2022

Link Scheduling in UAV-Aided Networks

Yawen Zheng

Link Scheduling in UAV-Aided Networks

A thesis submitted in partial fulfilment of the requirements for the award of the degree

Doctor of Philosophy

from

UNIVERSITY OF WOLLONGONG

by

Yawen Zheng

Bachelor of Engineering (Telecommunications)

School of Electrical, Computer and Telecommunications Engineering

April 2022

Statement of Originality

I, Yawen Zheng, declare that this thesis, submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institutions.

.

Signed

.

Yawen Zheng

October, 2021

Abstract

Unmanned Aerial Vehicles (UAVs) or drones are a type of low altitude aerial mobile vehicles. They can be integrated into existing networks; e.g., cellular, Internet of Things (IoT) and satellite networks. Moreover, they can leverage existing cellular or Wi-Fi infrastructures to communicate with one another. A popular application of UAVs is to deploy them as mobile base stations and/or relays to assist terrestrial wireless communications. Another application is data collection, whereby they act as mobile sinks for wireless sensor networks or sensor devices operating in IoT networks. Advantageously, UAVs are cost-effective and they are able to establish line-of-sight links, which help improve data rate. A key concern, however, is that the uplink communications to a UAV may be limited, where it is only able to receive from one device at a time. Further, ground devices, such as those in IoT networks, may have limited energy, which limit their transmit power. To this end, there are three promising approaches to address these concerns, including (i) trajectory optimization, (ii) link scheduling, and (iii) equipping UAVs with a Successive Interference Cancellation (SIC) radio.

Henceforth, this thesis considers data collection in UAV-aided, TDMA and SIC-equipped wireless networks. Its main aim is to develop novel link schedulers to schedule uplink communications to a SIC-capable UAV. In particular, it considers two types of networks: (i) one-tier UAV communications networks, where a

SIC-enabled rotary-wing UAV collects data from multiple ground devices, and (ii) Space-Air-Ground Integrated Networks (SAGINs), where a SIC-enabled rotary-wing UAV offloads collected data from ground devices to a swarm of CubeSats. A CubeSat then downloads its data to a terrestrial gateway. Compared to one-tier UAV communications networks, SAGINs are able to provide wide coverage and seamless connectivity to ground devices in remote and/or sparsely populated areas.

This thesis first considers an uplink schedule optimization problem. Its objective is to collect the maximum amount of data from ground devices within a fixed time horizon. The constructed link schedule guarantees that each ground device is activated at least once. The problem is first formulated as an Integer Linear Program (ILP). A key challenge, however, is that the number of link sets, where the links in each set satisfy SIC constraints, increases exponentially. Hence, this thesis also proposes two other centralized methods and a distributed method for use in large-scale networks. Specifically, these methods include a Cross-Entropy (CE) based method, a novel heuristic called Greedily Construct Transmission Set (GCTS) and a distributed Medium Access Control (MAC) called Collection Point Selection Protocol (CPSP). Numerical results show that the number of ground devices and data collection points along a trajectory as well as the speed and height of a UAV affect the resulting schedule and the amount of collected data.

This thesis also considers adapting the trajectory of a UAV to attain favourable channel condition to facilitate SIC decoding. Specifically, it proposes and studies an approach that jointly considers trajectory design and uplink scheduling to maximize the total amount of collected data and/or the energy efficiency of a SIC-enabled UAV. In this respect, this thesis contains three solutions; namely, ILP, a novel heuristic called Iteratively Construct Link Schedule and Trajectory (ICLST), and a State-Action-Reward-State-Action (SARSA)-based learning protocol. Numerical results show that SIC allows at most four simultaneous uplink transmissions from ground devices. Additionally, it helps double the amount of collected data at the UAV as compared to a conventional Time Division Multiple Access (TDMA) schedule. Moreover, placing devices at different heights/elevations enables a UAV to collect 15.8% additional data. Further, the novel heuristic ICLST is capable of producing a schedule that is near optimal.

Lastly, this thesis considers a novel problem that jointly optimizes routing and uplink scheduling in SAGINs. Unlike previous works, it considers a SIC-capable UAV that collects data from ground devices in an IoT network and also uploads data to a swarm of CubeSats. The problem's objective is to maximize the minimum flow among all ground devices to a terrestrial gateway over a fixed time horizon. A Mixed Integer Linear Program (MILP) solution is first proposed to select in each time slot that determines (i) the routing from a SIC-enabled UAV across CubeSats to the gateway, (ii) the optimal link schedule to schedule uplink transmissions from ground devices, and (iii) the flow rate over each active directed link. This thesis also proposes a novel protocol, called Iterative Flow and Path Reservation (IFPR), in which the UAV iteratively selects multiple paths with the least cost within a planning time horizon. Additionally, the UAV considers two methods to schedule ground devices and saturate the capacity of a selected path. The first method is a simplified MILP (SMILP) that schedules ground devices with the maximum sum-rate. The second method is a greedy algorithm called Less Data Schedule First (LDSF), which prioritizes ground devices that have uploaded the least amount of data to the gateway. Numerical results show that satellite links help the UAV collect 61% more data from ground devices. Moreover, as compared to the MILP or the optimal amount of data, IFPR collects 23% less data. Further, for both solutions, their Jain's fairness index reaches around one when the number of time slots is sufficiently large. Lastly, when IFPR uses SMILP to schedule ground devices, the gateway collects a higher amount of data but at the expense of fairness.

Acknowledgments

First and foremost, I would like to thank my supervisor, Associate Professor Kwan-Wu Chin, for his selfless and patient guidance as well as constant encouragement and inspiration. He spent a lot of his valuable time on teaching me how to find and solve a research problem as well as how to improve academic research and writing skills. Without his innovative ideas, hard work as well as strict and high requirements, it would be impossible for me to finish my PhD degree and grow into an independent researcher. On the other hand, his kindness and patience helped me overcome difficulties and frustrations experienced throughout my PhD studies. His sense of humor and wide interests always made our meetings and chats enjoyable. His constructive advice are valuable for my future career and life.

I would like to thank my co-supervisor, Associate Professor Raad Raad, for his contributions and helpful comments. I also want to thank Associate Professor Ashish Agalgaonkar for his constructive suggestions about research and life.

I would also like to thank my friends and office mates in Building 39A.201, Miss Zanru Yang, Dr Yishun Wang, Dr Luyao Wang, Dr Tianle Liu, Dr Ying Liu, Dr Yizhou Luo, Miss Yiwei Li, Miss Hang Yu, Mr Lei Zhang and Mr Honglin Ren, for the helpful discussions, joyful conversations and a pleasant working environment.

Special thanks to my landlady, Ms. Anne Bestman, for her kindness, companionship and understanding. Her warm care, delicious meals and desserts made me feel at home. Her Ukulele performances and mosaic designs brought me a lot of fun. Thanks to her cute pet dog, Bella, for staying next to me when I was studying and waiting at the door when I came home.

Lastly, I would like to thank my parents for their love and encouragement throughout my life. They are my powerful backing and best friends. Without their understanding and support, I could not achieve this stage.

Contents

List of Figures

List of Tables

Abbreviations

5G	Fifth Generation
B5G	Beyond 5G
HAP	High Altitude Platform
LAP	Low Altitude Platform
UAVs	Unmanned Aerial Vehicles
QoS	Quality of Service
3D	Three-Dimensional
IoTs	Internet of Things
WPCN	Wireless Powered Communication Network
CRN	Cognitive Radio Network
WSN	Wireless Sensor Network
WLAN	Wireless Local Area Network
GEO	Geostationary Earth Orbit
MEO	Medium Earth Orbit
LEO	Low Earth Orbit
NOMA	Non-Orthogonal Multiple Access
SIC	Successive Interference Cancellation
SINR	Signal-to-Interference-Plus-Noise Ratio
SNR	Signal-to-Noise Ratio

MAC	Medium Access Control
MPR	Multi-Packet Reception
MUD	Multi-User Detection
IC	Interference Cancellation
PIC	Parallel Interference Cancellation
SLIC	Single Link Interference Cancellation
TDMA	Time Division Multiple Access
FDMA	Frequency Division Multiple Access
CDMA	Code Division Multiple Access
OFDMA	Orthogonal Frequency Division Multiple Access
LP	Linear Program
IP	Integer Program
ILP	Integer Linear Program
MILP	Mixed Integer Linear Program
MINLP	Mixed Integer Non-Linear Program
MIMO	Multiple-Input-Multiple-Output
MDP	Markov Decision Process
OMA	Orthogonal Multiple Access

Chapter 1

Introduction

1.1 Background

To date, researchers have employed a variety of mobile vehicles to augment existing communication infrastructures [3]. Figure 1.1 shows two categories of mobile vehicles: aerial and terrestrial. An example terrestrial mobile vehicle is Unmanned Ground Vehicles (UGVs) [4]. Aerial vehicles can be divided into (i) High Altitude Platforms (HAPs), such as satellites/CubeSats, balloons and aircraft, and (ii) Low Altitude Platforms (LAPs), such as Unmanned Aerial Vehicles (UAVs) [5].

Table 1.1 shows the features and examples of both HAPs and LAPs. In particular, HAPs (i) are quasi-stationary, (ii) have high energy storage, and (iii) they are able to carry a heavier payload. Hence, they are preferred for use in large geographic areas and/or long-term missions [6]. In comparison, LAPs are more flexible and easier to acquire, deploy, and maintain [7]. Hence, they are more suitable for time-sensitive applications. LAPs are identified as an important component of 5G and beyond 5G (B5G) wireless technologies [5].

As shown in Figure 1.1, LAPs, especially UAVs, can be further categorized according to their wings: fixed or rotary. Fixed wing UAVs have a higher speed and altitude but are much heavier. They need to maintain continuous forward motion to

Object	Height Above Earth	Lifetime	Features	Examples
HAP	Above 17 km	Long endurance (days or months)	Quasi-stationary, larger, heavier and with higher energy storage	European Startobus [8], Airbus Zephyr [9], Google Project Loon [10]
LAP	600 m to 5.5 km	Several hours	Cost-effective, fast and flexible deployment	Parrot drones [11], Da-Jiang Innovations (DJI) drones [12]

Table 1.1: Features of HAPs and LAPs.

remain aloft, which are similar to small aircrafts [5]. By contrast, rotary-wing UAVs, such as quadcopters, can take off vertically and hover over a specific geographical area while remaining stationary if needed [13]. Consequently, rotary-wing UAVs are more popular among hobbyists, and they are commonly used for applications requiring short flight time such as search and monitor operations [14].

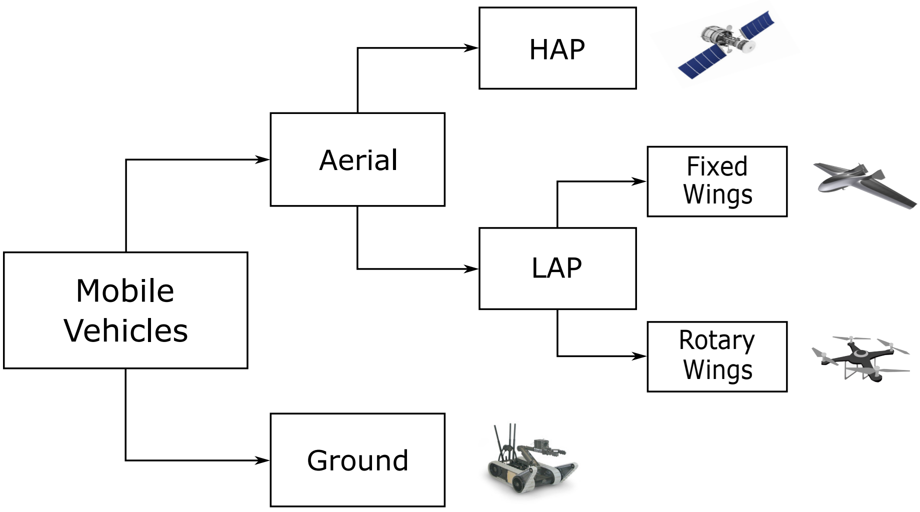

Figure 1.1: Classification of mobile vehicles.

UAVs have found applications in wide ranging areas [15]. Generally, these applications are divided into two categories: civilian or military [3]. Popular civilian applications include (i) disaster relief operations; e.g., reference [16] uses a low-altitude tethered balloon to support emergency medical communication services in natural disaster areas, (ii) providing Internet connectivity to rural areas; examples include Google Project Loon [10] and Facebook Aquila solar-powered airplanes [17], and (iii) public transportation and package delivery; for example, Amazon Prime Air [18] and Alphabet's drone company Wing [19]. In particular, Amazon Prime Air is designed to use small drones to safely deliver packages to customers within 30 minutes. Additionally, reference [19] reports that UAVs from Wing delivered 10,000 cups of coffee, 1,700 snack packs and 1,200 roast chickens to customers in Logan, Australia in 2020. Compared to civilian services, UAVs have been used in the military during the past decades [20]. They are mainly deployed in hostile territories to (i) track targets, (ii) provide area surveillance and patrolling, and (iii) support connectivity of tactical edge devices and networks, so as to reduce pilot losses [21]. For example, in 2015, a patent from Boeing [22] outlined a UAV that can carry out underwater missions. Apart from these applications, UAVs have also been proposed as mobile servers or cloudlets that provide application offloading opportunities to mobile users [23]. Moreover, UAVs are used to provide localization info and to aid navigation [24]. The aforementioned examples indicate that the applications of UAVs in both civilian and military are likely to grow significantly in the near future. In fact, the global UAV market size is estimated to reach USD $72320 Million by 2028 [25].

Recently, many researchers have studied employing UAVs in wireless communications. For example, the 3^{rd} Generation Partnership Project (3GPP) has a research study to understand existing obstacles, challenges, requirements, and possibilities when applying UAVs in LTE and 5G/B5G communication networks [26]. In addition, Qualcomm and AT&T plan to deploy UAVs to enable wide-scale wireless communications in 5G/B5G [27]. Compared to stationary terrestrial infrastructures, UAV communications have the following benefits [28]:

- **Dynamic deployment ability.** Compared to building traditional fixed communication infrastructures, deploying UAVs is cost-effective. Employing UAVs saves the cost of building communication towers and laying cables as well as site rentals to house communication equipment. Moreover, UAVs can be deployed dynamically and allocated to different users or controllers in an on-demand manner to handle various traffic requirements. Further, UAVs can be used to optimize delays, throughput, fair sharing of spectrum, and/or energy consumption of nodes. Hence, UAVs are ideal for increasing the robustness or performance Quality of Service (QoS) of a communication system against environment changes.

- **Line-of-sight links.** Mobile UAVs provide a higher probability to connect ground users via line-of-sight links to facilitate higher reliable transmissions over long distances. Compared to terrestrial fading channels, line-of-sight links have less channel variation in time and frequency. Hence, communication scheduling and resource allocation in UAV communications can be efficiently implemented at a slower pace.

- **UAV-based swarm networks.** A swarm of UAVs are capable of forming scalable and flexible multi-UAV networks that provide ubiquitous connection to ground users. Moreover, a multi-UAV network is ideal for restoring and expanding a communication infrastructure quickly.

In general, UAVs can be integrated into an existing network as aerial nodes and/or aerial communication platforms [5]. On one hand, they can leverage existing cellular or Wi-Fi infrastructures from the sky to communicate with one another or with ground nodes/devices [29][30]. This integrated case is commonly referred to as cellular-connected UAVs. On the other hand, UAVs are able to function as flying bases stations and/or mobile relays to assist terrestrial wireless communications by providing data access from the sky [6, 31–33]. Hence, this case is called UAV-assisted wireless communications. In particular, UAVs are employed as aerial base stations or access points to provide communication services to ground targets in high traffic demand and overloaded areas [6]. Additionally, they are deployed as aerial mobile relays by mounting communication transceivers. They are able to extend the communication range of existing wireless infrastructures; so as to provide reliable wireless connectivity between distant users or user groups [34]. For example, UAV-assisted communication is a promising technology to support information dissemination and data collection in Internet-of-Things (IoTs) and Wireless Sensor Networks (WSNs). In particular, WSNs have been used in many periodic sensing applications in recent years [35]. The traditional architecture of WSNs consists of multiple static battery-powered sensor nodes. A key aim is to reduce energy consumption and thus prolongs the lifetime of WSNs. Specifically, nodes close to a sink spend higher amount of energy than nodes that are far away. As for IoTs, the aim is to connect so called 'things' anytime, anywhere [5]. UAVs are particularly suited to address the challenges of IoT devices, which include small transmit power and short transmission distance [36]. By taking advantage of mobility, UAVs can periodically fly over sensor devices to collect/deliver data. This will help reduce the energy consumption of sensor nodes or 'things'. In addition, UAVs can help balance the workload among sensors [37].

The basic communication requirements for UAVs can be classified into two types [38]: (i) control and non-payload communication for command and control that requires high reliability and low latency, and (ii) payload communication for application such as high-rate video streaming for surveillance, infrastructure inspection as well as search and rescue. There are various wireless technologies that can be used to achieve the aforementioned two communication requirements, so as to provide seamless connectivity as well as high reliability and/or throughput for UAV communications [15]. In particular, candidate communication technologies include direct links, satellites, ad hoc networks, and cellular networks. The details of each technology are introduced as follows:

1. **Direct links.** Direct-link communication between a UAV and its associated ground nodes over the 2.4 GHz band is the most commonly used technology because of its simplicity and low cost. However, direct-link communication is not suitable for large-scale UAV deployments because of the following drawbacks: (i) limited operation range, (ii) easily blocked by obstacles, such as

trees and high-rise buildings, and (iii) insecure and vulnerable to interference and jamming.

2. **Satellites.** Due to the global coverage of satellites, they are employed to enable UAV communications. In particular, satellites can help relay data transmitted between UAVs and ground nodes that are widely separated or located in a remote area with no Wi-Fi or cellular coverage. However, satellite-enabled UAV communications also have various disadvantages. First, satellite communications have high operational cost. Second, long transmission distances between satellites and UAVs/ground nodes cause significant delay and propagation loss.

3. **Ad hoc networks.** A Mobile Ad Hoc Network (MANET) is an infrastructure-free and dynamically self-organizing network that enables peer-to-peer communications between mobile devices, such as laptops and cellphones. In particular, each device in a MANET can move randomly over time and communicate over bandwidth-constrained wireless links using IEEE 802.11 a/b/g/n. A Flying Ad Hoc Network (FANET), a type of MANET, supports communications between high mobility ground vehicles and UAVs in three-dimensional (3D) networks [39]. However, realizing a reliable routing protocol in a network with dynamic and intermittent connections between mobile UAVs is complex and difficult. Hence, FANET can only be used to support UAV communications in a small network [39].

4. **Cellular networks.** Existing and future-generation cellular networks can cost-effectively enable large-scale UAV communications [29]. This is because cellular networks have a (i) high-speed optical backhaul, and (ii) ubiquitous coverage. For example, a 5G cellular network is expected to support a peak data rate of 10 Gbps with only 1 ms round-trip latency [15]. In principle, these characteristics are adequate for high-rate and delay-sensitive UAV communications.

With the increasing development and utilization of Internet of Things (IoTs), smart devices are now deployed in remote areas, such as oceans, desserts and forests [40]. These smart devices require a network architecture that is capable of providing ubiquitous communication coverage, high data rates and low network latency services [41]. To this end, researchers have started to consider satellite-terrestrial networks [42]; see Figure 1.2. Advantageously, satellites, especially CubeSats[1], can form large constellations to provide global coverage and seamless connectivity to the Internet of Remote Things (IoRTs) [43]. For example, the Starlink project from SpaceX plans to establish a constellation with 40,000 Low Earth Orbit (LEO) satellites to provide high-speed and low-latency broadband Internet across the globe [44]. Compared to conventional satellites, CubeSats are small, cost-effective, and highly capable [45]. In addition to satellites, researchers are also considering aerial networks that include HAPs, such as balloons and aircraft, and LAPs, such as UAVs [14]. Compared to HAPs, LAPs/UAVs are more flexible and easier to acquire, deploy and maintain [46]. As UAVs are mobile, they are able to establish line-of-sight links to devices, and thus facilitate reliable transmissions [47]. Further, they can be used to overcome the large propagation delay between ground devices and satellites [48], and help reduce the power consumption of devices [41].

Figure 1.3 shows a Space-Air-Ground Integrated Network (SAGIN) with example nodes in satellite systems, aerial networks, and terrestrial communications [1]. Specifically, a space network is composed of satellites and constellations as well as their corresponding terrestrial infrastructures, e.g., ground stations and/or gateways. Satellites are classified into three categories including Geostationary Earth Orbit (GEO), Medium Earth Orbit (MEO) and Low Earth Orbit (LEO) satellites [49]. The space network has multiple inter-satellite links and inter-layered links between GEO, MEO and LEO satellites. An aerial network uses both HAPs and LAPs as carriers to acquire, transmit and process data. A ground network consists of existing terrestrial communication systems such as cellular networks, MANETs,

[1] See http://www.nanosats.eu/ for CubeSats launched to date.

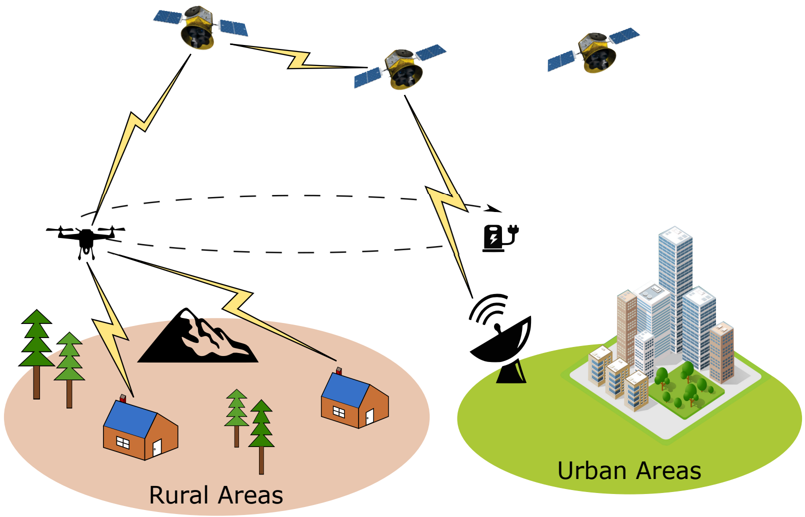

Figure 1.2: An example SAGIN. A mobile UAV provides connectivity to rural areas. The three CubeSats help the UAV relay collected data back to a gateway.

WSNs, and wireless local area networks (WLANs).

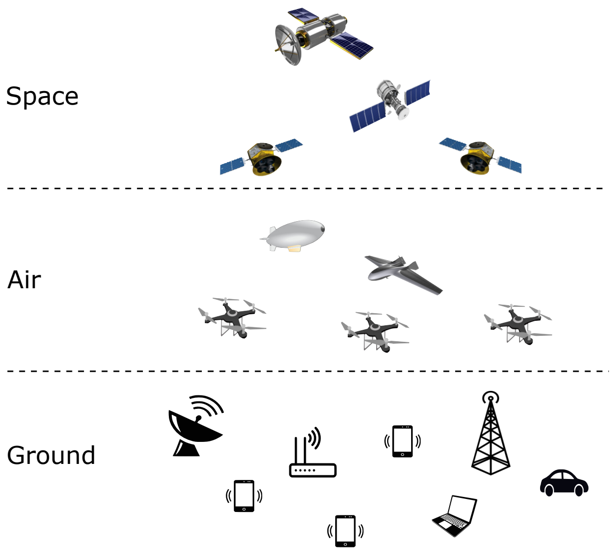

Figure 1.3: Example nodes in a three-layered SAGIN.

Table 1.2 summarizes and compares three networks of SAGINs in terms of their height, delay, data rate, advantages and limitations [1][2]. The benefits of SAGINs include (i) densely deployed terrestrial networks in urban areas that can support high data rate access, (ii) UAV communications are cost-effective and able to rapidly enhance terrestrial networks/services as well as offload traffic in crowded areas, and (iii) satellite networks that provide wide coverage and seamless connectivity to remote and/or sparsely populated areas.

Layer	Objects	Height Above Earth	One Way Delay	Data Rate	Advantages	Limitations
Space	GEO	35,786 km	About 270 ms	-	Large coverage, broadcast or multi-cast	Long propagation latency and high cost
	MEO	2,000 -35,786 km	About 110 ms	Up to 1.2 Gbps		
	LEO	160 - 2,000 km	Less than 40 ms	Up to 3.75 Gbps		
Air	Aircraft, balloon, UAV	17 - 30 km (HAP), 600 m - 5.5 km (LAP)	Medium	High data rates	Wide coverage, low cost, flexible deployment	Small capacity and unstable link
Ground	Terrestrial infrastructure/network	N/A	Lowest	High data rates	Rich resources and high throughput	Limited coverage and vulnerable to disaster

Table 1.2: Comparison of different networks in SAGINs [1][2].

1.2 Problem Space and Motivation

Uplink communication is relied upon by data collection applications. The main concerns of these applications include the throughput of ground nodes, lifetime of UAVs as well as the energy consumption of both UAVs and ground nodes. Moreover, due to their limited on-board energy, rotary-wing UAVs must collect data within a budgeted flying time [14]. Hence, a UAV needs to fly along a properly-designed trajectory that allows it to collect as much data as possible from ground nodes [50].

Another concern is interference, which limits the throughput of uplinks communications [51]. This is because concurrent transmissions from ground nodes over the same frequency or channel are likely to cause a low Signal-to-Interference-Plus-Noise Ratio (SINR) at nodes/receivers. Hence, decoding errors may occur. Interference can be managed with an appropriate link scheduler [52]. Briefly, link scheduling is a Medium Access Control (MAC) layer strategy to manage the activation of links. For example, a Time Division Multiple Access (TDMA) schedule consists of time slots [53], where interfering links are assigned to different slots. Advantageously, TDMA ensures no energy is wasted due to collisions, and allows ground nodes to only wake-up and transmit at predefined time slot(s). This is especially important in energy constrained IoT networks [36].

Another promising direction to enhance the throughput of a node is Successive Interference Cancellation (SIC) [54]. In particular, SIC is a key part of Non-Orthogonal Medium Access (NOMA) that has been adopted for use in 5G networks [55]. Briefly, SIC allows receivers to separate, decode, and remove signals from a composite signal over multiple stages; see Chapter 2 for details. Consequently, it allows multiple senders to transmit to the same receiver at the same time. Goussevskaia *et al.* [56] note that applying SIC improves the throughput of a single-hop wireless network by 20% where nodes are randomly distributed on a Euclidean area.

Henceforth, this thesis considers data collection in UAV-aided, TDMA and SIC equipped wireless networks. First, it considers a one-tier UAV communications network/system, in which a rotary-wing UAV collects data from multiple ground devices. In particular, ground devices have different heights/elevations. A UAV is equipped with a SIC radio that enables multiple simultaneous uplink transmissions with different data rates. It then flies according to a pre-computed trajectory in order to optimize SIC decoding successes at data collection points. Figure 1.4 shows an example one-tier UAV communication network. UAV u flies from left to right to collect data from four ground devices namely g_1, g_2, g_3 and g_4. An example link set/schedule is shown in Figure 1.4, where ground devices g_1, g_2 and g_4 are scheduled to transmit simultaneously.

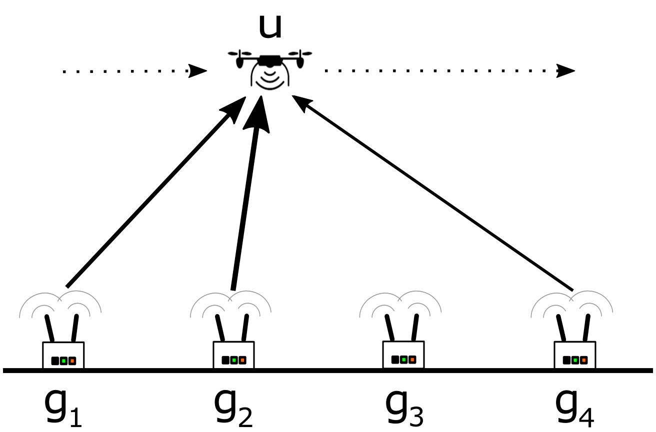

Figure 1.4: An example one-tier UAV communications network. The thickness of arrows indicates uplink data rate.

Second, this thesis considers SAGIN, where a SIC-enabled rotary-wing UAV flies in a circular trajectory with a fixed-altitude to collect data from ground devices. The UAV either stores its collected data and transports it back to a gateway located at its Start/End (S/E) station or offloads it to a swarm of LEO satellites/CubeSats. Each CubeSats swarm is connected via Inter-Satellite Links (ISLs) that allow data transmissions between CubeSats. Moreover, these CubeSats have a time-varying topology. Figure 1.5 shows an example SAGIN that consists of (i) two LEO CubeSats s_1 and s_2, (ii) a mobile rotary-wing UAV u, (iii) three ground devices g_1, g_2, and g_3, and (iv) a gateway GW. As shown in Figure 1.5, we see the following types of directed links: (i) uplinks from ground devices to the UAV; e.g., (g_1, u), (g_2, u)

and (g_3, u), (ii) uplinks from the UAV to CubeSats; e.g., (u, s_1), and (iii) ISLs; e.g., (s_1, s_2), (iv) a downlink from the UAV to the gateway, e.g. (u, GW), and (v) downlinks from CubeSats to the gateway; e.g., (s_2, GW).

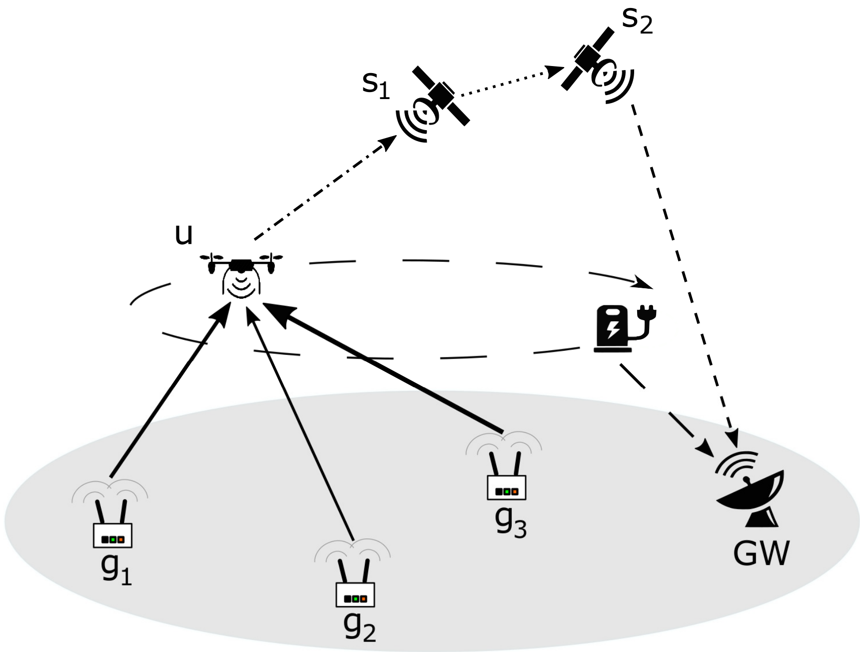

Figure 1.5: An example SAGIN. Different patterns indicate directed links between components with different altitudes.

Given the above networks, this thesis considers the following research questions: (i) how to construct the optimal uplink schedule to a UAV? (ii) how to jointly optimize UAV trajectory and uplink schedule in UAV-aided networks? and (iii) how to jointly optimize routing and uplink schedule in SIC-enabled SAGINs? The next sections explain these questions in detail.

1.2.1 Uplink Schedule Optimization

The objective of the first research problem is to compute the optimal uplink schedule that allows a single UAV to collect the maximum amount of data from ground nodes/devices within a fixed time horizon. The optimal constructed link schedule also needs to guarantee that ground devices are activated at least once. Figure 1.6 illustrates the link scheduling problem at hand. As shown in Figure 1.6, there are two data collection points A and B as well as four ground devices: g_1, g_2, g_3, g_4. The

Point A	Point B
g_1	g_2

Table 1.3: A possible TDMA link schedule for the example UAV network in Figure 1.6.

Point A	Point B
$\{g_1, g_3\}$	$\{g_1, g_2, g_4\}$

Table 1.4: A possible SIC-enabled TDMA link schedule for the example UAV network in Figure 1.6.

UAV flies from left to right. Without a SIC radio, the UAV is only able to receive from one ground device at each data collection point; see Table 1.3 for a possible TDMA schedule.

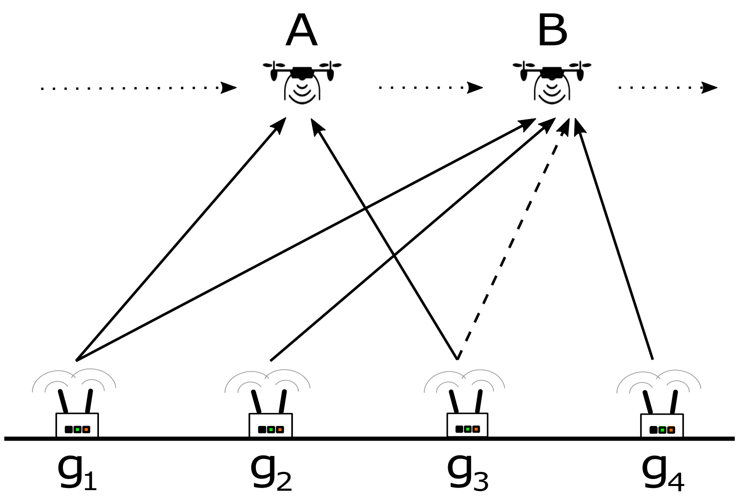

Figure 1.6: An example UAV network. Different patterns indicate uplinks that belong to different link schedules.

Now consider a SIC-aided UAV. As shown in Figure 1.6, data collection point A has one link set/schedule in which g_1 and g_3 transmit simultaneously. As for data collection point B, there are two schedules/sets: $\{g_1, g_2, g_4\}$ or $\{g_3\}$. The problem at hand is to choose one link set for each collection point. Table 1.4 shows a possible SIC-enabled TDMA link schedule, in which link sets $\{g_1, g_3\}$ and $\{g_1, g_2, g_4\}$ are selected at collection points A and B, respectively. Compared to the schedule in Table 1.3, the schedule shown in Table 1.4 yields a higher amount of collected data.

In the previous example, there are a number of issues to consider. First, the channel gain of ground devices varies over time. Moreover, there are multiple transmission/link sets. The UAV needs to select a transmission set for each data collection point that yields the highest total amount of collected data. A challenging aspect is that the number of transmission sets increases exponentially with the number of ground devices. Specifically, if there are N ground devices, then there are $2^N - 1$ possible transmission sets at each data collection point. For example, in Figure 1.6, data collection point A and B have up to 15 possible transmission sets, respectively.

1.2.2 Joint Trajectory and Uplink Schedule Optimization

The second research problem not only considers uplink scheduling, but also considers the problem of optimizing a UAV's trajectory. The objective is to maximize the total amount of data collected by a rotary-wing mobile UAV. Figure 1.7 presents an example UAV network that illustrates the joint trajectory and link scheduling problem of interest. As shown in Figure 1.7, there are four devices g_1, g_2, g_3, g_4 as well as four possible data collection locations/points A, B, C, D. The UAV flies from left to right and also changes its height over time. Device g_2 has a higher height than devices g_1, g_3, and g_4.

Each collection point has multiple possible link sets with individual sum-rate; see Table 1.5. For example, point A has two possible link sets including $\{g_1, g_4\}$ and $\{g_2, g_3\}$. The sum-rate of both link sets is 5 Mbits and 3 Mbits, respectively. At collection point B, devices g_2 and g_3 transmit simultaneously and yield a sum-rate of 4 Mbits. For point C, the sum-rate of link set $\{g_1, g_3\}$ is 4 Mbits. As for point D, there are two possible link sets, namely $\{g_1, g_2, g_3\}$ and $\{g_1, g_4\}$. The sum-rate of both link sets is 6 Mbits and 5 Mbits, respectively. The problem as hand is to select point A or B as the first collection point, and either point C or D to collect data from devices. At each selected collection point, the UAV needs to select a link set to activate simultaneous uplinks or devices. Table 1.6 shows a possible SIC-enabled TDMA link schedule, in which the UAV selects point A and D, that allows it to

Point A	**Point B**	**Point C**	**Point D**
$\{g_1, g_4\}$: 5 Mbits	$\{g_2, g_3\}$: 4 Mbits	$\{g_1, g_3\}$: 4 Mbits	$\{g_1, g_2, g_3\}$: 6 Mbits
$\{g_2, g_3\}$: 3 Mbits	-	-	$\{g_1, g_4\}$: 5 Mbits

Table 1.5: Possible SIC-enabled link sets and individual sum-rate for each data collection point in the example UAV network shown in Figure 1.7.

Point A	**Point D**
$\{g_1, g_4\}$: 5 Mbits	$\{g_1, g_2, g_3\}$: 6 Mbits

Table 1.6: A possible SIC-enabled TDMA link schedule for the example UAV network in Figure 1.7.

collect 11 Mbits of data.

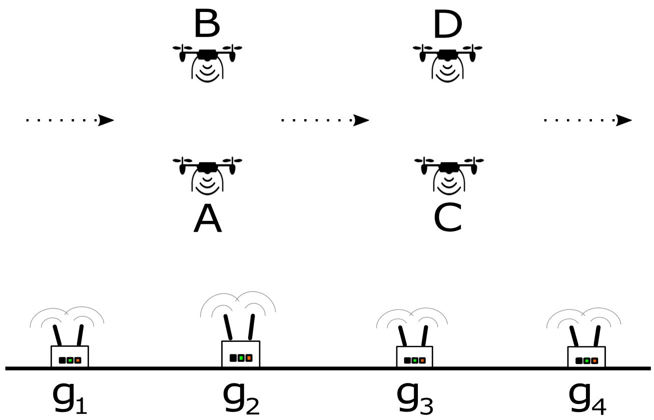

Figure 1.7: An example UAV network with four possible data collection points and four devices.

1.2.3 Joint Routing and Uplink Schedule Optimization in SAGINs

Lastly, this thesis considers a SIC-enabled multi-hop SAGIN. The objective is to maximize the minimum amount of flow that arrives at a gateway over a planning horizon T. Specifically, in each time slot, the problem at hand is to decide (i) uplinks between multiple ground devices and the UAV, an active UAV-satellite link, inter-satellite link(s), and a satellite-gateway link, and (ii) the amount of data to be forwarded over each active link. There are a number of challenges/issues. First, the UAV has limited on-board energy. Thus, at each data collection point, the UAV must select a CubeSat that allows it to upload/offload the maximum amount of data. Second, CubeSats have time varying topologies as well as a short and varying contact duration and channel condition with a gateway.

Figure 1.8 shows a SAGIN and routing over different time slots. Observe that each time slot has a specific network topology. Referring to Figure 1.8, in time slot t_1, ground device g_1 and g_3 communicate with the UAV u simultaneously. The UAV then offloads its collected data to CubeSat s_1. After that, CubeSat s_2 receives data from s_1 via ISL (s_1, s_2) before downloading its data to gateway GW. In time slot T, both ground device g_2 and g_3 upload data to the UAV. The UAV returns to the gateway at this time slot. Hence, it will download its data directly to the gateway at this point.

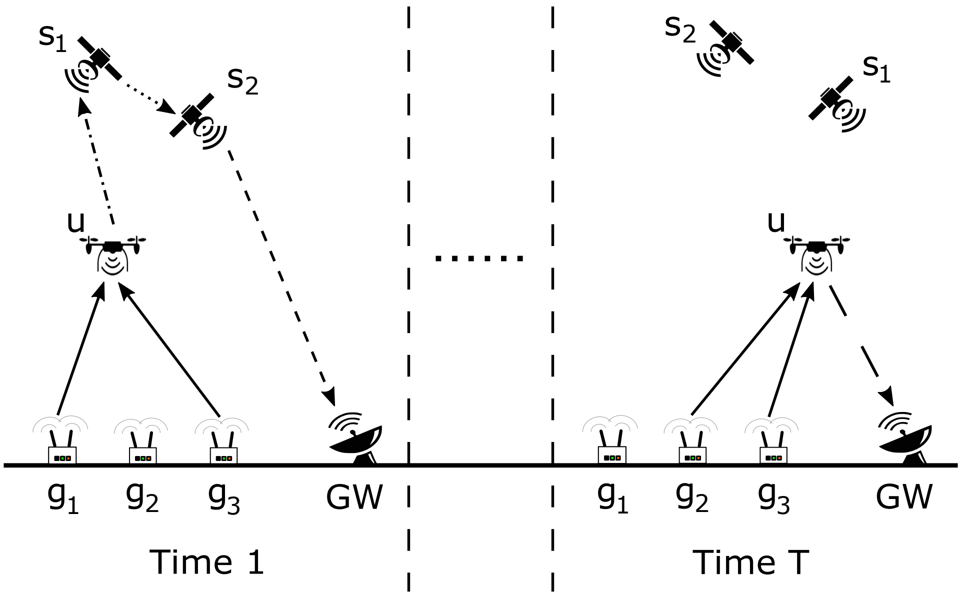

Figure 1.8: An example that shows the problem. Simultaneous active ground devices and CubeSats swarm change over time. Note that CubeSats may not have connectivity to the gateway.

1.3 Contributions

This thesis addresses the aforementioned problems and outlines a number of novel solutions/algorithms. Specifically, it contains the following contributions.

1.3.1 The Optimal Uplink Schedulers for Data Collection in SIC-enabled UAV Networks

The goal is to construct an optimal uplink schedule that maximizes the total collected data at a rotary-wing UAV with an equipped SIC radio. This work first formulates an Integer Linear Program (ILP) model to compute the optimal uplink schedule. Because the number of link sets increases exponentially with the number of ground devices, the considered scheduling problem is thus NP-hard. Hence, a Cross-Entropy (CE) method and a novel heuristic called Greedily Construct Transmission Set (GCTS) are proposed for use in large-scale networks. Further, a distributed MAC called Collection Point Selection Protocol (CPSP) is outlined that allows each ground device to independently learn the best data collection point to transmit data to the UAV. This work then studies how the following factors affect the resulting schedule and total collected data including (i) the number of ground devices and data collection points, (ii) speed and height of a UAV, and (iii) location of ground devices. Numerical results show that SIC allows at most four simultaneous uplinks and helps double the amount of collected data at the UAV. Moreover, both CE method and GCTS are capable of producing a schedule that is near optimal.

1.3.2 Joint Trajectory and Link Scheduling Optimization in SIC-enabled UAV Networks

The goal is to design an approach that jointly considers trajectory design and uplink scheduling. The objective of the considered problem is to maximize the total collected data and/or the energy efficiency of a rotary-wing UAV. This work proposes three solutions including an ILP model, a novel heuristic algorithm named Iteratively Construct Link Schedule and Trajectory (ICLST), and a State-Action-Reward-State-Action (SARSA)-based learning protocol. In particular, an ILP solution provides the optimal trajectory and uplink schedule. The novel heuristic ICLST can be used for large-scale networks. As for the proposed learning protocol, the UAV independently learns a trajectory and corresponding uplink schedule without a central server. Numerical results show that compared to devices with zero elevation, placing devices at different heights/elevations helps collect 15.8% more data. Moreover, flying UAV along a trajectory with different height helps collect 10% additional data. Further, the novel heuristic ICLST is able to collect the same amount of data as the optimal solution. Lastly, the proposed learning approach yields a schedule with the highest energy-efficiency.

1.3.3 Data Collection in SIC-enabled SAGINs

The goal is to design an approach that jointly considers routing and uplink scheduling to maximize the minimum flow among all ground devices. This work first formulates a Mixed Integer Linear Program (MILP) model to compute (i) the optimal path from a UAV to a gateway, (ii) the optimal uplink schedule from ground devices to a UAV, and (iii) the data forwarded on active links. As the search space in each time slot increases exponentially with the number of CubeSats and ground devices, the problem becomes computationally intractable for large-scale networks. Hence, this thesis proposes a novel protocol called Iterative Flow and Path Reservation (IFPR) for use in large-scale networks. Briefly, IFPR allows a UAV to independently select a path and an uplink schedule for each time slot as well as determining the flow for each ground device. IFPR considers two methods to schedule ground devices, namely Simplified MILP (SMILP) and Less Data Schedule First (LDSF). Additionally, IFPR uses the Dijkstra algorithm to select a least cost path and/or randomly selects a path. Numerical results show that CubeSats help collect 61% more data from ground devices. Moreover, compared to the formulated MILP, IFPR only collects 23% less data than the optimal value. Further, for both solutions, their Jain's fairness index reaches around one when the number of time slots is sufficiently large.

1.4 Publications

1. **Y.W Zheng**, K-W Chin and L.Y Wang. Download Traffic Scheduling for CubeSats Swarms with Inter-Satellite Links, *IEEE International Telecommunications and Applications Conference (ITNAC)*, Auckland, NZ, November, 2019.

2. **Y.W Zheng**, and K-W Chin. Link Scheduling for Data Collection in SIC-Capable UAV Networks, *IEEE International Telecommunications and Applications Conference (ITNAC)*, Auckland, NZ, November, 2019.

3. **Y.W Zheng**, and K-W Chin. Joint Trajectory and Link Scheduling Optimization in UAV Networks, *IEEE Access*, vol. 9, pp. 84756-84772, 2021

4. **Y.W Zheng**, and K-W Chin. On Data Collection in SIC-Capable Space-Air-Ground Integrated IoT Neteworks, *IEEE System Journal*, 2022. *To appear.*

5. **Y.W Zheng**, and K-W Chin. Uplinks Schedulers for Data Collection in Backhaul SIC-Capable Multi UAV Networks, *Under Review.*

1.5 Thesis Structure

1. *Chapter 2.* This chapter surveys works that consider SIC, low-altitude UAV communications with and without NOMA and SAGINs. In particular, the implementation details, link scheduling and cross-layer optimizations are summarized for works that consider SIC. The works on UAV communications with and without NOMA both contain trajectory and link scheduling design as well as combinatorial optimizations with multiple parameters. Moreover, the works that consider SAGINs include routing and scheduling problems.

2. *Chapter 3.* This chapter studies constructing an optimal uplink schedule that maximizes the sum-rate over multiple predefined data collection points. It

proposes an ILP solution, two heuristic algorithms including a CE-based approach and a novel heuristic algorithm called GCTS, and a distributed MAC named CPSP.

3. *Chapter 4.* This chapter considers a problem of jointly optimizing UAV's trajectory and link scheduling when a rotary-wing UAV operates on an area divided into a grid with multiple columns and rows. It presents an ILP solution, a novel heuristic algorithm named ICLST and a SARSA-based learning protocol.

4. *Chapter 5.* This chapter outlines a joint routing and link scheduling problem in a SIC-capable SAGIN. The aim is to maximize the minimum flow among all uplinks between ground devices and the UAV over a given planning time horizon. It presents an MILP model and a novel protocol called IFPR for use by a UAV.

5. *Chapter 6.* This chapter contains conclusions, a summary of key contributions, and future research directions.

Chapter 2

Literature Review

This chapter reviews prior works that study Successive Interference Cancellation (SIC), air-ground communications with and without NOMA and SAGINs. First, Section 2.1 classifies prior SIC works related to the implementation of SIC, link scheduling, and cross-layer optimizations. Then Section 2.2 reviews works that consider air-ground communications. In particular, it focuses on low-altitude UAV communication works that study UAV trajectory, link scheduling and resource allocation optimization. After that, Section 2.3 reviews works that apply NOMA to mobile UAV/nodes/users, followed by works that study scheduling and routing in SAGINs with full integration of three segments/networks, namely space, air and ground.; see Section 2.4. Lastly, Section 2.5 outlines limitations and gaps in past works.

2.1 Successive Interference Cancellation

Interference is the main factor that limits the throughput or network capacity of wireless networks [57]. Instead of avoiding interference, researchers have now designed schemes that exploit interference. For example, multi-packet reception is an effective way to combat interference [58], where a node is able to receive from multiple transmitters simultaneously. To achieve multi-packet reception, past approaches employ multi-user detection or interference cancellation [59]. There are different ways to perform interference cancellation, namely SIC, Parallel Interference Cancellation (PIC) or by employing a hybrid method consisting of both SIC and PIC [58]. If a receiver can only cancel one signal before decoding the wanted signal, we then have Single Link Interference Cancellation (SLIC) or single-stage interference cancellation [60].

SIC allows receivers to separate, decode, and remove signals from a composite signal in multiple stages [61]. Specifically, a receiver first decodes the strongest received signal. It then removes the decoded signal from the composite signal. The receiver repeats the said process until all signals are decoded successfully or the SINR of a transmission is no longer satisfied at some stage. In general, the study of SIC mainly consists of the following aspects: (i) link scheduling, (ii) joint topology control and link scheduling, and (iii) cross-layer optimization in multi-hop networks. Section 2.1.1, 2.1.2 and 2.1.3 summarize relevant respective works. As will be discussed later, the major problems addressed by past works include: (i) maximizing the potential of SIC, (ii) computing the minimum schedule length, (iii) maximizing link capacity or fairness among users, and (iv) maximizing the average and/or the minimum throughput.

Many works have considered the implementation of SIC. For example, Halperin *et al.* [61] built a ZigBee prototype and compare it to single packet ZigBee detectors and/or receivers in unmanaged wireless networks with carrier sensing. In particular, they consider two single-packet detectors/receivers: (i) a conventional single-packet ZigBee detector, and (ii) one that can re-synchronize another packet in a collision. The authors showed via testbed experiments with Zigbee receivers that SIC can effectively improve system throughput and bandwidth utilization.

2.1.1 Link Scheduling

As mentioned in Chapter 1, link scheduling manages the activation of links [52]. Efficient link scheduling together with SIC helps promote better spatial reuse as well as transmission concurrency, resulting in increased throughput. For example, for a given wireless network that adopts TDMA, the problem is to schedule links into time slots to avoid interference and/or to satisfy certain Quality of Service (QoS) requirements. Advantageously, with the help of SIC, multiple links can be scheduled simultaneously in the same time slot. To this end, Section 2.1.1.1 and 2.1.1.2 review works that aim to derive a schedule over multiple time slots or in a single time slot, respectively.

2.1.1.1 TDMA Link Schedule

Many works such as [62–71] have considered developing centralized and/or distributed TDMA link scheduling when SIC is applied to receiver(s). In general, these works have taken the following approaches to derive a link schedule: (i) mathematical optimization, such as MILP [72], (ii) heuristic algorithms, (iii) graph-based method, and (iv) reinforcement learning methods, such as Q-learning algorithm [73].

The work in [62] considers a wireless network with multiple stationary nodes and directed links. Receivers have a SIC radio to decode multiple signals simultaneously. The authors first prove the computational complexity of the scheduling problem with SIC and show that the problem is NP-hard. They also prove that the optimal decoding sequence is in terms of descending received power. The authors develop an ILP optimization model to minimize the schedule length that consists of one or more so called activation sets. Each activation set contains one or more links. Thus, the problem is to select the minimum number of activation sets to accommodate all links. However, the formulated ILP becomes intractable with increasing number of links and activation sets. The authors then propose a column generation [74] based method that decomposes the problem into a master and a sub-problem. The master problem is an Linear Program (LP)-relaxation of the ILP that replaces the collection of all activation sets with a subset that has a small cardinality. The sub-problem is called a pricing problem. The general idea of the pricing problem is to augment the LP solution of the master problem by selecting new activation sets to improve its objective value. The proposed approach stops when no new activation set can be selected.

In a similar work, Kontik *et al.* [63] jointly optimize scheduling and rate allocation of active links to derive the minimal length schedule that satisfies the traffic demand over each link. The authors consider single-hop multiple access wireless networks with SIC. They first formulate the scheduling as an LP problem where each variable represents a link set with ordered links. The transmission rate of links is calculated according to the decoding order of link sets that satisfy SIC constraints. Since the number of possible ordered link sets increases exponentially with the number of links in the network, the authors propose a column generation-based method to decompose the LP formulation as well. The master problem and its sub-problem are similar to [62]. The novelty of [63] is that the authors include the decoding order of simultaneous transmissions to construct link sets in SIC-based networks. They then use the obtained decoding order to determine the transmission rate of active links, so as to satisfy a given traffic demand.

Scheduling links whereby receivers have SIC is an NP-hard problem [62]. Hence, works such as [64–67] propose different greedy heuristic or approximation algorithms to construct a link schedule. Their objective is to minimize the schedule length. For example, the authors of [64] propose a heuristic algorithm to select links that can transmit simultaneously in each time slot. Specifically, in each time slot, the heuristic algorithm greedily chooses links that satisfy their SINR and remove unscheduled links that are unlikely to satisfy their SINR threshold or cause too much interference to other active links.

Lv *et al.* in [65] propose approximation algorithms that consider both the physical and protocol model [75]. Their aim is to investigate the schedule length and network capacity in a single channel SIC-based wireless network with multiple stationary transmitters and receivers. They consider grouping concurrent users to realize SIC. The proposed approximation algorithm first chooses and orders links to construct link sets. It selects a link with the least interference for each slot. Then in subsequent iterations, links with more interference are chosen sequentially.

Reference [66] considers a SIC-based wireless ad hoc network that consists of multiple directed links. Links have traffic demand, in terms of number of packets. The aim is to compute the minimal schedule length. The work in [66] constructs sets containing simultaneous links that satisfy their traffic demand in each time slot. The authors propose a novel metric to quantify the effect of adding a new link to a scheduled set. In particular, the metric quantifies the reduction in SINR when a new link is added to the set. The authors then propose a link scheduler that utilizes the proposed metric. The general idea of the proposed heuristic algorithm is to iteratively add a link that causes the least SINR reduction to existing scheduled links in each time slot.

In [67], the authors focus on uplink scheduling in a SIC-based wireless network that consists of multiple users communicating with a single receiver. The main problem is to schedule a set of concurrent users in each time slot and to determine the decoding order of each transmission. Their objective is either to maximize the link capacity or fairness among users, where the authors consider proportional fairness in terms of data rate.

A graph is widely used to model the effects of wireless interference [68]. For example, a conflict graph indicates links that mutually interfere and cannot be active simultaneously [76]. Each vertex in the graph indicates a link. There is an edge between two vertices if both links cannot be activated simultaneously. However, a conflict graph fails to model accumulated interference. Lv *et al.* in [68] and [69] propose two new graphs, called conflict set graph and weighted simultaneity graph, to model accumulated interference. For example, in [68], vertices of the proposed conflict set graph indicate single links as well as the conflict set of any single link.

Specifically, the conflict set of a link consists of the minimum amount of interference that can be decoded successfully by SIC receiver(s). Then based on the constructed conflict set graph, the authors propose an independent set based greedy scheduling scheme to schedule as many unscheduled links as possible in each time slot.

In a different paper [69], the same authors propose a weighted simultaneity graph to characterize link dependency and interference. Specifically, each vertex in the weighted simultaneity graph indicates both a link and its correlated link(s) that can transmit concurrently. The weighted simultaneity graph has two types of edges. One indicates two links cannot be activated simultaneously and the other reveals the decoding order of simultaneous links. The authors define the weight of vertices and edges as the receive power of links. Then they propose a new type of greedy scheduling scheme with two heuristic policies. The aim is to assign each link to the most suitable slot and schedule more links to the slots that are already chosen. The authors assume that each link can be activated in at most one slot. The key idea of the first heuristic policy is to schedule links to the first slot that can support the maximum number of concurrent links among available slots. This policy aims to balance the interference margin and the number of concurrent links between different slots. The other policy is to select a slot whereby a newly added link causes the minimum interference to other scheduled links. The aim of the said policy is to minimize the impact of a scheduled link on the capacity of the current slot, so as to activate more links in future slots. Compared to [68], the work in [69] considers aggregate interference of concurrent links and the maximum number of supported concurrent links when allocating slots and constructing link sets.

In [70], the authors study a Code Division Multiple Access (CDMA) based ad hoc network consisting of multiple mobile nodes with SIC capability. They assume that a SIC receiver has knowledge of the spreading sequence of all users. The aim is to improve the overall network throughput. The authors propose to use SEEDEX [77] for collision avoidance. The basic idea of SEEDEX is as follows: each node will generate a pseudo random schedule. Nodes within two-hop distances exchange their transmission schedules by transmitting a seed. Then a node is allowed to transmit if none of its neighbors are going to transmit. This exchange will repeat frequently to update schedule information and allow for the mobility of nodes. Then using the knowledge of which neighbors will possibly transmit at a given time, each node performs SIC to resolve packet collision and allows multiple concurrent transmissions. The schedule is divided into multiple time slots. Specifically, each time slot consists of transmit and receive parts.

The work in [71] uses the Q-learning algorithm [73] in a SIC-enabled wireless ad hoc network with a realistic channel model. Its objective is to maximize the number of transmitted packets. Each node independently determines time slots to transmit its packets using the Q-learning algorithm.

Table 2.1 summarizes the aforementioned works. We see that the objective of most works is to minimize the length of schedule used to activate all links. Only the authors of [62] and [63] have provided an optimization formulation for link scheduling problem with SIC. Reference [68] and [69] propose two new network graphs to model interference before proposing greedy algorithms to construct a link schedule. We see that all works consider stationary receivers and transmitters. Additionally, the transmit power is uniform for all links.

2.1.1.2 Single Time Slot Schedule

The authors of [60, 78–80] consider link scheduling over a single time slot. These authors formulate the link scheduling problem with SIC as an LP, ILP or MILP. Specifically, Yuan *et al.* in [60] consider a wireless system with multiple pairs of transmitters and receivers. Receivers have interference cancellation capability that allow multiple concurrent links. The authors consider three IC schemes, namely SIC, PIC and SLIC. In addition to receivers with interference cancellation capability, transmitters can perform cooperative transmissions. The aim of the work in [60] is to activate as many links as possible in one time slot. For each scheme, the authors propose an ILP model. Each link is assigned a weight that represents its queue size.

Prior Works	**Mobility of Nodes**	**Objective**	**Formulation**	**Solutions**
Yuan *et al.* [62]	Stationary	Minimize schedule length	IP	A column generation method
Kontik *et al.* [63]	Stationary	Minimize schedule length	LP	A column generation method
Goussevskaia *et al.* [64]	Stationary	Minimize schedule length	N/A	A heuristic algorithm
Lv *et al.* [65]	Stationary	Minimize schedule length and improve network capacity	N/A	An approximation algorithm
Kontik *et al.* [66]	Stationary	Maximize link capacity or fairness among users	N/A	A heuristic algorithm
Mollanoori *et al.* [67]	Stationary	Minimize schedule length	N/A	A heuristic algorithm
Lv *et al.* [68]	Stationary	Minimize schedule length	N/A	Construct a conflict set graph and run independent set based greedy scheme
Lv *et al.* [69]	Stationary	Best time slot selection	N/A	Construct a weighted simultaneity graph and run greedy algorithm with proposed heuristic policies
Lentz *et al.* [70]	Stationary	Improve the overall network throughput	N/A	A SEEDEX scheme
Mete *et al.* [71]	Stationary	Maximize the number of transmitted packets	N/A	A Q-learning algorithm

Table 2.1: A comparison of prior works that study TDMA link schedule with SIC.

Then the goal of each ILP model is to construct a transmission set with the highest total weight. The authors consider two threshold cases when determining the order of links for SIC scheme. In the first case, there is a fixed threshold value. For the second case, the threshold value is different; hence, there is no fixed order during the SIC decoding process.

Reference [78] focuses on uplink transmissions from a set of mobile stations to a set of dense small-cell base stations. The position of each mobile station and small-cell base station is fixed. The authors first propose a concurrent transmission graph model to reflect the conflict among different transmissions. Each vertex in the graph corresponds to an SIC opportunity. Two vertices are connected by an edge if the corresponding SIC opportunities conflict with each other. The authors then reduce the problem and use an independent set to represent a conflict free schedule. Each subset in an independent set consists of a small-cell base station and mobile stations with transmitted signals that can be decoded successfully. The authors define the weight of an independent set as total number of decoded mobile stations. The problem is to identify an independent set with the maximal weight in a single time slot.

Lei *et al.* in [79] study transmission scheduling as well as energy harvesting in a Wireless Powered Communication Network (WPCN). The network consists of one sink node (receiver), multiple users and one wireless power beacon that is responsible for charging users. Users transmit their data to a sink node. Time is divided into energy harvesting and data transmission. The objective is to maximize the throughput at the sink by jointly optimizing the time allocated for wireless charging and uplink data transmissions. Moreover, the authors construct transmission sets to achieve SIC. The authors formulate the throughput maximization problem as an LP. First, they use column generation to generate transmission sets. The throughput of each set is the product of its transmission time and sum transmission rate. Then the LP is used to determine the transmission time of each set. Three constraints are listed to balance the charging time and data transmission time. The first constraint ensures energy harvesting and data transmission time is within one block time. The second constraint ensures the energy used for data transmission cannot exceed harvested energy. The last constraint is to guarantee each user has enough time to communicate with the sink.

In [80], the authors study link scheduling in scenarios with multiple pairs of transmitters and receivers. Similar to [60], transmitters can perform cooperative transmission and receivers have SIC capability. Specifically, cooperative transmissions help create more interference and make it easier to cancel strong interference and perform SIC. The authors partition active transmitters into groups. They then determine the destination receiver of each group to reduce interference between receivers. Moreover, they ensure links in each group have sufficient SINR in each stage of SIC decoding process. Thus, to validate cooperative transmissions, the authors need to determine which transmitter should transmit and to which receiver as well as finding the optimal cancellation patterns to realize SIC. The authors first propose an ILP model. The objective is to maximize the number of concurrently active receivers. There are a number of constraints relating to (i) activation and grouping of transmitters/receivers, and (ii) SIC. The authors also introduce a bipartite graph and consider the problem of finding the maximum weight matching. The bipartite graph is divided into two sets that respectively represent transmission groups and receivers. If a receiver successfully performs SIC for a possible transmission group, a link between the receiver and that transmission group exists and its weight is set to one. The total weight corresponds to the total number of active receivers. In the proposed algorithm, the first step is to construct transmission groups, and then determine its total weight. They then search for a better transmission group via three ways: (i) add inactive transmitters to the group, (ii) delete transmitters, and (iii) swap transmitters between different groups.

Table 2.2 summarizes the aforementioned works. We see that except for [60], references [78–80] propose algorithms to schedule links in large-scale networks. We also see that all of these works consider stationary transmitters and receivers. More-

Prior Works	Mobility of Nodes	Objective	Formulation	Solutions
Yuan *et al.* [60]	Stationary	Maximize the number of activated links	ILP	N/A
Hou *et al.* [78]	Stationary	Maximize the number of activated links	N/A	Construct a concurrent transmission graph and run independent set based scheduling algorithm
Lei *et al.* [79]	Stationary	Maximize the network throughput	LP	A column generation method
He *et al.* [80]	Stationary	Maximize the number of activated links	ILP	Construct a bipartite graph

Table 2.2: A comparison of prior works that study single slot link scheduling with SIC.

over, these works do not consider power control and assume a fixed transmit power for all links.

2.1.2 Joint Topology Control and Link Scheduling

Topology control is a technique that is used to alter the underlying network to save energy, reduce interference between nodes and/or extend lifetime of the network [81]. To achieve these goals, some parameters can be modified, such as transmit power and active or sleep state of nodes. The work in [82–86] jointly studies link scheduling and topology control. The aforementioned papers are classified into: (i) active links with novel frameworks that favor SIC functionality and/or maintain the connectivity of the network; e.g., [82] and [83], and (ii) control transmit power of links to validate SIC; e.g., [83–86].

Gelal *et al.* in [82] consider nodes with SIC in multi-user Multiple-Input-Multiple-Output (MIMO) networks. The authors propose a framework that constructs topologies to favor SIC functionality. The framework consists of centralized and distributed solutions. These solutions aim to divide the network topology into several groups of links (sub-topologies) to facilitate SIC. The objective is to construct the minimum number of groups comprising of nodes that have a high decoding probability, so as to balance the medium access delay and the probability of successful reception. The difference between the proposed centralized solution and distributed solution is that with a distributed solution, each node requires only one-hop information to make topology control decisions. However, a centralized solution constructs a small number of sub-topologies first, so as to guarantee SIC decoding is successful with at least a certain probability.

Reference [83] studies channel assignment in SIC-based multi-hop Cognitive Radio Networks (CRNs). The authors consider an underlay paradigm that allows secondary users to communicate with both primary and secondary users whenever they do not cause interference to the transmissions of primary users [87]. In addition, secondary users can perform SIC to mitigate interference from primary users and other secondary users. The objective of [83] is to construct a conflict free CRN with the fewest number of channels. The authors also aim to guarantee the connectivity of CRN when primary users occupy a channel used by secondary users. The authors first proposed a centralized topology control algorithm that jointly considers transmit power control and channel assignment of SIC-equipped secondary users. They also design a distributed algorithm where secondary users construct a topology and assign a channel independently. Similar to [82], the authors of [83] use a directed graph to model a network topology. However, network connectivity is considered in [83].

Yuan *et al.* [84] consider a wireless network comprising of a number of co-channel links. Their aim is to evaluate the potential of interference cancellation in interference-limited environments. The authors focus on a max-min power control problem when interference cancellation is applied to receivers. Specifically, the authors jointly determine the transmit power of all transmitters and interference cancellation patterns to optimize the minimum SINR value. Their problems are to first determine the transmit power before selecting concurrent links and their decoding order in order to perform SIC and/or SLIC. The authors formulate the max-min power control problem with SLIC and SLIC as different MILP models. In particular, for the SIC case, the authors derive two MILP models to compare their performance and to gain insights into their relative merits. One is to exploit an optimality condition of SIC ordering, and the other explicitly models the SIC decoding order. The authors then propose a bisection algorithm [88] to solve their MILP formulations.

The work in [85] considers SIC link scheduling in a TDMA-based wireless network, where each link can be activated in multiple time slots for transmission. The aim is to efficiently utilize channel resources. Specifically, the authors define a demand satisfaction factor to address resource allocation fairness of links. The defined factor indicates the ratio between the amount of successfully transmitted traffic and traffic demand. Then the authors aim to maximize the minimum fairness to guarantee the transmission demand of the worst-case link. They formulate a link scheduling problem with joint power control and SIC as a Mixed-Integer Non-Linear Program (MINLP). However, non-linear constraints and mixed variables in the MINLP cause high computational complexity. The authors then propose an iterative algorithm to transform the MINLP into a maximization link scheduling problem and a series of minimization sub-problems. Specifically, formulated sub-problems are with linear constraints that minimize the total network power consumption. These decomposed problems are formulated as an ILP and/or LP. The authors further propose a two-stage algorithm with polynomial-time complexity. In the first stage, the authors built a conflict graph, and then adjust the transmit power of links with the same receiver in order to satisfy the SINR requirement at the receiver. Specifically, each vertex in the conflict graph corresponds to a link with a weight that represents the demand satisfaction factor of this link. After that, the authors update the conflict graph and choose the maximal independent set of the conflict graph as active links in each time slot.

Reference [86] considers a single-hop SIC-based industrial wireless network. The network consists of multiple users and one single-antenna base station that employs

SIC to decode and separate signals. The aim is to minimize the aggregate power consumption of users for uplink transmissions, so as to guarantee the real-time performance of users. The problem is to study the trade-off between power allocation and link scheduling that group users to realize SIC. The authors study both continuous and discrete transmit power cases. For both cases, they first solve the minimum power allocation problem whereby the aim is to determine the minimum aggregate power consumption. To solve the link scheduling problem, they use a bipartite graph and pose the problem of finding the maximum weight matching of the bipartite graph. The bipartite graph is divided into two parts that respectively model the list of users and the decoding indices that allow users to transmit concurrently. The edge between two nodes represents the scheduled slot of a user and the decoding order of that link at a base station. The authors relate the weight of each edge to an inverse number of the required minimal transmit power for scheduling and decoding of a user. They also propose a heuristic algorithm, specifically, a stochastic descent algorithm to solve the problem in polynomial time for the case with discrete transmit powers.

Table 2.3 summarizes the aforementioned works. All works joint consider topology control and link scheduling to realize SIC. References [82] and [83] propose both centralized and distributed algorithms to study SIC functionalities. Reference [85] and [86] propose graph-based algorithms. We see that most of these works assume perfect interference cancellation, except for [82]. The authors of [84] and [85] formulate the link scheduling problem with power control as an MILP and MINLP, respectively.

2.1.3 Cross-Layer Optimizations in Multi-Hop Networks

A number of works [57, 89–93] have also considered cross-layer optimizations that across the physical, MAC and network layer. Their aim is to study the benefits of SIC in multi-hop wireless networks. The main problems are to group links to

Prior Works	**Mobility of Nodes**	**Objective**	**Formulation**	**Solutions**
Gelal *et al.* [82]	Stationary	Minimize the constructed nodes groups	N/A	A framework with centralized and distributed solutions
Sheng *et al.* [83]	Stationary	Construct a conflict free CRN with the fewest required channels and maintain the connectivity of the CRN	N/A	Centralized and distributed algorithms
Karipidis *et al.* [84]	Stationary	Maximize the minimum SINR value	MILP	A bisection algorithm
Li *et al.* [85]	Stationary	Maximize the minimum resource allocation fairness	MINLP, LP, ILP	A conflict graph based algorithm
Xu *et al.* [86]	Stationary	Minimize aggregate power consumption of user equipments	N/A	A bipartite graph based algorithm and a heuristic algorithm

Table 2.3: A comparison of prior works that study topology control and link scheduling with SIC.

achieve SIC and balance flow over multiple time slots. The authors of these works assume unicast addressing and half-duplex channel.

Reference [57] jointly determines a set of concurrent links in each time slot and their corresponding transmission rates. The objective is to maximize the minimum throughput among all flows. Note that the number of links and the available modulation schemes at each node increases exponentially for large scale networks. The authors first use column generation to decompose the joint optimization problem. They then propose a tree-based greedy search method as well as a scalable simulated annealing based heuristic algorithm to solve an ILP scheduling sub-problem. In particular, a pricing algorithm is developed to generate feasible link schedules. The authors then solve the max-min flow routing master problem by using the generated schedules in the sub-problem.

In a different work [89], in addition to routing and scheduling, the same authors consider congestion control to maximize network utilities. In particular, the congestion control sub-problem can be solved at the source node of each flow by using local information. The routing and scheduling sub-problem is converted into a weight scheduling problem where the weight indicates the queue length at each node. The authors then consider a greedy maximal scheduling approach [94] for link scheduling problem in centralized settings. Additionally, they propose a search-based decentralized method to determine the minimum interference neighborhood of each link.

Ploumidis *et al.* [90] explore a distributed flow allocation scheme with the objective to maximize the average aggregate flow throughput as well as providing bounded delay when SIC is employed in a wireless mesh network. The authors implement a slotted-Aloha MAC mechanism for data transmission. They formulate the problem as a non-convex optimization model. Then based on the optimization model, they propose a scheme to determine the flow that is assigned to each path. The work in [91] provides a systematic study of SIC in multi-hop wireless networks by jointly considering time-based scheduling and flow routing with SIC. The authors use network throughput to quantify the potential of SIC. They formulate the cross-layer problem as an MILP.

Reference [92] aims to develop a bandwidth-aware routing protocol with SIC to achieve high end-to-end throughput. The authors note that not all SIC opportunities are amenable to throughput gains. Thus, they identify those SIC opportunities that can enhance throughput via novel SIC-able conditions. Moreover, these identified SIC opportunities can improve spatial reuse and guarantee transmission quality. Therefore, more simultaneous links are allowed to transmit. The authors propose a routing protocol with novel SIC-able conditions to identify these beneficial opportunities. The authors then formulate the problem of SIC-aware bandwidth computation as an LP to further study SIC benefits. They also develop a distributed heuristic algorithm to estimate the available path bandwidth in polynomial time.

Cheng *et al.* [93] propose an interference coordinated routing scheme for wireless multi-hop networks to achieve more concurrent transmissions, so as to lower the end-to-end delay. The proposed scheme is a distributed cross-layer design that consists of routing, link scheduling and interference-aware power control. Specifically, the scheme first constructs an initial path by an interference-aware routing algorithm. This routing algorithm captures end-to-end latency and spatial resource cost as routing metrics. Then the authors consider interference coordination and formulate the concurrent transmission of multiple links as an LP problem. Finally, the authors propose a distributed guard zone-based selection algorithm to iteratively explore the maximum feasible link set for each time slot.

Table 2.4 summarizes the aforementioned works. Except for [90] that considers a Slotted-Aloha MAC mechanism, other works [57, 89, 91–93] consider TDMA link scheduling. References [57, 91–93] formulate the cross-layer optimization problem as an LP, ILP and/or MILP. We see that except for [93], other aforementioned works do not consider power control of individual node. Block fading is assumed in all aforementioned works.

Prior Works	**Mobility of Nodes**	**Objective**	**Formulation**	**Solutions**
Qu *et al.* [57]	Stationary	Maximize the minimum throughput among all flows	ILP and LP	Column generation, a tree-based greedy search algorithm and heuristic algorithm
Qu *et al.* [89]	Stationary	Maximize the network utility	N/A	A greedy maximal scheduling approach and a search-based distributed approach
Ploumidis *et al.* [90]	Stationary	Maximize average aggregate flow throughput	N/A	An optimization-based scheme
Jiang *et al.* [91]	Stationary	Systematically study SIC in multi-hop wireless networks and maximize the throughput	MILP	CPLEX
Liu *et al.* [92]	Stationary	Achieve high end-to-end throughput	LP	A distributed heuristic algorithm
Cheng *et al.* [93]	Stationary	Achieve more transmission concurrence and lower the end-to-end delay	LP	A distributed guard zone based selection algorithm

Table 2.4: A comparison of prior works that study cross-layer optimization in multi-hop networks with SIC.

2.2 Air-Ground Communications

This section focuses on works that study low-altitude UAV communications. Recall that compared to communications with a fixed infrastructure, UAV communications afford a number of benefits, including (i) better coverage and capacity, especially for users located far from a base station [34], (ii) providing strong line-of-light links that facilitate reliable transmissions [14], (iii) serving as a platform for offloading traffic or computation [6], and (iv) prolonging the lifetime of WSNs or/and improving the amount of gathered data from a WSN [35]. However, UAVs are small in size, weight and have limited energy. Hence, there are constraints on their operational height, communication, coverage and lifetime. Thus, there is intense focus on improving UAV communications given the aforementioned resource constraints. In this respect, references [5, 14, 15, 51, 95] have provided a comprehensive survey and tutorial of past works on UAV communications in wireless networks. Specifically, these surveys/tutorials summarize UAV channel modeling methods [95], analytical frameworks and mathematical tools [5], issues encountered in UAV communications [51], and UAV communications for 5G and beyond [14][15].

The following sections group works according two aspects: (i) the optimal trajectory design of UAV(s); see Section 2.2.1, and (ii) joint optimization problems, such as trajectory planning, link scheduling and/or transmit power control; see Section 2.2.2.

2.2.1 Optimal Trajectory

A number of works such as [37, 96–102] have considered static hovering points and/or continuous trajectory design to (i) maximize the collected and/or forwarded data by UAV(s), (ii) minimize energy consumption of UAV(s), or (iii) minimize the flight time of UAV(s). References [37, 96–99] consider a single UAV that flies at a fixed altitude/height; see Section 2.2.1.1. References [100–102] study a path planning problem for multiple UAVs with variable heights; see Section 2.2.1.2.

2.2.1.1 Single UAV

Zeng *et al.* [37] consider a wireless communication system where a UAV is employed to send information to a ground terminal. Their aim is to maximize the energy efficiency (in bits/Joule) of a UAV. In particular, the authors consider the trade-off between communication throughput and the propulsion energy consumption of a single UAV. The problem is to optimize the UAV's trajectory. The authors propose an efficient algorithm to find an approximate optimal trajectory based on linear state-space approximation and sequential convex optimization [88] techniques. In [96], Li *et al.* consider a wireless network where a UAV acts as an aerial base station to serve multiple mobile users. The authors adopt Frequency Division Multiple Access (FDMA) for downlink communications between users and a UAV. Their objective is to maximize the sum-rate of downlinks. The problem in [96] is to find a control policy that determines the UAV's trajectory in each time slot. In particular, the authors consider two cases, where users move along specific or unknown paths. Under each case, they propose a deep reinforcement learning [103] based UAV control algorithm in which a UAV iteratively learns its trajectory.

Reference [97] studies a UAV-enabled communication system where ground users are subjected to latency constraints. In particular, the authors assume that the UAV moves to the location of each ground user for downlink communication. Each ground user must be visited within a predefined time window. They jointly optimize the UAV's trajectory and velocity. Their aim is to minimize the total energy consumption of the UAV while satisfying latency requirements of users and the UAV's energy budget. However, the considered joint optimization problem is non-convex and NP-hard. The authors then solve the problem via two consecutive steps. First, they propose two algorithms to obtain feasible UAV paths, namely dynamic programming and heuristic search. These two algorithms are designed based on a travelling salesman problem with time windows [104]. The difference between these two algorithms is that the heuristic search algorithm only foresees one hop ahead when checking the latency constraints of users. However, the dynamic programming method considers more outcomes in future hops when selecting a path. Second, for given feasible paths, the authors propose an energy minimization problem by optimizing the velocity of the UAV under an energy budget constraint. The energy minimization problem is convex and can be solved using standard methods [88].

The work in [98] considers a scenario where a UAV employs TDMA to collect data from a set of ground devices that are randomly distributed in a rectangular area. Each ground device is assumed to have a finite amount of data for transmission. The authors assume that a UAV does not know the exact position and data size of each ground device. Their aim is to maximize the total collected data of a UAV by optimizing the UAV's trajectory subject to a fixed flight time. They develop a Q-learning [73] based algorithm to overcome uncertainties in position and amount of data at ground devices and learn the optimal UAV's trajectory independently. Song *et al.* [99] consider a UAV-aided wireless cellular network that consists of multiple adjacent ground users and a single UAV. The problem is to design a UAV's trajectory with the objective to maximize collected data and ensure fairness of transmissions among all ground users. The authors first determine the hovering points of the UAV. These hovering points are then connected with a line to form a trajectory of the UAV. The authors then utilize a parallel projection algorithm [105] to calculate the location of hovering points.

2.2.1.2 Multiple UAVs

Reference [100] investigates a problem of fine-grained trajectory plan for multiple UAVs. These UAVs collaboratively collect data from a given WSN before transporting collected data to a ground base station. A fine-grained trajectory plan includes flight paths of UAVs as well as a detailed hovering and traveling plan on each path. The authors aim to minimize the maximum flight time of UAVs. They consider two cases with a single UAV and multi-UAVs for data gathering. For both cases, the authors first prove that the considered problem is NP-hard and then propose an approximation algorithm to obtain a path plan, respectively. In particular, for the case with multiple UAVs, the authors consider a bigger performance ratio for approximating the optimal solution of the considered problem.

In [101], the authors focus on studying an energy-aware three-dimensional (3D) deployment problem for a swarm of UAVs. In particular, they jointly consider travel time, flight altitude and battery lifetime to determine a 3D location of each UAV. Their aim is to maximize the total amount of data transmitted by UAVs within a limited network lifetime as well as mitigating interference between UAVs. The authors first formulate the considered problem as a non-convex non-linear optimization problem. They then transform the original optimization problem into an equivalent dual problem by applying a Lagrangian method [88]. This dual problem can be solved by a subgradient projection method that iteratively generates a minimal sequence of dual variables [106]. After that, the authors propose a heuristic algorithm that iteratively employs subgradient projection and interior-point methods [88]. The considered heuristic algorithm navigates each UAV to its target location where contributes the most to the total amount of data without severe interference.

The work in [102] considers data collection from distributed stationary IoT sensor devices with multiple UAVs. Communications between a UAV and ground sensor devices follow the standard TDMA protocol. The authors formulate a path planning problem for UAVs subjects to flying time and collision avoidance constraints. Their aim is to maximize the collected data from IoT sensor nodes. The authors first transform the considered path planning problem into a decentralized partially observable Markov Decision Process (MDP) [107]. They then propose a deep reinforcement learning [103] approach to approximate the optimal control policy of UAVs without prior knowledge of wireless channel characteristics. The novelty of [102] is to generate and apply control policies over a wide space of scenario parameters including (i) the number and the maximum flying time of UAVs, and (ii) the number, position and data amount of IoT devices.

Table 2.5 summarizes the aforementioned works that consider single UAV and multiple UAVs. We see that except for the work in [99] that considers OFDMA protocol, references [96, 98, 102] apply a TDMA protocol for communications between ground users/devices and UAV(s). We also see that the author of both [96] and [102] propose a deep reinforcement learning approach to obtain the optimal UAV's trajectory. Except for references [37, 96, 97] that consider downlink communications, other aforementioned works in [98–102] study uplink data collection.

2.2.2 Joint Optimization

A number of prior works that consider joint optimization to benefit the performance of UAV communication systems from different design dimensions. In particular, joint optimization problems mainly consist of (i) trajectory design of UAV(s), (ii) link scheduling, (iii) power control, (iv) resource allocation, and/or (v) energy harvesting. Ullah *et al.* [108] organize an extensive study that focuses on joint optimization problems of UAVs. To this end, Section 2.2.2.1 discusses works that jointly consider trajectory and link scheduling optimization. After that, Section 2.2.2.2 summarizes works that consider combinatorial optimizations with more than two parameters.

2.2.2.1 Joint Trajectory and Link Scheduling

The authors of past works such as [36, 109–114] have jointly considered developing link schedulers and UAV(s) trajectories in different networks. For example, references [36, 110] consider IoT data collection. The work in [111–114] considers WSNs.

Reference [109] studies a UAV-enabled wireless network where a UAV is employed as an aerial base station to serve multiple ground users. The authors aim to maximize the minimum throughput over ground users in a finite horizon. The original joint trajectory and scheduling design problem is formulated as a mixed integer non-convex optimization. The authors first relax binary variables for scheduling into continuous variables. They then propose an iterative algorithm by applying block coordinate descent technique to solve the problem. In particular, for a given UAV

Prior Works	Number of UAV(s)	Height of UAV(s)	Channel Access Method	Objective	Solutions
Zeng *et al.* [37]	Single	Fixed	N/A	Maximize the energy efficiency	An algorithm based on linear state-space approximation and sequential convex optimization techniques
Li *et al.* [96]	Single	Fixed	TDMA	Maximize the sum-rate	A deep reinforcement learning approach
Tran *et al.* [97]	Single	Fixed	N/A	Energy minimization with latency constraints	A heuristic search algorithm and a dynamic programming algorithm
Cui *et al.* [98]	Single	Fixed	TDMA	Maximize the cumulative collected data	A Q-learning algorithm
Song *et al.* [99]	Single	Fixed	OFDMA	Ensure fairness transmission	A parallel projection method
Luo *et al.* [100]	Multiple	Variable	N/A	Minimize the maximum flight time of UAVs	An approximation algorithm
Chou *et al.* [101]	Multiple	Variable	N/A	Maximize the total amount of data	A heuristic algorithm
Bayerlein *et al.* [102]	Multiple	Variable	TDMA	Maximize collected data	A deep reinforcement learning approach

Table 2.5: A comparison of prior works that study trajectory design in UAV communications.

trajectory, they optimize user scheduling by solving an LP. For any given scheduling, the UAV trajectory is optimized based on a successive convex approximation technique [88].

In [110], the authors propose a novel UAV-assisted IoT network, in which a low-altitude UAV is employed as a mobile data collector to assist terrestrial base stations in data collection and IoT devices' positioning. Their aim is to minimize the maximum energy consumption of all devices by jointly optimizing the UAV's trajectory and transmission schedule of devices. The authors first divide the original mixed integer non-convex optimization problem into three sub-problems. They then propose a differential evolution based method to iteratively solve these sub-problems. In particular, the first sub-problem is to select terrestrial base stations for each device that provide data collection and device positioning service. The second sub-problem is an LP problem that optimizes transmission schedule of devices for a given trajectory of the UAV. In each time slot, each device can choose to remain silent or transmit its data to a base station or the UAV. By solving this sub-problem, the authors can obtain the minimum energy consumption corresponding to a certain trajectory. The study of the second sub-problem is used in the third sub-problem to optimize the UAV's trajectory.

Shi *et al.* [36] study 3D trajectory design of multiple UAVs to facilitate IoT data collection. In particular, multiple UAVs periodically fly over IoT devices and relay their data to ground base stations. The authors aim to minimize the average path loss of device-to-UAV links. They first formulate the 3D trajectory design problem as an MINLP. Due to the quadratic and exponential terms as well as binary variables in the MINLP, the authors first transform the original problem into solvable forms by assuming some decision variables are constants. They then decompose the original problem into multiple sub-problems and iteratively solve them by applying a block coordinate descent method [115]. Specifically, sub-problems include designing scheduling of devices, horizontal trajectories and flying altitudes of UAVs.

You *et al.* [111] consider a UAV-enabled WSN that consists of multiple ground sensor nodes and a single UAV. The authors aim to maximize the minimum average data collection rate from all sensor nodes, while ensuring that data is received by the UAV under a given tolerable outage probability. They first formulate the problem as an optimization model. They then reformulate the original problem to a non-convex approximation form and propose an efficient algorithm to derive a sub-optimal solution. The proposed algorithm iteratively optimizes communication scheduling, horizontal and vertical trajectory of the UAV.

The work in [112] designs a framework for energy efficient data collection from a WSN using a mobile UAV. In particular, the authors assume that the UAV receives data only when hovering at collection stops. They formulate a joint optimization problem to determine (i) the position of UAV collection stops, (ii) a cluster of sensors to send data at each stop, and (iii) the optimal path among all stops that ensures data collection from all sensors. Their aim is to minimize the total energy consumption of both the UAV and sensors. The authors first formulate the problem as an MINLP model. They then propose a decomposition approach that iteratively achieve a sub-optimal solution. Specifically, they first use linearization to optimize UAV stop positions. Then they determine the subset of sensors for each stop. Each sensor is assigned to a collection stop that requires the lowest energy to collect data. After that, the authors use a travelling salesman problem algorithm [116] to determine the optimal path between collection stops.

In [113], the authors propose an autonomous UAV-enabled data gathering mechanism for delay-tolerant WSN applications. In particular, a self-trained UAV is employed as a flying mobile unit that collects data from ground sensor nodes during a pre-defined period of time. The authors develop an autonomous navigation and scheduling approach by combining two reinforcement learning based frameworks. Their objective is to minimize data collection time. In particular, a deep deterministic gradient descent algorithm [117] is used to autonomously decide the best trajectory in an obstacle-constrained environment. Additionally, a Q-learning algorithm [73] is developed to determine the order of nodes to visit for effective scheduling. Specifically, to obtain an effective scheduling, the authors consider the flying time between different nodes, energy consumption of the mobile UAV and data acquisition time windows of sensors.

Reference [114] studies multi-UAVs data collection from multiple sensor nodes in WSNs. UAVs are assumed to fly at a fixed altitude. The authors aim to (i) minimize the maximum mission completion time among all UAVs, and (ii) ensure each sensor node can successfully upload the targeting amount of data under a given energy budget. They jointly optimize 2D trajectory of UAVs as well as a wake-up scheduling and association for sensor nodes. The authors first propose a simple scheme where each UAV collects data while hovering. Under this setup, the original problem is reduced to finding the optimal hovering locations and time duration at each hovering location, as well as the flying speed and serving order at these locations. The authors propose an efficient algorithm by using the min-max multiple travelling salesman problem [116] and convex optimization [88] techniques. Next, the authors consider a more general scheme that enables continuous data collection for UAVs while flying. Under this scheme, the authors transform the original problem into a discretized equivalent with a finite number of variables. The transformed problem is then solved by applying time discretization [37] and successive convex approximation [88] techniques.

Table 2.6 summarizes the aforementioned works. We see that only references [36] and [114] consider multiple UAVs data collection. On the contrary, references [109–113] study a single UAV. The work in [109, 110, 114] assumes that UAV(s) fly at a fixed height. However, references [36, 111–113] obtain the trajectory of UAV(s) with variable heights. We also see that except for the work in [110], other aforementioned works all consider TDMA scheduling. Moreover, the authors of [36, 109–114] assume a fixed transmit power for all links. Both [112] and [114] apply a travelling salesman problem algorithm.

Prior Works	Number of UAV(s)	Height of UAV(s)	Channel Access Method	Objective	Solutions
Wu *et al.* [109]	Single	Fixed	TDMA	Maximize the minimum throughput among users	An iterative algorithm by applying block coordinate descent technique
Wang *et al.* [110]	Single	Fixed	N/A	Minimize the average path loss	A differential evolution based method
Shi *et al.* [36]	Multiple	Variable	TDMA	Minimize the average path loss	A block coordinate descent method
You *et al.* [111]	Single	Variable	TDMA	Maximize the minimum average data collection rate from sensor nodes	An efficient algorithm
Ghorbel *et al.* [112]	Single	Variable	TDMA	Minimize energy consumption of sensors and the UAV	A decomposition approach with linearization method and travelling salesman problem algorithm
Bouhamed *et al.* [113]	Single	Variable	TDMA	Minimize energy consumption of sensors and the UAV	An approach based on two reinforcement learning frameworks
Zhan *et al.* [114]	Multiple	Fixed	TDMA	Minimize the maximum mission completion time among all UAVs	An efficient algorithm by using min-max multiple travelling salesman problem and convex optimization techniques

Table 2.6: A comparison of prior works that jointly study trajectory design and link scheduling in UAV communications.

2.2.2.2 Combinatorial Optimization

This section focuses on works that consider combinatorial optimization problems with more than two parameters; see references [32, 118–123]. The key challenge is to balance a trade-off between (i) maximizing the collected data or average throughput, (ii) satisfying QoS requirements, and (iii) utilizing limited on-board energy for UAV communications. The considered combinatorial optimization problems in [32, 118–123] are non-convex. The authors first decompose original problems into multiple sub-problems and then propose efficient algorithms to iteratively solve sub-problems.

Mozaffari *et al.* [32] study multiple rotary-wing UAVs that act as aerial base stations to collect data from multiple ground IoT devices. Their aim is to enable reliable uplink communications for IoT devices with a minimum total transmit power. The problem is to jointly optimize the 3D placement and mobility of UAVs, device-UAV associations, and uplink power control. In particular, the authors consider a centralized FDMA uplink scheduling over the physical interference model. The proposed framework in [32] has two steps. First, given the location of IoT devices, the authors propose a solution to optimize deployment and association of UAVs. In this case, the formulated problem is decomposed into two sub-problems that are solved iteratively. The authors first fix the location of UAVs to jointly optimize device-UAV associations and transmit power of devices. Then under fixed device-UAV associations, they obtain the optimal 3D location of UAVs. In the second step of the considered framework, the authors analyze the optimal mobility patterns of UAVs to serve IoT devices in a time-varying network. In particular, UAVs dynamically update their locations depending on a time-varying activation process of IoT devices. In this case, the optimal 3D trajectory of each UAV is obtained to minimize the total mobility energy consumption of UAVs.

In a similar work, Wu *et al.* [118] consider a multi-UAVs enabled wireless communication system. UAVs are also regarded as aerial base stations to serve a group of ground users in a finite period by adopting TDMA. The authors aim to maximize the minimum throughput over all ground users in downlink communications. They jointly optimize communication scheduling and associations of multi-users, power control and trajectory design of UAVs. The authors propose an efficient iterative algorithm by applying block coordinate descent [115] and successive convex approximation [88] techniques. In particular, in each iteration, users' scheduling and associations, UAV's trajectory and transmit power are alternately optimized.

The work in [119] proposes a hybrid UAV-based cellular network with a single cell. The authors regard a single UAV as an aerial base station that flies cyclically along the cell. The UAV cooperates with a ground base station to offload traffic for cell-edge mobile terminals. According to the distance to a ground base station, mobile terminals are divided into two disjoint groups, namely inner disk region and exterior ring region. The authors aim to maximize the minimum throughput of all mobile terminals in the cell to achieve a fair throughput for all terminals. The problem is to jointly design (i) bandwidth allocation and user partitioning between a UAV and a ground base station, and (ii) the circular trajectory radius of a UAV. The authors propose a time-division based cyclical multiple access scheme [124] to schedule cell-edge mobile terminals communicating with a UAV.

Reference [120] studies a power efficient UAV-based WSN where a single UAV is deployed as an aerial base station to communicate with multiple ground sensor nodes. A given time horizon is equally divided into multiple time slots. The authors aim to minimize the total power consumption of a UAV and also guarantee a required transmission rate of sensor nodes. The problem is to jointly optimize downlink scheduling, power allocation, and UAV's trajectory. In particular, the authors assume that at most one node can communicate with the UAV in each time slot. Similar to the work in [118], the authors of [120] also propose an iterative algorithm that employs block coordinate descent and successive convex approximation techniques.

Zhan *et al.* [121] consider a cellular-connected UAV system that consists of a single UAV and multiple ground base stations. An energy-constrained UAV first collects data and then uploads its data to a cellular network under a given QoS requirement. The authors assume that a UAV uploads data to at most one ground base station at each time slot. They aim to maximize the uplink throughput by jointly optimizing the communication scheduling, operation time, trajectory and transmit power of the UAV. In [121], the authors study both online and offline design approaches. Specifically, offline approach only utilizes channel distribution information that is available prior to the UAV's flight. On the contrary, online approach utilizes instantaneous channel state information that is obtained by the UAV in real time along its flight. For offline approach, the authors propose an alternating optimization algorithm with the successive convex approximation technique [88]. Specifically, the proposed algorithm simultaneously updates UAV's velocity, time slot duration, transmit power allocation and communication time at each iteration. In online approach, the authors propose an adaptive online optimization algorithm and a low-complexity online algorithm based on receding horizon control [125].

Reference [122] considers uplink communications in a cellular-connected UAV network as well. However, compared to the work in [121], the authors of [122] consider multiple UAVs co-exist with ground user equipment. UAVs upload their inspected data to an individual ground base station in real time. The authors jointly exploit the optimal MIMO beamforming of ground base stations, association of UAVs, and UAV-height control. The authors assume that each UAV must be associated with one ground base station in each time slot. Their aim is to maximize the minimum achievable rate of UAVs. The authors propose a hierarchical bi-layer search algorithm that consists of inner layer and outer layer iterations to find locally optimal solutions iteratively. Specifically, they first fix hovering height of UAVs and use outer layer iterations to optimize the association of UAVs and the beamforming vectors of ground base stations. Outer layer iterations use bi-section search with a projection gradient method [126]. They then fix the association of UAVs and use inner layer iterations to optimize the height of UAVs and the beamforming vectors of ground base stations. Here, the authors exploit geometric program modeling [127] and a convex-concave procedure method [128].

In [123], the authors consider a wireless communication system with a mobile UAV and multiple ground users. They assume that the UAV flies at a constant height and collects data from ground users. Note that the UAV can only communicate with one ground user at a time. The objective is to minimize the total mission completion time. The problem of [123] is to jointly optimize the trajectory, altitude and velocity of a UAV as well as an uplink schedule of ground users over the physical interference model [75]. The authors first transform the original time minimization problem to a trajectory length minimization problem. Then they decompose the transformed problem into three optimization sub-problems. First, they optimize the UAV's trajectory by employing travelling salesman problem algorithm [116] and a convex optimization technique [88]. Second, they model a velocity and link scheduling optimization as an MILP problem and solve it via a block coordinate descent method [115]. Finally, in the altitude optimization problem, they use a Newton iteration method to compute the optimal UAV's altitude that maximizes the transmission range of the UAV.

Table 2.7 summarizes the aforementioned works. We see that references [118–120] and [32, 121–123] study downlink and uplink communications, respectively. We also see that a single UAV with fixed height is considered in references [119–121, 123]. The work in [32] and [122] studies multiple UAVs fly at variable heights. Additionally, the work in [118] considers multiple UAVs with fixed height trajectories. Moreover, the authors of [32, 118–120] regard UAV(s) as aerial base station(s) to communicate with ground users/devices. We see that block coordinate descent and successive convex approximation techniques are two popular methods adopted by past works that consider non-convex combinatorial optimization problems in UAV communications.

Prior Works	**Number of UAV(s)**	**Height of UAV(s)**	**Optimization Parameters**	**Objective**
Mozaffari *et al.* [32]	Multiple	Variable	3D placement and mobility of UAVs, device-UAV association, and uplink power control	Minimize total transmit power of IoT devices
Wu *et al.* [118]	Multiple	Fixed	Users' scheduling and association, UAV's trajectory and transmit power	Maximize the minimum throughput over ground users
Lyu *et al.* [119]	Single	Fixed	Bandwidth allocation and user partitioning between UAV and ground base station, and circular trajectory radius of the UAV	Maximize the minimum throughput of all mobile terminals
Hua *et al.* [120]	Single	Fixed	Downlink scheduling, power allocation, and UAV trajectory	Minimize the total power consumption of a UAV and guarantee required transmission rate of sensor nodes
Zhan *et al.* [121]	Single	Fixed	Communication scheduling, UAV operation time, trajectory and transmit power	Maximize the uplink throughput of UAV
Hou *et al.* [122]	Multiple	Variable	MIMO beamforming, user association, and UAV-height control	Maximize the minimum achievable rate of UAVs
Li *et al.* [123]	Single	Fixed	UAV trajectory, altitude, velocity, and link scheduling of ground users	Minimize the total mission time

Table 2.7: A comparison of prior works that consider combinatorial optimization in UAV communications.

2.3 Air-Ground Communications with NOMA

This section focuses on works that study UAV communications with the aid of NOMA. Recall that NOMA has been regarded as a key technology for 5G communication systems [54]. It improves spectrum efficiency and allows more users or devices to access networks by incorporating superposition coding at transmitters with SIC at receivers. The basic idea of NOMA is to exploit the difference in channel conditions between users. Compared to orthogonal multiple access (OMA), NOMA serves multiple users using power domain for multiple access. Consequently, a NOMA-equipped UAV is able to serve more ground users/devices so as to achieve a higher throughput and a lower access delay. To this end, Section 2.3.1 summarizes works that jointly consider trajectory and link scheduling optimizations in NOMA-aided UAV communications. After that, Section 2.3.2 discusses works that consider combinatorial optimizations with multiple variables, such as the altitude and trajectory of a single UAV, link scheduling, transmit power allocation, and/or bandwidth allocation.

2.3.1 Joint Trajectory and Link Scheduling

The works in [129–132] have jointly considered optimizing link scheduling and a UAV's trajectory or hovering locations. These works consider a single UAV. Reference [129] studies a UAV-enabled wireless network with a new proposed cyclical NOMA scheme. It exploits periodic channel variations and allows a UAV to cyclically communicate with two ground users in the same time slot. The authors of [129] aim to maximize the minimum throughput over all ground users. The problem is to jointly optimize downlink scheduling with cyclical NOMA and the trajectory of a UAV. The authors first formulate the problem as a non-convex MINLP. They then decompose the problem into two iterative optimization problems by applying a block coordinate descent method [115]. In particular, for a given UAV trajectory, the authors propose a two-layer optimization based algorithm. The proposed two-layer algorithm converts the communication scheduling problem of multiple users into two standard LPs that can be solved by CVX [133]. Then for a given schedule, they propose an iterative algorithm to optimize the trajectory of the UAV.

In a similar work [130], Wu *et al.* study an air-ground wireless network based on NOMA. A fixed-altitude UAV is deployed as an aerial base station to provide periodic service for a group of ground users. They aim to maximize the minimum sum rate over a time window. The problem is to jointly optimize downlink scheduling and the UAV's trajectory. Similar to the work in [129], the authors propose an iterative algorithm by employing block coordinate descent and successive convex approximation techniques. Before scheduling users, they first partition users with random locations into different subsets by applying the K-Means clustering algorithm [34]. They assume that in each time slot, a UAV can simultaneously serve two users in the same subset.

The work in [131] studies a NOMA-based cellular network with a single UAV, multiple ground users and ground base stations. A ground base station can serve a UAV and a static ground user simultaneously by utilizing NOMA. In particular, a UAV associates with at most one ground base station in each time slot. In addition, a UAV uploads data to a target ground base station when its horizontal location lies in the transmission region of that base station. The authors of [131] aim to minimize the mission completion time of the UAV. The problem is to jointly optimize the association between the UAV and ground base stations as well as the trajectory of the fixed-altitude UAV. In particular, the UAV associates with each ground base station at least once along its trajectory. To design the optimal UAV trajectory, the authors first design a fly-hover-fly trajectory and then propose two solutions based on this structure. The first one is an efficient solution with predefined hovering locations by using graph theory techniques [134]. The second solution is an iterative trajectory design algorithm that employs a successive convex approximation technique [88].

In [132], the authors focus on UAV-aided data collection from a NOMA-based wireless powered sensor network. In particular, a static single-antenna UAV first supplies energy to wireless powered sensor nodes that are located within a disk area. These sensor nodes then send back their information to the UAV. According to the Euclidean distance to the horizontal location of the UAV, nodes are divided into two groups. Each group provides a sensor node to construct a user pair. These two nodes transmit their respective data to the UAV simultaneously. The authors focus on designing a user pairing strategy and the optimal altitude of the static UAV. Their aim is to minimize the probability of unsuccessful transmissions by applying the designed pairing strategy.

Table 2.8 summarizes the aforementioned works. We see that references [129] and [130] consider downlink scheduling. The work in [131] and [132] studies uplink data collection. We also see that in all the aforementioned works, namely [129–132], a single UAV can serve only two users simultaneously. Except for reference [132] that considers a static UAV, the work in [129–131] studies a UAV that flies at a fixed altitude. Moreover, the considered problems in references [129–131] are non-convex. Thus, the successive convex approximation technique is used in [129–131] to find a locally optimal solution, respectively.

2.3.2 Combinatorial Optimization

This section summarizes works that study combinatorial optimization problems in NOMA-aided UAV communication networks. For example, reference [135] jointly optimizes the flying altitude of a UAV, transmit antenna beamwidth, the amount of transmit power and bandwidth allocated to multiple users. The work in [136] jointly optimizes scheduling, a UAV's trajectory and precoding vector. References [137–140] jointly optimize link scheduling, trajectory of a UAV and transmit power allocation. The formulated problems in the aforementioned works [135–140] are all mixed integer non-convex problems. The authors decompose the considered combinatorial optimization problems into multiple sub-problems and obtain the locally optimal solution iteratively.

Prior Works	**Downlink or Uplink**	**Number of Simultaneous Users**	**Objective**	**Solutions**
Sun *et al.* [129]	Downlink	Two	Maximize the minimum throughput	A block coordinate descent method, a two-layer optimization based algorithm, and an iterative algorithm
Wu *et al.* [130]	Downlink	Two	Maximize the minimum sum-rate	An iterative algorithm that employs block coordinate descent and successive convex approximation techniques
Mu *et al.* [131]	Uplink	Two	Minimize the mission completion time of a UAV	A graph theory based solution and a successive convex approximation technique based iterative algorithm
Shen *et al.* [132]	Uplink	Two	Minimize the probability of a node that fails to transmit its data	User pairing strategies

Table 2.8: A comparison of prior works that jointly study trajectory design and link scheduling in NOMA-aided UAV communications.

In [135], the authors consider a multi-user communication system. A single-antenna UAV is regarded as an aerial base station to serve multiple ground users by employing NOMA. Based on the Euclidean distance to the UAV, ground users are divided into two groups, namely cell-centered users and cell-edge users. The UAV employs NOMA to pair each cell-centered user with a cell-edge user. The authors aim to maximize the minimum rate (in nats/sec/Hz) of users under total power, total bandwidth, UAV altitude and antenna beamwidth constraints. They propose an inner convex approximation based path following algorithm [141] to solve the considered problem.

Zhao *et al.* [136] consider a cellular network with a mobile UAV and a static ground base station that serve multiple ground users separately. They assume that a fixed-altitude UAV serves associated ground users by employing a cyclical TDMA with a constant cycle duration. A ground base station employs NOMA to transmit data to its associated users. SIC is adopted at each base station-served user that allows it to decode composite signals from the UAV and other users. The objective in [136] is to maximize the sum-rate of all ground users. The authors first maximize the sum-rate of UAV-served users by optimizing the trajectory and scheduling of a UAV. An alternating optimization algorithm is proposed by using the block coordinate descent method [115]. The authors assume that the interference from a UAV to users served by a base station is limited to below a threshold value. Based on the obtained optimal scheduling and trajectory of the UAV, the authors design two precoding schemes to maximize the sum-rate of base station-served users. The first scheme intends to cancel the interference from a base station to users served by a UAV while the second scheme restricts the interference to a given threshold. In both schemes, the authors first transform the non-convex problem into a convex problem. They then propose an iterative algorithm to obtain the sub-optimal precoding vectors at the NOMA-aided ground base station.

Reference [137] considers a downlink UAV-assisted NOMA system. A fixed-altitude UAV and a NOMA-aided ground base station coordinate and transmit together to multiple ground users. Similar to the work in [135], ground users are divided into cell-center users and cell-edge users. The authors assume one cell-center user and one cell-edge user can be scheduled simultaneously in each time slot. The aim of [137] is to maximize the sum-rate of cell-edge users by taking advantage of the interference between the UAV and ground base station. The authors decompose the problem into three sub-problems and then alternately solve these sub-problems to obtain a sub-optimal low complexity solution [136][142]. In particular, to construct a schedule over the physical interference model [75], the authors first iterate through all cell-center users followed by cell-edge users until all users are scheduled .

Zhao *et al.* [138] consider a NOMA-assisted UAV communication system, where a UAV flies at a fixed altitude and collects data from large-scale IoT devices within a fixed flight time. Similar to references [135] and [137], the authors of [138] also divide IoT devices into two groups. NOMA allows each group to have an active device that transmit together in each time slot. The objective is to minimize the total energy consumption for data collection of IoT devices. The authors first use a generalized benders decomposition [143][144] to decouple the scheduling and transmit power allocation of IoTs to obtain the optimal scheduling. Then with a given scheduling, they propose a two-step iterative optimization algorithm to obtain the optimal trajectory of a UAV and the transmit power of IoTs by applying the successive convex approximation technique [88]. The authors also propose a low-complexity greedy algorithm to simplify the optimal trajectory and scheduling design.

In [139], the authors propose a time-efficient data collection scheme, in which multiple fixed ground devices upload their data to a UAV via NOMA. The UAV flies in a straight line with a fixed altitude. It prefers to collect data from nearby devices with better uplink channels. The objective of [139] is to minimize the flight time of the UAV and guarantee that the UAV collects sufficient data from ground devices. In particular, all ground devices are assumed to have the same minimum data transmission threshold. First, based on a given trajectory and channel gains of device-UAV uplinks, the authors propose an effective scheduling strategy to schedule simultaneous devices in each time slot. They then propose an alternating optimization based iterative algorithm to alternately optimize the transmit power of devices and the trajectory of the UAV. The successive convex approximation technique [88] is also applied in this step. The schedule of devices is updated accordingly at the end of each iteration.

The work in [140] considers a NOMA-based downlink wireless network with a fixed-altitude UAV. The UAV is regarded as an aerial base station that periodically serves two simultaneous ground users. The flying period of the UAV is divided into multiple sub-periods to denote the dynamic change of a UAV's trajectory. The authors aim to maximize the total energy efficiency and satisfy the QoS requirements of users. They first propose a novel matching and swapping algorithm that based on the matching theory [145] to schedule users in each sub-period. To solve the power allocation problem, the authors first transform it via a logarithmic approximation [146]. Then they use a Lagrangian method to obtain a power allocation solution. After that, the successive convex approximation technique [88] is used to obtain the optimal UAV trajectory. Finally, according to the proposed algorithm of each subproblem, the authors provide a joint iteration algorithm with a lower complexity. The iteration algorithm obtains the schedule and transmit power allocation of users as well as the trajectory of the UAV iteratively.

Table 2.9 summarizes the aforementioned works. We can see that the work in [135, 138, 139] considers uplink communications. On the contrary, references [136, 137, 140] study downlink transmissions. Except for [136] and [139], other references [135, 137, 138, 140] assume that two simultaneous signals can be decoded successfully by employing NOMA. We also see that all aforementioned works consider a single UAV that flies at a fixed altitude.

Prior Works	Downlink or Uplink	Number of Simultaneous Users/Devices	Objective	Solutions
Nasir *et al.* [135]	Uplink	Two	Maximize the minimum rate of users	An inner convex approximation based path following algorithm
Zhao *et al.* [136]	Downlink	N/A	Maximize the sum-rate of ground users	A block coordinate descent method and an iterative algorithm
Zeng *et al.* [137]	Downlink	Two	Maximize the sum-rate of cell-edge users	An iterative algorithm
Zhao *et al.* [138]	Uplink	Two	Minimize the total energy consumption of IoT devices and accomplish data collection	A successive convex approximation technique based iterative algorithm and a greedy algorithm
Wang *et al.* [139]	Uplink	Three	Minimize the flight time of a NOMA-aided UAV	A device scheduling strategy and an alternating optimization based iterative algorithm
Li *et al.* [140]	Downlink	Two	Maximize the total energy efficiency	A matching and swapping algorithm, the Lagrangian method, the successive convex approximation technique, and a joint iteration algorithm

Table 2.9: A comparison of prior works that study combinatorial optimization problems in NOMA-aided UAV communications.

2.4 SAGINs

This section focuses on works that study SAGINs. Recall that a SAGIN interconnects space, air, and ground segments/networks to enlarge coverage and increase network resilience [1]. They are able to support data delivery with low latency, high throughput and reliability. To achieve these QoS requirements, SAGINs adopt different communication protocols in each network or multiple interconnected networks. Specifically, these protocols address a number of issues pertaining to individual networks, such as distinct channel characteristics, interference, high transmission latency and limited energy storage. To this end, Section 2.4.1 summarizes works that investigate optimal routing issues in SAGINs. After that, Section 2.4.2 discusses combinatorial optimization problems in SAGINs.

2.4.1 Routing

References [147–151] study routing problems that aim to (i) load balance, and (ii) guarantee delay and/or throughput. For example, references [147] and [148] propose hierarchical routing algorithms. References [149–151] outline a greedy solution, a deep learning based method, and an MILP-based solution to solve routing problems, respectively.

Pace *et al.* [147] consider multiple ground terminals and control stations, a set of HAPs, and GEO satellites. Their aim is to (i) minimize the maximum link usage, and (ii) load balance the network. They propose a hierarchical routing algorithm in both HAP and GEO layers. For a given pair of source and destination terminals, the general idea is to find a set of candidate paths from inter-HAP links or HAP-satellite links with the minimum number of hops. Also, the proposed routing algorithm first selects candidate paths with a lower end-to-end delay when compared to that of direct terrestrial-satellite links. It then selects a path with the least congestion from candidate paths. If there is no available path, a source terminal will directly transmit its packets via a satellite to a destination terminal. In [148], the same authors apply the routing algorithm from [147] to various HAP constellations with different number of HAPs. They aim to (i) investigate the robustness and scalability of their novel routing algorithm, and (ii) guarantee delay and throughput in a SAGIN.

Reference [149] studies a cross-layer gateway selection problem for data delivery in a SAGIN with inter-layer link capacity constraints. The authors consider data delivery process from each terrestrial node to a satellite. They use the average expected transmission count [152] to measure the quality of wireless links. In particular, a small expected transmission count indicates a wireless link with higher quality. The objective of [149] is to minimize the average expected transmission count subject to the capacity constraint of satellite-aerial links. The authors assume gateway nodes in both terrestrial and satellite layers are given. The total traffic from terrestrial layer and the traffic distribution in aerial layer is known as well. The problem is to obtain the optimal set of gateway nodes at the aerial layer. In particular, multiple selected gateway nodes serve as transfer stations and cooperatively establish inter-layer connections for data exchange at the aerial layer. The authors propose two algorithms to select aerial gateway nodes, namely a basic enumeration algorithm and a greedy optimization algorithm. The basic enumeration algorithm generates and lists all possible combinations of gateway nodes from all aerial nodes before selecting gateway nodes. By contrast, the proposed greedy algorithm provides a solution that iteratively selects locally optimal gateway nodes.

In [150], Kato *et al.* consider the use of deep learning to optimize the performance of a SAGIN. They focus on utilizing a convolutional neural network [153] to improve traffic control performance at the satellite segment. The considered satellite segment consists of three layers, namely GEO, MEO and LEO. The authors only consider inter-layer links in both MEO and GEO layers as well as links that connect these two layers. They regard two MEO satellites as a pair of source and destination nodes. There are multiple paths between each pair of MEO satellites. The authors first combine paths for all MEO satellite pairs to construct a convolutional neural network. They then utilize an online training method proposed in [154] to train the convolutional neural network; so as to obtain the optimal path for each pair of MEO satellites. The objective is to minimize the delay caused by the training process.

The work in [151] studies a joint service placement and routing problem for on-board passenger services such as providing Internet connection on airplanes [155]. This can be achieved through satellites and/or direct air-to-ground links. However, due to the movement of airplanes, guaranteeing Internet connection with an acceptable QoS requirement and being low cost are important. Hence, the authors consider optimizing (i) ground data centers to deploy Internet connection service, and (ii) paths to provide Internet connection on airplanes. They aim to minimize Internet connection service cost and guarantee bandwidth and latency. The authors consider two cases: (i) a static case with a single time slot, and (ii) a mobility-aware case that considers the flight trajectory of airplanes. For each case, they formulate an MILP to solve the considered joint optimization problem.

Table 2.10 summarizes the aforementioned works. We see that references [147–150] only study a routing problem in SAGINs. However, the work in [151] jointly optimizes routing and service placement for on-board Internet connectivity services. The work in [147] and [148] focuses on selecting the optimal path in both satellite and aerial segments. However, references [149] and [150] consider routing problems in aerial segment and satellite segment, respectively. The work in [151] studies routing problem in all three network segments of a SAGIN.

2.4.2 Combinatorial Optimization

This section summarizes works that consider combinatorial optimization problems; see [156–161]. In particular, these works mainly consider (i) the deployment of UAV(s), (ii) scheduling tasks or uplinks from ground devices/nodes, (iii) offloading tasks, (iv) resource allocation, and/or (v) transmit power control. Note that a task can be executed by a ground network or offloaded to an air and/or satellite segment in SAGINs. Except for reference [158] that considers a single UAV, the work in

Prior Works	Effective Segments	Problem	Objective	Solution
Pace *et al.* [147]	Satellite and aerial	Select paths between HAP and GEO layers	Minimize the maximum link usage and load balance the network	A hierarchical routing algorithm
Pace *et al.* [148]	Satellite and aerial	Study the proposed routing algorithm in [147] under different topologies	Provide a solution to guarantee QoS requirements	A hierarchical routing algorithm
Shi *et al.* [149]	Aerial	Select gateway nodes in the aerial segment	Minimize the average expected transmission count	A basic enumeration algorithm and a greedy optimization algorithm
Kato *et al.* [150]	Satellite	Obtain the optimal path for each MEO satellite pair	Minimize the effect of delay on network performance	Combine an online training method with a convolutional neural network
Varasteh *et al.* [151]	Satellite, aerial and ground	Joint service placement and routing	Decrease Internet connection service cost, guarantee bandwidth and latency	An MILP solution

Table 2.10: A comparison of prior works that study routing problems in SAGINs.

[156, 157, 159–161] employs multiple UAVs for data collection and/or delivery.

Jia *et al.* [156] consider a SAGIN that consists of multiple LEO satellites, fixed-wing UAVs, and Internet of Remote Things (IoRT) sensors. In particular, the authors consider two transmission modes to support transmitting the data collected from IoRT sensors back to Earth, namely carry-store mode and satellite-relay mode. For delay-tolerant data, the carry-store mode is used by UAVs to first collect data from IoRT sensors and then carry the data to a ground destination station. Then the data will be transmitted from the ground destination station to a ground data processing center. Compared to the carry-store mode, the satellite-relay mode allows satellites to relay delay-sensitive data from UAVs to a ground data processing center via inter-satellite links and satellite-ground downlinks. The objective of [156] is to minimize the total energy consumption while collecting all data from IoRT sensors. The problem is to design the trajectory of UAVs, schedule IoRT sensors and set transmission modes. In particular, the authors assume each IoRT sensor can only be connected with one UAV at each time slot. The considered problem is formulated as an ILP and proved to be NP-hard. Hence, the authors first decompose the problem into a restricted master problem and a pricing problems by utilizing Dantzig-Wolfe decomposition [162]. They then use the column generation method [74] to determine the trajectory of UAVs and the uplink schedule of IoRT sensors. The proposed approach stops when the energy consumption of UAVs cannot be minimized by any new columns from pricing problems.

In [157], the authors present a joint communication and computation SAGIN framework that provides edge/cloud computing services to remote IoT users. An IoT user can execute tasks with computation requirements by itself or offload to UAV edge servers or to a cloud through LEO satellites. They jointly consider resource allocation and task scheduling for UAVs as well as a computing offloading problem for a SAGIN. In the resource allocation and task scheduling problem, the authors aim to minimize the total delay of tasks. They first formulate the problem as a mixed-integer non-convex model and then propose a heuristic algorithm to obtain a sub-optimal solution. The general idea of the proposed heuristic algorithm is to first schedule tasks that will cause less delay. The authors then propose a reinforcement learning based scheduling approach for task offloading. The objective is to minimize the total system cost in terms of tasks delay, energy consumption of IoT users, usage costs of UAVs and satellites.

In a similar work [158], Zhou *et al.* provide a task scheduling policy that considers dynamic task arrival from IoT devices. A single UAV collects delay-oriented computing tasks from IoT devices and then makes online offloading decisions. In particular, the collected tasks can be (i) locally processed at the UAV, (ii) offloaded to a nearby BS, and (iii) offloaded to a remote LEO satellite. The authors aim to minimize offloading and computing delay of all tasks over multiple time slots under a given UAV energy capacity constraint. The authors first re-formulate the considered delay-oriented tasks scheduling problem as a constrained Markov decision process [163] and use it to determine a time-invariant decision. Compared to the work in [157], the authors of [158] provide a deep risk sensitive reinforcement learning based algorithm that defines a risk function to capture whether the total energy consumption of the UAV violates the given energy capacity.

Reference [159] considers an IoT computation offloading system that consists of a single LEO satellite, multiple UAVs and ground IoT devices. IoT devices execute tasks with computation requirements. UAVs serve as edge nodes that provide edge computing and caching capability to IoT devices. A LEO satellite provides cloud computing services for its coverage area. The authors aim to minimize the maximum delay among IoT devices subject to the maximum available energy and tolerable delay constraints. The problem is to jointly (i) allocate computation tasks, transmit power, bandwidth and computation resource of UAVs, (ii) schedule the association between IoT devices and UAVs, and (iii) design the position of UAVs. The authors assume that IoT devices offload their computation tasks to UAVs via FDMA. In addition, each IoT device can only connect with one UAV at each time slot. The authors then solve the problem using block coordinate descent [115] and successive convex approximation [88].

The work in [160] considers a space-air-ground WPCN, where multiple UAVs charge ground nodes and relay data from ground nodes to a LEO satellite. The height of a LEO satellite and each UAV is fixed. When UAVs collect data from ground nodes, a FDMA protocol is applied. A time slot is divided into three parts: (i) ground nodes harvest energy from UAVs, (ii) UAVs collect data from ground nodes, and (iii) UAVs decode data and forward to the LEO satellite. The authors aim to maximize the system sum-rate by jointly optimizing time slot division, sub-channel allocation, transmit power control and the deployment of UAVs. They first apply an alternating optimization method and a successive convex approximation technique [88] to transform the non-convex problem into a tractable form. They then propose a near-optimal multi-variable alternating iterative algorithm to solve each sub-problem iteratively.

Wang *et al.* [161] consider a space-air-ground IoRT network with a LEO satellite, multiple fixed-altitude UAVs and smart ground devices. Similar to the work in [160], each UAV is regarded as a relay that amplifies and forwards data from ground devices to a LEO satellite. Moreover, in each time slot, each UAV serves at most one device, and vice versa. The authors do not consider direct communication between UAVs. The objective of [161] is to maximize system capacity. The problem is to jointly optimize smart devices connection scheduling, the trajectory of UAVs as well as the transmit power of ground devices and UAV relays. The authors first decompose the mixed integer non-convex problem into three sub-problem. They then propose an iterative algorithm to solve these sub-problems alternately by applying variable substitution [164], block coordinate descent [115] and successive convex approximation [88] techniques.

Table 2.11 summarizes the aforementioned works. We see that the work in [159] and [160] employs FDMA for data collection between ground nodes/devices and UAVs. We also see that LEO satellite(s) are utilized in all the aforementioned works; see [156–161]. In particular, references [156–158] study multiple LEO satellites.

Prior Works	Number of Satellite(s)	Height of UAV(s)	Objective	Solution
Jia *et al.* [156]	Multiple	Variable	Minimize the total energy consumption	A column generation method
Cheng *et al.* [157]	Multiple	Variable	Minimize the total system cost	A novel reinforcement learning based scheduling approach
Zhou *et al.* [158]	Multiple	Variable	Minimize offloading and computing delay of all tasks	A novel deep risk-sensitive reinforcement learning based algorithm
Mao *et al.* [159]	Single	Variable	Minimize the maximum delay among IoT devices	An alternating optimization algorithm based on block coordinate descent and successive convex approximation techniques
Jia *et al.* [160]	Single	Fixed	Maximize the system sum-rate	A near-optimal multi-variable alternating iterative algorithm
Wang *et al.* [161]	Single	Fixed	Maximize the system capacity	An iterative algorithm based on variable substitution, block coordinate descent and successive convex approximation

Table 2.11: A comparison of prior works that study combinatorial optimization problems in SAGINs.

By contrast, a single LEO satellite is considered in references [159–161]. Except for references [157] and [158] that study task scheduling, the work in [156, 159–161] obtains the optimal schedule between ground nodes and UAVs, respectively. Moreover, multiple fixed-height UAVs are studied in reference [160, 161]. On the contrary, the work in [156–159] considers UAV(s) that fly at variable heights.

2.5 Summary

To conclude, this chapter has discussed prior works that consider the following technologies:

1. *SIC*. The main advantage of SIC is allowing multiple receptions at a receiver.

There are three research directions/aims: (i) link scheduling, (ii) joint topology control and link scheduling, and (iii) cross-layer optimizations in multi-hop networks. The objective of works that consider link scheduling in a single or multiple time slots includes minimizing schedule length [62–68], maximizing link capacity or network throughput [65, 70, 71, 79], and maximizing the number of simultaneous links [60, 78, 80]. Works that jointly study topology control and link scheduling aim to (i) propose novel frameworks that favor SIC functionality [82][83], and control the transmit power of links [83–86]. The main problems addressed by works that consider cross-layer optimization are grouping of links to achieve SIC and balancing flows over multiple time slots; see [57, 89–93].

2. *Air-ground communications.* The key features of air-ground networks are (i) flexible deployment, (ii) better coverage and capacity, and (iii) strong line-of-sight connections. To this end, prior works study (i) the optimal trajectory design of UAV(s), and (ii) joint optimization problems that include link scheduling, transmit power control and allocation with trajectory design of UAV(s). The considered joint optimization problems are formulated as mixed linear non-convex models in [32, 36, 109–114, 118–123]. Block coordinate descent and successive convex approximation techniques are frequently utilized to obtain sub-optimal solutions; see [36, 109, 118, 120, 121, 123]. Moreover, techniques used to solve the travelling salesman problem are frequently employed to determine the optimal path; see [112, 114, 123].

3. *Air-ground communications with NOMA.* Networks that employ NOMA to use superposition coding at transmitters and SIC at receivers. These advances help improve spectrum efficiency and allow more users/devices to access networks. The summarized works mainly consider (i) trajectory design of UAV(s), (ii) link scheduling, (iii) resource allocation, and (iv) transmit power control. In particular, references [129–132] jointly consider optimizing link scheduling and

a UAV's trajectory or hovering locations. References [137–140] jointly optimize link scheduling, trajectory of a UAV and transmit power allocation. In addition, the work in [135] jointly optimizes the flying altitude of a UAV, transmit antenna beamwidth, the amount of transmit power and bandwidth allocated to multiple users.

4. *SAGINs.* The key features of SAGINs are interconnecting space, air and ground networks/segments to (i) achieve large coverage, and (ii) provide a high throughput and reliability data delivery and/or collection. The summarized works study (i) routing problems, and (ii) combinatorial optimizations include trajectory design of UAV(s), scheduling tasks and nodes/devices, offloading tasks, resource allocation and/or power control. The objectives of these works mainly include (i) maximizing the quality of wireless links [149], (ii) guaranteeing QoS requirements on latency and/or throughput; see [147, 148, 150, 151, 157–159], and (iii) maximize the sum-rate and/or capacity of the system [160] and [161].

Thus far, existing works have the following gaps. First, for TDMA link scheduling with SIC capable nodes, prior works only consider static transmitters and receivers; see [57, 62–69, 71, 89–93]. References [57, 66, 71, 90, 92] are the only works that aim to maximize throughput or the number of transmitted packets. Moreover, most works do not propose a distributed MAC that allows each user/device to obtain a schedule independently. Only the work in [71] outlined a Q-learning algorithm. The work in [92] and [93] provides a distributed heuristic algorithm, respectively. Second, most works that consider NOMA-assisted UAV communications assume that UAV(s) fly at a fixed altitude; see [129–131, 135–140]. Reference [139] is the only work that assumes a receiver can serve three nodes simultaneously. For most works, only two simultaneous nodes can be decoded successfully by SIC receivers. Moreover, works such as [131, 132, 135, 138, 139] do not provide a distributed MAC to obtain an uplink schedule with the maximum energy efficiency. Third, past works that consider routing problems in SAGINs only focus on aerial or satellite segment and assume gateways or paths in other networks/segments are given; see [147–150]. Moreover, prior works in [157] and [158] only consider scheduling tasks rather than links in SAGINs. Moreover, the work in [156, 159–161] considers scheduling uplinks between ground nodes/devices and UAV(s). These works do not aim to maximize the minimum flow among all uplinks. In addition, no past works jointly consider routing and link scheduling in SAGINs. Moreover, prior works that study SAGINs do not consider multi-user detection or interference cancellation to allow a node to receive from multiple transmitters simultaneously.

This thesis thus considers three research questions to fill in the aforementioned gaps: (i) obtain the optimal uplink schedule where multiple ground users upload data to a mobile UAV equipped with a SIC radio, (ii) jointly construct the optimal uplink schedule and altitude-changed trajectory of a SIC-enabled mobile UAV, and (iii) jointly obtain the optimal route and uplink schedule in a SIC-enabled SAGIN.

To this end, Chapter 3 presents an uplink scheduler that maximizes the total amount of data collected by a SIC-enabled rotary-wing UAV. In particular, Chapter 3 studies the impact of SIC on the number of simultaneous ground devices and the average throughput of each ground device.

Chapter 3

Link Scheduling for Data Collection in SIC-Capable UAV Networks

As shown in Chapter 2, past works that consider uplink scheduling in UAV communications with NOMA are focused on minimizing the energy consumption or flight time of the UAV [131, 138, 139]. However, they do not aim to maximize the sum-rate or average throughput. Moreover, these works assume UAV can only serve two users simultaneously. To this end, this chapter considers deriving a high throughput uplink schedule for a UAV equipped with a SIC radio. The main research question is to take advantage of the different channel gain from ground devices and determine an uplink transmission schedule for use over multiple pre-known data collection points. In this respect, this chapter makes the following contributions:

- The *novel* uplink scheduling problem is mathematically modeled as an ILP, which can be used to compute the optimal transmission schedule for each data collection point. Its goal is to maximize the amount of data collected by a UAV over multiple data collection points. The physical interference model [75] is considered when scheduling uplinks from ground devices to a single UAV. The maximum number of simultaneous uplinks that a UAV can decode follows the work in [61]. Note that works that consider power control are complementary to this work.

- As this problem is NP-hard, this chapter also contains a Cross-Entropy (CE)-based method [165] and a novel heuristic called Greedily Construct Transmission Schedule (GCTS) for use when there are a large number of ground devices and data collection points. The basic idea of the CE-based method is for the UAV to collect channel information of ground devices at all data collection points upfront. It then learns to identify the link set to be used at each data collection point. The basic idea of GCTS is to greedily include as many ground devices that have yet to transmit into a transmission set at each data collection point.

- The aforementioned ILP solution, CE-based method and novel heuristic GCTS are centralized approaches that are run by the UAV to construct a link schedule. This chapter then outlines a distributed approach that is run by both the UAV and ground devices. The proposed novel, distributed Medium Access Control (MAC) called Collection Point Selection Protocol (CPSP). It enables each ground device to independently learn the best data collection point it should used to transmit to the UAV.

- The evaluation in this chapter studies how the following factors affect the resulting schedule and average throughput; namely (i) different number of ground devices, (ii) different number of data collection points, (iii) speed and heights of the UAV, (iv) distance between the location of adjacent ground devices, and (v) different smoothing parameters and temperature parameters.

- The collected results show that SIC helps double the amount of data collected by the UAV. Moreover, the CE method and GCTS are capable of producing a schedule that is near optimal. Additionally, CPSP yields a schedule with higher average throughput than Slotted Aloha. Further, with more ground devices, the average throughput increases. On the contrary, the average throughput decreases with more data collection points. Moreover, a higher height of a mobile UAV results in a smaller average throughput.

The rest of this chapter is structured as follows. Section 3.1 introduces the network setup and notations. Section 3.2 presents the ILP model. Section 3.3 shows how the CE-based method can be used to compute the solution for large scale networks. Section 3.4 presents a novel heuristic algorithm GCTS. In Section 3.5, the details of novel MAC protocol CPSP is presented. Section 3.6 then discusses results and Section 3.7 concludes this work.

3.1 Preliminaries

Table 3.1 summarizes common nomenclature in this work. The considered system in this chapter consists of multiple single-antenna ground devices and a mobile SIC-capable UAV that operates on the same frequency. Let $\mathbf{G}$ be the set of ground devices, where $G = |\mathbf{G}|$. These ground devices are indexed as $1, 2, \ldots, G$. The first ground device, aka g_1, is set as the origin. For ease of exposition, ground devices are spaced equally along a straight line with a length of d meters. Also, these ground devices always have data to transmit.

UAV u flies horizontally at a fixed altitude h and is used to collect data from ground devices. Note that works that consider trajectory control are complementary to our work. The UAV moves with a constant speed s and is initially located above ground device g_1. It is assumed to collect data from ground devices at M data collection points. At each data collection point, the UAV will announce itself by sending a beacon message to inform ground devices of their transmission time. This is then followed by G uplink transmission slots; each slot is assigned to one ground device to transmit its channel coefficient to the UAV. Also, in practice, the beacon message will also consist of a preamble to synchronize ground devices. This allows ground devices to synchronize their transmission time. Let $\mathbf{M}$ be the set of data

Symbol	**Description**
1.Basic System Setting	
u	The mobile UAV
G	Ground devices
M	UAV data collection points
d	Distance between the location of adjacent ground devices
L_{max}	The maximum number of simultaneous uplinks that the UAV can decode
2. Sets	
$\mathbf{G}$	Set of ground devices
$\mathbf{M}$	Set of data collection points
L^m	Set of uplinks at collection point m
$\mathbf{C^m}$	Collection of link sets at collection point m that satisfy SIC
3. Variables and Parameters	
i	Index of ground devices $i \in \mathbf{G}$
m	Index of data collection points $m \in \mathbf{M}$
x_j^m	Indicate whether the link set C_j^m is active at collection point m
l_i^m	An uplink from ground device i to collection point m
d_i^m	Transmission distance between ground device i and collection point m
r_i^m	Data rate of uplink from ground device i to collection point m
P_i^m	Received power at the collection point m from ground device i
4. Functions	
$\mathcal{P}(d_i^m)$	Path loss of the uplink from ground device i to data collection point m
$\mathcal{N}(\mu, \sigma^2)$	Gaussian random variable
$\delta(C_j^m, i)$	Indicate whether ground device i is in the link set C_j^m at collection point m

Table 3.1: Common nomenclature.

collection points, where $m \in \mathbf{M}$ and $M = |\mathbf{M}|$. These collection points are indexed as $1, 2, \ldots, M$, and assume at each collection point m, each ground device i has one uplink that is denoted as l_i^m. Let L^m be a set that consists of all uplinks at collection point m, where $L^m = \{l_i^m \mid m \in \mathbf{M}, i \in \mathbf{G}\}$. Denote the data rate of uplink l_i^m at collection point m as r_i^m. At each data collection point m, the UAV hovers for one time slot, which can be set to the transmission time of one packet over the lowest data rate.

The path loss of uplinks from ground device i to the UAV at collection point m is denoted as $\mathcal{P}(d_i^m)$ (dBm), where d_i^m is the Euclidean distance from data collection point m to ground device i. The channel condition remains constant within each time slot but varies across multiple time slots. The path loss is calculated as

$$\mathcal{P}(d_i^m) = \mathcal{P}(d_0) + 10\alpha \log_{10} \frac{d_i^m}{d_0} + \mathcal{N}(\mu, \sigma^2), \tag{3.1}$$

where $\mathcal{P}(d_0)$ (in dBm) is the path loss at the reference distance d_0, and α is the path loss exponent. The Gaussian random variable, denoted as $\mathcal{N}(\mu, \sigma^2)$, has mean $\mu = 0$ and variance σ_g^2. All ground devices have a fixed transmit power P (dBm). The received power (in Watt) from ground device i when the UAV is at data collection point m is,

$$P_i^m = 10^{\frac{P - \mathcal{P}(d_i^m)}{10}}. \tag{3.2}$$

The UAV has a SIC radio, which it uses to decode up to L_{max} uplink transmissions [61]. To ensure decoding success, the receive power of each uplink transmission must be sufficiently different. Specifically, the UAV starts its decoding process by first extracting the strongest signal from the received composite signal; from Eq. (3.3), the decoding of a signal is only successful if its SINR is above a given threshold β. The decoded signal is then subtracted from the composite signal. After that, the UAV proceeds to the next stage where it decodes the next transmission with the strongest signal, and so forth. As an example, assume UAV u is receiving from G ground devices simultaneously, where $i \in \mathbf{G}$. Assume the received power at UAV u is in non-decreasing order: $P_1 \leq P_2 \leq \cdots \leq P_G$. The decoding order is thus $G, G-1, \ldots, 2, 1$. That is, the signal with received power P_G can be decoded if and only if all the preceding stronger signals are first decoded and removed [61]. A widely use set of constraints for the aforementioned SIC decoding process is as follows [60]:

$$
\begin{aligned}
&\text{Stage 1} && \frac{P_G}{N_0 + \sum_{i=1}^{G-1} P_i} \geq \beta, \\
&\text{Stage 2} && \frac{P_{G-1}}{N_0 + \sum_{i=1}^{G-2} P_i} \geq \beta, \\
&\quad\vdots && \quad\vdots \\
&\text{Stage (G-q+1)} && \frac{P_{q\varphi}}{N_0 + \sum_{i=1}^{q-1} P_i} \geq \beta.
\end{aligned}
\tag{3.3}
$$

For a given uplink, its SINR and/or Signal-to-Noise Ratio (SNR) must be no less than the threshold value β, which corresponds to a given Modulation and Coding Scheme (MCS) or data rate; see [166] for example values. In Eq. (3.3), N_0 denotes the ambient noise power. For a given SINR or SNR of the uplink l_i^m, Shannon-Hartley formula is used to calculate the asymptotic link capacity r_i^m. That is,

$$r_i^m = log_2(1 + SINR). \tag{3.4}$$

3.2 Problem Definition

The problem at hand is to find the optimal uplinks transmission schedule. In particular, it involves determining the links that are activated at each collection point. To do this, the considered problem exploits the difference in received power from ground devices at data collection points to maximize SIC decoding success.

The following notations are required to formalize the problem. At each collection point, there are multiple link sets. Each link set contains one or more uplinks from ground devices, and critically they satisfy SIC constraints; i.e., Eq. (3.3). This means if a link set is used at a data collection point, ground devices transmit at the data rate corresponding to the SINR threshold β. Each point m is defined to have a collection of link sets; i.e., $\mathbf{C^m}$. Each link set in the collection is denoted as C_j^m, where $j \in \{1, \ldots, |\mathbf{C^m}|\}$, and $C_j^m \subseteq L^m$. The maximum number of concurrent uplinks in each link set C_j^m is set to L_{max}. The value of L_{max} corresponds to the maximum number of signals that can be cancelled by a SIC radio [61]. The sum-rate of link set C_j^m is denoted as R_j^m, and is defined as $R_j^m = \sum_{i \in G} r_i^m$.

As an example, consider Figure 3.1; there is one UAV u and three ground devices g_1, g_2 and g_3. Each data collection point is denoted as $\xi_m, m \in \{1, \ldots, M\}$. Figure 3.1 uses different colors to indicate uplinks at each data collection point ξ_m. Additionally, a different pattern is used to indicate uplinks from different link sets C_j^m. At collection point ξ_1, there are two link sets $C_1^1 = \{l_1^1, l_2^1\}$ and $C_2^1 = \{l_2^1, l_3^1\}$. All three uplinks can transmit concurrently at point ξ_2 and the corresponding link set is $C_1^2 = \{l_1^2, l_2^2, l_3^2\}$. The two link sets at ξ_M are $C_1^M = \{l_1^M, l_2^M\}$ and $C_2^M = \{l_3^M\}$. Given these links sets, the aim is to choose one link set from each collection point that yields the maximum sum-rate over M data collection points. For example, one solution is $\{C_1^1, C_1^2, \ldots, C_2^M\}$, with a corresponding sum-rate of $r_1^1 + r_2^1 + r_1^2 + r_2^2 + r_3^2 + \cdots + r_3^M$.

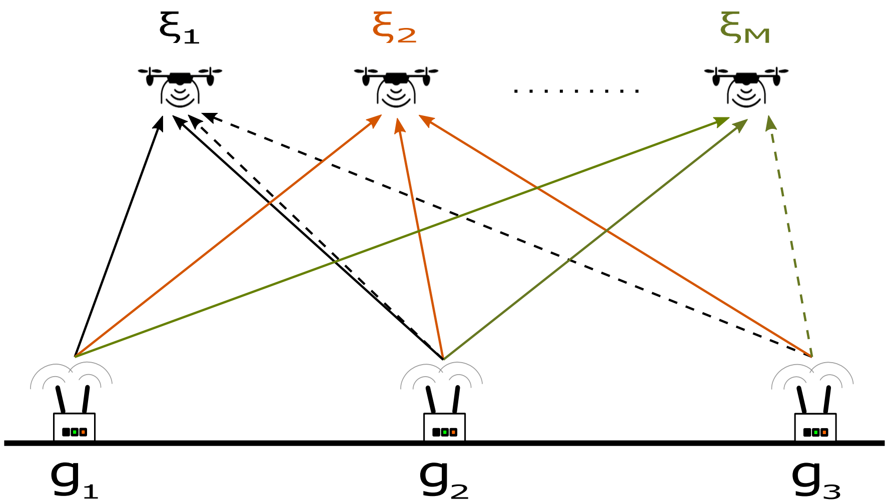

Figure 3.1: Example link sets at M collection points.

Next, this section presents an Integer Linear Program (ILP) to compute a schedule that maximizes the sum-rate over M data collection points, see (3.5). The proposed ILP has one binary decision variable x_j^m that indicates whether the link set

C_j^m is active ($x_j^m = 1$) at data collection point m. The indicator function $\delta(C_j^m, i)$ tracks whether ground device i is in the link set C_j^m ($\delta(C_j^m, i) = 1$) at collection point m. Mathematically, the following ILP aims to maximize the sum-rate of active link sets,

$$\max \quad \sum_{m \in \mathbf{M}} \sum_{j=1}^{|\mathbf{C^m}|} R_j^m x_j^m \tag{3.5a}$$

s.t.

$$\sum_{m \in \mathbf{M}} \sum_{j=1}^{|\mathbf{C^m}|} \delta(C_j^m, i) x_j^m \geq 1 \ \forall i \in \mathbf{G} \tag{3.5b}$$

$$\sum_{j=1}^{|\mathbf{C^m}|} x_j^m = 1 \qquad \forall m \in \mathbf{M} \tag{3.5c}$$

$$x_j^m \in \{0, 1\} \qquad \forall m \in \mathbf{M}, \forall j \in \mathbf{C^m} \tag{3.5d}$$

Constraint (3.5b) ensures each ground device is included in the derived schedule. Otherwise, the resulting schedule may only include ground devices with a high data rate. Constraint (3.5c) ensures one active link set at each data collection point m. Lastly, constraint (3.5d) ensures x_j^m is binary.

This section concludes with two remarks. First, the considered problem with just one SINR threshold and transmit power level can be reduced from the well-known NP-hard weighted set cover problem [167]. In particular, the problem is to find M set covers that maximize the sum-rate (weight) subject to ground devices being included in at least one of these M set covers. This motivates the development of the heuristics outlined in subsequent sections. Second, the formulation in this section is general and it is able to capture more complex setups; namely, ground devices with different SINR threshold values (or data rates) and transmit power levels. Briefly, the collection $\mathbf{C}^m$ at each data collection point m can include link sets for all possible combinations of SINR threshold values, defined as $\beta = \{\beta_1, \beta_2, \ldots, \beta_N\}$, and transmit power levels that is defined as $P = \{P_1, P_2, \ldots, P_M\}$ for each ground device. To generate link sets, for each SINR threshold in β, the formulated ILP can then compute all possible links and transmit power that satisfy the given SINR thresholds. Another extension is to generate link sets whereby each link has a different SINR threshold.

3.3 A Cross-Entropy (CE) Method

This section outlines a centralized CE-based heuristic for a large number of ground devices; interested readers are referred to [165] for more information on CE. Specifically, when applying the centralized CE-based solution, the UAV needs to collect the channel information of ground devices at all M data collection points upfront. Briefly, CE is an adaptive method for estimating probabilities of rare events as well as solving combinatorial optimization problems. It is able to iteratively create collection of solutions and improve the quality of solutions collection over multiple iterations. Moreover, CE is able to provide theoretical guarantees on the performance of the algorithm. The reason is that CE is able to find a solution that frequently yields a high reward in a large sample limit.

The main steps of the CE-based method are as follows: in each iteration, (i) it generates Z random transmission schedules, aka samples, according to a Probability Mass Function (PMF), (ii) it then determines the reward of each sample z_k, where $k = 1, \ldots, Z$. In our case, the reward of each sample z_k corresponds to the throughput of a sample or schedule over M collection points, (iii) with Z rewards in hand, it identifies so called 'elite' samples and records them in a vector called Z^*. To do this, it sorts the reward of Z samples in non-decreasing order. Given a threshold $\gamma \in [0, 1]$, it then identifies the $(1 - \gamma)$-th quantile reward value, which is denoted as φ. Using this reward value, it identifies those samples with a reward value that satisfies $r_k \geq \varphi$ and includes them into Z^*, and (iv) lastly, it uses the statistics of samples in Z^* to improve the said PMF so as to obtain better samples in the next iteration. The previous four steps repeat until the PMF converges.

Define a sample as z_k that has $N = |G| \times |M|$ binary variables x_i^m. Here, CE-based method has $x_i^m = 1$ when the ground device i is active at data collection point m. Let $X^m \in \{0,1\}^{|G|}$ be a binary vector that indicates the link set at collection point m. Hence, a schedule or sample is defined as $z_k = (X^1, X^2, \dots, X^{|M|})$. The sum-rate of each link set X^m is r^m. Therefore, CE-based method has $r_k = (r^1, r^2, \dots, r^{|M|})$. Each sample z_k is characterized by a multivariate Bernoulli distribution $f(z_k; V^c)$, i.e., $z_k \sim Ber(\mathbf{p}^r)$. The real-valued parameter (vector) $V^c \in [0,1]^N$ describes the success/failure probability of each item x_i^m in z_k at iteration c. Initially, at iteration $c = 1$, CE-based method assumes all ground devices have equal probability of being selected or not selected at each collection point; i.e., $V^1 = (0.5, 0.5, \dots)$. It defines the parameter ρ as a smoothing parameter that determines how fast the probabilities in V^c converge. The n-th element in V^c is denoted as V_n^c.

Referring to Algorithm 1, in Line 2-5, CE-based method uses V^1 to generate Z samples, and then proceeds to calculate the reward of each sample using the function $\mathcal{R}()$; see Algorithm 2. Specifically, Algorithm 2 iterates through the link set at each collection point and determine the sum-rate of each sample z_k. It checks whether SIC is successful for all links in set X^m in sample z_k, see Line 5. Assume the received power P_i^m of the G ground devices are in decreasing order; formally, $P_i^m \geq P_{i+1}^m \geq \dots \geq P_G^m$. The decoding order at the UAV u is thus $1, 2, \dots, G-1, G$. If the SINR of ground device i exceeds β, Algorithm 2 then adds its data rate to the sum r^m. In Line 12, Algorithm 2 sums the reward of all link sets and returns the reward r_k of sample z_k.

Referring to Algorithm 1, in Line 6, it sorts the rewards in non-decreasing order; denote the sorted list as $\mathbf{R}$. Then Line 7 uses φ^c as the cut-off reward threshold to identify elite samples; i.e., a value that is in the $(1-\gamma)$-th percentile of $\mathbf{R}$. In Line 8-9, Algorithm 1 update the probabilities in V^c and use the updated PMF to generate Z new samples for the next iteration. The probability of each item n in PMF V^c is computed via

$$V_n^c = \frac{\sum_{k=1}^{Z} \mathbb{1}_{\{r_k \geq \varphi^c\}} \mathbb{1}_{\{z_{kn}=1\}}}{\sum_{k=1}^{Z} \mathbb{1}_{\{r_k \geq \varphi^c\}}} \tag{3.6}$$

Here $\mathbb{1}_a$ is an indicator function that returns a value of one if the condition a is true. Eq. (3.6) counts how many times each item is active among all elite samples/schedules. Specifically, the denominator of Eq. (3.6) is the total number of elite samples. The numerator corresponds to the total number of times that the n-th item occurs in the elite samples. Note that instead of updating the PMF directly, CE-based method uses a smoothed version that considers the influence of past values V^{c-1}, see Line 9. This allows the CE-method to explore more samples before converging onto the best schedule that frequently yields a high reward given different channel conditions. Lastly, Algorithm 1 concludes that CE-based method has converged when the probability V^c of selecting a ground device at each data collection point is within a certain tolerance θ away from one or zero.

Algorithm 1: CE method based link scheduler.

Initialize: $V^1 = [0.5, \ldots, 0.5], c = 1, \gamma, \rho$
1 **while** *not Converge(V^c)* **do**
2   **for** $k \leftarrow 1$ *to* Z **do**
3         $z_k = Z(V^c)$;
4         $r_k = \mathcal{R}(z_k)$;
5   **end**
6     $\mathbf{R} = \text{Sort } (r_1, \ldots, r_Z)$;
7     $\varphi^c = \text{Percentile}((1-\gamma), \mathbf{R})$;
8   **for** $n \leftarrow 1$ *to* $|V^c|$ **do**
9         $V_n^c = \rho V_n^c + (1-\rho) V_n^{c-1}$;
10   **end**
11     $c \leftarrow c + 1$;
12 **end**

3.4 Heuristic Algorithm: GCTS

Note that the formulated ILP requires the nominal channel gain information to ground devices. Moreover, the formulated ILP is not suitable for large-scale networks. In contrast, Greedily Construct Transmission Schedule (GCTS) is more efficient and easier to realize, as well as yielding a near-optimal solution as the formulated ILP. Its basic idea is to greedily include ground devices into the transmission

Algorithm 2: The sum-rate of a sample.

input : z_k
output: r_k
for $m \leftarrow 1$ *to* M **do**
  $r^m = 0$
  for $i \leftarrow 1$ *to* G **do**
    for $g \leftarrow i + 1$ *to* G **do**
      if $\frac{P_i^m}{N_0 + \sum_g^G P_g^m} \geq \beta$ **then**
        $r^m = r^m + r_i^m$
      else
        break
      end
    end
  end
  $r_k = \sum_m^M r^m$
end

schedule at each data collection point. In particular, the UAV only needs to collect channel gain information of ground devices upon arrival at each collection point. This can be achieved by sending a beacon. After that, a dedicated mini-slot can be assigned to each ground device where it transmits its channel coefficient to the UAV. GCTS classifies ground devices into *high priority* and *low priority* according to whether they have been scheduled in a past data collection point. Ground devices that have the least opportunity to activate are classified as *high priority* and they are included into the group S. Otherwise, they are classified as *low priority* and placed in group $\hat{S}$. At each collection point m, GCTS will first greedily include a *high priority* ground device from the group S into the transmission set C^m. Once it has considered all devices in group S, it will add *low priority* ground devices from the group $\hat{S}$ into the transmission set C^m to increase the sum rate of the transmission set under construction. The following subsections explain the general structure of GCTS, transmission set construction followed by its run-time complexity.

3.4.1 General Structure of GCTS

Algorithm 3 shows an overview of GCTS. It first initializes the transmission schedule $\mathbb{S}$ to an empty set. Then it generates two groups S and $\hat{S}$ that consist of *high priority* and *low priority* ground devices, respectively. Initially, group S consists of all ground devices and group $\hat{S}$ is empty. At each data collection point m, GCTS calls function *HighPriority()* to iterate through ground devices in the group S to construct a transmission set $\bar{C}^m$, see Line 2. After that, in Line 3, function *LowPriority()* is used to greedily add *low priority* ground devices from group $\hat{S}$ into the constructed set $\bar{C}^m$. Once the sum-rate stops increasing, GCTS returns the constructed transmission set of collection point m, aka C^m. GCTS will then include the constructed transmission set C^m into the transmission schedule $\mathbb{S}$, see Line 4. After that, in Line 5, GCTS removes ground devices in C^m from group S and adds them into the group $\hat{S}$. After that, GCTS checks whether group S is empty. If it does, GCTS will empty group $\hat{S}$ and add all ground devices into group S, see Line 7. GCTS will return the transmission schedule $\mathbb{S}$, where $\mathbb{S} = C^1, C^2, \ldots, C^M$.

Algorithm 3: GCTS general structure.

Initialize: $\mathbb{S} = \emptyset, S = \mathbf{G}, \hat{S} = \emptyset$
1 **for** $m \leftarrow 1$ *to* M **do**
2     $\bar{C}^m = \mathit{HighPriority}(S)$. // see Algorithm 4 ;
3     $C^m = \bar{C}^m \cup \mathit{LowPriority}(\hat{S})$. // see Algorithm 5 ;
4     $\mathbb{S} = \mathbb{S} \cup C^m$;
5     $S = S \setminus C^m$; $\hat{S} = \hat{S} \cup C^m$;
6   **if** $S = \emptyset$ **then**
7         $S = \mathbf{G}$, $\hat{S} = \emptyset$
8   **end**
9 **end**
10 Return $\mathbb{S}$

3.4.2 Transmission Set Construction

The details of constructing a transmission set C^m is presented in Algorithm 4 and Algorithm 5. GCTS first calls Algorithm 4 to greedily add high priority ground devices in $|S|$ to construct a transmission set $\bar{C}^m$, where $i = 1, \dots, |S|$. GCTS calls *SIC()* to check whether the transmission set satisfies SIC constraints. Function *SumRate()* is used to calculate the sum-rate of a constructed transmission set. Specifically, the data rate of each ground device is calculated according to their individual SINR and/or SNR value. After iterating through all ground devices in the group S, GCTS calls Algorithm 5 to add one or more ground devices from group $\hat{S}$ into the constructed transmission set $\bar{C}^m$ to increase the sum rate of the transmission set under construction, where $j = 1, \dots, |\hat{S}|$.

Referring to Algorithm 4, GCTS includes *high priority* ground devices from group S to form transmission set $\bar{C}^m$. In Line 1, GCTS initializes the transmission set $\bar{C}^m$ to an empty set and sum-rate $\bar{R}^m$ to zero. GCTS then uses *Sort()* to sort ground devices in the group S in descending order according to their received power at the data collection point m, see Line 2. After that, in Line 4, GCTS greedily includes one ground device i into the transmission set $\bar{C}^m$. With a newly added ground device i, GCTS calls *SIC()* to check whether the transmission set C_i^m satisfies SIC constraints, see Line 5. If it does, in Line 6, GCTS calls *SumRate()* to calculate the sum-rate $\bar{R}^m$ of $\bar{C}^m$. Otherwise, GCTS will remove the newly added ground device i from the transmission set $\bar{C}^m$ and set the corresponding sum-rate $\bar{R}^m$ to zero, see Line 8. After iterating through all $|S|$ ground devices in the group S, GCTS will return the constructed transmission set $\bar{C}^m$ and the sum-rate $\bar{R}^m$.

Referring to Algorithm 5, GCTS greedily adds one *low priority* ground device j from group $\hat{S}$ into the constructed transmission set $\bar{C}^m$ to construct a new set C^m. In Line 1, GCTS initializes C^m as $\bar{C}^m$. The corresponding sum-rate R^m is equal to $\bar{R}^m$, initially. GCTS also calls *Sort()* to sort ground devices in the group $\hat{S}$ in descending order according to their received power, see Line 2. GCTS then greedily includes one ground device j into the transmission set C^m, see Line 4. With a newly added ground device j, GCTS calls *SIC()* and *SumRate()* to check whether SIC is successful and then calculate the sum-rate of transmission set C^m, see Line 5-9. GCTS will greedily include ground device until the sum-rate stops increasing, see

Algorithm 4: Include ground devices from group S.

input : S
output: $(\bar{C}^m, \bar{R}^m)$
1 $\bar{C}^m = \emptyset$, $\bar{R}^m = 0$
2 $S = Sort(S)$
3 **for** $i \leftarrow 1$ *to* $|S|$ **do**
4     $\bar{C}^m = \bar{C}^m \cup i$
5   **if** *SIC*($\bar{C}^m$) $\leftarrow$ *True* **then**
6         $\bar{R}^m = SumRate(\bar{C}^m)$
7   **else**
8         $\bar{C}^m = \bar{C}^m \setminus i$; $\bar{R}^m = 0$
9   **end**
10 **end**
11 Return $(\bar{C}^m, \bar{R}^m)$

Line 10-12. GCTS then returns the transmission set C^m and sum-rate R^m.

Algorithm 5: Include ground devices from group $\hat{S}$.

input : $\hat{S}, \bar{C}^m, \bar{R}^m$
output: C^m, R^m
1 $C^m = \bar{C}^m$, $R^m = \bar{R}^m$
2 $\hat{S} = Sort(\hat{S})$
3 **for** $j \leftarrow 1$ *to* $|\hat{S}|$ **do**
4     $C^m = C^m \cup j$
5   **if** *SIC*(C^m) $\leftarrow$ *True* **then**
6         $R^m = SumRate(C^m)$
7   **else**
8         $C^m = C^m \setminus j$; $R^m = 0$
9   **end**
10   **if** R^m *stops increasing* **then**
11   break
12   **end**
13 **end**
14 Return C^m, R^m

This section concludes with the run time complexity of GCTS. For each data collection point m, GCTS needs to construct a transmission set C^m. Hence, Line 1-9 run for $|M|$ times when constructing transmission sets. Therefore, $|C^m|$ is bounded by $\mathcal{O}(|M|)$. For Line 2-3, regardless of whether we are including ground devices from group S or group $\hat{S}$, GCTS has to check no more than $|G|^2$ ground devices. Consequently, the time complexity of GCTS or Algorithm 3 is $\mathcal{O}(|M||G|^2)$.

3.5 Protocol Design: CPSP

This section proposes a novel, iterative, distributed MAC called Collection Point Selection Protocol (CPSP). The basic idea of CPSP is depicted in Figure 3.2; the left and right branch correspond to the process at the UAV and ground devices, respectively. When the UAV is at a data collection point, it first sends a beacon to all ground devices to ascertain their channel condition. Ground devices maintain an individual probability distribution over all collection points, which they then use to select the best collection point. Specifically, a ground device determines the transmission probability of each collection point. During the learning process, the UAV uses the SINR or data rate of each transmission to calculate a reward, which it then sends to ground devices. The reward is then used by ground devices to update their probability distribution. In particular, each ground device considers past channel conditions when updating probability distribution. Briefly, CPSP has two main advantages: (i) each ground device is able to determine *independently* by itself when it should transmit to a UAV in order to obtain the highest transmission success. This means there is no need to collect topological and channel gain information, and send them to a central server, and (ii) link sets can be updated dynamically whenever there is a change in ground devices.

To make specific the learning process of each ground device, consider Figure 3.3. It shows the steps taken by a ground device i, where $i \in \mathbf{G}$, to learn the best collection point over T learning slots. Each ground device i maintains a PMF over M collection points. The PMF at time t is denoted as α_i^t, where $t = \{1, 2, \ldots, T\}$. The m-th element in α_i^t is denoted as $\alpha_i^t(m)$. Specifically, the real-valued parameter $\alpha_i^t(m) \in [0, 1]^M$ describes the probability that ground device i selects collection point m in learning frame t, where $m \in M$. All ground devices select their individual collection point according to the constructed PMF. The PMF of all ground devices is initialized to the uniform distribution; i.e., for ground device i, we have $\alpha_i^1 = (1/M, 1/M, \ldots)$.

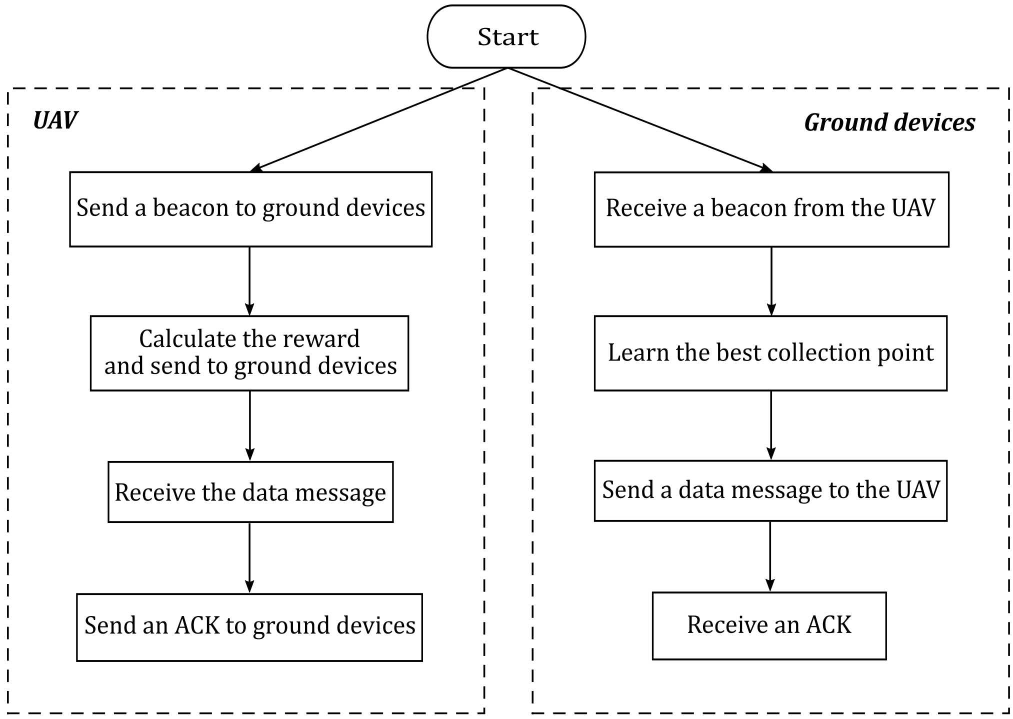

Figure 3.2: The process of CPSP.

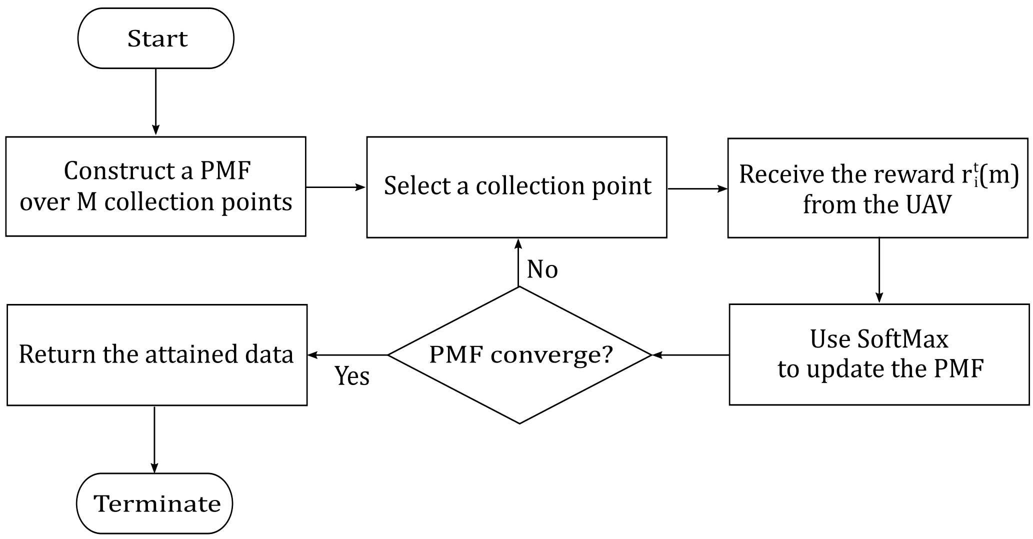

Figure 3.3: Learning process of a ground device using CPSP.

The reward obtained by ground device i for collection point m in learning slot t is denoted as $r_i^t(m)$, which is defined as

$$r_i^t(m) = \begin{cases} log_2(1 + SINR), & SINR \geq \beta \\ 0, & \text{Otherwise.} \end{cases} \tag{3.7}$$

In words, if a transmission at a collection point m is successful, the ground device receives a reward that corresponds to its SINR and/or SNR value; otherwise, it is set to zero. CPSP initializes the reward of all collection points to zero.

CPSP use an Exponential Weighted Average (EWA) to calculate the reward in the current learning slot t with respect to previous slots. That is,

$$r_i^t(m) = \rho r_i^t(m) + (1 - \rho) r_i^{t-1}(m), \tag{3.8}$$

where ρ is a smoothing parameter.

Given the reward of a collection point m, each ground device then updates its PMF over all collection points. Specifically, ground device i uses the following SoftMax function to convert the reward $r_i^t(m)$ to a probability value $\alpha_i^t(m)$ that determines the likelihood of transmitting when the UAV is at collection point m,

$$\alpha_i^t(m) = \frac{e^{r_i^t(m)/\tau}}{\sum_{m=1}^{M} e^{r_i^t(m)/\tau}}, \tag{3.9}$$

where τ is the temperature parameter that dictates how often ground devices explore different data collection points. We use EWA to update the PMF of ground device i as well. Specifically,

$$\alpha_i^t = \rho \alpha_i^t + (1 - \rho) \alpha_i^{t-1}. \tag{3.10}$$

A PMF has converged when the probability α_i^t of selecting a collection point is within a certain tolerance θ away from one or zero.

3.6 Evaluation

The proposed solutions are evaluated in Matlab [168]. There are up to ten ground devices. In particular, the formulated ILP and heuristic methods, namely the CE-based method and GCTS algorithm, are run on small problem instances. This allows ground devices to generate all possible link sets for each data collection point. More importantly, it acts as a benchmark for the proposed heuristic methods to compare against the optimal result over the same network setup. The UAV is assumed to have a known trajectory and the location of data collection points is given and fixed; this is reasonable as an operator knows where ground devices are located. SINR threshold and data rate mappings are from Cisco [166]. A transmission is successful if its SINR and/or SNR exceeds $\beta = 5$ (dB). The simulation settings are listed in Table 3.2 [37] [66]. The presented results include those from solving the formulated ILP, labeled as SIC-ILP, and four other methods: (i) CE, (ii) GCTS heuristic algorithm, (iii) CPSP, (iv)Slotted Aloha, and (v) TDMA. Additionally, it studies CE method when it has either a fixed or an adaptive cut-off reward threshold φ^c, which are labeled as CEF-φ^c and CEA-φ^c, respectively. Moreover, the evaluation considers two reward cases for CPSP: (i) normal reward $r_i^t(m)$, or (ii) amplified SIC reward $\omega \times r_i^t(m)$. Specifically, for transmissions that satisfy SIC, the reward is amplified by multiplying it with a factor ω. These two reward cases are labeled as CPSP-r and CPSP-$r\omega$, respectively.

1. Basic system settings				
Symbol	u	G	M	L_{max}
Value	1	10	10	4
2. UAV and ground devices deployment				
Symbol	d	h	s	P
Value	300 m	100 m	26 m/s	1 W
3. SNR and SINR calculation				
Symbol	α	β	N_0	σ^2
Value	2.7	5 dB	-90 dBm	2 dB^2

Table 3.2: Simulation settings for the considered link scheduling problem.

3.6.1 CE with a Fixed Cut-Off Reward Threshold

Firstly, the following experiments apply the CE-based method and study the impact of different parameters settings on the average throughput, the number of CE iterations, and throughput fairness of ground devices. Studying these parameters is significant because they determine whether CE is able to find as well as time taken to find the best result. Note that as CEA-φ^c yields the same trend as CEF-φ^c, This section thus only plots the results of CEF-φ^c. The cut-off reward parameter γ that identifies 'elite' samples is fixed at 0.95. The tolerance bound θ for convergence is 10^{-2}.

3.6.1.1 Smoothing Parameter

To investigate the impact of the smoothing parameter ρ values, its value is increased from 0.1 to 1. In addition, the evaluation considers 100, 300 and 500 samples. The number of ground devices G and data collection points M is fixed at six and five, respectively.

Figure 3.4 shows the average throughput with different number of generated samples. We see that with increasing ρ values, the average throughput of all cases with 100, 300 and 500 samples gradually becomes smaller. Specifically, the decrease in average throughput is 0.7 Mbps, 1.0 Mbps and 1.4 Mbps, respectively. Moreover, when the value of ρ is smaller than 0.8, increasing ρ by 0.1 causes the average throughput to drop by about 0.1 to 0.2 Mbps. When the value of ρ is in the range $[0.8, 1]$, the average throughput reduces by 0.2 to 0.4 Mbps, which is twice that for smaller ρ values. The reason is that with a higher smoothing parameter, the probability is affected more significantly by the current reward, and may cause CE to converge onto a local optima solution. Additionally, we also observe that with more samples, the average throughput will be higher. Specifically, the average throughput will be 0.2 Mbps higher with 200 more samples. This is because a larger number of samples means CE has a higher chance of finding better samples that have a higher reward.

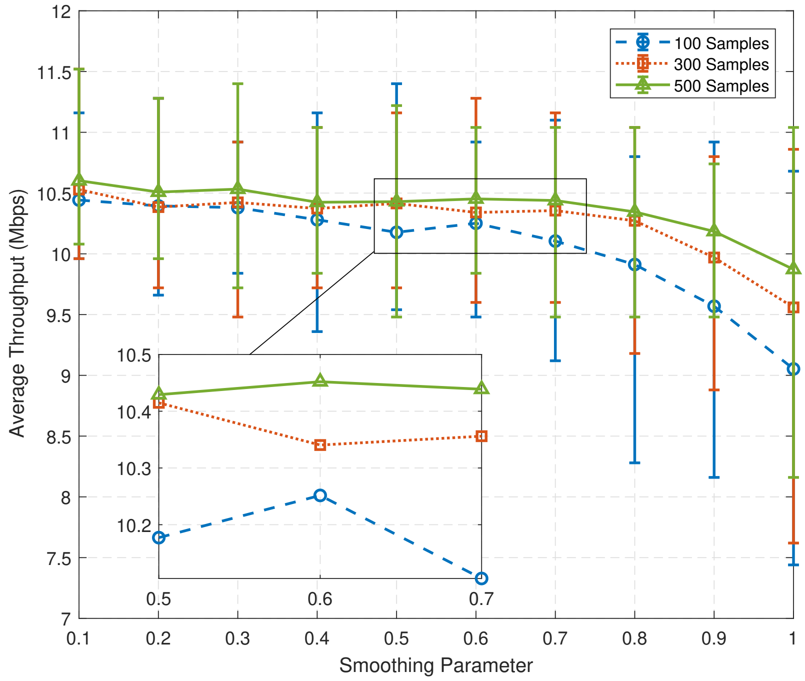

Figure 3.4: Average throughput with different smoothing parameters.

Figure 3.5 shows the average number of CE iterations with increasing ρ values. For all cases with different number of samples, the average number of CE iterations decreases with a higher smoothing parameter. The reason is because the current probability value has more influence as compared to past values. Therefore, CE updates the probability of each ground device quicker, which leads to a faster convergence time. From Figure 3.5, we can see that for the case with 500 samples, the average number of iterations before CE converges decreases from 2000 to 80. Moreover, when the smoothing parameter changes from 0.1 to 0.5, the average number of iterations is half that of smaller ρ values. In the range $[0.6, 1]$, the number of iterations reduces by 50 with ρ increasing by 0.1. Furthermore, we see that more samples lead to a higher average number of CE iterations. Figure 3.5 shows that the 500 samples case results in CE running for an additional 20 iterations as compared to the 300 samples case. Compared to the case with 300 samples, the 100 samples case requires 200 fewer iterations on average. The reason is that with more samples, CE takes longer to test all samples and find the corresponding reward. Therefore, the update process is relatively slower, and the PMF of CE takes longer to converge.

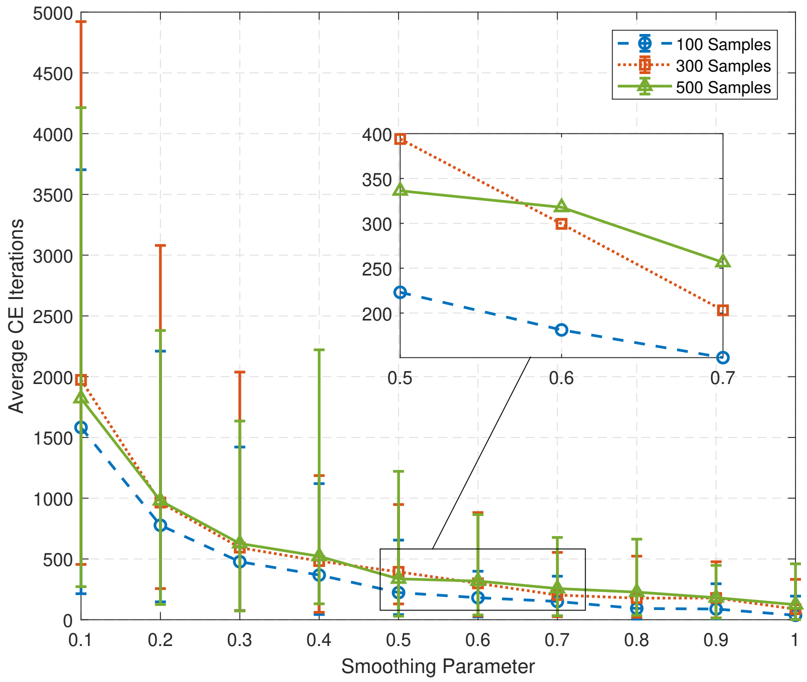

Figure 3.5: Average number of CE iterations with different smoothing parameters.

The evaluation also investigates the range of smoothing parameter ρ that balances the average throughput and the average number of CE iterations. The average throughput over the number of CE iterations is used to quantify the efficiency of each ρ value. Figure 3.6(a) is a three-dimensional (3D) line plot with different smoothing parameter ρ values. The other three subplots are the projection of Figure 3.6(a) in the x, y and z plane, respectively. We see the trend of each line in Figure 3.6(b) and Figure 3.6(c) is the same as the 500 samples case shown in Figure 3.4 and Figure 3.5. This experiment focuses on studying the relationship between the average throughput and the average number of CE iterations, see Figure 3.6(d).

As shown in Figure 3.6(d), we observe that when the average throughput changes from 10.42 Mbps to 10.6 Mbps, the average number of CE iterations has a sharp rise from 350 to 1800. In other words, CE needs to use 1450 more iterations to obtain an increase of 0.18 Mbps. From Figure 3.6(c), we observe that when ρ equals 0.1 and 0.5, the corresponding number of CE iterations is 1800 and 350, respectively. We then calculate the efficiency for ρ equals 0.1 and 0.5, that is 5.9×10^{-3} and 3.0×10^{-2}. Specifically, the efficiency of the case when ρ equals 0.5 is five times that when ρ equals 0.1. Moreover, we observe that when ρ is in the range $[0.5, 0.7]$, the individual efficiency is similar and in the range of $[3.0, 4.0] \times 10^{-2}$. We therefore conclude that the efficiency is the highest when the smoothing parameter ρ is within the range of $[0.5, 0.7]$. Thus, in all subsequent experiments, we will use a smoothing parameter ρ drawn from the said range.

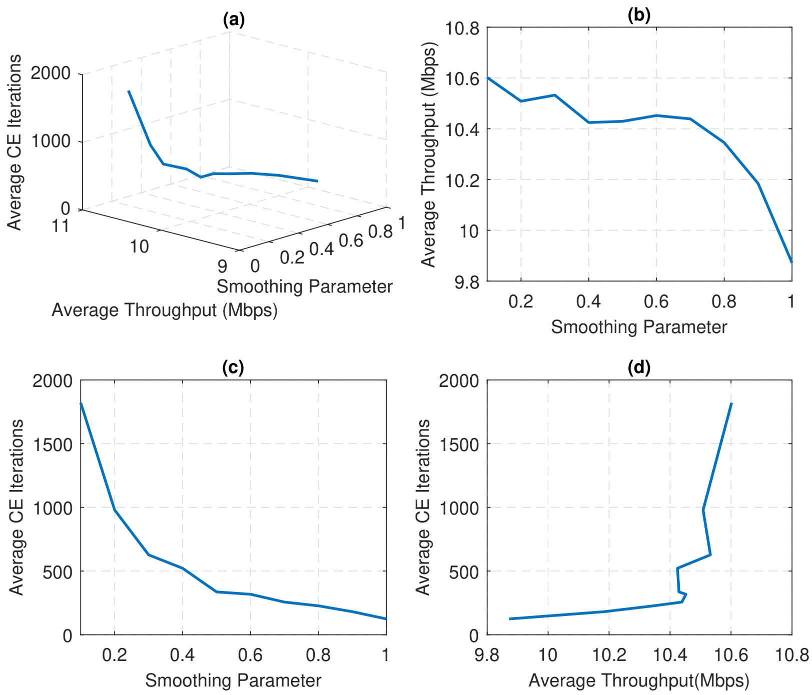

Figure 3.6: Relationship between the average throughput and the average number of CE iterations of 500 samples case. (a) A 3D-line plot with different smoothing parameters (b) Average throughput versus smoothing parameters (c) Average number of CE iterations versus smoothing parameters (d) Average number of CE iterations versus average throughput.

3.6.1.2 Fairness

Next, the following experiment studies how different number of collection points and number of ground devices affect fairness; see Figure 3.7. Jain's Fairness index

(JFi) [169] is applied to measure whether each ground device has equal opportunity to communicate with the UAV. Specifically, JFi quantifies whether ground devices have transmitted an equal amount of data.

Figure 3.7 shows the change in JFi for four different number of ground devices cases when we increase the number of data collection points. Referring to Figure 3.7, the JFi of one ground device is a constant value at one because it is able to transmit the same amount of data at each collection point. With more ground devices, JFi reduces. For example, when there are three collection points, the JFi of three, six and ten ground devices case is 0.79, 0.68 and 0.5, respectively. The reason is that when there are a large number of ground devices, and given that there is a limit on the number of concurrent transmissions, only some ground devices can be activated simultaneously. Moreover, as SIC requires a difference in received power, the data rate of ground devices will be different, which causes JFi to be smaller.

Referring to Figure 3.7, we also observe that JFi increases when the number of data collection points is smaller than or equal to the number of ground devices. The reason is that under this circumstance, the position of data collection points is within the coverage of ground devices. Thus, with a new collection point, the data transmission opportunity of each ground device is more likely to be equal. Therefore, the corresponding uploaded data of each ground device is similar. Consequently, JFi increases when there are more data collection points. However, when the number of data collection points is at least one more than the number of ground devices, part of the UAV data collection points will not be within the range of ground devices. Moreover, SIC is preferable when the received power levels are different. Therefore, ground devices that are the closest to the UAV and located far away from the UAV have a higher opportunity to transmit concurrently. Consequently, the amount of transmitted data from each ground is unlikely to be equal; hence, the value of JFi drops.

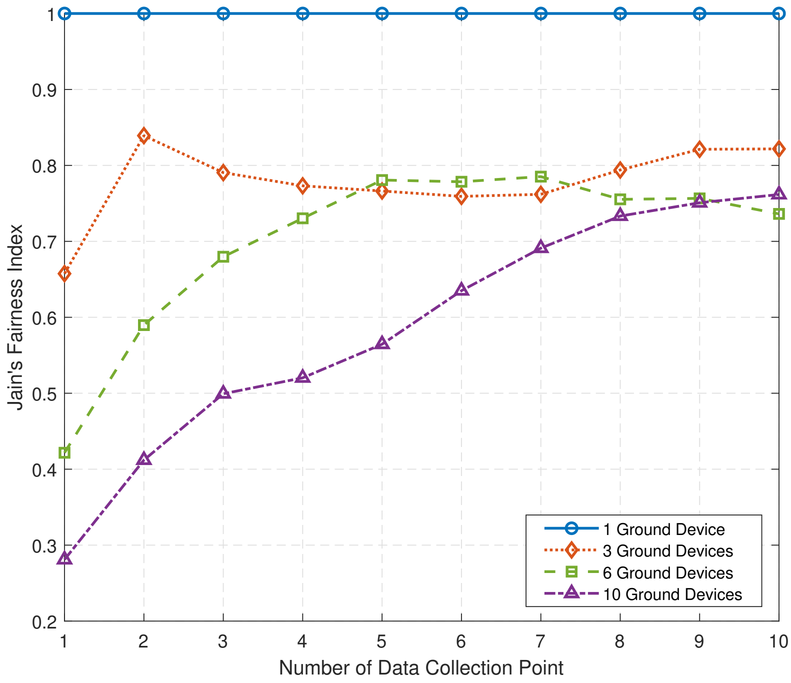

Figure 3.7: Jain's Fairness index with different number of data collection points.

3.6.2 GCTS Performance

The experiments in this section study the performance of GCTS. It runs each simulation 50 times and plot the average results. In particular, the following experiments study different number of ground devices and data collection points as well as the speed and height of the UAV. Besides, different distances between ground devices are also studied.

3.6.2.1 Number of Ground Devices and Data Collection Points

This section considers how the number of data collection points M affects the average throughput. The following G values are considered: 1, 3, 6 and 10. The number of data collection points M ranges from one to ten.

Figure 3.8 shows the average throughput with increasing number of collection points. We see that when there is only one ground device, the average throughput remains a constant at 5.4 Mbps for any number of collection points. This is because at each collection point, SIC is not used to decode signal because there is only one active ground device that transmits with the highest data rate of 54 Mbps. Thus, the average throughput remains a constant at 5.4 Mbps. However, for the case with three, six and ten ground devices, the average throughput decreases with increasing number of data collection points. For example, when there is only one collection point, the average throughput of three cases is 10.1 Mbps, 12.1 Mbps and 13.67 Mbps, respectively. However, when there are ten data collection points, the average throughput becomes 6.8 Mbps, 8.83 Mbps and 10.96 Mbps, respectively. This is because of SIC's decoding limit that restricts the number of simultaneous active uplinks. Additionally, the data rate of each uplink is determined by its individual SNR and/or SINR value. Therefore, when the number of collection points increases, the total transmitted data does not increase proportionally. Moreover, from Figure 3.8, we see that when there are multiple ground devices, the average throughput is twice that of the case with one ground device because SIC allows multiple simultaneous transmissions. However, if there is only one ground device, SIC does not apply. Therefore, the average throughput increases with increasing number of ground devices.

3.6.2.2 Deployment of the UAV and Ground Devices

This section presents a study of how the deployment of the UAV and ground devices affect the average throughput. Specifically, it considers different speed and altitude of the UAV as well as the distance between two adjacent ground devices. Note that the transmit distance from ground devices to the UAV changes with the varying speed and altitude of the UAV as well as the spacing between ground devices. The received power difference between uplinks will change correspondingly and thus impact the success of SIC decoding. Figure 3.9 shows the average throughput with increasing UAV altitude h. The fixed altitude of the UAV is increased from 50 m to 250 m. Additionally, we consider the following UAV speed s: 13 m/s, 26 m/s and 65 m/s. The distance between two adjacent ground devices is fixed at 300 m. From Figure 3.5, we see that the average throughput of all three cases with different UAV

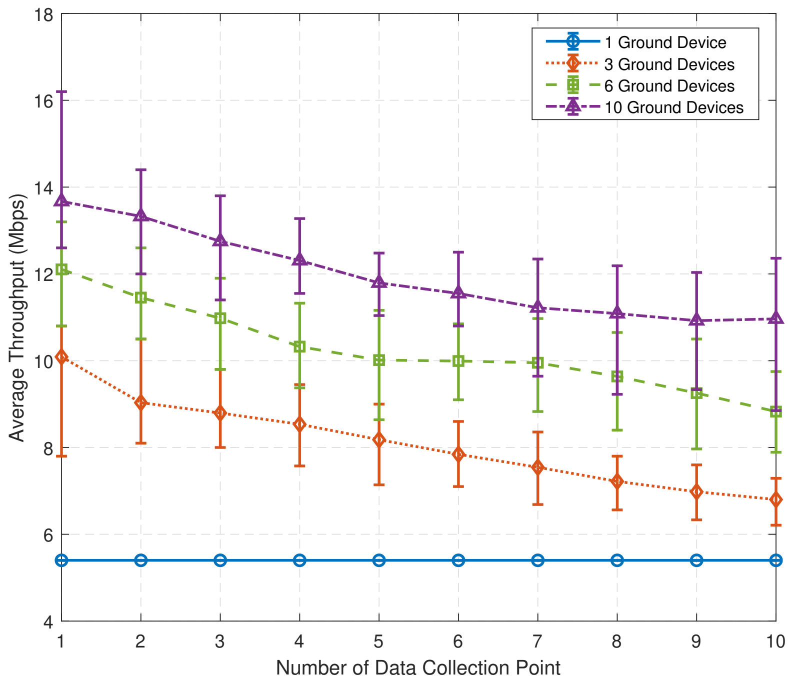

Figure 3.8: Average throughput with different number of data collection points.

speed decreases with increasing UAV altitude. Specifically, the average throughput decreases 3.0 Mbps, 2.85 Mbps and 1.8 Mbps, respectively. This is because the path loss of uplinks becomes larger at a high-altitude, which causes the received power from ground devices to be small at the UAV. Therefore, the number of simultaneous links that satisfy SIC decreases. Consequently, the average throughput gradually drops when the UAV flies at a higher altitude. Moreover, when the UAV speed s increases, the average throughput becomes smaller. Specifically, when the UAV speed is 26 m/s and 65 m/s, the average throughput is around 0.2 Mbps and 1.5 Mbps smaller than that of the case with the speed of 13 m/s. This is because when the UAV flies at a high speed, the distance flown over one slot will be further as compared to when it flies at a low speed. Thus, the transmission distance between the UAV and each ground device will be longer after each time slot, which results in a smaller channel gain. Therefore, the difference in received power is less likely to satisfy SIC or ground devices have to transmit with a lower data rate to ensure SIC is viable. Consequently, the corresponding sum-rate decreases when the UAV flies at a higher speed.

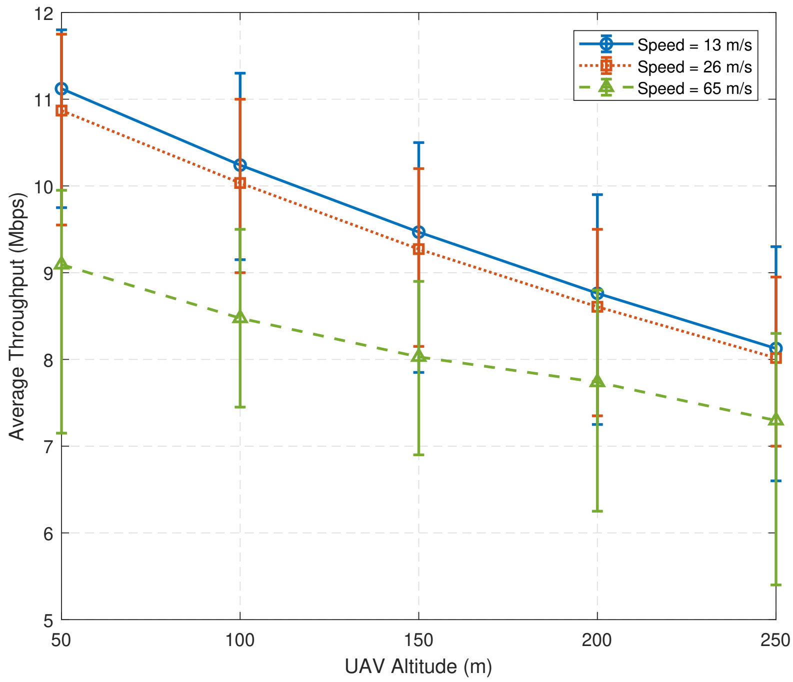

Figure 3.9: Average throughput with different altitudes of the UAV.

Figure 3.10 shows the average throughput with increasing distance d between adjacent ground devices. Specifically, the distance d increases from 100 m to 500 m. The altitude h of the UAV is set to 100 m. We can see that the average throughput is higher when the distance d is bigger because it makes the received power level difference of concurrent uplinks that satisfy SIC becomes bigger; thus, the corresponding SINR and/or SNR indicates a higher data rate. Consequently, the average throughput increases.

3.6.3 CPSP Performance

The experiments to follow investigate the impact of different parameter settings on the average throughput, PMFs convergence rate and SIC transmissions when ground stations use CPSP. For CPSP-r, these experiments change the value of the temperature parameter τ of the SoftMax function and the smoothing parameter ρ of EWA. Additionally, for CPSP-$r\omega$, these experiments study different reward

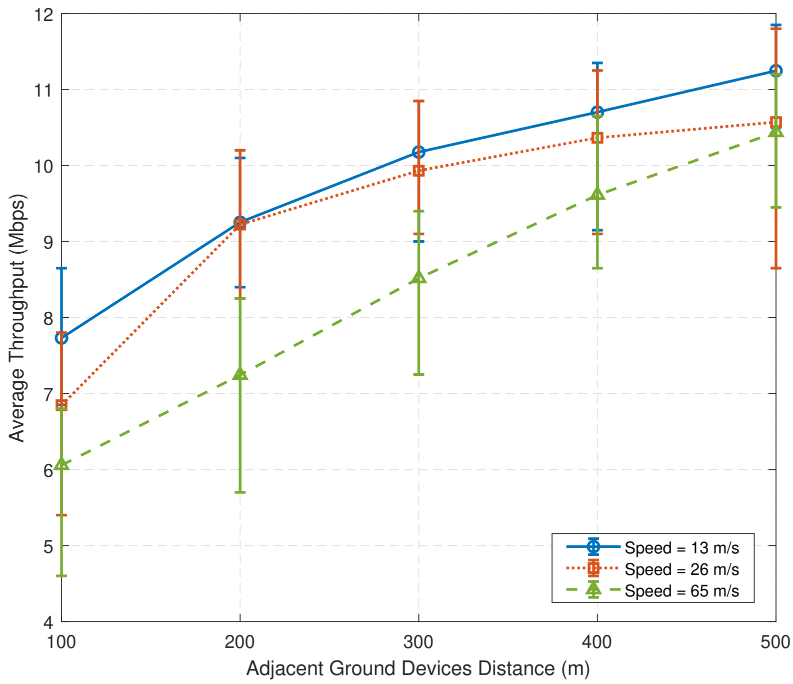

Figure 3.10: Impact of distance between adjacent ground devices.

amplification factor ω. CPSP-r and CPSP-$r\omega$ are trained over a period of 101,000 learning slots. Then every 1000 learning slots are regarded as a learning frame and plot the average throughput in each learning frame. These experiments consider three ground devices and three data collection points. The tolerance bound θ for use to detect convergence is 10^{-4}.

3.6.3.1 CPSP with Normal Reward

The following experiments investigate how different temperature parameter τ and EWA smoothing parameter ρ affect the average throughput and the PMF convergence rate of CPSP-r. Figure 3.11 shows the PMF convergence rate of CPSP-r when the temperature parameter τ is either fixed or adaptive. When τ is fixed, the experiments consider two different values, namely 15 and 105. For the case with adaptive τ, τ is assumed to decrease linearly after each learning frame. Specifically, it starts from 105 in the first frame and reaches a value of five in the last frame. The results are an average of five simulation runs. The EWA smoothing parameter

ρ is set to 0.1.

Referring to Figure 3.11, we see that when τ is fixed, the average throughput fluctuates around a certain value. For example, when τ is set to 15 and 105, the average throughput fluctuates around 5.0 Mbps and 4.0 Mbps, respectively. This is because ground devices are less likely to explore different data collection points; thus, the PMF will converge onto the local optimal solution. However, when ground devices adapt their τ value, the average throughput first fluctuates around 4.0 Mbps and then starts to increase in the 85-th frame. It finally converges at 5.4 Mbps. This is because ground devices spend time exploring for the best reward in earlier learning frames before converging onto the best solution. In all subsequent experiments, an adaptive temperature parameter τ is used, which decreases linearly from 105 to five.

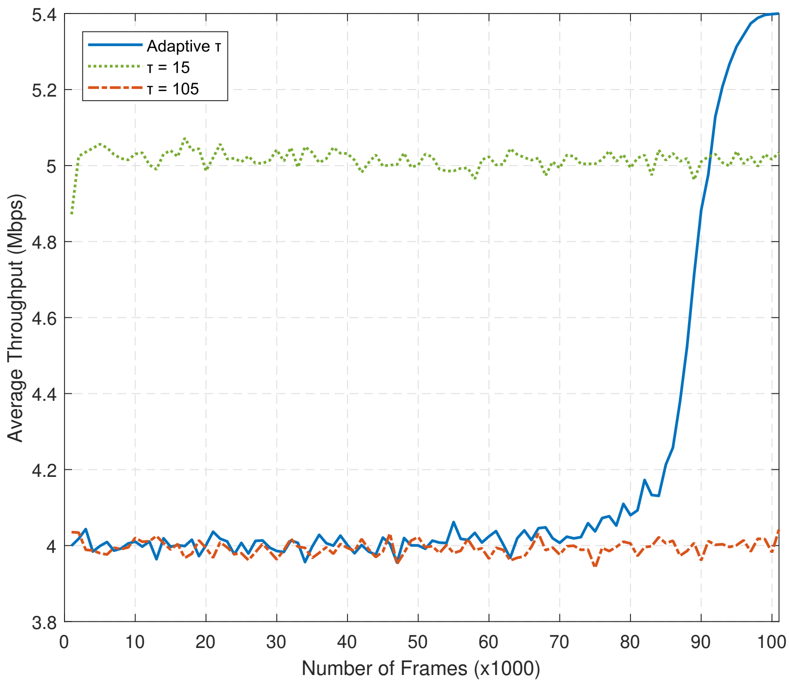

Figure 3.11: Average throughput with different temperature parameters.

This experiment studies the average throughput of CPSP-r with different EWA smoothing parameter ρ. Five different smoothing parameter values are considered: 0.1, 0.3, 0.5, 0.7 and 1.0. The temperature parameter τ of the SoftMax function adaptively decreases from 105 to five. The average result for 50 simulation runs are plotted in this experiment. From Figure 3.12, observe that between the first and the 50-th learning frame, all smoothing parameter values yield a similar average throughput that fluctuates around 4.0 Mbps. The reason is because ground devices are exploring all data collection points in earlier learning frames. When the frames are in the range of $[50, 89]$, for the ρ value between $[0.1, 0.7]$, the average throughput is 0.05 Mbps higher with ρ increasing by 0.2. Additionally, when ρ equals 1.0, the average throughput is 1.05 times more than the case when ρ equals 0.1. This is because a bigger ρ value yields a higher PMF converge rate that leads to a solution with higher transmission rate in each learning slot. Thus, the average throughput increases with increasing ρ value.

As shown in Figure 3.12, after the 89-th learning frame, the average throughput of smaller ρ values becomes higher. Specifically, compared to the case when ρ equals 0.1, the average throughput is 0.96 times smaller when ρ equals 1.0. This is because with a smaller smoothing parameter, ground devices are able to quickly smooth out the influence that caused by selecting a collection point with small reward. On the contrary, a higher smoothing parameter indicates that ground devices need to spend time to explore a better solution again. Thus, a smaller smoothing parameter yields a higher average throughput.

Figure 3.13 shows the change in throughput for learning slots in the range $[94950, 95050]$. The temperature parameter τ changes from 11 to 10 during these slots. The conducted experiments study three different ρ values: 0.1, 0.5 and 1.0. Only one simulation run is plotted. From Figure 3.13, we see that for $\rho = 0.1$, the throughput decreases to 4.4 Mbps and 4.8 Mbps in the 12-th and 57-th learning slot, respectively. Then in the next slot, the throughput will immediately return back to 5.4 Mbps. However, for the case with $\rho = 0.5$, throughput decreases to 1.8 Mbps in the 75-th and 77-th learning slot. Moreover, the throughput for $\rho = 1.0$ case has three fluctuations and each of them takes 6, 19 and 11 learning slots to return to 5.4 Mbps. Figure 3.13 confirms that with a higher smoothing parameter, ground

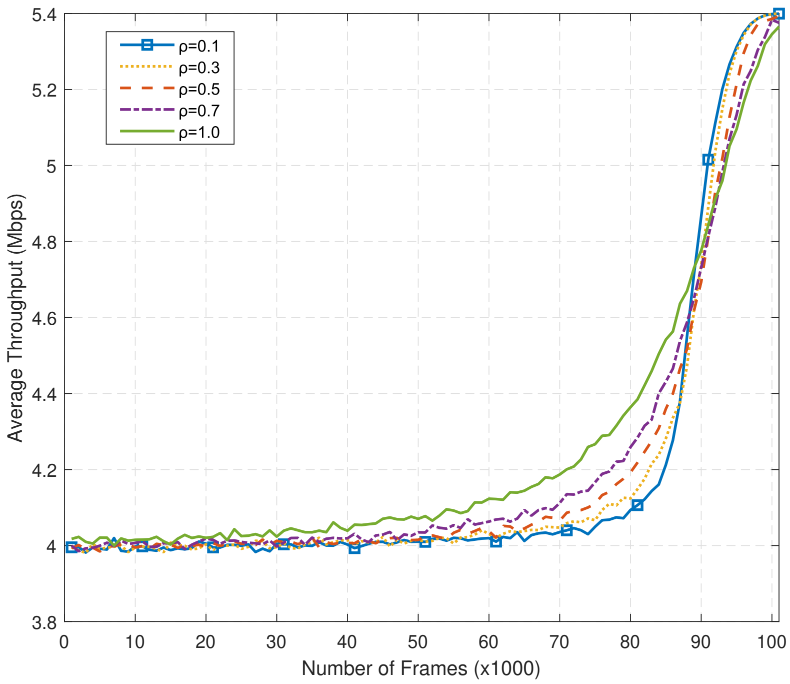

Figure 3.12: Average throughput with different smoothing parameters.

devices are easier to select a data collection point with smaller data rate. Additionally, ground devices require a longer time to learn a solution with higher throughput. Thus, the results show that the PMF of each ground device will converge onto the best solution when ρ equals 0.1. Therefore, in all subsequent experiments, the EWA smoothing parameter ρ is set to 0.1.

3.6.3.2 CPSP with Amplified SIC Reward

This experiment studies the impact of amplifying, i.e., $\omega \in \{1, 2\}$, the reward for transmissions that satisfy SIC. The value of ω is set to one when ground devices transmit independently. It performs 5000 simulation runs. In each simulation run, when the PMF of each ground device converges, the experiment in this section records the selected data collection point and corresponding transmitted data. Then the average throughput as well as the percentage of solutions that have SIC transmissions are calculated. Specifically, the percentage of solutions with SIC transmissions is denoted as p_ω. The temperature parameter τ adaptively decreases from 105 to

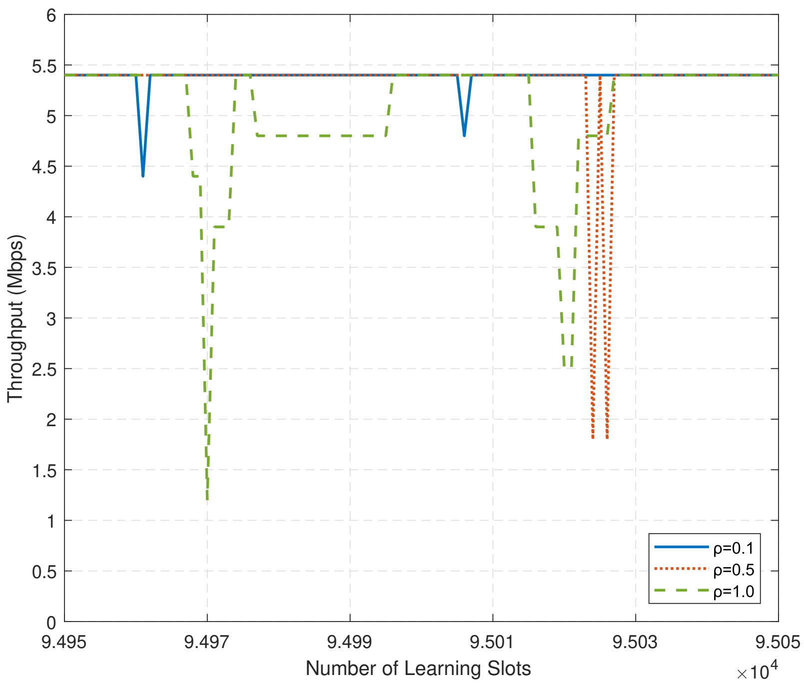

Figure 3.13: Throughput change details over 100 learning slots.

five. This experiment sets the EWA smoothing parameter to 0.1.

Referring to Figure 3.14(a), the value of p_ω increases substantially when ω increases from one to two. Specifically, the value of p_ω initially increases from 0.5% to 1.75% when ω is between one to 1.2. It then increases from 1.75% to 97.95% when ω is in the range $[1.3, 1.7]$. When the reward of SIC transmissions is doubled, the value of p_ω reaches 100%. The reason is that when we amplify the reward for transmissions that satisfy SIC, ground devices will select the same data collection point to get a higher reward. This means ground devices are more likely to take advantage of SIC.

Referring to Figure 3.14(b), we observe that the average throughput first remains stable and then decreases before increasing. We see that the average throughput first stabilizes at 5.4 Mbps when ω is between $[1, 1.2]$. The reason is that when ω is smaller than 1.2, the percentage of solutions with SIC transmissions is only 0.5%, see Figure 3.14(a). Specifically, ground devices continue to select a different collection point to transmit at the highest data rate. The average throughput then decreases from 5.39 Mbps to 4.95 Mbps when ω increases from 1.3 to 1.7. This is because within this range, the number of solutions with SIC transmissions substantially increases from 1.75% to 97.95%, as shown in Figure 3.14(a). However, the sum-rate of transmissions with SIC will be smaller than the case when ground devices transmit independently. This is because SIC requires a different received power; thus, each ground device has different transmission rate. Therefore, the average throughput decreases. Additionally, when ω is in the range $[1.7, 2]$, the average throughput increases from 4.95 Mbps to 5.05 Mbps. The reason is that a higher reward of SIC transmissions encourages ground devices to select a better collection point that yields a higher data rate. Thus, the average throughput increases. It can thus be concluded that when the SIC reward amplification factor ω is two, the percentage of solutions with SIC transmissions reaches 100%. Additionally, the average throughput approaches to that when ground devices transmit independently. Thus, in all subsequent experiments, we will set ω to two to calculate the reward for SIC transmissions in CPSP-$r\omega$.

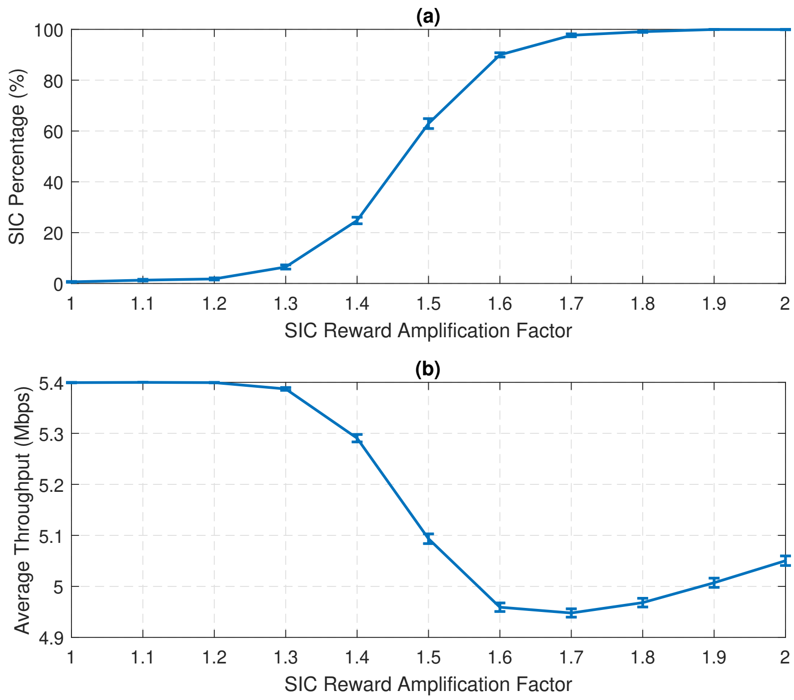

Figure 3.14: Average throughput and SIC percentage with different SIC reward amplification factors.

3.6.3.3 CPSP Reward Cases Comparison

The evaluation in this section compares the average throughput of ground devices and the average reward of selected data collection points for both CPSP-r and CPSP-$r\omega$. Same topology is used for both CPSP reward cases. The temperature parameter τ adaptively decreases from 105 to 5. EWA smoothing parameter; i.e., ρ, and SIC reward amplification factor; i.e., ω, is set to is set to 0.1 and two, respectively. The results to follow show the average reward and the average throughput for 20 simulation runs.

Referring to Figure 3.15, we see that the average reward of both methods first fluctuates around a small value before increasing substantially. Specifically, before the 70-th frame, the reward of CPSP-$r\omega$ fluctuates around 23 Mbps. It then increases from 23 Mbps to 81 Mbps within the learning frame $[70, 101]$. The average reward of CPSP-r first fluctuates around 16 Mbps and then increases to 54 Mbps in the 101-th learning frame. This is because ground devices in both CPSP-r and CPSP-$r\omega$ first explore all solutions and their reward. They then result in ground devices selecting the best solution, which helps improve the average reward. Moreover, we observe that the reward of CPSP-$r\omega$ is 1.5 times more than that of CPSP-r. The reason is that we amplify the reward of transmissions that satisfy SIC. Thus, ground devices are encouraged to select the same data collection point to get a higher reward.

As shown in Figure 3.16, the average throughput of CPSP-r and CPSP-$r\omega$ first varies around 4 Mbps and then significantly increases to 5.4 Mbps and 5.1 Mbps, respectively. This is because ground devices that use CPSP-r transmit independently and each of them is able to transmit with the highest sum-rate. However, CPSP-$r\omega$ encourages SIC transmissions that will cause some collection points to receive more data and others with zero data. Consequently, the average throughput of CPSP-$r\omega$ is 0.3 Mbps smaller than CPSP-r when the PMF of all ground devices converges.

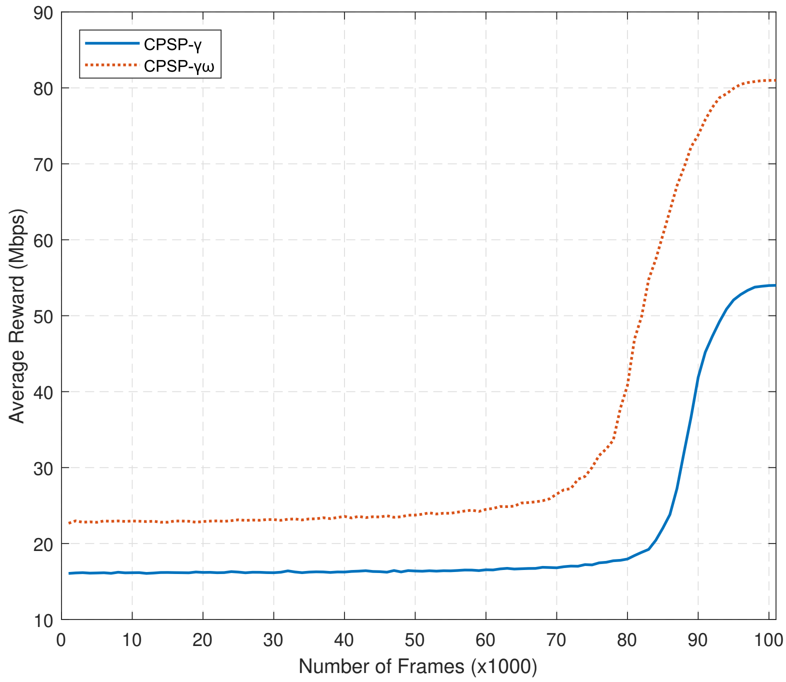

Figure 3.15: Average reward comparison of CPSP-r and CPSP-$r\omega$.

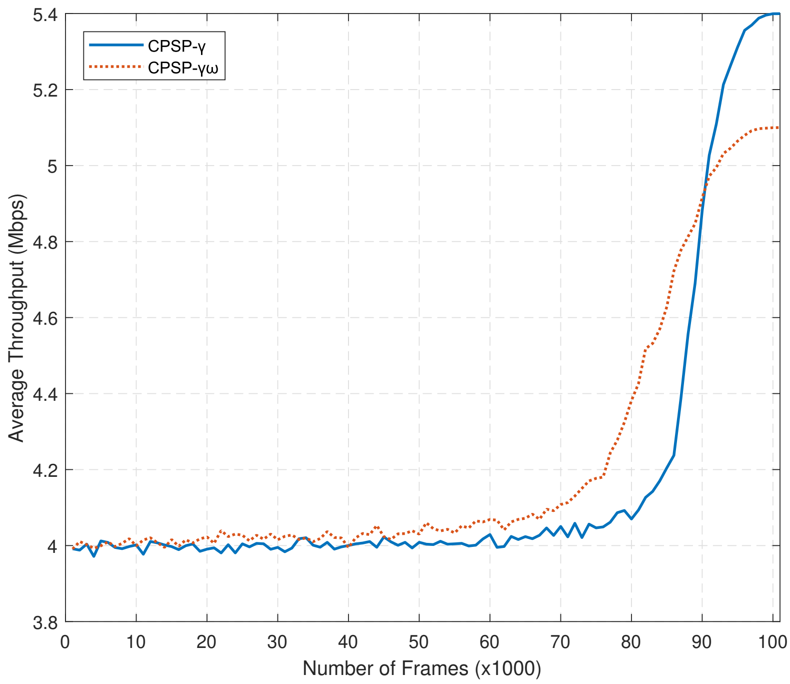

Figure 3.16: Average throughput comparison of CPSP-r and CPSP-$r\omega$.

3.6.4 Average Throughput Comparison of Different Methods

Here, the average throughput of SIC-ILP is compared against other proposed methods as well as conventional Slotted Aloha and TDMA. Specifically, two cases are considered for CE methods, namely CEF-φ^c and CEA-φ^c. The cut-off reward parameter γ that identifies 'elite' samples in CEA-φ^c adaptively increases from 0.95 to 0.99. Additionally, for CPSP, the average throughput of two reward cases are plotted; namely, CPSP-r and CPSP-$r\omega$. In particular, the SIC reward amplification factor ω is set to two for CPSP-$r\omega$. Both Slotted Aloha and TDMA do not have SIC. The number of data collection points is fixed at ten. The number of ground devices increases from one to ten. The same topology is used for all methods. The plotted results are an average of 50 simulation runs.

From Figure 3.17, we see that SIC-ILP outperforms the other five methods because it is able to find the optimal link sets at each data collection point that leads to the maximal average throughput. For example, in the case of ten ground devices, the average data rate is approximately 12.6 Mbps. However, CEA-φ^c, CEFφ^c and GCTS achieve 11.23 Mbps, 10.83 Mbps, and 10.83 Mbps, respectively, for the same number of ground devices. The average throughput of TDMA and CPSP-r is 5.4 Mbps, respectively. Additionally, CPSP-$r\omega$ achieves 5.2 Mbps. The average throughput of Slotted Aloha is only 3.8 Mbps. Referring to Figure 3.17, we find that with increasing number of ground devices, the average throughput of SIC-ILP, GCTS and Slotted Aloha increases. The reason is that a higher number of ground devices yields larger link sets at each data collection point. Therefore, when there are multiple ground devices, the chance to activate and/or construct link sets with better sum-rate increases, which results in a higher throughput. However, the average throughput of TDMA is fixed at 5.4 Mbps for any number of ground devices because it allows only one active uplink in each time slot.

As shown in Figure 3.17, the average throughput of CPSP-r and CPSP-$r\omega$ increases linearly from 0.54 Mbps to 5.4 Mbps and 5.2 Mbps, respectively. This is because for CPSP-r, ground devices transmit independently; thus, each of them is able to upload with the highest data rate of 54 Mbps. Therefore, with a newly added ground device, the corresponding sum-rate will increase by 54 Mbits, which helps increase the average throughput by 0.54 Mbps. For CPSP-$r\omega$, ground devices will learn the best data collection point that yields the highest transmission success and takes advantage of SIC. Therefore, 80% SIC transmissions are able to attain the highest data rate as ground devices transmit independently. Consequently, the average throughput of both CPSP-r and CPSP-$r\omega$ increases linearly with increasing number of ground devices. Moreover, the average throughput difference between CPSP-r and CPSP-$r\omega$ increases from zero to 0.2 Mbps when the number of ground devices increases from one to ten. The reason is that with more ground devices, the received power difference between ground devices increases; thus, more ground devices are able to transmit simultaneously. Therefore, the number of SIC transmissions increases that leads to a decrease in sum-rate. Consequently, the average throughput difference between CPSP-r and CPSP-$r\omega$ increases.

From Figure 3.17, we observe that when the number of ground devices is more than eight, the average throughput growth of both CEF-φ^c and CEA-φ^c decreases from 5.36 Mbps and 5.67 Mbps to 0.06 Mbps and 0.16 Mbps, respectively. This is due to SIC's decoding limit, which restricts the number of uplinks per link set to be no more than L_{max} [61]. Therefore, when the number of ground devices is twice L_{max}, the average throughput will only increase by around 0.1 Mbps.

3.7 Conclusion

This chapter considers deriving a link schedule that allows a SIC-capable UAV to collect data at fixed collection points. Its main contributions include an ILP, two heuristic methods called CE and GCTS as well as a distributed MAC protocol called CPSP. The results indicate that equipping a UAV with a SIC radio doubles the

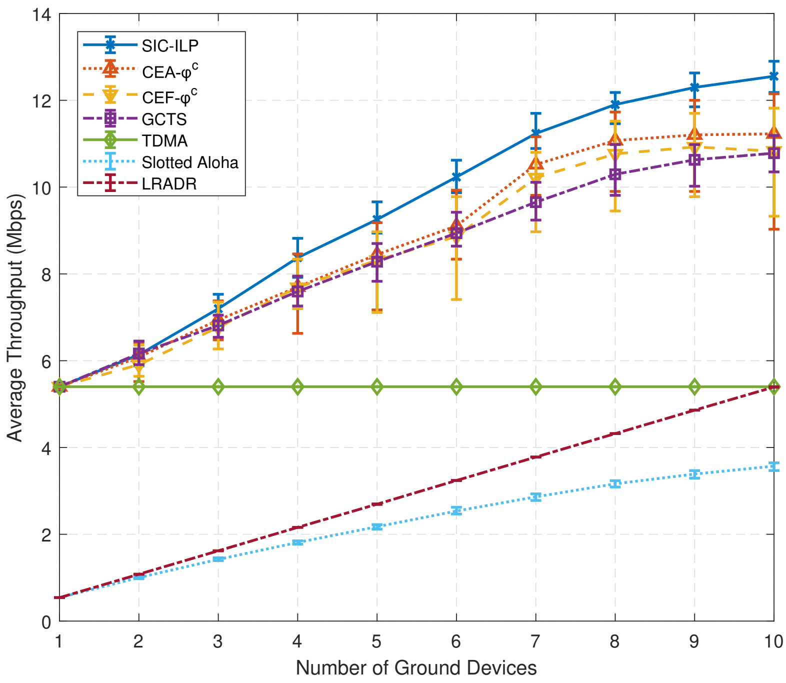

Figure 3.17: Average throughput comparison of proposed methods: SIC-ILP,CEF-φ^c, CEA-φ^c, GCTS, CPSP-r, CPSP-$r\omega$, Slotted Aloha and TDMA.

amount of uploaded data. In addition, a higher number of ground devices results in a higher throughput. Also, the number of ground devices and data collection points jointly affect the fairness of a ground device. Numerical results also show that the average throughput is affected by the number of data collection points, the speed and altitude of the UAV as well as the position of ground devices.

A limitation of the formulation in this chapter is that it assumes the UAV flies along a given fixed-height trajectory. Additionally, it does not consider the propulsion energy consumption of the UAV. Another limitation is that ground devices are with zero elevation. These limitations are addressed in next Chapter 4.

Chapter 4

Joint Trajectory and Link Scheduling Optimization in UAV Networks

This chapter extends the work studied in the previous chapter by considering (i) elevated devices, (ii) propulsion energy consumption of a UAV, and (iii) changing the height of a UAV's trajectory. Specifically, this chapter contains the following contributions:

- This work considers two approaches to maximize the number of devices transmitting to the UAV or uplinks at each data collection point. First, the UAV is equipped with a SIC radio that allows it to receive multiple transmissions simultaneously. In particular, each uplink transmission meets a given SINR threshold. A fundamental problem is the *classic* NP-hard link scheduling [64], where the UAV needs to decide which devices are scheduled to transmit together in each time slot. Second, the work in this chapter optimizes the trajectory of the UAV, whereby the data collection points used to gather data from devices are optimized correspondingly. In particular, this work seeks data collection points that allow a high number of SIC decoding successes or simultaneous uplink transmissions. This also means the trajectory selected by the UAV will affect the resulting uplinks transmission schedule. This is the first work that considers a combinatoric problem of selecting data collection points and transmission sets in order to maximize the total data collected by a SIC-capable UAV over multiple time slots. Note that works that consider power allocation are complementary to this work.

- A novel ILP is outlined to (i) select the best trajectory for a UAV, and (ii) compute a link schedule for each data collection point along the selected trajectory that maximizes the total amount of collected data. Additionally, this work presents a novel heuristic algorithm called Iteratively Construct Link Schedule and Trajectory (ICLST). In particular, the selection of link sets/schedules is according to the individual sum-rate. Moreover, the impact of randomly selecting link sets/schedules is studied when the UAV uses ICLST. This work also proposes a novel learning based protocol that is based on State-Action-Reward-State-Action (SARSA) [73]. The SARSA-based learning protocol allows the UAV to independently learn a trajectory and the corresponding link schedule that maximize the amount of collected data and minimize its energy usage.

- This chapter shows that the total amount of collected data and constructed link schedule is affected by the following factors: (i) transmission environments, (ii) different number of devices and devices' placement methods, (iii) different number of columns of the grid that a UAV flies, and (iv) different heights of a UAV.

- This chapter presents the following findings: (a) equipping a UAV with a SIC radio doubles the amount of collected data, (b) placing devices at different heights affects the average throughput. In particular, When devices are at an elevated height, the average throughput is 1.5 Mbits higher than the case when devices are placed on the ground or have zero elevation, (c) when a UAV flies at different heights, it is able to collect more data, (d) the average throughput of each device is affected by the planning horizon length and the position of devices, (e) the novel heuristic ICLST is capable of producing a schedule that is near optimal, and (f) the proposed novel learning protocol yields a schedule with the highest energy-efficiency.

The rest chapter is structured as follows. Section 4.1 introduces the network setup and notations. Section 4.2 presents the formulated ILP model. Section 4.3 presents a novel heuristic algorithm ICLST with a link set/schedule selection policy, called Highest Sum-Rate Selection (HSRS). Then Section 4.4 presents the details of a SARSA-based learning approach followed by Section 4.5, which discusses collected results. Section 4.6 concludes this work.

4.1 Preliminaries

Table 4.1 summarizes our nomenclature. The problem in this chapter considers a single-hop wireless system consisting of multiple devices and a mobile SIC-capable rotary-wing UAV. Let G be the set of devices. These devices are indexed as $1, 2, \ldots, |G|$, where $|.|$ denotes the cardinality of a set. The first device, aka g_1, is set at the origin. The distance between the first and last device is denoted as D; this is referred to as the *deployment range*. Devices always have data to transmit. Time is discrete, and indexed by t.

The rotary-wing UAV u operates on an area divided into a grid with $|M|$ columns and $|N|$ rows. Each cell on the grid has size $\tilde{s} \times \hat{s}$ (in m^2), where the horizontal and vertical side length of a cell is denoted as $\tilde{s}$ and $\hat{s}$, respectively. Each intersection point on the grid is a possible data collection point. This means a $|M| \times |N|$ grid size has $|M| \times |N|$ number of collection points. Let C^m be the m-th column of the said grid or data collection points, where $m \in M$. Define n to be a data collection point in column m, where $n \in C^m$. The height of data collection point n in column m is denoted as h_n^m. In each time epoch t, the UAV will select at most one data collection point from each column. At each data collection point (n, m), each device

Symbol	Description
1.	**Sets**
G	The set of devices
M	The set of columns over the grid
N	The set of rows over the grid
T	Time horizon of the UAV
$\mathbb{L}^{mn}$	Set of all uplinks at collection point (n, m)
2.	**Constants**
D	Deployment range
h	Height of the UAV at a starting point
$\tilde{s}$	Horizontal side length of each grid cell
$\hat{s}$	Vertical side length of each grid cell
α	Pass loss exponent
β	SINR threshold
N_0	Ambient noise power
P	Transmit power of devices
E_{max}	Energy budget of the UAV
L_{max}	The maximum number of simultaneous uplinks that the UAV can decode
3.	**Parameters**
l_{in}^{m}	Uplink from device i to a point n in column m of the grid
d_{in}^{m}	Transmission distance of uplink l_{in}^{m}
r_{in}^{m}	Data rate of uplink l_{in}^{m}
P_{in}^{m}	Received power at the point (n, m) from device i
$PL(d_{in}^{m})$	Path loss of uplink l_{in}^{m}
l_{nq}^{mk}	Edge between points (n, m) and (q, k)
$\hat{P}_{l_{nq}^{mk}}$	Power consumption of vertical movement
$\tilde{P}_{l_{nq}^{mk}}$	Power consumption of horizontal movement
$D_{l_{nq}^{mk}}$	Length of edge l_{nq}^{mk}
$\Delta h_{l_{nq}^{mk}}$	Height difference between data collection points (n, m) and (q, k)
$E_{l_{nq}^{mk}}$	Mobility energy consumption of the UAV that moves along the edge l_{nq}^{mk}

Table 4.1: A summary of notations.

i has one uplink that is denoted as l^m_{in}, where $i \in G$ and $n \in C^m$.

The UAV has a SIC radio [61]. It is capable of decoding up to L_{max} simultaneous uplink transmissions. In particular, it separates, decodes, and removes signals from a composite signal in multiple stages. To ensure decoding success, the receive power of each uplink transmission must be sufficiently different, where the UAV starts the decoding process from the strongest signal. This is because the UAV (or receiver) needs to first extract/decode the strongest signal from the received composite signal; from Eq. (4.1), this decoding is only successful if the SINR of the said strongest signal is above the threshold β. The decoded signal is then subtracted from the composite signal. After that, the UAV then continues to the next stage. It repeats the said process to decode the next transmission with the strongest signal. As an example, assume UAV u is receiving from $|G|$ devices simultaneously. Assume the received power at the UAV u is in non-decreasing order: $P_1 \leq P_2 \leq \cdots \leq P_{|G|}$. The decoding order is thus $|G|, |G|-1, \ldots, 2, 1$. That is, the signal with received power P_1 can be decoded if and only if all the preceding stronger signals are first decoded and removed. Formally, we have,

$$\begin{aligned}
&\text{Stage 1} && \frac{P_{|G|}}{N_0 + \sum_{i=1}^{|G|-1} P_i} \geq \beta \\
&\text{Stage 2} && \frac{P_{|G|-1}}{N_0 + \sum_{i=1}^{|G|-2} P_i} \geq \beta \\
&\quad\vdots && \quad\vdots \\
&\text{Stage } |G|-q+1 && \frac{P_{q\varphi}}{N_0 + \sum_{i=1}^{q-1} P_i} \geq \beta.
\end{aligned} \tag{4.1}$$

Eq. (4.1) shows that for a given uplink, its SINR and/or Signal-to-Noise Ratio (SNR) must be no less than the threshold value β, which corresponds to a given Modulation and Coding Scheme (MCS) or data rate; see [166] for example values. In Eq. (4.1), N_0 denotes the ambient noise power.

At each data collection point (n, m), there are multiple link sets. Each link set contains one or more uplinks from devices that satisfy inequality (4.1). Uplinks from all devices at data collection point (n, m) are stored in the set $\mathbb{L}^{mn} = \{l_{in}^{m} \mid \forall i \in G\}$. The j-th link set that satisfies SIC constraints at point (n,m) is denoted as L_j^{mn}, where $j \in \{1, \ldots, |L^{mn}|\}$, and $L_j^{mn} \subseteq \mathbb{L}^{mn}$. The maximum number of simultaneous uplinks in each link set L_j^{mn} is set to L_{max}, which is a technological limit that corresponds to the maximum number of signals that can be cancelled by a SIC radio [61]. The data rate of uplink l_{in}^{m} is denoted as r_{in}^{m}. Specifically, it is a function of β; as an example, if $\beta = 5$ (dB), then as per [166], an IEEE 802.11a access point will operate at 6 Mbps. The sum-rate of link set L_j^{mn} is denoted as R_j^{mn}. It is defined as $R_j^{mn} = \sum_{i \in L_j^{mn}} r_{in}^{m}$.

Figure 4.1 illustrates an example with one UAV and three devices, namely g_1, g_2 and g_3. The grid area used by the UAV has four columns and five rows that are labeled as $m_1, \ldots, m_4$ and $n_1, \ldots, n_5$, respectively. Note that the grid length is the same as the deployment range D of devices. The UAV is initially located at data collection point $(1, 1)$ and flies over the grid from m_1 to m_4 to collect data from devices. One possible link set at data collection point $(3, 4)$ is $\mathbb{L}^{43} = \{l_{13}^{4}, l_{23}^{4}\}$.

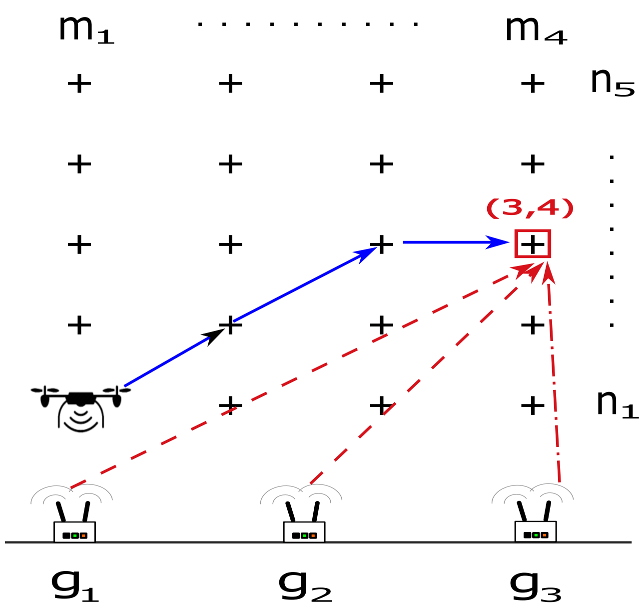

Figure 4.1: A system setting example with one UAV and three devices. The given grid has four columns and five rows.

Block fading is assumed in the problem of this chapter, where the channel remains static for each time epoch t. The ground-to-air path loss model is as per [32], which considers the effect of the environment on the occurrence of Line of Sight (LoS) uplinks. Specifically, according to the location of devices and the UAV as well as the urban environment, each device has some probability of having a LoS or Non-LoS (NLoS) uplink [170]. The LoS probability of uplink l_{in}^m is calculated as

$$p_{in}^m = \frac{1}{1 + a\exp(-b[\theta_{in}^m - a])}, \tag{4.2}$$

where a and b are constant values that depend on the carrier frequency and the type of environment, such as rural, urban and/or dense urban. Let θ_{in}^m be the elevation angle (in degree) between device i and data collection point (n, m). Specifically, $\theta_{in}^m = \frac{180}{\pi} \times \arcsin \frac{h_n^m}{d_{in}^m}$, where d_{in}^m is the Euclidean distance from device i to data collection point (n, m). The NLoS probability is $\hat{p}_{in}^m = 1 - p_{in}^m$.

The ground-to-air path loss consists of two parts: (i) free space path loss, and (ii) attenuation from shadowing and scattering in urban environment [170]. In addition, as per [170], there is a probability associated with the occurrence of LoS and non-LoS (NLoS), respectively. As per [170], this work assumes that all transmitters and receivers have an omni-directional antenna. Let $PL(d_{in}^m)$ be the average path loss between device i and data collection point (n, m). The average path loss $PL(d_{in}^m)$ is computed as

$$PL(d_{in}^m) = \left(\frac{4\pi f_c d_{in}^m}{c}\right)^{\alpha} \times \left(p_{in}^m \eta_{LoS} + \hat{p}_{in}^m \eta_{NLoS}\right), \tag{4.3}$$

where f_c is the carrier frequency, c is the speed of light, α is the pass loss exponent, η_{LoS} and η_{NLoS} are the additional attenuation coefficient of the LoS and NLoS case, respectively. All devices have a fixed transmit power P (Watt). The received power (in Watt) from device i when the UAV is at data collection point (n, m) is expressed as

$$P_{in}^m = \frac{P}{PL(d_{in}^m)}. \tag{4.4}$$

The UAV's energy consumption consists of two parts: (i) communication, and (ii) propulsion energy [13]. However, as noted in [171], communication related energy can be ignored because it is usually much smaller than a UAV's propulsion energy. Hence, this thesis ignores the energy consumption relating to SIC signal processing. Note that the total energy consumption of the UAV cannot exceed the given budget E_{max}. Let l_{nq}^{mk} be the edge between data collection points (n, m) and (q, k). The total energy consumed by the rotary-wing UAV to traverse the edge l_{nq}^{mk} is computed as [32],

$$E_{l_{nq}^{mk}} = \frac{D_{l_{nq}^{mk}}}{v} \left(\hat{P}_{l_{nq}^{mk}} + \tilde{P}_{l_{nq}^{mk}} \right), \tag{4.5}$$

where $D_{l_{nq}^{mk}}$ is the length of edge l_{nq}^{mk}, $D_{l_{nq}^{mk}}/v$ is the flight duration, $\hat{P}_{l_{nq}^{mk}}$ and $\tilde{P}_{l_{nq}^{mk}}$ correspond to the power consumption for vertical and horizontal movement, respectively. The height difference between data collection points (n, m) and (q, k) is denoted as $\Delta h_{l_{nq}^{mk}}$. Thus, the effective vertical and horizontal velocities are defined as $\hat{v}_{l_{nq}^{mk}} = v \sin \phi$ and $\tilde{v}_{l_{nq}^{mk}} = v \cos \phi$, respectively, where

$$\phi = \arcsin \frac{\Delta h_{l_{nq}^{mk}}}{D_{l_{nq}^{mk}}}. \tag{4.6}$$

The UAV's horizontal flight power consumption $\tilde{P}_{l_{nq}^{mk}}$ has three components, namely blade power profile, parasitic power and induced power [13]. The blade profile power and parasitic power are needed to overcome the profile drag of the blades and the fuselage drag, respectively. The induced power is needed for overcoming the lift-induced drag of the blades. The horizontal power consumption $\tilde{P}_{l_{nq}^{mk}}$ is computed as derived in [13]:

$$\begin{aligned} \tilde{P}_{l_{nq}^{mk}} = & \underbrace{P_0 \left(1 + 3 \left[\frac{\tilde{v}_{l_{nq}^{mk}}}{\Omega R} \right]^2 \right)}_{\text{Blade profile}} + \underbrace{\frac{1}{2} \rho d_0 s_0 A (\tilde{v}_{l_{nq}^{mk}})^3}_{\text{Parasitic}} \\ & + \underbrace{P_i \left(\sqrt{1 + \frac{1}{4} \left[\frac{\tilde{v}_{l_{nq}^{mk}}}{v_0} \right]^4} - \frac{1}{2} \left[\frac{\tilde{v}_{l_{nq}^{mk}}}{v_0} \right]^2 \right)^{1/2}}_{\text{Induced}}, \end{aligned} \tag{4.7}$$

where ρ is the air density (kg/m^3), Ω is the blade angular velocity in radians/second,

R is the rotor radius in meter, A is the rotor disc area that is defined as πR^2, v_0 is the mean rotor induced velocity during hovering, d_0 is the fuselage drag ratio and s_0 is the rotor solidity that is defined as the ratio of the total blade area to the rotor disc area. Specifically, it is $s_0 \triangleq \frac{N_b c_b}{\pi R}$, where c_b and N_b are the blade chord length and the number of blades, respectively. Define P_0 and P_i in Watt are two constants that represent the blade profile power and induced power during hovering, respectively. The blade profile power can be expressed as

$$P_0 = \frac{\delta}{8} \rho s_0 A \Omega^3 R^3, \tag{4.8}$$

where δ is the profile drag coefficient. The induced power during hovering is calculated as

$$P_i = (1 + \kappa) \frac{W^{3/2}}{\sqrt{2\rho A}}, \tag{4.9}$$

where κ is the incremental correction factor to induce power and W is the weight of the UAV (in Newton).

The power consumption $\hat{P}_{l_{nq}^{mk}}$ when the UAV climbs vertically and/or descends is computed as per [32]:

$$\hat{P}_{l_{nq}^{mk}} = \begin{cases} \frac{W}{2} \hat{v}_{l_{nq}^{mk}} + \frac{W}{2} \sqrt{\hat{v}_{l_{nq}^{mk}}^2 + \frac{2W}{\rho \pi R^2}}, & \text{Climbing;} \\ \frac{W}{2} \hat{v}_{l_{nq}^{mk}} - \frac{W}{2} \sqrt{\hat{v}_{l_{nq}^{mk}}^2 - \frac{2W}{\rho \pi R^2}}, & \text{Descending.} \end{cases} \tag{4.10}$$

Note that when the horizontal side length $\tilde{s}$ of grid cell is much longer than the vertical side length $\hat{s}$, the UAV will climb and/or descend slowly; thus, $\hat{v}_{l_{nq}^{mk}}^2$ is smaller than $\frac{2W}{\rho \pi R^2}$. Therefore, when $\hat{v}_{l_{nq}^{mk}}^2 < \frac{2W}{\rho \pi R^2}$, we assume $\hat{P}_{l_{nq}^{mk}} = \frac{W}{2} \hat{v}_{l_{nq}^{mk}}$.

4.2 Problem Definition

The problem at hand is to select M data collection points that maximize the total uploaded data from devices to a SIC-capable UAV. Specifically, it needs to (i) optimize the trajectory of the UAV. To form this trajectory, the UAV needs to select a data collection point from each column of the grid. As an example, referring to Figure 4.1, one possible trajectory consists of points $(1,1)$, $(2,2)$, $(3,3)$ and $(3,4)$, and (ii) select the link set at each data collection point. Referring to Figure 4.1, at data collection point $(3,4)$, there are the following links sets: $\{l_{13}^4, l_{23}^4\}$ and $\{l_{33}^4\}$.

There are three binary decision variables, namely x_j^{mn}, x^{mn} and x^{qk}, and an auxiliary binary variable that is denoted as $\xi(L_j^{mn}, i)$. They are defined as follows:

- x_j^{mn}, which indicates whether the link set L_j^{mn} is active ($x_j^{mn} = 1$) at data collection point (n,m). That is, whether it is selected by the UAV to schedule uplink transmissions at data collection point (n,m).
- x^{mn}, which indicates whether point (n,m) is active.
- x^{qk}, which indicates whether point (q,k) is active.
- $\xi(L_j^{mn}, i)$, which indicates whether ground device i is in the link set L_j^{mn} ($\xi(L_j^{mn}, i) = 1$) at collection point (n,m).

Mathematically, the ILP to follow aims to maximize the sum-rate of active link sets:

Constraint (4.11b) ensures at most one link set is selected at each possible data collection point in the grid. Constraint (4.11c) ensures one link set is selected in each column of the grid. A data collection point is selected only when it has an active link set; see (4.11d). Constraint (4.11e) ensures each device has an opportunity to transmit in the final schedule. Constraint (4.11f) ensures the UAV does not expend more than its available energy. Constraint (4.11f) ensures that only when two points are selected, the edge in between will be activated. The last set of constraints, namely (4.11g), (4.11h) and (4.11i), ensures variable x_j^{mn}, x^{mn} and x^{qk} are binary. Notice that (4.11e) and (4.11f) are not linear as it involves the product of two binary variables. To linearize (4.11f), for any two data collection points in the grid, the constraint is reformulated as follows: (i) when both variables x^{mn} and

$$\max_{x_j^{mn}, x^{mn}, x^{qk}} \quad \sum_{m\in M}\sum_{n\in C^m}\sum_{j\in L^{mn}} R_j^{mn} x_j^{mn} \tag{4.11a}$$

s.t.

$$\sum_{j\in L^{mn}} x_j^{mn} \leq 1, \ \forall m \in M, \forall n \in C^m, \tag{4.11b}$$

$$\sum_{n\in C^m}\sum_{j\in L^{mn}} x_j^{mn} = 1, \ \forall m \in M, \tag{4.11c}$$

$$x^{mn} = \sum_{j\in L^{mn}} x_j^{mn}, \ \forall m \in M, \forall n \in C^m, \tag{4.11d}$$

$$\sum_{m\in M}\sum_{n\in C^m}\sum_{j\in L^{mn}} \xi(L_j^{mn}, i) x_j^{mn} \geq 1, \ \forall i \in G, \tag{4.11e}$$

$$\sum_{m\in M}\sum_{k\in M\setminus m}\sum_{n\in C^m}\sum_{q\in C^k} E_{nq}^{mk} x^{mn} x^{qk} \leq E_{max}, \tag{4.11f}$$

$$x_j^{mn} \in \{0,1\}, \ \forall m \in M, \forall n \in C^m, \forall j \in L^{mn}, \tag{4.11g}$$

$$x^{mn} \in \{0,1\}, \ \forall m \in M, \forall n \in C^m, \tag{4.11h}$$

$$x^{qk} \in \{0,1\}, \ \forall k \in M \setminus m, \forall q \in C^k. \tag{4.11i}$$

x^{kq} have a value of one, the inequality $x^{mn}x^{qk} \geq x^{mn} + x^{kq} - 1$ forces $x^{mn}x^{qk}$ to equal one, (ii) $x^{mn}x^{qk} \leq x^{mn}$ ensures $x^{mn}x^{qk}$ is zero when x^{mn} equals to zero, and (iii) $x^{mn}x^{qk} \leq x^{kq}$ ensures $x^{mn}x^{qk}$ is zero when x^{qk} equals to zero. We can linearize (4.11e) with the same method that is used to linearize (4.11f).

Finally, there are four remarks. First, in the considered problem, if there is just one SINR threshold and transmit power level, it can be reduced from the well-known NP-hard weighted set cover problem [167]. In particular, the problem is to find M set covers that maximize the sum-rate (weight) subject to devices being included in at least one of these M set covers. Second, the formulation in Section 4.2 is general and it is able to capture more complex setups; namely, devices with different SINR threshold values and transmit power levels. Briefly, the collection $\mathbb{L}^{mn}$ at each possible data collection point (n, m) can include link sets for all possible combinations of SINR threshold values and transmit power levels for each ground device. Denote these as $\beta = \{\beta_1, \beta_2, \ldots, \beta_K\}$ and $P = \{P_1, P_2, \ldots, P_Z\}$, respectively. To generate link sets, for each SINR threshold in β, the formulated ILP computes all possible links and transmit power that satisfy the given SINR threshold. Third, this chapter does not consider packet scheduling. The focus of this work is to provide one or more transmission opportunities to devices. In each of these opportunities, a device can adopt any scheduler to transmit its packets to the UAV. Lastly, the formulated ILP, see (4.11), is not suitable for large-scale networks. This is because there is a decision variable for each link set. Hence, the number of decision variables in the ILP increases exponentially with the number of devices. The next section will propose a simplified ILP where there is only one decision variable for each data collection point.

4.3 Heuristic Algorithm: ICLST

The basic idea of Iteratively Construct Link Schedule and Trajectory (ICLST) is as follows. First, it selects one link set for each data collection point using a selection policy called Highest Sum-Rate Selection (HSRS). Briefly, HSRS considers the sum-rate of a transmission set and also how many times a device has been paired with the UAV. ICLST then uses a simplified ILP to select the candidate data collection points that form the trajectory of the UAV.

Algorithm 6 shows the general structure of ICLST. It uses the set $\mathcal{L}$ to store all selected link sets. The function *HSRS()* selects a link set using HSRS. The function *SILP()* returns a link schedule $\mathbb{S}$. Referring to Algorithm 6, Line 3 calls *HSRS()* to select the link set $\mathcal{L}^{mn}$ for data collection point (n, m). Then in Line 4, the selected link set $\mathcal{L}^{mn}$ is included into the set $\mathcal{L}$. After that, the UAV calls *SILP()* to obtain the link schedule $\mathbb{S}$ that maximizes the sum-rate, see Line 7.

4.3.1 Link Set Selection Policy - HSRS

HSRS aims to select the link set with the highest sum-rate R_j^{mn} for each data collection point (n, m) and includes it into the vector $\mathcal{L}$, see Algorithm 7. Algorithm 7 first defines *paired times* as the number of times that a device is activated and paired

Algorithm 6: Heuristic algorithm general structure.

input : M, N, L^{mn}
output: Link schedule $\mathbb{S}$
Initialize: $\mathcal{L} = \emptyset$
for $m \leftarrow 1$ *to* $|M|$ **do**
    for $n \leftarrow 1$ *to* $|N|$ **do**
        `/* Apply HSRS */`
        $\mathcal{L}^{mn}$ =*SelectLinkSets(*L^{mn}*)*
        $\mathcal{L} = \mathcal{L} \cup \mathcal{L}^{mn}$
    end
end
`/* Obtain the link schedule` $\mathbb{S}$ `*/`
$\mathbb{S} =$ *SILP(*$\mathcal{L}$*)*
Return $\mathbb{S}$

with the UAV. Devices that have not been paired with the UAV are defined as *unpaired devices*. Define $\mathcal{G}(1, |G|)$ to be a vector where the i-th element records the total paired times of device i, where $i = 1, \ldots, |G|$.

Referring to Algorithm 7, at the first data collection point $(1, 1)$, HSRS selects one link set with the highest sum-rate R_j^{mn} using the function *HighSumRate()*. This link set is then added into $\mathcal{L}$, see Line 4-5. Then in Line 6, HSRS updates $\mathcal{G}$ with the paired time of active devices at point $(1, 1)$. For subsequent data collection points, HSRS will select the link set that has the least number of paired times as well as yielding a high sum-rate. Specifically, in Line 8, HSRS sorts the set L^{mn} in decreasing order of their sum-rate. Then Line 9 extracts the first Z link sets from L^{mn} and adds them into the set $\hat{L}^{mn}$. This guarantees the selected link set with the most unpaired devices also yields a high sum-rate. After that, in Line 11, HSRS constructs a set $\mathcal{R}^{mn}(1, |\hat{L}^{mn}|)$ that records the total paired times for devices in each link set from the set $\hat{L}^{mn}$. Hence, HSRS uses the element in $\mathcal{R}^{mn}$ to indicate whether a link set has the most number of unpaired devices. HSRS then calls *SmallPairedTimes()* to select one link set with the most unpaired devices from the set $\hat{L}^{mn}$ for data collection point (n, m), see Line 13.

Algorithm 7: HSRS().

```
input  : M, N, L^{mn}
output: 𝓛
Initialize: 𝒢 = 0; 𝓡^{mn} = 𝓛 = ∅
for m ← 1 to |M| do
    for n ← 1 to |N| do
        if m = n = 1 then
            𝓛^{mn} = HighSumRate(L^{mn})
            𝓛 = 𝓛 ∪ 𝓛^{mn}
            Update devices' paired time in 𝒢
        else
            L^{mn} = Sort (R_1^{mn}, ..., R_{|L^{mn}|}^{mn})
            L̂^{mn} = (L_1^{mn}, ..., L_Z^{mn})
            for j ← 1 to |L̂^{mn}| do
                𝓡^{mn} = 𝓡^{mn} ∪ 𝓡_j^{mn}
            end
            𝓛^{mn} = SmallPairedTimes(L̂^{mn})
            𝓛 = 𝓛 ∪ 𝓛^{mn}
            Update devices' paired time in 𝒢
        end
    end
end
Return 𝓛
```

4.3.2 Simplified ILP

The function *SILP()* solves ILP (4.12) to select candidate data collection points to form a UAV trajectory. It has one binary decision variable, i.e., x^{mn}, that indicates whether the selected link set $\mathcal{L}^{mn}$ at point (n,m) is active. Moreover, the ILP also relies on an auxiliary binary variable, i.e., $\xi(\mathcal{L}^{mn},i)$, to track whether device i is in the select link set $\mathcal{L}^{mn}$ at collection point (n,m); i.e., $\xi(\mathcal{L}^{mn},i)=1$. Notice that this constraint is non-linear and can be linearized with the same method that is used to linearize (4.11e); see Section 4.2. Constraint (4.12b) ensures one data collection point must be selected in each column of the grid. Constraint (4.12c) guarantees each device is included in the derived schedule. Constraint (4.12d) ensures the total consumed propulsion energy will not exceed the energy budget of the UAV. As shown in (4.12d), only when two data collection points are active, the edge in between is active ($x^{mn}x^{qk}=1$); hence, this constraint is non-linear because it involves the product of two variables. It can be linearized with the same method that is used to linearize (4.11f); see Section 4.2. Constraints (4.12e), and (4.12f) ensure variables x^{mn} and x^{qk} are binary, respectively.

$$\max_{x^{mn}} \quad \sum_{m\in M}\sum_{n\in C^m} R^{mn}x^{mn} \tag{4.12a}$$

s.t.

$$\sum_{n\in C^m} x^{mn}=1,\ \forall m\in M, \tag{4.12b}$$

$$\sum_{m\in M}\sum_{n\in C^m}\xi(\mathcal{L}^{mn},i)x^{mn}\geq 1,\ \forall i\in G, \tag{4.12c}$$

$$\sum_{m\in M}\sum_{k\in M\setminus m}\sum_{n\in C^m}\sum_{q\in C^k} E^{mk}_{nq}x^{mn}x^{qk}\leq E_{max}, \tag{4.12d}$$

$$x^{mn}\in\{0,1\},\ \forall m\in M,\forall n\in C^m, \tag{4.12e}$$

$$x^{qk}\in\{0,1\},\ \forall k\in M\setminus m,\forall q\in C^k, \tag{4.12f}$$

This section concludes with the run time complexity of ICLST. For each data collection point (n,m), ICLST needs to select a link set $\mathcal{L}^{mn}$. Hence, Lines 1-6 of Algorithm 6 run for $|M||N|$ times when selecting link sets for all data collection points of the grid. Therefore, $|\mathcal{L}^{mn}|$ is bounded by $\mathcal{O}(|M||N|)$. For Line 3 of Algorithm 6, ICLST applies HSRS to select link sets. Regardless of whether ICLST selects link sets from set L^{mn} or set $\hat{L}^{mn}$, it has to check no more than $|L^{mn}|$ link sets. ICLST will call *SILP()* once to get the trajectory after selecting one link set for all data collection points. Consequently, the time complexity of the ICLST or Algorithm 6 is $\mathcal{O}(|M||N||L^{mn}|)$.

4.4 A Learning Protocol

This section outlines a learning-based protocol to select the best trajectory over a given grid. Compared to the previous solution, the main advantage of our SARSA-based protocol is that the UAV is able to learn by itself and does not need a central server.

This section first briefly explains SARSA. It is an on-policy reinforcement learning algorithm [73]. For each system state, it trains an agent to select the optimal action that yields the maximum expected reward. Define $\widehat{E}$ as a set of episodes. Each episode e consists of multiple time steps $\widehat{T}$. In each time step $t \in \widehat{T}$, the agent first observes a state s_t and then selects an action a_t as per its policy, where $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}(s_t)$. The term $\mathcal{S}$ and $\mathcal{A}(s_t)$ denote the state space and the action space of state s_t, respectively. For each action, there is a corresponding reward r_t. A SARSA agent maintains a Q table containing $Q(s_t, a_t)$ to represent the total expected reward for each state s_t and taking the action a_t. Specifically, $Q(s_t, a_t)$ is the sum of immediate reward r_t at the current time step and rewards obtained in future time steps and/or episodes. The Q-table is updated as follows,

$$Q(s_t, a_t) \leftarrow (1-\mu)Q(s_t, a_t) + \mu[r_t + \gamma Q(s_{t+1}, a_{t+1})], \tag{4.13}$$

where $\mu \in [0, 1]$ is the learning rate, and $\gamma \in [0, 1]$ is the discount factor that weighs the importance of the future Q-value $Q(s_{t+1}, a_{t+1})$. Referring to Eq. (4.13),

$Q(s_t, a_t)$ indicate the updated Q-value and old estimated Q-value of being in state s_t and executing action a_t, respectively. When the absolute value of the difference between the updated Q-vale and the old estimated Q-value is smaller than ε, we conclude that the Q-table has converged.

A general overview of our protocol is depicted in Figure 4.2. The UAV has four main tasks: (i) collect channel information from devices, (ii) select the link set at each data collection point, (iii) learn the best trajectory, and (iv) inform devices their transmission time. When the UAV is at a data collection point (n, m), it first sends a beacon to all devices to ascertain their channel condition. Then devices will send back their corresponding channel information $PL(d_{in}^m)$ to the UAV, where $i \in G, m \in M, n \in C^m$. After that, at each collection point (n, m), the UAV will use the received channel information to select a link set $\mathcal{L}^{mn}$. It then selects the link with the highest sum-rate R^{mn}. The UAV then uses our SARSA-based learning process to learn the best trajectory, i.e., $\mathcal{K}$, see Algorithm 8 for details. Then the UAV sends a message to devices and inform their transmission time. After that, they send their data to the UAV. Finally, the UAV will send an Acknowledgement (ACK) to all devices.

Define the state $s_t \in C^m$ as the data collection points of column m in the given grid that has $|M|$ columns. The set of action a_t for each state corresponds to the data collection points in the next column $m + 1$; i.e., $a_t \in C^{m+1}, m \in M$. The UAV starting and terminal state are defined as any data collection points in the first and last column of the grid, respectively. The immediate reward r_t for taking action a_t in state s_t in time step t consists of three parts: (i) if the total energy consumption exceeds the energy budget E_{max}, the immediate reward r_t will be set to zero, (ii) if the UAV reaches the terminal state, a bonus $\mathcal{B}$ is granted to the UAV, and (iii) the immediate reward is set as the energy efficiency in Mbits/Joule of (s_t, a_t) pair. Note that when the energy consumption exceeds the UAV's available fuel, the SARSA-based learning approach can also penalize the UAV by giving it a

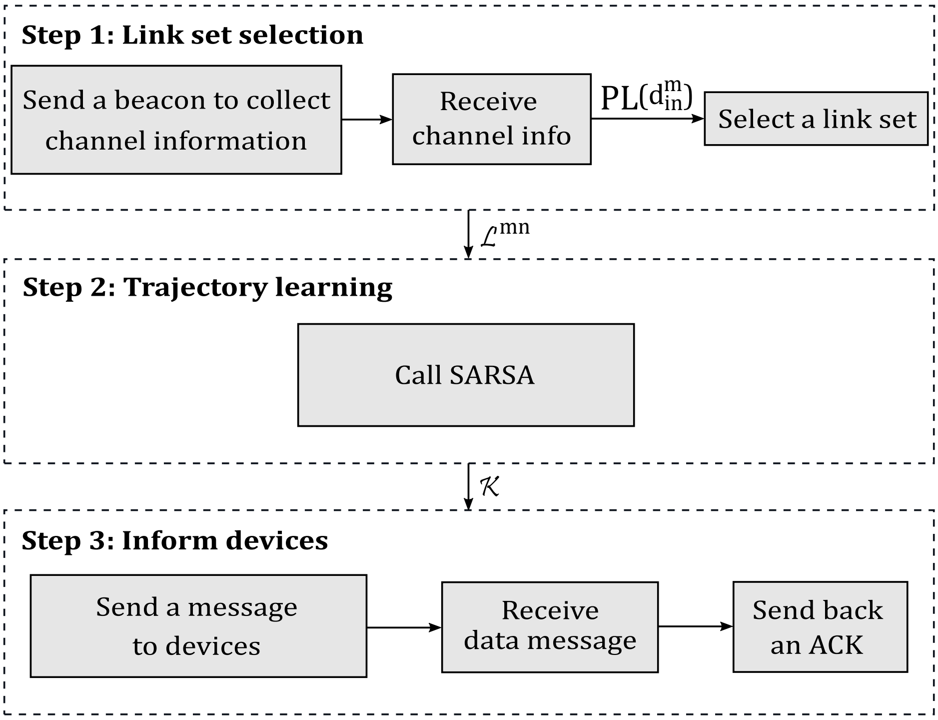

Figure 4.2: An overview of the novel SARSA-based protocol.

big negative reward, e.g., $-\mathcal{B}$. Formally, we have

$$r_t = \begin{cases} 0, & \text{if energy consumption exceed } E_{max} \\ \mathcal{E}^{s_t a_t} + \mathcal{B}, & \text{if } s_t \text{ is the terminal state} \\ \mathcal{E}^{s_t a_t}, & \text{otherwise} \end{cases} . \tag{4.14}$$

Define $\mathcal{E}^{s_t a_t}$ as the energy efficiency (Mbits/J) for taking action a_t in state s_t. The energy efficiency is calculated by

$$\mathcal{E}^{s_t a_t} = \frac{R^{s_t} + R^{a_t}}{E^{s_t a_t}}, \tag{4.15}$$

where $E^{s_t a_t}$ is the consumed propulsion energy between the state s_t and action a_t. The term R^{s_t} and R^{a_t} are the sum-rate of link set at points s_t and a_t, respectively.

The UAV uses the ε-greedy algorithm [172] to select the best action during the learning process. Specifically, in state s_t, the UAV selects the action a_t with the largest $Q(s_t, a_t)$ with probability ε, or randomly selects an action from the corresponding action space $\mathcal{A}(s_t)$ with probability $1 - \varepsilon$. Initially, all $Q(s_t, a_t)$ have the same value. This means the UAV will select an action uniformly.

Algorithm 8 illustrates the steps taken by a UAV to update its Q-table in each episode $e \in \widehat{E}$. The input is the starting point/state s_t and the old Q-table. The function ε-greedy() is used to select the action for any state. The UAV calls function $\mathcal{R}()$ to calculate the immediate reward of state-action pairs.

Referring to Algorithm 8, in each time step $t \in \widehat{T}$, the UAV first uses ε-greedy() to select the best action a_t for the state s_t, see Line 2. Then in Line 3, it uses $\mathcal{R}()$ to calculate the immediate reward r_t of the (s_t, a_t) pair. After that, in Line 4, the UAV calls ε-greedy algorithm again to select the action a_{t+1} for the next state s_{t+1}. Once the UAV gets the future Q-value $Q(s_{t+1}, a_{t+1})$, it is able to update the current Q-value via Eq. (4.13), see Line 5. If the terminal state has reached, the UAV will learn one possible trajectory and stop the current learning episode e before moving to the next episode $e + 1$.

Algorithm 8: SARSA-based learning process.

input : s_t, Old Q-table
output: New Q-table
1 **for** $t \leftarrow 1$ *to* $\widehat{T}$ **do**
2     $a_t = \varepsilon\text{-greedy}(s_t)$
3     $r_t = \mathcal{R}(s_t, a_t)$
4     $a_{t+1} = \varepsilon\text{-greedy}(s_{t+1})$
5   Update $Q(s_t, a_t)$. // see Eq. (4.13)
6   **if** *Terminal state has reached* **then**
7   break ;
8   **end**
9 **end**

4.5 Evaluation

We conduct our experiments in Matlab [168]. The considered system consists of up to 20 devices that are placed at different heights. The experiments in Section 4.5 use the SINR threshold and data rate from Cisco; see [166]. A transmission is successful if its SINR and/or SNR exceeds $\beta = 5$ (dB). The simulation settings are listed in Table 4.2. Specifically, the parameter values that are related to the ground-to-air path loss model and the calculation of UAV's propulsion energy consumption are from [170] [13]. For the novel proposed SARSA-based protocol, the value of ε increases from 0.1 to 1. Specifically, for every 2000 episodes, ε will be increased by 0.1. Lastly, except for Section 4.5.1.1, the deployment range D of devices is fixed at 200 meter. Moreover, except for Section 4.5.1.2, the system settings are in urban environment.

Symbol	**Value**	**Symbol**	**Value**
W	20 N	N_b	4
R	0.4 m	A	0.503 m^2
c_b	0.0157	s_0	0.05
v_0	4.03	d_0	0.6
δ	0.012	κ	0.1
ρ	1.225 kg/m^3	Ω	300 rad/s
a	11.95	b	0.14
α	2.3	β	5 dB
η_{LoS}	1.0	η_{NLoS}	20
c	3×10^8 m/s	f_c	2 GHz
v	30 m/s	N_0	-90 dBm
P	1 W	L_{max}	4
μ	0.03	γ	0.4
ε	0.1 - 0.9	λ	2
E_{max}	5000 J	D	200 m

Table 4.2: Simulation settings for the considered joint trajectory and link scheduling problem.

To this end, the experiments to follow compare the results obtained from solving the formulated ILP, labeled as SIC-ILP, which yields the optimal solution. By applying SIC-ILP, the experiments first study two cases: (i) devices with zero elevation; see Section 4.5.1, and (ii) devices with elevated height; see Section 4.5.2. Under each case, experiments are conducted to study the impact of related parameters, such as the devices placement methods, the size of the grid, the height distribution of elevated devices, etc., so as to get the optimal configuration under our system settings. For both cases, the UAV flies at a fixed or different heights that are determined via

Eq. (4.11) and/or (4.12). The experiments also study the performance of different link selection policies that are used by ICLST; see Section 4.5.1.4. It then compare SIC-ILP against three other proposed methods, namely HSRS, SARSA-based protocol and TDMA, under the obtained optimal configuration; see Section 4.5.3.

Figure 4.3 and 4.4 are two examples that show the system settings when devices with zero elevation and with elevated height, respectively. The number of devices $|G|$ is 20. When devices with zero elevation, the grid has seven rows and 20 columns, see Figure 4.3. When the height of devices varies from zero to 30 m, the grid has 12 rows and 20 columns, see Figure 4.4. The deployment range D of devices is fixed at 200 m. The distance between adjacent rows is 5 m.

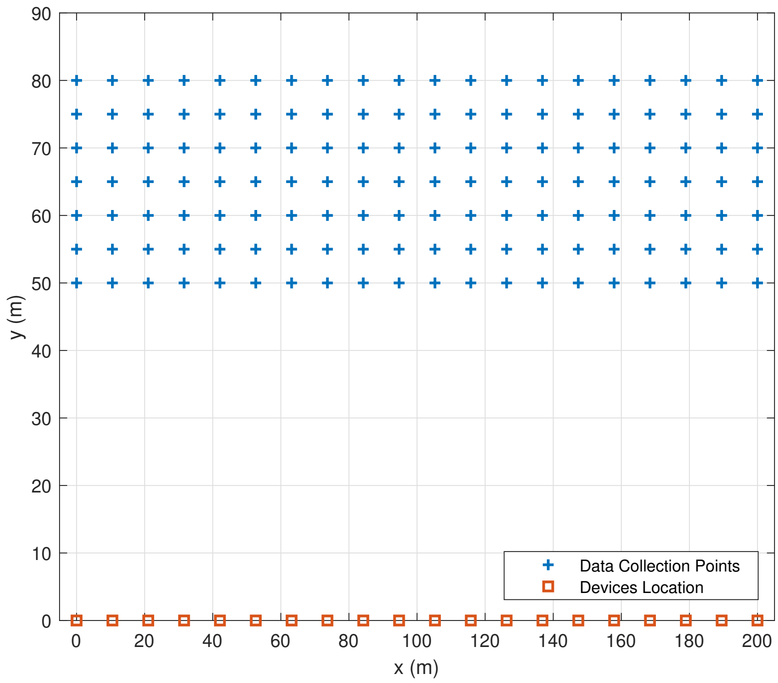

Figure 4.3: The system settings when devices have zero elevation.

4.5.1 Zero Elevation

The first set of experiments assume devices have zero elevation. It studies the effect of four parameters, namely the deployment range D, the height of the UAV, the number of devices $|G|$, and the number of columns $|M|$ over the grid, on the

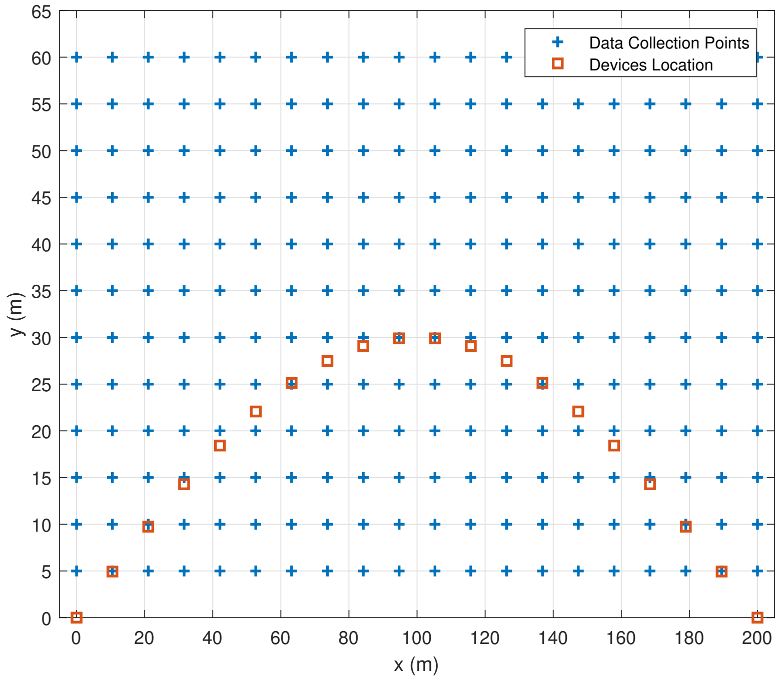

Figure 4.4: The system settings for elevated devices network.

sum-rate and/or the average throughput and the number of simultaneous active devices. These experiments also study how different environments, namely suburban, urban and dense urban, impact the probability of LoS/NLoS as well as the average throughput.

4.5.1.1 UAV Height and Deployment Range of Devices

The experiments in this section study the impact of the UAV height on the number of simultaneous active devices and the sum-rate. The UAV is assumed to fly at a fixed height and collect data from devices at 20 data collection points. The height of the UAV increases from 30 to 1000 meter. The experiments first consider a fixed deployment range D of 200 meter and consider the following number of devices $|G|$: 5, 10 and 20. After that the experiments use ten devices and varies the deployment range D from 200 to 1000 meter. Devices and UAV data collection points are spaced equally within the deployment range D. This experiment also assumes that the UAV has infinite energy budget in this experiment.

Figure 4.5 shows that when the UAV height is higher than 180 meter, the number of simultaneous active devices of all cases with different number of devices reduces from two to one. This is because the channel gain of uplinks decreases with increasing UAV height. Thus, the difference in receive power of uplinks reduces. Therefore, the number of simultaneous devices that can satisfy SIC constraints and transmit together decreases.

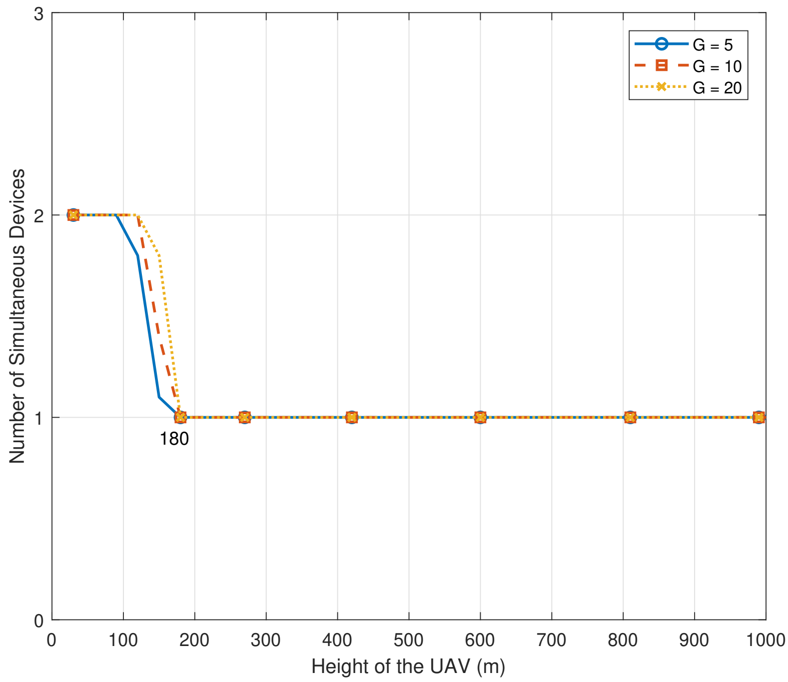

Figure 4.5: Simultaneous devices versus different heights of the UAV.

Figure 4.6 shows the relationship between the sum-rate and the height of the UAV as well as the deployment range D of devices. First, we see that when the height of the UAV is less than 180 meter, the deployment range D will affect the total collected data. Specifically, a larger D value results in a smaller sum-rate. For example, the sum-rate when D is set to 1000 meter is 462 Mbits less than that of the case when D equals to 200 m. This is because SIC takes effect when the height of the UAV is less than 180 meter. When we increase the deployment range, the distance between uniformly spaced devices increases correspondingly. Thus, the decrease in the SINR and/or SNR of uplinks from simultaneous active devices will result in smaller data rates. Therefore, the UAV will collect less data than the case with a smaller deployment range. We also observe that the sum-rate decreases with the trend when the UAV is at a higher height. In particular, the decreasing trend resembles a step function. This is because the UAV has a finite number of data rates. If the SINR/SNR is within a given range, the UAV transmits with the same data rate. For example, if the SNR of an uplink is no less than 22 dB, as per [166], the data rate will be 54 Mbps. Referring to Figure 4.5, we observe that when the UAV height is higher than 180 meter, only one uplink can transmit at each data collection point. Thus, the sum-rate is directly proportional to the data rate.

We therefore conclude that SIC takes effect when the height of the UAV is less than 180 meter. In addition, the UAV collects more data when the deployment range D is set to 200 meter. Thus, in all subsequent experiments, the height range of the grid and the deployment range D are designed to be less than 180 meter and 200 meter, respectively.

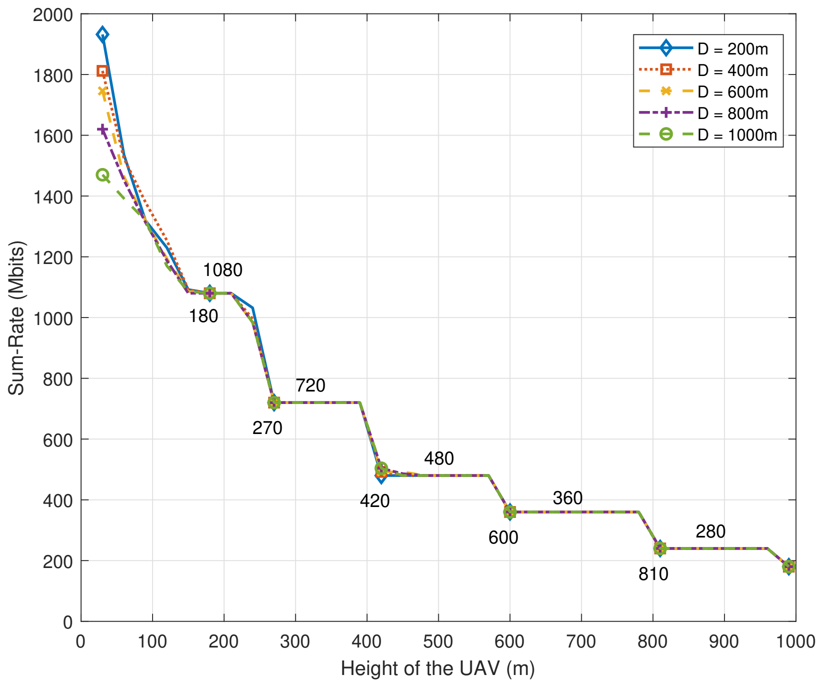

Figure 4.6: Relationship between UAV height and devices deployment range.

4.5.1.2 Environmental Impacts

In this section, the experiments study the impact of different environments on the average throughput. The propagation model of [170] models suburban, urban and dense urban scenarios. The number of devices $|G|$ increases from four to 20. We set $|N| = 6$, $|M| = 20$, $h = 50$ m, $\hat{s} = 5$ m, and $D = 200$ m.

Figure 4.7 shows that for any $|G|$ values, the suburban environment yields the highest average throughput of each device. For example, when the number of devices $|G|$ is ten, the average throughput for suburban is 8.73 Mbps. However, the average throughput of urban and dense urban is 7.87 and 7.49 Mbps, respectively. This is because compared to other two environments, the attenuation from shadowing and scattering in suburban is small. Therefore, the path loss calculated in Eq. (4.3) will be smaller correspondingly. Consequently, active device(s) will yield a higher data rate.

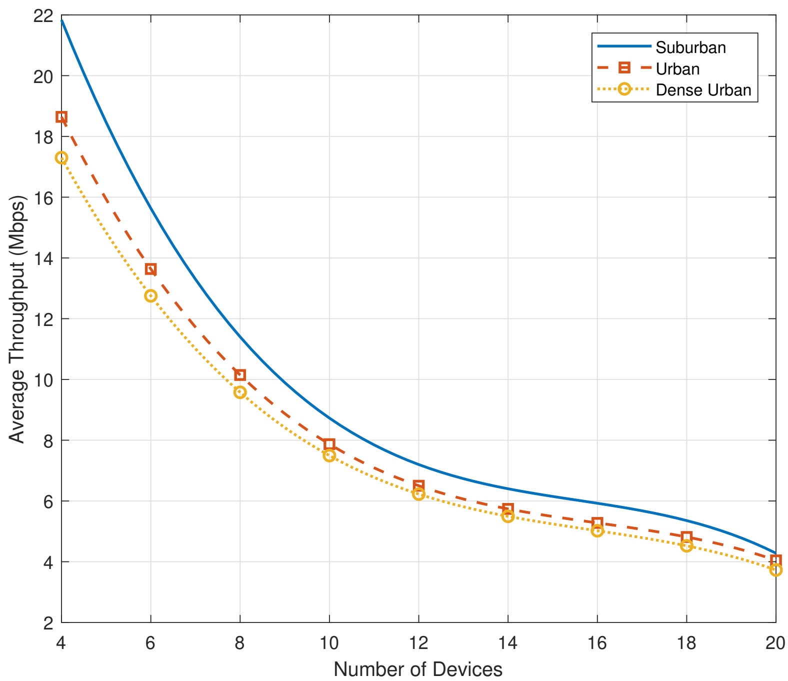

Figure 4.7: Average throughput for different environments.

4.5.1.3 Devices Placement Methods

Next, the experiments in this section study the impact of the number of devices $|G|$ as well as the number of columns $|M|$. In particular, the value of $|G|$ and $|M|$ is increased from four to 20 and 10 to 20, respectively. Moreover, the experiments also investigate the impact of different methods to place devices. They include,

- *Uniform*. Devices are uniformly located within the deployment range. In particular, the distance between adjacent devices are fixed.
- *Cluster*. Devices are divided into multiple groups. Each group has the same number of devices that are located closely to each other. Groups are spaced equally within the deployment range.
- *Random*. Devices are randomly located within the deployment range.
- *Poisson Point Process (PPP)* [173]. We place devices as per a PPP with average density λ.

In the experiment of Section 4.5.1.3, the given grid has six rows and the distance between adjacent rows is set to 5 m. The lowest row in the grid is 50 m higher than devices. The average density λ in PPP is set to two. The results in Section 4.5.1.3 are an average of 100 simulation runs.

Figure 4.8 shows the average throughput with increasing number of devices $|G|$ when the number of columns M is fixed at 20. In particular, the value of $|G|$ increases from four to 20. We see that with increasing number of devices, the average throughput of all four cases with different node position methods gradually becomes smaller. Specifically, the decrease in average throughput is 14.7, 12, 13.68 and 13.35 Mbps, respectively. The reason is because the number of simultaneous active devices is bounded by SIC's decoding limit. Thus, even when the number of devices increases, the total transmitted data will not increase proportionally.

Referring to Figure 4.8, we also observe that when the number of devices increases, the average throughput difference between all four node position methods decreases. For example, when there are only four devices, placing devices uniformly yields the highest average throughput at 18.9 Mbps. However, when devices are placed in a cluster, the average throughput is the lowest that is 16.2 Mbps. This is because under this case, devices are divided into two groups and each group consists of two devices. These groups are respectively located at the left and right side within the deployment range D. Therefore, compared to the case when devices are placed uniformly, devices will have long transmission distance for most UAV data collection points; thus, the receive power will be small. Consequently, the SNR and/or SINR value and the corresponding data rate will be small, as well. However, when we increase the number of devices, the location of devices in each node position method becomes more uniform. Therefore, the data rate of active devices as well as the average throughput become similar for all four methods.

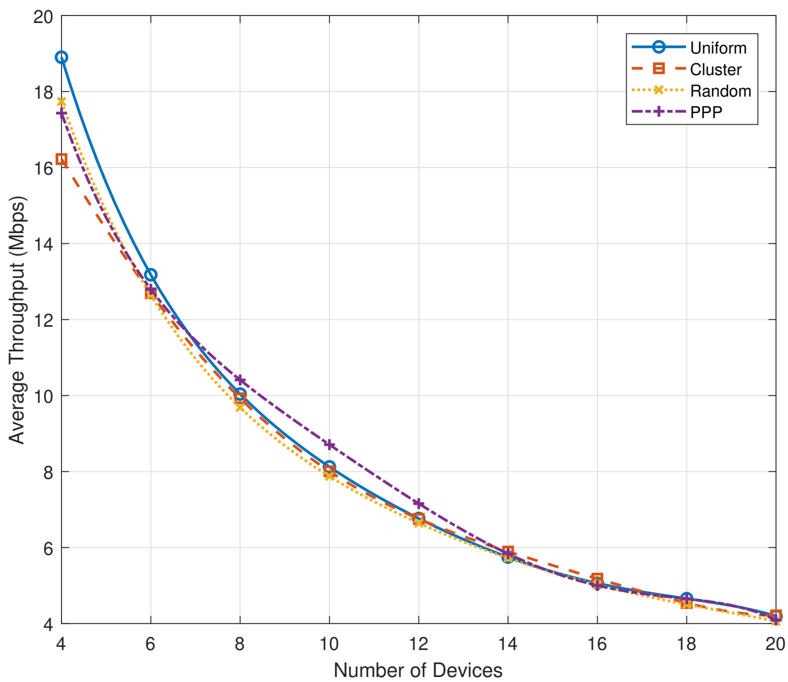

Figure 4.8: Average throughput versus different number of devices when devices have no elevation.

Next, Section 4.5.1.3 studies the effect of devices placement methods when the number of columns $|M|$ is increased from ten to 20. The number of devices $|G|$ is fixed at 20. The number of candidate data collection points increases with $|M|$

because each column can only have one active data collection point.

Figure 4.9 shows that the sum-rate increases linearly with more candidate data collection points. Specifically, when the number of candidate data collection points increases from ten to 20, the increase in sum-rate of all four node position methods is 840, 840, 818.73 and 818.29 Mbits, respectively. This means with a new added data collection point, the sum-rate will increase around 80 Mbits for all methods. The reason is that for all candidate data collection points, the number of devices that can transmit together is fixed at two. Additionally, the formulated ILP will always select the link set with the highest sum-rate for each data collection point. Thus, the collected data at each candidate point is similar.

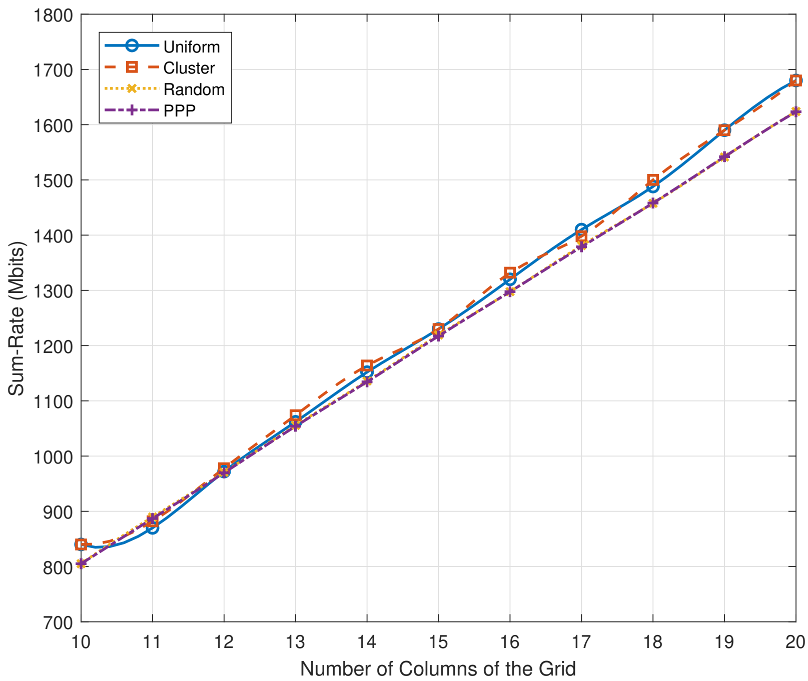

Figure 4.9: Total transmitted data versus different number of columns when devices have no elevation.

4.5.1.4 Heuristic Algorithms

This experiment studies the impact of different link selection policies that are used by ICLST. Besides HSRS, this experiment also considers the following policies:

- *Most Active Devices Selection (MADS).* The link set with the most number of devices will be selected at each data collection point.

- *Random Link Set Selection (RLSS).* A random link set will be selected at each data collection point.

The grid is set to have 20 columns and six rows. The number of devices $|G|$ increases from four to 20. The results in Section 4.5.1.4 are obtained from an average of 100 simulation runs.

Figure 4.10 shows the average number of simultaneous active devices for each link selection policy. We see that when we increase the number of devices, the average simultaneous active device for HSRS and RLSS is around two, respectively. This is because the proposed ILP, see Eq. (4.12), will select candidate data collection points with the highest sum-rate. Thus, active link sets have fewer devices. This results in a higher SINR and/or SNR value and data rate. However, for MADS, the average number of simultaneous active devices increases from three to four, which is the maximum number of simultaneous devices that the UAV can decode, i.e., L_{max}. The reason is that instead of selecting link sets with the highest sum-rate, MADS prefers to select link sets with the most number of devices.

Referring to Figure 4.11, we observe that HSRS yields the highest average throughput between three link selection policies. Specifically, when we increase the number of devices from four to 20, the average throughput difference between HSRS and RLSS or MADS is around 1.55 Mbps and 2.96 Mbps, respectively. This is because in HSRS, the SNR and/or SINR of active uplinks is higher; thus, the data rate as well as the sum-rate will be bigger, correspondingly. However, in MADS, the UAV needs to fly higher to allow more devices to transmit together at each data collection point. Therefore, the resulting longer transmission distance results in uplinks having a correspondingly smaller data rate.

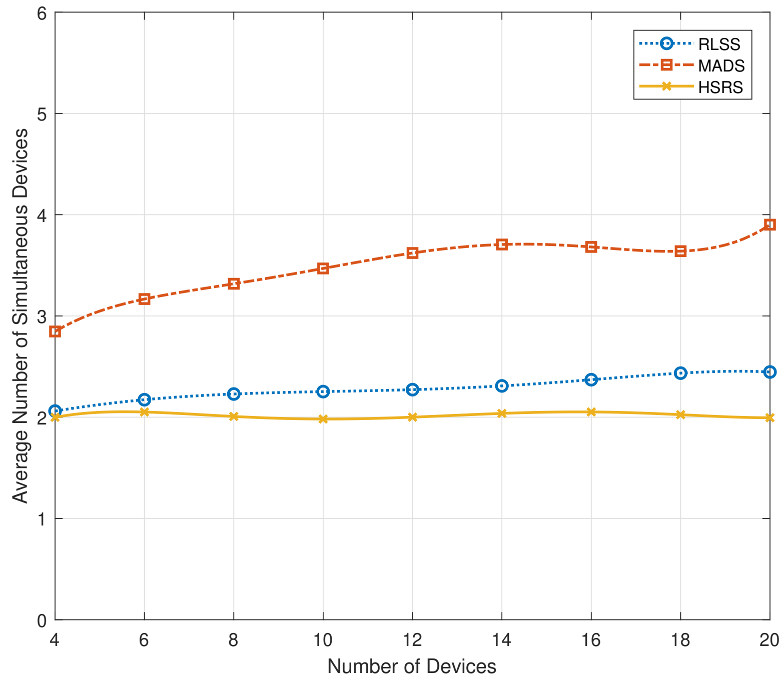

Figure 4.10: Simultaneous devices versus number of devices for different link selection policies.

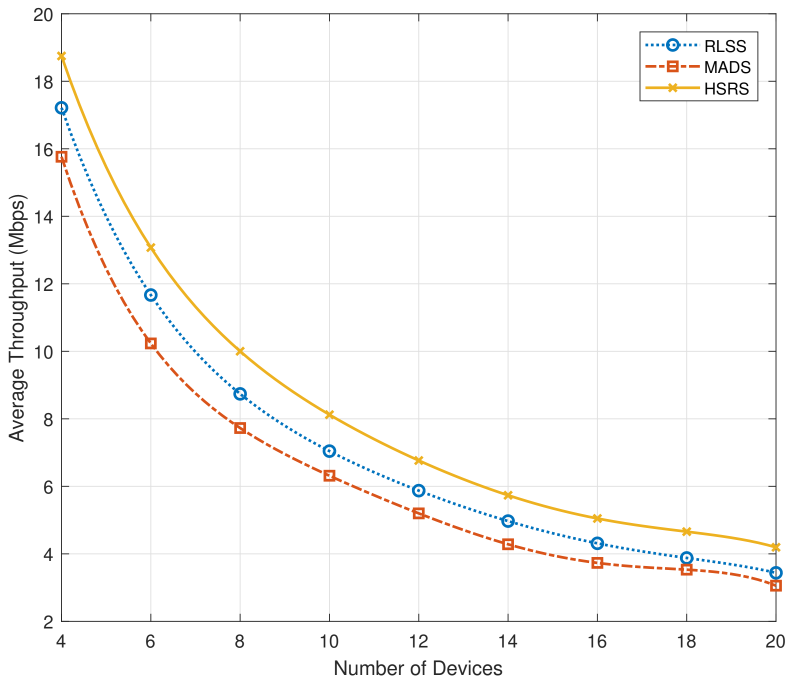

Figure 4.11: Average throughput versus number of devices for different link selection policies.

4.5.2 Varying Elevation

Devices are located at different elevated heights. This section studies the average throughput, the total energy consumption and the UAV trajectory. Specifically, the height of devices is distributed sinusoidally. This experiment changes the amplitude H_g and the period F_g of the sinusoid.

4.5.2.1 Fixed UAV Height

The UAV flies at a fixed height that is set to 50 m. The value of F_g is first fixed to 0.5 and the amplitude H_g is varied from zero to 30 m. Figure 4.12 is an example that shows the trajectory of the UAV as well as the location of devices for different amplitude values. As shown in Figure 4.12, there are 20 uniformly located devices and 20 data collection points along the UAV trajectory.

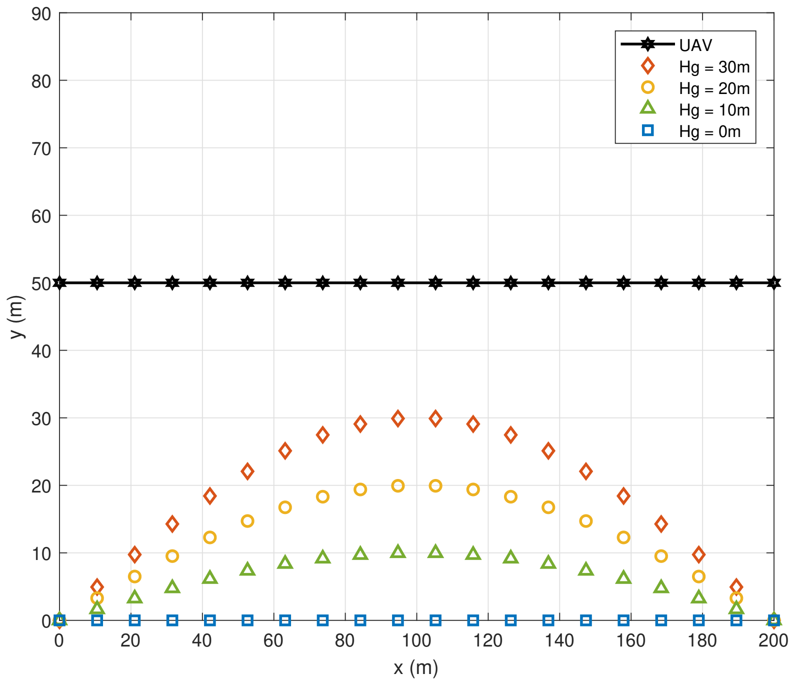

Figure 4.12: The UAV's trajectory and height distribution of devices with different amplitude values.

Figure 4.13 shows the impact of various H_g values on the average throughput. We also increase the number of devices from four to 20. Referring to Figure 4.13, we observe that the average throughput increases by 1 Mbps when the amplitude is ten meters higher. This is because the shorter transmission distance results in uplinks having a corresponding higher data rate. In all subsequent experiments, we will set the amplitude H_g of devices' height distribution to 30 m.

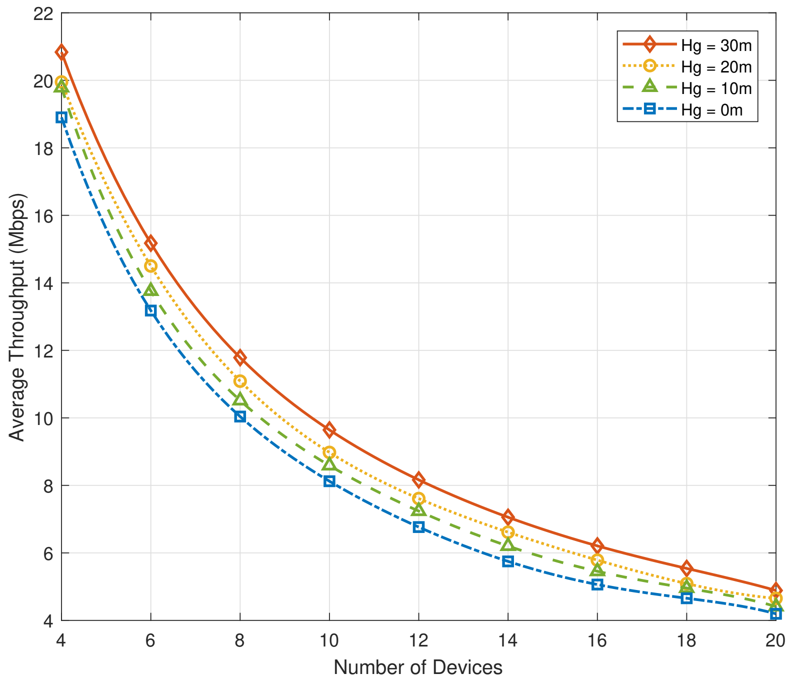

Figure 4.13: Average throughput versus number of devices for different height values.

The next experiment studies the impact of different methods to place devices, such as *Uniform*, *Cluster*, *Random* and *PPP*, see Section 4.5.1.3 for details. The number of devices $|G|$ is increased from four to 20 within the deployment range D. Referring to Figure 4.14, we see that the average throughput for all four methods decreases with increasing number of devices. In particular, when $|G|$ is four, the average throughput of four methods is 20.81, 15.75, 19.04 and 18.41 Mbps, respectively. However, when $|G|$ increases to 20, the throughput decreases to 4.89, 4.9, 4.85 and 5 Mbps, respectively. The reason is because the SIC decoding limit bounds the number of simultaneous active devices; thus, the total transmitted data will not increase continuously. We also observe that the average throughput difference between all proposed node position methods, namely *Uniform*, *Cluster*, *Random* and

PPP, becomes smaller when the number of devices increases. This is because the deployment range D is fixed; thus, when there are more devices within D, devices will be located closer together. This is true for all devices placement methods. Consequently, the transmission distance of active uplinks and corresponding data rate for all methods are similar. In all subsequent experiments, we will place devices uniformly.

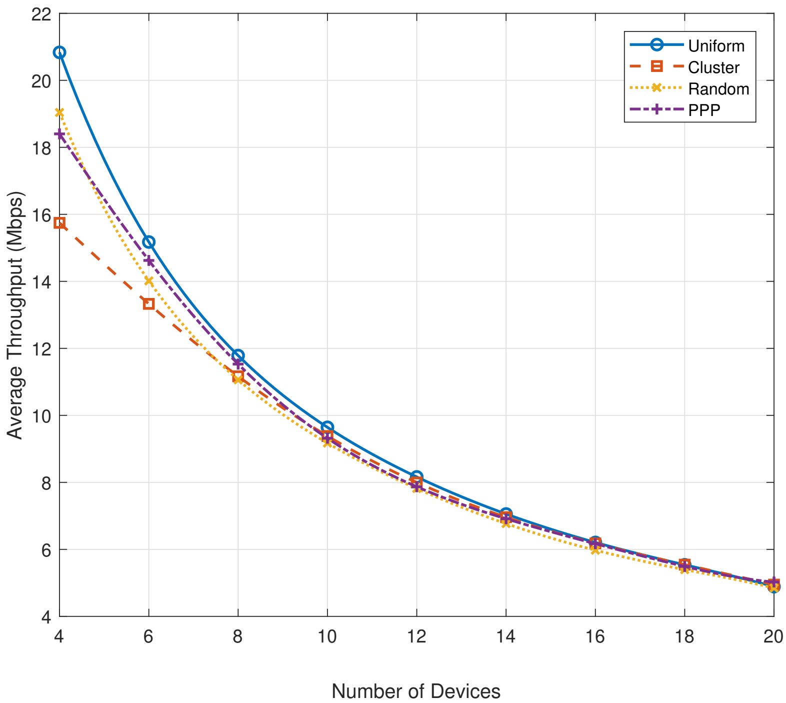

Figure 4.14: Average throughput versus number of devices for different node position methods.

4.5.2.2 Dynamic UAV Height

In this experiment, the UAV selects the optimal trajectory via Eq. 4.11 to collect data. The UAV collects data only when its height is higher than devices. For each data collection point, we denote the minimum distance between active devices and the UAV as d_{min}. The value of d_{min} is set as follows: 5, 10, 20 and 30 meter. The elevated height of devices is set according to a sinusoid with period F_g. This means the number of cycles within the fixed deployment range D changes with various F_g values. In particular, a larger F_g value leads to a higher frequency and more cycles within the range D. Therefore, the difference in the height of devices increases in accordance with F_g. The following experiment considers three F_g values: 0.5, 1.0 and 1.5. It also considers setting the height of devices randomly to a value in the range $[0, 30]$ (in meter). Lastly, we have $|N| = 15$, $|M| = 20$, $\hat{s} = h = 5$ m, $H_g = 30$ m, and $D = 200$ m.

In Figure 4.15, we see that the average throughput increases by 0.5 Mbps when the value of d_{min} decreases from 30 to 5 meter. In particular, when the number of devices $|G|$ is set to 20, the average throughput under various d_{min} cases is 6.35, 5.82, 5.35 and 4.72 Mbps, respectively. This is because when d_{min} increases, the longer transmission distance between devices and the UAV results in a higher path loss as well as a smaller data rate.

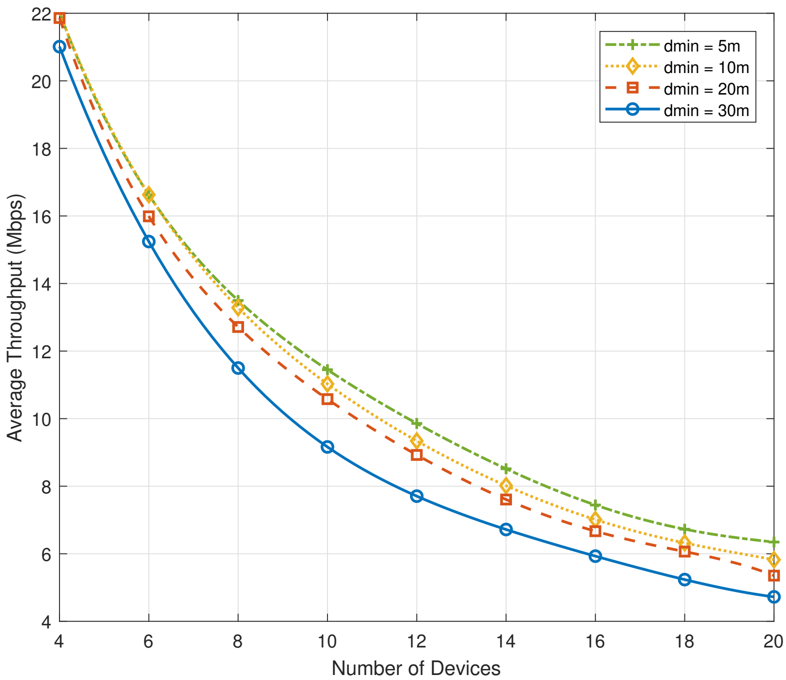

Figure 4.15: Impact of minimum distance between devices and the UAV - average throughput versus devices numbers.

Referring to Figure 4.16, we notice that the total energy consumption decreases with a longer distance between the UAV and devices. Specifically, when there are ten devices, the total consumed energy is 3630 J, 3087 J, 2869 J and 2829 J, respectively.

This is because when d_{min} is small, the UAV will consume more energy to fly higher to obtain better LoS uplinks.

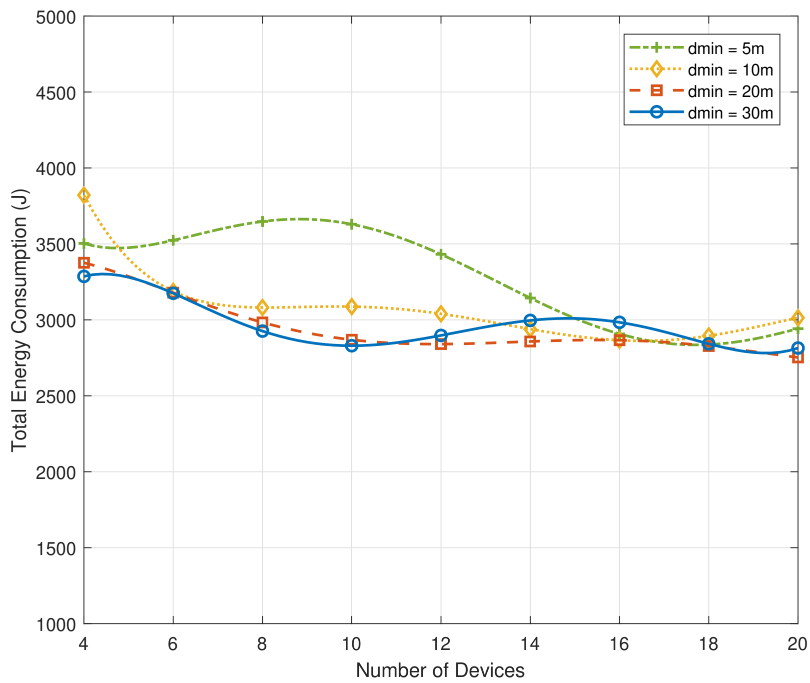

Figure 4.16: Impact of minimum distance between devices and the UAV - energy consumption versus devices numbers.

Figure 4.17 shows the optimal trajectory for various d_{min} cases when $|G|$ is ten. In particular, the optimal trajectory is obtained via Eq. (4.11). We see that when d_{min} is set to 5 m, the vertically ascent length along the UAV trajectory is around 59 m. However, when d_{min} is 30 m, the vertical length is 33 m shorter than that of the case when $d_{min} = 5$ m. Thus, compared to the case when d_{min} is 30 m, the propulsion energy for the case with $d_{min} = 5$ m is 801 J higher. Hence, larger d_{min} values lead to lower energy consumption by the UAV.

From Figure 4.15 and 4.16, we see that when d_{min} equals 20 m, the UAV consumes the least energy. The average throughput of this case is similar to the case when d_{min} is set to 5 m. Therefore, the subsequent experiments will use the minimum distance d_{min} of 20 m.

This experiment studies how the height of devices affects the average throughput

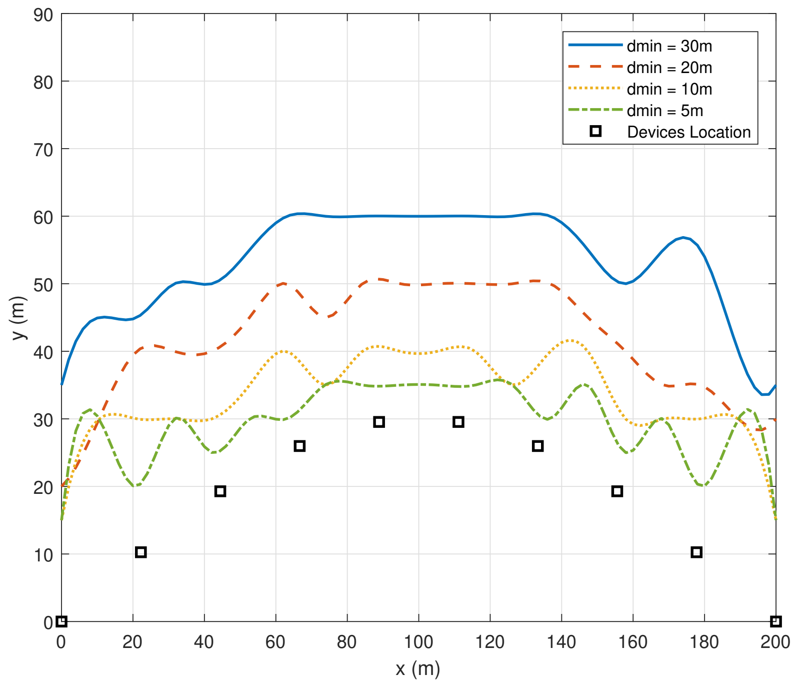

Figure 4.17: Impact of minimum distance between devices and the UAV - UAV trajectory when there are ten devices.

and consumed propulsion energy. Referring to Figure 4.18, we see that for any number of devices, the average throughput is similar for all F_g values. When the number of devices increases from four to 20, the average throughput of all four cases with various F_g values decreases from 22 Mbps to 5.4 Mbps, respectively. This is because the UAV prefers to select candidate data collection points that are close to devices to obtain a higher data rate. Therefore, the transmission distance of active devices and corresponding date rate are similar for all F_g values cases.

From Figure 4.19, we see that the UAV will consume more energy when F_g increases in value. For example, when there are ten devices, the total energy consumption under various F_g cases is 2862, 2959 and 3270, respectively. When the height of devices is randomly set in the range $[0, 30]$ m, the total energy consumption is 3200 J when $|G|$ is ten. The reason is because when F_g becomes larger, the frequency of the sinusoid increases; thus, the difference in the height of devices increases. Additionally, the UAV prefers to fly closer to devices. This helps de-

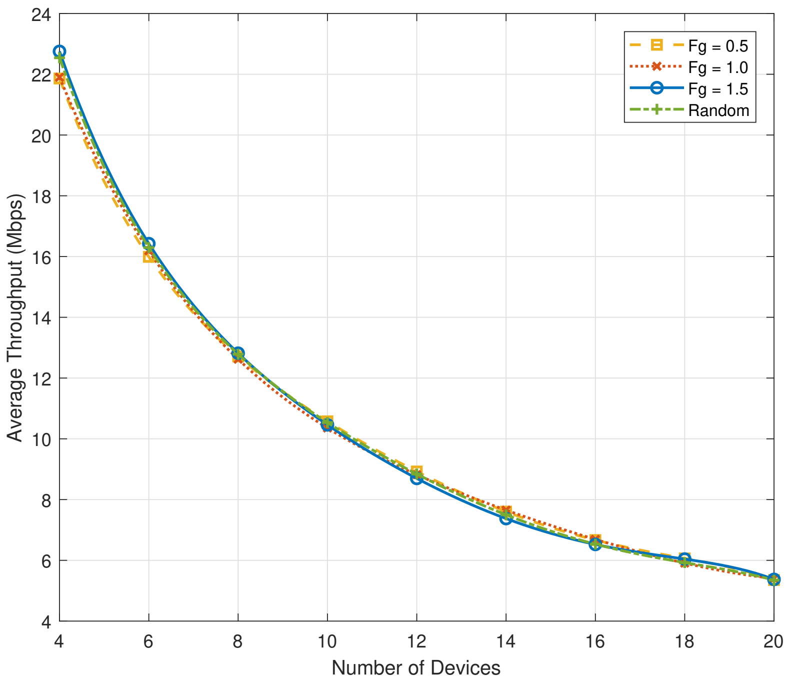

Figure 4.18: Impact of the period of devices' height distribution - average throughput versus devices numbers.

crease path loss and select uplinks with a higher data rate. Therefore, a larger F_g results in the UAV changing its height frequently, which results in a higher energy consumption rate.

Figure 4.20 shows the obtained optimal trajectory for various F_g values when $|G|$ is 20. We observe that the height change along the UAV trajectory follows the height distribution of devices. In particular, the vertically ascent length along the UAV trajectory under various F_g cases is 20, 30, 35 and 39 m, respectively. Therefore, compared to the case when $F_g = 0.5$, the propulsion energy for the case with $F_g = 1.5$ is 250 J higher. Hence, larger F_g value results in higher energy consumption by the UAV. We then conclude that in all subsequent experiments, the period F_g is set to 0.5.

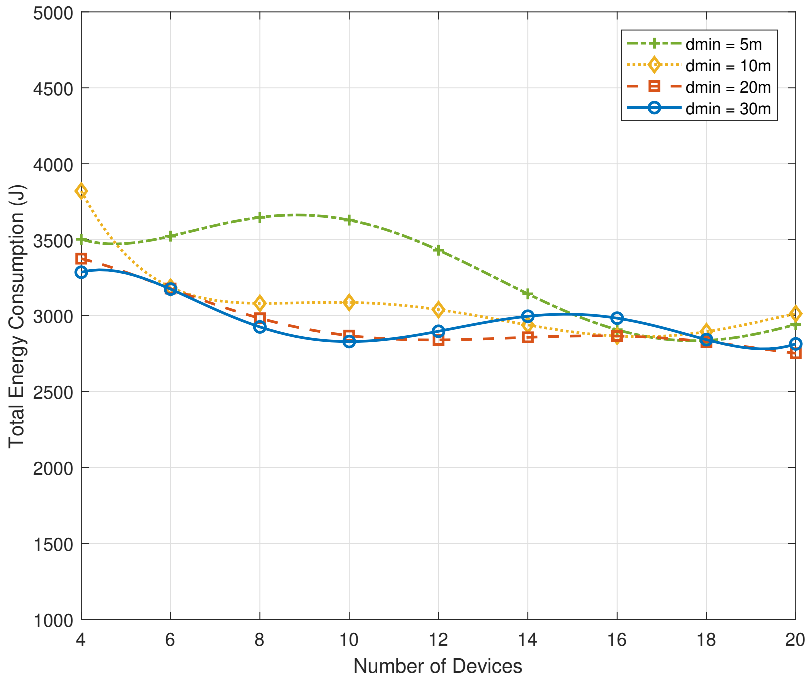

Figure 4.19: Impact of minimum distance between devices and the UAV - energy consumption versus devices numbers.

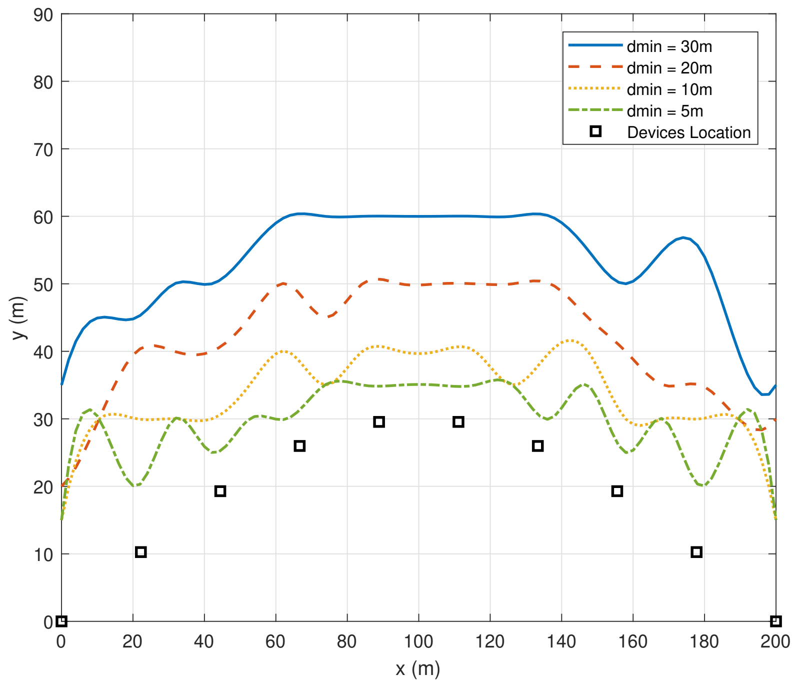

Figure 4.20: Impact of minimum distance between devices and the UAV - UAV trajectory when there are 20 devices.

4.5.3 SIC-ILP versus other Methods

This experiment compares the results of SIC-ILP with HSRS, SARSA-based protocol and TDMA under the following cases: (i) devices with zero elevation (DZE), and (ii) devices with elevated height (DEH). In particular, it compares the average throughput and the total consumed propulsion energy for each method under cases DZE and DEH. Note, for TDMA, there is only one transmitting device in each time slot. Lastly, experiments to follow set $|M| = 20$, $\hat{s} = h = 5$ m, $H_g = 30$ m, $d_{min} = 20$ m, $F_g = 0.5$, and $D = 200$ m. The number of rows $|N|$ for cases DZE and DEH is set to 6 and 15, respectively. The experiments study four to 20 uniformly located devices.

From Figure 4.21, SIC-ILP has the best performance. This is because the UAV uses the optimal trajectory that yields the highest sum-rate; also, each device has the maximal average throughput. For example, for DEH, the average throughput of SIC-ILP is 5.35 Mbps when $|G|$ is 20. However, HSRS and SARSA-based protocol achieve 5.29 Mbps and 5.25 Mbps, respectively. The average throughput of TDMA is only 2.69 Mbps. We also observe that the average throughput of SIC-ILP, HSRS and SARSA-based protocol is twice that of TDMA. The reason is because SIC allows multiple simultaneous transmissions.

Referring to Figure 4.21, we notice that when the height of devices is in the range $[0, 30]$ (in meter), the average throughput of each device increases. In particular, for any $|G|$ values, the average throughput for DEH is 1.5 Mbps higher than DZE. The reason is because the transmission distance between devices and the UAV decreases; thus, active uplinks will have a higher SINR and/or SNR value and data rate. We also observe that compared to the case when the UAV trajectory is fixed at 50 m, the UAV flies in the range $[30, 50]$ m will yield a 0.5 Mbps higher throughput. This is because the UAV is able to change its height to select data collection points that consist of uplinks with better channel condition.

Figure 4.22 and 4.23 show the total consumed propulsion energy of the UAV for

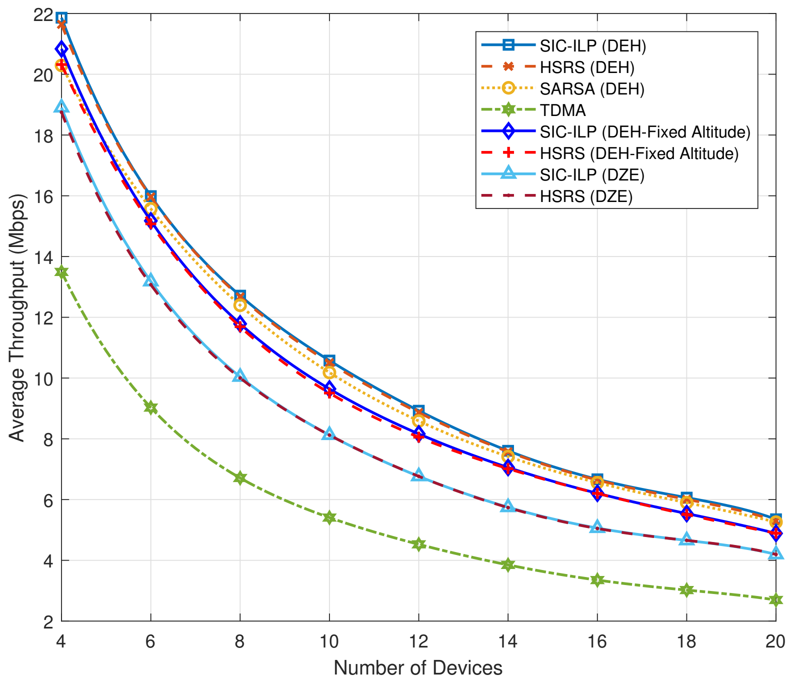

Figure 4.21: Average throughput comparison.

DZE and *DEH*, respectively. Note that in both settings, the UAV flies at different heights that are obtained via Eq. (4.11) and/or (4.12).

Referring to Figure 4.22, for TDMA, the energy consumption of the UAV is the highest on average. In particular, its consumed energy fluctuates around 3250 J. This is because there is only one active uplink at each data collection point; thus, the simplified ILP, see Eq. (4.12), will select candidate data collection points with uplinks that have the highest data rate. Therefore, the UAV will fly to a high height to select uplinks with higher LoS probability and better channel condition. We see that the total energy consumption of the SARSA-based protocol is fixed at 2350 J for any number of devices. The reason is that the UAV learns the best trajectory that yields the highest energy efficiency via SARSA, see Algorithm 8. Therefore, when the height of devices is set to zero, the learned trajectory is the closest to devices. Thus, the energy consumption is the minimum and active uplinks will yield the highest data rate. We observe that the energy consumption for SIC-ILP fluctuates significantly in the range $[2500, 3560]$ J. In particular, when the number of devices is set to five, ten, and 15, the consumed propulsion energy reduces to less than 2500 J. This is because devices are uniformly located within the deployment range D. When there are 20 columns in the grid, devices are placed directly below certain columns. For example, when $|G|$ is five, the x-axis of four devices is the same as columns $\{1, 6, 15, 20\}$. Additionally, the other device's location is in the middle of columns 10 and 11. Thus, for these columns, the selected data collection point is the nearest from devices. Therefore, active uplinks will have the best channel condition and the UAV will consume 1000 J less propulsion energy.

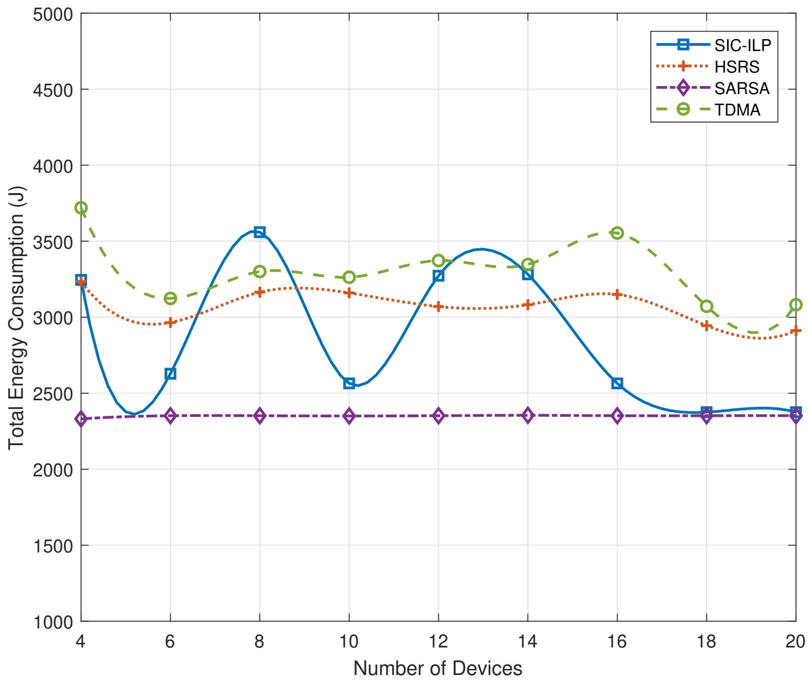

Figure 4.22: Total energy consumption comparison when devices have zero height.

Referring to Figure 4.23, we observe that when the height of devices is in the range $[0, 30]$ (in meter), the UAV consumes less propulsion energy. In particular, for any number of devices, the UAV's energy consumption is around and/or less than 3000 J. The reason is because the transmission distance between devices and the UAV decreases; thus, the path loss of uplinks decreases. Therefore, the UAV does not need to fly to a high height to select uplinks with higher LoS probability and better channel condition. We also see that the total energy consumption of the

UAV reduces with increasing number of devices. This is because the UAV has a higher probability to find a nearby device that can transmit with the highest data rate. Therefore, the UAV will select a low-height trajectory that yields less energy consumption.

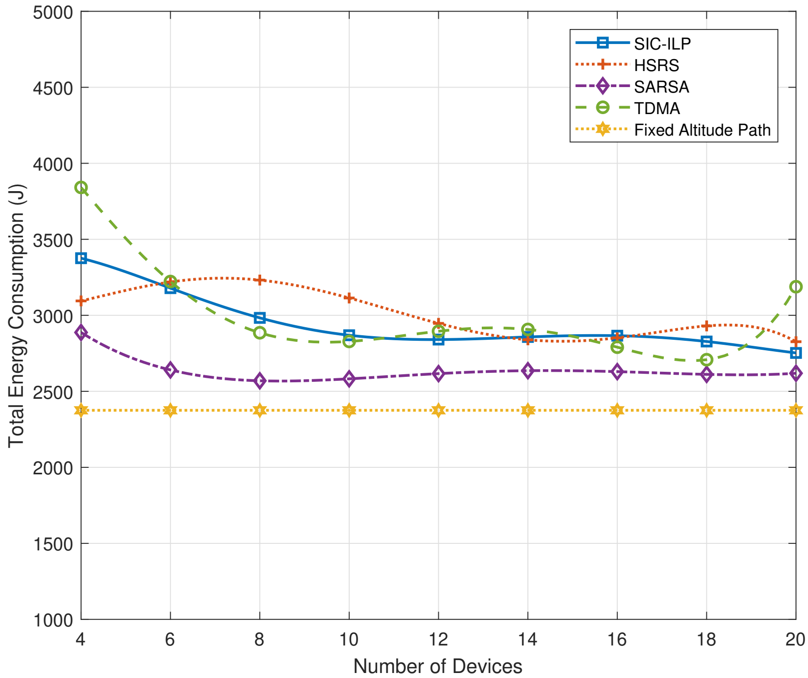

Figure 4.23: A comparison of total energy consumption when devices have an elevated height.

4.6 Conclusion

This chapter considers data collection using a SIC-capable UAV. The problem at hand is to optimize the UAV trajectory or data collection locations and also the the uplink schedule at each of these locations. To this end, Chapter 4 proposes three novel solutions that include an ILP, a heuristic algorithm called ICLST and a SARSA-based distributed protocol. Numerical results indicate that equipping a UAV with a SIC radio doubles the amount of uploaded data. Moreover, different heights help the UAV collects more data. They also show that when devices are at an elevated height, a shorter transmission distance between devices and the UAV results in uplinks with a higher data rate. The average throughput of each ground device is affected by the size of the grid as well as the position of devices.

A limitation of the work in Chapter 3 and Chapter 4 is that the UAV does not have a connection with a sink or a gateway. However, the limited data storage of the UAV constrains the total amount of collected data. In addition, the coverage range of a single UAV is limited. These limitations are addressed in Chapter 5, which considers data collection in Space-Air-Ground Integrated Networks (SAGINs).

Chapter 5

Data Collection in SAGINs

Thus far, the network in Chapter 3 and 4 considers a UAV without connection to a sink or gateway. Different from previous chapters, this chapter considers data collection in a SAGIN comprising of a CubeSat swarm, a SIC-enabled rotary-wing UAV, multiple ground devices and a gateway. Specifically, these ground devices send their data to the gateway via the UAV, which either carries the data back to the gateway or uploads it to a CubeSat. The objective in this chapter is to maximize the minimum flow of ground devices over a planning time horizon. Specifically, in each time slot, the problem at hand is to determine (i) a route/path between a UAV and the gateway, where a route may consist of a UAV-CubeSat link, inter-satellite link(s), and a CubeSat-gateway link, (ii) an uplink transmission schedule from ground devices to the UAV, and (iii) the amount of data to be forwarded over each active link.

Figure 5.1 illustrates the said objective over a SAGIN depicted as a time-varying graph. It has three data transmission paths that originate from each time slot; namely, $u \to s_1 \to s_2 \to GW$, $u \to s_2 \to GW$ and $u \to GW$. The path between a node, e.g., UAV or a CubeSat, to itself from time t to $t+1$ indicates that the node carries the same data from time t to $t+1$. For example, for path $u \to s_1 \to s_2 \to GW$, UAV u collects data from ground device g_1 and g_3 in the first time slot.

It then carries the collected data and offloads it to CubeSat s_1 in time slot $t = 2$. CubeSat s_1 then forwards the data to its neighbor CubeSat s_2 before downloading it to gateway GW.

There are a number of issues to consider. First, when selecting a path, we need to guarantee that the data collected from ground devices will arrive at the gateway. Note that the number of paths to the gateway increases exponentially with increasing number of CubeSats. Moreover, the space network is comprised of a time-varying network topology. This means a UAV or CubeSat may not have end-to-end connectivity to the gateway in a given time slot. Hence, the UAV and CubeSats need to determine whether to carry data to a future time slot or offload some data to a CubeSat. Second, we need to check the available resources, i.e., storage and link capacity, of every node on the path. For example, we see that in time slot 4 of Figure 5.1, paths $u \rightarrow s_1 \rightarrow s_2 \rightarrow GW$ and $u \rightarrow s_2 \rightarrow GW$ use the same downlink between CubeSat s_2 and GW. This means s_2 and its downlink must have sufficient capacity to support the data from time slot 1 and 2. Third, in each time slot, the UAV needs to schedule multiple ground devices and corresponding uplink to make full use of the capacity of a selected path. A challenging aspect is that the number of ground devices combinations that satisfy SIC constraints increases exponentially with the number of ground devices. Specifically, if there are $|\mathcal{G}|$ ground devices, then there are $2^{|\mathcal{G}|} - 1$ possible combinations of uplink transmissions. For example, in Figure 5.1, we have up to seven possible ground devices combinations in each time slot.

This chapter contains the following contributions:

- It considers two approaches to maximize the minimum flow of ground devices over multiple time slots. First, a novel MILP is outlined to (i) select the optimal path between the UAV and gateway, (ii) schedule ground devices subject to these devices satisfying constraints relating to SIC, and (iii) determine the amount of data to upload from scheduled ground devices. Second, this chapter outlines a novel distributed algorithm called Iterative Flow and Path Reser-

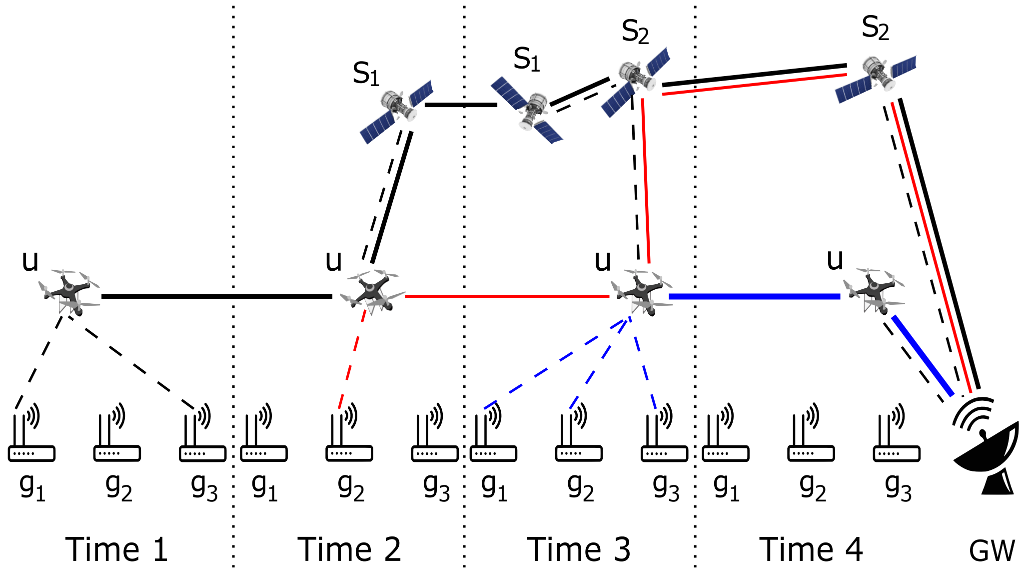

Figure 5.1: A time-varying graph for an example SAGIN. There are three ground devices , two CubeSats , a single UAV and a gateway . A dotted line/link between nodes indicate a communication channel. Different colors indicate path and links selected in different time slots. The thickness of paths indicates the amount of data forwarded on different paths.

vation (IFPR). The UAV independently selects a path with fewest number of hops to the gateway over time in each time slot. Additionally, the UAV also considers randomly selecting a path for each time slot. In addition, it also schedules ground devices and their individual uploaded data. To schedule ground devices, the UAV considers two methods: (i) a simplified MILP called IFPR-SMILP, and (ii) a greedy algorithm named IFPR-LDSF. Specifically, in each time slot, IFPR-SMILP schedules a set of ground devices with the highest sum-rate. As a comparison, IFPR-LDSF prefers to greedily schedule ground devices that have uploaded the least amount of data in past time slots.

- It presents the first study of the proposed approaches, and presents the following findings: (i) compared to one-tier UAV communications, CubeSats help increase the total amount of collected data by 61%, (ii) compared to a single CubeSat case, gateway receives 63.6% additional data that is transferred over ISLs, (iii) Jain's Fairness index increases with more time slots. Additionally, all ground devices have equal opportunities to be scheduled when the number of time slots is sufficiently large, and (iv) applying IFPR-SMILP to schedule ground devices yields a higher amount of collected data but at the expense of fairness.

The remainder of this chapter is structured as follows. The network setup and notations are introduced in Section 5.1. Section 5.2 presents the formulated MILP model. Section 5.3 presents a novel distributed algorithm IFPR. Its evaluation is presented in Section 5.4. This chapter concludes in Section 5.5.

5.1 Network Model

This section first formalizes the SAGIN under consideration; see Section 5.1.1 and Section 5.1.2, respectively. After that, Section 5.1.3 to 5.1.6 respectively discuss link activation, data transmission and/or collection, data storage and routing.

5.1.1 Preliminaries

Table 5.1 summarizes the nomenclature used in this chapter. Next, this section presents the considered model for (i) SAGIN, (ii) CubeSats, (iii) UAV, and (iv) time-varying network topology.

5.1.1.1 SAGIN Architecture

The SAGIN under consideration has three layers: ground, aerial and space. The ground layer consists of devices and a gateway. The aerial layer has a single rotary-wing UAV. The space layer consists of a swarm of Low Earth Orbit (LEO) CubeSats. The planning time horizon T is divided into time slots and indexed by $t \in T$. Let w and u denote the gateway and the UAV, respectively. Let S and $\mathcal{G}$ denote the set of all CubeSats and ground devices. Define S^t and $\mathcal{G}^t$ as a set of CubeSats and ground devices at time t, respectively. For ground devices, we use $g_j \in \mathcal{G}^t$. A CubeSat is indexed as $s_i \in S^t$. Define $N^t_{s_i} \subseteq S^t$ as the set of neighbors of CubeSat s_i in slot t,

Symbol	Description
1.	**Sets**
T	Fixed time horizon $t \in T$
S	Set of CubeSats
$\mathcal{G}$	Set of ground devices
S^t	CubeSats at time t; i.e., $S^t \subseteq S$
$\mathcal{G}^t$	Ground devices at time t
V	Set of all nodes for planning horizon T
L	Set of all links for planning horizon T
$\bar{V}$	Set of nodes in aerial and satellite networks
$\bar{L}$	Set of links in aerial and satellite networks
V^t	Set of nodes at time t
L^t	Set of directed links at time t
$L^+(v,t)$	Outgoing links of node v at time t
$L^-(v,t)$	Incoming links of node v at time t
2.	**Constants**
h_u	The UAV's flying height
α	Path loss exponent
β	SINR threshold
N_0	Ambient noise power
P	Transmit power of ground devices
L_{max}	Maximum number of uplinks
B_{max}	Maximum data storage capacity of the UAV
Q_{max}	Maximum data storage capacity of CubeSats
3.	**Parameters**
v	A node $v \in V$
l	A directed link $l \in L$
c_v^t	Storage capacity of node v at time t
c_l^t	Bandwidth of link l at time t
f_l^t	Forwarded data on link l at time t
l_j^t	Uplink between ground device g_j and the UAV at time t
d_j^t	Euclidean distance from ground device g_j to the UAV at time t
$PL(d_j^t)$	Path loss of uplink l_j^t
$\mathcal{N}(\mu,\sigma^2)$	Gaussian random variable with mean μ and variance σ^2

Table 5.1: A summary of notations.

where $s_i \in S^t$. For example, in the third time slot of Figure 5.1, CubeSat s_1 only has one neighbor s_2; thus, we have $N_{s_1}^3 = \{s_2\}$ when $S^3 = \{s_1, s_2\}$. Assume that the operator knows the orbit and speed of deployed CubeSats, the trajectory and speed of the UAV, and the location of ground devices [174]. Hence, the gateway is aware of the contact time and duration of each satellite and the UAV.

5.1.1.2 CubeSats

Assume that each CubeSat is aware of neighboring CubeSats at each time slot; e.g., they can broadcast HELLO messages periodically to discover each other [175]. Each CubeSat is assumed to have two radios [49]; one to communicate with other CubeSats, and the other with the gateway and/or the UAV. Communications between CubeSats or ISLs operate over the S-band (2.45 GHz). Each CubeSat can transmit or receive to/from one other neighbor in each time slot. There is no interference between ISLs; i.e., ISLs are assigned a distinct orthogonal channel. Downlinks to the gateway operate in the UHF band or a frequency of 0.3 GHz. At any given time, only one CubeSat can communicate with the gateway [1].

5.1.1.3 Unmanned Aerial Vehicle

UAV u flies in a given trajectory with a fixed height h_u. It collects data from ground devices at multiple data collection points located on the given trajectory. Note that there is only one data collection point in each time slot. Hence, the total number of data collection points is equal to the planning time horizon T. Note that UAV trajectory optimization is complementary to our work. *Any* trajectory optimization solutions that also consider transmit power control can be used in our system. For example, the work in [138] can be used to optimize the trajectory of a UAV and uplink power control in order to minimize the total energy consumption for data collection of IoT devices or maximize the total amount of collected data at the UAV over a planning time horizon.

The UAV has two radios: (i) a SIC radio that is used to collect data from multiple ground devices simultaneously [61], and (ii) uplinks to CubeSats operate over an Ultra High Frequency (UHF) band (0.45 GHz) [49]. Note, when the UAV is co-located with the gateway, it is able to download all its collected data to the gateway.

5.1.1.4 Network Topology

The network topology varies over time but it is fixed in each time slot. Let V and L denote the set of all nodes and directed links for the planning time horizon T, respectively. Moreover, define $\bar{V}$ and $\bar{L}$ to respectively contain all nodes and links in both aerial and satellite networks. Define V^t and L^t to contain nodes and directed links of the time-varying network topology at time t, respectively. Denote a node and a directed link as v and l, where $v \in V^t$ and $l \in L^t$. The capacity of each node and link is denoted as c_v^t and c_l^t, respectively. Let $< C_V^t, C_L^t >$ denote a tuple that includes storage and bandwidth capacity; i.e., c_v^t and c_l^t, of nodes and links at time slot t. In the considered network topology, there are five types of directed links: (i) uplinks from ground devices to the UAV; i.e., (g_j, u), (ii) uplinks from the UAV to CubeSats; i.e., (u, s_i), (iii) ISLs; i.e., (s_i, s_j), (iv) downlinks from CubeSats to the gateway; i.e., (s_i, w), and (v) downlinks from the UAV to the gateway; i.e., (u, w), where $g_j, s_i, s_j \in V^t$.

5.1.2 Successive Interference Cancellation

A SIC radio allows the UAV to separate, decode, and remove signals from a composite signal iteratively [61]. Briefly, the UAV first decodes the strongest received signal from the composite or received signal. This decoding is only successful if the SINR of the said strongest signal is above the threshold β. The decoded signal is then subtracted from the composite signal. After that, the UAV continues to the next stage. It repeats the said process to decode the next transmission with the strongest signal. As an example, assume UAV u is receiving from $|\mathcal{G}|$ devices simultaneously. Assume the receive power at the UAV u is in non-decreasing order: $P_1 \leq P_2 \leq \cdots \leq P_{|\mathcal{G}|}$. The decoding order is thus $|\mathcal{G}|, |\mathcal{G}|-1, \ldots, 2, 1$. That is, the signal with receive power P_1 can be decoded if and only if all the preceding stronger signals are first decoded and canceled from the composite signal. In practice, due to noise in the SIC decoding process, the UAV is only able to decode up to L_{max} simultaneous uplink transmissions [61].

Define l_j^t as the directed link between ground device g_j and the UAV at time t, where $j \in \mathcal{G}^t$. Let $PL(d_j^t)$ (dB) denote the path loss of link l_j^t, where d_j^t is the Euclidean distance from device g_j to the UAV. All channels experience block fading, where they remain constant over one time slot but vary across time slots. The path loss over distance d_j^t is

$$PL(d_j^t) = PL(d_0) + 10\alpha \log_{10} \frac{d_j^t}{d_0} + \mathcal{N}(\mu, \sigma^2), \tag{5.1}$$

where $PL(d_0)$ (in dB) is the path loss at the reference distance d_0, and α is the path loss exponent. The Gaussian random variable, denoted as $\mathcal{N}(\mu, \sigma^2)$, has mean $\mu = 0$ and variance σ^2 [176]. All ground devices have a fixed transmit power P (dBm). The receive power (in Watt) of link l_j^t is

$$P_j^t = 10^{\frac{P - PL(d_j^t)}{10}}. \tag{5.2}$$

For each directed link l_j^t between ground device g_j and the UAV, a set of links is constructed and defined as $L_j^t = \{l_k^t \mid P_k^t \leq P_j^t, \forall k \in \mathcal{G}^t \setminus j\}$. In particular, set L_j^t consists of uplinks l_k^t that have smaller or equal receive power than link l_j^t in time slot t. Let x_j^t denote whether uplink from ground device g_j is active at time t $(x_j^t = 1)$. Formally, to enable SIC decoding, we have

$$\frac{P_j^t + M(1 - x_j^t)}{\sum_{k \in L_j^t} x_k^t P_k^t + N_0} \geq \beta, \quad \forall j, k \in \mathcal{G}^t; j \neq k, \tag{5.3}$$

$$x_j^t, x_k^t \in \{0, 1\}, \quad \forall j, k \in \mathcal{G}^t; j \neq k. \tag{5.4}$$

In Eq. (5.3), N_0 denotes the ambient noise power and β denotes a given Signal-to-Interference-Plus-Noise Ratio (SINR) threshold value. M is a suitable large value, e.g., 100, that is used to disable Eq. (5.3) when x_j^t is zero. The first term in the denominator of Eq. (5.3) indicates the total interference from other signals in the composite signal with a smaller receive power. If any signal k is not selected to communicate with the UAV, e.g., $x_k^t = 0$, it will not be included in the total interference. Eq. (5.4) ensures variables x_j^t and x_k^t are binary.

For a given SINR threshold value β and bandwidth $\mathcal{B}$ (in MHz), the corresponding asymptotic link capacity (in Mbps) of the uplink from ground devices $g_j \in \mathcal{G}^t$ and the UAV u is

$$c_{(g_j,u)} = \mathcal{B}log_2(1+\beta), \tag{5.5}$$

5.1.3 Link Activation

Define function $L^+(v,t)$ and $L^-(v,t)$ to return the set of outgoing and incoming links of node v at time t, where $v \in V^t$. Specifically, $L^+_{ISL}(v,t)$ and $L^-_{ISL}(v,t)$ return the set of outgoing and incoming ISLs of node v at time t. Similarly, $L^+_{UL}(v,t)$, $L^-_{UL}(v,t)$, $L^+_{DL}(v,t)$, and $L^-_{DL}(v,t)$ return the set of outgoing and incoming uplinks (UL) and downlinks (DL) of node v at time t, respectively. As shown in Figure 5.1, for example, we have $L^+_{ISL}(s_1,3) = \{(s_1,s_2)\}$, $L^+_{UL}(u,3) = \{(u,s_2)\}$ and $L^+_{DL}(s_2,4) = \{(s_2,GW)\}$.

As each CubeSat s_i has a half-duplex radio, only one link in the set $\mathcal{L}(s_i,t) = L^+_{ISL}(s_i,t) \cup L^-_{ISL}(s_i,t)$ can be active. Formally, for each CubeSat s_i, we have

$$\sum_{l \in \mathcal{L}(s_i,t)} x_l^t \leq 1 \tag{5.6}$$

in each time slot t. Similarly, a UAV u is paired with at most one CubeSat in each time slot t,

$$\sum_{l \in L^+_{UL}(u,t)} x_l^t \leq 1. \tag{5.7}$$

Conversely, a CubeSat s_i can only connect to one UAV at any given time slot t; i.e., at most one link in $L^-_{UL}(s_i, t)$ is active at time t. Mathematically,

$$\sum_{l \in L^-_{UL}(s_i,t)} x^t_l \leq 1. \tag{5.8}$$

The gateway w communicates with at most one CubeSat at each time t. Formally,

$$\sum_{l \in L^-_{DL}(w,t)} x^t_l \leq 1 \tag{5.9}$$

$$\sum_{l \in L^+_{DL}(s_i,t)} x^t_l \leq 1. \tag{5.10}$$

5.1.4 Data Transmission/Collection

Symbol	Description
1.	**UAV**
D^t_u	Data arrives at UAV u from ground devices at time t
$\hat{D}^t_u$	Data transferred from UAV u to CubeSats at time t
$\tilde{D}^t_u$	Data transferred from UAV u to the gateway at time t
B^t_u	Data stored by UAV u at time t
2.	**CubeSats**
D^t_i	Data arrives at CubeSat s_i from the UAV at time t
$\hat{D}^t_i$	Data arrives at CubeSat s_i over ISLs at time t
$\tilde{D}^t_i$	Data transferred over ISLs from CubeSat s_i at time t
$\bar{D}^t_i$	Data transferred over downlinks to the gateway at time t
B^t_i	Data stored by CubeSat s_i at time t
3.	**Gateway**
D^t_w	Data transferred from CubeSats to the gateway at time t
$\hat{D}^t_w$	Data transferred from the UAV to the gateway at time t
B^t_w	Data stored by gateway w at time t

Table 5.2: A summary of notations related to data transmission/collection.

Table 5.2 lists the notations used for data transmission and/or collection. Let f^t_l be the data that is forwarded on directed link l. Note that the amount of data forwarded on link l must not exceed the link capacity c^t_l, meaning for all directed links $l \in L^t$, we have

$$0 \leq f_l^t \leq c_l^t, \quad \forall l \in L^t, \forall t \in T. \tag{5.11}$$

Next, we present data collection and/or transmission formulation at (i) UAV, (ii) CubeSats, and (iii) gateway.

5.1.4.1 Unmanned Aerial Vehicle

Let D_u^t denote the toal data that arrives at UAV u from ground devices at time t. Formally,

$$D_u^t = \sum_{l \in L_{UL}^-(u,t)} f_l^t x_l^t. \tag{5.12}$$

Denote the data transferred from the UAV to CubeSats at time t as $\hat{D}_u^t$, we have

$$\hat{D}_u^t = \sum_{l \in L_{UL}^+(u,t)} f_l^t x_l^t. \tag{5.13}$$

Note that the UAV can only communicate with the gateway directly when it flies back to the gateway. Let $\tilde{D}_u^t$ denote the amount of data transfer from the UAV to the gateway at time t.

5.1.4.2 CubeSats

For a CubeSat s_i, at time t, define (i) D_i^t as the data received from the UAV, (ii) $\hat{D}_i^t$ as the data that arrives over ISLs, (iii) $\tilde{D}_i^t$ is the data transferred over ISLs from s_i, and (iv) $\bar{D}_i^t$ is the data transferred over downlinks to the gateway. Formally, we have

$$D_i^t = \sum_{l \in L_{UL}^-(s_i,t)} f_l^t x_l^t \tag{5.14}$$

$$\hat{D}_i^t = \sum_{l \in L_{ISL}^-(s_i,t)} f_l^t x_l^t \tag{5.15}$$

$$\tilde{D}_i^t = \sum_{l \in L_{ISL}^+(s_i,t)} f_l^t x_l^t \tag{5.16}$$

$$\bar{D}_i^t = \sum_{l \in L_{DL}^+(s_i,t)} f_l^t x_l^t. \tag{5.17}$$

5.1.4.3 Gateway

The data transferred from CubeSats and the UAV to the gateway at time slot t is denoted respectively as D_w^t and $\hat{D}_w^t$, respectively. Specifically, the amount of downloaded data D_w^t from CubeSats is calculated as

$$D_w^t = \sum_{l \in L_{DL}^-(w,t)} f_l^t x_l^t. \tag{5.18}$$

When UAV u is at gateway w, it downloads all its data to the gateway. We have

$$\hat{D}_w^t = \tilde{D}_u^t. \tag{5.19}$$

5.1.5 Data Storage

In each time slot t, define the data stored by UAV u, CubeSat s_i and gateway w as B_u^t, B_i^t and B_w^t, respectively. Formally, B_u^t, B_i^t and B_w^t are updated as follows:

$$B_u^t = B_u^{t-1} + D_u^t - \hat{D}_u^t - \tilde{D}_u^t, \tag{5.20}$$

$$B_i^t = B_i^{t-1} + D_i^t + \hat{D}_i^t - \tilde{D}_i^t - \bar{D}_i^t, \tag{5.21}$$

$$B_w^t = B_w^{t-1} + D_w^t + \hat{D}_w^t. \tag{5.22}$$

Note that Eq. (5.20) and (5.21) must be positive to ensure that the UAV and/or a CubeSat only transmit their available data. Formally, the above considerations are modeled as Further, a node cannot transmit data that is not in the buffer. Formally, we have

$$0 \leq \hat{D}_u^t \leq B_u^{t-1}, \tag{5.23}$$

$$\tilde{D}_u^t = B_u^{t-1}, \tag{5.24}$$

$$0 \leq \tilde{D}_i^t \leq B_i^{t-1}, \tag{5.25}$$

$$0 \leq \bar{D}_i^t \leq B_i^{t-1}. \tag{5.26}$$

Eq. (5.23) ensures that the data sent from the UAV to a CubeSat at time t is no more than the UAV stored data at time $t-1$. Eq. (5.24) indicates that the data transferred from the UAV to the gateway is equal to the data stored in the UAV buffer at time $t-1$. Eq. (5.25) and (5.26) ensure that the data transferred over ISL and downlink to the gateway is no more than the stored data of CubeSat s_i at time $t-1$, respectively. Further, the data stored at UAV u and any CubeSat s_i must not exceed the respective maximum storage capacity; i.e., B_{max} and Q_{max}. Formally, for all time t, we have

$$0 \leq B_u^t \leq B_{max}, \quad \forall t \in T, \tag{5.27}$$

$$0 \leq B_i^t \leq Q_{max}, \quad \forall t \in T. \tag{5.28}$$

5.1.6 Routing

A feasible routing from a ground device to gateway w via the UAV must satisfy constraints relating to flow conservation. Specifically, any incoming flow into a node must equal to the amount of outgoing flow. Formally, any routing must satisfy

$$\sum_{t \in T} D_u^t - \sum_{t \in T} D_w^t - \sum_{t \in T} \hat{D}_w^t = 0 \tag{5.29}$$

$$\sum_{t \in T} D_u^t - \sum_{t \in T} \hat{D}_u^t - \sum_{t \in T} \tilde{D}_u^t = 0 \tag{5.30}$$

$$\sum_{t \in T} \hat{D}_u^t - \sum_{t \in T} \sum_{i \in S^t} D_i^t = 0 \tag{5.31}$$

$$\sum_{t \in T} D_i^t + \sum_{t \in T} \hat{D}_i^t - \sum_{t \in T} \tilde{D}_i^t - \sum_{t \in T} \bar{D}_i^t = 0 \tag{5.32}$$

$$\sum_{t \in T} D_w^t + \sum_{t \in T} \hat{D}_w^t - \sum_{t \in T} \sum_{i \in S^t} \bar{D}_i^t - \sum_{t \in T} \tilde{D}_u^t = 0. \tag{5.33}$$

Eq. (5.29) ensures that gateway w receives the total amount of data sent by ground devices over the given planning horizon T. Eq. (5.30) ensures that the UAV sends all collected data from ground devices to CubeSats and/or the gateway over planning horizon T. Eq. (5.31) ensures that over the planning horizon T, the data transferred over uplinks between the UAV and CubeSats is received by all CubeSats. Eq. (5.32) indicates that the received data at each CubeSat s_i is transmitted over ISLs and downloaded to the gateway. Lastly, Eq. (5.33) ensures the data received at the gateway equals to the data transferred over downlinks from both CubeSats and the UAV.

5.2 The Problem

The objective of interest is to maximize the minimum flow among all uplinks from ground devices to gateway w over planning time horizon T. For convenience, let $f_l = \sum_{t \in T} f_l^t$, where $l \in \mathcal{G}$, define the total flow of each uplink from the set of ground devices over the given time horizon T. Denote the minimum flow as $f = \min\{f_l\}, \forall l \in \mathcal{G}$. To compute the said objective, the following MILP is formulated:

$$\begin{aligned} \underset{x_l^t, f_l^t}{\text{MAX}} \quad & f \\ \text{s.t.} \quad & (5.3) - (5.4), (5.6) - (5.33). \end{aligned} \tag{5.34}$$

The previous MILP has two decision variables: (i) binary variable x_l^t, which indicates whether a directed link is active in time t, and (ii) f_l^t, which corresponds to the amount of data forwarded over an active link. Notice that constraints (5.12) - (5.18) are non-linear as they involve the product of two decision variables. They can be linearized as follows[1]. Consider the expression $Z_a^t = x_a^t f_a^t$, where x_a^t is a binary variable and f_a^t is a real number. Define the following constraint: $f_a^t \leq M x_a^t$, where M is a suitable large integer; e.g., 100. Then Z_a^t can be rewritten as $Z_a^t = f_a^t$. Observe that if the binary variable x_a^t is zero, the new constraint will force f_a^t to

[1]Readers who are unfamiliar with modelling tricks are referred to [177].

zero; otherwise, f_a^t is allowed to be non-negative.

The above MILP becomes computationally intractable for large-scale networks. This is because the number of binary decision variables increases significantly with the number of (i) directed links, (ii) combinations of ground devices that satisfy SIC constraints, (iii) CubeSats, and (iv) increasing T values. To see this, assume there are $n = |\mathcal{S}|$ CubeSats and $g = |\mathcal{G}|$ ground devices. Then in each time t, there are g uplinks from ground devices to the UAV, n uplinks from the UAV to CubeSats, $\eta = n(n-1)$ ISLs and n downlinks from CubeSats to the gateway. Hence, the total number of binary decision variables is $g + \eta + 2n$. Define $p = \lfloor n/2 \rfloor$; this is the number of CubeSats pairs given n CubeSats. Then the search space in each time t is of size $\mathcal{O}\left(\binom{\eta}{p} + n! + n! + g!\right)$, which increases exponentially with the number of CubeSats and ground devices. This fact motivates the development of a heuristic.

5.3 A Novel Distributed Algorithm: IFPR

This section outlines a distributed algorithm, called Iterative Flow and Path Reservation (IFPR). It is run by the UAV to determine (i) a path to the gateway in each time slot, (ii) the corresponding storage and bandwidth reservation for nodes and links on the selected path, and (iii) the amount of data to upload from ground devices. Recall that the orbit and speed of CubeSats are pre-determined. Hence, the UAV knows the time-varying topology of CubeSats The main advantage of IFPR is that the UAV is able to determine a path with sufficient resources to the gateway by itself without coordination with CubeSats or the gateway. The UAV then downloads as much data as possible from ground devices to fill the selected path in each time slot.

A general overview of IFPR is as follows. First, the UAV is aware of V^t and L^t, where $t = 1, 2, \dots, T$. In each time slot t, it computes a path P^t to the gateway with the fewest number of hops. It then determines the maximum capacity or flow rate that can be transmitted over path P^t; let this quantity be F^t. After that, the UAV collects $R^t \geq F^t$ amount of data from ground devices. The UAV then updates the storage and bandwidth resources on path P^t. The collected data is then packetized, whereby each packet contains a source route to the gateway. Specifically, it contains the tuple $< t, src, dst >$, which describes the source and destination address of each packet for time slot t; note src and dst are nodes on path P^t. Referring to Figure 5.2, in the third time slot ($t = 3$), CubeSat s_1 receives a packet transmitted in time slot-1 containing the tuple $< 3, s_1, s_2 >$. Similarly, after receiving the packet from s_1, which contains the tuple $< 4, s_2, w >$, CubeSat s_2 downloads the packet to gateway w in time slot $t = 4$.

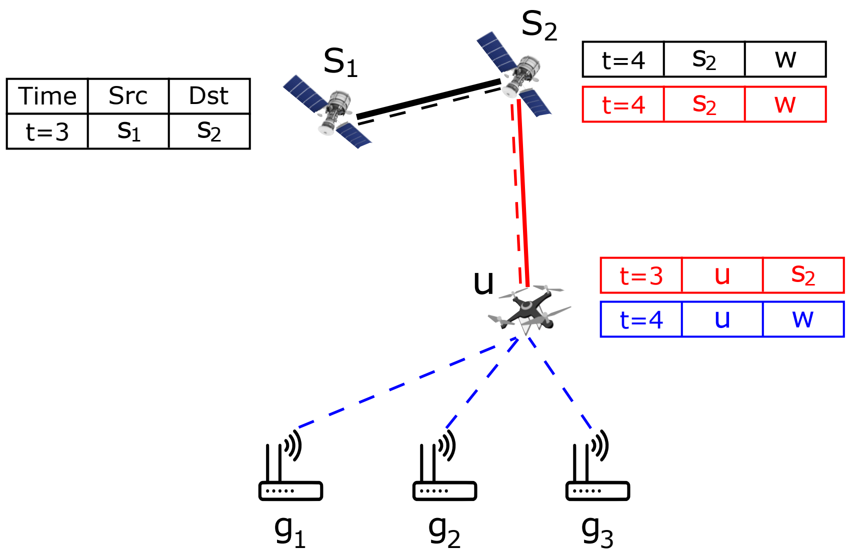

Figure 5.2: An example time slot of Figure 5.1. UAV u, CubeSats s_1 and s_2 have packets that originated from different time slots. Different table colors indicate packets transmitted along different paths. *Src* and *Dst* correspond to the source and destination address of a packet, respectively.

Algorithm 9 illustrates the steps taken by a UAV when it executes IFPR. For each time slot t, according to nodes V^t and directed link L^t, the UAV first calls Dijkstra() to compute the least cost path P^t to gateway w; see Line 2. Specifically, Dijkstra() implements the Dijkstra algorithm [178]. The weight of each link in time t is defined as $1/(c_l^t + c_v^t)$, where c_l^t and c_v^t are the available bandwidth and storage capacity of link l and its source node v. There are three types of paths: (i) UAV-gateway; i.e., $\{(u, w)^{t+1}\}$, (ii) UAV-CubeSat-gateway; i.e., $\{(u, s_i)^{t+1}, (s_i, w)^{t+2}\}$, and (iii) UAV-

CubeSat-CubeSat-gateway i.e., $\{(u, s_i)^{t+1}, (s_i, s_j)^{t+2}, (s_j, w)^{t+3}\}$. Then in Line 3, the UAV determines the maximum capacity, denoted as F^t, of path P^t by finding the minimum available capacity of nodes and links on path P^t. Specifically, denote $\hat{V}^t$ and $\hat{L}^t$ as two sets that consist of nodes and links on path P^t, respectively. The respective storage and bandwidth capacity of nodes and links in sets $\hat{V}^t$ and $\hat{L}^t$ are stored in set $C_{\hat{V}^t}$ and $C_{\hat{L}^t}$. For example, as shown in Figure 5.1, UAV u selects a path $\{(u, s_2)^3, (s_2, w)^4\}$ in time $t = 2$. Specifically, it will offload data to CubeSat s_2 in time slot 3 before downloading to gateway w at time $t = 4$. The set $\hat{V}^2$ and $\hat{L}^2$ contains $\{u, s_2\}$ and $\{(u, s_2), (s_2, w)\}$, respectively. The set $C_{\hat{V}^2}$ stores the storage capacity of nodes included in $\hat{V}^2$; i.e., $C_{\hat{V}^2} = \{c_u^2, c_{s_2}^2\}$. Additionally, the set $C_{\hat{L}^2}$ stores the bandwidth of directed links included in $\hat{L}^2$; i.e., $C_{\hat{L}^2} = \{c^2_{(u,s_2)}, c^2_{(s_2,w)}\}$. The maximum capacity F^2 of path $\{(u, s_2)^3, (s_2, w)^4\}$ is the minimum of c_u^2, $c_{s_2}^2$, $c^2_{(u,s_2)}$ and $c^2_{(s_2,w)}$.

Referring to Line 4 of Algorithm 9, the UAV calls SIC() to (i) schedule uplink transmissions from ground devices included in $\mathcal{G}^t$, where these transmissions satisfy SIC constraints; see Eq. (5.3) and (5.4), and (ii) determine the total amount of uploaded data R^t from scheduled ground devices $\hat{\mathcal{G}}^t$. In Line 5, the UAV calls CollectData(), which informs ground devices the computed transmission schedule $\hat{\mathcal{G}}^t$. Scheduled ground devices then upload their data to the UAV. Let D^t denote the data collected by the UAV from scheduled ground devices in time slot t. The UAV then calls Packetize() to packetize data D^t; for each packet $\mathbb{P}^t$, it includes a source route of the source and destination address of links in path P^t; see Line 6. After that, in Line 7-12, the UAV updates the available capacity of nodes and links on path P^t; i.e., $C_{\hat{V}^t}$ and $C_{\hat{L}^t}$. Specifically, it subtracts R^t from $C_{\hat{V}^t}$ and $C_{\hat{L}^t}$, respectively.

Algorithm 9: IFPR algorithm.

Input : V^t, L^t, T

for $t \leftarrow 1$ *to* T **do**

  /* Determine a path to the gateway */

  $P^t = \text{Dijkstra}(G(\tilde{V}, \tilde{L}, T))$;

  /* Determine the maximum path capacity */

  $F^t = MIN\{c_v^t, c_l^t \mid v \in \hat{V}^t, l \in \hat{L}^t\}$;

  /* Determine scheduled ground devices with total flow */

  $[\hat{\mathcal{G}}^t, R^t] = \text{SIC}(\mathcal{G}^t, F^t)$;

  /* Collect data from scheduled ground devices $\hat{\mathcal{G}}^t$ */

  $D^t = \text{CollectData}(\hat{\mathcal{G}}^t, R^t)$;

  /* Packetize collected data for transmission */

  $\mathbb{P}^t = \text{Packetize}(P^t, D^t)$;

  /* Update the capacity on $G(\tilde{V}, \tilde{L}, T)$ */

  for $\hat{V}^t \in P^t$ **do**

    $C_{\hat{V}^t} = C_{\hat{V}^t} - R^t$;

  end

  for $\hat{L}^t \in P^t$ **do**

    $C_{\hat{L}^t} = C_{\hat{L}^t} - R^t$;

  end

end

5.3.1 Ground Devices Schedulers

This section introduces two methods to schedule ground devices and determine their corresponding uploaded data in each time slot t, namely a simplified MILP (SMILP) and a greedy algorithm called Less Data Schedule First (LDSF). These two methods are denoted as IFPR-SMILP and IFPR-LDSF, respectively. Specifically, IFPR-SMILP requires solving an MILP to schedule one or more ground devices to transmit simultaneously. IFPR-LDSF is a heuristic method that greedily schedules ground devices that have uploaded the least amount of data in past time slots. Compared to SMILP, LDSF is suitable for large-scale networks because it does not involve solving a MILP.

5.3.1.1 IFPR-SMILP

IFPR-SMILP solves the following MILP (5.35) to schedule ground devices in each time slot: IFPR-SMILP has two decision variables: (i) x_j^t, which represents whether

$$\max_{x_j^t, r_j^t} \quad \sum_{j \in \mathcal{G}^t} r_j^t x_j^t \tag{5.35a}$$

s.t.

$$\frac{P_j^t + M(1 - x_j^t)}{\sum_{k \in L_j^t} x_k^t P_k^t + N_0} \geq \beta, \ \forall j, k \in \mathcal{G}^t, j \neq k, \tag{5.35b}$$

$$r_j^t \leq c_{(g_j,u)}^t, \ \forall j \in \mathcal{G}^t, \tag{5.35c}$$

$$\sum_{j \in \mathcal{G}^t} r_j^t x_j^t \geq F^t, \tag{5.35d}$$

$$x_j^t, x_k^t \in \{0, 1\}, \ \forall j, k \in \mathcal{G}^t, j \neq k \tag{5.35e}$$

there is an uplink between ground device j and UAV u, and (ii) r_j^t, which represents the corresponding flow over uplink (g_j, u), where $j \in \mathcal{G}^t$.

Referring to Eq. (5.35), IFPR-SMILP has four constraints; see (5.35b) - (5.35e). Specifically, Constraint (5.35b) checks whether the SINR and/or SNR of an uplink between ground device j and the UAV is no less than the SINR threshold β. Constraint (5.35c) bounds the total flow of active uplinks to be no more than the determined path capacity F^t. Constraint (5.35d) ensures that the total uploaded data R^t from scheduled devices is no less than the maximum capacity F^t of a selected path P^t. Constraint (5.35e) ensures variables x_j^t and x_k^t are binary.

5.3.1.2 IFPR-LDSF

Referring to Algorithm 10, IFPR-LDSF first sorts ground devices $\mathcal{G}^t$ according to their receive power and total flow in previous time slots; see Line 1. Assume the receive power P_j^t of $|\mathcal{G}^t|$ ground devices are in decreasing order; formally, $P_j^t \geq P_{j+1}^t \geq \ldots P_{|\mathcal{G}^t|}^t$. It then iterates through ground devices in sorted $\mathcal{G}^t$ and determines the total amount of uploaded data R^t; see Line 3-11. Specifically, it checks whether SIC is successful for ground device j; see Line 5. If the SINR and/or SNR of ground device j is no less than the SINR threshold β, the data rate of ground device j, denoted as r_j^t, is added to the total amount of uploaded data R^t; see Line 6. Algorithm 10 will stop adding ground devices when the total uploaded data R^t from scheduled devices exceeds the maximum capacity F^t of path P^t.

Algorithm 10: IFPR-LDSF.

Input: $\mathcal{G}^t, F^t$
Initialize: $\hat{\mathcal{G}}^t = \emptyset, R^t = 0$
$\mathcal{G}^t = \text{Sort}(\mathcal{G}^t)$;
while $R^t \leq F^t$ **do**
    for $j \leftarrow 1$ *to* $|\mathcal{G}^t|$ **do**
        for $k \leftarrow j+1$ *to* $|\mathcal{G}^t|$ **do**
            if $\frac{P_j^t}{N_0 + \sum_k^{|\mathcal{G}^t|} P_k^t} \geq \beta$ **then**
                $R^t = R^t + r_j^t$
            else
                break
            end
        end
    end
end

5.3.2 Analysis

This section presents the following propositions: (i) the run time complexity of IFPR, and (ii) that IFPR yields the maximum flow of a time-varying graph.

Proposition 1. *IFPR has run time complicity* $\mathcal{O}(T|\bar{V}|^2)$.

Proof. For each time slot t, IFPR calls the Dijkstra algorithm once, which takes $\mathcal{O}(|\bar{V}|^2)$, where $\bar{V}$ is the set that contains all nodes in both aerial and satellite network of a SAGIN.. For Line 4 of Algorithm 9, IFPR applies SIC() to schedule ground devices. Hence, it needs to check no more than $|\mathcal{G}^t|$ ground devices. In Algorithm 9, IFPR will run Line 7-9 for $\mathcal{O}(|S|+2)$ times when updating the storage capacity of nodes on path P^t. Similarly, IFPR runs Line 10-12 for $\mathcal{O}(|S|/2+2)$ times to update the available bandwidth capacity of selected links on path P^t. Hence, the total run time complexity of the above steps for all T time slots is $\mathcal{O}\left(T(|\bar{V}|^2 + |\mathcal{G}^t| + |S| + 2 + |S|/2 + 2)\right) = \mathcal{O}(T|\bar{V}|^2)$. ∎

Define a time-varying graph $G(V, L)$ to model the topology over time horizon T. Specifically, $G(V, L)$ has a virtual source s that connects to UAV u when it is located in each time slot t. Define $\hat{R}^t$ as the maximum amount of data collected that can be collected by the UAV in time slot t. Let F^* be the maximum flow of $G(V, L)$ from virtual source s to gateway w. As per the min-cut theorem, the quantity F^* equates to the total flow over the links in the minimum cut; let set $\mathcal{C}$ contain these links. Define $\bar{R}^t$ as the residual capacity of $G(V, L)$, where $\bar{R}^t = F^* - \sum_{k=1}^{t} F^t$.

Proposition 2. *If $\sum_{t=1}^{T} \hat{R}^t \geq F^*$, then IFPR guarantees that $\sum_{t=1}^{T} F^t = F^*$.*

Proof. In each time slot t, we have $0 \leq F^t \leq \hat{R}^t$. In Line 2 of Algorithm 9, IFPR runs the Dijkstra algorithm on $G(V, L)$ to obtain a path P^t with non-negative capacity to gateway w. If there is no such path, we then have $F^t = 0$; otherwise, IFPR determines a value of F^t that must satisfy $0 \leq F^t \leq F^*$, $0 \leq F^t \leq \bar{R}^{T-1}$, and $0 \leq F^t \leq F^* - \sum_{k=1}^{t} \hat{R}^k$. Note that the term $F^* - \sum_{k=1}^{t} \hat{R}^k$ is monotonic decreasing. This is because for each time slot t, either Dijkstra algorithm finds a path with zero capacity or there is a path to the gateway that crosses the link in $\mathcal{C}$, which decreases the said term after Line 2 of Algorithm 9. Moreover, we are given $\sum_{t=1}^{T} \hat{R}^t \geq F^*$, meaning at time T we have $F^* - \sum_{k=1}^{T-1} \hat{R}^k \leq 0$. However, we have $\sum_{t=1}^{T} F^t \leq F^*$ because $F_{(u,v)}^t \leq c_{(u,v)}^t$. This means $\sum_{t=1}^{T} F^t = F^*$; i.e., at the end of time T, IFPR saturates the links in $\mathcal{C}$. ∎

5.4 Evaluation

All experiments are conducted in Matlab [168]. The simulation settings are listed in Table 5.3. In each time slot, a random topology is generated with up to six CubeSats and up to ten ground devices. A single UAV flies in a given circular trajectory with a fixed height $h_u = 80m$ and a fixed radius $R_u = 100m$ [37]. Ground devices are located in urban environment, in which the path loss exponent α is set to 2.7 [176]. The bandwidth B is set to 1 MHz and the SINR threshold is $\beta = 5$ (dB) [166]. According to Shannon-Hartley formula; see Eq. (5.5), the corresponding data rate of uplink between ground devices and the UAV is 2.58 Mbps. The link capacity of uplinks between the UAV and CubeSats; i.e., (u, s_i), ISLs; i.e., (s_i, s_j), and downlinks between CubeSat and the gateway; i.e., (s_i, GW) is set to 2 Mbps, 3 Mbps, and 5 Mbps, respectively [179].

The optimal results obtained from solving the formulated MILP are labeled as SIC-MILP. The experiments first apply SIC-MILP and study the impact of the number of ground devices $|\mathcal{G}|$, the number of CubeSats $|\mathcal{S}|$ and the maximum storage capacity of the UAV; see Section 5.4.1. The experiments in Section 5.4 then compare SIC-MILP against IFPR under the obtained configuration from Section 5.4.1; see Section 5.4.2. In both Section 5.4.1 and Section 5.4.2, The experiments results are an average of 20 simulation runs. Note that the considered problem in this Chapter is new. There is no existing solutions that solve the same problem. Hence, the proposed methods do not compare against other works.

Symbol	**Value**	**Symbol**	**Value**
σ^2	2 dB^2	d_0	1 m
P	1 W	N_0	-110 dBm
L_{max}	4 [61]	Q_{max}	15 Mbits

Table 5.3: Simulation settings.

5.4.1 SIC-MILP

The results herein consider SIC-MILP and include two cases: (i) UAV u downloads data to gateway w directly; see Section 5.4.1.1, and (ii) UAV u offloads data to CubeSat(s) before downloading to gateway w; see Section 5.4.1.2. The planning time horizon T is 20. For every five time slots, UAV u will have a directed downlink to gateway w.

5.4.1.1 No CubeSats Relay

This evaluation first studies the impact of the maximum storage capacity B_{max} of UAV u and the number of ground devices $|\mathcal{G}|$. Specifically, it considers the following B_{max} values (Mbits): 10, 15 and 20. The number of ground devices $|\mathcal{G}|$ increases from one to ten. Moreover, it also investigates different ground devices placement methods; see Figure 5.3 for an example. They include

- *P-Circle*. Ground devices are uniformly located on the perimeter of a circle with a radius; i.e., R_g, of 100 meter.
- *I-Circle*. Ground devices are randomly located in a circular area with a radius of 100 meter.

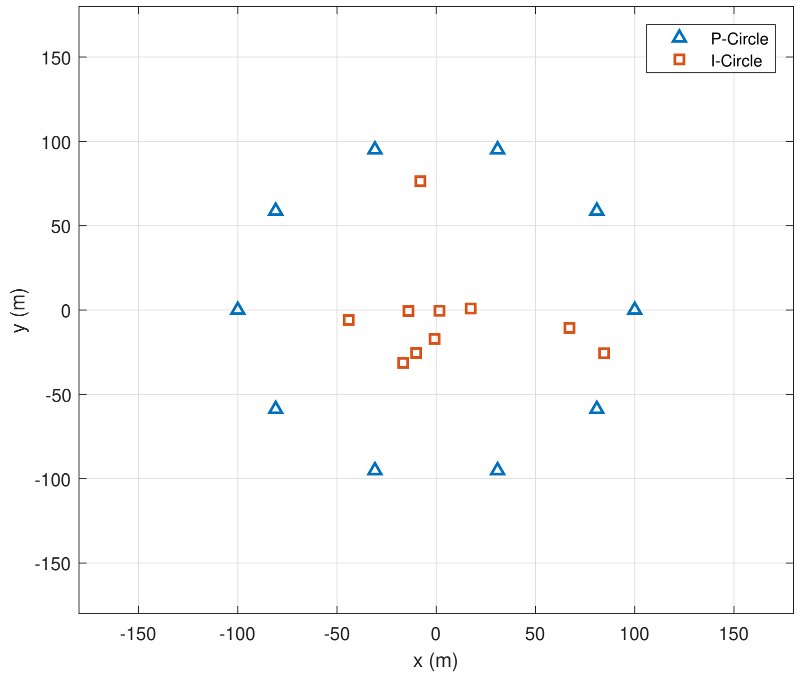

Figure 5.3: An example of two ground devices placement methods.

Figure 5.4 shows the total transmitted data with (i) increasing number of ground devices $|\mathcal{G}|$, (ii) different storage capacity B_{max}, and (iii) different placement methods of ground devices. First, we see that when B_{max} is 20 and 15 Mbits, the total amount of data collected from ground devices first increases and then remains constant. Specifically, when $B_{max} = 20$ Mbits and $|\mathcal{G}|$ increases from one to four, the total amount of collected data for both P-Circle and I-Circle increases from 50 to 80 and 55 Mbits, respectively. Similarly, when $B_{max} = 15$ Mbits, the total amount of collected data for both placement methods increases from 50 to 60 and 53 Mbits, respectively. This is because the UAV is able to schedule an additional ground device with the help of its SIC radio. When $|\mathcal{G}|$ increases from four to ten, the total amount of collected data remains constant at 80, 55, 60 and 53 Mbits, respectively. This is because for any $|\mathcal{G}|$ values, the UAV is able to successfully decode at most four uplinks. Additionally, each active ground device will be scheduled to upload with the maximum uplink capacity by SIC-MILP. Hence, the total amount of collected data remains constant when $|\mathcal{G}|$ is more than four.

Second, referring to Figure 5.4, a larger storage capacity yields more data from ground devices. For example, for P-Circle, the total amount of collected data is 20 Mbits larger when B_{max} varies from 15 Mbits to 20 Mbits. This is because a higher storage capacity allows the UAV to schedule more simultaneous ground devices in each time slot to upload data. We also see that when the maximum storage capacity of the UAV; i.e., B_{max}, is set to 10 Mbits, the total amount of collected data remains constant at 40 Mbits for both P-Circle and I-Circle. This is because a small data storage capacity can only store uploaded data from a single ground device for each time slot. Thus, the total amount of collected remains constant for any number of ground devices.

Third, as shown in Figure 5.4, we see that P-Circle yields a larger amount of collected data as compared to I-Circle. For example, when the maximum storage capacity, i.e., B_{max}, is set to 20 Mbits, P-Circle collects 45% additional data than I-Circle. This is because the receive power of ground devices have bigger differences when considering P-Circle. Therefore, the number of simultaneous ground devices that satisfy SIC constraints; see Eq. (5.3), and transmit together increases correspondingly. Consequently, the UAV is able to collect more data from ground devices.

5.4.1.2 CubeSats Aided Transmissions

This section studies a swarm of CubeSats, where $|\mathcal{S}|$ ranges from zero to six. Six ground devices are placed using P-Circle. The maximum data storage capacity at

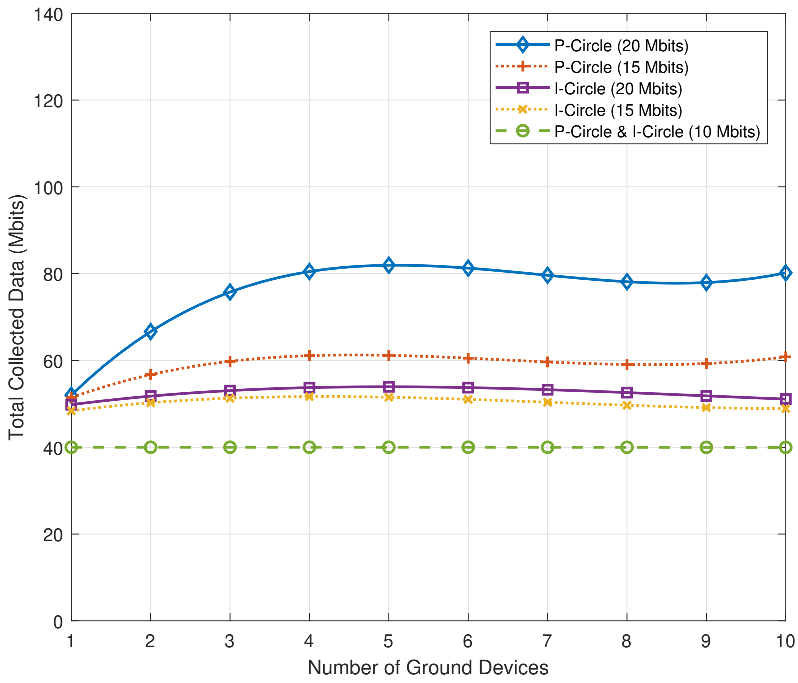

Figure 5.4: Total amount of transmitted data versus number of ground devices.

the UAV; i.e., B_{max}, is set to 10 Mbits.

Figure 5.5 shows (i) the total amount of received data D_u by the gateway, (ii) the total amount of data downloaded from CubeSats swarm and/or the UAV; i.e., D_w and $\hat{D}_w$. Note that for each $|\mathcal{S}|$ value, the total amount of data received by the gateway is the summation of downloaded data from both the UAV and CubeSats swarm; i.e., $D_u = D_w + \hat{D}_w$. We see that total amount of data D_u is higher with more CubeSats; i.e., $|\mathcal{S}|$. For example, when $|\mathcal{S}|$ increases from zero to six, D_u increases by 24.3 Mbits; i.e., from 40 to 64.3 Mbits. This is because there are more ISLs. In addition, extra data can be transferred to CubeSats with individual downlink to the gateway over ISLs. Thus, an increasing number of ISLs indicates that more data can be collected from ground devices and then transferred to the gateway.

Referring to Figure 5.5, the rate in which D_u increases reduces with additional CubeSats. For example, when $|\mathcal{S}|$ increases from zero to four, the increase rate of D_u reduced by half with a newly added CubeSat. In addition, when $|\mathcal{S}|$ increases from four to six, D_u remains at 64.3 Mbits. This is because the maximum data storage B_{max} of the UAV limits the amount of offloaded data. Hence, increasing the number of CubeSats has little impact on D_u.

As shown in Figure 5.5, the amount of data downloaded from the UAV to the gateway; i.e., $\hat{D}_w$, is fixed around 40 Mbits. This is because besides offloading data to CubeSats, the UAV will also collect data and fill its storage. When it has a direct downlink to the gateway, which is set to every five slots in this experiment, it will download all collected data to the gateway.

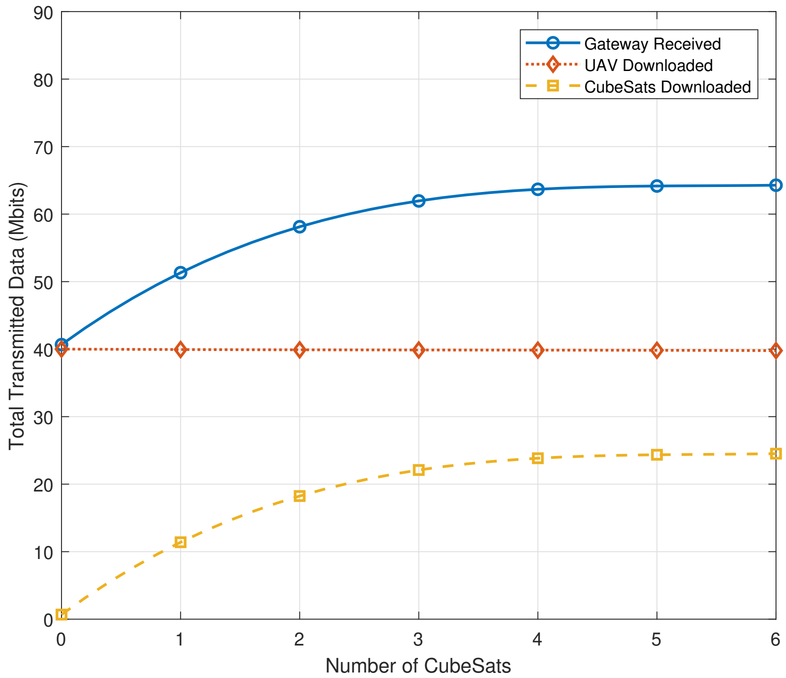

Figure 5.5: The impact on CubeSats numbers on transmitted data.

5.4.2 SIC-MILP Versus IFPR

This section compares SIC-MILP and IFPR. Recall that IFPR uses SMILP or LDSF to schedule ground devices. For both SMILP and LDSF, UAV u either randomly selects a Path (RP) or applies Dijkstra's algorithm to select the Shortest Path (SP). Hence, there are four cases to consider. They are labeled as IFPR-SMILP-SP, IFPR-SMILP-RP, IFPR-LDSF-SP, and IFPR-LDSF-RP, respectively. The experiments use the following parameter values: $|\mathcal{G}| = 6$, $|\mathcal{S}| = 3$, and $B_{max} = 10$ Mbits. The number of time slots T increases from five to 30. Ground devices are placed using P-Circle.

5.4.2.1 The Minimum Flow

In this section, the evaluation investigates how different number of time slots T affects the minimum flow of ground devices. Figure 5.6 shows that the minimum flow of ground devices increases linearly with more time slots. For example, for SIC-MILP, the minimum flow of ground devices increases from 2.37 to 16.01 Mbits when T increases from five to 30. Specifically, the minimum flow increases by 2.75 Mbits when five time slots are added into time horizon T. This is because ground devices have more opportunities to be scheduled when there are more time slots. In addition, once a ground device is scheduled, it will upload with the maximum bandwidth capacity that results in a fixed flow rate.

From Figure 5.6, we see that SIC-MILP has the best performance. For example, when $T = 30$, SIC-MILP yields a minimum flow of 16.01 Mbits over all ground devices. However, IFPR-SMILP-SP, IFPR-SMILP-RP, IFPR-LDSF-SP, and IFPR-LDSF-RP achieve 8.2, 8.2, 10.7 and 10.0 Mbits, respectively. This is because SIC-MILP schedules ground devices and determines the corresponding flow rate over the entire planning time horizon T. In contrast, IFPR uses only the channel condition of the current time slot.

5.4.2.2 Jain's Fairness Index

Here, the experiments study the fairness of flow rates from ground devices. In particular, the performance of ground devices' flow rate is measured using Jain's fairness index, which has the label JFi [169]. Referring to Figure 5.7, the JFi value of SIC-MILP is fixed at one for all T values. In other words, all ground devices have the same flow rate. This is because SIC-MILP schedules ground devices to upload with the maximum bandwidth capacity. Thus, the flow rate is fixed for all ground devices when they are activated.

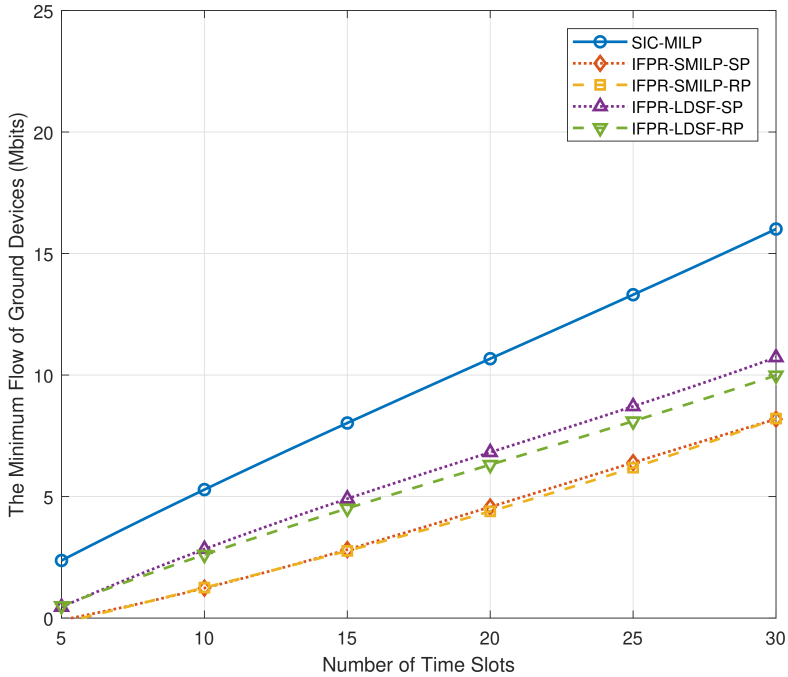

Figure 5.6: Minimum flow of ground devices versus different number of time slots.

As shown in Figure 5.7, the respective JFi value of IFPR-LDSF-SP, IFPR-LDSF-RP, IFPR-SMILP-SP and IFPR-SMILP-RP is 0.77, 0.79, 0.68 and 0.70 when T is five. As T increases to 30, the JFi value of IFPR-LDSF-SP, IFPR-LDSF-RP, IFPR-SMILP-SP and IFPR-SMILP-RP increases by 29.9%, 26.6%, 38.2% and 35.7%, respectively. This is because additional ground devices are able to upload their data with more time slots or when the planning horizon T is longer.

Referring to Figure 5.7, the JFi value of IFPR-SMILP-SP and IFPR-SMILP-RP is lower than SIC-MILP, IFPR-LDSF-SP and IFPR-LDSF-RP. For example, when $T = 30$, IFPR-SMILP-SP and IFPR-SMILP-RP yields a JFi value of 0.94 and 0.95, respectively. However, the JFi of the other three methods is one. This is because the objective of IFPR-SMILP; see Eq. (5.35), is to schedule ground device(s) that can upload the maximum amount of data. Hence, ground devices with a higher data rate might be scheduled multiple times. Consequently, the JFi value is lower than SIC-MILP and the other two greedy methods.

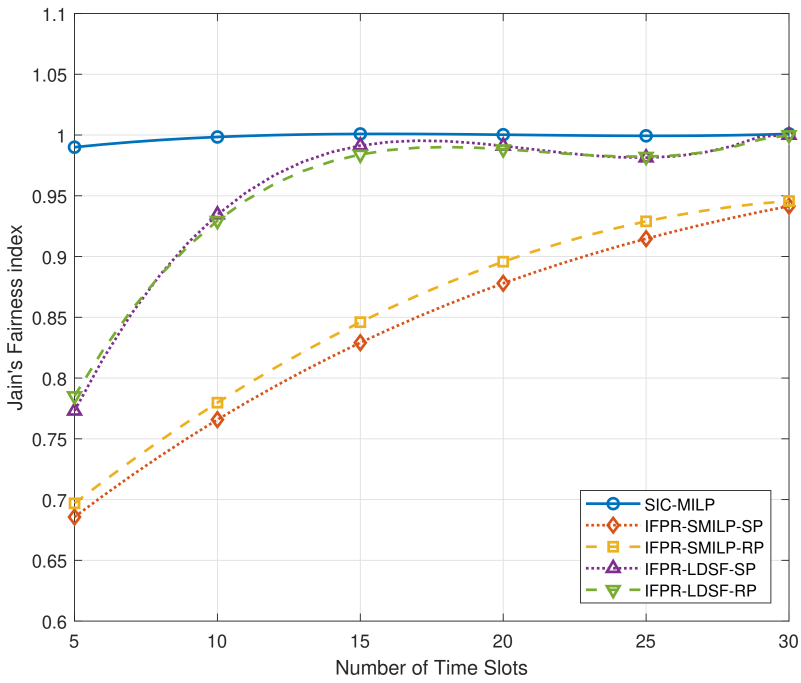

Figure 5.7: Jain's fairness index versus different number of time slots.

5.4.2.3 Total Collected Data

This section presents a study of how the number of time slots T impacts the total amount of collected data. Figure 5.8 first shows that the total amount of collected data increases linearly with more time slots. Specifically, each additional time slot increases the total amount of data for SIC-MILP, IFPR-SMILP-SP, IFPR-SMILP-RP, IFPR-LDSF-SP and IFPR-LDSF-RP by 3.26, 2.59, 2.36, 2.53 and 2.37 Mbits, respectively. This is because for each method, the number of scheduled ground devices and corresponding flow rate are the same in each time slot. Therefore, with a newly added time slot, the increase in the total amount of collected data is a constant value.

Referring to Figure 5.8, as compared to IFPR-LDSF, IFPR-SMILP IFPR-SMILP results in more uploaded data. For example, when $T = 30$, the total amount of data collected by IFPR-SMILP-SP is 5.52 Mbits higher than that of IFPR-LDSF-SP. Similarly, for the same T value, this quantity for IFPR-SMILP-RP is 3.03 Mbits higher than IFPR-LDSF-RP. This is because instead of uploading the most amount of data, IFPR-LDSF focuses on improving the fairness of flow rate from ground devices.

As shown in Figure 5.8, when the UAV randomly selects a path, it will collect less data from ground devices. For example, when $T = 30$, IFPR-SMILP-SP collects 5.75 Mbits more data than IFPR-SMILP-RP. This is because the Dijkstra algorithm computes a path with the maximum available capacity for each time slot. Therefore, ground devices are able to upload more data to the UAV. However, this is not the case when the UAV selects a random path, meaning the UAV may select a path with a low capacity and hence, it transmits less than the optimal amount of data to the gateway.

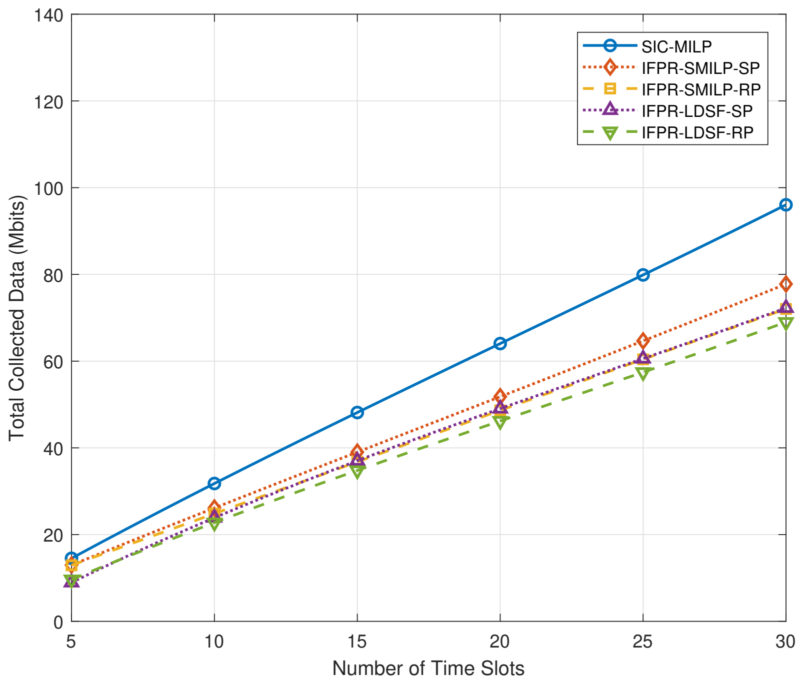

Figure 5.8: Total collected data versus different number of time slots.

5.5 Conclusion

This chapter considers data collection in a SAGIN. The problem at hand is to (i) optimize the path from a SIC-enabled UAV to a gateway, (ii) schedule ground devices as per SIC constraints, and (iii) determine the flow of each selected path. The objective is to maximize the minimum flow of ground devices over a planning time horizon. To this end, this work contains two novel solutions. The first is an MILP and the second is a novel distributed algorithm called IFPR. The numerical results indicate that with the help of CubeSats, the UAV is able to collect 61% more data from ground devices. Moreover, the total amount of collected data increases when there are more CubeSats. Further, SIC allows the UAV to schedule two simultaneous ground devices on average. Additionally, the flow rate of ground devices becomes fairer with increasing number of time slots. Lastly, compared to the optimal result obtained by solving the formulated MILP, IFPR collects only 23% less data.

Chapter 6

Conclusion

This thesis has investigated numerous link scheduling approaches for Unmanned Aerial Vehicle (UAV)-aided wireless networks, including one-tier UAV communications networks and Space-Air-Ground Integrated Networks (SAGINs). Its key aim is to collect the maximum amount of data from ground devices. As shown in this thesis, link schedulers have a direct impact on the average throughput and throughput fairness of ground devices as well as the lifetime and/or energy consumption of UAVs. Unlike existing works, this thesis considers a single rotary-wing UAV that flies at a fixed as well as different heights. Advantageously, it has a Successive Interference Cancellation (SIC) radio, which enables the UAV to collect data from multiple ground nodes/transmitters at the same time. The UAV then downloads its collected data to a terrestrial sink or gateway directly or via CubeSats acting as relays. In this respect, the main problem addressed in this thesis is to schedule the transmission of ground devices or/and optimize routing via CubeSats in order to maximize the amount of data collected by the UAV.

To this end, this thesis proposes and studies three novel problems: (i) uplink schedule optimization to a UAV, (ii) joint UAV trajectory and uplink schedule optimization, and (iii) joint routing and uplink schedule optimization in SIC-enabled SAGINs. Specifically, as per Chapter 3, a single SIC-enabled UAV collects the maximum amount of data from ground devices within a fixed time horizon by computing the optimal Time Division Multiple Access (TDMA) uplink schedule. Additionally, each ground device is required to be scheduled at least once. Chapter 3 first presents a novel Integer Linear Program (ILP) solution. However, it is intractable due to the exponential number of possible link sets that satisfy SIC constraints. Hence, this problem can be reduced from the well-known NP-hard weighted set cover problem. To solve the problem in large-scale networks, Chapter 3 proposes three novel approaches, including a Cross-Entropy (CE) based method, a heuristic algorithm Greedily Construct Transmission Set (GCTS) and a distributed Medium Access Control (MAC) Collection Point Selection Protocol (CPSP). Numerical results show that equipping a UAV with a SIC radio doubles the amount of collected data. Also, the number of ground devices and data collection points along a UAV's trajectory affect the average throughput and the fairness of ground devices. Moreover, the average throughput is also affected by the speed and height of a UAV as well as the position of ground devices. Further, CE-based method is capable of producing a schedule that is near optimal.

Another significant problem addressed in this thesis is to jointly optimize the height or trajectory of a SIC-enabled UAV and uplink schedule at each data collection point along the determined trajectory. This is significant because in most existing Non-Orthogonal Multiple Access (NOMA)-assisted UAV communications works, UAV(s) fly at a fixed height. In Chapter 4, an ILP model is first formulated to compute the optimal trajectory and data transmission schedule. The combinatorial problem in Chapter 4 can be reduced from a weighted set cover problem as well. In particular, this problem is to find multiple set covers that maximize the sum-rate (weight) subject to devices being included in at least one of these selected set covers. Thus, the formulated ILP solution is not suitable for large-scale networks. Chapter 4 then proposes two novel approaches, namely a heuristic called Iteratively Construct Link Schedule and Trajectory (ICLST) and a learning protocol based on State-Action-Reward-State-Action (SARSA). The conducted experiments in Chapter 4 consider placing devices at different heights. Numerical results show that placing devices at different elevated heights helps the UAV collect 15.8% additional data. Moreover, when the UAV flies along a trajectory with different heights, it is able to collect more data. Evaluation results also show that the average throughput of each devices is affected by the position of devices. Further, the novel heuristic ICLST is capable of producing a schedule that is near optimal. Additionally, SARSA-based learning protocol yields a schedule with the highest energy-efficiency.

Lastly, Chapter 5 studies data collection in a SIC-enabled SAGIN. Specifically, a rotary-wing UAV is equipped with a SIC radio and has connection to a terrestrial sink or gateway. This allows the UAV to download data collected from ground devices to the gateway directly. Additionally, a swarm of CubeSats act as relays to download the data offloaded from the UAV to the gateway. Therefore, Chapter 5 studies the combinatorial problem that determines (i) a path from the UAV to the gateway, (ii) transmitting ground devices or uplinks, and (iii) the flow over each active link. This problem is significant because no existing works that study SAGINs have considered multi-user detection or interference cancellation. Additionally, existing works that consider routing problems in SAGINs mostly focus on aerial or satellite segment and assume gateways or paths in other segments/networks are given. The joint routing and uplink scheduling optimization problem is first formulated as a Mixed Integer Linear Program (MILP) model. As a comparison, a novel protocol called Iterative Flow and Path Reservation (IFPR) is proposed. Specifically, IFPR considers two methods to schedule ground devices in each time slot, including a Simplified MILP (SMILP) and a greedy algorithm called Less Data Schedule First (LDSF). Additionally, IFPR randomly selects a path and/or applies Dijkstra algorithm to select a path with the least cost. This chapter investigates the impact of the following factors: the number of ground devices, CubeSats and time slots as well as the maximum data storage capacity of the UAV. Numerical results show that the performance of IFPR is close to that of the formulated MILP in experiments with varying number of time slots. However, the gap between IFPR and MILP rises when the number of time slots increases. Moreover, CubeSats help collect 61% more data compared to one-tier UAV communications. Further, for both MILP solution and IFPR, Jain's Fairness index reaches around one when the number of time slot is large.

There remains many interesting problems for future research. For example, a key assumption in Chapter 3 to 5 is block fading, where channel gain of uplinks between ground devices and the UAV remains constant for each time slot but varies across slots. Thus, a possible direction is to consider random channel gains and non-ideal SIC decoding at the UAV. Then the joint trajectory and uplink scheduling optimization problem can be cast as a stochastic or robust optimization problem. In Chapter 5, a single UAV collects data from ground devices. In addition, the trajectory of the UAV is given. Thus, another possible direction is to jointly consider trajectory design of multiple UAVs, routing and uplink scheduling to maximize the minimum flow of ground devices. Additionally, another objective is to maximize the minimum collected data of UAVs. Moreover, Chapter 5 does not consider CubeSats with energy harvesting capabilities. Thus, a possible future work is to study the impact of varying energy harvesting rates at CubeSats on the amount data collected by a gateway.

Bibliography

[1] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, “Space-air-ground integrated network: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2714–2741, 2018.

[2] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. S. Shen, “Software defined space-air-ground integrated vehicular networks: Challenges and solutions,” *IEEE Communications Magazine*, vol. 55, no. 7, pp. 101–109, 2017.

[3] N. H. Motlagh, T. Taleb, and O. Arouk, “Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives,” *IEEE Internet Things J.*, vol. 3, no. 6, pp. 899–922, Dec. 2016.

[4] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan, and Z. Ming, “A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9585–9596, 2016.

[5] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, “A tutorial on UAVs for wireless networks: Applications, challenges, and open problems,” 2019.

[6] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned

aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, 2016.

[7] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *in Proc. IEEE GLOBECOM*. IEEE, 2014, pp. 2898–2904.

[8] Finabel, *Stratobus: A Feature Hybrid of Unmanned Air Vehicles for the European Air Defence Structure*, sUAS News, Jun. 2021. [Online]. Available: https://finabel.org/stratobus-a-feature-hybrid-of-unmanned-air-vehicles-for-the-european-air-defence-structure/

[9] Airbus, *The Airbus Zephyr, Solar High Altitude Platform Station (HAPS) concludes a successful new test flight campaign in Arizona, USA*, Dec. 2020. [Online]. Available: https://www.airbus.com/newsroom/press-releases/en/2020/12/the-airbus-zephyr-solar-high-altitude-platform-station-haps-concludes-a-successful-new-test-flight-campaign-in-arizona-usa.html

[10] Google, *Project Loon*, Mar. 2018. [Online]. Available: https://www.google.com/intl/en-US/loon/

[11] Press, *Parrot unveils ANAFI Ai: The first 4G connected robotic UAV*, Feb. 2020. [Online]. Available: https://www.suasnews.com/2021/06/parrot-unveils-anafi-ai-the-first-4g-connected-robotic-uav/

[12] B. Rose, *This Changed Everything: DJI's Phantom Drone*, sUAS News, Jan. 2021. [Online]. Available: https://www.popularmechanics.com/technology/gear/a34887991/dji-phantom-drone/

[13] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, 2019.

[14] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, 2019.

[15] Y. Zeng, Q. Wu, and R. Zhang, "Accessing from the sky: A tutorial on UAV communications for 5G and beyond," *Proc. IEEE*, vol. 107, no. 12, pp. 2327–2375, 2019.

[16] A. Qiantori, A. B. Sutiono, H. Hariyanto, H. Suwa, and T. Ohta, "An emergency medical communications system by low altitude platform at the early stages of a natural disaster in indonesia," *Journal of medical systems*, vol. 36, no. 1, pp. 41–52, 2012.

[17] A. Hern *et al.*, "Facebook launches aquila solar-powered drone for internet access," *The Guardian*, vol. 30, 2015.

[18] Amazon, "Amazon prime air," accessed on: May. 2018. [Online]. Available: http://www.amazon.com/b?node=8037720011

[19] S. Subin, *Alphabet's drones delivered 10,000 cups of coffee and 1,200 roast chickens in the last year*, CNBC, Aug. 2021. [Online]. Available: https://www.cnbc.com/2021/08/25/alphabet-wing-drones-delivered-10000-cups-of-coffee-in-the-last-year.html

[20] U. S. D. of Defense, *Unmanned Systems Roadmap: 2007-2032.* Department of Defense, 2007.

[21] D. Orfanus, E. P. de Freitas, and F. Eliassen, "Self-organization as a supporting paradigm for military UAV relay networks," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 804–807, 2016.

[22] D. Nield, "Boeing's latest patent reveals a drone that can transform into a submarine," *ScienceAlert*, 2016.

[23] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, 2018.

[24] M. Y. Arafat and S. Moh, "Localization and clustering based on swarm intelligence in [uav] networks for emergency communications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8958–8976, 2019.

[25] *Unmanned Aerial Vehicle (UAV) Market Size to Reach USD 72320 Million by 2028 at a CAGR of 14.4%*, Valuates Reports, Apr. 2022. [Online]. Available: https://www.prnewswire.com/in/news-releases/unmanned-aerial-vehicle-uav-market-size-to-reach-usd-72320-million-by-2028-at-a-cagr-of-14-4-valuates-reports-870953616.html

[26] J. Korhonen, "Enhanced LTE support for aerial vehicles," *3rd Generation Partnership Project (3GPP), Tech. Rep*, 2018.

[27] "Paving the path to 5G: Optimizing commercial lte networks for drone communication," accessed on: May. 2018. [Online]. Available: https://www.qualcomm.com/news/onq/2016/09/06/paving-path-5goptimizing-commercial-lte-networks-drone-communication

[28] W. Shi, H. Zhou, J. Li, W. Xu, N. Zhang, and X. Shen, "Drone assisted vehicular networks: Architecture, challenges and opportunities," *IEEE Network*, vol. 32, no. 3, pp. 130–137, 2018.

[29] Y. Zeng, J. Lyu, and R. Zhang, "Cellular-connected UAV: Potential, challenges, and promising technologies," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 120–127, 2019.

[30] M. Mozaffari, A. Taleb Zadeh Kasgari, W. Saad, M. Bennis, and M. Debbah, "Beyond 5G with UAVs: Foundations of a 3D wireless cellular network," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 357–372, 2019.

[31] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Wireless communication using unmanned aerial vehicles (UAVs): Optimal transport theory for hover time optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 8052–8066, 2017.

[32] ——, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, 2017.

[33] B. Li, Z. Fei, Y. Dai, and Y. Zhang, "Secrecy-optimized resource allocation for UAV-assisted relaying networks," in *in Proc. IEEE GLOBECOM*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.

[34] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4983–4996, 2016.

[35] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 1, p. 7, 2011.

[36] W. Shi, J. Li, N. Cheng, F. Lyu, Y. Dai, H. Zhou, and X. S. Shen, "3d multi-drone-cell trajectory design for efficient iot data collection," in *in Proc. IEEE ICC*, Shanghai, China, May 2019, pp. 1–6.

[37] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, 2017.

[38] Y. Huang, Q. Wu, R. Lu, X. Peng, and R. Zhang, "Massive MIMO for cellular-connected UAV: Challenges and promising solutions," *IEEE Commun. Mag.*, vol. 59, no. 2, pp. 84–90, 2021.

[39] İlker Bekmezci, O. K. Sahingoz, and Şamil Temel, "Flying ad-hoc networks (FANETs): A survey," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, 2013.

[40] Q. Wang, H.-N. Dai, Q. Wang, M. K. Shukla, W. Zhang, and C. G. Soares, "On connectivity of UAV-assisted data acquisition for underwater Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5371–5385, 2020.

[41] Y. Wang, Z. Li, Y. Chen, M. Liu, X. Lyu, X. Hou, and J. Wang, "Joint resource allocation and UAV trajectory optimization for Space–Air–Ground Internet of Remote Things networks," *IEEE Syst. J.*, vol. 15, no. 4, pp. 4745–4755, 2021.

[42] X. Zhu and C. Jiang, "Integrated satellite-terrestrial networks toward 6G: Architectures, applications, and challenges," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 437–461, 2022.

[43] O. Kodheli, E. Lagunas, N. Maturo, S. K. Sharma, B. Shankar, J. F. M. Montoya, J. C. M. Duncan, D. Spano, S. Chatzinotas, S. Kisseleff, J. Querol, L. Lei, T. X. Vu, and G. Goussetis, "Satellite communications in the new space era: A survey and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 70–109, 2021.

[44] *Starlink*. [Online]. Available: https://www.starlink.com/

[45] D. Zhou, M. Sheng, J. Luo, R. Liu, J. Li, and Z. Han, "Collaborative data scheduling with joint forward and backward induction in small satellite networks," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3443–3456, 2019.

[46] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *IEEE GLOBECOM*, Austin, TX, USA, 2014, pp. 2898–2904.

[47] H. Wu, F. Lyu, C. Zhou, J. Chen, L. Wang, and X. Shen, "Optimal UAV caching and trajectory in aerial-assisted vehicular networks: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2783–2797, 2020.

[48] X. Zhu, C. Jiang, L. Kuang, and Z. Zhao, "Cooperative multilayer edge caching in integrated satellite-terrestrial networks," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2021.

[49] R. Radhakrishnan, W. W. Edmonson, F. Afghah, R. M. Rodriguez-Osorio, F. Pinto, and S. C. Burleigh, "Survey of inter-satellite communication for small satellite systems: Physical layer to network layer view," *IEEE Commun. Surveys Tuts*, vol. 18, no. 4, pp. 2442–2473, 2016.

[50] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in UAV enabled wireless sensor network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 328–331, Jun. 2018.

[51] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2016.

[52] P.-J. Wan, O. Frieder, X. Jia, F. Yao, X. Xu, and S. Tang, "Wireless link scheduling under physical interference model," in *in Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 838–845.

[53] A. Sgora, D. J. Vergados, and D. D. Vergados, "A survey of TDMA scheduling schemes in wireless multihop networks," *ACM Comput. Surv.*, vol. 47, no. 3, pp. 1–40, Apr. 2015.

[54] Y. Liu, Z. Qin, M. Elkashlan, Z. Ding, A. Nallanathan, and L. Hanzo, "Nonorthogonal multiple access for 5g and beyond," *Proc. IEEE*, vol. 105, no. 12, pp. 2347–2381, 2017.

[55] O. Maraqa, A. S. Rajasekaran, S. Al-Ahmadi, H. Yanikomeroglu, and S. M. Sait, "A survey of rate-optimal power domain NOMA with enabling technologies of future wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2192–2235, 2020.

[56] O. Goussevskaia and R. Wattenhofer, "Scheduling wireless links with successive interference cancellation," in *in Proc. International Conference on Computer Communications and Networks (ICCCN)*, Munich, Germany, 2012, pp. 1–7.

[57] L. Qu, J. He, and C. Assi, "Understanding the benefits of successive interference cancellation in multi-rate multi-hop wireless networks," *IEEE Trans. Commun.*, vol. 62, no. 7, pp. 2465–2477, 2014.

[58] S. Lv, W. Zhuang, X. Wang, and X. Zhou, "Scheduling in wireless ad hoc networks with successive interference cancellation," in *in Proc. IEEE INFOCOM*, Shanghai, China, 2011, pp. 1287–1295.

[59] J.-L. Lu, W. Shu, and M.-Y. Wu, "A survey on multipacket reception for wireless random access networks," *Journal of Computer Networks and Communications*, 2012.

[60] D. Yuan, V. Angelakis, L. Chen, E. Karipidis, and E. G. Larsson, "On optimal link activation with interference cancelation in wireless networking," *IEEE Trans. Vehicular Technol.*, vol. 62, no. 2, pp. 939–945, 2013.

[61] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: Interference cancellation for wireless LANs," in *in Proc. ACM MOBICOM*, San Francisco, CA, USA, 2008, pp. 339–350.

[62] D. Yuan, V. Angelakis, and L. Lei, "Minimum-length scheduling in wireless networks with multi-user detection and interference cancellation: Optimization and performance evaluation," in *IEEE International Conference on Communication Systems (ICCS)*, Singapore, 2012, pp. 315–319.

[63] M. Kontik and S. C. Ergen, "Scheduling in single-hop multiple access wireless networks with successive interference cancellation," *IEEE Wireless Commun. Lett.*, vol. 3, no. 2, pp. 197–200, 2014.

[64] O. Goussevskaia and R. Wattenhofer, "Scheduling with interference decoding: Complexity and algorithms," *Ad Hoc Networks*, vol. 11, no. 6, pp. 1732–1745, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870513000425

[65] S. Lv, W. Zhuang, M. Xu, X. Wang, C. Liu, and X. Zhou, "Understanding the scheduling performance in wireless networks with successive interference cancellation," *IEEE Trans. Mobile Comput.*, vol. 12, no. 8, pp. 1625–1639, 2013.

[66] M. Kontik and S. Coleri Ergen, "Scheduling in successive interference cancellation based wireless ad hoc networks," *IEEE Commun. Lett.*, vol. 19, no. 9, pp. 1524–1527, 2015.

[67] M. Mollanoori and M. Ghaderi, "Uplink scheduling in wireless networks with successive interference cancellation," *IEEE Trans. Mobile Comput.*, vol. 13, no. 5, pp. 1132–1144, 2014.

[68] S. Lv, X. Wang, and X. Zhou, "Scheduling under SINR model in ad hoc networks with successive interference cancellation," in *in Proc. IEEE GLOBECOM*, Miami, FL, USA, 2010, pp. 1–5.

[69] S. Lv, W. Zhuang, X. Wang, and X. Zhou, "Context-aware scheduling in wireless networks with successive interference cancellation," in *in Proc. IEEE ICC*, Kyoto, Japan, 2011, pp. 1–5.

[70] E. Lentz and J. Zhang, "Joint scheduling and interference cancellation in ad hoc networks," in *IEEE Military Communications Conference (MILCOM)*, vol. 1, Boston, MA, USA, 2003, pp. 711–715.

[71] E. Mete and T. Girici, "Q-learning based scheduling with successive interference cancellation," *IEEE Access*, vol. 8, pp. 172 034–172 042, 2020.

[72] A. Schrijver, *Theory of Linear and Integer Programming.* USA: John Wiley & Sons, Inc., 1986.

[73] A. Gosavi, "Reinforcement learning: A tutorial survey and recent advances," *INFORMS Journal on Computing*, vol. 21, no. 2, pp. 178–192, May 2009.

[74] P. Bjorklund, P. Varbrand, and D. Yuan, "Resource optimization of spatial TDMA in ad hoc radio networks: a column generation approach," in *in Proc. IEEE INFOCOM*, vol. 2, San Francisco, CA, USA, Apr. 2003, pp. 818–824 vol.2.

[75] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[76] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '03. New York, NY, USA: Association for Computing Machinery, 2003, p. 66–80. [Online]. Available: https://doi-org.ezproxy.uow.edu.au/10.1145/938985.938993

[77] R. Rozovsky and P. R. Kumar, "SEEDEX: A MAC protocol for ad hoc networks," in *ACM International Symposium on Mobile Ad Hoc Networking & Computing*, ser. MobiHoc '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 67–75. [Online]. Available: https://doi-org.ezproxy.uow.edu.au/10.1145/501426.501427

[78] R. Hou, Y. Xie, K.-S. Lui, and J. Li, "Scheduling in dense small cells with successive interference cancellation," *IEEE Commun. Lett.*, vol. 18, no. 6, pp. 1035–1038, 2014.

[79] M. Lei, X. Zhang, T. Zhang, L. Lei, Q. He, and D. Yuan, "Successive interference cancellation for throughput maximization in wireless powered com-

munication networks," in *IEEE Vehicular Technology Conference (VTC-Fall)*, Montreal, QC, Canada, 2016, pp. 1–6.

[80] Q. He, D. Yuan, and A. Ephremides, "Maximum link activation with cooperative transmission and interference cancellation in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 2, pp. 408–421, 2017.

[81] M. Li, Z. Li, and A. V. Vasilakos, "A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues," *Proc. IEEE*, vol. 101, no. 12, pp. 2538–2557, 2013.

[82] E. Gelal, J. Ning, K. Pelechrinis, T.-S. Kim, I. Broustis, S. V. Krishnamurthy, and B. D. Rao, "Topology control for effective interference cancellation in multiuser MIMO networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 2, pp. 455–468, 2013.

[83] M. Sheng, X. Li, X. Wang, and C. Xu, "Topology control with successive interference cancellation in cognitive radio networks," *IEEE Trans. Commun.*, vol. 65, no. 1, pp. 37–48, 2017.

[84] E. Karipidis, D. Yuan, Q. He, and E. G. Larsson, "Max–min power control in wireless networks with successive interference cancelation," *IEEE Trans. Wireless Commun.*, vol. 14, no. 11, pp. 6269–6282, 2015.

[85] X. Li, Y. Shi, X. Wang, C. Xu, and M. Sheng, "Efficient link scheduling with joint power control and successive interference cancellation in wireless networks," *Science China Information Sciences*, vol. 59, no. 12, p. 122301, 2016.

[86] C. Xu, K. Ma, Y. Xu, Y. Xu, and Y. Fang, "Optimal power scheduling for uplink transmissions in sic-based industrial wireless networks with guaranteed real-time performance," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 3216–3228, 2018.

[87] D. Gözüpek, M. Shalom, and F. Alagöz, "A graph-theoretic approach to scheduling in cognitive radio networks," *IEEE/ACM Trans. Netw.*, vol. 23, no. 1, pp. 317–328, 2015.

[88] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge, UK.: Cambridge Univ. Press, 2004.

[89] L. Qu, J. He, and C. Assi, "Congestion control, routing, and scheduling in wireless networks with interference cancelation capabilities," *IEEE Trans. Vehicular Technol.*, vol. 64, no. 7, pp. 3108–3119, 2015.

[90] M. Ploumidis, N. Pappas, and A. Traganitis, "Performance of flow allocation with successive interference cancelation for random access WMNs," in *IEEE Globecom Workshops (GC Wkshps)*, San Diego, CA, USA, 2015, pp. 1–7.

[91] C. Jiang, Y. Shi, X. Qin, X. Yuan, Y. T. Hou, W. Lou, S. Kompella, and S. F. Midkiff, "Cross-layer optimization for multi-hop wireless networks with successive interference cancellation," *IEEE Trans. Wireless Commun.*, vol. 15, no. 8, pp. 5819–5831, 2016.

[92] R. Liu, Y. Shi, K.-S. Lui, M. Sheng, Y. Wang, and Y. Li, "Bandwidth-aware high-throughput routing with successive interference cancelation in multihop wireless networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5866–5877, 2015.

[93] J. Cheng, P. Yang, K. Navaie, Q. Ni, and H. Yang, "A low-latency interference coordinated routing for wireless multi-hop networks," *IEEE Sens. J.*, vol. 21, no. 6, pp. 8679–8690, 2021.

[94] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, pp. 1132–1145, 2009.

[95] A. A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, "A survey of channel modeling for UAV communications," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2804–2821, 2018.

[96] X. Li, Q. Wang, J. Liu, and W. Zhang, "Trajectory design and generalization for UAV enabled networks:a deep reinforcement learning approach," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Seoul, Korea (South), 2020, pp. 1–6.

[97] D.-H. Tran, T. X. Vu, S. Chatzinotas, S. ShahbazPanahi, and B. Ottersten, "Coarse trajectory design for energy minimization in UAV-enabled wireless communications with latency constraints," *IEEE Trans. Vehicular Technol.*, vol. 69, no. 9, pp. 9483–9496, 2020.

[98] J. Cui, Z. Ding, Y. Deng, and A. Nallanathan, "Model-free based automated trajectory optimization for UAVs toward data transmission," in *in Proc. IEEE GLOBECOM*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[99] D. Song, X. B. Zhai, X. Liu, and C. W. Tan, "Jointly optimal fair data collection and trajectory design algorithms in uav-aided cellular networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Nanjing, China, Apr. 2021, pp. 1–6.

[100] C. Luo, M. N. Satpute, D. Li, Y. Wang, W. Chen, and W. Wu, "Fine-grained trajectory optimization of multiple UAVs for efficient data gathering from WSNs," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 162–175, 2021.

[101] S.-F. Chou, A.-C. Pang, and Y.-J. Yu, "Energy-aware 3d unmanned aerial vehicle deployment for network throughput optimization," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 563–578, 2020.

[102] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "Multi-uav path plan-

ning for wireless data harvesting with deep reinforcement learning," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 1171–1187, 2021.

[103] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *Foundations and Trends in Machine Learning*, vol. 11, no. 3-4, p. 219–354, 2018. [Online]. Available: http://dx.doi.org/10.1561/2200000071

[104] Y. Dumas, J. Desrosiers, E. Gelinas, and M. M. Solomon, "An optimal algorithm for the traveling salesman problem with time windows," *Opeartions Res.*, vol. 43, no. 2, pp. 367–371, Apr. 1995.

[105] T. Jia and R. M. Buehrer, "Collaborative position location for wireless networks using iterative parallel projection method," in *in Proc. IEEE GLOBECOM*, Miami, FL, USA, Dec. 2010, pp. 1–6.

[106] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *Lecture notes of EE392o, Stanford Univ., Stanford, CA, USA*, Oct. 2003.

[107] F. A. Oliehoek and C. Amato, *A Concise Introduction to Decentralized POMDPs*, 1st ed. Springer, Cham, 2016.

[108] Z. Ullah, F. Al-Turjman, U. Moatasim, L. Mostarda, and R. Gagliardi, "UAVs joint optimization problems and machine learning to improve the 5G and beyond communication," *Computer Networks*, vol. 182, p. 107478, Dec. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620311518

[109] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for UAV-enabled multiple access," in *in Proc. IEEE GLOBECOM*, Singapore, Dec. 2017, pp. 1–6.

[110] Z. Wang, R. Liu, Q. Liu, J. S. Thompson, and M. Kadoch, "Energy-efficient

data collection and device positioning in UAV-assisted iot," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1122–1139, 2020.

[111] C. You and R. Zhang, "3D trajectory optimization in rician fading for UAV-enabled data harvesting," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3192–3207, 2019.

[112] M. B. Ghorbel, D. Rodríguez-Duarte, H. Ghazzai, M. J. Hossain, and H. Menouar, "Joint position and travel path optimization for energy efficient wireless data gathering using unmanned aerial vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2165–2175, Mar. 2019.

[113] O. Bouhamed, H. Ghazzai, H. Besbes, and Y. Massoud, "A UAV-assisted data collection for wireless sensor networks: Autonomous navigation and scheduling," *IEEE Access*, vol. 8, pp. 110 446–110 460, 2020.

[114] C. Zhan and Y. Zeng, "Completion time minimization for multi-UAV-enabled data collection," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4859–4872, 2019.

[115] Q. Wu and R. Zhang, "Common throughput maximization in UAV-enabled OFDMA systems with delay consideration," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6614–6627, 2018.

[116] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305048304001550

[117] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement

learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. [Online]. Available: https://doi.org/10.1038/nature14236

[118] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, 2018.

[119] J. Lyu, Y. Zeng, and R. Zhang, "UAV-aided offloading for cellular hotspot," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3988–4001, 2018.

[120] M. Hua, Y. Wang, Z. Zhang, C. Li, Y. Huang, and L. Yang, "Power-efficient communication in UAV-aided wireless sensor networks," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1264–1267, 2018.

[121] C. Zhan and Y. Zeng, "Energy-efficient data uploading for cellular-connected UAV systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7279–7292, 2020.

[122] J. Hou, Y. Deng, and M. Shikh-Bahaei, "Joint beamforming, user association, and height control for cellular-enabled UAV communications," *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 3598–3613, 2021.

[123] J. Li, H. Zhao, H. Wang, F. Gu, J. Wei, H. Yin, and B. Ren, "Joint optimization on trajectory, altitude, velocity, and link scheduling for minimum mission time in UAV-aided data collection," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1464–1475, 2020.

[124] J. Lyu, Y. Zeng, and R. Zhang, "Cyclical multiple access in UAV-aided communications: A throughput-delay tradeoff," *IEEE Wireless Commun. Lett.*, vol. 5, no. 6, pp. 600–603, 2016.

[125] J. Zhang, Y. Zeng, and R. Zhang, "Receding horizon optimization for energy-efficient uav communication," *IEEE Wireless Commun. Lett.*, vol. 9, no. 4, pp. 490–494, 2020.

[126] M. Sanjabi, M. Razaviyayn, and Z.-Q. Luo, "Optimal joint base station assignment and beamforming for heterogeneous networks," *IEEE Trans. Signal Process.*, vol. 62, no. 8, pp. 1950–1961, 2014.

[127] S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Optim. Eng.*, vol. 8, no. 1, pp. 67–127, 2007.

[128] M. Tao, E. Chen, H. Zhou, and W. Yu, "Content-centric sparse multicast beamforming for cache-enabled cloud RAN," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6118–6131, 2016.

[129] J. Sun, Z. Wang, and Q. Huang, "Cyclical NOMA based UAV-enabled wireless network," *IEEE Access*, vol. 7, pp. 4248–4259, 2019.

[130] X. Wu, Z. Wei, Z. Cheng, and X. Zhang, "Joint optimization of UAV trajectory and user scheduling based on NOMA technology," in *in Proc. IEEE WCNC*, Seoul, Korea (South), May 2020, pp. 1–6.

[131] X. Mu, Y. Liu, L. Guo, and J. Lin, "Non-orthogonal multiple access for air-to-ground communication," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2934–2949, 2020.

[132] T. Shen and H. Ochiai, "A uav-enabled wireless powered sensor network based on noma and cooperative relaying with altitude optimization," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 21–34, 2021.

[133] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," 2014. [Online]. Available: http://cvxr.com/cvx

[134] D. B. West, *Introduction to Graph Theory.* Prentice Hall, 1996.

[135] A. A. Nasir, H. D. Tuan, T. Q. Duong, and H. V. Poor, "UAV-enabled communication using NOMA," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 5126–5138, 2019.

[136] N. Zhao, X. Pang, Z. Li, Y. Chen, F. Li, Z. Ding, and M.-S. Alouini, "Joint trajectory and precoding optimization for UAV-assisted NOMA networks," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3723–3735, 2019.

[137] H. Zeng, X. Zhu, Y. Jiang, and Z. Wei, "UAV-ground BS coordinated NOMA with joint user scheduling, power allocation and trajectory design," in *in Proc. IEEE ICC*, Dublin, Ireland, Jul. 2020, pp. 1–6.

[138] J. Zhao, Y. Wang, Z. Fei, X. Wang, and Z. Miao, "NOMA-aided UAV data collection system: Trajectory optimization and communication design," *IEEE Access*, vol. 8, pp. 155 843–155 858, 2020.

[139] W. Wang, N. Zhao, L. Chen, X. Liu, Y. Chen, and D. Niyato, "Time-efficient uplink data collection for UAV-assisted NOMA networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Nanjing, China, Apr. 2021, pp. 1–6.

[140] Y. Li, H. Zhang, K. Long, C. Jiang, and M. Guizani, "Joint resource allocation and trajectory optimization with QoS in UAV-based NOMA wireless networks," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2021.

[141] H. H. M. Tam, H. D. Tuan, and D. T. Ngo, "Successive convex quadratic programming for quality-of-service management in full-duplex mu-mimo multicell networks," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2340–2353, 2016.

[142] F. Cheng, S. Zhang, Z. Li, Y. Chen, N. Zhao, F. R. Yu, and V. C. M. Leung, "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6732–6736, 2018.

[143] A. M. Geoffrion, "Generalized benders decomposition," *Journal of optimization theory and applications*, vol. 10, no. 4, pp. 237–260, Oct. 1972.

[144] M. Bagajewicz and V. Manousiouthakis, "On the generalized benders decomposition," *Computers & Chemical Engineering*, vol. 15, no. 10, pp.

691–700, Oct. 1991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/009813549185015M

[145] A. E. Roth and M. A. O. Sotomayor, *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*, ser. Econometric Society Monographs. Cambridge University Press, 1990.

[146] J. Papandriopoulos and J. S. Evans, "SCALE: A low-complexity distributed protocol for spectrum balancing in multiuser DSL networks," *IEEE Trans. Inf. Theory*, vol. 55, no. 8, pp. 3711–3724, 2009.

[147] P. Pace and G. Aloi, "Effective routing algorithm for multilayered terrestrial-HAP-satellite networks," *IEEE Commun. Lett.*, vol. 11, no. 6, pp. 510–512, 2007.

[148] ——, "Routing and scalability issues for multi-layered satellite-HAPs networks," in *IEEE International Conference on Advances in Satellite and Space Communications*, Athens, Greece, Jun. 2010, pp. 64–69.

[149] Y. Shi, J. Liu, Z. M. Fadlullah, and N. Kato, "Cross-layer data delivery in satellite-aerial-terrestrial communication," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 138–143, 2018.

[150] N. Kato, Z. M. Fadlullah, F. Tang, B. Mao, S. Tani, A. Okamura, and J. Liu, "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 140–147, 2019.

[151] A. Varasteh, S. Hofmann, N. Deric, M. He, D. Schupke, W. Kellerer, and C. M. Machuca, "Mobility-aware joint service placement and routing in space-air-ground integrated networks," in *in Proc. IEEE ICC*, Shanghai, China, May 2019, pp. 1–7.

[152] Q. Dong, S. Banerjee, M. Adler, and A. Misra, "Minimum energy reliable paths using unreliable wireless links," in *Proc. ACM MobiHoc*, ser. MobiHoc

'05. New York, NY, USA: Association for Computing Machinery, 2005, p. 449–459. [Online]. Available: https://doi-org.ezproxy.uow.edu.au/10.1145/1062689.1062744

[153] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[154] F. Tang, B. Mao, Z. M. Fadlullah, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "On removing routing protocol from future wireless networks: A real-time deep learning approach for intelligent traffic control," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 154–160, 2018.

[155] D. Tagliaferri and C. Capsoni, "High-speed wireless infrared uplink scheme for airplane passengers' communications," *Electronics Letters*, vol. 53, no. 13, pp. 887–888, 2017.

[156] H. Jia, Y. Wang, M. Liu, and Y. Chen, "Sum-rate maximization for UAV aided wireless power transfer in space-air-ground networks," *IEEE Access*, vol. 8, pp. 216 231–216 244, 2020.

[157] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for iot applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, 2019.

[158] C. Zhou, W. Wu, H. He, P. Yang, F. Lyu, N. Cheng, and X. Shen, "Deep reinforcement learning for delay-oriented iot task scheduling in SAGIN," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 911–925, 2021.

[159] S. Mao, S. He, and J. Wu, "Joint UAV position optimization and resource scheduling in space-air-ground integrated networks with mixed cloud-edge computing," *IEEE Syst. J.*, pp. 1–11, 2020.

[160] H. Jia, Y. Wang, Y. Chen, M. Liu, and Z. Li, "Sum rate maximization for multi-UAV enabled space-air-ground wireless powered communication networks," in *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Nanjing, China, Mar. 2021, pp. 1–6.

[161] Y. Wang, Z. Li, Y. Chen, M. Liu, X. Lyu, X. Hou, and J. Wang, "Joint resource allocation and UAV trajectory optimization for space–air–ground internet of remote things networks," *IEEE Syst. J.*, pp. 1–11, 2020.

[162] F. Vanderbeck and M. W. Savelsbergh, "A generic view of dantzig–wolfe decomposition in mixed integer programming," *Operations Research Letters*, vol. 34, no. 3, pp. 296–306, 2006.

[163] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[164] C. Guo, B. Liao, L. Huang, Q. Li, and X. Lin, "Convexity of fairness-aware resource allocation in wireless powered communication networks," *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 474–477, 2016.

[165] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, Feb 2005. [Online]. Available: https://doi.org/10.1007/s10479-005-5724-z

[166] *"Cisco Wireless Mesh Access Points, Design and Deployment Guide, Release 7.3 x"*, www.cisco.com, Aug. 2012. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/wireless/technology/mesh/7-3/design/guide/Mesh.pdf

[167] U. Feige, "A threshold of ln n for approximating set cover," *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.

[168] *Matlab*. [Online]. Available: https://au.mathworks.com/products/matlab

[169] A. B. Sediq, R. H. Gohary, R. Schoenen, and H. Yanikomeroglu, "Optimal tradeoff between sum-rate efficiency and Jain's fairness index in resource allocation," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3496–3509, July 2013.

[170] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, 2014.

[171] D. Yang, Q. Wu, Y. Zeng, and R. Zhang, "Energy tradeoff in ground-to-UAV communication via trajectory design," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6721–6726, 2018.

[172] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[173] R. L. Streit, *The Poisson Point Process.* Boston, MA: Springer US, 2010, pp. 11–55. [Online]. Available: https://doi.org/10.1007/978-1-4419-6923-1_2

[174] T.D.Cola, A.Ginesi, G.Giambene, G.C.Polyzos, N.Fotiou, V.A.Siris, and Y.Thomas, *Network and Protocol Architectures for Future Satellite Systems.* Now Foundations and Trends(r) in Networking Series, 2017.

[175] M. A. Bergamo, "High-throughput distributed spacecraft network: architecture and multiple access technologies," *Computer Networks*, vol. 47, no. 5, pp. 725–749, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128604002221

[176] M. Viswanathan, *Wireless Communication Systems in Matlab.* Independently published, 2018.

[177] M. Asghari, A. M. Fathollahi-Fard, S. M. J. Mirzapour Al-e hashem, and M. A. Dulebenets, "Transformation and linearization techniques in

optimization: A state-of-the-art survey," *Mathematics*, vol. 10, no. 2, 2022. [Online]. Available: https://www.mdpi.com/2227-7390/10/2/283

[178] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[179] J. Alvarez and B. Walls, "Constellations, clusters, and communication technology: Expanding small satellite access to space," in *IEEE Aerospace Conference*, Big Sky, MT, USA, Jun. 2016, pp. 1–11.