# Joint Resource Management and Pricing in Edge Computing

by

Tarannum Nisha

B.Tech., Malaviya National Institute of Technology, India, 2018

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

October 2021

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the dissertation entitled:

Joint Resource Management and Pricing in Edge Computing

submitted by Tarannum Nisha in partial fulfillment of the requirements for

the degree of Master of Applied Science

in Electrical and Computer Engineering

**Examining Committee:**

Dr. Vijay K. Bhargava, Professor, Electrical and Computer Engineering, UBC

Supervisor

Dr. Vincent W.S. Wong, Professor, Electrical and Computer Engineering, UBC

Supervisory Committee Member

Dr. Mohammad Shahrad, Assistant Professor, Electrical and Computer Engineering, UBC

Supervisory Committee Member

# Abstract

Edge computing (EC) has emerged as a vital technology that works in tandem with the cloud to reduce network traffic and enhance user experience by distributing computational and storage resources closer to end-users and data sources. Despite the tremendous advancements made in EC technology and the enormous potential it holds, it is still in its infancy stage with numerous open challenges to overcome. In this thesis, we particularly aim to design efficient algorithms for pricing, service placement, resource management, and workload allocation in EC.

While considering the joint resource management and pricing problem in EC, we take into account the preferences of the services. Specifically, we propose a novel bi-level optimization framework to assist the EC platform to determine the optimal edge resource prices not only to maximize its profit, but also help each service find an optimal resource procurement and workload allocation solution to minimize its cost while improving the user experience. When there is a single edge node (EN), we derive a simple analytic solution for the underlying problem. However, for general case with multiple ENs, the follower problem becomes sophisticated. To this end, we develop two efficient approaches based on the Karush–Kuhn–Tucker (KKT) conditions and linear programming duality, respectively, combined with a series of linearization techniques to optimally solve the underlying bi-level optimization problem. The proposed optimal solution will maximize the profit of the EC platform and improve the edge resource utilization while minimizing the cost of every service. Numerical results demonstrate the superior performance of the proposed dynamic pricing scheme.

When the services need to pay for the service placement costs, the follower problems contain integer variables, which results in an extremely challenging bi-level mixed integer optimization problem. Due to the non-convex lower-level problems, we cannot use the KKT or duality-based approach to transform each lower problem equivalently into a set of linear constraints. Inspired by the column-and-constraint generation method from the adaptive robust optimization lit-

erature, we design an iterative algorithm to find an exact optimal solution to the formulated bi-level integer optimization problem.

# Lay Summary

The emerging edge computing paradigm promises to provide low-latency and ubiquitous computation to numerous mobile and Internet of Things (IoT) devices by bringing storage, computing, control, and networking functions closer to the end-users. How to efficiently allocate numerous geographically distributed heterogeneous edge resources to a variety of services (e.g., Google, Amazon, Uber) is a challenging task. In this thesis, we examine and formulate the interaction between an EC platform and multiple services as a bi-level optimization problem, which explicitly takes the service preferences into consideration. Given the optimal service placement and resource pricing computed by the platform, the goal of each service is to minimize the resource procurement cost while enhancing the user experience. The proposed framework not only maximizes the profit of the platform but also minimizes the cost of every service. We also extend the model to tackle the case where the services need to pay for the service placement costs. Extensive numerical results confirm the superior performance of the proposed dynamic pricing scheme.

# Preface

Submitted journal that resulted from the research presented in this thesis is as follows:

- T. Nisha, D. T. Nguyen, and V. K. Bhargava, "A Bilevel Programming Framework for Joint Edge Resource Management and Pricing," Accepted for publication subject to revision in *IEEE Internet of Things* journal (Date of decision: 28 October, 2021) (Accepted for presentation at *INFORMS* Annual Meeting, 2021).

I am the primary researcher and author for all the research contributions made in this thesis. I conducted the literature review to develop the original ideas and identified the research problems. I formulated the research problems, developed solution approaches, performed mathematical analysis, and carried out the numerical simulations. I also wrote the manuscript for the above submitted paper for publication. The contributions of my co-authors are as follows. Prof. Vijay K. Bhargava is my honorable supervisor and he provided valuable guidance as well as gave insightful feedback during the preparation of the manuscript. Prof. Duong Tung Nguyen is my unofficial co-supervisor. He guided me in identifying the research problems and provided constructive technical feedback during the problem formulation and mathematical analysis stage. He also helped in editorial corrections while preparing the corresponding manuscript for publication.

# Table of Contents

# List of Tables

# List of Figures

# List of Abbreviations

| | |
|---|---|
| **ADMM** | Alternating Direction Method of Multipliers |
| **AP(s)** | Access Point(s) |
| **AR** | Augmented Reality |
| **BiMIP** | Bilevel Mixed Integer Programming |
| **BS(s)** | Base Station(s) |
| **CCG** | Column and Constraint Generation |
| **DC(s)** | Data Center(s) |
| **DM** | Decision Maker |
| **EC** | Edge Computing |
| **EN(s)** | Edge Node(s) |
| **FLEN** | First Layer Edge Node |
| **IoT** | Internet of Things |
| **KKT** | Karush–Kuhn–Tucker |
| **ME** | Market Equilibrium |
| **MILP** | Mixed Integer Linear Programming |
| **MINBP** | Mixed Integer Non-Linear Bilevel Programming |
| **MINLP** | Mixed Integer Non-Linear Programming |

| | |
|---|---|
| **MP** | Master Problem |
| **MPCC** | Mathematical Program with Linear Complementarity Constraints |
| **MPEC** | Mathematical Program with Equilibrium Constraints |
| **OTT** | Over-the-Top |
| **PC** | Personal Computer |
| **POA** | Point of Aggregation |
| **SLEN** | Second Layer Edge Node |
| **SP** | Service Provider |
| **UHD** | Ultra-High-Definition |
| **VR** | Virtual Reality |

# Acknowledgements

First and foremost, I would like to express my gratitude to my MASc supervisor, Professor Vijay K. Bhargava. Without his assistance and dedicated involvement in every step throughout the process, this research work would have never been accomplished. His advises, patience and encouragement with a perfect blend of insight and humour have pushed me to sharpen my thinking and fully explore my research interests without any pressure. My time working with him is something I'm proud of, and I'm grateful for it. Indeed, I will truly miss listening to his inspiring life lessons and experiences during our virtual group lunch meetings. Professor Vijay's motivation, vision, generosity, and integrity will forever stay with me for my future career.

I would also like to extend my deepest gratitude and immeasurable appreciation to my co-supervisor, Professor Duong T. Nguyen for introducing me to this interesting line of research. The many hours of thought-provoking conversations I had with him were extremely beneficial to me. His valuable technical suggestions and feedback were instrumental in presenting my research work as clearly as possible. Professor Duong's immense knowledge and plentiful experience have aided me throughout my research and daily life. A special thanks goes to my colleagues, Surya Vara Prasad, Kevin Bradley Dsouza, and, Jiaming Cheng for not only taking time to review my thesis but also for their friendly support and fruitful discussions.

Last but not the least, I especially thank my dear family and friends for patiently listening to me and for their unfailing emotional support over the past two years. This journey would have been much more difficult and a lot less enjoyable if it hadn't been for their unwavering support, unconditional love, and encouragement.

# Dedication

*Dedicated to my beloved parents, siblings, and close friends,*
*who have been my source of inspiration and strength.*

# Chapter 1

# Introduction

## 1.1 Background and Motivation

The widespread use of mobile devices, combined with the rapid growth of cloud computing, has resulted in unprecedented volumes of data traffic across the communication network. Cloud computing provides users with a variety of benefits by allowing them to access a wide range of services and virtually unlimited resources and storage capacity [1]. This trend is expected to continue in the near future, with new applications such as 4K/8K Ultra-High-Definition (UHD) video, interactive gaming applications, tactile Internet, virtual/augmented reality (VR/AR), mission-critical communication, smart grids, smart cities, and a variety of Internet of Things (IoT) applications [2]. Owing to this trend, the network will be put under a tremendous amount of pressure as the cloud infrastructure and the number of devices continue to grow at a rapid pace. Thus, it is vital for the network operators to make fundamental decisions of handling increasing traffic alongside accommodating the diverse requirements of various services and use cases in the next generation communication network.

Cloud computing will likely continue to play a prominent role in the future computing landscape due to significant economies of scale resulting from providing the same services to a wide range of customers and supercomputing performance advantages. However, due to geographically distant placement of cloud data centers (DCs), there arises the issue of enormous network traffic that results in communication delay and jitter when processing time-driven data. Furthermore, the cloud server performance is severely affected in the event of an attack or slow Internet connectivity, among other things. Hence, despite the immense power and potential, cloud computing alone is facing growing limitations in satisfying the stringent requirements in terms of latency, reliability, security, mobility, and localization of many new systems and applications (e.g., embedded artificial intelligence, manufacture automation, 5G

wireless systems) [2]. To this end, edge computing [2, 3] has emerged as a new computing paradigm that augments the traditional cloud computing model to enable the implementation of a variety of services closer to the end-users right at the network edge.

Virtualization technology powers the edge networks to distribute computing, storage, control, and networking services closer to the end-users. This proximity to data at its source can deliver strong business advantages, including faster insights, improved response time and better bandwidth connectivity [4]. Generally, "edge" can be referred to any network resource between data sources and cloud data centers. Moreover, the size of an edge node can vary from smartphones, personal computers (PCs), smart access points (APs), base stations (BSs) to edge clouds [5]. For example, a smartphone is the edge between wearable devices and the cloud, a home gateway is the edge between smart appliances and the cloud, a cloudlet, a telecom central office, a micro DC is the edge between mobile devices and cloud core network. In this thesis, edge clouds, micro DCs in campus buildings, enterprises, hospitals, malls, and telecom central offices, servers at BSs, and idle PCs in research labs are all examples of ENs [2, 6, 7]. EC offers many remarkable capabilities by providing cloud resources and intelligence at the edge of the network. This includes local data processing and analytics, distributed caching, location awareness, resource pooling and scaling, enhanced privacy and security, and mobility support.

These capabilities in combination with short distance communication allows the operators to efficiently handle the traffic load both upstream and downstream between the cloud and the edge users, which in turn results in drastic reduction in the network traffic thereby improving the user experience. EC with its growing computing capabilities such as edge caching, near real-time data processing and analysis, can not only power innovation to enhance network performance but also help service providers serve user content requests locally. Moreover, EC helps to unlock the potential of large amounts of untapped data generated by connected devices [4]. This opens up numerous business opportunities to improve operation of the network, customer experience, workload management and secure operation.

In addition, most data generated by IoT sensors is expected to be processed at the edge and only important information is sent to the cloud for further analysis. Besides, EC is the key enabler for ultra-reliable low-latency applications such as AR/VR, multimedia streaming,

video transcoding, autonomous driving, industrial automation, remote robotics, and healthcare. Numerous advantages and other use cases (e.g., offloading, caching, advertising, smart homes/grids/cities) of EC can be found in [2, 3, 5]. It's worth noting that the intersection of EC and artificial intelligence (AI), has recently emerged as a fascinating research topic with numerous intriguing and practical applications [8–11].

Due to the tremendous potential that EC holds, it has attracted significant attention from both academia and industry. Indeed, large telecommunication companies (e.g., AT&T, Verizon) and cloud providers (e.g., Amazon, Google, Microsoft), as well as many startups and third parties, have made significant investments in EC technology. [4, 12–18]. EC is still in its early stages of development and faces numerous new challenges, including network architecture design, programming models and abstracts, IoT support, service placement, resource provisioning and management, security and privacy, incentive design, resource pricing and edge device reliability and scalability [2, 3, 5]. To fully realise the enormous potential of this new technology, significant collaboration between industry and academic professionals is required.

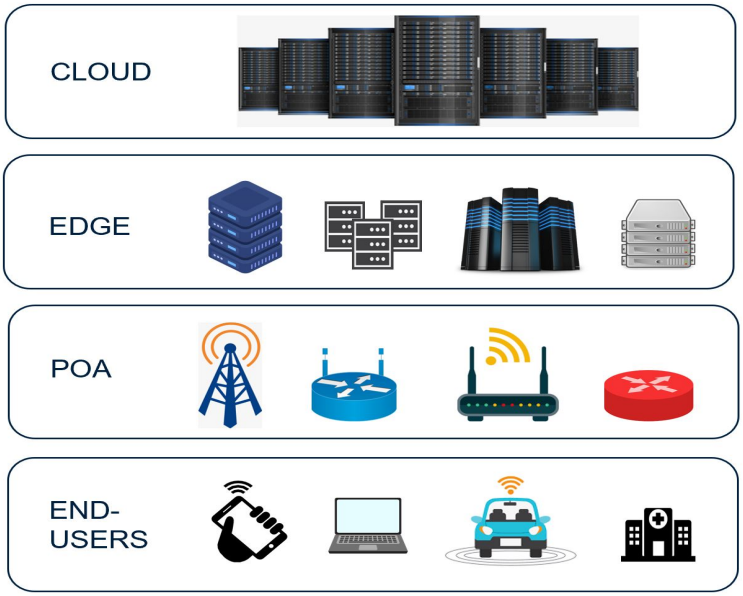



Figure 1.1: Edge network architecture

Fig. 1.1 depicts the new network architecture where the EC layer lies between the cloud and the aggregation layer. In particular, the aggregation layer consists of numerous Points of Aggregation (POA) also known as access points, such as BSs and network routers/switches,

which aggregate data and requests from end-users/IoT devices. As mentioned earlier, various sources can act as edge nodes. An EN can, in fact, be co-located with a POA. For instance, edge servers can be placed at a base station. The service requests first arrives at the APs. Thereafter, depending on whether the service is placed on the ENs or cloud, the requests are routed for further processing. For example, with EC, when a user submits an Amazon order request or an Uber ride request, the request can now be handled by an EN rather than going all the way to Amazon or Uber's remote servers. Clearly, EC can help the SP not only significantly improve service quality but also significantly reduce network bandwidth consumption.

In this thesis, we focus on addressing the pricing, resource procurement, service placement, and workload allocation in EC. Specifically, we model the interaction between an EC platform and multiple services as a bi-level optimization model to capture the diverse interests of decision makers at both levels. Optimization problems involving two decision-makers interacting sequentially are called bi-level optimization problems. When decisions are made in this hierarchical fashion, the first player in the upper level makes his decision first and conveys the same to the second player in the lower-level [19]. The first player is commonly referred to as the leader and the second player is referred to as the follower. As a result, the feasible decision set of the follower problem is influenced by the leader and since the follower problem serves as nested optimization problem to the leader, the leader's optimization problem can be solved optimally by anticipating the optimal lower-level decisions, thus reflecting the inter-dependency in decisions made by the players at both levels of the problem.

In our model, the platform manages a set of edge nodes (ENs) and acts as the leader while the services are the followers. The terms EC platform and edge infrastructure provider are used interchangeably in this thesis. The content/application/service providers (e.g., AR/VR companies, Google, Netflix, Facebook, Uber, Apple, and other over-the-top (OTT) providers) can proactively install their applications, particularly latency-sensitive and/or data-intensive ones like AR/VR, cloud gaming, and video analytics onto selected ENs closer to end-users to improve their experience and reduce bandwidth usage. By selling edge resources to the services, the platform can maximize its revenue. The services in turn compete for those edge resources given the prices announced by the platform to improve the user experience by serving the requests directly at the network edge. In particular, these decisions are quite challenging

due to the interdependence between the decisions of the platform and the services. While the resource procurement decisions of the services depend on the prices set by the platform, the pricing decisions made by the platform depends on the demand for resources by the services.

Also, because of the heterogeneity of the ENs, the services may have diverse preferences towards them. Consequently, the valuations of different ENs to a service can be different. In general, a service prefers low-priced edge resources as well as ENs with powerful hardware and geographically close to it. Intuitively, to minimize the network delay between its users and computing nodes, a service tends to procure resources from its closest ENs. Hence, some ENs can be over-demanded (e.g., ENs in or near high-demand areas) while some other nodes are under-demanded, which leads to low resource utilization. In such cases, the platform can reduce the resource prices of underutilized ENs to encourage load shifting from the overloaded ENs. Moreover, these services needs to serve a large number of users/subscribers located in different areas.

In our proposed model, each service aims to minimize the total resource procurement cost while maximizing the service quality (i.e., minimizing the delay of the requests) and the EC platform aims to maximize profit, selling these resources to multiple latency-sensitive services. Thus, how to jointly optimize the service placement, sizing, and workload allocation decisions is an important and challenging problem, which becomes even more sophisticated due to the interdependence between the decisions of the platform and the services. To this end, we present a single-leader-multi-follower bi-level optimization model where the EC platform interacts with multiple services. Given the resource pricing and placement decisions computed by the leader, each follower optimizes its resource procurement and workload allocation strategies. When there is only one EN, we can deduce a simple analytic solution to the underlying problem.

Most of the existing literature on pricing and resource provisioning issues in edge and cloud networks have special structural properties in their follower problems, that makes analysis of these problems quite restrictive. However, in our formulation, we design flexible objective functions and constraints for the follower problems. Furthermore, up to know, majority of the previous work tackles these issues from the viewpoint of a single entity, namely, the edge infrastructure provider. This motivated us to study the problem of joint pricing, placement, and resource management by incorporating price responsive agents, namely, services to utilize

the price information revealed by edge infrastructure provider to enhance user experience. Computing bi-level optimization problems is generally extremely hard. The problem becomes even more sophisticated due to several bilinear terms, which are the products of two variables.

Due to the large size of certain applications, significant downloading/data transferring costs and storage costs are incurred by the platform. As a result, the services may have to pay for the service placement cost which consists of downloading, installation and storing services of desirable size onto different edge nodes. This motivated us to extend the previous bi-level model to capture the service placement cost in the lower-level problem. The resulting problem becomes a bi-level mixed integer formulation with integer placement variables in the lower-levels, which is extremely challenging to solve. Given the computing prices computed by the leader, each service will decide on the service placement decisions. Different from the previous model, the platform is no longer responsible for the placement decisions. Thus, we model the interaction between the platform and the services such that the platform can maximize its revenue and the services can minimize their placement and procurement costs.

We propose and formulate a novel mathematical model to tackle the extended problem. We develop an iterative decomposition algorithm inspired by the column-and-constraint generation method from the robust optimization literature to present a clear mathematical foundation for solving the extended problem. Based on this motivation, we believe our work is novel and is of great significance in practice.

## 1.2 Outline of Thesis

The rest of this thesis is organized as follows:

- In Chapter 2, we study the problem of joint edge resource management and pricing. The interaction between EC platform and services is modeled as a bi-level optimization problem, which explicitly takes the service preferences into consideration. The platform solves for optimal edge resource prices whereby the services decide on optimal resource procurement and workload allocation decisions. The proposed bi-level program is then solved by reformulation using a Karush–Kuhn–Tucker-based solution and a duality-based solution, combining with a series of linearizations. We further examine a special case with

single edge node managed by the platform and propose to solve the underlying bi-level problem analytically.

- In Chapter 3, we first present the simulation settings and then compare the performance of the duality and KKT based reformulation solution approaches. This provides insights on the running time of the solution techniques. Then we compare the proposed dynamic pricing scheme against benchmark schemes. Finally, we perform sensitivity analysis to analyze the impact of important system parameters on the optimal solution.

- In Chapter 4, we present the extended bi-level mixed integer model to capture the service placement cost in the lower-level problems. The formulation now has integer service placement variables in the follower problems. Inspired by the CCG method, we elaborate on the iterative decomposition algorithm development to solve the underlying problem and provide proof of convergence.

- In Chapter 5, concluding remarks are provided and possible future research directions are discussed.

# Chapter 2

# Bi-level Programming for Joint Edge Resource Management and Pricing

One of the most important challenges in EC is the problem of multitenancy of shared and heterogeneous edge resources, which becomes even more complicated due to the interdependence between the decisions of the platform and the services. This chapter focuses on the problem of joint edge resource pricing, service placement, and workload allocation, formulating an interactive bi-level model between an EC platform and multiple services. Our model not only maximizes the profit of the platform but also minimizes the cost for every service. We present two solution approaches based on the Karush–Kuhn–Tucker conditions and the LP duality to convert the bi-level optimization problem into a single-level optimization problem. However, the resulting single-level problem contains several bilinear terms. Therefore, we introduce a series of linearization techniques to transform the bilinear terms into a set of equivalent linear constraints. It is worth noting that the proposed model can be easily modified to study other applications such as robust edge network planning and robust network slicing.

## 2.1 Introduction and Research Contributions

In this chapter, we study the interaction between an EC platform (i.e., an edge infrastructure provider), and multiple services (e.g., AR/VR applications, Google Maps). The platform (e.g., a telco, a cloud provider, a third-party [20]) manages a set of ENs and can monetize the edge resources by selling them to the services. By placing and running the services at the ENs, the service providers (SPs) can drastically enhance the quality of experience for their users since

the user requests can be served directly at the network edge.

Our work aims to address two fundamental questions: *(1) how the platform can set the edge resource prices optimally*, and *(2) how much resources a service should purchase from each EN.* These questions are challenging due to the interdependence between the decisions of the platform and the services. Specifically, the resource procurement decisions of the services depend on the resource prices set by the platform. On the other hand, the pricing decisions of the platform depends on the resource demands of the services.

To this end, we formulate a joint edge resource management and pricing problem between the platform and the services, and propose to cast it as a bi-level optimization model [21] (i.e., a Stackelberg game). It is designed such that the platform can derive optimal edge resource prices to maximize its profit, and also help each service find an optimal resource procurement and workload allocation solution to minimize its cost while improving the user experience through delay reduction. Consequently, this new market-based framework aims to harmonize the interests of different market participants so that every service is happy with its allotment while the system maintains high resource utilization. In the formulated bi-level problem, the platform is the leader and each service is a follower. The leader decides the optimal resource prices to assign to different ENs, while anticipating the reaction of the followers. Given the edge resource prices computed by the leader, each service solves a follower problem to identify the optimal amount of resource to buy from each EN, considering its delay and budget constraints.

On that account, we consider a fog node as any edge node consisting of one or more computing units (e.g., edge clouds, micro data centers in campus buildings, enterprises, hospitals, malls, and telecom central offices, servers at base stations, and idle PCs in research labs) [5]. Unlike the traditional cloud with virtually infinite capacity, ENs have limited computational power. They also come with different sizes and configurations. Furthermore, in contrast to a small number of cloud data centers (DC), there are numerous distributed ENs. Therefore, a primary concern is how to intelligently price these limited edge resources and efficiently allocate them to the competing services with diverse characteristics and preferences, considering the service priority and fairness. When the market clears, the resulting prices and allocation form a market equilibrium (ME) [6], [22].

To the best of our knowledge, this is the first bi-level programming formulation for the

joint edge service placement, resource procurement, and pricing problem. Note that while Stackelberg games have been used extensively to study various problems in EC, most of the existing models contain a simplified follower problem that normally has a special closed-form solution to facilitate the backward induction method. For our problem, we followed a similar procedure to obtain an analytic solution for the case with a single EN in the system. However, for the general case with multiple ENs, the follower problem becomes sophisticated. Also, our formulation contains integer service placement variables. Hence, backward induction cannot be applied to solve our bi-level optimization problem. Our formulation makes it easier and more flexible for the services to express their objective functions and constraints.

Bi-level optimization problems are generally extremely hard to solve [21]. In this paper, we present two solutions to compute an exact optimal solution to the formulated mixed integer non-linear bi-level program (MINBP) in the general case. The first solution relies on the Karush–Kuhn–Tucker (KKT) conditions to convert the bi-level problem into an equivalent mathematical program with equilibrium constraints (MPEC) [23], which is a mixed integer nonlinear program (MINLP). By employing the strong duality theorem and some linearization techniques, we transform this MINLP into a mixed integer linear program (MILP) that can be solved efficiently using off-the-shelf MILP solvers such as Gurobi[1] and Cplex[2].

Although the first solution can solve the bi-level program optimally, the resulting MILP has a large number of constraints and auxiliary integer variables due to the complimentary slackness constraints from the KKT conditions. Therefore, we propose an alternative solution that uses linear programming (LP) duality and a series of linearizations to convert the original bi-level problem into an equivalent MILP with significantly less number of constraints and integer variables compared to the one obtained by the KKT-based approach. Our main contributions can be summarized as follows:

- *Modeling:* We propose a novel bi-level optimization framework for joint edge resource management and pricing, where the platform optimizes the resource pricing, EN activation, and service placement decisions in the upper level while each service optimizes its workload allocation decisions in the lower level. The service preferences are explicitly

[1]https://www.gurobi.com/

[2]https://www.ibm.com/analytics/cplex-optimizer

captured in the proposed framework.

- *Solution approach:* The formulated problem is a challeging MINBP. We first present an analytic solution for the special case with a single EN. When there are multiple ENs in the system, we develop two efficient approaches based on the KKT conditions and LP duality, respectively, to optimally solve the bi-level problem.

- *Simulation:* Extensive numerical results are shown to illustrate the effectiveness of the proposed scheme, which provides a win-win solution for both the EC platform and the services. In particular, it can help increase the profit for the platform, decrease the costs for the services, and improve the edge resource utilization.

## 2.2 Related Work

The emerging EC paradigm has attracted a lot of attention from the research community. Most of the previous work has focused on the joint optimization of communication and computational resources in mobile edge computing [6]. References [7] and [24] introduce a market equilibrium approach for fair and efficient allocation of heterogeneous edge resources to budget-constrained services. A primal-dual method for online matching between edge resources and multiple services is presented in [25] to maximize the system efficiency. In [22], Zhang *et al.* combine Stackelberg game and matching theory to address the edge resource allocation problem.

Recently, the service placement and workload scheduling problem has been studied extensively. In reference [26], Lyapunov dynamic stochastic optimization method is utilized to solve the problem of joint admission control and resource allocation in edge computing for the internet of things. In [27], a two-stage robust optimization framework is proposed to optimize the service placement and sizing decisions for a service provider, taking into account the demand uncertainty. Jia *et al.* [28] employs queuing models to jointly optimize the cloudlet placement and workload allocation decisions to minimize the system response time, considering a fixed number of cloudlets. In [29], the authors propose a ranking-based heuristic algorithm for efficient cloudlet deployment in an IoT network. Reference [30] optimizes the cloudlet placement and task allocation to minimize the energy consumption subject to delay constraints. A novel workload allocation model in a hybrid cloud-fog system to minimize the total energy cost

subject to latency constraints is proposed in [31].

Reference [32] introduces a unified service placement and request dispatching model to limit service placement transitions and reduce data transmission costs. In [33], the authors present an application image placement and task scheduling problem in a fog network with dedicated storage and computing servers to minimize the makespan. R. Yu *et al.* [34] considers an IoT application provisioning problem that jointly optimizes application placement and data routing to support all data streams with both bandwidth and delay guarantees. A constant-factor approximation algorithm is presented in [35] to find a feasible service placement that maximizes the total user utility considering the heterogeneity of ENs and user locations. In [36], the authors present a joint application placement and workload allocation scheme to minimize the response time of IoT application requests. The authors of [37] present a two-timescale optimization framework to optimize service placement and request scheduling under the budget and multi-dimensional resource constraints. In [38], A. Yousefpour *et al.* introduce an edge service provisioning model to minimize the total system cost by dynamically deploying and releasing applications on different ENs. Joint optimization of access point selection and service placement is addressed in [39] to enhance user QoS by balancing the access delay, communication delay, and switching delay. The joint service placement and request routing in mobile edge computing (MEC) is investigated in [40] to minimize the workload to the cloud, considering the asymmetric bandwidth requirements of the services and the limited storage capacities of ENs. In [41], T. Ouyang *et al.* formulates the dynamic service placement problem as a contextual multi-armed bandit problem and propose a Thompson-sampling based online learning algorithm to assist a user to select an EN for offloading considering the trade-off between latency and service migration cost. Wang *et al.* in [42], examines the service placement problem for social VR applications to minimize the total application deployment cost, including the cloudlet activation, service placement, proximity, and colocation costs.

A substantial amount of research has also been carried out on pricing design in cloud and edge networks. In [43], H. Xu *et. al.* propose a revenue maximization model and employ stochastic dynamic programming to tackle the dynamic pricing problem in an IaaS cloud. In [44], various pricing frameworks have been studied to tackle the problem of joint resource provisioning and procurement applying an online procurement algorithm. Similar research on

joint virtual machine pricing, task scheduling, and server provisioning is studied in [45] via an online profit maximization algorithm. Reference [46] studies the problem of resource pricing by thoroughly analyzing several dynamic pricing schemes based on auctions and fairness-seeking properties from the perspective of game-theory and existence of unique Nash or Stackelberg equilibrium. In [47], the Lagrange multiplier method and a dynamic closed loop control scheme are integrated to solve the user perceived value-based dynamic pricing problem.

Stackelberg games and bi-level optimization have been proposed for studying resource allocation and pricing in cloud and edge computing. In [48], the authors introduce a bi-level model and a heuristic algorithm to study the task allocation problem in a two-layer multi-community cloud/cloudlet social collaborative computational framework. The authors of [49] propose a Stackelberg game between a single EN and multiple mobile users, in which the former seeks to maximize revenue within capacity constraints, while the latter seeks to minimize cost performing optimal task allocation. In [50], the authors present a bi-level optimization model, in which the upper-level model represents the task allocation problem and the lower-level captures the resource allocation problem, to minimize energy consumption under delay constraints.

Reference [51] studies the interaction among cloud/edge providers that sell computing services at the upper-level and a set of peer nodes called miners that decide on the service demand to be purchased at the lower-level. The alternating direction method of multipliers (ADMM) method is employed to solve this multi-leader multi-follower Stackelberg game. On the similar lines, a sophisticated Stackelberg game between first and second layer edge nodes (FLEN & SLEN) for energy-aware profit maximization is tackled where service price is determined by SLEN and the workload distribution by the FLEN leveraging the flat-rate pricing scheme [52]. Additionally, a sophisticated Stackelberg game has been dealt with the authors of [53] to study the bi-level resource pricing problem in mobile edge computing and solved by evolutionary algorithms. To solve the pricing and purchasing dilemma of the data consumer and the market-agency, a two-stage Stackelberg game is formulated in [54]. In [55], the authors propose a two-stage dynamic game between wireless devices and a base station connected to an edge server. In the first stage, the base station determines service pricing and placement decisions to maximize its profit whereas in the second stage, each device executes task offloading with the goal of reducing the service latency and cost.

In this work, we consider both cloudlet placement and service placement aspects of the edge resource allocation problem. The service placement and cloudlet placement/activation problems are often addressed independently in the previous literature. Furthermore, most of the existing optimization models are studied from the perspective of a centralized service provider. In this work, we jointly address the problem of optimal service placement and EN activation. Furthermore, the preferences of the services are also integrated into our proposed optimization model, which have been largely ignored in the existing literature.

In most of the existing Stackelberg games and bi-level optimization models for cloud/edge pricing, the follower problems are quite restrictive and have closed-form solutions that facilitate the application of the backward induction method to find a Stackelberg equilibrium. Some of the proposed algorithms are also heuristic and provide only suboptimal solutions. Unlike the previous literature, our proposed bi-level model allows the services to be more flexible in defining their objective functions and operating constraints, which enables the follower problems to be more expressive. Furthermore, our proposed solutions are exact and give an optimal solution to the bi-level program. Our proposed model and design objective are also different from the previous work.

## 2.3 System Model

We consider a system that consists of an EC platform, also known as an edge infrastructure provider, and a set $\mathcal{K}$ of K services. The platform manages a set $\mathcal{N}$ of N ENs to provide computational resources to the services. The services can be proactively installed onto selected ENs to reduce the communication latency and improve service quality. In practice, various sources (e.g., under-utilized DCs in schools/malls/enterprises, idle PCs in research labs, edge servers at base stations, telecom central offices) can serve as ENs [5]. In addition to its own ENs, the platform may also control ENs owned by other entities (e.g., telcos, malls, universities). The EN owners can offer their idle edge resources to the platform in exchange for a certain compensation. The service requests from end-devices normally arrive at a point of aggregation (e.g., switches, routers, base stations, WiFi access points), then will be forwarded to an EN or the cloud for processing.

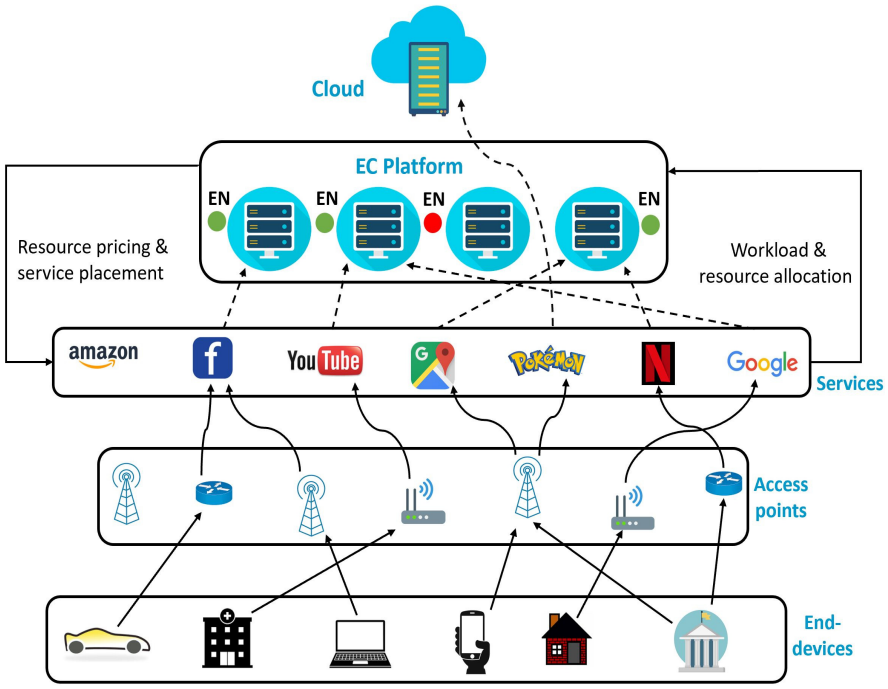



Figure 2.1: System model

Throughout this chapter, the points of aggregation are referred to as access points (AP). We assume there is a set $\mathcal{M}$ of M APs in the system. Each service serves users located in different areas, each of which is represented by an AP. Note that an EN can be co-located with an AP. For instance, edge servers can be placed at a base station. In enterprise data centers, edge servers can be deployed near routers/switches. Similar to the previous literature [7, 24, 27, 28, 30, 37, 38, 40, 42], we study the service placement and request scheduling problem from the APs to the ENs and the cloud only. Fig. 2.1 depicts the system model.

Let $i, j$, and $k$ be the AP index, EN index, and service index, respectively. The network delay between AP $i$ and EN $j$ is $d_{i,j}$, and the delay between AP $i$ and the cloud is $d_{i,0}$. The goal of each service is to minimize not only the resource procurement cost but also the network delay for its users. Define $R_i^k$ as the resource demand (i.e., workload) of service $k$ at AP $i$. Given the resource prices, locations, and specifications of the ENs, each service decides how to optimally divide its workload to the active ENs and the cloud for processing. In Fig. 2.1, each active EN is represented by a green dot while a red dot indicates an inactive EN.

Each service $k$ has a budget $B^k$ for resource procurement. The amount of workload of service $k$ at AP $i$ assigned to EN $j$ is denoted by $x_{i,j}^k$. Also, $x_{i,0}^k$ is the amount of workload of service

$k$ at AP $i$ routed to the cloud. Define $x_0^k = (x_{1,0}^k, x_{2,0}^k, \ldots, x_{M,0}^k)$, $x_i^k = (x_{i,1}^k, x_{i,2}^k, \ldots, x_{i,N}^k)$, and $x^k = (x_1^k, x_2^k, \ldots, x_M^k)$. Clearly, to enhance the user experience, a service prefers to have its requests processed by ENs closer to its users rather than the remote cloud. Let $y_0^k$ and $y_j^k$ represent the amounts of computing resources that service $k$ purchases from the cloud and EN $j$, respectively. Also, $y^k = (y_1^k, y_2^k, \ldots, y_N^k)$. Define $D^{k,\mathsf{m}}$ as the maximum delay threshold of service $k$. The average delay of service $k$ at AP $i$ is $d_i^{k,\mathsf{a}}$. Denote by $w^k$ the delay penalty parameter for service $k$. Let $s^k$ be the size of service $k$. The placement cost of service $k$ at EN $j$ is $\phi_j^k$, which includes the downloading, installation, and storage costs. The binary variable $t_j^k$ indicates if service $k$ is installed at EN $j$ or not. Define $t^k = (t_1^k, t_2^k, \ldots, t_N^k)$ and $t = (t^1, t^2, \ldots, t^K)$.

For each EN $j$, its storage capacity and computing capacity are denoted by $S_j$ and $C_j$, respectively. Since the services may have different preferences towards the ENs, some ENs can be over-demanded while others are under-demanded. Hence, a natural solution is to efficiently price the edge resources to balance supply and demand. The unit price of computing resource of EN $j$ is denoted by $p_j$. Define $p = (p_1, p_2, \ldots, p_N)$. Moreover, due to the limited storage resource, each EN can host only a subset of services. The operating cost of an active EN $j$ includes a fixed cost $c_j$ and a variable cost depending on its computing resource utilization. Let $z_j$ be a binary variable that equals one if EN $j$ is active and zero otherwise. Define $z = (z_1, z_2, \ldots, z_N)$. The platform needs to jointly decide which ENs to be active, which service to place at which node, and the resource prices of individual ENs to maximize its revenue while minimizing the total operation cost. The main notations are summarized in Table 2.1.

Our work focuses on the interaction between the platform and multiple latency-sensitive services. The platform needs to properly price resources at different ENs to maximize its profit and ensure load balancing, while considering diverse service preferences. The edge resource prices are interdependent because whether a service chooses to offload its tasks to an EN or not depends not only on the price at that EN but also on the prices at other ENs. Besides resource pricing, the platform is also responsible for downloading and installing the services onto different ENs. The placement decision is subject to the storage capacity constraints of the ENs.

Table 2.1: Notations

| Notation | Meaning |
|---|---|
| $i$, M, $\mathcal{M}$ | Index, number, and set of APs |
| $j$, N, $\mathcal{N}$ | Index, number, and set of ENs |
| $k$, K, $\mathcal{K}$ | Index, number, and set of services |
| $S_j$ | Storage capacity of EN $j$ |
| $C_j$ | Computing resource capacity of EN $j$ |
| $c_j$ | Fixed cost of EN $j$ |
| $q_j$ | Variable cost of EN $j$ |
| $d_{i,j}$ | Network delay between AP $i$ and EN $j$ |
| $d_{i,0}$ | Network delay between AP $i$ and the cloud |
| $D^{k,\mathsf{m}}$ | Delay threshold of service $k$ |
| $d_i^{k,\mathsf{a}}$ | Average delay of service $k$ in area $i$ |
| $B^k$ | Budget of service $k$ |
| $s^k$ | Storage resource requirement of service $k$ |
| $w^k$ | Delay penalty parameter for service $k$ |
| $R_i^k$ | Resource demand of service $k$ at AP $i$ |
| $\phi_j^k$ | Placement cost of service $k$ at EN $j$ |
| $t_j^k$ | Binary variable, 1 if service $k$ is placed at EN $j$ |
| $z_j$ | Binary variable, 1 if EN $j$ is active |
| $p_j$ | Unit price of computing resource at EN $j$ |
| $x_{i,j}^k$ | Workload of service $k$ at AP $i$ assigned to EN $j$ |
| $x_{i,0}^k$ | Workload of service $k$ at AP $i$ assigned to the cloud |
| $y_j^k$ | Amount of resource of EN $j$ allocated to service $k$ |
| $y_0^k$ | Amount of resource of cloud allocated to service $k$ |

By anticipating the reaction of the services, the platform optimizes the resource prices and service placement. Given the pricing and placement decisions announced by the platform, each service responds by computing its favorite edge resource bundle (i.e., the optimal amount of

resource to purchase from each EN). Since the platform acts first and the services make their decisions based on the platform's decisions, the process is sequential. Thus, we model the interaction between the platform and the services as a bi-level program, where the platform and services are the leader and followers, respectively.

## 2.4 Problem Formulation

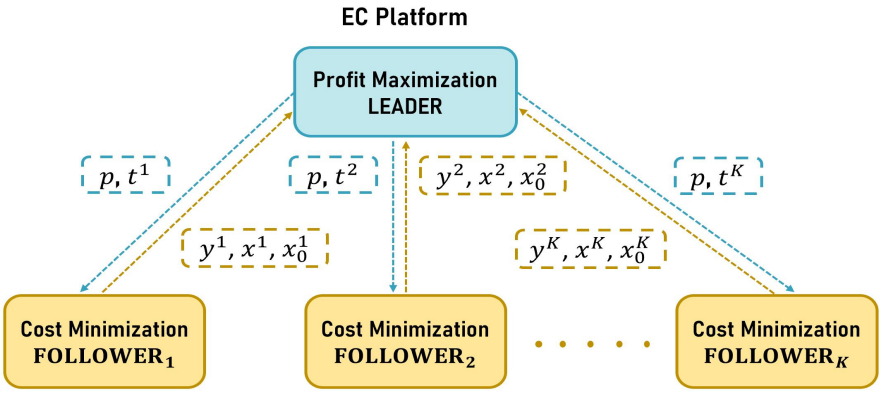



Figure 2.2: Interaction between the platform and services

In this section, we formulate the interaction between the platform (i.e., the leader) and the services (i.e., the followers) as a bi-level program which consists of an upper-level optimization problem and $K$ lower-level problems, each for one service. The platform solves the upper-level problem to maximize its profit, and then announces the resource prices and service placement decisions to the services. After receiving the information from the platform, each service solves a lower-level problem to minimize its cost under the delay and budget constraints, and then send the optimal resource procurement and workload allocation solution back to the platform.

Fig. 2.2 summarizes the interaction between the platform and services. In bi-level programming, the upper-level problem is commonly referred to as the leader problem while the lower-level ones are the follower problems. The optimal solutions of the followers and the leader are interdependent. In particular, the decisions of the followers serve as input to the profit maximization problem of the leader. The output of the leader problem also directly affects the followers' decisions. The follower problems are indeed constraints to the leader problem. In the following, we will describe the follower problem for each service, the leader problem for

the platform, as well as the bi-level optimization model.

### 2.4.1 The Follower Problem

Given the resource prices and service placement decisions announced by the platform, each service aims to minimize not only the resource procurement cost but also the total network delay by judiciously distributing its workload to the cloud and the ENs that have installed the service. The cost of service $k$ for purchasing cloud resource is $p_0 y_0^k$, where $p_0$ is the unit resource price at the cloud. The total cost of service $k$ for purchasing edge resources is $\sum_j p_j y_j^k$. Thus, the total resource procurement cost for service $k$ is $p_0 y_0^k + \sum_j p_j y_j^k$. The delay cost between AP $i$ and EN $j$ is proportional to the amount of workload allocation from AP $i$ to EN $j$ and the network delay between them. Hence, the delay cost of service $k$ can be expressed as $w^k(\sum_i x_{i,0}^k d_{i,0} + \sum_{i,j} x_{i,j}^k d_{i,j})$. The goal of service $k$ is to minimize the following objective function, which is the sum of its resource cost and delay cost:

$$p_0 y_0^k + \sum_j p_j y_j^k + w^k \left( \sum_i x_{i,0}^k d_{i,0} + \sum_{i,j} x_{i,j}^k d_{i,j} \right), \tag{2.1}$$

where the delay penalty parameter $w^k$ can be adjusted by the service to control the trade-off between the resource procurement cost and the delay cost. A higher value of $w^k$ implies that the service is more delay-sensitive and willing to pay more to buy edge resources to reduce the overall delay. Note that the actual payment of each service is the resource procurement cost only. The delay penalty cost is a virtual cost, which is used to express the delay-sensitive level of the service.

The constraints of the follower problem are described in the following. First, the total workload of service $k$ allocated to EN $j$ cannot exceed the amount of computing resource purchased from the EN, i.e., we have:

$$\sum_i x_{i,j}^k \leq y_j^k, \quad \forall j, k. \tag{2.2}$$

Similarly, for the resource purchased from the cloud, we have:

$$\sum_i x_{i,0}^k \leq y_0^k, \quad \forall k. \tag{2.3}$$

The resource demand from AP $i$ must be served by either the cloud or some EN, which implies:

$$x_{i,0}^k + \sum_j x_{i,j}^k = R_i^k, \quad \forall i, k. \tag{2.4}$$

While the capacity of the cloud is virtually unlimited, the resource of each EN is limited. Hence, the amount of resource purchased from an EN cannot exceed the capacity of that node. Additionally, service $k$ buys resources from EN $j$ only if the service is placed on EN $j$ (i.e., $t_j^k = 1$). Therefore:

$$y_j^k \leq C_j t_j^k, \quad \forall j, k. \tag{2.5}$$

Since the total resource procurement cost of a service is limited by its budget, we have:

$$p_0 y_0^k + \sum_j p_j y_j^k \leq B^k, \quad \forall k. \tag{2.6}$$

The average delay of service $k$ in area $i$ can be expressed as:

$$d_i^{k,\mathsf{a}} = \frac{x_{i,0}^k d_{i,0} + \sum_j x_{i,j}^k d_{i,j}}{R_i^k}, \quad \forall i, k. \tag{2.7}$$

For a delay-sensitive service, it may require that the average delay in every area should not exceed a certain delay threshold, which imposes $d_i^{k,\mathsf{a}} \leq D^{k,\mathsf{m}}, \ \forall i, k$.

Furthermore, each service may have certain hardware and software requirements for the ENs that can host the service. For example, some service can only be deployed on ENs that support TensorFlow and Ubuntu. Additionally, if a service is delay-sensitive, its requests from any area should be handled by ENs that are not too far from that area. Thus, we use a binary indicator $a_{i,j}^k$ to indicate whether EN $j$ is eligible to serve the demand of service $k$ at AP $i$ or not. Clearly, we have:

$$x_{i,j}^k \leq a_{i,j}^k R_i^k, \quad \forall i, j, k. \tag{2.8}$$

Overall, the follower problem for service $k$ can be written as follows:

$$\min_{x^k, y^k} \ p_0 y_0^k + \sum_j p_j y_j^k + w^k \left( \sum_i x_{i,0}^k d_{i,0} + \sum_{i,j} x_{i,j}^k d_{i,j} \right) \tag{2.9}$$

subject to

$$x_{i,0}^k + \sum_j x_{i,j}^k = R_i^k, \quad \forall i \quad (\xi_i^k) \tag{2.10}$$

$$\sum_i x_{i,0}^k \leq y_0^k, \quad (\mu_1^k) \tag{2.11}$$

$$\sum_i x_{i,j}^k \leq y_j^k, \quad \forall j \quad (\lambda_j^k) \tag{2.12}$$

$$y_{i,j}^k \leq C_j t_j^k, \quad \forall j \quad (\Gamma_j^k) \tag{2.13}$$

$$x_{i,j}^k \leq a_{i,j}^k R_i^k, \quad \forall i,j \quad (\eta_{i,j}^k) \tag{2.14}$$

$$\sum_j p_j y_j^k + p_0 y_0^k \leq B^k \quad (\mu_2^k) \tag{2.15}$$

$$x_{i,0}^k d_{i,0} + \sum_j x_{i,j}^k d_{i,j} = d_i^{k,\mathsf{a}} R_i^k, \quad \forall i \quad (\sigma_i^k) \tag{2.16}$$

$$d_i^{k,\mathsf{a}} \leq D^{k,\mathsf{m}}, \quad \forall i \quad (\tau_i^k) \tag{2.17}$$

$$x_{i,0}^k \geq 0, \quad \forall i \quad (\zeta_{i,0}^k) \tag{2.18}$$

$$x_{i,j}^k \geq 0, \quad \forall i,j, \quad (\epsilon_{i,j}^k) \tag{2.19}$$

where the notations in the parentheses associated with the constraints are the Lagrange multipliers of the corresponding constraints. It is easy to see from the follower problem (2.9)-(2.19) that a service buys resource from an EN only if the gain from delay reduction outweighs the cost increment due to the price difference between the cloud resource and edge resource. Note that we have $K$ follower problems, each for one service. In addition, although $p$ and $t$ are variables in the leader problem, they are parameters in the follower problems.

### 2.4.2 The Leader Problem

The objective of the platform is to maximize its profit which is equal to revenue minus cost. The revenue of the platform obtained from selling computing resources is $\sum_j p_j \sum_k y_j^k$, where $\sum_k y_j^k$ is the total amount of computing resource from EN $j$ allocated to the services. The total cost of the platform includes the operating cost of the ENs and the service placement cost. The operating cost of an EN depends on the electricity price and power consumption of the node. For simplicity, as commonly assumed in the literature [56, 57], the operating cost of a node is approximated by a linear function. When an EN is active, its operating cost is the sum of a fixed cost and a variable cost which depends on its computing resource utilization.

Thus, the operation cost of EN $j$ can be expressed as:

$$Cost_j^{\mathsf{e}} = c_j z_j + q_j \frac{\sum_k y_j^k}{C_j}, \quad \forall j. \tag{2.20}$$

The second term is actually $q_j z_j \frac{\sum_k y_j^k}{C_j}$. However, we later enforce that $y_j^k = 0, \forall k$ if $z_j = 0$ (i.e., if EN $j$ is not active). Hence, we can ignore $z_j$ in the second term. Note that if EN $j$ is owned by a third party, we can simply set $q_j = 0$ in (2.20), and interpret $c_j$ as the price of the EN offered by the third party and $z_j$ as a binary indicator which equals one if the platform buys EN $j$ and zero otherwise.

Besides the electricity cost, in this work, we consider the setting where the platform is also responsible for the service placement cost. The placement cost $\phi_j^k$ captures the downloading, installation, and storage costs of service $k$ at EN $j$. Since a service can only operate on an active EN, the cost of running service $k$ on EN $j$ is $\phi_j^k t_j^k, \ \forall j, k$.

Overall, the profit of the platform is:

$$\mathcal{P} = \sum_j p_j \sum_k y_j^k - \sum_j \left[ c_j z_j + q_j \frac{\sum_k y_j^k}{C_j} + \sum_k \phi_j^k t_j^k \right]. \tag{2.21}$$

Next, we describe the leader problem's constraints. The EN activation and service placement decisions are binary. Thus:

$$t_j^k \in \{0,\ 1\},\ \forall j,\ k; \quad z_j \in \{0,\ 1\},\ \forall j. \tag{2.22}$$

Since a service can only be installed on active ENs, we have:

$$t_j^k \leq z_j, \quad \forall j,\ k. \tag{2.23}$$

We can only allocate computing resource from an active EN to the services. Furthermore, the total allocated computing resource from an EN cannot exceed its computing capacity. Therefore, we have:

$$\sum_k y_j^k \leq z_j C_j, \quad \forall j, \tag{2.24}$$

which implies if $z_j = 0$, then $y_j^k = 0,\ \forall j, k$. Hence, the services cannot receive computing resource from an inactive EN. Similarly, the total storage resource of an EN allocated to the

services is limited by its storage capacity, i.e., we have:

$$\sum_k s^k t_j^k \leq z_j S_j, \quad \forall j, \tag{2.25}$$

where $s^k$ is the storage size required for storing service $k$.

We assume that the unit resource price at each EN belongs to a predefined discrete set, i.e., we have

$$p_j \in \{p_j^1, \ldots, p_j^V\}, \ \forall j, \tag{2.26}$$

where $v \in \{1, \ldots, V\}$ represent different price options $p_j^1 < p_j^2 < \ldots < p_j^V$. This is a natural assumption since the price options can express different levels of the price (e.g., very low price, low price, medium price, high price, very high price). Another reason that we use discrete sets to express the prices is due to the linearization procedure described later in the solution approach section. Note that if the price is continuous, we can discretize the price range into a large number of intervals.

When the price $p_j$ is continuous and belongs to the range of $\left[p_j^{\mathsf{min}}, p_j^{\mathsf{max}}\right]$, we can discretize this range into $2^{H_j}$ intervals of equal length, and the price should belong to an interval. The length of an interval is $\Delta_j = \frac{p_j^{\mathsf{max}} - p_j^{\mathsf{min}}}{2^{H_j}}$. If $p_j$ lies in interval $l$, it can be expressed approximately as: $p_j = p_j^{\mathsf{min}} + \Delta_j(l-1)$. Thus, we can express the resource price as:

$$p_j = p_j^{\mathsf{min}} + \sum_{l=1}^{2^{H_j}} \Delta_j (l-1) b_l, \tag{2.27}$$

$$\sum_{l=1}^{2^{H_j}} b_l = 1; \quad b_l \in \{0,1\}, \ \forall l, \tag{2.28}$$

where the binary variable $b_l$ indicates if $p_j$ lies in interval $l$. Since $p_j$ appears in several bi-linear terms, we write it in the form of (2.27)-(2.28) and further linearize the product of a binary variable and a continuous variable, as shown in (2.66)-(2.67).

As $H_j$ increases, the approximation gap decreases and the accuracy increases. However when $H_j$ is large, expressing $p_j$ in forms of (2.27)-(2.28) will create a large number of binary variables. To address this issue, we can use binary expansion to express the price more

efficiently. We have:

$$p_j = p_j^{\mathsf{min}} + \sum_{l=1}^{2^{H_j}} \Delta_j (l-1), \tag{2.29}$$

$$l = \sum_{h=0}^{H_j} 2^h b_h; \quad b_h \in \{0,1\}, \ \forall h. \tag{2.30}$$

Therefore, by expressing $p_j$ as (2.27)-(2.28) or (2.29)-(2.30), we can linearize all the bilinear terms related to $p_j$ during KKT-based reformulation and duality-based reformulation.

Note that in our formulation we consider a binary variable $r_j^v$ which equals one if the resource price at EN $j$ is $p_j^v$. Since the price can take only one value, we have:

$$p_j = \sum_v p_j^v r_j^v, \ \forall j; \ \sum_v r_j^v = 1, \ \forall j; \ r_j^v \in \{0,1\}, \forall j, v. \tag{2.31}$$

We are now ready to present the leader problem, which is indeed a bi-level program as presented below:

$$\max_{p,z,t,x,y} \sum_{j,k} p_j y_j^k - \left[ \sum_j \left( c_j z_j + q_j \frac{\sum_k y_j^k}{C_j} \right) + \sum_{j,k} \phi_j^k t_j^k \right] \tag{2.32}$$

subject to

$$\begin{aligned} &(2.22) - (2.31) \\ &x^k, y^k \in \underset{(x^k,y^k)\in\mathcal{F}^k}{\operatorname{argmin}} \left\{ \sum_j p_j y_j^k + p_0 y_0^k + \right. \\ &\qquad \left. w^k \left( \sum_i x_{i,0}^k d_{i,0} + \sum_{i,j} x_{i,j}^k d_{i,j} \right) \right\}, \ \forall k, \end{aligned} \tag{2.33}$$

where $\mathcal{F}^k$ is the feasible set of $(x^k, y^k)$ satisfying constraints (2.10) – (2.19) of the follower problem for service $k$. The platform aims to maximize its profit by jointly optimizing the EN activation, service placement, and resource pricing decisions. The follower problems (i.e., the lower-level problems) serve as constraints of the leader problem, as shown in (2.33).

## 2.5 Solution Approaches

The bi-level program (2.32)-(2.33) is generally hard to solve due to not only the constraints (2.33) in forms of optimization problems but also the bilinear terms $p_j y_j^k$ in the objective function (2.32).

For the special case with only a single EN (e.g., an edge cloud serving several areas) in the system, we can solve the bi-level problem analytically as the number of possible resource prices (i.e., $V$) at the EN is finite and small. In particular, when there is only an EN, called EN1, the workload of any service can either be served by that EN or the cloud. This can be expressed as:

$$y_0^k + y_1^k = \sum_i R_i^k, \quad \forall k. \tag{2.34}$$

By replacing $y_0^k$ by $y_1^k$ from (2.34), the budget constraint (2.15) of service $k$ implies:

$$p_0\Big(\sum_i R_i^k - y_1^k\Big) + p_1 y_1^k \le B^k, \quad \forall k \tag{2.35}$$

$$\Rightarrow \ (p_1 - p_0) y_1^k \le B^k - p_0 \sum_i R_i^k, \quad \forall k. \tag{2.36}$$

There are two cases:

i) *Case 1*: $p_1 \le p_0$. Then, all the services will buy edge resources due to lower price and latency. If the total workload of all the services exceeds the capacity of the EN (i.e., $\sum_{i,k} R_i^k \ge C_1$), the EN becomes overloaded and our problem becomes infeasible. On the other hand, if $\sum_{i,k} R_i^k \le C_1$, then workload of each service will be fully served at the EN, i.e., $y_1^k = \sum_i R_i^k, \ \forall k$. In this case, the service placement and energy costs of the platform are fixed. The amount of resource sold to the services is also fixed. Thus, the platform should set the price as large as possible to maximize its revenue and profit, i.e., the optimal unit resource price at the EN is

$$p_1^{1,*} = \max_{p_1} \left\{ p_1 \in \{p_1^1, \ldots, p_1^V\}; \ p_1 \le p_0 \right\}. \tag{2.37}$$

The platform's profit is:

$$profit_1^* = \sum_{k,i} p_1^{1,*} R_i^k - \left[ \left( c_1 + q_1 \frac{\sum_{k,i} R_i^k}{C_1} \right) + \sum_k \phi_1^k \right]. \tag{2.38}$$

ii) *Case 2*: $p_1 \geq p_0$. From (2.36), we have $y_1^k \leq \alpha^k,\ \forall k$, where $\alpha^k = \frac{B^k - p_0 \sum_i R_i^k}{p_1 - p_0}$. From the objective function of the follower problem, the cost for serving one unit of workload of service $k$ from AP $i$ at the cloud is $p_0 + w^k d_{i,0}$. Similarly, the cost for serving one unit of workload of service $k$ from AP $i$ at the EN is $p_1 + w^k d_{i,1}$. Therefore, if $p_0 + w^k d_{i,0} \leq p_1 + w^k d_{i,1}$ or equivalently $w^k(d_{i,0} - d_{i,1}) \leq p_1 - p_0$ (i.e., the increased resource cost outweighs the gain from offloading), the workload of service $k$ from AP $i$ will be fully served at the cloud (i.e., $x_{i,1} = 0$ and $x_{i,0} = R_i^k$). On the other hand, if $p_0 + w^k d_{i,0} > p_1 + w^k d_{i,1}$, service $k$ will offload its workload at AP $i$ to the EN as much as possible.

Without loss of generality, we can index the APs such that $d_{1,1} \leq d_{2,1} \leq \ldots \leq d_{M,1}$. Then, we have $p_1 + w^k d_{M,1} \geq \ldots \geq p_0 + w^k d_{i,0} > p_1 + w^k d_{h,1} \geq \ldots \geq p_1 + w^k d_{i,1}$, for some $h$. It is easy to see that the benefit of offloading increases from AP 1 to AP h. Hence, to minimize its cost, service $k$ will schedule the workload from AP 1 to AP h to the EN until the total amount of offloaded workload is equal to $\alpha^k$ due to the budget constraint. If the average delay of service $k$ is larger than $D^{k,\mathsf{m}}$, the follower problem of service $k$ is infeasible for the given value of $p_1$. By using the procedure above, given $p_1$, each service can solve its corresponding follower problem analytically, without solving problem (2.9)-(2.19).

Thus, the platform can find an optimal price $p_1^*$ by enumerating the set of possible prices $p_1 \geq p_0$. In particular, the platform can start by announcing the maximum price $p_1^V$. Then, each service responds by optimizing its resource procurement and workload allocation strategy using the procedure above and send $y_1^{k,*}$ to the platform. If $\sum_k y_1^{k,*} \leq C_1$, the platform computes its profit at the current price. Then, it announces the next price $p_1^{V-1}$ to the services. The procedure repeats until $\sum_k y_1^{k,*} > C_1$ at price $p_1^m$, which means the EN is over-demanded and the algorithm stops. By comparing its profits for different prices from $p_1^{m+1}$ to $p_1^V$, the platform selects the price $p_1^{2,*}$ that gives it the highest profit, called $profit_2^*$.

Finally, if case 1 and case 2 are both feasible, the platform will choose the higher optimal profit. Specifically, if $profit_1^* < profit_2^*$, the optimal price is $p_1^* = p_1^{2,*}$. Otherwise, $p_1^* = p_1^{1,*}$.

In the following, we tackle the general case with multiple ENs. First, we present the KKT-based approach to reformulate the bi-level problem into an equivalent single-level MILP. Specifically, the bi-level program is transformed into an MPEC by replacing each follower problem by its KKT conditions. Then, by combining several linearization approaches and

the strong duality theorem [23, 58], the resulting MPEC can be recast as an MILP. Instead of relying on the KKT conditions, the second approach employs LP duality to convert the bi-level problem into an equivalent MILP with considerably less number of constraints and integer variables compared to the one obtained from the KKT-based approach.

### 2.5.1 KKT-based Reformulation

The Karush–Kuhn–Tucker (KKT) conditions are first derivative tests (also known as first-order necessary conditions) that determine whether a solution in nonlinear programming is optimal if certain regularity conditions are met. Constrained optimization theory and algorithm development rely heavily on these conditions for a solution to be optimal. The KKT conditions are sufficient when the constraint set (i.e., solution space) is convex and the maximizing (minimizing) objective function is concave (convex). When applied to a linear-programming problem, these conditions yield the complementary slackness conditions of the primal and dual problems [59]. Consider a constrained optimization problem where we aim to minimize a function $f(x)$ under a given constraint:

$$\begin{aligned}(\mathbf{P}) : \min\ f(x)\ \ , x \in \mathbb{R} \\ s.t\ \ g(x) \leq 0 \\ h(x) = 0.\end{aligned}$$

Using the method of Lagrange multipliers, the Lagrangian is given by:

$$L(x, \lambda, \nu) = f(x) + \lambda g(x) + \nu h(x)$$

The KKT conditions are as follows, where the optimal solution for this problem, $x^*$ must satisfy all conditions below:

$$\begin{aligned}g(x) \leq 0; \quad h(x) = 0, \\ \lambda \geq 0; \quad \nu \ \text{ is unrestricted in sign}, \\ \lambda g(x) = 0, \\ f_x(x) + \lambda g_x(x) + \nu h_x(x) = 0.\end{aligned}$$

The first condition is called "primal feasibility", and it states that $x$ must satisfy all of the constraints specified in problem. According to the second condition, also known as "dual

feasibility" constraint, the dual variables associated with the inequality constraints must be non-negative. Third condition is called "complementary slackness" condition that applies only to inequality constraints and enforce a positive Lagrange multiplier when the constraint is active (=0) and a zero Lagrange multiplier when the constraint is inactive ($\geq$0). The last condition is called "stationarity" which tells that for the given dual variables $\lambda$ and $\nu$, the point $x$ minimizes the lagrangian $L(x, \lambda, \nu)$ [60].

Getting back to our problem, first, recall that the optimization variables $p$ and $t$ of the leader problem are parameters of the follower problems. Thus, for fixed values of $p$ and $t^k$, the lower-level problem (2.9)-(2.19) is a linear program, and thus convex. As a result, the KKT conditions are necessary and sufficient for optimality. Consequently, we can replace each follower problem by its corresponding KKT conditions, including the stationary, complementary slackness, primal feasibility, and dual feasibility conditions [60]. The primal feasibility conditions are (2.10)–(2.19). The Lagrangian of the follower problem (2.9)–(2.19) is:

$$
\begin{aligned}
& L^k(x^k, y^k, d^{k,\mathsf{a}}, \xi^k, \sigma^k, \tau^k, \mu_1^k, \lambda^k, \Gamma^k, \eta^k, \mu_2^k, \zeta^k, \epsilon^k) \qquad (2.39)\\
&= \sum_j p_j y_j^k + w^k \left( \sum_i x_{i,0}^k d_{i,0} + \sum_{i,j} x_{i,j}^k d_{i,j} \right) + p_0 y_0^k \\
&\quad + \sum_j \lambda_j^k \left( \sum_i x_{i,j}^k - y_j^k \right) + \mu_1^k \left( \sum_i x_{i,0}^k - y_0^k \right) \\
&+ \sum_i \xi_i^k \left( R_i^k - x_{i,0}^k - \sum_j x_{i,j}^k \right) + \sum_j \Gamma_j^k \left( y_j^k - C_j t_j^k \right) \\
&\quad + \sum_{i,j} \eta_{i,j}^k \left( x_{i,j}^k - a_{i,j}^k R_i^k \right) + \sum_i \tau_i^k \left( d_i^{k,\mathsf{a}} - D^{k,\mathsf{m}} \right) \\
&+ \sum_i \sigma_i^k \left( x_{i,0}^k d_{i,0} + \sum_j x_{i,j}^k d_{i,j} - d_i^{k,\mathsf{a}} R_i^k \right) - \sum_i x_{i,0}^k \zeta_{i,0}^k \\
&\quad + \mu_2^k \left( p_0 y_0^k + \sum_j p_j y_j^k - B^k \right) - \sum_{i,j} x_{i,j}^k \epsilon_{i,j}^k .
\end{aligned}
$$

Thus, the KKT stationary conditions are given as follows:

$$\frac{\delta L}{\delta x_{i,0}^k} = w^k d_{i,0} - \xi_i^k + \sigma_i^k d_{i,0} + \mu_1^k - \zeta_{i,0}^k = 0,\ \forall i,k \tag{2.40}$$

$$\frac{\delta L}{\delta x_{i,j}^k} = w^k d_{i,j} - \xi_i^k + \sigma_i^k d_{i,j} + \lambda_j^k + \eta_{i,j}^k - \epsilon_{i,j}^k = 0,\ \forall i,j,k \tag{2.41}$$

$$\frac{\delta L}{\delta d_i^{k,\mathsf{a}}} = -\sigma_i^k R_i^k + \tau_i^k = 0, \quad \forall i,k. \tag{2.42}$$

$$\frac{\delta L}{\delta y_0^k} = p_0 - \mu_1^k + p_0 \mu_2^k = 0, \quad \forall k \tag{2.43}$$

$$\frac{\delta L}{\delta y_j^k} = p_j - \lambda_j^k + \Gamma_j^k + p_j \mu_2^k = 0, \quad \forall j,k \tag{2.44}$$

Also, the complementary slackness, dual feasibility, and the primal feasibility conditions of the follower problems render:

$$0 \le \tau_i^k \perp D^{k,\mathsf{m}} - d_i^{k,\mathsf{a}} \ge 0,\ \forall i,k \tag{2.45}$$

$$0 \le \mu_1^k \perp y_0^k - \sum_i x_{i,0}^k \ge 0, \forall k \tag{2.46}$$

$$0 \le \lambda_j^k \perp y_j^k - \sum_i x_{i,j}^k \ge 0,\ \forall j,k \tag{2.47}$$

$$0 \le \Gamma_j^k \perp C_j t_j^k - y_j^k \ge 0,\ \forall j,k \tag{2.48}$$

$$0 \le \eta_{i,j}^k \perp a_{i,j}^k R_i^k - x_{i,j}^k \ge 0,\ \forall i,j,k \tag{2.49}$$

$$0 \le \mu_2^k \perp B^k - p_0 y_0^k - \sum_j p_j y_j^k \ge 0,\ \forall k \tag{2.50}$$

$$0 \le \zeta_{i,0}^k \perp x_{i,0}^k \ge 0,\ \forall i,k \tag{2.51}$$

$$0 \le \epsilon_{i,j}^k \perp x_{i,j}^k \ge 0,\ \forall i,j,k. \tag{2.52}$$

Note that $0 \le a \perp b \ge 0$ means $a \ge 0,\ b \ge 0$, and $ab = 0$. Constraints (2.45)-(2.52) are called complementarity constraints or equilibrium constraints. By replacing constraints (2.33) for the follower problems with the set of constraints (2.10)–(2.19) and (2.40)–(2.52), the bi-level program (2.32)-(2.33) becomes an MPEC problem. This MPEC problem has three sources of nonlinearity, including: i) the complementarity constraints (2.45)-(2.52); ii) the bilinear terms $p_j \mu_2^k$ in (2.44); and iii) the bilinear term $\sum_{j,k} p_j y_j^k$ in the objective function (2.32). To convert the MPEC problem (i.e., an MINLP) into an MILP, we need to linearize these nonlinear terms.

First, the nonlinear complementarity constraints (2.45)–(2.52) can be transformed into equivalent exact linear constraints by using the Fortuny-Amat transformation [61]. Specifically,

the complementarity condition $0 \leq a \perp b \geq 0$ is equivalent to the following set of mixed-integer linear constraints:

$$a \leq (1-u)M;\ b \leq uM;\ a \geq 0;\ b \geq 0;\ u \in \{0,\ 1\}, \tag{2.53}$$

where M is a sufficiently large constant, which is often referred to as "bigM". Therefore, the set of constraints (2.45)–(2.52) can be rewritten as:

$$D^{k,\mathsf{m}} - d_i^{k,\mathsf{a}} \leq \psi_i^k M_1;\ \tau_i^k \leq (1-\psi_i^k)M_1, \quad \forall i,k \tag{2.54}$$

$$y_0^k - \sum_i x_{i,0}^k \leq v_1^k M_2;\ \mu_1^k \leq (1-v_1^k)M_2, \quad \forall k \tag{2.55}$$

$$y_j^k - \sum_i x_{i,j}^k \leq \kappa_j^k M_3;\ \lambda_j^k \leq (1-\kappa_j^k)M_3, \quad \forall j,k \tag{2.56}$$

$$C_j t_j^k - y_j^k \leq \theta_j^k M_4;\ \Gamma_j^k \leq (1-\theta_j^k)M_4, \quad \forall j,k \tag{2.57}$$

$$a_{i,j}^k R_i^k - x_{i,j}^k \leq \rho_{i,j}^k M_5;\ \eta_{i,j}^k \leq (1-\rho_{i,j}^k)M_5,\ \forall i,j,k \tag{2.58}$$

$$B^k - p_0 y_0^k - \sum_j p_j y_j^k \leq v_2^k M_6;\ \mu_2^k \leq (1-v_2^k)M_6,\ \forall k \tag{2.59}$$

$$x_{i,0}^k \geq 0;\ x_{i,0}^k \leq \Phi_{i,0}^k M_7;\ \varsigma_{i,0}^k \leq (1-\Phi_{i,0}^k)M_7,\ \forall i,k \tag{2.60}$$

$$x_{i,j}^k \geq 0;\ x_{i,j}^k \leq \Omega_{i,j}^k M_8;\ \epsilon_{i,j}^k \leq (1-\Omega_{i,j}^k)M_8,\ \forall i,j,k \tag{2.61}$$

$$\tau_i^k \geq 0;\ \mu_1^k \geq 0;\ \lambda_j^k \geq 0;\ \Gamma_j^k \geq 0 \tag{2.62}$$

$$\eta_{i,j}^k \geq 0;\ \mu_2^k \geq 0;\ \varsigma_{i,0}^k \geq 0;\ \epsilon_{i,j}^k \geq 0 \tag{2.63}$$

$$\psi_i^k,\ \kappa_j^k,\ \theta_j^k,\ \rho_{i,j}^k,\ \Phi_{i,0}^k,\ \Omega_{i,j}^k,\ v_1^k,\ v_2^k \in \{0,1\},\ \forall i,j,k, \tag{2.64}$$

where $M_1$, $M_2$, $M_3$, $M_4$, $M_5$, $M_6$, $M_7$ and $M_8$ are sufficiently large numbers. For the bilinear terms $p_j\mu_2^k$, using (2.31), we can rewrite it as:

$$p_j\mu_2^k = \sum_v p_j^v r_j^v \mu_2^k = \sum_v p_j^v \pi_j^{v,k}, \tag{2.65}$$

where $\pi_j^{v,k} = r_j^v \mu_2^k$. Note that $\pi_j^{v,k}$ is a continuous variable and we have $\pi_j^{v,k} = \mu_2^k$ if $r_j^v = 1$ and $\pi_j^{v,k} = 0$, otherwise. Hence, using (2.65), the bilinear term $p_j\mu_2^k$ can be written as a linear function of $\pi_j^k = (\pi_j^{1,k}, \ldots, \pi_j^{V,k})$. Additionally, the constraints $\pi_j^{v,k} = r_j^v \mu_2^k, \forall j,k$ can be implemented through the following linear inequalities [62]:

$$\pi_j^{v,k} \leq M r_j^v, \quad \forall j,k,v; \quad \pi_j^{v,k} \leq \mu_2^k, \quad \forall j,k,v \tag{2.66}$$

$$\pi_j^{v,k} \geq 0, \quad \forall j,k,v; \quad \pi_j^{v,k} \geq \mu_2^k + M r_j^v - M, \quad \forall j,k,v, \tag{2.67}$$

where $M$ is a sufficiently large number.

We assume that the bi-level problem has an optimal solution and the strong duality theorem holds for every follower problem. Then, the strong duality theorem gives us the following:

$$\sum_j p_j y_j^k + p_0 y_0^k + w^k \left( \sum_i x_{i,0}^k d_{i,0} + \sum_{i,j} x_{i,j}^k d_{i,j} \right) = - \sum_{i,j} R_i^k a_{i,j}^k \eta_{i,j}^k - \sum_j C_j t_j^k \Gamma_j^k - B^k \mu_2^k + \sum_i R_i^k \xi_i^k - \sum_i D^{k,\mathsf{m}} \tau_i^k, \quad \forall k. \tag{2.68}$$

Hence, using (2.68), the bilinear term $\sum_{j,k} p_j y_j^k$ can be written as the sum of several linear terms. Note that the bilinear terms $t_j^k \Gamma_j^k$ in (2.68) is a product of a continuous variable and a binary variable. Therefore, we can linearize it similar to what we did for the bilinear terms $r_j^v \mu_2^k$ using (2.66) and (2.67).

Based on the linearization steps described above, we can then express the bi-level problem (2.32)-(2.33) with an equivalent single-level MILP as follows:

$$(\mathcal{P}_1): \max_{p,z,t,x,y} \; Rev - \left[ \sum_j \left( c_j z_j + q_j \frac{\sum_k y_j^k}{C_j} \right) + \sum_{j,k} \phi_j^k t_j^k \right]$$

subject to

$$Rev = - \sum_k \left[ p_0 y_0^k + w^k \left( \sum_i x_{i,0}^k d_{i,0} + \sum_{i,j} x_{i,j}^k d_{i,j} \right) + \sum_{i,j} R_i^k a_{i,j}^k \eta_{i,j}^k + \sum_j C_j t_j^k \Gamma_j^k + B^k \mu_2^k + \sum_i R_i^k \xi_i^k + \sum_i D^{k,\mathsf{m}} \tau_i^k \right]$$

$$p_j - \lambda_j^k + \Gamma_j^k + \sum_v p_j^v \pi_j^{v,k} = 0; \; (2.66), \; (2.67)$$

$$(2.10) - (2.19), \; (2.22) - (2.31), \; (2.40) - (2.43), \; (4.34) - (4.45),$$

where $Rev$ is the revenue of the platform from selling edge resources, i.e., $Rev = \sum_{j,k} p_j y_j^k$. Problem $(\mathcal{P}_1)$ is a large-scale MILP, which can be solved by MILP solvers.

### 2.5.2 Duality-based Reformulation

In linear programming, duality means that each problem can be analyzed from two perspectives, the primal problem or the dual problem. The feasible and optimal solutions of the dual provide very useful information about the original primal LP. The dual can be used to find upper bounds on the optimal value of the primal LP if it is a maximization problem (Similarly, if the primal problem is a minimization problem, the dual provides lower bounds). It is worth noting the term, duality gap which refers to the difference between the dual optimal solution and the primal optimal solution.

The duality gap is zero for linear and convex nonlinear problems. A solution to these problems cannot be feasible to both primal and dual unless it is the optimal solution. Therefore, searching for optimality is equivalent to searching for a solution that are feasible in both primal and dual [63]. Given the primal problem in section 2.5.1, we define its Lagrange dual problem as:

$$(\mathbf{D}) : \max_{\lambda,\nu} \ \inf_{x} L(x, \lambda, \nu)$$
$$s.t \ \ \lambda \geq 0.$$

When ($\mathbf{P}$) has a finite optimal value that coincides with the optimal value of ($\mathbf{D}$) , strong duality holds. Strong duality usually (but not always) holds for convex problems. For cases with quadratic objective and inequality constraints, strong duality holds provided Slater's condition holds [60]. Informally, the feasible region must have an interior point, according to Slater's condition.

For our problem instead of using KKT conditions, we can utilize the LP duality to transform the original bi-level problem into an equivalent MILP. We first write the dual maximization form of each lower-level minimization problem (2.9)–(2.19). Subsequently, we can replace each lower-level problem by its corresponding dual feasibility conditions, as well as equating the primal and dual objective functions [60]. The dual problem of the follower problem (2.9)–(2.19) of service $k$ is given below:

$$\underset{\xi^k,\ \sigma^k,\ \tau^k,\ \mu_1^k,\ \mu_2^k,\ \lambda^k,\ \eta^k,\ \Gamma^k}{\text{maximize}} \ \sum_i R_i^k \xi_i^k - B^k \mu_2^k$$
$$- \sum_i \sum_j R_i^k a_{i,j}^k \eta_{i,j}^k - \sum_j C_j t_j^k \Gamma_j^k - \sum_i D^{k,\mathsf{m}} \tau_i^k \tag{2.69}$$

subject to

$$\lambda_j^k - \Gamma_j^k \leq \ p_j(1+\mu_2^k), \quad \forall j \tag{2.70}$$

$$\mu_1^k \leq p_0(1+\mu_2^k) \tag{2.71}$$

$$-R_i^k \sigma_i^k - \tau_i^k \leq 0, \quad \forall i \tag{2.72}$$

$$\xi_i^k + \sigma_i^k d_{i,j} - \lambda_j^k - \eta_{i,j}^k + \epsilon_{i,j}^k \leq w^k d_{i,j}, \quad \forall i,j \tag{2.73}$$

$$\xi_i^k + \sigma_i^k d_{i,0} - \mu_1^k + \zeta_{i,0}^k \leq w^k d_{i,0}, \quad \forall i \tag{2.74}$$

$$\eta_{i,j}^k \geq 0, \ \forall i,j; \quad \mu_1^k \geq 0; \quad \mu_2^k \geq 0 \tag{2.75}$$

$$\tau_i^k \geq 0, \ \forall i; \quad \lambda_j^k \geq 0, \ \forall j; \quad \Gamma_j^k \geq 0, \ \forall j. \tag{2.76}$$

The dual feasibility constraints are (2.70)-(2.76). Thus, the complete form of the final MILP optimization problem resulting from the duality-based reformulation is given as follows:

$$(\mathcal{P}_2): \underset{p,x,y,z,t,\lambda,\xi,\eta,\tau,\mu_1,\mu_2,\Gamma,\sigma}{\text{maximize}} \ \sum_k Rev^k \tag{2.77}$$
$$-\left[\sum_j \left(c_j z_j + q_j \frac{\sum_k y_j^k}{C_j}\right) + \sum_{j,k} \phi_j^k t_j^k\right]$$

subject to

$$Rev^k + p_0 y_0^k + w^k \left(\sum_i x_{i,0}^k d_{i,0} + \sum_{i,j} x_{i,j}^k d_{i,j}\right) =$$
$$-\sum_{i,j} R_i^k a_{i,j}^k \eta_{i,j}^k - \sum_j C_j t_j^k \Gamma_j^k - B^k \mu_2^k$$
$$+\sum_i R_i^k \xi_i^k - \sum_i D^{k,\mathsf{m}} \tau_i^k, \quad \forall k \tag{2.78}$$

$$(2.70) - (2.76), \quad \forall k \tag{2.79}$$

$$(2.10) - (2.19), \ (2.22) - (2.31).$$

Note that $Rev^k$ is the revenue from selling edge resources to service $k$, i.e., $Rev^k = \sum_j p_j y_j^k, \ \forall k$. Constraints (2.78) in problem $(\mathcal{P}_2)$ enforce the primal objective function equals the dual objective function, which indeed expresses the strong duality theorem. We can linearize the bilinear terms $t_j^k \Gamma_j^k$ in (2.78) by applying the same procedure that we employed in the KKT-based

transformation approach. Consequently, using (2.78), we can linearize the bilinear terms $p_j y_j^k$. Finally, the dual feasibility constraints (2.70)-(2.76) should hold for all $k$.

Compared to the MILP problem ($\mathcal{P}_1$), it is easy to see that the MILP problem ($\mathcal{P}_2$) does not need to deal with the complementarity constraints (2.45)-(2.52) or their equivalent linear constraints (4.34)-(4.45). Thus, it drastically reduces the number of constraints and auxiliary binary variables in problem ($\mathcal{P}_1$). As a result, solving ($\mathcal{P}_2$) is normally faster than solving ($\mathcal{P}_1$).

## 2.6 Summary

In this chapter, we studied the joint optimization problem of resource pricing, service placement, resource sizing, and workload allocation taking into consideration the service preferences. We derive a simple analytic solution for a single EN case and present a Karush–Kuhn–Tucker-based solution and a duality-based solution for the general case with multiple ENs and multiple services, to solve the bi-level problem. We thus transform the MINLP into a mixed integer linear program (MILP) that can be solved efficiently using off-the-shelf MILP solvers such as CPLEX or Gurobi. The aim of the formulated problem is to highlight the benefits of adopting dynamic pricing in edge computing networks compared to the fixed pricing schemes in addition to minimizing sum of the resource procurement cost, and the total network delay cost of the services, while considering various system design criteria such as resource capacity limits, budget constraint, and delay preference.

One major drawback of bi-level optimization problems is that they are non-convex and hence difficult to solve. Because of the inter-dependency between the decision variables in the upper and lower level problems, the resulting bi-level program would be non-convex even if the problem at both levels are convex. Additionally, if the optimization problems at both levels were limited to linear programs, the bi-level program would still be challenging to solve. Another hiccup with Stackelberg games (or bi-level programs) is the underlying assumption that the leader foresees how the other players will react. In other words, it is assumed that the other players will react rationally. However, in practice, this may not be the case. Using bounded rationality to model the behaviour of players is one way to deal with this. It includes using behavioural models that account for errors in a player's best response selection as well as

the fact that players are not always rational. Nevertheless, computational difficulties arise in this modeling approach due to extra parameters involved. Thus our work focuses on providing flexible and extensible tools for services and infrastructure providers to optimize their operation and planning strategies.

# Chapter 3

# Numerical Results

## 3.1 Simulation Setting

Similar to the previous work [24, 27, 28], we adopt the widely-used Barabasi-Albert model [64] to generate a random scale-free edge network topology. The number of nodes in the graph (N), initial number of nodes in the network (m0), number of nodes with which a new node in the network connects to (m), minimum and maximum link delay are the important parameters of a Barabasi-Albert network generator. We start with an initial connected graph of m0 nodes and gradually add new nodes to the network. Each new node "attaches" to a maximum of m0 nodes in the network at random (m $<$ m0), resulting in a maximum of m0 new links. The network generator will continue in this manner until the network has N nodes. We generate an edge network topology with 100 nodes and an attachment rate of 2. The link delay between each pair of nodes is randomly generated in the range of [2, 5] *ms*. The network delay between any two nodes is computed as the delay of the shortest path between them. In the **base case** scenario, we consider a small system with 6 services, 10 APs and 4 ENs, which are picked randomly in the set of 100 nodes. Thus, in the base case, M = 10, N = 4, K = 6. We will also run the proposed algorithms on different system sizes for sensitivity analysis later. The delay between each AP and the remote cloud is set to be 60 *ms*. The maximum delay threshold $D^{k,\mathsf{m}}$ for each service is selected randomly between 30 *ms* and 100 *ms*. During the scheduling horizon, for each service, the resource demand (i.e., workload) in each area is randomly drawn in the range of [20, 35] vCPUs.

Each EN is chosen randomly from the set of Amazon EC2 M5 instances. Using the hourly price of a general purpose m5d.xlarge Amazon EC2 instance [65] as reference, the unit resource price at the cloud is set to be 0.01 while the set of possible unit prices of edge resources is [0.01, 0.02, 0.03, 0.04, 0.05]. The fixed and variable operational costs $c_j$ and $q_j$ of each EN $j$ are set

in the range of [0.05, 1.8] and [0.04, 1.44], respectively, depending on the size of the EN. The delay penalty parameters $w^k$ are generated randomly between $10^{-5}$ and $10^{-3}$. Additionally, the placement cost of each service at each EN is set to be 0.02 (i.e., $\phi_j^k = 0.02,\ \forall j, k$). The size of each service is randomly generated between 10 and 100. The budget of each service is chosen in the range of [150, 300].

Unless stated otherwise, the default setting is used in most experiments. Our computational study is made through Matlab 2020b software and solved by Gurobi solver on an Intel Core i7-10510U CPU and 16 GB RAM laptop. The computational time limit is set to 10,000 seconds.

## 3.2 Performance Evaluation

### 3.2.1 Comparison between the KKT-based and duality-based reformulation approaches

We compare the computational time between the KKT-based and duality-based solution methods. Both methods allow us to compute an optimal solution to the original bi-level problem. The two methods are compared under different system sizes by varying the numbers of APs, ENs, and services. The computational results are reported in Table 3.1. Note that "NA" implies a method cannot produce a solution within the time limit. As expected, the duality-based approach offers superior performance compared to the KKT-based approach. It is because of the smaller size of the MILP obtained from the duality-based method compared to the one obtained from the KKT-based method. Another disadvantage of the KKT-based method is that we have to choose suitable bigM values, which greatly affects its running time. We thus adopt the duality-based method to generate results in the following experiments.

### 3.2.2 Comparison between dynamic, flat, and average pricing schemes

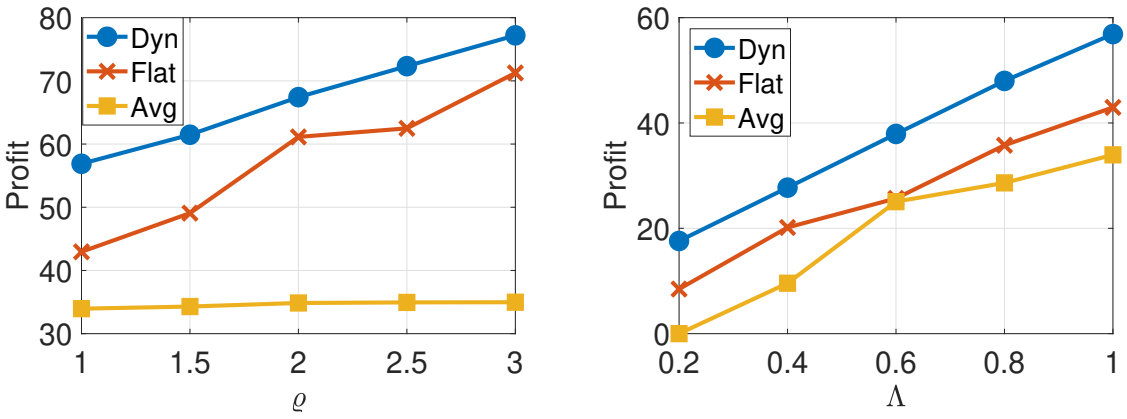
The dynamic pricing scheme ($Dyn$) is the proposed model where the resource prices at the ENs can be different to balance supply and demand. In the flat pricing scheme ($Flat$), we solve the same bi-level model with an extra constraint in the leader problem that enforces the resource prices at all the ENs to be equal. In the average pricing scheme ($Avg$), the unit resource prices at the ENs are the same, which are simply set to the average value in the set of possible prices.

| K = 6, N = 4, varying M | | |
|---|---|---|
| M | Duality (seconds) | KKT (seconds) |
| 2 | 5.8930 | 8.8937 |
| 4 | 7.9730 | 25.7110 |
| 6 | 6.9494 | 99.5641 |
| K = 6, M = 10, varying N | | |
| N | Duality (seconds) | KKT (seconds) |
| 4 | 11.993 | 53.4655 |
| 6 | 175.0676 | 515.0104 |
| 8 | 288.42 | 4919.8 |
| M = 10, N = 4, varying K | | |
| K | Duality (seconds) | KKT (seconds) |
| 4 | 3.9131 | 39.5804 |
| 6 | 9.8309 | 53.5309 |
| 8 | 74.3804 | NA |
| 10 | 146.3724 | NA |

Table 3.1: Computational time comparison between the duality-based and KKT-based reformulation methods

We first examine the impact of cloud resource price on the profit of the platform, as shown in Fig. 3.1(a). We define $\varrho$ as the scaling factor of the cloud price. For example, when $\varrho = 1.5$, the cloud resource price in the base case increases 1.5 times.

It can be seen that the proposed dynamic pricing scheme significantly outperforms the flat and average pricing schemes. Indeed, with dynamic pricing, the platform can reduce the prices of under-demanded ENs, which incentivizes the services to reallocate their workload to these nodes, thus, improving revenue and resource utilization of these nodes. Also, the prices of highly-demanded ENs are often set to the maximum price value. Hence, the profit in *Dyn* is highest due to higher edge resource demand and generally higher prices. Since the service buys more edge resources, the average network delay of each service tends to decrease. In both the




(a) Varying cloud price

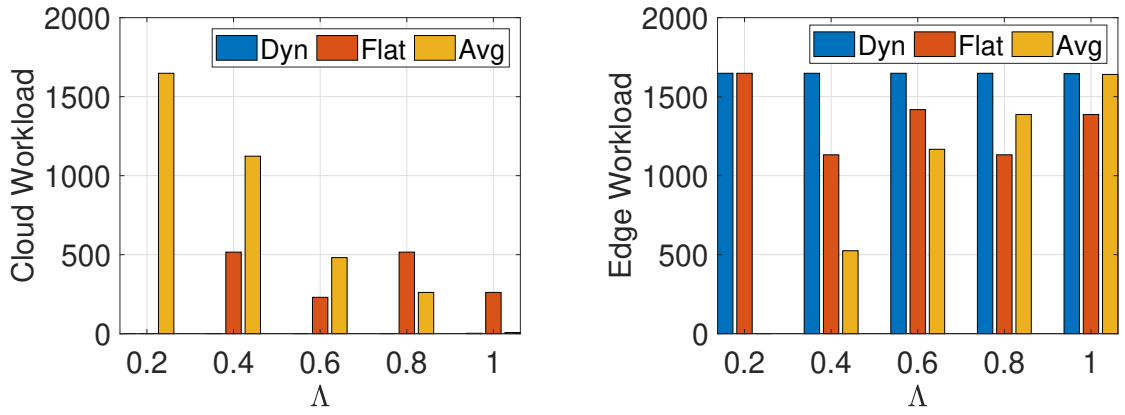
(b) Varying delay penalty

Figure 3.1: Performance comparison between *Dynamic*, *Flat*, and *Average* pricing schemes

flat and average pricing schemes, the prices at all the ENs are the same. Thus, the services do not have any incentive for load shifting. Furthermore, we can see that the profits in all these schemes increase as the cloud resource price increases. This is because an increase in the cloud resource price would encourage the services to shift more workload to the ENs. Thus, the platform can sell more edge resources and increase it revenue and profit.

Fig. 3.1(b) shows the impact of delay penalty on the profit of the platform. Let $\Lambda$ be the scaling factor of the delay penalty parameters $w^k$. It is easy to see the superior performance of the dynamic pricing scheme compared to the flat and average pricing schemes. Also, the profit increases as the delay penalty parameters increase. It is because when the services are more delay-sensitive, they are willing to pay more for edge resources to reduce the overall delay for their users. Hence, the platform can increase the edge resource prices to increase its profit.

Figs. 3.2(a)-3.2(b) depict the total workload at the cloud and at the edge under the three pricing models with varying delay penalty. In *Dyn*, all the demands are served at the edge and there is no cloud traffic. The reason is that *Dyn* allows the platform to adjust the edge prices. Hence, the prices at under-demanded ENs can be reduced so that the gain from offloading at these nodes outweighs the price difference between the cloud and each node. In *Flat*, the platform is less flexible in setting the prices since the prices at all the ENs are equal. A low edge resource price will affect the revenue of the platform. Thus, the flat price should not be too low to maximize the profit. As a result, in *Flat*, a portion of demand will go to the




(a) Total workload at the cloud

(b) Total workload at the edge

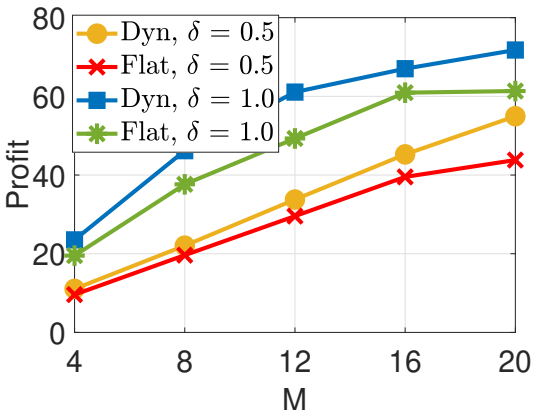
Figure 3.2: Workload allocation comparison

cloud. Finally, in the average pricing scheme, the prices at the ENs are the same and fixed. Thus, when the delay penalty increases, the gain from offloading to the edge increases and less workload goes to the cloud.

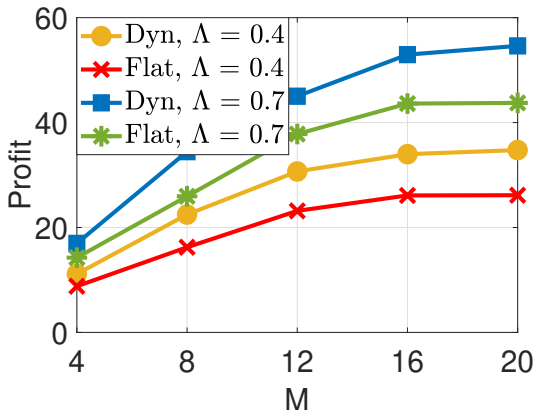
### 3.2.3 Sensitivity Analysis

We now study the effects of various design parameters on system performance. Figs. 3.3(a)-3.3(d) summarizes the impact of number of APs on the system performance with varying demand, delay penalty, capacities of ENs, and cloud price by factors of $\delta$, $\Lambda$, $\gamma_0$, and $\varrho$ respectively. These figures further confirm the superior performance of *Dyn*. Also, when there are more APs, the total workload in the system increases. Thus, we can see that the profit increases when M increases due to increasing workload.

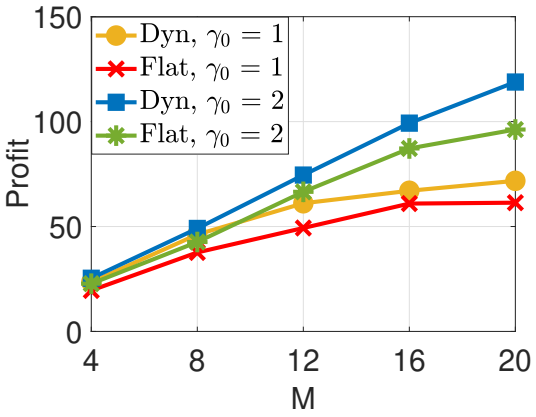
When we increase the demand, the profit further increases, as shown in Fig. 3.3(b). In Fig. 3.3(c), the profit increases when there are more edge resources (i.e., when the EN capacities increase) because the services can buy more resources from the ENs that are beneficial for them to do offloading. Fig. 3.3(d) show that when the cloud price increases, the services will allocate more workload to the edge, which leads to increasing profits for the platform. Figs. 3.4(a) and 3.4(b) further illustrate the impacts of the resource demands of the services on the system performance. Similar to the results in Figs. 3.3(a)-3.3(d), the profit increases as the resource demands, the delay penalty, and the capacities of ENs increase.



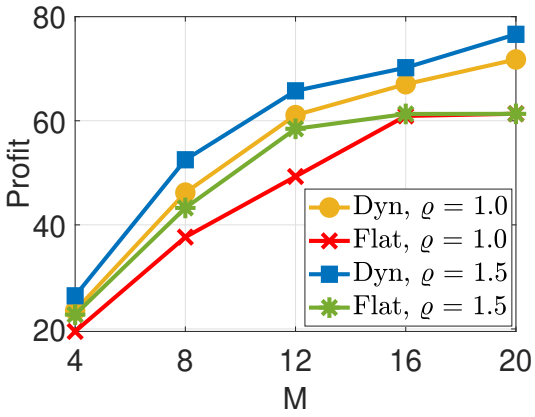

(a) Varying M and resource demand




(b) Varying M and delay penalty



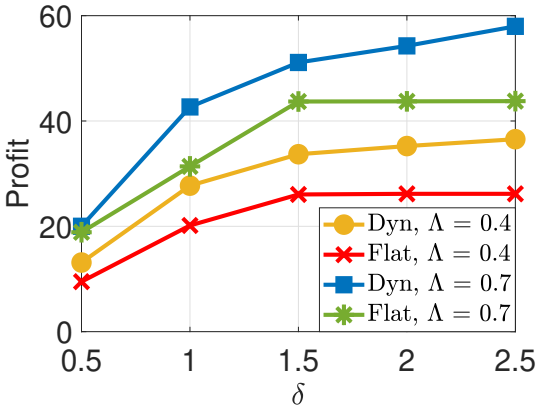

(c) Varying M and EN capacities



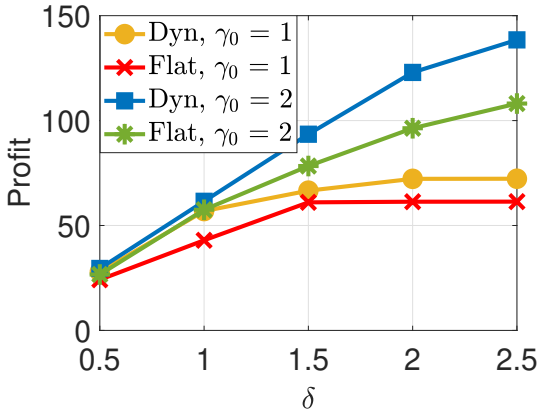

(d) Varying M and cloud price

Figure 3.3: Impacts of number of APs on the system performance

Figs. 3.5(a)-3.5(d) present the impacts of the number of services on the optimal solution. It can be observed that the profit increases as the number of services increases. It is because when there are more services, it imposes higher resource demand in the system. Furthermore, when more services compete for edge resources, the platform can raise the edge resource prices to increase its profit. Figs. 3.5(c)-3.5(d) show the total amount of edge resource procurement of all the services. It is easy to see that the total workload at the ENs increases as the number of services increases due to increasing demand. When the resource demand increases (i.e., from $\delta = 0.5$ to $\delta = 1$), the amount of procured edge resources also increases.

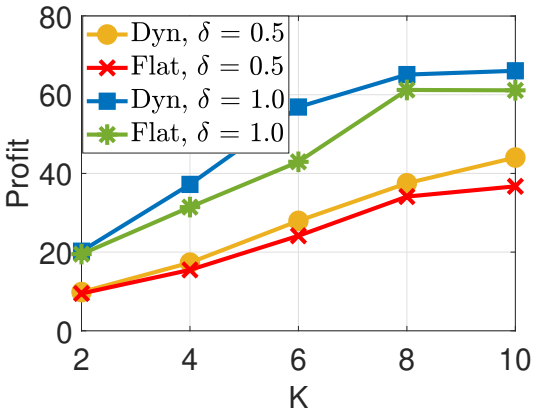



(a) Varying R and delay penalty

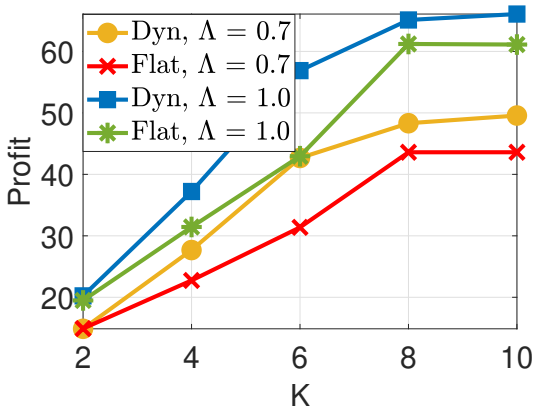



(b) Varying R and EN capacities

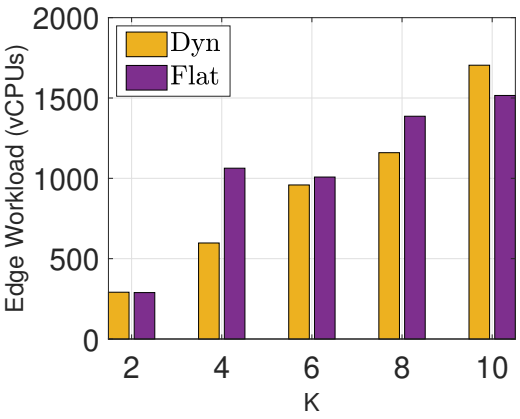
Figure 3.4: Impacts of resource demand on the system performance



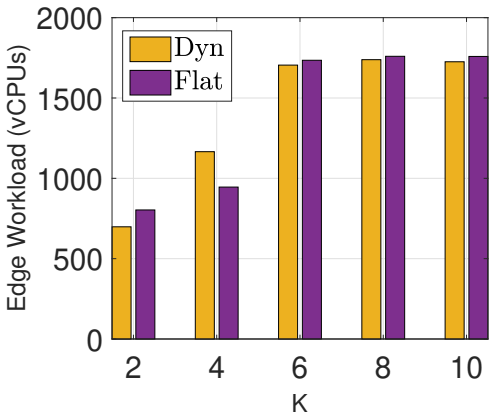

(a) Varying K and resource demand




(b) Varying K and delay penalty




(c) Demand scaling factor $\delta = 0.5$




(d) Demand scaling factor $\delta = 1$

Figure 3.5: Impacts of number of services on the system performance

# Chapter 4

# Extension: Bi-level Mixed Integer Optimization Problem

## 4.1 Motivation

Due to the large size of certain applications (e.g., image and video processing), greater storage cost is incurred by the EC platform and thus it becomes crucial to design a model that largely focuses on maximizing the revenue of the platform. To address this, we now have the service providers pay for the service placement costs in order for the services to minimize this cost by intelligently deciding on placing and running services of desirable size on suitable edge nodes i.e., at edge nodes that has its favorite resource bundle. This motivated us to discuss a variant of the previous model in this chapter.

One of the most important challenges that arises now in this variant formulation is the integer service placement decisions in lower-levels of the problem. For a bi-level problem with convex linear follower problem, we can reformulate using the classical KKT or LP duality-based reformulation techniques to obtain a single-level mixed integer linear program. Indeed, because these key structural features apply solely to LP, strong duality does not hold for bi-level MIP with integer variables in lower-levels of the problem resulting in optimality gap. For each follower problem, we have integer constraints and we need to enumerate over all possible integer variables, resulting in a large bi-level MIP. Hence, the computational complexity greatly increases, making it difficult to derive exact global optimal solutions.

To this end, we propose and design a decomposition algorithm to enumerate over all possible integer variables in an iterative fashion to compute exact global optimal solutions in finite iterations. For a given optimal lower-level integer variable, the bi-level MIP reduces to lower-level LP that can be solved by an extended set of equivalent constraints using KKT or duality-

based reformulation approaches.

Specifically, we decompose the comprehensive bi-level MIP into a master problem (MP) and two subproblems (SPs). At every iteration we solve the master and subproblems to obtain lower and upper bounds respectively. First step is to obtain the master problem for which we begin by reformulating the lower-level problem for a given optimal lower-level integer variable obtained in the previous iteration. These reformulated constraints serve as cuts to the single-level mixed integer linear program and allow the upper-level decision maker to use these cuts to generate a feasible solution space that is in favor of the lower-level decision maker.

In order to adopt and expand on the iterative decomposition algorithm, we first solve the leader problem and communicate the optimal upper-level decisions to the follower and obtain associated lower bound. Given the optimal leader decisions, we solve the subproblems. A second subproblem is generally necessary when the first subproblem has multiple solutions for given upper-level decisions. This helps us compute an optimal solution which is in favor of the upper-level problem and a smaller upper bound. We then update our discrete set to include the optimal lower-level solutions at a given iteration and expand our leader problem. In a new iteration, we solve the leader problem augmented with reformulated constraints and enumerate for a given optimal integer solution from the previous iteration to obtain new upper-level decisions and a new stronger lower bound (weakly increasing). Solving the problem in this iterative fashion, we anticipate the lower and upper bounds to converge to exact global optimal solution in finite iterations.

## 4.2 Problem Formulation

In this section, we present extensively the mathematical formulation of the problem with the necessary notations listed in the following paragraphs. Generally, a cloud or edge node can offer various types of resources like vCPUs, memory, storage, etc for demand fulfillment. The edge nodes owned by EC platform offers resources (vCPUs) to $K$ services, each requiring a specific bundle of resource type for meeting the network traffic.

### 4.2.1 Follower Problems

Given the EN activation, number of active servers and pricing decisions made by the leader, the followers i.e., the services aim at maximizing the user experience in terms of delay associated in fulfilling the demands by minimizing their delay cost in addition to the resource procurement and placement cost by deciding on optimal service placement, resource sizing and workload allocation.

It's worth noting that the delay penalty cost is a virtual cost that reflects the sensitivity of services towards latency. The actual payment of services only includes the resource procurement and service placement costs. Furthermore, the placement cost of a service at any EN encapsulates the cost of downloading, installation and storing services onto different ENs. Clearly, a service can operate only on an active EN. Thus the goal of service $k$ is to minimize the objective function (4.1).

Consequently, the follower problem for service $k$ can be represented as follows:

$$\min_{x^k, y^k, t^k} \; p_0 y_0^k + \sum_j p_j y_j^k + \sum_j \phi_j^k t_j^k + w^k \left( \sum_i x_{i,0}^k d_{i,0} + \sum_{i,j} x_{i,j}^k d_{i,j} \right) \tag{4.1}$$

subject to

$$p_0 y_0^k + \sum_j p_j y_j^k + \sum_j \phi_j^k t_j^k \leq B^k \tag{4.2}$$

$$t_j^k \leq z_j, \quad \forall j \tag{4.3}$$

$$y_j^k \geq \sum_i x_{i,j}^k, \quad \forall j \tag{4.4}$$

$$y_0^k \geq \sum_i x_{i,0}^k \tag{4.5}$$

$$y_j^k \leq Q q_j t_j^k, \quad \forall j \tag{4.6}$$

$$x_{i,0}^k d_{i,0} + \sum_j x_{i,j}^k d_{i,j} = d_i^{k,\mathsf{a}} R_i^k, \quad \forall i \tag{4.7}$$

$$d_i^{k,\mathsf{a}} \leq D^{k,\mathsf{m}}, \quad \forall i \tag{4.8}$$

$$x_{i,j}^k \leq a_{i,j}^k R_i^k, \quad \forall i, j \tag{4.9}$$

$$\sum_j x_{i,j}^k + x_{i,0}^k = R_i^k, \quad \forall i \tag{4.10}$$

$$x_{i,j}^k, x_{i,0}^k \geq 0; \; y_j^k, y_0^k \geq 0; \; d_i^{k,\mathsf{a}} \geq 0; \; t_j^k \in \{0, \; 1\}, \; \forall i, j, k. \tag{4.11}$$

The budget constraint (4.2) reflects all possible combination of edge resources that can be

bought given the prices are less than the maximum budget of the service. This constraint is useful for utility maximization. Constraint (4.3) indicates that a service can operate only on an active EN. The amount of resource $y_0^k$ and $y_j^k$ purchased from cloud and each EN $j$ should be adequate to meet the resource demand of all requests. This is reflected by constraints (4.4) and (4.5). Equation (4.6) indicates that given the total number of active servers per EN ($q_j$), a service $k$ placed on an edge node $j$ can buy no more than a specific amount $q_j Q$ of resource, where $Q$ is number of vCPUs per server.

For a delay-sensitive service $k$, constraints (4.7) and (4.8) ensure that the average delay in each location $i$ does not exceed a given threshold. Furthermore, every service may have different requirements in terms of hardware systems and software environments to operate on any given edge node. Thus, we use $a_{i,j}^k$ in (4.9) as a binary indicator to denote the EN $j$ that satisfy the prerequisites for serving the requests of service $k$ in area $i$. Equation (4.10) implies that the resource demand be served either by ENs or cloud. Constraints on variables are expressed in (4.11). The workload allocation $x$ and resource procurement $y$ must be non-negative . Also, the service placement $t$ decisions are binary. In addition, although $p$, $q$, and $z$ are variables in the leader problem, they are parameters in the follower problems.

### 4.2.2 Leader Problem

While many issues on the path to edge computing remain unclear, the development of new and creative use cases is easy to envisage. Investment strategy that improves network performance in terms of network parameters like latency, scalability and reliability are major factors for future profits. We are thus interested in maximizing the profit of EC platform by optimizing the edge resource prices that help services reduce cost and improve quality of service. For most operators, this evolutionary approach will be the natural direction that will allow them to reduce investment while EC's incremental revenue potential remains uncertain and is still in its infancy stage. We thus express the objective of the leader as profit maximization problem which is equal to revenue minus cost. The revenue of the infrastructure provider is generated from selling edge resources to services for demand fulfilment i.e., $\sum_j p_j \sum_k y_j^k$.

Furthermore, the cost function of the leader includes cost for operating ENs. For an active EN, its operating cost can then be expressed as the sum of a fixed cost and a variable cost

which directly depends on the amount of computing resources used giving us the following expression:

$$Cost = \sum_j \left[ c_j z_j + A_j \frac{\sum_k y_j^k}{C_j} \right]. \tag{4.12}$$

Finally, the difference between revenue and cost functions can be used to establish the leader's profit function:

$$\mathcal{P} = \sum_j p_j \sum_k y_j^k - \sum_j \left[ c_j z_j + A_j \frac{\sum_k y_j^k}{C_j} \right]. \tag{4.13}$$

Next, we describe the leader problem's constraints. Clearly, the number of active servers on each EN must be less than or equal to its computing capacity of active ENs.

$$q_j \leq C_j z_j, \ \forall j. \tag{4.14}$$

$$q_j \in \mathbb{Z}_+; \quad z_j \in \{0, \ 1\}, \ \forall j. \tag{4.15}$$

Also, it is necessary to ensure that the amount of computing resource allocated should be less than the total available active vCPUs on an EN. Equation (4.15) indicates that the number of active servers $q$ must be a positive integer and EN activation $z$ decisions are binary.

$$\sum_k y_j^k \leq q_j Q, \ \forall j. \tag{4.16}$$

Likewise equation (4.17) ensures that the total storage resource allocated to each EN is constrained by its storage capacity.

$$\sum_k b^k t_j^k \leq S_j q_j, \ \forall j. \tag{4.17}$$

As discussed earlier, we assume a predefined discrete set of unit resource prices at each EN, i.e., we have

$$p_j \in \{p_j^1, \ldots, p_j^V\}, \ \forall j, \tag{4.18}$$

where $v \in \{1, \ldots, V\}$ represent different options for price $p_j^1 < p_j^2 < \ldots < p_j^V$. Let $r_j^v$ be a binary variable which equals one if the resource price at EN $j$ is $p_j^v$. Since the price can take

only one value, we have:

$$p_j = \sum_v p_j^v r_j^v,\ \forall j;\ \sum_v r_j^v = 1,\ \forall j;\ r_j^v \in \{0,1\}, \forall j, v. \tag{4.19}$$

We now present the leader problem, which is indeed a bi-level mixed integer program in the compact form as follows:

$$\textbf{BiMIP}: \max_{p,q,z} \sum_j p_j \sum_k y_j^k - \sum_j \left[ c_j z_j + A_j \frac{\sum_k y_j^k}{C_j} \right] \tag{4.20}$$

subject to

$$(4.14) - (4.19)$$

$$x^k, y^k, t^k \in \underset{(x^k, y^k, t^k) \in \mathcal{S}^k}{\operatorname{argmin}} \left\{ p_0 y_0^k + \sum_j p_j y_j^k + \sum_j \phi_j^k t_j^k \right.$$

$$\left. + w^k \left( \sum_i x_{i,0}^k d_{i,0} + \sum_{i,j} x_{i,j}^k d_{i,j} \right) : (4.2) - (4.11) \right\},\ \forall k, \tag{4.21}$$

where $\mathcal{S}^k$ is the feasible set of $(x^k, y^k, t^k)$ satisfying constraints $(4.2) - (4.11)$ of the follower problem for service $k$. The platform aims to maximize its profit by jointly optimizing the EN activation, number of active servers and resource pricing decisions. The follower problems (i.e., the lower-level problems) serve as constraints of the leader problem, as shown in (4.21). Note $q_j$ and $z_j$ represents the integer variables at the upper-level of the problem and $t_j^k$ represents the lower-level integer variables. Thus we obtain a bi-level mixed integer programming problem.

Furthermore, we reformulate the original bilevel MIP model (4.20)-(4.21) to provide a decomposable structure for algorithm development which can be represented as follows:

$$\textbf{BiMIP}_\textbf{d}: \min_{p,q,z} - \sum_j p_j \sum_k y_j'^k + \sum_j \left[ c_j z_j + A_j \frac{\sum_k y_j'^k}{C_j} \right] \tag{4.22}$$

subject to

$$(4.14) - (4.19)$$

$$-\left[w^k\left(\sum_i x'^k_{i,0} d_{i,0} + \sum_{i,j} x'^k_{i,j} d_{i,j}\right) + p_0 y'^k_0 + \sum_j p_j y'^k_j + \sum_j \phi^k_j t'^k_j\right] \geq$$

$$\max_{(x^k, y^k, t^k) \in \mathcal{S}^k} \left\{ - w^k\left(\sum_i x^k_{i,0} d_{i,0} + \sum_{i,j} x^k_{i,j} d_{i,j}\right) - p_0 y^k_0 - \sum_j p_j y^k_j - \sum_j \phi^k_j t^k_j : \right.$$

$$\left. (4.2) - (4.11) \right\}, \ \forall k. \qquad (4.23)$$

Here $(x'^k, y'^k, t'^k)$ set represents the set of duplicate lower-level decision variables. In simple terms, by duplicating variables (and constraints), a complete variable set is at the control of the upper-level decision maker. Eventually, the upper-level DM would be able to use $(x'^k, y'^k, t'^k)$ to simulate the response of the lower-level DM and assess how that response affects her decision scheme [66].

Also, for each follower problem, we have integer constraints and we need to enumerate over all possible integer variables. This results in a very large MILP problem which motivated us to adopt an iterative decomposition algorithm to iteratively enumerate over all possible integer variables in finite iterations.

## 4.3 Solution Approaches

We now tackle the challenging bi-level programming problem in which the upper-level DM solves an optimization problem which includes another optimization problem (lower-level problem) in its constraints [67]. In this literature, we present an iterative decomposition algorithm for solving the proposed bi-level mixed integer model based on column-and-constraint generation method using reformulations and decomposition [66]. This solution approach is an extension to the already existing CCG algorithm proposed for solving two-stage robust optimization problems in [68]. The CCG algorithm for bi-level MIP follows the same master-subproblem framework as that of [68] with an additional advantage of dealing with challenging problems

that involve discrete variables in the lower-level of the problem. The optimal value of the master problem in each iteration offers a lower bound, while the optimal solution to the subproblem allows us to determine the upper bound of the bi-level MIP. We expect that the lower and upper bounds converge to the exact optimal values by iteratively solving those updated master and subproblems.

Basically, solving the upper-level problem yields solutions $p^*$, $q^*$ and $z^*$ which can be used for solving lower-level problems to derive optimal ($x^{k*}$, $y^{k*}$, $t^{k*}$). As ($p^*$, $q^*$, $z^*$, $x^{k*}$, $y^{k*}$, $t^{k*}$) represents the feasible solution set, we can obtain stronger lower and upper bounds by expanding $t^{k*}$ set and evaluating corresponding optimal value of master problem. Thus solving the augmented master problem leads to new values for $p^*$, $q^*$ and $z^*$ and a better lower bound in every new iteration. For a fixed ($p^*$, $q^*$, $z^*$), optimal solution to the lower-level problem provides upper bound (UB). Solving the master and subproblems in this iterative fashion results in convergence of lower and upper bounds.

We now define the master problem with augmented decision variables and constraints. Then, given the first-stage decisions, we expand on how to solve the subproblems (i.e., the second-stage problems). Finally, in a master-subproblem setting, we present the iterative decomposition algorithm for solving the bi-level MIP for optimal pricing design, resource management and service placement. For reformulation step, first we present KKT-based approach that can easily be converted into a regular MIP using the linearization techniques. Next we present another popular approach based on strong duality theorem of linear programming. We finally elaborate on the strong duality approach due to considerably less number of constraints and integer variables compared to the one obtained from the KKT-based approach.

### 4.3.1 KKT-based Reformulation

We present a single-level equivalent reformulation of **BiMIP** in this subsection which serves as the foundation for our solution scheme. The primary goal of this reformulation is to expand (4.23) by enumeration. We assume that the remaining lower-level problem has a finite optimal value for any possible $(p, q, z, t)$. First, we separate discrete and continuous variables in the

lower-level to restructure the righthand-side of (4.23) as follows:

$$-w^k\left(\sum_i x'^k_{i,0}d_{i,0}+\sum_{i,j}x'^k_{i,j}d_{i,j}\right)-p_0y'^k_0-\sum_j p_jy'^k_j-\sum_j\phi^k_jt'^k_j\geq$$

$$\max_{\mathbf{t}^k\in\mathbf{T}^k}-\sum_j\phi^k_jt^k_j+\max_{x^k,y^k}\Bigg\{-w^k\left(\sum_i x^k_{i,0}d_{i,0}+\sum_{i,j}x^k_{i,j}d_{i,j}\right)$$

$$-p_0y^k_0-\sum_j p_jy^k_j:(4.2)-(4.11),\Bigg\},\ \forall k, \tag{4.24}$$

where $\mathbf{T}^k$ represents the collection of all possible $t^k$ obtained from the lower-levels of the problem. We now apply the classical reformulation method using KKT conditions to the second maximization problem on the right-hand side of (4.24). Thus the complementary slackness, dual feasibility, and the primal feasibility conditions of the follower problems render:

$$0\leq\mu^k_1\perp B^k-\sum_j\phi^k_jt^k_j-p_0y^k_0-\sum_j p_jy^k_j\geq 0,\ \forall k \tag{4.25}$$

$$0\leq\mu^k_2\perp y^k_0-\sum_i x^k_{i,0}\geq 0,\ \forall k \tag{4.26}$$

$$0\leq\Gamma^k_j\perp y^k_j-\sum_i x^k_{i,j}\geq 0,\forall j,k \tag{4.27}$$

$$0\leq\sigma^k_j\perp Qq_jt^k_j-y^k_j\geq 0,\ \forall j,k \tag{4.28}$$

$$0\leq\tau^k_{i,j}\perp a^k_{i,j}R^k_i-x^k_{i,j}\geq 0,\ \forall i,j,k \tag{4.29}$$

$$0\leq y^k_0\perp p_0(1+\mu^k_1)-\mu^k_2\geq 0,\ \forall k \tag{4.30}$$

$$0\leq y^k_j\perp p^k_j\mu^k_1-\Gamma^k_j+\sigma^k_j+p_j\geq 0,\ \forall j,k \tag{4.31}$$

$$0\leq x^k_{i,0}\perp\mu^k_2+d_{i,0}\xi^k_i+\eta^k_i-w^kd_{i,0}\geq 0,\ \forall i,k \tag{4.32}$$

$$0\leq x^k_{i,j}\perp\Gamma^k_j+d_{i,j}\xi^k_i+\tau^k_{i,j}+\eta^k_i-w^kd_{i,j}\geq 0,\ \forall i,j,k. \tag{4.33}$$

Note that $0\leq a\perp b\geq 0$ means $a\geq 0,\ b\geq 0$, and $ab=0$. Constraints (4.25)-(4.33) are called complementarity constraints or equilibrium constraints. Here $\mu^k_1,\mu^k_2,\Gamma^k_j,\sigma^k_j$, and $\tau^k_{i,j}$ are dual variables associated with the lower-level constraints involving continuous variables to be reformulated. By replacing constraints (4.23) for the follower problems with the set of constraints (4.2)–(4.11), and (4.25)–(4.33), the bilevel program (4.22)-(4.23) becomes a mathematical program with complementarity constraints (MPCC). This MPCC problem has

four sources of nonlinearity, including: i) the complementarity constraints (4.25)-(4.33); ii) the bilinear terms $p_j\mu_1^k$ in (4.31); iii) the bilinear term $\sum_{j,k} p_j y_j^k$ in the objective function (4.22); and iv) the bilinear term $\sum_j p_j y_j^k$ in (4.25). To convert the MPCC problem (i.e., an MINLP) into an MILP, we need to linearize these nonlinear terms.

First, the nonlinear complementarity constraints (4.25)–(4.33) can be transformed into equivalent exact linear constraints by using the Fortuny-Amat transformation [61] mentioned earlier in chapter 2. Therefore, the set of constraints (4.25)–(4.33) can be rewritten as:

$$B^k - \sum_j \phi_j^k t_j^k - p_0 y_0^k - \sum_j p_j y_j^k \le v_1^k M_1;\ \mu_1^k \le (1 - v_1^k) M_1, \quad \forall k \tag{4.34}$$

$$y_0^k - \sum_i x_{i,0}^k \le v_2^k M_2;\ \mu_2^k \le (1 - v_2^k) M_2, \quad \forall k \tag{4.35}$$

$$y_j^k - \sum_i x_{i,j}^k \le \kappa_j^k M_3;\ \Gamma_j^k \le (1 - \kappa_j^k) M_3, \quad \forall j,k \tag{4.36}$$

$$Q q_j t_j^k - y_j^k \le \theta_j^k M_4;\ \sigma_j^k \le (1 - \theta_j^k) M_4, \quad \forall j,k \tag{4.37}$$

$$a_{i,j}^k R_i^k - x_{i,j}^k \le \rho_{i,j}^k M_5;\ \tau_{i,j}^k \le (1 - \rho_{i,j}^k) M_5,\ \forall i,j,k \tag{4.38}$$

$$p_0(1 + \mu_1^k) - \mu_2^k \le v_3^k M_6;\ y_0^k \le (1 - v_3^k) M_6,\ \forall k \tag{4.39}$$

$$p_j \mu_1^k - \Gamma_j^k + \sigma_j^k + p_j \le \Phi_j^k M_7;\ y_j^k \le (1 - \Phi_j^k) M_7,\ \forall j,k \tag{4.40}$$

$$\mu_2^k + d_{i,0}\xi_i^k + \eta_i^k - w^k d_{i,0} \le \Omega_{i,0}^k M_8;\ x_{i,0}^k \le (1 - \Omega_{i,0}^k) M_8,\ \forall i,k \tag{4.41}$$

$$\Gamma_j^k + d_{i,j}\xi_i^k + \tau_{i,j}^k + \eta_i^k - w^k d_{i,j} \le \zeta_{i,j}^k M_9;\ x_{i,j}^k \le (1 - \zeta_{i,j}^k) M_9,\ \forall i,j,k \tag{4.42}$$

$$x_{i,0}^k \ge 0;\ x_{i,j}^k \ge 0;\ y_0^k \ge 0;\ y_j^k \ge 0 \tag{4.43}$$

$$\mu_1^k \ge 0;\ \mu_2^k \ge 0;\ \Gamma_j^k \ge 0;\ \sigma_j^k \ge 0;\ \tau_{i,j}^k \ge 0 \tag{4.44}$$

$$\Phi_j^k,\ \kappa_j^k,\ \theta_j^k,\ \rho_{i,j}^k,\ \Omega_{i,0}^k,\ \zeta_{i,j}^k,\ v_1^k,\ v_2^k,\ v_3^k \in \{0,1\}, \tag{4.45}$$

where $M_1$, $M_2$, $M_3$, $M_4$, $M_5$, $M_6$, $M_7$, $M_8$ and $M_9$ are sufficiently large numbers. For the bilinear terms $p_j\mu_1^k$, using (4.19), we can rewrite it as:

$$p_j \mu_1^k = \sum_v p_j^v r_j^v \mu_1^k = \sum_v p_j^v \pi_j^{v,k}, \tag{4.46}$$

where $\pi_j^{v,k} = r_j^v \mu_1^k$. Note that $\pi_j^{v,k}$ is a continuous variable and we have $\pi_j^{v,k} = \mu_1^k$ if $r_j^v = 1$ and $\pi_j^{v,k} = 0$, otherwise. Hence, using (4.46), the bilinear term $p_j\mu_1^k$ can be written as a linear function of $\pi_j^k = (\pi_j^{1,k}, \ldots, \pi_j^{V,k})$. Additionally, the constraints $\pi_j^{v,k} = r_j^v \mu_1^k, \forall j,k$ can be

implemented through the following linear inequalities [62]:

$$\pi_j^{v,k} \leq M r_j^v, \quad \forall j,k,v; \quad \pi_j^{v,k} \leq \mu_1^k, \quad \forall j,k,v \tag{4.47}$$

$$\pi_j^{v,k} \geq 0, \quad \forall j,k,v; \quad \pi_j^{v,k} \geq \mu_1^k + M r_j^v - M, \quad \forall j,k,v, \tag{4.48}$$

where $M$ is a sufficiently large number.

Note that the bilinear terms $\sum_{j,k} p_j y_j^k$ in (4.22) and $\sum_j p_j y_j^k$ in (4.25) are both products of a discrete and continuous variable. By expressing, $p_j$ using (4.19) it becomes similar to what we did for the bilinear terms $r_j^v \mu_1^k$ using (4.47) and (4.48).

Based on the linearization steps described above, we can then express the bi-level problem (4.22)-(4.23) with an equivalent single-level MILP. Due to large number of variables and constraints in the reformulation, we limit on presenting the overall expanded single-level formulation using KKT-based approach and instead use duality-based approach as a platform to point out the whole strategy of algorithm development.

### 4.3.2 Duality-based Reformulation

Instead of using KKT conditions, we can utilize the LP duality to transform the original bilevel problem into an equivalent MILP. As the lower-level problem involves discrete variables, it is indeed challenging to solve. We propose to convert the optimization problem of the services into a single-level comparable reformulation. Similar to what we did in (4.3.1), we derive (4.23) by enumeration. First, we separate discrete and continuous variables in the lower-level as follows:

$$\max_{\mathbf{t}^k \in \mathbf{T}^k} -\sum_j \phi_j^k t_j^k + \max_{x^k, y^k} \; -w^k \left( \sum_i x_{i,0}^k d_{i,0} + \sum_{i,j} x_{i,j}^k d_{i,j} \right) - p_0 y_0^k - \sum_j p_j y_j^k \tag{4.49}$$

subject to:

$$p_0 y_0^k + \sum_j p_j y_j^k \leq B^k - \sum_j \phi_j^k t_j^k, \qquad (\mu_1^k) \tag{4.50}$$

$$\sum_i x_{i,j}^k - y_j^k \leq 0, \quad \forall j \quad (\Gamma_j^k) \tag{4.51}$$

$$\sum_i x_{i,0}^k - y_0^k \leq 0, \qquad (\mu_2^k) \tag{4.52}$$

$$y_j^k \leq Q q_j t_j^k, \quad \forall j \quad (\sigma_j^k) \tag{4.53}$$

$$x_{i,0}^k d_{i,0} + \sum_j x_{i,j}^k d_{i,j} = d_i^{k,\mathsf{a}} R_i^k, \quad \forall i \quad (\xi_i^k) \tag{4.54}$$

$$\sum_j x_{i,j}^k + x_{i,0}^k = R_i^k, \quad \forall i \quad (\eta_i^k) \tag{4.55}$$

$$x_{i,j}^k \leq a_{i,j}^k R_i^k, \quad \forall i,j \quad (\tau_{i,j}^k) \tag{4.56}$$

where $\mu_1^k$, $\Gamma_j^k$, $\mu_2^k$, $\sigma_j^k$, $\xi_i^k$, $\eta_i^k$ and $\tau_{i,j}^k$ are non-negative continuous dual variables. Thus the lower level problem becomes:

$$\begin{aligned} \underset{p,\, q,\, x,\, y,\, \mu_1^k,\, \mu_2^k,\, \Gamma^k,\, \sigma^k,\, \xi^k,\, \eta^k,\, \tau^k}{\text{maximize}} \; & -\sum_j \phi_j^k t_j^k + \mu_1^k \left( B^k - \sum_j \phi_j^k t_j^k \right) + \sum_i R_i^k \xi_i^k \\ & + \sum_i R_i^k \eta_i^k + \sum_j Q q_j t_j^k \sigma_j^k + \sum_{i,j} a_{i,j}^k \tau_{i,j}^k R_i^k \end{aligned} \tag{4.57}$$

subject to

$$p_0 \left(1 + \mu_1^k\right) - \mu_2^k \geq 0, \ \forall k \tag{4.58}$$

$$p_j \mu_1^k - \Gamma_j^k + \sigma_j^k + p_j \geq 0, \ \forall j,k \tag{4.59}$$

$$\mu_2^k + d_{i,0} \xi_i^k + \eta_i^k \geq w^k d_{i,0}, \ \forall i,k \tag{4.60}$$

$$\Gamma_j^k + d_{i,j} \xi_i^k + \tau_{i,j}^k + \eta_i^k \geq w^k d_{i,j}, \ \forall i,j,k. \tag{4.61}$$

We note that this reformulation, as compared to the reformulation based on KKT conditions, has simpler structure, fewer variables and constraints and is usually far less computationally expensive. Indeed, it can be estimated by professional MIP solvers which are conveniently accessible in practice. In the next subsection, utilizing the discussed reformulation strategy, a master-subproblem framework is constructed for algorithm development.

### 4.3.3 Master Problem

At iteration $L$ the master problem (MP) is given by:

$$\mathbf{MP}: \underline{\Theta}^* = \min_{p,q,z,x',y',t'} -\sum_j p_j \sum_k y_j'^k + \sum_j \left[ c_j z_j + A_j \frac{\sum_k y_j'^k}{C_j} \right] \tag{4.62}$$

subject to

$$t_j'^k \le z_j; \quad q_j \le C_j z_j, \tag{4.63}$$

$$\sum_k y_j'^k \le q_j Q; \quad y_j'^k \le Q q_j t_j'^k; \quad \sum_k b^k t_j'^k \le S_j q_j, \tag{4.64}$$

$$p_0 y_0'^k + \sum_j p_j y_j'^k + \sum_j \phi_j^k t_j'^k \le B^k, \tag{4.65}$$

$$y_j'^k \ge \sum_i x_{i,j}'^k; \quad y_0'^k \ge \sum_i x_{i,0}'^k, \tag{4.66}$$

$$x_{i,j}'^k \le a_{i,j}^k R_i^k; \quad \sum_j x_{i,j}'^k + x_{i,0}'^k = R_i^k, \tag{4.67}$$

$$x_{i,0}'^k d_{i,0} + \sum_j x_{i,j}'^k d_{i,j} = d_i'^{k,\mathsf{a}} R_i^k; \quad d_i'^{k,\mathsf{a}} \le D^{k,\mathsf{m}}, \tag{4.68}$$

$$\begin{aligned} w^k \left( \sum_i x_{i,0}'^k d_{i,0} + \sum_{i,j} x_{i,j}'^k d_{i,j} \right) + p_0 y_0'^k + \sum_j p_j y_j'^k + \sum_j \phi_j^k t_j'^k \ge -\sum_j \phi_j^k t_j^{k,l} + \\ \mu_1^{k,l} \left( B^k - \sum_j \phi_j^k t_j^{k,l} \right) + \sum_i R_i^k \xi_i^{k,l} + \sum_i R_i^k \eta_i^{k,l} + \sum_j Q q_j t_j^{k,l} \sigma_j^{k,l} \\ + \sum_{i,j} a_{i,j}^k \tau_{i,j}^{k,l} R_i^k, \quad 1 \le l \le L \end{aligned} \tag{4.69}$$

$$p_0 \left( 1 + \mu_1^{k,l} \right) - \mu_2^{k,l} \ge 0, \quad 1 \le l \le L \tag{4.70}$$

$$p_j \mu_1^{k,l} - \Gamma_j^{k,l} + \sigma_j^{k,l} + p_j \ge 0, \quad 1 \le l \le L \tag{4.71}$$

$$\mu_2^{k,l} + d_{i,0} \xi_i^{k,l} + \eta_i^{k,l} \ge w^k d_{i,0}, \quad 1 \le l \le L \tag{4.72}$$

$$\Gamma_j^{k,l} + d_{i,j} \xi_i^{k,l} + \tau_{i,j}^{k,l} + \eta_i^{k,l} \ge w^k d_{i,j}, \quad 1 \le l \le L \tag{4.73}$$

$$\begin{aligned} x_{i,j}'^k, x_{i,0}'^k \ge 0; \; y_j'^k, y_0'^k \ge 0; \; d_i'^{k,\mathsf{a}} \ge 0; \; \mu_1^{k,l}, \Gamma_j^{k,l}, \mu_2^{k,l}, \sigma_j^{k,l}, \xi_i^{k,l}, \eta_i^{k,l}, \tau_{i,j}^{k,l} \ge 0 \\ z_j, t_j'^k \in \{0,1\}; \quad (4.19). \end{aligned} \tag{4.74}$$

where duplicated lower-level variables (and constraints) are denoted in the upper level by $x'^k$, $y'^k$, and $t'^k$ to provide a complete variable set which is under the control of the leader. Since

each new iteration adds more constraints to the MP, we obtain a stronger lower bound. Thus:

$$LB = \Theta^* \tag{4.75}$$

Note that the bilinear terms $q_j t'^k_j$ in (4.64) can be replaced with a new integer variable $g^k_j$ and the constraints $g^k_j = q_j t'^k_j$ can be implemented through the following linear inequalities:

$$g^k_j \leq C_j t'^k_j, \quad \forall j,k; \quad g^k_j \leq q_j, \quad \forall j,k \tag{4.76}$$

$$g^k_j \geq q_j - (1 - t'^k_j)C_j, \quad \forall j,k; \quad g^k_j \geq 0, \quad \forall j,k, \tag{4.77}$$

where $q_j$ is upper bounded by $C_j$ and lower bounded by 0. Additionally, the bilinear terms $q_j \sigma^{k,l}_j$ is the product of an integer and continuous variable. To implement this, first step is to perform binary expansion of the integer variable. This way we convert integer variable to binary variable and further perform linearization of binary and continuous variables. Following the same steps from [62] we enforce the binary expansion on $q_j$ to obtain a linear MIP. Since $q_j$ is upper bounded by $C_j$, we can rewrite $q_j$ as follows:

$$q_j = \sum_{e \in E_j} 2^{e-1} D_{j,e}, \tag{4.78}$$

where $D_{j,e} \in \{0,1\}$ and $E_j = \{1,2,...,\lfloor \log_2 C_j \rfloor + 1\}, \forall j$. Thus we can rewrite the nonlinear term as follows:

$$q_j \sigma^{k,l}_j = \sum_{e \in E_j} 2^{e-1} D_{j,e} \sigma^{k,l}_j \tag{4.79}$$

Note that the bilinear terms $D_{j,e}\sigma^{k,l}_j$ is a product of a continuous variable and a binary variable which we can linearize similar to what we did for bilinear terms $r^v_j \mu^k_1$ using (4.47) and (4.48).

### 4.3.4 Subproblems

We formulate and evaluate the following subproblems, SP1 for a given upper-level resource selling price decision. The second subproblems, i.e. SP2, are generally necessary because the lower-level problems may have multiple optimal solutions for SP1 and doing so derives the one in support of the upper-level problem (i.e., valid inequalities). Thus we denote the two

subproblems as follows:

$$\mathbf{SP1}: \varphi^k(\mathbf{p}^*, \mathbf{q}^*, \mathbf{z}^*) = \max_{x^k, y^k, t^k} - w^k \left( \sum_i x^k_{i,0} d_{i,0} + \sum_{i,j} x^k_{i,j} d_{i,j} \right) - p_0 y^k_0 - \sum_j p_j y^k_j - \sum_j \phi^k_j t^k_j \quad (4.80)$$

subject to :

$$(4.2) - (4.11)$$

$$\mathbf{SP2}: \Theta_o(\mathbf{p}^*, \mathbf{q}^*, \mathbf{z}^*) = \min_{x^k, y^k, t^k} - \sum_j \sum_k p_j y^k_j + \sum_j \left[ c_j z_j + A_j \frac{\sum_k y^k_j}{C_j} \right] \quad (4.81)$$

subject to :

$$(4.2) - (4.11)$$
$$-w^k \left( \sum_i x^k_{i,0} d_{i,0} + \sum_{i,j} x^k_{i,j} d_{i,j} \right) - p_0 y^k_0 - \sum_j p_j y^k_j$$
$$- \sum_j \phi^k_j t^k_j \geq \varphi^k(\mathbf{p}^*, \mathbf{q}^*, \mathbf{z}^*), \ \forall k. \quad (4.82)$$

As integer variables are involved at both levels, they can be easily computed using existing MIP solvers such as Gurobi and Mosek. Also we can express the upper bound as follows:

$$UB = \min \left\{ UB, \Theta_o(\mathbf{p}^*, \mathbf{q}^*, \mathbf{z}^*) \right\} \quad (4.83)$$

### 4.3.5 Iterative Algorithm and Proof of Convergence

Basically solving the upper problem gives optimal solution set $(\mathbf{p}^*, \mathbf{q}^*, \mathbf{z}^*)$. Using this we solve subproblems for obtaining optimal $(\mathbf{x}^{k*}, \mathbf{y}^{k*}, \mathbf{t}^{k*})$. As $(\mathbf{p}^*, \mathbf{q}^*, \mathbf{z}^*, \mathbf{x}^{k*}, \mathbf{y}^{k*}, \mathbf{t}^{k*})$ is a feasible solution set, its value , $\Theta_o(\mathbf{p}^*, \mathbf{q}^*, \mathbf{z}^*)$, provides an upper bound (non-increasing) to the problem. Then we update the $\mathbf{T}^k$ set by including $\mathbf{t}^{k*}$ and expand our master problem $\underline{\Theta}$. Solving the augmented master problem leads to new $\mathbf{p}^*$, $\mathbf{q}^*$, and $\mathbf{z}^*$ values, as well as a stronger lower bound (weakly-increasing) for the new iteration. By iteratively solving the master and subproblems, we anticipate the lower and upper bounds to converge to the optimal value. Using **Algorithm 1**, we can derive optimal resources prices, service placement, workload allocation and resource procurement decisions.

**Algorithm 1** Iterative Decomposition Algorithm

1: Initialization: set $L = 0$, $LB = -\infty$, and $UB = +\infty$.

2: Solve the master problem in (4.3.3). Derive optimal solutions ($p^*$, $q^*$, $z^*$, $x'^*$, $y'^*$, $t'^*$, $x^{1*}$,...,$x^{L*}$, $y^{1*}$,...,$y^{L*}$, $t^{1*}$,...,$t^{L*}$, $\Gamma^{1*}$,...,$\Gamma^{L*}$, $\mu^{1*}$,...,$\mu^{L*}$, $\sigma^{1*}$,...,$\sigma^{L*}$, $\xi^{1*}$,...,$\xi^{L*}$, $\eta^{1*}$,...,$\eta^{L*}$, $\tau^{1*}$,...,$\tau^{L*}$) and update $LB = \underline{\Theta}^*$.

3: If $\frac{UB-LB}{UB} \leq \epsilon$, return the associated $UB$ and terminate. Otherwise, go to step 4.

4: Solve **SP1** for given optimal solutions of step 2 and obtain $\varphi^k(\mathbf{p}^*, \mathbf{q}^*, \mathbf{z}^*)$.

5: Then solve **SP2**. Report optimal ($x^{k*}$,$y^{k*}$,$t^{k*}$) and $\Theta_o(\mathbf{p}^*, \mathbf{q}^*, \mathbf{z}^*)$. Update $UB$ following (4.83).

6: Add cuts (4.69)-(4.73) to the **MP**. Set $L = L + 1$ and go to Step 2.

**Proposition 4.3.1** *Let $\epsilon = 0$ and assume that integer set $\boldsymbol{T}$ is finite. The presented iterative decomposition algorithm converges to the optimal value of $\boldsymbol{BiMIP_d}$ within $\mathcal{O}(|T|)$ iterations.*

**Proof:**

This can be shown by contradiction that any repeated $\mathbf{t}^{k*}$ implies $LB = UB$. Specifically, assume that the current iteration index is $L_1$, ($\mathbf{p}^*$, $\mathbf{q}^*$, $\mathbf{z}^*$, $\mathbf{x}'^{k*}$, $\mathbf{y}'^{k*}$, $\mathbf{t}'^{k*}$) is obtained in Step 2 with $LB \leq UB$, and ($\mathbf{x}^{k*}$, $\mathbf{y}^{k*}$, $\mathbf{t}^{k*}$) is obtained from Step 4. From step 5 of **Algorithm 1**, we have: $UB \leq \Theta_o(\mathbf{p}^*, \mathbf{q}^*, \mathbf{z}^*) = -\sum_j p^* \sum_k y_j'^{k*} + \sum_j \left[ c_j z_j^* + A_j \frac{\sum_k y_j'^{k*}}{C_j} \right]$ and optimal ($\mathbf{x}^{k*}$, $\mathbf{y}^{k*}$, $\mathbf{t}^{k*}$) that is in favor of the upper-level decision maker.

We further assume that in some previous iteration $L_0 (< L_1)$, $\mathbf{t}^{k*}$ was also derived. Since $\frac{UB-LB}{UB} > \epsilon$, we augment the **MP** with a set of new variables and constraints associated with $\mathbf{t}^{k*}$ (=$\mathbf{t}^{k,L_1+1}$), $\forall k$. However, since those variables and constraints are the same as those created and included in iteration $L_0$, the increase does not essentially change **MP**. Thus in iteration $L_1 + 1$ it yields the same optimal value as iteration $L_1$. Therefore, when the algorithm goes from iteration $L_1$ to iteration $L_1 + 1$, LB does not change. We have: $LB \geq -\sum_j p^* \sum_k y_j'^{k*} + \sum_j \left[ c_j z_j^* + A_j \frac{\sum_k y_j'^{k*}}{C_j} \right]$.

As $\mathbf{t}^{k,L_1+1}$ is optimal to $\varphi^k(\mathbf{p}^*, \mathbf{q}^*, \mathbf{z}^*)$, $\forall k$ and constraints from strong duality ensures that (at iteration $L_1 + 1$ ), $-\sum_j \phi_j^k t_j^{k,L_1+1} + \mu_1^{k,L_1+1} \left( B^k - \sum_j \phi_j^k t_j^{k,L_1+1} \right) + \sum_i R_i^k \xi_i^{k,L_1+1} + \sum_i R_i^k \eta_i^{k,L_1+1} + \sum_j Q q_j t_j^{k,L_1+1} \sigma_j^{k,L_1+1} + \sum_{i,j} a_{i,j}^k \tau_{i,j}^{k,L_1+1} R_i^k = \varphi^k(\mathbf{p}^*, \mathbf{q}^*, \mathbf{z}^*)$, $\forall k$. Then, it's easy to see from Step 3 that LB $\geq$ UB, which typically converges the algorithm in a finite

number of iterations. □

Note that the algorithm's actual implementation is independent of the cardinality of **T** set, which could be infinite. The proposed algorithm will converge to an optimal solution over finite iterations if a finite optimal solution exists. The algorithm produces an optimal solution in a small number of iterations, which may be significantly less than the cardinality of **T** [66].

## 4.4 Summary

In this chapter, we studied the problem of joint service placement and pricing in EC and designed an iterative computing scheme based on reformulations and decomposition strategies to solve the resulting challenging bi-level mixed integer programming problem. Our proposed model not only maximized the profit of EC platform but also helped each service find their desirable edge nodes for placement and resource procurement to minimize its cost while enhancing user experience. As a result, the solution to underlying problem proves economical, working in favor of both EC platform and services to minimize their costs and improve edge resource utilization. It is worth noting that the solution approach based on column-and-constraint generation method used in this algorithm can reduce the impact of enumeration and greatly improve the overall computational efficiency.

# Chapter 5

# Conclusions and Future Work

By bringing cloud resources closer to the network edge, EC is beginning to reshape the future landscape of a number of industries, resulting in increased innovation that promises to deliver enhanced user experience and enable a wide array of IoT applications. EC is transforming the network edge into an intelligent platform that benefits largely from local data processing and analytics, distributed caching, localization, resource pooling and scaling, enhanced privacy and security, and reliable connectivity by distributing storage, computing, control, and networking functions closer to end-users and data sources. Additionally, it is the key to satisfying the stringent requirements of exciting new systems and low-latency applications such as virtual/augmented reality (VR/AR), embedded artificial intelligence, autonomous driving, manufacturing automation, and tactile Internet. Like any other emerging field, this technology is in its infancy stage with many challenging questions to be addressed, opening a plethora of possibilities for academic researchers [5].

In this thesis, we focus on the joint resource pricing, service placement, EN activation, and workload allocation problems in EC. Specifically, we first developed two efficient approaches based on the KKT conditions and LP duality, respectively, to optimally solve the proposed novel bi-level optimization model which not only maximizes the profit of EC platform but also minimizes the cost of every service. We further studied the same problem statement but now having the services to pay for the service placement costs which results in integer variables in lower-levels of the problem. We then propose to solve this bi-level mixed integer programming (MIP) problem using a novel computing scheme based on reformulations and decomposition strategy for allocating resources of heterogeneous capacity-limited ENs to multiple competing services at the network edge in a fair and efficient manner [66].

Throughout this thesis, we leveraged techniques from various fields to develop advanced algorithms addressing different pricing, service placement and resource allocation challenges.

The effectiveness of the developed algorithm was illustrated through extensive numerical studies and theoretical analysis. For the case with continuous variables in the follower problems, we leveraged the single-level reformulation approaches in addition to the linearization techniques. Our recommendation is to use the duality-based reformulation ahead of the KKT-based solution, for it takes lesser computational time due to smaller size of MILP obtained. However, in our extended model, the follower problems had integer variables and this inspired us to explore the column-and-constraint method from adaptive robust optimization literature to solve the challenging BiMIP problem that guarantees convergence in finite iterations.

The proposed models and algorithms can help shed a light on how EC platform and services can optimize their operation and planning strategies. Furthermore, they can explore the benefits of adopting a dynamic pricing policy that strategically sets prices to drive the demand for resources and utilize the unused capacity to maximize the revenue of decision-makers at both levels.

## 5.1 Future Research Directions

In this section, we present some possible directions of future research.

- In our formulation, we have only considered one type of edge resource (vCPUs). We would like to incorporate virtual network functions (VNFs), multiple resource types and multi-period pricing and workload allocation, to our existing model. In addition, we plan to extend the proposed model to account for various system uncertainties such as resource demand and component failures to build a resilient network.

- Our current bilevel model is of the type, single-leader-multi-follower, basically having only one edge infrastructure provider interacting with multiple services. We would like to deal with the case of multiple competing EC platforms and services. This way the services can find the best-in-class providers. Additionally, as the number of edge infrastructure providers grows, a competitive market is set that aims at providing the best price for various resource capacities, while attracting most services. This also gives more freedom to services for selecting between the providers in the event of power outage or natural disaster alongside securing best prices that cater to their specific needs.

- How to further improve the computational efficiency of the proposed algorithms is another interesting direction.

# Bibliography


[1] E. Ahmed and M. H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," *Future Generation Computer Systems*, vol. 70, pp. 59–63, 2017.

[2] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[3] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[4] https://www.ibm.com/cloud/edge-computing.

[5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[6] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[7] D. T. Nguyen, L. B. Le, and V. Bhargava, "Price-based resource allocation for edge computing: A market equilibrium approach," *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 302–317, 2021.

[8] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[9] W. Sun, J. Liu, and Y. Yue, "Ai-enhanced offloading in edge computing: When machine learning meets industrial iot," *IEEE Network*, vol. 33, no. 5, pp. 68–74, 2019.

[10] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[11] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an intelligent edge: Wireless communication meets machine learning," *IEEE Communications Magazine*, vol. 58, no. 1, pp. 19–25, 2020.

[12] https://about.att.com/story/2020/google_cloud.html.

[13] https://www.datacenterknowledge.com/edge-computing/why-google-cloud-and-att-may-merge-their-telco-edges.

[14] https://aws.amazon.com/wavelength/.

[15] https://www.lightreading.com/the-edge/verizon-vodafone-embrace-amazons-aws-for-edge-computing-with-5g/d/d-id/756060.

[16] https://news.microsoft.com/2019/11/26/att-integrating-5g-with-microsoft-cloud-to-enable-next-generation-solutions-on-the-edge/.

[17] https://www.opennetworking.org/cord/.

[18] https://www.lfedge.org/projects/akraino/.

[19] D. V. Kalashnikov, S. Dempe, T. Matis, J.-F. Camacho-Vallejo, and S. Kavun, "Bilevel programming, equilibrium, and combinatorial problems with applications to engineering 2016," *Mathematical Problems in Engineering*, vol. 2016, pp. 1–3, 01 2016.

[20] https://www.equinix.com.

[21] A. Sinha, P. Malo, and K. Deb, "A review on bilevel optimization: from classical to evolutionary approaches and applications," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 276–295, 2018.

[22] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining Stackelberg game and matching," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1204–1215, 2017.

[23] S. Gabriel, A. J. Conejo, B. Hobbs, D. Fuller, and C. Ruiz, "Complementarity modeling in energy markets," *Springer*, 2012.

[24] D. T. Nguyen, L. B. Le, and V. K. Bhargava, "A market-based framework for multi-resource allocation in fog computing," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1151–1164, 2019.

[25] D. T. Nguyen, L. B. Le, and V. Bhargava, "Edge computing resource procurement: An online optimization approach," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pp. 807–812, 2018.

[26] S. Li, N. Zhang, S. Lin, L. Kong, A. Katangur, M. K. Khan, M. Ni, and G. Zhu, "Joint admission control and resource allocation in edge computing for internet of things," *IEEE Network*, vol. 32, no. 1, pp. 72–79, 2018.

[27] D. T. Nguyen, L. B. Le, and V. K. Bhargava, "Two-stage robust edge service placement and sizing under demand uncertainty," *IEEE Internet of Things Journal*, to be published.

[28] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 725–737, 2017.

[29] L. Zhao, W. Sun, Y. Shi, and J. Liu, "Optimal placement of cloudlets for access delay minimization in sdn-based internet of things networks," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1334–1344, 2018.

[30] S. Yang, F. Li, M. Shen, X. Chen, X. Fu, and Y. Wang, "Cloudlet placement and task allocation in mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5853–5863, 2019.

[31] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171–1181, 2016.

[32] L. Yang, J. Cao, G. Liang, and X. Han, "Cost aware service placement and load dispatching in mobile cloud systems," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1440–1452, 2016.

[33] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Transactions on Computers*, vol. 65, no. 12, pp. 3702–3712, 2016.

[34] R. Yu, G. Xue, and X. Zhang, "Provisioning QoS-aware and robust applications in Internet of things: A network perspective," *IEEE/ACM Transactions on Networking*, vol. 27, no. 5, pp. 1931–1944, 2019.

[35] S. Pasteris, S. Wang, M. Herbster, and T. He, "Service placement with provable guarantees in heterogeneous edge computing systems," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 514–522, 2019.

[36] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing-based IoT," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2146–2153, 2018.

[37] V. Farhadi, F. Mehmeti, T. He, T. L. Porta, H. Khamfroush, S. Wang, and K. S. Chan, "Service placement and request scheduling for data-intensive applications in edge clouds," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 1279–1287, 2019.

[38] A. Yousefpour, A. Patil, G. Ishigaki, I. Kim, X. Wang, H. C. Cankaya, Q. Zhang, W. Xie, and J. P. Jue, "FogPlan: A lightweight QoS-aware dynamic fog service provisioning framework," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5080–5096, 2019.

[39] B. Gao, Z. Zhou, F. Liu, and F. Xu, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 1459–1467, 2019.

[40] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, "Joint service placement and request routing in multi-cell mobile edge computing networks," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 10–18, 2019.

[41] T. Ouyang, R. Li, X. Chen, Z. Zhou, and X. Tang, "Adaptive user-managed service placement for mobile edge computing: An online learning approach," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 1468–1476, 2019.

[42] L. Wang, L. Jiao, T. He, J. Li, and M. Mühlhäuser, "Service entity placement for social virtual reality applications in edge computing," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pp. 468–476, 2018.

[43] H. Xu and B. Li, "Dynamic cloud pricing for revenue maximization," *IEEE Transactions on Cloud Computing*, vol. 1, no. 2, pp. 158–171, 2013.

[44] J. He, Y. Wen, J. Huang, and D. Wu, "On the cost–qoe tradeoff for cloud-based video streaming under amazon ec2's pricing models," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 669–680, 2014.

[45] J. Zhao, H. Li, C. Wu, Z. Li, Z. Zhang, and F. C. M. Lau, "Dynamic pricing and profit maximization for the cloud with geo-distributed data centers," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 118–126, 2014.

[46] B. Baek, J. Lee, Y. Peng, and S. Park, "Three dynamic pricing schemes for resource allocation of edge computing for iot environment," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4292–4303, 2020.

[47] P. Cong, L. Li, J. Zhou, K. Cao, T. Wei, M. Chen, and S. Hu, "Developing user perceived value based pricing models for cloud markets," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 12, pp. 2742–2756, 2018.

[48] F. Hao, D.-S. Park, J. Kang, and G. Min, "2l-mc3: A two-layer multi-community-cloud/cloudlet social collaborative paradigm for mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4764–4773, 2019.

[49] M. Liu and Y. Liu, "Price-based distributed offloading for mobile-edge computing with computation capacity constraints," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 420–423, 2018.

[50] P.-Q. Huang, Y. Wang, K. Wang, and Z.-Z. Liu, "A bilevel optimization approach for joint offloading decision and resource allocation in cooperative mobile edge computing," *IEEE Transactions on Cybernetics*, vol. 50, no. 10, pp. 4228–4241, 2020.

[51] Z. Xiong, J. Kang, D. Niyato, P. Wang, and H. V. Poor, "Cloud/edge computing service management in blockchain networks: Multi-leader multi-follower game-based admm for pricing," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 356–367, 2020.

[52] Z.-L. Chang, C.-Y. Wang, and H.-Y. Wei, "Flat-rate pricing and truthful offloading mechanism in multi-layer edge computing," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021.

[53] P.-Q. Huang and Y. Wang, "A framework for scalable bilevel optimization: Identifying and utilizing the interactions between upper-level and lower-level variables," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 6, pp. 1150–1163, 2020.

[54] K. Liu, X. Qiu, W. Chen, X. Chen, and Z. Zheng, "Optimal pricing mechanism for data market in blockchain-enhanced internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9748–9761, 2019.

[55] J. Yan, S. Bi, L. Duan, and Y.-J. A. Zhang, "Pricing-driven service caching and task offloading in mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4495–4512, 2021.

[56] S. Rivoire, P. Ranganathan, and C. Kozyrakis, "A comparison of high-level full-system power models," in *Proceedings of the 2008 Conference on Power Aware Computing and Systems*, HotPower'08, (USA), pp. 3–7, USENIX Association, 2008.

[57] X. Sun and N. Ansari, "Green cloudlet network: a sustainable platform for mobile cloud computing," *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 180–192, 2020.

[58] G. B. Dantzig, *Linear Programming and Extensions.* Princeton, NJ, USA: Princeton University Press, 1963.

[59] S. I. Gass and M. C. Fu, eds., *Karush-Kuhn-Tucker (KKT) Conditions*, pp. 833–834. Boston, MA: Springer US, 2013.

[60] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[61] J. Fortuny-Amat and B. McCarl, "A representation and economic interpretation of a two-level programming problem," *The Journal of the Operational Research Society*, vol. 32, no. 9, pp. 783–792, 1981.

[62] M. H. Zare, J. S. Borrero, B. Zeng, and O. A. Prokopyev, "A note on linearized reformulations for a class of bilevel linear integer problems," *Ann. Oper. Res.*, vol. 272, pp. 99–117, 2019.

[63] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*, vol. 196 of *International Series in Operations Research  Management Science*. Springer US, 2014.

[64] R. Albert, H. Jeong, and A. L. Barabasi, "Internet: diameter of the world–wide web," *Nature*, vol. 401, no. 6749, pp. 130–131, 1999.

[65] https://aws.amazon.com/ec2/pricing/.

[66] B. Zeng and Y. An, "Solving bilevel mixed integer program by reformulations and decomposition," tech. rep., University of South Florida, 2014.

[67] S. Avraamidou, N. A. Diangelakis, and E. Pistikopoulos, "Mixed integer bilevel optimization through multi-parametric programming," 2016.

[68] B. Zeng and L. Zhao, "Solving two-stage robust optimization problems using a column-and-constraint generation method," *Operations Research Letters*, vol. 41, no. 5, pp. 457 – 461, 2013.