

*Digital Comprehensive Summaries of Uppsala Dissertations  
from the Faculty of Science and Technology 2638*

# Exploring Resource Allocation for Evolving Wireless and Mobile Computing Systems

YI ZHAO



ACTA UNIVERSITATIS  
UPSALIENSIS  
2026



UPPSALA  
UNIVERSITET

Dissertation presented at Uppsala University to be publicly examined in 101121, Sonja Lyttkens, Ångströmlaboratoriet, Regemenstvägen 10, Uppsala, Wednesday, 25 March 2026 at 13:15 for the degree of Doctor of Philosophy. The examination will be conducted in English. Faculty examiner: Professor Viktoria Fodor (Department for Network and Systems Engineering, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology).

### **Abstract**

Zhao, Y. 2026. Exploring Resource Allocation for Evolving Wireless and Mobile Computing Systems. *Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology* 2638. 50 pp. Uppsala: Acta Universitatis Upsaliensis. ISBN 978-91-513-2736-5.

Wireless communication and computing systems, spanning wireless mobile networks, edge computing infrastructures, and satellite networks, are becoming increasingly heterogeneous, dynamic, and data-intensive. Despite differences in technologies and application scenarios, these systems share a fundamental challenge: The need to accommodate a variety of services using limited system resources. This dissertation addresses this challenge by developing resource allocation strategies to achieve enhanced services and support the emerging ones: 1) Allocation of radio resource for wireless mobile networks, 2) caching, recommendation, and computation of contents for edge computing, and 3) allocation of models, computing resource, and inter-satellite link capacity for federated learning (FL) in satellite networks.

The second part of this dissertation comprises six papers. For research topic 1), Paper I investigates an array of integer linear programming models for radio channel allocation under coupled rate constraints, the relationships between the models' linear programming approximation. Paper II presents a method based on the derived closed-form solutions and matching theory, for the allocation of frequency-time resource blocks, towards throughput maximization in multi-cell non-orthogonal multiple access (NOMA) scenarios with interference coupling. For research topic 2), Paper III derives approximation algorithms based on problem decomposition and submodularity for cache-hit-ratio maximization, via jointly optimizing content caching and recommendation at network edge. Paper IV considers in particular artificial intelligence (AI)-generated contents and utilizes convex optimization for the allocation of content delivery mode, computing resource, and communication resource, for total utility maximization. For research topic 3), both Paper V and Paper VI present solutions of client selection and inter-satellite routing for fast convergence of FL, based on derived upper bounds of the empirical risk, convergence analysis, and network flow optimization. Paper V focuses on the classical FL framework, while Paper VI designs a novel FL architecture with knowledge distillation, accommodating both the teacher and student models.

Taken together, the dissertation demonstrates that resource allocation, guided by optimization techniques, is a unifying thread connecting wireless, edge, and satellite systems, to deliver enhanced and emerging services.

*Keywords:* mobile computing, wireless communication, resource allocation, mathematical optimization

*Yi Zhao, Department of Information Technology, Computing Science, Box 337, Uppsala University, SE-75105 Uppsala, Sweden.*

© Yi Zhao 2026

ISSN 1651-6214

ISBN 978-91-513-2736-5

URN urn:nbn:se:uu:diva-578505 (<http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-578505>)

*To myself.*



# List of papers

This thesis is based on the following papers, which are referred to in the text by their Roman numerals.

- I **Y. Zhao** and D. Yuan, "On optimization formulations for radio resource allocation subject to common transmission rate," *Computers & Operations Research*, vol. 161, pp. 106427, 2024.
- II **Y. Zhao**, D. Yuan and L. You, "Delivering more to cell edge via joint multi-cell NOMA and traffic offloading," *IET Networks*, vol. 10, no. 6, pp. 265-277, 2021.
- III **Y. Zhao**, Z. Yu, and D. Yuan, "Caching with personalized and incumbent-aware recommendation: Modeling and optimization," *IEEE Transactions on Mobile Computing (TMC)*, vol. 23, no. 10, pp. 9595–9613, 2024.
- IV **Y. Zhao**, D. Yuan, X. Chu, and S. Sun, "What to deliver? When resource allocation meets AIGC on network edge and user device," in *Proceedings of 2025 IEEE Global Communications Conference (Globecom)*, 2025.
- V **Y. Zhao**, Z. Yu, C. Feng, L. You, L. Lei, and D. Yuan, "Orchestrating in the sky: Joint routing and client selection for federated learning in LEO networks," in *Proceedings of 2025 IEEE Global Communications Conference (Globecom)*, 2025.
- VI **Y. Zhao**, L. You, L. Lei, T. Deng, and D. Yuan, "Distilling Intelligence in Space: Optimized Dual Federated Learning amid Orbital Dynamics", *Preprint*

Reprints were made with permission from the publishers.

# List of other Papers

The following papers were written by the author during the PhD study, but not included in this dissertation.

- 1 **Y. Zhao**, Z. Yu, T. Deng, and D. Yuan, "Robust online temperature management for passively cooled base stations," in *Proceedings of IEEE 99th Vehicular Technology Conference (VTC-Spring)*, 2024.
- 2 **Y. Zhao**, Z. Yu, Q. He, and D. Yuan, "Content caching with personalized and incumbent-aware recommendation: An optimization approach," in *Proceedings of the 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, 2022.
- 3 **Y. Zhao**, Z. Yu, T. Deng, and D. Yuan, "Adjustable robust optimization for passively cooled base stations in mobile communications," *Submitted*.
- 4 Z. Yu, **Y. Zhao**, X. Chu, and D. Yuan, "Online learning for intelligent thermal management of interference-coupled and passively cooled base stations," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 3, pp. 64-79, 2024.
- 5 Z. Yu, **Y. Zhao**, T. Deng, L. You, and D. Yuan, "Less carbon footprint in edge computing by joint task offloading and energy sharing," *IEEE Networking Letters*, vol. 5, no. 4, pp. 245-249, 2023.
- 6 Z. Yu, **Y. Zhao**, L. You, and D. Yuan, "Learn to stay cool: Online load management for passively cooled base stations," in *Proceedings of 2024 IEEE Wireless Communications and Networking Conference (WCNC)*, 2024.
- 7 Z. Yu, **Y. Zhao**, and D. Yuan, "Robust divergence angle for inter-satellite laser communications under target deviation uncertainty," in *Proceedings of IEEE 98th Vehicular Technology Conference (VTC-Fall)*, 2023.
- 8 Z. Yu, **Y. Zhao**, and D. Yuan, "Multi-cell caching: Fresh information with minimum cost," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, 2022.

- 9 Z. Yu, T. Deng, **Y. Zhao**, and D. Yuan, "Multi-cell content caching: Optimization for cost and information freshness," *Computer Networks*, vol. 247, pp. 110420, 2024.



# Sammanfattning

Trådlösa kommunikations- och beräkningssystem, som omfattar trådlösa mobilnät, edge infrastrukturer (även känd som utkanten av nätverket) och satellitnätverk, blir alltmer heterogena, dynamiska och dataintensiva. Trots skillnader i teknologier och tillämpningsscenarier delar dessa system en grundläggande utmaning: behovet av att tillhandahålla en mångfald av tjänster med begränsade systemresurser. Denna avhandling behandlar denna utmaning genom att utveckla resursallokeringsstrategier för att möjliggöra förbättrade tjänster samt stödja nya framväxande tjänster, nämligen: 1) allokering av radioresurser i trådlösa mobilnät, 2) cachelagring, rekommendation och generering av medie innehåll nätverksutkant, samt 3) allokering av modeller, beräkningsresurser och kapacitet för inter-satellitlänkar för federerad inläring (FL) i satellitnätverk.

Den andra delen av denna avhandling består av sex artiklar. Den första artikeln studerar problemet av radioresursallokering under begränsat spektrum. Inledningsvis genomförs en komplexitetsanalys. Därefter härleds ett antal heltalsprogrameringsmodeller för problemet, inklusive såväl kompakta som icke-kompakta modeller. Vidare presenteras en studie av deras linjärprogramering (LP) relaxationer för att klarlägga relationerna mellan formuleringarna ur hur starka övre gränser de leder till. Numeriska resultat avseende denna aspekt samt LP-assisterad problemlösning redovisas. Studien utgör en grund för nästa steg, nämligen utvecklingen av modellbaserade och skräddarsydda schemalaggningsalgoritmer på millisekundnivå.

Den andra artikeln behandlar tekniken icke-ortogonal multipel åtkomst (non-orthogonal multiple access, NOMA) och trafikavlastning i syfte att förbättra genomströmningen för användare vid cellkanten i ett mobilt nät. Vi formulerar ett problem för maximering av genomströmning, där resursblockstilldelning, effektstilldelning, användarparning och avlastning optimeras med beaktande av interferens. Därefter utvecklas en iterativ algoritm baserad på problemuppdelning och strukturell analys. Numeriska resultat visar att det föreslagna metoden avsevärt förbättrar genomströmningen för cellkantsanvändare jämfört med konventionella metoder.

Den tredje artikeln behandlar optimering av cachelagring och innehållsrekommendation. Optimeringen tar hänsyn till så kallade incument, so anser det medieinnehåll medie som användaren för närvarande konsumerar, vilket återspeglar användarens intresse. Vi undersöker samspelet mellan cachelagring och rekommendation genom innehållspopularitet. Optimeringsproblemet handlar om maximering av cacheeffektivitet med krav på användarnöjdhet bevisas att vara NP-svårt, och en heltalsprogrameerings-formulering samt

tre polynomiska algoritmer har framtagits. De två första algoritmerna bygger på submodularitet och erbjuder approximationsgarantier under milda villkor, medan den tredje är en alternerande algoritm med snabb konvergens. Numeriska resultat visar att de föreslagna algoritmerna uppnår nära-optimala resultat.

Den fjärde artikeln optimerar tilldelning av kommunikationsresurser, beräkningsresurser samt val av leveranssätt för AI-genererade medieinnehåll (känd som AIGC). AIGC-innehåll kan cachelagras vid nätverks utkant, eller genereras där, eller genereras på användarenheten. Vi betraktar avvägningarna mellan innehållskvalitet och resursförbrukning och utvecklar en effektiv lösning smetod baserad på konvex optimering och Lagrangerelaxation, med nära-optimal problemlösningar.

Den femte artikeln undersöker klienturval (client selection, CS) och ruttplanering för federerad inläring i ett satellitkommunikationssystem med syfte att uppnå snabb konvergens. Vi tar fram och minimerar en övre gräns för den globala empiriska förlusten som vår målfunktion. Den begränsade kapaciteten i ruttplaneringsproblemet modelleras och behandlas med hjälp av tidsvarierande grafer och nätverksflödesoptimering. Både exakta och approximativa lösningar föreslås för optimeringsproblemet som innehåller både klienturval och ruttplanering. Vidare formaliseras och bevisas konvergensegenskaperna hos den föreslagna metoden. Slutligen visas effektiviteten och överlägsenheten av det föreslagna metoden i realistiska satellitnätverksscenarioer via simuleringar.

Den sjätte artikeln, som är en vidareutveckling av den femte artikeln, tar ytterligare ett steg framåt, och behandlar heterogena inlärnings modeller och kunskapsdestillering (knowledge distillation, KD) för energieffektivitet och snabb konvergens för federerad inläring. Vi utvecklar ett KD-assisterat dual-FL-ramverk med konvergensanalys, härleder övre gränser för den globala empiriska förlusten för båda modellerna samt optimerar både klienturval och ruttoptimering. Simuleringar baserade på verkliga satellitkonstellationer visar att dual-modell-inläring med KD avsevärt förbättrar effektiviteten. Sammantaget visar avhandlingen att resurstilldelning, med hjälp av optimeringsmetoder, utgör en sammanhållande ground som förenar trådlösa kommunikation, samt intelligence på nätverksutkant och satellitsystem för att förbättra nuvarande tjänster och möjliggöra nya tjänster i framtiden.

# Contents

Acknowledgments .....	13
1 Introduction .....	15
1.1 Background .....	15
1.2 Motivation .....	15
1.3 Organization .....	16
2 Scenarios and Technologies .....	17
2.1 Mobile Communications .....	17
2.1.1 uRLLC and Ultra-Fast Scheduling .....	17
2.1.2 mMTC and NOMA .....	18
2.1.3 eMBB and Satellite Networks .....	18
2.2 Edge Computing .....	19
2.2.1 Edge Caching .....	20
2.2.2 Recommendation .....	21
2.2.3 AIGC .....	22
2.2.4 Federated Learning .....	23
2.2.5 Knowledge Distillation .....	24
3 Optimization Methodologies .....	27
3.1 Convex Optimization .....	27
3.2 Linear Optimization .....	28
3.2.1 LP .....	28
3.2.2 ILP and MILP .....	29
3.3 Lagrangian Relaxation .....	31
3.4 Network Flows .....	34
3.4.1 Maximum Flow .....	34
3.4.2 Minimum Cost Flow .....	35
3.4.3 Multi-Commodity Flows .....	36
3.4.4 Matching .....	36
3.5 Approximation Algorithms .....	37
3.6 Greedy Heuristics .....	38
4 Scope and Contributions .....	41
4.1 Scope .....	41
4.2 Contributions .....	41
References .....	45



# Acknowledgments

Completing this thesis has been a challenging yet rewarding journey, and I would like to express my sincere gratitude for the support of many people along the way.

First and foremost, I wish to express my profound gratitude to my main supervisor Di Yuan. During the five years, I have learned a lot from him, including professional knowledge, rigorous academic attitude, and positive mindset of facing challenges. His guidance is very supportive while maintaining a free and exploratory research atmosphere. I also appreciate his tolerance of my shortcomings and encouragement at the moments of my frustration and depression. In addition, I want to say thank you truly to my vice-supervisor Justin Pearson for his guidance on English language and interesting recommendations.

Second, I want to express my sincere gratitude to the co-authors of my papers for their contributions, and they are: Zhanwei Yu, Tao Deng, You Lei, Lei Lei, Qing He, Chenyuan Feng, Xiaoli Chu, and Sumei Sun. In particular, as the closest collaborator, thanks a lot to Zhanwei Yu for the fruitful academic cooperations as well as his genuine help on practical matters.

Third, I would like to extend my sincere gratitude to the opponent and committee members of the doctoral defense, and also the anonymous reviewers of my papers. Many thanks for their time and effort on reading my research work and giving the valuable comments.

Fourth, I want to say thank you to some colleagues in our division. Thanks a lot to Sven-Olof Nyström, Sofia Ouhbi, Georgios Fakas, and Maria Andreina Francisco Rodriguez for the good cooperation on teaching. Many thanks to Pierre Flener, Sofia Ouhbi, and Paul Fiterau Brostean for their kind support and active participation in my activities. In particular, many thanks to Frej Knutar Lewander for his positive effort on involving me into the local cultural fabric. All the above factors, although not directly relevant to the thesis, indeed generally fueled my passion for research.

In addition, I would also like to express the deep gratitude to my family members and friends for their remote support, in particular to my mother Yongmei Zhao (Mom, I love you.) and an old friend Chu Wang.

Last but not least, I am humbly grateful to virtual characters, in novels, anime, and TV series, who gave me many inspirations in thinking about the form of existence of a person. These thoughts support me through the challenges along the way. The greatness of art lies in that its content may be false, its creator may be flawed, but the readers it inspired toward idealism are existing for real. Taking root in the soil of illusion but growing into reality is the

most beautiful miracle. The fact that I discovered (on 2018-01-22), managed to come, and completed my PhD studies here is such a proof, for which I am and will be proud of myself for the rest of my life.

Yi Zhao  
2026.02.11

# 1. Introduction

## 1.1 Background

To meet the demands of low latency and high reliability, seamless connectivity, and data-intensive applications, technologies of wireless communication [1] and edge computing [2] are evolving toward intelligence and integration.

*Wireless Communication:* It refers to the technology of transmitting information via electromagnetic waves without physical cables. For beyond-5G wireless communication, ultra-reliable low latency communication (uRLLC) [3], enhanced mobile broadband (eMBB) [4], and massive machine type communications (mMTC) [5] are three key scenarios. URLLC networks with millisecond-level latency can support mission-critical applications. For such networks, the requirement on radio resource scheduling has to be ultra-fast. Among the technologies enabling mMTC networks, non-orthogonal multiple access (NOMA) [6] is an advanced one that enhances spectral efficiency, connection capacity, and user fairness. For eMBB networks aiming at large and stable capacity, satellite networks [7] can provide wide-area and seamless broadband coverage, filling the gaps of terrestrial networks.

*Edge Computing:* The exponential growth of data traffic in communication networks has fundamentally challenged the traditional cloud-centric computing paradigm [8] and further catalyzed the emergence of the concept of edge computing. It refers to a distributed architecture that brings computational resources and data storage closer (e.g., in a base station) to users. By processing data at the network edge, this paradigm significantly reduces delivery latency and backhaul resource consumption. Some paradigms spurred by edge computing include edge caching [9], content recommendation [10], AI generated content (AIGC) [11], federated learning (FL) [12], and knowledge distillation (KD) [13].

The integration of evolving wireless communication and edge computing systems stimulates novel applications such as on-edge production of AIGC and FL in satellite networks, but also brings a fundamental challenge of accommodating a variety of services using multiple, limited, heterogeneous, and dynamic system resources. Here the types of resource include communication resource, computing resource, storage resource.

## 1.2 Motivation

This dissertation addresses how to perform resource allocation to support enhanced and emerging services. While the research papers included in this

dissertation explore a diversity of scenarios and problem settings of resource allocation, they share the following common aspects. First, the systems all process and accommodate “something” – whether it is user traffic in wireless networks, content to be cached or generated at the edge, or machine learning (ML) models distributed across satellites. Second, each system requires efficient resource allocation mechanisms to manage limited bandwidth, storage, or computational capacity.

To address this challenge, all studies in this dissertation employ optimization-based approaches. Optimization provides a systematic framework for designing algorithms that determine how to allocate system resources – such as time-frequency blocks in wireless networks, storage and computation resource at the edge, or communication capacity in satellite networks – so that service performance is maximized while system constraints are respected. Although the specific types of resources and system models differ across domains, the underlying principle of optimizing resource allocation to support service delivery remains the common theme.

In the domain of wireless mobile networks, the research has focused on both macro-level and micro-level network models, optimizing the allocation of radio resources to deliver user data traffic efficiently. For edge computing systems, the studies have investigated resource allocation for content caching, recommendation, and generation, including scenarios where AI-generated contents impose additional demands on computing resources. In satellite networks, the research addresses the distribution and collection of FL models, optimizing resource usage to ensure effective communication and computation among satellites. Taken together, the dissertation aims at demonstrating that resource allocation, guided by optimization techniques, is a unifying thread connecting wireless, edge, and satellite systems, to deliver enhanced and emerging services.

### 1.3 Organization

This dissertation is structured into two parts. Part I, offers an overview of the fundamental concepts, core technologies, and key optimization methodologies that underpin the research conducted in this thesis. It is designed to provide the necessary preliminaries for the works presented in Part II. In the remainder of Part I, Chapter 2 delves into the key application scenarios and technologies in wireless communications and edge computing. Chapter 3 provides a detailed survey of the optimization methodologies employed throughout the research.

The papers in Part II are along three research lines: Papers I and II for radio resource scheduling, Papers III and IV for content caching, computing, recommendation, and delivery, and Papers V and VI for FL in satellite networks.

## 2. Scenarios and Technologies

### 2.1 Mobile Communications

#### 2.1.1 uRLLC and Ultra-Fast Scheduling

As one of the three main service categories defined for beyond 5G (B5G) systems, uRLLC is characterized by its stringent performance targets of providing extremely high reliability (typically 99.999% or higher), coupled with very low user-plane radio latency (as low as 1 ms), for small data packet transmissions [3]. It supports mission-critical applications such as industrial automation [14], intelligent vehicle systems [15], remote control and teleoperation [16], etc. The key technologies enabling to achieve the stringent targets include flexible numerology [17], mini-slots [18], finite blocklength coding [19], packet duplication [20], grant-free transmission [21], preemptive scheduling [22], network slicing [23], edge computing, and cross-layer optimization.

The core challenge of uRLLC lies in jointly guaranteeing ultra-high reliability and ultra-low latency. These dual constraints fundamentally elevate the role of resource scheduling from a mere efficiency-oriented task to a critical determinant of system feasibility. Unlike eMBB, where scheduling aims to maximize throughput or fairness over time, uRLLC scheduling must ensure that every packet is assigned the necessary radio resources (such as time, frequency, and power) within an extremely short time window, typically on the order of milliseconds or even sub-milliseconds. If the scheduling decision itself is slow – due to complex algorithms, excessive signaling, or processing delays – the total latency budget will be consumed before transmission even starts, rendering the physical-layer enhancements futile. Therefore, the scheduling mechanism must itself be ultra-fast, with execution times comparable to or shorter than the transmission time intervals (TTIs), which can be as low as one mini-slot (e.g., 0.125 ms) [24, 25]. This necessitates low-complexity, ultra-fast scheduling algorithms that can make reliable decisions, under complicated factors including traffic diversity, heterogeneous quality-of-service (QoS) requirements, etc.

As an investigated instance of the above class of resource scheduling problems, **Paper I** studies radio resource allocation optimization, where frequency channels allocated to a user have to share a common transmission rate, with QoS guaranteed. We design low-complexity LP-based approximations to address this NP-hard problem, which allows further parallelization enabling scheduling solution at milli-second level in the future.

### 2.1.2 mMTC and NOMA

It is designed to support the connectivity of an extremely large number of low-power devices that typically transmit small data payloads sporadically and asynchronously [5]. The key challenge lies in the efficient management of massive access with infrequent activity, while minimizing signaling overhead and power consumption to enable long battery life. mMTC is the foundational enabler for large-scale Internet of Things (IoT) [26] deployments, with representative application scenarios including smart cities [27], smart metering [28], environmental monitoring [29], etc. To address the massive connectivity challenge, the key technologies include narrowband transmission (e.g., NB-IoT) [30], repetition and robust modulation and coding [31], NOMA, massive multiple-input multiple-output (MIMO) [32], grant-free access, network slicing, lightweight core network protocols, and cross-layer optimization.

It is a multiple access technique that allows multiple users to transmit simultaneously over the same time-frequency resources through power-domain or code-domain superposition, with receivers employing successive interference cancellation to separate the signals [6]. For mMTC, NOMA is particularly important because it efficiently supports massive connectivity by enabling numerous devices to share the same resource block without strict orthogonality, thereby significantly improving spectral efficiency and access capacity while reducing signaling overhead and latency, which are the key requirements for mMTC. Moreover, NOMA naturally supports user fairness and differentiated QoS through power-domain multiplexing. However, when NOMA is deployed across multiple cells, the non-orthogonal transmissions inherently cause both intra-cell and inter-cell interference. The complex interference environment implies that the allocation of user pairing, power allocation, and decoding order in one cell directly impact the performance in neighboring cells. Therefore, performing cross-cell wireless resource optimization becomes critical to ensure system-wide fairness, reliability, and energy efficiency.

As a specific instance of the above class of NOMA-based resource scheduling problems, **Paper II** investigates a multi-cell resource allocation problem, to improve the performance of cell-edge users, leveraging traffic offloading. The user pairing, user-cell association, resource blocks (RBs), and power control are optimized via an iterative algorithm based on structural analysis and problem decomposition.

### 2.1.3 eMBB and Satellite Networks

It is characterized by the pursuit of gigabit-level peak data rates, massive network capacity, and seamless user experience to support data-intensive applications [4]. It is the primary driver for consumer-centric services such as ultra-high-definition (4K/8K) video streaming, immersive virtual/augmented reality (VR/AR) [33], and mobile cloud computing. The key enabling tech-

nologies for eMBB include millimeter-wave [34], MIMO, flexible numerology, advanced channel coding, and network densification [35].

While terrestrial networks form the backbone for eMBB, satellite networks are increasingly recognized as a vital complementary technology to achieve truly ubiquitous and resilient broadband coverage [36]. Their inherent capability to serve vast geographical areas, including remote, maritime, and aerial domains, addresses the coverage limitations of terrestrial infrastructure, ensuring service continuity for eMBB applications. Furthermore, the integration of satellite networks with edge computing gives rise to the concept of in-orbit edge computing [37], utilizing the computing, communication, and storage resource on satellites. This enables faster content delivery to globally distributed users and alleviates backhaul burdens. The dynamic satellite networks, inherent heterogeneous characteristics, and limited energy supply, however, pose challenges to resource allocation for cross-satellite cooperation. One example is in-orbit FL [38], where satellites collaboratively train a model without sharing local data.

For satellite networks with edge computing, we look into one type of cross-satellite collaboration – FL (see more details in Section 2.2.4). In **Papers V** and **VI**, we model the network dynamics, and address the routing optimization problem for parameter transmission of FL via inter-satellite links, for single model and dual models, respectively.

## 2.2 Edge Computing

Edge computing [2] refers to a distributed computing paradigm where data is processed at network's edge, instead of in centralized cloud data centers. The evolution from cloud computing [8] to edge computing has been driven by the massive devices and data for real-time applications. The advantages of edge computing include low latency, alleviation of backhaul burden, and enhanced privacy and security, making itself a crucial enabling technology for 5G typical scenarios:

- For uRLLC, edge computing helps meet the stringent millisecond-level service requirement, avoiding the unpredictable delay inherent in sending data back and forth to a remote cloud [39].
- For mMTC, edge computing performs local data filtering, aggregation, and preliminary analysis, thus sending only crucial information upstream. This prevents network congestion caused by transmitting all the raw data from IoT devices to the cloud.
- For eMBB, edge computing caches contents such as videos and game data, allowing users to access services with high bandwidth and low latency.

Despite of the potential, edge computing faces some challenges, such as limited resource at network edge, and efficient cooperation with cloud com-

puting. Next, we introduce some application scenarios in edge computing, including edge caching, recommendation, AIGC, FL, and KD, which are relevant to the thesis. Note that except for caching, the other four scenarios are not exclusive to edge computing (as they can also be implemented in cloud environments), this paper specifically addresses the challenge of resource allocation when deploying them within edge computing architectures.

### 2.2.1 Edge Caching

Edge caching [9] is a key technology in edge computing and content delivery networks (CDNs) [40], designed to bring popular contents (e.g. videos, games, and apps) closer to end-users by proactively storing them at the network edge such as BSs. The fundamental principle is to exploit the temporal and spatial locality of content requests – a phenomenon where a small subset of contents accounts for a large fraction of requests over a certain period and within a specific geographical area. By caching these popular contents at the edge, subsequent requests can be served directly from the local cache, significantly reducing data access latency, alleviating the burden on backhaul links, and improving the overall quality of experience (QoE) for users. Two examples of edge caching for practical use are 1) fast and high-quality streaming of Netflix<sup>©</sup> and YouTube<sup>©</sup> with their own CDNs [41, 42], and 2) caching of road safety information, map updates, and popular entertainment at roadside units for vehicular network, in project 5G-CARMEN [43].

The core optimization problem of edge caching at a single BS typically involves deciding which contents to cache, given the limited and often costly storage capacity. This can be formulated as a knapsack problem, aiming to maximize a utility function (e.g., cache hit rate) subject to the storage capacity constraint. A classic formulation is given below:

$$\max_{\mathbf{x}_i \in \{0,1\}} \sum_{i \in \mathcal{I}} q_i x_i \quad (2.1a)$$

$$\sum_{i \in \mathcal{I}} s_i x_i \leq C \quad (2.1b)$$

where  $\mathcal{I}$  is the set of contents,  $C$  is the cache capacity, and  $s_i$ ,  $q_i$ , and  $x_i$  represent the size, popularity, and caching decision variable for content  $i$ . Although beyond the scope of our research, it is worth pointing out that the problem in (2.1) can also be extended to collaboratively caching in multiple BSs, and in that case where to cache is also to be optimized.

For caching optimization, classical algorithms such as least recently used (LRU) [44] and least frequently used (LFU) [45] serve as fundamental baselines; they rely on recency or frequency of user requests for caching update decisions. More advanced approaches leverage content popularity prediction [46], often using statistical or ML to proactively cache contents that

are expected to be in high demand. In dynamic environments, reinforcement learning-based algorithms [47] have gained prominence for their excellent performance. Additionally, freshness-aware policies [48] are employed for time-varying contents. The core trade-off lies in balancing cache hit rate, latency reduction, and computational overhead under constrained resources.

There exist some challenges for caching optimization. First, even for the single-BS problem, the optimization faces the dynamic and time-varying nature of content popularity, leading to its online nature, i.e., placement and replacement of contents along time. Second, the content popularity can be reshaped by recommendation (see Section 2.2.2), which further affects the decision of caching optimization. Third, for new types of contents like AIGC, not only a content but also its prompts can be cached for use (see Section 2.2.3). Note that a content and its prompts differ a lot in their sizes, and accordingly cost different resource of storage, communication, and computing, under several potential content delivery modes. This leads to the necessity of joint optimization of caching, resource allocation, and content delivery mode.

As two research works within the above context, **Paper III** jointly optimizes caching and recommendation to maximize the cache efficiency. We prove the NP-hardness of the problem and develop approximation algorithms with theoretical guarantees. **Paper IV**, in the context of AIGC, optimizes content delivery mode and allocation of computing and communication resources, under given cached contents and prompts, via Lagrangian relaxation and convex optimization.

## 2.2.2 Recommendation

Recommendation systems [10] are information filtering engines widely deployed by online platforms to suggest relevant contents (e.g., videos, games, and products) based on user preferences. The principle is to model and learn the relevance between users and contents from their historical interactions, such as clicks, ratings, or watch time, and other behavior patterns. With recommendation, information overload is alleviated and user engagement and satisfaction are enhanced. Typical real-world applications include the personalized video feeds on video websites YouTube<sup>©</sup> and TikTok<sup>©</sup>, and “*customers who bought this also bought*” recommendation panels on e-commerce platforms Amazon.

The recommendation optimization problem is typically formulated as a prediction task. Let  $\mathcal{U}$  denote the set of users, and  $\mathcal{I}$  denote the set of contents. The target is to predict the relevance score  $r_{ui}$  for user  $u$  and content  $i$  (denoted by function  $f: \mathcal{U} \times \mathcal{I} \rightarrow \mathcal{R}$ ), based on observed interactions, minimizing the loss between predicted scores and observed feedback, i.e.:

$$\min_{\Theta} \sum_{(u,i) \in \mathcal{O}} \mathcal{L}(r_{ui}, f(u, i; \Theta)) \quad (2.2)$$

where  $\mathcal{O}$  is the set of observed interactions,  $\Theta$  represents model parameters,  $\mathcal{L}$  is a loss function. The system ranks contents by their predicted scores to generate a personalized recommendation list.

Conventional recommendation algorithms can be classified to collaborative filtering (CF) methods [49], including matrix factorization [50] and neighborhood-based approaches [51], content-based filtering [52] (based on video-specific features like genre, creator, and thumbnail, etc.), and their hybrid methods. In recent years, deep learning (DL) based sequential methodologies [53, 54] have become dominant in modern video websites such as YouTube<sup>©</sup> and TikTok<sup>©</sup>, due to the capability to capture complex, non-linear relationships. These DL models, often built upon architectures like recurrent neural networks (RNNs) [53] or Transformers [54], treat a user’s watch history as a temporal sequence to predict the next video of interest.

Recommendation systems face intrinsic challenges including the cold-start problem, data sparsity, fairness, and explainability, etc. When integrated with edge caching, additional complexity arises from the influence between the two systems. The recommendation engine actively reshapes content popularity, i.e.,  $q_i$  in Equation (2.1), by determining which contents are exposed to users. For instance, a video recommended to a user will experience a sudden surge in requests, thereby altering the underlying request distribution that the caching policy has been designed for. If the cache is not updated in time to reflect this recommendation-driven popularity change, it will contain outdated content, leading to low cache hit rates and increased latency. Conversely, as contents already stored at the edge can be delivered with very-low latency, caching affects users’ QoE of recommendation. Consequently, the joint optimization of caching and recommendation is necessary and requires a trade-off of recommendation accuracy, cache hit rate, and user-perceived latency, under limited resources at edge.

As one of the concrete cases of such trade-off, in **Paper III**, we jointly optimizes caching and recommendation, with objective of cache efficiency maximize, and constraints on QoE for recommendation. In particular, we consider short-term user interests based recommendation, called incumbent-based recommendation.

### 2.2.3 AIGC

AIGC [11] refers to the paradigm of using advanced AI models, particularly generative AI, to create digital contents, such as text, images, audio, and videos, automatically based on user-provided instructions, a.k.a. prompts. The role of AIGC is a productivity multiplier and creativity catalyst. The integration of AIGC with edge computing heralds a paradigm shift from the traditional model of storing and transmitting pre-created contents to a dynamic, on-demand generation scheme. Within this new paradigm, the network can

also store compact prompts and deploy lightweight generative models at the edge [55] or even on user devices [56], rather than solely caching the final multimedia files. This new paradigm enables low-latency content generation for real-time applications (e.g., instant AR filters [57] and live video synthesis [58]), enhances data privacy, and reduces bandwidth consumption, as only prompts of very small size need to be transmitted.

The AIGC process can be abstracted as a function  $G(M, P)$ , where  $G$  is the generative model,  $M$  represents the model parameters (which can be large, requiring significant computational resources), and  $P$  is the input prompt. The output is the generated content  $C_{gen} = G(M, P)$ . Foundational technologies underpinning AIGC include large language models (LLMs) like chat-GPT [59] and generative diffusion models (GDMs) [60].

AIGC introduces challenges in terms of trade-offs between content quality, resource consumption (for computation, storage, and communication), and latency, in resource-constrained edge environments. First, caching strategies become more complex, as the system can now cache a content or its prompts. Second, on-device AIGC using small models save network bandwidth and backhaul but lead to inferior quality of contents, while on-edge AIGC with medium-size models produce higher-quality contents but consume significant computational resources and require more time. Third, for each content request, the system has multiple options of delivery modes (original content, cached content, on-edge AIGC, or on-device AIGC). All these characteristics make the joint optimization of caching, content delivery mode, and resource allocation for AIGC challenging.

For AIGC, in **Paper IV**, we investigate the joint optimization of content delivery mode and resource allocation, under given cached contents and prompts, to maximize the caching utility.

## 2.2.4 Federated Learning

FL [12] is a distributed ML paradigm designed to train a shared global model collaboratively across multiple decentralized devices without exchanging raw data. Key advantages of FL include privacy protection and reduced communication overhead. FL is inherently synergistic with edge computing: edge nodes (e.g., IoT devices, vehicles, and smartphones) serve as ideal participants of FL. Typical applications include personalized keyboard prediction on smartphones, cross-hospital medical imaging diagnosis, traffic forecasting in vehicular networks, etc.

The standard FL workflow, often coordinated by a central server, typically involves the following iterative steps:

- *Initialization*: The central server initializes a global model  $\mathcal{M}_{\text{global}}^{(0)}$  and defines the training task.

- *Selection and Distribution:* In each communication round  $t$ , the server selects a subset of clients  $\mathcal{S}^{(t)}$  and distributes the current global model  $\mathcal{M}_{\text{global}}^{(t)}$  to them.
- *Local Training:* Each client  $k$  trains the model locally using its private dataset  $\mathcal{D}_k$ , minimizing a local loss function  $F_k(\mathbf{w})$  to produce a local model weight update  $\mathbf{w}_k^{(t+1)}$ .
- *Upload and Aggregation:* The clients send their local updates to the server, which aggregates the updates using the FedAvg algorithm [12]:

$$\mathbf{w}_{\text{global}}^{(t+1)} \leftarrow \sum_{k \in \mathcal{S}^{(t)}} \frac{n_k}{n} \mathbf{w}_k^{(t+1)},$$

where  $n_k = |\mathcal{D}_k|$  and  $n = \sum_{k \in \mathcal{S}^{(t)}} n_k$ .

- *Global Model Update:* The server updates the global model to  $\mathcal{M}_G^{(t+1)}$  with the aggregated parameters, and the process repeats until convergence.

Despite its promise, FL faces several challenges in practical deployment. One is the statistical heterogeneity: Data across clients are typically not independently and identically distributed (non-IID), leading to model bias, convergence difficulties, and performance degradation. To address this issue, client selection (CS) [61] is a promising approach. It determines which devices participate in each round of FL. Compared with random sampling, CS speeds up the convergence rate by prioritizing the devices with high-quality updates [61]. Common metrics for CS include data utility (e.g., dataset size and distribution), model utility (e.g., loss and gradient norm), and system utility (e.g., computational frequency, memory, battery, and channel). The existing CS algorithms can be classified into metric-based methods [62, 63] and mechanism-based methods such as clustering [64], optimization [65], and ML [66].

On-orbit FL refers to the application of FL in satellite networks, enabling edge intelligence in space. Here a client is a satellite that can undertake local training with its own data, and the sever can typically be a ground station or a satellite. Besides data heterogeneity, on-orbit FL faces unique challenges, including dynamic network topology, limited energy supply, heterogeneous hardware resource, and even heterogeneous models. Note that in this case, CS is no longer device-independent, but coupled with routing for data transmission over inter-satellite links (ISLs).

To improve on-orbit FL efficiency, **Papers V** and **VI** model the network dynamics, and jointly optimize CS and routing towards fast convergence.

## 2.2.5 Knowledge Distillation

KD [13] is a model compression and transfer learning technique that enables a compact, efficient student model to learn the rich representations and generalized knowledge embedded within a larger, more complex teacher model.

The distilled student model can achieve an accuracy comparable to or even surpassing that of its teacher, while being significantly smaller, faster, and more energy-efficient. This makes KD exceptionally valuable for deploying advanced AI on resource-constrained devices – a natural fit for edge computing scenarios. Furthermore, KD can be synergistically combined with FL to address model heterogeneity and communication bottlenecks. An example of the applications of KD is distilling LLMs for efficient on-device chatbots [67].

The standard KD process involves the following key steps [13]:

- *Teacher Model Training*: A high-capacity teacher model  $\mathcal{M}_T$  is pre-trained.
- *Knowledge Transfer via Soft Targets*: The student model  $\mathcal{M}_S$  is trained using the teacher’s softened output probabilities (a.k.a. soft-decisions), rather than solely using the ground-truth hard labels. This is achieved by introducing a temperature parameter  $t > 1$  to the softmax function:

$$p_i^t = \frac{\exp(z_i/t)}{\sum_j \exp(z_j/t)},$$

where  $z_i$  is the  $i$ th logit from the teacher (or student). A higher  $t$  produces a softer probability distribution that reveals inter-class similarities.

- *Student Model Training*: The student is trained by minimizing a composite loss function  $\mathcal{L}_{\text{KD}}$  that combines:
  - *Distillation Loss* ( $\mathcal{L}_{\text{Distill}}$ ): It measures the divergence between the student’s softened output  $p_S^t$  and the teacher’s softened output  $p_T^t$ .
  - *Student Loss* ( $\mathcal{L}_{\text{Student}}$ ): The loss between the student’s output and the true hard labels.

The total loss is:  $\mathcal{L}_{\text{KD}} = \alpha \cdot \mathcal{L}_{\text{Distill}} + (1 - \alpha) \cdot \mathcal{L}_{\text{Student}}$ , where  $\alpha$  is a balancing weight.

In **Paper VI**, we utilize KD to assist FL with two heterogeneous models, called dual FL, to speed up the model convergence.



## 3. Optimization Methodologies

### 3.1 Convex Optimization

Convex optimization [68] is a major branch of mathematical optimization, widely applied in practice due to its favorable property that a local optimum is also the global optimum and the availability of efficient and reliable numerical solvers.

A set  $C \subseteq \mathbb{R}^n$  is called a *convex set* if the line segment between any two points in  $C$  lies entirely in  $C$ . Formally, for any  $\mathbf{x}, \mathbf{y} \in C$  and any scalar  $\theta \in [0, 1]$ , we have:

$$\theta \mathbf{x} + (1 - \theta) \mathbf{y} \in C. \quad (3.1)$$

A function  $f : C \rightarrow \mathbb{R}$  defined on a convex set  $C$  is called a *convex function* if for any two points on its graph, the line segment connecting them lies above or on the graph. That is, for any  $\mathbf{x}, \mathbf{y} \in C$  and any  $\theta \in (0, 1)$ , the following inequality holds:

$$f(\theta \mathbf{x} + (1 - \theta) \mathbf{y}) \leq \theta f(\mathbf{x}) + (1 - \theta) f(\mathbf{y}). \quad (3.2)$$

If the inequality is strict for all  $\mathbf{x} \neq \mathbf{y}$  and  $\theta \in (0, 1)$ ,  $f(\cdot)$  is said to be *strictly convex*. A function  $f(\cdot)$  is called *concave* if  $-f(\cdot)$  is convex. A standard convex optimization problem can be written in the following form:

$$\min_{\mathbf{x}} f_0(\mathbf{x}) \quad (3.3a)$$

$$\text{s.t. } f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m, \quad (3.3b)$$

$$\mathbf{a}_j^T \mathbf{x} = b_j, \quad j = 1, \dots, p, \quad (3.3c)$$

where the objective function  $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$  and the inequality constraint functions  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$  are convex, and the equality constraints are affine. The feasible set defined by the constraints is convex, which guarantees the problem's key property: A locally optimal point is globally optimal.

Common types of convex optimization problems include [68, Chapter 4]:

- *Linear Programs (LP)*: Both objective function and constraints are affine.
- *Quadratic Programs (QP)*: Convex quadratic objective function with affine constraints.
- *Semidefinite Programs (SDP)*: Linear objective function over the intersection of the convex cone of positive semidefinite matrices with an affine subspace.

- *Second-Order Cone Programs (SOCP)*: Linear objective function with second-order cone constraints.

Several efficient algorithms [68, Chapters 9-11] exist for solving convex optimization problems, including *gradient descent* and *interior-point methods*, often leveraging the theory of *Lagrangian duality* and the *Karush-Kuhn-Tucker (KKT) conditions* for their derivation and analysis. Here the KKT conditions provide necessary and sufficient optimality criteria for convex problems with differentiable objectives and constraints.

In **Paper II**, Lemma 2 proves the communication capacity function for a single user is strictly convex in cell loads, which is one of the key properties that enable efficient fixed-point iterations for the sub-problem of radio resource consumption minimization. In **Paper IV**, the Lagrangian subproblem is decomposed, with respect to each content and each delivery mode. The sub-subproblems of resource allocation are proved to be convex, and closed-form solutions are derived (in the Propositions and Corollaries 2-5).

## 3.2 Linear Optimization

### 3.2.1 LP

Linear programming (LP) [69] is a foundational mathematical optimization paradigm that concerns a problem with a linear objective function subject to a set of linear equality or inequality constraints.

Mathematically, an LP problem can be expressed in its *standard inequality form*:

$$\text{maximize } \mathbf{c}^\top \mathbf{x} \quad (3.4a)$$

$$\text{subject to } \mathbf{Ax} \leq \mathbf{b}, \quad (3.4b)$$

$$\mathbf{x} \in \mathbb{R}^n \quad (3.4c)$$

where  $\mathbf{x} \in \mathbb{R}^n$  is the vector of decision variables,  $\mathbf{c} \in \mathbb{R}^n$  defines the coefficients of the objective function,  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is the constraint matrix, and  $\mathbf{b} \in \mathbb{R}^m$  is the right-hand-side vector. A *feasible region* refers to the set of points satisfying all constraints, i.e.,  $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\}$ , and  $\mathcal{P}$  is a *convex polyhedron*. For theoretical analysis and algorithm design, there is also the *standard equality form* of LP where all constraints are equalities. Any LP in standard inequality form can be transformed into standard equality form through the introduction of *slack or surplus variables*.

A fundamental theoretical result for LP is the *theorem on existence of optimal solution* [69, Chapter 3.2]:

1. If  $\mathcal{P}$  is bounded, then an optimal solution exists, and there exists at least one optimal solution that is an extreme point (vertex) of  $\mathcal{P}$ . Here an extreme point refers to a point in  $\mathcal{P}$  that cannot be expressed as a convex combination of any two distinct points in  $\mathcal{P}$ .

2. If  $\mathcal{P}$  is unbounded and the objective is bounded above (for maximization) on  $\mathcal{P}$ , then an optimal solution exists at an extreme point.
3. If the objective is unbounded on  $\mathcal{P}$ , the problem is unbounded.

This theorem is pivotal because it reduces the search for an optimum from an infinite continuum to the finite set of extreme points. This enables the design of the classical simplex method [69, Chapters 4] for LP solution. The simplex method is efficient in practice, although it is not polynomial-time in theory. More advanced polynomial-time algorithms to achieve LP optimality include interior-point and ellipsoid methods [69, Chapters 5].

*Duality* is a key concept in LP that establishes a tight relationship between a *primal* problem and its associated *dual* problem [69, Chapter 6]. For a primal LP in standard inequality form  $\max\{\mathbf{c}^\top \mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ , its dual is:  $\min\{\mathbf{b}^\top \mathbf{y} : \mathbf{A}^\top \mathbf{y} \geq \mathbf{c}, \mathbf{y} \geq \mathbf{0}\}$  where  $\mathbf{y}$  represents the dual variables. The following properties hold [69, Chapter 6]:

- *Weak Duality*: For any feasible primal solution  $\mathbf{x}$  and dual solution  $\mathbf{y}$ ,  $\mathbf{c}^\top \mathbf{x} \leq \mathbf{b}^\top \mathbf{y}$ .
- *Strong Duality*: If an LP has a finite optimal solution, then its dual also has a finite optimal solution, and their optimal objective values are equal.
- *Complementary Slackness*: At an optimal primal-dual pair  $(\mathbf{x}^*, \mathbf{y}^*)$ , we have  $y_i^*(b_i - \mathbf{a}_i^\top \mathbf{x}^*) = 0$  and  $x_j^*(\mathbf{a}_j^\top \mathbf{y}^* - c_j) = 0$  for all  $i, j$ . Complementary slackness, primal feasibility, and dual feasibility, together form the KKT conditions, which are the necessary and sufficient conditions for LP optimality.

These properties enables powerful tools for sensitivity analysis, certificate of optimality, and bounding.

In **Paper II**, a theoretical comparison of the optimal objective values of LP relaxations of multiple integer models are provided. In **Paper II**, the routing subproblems with given sink and source nodes are modeled as LPs.

### 3.2.2 ILP and MILP

Integer linear programming (ILP) and mixed-integer linear programming (MILP) [70] share the linear objective and constraints of LP but impose a critical additional requirement: all (for ILP) or some (for MILP) decision variables must take integer values. This discrete nature enables the modeling of discrete decisions in real-world scheduling, routing, resource allocation, and network design problems. However, this modeling power comes at a significant computational cost. The introduction of integer constraints transforms the feasible region from a *convex polyhedron* into a *discrete set of points* (a lattice within the polyhedron), fundamentally altering the problem's geometry and complexity. Although not all, most non-trivial ILP/MILP problems are *NP-hard* [70, Chapter 6]. A strict mathematical proof can be done via *reduction*, from a NP-complete problem to the ILP/MILP problem (see [70, Chapter 6.3] for the

definition and examples of reduction). NP-hardness implies that, assuming  $P \neq NP$ , no algorithm exists that can solve all instances to optimality in time that is polynomial in the problem size.

The *LP relaxation* [70, Chapter 2.2], is an important concept in solving ILP or MILP problems. The LP relaxation is obtained by relaxing the integrality constraints. Solving the LP relaxation, first, provides a bound on the optimal integer solution value (an upper bound for maximization). Second, the LP optimum is often used as benchmark to measure the quality of heuristic solutions.

Interestingly, a combinatorial optimization problem can often be formulated using different, yet equivalent, ILPs/MILPs (e.g., using different variable definitions or constraint aggregations). Although the integer optimum derived by these models coincide, the tightness of their LP relaxations – how closely the LP optimal values approximate the integer optimum – may vary a lot depending on the problem formulations. Comparing their LP bounds requires a *solution mapping* (a.k.a., *projection analysis*). Denote by  $LP_1: \max\{\mathbf{c}_1^\top \mathbf{x} : \mathbf{A}_1 \mathbf{x} \leq \mathbf{b}_1\}$  and  $LP_2: \max\{\mathbf{c}_2^\top \mathbf{z} : \mathbf{A}_2 \mathbf{z} \leq \mathbf{b}_2\}$  the LP relaxations of two ILP/MILP formulations of the same underlying problem, respectively. A solution mapping is a (often linear) transformation  $\phi : \mathcal{P}_1 \rightarrow \mathcal{P}_2$  or  $\psi : \mathcal{P}_2 \rightarrow \mathcal{P}_1$  that maps feasible solutions of one formulation to that of the other, preserving the objective value. Let  $\mathcal{P}'_1$  be the set of objective-preserving mapped solutions  $\phi(\mathbf{x})$  for all  $\mathbf{x} \in \mathcal{P}_1$ . If  $\mathcal{P}'_1 \subseteq \mathcal{P}_2$ , then  $LP_2$  provides a bound that is at least as tight as that of  $LP_1$ . And vice versa.

For exact integer solution, the branch-and-bound (B&B) algorithm, cutting plane algorithm, and heuristics are common options. The B&B algorithm [70, Chapter 7], with the core of smart enumeration in a tree structure, consists of three processes operating iteratively:

- *Bounding*: At each node, the LP relaxation is solved. Its optimal value provides a *local bound* (upper bound for maximization) for that subtree.
- *Branching*: When the solution of the LP relaxation at a node is fractional, the algorithm creates two or more child nodes by adding mutually exclusive constraints that force fractional variables towards integer values (e.g.,  $x_j \leq \lfloor f \rfloor$  and  $x_j \geq \lceil f \rceil$  where  $f$  is the fractional value of variable  $x_j$ ).
- *Pruning*: A node is *pruned* if 1) its LP is infeasible, 2) its bound is no better than the value of the best known integer solution (*incumbent*), or 3) its LP solution is integer feasible.

The efficiency of B&B depends critically on the quality of the bounds obtained from solving the LP relaxation and the speed of finding good integer solutions. Cutting planes [70, Chapter 8], improve B&B by iteratively adding valid inequalities to tighten the LP relaxation at B&B nodes, thereby providing stronger bounds. Commercial solvers deploying B&B and cutting plane methods include CPLEX [71] and Gurobi [72].

Understanding the principle of LP relaxation strength and the solution process of the B&B algorithm is critical for designing efficient ILP/MILP models. Some common empirical results are as follows.

- *Compact vs. non-compact models*: A compact model has a polynomial number of variables and constraints in the input size, while a non-compact model has a super-polynomial (often exponential) number of variables or constraints. Considering model reformulation based on a known model may result in tighter relaxation and more efficient algorithms. Dantzig-Wolfe reformulation is a method that transforms a compact model with decomposable structure into a non-compact one with super-polynomial variables, which can be further solved by column generation [70, Chapter 11.2-11.3]. Benders reformulation, utilizing a so-called staged decision structure, can transform a compact model into a non-compact one with super-polynomial constraints, which can be further solved by row generation [70, Chapter 12]. However, there is no systematic method for reformulating a non-compact model into a compact one. Designing such a model relies mainly on a direct and problem-specific insight.
- *Symmetry*: An ILP is symmetric if its variables can be permuted without changing the structure of the problem. The presence of symmetry often leads to inefficient performance of B&B. Therefore, identifying and breaking symmetry is a crucial step in modeling.
- *Big- $M$  constraints*: It refers to constraints containing a big constant  $M$  for modeling logical constraints. While valid for ensuring correctness and a carefully selected value can mitigate the effect, the introduction of  $M$  inherently weakens the LP relaxation, often leading to loose bound and inefficient performance of solvers. This limitation motivates the use of stronger  $M$ -free formulations (e.g., convex hull representations) when applicable.

In **Paper I**, the resource optimization of channel and rate allocation problem is modeled as multiple ILPs (including compact and non-compact ones), and the relationships between their LP relaxation bounds are analyzed, and further heuristic-based integer solutions are derived. In **Paper III**, the joint optimization problem of caching and recommendation is formulated as an ILP via linearization and can be solved to optimum via solvers, serving as the benchmark on small instances for performance comparison. In **Papers V** and **VI**, the problems of joint CS and routing are modeled as MILPs, and can be solved by solvers or designed heuristics.

### 3.3 Lagrangian Relaxation

Lagrangian relaxation [70, Chapter 10], is a widely used technique to obtain tractable subproblems and useful bounds, often for ILPs and MILPs. The prin-

principle is to relax complicating constraints by incorporating them, weighted by Lagrangian multipliers, into the objective function, yielding a relaxed problem that is easier to solve, while iteratively adjusting the values of multipliers, to get a tight bound of the optimum of the original problem.

Consider a general constrained maximization problem, often referred to as the *primal problem*:

$$[\text{Primal}] \max_{\mathbf{x}} f(\mathbf{x}) \quad (3.5a)$$

$$\text{s.t. } g_i(\mathbf{x}) \leq 0, \quad i \in \{1, \dots, m\} \quad (3.5b)$$

$$\mathbf{x} \in \mathcal{X}, \quad (3.5c)$$

where  $\mathcal{X}$  represents the set, possibly discrete, formed by the remaining constraints such that the structure is easy to handle. The constraints  $g_i(\mathbf{x}) \leq 0$  are considered “complicating” in the sense that they make the problem hard to solve directly.

The *Lagrangian function*, denoted by  $L(\mathbf{x}, \boldsymbol{\lambda})$ , is formed by introducing non-positive Lagrangian multipliers  $\lambda_i \leq 0$  for the relaxed constraints and incorporating them into the original objective function:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{x}). \quad (3.6)$$

The *Lagrangian subproblem* (SP) under given multipliers  $\boldsymbol{\lambda}$ , denoted by  $\theta(\boldsymbol{\lambda})$ , is defined as:

$$[\text{SP}] \max_{\mathbf{x} \in \mathcal{X}} L(\mathbf{x}, \boldsymbol{\lambda}). \quad (3.7)$$

Note that the SP is typically much easier to solve than the primal problem, and *decomposability* is one of the features that worth to be utilized. Specifically, if the feasible region and the objective and constraint functions are additively separable with respect to a partition of the variable vector  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$  (where each  $\mathbf{x}_j$  is a sub-vector of variables), i.e.,  $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_N$ ,  $f(\mathbf{x}) = \sum_j f_j(\mathbf{x}_j)$ ,  $g_i(\mathbf{x}) = \sum_j g_{ij}(\mathbf{x}_j)$  for all  $i$  and  $j$ , the SP decomposes into  $N$  independent subproblems.

To find the values of multipliers that result in the best possible upper bound, the *Lagrangian dual problem* is defined as:

$$[\text{Lagrangian-Dual}] \min_{\boldsymbol{\lambda} \leq \mathbf{0}} \theta(\boldsymbol{\lambda}). \quad (3.8)$$

Denote by  $\text{OPT}_{\text{primal}}$ ,  $\text{OPT}_{\text{SP}}(\boldsymbol{\lambda})$  and  $\text{OPT}_{\text{Dual}}$  the optimal objective values of (3.5), (3.7), and (3.8), respectively, we have the following properties [70, Chapter 10]:

- *Weak Duality*: Since  $\lambda_i \leq 0$  and  $g_i(\mathbf{x}) \leq 0$  for any feasible  $\mathbf{x}$  in [Primal], we have  $L(\mathbf{x}, \boldsymbol{\lambda}) \geq f(\mathbf{x})$  for all primal feasible  $\mathbf{x}$  and any  $\boldsymbol{\lambda} \leq \mathbf{0}$ . Consequently,  $\theta(\boldsymbol{\lambda})$  provides an *upper bound* on the optimal objective value of

[Primal] for any  $\lambda$ , including the best one found by solving [Lagrangian-Dual]. Hence, it always holds that

$$\text{OPT}_{\text{Lagrangian-Dual}} \geq \text{OPT}_{\text{Primal}} \quad (3.9)$$

- *Strong Duality*: It means that  $\text{OPT}_{\text{Dual}} = \text{OPT}_{\text{Primal}}$ , which only holds under certain cases, in particular convex optimization problems that satisfy the Slater condition [68] and ILPs/MILPs with totally unimodular matrices.
- *More Stringent Bound than LP*: If [Primal] is a MILP or ILP, another common relaxation is LP relaxation. Denote by the  $\text{OPT}_{\text{LP}}$  the optimal bound achieved by solving the LP, it always holds that:

$$\text{OPT}_{\text{LP}} \geq \text{OPT}_{\text{Lagrangian-Dual}} \quad (3.10)$$

Equality holds, if the LP relaxation possesses the integer property, meaning that at least one of its optimal extreme point solutions is integral.

For the solution of [Lagrangian-Dual], the *subgradient method* is one of the standard approaches. The dual function  $\theta(\lambda)$  can be shown to be convex but generally non-differentiable. Therefore, gradient-based methods cannot be applied directly. A vector  $\mathbf{s} \in \mathbb{R}^m$  is a *subgradient* of  $\theta$  at  $\lambda$  if for all  $\lambda' \leq 0$ ,  $\theta(\lambda') \geq \theta(\lambda) + \mathbf{s}^\top(\lambda' - \lambda)$  holds. For [Lagrangian-Dual], a subgradient at  $\lambda$  is given by the constraint violation vector  $\mathbf{s}(\lambda) = (g_1(\mathbf{x}^*), g_2(\mathbf{x}^*), \dots, g_m(\mathbf{x}^*))^\top$  where  $\mathbf{x}^* \in \arg \max_{\mathbf{x} \in \mathcal{X}} L(\mathbf{x}, \lambda)$ . The subgradient method iteratively updates the multipliers as follows:

$$\lambda^{k+1} = \max \left( \mathbf{0}, \lambda^k + \alpha_k \mathbf{s}(\lambda^k) \right), \quad (3.11)$$

where  $\alpha_k > 0$  is the step size in iteration  $k$ . Common step size rules include [70, Chapter 10.3]:

- *Diminishing step size*:  $\alpha_k \rightarrow 0$ ,  $\sum_{k=1}^{\infty} \alpha_k = \infty$ , which guarantees convergence to the optimal dual value.
- *Polyak's step size*:  $\alpha_k = \gamma_k \frac{\theta^* - \theta(\lambda^{(k)})}{\|\mathbf{s}(\lambda^{(k)})\|^2}$ , where  $\theta^*$  is the optimal dual value (or an estimate of it) and  $0 < \gamma_k < 2$ .

In **Paper I**, Lagrangian relaxation is used as a theoretical proof technique to establish the equivalence between the LP relaxations of two ILP formulations. In **Paper IV**, Lagrangian relaxation is applied to three resource limit constraints. The resulting SP is decomposed with respect to each content and further each content delivery mode. The subgradient method is adopted to solve the Lagrangian dual problem, while SP is solved by the derived closed-form solutions and bi-section search.

## 3.4 Network Flows

Network flow problems [73] constitute an important class of combinatorial optimization problems defined on graphs. Examples include *maximum flow*, *minimum cost flow*, *multi-commodity flows*, and *matching* problems.

### 3.4.1 Maximum Flow

Given a directed graph  $G = (V, A)$  with node set  $V$ , arc set  $A$ , arc capacities  $u_{ij} \geq 0$  for any arc  $(i, j) \in A$ , a source node  $s \in V$ , and a sink node  $t \in V$ , the maximum flow problem [73, Chapters 6-8], aims to send the maximum possible amount of flow from  $s$  to  $t$  while respecting arc capacities and flow balance at all other nodes (i.e., inflow equals outflow). The problem admits a natural LP:

$$\max \quad v \tag{3.12a}$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = \begin{cases} v, & \text{if } i = s \\ -v, & \text{if } i = t \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V \tag{3.12b}$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in A \tag{3.12c}$$

where  $x_{ij}$  is the variable representing the flow on arc  $(i, j)$  and  $v$  is the total flow value. The first set of constraints enforces flow balance, and the second set is the arc capacity constraints. Note that this LP has a very special structure: Its constraint matrix is totally unimodular, and when capacities are integers, the LP has integer property (i.e. at least one of its optimal extreme point solutions is integral).

Besides, some algorithms [73, Chapters 6-8], exploit the combinatorial structure of the problem rather than solving the LP generically.

- *Ford-Fulkerson method*: It iteratively augments flow along  $s$ - $t$  paths in the residual graph. Its complexity depends on the augmentation strategy, with worst-case time complexity of  $O(|A| \cdot v_{\max})$ , where  $v_{\max}$  is the maximum flow value, making it pseudo-polynomial.
- *Edmonds-Karp Algorithm*: An implementation of Ford-Fulkerson that uses breadth-first search to find the shortest augmenting path at each iteration. This simple refinement guarantees termination in  $O(|V||A|^2)$  time, establishing polynomial complexity.
- *Push-Relabel Algorithms (Preflow-Push)*: These algorithms use a local strategy, maintaining a *preflow* and node *height labels*. Operations involve pushing excess flow from higher to lower height neighbors and relabeling nodes when necessary. Variants like the highest-label method achieve efficient theoretical bounds (e.g.,  $O(|V|^2 \sqrt{|A|})$ ).

The *max-flow min-cut theorem* [73, Chapters 6.3], is an important theorem based on LP duality, stating that the maximum value of an  $s$ - $t$  flow is equal

to the minimum capacity of an  $s$ - $t$  cut. Here an  $s$ - $t$  cut is a partition of  $V$  into sets  $S$  (containing  $s$ ) and  $T$  (containing  $t$ ); its capacity is the sum of capacities of arcs from  $S$  to  $T$ . The theorem not only provides a powerful structural optimality condition but also inspires solution algorithms.

In **Paper I**, the max-flow min-cut theorem is utilized as an analytical tool to establish an equivalence relationship between the LP relaxations of two ILP formulations.

### 3.4.2 Minimum Cost Flow

Given a directed graph  $G = (V, A)$  with arc cost  $c_{ij}$  and capacity  $u_{ij}$  for any arc  $(i, j)$ , and  $b_i$  representing the supply or demand of node  $i$  (depending on whether  $b_i > 0$  or  $b_i < 0$ ), the minimum cost flow problem [73, Chapters 9-11], seeks a feasible flow that satisfies all supplies/demands and arc capacities at minimum total cost  $\sum_{(i,j) \in A} c_{ij}x_{ij}$ . This problem can be formulated as an LP:

$$\min \sum_{(i,j) \in A} c_{ij}x_{ij} \quad (3.13a)$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = b_i \quad \forall i \in V \quad (3.13b)$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in A \quad (3.13c)$$

where  $d$ . Note that the constraint matrix is also totally unimodular, hence if  $d$  and capacities  $u_{ij}$  are integers, the integer property of this LP holds.

The optimal conditions for minimum cost flows are the following [73, Chapter 9.3]: Let  $\pi_i$  be a potential (dual variable) for node  $i$ . The reduced cost of arc  $(i, j)$  is  $c_{ij}^\pi = c_{ij} - \pi_i + \pi_j$ . A feasible flow  $x^*$  is optimal if and only if there exist node potentials  $\pi$  such that 1) if  $c_{ij}^\pi > 0$ , then  $x_{ij}^* = 0$ , 2) and if  $0 < x_{ij}^* < u_{ij}$ , then  $c_{ij}^\pi = 0$ , and 3) if  $c_{ij}^\pi < 0$ , then  $x_{ij}^* = u_{ij}$ . Equivalently,  $c_{ij}^\pi \geq 0$  for all arcs in the residual network of  $x^*$ .

Algorithms for the minimum cost flow problem exploit this optimality conditions [73, Chapters 9-11]:

- *Negative Cycle Canceling*: A negative cycle refers to a directed cycle in the residual network whose sum of arc costs is negative. Based on the optimality condition, the algorithm finds such cycles and sends flow around them to reduce cost. The algorithm runs in polynomial time, if specific cycle-selection strategy is deployed, such as always canceling the cycle with the most negative average cost.
- *Successive Shortest Path Algorithm*: It incrementally satisfies node supply/demand by iteratively sending flow along shortest paths in the reduced-cost residual network, using Dijkstra's algorithm. It runs in polynomial time,  $O(|V|^2 \log |V| + |V||E|)$  per augmentation.

- *Network Simplex Method*: A empirically fast version of the simplex method that operates on a spanning tree structure. While having exponential worst-case complexity, it is exceptionally efficient in practice.

In **Paper I**, the channel allocation subproblem under fixed user rate is proved to be equivalent to a minimum-cost flow problem (with negative costs representing utility) on an acyclic graph. In **Paper V**, a routing subproblem with given source and sink nodes in a time-varying graph (TVG) is modeled as a minimum-cost flow problem and solved via LP solver.

### 3.4.3 Multi-Commodity Flows

In a multi-commodity flow problem,  $K$  distinct commodities share a network. For each commodity  $k$ , denote by  $b_i^k$  the supply or demand of node  $i$  (depending on whether  $b_i^k > 0$  or  $b_i^k < 0$ ). The goal is to route all commodities simultaneously, respecting shared arc capacities, to maximize the total flow or minimize the total cost. For cost minimization, this problem can be formulated as the LP below:

$$\min \sum_{k=1}^K \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k \quad (3.14a)$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in A} x_{ij}^k - \sum_{j:(j,i) \in A} x_{ji}^k = b_i^k \quad \forall i \in V, \forall k \quad (3.14b)$$

$$\sum_{k=1}^K x_{ij}^k \leq u_{ij} \quad \forall (i,j) \in A \quad (3.14c)$$

$$x_{ij}^k \geq 0 \quad \forall (i,j) \in A, \forall k \quad (3.14d)$$

While this problem can be solved by general-purpose LP algorithms, the predominant classical approaches for large-scale instances are usually specialized techniques such as Lagrangian relaxation.

In **Paper VI**, the parameters of teacher and student FL models correspond to the two commodities. The routing sub-problem for them with given source and sink nodes in a TVG is modeled as a minimum-cost multi-commodity flow problem.

### 3.4.4 Matching

A *matching* in an undirected graph  $G = (V, E)$  is a subset of edges  $M \subseteq E$  such that the edges in  $M$  do not have any common vertex. Matching problems [73, Chapter 12], model pairwise assignment under exclusivity constraints. One of its well studied branches is matching on bipartite graphs [74]. The key variants of matching includes:

- *Maximum Cardinality Matching*: The task is to find a matching with the greatest number of edges.

- *Perfect Matching*: The task is to find a matching that covers *all* vertices of the graph ( $|M| = |V|/2$ ). A perfect matching is also a maximum cardinality matching.
- *Maximum-Weight Matching*: Given edge weights  $w_e$  for edge  $e \in E$ , the problem is to find a matching that maximizes the total weight  $\sum_{e \in M} w_e$ .
- *Complete Matching*: Consider a bipartite graph  $G = (X, Y, E)$  where  $X$  and  $Y$  are the two disjoint vertex sets. A matching is called a complete matching with respect to  $X$  if each vertex in  $X$  is covered by an edge in  $M$ .

For maximum cardinality matching in bipartite graphs, the Hopcroft-Karp algorithm [75] achieves a time complexity of  $O(|E|\sqrt{|V|})$ . For maximum-weight bipartite matching, the classic Hungarian algorithm [76] provides a polynomial-time solution. For general graphs, Edmonds' Blossom algorithm [77] solves the maximum cardinality and maximum-weight matching problems in polynomial time. In addition, a maximum cardinality bipartite matching can be solved by converting it into a maximum flow problem: We add a source connected to all vertices in  $X$  and a sink from all vertices in  $Y$ , assign unit capacity to all edges, and compute the maximum flow. Similarly, the maximum-weight bipartite matching problem is equivalent to a minimum-cost flow problem. This equivalence enables the design of network-flow-based algorithms which is polynomial-time, for matching in bipartite graphs.

*Hall's marriage theorem* provides a necessary and sufficient condition for the existence of a complete matching from  $X$  to  $Y$ : For every subset  $S \subseteq X$ , the number of its neighbors in  $Y$ , denoted by  $N(S)$ , must satisfy  $|N(S)| \geq |S|$ .

In **Paper I**, the resource optimization problem under given rate allocation is modeled as a bipartite complete matching problem. Hall's theorem serves as an analytical tool to prove the correctness of a non-compact model. In **Paper II**, a user pairing subproblem is modeled as the maximum-weight matching problem on a generic graph, and solved via Edmonds' Blossom algorithm.

### 3.5 Approximation Algorithms

In computational complexity theory, many optimization problems of practical importance are NP-hard, meaning that unless  $P = NP$  [78], there exists no algorithm that can find an optimal solution for all instances of the problem in time polynomial in the input size. In some cases, *approximation algorithms* [79] provide a viable alternative for computationally intractable problems. Such an algorithm runs in polynomial time and guarantees that the objective value of its solution is within a provable approximation ratio range of the optimum.

Formally, for a minimization problem, an algorithm  $\mathcal{A}$  is said to be a  $\rho$ -approximation algorithm (for  $\rho \geq 1$ ) if for any instance  $I$  of the problem, the algorithm runs in polynomial time and produces a solution whose objective

function value  $f_{\mathcal{A}}(I)$  satisfies:

$$\frac{f_{\mathcal{A}}(I)}{OPT(I)} \leq \rho, \quad (3.15)$$

where  $OPT(I)$  denotes the optimal objective value for instance  $I$ .

For a maximization problem, an algorithm  $\mathcal{A}$  is a  $\rho$ -approximation (for  $0 < \rho \leq 1$ ) if:

$$\frac{f_{\mathcal{A}}(I)}{OPT(I)} \geq \rho. \quad (3.16)$$

The value  $\rho$  is called the *approximation ratio* or *approximation factor*. A *polynomial-time approximation scheme (PTAS)* [79, Chapter 8], is a family of algorithms that, for any fixed  $\epsilon > 0$ , provides a  $(1 + \epsilon)$ -approximation (for minimization) or  $(1 - \epsilon)$ -approximation (for maximization) in time polynomial in the input size (though the running time may depend exponentially on  $1/\epsilon$ ).

*Submodularity* [79, Chapter 23], is one of the key characteristics that enable the design of approximation algorithms. It refers to the property of set functions with the notion of diminishing returns. A set function  $f : 2^V \rightarrow \mathbb{R}$  is *submodular* if for all subsets  $A \subseteq B \subset S$  and any element  $e \in S \setminus B$ , it holds that:

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B).$$

If additionally  $f(\emptyset) = 0$  and  $f(A) \leq f(B)$ , then  $f$  is *monotone non-decreasing*. These properties enable the design of algorithms with approximation guarantees, under different constraints:

- *Cardinality Constraint* (i.e., set  $|S|$  has at most  $K$  elements): The greedy algorithm [80] starts with  $S = \emptyset$  and iteratively adds the element  $e$  maximizing the marginal gain  $\Delta_f(e|S) = f(S \cup \{e\}) - f(S)$ , until  $|S| = K$ , yielding a  $(1 - 1/e)$  approximation ratio.
- *Knapsack Constraint* (i.e., the sum weight of items in set  $|S|$  is at most  $C$ ): A modified greedy algorithm combined with partial enumeration [81] yields a  $(1 - 1/e)$ -approximation guarantee.

In **Paper I**, the optimal cache-hit-ratio via optimizing recommendation decision under given caching is proved to be monotone and submodular with respect to the set of cached contents. Hence the optimal recommendation algorithm is embedded into the modified greedy algorithms in [80] and [81] to achieve a  $(1 - 1/e)$ -approximation guarantee for the overall joint optimization of caching and recommendation.

## 3.6 Greedy Heuristics

Greedy heuristics are a class of iterative algorithms for combinatorial optimization. The principle is to construct a solution step-by-step, making locally

optimal choices at each stage without reconsidering previous decisions. At every iteration, the algorithm selects the element that appears most beneficial according to a myopic criterion – such as the highest immediate gain or the best value-to-cost ratio – and adds it to the solution set. This process repeats until a complete feasible solution is formed. Known for their simplicity, efficiency, and ease of implementation, greedy heuristics are widely used in practice. Although greedy strategies sometimes reach approximation guarantee, e.g., the submodularity-based one in Section 3.5, in most cases, there is no guarantee on solution quality.

In **Paper I**, simple rounding and iterative rounding strategies represent greedy heuristics for constructing integer feasible solutions from the fractional LP solutions. In **Paper II**, the traffic offloading decisions are made via greedy heuristic based on channel gain information. In **Paper III**, the caching sub-problem within an algorithm is solved via greedy heuristic. In **Paper IV**, greedy heuristic is applied in a repairing algorithm in searching for feasible solutions. In **Papers V and VI**, greedy heuristics are used for low-complexity solution of CS along with feasibility check of routing.



## 4. Scope and Contributions

### 4.1 Scope

This thesis, composed of six papers, explores a set of optimization problems for resource allocation in mobile communications and edge computing. These problems share the common characteristics that 1) the systems need to process and accommodate “something” – whether it is user traffic in wireless networks, content to be cached or generated at the edge, or ML models distributed across satellites, and 2) the systems require efficient resource allocation mechanisms to manage limited bandwidth, storage, or computational capacity. For an overview, please see Table 4.1.

**Table 4.1.** *Summary of the papers involved in this thesis.*

Paper	Mobile Communications			Edge Computing				
	uRLLC and Ultra-Fast Scheduling	mMTC and NOMA	eMBB and Satellite Networks	Edge Caching	Recommendation	AIGC	FL	KD
I	✓							
II		✓						
III				✓	✓			
IV				✓		✓		
V			✓				✓	
VI			✓				✓	✓

A summary of the optimization methodologies utilized in this thesis are provided in Table 4.2.

**Table 4.2.** *Summary of the optimization methodologies utilized in this thesis.*

Paper	Convex Optimization	Lagrangian Relaxation	LP and ILP			Network Flow		Approximation Algorithms	Greedy-based Heuristic
			LP	ILP	MILP	Flow Problems	Matching		
I		✓	✓	✓		✓	✓	✓	
II	✓						✓	✓	
III				✓			✓	✓	
IV	✓	✓						✓	
V			✓		✓	✓		✓	
VI					✓	✓		✓	

### 4.2 Contributions

The papers in Part II are along three research lines: Papers I and II on radio resource scheduling, Papers III and IV on content caching, computing, recommendation, and delivery, and Papers V and VI on FL in satellite networks.

### **Paper I: On Optimization Formulations for Radio Resource Allocation Subject to Common Transmission Rate**

This paper studies the radio resource allocation problem under common transmission rate constraints. We first provide a complexity analysis. Next, several ILP formulations for the problem, including compact as well as non-compact models, are derived. We then provide a rigorous comparative study of their LP relaxations, to reveal the relationship between the formulations in terms of bounding. Numerical results in LP bounding and LP-assisted problem solving are presented. This study sets a ground for the next step of developing model-based and tailored millisecond-level scheduling algorithm.

Personal contributions: Investigation of related work, conceptualization of model formulations, formal analysis (partial), algorithm design, simulation and visualization, writing of the original draft (partial), and revision according to the review comments.

### **Paper II: Delivering More to Cell Edge via Joint Multi-Cell NOMA and Traffic Offloading**

This paper leverages NOMA and traffic offloading to benefit the throughput of cell-edge users. We formulate the throughput maximization problem, with RB allocation, power allocation, user pairing, and offloading decision to be optimized, accounting for the impact of inter-cell interference. We then develop an iterative algorithm based on problem decomposition and structural analysis to address this problem. Numerical results show that the proposed scheme considerably improves the cell-edge throughput, compared with the conventional schemes.

Personal contributions: Investigation of related work, formal analysis and algorithm design, simulation and visualization, writing of the original draft, and revision according to the review comments.

### **Paper III: Caching with Personalized and Incumbent-Aware Recommendation: Modeling and Optimization**

This paper addresses the joint optimization of caching and incumbent-aware recommendation. The incumbent content refers to the content that a user is currently browsing, resulted by the user's short-term interest. We investigate the influence between caching and recommendation through content popularity. For the proposed cache efficiency maximization problem subject to user satisfaction requirements, we prove its NP-hardness, and derive an ILP formulation and three polynomial-time algorithms. Among them, the first two are based on sub-modularity, with approximation guarantee under mild conditions, while the last one is an alternation-based algorithm with fast convergence. Numerical results show the close-to-optimal performance of the proposed algorithms.

Personal contributions: Proposal of the research topic, conceptualization of the problem under study, investigation of related work, formal formulation and analysis, algorithm design, simulation and visualization, writing of the original draft, and revision according to the review comments.

#### **Paper IV: What to Deliver? When Resource Allocation Meets AIGC on Network Edge and User Device**

This paper optimizes communication resource allocation, computing resource allocation, and delivery mode selection for AIGC. AIGC can be cached at edge, generated at edge, or generated on device. We account for the trade-offs between content quality and resource consumption, and develop an efficient solution based on convex optimization and Lagrangian relaxation, with near-optimal performance in simulation.

Personal contributions: Proposal of the research topic, conceptualization of problem and formulations, theoretical proof (partial), algorithm design (partial), simulation and visualization, writing of the original draft (partial), revision according to the review comments, and presentation.

#### **Paper V: Orchestrating in the Sky: Joint Routing and Client Selection for Federated Learning in LEO Networks**

This paper investigates CS and inter-satellite routing for on-orbit FL towards fast convergence. We derive and minimize an upper bound of the global empirical loss as the objective function, to speed up the convergence. We model the constraints of inter-satellite routing via TVGs and network flow theory. We propose both exact and approximate solutions for the joint optimization problem of CS and routing. In addition, we formalize and prove the convergence property of our approach. Last, by simulation we demonstrate the efficiency and superiority of the proposed scheme for realistic satellite networking scenarios.

Personal contributions: Proposal of the research topic, conceptualization of the problem under study, investigation of related work, formal formulation and analysis, algorithm design, simulation and visualization, writing of the original draft, revision according to the review comments, and presentation.

#### **Paper VI: Distilling Intelligence in Space: Optimized Dual Federated Learning amid Orbital Dynamics**

As an extended work of Paper V, this paper further considers heterogeneous models and KD, for smart energy utilization and fast convergence of on-orbit FL. We develop a KD-assisted dual-FL framework with dedicated convergence analysis, derive upper bounds on both models' global empirical loss, and jointly optimize CS and inter-satellite routing. Simulations on real-world satellite constellations showcase a significant performance edge of dual-model FL with KD in pushing the boundaries of on-orbit intelligence.

Personal contributions: Proposal of the research topic, conceptualization of the problem under study, investigation of related work, formal formulation and analysis, algorithm design, simulation and visualization, and writing of the original draft.



# References

- [1] Andreas F Molisch. *Wireless Communications*, volume 34. John Wiley & Sons, 2012.
- [2] Keyan Cao, Yefan Liu, Gongjie Meng, and Qimeng Sun. An overview on edge computing research. *IEEE Access*, 8:85714–85728, 2020.
- [3] Zexian Li, Mikko A Uusitalo, Hamidreza Shariatmadari, and Bikramjit Singh. 5G uRLLC: Design challenges and system concepts. In *2018 15th international symposium on wireless communication systems (ISWCS)*, pages 1–6. IEEE, 2018.
- [4] Benish Sharfeen Khan, Sobia Jangsher, Ashfaq Ahmed, and Arafat Al-Dweik. uRLLC and eMBB in 5G industrial IoT: A survey. *IEEE Open Journal of the Communications Society*, 3:1134–1163, 2022.
- [5] Carsten Bockelmann, Nuno K Pratas, Gerhard Wunder, Stephan Saur, Monica Navarro, David Gregoratti, Guillaume Vivier, Elisabeth De Carvalho, Yalei Ji, Čedomir Stefanović, et al. Towards massive connectivity support for scalable mMTC communications in 5G networks. *IEEE Access*, 6:28969–28992, 2018.
- [6] Aamina Akbar, Sobia Jangsher, and Farrukh A Bhatti. NOMA and 5G emerging technologies: A survey on issues and solution techniques. *Computer Networks*, 190:107950, 2021.
- [7] Bassel Al Homssi, Akram Al-Hourani, Ke Wang, Phillip Conder, Sithamparanathan Kandeepan, Jinho Choi, Ben Allen, and Ben Moores. Next generation mega satellite networks for access equality: Opportunities, challenges, and performance. *IEEE Communications Magazine*, 60(4):18–24, 2022.
- [8] Ling Qian, Zhiguo Luo, Yujian Du, and Leitao Guo. Cloud computing: An overview. In *IEEE international conference on cloud computing*, pages 626–631. Springer, 2009.
- [9] Rafat Aghazadeh, Ali Shahidinejad, and Mostafa Ghobaei-Arani. Proactive content caching in edge computing environment: A review. *Software: Practice and Experience*, 53(3):811–855, 2023.
- [10] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web: methods and strategies of web personalization*, pages 325–341. Springer, 2007.
- [11] F. Wang, L. Jiao, K. Zhu, L Pu, and L Zhang. Toward sustainable diffusion-based AIGC: Design and online orchestration in distributed edge networks. *Computer Networks*, 259, 2025. Article 111077.
- [12] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.
- [13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint: 1503.02531*, 2015.

- [14] Fatemeh Hamidi-Sepehr, Masoud Sajadieh, Sergey Panteleev, Toufique Islam, Ingolf Karls, Debdeep Chatterjee, and Junaid Ansari. 5G uRLLC: Evolution of high-performance wireless networking for industrial automation. *IEEE Communications Standards Magazine*, 5(2):132–140, 2021.
- [15] Qiong Wu, Wenhua Wang, Pingyi Fan, Qiang Fan, Jiangzhou Wang, and Khaled B Letaief. URLLC-awared resource allocation for heterogeneous vehicular edge computing. *IEEE Transactions on Vehicular Technology*, 73(8):11789–11805, 2024.
- [16] Bo Chang, Lei Zhang, Liying Li, Guodong Zhao, and Zhi Chen. Optimizing resource allocation in uRLLC for real-time wireless control systems. *IEEE Transactions on Vehicular Technology*, 68(9):8916–8927, 2019.
- [17] Josue Flores de Valgas, Jose F Monserrat, and Hüseyin Arslan. Flexible numerology in 5G NR: Interference quantification and proper selection depending on the scenario. *Mobile Information Systems*, 2021(1):6651326, 2021.
- [18] Wonjun Kim and Byonghyo Shim. Ultra-mini slot transmission for 5G+ and 6G uRLLC network. In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pages 1–5. IEEE, 2020.
- [19] Chentao Yue, Vera Miloslavskaya, Mahyar Shirvanimoghaddam, Branka Vucetic, and Yonghui Li. Efficient decoders for short block length codes in 6G uRLLC. *IEEE communications Magazine*, 61(4):84–90, 2023.
- [20] Jaya Rao and Sophie Vrzić. Packet duplication for uRLLC in 5G: Architectural enhancements and performance analysis. *IEEE Network*, 32(2):32–40, 2018.
- [21] Yan Liu, Yansha Deng, Maged Elkashlan, Arumugam Nallanathan, and George K Karagiannidis. Analyzing grant-free access for uRLLC service. *IEEE Journal on Selected Areas in Communications*, 39(3):741–755, 2020.
- [22] Ali A Esswie and Klaus I Pedersen. Multi-user preemptive scheduling for critical low latency communications in 5G networks. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00136–00141. IEEE, 2018.
- [23] Petar Popovski, Kasper Fløe Trillingsgaard, Osvaldo Simeone, and Giuseppe Durisi. 5G wireless network slicing for eMBB, uRLLC, and mMTC: A communication-theoretic view. *IEEE Access*, 6:55765–55779, 2018.
- [24] Shashi Raj Pandey, Madyan Alsenwi, Yan Kyaw Tun, and Choong Seon Hong. A downlink resource scheduling strategy for uRLLC traffic. In *2019 IEEE international conference on big data and smart computing (BigComp)*, pages 1–6. IEEE, 2019.
- [25] Md Emdadul Haque, Faisal Tariq, Muhammad RA Khandaker, Kai-Kit Wong, and Yangyang Zhang. A survey of scheduling in 5G uRLLC and outlook for emerging 6G systems. *IEEE Access*, 11:34372–34396, 2023.
- [26] Wan Haslina Hassan et al. Current research on internet of things (IoT) security: A survey. *Computer networks*, 148:283–294, 2019.
- [27] Tai-hoon Kim, Carlos Ramos, and Sabah Mohammed. Smart city and IoT. *Future Generation Computer Systems*, 76:159–162, 2017.
- [28] Jaime Lloret, Jesus Tomas, Alejandro Canovas, and Lorena Parra. An integrated IoT architecture for smart metering. *IEEE Communications Magazine*, 54(12):50–57, 2016.

- [29] Silvia Liberata Ullo and Ganesh Ram Sinha. Advances in smart environment monitoring systems using IoT and sensors. *Sensors*, 20(11):3113, 2020.
- [30] Mona Bakri Hassan, Elmustafa Sayed Ali, Rania A Mokhtar, Rashid A Saeed, and Bharat S Chaudhari. NB-IoT: Concepts, applications, and deployment challenges. In *LPWAN Technologies for IoT and M2M Applications*, pages 119–144. Elsevier, 2020.
- [31] Harini Grama Srinath, Mrinal Rana, and Naveen Mysore Balasubramanya. Grant-free access for mMTC: A performance analysis based on number of preambles, repetitions, and retransmissions. *IEEE Internet of Things Journal*, 9(16):15169–15183, 2022.
- [32] Lu Lu, Geoffrey Ye Li, A Lee Swindlehurst, Alexei Ashikhmin, and Rui Zhang. An overview of massive MIMO: Benefits and challenges. *IEEE journal of selected topics in signal processing*, 8(5):742–758, 2014.
- [33] Syed Mohammad Haseeb Ul Hassan, Attracta Brennan, Gabriel-Miro Muntean, and Jennifer McManis. NSM 2: network slice management and monitoring using machine learning for AR/VR applications. In *2024 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–7. IEEE, 2024.
- [34] Xinying Li, Jianjun Yu, and Gee-Kung Chang. Photonics-aided millimeter-wave technologies for extreme mobile broadband communications in 5G. *Journal of Lightwave Technology*, 38(2):366–378, 2019.
- [35] Behnam Ojaghi, Ferran Adelantado, Angelos Antonopoulos, and Christos Verikoukis. Impact of network densification on joint slicing and functional splitting in 5G. *IEEE Communications Magazine*, 60(7):30–35, 2022.
- [36] Tomaso De Cola and Igor Bisio. QoS optimisation of eMBB services in converged 5G-satellite networks. *IEEE Transactions on Vehicular Technology*, 69(10):12098–12110, 2020.
- [37] Yuxuan Wang, Jun Yang, Xiye Guo, and Zhi Qu. Satellite edge computing for the internet of things in aerospace. *Sensors*, 19(20):4375, 2019.
- [38] Bho Matthiesen, Nasrin Razmi, Israel Leyva-Mayorga, Armin Dekorsy, and Petar Popovski. Federated learning in satellite constellations. *IEEE Network*, 38(2):232–239, 2023.
- [39] Ntshuxeko Makondo, Hlabishi I. Kobo, Topside E. Mathonsi, Deon Du Plessis, Thoriso M. Makhosa, and Lusani Mamushiane. An efficient architecture for latency optimisation in 5G using edge computing for uRLLC use cases. In *2024 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pages 1–7, 2024.
- [40] Jihoon Sung, Minseok Kim, Kyongchun Lim, and June-Koo Kevin Rhee. Efficient cache placement strategy in two-tier wireless content delivery network. *IEEE Transactions on Multimedia*, 18(6):1163–1174, 2016.
- [41] Trinh Viet Doan, Vaibhav Bajpai, and Sam Crawford. A longitudinal view of Netflix: Content delivery over IPv6 and content cache deployments. In *Proceedings of IEEE INFOCOM*, pages 1073–1082, 2020.
- [42] Trinh Viet Doan, Ljubica Pajevic, Vaibhav Bajpai, and Jorg Ott. Tracing the path to YouTube: A quantification of path lengths and latencies toward content caches. *IEEE Communications Magazine*, 57(1):80–86, 2019.

- [43] Hamid R. Barzegar, Van Thanh Le, Nabil El Ioini, and Claus Pahl. Service continuity for CCAM platform in 5G-CARMEN. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1764–1769, 2020.
- [44] Laszlo A. Belady. A study of replacement algorithms for a virtual-storage computer. *IBM Systems journal*, 5(2):78–101, 1966.
- [45] Dhruv Matani, Ketan Shah, and Anirban Mitra. An  $o(1)$  algorithm for implementing the LFU cache eviction scheme. *arXiv preprint arXiv:2110.11602*, 2021.
- [46] Navneet Garg, Mathini Sellathurai, Vimal Bhatia, BN Bharath, and Tharmalingam Ratnarajah. Online content popularity prediction and learning in wireless edge caching. *IEEE Transactions on Communications*, 68(2):1087–1100, 2019.
- [47] Hao Zhu, Yang Cao, Wei Wang, Tao Jiang, and Shi Jin. Deep reinforcement learning for mobile edge caching: Review, new features, and open issues. *IEEE Network*, 32(6):50–57, 2018.
- [48] Shan Zhang, Liudi Wang, Hongbin Luo, Xiao Ma, and Sheng Zhou. AoI-delay tradeoff in mobile edge caching with freshness-aware content refreshing. *IEEE Transactions on Wireless Communications*, 20(8):5329–5342, 2021.
- [49] Yehuda Koren, Steffen Rendle, and Robert Bell. Advances in collaborative filtering. *Recommender systems handbook*, pages 91–142, 2021.
- [50] Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. Matrix factorization techniques for context aware recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 301–304, 2011.
- [51] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. *Recommender systems handbook*, pages 107–144, 2010.
- [52] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web: methods and strategies of web personalization*, pages 325–341. Springer, 2007.
- [53] Qiang Cui, Shu Wu, Qiang Liu, Wen Zhong, and Liang Wang. MV-RNN: A multi-view recurrent neural network for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 32(2):317–331, 2018.
- [54] Yinwei Wei, Wenqi Liu, Fan Liu, Xiang Wang, Liqiang Nie, and Tat-Seng Chua. Lightgt: A light graph transformer for multimedia recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 1508–1517, 2023.
- [55] Bingkun Lai, Jinbo Wen, Jiawen Kang, Hongyang Du, Jiangtian Nie, Changyan Yi, Dong In Kim, and Shengli Xie. Resource-efficient generative mobile edge networks in 6G era: Fundamentals, framework and case study. *IEEE Wireless Communications*, 31(4):66–74, 2024.
- [56] Changshi Zhou, Weiqi Liu, Tao Han, and Nirwan Ansari. Deploying on-device AIGC inference services in 6G via optimal MEC-device offloading. *IEEE Networking Letters*, 2024.
- [57] Yiliu Tang, Jason Situ, Andrea Yaoyun Cui, Mengke Wu, and Yun Huang. LLM integration in extended reality: A comprehensive review of current trends, challenges, and future perspectives. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025.

- Association for Computing Machinery.
- [58] Pengyuan Zhou, Lin Wang, Zhi Liu, Yanbin Hao, Pan Hui, Sasu Tarkoma, and Jussi Kangasharju. A survey on generative AI and LLM for video generation, understanding, and streaming. *arXiv preprint arXiv:2404.16038*, 2024.
  - [59] Yuntao Wang, Yanghe Pan, Miao Yan, Zhou Su, and Tom H Luan. A survey on ChatGPT: AI-generated contents, challenges, and solutions. *IEEE Open Journal of the Computer Society*, 4:280–302, 2023.
  - [60] Dongdong Ye, Shuting Cai, Hongyang Du, Jiawen Kang, Yinqiu Liu, Rong Yu, and Dusit Niyato. Optimizing AIGC services by prompt engineering and edge computing: A generative diffusion model-based contract theory approach. *IEEE Transactions on Vehicular Technology*, 2024.
  - [61] Samara Mayhoub and Tareq M. Shami. A review of client selection methods in federated learning. *Archives of Computational Methods in Engineering*, 31(2):1129–1152, 2024.
  - [62] Ouiame Marnissi, Hajar El Hammouti, and El Houcine Bergou. Client selection in federated learning based on gradients importance. In *AIP Conference Proceedings*, volume 3034. AIP Publishing, 2024.
  - [63] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint: 2010.01243*, 2020.
  - [64] Joel Wolfrath, Nikhil Sreekumar, Dhruv Kumar, Yuanli Wang, and Abhishek Chandra. HACCS: Heterogeneity-aware clustered client selection for accelerated federated learning. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 985–995, 2022.
  - [65] Chenyuan Feng, Yidong Wang, Zhongyuan Zhao, Tony Q. S. Quek, and Mugen Peng. Joint optimization of data sampling and user selection for federated learning in the mobile edge computing systems. In *2020 IEEE International Conference on Communications Workshops*, pages 1–6, 2020.
  - [66] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on Non-IID data with reinforcement learning. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 1698–1707, 2020.
  - [67] Jiajun Xu, Zhiyuan Li, Wei Chen, Qun Wang, Xin Gao, Qi Cai, and Ziyuan Ling. On-device language models: A comprehensive review. *arXiv preprint arXiv:2409.00088*, 2024.
  - [68] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.
  - [69] Saul I Gass. *Linear Programming: Methods and Applications*. Courier Corporation, 2003.
  - [70] Laurence A Wolsey. *Integer Programming*. John Wiley & Sons, 2020.
  - [71] IBM ILOG CPLEX Optimizer:  
<https://www.ibm.com/se-en/analytics/cplex-optimizer>.
  - [72] Gurobi Optimizer:  
<https://www.gurobi.com/products/gurobi-optimizer/>.
  - [73] Ravindra K Ahuja, Thomas L Magnantl, and James B Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-hall, 1993.

- [74] Armen S Asratian, Tristan MJ Denley, and Roland Häggkvist. *Bipartite Graphs and Their Applications*, volume 131. Cambridge university press, 1998.
- [75] John E. Hopcroft and Richard M. Karp. A  $n^2/2$  algorithm for maximum matchings in bipartite. In *12th Annual Symposium on Switching and Automata Theory (swat 1971)*, pages 122–125, 1971.
- [76] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- [77] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449–467, 1965.
- [78] Christos H Papadimitriou. Computational complexity. In *Encyclopedia of computer science*, pages 260–265. 2003.
- [79] Vijay V Vazirani. *Approximation Algorithms*, volume 1. Springer, 2001.
- [80] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions-i. *Mathematical programming*, 14(1):265–294, 1978.
- [81] Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.



# Acta Universitatis Upsaliensis

*Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 2638*

Editor: The Dean of the Faculty of Science and Technology

A doctoral dissertation from the Faculty of Science and Technology, Uppsala University, is usually a summary of a number of papers. A few copies of the complete dissertation are kept at major Swedish research libraries, while the summary alone is distributed internationally through the series Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology. (Prior to January, 2005, the series was published under the title “Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology”.)

Distribution: [publications.uu.se](http://publications.uu.se)  
urn:nbn:se:uu:diva-578505



ACTA UNIVERSITATIS  
UPSALIENSIS  
2026