Resource Allocation in Future Radio Access Networks

Mahdi Sharara

université PARIS-SACLAY

Resource Allocation in Future Radio Access Networks

Allocation des ressources dans les futurs réseaux d'accès radio

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat: Sciences des réseaux, de l'information et de la communication
Graduate School : Informatique et sciences du numérique
Référent: Faculté des sciences d'Orsay

Thèse préparée au **Laboratoire des signaux et systèmes** (Université Paris-Saclay, CNRS, CentraleSupélec), sous la direction de **Véronique VÈQUE**, Professeur, et le co-encadrement de **Sahar HOTEIT**, Maître de Conférences

Thèse soutenue à Paris-Saclay, le 21 Mars 2023, par

Mahdi SHARARA

Composition du jury

Membres du jury avec voix délibérative

Hervé RIVANO Professeur, INSA Lyon	Président
Rami LANGAR Professeur, ETS-Montréal	Rapporteur & Examinateur
Xavier LAGRANGE Professeur, IMT Atlantique / IRISA	Rapporteur & Examinateur
Kinda KHAWAM Maître de Conférences, Université Paris-Saclay	Examinatrice
Nancy PERROT Chercheuse, Orange Labs	Examinatrice

THESE DE DOCTORAT
NNT : 2023UPASG024

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Allocation des ressources dans les futurs réseaux d'accès radio

Mots clés: Cloud-RAN, Open-RAN, Allocation de ressources, Allocation conjointe, Ressources radio, Ressources de calcul

Résumé: Cette thèse considère l'allocation des ressources radio et de calcul dans les futurs réseaux d'accès radio et plus précisément dans les réseaux Cloud-RAN *(Cloud-Radio Access Networks)* ainsi que les réseaux Open-RAN *(Open-Radio Access Networks)*. Dans ces architectures, le traitement en bande de base de plusieurs stations de base est centralisé et virtualisé. Cela permet une meilleure optimisation du réseau et une réduction des dépenses d'investissement et d'exploitation.

Dans la première partie de cette thèse, nous considérons un schéma de coordination entre les ordonnanceurs radio et de calcul. Dans le cas où les ressources de calcul ne sont pas suffisantes, l'ordonnanceur de calcul envoie un retour d'information à l'ordonnanceur radio pour mettre à jour les paramètres radio. Bien que cela réduise le débit radio de l'utilisateur, il garantit que la trame sera traitée au niveau de l'ordonnanceur de calcul. Nous modélisons ce schéma de coordination à l'aide de la programmation linéaire en nombres entiers (ILP) avec comme objectifs de maximiser le débit total ainsi que la satisfaction des utilisateurs. Les résultats montrent la capacité de ce schéma de coordination à améliorer différents paramètres, notamment la réduction du gaspillage de puissance de transmission. Ensuite, nous proposons des heuristiques à faible complexité et nous les testons dans un environnement de services multiples avec des exigences différentes.

Dans la deuxième partie de cette thèse, nous considérons l'allocation conjointe des ressources radio et de calcul. Les ressources radio et de calcul sont allouées conjointement dans le but de minimiser la consommation énergétique. Le problème est modélisé à l'aide de la programmation linéaire mixte en nombres entiers (MILP), et est ensuite comparé à un autre problème MILP ayant comme objectif de maximiser le débit total. Les résultats montrent que l'allocation conjointe des ressources radio et de calcul est plus efficace que l'allocation séquentielle pour minimiser la consommation énergétique. Enfin, nous proposons un algorithme basé sur la théorie de matching *(matching theory)* à faible complexité qui pourra être une alternative pour résoudre le problème MILP à haute complexité.

Dans la dernière partie de cette thèse, nous étudions l'utilisation des outils de l'apprentissage machine (machine learning). Tout d'abord, nous considérons un modèle d'apprentissage profond (deep learning) qui vise à apprendre comment résoudre le problème de coordination ILP, mais en un temps beaucoup plus court. Ensuite, nous considérons un modèle d'apprentissage par renforcement (reinforcement learning) qui vise à allouer des ressources de calcul aux utilisateurs afin de maximiser le profit de l'opérateur.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: Resource Allocation in Future Radio Access Networks

Keywords: Cloud-RAN, Open-RAN, Resource allocation, Joint allocation, Radio resources, Computing resources

Abstract: This dissertation considers radio and computing resource allocation in future radio access networks and more precisely Cloud Radio Access Network (Cloud-RAN) and Open Radio Access Network (Open-RAN). In these architectures, the baseband processing of multiple base stations is centralized and virtualized. This permits better network optimization and allows for saving capital expenditure and operational expenditure.

In the first part, we consider a coordination scheme between radio and computing schedulers. In case the computing resources are not sufficient, the computing scheduler sends feedback to the radio scheduler to update the radio parameters. While this reduces the radio throughput of the user, it guarantees that the frame will be processed at the computing scheduler level. We model this coordination scheme using Integer Linear Programming (ILP) with the objectives of maximizing the total throughput and users' satisfaction. The results demonstrate the ability of this scheme to improve different parameters, including the reduction of wasted transmission power. Then, we propose low-complexity heuristics and we test them in an environment of multiple services with different requirements.

In the second part, we consider the joint radio and computing resource allocation. Radio and computing resources are jointly allocated with the aim of minimizing energy consumption. The problem is modeled as a Mixed Integer Linear Programming Problem (MILP) and is compared to another MILP problem that maximizes the total throughput. The results demonstrate the ability of joint allocation to minimize energy consumption in comparison with the sequential allocation. Finally, we propose a low-complexity matching game-based algorithm that can be an alternative for solving the high-complexity MILP problem.

In the last part, we investigate the usage of machine learning tools. First, we consider a deep learning model that aims to learn how to solve the coordination ILP problem, but with a much shorter time. Then, we consider a reinforcement learning model that aims to allocate computing resources for users to maximize the operator's profit.

Acknowledgments

I would like to express my deepest gratitude to my supervisors, Professors **Sahar Hoteit** and **Véronique Vèque**, for their invaluable guidance, support, and encouragement throughout my research journey. Their expertise and dedication have been instrumental in shaping my work and helping me to achieve my goals.

I would also like to extend my sincere appreciation to the members of the evaluation committee for their willingness to serve in this committee and for their time and effort in reviewing my work: Professors **Hervé Rivano**, **Rami Langar**, **Xavier Lagrange**, **Kinda Khawam**, and **Nancy Perrot**.

I would like to express my gratitude to Professor **Melike Erol-Kantarci** who hosted me at NETCORE lab in uOttawa. The visit allowed me to profit from their expertise in the field of reinforcement learning and mobile networks.

I would like to acknowledge the researchers and their respective teams for their highly valuable comments and suggestions. In order of content exposition: **Patrick Brown** who was at Orange Labs on the coordination between radio and computing schedulers, presented in Chapter 3, **Francessca Fossati** from CentraleSuepelec and **Francessca Bassi** from IRT SystemX on the joint radio and computing resource allocation, presented in Chapter 4, and **Turgay Pamuklu** and **Melike Erol-Kantarci** on reinforcement learning for profit maximization, presented in Chapter 5.

I am grateful for the support and assistance of my colleagues at the Laboratory of Signals and Systems (L2S), who have provided a stimulating and collaborative working environment.

I would also like to express my appreciation to my friends who have been a source of inspiration and motivation throughout this journey.

Lastly, I am eternally grateful to my family for their unwavering love, support, and understanding. This work would not have been possible without their constant encouragement and sacrifices. Without them, this journey would not have been possible.

Contents

List of Figures

List of Tables

Acronyms

1G: First Generation
2G: Second Generation
3G: Third Generation
3GPP: 3rd Generation Partnership Project
4G: Fourth Generation
5G: Fifth Generation
5G-NR: 5G New Radio
AI: Artificial Intelligence
BBU: Base Band Unit
BS: Base Station
BSC: Base Station Controller
BSS: Base Station Subsystem
BTS: Base Transceiver Station
CA: Carrier Aggregation
CAPEX: Capital Expenditure
CDMA: Code Division Multiple Access
C-RAN: Cloud RAN
CU: Central Unit
D-RAN: Distributed-RAN
DL: Deep Learning
DU: Distributed Unit (in 2G: Digital Unit)
EDGE: Enhanced Data Rate for GSM Evolution
eMBB: enhanced Mobile Broad Band
eUTRAN: evolved Universal Mobile Telecommunications System
FDMA: Frequency Division Multiple Access
GERAN: GSM EDGE RAN
gNB: Next-generation NodeB
GPRS: General Packet Radio Service
GSM: Global System for Mobile communication
GT: Game Theory HSPA: High-Speed Packet Access

ILP: Integer Linear Programming
ICT: Information and Communication Technology
ITU: International Telecommunication Unit
LTE: Long Term Evolution
LTE-A: LTE-Advanced MAC: Medium Access Control Layer
MCS: Modulation and Coding Scheme
MEC: Multi-Access Edge Computing
MILP: Mixed Integer Linear Programming
MINLP: Mixed Integer non-Linear Programming
MIMO: Multiple-Input-Multiple-Outputs
ML: Machine Learning
mMTC: massive Machine Type Communication
MS: Mobile Station
NFV: Network Functions Virtualization
NG-RAN: Next Generation RAN
NSS: Network Switching Subsystem OAI: Open Air Interface
O-CU: Open Central Unit
O-DU: Open Distributed Unit
OFDMA: Orthogonal Frequency Division Multiple Access
O-RAN: Open-RAN
OPEX: Operational Expenditure
PCU: Packet Control Unit
PDCP: Packet Data Convergence Protocol
PHY: Physical Layer
PSTN: Public Switched Telephone Network
RAN: Radio Access Network
RAT: Radio Access Technology
REC: Radio Equipment Controller
RB: Resource Block
RF: Radio Frequency
RIC: Radio Intelligent Controller
RL: Reinforcement Learning
RLC: Radio Link Control
RNC: Radio Network Controller
RNS: Radio Network Subsystem
RRC: Radio Resource Control
RRH: Radio Remote Head
RRM: Radio Resource Management
SDN: Software Defined Networking
SDR: Software Defined Radio
SISO: Single-Input-Single-Output
TBS: Transport Block Size (TBS)

TDMA: Time Division Multiple Access
TTI: Transmission Time Interval
UMTS: Universal Mobile Telecommunications System
UTRAN: Universal Mobile Telecommunications System
URLLC: Ultra-Reliable Low Latency
USRP: Universal Software Radio Peripheral
VM: Virtual Machine
VOIP: Voice over Internet Protocol

Résumé long en français

Dans le monde entier, la demande de données augmente massivement. Selon [9], le trafic mobile mondial devrait dépasser 350 EB (exabyte) par mois en 2027. La figure 1.1 montre cette tendance. De plus, le trafic devrait atteindre 5016 EB/mois en 2030 selon [10]. Pour répondre à cette demande croissante, les opérateurs de téléphonie mobile doivent continuer à moderniser leurs réseaux. Au cours des dernières décennies, les réseaux mobiles ont énormément évolué. Après les communications analogiques de voix dans la première génération (1G), les communications cellulaires sont passées entièrement en numérique dans la deuxième génération (2G), posant ainsi la première pierre de la migration vers les communications orientées données à l'aide du GPRS(*(General Packet Radio Service)*. Les communications mobiles ont dû satisfaire la demande croissante de trafic de voix et de données en améliorant la technologie de transmission, passant de l'accès multiple par répartition en fréquence (Frequency Division Multiple Access - FDMA) à l'accès multiple par répartition dans le temps (Time Division Multiple Access - TDMA) dans la deuxième génération, puis à l'accès multiple par répartition en code à large bande (Wideband Code Division Multiple Access - WCDMA) dans la troisième génération (3G). Cependant, la demande massivement croissante de débits plus élevés et de plus faibles latences a nécessité que la quatrième génération (4G) adopte l'accès multiple par répartition en fréquence orthogonale (Orthogonal Frequency Division Multiple Access - OFDMA), passant ainsi à une communication entièrement commutée par paquets pour les données et la voix (par exemple, la voix sur IP (VoIP)) [11]. Malgré les améliorations apportées à l'architecture des réseaux mobiles, la demande en termes de débit continue de croître dans la cinquième génération (5G) et au-delà. Diverses améliorations ont été apportées au niveau radio dans la 5G pour répondre aux exigences de débit très élevé pour les services eMBB *(enhanced Mobile BroadBand)*. Ces améliorations impliquent aussi la réduction de la latence et l'augmentation de la fiabilité pour répondre aux exigences des services de communications ultra-fiables à faible latence (Ultra Reliable Low Latency Communication- URLLC).

Les techniques traditionnelles de partage de la bande passante ne sont pas suffisantes seules pour faire face à l'augmentation continue du trafic, et les régulateurs des réseaux mobiles font évoluer l'architecture des réseaux mobiles pour optimiser l'allocation des

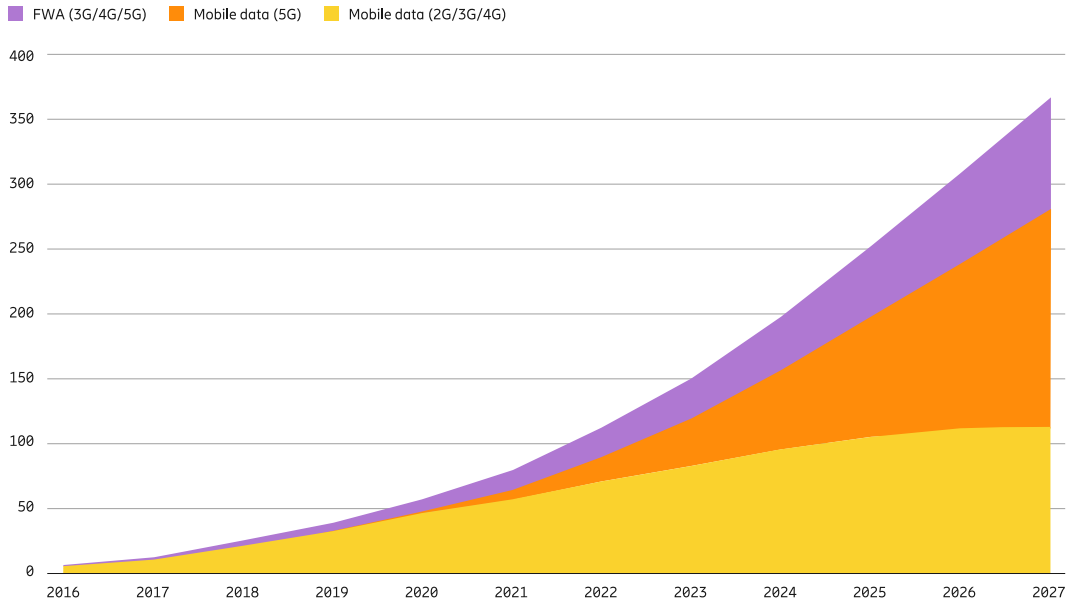


Figure 1: Prévisions de trafic selon [9] dans ExaBytes EB

ressources. En effet, les réseaux mobiles actuels se composent de deux parties principales: le réseau d'accès radio (Radio Access Network-RAN) et le réseau coeur (Core Network-CN). Le RAN est composé de l'équipement utilisateur (UE) et de la station de base (BS), et il est responsable de la communication sur les fréquences radio. Le CN est chargé de coordonner le réseau, de connecter les différents composants du RAN les uns aux autres et de fournir une connexion Internet.

Dans la quatrième génération, la station de base, appelée eNodeB, est composée de deux éléments principaux déployés sur le site cellulaire : l'unité radio distante (Radio Remote Head-RRH) et l'unité de bande de base (Base Band Unit - BBU). La RRH est responsable de l'exécution des fonctions de radiofréquence, et la BBU exécute toutes les autres fonctions de bande de base (c'est-à-dire les fonctions des couches physique (PHYsical - PHY), de contrôle d'accès au support (Multiple Access Control - MAC), de contrôle des ressources radio (Radio Resource Control - RRC), et autres). Les deux unités existaient dans chaque station de base en tant que composants physiques. Cette architecture est appelée réseau d'accès radio distribué (Distributed Radio Access Network - D-RAN).

Par conséquent, l'architecture RAN 4G est statique ce qui la rend moins flexible et moins efficace. En effet, alors que le trafic et la demande de débit varient tout au long de la journée en temps et en espace, la répartition des ressources radio ne change pas. Par exemple, un eNodeB peut être peu utilisé à un moment de la journée alors qu'un autre est surchargé. Durant les heures de travail, les zones d'activités sont très chargées alors que les zones pavillonnaires ne génèrent pas de trafic. Par conséquent, l'architecture statique laisse beaucoup de ressources BBU inutilisées lorsque la demande est faible dans certaines zones alors que d'autres zones sont débordées. Le RAN actuel offre donc aux

opérateurs peu de flexibilité et entraîne une augmentation des dépenses d'investissement (Capital Expenditure - CAPEX) et des dépenses d'exploitation (Operational Expenditure - OPEX). Toutes ces limitations ont motivé une profonde transformation de l'architecture RAN.

Stimulée par le succès de la virtualisation dans différents domaines des technologies de l'information et de la communication (TIC) [11], [12], l'architecture RAN évolue désormais vers une architecture centralisée, virtualisée et logicielle. Pour répondre aux limites posées par le D-RAN, un paradigme récent connu sous le nom de réseau d'accès radio en nuage (Cloud Radio Access Network - Cloud-RAN) a été proposé. Il centralise le traitement des fonctions de traitement des signaux bande de base, de sorte que les BBU de plusieurs stations de base sont centralisées et exécutées en tant que machines virtuelles (Vitrual Machine - VM) dans un nuage, connu sous le nom de BBU Pool [13].

Dans le Cloud-RAN, grâce à la virtualisation, le RAN gagne en flexibilité et en adaptativité. Ainsi, il est plus facile pour les opérateurs d'approvisionner différentes zones en ressources selon les besoins instantanés, ou d'allumer/éteindre dynamiquement les stations de base en fonction de la demande. Par conséquent, les opérateurs vont optimiser le déploiement du RAN en réduisant le nombre de BBU et ainsi réduire leurs coûts opérationnels. De plus, la centralisation et la virtualisation offrent la possibilité d'améliorer les performances du réseau, de mieux gérer les interférences et de réduire la consommation d'énergie [11], [14].

Récemment, le réseau d'accès radio ouvert (Open RAN - O-RAN) a vu le jour. Normalisé par l'alliance O-RAN, il repose sur quatre principes fondamentaux : virtualisation, désagrégation, intelligence et interfaces ouvertes. Le réseau est virtualisé et désagrégé en plusieurs composants ayant des fonctions différentes. De plus, O-RAN normalise les contrôleurs radio intelligents (Radio Intelligent Controller - RIC) ayant des échelles de temps et des boucles de contrôle différentes. Les RIC sont capables d'assurer la coordination entre les différents nœuds du réseau. De plus, les RIC prennent en charge la formation et le déploiement d'algorithmes d'apprentissage automatique. De surcroit, O-RAN est basé sur l'ouverture des différentes interfaces afin d'assurer l'interopérabilité entre les nœuds du réseau qui peuvent être fournis par différents fournisseurs. Cela encourage l'innovation parmi les acteurs existants et permet à davantage d'acteurs d'entrer sur le marché. La concurrence se traduira par des économies de CAPEX et d'OPEX [15].

Tirer parti et exploiter les avantages du Cloud-RAN et de l'Open-RAN n'est pas une tâche aisée car elle se heurte à de nombreux obstacles [11]. Si l'on se concentre d'abord sur le Cloud-RAN entièrement centralisé, l'optimisation centralisée s'accompagne d'une complexité temporelle accrue. Par exemple, les exigences de faible délai dans les réseaux mobiles peuvent rendre le Cloud-RAN inefficace. Il est donc nécessaire d'envisager des algorithmes qui peuvent produire des résultats dans le temps requis. En plus, la centralisation du traitement de la bande de base met à rude épreuve les liaisons de fronthaul qui relient les RRH et les BBU dans le Cloud-RAN. Dans l'architecture traditionnelle, de courtes liaisons de fronthaul sont nécessaires pour connecter un RRH à la BBU, étant donné que les deux se trouvent dans le même site. Cependant, la centralisation nécessite

le déploiement de câbles beaucoup plus longs pour connecter les RRHs distants au pool de BBU centralisés. Ce qui risque d'augmenter le CAPEX. Lorsque le signal parcourt de longues distances par le biais de la fibre ou des liaisons hertziennes, il est plus difficile de respecter les exigences en matière de délai. Cela a ouvert la voie au concept de divisions fonctionnelles (functional split), où toutes les fonctions de bande de base ne sont pas centralisées. Certaines de ces fonctions sont mises en œuvre sur le RRH, tandis que d'autres sont exécutées dans le pool de BBU. Cela réduit l'overhead à transmettre sur les liaisons fronthaul.L'exploitation de l'architecture Cloud-RAN pour optimiser différentes métriques peut être traitée de manière optimale en résolvant des problèmes d'optimisation mathématique. Cependant, ces problèmes sont très complexes, et il faut recourir à des algorithmes sous-optimaux moins complexes qui peuvent améliorer les performances par rapport au RAN traditionnel. Notre objectif dans cette thèse est de concevoir des algorithmes à faible complexité capables d'améliorer les performances par rapport au RAN existant, tout en vérifiant que la complexité supplémentaire soit justifiée. En d'autres termes, il peut ne pas être avantageux d'envisager une optimisation centralisée pour certains problèmes du Cloud-RAN lorsque les solutions centralisées apportent des gains négligeables ou nuls. De telles solutions auront une complexité temporelle plus élevée et consommeront plus de ressources de calcul sans avoir de valeur ajoutée. Dans ce contexte, l'utilisation de modèles d'optimisation mathématique est un outil utile pour décider si une solution centralisée est bénéfique. Dans l'affirmative, l'étape suivante consisterait à trouver des algorithmes efficaces et moins complexes qui produisent des solutions proches des solutions optimales, obtenues par la résolution optimale de problèmes d'optimisation mathématique.

Dans la 5G et au-delà, différents services aux contraintes différentes devront coexister. Par exemple, les services Enhanced Mobile BroadBand (eMBB) et Ultra Reliable Low Latency (URLLC) ont des exigences différentes ; le premier vise à fournir des débits de données élevés, tandis que le second vise à transmettre des données avec un retard minimal et de faibles taux de perte de paquets. Par exemple, les services URLLC sont nécessaires pour réaliser des communications véhiculaires pour les véhicules autonomes [16] ou pour réaliser l'automatisation des usines [17]. Les services URLLC et eMBB pourraient être logiquement isolés tout en fonctionnant sur la même radio physique et le même matériel de calcul. Un tel mécanisme est connu sous le nom de *Network Slicing (NS)* [10]. Avec son architecture centralisée et virtualisée, le Cloud-RAN est un élément clé du network slicing.

Dans notre travail, nous nous sommes concentrés sur l'allocation des ressources radio et de calcul dans le futur RAN. Ces ressources comprennent les blocs de ressources radio (Resource Blocks - RB), le schéma de modulation et de codage (Modulation and Coding Scheme - MCS), la puissance, les processeurs, la mémoire, etc. Les architectures du Cloud-RAN et de l'Open-RAN permettent une plus grande coopération et une allocation conjointe des ressources radio et de calcul. Par exemple, l'allocation des ressources radio affecte les besoins en ressources de calcul, étant donné que le temps de traitement des trames dépend du nombre de RB et du MCS. La centralisation et la virtualisation

permettent de gérer conjointement les paramètres de nombreuses stations de base, ce qui devrait contribuer à optimiser les performances du réseau. L'allocation des ressources a différents objectifs tels que l'optimisation du débit, du délai, de la consommation d'énergie, du CAPEX, de l'OPEX, des profits, de la qualité de service (Quality of Service - QoS), etc.

De nombreux problèmes d'allocation de ressources peuvent être modélisés comme des problèmes d'optimisation mathématique. Cependant, nombre de ces problèmes peuvent être difficiles à résoudre en temps réel en raison de leur grande complexité. Par exemple, la résolution d'un problème de programmation linéaire en nombres entiers (Integer Linear Programming - ILP) ou d'un problème de programmation linéaire mixte en nombres entiers (Mixed Integer Linear Programming - MILP) est NP-hard (Non-deterministic Polynomial-time hardness)[18]. Si ces problèmes peuvent être utiles pour démontrer l'efficacité et les améliorations d'une idée, leur grande complexité les rend impropres à une mise en œuvre pratique. Il est donc nécessaire de trouver des solutions de rechange peu complexes.

Pour concevoir des algorithmes à faible complexité, différents outils peuvent être utilisés. Parfois, de simples heuristiques de base peuvent être utilisées. Sinon, d'autres outils tels que la théorie des jeux (Game Theory - GT) et l'apprentissage automatique (Machine Learning - ML) peuvent être utilisés pour trouver des solutions efficaces. La théorie des jeux a été initialement utilisé pour modéliser les interactions entre les acteurs de l'économie et de la politique [19]. Les solutions sont atteintes lorsque les joueurs arrivent à un équilibre où il devient non bénéfique pour chaque joueur de modifier unilatéralement son action. Les cadres de la théorie des jeux sont adaptés à l'optimisation distribuée où il existe plusieurs décideurs. Par ailleurs, l'apprentissage automatique a récemment fait l'objet d'une grande attention [20], [21]. L'apprentissage automatique et, plus précisément, l'apprentissage profond (Deep Learning - DL) ont permis des améliorations significatives, notamment dans les domaines de la vision par ordinateur et du traitement du langage naturel. Ce succès a donc attiré l'attention de chercheurs de différents domaines. L'apprentissage profond a la capacité d'apprendre des schémas ou relations cachés et de produire des prédictions très précises. Il est également capable d'approximer des fonctions complexes et de produire des résultats en un temps plus court. L'apprentissage profond est donc aussi prometteur dans les réseaux de télécommunication, notamment en raison de la dépendance accrue à l'égard des applications à faible latence. Dans ce cas, DL peut exploiter sa structure parallélisée pour s'exécuter sur plusieurs GPU en un temps réduit. En outre, l'apprentissage par renforcement (Reinforcement Learning - RL) est un schéma dans lequel un agent apprend la qualité d'une action, compte tenu d'un état ou d'une observation, en interagissant avec l'environnement et en recevant des récompenses ou des pénalités en conséquence. L'apprentissage par renforcement convient donc aux applications de contrôle dans les réseaux mobiles. Les agents peuvent être des utilisateurs, des stations de base, des opérateurs mobiles, etc., et ils peuvent apprendre les meilleures actions en interagissant avec l'environnement. Si l'environnement est dynamique, l'agent RL doit s'adapter à sa nature dynamique et

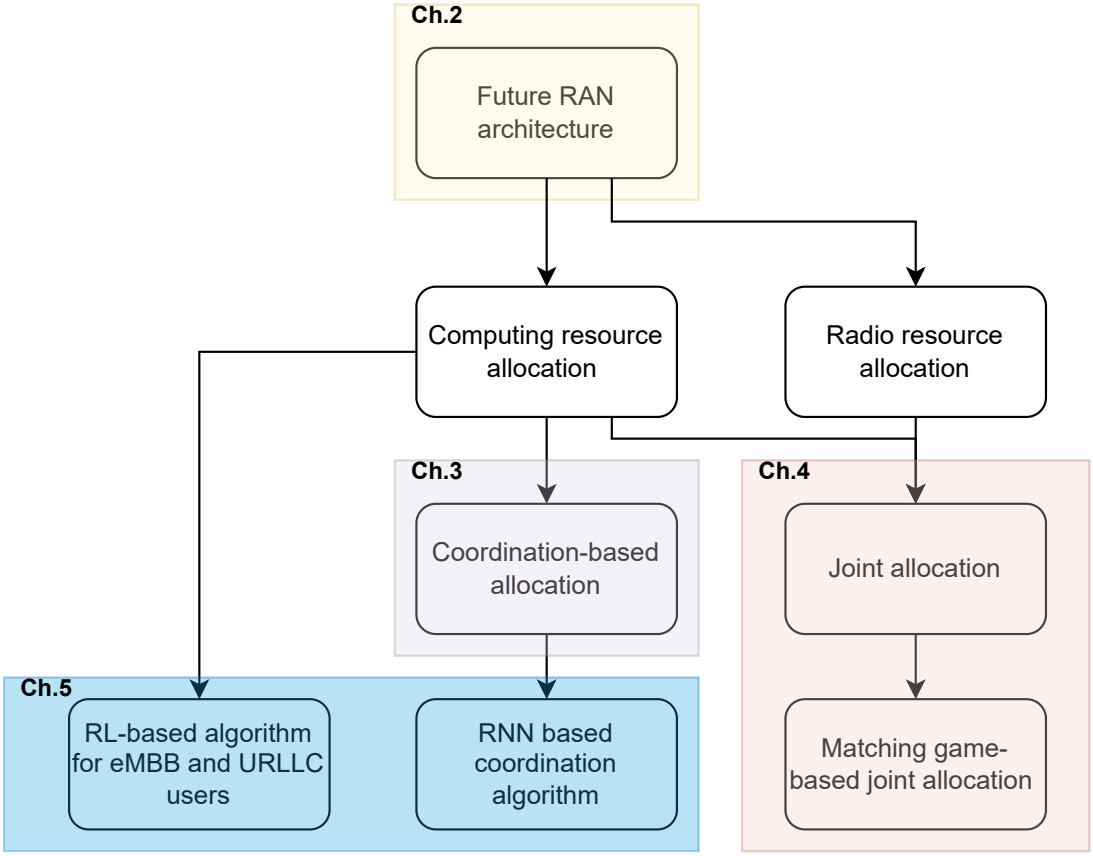


Figure 2: La structure de la dissertation

sélectionner les meilleures actions en conséquence.

L'objectif de cette thèse est de proposer de nouveaux algorithmes d'allocation de ressources adaptés aux architectures Cloud-RAN ou Open-RAN. En particulier, nous analysons les applications potentielles qui peuvent bénéficier de l'architecture centralisée et virtualisée Cloud-RAN et Open-RAN. Nous nous appuyons sur l'utilisation de problèmes d'optimisation mathématique. De plus, nous proposons des algorithmes peu complexes qui peuvent produire des solutions proches des solutions optimales des problèmes d'optimisation.

Structure de la dissertation

La structure de la thèse est présentée dans la Fig. 2. Cette thèse est composée de cinq chapitres à l'exclusion de cette introduction présentée ci-dessous. De plus, deux annexes sur l'apprentissage profond et l'apprentissage par renforcement sont fournies.

Le Chapitre 2 présente les architectures Cloud-RAN et O-RAN, leurs avantages et les défis auxquels elles sont confrontées. Ensuite, divers travaux de recherche sur

l'allocation des ressources radio et de calcul sont abordés. Ce chapitre introduit également les outils de la théorie des jeux et de l'apprentissage automatique utilisés dans la communication mobile. Il s'agit notamment des jeux d'appariement, de l'apprentissage profond et de l'apprentissage par renforcement.

Le Chapitre 3 considère un schéma de coordination entre les ordonnanceurs d'allocation des ressources radio et de calcul pour allouer les ressources de calcul dans le Cloud-RAN. Étant donné que les ressources de calcul requises dépendent des paramètres radio (par exemple, le schéma de modulation et de codage (MCS) et le nombre de blocs de ressources (RB)), et compte tenu du scénario de ressources de calcul limitées, nous proposons et analysons un schéma de coordination qui permet un retour d'information entre les ordonnanceurs radio et de calcul afin d'adapter l'allocation radio à la disponibilité des ressources de calcul.

Le Chapitre 4 porte sur une allocation conjointe des ressources radio et de calcul dans le Cloud-RAN et analyse son impact sur la consommation d'énergie à l'aide de la programmation linéaire mixte en nombres entiers (MILP). L'objectif est de définir les limites dans lesquelles l'allocation conjointe est bénéfique et de trouver le moment où cela n'est plus avantageux. En complément, nous proposons un algorithme de faible complexité basé sur un jeu d'appariement qui vise à produire des solutions proches de la solution optimale du problème MILP.

Le Chapitre 5 examine le potentiel de l'utilisation de l'apprentissage automatique dans le futur RAN, et plus précisément dans Open-RAN. Deux applications sont envisagées : la première utilise un réseau neuronal récurrent (RRN) pour apprendre à se rapprocher du solveur de programmation linéaire en nombres entiers (ILP) qui applique le schéma de coordination du chapitre 3. La deuxième application utilise l'apprentissage par renforcement avec la méthode Policy-Gradient pour allouer des ressources de calcul dans un scénario multiservice composé de deux services aux exigences hétérogènes, l'eMBB (Enhanced Mobile BroadBand) et l'URLLC (Ultra-Reliable Low-Latency Communication).

Le chapitre 6 conclut notre travail et fournit des perspectives pour l'avenir vers une allocation hiérarchique et désagrégée des ressources du réseau d'accès radio ouvert (O-RAN).

L'annexe A présente brièvement les bases de l'apprentissage automatique supervisé, des réseaux neuronaux et de l'apprentissage profond.

L'annexe B présente brièvement les bases de l'apprentissage par renforcement, y compris ses éléments, l'apprentissage basé sur la valeur et l'apprentissage par "policy-gradient".

Publications

Les travaux de cette thèse ont conduit aux publications suivantes:

Article de journaux publié

- M. Sharara, S. Hoteit, P. Brown, and V. Vèque, "On Coordinated Scheduling of Radio and Computing Resources in Cloud-RAN," *IEEE Transactions on Network and Service Management*, 2022

Articles de conférences publiés

- M. Sharara, S. Hoteit, and V. Vèque, "Reinforcement Learning for Profit Maximization in Open-RAN," in *Accepted to 2023 IEEE/IFIP International Conference on Network Operations and Management Symposium (NOMS)*, 2023

- M. Sharara, T. Pamuklu, S. Hoteit, V. Vèque, and M. Erol-Kantarci, "Policy-Gradient-Based Reinforcement Learning for Computing Resources Allocation in O-RAN," in *2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*, 2022

- M. Sharara, S. Hoteit, V. Vèque, and F. Bassi, "Minimizing Power Consumption by Joint Radio and Computing Resource Allocation in Cloud-Ran," in *2022 IEEE Symposium on Computers and Communications (ISCC)*, 2022

- M. Sharara, S. Hoteit, and V. Vèque, "A Recurrent Neural Network Based Approach for Coordinating Radio and Computing Resources Allocation in Cloud-RAN," in *2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*

- M. Sharara, S. Hoteit, P. Brown, and V. Vèque, "Coordination between Radio and Computing Schedulers in Cloud-RAN," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2021

- M. Sharara, S. Hoteit, P. Brown, and V. Vèque, "Coordination de l'ordonnancement radio et de calcul dans Cloud-RAN," in *ALGOTEL 2021 - 23èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, La Rochelle, France, 2021

Article soumis

- M. Sharara, F. Fossati, S. Hoteit, V. Vèque, and F. Bassi, "Minimizing Energy Consumption by Joint Radio and Computing Resource Allocation in Cloud-Ran," *Submitted to the Journal of Computer Networks*,

Chapter 1

Introduction

Worldwide, the demand for data is massively growing. According to [9], the global mobile traffic is expected to exceed 350 EB (exabyte) per month in 2027. The trend is shown in Fig. 1.1. Furthermore, the traffic is expected to reach 5016 EB/month in 2030 according to [10]. Hence, mobile operators should keep upgrading their networks to satisfy the growing demands. Over the past few decades, mobile networks have tremendously evolved. Starting with analog voice communication in the first Generation, (1G), cellular communications switched to digital communication in the second Generation (2G), putting the foundation stone for migrating towards data-oriented communication using the Generalized Packet Radio Service (GPRS). Mobile communications had to satisfy the growing demand for voice and data traffic by upgrading the transmission technology, moving from Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA) in 2G to Wideband Code Division Multiple Access (WCDMA) in the third Generation (3G). However, the massively growing demand for higher throughput and lower latency required that the fourth Generation (4G) adopts Orthogonal Frequency-Division Multiple Access (OFDMA), moving to fully packet-switched communication for both data and voice communication (e.g., Voice over IP (VoIP)) [11]. Despite the improvements in mobile network architecture, the demand continues to grow in the fifth Generation (5G) and beyond. Various enhancements have been done at the radio level in 5G to respond to the very high throughput requirement for enhanced Mobile Broadband (eMBB) services. Additionally, these enhancements involve reducing latency and increasing reliability to satisfy the requirements of Ultra Reliable Low Latency Communication (URLLC). Moreover, massive Machine Type Communication (mMTC) aims at providing connectivity for a massive amount of machines concentrated in one area. Initially considered for 5G in [22], 5G ended up focusing only on eMBB and URLLC.

As traditional bandwidth sharing techniques alone are not sufficient to cope with the continuous increase in traffic, mobile network regulators are evolving the architecture of mobile networks to optimize resource allocation. Current mobile networks consist of two main parts: the Radio Access Network (RAN) and the Core Network (CN). The RAN consists of the User Equipment (UE) and the Base Station (BS), and it is responsible

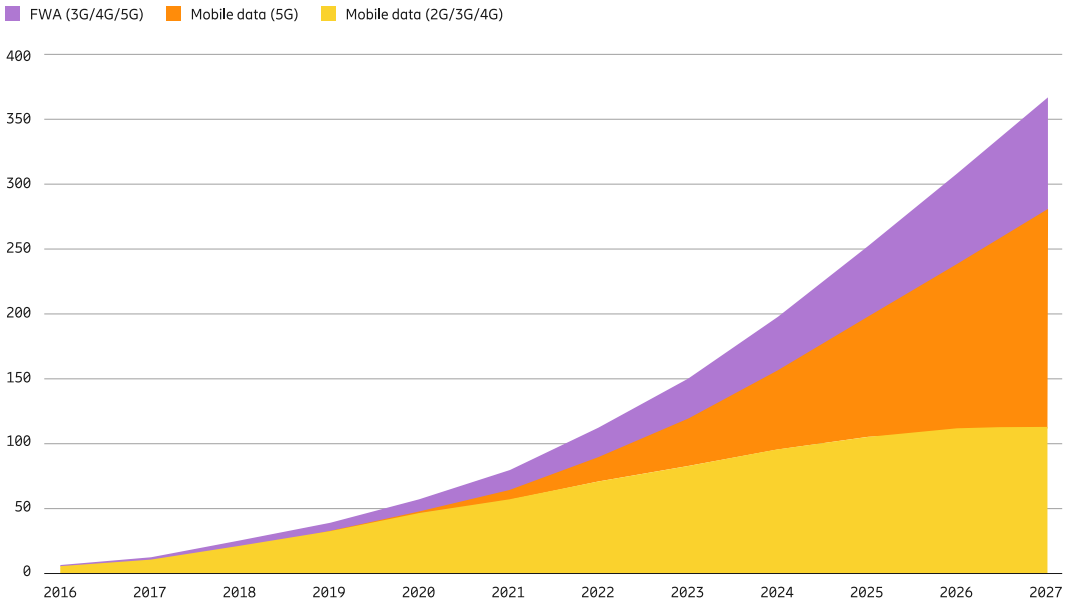


Figure 1.1: Traffic forecast according to [9] in ExaBytes EB. FWA stands for Fixed Wireless Access

for communication over radio frequencies. The CN is responsible for coordinating the network, connecting the different components of the RAN to each other, and providing Internet connection.

In the fourth generation, the base station, known as eNodeB, is composed of two main components deployed on the cellular site; the Radio Remote Head (RRH) and the Base Band Unit (BBU). The RRH is responsible for executing radio frequency functions, and the BBU executes all other baseband functions (i.e., functions of the Physical (PHY), Medium Access Control (MAC), Radio Resource Control (RRC) layers, and above). Both units existed in every base station as physical components. This is called Distributed RAN (D-RAN) architecture. In the 4G architecture, the D-RAN architecture lacks flexibility and efficiency. Despite that the traffic and the demand vary throughout the day temporally and spatially, the provisioned resources can't be dynamically and flexibly altered. During working hours in the morning, the traffic is mainly generated in business areas. In the evening, the traffic shifts to residential areas. This results in unbalanced traffic where some base stations at a specific time in a specific area have a high demand while other base stations are underutilized. The static architecture leaves a lot of BBU resources unused when the demand gets low, while the BBUs in other areas are overwhelmed. The legacy RAN restricts the options for operators as it is not scalable nor flexible, resulting in increased Capital Expenditure (CAPEX) and Operation Expenditure (OPEX). All these limitations have driven the RAN architecture into a profound transformation [11].

Driven by the success of virtualization in different fields in Information and Communication Technology (ICT) [11], [12], the RAN architecture is moving towards a cen-

tralized, virtualized, and softwarized architecture. To respond to the limitations posed by D-RAN, a recent paradigm known as Cloud Radio Access Network (Cloud-RAN) has been proposed. It centralizes the processing of the baseband functions, such that the BBUs of multiple base stations are centralized and run as Virtual Machines (VMs) in a cloud computing center known as the BBU Pool [13].

In Cloud-RAN, the RAN gains flexibility and adaptability thanks to virtualization. Thus, it is easier for operators to redistribute resources to different areas according to instantaneous needs, release resources when the demand is lower, and dynamically turn base stations on/off depending on the demand. Therefore, operators will optimize the deployment of the RAN by reducing the number of BBUs. As a result, the deployment and operational costs are reduced. In addition to reducing CAPEX and OPEX, centralization and virtualization provide the opportunity to improve network performance, better manage interference, and reduce power consumption [11], [14].

Recently, Open Radio Access Network (Open-RAN) has seen the light of day [15]. Standarized by O-RAN alliance, it is based on four foundational principles: virtualization, disaggregation, intelligence, and open interfaces. The network is virtualized and further disaggregated into multiple components having different functions. Also, O-RAN standardizes the Radio Intelligent Controllers (RICs) having different time scales and control loops. The RICs are able to coordinate between the different nodes in the network. Additionally, the RICs support training and deploying machine learning algorithms. Moreover, O-RAN is based on opening the different interfaces. The aim is to ensure interoperability between network nodes that can be provided by different vendors. This encourages innovation among existing players and allows more players into the market. The competition will lead to more CAPEX and OPEX savings [15].

Leveraging and exploiting the benefits of Cloud-RAN and Open-RAN is not a straightforward task and encounters many obstacles [11]. Focusing first on the fully centralized Cloud-RAN, centralized optimization comes with increased time complexity. For instance, low delay requirements in mobile networks may render Cloud-RAN inefficient. Hence, it is obligatory to consider algorithms that can output results in the required amount of time. In addition, centralizing the baseband processing puts a lot of strain on the fronthaul links that connect the RRHs and BBUs in Cloud-RAN. In the legacy architecture, short fronthaul links are required to connect an RRH to the BBU, given that both exist in the same remote site. However, centralization requires deploying much longer cables to connect the remote RRHs to the centralized BBU pool. This would increase CAPEX. Also, not violating the delay requirements will be challenging given that the signal travels for long distances through the fronthaul fiber or microwave links. This has paved the way for the concept of functional splits, where not all baseband functions are centralized. Some of these functions are implemented in the RRH, while others are executed in the BBU pool. This reduces the overhead to be transmitted over the fronthaul links.

Moreover, exploiting the Cloud-RAN architecture to optimize different metrics can be optimally handled by solving mathematical optimization problems. However, such prob-

lems have high complexity, and it is indispensable to resort to low-complexity algorithms that can improve performance compared to legacy RAN. In addition, it is important to ensure that the added complexity is worthwhile. In other words, it may not be beneficial to consider centralized optimization for some problems in Cloud-RAN when the centralized solutions bring negligible or null gains. Such solutions will have higher time complexity and will consume more computing resources without having added value. In this context, using mathematical optimization models would be a useful tool to decide whether a centralized solution is beneficial. If yes, the next step would be to find efficient lower-complexity algorithms that output solutions close to the optimal ones, yielded by optimally solving mathematical optimization problems.

In 5G and beyond, different services are expected to coexist. For example, enhanced Mobile BroadBand (eMBB) and Ultra Reliable Low Latency (URLLC) services have different requirements; the former aims at providing high data rates, while the latter aims at transmitting data with minimal delay and low packet loss rates. For example, URLLC services are needed to realize vehicular communications for autonomous vehicles [16] or to realize factory automation [17]. URLLC and eMBB services could be logically isolated while running on the same physical radio and computational hardware. Such a mechanism is known as Network Slicing (NS) [10]. With its centralized and virtualized architecture, Cloud-RAN is a key enabler of network slicing.

In our work, we have focused on radio and computing resource allocation in future RAN. These resources include the Resource Blocks (RBs), Modulation and Coding Scheme (MCS), power, CPUs, memory, etc. The architecture of Cloud-RAN and Open-RAN permits more cooperation and joint allocation of radio and computing resources. For example, the radio resource allocation affects the computing resource requirement, given that the processing time of frames depend on the number of RBs and the MCS [23]. Centralization and virtualization allow for jointly managing the parameters of many base stations, and this should help optimize the network performance. Resource allocation has different objectives; these include optimizing throughput, delay, power consumption, CAPEX, OPEX, profits, Quality Of Service (QoS), etc.

Many resource allocation problems can be modeled as mathematical optimization problems. However, many of these problems can be hard to solve in real-time due to their high complexity. For example, solving an Integer Linear Programming (ILP) problem or a Mixed Integer Linear Programming (MILP) problem is NP-hard [18]. While optimization problems could be useful to demonstrate the efficiency and the improvements of an idea, their high-complexity makes them non-suitable for practical implementation. Hence, it is necessary to come up with low-complexity alternatives.

Different frameworks can be used to devise low-complexity algorithms. Sometimes, simple baseline heuristics could be used. Otherwise, frameworks such as Game Theory (GT) and Machine Learning (ML) could be used to come up with efficient solutions. Game theory is a framework initially used to model interactions between players in the economy and politics [19]. Solutions are reached when players arrive at an equilibrium where it becomes non-beneficial for each player to unilaterally modify its action. Game

theoretical frameworks are suitable for distributed optimization where multiple selfish decision-makers exist. Moreover, Machine learning has gained a lot of attention recently [20], [21]. Machine learning and, precisely, deep learning have scored significant improvements, especially in the fields of computer vision and Natural Language Processing (NLP). Such success has grasped the attention of researchers from different fields. Deep learning has the ability to learn hidden patterns or relations and output highly accurate predictions. It is also able to approximate complex functions and produce results in a shorter amount of time. Thus DL shows potential in telecommunication networks, especially with increased reliance on low-latency applications. In such a case, DL can exploit its parallelized structure to run over multiple GPUs in shorter period of time. Moreover, Reinforcement Learning is a framework where an agent learns the quality of doing an action, given a state or observation, by interacting with the environment and receiving rewards or penalties accordingly. This makes RL suitable for control applications in mobile networks.

The purpose of this dissertation is to propose novel resource allocation algorithms adapted for either Cloud-RAN or Open-RAN architectures. In particular, we analyze the advantages of the cooperation between the radio resource scheduler and the computing resources scheduler. We rely on mathematical optimization problems. Additionally, we propose low-complexity algorithms that output solutions close to the optimal solutions of the optimization problems.

1.1 Structure of the Dissertation

This dissertation is composed of five chapters excluding this introduction as presented below. Additionally, two appendixes about deep learning and reinforcement learning are provided. The structure of the dissertation is presented in Fig. 1.2.

Chapter 2 surveys the architecture of Cloud-RAN and Open-RAN, the advantages, and the challenges they face. Then various research works on radio and computing resource allocation are discussed. The chapter also considers game-theoretic and machine learning tools used in mobile communication. These include matching games, deep learning, and reinforcement learning.

Chapter 3 considers a coordination scheme between radio and computing resource allocation schedulers to allocate computing resources in Cloud-RAN. Given that the required computing resources depend on the radio parameters (e.g. Modulation and Coding Scheme (MCS) and number of Resource Blocks (RB)), and considering the scenario of limited computing resources, we propose and analyze a coordination scheme that permits feedback between radio and computing schedulers in order to adapt the radio allocation to the availability of computing resources. Additionally, we propose low-complexity heuristics that perform close to the optimal ILP solutions, and we test their performance in a multi-services scenario.

Chapter 4 deals with joint allocation of radio and computing resources in Cloud-RAN and analyzes its impact on energy consumption using Mixed Integer Linear Programming

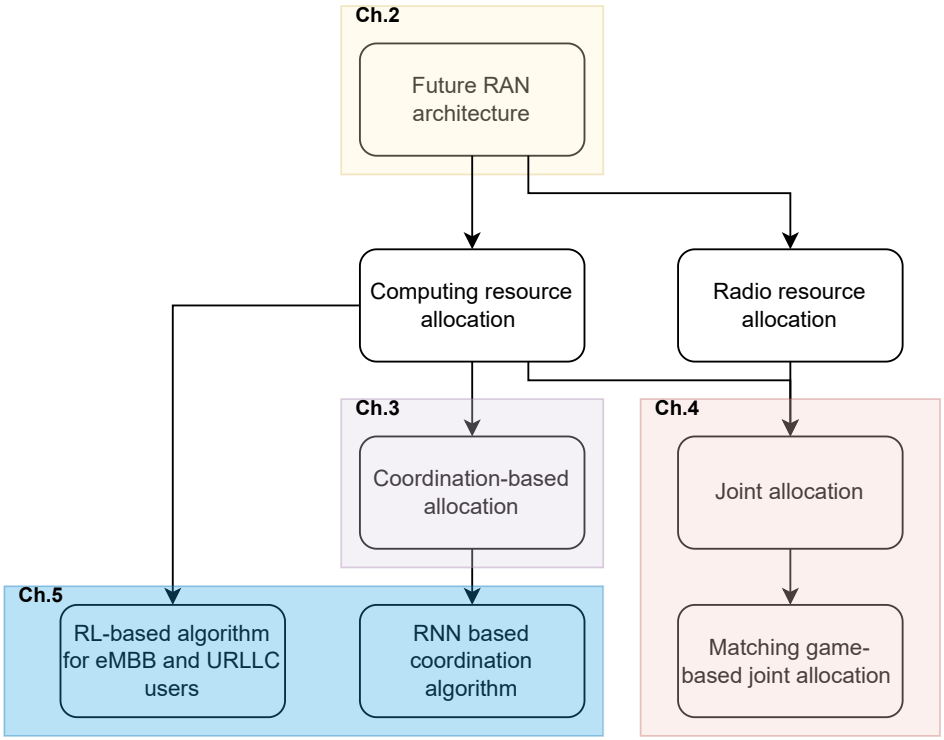


Figure 1.2: The structure of the dissertation

(MILP). The objective is to determine the limits within which the joint allocation is beneficial and find when it becomes non-beneficial to consider joint allocation. Additionally, we propose a lower-complexity algorithm based on a matching game that aims to output solutions that are close to the optimal solution of the MILP problem.

Chapter 5 investigates the potential of using machine learning in future RAN, and more precisely in Open-RAN. Two applications are considered; the first uses a Recurrent Neural Network (RRN) to allocate computing resources and enables the coordination scheme presented in Chapter 3. The model learns to perform close to the Integer Linear Programming (ILP) solver. The second application uses Policy-Gradient-based reinforcement learning to allocate computing resources in a multi-service scenario consisting of two services with heterogeneous requirements, enhanced Mobile BroadBand (eMBB) and Ultra-Reliable Low-Latency Communication (URLLC). The objective of this allocation is maximizing the operator's profits.

Chapter 6 concludes our work and provides perspectives for the future towards hierarchical and disaggregated resource allocation Open-Radio Access Network (O-RAN).

Appendix A briefly introduces the basics of supervised machine learning, neural networks, and deep learning.

Appendix B briefly discusses the basics of reinforcement learning, including it elements, value based-learning, and policy-gradient learning.

1.2 Publications

The work of this thesis has resulted in the following publications:

Published Journal Papers

- M. Sharara, S. Hoteit, P. Brown, and V. Vèque, "On Coordinated Scheduling of Radio and Computing Resources in Cloud-RAN," *IEEE Transactions on Network and Service Management*, 2022

Published Conference Papers

- M. Sharara, S. Hoteit, and V. Vèque, "Reinforcement Learning for Profit Maximization in Open-RAN," in *Accepted to 2023 IEEE/IFIP International Conference on Network Operations and Management Symposium (NOMS)*, 2023
- M. Sharara, T. Pamuklu, S. Hoteit, V. Vèque, and M. Erol-Kantarci, "Policy-Gradient-Based Reinforcement Learning for Computing Resources Allocation in O-RAN," in *2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*, 2022
- M. Sharara, S. Hoteit, V. Vèque, and F. Bassi, "Minimizing Power Consumption by Joint Radio and Computing Resource Allocation in Cloud-Ran," in *2022 IEEE Symposium on Computers and Communications (ISCC)*, 2022
- M. Sharara, S. Hoteit, and V. Vèque, "A Recurrent Neural Network Based Approach for Coordinating Radio and Computing Resources Allocation in Cloud-RAN," in *2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*
- M. Sharara, S. Hoteit, P. Brown, and V. Vèque, "Coordination between Radio and Computing Schedulers in Cloud-RAN," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2021
- M. Sharara, S. Hoteit, P. Brown, and V. Vèque, "Coordination de l'ordonnancement radio et de calcul dans Cloud-RAN," in *ALGOTEL 2021 - 23èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, La Rochelle, France, 2021

Submitted Papers

- M. Sharara, F. Fossati, S. Hoteit, V. Vèque, and F. Bassi, "Minimizing Energy Consumption by Joint Radio and Computing Resource Allocation in Cloud-Ran," *Submitted to the Journal of Computer Networks*,

It is worth mentioning that I spent three months during my PhD studies at NETCORE lab at the University of Ottawa, headed by Professor Melike Erol-Kantarci. The visit allowed me to profit from the expertise of other researchers working in the field of reinforcement learning and mobile networks.

Chapter 2

State-of-the-Art

The Radio Access Network (RAN) have been continuously evolving throughout the different mobile generations. The migration from the distributed architecture in 2G, 3G, and 4G has started; future RAN such as Cloud-RAN and Open-RAN are expected to become dominant in beyond 5G networks, given their centralized and virtualized architectures. This chapter discusses the evolution of the Radio Access Network (RAN) from the Distributed-RAN to a centralized and virtualized RAN. It surveys the Cloud-RAN along with its benefits and challenges. Then, the architecture of Open-RAN is presented along with the advantages and challenges. Afterward, this chapter surveys different radio and computing resource allocation works done in the literature. Finally, low-complexity tools used for resource allocation, precisely, Game Theory, Matching games, Deep Learning, and Reinforcement Learning, are briefly discussed.

2.1 Evolution of Radio Access Networks

The mobile network is traditionally divided into two main parts, the Radio Access Network (RAN) and the Core Network (CN). The RAN is responsible for providing radio connectivity to end-users and connecting them to the CN. At the same time, the CN coordinates the network functions and connects users from different RANs to each other and the Internet. The architecture of radio access networks has evolved through the different mobile network generations. In 2G, 3G, and 4G, the signal processing functions (baseband processing) have been held in a distributive manner. In fact, 2G and 3G architectures contain a centralized Base Station Controller (BSC) and a centralized Radio Network Controller (RNC), respectively. These units are connected to the distributed units called Base Transceiver Station (BTS) and nodeB, respectively. However, the role of these central entities is to manage mobility and handover, while the signal processing functions are done in the distributed units. In 4G, the architecture has evolved such that the RNC and the NodeB have been fused into a single entity named eNodeB.

In contrast to the traditional architectures, where baseband processing is held dis-

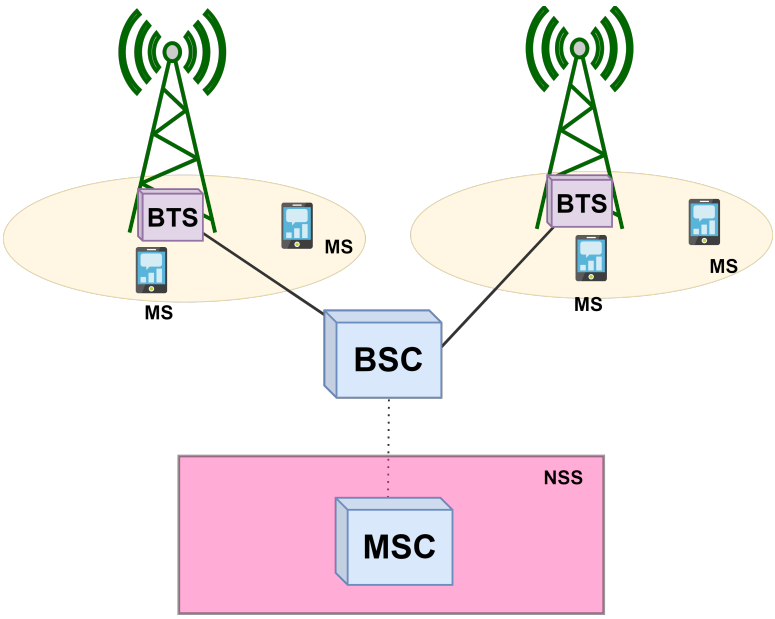


Figure 2.1: GSM simple architecture

tributively in each cellular site, Cloud-RAN aims to centralize the baseband processing of multiple base stations. Moreover, Cloud-RAN benefits from the advancement in virtualizing technologies. Hence, network components in the RAN can run as virtual machines that share powerful data and computing centers [11].

2.1.1 The second generation (2G)

The Radio Access Network has evolved through the different mobile network generations. While the first Generation was based on analog communication, and its usage was limited, the significant breakthrough came with the second Generation 2G, introducing digital communication in the wake of the fixed Integrated Services Digital Network (ISDN) [24]. The initial radio technology standard was called Global System for Mobile communication (GSM) [25]. In GSM, the network is split into three parts: the user (i.e., also called the Mobile Station (MS)), the Base Station Subsystem (BSS), and the Network Switching Subsystem (NSS). The main component in the NSS is called the Mobile switching Center (MSC), and it is responsible for controlling the elements of the NSS and communicating with BSCs connected to it, with the Public Switched Telephone Network (PSTN), and with other NSSs in different geographical areas. The BSS consists of the Base Station Transceiver (BTS), the Base Station Controller (BSC), and the interfaces connecting them to each other and to the MS and NSS. The simplified architecture of GSM network is presented in Fig 2.1. The BTS consists of antennas and radio front-end hardware that communicate with MS. The BSC is responsible for the radio resource management and the mobility management for the BTSs connected to it. This architecture was designed considering low-rate circuit-switched audio communication and SMS. The Radio Access Technology (RAT) used in GSM is based on Frequency

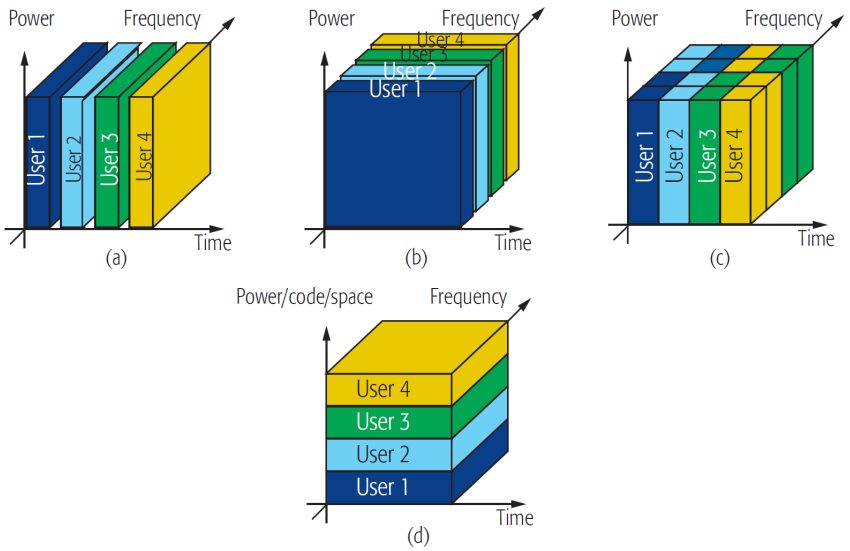


Figure 2.2: Multiple Access Techniques: a) CDMA b)FDMA c) OFDMA d) CDMA [26]

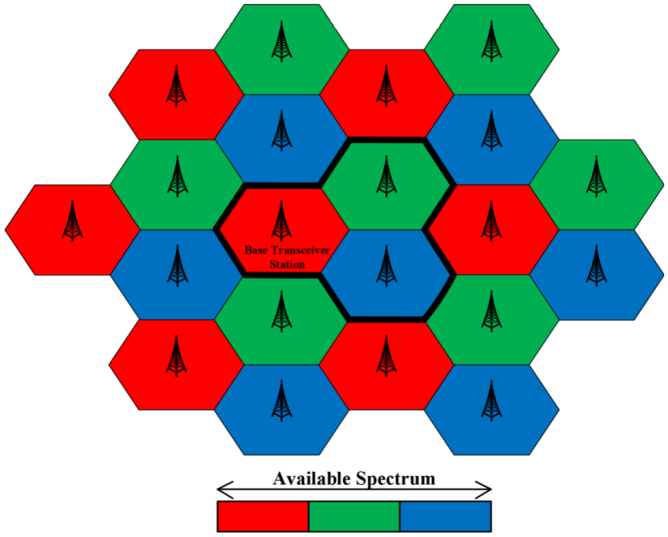


Figure 2.3: Frequency re-use [27]

Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA). These techniques are presented in Fig. 2.2.

In GSM, the concept of frequency reuse is adopted. The specturm is divided into portions and only a portion of the available spectrum is used in each cell. To decrease interference, Adjacent cells (i.e., base stations) do not reuse the same portion [27].

With the increasing demand for the internet, adding support for General Packet Radio Service (GPRS) functionality to GSM became necessary [28]. To handle the packet-switched data, an element called Packet Control Unit (PCU) was added to the

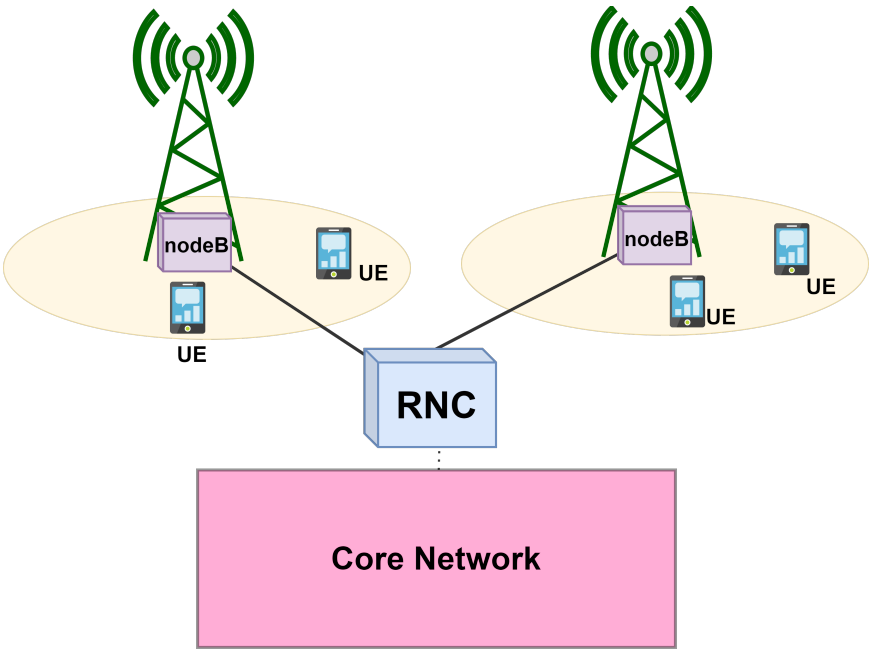


Figure 2.4: 3G simple architecture

BSS. This resulted in a modified BSS called GPRS BSS. The GPRS would provide a data rate approximately equal to 20 kbps per time slot [29]. The low rate of GPRS got enhancement by moving towards the Enhanced Data Rate for GSM Evolution (EDGE). The RAN became GSM EDGE RAN (GERAN). This technology permitted more efficient modulation schemes, increasing the data rate per time slot to around 60 kbps.

In 2G, the base station consisted of two devices, the Radio Equipment Controller (REC), and the Digital Unit (DU). The REC is responsible for the baseband functions, while the DU is responsible for Radio Frequency (RF) functions such as digital/analog conversion, amplification, RF filtering, and frequency conversion, in addition to modulation and demodulation [30].

2.1.2 The third generation (3G)

As the demand for higher data rates continued to grow, the 3rd Generation Partnership Project (3GPP) standardized the radio technology called Universal Mobile Telecommunications System (UMTS) in the third Generation (3G). The BSS of 2G evolved to become the Radio Access Network (RAN), while the NSS became the Core Network (CN). Hence, the RAN of 3G is known as UMTS Terrestrial Radio Access Network (UTRAN). The UTRAN consists of the User Equipment (UE), the base station called nodeB, the Radio Network Controller (RNC), and the interfaces connecting these units to each other and to the Core Network. The simple 3G RAN architecture is presented in Figure 2.4. The UTRAN uses circuit-switching for audio calls and packet-switching for data. In UTRAN, a Radio Network Subsystem (RNS) consists of at least one RNC controlling one or more nodeB [31], [32]. Similar to the BSC in 2G, the RNC is responsible

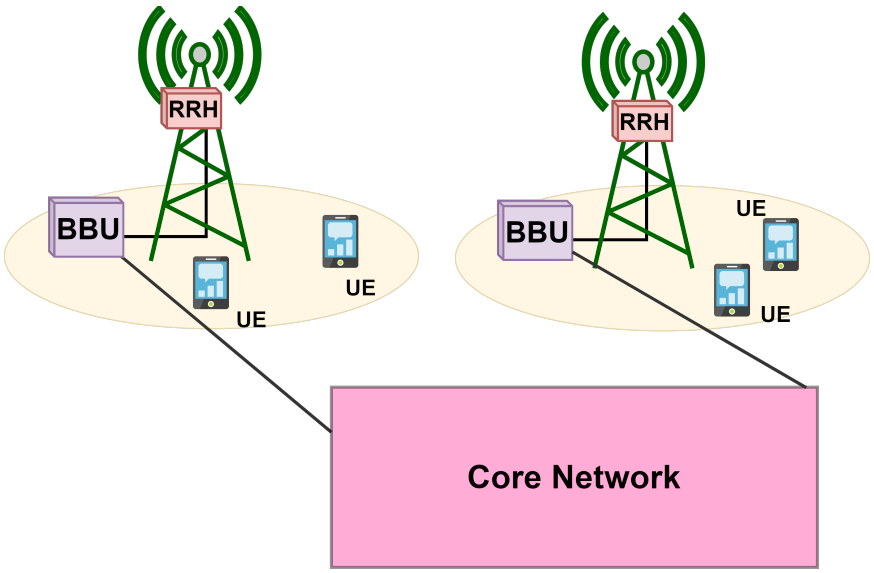


Figure 2.5: 4G simple architecture

for the mobility management of users and for Radio Resource Management (RRM). The Radio Access Technology used in 3G is Code Division Multiplexing (CDMA), shown in Figure 2.2. The original 3G supported data rates up to 384 Kbps. Hence, High Speed Packet Access (HSPA) was introduced, achieving a 14 Mbps peak rate in the downlink and 5.8 Mbps in the uplink [31], [32]. Using CDMA, the data rates in 3G improved in comparison with 2G; however, these rates were ideal theoretical rates because the interference in CDMA is higher due to non-perfect orthogonality between codes, which led to practically much lower data rates. With the increasing demands, the movement towards 4G became inevitable [11].

2.1.3 The fourth generation (4G)

The fourth Generation of Mobile Communication (4G) presents the first fully IP-based packet-switched mobile network where both voice and data are transmitted through a packet-switched network. In 4G, the Radio Access Network is called evolved UTRAN (eUTRAN). In the eUTRAN, there is no centralized controller as in 2G and 3G (i.e., BSC and RNC). The eUTRAN consists of the User Equipment (UE), the evolved NodeB (eNodeB), and the interfaces connecting the components to each other and to the Core Network [33], [34]. Functions that were previously part of the BSC and the RNC in 2G and 3G, such as mobility management and Radio Resource Management, became integrated into the eNodeB. The simplified 4G architecture is shown in Figure 2.5. In 4G, the radio technology used is Long Term Evolution (LTE), and the Radio Access Technology is Orthogonal Frequency Division Multiple Access (OFDMA) [11]. In comparison to CDMA, OFDMA is more robust against multi-path propagation and interference, allowing for much higher theoretical and practical data rates. The OFDMA technique is shown in Figure 2.2. LTE-Advanced (LTE-A) is an evolution of LTE that provides

different enhancements allowing higher order modulation, higher order of Multiple-Input-Multiple-Outputs Antennas, and Carrier Aggregation (CA) that allows different bands to be used altogether. The theoretical downlink data rate in LTE-A reaches 3Gbps [35].

2.1.4 The fifth generation (5G)

Recently, the fifth Generation of Mobile Networks (5G) incorporated additional upgrades. 5G New Radio (5G-NR) refers to the radio technology used in 5G. it supports using mmWave (i.e., above 24 GHz) and mid-band (i.e., sub 6 GHz) frequency ranges. Additionally, it permits using more carriers for aggregation. In 5G, the RAN is called Next Generation RAN (NG-RAN). It supports using both LTE and 5G-NR technologies. The base station is called Next-generation NodeB (gNB) when 5G-NR is used. If LTE is used, it is called Next-generation eNodeB (ng-eNB) [36], [37]. In 5G, the OFDMA parameters are flexible. Depending on the required services and latency, different numerologies that control the symbol time and the sub-carrier bandwidth can be used [22], [38].

5G supports different services including enhanced Mobile BroadBand (eMBB), Ultra Reliable Low Latency Communication (URLLC), and massive Machine Type Communication (mMTC). The eMBB aims to provide high throughput for users, while URLLC targets reliability and low latency. The mMTC service was initially considered for 5G to provide connectivity for a massive number of machines, but the 3GPP standard ended up focusing only on eMBB and URLLC, given that mMTC could use Narrow Band - IoT technology [22]. For the eMBB service, the peak data supports the requirement of 20 Gbps in the downlink and 10 Gbps in the uplink, mandated by the International Telecommunication Unit (ITU) [22]. For Ultra Reliable Low Latency Communication (URLLC), a success probability of 99.999% is supported within a delay of 1ms at the cell edge.

To meet the requirements of different services and use cases in 5G while supporting a flexible architecture, the concept of network slicing has emerged. In network slicing, the physical resources of the network, which include the radio resources, computing resources, the components of the RAN and the CN, the infrastructure, etc., can be virtually partitioned into different logically independent networks that share the physical resources. Virtualization and softwarization is a key enabler of network slicing. One of the essential solutions to deal with the massive increasing demand in mobile networks is to introduce Software Defined Networking (SDN) and Network Functions Virtualization (NFV) concepts. Using SDN and NFV to virtualize all network resources and functions leads to a more flexible and scalable network. For example, some applications, such as remote surgery, would have specific delay and reliability requirements. Hence, instead of deploying physical hardware, the operator could partition the network and allocate a slice to meet the requirements of this application. Outside working hours, the operator could flexibly release the resources. This elastic and scalable mechanism allows operators to reduce costs as it is not deploying dedicated infrastructure, and the operator can flexibly partition the network to adapt to the dynamic needs of its customers [11].

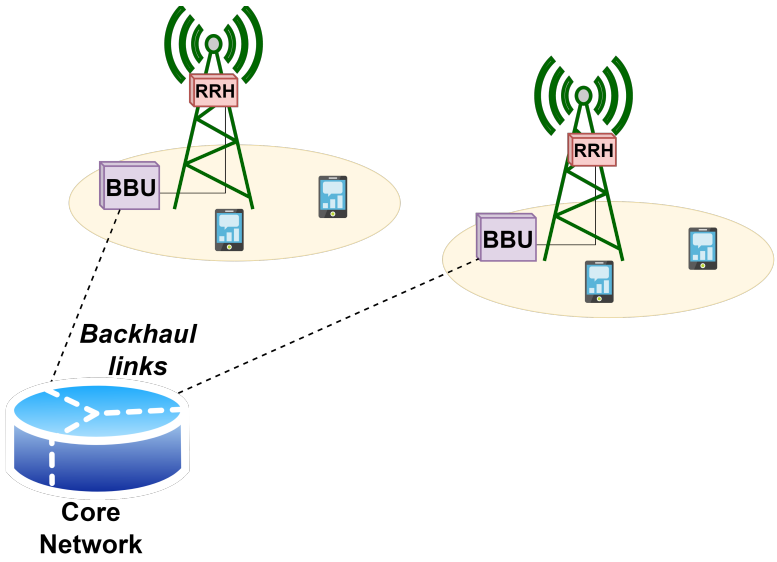


Figure 2.6: Distributed-RAN Architecture

2.1.5 Distributed Radio Access Network (D-RAN)

The RAN in 2G, 3G, and 4G is a Distributed RAN (D-RAN). In 3G and 4G architectures, the UTRAN and eUTRAN contain two elements responsible for RF functions and baseband signal processing: The Radio Remote Head (RRH), also called Radio Remote Unit (RRU), and the Base Band Unit (BBU), respectively [39]. The RRH is responsible for the RF functions such as Analog/Digital Conversion, Up/Down sampling, RF modulation, amplification, etc., whereas the BBU is responsible for the baseband functions such as Fast Fourier Transform, modulation/demodulation, encoding/decoding, and all the other functions from the physical layer and above. The RRHs are connected to the BBU through the Common Protocol Radio Interface CPRI interface. This interface transmits the In-phase and Quadrature I/Q signals between the RRHs and the BBUs. These two elements can be connected through fiber optical or microwave links, called fronthaul links. The distance separating these two units should be limited between 15-25 Km [40], though, practically, the RRHs and BBUs were colocated at the cellular site. The BBU is connected to the core network through backhaul links that are microwave or fiber links. User, control, and management data are carried through the backhaul links to the core network. The architecture of D-RAN in 4G is shown in Fig. 2.6.

Given that each base station has its own RRH and BBU deployed at the cellular site, the D-RAN is neither scalable nor cost-efficient. It is likely that the BBU resources will be underutilized, but operators will not be able to use these resources to satisfy increased demand elsewhere. This incurs a lot of deployment and operational costs.

In 5G and beyond, Cloud-RAN emerges as an efficient alternative to D-RAN.

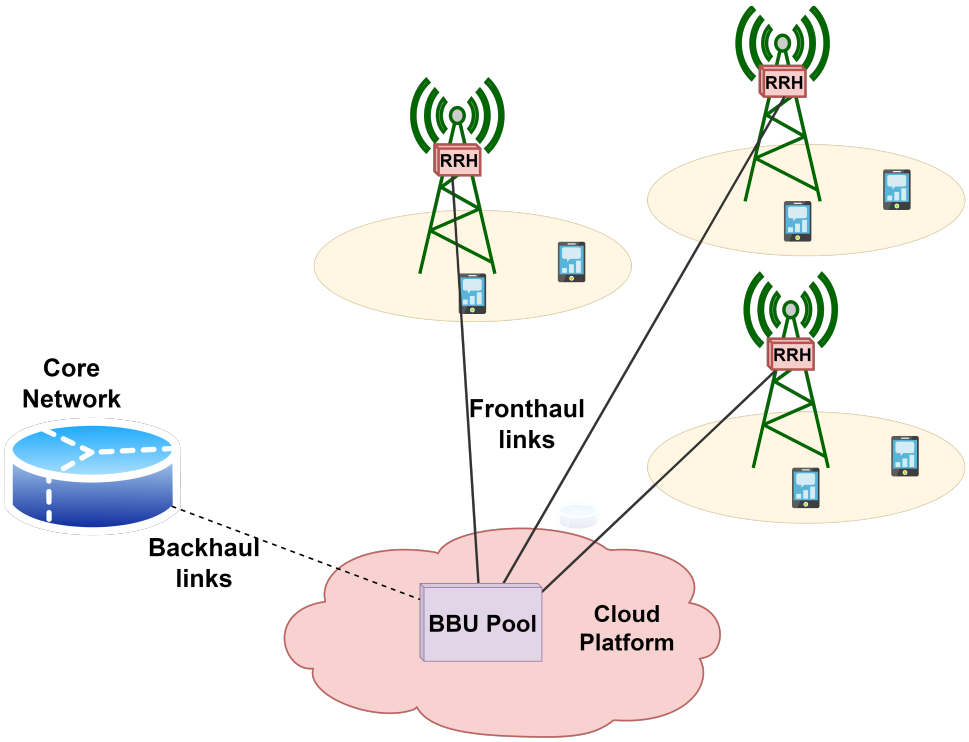


Figure 2.7: Cloud-RAN Architecture

2.1.6 Cloud Radio Access Network (Cloud-RAN)

In the Distributed RAN (D-RAN) architecture in 4G, the radio and baseband processing was split into two separate nodes coexisting in the same cellular site, i.e., RRH and BBU. While the RRH is responsible for the Radio Frequency (RF) functions such as Analog-Digital conversion and up-down conversion, all the baseband functions would be executed in the BBU. In contrast, a centralized and cloudified architecture of RAN has been proposed to cope with the massive increase in demand for network resources while saving the CAPEX and OPEX of the operator.

2.1.6.1 Concept and architecture

The Cloud-RAN architecture was first proposed and discussed in [14], [41]. In Cloud-RAN, all BBUs are separated from their corresponding RRHs, to which they are connected via fronthaul links that are either optical or microwave links. The BBUs are pooled into centralized and virtualized BBU pools that can host the baseband processing of many RRHs. The BBU pools are connected to the core network via backhaul links. The BBU pool consists of powerful data and computing centers. The BBUs in the BBU pool run as virtual machines sharing together the available computing resources. The architecture of Cloud-RAN is shown in Fig. 2.7.

Cloud-RAN can be regarded as embedding cloud computing into the RAN architecture. It is based on two main concepts; centralization and virtualization of the based

band processing. Centralization allows for optimizing the network performance, energy consumption, and spectral efficiency. On the other hand, virtualization is the reason why Cloud-RAN saves CAPEX and OPEX [14].

In Cloud-RAN, computing resources of Base Stations (BS) are shared. This would reduce capital and operational costs by reducing the number of deployed physical BBUs, and running them as Virtual Machines (VM) that share the resources of the physical infrastructure. Also, it would be possible to dynamically switch base stations on and off, depending on the instant needs. Since a separate entity may own the cloud resources, it would be possible to release resources that are no longer required or demand additional resources depending on the needs. Additionally, this flexibility allows for cutting down power consumption. The advantages of Cloud-RAN will be elaborated in Section 2.1.6.3.

The simplified architecture of Cloud-RAN is shown in Fig. 2.7.

2.1.6.2 Centralization level

In Cloud-RAN, it is possible to have different levels of centralization [11]. Full centralization means that all layers from the physical layer and above are executed in the BBU pool. The network stack includes the physical (PHY) layer, the Multiple Access (MAC), Radio Link Control (RLC), Packet Data Convergence Protocol (PDCP), and Radio Resource Control (RRC). Hence, physical layer functions such as resource block mapping, encoding, decoding, modulation, demodulation, antenna mapping, and quantization, functions from layer 2 related to transport and multiple access control, and functions from layer 3 related to radio resource control, are all centralized in the BBU pool. Full centralization permits easier modification of the network coverage, easier upgrade to network capacity, enhanced sharing of radio resources, better collaborative multi-cell processing, and better interference management. With full centralization, only Radio Frequency functions are left close to the antenna, allowing the use of Software Defined Radio (SDR) equipment, and this allows easier modifications of the RRH through software upgrades. On the other hand, full centralization faces a critical challenge; it is required to transmit raw I/Q signals between the RRH and the BBU and this requires a massive fronthaul capacity [14]. The other option is to partially centralize the functions. Some would be executed in the RRH or in equipment close to it, whereas the others are done in the BBU pool. The 3GPP standard [37] divides the BBU into a Central Unit (CU) and a Distributed Unit (DU). The concept of functional split defines how baseband functions are split among the CU and DU. While different decisions could be made, the split that the 3GPP standard has adopted is option 2 as shown in Figure 2.8. In this option, the RRC and PDCP layers are managed by the CU, while the other layers are executed in the DU. While this option enormously decreases the data rate requirement between the CU and the DU, it causes lower flexibility in the network upgrades and less dynamicity, and it may become less convenient to jointly process multi cells.

2.1.6.3 Advantages

Cloud-RAN comes with several advantages. One advantage lies in its ability to minimize energy consumption, thanks to the centralization [14], [42]. The reason is that less

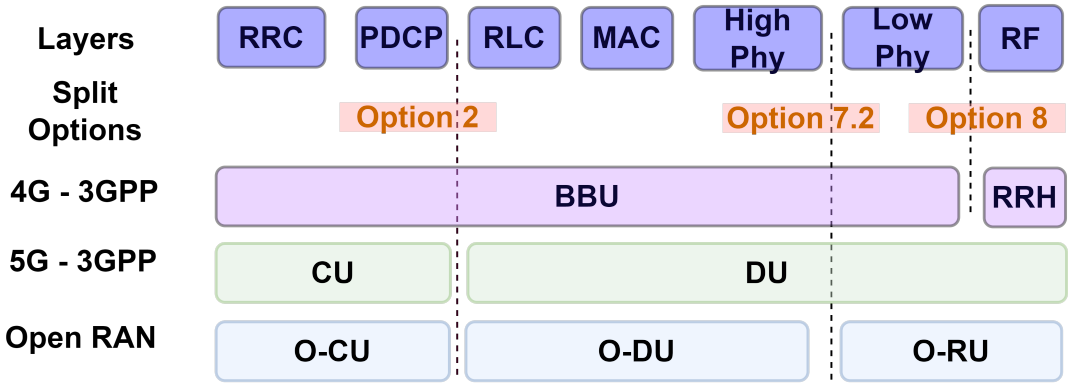


Figure 2.8: Standardized Functional Splits in 4G, 5G, and Open-RAN

air conditioning and less equipment are required at the cell site. Overall, Cloud-RAN has the ability to reduce power consumption by 70% and more [43]. Another advantage of Cloud-RAN is the reduction of CAPEX and OPEX. As [44] reveals, a network operator spends 80% of CAPEX on the RAN architecture, and 41% of OPEX at a cell site is related to power consumption. As [14] shows, Cloud-RAN would reduce the CAPEX and OPEX by 15% and 50%, respectively. The flexibility in RAN deployment enabled by Cloud-RAN allows for a more scalable network, where the operator can deploy more virtual BBUs, or shut down running ones depending on the needs. Thanks to centralization, flexibility, and scalability, Cloud-RAN facilitates load balancing and it clears the way for joint and cooperative network management and maintenance. Another advantage of Cloud-RAN is network performance improvement. According to [45], the downlink rate at the cell edge can be improved by 40% to 70% while the uplink rate could be improved by 2 to 3 times. Another advantage brought by Cloud-RAN is related to spectrum utilization. According to [42], Cloud-RAN has the ability to improve spectral efficiency. The centralized decision-making would allow better interference management, which leads to improved spectrum efficiency. Additionally, Cloud-RAN enables enhanced mobility management. In fact, Cloud-RAN architecture has an important role in decreasing the percentage of handovers by 20% [46]. It can also decrease the handover delay while reducing the risk of end-users losing connection [47].

2.1.6.4 Challenges

Cloud-RAN faces numerous obstacles. One challenge is the high required bandwidth on the fronthaul links connecting the RRHs to the BBUs [44]. While the 3GPP standardization has split the BBU into a DU and the CU and allocated a lot of functions to the DU, not adopting a fully centralized architecture would limit the possible reaped improvements of full centralization discussed earlier.

Another challenge is the cooperation and clustering of the BBUs of different cells [11]. How to group the BBUs into a cluster served by the same BBU pool is a critical decision. The BBUs that are part of the same cluster are the ones that are able to best cooperate

together. Grouping BBUs together should take into account the need of the member BBUs to cooperate together to minimize interference and optimize energy consumption (i.e., turning RRHs on and off, depending on the needs). It is challenging to get the BBUs to cooperate when they themselves may have opposed interests, and the improvement of the performance (i.e., throughput or latency) of one BBU would affect the others negatively.

Another issue is the virtualization techniques and computing resource allocation. Using dedicated hardware for each BBU is the easiest solution; however, it is likely to leave a lot of resources underutilized. Hence the virtualized resources should be as perfectly provisioned as possible, and resources should be optimally allocated to the BBUs running as virtual machines.

Ensuring the security of the virtualized BBUs is also an important challenge. The centralization and softwarization of the RAN make it more susceptible to security-related attacks, and this has to be carefully dealt with [11], [42].

One obstacle in Cloud-RAN lies in the ability to devise efficient low-complexity algorithms that reap the benefits of centralized optimization in Cloud-RAN. However, achieving the optimal gains could require solving very high-complexity problems. Solving these optimization problems could be infeasible in real-time, making the gains unattainable. It is required to devise low-complexity algorithms that are able to perform as close as possible to the optimal solutions. Our work in this dissertation devotes a lot of focus to this particular challenge.

2.1.7 Open Radio Access Network (Open-RAN)

In 2018, the Open-RAN initiative has been launched to further virtualize and disaggregate the components and open the interfaces of the RAN [15]. Open-RAN has native support of artificial intelligence algorithms, supporting multiple control loops with different time scales. The near Real-Time Radio Intelligent Controller has a time order between 10ms and 1s while the non-Teal-Time Radio Intelligent Controller has a time scale greater than 1s. Open-RAN paves the way for multi-vendor architecture where various interoperable network components can belong to different vendors [48]. This would encourage competition and make operators less dependent on a limited number of telecom equipment suppliers. Hence, Open-RAN should further help operators reduce their CAPEX and OPEX [49]. [15] presents a comprehensive survey on Open-RAN, its architecture, open interfaces, and the research challenges facing Open-RAN.

2.1.7.1 Architecture

It is worth noting that the notions differ in recent 3GPP and Open-RAN releases, the BBU is split into a Central Unit (CU) and a Distributed Unit (DU) in 3GPP. In Open-RAN, the CU is the Open CU (O-CU) or virtual O-CU (vO-CU), while the DU is the Open DU (O-DU) or virtual O-DU (vO-DU). The RRH became the Open Radio Unit (O-RU).

Open-RAN is based on four pillars [15]. The first is disaggregating network components (i.e., O-RU, O-DU, O-CU, etc.) while allowing these components to jointly

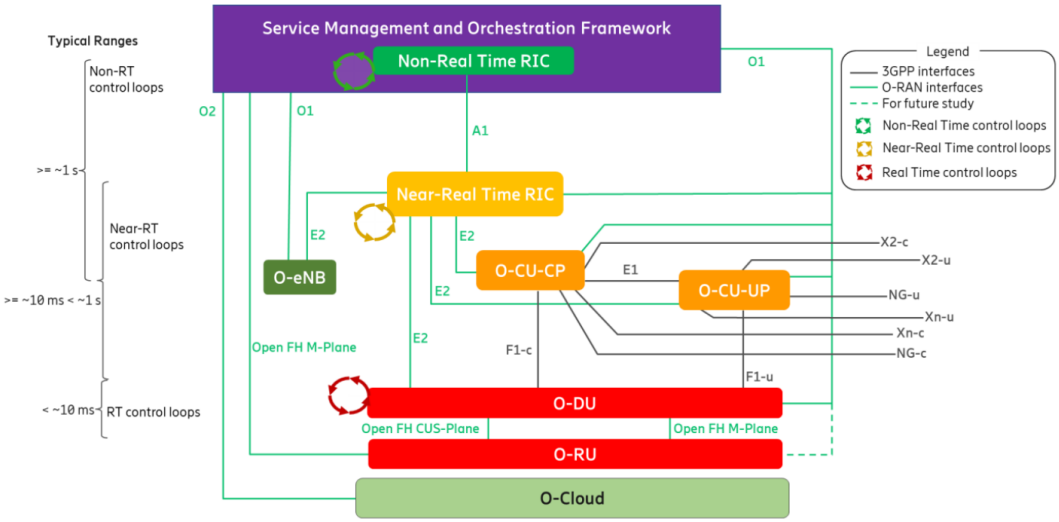


Figure 2.9: Open-RAN Architecture [50]

control multiple nodes (i.e., one O-CU can control multiple O-DUs, and one O-DU can control multiple O-RUs). The second pillar is Radio Intelligent Controllers (RICs) and closed-loop control. Different network nodes benefit from data-driven intelligence and cooperate together through the RICs to optimize the system's performance. The third pillar is the virtualization of the network components, as in Cloud-RAN, while the fourth is opening the interfaces to ensure multi-vendor interoperability. While the architecture of Open-RAN is compliant with the 3GPP standard [50], it includes additional components and interfaces.

The architecture of Open-RAN is shown in Fig. 2.9. Regarding the functional split, the 3GPP-compliant option 2 is employed between the O-CU and the O-DU. Thus the functions related to the Radio Resource Control (RRC) and Packet Data Convergence Protocol (PDCP) layers are held in the CU. Additionally, Open-RAN standardizes option 7.2 between the O-RU and the O-DU [51]. Thus the O-RU hosts the Radio Frequency (RF) and low-Phy functions, while the rest (i.e., high-Phy (e.g., beamforming, fast Fourier transforms, inverse fast Fourier transforms, etc.), RLC, and MAC functions) are held in the O-DU. In the 3GPP standard, the CU is split in O-CU-CP and O-CU-UP, where CP stands for Control Plane and UP stands for User Plane. These two components communicate with each other through the E1 interface. The O-CU-CP communicates with the O-DU through the F1-c interface, while the O-CU-UP communicates with the O-DU through the F1-u. The O-CU-CP is connected to other 4G eNB nodes through the X2-c interface, to other O-CU-CP nodes through Xn-c interface, and to the core network through the NG-c interface. Similarly, the O-CU-UP is connected to other 4G eNB nodes through the X2-u interface, to other O-CU-UP nodes through Xn-u interface, and to the core network through the NG-u interface. Open-RAN defines an interface between the O-RU and O-DU, which is the Open Fronthaul (Open FH). It consists of 4 planes: Control (C), User (U), Synchronization (S), and Management (M). Open-RAN specifies

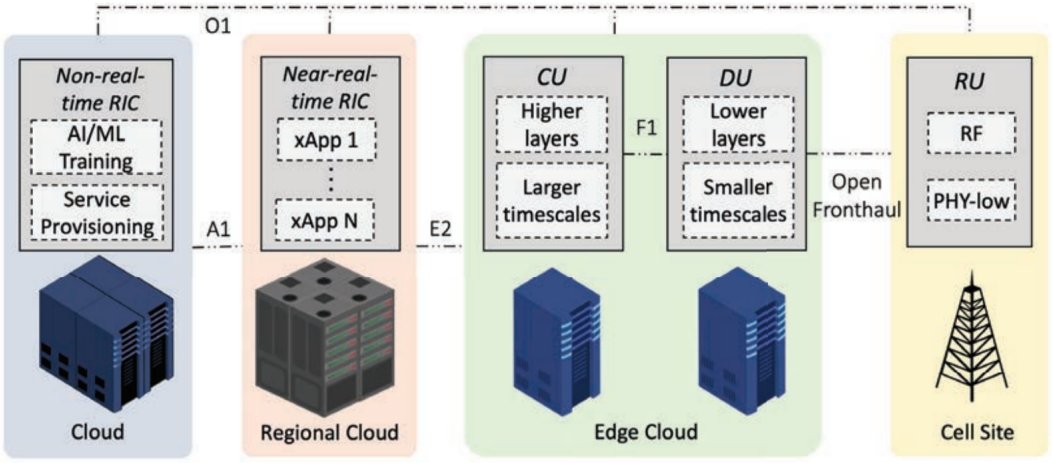


Figure 2.10: An Open-RAN Deployment Scenario [48]

the Service Management and Orchestration (SMO) Framework. It is responsible for network-wide management. It interacts with the other nodes through the O1, O2, and A2 interfaces. The O1 connects it to the Near-Real-Time Radio Intelligent Controller (Near-RT-RIC), O-CU-CP, O-CU-UP, O-DU, O-RU, and O-eNB. The O-eNB is the Open-RAN-compliant 4th Generation eNB. Additionally, the SMO manages the Open Cloud (O-Cloud) through the O2 interface. The Open-RAN standardizes two Radio Intelligent Controllers: the Non-Real-Time (Non-RT-RIC), which is placed in the SMO, and the Near-Real-Time RIC. The first supports time scales higher than 1 second, while the latter supports times scales between 10 milliseconds and 1 second [52], [53]. The two RICs are interconnected through the A1 interface. The Near-RT-RIC is connected to the O-CU-CP, O-CU-UP, O-eNB, and O-DU through the E2 interface. While not standardized yet, proposals are being made to standardize a third RIC responsible for time scales less than 10 milliseconds [54].

The Open-RAN architecture aims at opening the interfaces. Given that the RAN nodes may come from different vendors, it is necessary to ensure the interoperability between these nodes, and hence opening the interfaces becomes necessary. Open-RAN architecture provides native support for training and deploying Machine Learning (ML) models. The applications running in the Non-RT-RIC and Near-RT-RIC are called rApps and xApps, respectively [55]. These apps subscribe to different information (i.e., Key Performance Indicators) in the network provided by the different nodes and run different tasks using this information. One of these tasks could be to train or execute ML models. Finally, the Open-RAN architecture supports flexible deployment. Cloud servers will exist in different geographical areas, at the edge, in the region, or even much further. Depending on the QoS requirements and the profit/cost tradeoff, operators can dynamically place the RAN nodes. Fig. 2.10 presents a scenario where the O-CU and O-DU are placed in the edge server close to the O-RU, while the Near-RT-RIC is located in the regional servers. The Non-RT-RIC, due to its relatively big timescale of 1 second, is

placed in the cloud, which could be far from the regional cloud.

2.1.7.2 Advantages and use cases

The Open-RAN architecture insures that the benefits of the Cloud-RAN are reaped. These include reducing power consumption, CAPEX, and OPEX. Also, the performance would be improved due to centralized management. Moreover, as in Cloud-RAN, Open-RAN presents a flexible network architecture, where nodes are run as virtual machines instead of hardware nodes that require more CAPEX and OPEX and take more time to be deployed. Additionally, Open-RAN adds some extra advantages. The open interfaces in Open-RAN allow more competition in the market. This would encourage innovation and allow new market players. As a result, the competitive atmosphere would help decrease the CAPEX of operators. Moreover, the open architecture will speed up innovation and accelerate the evolution of the RAN. In fact, the open interfaces and the softwarization paved the way for integrating intelligent, data-driven closed-loop control [56]. This allows the system to benefit from the massively available data to self-adjust and self-optimize its performance.

Open-RAN tackles different use cases. For example, Open-RAN facilitates mobility management. This involves multiple decisions regarding handover, load balancing, and multi-connectivity. Open-RAN architecture allows intelligent closed-loop control so that the system itself is able to learn the optimal decisions and adapt to the dynamic nature of the network. Open-RAN technical specifications in [57] and [58] present a use case where an xApp exploits enrichment information provided by the non-RT-RIC and applies an AI model on the data from the RAN to infer decisions on handover for vehicular networks and for Unmanned Aerial Vehicles. Resource Allocation for multi-type services is facilitated by Open-RAN. For example, the RICs can aid in allocating radio resources to guaranteed-bit-rate URLLC sessions [59] and can help in adopting good slicing mechanisms to achieve the required Quality of Service for URLLC users [60]. Due to softwarization, RAN sharing between different operators is facilitated. This would include sharing the radio spectrum [61], the computing resources, the infrastructure, the nodes, etc. [57], [62]. Instead of deploying independent infrastructure, operators can share the infrastructure while their networks are logically isolated. This would allow operators to reduce deployment costs in some places where there is only a little number of customers.

2.1.7.3 Challenges

Despite the benefits of Open-RAN, it still faces different challenges [15], [55]. The Open-RAN architecture should keep evolving. For example, the current standardization supports two RICs, one with a time scale between 10 ms and 1s and the other with a time scale above 1s. Standardizing a RIC with a timescale lower than 10 ms is yet to be made. The authors in [54] have proposed a real-time RIC with a time scale of less than 10 ms. Applications called dApps take different radio decisions (i.e., scheduling, beamforming, etc.) and can possibly use Machine Learning in real-time. These efforts should continue to finally arrive at an architecture fully supporting real-time decision-making.

Given that integrating Machine Learning has been a key motive for Open-RAN, it is necessary to ensure that the deployed ML models perform close to the optimal as much as possible. Given that ML is used to solve highly complex problems (i.e., NP-Hard), it would be difficult to find the optimal solution to the problem to compare it with ML model. This could affect the ability to find the margin of improvement that an ML model can achieve. The ML models should continuously evolve to benefit from the recent ML architecture in different fields, such as computer vision and Natural Language Processing.

As mentioned earlier for Cloud-RAN, another challenge for Open-RAN is related to its security. With a multi-vendor approach, an attacker would attempt to exploit the vulnerabilities of different nodes and penetrate the network. In the case of inter-operator sharing, an attacker could try to penetrate into the network of one operator by hijacking another. Also, different attacks could be done against the ML models, modifying their parameters, altering the data, etc [15], [55], [63]. The security threats in Open-RAN must be addressed and tackled to ensure the benefits of Open-RAN are reaped. Finally, Open-RAN should take more initiatives that define optimization routines that embed energy efficiency optimization as an optimization objective. The virtualization itself should optimize energy efficiency. However, currently, the objectives of Open-RAN do not focus on energy efficiency and the roles it should play in the pursuit of greener zero-emission energy [15].

To benefit from the architecture of both Cloud-RAN and Open-RAN, different resource allocation problems are considered, and various solutions are proposed. We discuss these in the next section.

2.2 Radio and Computing Resources Allocation

One important issue that has been investigated in 5G is the allocation of radio and computing resources. The radio resource allocation involves allocating OFDM resource blocks, the Modulation and Coding Scheme (MCS) indexes, and power. The computing resources are those available in the BBU pool and are dedicated to the baseband processing functions. Different papers from the literature consider resource allocation problems in Cloud-RAN and Open-RAN, aiming to optimize the performance from different perspectives. For example, the objectives have been minimizing energy consumption, minimizing CAPEX/OPEX, maximizing throughput, and minimizing latency for Cloud-RAN and Open-RAN architectures.

2.2.1 Radio resource allocation

A variety of papers have focused on radio resource allocation. The prospective gains of resource virtualization in Cloud-RAN have been considered in [64]. The authors investigate the potential benefits of the deployment of their resource virtualization algorithm that can be used to share radio resources among different mobile operators. The air

interface resources (Resource Blocks) are coordinated and allocated dynamically by a hypervisor that allocates resources among virtual operators (VOs). The algorithm has been designed to adapt resource virtualization dynamically according to the VOs traffic load balance. By using the algorithm, the throughput, end-to-end packet delay, and jitter are improved when comparing it to a round-robin mechanism.

The issue of allocating resources to different network slices (i.e., Ultra-Reliable Low Latency (URLLC), enhanced Mobile Broad Band (eMBB), massive Machine Type Communication (mMTC)) is investigated in [65] where the authors model the resource allocation of network slices as a bankruptcy game. Precisely, the game allocates the Resource Blocks (RBs) to the different network slices. In the same context of coexisting multi-services, the authors in [66] study the user-plane latency of URLLC for different numerologies and consider the impact of allocating fixed bandwidth for URLLC service on URLLC users' latency and on eMBB users' throughput. In [67], the authors propose a dynamic network slicing prototype and test it for practical implementation using Open Air Interface software. In [68], the same authors develop a dynamic network slicing algorithm to allocate slices to IoT users coexisting with eMBB slices. A slice is defined by the resource blocks it uses, the usage duration, and the required QoS.

In [69], the problem of resource allocation of sub-channels in Heterogeneous Cloud-RAN is addressed. The heterogeneous Cloud-RAN includes macro base stations, micro base stations, and Device-to-Device users. The authors consider a downlink model that consists of one eNB, which is the high-power node transmitting to mobile users, RRHs of low-power nodes transmitting to RRH users, and D2D transmitters transmitting to D2D receivers. The different transmitters in the system reuse the radio sub-channels. The authors model an optimization problem to maximize the system throughput by assigning Resource Blocks (RBs) to the different transmitters. Since the problem is modeled as a mixed integer non-linear programming, the authors use a two-step algorithm based on matching theory and coalition games to find sub-optimal solutions. Focusing on guaranteeing a minimum transmission rate for users, [70] considers the issue of joint user-centric clustering and frequency allocation in ultra-dense networks where a user is connected to several RRHs. Maximizing energy efficiency is prioritized in [71]. The authors consider the resource allocation in a Non-Orthogonal Multiple Access (NOMA)-based Fog-RAN. Fog-RAN combines Cloud-RAN and fog-computing. In the latter scheme, users with limited computational resources can offload their computation to fog access points located near the users. As the formulated optimization problem is a mixed integer non-linear programming problem that is NP-Hard, a Stackelberg game is formulated with a follower, which involves solving a one-to-many matching game to assign fog access point to RRH sub-channels, and a leader defined as the non-cooperative game for NOMA-based power allocation; it is solved using dynamic best response.

As previously mentioned, one of the important challenges in Cloud-RAN lies in how to group RRHs in clusters, where one BBU pool manages each cluster. Such a problem is referred to as clustering. In [72], the authors focus on RRH-clustering, where each cluster is processed by a specific BBU that has limited and constrained available compu-

tational resources. The RRHs should be allocated to different clusters, and each cluster should have its throughput maximized. The authors consider RRH clustering and power allocation with beamforming, where the objective is to maximize the sum of clusters' throughput. Given that the modeled problem is NP-hard, and since the objective function is a sum of convex and concave functions, the authors solve the problem using the successive convex approximation method. A coalition-formation game has been used for RRH clustering in [73], where a cluster forms a coalition of RRHs managed together by a Base Band Unit (BBU). A cluster utility is based on three metrics; the total throughput, power consumption, and frequency of handover of member RRHs. The handover here is defined as the movement of an RRH from one cluster to another. The results show that the proposed approach converges after a small number of iterations achieving near-optimal solutions. In the same context, [74] proposes a hybrid solution that associates RRHs to BBU clusters using a potential game and then relies on a centralized entity to activate/deactivate BBUs to reduce signaling between BBUs. Given a set of possible clusters, RRHs act as players who aim to join the most beneficial cluster; the one that best minimizes the cost function. The latter is a function that takes into account the load, the inter-cluster interference, and the re-association of RRHs to another BBU (i.e., handover). The handover is included as a penalty for switching from one BBU to another. To find the Nash equilibrium, the best-response-dynamics algorithm is used. Afterward, a BBU activation/deactivation is used to decrease the number of BBUs in the system leading to a decrease in the signaling exchange between different BBUs. Moreover, [75] considers jointly mitigating interference by having cooperative interference management among RRHs forming a cluster/coalition. In this context, a distributed coalition formation algorithm is developed, and its stability is analyzed.

In an Open-RAN context, [76] considers team learning to mitigate the conflicts between two xApps that allocate resource blocks and power, respectively. In team learning, different independent xApps collaborate and exchange information before issuing the final decisions. The proposed algorithm increases the throughput and decreases the packet loss rate. In the same Open-RAN context, [77] formulates a Mixed Integer Non-Linear Programming (MINLP) problem that maximizes the throughput of eMBB users and minimizes the latency of URLLC users subject to limited power and fronthaul constraints. Then the problem is relaxed, and an algorithm based on successive convex approximation is used to solve the problem.

In another context, [78] formulates a radio allocation MILP problem to maximize the sum-rate. It considers RBs and MCS assignments in addition to power allocation. However, the model considers only one base station and neglects the interference caused by other base stations. Additionally, the problem only considers radio allocation without considering computing resource allocation.

2.2.2 Computing resources allocation

Due to the centralization in Cloud-RAN, multiple RRHs should share the computing resources of the BBU pool. Hence it is necessary to characterize the computing resources required to execute the baseband functions. In [79], the authors use Open Air Interface

(OAI) software and Universal Software Radio Peripheral (USRP) equipment, to test the performance of the BBU pool according to throughput and latency and to characterize the processing time. Using the OAI simulator in [80], the authors study the different processing times of BBU uplink functions and show that the decoding function is the largest consumer of computational resources. The authors show that Fast Fourier Transform (FFT) and demodulation functions do not depend on the MCS index and require less processing time than the decoding function, whose processing time increases with the MCS index. Besides, the authors develop some models for processing time prediction using interpolation and deep learning techniques. A better processing time model has been proposed in [81]. Unlike the model in [80], which provides the processing time as a function of the MCS index, the work in [81] models the processing time as a function of the MCS index, the number of RBs, and the CPU frequency. [82] aims for characterizing the processing time related to the PDCP layer. In 5G, the PDCP layer is implemented in the gNB-CU based on the functional split option-2. Moreover, [83] focuses on characterizing the computational load in BBUs and then aims at forecasting this load. Based on the forecasted results, the BBU pool computational resources should be distributed on virtual machines to which a cluster of RRHs is connected. These RRHs should share the available computing resources of the VM. The authors formulate an optimization problem to minimize the cost paid when RRHs are associated to VMs and to minimize the re-associations between VMs and RRHs. The authors use Open Air interface software for simulation. Considering the BBU RRH assignment problem, the authors in [84] formulate an ILP problem to minimize the total BBU pool processing power, while minimizing the failure probability. The authors use a Branch-and-Price algorithm to solve the ILP problem. Aiming to minimize power consumption and virtualization cost and to maximize the load of the low-loaded gNBs, the authors [85] formulate an ILP problem to assign gNB to Virtual machines. They devise an algorithm Branch, Cut, and Price (BCP) that demonstrates performance improvements in comparison to other algorithms. The authors in [86] study the effect of applying the decoding function in parallel and propose two algorithms that ensure parallelism. Their results show that such an option can reduce the decoding function's run time.

In [13], the authors propose two different computing scheduling algorithms to schedule the processing of RRHs' data arriving in every transmission time interval (TTI). These algorithms aim to increase the number of correctly decoded sub-frames and the system throughput. The authors evaluate the performance of these algorithms as a function of the number of RRHs assigned to the BBU pool. Additionally, they compare their proposed algorithms to known scheduling heuristics that only focus on computing scheduling without considering radio scheduling. The optimal placement of vBBUs that minimizes the latency has been investigated in [87]. Given a model constituted of dual LTE-5G connectivity with dual base stations, the authors propose a Neighboring Aware Placement (NAP) algorithm to optimally place the vBBUs. Since the communication between vBBUs of co-located 4G and 5G base stations could increase the latency, the algorithm tends to place the vBBUs in the same cloud server; such a proposal minimizes

the effect of inter-vBBU communication. Targeting the optimal placement of network functions, [88] models the virtualized and containerized network functions placement in Open-RAN as a Mixed Integer Linear Programming (MILP) problem aiming to aiming to minimize the end-to-end delay. To solve the NP-Hard problem, the authors design their own gradient-based algorithm. [89] considers maximization of the allocation of computing resources of the BBU pool. In the case of resource shortage, it uses a Nash bargaining solution to allocate resources, ensuring that each user receives at least a minimum amount of resources.

2.2.3 Joint radio and computing resource allocation

In the previous sections, we discussed various research works that focus either on radio resource allocation or computing resource allocation. In this part, we survey multiple research works that consider at least some cooperation between radio and computing resource allocation. In [90], the authors consider Mobile Network Operator (MNO) sharing of resources. Resources such as spectrum, BBUs, and RRHs are shared among Mobile Network Operators to increase energy efficiency. The authors propose a game-theoretic approach and investigate when the grand coalition of all MNOs (i.e., where all MNOs cooperate) remains desired by all members. The utility of each operator is the revenue from users minus their share of the costs of OPEX power consumption.

Focusing on optimizing throughput while reducing costs, [91] considers the issue of joint radio and computing resources in Internet of Things (IoT) Fog Computing. The problem considers the assignment of users to Fog Nodes of service providers. A fog node has spectrum resources and CPUs of different cycles. The service provider allocates a resource pair composed of a spectrum and a CPU to users. The authors use a student-project matching game to assign users to the radio-computing pairs. Similarly, [92] aim to maximize throughput satisfaction and reduce deployment cost. The authors consider the joint allocation of functional split and radio, computing, and link resources. They model their problem as an ILP problem and devise an algorithm called Branch-and-Cut to solve it. To maximize the network sum rate, the authors in [93] formulate a binary integer non-linear programming problem constrained by limited available computing resources in the BBU pool. The decision variables associate users with RRHs. To jointly optimize the throughput and the functional split, the authors of [94] formulate the problem as an ILP problem that allocates resource blocks, fronthaul bandwidth, and computing resources.

Considering a Cloud-RAN allocation scheme, [95] proposes an energy-efficient resource allocation scheme and jointly allocates antenna resources (RRH) and computation (BBU) resources. The problem is decomposed into two sub-problems; first, the beamforming vector optimization problem is formulated and solved using the weighted mean square error approach. The assigned weights of the beamforming vector determine the RRHs that are jointly serving a user. The second subproblem of scheduling users to BBUs is solved using bin-packing to minimize the number of BBU pools, which in turn saves more energy. The authors in [96], [97] consider radio resource allocation and power minimization problems in Downlink in Cloud RAN. They propose an algorithm to dynamically allocate resources and assign BBUs to RRHs. [98] proposes an algorithm to

allocate power and bandwidth to users while meeting their quality of service. They use a bin-packing problem to minimize the number of active virtual machines that process users' data. These problems aim to minimize energy consumption while maintaining the required quality of service. The paper analyzes the performance of Cloud-RAN using an OAI-testbed. It derives a formula for the processing time as a function of the MCS index and the number of RBs, and a formula for the CPU utilization as a function of the throughput. The processing time formula considers only three values for the number of RBs. The authors of [99] consider radio resource allocation followed by the RRH-BBU association aiming at minimizing power consumption. Their problem aims to minimize the number of BBUs used to process computational requirements. They also propose two heuristic algorithms to solve these two problems. The authors of [100] consider the problem of user-to-RRH association followed by the BBU processing allocation with the goal of power minimization. They formulate a Bin-Packing algorithm to decide which BBUs should process users' data, and then they propose lower-complexity algorithms to solve these two problems. Focusing on joint communication and computing allocation, the authors in [101] formulate the joint problem using queuing theory. They propose an optimization problem that tries to ensure the stability of RRHs and BBU queues by controlling the assignment of users to RRHs and the assignment of RRHs to BBUs with the aim of decreasing the response time. The problem is solved by an auction-based algorithm. The authors in [102] consider joint beamforming vector design and BBU computational resources allocation. They aim to minimize the total system power consumption while considering the constraints of users' Quality of Service (QoS), fronthaul capacity, transmit power per RRH, and per Antenna. Aiming to maximize the sum-rate under limited computing resources, the authors in [103] propose an algorithm that schedules users who contribute less to computational outages and permits downgrading MCS indexes if the computing resources are insufficient. Considering service-aware resource allocation in an Open-RAN context and aiming to maximize the sum-rate, [104] formulates a Mixed Integer Non-Linear Programming (MINLP), and allocate RBs, power, and VNF processing resources subject to limited fronthaul capacity and end-to-end delay constraints. Due to the NP-hardness of the problem, the authors devise a low-complexity sub-optimal algorithm to solve the problem.

To jointly manage radio and computation resources to achieve low latency in Multi-Access Edge Computing (MEC), the authors of [105] propose an algorithm to minimize service latency by optimizing the uplink transmission power, receive beamforming, computation task assignment, and computation resource allocation. In comparison, [106] proposes a joint communication and computation cooperation in MEC. However, this proposal aims to improve energy efficiency. The proposed scheme jointly optimizes the task partition, time allocation, and transmit power for offloading, in addition to the CPU frequencies of local computing at the user and the helper nodes. The problem is formulated as a non-convex problem then it is relaxed to a convex problem that can be solved using interior-point methods. In [107], a NOMA-based MEC model is considered, aiming to improve energy efficiency. A joint optimization problem is formulated where

	Problem - Task	Allocated Resources	Methods - Tools
[90]	Mobile operators sharing to increase energy efficiency	Spectrum, RRHs, BBUs	Coalitional game-theory
[91]	Optimizing throughput and costs when assigning Fog Access Points to users.	Spectrum, CPU	Matching game
[92]	Optimizing throughput and cost	Functional split, radio, computing, link	Integer Linear Programming and Branch-and-Cut algorithm
[93]	Optimize the throughput and the functional split	resource blocks, fronthaul bandwidth, and computing resources	Binary Integer Non Linear Programming and heuristic
[94]	Maximize network sum-rate	Transmit power, RRH association, computing resources	Integer Linear Programming and heuristic
[95]	Optimizing energy efficiency when allocating antenna resources and computing resources to users	RRH association, beamforming vector, BBU computing resources	Weigthed mean square error for radio resources allocation, and Bin-Packing for computing resources
[96], [97]	Power minimization	Resource blocks, power allocation, user-RRH association, RRH-BBU association, BBU computing resources	Mixed Integer Linear Programming, Knaspsack, and Branch-and-Cut algorithm
[98]	Satisfying Quality of Service of users	Power allocation, bandwidth, computing resources, active BBUs (computing resources)	Mixed Integer Non Linear Programming relaxed to convex optimization, and Bin-Packing
[99]	Radio resource allocation and RRH-BBU association to minimize power consumption	Resource Blocks, RRH-BBU association, activating BBUs (computing resources)	Binary Integer programming and heuristics
[100]	Minimizing power consumption for user-RRH association and BBU processing allocation	user-RRH association, subchannel allocation, computing resources	Non linear combinatorial problems, convex relaxations, Bin-Packing
[101]	Ensuring the stability of RRH and BBU queues and minimization of response time	user-RRH association: RBs and power, RRH-BBU assignment: computing resources of virtual machines	Mixed Integer Non Linear Programming and auction game-theory
[102]	Minimizing total power consumption while satisfying QoS of users	transmit power, fronthaul capacity, computing resources	Non-convex optimization, weighted mean square error
[104]	Maximizing users' sum-rate	RBs, power, VNF processing resources	Mixed Integer Non Linear Programming, sub-optimal heuristic
[105]	Minimizing service latency in Multi-Access Edge Computing	uplink transmission power, receive beamforming, computation task assignment, and computing resources	Mixed Integer Non Linear Programming, sub-optimal heuristic
[106]	Optimizing energy efficiency in Multi-Access Edge Computing	task partition, time allocation, transmit power, CPU frequencies	Convex relaxation and Interior-point method
[107]	Optimizing energy efficiency in NOMA-based Multi-Access Edge Computing	user association, power control, computing resources	Mixed Integer Non Linear Programming, Matching game, Coalitional game

Table 2.1: A summary of the papers that consider joint radio and computing resource allocation

the problems of user association, power control, and computational resource allocation are combined together to optimize energy consumption. The problem is solved using a matching-coalition approach.

Table 2.1 summarizes the works related to joint radio and computing resource allo-

cation. The works presented in this section considered different objectives in tune with the objectives of future RAN. One limitation is that these problems had to focus on one objective. It is very challenging to consider multiple objectives problems, given that single-objective problems are already highly complex. Another limitation is that these problems could not focus on allocating all possible resources simultaneously, although this is justified due to the complexity of the problems. Moreover, the improvement of joint resource allocation with respect to non joint is not well quantified.

The majority of these papers use mathematical optimization to model their problems. However, to solve the problem, they would propose algorithms with tractable solutions due to the NP-hardness of solving the non-convex optimization problems. Optimization problems act as benchmarks to which the performance of lower complexity alternative algorithms is compared. In the following chapters, we always model our problems as Integer Linear Programming (ILP) or as Mixed Integer Linear Programming (MILP) problems and compare their performance with our proposed algorithms based on matching games, deep learning, and reinforcement learning.

In the literature, different methods and mathematical tools have been used to allocate resources. In the next part, we focus on specific low-complexity tools that are relevant to our work in the upcoming chapters.

2.3 Low Complexity Tools for Resource Allocation

Given the NP-Hard nature of solving Integer Programming or non-Convex problems, [18], it it impractical to attempt to solve such problems for real-time tasks. For example, the scheduling of radio resources should be done every Transmission Time Interval (TTI) (i.e., every 1 ms) [13]. The TTI is the time during which a transport block (i.e., the payload of the MAC layer) is transmitted. 5G allows flexible TTI depending on the numerology. Thus, using a shorter TTI duration is allowed to meet low latency requirements. To schedule the transmission at every TTI, low-complexity tools should be used. The challenge is to ensure that these tools, algorithms, or heuristics can perform as close as possible to the optimal solutions. In this part, we focus on the potential of game theory (GT), including Matching games, as well as Machine Learning (ML) tools (i.e., Deep Learning (DL) and Reinforcement Learning (RL)) for resource allocation.

2.3.1 Game theory and matching theory

Game-theoretic approaches could provide sub-optimal solutions but with much lower time complexity [19]. Additionally, Matching-Theory-based games offer solutions for different problems in mobile networks. In this part, various works on Game Theory are surveyed first. Then, Matching-based games are discussed.

2.3.1.1 Game theory

Game theoretic approaches have different use cases in cellular networks [19]. Game Theory (GT) models the interaction between players, where each player aims to optimize its utility/profit. These players may have conflicting interests. This leads to the concept

of non-cooperative Game Theory. The interest lies in finding an equilibrium, such as Nash equilibrium, where the players have no interest in unilaterally deviating from their adopted strategies. Such an equilibrium will be a solution for the game. With the potential existence of multiple equilibria, the challenge is to design a game such that the equilibrium is efficient and close to the optimal solution. On the other hand, some players may realize that by cooperating, they would improve their utilities. In such a case, it is important to find the use cases where cooperation is desired and to design games in which users arrive at a stable coalition. That is to say, members have no incentive to leave their coalitions.

A coalition-formation game has been used for RRH clustering in [73], where a cluster is modeled as a coalition of RRHs. A cluster utility is based on three metrics: The total throughput, power consumption, and frequency of handover of member RRHs. The authors of [108] use a coalition-formation game, where RRHs will join a coalition or leave it if the throughput is increased. To cluster RRHs, the authors of [74] use a potential game to associate RRHs with BBU clusters. To arrive at a Nash equilibrium, the Best-Response-Dynamic algorithm is used. In [65], a bankruptcy game is proposed to model the problem of fair allocation of RBs to the URLLC, eMBB, and mMTC services. In such a game, slices are treated as debtors, and resource blocks are supposed to be the assets owned by a bankrupt company, which is the cloud in this model. When this company is bankrupt, its assets will be distributed to the debtors (i.e., URLLC, eMBB and mMTC slices). Coalitional game theory provides stable solutions to fairly distribute the resources among the debtors. By assuming the company (i.e., cloud) is bankrupt, it would be possible to use solutions provided by coalitional game theory. Aiming to optimize throughput and fairness, the authors in [109], [110] use a coalitional game to model the problem of spectrum sharing between Small Base Stations (SBSs) and Macro User Equipment (MUEs). The players (i.e., SBS and MUEs) aim to form the grand coalition, which means all players join this coalition. The authors demonstrate the stability of this coalition. To improve aggregate throughput, [111] considers a user-centric coalition-formation game. Different users form coalitions and cooperate together to decrease interference and increase their aggregated throughput. To improve spectral efficiency in Device-to-Device enabled Cloud-RAN, [112] uses a Stackelberg game to control the power of the Device-to-Device pair. To maximize the allocation of computing resources in the BBU pool, [89] considers the bargaining concept in cooperative game theory and uses a Nash bargaining solution to allocate resources, ensuring that a minimum amount of resources will be supplied to users while minimizing resource usage. Nash bargaining solution is also used in [113] to allocate the computing resources of the BBU pool, but the authors aim to ensure fair allocation.

In fact, game theory provides frameworks to model the interactions among rational selfish players. Many mobile network problems can be modeled as games, and this would allow solving these problems using the low-complexity game-theoretic approaches.

2.3.1.2 Matching games

The fundamental properties of matching games in wireless networks, along with several applications, have been presented in [114]. Different matching models have been used, including one-to-one, one-to-many, and many-to-many matching models, and algorithms have been proposed to solve these problems. In matching games, players from a set can match with players from another set. Each player would have a preference on whom it wants to match with. The players from one set propose. The players of the other set either accept or reject. A player would propose to the most preferred match. The one who receives proposals will always accept to match with the one it prefers the most.

The stable marriage problem is an example that can help us understand the basics of matching theory [115]. Consider a set of men $\mathcal{M} = A, B, C$ and a set of women $\mathcal{W} = a, b, c$. Each man and each woman has a preference list for the opposite gender; the ones they mostly prefer to be married to. The fundamental solution concept for a matching game is the notion of stable matching. A stable matching is a matching where no blocking pair exists. Suppose that A is matched with a, B is matched with b, and C is matched with c. However, A prefers b over its current match a, and b prefers A over its current match B. Hence the pair (A,b) is a blocking pair as each has the incentive to leave its current partner and switch to another. Hence, a matching algorithm should converge to a stable matching with no blocking pair.

To solve the Stable-Marriage problem, Gale-Shapely proposed the Deferred Acceptance algorithm. In this algorithm, each player from each set creates its preference list. Players from one set make proposals, and the players from the other set accept or reject the proposals. For example, the men can be the ones who propose, and the women accept or reject. The other option is to reverse the roles. Supposing that men are the ones who propose, each man proposed to his most preferred woman. Then each woman examines the proposals she received, accepts the most preferred man among those who proposed, and rejects the other proposals. In the next iteration, men who are already matched will not propose, but those who are unmatched will propose to their second preferred woman. The women who receive the proposals accept the proposals if they are better than their current matches and reject them if they are not. Men who become unmatched will join the next round and propose to the next preferred women on their list. As this algorithm is guaranteed to converge to a stable matching, this iterative process will be repeated until a stable matching is attained.

In [116], a bipartite one-to-one matching game is used to model the assignment of LTE-U users to Unlicensed bands. The famous Gale-Shapley algorithm is used to realize a stable matching. Then, an algorithm is proposed to improve the total throughput by allowing users to switch their assignments if this increases their utilities. In [117], the issue of caching in Small Base Stations is modeled as a many-to-many matching game between Small Base Station and video contents from Service Providers. A many-to-many matching is used in [118]. The matching game is formulated to model the assignment in relay networks between source nodes and sub-channels where Non-Orthogonal Multiple

Access (NOMA) is used as a channel access technology. In [119], the authors consider the issue of resource allocation for a full duplex OFDMA Network. A full duplex base station communicates with half duplex uplink and downlink users. The objective is to maximize the network sum-rate by joint user pairing, sub-channel assignment, and power allocation. Because of the non-convexity of the optimization problem, it has been solved using a cyclic three-sided matching between the sets of transmitting users (i.e., uplink), sub-channels, and receiving users (i.e., downlink).

The matching-coalition approach is used in [107], where a NOMA-based MEC model that aims at improving energy efficiency is considered. A joint optimization problem is formulated where the problems of user association, power control, and computational resource allocation are combined together to optimize energy consumption. Due to the NP-Hardness of the model, the authors propose to use a matching and coalition framework. First, the generalized (resident-oriented) Gale-Shapely algorithm is used to formulate a matching between users and Access Points while neglecting co-channel interference. Then, the coalition approach is used to allow users to enhance their results by considering externalities. Externalities mean that assignments/matchings lead to changes in players' preferences. The preference lists are assumed to be static to ensure the problem theoretically converges to a stable matching. Hence, a preference list that is used at the start of a matching algorithm and that converges theoretically to a stable matching may lead to less optimal results. Accordingly, a coalitional game is proposed allowing the modification of choices, but this time accounting for co-channel interference. Matching theory and coalition games have been used in [69]. The aim is to assign sub-channels to RRH users and D2D users.

As matching games have the ability to model resource allocation problems, we have considered a matching game to allocate radio and computing resources for energy consumption minimization in chapter 4.

2.3.2 Machine learning techniques

Artificial Intelligence (AI) and Machine Learning (ML) have recently shown much potential in wireless networks. ML could provide very efficient low-complexity solutions. Machine learning has gained a big momentum in 5G-related research. Many researchers are examining the ability of ML techniques to solve different highly complex tasks in mobile and wireless networks. ML have three main categories: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, a model receives an input and the desired output. This desired output is called a label. The objective is to learn a function that maps the input to the output. In Deep Learning (DL), this function is a deep neural network. More on DL and neural networks is discussed in Appendix A. Unsupervised learning is used to cluster unlabeled data. This category is out of the scope of our work. In reinforcement learning, an agent interacts with the environment and receives rewards and penalties for its action. Based on the rewards and penalties, it learns what actions can best maximize its cumulative reward. More on reinforcement learning is discussed in Appendix B.

In this part, we first focus on Deep Learning. Then, we move to discuss work related

to Reinforcement Learning (RL).

2.3.2.1 Deep learning

One benefit of Deep Learning is its ability to approximate the performance of highly complex functions. The authors in [120] discuss two applications of deep learning in the physical layer. One of these applications is the approximation of highly-complex functions using a deep neural network. Using Graphical Processor Units (GPU), it could be possible to significantly reduce the function run-time, making it possible to respect the different stringent run-time requirements. The authors in [80] use a deep learning model to predict the processing time at the BBU pool. Moreover, the authors in [121] target optimally allocating resources to D2D users and formulates a Mixed Integer Linear Programming Problem (MILP). It exploits machine learning using support vector machine algorithm to solve the Branch and Bound algorithm quicker. The latter is used by MILP solvers to find solutions for MILP problems. The authors in [122] use a Long Short Term Memory (LSTM)-based DL model to predict mobile traffic, which would help with accurate resource provisioning. In [123], an LSTM-based DL model is used to predict future traffic based on past and current traffic. This model is used by resource allocation algorithms to estimate future requirements and act accordingly. An LSTM-based model is used in [124] to predict a network parameter caller Power Head-Room (PHR). The authors show that the LSTM-based model performed better than other deep learning models. In [125], the handover problem resulting from users' mobility has been considered. A DL model is used to predict the best next base station to which the user should be bound based on the received signal power. Moreover, [126] uses DL for OFDM networks. Usually, a signal encounters attenuation and distortion. These effects have to be measured to recover the signal. Reference signals are used for channel estimation, which will be used to recover the signal back. In contrast, the output of the DL model recovers the input signal without the need for channel estimation. Plenty of works on the usage of DL for different tasks have been presented in [127]. To autoconfigure radio resources for network slices, the authors of [128] use a regression tree-based model to forecast the required ratio of resources of each slice, and provision accordingly. Considering resource orchestration for 5G slices, the authors in [129], [130] evaluate different deep learning models for classification and prediction of ratio of resources to be allocated to slices. They show that Regression Trees outperform other learning models. To allocate radio, computing, and link resources for RAN slices, the authors of [131] formulate two ILP problems to optimize link and computing resources usage and the total users' throughput. A dataset is created by solving the ILP problem for different Inputs. Then the authors use a Bi-Directional LSTM model to learn to output the optimal solutions of the ILP solver.

The proven abilities of deep learning motivate us to consider them in mobile network problems. Inspired by the ability of deep learning to learn to do complex tasks in a shorter amount of time, we use in Chapter 5 a low-complexity deep learning model that aims to learn how to perform close to the ILP problem-solver. However, it outputs results in much less time.

2.3.2.2 Reinforcement learning

Reinforcement Learning (RL) is being used widely to learn autonomously by interacting with the environment. Given a specific state/observation, an agent executes an action and receives a reward or a penalty from the environment. Accordingly, the agent learns to execute the actions that maximize rewards.

The authors of [132], [133] tackle the problem of the coexistence of URLLC and eMBB and formulates a Q-learning algorithm that aims at decreasing latency and increasing reliability for URLLC users without hindering eMBB throughput. The algorithm is shown to outperform the proportional fairness algorithm.

The authors of [134] deal with the resource allocation of RBs to tactile applications. The goal is to maximize throughput and minimize latency since tactile applications are delay-sensitive and data-intensive. The authors formulate the problems as an optimization problem, then propose to use the Q-Learning algorithm, namely TMQ, to learn an efficient allocation of RBs.

The paper [135] uses the Q-learning algorithm to deal with inter-beam interference by jointly applying user association and power allocation to beams. The reward reflects the quality of the average SINR in each gNB. The performance according to sum-rate, latency, and the packet drop rate is measured and compared to a baseline algorithm that uses uniform power allocation. In the same context, [136] considers the problem of beam clustering and RB allocation for multi services users URLLC and eMBB. They propose a clustering algorithm with a Long-Short-Term-Memory LSTM-based Deep Reinforcement Learning (DRL) to cluster users into beams before allocating Resource Block Groups (RBGs) to them. The algorithm outperforms a baseline algorithm that uses K-means for clustering and proportional fairness for RBG allocation. [137] uses Transfer reinforcement learning to learn how to associate users to gNodeBs and to select the number of beams to be used. Since inter-cell interference control have similarities with to inter-beam interference control, the knowledge learned when dealing with inter-cell interference is used to learn to associate a user with a beam. Instead of learning from scratch, Transfer Learning (TL) allows using the knowledge about one task to solve a different task, especially when the two tasks have similarities.

In [138] and [139], resource block allocation has been considered for mission-critical services and micro-grid communication, where the goal is to minimize the delay. The base stations are modeled as agents that interact with the environment by selecting an action given their state and receiving a reward plus their new state. Based on the rewards, the agents learn the quality of the action they take. Instead of storing the quality value of each action at each state, which may require a tremendous amount of memory, the authors use an LSTM network that approximates the Q-value of each state-action pair and also takes into consideration the effect of actions done in the previous time steps.

RL-based algorithms are widely used nowadays for Open-RAN. [140] investigates the usage of dynamic functional splitting in Open-RAN, where the functions can be split among the CU and DU depending on the traffic type. To minimize energy consumption costs, and given that both solar cells and electrical grids coexist, an optimization

problem is formulated to prioritize the usage of solar cells. To solve this problem, RL algorithms are used. The results convey the ability of an RL algorithm called SARSA (i.e., SARSA stands for State Action Reward State Action) and Q-learning to reduce energy consumption and costs compared with Centralized-RAN and Distributed-RAN.

The goal of [141] is to maximize throughput and fairness and to minimize packet loss rate. The authors use a policy-gradient-based RL model with a flexible neural network to assign users to RBs. [142] aims to satisfy users' requests for radio resources, latency, computing resources, and transmit power. It uses an RL algorithm that aims to maximize the total number of satisfied requests. A state reflects the available communication (frequency-time blocks), computational and transmission power resources, and the action is to allocate resources to requests.

[143] uses federated reinforcement learning where the goal is to learn the optimal actions regarding the selection of BS association and the optimal RB to maximize throughput while reducing frequent handover in a dense 5G system. In federated learning, models are trained using local data, and the local models are shared with a central entity that aggregates the model's weights. The RL algorithm is based on a Dueling Deep Q-Network where the Q network consists of a common part, a value function, and an advantage. Each user is an RL agent that trains itself and partially shares its model; the common and state value parts are shared with the global model, aggregating the weights and sending them back. The advantage is kept unique for each user to account for user-dependent variations.

To minimize queuing latency, [145] uses Multi-Armed Bandit to select the combination of TTI length and subcarrier spacing. In [48], the authors demonstrate the feasibility of near-real-time radio access control using deep reinforcement learning. They test the RL-based scheduling algorithms on a large-scale Open-RAN-compliant softwarized cellular network called Colosseum.

Various studies demonstrate the ability of RL-based solutions to replace traditional network functions. In [146], an RL model has been used to satisfy the demands of users from different slices/services. These demands include communication and computational resource requests. In the context of network slicing, [147] proposes two RL algorithms to realize upper and lower control. In particular, a Deep Deterministic Policy Gradient (DDPG) is used for the lower control: Resource Blocks (RBs) allocation and power allocation. On the upper level, a Double Deep Q-Network-based RL is used to learn the optimal RAN slicing strategies. Authors of [148] used Policy Gradient RL to learn the optimal functional split to minimize the computational and routing cost.

RL methods have shown a lot of potential in solving mobile network problems, especially in environments with unknown dynamics or dynamics that continuously change. This inspires us to study, in Chapter 5, the ability of a policy-gradient-based RL method to optimize the profits of an operator.

Finally, Table 2.2 lists the papers that we have discussed in this chapter and deal with radio or computing resource allocation along with the objectives of these papers.

Optimization Objective	Radio Resources	Computing Resources
Throughput	[64]–[69], [72], [73], [76]–[78], [90]–[94], [103], [104], [111], [112], [116], [118], [119], [131]–[137], [141], [143], [144]	[13], [75], [87], [91]–[94], [103], [104], [131], [144]
Power Consumption or Energy Efficiency	[71], [74], [90], [95]–[100], [102], [106], [107], [144]	[85], [90], [95]–[100], [102], [106], [107], [144]
Delay/Latency	[64], [66], [77], [105], [132]–[134], [136], [138], [139], [145]	[87], [88], [105]
Fairness/Satisfying Requests	[65], [69], [141], [142], [146], [147]	[113], [142], [146]
Costs	[83], [90]–[92]	[85], [91], [92]
Resource Utilization	[142]	[89], [142]

Table 2.2: A comparison of the different objectives of the papers aiming at radio and computing resources allocation in future Radio Access Networks

2.4 Summary

In this chapter, we have surveyed the architecture of Cloud-RAN and Open-RAN. We have discussed the advantages and challenges these architectures face. Then, various research works related to radio and computing resource allocation have been presented. We have also surveyed the different tools used for resource allocation in Radio Access Networks: Game Theory, Matching Theory, Deep Learning, and Reinforcement Learning. The discussed works and tools set the basis for radio and computing resource allocation in the following chapters.

Chapter 3

Coordination between Radio and Computing Resource Schedulers in Cloud-RAN

Cloud-RAN brings the opportunity to centralize and virtualize the baseband processing of many base stations by decoupling the Base Band Units (BBUs) from the Radio Remote Heads (RRHs). With the centralization and virtualization of baseband functions, it is important to optimize the allocation of different resources including computing resources. While efficient provisioning of resources should be made, unexpected demands may suddenly occur, leading to resource scarcity. In this chapter, we consider the case where the limited computing resources become insufficient to process all the frames from all users. Given that both the required processing time and the throughput depend on the Modulation and Coding Scheme (MCS) index, we propose a coordination scheme between radio and computing schedulers that allows adjusting the MCS index so that the processing time requirements is lower enough for the frame to be processed. Hence, a frame can be transmitted and processed but with lower throughput. We model this coordination scheme using Integer Linear Programming (ILP) problems that optimize throughput and fairness. Additionally, we propose lower complexity heuristics that perform close to the ILP problems. Moreover, given the existence of different services in 5G such as enhanced Mobile BroadBand (eMBB) and Ultra Reliable Low Latency Communication (URLLC), each with different Quality of Service (QoS) requirements, we test the proposed coordination solutions under such multi-services scenario.[1]

3.1 Introduction

The proposed centralization of baseband functions in Cloud RAN opens up a lot of opportunities. Instead of deploying hardware that is likely to be underutilized at each

[1]The contents of this chapter are presented in [1], [6], [7].

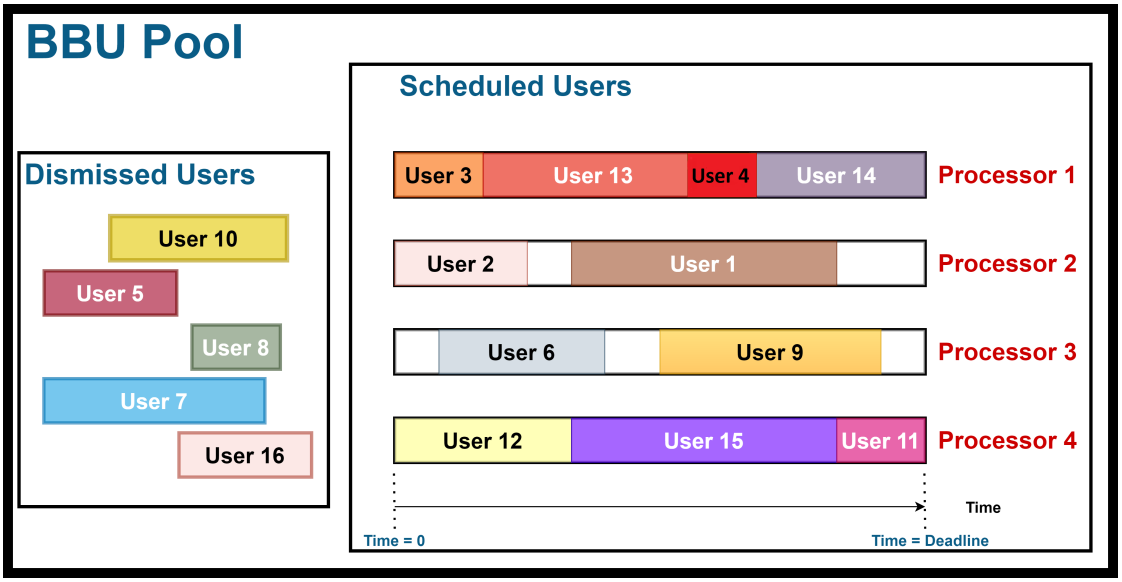


Figure 3.1: Scheduling the processing of users in the BBU pool. For an overloaded BBU pool, some users will be dismissed

.

remote Base Station (BS), the baseband functions can be processed in the cloud in centralized data centers. This offers operators the chance to optimize the network better, reduce energy consumption, and reduce costs. The operator can reserve resources for peak and non-peak hours and dynamically allocates these resources to the remote BSs according to their needs. One problem that could be faced is the underprovisioning of resources, either because of non-optimized provisioning or because of a sudden unexpected increase in demand. In such cases, computing resources of the BBU pool may become limited as they are shared among a large number of Radio Remote Heads (RRHs) connected to the BBU pool. For that, it is necessary to efficiently manage the BBU pool, especially when it is overloaded with many RRHs. The BBU pool has to ensure it maintains the ability to process users' data before exceeding the deadline imposed by the MAC-layer mechanism, known as Hybrid Automatic Repeat Request (HARQ). Knowing that the processing times of users' data depend on radio parameters such as the Signal to Interference plus Noise Ratio (SINR) and the Modulation Coding Scheme index (MCS), the choice of the MCS index affects the users' throughput by specifying the modulation and the code rate to be used [80] [13]. This raises the coordination between radio and computing schedulers as a candidate to efficiently manage the resources of an overloaded BBU pool. This coordination should help achieve the required Quality of Service (QoS) for users and ensure a good level of fairness.

In this chapter, we propose different schemes that implement coordination between radio and computing schedulers in Cloud-RAN. Aiming to optimize throughput and fairness, these coordination schemes permit the adjustment of users' MCS indexes to ensure the processing of their data in the BBU pool. In other terms, these schemes allow

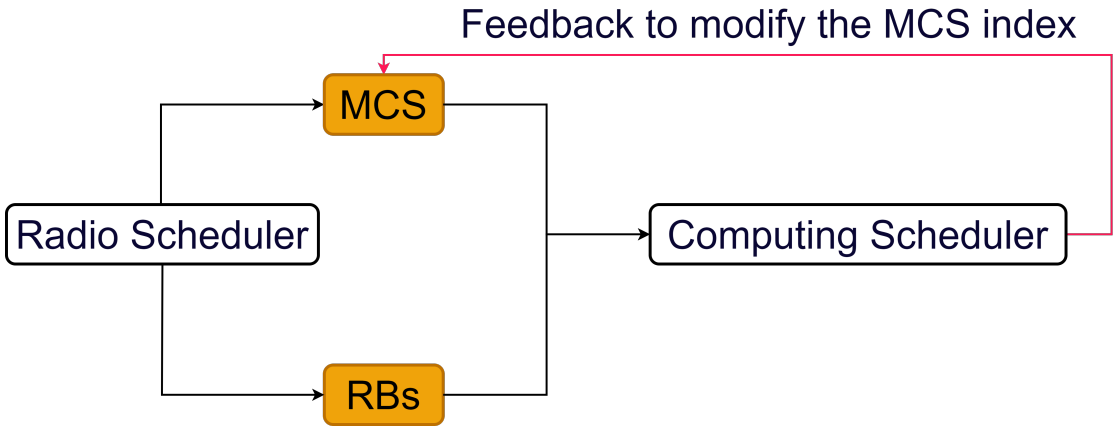


Figure 3.2: The proposed coordination: The radio scheduler receives feedback from the computing scheduler to adjust the MCS index

.

the radio scheduler to assign users' MCS indexes based not only on the radio quality but also on the availability of computing resources. Figure 3.2 shows the proposed coordination scheme. Thus, when the BBU pool gets overloaded, it will not be able to process all users' data while respecting the HARQ deadlines requirements (Fig. 3.1). Users of non-processed data should retransmit the data as part of the HARQ mechanism. Such retransmission turns out to be an energy-inefficient phenomenon that reduces network performance and wastes radio resources. Hence, instead of retransmitting the data, it could be more efficient to reduce data processing times, and this can be done by decreasing the corresponding MCS indexes of users [13]. While lowering the MCS index will decrease the radio throughput and the required processing time, it should guarantee the ability to process users' data instead of dropping them. We model the coordination schemes as ILP problems which are known to be NP-Hard [18]. Furthermore, we propose three low-complexity heuristics as alternatives to solving the ILP problems solutions.

The fifth-generation (5G) of mobile communications has been designed to accommodate different coexisting services with heterogeneous requirements such as enhanced Mobile Broadband (eMBB) and Ultra-Reliable Low-Latency Communication (URLLC) scenarios [149]. While eMBB aims at achieving very high data rates, URLLC supports low-latency transmissions with very high reliability. In a multi-services scenario, once URLLC frames arrive at the BBU pool, they should be scheduled without delay due to their strict latency requirement. Hence the BBU pool may preempt the processing of eMBB data and process URLLC data instead. For that, we analyze the performance of the heuristics in a multi-services scenario when the existence of URLLC traffic harms eMBB users.

Characterizing the functions of the BBU has been the interest of various research works. A processing time model has been proposed in [81], and it models the processing time as a function of the MCS index, the number of RBs, and the CPU frequency. The authors in [86] considered applying the decoding function in parallel and proposed two

algorithms that ensure parallelism. In [13], the authors propose two different computing scheduling algorithms to schedule the processing of RRHs' data arriving at every transmission time interval (TTI). These algorithms aim to increase the number of correctly decoded sub-frames and the system throughput. These papers limited their study to one service type and excluded URLLC service. In [144], the authors consider the issue of joint radio and computing resources allocation in Cloud-RAN. They map users to BBUs running as virtual machines, then the radio resource allocation is carried out by controlling each user's beamforming vector and the power. They model the problem and devise two algorithms to allocate the radio and computing resources. However, they did not quantify improvement of the joint allocation versus the non-joint. In the same context, the authors in [101] propose an optimization problem that tries to ensure the stability of RRHs and BBU queues, then they use auction theory. While the auction game represents a sequential allocation for radio and computing resources allocation problem, our work focuses on the coordination problem, which permits feedback between radio and computing schedulers.

The main contributions of this chapter are recapped here:

1. We propose a coordination scheme between radio and computing schedulers that considers the availability of computing resources when assigning MCS indexes for transmissions.

2. We propose three ILP-based algorithms that enable coordination: Maximize Total Throughput (MTT), Admit All Users (AAU), and Maximize Users' Satisfaction (MUS).

3. We analyze the performance of the coordination algorithms with respect to no-coordination counterparts.

4. We propose heuristics that perform close to the ILP-based algorithms but significantly reduce execution time.

5. We analyze the performance of the coordination policies when eMBB and URLLC services coexist, and we study how prioritizing URLLC frames over eMBB ones affects the performance of the heuristics.

3.2 Context and Problem Formulation

The system under study consists of a set of RRHs connected to a centralized BBU pool composed of homogeneous CPU cores with the same execution speed. We suppose each RRH has one antenna.[2] Furthermore, we assume there is no bottleneck at the fronthaul links connecting the RRHs to the BBU pool. As the uplink processing time is at least 7 times larger than that in downlink [80], it is thus a dominating issue for the

[2]Modifying the number of antennas should not affect the tendency of our results, except that it would only overload the BBU pool at a lower number of RRHs.

Table 3.1: Summary of the general notations

Parameters	***Definition***
$AdjMargin$	Adjustment Margin; sets a limit on how much the MCS index can be adjusted
$AvTime(c)$	Available processing time on CPU $c \in \mathcal{C}$
$b_{u,m}$	Data length (in bits) of user $u \in \mathcal{U}_r$ using an MCS index $m \in \mathcal{M}$ during one TTI
$b_{u,max}$	Data length (in bits) of user $u \in \mathcal{U}_r$ using its maximum MCS index $M_{u,max}$ during one TTI
$\mathcal{C}$	Set of CPU cores in the shared BBU pool (multi-core data center).
d	Processing time deadline
$\mathcal{M}$	Set of MCS indexes that can be used in the system
$M_{u,max}$	Maximum MCS index user $u \in \mathcal{U}_r$ may use
$MaxMCS$	$\max(\{M_{u,max} : u \in \mathcal{U}_r, r \in \mathcal{R}\})$
$N^{RE}_{MiniSlot}$	The number of Resource Elements (OFDM symbols) per a mini-slot per 1 sub-carrier
N^{SC}_{RB}	The number of sub-carriers per a Resource Block
N_u	The number of RBs used by user $u \in \mathcal{U}_r$
OH	Transmission Overhead of control data
P_t	Transmission Power for each user.
Q_m	The number of bits per symbol.
$\mathcal{R}$	Set of RRHs
$\mathcal{U}_r$	Set of users for each RRH $r \in R$
$selectedCPU_u$	Selected CPU to process the data of user $u \in \mathcal{U}_r$
$selectedMCS_u$	Selected MCS index for user $u \in \mathcal{U}_r$
$SysMCS$	Adjustable system-wide MCS index, no user is allowed to use a higher one
$t_{u,m}$	Data processing time of user $u \in \mathcal{U}_r$ having an MCS index $m \in \mathcal{M}$
$x^c_{u,m}$	A binary variable that assigns the data of user $u \in \mathcal{U}_r$ having an MCS index m to the core $c \in \mathcal{C}$

BBU pool's bottleneck. For that, we focus on the uplink direction where users connected to each RRH share the available resource blocks that can be used for transmission at the start of every transmission time interval TTI. The RRHs send the received users' data to the BBU pool, which has to process all the incoming data from the RRHs' users within a specified amount of time equal to $2ms$, as instructed by (HARQ) mechanism.

In HARQ, the data sent from a user are transmitted by the user over 1 ms (i.e., 1 TTI), received by the BBU, processed by the BBU, and then the BBU send the acknowledgment back. The sender should receive the acknowledgment in no more than

$8ms$. The delay in the uplink direction includes the propagation delay, the fronthaul delay, 1 ms for the acquisition time of a user frame given that it takes a TTI duration for the frame to fully arrive, and the processing time at the BBU pool equals 2 ms. Then the acknowledgment has to be prepared and sent back. 1 ms is available for processing the acknowledgment at the BBU pool before transmission, the fronthaul and propagation impose additional delay, and finally 2ms are left to process the acknowledgment frame at the user. The deadline for completing the BBU processing of user's data in the uplink is equal to 2ms after deducting the expected latency in fronthaul, transmission, acquisition, etc. [86].

We further consider that a user's MCS index is determined by jointly considering the channel conditions of all the RBs in the associated RRH. This allows the radio scheduler to attribute the same modulation and coding scheme (MCS) index to a given user over all its resource blocks. It is worth mentioning that the radio scheduler attributes the maximum allowed MCS index to a given user by considering its radio conditions measured by the user's equipment. More specifically, the Channel Quality Indicator (CQI), which is related to the Signal-to-Noise-and-Interference ratio, is sent by the user equipment (UE); the CQI carries information on how good/bad the communication channel quality is [80]. Based on this indicator, the radio scheduler determines the maximum allowed Modulation Coding Scheme (MCS) index for each user. As shown in [80], the processing time of the BBU sub-functions (more particularly, the decoding function) strongly depends on the MCS index; it increases as the MCS index increases. Hence, if the BBU pool is overloaded and all users use their maximum allowed MCS index, the BBU pool will fail to process all the incoming users' data by the specified deadline. We note that if the BBU pool fails to deliver the HARQ-acknowledgment before the deadline, users of non-processed data should retransmit the data.

Next, we present three Integer-Linear-Programming coordination solutions, each with a different objective to maximize.

3.2.1 Notations

Let $\mathcal{R}$ be the set of RRHs, $\mathcal{U}_r$ the set of users connected to RRH r, $\mathcal{M}$ the set of possible MCS indexes that can be assigned for the radio transmission by the radio scheduler, and $\mathcal{C}$ be the set of homogeneous CPU cores in the BBU pool. For each RRH r, the coordination policy must attribute to each user $u \in \mathcal{U}_r$ an MCS index $m \in \mathcal{M}$ that is lower or equal to the maximum allowed MCS index which would initially be chosen by the radio scheduler $M_{u,max}$. We assume that the Signal-to-Interference-plus-Noise (SINR) is known at the base station, and the maximum allowed MCS is chosen accordingly. On the other hand, we assume that each user has pre-assigned number of resource blocks, such that all the RBs of an RRH are allocated to its users. Based on the selected index m, user u transmits an amount of data that is equal to $b_{u,m}$; the latter is determined according to Table 7.1.7.2.1-1 in [150] that maps the transport block size (i.e., the payload that can be carried by the physical layer) to the modulation and coding scheme index and the number of resource blocks. Besides, the time required for processing user's $u \in \mathcal{U}_r$ data on the BBU pool is equal to $t_{u,m}$; the latter is determined

using the formula in [81] where the Open Air Interface (OAI) RAN simulator is used. The formula is given as follows:

$$t_{u,m[\mu s]} = \frac{N_u}{f^2_{[GHz]}} \sum_{i=0}^{2} \alpha_i m^i \tag{3.1}$$

- $t_{u,m}$: processing time of user $u \in \mathcal{U}_r$ using MCS index m.
- N_u: The number of RBs used by user $u \in \mathcal{U}_r$.
- f: the clock frequency of the CPU
- m: the MCS index used by user $u \in \mathcal{U}_r$
- α_i: polynomial coefficients

According to [81], the values of α_i corresponding to the overall uplink processing time are: $\alpha_0 = 35.545$, $\alpha_1 = 1.623$, and $\alpha_2 = 0.086$. Arbitrarily, we set the CPU frequency to 4GHz. It is worth mentioning that the processing times strongly increase with the MCS index for a fixed number of RBs. However, we note that many more bytes are processed for larger MCS, and the processing time per byte decreases as the MCS index increases.

We suppose that each user transmits its data with a constant power P_t. Table 3.1 presents the notations used throughout this chapter.

3.2.2 The ILP coordination schemes

To model the proposed coordination solutions, we consider three Integer Linear Programming (ILP) optimization problems in which the coordination management entity acts as a single centralized decision-maker.

1. *Maximize Total Throughput (MTT)*: As one of the major objectives in 5G networks is to provide high overall throughput, the first solution we examine tackles this issue by solving the following ILP optimization problem:

$$\text{maximize} \quad \sum_{r\in\mathcal{R}} \sum_{u\in\mathcal{U}_r} \sum_{m\in\mathcal{M}} \sum_{c\in\mathcal{C}} x^c_{u,m} b_{u,m} \tag{3.2}$$

$$\text{subject to} \quad x^c_{u,m} \in \{0,1\},\ \forall r \in \mathcal{R}, u \in \mathcal{U}_r, m \in \mathcal{M}, c \in \mathcal{C} \tag{3.3}$$

$$\sum_{c\in\mathcal{C}} \sum_{m\in\mathcal{M}} x^c_{u,m} \leq 1,\ \forall r \in \mathcal{R}, u \in \mathcal{U}_r \tag{3.4}$$

$$x^c_{u,m} = 0,\ \forall r \in \mathcal{R}, u \in \mathcal{U}_r, c \in \mathcal{C}, m > M_{u,max} \tag{3.5}$$

$$\sum_{r\in\mathcal{R}} \sum_{u\in\mathcal{U}_r} \sum_{m\in\mathcal{M}} x^c_{u,m} t_{u,m} \leq d, \forall c \in \mathcal{C} \tag{3.6}$$

where $x^c_{u,m}$ is a single binary variable equal to 1 if the data of user $u \in \mathcal{U}_r$ is coded using MCS $m \in \mathcal{M}$ and is processed on CPU core $c \in \mathcal{C}$. If not, it is equal to 0.

The objective function (1) maximizes the sum of overall users' throughput in the system or total system throughput. MTT solution possesses the following constraints: (2) ensures that the decision variable $x^c_{u,m}$ only takes values 0 or 1. Equation (3) ensures that the data belonging to a given user $u \in \mathcal{U}_r$ are encoded using at most one MCS index m and are processed on at most one CPU core c. Equation (4) ensures that the decision-maker should not assign to any user an MCS index higher than the maximum allowed one because a higher MCS index would increase the decoding error to more than the upper limit of 10% [151]. Finally (5) ensures that the data to be processed on core c have to finish before the deadline d. Intuitively, MTT favors users with high MCS indexes as they possess high throughput and hence it sacrifices users with lower MCS indexes.

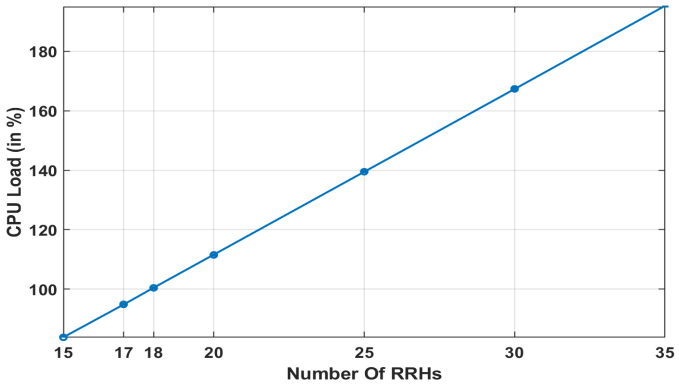
2. *Admit All Users (AAU)*: Instead of privileging users with high throughput over others as done in MTT, AAU solution keeps the same objective function of maximizing the overall system's throughput while ensuring that all users have to be scheduled on the BBU pool. Compared to MTT, there is a slight modification in only one constraint: the single-core and MCS assignment constraint (3), while the objective function and other constraints remain the same as in MTT. This constraint can now be modified as follows:

$$\sum_{c \in \mathcal{C}} \sum_{m \in \mathcal{M}} x^c_{u,m} = 1, \ \forall r \in \mathcal{R}, u \in \mathcal{U}_r \tag{3.7}$$

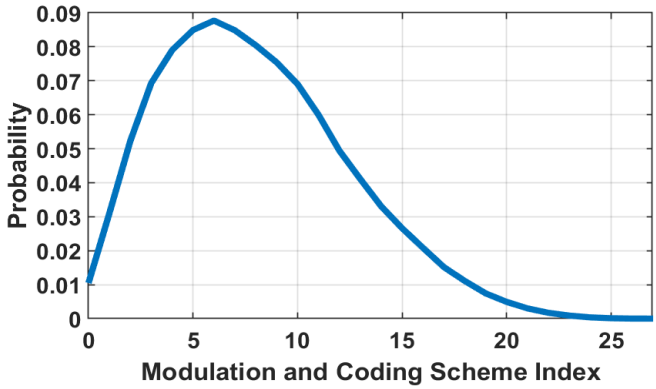
It is worth mentioning that since AAU solution requires all users to be scheduled on the BBU pool, there is an upper bound on the number of users that can be admitted in the BBU pool depending on the capacity of the CPU cores in the BBU pool. When the BBU pool becomes highly overloaded, AAU solution schedules the users and assigns low MCS indexes to ensure all of them are admitted into the BBU pool. However, sometimes even the lowest MCS indexes are not enough to ensure the admission of all users, and in that case, AAU solution turns out to be infeasible.

3. *Maximize total Users' Satisfaction (MUS)*: Intuitively, the two previous solutions do not ensure a good level of fairness among the users. For that reason, we propose the MUS policy that aims to maximize the total users' satisfaction ratio and ensure a good level of fairness. We define the user satisfaction ratio as the ratio of the throughput achieved when the user operates using an adjusted MCS index, to the maximum throughput obtained when operating using the maximum allowed MCS index. The objective function of MUS solution is the following:

$$\sum_{r \in \mathcal{R}} \sum_{u \in \mathcal{U}_r} \sum_{m \in \mathcal{M}} \sum_{c \in \mathcal{C}} x^c_{u,m} \times \frac{b_{u,m}}{b_{u,max}} \tag{3.8}$$



(a) BBU pool load as a function of RRHs' number



(b) Probability distribution function of MCS indexes as in [13]

Figure 3.3: CPU load and MCS Distribution

It ensures that when a given user $u \in \mathcal{U}_r$ has a maximum allowed MCS index $M_{u,max}$, the coordination entity between radio and computing schedulers does not assign him an MCS index that deviates much from the maximum allowed MCS index. Hence, the user satisfaction ratio is maximized when the maximum allowed MCS index is used. We note that MUS solution maintains the same constraints as those of MTT.

3.3 Simulation Settings and Performance Metrics

In this section, we present the simulation settings and performance metrics that we use to evaluate the performance of our proposed coordination solutions.

3.3.1 Simulation environment

We consider a BBU pool composed of 4 CPU cores that process the incoming data from the RRHs' users. We vary the number of RRHs connected to the BBU pool from 15 to 35, which in turn varies the load of the BBU pool. Supposing that each user

Table 3.2: Simulation Parameters

Number of CPU cores	4
Number of RBs per BS	100
Number of RBs per User	10-30
Number of RRHs	15-35
Deadline	2ms
Processing Time Model	[81]
MCS Model	[13]
Throughput Model	[150]

operates with its maximum allowed MCS index, Fig. 3.3a provides the BBU pool load as a function of the number of RRHs. The figure shows that the load varies from 83% to 195%. It is worth mentioning that the BBU pool starts to be fully loaded when the number of RRHs connected to the BBU pool is 18. Intuitively, our priority is to focus on such an overloaded BBU pool scenario because the case of a non-fully-loaded BBU pool allows all the users to operate using their maximum allowed MCS indexes; hence, decreasing their MCS indexes will not be beneficial in this case. The RRHs operate using a 20 MHz bandwidth, so the number of available physical resource blocks (RBs) per TTI equals 100. The number of users per RRH is variable. A uniform random variable between 10 and 30 is used to sample the number of RBs. The sampled value is assigned to a user, and this user is assigned to the RRH. Then another number of RBs is sampled from the remaining RBs and added to a new user, and the process is repeated. When the number of remaining RBs is lower than 30, a value between 10 and the number of remaining RBs is sampled. If the remaining RBs are less than 10, they are fully assigned to a new user. In order to use a real traffic distribution as a function of the MCS indexes, we consider the same probability distribution function as in [13] that is obtained using real measurements from [152]; this distribution is shown in Fig. 3.3b, and we use it to sample the maximum allowed MCS indexes for the different users.

Moreover, each user's throughput is determined using the technical specification of ETSI [150]. The throughput of one user is determined by mapping its number of allocated resource blocks and its MCS index to the transport block size TBS (i.e., the data payload that can be carried by the physical layer). We note that the TBS of a user increases with the MCS index or the number of resource blocks. We get the throughput of each user by dividing its TBS by the transmission time interval (TTI) that is set to 1 ms. Additionally, we use MATLAB to code and run the simulation, and we use CPLEX MILP solver interfaced with MATLAB to solve the ILP problems. The simulation parameters are summarized in Table 3.2.

3.3.2 Performance metrics

We compare the performance of the three proposed scheduling policies using different performance metrics, and we monitor the evolution of their performance as a function of the BBU pool load. The performance metrics used in this study are the following:

- Average throughput: The average user throughput.
- The number of admitted users: The number of users scheduled in the BBU pool and processed before the deadline.
- Fairness: We used the Jain's fairness index J_I [153] to compare the fairness of the three policies; it is given by:

$$J_I = \frac{\left(\sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r} s_u\right)^2}{\left(N \times \sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r} s_u^2\right)} \tag{3.9}$$

  For each user $u \in \mathcal{U}_r$, s_u is its satisfaction ratio (i.e., the ratio of the attained throughput to the maximum achievable throughput achieved when using the maximum allowed MCS index). Also, N is the total number of users from all RRHs. A user is most satisfied if it gets the maximum throughput that can be achieved, i.e., being assigned its maximum allowed MCS index.
- Wasted power: This metric shows the ratio of the wasted power to the total emitted power. The power is useful when the data carried by the signals get processed before the deadline of 2ms. In contrast, data that is not processed before the deadline must be retransmitted. Hence the signal, and consequently its power, will be wasted.

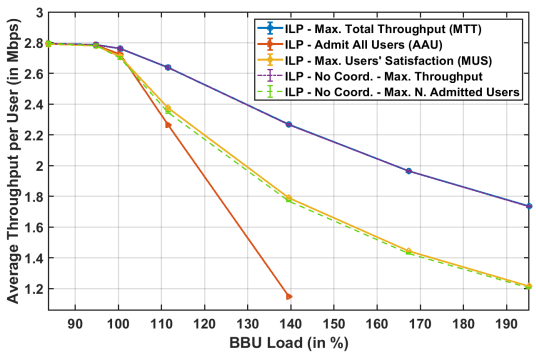
3.4 Performance Evaluation of the ILP-Based Coordination policies

In the following subsections, we evaluate the performance of the three different scheduling solutions (i.e., MTT, AAU, MUS) with respect to the four metrics presented in Section 3.3.2. We limit the study and analysis to only one TTI instance. In addition, we compare our approaches to two other basic approaches in the literature [13] that do not consider any coordination between radio and computing schedulers. Their objectives are to maximize throughput and to maximize the number of admitted users, respectively. It is worth mentioning that 100 simulations were performed, and the confidence intervals of 95% are provided in the following results.

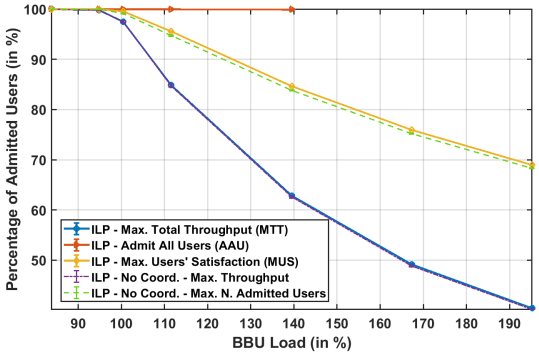
3.4.1 Average throughput per user

Fig. 3.4a shows the average throughput per user obtained by each proposed approach as a function of the BBU pool load. The MTT solution clearly outperforms the other two coordination solutions in average throughput when the BBU pool load surpasses 100%. In contrast, when the BBU pool load is less than 100%, the different solutions achieve the same performance as the BBU pool can process all users' data while operating using their maximum allowed MCS indexes.

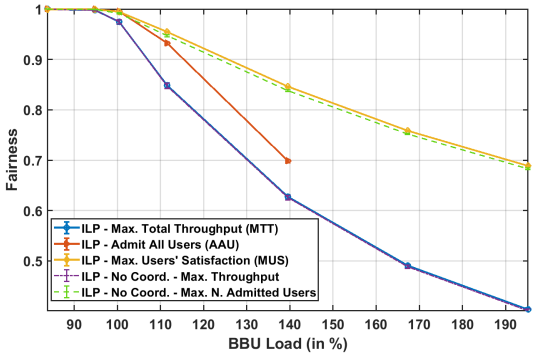
On the other hand, AAU shows the worst performance among the coordination policies with respect to the throughput metric because it requires the admission of all users from all RRHs to the BBU pool; hence the radio scheduler has to decrease the MCS indexes (which lowers the throughput) of many users so they may be scheduled



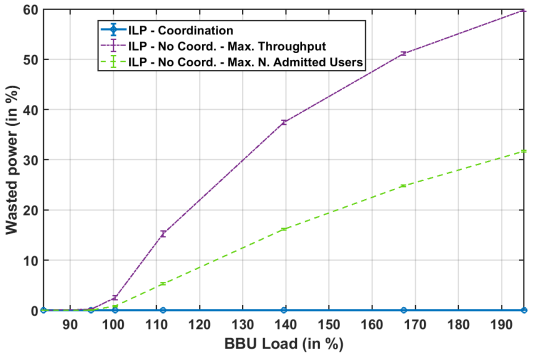
(a) Offered throughput as a function of BBU pool load



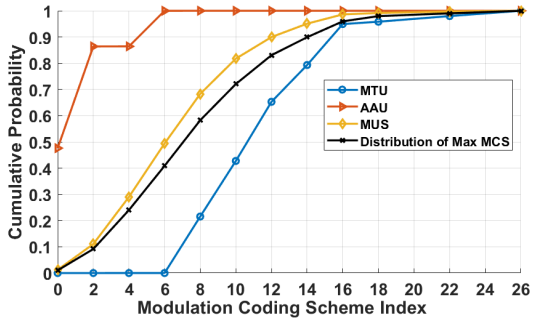
(b) Admitted Users (in %) as a function of BBU pool load



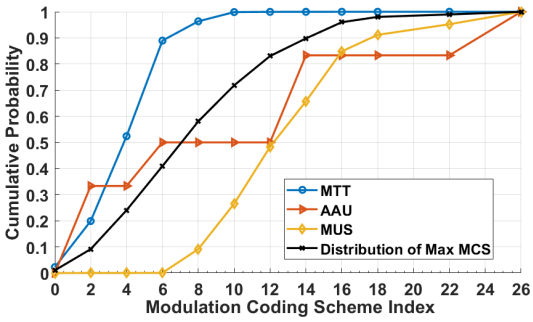
(c) Jain's Fairness Index as a function of BBU pool load



(d) Wasted Power (in %) as a function of BBU pool load



(e) Cumulative Distribution Function of the users' MCS indexes when the RRHs number is equal to 25



(f) Cumulative Distribution Function of the maximum MCS indexes of dismissed users when the RRHs number is equal to 25

Figure 3.4: Performance evaluation of the different scheduling solutions

on the BBU pool. Consequently, this severely degrades the total system throughput; solving AAU when the BBU load exceeds 140% is infeasible.

The MUS policy scores in-between results compared to the other policies. When

comparing MTT and MUS policies to the non-coordination schemes, we find that MTT results almost coincide with Maximizing Throughput objective, and MUS results almost coincide with those found for Maximizing the Number of Admitted Users; the proposed coordination brings a slight improvement of less than 1%. The coordination policies present an advantage over the non-coordination schemes. Due to the allowed reassignment of MCS indexes, coordination schemes can use the CPU idle time to allocate more users and improve the system throughput. When it is impossible to use the maximum MCS for some users, it could be possible to use a lower MCS index making a place for one or more additional users. As can be seen in constraint (3.4), this extra degree of liberty compared to the non-coordination schemes (where a single MCS may be assigned) can only be beneficial, producing higher throughput and number of served users or at least as good as the no-coordination.

3.4.2 Number of admitted users

Fig. 3.4b shows the percentage of admitted users as a function of the BBU pool load; this percentage is relative to the total number of users from all RRHs. Obviously, the AAU policy ensures the admission of all users (i.e., the percentage of admitted users is always 100%) as long as the problem is feasible, and it clearly outperforms all other solutions with respect to this metric. With respect to AAU, we see that the performance of MTT policy drops, admitting only 40.2% of users for an RRH number of 35 per BBU pool (i.e., equivalent to a BBU pool load of 195%). MTT tends to admit fewer users with higher throughput than to admit more users with low throughput. For the third policy (MUS), its performance gradually drops until it reaches 68.9% when the BBU pool load is 195%. Again, the performance of the third policy comes in-between that of the other two policies, as it aims to maximize fairness without necessarily admitting all users. In comparison with no-coordination schemes, we again notice a slight improvement, as explained in 3.4.1.

3.4.3 Fairness index

The performance of the different policies is also measured with respect to the fairness in resource distribution, and we use for that purpose Jain's fairness index J_I as defined in Equation 3.9. We note that $J_I = 1$ is the maximum fairness value while $J_I = 0$ expresses the most unfair scenario. Here again in Fig. 3.4c, for a load less than 100%, all users may use their maximum MCS index resulting in a Jain's fairness index equal to 1 for all policies. However, when the number of RRHs per BBU pool increases, the fairness index declines.

The MUS policy outperforms all the other policies in terms of overall fairness as it maximizes users' satisfaction rate. On the contrary, the MTT policy is the least fair among the coordination policies. The reason is that it favors users who can achieve high throughput (i.e., those with high MCS indexes) and sacrifices those with lower MCS indexes that provide lower throughput. The AAU policy, with its objective of admitting the maximum number of users, for some time achieves the highest Jain index. Then beyond a certain load level, this objective results in an unfair share; adding more users

to the system worsens the fairness index since many users would take a small portion of their maximum allowed throughput.

In comparison with no-coordination schemes, here again, we observe slight improvements, as explained earlier. The coordination schemes are slightly fairer compared to the no-coordination because they allow more users to get a chance to transmit by adjusting their MCS indexes to lower values. In contrast, the no-coordination schemes ignore these users since it is impossible to process their data if the maximum MCS is used. As a result, the coordination schemes are fairer than their no-coordination counterpart.

We recall that, when analyzing all the performance metrics, all the policies perform similarly when the BBU load is less than 100%. However, they behave differently when the BBU pool becomes fully loaded. When the BBU pool becomes overloaded, the different policies begin to adjust the MCS indexes of users since it is impossible to fit all users if they operate using their maximum MCS indexes. The selection of the users to be scheduled and their corresponding MCS indexes differentiate the coordination policies one from another.

3.4.4 Wasted power

In Fig. 3.4d, we plot the percentage of wasted power to the total emitted power. We define the wasted power as the power used to transmit user frames that the BBU pool will not process before the HARQ deadline due to the lack of processing resources and that will thus be retransmitted. We consider that all users use a constant transmit power P_t for signal transmission. In the coordination schemes, only users whose frames can be processed (possibly with a reduced MCS index) before the HARQ deadline would transmit data. Hence, they present a 0% waste of transmission power.

When we consider the no-coordination schemes, transmission decisions are taken by the radio scheduler alone without knowing whether the BBU pool will be able to process users' data. In this case, we notice a significant degradation of wasted transmission power. For the Maximizing Throughput objective and Maximizing the Number of Admitted Users objective, the wasted power increases until it reaches 59.8% and 31.7%, respectively, when the BBU load is 195%. Here we notice a significant benefit of the proposed coordination between radio and CPU scheduling: it saves considerable power.

3.4.5 MCS selection distribution

To better understand the strategy each policy follows to select the users to be scheduled and to assign their MCS indexes, we plot in Fig. 3.4e the cumulative distribution function of the selected MCS indexes when the number of RRHs per BBU pool is equal to 25, along with the curve of the maximum allowed MCS indexes. The latter distribution includes all users from all RRHs, whether they were admitted or not. The previous curves were only concerned with users who were admitted. The results show that the AAU policy, to ensure the admission of all the users in the BBU pool, forces the radio scheduler to enormously decrease the MCS indexes of users. In particular, the median value in the AAU policy is 0, meaning that 50% of users operate with the lowest MCS index, which is 0. However, no user under this policy uses an MCS index higher than

6. Looking at the MTT policy, we notice that it favors users with higher MCS indexes. The median of the corresponding CDF is equal to 10, which means that 50% of the users operate using an MCS index higher than 10. Moreover, the 90^{th} percentile for MTT is around the MCS 15, meaning that 10% of the users have an MCS index higher than 15. This behavior emphasizes that MTT's strategy is to schedule almost all the high-MCS users, leaving those with low MCS indexed with no resources. On the other hand, the CDF of the MUS policy is similar to that of the MAX MCS initially assigned to users. This justifies its fairness since the similarity of the distributions indicates that the MUS policy tries to assign each user an MCS close to the maximum allowed one; it attempts to make users fully satisfied as much as possible. With a probability greater than 0.6, the MUS policy will select an MCS between 4 and 12. Moreover, we plot the curve of the cumulative distribution function of the maximum MCS indexes of dismissed users when the number of RRHs per BBU pool is equal to 25 in Fig. 3.4f. As the curve shows, MTT prefers to dismiss users with low maximum MCS indexes while AAU prefers to dismiss users with high maximum MCS indexes.

In conclusion, we can confirm that while the MTT policy favors the selection of high MCS index users and the AAU policy favors the selection of low MCS indexes, the MUS manages to strike a balance that minimizes the harm both on high and low throughput users.

Moreover, we have shown that the proposed coordination scheme brings improvements, especially in reducing the amount of wasted power by up to 48%. Furthermore, with respect to the other metrics, the slight improvement of (1% - 2%) could be of noticeable importance for operators, given the limited network resources. On the other hand, a practical implementation of the proposed coordination cannot be based on solving an optimization problem that may require heavy computational resources. It is necessary to propose low-complexity heuristics that can achieve a performance close to that of the ILP coordination solutions and allocate resources in real-time. In the following section, we discuss a few proposed heuristics.

3.5 The Proposed Low-Complexity Heuristics

In practice, mobile network operators should be able to dynamically allocate resources in a relatively short duration. While the proposed ILP coordination solutions manage to show enhancements over the non-coordination ones, it is not feasible for operators to solve an Integer Linear programming whenever it needs to allocate resources to users. Solving ILP problems requires a lot of computational resources. It could thus be computationally infeasible to solve them in real-time. Hence we switch our focus to low-complexity algorithms that utilize the coordination principle and can output sub-optimal MCS allocations in a short computation time. For this reason, we propose and evaluate three heuristics that can be used as alternatives to the ILP-based algorithms presented in section 3.2.2.

Algorithm 1: Heuristic 1 - Prioritize High MCS **&** Heuristic 3 - Prioritize Low Throughput

input : $\mathcal{R}, \mathcal{U}_{r\in\mathcal{R}}, \{M_{u,max} : u \in \mathcal{U}_r, r \in \mathcal{R}\}$.
initialize:
1) Put all users $u \in \mathcal{U}_r$ from all RRHS $r \in \mathcal{R}$ in a list $\mathcal{L}$ and sort them:

- Heuristic 1:
  $\mathcal{L}_m = \{u_1, u_2, ..u_i, u_{i+1}... \mid u_i \in \mathcal{U}_r, r \in \mathcal{R}, M_{u_i,max} = m, b_{u_i,max} <= b_{u_{i+1},max}\}$;
  $\mathcal{L} = \mathcal{L}_{|\mathcal{M}|} \cup ..\mathcal{L}_{m+1} \cup \mathcal{L}_m \cup ..\mathcal{L}_1 \cup \mathcal{L}_0$ ***(sorted list)***
- Heuristic 3:
  $\mathcal{L} = \{u_1, u_2, ..u_i, u_{i+1}... \mid u_i \in \mathcal{U}_r, r \in \mathcal{R}, b_{u_i,max} <= b_{u_{i+1},max}\}$ ***(sorted list)***;

2) $AdjMargin \leftarrow 0$;
3) $maxMCS \leftarrow \max(\{M_{u,max} : u \in \mathcal{U}_r, r \in \mathcal{R}\})$;
4) $AvTime(c) = d, \forall c \in \mathcal{C}$;
5) $x^c_{u,m} \leftarrow 0, \forall r \in \mathcal{R}, u \in \mathcal{U}_r, m \in \mathcal{M}, c \in \mathcal{C}$;
while $AdjMargin \leq maxMCS$ **do**
  for $u \in \mathcal{L}$ **do**
    $m \leftarrow (M_{u,max} - AdjMargin)$;
    if $m >= 0$ **then**
      if $\exists c \in \mathcal{C}$ *such that* $t_{u,m} < AvTime(c)$ **then**
        $x^c_{u,m} \leftarrow 1$;
        Remove u from $\mathcal{L}$;
        $AvTime(c) \leftarrow (AvTime(c) - t_{u,m})$;
      end
    end
  end
  $AdjMargin \leftarrow AdjMargin + 1$;
end
output : $x^c_{u,m}, \forall r \in \mathcal{R}, u \in \mathcal{U}_r, m \in \mathcal{M}, c \in \mathcal{C}$.

3.5.1 The proposed heuristics

In this section, we propose three heuristics that consider the adjustment of the MCS indexes of users and aim to perform close to the optimal ILP solutions. We refer to the parameters and variables presented in Table 3.1:

- *Heuristic 1 - Prioritize High MCS*: Apply a two-level sorting to all users from all RRHs, firstly in descending order of maximum allowed MCS-Index and then in ascending order of maximum achievable throughput. An adjustment margin variable $AdjMargin$ is initialized to zero; this variable limits how much a user's MCS can deviate from the Maximum allowed MCS-Index. Then, the algorithm loops over the sorted users trying to admit them. After each complete loop, the algorithm increases the variable $AdjMargin$ by 1, then loops again over all sorted users. The algorithm stops when

Algorithm 2: Heuristic 2 Admit All Users

input : $\mathcal{R}, \mathcal{U}_{r\in\mathcal{R}}, \{M_{u,max} : u \in \mathcal{U}_r, r \in \mathcal{R}\}$.
initialize:
1) Put all users $u \in \mathcal{U}_r$ from all RRHS $r \in \mathcal{R}$ in a list $\mathcal{L}$ and sort them:

- $\mathcal{L}_m = \{u_1, u_2, ..u_i, u_{i+1}... \mid u_i \in \mathcal{U}_r, r \in \mathcal{R}, M_{u_i,max} = m, b_{u_i,max} <= b_{u_{i+1},max}\}$;
  $\mathcal{L} = \mathcal{L}_0 \cup \mathcal{L}_1 \cup ..\mathcal{L}_m \cup ..\mathcal{L}_{m+1} \cup ..\mathcal{L}_{|\mathcal{M}|}$ ***(sorted list)***

2) $SysMCS \leftarrow 0$;
3) $maxMCS \leftarrow \max(\{M_{u,max} : u \in \mathcal{U}_r, r \in \mathcal{R}\})$;
4) $AvTime(c) \leftarrow d \; \forall c \in \mathcal{C}$;
5) $x^c_{u,m} \leftarrow 0, \forall r \in \mathcal{R}, u \in \mathcal{U}_r, m \in \mathcal{M}, c \in \mathcal{C}$;
6) $selectedMCS_u \leftarrow -1, \forall r \in \mathcal{R}, u \in \mathcal{U}_r$;
7) $selectedCPU_u \leftarrow -1, \forall r \in \mathcal{R}, u \in \mathcal{U}_r$;
8) $t_{u,-1} \leftarrow 0, \forall r \in \mathcal{R}, u \in \mathcal{U}_r$;
9) $AvTime(-1) \leftarrow 0$;
while $SysMCS \leq maxMCS$ **do**
  for $u \in \mathcal{L}$ **do**
    $m \leftarrow \min(\{M_{u,max}, SysMCS\})$;
    if $m > selectedMCS_u$ **then**
      $AvTime(selectedCPU_u) \leftarrow$
      $(AvTime(selectedCPU_u) + t_{u,selectedMCS_u})$;
      if $\exists c \in \mathcal{C}$ *such that* $t_{u,m} < AvTime(c)$ **then**
        $selectedCPU_u \leftarrow c$;
        $selectedMCS_u \leftarrow m$;
      end
      $AvTime(selectedCPU_u) \leftarrow$
      $(AvTime(selectedCPU_u) - t_{u,selectedMCS_u})$;
    end
  end
  $SysMCS \leftarrow SysMCS + 1$;
end
$x^{selectedCPU_u}_{u,selectedMCS_u} \leftarrow 1, \forall r \in \mathcal{R}, u \in \mathcal{U}_r$;
output : $x^c_{u,m}, \forall r \in \mathcal{R}, u \in \mathcal{U}_r, m \in \mathcal{M}, c \in \mathcal{C}$.

$AdjMargin$ becomes greater than $MaxMCS$ parameter; the latter is defined as the highest MCS index among all users. The detailed algorithm is presented in Algorithm 1.

- *Heuristic 2 - Admit All Users*: Apply a two-level sorting to all users from all RRHs; firstly in ascending order of maximum MCS-Index; then in ascending order of maximum achievable throughput. A variable called $SysMCS$ is initialized to zero. This variable defines a limit on the MCS index that all users can use. Afterward, the algorithm loops

over the sorted users. In each loop, the algorithm attempts to admit the users with the minimum of two indexes; $SysMCS$, and the maximum allowed MCS index of a user, $M_{u,max}$. Once the loop is completed, the $SysMCS$ is increased by one, and the users attempt to use the modified $SysMCS$ depending on the available computing resources. The algorithm terminates when $SysMCS$ exceeds the highest MCS index among all users, $MaxMCS$. The complete algorithm is presented in Algorithm 2.

- *Heuristic 3 - Prioritize Low Throughput*: The algorithm acts the same as in Heuristic 1 except in the sorting order; instead of applying a two-level sorting, all users are sorted in ascending order of maximum achievable throughput. Again, the detailed algorithm is presented in Algorithm 1.

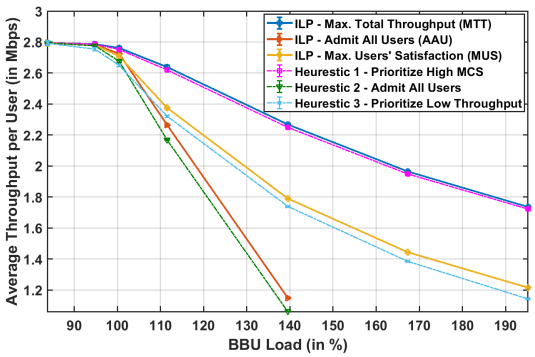
3.5.2 Performance analysis of the heuristics

The proposed heuristics are evaluated and compared to the ILP-based policies of section 3.2.2 with respect to the same metrics of section 3.3.2: Average Throughput per User, Number of Admitted Users, and Fairness Index. Recall that for the wasted power metric the ILP-based policies and the proposed heuristics enable the coordination between radio and computing schedulers, and hence they have the same performance with respect to this metric. The results are depicted in figures 3.5a, 3.5b, and 3.5c, respectively. We note from the figures that:

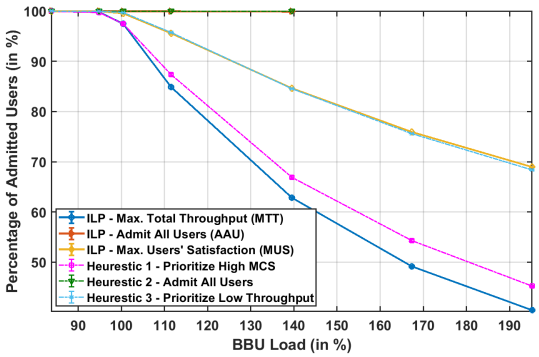
- *Heuristic 1 - Prioritize High MCS* shows very close results to those obtained by the first ILP problem, MTT, especially concerning the throughput metric because it prioritizes users with high MCS indexes. However, it outperforms MTT with respect to the other two metrics: the number of admitted users and the fairness metrics. Compared to MTT, this heuristic can score up to a 4.5% improvement concerning the percentage of admitted users metric and up to 0.049 of improvement for the fairness metric.

- *Heuristic 2 - Admits All Users* aims to admit all users in the system; hence its performance regarding the percentage of admitted users is the same as AAU policy. It deviates slightly from AAU policy with respect to the throughput metric and can worsen the performance with a maximum drop of 4%. With respect to the fairness index metric, this heuristic and AAU can deviate from each other with a difference not larger than 0.06. As mentioned earlier, AAU is no longer feasible once the BBU pool exceeds 140%.

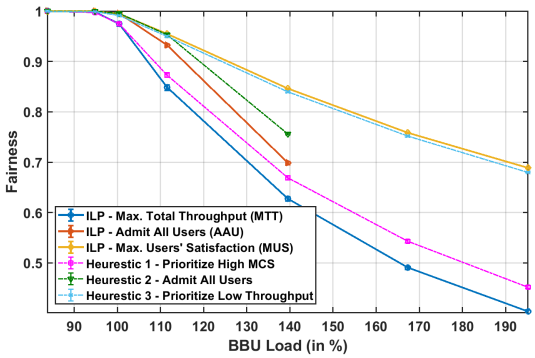
- *Heuristic 3 - Prioritize Low Throughput* has more or less a similar performance compared to MUS policy concerning all metrics. Concerning the throughput metric, the performance of this heuristic would drop by up to 6%. The difference between the ILP and the heuristic is very slight for the metrics of admitted users and fairness.



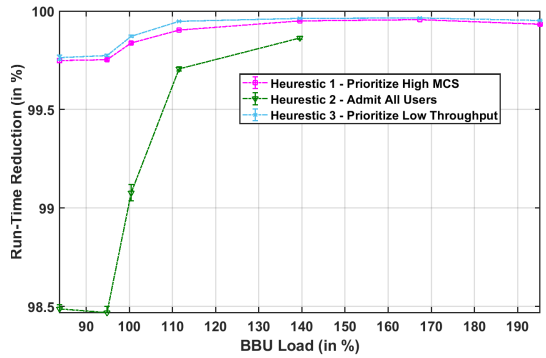
(a) Offered throughput (in %) as a function of BBU pool load



(b) Admitted Users (in %) as a function of BBU pool load



(c) Jain's Fairness Index as a function of BBU pool load



(d) Reduction of Elapsed Time of each heuristic as a function of the number of RRHs with respect to its ILP algorithm

Figure 3.5: Comparison of the performance of the heuristics in comparison to ILP problems

Compared with its corresponding ILP counterpart, each of the three heuristics scored close results concerning all performance metrics. We recall that the three heuristics score 0% concerning the metric of wasted power. Like the ILP coordination solutions, users aware that the BBU can not process their data will not transmit, thus saving the transmission power. In short, the proposed heuristics can serve as practical replacements for the high-complexity ILP algorithms and can be implemented by the mobile operators for real-time scheduling.

3.5.3 Computational complexity

The main motive behind proposing the heuristics is the high computational complexity of ILP solvers. Finding real-time scheduling results is essential for mobile operators; otherwise, our proposal will not have practical grounds and will remain theoretical.

For both Algorithm 1 and Algorithm 2, the sorting part should, in the worst case,

take no more than $|\mathcal{R}|^2 \cdot |\max_{r\in\mathcal{R}} \mathcal{U}_r|^2$, supposing that in each iteration, each user will be compared with all other users. For the second part of each algorithm, the worst-case scenario corresponds to iterating over all MCS indexes, and for each MCS index, iterating over all users. Hence, the number of iterations for the second part of each algorithm is $|\mathcal{R}| \cdot |\max_{r\in\mathcal{R}} \mathcal{U}_r| \cdot |\mathcal{M}|$.

Experimentally, the proposed heuristics can find solutions much faster than the corresponding ILP algorithms. In figure 3.5d, we plot the graphs of the percentage of reduction of Elapsed Time of each heuristic as a function of the number of RRHs with respect to its corresponding ILP algorithm. We have made the study using a computer running on Intel Core i9-9880H Processor, and the ILP solver is CPLEX for MATLAB. While Heuristic 2 has achieved more than 98.6% reduction in run-time with respect to AAU, Heuristics 1 and 3 achieved more than 99.7% reduction with respect to MTT and MUS, respectively. The achieved reduction in run-time is very significant.

3.6 Multi-Services Scenario

The fifth-generation (5G) New Radio (NR) of mobile communications has been designed to support two major classes of services with vastly heterogeneous requirements: ultra-reliable low-latency communication (URLLC) and enhanced mobile broadband (eMBB) [149]. On the one hand, eMBB supports stable connections with very high peak data rates and moderate rates for cell-edge users. On the other hand, URLLC supports low-latency transmissions of small payloads with very high reliability from a limited set of terminals.

Few approaches have been adopted in the Third Generation Partnership Project (3GPP) standard [154] to handle the coexistence of these two services. One possible way is to slice the radio resources and reserve a portion for URLLC traffic [149]. Another approach is multiplexing eMBB and URLLC on shared radio resources while prioritizing the latter [149]. The latter can puncture ongoing eMBB transmissions and transmit instead of them. URLLC transmission can happen at the start of an sTTI (Short Transmission Time Interval). Hence their transmission time is much shorter than eMBB transmission time.

In the previous sections, we have demonstrated the benefits of the proposed coordination and proposed low-complexity heuristics, alternatives to the ILP-based coordination algorithms. The scenarios we tested in the previous sections go under the eMBB category, and we have not considered the effect of URLLC service transmission subject to tight latency constraints. While the coordination policies achieve better results than no-coordination for eMBB traffic, it is interesting to study the impact of URLLC traffic on the performance of our proposed coordination policies. The impact of URLLC frames exists when the computing power is shared by both resources, as it is the case in our study. In particular, the incoming URLLC traffic during the processing period of eMBB transmissions cannot be delayed due to the strict latency requirement. The coordination

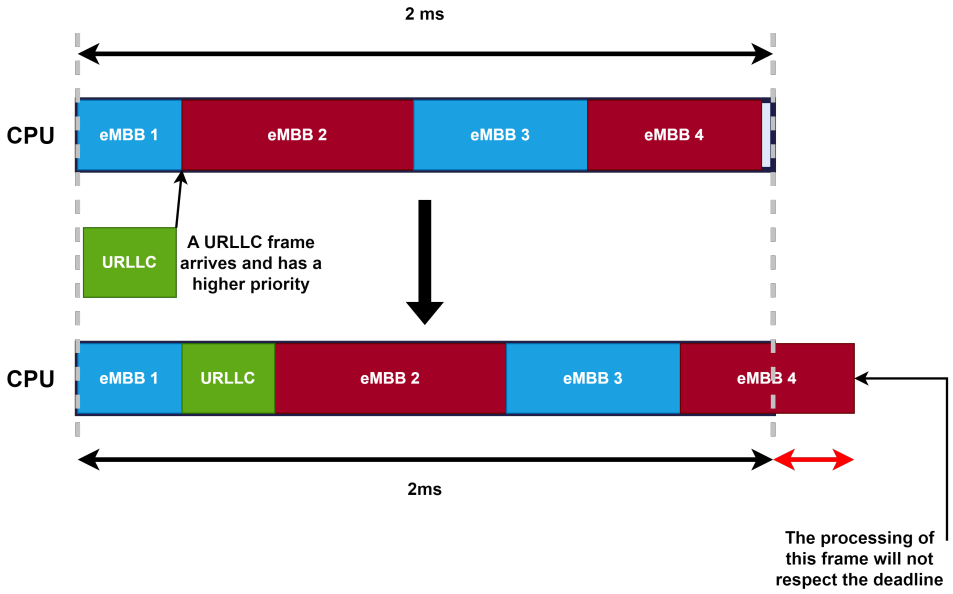


Figure 3.6: A URLLC frame is prioritized over eMBB frames. As a result, the processing of some eMBB frames exceeds the deadline

does not modify the MCS index of URLLC transmission but only controls the MCS index of eMBB frames. However, the URLLC frames are prioritized over eMBB frames. In other words, once URLLC frames arrive, the BBU pool should process them and preempt eMBB frames, as Figure 3.6 shows.

Given that URLLC frames arrive at the BBU pool every sTTI and are prioritized over eMBB transmission, the performance of the heuristics may change depending on the arrival rate of URLLC frames. Hence it remains important to analyze how the heuristics perform for different URLLC frames arrival rates. It is worth mentioning that on the radio level, it is possible that either there are radio resources reserved for URLLC transmissions or that URLLC transmission punctures ongoing eMBB transmissions and transmits instead. However, in this chapter, regardless of how URLLC traffic was transmitted at the radio level, we only focus on the effect of the competition for computing resources on the coordination performance, and we leave the effect of puncturing on the radio level for future work.

3.6.1 Scenario description

Targeting reliable and low latency communications, the 5G-REL15 [154] [155] introduced mini-slots; the time required to transmit a transport block over this mini-slot is equal to the sTTI length. Hence, URLLC traffic arrives at the BBU pool every sTTI duration. Enabling the mini-slots option paves the way for URLLC users to achieve their low latency requirement, although this may come at the expense of eMBB users. As URLLC users have higher priority, they need to be processed, even if this leaves eMBB frames unprocessed before the 2ms deadline. Figure 3.7 shows the scenario under study.

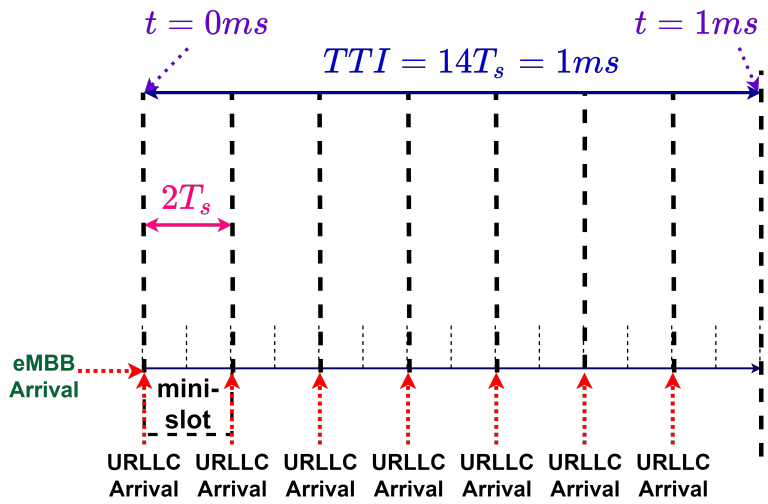


Figure 3.7: eMBB and URLLC frames arriving in the same TTI share a set of computing resources. T_s is the duration of one OFDM symbol.

We suppose that eMBB frames and URLLC frames that arrive during the same TTI will be processed by the same set of CPUs so that the processing of eMBB frames from different TTIs will not happen on the same CPUs to avoid overlapping. However, we limit the study to only one TTI, so only one set of CPUs is needed. One TTI consists of 14 OFDM symbols. In addition, we consider the underlying short TTIs (sTTIs) to be 2-OFDM symbols long. At each sTTI, URLLC frames arrive, and the CPUs should process them while preempting the ongoing processing of eMBB frames.

We consider the same scenario described in Section 3.3, but with the existence of URLLC or eMBB users. We suppose that the arrival of URLLC packets in each mini-slot follows a Poisson process with an arrival rate λ (in our simulation, λ varies between 0 and 5 users per RRH per sTTI). We note that for $\lambda = 0$, we get the scenario with eMBB users only, while $\lambda = 5$ represents a very extreme case of an average URLLC frames arrival of 35000 frames per second. To estimate the processing load of URLLC packets on the BBU pool, we additionally need to determine their processing time. Referring to equation (3.1), this processing time depends on the number of RBs used by these frames. The MCS used by URLLC users is sampled as in section 3.3.2. For the required number of RBs to be used by URLLC users, we use the following formula [66]:

$$N_u^{LC} = \frac{PacketLength}{Q_m \times N_{MiniSlot}^{RE} \times N_{RB}^{SC} \times CodeRate \times OH} \tag{3.10}$$

Q_m is the number of bits per symbol. Q_m and the $CodeRate$ are determined from the tables in [156]. $N_{MiniSlot}^{RE}$ is the number of Resource Elements (OFDM symbols) per a mini-slot per 1 sub-carrier (i.e., 2, 4, or 7), N_{RB}^{SC} is the number of sub-carriers per a Resource Block (i.e., 12), and OH is the overhead. Similar to [132], we suppose that the length of URLLC packets, $PacketLength$ is 32 bytes. As in [66], OH is equal to 0.715. For the sake of benchmarking, we consider two additional No-Coordination

heuristics from [13]: Highest Throughput First (HTF), which prioritizes users with the highest throughput, and Shortest Time First (STF), which prioritizes users with the shortest processing time. In these algorithms, users transmit with their max MCS, and the BBU pool can only decide to process users or dismiss them. The simulation is run 100 times, and the 95% confidence intervals are shown. We note that the confidence interval may appear very tight in the figures. All the other parameters in the simulation environment described in section 3.3 are maintained.

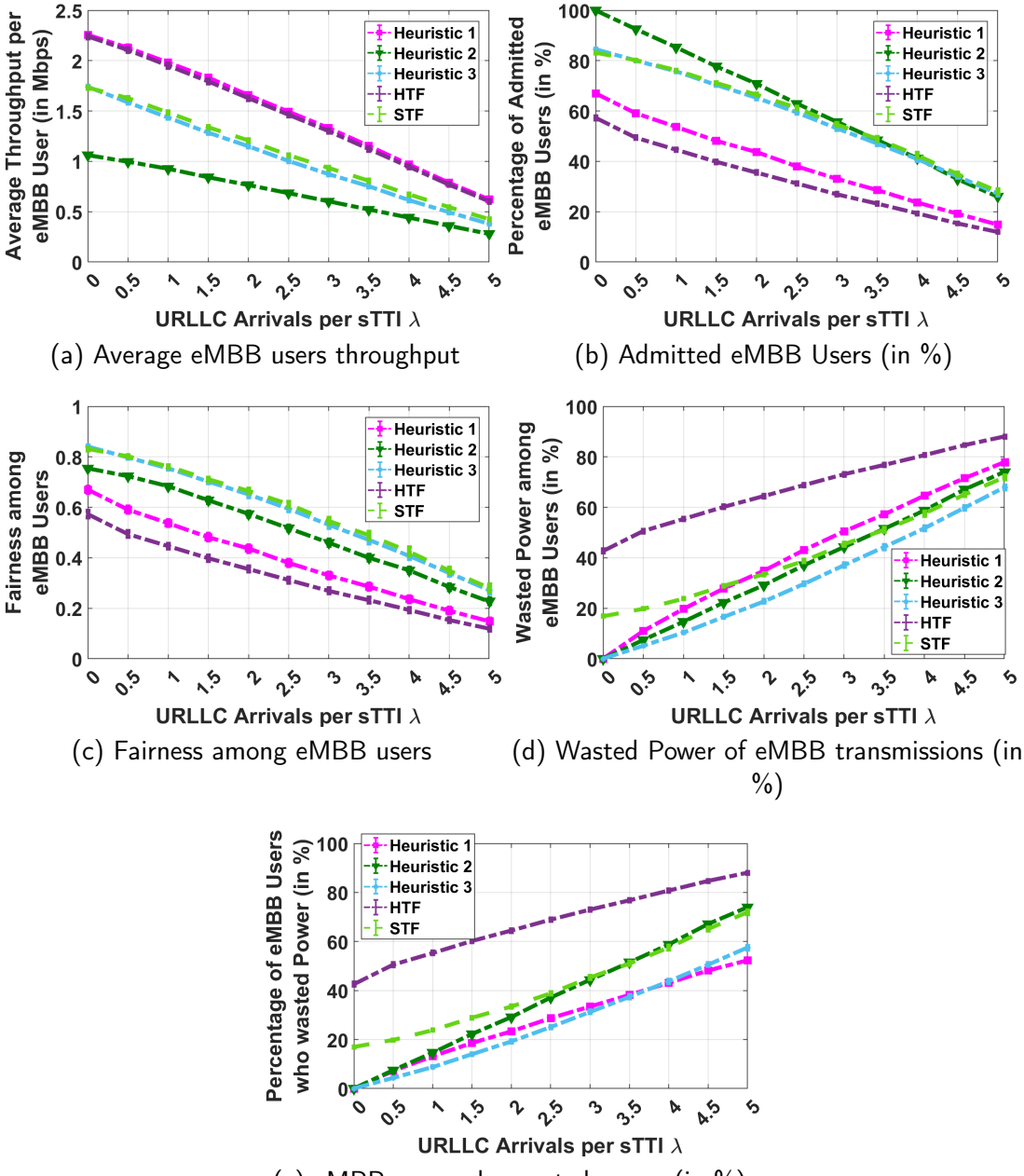
3.6.2 Performance evaluation in a multi-services environment

In the following, we evaluate the performance of our proposed heuristics in Section 3.5 with the coexistence of eMBB and URLLC users, using the metrics of throughput, Admitted Users, Fairness Index, and wasted power. We note that these metrics depict the performance of eMBB users only; because URLLC users cannot adjust their MCS indexes, the coordination is irrelevant to them.

Fig. 3.8a, 3.8b, 3.8c, 3.8d, and 3.8e show the performance of the coordination heuristics as a function of URLLC users' arrival rate, concerning the metrics of eMBB average throughput per user, the total number of admitted eMBB users, Jain's Fairness Index, and the wasted transmission power. Fixing the number of RRHs to 25, we monitor the effect of URLLC users' arrival on the performance of eMBB users.

Concerning all metrics, the performance of the heuristics degrades as the URLLC arrival rate increases. As before, Heuristic 1 achieves the highest average throughput among eMBB users in comparison to other heuristics. Additionally, while Heuristic 3 achieves higher throughput than Heuristic 2 for low URLLC arrival rate, as explained in previous sections, the performance of the two heuristics converges at the end. The reason is that these two heuristics process users based on the admission order. Heuristic 2 sorts users in the ascending order of the MCS index, then in the ascending order of throughput, while Heuristic 3 sorts users in the ascending order of throughput. This increases the tendency to process users with low frame sizes first and keep users with high frame sizes until the end. On the other hand, Heuristic 2 tends to admit potentially high throughput users with a more reduced MCS index to accommodate more low throughput users, similar to the behavior of ILP-AAU in Fig. 3.3b. This justifies why Heuristic 3 achieves higher throughput at low arrival rates. When URLLC frames arrive, they cause the CPU to preempt eMBB users' processing and start processing the URLLC frames that arrived. This would make it impossible for the eMBB users with longer frames, and thus higher throughput, to be processed before the deadline. At a very high URLLC arrival rate, most users with high and medium frame sizes fail to be processed before the 2ms deadline; the BBU pool would only process eMBB users with low MCS and throughput.

As figure 3.8b shows, Heuristic 2 can no longer admit all eMBB users in the presence of URLLC frames. Since the BBU pool is already fully loaded for a number of RRHs equal to 25, the computing resources are insufficient to process all eMBB and URLLC frames in 2 ms. Hence the arrival of URLLC frames leads to failure to process eMBB users. This violates the heuristic's primary goal, which is to admit all eMBB users. For



(a) Average eMBB users throughput

(b) Admitted eMBB Users (in %)

(c) Fairness among eMBB users

(d) Wasted Power of eMBB transmissions (in %)

(e) eMBB users who wasted power (in %)

Figure 3.8: Performance evaluation of our proposed heuristics as a function of URLLC users' arrival rate when the number of RRHs in the BBU pool is 25

the same reasons explained above, figure 3.8c shows the performances of heuristics 2 and 3 with respect to the fairness metric converge at higher URLLC arrival rates.

Analyzing Fig. 3.8d, the heuristics employing coordination can no longer achieve 0 wasted transmission power. Users are initially promised to be admitted by the BBU

pool. However, an increased arrival rate of URLLC frames leads to increased wasted transmission power. It is good to note that at a high URLLC arrival rate, selecting Heuristic 2 becomes a bad idea compared to Heuristic 3. The latter becomes a better choice considering all the metrics.

When comparing the coordination heuristics to the no coordination heuristics, HTF and STF, the existence of URLLC traffic reduces and diminishes the already slight improvement (1%-2%, as section 3.4 shows) that the coordination brings concerning the metrics of throughput, admitted users, and fairness. The reason is that URLLC traffic negatively affects the users who managed to be admitted at the BBU pool by reducing their MCS index. Concerning the metric of wasted power, the coordination heuristics 1 and 3 remain better than the no-coordination counterparts, HTF and STF, respectively, because regardless of URLLC traffic, eMBB users, whom the BBU pool is initially unable to process, will not transmit and thus save transmission power. However, since the metric of wasted power is normalized with respect to the total transmission power and the coordination decreases the total transmission power, it would delude us to think that STF is better than Heuristic 2. Hence, we plot the graphs of the percentage of eMBB users who wasted their transmission power with respect to the total number of users in the BBU pool in fig. 3.8e. It is clear that the percentage of users who wasted their power for heuristics 1 and 3 is much better than the no-coordination heuristics. However, since Heuristic 2 admits all users and then removes a lot of them to prioritize URLLC, it wastes more power than STF at $\lambda = 5$.

In short, even in the extreme case considered in this section, where random URLLC frames arrive with no predetermined processing resources in the BBU pool, we show that coordination heuristics between the radio MCS assignment and the BBU resources is effective in saving radio power and reducing the percentage of unprocessed eMBB frames. Among the heuristics presented, Heuristic 3 seems to be more robust and offers the best performance in this context.

3.7 Summary

In this chapter, we investigated three ILP policies that implement coordination between radio and computing schedulers in Cloud-RAN context. Motivated by the fact that the data processing time strongly depends on the transmission MCS index, the coordination policies allow the radio scheduler to set the MCS index for users' transmission not only based on the radio conditions but also on the ability of the BBU pool to process users' data. The three coordination schemes (namely MTT, AAU and MUS) aim to maximize total throughput, admit all users, and maximize users' satisfaction. We have evaluated them according to different performance metrics. Results show that the proposed coordination achieves a vital improvement by significantly reducing the amount of wasted transmission power and bringing a slight but systematic improvement to the other metrics. Among the ILP coordination policies, the MUS policy is the fairest; it achieves in-between values of throughput and the number of allocated users in comparison with

the other coordination policies. In addition, we proposed three low-complexity heuristics and compared their performance to that of the high-complexity ILP algorithms. We showed that the heuristics are good candidates to replace the ILP algorithms to achieve real-time performance. Moreover, we analyzed the performance of the heuristics in a multi-service environment, where users of different services (i.e., eMBB and URLLC) coexist. The results show that the heuristics employing coordination can no more avoid wasting transmission power when URLLC traffic exists. However, they still reduce power consumption in comparison with no-coordination heuristics.

Chapter 4

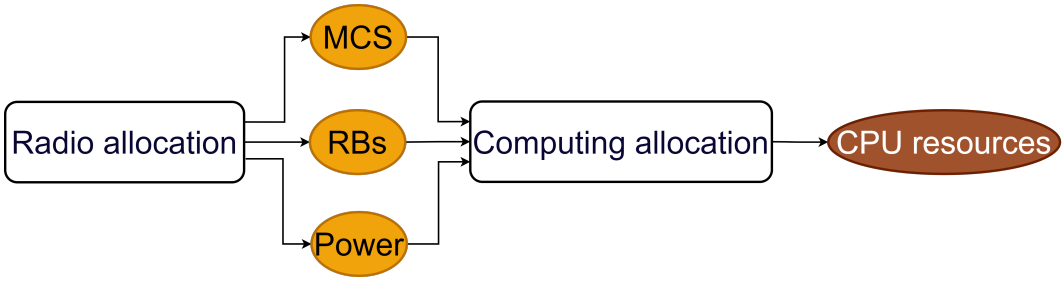
Energy Consumption Minimization through Joint Radio and Computing Resource Allocation

In the previous chapter, we investigated the benefits of the coordination policies between radio and computing schedulers. The coordination schemes enable limited cooperation between radio and computing schedulers such that only the MCS selection can be modified when computing resources are insufficient. In contrast, the centralized architecture of Cloud-RAN permits joint centralized resource allocation, and this helps better optimize the network performance. In this chapter, we consider a joint allocation of radio and computing resources in Cloud-RAN. We aim to analyze the benefits of applying this joint allocation with the goal of minimizing the total energy consumption. We model a Mixed Integer Linear Programming (MILP) problem to jointly allocate Resource Blocks (RBs), the Modulation and Coding Scheme (MCS), radio transmission power, and CPU resources to process user frames while aiming at minimizing energy consumption. We compare it to another objective of throughput maximization, and we compare the joint allocation with a sequential radio allocation followed by computing resource allocation. Given that solving an MILP problem is NP-hard, we devise a matching game-based algorithm that aims at performing as close as possible to the optimal problem.[1]

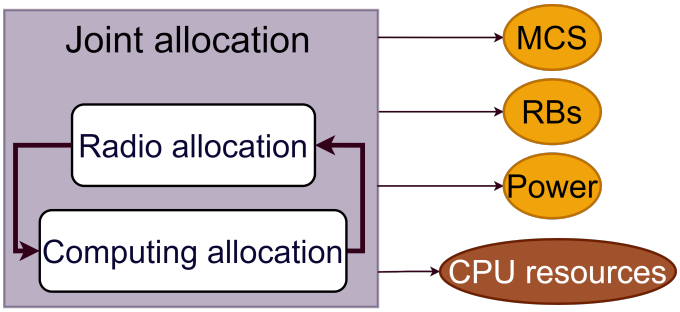
4.1 Introduction

Joint allocation of radio and computing resources in Cloud-RAN has the potential to minimize the energy consumed for radio transmission and baseband processing. Radio resource allocation consists in assigning to each user some Resource Blocks (RBs) and a Modulation and Coding Scheme (MCS) index, along with the transmission power. On the other hand, computing resource allocation consists of assigning users' radio frames

[1] The contents of this chapter are presented in [4], [8].



(a) Sequential allocation



(b) Joint allocation

Figure 4.1: Sequential allocation vs Joint allocation of radio and computing resources

to CPUs in the BBU pool to meet the processing deadline.

The throughput of users depends on their MCSs and on the number of RBs over which they transmit [6], [156]. The transmission power assigned to users affects the Signal-to-Noise-Ratio (SINR) of each user, which in turn controls the maximum MCS that the user is allowed to adopt. Moreover, the execution time of the baseband processing of a user's radio frame increases with the number of RBs and especially with the MCS [81]. Hence, these two radio parameters affect the required computing resources, which in turn affect the computing energy consumption.

In this chapter, we consider the joint allocation of radio and computing resources in Cloud-RAN by formulating a Mixed Integer Linear Programming (MILP) problem that jointly allocates the transmission power, the resource blocks, the MCS indexes, and the CPU time to process the data of each user. We consider the goal of minimizing energy consumption and compare the results to those obtained when the goal is to maximize system's throughput. To quantify the impact of joint radio and computing resource, we compare it to a sequential scheme that performs radio resource allocation (via MILP) followed by computing resource allocation.

Figure 4.1 shows the difference between sequential and joint allocation of radio and computing resources. In a setting where radio resources (transmission power, RB number, MCS) and computing resources (CPU time) are allocated sequentially, as in Figure 4.1a, minimizing a user's transmission power while still meeting its throughput target may result in MCS and RBs assignments that require more computing resources,

consuming more energy. A joint radio and computing resource allocation scheme that controls the radio and computing resource parameters at the same time would be better able to minimize total energy consumption (i.e., the sum of transmission *and* processing energy). Even though joint allocation may be more computationally complex than sequential allocation, it would be advantageous if it exhibits significant energy consumption reductions.

As discussed in Chapter 2, the matching theory has gained a lot of ground in wireless communication. Considering that solving an MILP problem is highly complex and is NP-Hard [18], we design a low-complexity two-step algorithm that allocates RBs, transmission power, and MCS indexes to users using a matching game and a transmission power adjustment mechanism. We analyze the convergence and the complexity of the matching-based algorithm and compare its results to the optimal performance of the MILP problem.

As we surveyed in Chapter 2, various works have considered radio or computing resource allocation. The authors of [144] aimed to optimize system throughput and energy efficiency and devised a two-step algorithm. The computing resource allocation occurs first by mapping users to Virtual Machines (VM). Secondly, the radio resource allocation is done by controlling the beamforming vectors. The algorithm does not consider the existence of multiple RBs or sub-carriers nor the selection of MCS indexes based on the SINR. While the authors of [102] consider joint beamforming vector design and BBU computational resource allocation and aimed to minimize the total system power consumption, they also do not consider the RBs or MCS assignments nor the effect on the required processing time. Moreover, [101] uses an auction-theory-based algorithm to jointly allocate radio and computing resources. The objective is to minimize the mean response time at the queues of the BBUs and RRH; however, this paper does not tackle the joint allocation problem for minimizing the total energy consumption. In another context, [78] formulated a radio allocation MILP problem that considers RBs and MCS assignment and power allocation. However, the model is restricted to one base station, and it does not take the interference caused by other base stations into account. Additionally, the problem only considers radio allocation without considering computing resource allocation, and it does not aim at minimizing the total energy consumption. In [107], where a NOMA-based MEC model that aims at improving energy efficiency is considered, a joint optimization problem is formulated where the problems of user association, power control, and computational resource allocation are combined together to optimize energy consumption. Due to the NP-Hardness of the model, the authors propose to use a matching and coalition framework. This paper does not consider OFDM nor allocate MCS indexes to users . Using MCS indexes is more accurate in detecting the achieved throughput than using Shannon's capacity formula, which only provides a limit on the achievable capacity. Also, it is more realistic given that the MCS index is a necessary parameter when communicating over a cellular network. In [69], the problem of resource allocation of sub-channels in Heterogeneous Cloud RAN is addressed. They use a matching-coalition based-algorithm. Their algorithm restricts RB allocation to

one per user. All these papers do not consider the joint and simultaneous allocation of RBs, MCS, power, and computing resources for minimizing energy consumption. In the current literature and to the best of our knowledge, there is no explicit comparison of joint vs. sequential resource allocation. Furthermore, the advantages, limitations, and influence of the chosen objective function have not yet been investigated.

The main contributions of this paper are recapped:

- We model the joint radio and computing resource allocation problem as an MILP problem that allocates RBs, radio power, MCS indexes, and CPU resources to users.
- We consider two objectives; energy consumption minimization and throughput maximization.
- We compare the performance of the joint scheme of radio and computing resource allocation to that of a sequential scheme in which radio and computing resource allocation are done sequentially.
- We propose a two-step algorithm based on a matching game and a radio power adjustment mechanism to achieve solutions close to the optimal solution of the joint allocation MILP problem.
- We provide the convergence and complexity analysis of the matching-based algorithm.
- We compare the performance of the proposed matching-based algorithm to the optimal MILP problem.

4.2 Problem Formulation

In this section, we first present the problem of joint allocation of radio and computing resources allocation in Cloud-RAN, and we model it as a Mixed Integer Linear Programming (MILP) problem.

4.2.1 Model input and parameters

The system model is shown in Fig. 4.2. To study the performance of joint radio and computing resource allocation, we consider the following: a set of Base Stations $\mathcal{R}$, a set of users $\mathcal{U}_r$ of each BS r, a set of Resource Blocks $\mathcal{B}$ for each base station, a set of MCS indexes $\mathcal{I}$ that can be used in the system, and a set of CPU cores $\mathcal{C}$ in the shared BBU pool (multi-core data center). We focus on the uplink direction in which the BBU pool should execute the complex and energy-consuming decoding function [13]. We assume that each user has a maximum transmission power equal to P_{Tx}^{max} and that the CPU power consumption is equal to P_{comp}^{c}. We define the following parameters: $g_b^{r,u,r'}$ is the channel gain between user $u \in \mathcal{U}_r$ of base station r, and the base station

Table 4.1: Summary of the general notations

Parameters	***Definition***
$ACCEPT^b(u)$	A Boolean to indicate RB b accepts user u
$\mathcal{B}$	Set of RBs that can be used in the system
$\mathcal{B}^{r,u}$	Set of RBs assigned to user $u \in \mathcal{U}_r$
$\beta_i^{r,u}$	Binary variable that indicates that user $u \in \mathcal{U}_r$ uses MCS index i
$\mathcal{C}$	Set of CPU cores in the shared BBU pool (multi-core data center)
d	Processing time deadline
$g_b^{r',u',r}$	The channel gain between user $u' \in \mathcal{U}_{r'}$ and the base station $r \in R$ on RB $b \in \mathcal{B}$
γ_i^{th}	The minimum SINR threshold to use MCS index i
$\mathcal{I}$	Set of MCS indexes that can be used in the system
$MCS_{r,u}$	The MCS index assigned to user $u \in \mathcal{U}_r$
N	The total number of users in the system
$p_b^{r,u}$	A continuous variable that indicates the radio transmission power of user $u \in \mathcal{U}_r$ over RB b.
$P_{T_x}^{max}$	Maximum total radio transmission Power of user $u \in \mathcal{U}_r$
P_{comp}^c	Computing Power of a CPU $c \in \mathcal{C}$
PL_b	The preference list of an RB b
PL_u	The preference list of a user u
PL_u^1	The first element in the preference list of a user u
$prop(u)$	The RB the that a user u proposes to be matched with
$prop(b)$	The users that have proposed to the RB b in the current iteration
$\mathcal{R}$	Set of base stations
$R_{s,i}$	Data length (in bits) when a frame is transmitted over s number of resource blocks using MCS index i
$R_{min}^{r,u}$	The minimum required rate for a user $u \in \mathcal{U}_r$
$REJECT^b(u)$	A Boolean to indicate RB b rejects user u
$SINR^{r,u}(b)$	The possibly achieved SINR of user $u \in \mathcal{U}_r$ if it uses RB b in the next iteration
$\sigma^{r,u}$	The channel noise for user $u \in \mathcal{U}_r$
$th_{r,u}$	The achieved throughput by user $u \in \mathcal{U}_r$
$th_{r,u}(b)$	The possible achieved throughput by user $u \in \mathcal{U}_r$ if it uses RB b
$t_{s,i,c}$	Data processing time when a frame is transmitted over s number of resource blocks using MCS index i, and processed over CPU c
t_{TTI}	The duration of one Transmission Time Interval (TTI)
$\mathcal{U}_r$	Set of users for each BS $r \in \mathcal{R}$
$\overline{\mathcal{U}}_r$	Set of users in BS $r \in R$ who have achieved the demanded throughput
$x_{b,i}^{r,u}$	Binary variable that assigns the RB b and MCS index i to user $u \in \mathcal{U}_r$
$y_{s,i,c}^{r,u}$	A binary variable that indicates a user $u \in \mathcal{U}_r$ uses s total number of RBs and MCS index i and that its frame is processed on CPU c

$r' \in \mathcal{R}$ on RB $b \in \mathcal{B}$. This is required to calculate the interference made by user u on base station r'. If r and r' are the same, then $g_b^{r,u,r'}$ provides the channel gain between the user and its own base station. γ_i^{th} is the minimal SINR threshold that allows to use a given MCS index $i \in \mathcal{I}$. If the SINR is lower than the threshold, then using MCS i will increase the decoding error above the limit of 10% [151]. We denote by $\sigma^{r,u}$ the channel noise for user $u \in \mathcal{U}_r$, $R_{s,i}$ is the throughput of a transmission when the data are

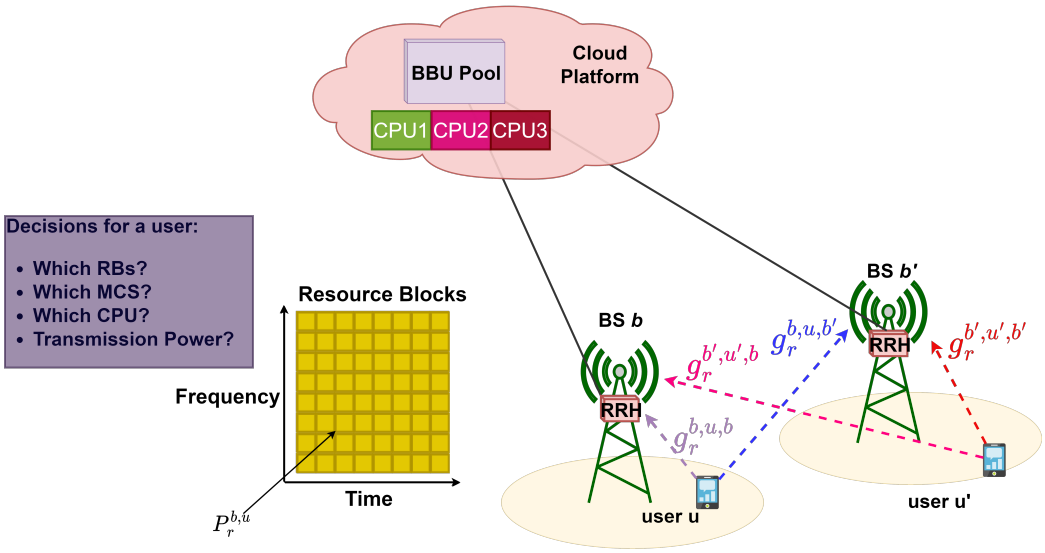


Figure 4.2: The system model where uplink transmission is considered. Resource Blocks, radio power, MCS indexes, and CPUs are the decision variables.

transmitted over s number of RBs using an MCS index $i \in \mathcal{I}$, and $t_{s,i,c}$ is the required time to process these data on CPU core $c \in \mathcal{C}$. Each CPU should process the assigned data before the deadline d. This deadline is imposed by the Hybrid Automatic Repeat Request (HARQ) mechanism and is equal to 2ms [13]. Not respecting this deadline will lead to retransmitting data, thus, wasting the initial transmission. We also suppose that users have different QoS requirements; each user requests a minimum throughput $R^{r,u}_{min}$ that must be satisfied. We note that the duration t_{TTI} of the Transmission Time Interval (TTI) over which a user transmits its frame is 1 ms.

We use the model from [81] modeled using the Open Air Interface (OAI) RAN simulator to determine how much time is needed to process each user's data. This model provides the required processing time, $t_{s,i,c}$, of a user's data as a function of the total number of used RBs, the MCS index, and the CPU frequency. The formula was presented in section 3.3. On the other hand, we use the 3GPP standard [156] to determine the Transport Block Size (TBS), which is the number of bits transmitted by a transport block in 1 ms, as a function of the number of RBs and the MCS index. Then we get the throughput by dividing the TBS by the transmission duration. We note that using the MCS index to calculate the throughput is more realistic than using Shannon's capacity formula, as the latter just gives the upper-bound of the channel's throughput and does not distinguish useful bits from redundancy and physical layer overhead bits. The summary of general notations used throughout this chapter are resumed in Table 4.1.

4.2.2 MILP problem

We formulate a Mixed Integer Linear Programming Model (MILP) for joint radio and computing resource allocation, which minimizes the total energy consumption. This

MILP problem should be optimized by assigning RBs and MCS indexes to users, the power of their signals, and the CPUs that will process their data. The MILP problem contains the following decision variables:

- $x_{b,i}^{r,u}$ is a binary decision variable equal to 1 if user $u \in \mathcal{U}_r$ uses an MCS index $i \in I$ on RB $b \in \mathcal{B}$; otherwise, it is zero.
- $y_{s,i,c}^{r,u}$ is a binary decision variable that is equal to 1 if and only if a user $u \in \mathcal{U}_r$ transmits data using an MCS index $i \in I$ over a total of s resource blocks, and the user's data are processed on CPU $c \in \mathcal{C}$; and zero otherwise.
- The binary decision variable $\beta_i^{r,u}$ is equal to 1 if and only if a user $u \in \mathcal{U}_r$ uses MCS $i \in I$ on any of its RBs; and zero otherwise.
- Finally, $p_b^{r,u}$ is a continuous variable that indicates the transmission power of user $u \in \mathcal{U}_r$ on RB $b \in \mathcal{B}$.

We note that M is the big-M notation [157] and is used to enforce the conditions explained below. The formulated MILP optimization problem that minimizes the total energy consumption for the data transmitted in 1 TTI is:

$$\text{Minimize} \quad \sum_{b\in\mathcal{B}}\sum_{u\in\mathcal{U}_r}\sum_{r\in\mathcal{R}} p_b^{r,u}\cdot t_{TTI} + \sum_{b\in\mathcal{B}}\sum_{u\in\mathcal{U}_r}\sum_{s\in\mathbb{N}\cap[1,|\mathcal{B}|]}\sum_{i\in\mathcal{I}}\sum_{c\in\mathcal{C}} t_{s,i,c}\cdot y_{s,i,c}^{r,u}\cdot P_{comp}^{c} \tag{4.1}$$

$$\text{such that} \quad x_{b,i}^{r,u} \in \{0,1\},\ \forall b\in\mathcal{B}, u\in\mathcal{U}_r, r\in\mathcal{R}, i\in\mathcal{I} \tag{4.2}$$

$$y_{s,i,c}^{r,u} \in \{0,1\},\ \forall r\in\mathcal{R}, u\in\mathcal{U}_r, s\in\mathbb{N}\cap[1,|\mathcal{B}|], i\in\mathcal{I}, c\in\mathcal{C} \tag{4.3}$$

$$\beta_i^{r,u} \in \{0,1\},\ \forall r\in\mathcal{R}, u\in\mathcal{U}_r, i\in\mathcal{I} \tag{4.4}$$

$$p_b^{r,u} \geq 0,\ \forall b\in\mathcal{B}, u\in\mathcal{U}_r, r\in\mathcal{R} \tag{4.5}$$

$$\sum_{u\in\mathcal{U}_r}\sum_{i\in\mathcal{I}} x_{b,i}^{r,u} \leq 1,\ \forall b\in\mathcal{B}, r\in\mathcal{R} \tag{4.6}$$

$$\sum_{s\in\mathbb{N}\cap[1,|\mathcal{B}|]}\sum_{i\in\mathcal{I}}\sum_{c\in\mathcal{C}} y_{s,i,c}^{r,u} R_{s,i} \geq R_{min}^{r,u}, \forall r\in\mathcal{R}, u\in\mathcal{U}_r \tag{4.7}$$

$$\sum_{b\in\mathcal{B}} p_b^{r,u} \leq P_{Tx}^{max}, \forall r\in\mathcal{R}, u\in\mathcal{U}_r \tag{4.8}$$

$$p_b^{r,u} \leq M\sum_{i\in\mathcal{I}} x_{b,i}^{r,u}, \forall b\in\mathcal{B}, u\in\mathcal{U}_r, r\in\mathcal{R} \tag{4.9}$$

$$x_{b,i}^{r,u} \leq \beta_i^{r,u},\ \forall b\in\mathcal{B}, u\in\mathcal{U}_r, r\in\mathcal{R}, i\in\mathcal{I} \tag{4.10}$$

$$\sum_{i\in\mathcal{I}} \beta_i^{r,u} \leq 1, \forall r\in\mathcal{R}, u\in\mathcal{U}_r \tag{4.11}$$

$$g_b^{r,u,r} p_b^{r,u} \geq \gamma_i^{th}(\sigma^{r,u} + \sum_{r'\in\mathcal{R}-\{r\}}\sum_{u'\in\mathcal{U}_{r'}} g_b^{r',u',r} p_b^{r',u'}) - M(1-x_{b,i}^{r,u}), \forall b\in\mathcal{B}, u\in\mathcal{U}_r, r\in\mathcal{R}, i\in\mathcal{I} \tag{4.12}$$

$$\sum_{b\in\mathcal{B}}\sum_{i\in\mathcal{I}} x_{b,i}^{r,u} = \sum_{s\in\mathbb{N}\cap[1,|\mathcal{B}|]}\sum_{i\in\mathcal{I}}\sum_{c\in\mathcal{C}} s \times y_{s,i,c}^{r,u}, \forall r \in \mathcal{R}, u \in \mathcal{U}_r \tag{4.13}$$

$$\sum_{s\in\mathbb{N}\cap[1,|\mathcal{B}|]}\sum_{c\in\mathcal{C}} y_{s,i,c}^{r,u} \leq \beta_i^{r,u},\ \forall r \in \mathcal{R}, u \in \mathcal{U}_r, i \in \mathcal{I} \tag{4.14}$$

$$\sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r}\sum_{s\in\mathbb{N}\cap[1,|\mathcal{B}|]}\sum_{i\in\mathcal{I}} t_{s,i,c} \times y_{s,i,c}^{r,u} \leq d,\ \forall c \in \mathcal{C} \tag{4.15}$$

The objective function in (4.1) minimizes the total energy consumption of radio transmission and BBU processing. Equations (4.2), (4.3), and (4.4) ensure that the decision variables are binary, while (4.5) ensures that the power variable is continuous and non-negative. Equation (4.6) ensures that users belonging to one base station cannot use the same RB and that no more than one MCS can be used on this RB. The minimum throughput requirement of a user is ensured by (4.7), while the limit on the total radio transmission power of a user is imposed by (4.8). Equation (4.9) ensures that the signal power of a user on an RB is zero if this RB is not used. Equations (4.10) and (4.11) together ensure that a user transmits using the same MCS index over all its assigned RBs. Knowing that using an MCS index requires the SINR to be above a threshold, equation (4.12) makes sure that if the SINR is lower than the threshold of an MCS index, then the user cannot use this MCS index. To find the processing time and throughput for a user, it is necessary to know the total number of used resource blocks by a user [81], [156]; this is done by (4.13) and (4.14). Finally, (4.15) ensures that each CPU can process the data assigned to it without violating the deadline constraint.

On the other hand, to understand how different objectives can affect the benefit of joint allocation, we consider a modified optimization problem that maximizes the total throughput while maintaining the same constraints as before. The objective function becomes as follows:

$$\text{maximize} \qquad \sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r}\sum_{s\in\mathbb{N}\cap[1,|\mathcal{B}|]}\sum_{i\in\mathcal{I}}\sum_{c\in\mathcal{C}} R_{s,i} \times y_{s,i,c}^{r,u} \tag{4.16}$$

4.3 Matching Game-Based Solution

The optimization problem proposed in the previous section is an NP-hard problem [18]. For this reason, we are interested in proposing a lower-complexity algorithm that can perform as close to the MILP problem solver's optimal solution as possible, which jointly allocates radio and computing resources to reduce overall energy consumption. In other terms, such an algorithm seeks to output the optimal MILP problem solutions in terms of the MCS index, the number of RBs, the power, and the CPU resources. To that end, we propose a two-phase algorithm described as follows:

- Step 1: In the first step of the algorithm, we apply matching theory to associate RBs and MCS index to each user. The details of this step are shown in Section 4.3.1.

- Step 2: The second step aims at adjusting the radio transmission power to minimize the total energy consumption. This step is detailed in Section 4.3.2.

Before delving into the details of each of these two steps, we first go over the definitions of the matching games and stable matching..

Definition 1. *Consider the set of users $\mathcal{U}_r$ associated to a base station b and the set of resource blocks $\mathcal{B}$, a many-to-one matching.*[2] *μ is a function on the set $\mathcal{U}_r \cup \mathcal{B}$ such that:*

- *$\forall b \in \mathcal{B}$ either $\mu(b) \in \mathcal{U}_r$ or $\mu(b) = \emptyset$, i.e., each RB is either matched to one user or unmatched;*
- *$\forall u \in \mathcal{U}_r$, $\mu(u) \in \mathcal{P}(B)$ where $\mathcal{P}(\mathcal{B}) = \{B' \subseteq \mathcal{B}\}$, i.e., each user is matched to a feasible set of resource blocks;*
- *$\forall b \in \mathcal{B}$ and $\forall u \in \mathcal{U}_r$, $\mu(b) = u$ if and only if $b \in \mu(u)$, i.e., an RB b is matched to a user u if and only if the user u is matched to the RB b.*

In a matching game, the two sets of agents rank their preferences and attempt to find a stable match [158]. We define $\succ_u$ as the "prefer" relation of a user u such that $b \succ_u b'$ when user u prefers b to b'. Similarly, $\succ_b$ is the "prefer" relation of RB b such that $u \succ_b u'$ when RB b prefers u to u'.

The definition of a stable matching is given as follows:

Definition 2. *A matching is said to be stable if there are no blocking pairs and it is individually rational, where*

- *a blocking pair is a couple of agent (b,u), such that $u \succ_b \mu(b)$ and (a) $b \succ_u \emptyset$ or (b) there exist $t \in \mu(u)$ such that $b \succ_u t$;*
- *an individually rational matching is when no agents of the two sets would be better off by breaking the current matching, i.e., if $\mu(b) = u$ then $u \succ_b \emptyset$ and $\mu(u) \succ_u (\mu(u) \setminus b)$.*

The problem of allocating RBs to users from different base stations can be seen as a many-to-many matching that maps the users of all base stations to the available RBs. In fact, each user can be associated to many RBs and each RB to many users, under the constraint that no two users from the same base station can use the same RB.

4.3.1 Step 1: Matching algorithm for associating RBs and MCS index to users

We describe and analyze in this section the Step 1 of our proposed matching game-based solution.

[2]This model is also known as the *college admission problem*. When each element of the set $\mathcal{U}_r$ can be matched with at most one element of $\mathcal{B}$ and vice-versa, each element of the set $\mathcal{B}$ can be matched with at most one element of $\mathcal{U}_r$, the matching μ is said *one-to-one*. This model is also known as the *marriage problem*. When each element of $\mathcal{U}_r$ can be matched with many element of $\mathcal{B}$ and vice-versa the matching μ is said *many-to-many*.

4.3.1.1 Algorithm description

In this step, users assume that they fully use the total available radio transmission power P_{Tx}^{max}. The algorithm operates in an iterative fashion. In each iteration, each user forms a Preference List (PL) by ranking the set of available RBs in decreasing order of achievable SINR. The user proposes to only one RB in each iteration. To find the achievable SINR, each user will equally divide P_{Tx}^{max} on all the RBs already assigned to the user plus the RB it is measuring the SINR on. In contrast, RBs rank users based on the decreasing channel gain. After receiving all users' proposals, the RBs accept or reject users. We note that an important condition must be met: two users from the same base station cannot be assigned to the same RB.

In the next iteration, a user only proposes to a new RB if its minimal throughout target is not yet satisfied and if this new RB will not worsen the throughput. We note that if the interference on this new RB is high, the MCS index may become very low that the throughput becomes worse.

We should point out that a user only proposes to a given RB once and does not propose again if it is rejected. Algorithm 3 describes this procedure. Table 4.1 presents the notations used in this algorithm. At the end of this algorithm, all users who transmit operate using their maximum transmission power P_{Tx}^{max}.

It is worth mentioning that in each iteration, the preference of users on RBs could change due to dynamics caused by other users matching with other RBs. However, only semi-dynamic Preference Lists (PLs) are considered to ensure the algorithm converges to a stable matching; this practically means that a user can change its preference for the RBs to which it has not yet proposed, but it does not change the preference over RBs to which it has proposed earlier. The stability proof is demonstrated in the following subsection. At the end of the algorithm, the SINR for each user is calculated on each RB assigned to this user. The assigned MCS is chosen such that it is the highest index among the indexes having threshold lower then the minimum SINR.

4.3.1.2 Stability

In many-to-one matching games, a stable matching may not exist. We introduce a restriction on users' preferences to guarantee the solution's stability. In particular, we assume that users can modify their preferences for RBs to which they have not proposed yet, but they can not change their preferences for RBs to which they have already proposed. Under this assumption, we can state the following stability theorem along with its proof.

Theorem 1. *Step 1 of the matching-based algorithm for associating RBs and MCS index to users converges to a stable matching.*

Proof. Using Definition 2, in order to show that our algorithm converges to a stable matching, we have to prove that (i) there are no blocking pairs and (ii) the matching is individually rational.

Given that the algorithm has converged such that RB b is matched with user u' and RB b' is matched with user u, suppose there is a blocking pair (u, b) that

Algorithm 3: Matching Game for RBs and MCS index allocation

Input: $\mathcal{B}, \mathcal{U}_{r\in\mathcal{R}}, \mathcal{R}, \mathcal{I}...$
Output: RB and MCS assignment.

for $\forall b \in \mathcal{B}$ **do**
$$PL_b = \{u_1, u_2...., u_i...|u_i \in \mathcal{U}_r, \forall r \in \mathcal{R}\ and g_b^{r',u_i,r'} >= g_b^{r'',u_{i+1},r''}, \forall r', r'' \in \mathcal{R}\}$$
end for
for $r \in \mathcal{R}$ **do**
  for $u \in \mathcal{U}_{r\in\mathcal{R}}$ **do**
    $PL_u = \{b_1, b_2, ..., b_i, ...|b_i \in \mathcal{B}\ and\ SINR^{r,u}(b_i) \geq SINR^{r,u}(b_{i+1})\}$
  end for
end for
REPEAT=1
while REPEAT **do**
  REPEAT=0
  for $r \in \mathcal{R}$ **do**
    for $u \in \mathcal{U}_{r\in\mathcal{R}}$ **do**
      $PL_u = \{b_1, b_2, ..., b_i, ...|b_i \in PL_u\ and\ SINR^{r,u}(b_i) \geq SINR^{r,u}(b_{i+1})\}$
      if $th_{r,u} < R_{min}^{r,u}$ and $th_{r,u} < th_{r,u}(PL_u^1)$ **then**
        $prop(u)$ ={PL_u^1}
        $prop(b)$ =$prop(b) \bigcup \{u\}$
        PL_u = $PL_u \setminus prop(u)$
      else
        prop(u) = ϕ
      end if
    end for
  end for
  for $\forall b \in \mathcal{B}$ **do**
    for $\forall u \in prop(b)$ **do**
      REPEAT = 1
      $ACCEPT^b(u)$ or $REJECT^b(u)$
    end for
  end for
end while

consists of a user u and an RB b, such that b prefers u to its current matching u' and u prefers b to one of its associated RB b'. However, if user u prefers b to b', it should have proposed to it following the preference list, and either it did so and got rejected by b which means that b prefers its current matching u' to u, and this is a contradiction of the assumption above, or u prefers b' to b and did not propose to b which also contradicts the assumption above. Consequently, our algorithm

converges to a stable matching without any blocking pair.

The matching is individually rational because a user u will have proposed to an RB b only because its utility (i.e., throughput) increases such that $\mu(u) \succ_u (\mu(u) \backslash b)$, and on the other hand an RB does not accept a user that is not in its preference list, i.e., a user that is ranked inferior to the "no match" option. □

4.3.2 Step 2: Transmission power adjustment for energy consumption reduction

For each user, the transmission radio power is equal. This could lead to having a high SINR on an RB but a lower one on another. The MCS index that can be used is limited by the SINR on the lowest RB. Hence in this second step of our proposed solution, users who are already satisfied decrease their MCS indexes as much as possible under the condition that they remain satisfied. Then all the satisfied users minimize their radio transmission power by solving the following convex problem. To reduce the radio power, users need to set their SINR to be precisely equal to the threshold of the MCS they are using. Every time this problem is run, if at least one previously non-satisfied user becomes satisfied due to the reduction of interference and the ability to use a higher MCS index, this problem will be repeated, but including the newly satisfied users. We define $\overline{\mathcal{U}}_r$ as the set of users of BS r who attained their requirement, $\mathcal{B}^{r,u}$ as the set of RBs assigned to user $u \in \mathcal{U}_r$, and $MCS_{r,u}$ as the lowest MCS index a user can use while remaining satisfied. The problem to be solved is:

$$\text{minimize} \quad \sum_{b \in \mathcal{B}} \sum_{u \in \overline{\mathcal{U}}_r} \sum_{r \in \mathcal{R}} p_b^{r,u} \tag{4.17}$$

$$\text{such that} \quad \sum_{b \in \mathcal{B}^{r,u}} p_b^{r,u} \leq P_{Tx}^{max}, \forall r \in \mathcal{R}, u \in \overline{\mathcal{U}}_r \tag{4.18}$$

$$\sum_{b \notin \mathcal{B}^{r,u}} p_b^{r,u} = 0, \forall r \in \mathcal{R}, u \in \overline{\mathcal{U}}_r \tag{4.19}$$

$$p_b^{r,u} \geq \frac{\gamma_{MCS_{r,u}}^{th}}{g_b^{r,u,r}} (\sigma^{r,u} + \sum_{r' \in \mathcal{R} - \{r\}} \sum_{u' \in \mathcal{U}_{r'}} g_b^{r',u',r} p_b^{r',u'})$$
$$\forall r \in \mathcal{R}, u \in \overline{\mathcal{U}}_r, b \in \mathcal{B}^{r,u}, i \in \mathcal{I} \tag{4.20}$$

4.3.3 Algorithmic complexity

The worst-case scenario of the matching game algorithm (i.e., Step 1 of our proposed solution) refers to the case where all users propose to all RBs. The maximum number of iterations, in this case, is equal to: $|\mathcal{R}| \cdot |\max \mathcal{U}_r| \cdot |\mathcal{B}|$. Regarding the power adjustment algorithm (i.e., Step 2 of our proposed solution), it is a convex optimization problem; hence it can be solved efficiently with simplex/interior point methods. Given that the power adjustment would allow some users to become satisfied because the interference would decrease allowing a higher MCS to be used, the algorithm will be repeated every

time a user becomes satisfied. The worst scenario for this step refers to the case where at each iteration, only one more user gets satisfied. If all users will eventually be satisfied, the number of iterations will not be repeated more than $|\mathcal{R}| \cdot |\max \mathcal{U}_r|$ times.

4.4 Simulation Settings

In this section, the simulation environment is presented along with the performance metrics used for evaluating our proposed solution.

4.4.1 Simulation environment

To code and run the simulation, we use MATLAB. The MATLAB code calls GUROBI Optimizer to solve the MILP problem. We acknowledge that solving an MILP problem is an NP-hard problem, and it is impossible to use it in a real setting where allocation decisions have to be made every 1 ms; however, the MILP solver allows us to measure the potential gains of optimal joint allocation of radio and computing resources. Our study considers an area with a variable number of Base Stations ranging from 1 to 8. Each base station is separated from its neighboring base stations by a minimal distance that follows a uniform distribution between 0.4 km and 0.6 km. Each base station has 24 RBs for transmission, where the frequency reuse equals 1. Each base station serves two users. One would argue that two users per base station is not realistic. However, the reason behind this simple choice is two-fold: From the one hand, as the MILP problem in NP-hard, adding more users and RBs could prevent our MILP problem from producing an optimal solution. On the other hand, we recall that our goal is to compare the performance of the joint allocation of radio and computing resources vs. the sequential allocation and to study the potential benefits of the joint allocation when the goal is energy consumption minimization. As we are targeting the case where radio and computing resources are sufficient to satisfy the QoS of users, having only two users is adequate to model this scenario. The reason why we decreased the size of the problem is the NP-hardness of solving the MILP problem. As we increase the number of users and the number of RBs, the MILP solver fails to output a solution due to the very high complexity of our problem.

The position of each user follows a Poisson Point Process (PPP) such that each user is located in a disk of a radius of 200m and centered at the base station. Each user demands a throughput that follows a uniform distribution between 0.25 and 8 Mbps under the condition that the total demand of the users from the same base station does not exceed 8 Mbps.

To find the SINR threshold of the MCS indexes γ_i^{th}, we used the tables in [159], which map the SINR threshold to a Channel Quality Indicator (CQI) with specified modulation order and code rate. Then we use the MCS table in [156] to map each CQI to its corresponding MCS index. Hence, we end up with a set of possible MCS indexes:{0, 2, 4, 6, 8, 11, 13, 15, 18, 20, 22, 24, 26, 28}.

The maximum user transmission power respects the 3GPP specifications in [160].

Based on it, P_{Tx}^{max} should be equal to 23 dBm with a tolerance of +/- 2dBm. Hence we fix P_{Tx}^{max} = 250mW ≈ 24dBm. We suppose the noise spectral density is -174 dBm/Hz, and the Noise Figure is 8dB.

For the channel gain, we use the ABG model in [161] that models path loss and shadowing at a carrier frequency equal to 2GHz. The path loss equals as function of the distance d and carrier frequency f:

$$PL^{ABG}(f,d)[dB] = 10\alpha \log_{10}(d/1m) + \beta + 10\gamma \log_{10}(f/1GHz) + \chi_{\sigma}^{ABG}$$

where $\alpha = 3.5$, $\beta = 25dB$, $\gamma = 2$. Additionally, χ_{σ}^{ABG} models shadowing which follows a zero-mean normal distribution with a standard deviation $\sigma_{SF}^{ABG} = 7.6dB$. Moreover, we consider the effect of Rayleigh fading such that it follows an exponential distribution with unit mean.

Considering a Cloud-RAN architecture, the baseband processing of these base stations is hosted in a shared BBU pool. We consider just one CPU core with power consumption of P_{comp}^{c} = 30W and a clock frequency equal to 2.4GHz. We also assume that when the CPU core executes BBU functions for users, it consumes the max CPU power P_{comp}^{c}. In contrast, we suppose the power consumption is zero when the CPU is idle.

Our simulation setting focuses on the case where the sum of users' throughput demand is smaller than the system capacity and that the computing resources are sufficient to process the data of all users.

4.4.2 Performance metrics

To analyze the performance of our model, we consider the following performance metrics:

- *Radio transmission energy consumption;* the total radio transmission energy of all users in the system.
- *Computing energy consumption;* the total computing energy consumption of all users in the system.
- *Total energy consumption;* the sum of the total transmission and computing energy consumption in the system.
- *Throughput;* the sum of throughput of all users.
- *CPU Idle time;* The percentage of time for which the CPU is idle.
- *Satisfaction ratio*; The ratio of achieved throughput of a user divided by its minimum requirement.
- *Fairness;* using Jain's fairness index [153] defined in Equation (3.9).

- *Non-satisfied users;* how many users failed to achieve the throughput they demanded?
- *RB utilization;* The percentage of utilization of the total RBs from all base stations.

4.5 Results

In this section, we plot and analyze the performance of the joint radio and computing resource allocation vs. the sequential resource allocation, considering the two objectives of minimizing the total energy consumption and maximizing the total throughput. As we mentioned before, the sequential allocation separates the allocation of radio resources from that of computing resources by solving them in order. So, when considering the objective of energy consumption minimization in the sequential allocation, the radio allocation aims at minimizing the radio transmission energy consumption; while the computing allocation minimizes the computing energy consumption. Moreover, we compare the performance of the MILP-based solutions to that of the proposed low complexity algorithms:

- the matching with fixed preference lists; this is step-1 of our solution, except that the preference lists are initialized in the beginning and are no more modified in each iteration;
- the matching with semi-dynamic preference lists; this is step-1 of our algorithm.
- the matching with semi-dynamic preference lists followed by radio power adjustment; this is our proposed two-step solution.

The performance is measured using the metrics defined in Section 4.4.2 as a function of the number of base stations connected and managed by the same BBU pool. In the simulation, we only consider the allocation for one Transmission Time Interval (TTI), which is equal to 1 ms. The simulation is repeated 25 times for the ILP problems and 100 times for the matching algorithms, and the 95% confidence intervals are plotted.

4.5.1 Transmission and computing energy consumption

Figures 4.3a, 4.3b, and 4.3c show the performance of the joint allocation problems vs. the sequential one concerning the transmission energy consumption, computing energy consumption, and total energy consumption, respectively. The energy consumption is measured during one TTI. We clearly notice that :

- When considering the objective of minimizing total energy consumption, the joint radio and computing resources allocation, compared to the sequential one, consumes more energy for radio transmission (as shown in Figure 4.3a), less energy for computing (Figure 4.3b) and less total energy (Figure 4.3c). For instance, when the number of BSs is equal to 8, the joint radio and computing allocation reduces the total energy consumption by 21.3% compared to the sequential counterpart.

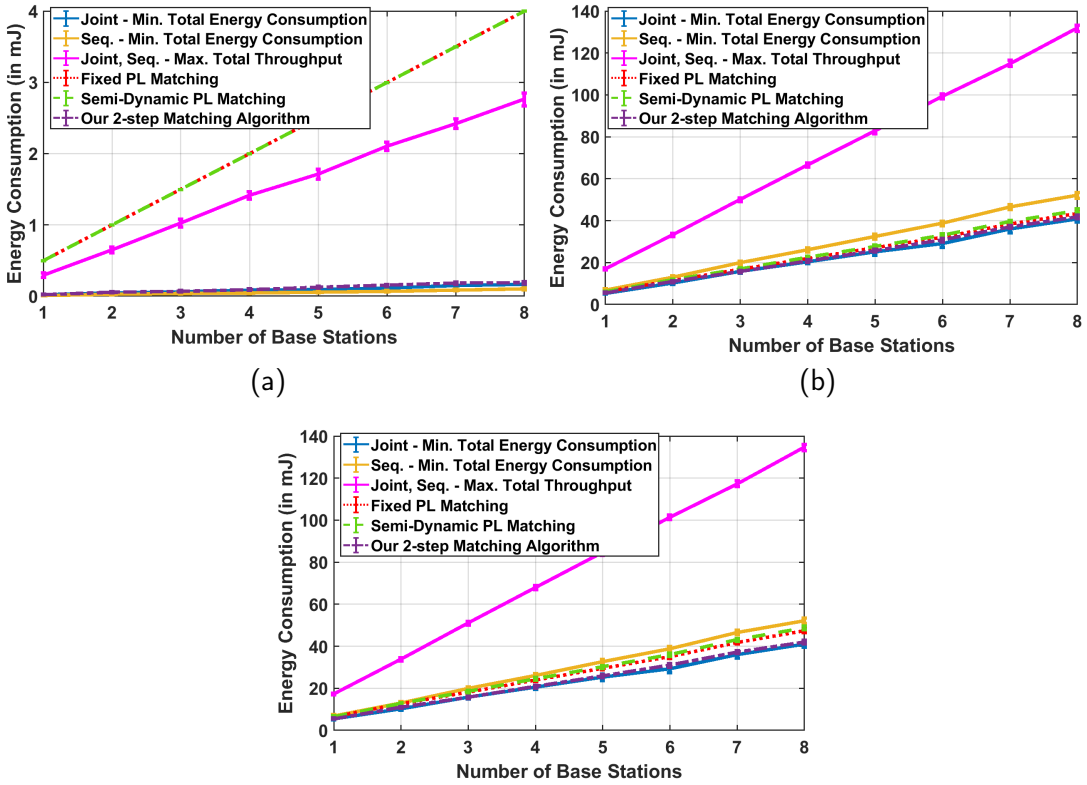


Figure 4.3: Energy consumption as a function of the number of base stations in the BBU pool: (a) Tx energy consumption (in mJ), (b) Computing energy consumption (in mJ), (c) Total energy consumption (in mJ)

- When considering the objective of throughput maximization, both joint and sequential allocation use all the available radio and computing resources to maximize the throughput. Hence, they both have identical performance for throughput maximization. This will be further explained later on in this section. On the other hand, maximizing throughput objective would consume up to 325% more total energy than the joint allocation with the objective of minimizing energy consumption would do.

- The proposed matching algorithms without power adjustment fully use the available transmission power. However, when combining the power adjustment algorithm with the matching-based algorithm, the radio transmission energy consumption is very close to that of the joint variation of the energy consumption minimization algorithm.

- Regarding the computing energy, the fixed PL and the Semi-Dynamic-PL matching combined with power adjustment perform very close to the joint allocation

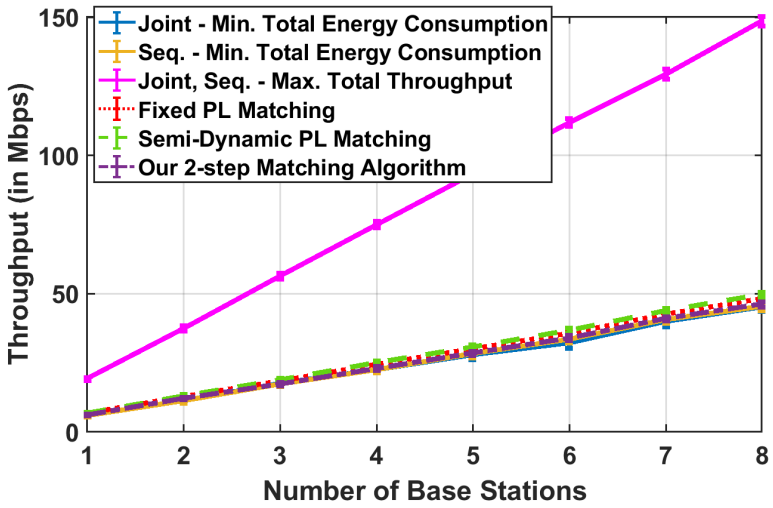


Figure 4.4: Throughput as a function of the number of base stations in the BBU pool

with the objective of minimizing the energy consumption. The Semi-Dynamic-PL Matching consumes a little more. The reason behind these results is that Semi-Dynamic-PL Matching algorithm allows for using even higher MCS indexes, which require more computing resources. On the other hand, combining it with power adjustment allows for decreasing the MCS to the least one that can satisfy the required throughput of a user. This permits to reduce the excess allocated resources and help satisfy more users. Hence the computing energy consumed by the latter algorithm is close to the MILP-based joint allocation of the energy consumption minimization. Overall the total energy consumption for the combined matching and power adjustment is the closest to the joint energy consumption minimization algorithm and is less than the other two matching algorithms.

4.5.2 Throughput

The performance concerning the throughput metric as a function of the number of base stations in the BBU pool is plotted in Fig. 4.4. Since the objective of minimizing the total energy consumption must guarantee the requested throughput for every user, both the joint and sequential allocation achieve similar results with this objective. The slight differences result from the different decisions on the MCS indexes and number of RBs; together, they control the TBS size, which indicates the throughput. The matching combined with power adjustment achieves the lowest throughput while it remains close to the optimal solution of the MILP problem. The other matching algorithms achieve slightly higher throughput. The reason behind this tendency is that some users can use high MCS indices when employing high radio power, allowing them to do far more than what they had expected.

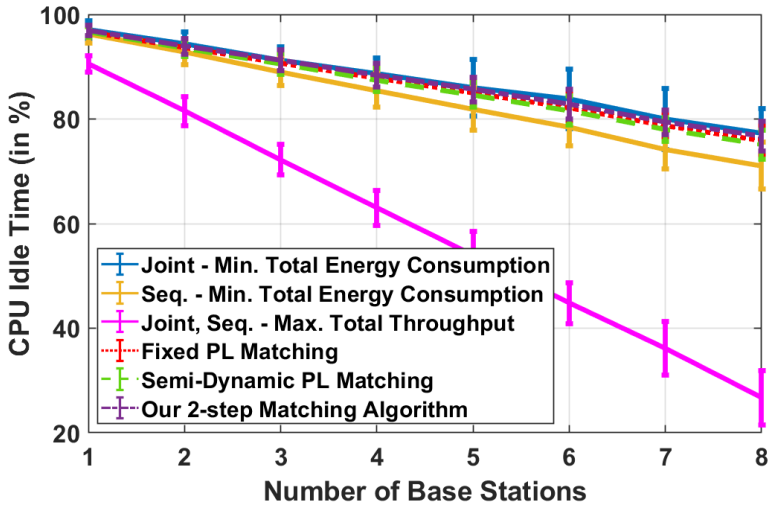


Figure 4.5: CPU Idle time as a function of the number of base stations in the BBU pool

4.5.3 CPU idle time

Fig. 4.5 shows the percentage of CPU idle time as a function of the number of base stations connected to the BBU pool. Using the computing model described in section 4.2.1, minimizing the energy consumption is consistent with reducing the CPU utilization, or in other words, increasing the CPU idle time. This explains why the joint allocation with the objective of minimizing energy consumption, which is the best algorithm that reduces compute energy consumption, achieves higher CPU idle times than the sequential allocation. Again, the joint and sequential allocation with the objective of maximizing throughput have much lower CPU idle time than energy consumption minimization objectives counterparts. This is interpreted by the fact that maximizing throughput algorithms make the most use of all available computing resources. On the other hand, the matching-based algorithms, and especially the one that enables power adjustment, preserve outcomes that are similar to the joint MILP-based problem with energy consumption minimization.

4.5.4 Fairness and users' satisfaction

Given that the proposed matching algorithms are not guaranteed to satisfy the requirements of all users as the MILP problems do, we study and plot the graphs of fairness and the percentage of non-satisfied users for the matching-based algorithms as a function of the number of base stations managed by the BBU pool in figures 4.6 and 4.7. Compared to the fixed PL matching, the results show that allowing the semi-dynamic preference lists increases the fairness and the number of satisfied users. Additionally, the performance in terms of fairness and the number of satisfied users is further enhanced when enabling the power adjustment algorithm. Furthermore, we plot the box-plots of the satisfaction ratio of non-satisfied users in figure 4.8. The results show that combining matching and power adjustment is better than the other two algorithms, recalling

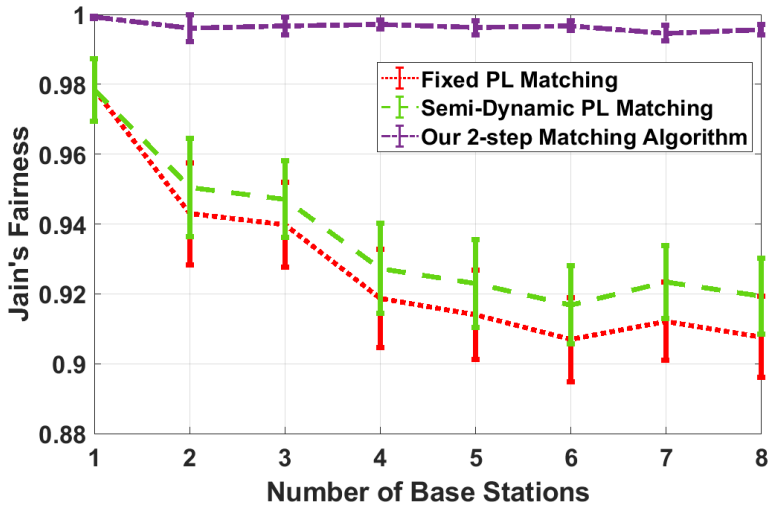


Figure 4.6: Fairness as a function of the number of base stations in the BBU pool

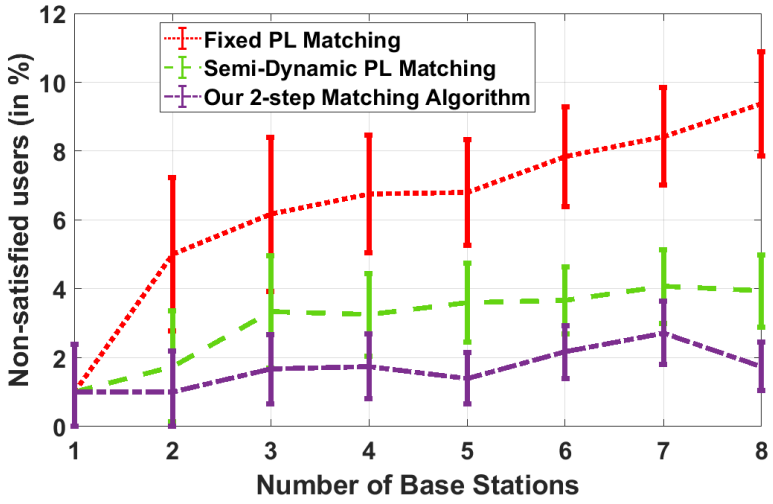


Figure 4.7: Percentage of Non-satisfied users as a function of the number of base stations in the BBU pool

that the number of non-satisfied users dropped when combining the algorithms. This combination not only satisfies more users but also improves the satisfaction ratio among non-satisfied users.

4.5.5 MCS and RB selection

To understand how the various algorithms behave, we observe the algorithms' decisions based on the MCS indexes assignment and the number of RBs assigned to users. Fig. 4.9 shows the percentage of utilized RBs in each base station as a function of the total number of base stations connected to the BBU pool, and Fig. 4.10 shows the cumulative distribution probability of the selected MCS indexes for each algorithm. Consider-

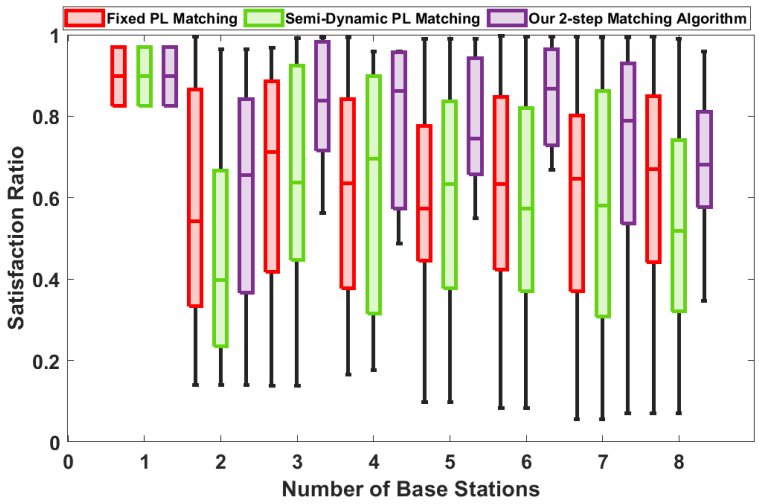


Figure 4.8: Box plot of the satisfaction ratio of nonsatisfied users

ing the energy consumption minimization objective and analyzing both figures, the joint algorithm tends favors allocating users with low number of RBs but high MCS indexes. In fact, achieving the same throughput for a given user could be done by either using a lower number of RBs and a high MCS index if the SINR level permits so, or using a higher number of RBs with a lower MCS index [156]. The joint allocation with the objective of minimizing energy consumption would go for the first alternative which is increasing the MCS index but transmitting over a low number of RBs. This requires increasing the transmission power to increase the MCS over these RBs but would decrease the required computing resources. As a result, the computing energy consumption decreases, thus decreasing the total combined energy consumption. In contrast, the sequential allocation firstly solves the radio allocation that minimizes transmission power independently of the computing resource allocation. While the results of the radio allocation (i.e., the MCS index and the number of resource blocks) affect the computing resources requirements, the computing resource allocation has to satisfy these requirements without modifying any of the radio decisions such as the MCS index or the number of RBs. In general, this leaves a tighter room to minimize computing energy consumption. It could only be possible to adjust the CPU frequency and select which CPU to process the data in case of non-homogeneous CPU power consumption. The results show that the sequential algorithm minimizes the transmission power and spreads the data over a bigger number of RBs but with a lower MCS index. On the other hand, the maximizing throughput algorithm uses all the RBs in each base station and the maximum transmission power for every user. This justifies the very high RB utilization and the usage of high MCS indexes, as Fig. 4.9 and Fig. 4.10 show. However, the maximizing throughput algorithm should ensure its selections do not increase the interference, worsening performance. On the other hand, the fixed PL matching algorithm has an RB utilization that is very close to that of the joint energy consumption minimization. When the preference lists are

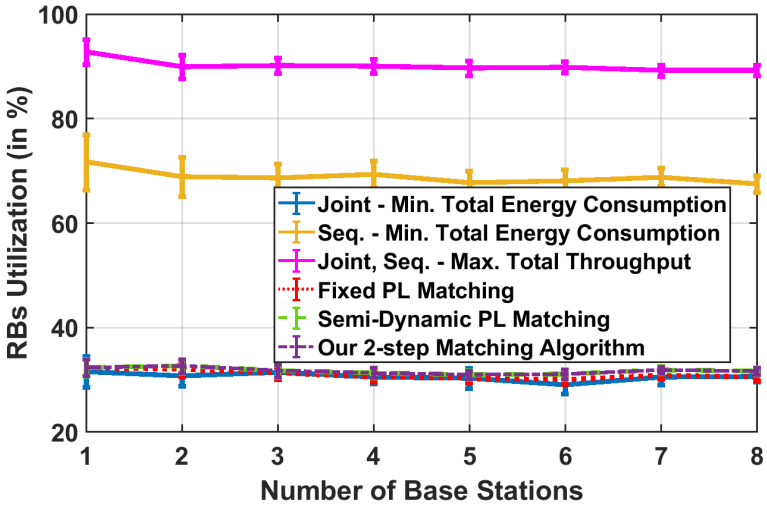


Figure 4.9: RBs Utilization as a function of the number of base stations in the BBU pool

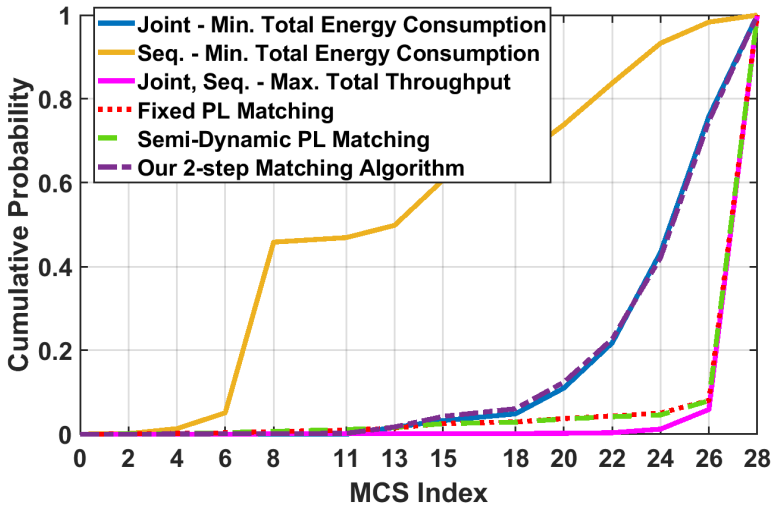
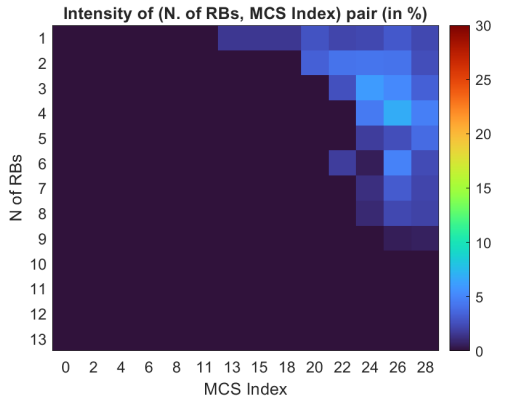


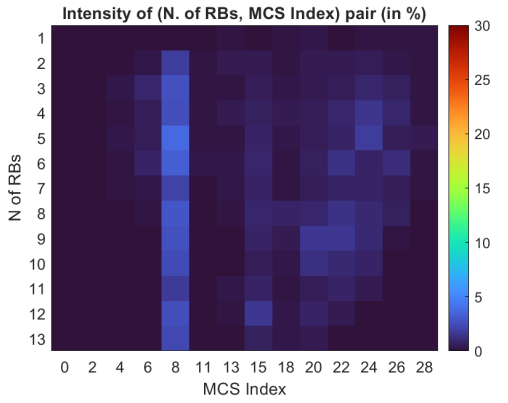
Figure 4.10: The cumulative distribution function of MCS assignment

allowed to be partially modified, some users will take advantage of that in an attempt to improve their throughput by using more resource blocks. That is why the semi-dynamic PL matching with and without power adjustment uses a slightly bigger number of RBs. Regarding the MCS, Fig. 4.10 shows that both matching algorithms without power adjustment use high MCS indexes. When power adjustment is applied, and the power gets decreased so that a user does not receive more than its request, it would be possible to use lower MCS indexes.

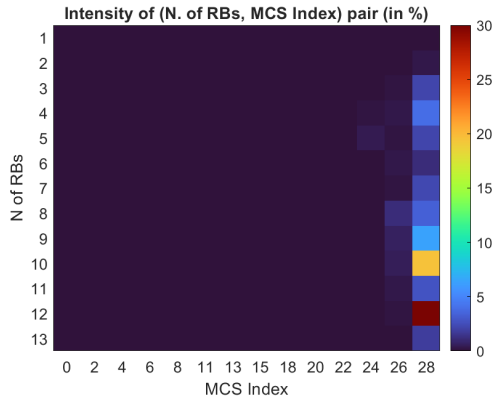
Fig. 4.11 further supports this previous explanation. In fact, these heat maps show the intensity of assigning the pair composed of 1) the number of resource blocks and 2) MCS indexes to users. While the Joint energy consumption minimization algorithm intensely allocates a lower number of RBs to users and a very high MCS index, the



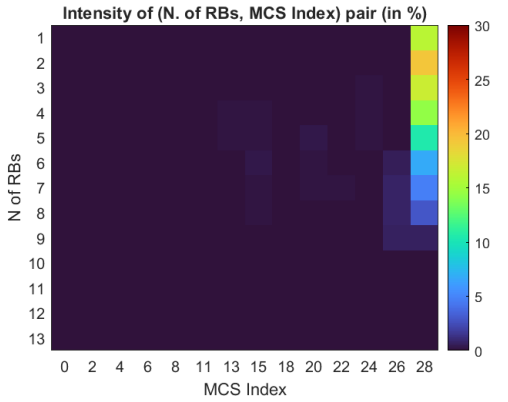
(a) Joint - Minimize energy consumption



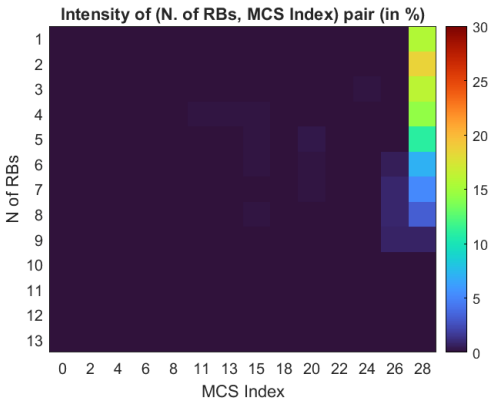
(b) Sequential - Minimize energy consumption



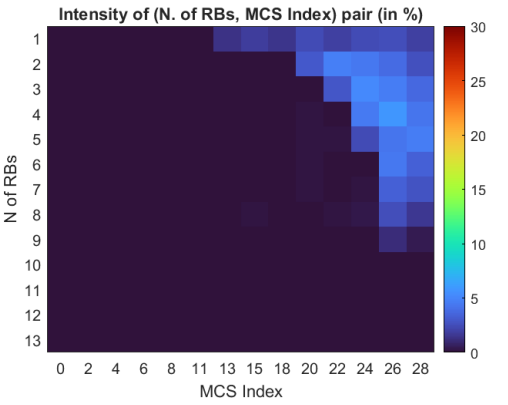
(c) Joint, Sequential - Maximize throughput



(d) Matching - Fixed PL



(e) Matching - Semi-Dynamic PL



(f) Matching + Power adjustment

Figure 4.11: The intensity of assigning a pair of (number of Resource Blocks, MCS index) to users

sequential counterpart favors assigning a bigger number of RBs but lower MCS indexes. On the other hand, the maximizing throughput algorithms assign more RBs and high MCS indexes to users. Moreover, the matching algorithms without power adjustment tend to use fewer RBs and excessively high MCS indexes. In contrast, the matching with power adjustment decreases the MCS index for some users, which automatically improves the MCS for others. Hence, while it uses a lower number of RBs, this algorithm uses high, but not excessively high, MCS indexes to satisfy more users.

4.5.6 Scarce radio resources case

As final notes, we have reduced the size of the MILP problem and made the problem tractable by using a small number of resource blocks, a small number of users, and a small number of base stations. The reason is that an MILP problem is known to be NP-hard. However, we can generalize our conclusions to a higher dimensional problem. Moreover, the results of the maximizing throughput objective help us understand the performance of the energy minimization objective in case all the radio resources are required to satisfy the demands of users. Suppose that the QoS requirements (i.e., minimum throughput) of users are increased so that the throughput maximization objective can fulfill the needs without being able to improve the assigned data rates. This means, more or less, all the radio resources (i.e., RBs and transmission power) are needed to satisfy the minimum throughput requirement (i.e. Eq.(4.7)). On the other hand, since the energy consumption minimization objective should fulfill the minimum requirement constraint, it will give the same results as the throughput maximization objective. In other terms, all the radio resources are needed, and it is not possible to use less to save energy. In such a case, the joint and sequential allocation will perform the same, even if the goal is energy consumption minimization, as long as the bottleneck happens at the level of the radio resources. In case the computing resources are scarce, and the radio resources are sufficient, Eq. (4.7) must be relaxed. In such a case, the joint allocation should perform better than the sequential one, as we showed in the previous chapter.

Finally, the proposed matching algorithm can perform close to the joint energy consumption minimization MILP algorithm, with good fairness among users and much lower complexity. Hence, it can serve as a suitable alternative to the MILP problem for efficient implementation by operators.

4.6 Summary

In this chapter, we have studied the performance of joint radio and computing resource allocation in Cloud RAN. We have formulated a Mixed Integer Linear Programming (MILP) model and compared the performance of the joint allocation with respect to a sequential model, considering the objectives of minimizing energy consumption and throughput, respectively. The results demonstrate that when the computing resources are sufficient, the joint allocation is beneficial and achieves performance gains by reducing the total energy consumption when the objective is energy consumption minimization.

Given that we used a high-complexity problem solver to analyze the benefits of joint allocation and that it is impractical to use such a solver in a real implementation, we proposed a lower-complexity matching algorithm with a power adjustment mechanism to perform close to the MILP optimization problem. Our results show the proposed algorithm can perform close to the joint allocation for the energy consumption minimization problem but with much lower complexity.

Chapter 5

Machine Learning-Based Application for Computing Resource Allocation in O-RAN

Machine Learning (ML) has gained a lot of momentum in wireless and mobile networks. This has driven the novel cloud and virtualized RAN architecture known as Open-RAN to natively support ML models. In this chapter, we investigate two ML applications to allocate computing resources. The first application uses Recurrent Neural Network (RNN) as a Deep Learning (DL) model that aims at learning the performance of the Integer Linear Programming (ILP) solver of the models of coordination between radio and computing schedulers presented in Chapter 3. Given the NP-hardness of solving an ILP model, the proposed DL model demonstrates its ability to perform close to the ILP solver while reducing the required execution time. The second application that we study in this chapter is related to maximizing the operator's profit. Considering the case of limited computing resources, we formulate a Mixed Integer Linear Programming (MILP) model that allocates computing resources to users such that the operator profits are maximized. However, the high complexity of solving an MILP problem inspired us to propose solving the problem using a policy gradient-based Reinforcement Learning (RL) model. The proposed RL model achieves suboptimal results but with lower complexity.[1]

5.1 Introduction

The success of Machine Learning (ML) in different fields raised its stakes in wireless and mobile networks. The ability of ML to provide efficient solutions with low time complexity makes it suitable for different applications. Indeed, ML has recently shown many potentials for resource allocation in Mobile Networks [20]. Machine learning,

[1]The contents of this chapter are partially presented in [2], [3], [5]. Additionally, part of this work was completed during my research visit at NETCORE lab in uOTTawa

precisely Deep Learning (DL), has demonstrated its ability to provide solutions to various problems. Several papers have investigated the ability of DL to provide low-complexity alternatives to high-complexity optimal algorithms as in [120] and [121]. DL networks are not only able to closely imitate the performance, but they can also have much lower execution time, making them suitable for practical implementation. Lately, RL has been widely used due to its ability to provide models that can autonomously solve problems and adapt to dynamic environments. In RL, an agent, in a given state or observation, interacts with its environment and receives rewards or penalties. Intending to maximize rewards, the agent selects actions that maximize its reward. Reinforcement Learning (RL) was used in [142] to satisfy the different communication and computing users' demands. Q-Learning is used in [133] to allocate radio resource blocks (RB) and power. Moreover, in [147], Deep Deterministic Policy Gradient (DDPG) is used to allocate RBs and power, while a Double Deep Q-Network-learns the optimal RAN slicing strategies. Additionally, RL was used in [148] to select the functional split that minimizes the routing and computational cost. In [138] and [139], resource block allocation has been considered for mission-critical services and micro-grid communication using RL. The usage of Machine learning techniques in the literature has been discussed in chapter 2. For more details about DL and RL, please refer to the Appendices A and B.

The demonstrated abilities of ML have lead O-RAN Alliance, which is the entity responsible for the standardization of the Open-RAN architecture, to move towards a RAN architecture that supports machine learning models [15]. In O-RAN, machine learning can be used in different control loops for different timescales. The O-RAN Alliance defines the near Real Time Radio Intelligent Controller (near-RT-RIC) and non Real Time Radio Intelligent Controller (non-RT-RIC). ML models can run in these controllers as xApps or rApps, respectively. While the controller support timescales of 10 ms and 1 s, respectively, recent proposals have been made to support timescales of 1 ms by running ML models as distributed Apps (dApps) [54].

In Chapter 3, we have focused on coordinating the allocation of radio and computing resources in Cloud-RAN. We have studied how to manage the limited computing resources when they are insufficient for processing the data frames of all users' data. Two Integer-Linear-Programming (ILP) problems have been formed to achieve optimal resource allocation while considering the objectives of maximizing users' throughput and satisfaction rate, respectively. These ILP problems permit coordination between radio and computing resource schedulers, requiring not only the availability of radio resources but also the availability of computing resources at the O-Cloud. This coordination specifically allows for adjusting the Modulation and Coding Scheme (MCS) index, which controls the modulation order and the coding rate of the data to be transmitted. While the throughput of a user increases with the MCS index, the time required to process this user's data also increases. Solving ILP problems is known to be NP-hard with exponential time-complexity [18]. In fact, it could be impractical for mobile operators to implement them, given that scheduling decisions should be made in a limited amount of time.

In this chapter, we consider the context of limited computing resources in Open-RAN. Given that the scheduling decisions have to be taken every 1ms, we suppose that our ML proposed models are run in an intelligent decision maker that has control over all the connected Open-Distributed-Units (O-DUs), which was discussed earlier in Chapter 2. Given that the Open-RAN adopts the functional split option-7.2, the most resource-consuming functions are executed in the O-DU. Hence, this entity using our proposed application decides which user frames have to be processed and which user frames have to be dismissed.

This chapter consists of two parts. In the first part, we investigate the ability of Recurrent Neural Networks (RNN), a branch of deep learning, to sub-optimally depict the performance of two ILP-based algorithms proposed in chapter 3. This depiction should provide a practical alternative to these algorithms, where the first aims at maximizing the total throughput, and the second maximizes the total users' satisfaction.

In the second part, we consider a scenario in which the mobile operator requests computing resources to process its users' frames while considering the various service types with different quality of service (QoS) and deadline requirements. In particular, we consider the two types of services, the enhanced Mobile BroadBand (eMBB) and the Ultra Reliable Low Latency (URLLC) services. The operator aims to maximize its profits; so we formulate a Mixed Integer Linear Programming (MILP) problem that allocates computing resources to users' frames while considering different services' priorities. However, as solving the MILP problem is highly complex. We resort to Reinforcement Learning (RL) techniques, particularly the Policy-Gradient-based model, to allocate computing resources to users.

5.2 The Deep Learning-Based Model for Computing Resource Allocation

In this part, we present the DL-based model to allocate computing resources in Open-RAN. First, the ILP formulations from section 3.2.2 are recalled. Then, the DL architecture is presented.

5.2.1 Problem formulation

In our study, we consider an Open-RAN context that consists of an O-Cloud hosting the baseband processing of the Open Central Units (O-CUs) and the Open Distributed Unit (O-DUs) connected to a set of Open Radio Units (O-RUs). The O-DUs and the O-CUs use the resources of the O-Cloud. We consider the uplink direction since the most computing-heavy operation in this direction is the decoding function [13]; it has an execution time seven times higher than the encoding function's execution time. Thus, the demand for computing resources overwhelmingly comes from uplink traffic [80]. When the computing resources are insufficient to process the data from all users, a group of users will have their data processed, while others will be dismissed. For that, operators should adopt the selection strategy depending on the service-level agreements. Excluding some users is unavoidable, and it comes with negative consequences for these excluded

users. First, these users will have wasted transmission power. Second, the Hybrid Automatic Repeat Request (HARQ) mechanism will be triggered when a transmission is not acknowledged within 8 ms, as explained in Chapter 3.2. According to [86], this leaves a duration of 2ms to process uplink data. Otherwise, not respecting this deadline will trigger an HARQ re-transmission. If this re-transmission is successfully processed, the delay in the system will be increased. If it is not, the transmission power will again be wasted. This has driven us in chapter 3 to propose a coordination scheme between the radio and computing schedulers that allows for adjusting the MCS index of users. Initially, the MCS index is selected depending on the channel condition while ensuring that the error probability should be no more than 0.1 [151]. However, adjusting the MCS index to a lower one allows for decreasing the required processing time at the expense of reducing throughput. This would grant an additional degree of freedom and can better help exploit computing resources. Regarding users who will not be processed, they will be notified not to send data so that they do not waste their transmission power.

The system under study consists of: a set of O-RUs $\mathcal{R}$, sets of users $\mathcal{U}_r$ per each O-RU r, and a set of homogeneous CPU cores $\mathcal{C}$ available in the O-Cloud. The set of possible MCS indexes used for the radio transmission is $\mathcal{M}$. For each O-RU r, the coordination policy between radio and computing schedulers attributes to user $u \in \mathcal{U}_r$ an MCS index $m \in \mathcal{M}$ lower or equal to the maximum allowed one $M_{r,u,max}$; the one initially chosen by the radio scheduler considering only the radio conditions. We note that the coordination policy does not adjust the MCS index to a higher index to avoid a probability of error higher than 0.1. Based on the chosen index m, user u transmits an amount of data equal to $b_{r,u,m}$ (i.e., measured in bits) which is determined according to [150]. The latter is determined based on the MCS index and the number of RBs which are mapped to the Transport Block Size (TBS). We note that the TBS is the payload that is carried over the physical layer. When a user uses, $M_{r,u,max}$ index, $b_{r,u,m}$ is denoted as $b_{r,u,max}$. Additionally, the processing time model from [81] is used to determine the time, denoted by $t_{r,u,m}$, which is required for processing user's u data. It has been presented in Section 3.3.

Two coordination policies are considered, the first aims to maximize the total system throughput, and the second aims to maximize the users' satisfaction. As defined in Section 3.3.2, the user satisfaction ratio is the ratio of the throughput achieved when the user operates using the adjusted MCS index to the maximum throughput obtained when operating using the maximum allowed MCS index (i.e., the one that depends solely on the channel conditions). We recall the two ILP problems presented in Section 3.2.2:

1. *Maximize Total Throughput (MTT)*: It aims to maximize the overall system throughput; in particular, it favors users with high MCS indexes and sacrifices those with lower indexes

2. *Maximize total Users' Satisfaction (MUS)*: It aims at maximizing the total users'

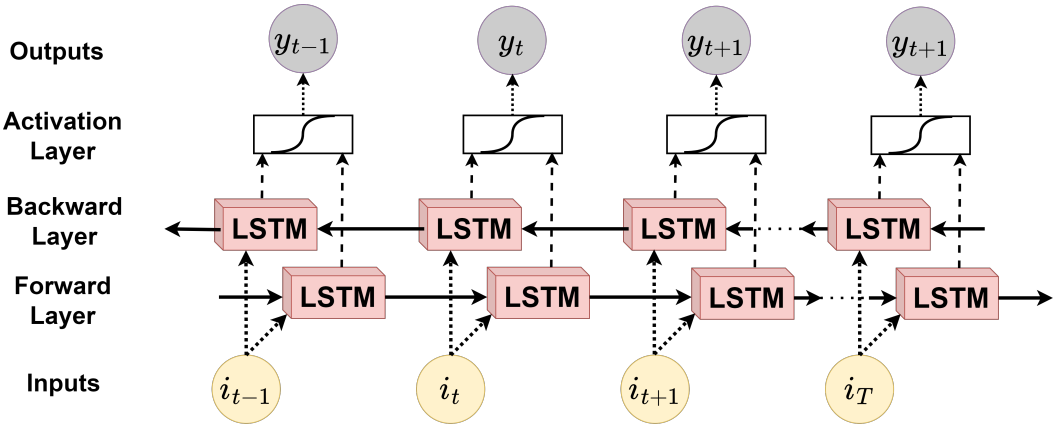
satisfaction ratio. It permits to assign MCS indexes to users, such that these MCS indexes do not deviate much from the maximum allowed MCS indexes.

Given that Integer Linear Programming is known to have an NP-hard complexity, we propose in this chapter to replace these algorithms with a Recurrent Neural Network (RNN), which should imitate the performance of these algorithms with a much more reduced execution time. In the next section, we discuss the RNN-based algorithms.

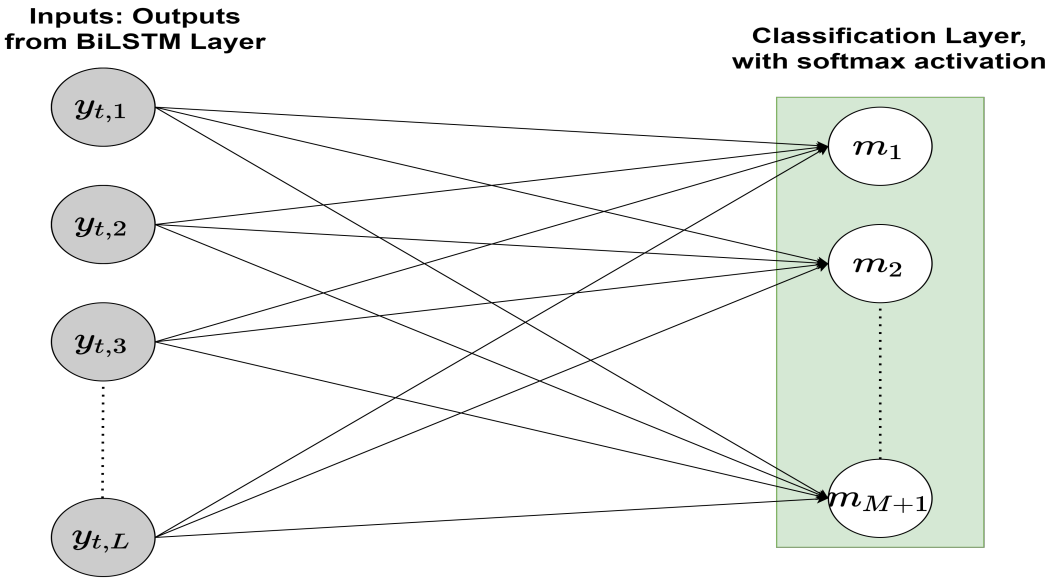
5.2.2 Recurrent Neural Network algorithms

In supervised Machine Learning, a model is given a set of inputs and a set of outputs (i.e., labels) and aims to learn a mapping between the inputs and these desired outputs. Recurrent Neural Networks are suitable for applications on data with sequential structures. This makes them a great choice for naturally sequential data such as texts, speeches, etc. In such types, the elements in a sequence are more or less dependent on each other. For example, in English language translation use cases, the meaning and the role of each word in the sentence depend on the previous and following elements in the sequence. Hence, such dependency should be taken into consideration by the ML model for it to be able to give accurate results. RNN can process sequential input data of a variable sequence length and can detect dependencies between the elements of the sequence. A very famous layer used in RNN is known as Long-Short-Term-Memory (LSTM). This model is designed to counter the phenomenon of the vanishing gradient problem [20] that slows the learning speed and can hinder the model from improving. While an LSTM can memorize dependencies in one direction, a Bi-Directional LSTM (BiLSTM) uses two LSTM layers instead to traverse the sequence in two directions: a forward direction and a backward direction. Thus, it can capture more dependencies and have a wider scope when making predictions. A BiLSTM-based RNN suits our resource allocation problem presented in the previous section. Given a sequence of users from all O-RUs, each user may or may not be admitted. If admitted, the RNN should decide what MCS should be used. Besides, users' selections are dependent because computing resources are limited and shared among them; hence, the decision of a user can affect the decision of another. The RNN model should exploit such dependencies to make decisions that align with the two objectives presented in section 5.2.1. Precisely, this task is handled by the BiLSTM layer.

Fig. 5.1a shows the basic architecture of BiLSTM where $i_{t-1}, i_t, i_{t+1}, ..., i_T$ is the input sequence. Each element in this sequence is a feature vector that carries information about the corresponding item (i.e., a user in our case). This element is input to the two LSTM layers that correspond to the forward and backward directions. The output vectors from each LSTM layer are concatenated and fed into an activation layer; a layer used to increase the ability of the network to learn non-linear dependencies. We use the hyperbolic tangent function, as an activation function that produces an output vector of $L = 2H$ elements. We note that H is a hyperparameter that indicates the number of hidden layers in an LSTM. Each element in the output vector has a value between -1 and 1.



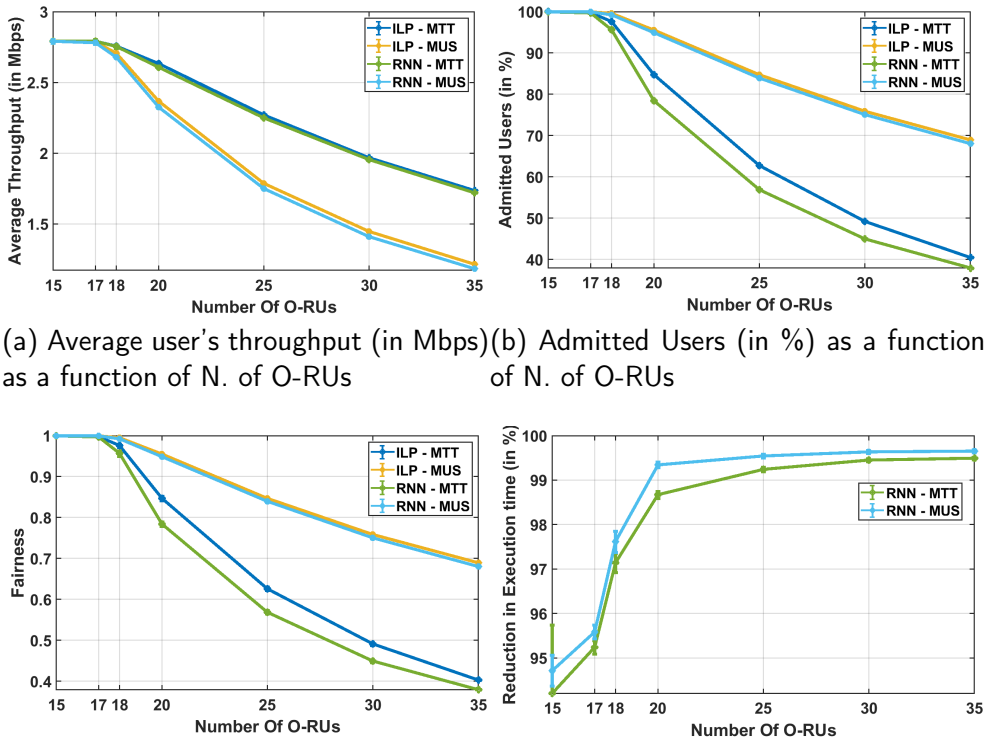
(a) BiLSTM Architecture [162]



(b) Fully-Connected and Classification Layers

Figure 5.1: RNN Model Architecture

In our model, we use a sequence-to-sequence classification. This means that each element (i.e., that represents a user) in the sequence should have an output (i.e., a decision on the MCS index). Each element in the input sequence is in fact a vector of features. The feature vector for a user $u \in \mathcal{U}_r$ contains these features: the MCS indexes m such that $m \leq M_{r,u,max}$, the corresponding throughput $b_{r,u,m}$ and processing time $t_{r,u,m}$ for each MCS index m, the number of RBs of this user, the total demanded throughput of all users, and the total demanded processing time of all users. Our proposed RNN model consists of a BiLSTM layer with $H = 25$ hidden layers and $L = 50$; the value of the hyperparameter H is determined by trial and error. For an input i_t, the output of the BiLSTM is y_t is a vector of size L and consists of the elements



(a) Average user's throughput (in Mbps) as a function of N. of O-RUs

(b) Admitted Users (in %) as a function of N. of O-RUs

(c) Jain's Fairness Index as a function of N. of O-RUs

(d) Reduction in Execution time (in %) with respect to the ILP counterparts as a function of N. of O-RUs

Figure 5.2: Performance evaluation of the different scheduling solutions

$[y_{t,1}, y_{t,2}, ..., y_{t,L}]$. This vector is fed into a fully-connected network shown in Fig. 5.1b. The fully-connected layer has an activation function known as softmax, and it is used for multi-class classification [20]. The classification layer contains $M+1$ neurons where $M = |\mathcal{M}|$ (i.e., the possible labels of the classifier are the MCS indexes in the system plus another label to indicate a user is dismissed. Among the $M+1$ neurons, the one with the highest activation value gives the label (i.e., the decision).

To train the RRN model, we use a large data set labeled according to the ILP solver results. Our RNN model uses the training dataset to learn how to imitate the performance of the ILP problems as accurately as possible.

5.2.2.1 Simulation settings

In our simulation, we consider an O-Cloud that has 4 available CPU cores. The number of O-RUs connected to the O-Cloud varies from 15 to 35. Each O-RU operates on 20 MHz bandwidth, and has 100 RBs available for allocation. The 100 RBs are allocated to the users of each O-RU. Each user is assigned a uniformly random number of RBs between 10 and 30. As in Chpater 3, we use the probability distribution function of MCS indexes shown in Figure 3.3b which is based on real traces and also used in [13].

In addition, we use the processing time model presented in Section 3.3. We suppose that the 4 CPU cores have a 4GHz clock speed. To determine the throughput, we refer to [150] to find the Transport Block Size (TBS). The TBS is determined as a function of the number of the allocated RBs and the used MCS index, over the Transmission Time interval (TTI) that is equal to 1ms. The simulation is coded using MATLAB, and the ILP problems are solved using CPLEX for MATLAB.

5.2.3 Model training

To train the RNN model to imitate the ILP-problems performance, we generate a dataset by running the simulation 7500 times. For each run, the results of the ILP solvers were added to the data set. Given a group of users from all O-RUs, the ILP solver provides the allocation decisions when MTT and MUS problems are solved. To adapt the data to our model, the users in each run are arranged as a sequence, and the allocation results for MTT and MUS are used as labels to train the two RNN models. 67% of the set is used for training, and 33% is used for testing. The accuracy of the training and testing sets are provided in table 5.1.

Table 5.1: Train/Test Accuracy

Model	**MTT**	**MUS**
Training-set Accuracy	*97.27%*	*97.85%*
Testing-set Accuracy	*97.23%*	*97.84%*

5.2.4 ILP vs. RNN comparison

To compare the performance of the ILP problems and RNN models, we are interested in four performance metrics of Section 3.3.2:

1. *Average Throughput*: The average user's throughput.
2. *Admitted Users*: The percentage of admitted users with respect to all users in the system.
3. *Fairness*: Jain's Fairness index [153] is used to compare the fairness among the algorithms. It is defined in section 3.3.2.

   For each user $u \in \mathcal{U}_r$, $s_{r,u}$ is its satisfaction ratio (i.e., the ratio of the attained throughput to the maximum achievable throughput achieved when using the maximum allowed MCS index
4. *Execution Time*: The percentage of reduction of execution time for RNN models with respect to the ILP models.

We consider the allocation for only one TTI, and we ran the simulation 1000 times. We provide the 95% confidence intervals.

Fig. 5.2 shows the performance of the ILP algorithms and their RNN counterparts as a function of the number of O-RUs connected to the O-DUs in the O-Cloud. We note that the O-Cloud becomes fully-loaded when the number of O-RUs is equal to 17, after which the algorithms start behaving differently. Before this point, the algorithms perform equally as there would be enough computation resources to admit all users using their maximum MCS index.

The MTT algorithms achieve higher average throughput per user in comparison to MUS algorithms. However, the latter achieves better performance concerning the metrics of Admitted Users and Fairness. From the real MSC-distribution provided in [13], the bulk of users have MCS indexes between 4 and 10. The users would request less processing time than users with high MCS indexes would. Hence, maximizing the sum of users' satisfaction will lead to favoring the allocation of the bulk of users with low MCS indexes; this justifies the better performance of MUS algorithms with respect to the Admitted Users and Fairness metrics.

More importantly, the figures show that the RNN networks were able to learn MTT and MUS's objectives, respectively. Fig. 5.2a shows that RNN-MTT performs very closely to its ILP counterpart with respect to the average throughput metric. The highest difference between the graphs is minimal and is equal to 0.02 Mbps. Concerning the other metrics in Fig. 5.2b and Fig. 5.2c, the margin of difference between ILP-MTT and RNN-MTT increases. RNN-MTT dismisses up to 6.2926% more than the MTT-ILP does and may lose up to 0.062 points in the fairness index, in comparison with MTT-ILP. On the other hand, RNN-MUS captures the performance of ILP-MUS, especially with respect to the fairness metric. The margin of difference between the graphs ILP-MUS and RNN-MUS is not more than 0.0385 Mbps, 0.696%, and 0.0084 concerning the metrics of average throughput, admitted users, and fairness, respectively.

Fig. 5.2d shows the reduction of execution time for the RNN models with respect to the ILP problems. The simulation is done in a MATLAB environment running on an Octa-core intel CPU core i9-9880H. Compared to ILP-MTT, RNN-MTT reduces the execution time by more than 94% and can reach up to 99.49% of reduction. Similarly, RNN-MUS reduces the execution time by more than 94.7% and can reach up to 99.65% of reduction.

In conclusion, while the performance of RNN models, in comparison to the ILP models, slightly degrades with respect to the throughput, admitted users, and fairness metrics, the significant reduction of the execution time of RNN models makes them more practical for real-time implementation. The usage of the DL model becomes important when only partial information is available; for example, the processing time is not deterministic and can be variable. Also, decision-making could be done on a more decentralized level. In such a case, DL would be used to learn the hidden patterns and dependencies that are not available

5.3 Reinforcement Learning-Based Model for Computing Resource Allocation in Open-RAN

O-RAN enables driving down the costs of network deployments and allows the entry of new players into the RAN market. It enables network operators to maximize resource utilization and deliver new network edge services at a lower cost, resulting in higher profits for operators. Different Open-RAN components, such as the O-DU and O-CU components, can run as virtual machines in an Open-Cloud (O-Cloud), which is a cloud computing platform that is made up of physical infrastructure nodes [51]. The O-Cloud would be owned by an independent infrastructure provider (InP), and the mobile operator would be just one of the customers of this InP [15]. As the computing infrastructure in O-RAN is shared, mobile operators receive subscribers' payments and pay the infrastructure provider's costs. Operators must request computing resources to meet the demands of their users while maximizing their own profits. In this part, we consider the problem of maximizing the profits of an operator which has to pay for the computing resources provided by an infrastructure provider and allocate these resources to eMBB and URLLC users. These services have different deadline requirements. On the other hand, the operator uses the CPUs of the infrastructure provider. These CPUs are not fully available to the operator as they are shared with other customers (e.g., other operators). The operators can use these CPUs with two priority levels. The first priority level preempts ongoing processing and immediately serves the operator's prioritized frames. The other is based on best effort. The first priority level is suitable for URLLC users whose data may need to be prioritized to satisfy the deadline requirement. However, using the first priority level is more expensive.

5.3.1 Context and problem formulation

We consider a set $\mathcal{R}$ of Open Radio Units (O-RU). The set of users connected to an O-RU $r \in \mathcal{R}$ is denoted by $\mathcal{U}_r$; it combines the sets of both eMBB users $\mathcal{U}_r^E$ and URLLC users $\mathcal{U}_r^U$. For each O-RU, there exists two O-DUs (Open Distributed Unit) serving each service type (i.e., eMBB and URLLC), and each O-DU is connected to an O-CU (Open Central Unit). The O-DUs and O-CUs run as virtual machines in the Open Cloud (O-Cloud) and have to share the computing resources available at this O-Cloud, which is not owned by the operator. As each service type has different delay tolerance, the processing deadlines for eMBB users and URLLC users are expressed by d^E and d^U, respectively, where $d^E > d^U$.

Fig. 5.3 shows the system architecture. We consider a set of CPUs $\mathcal{C}$. These CPUs are owned by an infrastructure provider (InP), and the operator must share the available computing resources with different users (i.e., other operators). For each CPU, $k \in \{1, 2\}$ defines the priority level of a frame. Frames processed with priority $k = 1$ will force the CPU to preempt what it currently processes and process these frames

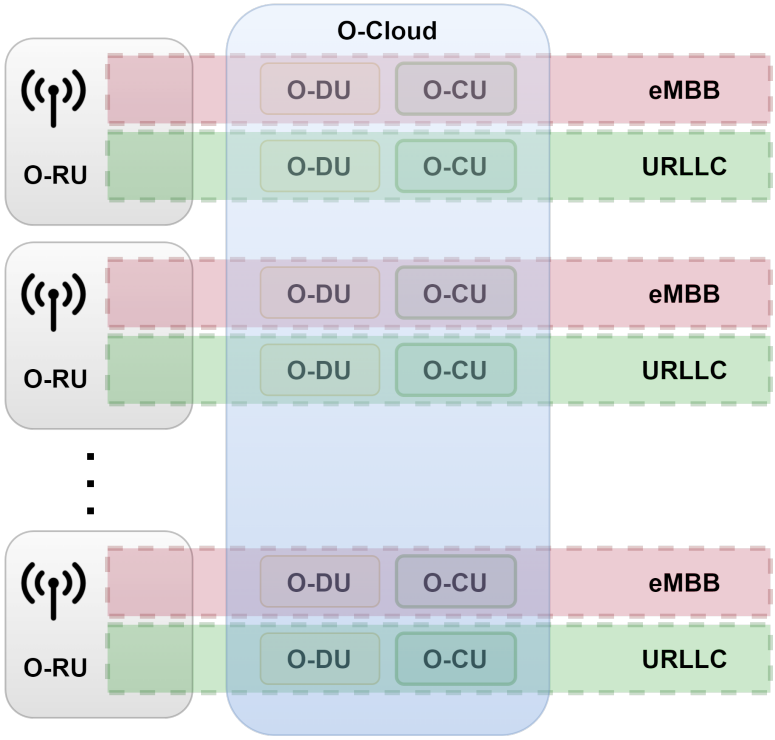


Figure 5.3: System Architecture

immediately. When $k = 2$, the CPU could process the frames after fully processing the ongoing tasks. For each CPU core $c \in \mathcal{C}$, $F^{c,2}$ is the amount of the total available computing resources for the operator over the next 2ms, which is the eMBB processing deadline ($F^{c,2} <= d^E$). $F^{c,1}$ is the portion of $F^{c,2}$ which could be immediately used for processing user frames without waiting for the current tasks to finish (i.e., priority $k = 1$). Each user pays the operator $p^{r,u}_{user}$ unit of money per bit. The operator pays for the InP when it processes the users' frames an amount equal to $p^{c,k}_{op}$ unit of money per second, depending on which CPU c and priority level k are used. Using priority $k = 1$ is more expensive, and the operator should only use it when it is necessary. We define $b^{r,u}$ as the number of bits carried by the frame of user $u \in \mathcal{U}_r$, while $e^{r,u}$ is the amount of time required to process the frame. We recall that both the number of bits transmitted in a frame during one TTI and the processing time of this frame depend on the used Modulation and Coding Scheme (MCS) and the number of RBs [6]. To maximize the operator's profits, we formulate the following MILP problem. The problem uses the binary variable $x^{r,u}_{c,k}$ which is equal to 1 if user $u \in \mathcal{U}_r$ is processed on CPU $c \in \mathcal{C}$ with priority k:

$$\text{maximize} \quad \sum_{r\in\mathcal{R}} \sum_{u\in\mathcal{U}_r} \sum_{c\in\mathcal{C}} \sum_{k\in\{1,2\}} x^{r,u}_{c,k}(b^{r,u}p^{r,u}_{user} - e^{r,u}p^{c,k}_{op}) \tag{5.1}$$

$$\text{subject to} \quad x^{r,u}_{c,k} \in \{0,1\}, \forall r \in \mathcal{R}, u \in \mathcal{U}_r, c \in \mathcal{C}, k \in \{1,2\} \tag{5.2}$$

$$\sum_{c\in\mathcal{C}}\sum_{k\in\{1,2\}} x_{c,k}^{r,u} \leq 1,\ \forall r\in\mathcal{R}, u\in\mathcal{U}_r, \tag{5.3}$$

$$\sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r}\sum_{k\in\{1,2\}} x_{c,k}^{r,u} e^{r,u} \leq F^{c,2}, \forall c\in\mathcal{C} \tag{5.4}$$

$$\sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r} x_{c,1}^{r,u} e^{r,u} \leq F^{c,1}, \forall c\in\mathcal{C} \tag{5.5}$$

$$\sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r^U} x_{c,1}^{r,u} e^{r,u} \leq min(F^{c,1}, d^U), \forall c\in\mathcal{C} \tag{5.6}$$

$$v_c \in \mathbb{R}_{\geq 0},\ \forall c\in\mathcal{C} \tag{5.7}$$

$$z_c \in \{0,1\},\ \forall c\in\mathcal{C} \tag{5.8}$$

$$\sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r^U} x_{c,2}^{r,u} e^{r,u} \leq v_c, c\in\mathcal{C} \tag{5.9}$$

$$v_c \geq F^{c,2} - (d^E - d^U) - \sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r} x_{c,1}^{r,u} e^{r,u}, \forall c\in\mathcal{C} \tag{5.10}$$

$$v_c <= M z_c, \forall c\in\mathcal{C} \tag{5.11}$$

$$v_c \leq F^{c,2} - (d^E - d^U) - \sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r} x_{c,1}^{r,u} e^{r,u} + M(1-z_c), \forall c\in\mathcal{C} \tag{5.12}$$

The objective (5.1) maximizes the total operator profit. (5.2) ensures the binary nature of the integer variable. (5.3) ensures that a frame is processed once. (5.4) ensures that a CPU is able to process all the assigned frames, while (5.5) sets the limit on the total frames processed with priority $k = 1$. The total time of processed URLLC frames with priority $k = 1$ can not exceed the minimum of the available CPU time and the URLLC deadline as (5.6) shows. On the other hand, URLLC frames can only be processed with priority $k = 2$ if there are enough resources before the deadline, as (5.9) mandates. Over the duration of 2 ms, which is the eMBB deadline, the difference between the eMBB and URLLC deadline indicates the amount of time that can only be used to process eMBB users. Hence, the only way to process URLLC frames with priority $k = 2$ on a CPU c is if the remaining available computing resource for priority $k = 2$ is higher than the difference in deadlines. Here, the remaining available computing resource is the difference between the initially total available $F^{c,2}$ minus what is admitted with prioity $k = 1$. In this case, the amount of available resources is tracked by the auxiliary continuous variable v_c defined in (5.7). However, this difference could be negative; hence v_c is equal to the value explained above if it's non-negative, and it is equal to zero otherwise. (5.7), (5.10), (5.11), (5.12) together enforce this condition. They use the auxiliary binary decision variable z_c defined in (5.8). M is the big-M notation [157]. M denotes a very large number. This MILP problem is NP-hard [18]; thus, we propose to solve it using reinforcement learning techniques.

5.3.2 Reinforcement learning model

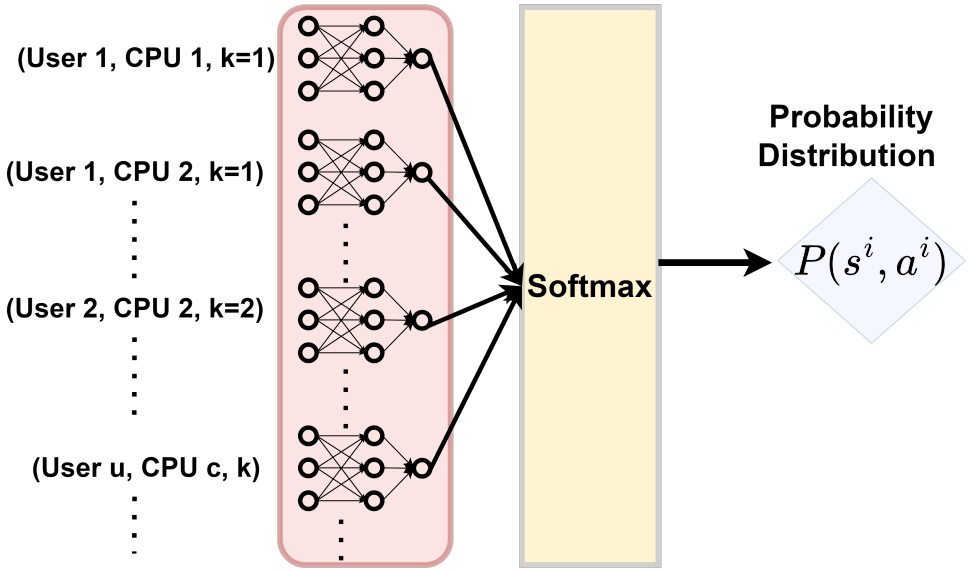


Figure 5.4: Neural Network Architecture

We consider a policy gradient-based RL model where each Transmission Time Interval (TTI) is an episode composed of steps. Resources are scheduled to users for a duration of one TTI. Each (user, CPU, priority) tuple is represented by $\delta_{r,u}^{i,c,k}$. In policy gradient algorithms, the agent learns the optimal probability distribution of actions that maximizes its rewards. We use a flexible Neural Network that scales with the number of users and CPUs. Each $\delta_{r,u}^{i,c,k}$ is fed into the same neural network and outputs a value. Hyperbolic tangent tanh is used as an activation function in the neural network, given it scales its values between -1 and 1, which could aid convergence. The outputted values from all $\delta_{r,u}^{i,c,k}$ are fed into a softmax function that outputs a probability distribution. The agent selects a tuple according to this probability distribution. At each step, a tuple is selected, assigning a user to a CPU and a priority level. It would have been possible to use a static NN architecture with a defined max number of users. However, the non-occupied tuples should be zero-padded. This will make the size of the NN huge and penalize the convergence time. As in [141] and [3], we use the dynamic architecture shown in Fig.5.4. The Markov Decision Process (MDP) model is defined as follows:

5.3.2.1 State

In each episode (i.e., TTI), one user will be selected in step i. The state includes the execution time $e^{r,u}$, the number of bits $b^{r,u}$, $service^{r,u}$ that indicates if the service is eMBB or URLLC, the payment of the user $b^{r,u}p_{user}^{r,u}$, the cost of processing $e^{r,u}p_{op}^{c,k}$,

and the profit of the operator (i.e., payment minus cost). Then:

$$\delta_{r,u}^{i,c,k} = \left\{ e^{r,u}, b^{r,u}, service^{r,u}, b^{r,u}p_{user}^{r,u}, e^{r,u}p_{op}^{c,k}, (b^{r,u}p_{user}^{r,u} - e^{r,u}p_{op}^{c,k}) \right\}$$

Given that $F^{c,k}$ is the available computing resources of CPU c at step i and that a^j is the action at step j, the state at step i of a TTI is defined as

$$s^i = \left\{ \delta_{r,u}^{i,c,k} : \forall r \in \mathcal{R}, u \in \mathcal{U}_r, c \in \mathcal{C}, k \in \{1,2\}, u \notin a^j, \forall j < i, e^{r,u} \leq F^{c,k} \right\}$$

Suppose that user $u \in \mathcal{U}_r$ has been selected. Given our adopted Neural Network architecture, as shown in the next part, removing $\delta_{r,u}^{i,c,k}$ from the state representation is necessary to ensure this user is no more re-selected. In each step, users have two priority choices and a set of CPUs to be assigned to. Once selected, the user can not be reselected and is omitted from the state and action space. Hence, the user and all the corresponding CPUs c and priorities k tuples are removed from the state representation. The state space dimensions decrease after making each selection. After each step, an already selected user should not be reselected; it should be removed from the action space. Moreover, the state only includes feasible choices; if there are no more resources to assign a user to a specific CPU and a priority, this tuple will be excluded from the state.

5.3.2.2 Action

At each step i of one TTI, the action a^i assigns a user $u \in U_r, r \in \mathcal{R}$ to CPU c and priority k, hence the action is the tuple:

$$a^i = (u, c, k), u \in \mathcal{U}_r, r \in \mathcal{R}, c \in \mathcal{C}, k \in \{1,2\}$$

5.3.2.3 Reward

The goal is to optimize the profit. The reward of action $a^i = (u, c, k)$ at step i being in state s^i is:

$$r^i(s^i, a^i) = \frac{2}{\pi} arctan(b^{r,u}p_{user}^{r,u} - e^{r,u}p_{op}^{c,k})$$

Using $arctan$ in the reward function allows us to finally scale the reward to be between -1 and 1, which aids convergence.

5.3.2.4 RL algorithm

The RL algorithm is based on REINFORCE algorithm with a baseline [163]. The weights θ of the Neural Network are initialized. For each TTI, the $\delta_{r,u}^{i,c,k}$ and state s^i should be initialized. Then the agent executes action a^i, gets r^i, and moves to s^{i+1}. This will be repeated until the terminal state is reached (i.e., no more resources or no users to be allocated). The weights θ should be updated using the following formula:

$$\theta \leftarrow \theta + \alpha v^i \nabla log(P(s^i, a^i))$$

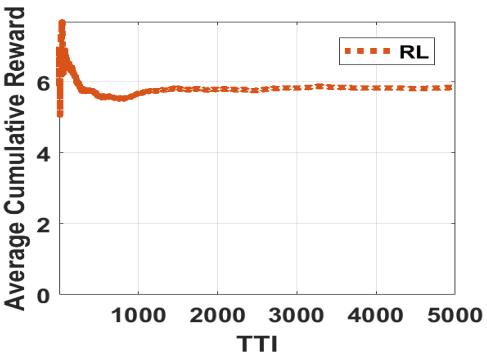
Algorithm 4: RL algorithm

1) initialize users $u \in \mathcal{U}_r$ from all O-RUs $r \in \mathcal{R}$: their MCS, RB, throughput, processing time, cost
2) Initialize CPUs available Time and Cost
3) Initialize weights θ of NN
while *training* **do**
  Initialize $i = 1$
  Initialize $\delta_{r,u}^{i,c,k}$ then s^i;
  while s^i *is not a terminal state* **do**
    execute a^i, get r^i, and s^{i+1}
    i=i+1
  end
  calculate v^i
  Apply the updates at the end of the episode: $\theta \leftarrow \theta + \alpha v^i \nabla log(P(s^i, a^i)))$
end

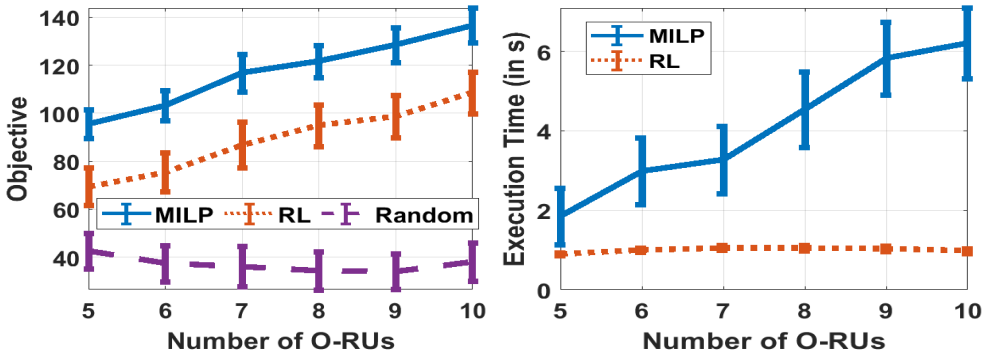
where α is the learning rate, $P(s^i, a^i)$ is the probability value yielded by the NN, and v^i is the discounted reward, normalized by subtracting the mean of the rewards in an episode and divided by the standard deviation of the rewards. The full algorithm is shown in In Algorithm.4

5.3.3 Simulation settings

To test our simulation, we consider a scenario where the number of O-RUs competing for computing resources in the O-Cloud varies from 5 to 10. In each O-RU, 90 RBs are reserved for eMBB users, while 10 RBs for URLLC users. The number of RBs of an eMBB user follows a uniform distribution between 20 and 40, while it is between 1 to 5 for a URLLC user. The eMBB and URLLC deadlines d^E and d^U are 2ms and 0.25 ms [3], respectively. As previously, we use the real traffic distribution in Figure 3.3b to sample the MCS index for each user. Users are resampled at each TTI. To calculate the number of transmitted bits, we use the 3GPP specifications [156], as done in the previous part of this chapter. The 3GPP specifications [156] provides the procedure to calculate the Transport Block Size (TBS), which is the number of bits transmitted in 1 TTI, as a function of the number of RBs and MCS. We also use the processing time model as a function of the MCS, the number of RBs, and the CPU frequency [81] which has been presented in Section 3.3. We set the CPU frequency to 2.6 GHz. The total available CPU time $F^{c,2}$ is a uniform random variable between 0.001 ms and 2 ms. The percentage of the amount of total computing resources $F^{c,2}$ available for prioritized processing is a uniform random variable between 0 and 100% (i.e., this percentage indicates how much $F^{c,1}$ is out of $F^{c,2}$, where $F^{c,1}$ is a partial amount of $F^{c,2}$). The cost of the non-prioritized processing per CPU per μs is a uniform random variable between 0.01 and 0.05 monetary units. The cost of prioritized processing, in which ongoing tasks are preempted to immediately schedule frames with priority $k = 1$, is uniformly random, between 1 to 3 times more than the cost of the non-prioritized one. A user pays a



(a) Average Cumulative Reward



(b) Objective value in monetary units

(c) Execution time of algorithms

Figure 5.5: Performance Figures of the RL algorithm

uniformly distributed value from 0.2 to 3.6 monetary units per Kilo bit. The reason for this extreme randomness is that we wanted to ensure the RL agent could adapt to diverse scenarios. For the RL agent, 8 neurons represent the input to the NN, which has 1 hidden layer with 10 neurons, and an output layer with 1 neuron. The learning rate for the agent is 0.01, while the rewards discount factor is 0.999. We use MATLAB for the simulation, the Deep Learning toolbox for training the RL agent, and CPLEX to solve the MILP problem.

5.3.4 Simulation results

Fig.5.5a shows the average cumulative reward. The RL agent converges after 1250 TTI. The agent is initially trained by varying the number of O-RUs, which varies the number of users and the total demand, from 5 to 10. This justifies why we initially have some fluctuations in the curve because the profit is affected by the number of existing users, strongly varying the total reward from one TTI to another. Fig. 5.5b compares the value of the objective function when the MILP solver is used versus when the RL model is used. Additionally, an algorithm named "Random" is used for benchmarking; it randomly allocates users to a CPU with random priority. Results show that the MILP can perfectly pick the user-CPU-priority tuples that maximize the operator's profit. The

RL model achieves at least 73% of the optimal MILP solution and reaches 80% when the number of O-RUs is 10. In comparison, the random allocation algorithm does not exceed 44% of the MILP solution and drops to 26% of the optimal solution. Fig. 5.5c shows the execution time of the MILP solver versus the RL algorithm. The RL can reduce the execution time by up to 85% compared with the MILP. We note that the study is done on a core i9-9880H, and the GPU used for RL is Nvidia Quadro P620. Given that scheduling decisions are made every TTI equal to 1ms, running the MILP problem in a real scenario is impossible. The RL model has been tested on the mentioned GPU, which is not optimized for Machine Learning computation. However, the RL model execution speed could be accelerated by using powerful GPUs that can parallelize the solution and output results in the required amount of time. Additionally, using MATLAB will add overhead, increasing the execution time.

5.3.5 Improvement perspectives

The MILP algorithm can perfectly pick the optimal solution at the cost of high complexity. In contrast, random allocation is a simple algorithm that curbs potential profits. The RL model presents a trade-off between lower complexity and optimality. However, there are a lot of potential paths for improvement. In addition to improving the hardware, as stated above, tweaking the Neural Network architecture and the activation function could have potential benefits in improving the performance of the RL model. Moreover, the MILP problem and the RL models assumed perfect processing time and cost knowledge. Even the infrastructure provider may refuse to provide detailed information regarding the availability of resources for operational and security reasons. If the information is missing, it is impossible to run the MILP problem. However, it would be possible to tune the state representation of the RL model to exclude the missing information. The RL would still learn the policy by receiving rewards from the environment. This is an important advantage for the RL versus the MILP problem. In addition, we have heavily randomized the cost models to account for diverse scenarios. The payment and cost model would be more deterministic in the real world. This would allow the RL model to converge faster and probably improves its performance, given that it will have less to learn.

5.4 Summary

In this chapter, we have investigated the usage of Recurrent Neural Networks for resource allocation in Open-RAN as an alternative to Integer Linear Programming. We considered two ILP problems to allocate computing resources to process users' data and to select their transmission MCS indexes. As solving the NP-hard ILP problems requires a lot of computing resources, ILP is a bad choice for real-time scheduling. The RNN models have demonstrated their ability to closely depict the performance of the ILP problems with a significantly lower execution time. Moreover, we have considered computing resource allocation for maximizing the operator's profit in Open-RAN. Open

RAN enables the virtualization of RAN components. Instead of owning infrastructure that is not always needed and would increase the CAPEX and OPEX, the operator can request computing resources from InPs when required. However, this mechanism must be profitable for the operator. We have formulated an MILP Problem that allocates computing resources to process users' frames to maximize the operator's profits. We have also proposed an RL model with the same objective but lower complexity. The simulation results highlight the ability of the RL model to score high profits while having lower complexity.

Chapter 6

Conclusion and Perspectives

Centralization and virtualization in future Radio Access Networks have brought various advantages that include minimizing power consumption and cutting down CAPEX and OPEX. Throughout this dissertation, we have studied different tasks that can benefit from centralization and joint decision-making in the novel RAN architectures, more precisely, in Cloud-RAN and Open-RAN architectures.

6.1 Conclusion

In Chapter 2, we discuss the architecture of Cloud RAN and Open RAN, and we surveyed different research papers that deal with radio and computing resource allocation, along with a few tools in game theory and machine learning used in mobile network research. Machine learning has a lot of potentials in mobile communications.

In Chapter 3, we examine the problem of allocating computing resources to users, when the BBU pool has insufficient resources to process the data of all users and we have considered a coordination scheme where the selection of the radio parameter MCS takes into account the availability of computing resources. The proposed coordination scheme is able to significantly reduce the wasted transmission power. Additionally, the proposed low-complexity heuristics are able to perform close to the optimal Integer Linear Programming Problems (ILP) solutions. Moreover, the heuristics maintain their benefits in multi-services scenarios where the processing of URLLC users is prioritized over eMBB users, though the improvements eventually drop.

In Chapter 4, we investigate the benefits of jointly allocating radio and computing resources in Cloud RAN. For this reason, we formulate a Mixed Integer Linear Programming Problem (MILP) that allocates Resource Blocks (RBs), MCS, power, and CPUs to process user frames, with the objectives of minimizing the total energy consumption and maximizing the total throughput. Considering the case where both radio and computing resources are sufficient to satisfy the requirements of users, the joint allocation is able to minimize energy consumption with respect to the sequential allocation where

radio resources are allocated independently of computing resources. However, the joint allocation is not better than the sequential when the goal is throughput maximization. Additionally, as radio resources become scarce, the joint and sequential allocation produce similar results given that the ultimate constraint of both objectives is to satisfy the QoS of users. In addition, the proposed low-complexity matching-based algorithm which uses the semi-dynamic preference lists, and the power-adjustment mechanism is able to perform close to the optimal MILP solution.

In Chapter 5, we focus on Open-RAN architecture, and we consider the problem of computing resource allocation in future RAN networks. First, we propose a Deep Learning (DL) model based on a Recurrent Neural Network (RNN) to solve the problem presented in Chapter 3. The model is able to learn how to output results close to the optimal ILP solver and with much lower complexity. Using machine learning models is useful in the case where some information such as the processing time or the traffic patterns are hidden. In contrast, the heuristics in Chapter 3 require full information and can not be used if some information are missing. Then we consider using Policy-Gradient based Reinforcement Learning (RL) to allocate computing resources to process users' data such that the operator's profit is maximized. The RL model is able to learn to perform close to the optimal MILP problem but with much lower complexity.

Each proposition brings novelties to the corresponding research area on radio and computing resource allocation in future RAN networks.

6.2 Perspectives

The RAN architecture continues to evolve towards fully softwarized radio access networks. The Open-RAN initiative is still in its early stages and many blocks in the architecture are under development and are expected to evolve in the future. For instance, while the current Open-RAN alliance has standarized the two RICs (i.e., the near-RT-RIC and the non-RT-RIC) the Real-Time RIC has not been standardized yet. Many of the promising applications require decision-making every 1ms or even less, and it is vital to support such a time scale [54]. Our role as researchers is to keep demonstrating the benefit of this small-scale control loop, and to propose different modifications and enhancement that enable the Real-Time RIC.

Besides, the open interfaces, multi-vendor interoperability, and virtualization in Open-RAN would increasingly encourage operators to explore the possibility of inter-operator sharing. With Open-RAN, it could be easier than ever for operators to share different resources such as the spectrum, RUs, CUs, DUs. Operators are interested in sharing only if this is beneficial for all the parties involved. Operators always target maximizing their profits and cutting down expenses. It is essential to define and explore the use cases where inter-operator sharing is beneficial and incentivized. In such cases, sharing mechanisms and algorithms should be devised while being compliant with the Open-RAN architecture.

In addition, the current Open-RAN architecture does not define mechanisms that

prioritize energy efficiency. Defining such mechanisms and highlighting how to embed energy efficiency in algorithms should be prioritized. In fact, the network components that come from different vendors could be less energy-efficient. It should be imperative to target energy efficiency in the pursuit of interoperability [15]. So far, it is still uncommon to find research works quantifying the energy efficiency in multi-vendor deployments in Open-RAN.

Furthermore, positioning the network components in Open-RAN is another challenge. As the O-DUs, O-CUs, and the RICs can be softwarized components, it is indispensable to optimize their placement in the different edge and regional clouds such that the QoS is guaranteed while the capital and operational costs are minimized. In addition, the advantages of using dynamic functional splits among the O-RUs, O-DUs, and O-CUs should be investigated, and feasbility studies should be made to decide whether such an option should be part of future Open-RAN releases.

Open-RAN has taken the initiative to support the use of machine learning in its architecture. However, a lot of work is yet to be completed, especially regarding small timescale ML models. On the other hand, different Deep Learning and Reinforcement Learning models have shown their abilities to solve different tasks in mobile networks. DL and RL keep evolving and novel architectures and models can be exploited. It remains important to continuously look for mobile network applications in which the evolving ML models are applicable. ML models that could be worthy of study include but are not limited to Graph Neural Networks, Transformers, and Federated Learning, among others.

References

Related Own References

International Journals with peer review

[1] M. Sharara, S. Hoteit, P. Brown, and V. Vèque, "On Coordinated Scheduling of Radio and Computing Resources in Cloud-RAN," *IEEE Transactions on Network and Service Management*, 2022.

International conferences with peer review

[2] M. Sharara, S. Hoteit, and V. Vèque, "Reinforcement Learning for Profit Maximization in Open-RAN," in *Accepted to 2023 IEEE/IFIP International Conference on Network Operations and Management Symposium (NOMS)*, 2023.

[3] M. Sharara, T. Pamuklu, S. Hoteit, V. Vèque, and M. Erol-Kantarci, "Policy-Gradient-Based Reinforcement Learning for Computing Resources Allocation in O-RAN," in *2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*, 2022.

[4] M. Sharara, S. Hoteit, V. Vèque, and F. Bassi, "Minimizing Power Consumption by Joint Radio and Computing Resource Allocation in Cloud-Ran," in *2022 IEEE Symposium on Computers and Communications (ISCC)*, 2022.

[5] M. Sharara, S. Hoteit, and V. Vèque, "A Recurrent Neural Network Based Approach for Coordinating Radio and Computing Resources Allocation in Cloud-RAN," in *2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*.

[6] M. Sharara, S. Hoteit, P. Brown, and V. Vèque, "Coordination between Radio and Computing Schedulers in Cloud-RAN," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2021.

[7] M. Sharara, S. Hoteit, P. Brown, and V. Vèque, "Coordination de l'ordonnancement radio et de calcul dans Cloud-RAN," in *ALGOTEL 2021 - 23èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, La Rochelle, France, 2021.

Submitted

[8] M. Sharara, F. Fossati, S. Hoteit, V. Vèque, and F. Bassi, "Minimizing Energy Consumption by Joint Radio and Computing Resource Allocation in Cloud-Ran," *Submitted to the Journal of Computer Networks*,

Other References

[9] "Ericsson Mobility Report." (2022).

[10] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions," *IEEE Open Journal of the Communications Society*, vol. 1, 2020.

[11] 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, 2019, issn: 2169-3536.

[12] J. Lee and I. Moon, "Research on Virtual Network for Virtual Mobile Network," in *2010 Second International Conference on Computer and Network Technology*, 2010.

[13] H. Khedher, S. Hoteit, P. Brown, V. Véque, R. Krishnaswamy, W. Diego, and M. Hadji, "Real Traffic-Aware Scheduling of Computing Resources in Cloud-RAN," in *2020 International Conference on Computing, Networking and Communications (ICNC)*, 2020.

[14] C. Mobile, "C-RAN: the road towards green RAN," *White Paper, ver*, vol. 2, 2011.

[15] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," *CoRR*, vol. abs/2202.01032, 2022. arXiv: 2202.01032.

[16] S. Husain, A. Kunz, A. Prasad, E. Pateromichelakis, K. Samdanis, and J. Song, "The Road to 5G V2X: Ultra-High Reliable Communications," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2018.

[17] F. Salah, L. Kuru, and R. Jäntti, "Reliability and Availability Enhancements of the 5G Connectivity for Factory Automation," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1, 2019.

[18] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 5th. Springer Publishing Company, Incorporated, 2012, isbn: 3642244874.

[19] G. Bacci, S. Lasaulce, W. Saad, and L. Sanguinetti, "Game Theory for Networks: A Tutorial on Game-Theoretic Tools for Emerging Signal Processing Applications," *IEEE Signal Processing Magazine*, vol. 33, no. 1, 2016.

[20] A. Ly and Y. Yao, "A Review of Deep Learning in 5G Research: Channel Coding, Massive MIMO, Multiple Access, Resource Allocation, and Network Security," *IEEE Open Journal of the Communications Society*, vol. 2, 2021.

[21] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, 2019.

[22] S. Parkvall, E. Dahlman, A. Furuskar, and M. Frenne, "NR: The New 5G Radio Access Technology," *IEEE Communications Standards Magazine*, vol. 1, no. 4, 2017.

[23] S. Khatibi, K. Shah, and M. Roshdi, "Modelling of Computational Resources for 5G RAN," in *2018 European Conference on Networks and Communications (EuCNC)*, 2018.

[24] A. Kukushkin, "Global System Mobile, GSM, 2G," in *Introduction to Mobile Network Engineering: GSM, 3G-WCDMA, LTE and the Road to 5G*. 2018.

[25] F. Hillebrand, *GSM and UMTS: The Creation of Global Mobile Communication*. Hoboken, NJ: Wiley, Oct. 2001.

[26] Y. Chen, A. Bayesteh, Y. Wu, B. Ren, S. Kang, S. Sun, Q. Xiong, C. Qian, B. Yu, Z. Ding, S. Wang, S. Han, X. Hou, H. Lin, R. Visoz, and R. Razavi, "Toward the Standardization of Non-Orthogonal Multiple Access for Next Generation Wireless Networks," *IEEE Communications Magazine*, vol. 56, no. 3, 2018.

[27] M. Yassin, M. A. AboulHassan, S. Lahoud, M. Ibrahim, D. Mezher, B. Cousin, and E. A. Sourour, "Survey of ICIC Techniques in LTE Networks Under Various Mobile Environment Parameters," *Wireless Networks*, vol. 23, no. 2, Dec. 2015.

[28] G. Heine and H. Sagkob, *GPRS: Gateway to Third-Generation Mobile Networks*. Artech House, 2003.

[29] M. Sauter, *From GSM to LTE-Advanced: An Introduction to Mobile Networks and Mobile Broadband*. Wiley, 2014.

[30] I. A. Alimi, A. L. Teixeira, and P. P. Monteiro, "Toward an Efficient C-RAN Optical Fronthaul for the Future Networks: A Tutorial on Technologies, Requirements, Challenges, and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, 2018.

[31] H. Holma and A. Toskala, *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*, 1st. USA: John Wiley & Sons, 2000, isbn: 0471720518.

[32] H. Holma and A. Toskala, *WCDMA for UMTS: HSPA Evolution and LTE*. USA: John Wiley & Sons, 2007, isbn: 047031933X.

[33] 3GPP, *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); TS 36 300, 3GPP, Release 9*. 2006.

[34] Alcatel, "The LTE Network Architecture: A Comprehensive Tutorial," Sep. 2018.

[35] P. E. Mogensen, T. Koivisto, K. I. Pedersen, I. Z. Kovacs, B. Raaf, K. Pajukoski, and M. J. Rinne, "LTE-Advanced: The path towards gigabit/s in wireless mobile communications," in *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, 2009.

[36] O. O. Erunkulu, A. M. Zungeru, C. K. Lebekwe, M. Mosalaosi, and J. M. Chuma, "5G Mobile Communication Applications: A Survey and Comparison of Use Cases," *IEEE Access*, vol. 9, 2021.

[37] *NG-RAN; Architecture Description (Release 16), 3GPP TS 38.401, V16.1.0.*

[38] A. A. Zaidi, R. Baldemair, V. Moles-Cases, N. He, K. Werner, and A. Cedergren, "OFDM Numerology Design for 5G New Radio to Support IoT, eMBB, and MBSFN," *IEEE Communications Standards Magazine*, vol. 2, no. 2, 2018.

[39] E. Dahlman, J. Skold, and P. Beming, *3G Evolution: HSPA LTE for Mobile Broadband*, 2nd ed. Academic Press, Jul. 2010.

[40] *The Path to 5G Requires a Strong Optical Network: From C-RAN to Cloud-RAN*. Expo, Houston, TX, USA, 2017.

[41] Y. Lin, L. Shao, Z. Zhu, Q. Wang, and R. K. Sabhikhi, "Wireless network cloud: Architecture and system requirements," *IBM Journal of Research and Development*, vol. 54, no. 1, 2010.

[42] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, "Cloud radio access network (C-RAN): a primer," *IEEE Network*, vol. 29, no. 1, 2015.

[43] C. I, C. Rowell, S. Han, Z. Xu, G. Li, and Z. Pan, "Toward Green and Soft: a 5G Perspective," *IEEE Communications Magazine*, vol. 52, no. 2, 2014.

[44] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for Mobile Networks A Technology Overview," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, 2015.

[45] *Cloud RAN Architecture for 5G-White Paper*. Ericsson, Telefonica, Stockholm, Sweden, White Paper, 2017.

[46] D. Naboulsi, A. Mermouri, R. Stanica, H. Rivano, and M. Fiore, “On User Mobility in Dynamic Cloud Radio Access Networks,” in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018.

[47] L. Liu, F. Yang, R. Wang, Z. Shi, A. Stidwell, and D. Gu, “Analysis of Handover Performance Improvement in Cloud-RAN Architecture,” in *7th International Conference on Communications and Networking in China*, 2012.

[48] L. Bonati, S. D’Oro, M. Polese, S. Basagni, and T. Melodia, “Intelligence and Learning in O-RAN for Data-Driven NextG Cellular Networks,” *IEEE Communications Magazine*, vol. 59, no. 10, 2021.

[49] A. K. U and G. Gundu Hallur, “Economic and Technical Implications of Implementation of OpenRAN by "RAKUTEN MOBILE",” in *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, 2022.

[50] O-RAN-WG1, “Architecture Description,” 2021.

[51] O-RAN-WG6, “Cloud Architecture and Deployment Scenarios for O-RAN Virtualized RAN,” 2019.

[52] O-RAN-WG2, “Non-RT RIC: Architecture,” 2021.

[53] O-RAN-WG3, “Near-RT RIC Architecture,” 2020.

[54] S. D’Oro, M. Polese, L. Bonati, H. Cheng, and T. Melodia, *dApps: Distributed Applications for Real-time Inference and Control in O-RAN*, 2022.

[55] A. S. Abdalla, P. S. Upadhyaya, V. K. Shah, and V. Marojevic, “Toward Next Generation Open Radio Access Network-What O-RAN Can and Cannot Do!” *CoRR*, vol. abs/2111.13754, 2021. arXiv: 2111.13754.

[56] O-RAN-WG2, “AI/ML workflow description and requirements-v01.03,” 2020.

[57] O-RAN-WG1, “O-RAN Use Cases Detailed Specification v06.00,” 2021.

[58] O-RAN-WG1, “O-RAN Use Cases Analysis Report v6.0,” 2021.

[59] S. Lien, D. Deng, and B. Chang, “Session Management for URLLC in 5G Open Radio Access Network: A Machine Learning Approach,” in *2021 International Wireless Communications and Mobile Computing (IWCMC)*, 2021.

[60] A. Filali, B. Nour, S. Cherkaoui, and A. Kobbane, *Communication and Computation O-RAN Resource Slicing for URLLC Services Using Deep Reinforcement Learning*, 2022.

[61] L. Baldesi, F. Restuccia, and T. Melodia, “ChARM: NextG Spectrum Sharing Through Data-Driven Real-Time O-RAN Dynamic Control,” in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022.

[62] O-RAN-WG1, “O-RAN Use Cases Analysis Report v9.0,” 2022.

[63] X. Liu, L. Xie, Y. Wang, J. Zou, J. Xiong, Z. Ying, and A. V. Vasilakos, “Privacy and Security Issues in Deep Learning: A Survey,” *IEEE Access*, vol. 9, 2021.

[64] I. Al-Samman, M. Artuso, H. Christiansen, A. Doufexi, and M. Beach, "A framework for Resources Allocation in Virtualised C-RAN," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, IEEE, 2016, isbn: 9781509032549.

[65] Y. Jia, H. Tian, S. Fan, P. Zhao, and K. Zhao, "Bankruptcy Game based Resource Allocation Algorithm for 5G Cloud-RAN Slicing," *IEEE Wireless Communications and Networking Conference, WCNC*, vol. 2018-April, 2018, issn: 15253511.

[66] D. Maaz, A. Galindo-Serrano, and S. E. Elayoubi, "URLLC User Plane Latency Performance in New Radio," in *2018 25th International Conference on Telecommunications (ICT)*, 2018.

[67] S. Costanzo, I. Fajjari, N. Aitsaadi, and R. Langar, "A Network Slicing Prototype for a Flexible Cloud Radio Access Network," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2018.

[68] S. Costanzo, I. Fajjari, N. Aitsaadi, and R. Langar, "Dynamic Network Slicing for 5G IoT and eMBB services: A New Design with Prototype and Implementation Results," in *2018 3rd Cloudification of the Internet of Things (CIoT)*, 2018.

[69] B. Zhang, X. Mao, J. Yu, and Z. Han, "Resource Allocation for 5G Heterogeneous Cloud Radio Access Networks With D2D Communication: A Matching and Coalition Approach," *IEEE Transactions on Vehicular Technology*, 2018.

[70] Q. Liu, S. Sun, and H. Gao, "Joint User-Centric Clustering and Frequency Allocation in Ultra-Dense C-RAN," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020.

[71] X. Cao, M. Peng, and Z. Ding, "A Game-Theoretic Approach of Resource Allocation in NOMA-Based Fog Radio Access Networks," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019.

[72] C. Han, W. Wang, Y. Wang, and Z. Zhang, "Computational resource constrained multi-cell joint processing in cloud radio access networks," in *IEEE International Conference on Communications*, IEEE, 2017.

[73] H. Taleb, M. El Helou, K. Khawam, S. Lahoud, and S. Martin, "Centralized and distributed RRH clustering in Cloud Radio Access Networks," in *Proceedings - IEEE Symposium on Computers and Communications*, 2017, isbn: 9781538616291.

[74] K. Boulos, K. Khawam, M. El Helou, M. Ibrahim, H. Sawaya, and S. Martin, "An Efficient Scheme for BBU-RRH Association in C-RAN Architecture for Joint Power Saving and Re-Association Optimization," in *Proceedings of the 2018 IEEE 7th International Conference on Cloud Networking, CloudNet 2018*, IEEE, 2018.

[75] S.-C. Zhan and D. Niyato, "A Coalition Formation Game for Remote Radio Head Cooperation in Cloud Radio Access Network," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, 2017.

[76] H. Zhang, H. Zhou, and M. Erol-Kantarci, "Team learning-based resource allocation for open radio access network (o-ran)," in *ICC 2022 - IEEE International Conference on Communications*, 2022.

[77] F. Kavehmadavani, V.-D. Nguyen, T. X. Vu, and S. Chatzinotas, "Traffic steering for embb and urllc coexistence in open radio access networks," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2022.

[78] S. Bian, J. Song, M. Sheng, Z. Shao, J. He, Y. Zhang, Y. Li, and I. Chih-Lin, "Sum-rate maximization in OFDMA downlink systems: A joint subchannels, power, and MCS allocation approach," in *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*.

[79] T. X. Tran, A. Younis, and D. Pompili, "Understanding the Computational Requirements of Virtualized Baseband Units Using a Programmable Cloud Radio Access Network Testbed," in *2017 IEEE International Conference on Autonomic Computing (ICAC)*, 2017.

[80] H. Khedher, S. Hoteit, P. Brown, R. Krishnaswamy, W. Diego, and V. Veque, "Processing Time Evaluation and Prediction in Cloud-RAN," in *IEEE International Conference on Communications (ICC)*, May 2019.

[81] S. Khatibi, K. Shah, and M. Roshdi, "Modelling of Computational Resources for 5G RAN," in *2018 European Conference on Networks and Communications (EuCNC)*.

[82] M. A. Alam and S. Khatibi, "CPU Resource Usage Analysis for Downlink PDCP Processing in CRAN," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020.

[83] A. Okic and A. E. C. Redondi, "Optimal Resource Allocation in C-RAN through DSP Computational Load Forecasting," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020.

[84] M. Y. Lyazidi, L. Giupponi, J. Mangues-Bafalluy, N. Aitsaadi, and R. Langar, "A Novel Optimization Framework for C-RAN BBU Selection Based on Resiliency and Price," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017.

[85] N. Salhab, R. Rahim, and R. Langar, "Optimization of Virtualization Cost, Processing Power and Network Load of 5G Software-Defined Data Centers," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, 2020.

[86] V. Q. Rodriguez and F. Guillemin, "Towards the deployment of a fully centralized Cloud-RAN architecture," in *13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017.

[87] S. Ramakrishnan, S. Kar, and S. Dharmaraja, "Analysis of mid-haul characteristics for LTE-nr multi-connectivity in heterogeneous cloud RAN," in *25th National Conference on Communications, NCC 2019*, IEEE, 2019, isbn: 9781538692868.

[88] N. Kazemifard and V. Shah-Mansouri, "Minimum delay function placement and resource allocation for Open RAN (O-RAN) 5G networks," *Computer Networks*, vol. 188, 2021, issn: 1389-1286.

[89] M. Barahman, L. M. Correia, and L. S. Ferreira, "An Efficient QoS-Aware Computational Resource Allocation Scheme in C-RAN," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020.

[90] M. Vincenzi, A. Antonopoulos, E. Kartsakli, J. Vardakas, L. Alonso, and C. Verikoukis, "Cooperation incentives for multi-operator C-RAN energy efficient sharing," in *IEEE International Conference on Communications*, IEEE, 2017, isbn: 9781467389990.

[91] Y. Gu, Z. Chang, M. Pan, L. Song, and Z. Han, "Joint Radio and Computational Resource Allocation in IoT Fog Computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, 2018.

[92] S. Matoussi, I. Fajjari, N. Aitsaadi, R. Langar, and S. Costanzo, "Joint Functional Split and Resource Allocation in 5G Cloud-RAN," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019.

[93] Y. Liao, L. Song, Y. Li, and Y. A. Zhang, "Radio resource management for cloud-RAN networks with computing capability constraints," in *2016 IEEE International Conference on Communications (ICC)*, 2016.

[94] S. Matoussi, I. Fajjari, N. Aitsaadi, and R. Langar, "User Slicing Scheme with Functional Split Selection in 5G Cloud-RAN," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020.

[95] K. Wang, W. Zhou, and S. Mao, "On Joint BBU/RRH Resource Allocation in Heterogeneous Cloud-RANs," *IEEE Internet of Things Journal*, vol. 4, no. 3, 2017.

[96] M. Y. Lyazidi, N. Aitsaadi, and R. Langar, "Dynamic resource allocation for Cloud-RAN in LTE with real-time BBU/RRH assignment," in *IEEE International Conference on Communications (ICC)*, May 2016.

[97] M. Y. Lyazidi, N. Aitsaadi, and R. Langar, "A Dynamic Resource Allocation Framework in LTE Downlink for Cloud-Radio Access Network," *Computer Networks*, vol. 140, 2018, issn: 1389-1286.

[98] A. Younis, T. X. Tran, and D. Pompili, "Bandwidth and Energy-Aware Resource Allocation for Cloud Radio Access Networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, 2018.

[99] E. Aqeeli, A. Moubayed, and A. Shami, "Power-Aware Optimized RRH to BBU Allocation in C-RAN," *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, 2018.

[100] M. M. Abdelhakam, M. M. Elmesalawy, M. K. Elhattab, and H. H. Esmat, "Energy-Efficient BBU Pool Virtualisation for C-RAN with Quality of Service Guarantees," *IET Communications*, vol. 14, no. 1, 2020.

[101] L. Ferdouse, A. Anpalagan, and S. Erkucuk, "Joint Communication and Computing Resource Allocation in 5G Cloud Radio Access Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, Sep. 2019, issn: 1939-9359.

[102] M. M. Abdelhakam and M. M. Elmesalawy, "Joint Beamforming Design and BBU Computational Resources Allocation in Heterogeneous C-RAN with QoS Guarantee," in *2018 International Symposium on Networks, Computers and Communications (ISNCC)*.

[103] D. Bega, A. Banchs, M. Gramaglia, X. Costa-Pérez, and P. Rost, "CARES: Computation-Aware Scheduling in Virtualized Radio Access Networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, 2018.

[104] M. K. Motalleb, V. Shah-Mansouri, S. Parsaeefard, and O. L. A. López, "Resource allocation in an open ran system using network slicing," *IEEE Transactions on Network and Service Management*, 2022.

[105] Q. Liu, T. Han, and N. Ansari, "Joint Radio and Computation Resource Management for Low Latency Mobile Edge Computing," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018.

[106] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for mobile edge computing," in *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2018.

[107] C. Xu, G. Zheng, and L. Tang, "Energy-aware user association for NOMA-based mobile edge computing using matching-coalition game," *IEEE Access*, vol. 8, 2020, issn: 21693536.

[108] S.-C. Zhan and D. Niyato, "A Coalition Formation Game for Remote Radio Head Cooperation in Cloud Radio Access Network," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, 2017.

[109] M. Hajir, R. Langar, and F. Gagnon, "Solidarity-Based Cooperative Games for Resource Allocation with Macro-Users Protection in HetNets," in *2016 IEEE International Conference on Communications (ICC)*, 2016.

[110] M. Hajir, R. Langar, and F. Gagnon, "Coalitional Games for Joint Co-Tier and Cross-Tier Cooperative Spectrum Sharing in Dense Heterogeneous Networks," *IEEE Access*, vol. 4, 2016.

[111] J. Cao, T. Peng, Z. Qi, R. Duan, Y. Yuan, and W. Wang, "Interference Management in Ultradense Networks: A User-Centric Coalition Formation Game Approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, 2018.

[112] Y. Sun, M. Peng, and H. V. Poor, "A Distributed Approach to Improving Spectral Efficiency in Uplink Device-to-Device-Enabled Cloud Radio Access Networks," *IEEE Transactions on Communications*, vol. 66, no. 12, 2018.

[113] M. Barahman, L. M. Correia, and L. S. Ferreira, "A Fair Computational Resource Management Strategy in C-RAN," in *2018 International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom)*, 2018.

[114] Z. Han, Y. Gu, and W. Saad, *Matching Theory for Wireless Networks*. Springer, 2017, isbn: 978-3-319-56251-3.

[115] Z. Han, D. Niyato, W. Saad, and T. Başar, *Game Theory for Next Generation Wireless and Communication Networks*. Cambridge University Press, 2019.

[116] Y. Gu, C. Jiang, L. X. Cai, M. Pan, L. Song, and Z. Han, "Dynamic Path to Stability in LTE-Unlicensed with User Mobility: A Matching Framework," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, Jul. 2017, issn: 15361276.

[117] K. Hamidouche, W. Saad, and M. Debbah, "Many-to-many matching games for proactive social-caching in wireless small cell networks," in *2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, WiOpt 2014*, IEEE Computer Society, 2014.

[118] S. Zhang, B. Di, L. Song, and Y. Li, "Radio resource allocation for non orthogonal multiple access (NOMA) relay network using matching game," in *2016 IEEE International Conference on Communications, ICC 2016*, Institute of Electrical and Electronics Engineers Inc., Jul. 2016, isbn: 9781479966646.

[119] B. Di, S. Bayat, L. Song, Y. Li, and Z. Han, "Joint User Pairing, Subchannel, and Power Allocation in Full-Duplex Multi-User OFDMA Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, 2016.

[120] E. Bjornson and P. Giselsson, "Two Applications of Deep Learning in the Physical Layer of Communication Systems [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 37, no. 5, 2020.

[121] M. Lee, G. Yu, and G. Y. Li, "Accelerating Resource Allocation for D2D Communications Using Imitation Learning," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019.

[122] S. Jaffry and S. F. Hasan, "Cellular Traffic Prediction using Recurrent Neural Networks," in *2020 IEEE 5th International Symposium on Telecommunication Technologies (ISTT)*, 2020.

[123] M. S. Hossain and G. Muhammad, "A Deep-Tree-Model-Based Radio Resource Distribution for 5G Networks," *IEEE Wireless Communications*, vol. 27, no. 1, 2020.

[124] N. Salhab, R. Rahim, R. Langar, and R. Boutaba, "Deep Neural Networks Approach for Power Head-Room Predictions in 5G Networks and Beyond," in *2020 IFIP Networking Conference (Networking)*, 2020.

[125] C. Lee, H. Cho, S. Song, and J.-M. Chung, "Prediction-Based Conditional Handover for 5G mm-Wave Networks: A Deep-Learning Approach," *IEEE Vehicular Technology Magazine*, vol. 15, no. 1, 2020.

[126] H. Ye, G. Y. Li, and B.-H. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, 2018.

[127] B. Brik, K. Boutiba, and A. Ksentini, "Deep Learning for B5G Open Radio Access Network: Evolution, Survey, Case Studies, and Challenges," *IEEE Open Journal of the Communications Society*, vol. 3, 2022.

[128] N. Salhab, R. Langar, R. Rahim, S. Cherrier, and A. Outtagarts, "Autonomous Network Slicing Prototype Using Machine-Learning-Based Forecasting for Radio Resources," *IEEE Communications Magazine*, vol. 59, no. 6, 2021.

[129] N. Salhab, R. Langar, and R. Rahim, "5G Network Slices Resource Orchestration Using Machine Learning Techniques," *Computer Networks*, vol. 188, 2021, issn: 1389-1286.

[130] N. Salhab, R. Rahim, R. Langar, and R. Boutaba, "Machine Learning Based Resource Orchestration for 5G Network Slices," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019.

[131] S. Matoussi, I. Fajjari, N. Aitsaadi, and R. Langar, "Deep Learning based User Slice Allocation in 5G Radio Access Networks," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, 2020.

[132] M. Elsayed and M. Erol-Kantarci, "Reinforcement Learning Based Joint Power and Resource Allocation for URLLC in 5G," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019.

[133] M. Elsayed and M. Erol-Kantarci, "AI-Enabled Radio Resource Allocation in 5G for URLLC and eMBB Users," in *2019 IEEE 2nd 5G World Forum (5GWF)*, 2019.

[134] M. Elsayed and M. Erol-Kantarci, "Learning-Based Resource Allocation for Data-Intensive and Immersive Tactile Applications," in *2018 IEEE 5G World Forum (5GWF)*, 2018.

[135] M. Elsayed, K. Shimotakahara, and M. Erol-Kantarci, "Machine Learning-based Inter-Beam Inter-Cell Interference Mitigation in mmWave," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020.

[136] M. Elsayed and M. Erol-Kantarci, "Radio Resource and Beam Management in 5G mmWave Using Clustering and Deep Reinforcement Learning," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020.

[137] M. Elsayed, M. Erol-Kantarci, and H. Yanikomeroglu, "Transfer Reinforcement Learning for 5G New Radio mmWave Networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, 2021.

[138] M. Elsayed and M. Erol-Kantarci, "Deep Reinforcement Learning for Reducing Latency in Mission Critical Services," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018.

[139] M. Elsayed and M. Erol-Kantarci, "Deep Q-Learning for Low-Latency Tactile Applications: Microgrid Communications," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2018.

[140] T. Pamuklu, M. Erol-Kantarci, and C. Ersoy, "Reinforcement Learning Based Dynamic Function Splitting in Disaggregated Green Open RANs," in *ICC 2021 - IEEE International Conference on Communications*, 2021.

[141] J. S. Shekhawat, R. Agrawal, K. G. Shenoy, and R. Shashidhara, "A Reinforcement Learning Framework for QoS-Driven Radio Resource Scheduler," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020.

[142] Y. Shi, Y. E. Sagduyu, and T. Erpek, "Reinforcement Learning for Dynamic Resource Optimization in 5G Radio Access Network Slicing," in *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2020.

[143] Y. Cao, S.-Y. Lien, Y.-C. Liang, and K.-C. Chen, "Federated Deep Reinforcement Learning for User Access Control in Open Radio Access Networks," in *ICC 2021 - IEEE International Conference on Communications*, 2021.

[144] Y. Li, H. Xia, J. Shi, and S. Wu, "Joint optimization of computing and radio resource for cooperative transmission in C-RAN," in *IEEE/CIC International Conference on Communications in China (ICCC)*, 2017.

[145] A. Nauman, T. N. Nguyen, Y. A. Qadri, Z. Nain, K. Cengiz, and S. W. Kim, "Artificial Intelligence in Beyond 5G and 6G Reliable Communications," *IEEE Internet of Things Magazine*, vol. 5, no. 1, 2022.

[146] Y. Shi, Y. E. Sagduyu, and T. Erpek, "Reinforcement Learning for Dynamic Resource Optimization in 5G Radio Access Network Slicing," in *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2020.

[147] J. Mei, X. Wang, K. Zheng, G. Boudreau, A. B. Sediq, and H. Abou-Zeid, "Intelligent Radio Access Network Slicing for Service Provisioning in 6G: A Hierarchical Deep Reinforcement Learning Approach," *IEEE Transactions on Communications*, vol. 69, no. 9, 2021.

[148] F. W. Murti, S. Ali, and M. Latva-aho, "Deep Reinforcement Based Optimization of Function Splitting in Virtualized Radio Access Networks," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021.

[149] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View," *IEEE Access*, vol. 6, 2018.

[150] 3GPP. "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures 3GPP TS 36.213 v.12.3.0 Rel.12." (2014).

[151] F. Bassi and H. I. Khedher, "HARQ-aware allocation of computing resources in C-RAN," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, 2020.

[152] H. D. Trinh, N. Bui, J. Widmer, L. Giupponi, and P. Dini, "Analysis and modeling of mobile traffic using real traces," in *International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017.

[153] R. Jain, D. Chiu, and W. Hawe, *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems*. DEC Research Report TR-301, Sep. 1984.

[154] 3GPP. "ETSI TS 136 213 V12.3.0, 3GPP, "Technical specification group services and system aspects; release 15 description"." (2019).

[155] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-Reliable and Low-Latency Communications in 5G Downlink: Physical Layer Aspects," *IEEE Wireless Communications*, vol. 25, no. 3, 2018.

[156] "5G; NR; Physical layer procedures for data, ETSI TS 138 214 V15.3.0." (Oct. 2018).

[157] D. G. Luenberger and Y. Ye, *Linear and NonLinear Programming*, 3rd ed. New York, NY: Springer, 2008.

[158] K. Bando, "Many-to-one matching markets with externalities among firms," *Journal of Mathematical Economics*, vol. 48, no. 1, 2012.

[159] J. Fan, Q. Yin, G. Y. Li, B. Peng, and X. Zhu, “MCS Selection for Throughput Improvement in Downlink LTE Systems,” in *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*.

[160] “5G; NR; User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone.” (Jul. 2018).

[161] S. Sun, T. S. Rappaport, S. Rangan, T. A. Thomas, A. Ghosh, I. Z. Kovacs, I. Rodriguez, O. Koymen, A. Partyka, and J. Jarvelainen, “Propagation Path Loss Models for 5G Urban Micro- and Macro-Cellular Scenarios,” in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*.

[162] R. Dhumal Deshmukh and A. Kiwelekar, “Deep Learning Techniques for Part of Speech Tagging by Natural Language Processing,” in *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2020.

[163] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018, isbn: 0262039249.

[164] C. Zhang, P. Patras, and H. Haddadi, “Deep Learning in Mobile and Wireless Networking: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, 2019.

[165] O. Obulesu, M. Mahendra, and M. ThrilokReddy, “Machine Learning Techniques and Tools: A Survey,” in *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2018.

[166] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, “A Brief Survey of Machine Learning Methods and Their Sensor and IoT Applications,” in *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*, 2017.

Appendices

Appendix A

Neural Networks and Deep Learning

Machine learning is classified into three categories: Supervised learning, unsupervised learning, and reinforcement learning. In this appendix, we focus on supervised learning, where we briefly discuss some basic concepts of neural networks and deep learning.

A.1 Introduction

In mathematics, a function $f(x)$ receives an input x and yields an output $y = f(x)$. The input and the function are known, and the output is the missing part but can be determined from the function. In contrast, in a supervised learning task, both the input and the output are known. However, the function that produces the output given the input is unknown. A learning task aims to learn this unknown function using known inputs and outputs. Then, it uses this function to predict unknown outputs. The acquired knowledge is only based on information seen during training.

The basic supervised learning tasks are classified into two branches:

- Regression: A model is called a regression model if it takes an input vector x and outputs a continuous output. For example, predicting the weather, the rainfall, or the stock values is a regression task because the output can be any value in a continuous interval. Some methods used for regression tasks are linear regression, regression trees, non-linear Regression, bayesian linear regression, and polynomial regression [164].

- Classification: A model is a classification model if it outputs discrete values. These values represent categories, such as predicting if a person is a male or a female, a cat or a dog, true or false, etc. In a classification model, the output is a probability distribution used to assign the most likelihood category. This final output is usually called a label. Some methods used for classification tasks are Random Forest, Decision Trees, Logistic Regression, and Support Vector Machines [164].

Among the different methods used for regression and classification, we focus on *neural networks*, which go under the umbrella of non-linear regression or non-linear classification.

Generally, a simple learning model consists of three items:

- An input consisting of features.
- An output which is either a continuous value for regression tasks or a discrete value representing a category in classification tasks.
- The function that maps this input to the output. This function has to be **learned**.

Before diving into the concept of neural networks, we present the simplest learning task which is linear regression, and an extension of it known as logistic regression that is used for classification

A.2 Linear Regression

A basic learning model is the linear regression [165], [166]. Given a scalar input x_i and output y_i, the goal is to learn a linear function $f(x_i)$ such that:

$$f(x_i) = y_i = W \cdot x_i + b$$

W is a scalar called the slope or weight, and b is a scalar called the bias. Precisely, the task is to learn W and b. This function is a line that can pass through two points (i.e., (x_1, y_1) and (x_2, y_2)). In case there are more than two points, the line can not fit all of them. Supposing there exist M training points, the mean square error function should be minimized:

$$\min \frac{1}{M} \sum_{i}^{M} (f(x_i) - y_i)^2$$

In a 2-dimensional plane, the linear regression can only cut the plane into two parts. Its accuracy is likely to be very low. Hence it is required to use non-linear functions that have higher accuracy.

It is worth noting that the input and the weight can be vectors instead of being scalars such that $x_i = [x_i^{(1)}, x_i^{(2)}, x_i^{(3)}...., x_i^{(n)}]$ and $W = [w^{(1)}, w^{(2)}, w^{(3)}...., w^{(n)}]$. In this case, the output is summation of linear functions.

A.3 Logistic Regression for Classification

Given that a linear function produces outputs belonging to continuous intervals, they can only be used for regression tasks. One way to discretize the output of a linear function is to use a sigmoid function as such:

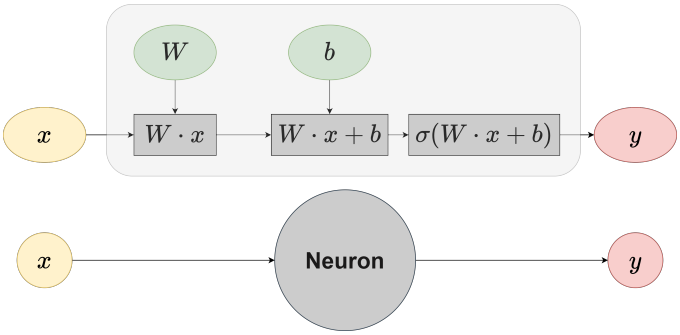


Figure A.1: The operations in a neuron

$$\sigma(f(x_i)) = \frac{1}{1 + e^{-f(x_i)}} = \frac{1}{1 + e^{-(W \cdot x_i + b)}}$$

The output here is a value between 0 and 1. This can be considered as a probability distribution such that if the value is less than 0.5, the output is considered as *False*. Otherwise, it is considered *True*. Because of discretization thanks to the sigmoid function, the linear regression becomes logistic regression. This is the simplest case of binary classification [166].

Both the linear and logistic regressions are simple models that can not serve complex tasks with many features, and many dependencies. Hence, neural networks raise as much more powerful alternatives.

A.4 Neural Network Architecture

A neural network consists of small atomic elements called neurons.

A.4.1 Neuron

The simplest element in a neural network is a called a neuron [164]. The perception contains a group of neuros. Each neuron consists of:

- input x; a vector of size $N \times 1$, where N is the number of features.
- weight W; a vector of size $N \times 1$.
- bias b; a scalar.
- activation function σ; a function used to invoke non-linearity.
- output y; a scalar.

The output of a neuron is:

$$y = \sigma(W \cdot x + b)$$

$W \cdot x$ is the Dot product of vectors. While the neuron output seems similar to the logistic regression presented above, the difference is that the output of the neuron is more general. The latter's objective is not to have a value between 0 and 1 as in logistic regression. It aims to add non-linearity to the model to be able to learn harder tasks. The flow of operations in a neuron is shown in Figure A.1. Some commonly used activation functions are: For an input z such that $z = W \cdot x + b$

- The sigmoid function:
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$
- The tanh function:
$$\sigma(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$
- The relu function:
$$\sigma(z) = max(z, 0)$$
- The softmax function:
$$\sigma(z) = \frac{e^{-z_i}}{\sum_{j=0}^{k} e^{-z_j}}$$

The sigmoid yields an output between 0 and 1, while tanh yields an output between -1 and 1. The relu outputs either z or 0 when $z < 0$. The softmax is used in multi-class classification tasks (e.g., non-binary, as in classifying an animal as a cat, a dog, a horse, or not an animal), and it outputs probabilities. One of the problems with using tanh and sigmoid is that the gradient becomes 0 as $|z|$ grows in a phenomenon known as the vanishing gradient. This affects the ability of the model to learn. On the other hand, the gradient of relu is either 1 or 0. This helps the model learn faster [164].

A.4.2 Neural Networks: Multi-Layer Perceptron

In the previous section, we discussed the basic element of a neural network called a neuron. A neuron can take a part in a much more complex network called a neural network. It is also called multi-layer perceptron [164]. A neural network consists of multiple layers: The input layer, the output layer, and one or more hidden layers. Each layer consists of a set of neurons. The input layer consists of different features represented by numerical values (i.e., continuous or discrete). Its features are forwarded to the next layers without applying mathematical operations on them. The neurons from one layer are input to each neuron in the following layer. Hence, each neuron in the hidden layers receives the outputs of the neurons of the previous layer, aggregates them and adds the bias, and then applies the activation function. The output of a hidden layer is fed to all the neurons in the following layer. Finally, the output layer outputs a numerical value or a label depending on whether the task is regression or classification.

When there are more than three hidden layers and more than three neurons per layer, the neural network is typically called a deep neural network. The concept of learning

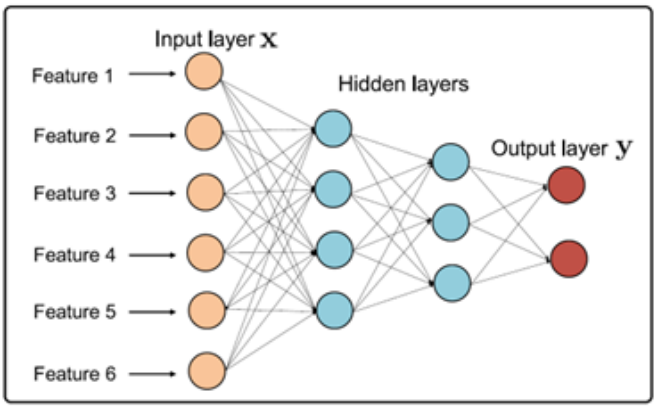


Figure A.2: The Neural Network [164]

when a deep neural network is used is called *deep learning*. The basic architecture of a neural network is shown in Fig. A.2.

While this appendix is quite limited to the basic architecture of neural networks (i.e., multi-layer perceptron, the neural network could be modified to different architectures. For example, it is possible to support sequential inputs as in recurrent neural networks used for time series and natural language processing. In convolution neural networks, the neural network is modified such that the convolution operation is used to learn different filters. In the multi-layer perceptron, the features are known, and the model learns how to perform the task. In convolutional neural networks, the features are not known. The model has to learn how to predict correctly and what features it needs to do so. These features are learned by learning different filters that apply the convolution function. These filters aim to detect features in images that aid the classification process. A convolution neural network just takes the values of the pixels of an image as input. Moreover, graph neural networks allow cyclic connections between neurons. As a note, given that we used BiLSTM in Chapter 5, we present its architecture in the last section of this appendix.

A.5 The learning process: The workflow

To train a neural network, a procedure has to be followed.

A.5.1 Defining the task

The first thing to do is to define the task that the machine learning model should learn, what features could be required to execute this task, and what architecture could be used. For example, consider the task of predicting whether an image shows a dog or a cat. The architecture could be convolutional neural networks since the tasks deal with image classification. Given that this architecture learns the features during the training process by learning the filters, defining the features (e.g., the size, shape, weight, height,

etc.) at this phase is not necessary.

A.5.2 Data collection and pre-processing

Next, a dataset should be gathered. A dataset consists of multiple instances. Each instance contains a feature vector and the output; either a value or a label depending on the task. The pre-processing consists of modifying the dataset such that the features are normalized. This would aid the convergence of the gradient descent process used to train the network. In case the features represent an image (i.e., represent the pixel of an image) some pre-processing techniques include resizing the images such that all images have the same pixel size. Additionally, data augmentation can be used. In this case, an image can be rotated and flipped and all these versions could be added to the dataset to help the model learn that all these variations have the same output. In addition, the dataset can be divided into a training set, a validation set, and a test set. The training set is used to train the model. The validation set is used during training to decide if the model is performing well on data it did not see and whether it needs further training or not. The test set is used at the end to test the accuracy of the model.

A.5.3 Defining the architecture

The next step is to define the neural network architecture. As mentioned above, different neural network architectures can be used and the one that best suits the task should be chosen. The parameters used to control the training process before it starts are called hyperparameters. They include the number of layers in the neural network, the number of neurons per layer, the activation functions, etc., and they are selected through trial and error. However, one can look for similar tasks that have been already trained, and use the hyperparameters of these models as a starting point.

A.5.4 Loss function

The performance of a model is measured by how much accurate it is; how much it successfully predicts the correct value. The loss function determines how much the predicted value deviates from the correct value. In regression tasks, the mean square error function can be used. In classification tasks, the cross-entropy function is used. Supposing that $y_{i,k}$ is the actual value $\hat{y}_{i,k}$ is the predicted value and that there are M samples and K possible classes (i.e., labels), the cross-entropy loss functions is:

$$-\sum_{i}^{M}\sum_{k}^{K}(y_{i,k}\log(\hat{y}_{i,k}))$$

A.5.5 Training: Forward and backward propagation

The training phase consists of repeatedly applying the forward propagation and backward propagation. Forward propagation is the process of passing the input vector in the model and propagating the output of each layer into the following layer until the output value is finally yielded. This output value is compared to the desired output (i.e., the label in classification). Since the goal is to minimize the error, the gradient with respect

to all the weights and biases must be calculated. The process of calculating the gradient of the loss function with respect to the weights, one after the other, is called backward propagation. To calculate the gradient, it is extremely complex to find the closed-form solution (i.e., by setting the derivative function to zero and solving for each variable). Hence, the method of gradient descent is used. The gradient descent calculates the gradient with respect to each learnable parameter (i.e., weights and bias). Then it updates the weights by taking a small step in the opposite direction of the gradient.

During training, it is unlikely that the full data set will be input simultaneously to the model due to the computational limitations of the available hardware. Hence, a mini-batch of instances is selected at each iteration, and their combined loss is calculated and is compared to the performance of the validation set, to decide whether to stop training or not. The criteria for stopping depend on the task requirements defined by the users of this model. One stopping criterion could be to stop once the accuracy of the mini-batch and the validation set is more than 90% (i.e., the model is correctly classifying 90% of the dataset).

A.5.6 Testing and deployment

The last phase is to test the model on a dataset wasn not used for training. Accordingly, one would decide if the model has to be retrained, or if it is ready for deployment.

A.6 Bi-Directional Long-Short-Term-Memory (Bi-LSTM)

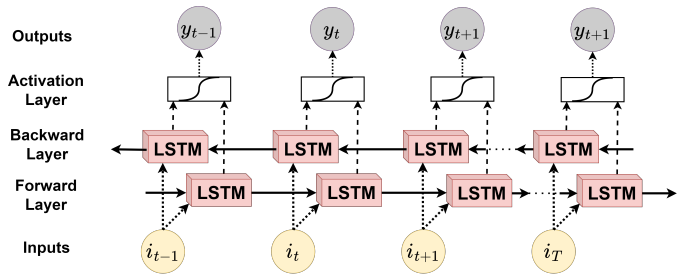


Figure A.3: A BiLSTM-based Recurrent Neural Network Architecture

In Chapter 5, we have used the Bi-Directional Long-Short-Term-Memory (Bi-LSTM) architecture. This section explains the mathematical operations done inside this architecture. Recurrent neural networks are designed to output results when the input is a sequence of input vectors representing interdependent elements. For example, in language translation tasks, the positions of words in the sentence are dependent as the subject comes before the verb, which in turn comes before the object. Another example is predicting the weather based on the previous several days, where there could be interdependence among the temperatures of the consecutive days. The architecture of a recurrent neural network that uses a BiLSTM is shown in Fig. A.3 [164]. The LSTM preserves the memory from previous inputs in the sequence. The output for an element

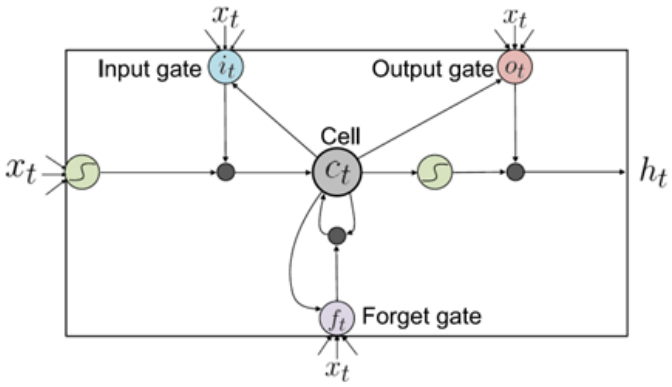


Figure A.4: The internal architecture of LSTM layer [164]

is not affected by the elements that follow it in the sequence. In contrast, the BiLSTM captures the dependencies in both directions. Hence before producing an output for an element, it would be possible to traverse all precedent and subsequent elements. Hence, the decision for an element take the whole sequence into account.

The internal architecture of the LSTM is shown in Fig. A.4. The LSTM layer applies the following operations:

$$i_t = \sigma(W_{xi} \cdot x_t + W_{hi} \cdot H_{t-1} + b_i)$$

$$f_t = \sigma(W_{xf} \cdot x_t + W_{hf} \cdot H_{t-1} + b_f)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot tanh(W_{xc} \cdot x_t + W_{hc} \cdot H_{t-1} + b_c)$$

$$o_t = \sigma(W_{xo} \cdot x_t + W_{ho} \cdot H_{t-1} + b_o)$$

$$h_t = o_t \odot tanh(C_t)$$

In Fig. A.4 and the equations above, x_t is the input to the LSTM at step t. The $\odot$ represents the Hadamard product (i.e., element-wise product), C_t denotes the cell outputs, H_t represents the hidden states that are propagated to the following element (i.e., the element at $t+1$) in the sequence, i_t represents the operations at the input gates, f_t at the forget gates, and o_t at the output gates.

W_{xi} and W_{hi} are learnable weight matrices at the input gate and b_i is the bias. W_{xf} and W_{hf} are learnable weight matrices at the forget gate and b_f is the bias. W_{xo} and W_{ho} are learnable weight matrices at the output gate and b_o is the bias. W_{xc} and W_{hc} are learnable weight matrices at the cell output and b_c is the bias. The forget gate determines what information from previous steps is needed. The input gate decides what information from the input x_t is needed at the current step t, and the output gate decides what information should be transferred to the next step $t+1$.

Appendix B

Reinforcement Learning

Reinforcement learning is a category of machine learning where an agent interacts with the environment by executing an action and evaluating it through the reward and penalty it receives. This is similar to the way a child learns. This appendix explains some basics of reinforcement learning.

B.1 The elements of reinforcement learning

The elements of reinforcement learning are [163]:

- Agent: The agent is the learner. The agent has the ability to execute actions and explore the actions that best-maximize his rewards.
- Environment: The environment is the surrounding with which the agent interacts and receives rewards and penalties.
- State: The state is the situation/condition of the agent.
- Action: An action is executed by the agent in the given environment.
- Reward: After executing an action, the agent receives a reward or penalty from the environment.
- Value function: The value function represents the quality of doing an action given a state. It includes the long-term reward of doing an action taking into account the probable future actions and states that are likely to follow after executing this action.
- Policy: The policy defines how the agent should behave, more precisely, which actions the agent should perform for each state. The policy maps the perceived states of the environment to the actions to be taken in those states. This mapping can be deterministic or probabilistic.

- Model: The model of the environment provides the expected behavior and reactions of the environment when the agent interacts with it. It can be used to plan what actions to take in advance. If the model is known, the agent can know what actions to select given a specific state, such that the total accumulative reward is maximized. When the model is unknown, the agent attempts to learn it by interacting with the environment

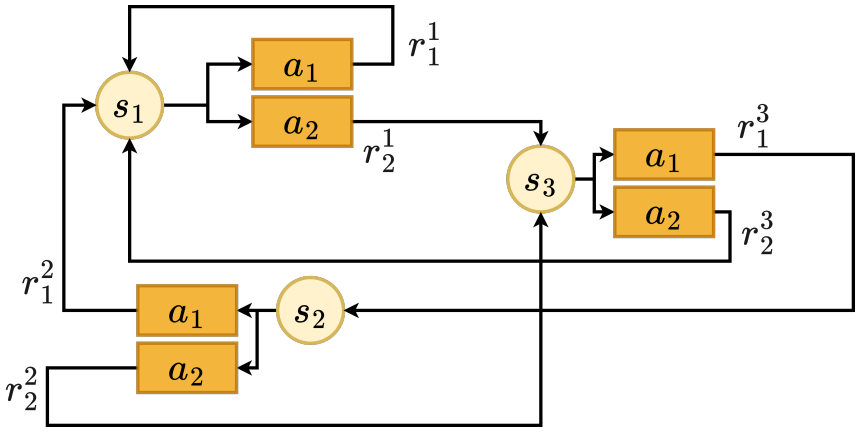


Figure B.1: Markov Decision Process

B.2 Markov Decision Process modeling

The model of an environment can be represented as a Markov Decision Process (MDP). An MDP is represented by:

- states
- actions
- rewards received when an action is executed

The MDP is shown in Figure B.1. For each state s_i, it is possible to execute an action a_j, get a reward r_j^i and move to the next state. This forms the basis for how the agent learns. The agent is unlikely to have prior knowledge of the dynamics of the environment and the possible rewards. The agent has to learn these by interacting with the environment and registering the experience. In the MDP, it is possible that the transitions and rewards are either deterministic or stochastic. In the latter, for a given state, executing the same action may give different rewards and the agent may transit to different states. Hence, the agent should choose actions that maximize the expectation of the long-term reward.

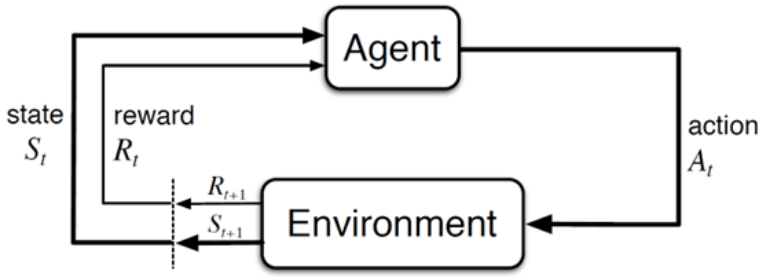


Figure B.2: The Reinforcement Learning Procedure [163]

B.3 Learning approaches

The learning process consists of two approaches:

- Value-based learning; the agent learns the value function that represents the quality of doing each action. Then, it can select its own policy, which can be either stochastic or deterministic. During the learning process, the estimation of the value function is updated and improved.
- Policy-based learning; the agent directly learns the optimal policy it should follow. It does not learn the value function. In this type, the agent learns a probability distribution on the actions and follows this distribution when choosing the action. During the learning process, the probability distribution of selecting actions is updated and improved.

B.3.1 Value-based learning

Considering a Markov Decision Process (MDP) in Fig. B.2, an agent at instant t and at state S_t executes an action A_t to receive a reward R_t and gets moved to state S_{t+1}. Let $\mathcal{S}$ be the set of states, $\mathcal{A}(s)$ be the set of actions for state s, and $\mathcal{R}$ the set of rewards [163].

The probability for an agent to be in state s' and to receive a reward r given it is at state s and will do action a is:

$$p(s', r|s, a) \doteq Pr\{S_t = s', R_t = r|S_{t-1} = s, A_t = a\}$$

$$\sum_{\forall s' \in \mathcal{S}} \sum_{\forall r \in \mathcal{R}} p(s', r|s, a) = 1, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}(s)$$

Let γ be the discount factor. The *discounted reward* also called *return* is defined by

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

γ is between 0 and 1. If it is 0, the agent does not care about future rewards. When it is 1, the agent fully considers the future rewards and takes actions such that the return is maximized. The value function of state s is defined as

$$v(s) \doteq \mathbb{E}_\pi = [G_t|S_t = s] = \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s\Big]$$

The value of taking action a at state s which is known as the action-value function:

$$q(s,a) \doteq \mathbb{E}_\pi = [G_t|S_t = s, A_t = a] = \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a\Big]$$

The Bellman optimality equations of the state value and action values functions are:

$$v_*(s) \doteq \max_\pi v_\pi(s) = \max_a \sum_{\forall s' \in \mathcal{S}} \sum_{\forall r \in \mathcal{R}} p(s', r|s, a)[r + \gamma v_*(s')]$$

$$q_*(s,a) \doteq \max_\pi v_\pi(s) = \sum_{\forall s' \in \mathcal{S}} \sum_{\forall r \in \mathcal{R}} p(s', r|s, a)[r + \gamma \max_{a'} q_*(s', a')]$$

The goal of the agent is to learn $q_*(s,a)$ or $v_*(s)$. $q_*(s,a)$ could be preferred given that it directly tills the agent which action to take. Given that solving the bellman equations is unlikely to be feasible, especially when the environment model is not available, dynamic programming is used to converge to the optimal value function iteratively. A well-used method is known as Q-Learning. In this method, the update rule is

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\Big]$$

In this rule, α is the learning rate between 0 and 1, and it indicates how quickly a model learns from new updates.

The learning process can be on-policy or off-policy. In on-policy learning, the same policy is used to execute the actions and to update the action-value functions. In off-policy, as in Q-learning, a probabilistic policy (e.g., ϵ-greedy) is used for executing actions, while a deterministic policy is used in the update rule.

Different learning algorithms can be used to learn action-value functions. These include Temporal Difference, SARSA, and Q-learning, among others. While the update rules involve either updating the value functions or the action-value functions, the difference between these algorithms lie in how and when to apply the updates, and whether it is on-policy learning or off-policy learning.

The Q-learning algorithm is shown in Fig. B.3. The agent aims at updating its action-value function until it converges to the optimal. The agent follows an ϵ-greedy policy. In this policy, the agent selects a random action with probability ϵ and chooses the action that maximizes $q(s,a)$ with a probability of $1-\epsilon$. After the action is selected, the agent receives a reward and moves to the following state. It updates the q-value and repeats.

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\left[R + \gamma \max_a Q(S', a) - Q(S, A)\right]$
    $S \leftarrow S'$
  until S is terminal

Figure B.3: Q-Learning Algorithm [163]

B.3.2 Policy-based learning

In Chapter 5, we used policy-gradient-based RL method. This method learns the optimal policy without learning the value functions. While the value-based learning finds the gradient of the value function, policy-based learning finds the gradient of the policy itself.

A famous policy gradient algorithm called REINFORCE is used in Chapter 5. Suppose that the $P(s^i, a^i) = f(\theta)$ represents the probability of executing action a^i when in state s^i, where f is a function that is parameterized by the learnable variables θ. θ can be regarded as a matrix of variables. These variables can be called weights when $f(\theta)$ is a neural network. Additionally, let v^i the discounted rewards:

In REINFORCE, the weights θ should be updated using the following formula:

$$\theta \leftarrow \theta + \alpha v^i \nabla log(P(s^i, a^i))$$

where α is the learning rate. Additionally, there exist different policy gradient algorithms, which include but are not limited to: Actor-Critic, Proximal Policy Optimization, Deep Deterministic Policy Gradient, among others.

B.4 Function approximation

One way to store the action-value function $q(s, a)$ is to put the values in a table. This method is called Tabular-Q-Learning. However, this method becomes infeasible as the state space or the action space grows. Hence, a function that takes the state s and action a as an input and outputs the q-value can be used. This way, only the function parameters have to be learned and stored. One famous function is the Deep Neural Network. Deep Q-Learning uses a Deep Neural Network as the q-value approximator. Taking the state as an input, the neural network outputs a value corresponding to each action. Then an action can be selected according to the adopted policy (e.g., ϵ-greedy). Using Q-learning or a Deep-Q-Network is not suitable when the action is continuous. One method is to discretize the continuous action into discrete levels. This could be

inefficient, as the action space may bulk. The other option is to use the policy-gradient methods such as Actor-Critic, Proximal Policy Optimization, Deep Deterministic Policy Gradient. In fact, policy-gradient algorithms need a function that outputs a probability distribution over actions, whether they are continuous or discrete. Neural networks are good function approximators in this case.