USF Tampa Graduate Theses and Dissertations　　USF Graduate Theses and Dissertations

April 2020

Socially Aware Network User Mobility Analysis and Novel Approaches on Aerial Mobile Wireless Network Deployment

Ismail Uluturk
University of South Florida

Follow this and additional works at: https://digitalcommons.usf.edu/etd

Part of the Electrical and Computer Engineering Commons

Socially Aware Network User Mobility Analysis and

Novel Approaches on Aerial Mobile Wireless Network Deployment

by

Ismail Uluturk

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering
Department of Electrical Engineering
College of Engineering
University of South Florida

Co-Major Professor: Ismail Uysal, Ph.D.
Co-Major Professor: Kwang-Cheng Chen, Ph.D.
Gisele Bennett, Ph.D.
Giovanni Luca Ciampaglia, Ph.D.
Robert Hooker, Ph.D.
Ravi Sankar, Ph.D.

Date of Approval:
April 8, 2020

Keywords: Mobility Networks, Aerial Wireless Networks, Mobile Wireless Network
Deployment, Reinforcement Learning, Inter-agent Communication

Table of Contents

List of Tables

List of Figures

Abstract

Service demand patterns for wireless networks are evolving with the technological developments in areas such as personal computing, unmanned vehicles, and internet-of-things, where increasing mobile service demand is one of the significant challenges introduced. In addition to these new intrinsic dynamics, natural disasters and societal upheaval are also disrupting the conventional patterns of network demand. Situations like damaged infrastructure due to a natural disaster or large numbers of displaced people caused by political strife and social upheaval demand flexible, rapidly deployable network architectures. The increasing demands of next-generation communication services are straining the capabilities of the traditional approach of the permanent deployments of fixed infrastructure, presenting a need for novel approaches.

The availability of previously unprecedented data sources also brings data-driven approaches to the spotlight. Having access to large-scale longitudinal datasets of network user behavior makes it possible to study and model user behavior which can then be used to improve services provided to users.

This dissertation has its primary contributions in socially aware analysis of cellular network usage data and developing methods for forming flexible aerial wireless network infrastructures in applications challenging for traditional infrastructure deployments. Specifically, this dissertation first presents an investigation of the wireless network user data to generate actionable intelligence that can guide policymakers and point out the changing demands of wireless networks discovering user modalities outside the common user patterns and then

explores the use of aerial platforms for rapid deployment of flexible wireless networks to satisfy the mobility and reliability demands of the future wireless communications.

Chapter 1: Introduction

This chapter provides a summary of the contributions presented throughout this dissertation. Demands and expectations from wireless networks have been evolving in the recent years. Following the technological developments in areas such as personal computing, unmanned vehicles, and internet of things (IoT) and shifting societal norms due to the ever increasing number of services moving online and relying heavily on cellular phones for delivery creates a significant wireless service demand for increasingly more mobile users. The increasing demands of the next generation communication services are straining the capabilities of the traditional approach of the permanent deployments of fixed infrastructure, presenting a need for novel approaches.

In addition to these new intrinsic dynamics, natural disasters are also disrupting the conventional patterns of network demand. As an example, in the aftermath of Hurricane Irma, Puerto Rico has lost its wireless infrastructure for an extended period. Scenarios like this can benefit from flexible, temporary network deployments that can serve as a stop-gap until the network infrastructure is restored. This can also be used to provide service in remote areas with temporary or irregular service demand where a permanent infrastructure investment is not practical.

Another significant source of perturbation in the network demand patterns is the displaced populations of regions that have been in turmoil due to political strife and social upheaval. Last decade has seen refugee numbers unprecedented in recent history. Ubiquitousness of online information sources and services also means that cellular phone service is a vital resource for the displaced populations. In fact, field reports indicate that refugees

often ask for phone chargers and cellular service before food and water [2]. Wireless service demands for these displaced populations is likely to differ from the daily lives of local populations. This calls for an investigation into the network usage and mobility patterns of refugee users for satisfactory service coverage. Of course, refugee populations are not the only group with diverging usage patterns compared to local populations and these studies can be extended to any other group.

The primary goals of this dissertation then is to first understand the network usage and mobility patterns of distinct cellular user groups and develop methods for rapidly deploying a flexible mobile wireless network to cover scenarios where conventional network deployments fall short.

1.1 Socially Aware Cellular User Mobility Analysis

Widespread adoption of cellular phones has made big longitudinal datasets of Call Detail Records (CDR) available. Combined with the advances in big data analysis this has opened many avenues of research and application [3]. Some of the fields that utilize CDR datasets with example works are; public health with studies on how mobility effects infectious disease spread [4, 5], development of new agent based models to explain disease migration [6], effects of mobility during holidays and religious events on public health [7], a study of disease containment scenarios based on mobility [8], urban planning with studies on managing disasters and resilience [9], data-driven utility infrastructure deployment plans [10], analysis of social phenomena with studies on data-driven improved poverty mapping [11], detecting unusual and emergency events [12], and mapping of real-world events and locations to people [13]. It should be noted that often the availability of mobility models developed through the analysis of CDR datasets is the main driving factor behind these works.

Availability of CDR datasets to researchers can also drive public good projects and help researchers produce works that can be referenced by policymakers, as it has been demon-

strated by the "Data 4 Good" challenges [14, 15]. In fact, this chapter has been originally prepared for the "Data 4 Refugees" challenge [16, 17] that aimed to produce actionable data to policy makers of The Republic of Turkey during the unprecedented flow of refugees into the country. This chapter introduces the tools used in mobility analysis, and studies a real CDR dataset collected from 50000 users over a year to highlight unique patterns of cellular usage and mobility of refugee users and how they differ from non-refugee users. Our findings suggest that refugee users are not well integrated into the local daily life through data-driven observations. We uniquely apply Variable Higher-Order Networks to the mobility analysis based on CDR data, which removes the first-order dependency introduced by more common models following the Markov property to better represent the sequential trajectory data necessary for mobility analysis.

While the original goal of this study is to provide actionable data for policy makers, the work in this chapter has significant implications for the field of wireless networks as well. First of all, the existence of accurate mobility models of users can significantly improve network design and deployments especially for highly mobile user bases. Furthermore, the findings suggesting the existence of a large group of refugee users that have differing cellular traffic and mobility patterns create a unique challenge to provide coverage for groups that do not follow the common local patterns, especially vulnerable populations such as refugees, that may be present in the areas only temporarily. This indicates a demand for flexible, temporary infrastructures that can be deployed rapidly.

1.2 Aerial Wireless Network Deployment

Usage of aerial platforms for wireless network deployment brings an additional degree of freedom to network design that can answer the mobile heterogeneous traffic demand of next the generation. They are also suitable for answering the challenges laid down in 1.1, enabling quick, flexible, and temporary deployments. There are significant attempts from

the industry towards aerial deployments as well, for example AT&T deployed it's Flying Cell-on-Wheels (COW) experimentally after Hurricane Maria in Puerto Rico to provide service when terrestrial infrastructure was damaged.

In this chapter, the placement problem for multiple access points (APs) in full 3D is presented and addressed. Placement problem on a 2D plane is a well studied problem, however it's extension to 3D is not trivial and there may not even be a direct extension of these results [18]. The Kepler Conjecture, more informally known as the densest sphere packing problem, is an analogue for this 3D placement problem and has only been formally proved recently by a large team after remaining an open problem for 400 years [19].

Works in the literature have considered single [20, 21, 22, 23] or multiple [24, 25, 26] UAV deployments, however these works follow optimization schemes that are not tractable in full 3D and often make limiting assumptions to reduce the problem into a circle coverage problem on a 2D surface which represents a more trivial problem.

This chapter presents a formulation for the genuine 3D placement of multiple low-altitude short-range aerial APs directly, utilizing concepts from the analysis of random graphs, particularly Random Geometric Graphs (RGGs). An efficient and scalable sequential iterative algorithm is then proposed as a practical solution for the optimal placement problem in 3D against the rapidly increasing search space of AP locations as the number of APs increase. Resulting individual optimization problems are solved utilizing evolutionary methods as the unpredictable environment conditions and network dynamics make the gradient based methods unfeasible. Furthermore, a procedure to determine the adequate number of APs to be deployed in the given area for the desired fraction of coverage is also presented. Detailed agent based simulations with ray-tracing line-of-sight calculation shows that operating in full 3D is advantageous especially when avoiding obstacles and conforming to the environment is crucial to get a good connection between the users and APs.

1.3 Aerial Mobile Radio Access Network Design

The deployment scheme in 1.2 works on static snapshots of the system, and have strong priors. In this chapter, these assumptions are relaxed towards a more practical system. A flexible and rapidly deployable aerial radio access network (aRAN) is developed, which will be crucial in both post-disaster communications and highly dynamic usage patterns of humans and machines expected in the near future.

Operating in a dynamic environment with limited prior knowledge of the network and user dynamics, tight closed loop of UAVs to implement an aRAN is a luxury. Following this, designing trajectories for the UAVs to effectively serve user traffic results in a significant challenge that is computationally unfeasible with the conventional methods in the literature [27, 28].

In this chapter each UAV based aerial Access Point (aAP) is treated as an autonomous mobile robot, and use Reinforcement Learning (RL) for the trajectory design control problem of the aRAN. This addresses both the computational feasibility challenge by constructing and iteratively improving policies, and the lack of prior information for user distribution and traffic and mobility models via online estimation. While there are other works applying RL to similar scenarios [29, 30, 31, 32], they suffer from slow converge, lack of consideration towards fairness of service among users (potentially neglecting users that are not clustered together), and in some cases significant prior information requirements.

We uniquely consider the aAPs that form the aRAN as collaborative agents. The problem is modeled as a decentralized multi-agent trajectory control problem. State and reward for the proposed adapted Q-learning based solution is carefully designed to take fairness of service among users into account. The only prior information required by the system is the number of users active in the operation area, significantly relaxing the prior information requirements from Section 1.2.

Another main contribution presented in this chapter is the development and investigation of inter-agent communication schemes to facilitate improved collaboration among decentralized agents. Three different levels of inter-agent communication are investigated to observe the effects of the level of communication among agents.

1.4 Enhanced Multi-Agent Mobile Aerial RAN Deployment

Chapter 4 and works in the literature [33] focus on Markov Decision Process (MDP) based approaches where policies are defined over atomic actions, and trajectories are built by a sequence of many small actions. This results in very large search spaces, convergence rates and early performances of the solutions. However, flight control of UAVs is an orthogonal problem to optimization of service for users.

This chapter uniquely applies temporal abstraction and hierarchical RL to take advantage of temporally-extended actions of varying lengths through a Semi-Markov Decision Process (SMDP) approach. An algorithm inspired by the landmark-state methods [34, 35] for the single-agent case is developed and then extended for multi-agent scenarios. The round-robin inter-agent communication channel shown effective in Chapter 4 is utilized to pass control messages between agents.

The final contribution of this chapter is to introduce a simple planning method that takes advantage of the shared option set as a shared vocabulary together with limited inter-agent communication to improve the decentralized multi-agent collaboration effectiveness towards a single shared goal and reduce repeated-work.

1.5 Contributions

In this chapter, the main contributions of this dissertation is summarized. Specifically, this dissertation;

1. Uniquely employs variable high-order networks on modelling cellular user mobility from a CDR dataset.

2. Introduces an iterative process to decompose the intractable Joint-Optimization problem arising from placement of multiple aerial access points in 3D into tractable independent sequential placement problems.

3. Introduces a scheme that can prioritize service fairness among users in collaborative multi-agent trajectory optimization problems for aRAN.

4. Develops a scheme to utilize inter-agent communications for decentralized multi-agent collaboration in trajectory optimization problems for aRAN, and investigates the effectiveness of different levels of inter-agent communication.

5. Employs temporal abstraction and hierarchical RL to improve the convergence characteristics of the collaborative multi-agent trajectory optimization with respect to fairness problem for deploying an aRAN.

6. Employs the options framework for temporal abstraction to create a shared vocabulary among decentralized agents that can be utilized to improve planning and collaboration effectiveness.

1.6   Organization

This dissertations is organized as follows. Chapter 2 presents an analysis of a real Call Detail Records (CDR) dataset comparing mobility and network usage patterns of refugee users and non-refugee users, showing existence of groups with significantly different service demands. Chapter 3 presents a method to deploy Aerial Wireless Networks in full 3D space and provides a tractable and scalable solution through a decomposition of the joint-optimization problem into sequential individual problems. Chapter 4 then develops

a method to deploy a mobile aerial Radio Access Network and provides a Reinforcement Learning based tractable solution to the trajectory design problem for multiple aerial Access Points while significantly relaxing the prior information assumptions from 3. Chapter 5 further extends the work from Chapter 4 by incorporating a practical channel model and interference. The significantly increased solution search space size and constraints resulting from this extension is addressed by the utilization of temporal-abstraction and Hierarchical Reinforcement Learning. Finally, concluding remarks and future research directions are presented in Chapter 6.

Chapter 2: Socially Aware Cellular User Mobility Analysis

Recent advances in technology enabling collection and analysis of very large datasets has created new avenues of improvement in a large array of fields. For example, availability of vast longitudinal datasets containing the Call Detail Records (CDR) of cellular mobile network users has made it possible for extensive analysis and data driven models of human mobility offering unique insights [36, 37].

Wireless networks can strongly benefit from accurate models of human mobility, with some recent works starting to exploit available historical data [38]. Accurate mobility models for wireless networks can be exploited to improve coverage, capacity, and offloading management offering reduced costs and improved quality of service.

Beyond improvements on conventional challenges, there are new challenges facing wireless networks that can significantly benefit from more accurate mobility models and data driven insights on human mobility. Recent social upheavals and political conflicts have displaced people in unprecedented numbers in recent history. It has been documented that extensive mobile and smart phone usage among displaced people and refugees for coordination and communication among themselves as well as people they had to leave behind is common and most refugees ask for charging and wireless data services before needs that may be considered more vital such as food and water [2]. This creates a unique challenge within the context of wireless networks to provide coverage to displaced and vulnerable populations who are outside the common usage patterns wireless networks are often designed for.

In this chapter mobility networks, a significant tool in the analysis and modeling of human mobility, is introduced. Furthermore, Variable Higher-Order Networks are introduced as a

significant improvement over more common networks with first-order dependencies through better representation of sequential trajectory data that is crucial for mobility. Finally a study of real CDR dataset provided by a cellular carrier is presented with unique insights regarding the cellular usage patterns of refugee users and how they may differ from non-refugee users, and the unique contribution of the application of Variable Higher-Order Networks to the cellular user data.

2.1 Data-driven Analysis of Mobility through Networks

Study of human motion has a significant role in applications such as virology [39, 40, 41], cyber-security on mobile devices [42], urban planning [43], and traffic forecasting [44]. With the availability of data from cellular networks on mobility, estimated models of data-driven human mobility models are also beginning to be a part of modern wireless network design [38].

Network science has been the driving tool in the study of human motion [45, 46]. When combined with the large aggregate datasets and longitudinal studies, this has lead to important recent findings showing that there indeed are underlying patterns that can be discovered governing human mobility [36, 37]. Social contact networks of individuals also strongly influence their mobility, and availability of cellular carrier data that includes both location thanks to data from cellular tower and the call detail records makes it possible to study the connection between the mobility and social contact networks [47, 36].

A simple data-driven mobility network can be constructed by having each vertex represent a distinct location and each directed weighted edge between vertices represent the strength of the relationship between the vertex pairs. Edge weights can be realized as the probability of moving from one vertex to the next, or the total number of connections between the vertex pair from which the probabilities can be calculated [48, 49, 50, 51]. This network captures the mobility relationship between locations. However, this simple model presented here implicitly follows the Markov property where each state change only depends on the

current state (vertex or location in this case) and the transition probability. This results in an inability to capture information from sequential data of longer trajectories and longitudinal studies. As a result, we investigate Variable Higher-Order networks [1, 52] and apply them to cellular user mobility data.

2.1.1 Variable Higher-Order Networks [1]

Constructing networks directly from data by using the total number of connections for each vertex pair as edge weights from sequential data of movement between locations [48] is found commonly in the literature; studying urban movements of the people [50], interurban traffic [49], and flights between airports [51]. This simple construction results in the implicit assumption of Markov property where the transition to next state only relies on the current state; i.e. first-order dependency. A significant loss of information for sequential data such as movement trajectories, or data from longitudinal studies. A simple network of ports in the international shipping network [1] seen in Figure 2.1 provides an illustrative example of this first-order dependency. Which port a ship in Shanghai has originated from has no effect on the probability of whether continuing on to Seattle or Los Angeles. This Markovian dependency causes the information from higher order dependencies, such as common shipping routes, to be lost. Higher-order network illustrated in Figure 2.2 is better able to capture the sequential trajectory nature of the underlying data by introducing high-order nodes by splitting the Singapore node into two nodes depending on the originating node [1].

It has been shown that Variable Higher-Order Networks improve modeling performance in tasks of prediction, clustering, ranking [1], and anomaly detection [52].
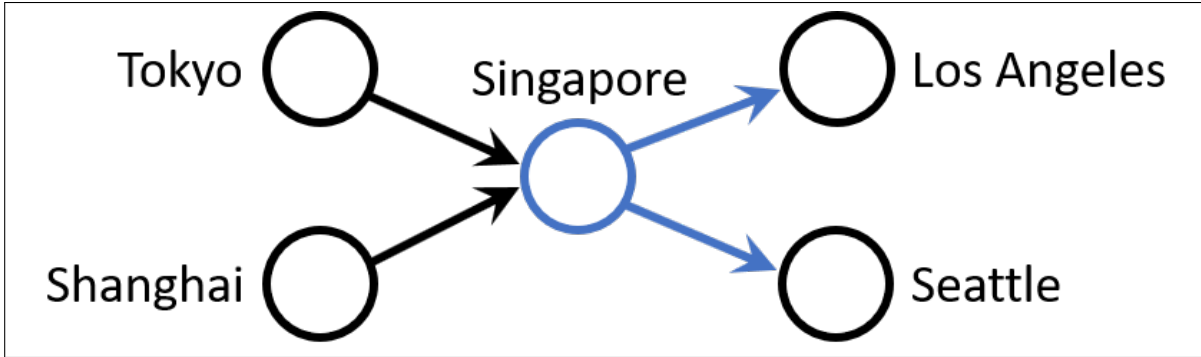
Figure 2.1: An illustrative first-order network of global ship movements. Due to the implicit Markov assumption, a ship in Singapore has the same probability to continue to Seattle or Los Angeles no matter where their origin port was.

2.2 Refugee Integration in Turkey: A Study of Mobile Phone Data for D4R Challenge

2.2.1 Motivation

Unprecedented numbers of displaced refugees, caused by social upheavals and political conflicts in recent history have overwhelmed the support systems in place for refugees and is considered to be one of the most important humanitarian crises of our times. The situation is especially critical for minors, as their experiences can have a direct influence over their mental health and educational development [54, 55]. While significant work is being done by governments and non-governmental organizations to alleviate some of these problems [56], the current state of technology allows for additional unique opportunities for determining the shortcomings and improving conditions not possible in the past.

Extensive mobile and smart phone usage by refugees for coordination and communication among themselves as well as people they have left behind has been consistently reported by officials and volunteers in the field. In fact, it has even been reported that most refugees ask for charging and data services for their mobile phones before food or water, showing that

---

[0]This section has been previously published in [53]. Permission is included in Appendix A.
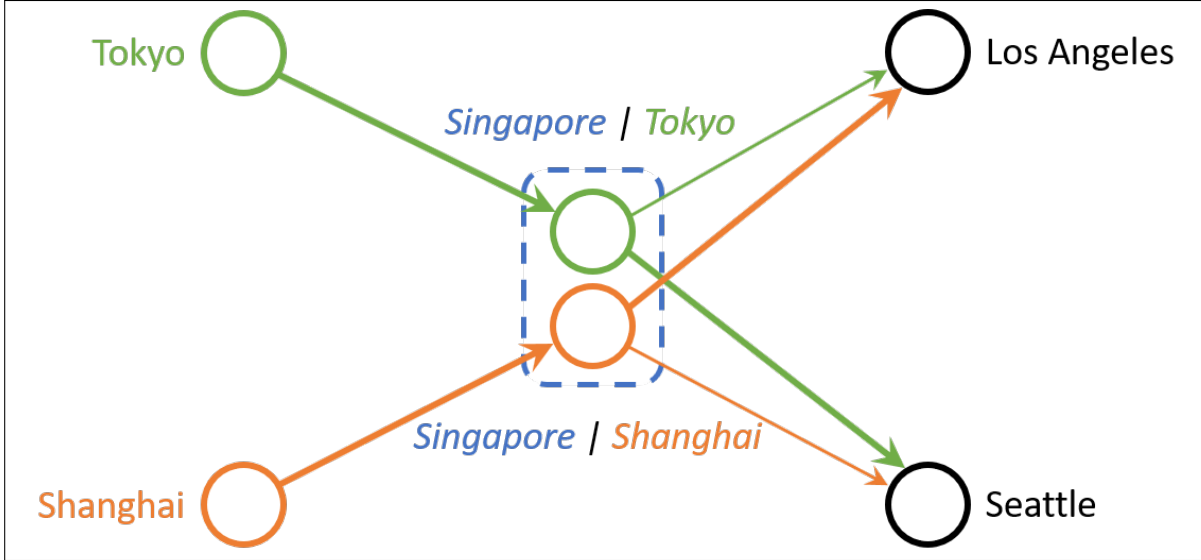
Figure 2.2: Introducing high-order nodes to Figure 2.1. Ships port of origin is represented in the high-order node for Singapore in the center, which enables to incorporate past information for the prediction of next step and representing sequential trajectory data more accurately.

they consider their mobile phones a vital tool for survival [2]. This phenomena allows for a wealth of data, uniquely presented by the current state of communication technologies, that can be leveraged.

In this work, we focus on two main issues: *social integration* and *unemployment*. We have identified these issues as our main focus, given the nature and limitations of the provided dataset. We have utilized the base station traffic data (`Dataset1`) for studying communication patterns, and coarse-grained mobility data (`Dataset3`) for studying individual and aggregate movement patterns, with the goal of identifying markers of *social integration* and *unemployment*.

2.2.2 Related Work

Human migration, both globally and locally, has been a common topic of study for scientists in various fields [57, 58]. However, recent improvements in communication technologies

and availability of mobile phone data have enabled studying movement of individuals in urban areas. Recent studies on individual mobility data show that most individuals have a predictable mobility pattern, where they travel between a relatively small number of locations regularly, such as their workplaces and accommodations [59, 60, 61].

Previous efforts on analyzing mobility patterns for past Data for Development(D4D) challenges point out the importance of mobile phone data [62, 14]. Researchers have also investigated the relationship between mobile phone usage and regional economic development [63, 64]. Information on mobility data has also been used for predicting human behavior [65] and daily pulse of cities [15, 66], studying transformation in metropolitan cities [67], and development of vaccination strategies for disease prevention [68, 8, 6].

2.2.3 Dataset

Dataset is provided by Data for Refugees: The D4R Challenge on Mobility of Syrian Refugees in Turkey. It includes anonymized mobile phone data of both refugee and non-refugee user samples. Data is provided in 3 distinct datasets and further details on the dataset can be found in the paper published by the organizers [16].

- `Dataset1`: Antenna traffic captures one year site-to-site traffic information on an hourly intervals.

- `Dataset2`: Fine grained mobility dataset contains usage information for randomly sampled accounts. Accounts selected for analysis resampled in every two weeks to prevent security concerns.

- `Dataset3`: Coarse grained mobility information is one of the most valuable resources to be able to study mobility of individuals. Trajectories of randomly selected refugees and non-refugees contain locations of serving base station and time of record for 50,000 individuals.

14
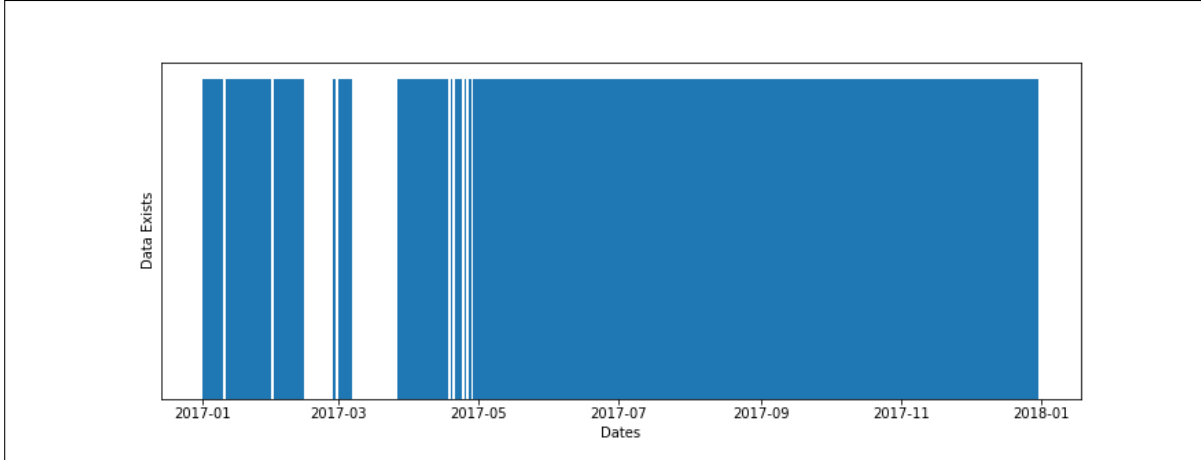
Figure 2.3: Visual representation of `Dataset3`. Illustrating dates where data exists or is missing.

2.2.4 Limitations

Dataset provided includes data collected from only one major cellular carrier. Furthermore, three different identifiers are used to flag a user as a refugee when they are registering; having an ID number given to refugees and foreigners in Turkey, registering with a Syrian passport, or using special tariffs reserved for refugees. All three of these methods are noisy, and there is no guarantee that a user flagged as a refugee is actually one. This implies that statistics of this dataset may not necessarily generalize to national statistics and care should be taken when inferring results from them.

There are significant gaps in the temporal data as well. Dates with missing data for `Dataset3` can be seen in Figure 2.3. As a result, first 6 months for `Dataset3` is discarded for reliable analysis of temporal features. This brings the total number of entries from 66,000,731 to 49,830,623, which is still sufficiently large for large scale data analysis.

`Dataset2` provides incoming and outgoing traffic data for individual users. However, traffic data is captured from different subsets of users for incoming and outgoing traffic. This means that it is not possible to study both incoming and outgoing traffic patterns for individual users in this dataset.

2.2.5 Antenna Traffic Analysis

Monthly base station traffic information is valuable for studying overall usage patterns of groups. For all the analyses found in this section, total activity between source and target stations over the entire range of observation period is aggregated together in pairs. Pairs of stations that has less than 10 activity records between them are filtered out since the effects of these stations could be considered negligible.

Locations with most and least refugee activity is identified by the fraction of refugee traffic over all traffic between every pair of stations. A histogram of the refugee traffic fractions is presented in Figure 2.4. It is observed that there are base station pairs where at least 90% of the total traffic involves refugees. Bimodal shape of the distribution indicates existence of majority refugee and non-refugee station pairs, which may be a marker of geographical segregation. Official refugee accommodation centers can be one explanation for the existence of these refugee dominated base station pairs. The locations of Disaster and Emergency Management Presidency (AFAD) Temporary Protection Centers[1] (TPC) are show in Figure 2.6, together with the base station pairs that fall within top and bottom %10 percent on Figure 2.4. It is observed that the TPCs are not sufficient to explain all of these refugee dominated pairs, especially in central and western parts of the country.

Distances between base stations in these refugee and non-refugee dominated station pairs are another topic of interest. Figure 2.5 shows the distribution of inter-pair distances of base station pairs that are dominated by refugee and non-refugee activity respectively. It can be observed from Figure 2.5 that the base station pairs dominated by refugee activity tend to be closer to each other compared to their non-refugee dominated counterparts.

Geographical spread of majority refugee and non-refugee traffic can also be observed by visualizing the traffic between base stations over a map. Figure 2.6 shows the traffic between

---

[1]Compiled on December 6th, 2018 from reports found in: https://www.afad.gov.tr/tr/2374/Barinma-Merkezlerinde-Son-Durum
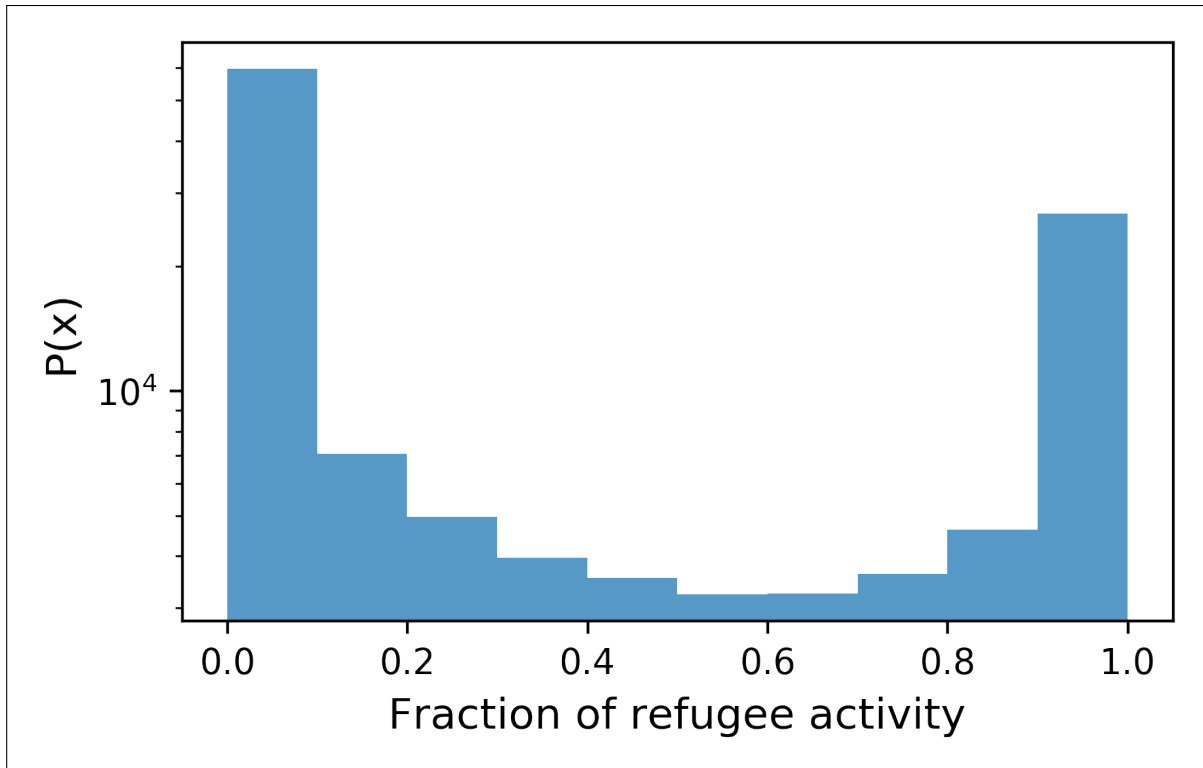
Figure 2.4: Histogram of refugee activity density for pairs of stations.

the base stations that fall within the top and bottom 10% of Figure 2.4, respectively. Only the voice call counts data is displayed on the maps for the sake of brevity, however both voice call duration and SMS data also show very similar results. It can be observed from Figure 2.6 that, base station pairs with refugee dominated traffic are less spread out geographically, and are mainly focused between certain centers of dense activity. On the other hand, non-refugee dominated cellular traffic appears to be spread out much more evenly.

Since Istanbul is one of the largest metropolitan areas in the world, it merits a closer look. Figure 2.7 displays the same data as Figure 2.6, only within the city limits of Istanbul. It can be observed that, refugee dominated traffic is similarly less spread around. This divide can most easily be observed on the Asian side of the city, where refugee dominated traffic is lesser than its non-refugee dominated counterpart.
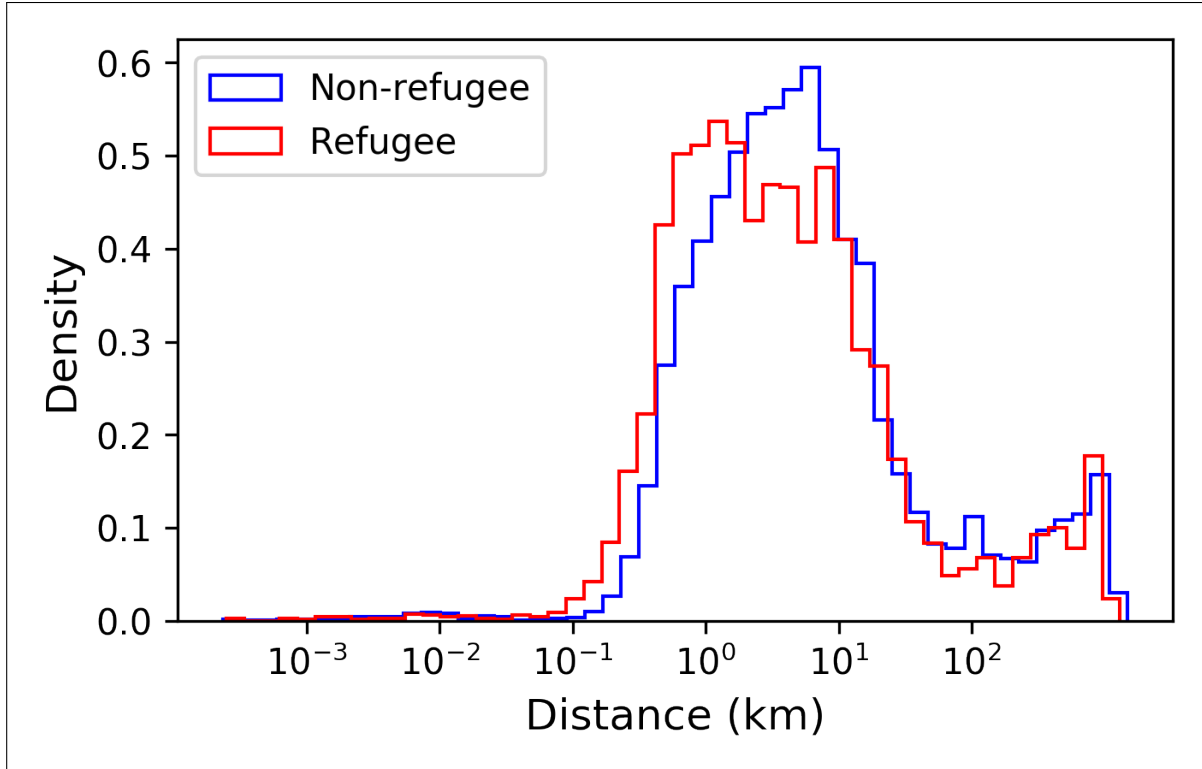
Figure 2.5: Distance distribution of refugee and non-refugee dominated base station pairs.

This difference between mobile phone usage behavior and geographical spread of mobile phone traffic suggests a lack of social integration for refugee users. Figure 2.6 shows that refugees are not in contact with a significant portion of the nation. This, together with the bimodal nature of Figure 2.4 also suggests existence of spatial segregation, which will be further investigated in the following sections.

2.2.6 Movement Analysis from Coarse-grained Mobility

Coarse-grained mobility data from `Dataset3` provides high temporal resolution and continuity for each user. Therefore, it is especially suitable for the analysis of individual movement patterns over time.
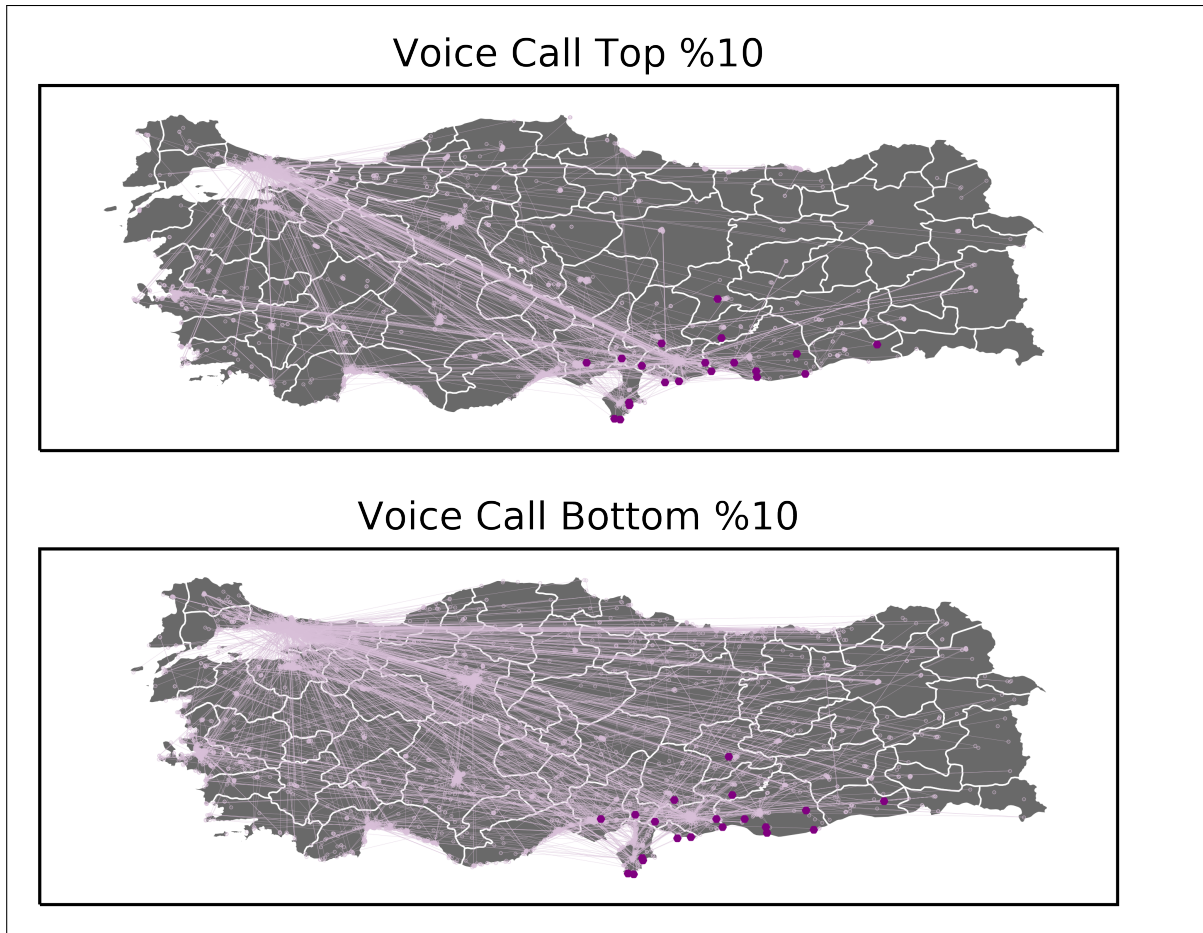
Figure 2.6: Voice call traffic between base station pairs that fall within the top and bottom 10% of Figure 2.5. Locations of AFAD Temporary Accommodation Centers are marked with purple hexagons.

2.2.6.1   *High-order Mobility Network Analysis*

A user mobility network for both refugee and non-refugee users are constructed. Traditionally, mobility networks are constructed with a first-order assumption. This means that, probability of a user moving to a certain node is assumed to be only dependent on the current node they are in. It has been shown that this single order assumption is lacking in representing complex mobility data [69]. Therefore, a variable order high-order network is
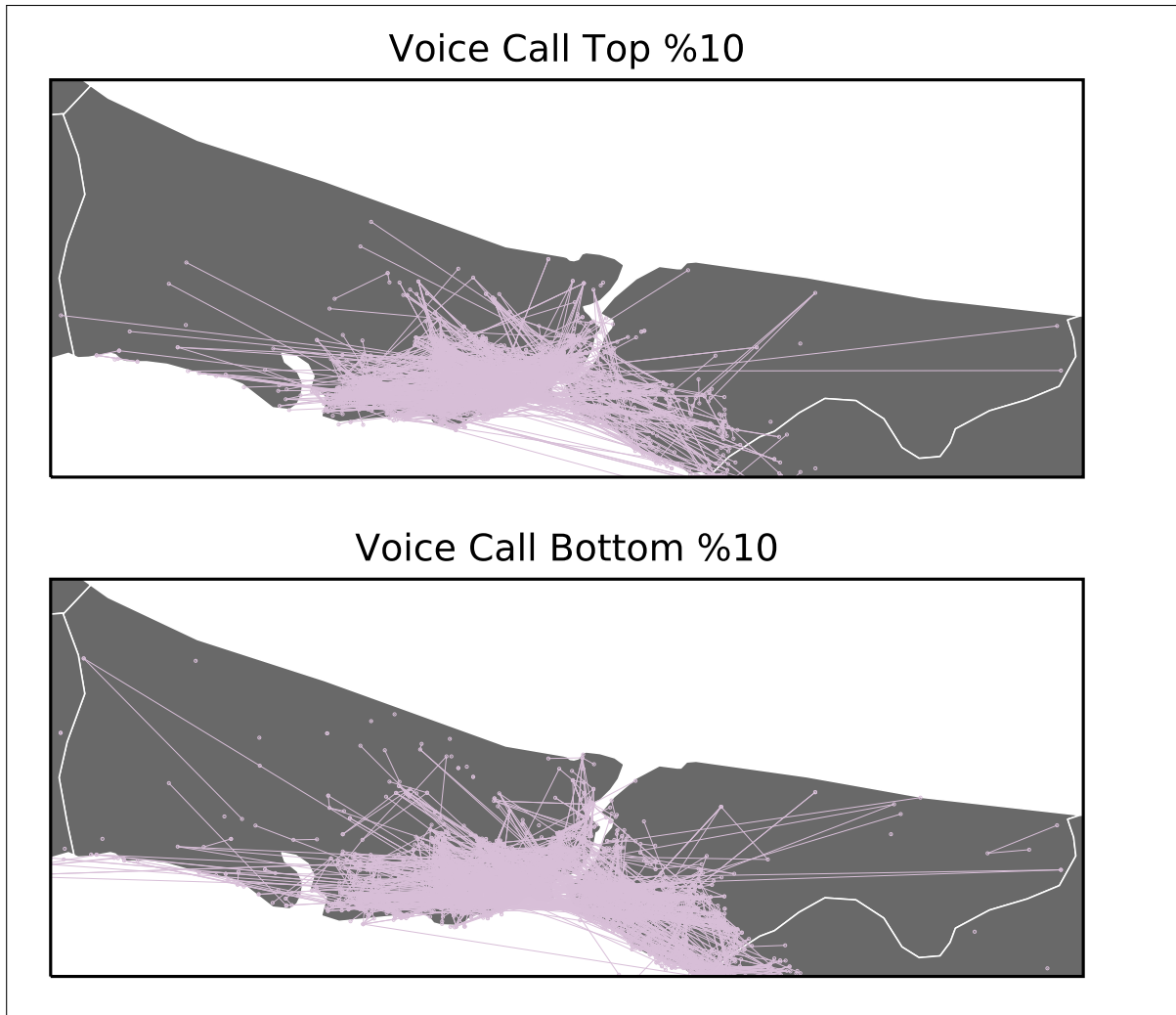
Figure 2.7: Voice call traffic between base station pairs that fall within the top and bottom 10% of Figure 2.4. Data is confined within the city limits of Istanbul.

generated for analysis in this section, using `BuildHON+`[2] algorithm [70]. Table 2.1 provides some details on the dataset and the constructed variable higher-order mobility networks.

Pagerank [71] is a widely used algorithm that is used to rank the importance of a node in a network, calculated based on incoming and outgoing links to the node. Intuitively, pagerank of a node in a mobility network will show how many paths include that node, and how often users take these paths. A node with a high pagerank in a mobility network can be

---

[2]Software implementation: https://github.com/xyjprc/hon

Table 2.1: The dataset and basic properties of the constructed variable higher-order mobility networks.

	Refugee Users	Non-Refugee Users	Total
Dataset			
Data Points	20483496	29347127	49830623
Unique Users	33270	16597	49867
Unique Locations	663	847	925
Mobility Networks			
Order (Node count)	11583	18243	N/A
Size (Edge count)	37934	59779	N/A

viewed as a hub that is visited by users from all over. Some good examples of high pagerank nodes will be dense residential areas and public transportation transfer stations.

A visualization of district rankings over the generated variable high-order user mobility networks, based on pagerank values, can be seen in Figure 2.8. First of all, we would like to discuss how to interpret this figure. Calculated pagerank values are, by definition, dependent on network properties such as size, and are used to compare nodes within the same network. The mobility networks calculated are of different sized for the two groups, due to some locations never being visited in the dataset, the pagerank values are not comparable between these two networks. The intent of this figure is to compare the spatial distribution of high-pagerank nodes from both networks, which can inform us about spatial segregation.

First observation that can be made is that the size of the mobility network for refugee user mobility is smaller. This is because there are zones that are never visited by any refugee users, meaning that refugee movement is confined to a more limited set of locations compared to non-refugee users. Furthermore, regions of importance appear to be more spread out in the non-refugee users mobility network, while important regions are more focused in smaller regions for refugee users. Finally, while non-refugee users mobility network has several distinct centers of high importance, network for refugee users is dominated by İstanbul.
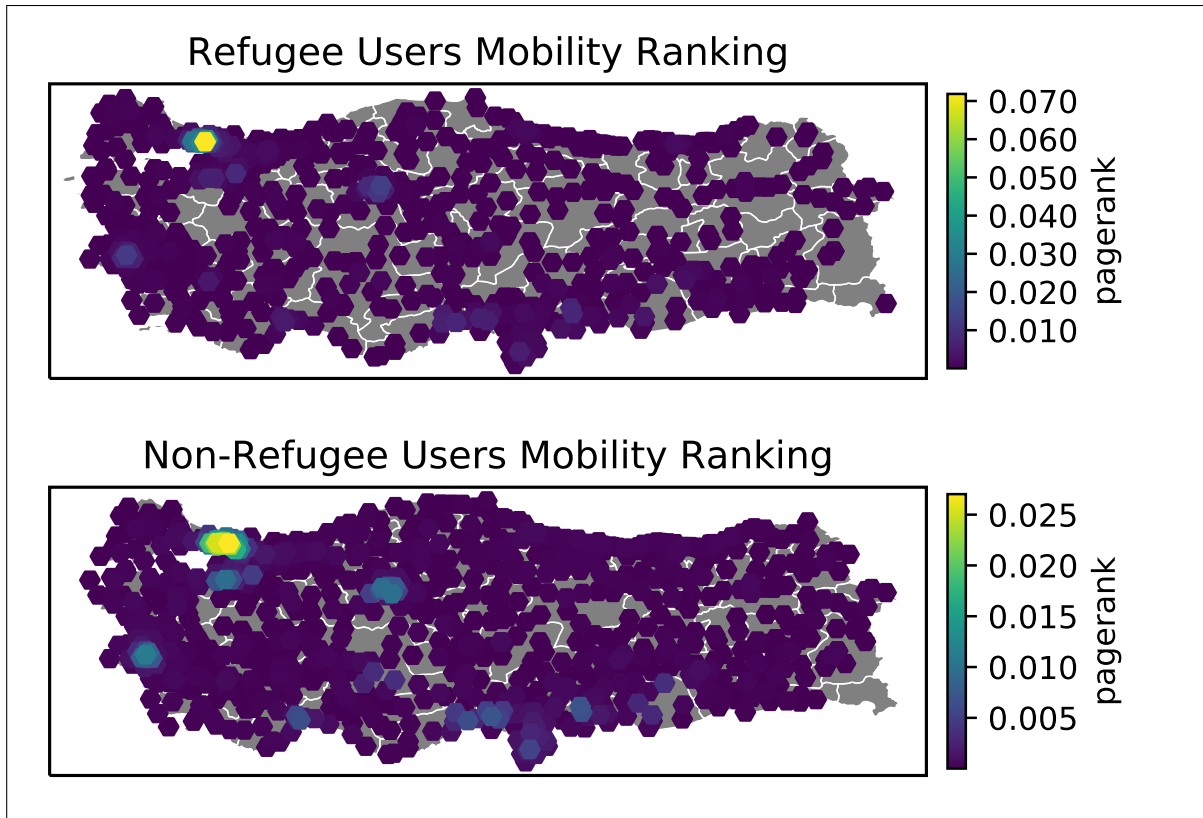
Figure 2.8: Visualization of district rankings based on pagerank values over high-order user mobility networks.

Last two points merit a closer look into the districts within İstanbul, which is one of the worlds largest metropolitan areas.

It can be seen from Figure 2.9 that, even within the city limits of İstanbul, mobility of refugee users is still confined to focused and denser regions, as opposed to the almost evenly spread out non-refugee users. For example, there is very little movement by refugee users on the Anatolian side (on the right) of the İstanbul compared to non-refugee users. This implies that starting out from any node in İstanbul on the refugee users mobility network, there is a high probability of eventually ending up in this small high-pagerank area. It can be inferred that, while refugees and non-refugees share the same urban public spaces, they are segregated in their use of these spaces. This suggests that a significant portion of the
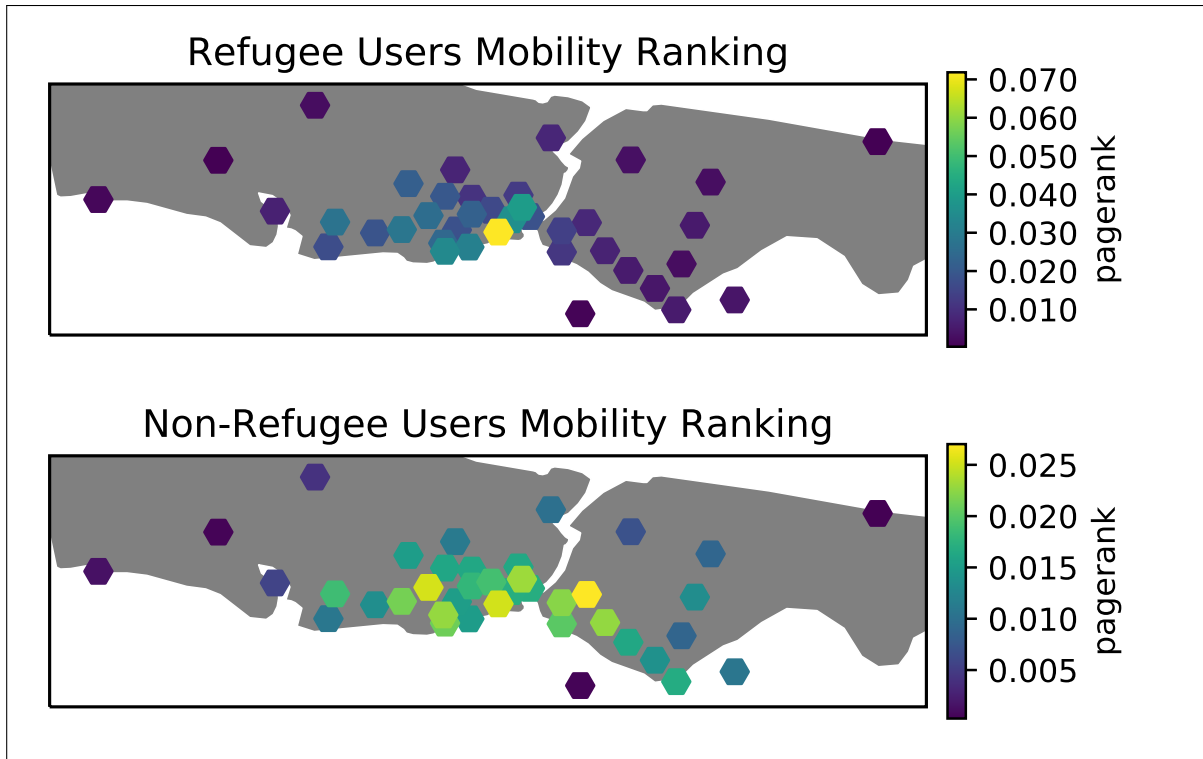
Figure 2.9: A closer look into the districts within the city limits of Istanbul from Figure 2.8.

refugee users are not integrated in the local social life. It can also be an indicator of differing employment statistics and employment opportunities available to refugee users.

2.2.6.2   Inter-event Analysis

Data from incoming and outgoing traffic is combined for movement analysis, and every time a location change for a user occurs it is marked as a transition event. A normalized histogram for the transition event count of each user can be seen in Figure 2.10. This figure shows that a refugee user is more likely to have a smaller number of transition events compared to a non-refugee user.

The average time each user waits before a new movement event occurs can also be enlightening. A normalized histogram for inter-event time averaged for each user is provided in
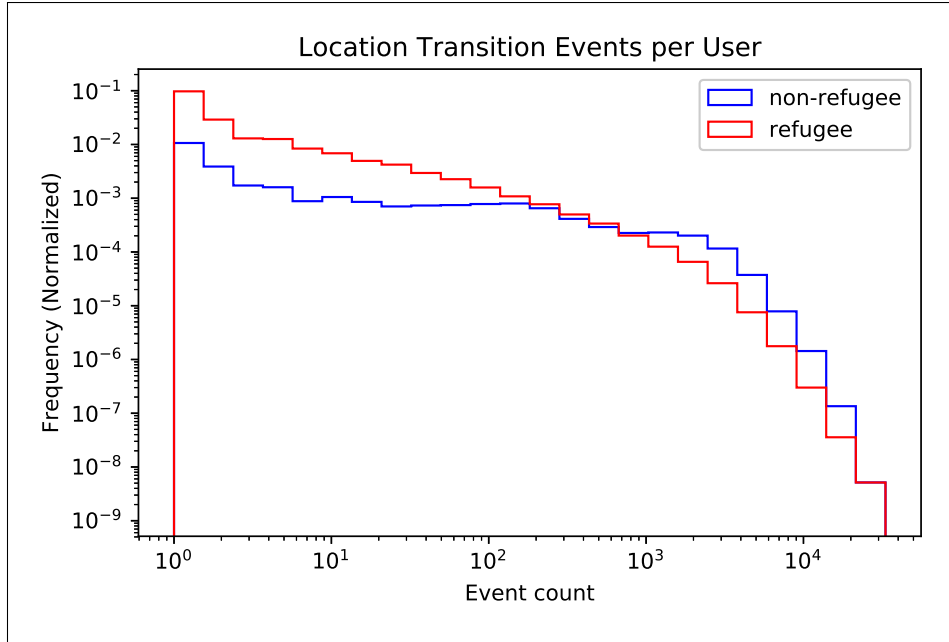
Figure 2.10: Histogram of transition events on log-log scale. Marks every time a user changes locations.

Figure 2.11. This figure shows that a refugee user is more likely spend a shorter amount of time at each location compared to a non-refugee user, moving more frequently. Figure 2.11 also shows a significant number of refugee users, with a mean inter-event time under 10^3 seconds, or 17 minutes. This can be explained by spending time in moving vehicles or between regional borders, where transition-events can occur more frequently or with smaller movements.

Distance traveled between each transition event will help paint a more complete picture of the user movement, when combined with the previous results. A histogram of average distance traveled by each user is given in Figure 2.12. This figure shows that a refugee user is more likely to travel shorter distances between transition events compared to a non-refugee user.

Finally, we will take a look at the size of the overall movement regions of each user. Movement bounding-boxes are computed by calculating the smallest rectangle that encompasses
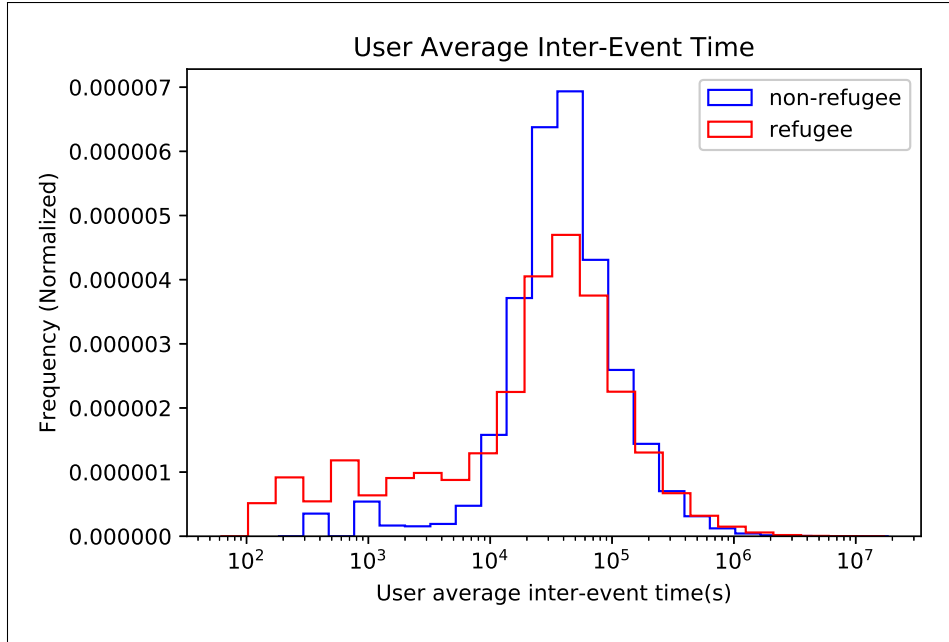
Figure 2.11: Histogram of the average time elapsed between subsequent transition events for each user.

the center coordinates of every region each user has visited over the observation period. The geographical distance between diagonal corners of the bounding box is used as an indicator of the size of the overall movement regions for each user. A histogram of this bounding-box diagonal distances can be seen in Figure 2.13. This figure shows that a refugee user is more likely to travel within a smaller region compared to a non-refugee user, confirming the previous observations.

Looking at the results in this subsection, it will be possible to infer that refugee users move more frequently in a smaller regions via short distances. Individual movement is expected to be significantly regular, where a person frequently returns to a small number of anchor locations, such as work and home [72]. These observations can suggest that refugee users are more likely to be unemployed, or hold irregular jobs that require them to frequent larger than usual number of locations.
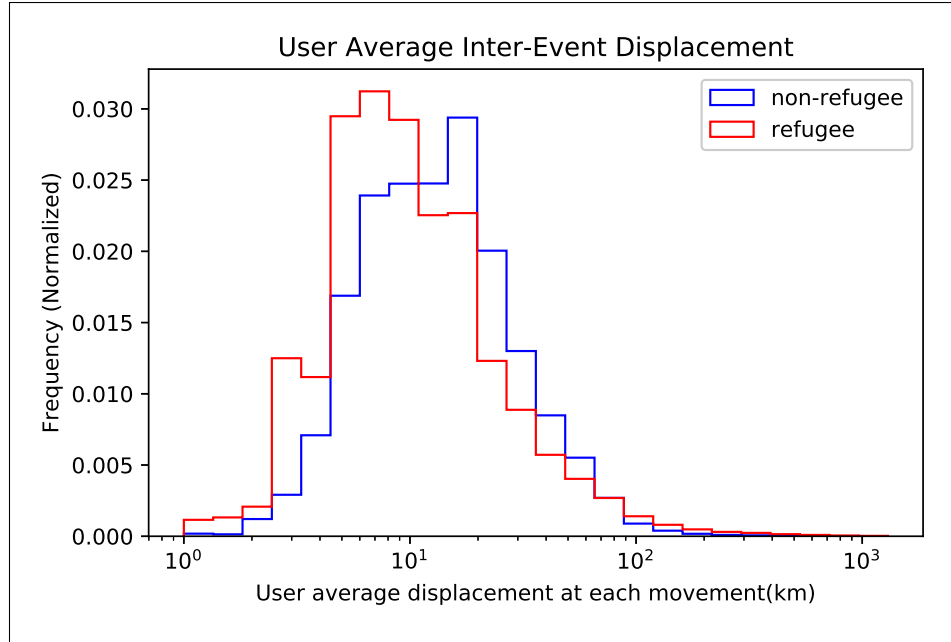
Figure 2.12: Histogram of mean distance traveled between subsequent transition events for each user.

2.2.7 Discussion

In this work, mobile phone data belonging to refugee and non-refugee users have been analyzed with a focus on discovering individual and aggregate mobility and connectivity patterns. Since connectivity is linked with employment and social life, and human mobility is very regular and dependent on a few frequented locations, such as places of work and residence, any irregularities on these patterns can be interpreted as indicators of employment and integration issues.

Connectivity analysis showed a divide between base station pairs that are servicing majority refugee and non-refugee traffic. It is also found that the distance of refugee connections tends to be shorter. Visualization over a map also showed a much smaller geographical spread of refugee traffic compared to its non-refugee counterpart. These observations suggest that refugee and non-refugee users are segregated geographically and refugee users do not interact with a significant portion of the nation.
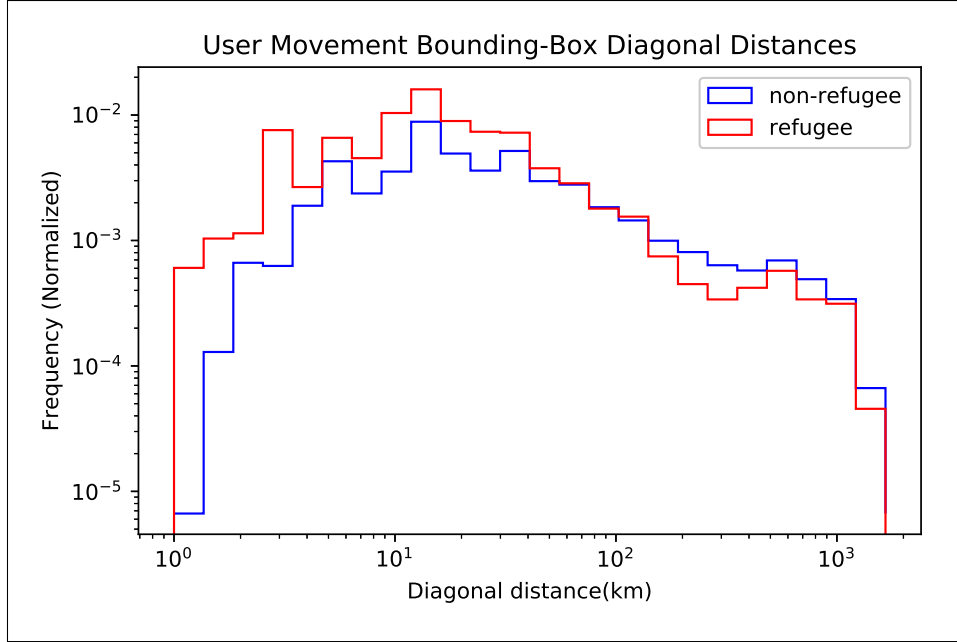
Figure 2.13: Histogram of mean distance traveled between subsequent transition events for each user.

User mobility network analysis yielded the results that refugee users move around in smaller regions, and movement of refugee users is less spread out compared to non-refugee users. Focusing on a metropolitan area, Figure 2.9 showed that refugee movement is confined in a smaller and denser area. This supports that, while the urban areas and public spaces are available to both groups, use of these spaces is segregated. Furthermore, results from inter-event analysis also support these findings, as they suggest that refugee users move more frequently in smaller regions and with shorter distance steps.

Combining results from these analyses, it is possible to infer that refugee users are largely not integrated into the local social life of the regions they reside in. It is likely that, this integration issue is driven by the socioeconomic conditions and especially availability of employment options, which focuses refugee users into areas where there is work available for them. Providing suitable employment opportunities outside the current centers of focused

refugee activity can encourage and speed up the distribution of refugees more evenly within the nation and facilitate their integration to the local societies.

Chapter 3: Aerial Wireless Network Deployment

Aerial platforms bring a new dimension to wireless network design enabling highly reliable, cost-effective networks that can be easily deployed in areas otherwise inaccessible to traditional approaches. Considering the mobile heterogeneous nature of the next generation traffic demand, the mobility and availability brought by the use of aerial platforms represent considerable advantages.

Industry is also quickly adapting aerial platforms in wireless networks; Alphabet's LOON and AT&T's Flying Cell-on-Wheels (COW) are two of the high profile examples for providing wireless service using aerial platforms. Alphabet's Project WING and Amazon's Air Delivery on the hand are examples where aerial platforms act as user equipment subscribing to existing networks.

Disaster recovery is another important application of aerial mobile wireless networks to restore service to the areas struck by disasters and support ongoing relief efforts. As an example, AT&T's Flying COW have been experimentally deployed in Puerto Rico on the aftermath of Hurricane Maria, providing service to the affected users while terrestrial infrastructure is out-of-order.

Aerial platforms can be divided into two depending on their operation altitude; high altitude platforms (HAPs) and low altitude platforms (LAPs) [33]. We should also note that operations altitudes are highly dependent on local regulations, and relevant regulations must be checked at the area of deployment.

- High Altitude Platforms (HAPs) operate at altitudes of 17 km or higher, and are often stationary. HAPs are in general much more difficult to deploy and are used as long-term platforms.

- Low Altitude Platforms (LAPs) can operate at altitudes as low as several meters and they are often limited to altitudes of a few kilometers. They generally posses much higher mobility compared to HAPs and can be deployed much more easily and temporarily. The name Unmanned Aerial Vehicle (UAV) often refers to low-altitude aerial platforms.

3.1 Unmanned Aerial Vehicles (UAVs) in Aerial Wireless Networks

There are several main use cases of low altitude Unmanned Aerial Vehicles (UAVs) in wireless network scenarios found in the literature, which will be explored in following subsections [33].

3.1.1 Aerial Base Stations

Radio equipment mounted on UAVs can serve as basestations. Aerial base stations can be used to support terrestrial networks in coverage and capacity, and provide this support on demand with a quick deployment. This allows them to be used for temporary traffic offloading during high-traffic events such as concerts [73, 74, 75]. Other potential uses of aerial base stations include public safety networks [76], information dissemination in device-to-device networks [77, 78] relaying taking advantage of the LoS dominated channel, and IoT applications [79, 80, 81, 82].

3.1.2 User Equipment

UAVs are employed in many different applications ranging from surveillance to delivery. Reliable, low-latency communications are crucial to support the many applications of UAVs, and one of the most practical ways to achieve this is UAVs subscribing to existing terrestrial cellular communication networks [83]. Some application scenarios like surveillance will also demand a high-throughput channel beyond the reliable, low-latency communications necessary to control the UAVs.

3.1.3 Flying Ad-Hoc Networks (FANET)

FANETs can be employed to deliver service to areas where a traditional terrestrial network deployment is not feasible thanks to their mobility and decentralized operation [84]. Other applications of FANETs include surveillance, disaster response, relaying, and remote sensing [84, 85]. A FANET consisting of smaller UAVs also have advantages of scalability, survivability, and cost advantages compared to a single, more complex UAV [86].

3.2 Presented Scenarios

We are uniquely considering low-power, short-range wireless Access Points mounted on UAVs. These aerial Access Points (aAPs) do not have enough coverage to cover the entire operation area at one point in time and therefore must stay mobile to satisfy the QoS requirements of all users, creating an aerial Radio Access Network (aRAN) that is connected to all users over time.

Before moving on to the problem of trajectory optimization for these mobile aAPs, we are first tackling the sub-problem of static deployment. We provide a sequential optimization method that makes the otherwise intractable joint-optimization problem of placing multiple aAPs tractable.

3.3 Efficient 3D Placement of Access Points in an Aerial Wireless Network

3.3.1 Introduction

Aerial platforms for wireless networks have been receiving a lot of attention recently as an option to augment cellular networks for the demands of the next generation. High altitude platforms deployed in stratosphere had been a popular option for researchers and private companies alike to bring connectivity to regions lacking in infrastructure [88, 89]. Thanks to the recent advancements in unmanned aerial vehicles (UAV), low altitude, mobile UAVs have also started to be viable options for wireless network design to rapidly provide network services in a wide range of scenarios [90, 91].

The use of aerial platforms makes altitude available as a new dimension to exploit, however efficient deployment of short range, low power aerial wireless networks in 3D space has been only developed in a limited scope. While the placement of basestations on a 2D plane is well studied, it is not trivial to extend the results from 2D to 3D, and a direct extension may not even exist [18]. This 3D placement problem can be related to the densest sphere packing problem, also known as the Kepler Conjecture, which has been formally proven very recently by a large team after remaining as an open problem for 400 years [19]. There exists a relevant body of previous research on the 3D deployment problem in the field of wireless sensor networks [92], especially for underwater acoustic sensor networks [93, 94]. However, the results from this field are not directly applicable due to deployment constraints and application goals being significantly different compared to traditional wireless communication networks.

Recent works have considered the use of UAVs as aerial basestations for cellular networks under various scenarios. Cases are considered for a single UAV basestation [20, 21], backhaul aware single UAV basestation [22], two UAV basestations with interference [24], arbitrary

---

[1]This section has been previously published in [87]. Permission is included in Appendix A.

number of UAV basestations utilizing a joint heuristic optimization procedure [25] and optimal transport theory [26], and cooperation between UAVs [95] exist. Another recent effort considers a single UAV under realistic channel conditions [23]. In [96], small cell association with flying network platforms for centralized radio access networks is explored. Assisting public safety communications through UAV basestations is studied in [97]. However, most of the previous works towards the 3D deployment simplify the problem to a circle coverage problem on a 2D surface. This simplification is justified based on a stochastic air-to-ground channel model [98], where the coverage area is modeled as a circle on the surface whose radius is dependent on the altitude, and area statistics. A recent study shows that this model is insufficient in dense urban areas where the altitude of an aerial platform may be lower than building heights [99].

Furthermore, the focus on cellular basestations places limits on the applicability of UAV platforms, especially for a post-disaster scenario. For example, AT&T's flying cell-on-wheels (COW) that supplied services for the first time in the aftermath of Hurricane Maria in Puerto Rico relies on a constant tether connection with ground for power. It should also be highlighted that short-range communications will not only reduce the payload and battery demand on the UAVs, but will also lead to compatibility with a wider range of user devices and provide better power efficiency for them. Under scenarios such as post disaster recovery where infrastructure have failed, or bringing communication capabilities to areas where there is no infrastructure in the first place, this can be an important consideration. After major disaster events, such as earthquakes, tsunami, and hurricanes, serious damage can be expected on power, communication and transportation infrastructure. Restoration of these infrastructure can take a significant period of time and being able to rapidly establish a communication network with minimal reliance to the pre-existing infrastructure can help rescue and recovery efforts.

In this work, we are proposing an initial formulation for the genuine 3D placement of multiple low-altitude short-range aerial APs directly, utilizing concepts from the analysis of random graphs. An efficient iterative algorithm is then proposed as a practical solution for the optimal placement problem, that can respond to network dynamics and determine the adequate number of APs. Finally, we provide results through an agent based simulation on uneven terrain with mid- or high-rise buildings.

3.3.2 Formulation

In this section, we define the problem for efficiently placing APs in 3D space with the goal of connecting existing users to a single communication network.

3.3.2.1 Network Architecture

In the considered network architecture, aerial 3D APs are deployed to provide users access to a radio access network (RAN). Aerial 3D APs and the RAN together is named 3D-RAN. 3D-RAN can then forward data to an application server or any other data consumer through a backhaul connection. This paper considers only the deployment of the aerial 3D APs, and the RAN and backhaul connections are assumed to be available. We consider two different types of nodes on the network, user devices and APs. Without loss of generality, all user devices and APs are assumed to have identical communication specifications within their respective groups. The maximum number of available APs is assumed to be provided as a constraint. Initial locations of every user is assumed to be known. For the time being, we temporarily do not take fading and interference into consideration and assume that all antennas are isotropic, resulting in ideal spherical communication ranges. The radio access for user devices is assumed to be an ideal multi-hop mesh network. Each AP is assumed to a part of a 3D-RAN with backhaul infrastructure. It is assumed that every user is equipped with a longer-range device such as WiFi, SMS, mobile Internet and/or a lower-power short-

range device such as Bluetooth. A diagram of the proposed network architecture can be seen in Figure 3.1.
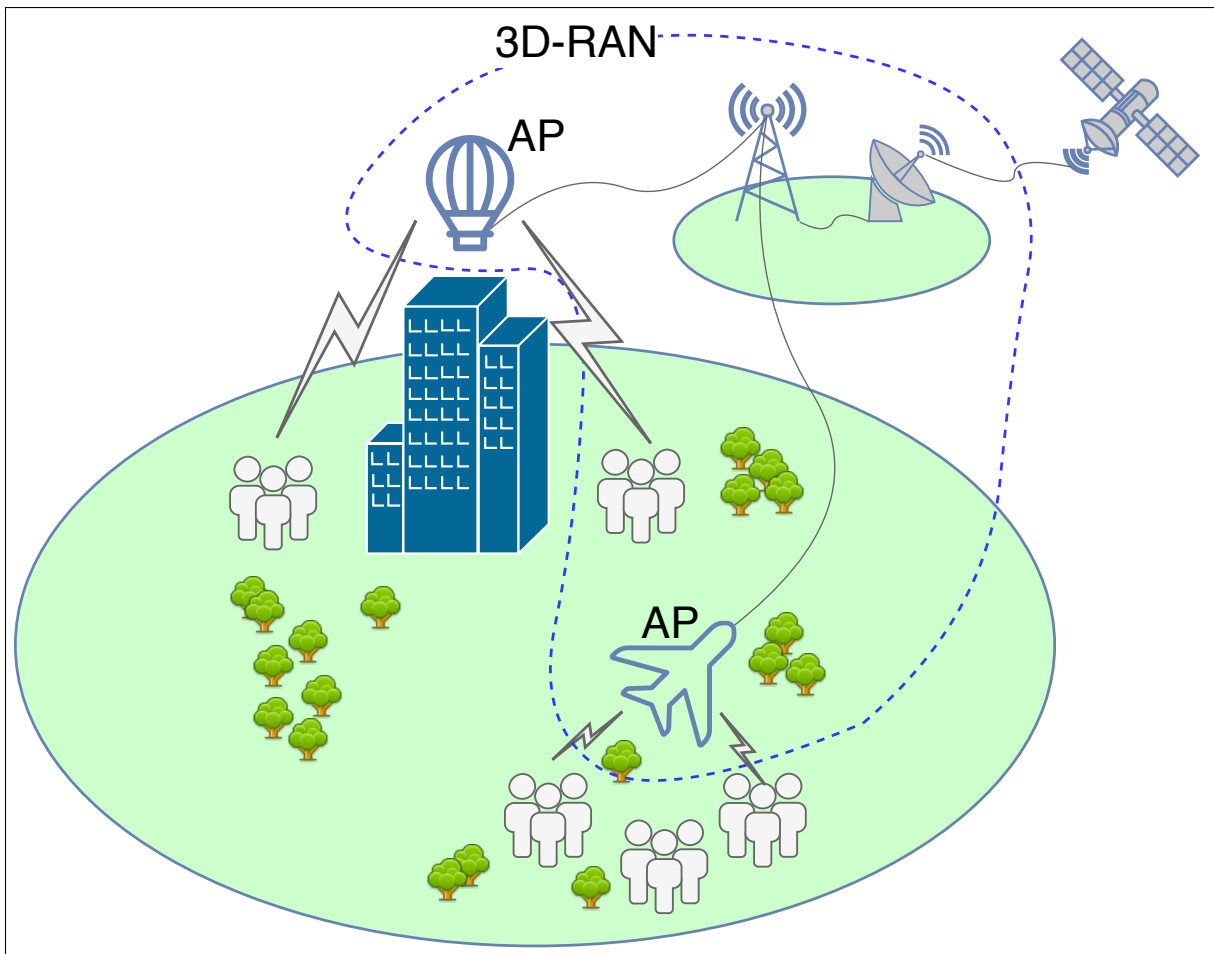
Figure 3.1: Diagram for the proposed network. AP designates 3D deployed aerial APs providing access to a RAN, together denoted as 3D-RAN. Users are shown as silhouettes.

3.3.2.2   3D Graphical Model

Such an aerial wireless network can be represented by a graph toward our purpose. Let $\mathcal{M} = \{m_1, m_2, \ldots, m_i, \ldots, m_N\}$ and $\mathcal{B} = \{b_1, b_2, \ldots, b_i, \ldots, b_K\}$ be the sets of vertices, where m_i and b_i represent a user device node and an AP respectively. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_i, \ldots, u_{N+K}\}$ be the set of all vertices, where $\mathcal{U} = \mathcal{M} \cup \mathcal{B}$, meaning that

u_i is either a user node or an AP, and N and K are the total number of existing user device nodes on the network and the total number of available APs, respectively. Let $\mathcal{C} = \{\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_i}, \ldots, \mathbf{c_{N+K}}\}$ be the set of coordinates, where $\mathbf{c_i} = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}$ is the location of the vertex u_i in the 3D cartesian coordinate space, $\mathbf{c}_i \in \Re^3$.

Let the arbitrary probability density function (PDF) $f_c(\mathbf{c})$ be the distribution that defines the initial placement of the user nodes $\mathcal{M}$ in $\Re^3$, giving the probability of a user node existing at a coordinate $\mathbf{c}$. Since full information is assumed on the initial location of every user, $f_c(\mathbf{c})$ is assumed as known initially and remains the same during the execution of the placement algorithm.

Let $f_{\text{vol}}(u_i)$ define the volume that is within communication range for vertex u_i in $\Re^3$, where the vertex represented by u_i can communicate with any other vertex that exists within $f_{\text{vol}}(u_i)$. Communication volumes are assumed to be ideal spheres, and therefore $f_{\text{vol}}(u_i)$ is a sphere with the radii r_i, where r_i is the communication radius for vertex u_i. Communication volume then can also be written as $f_{\text{vol}}(\mathbf{c_i}, r_i)$, which puts a sphere of radius r_i centered on the vertex coordinates $\mathbf{c_i}$. Since all users and APs are also assumed to be identical within their groups, it can be further simplified to $r_i = r_{\text{M}}, \forall i \implies u_i \in \mathcal{M}$ and $r_i = r_{\text{B}}, \forall i \implies u_i \in \mathcal{B}$ respectively.

Extending common ad hoc networking research [100, 101], let G be the modified random geometric graph [102], where two vertices are connected by an edge if and only if one is within the others communication volume, defined by the generative function $\text{RGG}(\mathcal{U}, f_{\text{vol}}(u_i))$. Figure 3.2 depicts the modified 3D RGG and associated parameters that are used in this paper.

3.3.3 Optimization of 3D Placement

In this section, we explore the optimal 3D placement of APs. For K APs, this problem can be represented as a search in $3K$ dimensional space, $\Re^{3K}$. Combined with perturbations such as environmental obstacles, network dynamics, and stochastic variables, a straight-
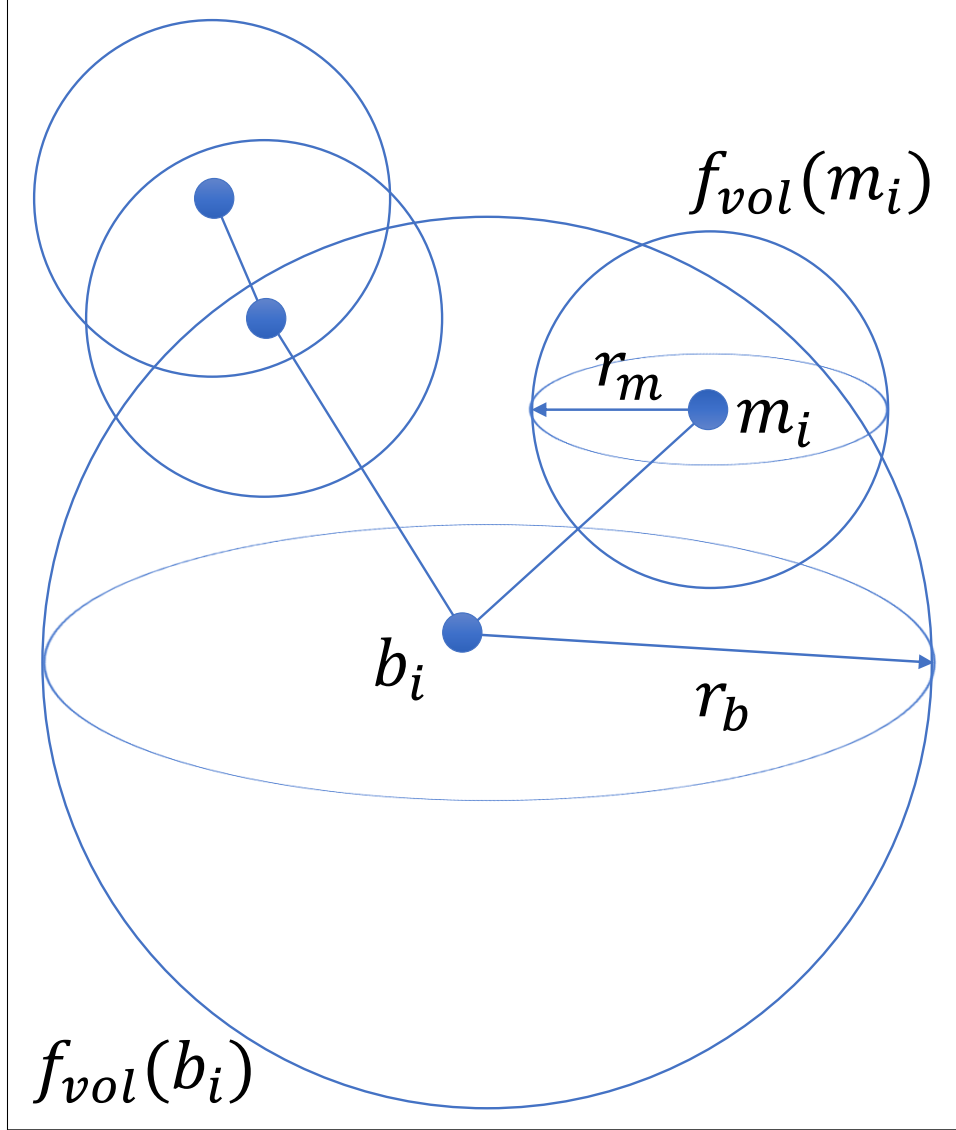
Figure 3.2: A graphical visualization that shows vertices and edges for $K = 1$ and $N = 3$

forward solution becomes unfeasible even for modest values of K. Instead, we employ RGGs, which lend itself well to wireless networks and historically have been used to solve similar problems [100, 101], to make this 3D AP deployment problem more tractable.

The final goal of the 3D AP deployment optimization is to achieve radio access functionality with the smallest number of APs such that all user nodes on the ground of different heights can connect to this 3D-RAN infrastructure. If it is not possible to connect all user

nodes to this 3D-RAN, then the goal is to achieve the highest fraction of users connected to the network with the available number of APs. Two nodes being connected with an edge on the RGG G means that they can communicate with each other, according to the model we have developed in the previous section. Therefore, the set of users that can connect to the 3D-RAN infrastructure is equivalent to the user nodes, $m \in \mathcal{M}$, that are connected to the AP nodes, $b \in \mathcal{B}$, in the RGG G. The goal function, $g(\mathcal{B})$, therefore can be defined as the fraction of user nodes that are connected to the given set of APs on the RGG G that represents the wireless network.

Let $\text{path}(u_i, u_j)$ be a simple path between u_i and u_j, denoting a sequence of edges and vertices that connect u_i and u_j to each other, where every vertex on the path is distinct from each other, and the length of the path is denoted by $|\text{path}(u_i, u_j)|$, defined as the number of edges on the path. Let $\mathcal{Q}_{k,h}$ be the set of user nodes, for which a path of length h, i.e. h-hops, exists to the AP b_k, defined as $\mathcal{Q}_{k,h} \triangleq \{m_i : \forall i \implies \text{path}(m_i, b_k) \text{ exists where } |\text{path}(m_i, b_k)| = h\}$. Let H be the maximum number of hops permitted for a transmission, and $\mathcal{Q}_k^H$ be the set of all user nodes that are connected to b_k with a path length of at most H. $\mathcal{Q}_k^H$ can then be written as

$$\mathcal{Q}_k^H \triangleq \bigcup_{h=1}^{H} \mathcal{Q}_{k,h}. \tag{3.1}$$

Let $\text{N}_G(u)$ give the neighborhood of vertex u on graph G, which means that $\mathcal{Q}_{k,1} = \text{N}_G(b_k)$. Therefore, (3.1) can be computed as

$$\mathcal{Q}_k^H = \left(\bigcup_{h=1}^{H-1} \text{N}_G(\mathcal{Q}_{k,h}) \right). \tag{3.2}$$

Since the purpose of AP placement is to connect largest fraction of users nodes possible, $g(\mathcal{B})$ can be written as

$$g(\mathcal{B}) = \frac{|\bigcup_{k=1}^{K} Q_k^H|}{N}. \tag{3.3}$$

We propose two approaches to optimize the goal function. First approach is the joint optimization for all the APs at once, which is the most straightforward solution, albeit impractical for large K. Second approach is making an assumption of considering placement problem for each AP as mutually independent. We consider both cases in following sections and start with calculating certain probabilities and graph properties that will be useful.

3.3.3.1 Formation of Optimization

This formulation uses the excellent research from [100] and [101] as a starting point. This analysis considers a single-hop network, $H = 1$. Let u_i be any vertex, $u_i \in \mathcal{U}$, placed at coordinates $\mathbf{c_i}$. Let m_j be any user node, $m_j \in \mathcal{M}$, placed at coordinates $\mathbf{c_j}$ according to the PDF $f_c(\mathbf{c})$. Vertices u_i and m_j are connected with an edge if $\mathbf{c_j}$ falls within the communication volume $f_{\mathrm{vol}}(u_i)$, or vice versa. Let $f_{\mathrm{maxvol}}(u_i, u_j)$ give the larger communication volume between u_i and u_j, $f_{\mathrm{maxvol}}(u_i, u_j) \triangleq f_{\mathrm{vol}}(c_k, r_k)$, where $k \triangleq \arg\max_i(r_i, r_j)$. Therefore, the probability that an edge exists between an arbitrary vertex u_i and a user m_j can be written as (3.4).

$$P_v(u_i, m_j) \triangleq P_0(u_i) = \iiint\limits_{f_{\mathrm{maxvol}}(u_i, m_j)} f_c(\mathbf{c}) \, d\mathbf{c} \tag{3.4}$$

Denote D_{B} as the random variable representing the degree of an AP $b \in \mathcal{B}$, where $D_{\mathrm{B}} \in \{0, 1, \ldots, N\}$. Probability that a given AP b_i has the given degree $D_{\mathrm{B}} = d$ can be written as (3.5), and its expected degree $\mu_{D_{\mathrm{B}}}(b_i)$ can be written as (3.6).

$$P(D_{\mathrm{B}} = d \mid b_i) = \binom{N}{d} P_0(b_i)^d (1 - P_0(b_i))^{N-d} \tag{3.5}$$

$$E\{D_{\mathrm{B}} \mid b_i\} \triangleq \mu_{D_{\mathrm{B}}}(b_i) = (N)P_0(b_i) \tag{3.6}$$

3.3.3.2 Joint Optimization

In this approach, locations of all APs are collectively considered at the same time as a single joint optimization problem. Let the volume $A_\cup$ be the union of the communication volumes of each AP, written as

$$A_\cup \triangleq \bigcup_{i=1}^{K} f_{\mathrm{vol}}(b_i). \tag{3.7}$$

The goal function, expected fraction of nodes that are connected to any AP, can then be written as

$$\mathrm{g_{joint}}(\mathcal{B}) = \iiint_{A_\cup} f_c(\mathbf{c}) \, \mathrm{d}\mathbf{c}. \tag{3.8}$$

Let $A_\cap$ be the union of the intersection of the communication volumes of every AP pair, denoting the volume where communication volumes overlap, defined as

$$A_\cap \triangleq \bigcup_{i=1}^{K} \bigcup_{\substack{j=1 \\ j \neq i}}^{K} \left(f_{\mathrm{vol}}(b_i) \cap f_{\mathrm{vol}}(b_j) \right). \tag{3.9}$$

It is possible to expand (3.8) by using (3.6) and (3.9) as

$$\mathrm{g_{joint}}(\mathcal{B}) = \left(\sum_{i=1}^{K} \mu_{D_{\mathrm{B}}}(b_i) \right) / N - \iiint_{A_\cap} f_c(\mathbf{c}) \, \mathrm{d}\mathbf{c}. \tag{3.10}$$

Let A_{lim} be the 3D volume that bounds the space under consideration, defined as the rectangular prism that is described by the ranges, $\{[x_{\min}, x_{\max}], [y_{\min}, y_{\max}], [z_{\min}, z_{\max}]\}$. Then, the optimization problem can be stated as

$$\underset{\mathcal{B}}{\text{maximize}} \quad \text{g}_{\text{joint}}(\mathcal{B})$$

$$\text{subject to} \quad |\mathcal{B}| = K, \tag{3.11}$$

$$\mathbf{c_i} \in A_{\text{lim}}, \quad \forall u_i \in \mathcal{B}.$$

(3.11) is hard to compute and thus proposed to be solved by an algorithmic optimization method. Existence of unpredictable perturbations such as buildings, and the dynamic nature of the users in the application environment makes calculation of a gradient unfeasible. Therefore, evolutionary optimization methods are employed for the solution of (3.11). For this purpose, we have utilized particle swarm optimization [103]. More details about computational optimization is provided in section 3.3.4. Please note that the optimization problem in (3.11) is a search in $3K$ dimensional space. Therefore, even for modest values of K, search space for joint optimization can be too large to reliably achieve global optimum with evolutionary methods, and there is no guarantee of convergence. This convergence issue is an important subject for future research and discussed in following sections.

3.3.3.3 Iterative Optimization

Joint optimization in the previous subsection is computationally challenging for even modestly large values of K. When considering network dynamics such as varying communication channels, dynamic user traffic patterns, or varying number of available APs, K, the desired optimization has to update the solution for each snapshot in time. Therefore, an iterative optimization algorithm that reduces the search space significantly to meet practical needs is developed to answer these challenges.

This approach assumes that the placement of each AP is an independent optimization problem, and therefore they can be sequentially placed, one by one. While this may appear to be a strong assumption, it is actually reasonable since all APs are connected to the same

network infrastructure through the 3D-RAN, and the only goal considered in the context of this paper is network connectivity.

Considering that the final goal is to achieve a rapidly deployed network with the highest fraction of connected users possible, the goal function for the placement of an individual, independent AP b_i can be written as the fraction of nodes that are connected to the network through b_i:

$$g_{\text{single}}(b_i) = |\mathcal{Q}_i^H|/N. \tag{3.12}$$

When the network is considered as single-hop, $H = 1$, (3.12) can also be expressed in terms of the expected degree of the basestation node b_i on the graph G, calculated as (3.6):

$$g_{\text{single}}(b_i) = \left(\mu_{D_{\text{B}}}(b_i)\right)/N = P_0(b_i), \tag{3.13}$$

which makes the optimization problem to be solved for each node of the network that represent the independent APs $u_i \in \mathbf{B}$, where $|\mathbf{B}| = K$ and A_{lim} is the boundaries in 3D space that was defined previously for Joint Optimization, as

$$
\begin{aligned}
\underset{u_i}{\text{maximize}} \quad & g_{\text{single}}(u_i) \\
\text{subject to} \quad & u_i \in \mathcal{B}, \\
& \mathbf{c_i} \in A_{\text{lim}}.
\end{aligned}
\tag{3.14}
$$

Optimization problem for placing an individual, independent AP is given in (3.14). However, most practical problems require multiple APs, and therefore a procedure for placing multiple APs is also necessary. We are proposing an iterative optimization procedure that can be used to place any number of APs with the goal of maximizing the fraction of connected users. This procedure can also respond to network dynamics by making corrections to

positions of APs individually and sequentially. It can also be used to determine the adequate number of APs, K, that achieves a desired fraction of connected users.

Let b_i and b_j be any two APs, $b_i, b_j \in \mathcal{B}$, that are placed individually according to the optimization problem (3.14). The sets of nodes that are connected to the network through b_i and b_j are calculated as $\mathcal{Q}_i^H$ and $\mathcal{Q}_j^H$ respectively. $\mathcal{Q}_i^H$ and $\mathcal{Q}_j^H$ are not necessarily disjoint, and in the case of a single unique solution for (3.14), they are equivalent. Since the final goal of the procedure is to maximize the fraction of connected user nodes, there is no advantage in connecting a single node to multiple APs, which aligns with the target application scenarios such as post-disaster relief networks. Therefore, after each placement of an AP, the nodes that are connected to the network through it can be safely removed from the network.

Let G^k be the RGG that represents the network when AP b_k, $\forall b_k \in \mathcal{B}$, is being placed, $G^k \triangleq \mathrm{RGG}(\mathcal{U}^k, f_{\mathrm{vol}}(u_i))$. Let $\mathcal{U}^k$ here be the set of all vertices excluding the ones that represent the user nodes that are connected to the network when AP b_k, $\forall b_k \in \mathcal{B}$, is being placed, defined as $\mathcal{U}^k \triangleq \mathcal{U} \setminus \left(\bigcup_{i=1}^{k} \mathcal{Q}_{i-1}^H \right)$, initialized with $\mathcal{Q}_0^H = \varnothing$.

Accordingly, optimization for K number of APs can be done by sequentially solving the individual AP placement problem (3.14) for each AP $b_k, \forall b_k \in \mathcal{B}$, on graph G^k. This guarantees that a single user node is not connected to multiple APs, meaning that the elements of the set $\{\mathcal{Q}_k^H : \forall k \implies k \in \{1, 2, \ldots, K\}\}$ are mutually disjoint. Practical concerns regarding the optimization from Joint Optimization stand here as well, and therefore evolutionary methods, particularly particle swarm optimization, are chosen again for optimization. The following algorithm details this Sequential Placement procedure.

Next, an iterative algorithm that can update AP locations individually is developed for responding to network dynamics. Since AP placements are assumed to be placed independently and individually in a sequence, Sequential Placement procedure can be modified to work iteratively, where each iteration starts from the previous iteration and recalculates the position of each AP sequentially.

```
Input: G, K
Output: B
 1: B ← ∅
 2: for all k ∈ {1, 2, . . . , K} do
 3:     b_k ← solve for (3.14) on G^k
 4:     B ← B ∪ {b_k}
 5: end for
 6: return  B
```

Procedure 1: The Sequential Placement procedure.

Let G_n be the graph that excludes user nodes connected to the network at iteration n, $G_n \triangleq RGG(\mathcal{M}_n \cup \mathcal{B}, f_{\text{vol}}(u))$, where $\mathcal{M}_n$ is the set of all user nodes excluding the user nodes that are connected to the network at iteration n, $\mathcal{M}_n \triangleq \mathcal{M} \setminus \bigcup_{i=1}^{K} \mathcal{Q}_i^H$. Let $b_k \in \mathcal{B}$ denote the AP to be relocated. Iterative Update Process is done by first removing b_k from the graph G_n. This means that all the user nodes that were connected to the network through b_k, $\mathcal{Q}_k^H$, are returned back to the graph G_n. Let G_n^k be the graph that excludes user nodes connected to the network at iteration n when relocating AP $b_k, \forall b_k \in \mathcal{B}$, defined as $G_n^k \triangleq \text{RGG}(\mathcal{M}_n^k \cup (\mathcal{B} \setminus \{b_k\}), f_{\text{vol}}(u))$. Let $\mathcal{M}_n^k$ here be the set of all user nodes excluding the user nodes that are connected to the network at iteration n when relocating AP $b_k, \forall b_k \in \mathcal{B}$, defined as $\mathcal{M}_n^k \triangleq \mathcal{M}_n \cup \mathcal{Q}_k^H$.

After the graph G_n^k is constructed, the single AP placement problem (3.14) for AP b_k is solved on this graph, and this is repeated for each AP sequentially. After a single pass over the set of all APs, $\mathcal{B}$, is done, a single iteration is completed. This process is then iterated until a termination condition is reached or next snapshot arrives. Termination condition can be chosen as when the solution stops improving through iterations, a given connectivity goal, or a combination of them. Following algorithm details this process.

It is also possible to determine the adequate number of APs, K, by starting the algorithm from a single AP, $K = 1$, and incrementing K one by one until the required network connectivity is reached.

```
Input: G
Output: B
 1: n ← 0
 2: while true do
 3:    n ← n + 1
 4:    for all k ∈ {1, 2, . . . , K} do
 5:       B ← B \ {b_k}
 6:       b_k ← solve for (3.14) on G_n^{'k}
 7:       B ← B ∪ {b_k}
 8:       Update connections on G for B
 9:       if termination condition is met then
10:          return  B
11:       end if
12:    end for
13: end while
```

Procedure 2: The iterative update process.

3.3.4 Simulations

In this section, we present the simulation results for the two proposed methods for genuine 3D AP placement explained in the previous section, as well as the Joint Optimization method with a 2D plane assumption where all APs are placed at the same altitude on a plane.

3.3.4.1 Setup

Simulation environment boundaries are limited to a unit cube. Within this unit cube, two kinds of perturbations are placed. First, ground surface is arbitrarily modified to represent possible geographical features that may effect user height. After that, rectangular prisms are placed arbitrarily to represent buildings and other obstacles that could limit LoS. It was assumed earlier that if there is LoS between two nodes within range, they can communicate with each other. If LoS is obstructed between two nodes, then they can not communicate, regardless of the distance. We post this constraint on purpose as the path exponents are high for many post-disaster scenarios. LoS between nodes is determined through ray casting in 3D. N number of users are distributed uniformly on the environment surface, while holding

out a fraction of users, n_b to place within buildings to achieve more realistic user densities around buildings. Figure 3.3 shows the simulation stage with APs deployed, for $N = 400$ and $n_b = 0.375$. Users that end up within a building are placed uniformly on its viable surfaces to avoid users encased within perturbations.

3.3.4.2 Experiments

Simulations are done for placement of $\mathcal{K} = \{4, 9, 16\}$ number of APs, with normalized communication ranges r_B, in the range $[0.075, 0.3]$ with the increments of 0.025. Therefore, a total of 30 simulations are done for each method. Local Best Particle Swarm Optimization implementation PySwarms [104] software module is used as the optimizer, with 1000 particles and 50 iterations for each optimization problem, with following hyperparameters: cognitive parameter $c1 = 0.5$, social parameter $c2 = 0.3$, inertia $w = 0.4$, neighbors $k_n = 10$, and $L2$ norm for determination of nearest neighbors. Figure 3.4 summarizes the results for these simulations. The iterative method is generally more successful, and the superior performance is especially notable for smaller communication ranges where utilization of the 3D placement becomes more crucial to be able to avoid obstacles. It should be stated that Joint Optimization in 3D was expected to perform at least as well, or better than the Iterative Optimization in 3D. The lacking performance can be explained by the very large size of the search space for joint optimization, which makes reaching the global optimum much harder compared to the much smaller individual optimization problems that are solved by the iterative method. Figure 3.3 shows the final placement on the simulation environment by the Iterative Optimization in 3D and Joint Optimization with 2D plane assumption methods, for $K = 16$ and $r_B = 0.125$. How genuine 3D placement takes advantage of the altitude dimension can be observed in this figure.

Selection of the adequate number of APs, K, is also simulated on the same simulation stage for a number of AP communication range values for single-hop networking, $H = 1$.

Figure 3.5 shows the results, where the adequate value of K for a given connected user fraction can be determined by stopping after the desired user connectivity is reached.

3.3.5 Conclusion

We have presented a formulation for the AP placement in genuine 3D problem based on the analysis of RGGs. An iterative algorithm is developed to alleviate the issue of rapidly increasing search space of AP locations as the number of APs increase. An evolutionary optimization method is utilized for the individual optimization problems as the unpredictable nature of the environment and network dynamics makes gradient based methods unfeasible. Proposed Iterative Optimization in 3D algorithm can find effective placements for APs in 3D space, respond to network dynamics with sequential small updates and determine the adequate number of APs for the desired coverage. Agent based simulation results, with consideration to uneven terrain and LoS blocking obstacles, have shown that the deployment locations produced by the proposed Iterative Optimization method in 3D are effective, especially for larger numbers of available APs with short range, where avoiding obstacles presented by the environment and conforming to the ground surface height changes are especially crucial.

(a) *2D-Joint*
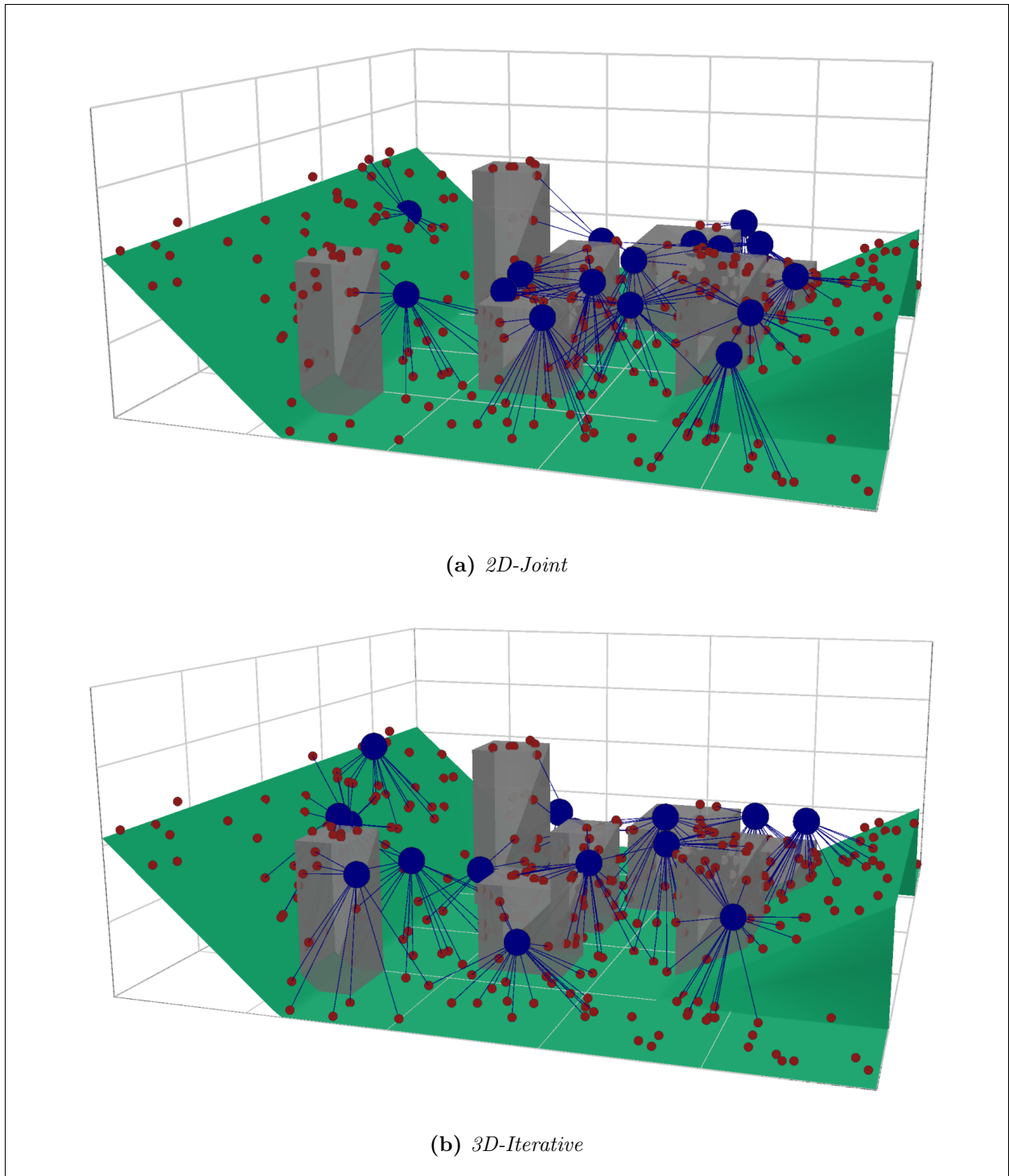
(b) *3D-Iterative*

Figure 3.3: Final placement for (a) Joint Optimization with 2D plane assumption and (b) Iterative Optimization in 3D methods, $K = 16$ and $r_{\mathrm{B}} = 0.125$. Blue spheres are APs and connections are denoted by blue lines. The difference for genuine 3D placement can be clearly observed.
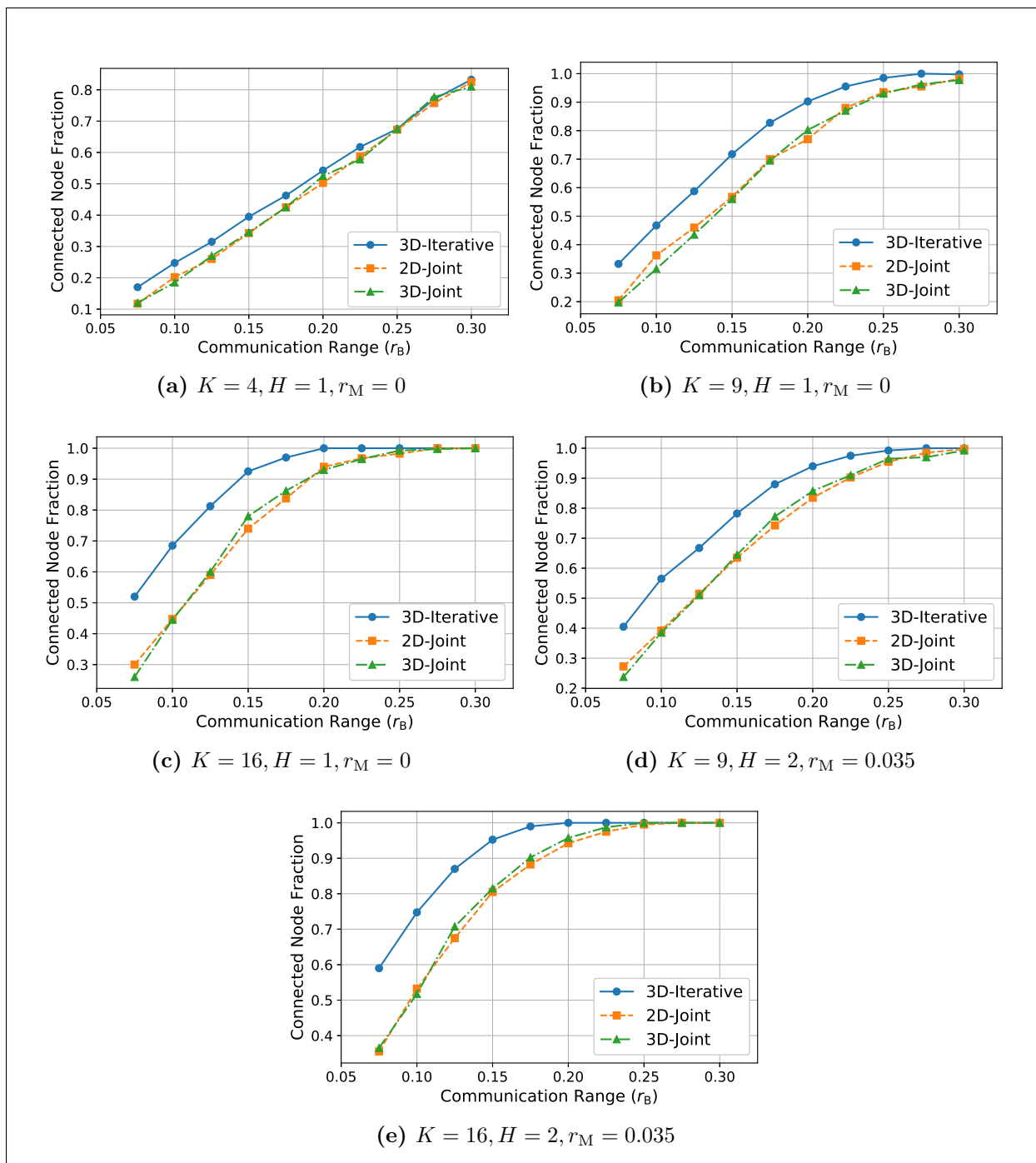
Figure 3.4: Simulation results comparing Joint Optimization with 2D plane assumption (2D-Joint), Joint Optimization in 3D (3D-Joint) and Iterative Optimization in 3D (3D-Iterative) methods. Showing connected fraction of users versus AP communication range where H is the number of hops permitted, r_M is the range of the short-range user-to-user link, K is the number of available APs, and r_B is AP communication range.
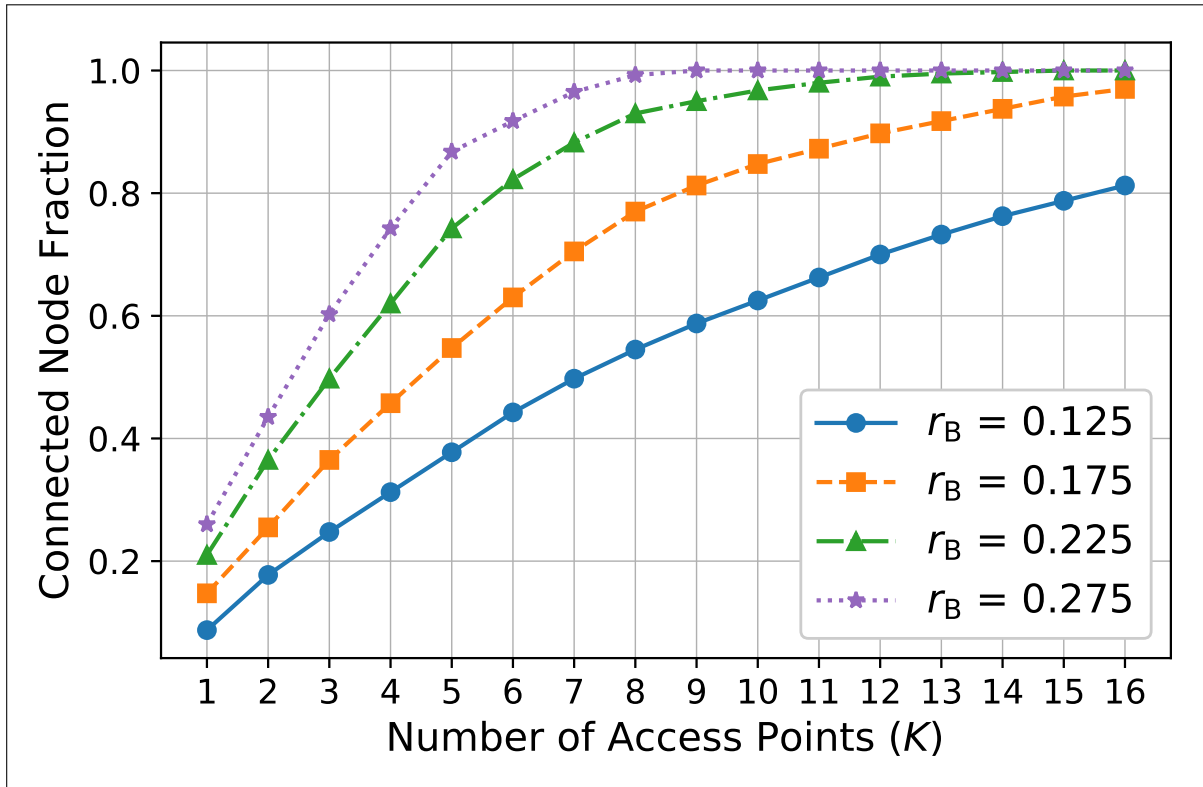
Figure 3.5: Number of APs, K, versus connected node fraction. Different traces are provided for varying AP communication ranges, r_B. The adequate value of K can be determined by looking up when the trace crosses the desired level of node connectivity.

Chapter 4: Aerial Mobile Radio Access Network Design

4.1   Introduction

Advances in unmanned aerial vehicles (UAVs) suggest new application scenarios for mobile communications and possibilities to establish flexible radio access networks (RAN) and thus wireless infrastructure [28, 105]. Flexible and rapid deployment of an aerial RAN (aRAN) is critical to post-disaster communications and, in the near future, highly dynamic usage patterns of humans and machines. Alphabet's Project Loon and AT&T's flying cell-on-wheels (COW) are some experimental attempts to bring connectivity to areas without the traditional infrastructure. The latter example has been deployed in Puerto Rico in the aftermath of Hurricane Maria.

Most common applications of UAV based wireless networks consider UAVs as aerial basestations for augmenting existing cellular infrastructure. Research have been done on both a single UAV basestation [20, 21], and multiple UAVs [95, 87]. However, these works deal with static deployments and follow traditional optimization approaches with impractical priors such as known user locations.

An aRAN consisting of UAV aBSs or aerial access points (aAPs) must deal with a highly dynamic operating environment with limited knowledge of network and traffic dynamics, where requfirements of tight closed loop control can be considered a luxury. Consequently, the trajectory design of aAPs in the aRAN to effectively serve user traffic faces a critical challenge that is computationally unfeasible [27, 28]. A disruptive approach is to treat

---

[2]Material from this chapter has been adapted to a manuscript and submitted to Wireless Communication Letters for potential publication.

maneuvering of an aAP as an autonomous mobile robot, which suggests employing Reinforcement learning (RL) for the trajectory design problem of a mobile UAV-based aRAN. RL in particular is suitable for controlling agents in scenarios where prior information is limited and the agent has to explore to gain further information and exploit the current information it has. There are works in the literature that tackle the trajectory design problem of mobile UAV based networks for augmenting existing cellular networks through aerial base stations; [29] provides a Q-Learning based approach for a static scenario where the best-case agent presented takes 27000 episodes to learn on a 2 user case, [30] presents a Q-Learning based approach that attempts to compensate for simulated user movement, and [31] presents another Q-Learning approach that takes advantage of Twitter data to predict user movements whose initial locations are assumed to be known a priori. [32] presents a case for a single-cell UAV network, however they assume all user locations are static and known a priori. There is also a particular limitation all of these approaches share where they aim to optimize the sum of metrics over all users, which will likely result in ignoring the fairness of service among users and neglecting remotely located users that an aRAN must cover.

In this work we present a solution for quickly building an aerial communication infrastructure for a dynamic network of mobile users where aAPs are the only method of radio access. Highly dynamic nature of the network and the lack of user traffic and mobility models make the control of the trajectories for aAPs unfeasible following traditional trajectory optimization approaches, and we therefore approach this challenge by modeling it as a RL problem. Building the entire infrastructure as opposed to simply augmenting existing cellular infrastructure with aerial basestations is challenging because any users that are not reached by the aerial APs will receive no service at all, since there is no alternate infrastructure to fall back on. We take fairness of service among users into account through careful design of state and reward. Another important challenge is achieving a rapid deployment and quickly bringing service to the area which is represented by the model's convergence speed. We

take advantage of the symmetries in the state-action space to its size 5-fold, significantly increasing the sample efficiency utilizing after-states. We also uniquely develop a scheme to incorporate wireless communication among collaborative aAPs, which is crucial in effectively coordinating multiple aAPs towards a single shared goal. We require only the total number of users in the area as prior information, as opposed to the impractical requirements of exact locations or distributions of all users as in prior art. Finally, we present simulation results for a range of scenarios.

4.2   Network System Architecture
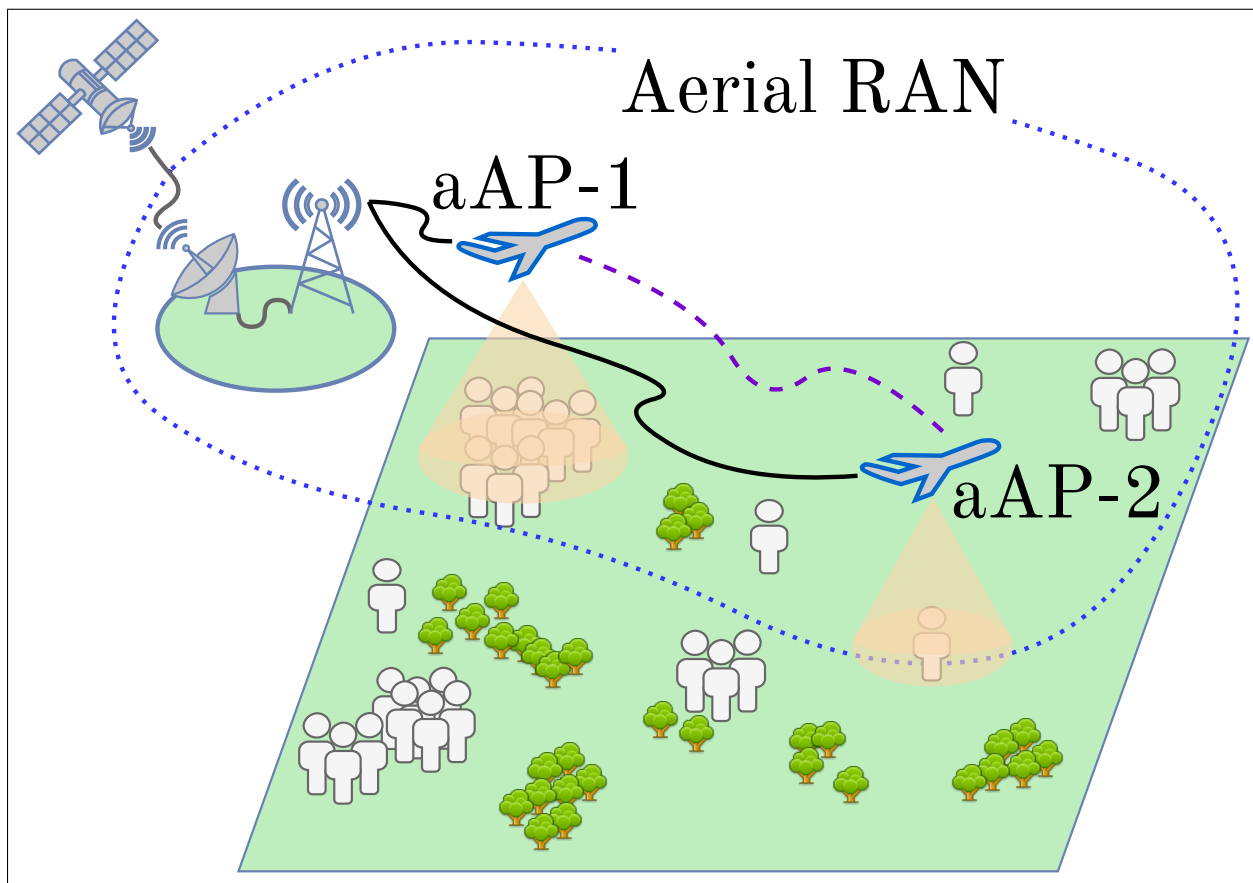
Figure 4.1: A diagram of the system model with two aAPs in operation, forming the aRAN.
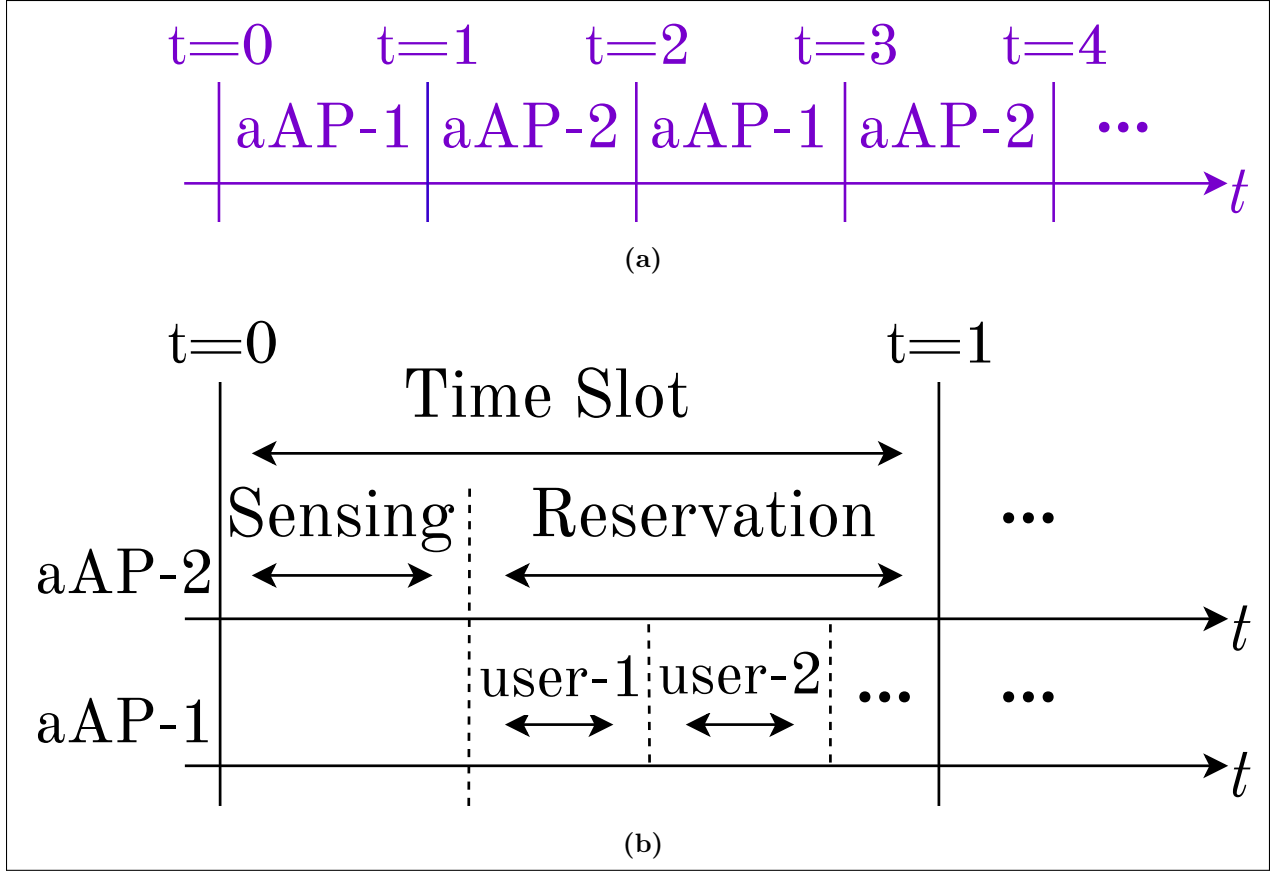
Figure 4.2: Timing diagrams of the communication links. (a) Timing diagram of the round-robin communication between aAPs. (b) Timing diagram of the communication between aAPs and users, where sensing represents general information collection about user traffic and polling represents a reservation-based multiple access mechanism.

Among diverse application scenarios, an aRAN generally serves an area without an adequate communication infrastructure. Assuming all aAPs are connected to backhaul through a reservation-based wireless access protocol, a flight control mechanism attached to the backhaul network monitors the battery lifetime and can call back an aAP before its battery dies out and send out a substitute aAP following a UAV swapping scheme [106]. This call back is realized by sending a control signal at the end of an episode. After a possible pilot tour to estimate the total number of users, the service area of the aRAN is discretized into $N \times N$ square grids. The aRAN is assumed synchronized in time-slots.

To focus on the trajectory design for an aRAN of multiple aAPs, for the purpose of feasibility to a broad range of wireless communication systems, we adopt the communication system model illustrated by Figure 4.1 that involves three kinds of multiple-access communications. (i) Air-interface between an aAP and users in a grid: Given a reservation-based multiple access, one unit of service is abstractly defined as the service that one user receives from an aAP in one time-slot. As illustrated in Figure 4.2b, at the beginning of each time-slot aAPs sense the traffic demand in their grid, and then poll the users within the grid. Only one aAP operates in each grid at one time and the interference from neighboring grids is assumed to be negligible, while polling here represents a general reservation-based multiple access such as the IEEE 802.11 mechanism [107]. (ii) Backhaul network and aAPs of aRAN: We assume that each aAP is connected to a backhaul network infrastructure (or core network), which may be a direct satellite connection or a ground BS connected to a backbone network. The multiple access between aAPs and backhaul network is assumed to be a reservation-based mechanism with reliable wireless communication between each aAP and the backhaul. (iii) Inter-Communication between two aAPs: Leveraging the communication between backhaul and any aAP, two-hop communication ensures that inter-communication between two aAPs is equivalent to a "dedicated" (i.e. reserved) channel in a round-robin manner such that at each time-slot one aAP multicasts its control message to all other aAPs [108] as illustrated in Figure 4.2a, where the control message may contain the most recent state, action, and the resulting state and reward under a latency constraint.

A user is flagged as served when they receive one unit of service, and the flags are reset when all the users are flagged as served. Without loss of generality, the service requirements (such as QoS) of all users are assumed equivalent.

4.3   Reinforcement Learning

In this section, we will provide a brief summary of reinforcement learning, and particularly the Q-Learning algorithm that will be employed in the proposed solution.

As opposed to supervised and unsupervised learning paradigms where data is provided to the model for training, reinforcement learning attempts to solve problems where there is no previous information available and an agent must learn by interacting with an environment through actions and observing the results of these actions, which can be more formally defined as the optimal control of an incompletely known Markov decision processes [109].

To have a well defined reinforcement learning problem; agent, environment, interaction between them, and a goal for the agent must be defined. Agent acts on the environment by choosing an action a from a space of all actions A, considering their state $\mathbf{s}$ following a policy $\pi(\mathbf{s})$, and receives information from the environment in the form of an updated state variable $\mathbf{s}^{+1}$ that represents the new state of the agent after the action a, a reward signal $R(\mathbf{s}, a)$ that represents the contribution of the chosen action towards the goal, and a value function that gives the expected total reward from the new state $\mathbf{s}^{+1}$, representing the long term rewards that can be achieved starting from that state [109].

Learning to act towards a specific goal is often facilitated by estimating a state-action value function $Q(\mathbf{s}, a)$. Value here represents the expected total reward by taking action a from state $\mathbf{s}$. This estimated state-action value function can be realized by tabulating all the state-action pairs and their estimated values, or by function approximators such as neural networks.

Next, we need to determine which reinforcement learning method is suitable. As we have mentioned earlier, main challenges of this problem is the lack of an environment model and the dynamic nature of the network. Lack of a model can be addressed by using a model-free method, and a dynamic environment will require a method that works online, in small

incremental steps. Time Difference (TD) learning is a family of model-free reinforcement learning methods that are naturally implemented as online algorithms, meaning that they update their estimations after every step and learn in an incremental fashion. This makes it possible to follow and adapt to the changes in the environment as they are happening with small updates to the estimated models.

Time Difference learning methods can be divided into two categories as on-policy and off-policy methods. On-policy methods estimate the value function $Q_\pi(\mathbf{s}, a)$ for the policy π that is being followed. Off-policy methods, on the other hand can estimate the state-action value function for an arbitrary policy, unrelated to the policy π agent is following. Off-policy methods are particularly desirable for non-stationary scenarios as making sub-optimal decisions are necessary for the constant exploration to track the changes in the environment and adapt to them. Following an on-policy method, these sub-optimal decisions will result in estimation of a sub-optimal state-value function. Q-Learning is a popular algorithm that is widely studied and used for implementing off-policy TD learning. It estimates the optimal state-action value function directly regardless of the policy being followed [109]. This makes it possible to explore as much as necessary without degradation of the estimated state-value function.

4.3.1   Q-Learning

Q-Learning is an algorithm that is used to implement off-policy time-difference learning [110]. Off-policy here means that the learned action-value function $Q(\mathbf{s}, a)$ estimates the optimum action-value function, regardless of the particular policy being followed. If the state at time t is $\mathbf{s}_t$ and the action a_t is taken from this state, (4.1) gives the update formula used to estimate the new value for the state-action pair at time t, $Q^{+1}(\mathbf{s}_t, a_t)$.

$$Q^{+1}(\mathbf{s}_t, a_t) = Q(\mathbf{s}_t, a_t) + \alpha[R(\mathbf{s}_t, a_t) + \gamma \max_a Q(\mathbf{s}_{t+1}, a) - Q(\mathbf{s}_t, a_t)] \qquad (4.1)$$

Here $0 < \alpha < 1$ is the step-size parameter that determines the speed of learning, and $0 < \gamma < 1$ is the discount factor that determines how much weight is given to future rewards against immediate rewards. These two parameters should be tuned carefully to ensure a timely convergence to a good solution. It has been proven that the estimated state-action value function $Q(\mathbf{s}, a)$ will converge to the optimal state-action value function with probability 1 given that all state-action pairs are visited infinitely, and the step-size parameter is reduced over following stochastic approximation conditions [109]. However, this convergence condition is not practical for a non-stationary environment, since we would like to have the estimated Q function track changes in the environment.

4.4   Collaborative Multi-Trajectory by RL

The main objective is to formulate the trajectory control of multiple aAPs that fairly prioritizes users based on limited *a priori* information about the total number of users in the area, which differs from existing literature requiring exact locations of users and their mobility models [95, 87]. The state-action value function $Q(\mathbf{s}, a)$ is implemented via tabulation. We first adopt a basic Q-Learning approach for a single-agent scenario, and then present a modified Q-learning solution suitable for a dynamic wireless network with mobile users through state design and reward shaping, and the use of after-states. It is then further modified for multi-agent scenarios with an arbitrary number of aAPs through decentralized inter-agent collaboration where each agent executes its policy independently, and effective collaboration relies on inter-AP communication.

The trajectory of a single aAP can be controlled by basic Q-learning as follows.

4.4.1 State

State variable $\mathbf{s}$ for aAP is defined as the grid coordinates of the aAP, $\mathbf{s} = \begin{bmatrix} \mathbf{c} \end{bmatrix}$, where $\mathbf{c} = \begin{bmatrix} x & y & z \end{bmatrix}$.

4.4.2 Action Set

Without loss of generality, action set $\mathcal{A}$ of the aAP is limited to 5 actions, moving forward, backward, left, right, and flying in place, by fixing the altitude z.

4.4.3 Policy

To balance exploration of new information with exploitation of current information, an ϵ-greedy approach is used, which uniformly samples an action from $\mathcal{A}$ with the probability ϵ, and takes a greedy action otherwise, as (4.2).

$$\pi(\mathbf{s}) = \begin{cases} \mathrm{argmax}_a\, Q(\mathbf{s}, a), & \text{with probability } 1 - \epsilon \\ \text{random action } a, & \text{with probability } \epsilon \end{cases} \tag{4.2}$$

4.4.4 Reward

We can capture the goal of the system in a straightforward manner by providing a positive reward for finding users, and a negative reward for any other action. As Q-Learning adapts a policy to maximize the total expected reward, aAPs will learn the shortest path to users. Exact values of positive and negative reward are relevant relative to each other, as actions are chosen through comparison of state-action value estimates $Q(\mathbf{s}, a)$, which give the estimated total reward for each action a from state $\mathbf{s}$. The fraction of positive to negative reward can effect the convergence speed, and we chose the values as $+1$ and -1 following empirical tests.

In each time-slot, the environment provides a reward of $+1$ if aAP is in the same grid with users, and -1 otherwise. This leads agents to follow the shortest known path to users in order to maximize the reward. Let $\|\cdot\|_{\mathbf{p}}$ be the p-th norm. Let $\mathcal{U}$ be the set of users in the same grid as the aAP. Let $\boldsymbol{\omega} = \left[\omega_i\right]$ where $\omega_i = \frac{1}{|U|}$ if $|U| \neq 0$ and $\omega_i = 0$ if $|U| = 0$.

$$R(\mathbf{s}, a) = 2 \|\boldsymbol{\omega}\|_1 - 1 \tag{4.3}$$

4.4.5 Learning Parameters

The agent presented in this sections is based on formulation of the trajectory control problem into a standard Q-Learning problem, and as such, it has the same convergence characteristics as Q-Learning detailed in the previous section; for a stationary environment, the state-action value function $Q(\mathbf{s}, a)$ of the agent will converge to the optimal action-value function with probability 1 given that all state-action value pairs are visited infinite times, and the step-size parameter is reduced appropriately. However, the scenario we are interested in is non-stationary, and therefore we would like to give a higher weight to more recent observations to be able to better track the changes in the environment. A common and practical way of achieving this is setting the step-size parameter α as a constant, which results in the decaying of older observation weights exponentially [109].

Discount factor γ determines how much weight is placed in future rewards, as opposed to immediate rewards. When discount factor is set $\gamma = 0$, only immediate rewards are taken into account, and potential for future rewards is disregarded. Suitable values for γ depend on the scenario, and must be selected carefully according to the application at hand.

Selection of values for hyperparameters is still an open problem, and they are tuned empirically in practice. We have tuned our parameters following an empirical grid-search over these variables repeated for $K = 1, 2, 3, 4$ number of agents. Grid-search is conducted by evaluating performance of all parameter 3-pairs for increments of 0.1 for α and γ, and
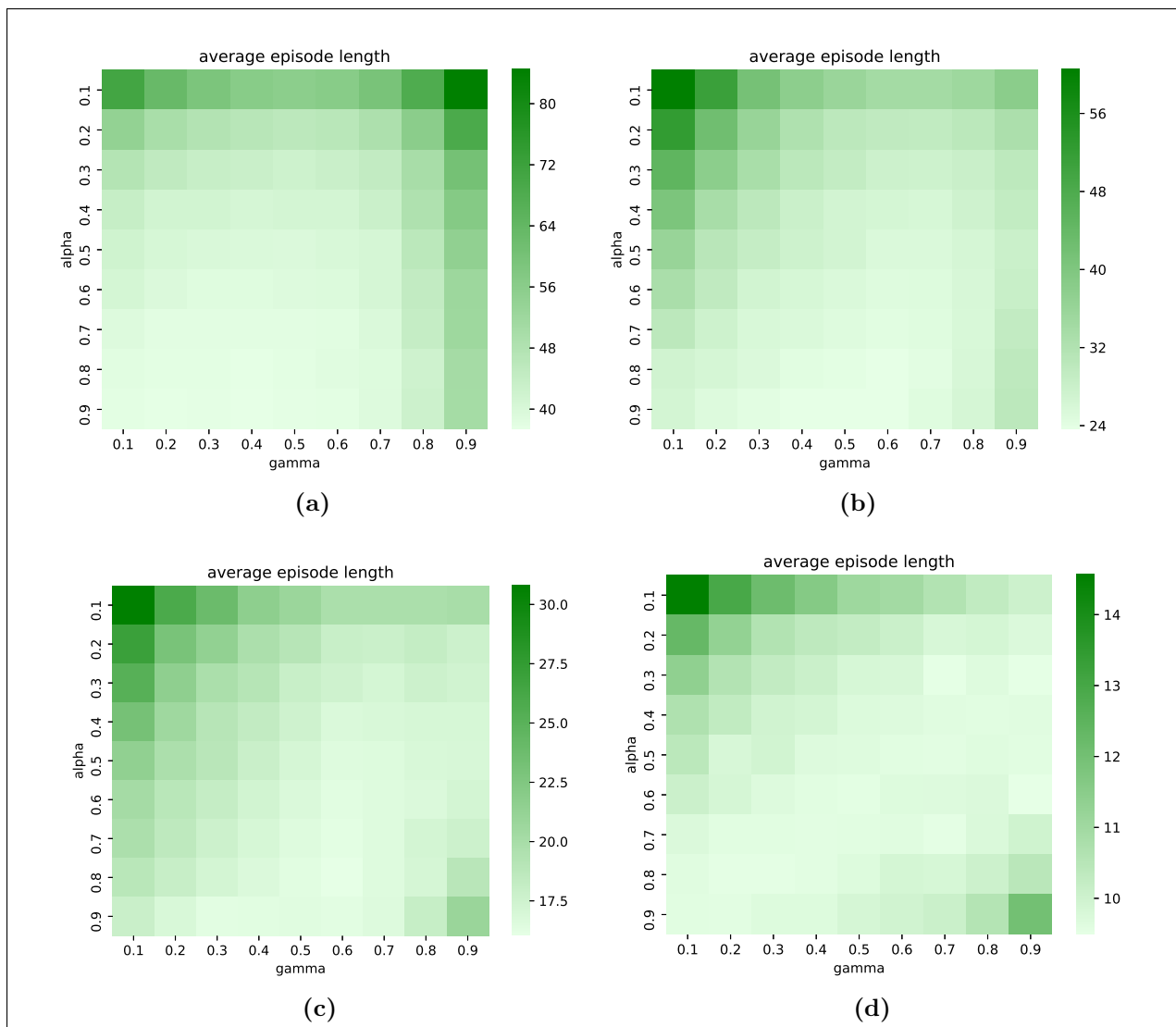
Figure 4.3: Average episode lengths (smaller is better) of a grid-search over parameters α and γ for fixed $\epsilon = 0.15$ and following number of agents; (a) 1, (b) 2, (c) 3, (d) 4.

an increment of 0.05 for ϵ. Figure 4.3 illustrates one example step of this search for 0.1 increments of α and γ, and $\epsilon = 0.15$. Based on this selection process we chose a single parameter set of $\alpha = 0.7$, $\gamma = 0.6$, and $\epsilon = 0.15$ because they performed relatively well for all 4 test cases. We should also state that we are intentionally tuning parameters coarsely on a randomized environment. The scenario we are interested in assumes that environment

conditions are unknown beforehand, and therefore it is not reasonable to fine-tune parameters to a specific environment.

4.4.6 Proposed Agent

The basic Q-Learning agent is insufficient to handle the non-stationary reward structure arising from the dynamic environment of mobile users and decentralized aAPs, and enhancement is required to fit a dynamic aerial wireless network scenario. Furthermore, the basic Q-learning approach tends to overlook minor traffic and isolated users, creating a concern of fairness in service. We therefore amend the state and modify the reward to handle the non-stationary rewards and introduce a prioritization scheme that can serve all users fairly through system design, and incorporate after-states to take advantage of the symmetries in the state-action space and reduce its size in 5-fold, significantly improving the sample efficiency.

4.4.6.1 State Variable Design

A vector of binary variables that hold the status information for each user, $\mathbf{g}$, designating which users have been satisfactorily served so far within an episode is added to the state variable $\mathbf{s}$. The user status vector is defined as $\mathbf{g} = \begin{bmatrix} g_k \end{bmatrix}, k \in [1, K]$ where K is the total number of users and the user status flag g_k is defined by (4.4).

$$g_k = \begin{cases} 0, & \text{if user } k \text{ received one unit of service in episode} \\ 1, & \text{otherwise.} \end{cases} \tag{4.4}$$

This vector serves as a memory of users that have been served so far within the episode, which is also incorporated to the reward to encourage seeking unserved users. The amended state variable then becomes $\mathbf{s} = \begin{bmatrix} \mathbf{c} & \mathbf{g} \end{bmatrix}$.

4.4.6.2   Reconstruction of Reward

We would like to encourage fairness of service among users by rewarding agents for finding unserved users within that episode. This is achieved by modifying the reward so that once an agent receives a reward for a user k setting its status flag $g_k = 0$ as served, there will be no further positive reward for serving that user until all other users are served and episode is concluded. With this modifications, aAPs will be required to find all users and serve them regularly to maximize total expected reward, and each user will receive equal amount of service within that period. This aspect of reward is achieved through the incorporation of user status vector $\mathbf{g}$ defined as (4.4), which serves as memory of users that have been served so far.

Let $|\cdot|$ be the set cardinality. Let $\mathcal{U}'$ be the set of unserved users in the same grid as aAP. Let $\boldsymbol{\omega}' = \begin{bmatrix} \omega'_i \end{bmatrix}$ where $\omega'_i = \frac{1}{|U'|}$ if $|U'| \neq 0$ and $\omega'_i = 0$ if $|U'| = 0$. Modifying (4.3), reconstructed reward is given by (4.5).

$$R(\mathbf{s}, a) = 2\boldsymbol{\omega}'^T \mathbf{g} - 1 \qquad (4.5)$$

The memory provided by $\mathbf{g}$ is incorporated to the reward through the vector multiplication $\boldsymbol{\omega}'^T \mathbf{g}$. If there is at least one unserved user in the same grid as the aAP, the vector multiplication will be $\boldsymbol{\omega}'^T \mathbf{g} = 1$, and if there are no unserved users $\boldsymbol{\omega}'^T \mathbf{g} = 0$. The reward (4.5) leads to a reward of $+1$ if there are any unserved users in the grid and -1 if there are no unserved users in the grid. This encourages aAPs to seek out unserved users to maximize reward.

Directly capturing the goal in a simple reward instead of using proxies or complex reward functions is desirable to avoid reward-hacking or wireheading problems [109, 111], where agents find an undesirable or unexpected way to receive rewards resulting in unintended behavior that does not contribute toward the goal. Our reward simply provides $+1$ for

finding unserved users, and -1 otherwise. This means that the only way to continuously receive positive reward is to find unserved users.

4.4.6.3 After-states

There are 5 separate actions in the action-set of the aAP, which means there are 5 state-action pairs that can move an aAP to the same internal grid coordinate $\mathbf{c}$. Since the state only includes the current coordinate, it is possible to take advantage of this symmetry and replace the state-action pair $(\mathbf{s}, a)$ with the after-state $\mathbf{s}_a$, where $\mathbf{s}_a = \begin{bmatrix} \mathbf{c}_a & \mathbf{g} \end{bmatrix}$. Here $\mathbf{c}_a$ indicates the resulting coordinate after taking the action a and $\mathbf{g}$ is the user status vector.

4.4.7 Multi-Agent Scenario

With the terminology we are adopting, multiple agents working towards a common goal can be achieved via cooperation with centralized execution or collaboration with decentralized execution. Centralized execution implies there is a central controller that has access to all the information of all agents, or in other words the global reference, and only the central controller can execute the learned policy. In a pure cooperative system, agents can't operate when their communication link with the central controller is down. This requires extremely reliable and ultra-low latency communication between the central controller and all the agents just for the control messages to the agents rather than serving actual data traffic. This is a strong requirement that is not very practical for mobile aerial wireless networks in general due to reliability constraints. Since aRAN aims at forming a flexible, dynamic, and scalable wireless infrastructure that can be quickly deployed in an unknown environment, we therefore consider collaborative aAPs with decentralized policy execution. In this case, there is no central controller but instead each agent keeps their own private references. Each agent also learns their own policy and executes this policy on their own, meaning that each aAP follows the algorithm described in subsection 4.4.6 individually.

We further explore the effects of the level of communication between aAPs on collaboration by implementing three possible inter-agent communication schemes. Private references of each agent is likely to lack information compared to the global reference, and this difference causes a loss of performance for collaboration compared to cooperation. However, decentralized agents can attempt to bridge this information gap by exchanging information with each other through inter-agent communication whenever available. The level of inter-agent communication is a deciding factor in how effective this attempt will be, and that is the reason we are implementing three schemes with different levels of inter-agent communication to explore this factor. The messages agents share with each other include their current state, action, reward received from the environment, and the post-action state. This is the entire observation of an agent, and it can be used to learn from other agents' experience and update the local policy accordingly through 4.1. This enables agents to incorporate information on parts of the state-action space they have never explored into their policies and therefore their decisions.

Let $\mathcal{V} = \{v_i\}, |\mathcal{V}| = M$, be the set of M aAPs where v_i represents aAP i. State variable of v_i is represented by $\mathbf{s}_i = \begin{bmatrix} \mathbf{c}_i & \mathbf{g}_i \end{bmatrix}$ where $\mathbf{c}_i$ is the coordinates and $\mathbf{g}_i$ is the user status vector belonging to v_i. The state-action value function is represented by $Q_i(\mathbf{s}_a)$. The private reference of v_i consists of $Q_i(\mathbf{s}_a)$ and $\mathbf{g}_i$. Each aAP $v_i \in \mathcal{V}$ operates decentralized, following the single-agent algorithm in Section III-B and executing the policy learned through $Q_i(\mathbf{s}_a)$ individually.

4.4.7.1 Multiple-Access Communication

This case represents the situation when there is a practical communication channel between aAPs. Messages between aAPs are communicated multi-cast through a dedicated round-robin connection explained previously where messages arrive with a time delay, as demonstrated in Figure 4.2a. The practical implication of this is that these messages arrive
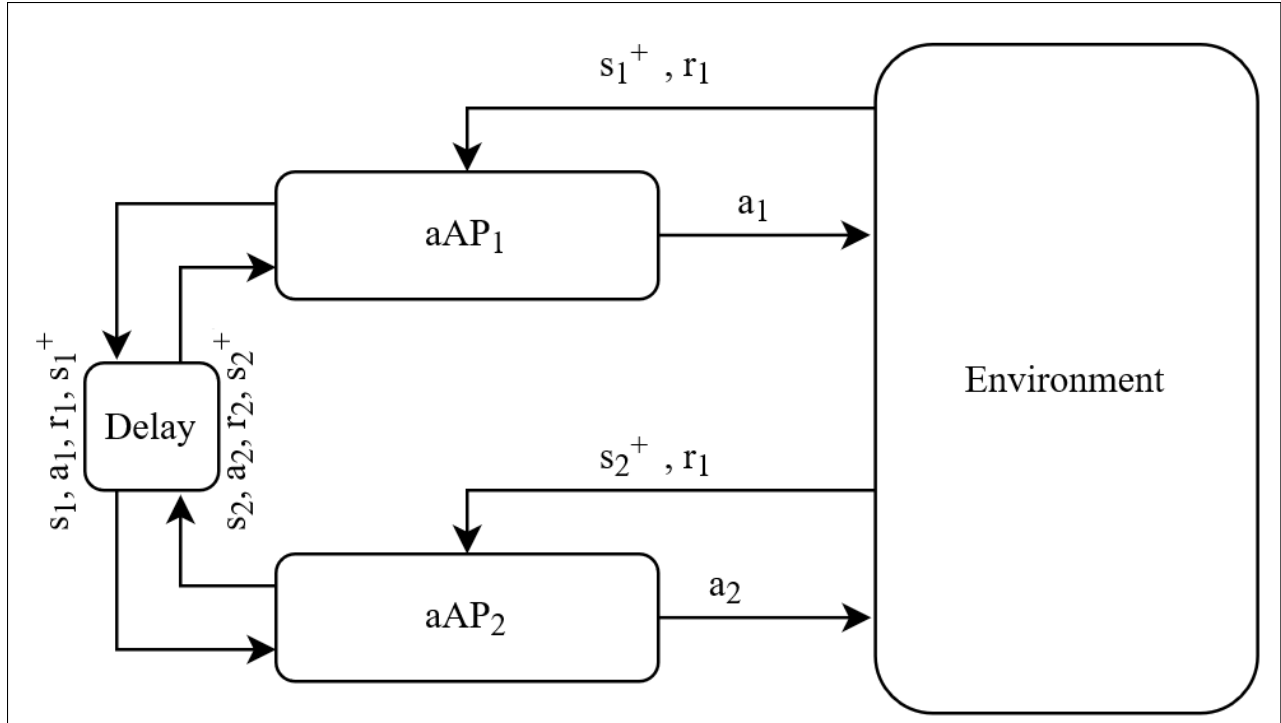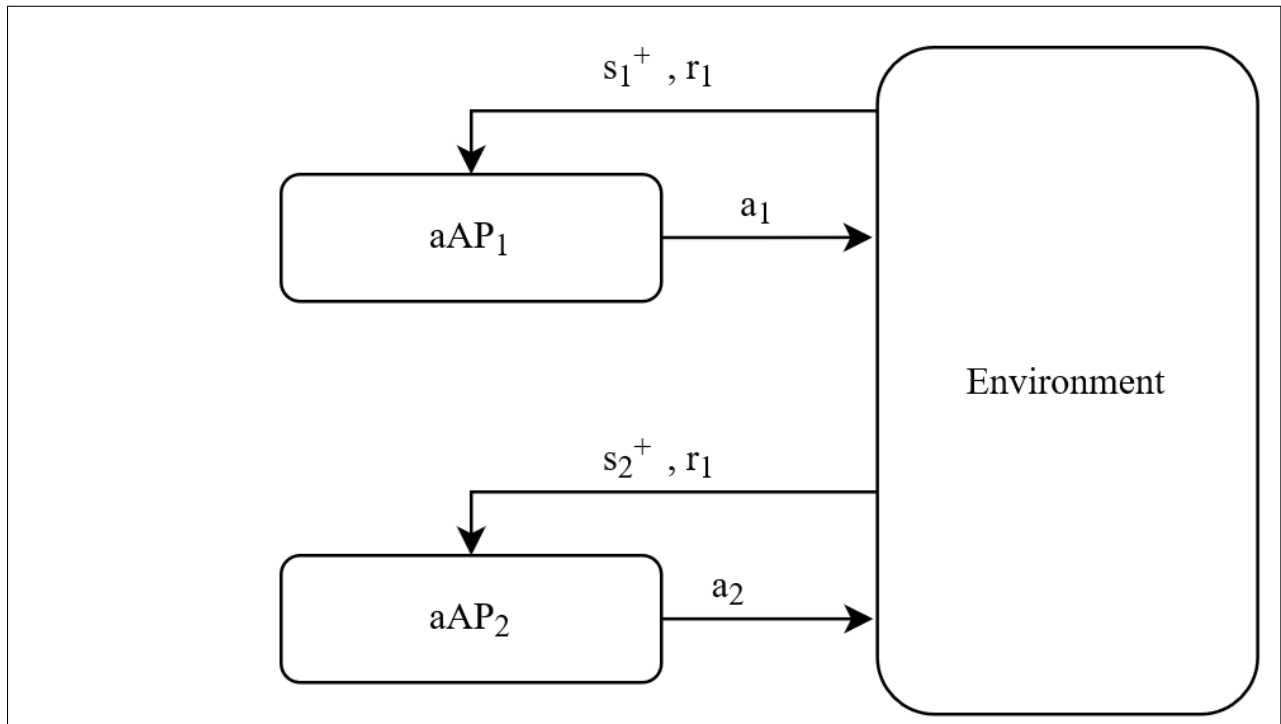
Figure 4.4: System diagram for multi-agent collaboration under the Multiple-Access Communication scheme for two aAPs. For aAP_i, $\mathbf{s}_i$ is the current state, a_i is the action, r_i is the reward received for action a_i, and $\mathbf{s}_i^+$ is the new state resulting from action a_i. Each aAP follow the algorithm described in subsection 4.4.6.

delayed. This means while aAPs have access to the state, action, and reward information of other aAPs, the information they receive is already stale. This prevents aAPs from operating on the global reference and each aAP has to operate on their own private references, however, in this case aAPs receive delayed information they can use to improve their private references. Each aAP has their private references, $Q_i(\mathbf{s}_a)$ and $\mathbf{g}_i$, as well as the delayed observations from other aAPs under this scheme. Relying on the limited latency of inter-aAP communication within the operation time scale of the system and expecting the inaccuracies caused by the latency to be small, we update the private references of each aAP v_i by treating received messages as nearly real-time. This is achieved by updating user status vector $\mathbf{g}_i$ and updating $Q_i(\mathbf{s}_a)$ following (4.1) with the received observation. This consists of (i) merging the received status vector $\mathbf{g}_i$ with all the received status vectors from other aAPs $\mathbf{g}_j, j \neq i$,

where $\mathbf{g}_i$ and $\mathbf{g}_j$ are merged by taking their Hadamard (elementwise) product $\mathbf{g}_i \odot \mathbf{g}_j$ and (ii) backing up the individual state-action value functions $Q_i(\mathbf{s}_a)$ with state, action, and reward information received from other aAPs.

4.4.7.2 Naive Estimation

Figure 4.5: System diagram for multi-agent collaboration under the Naive Estimation scheme for two aAPs (no inter-agent communication). The only connection between aAPs is through the environment. For aAP$_i$, $\mathbf{s}_i$ is the current state, a_i is the action, r_i is the reward received for action a_i, and $\mathbf{s}_i^+$ is the new state resulting from action a_i. Each aAP follow the algorithm described in subsection 4.4.6.

This represents the case with no inter-agent communication. However, even with no explicit message passing between aAPs, their actions cause changes in the environment. aAPs can collaborate by observing the results of each others actions through rewards received from the environment, which is realized as an estimation of expected value within the state-action value function $Q(\mathbf{s}_a)$ through Q-Learning. These observations act as an implicit channel of

communication between aAPs, which can lead to effective partitioning of a shared goal as aAPs will estimate low value through their state-action value function $Q_i(\mathbf{s}_a)$ in areas where other agents are active due to low rewards. Naive estimation is named after aAPs trusting their observations as if they represent the global reference with no way to validate through inter-agent message passing. In this case, each aAP operates on their own private references as they have no way of directly sharing their references with each other, meaning that each aAP operates on a separate state-action value function $Q_i(\mathbf{s}_a)$ and user status vector $\mathbf{g}_i$ both of which may have different and out-of-date values. These inaccurate values may cause aAPs to take suboptimal actions with regards the global reference, potentially resulting in multiple agents acting towards the same user, reducing effectiveness of collaboration. This scheme also reduces the learning speed as aAPs miss the opportunity to learn from the observations of other aAPs. Figure 4.5 illustrates this scheme. It can be seen that the only connection between agents is indirectly through the environment.

4.4.7.3   Ideal Communication

Having an ideal (real-time and error-free) channel between aAPs represents the best-case inter-aAP communication, where all aAPs have access to exactly the same information at all times. Since all aAPs have access to each others state, action, and reward information real-time, they are able to operate on a global reference. Practically, this means they share a single state-action value function $Q(\mathbf{s}_a)$ and user status vector $\mathbf{g}$ and the only difference between states $\mathbf{s}_a$ is the individual locations of aAPs, as seen in (4.6). This means all agents have access to most up-to-date information and won't take suboptimal actions with regards
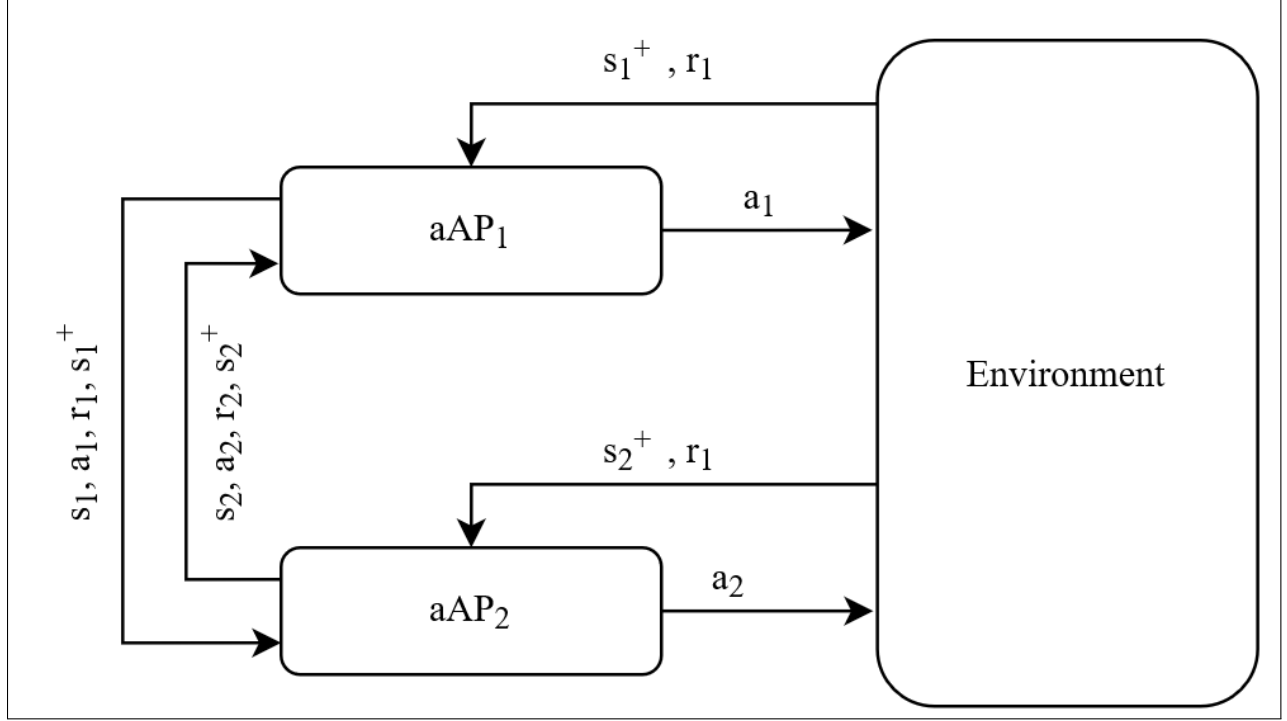
Figure 4.6: System diagram for multi-agent collaboration under the Ideal Communication scheme for two aAPs. For aAP_i, $\mathbf{s}_i$ is the current state, a_i is the action, r_i is the reward received for action a_i, and $\mathbf{s}_i^+$ is the new state resulting from action a_i. Each aAP follow the algorithm described in subsection 4.4.6.

to the global reference due to stale information. Figure 4.6 illustrates this scheme.

$$Q_i(\mathbf{s}_a) = Q(\mathbf{s}_a), \quad \forall i \text{ where } v_i \in \mathcal{V} \tag{4.6a}$$

$$\mathbf{g}_i = \mathbf{g}, \quad \forall i \text{ where } v_i \in \mathcal{V} \tag{4.6b}$$

$$\mathbf{s}_i = \begin{bmatrix} \mathbf{c}_i & \mathbf{g} \end{bmatrix}, \forall i \text{ where } v_i \in \mathcal{V} \tag{4.6c}$$

It is important to note that while all aAPs operate on a shared global reference, where each observation is used to update the shared state-action value function $Q(\mathbf{s}_a)$, the learned policy is executed in a decentralized manner by each individual aAP.

69

4.5 Simulations

Simulations proceed on a 13×13 grid. Initial locations of users are distributed following a 2D Spatial Poisson Point Process where each point placed by the point process represents one user, and intensity parameter is $\lambda = 0.1$. Users are placed using the following simple procedure.

Let N be the one side of a square simulation area, $N = 13$. Let S represent the entire bounded simulation area, where $|S| = N \times N$.

1. Sample the number of points within the area S by calculating Poisson($|S|\lambda$)

2. Place the calculated number of points within the area S, distributed uniformly.

Therefore, the expected number of points (or users) within the entire simulation area S is given by $|S|\lambda$ and user density is $\lambda \frac{\text{users}}{\text{grid}}$.

The learning rate and discount factor for Q-Learning are set to $\alpha = 0.7$ and $\gamma = 0.6$, and the exploration probability for the ϵ-greedy policy is set to $\epsilon = 0.15$, determined empirically following a coarse grid-search. Figure 4.7 illustrates a sample episode. Simulations are repeated for 2 and 4 aAPs with three inter-agent communication schemes: multiple-access, ideal, and naive estimation with no explicit communication between aAPs at all. Simulations proceed in two scenarios: (i) aAPs operating in an environment where user locations are fixed to demonstrate convergence characteristics of the algorithm, (ii) trained aAPs operating in a non-stationary environment, where a random user moves randomly to one of the adjacent grid squares, both sampled uniformly at the end of each episode with a probability of $p = 0.1$ to show the dynamics in aRAN. All presented results are averaged over 100 randomized runs.

Figure 4.8 shows the episode lengths to serve all users against episode the number in a stationary environment to display convergence characteristics. Naive estimation eventually converging to a similar performance with the other schemes shows that it can partition tasks
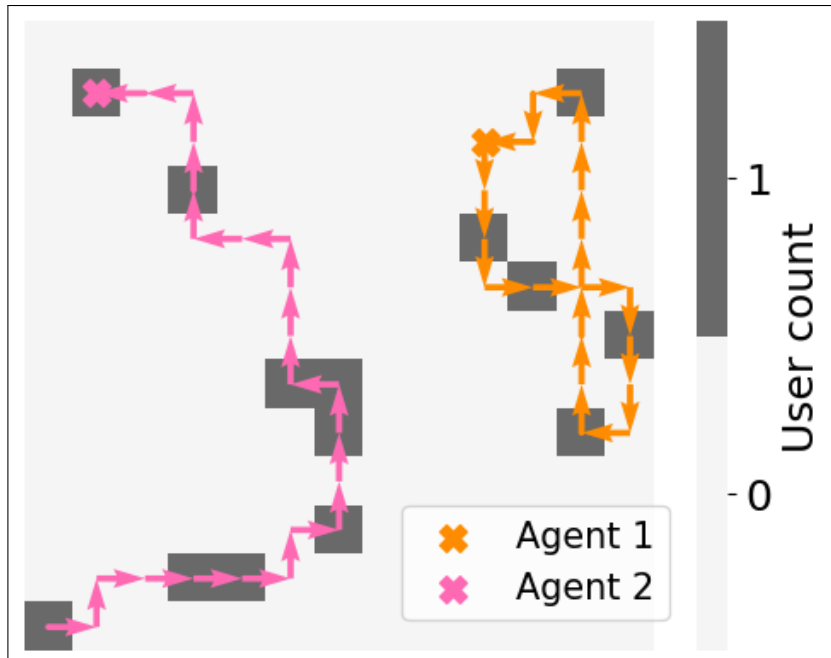
Figure 4.7: An illustrative episode showing the environment and aAP trajectories.
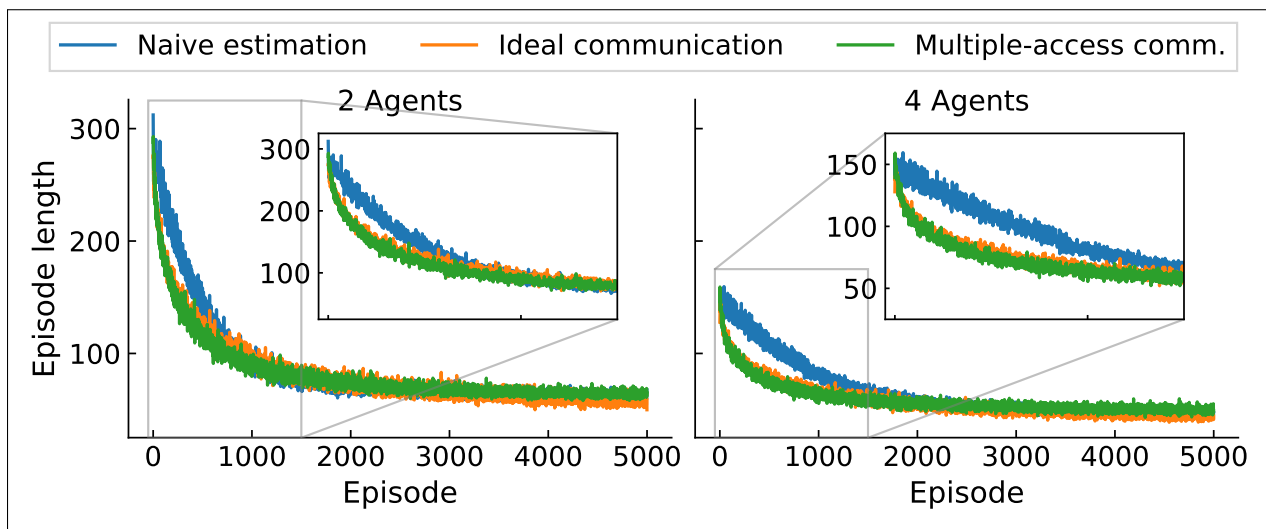
Figure 4.8: Episode lengths when aAPs are operating in a stationary environment showing convergence. Averaged over 100 runs.

effectively in a static environment. Delayed information from multiple-access inter-agent communications have relatively little effect as we expected, since the inter-aAP communication delays from the round-robin are small in the general time scale of the system.

Particularly, inter-aAP communication improves the convergence speed by enhancing the RL and this collaborative multi-agent system of aRAN.
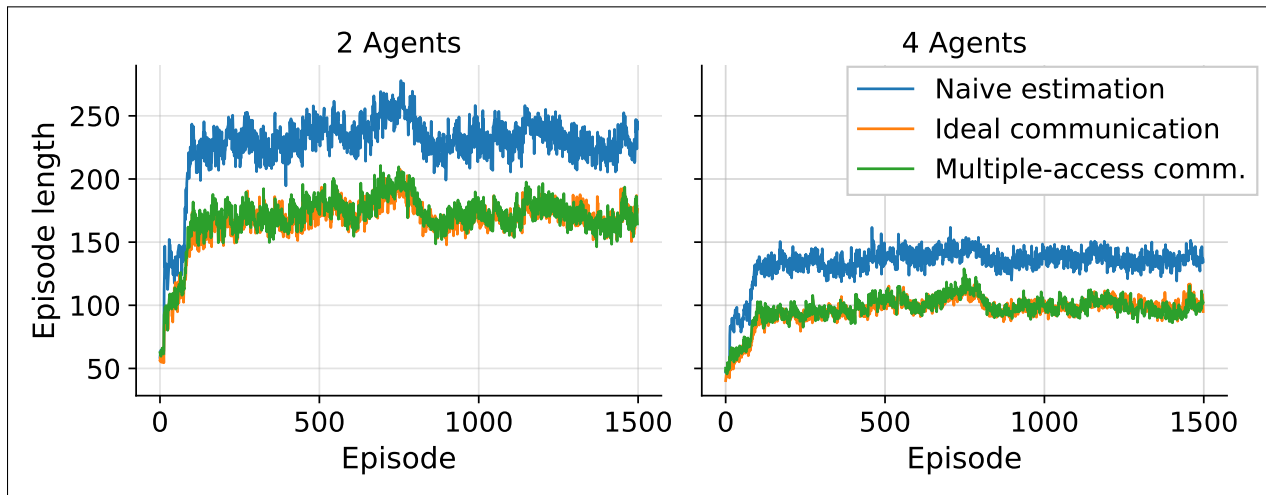
Figure 4.9: Episode lengths when users can move, showing ability to adaptively track changes in the environment. Averaged over 100 runs.

Figure 4.9 shows the length of episode vs. episode number for aAPs operating in a non-stationary environment, which demonstrates the capability of tracking dynamic changes. More aAPs operating in the environment and inter-aAP communication enhances adaptability to traffic dynamics. Naive estimation scheme shows the worst performance, as it has the worst convergence rate as seen in Figure 4.8. The demand for the effective reconvergence of the RL algorithms in a highly dynamic environment is met thanks to the improved convergence characteristics of the proposed RL scheme.

As for fairness, we are employing a RL based solution where we designed our RL system carefully to achieve fair trajectories without any analytical guarantees towards fairness. Therefore we present the results from our simulations to demonstrate the achieved fairness. Figure 4.10 shows the distribution of the total service each user receives within 250000 time-slots from 4 aAPs, which as expected is uniform among users. This is because aAPs are not rewarded for providing more than 1 unit of service per episode to a user, therefore they learn not to serve more than 1 unit of service per user within an episode.

72
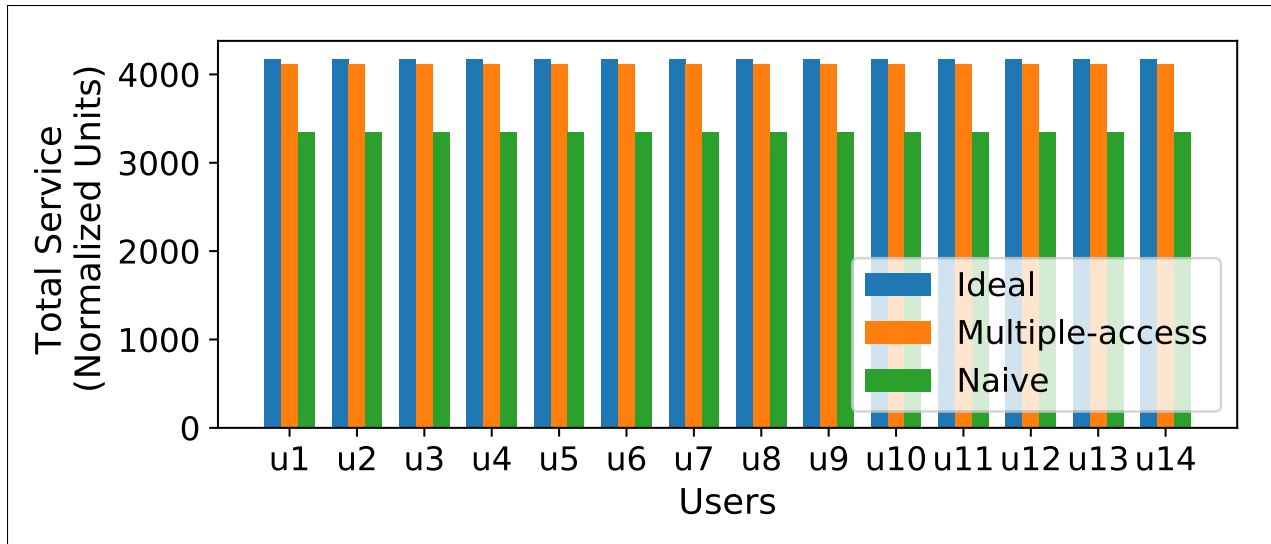
Figure 4.10: Total service each of the 14 users receives in 250000 time-slots from 4 aAPs, under each inter-aAP communication scheme.

4.6 Conclusions

Establishing an aRAN for flexible, dynamic, rapidly deployed wireless infrastructures under a dynamic operating environment without proper models and insufficient traffic information fundamentally differs from traditional wireless network design. We successfully, and more practically, develop effective trajectories for multiple decentralized collaborative aAPs to form an aRAN in dynamic operating environments by adapting RL and introducing a novel learning scheme which can exploit inter-agent communication and empirically ensure fairness of service among users.

Chapter 5: Enhanced Multi-Agent Mobile Aerial RAN Deployment

5.1 Introduction

Work presented in Chapter 4 models the trajectory optimization problem for multiple decentralized aerial Access Points (aAP) forming an aerial Radio Access Network (aRAN) as a Reinforcement Learning problem. Each agent follows a Markov Decision Process (MDP) defined over atomic actions of small movements where trajectories are built incrementally at each step through sequential decision over atomic actions.

Works in the literature focus on the control of the UAV trajectories thorough atomic actions [33] as well. However, flight control of a UAV to move it from a coordinate to another is an orthogonal problem to service optimization for the users. In this chapter it is assumed without loss of generality that a flight controller exists to take the UAVs to a given coordinate through the shortest path when these temporally extended actions are called.

Building trajectories incrementally, one atomic action at a time also introduces challenges in decentralized collaboration among agents. Even if agents have access to an inter-agent communication channel as in Subsection 4.4.7, they can't announce their future trajectories to other agents unless a computationally expensive long-horizon planning algorithm is employed by each agent individually which limits the scalability over the number of agents, resulting in duplicate work and reduced collaboration efficiency.

Temporal Abstraction is a method that allows agents to learn and follow a policy over temporally-extended actions of varying lengths. This violates the first-order dependency property of MDPs, resulting in a Semi-Markov Decision Process (SMDP) [112, 113, 114].

However many RL methods such as Time-Difference Learning and Q-Learning can also be defined over SMDPs [115].

In this chapter a method for collaborative trajectory optimization of multiple decentralized aAPs forming an aRAN is developed. We demonstrate improved convergence characteristics with the application of temporal abstraction within the options framework for rapid deployment, following a method inspired by the landmark-state methods [34, 35]. Options framework is then further utilized to improve decentralized multi-agent collaboration towards a shared goal by maintaining a set of temporally-extended actions to be used as a shared vocabulary to achieve effective and scalable decentralized planning under bounded inter-agent communication.

5.2 Network System Architecture

This chapter follows the same generally applicable abstracted system architecture presented in Section 4.2, with an important modification to the air interface between an aAP and users.

5.2.1 Modified Air Interface Between aAPs and Users

Given a reservation-based multiple access scheme, one unit of service is abstractly defined as the service that one user receives from an aAP in one time-slot. Sensing and communication range of aAPs are defined as a 3×3 region with aAP in the center. As illustrated by the time-diagram given in Figure 4.2b, at the beginning of each time-slot aAPs sense the traffic demand in their 3×3 service range, and then poll all the sensed users within range.

The one unit of service achievable in one time-step is shared uniformly by all the users within range, each receiving an equal fraction of it. If one user is within the service range of multiple aAPs, it associates with only one aAP, breaking ties randomly. Polling here repre-
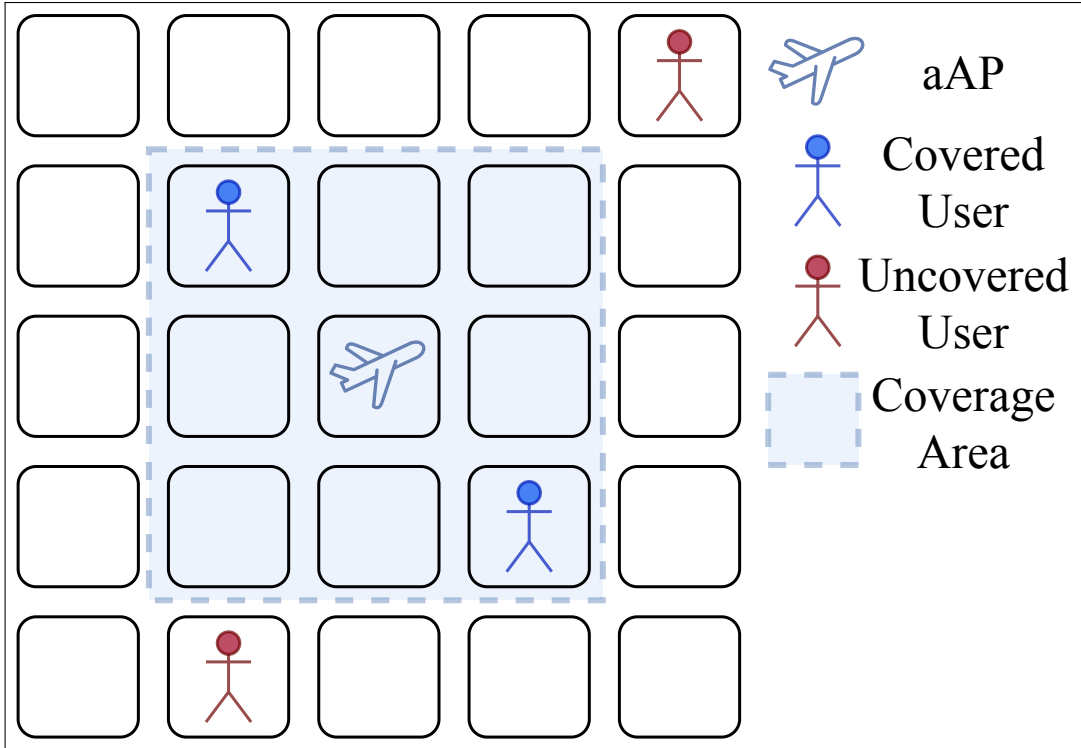
**Figure 5.1:** A diagram illustrating the 3×3 service range.

sents a general reservation-based multiple access such as the IEEE 802.11 mechanism [107]. Figure 5.1 illustrates this service area.

5.3 Background

In this section, tools employed in the development of the proposed method; temporal abstraction, SMDPs, and the options framework are introduced.

5.3.1 SMDPs and Temporal Abstraction

Semi-Markov Decision Processes (SMDP) [112, 114, 113] are extensions of Markov Decision Processes (MDP) to model continuous-time discrete event systems where actions take variable amounts of time to model temporally-extended courses of action [115]. Figure 5.2 illustrates the difference by depicting state trajectories of an MDP and an SMDP over time.
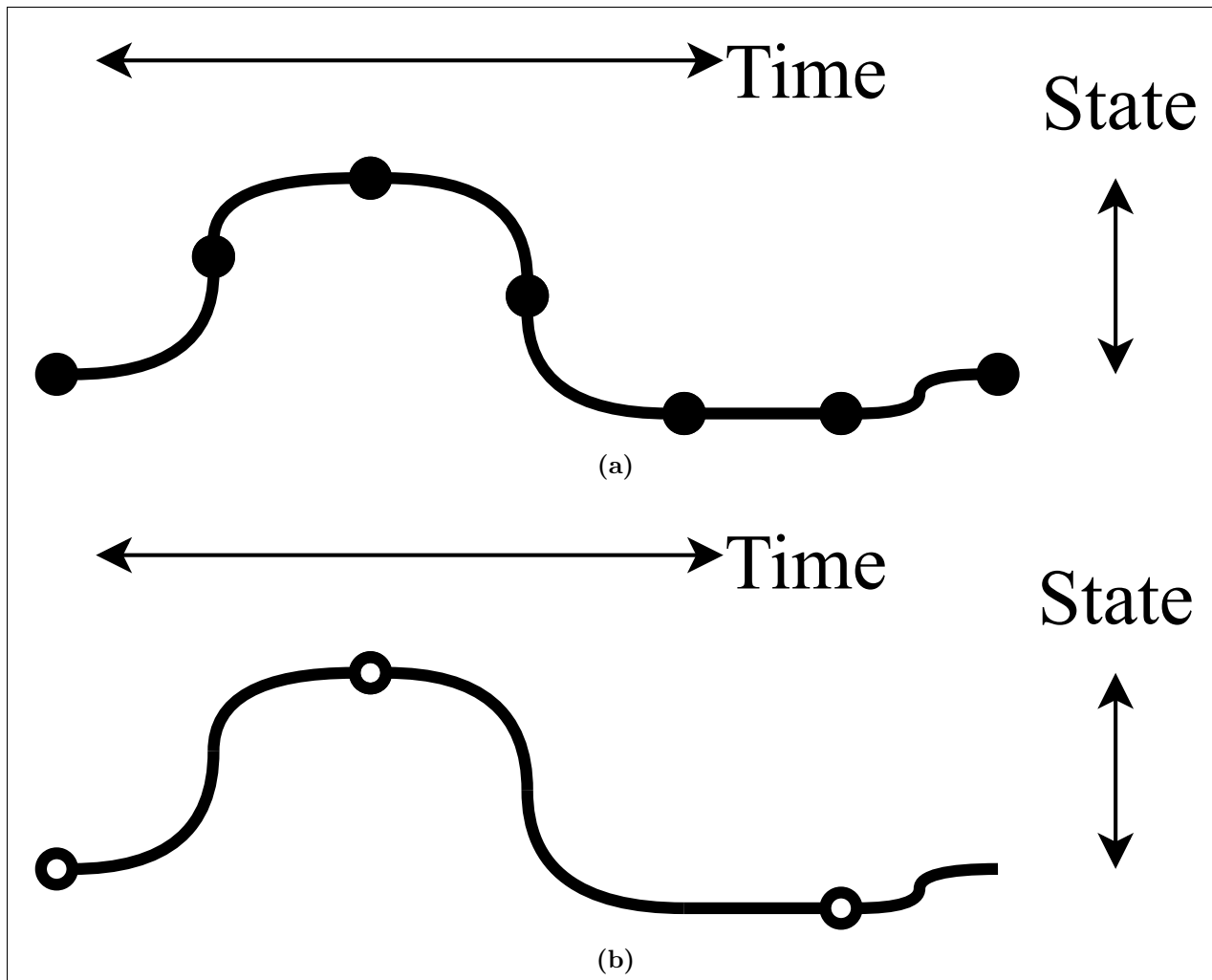
Figure 5.2: An illustrative example of state trajectories over time for; (a) an MDP and (b) an SMDP. The state trajectory of the MDP is comprised of small (atomic) discrete-time transitions. In comparison, the state trajectories of the SMDP is comprised of larger, continous-time transitions. Adopted from [115].

The state trajectory of the MDP as seen in Figure 5.2a is comprised of small (atomic) discrete-time transitions while that of the SMDP as seen in Figure 5.2b is comprised of larger, continuous-time transitions.

Temporally-extended actions can span different ranges of time scales. Throughout our daily lives we make many decisions over temporally-extended actions. For example, deciding what to cook for dinner requires consideration towards what ingredients are available, skill

level, dietary restrictions of the diners, and so on. After making the decision to cook, the temporally-extended action of cooking involves many smaller steps such as opening the fridge door, gathering and preparing ingredients, turning the stove on, and so on. Each of these steps are also made up of a sequence of smaller steps, down to the most minute detail like individual muscle contractions. Temporal abstraction provides a way to handle such multiple overlapping time scales and it's exploration in AI literature goes as far as 1970s [115].

5.3.2 Options
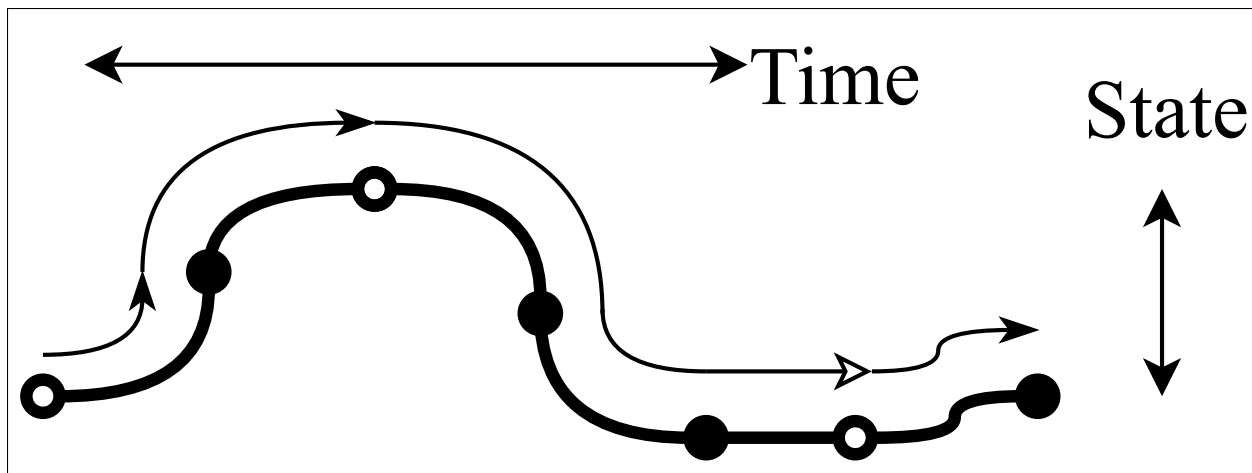
Figure 5.3: An illustrative example of state trajectories over time for and MDP over options. Options enable the MDP to take advantage of both small, discrete actions and large, continuous-time actions. Adopted from [115].

Options enable incorporating temporal abstraction into the RL framework in a general and natural way [115, 116].

Let $h_{t,T} = s_t, a_t, r_{t+1}, a_{t+1}, \ldots, r_T, s_T$ be the history sequence between times $t \leq T$ and T. Let ρ be the set of all possible histories given a Markov Decision Process. An option $\omega = \langle \mathcal{I}, \pi, \beta \rangle$ is defined by an initiation set $\mathcal{I} \subseteq \mathcal{S}$, policy $\pi : \rho \times \mathcal{A} \to \left[0, 1\right]$, and termination condition $\beta : \rho \to \left[0, 1\right]$. An option ω is available in a state $\mathbf{s}$ if and only if $\mathbf{s} \in \mathcal{I}$. If ω is chosen at state $\mathbf{s}_t$, next action a_t is determined following policy $\pi(\mathbf{s}_t, \cdot)$. Following action a_t,

the environment then transitions into $\mathbf{s}_{t+1}$, and ω is terminated following the termination condition with probability $\beta(h_{t,t+1})$. If not terminated, following actions are kept being chosen following the policy π, until the termination condition is satisfied. Options as defined here are called semi-Markov, because they depend on the history sequence h_t, T, rather than only the current state as Markov condition is defined. Policy π of an option ω can be defined as a sub-goal, and can be learned over the limited state-space defined by initiation set $\mathcal{I}$ and termination condition β following RL methods [115]. Figure 5.3 illustrates a state trajectory for an MDP over options, taking advantage of both atomic (filled in dots) and temporally-extended (empty dots) actions seen in Figure 5.2.

It is also shown that the use of options can improve the convergence rate of RL tasks [115, 117, 116]. Furthermore, for decentralized multi-agent scenarios with a shared goal, they provide a shared vocabulary of temporally-extended tasks that can be used to plan and collaborate effectively with limited inter-agent communication.

5.3.3   Hierarchical RL

Hierarchical RL can be utilized to separate the higher-level planning of which users or areas to serve and in what order, and the lower-level execution of this plan through the control of agents over atomic actions. Figure 5.4 provides a simple example of such a system, where the higher-level controller executes the planning and strategizing over the set of temporally extended actions and the lower-level controller handles the execution of the temporally-extended actions decided by the higher-level controller.

As it was stated in the Subsection 5.3.2, an option ω can be defined with a sub-goal, and a policy can be learned over the limited state-space defined by initiation set I and termination condition β following RL methods to achieve this sub-goal. This learning of policies to achieve the sub-goals over the set of primitive (atomic) actions is the lower-level execution part of the hierarchical model, whereas the higher-level planning part of the Hierarchical
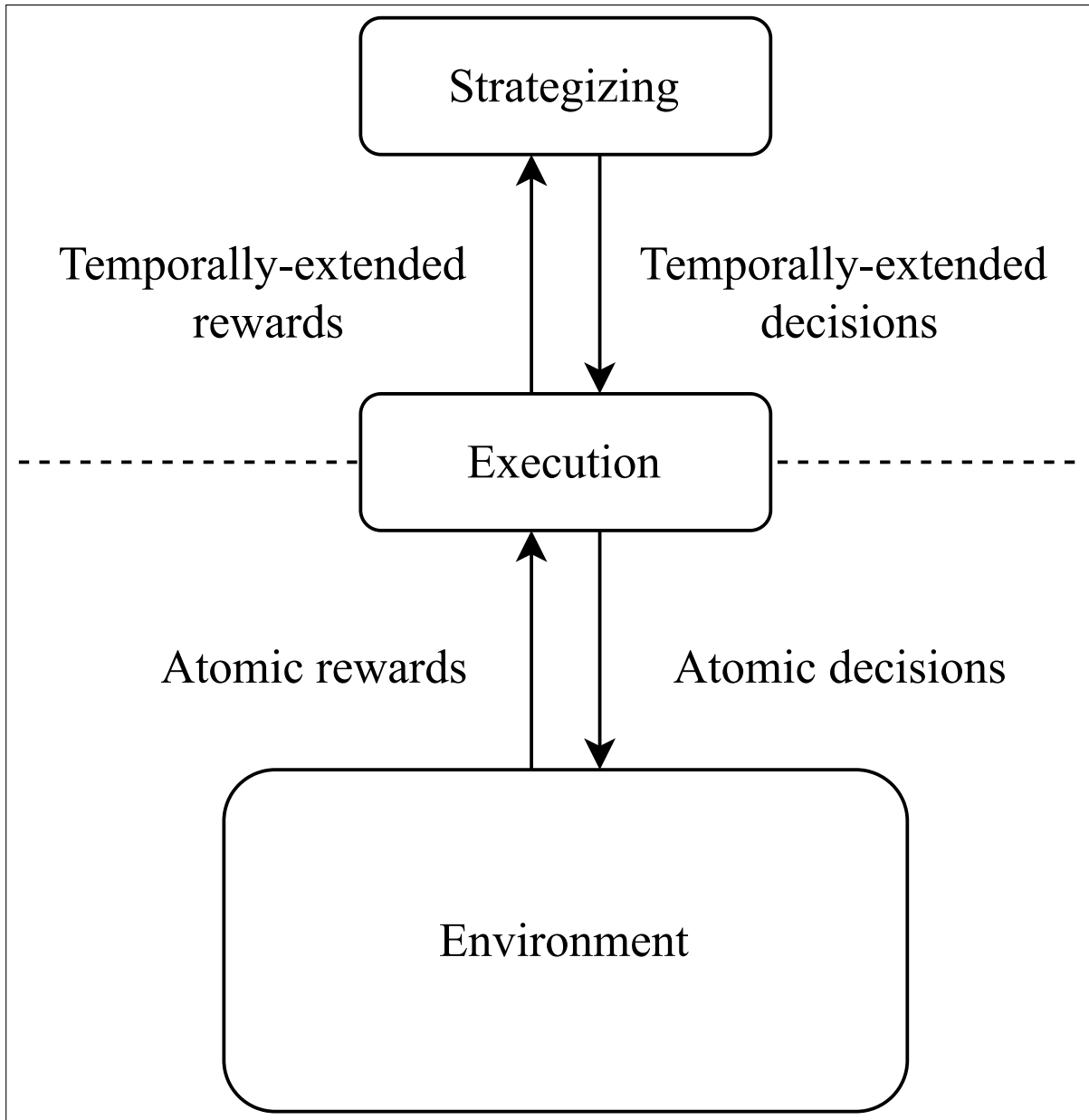
Figure 5.4: System diagram illustrating a simple hiearchical RL system. Higher level controller strategizes over the available set of temporally-extended actions, while the lower-level controller handles the interaction with the environment and the execution of the decisions made by the higher-level controller.

RL controller is the over-arching policy learned over the option set. Literature shows that standard RL methods such as TD-learning and Q-learning can be defined for SMDPs and applied over options [115, 116]. It should be noted that convergence guarantees in this case

apply to the optimal policy over the option set, which may be inferior to the optimal policy over the primitive action set [115, 116].

Looking again the the Figure 5.4, the higher-level controller represents the planner applied over options, while the lower-level controller represents the policies and RL controllers in place to achieve the individual sub-goals of options.

5.4   Proposed Algorithm

In this section, first the aAP trajectory optimization is proposed as a SMDP with the use of the options framework for a single agent. Then a method to facilitate collaboration among decentralized agents following the single agent model utilizing limited inter-agent communication is presented.

5.4.1   Single Agent Model

In order to be able to model the single-agent problem as a SMDP and take advantage of temporal abstraction, a set of options each with their own initiation sets, policies, and termination conditions, and a policy over this option set is defined. Figure 5.5 illustrates such a system, where the policy defined over options, π_Ω, determines ω_t, the option to execute at time t. Then the chosen option ω_t is executed following its policy π_{ω_t} defined over atomic actions, interacting with the environment directly. When the option ω_t terminates following its termination condition β_{ω_t}, the reward for executing it is calculated and used to update the policy over options π_Ω. Therefore, π_Ω is only updated after every option execution terminates, not after every atomic action.

Policy over options, π_Ω, is learned online following Q-Learning. State, action set, execution policy, and reward must be determined to define a functional Q-Learning system. As this is the policy over options, action set is replaced by the option set, and state and reward are not the atomic state and reward, but instead are defined over their atomic counterparts.
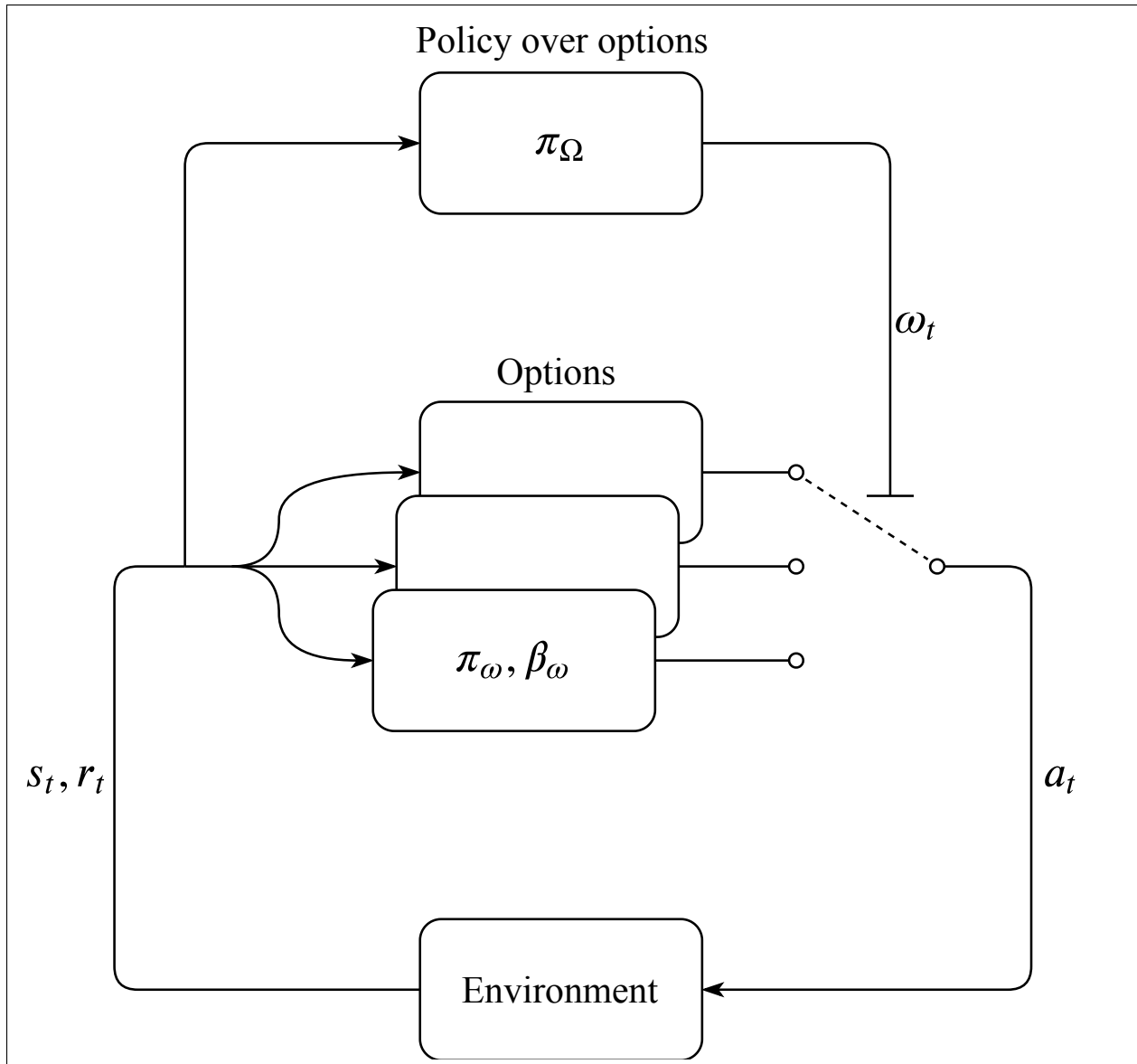
Figure 5.5: Diagram of the proposed system, learning a policy over options. π_Ω is the policy over options, determining which option to execute at time t, ω_t. The chosen option π_ω then interacts with the environment, where a_t is the atomic action determined by π_ω, r_t is the reward received from a_t, and s_t is the state after executing a_t.

5.4.1.1 Option Set

The option set $\mathcal{O}$ is defined over the atomic actions set for the aAP. Without loss of generality, the atomic action set $\mathcal{A}$ of the aAP is limited to 5 actions, moving forward, backward, left, right, and flying in place; $\mathcal{A} = \{\text{forward}, \text{backward}, \text{left}, \text{right}, \text{in-place}\}$.

The atomic state $\mathbf{s}_\omega$ under the option is defined as the coordinates $\mathbf{c}$ of the aAP, $\mathbf{s} = \begin{bmatrix} \mathbf{c} \end{bmatrix}$, where $\mathbf{c} = \begin{bmatrix} x & y & z \end{bmatrix}$. Within the context of this chapter, the height z is fixed to a constant value.

Inspired by the landmark-state approaches [34, 35], we create the option set $\mathcal{O}$ by dividing the operation area into uniform non-overlapping zones, or pseudo-cells. The center of each pseudo-cell is determined as a landmark-state $\mathbf{s}'_\omega$, and the option set consists of an option for every landmark-state that moves the agent to its coordinates following the shortest path. It should be noted that while it would be possible to learn the policies of options by defining reaching landmark-states as sub-goals as explained in Section 5.3.3, within the context of this chapter the option policies are deterministic and the flight control of the agent is assumed to be handled by a separate flight controller. Termination conditions β_ω for these options are defined as reaching the landmark-state. Figure 5.6 illustrates the pseudo-cell division and landmark-state assignment process with a pseudo-cell size of 3×3 grid tiles.

Let $\mathbf{s}'_{\omega_i}$ be the i-th landmark-state, and π_{ω_i} be the deterministic policy for the i-th option that corresponds to the i-th landmark-state. π_{ω_i} selects the next atomic action $a \in \mathcal{A}$ that brings the aAP to the coordinates of $\mathbf{s}'_{\omega_i}$ following the shortest path. When multiple shortest-paths exist, ties are broken randomly. π_{ω_i} is then defined as:

$$\pi_{\omega_i}(\mathbf{s}_\omega) = \text{next action } a \text{ on the shortest path from } \mathbf{s}_\omega \text{ to } \mathbf{s}'_{\omega_i}, \tag{5.1}$$

and the termination condition for the i-th option, $\beta_{\omega,i}$ is given by:

$$\beta_{\omega_i}(\mathbf{s}_\omega) = \begin{cases} 1, & \text{if } \mathbf{s}_\omega = \mathbf{s}'_{\omega_i}, \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$
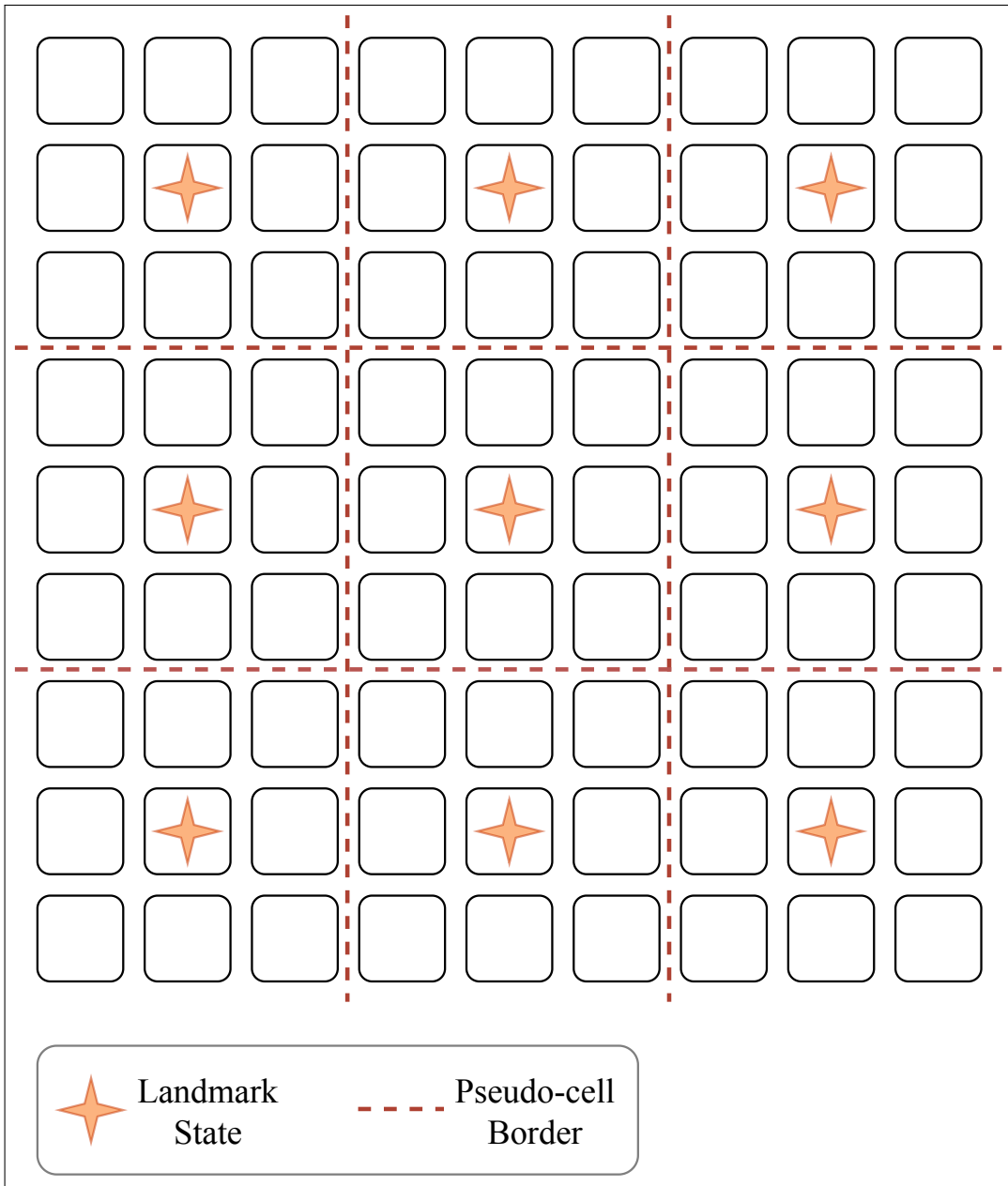
Figure 5.6: An illustration of the division of zones and landmark-states. The available area is divided into uniform non-overlapping sections of 3×3 grid tiles, and the center of each section is determined as a landmark.

Transitions between pseudo-cells are limited to spatially neighboring cells, focusing the state-space exploration to connected trajectories. This limitation is implemented through initiation sets, where the initiation set for the i-th option, $\mathcal{I}_{\omega_i}$, only includes the neighboring

zones (i.e. sharing a side) with the i-th zone represented by the option ω_i. A relaxation to this constraint will be introduced in Section 5.4.3.2 with the multi-agent extension.

5.4.2 State

The state over options $\mathbf{s}_\Omega$ is defined similarly to state defined in Section 4.4.6.1, with an abstraction that replaces the individual coordinates of the agent $\mathbf{c}\begin{bmatrix} x & y & z \end{bmatrix}$ with a higher level variable c_Ω that is defined as the index i of the landmark state $\mathbf{s}'_{\omega_i}$ that belongs to the zone agent is currently in. The state over options is then defined as:

$$\mathbf{s}_\Omega = \begin{bmatrix} c_\Omega & \mathbf{g} \end{bmatrix}, \tag{5.3}$$

where c_Ω is the index of the zone that includes $\mathbf{c}$ and $\mathbf{g}$ is the user status vector that has been defined in Section 4.4.6.1.

5.4.2.1 Policy

To balance the exploration of new information with exploitation of current information, an ϵ-greedy approach is used, which uniformly samples an option from $\mathcal{O}_{\mathbf{s}_\Omega}$ with the probability ϵ, and selects the option greedily otherwise. $\mathcal{O}_{\mathbf{s}_\Omega}$ here represents the subset of $\mathcal{O}$ which only includes options ω_i where $\mathbf{s}_\Omega$ is in their initiation sets $\mathcal{I}_{\omega_i}$.

5.4.2.2 Reward

Reward used here is an extension of the reward explained in Section 4.4.6.2 for a policy over option set $\mathcal{O}$. Reward for executing option ω is given by subtracting the number of time-steps it took to execute the option from the total number of users whose traffic demand were satisfied within the option. Let $|\cdot|$ be the set cardinality. Let $\mathcal{U}'$ be the set of unserved users that have been served by the aAP during the execution of option ω, and T_ω be the

number of time-steps it took to execute the option ω. A user being served within an option is determined by the change of its user service status flag defined by 4.4 to satisfied. Reward for executing option ω is given by:

$$R(\mathbf{s}_{\Omega}, \omega) = |\mathcal{U}'| - T_{\omega}. \tag{5.4}$$

5.4.3 Decentralized Multi-Agent Collaboration

Each agent individually follows the single-agent model described in Section 5.4.1, following their local references (information available to them locally) as they are decentralized. Agents improve their local references by sharing small control messages with each other. Making use of the options framework in this model means that all the agents have the same option set $\mathcal{O}$. This shared option set can be utilized as a shared vocabulary, and with the introduction of inter-agent communication, to improve the decentralized collaboration efficiency by reducing the amount of repeated work.

5.4.3.1 Inter-Agent Communication

Inter-agent communication occurs on a round-robin multiple-access channel as described in Section 4.4.7. It has already been shown in Chapter 4 that this round-robin channel is nearly as effective as an ideal communication channel when it is used to share control messages between agents to coordinate.

Agents multi-cast a control message to all other agents every time they terminate an option. This control message includes the observation of the agent for the recently concluded option, and the next option they will be executing. The observation consists of the initiating state, option executed, reward received from the environment, and the terminating state. Together with the next option, this is all the information necessary to backup under Q-Learning, allowing agents to synchronize their policies with limited communication.

Announcing the next option that will be executed to all the other agents also provides an opportunity for improved planning.

The control messages shared between agents are small, and they are shared infrequently as they are only shared when an option is terminated. Following this, and the results from Chapter 4, it is assumed within this chapter that the control messages arrive without delay.

5.4.3.2 Decentralized Planning and Collaboration

A simple planning method with a single-step horizon is adopted to improve the collaboration efficiency taking advantage of the limited inter-agent communication and shared option sets. The shared control messages mean that each agent knows the options all other agents are working towards executing. This information can be utilized by removing the options that are being executed by other agents from the set of available options the decision is based on. Let $\cdot \setminus \cdot$ be the set exclusion operator and $\mathcal{O}^o$ be the set of options that are currently being executed by agents. The set of free options $\mathcal{O}'$ is then defined as:

$$\mathcal{O}' = \mathcal{O} \setminus \mathcal{O}^o. \tag{5.5}$$

Furthermore, if there are no viable options in $\mathcal{O}'$ following the spatial-neighborhood constraint explained in Section 5.4.1.1, this constraint will be removed and the agent can chose the next option from the entire free option set $\mathcal{O}'$. This allows agents to leave heavily served ares of the environment quickly and serve other sections that are possibly being underserved. An important advantage of this simple planning method is that it does not require constant, reliable communication between agents. It can exploit inter-agent communication when it is available, however when inter-agent communication is unavailable as it can be expected intermittently in the field, it can still make a decisions based solely on the agents experiences.

5.5   Simulations

Simulations proceed on a 15×15 grid. The size of the service areas of aAPs are set as 3×3. Initial locations of users are distributed following a 2D Spatial Poisson Point Process where each point placed by the point process represents one user, and intensity parameter is $\lambda = 0.1$, where the expected number of users is given by $(15 \times 15)\lambda = 22.5$.
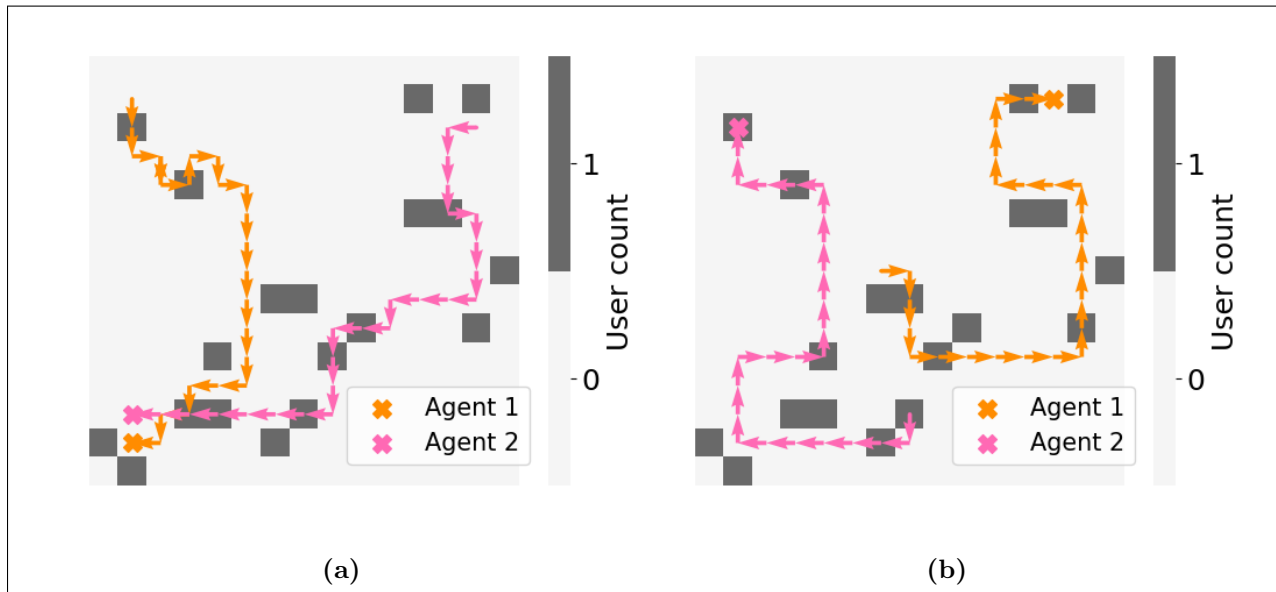

Figure 5.7: An illustrative episode showing the environment and aAP trajectories for; (a)MDP and (b)SMDP approaches.

The learning rate and discount factor for Q-Learning are set to $\alpha = 0.7$ and $\gamma = 0.6$ determined empirically as in to Chapter 4. As we are exploring the case with static users, therefore the exploration probability for the ϵ-greedy policy is set to $\epsilon = 0.01$ and further exploration is encouraged by using an optimistic Q initialization scheme [109], setting the initial Q values as $Q_o = 1$. Simulations are repeated for 2, 4 and 6 aAPs, comparing the SMDP approach explained in this chapter with the MDP based Ideal Communication case from Chapter 4 as a benchmark. Figure 5.7 illustrates two sample episodes for 2 agents following each of the two approaches. All presented results are have been averaged over 50 randomized runs.

Figure 5.8 shows the episode lengths to serve all users against episode the number in a stationary environment to display convergence characteristics. It is seen that for all cases the introduction of temporal abstraction and hierarchical RL have significantly improved the convergence rate and the early worst-case episodes. Furthermore, the simple planning method introduced in Section 5.4.3.2 have also improved the collaboration efficiency of the decentralized agents. As the number of agents increase the divide between two approaches gets wider, further supporting improved collaboration since increased number of agents means a higher chance of repeated work. Readers should be reminded that both approaches utilize the same level inter-agent communication, and these improvements are not a result of having access to more information.

5.6   Conclusions

This chapter presented a method for decentralized multi-agent trajectory optimization of aAPs to form an aRAN. Convergence rate and early performance is crucial for such a system to reduce the training time and start serving users as quickly as possible. The proposed SMDP based approach utilizing temporal-abstraction and hierarchical RL have significantly improved the convergence characteristics compared to a MDP based approach that operates over atomic actions, which is the common approach in literature for this problem. To the best of authors knowledge, this is the first work applying temporal-abstraction to the aerial wireless network deployment problem. Furthermore, a simple planning method is proposed to improve decentralized collaboration efficiency taking advantage of the shared option set and limited inter-agent communication. Simulation results verify that the proposed method improves the effectiveness of collaboration by reducing the amount of repeated work, and the effect size increases with the increasing number of agents.
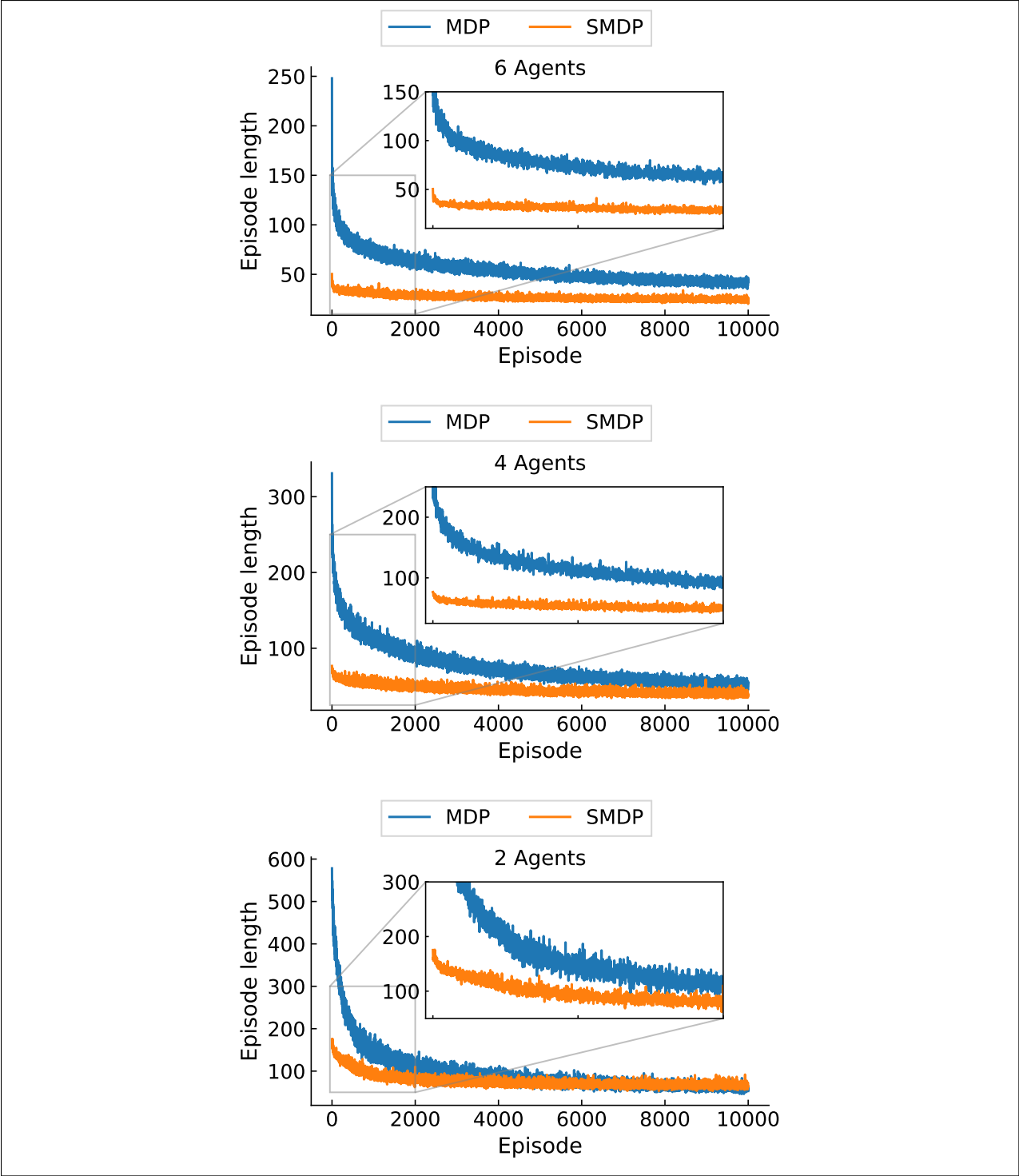
Figure 5.8: Convergence results for MDP and SMDP approaches for; 6,4, and 2 agents. Significantly improved convergence characteristics and improved collaboration efficiency as the number of agents increase can be observed.

Chapter 6: Concluding Remarks and Future Research

In this chapter concluding remarks regarding the previous chapters is presented, and future research plans and directions are detailed.

6.1 Concluding Remarks

This dissertation has its major contributions in both the socially aware analysis and the development of wireless network infrastructures for the applications that are challenging for traditional infrastructure deployments. Specifically, this dissertation first presents an investigation of the wireless network user data to generate actionable intelligence that can guide policy makers and point out the changing demands of wireless networks discovering user modalities outside the common user patterns, and then explores the use of aerial platforms for rapid deployment of flexible wireless networks to satisfy the mobility and reliability demands of the future wireless communications.

Chapter 2 studies a novel Call Detail Record dataset that includes data from refugee and non-refugee users focusing on discovering mobility and connectivity patterns for both individuals and the aggregate. Connectivity among users is tightly coupled with their social lives and employment status, and user mobility is very dependent on places of residence and employment, where these patterns can serve as markers of employment and social integration issues for the refugee users. Seeing distinct divergences in both connectivity patterns where refugee users interact with a much smaller portion of the nation, and mobility patterns where refugee users appear to be spatially segregated and move much more frequently and in shorter steps, suggest that refugee users are not very well integrated into the local social life and

may have less stable employment options. These results are also relevant for wireless network design, as they suggest existence of large groups whose connectivity needs significantly differ from the local population network infrastructures are often designed for.

Chapter 3 approaches the AP placement in genuine 3D problem for aerial wireless network deployment from the perspective of random graphs, particularly Random Geometric Graphs (RGGs). After the formulation of the optimization problem with the help of RGGs, an iterative algorithm that decomposes the intractable joint-optimization problem of AP positions into a sequence of individual optimization problems is developed. A procedure to determine the adequate number of APs for a desired level of connectivity is also presented. Detailed agent-based simulations showed that operation in full 3D space provides significant improvements specifically when avoiding obstacles to achieve line-of-sight and conforming to the environment are important, and the proposed iterative optimization scheme can effectively find good placements.

Chapter 4 presents a method to form a flexible, dynamic, and rapidly deployed wireless infrastructure through the formation of an aerial RAN. This method significantly relaxes the prior information assumptions from Chapter 3 and majority of prior art by requiring no models on traffic and user mobility and no prior knowledge of the user distribution, and the resulting problem differs from the traditional wireless network design fundamentally. A reinforcement learning based solution to control the trajectories of multiple decentralized collaborative aerial APs with a learning scheme that can exploit inter-agent communication to achieve effective collaboration is developed to form an aerial RAN operating over dynamic environments. This solution also uniquely considers the fairness of service among users, and is designed and empirically shown to provide equal service to all users.

Chapter 5 presents a method that extends the work from Chapter 4 to significantly improve the convergence characteristics and multi-agent collaboration effectiveness as convergence rate and early performance is crucial for such a system to reduce the training time

and start serving users as quickly as possible. A SMDP based approach utilizing temporal-abstraction and hierarchical RL that significantly improves the convergence characteristics is presented. The incorporated option set is then further utilized together with limited inter-agent communication to improve the effectiveness of decentralized multi-agent collaboration by providing agents a shared set of temporally-extended tasks that can be used as a shared vocabulary to coordinate. This chapter shows that a SMDP approach can bring us closer to a practically applicable system.

6.2 Future Research

Future research will focus on the modeling of aAPs as decentralized agents in a multi-agent system. One significant challenge is taking advantage of bounded/opportunistic inter-agent communication for the purposes of effective collaboration for decentralized agents. In practical systems, an inter-agent channel between all agents is unlikely to be present, which puts emphasis on being able to take advantage of inter-agent communication opportunistically whenever it is available. This also requires being able to plan with partial information of the system, as having all agents in communication with each other at every instance of planning is also not practical.

Another important challenge is handling stochastic/unreliable observations from the environment. Throughout this dissertation, the observations agents receive from the environment have been assumed to be accurate. However these observations can have many sources of error from sensor reliability to communication errors. Modeling and handling of these errors are crucial for a system to operate reliably.

One more avenue of potential improvement is through transfer learning and skill transfer. Training agents beforehand in a way they can generalize the learned policies to unknown environments quickly can significantly improve the deployment time necessary until the aRAN starts to provide a satisfactory level of service.

A final direction for future research is Federated Learning, where each decentralized agent contributes their individual experiences to improve a single shared model whenever communication with a centralized control center is available which can bring the advantages of centralized agents without the stringent communication requirements.

References

[1] Jian Xu, Thanuka L Wickramarathne, and Nitesh V Chawla. Representing higher-order dependencies in networks. *Science advances*, 2(5):e1600028, 2016.

[2] The GSM Association. The Importance of Mobile for Refugees: A Landscape of New Services and Approaches. Technical report, The GSM Association, 2017.

[3] Vincent D Blondel, Adeline Decuyper, and Gautier Krings. A survey of results on mobile phone datasets analysis. *EPJ data science*, 4(1):10, 2015.

[4] Nicola Baldo and Pau Closas. Disease outbreak detection by mobile network monitoring: a case study with the d4d datasets. *NetMob D4D challenge*, pages 1–4, 2013.

[5] Lorenzo Mari, Marino Gatto, Manuela Ciddio, Elhadji D Dia, Susanne H Sokolow, Giulio A De Leo, and Renato Casagrandi. Big-data-driven modeling unveils country-wide drivers of endemic schistosomiasis. *Scientific reports*, 7(1):1–11, 2017.

[6] Adrian M Tompkins and Nicky McCreesh. Migration statistics relevant for malaria transmission in senegal derived from mobile phone data and used in an agent-based migration model. *Geospatial health*, 11(1 Supp):408, 2016.

[7] Christelle Scharff, Khadidiatou Ndiaye, Meghan Jordan, Aminata Niang Diene, and Fatou Maria Drame. Human mobility during religious festivals and its implications on public health in senegal: A mobile dataset analysis. In *2015 IEEE Global Humanitarian Technology Conference (GHTC)*, pages 108–113. IEEE, 2015.

[8] A Lima, M De Domenico, V Pejovic, and Mirco Musolesi. Disease containment strategies based on mobility and information dissemination. *Scientific reports*, 5:10650, 2015.

[9] Brian Tomaszewski. *Geographic information systems (GIS) for disaster management.* CRC Press, 2014.

[10] Eduardo Alejandro Martinez-Cesena, Pierluigi Mancarella, Mamadou Ndiaye, and Markus Schläpfer. Using mobile phone data for electricity infrastructure planning. *arXiv preprint arXiv:1504.03899*, 2015.

[11] Neeti Pokhriyal and Damien Christophe Jacques. Combining disparate data sources for improved poverty prediction and mapping. *Proceedings of the National Academy of Sciences*, 114(46):E9783–E9792, 2017.

[12] Didem Gundogdu, Ozlem D Incel, Albert A Salah, and Bruno Lepri. Countrywide arrhythmia: emergency event detection using mobile phone data. *EPJ Data Science*, 5(1):25, 2016.

[13] Ramona Trestian, Purav Shah, H Nguyen, Q-T Vien, Orhan Gemikonakli, and Balbir Barn. Towards connecting people, locations and real-world events in a cellular network. *Telematics and Informatics*, 34(1):244–271, 2017.

[14] Yves-Alexandre de Montjoye, Zbigniew Smoreda, Romain Trinquart, Cezary Ziemlicki, and Vincent D Blondel. D4d-senegal: the second mobile phone data for development challenge. *arXiv preprint arXiv:1407.4885*, 2014.

[15] Clio Andris and Luis MA Bettencourt. Development, information and social connectivity in côte d'ivoire. *Infrastructure Complexity*, 1(1):1, 2014.

[16] Albert Ali Salah, Alex Pentland, Bruno Lepri, Emmanuel Letouzé, Patrick Vinck, Yves-Alexandre de Montjoye, Xiaowen Dong, and Özge Dağdelen. Data for refugees: The d4r challenge on mobility of syrian refugees in turkey. *arXiv preprint arXiv:1807.00523*, 2018.

[17] Albert Ali Salah, M. Tarık Altuncu, Selim Balcisoy, Erika Frydenlund, Marco Mamei, Mehmet Ali Akyol, Kerem Yavuz Arslanlı, Ivon Bensason, Christine Boshuijzen-van Burken, Paolo Bosetti, Jeremy Boy, Tugba Bozcaga, Seyit Mümin Cilasun, Oğuz Işık, Sibel Kalaycıoğlu, Ayse Seyyide Kaptaner, Ilker Kayi, Özgün Ozan Kılıç, Berat Kjamili, Huseyin Kucukali, Aaron Martin, Marco Lippi, Francesca Pancotto, Daniel Rhoads, Nur Sevencan, Ervin Sezgin, Albert Solé-Ribalta, Harald Sterly, Elif Surer, Tuğba Taşkaya Temizel, Semih Tümen, and Ismail Uluturk. *Policy Implications of the D4R Challenge*, pages 477–495. Springer International Publishing, Cham, 2019.

[18] Sameera Poduri and Sundeep Pattem. Sensor Network Configuration and the Curse of Dimensionality. In *The Third IEEE Workshop on Embedded Networked Sensors*, pages 1–5, 2006.

[19] Thomas Hales, Mark Adams, Gertrud Bauer, Tat Dat Dang, John Harrison, Hoang Le Truong, Cezary Kaliszyk, Victor Magron, Sean McLaughlin, Tat Thang Nguyen, et al. A formal proof of the kepler conjecture. In *Forum of Mathematics, Pi*, volume 5. Cambridge University Press, 2017.

[20] R. Irem Bor-Yaliniz, Amr El-Keyi, and Halim Yanikomeroglu. Efficient 3-D placement of an aerial base station in next generation cellular networks. In *2016 IEEE International Conference on Communications, ICC 2016*, 2016.

[21] 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.

[22] Elham Kalantari, Halim Yanikomeroglu, and Abbas Yongacoglu. On the number and 3D placement of drone base stations in wireless cellular networks. In *IEEE Vehicular Technology Conference*, 2017.

[23] Mohammad Mahdi Azari, Fernando Rosas, Kwang Cheng Chen, and Sofie Pollin. Ultra Reliable UAV Communication Using Altitude and Cooperation Diversity. *IEEE Transactions on Communications*, 66(1):330–344, 2018.

[24] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Merouane Debbah. Drone small cells in the clouds: design, deployment and performance analysis. In *2015 IEEE Global Communications Conference, GLOBECOM 2015*, 2015.

[25] Elham Kalantari, Muhammad Zeeshan Shakir, Halim Yanikomeroglu, and Abbas Yongacoglu. Backhaul-aware robust 3D drone placement in 5G+ wireless networks. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 109–114. IEEE, may 2017.

[26] 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, 2016.

[27] C. T. Cicek, H. Gultekin, B. Tavli, and H. Yanikomeroglu. UAV base station location optimization for next generation wireless networks: Overview and future research directions. In *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*, pages 1–6, Feb 2019.

[28] Q. Wu, L. Liu, and R. Zhang. Fundamental trade-offs in communication and trajectory design for UAV-enabled wireless network. *IEEE Wireless Communications*, 26(1):36–44, February 2019.

[29] Harald Bayerlein, Paul De Kerret, and David Gesbert. Trajectory optimization for autonomous flying base station via reinforcement learning. In *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5. IEEE, Jun 2018.

[30] Rozhina Ghanavi, Elham Kalantari, Maryam Sabbaghian, Halim Yanikomeroglu, and Abbas Yongacoglu. Efficient 3D aerial base station placement considering users mobility by reinforcement learning. In *IEEE Wireless Communications and Networking Conference, WCNC*, 2018.

[31] Xiao Liu, Yuanwei Liu, Yue Chen, and Lajos Hanzo. Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach. *IEEE Transactions on Vehicular Technology*, 2019.

[32] Jingzhi Hu, Hongliang Zhang, and Lingyang Song. Reinforcement learning for decentralized trajectory design in cellular UAV networks with sense-and-send protocol. *IEEE Internet of Things Journal*, 6(4):6177–6189, Aug 2019.

[33] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah. A tutorial on uavs for wireless networks: Applications, challenges, and open problems. *IEEE Communications Surveys Tutorials*, 21(3):2334–2360, thirdquarter 2019.

[34] Timothy A Mann, Shie Mannor, and Doina Precup. Approximate value iteration with temporally extended actions. *Journal of Artificial Intelligence Research*, 53:375–438, 2015.

[35] Nicholas Denis and Maia Fraser. Options in multi-task reinforcement learning - transfer via reflection. In Marie-Jean Meurs and Frank Rudzicz, editors, *Advances in Artificial Intelligence*, pages 225–237, Cham, 2019. Springer International Publishing.

[36] Laura Alessandretti, Sune Lehmann, and Andrea Baronchelli. Understanding the interplay between social and spatial behaviour. *EPJ Data Science*, 7(1):36, 2018.

[37] Laura Alessandretti, Piotr Sapiezynski, Vedran Sekara, Sune Lehmann, and Andrea Baronchelli. Evidence for a conserved quantity in human mobility. *Nature Human Behaviour*, 2(7):485–491, 2018.

[38] X. Liu, Y. Liu, Y. Chen, and L. Hanzo. Trajectory design and power control for Multi-UAV assisted wireless networks: A machine learning approach. *IEEE Transactions on Vehicular Technology*, 68(8):7957–7969, Aug 2019.

[39] Vittoria Colizza, Alain Barrat, Marc Barthelemy, Alain-Jacques Valleron, and Alessandro Vespignani. Modeling the worldwide spread of pandemic influenza: Baseline case and containment interventions. *PLOS Medicine*, 4(1):1–16, 01 2007.

[40] S Eubank, H Guclu, VSA Kumar, M Marathe, A Srinivasan, Z Toroczkai, and N Wang. Controlling epidemics in realistic urban social networks. *Nature*, 429(180-184):25, 2004.

[41] L. Hufnagel, D. Brockmann, and T. Geisel. Forecast and control of epidemics in a globalized world. *Proceedings of the National Academy of Sciences*, 101(42):15124–15129, 2004.

[42] Jon Kleinberg. The wireless epidemic. *Nature*, 449(7160):287–288, 2007.

[43] Mark W Horner and Morton E O'Kelly. Embedding economies of scale concepts for hub network design. *Journal of Transport Geography*, 9(4):255–265, 2001.

[44] Ryuichi Kitamura, Cynthia Chen, Ram M Pendyala, and Ravi Narayanan. Micro-simulation of daily activity-travel patterns for travel demand forecasting. *Transportation*, 27(1):25–51, 2000.

[45] Dirk Brockmann, Lars Hufnagel, and Theo Geisel. The scaling laws of human travel. *Nature*, 439(7075):462–465, 2006.

[46] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *nature*, 453(7196):779–782, 2008.

[47] Pierre Deville, Chaoming Song, Nathan Eagle, Vincent D. Blondel, Albert-László Barabási, and Dashun Wang. Scaling identity connects human mobility and social interactions. *Proceedings of the National Academy of Sciences*, 113(26):7047–7052, 2016.

[48] Mark EJ Newman. Analysis of weighted networks. *Physical review E*, 70(5):056131, 2004.

[49] Andrea De Montis, Marc Barthélemy, Alessandro Chessa, and Alessandro Vespignani. The structure of interurban traffic: a weighted network analysis. *Environment and Planning B: Planning and Design*, 34(5):905–924, 2007.

[50] Gerardo Chowell, James M Hyman, Stephen Eubank, and Carlos Castillo-Chavez. Scaling laws for the movement of people between locations in a large city. *Physical Review E*, 68(6):066102, 2003.

[51] Ganesh Bagler. Analysis of the airport network of india as a complex weighted network. *Physica A: Statistical Mechanics and its Applications*, 387(12):2972–2980, 2008.

[52] Jian Xu, Mandana Saebi, Bruno Ribeiro, Lance M Kaplan, and Nitesh V Chawla. Detecting anomalies in sequential data with higher-order networks. *arXiv preprint arXiv:1712.09658*, 2017.

[53] Ismail Uluturk, Ismail Uysal, and Onur Varol. Refugee integration in Turkey: A study of mobile phone data for D4R challenge. In *Data for Refugees Challenge Workshop*, 2019.

[54] Serap Keles, Oddgeir Friborg, Thormod Idsøe, Selcuk Sirin, and Brit Oppedal. Depression among unaccompanied minor refugees: The relative contribution of general and acculturation-specific daily hassles. *Ethnicity & health*, 21(3):300–317, 2016.

[55] Selcuk R Sirin and Lauren Rogers-Sirin. *The educational and mental health needs of Syrian refugee children.* Migration Policy Institute Washington, DC, 2015.

[56] Regional Refugee & Resilience Plan 2018-2019 in Response to the Syria Crisis. 3RP Regional Strategic Overview 2018-19. Technical report, The 3RP, 2017.

[57] Michael J Greenwood. Human migration: Theory, models, and empirical studies. *Journal of regional Science*, 25(4):521–544, 1985.

[58] Richard Black, W Neil Adger, Nigel W Arnell, Stefan Dercon, Andrew Geddes, and David Thomas. The effect of environmental change on human migration. *Global environmental change*, 21:S3–S11, 2011.

[59] Luca Pappalardo, Filippo Simini, Salvatore Rinzivillo, Dino Pedreschi, Fosca Giannotti, and Albert-László Barabási. Returners and explorers dichotomy in human mobility. *Nature communications*, 6:8166, 2015.

[60] Filippo Simini, Marta C González, Amos Maritan, and Albert-László Barabási. A universal model for mobility and migration patterns. *Nature*, 484(7392):96, 2012.

[61] Julián Candia, Marta C González, Pu Wang, Timothy Schoenharl, Greg Madey, and Albert-László Barabási. Uncovering individual and collective human dynamics from mobile phone records. *Journal of physics A: mathematical and theoretical*, 41(22):224015, 2008.

[62] Vincent D Blondel, Markus Esch, Connie Chan, Fabrice Clérot, Pierre Deville, Etienne Huens, Frédéric Morlot, Zbigniew Smoreda, and Cezary Ziemlicki. Data for development: the d4d challenge on mobile phone data. *arXiv preprint arXiv:1210.0137*, 2012.

[63] Huina Mao, Xin Shuai, Yong-Yeol Ahn, and Johan Bollen. Mobile communications reveal the regional economy in côte d'ivoire. *Proc. of NetMob*, 2013.

[64] Sanja Šćepanović, Igor Mishkovski, Pan Hui, Jukka K Nurminen, and Antti Ylä-Jääski. Mobile phone call data as a regional socio-economic proxy indicator. *PloS one*, 10(4):e0124160, 2015.

[65] Filippo Maria Bianchi, Antonello Rizzi, Alireza Sadeghian, and Corrado Moiso. Identifying user habits through data mining on call data records. *Engineering Applications of Artificial Intelligence*, 54:49–61, 2016.

[66] Gianni Barlacchi, Marco De Nadai, Roberto Larcher, Antonio Casella, Cristiana Chitic, Giovanni Torrisi, Fabrizio Antonelli, Alessandro Vespignani, Alex Pentland, and Bruno Lepri. A multi-source dataset of urban life in the city of milan and the province of trentino. *Scientific data*, 2:150055, 2015.

[67] Feng Liu, Davy Janssens, JianXun Cui, YunPeng Wang, Geert Wets, and Mario Cools. Building a validation measure for activity-based transportation models based on mobile phone data. *Expert Systems with Applications*, 41(14):6174–6189, 2014.

[68] Antonio Lima, Manlio De Domenico, Veljko Pejovic, and Mirco Musolesi. Exploiting cellular data for disease containment and information campaigns strategies in country-wide epidemics. *arXiv preprint arXiv:1306.4534*, 2013.

[69] Jian Xu, Thanuka L Wickramarathne, and Nitesh V Chawla. Representing higher-order dependencies in networks. *Science advances*, 2(5):e1600028, 2016.

[70] Jian Xu, Mandana Saebi, Bruno Ribeiro, Lance M Kaplan, and Nitesh V Chawla. Detecting anomalies in sequential data with higher-order networks. *arXiv preprint arXiv:1712.09658*, 2017.

[71] Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd, et al. The pagerank citation ranking: Bringing order to the web. 1998.

[72] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *nature*, 453(7196):779, 2008.

[73] I. Bucaille, S. Héthuin, A. Munari, R. Hermenier, T. Rasheed, and S. Allsopp. Rapidly deployable network for tactical applications: Aerial base station with opportunistic links for unattended and temporary events absolute example. In *MILCOM 2013 - 2013 IEEE Military Communications Conference*, pages 1116–1120, Nov 2013.

[74] I. Bor-Yaliniz and H. Yanikomeroglu. The new frontier in ran heterogeneity: Multi-tier drone-cells. *IEEE Communications Magazine*, 54(11):48–55, November 2016.

[75] I. Bucaille, S. Héthuin, A. Munari, R. Hermenier, T. Rasheed, and S. Allsopp. Rapidly deployable network for tactical applications: Aerial base station with opportunistic links for unattended and temporary events absolute example. In *MILCOM 2013 - 2013 IEEE Military Communications Conference*, pages 1116–1120, Nov 2013.

[76] G. Baldini, S. Karanasios, D. Allen, and F. Vergari. Survey of wireless communication technologies for public safety. *IEEE Communications Surveys Tutorials*, 16(2):619–641, Second 2014.

[77] Y. Zeng, R. Zhang, and T. J. Lim. Wireless communications with unmanned aerial vehicles: opportunities and challenges. *IEEE Communications Magazine*, 54(5):36–42, May 2016.

[78] A. Merwaday and I. Guvenc. Uav assisted heterogeneous networks for public safety communications. In *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 329–334, March 2015.

[79] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah. Mobile unmanned aerial vehicles (uavs) for energy-efficient internet of things communications. *IEEE Transactions on Wireless Communications*, 16(11):7574–7589, Nov 2017.

[80] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah. Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs. *IEEE Transactions on Wireless Communications*, 15(6):3949–3963, June 2016.

[81] S. Lien, K. Chen, and Y. Lin. Toward ubiquitous massive accesses in 3gpp machine-to-machine communications. *IEEE Communications Magazine*, 49(4):66–74, April 2011.

[82] Y. Pang, Y. Zhang, Y. Gu, M. Pan, Z. Han, and P. Li. Efficient data collection for wireless rechargeable sensor clusters in harsh terrains using uavs. In *2014 IEEE Global Communications Conference*, pages 234–239, Dec 2014.

[83] Yan Li, Xiaodong Ji, Mugen Peng, Yuan Li, and Chao Huang. An enhanced beamforming algorithm for three dimensional mimo in lte-advanced networks. In *2013 International Conference on Wireless Communications and Signal Processing*, pages 1–5, Oct 2013.

[84] W. Zafar and B. Muhammad Khan. Flying ad-hoc networks: Technological and social implications. *IEEE Technology and Society Magazine*, 35(2):67–74, June 2016.

[85] Ozgur Koray Sahingoz. Networking models in flying ad-hoc networks (fanets): Concepts and challenges. *Journal of Intelligent & Robotic Systems*, 74:513–527, 2014.

[86] Ilker Bekmezci, Ozgur Koray Sahingoz, and Samil Temel. Flying ad-hoc networks (fanets): A survey. *Ad Hoc Networks*, 11:1254–1270, 2013.

[87] I. Uluturk, I. Uysal, and K. Chen. Efficient 3D placement of access points in an aerial wireless network. In *2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 1–7, Jan 2019.

[88] Alejandro Aragón-Zavala, José Luis Cuevas-Ruíz, and José Antonio Delgado-Penín. *High-Altitude Platforms for Wireless Communications*. John Wiley & Sons, 2008.

[89] Stylianos Karapantazis and Fotini Niovi Pavlidou. Broadband communications via high-altitude platforms: A survey, 2005.

[90] Yong Zeng, Rui Zhang, and Teng Joon Lim. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Communications Magazine*, 54(5):36–42, 2016.

[91] Sathyanarayanan Chandrasekharan, Karina Gomez, Akram Al-Hourani, Sithamparanathan Kandeepan, Tinku Rasheed, Leonardo Goratti, Laurent Reynaud, David Grace, Isabelle Bucaille, Thomas Wirth, and Sandy Allsopp. Designing and implementing future aerial communication networks. *IEEE Communications Magazine*, 54(5):26–34, 2016.

[92] F. Al-Turjman, H. S. Hassanein, and M. a. Ibnkahla. Connectivity Optimization for Wireless Sensor Networks Applied to Forest Monitoring. *2009 IEEE International Conference on Communications*, pages 1–6, 2009.

[93] Dario Pompili, Tommaso Melodia, and Ian F. Akyildiz. Three-dimensional and two-dimensional deployment analysis for underwater acoustic sensor networks. *Ad Hoc Networks*, 7(4):778–790, 2009.

[94] Guangjie Han, Chenyu Zhang, Lei Shu, Ning Sun, and Qingwu Li. A survey on deployment algorithms in underwater acoustic sensor networks, 2013.

[95] Qingqing Wu, Yong Zeng, and Rui Zhang. Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks. *IEEE Transactions on Wireless Communications*, 17(3):2109–2121, mar 2018.

[96] S. A. W. Shah, T. Khattab, M. Z. Shakir, and M. O. Hasna. Association of networked flying platforms with small cells for network centric 5g c-ran. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–7, Oct 2017.

[97] A. Merwaday and I. Guvenc. UAV assisted heterogeneous networks for public safety communications. In *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 329–334, March 2015.

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

[99] Irem Bor-Yaliniz, Sebastian S Szyszkowicz, and Halim Yanikomeroglu. Environment-aware drone-base-station placements in modern metropolitans. *IEEE Wireless Communications Letters*, 2017.

[100] P. Gupta and P. R. Kumar. Critical power for asymptotic connectivity. In *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, volume 1, pages 1106–1110 vol.1, 1998.

[101] C. Bettstetter. On the Connectivity of Ad Hoc Networks. *The Computer Journal*, 47(4):432–447, apr 2004.

[102] Mathew Penrose. *Random geometric graphs*. Number 5. Oxford university press, 2003.

[103] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, Nov 1995.

[104] Lester James V. Miranda. PySwarms, a research-toolkit for Particle Swarm Optimization in Python. *Journal of Open Source Software*, 3, 2018.

[105] B. Li, Z. Fei, and Y. Zhang. UAV communications for 5G and beyond: Recent advances and future trends. *IEEE Internet of Things Journal*, 6(2):2241–2263, April 2019.

[106] B. Galkin, J. Kibilda, and L. A. DaSilva. UAVs as mobile infrastructure: Addressing battery lifetime. *IEEE Communications Magazine*, 57(6):132–137, June 2019.

[107] D. Deng, Y. Lin, X. Yang, J. Zhu, Y. Li, J. Luo, and K. Chen. IEEE 802.11ax: Highly efficient WLANs for intelligent information infrastructure. *IEEE Communications Magazine*, 55(12):52–59, Dec 2017.

[108] K. Chen and H. Hung. Wireless robotic communication for collaborative multi-agent systems. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2019.

[109] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An Introduction (2nd edition)*. MIT Press, 2018.

[110] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[111] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016.

[112] Steven J. Bradtke and Michael O. Duff. Reinforcement learning methods for continuous-time markov decision problems. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, NIPS'94, page 393–400, Cambridge, MA, USA, 1994. MIT Press.

[113] Sridhar Mahadevan, Nicholas Marchalleck, Tapas K. Das, and A. Gosavi. Self-improving factory simulation using continuous-time average-reward reinforcement learning. In *Proceedings of the 14th International Conference on Machine Learning*, pages 202–210. Morgan Kaufmann, 1997.

[114] Ronald Edward Parr and Stuart Russell. *Hierarchical Control and Learning for Markov Decision Processes*. PhD thesis, 1998. AAI9902197.

[115] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[116] Doina Precup. Temporal abstraction in reinforcement learning, 2000.

[117] Richard S Sutton, Satinder P Singh, Doina Precup, and Balaraman Ravindran. Improved switching among temporally abstract actions. In *Advances in neural information processing systems*, pages 1066–1072, 1999.

Appendix A: Copyright Permissions

A.1 Chapter 2

Copyright permission notice for the Chapter 2 is provided below.

A.2 Chapter 3

Copyright permission notice for the Chapter 3 is provided below.