Functional Split in 5G Cloud Radio Access Network using Particle Swarm Optimization

Mr. Wai Phyo

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Electrical Engineering
Department of Electrical Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2022
Copyright of Chulalongkorn University

การแบ่งฟังก์ชันในโครงข่ายการเข้าถึงวิทยุแบบคลาวด์ 5G โดยใช้การหาค่าเหมาะกลุ่มอนุภาค

นายเว่ย โผ

Thesis Title	Functional Split in 5G Cloud Radio Access Network using Particle Swarm Optimization
By	Mr. Wai Phyo
Field of Study	Electrical Engineering
Thesis Advisor	Associate Professor LUNCHAKORN WUTTISITTIKULKIJ, Ph.D.

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Engineering

.... Dean of the FACULTY OF ENGINEERING

(Professor SUPOT TEACHAVORASINSKUN, Ph.D.)

THESIS COMMITTEE

.... Chairman

(Associate Professor CHAIYACHET SAIVICHIT, Ph.D)

.... Thesis Advisor

(Associate Professor LUNCHAKORN WUTTISITTIKULKIJ, Ph.D.)

.... Examiner

(Assistant Professor PASU KAEWPLUNG, Ph.D.)

.... External Examiner

(Pisit Vanichchanunt, Ph.D.)

เว่ย โผ : การแบ่งฟังก์ชันในโครงข่ายการเข้าถึงวิทยุแบบคลาวด์ 5G โดยใช้การหาค่าเหมาะกลุ่มอนุภาค. (Functional Split in 5G Cloud Radio Access Network using Particle Swarm Optimization) อ.ที่ปรึกษาหลัก : Associate Professorลัญฉกร วุฒิสิทธิกุลกิจPh.D.

โครงข่ายเข้าถึงวิทยุคลาวด์ (ซี-แรน) เป็นแนวทางใหม่ที่มีศักยภาพในการลดค่าใช้จ่ายในการติดตั้งและใช้งานเครือข่ายไร้สายได้อย่างมาก การจัดวางฟังก์ชันแรนของซี-แรนเป็นองค์ประกอบที่สำคัญในการลดปริมาณการใช้แบนด์วิดท์และต้นทุนการคำนวณ ในการศึกษานี้มีเป้าหมายหลักคือการลดต้นทุนที่เกี่ยวข้องกับการวางแรนตามหน้าที่ในขณะที่คำนึงถึงค่าใช้จ่ายในการคำนวณและปริมาณการใช้แบนด์วิดท์ของฟรอนต์ฮอลของผู้ใช้ต่างๆ เพื่อให้บรรลุเป้าหมายนี้เราเสนอให้ใช้การหาค่าเหมาะกลุ่มอนุภาค (พีเอสโอ) เพื่อให้การจัดสรรทรัพยากรการคำนวณและแบนด์วิดท์ของฟรอนต์ฮอลมีประสิทธิภาพ เพื่อให้มั่นใจว่าจะได้การออกแบบซี-แรนที่มีประสิทธิภาพและคุ้มค่า ผลการทดลองภายใต้ทราฟฟิกที่แตกต่างกันแสดงให้เห็นว่าพีเอสโอที่เสนอสามารถให้การออกแบบซี-แรนที่ใช้ต้นทุนคุ้มค่าเมื่อเทียบกับผลเฉลยเหมาะสุดที่ได้จากวิธีโปรแกรมเชิงเส้นจำนวนเต็ม

สาขาวิชา	วิศวกรรมไฟฟ้า	ลายมือชื่อนิสิต ....
ปีการศึกษา	2565	ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

6272083321 : MAJOR ELECTRICAL ENGINEERING

KEYWORD: 5G, C-RAN, Functional Split, Integer Linear Programming, Particle Swarm Optimization (PSO)

Wai Phyo : Functional Split in 5G Cloud Radio Access Network using Particle Swarm Optimization. Advisor: Assoc. Prof. LUNCHAKORN WUTTISITTIKULKIJ, Ph.D.

The Cloud Radio Access Network (C-RAN) is an innovative approach that has the potential to significantly reduce the expenses of setting up and operating wireless networks. C-RAN's placement of RAN functions, which strives to reduce bandwidth utilization and computation costs, is a critical component. In this study, our main goal is to reduce the costs associated with functionally placing the RAN while accounting for the computational expense and the front-haul bandwidth usage among various users. To achieve this, we propose to apply Particle Swarm Optimization (PSO) to achieve effective allocation of computational resources and the front-haul bandwidth, ensuring an efficient and cost-effective C-RAN design. Experimental results on different traffic have shown that the proposed PSO can provide cost-effectiveness design of the C-RAN as compared to the optimal solution from the integer linear programing approach.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Field of Study:	Electrical Engineering	Student's Signature	....
Academic Year:	2022	Advisor's Signature	....

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

List of tables

List of figures

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

The annual growth in mobile device traffic expects to continue because of the growing variety of services like Metaverse, Virtual Reality (VR), Augmented Reality (AR), self-driving cars, Internet of Things (IoT), cloud computing, and other cutting-edge technologies has been a significant factor in the exponential increase of mobile device traffic in recent years. As traffic and users increase, we will need to build a cost-effective network, improve the Quality of service, and diversify architecture technologies. So, the fifth generation (5G) will be able to solve such increased traffic and user problems [1]. To meet the enlarged demand, 5G needs to provide adequate scalability. Radio Access Network (RAN) has become a crucial sector of the wireless telecommunication industry and consists of Base stations (BS) and antennae that enable incredible speeds and mobility. To solve the deployment cost and RAN optimization, the operator introduced Cloud Radio Access Network (C-RAN) in 2011 and aims to reduce the RAN deployment costs using the cloud infrastructure capabilities [2] .

A cloud radio access network (C-RAN) is an architecture for radio access networks that is centralized and based on cloud computing and supports 2G, 3G, 4G, and future wireless communication protocols. This approach aims to make a cost-effective RAN design and energy-efficient solution for RAN. This C-RAN architecture helps us reduce the Base Stations' complexity, which shares signal capabilities with multiple antennas [3]. This thesis research aims to optimize the C-RAN functional splits to minimize the cost. RAN optimization seeks to enhance the Quality of service (QoS) performance, and another ambition is to find the optimum set of RAN parameters [4]. It can be achieved by increasing flexibility and lowering the cost of infrastructure deployment by implementing the functional split in RAN.

To reduce costs and optimize bandwidth use, we will propose the Particle Swarm Optimization (PSO) method in this thesis research. Additionally, we will minimize the

cost and bandwidth usage of front-haul by using the already-in-use integer linear programming (ILP) [5]. The final findings regarding bandwidth and computational cost will be compared to ILP and PSO.

1.2 OBJECTIVE

The objective of this thesis aims to optimize computational cost and utilization of bandwidth by focusing on the performance of Cloud RAN (C-RAN) architecture with Particle Swarm Optimization technique (PSO). To compare with my particle swarm optimization approach, we will use existing ILP algorithms. Consequently, the goal is to reduce the cost of RAN functional placement, which includes computational cost and then bandwidth utilization on the Fronthaul across the different user.

1.3 PROBLEM STATEMENT

The Cloud Radio Access Network (C-RAN) is an innovative technology that can reduce the expense of wireless networks. C-RAN, however, also brings different challenges, like optimizing bandwidth utilization and computing cost. Integer linear programming (ILP) is the current state-of-the-art methodology for optimizing computational cost and utilization of bandwidth in C-RAN. ILP techniques, on the other hand, are computationally complex and might be challenging to scale to extensive networks. In this thesis, we will provide a particle swarm optimization (PSO)-based method for minimizing computational cost and bandwidth utilization in C-RAN. PSO is a metaheuristic algorithm that is renowned for having the ability to address effectively challenging optimization issues. The results of this thesis will help develop C-RAN networks that are more effective and affordable.

1.4 SCOPE OF THESIS
The scope of this thesis paper is as follows:

1. To analyze how cloud radio access networks (C-RAN) might optimize bandwidth and computing costs using particle swarm optimization (PSO). The goal is to reduce the RAN operating placement cost, including computational expenses and Fronthaul bandwidth usage across different users.

2. To achieve this, we will compare the performance of the new PSO-based method alongside existing Integer Linear Programming (ILP) methods, which are the most advanced technique for minimizing computational cost and bandwidth usage in C-RAN.

1.5 CONTRIBUTION

This thesis contributes to the field of cloud radio access networks (C-RAN) by exploring the utilization of particle swarm optimization (PSO) for optimizing bandwidth and processing costs. The primary objective is to minimize the RAN functional placement cost, which encompasses computational costs and fronthaul bandwidth utilization across multiple users. To achieve this goal, the performance of a novel PSO-based method was evaluated, revealing its effectiveness in generating good solutions. Through comprehensive evaluations, it was discovered that the PSO-based method exhibits several advantages. Firstly, it has the capability to explore a wider range of potential solutions, allowing for a more thorough search of the optimization space. Secondly, the PSO algorithm demonstrates flexibility in adapting to environmental changes, enabling it to adapt and optimize the system's performance based on varying conditions. The results obtained from the evaluations showcase the convergence of the PSO-based method towards optimal solutions. A notable observation is that as the number of iterations increases, the performance of PSO closely approximates the optimal solution achieved by Integer Linear Programming (ILP). This indicates that PSO can serve as a viable alternative to ILP for a variety of optimization problems, particularly those where the number of iterations is not a limiting factor. Furthermore, the study highlights the importance of fine-tuning the iteration parameter in the deployment processes to achieve desirable cost outcomes. By carefully adjusting the number of iterations, the deployment costs tend to stabilize and converge, providing

insights into optimizing the PSO-based process for reducing the cost of bandwidth and computing in C-RAN. The findings of this research suggest that the PSO-based approach holds significant promise as a new strategy for cost reduction in C-RAN, specifically in terms of bandwidth and computing. The integration of PSO into C-RAN optimization processes provides a fresh perspective and opens avenues for further investigation and improvement in this domain.

In summary, this thesis contributes to the field by demonstrating the efficacy of PSO in optimizing bandwidth and processing costs in C-RAN. The research findings shed light on the benefits of the PSO-based approach, its convergence to optimal solutions. Ultimately, this work paves the way for leveraging PSO as a valuable tool for cost-effective optimization in C-RAN environments.

1.6 OUTLINE OF THESIS

This thesis contains five chapters. The objectives, problem description, range of ideas, and contributions are all described in Chapter 1 of the thesis. Background technologies covered in Chapter 2 include 5G, C-RAN, Functional Split, ILP, and PSO. In Chapter 3, the literature reviews for this thesis project are described. The methods and simulation testing are shown in Chapter 4. Chapter 5 will finally wrap up the thesis research. After Chapter 5, references are also listed.

CHAPTER 2: BACKGROUND

2.1. FIFTH GENERATION (5G)

Modern wireless communication has become progressively more familiar with the development of wireless technologies. Despite 3G and 4G, which have shifted their focus to data and mobile internet service, 2G primarily focused on phone service. The wireless cellular network's initial generation was known as 2G. A maximum data rate of 144 kbps was available when it was first launched in the early 1990s. Despite supporting simple data services, such as text messaging, 2G was primarily used for voice conversations. In terms of wireless cellular networks, 3G was the second generation. With a maximum data rate of 2 Mbps, it was first launched in the late 1990s. Streaming video, email, and web browsing are just a few of the data services that 3G was built to serve. The fourth generation of wireless cellular networks is known as 4G. It was released in the early 2000s and had a 100 Mbps top data rate. High-definition video streaming and online gaming are only two examples of the additional data services that 4G was created to offer.

The fifth generation of wireless technology, or 5G, has become disruptive in the connection and communication industries. How we live, work, and interact with technology will change because 5G promised previously unheard-of speed, low latency, and enormous capacity. Three groups of 5G services can be described, including eMBB, uRLLC, and mMTC. When seeking to meet with a typical end customer, eMBB, which stands for improving mobile broadband, is essential. Additionally, uRLLC stands for ultra-reliable low latency communications, employed in some remote factories and autonomous industries because sub-millisecond latency is typically required. The term "mMTC" refers to massive machine-type communication mainly employed in smart factories [6]. The International Telecommunication Union (TTU) summarized the 5G specifications by using case in Figure 1. The 5G network's internet rates are among its most impressive characteristics. 5G provides blazing-fast downloads, seamless streaming of high-definition video, and immersive virtual reality

experiences with peak download speeds of up to 10 gigabits per second (Gbps). This astounding speed creates many opportunities, from increasing telemedicine and remote collaboration to realizing the full potential of cutting-edge technologies like autonomous vehicles and augmented reality. Low latency is also another feature that sets 5G apart. Latency is the time elapsed between delivering a command and receiving a response. As little as one millisecond (ms) of ultra-low latency is available on 5G networks, enabling real-time interactions and applications that need immediate responsiveness. This ultra-low latency is essential for applications requiring split-second judgments and actions, such as driverless vehicles, industrial automation, and remote surgeries. 5 G's enormous capacity allows it to accommodate an unprecedented number of connected devices simultaneously.

In the Internet of Things (IoT) era, when billions of devices are connected and exchanging data, this feature is crucial. 5G allows seamless connectivity for various IoT applications, from smart cities and homes to industrial automation and agriculture. Additionally, 5G improves wireless networks' security and reliability. It includes strong encryption, authentication, and privacy methods to protect data transfer. This increased security is crucial as more vital infrastructure and services relieve wireless networks to function. 5G technology aims to provide wider bandwidth, broader coverage, and higher-speed internet service with high throughput. Improvement in the advanced data coding/modulation and smart antennas are available at 100Mbps (full mobility) and almost 1Gbps (low mobility) [5, 6] . In addition to high-speed internet service, it also provides users with the following advantages and features:

- Multiple data transfer paths concurrently
- More secure and better QoS (Quality of Service)
- Flexible architecture and low power consumption
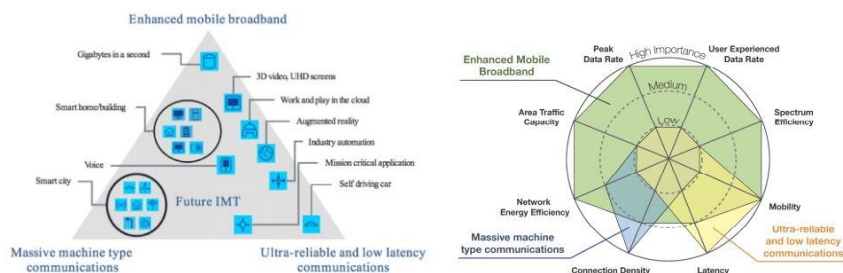
Figure: 1 5G Use Case and Capabilities for different use cases[2, 3]

These are the 5G's benefits and drawbacks:

Pros:

Faster speeds: With theoretical speeds of up to 20 Gbps, 5G is far quicker than previous generation networks. As a result, you can stream music, play games, and download movies considerably more quickly than you could with 4G.

Lower latency: The time it takes for data to go between your device and the network and back is known as latency. Since 5G has lower latency than 4G, your gadgets will respond more quickly.

Greater capacity: Compared to 4G, 5G can accommodate more devices. This implies that there will be less network congestion and a lower probability of slowdowns.

Higher bandwidth: Compared to 4G, 5G has higher bandwidth. This implies that you can simultaneously download and upload more data.

Transformational Applications: 5G technology opens new avenues for creative programs and services. It makes advancing technologies like telemedicine, autonomous vehicles, virtual and augmented reality, innovative infrastructure, and precision agriculture possible.

Network Slicing: The 5G standard adds the idea of network slicing, enabling network operators to build fictitious, separate networks suited to specific use cases. For sectors like healthcare, manufacturing, and transportation, this offers tailored services with particular performance characteristics.

Cons:

Limited Coverage: Because 5G network deployment is ongoing, coverage may need to improve in some places. Compared to urban areas, access to 5G infrastructure may be delayed in rural and isolated areas.

Device Compatibility: Users must have compatible devices to utilize 5G fully. Smartphones and other devices with 5G capabilities may be limited initially, and upgrading older machines can be expensive.

Expensive: 5G equipment and plans cost more than 4G equivalents. However, prices are anticipated to decline when 5G becomes more commonplace.

Overall, 5G technology is anticipated to change several industries, including manufacturing, health care, transport, and entertainment. It will allow innovative services and applications to take advantage of expanded capacity, dependable connectivity, quicker speeds, and lower latency.

2.2 OVERVIEW OF RADIO ACCESS NETWORK (RAN)

One of the significant parts of the network that manages radio communication is the Radio Access Network (RAN). To create a network connection within the mobile web, the User Equipment (UE) must first establish a connection to the radio access network (RAN). User Equipment (UE) and Core Network (CN) can connect with users directly over radio waves through the Radio Access Network (RAN), which also serves as a gateway. We have employed conventional as a component of RAN in 2G, 3G, and 4G cellular systems. The Baseband Processing Unit (BBU) and Radio Unit (RU), located at the site, comprise the traditional RAN's two primary nodes [6].

Traditional RAN characteristic:

- Each Base Station can connect to the fixed number of antennas elements

- Antenna can cover small areas and can handle transmission and reception within the area.
- Capacity is affected by such parameters that means capacity is limited by interference.

These are the challenges in the traditional RAN system:

- We require a larger number of base stations.
- Requires initial investment, site support system, site management and rental, etc.,
- The base station utilization rate is low.
- Baseband Processing Unit
- Faster data service upgrade network

2.2.1 DISTRIBUTED RADIO ACCESS NETWORK (D-RAN)

The baseband processing units (BBUs) from the base stations are distributed to a central location by a radio access network design called a distributed radio access network (D-RAN). This enables better performance and the more effective use of resources. The BBUs are situated at the base stations in a conventional RAN architecture. Because of this, every base station has a unique BBU. Since not all base stations are constantly in use, this may not be resource efficient.

Furthermore, the separation between the base stations and the central office may result in delays, impairing performance. The BBUs are relocated to a central place by D-RAN to overcome these problems. Resources can be used more effectively since several base stations can share the BBUs. Performance may also be enhanced by shortening the distance between the base stations and the central office. A crucial technology for 5G networks is D-RAN. Compared to 4G networks, 5G networks are anticipated to require substantially more bandwidth and capacity. D-RAN's improved performance and resource-use

efficiency can assist in meeting these demands[7]. The following are a few advantages of utilizing D-RAN:

Performance improvement: D-RAN can increase performance by supplying additional bandwidth and lowering latency. Resource sharing across different base stations allows D-RAN to make better use of its resources.

Cost savings: D-RAN can save money by minimizing the number of BBUs that must be deployed.

D-RAN is a promising technology that could raise the effectiveness and performance of 5G networks. Figure 2 illustrates the D- RAN architecture. The Fronthaul interface is the link between BBU and RRU while the connection between BBU and Core Network is also known as Backhaul. Although the D-RAN architecture catered to the needs of previous generations of mobile networks, it would not be appropriate for 5G's rigorous list of requirements due to its lack of centralized coordination of radio resources.
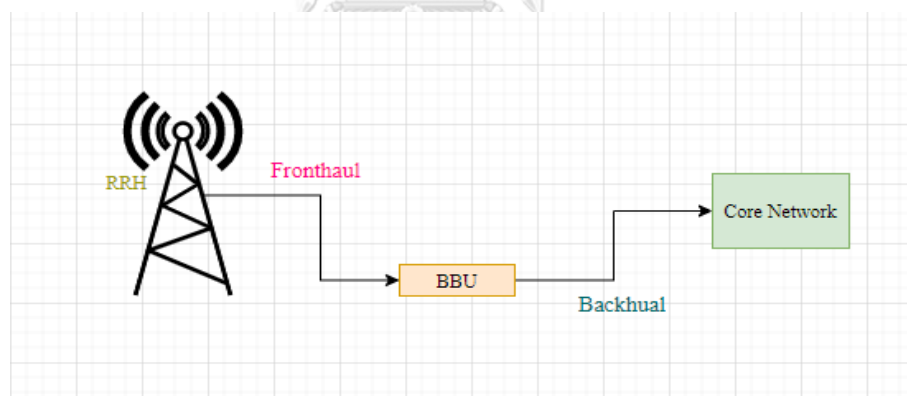
Figure: 2 Distributed RAN Architecture (D-RAN)

2.2.2.   CLOUD RADIO ACCESS NETWORK (C-RAN)

The Cloud Radio Access Network (C-RAN) is a next-generation radio access network architecture that centralizes the baseband processing responsibilities of the RAN in a cloud computing setting. This enables better performance, lower costs, and more effective resource use. The baseband processing tasks are carried out in the traditional RAN architecture at each base

station. Since not all base stations are constantly in use, this may not be resource efficient. Furthermore, the separation between the base stations and the central office may result in delays, impairing performance. C-RAN addresses these problems by centralizing the baseband processing operations. As a result, numerous base stations can share the baseband processing tasks, resulting in more effective use of resources[8].

Additionally, less space between the base stations and the central office could lead to better performance. The primary goal of C-RAN is to divide the standard base station in traditional RAN into two components, like RRH, which primarily handles radio frequency and baseband unit processing. C-RAN only emphasizes three key elements. These are Front-haul network, RRH, and BBU pool [9].
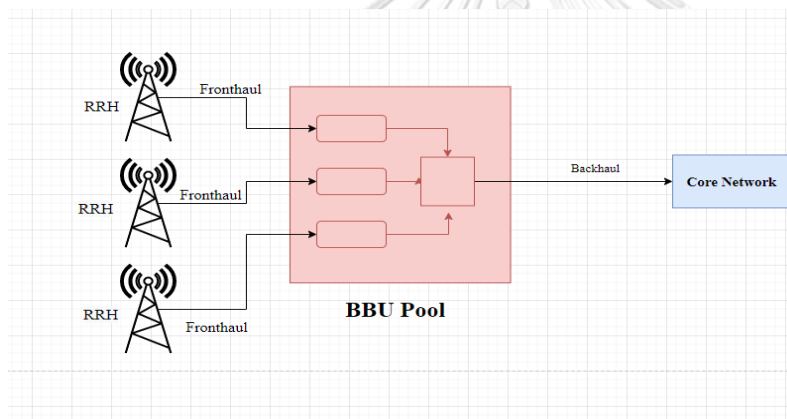
Figure: 3 Cloud Radio Access Network (C-RAN)

Sites can be challenging to manage and track on a traditional network. C-RAN is thus made available for centralized site management and monitoring. Furthermore, it saves less space because the CRAN needs more equipment. Both rent and the cost of energy dissipation are reduced. The 5G CRAN can also offer increased operational effectiveness. It is simple to meet the bandwidth and data rate needs in 4G since one BBU can only connect to one RRU; however, with the 5G C-RAN architecture, many RRUs can connect to one BBU [4]. Instead of the numerous BBUs everywhere, the conventional RAN employs a single central BBU (C-BBU). An optical link connects

the C-BBU to an IPAG, and an eCPRI interface connects it to distributed RRHs. It is a centralized technology-based architecture.

Additionally, it offers different cellular technologies, including 2G, 3G, and 4G. Clean space, centralized management, and real-time radio integration are some of the primary C-RAN (centralized RAN) capabilities. Increased flexibility is provided by C-RAN, which also adds a new communication channel called Fronthaul (FH) —the C-RAN design attempts to improve RAN performance by integrating virtualization and cloud computing ideas. There are several benefits to C-RAN, as well as some drawbacks, which are discussed in [1],[2].

There are some Cloud RAN benefits and drawbacks:

Pros:

Increased performance: By lowering latency and supplying greater bandwidth, cloud RAN can increase performance.

More effective use of resources: By distributing baseband processing tasks among several base stations, Cloud RAN may make better use of its resources.

Cost savings: By lowering the quantity of baseband processing units that must be deployed, cloud RAN can lower costs.

Flexibility: Cloud RAN can be readily scaled up or down to suit changing demands, making it more versatile than traditional RAN designs.

Scalability: Cloud RAN can accommodate a huge number of users and devices, making it very scalable.

Energy efficiency: Cloud RAN may use less energy than conventional RAN architectures since it can centralize baseband processing functions.

Cons:

Complexity: Cloud RAN is a sophisticated technology that calls for careful design and execution.

Security: Cloud RAN presents new security issues that need to be resolved.

Standardization: There is no single Cloud RAN standard because the technology is currently being developed.

Latency: Cloud RAN can cause latency since baseband processing operations must be sent over a network.

Cost: Since cloud RAN needs the deployment of a cloud infrastructure, it may be more expensive than traditional RAN architecture.

In conclusion, the Cloud Radio Access Network (C-RAN) offers increased efficiency, cost savings, flexibility, and more remarkable performance, and it marks a crucial turning point in mobile network architecture. With the help of C-RAN's centralized and virtualized approach, operators can optimize network resources, deliver outstanding products and services, and prepare for the coming era of mobile communications. C-RAN will shape the next wave of wireless connectivity as mobile networks develop[8].

2.2.3. CPRI vs eCPRI

To transfer data between baseband processing units and remote radio heads, also known as front-haul, mobile operators used the CPRI interface in 4G. However, in 5G, they switched to the eCPRI interface because the CPRI interface is incompatible with the 5G system. The Common Public Radio Interface, or CPRI, outlines the requirements for the REC and RE interface. Evolved Common Public Radio Interface, or eCPRI, is a CPRI publication standard that was later adopted. Purpose of deployment of CPRI & eCPRI :

To make the base station architecture simpler, divide the radio base station's functionality into two modules, such as eREC and eRE. They are linked by a transportation network.

Features of CPRI

- P-to-P Interface
- Master ports and Slave ports are connected directly to CPRI by optical or electrical cables
- Network Layer Function is not supported
- Technologies at CPRI depend on REC / RE functionality

- CPRI can support the following logical connections:

- P to P (From one REC and one RE)

- Point to multipoint (from one REC to multiple RE)

- REC / RE also supports Redundancy, Security, QoS

Function and Features of eCPRI

- Network includes eCPRI nodes (eREC and eRE)

- At physical level, master and slave ports are not supported.

- Located above the transport layer.

- eCPRI maintains the complexity of the eRE with flexible functional decomposition and reduces the data requirements between the eREC and the eRE compared to CPRI.

- The eCPRI specification makes it more flexible by placing it functionally within the physical layer of the base station.

2.3 C-RAN OPTIMIZATION

Many mobile network operators are looking for a new algorithm to solve the resource allocation (RA), load balancing, power efficiency, etc., problems to validate and streamline C-RAN deployment. Combinatorial optimizations are one of the effective methods [10] because they can tackle a wide range of issues, including scheduling, planning, and routing. These are all complicated issues that are challenging to tackle. One of the most excellent techniques for optimizing and resolving issues is linear programming (LP), which comes in two flavors: mixed integer linear programming (MILP) and integer linear programming (ILP). Managing and designing a Cloud Radio Access Network (C-RAN) network to achieve optimum performance is known as C-RAN optimization. This can be a challenging operation as there are many things to consider, including the number of base stations, the network's bandwidth, and user traffic patterns. To maximize or minimize the performance, efficacy, and cost-effectiveness of C-RAN setups, C-RAN (Cloud Radio Access Network) optimization is a crucial

component. Assigning the user demand to each split and evaluating cost optimization are critical components of this thesis. Baseband processing for the Cloud RAN is centralized in a cloud-based data center. It contributes to better performance and more effective resource use. Still, it also creates new difficulties, such as balancing the functional roles of the network's split and centralized components. Available split describes the splitting of baseband functions among the central unit (CU) and the remote radio unit (RRU) [1]. There are three primary functional split options:

- Full split: The CU is the central location for all baseband operations.
- Partial split: Some baseband operations are spread to the RRUs, while others are consolidated in the CU.
- No Split; the RRUs get all baseband functions.

The effectiveness of Cloud RAN depends on functional split optimization, a challenging but crucial issue. Optimization of the functional split aims to identify the partition that maximizes network performance while minimizing expense. Functional split optimization can be done in a variety of different ways. One method to discover the ideal split is to define the network mathematically and then use optimization techniques to determine it. The use of simulation to compare many split choices and select the one that performs the best is an alternative method. Operators can enhance their networks' efficiency and lower costs by optimizing the functional split.

The following are some advantages of functional split optimization:

- Enhanced performance: Functional split optimization can enhance network performance by lowering latency and boosting throughput.
- Reduced cost: By lowering the number of baseband units (BBUs) needed, functional split optimization can lower the cost of the network.

- Increased adaptability: By enabling operators to alter the split as necessary, functional split optimization can increase the adaptability of the network.

The following are some difficulties with functional split optimization:

- Complexity: Because there are so many variables to consider, functional split optimization is a challenging problem.
- Lack of standards: It is challenging for operators to assess various solutions since there are no standards for functional split optimization.
- Lack of information: Operators find it challenging to select the best split because there needs to be more information available on the effectiveness of the various functional split alternatives.

2.4. 3GPP FUNCTIONAL SPLIT OPTIONS

In a cellular network design, the baseband processing unit (BBU) and the radio unit (RU) are divided into different functional categories according to the 3rd Generation Partnership Project (3GPP) practical split idea. It outlines how the processing responsibilities are split between these two entities for a scalable and effective network operation. The concept of a functional split is especially applicable in the case of cloud radio access network (C-RAN) deployments, as the traditional baseband processing operations are split from the remote radio units and compiled in a BBU pool. This centralization enables enhanced coordination, better resource usage, and simpler network management. The 3GPP functional split offers a variety of alternatives for splitting the baseband processing tasks, allowing network operators and equipment suppliers to select the best configuration following their unique needs and deployment scenarios.

Functional Split Options [1][2][4]:

Option 1: RRC/PDCP - Baseband processing tasks are split between the distributed unit (DU) and the centralized unit (CU) in this option. In contrast to the DU, which is

in charge of lower-layer tasks like physical layer processing, the CU handles higher-layer activities like RRC and PDCP. As the processing load is distributed, this division allows for centralized control and management. Option 1 provides a balance between centralization and decentralization, enabling effective resource management and enhanced network performance.

Option 2: PDCP/RLC - The distributed unit (DU) and the centralized unit (CU) each perform a different baseband processing task in this option. While the CU is in charge of higher-layer functions, the DU is in charge of PDCP and RLC. The distributed processing of packet-level operations in Option 2 enables effective resource management and decreased latency. It provides a centralization/distributed processing trade-off for better network performance.

Option 3: intra RLC - Option 3 separates the RLC layer into the upper RLC and the lower RLC. While the lower RLC is situated in the DU, the upper RLC is found in the CU. This split enables better performance and more effective resource use.

Option 4: RLC/MAC - Baseband processing tasks are split between the distributed unit (DU) and the centralized unit (CU) in this option. While the DU is generally in charge of lower-layer tasks, the CU is in charge of higher-layer tasks like RLC and MAC. With Option 4, a sizable percentage of baseband processing may be centrally managed, improving network management and resource efficiency. The RLC and MAC functions are divided, allowing for flexibility in resource allocation and optimization tactics. As a result of the dispersed processing in the DU, lower-layer functions perform better, and the network as a whole is more efficient.

Option 5: intra MAC - In this option, a number of distributed units (DUs), each of which is in charge of a portion of the MAC functions, are used to distribute the baseband processing functions. Control and higher-layer duties remain with the centralized unit (CU). Option 5 increases resource efficiency and scalability by enabling load balancing and parallel processing by dividing MAC functions among several DUs. This division maintains centralized control for higher-layer operations while allowing for the effective management of medium access control activities like scheduling and resource allocation. In order to manage the MAC layer operations

within the cellular network, the intra MAC split offers flexibility and chances for optimization.

Option 6: MAC-PHY - Option 6 is a functional split option that is also known as the MAC-PHY split. In this arrangement, the distributed unit (DU) and the centralized unit (CU) each perform a different baseband processing task. The DU typically performs lower-layer tasks associated with the physical layer (PHY), whereas the CU is in charge of higher-layer tasks, including MAC. Option 6 enables effective resource allocation and optimization techniques by separating MAC and PHY functions. The divide improves overall network performance by enabling centralized control and maintenance of MAC operations while outsourcing PHY processing to the DU. This division allows for scalability and adaptability to various deployment circumstances and network requirements, offering flexibility in network deployments.

Option 7: Intra PHY - In this approach, a number of distributed units (DUs) are used to distribute the baseband processing functions, with each DU being in charge of a different set of PHY functions. The centralized unit (CU) continues to exercise control and management. By dividing PHY functions across many DUs, Option 7 provides parallel processing and load balancing, improving resource consumption and enhancing scalability. Through this division, physical layer duties like modulation, coding, and beamforming can be handled effectively while remaining under centralized control for higher-layer operations. The intra PHY split gives the cellular network architecture's physical layer operations flexibility and chances for optimization.

Option 8: PHY/RF - Baseband processing functions are split between the distributed unit (DU) and the centralized unit (CU) in this option. In contrast to the DU, which is in charge of RF-related tasks, the CU is in order of higher-layer tasks, such as PHY processing. Option 8 permits the division of PHY and RF operations, providing centralized control and management of PHY processing while dispersing RF operations closer to the radio units. With this division, the CU and DU can work together more effectively, quickly, and with better resource utilization. The cellular network can be deployed and optimized well with Option 8's division of the

processing jobs, primarily when RF processing must be carried out close to the radio units.
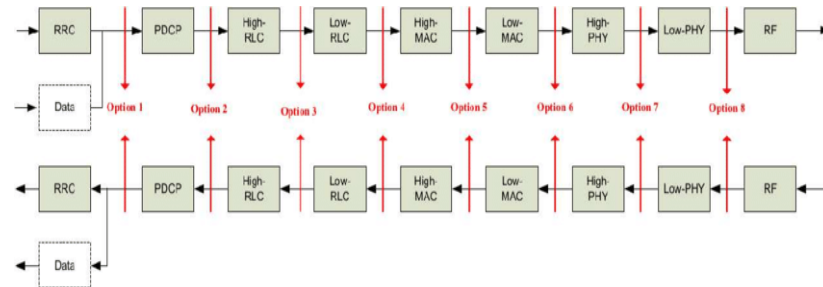

Figure: 4 3GPP Functional Split Options [2]

2.5.   INTEGER LINEAR PROGRAMMING

A robust optimization method for solving optimization issues where decision variables must take integer values is the Integer Linear Programming (ILP) algorithm. ILP is a subfield of mathematical programming that searches for an optimal answer by expressing the issue as a linear program with additional constraints on integer variables. ILP problems are frequently used to simulate resource allocation, scheduling, and routing issues. Different algorithms can be used to tackle ILP problems. Making use of a branch-and-bound algorithm is one typical strategy [24]. Branch-and-bound algorithms begin by analyzing every potential answer to the query. After that, they divide the issue into more manageable subproblems and recursively solve each. The algorithm ends when it discovers an optimal solution or when it can demonstrate that there isn't a better one. It can be challenging to address ILP issues. Since they are NP-hard, no polynomial-time method has been discovered to solve them consistently. However, ILP algorithms can be made to perform better by utilizing a variety of heuristics. Numerous problems in the real world have been resolved using ILP algorithms. For instance, ILP algorithms have been used to schedule airplanes, route trucks, and distribute resources in manufacturing facilities. To improve the performance of already-existing networks and to develop brand-new wireless ones, ILP algorithms are also used. ILP issues can be resolved using a variety of methods, including:

Branch and Bound: The branch and bound algorithm divide the problem space into smaller subproblems or branches and then iteratively explores each branch. The

branches that can't contain the best answer are cut off using upper and lower constraints.

Cutting Plane Techniques: Cutting plane techniques lower the viable zone by progressively adding more limitations (cutting planes) to the ILP formulation. These techniques assist in simplifying the formulation and removing subpar answers.

Heuristic: Heuristic approaches offer approximations of ILP issue solutions, which may not always be ideal but are computationally effective. Heuristics employ problem-specific rules or algorithms to direct the search for practical solutions.

Mixed Integer Programming: In mixed integer programming (MIP), when the integer constraints are relaxed to allow fractional solutions, ILP, and linear programming relaxation are combined. An initial answer is provided by the linear programming relaxation, which is then iteratively improved by branching and bounding on the integer variables.

The following are a few benefits of employing ILP algorithms:

Numerous real-world issues can be modeled using ILP algorithms.

ILP algorithms can be used to locate the optimal solutions to issues.

Even when an ideal solution cannot be identified, ILP algorithms can still identify workable answers to issues.

The following are some drawbacks of employing ILP algorithms:

ILP issues might be challenging to resolve.

ILP algorithms can take a lot of time.

Utilizing ILP algorithms can be costly.

In conclusion, the Integer Linear Programming (ILP) algorithm is a powerful optimization technique for addressing discrete decision-based combinatorial optimization problems. ILP offers a mathematical framework for determining the best solutions by presenting the problem as a linear program with additional constraints on integer variables. Although ILP problems might be computationally tricky, numerous solution strategies can address these issues, such as branch and bound, cutting plane methods, and heuristics [11].

2.6 PARTICLE SWARM OPTIMIZATION (PSO)

Particle Swarm Optimization (PSO) is a well-known population-based optimization technique motivated by the social behavior of fish schools and bird flocks. Kennedy and Eberhart created it in 1995. PSO is a metaheuristic algorithm that seeks out the most appropriate answer in a problem space by simulating the collective intelligence of a collection of particles. A group of particles in PSO stands for possible solutions to the optimization problem. Iterative adjustments are made to each particle's position and speed based on its knowledge and that of the other particles. The best-known solution for the entire swarm (global best) and the best-known solution for each particle (personal best) controls how the particles move. Particles explore the search space iteratively as they move closer and closer to the ideal outcome [12]. PSO is renowned for being straightforward and successfully resolving various optimization issues. It applies to continuous and discrete optimization domains because it does not require explicit knowledge of the case or gradients. Engineering, economics, data mining, and machine learning are just a few domains where PSO has been successfully used. The algorithm's success depends on its ability to balance exploitation and exploration. The particles utilize knowledge from the most successful solutions so far while scouring the search space for new, promising places. This equilibrium enables PSO to efficiently explore complicated and high-dimensional problem spaces in pursuit of optimal or nearly optimal solutions [13].

Pros:

Simplicity: PSO is comparatively simple to comprehend and apply compared to other optimization techniques. Users without a background in complex mathematics can utilize it thanks to its simple concept and few parameters. The following are some benefits and drawbacks of using PSO:

Exploration at a Global Scale: PSO encourages particle search across the solution space at a global scale. As a result, there is a higher chance that the algorithm will find the overall optimum by identifying alternative solutions in uncharted territory.

Population Diversity: PSO keeps its population of particles diversified, preventing an early convergence to less-than-ideal solutions. The exchange of knowledge and

learning among the particles enables a balanced exploration and exploitation of the search space.

Flexibility: PSO can address both continuous and discrete optimization issues, so it is adaptable to various disciplines. It is also simple to modify and expand to consider certain problem restrictions or add knowledge related to that challenge [5][11].

Cons:

Convergence to Local Optima: Unlike many other optimization techniques, PSO does not always find the global optimum. Its exploration capabilities may benefit from careful parameter setting and adjusting. In complicated and multimodal issue spaces, it may converge to local optima.

Sensitivity to Parameters: PSO performance may vary depending on the population size, social and cognitive characteristics, and velocity constraints. Choosing the correct parameter values for a particular situation requires trial and error or tweaking strategies to get the best results.

Cost of computation: PSO requires assessing the fitness function for each particle in each iteration, which can be time-consuming or costly for complicated issues or huge populations. The effectiveness of the method and the rate of convergence rely on the difficulty of the fitness evaluation and the size of the solution space [5][11].

CHAPTER 3

LITERATURE REVIEW AND RELATED WORKS

3.1. LITERATURE REVIEWS

The authors used a novel functional split orchestration scheme [5] to minimize the deployment cost of 5G C-RAN. They use integer linear programming (ILP) to generate the cost function for each split and particle swarm optimization (PSO) to optimize the cost function for each functional split. They proved their solutions had better performance for the resolution time and total deployment cost. The authors in [14] provide an end-to-end system analysis considering overall costs and energy usage. They suggest a mixed integer programming (MIP) formulation for the problem and use the IBM CPLEX Optimizer to analyze the impact of consumer delay requirements on decision-making. However, their model's application is constrained by its singular user emphasis. In [15], the authors provide graph-based architecture, considering the advantages of the resulting path established by the front-haul connection and the latency requirements set by each of the cells to reduce the cost of computing the allocation of resources (RA) at two locations. Although the fundamental characteristics of natural systems reflect by the assumption of RA costs and latency constraints, The author formulates the problem using graph clustering and applies a genetic algorithm.

The author in [16]provides a detailed explanation of the tele-traffic theory to analyze the allocation of resource periods and the resource allocation rate at the front-haul link. And then, based on this, the author explains in detail the purpose of saving energy and costs to realize a great deal for baseband processing sub-units when front-haul or baseband processing resources become further expensive for an operator. Moreover, it proves that user traffic has a high impact on segregation. The author formulates the issue using the tele-traffic technique and uses OPNET, a discrete event-based simulator. The study by the authors [17] presents a user-split orchestration approach that aims to reduce the front-haul link's energy and bandwidth consumption. The model uses quantitative models to calculate computing and link needs for each split. This strategy, however, is based on an inaccurate split model that treats the platform control function, including the MAC scheduler, as a user-centric

processing unit. The IBM CPLEX optimizer handles the optimization work and the problem construct using Mixed Integer Programming (MIP). The authors of [18] propose an integer linear programming (ILP) formulation of the problem and use the Lagrangian relaxation algorithm to minimize the number of routes that may lead to the release of critical services delay while also optimizing the cost of using baseband processing, which is used across multiple websites. Despite the network calculus technique, this function does not offer a quantitative standard method to calculate the cost of the requirements for each split. It is comprehensive to measure the delay on links on optical and wireless networks.

The authors [19] focus on choosing functional divisions despite considering various types of cell interference. Integer linear programming (ILP) and a heuristic approach solve the problem. They provide a novel heuristic technique to reduce the bandwidth used in the transport network and inter-cell interference. The functional split method performs at the cell level, which may have drawbacks in situations that call for a more fine-grained or user-centric approach. In the paper [20], a comprehensive model introduces to optimize the total cost of ownership (TCO) of a fiber-based RAN with split baseband processing units (BBU). The model takes quantifiable resource requirements for computation and links into account. Although it generates a split for each connected user in a cell, this coarse-grained approach might need to be revised. Integer linear programming (ILP) is used to define the problem, and the IBM CPLEX optimizer is used to resolve it.

In a paper [21], the authors present a technique for achieving energy efficiency in the 5G infrastructure that utilizes integer linear programming (ILP) and an LSTM-based neural network. Their strategy focuses on maximizing the functional split of optical transport to reduce overall energy consumption. It is crucial to remember that the split's deployment is generally cell-centric, which could limit its application in situations calling for flexibility or user-centric split setups. For RAN optimization and control, the author in [22] proposed deep reinforcement learning based on the double Q network. It focused on choosing the suitable schedule configuration for each real

eNodeB. They proved that the network performance outperformed the existing rule-based algorithm by applying deep reinforcement learning to an entire RAN system.

3.2. A COMPREHENSIVE SURVEY OF RESOURCE ALLOCATION ALGORITHMS IN CLOUD-RAN: AN OVERVIEW

Table 1 provides a comprehensive survey of cloud RAN resource allocation algorithms found in the literature. It highlights the taxonomy of these strategies in terms of: Reference, Title, Objective function, Problem Modeling and Algorithm.

Table: 1 Overview of Resource Allocations

Ref	Title	Objective Function	Problem Modeling	Algorithm
[5]	5G RAN: Functional Split Orchestration Optimization	To minimize both front-haul bandwidth and computational resources utilized	Integer Linear Problem (ILP)	Hybrid Particle Swarm Optimization (PSO)
[14]	Delay-aware Green Hybrid CRAN	To minimize the end-to-end delay, Front-haul bandwidth utilization and computational cost	Mixed Interger Linear Programming	IBM CPLEX optimizer
[15]	Graph-based Framework for Flexible Baseband Function Splitting and Placement in C-RAN	To minimize Front-haul delay and computational costs	Graph Clustering	Genetic
[16]	Evaluating C-RAN fronthaul functional splits in terms of network level energy and cost savings	To explore and analyze multiple split points for multiplexing improvements that relate to the cost and energy	Teletraffic theory	Discrete event-based simulator - OPNET

[17]	Interplay of energy and bandwidth consumption in cran with optimal function split	To reduce the front-haul link's energy and bandwidth consumption	MIP	IBM CPLEX optimizer
[18]	5G Infrastructures Supporting End-User and Operational Services: The 5G-XHaul Architectural Perspective	-To reduce the front-haul network's operational costs by lowering power consumption while maintaining rigorous delay limitations. -To reduce back-haul (BH) network end-to-end cloud service delay. Both goals are crucial for improving the effectiveness and performance of the entire network design.	Multi-objective service provisioning ILP	Lagrangian Relaxation
[19]	Flexible Functional Split in 5G Networks	To choose the most efficient split option that minimizes both fronthaul bandwidth usage and inter-cell interference.	ILP problem	Heuristic
[20]	Centralize or distribute? A Techno-Economic	To reduce the Base Station's (BS) total cost of owner ship (TOC) in H-RAN	Constraint Programming (CP)	IBM CPLEX/CP Optimizer

	Study to Design a Low-Cost Cloud Radio Access Network			
[21]	Provisioning of 5G Services Employing Machine Learning Techniques	To reduce the overall energy consumption of the 5G infrastructure.	ILP	Long Short-Term Memory (LSTM) NN model
[22]	Deep Reinforcement Learning-based Policy for Baseband Function Placement and Routing of RAN in 5G and beyond	To create an effective placement and routing strategy for BBF (Baseband Function) in both NG-RAN and C-RAN environments	ILP	Deep Reinforcement Learning (DRL)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

CHAPTER 4: METHODOLOGY AND IMPLEMENTATION

In this chapter, we aim to evaluate the computational cost and front-haul bandwidth utilization of Cloud Radio Access Network (C-RAN) architecture. We propose a Particle Swarm Optimization (PSO) algorithm and compare it with the existing Integer Linear Programming (ILP) approach. This chapter presents the methodology design and implementation to achieve our thesis objective. First, we evaluate the computation cost of RRU and RCC and front-haul bandwidth utilization using ILP. The ILP algorithm is implemented in experimental simulations to assess how well it optimizes the C-RAN functional split. In our study, we consider a set of K available splits that can be connected to N users. Each user is assigned to one division, represented by an ideal binary matrix. When split k is chosen for user i, each element of this binary matrix equals 1; otherwise, 0. We describe the RRU as a set of data units, each of which has a computational capacity of C_{RRU} GOPS, Power Consumption (P_U) and weight factor (α), and Power Usefulness Effectiveness (PUE_U). Additionally, we also describe the RCC as a set of data units, each of which is capable of doing computations with the following: computational capacity of C_{RCC} GOPS, Power Consumption (P_C), weight factor (β), and Power Usefulness Effectiveness (PUE_C). A_i^k (E_i^k) stands for the GOPS used at the RRU (or RCC) for the split k of user i, while is the front-haul bandwidth that the split k of user i generated. Front-haul link (FH) is used to connect RRU to RCC [15]. A technology-specific factor that shows the number of operations likely per second per Watt (W) of power consumption can be used to multiply the BBU complexity, measured in Giga Operations Per Second (GOPS). This factor is 40 GOPS/W in the reference scenario—furthermore, a fixed number of N static users within a single cell's coverage area. Using already established performance criteria, we compared the approaches. The objective function is to find a balance between the centralized and decentralized levels of the C-RAN. The centralized level is weighted by β and takes into account the RCC, PUE_C , and P_C. The decentralized level is weighted by α and takes into account the RRU,. PUE_U, and P_U . The traffic load on the fronthaul is taken into account by calibrating the weighting factor γ [5].

Thus, we model as ILP problem as follows:

Minimize:

$$\alpha. PUE_U. \frac{P_{RRU}}{P_U} + \beta. PUE_C. \frac{P_{RCC}}{P_C} + \gamma. \frac{FH_{rate}}{B}$$

Subject to:

$$\sum_i^K x_i^k = 1 \quad \forall i \in N \quad \text{(1)}$$

$$\sum_{i=1}^N \sum_{k=0}^K x_i^k R_i^k \le B \quad \text{(2)}$$

$$\sum_{i=1}^N \sum_{k=0}^K x_i^k A_i^k \le C_{RRU} \quad \text{(3)}$$

$$\sum_{i=1}^N \sum_{k=0}^K x_i^k E_i^k \le C_{RCC} \quad \text{(4)}$$

$$x_i^k \in [3], \forall i \in N, \forall k \in K$$

Where:

$$P_{RRU} = (1/P_f). \sum_{i=1}^N \sum_{k=0}^K x_i^k A_i^k$$

$$P_{RCC} = (1/P_f). \sum_{i=1}^N \sum_{k=0}^K x_i^k E_i^k$$

$$FH_{rate} = \sum_{i=1}^N \sum_{k=0}^K x_i^k R_i^k$$

The 1st constraint ensures that each user can only select one split option. One RRU and one RCC can only serve each user. 2nd constraint specifies the total supplied rate of the FH link. The total amount of data that can be transferred between the RRUs, and the RCC.3rd constraint specifies the RRU computation cost. The cost of processing data at the RRU.4th constraint determines the RCC computation cost. The cost of processing data at the RCC. P_{RRU} is used to evaluate the total power consumption within the Radio Remote Head (RRH). This's specifically considers the amount of power utilized to operate the RRH. As well as the P_{RCC} is used to evaluate the total power consumption within the Remote Cloud Center (RCC). This's specifically considers the amount of power utilized to operate the RCC. FH_{rate} is used to evaluate the total generated rate of the Fronthaul link. The total amount of data that can be transferred

between the RRUs and the RCCs [4, 5].

4.1. SIMULATIONS PERFORMANCE

In this section, we will delve into the simulation performance of our system. Firstly, we will outline the performance metrics that we have employed to evaluate the effectiveness of our system. Subsequently, we will present and analyze the results obtained from our simulations. To provide a comprehensive understanding, Table 2 shows the specific parameters and corresponding values that were utilized during the implementation of our system.

There are a total of seven split choices included in the 3GPP functional split, referred to as Splits 0 through 6. Within the Cloud Radio Access Network (C-RAN) architecture, each split choice denotes a certain arrangement of processing operations. The background chapter of this article mentioned the 3GPP functional split. The processing tasks carried out at the Remote Radio Head (RRH) level are included in Split 0. Processing tasks are no longer carried out at the RRH but are subsequently offloaded to the cloud or Baseband Unit (BBU) as we progress from Split 0 to Split 6. A more centralized and adaptable strategy can be used as a result of the split of processing tasks, with the computational workload being handled by the cloud or BBU. The numerous split choices enable the C-RAN architecture to adapt to varied deployment environments and optimize utilization of resources.

Table: 2 Simulation Parameters (ILP) [5]

| Parameters	Values
N users	30,40,50,60
K (Splits)	7
α, β, γ	0.8, 0.1, 0.1
B	1228 Mbps
PUE_U, PUE_C	2.3, 1.5
C_{RRU}, C_{RCC}	1060 GOPS
P_f	40

Results:
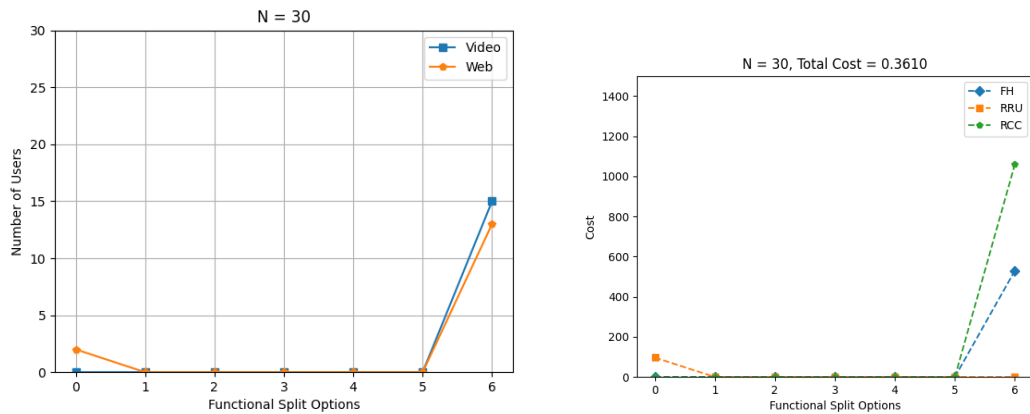
Users Assign different functional split   (b) Computational Costs

Figure: 5 Users Assign different functional split and Computational Costs (N=30)

The optimization process using the ILP algorithm yielded an optimal solution with a cost of 0.3610 when we assumed the *N* user is 30. Figure 5(a) illustrates the allocation of users across various splits using the ILP technique. Among the 30 total users, 2 users are assigned to Split 0, and the majority of users, 28 are distributed to Split 6. Notably, Splits 1 through 5 remain unassigned to any users.
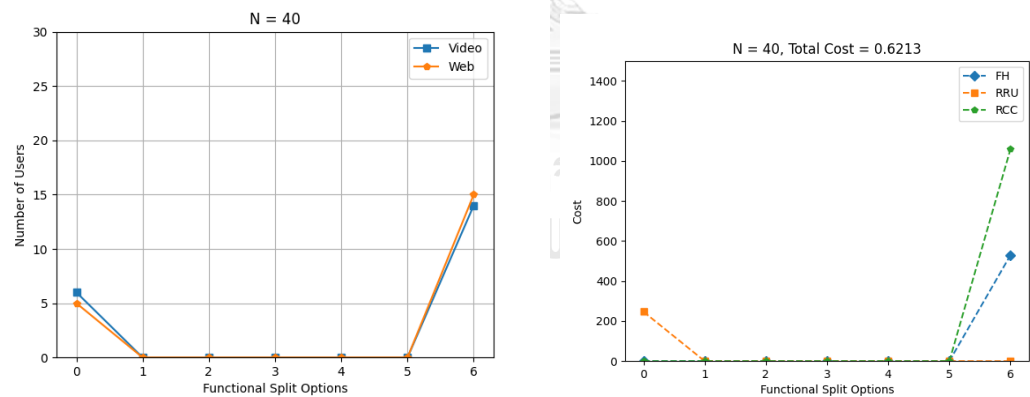
Users Assign different functional split   (b) Computational Costs

Figure: 6 Users Assign different functional split and Computational Costs (N=40)

Figure 6(a) shows the ILP technique to show how the users are distributed throughout the various splits. Figure 6(b) illustrates how the ILP algorithm successfully produced an optimal solution at a cost of 0.6213. When we increased the user count to 40 and took into account the two categories of user demand—web and video—the distribution of users among the various functional divides appeared like this: 15 people

were also assigned to Split 6 for web, out of the total 40 users, whereas 5 users were assigned to Split 0 for web services. Additionally, Split 0 was given access to 6 users for video services, and Split 6 was given access to 14 additional users for video services. It is important to note that in this arrangement, none of the users had their splits 1 through 5 assigned to them.
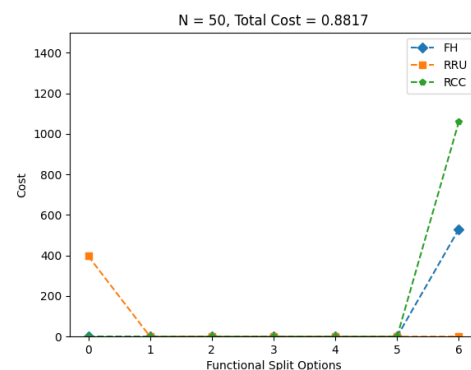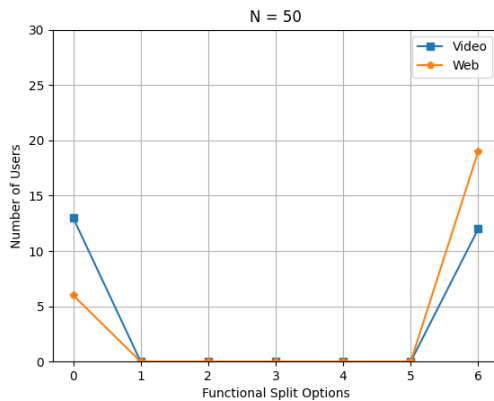
User Assign different split   (b) Computational Costs

Figure: 7 Users Assign different functional split and Computational Costs (N=50)

The ILP algorithm was able to arrive at an optimal solution with a cost of 0.8817 after taking into account a total of 50 users. The distribution of these users among several splits using the ILP technique is shown visually in Figure 7(a). 6 customers were given Split 0, which was created especially for web services. Additionally, Split 6—which served web services as well—was given 19 users. Moving on to video services, Split 0 was given to 13 customers, and Split 6 was given to another 12 people. It is significant to notice that in this arrangement, users were not assigned to Splits 1 through 5.
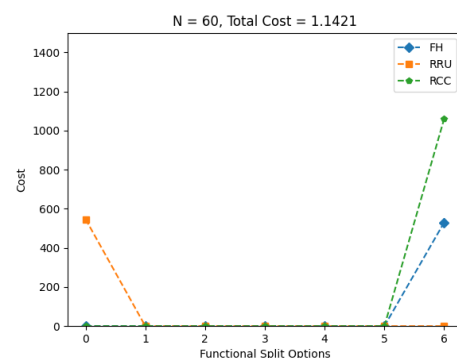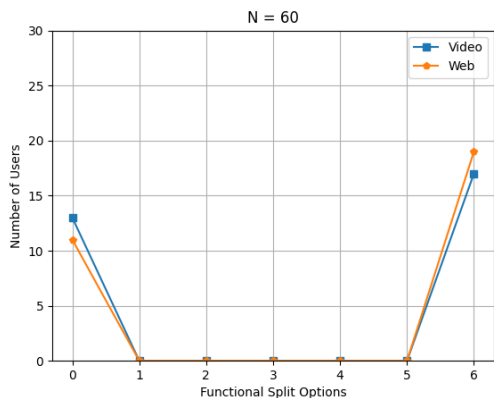
(a) User Assign different split   (b) Computational Costs

Figure: 8 Users Assign different functional split and Computational Costs (N=60)

Figure 8(a) displays, using an ILP technique, the distribution of 60 users among several splits. With a cost of 1.1420, the ILP algorithm was able to arrive at an ideal solution. 19 users could access web services on Split 6 as well. With regard to the video services, Split 0 was given to 13 users, while Split 6 was given to 17 users. It is interesting to note that in this particular arrangement, users were not assigned to Splits 1 through 5. In summary, in the optimization process utilizing the ILP method, the distribution of users across various splits is decided depending on the objective function and constraints defined in the algorithm. Since it was discovered that assigning users to Split 0 and Split 6 provided the most effective allocation approach, it is significant that users are assigned to these splits in Figures 5(a), 6(a), 7, and 8(a). However, no users were assigned to Splits 1 through 5. Because of the costing model and linearity included in the ILP formulation, users were not assigned to Splits 1 through 5. The ILP problem's costing model takes into account a number of cost variables related to each split option. These cost elements may include computational costs, resource usage, bandwidth needs, or any other pertinent metrics. The ILP algorithm seeks to identify the most cost-effective user allocation to reduce overall costs by taking these costs into account. In the ILP problem formulation, linear constraints are used to make sure that the distribution of users complies with specified requirements. Limitations on the number of users assigned to each split and the resources that can be used in each split are just two examples of these restrictions.

The ILP algorithm finds the most optimal user allocation by utilizing the costing model and the linear constraints to minimize costs while meeting the requirements. In this situation, the linearity of the ILP formulation and the costing model proposed assigning users mainly to Splits 0 and 6, since these splits were determined to be more effective and efficient in terms of cost and efficiency based on the provided costs and linear constraints.

4.2 PARTICLE SWARM OPTIMIZATION APPROACH

The optimization problem of the functional split, which was posed in the previous section III, is resolved in this section using the PSO algorithm. A population-

based optimization technique, the PSO algorithm is motivated by the behavior of fish schools and bird flocks. Each particle in the population of particles used by the algorithm initially stands for one potential solution to the problem [6]. The three main characteristics of each particle in the PSO algorithm are position, velocity, and personal best (Pbest). While the velocity is the change vector that allows the particle to advance to the next position, the position represents the possible solution configuration. The particle's best solution configuration, as determined by its evaluation using a cost function, is stored in the personal best (Pbest) memory function. The best solution configuration among all the best local solutions for particles is what we refer to as the global best (Gbest). It stands for the overall best answer any swarm particle has come up with. It compares the Pbest values of each particle in the swarm to arrive at this Gbest value and choose the configuration with the lowest cost.

The PSO algorithm enables particles to explore and optimize their solution configurations iteratively, seeking to converge towards the best feasible solution for the given problem, using these position, velocity, Pbest, and Gbest variables. After the algorithm has run through all steps, each particle iteratively works with the others to define its new velocity component[19]. (8) presents the procedure where the new velocity is created based on the old velocity of the previous iteration, Split P, Pbest, and Gbest. The coefficients C_1 and C_2 in (5) and inertia weight W are intended to enhance the process's randomness of evaluation. The random numbers r_1 and r_2 add a stochastic element to the velocity update equation, which helps prevent the swarm from converging too quickly to a local optimum [2, 19]. The new position P is updated once the new velocity has been calculated using (6).

$$V_{new} = V_{old} + C_1 r_1 (P_{best} - Split_P) + C_2 r_2 (G_{best} - Split_P) \qquad (5)$$

$$Split_P = Split_P + V_{new} \qquad (6)$$

4.2.1. SIMULATION PERFORMANCE (PSO)

In this section, we will discuss how well our system performs in simulation. We will define the metrics that we will use to measure performance, and we will discuss the results of our simulations. Table (3) shows the parameters and values that we used in our implementation.

Table: 3 Simulation Parameters (PSO) [5]

Parameters	Values
P (particle)	10
N (user)	30,40,50
MaxIt (maximum iteration)	10,15,20,25
K (split option)	7
α, β, γ	0.8, 0.1, 0.1
B	1228 Mbps
PUE_U, PUE_C	2.3, 1.5
C_{RRU}, C_{RCC}	1060 GOPS
P_f	40

When we consider a scenario with 30 users (*N*), the ILP algorithm yielded a deployment cost of **0.3610**, as indicated in the previous results. In our implementation, we utilized the seed function to initialize the random number generator with a specific seed value. This deliberate control over the seed value ensures reproducibility in the generation of random numbers, resulting in consistent outcomes across multiple program executions. By maintaining the same seed value, the random number generator consistently produces the same sequence of random numbers every time the application is run. Moreover, we observed that increasing the maximum number of iterations (*MaxIt*) led to a reduction in deployment costs. This implies that allowing the algorithm to iterate more times resulted in improved optimization and decreased expenses. To highlight the impact of different random number generator seeds on the

deployment cost, Figure (9) and Table (4) present the cost values obtained for various seed values in our experiments.
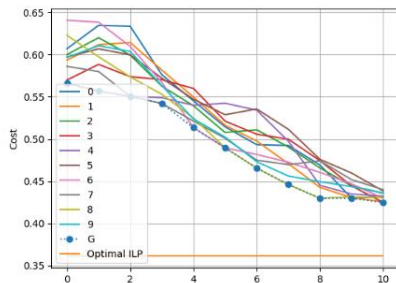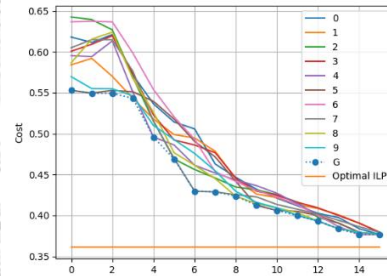

(a)   Seed Value - 62720
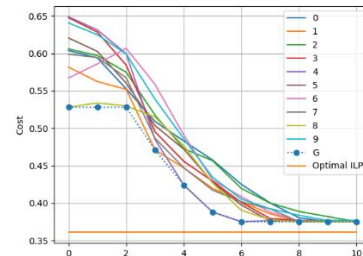
(b) Seed Value - 34342
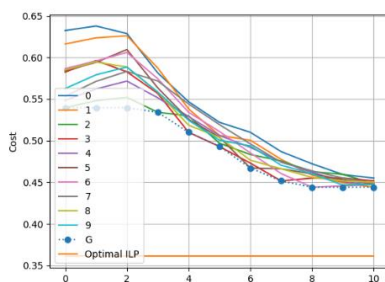
(c) Seed Value - 83321
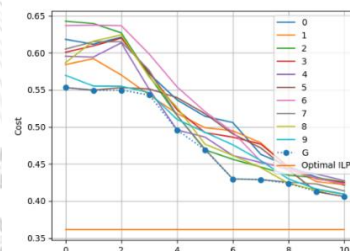
(d) Seed Value - 52454

Figure: 9 Total Deployment Cost when N=30, P =10, MaxIt = 10

In order to investigate how randomization affected the outcomes, we ran many optimizations runs with various seed values. We employed four distinct seed values, namely 62720, 34342, 83321, and 52454, for a maximum of 10 iterations. For each run, a different beginning point was selected using these seed values at random. The following outcomes were attained when we ran the optimization method with each seed value: Figure 9 (a) Seed: 62720, outcome: 0.4473 (b) Seed: 34342; outcome: 0.4247 (c) Seed: 83321; outcome: 0.4057 (d) Seed: 52454, outcome: 0.3768. With respect to the appropriate seed value, each result reflects the optimization result that was attained after the specified number of iterations.
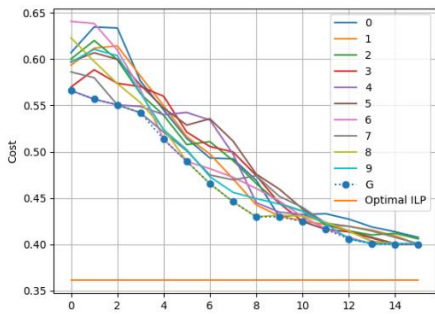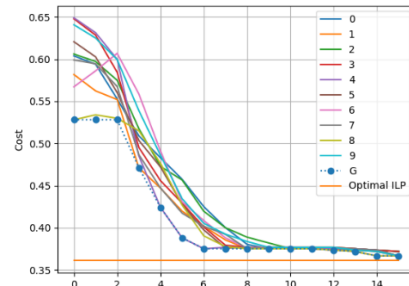
(a) Seed Value - 62720   (b) Seed Value - 34342
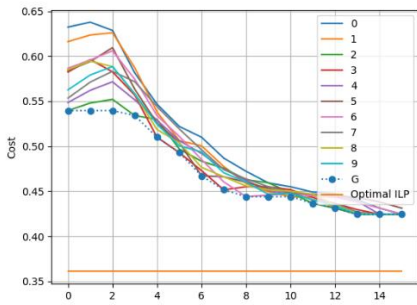
(c) Seed Value - 83321   (d) Seed Value - 52454

Figure: 10 Total Deployment Cost when N =30,P =10, Maxit =15

We carried on investigating the results of the optimization after increasing the iteration count from 10 to 15. For each seed value, the following results were obtained: Figure 10 (a) seed: 62720, result: 0.4243; (b) seed: 34342, result: 0.3998; (c) seed: 83321, result: 0.3768;(d) seed: 52454, result: 0.3667.
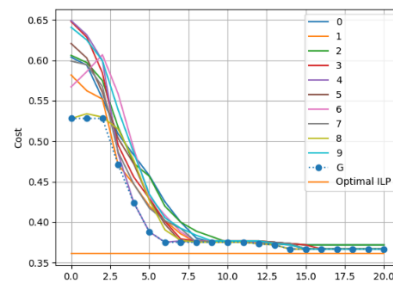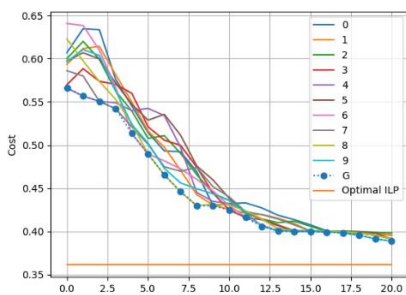
(a) Seed Value - 62720   (b) Seed Value - 34342
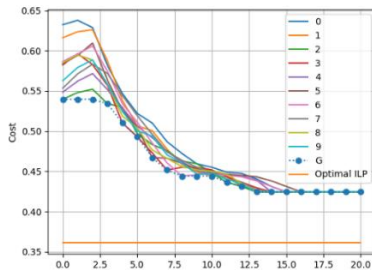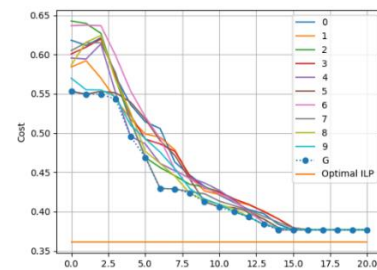
(c) Seed Value - 83321   (d) Seed Value - 52454

Figure: 11 Total Deployment Cost when N =30,P =10, MaxIt =20

The optimization results were further explored while using the same set of seed values as we increased the number of iterations from 15 to 20. For each seed value, the following results were obtained: Figure 11. (a) Seed: 62720; Result: 0.4243; (b) Seed: 34342; Result: 0.3887; (c) Seed: 83321; Result: 0.3768; (d) Seed: 52454; Result: 0.3667. These findings provide more information about the iterative nature of optimization and the effects of various seed values on the results attained. Our understanding of the behavior of the algorithm and its convergence to nearly ideal answers improved as we increased the number of iterations.
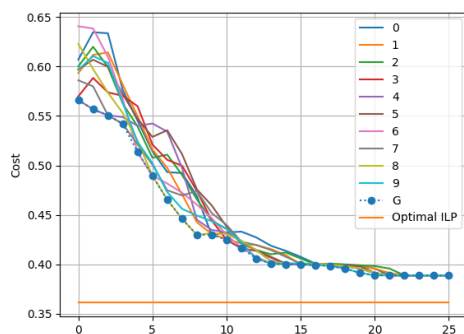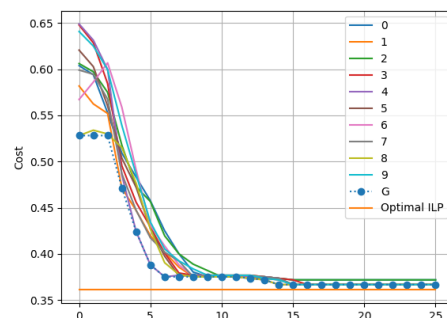
(a) Seed Value – 62720   (b) Seed Value - 34342
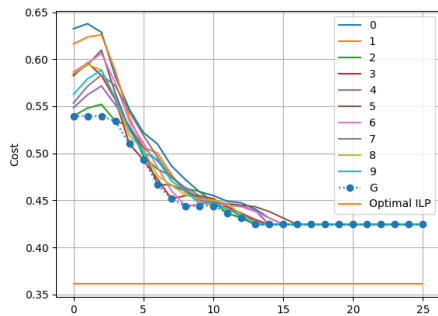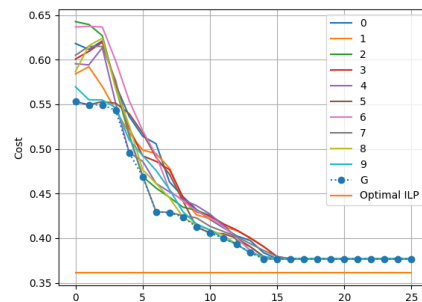
(c) Seed Value - 83321   (d) Seed Value - 52454

Figure: 12 Total Deployment Cost when N=30, P=10, MaxIt=25

We continued our analysis of the optimization results using the same set of seed values as we increased the iteration count from 20 to 25. The findings obtained for each seed value remained consistent and did not demonstrate any notable changes: Figure 12.(a) Seed: 62720, Result: 0.4243, (b) Seed: 34342, Result: 0.3887, (c) Seed: 83321, Result: 0.3768, (d) Seed: 52454, Result: 0.3667.These findings suggest that the optimization process has stabilized because no substantial improvements or deviations were seen after the further iterations. The results' convergence indicates that the algorithm succeeded in finding an effective approach that consistently yields the same results.

When N is set to 30, Table 4 provides a comprehensive summary of the deployment costs discovered throughout the optimization process.

Table: 4 Summarize the results of Deployment cost when N = 30, P = 10

| Seed values	Maximum iterations			
	10	15	20	25
62720	0.4473	0.4243	0.4243	0.4243
34342	0.4247	0.3998	0.3887	0.3887
83321	0.4058	0.3768	0.3768	0.3768
52454	0.3768	0.3667	0.3667	0.3667
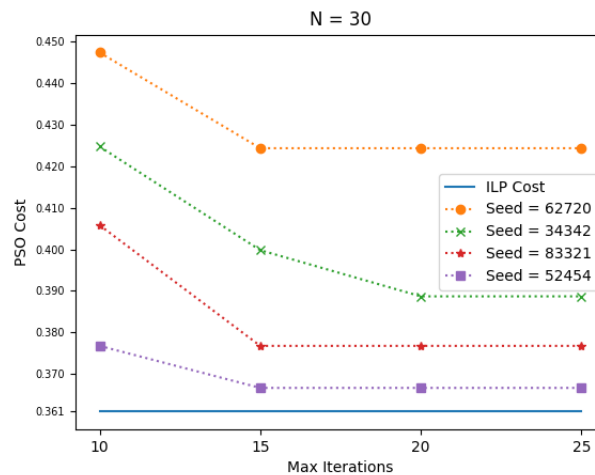
Figure: 13 Deployment Cost Convergence with Increasing Number of Iterations (N=30)

When we assume there are 40 *N* users, we obtained a deployment cost of **0.6213** for ILP, as shown above. We use seed function to initialize the random number generator with a predefined seed value for each execution in our implementation. Controlling the seed value enables us to establish reproducibility in the generated random numbers, fostering consistent behavior across several program runs. The random number generator will reliably provide the same set of random numbers to this intentional initialization every time the application is run. We observed that increasing the maximum number of iterations (*MaxIt*) led to a decrease in deployment costs. Table (5) shows the cost of deploying a system with different random number generator seeds.
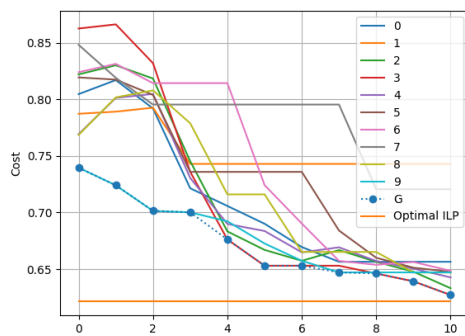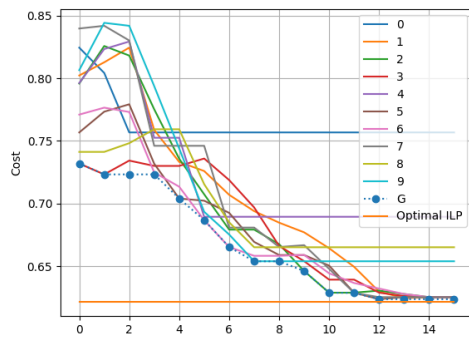
(a) Seed Value – 62720
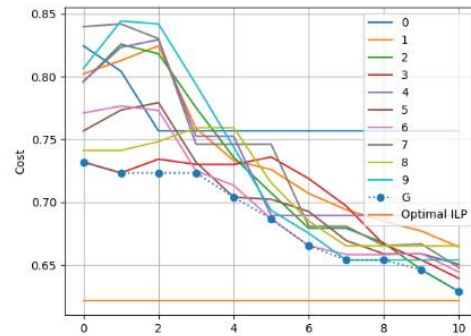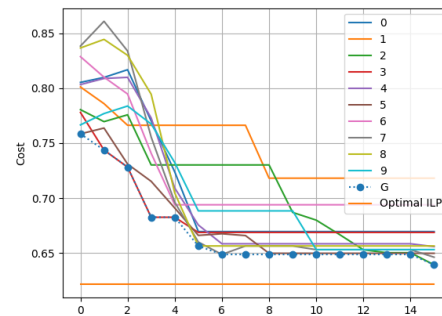
(b) Seed Value - 34342
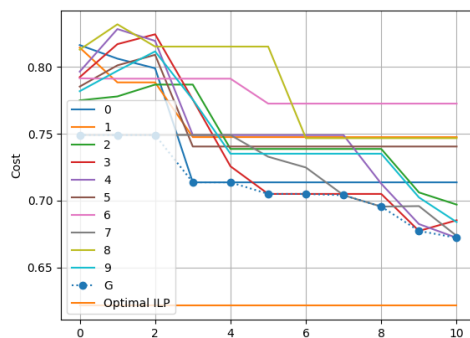
(c) Seed Value – 83321
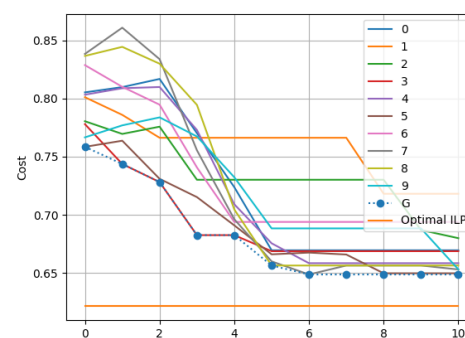
(d) Seed Value - 52454

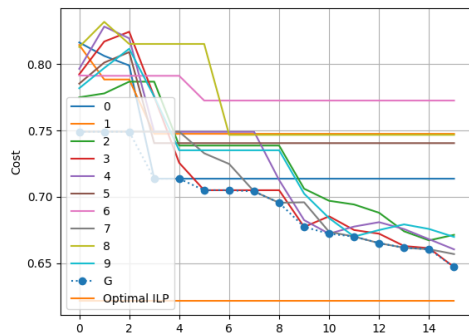*Figure: 14 Total Deployment Cost when N =40,P =10, MaxIt =10*

To find out the effect of random assignments on the results, we executed numerous optimizations runs with various seed values. We specifically chose the four random seed values 62720, 34342, 83321, and 52454 and set the maximum number of iterations to 10. The optimization algorithm has a different starting point for each seed value. The algorithm was run with each seed value, and the following results were attained: The following results were obtained: Figure 14. (a) Seed: 62720, Result: 0.6288 (b) Seed: 34342, Result: 0.6529 (c) Seed: 83321, Result: 0.6721 (d) Seed: 52454, Result: 0.6721. Each results reflect the optimization result attained for the corresponding seed value after the set number of iterations.
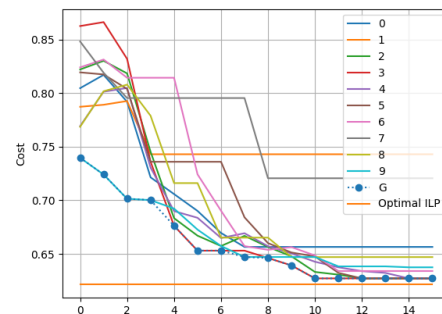
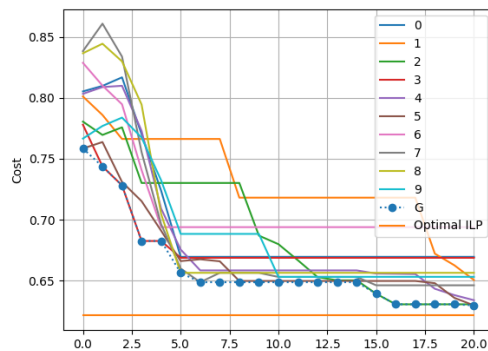(a) Seed Value – 62720

(b) Seed Value - 34342
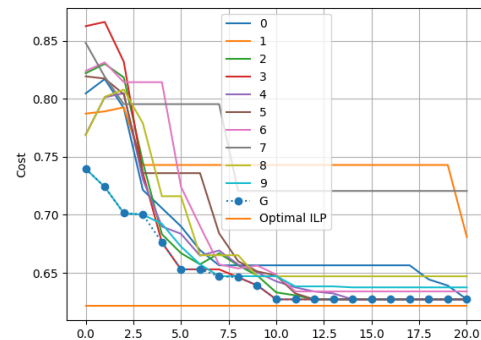

(c) Seed Value – 83321

(d) Seed Value – 52454

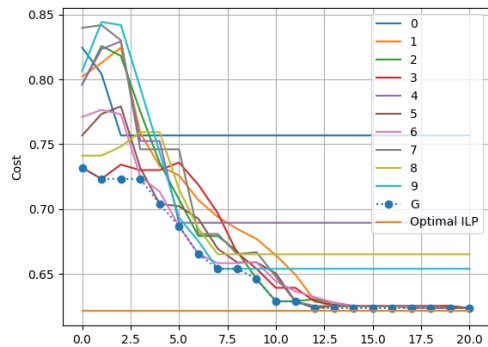*Figure: 15 Total Deployment Cost when N =40,P =10, MaxIt =15*

We continued exploring the optimization results while using the same set of seed values as we increased the number of iterations from 10 to 15. The outcomes are as follows for each seed value: Figure 15. (a) Seed: 62720, Result: 0.6252; (b) Seed: 34342, Result: 0.6391; (c) Seed: 83321, Result: 0.6469; (d) Seed: 52454, Result: 0.6271. The results demonstrate the outcomes of optimization attained for each seed value following the number of iterations that was provided. We were able to assess the consistency of the outcomes across various seed values by extending the iteration process and gaining greater insights into the convergence and stability of the optimization algorithm.
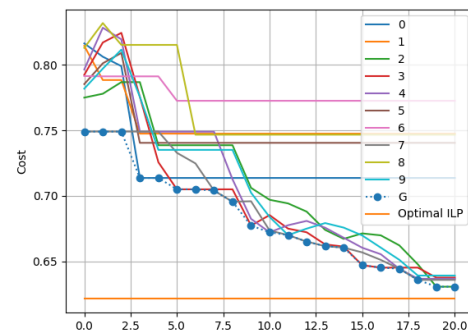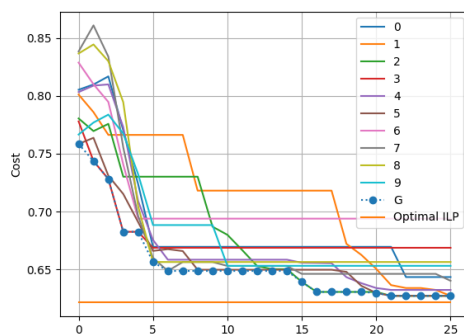
(a) Seed Value – 62720
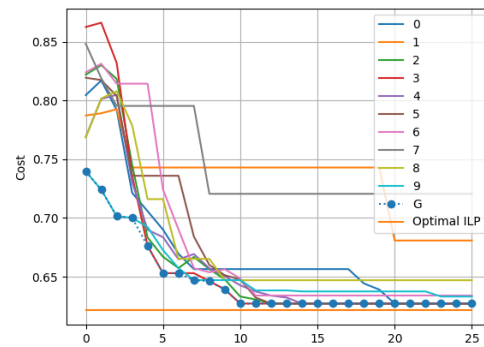
(b) Seed Value - 34342

(c) Seed Value – 83321

(d) Seed Value – 52454

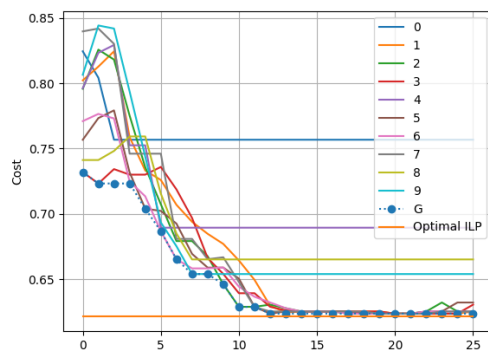*Figure: 16 Total Deployment Cost when N =40,P =10, MaxIt =20*

We performed a more thorough investigation of the optimization results using the same set of seed values by increasing the number of iterations from 15 to 20. The outcomes for each seed value are as follows: Figure 16. (a) Seed: 62720, result: 0.6236; (b) Seed: 34342, result: 0.6296; (c) Seed: 83321, result: 0.6305; (d) Seed: 52454, result: 0.6271. These results shed more light on the iterative nature of optimization and the impact of varying seed values on the final results. We were able to comprehend the behavior of the algorithm and its convergence to near-optimal solutions better by increasing the number of iterations.
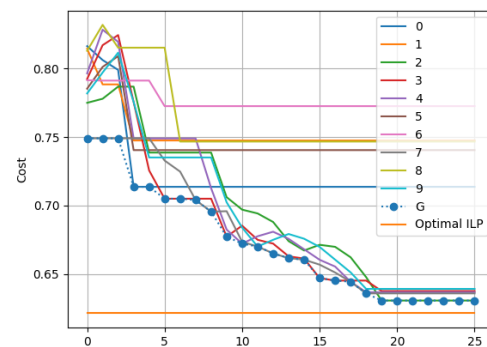
(a) Seed Value – 62720

(b) Seed Value - 34342

(c) Seed Value – 83321

(d) Seed Value – 52454

Figure: 17 Total Deployment Cost when N =40,P =10, MaxIt =25

We subsequently investigated the optimization results using the same set of seed values after increasing the number of iterations from 20 to 25. Surprisingly, the outcomes for every seed value remained the same and did not show any notable variations. Figure 17 shows the seed values in more detail. (a) Seed: 62720, Result: 0.6231 (b) Seed: 34342, Result: 0.6305, (c) Seed: 83321, Result: 0.6305, (d) Seed: 52454, Result: 0.6271. These results suggest that the optimization process has stabilized because no appreciable advancements or departures were seen with the additional iterations. The repeatable findings suggest that the algorithm has reached a robust solution, regularly producing the same results. This consistency of the optimization results indicates that the method has successfully solved the optimization problem

within the specified limitations, and that additional iterations may not result in appreciable improvements.

Table 5 offers a comprehensive summary of the deployment costs identified throughout the optimization process when N is set to 40.

Table: 5 Summarize the results of Deployment cost when N =40,P = 10

Seed values	Maximum iterations			
	10	15	20	25
62720	0.6288	0.6252	0.6236	0.6231
34342	0.6529	0.6391	0.6296	0.6271
83321	0.6721	0.6469	0.6305	0.6305
52454	0.6271	0.6271	0.6271	0.6271

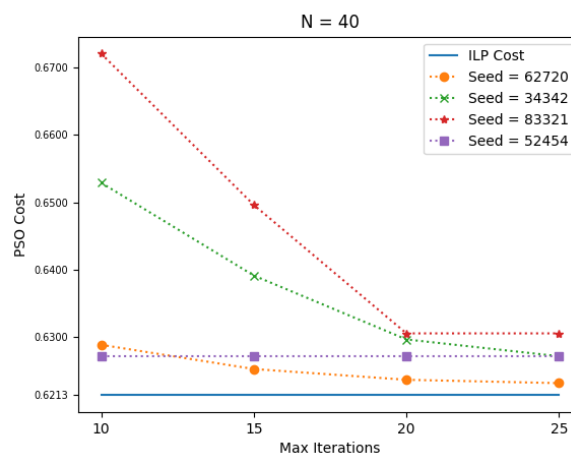*Figure: 18 Deployment Cost Convergence with Increasing Number of Iterations (N=40)*

Assuming a scenario with 50 users (*N*), the ILP algorithm yielded a deployment cost of **0.8817**, as evidenced in the preceding results. In our implementation, we employed the seed function to initialize the random number generator using a predetermined seed value for each program execution. By controlling the seed value, we ensure the reproducibility of the generated random numbers, ensuring consistent behavior across

multiple runs of the program. With the same seed value, the random number generator consistently generates the exact set of random numbers during each execution of the application. Furthermore, we observed a correlation between the increase in the maximum number of iterations (*MaxIt*) and the decrease in deployment costs. This indicates that allowing the algorithm to iterate for a greater number of times leads to improved optimization and reduced expenses. To provide further insights, Table (6) presents the deployment cost for different random number generator seeds, highlighting the impact of seed values on the system's deployment expenses.
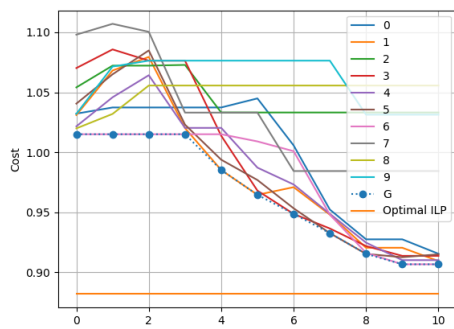
(a) Seed Value – 62720
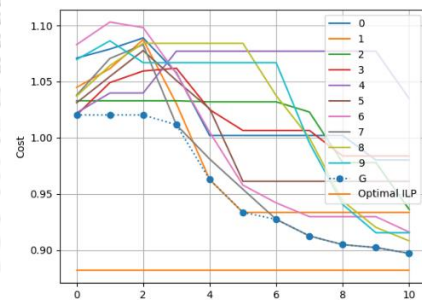
(b) Seed Value - 34342
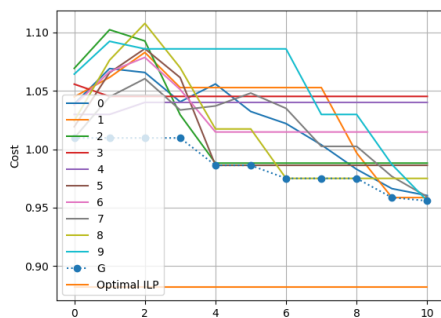
(c) Seed Value – 83321
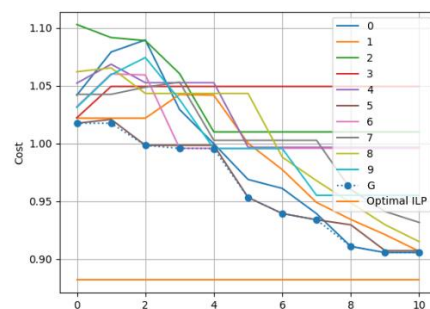
(d) Seed Value – 52454

Figure: 19 Total Deployment Cost when N =50,P =10, MaxIt =10

We conducted several runs using various seed values during the optimization process to look at the impact of randomness on the results. In particular, we defined a maximum iteration number of 10 and carefully chosen four random seed values:

62720, 34342, 83321, and 52454. Each seed value served as a crucial starting point for the optimization procedure. We obtained the following results by running the method with each seed value: Figure 19 (a) Seed: 62720, Result: 0.9065; (b) Seed: 34342, Result: 0.8965; (c) Seed: 83321, Result: 0.9558; and (d) Seed: 52454, Result: 0.9055.
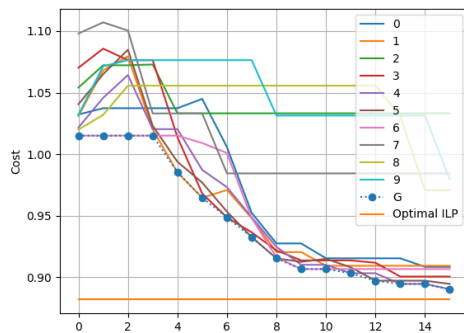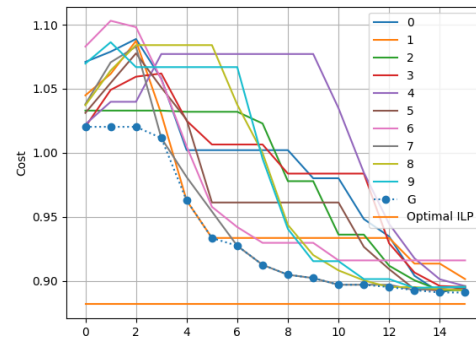

(a) Seed Value – 62720　　　　　　　(b) Seed Value - 34342
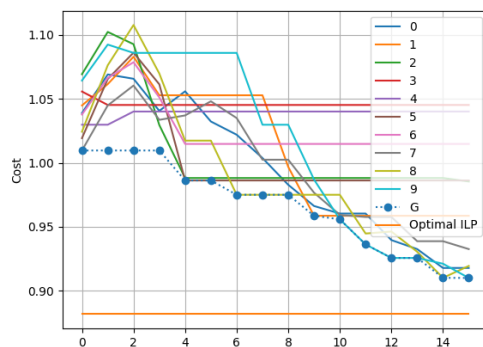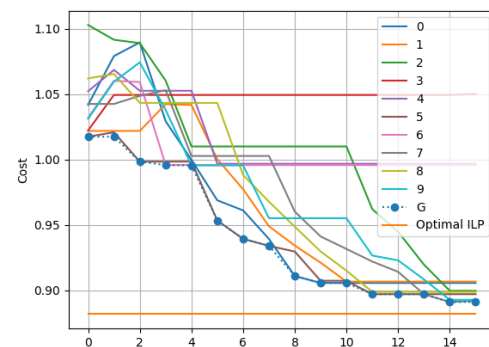
(c) Seed Value – 83321　　　　　　　(d) Seed Value – 52454

Figure: 20 Total Deployment Cost when N =50,P =10, MaxIt =15

We increased the number of iterations from 10 to 15 while using the same set of seed values in order to acquire more insight into the optimization results. Following are the outcomes for each seed value: Figure 20. (a) Seed: 62720, Result: 0.8901; (b) Seed: 34342, Result: 0.8951; (c) Seed: 83321, Result: 0.9099; and (d) Seed: 52454, Result: 0.8918. These results show the optimization results for each seed value after the predetermined number of iterations. We got deeper understanding of the convergence

and stability of the optimization method by extending the iteration procedure, which allowed us to assess the consistency of the outcomes for various seed values.

(a) Seed Value – 62720   (b) Seed Value - 34342

(c) Seed Value – 83321   (d) Seed Value – 52454

Figure: 21 Total Deployment Cost when N =50, P =10, MaxIt =20

We raised the number of iterations from 15 to 20 while using the same set of seed values in order to acquire a deeper grasp of the optimization results. The outcomes we got for each seed value are as follows: Figure 21 (a) Seed: 62720, Result: 0.8848, (b) Seed: 34342, Result: 0.8840, (c) Seed: 83321, Result: 0.8908, and (d) Seed: 52454, Result: 0.8910. The iterative nature of the optimization process and the impact of various seed values on the results attained are both useful insights offered by these findings. We were able to comprehend the algorithm's behavior and its convergence to almost ideal answers by increasing the number of iterations.
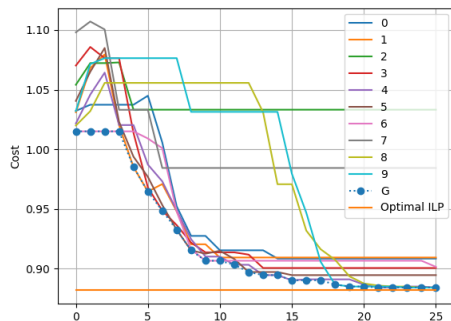
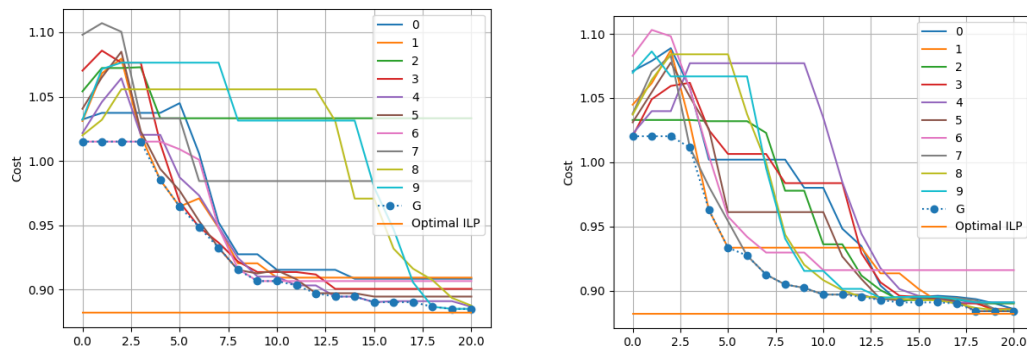(a) Seed Value – 62720
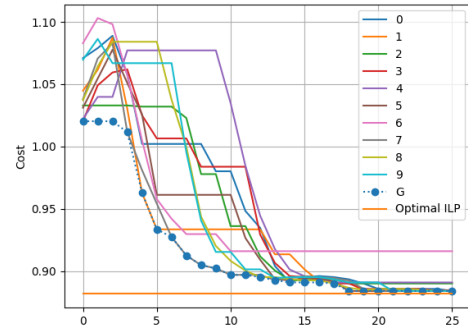
(b) Seed Value - 34342
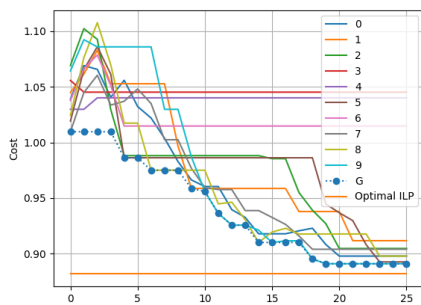
(c) Seed Value – 83321
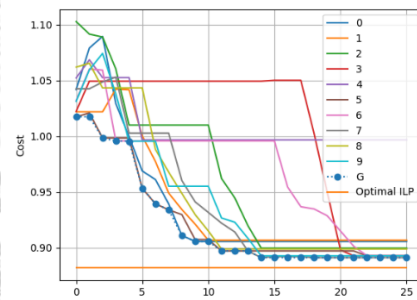
(d) Seed Value – 52454

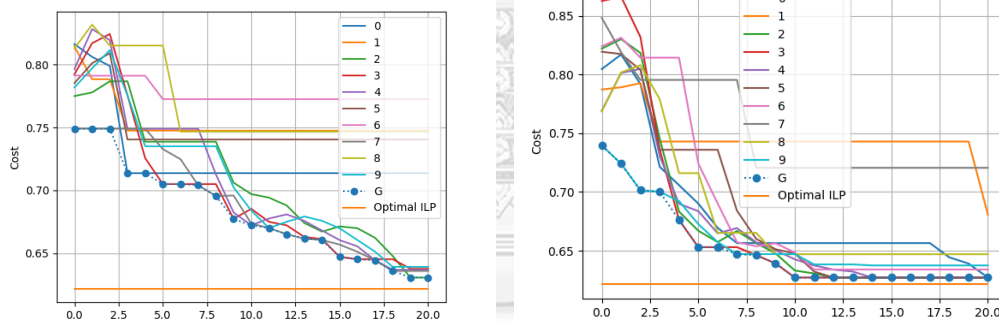*Figure: 22 Total Deployment Cost when N =50,P =10, MaxIt =25*

We carried out deeper studies into the optimization results utilizing the same set of seed values after increasing the number of iterations from 20 to 25. Surprisingly, the outcomes for every seed remained the same and showed no significant variations. For the seed values in particular: figure 22. (a) Seed: 62720, Result: 0.8840, (b) Seed: 34342, Result: 0.8840, (c) Seed: 83321, Result: 0.8908, and (d) Seed: 52454, Result: 0.8909. These results show that the optimization process has stabilized as no significant advancements or deviations were found with the extra iterations. This stability in the optimization results indicates that more iterations might not result in appreciable gains and that the method has successfully solved the optimization problem within the set limitations.

Table 6 offers a comprehensive summary of the deployment costs identified throughout the optimization process when N is set to 50.

Seed values	Maximum iterations			
	10	15	20	25
62720	0.9065	0.8901	0.8848	0.8840
34342	0.8969	0.8951	0.8840	0.8840
83321	0.9558	0.9099	0.8908	0.8908
52454	0.9055	0.8918	0.8910	0.8909

Table: 6 Summarize the results of Deployment cost when P=10, N = 50
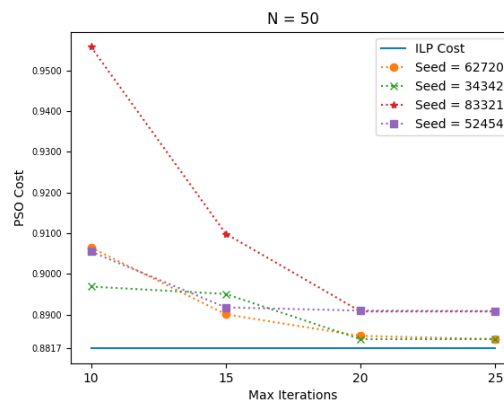
Figure: 23 Deployment Cost Convergence with Increasing Number of Iterations (N=50)

Figure 13, figure 18 and figure 23 illustrate our exploration of the impact of increasing the number of iterations on deployment costs. As we progressively increased the iterations to values of 10, 15, 20, and 25, we observed a convergence in the deployment costs. These results were obtained through experimentation and analysis, allowing us to gain insights into the relationship between iterations and costs in our scenario. By visualizing this data, we can better understand the effect of varying iterations on the convergence of deployment costs.

During the optimization process with N = 30, the computation time in seconds was recorded for various seed values and the maximum iterations. In the table below, the findings are summarized:

Table: 7 The computational time in seconds for N=30,

Seed value	Maximum iterations			
	10	15	20	25
62720	0.132010	0.104009	0.079012	0.094023
34342	0.087988	0.062243	0.073546	0.090364
83321	0.079543	0.104752	0.099994	0.077290
52454	0.089540	0.120098	0.088963	0.109535

We can see from the table that the computation time varies for various seed values and maximum iterations. When the maximum number of iterations is increased, the computational time for the seed value of 62720 reduces. It starts at 0.132010 seconds at 10 iterations and drops to 0.094023 seconds at 25 iterations. Computing time varies in diverse ways across the maximum iterations for the seed values 34342, 83321, and 52454. These findings demonstrate how the number of iterations and seed values both affect the amount of time needed for the optimization process to be computed. The effectiveness and performance of the algorithm for various problem scenarios can be better understood by analyzing and comprehending these changes.

During the optimization process with N=40, the computing time in seconds for various seed values and maximum iterations was measured. The findings are shown in the table below:

Table: 8 The computational time in seconds for N=40,

Seed value	Maximum iterations			
	10	15	20	25

62720	0.082582	0.086007	0.073386	0.111707
34342	0.103164	0.087823	0.086892	0.09880
83321	0.074004	0.079721	0.085021	0.106060
52454	0.075006	0.090069	0.080992	0.096529

According to the findings, the computational time varies for various seed values and maximum iterations. When the maximum iterations are reached for the seed value 62720, the computational time varies and gradually gets shorter. Between 10 and 25 iterations, the time varies from 0.082582 to 0.111707 seconds. Likewise, the computational time distribution throughout the maximum iterations differs for the seed numbers 34342, 83321, and 52454.

When using N=50 for optimization, the computation time in seconds was captured for various seed values and the maximum iterations. Given below is a summary of the findings in a table:

Table: 9 The computational time in seconds for N=50,

Seed value	Maximum iterations			
	10	15	20	25
62720	0.081006	0.078142	0.081816	0.112000
34342	0.097148	0.088656	0.084475	0.104992
83321	0.072767	0.087628	0.099333	0.106663
52454	0.068957	0.088009	0.100640	0.122741

The results indicate that the outcomes, the computational time varies for various seed values and maximum iterations. With a range of 0.081006 seconds at 10 iterations to 0.112000 seconds at 25, the computational time for the seed value 62720 remains largely constant over the maximum number of iterations. Likewise, the computational time distribution throughout the maximum iterations differs for the seed numbers

34342, 83321, and 52454. These versions demonstrate how the number of iterations, and the seed values affect the amount of time needed for the optimization process to run on a computer. For larger problems, such N=50, being aware of these trends can be quite helpful in understanding how well the algorithm works and performs.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

CHAPTER 5: Conclusion

5.1 CONCLUSION

Cloud Radio Access Network (C-RAN) shows significant potential for cutting wireless network construction and maintenance expenses. The baseband processing tasks of a wireless network are centralized by C-RAN, which can drastically reduce the number of base stations needed and the equipment required at each base station. This may result from significant cost reductions, enhanced capacity, and performance. The placement of RAN functions must be optimized to reduce computational costs and bandwidth utilization. It necessitates considering the needs of many users and the resources available, which is a challenging problem. Optimizing the placement of RAN functions in C-RAN is something we want to do in our study by introducing a novel method [2, 8]. Our method is based on Particle Swarm Optimization (PSO), a metaheuristic method that has effectively solved various optimization issues. As we demonstrate by evaluating our approach, the overall cost of building a C-RAN network can significantly decrease. By using our method, C-RAN networks' efficiency and efficacy can be increased. We intend to conduct more studies in this area and create new systems to enhance the placement of RAN functions in C-RAN. Based on our research, wireless network deployment and upkeep using C-RAN may be done affordably. We may reduce deployment costs and boost network performance by carefully optimizing where RAN functionalities are placed[8].

In this thesis, we used particle swarm optimization (PSO) to reduce computational and bandwidth expenses in cloud radio access networks (C-RAN). We concentrated on minimizing the functional placement cost of the RAN, which comprises the cost of computation and the use of fronthaul bandwidth by several users. We thoroughly examined the PSO-based method and found that, because of its capacity to consider a wide variety of potential solutions and respond to environmental changes, it can produce effective answers. The outcomes of our research show that the PSO-based approach converges with the best options. When

comparing it to random search, we also found a number of noteworthy features. Notably, the performance of PSO closely resembles the optimal result obtained by Integer Linear Programming (ILP) as the number of iterations increases. This suggests that PSO can be a useful substitute for ILP for a range of optimization issues, particularly when the number of iterations is not a limiting constraint. We found that the deployment costs tend to stabilize and converge as the number of iterations expands through meticulous experimentation and observation. This result illustrates how crucial it is to optimize the iteration parameter in our deployment operations in order to get the desired cost results. We have shown the promise of the PSO-based technique as a promising method for optimizing the bandwidth as well as computational costs in C-RAN environments.

Overall, our research demonstrates PSO's effectiveness in optimizing C-RAN systems and offers insightful information for the field's future development. In the context of cloud radio access networks, the PSO-based method demonstrates promising performance and provides a flexible and effective approach for dealing with challenging optimization issues.

List of Symbols and Abbreviations

3GPP   3rd Generation Partnership Project

BW   Bandwidth

BBU   Baseband Unit

CAPEX Capital Expenditure

CPRI   Common Public Radio Interface

CPF   Cell-centric Processing Function

C-RAN Cloud RAN

DL   Downlink

D-RAN Distributed RAN

eCPRI Evolved Common Public Radio Interface

eMBB   Enhanced Mobile Broadband

eNB   Evolved NodeB

EPC   Evolved Packer Core

FSI   Functional Split Interface

ILP   Integer Linear Programming

NFV   Network Function Virtualization

NGFI   Next Generation Fronthaul Interface

NGMN Next Generation Mobile Networks

MAC   Media Access Control

mMTC Massive Machine-Type Communication

OPEX   Operational Expenditure

PDCP   Packet Data Convergence Protocol

PF   Processing Function

PUE   Power Usefulness Effectiveness

PSO   Particle Swarm Optimization

QAM   Quadrature Amplitude Modulation

RAN   Radio Access Network

RAU   Radio Aggregation Units

RB   Resource Block

RCC   Radio Cloud Center

RLC   Radio Link Control

RRU   Radio Remote Units

SCF   Small Cell Forum

TCO   Total cost of Owner ship

UL   Uplink

UPF   User-centric Processing Function

URLLC Ultra-Reliable Low-Latency Communications

REFERENCES

[1]   K. Chen and R. Duan, "C-RAN the road towards green RAN," *China Mobile Research Institute, white paper,* vol. 2, 2011.

[2]   F. INC, "The benefits of cloud-ran architecture in mobile network expansion," *FUJITSU NETWORK COMMUNICATIONS INC,* 2014.

[3]   K. Chen *et al.*, "C-RAN: a green RAN framework," in *Green Communications: Theoretical Fundamentals, Algorithms and Applications*: CRC Press, 2013, pp. 279-304.

[4]   S. Matoussi, I. Fajjari, S. Costanzo, N. Aitsaadi, and R. Langar, "A user centric virtual network function orchestration for agile 5G cloud-RAN," in *2018 IEEE International Conference on Communications (ICC),* 2018: IEEE, pp. 1-7.

[5]   S. Matoussi, I. Fajjari, S. Costanzo, N. Aitsaadi, and R. Langar, "5G RAN: Functional split orchestration optimization," *IEEE Journal on Selected Areas in Communications,* vol. 38, no. 7, pp. 1448-1463, 2020.

[6]   M. A. Habibi, M. Nasimi, B. Han, and H. D. Schotten, "A comprehensive survey of RAN architectures toward 5G mobile communication system," *Ieee Access,* vol. 7, pp. 70371-70421, 2019.

[7]   P. Marsch *et al.*, "5G radio access network architecture: Design guidelines and key considerations," *IEEE Communications Magazine,* vol. 54, no. 11, pp. 24-32, 2016.

[8]   A. Checko *et al.*, "Cloud RAN for mobile networks—A technology overview," *IEEE Communications surveys & tutorials,* vol. 17, no. 1, pp. 405-426, 2014.

[9]   M. Labana and W. Hamouda, "Advances in CRAN performance optimization," *IEEE Network,* vol. 35, no. 3, pp. 140-146, 2020.

[10]   N. Mharsi, "Cloud-Radio Access Networks: design, optimization and algorithms," Université Paris-Saclay (ComUE), 2019.

[11]   G. R. Waissi, "Network flows: Theory, algorithms, and applications," ed: JSTOR, 1994.

[12] J. Fang, W. Liu, L. Chen, S. Lauria, A. Miron, and X. Liu, "A survey of algorithms, applications and trends for particle swarm optimization," *International Journal of Network Dynamics and Intelligence,* pp. 24-50, 2023.

[13] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*, 1995: Ieee, pp. 39-43.

[14] A. Alabbasi and C. Cavdar, "Delay-aware green hybrid CRAN," in *2017 15th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt),* 2017: IEEE, pp. 1-7.

[15] J. Liu, S. Zhou, J. Gong, Z. Niu, and S. Xu, "Graph-based framework for flexible baseband function splitting and placement in C-RAN," in *2015 IEEE International Conference on Communications (ICC),* 2015: IEEE, pp. 1958-1963.

[16] A. Checko, A. P. Avramova, M. S. Berger, and H. L. Christiansen, "Evaluating C-RAN fronthaul functional splits in terms of network level energy and cost savings," *Journal of Communications and Networks,* vol. 18, no. 2, pp. 162-172, 2016.

[17] X. Wang, A. Alabbasi, and C. Cavdar, "Interplay of energy and bandwidth consumption in CRAN with optimal function split," in *2017 IEEE International Conference on Communications (ICC),* 2017: IEEE, pp. 1-6.

[18] A. Tzanakaki *et al.,* "5G infrastructures supporting end-user and operational services: The 5G-XHaul architectural perspective," in *2016 IEEE International Conference on Communications Workshops (ICC),* 2016: IEEE, pp. 57-62.

[19] D. Harutyunyan and R. Riggio, "Flex5G: Flexible functional split in 5G networks," *IEEE Transactions on Network and Service Management,* vol. 15, no. 3, pp. 961-975, 2018.

[20] X. Wang, L. Wang, S. E. Elayoubi, A. Conte, B. Mukherjee, and C. Cavdar, "Centralize or distribute? A techno-economic study to design a low-cost cloud radio access network," in *2017 IEEE International Conference on Communications (ICC),* 2017: IEEE, pp. 1-7.

[21] A. Pelekanou, M. Anastasopoulos, A. Tzanakaki, and D. Simeonidou, "Provisioning of 5G services employing machine learning techniques," in *2018 International*

Conference on Optical Network Design and Modeling (ONDM), 2018: IEEE, pp. 200-205.

[22]	Z. Gao *et al.*, "Deep reinforcement learning-based policy for baseband function placement and routing of RAN in 5G and beyond," *Journal of Lightwave Technology,* vol. 40, no. 2, pp. 470-480, 2021.

VITA

NAME	Wai Phyo
DATE OF BIRTH	24 January 1996
PLACE OF BIRTH	Yangon
INSTITUTIONS ATTENDED	Chulalongkorn University
HOME ADDRESS	No.189,ThiriMingalar Street, Insein,Yangon, Myanmar
PUBLICATION	Simulation Software for Functional Split Optimization In 5G Cloud Radio Access Network using Binary Integer Linear Programming: Software Implementation