the suggested micro-cell urban test environment settings of ITU-R [196] (Radiocommunication Sector of International Telecommunication Union). ITU-R aims for a rational, equitable, efficient and economical use of the radio-frequency spectrum, so the BS transmit power (41 dBm), channel bandwidth (5 MHz), noise (-174 dBm/Hz), and antenna

gain (17 dBi) in this work are all set accordingly. The detailed values of communication environment setting are shown in Table 6.3.

Besides, this work also simulates an urban environment with event-related hotspots emerging and disappearing. The characteristics of users' and cells' density follow Bassoy's 5G load-balancing model [28], in which Small Cell Density Medium Deployment has 40 cells/km^2 and UE Density Hotspot High Load has 12000 UEs/km^2. The carrier frequency is selected as 5 GHz for higher capacity. The detailed parameters are listed in Table 6.3.

Based on the reference parameters, this work also designs special parameters for this simulation. For example, the event hotspot area, the event's start and end time, and the minimum data rate requirement. They are listed and explained as follows.

- There is a 500 $m \times 500$ m square covered by $\mathscr{B}$ cells generated through Poisson Point Process. This work assumes that one event happens in a cell (which will have a high load in future) located close to the centre of the area.

- Event Hotspot Area Radius: The event will bring a growing hotspot with the number of users increasing from t_1 (10% low load) to t_2 (high load) as a Sigmoid function in an area with a 100 m radius.

- Event Hotspot UE number: the number of UEs in the event hotspot at $t_1 = 10$ is 18 ($10\% \times 6000\,\text{UE}/\text{km}^2 \times 0.01\,\text{km}^2 \times \pi$), and at $t_2 = 70$ it grows to 377 ($12000\,\text{UE}/\text{km}^2 \times 0.01\,\text{km}^2 \times \pi$). The event hotspot is generated according to a normal distribution with a mean value $\mu = 0.01$ and a standard deviation $\sigma = 0.2$.

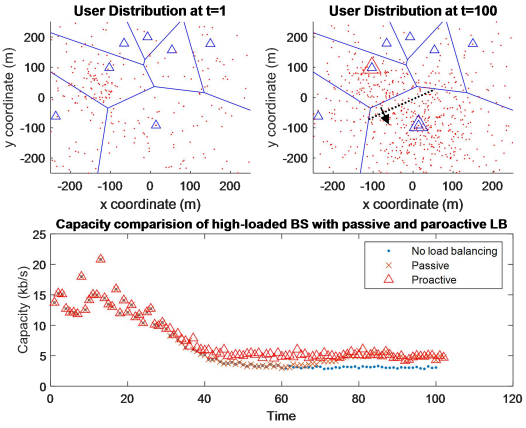
- Regular Hotspot Area and UE: There is also a regular hotspot (e.g., commercial area) close to the event hotspot. This hotspot has a 50 m radius and a medium load of UE density, so it has 78 UEs generated from another normal distribution ($\mu = 0.01, \sigma = 0.1$).

- UE Start/End Increasing Time: In the time line of this experiment, the number of UEs starts increasing at $t_1 = 10$ and ends at $t_1 = 70$. Proactive optimisation starts earlier than the passive optimisation, so the time points are set as 30 and 60 separately.

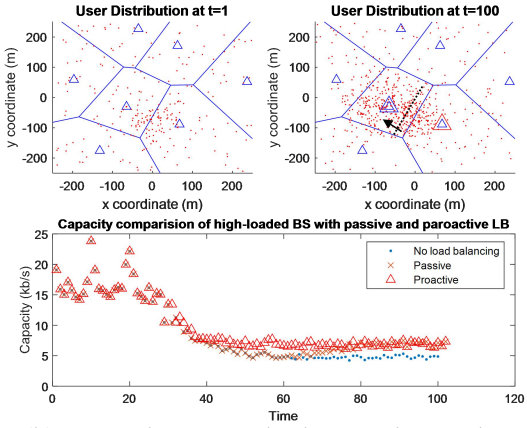
- Handoff Threshold: According to Chen's estimation of the minimum data rate requirements of different services [180], the minimum data rate for Web browsing is 30 kb/s. This work sets the handoff threshold as 50 kb/s to ensure the basic Web browsing.

This work performs a Monte Carlo simulation through repeating random sampling for 100 times. All users are assumed to have equal demands, and the channel allocation can satisfy them equally with allocating proper resource blocks. In that case, when the BSs own equivalent resource, the associated users with fewer competitors will experience owning more resources and better networks. Moreover, both the context-aware and traditional load balancing schemes use the same controllers to ensure that the convergence ability is the same.

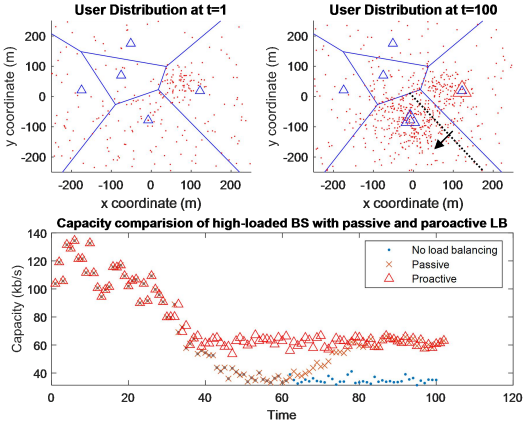
The Fig. 6.8 (a)-(c) proposed the cell layout (blue triangles are BSs) with user distributions (red dots) at the beginning time ($t = 1$) as well as the end time ($t = 100$). The cell margins are visualised through using the Voronoi diagram. They shrank in the event cells (nested blue triangles) and moved as shown as dash lines. The corresponding network performances are displayed (in the bottom line charts separately) as average capacity (data rate) changing along with time. For example in 6.8 (c), the network performance drops to around 40 $\mathrm{kb/s}$ at $t = 50$, and it stays in the condition with poor performance if there is no load balancing algorithm (blue dash line). The yellow line indicates the condition with load balancing, but it is reactive to the event (cold-start with no prior knowledge about traffic increase). This work denotes it as a passive (without context-awareness) load balancing because they are not able to absorb the burst of traffic. It starts at $t = 60$ and finally converges to around 65 $\mathrm{kb/s}$ but causes a poor-performance period from $t = 38$ to $t = 80$. That provides an evaluation of the negative effect of the cold-start problem. In contrast, with forecasting upcoming events traffic, the proactive load balancing (red triangle line) experience almost no poor-performance as it starts at $t = 30$ (before the traffic peak arrives). After the Monte Carlo simulation, Fig. 6.8 (d) displays the average capacity. The network indeed experiences low capacity from $t = 35$ to $t = 80$, which is combined with the time of reaction and the time of convergence. Such a cold start problem is not avoidable unless it is benefited by context-awareness. In the above simulation, the activation time of the proactive optimisation is assumed according to



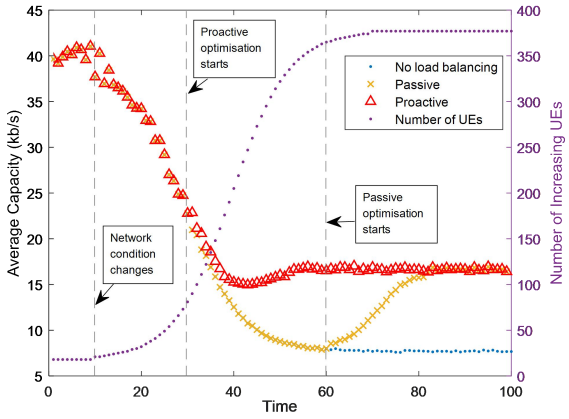
(a) Example 1: Regular and event hotspots are in the same cell.



(b) Example 2: Regular hotspot is on edge.



(c) Example 3: Regular hotspot is in the neighbour cell.



(d) Capacity comparison after the Monte Carlo simulation.

Fig. 6.8 The layouts and results of capacity comparison between proactive (with context-awareness) and passive (without context-awareness) optimisation. There are three random examples in the 100 loops.

the experience. The next problem is to optimise the strategy in this area by automatically determining the best start (activation) time.

The activation time of load balancing determines the width of the poor-performance period which looks like a 'pit' dug by the burst traffic (see the yellow crosses lower than red triangles in Fig. 6.8 (d)). Moreover, the depth of 'pit' is the difference of capacity between with and without optimisation. The author manually triggers the load balancing at each discrete time from $t = 0$ to $t = 80$ and display the change of 'pit' width and depth in Fig. 6.9. As shown by the blue dot line, the bottom of the 'pit' drops fast after it appears (16.5 kb/s

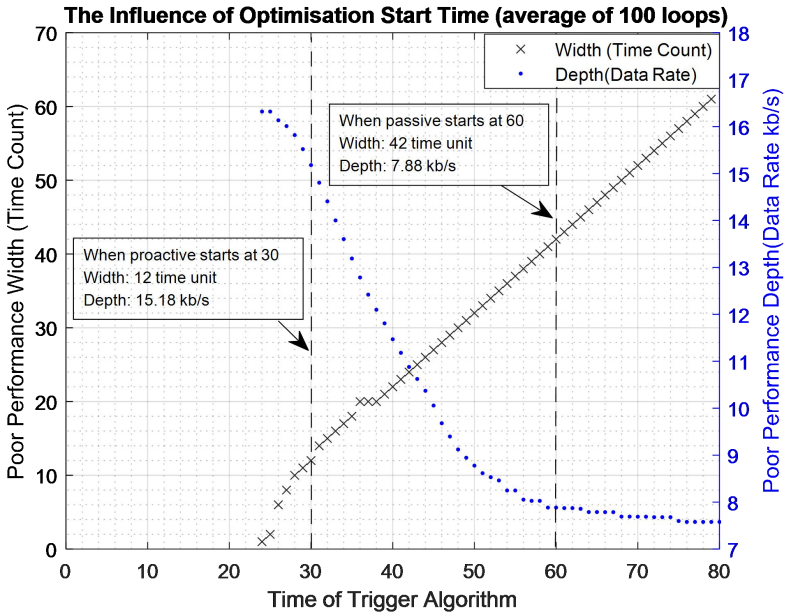


Fig. 6.9 This result indicates how wide and deep the poor performance 'pit' is when different trigger times are selected.

at $t = 24$) until approximately 7.6 kb/s at $t = 80$. Next, the width of poor-performance (black cross line) shows that with different activation times, the degradation 'pit' firstly does not occur (from $t = 0$ to $t = 24$), then increases non-linearly (from $t = 25$ to $t = 38$), finally increases linearly (from $t = 39$ to $t = 80$). The best time to trigger the proactive load balancing is at the end of the 'first period' ($t = 24$) because it has no poor-performance 'pit' while reserving the maximum time for the prediction calculation. Therefore, it is valuable to find this time point. However, currently, this work has to simulate all the discrete time points to visualise the phenomenon. It is inefficient and impossible in real-world conditions. In that case, this work proposes a feasible design to take advantage of the 'constant-nonlinear-linear' characteristic for reducing the complexity of simulation as much as possible.

The design flowchart of approximating the best trigger time is shown in Fig. 6.10. It follows the following steps:

- Choose primary points: this step chooses two time points at the very beginning t_{b1} and t_{b2}, then simulates two time points at the end t_{e1} and t_{e2}.

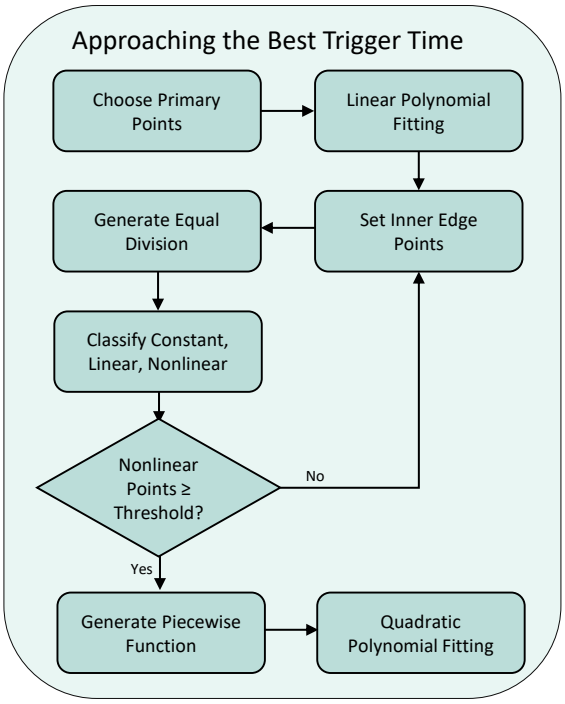


Fig. 6.10 The process of modelling the poor performance width for approaching the best trigger time for proactive load balancing

- Linear polynomial fitting: The corresponding width value for t_{b1} and t_{b2} is zero, and the t_{e1} and t_{e2} will generate a linear model through linear polynomial fitting.
- Set inner edge points: the points t_{b2} and t_{e1} are currently the closest points to the best triggering time, they are denoted as inner edge points.
- Generate equal division: the period between the two inner edge points is equally divided into n_d segments that there are $n_d - 1$ new time points.
- Classify constant, linear, and nonlinear: the system simulates the 'pit' width of these three time points, then classifies them to constant (0) or linear model. Otherwise, the time points belong to the non-linear part. The author needs to count the nonlinear time points $n_{\text{nonlinear}}$.
- Check number of nonlinear points: this work manually sets a threshold $\theta_{\text{nonlinear}}$ to check if there are enough nonlinear points for quadratic polynomial fitting ($n_{\text{nonlinear}} \geq \theta_{\text{nonlinear}}$).

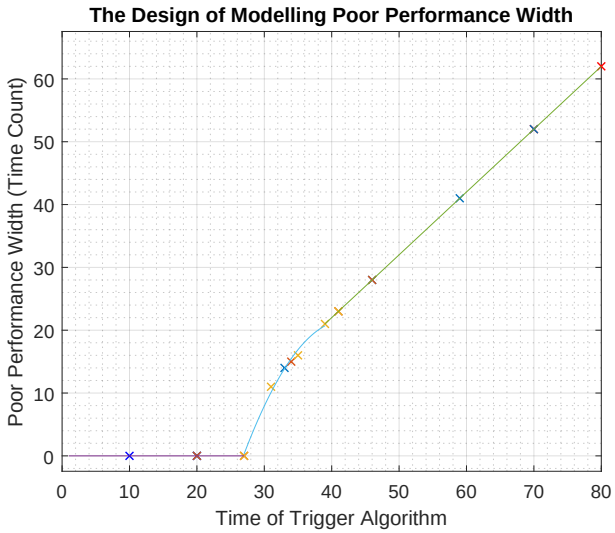


Fig. 6.11 The results of applying the design to model the poor-performance width and approach the best time for activating proactive load balancing. The time point $t = 27$ is the edge between the constant function and the nonlinear function, so it is the expected triggering time.

- If no ($n_{\text{nonlinear}} < \theta_{\text{nonlinear}}$): The system needs more iterations by setting new inner edge point, classifying, and checking nonlinear again. This process will approach the best triggering time.

- If yes ($n_{\text{nonlinear}} \geq \theta_{\text{nonlinear}}$): the system will generate piecewise function (constant, nonlinear function, linear function). The nonlinear part is modelled by the quadratic polynomial fitting.

- Finally, the best time for activating proactive optimisation is at the edge between the constant function and nonlinear function.

The above design is tested through setting $t_{b1} = 10$, $t_{b2} = 20$, $t_{e1} = 70$, $t_{e2} = 80$, $n_d = 4$, and $\theta_{\text{nonlinear}} = 3$. This work presents the result in Fig. 6.11. The line indicates the generated piecewise function, and the thirteen crosses represent the total simulation attempts. This result concludes that the design is feasible to find the best activation time of proactive load balancing ($t = 27$ in the current case study) with reducing the simulation times form 80 to 13.

6.5 Conclusion

In general, this work has three main contributions, the first one is designing the context-aware module about event (irregularity) detection associated with hotspots in urban regions, the second contribution is coupling the context-awareness and the proactive load balancing in urban regions, and the third one is optimising the proactive schemes about forecasting the best activation time. This design for approaching the best triggering time satisfies the requirements of reducing time complexity. Moreover, the proposed frameworks benefit not only the proactive load balancing but also other optimisations, such as proactive caching and interference management. These optimisation algorithms also require a traffic pattern for resource allocation.

Chapter 7

Conclusions and Future Work

With the development of the cellular network, the network optimisation has been developed from reactive towards proactive. The proactive optimisation can determine the network configuration in advance, so the time-sensitive services receive the optimised configuration in a short time. For example, the network congestion occurs when the network node carries more than it can handle, which is time sensitive because users are continuously experiencing reduced QoS, but some early signals (e.g., popular events) can trigger the attention for preparing the configuration. In that case, the period of poor-QoS lasts much shorter. In contrast, the base stations' malfunction hardly owns early signals, so the proactive optimisation does off-line computation and saves all possible solutions. When the malfunction occurs, machine learning techniques will select the best solution. Thanks to machine learning techniques, the heterogeneous data and the context-awareness became enablers for the proactive optimisations. Many contexts, such as geolocation, spatial-temporal traffic and behaviours have an easier way to model and improve the accuracy while meeting the complicate environments. In this thesis, the author combined the novelty of the context-awareness with the proactive network optimisations (named CAPO) for time-sensitive services, especially for dealing with the notwork congestion and malfunctions.

How can the CAPO be implemented to solve the problems with time-efficiency? This thesis provided three research lines: 1) a joint optimisation of UAV-BSs' locations and user association to support terrestrial BS's malfunction; 2) a hotspot off-loading optimisation

to avoid network congestion by UAV-BSs with addressing the multi-target and few-shot data problem; and 3) proactive load balancing to avoid network congestion during anomaly conditions. By following these research lines, this thesis contributed to the following aspects.

This thesis first assumed a region with base station malfunction and requiring to deploy UAV-BSs and associate users. To save the limited battery, the energy-efficiency is maximised. For example, compared with no optimisation (just randomly deploy the UAV-BSs), 28% power consumption was saved (user distribution parameters $\sigma = 1, \mu = -0.2$). Such an improvement was achieved by transforming the joint deployment-association optimisation problem into an MINLP problem, but this problem owned no direct solution. This work employed the divide-conquer algorithm to divide it into two sub-problems, then they can be conquered with optimum solutions, respectively. In detail, one sub-problem about optimal UAV-UE(s) association was solved by an improved Kuhn-Munkres (KM) algorithm, which was for minimising transmission power of UAV-BSs. In the test, compared with the current widely-used Greedy method, the proposed UAV-UE association method contributed 4% power reduction in the best condition ($\sigma = 1, \mu = -0.2$). The other sub-problem about UAV-BSs deployment was addressed through a time-efficient algorithm based on K Nearest Neighbour (KNN). In the experiment, the proposed method is compared with the widely used Simulated Annealing algorithm. The proposed method (UDUA) reduced the computation time from more than 10 s to at most 0.14 s.

Consequently, the proposed method could achieve similar performance on minimising the power consumption compared with existing widely-used methods, it also enabled outstanding time-efficiency improvement. The suggestions to network operators and pitfalls of using this method are listed.

- Best application scenario of this technique: the regions with BS malfunction, and the network has limited computational power and/or strict latency requirement.
- The proposed UDUA algorithm needs storage to save the spatial user distribution matrix and its corresponding UAV-BSs' configurations. This storage can be implemented on cloud for lower cost. Or, it is suggested to be enabled in mobile-edge base stations for lower transmission latency.

- The proposed UDUA algorithm can help utilise the idle period, such as night time. During this period, the nodes have low load and extra computation resources, so the computation ability can be used to do the off-line computing for high accuracy of the proposed UDUA.

Then, this thesis investigated solving the network congestion or overload through off-loading by UAV-BS aided network. This work aims to let the UAV-BSs serve as many users as possible but the resources used should be minimised. The UAV-BSs could determine their locations and other configurations automatically. For example in the experiment, with a particular user distribution given, the algorithm decided to dispatch 2 UAV-BSs to cover hotspots with around 60 resource blocks. This result was achieved by optimising the UAV-BSs' locations and resources as a multi-target joint optimisation problem. In order to solve it, this work further transformed it into a combinatorial problem which owned direct solution, the heuristic method. In that case, this work chose Simulated Annealing as the basic solution. The results proved that Simulated Annealing was able to solve the problem and cost 10^5 computational iterations. However, 10^5 computational iterations needed tens of seconds' computation on the experimental platform which is not time-efficient. This work further proposed CAPO-based (data-aware) Simulated Annealing for fast solving the problem. To show the ability of generalisation, the test user distributions are expanded to 60 frames. Among these 60 tests, CAPO-based method's average computational iteration is 3107 which is much smaller than the basic method (11180). If I require the algorithm to be finished in 10^4 iterations, the CAPO-based method made 76.67% tests successful, but the basic method only achieved 46.67%. Such a 30% improvement was due to the extra information in historical data. In other words, the quality of data had an unavoidable impact on CAPO-based methods performance. For example, if only $1/4$ of the total data can be used, CAPO's performance degraded from 76.67% to $< 20\%$, which proves its vulnerability to the data scarcity. To alleviate this threat, this work proposed the GAN-based CAPO method (short as generative method). After the implementation of GAN, CAPO's performance rose to $> 60\%$ while facing the same scarce data. That indicated an improvement of robustness while facing the data scarcity.

All of these methods can solve the multi-target optimisation problem but own different advantages. The traditional method requires no data, the data-aware method owns high time-efficiency, and the generative data-aware method is robust to data scarcity. In contrast, these methods also own their shortages, the traditional method is too slow, the data-aware method is sensitive to data quality, and the generative data-aware method needs to train a new component GAN. Operators can select these methods according to the actual requirement. The suggestions to network operators and pitfalls of using this method are listed.

- Best application scenario of this technique: the terrestrial BSs are estimated to be overloaded. A lot of users are crowded in this cell and the neighbouring cells. Extra communication supporting facilities are required urgently.

- By solving the proposed problem, UAV-BSs are automatically sent to the optimised locations to cover high-demand hotspots. This method is outstanding because the number of UAV-BSs can also be optimised, which is not included in the major of literature. Note, the operators need to have the information of users distribution and a network simulation tool.

- This work verified that relying on data will result in the vulnerable characteristic to the data scarcity. Using GAN is a feasible solution (as in this work), but GAN is also difficult to converge. The author suggests to change optimizers while training and avoids overfitting. If no data can be used, the author suggests to implement a storage to start collecting because the time-efficiency can have an increase by an order of magnitude.

Finally, this work addressed the network congestion/overload by balancing the load between base stations. Considering a cell where a popular event was happening and the cell was going to be overload, the proposed algorithm could handover the extra load to neighbouring cells and ensure the basic data rate. According to the Monte Carlo simulation, the average data rate of all the users in the involved cells has a 44.2% improvement. The proposed load balancing was based on Fuzzy control, so the optimisation rules were defined.

However, the time to start optimisation could have a magnificent influence on users experience. For example, users would not be benefited by the data rate improvement until the optimisation has started and finished the whole process. Such a problem was characterised as a cold-start problem, and CAPO was used to provide event detection. The application of CAPO reduces over 80% the poor performance period. Moreover, the proposed event-detection method was tested on Twitter data and successfully found 4 popular events as well as their spatial temporal information. When this information was provided to load balancing, the operators could proactively optimise the network configuration, and the computation time was reduced from 80 to 13 while searching the best triggering time.

Consequently, Fuzzy control is able to address the imbalance spatial traffic distribution, but a poor performance period exists for the network performance to recover. A social network based event detection could reduce the poor performance period. The operators require the social network data to support this algorithm. The following items are the suggestions and pitfalls for the operators.

- Best application scenario for this technique: urban region with irregular events happening, and the resources is not enough to serve the upcoming high traffic demands. UAV-BSs are not allowed to be used. And the only way is to re-organise current resources allocation.

- Operators require a stable data source of the social network data. Network congestion is generated by high communication demands. And there should be some reason for mobile users to aggregate to a cell and intensively use the network. To find the reason and proactively alert the potential congestion, operators can use the proposed event-detection method to analyse the public social networks, like Twitter, to automatically crawl key information to make the network to user-oriented.

- Event detection error exists. If the network traffic is not increasing as detected from social network, the BSs should be allowed to switch back to the normal operating condition.

7.1 Future Work

This thesis has indicated the early attempts to couple the potential of heterogeneous context awareness, statistical machine learning, and proactive network optimisation in a common cross-layer wireless framework. The broader impact of this work includes better cross-fertilising the academic fields of data analytics, mobile edge computing, AI, and wireless communications, as well as informing the industry of the promising potentials in this area. Nevertheless, despite the promising parts, there are still open issues and challenges to complete the circle of making heterogeneous data as a reliable data source. In this section, the author discusses these challenges and issues along with future research directions.

7.1.1 Prediction Error Impact

Prediction is the process of future context deduction according to experience, so the prediction error refers to the context information that is irrelevant for the future. Such unwanted variation always exists during sampling, training and testing. Moreover, both noisy data quality and rapid changing circumstance can cause it. For example, in geolocation prediction based on Twitter data, there are a lot of GPS coordinates' shifts caused by weak signals, especially for indoor Tweets. In that case, these indoor Tweets provide a location range (approximate a rectangular area) if Twitter cannot estimate an accurate coordinate. Such indoor data takes a large number of all geo-tagged Tweets. On the one hand, the geolocation prediction is inaccurate if all location records are considered. On the other hand, the prediction cannot represent the indoor user' context after filtering.

Accordingly, reducing the prediction errors and limiting the impact of unavoidable ones deserve researching efforts. Online data is unavoidably noisy and related to uncertain predictions. Therefore, future online-data based proactive optimisation algorithms have to consider the impact of prediction errors.

7.1.2 The Quantification of Uncertainty in Proactive Optimisation

To estimate the overhead caused by data scarcity and malicious attack, the certainty of the prediction needs to be quantified to regret the poor performance. The above process is described in Fig. 7.1. The detail steps are shown as follow:

(i) Gaussian Process and Bayesian learning can generate the posterior distribution based on the observations [197, 132]. Such distribution describes the certainty (confidence region) of demand prediction.

(ii) Then, in the proactive optimisation, the input is the demand (e.g., traffic) samples generated by the posterior distribution, and the output is the corresponding network quality which can be statistically counted by the histograms or the Kernel Density Estimation after several simulations. In general, the simulator outputs the QoS metric according to the traffic posterior distribution and provides a cascade QoS distribution.

(iii) Such a cascade distribution will quantify the confidence region of the proactive optimisation to compare with the reactive optimisation. In other words, its confidence area describes how the network can operate in future and its probability.

If the QoS of proactive optimisation is estimated to have poor performance worse than reactive optimisation, the regret of poor performance occurs. In contrast, the better performance area becomes benefit or profit. Based on the framework, the final decision is made according to the difference between profit and cost. In general, the uncertainty offers a probabilistic numerical estimation of the profit of proactive-optimisation decisions while facing data scarcity or malicious attacks.

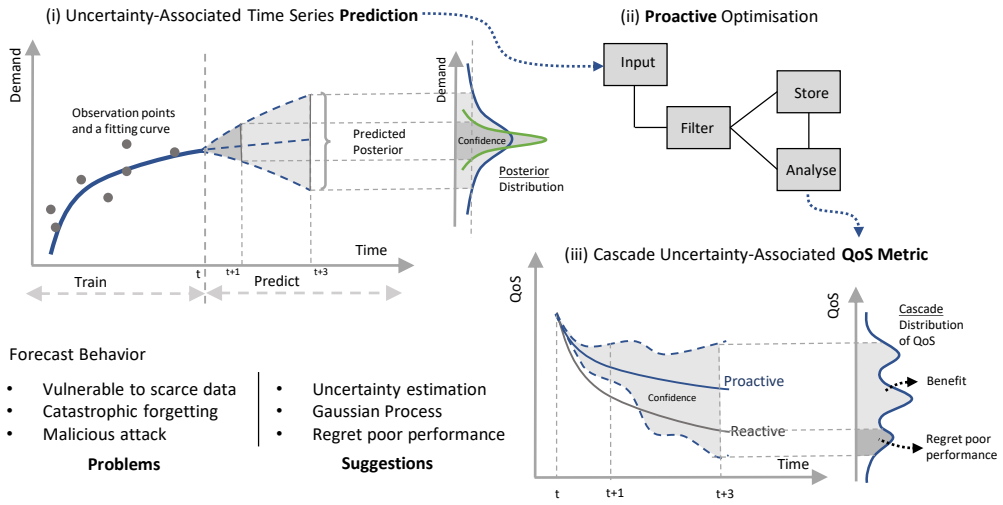


Fig. 7.1 The framework and sketch diagram of forecasting behaviours with uncertainty estimation. The proactive optimisation is fed with prediction and its confidence range. It then provides a cascade distribution of QoS. The system can estimate the potential cost and profit to quantify the overhead and make decisions.

References

[1] Bowei Yang, Weisi Guo, Bozhong Chen, Guangpu Yang, and Jie Zhang. Estimating mobile traffic demand using twitter. *IEEE Wireless Communications Letters*, 5(4):380–383, 2016.

[2] Elham Kalantari, Halim Yanikomeroglu, and Abbas Yongacoglu. On the number and 3d placement of drone base stations in wireless cellular networks. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pages 1–6. IEEE, 2016.

[3] Jingjing Yao, Tao Han, and Nirwan Ansari. On mobile edge caching. *IEEE Communications Surveys & Tutorials*, 21:2525–2553, thirdquarter 2019.

[4] Jeffrey G Andrews, Stefano Buzzi, Wan Choi, Stephen V Hanly, Angel Lozano, Anthony CK Soong, and Jianzhong Charlie Zhang. What will 5g be? *IEEE Journal on selected areas in communications*, 32(6):1065–1082, 2014.

[5] Ali Imran, Ahmed Zoha, and Adnan Abu-Dayya. Challenges in 5g: how to empower son with big data for enabling 5g. *IEEE network*, 28(6):27–33, 2014.

[6] Ajay R Mishra. *Advanced cellular network planning and optimisation: 2G/2.5 G/3G... evolution to 4G*. John Wiley & Sons, 2007.

[7] 3rd generation partnership project; technical specification group TSG RAN minimization of drive tests for E-UTRAN and UTRAN; overall description; stage 2 (Release 10), 3GPP TS 37.cde V0.3.0 , 2010-02.

[8] Robert Joyce and Li Zhang. Self organising network techniques to maximise traffic offload onto a 3G/WCDMA small cell network using MDT UE measurement reports. In *Global Communications Conference (GLOBECOM), 2014 IEEE*, pages 2212–2217. IEEE, 2014.

[9] Donna Fagen, Pablo A Vicharelli, and Jay Weitzen. Automated wireless coverage optimization with controlled overlap. *IEEE Transactions on Vehicular Technology*, 57(4):2395–2403, 2008.

[10] 3GPP work items on self-organizing networks, v0.1.3 (2014-06).

[11] Rouzbeh Razavi, Siegfried Klein, and Holger Claussen. Self-optimization of capacity and coverage in LTE networks using a fuzzy reinforcement learning approach. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, pages 1865–1870. IEEE, 2010.

[12] Weisi Guo and Tim O'Farrell. Dynamic cell expansion with self-organizing cooperation. *IEEE Journal on Selected Areas in Communications (JSAC)*, pages 851–860, 2013.

[13] Ingo Viering, Martin Dottling, and Andreas Lobinger. A mathematical perspective of self-optimizing wireless networks. In *2009 IEEE International Conference on Communications*, pages 1–6. IEEE, 2009.

[14] Andreas Lobinger, Szymon Stefanski, Thomas Jansen, and Irina Balan. Coordinating handover parameter optimization and load balancing in LTE self-optimizing networks. In *Vehicular technology conference (VTC spring), 2011 IEEE 73rd*, pages 1–5. IEEE, 2011.

[15] Alejandro Aguilar-Garcia, Sergio Fortes, Alfonso Fernandez Duran, and Raquel Barco. Context-aware self-optimization: Evolution based on the use case of load balancing in small-cell networks. *IEEE Vehicular Technology Magazine*, 11(1):86–95, Mar 2016.

[16] W. Guo and J. Zhang. Uncovering wireless blackspots using Twitter data. *Electronics Letters*, 53(12):814–816, Jun 2017.

[17] Xiaokang Wang, Laurence T Yang, Huazhong Liu, and M Jamal Deen. A big data-as-a-service framework: State-of-the-art and perspectives. *IEEE Transactions on Big Data*, 4(3):325–340, 2017.

[18] Bharath Keshavamurthy and Mohammad Ashraf. Conceptual design of proactive SONs based on the big data framework for 5G cellular networks: A novel machine learning perspective facilitating a shift in the SON paradigm. In *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*, pages 298–304, 2016.

[19] Run-Fa Liao, Hong Wen, Jinsong Wu, Huanhuan Song, Fei Pan, and Lian Dong. The rayleigh fading channel prediction via deep learning. *Wireless Communications and Mobile Computing*, 2018, 2018.

[20] Jaime Rodriguez, Isabel De la Bandera, P Munoz, and R Barco. Load balancing in a realistic urban scenario for LTE networks. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–5. IEEE, 2011.

[21] Li-Chun Wang, Shao-Hung Cheng, and Ang-Hsun Tsai. Bi-SON:big-data self organizing network for energy efficient ultra-dense small cells. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pages 1–5, Sep 2016.

[22] 3rd generation partnership project; technical specification group services and system aspects; study on non-MTC mobile data applications impacts (Release 11), 3GPP TR 22.801 V2.0.0, (2011-12).

[23] Cesar A. Sierra Franco and Jose Roberto B. de Marca. Load balancing in self-organized heterogeneous LTE networks: A statistical learning approach. In *2015 7th IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–5, Nov 2015.

[24] Pablo Munoz, Raquel Barco, Jose Maria Ruiz-Aviles, Isabel de la Bandera, and Alejandro Aguilar. Fuzzy rule-based reinforcement learning for load balancing techniques in enterprise LTE femtocells. *IEEE Transactions on Vehicular Technology*, 62(5):1962–1973, Jun 2013.

[25] 3rd generation partnership project; technical specification group radio access network; study on context aware service delivery in RAN for LTE; (Release 14),3GPP TR 36.933 V14.0.0, (2017-03).

[26] Stephen S. Mwanje and Andreas Mitschele-Thiel. A Q-Learning strategy for LTE mobility load balancing. In *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 2154–2158, Sep 2013.

[27] Holger Claussen, Lester TW Ho, and Louis G Samuel. Self-optimization of coverage for femtocell deployments. In *Wireless Telecommunications Symposium*, pages 278–285. IEEE, 2008.

[28] Selcuk Bassoy, Mona Jaber, Muhammad Ali Imran, and Pei Xiao. Load aware self-organising user-centric dynamic CoMP clustering for 5G networks. *IEEE Access*, 4:2895–2906, 2016.

[29] 3rd generation partnership project; technical specification group; services and system aspects; system architecture for the 5G system; stage 2 (Release 15), 3GPP TS 23.501 V0.0.0, (2017-01).

[30] 3rd generation partnership project; technical specification group services and system aspects; procedures for the 5G system; stage 2 (Release 15), 3GPP TS 23.502 V1.1.0, (2017-09).

[31] 3rd generation partnership project; technical specification group services and system aspects; policy and charging control framework for the 5G system; stage 2 (Release 15),3GPP TS 23.503 V15.1.0, (2018-03).

[32] Pol Blasco and Deniz Gündüz. Learning-based optimization of cache content in a small cell base station. In *Communications (ICC), 2014 IEEE International Conference on*, pages 1897–1903. IEEE, 2014.

[33] Veljko Pejovic and Mirco Musolesi. Anticipatory mobile computing: A survey of the state of the art and research challenges. *ACM Computing Surveys (CSUR)*, 47(3):47, 2015.

[34] Talal Alsedairy, Yinan Qi, Ali Imran, Muhammad Ali Imran, and Barry Evans. Self organising cloud cells: a resource efficient network densification strategy. *Transactions on Emerging Telecommunications Technologies*, 26(8):1096–1107, 2015.

[35] Jessica Moysen and Lorenza Giupponi. From 4G to 5G: Self-organized network management meets machine learning. *Computer Communications*, 2018.

[36] Paulo Valente Klaine, Muhammad Ali Imran, Oluwakayode Onireti, and Richard Demo Souza. A survey of machine learning techniques applied to self-organizing cellular networks. *IEEE Communications Surveys & Tutorials*, 19(4):2392–2431, 2017.

[37] Jinsong Wu, Igor Bisio, Chris Gniady, Ekram Hossain, Massimo Valla, and Haibo Li. Context-aware networking and communications: Part 1 [guest editorial]. *IEEE Communications Magazine*, 52(6):14–15, 2014.

[38] Jinsong Wu, Igor Bisio, Chris Gniady, Ekram Hossain, Massimo Valla, and Haibo Li. Context-aware networking and communications: Part 2 [guest editorial]. *IEEE Communications Magazine*, 52(8):64–65, 2014.

[39] John Krumm, Nigel Davies, and Chandra Narayanaswami. User-generated content. *IEEE Pervasive Computing*, 7(4):10–11, Oct 2008.

[40] Ejder Bastug, Mehdi Bennis, and Mérouane Debbah. Living on the edge: The role of proactive caching in 5G wireless networks. *IEEE Communications Magazine*, 52(8):82–89, Aug 2014.

[41] Konglin Zhu, Wenting Zhi, Lin Zhang, Xu Chen, and Xiaoming Fu. Social-aware incentivized caching for d2d communications. *IEEE Access*, 4:7585–7593, 2016.

[42] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *1994 First Workshop on Mobile Computing Systems and Applications*, pages 85–90. IEEE, 1994.

[43] Guanling Chen and David Kotz. A survey of context-aware mobile computing research. *Dartmouth Computer Science Technical Report TR2000-381*, 2000.

[44] Anthony D Wood, John A Stankovic, Gilles Virone, Leo Selavo, Zhimin He, Qiuhua Cao, Thao Doan, Yafeng Wu, Lei Fang, and Radu Stoleru. Context-aware wireless sensor networks for assisted living and residential monitoring. *IEEE network*, 22(4):26–33, 2008.

[45] Sabrina Müller, Onur Atan, Mihaela van der Schaar, and Anja Klein. Context-aware proactive content caching with service differentiation in wireless networks. *IEEE Transactions on Wireless Communications*, 16(2):1024–1036, 2016.

[46] Bo Ma, Weisi Guo, and Jie Zhang. A survey of online data-driven proactive 5g network optimisation using machine learning. *IEEE Access*, 8:35606–35637, 2020.

[47] Xiaokang Wang, Laurence T. Yang, Xia Xie, Jirong Jin, and M. Jamal Deen. A Cloud-Edge Computing Framework for Cyber-Physical-Social Services. *IEEE Communications Magazine*, 55(11):80–85, Nov 2017.

[48] Xiaokang Wang, Laurence T Yang, Jun Feng, Xingyu Chen, and M Jamal Deen. A tensor-based big service framework for enhanced living environments. *IEEE Cloud Computing*, 3(6):36–43, 2016.

[49] Dong Liu, Binqiang Chen, Chenyang Yang, and Andreas F Molisch. Caching at the wireless edge: design aspects, challenges, and future directions. *IEEE Communications Magazine*, 54(9):22–28, 2016.

[50] Neha Gupta, Henry Crosby, David Purser, Stephen Javis, and Weisi Guo. Twitter usage across industry: A spatiotemporal analysis. In *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 64–71. IEEE, 2018.

[51] Laurence T Yang, Xiaokang Wang, Xingyu Chen, JianJun Han, and Jun Feng. A tensor computation and optimization model for cyber-physical-social big data. *IEEE Transactions on Sustainable Computing*, 2017.

[52] Xiaokang Wang, Laurence T. Yang, Liwei Kuang, Xingang Liu, Qingxia Zhang, and M. Jamal Deen. A Tensor-Based Big-Data-Driven Routing Recommendation Approach for Heterogeneous Networks. *IEEE Network*, 33(1):64–69, Jan 2019.

[53] Liwei Kuang, Laurence T. Yang, Xiaokang Wang, Puming Wang, and Yaliang Zhao. A tensor-based big data model for QoS improvement in software defined networks. *IEEE Network*, 30(1):30–35, Jan 2016.

[54] Xiaokang Wang, Wei Wang, Laurence T Yang, Siwei Liao, Dexiang Yin, and M Jamal Deen. A distributed hosvd method with its incremental computation for big data in cyber-physical-social systems. *IEEE Transactions on Computational Social Systems*, 5(2):481–492, 2018.

[55] Yuzhou Li, Yu Zhang, Kai Luo, Tao Jiang, Zan Li, and Wei Peng. Ultra-dense HetNets meet big data: Green frameworks, techniques, and approaches. *IEEE Communications Magazine*, 56(6):56–63, 2018.

[56] Maj FJ Pinkney, Dan Hampel, and Stef DiPierro. Unmanned aerial vehicle (uav) communications relay. In *Proceedings of MILCOM'96 IEEE Military Communications Conference*, volume 1, pages 47–51. IEEE, 1996.

[57] Louis J Lanzerotti and Robert Evan Myer. Wireless telecommunications system having airborne base station, November 27 2001. US Patent 6,324,398.

[58] Ha Yoon Song. A method of mobile base station placement for high altitude platform based network with geographical clustering of mobile ground nodes. In *2008 International Multiconference on Computer Science and Information Technology*, pages 869–876. IEEE, 2008.

[59] Akram Al-Hourani, Sithamparanathan Kandeepan, and Simon Lardner. Optimal lap altitude for maximum coverage. *IEEE Wireless Communications Letters*, 3(6):569–572, 2014.

[60] Muhammad Junaid Farooq and Quanyan Zhu. A multi-layer feedback system approach to resilient connectivity of remotely deployed mobile Internet of Things. *IEEE Transactions on Cognitive Communications and Networking*, 4(2):422–432, 2018.

[61] Mohamed Alzenad, Amr El-Keyi, and Halim Yanikomeroglu. 3-D placement of an unmanned aerial vehicle base station for maximum coverage of users with different qos requirements. *IEEE Wireless Communications Letters*, 7(1):38–41, 2017.

[62] Mohamed Alzenad, Amr El-Keyi, Faraj Lagum, and Halim Yanikomeroglu. 3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage. *IEEE Wireless Communications Letters*, 6(4):434–437, 2017.

[63] Miao Jiang, Yiqing Li, Qi Zhang, and Jiayin Qin. Joint position and time allocation optimization of UAV enabled time allocation optimization networks. *IEEE Transactions on Communications*, 67(5):3806–3816, 2019.

[64] Xiao Liu, Yuanwei Liu, and Yue Chen. Reinforcement learning in multiple-UAV networks: Deployment and movement design. *IEEE Transactions on Vehicular Technology*, 68(8):8036–8049, 2019.

[65] Hajar El Hammouti, Mustapha Benjillali, Basem Shihada, and Mohamed-Slim Alouini. Learn-as-you-fly: A distributed algorithm for joint 3D placement and user association in multi-UAVs networks. *IEEE Transactions on Wireless Communications*, 18(12):5831–5844, 2019.

[66] Yong Zeng, Jie Xu, and Rui Zhang. Energy minimization for wireless communication with rotary-wing uav. *IEEE Transactions on Wireless Communications*, 18(4):2329–2345, 2019.

[67] Chi Harold Liu, Zheyu Chen, Jian Tang, Jie Xu, and Chengzhe Piao. Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach. *IEEE Journal on Selected Areas in Communications*, 36(9):2059–2070, 2018.

[68] Hasini Viranga Abeywickrama, Ying He, Eryk Dutkiewicz, Beeshanga Abewardana Jayawickrama, and Markus Mueck. A reinforcement learning approach for fair user coverage using UAV mounted base stations under energy constraints. *IEEE Open Journal of Vehicular Technology*, 1:67–81, 2020.

[69] Xiaowei Li, Haipeng Yao, Jingjing Wang, Xiaobin Xu, Chunxiao Jiang, and Lajos Hanzo. A near-optimal UAV-aided radio coverage strategy for dense urban areas. *IEEE Transactions on Vehicular Technology*, 68(9):9098–9109, 2019.

[70] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Mérouane Debbah. Wireless communication using unmanned aerial vehicles (UAVs): Optimal transport theory for hover time optimization. *IEEE Transactions on Wireless Communications*, 16(12):8052–8066, 2017.

[71] Kevin Dorling, Jordan Heinrichs, Geoffrey G Messier, and Sebastian Magierowski. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85, 2016.

[72] Meng Hua, Yi Wang, Zhengming Zhang, Chunguo Li, Yongming Huang, and Luxi Yang. Power-efficient communication in UAV-aided wireless sensor networks. *IEEE Communications Letters*, 22(6):1264–1267, 2018.

[73] Qianqian Zhang, Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Mérouane Debbah. Machine learning for predictive on-demand deployment of UAVs for wireless

communications. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2018.

[74] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Merouane Debbah. Optimal transport theory for power-efficient deployment of unmanned aerial vehicles. In *2016 IEEE international conference on communications (ICC)*, pages 1–6. IEEE, 2016.

[75] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Mérouane Debbah. Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications. *IEEE Transactions on Wireless Communications*, 16(11):7574–7589, 2017.

[76] Yan Kyaw Tun, Yu Min Park, Nguyen H Tran, Walid Saad, Shashi Raj Pandey, and Choong Seon Hong. Energy-efficient resource management in uav-assisted mobile edge computing. *IEEE Communications Letters*, 2020.

[77] Qianqian Zhang, Walid Saad, Mehdi Bennis, Xing Lu, Mérouane Debbah, and Wangda Zuo. Predictive deployment of uav base stations in wireless networks: Machine learning meets contract theory. *IEEE Transactions on Wireless Communications*, 2020.

[78] Linyan Lu, Ye Hu, Yirun Zhang, Guangyu Jia, Jiangtian Nie, and Mohammad Shikh-Bahaei. Machine learning for predictive deployment of uavs with rate splitting multiple access. In *2020 IEEE Globecom Workshops (GC Wkshps*, pages 1–6. IEEE, 2020.

[79] Luyao Xu, Ming Chen, Mingzhe Chen, Zhaohui Yang, Christina Chaccour, Walid Saad, and Choong Seon Hong. Joint location, bandwidth and power optimization for thz-enabled uav communications. *IEEE Communications Letters*, 2021.

[80] Holger Claussen and Doru Calin. Macrocell offloading benefits in joint macro-and femtocell deployments. In *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 350–354. IEEE, 2009.

[81] Doru Calin, Holger Claussen, and Huseyin Uzunalioglu. On femto deployment architectures and macrocell offloading benefits in joint macro-femto deployments. *IEEE Communications Magazine*, 48(1):26–32, 2010.

[82] Bo Han, Pan Hui, and Aravind Srinivasan. Mobile data offloading in metropolitan area networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 14(4):28–30, 2010.

[83] Bo Han, Pan Hui, VS Anil Kumar, Madhav V Marathe, Guanhong Pei, and Aravind Srinivasan. Cellular traffic offloading through opportunistic communications: a case study. In *Proceedings of the 5th ACM workshop on Challenged networks*, pages 31–38, 2010.

[84] Kyunghan Lee, Joohyun Lee, Yung Yi, Injong Rhee, and Song Chong. Mobile data offloading: How much can wifi deliver? In *Proceedings of the 6th International COnference*, pages 1–12, 2010.

[85] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, Young-Han Nam, and Mérouane Debbah. A tutorial on uavs for wireless networks: Applications, challenges, and open problems. *IEEE Communications Surveys & Tutorials*, 21(3):2334–2360, 2019.

[86] Walid Saad, Mehdi Bennis, and Mingzhe Chen. A vision of 6g wireless systems: Applications, trends, technologies, and open research problems. *IEEE network*, 34(3):134–142, 2019.

[87] Fen Cheng, Shun Zhang, Zan Li, Yunfei Chen, Nan Zhao, F Richard Yu, and Victor CM Leung. Uav trajectory optimization for data offloading at the edge of multiple cells. *IEEE Transactions on Vehicular Technology*, 67(7):6732–6736, 2018.

[88] Jiangbin Lyu, Yong Zeng, and Rui Zhang. Uav-aided offloading for cellular hotspot. *IEEE Transactions on Wireless Communications*, 17(6):3988–4001, 2018.

[89] Irem Bor-Yaliniz and Halim Yanikomeroglu. The new frontier in ran heterogeneity: Multi-tier drone-cells. *IEEE Communications Magazine*, 54(11):48–55, 2016.

[90] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Mérouane Debbah. Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. *IEEE Communications Letters*, 20(8):1647–1650, 2016.

[91] Jože Košmerl and Andrej Vilhar. Base stations placement optimization in wireless networks for emergency communications. In *2014 IEEE international conference on communications workshops (ICC)*, pages 200–205. IEEE, 2014.

[92] Weisen Shi, Junling Li, Wenchao Xu, Haibo Zhou, Ning Zhang, Shan Zhang, and Xuemin Shen. Multiple drone-cell deployment analyses and optimization in drone assisted radio access networks. *IEEE Access*, 6:12518–12529, 2018.

[93] Jun-Woo Cho and Jae-Hyun Kim. Performance comparison of heuristic algorithms for uav deployment with low power consumption. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1067–1069. IEEE, 2018.

[94] Zhe Yu, Yanmin Gong, Shimin Gong, and Yuanxiong Guo. Joint task offloading and resource allocation in uav-enabled mobile edge computing. *IEEE Internet of Things Journal*, 7(4):3147–3159, 2020.

[95] Bo Yang, Xuelin Cao, Chau Yuen, and Lijun Qian. Offloading optimization in edge computing for deep learning enabled target tracking by internet-of-uavs. *IEEE Internet of Things Journal*, 2020.

[96] Yi Liu, Shengli Xie, and Yan Zhang. Cooperative offloading and resource management for uav-enabled mobile edge computing in power iot system. *IEEE Transactions on Vehicular Technology*, 69(10):12229–12239, 2020.

[97] Mesut E Baran and Felix F Wu. Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Power Engineering Review*, 9(4):101–102, 1989.

[98] Sajal K Das, Sanjoy K Sen, and Rajeev Jayaram. A dynamic load balancing strategy for channel assignment using selective borrowing in cellular mobile environment. *Wireless Networks*, 3(5):333–347, 1997.

[99] Andreas Lobinger, Szymon Stefanski, Thomas Jansen, and Irina Balan. Load balancing in downlink LTE self-optimizing networks. In *2010 IEEE 71st Vehicular Technology Conference*, pages 1–5, 2010.

[100] Honglin Hu, Jian Zhang, Xiaoying Zheng, Yang Yang, and Ping Wu. Self-configuration and self-optimization for LTE networks. *IEEE Communications Magazine*, 48(2):94–100, Feb 2010.

[101] Jose Maria Ruiz-Avilés, Salvador Luna-Ramírez, Matias Toril, and Fernando Ruiz. Traffic steering by self-tuning controllers in enterprise LTE femtocells. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):337, Dec 2012.

[102] Juanxiong Xu, Lun Tang, Qianbin Chen, and Li Yi. Study on based reinforcement Q-Learning for mobile load balancing techniques in LTE-A HetNets. In *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*, pages 1766–1771. IEEE, 2014.

[103] Sven Tomforde, Alexander Ostrovsky, and Jörg Hähner. Load-aware reconfiguration of lte-antennas dynamic cell-phone network adaptation using organic network control. In *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 1, pages 236–243. IEEE, 2014.

[104] Ogechi Akudo Nwogu, Gladys Diaz, and Marwen Abdennebi. An optimized approach to load balancing and resource usage in 5g multi-tiered cellular networks. In *2020 Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1–5. IEEE, 2020.

[105] Rizwana Ahmad, Mohammad Dehghani Soltani, Majid Safari, Anand Srivastava, and Abir Das. Reinforcement learning based load balancing for hybrid lifi wifi networks. *IEEE Access*, 8:132273–132284, 2020.

[106] Trinh Van Chien, Emil Björnson, and Erik G Larsson. Joint power allocation and load balancing optimization for energy-efficient cell-free massive mimo networks. *IEEE Transactions on Wireless Communications*, 19(10):6798–6812, 2020.

[107] P Munoz, R Barco, Isabel de la Bandera, Matías Toril, and Salvador Luna-Ramirez. Optimization of a fuzzy logic controller for handover-based load balancing. In *Vehicular technology conference (VTC Spring), 2011 IEEE 73rd*, pages 1–5. IEEE, 2011.

[108] Aderemi A Atayero, Matthew K Luka, and Adeyemi A Alatishe. Neural-encoded fuzzy models for load balancing in 3GPP LTE. 2012.

[109] Toshihito Kudo and Tomoaki Ohtsuki. Q-learning based cell selection for ue outage reduction in heterogeneous networks. In *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*, pages 1–5. IEEE, 2014.

[110] Henrik Klessig, Henning Kuntzschmann, Lucas Scheuvens, Bjoern Almeroth, Philipp Schulz, and Gerhard Fettweis. Twitter as a source for spatial traffic information in big data-enabled self-organizing networks. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–5, Mar 2017.

[111] Guangxia Xu, Shiyi Gao, Mahmoud Daneshmand, Chonggang Wang, and Yanbing Liu. A survey for mobility big data analytics for geolocation prediction. *IEEE Wireless Communications*, 24(1):111–119, Feb 2017.

[112] Mingzhe Chen, Mohammad Mozaffari, Walid Saad, Changchuan Yin, Mérouane Debbah, and Choong Seon Hong. Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience. *IEEE Journal on Selected Areas in Communications*, 35(5):1046–1061, 2017.

[113] Daniel Ashbrook and Thad Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, Oct 2003.

[114] Daniel A. Kisilevich, Slava; Mansmann, Florian; Keim. P-DBSCAN : A density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. In *COM.Geo*, page 274, 2010.

[115] Keiichi Tamura and Takumi Ichimura. Density-based spatiotemporal clustering algorithm for extracting bursty areas from georeferenced documents. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 2079–2084, Oct 2013.

[116] Hila Becker, Dan Iter, Mor Naaman, and Luis Gravano. Identifying content for planned events across social media sites.

[117] Ryong Lee and Kazutoshi Sumiya. Measuring geographical regularities of crowd behaviours for Twitter-based geo-social event detection. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks - LBSN '10*, page 1, 2010.

[118] Kazufumi Watanabe, Masanao Ochi, Makoto Okabe, and Rikio Onai. Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs. In *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, page 2541, New York, New York, USA, 2011.

[119] Guangshuo Chen, Sahar Hoteit, Aline Carneiro Viana, Marco Fiore, and Carlos Sarraute. *Spatio-Temporal Predictability of Cellular Data Traffic*. PhD thesis, INRIA Saclay-Ile-de-France, 2017.

[120] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

[121] Ketan R Pandhare and Medha A Shah. Real time road traffic event detection using twitter and spark. In *Inventive Communication and Computational Technologies (ICICCT), 2017 International Conference on*, pages 445–449. IEEE, 2017.

[122] Keiji Yanai, Keita Yaegashi, and Bingyu Qiu. Detecting cultural differences using consumer-generated geotagged photos. In *Proceedings of the 2nd International Workshop on Location and the Web - LOCWEB '09*, pages 1–4, New York, New York, USA, 2009.

[123] Sanam Narejo and Eros Pasero. An application of Internet traffic prediction with deep neural network. In *Multidisciplinary Approaches to Neural Computing*, pages 139–149. Springer, 2018.

[124] Chen Qiu, Yanyan Zhang, Zhiyong Feng, Ping Zhang, and Shuguang Cui. Spatio-temporal wireless traffic prediction with recurrent neural network. *IEEE Wireless Communications Letters*, 7(4):554–557, 2018.

[125] Arjuna Sathiaseelan, M. Said Seddiki, Stoyan Stoyanov, Dirk Trossen, Arjuna Sathiaseelan, M. Said Seddiki, Stoyan Stoyanov, and Dirk Trossen. Social SDN: online social networks integration in wireless network provisioning. In *Proceedings of the 2014 ACM conference on SIGCOMM - SIGCOMM '14*, volume 44, pages 375–376, New York, New York, USA, 2014.

[126] Yuxiu Hua, Zhifeng Zhao, Rongpeng Li, Xianfu Chen, Zhiming Liu, and Honggang Zhang. Traffic prediction based on random connectivity in deep learning with long short-term memory. *arXiv preprint arXiv:1711.02833*, 2017.

[127] Meng Xu, Qiaoling Wang, and Qinliang Lin. Hybrid holiday traffic predictions in cellular networks. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6. IEEE, 2018.

[128] Denis Tikunov and Toshikazu Nishimura. Traffic prediction for mobile network using holt-winter's exponential smoothing. In *2007 15th International Conference on Software, Telecommunications and Computer Networks*, pages 1–5. IEEE, 2007.

[129] Fengli Xu, Yuyun Lin, Jiaxin Huang, Di Wu, Hongzhi Shi, Jeungeun Song, and Yong Li. Big data driven mobile traffic understanding and forecasting: A time series approach. *IEEE transactions on services computing*, 9(5):796–805, 2016.

[130] Utpal Paul, Anand Prabhu Subramanian, Milind Madhav Buddhikot, and Samir R Das. Understanding traffic dynamics in cellular data networks. In *2011 Proceedings IEEE INFOCOM*, pages 882–890. IEEE, 2011.

[131] Rongpeng Li, Zhifeng Zhao, Jianchao Zheng, Chengli Mei, Yueming Cai, and Honggang Zhang. The learning and prediction of application-level traffic data in cellular networks. *IEEE Transactions on Wireless Communications*, 16(6):3899–3912, 2017.

[132] Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

[133] Sheng Zhang, Shenglin Zhao, Mingxuan Yuan, Jia Zeng, Jianguo Yao, Michael R Lyu, and Irwin King. Traffic prediction based power saving in cellular networks: A machine learning method. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 29. ACM, 2017.

[134] Janne Riihijarvi and Petri Mahonen. Machine learning for performance prediction in mobile cellular networks. *IEEE Computational Intelligence Magazine*, 13(1):51–60, 2018.

[135] Federico Botta, Helen Susannah Moat, and Tobias Preis. Quantifying crowd size with mobile phone and Twitter data. *Royal Society Open Science*, 2(5):150162, May 2015.

[136] Md Salik Parwez, Danda B Rawat, and Moses Garuba. Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network. *IEEE Transactions on Industrial Informatics*, 13(4):2058–2065, 2017.

[137] Luong-Vy Le, Do Sinh, Bao-Shuh Paul Lin, and Li-Ping Tung. Applying big data, machine learning, and SDN/NFV to 5G traffic clustering, forecasting, and management. In *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, pages 168–176. IEEE, 2018.

[138] Luong-Vy Le, Do Sinh, Li-Ping Tung, and Bao-Shuh Paul Lin. A practical model for traffic forecasting based on big data, machine-learning, and network KPIs. In *2018*

15th IEEE Annual Consumer Communications & Networking Conference (CCNC), pages 1–4. IEEE, 2018.

[139] M Zubair Shafiq, Lusheng Ji, Alex X Liu, and Jia Wang. Characterizing and modeling internet traffic dynamics of cellular devices. *ACM SIGMETRICS Performance Evaluation Review*, 39(1):265–276, 2011.

[140] Rong Xie, Yang Chen, Qinge Xie, Yu Xiao, and Xin Wang. We know your preferences in new cities: Mining and modeling the behavior of travelers. *IEEE Communications Magazine*, 56(11):28–35, 2018.

[141] Feng Zeng, Runhua Wang, and Jinsong Wu. How mobile contributors will interact with each other in mobile crowdsourcing with word of mouth mode. *IEEE Access*, 7:14523–14536, 2019.

[142] Adam Tsakalidis, Maria Liakata, Theodoros Damoulas, Brigitte Jellinek, Weisi Guo, and Alexandra I Cristea. Combining heterogeneous user generated data to sense well-being. The COLING 2016 Organizing Committee, 2016.

[143] Datasets on Kaggle.

[144] Open Big Data Italia Cellular Traffic and Tweets.

[145] Chaoyun Zhang, Xi Ouyang, and Paul Patras. ZipNet-GAN: Inferring fine-grained mobile traffic patterns via a generative adversarial neural network. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, pages 363–375, 2017.

[146] Ghazaleh Khodabandelou, Vincent Gauthier, Marco Fiore, and Mounim El Yacoubi. Estimation of static and dynamic urban populations with mobile network metadata. *IEEE Transactions on Mobile Computing*, 18:2034–2047, 9 2019.

[147] B Yang, W Guo, B Chen, G Yang, and J Zhang. Data from: Estimating mobile traffic demand using twitter, 2016.

[148] Comparing different clustering algorithms on toy datasets — scikit-learn 0.20.3 documentation.

[149] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[150] Anna L Buczak and Erhan Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2):1153–1176, 2016.

[151] Comparing different clustering algorithms on toy datasets — scikit-learn 0.20.3 documentation.

[152] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

[153] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014.

[154] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014.

[155] Liudmyla Nechepurenko, Viktor Voss, and Vyacheslav Gritsenko. Comparing knowledge-based reinforcement learning to neural networks in a strategy game. *arXiv preprint arXiv:1901.04626*, 2019.

[156] Sven Koenig and Reid G Simmons. Complexity analysis of real-time reinforcement learning. In *AAAI*, pages 99–107, 1993.

[157] Mingzhe Chen, Walid Saad, Changchuan Yin, and Mérouane Debbah. Echo state networks for proactive caching in cloud-based radio access networks with mobile users. *IEEE Transactions on Wireless Communications*, 16(6):3520–3535, 2017.

[158] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5353–5360, 2015.

[159] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[160] Norbert Jankowski. Complexity measures for meta-learning and their optimality. In *Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence*, pages 198–210. Springer, 2013.

[161] Jun He and Xin Yao. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 127(1):57–85, 2001.

[162] Philipp Hennig, Michael A Osborne, and Mark Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2179):20150142, 2015.

[163] Krishna B Misra. Reliability optimization of a series-parallel system. *IEEE Transactions on Reliability*, 21(4):230–238, 1972.

[164] Lori M Kaufman. Data security in the world of cloud computing. *IEEE Security & Privacy*, 7(4):61–64, 2009.

[165] Richard Chow. The last mile for iot privacy. *IEEE Security & Privacy*, 15(6):73–76, 2017.

[166] Cookies consent under the GDPR - EU GDPR Compliant.

[167] Diana Maimut and Reza Reyhanitabar. Authenticated encryption: Toward next-generation algorithms. *IEEE Security & Privacy*, 12(2):70–72, 2014.

[168] Songlin Chen, Hong Wen, Jinsong Wu, Jie Chen, Wenjie Liu, Lin Hu, and Yi Chen. Physical-layer channel authentication for 5G via machine learning algorithm. *Wireless Communications and Mobile Computing*, 2018, 2018.

[169] Kian Hamedani, Lingjia Liu, Rachad Atat, Jinsong Wu, and Yang Yi. Reservoir computing meets smart grids: Attack detection using delayed feedback networks. *IEEE Transactions on Industrial Informatics*, 14(2):734–743, 2018.

[170] Rachad Atat, Lingjia Liu, Jinsong Wu, Guangyu Li, Chunxuan Ye, and Yi Yang. Big data meet cyber-physical systems: A panoramic survey. *IEEE Access*, 6:73603–73636, 2018.

[171] Zaobo He, Zhipeng Cai, and Jiguo Yu. Latent-data privacy preserving with customized data utility for social network data. *IEEE Transactions on Vehicular Technology*, 67(1):665–673, 2018.

[172] Craig Gentry et al. Fully homomorphic encryption using ideal lattices. In *Stoc*, volume 9, pages 169–178, 2009.

[173] Harold W Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[174] Zitian Zhang, Yue Wu, Yuteng Hou, and Jie Zhang. Downlink data scheduling to optimize the serviceability in fog radio access networks. In *International Conference in Communications, Signal Processing, and Systems*, pages 837–842. Springer, 2018.

[175] Xinran Luo, Yan Zhang, Zunwen He, Guanshu Yang, and Zijie Ji. A two-step environment-learning-based method for optimal UAV deployment. *IEEE Access*, 7:149328–149340, 2019.

[176] David López-Pérez, Alvaro Valcarce, Guillaume De La Roche, and Jie Zhang. Ofdma femtocells: A roadmap on interference avoidance. *IEEE communications magazine*, 47(9):41–48, 2009.

[177] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[178] 3GPP. Study on enhanced LTE support for aerial vehicles (release 15). Technical Report (TR) 36.777, 3rd Generation Partnership Project (3GPP), 12 2017. Version 15.0.0.

[179] Yuzhou Li, Min Sheng, Yuhua Sun, and Yan Shi. Joint optimization of BS operation, user association, subcarrier assignment, and power allocation for energy-efficient hetnets. *IEEE journal on selected areas in communications*, 34(12):3339–3353, 2016.

[180] Yan Chen, Toni Farley, and Nong Ye. Qos requirements of network applications on the internet. *Information Knowledge Systems Management*, 4(1):55–76, 2004.

[181] Dongheon Lee, Sheng Zhou, Xiaofeng Zhong, Zhisheng Niu, Xuan Zhou, and Honggang Zhang. Spatial modeling of the traffic density in cellular networks. *IEEE Wireless Communications*, 21(1):80–88, 2014.

[182] Lucas P Behnck, Dionisio Doering, Carlos Eduardo Pereira, and Achim Rettberg. A modified simulated annealing algorithm for SUAVs path planning. *IFAC-PapersOnLine*, 48(10):63–68, 2015.

[183] Xiang Zhang, Fanqin Zhou, Jiayi Ning, Peng Yu, and Wenjing Li. Hotspot localization and prediction in wireless cellular networks via spatial traffic fitting. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6. IEEE, 2018.

[184] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[185] Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 46(sup1):234–240, 1970.

[186] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[187] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[188] Mark Veillette. Stbl: Alpha stable distributions for matlab. *Matlab Central File Exchange, retreived October*, 10:2012, 2012.

[189] Bo Ma, Bowei Yang, Zitian Zhang, and Jie Zhang. Modelling mobile traffic patterns using a generative adversarial neural networks. In *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–7. IEEE, 2020.

[190] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[191] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 2019.

[192] Zhenyu Zhou, Chuntian Zhang, Chen Xu, Fei Xiong, Yan Zhang, and Tariq Umer. Energy-efficient industrial Internet of UAVs for power line inspection in smart grid. *IEEE Transactions on Industrial Informatics*, 14(6):2705–2714, 2018.

[193] Sergio Fortes, David Palacios, Inmaculada Serrano, and Raquel Barco. Applying social event data for the management of cellular networks. *IEEE Communications Magazine*, 56(11):36–43, 2018.

[194] Alejandro Aguilar-Garcia, Sergio Fortes, Alfonso Fernandez Duran, and Raquel Barco. Context-aware self-optimization: Evolution based on the use case of load balancing in small-cell networks. *IEEE Vehicular Technology Magazine*, 11(1):86–95, 2016.

[195] Two-ray propagation channel - MATLAB.

[196] M Series. Guidelines for evaluation of radio interface technologies for imt-advanced. *Report ITU*, 638:1–72, 2009.

[197] Houman Owhadi, Clint Scovel, and Tim Sullivan. On the brittleness of bayesian inference. *SIAM Review*, 57(4):566–582, 2015.