UNIVERSITÉ DU QUÉBEC À MONTRÉAL

GESTION DES RESSOURCES ET FONCTIONS RÉSEAU VIRTUELLES: DU MONO-DOMAINE AU MULTI-DOMAINE

THÈSE

PRÉSENTÉE

COMME EXIGENCE PARTIELLE

DU DOCTORAT EN INFORMATIQUE

PAR

MOUHAMAD DIEYE

SEPTEMBRE 2023

REMERCIEMENTS

Je désire en premier lieu exprimer ma sincère gratitude à ma directrice de recherche, Prof. Halima Elbiaze, professeure titulaire au département informatique de l'UQAM. Je tiens à la remercier sincèrement de son soutien indéfectible, de sa disponibilité, de ses conseils et de ses excellentes qualités humaines tout au long de ce parcours.

À mes très chers parents, M. Abdoulaye Dieye et Mme Fatoumata Hawra Diop, je voudrais réitérer ma gratitude et mon respect pour leur soutien indéfectible et inconditionnel, leurs prières, leurs conseils ainsi que leurs sacrifices. J'espère qu'ils retrouveront à travers ce travail, un témoignage de l'estime et de la gratitude que je leur porte.

Mes sincères remerciements au Prof. Roch Glitho, professeur titulaire au Concordia Institute of Information Systems Engineering (CIISE) de l'Université Concordia, pour sa disponibilité et ses conseils.

Je désire exprimer ma sincère gratitude au Prof. Wael Jaafar, professeur à ETS, pour sa disponibilité, ses encouragements et son assistance. J'étends mes remerciements à mes co-auteurs pour leur coopération, leur disponibilité ainsi que leurs conseils.

Un remerciement spécial aux membres du jury, les professeurs Walid Saad, Elmahdi Driouch et Abdoulaye Baniré Diallo pour leur travail acharné dans l'évaluation et l'amélioration de la thèse.

Je voudrais, en particulier, exprimer ma sincère gratitude à mes grands-parents M. Serigne Mor Diop et Mme Sokhna Khady Ndiaye pour leur soutien moral et leurs encouragements indéfectibles. Au-delà de leurs précieuses prières, j'admire leur aptitude à l'écoute, à l'humilité et à faire preuve d'ouverture d'esprit.

Je rends également hommage à mes regrettés grands-parents, M. Assane Dieye et Mme Sokhna Khady Diop pour leurs prières et leurs encouragements.

Je remercie mes sœurs, Mme Khadija et Mme Mariétou ainsi que mes frères, M. Ibrahim, M. Ahmad Tijani, M. Al Hadji Malick, M. Assane, et M. Mor. Je vous adresse mes meilleurs voeux de succès et de bonheur.

J'exprime mes vifs remerciements au Prof. Abdoulaye Baniré Diallo pour sa disponibilité et son mentorat.

Mes remerciements sincères s'adressent également à tous mes collègues du laboratoire TRIME de l'UQAM. Je remercie également tous les professeur(e)s de l'UQAM qui ont gracieusement partagé avec moi leur expertise.

Tout au long de ce cheminement, j'ai eu l'honneur de bénéficier de l'aide financière de ma direction de recherche, la Fondation UQAM, la Fondation J. -A. DeSève, M. Fernand Lafleur, et le département informatique de l'UQAM. Merci sincèrement.

À toute ma famille, mes ami(e)s, tous ceux qui, de près ou de loin, m'ont aidé à atteindre mes objectifs. Je vous exprime mes sincères remerciements.

DÉDICACES

À ma famille, pour votre soutien et vos encouragements.

In memory of M. Alioune Aïdara Karara and Mme Aïda Sarr, who have shown me so much kindness. It mattered. RIP.

TABLE DES MATIÈRES

TABLE DES FIGURES xi

LISTE DES TABLEAUX xiv

ACRONYMES xvii

RÉSUMÉ xix

ABSTRACT xxii

CHAPITRE I INTRODUCTION GÉNÉRALE 2

1.1 Mise en contexte et problématiques 2

1.1.1 Contexte actuel des réseaux: un aperçu 2

1.1.2 NGNs et 5G: De nouveaux types de services et applications . 9

1.1.3 Partage des responsabilité d'investissements et impacts 13

1.1.4 Le découpage en tranches du réseau 15

1.2 Orchestration de ressources dans les réseaux multi-domaines 21

1.2.1 Dynamisme et instabilité 21

1.3 Objectifs 25

1.4 Organisation de la thèse 25

CHAPITRE II MÉTHODOLOGIE 30

2.1 Problématiques et motivations méthodologiques 30

2.1.1 Premier enjeu : la taille et la nature fondamentale du problème 30

2.1.2 Second enjeu : la nature des applications et services 31

2.1.3 Troisième enjeu : la nature des décisions à prendre 33

2.2 Contributions de la thèse 37

2.3 Critères de la qualité d'étude 42

2.3.1 Expérimentations 43

2.4 Rappels sur les algorithmes utilisés 44

2.4.1 Programmation linéaire en nombres entiers (Bradley et al., 1977) . . . 45
2.4.2 PageRank (PR) . . . 49
2.4.3 Évolution différentielle (DE) . . . 50
2.4.4 Apprentissage par renforcement profond (DRL) . . . 54
CHAPITRE III ARTICLE 1 - CPVNF : COST-EFFICIENT PROACTIVE VNF PLACEMENT AND CHAINING FOR VALUE-ADDED SERVICES IN CONTENT DELIVERY NETWORKS . . . 63
AVANT-PROPOS À L'ARTICLE 1 . . . 65
ABSTRACT . . . 67
3.1 Introduction . . . 68
3.2 Illustrative Use Case, Requirements and Related Work . . . 71
3.2.1 Illustrative Use Case . . . 71
3.2.2 Requirements . . . 73
3.2.3 Related Work . . . 74
3.3 VNF Placement Problem . . . 80
3.3.1 Problem Description . . . 80
3.3.2 ILP Formulation . . . 82
3.4 Cost-efficient Proactive VNF Placement (CPVNF) . . . 90
3.4.1 Selecting the content server . . . 91
3.4.2 Selecting the surrogate servers . . . 92
3.5 Performance Evaluation . . . 96
3.5.1 Experimental Set up . . . 96
3.5.2 Performance Metrics . . . 98
3.5.3 Results and Discussions . . . 99
3.6 Conclusion and future work . . . 105
REFERENCES . . . 107
CHAPITRE IV ARTICLE 2 - TOWARDS RELIABLE REMOTE HEALTH MONITORING IN FOG COMPUTING NETWORKS . . . 114
AVANT-PROPOS À L'ARTICLE 2 . . . 115

ABSTRACT . . . 117
4.1 Introduction . . . 118
4.2 Related Work . . . 123
4.2.1 Cloud-based remote health monitoring . . . 123
4.2.2 Fog Computing-based Remote Health Monitoring . . . 124
4.2.3 COVID-19 Centric Remote Health Monitoring . . . 125
4.2.4 Reliable Service Placement in FC Networks . . . 127
4.3 Fog-based remote health monitoring model . . . 128
4.3.1 Use Case Description . . . 128
4.3.2 F-RHM Framework and Modules . . . 132
4.4 Problem Formulation . . . 136
4.4.1 QoS Requirements Expressions . . . 138
4.4.2 Problem Formulation . . . 141
4.5 Proposed Smart Solution : Differential Evolution-Deep-Q-Networks (DE-DQN) . . . 144
4.5.1 Background on Differential Evolution . . . 144
4.5.2 Adaptation of Differential Evolution . . . 147
4.5.3 DQN State and Reward Definitions . . . 149
4.5.4 Complexity Analysis . . . 154
4.6 Experimental evaluation . . . 155
4.7 Conclusion . . . 162
REFERENCES . . . 163
ANNEXE A . . . 171
ANNEXE B . . . 173
B.1 Fiabilité . . . 173
B.2 Taux de pannes . . . 174
B.3 Maintenabilité . . . 175
B.4 Modélisation de la disponibilité . . . 176

CHAPITRE III ARTICLE 3 - DRL-BASED GREEN RESOURCE PROVISIONING FOR 5G NETWORKS 182
AVANT-PROPOS À L'ARTICLE 3 183
ABSTRACT 187
3.1 Introduction 188
3.2 Related Work 192
3.2.1 ML-driven Admission Control 193
3.2.2 ML-driven Resource Allocation 194
3.3 System Model 196
3.4 Problem formulation 197
3.4.1 MNO's Green Wireless Resource Allocation Problem 197
3.4.2 MNO's Green VNF Resource Provisioning Problem 202
3.5 Proposed Solutions 212
3.5.1 The Architectural Framework for Machine Learning in Future Networks 212
3.5.2 DRL-MCTS : A Hybrid DRL and MCTS-based Wireless Resource Allocation Strategy 212
3.5.3 DRL-VNF : A Green DRL-based VNF Provisioning Strategy . 219
3.5.4 Complexity analysis 226
3.6 Experimental evaluation 226
3.6.1 MNO's Wireless Resource Allocation 227
3.6.2 MNO's VNF Resource Provisioning 230
3.7 Discussion 236
3.8 Conclusion 240
REFERENCES 242
CHAPITRE IV ARTICLE 4 - MARKET DRIVEN MULTI-DOMAIN NETWORK SERVICE ORCHESTRATION IN 5G NETWORKS 252
AVANT-PROPOS À L'ARTICLE 4 253
ABSTRACT 254
4.1 Introduction 255

4.2 Massive Multiplayer Online Game use case 259
4.2.1 Overview 259
4.2.2 Requirements 262
4.3 Related Work 263
4.3.1 SDN-based Orchestration 264
4.3.2 NFV-based Orchestration 264
4.4 System Model 268
4.5 Problem formulation 270
4.6 Auction scenarios 274
4.6.1 Scenario 1 274
4.6.2 Scenario 2 275
4.7 Proposed solutions 276
4.7.1 From the InP's Perspective 276
4.7.2 From the SPs' Perspective : A Distributed Multi-agent Deep Reinforcement Learning Approach 280
4.7.3 Bidding analysis 282
4.8 Experimental evaluation 284
4.8.1 From the InP perspective 287
4.8.2 From the SPs perspective 289
4.9 Discussion 293
4.9.1 Context 293
4.9.2 Architectural implications for Future Networks 294
4.9.3 Multiple InPs 295
4.10 Conclusion 296
REFERENCES 297
CHAPITRE V CONCLUSIONS ET TRAVAUX FUTURS 309
5.1 Conclusions 309
5.2 Limites et travaux de recherche futurs 316

BIBLIOGRAPHIE 318

TABLE DES FIGURES

Figure | Page

1.1 Interactions entre les sources d'instabilité dans un réseau dynamique (Clayman et al., 2012) . . . 23

2.1 Approche méthodologique de la thèse . . . 43

3.1 Video Advertisement Overlay Use case . . . 72

3.2 Number of servers used . . . 100

3.3 Communication cost . . . 100

3.4 Operational cost . . . 101

3.5 Total cost . . . 101

3.6 Average response time . . . 102

3.7 Number of content servers used . . . 103

3.8 Comparison of total cost with reduced α_k and δ_n - 9 servers . . . 103

3.9 Number of server used with tighter specifications - 9 servers . . . 104

4.1 F-RHM system model. . . . 128

4.2 F-RHM flow diagram. . . . 131

4.3 Components of F-RHM modules. . . . 133

4.4 Flowchart of the DE-DQN training algorithm . . . 150

4.5 Performance comparison of proposed and benchmark algorithms. . . 156

4.6 Training convergence behavior of DE-DQN algorithm. . . . 157

4.7 Success ratio versus the number of fog nodes - 100 fog nodes . . . 157

4.8 Success ratio versus the number of fog nodes - 150 fog nodes . . . 158

4.9 Success ratio versus the number of fog nodes - 200 fog nodes . . . 158

4.10 Success ratio versus the fog nodes' AFR - Fog nodes' AFR less than 3% . . . 159

4.11 Success ratio versus the fog nodes' AFR - Fog nodes' AFR in range [3%-15%] . . . 159

4.12 Success ratio versus the fog nodes' AFR - Fog nodes' AFR over 15% 160

3.1 System model . . . 196

3.2 Potential S-E associations. . . . 213

3.3 DRL-MCTS iterative run. . . . 213

3.4 Overview of the multi-agent hierarchical DRL framework, inspired by (Kumar et al., 2017). . . . 216

3.5 Overview of the DRL-VNF process. . . . 224

3.6 Overview of the interactions between path selector and path evaluator. 224

3.7 Sum-rate per number of users - Empirical CDF for 10 users. . . . 228

3.8 Sum-rate per number of users - Empirical CDF for 20 users. . . . 229

3.9 Sum-rate per number of users - Empirical CDF for 30 users. . . . 230

3.10 Average cumulative rewards for DRL-MCTS and WMMSE. . . . 231

3.11 Profit vs. number of users. . . . 231

3.12 Average cumulative rewards vs. number of nodes. . . . 234

3.13 Success ratio. . . . 234

3.14 Server costs . . . 235

3.15 Communication costs . . . 236

3.16 Total costs . . . 237

4.1 MMOG scenario. . . . 261

4.2 Bidding process. . . . 269

4.3 Average InP profits in different auction environments. . . . 288

4.4 Average number of excluded SPs per episode. 288

4.5 Average total profits per episode under each algorithm. 289

4.6 Average total SP profits under different auction settings. 290

4.7 Average SP profits under full competition auction environment. . 291

4.8 Impact of penalty function on agent behaviour. 292

LISTE DES TABLEAUX

Tableau Page

1.1 Exigences QoS pour les services et applications prévues pour la 5G et les NGNs, extraites de (Saad et al., 2020) 11

1.2 Résumé des classes de services avec leurs indicateurs de performance, extrait de (Saad et al., 2020) 12

3.1 Input Parameters and Variables 83

3.2 Simulation Parameters (Bouet et al., 2015 ; Cacheda et al., 2007) 97

3.3 Topology parameters 98

4.1 Fog node monitoring subsystem database excerpt
Source : Manufacturer datasheet, unless otherwise specified.
* Blancco Q4 2017 State of Mobile Device Repair & Security report. 136

4.2 Table of notations 140

4.3 Summary of state features (Sharma et al., 2019) 153

4.4 Hyperparameters of DE-DQN 154

3.1 Notations (Wireless resource allocation) 203

3.2 CO_2 emissions (Ricciardi et al., 2013) 207

3.3 Notations (Green VNF resource provisioning) 210

3.4 Node Power consumption profile (Zhang et al., 2013) 232

3.5 Power consumption measurements of switches/routers (Vishwanath et al., 2014) 232

4.1 Notations 273

4.2 Sim. Parameters 286

4.3 Budget Distribution 287

ACRONYMES

C

CDN Content Delivery Networks.

D

DE Differential Evolution.

DRL Deep Reinforcement Learning.

I

IA Intelligence Artificielle.

ILP Integer Linear Programming.

M

ML Machine Learning.

N

NFV Network Function Virtualization.

NGN Next Generation Networks.

Q

QoE Quality of Experience.

QoS Quality of Service.

T

TCP Transmission Control Protocol.

V

VNFs Virtual Network Functions.

RÉSUMÉ

Depuis toujours, la priorité de chaque génération de technologies de réseau, de la fibre optique aux réseaux cellulaires, a été d'améliorer l'efficacité spectrale afin de parfaire l'expérience des utilisateurs. Autrefois, les téléphones portables ne servaient qu'à envoyer des messages textes et à passer des appels vocaux. Aujourd'hui, nos appareils sont dotés de nombreuses fonctionnalités avancées, allant de la diffusion de médias en continu au traitement de tâches de calcul complexes.

Le développement constant d'applications et de services innovants, qui requièrent une qualité de service de bout en bout de plus en plus exigeante, combiné à la mobilité croissante des utilisateurs finaux, a eu un impact considérable sur l'échelle et les coûts d'exploitation des réseaux, le volume et l'hétérogénéité du trafic, ainsi que sur les modèles d'accès. Cette situation a contraint les opérateurs de réseau à apporter de nombreux changements de paradigme importants dans la gestion de leurs réseaux. En particulier, les opérateurs de réseaux de la prochaine génération (*Next Generation Networks - NGNs*) se sont fortement concentrés sur la conception de mécanismes de fourniture de services à faible latence de bout en bout (moins de dix millisecondes).

L'intégration de technologies établies et émergentes, comme la 5G, l'intelligence artificielle (IA), l'intelligence en périphérie, la logiciellisation des réseaux et la programmabilité du plan de données, devrait permettre aux réseaux de la prochaine génération d'atteindre des vitesses ultrarapides et une latence très faible. Cependant, face à la croissance exponentielle du trafic et à l'augmentation de la complexité des systèmes, de nouvelles stratégies de gestion sont nécessaires pour s'adapter à la nature dynamique de la configuration des réseaux et pour permettre l'automatisation des systèmes de bout en bout grâce à des systèmes d'apprentissage distribués. En utilisant des solutions de gestion multi-domaines, coopératives, distribuées et basées sur l'apprentissage, les utilisateurs finaux peuvent bénéficier de services et d'applications plus fiables, efficaces, évolutifs, sécurisés et diversifiés.

Cette thèse, composée de plusieurs articles, étudie les différentes approches méthodologiques pour l'orchestration multi-domaines de ressources virtuelles dans les réseaux de la prochaine génération afin de fournir des services à valeur ajoutée à des exigences de qualité de service (*Quality of Service - QoS*) strictes. En raison de la complexité de la question de l'orchestration, chaque article se concentre sur

une partie du problème, en commençant par les réseaux mono-domaines et en progressant vers les réseaux multi-domaines. Nous utilisons les enseignements tirés de nos publications précédentes pour élaborer de nouvelles stratégies pour nos travaux futurs.

Le premier article de cette thèse, intitulé **CPVNF : Cost-efficient Proactive VNF Placement and Chaining for Value-Added Services in Content Delivery Networks**, porte sur le problème de fourniture de ressources virtuelles dans un réseau de diffusion de contenu mono-domaine (Content Delivery Networks - CDN) afin de garantir les exigences QoS d'applications multimédias. Le problème présente deux principaux obstacles : (1) la présence de plusieurs sources de contenus dans un réseau CDN et (2) l'hétérogénéité de la chaîne de fonctions réseaux virtuelles (*Virtual Network Functions - VNFs*) à établir selon l'utilisateur final. Pour remédier à ces difficultés, nous proposons un algorithme proactif de placement et de chaînage d'instances VNFs, avec comme contraintes la minimisation des coûts d'exploitation et le respect des exigences QoS. Le problème est modélisé à travers une formulation de programmation linéaire en nombres entiers (*Integer Linear Programming - ILP*). Après avoir démontré la NP-difficulté du problème, nous proposons une solution algorithmique basée sur PageRank.

Le deuxième article de la thèse, intitulé **Towards Reliable Remote Health Monitoring in Fog Computing Networks**, se concentre sur la question de la fourniture de services critiques dans un environnement de réseau non fiable, tel que les réseaux de fog computing. Plus précisément, nous étudions ce problème à travers la mise en place d'une plateforme de suivi médical à distance, dans le contexte de la pandémie COVID-19. Cet article explore les avantages de l'utilisation d'un algorithme d'apprentissage pour résoudre un problème de fourniture de ressources dans un réseau à grande échelle, avec des objectifs critiques en termes de délais à respecter. L'objectif est de maximiser le nombre de tâches de traitement exécutées en respectant les contraintes de délais et de disponibilité. Après avoir montré la NP-difficulté du problème dans le cas général, nous proposons un algorithme basé sur l'apprentissage par renforcement profond (Deep Reinforcement Learning - DRL) et l'évolution différentielle (Differential Evolution - DE).

Le troisième article de cette thèse, intitulé **DRL-based Green Resource Provisioning for 5G Networks**, porte sur la fourniture de ressources virtuelles à grande échelle dans les réseaux mobiles et fixes. Notre approche tient compte de l'impact écologique ainsi que des contraintes de qualité de service. En particulier, cet article se concentre sur la capacité de généralisation d'un algorithme basé sur l'apprentissage par renforcement lorsque l'environnement de réseau est caractérisé par un dynamisme (c'est-à-dire, la mobilité des utilisateurs, l'indisponibilité d'hôtes, la libération de ressources, etc.) et une hétérogénéité notable. Nous proposons des algorithmes d'apprentissage par renforcement auxquels nous associons des algorithmes parallélisables afin d'améliorer les performances d'apprentissage. En parallèle, nous explorons les avantages de la simplification des mécanismes de récompense pour les algorithmes d'apprentissage par renforcement.

Le dernier article de la thèse est intitulé **Market Driven Multi-domain Network Service Orchestration in 5G Networks** et porte sur la fourniture de ressources dans les réseaux multi-domaines, multi-opérateurs, en respectant les contraintes d'exigences de qualité de service et en maximisant les profits. Nous établissons un environnement de libre marché pour les ressources dans lequel différents modèles de coopération sont permis aux opérateurs de réseau, en fonction de leur capacité financière. Nous proposons un algorithme basé sur l'apprentissage par renforcement multi-agents.

Mots clés : fonctions réseau virtuelles, allocation de ressources virtuelles, allocation d'utilisateurs, allocation de fréquences, gestion d'interférences, multi-domaines, mono-domaines, complexité asymptotique, NP-difficulté, algorithmes gloutons, algorithmes compétitifs, algorithmes d'approximation, algorithmes d'apprentissage, apprentissage par renforcement, apprentissage par renforcement multi-agents.

ABSTRACT

The constant evolution of network technology, from fiber optics to cellular networks, has consistently focused on increasing spectral efficiency, thereby enhancing our online experiences. Previously, cell phones were primarily used for sending text messages and making phone calls. However, with the advent of advanced features such as media streaming and complex computing capabilities, the use of these devices has become more widespread and demanding in terms of Quality of Service (QoS) requirements.

The proliferation of innovative applications and services, coupled with the widespread mobility and localization of end users, has had a significant impact on the scale, operational costs, traffic volume and diversity, and access patterns of networks, leading network operators to implement several significant changes in network management.

In recent years, network operators for Next-Generation Networks (NGNs) have shifted their focus from spectral efficiency to low end-to-end latency service delivery, with a goal of achieving latencies of less than 10 milliseconds. It is expected that NGNs will be able to achieve ultra-fast data rates and minimal latency through the integration of various established and emerging technologies, such as 5G, artificial intelligence (AI), edge intelligence, network softwarization, and data-plane programmability. However, due to the increase in system complexity and exponential traffic growth, different management strategies are needed to accommodate the dynamic nature of network configuration and enable end-to-end system automation through distributed learning systems. By adopting learning-based, cooperative, and distributed multi-domain management solutions, end users can benefit from more reliable, efficient, scalable, secure, and diverse services and applications.

This article-based thesis investigates various approaches for the multi-domain orchestration of virtual resources in NGNs, with a focus on the end-to-end provisioning of value-added services that have stringent Quality of Service (QoS) requirements. Given the complexity of the subject, each paper in the thesis examines a subset of the orchestration problem, starting with single-domain networks and progressing to multi-domain networks. The insights gained from prior publications are used to develop innovative strategies for subsequent works.

The first article in this thesis, entitled **CPVNF: Cost-efficient Proactive VNF Placement and Chaining for Value-Added Services in Content Delivery Networks**, addresses the challenge of provisioning virtual resources in a Content Delivery Network (CDN) in order to meet the Quality of Service (QoS) requirements of multimedia applications. This problem is complex due to the presence of multiple content sources in a CDN and the heterogeneity of the virtual network function (VNF) chain that must be configured according to the end-user. To address this issue, the authors present a proactive algorithm for deploying and chaining VNF instances in order to minimize operating costs and meet QoS requirements. The problem is modeled using integer linear programming (ILP) and shown to be NP-hard. To solve the problem, the authors propose a PageRank-based algorithmic solution.

The second article in the thesis, titled **Towards Reliable Remote Health Monitoring in Fog Computing Networks**, focuses on the challenge of providing critical services in an unreliable network environment, such as fog computing networks. Specifically, the authors examine this issue by implementing a remote medical monitoring platform in the context of the COVID-19 pandemic. The research investigates the benefits of using a learning approach to handle a resource provisioning problem in a large-scale network with strict delay goals, with the aim of maximizing the number of processing activities completed despite deadline and availability constraints. The problem is shown to be NP-hard in general, and the authors offer a Deep Reinforcement Learning (DRL) and Differential Evolution (DE)-based approach as a solution.

The third article in this thesis, titled **DRL-based Green Resource Provisioning for 5G Networks**, focuses on large-scale virtual resource provisioning in mobile and fixed networks, taking into account both the ecological impact and quality of service constraints. This paper specifically focuses on the generalizability of a reinforcement learning-based algorithm in a network environment characterized by significant dynamism (such as user mobility, host unavailability, resource release, etc.) and heterogeneity. The authors propose reinforcement learning algorithms with parallelizable components to improve learning performance and investigate the benefits of simplifying reward schemes for reinforcement learning algorithms.

The final article in the thesis is titled **Market Driven Multi-domain Network Service Orchestration in 5G Networks**. It focuses on resource provisioning in multi-domain, multi-operator networks subject to QoS requirements and the goal of profit maximization. The authors establish a free market environment for resources in which different cooperation models are allowed for network operators based on their financial capacity. They propose a multi-agent reinforcement learning-based algorithm to solve this problem.

Keywords: virtual network functions, virtual resource allocation, user allocation, frequency allocation, interference management, multi-domain, single-domain, asymptotic complexity, NP-hardness, greedy algorithms, competitive algorithms, approximation algorithms, learning algorithms, reinforcement learning, multi-agent reinforcement learning.

CHAPITRE I

INTRODUCTION GÉNÉRALE

1.1 Mise en contexte et problématiques

1.1.1 Contexte actuel des réseaux : un aperçu

La pandémie de COVID-19 a mis en évidence l'indéniable hyperconnectivité de notre monde actuel ainsi que son importance, lorsqu'elle a conduit à une migration en ligne de nombreuses entreprises et services essentiels, révolutionnant dans un court laps de temps notre accès à divers services allant de l'enseignement aux soins médicaux. Les technologies de l'information et de la communication (TIC) ont un impact considérable sur l'ensemble de nos activités quotidiennes grâce à des innovations révolutionnaires qui ont touché différents secteurs de l'économie et de la société, donnant naissance à de nouvelles industries.

Cette situation modifie radicalement les relations entre individus (ou consommateurs/citoyens), entreprises et gouvernements et offre de nouvelles opportunités pour augmenter la productivité et le bien-être en réinventant les processus commerciaux et en améliorant la qualité des services publics. Selon le Forum économique mondial (*World Economic Forum* 2022), le secteur des TIC représente 25% de la croissance du Produit Intérieur Brut (PIB) de l'Union européenne et 40% de la croissance de sa productivité. Les avantages ne sont pas uniquement économiques, mais aussi socio-politiques, car les TIC favorisent la transparence et l'efficacité des pouvoirs publics, réduisent le chômage, améliorent la qualité de vie et renforcent l'accès des citoyens aux services publics (*World Economic Forum* 2022).

L'hyperconnectivité est favorisée par plusieurs facteurs, tels que la prolifération croissante d'appareils mobiles (et en réseau) à la performance et aux capacités de calcul de plus en plus élevées, l'accès sans fil de plus en plus répandu, la prédominance des médias sociaux dans la vie quotidienne et l'utilisation croissante des Big Data. L'une des conséquences les plus importantes de cette augmentation de la quantité et de la diversité des appareils connectés au réseau est l'accroissement exponentiel des données et du trafic de données, au point que l'univers numérique est prévu de doubler tous les deux ans. Cette explosion du volume de données est également causée par l'apparition de services et d'applications innovants et dynamiques tels que la diffusion en continu multimédia, le stockage infonuagique, les jeux en ligne, la réalité augmentée (AR) et la réalité virtuelle (VR). Ces services et applications connectent, par le biais du réseau, de nombreux aspects de la société, de l'infrastructure critique comme la santé en ligne, les transports et les systèmes de réseaux électriques aux maisons intelligentes (*World Economic Forum* 2022).

Les opérateurs de réseau sont confrontés à un défi de taille en devant assurer l'approvisionnement de services exigeants en termes de QoS, notamment une capacité massive, une latence presque imperceptible et une très haute fiabilité. Ces services sont également très gourmands en données et centrés sur l'utilisateur, avec des utilisateurs finaux dispersés à travers le monde via des appareils intelligents diversifiés. En outre, les exigences de QoS des utilisateurs du réseau deviennent de plus en plus diversifiées et personnalisées.

Avec la croissance rapide et significative des réseaux, les opérateurs de réseau ont davantage du mal à suivre le rythme. Cela est principalement dû à l'ossification de l'infrastructure réseau, un phénomène qui décrit la rigidité des réseaux traditionnels qui ont tendance à être configurés de manière statique et prédéfinie, ce qui limite le développement et l'approvisionnement de services et d'applications innovants. En effet, l'approche traditionnelle de création, déploiement et gestion des services de bout en bout implique souvent des opérations manuelles et fastidieuses qui sont entravées par des infrastructures réseau limitées par des topologies physiques et des contraintes matériel (De Sousa et al., 2018). Dans un environnement où chaque fabricant de matériel offre des interfaces et des schémas de configuration propriétaire, il est nécessaire pour les opérateurs de réseau de faire appel à du personnel qualifié, dont le niveau de compétence est proportionnel à la complexité des services. Cela entraîne une augmentation des coûts et une potentielle diminution de la qualité de l'expérience utilisateur finale (Boucadair et al., 2015).

Dans l'optique de contrôler l'explosion du volume de données et garantir les exigences de qualité de service dans cet environnement, les opérateurs de réseaux ont traditionnellement eu recours à des méthodes de hiérarchisation strictes du trafic, telles que la limitation de données, l'étranglement du trafic, le filtrage et l'utilisation de seuils de données. Cependant, cette approche de gestion des ressources du réseau s'est révélée peu efficace, peu flexible et coûteuse car elle ne permet pas d'ajuster de manière dynamique l'allocation et la configuration des ressources en fonction des exigences des utilisateurs et des services en cours.

La complexité croissante des réseaux amplifie le défi d'une gestion adéquate, en partie en raison du fait que les mécanismes de contrôle et de configuration actuels sont principalement basés sur des interventions humaines qui ne sont pas assez évolutifs ou durables pour gérer de manière efficace de tels systèmes complexes. Afin d'améliorer les flux de trafic et bien gérer les classes et priorités de trafic, des investissements importants dans l'infrastructure du réseau et de nouvelles approches de gestion sont nécessaires. Par exemple, l'automatisation de différentes entités et fonctions des réseaux est l'une des principales préoccupations majeures dans le but de réduire les dépenses opérationnelles. Elle permettrait d'améliorer l'efficacité, la rapidité et la précision des processus, ainsi qu'à réduire les coûts en supprimant la nécessité de main-d'œuvre humaine pour certaines tâches. Cependant, il est important de gérer correctement l'intégration de l'automatisation afin de minimiser les risques de défaillance et d'assurer la continuité des services. En outre, il y a une prise de conscience croissante de l'impact environnemental des réseaux, en particulier lorsque l'on considère que l'industrie des télécommunications est responsable d'environ 4 % des émissions totales de CO2 dans le monde (ISMAIL et al., 2015). Par conséquent, on constate une poussée vers des technologies plus vertes d'autant plus que pour les opérateurs, les coûts énergétiques représentent une part importante de leurs coûts annuel d'exploitation, allant de 18% à 50% (ISMAIL et al., 2015).

En réponse aux préoccupations ci-dessus, les parties prenantes du réseau ont entrepris de profondes transformations, notamment en modernisant les infrastructures et en introduisant de nouvelles technologies et paradigmes innovants. Cela a entraîné des changements radicaux dans les rôles traditionnels des acteurs du réseau (opérateurs, fournisseurs de services ou d'infrastructures, etc.) et périréseau (vendeurs de solutions, intermédiaires client-opérateur, etc.), ainsi que dans les mécanismes de gestion du réseau et de fourniture (c.-à-d, l'allocation, le déploiement, la distribution et la gestion) de services.

Il est indéniable que ces transformations entraîneront de nouveaux défis à relever. En particulier, les NGNs (c-à-d. *Next Generation Networks*) et, de manière plus spécifique, la cinquième génération de réseaux mobiles (5G) doivent s'adapter à de nouveaux services et applications dont l'approvisionnement ne se limite pas uniquement à la quantité de ressources réseaux disponibles. En effet, chaque nouvelle génération de réseaux sans fil a permis de dépasser de nouveaux records en termes de débit et de capacité de données. Cependant, il serait trop simpliste de ne considérer l'approvisionnement de nouveaux services que sous cet angle et de ne pas tenir compte de facteurs supplémentaires.

Un des changements de paradigme avec les NGNs est que les services vocaux, de données et multimédias convergeront vers une seule et même plateforme de réseau complexe et hétérogène combinant des systèmes de réseaux sans fil et optiques, ainsi que des plateformes de développement et traitement logiciel. Cela s'appelle la *convergence des réseaux* qui, en particulier celle fixe-mobile, permet d'étendre la portée des réseaux en favorisant l'interopérabilité entre différents segments de réseaux. Elle entraîne donc une augmentation du nombre d'acteurs du réseau, incluant des opérateurs (virtuels), des fournisseurs de services et d'infrastructure (virtuels) ainsi que des utilisateurs finaux attirés par l'émergence d'applications de nouvelle génération. Elle permet également de simplifier la gestion des ressources et de réduire les coûts d'exploitation et d'opérations, ce qui crée un environnement de concurrence féroce pour les opérateurs et les fournisseurs de services. En raison de cette concurrence, les recettes des opérateurs ont continué à diminuer ces dernières années pour deux raisons conjointes : (1) l'augmentation de la demande de ressources pour les utilisateurs finaux, qui entraîne une augmentation des dépenses opérationnelles, et (2) l'impossibilité d'augmenter les tarifs des abonnés sans nuire à leur attractivité (Mijumbi et al., 2016).

Il est donc crucial pour les opérateurs de réseaux de maintenir et renforcer leur avantage concurrentiel en fournissant de manière rentable et efficace les services et applications nécessaires, tout en répondant aux exigences strictes des utilisateurs du réseau. Cela implique la prise de décisions de gestion et de contrôle optimales de bout en bout du réseau pour chaque client, en tenant compte du protocole de transport IP basé sur le principe du *meilleur effort*, de la nécessité pour les opérateurs de s'adapter rapidement aux changements des exigences des utilisateurs, des conditions du réseau, de l'état de l'infrastructure et du réseau, et des objectifs commerciaux, ainsi que de l'augmentation des données semi-structurées et non-structurées nécessitant des mécanismes de traitement dans le réseau.

Cela dit, un des principaux obstacles à l'interopérabilité réside dans l'absence de normes techniques unifiées entre les différents secteurs du réseau, ainsi que dans la réticence de certains opérateurs à céder leur monopole. De plus, l'hétérogénéité et la complexité des réseaux imposent la mise au point de systèmes et algorithmes de contrôle et de gestion spécifiques et efficaces pour gérer des scénarios complexes en fonction des caractéristiques de segments du réseau et des demandes des utilisateurs. Dans ces circonstances, certaines décisions de contrôle et de gestion devront être prise à la périphérie du réseau (*network edge*) afin d'éviter les délais et la consommation de ressources excessifs liés à l'acheminement et au traitement d'une grande quantité de données vers un emplacement centralisé comme dans l'infonuagique (*cloud computing*). Ainsi, l'*intelligence réseau* doit être étendue jusqu'à la périphérie du réseau, en coordonnant tous les autres segments de réseau comprenant les réseaux d'accès radio et de transport, et les ressources de traitement. Ces décisions doivent être prises de manière rapide, fiable et surtout en évitant d'introduire des conflits ou instabilités dans le réseau global(Velasco et al., 2021).

Un autre changement de paradigme est la logiciellisation des réseaux (*network softwarization*), résultante de l'avènement de la virtualisation des ressources. Elle implique la transformation d'une infrastructure réseau composée de dispositifs dédiés et de logiciels propriétaires en l'utilisation de ressources matérielles et logicielles ouvertes, afin de s'adapter de manière flexible et dynamique aux évolutions du marché et aux besoins des utilisateurs. La logiciellisation du réseau permet d'augmenter l'utilisation et la flexibilité des ressources, d'accélérer considérablement le déploiement de services et d'améliorer les capacités de traitement et l'interopérabilité entre les différentes parties prenantes du réseau.

La virtualisation du réseau est une technique qui permet de découpler les infrastructures et les services dans les réseaux conventionnels, en créant deux nouvelles entités : le fournisseur de service (SP) et le fournisseur d'infrastructure (InP). Dans ce modèle, les InPs sont propriétaires de leurs réseaux physiques et sont responsables de l'attribution d'une partie de ces réseaux à leurs SPs clients, en échange d'une rémunération. Pour ce faire, ils incorporent sur un ou plusieurs réseaux physiques un réseau virtuel qui consiste en une collection de nœuds virtuels et de liens virtuels. Ces nœuds et liens virtuels sont principalement caractérisés par leur capacité de traitement et de bande passante respectivement, et peuvent également prendre en compte d'autres ressources telles que la mémoire et le stockage (Belbekkouche et al., 2012).

La virtualisation de réseau permet la coexistence de plusieurs réseaux virtuels sur un seul réseau physique, offrant ainsi la possibilité de fournir des services de bout en bout sans nécessiter de protocoles, d'applications, de contrôles et de plans d'administration standardisés. Cela permet à chaque InP de gérer de manière autonome sa propre infrastructure physique tandis que chaque SP peut se concentrer sur la fourniture et l'amélioration de services de bout en bout pour les utilisateurs finaux. La virtualisation de réseau permet également la récursion et l'héritage de propriétés entre réseaux virtuels. Elle vise également à améliorer la flexibilité, la gestion, l'évolutivité, l'isolement, la stabilité et la convergence des réseaux, ainsi que la gestion de l'hétérogénéité. De plus, elle peut être utilisée pour partager l'infrastructure réseau entre les opérateurs, ce qui permet de réduire les coûts de déploiement du réseau, d'offrir des services moins chers aux utilisateurs finaux, d'accélérer le déploiement de services et de permettre une gestion efficace des ressources. Cependant, la virtualisation de réseau peut également poser de nouveaux défis en termes de gestion de la qualité de service, de sécurité et de confidentialité des données, ainsi que de gestion de la complexité et de l'intégration des réseaux virtuels (Devlic et al., 2017).

1.1.2 NGNs et 5G : De nouveaux types de services et applications

La 5G vise à révolutionner l'exploitation et la gestion des réseaux en proposant des mécanismes de communication programmables, hautement flexibles et robustes, sur une infrastructure de réseau et de traitement convergente, hétérogène, intelligente et à accès ouvert. Elle permet l'émergence de nouveaux modèles commerciaux qui favorisent l'entrée sur le marché de secteurs verticaux et de fournisseurs de services tiers. Par ailleurs, la 5G offre une nouvelle génération de services et d'applications innovants et riches en contenus multimédias, qui nécessitent une prise en charge adéquate de divers profils de trafic avec un ensemble de métriques de performance/QoS (*Quality of Service*) distinct.

Selon le standard IMT-2020, trois catégories de services sont prévues pour la 5G, chacune correspondant aux exigences de performance nécessaires au bon fonctionnement des services et applications de nouvelle génération : eMBB (*enhanced mobile broadband*), mMTC (*massive machine type communications*) et URLLC (*ultra-reliable low-latency communications*) (Saad et al., 2020). En bref, la 5G représente donc une transformation fondamentale de l'architecture traditionnelle orientée données vers une architecture plus flexible et orientée service capable de prendre en charge une grande variété de services aux exigences QoS diverses sur un ensemble commun de ressources de réseau physique.

Les réseaux NGNs décrivent les réseaux futurs conçus pour desservir une grande variété d'applications et de services, allant de la chirurgie à distance à la conduite autonome en passant par les villes intelligentes, caractérisés par des exigences de performance strictes et dynamiques. La satisfaction de ces exigences requiert une prise en charge personnalisée, de bout en bout, de mécanismes de communication, de contrôle et de traitement dans divers segments hétérogènes de réseau, de traitement et de stockage (accès, transport, cœur, *fog/cloud/edge*, etc.) (Saad et al., 2020 ; Baranda et al., 2018). Cela a conduit à l'émergence du concept de découpage de réseau (network slicing en anglais), qui permet de créer plusieurs réseaux virtuels indépendants sur une même infrastructure physique de réseau, chacun étant conçu pour répondre aux exigences spécifiques d'un service ou d'une application. Le découpage de réseau permet une meilleure personnalisation et une utilisation plus efficace des ressources de réseau, en offrant une meilleure flexibilité et une qualité de service améliorée grâce à des garanties de performance dédiées pour chaque service.

Nous résumons les exigences de performance de chaque catégorie de services dans les tableaux 1.1 et 1.2.

	5G	NGNs
Types d'applications	• eMBB • URLLC • mMTC	• Reliable eMBB • URLLC • mMTC • Hybrid(URLLC + eMBB) • MBRLLC • mURLLC • HCS • MPS
Types d'appareils	• Téléphones intelligents • Capteurs • Drones	• Téléphones intelligents • Capteurs • DLT (Distributed Ledger Technology) • Drones • Appareils XR • Appareils BCI (Brain-computer interactions) • CRAS (Connected Robotics and Autonomous Systems) • Implants intelligents
Gain en termes d'efficacité spectrale et énergétique	$\times 10 - bps/Hz/m^2/Joules$	$\times 100$ à $1000 - bps/Hz/m^2/Joules$
Exigences pour le débit	1 Gb/s	100 Gb/s à 1 Tb/s
Exigences pour le délai de bout en bout	5 ms	$\leq$ 1ms
Exigences pour le délai de traitement	100 ns	10 ns - 50 ns
Exigences pour la fiabilité de bout en bout	99.9999 %	99.9999 % - 99.99999 %
Bandes de fréquences	• Sub-6 GHz (c.-à-d. fréquences en dessous de 6 GHz) • mmWave	• Sub-6 GHz • MmWave • THz • Non-RF (par ex. liaisons optiques, VLC, etc.)
Architecture	• Petites stations de base denses sub-6 GHz avec stations de base macro • Petites cellules mmWave d'environ 100 m	• Petites stations de bases plus denses sub-6 GHz avec stations de base macro • < Petites cellules mmWave denses de moins de 100m • Surfaces intelligentes soutenues par de minuscules cellules mmWave • Essais de petites cellules THz • Points d'accès temporaires desservis par des stations de base sur drones ou ballons captifs.

TABLE 1.1 Exigences QoS pour les services et applications prévues pour la 5G et les NGNs, extraites de (SAAD et al., 2020)

Types de services et applications	**Indicateurs de performance**
eMBB	• Hauts débits • Gros volume de données • Connectivité massive • Faible latence (meilleur effort)
URLLC	• Latence très faible • Fiabilité et disponibilité très élevées
mMTC	• Nombre massif de dispositifs à faible coût, avec batterie à longue durée de vie • Efficacité énergétique
Reliable eMBB	• eMBB avec fiabilité et disponibilité très élevées
Hybrid (URLLC + eMBB)	• eMBB + URLLC • Précision de la localisation
MBRLLC	• Exigences strictes de débit, de fiabilité et de latence • Efficacité énergétique
mURLLC	• Ultra haute fiabilité • Connectivité massive • Fiabilité massive • URLLC évolutive
HCS	• Qualité de l'expérience physique via capture de métriques sans fils et dépendant de facteurs humains et physiques.
MPS	• Contrôle de la stabilité • Délai de traitement • Précision de la localisation • Précision de détection et de cartographie • Fiabilité et délai de communications • Efficacité énergétique

TABLE 1.2 Résumé des classes de services avec leurs indicateurs de performance, extrait de (SAAD et al., 2020)

1.1.3 Partage des responsabilité d'investissements et impacts

Le succès phénoménal de l'Internet a eu pour conséquence de mettre à rude épreuve la capacité des réseaux et les revenus des opérateurs chargés de leur maintenance. En réponse, les opérateurs de réseaux mettent en place diverses stratégies pour gérer les volumes de trafic et protéger contre les attaques malveillantes, y compris l'utilisation de technologies avancées telles que l'inspection approfondie de paquets pour analyser le contenu du trafic et identifier les éventuelles menaces pour les performances du réseau. Cependant, certains ont argumenté que ces mesures pourraient être perçues comme en conflit avec le principe de neutralité du réseau, qui prône un traitement égal de tout le trafic internet et interdit aux fournisseurs de services internet de discriminer ou de manipuler la livraison de certains types de trafic (*World Economic Forum* 2022).

Le développement de l'internet et de ses principes, y compris la neutralité du réseau, s'est déroulé à une époque où les fournisseurs de services internet pouvaient répondre à la demande de trafic internet sans dépenser de capitaux considérables et où les services internet ne constituaient pas une menace pour les flux de revenus essentiels des entreprises responsables de l'infrastructure physique de l'internet. Cependant, l'état actuel de l'industrie est très différent, avec l'augmentation rapide du trafic internet mettant la pression sur les opérateurs pour augmenter la capacité du réseau (*World Economic Forum* 2022).

Pour faire face à ces défis, les fournisseurs d'accès doivent investir du capital pour fournir les services différenciés et la capacité accrue nécessaires aux fournisseurs de services tiers comme Google ou Facebook et aux utilisateurs finaux. En même temps, les opérateurs font face à une baisse de leurs revenus et de leurs bénéfices, en particulier à cause des services innovants proposés par les fournisseurs de services tiers. Ces investissements ne seront réalisés que si des rendements raisonnables peuvent être générés étant donné que la mise à niveau de l'infrastructure fixe vers des réseaux NGNs est coûteuse, avec des estimations allant de 200 à 250 milliards d'euros pour les pays de l'UE-15 (*World Economic Forum* 2022).

Du point de vue économique, l'industrie des télécommunications affronte une pléthore de défis, notamment la nécessité de sécuriser le financement des mises à niveau d'infrastructure nécessaires et la question de qui devrait supporter les coûts de ces améliorations. Le débat est toujours en cours sur la question de savoir si fournisseurs de services tiers devraient contribuer à ces dépenses. Pour illustrer ce point, fin 2005, Ed Whitacre de SBC (maintenant AT&T) a commenté les perspectives d'affaires des entreprises telles que Google en déclarant :

> "Maintenant, ce qu'ils aimeraient faire, c'est utiliser mes tuyaux gratuitement, mais je ne vais pas les laisser faire parce que nous avons dépensé ce capital et nous devons avoir un retour sur investissement. Il va donc falloir qu'il y ait un mécanisme pour que ces gens qui utilisent ces tuyaux paient pour la partie qu'ils utilisent. Pourquoi devraient-ils être autorisés à utiliser mes tuyaux ? L'Internet ne peut pas être gratuit dans ce sens, parce que nous et les câblo-opérateurs avons fait un investissement, et pour un Google ou Yahoo ! ou Vonage ou n'importe qui d'autre, s'attendre à utiliser ces tuyaux gratuitement, c'est de la folie".

Cette déclaration met en lumière la complexité de la question de la neutralité du réseau et les défis auxquels sont confrontées les entreprises de télécommunications pour sécuriser un rendement juste sur leur investissement pour l'infrastructure qu'elles fournissent. Elle souligne également le fait que les fournisseurs de services internet ont investi des sommes considérables dans l'infrastructure nécessaire pour fournir l'accès à internet et qu'ils ont le droit de récupérer ces coûts et de générer des profits raisonnables grâce à cet accès. Cependant, cette question est controversée et fait l'objet de débats en cours sur la manière dont les fournisseurs de services internet devraient être rémunérés pour l'accès à leur infrastructure et sur la manière dont le trafic internet devrait être géré pour garantir une expérience utilisateur satisfaisante pour tous les utilisateurs.

Cette déclaration met également en lumière l'importance croissante de la profitabilité pour les opérateurs dans la fourniture de services. Si le principe de neutralité du réseau n'est plus le fondement sur lequel reposent le fonctionnement des réseaux, l'approvisionnement de ces services dépendra davantage des impératifs économiques et de la capacité des opérateurs à générer des revenus raisonnables à partir de leur fourniture. Cela peut être particulièrement vrai dans le cas de services qui sont caractérisés par des exigences QoS strictes et qui sont gourmands en ressources réseau. En d'autres termes, les opérateurs seront davantage enclins à fournir des services à des tarifs différenciés et à des utilisateurs prêts à payer pour des services de qualité supérieure, plutôt que d'offrir des services "à coût constant" à tous les utilisateurs. Cela peut entraîner une divergence des services offerts par les opérateurs et une certaine inégalité dans l'accès aux services internet en fonction de la capacité des utilisateurs à payer pour des services de qualité supérieure.

1.1.4 Le découpage en tranches du réseau

Comme souligné aux tableaux 1.1 et 1.2, les réseaux modernes doivent être en mesure de gérer une grande variété de services et d'applications, qui peuvent avoir des exigences très différentes en termes de qualité de service (QoS) et de criticité. Pour cette raison, les réseaux doivent être flexibles et capables de fournir des performances de qualité constante, quelles que soient les exigences des services et applications en question.

Le découpage en tranches du réseau (également connu sous le nom de *network slicing*) utilise la virtualisation de réseau pour créer des partitions logiques, indépendantes et entièrement fonctionnelles, adaptées à un type de service spécifique à partir d'une infrastructure commune de réseau, de stockage et de traitement. En partageant la même infrastructure, les coûts d'exploitation du réseau peuvent être considérablement réduites. Chaque tranche représente un réseau virtualisé isolé, de bout en bout, ce qui permet aux opérateurs de mettre en place différentes approches d'exploitation et de gestion pour une utilisation efficiente des ressources et pour garantir la satisfaction des exigences QoS des services, en particulier pour ceux ayant des implications critiques, comme la conduite autonome. Le concept de tranche est donc une abstraction naturelle permettant de séparer plusieurs locataires qui utilisent les mêmes ressources physiques.

Le découpage en tranches du réseau est une technique qui s'appuie sur la virtualisation des fonctions de réseau (NFV) et sur les réseaux définis par logiciel (SDN) pour gérer et contrôler de façon programmable les ressources et fonctions du réseau, qu'elles soient physiques ou virtuelles.

NFV permet de déployer sur demande des fonctions de réseau virtualisées (Virtual network functions - VNFs). Ces fonctions, qui incluent des éléments tels que des pare-feux, des équilibreurs de charge et des routeurs, sont habituellement implémentées à l'aide de dispositifs matériels spécialisés, comme des appareils ou des systèmes propriétaires. En décomposant ces fonctions en composants virtuels plus petits et modulaires exécutés sur des serveurs standards, NFV permet aux fournisseurs de services de déployer et de mettre à l'échelle rapidement et facilement leurs fonctions de réseau, tout en pouvant adapter leur réseau aux besoins spécifiques de différents services ou utilisateurs.

SDN vise à établir une séparation nette entre la couche de contrôle du réseau et celle de transport de données. Cela est réalisé à travers l'abstraction de l'infrastructure de réseau sous-jacente et le déplacement de l'intelligence de réseau vers un contrôleur logiciel centralisé, ce qui permet une programmabilité complète des fonctionnalités d'acheminement et des capacités de prise de décision améliorées sur la base d'une vue globale du réseau. Le contrôleur peut facilement être configuré et modifié pour répondre aux exigences changeantes de l'environnement réseau. SDN offre également une interface standard ouverte pour la communication entre le contrôleur et l'équipement de la couche de transport de données, ce qui augmente la flexibilité et la programmabilité du réseau, une meilleure utilisation des ressources de réseau, une visibilité accrue et un meilleur contrôle sur le trafic de réseau, tout en simplifiant les éléments de la couche de transport de données. Cependant, il convient de noter que l'utilisation d'un contrôleur centralisé présente des inconvénients tels qu'une dépendance excessive de chaque nœud envers le contrôleur, la grande quantité d'informations à traiter par le contrôleur et un délai supplémentaire pour la création de garanties de service (QoS) spécifiques, étant donné que les paquets doivent attendre la réponse du contrôleur avant d'être acheminés.

Par souci de concision, nous renvoyons le lecteur intéressé par les concepts de SDN et de NFV aux références (Ietf, 2018 ; Bernardos et al., 2019 ; Fischer et al., 2013 ; Toumi et al., 2021 ; Barakabitze et al., 2020 ; Alam et al., 2020 ; Benzekki et al., 2016 ; Herrera et Botero, 2016 ; Hawilo et al., 2014) pour en savoir plus.

Les avantages de NFV dans le contexte du network slicing incluent notamment :

- la simplicité de création de services grâce au concept de chaînage de fonctions virtuelles (Service Function Chaining - SFC), qui permet de gérer de manière efficace la livraison, le contrôle et la surveillance de services spécifiques en chaînant de manière séquentielle plusieurs VNFs ;

- la possibilité de consolider les ressources pour réduire le surdimensionnement et la surallocation de ressources réseaux, ainsi que d'améliorer l'efficience énergétique ;
- l'orchestration dynamique des VNF en fonction des besoins actuels du service grâce à un processus d'instanciation et de placement automatisé, ainsi que la possibilité de reconfigurer les réseaux virtuels en fonction des changements de charges de travail et de besoins.

Ces avantages sont importants d'autant plus les tranches de réseau allouées peuvent couvrir de grandes zones géographiques, compte tenu de la mobilité et de la dispersion des utilisateurs finaux. Cependant, il est à noter que la fragmentation progressive du marché des télécommunications a entraîné l'apparition d'une multitude d'opérateurs de réseau, chacun axé sur des pays et régions spécifiques. Cette situation rend pratiquement inexistante l'existence d'une infrastructure à l'échelle continentale ou couvrant plusieurs pays. Dans ce contexte, la couverture, et par conséquent l'approvisionnement en applications et services, ne peut être garantie qu'en consolidant les ressources de différents opérateurs de réseau. L'approvisionnement se fait donc dans un contexte multi-domaine où il est crucial de prendre en compte les défis liés à l'allocation efficace des ressources de réseau parmi les tranches réseau pour optimiser les performances, en tenant compte du dynamisme et de la diversité des exigences QoS de ces tranches et de la limitation inhérente des ressources de réseau. Cela nécessite de considérer une gamme de contraintes, de compromis et de limitations pour prendre des décisions éclairées en matière d'allocation de ressources.

Definition 1.1.1 (Réseau multi-domaine)**.** Le concept de réseau multi-domaine fait référence aux réseaux convergents qui intègrent différents réseaux, domaines, opérateurs et fournisseurs, ce qui est considéré comme étant crucial pour les réseaux

NGNs. Un domaine réseau peut être défini comme une ensemble d'éléments du réseau gérés de manière autonome par un opérateur réseau.

Dans un environnement multi-domaine, la mise en œuvre de telles tranches est complexe non seulement du point de vue de la décomposition de la demande de tranche en domaines respectifs, mais aussi pour assurer le maintien de leurs performances. Cela, d'autant plus que déterminer la granularité optimale d'une tranche réseau, c'est-à-dire savoir s'il doit y avoir une tranche pour chaque type de service, chaque critères QoS, chaque utilisateur ou une combinaison de ces éléments, est un problème ouvert. En plus, il existe généralement plusieurs options pour décomposer un service ou application à travers le SFC en fonctions virtuelles plus fines et interconnectées. Il est donc nécessaire de mettre en œuvre une combinaison efficace de ressources fédérées, non seulement pour fournir les ressources requises (par exemple, espace de stockage ou bande passante), mais aussi pour faire face aux contraintes additionnelles (par exemple, la latence ou le jitter) et aux contraintes multiplicatives (par exemple, la probabilité de taux d'erreur de bout en bout) à travers plusieurs domaines administratifs. Il y a également les contraintes de l'opérateur réseau qui souhaite minimiser les coûts d'exploitation à tenir en compte (Shen et al., 2020 ; Taleb et al., 2019).

À un niveau plus granulaire, le déploiement de SFCs dans un réseau multi-domaine impliquent que des VNFs soient déployés sur différents serveurs répartis dans plusieurs domaines différents. Tout d'abord, il y a la question du routage à prendre en compte pour s'assurer que les flux de données sont dirigés vers les domaines appropriés. Toutefois, contrairement à un réseau monodomaine, les opérateurs dans un environnement multi-domaine ne disposent généralement pas d'une vue d'ensemble de la topologie, car ils sont souvent réticents à divulguer des informations relatives à leur domaine ou infrastructure, telles que la disponibilité des ressources et les dynamiques de trafic, à des opérateurs potentiellement concurrents (C. Zhang et al., 2019). Par conséquent, chaque opérateur ne dispose que d'une visibilité limitée sur son propre réseau. Ainsi, l'opérateur de chaque domaine ne dispose que d'une visibilité limitée sur son propre réseau. Un mécanisme de coopération entre opérateurs est donc nécessaire pour l'approvisionnement de services multi-domaines.

Un autre problème est que les VNFs qui composent la chaîne de VNF peuvent dépendre des utilisateurs finaux et, par conséquent, peuvent être configurées de manière différente. Cela est pertinent dans la mesure où les VNFs ne soient pas entièrement génériques et qu'elles soient exécutées sur différentes plateformes matérielles qui peuvent ou non influencer leur comportement ajoute une complexité supplémentaire. Même si en théorie cela ne devrait pas avoir d'importance, car elles sont virtualisées, l'incertitude demeure quant à l'influence éventuelle de la plateforme matérielle sur le comportement des VNFs. Cela peut rendre difficile la prédiction et la gestion des performances des chaînes de VNF dans un environnement multi-domaine. De plus, le processus de virtualisation d'une fonction réseau peut entraîner une augmentation du délai de bout en bout observé en raison de l'ajout de couches supplémentaires de virtualisation requises. De même, l'emplacement des VNFs peut également avoir un impact significatif sur le délai de bout en bout, en particulier lorsqu'elles sont placées à des emplacements qui promettent des coûts de ressources faibles mais qui sont éloignés des utilisateurs.

Un défi supplémentaire consiste à pouvoir s'adapter à des environnements de réseau dynamiques dans le temps en raison de la mobilité des utilisateurs, des conditions de canal variables, de la distribution de charge de trafic en constante évolution et de la variabilité temporelle de la popularité du contenu (Shen et al., 2020).

Tout cela représente la problématique de l'orchestration de ressources et de fonctions virtuelles dans les réseaux multi-domaines.

Definition 1.1.2 (Orchestration)**.** L'orchestration de ressources et de fonctions virtuelles dans un réseau multi-domaine consiste en l'ensemble des décisions prises tout au long du cycle de vie d'une tranche de réseau (ou d'un service ou d'une application) déployée dans un réseau multi-domaine, dans le but de répondre aux exigences QoS. Elle inclut la planification, la mise en œuvre et le suivi des ressources et fonctions nécessaires pour garantir que les exigences QoS sont satisfaites

1.2 Orchestration de ressources dans les réseaux multi-domaines

1.2.1 Dynamisme et instabilité

Avant l'adoption du paradigme de logiciellisation, les réseaux étaient généralement considérés comme une couche relativement stable sur laquelle le trafic était acheminé, avec des changements topologiques lents au fil du temps. Effectivement, la logiciellisation des réseaux, notamment par le biais de la virtualisation des ressources, a permis l'émergence de réseaux dynamiques comprenant des ressources hétérogènes de traitement, de stockage et de communication agrégées de différents domaines et réseaux pour soutenir la prestation de n'importe quel service.

Ces réseaux présentent un comportement variable et susceptible de variations rapides et importantes, notamment en termes de modification de la topologie des liens et des nœuds sur des échelles de temps courtes (quelques secondes ou minutes), ainsi que de fluctuations du trafic et du routage pouvant entraîner des instabilités qui peuvent avoir des conséquences néfastes sur la performance du système et sont particulièrement préoccupantes compte tenu des exigences de qualité de service de bout en bout qui s'appliquent aux services et applications de nouvelle génération (Clayman et al., 2012). De plus, en raison de la mobilité et de la grande dispersion des utilisateurs finaux, ces exigences de qualité de service sont également sujettes à des variations dynamiques, ce qui ajoute une couche supplémentaire d'instabilité.

Plus spécifiquement, garantir les exigences QoS réseau de bout en bout nécessite que les fonctions de transport et les configurations de paramètres au niveau application soient adaptatives par rapport à d'autres facteurs pouvant fluctuer, tels que les capacités des terminaux, les propriétés des applications et les caractéristiques de l'environnement physique de l'utilisateur, ce qui rend plus difficile la garantie des exigences QoS de bout en bout. Par exemple, dans les communications sans fil, la propagation multipath et le shadowing entraînent des conditions de canal sans fil changeantes de manière dynamique, ce qui aura un impact significatif sur la force du signal reçu par l'utilisateur et, par conséquent, sur la QoS au niveau du réseau (J. Zhang et Ansari, 2011).

Dans ce contexte, l'opérateur de réseau a besoin d'un système d'orchestration de ressources intelligent capable de prendre et d'ajuster adéquatement des décisions d'orchestration à tous les niveaux de la pile de protocoles de chaque domaine traversé afin de satisfaire les demandes de qualité de service de tous les utilisateurs. Plus spécifiquement, les fonctions telles que le contrôle d'admission, la sélection de réseau d'accès, le routage, l'allocation de ressources, le contrôle de transmission et le codage de source doivent être capables de s'adapter aux exigences de qualité de service dynamiques des utilisateurs finaux (J. Zhang et Ansari, 2011).

En même temps, l'opérateur de réseau se trouve dans une situation où toute nouvelle décision d'orchestration prise pour assurer une garantie continue des exigences de qualité de service, pour répondre aux événements topologiques des réseaux (par exemple en cas de panne ou d'expiration de l'utilité d'un service) ou pour minimiser les coûts d'exploitation pourrait également entraîner de l'instabilité et déclencher une cascade de ré-orchestration coûteuse et complexe.

La figure 1.1 présente les différentes sources d'instabilités dans les réseaux en raison de leur dynamisme selon (Clayman et al., 2012). Il existe trois grandes classes :

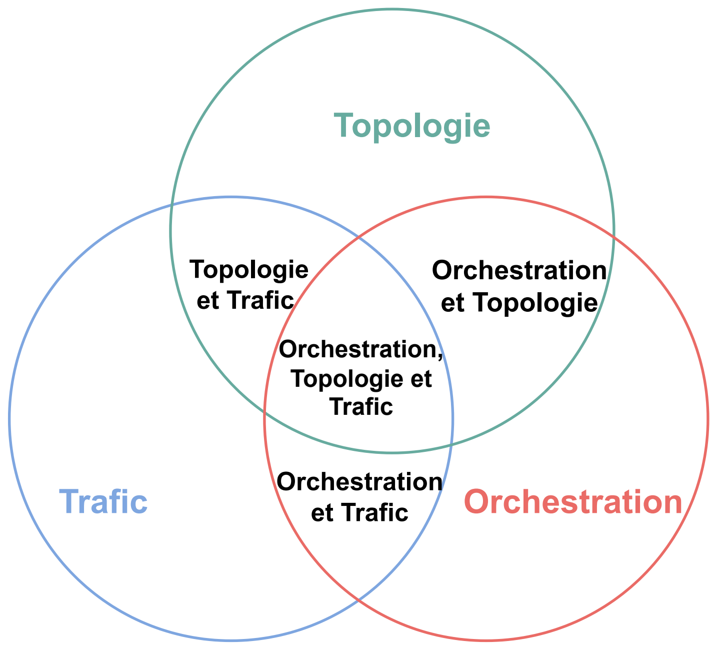

FIGURE 1.1 Interactions entre les sources d'instabilité dans un réseau dynamique (CLAYMAN et al., 2012)

- **Topologie :** Une instabilité topologique se réfère à l'impact sur les performances du réseau suite à l'ajout, l'indisponibilité ou la suppression rapide de nœuds ou de liens (physiques ou virtuels). Souvent, ces opérations font partie intégrante du fonctionnement normal d'un réseau dynamique et sont sans danger.

- **Trafic :** Une instabilité du trafic fait référence aux changements de volume de trafic sur un ou plusieurs liens de réseau, qui peuvent être causés par des fluctuations importantes de la demande (par exemple, un pic de trafic soudain), des instabilités de protocole ou des interactions de protocole normales telles que les interactions TCP.

- **Orchestration :** Une instabilité d'orchestration se réfère aux changements résultant d'une mauvaise décision d'orchestration. Par exemple, une mauvaise décision de routage pourrait entraîner des pertes de paquets, entraînant un renvoi massif de paquets et nuisant ainsi à la performance du réseau.

Il est important de noter que ces différents types d'instabilité ont tendance à interagir entre eux de manière bidirectionnelle, c'est-à-dire que l'un peut influencer l'autre et réciproquement. Par exemple, une instabilité au niveau de la topologie des liens et des nœuds peut entraîner des fluctuations du trafic et du routage, qui à leur tour peuvent affecter la stabilité de la topologie :

- **Orchestration et Trafic** : L'instabilité orchestration et trafic se réfère à l'impact néfaste d'une instabilité sur le trafic suite à une mauvaise d'orchestration et vice-versa. Un exemple classique est le "route flap", c'est-à-dire de basculement de routage d'un lien à un autre en raison de la charge de trafic importante sur ce lien, ce qui à son tour peut entraîner une charge de trafic importante sur le deuxième lien et un retour au premier lien. Cette instabilité peut avoir des conséquences néfastes sur la performance du réseau et sur la qualité de service perçue par les utilisateurs.

- **Topologie et Trafic :** L'instabilité Topologie et Trafic désigne les interactions entre les instabilités topologiques et celles du trafic. Par exemple, une charge de trafic élevée sur les liens peut rendre un nœud coupé du réseau, ce qui peut entraîner des fluctuations de la topologie des réseaux et des instabilités au niveau du trafic.

- **Orchestration et Topologie** : L'instabilité Orchestration et Topologie fait référence aux interactions dans lesquelles une mauvaise décision d'orchestration induit des instabilités topologiques et vice-versa. Un exemple serait la décision d'assigner l'exécution d'une fonction réseau virtuelle centrale à un noeud n'ayant pas les capacités de traitement suffisantes, rendant le noeud indisponible.

- **Orchestration, Topologie et Trafic :** La combinaison de tous les types d'instabilités.

Les conséquences potentielles d'une décision d'orchestration décrites ci-dessus explique la place centrale qu'aujourd'hui occupe la rentabilité dans la prise de décision d'orchestration. Une décision d'orchestration optimale pour l'opérateur sera celle qui tient compte à la fois de ses objectifs commerciaux et de ses contraintes opérationnelles.

1.3 Objectifs

L'objectif principal de cette thèse est de comprendre les conséquences algorithmiques et architecturales de la fourniture de ressources multi-domaines dans les réseaux de nouvelle génération. En particulier, il s'agit de comprendre les interactions et les mécanismes de coopération entre les parties prenantes ayant des obligations strictes vis-à-vis de leurs utilisateurs finaux, ainsi que les mécanismes de distribution de l'intelligence pour la prise de décision d'orchestration dans les réseaux à grande échelle. Pour atteindre cet objectif, nous proposons des algorithmes simples et efficaces pour améliorer le processus d'orchestration de ressources, tout en optimisant simultanément la qualité de service (QoS) pour les utilisateurs finaux et la rentabilité pour les opérateurs de réseau.

1.4 Organisation de la thèse

La présente thèse se compose de sept chapitres, dont le premier traite des principes fondamentaux de la gestion des ressources et des services de réseaux virtuels dans les réseaux de nouvelle génération. Ce chapitre établit la toile de fond de la thèse et présente les idées clés qui permettent de comprendre les travaux décrits dans cette thèse, tout en explorant les défis fondamentaux liés à la gestion des ressources et aux fonctions de réseaux virtuels dans les réseaux de l'avenir.

Le chapitre d'introduction générale de cette thèse vise à fournir une vue d'ensemble de cette étude et à établir un lien entre les quatre articles qui la composent, en synthétisant le contexte, les objectifs et les questions de recherche pertinentes. Il est en effet essentiel de définir le contexte général avant de s'immerger dans le sujet de recherche, afin de comprendre les causes profondes, les défis auxquels les différentes parties prenantes sont confrontées, ainsi que les conséquences algorithmiques et architecturales du problème de gestion des ressources et des fonctions de réseaux virtuels dans les réseaux multi-domaines.

La littérature pertinente au sujet recherché indique les facteurs problématiques suivants :

- l'absence d'un consensus architectural pour guider les interactions entre opérateurs ayant des modèles économiques, des infrastructures et des mécanismes de gestion de ressources différents, ainsi que des caractéristiques de trafic hétérogènes ;
- l'explosion du trafic de données et l'hétérogénéité des modèles d'accès causées par l'émergence d'applications innovantes de nouvelle génération. Le rythme de développement de ces applications est nettement plus rapide que le rythme d'innovation des infrastructures de réseau, et elles présentent des exigences de qualité de service (QoS) et de qualité de l'expérience (QoE) plus strictes, ainsi que des besoins en ressources plus importants. Cela réduit la marge d'erreur acceptable en matière d'orchestration de ressources ;
- la nature dynamique des applications et des conditions de réseau nécessite l'utilisation de stratégies d'orchestration de ressources adaptatives, capables

de prendre des décisions optimales en fonction d'évènements futurs incertains et des préférences des utilisateurs finaux et des opérateurs de réseau. En conséquence, la définition de l'optimalité d'une décision d'orchestration de ressources doit être redéfinie.

Le deuxième chapitre expose les méthodologies utilisées dans cette étude, en fournissant les justifications qui sous-tendent nos choix méthodologiques. Nous présentons également un aperçu des contributions de cette thèse, avant de rappeler brièvement les concepts clés liés aux méthodes sélectionnées.

En vue de favoriser la lisibilité et la pertinence, nous avons réalisé une revue de la littérature spécifique pour chaque article examiné dans cette thèse.

Le troisième chapitre est consacré au premier article de cette thèse, qui porte sur le placement et le chaînage des fonctions de réseaux virtuels dans les réseaux de diffusion de contenu (Content Delivery Networks - CDN) pour les services multimédias à valeur ajoutée. Cette étude établit les fondements de la compréhension des difficultés auxquelles les réseaux seront confrontés avec l'émergence d'applications de nouvelle génération, caractérisées par des exigences de qualité de service (Quality of Service - QoS) très strictes.

Le quatrième chapitre présente le deuxième article de cette thèse, qui explore les avantages d'une approche méthodologique hybride fondée sur l'apprentissage par renforcement profond (Deep Reinforcement Learning - DRL) pour l'approvisionnement de ressources virtuelles dans un environnement réseau dynamique, instable et à grande échelle.

Le cinquième chapitre présente le troisième article, qui poursuit l'exploration des mérites de la technique d'orchestration de ressources DRL, en mettant l'accent sur l'évolutivité et la généralisation de l'apprentissage.

Le sixième chapitre est consacré au quatrième article, qui porte sur les interactions entre opérateurs réseau dans un contexte de réseaux multi-domaines. Ce chapitre fournit un cadre conceptuel dans lequel divers acteurs s'engagent, de manière coopérative ou concurrentielle dans un marché, à obtenir les ressources de réseau nécessaires pour répondre aux besoins de qualité de service de leurs utilisateurs finaux.

Enfin, le septième chapitre conclut la thèse par une discussion et des pistes de recherches ultérieures.

CHAPITRE II

MÉTHODOLOGIE

Le chapitre suivant de cette thèse décrit les différentes approches méthodologiques utilisées dans cette étude. Il comprend quatre parties :

- La première partie discute des motivations qui sous-tendent les choix méthodologiques de cette thèse.
- La seconde partie décrit les différentes contributions de cette thèse.
- La troisième partie passe en revue les mesures prises pour maintenir une qualité satisfaisante de cette étude.
- La dernière partie fournit une explication succincte des algorithmes utilisés.

2.1 Problématiques et motivations méthodologiques

Ci-contre, un résumé des facteurs ayant influencé les décisions méthodologiques de la thèse est présenté.

2.1.1 Premier enjeu : la taille et la nature fondamentale du problème

À bien des égards, l'orchestration de ressources multi-domaines est un processus complexe pour plusieurs raisons. D'abord, c'est un problème d'optimisation combinatoire, ce qui signifie qu'il faut identifier une solution optimale, si elle existe, parmi un ensemble de solutions candidates. Compte tenu de la taille importante des réseaux et des nombreux changements d'état qui y ont lieu (topologie, trafic, etc.), le nombre de solutions candidates à évaluer devient rapidement exponentiel, ce qui rend la tâche difficile pour les problèmes d'optimisation combinatoire NP-difficiles.

La NP-difficulté d'un problème permet de mesurer la complexité computationnelle associée à la recherche de solutions optimales ou quasi optimales pour le problème étudié. Toutefois, sous l'hypothèse que $P \neq NP$, il est peu probable qu'une tâche NP-difficile puisse être résolue de manière optimale dans un temps raisonnable. Le but est alors de trouver des techniques heuristiques efficaces.

2.1.2 Second enjeu : la nature des applications et services

La nature complexe des applications que les opérateurs doivent fournir est un autre facteur qui rend difficile l'orchestration de ressources multi-domaines. Pour mieux comprendre les défis posés par le sujet traité dans cette thèse et nos choix méthodologiques, voici un aperçu du contexte :

Les opérateurs de réseaux sont confrontés au défi de savoir comment répondre aux diverses attentes et exigences d'une grande variété d'applications et de secteurs industriels pour des utilisateurs finaux situés partout dans le monde, tout en minimisant les coûts opérationnels.

L'exemple de TikTok, une application moderne représentative des applications prévues pour les réseaux de nouvelle génération (NGN), illustre bien les défis auxquels les opérateurs de réseaux sont confrontés. TikTok appartient à une société chinoise basée à Pékin et a une base mondiale d'utilisateurs (plus de 143 millions). Pour saisir l'étendue et l'ampleur du trafic quotidien créé par TikTok, il est important de noter que l'utilisateur le plus influent de TikTok (au présent) est un Sénégalais qui voyage fréquemment dans d'autres pays et publie régulièrement du contenu multimédia à l'intention de plus de 148 millions d'utilisateurs dans le monde qui interagissent avec ce contenu. En d'autres termes, la notion de restriction géographique est non existante et que chaque utilisateur doit avoir accès à l'application et à ses données en tout temps.

TikTok est une plateforme qui offre à ses utilisateurs la possibilité de créer des vidéos courtes et de les personnaliser avec divers effets, filtres et contenus. Ces modifications peuvent être apportées par l'utilisateur créateur lui-même, en utilisant par exemple de la musique de fond, du montage ou des effets de ralentissement ou d'accélération, ou bien par l'opérateur commercial, qui peut ajouter du contenu publicitaire ciblé pour capter l'attention de l'utilisateur final. Bref, le modèle de contenu généré en masse tel qu'on le connaissait avec les sites d'information n'est plus d'actualité. À la place, les données générées seront modifiées et personnalisées à différents moments du processus de création de contenu pour répondre aux exigences et aux attentes des multiples acteurs du réseau. Par exemple, le contenu généré pourrait être adapté en fonction des conditions du réseau de l'utilisateur final (par exemple, faibles débits de données ou fortes interférences) ou des capacités du dispositif de réception (par exemple, nécessité d'un encodage spécifique). En outre, il ne fait guère de doute que l'impressionnante popularité de TikTok est due à l'efficacité de son ingénieux moteur de suggestion. Ce moteur est capable de recommander fréquemment un flux unique de vidéos à chaque utilisateur en fonction de l'activité de son application, comme le fait d'avoir aimé/désaimé, effectué des recherches, interagi ou mis en favori un contenu. TikTok prend également en compte le profil démographique de l'utilisateur (par exemple, le contenu adapté aux mineurs).

Tout cela illustre le niveau d'interaction, de personnalisation, de traitement et d'intelligence requis aux opérateurs de réseau à charge d'assurer un approvisionnement adéquat d'applications modernes à une échelle planétaire, rien de moins. Le succès et la durabilité à long terme de TikTok dépendent grandement de la qualité de l'expérience des utilisateurs (QoE), qui dépend de la capacité des opérateurs de réseau à respecter les exigences de qualité de service associées, notamment en termes de latence et de taux de perte de paquets.

Il existe également d'autres cas où les opérateurs de réseau doivent absolument respecter des exigences de qualité de service strictes ou critiques. Par exemple :

- L'automatisation industrielle, dans laquelle une usine de fabrication utilise des robots pour effectuer des opérations coûteuses et exigeant une précision absolue. Une latence minimale est essentielle dans ce cas.
- La télémédecine dans un milieu hospitalier. En plus d'une faible latence, une bande passante dédiée est cruciale pour réduire au minimum la probabilité de perte de signal dans un contexte aussi critique.
- Les véhicules autonomes nécessitent une faible latence, une bande passante dédiée et une haute fiabilité pour fonctionner de manière sûre et efficace.

2.1.3 Troisième enjeu : la nature des décisions à prendre

Une solution algorithmique, efficace et évolutive capable d'atteindre les normes QoS stipulées est malheureusement insuffisante pour les applications de la prochaine génération.

En fait, la rapidité et la manière de prendre une décision d'orchestration des ressources sont tout aussi importantes que son optimalité (ou sa quasi-optimalité). Plusieurs raisons expliquent cette affirmation.

Pour la majorité des applications de la future génération, une latence maximale de

10 millisecondes (ms) est la norme (Panwar, 2020). À titre d'illustration, les utilisateurs d'une application de réalité virtuelle peuvent être gênés si le rendu et l'affichage de graphiques en réponse à leurs mouvements de tête sont retardés de plus de 10 ms. Compte tenu de la dispersion géographique des utilisateurs finaux et des nombreuses sources de retard du réseau (comme le codage, le décodage, le temps de trame, le routage et la transmission), il est difficile de prendre des décisions concernant l'orchestration et l'approvisionnement de services en moins de 10 ms.

Il convient donc de prendre en considération deux éléments supplémentaires lors de la prise de décision concernant l'orchestration de services : le timing et l'emplacement. Pour le timing, il faut se demander s'il est possible de prévoir une partie ou la totalité de la décision à l'avance et de l'appliquer rapidement. En ce qui concerne l'emplacement, il est essentiel de déterminer si la prise de décision doit se faire à proximité des utilisateurs finaux afin de minimiser les temps de transmission.

Le défi de la localisation est encore accentué lorsque l'on considère que l'application doit être déployée sur de nombreux domaines réseau, chacun étant géré de manière autonome par des opérateurs potentiellement concurrents. Dans ce contexte, les formes centralisées (par le biais d'un orchestrateur multi-domaine global) et décentralisées de prise de décision sont examinées dans la littérature. Les paradigmes tels que le réseautage axé sur l'intention (*intent-driven networking*) sont également pris en compte (par exemple, (Arezoumand et al., 2017)).

Dans tous les cas, il est crucial de mettre en place un cadre de coopération et d'incitation (technique, réglementaire et financier) entre opérateurs pour assurer une prise de décision efficace en matière d'orchestration dans les réseaux multi-domaines. Cela permettra de garantir une collaboration durable entre les différents opérateurs et de s'assurer que les utilisateurs finaux bénéficient de leurs applications et services dans de bonnes conditions.

De nos jours, l'approvisionnement d'applications réseau s'effectue à travers les paradigmes de *network slicing* et NFV où l'infrastructure réseau et les services réseaux sont segmentés en des composants logiques virtualisés, enchaînés et distribués sur plusieurs domaines réseau. Cependant, il est important de souligner que l'impact d'une mauvaise décision prise sur un segment de réseau spécifique peut s'étendre à d'autres segments de la chaîne, ainsi qu'à d'autres services ou à l'infrastructure réseau.

D'autant plus que la nature dynamique des conditions du réseau (interférences, pannes, congestion, etc.) et des ressources (capacité de stockage, pannes, etc.) rend le processus de prise de décision d'orchestration encore plus complexe. Un choix de mauvais ou de mauvais timing peut entraîner un gaspillage de dépenses, de consommation d'énergie et d'utilisation des ressources, et même menacer l'approvisionnement futur de services réseau. Ainsi, il est important de prendre en compte les incertitudes liées aux événements futurs lors de la prise de décision d'orchestration. Dans la littérature, il est courant de proposer des approches basées sur la prédiction de ces événements.

Cependant, comme l'exemple de TikTok le montre, il est également important de tenir compte des "préférences de l'utilisateur/opérateur" (ou élicitation des préférences) telles qu'elles sont exprimées par les fonctions d'utilité multiobjectives à moyen et long termes. Par exemple, dans les processus de décision de Markov, l'objectif est de trouver la politique qui maximise l'espérance d'utilité amortie. Cela signifie qu'il peut y avoir des cas où l'opérateur de réseau chargé de fournir une application préfère ne pas le faire, même s'il doit payer une pénalité (par exemple pour maximiser ses bénéfices futurs, prioriser une maintenance, etc.). De même, l'utilisateur final peut choisir de ne pas obtenir une application dans certaines circonstances.

Tous ces éléments ont une influence sur la notion d'*optimalité* d'une décision d'orchestration. En raison de l'imprévisibilité des événements futurs et des préférences des utilisateurs et des opérateurs, une décision d'orchestration optimale est un compromis entre la rapidité et la précision. Il est donc important de trouver un équilibre entre ces deux facteurs afin de prendre une décision efficace.

La profitabilité est également devenue un élément clé de la prise de décision en matière d'orchestration de ressources. Contrairement à ce qui est souvent présenté dans la littérature, elle n'est plus seulement considérée comme une conséquence secondaire de la prise de décision, mais plutôt comme l'un des indicateurs les plus importants de l'optimalité d'une prise de décision d'orchestration de ressources. Cette importance de la profitabilité s'explique par plusieurs raisons. Tout d'abord, l'énorme écart entre le rythme de développement des applications et des services et le rythme d'expansion de la capacité de l'infrastructure du réseau pousse souvent les opérateurs à privilégier certains segments du trafic du réseau, en violation du principe de neutralité du réseau de plus en plus contesté. Bien que l'argument de la gestion du trafic en faveur de l'abandon de la neutralité du réseau soit discutable, il est indéniable que les opérateurs sont confrontés à une forte concurrence de la part de tiers, ce qui entraîne des pertes économiques importantes et oblige à chercher de nouveaux modèles économiques (ODLYZKO, 2009 ; KIISKI, 2006 ; KRÄMER et al., 2013 ; GRZYBOWSKI et KARAMTI, 2010). En outre, la profitabilité joue également un rôle central en tant que mécanisme de coopération entre opérateurs de réseaux multi-domaines.

Les arguments précédents justifient également notre choix, tout au long de cette thèse, d'utiliser des méthodes basées sur l'apprentissage machine ML et l'intelligence artificielle (IA). En effet, la prise de décision doit être intelligente pour s'adapter aux dynamiques hétérogènes des environnements réseau. De plus, en raison des contraintes de temps et d'échelle des réseaux, une approche permettant de distribuer l'intelligence réseau est nécessaire.

En effet, les approches basées sur le ML et l'IA sont particulièrement efficaces pour apprendre les préférences et les probabilités d'événements futurs ou inconnus, ce qui peut être utilisé pour définir des modèles formels (par exemple, des processus de décision de Markov - MDP), des arbres de décision et d'autres modèles d'approximation. Ces outils peuvent ensuite être utilisés pour déduire des décisions "optimales".

2.2 Contributions de la thèse

Cette thèse vise à apporter des contributions à la question de l'orchestration de ressources multi-domaines dans les réseaux de nouvelle génération, qui seraient à la fois pertinentes sur le plan académique et industriel. Ainsi, chaque article de cette thèse se concentre sur un aspect spécifique de ce problème et propose des solutions simples et efficaces. L'analyse et la formulation du problème sont basées sur des approches théoriques bien établies, tandis que les expérimentations s'appuient sur des outils testés en pratique.

Le premier article de cette thèse, intitulé **CPVNF : Cost-Efficient Proactive VNF Placement and Chaining for Value-Added Services in Content Delivery Networks**, a été publié dans la revue *IEEE Transactions on Network and Service Management*. L'auteur de la thèse a participé activement au développement de la formulation de l'ILP (Integer Linear Programming) et de l'heuristique, à la mise en œuvre de l'évaluation expérimentale, à l'acquisition des résultats et à diverses tâches d'édition et de correction en réponse aux commentaires des évaluateurs de la revue.

Cet article présente les points suivants :

- Une étude de cas illustrant le problème de l'allocation et du chaînage de ressources virtuelles dans un réseau CDN (Content Delivery Network) mono-domaine. Il décrit les conditions préalables nécessaires à une solution potentielle et met en évidence les différences de notre travail par rapport aux articles existants dans le domaine.
- Une modélisation et une formulation ILP du problème.
- Une méthode proactive basée sur le PageRank pour l'allocation et le chaînage de ressources virtuelles, en tenant compte de différentes sources de données et de contraintes de QoS (Quality of Service).
- Une évaluation expérimentale comparative de notre stratégie et de la solution idéale.

Le second article, intitulé **Towards Reliable Remote Health Monitoring in Fog Computing Networks**, a été publié dans le journal *IEEE Transactions on Network and Service Management*. L'auteur de la thèse a participé activement au développement de la formulation de l'ILP (Integer Linear Programming) et de l'heuristique, à la mise en œuvre de l'évaluation expérimentale, à l'acquisition des résultats et à diverses tâches d'édition et de correction en réponse aux commentaires des évaluateurs de la revue.

Cet article propose :

- Un modèle de plateforme de fog computing capable de identifier et d'estimer la disponibilité de modèles d'appareils IoT/fog.
- Une formulation du problème du déploiement d'applications dans les nœuds fog sur lesquels s'exécutent des tâches de traitements provenant de capteurs

de patients, avec l'objectif de maximiser le nombre de tâches satisfaites en tenant compte des contraintes de latence et de disponibilité pour les nœuds fog.

- En raison de la difficulté NP (non-polynomiale) du problème, l'article propose un nouvel algorithme d'évolution différentielle auquel est appliqué un mécanisme de sélection adaptative des opérateurs (AOS) basé sur l'apprentissage par renforcement profond (DRL).
- Les résultats numériques montrent que notre approche est supérieure en termes de fiabilité par rapport aux solutions de base. De plus, en étudiant l'impact de différents paramètres clés, nous identifions un compromis entre le nombre de nœuds fog et les taux de défaillance intrinsèques de ces derniers.

Le troisième article, **DRL-based Green Resource Provisioning for 5G Networks** a été publié dans le journal *IEEE Transactions on Green Communications and Networking*. L'auteur de la thèse a participé activement au développement de la formulation de l'ILP (Integer Linear Programming) et de l'heuristique, à la mise en œuvre de l'évaluation expérimentale, à l'acquisition des résultats et à diverses tâches d'édition et de correction en réponse aux commentaires des évaluateurs de la revue.

L'article présente les points suivants :

- une modélisation du problème d'allocation de ressources, de façon verte, pour les réseaux sans fil du point de vue des opérateurs de réseaux mobiles en considérant des stations de base MIMO massive et une acquisition imparfaite des informations sur l'état du canal. Notre modèle de système prend également en compte les exigences en termes de débit de données des utilisateurs finaux, l'utilisation des ressources et la maximisation des profits.

- Une formulation du problème d'approvisionnement "verte" de ressources VNF dans les réseaux câblés, à grande échelle, du point de vue des opérateurs de réseaux mobiles. Nous prenons conjointement en compte les impacts environnementaux induits par l'approvisionnement des ressources, le respect des exigences QoS des utilisateurs finaux et l'utilisation des ressources.

- Deux solutions basées sur l'apprentissage par renforcement profond sont proposées pour résoudre les problèmes d'approvisionnement "verte" des ressources VNF et sans fil. Les solutions développées sont conçues pour être parallélisables pour les systèmes à grande échelle.

- Grâce à des évaluations expérimentales réalistes, nous illustrons l'efficacité des approches proposées pour réduire l'empreinte environnementale du réseau tout en satisfaisant les exigences de qualité de service des utilisateurs finaux.

Le dernier article, **Market Driven Multi-domain Network Service Orchestration in 5G Networks** a été publié dans la revue *IEEE Journal on Selected Areas in Communications* . L'auteur de la thèse a personnellement contribué à l'élaboration de l'étude de cas, de la formulation ILP et de l'heuristique, à l'implémentation de l'évaluation expérimentale, à l'acquisition des résultats ainsi que diverses tâches de rédaction et de correction suite aux remarques des *reviewers*. L'article propose :

- un exposé notant l'importance des interactions multi-domaines pour l'approvisionnement de services de bout en bout, à travers le cas d'utilisation MMOG (Massive Multiplayer Online Gaming).

- la modélisation d'une plateforme d'interaction multi-domaine, basée sur des mécanismes de libre marché, pour l'allocation de ressources entre un fournisseur d'infrastructures et plusieurs fournisseurs de services candidats. Nous formulons le problème d'allocation de ressources associé, où le fournisseur d'infrastructure et les fournisseurs de service visent à maximiser leurs profits en vendant/achetant des ressources sur le marché. Nous définissons différents environnements de marché, où les fournisseurs de services peuvent être en concurrence et/ou coopérer dans le processus d'appel d'offres, et où le fournisseur d'infrastructure peut décider d'appliquer, ou non, l'équité entre les fournisseurs de services.

- En raison de la complexité du système considérant avec plusieurs fournisseurs de services, nous proposons une approche d'apprentissage par renforcement profond multiagent distribué, où nous équipons chaque fournisseur de services d'un agent d'apprentissage capable de percevoir l'environnement et de prendre des actions stratégiques pour gagner des enchères, donc de satisfaire ses exigences de qualité de service et d'augmenter son profit. De plus, les agents de différents fournisseurs de services peuvent échanger des informations dans des scénarios de coopération afin d'améliorer leurs profits mutuels.

- À travers des scénarios d'expérimentation, nous avons formé des agents de fournisseurs de services et évalué les performances du fournisseur d'infrastructure et des fournisseurs de services dans différents scénarios de marché. Il a été démontré qu'un marché entièrement concurrentiel (sans coopération) est le plus rentable pour le fournisseur d'infrastructure. En revanche, un marché coopératif est en moyenne le plus lucratif pour les fournisseurs de services. De plus, les agents à apprentissage double-deep-Q surpassent les autres agents à apprentissage et sans apprentissage en termes de profits pour les fournisseurs de services. Enfin, l'impact de certains paramètres est souligné.

2.3 Critères de la qualité d'étude

Nous adoptons la méthodologie de recherche suivante afin de maintenir une qualité de recherche satisfaisante.

Pour la rédaction de chaque article, nous avons réalisé une recherche bibliographique approfondie de la littérature dans le but d'identifier les facteurs pertinents au sujet de chaque article. Ainsi, chaque article comprend une section dédiée à la revue de la littérature. Cette recherche bibliographique nous permet d'obtenir une compréhension approfondie du sujet et d'analyser les compromis des solutions existantes.

Ensuite, nous construisons un modèle système représentatif du problème en question, permettant une analyse formelle rigoureuse. Nous présentons les hypothèses utilisées pour chaque modèle, avant de définir une formulation ILP du problème en précisant les fonctions objectifs et les contraintes correspondantes. Lorsque pratiques, à savoir présentant des délais d'exécution raisonnables, nous utilisons des outils mathématiques pour la résolution et d'implémentation de problèmes combinatoires tels que CPLEX, PuLP, LP_solve ou Julia/JuMP (IBM, 2022 ; PuLP, 2022 ; LP_solve, 2022 ; JuMP, 2022)

En général, nous utilisons la méthode de réduction par restriction pour démontrer la NP-difficulté d'un problème. Ensuite, nous créons des algorithmes heuristiques qui traitent efficacement le problème. La dernière étape consiste à évaluer les performances à travers des simulations. Nous validons les résultats acquis en les comparant à des méthodes similaires décrites dans la littérature. La figure 2.1 résume notre méthodologie.

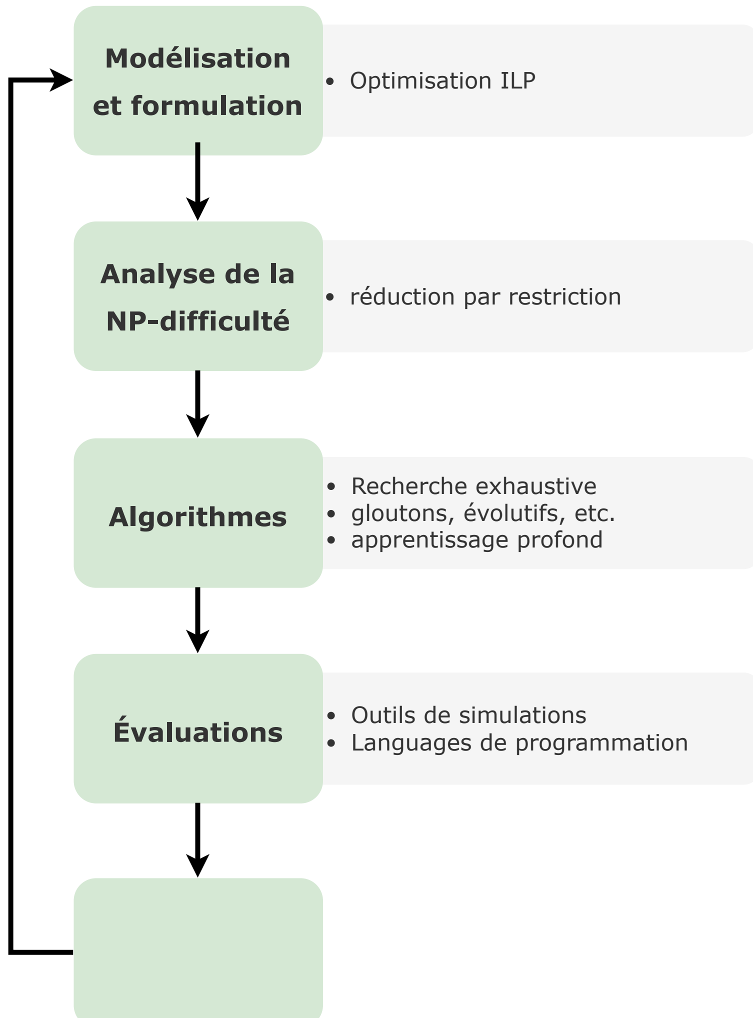

FIGURE 2.1 Approche méthodologique de la thèse

2.3.1 Expérimentations

Nous implémentons nos solutions en utilisant une combinaison de langages de programmation de haut niveau (Python, Java, Julia, etc.) ; d'outils de simulation et de plateformes de développement utilisés dans l'industrie (OpenAI, Pytorch, Tensorflow, stable baselines, etc.).

Afin de limiter les effets stochastiques inhérents à la nature des solutions algorithmiques proposées (par exemple, apprentissage par renforcement profond, algorithme évolutionnaire, etc.) ou aux paramètres de simulation, nous répétons généralement les simulations plusieurs fois et, si nécessaire, nous éliminons les résultats jugés aberrants. Les références suivantes (Henderson et al., 2018 ; Chan et al., 2020 ; Agarwal et al., 2021) sont les principales sources d'inspiration de cette stratégie.

Pour minimiser les erreurs de mise en œuvre des algorithmes d'apprentissage, nous utilisons souvent la procédure suivante :

- Lorsqu'une implémentation d'un algorithme d'apprentissage avec suffisamment de documentation est disponible, nous réutilisons et adaptons le code si nécessaire (par exemple, Stable baselines, dépots github, OpenAI baselines, etc.).
- Sur la base de la structure et des méthodologies de programmation OpenAI Gym, nous concevons et personnalisons les environnements d'apprentissage.
- Les paramètres d'apprentissage sont initialement inspirés de travaux pertinents dans la littérature ou de simulations antérieures. Ensuite, nous les modifions expérimentalement en fonction des performances d'apprentissage.

2.4 Rappels sur les algorithmes utilisés

À travers cette section, nous effectuons un bref rappel sur les principales méthodes employées dans cette thèse. La personne familier(e) avec les concepts décrits ci-dessous peut sauter directement à la page 63.

2.4.1 Programmation linéaire en nombres entiers (Bradley et al., 1977)

Un problème de programmation linéaire en nombres entiers (en anglais *integer linear programming*, abrégé *ILP*) est un problème d'optimisation NP-difficile composé d'une fonction objectif et d'un ensemble de contraintes linéaires, et dont une partie ou la totalité des variables de décision sont des nombres entiers.

Un problème ILP est généralement décrit selon la forme standard suivante :

$$\max \sum_{j=1}^{n} c_j x_j,$$

tel que :

$$\begin{aligned} \sum_{j=1}^{n} a_{ij} x_j &= b_i && (i = 1, 2, \cdots, m), \\ x_j &\geq 0 && (j = 1, 2, \cdots, n), \\ x_j &\in \mathbb{Z} && (\text{for some or all } j = 1, 2, \cdots, n). \end{aligned}$$

Les restrictions d'intégralité reflètent les indivisibilités naturelles du problème modélisé. Par exemple, il serait illogique de considérer des valeurs fractionnelles concernant le nombre d'infirmières à recruter pour un hôpital. Les restrictions peuvent aussi découler des contraintes logiques ou des non-linéarités (p.ex. des coûts fixes).

Nous présentons ci-dessous quelques méthodes notables de formulation ILP.

2.4.1.1 Variables binaires(0-1)

Prenons l'exemple où nous devons déterminer l'exécution ou non de plusieurs tâches. Dans chaque cas, la décision à prendre est de type oui/non. Ces choix peuvent être modélisés en considérant une variable de décision $x_j = 1$ pour dénoter une réponse positive pour la j-ème activité et $x_j = 0$ sinon. Ces variables sont qualifiées de binaires, bivalentes, logiques ou 0-1. Elles sont d'une grande importance, car elles apparaissent régulièrement dans les formulations de modèles.

2.4.1.2 Contraintes logiques

Souvent, les problèmes imposent des contraintes logiques sur les variables de décision (p.ex. restrictions de temps, alternatives conflictuelles, etc.). Ci-dessous, nous passons en revue des exemples notables de relations logiques.

- *Contrainte de faisabilité*

  Ce type de contrainte permet de savoir si un choix donné de variables de décision satisfait à une contrainte. Soit une variable binaire y tel que :

  $$y = \begin{cases} 0 & \text{lorsque la contrainte est satisfaite,} \\ 1 & \text{sinon,} \end{cases}$$

  où

  $$f(x_1, x_2, \cdots, x_n) - By \leq b, \tag{2.1}$$

  avec B, une constante, choisie suffisamment grande pour que la contrainte soit toujours satisfaite si y = 1, c'est-à-dire

  $$f(x_1, x_2, \cdots, x_n) \leq b + B, \quad \forall\ x_1, x_2, \cdots x_n \tag{2.2}$$

  Chaque fois que $y = 0$ donne une solution réalisable pour la contrainte 2.2, cela implique que la contrainte 2.1 est forcément satisfaite. En pratique, il est généralement aisé de trouver un grand nombre pour B, bien qu'il soit généralement préférable d'utiliser la plus petite valeur possible afin d'éviter les difficultés numériques lors des calculs.

- *Contraintes alternatives*

  Soit le cas suivant où au moins l'une des contraintes, mais pas nécessairement les deux, doit être satisfaite :

$$f_1(x_1, x_2, \cdots, x_n) \leq b_1,$$
$$f_2(x_1, x_2, \cdots, x_n) \leq b_2.$$

  Cette restriction peut être modélisée de la manière suivante :

$$\begin{aligned} f_1(x_1, x_2, \cdots, x_n) - B_1 y_1 &\leq b_1, \\ f_2(x_1, x_2, \cdots, x_n) - B_2 y_2 &\leq b_2, \\ y_1 + y_2 &\leq 1, \\ y_1, y_2 &\in \{0, 1\}. \end{aligned}$$

- *Contraintes conditionnelles*

  Ce type de contrainte a la forme suivante :

$$f_1(x_1, x_2, \cdots, x_n) > b_1 \quad \text{implique} \quad f_2(x_1, x_2, \cdots, x_n) \leq b_2.$$

  Puisque l'implication est uniquement non satisfaite lorsque $f_1(x_1, x_2, \cdots, x_n) > b_1$ et $f_2(x_1, x_2, \cdots, x_n) > b_2$, la contrainte conditionnelle ci-haut est logiquement équivalente aux contraintes alternatives suivantes :

$$f_1(x_1, x_2, \cdots, x_n) \leq b_1 \quad \text{et/ou} \quad f_2(x_1, x_2, \cdots, x_n) \leq b_2,$$

  où au moins une des contraintes doit être satisfaite.

- *k-Plis alternatifs* Supposons que nous devions satisfaire au moins k des contraintes. En d'autres termes, nous pouvons ignorer certaines contraintes si au k nombre de contraintes sont satisfaites :

$$f_j(x_1, x_2, \cdots, x_n) \leq b_j \quad (j = 1, 2, \cdots, p).$$

En supposant que les valeurs de $B_j, \ \forall\, j = 1, 2, \cdots, p$ sont choisis de manière à ce que les contraintes ignorées ne soient pas contraignantes, le problème peut être généralement formulé comme suit :

$$f_j(x_1, x_2, \cdots, x_n) - B_j(1 - y_j) \leq b_j \quad (j = 1, 2, \cdots, p),$$

$$\sum_{j=1}^{p} y_j \leq k,$$

$$y_j \in \{0, 1\} \quad (j = 1, 2, \cdots, p).$$

Autrement dit, $y_j = 1$ si la j-ème contrainte doit être satisfaite, et au moins k des contraintes doivent être satisfaites. Si on définit $y'_j \equiv 1 - y_j$, et que l'on remplace y_j dans ces contraintes, la forme des contraintes résultantes est analogue à celle donnée précédemment pour la modélisation des contraintes alternatives.

2.4.1.3 Représentation de fonctions non linéaires

- *Coûts fixes*

  Typiquement utilisé dans la fonction objectif (coûts de conception préliminaire, les coûts d'investissement fixes, les contrats fixes, et ainsi de suite).

- *Représentation linéaire par morceaux*

  Un autre type de fonction non linéaire qui peut être représenté par des variables entières est une courbe linéaire par morceaux.

  Par exemple, pour modéliser la courbe de coût, nous exprimons toute valeur de x comme étant la somme de trois variables $\delta_1, \delta_2, \delta_3$ de sorte que le coût de chacune de ces variables soit linéaire. Ainsi,

$$x = \delta_1 + \delta_2 + \delta_3,$$

avec

$$
\begin{aligned}
&0 \leq \delta_1 \leq a_1,\\
&0 \leq \delta_2 \leq a_2,\\
&0 \leq \delta_3 \leq a_3.\\
&a_1, a_2, a_3 \in \mathbb{Z}
\end{aligned}
$$

2.4.2 PageRank (PR)

Cette sous-section fournit un bref résumé de l'approche PageRank. Nous invitons la personne intéressée de consulter les références suivantes pour plus de détails sur le sujet ainsi que les approches connexes (Chung, 2014 ; Haveliwala, 1999 ; Brin et Page, 1998 ; Bianchini et al., 2005 ; Berkhin, 2005).

PR est un algorithme, basé sur le théorème de point fixe, utilisé par le moteur de recherche Google pour classer les pages Web en fonction de leur pertinence. Plus précisément, PR fonctionne en comptant le nombre et la qualité des liens vers une page pour déterminer une estimation approximative de l'importance du site Web. L'hypothèse sous-jacente étant que les sites Web les plus importants sont susceptibles de recevoir plus de liens depuis d'autres sites Web.

D'une manière générale, PR est une technique d'analyse des liens applicable à tout ensemble d'entités comportant des citations et des références réciproques, qui attribue une pondération numérique à chaque élément d'une collection pour déterminer leur importance relative au sein de la collection. Dans le contexte de cette thèse, nous l'utilisons pour exploiter la structure du graphe réseau et déterminer l'importance des nœuds de traitement pour une requête de placement de service donnée.

Un inconvénient notable de PR est qu'il favorise les pages plus anciennes. Le développement de techniques permettant d'accélérer le calcul du PR reste un sujet de recherche actif dans la littérature.

2.4.3 Évolution différentielle (DE)

Cette sous-section fournit un bref résumé de l'approche DE. Nous invitons la personne intéressée de consulter les références suivantes pour plus de détails sur le sujet ainsi que les approches connexes (Sharma et al., 2019 ; Das et Suganthan, 2010 ; Storn et Price, 1997).

DE est une technique d'optimisation qui tente de manière itérative d'améliorer une solution candidate en fonction d'un critère de qualité particulier. En résumé, DE optimise un problème en conservant une population de solutions candidates et en générant de nouvelles solutions candidates en combinant les solutions existantes, puis en conservant la solution candidate qui a le meilleur score ou la meilleure aptitude pour la tâche d'optimisation en cours.

DE n'exige pas que le problème d'optimisation soit différentiable, contrairement aux méthodes d'optimisation classiques telles que l'approche par descente de gradient. Ainsi, DE est applicable aux problèmes d'optimisation qui sont non continus, bruyants, dynamiques, etc. En outre, DE ne fait que peu ou pas d'hypothèses sur le problème à résoudre et est capable de rechercher de très grands espaces de solutions candidates.

Comparé à d'autres algorithmes évolutionnaires (EA), DE a l'avantage d'être une technique efficace de recherche de population simple à mettre en œuvre, nécessite moins de paramètres de contrôle et fournit des performances supérieures dans une variété de domaines (Das et Suganthan, 2010).

Comme les algorithmes EA classiques, DE comporte quatre étapes principales :

1. Initialisation :

   Selon la nature du problème, la population comprend généralement des centaines ou des milliers de solutions potentielles. Pour cette raison, la population initiale est produite de manière aléatoire, ce qui permet d'examiner le spectre complet des solutions possibles (l'espace de recherche).

2. Croisement :

   Le croisement, également appelé recombinaison, est un opérateur génétique qui combine le matériel génétique de deux parents pour produire une descendance. Il s'agit d'une technique permettant de générer de nouvelles solutions de manière stochastique à partir d'une population existante. En général, les solutions nouvellement développées subissent des mutations avant d'être introduites à la population.

3. Mutation :

   La mutation est un opérateur génétique utilisé pour maintenir la diversité génétique au sein d'une population de génération en génération. L'objectif de la mutation est d'introduire de la diversité dans un échantillon de population. Les opérateurs de mutation sont utilisés pour éviter les minimums locaux en empêchant la population chromosomique de devenir trop homogène, ce qui ralentit ou empêche la convergence vers l'optimum global.

4. Sélection :

   À chaque génération, un sous-ensemble de la population existante est choisi pour former la génération suivante. Les solutions individuelles sont choisies en fonction de leur aptitude, déterminée par une fonction d'aptitude. L'objectif est de préserver les solutions les plus susceptibles de durer jusqu'à la génération suivante. La fonction d'aptitude dépend toujours du problème.

Dans la terminologie DE, un "parent" de la génération actuelle est appelé "cible", un mutant obtenu par l'opération de mutation est appelé "donneur", et enfin, une progéniture formée par recombinaison du donneur avec la "cible" est appelée "essai".

DE commence par une génération de NP individus (étape d'initialisation). Contrairement aux autres EA, DE génère la génération suivante en perturbant les différences d'échelle des membres distincts sélectionnés de la population (étape de mutation). Chaque stratégie de mutation a ses compromis d'exploration/exploitation (Sharma et al., 2019 ; Das et Suganthan, 2010).

Supposons que o_j et v_j sont respectivement la j^{ime} progéniture et le parent à la génération courante. Les stratégies de mutation suivantes sont définies :

$$\begin{aligned}
&\textbf{DE/rand/1} : o_j = v_{b_1} + M(v_{b_2} - v_{b_3}) \\
&\textbf{DE/rand/2} : o_j = v_{b_1} + M(v_{b_2} - v_{b_3}) + M(v_{b_4} - v_{b_5}) \\
&\textbf{DE/best/1} : o_j = v_{\text{best}} + M(v_{b_1} - v_{b_2}) \\
&\textbf{DE/best/2} : o_j = v_{\text{best}} + M(v_{b_1} - v_{b_2}) + M(v_{b_3} - v_{b_4}) \\
&\textbf{DE/target-to-best/1} : o_j = v_j + M(v_{\text{best}} - v_j) \\
&\qquad\qquad\qquad\qquad + M(v_{b_1} - v_{b_2}) \\
&\textbf{DE/rand-to-best/2} : o_j = v_{b_1} + M(v_{\text{best}} - v_{b_1}) \\
&\qquad\qquad\qquad\qquad + M(v_{b_2} - v_{b_3}) + M(v_{b_4} - v_{b_5}),
\end{aligned}$$

où M est un facteur d'échelle, v_{best} est le meilleur parent dans la population, et b_i ($i = 1, \ldots, 5$) sont des indices générés aléatoirement dans la plage $[1, NP]$.

Le croisement est réalisé sur un individu cible $\theta_j = \{\theta_{j,1}, \ldots, \theta_{j,J'}\}$ et un individu donneur $\psi_j = \{\psi_{j,1}, \ldots, \psi_{j,J'}\}$ pour générer un essai $\phi_j = \{\phi_{j,1}, \ldots, \phi_{j,J'}\}$, tel que $\forall j' = 1, \ldots, J'$,

$$\phi_{j,j'} = \begin{cases} \psi_{j,j'} & \text{si } rand_{j,j'}[0,1] \leq C \text{ or } j' = j_{rand} \\ \theta_{j,j'} & \text{sinon} \end{cases} \tag{2.3}$$

où $C \in [0, 1]$ est le paramètre de croisement qui détermine si la stratégie de mutation est appliquée ou non. La fonction $rand_{j,j'}[0, 1]$ génère un nombre aléatoire uniformément distribué entre $[0, 1]$ et j_{rand} est un indice choisi au hasard qui garantit qu'au moins un composant de l'individu donneur ψ_j est hérité dans l'essai ϕ_j.

Généralement, une phase de sélection suit la phase de croisement afin de maintenir une population stable au cours des générations suivantes. L'opération de sélection détermine si une descendance cible ou une descendance d'essai doit être transmise à la génération suivante :

$$\theta_j^+ = \begin{cases} \phi_j & \text{if } \xi(\phi_j) \leq \xi(\theta_j) \\ \theta_j & \text{if } \xi(\phi_j) > \xi(\theta_j), \end{cases} \tag{2.4}$$

où θ_j^+ représente la cible de la génération suivante, et $\xi(\cdot)$ est une fonction d'évaluation qui doit être maximisée. Cela permet de garantir que la qualité de la population ne diminue jamais au fil du temps.

L'analyse expérimentale a montré que différents opérateurs, par exemple, les schémas d'encodage, les réglages de paramètres, les stratégies de mutation, de croisement, etc. sont plus performants pour des problèmes d'optimisation spécifiques. Par exemple, les auteurs de (Sharma et al., 2019 ; Das et Suganthan, 2010) ont montré que le choix adéquat de la stratégie de mutation à des étapes spécifiques peut améliorer significativement les performances algorithmiques.

2.4.4 Apprentissage par renforcement profond (DRL)

Cette sous-section fournit un bref résumé de l'approche d'apprentissage par renforcement profond. Nous invitons la personne intéressée à consulter les références suivantes pour plus de détails sur le sujet ainsi que les approches connexes (Sutton et Barto, 2018 ; Kaelbling et al., 1996 ; Arulkumaran et al., 2017a ; François-Lavet et al., 2018 ; Arulkumaran et al., 2017b ; François-Lavet et al., 2018 ; Henderson et al., 2018 ; Li, 2017 ; Mousavi et al., 2016 ; LeCun et al., 2015 ; Jordan et Mitchell, 2015 ; Shafique et al., 2018).

L'omniprésence croissante des données à grande échelle dans tous les domaines de l'activité humaine impose un nouvel ensemble d'exigences en matière de calcul. Par exemple, les grands ensembles de données nécessitent des algorithmes d'une complexité algorithmique acceptable, tandis que les données hautement sensibles nécessitent des algorithmes qui minimisent les implications en matière de confidentialité.

Cette situation a facilité le renouveau des algorithmes d'apprentissage automatique (ML). Essentiellement, l'utilisation d'une technique d'apprentissage pour résoudre un problème implique de tenter d'améliorer les indicateurs de performance en fonction des informations acquises par un entraînement répété sur un vaste univers de données. Cependant, en raison de la grande quantité de données non étiquetées, la construction d'algorithmes d'apprentissage reste une tâche difficile.

L'apprentissage par renforcement profond (DRL) est un sujet du ML qui combine l'apprentissage profond (DL) et l'apprentissage par renforcement (RL). David Silver, un pionnier de l'intelligence artificielle (IA) et un contributeur important d'AlphaGo, estime que l'intelligence générale = DRL = RL + DL.

Le DRL est une technique qui permet à un agent d'interagir avec son environnement et de découvrir, par essais et erreurs, la politique de décision optimale pour un problème et un objectif donnés. L'approche DRL repose sur des données d'entrée non structurées pour lesquelles il n'est pas nécessaire de procéder à une ingénierie manuelle de l'espace d'état. De toute évidence, étant donné l'importance des données d'entrée, l'espace d'état est vaste.

La grande difficulté de corréler des actions immédiates à des conséquences futures justifie l'emploi d'une méthodologie par essais et erreurs. Comme les humains, les systèmes d'apprentissage par renforcement doivent surveiller les résultats de leurs décisions avant de procéder à des ajustements à long terme. Ces approches sont applicables dans des situations où il existe un long délai entre une activité et son résultat possible, ce qui rend difficile de tirer des conclusions causales.

L'apprentissage profond s'est avéré essentiel au succès de l'apprentissage par renforcement, car il minimise ou élimine la nécessité d'une expertise approfondie du domaine dû à la capacité des algorithmes DL et des réseaux neuronaux profonds à effectuer un apprentissage des représentations à partir des données d'entrées non structurées. En effet, pour surmonter les difficultés exponentielles causées par la malédiction de la dimensionnalité, une représentation profonde et distribuée qui utilise la composition hiérarchique des composants de données est nécessaire. Malgré cela, la DL présente un certain nombre de limites, notamment l'incapacité à concurrencer l'intelligence humaine sur des tâches spécifiques et la nature de boîte noire de nombreuses décisions, qui les rend incompréhensibles pour les humains.

Il existe de multiples techniques, chacune d'entre elles présentant des avantages, pour établir des politiques d'apprentissage pour les algorithmes DRL. De façon basique, l'apprentissage par renforcement basé sur un modèle se distingue de l'apprentissage par renforcement sans modèle par le fait que l'algorithme cherche à créer un modèle prédictif de la dynamique de l'environnement. Pour ce dernier (l'élaboration d'un modèle prédictif), l'apprentissage supervisé par réseau de neurones est souvent utilisé pour obtenir les actions de l'agent. Cependant, l'agent devra ajuster ses décisions au fur et à mesure qu'il interagit avec l'environnement, car la dynamique réelle de l'environnement diffère généralement de la dynamique apprise. Des approches Monte-Carlo, telles que la méthode de l'entropie croisée (*cross-entropy*), ou une combinaison de techniques d'apprentissage sans modèle (*model-free*) et basées sur un modèle peuvent être utilisées pour optimiser les actions spécifiées.

Dans les méthodes d'apprentissage sans modèle, une politique est apprise sans modéliser explicitement la dynamique de l'environnement. Elle peut être configurée pour maximiser les rendements en estimant directement le gradient de la politique (*policy gradient*), mais sa variance élevée la rend peu pratique dans DRL pour l'approximation de fonction (*function approximation*). D'autres méthodes d'apprentissage sans modèle notables et plus récentes sont l'apprentissage par différence temporelle et le *Q-learning* basés sur la programmation dynamique. Pour des espaces d'action discrets, le *Q-learning* utilise une fonction Q qui prédit les rendements futurs lorsqu'une action est exécutée à partir d'un état donné. Pour des espaces d'action continus, une estimation de la valeur d'un état et une politique sont généralement obtenues.

2.4.4.1 L'apprentissage automatique (ML)

Les algorithmes ML sont typiquement regroupés en 3 classes d'apprentissage : (1) supervisée dans laquelle les données sont étiquetées ; (2) non supervisée avec des données non étiquetées et (3) par renforcement, caractérisée par la présence d'un mécanisme de rétroaction pour guider le processus d'apprentissage.

Ils nécessitent typiquement :

- **un ensemble de données** : séparé en sous-groupes distincts pour l'entraînement, la validation et le test ;
- **un processus d'optimisation** ;
- **une fonction d'évaluation** (coût/perte), évaluant les performances du modèle en termes de précision ou d'erreurs, plus précisément les erreurs d'apprentissage et de généralisation.

  L'erreur d'apprentissage mesure l'erreur sur les données d'apprentissage, dont la minimisation est un problème d'optimisation. L'erreur de généralisation ou erreur de test mesure l'erreur sur les nouvelles données d'entrée, ce qui différencie l'apprentissage automatique de l'optimisation ;

  L'objectif d'un algorithme ML est de minimiser à la fois l'erreur d'apprentissage ainsi que l'écart entre l'erreur d'apprentissage et l'erreur de test. Un modèle est sous-adapté (*underfit*) s'il ne peut pas atteindre une erreur d'apprentissage faible. Il est suradapté (*overfit*) si l'écart entre l'erreur d'apprentissage et l'erreur de test est important.

  De nombreux algorithmes ML se concentrent sur les problèmes d'approximation de fonctions, dans lesquels l'objectif d'optimisation est intégré à une fonction d'estimation. Le défi de l'apprentissage consistant à augmenter la précision de cette fonction en utilisant l'expérience acquise lors de l'entraînement.

  Dans certaines circonstances, la fonction est représentée explicitement comme une forme fonctionnelle paramétrée. Dans d'autres, la fonction est implicite et générée par un processus de recherche, une factorisation, une optimisation ou une approche basée sur la simulation. Même si elle est implicite, la fonction dépend généralement de paramètres ajustables. L'apprentissage consiste alors à trouver des valeurs pour ces paramètres qui maximisent la mesure de performance.

- **un modèle** : le processus élaboration d'un modèle ML suit le principe heuristique du rasoir d'Occam, en ce sens qu'à expressivité (ou capacité) égale, les modèles les plus simples sont préférés. La capacité d'un modèle mesure l'étendue des fonctions exprimables par le modèle. Cependant, il existe un compromis entre d'une part, la capacité d'un modèle et d'autre part, les erreurs de généralisation et d'apprentissage.

  L'objectif est donc de trouver une capacité optimale permettant d'obtenir à la fois une erreur d'apprentissage faible et un faible écart entre l'erreur d'apprentissage et l'erreur de généralisation. À cet effet, deux facteurs d'influence doivent être pris en compte surtout par rapport à l'erreur de généralisation : (1) le biais qui mesure la déviation espérée de la fonction d'estimation par rapport à la value réelle et (2) la variance qui mesure la variabilité de la fonction d'estimation, particulièrement lorsque les données d'entrée changent. À mesure que la capacité du modèle augmente, le biais diminue et la variance augmente, présentant de ce fait un autre compromis entre l'erreur de généralisation et la capacité du modèle. Une approche permettant de réduire l'erreur de généralisation est la régularisation où une pénalité est ajoutée à la fonction d'évaluation.

2.4.4.2 L'apprentissage profond (DL)

Comme indiqué précédemment, les techniques ML antérieures étaient limitées par la quantité de données brutes qu'elles pouvaient analyser. La construction d'un système d'apprentissage automatique nécessitait typiquement une ingénierie minutieuse et une connaissance approfondie du domaine pour extraire les caractéristiques (*features*) qui transforment les données brutes en une représentation interne ou un vecteur de caractéristiques que le système d'apprentissage peut utiliser pour générer un résultat pertinent. Cette situation pousse à l'émergence de l'apprentissage par représentation qui est un ensemble d'algorithmes permettant d'alimenter un système d'apprentissage en données brutes et de découvrir automatiquement les représentations appropriées pour exécuter des tâches spécifiques (par exemple, détection, classification, etc.).

DL est une collection de méthodes d'apprentissage par représentation ayant plusieurs couches de représentation. Celles-ci (couches de représentation) sont des modules simples, mais non linéaires qui d'une couche à l'autre transforment une représentation donnée. Après chaque couche, la représentation obtenue devient légèrement plus abstraite permettant l'apprentissage de fonctions très complexes après un nombre suffisant de ces transformations. Pour DL, il est essentiel de comprendre que ces couches de représentation ne sont pas manuellement conçues, mais plutôt acquises automatiquement à l'aide d'une technique d'apprentissage générale. Également, la "profondeur" de l'apprentissage pour une méthode DL fait référence au nombre de couches utilisé pour l'apprentissage par représentation.

Pour les méthodes DL modernes, l'apprentissage par représentation s'effectue à travers des réseaux de neurones artificiels profonds qui sont vaguement inspirés des neurones d'un cerveau biologique. Un réseau de neurones est une collection de neurones connectés qui se transmettent des signaux à l'instar des synapses pour un cerveau biologique. Plus spécifiquement, le rôle d'un neurone est de recevoir des signaux, les traiter avant d'envoyer des signaux à d'autres neurones. Le signal de sortie d'un neurone est un nombre réel déterminé par une fonction non linéaire de la somme de ses entrées. Chaque neurone et connexion possèdent un poids qui fluctue lors de l'apprentissage et qui détermine l'intensité du signal au niveau d'une connexion. Les neurones peuvent avoir un seuil au-delà duquel un signal n'est pas transmis.

Les architectures de réseaux neuronaux DL comprennent les réseaux neuronaux profonds (*Deep Neural Networks - DNN*), les réseaux de croyance profonds (*Deep Belief Networks - DBN*), les réseaux neuronaux récurrents (*Recurrent Neural Networks - RNN*), la mémoire à long terme (LSTM), les réseaux neuronaux convolutifs (*Convolutional Neural Networks - CNN*) et les transformateurs, entre autres. Nous suggérons à la personne intéressée de consulter (Liu et al., 2017 ; Hosseini et al., 2020 ; Shafique et al., 2018) pour une revue des diverses architectures des réseaux neuronaux d'apprentissage profond.

2.4.4.3 L'apprentissage par renforcement profond (DRL)

Les algorithmes RL cherchent à apprendre la meilleure séquence de décisions à partir de données brutes massives pour atteindre un objectif ou optimiser une métrique dans le temps. À cette fin, ils emploient des agents de décision formés à l'aide de systèmes de punition et de récompense. Dans des conditions optimales, les algorithmes RL peuvent atteindre des performances remarquables en partant de zéro. La principale difficulté des algorithmes RL est qu'ils fonctionnent dans un environnement à rendement différé, ce qui rend difficile la corrélation entre les actions actuelles et les conséquences futures.

En résumé, les problèmes résolus à travers RL sont souvent modélisés comme étant un MDP dans lequel, à chaque instant, un agent prend une action a pour un état s, obtient une récompense scalaire puis transite vers un prochain état s' selon la dynamique de l'environnement $p(s'|s'a)$. Notez qu'un algorithme RL n'a pas accès à la totalité de la dynamique de l'environnement, mais seulement à des échantillons. Le but d'un agent est d'apprendre une politique $\pi(a|s)$ ou une correspondance observation-action qui lui permettra de maximiser ses revenus, à savoir la somme espérée des récompenses obtenues.

Les algorithmes RL classiques atteignent leurs limites lorsque l'on considère un MDP avec des états à haute dimension. Cette situation a favorisé l'émergence des algorithmes DRL qui relient les réseaux de neurones artificiels au RL en combinant l'approximation des fonctions et l'optimisation d'objectifs (*target optimization*) pour faire la correspondance états-actions à leurs récompenses appropriées.

CHAPITRE III

ARTICLE 1 - CPVNF: COST-EFFICIENT PROACTIVE VNF PLACEMENT AND CHAINING FOR VALUE-ADDED SERVICES IN CONTENT DELIVERY NETWORKS

Mouhamad Dieye
Université du Québec à Montréal
Montréal (Québec), Canada

Shohreh Ahvar, Professeur associé
Institut Supérieur d'Électronique de Paris
Paris, France

Jagruti Sahoo, Professeur assistant
South Carolina State University
Orangeburg (South Carolina), États-Unis d'Amérique

Ehsan Ahvar, Professeur associé

École Supérieure d'Informatique Électronique Automatique

Paris, France

Roch Glitho, Professeur titulaire

Concordia University

Montréal (Québec), Canada

Halima Elbiaze, Professeur titulaire

Université du Québec à Montréal

Montréal (Québec), Canada

Noel Crespi, Professeur titulaire

Institut Mines-Telecom, Institut Polytechnique de Paris

Paris, France

AVANT-PROPOS À L'ARTICLE 1

Le premier article de la thèse traite du problème de l'approvisionnement de services à valeur ajoutée à travers l'enchaînement de VNFs dans les réseaux CDNs avec comme contraintes, la minimisation des coûts d'exploitation et le respect des exigences de qualité de service. Cet article vise à explorer l'impact du placement et du chaînage des fonctions réseau virtuelles sur le respect des exigences de qualité de services, particulièrement lorsqu'un des points finaux de la chaîne VNF demeure indéterminé avant le processus de placement.

Cet article a été publié dans la revue (journal) scientifique **IEEE Transactions on Network and Service Management (IEEE TNSM), Volume 15, Issue 2, June 2018.** IEEE TNSM est classé Q1 dans la catégorie "Computer Networks and Communications" (SJR 2021) avec un facteur d'impact sur 5 ans de 6.048 (H-Index : 57).

- Dieye, M., Ahvar, S., Sahoo, J., Ahvar, E., Glitho, R., Elbiaze, H., & Crespi, N. (2018). **CPVNF : Cost-efficient proactive VNF placement and chaining for value-added services in content delivery networks.** *IEEE Transactions on Network and Service Management, 15(2), 774-786.* DOI: 10.1109/TNSM.2018.2815986

Un article connexe à l'article présenté ci-haut a été publié dans la revue (conférence) scientifique **IEEE 4th conference on network Softwarization and workshops (NetSoft)**, June 2018.

- Ahvar, S., Sahoo, J., Ahvar, E., Dieye, M., Glitho, R., Elbiaze, H., & Crespi, N. (2018, June). **PCPV : Pattern-based Cost-efficient Proactive VNF placement and chaining for value-added services in content delivery**

networks. In *2018 4th IEEE conference on network Softwarization and workshops (NetSoft) (pp. 313-317).*
DOI: 10.1109/NETSOFT.2018.8459986

Les articles publiés dans IEEE TNSM et IEEE Netsoft sont examinés par des pairs conformément aux exigences énoncées dans le manuel d'opérations de l'IEEE PSPB (sections 8.2.1.C & 8.2.2.A). Chaque article publié a été révisé par un minimum de deux réviseurs indépendants utilisant un processus de révision par les pairs en simple aveugle, où l'identité des réviseurs n'est pas connue des auteurs, mais les réviseurs connaissent l'identité des auteurs. Les articles sont soumis à un contrôle de plagiat avant d'être acceptés.

ABSTRACT

Value-added services (e.g., overlaid video advertisements) have become an integral part of today's content delivery networks (CDNs). To offer cost-efficient, scalable, and more agile provisioning of new value-added services in CDNs, network functions virtualization paradigm may be leveraged to allow implementation of fine-grained services as a chain of virtual network functions (VNFs) to be placed in CDN.

The manner in which these chains are placed is critical as it both affects the quality of service (QoS) and provider cost. The problem is however, very challenging due to the specifics of the chains (e.g., one of their end-points is not known prior to the placement).

We formulate it as an integer linear program and propose a cost efficient proactive VNF placement and chaining algorithm. The objective is to find the optimal number of VNFs along with their locations in such a manner that the cost is minimized while QoS is met. Apart from cost minimization, the support for large-scale CDNs with a large number of servers and end-users is an important feature of the proposed algorithm.

Through simulations, the algorithm's behavior for small-scale to large-scale CDN networks is analyzed.

Keywords: Servers, Quality of service, Licenses, Network function virtualization, cloud computing, integer linear programming, virtualization, content delivery networks, value-added services, virtual network functions, large-scale CDNs, video advertisements, agile provisioning, proactive VNF placement,

3.1 Introduction

Content Delivery Networks (CDNs) are largely distributed infrastructures of surrogate servers placed in strategic locations (Pallis et Vakali, 2006; Mukaddim Pathan et al., 2008). Content is replicated on these servers in order to serve end-users with reduced latency. Beyond hosting content, the popularity of CDNs as a platform for delivering value-added services has increased over the years.

Many CDN providers such as Akamai, Limelight, etc. offer value-added services. Currently around 47% of Akamai's revenues come from its value-added service offerings, which also provide higher margins compared to its basic CDN services (Trefis, 2015). Typical value-added services provided by CDNs include website/application acceleration (e.g., route optimization, TCP optimization, stream splitting) (M. Pathan et al., 2014), analytics, content protection, advertisement overlays/tickers and content adaptation (e.g., transcoding, compression). Website/application acceleration relies on a combination of optimization techniques and allows end-users (mostly enterprises) to access a website/application with improved response time. CDN analytics services provide awareness at various levels including network, device and content, thereby enabling content providers to make more informed and business-friendly decisions.

Increasingly, multimedia content providers are finding it useful to outsource their media-related services such as digital rights management and content adaptation to the CDN provider domain. This is because providing such services requires the content provider to store customized content for every end-user. The overlay/ticker service is perceived as a fast monetization strategy by content providers with, for instance local news/weather updates inserted on the top of the video as static or scrolling tickers.

Similarly, short video advertisements are linearly overlaid on the video delivered to end-users. Traditional CDNs face numerous obstacles towards efficiently provisioning value-added services. First, the value-added services are deployed on a dedicated hardware in the CDN infrastructure. As a result, it is both time-consuming and expensive to deploy and manage the service, resulting in more time to market and cost-inefficiency. Second, the explosive growth in end-users and amount of content (Cisco, 2015) delivered to them raise the need to scale the deployed services as needed. Third, end-users' growing interest in customized content fueled by a continual innovation in video formats over the years requires rapid provisioning of novel video-based value-added services.

Network Functions Virtualization (NFV) (Han et al., 2015 ; Mijumbi et al., 2016) is an emerging paradigm that can be used to make CDNs meet the above-mentioned requirements. NFV is a novel telecommunication service provisioning approach in which the network function is decoupled from the physical devices on which they run and are implemented as virtualized software, termed as virtual network functions (VNFs) which run on top of a virtualized infrastructure and chained together to provide a required service. They can be implemented on any computational node (e.g., CDN surrogate server, switches, data center) that meets their resource demand. The computational nodes must provide Network Functions Virtualization Infrastructure (NFVI) functions to support the execution environment of VNFs. These nodes are referred to as NFVI nodes. It should be noted that NFV has been traditionally used to virtualize middle-boxes (e.g., firewall, Network Address Translator (NAT) and Deep Packet Inspection (DPI)) (Martins et al., 2014).

Recently, NFV has been investigated in other domains such as virtualized Wireless Sensor Networks (vWSNs) (Mouradian et al., 2015), optical networks (Muñoz et al., 2015) and IP Mutimedia Subsystems (IMSs) (Carella et al., 2014). NFV could enable CDNs to provision value-added services with significantly lower deployment and maintenance costs. Thanks to virtualization techniques, NFV allows the value-added services to scale in an elastic manner. Moreover, due to the dynamic service chaining feature of NFV, the update of an existing value-added service or the introduction of new value-added services could be achieved with increased agility by inserting and/or removing VNFs to and from an existing service chain on-the-fly. We term the CDN architecture that relies on NFV as NFV-based CDN.

The placement and chaining of VNFs affects the desired QoS (e.g., delay) of a value-added service and the cost of the CDN provider. It is modeled as an optimization problem where the objective is to find the optimal number and location as well as efficient chaining of VNFs instances such that the CDN provider cost is minimized and QoS is satisfied.

In this paper, the cost includes the license, computing and communication cost. Where as the license cost includes instances and sites license and is computed based on the number of utilized VNF instances and sites (i.e., the servers). The computing cost includes the cost of running VNFs on servers and the communication cost is defined as the sum of the bandwidth used by the chains in the network. Although, some studies (e.g., (Bari et al., 2015)) consider the cost of transferring, booting and attaching a VM image to devices before deploying a VNF, this research does not consider this cost in the CDN environment as it is assumed that the CDN provider is the NFVI owner. In this paper, we focus on a proactive placement of VNFs where VNFs are deployed in an optimal manner before any request is received from the end-users to access the service. This type of deployment is triggered when a content provider requests the CDN provider to deploy a set of value-added services. The request specifies the service-related parameters (e.g., the description of functionalities that constitute the services) including the QoS threshold to be satisfied.

The rest of the paper is organized as follows : The next section illustrates the problem via a use case, introduces the key requirements and reviews related works. Section III presents the VNF placement problem in CDNs and ILP formulation. Section IV describes the proposed algorithm. Section V portrays the simulation results. Section VI concludes the paper and outlines some future works.

3.2 Illustrative Use Case, Requirements and Related Work

To the best of our knowledge, the only paper to discuss a use case for VNF–based value-added service provisioning in CDN is (Jahromi et al., 2017) which focuses on the architectural aspects while we focus on the algorithmic aspects. The European Telecommunications Standards Institute (ETSI) has also proposed a use case on the virtualization of CDNs entities (e.g., surrogate server, CDN controller, etc.) (*ETSI GS NFV 001, Network Function Virtualization (NFV) Use cases, V1.1.1.* 2013), but with no bearing on value-added service provisioning. Besides, few NFV architectures for CDN have been proposed in (Giotis et al., 2015) and (Herbaut, Negru, Xilouris et al., 2015), yet with no bearing on value-added services provisioning.

3.2.1 Illustrative Use Case

We assume a business model with the following entities : Content provider (e.g., YouTube), CDN provider (e.g., Limelight), VNF provider and end-users. Content provider provides the value-added services to its end-users. The CDN provider owns surrogate servers and operates NFVI on the surrogate servers. The VNF provider provides VNFs. End-user consumes the value-added service. The reader should note that like in any business model, the same actor may play several roles

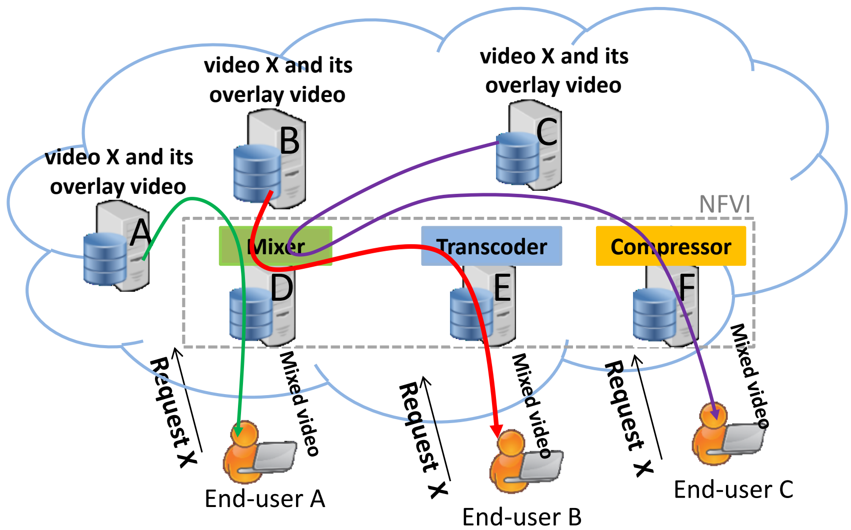

Figure 3.1 Video Advertisement Overlay Use case

at the same time.

Let us consider a simplified scenario as shown in Figure 3.1. It depicts three end-users A, B and C, with mobile devices of different capabilities (e.g., supported codecs, resolutions, bandwidth and processing power) who wish to access a video X. As per CDN principles, the video is duplicated over several servers (i.e., A, B, and C in this case). When the end-users request the video, the CDN provider adds an overlay video to it as a value-added service. Such an overlaid video enriches the viewing experience of end-users by providing a clip of advertisement based on the end-user's interest. In this use case, it is assumed that video X and the video which is overlaid on it reside on the same servers. It should be noted that the video overlay service requires a video mixer functionality to embed the advertisement in the main video.

The end-users may however not be able to play the mixed video on their devices depending on their capabilities. In this scenario, it is only end-user A who is able to play the mixed video. End-user B does not have the required codec, therefore the video must goes through a transcoder before being delivered to her/him. As for end-user C she/he has a limited bandwidth capacity, and therefore, its device cannot play the video in its original resolution. Consequently, the mixed video needs to be compressed before delivery. Three fine-grained functionalities are therefore required this value added service : mixer, transcoder and compressor. Their deployment starts when the content provider sends a request to the CDN provider to deploy the video value-added services. The CDN provider then gets them from the VNF provider and uses a VNF placement algorithm to deploy them on a selected subset of surrogate servers to optimize given criteria (e.g., QoS) delay when the video is viewed. The reader should note that surrogate A, B, and C which will serve end-user A, B and C, respectively are not known when the deployment is made. They will be selected dynamically by the CDN controller server when the end-users will request the services.

3.2.2 Requirements

Achieving optimal placement of VNFs is an issue in provisioning value-added services in CDNs because of the inherent goals associated. First, the services must be delivered to end-users with high QoS (e.g., reduced service delay). It is in fact desirable to provide the service with a strict QoS guarantee. Second, the service placement should be operational in small and large scale CDNs. Third, the cost incurred in deploying services and delivering them to the end-users must be minimized for cost-efficiency. Thus, the optimal placement of VNFs in CDN consists of determining how many VNF instances are required to meet the end-user workload and on which surrogate servers they should be deployed so that the above-mentioned goals are reached. The placement of VNFs in CDN can be achieved in two ways : 1) Proactive, and 2) Reactive. This paper focuses on the proactive placement. The general requirements for the proactive placement are summarized as follows :

- To ensure QoS, especially in terms of service delay
- To ensure low cost
- To be operational in small and large scale CDNs

3.2.3 Related Work

Although VNF placement algorithms constitute a large area of interest for many researchers, very few of them have tackled the problem of placing VNFs in CDNs. Here, we first summarize the main research results on VNF placement in general and also on VNF placement in CDNs. We then show why these results are not applicable to our specific problem of placing VNFs for VAS in CDNs.

3.2.3.1 VNF placement algorithms in general

VNF placement algorithms constitute a large area of interest for many researchers. Here, we first summarize the main results in the area, and then, show why they are not applicable to the CDN context.

Most VNF placement algorithms deal with cost as an optimization objective. A VNF cost is generally made up of a set of individual costs (e.g., instance license, site license, deployment and communication cost). Some of the existing work focus on specific individual costs while others focus on a set of individual costs. They are discussed below.

Algorithms with single costs as objective : Luizelli et al. (Luizelli et al., 2015) propose an ILP model to minimize the number of instances in order to reduce license costs. They further propose a binary search-based algorithm to improve the ILP run-time. Fang et al. (Fang et al., 2016) also attempt to minimize the number of deployed instances by proposing an ILP and the longest common sub-sequence (LCS)-based heuristic in inter-datacenter elastic optical networks (inter-DC EONs). They also consider the spectrum utilization cost for fiber links, which is the specialized cost for optical network. Moens and Turck (Moens et Turck, 2014) present an ILP to minimize the number of used servers (or compute resources) for the resource allocation of VNFs in NFVI and also for hybrid infrastructures where some NFs are virtualized and others use specific hardware appliances.

Some other studies have mainly focused on the communication cost. Qu et al. (Qu et al., 2016) propose an ILP and a greedy shortest-path-based heuristic to construct chains through highly reliable VNFs in the NFV-enabled enterprise datacenter networks with the goal of minimizing the communication bandwidth usage across the network. Xia et al. (Xia et al., 2015) formulate the problem in binary integer programming (BIP) and propose a heuristic with the goal of minimizing the overall optical-electrical-optical (O/E/O) conversions (inter-DC traffic) in packet/optical DCs.

Algorithms for multiple costs : Unlike the above-mentioned works with the simple objective, the following studies have considered more complex cost models. Ghaznavi et al. (Ghaznavi, Khan et al., 2015) present a solution called Simple

Lazy Facility Location (SLFL) to optimize the placement of the same-type VNF instances in response to the on-demand workload. In this study, the elasticity overhead and the trade-off between bandwidth and host resource consumption are considered together.

In another study, Ghaznavi et al. (Ghaznavi, Shahriar et al., 2016) propose a Mixed Integer Programming (MIP) model and a heuristic called Kariz for multiple VNF instances placement to provide the functionality of a middlebox.

Mechtri et al. (Mechtri et al., 2016) provide decomposition-based approach for the placement of virtual and physical network functions chains to maximize the provider's revenue based on the number of accepted CPU and bandwidth resources.

Riggio et al. (Riggio, Rasheed et al., 2015) propose a VNF placement scheme to minimize the links and nodes utilization to increase the accepted service chain requests in enterprise WLANs. The authors then have extended their work in (Riggio, Bradai et al., 2016) where a VNF placement heuristic called WiNE (Wireless Network Embedding) is proposed.

Sun et al. (Sun et al., 2016) consider cost as the IT resources used for deploying the VNFs and bandwidths cost. They propose an ILP as well as two versions of a heuristic to solve the VNF placement in online and offline manners. The goal in the online heuristic is to maximize the revenue and the goal in the offline version is to minimize the cost.

Finally, a few studies have attempted to consider more comprehensive cost models. Lin et al. (Lin et al., 2016) present a MILP and Game Theory based VNF placement with the goal of minimizing the cost to deploy NF instances as well as the computing and network cost in optical networks.

Zeng et al. (Zeng et al., 2016) consider the cost of IT resource and spectrum utilization of fiber links as their objective in addition to the cost of VNF deployment (instantiating) in the VNF placement in optical datacenters. They propose a MILP and some heuristics to solve the problem.

Bouet et al. (Bouet et al., 2015) propose an ILP and a centrality-based greedy algorithm to minimize the cost in virtual DPI (vDPI) placement where the cost includes the network cost, the license cost per site and the one per vCPU for VNF instances.

Bari et al. (Bari et al., 2015) propose an ILP and a heuristic to solve the optimal VNF placement by running the Viterbi algorithm. The authors have also considered a penalty cost to be paid to the customer for the service level objective (SLO) violations.

3.2.3.2 VNF placement algorithms in CDNs

Herbaut et al. (Herbaut, Negru, Dietrich et al., 2017) is an example of works that deal with the algorithmic solutions to the VNF placement in CDNs. It focuses on CDN operators who aim to deploy surrogate servers as VNFs in Internet Service Providers' (ISP) network. The ISP provides the NFVI Point of Presence (PoP) for the deployment. However it maintains the confidentiality of its network and resources by offering an abstract view to the CDN operator. The CDN operator just has access to an overlay that connects its end-users to the surrogate servers which run as VNFs in the ISP network. It does not know where exactly these surrogate servers are deployed. A high level Service Level Agreement (SLA) is used for the negotiation of computing and connectivity resources between the CDN operators and the ISP. This SLA is expressed as a service function chain. The paper discusses the collaboration model between the ISP and CDN operator, proposes an ILP formulation for the service chain embedding problem, and also a heuristic to increase problem tractability.

Ibn-Khedher et al. (Ibn-Khedher et al., 2017) is another example. It proposes OPAC, an optimal placement algorithm for virtual CDNs. An exact algorithm is proposed and evaluated. The algorithm takes as an input the topology of the underlying network, and optimally places, and migrates the virtual surrogate servers in order to increase user satisfaction and decrease server and network loads. A few works have addressed the architectural solutions to the VNF placement problems in CDNs.

Frangoudis et al. (Frangoudis et al., 2017) details the design and the implementation of an architecture that allows a telecommunication operator to lease its infrastructure to content providers for the deployment of surrogate servers as VNFs. The focus is on the northbound REST APIs used for the interactions between the content providers and the telecommunication operator. The functional entities of the architecture are also specified. Some examples are the Customer Interface Manager which exposes the actual REST APIs, the Service Orchestrator which coordinates the actual deployment. Algorithmic issues are not explicitly excluded from the scope of the work.

3.2.3.3 Why are previous work not adequate for the problem at hand ?

The previous work is not adequate for the problem at hand for two main reasons. The first reason is that VNF placement in CDN for value-added service provisioning is fundamentally different from the VNF placement as considered so far by researchers, be it generally or in the specific case of CDNs. This due to the fact that in previous works, each chain has a distinct pair of end-points (i.e., source and destination) which both are known prior to the VNF placement. In (Pallis et Vakali, 2006) and (Mukaddim Pathan et al., 2008) for instance which deal with the specific case of CDNs, end-users are assigned to specific surrogate servers prior to the placement. In other words the specific surrogate server which will serve any given end-user is known prior to the placement. However, in this work while the destination (i.e., the end-user) is known, the source (i.e., the surrogate server which will serve the end-user) is not known prior to the VNF placement because it is dynamically selected by the CDN controller after the VNF placement, Going back to the illustrative use case, while the locations of end-users A, B, C are known as the destinations, there is no way to know prior to the placement that surrogate server A will be ultimately selected by the CDN controller to serve end-user A. The same applies to the choice of surrogate server B for end-user B, and surrogate server C for end-user C. This brings unique challenges to the service chain placement in CDNs. To the best of our knowledge, our work is the first VNF placement scheme where an end-point of a service chain cannot be known prior to the placement.

The second reason is that some of the previously reviewed works do not meet all our requirements. The ILPs proposed by (Luizelli et al., 2015) and (Moens et Turck, 2014) for instance are not suitable for large scale environments. References (Xia et al., 2015 ; Ghaznavi, Khan et al., 2015 ; Ghaznavi, Shahriar et al., 2016 ; Mechtri et al., 2016) do not consider QoS (i.e., service delay threshold) and sharing VNFs among the chains. References (Luizelli et al., 2015 ; Fang et al., 2016 ; Moens et Turck, 2014 ; Qu et al., 2016 ; Xia et al., 2015 ; Ghaznavi, Khan et al., 2015 ; Ghaznavi, Shahriar et al., 2016 ; Mechtri et al., 2016 ; Riggio, Rasheed et al., 2015 ; Riggio, Bradai et al., 2016 ; Sun et al., 2016 ; Lin et al., 2016 ; Zeng et al., 2016 ; Bouet et al., 2015) do not take a complete cost model into account. Unlike them, our work is appropriate for a large scale environment while considering all the above-mentioned points.

3.3 VNF Placement Problem

3.3.1 Problem Description

This paper focuses on the proactive placement of VNFs. The placement problem is formalized as follows : Content providers request the CDN provider to deploy a set of value-added services. The request specifies the service-related parameters (e.g., the description of functionalities that constitute the services) including the QoS threshold to be satisfied.

For ease of reading, we specifically denote the surrogate servers containing content as content servers. Therefore, given a content X, the content servers containing X, a set of end-users requesting the content, their workload and a set of services to be accessed by the end-users, the VNF placement and chaining problem consist of : (i) jointly finding the number and location of surrogate servers to host the VNF instances and the content servers having content X, (ii) chaining the VNFs and connecting end-users, while minimizing the cost of CDN provider and satisfying the QoS threshold of all services offered to the end-users.

The reader should note that for each end-user, the delay for viewing the video is the sum of the following delays :

- the delay from the content server (selected to stream the video) to the surrogate server hosting the first VNF of the chain,
- the delay for transmitting and processing the video in the chain, and
- the delay from the surrogate server hosting the last VNF of the chain to the end-user.

As the first delay is unknown when the VNFs are placed, the total delay might violate QoS requirements when end-users access value added services.

In this paper, we jointly optimize the selection of serving surrogate servers and VNF placement. It is therefore assumed that the serving surrogate servers are selected by new functional entity the "VAS serving surrogate server selector" instead of the traditional CDN controller. The impact on CDN functioning remain minimal since this new entity (which incorporates the results of our novel joint optimization algorithm) might be incorporated in the traditional CDN controller.

Our problem is a slight variation of the well-established Bin Packing problem (Monaci et Toth, 2006) and the Hierarchical Facility Location-Allocation problem(Teixeira et Antunes, 2008 ; Farahani et al., 2014).

Our problem can be an extension of the Bin Packing problem to Bin Packing plus Packer problem, including selecting the bins (i.e., surrogate servers) to drop the items (i.e., VNFs). The capacity of the bins and the load (i.e., end-user workload for a given VNF) of the items are known. The goal is to optimize the sum of the cost of the bins and the communication cost between the packers (i.e., end-users) and the bins.

In the Hierarchical Facility Location-Allocation problem, the sites are the equivalent of surrogate servers and each demand location is equivalent to end-user locations in our case. The facilities to be placed are equivalent to the VNFs in our case.

The problem is decomposed into two phases : The location phase and the allocation phase, as follows : The location phase consists of determining the number of required facilities and the site to locate them. The allocation phase is for allocating the corresponding demand to each facility in the last layer (i.e., the last VNF of the VNF chain in our case). Besides, in the allocation phase, each facility (i.e., VNF) in each layer is allocated to one of the facilities (i.e., VNF) in the previous layer, which is equivalent to placing the instances of a VNF chain by connecting surrogate servers.

The main difference between the Hierarchical Facility Location-Allocation problem and our VNF placement problem is that, in the former problem, the location of the primary level facilities (i.e., the first VNFs of the VNF chains in our case) are known beforehand, whereas, in our VNF placement problem, it is not.

3.3.2 ILP Formulation

Let us consider N as a set of surrogate servers, W as a set of content servers and U as a set of end-users. The physical topology of the network is represented by a directed graph $G = (V, E)$, where $V = N \cup W \cup U$ is the set of nodes composed of the surrogate servers, content servers and end-users connected by directional edges E.

Let us also consider K as a set of VNFs of different types, such as Video Transcoder, Video Mixer and Video Compressor (see Section 3.2.1). Each VNF of type $k \in K$ has a predetermined resource requirement and processing capacity, R_k and P_k , respectively. The VNF requirement (R_k) can be a vector of vCPU, memory, disk, and I/O bandwidth (i.e., VNF processing capacity). If we assume that the bandwidth of the links connected to the NFVI node is less than the input/output bandwidth of that NFVI node, considering the I/O bandwidth in the requirement vector of VNFs avoids link overload. The CDN provider also delineates a set of instances for each VNF type $k \in K$, I_k. Such specification may be the result of the license model adopted while acquiring the VNFs from the VNF provider or as a result of management restrictions.

A service request $h_f \in H$, generated by an end-user $f \in U$, is represented by a directed graph $\mathcal{G}(\mathcal{V}^f, \mathcal{E}^f)$, where $\mathcal{V}^f$ is the set of VNFs that will be installed on nodes in N and $\mathcal{E}^f$ is the set of virtual edges. For the sake of simplicity, we further define the following sets defined by :

$$\begin{aligned}
\Lambda^f &: \left\{a \mid a \text{ is first element of VNF chain } h^f\right\} \\
\Delta^f &: \left\{z \mid z \text{ is last element of VNF chain } h^f\right\} \\
\Phi^f &: \left\{n \in W \mid w_n^f = 1\right\}
\end{aligned}$$

where we denote respectively Λ^f as the singleton containing the first VNF within the VNF chain h_f, Δ^f as the singleton containing the last VNF within the VNF chain and Φ^f as the set containing content servers which host the requested content for the service request $h_f \in H$.

Table 3.1 delineates the inputs and variables used in our ILP formulation.

TABLE 3.1: Input Parameters and Variables

Symbol	**Meaning**
Network Inputs	
N	Set of surrogate servers in the network, $N \subseteq V$
U	Set of end-users in the network, $U \subseteq V$
W	Set of content servers in the network, $W \subseteq V$
E	The set of edges (i.e., logical communication links) in the network
$BW_(u, v)$	The bandwidth capacity of edge $(u, v) \in E$
$D_(u, v)$	Delay of unit load (1 Gbps) for edge $(u, v) \in E$
$\sigma(u, v)$	Hop count of the edge $(u, v) \in E$
B_n	Bandwidth cost (in dollars) of unit load (1 Gbps) per hop from surrogate server n
	Continued on next page

Table 3.1 – continued from previous page

Symbol	Meaning
$\beta^f_(u,v)$	Bandwidth cost incurred by sending load of end-user f along edge $(u,v) \in E$
C_n	Capacity of surrogate server $n \in N$ in terms of resource units
γ	Site license cost
δ_n	Operational cost for unit resource (vCPU) for surrogate server n
Service Inputs	
H	Set of services h_f requested by user $f \in U$
K	Set of VNF types that constitute all services $h_f \in H$
$\mathcal{V}^f$	$\mathcal{V}_f \subseteq K$ is a set of VNFs that constitute service $h_f \in H, f \in U$
$\mathcal{E}^f$	Set of VNF edges of the VNF chain for service $h_f \in H$, $f \in U$
Δ^f	The first VNF in the VNF chain for service $h_f \in H$, $f \in U$
Λ^f	The last VNF in the VNF chain for service $h_f \in H$, $f \in U$
Φ^f	Set of content servers hosting content requested for service $h_f \in H$, $f \in U$
I_k	Set of VNF instances of type $k \in K$
α_k	Software license cost of a VNF instance of type $k \in K$
T_k, n	Processing delay of VNF instance of type $k \in K$ on surrogate server $n \in N$ for unit load (1 Gbps)
R_k	Resource requirement for VNF type $k \in K$
P_k	Processing capacity (Gbps) of VNF type $k \in K$
L_f	Load of end-user $f \in U$
D_{h_f}	QoS (i.e., Service Delay),threshold of service $h_f \in H$
Variables	

Continued on next page

Table 3.1 – continued from previous page

Symbol	Meaning
x_k, n, j	1, if instance j of VNF type k is assigned to surrogate server $n \in N$ and 0, otherwise
λ_k^f, n, j	1, if VNF type k belonging to VNF chain of end-user f is mapped to its instance j on surrogate server n and 0, otherwise
y_u, v^f, p, q	1, if edge (u, v) hosts VNF edge (p, q) of VNF chain of end-user f and 0, otherwise
z_n	1, if surrogate server n is used and 0, otherwise
w_n^f	1, if a content server n has the content requested for service $h_f \in H$ and 0, otherwise

Objective function :

$$\mathbf{min} \sum_{\substack{\forall k \in K \\ \forall n \in N \\ \forall j \in I_k}} x_{k,n,j} \cdot \alpha_k + \sum_{\substack{\forall f \in U \\ \forall (p,q) \in \mathcal{E}^f \\ \forall (u,v) \in E}} \beta^f_{(u,v)} \cdot y^{f,p,q}_{u,v} +$$
$$\sum_{\substack{\forall f \in U \\ \forall k \in \Lambda^f \\ \forall w \in \Phi^f \\ \forall n \in N \\ \forall j \in I_k \\ \forall (w,n) \in E}} \beta^f_{(w,n)} \cdot \lambda^f_{k,n,j} + \sum_{\substack{\forall f \in U \\ \forall k \in \Delta^f \\ \forall n \in N \\ \forall j \in I_k \\ \forall (n,f) \in E}} \beta^f_{(n,f)} \cdot \lambda^f_{k,n,j} + (z_n \cdot \sum_{\forall n \in N} (\gamma + \sum_{\substack{\forall k \in K \\ \forall j \in I_k}} R_k \cdot \delta_n \cdot x_{k,n,j})) \tag{1}$$

QoS guarantee :

$$\sum_{\substack{\forall (p,q)\in\mathcal{E}^f\\ \forall (u,v)\in E\\ \forall f\in U}} D_{(u,v)}\cdot L_f\cdot y^{f,p,q}_{u,v}+\sum_{\substack{\forall k\in\mathcal{V}^f|\\ \forall i\in\mathcal{V}^f\\ \exists (k,i)\in E_f\\ \forall u\in N\\ \forall j\in I_k\\ \forall f\in U}} T_{k,u}\cdot L_f\cdot\lambda^f_{k,u,j}+\sum_{\substack{\forall f\in U\\ \forall k\in\Lambda^f\\ \forall w\in\Phi^f\\ \forall n\in N\\ \forall j\in I_k\\ \forall (w,n)\in E}} (T_{k,n}+D_{w,n})\cdot L_f\cdot\lambda^f_{k,n,j}+$$

$$\sum_{\substack{\forall f\in U\\ \forall k\in\Delta^f\\ \forall n\in N\\ \forall j\in I_k\\ \forall (n,f)\in E}} (T_{k,n}+D_{n,f})\cdot L_f\cdot\lambda^f_{k,n,j}\preceq D_{h_f} \quad (2)$$

Our objective is to minimize the cost of VNF placement for value-added services in CDN, including the cost of deploying VNF instances, the cost of using surrogate servers and the cost of communication as shown in (1).

The cost of deploying VNFs is characterized by a software license cost per instance denoted α_k. The cost of using a surrogate server $n \in N$ is the sum of a fixed license cost γ and the operational costs for all VNF type instances. The license cost is the same for every surrogate server. The operational cost for a VNF instance of type k is $R_k \cdot \delta_n$.

The cost of communication in VNF placement for value-added services is the sum of the bandwidth costs amongst each pair of surrogate servers $u, v \in N$ hosting VNFs of the VNF chain in each end-user service request $h_f \in H$,$f \in U$, $\beta^f_{(u,v)}$. It further includes the bandwidth costs $\beta^f_{(n,f)}$ between the surrogate servers $n \in N$ hosting the tail VNFs of the VNF chain to an end-user $f \in U$ and the bandwidth costs $\beta^f_{(w,n)}$ between the surrogates servers $n \in N$ hosting the head VNFs of the VNF chain to a content server $W \in \Phi^f$ hosting the content requested by the user $f \in U$.

Note that we compute the cost of using bandwidth $\beta^f_{(u,v)}$ between $u \in V$ and $v \in V$ using $\beta^f_{(u,v)} = L_f \cdot \sigma_{(u,v)} \cdot B_u$ with hop count $\sigma_{(u,v)}$, for load L_f for end-user $f \in U$.

Content server selection :

$$\sum_{n \in W \,|\, n \in \Phi^f} w_n^f = 1, \quad \forall h_f \in H \tag{3}$$

VNF Placement :

$$\sum_{n \in N} \sum_{j \in I_k} \lambda_{k,n,j}^f \geq 1, \quad \forall f \in U, k \in K \tag{4}$$

$$\sum_{f \in U} L_f \cdot \lambda_{k,n,j}^f \preceq P_k \cdot x_{k,n,j} \quad \forall k \in K, n \in N, j \in I_k \tag{5}$$

$$\sum_{k \in K} \sum_{j \in I_k} R_k \cdot x_{k,n,j} \preceq C_n \cdot z_n \quad \forall n \in N \tag{6}$$

VNFs chain mapping :

$$\sum_{j \in I_p} \lambda_{p,u,j}^f \cdot \sum_{j' \in I_q} \lambda_{q,v,j'}^f = y_{u,v}^{f,p,q} \quad \forall f \in U, (p,q) \in \mathcal{E}^f, (u,v) \in E \tag{7}$$

$$\sum_{(u,v) \in E} y_{u,v}^{f,p,q} = 1 \quad \forall f \in U, (p,q) \in \mathcal{E}^f \tag{8}$$

$$\sum_{f \in U} \sum_{(p,q) \in \mathcal{E}^f} L_f \cdot y_{u,v}^{f,p,q} \preceq BW_{(u,v)} \quad \forall (u,v) \in E \tag{9}$$

Variables $x_{k,n,j}$ are used to identify unique instance $j \in I_k$ of VNF type $k \in K$ installed on surrogate server $n \in N$. Variables z_n are used to record surrogate servers hosting VNFs.

Constraint (2) is the QoS constraint that guarantees that the end-user service request is delivered within the predefined delay threshold. The delay in delivering the service consists of two components, the network communication delay and the VNF processing delay on surrogate servers.

The first term of constraint (2) calculates the network communication delay as the sum of delay between each pair of surrogate servers hosting the VNFs of the VNF chain in the end-user service requests.

The second term of constraint (2) denotes the processing delay of VNFs hosted surrogates servers, which is proportional to end-user load assigned to the VNFs.

In the third term, we consider the communication delay $D_{w,n}$ between the content server $w \in \Phi^f$, selected to serve the service request and the surrogate server $n \in N$ hosting the head VNF of the VNF chain.

We further include the variable $T_{k,n}$ to denote the processing delay incurred at the surrogate server $n \in N$. Similarly, the fourth term represents the communication delay $D_{n,f}$ between the surrogate servers $n \in N$ hosting tail VNFs to the end-users $f \in U$. Naturally, we also take into account the processing delay $T_{k,n}$ incurred at the servers n during the transmission to the end-users f.

Hence, constraint (2) represent the end-to-end service delay from content server selection until end-user delivery.

We also ensure through constraint (3) that only one content server $n \in W$ is selected to serve the request from end-user $f \in U$.

We guarantee in (4) that, for each end-user service request $h_f \in H$, $f \in U$, an instance of the requested VNF type must be assigned to a surrogate server. It is assumed that an instance of a VNF type accommodates the load from an end-user.

Though an instance of a VNF type can cater to multiple users, it must satisfy at least one end-user load. This can be trivially extended to split larger loads of an end-user into smaller chunks, since our formulation allows end-users f and f' to request the same service, that is $h_f = h_{f'}$.

Constraint (5) ensures that the capacity of an instance of a VNF of type $k \in K$ is not exceeded by the total load requested by all the end-users assigned to it while constraint (6) ensures that the total resource required by instances of all VNF types does not exceed the capacity of the host surrogate server.

In our model, we map the nodes (i.e., the VNFs) and their edges (i.e., the chain), in each end-user service request to the physical network in G. Therefore, the edge (p, q) between two consecutive VNFs in each end-user service request must be assigned to a physical edge (u, v) between two surrogate servers u and v, in (7). It should be noted that (7) is a non-linear constraint and can be trivially linearized by replacing it with linear constraints (10)-(12) as follows.

$$y_{u,v}^{f,p,q} \preceq \sum_{j \in I_k} \lambda_{p,u,j}^{f} \quad \forall f \in U, (p,q) \in \mathcal{E}^f, (u,v) \in E, u \in N, v \in N \tag{10}$$

$$y_{u,v}^{f,p,q} \preceq \sum_{j \in I_k} \lambda_{q,v,j}^{f} \quad \forall f \in U, (p,q) \in \mathcal{E}^f, (u,v) \in E, u \in N, v \in N \tag{11}$$

$$y_{u,v}^{f,p,q} \succeq \sum_{j \in I_k} \lambda_{q,v,j}^{f} + \sum_{j \in I_k} \lambda_{p,u,j}^{f} - 1 \quad \forall f \in U, (p,q) \in \mathcal{E}^f, (u,v) \in E, u \in N, v \in N \tag{12}$$

As is ensured in (8), the VNFs and their respective ordered edges are mapped to only one pair of physical surrogate servers and their edge. With respect to the underlying physical network, we guarantee that the total load on an edge in the physical network does not exceed the bandwidth capacity, in (9).

The VNF placement and chaining for value-added services in CDN is an NP-Hard problem, calling for an efficient heuristic.

3.4 Cost-efficient Proactive VNF Placement (CPVNF)

In this section, we discuss the design choices and insights for our Cost-efficient Proactive VNF Placement heuristic for deploying value-added services in CDN. There are three main constraints for VNF placement problem that should be considered : satisfying QoS as well as preventing VNF and server overloading.

We base our heuristic on the PageRank algorithm (Brin et Page, 1998) pioneered on the Google search engine. It can be described as a variant of the eigenvector centrality method. PageRank has also been known to perform well in scale-free networks and thus has been widely employed in many fields (Cheng et al., 2011).

In summary, page ranking leverages the idea that a Web page's importance is a factor of both the quantity and the quality of the pages linked to it. Comparatively, a surrogate server's fitness to host a VNF instance can also be viewed as dependent of the quality of the surrogate server itself (e.g., capacity, processing power, energy consumption, etc.,) as well as the quality and quantity of outgoing links towards other VNF instances in the VNF chain.

We start by ordering every user request received at a given time period t based on its aggregated requirements (e.g., vcpu, processing capacity,etc.). This is done, in supposition that user requests with stricter QoS requirements may be harder to map at later stages in the network. In a sense that, as we map each request, the remaining surrogate capacity and edge bandwidth capacity lessens within the network, thereby rendering subsequent requests mapping more complex.

3.4.1 Selecting the content server

As previously mentioned in Section 3.2.1, a given content is duplicated over several content servers as per CDN principles. Hence, we must determine, upon choosing a surrogate server to host the first VNF in the VNF chain, the content server upon which the content is routed from. The choice of a content server to provide for a request is an important factor towards guaranteeing QoS requirements in terms of delay. Therefore, it also influences the fitness of the surrogate server hosting the first VNF.

Hence, we assume there is a convex function $Q : \mathbb{R} \rightarrow \mathbb{R}_+$, which computes the fitness of a content server from a surrogate server perspective and also acts as a penalty function penalizing content servers with large delays to the surrogate server hosting the first VNF. We hence define $Q(\cdot)$ as a quadratic function, widely used in the control theory literature (Bertsekas, 1976) such that

$$Q(w, n) = \mu \cdot (\frac{D(w, n)}{D_{h_f}})^2 \tag{13}$$

with $w \in \Phi^f$, (w, n) denoting the path with the least delay between $w \in \Phi^f$ and $n \in N$. Where $D(w, n)$ represents the delay between one of the content servers w and a surrogate server $n \in N$ candidate towards hosting the first VNF of the chain, D_{h_f} is the delay threshold for user request h_f and μ is a constant.

Note that in this context, the penalty function obviously penalizes content servers which predominantly contributes to the violation of the delay constraint (detailed below).

3.4.2 Selecting the surrogate servers

As evident from our objectives previously stated, we are motivated to guarantee SLA bounds on the QoS, with respect to end-user perceived latency as well as preventing VNF and server overloading. In this regard, it is critical for the VNF mapping process to place VNFs onto surrogate nodes which contribute most to this goal.

We therefore employ a surrogate importance rank metric (SIR), we note $\varphi_{k,n}$ to infer the relative fitness of a given candidate surrogate server n to host a VNF instance k in comparison to its peers. $\varphi_{k,n}$ indicates the surrogate server importance rank with regard to the current network topology properties such as the available bandwidth between edges, the remaining capacities of reachable surrogate servers, etc.

A surrogate server's SIR is determined notably by its remaining capacity, the aggregated remaining bandwidth of its outgoing links, the rank of its neighbours as well as a weight Γ to denote whether the nodes already hosts an instance of the VNF considered. The latter parameter is to help ensure that a minimal number of instances are used throughout the network in order to reduce costs.

We justify considering the number of outgoing links in our model as it also increases the probability to reach the requesting end-users and thus ensure that the location of the last VNF is accessible to the requesting end-user. Of course, all of this is done while always keeping in mind to respect QoS constraints. The actual value of the instance weight can be determined experimentally by finding a fair tradeoff between VNF instance consolidation, server license costs and success of QoS guarantee. We therefore consider

$$M_n = (\pi \cdot C'_n) \cdot \left[(1-\pi) \sum_{a \in A(n)} BW'(n,a) \right] \quad \forall n \in N \tag{14}$$

where C'_n and $BW'(n,a)$ respectively denote the remaining capacity at surrogate server n and the remaining bandwidth capacity between edge (n,a). The set $A(n)$ representing the surrogate servers adjacent to the surrogate server n.

The weight π is used to bias node selection by focus on surrogate servers' capacity or aggregate outgoing bandwidth depending on network topology state. Indeed, if we suppose a scenario where a majority of surrogates servers have high capacity, the VNF mapping should focus more on the quality of outgoing links towards other surrogates servers and vice-versa. Hence, these parameters can be seen as a intensification or diversification parameter during the search process of the most suitable candidate surrogate servers.

Upon launching our heuristic and before any user request is mapped, the initial values of each surrogate server's SIR is determined by :

$$\varphi_{k,n} = \frac{M_n}{\sum_{o \in N} M_o} \quad \forall n \in N \tag{15}$$

Once initialized, we make use of personalized pagerank where we bias towards nodes with preexisting instance of the vnf type k to reduce the number of surrogate servers used :

$$\varphi_{k,n} = \Gamma_k * \frac{1-\psi}{|N|} + \psi \cdot \left(\sum_{i \in A(n)} \frac{\varphi_{k,i}}{|A(i)|} \right) \tag{16}$$

with :

$$\Gamma_k = \begin{cases} \Gamma & \text{if } x_{k,n,j} = 1 \\ 1 & \text{otherwise} \end{cases}$$

where ψ denotes a damping factor usually set optimally to 0.85 (Son et al., 2012) and $|A(i)|$ the amount of neighbouring surrogates servers to server i. Note that the SIR values of each surrogate server can be effectively computed using both iterative and algebraic methods (Lawrence et al., 1998).

Once the SIR values computed, we proceed towards mapping the VNF chain in two phases : the VNF to surrogate server mapping and the VNF to VNF link mapping.

In the VNF to surrogate mapping, we first sort each surrogate server according to their SIR to determine their fitness. We attempt to place the VNFs of the VNF chain accordingly. However, keep in mind that we must take into account the selection of content server for the first VNF.

Hence, the selection of the surrogate server to host the first VNF in the VNF chain is done based not only on its SIR value but also on the penalty incurred by the content source as previously described in Section 3.4.1.

Otherwise, a VNF is placed on the best ranked surrogate server. After a VNF is placed on a given surrogate server, the SIR values of every surrogate server in the network is updated to reflect the current network topology state.

The VNF to VNF link mapping consists of finding the k-shortest paths between the surrogates servers hosting VNFs for the VNF chain. A given path is retained when it satisfies the specified delay requirements. Note that the delay between the last VNF of the chain and the requesting end-user is also included for the path choice. We summarize the VNF mapping process described above in Algorithm 1.

Algorithm 1: CPVNF

Input: Network topology, H :user requests, N : surrogate servers, *stop* : search limit

Output: VNF mapping

```
H' = {}
searchLimit = 0
for n ∈ N do
    compute initial SIR values using eq. 15
for h_f ∈ H do
    H'= rank (h_f ∈ H) based on added requirements
for h'_f ∈ H' do
    for k ∈ h'_f do
        if k : first VNF of the chain then
            for n ∈ N do
                compute SIR of n for VNF k using eq. 15-16
                compute Q(·) for every content server w using eq.13
            retain a host for VNF k, surrogate server n with highest compound score (SIR + 1/Q(w,n))
        else
            for n ∈ N do
                compute SIR of n for VNF k using eq. 16
                rank n based on SIR value
            if path between n and surrogate hosting previous vnf in chain then
                place VNF k on highest ranked n
            else
                select next ranked n
        for k, k' ∈ h'_f do
            find k-shortest paths of hosts of k and k'
            select path with lowest delay
        for k :last VNF in the chain do
            find k-shortest paths between k to end-user f
            select path with lowest delay
    if QoS constraints respected then
        restore π and κ to default values
        confirm removal of server and edges capacities continue to next request
    else
        if searchLimit < stop then
            decrease π redo for loop above
        else
            reject request
```

3.5 Performance Evaluation

This section describes our simulation settings and presents the results of the performance evaluation of our proposed proactive VNF placement solution in CDN networks. As mentioned earlier, this is the first contribution on VNF chaining placement in CDN where the service chain format boundary (i.e., source and destination of the service chain) is different. Thus, CPVNF cannot be compared with existing state-of-the-art algorithms.

Besides, as basic algorithms such as random and greedy VNF placements usually do not consider QoS (e.g., service threshold delay), therefore comparing CPVNF to them is not adequate. To evaluate the effectiveness of our proposed algorithm, we compare its performance against the optimal solution of the ILP model for a relatively small environment size. Next, extensive simulations, using several setting parameters, are driven to further evaluate our algorithm.

3.5.1 Experimental Set up

We setup our network topologies using a customized version of the topology generator tool : topology-generator (Ghali, 2015). More specifically, we generate multiple scenarios with the objective of evaluating the influence of several end-user, network, surrogate server and VNF type related parameters with regard to the VNF placement problem. Hence, while retaining the core network topology (i.e., edge connections between surrogates), we vary characteristics such as server capacity, edge bandwidth, VNF resource requirements etc. In summary, we consider a base scenario in which 9 surrogate servers are randomly interconnected each by 1 to 4 outgoing links towards other surrogate servers. Furthermore, we consider in our simulations, 5 content servers uniformly spread across the network, from which content needed to provide for service requests are stored. For added realism,

TABLE 3.2 Simulation Parameters (BOUET et al., 2015; CACHEDA et al., 2007)

Parameters	
Number of servers	9
Number of end-users	9, 12, 15, 18, 25
Number of VNFs per chain	3
Service Delay Threshold (ms) (D_h^{th} in ILP)	80-250
Bandwidth Cost (Dollar/Gbps) (B in ILP)	10
End-user Load (Mbps) (L_f in ILP)	15-50
Surrogate servers capacity (vCPU) C_n in ILP)	16-64
Site license cost(Dollar)(γ in ILP)	1000
Surrogate servers Operational cost (Dollar/vCPU) (δ_n in ILP)	5-10
VNF license cost (Dollar/vCPU) (α_k in ILP)	100
Capacity weight (π)	0.8
VNF instance weight (Γ)	2
Penalty function coefficient (μ)	0.5

we ensure that each content can be found in at least 3 of the 5 content servers as expected in CDN networks where content are duplicated. Furthermore, We configure each content server to randomly have 1 to 3 outgoing links towards surrogates servers in the network. Finally, we vary the number of end users. It is important to note that we consider directed edges between nodes and the bandwidth capacity of each edge is randomly taken between 100, 1000 and 10000 Mbps. Moreover, we consider for each service request, a delay threshold ranging from 80 to 250(ms) based on (CACHEDA et al., 2007). All simulation and network topology parameters are summarized respectively in Table 3.2 and 3.3 with VNF license cost and site license cost selected from (BOUET et al., 2015).

Table 3.3 Topology parameters

Parameters	**Values**
Number of surrogate servers	9
Number of end-users	9,12,15,18,25
Number of content servers	5
Bandwidth per edge (Mbps)	(100,1000,10000)
Max # of outgoing links surrogate to surrogate	1-4
Max # of outgoing links end user to surrogate	1-2
Max # of outgoing links content server to surrogate	1-3

3.5.2 Performance Metrics

To evaluate the effectiveness of the proposed algorithm, four performance metrics are taken into account.

1. *Operational cost* : It is defined as the total cost of using the surrogate servers on which VNFs are deployed as well as the VNF and site license cost. It is expressed in dollars.

2. *Communication cost* : It is defined as the total bandwidth cost incurred by serving the end-users service requests. It includes the cost of communication between the content server to the surrogate server hosting the first VNF, between the surrogate servers that host VNFs and the cost of communication between the surrogate servers that host the last partition (or last VNF) of the VNF chain and the end-user.

3. *Total cost (Dollar)* : It is defined as the sum of operational cost and communication cost. Reduced total cost indicates the cost-efficiency of a VNF placement algorithm.

4. *Average response time :* It is defined as the average duration in which a given content is retrieved from a selected content server, traverse the required surrogate servers hosting the VNFs specified in its VNF chain to its final delivery to the end-user.

3.5.3 Results and Discussions

Our heuristic CPVNF, heavily focuses on reducing the operational cost in order to achieve a reduction of the total cost.

As such, we initially set the capacity weight π at 0.8 such that surrogate servers with higher capacities are favoured. Furthermore, once a VNF instance is scheduled for a given surrogate server, we apply a VNF instance Γ_k to ensure continued utilization of the surrogate server.

Hence, as it can be observed in Fig. 3.2, CPVNF globally allows for a slightly fewer number of surrogate servers to be used to host VNFs. It is worth noting however, that one should not assume that CPVNF performs better than the ILP solution.

In fact, our strategy while globally beneficial in terms of operational cost as shown in Fig. 3.4 also has an adverse effect in terms of communications costs shown in Fig. 3.3 where the ILP performs much better. Such trade-off between the number of servers used and the communication cost allows for the ILP to obtain reduced total costs as illustrated in Fig. 3.5.

In short, our approach for CPVNF consist of finding the most appropriate surrogates, determine the shortest paths to both the selected content server and the end-user

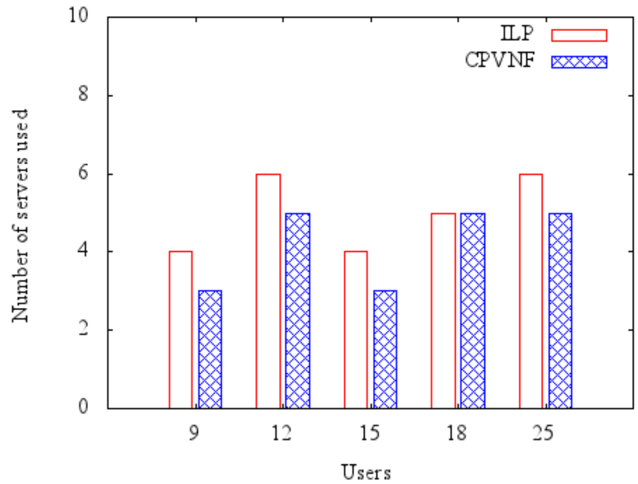

Figure 3.2 Number of servers used

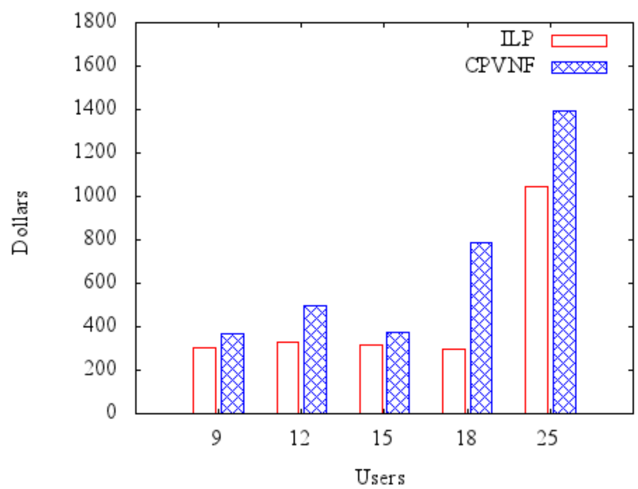

Figure 3.3 Communication cost

with the aim of respecting the required QoS threshold. We base this strategy and our trade-off choice upon the a priori fact that the site license cost is much higher than the bandwidth cost.

Therefore, to focus on reducing the number of servers used is computationally more efficient than exploring optimal network paths as we keep in mind that we are also

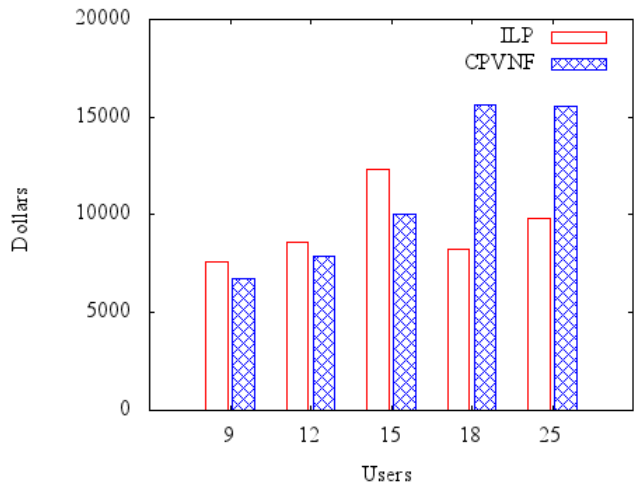

FIGURE 3.4 Operational cost

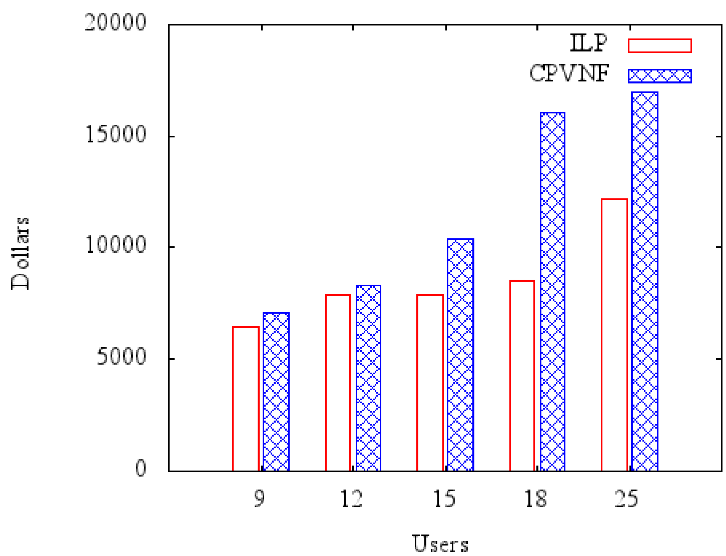

FIGURE 3.5 Total cost

designing for large scale networks. Note however that, such placement policy does not always lead to a successful mapping with respect to the QoS constraints due to the fact that both content servers and end-users may be located relatively far from the surrogate servers resulting in higher number of hops and consequently more bandwidth resource consumption.

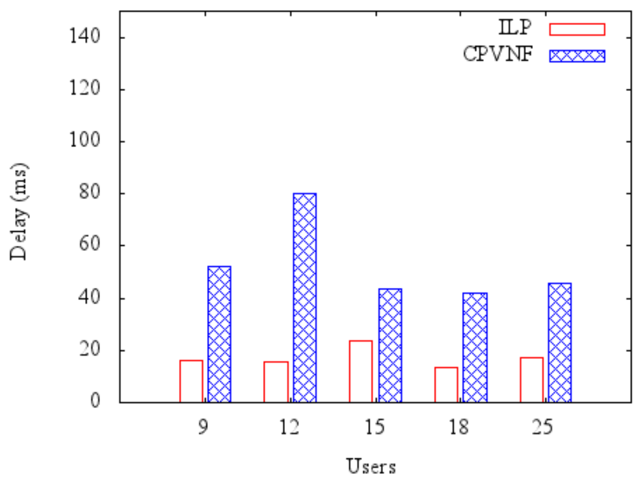

FIGURE 3.6 Average response time

In this case, we allow several iterations in which we temporarily focus on nodes with higher outgoing bandwidth capacities rather than the node capacity itself by progressively reducing the node capacity weight π in order to find more appropriate nodes. A consequence of this exploration is that we are using more content servers as shown in Fig. 3.7, given the fact that our penalty function always favours the content servers with the least delay from a surrogate server's perspective.

We do however argue that CPVNF could also achieve better results with an optimal value of π which better reflect the network topology. Precise determination of such value is left for future work. We further evaluated in Fig. 3.6, the average response time of requests in our simulations. While the ILP solution naturally achieves better response times for requests, those obtained by CPVNF are relatively low.

Another important characteristic of a VNF placement consists of its time complexity. It is shown that for a given precision parameter ϵ, page rank can be iteratively computed with a number of iterations proportional to max$\{1\text{-}\epsilon\}$ (BIANCHINI

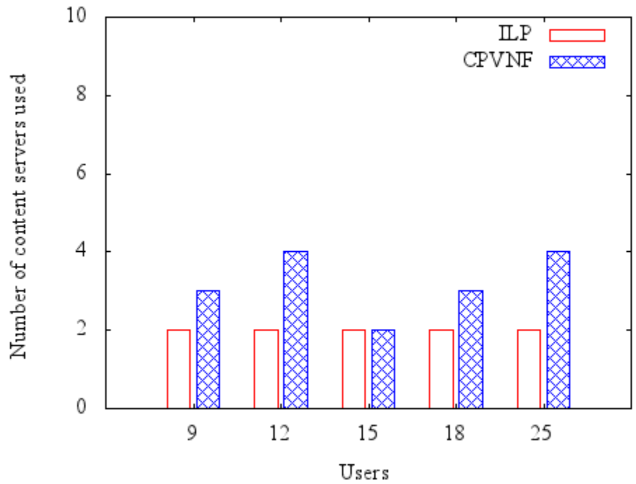

FIGURE 3.7 Number of content servers used

et al., 2005; CHENG et al., 2011). Furthermore, it is possible to solve the VNF to VNF link mapping problem in polynomial time according to (AHUJA et al., 1993) and (CHENG et al., 2011). Hence, CPVNF is a polynomial-time algorithm in terms of N, K and max{1-ϵ}.

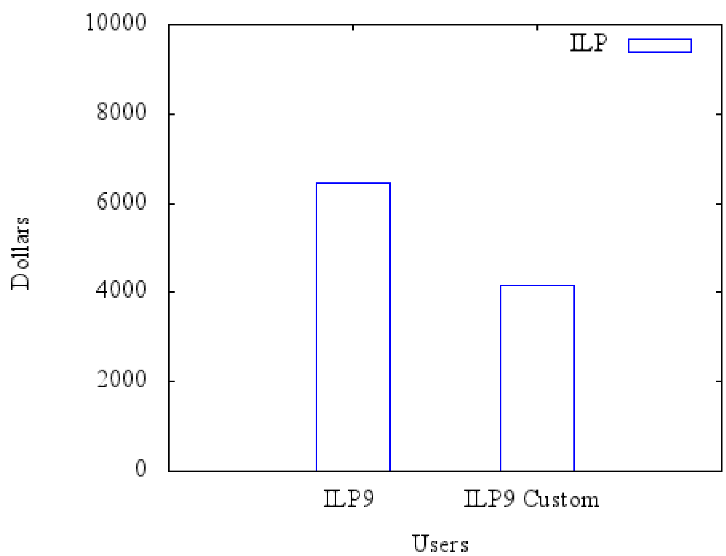

FIGURE 3.8 Comparison of total cost with reduced α_k and δ_n - 9 servers

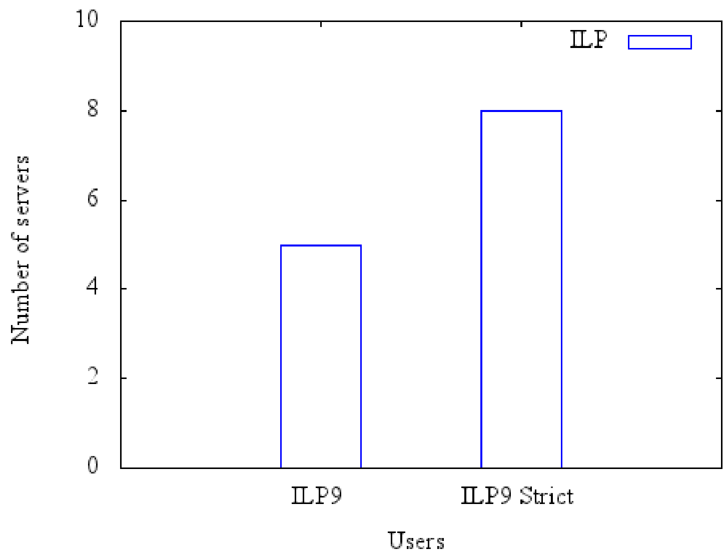

FIGURE 3.9 Number of server used with tighter specifications - 9 servers

We are also interested in identifying which parameters has the most influence in VNF placement. Interestingly, in one scenario (15 users) illustrated in Fig. 3.2, we observe a fewer number of server used compared to scenarios with less end-users.

This scenario is characterized by a large number of service requests composed of the same VNF chain combined to delay thresholds ranging from 127–250 (ms). We thus hypothesize that the characteristics of service requests (variety of VNF types across requests, delay threshold) have more influence than the number of end users.

To confirm this, we evaluate in Fig. 3.9, the number of servers used as we tighten the delay threshold for service requests within the range of 40–90 (ms) and increase the resource requirement R_k for VNF types to be randomly taken within the range of (4–8) while reducing surrogate servers capacity C_n to be either 8 or 16. We then note, an increase in the number of servers used, obviously leading to higher operational costs.

In contrast, in our simulations, sensibly reducing the values of α_k and δ_n while maintaining vnf resource requirements and surrogate server capacity did not yield significant changes in vnf placement. Obviously, as shown in Fig. 3.8, there is a reduction of the total cost mainly due to the multiplicative reduction of these parameters.

3.6 Conclusion and future work

This paper proposes a cost-efficient proactive VNF placement algorithm to guarantee the service delay in offering VNF based value-added services to end-users. The objective is to place VNFs in a way that it leads to the optimum number of VNFs to reduce the cost while still satisfying the service delay threshold.

Providing a maximum delay to end-users located anywhere in the network is one of the main benefits of the proposed algorithm. An ILP has been presented to model the VNF placement problem for value-added services in CDNs. In addition, a systematic view of the VNF lifecycle has been presented. As the first to propose VNF chaining placement in CDNs, (since the service chain format of our work is different from those of related works), we only compare CPVNF with ILP for small size environments.

As a future work, the proactive VNF placement algorithm will be integrated into an appropriate scaling algorithm to handle fluctuations in end-user workload over time. The scaling algorithm will be triggered in the case of violation in some performance indicators such as QoS and/or resource utilization at the run-time. Adding a prediction model to predict the future end-user workload in the environment and deploying VNF instances accordingly is also considered as one of the future extension of this paper.

Acknowledgment

This work was supported in part by Ericsson Canada, and the Natural Science and Engineering Council of Canada (NSERC). The authors would also like to thank Elaheh T. Jahromi for her contributions to the use case described in the paper.

REFERENCES

Ahuja, Ravindra K, Thomas L Magnanti et James B Orlin (1993). *Network flows : theory, algorithms, and applications.* Prentice hall.

Bari, M. F. et al. (2015). "On orchestrating virtual network functions". In : *11th International Conference on Network and Service Management (CNSM)*, p. 50-56.

Bertsekas, Dimitri P (1976). "On penalty and multiplier methods for constrained minimization". In : *SIAM Journal on Control and Optimization* 14.2, p. 216-235.

Bianchini, Monica, Marco Gori et Franco Scarselli (2005). "Inside pagerank". In : *ACM Transactions on Internet Technology (TOIT)* 5.1, p. 92-128.

Bouet, M., J. Leguay et V. Conan (2015). "Cost-based placement of vDPI functions in NFV infrastructures". In : *Proceedings of the 1st IEEE Conference on Network Softwarization (NetSoft)*, p. 1-9.

Brin, Sergey et Lawrence Page (1998). "The anatomy of a large-scale hypertextual Web search engine". In : *Computer Networks and ISDN Systems* 30.1, p. 107-117.

Cacheda, Rafael Asorey et al. (2007). "QoS requirements for multimedia services". In : *Resource Management in Satellite Networks.* Springer, p. 67-94.

CARELLA, G. et al. (2014). “Cloudified IP Multimedia Subsystem (IMS) for Network Function Virtualization (NFV)-based architectures”. In : *IEEE Symposium on Computers and Communications (ISCC)*. T. Workshops, p. 1-6.

CHENG, Xiang et al. (2011). “Virtual network embedding through topology-aware node ranking”. In : *ACM SIGCOMM Computer Communication Review* 41.2, p. 38-47.

CISCO (2015). *Cisco Visual Networking Index : Forecast and Methodology, 2014–2019*. Rapp. tech. Cisco.

ETSI GS NFV 001, Network Function Virtualization (NFV) Use cases, V1.1.1. (2013). Etsi.

FANG, W. et al. (2016). “Joint Spectrum and IT Resource Allocation for Efficient VNF Service Chaining in Inter-Datacenter Elastic Optical Networks”. In : *IEEE Communications Letters* 20.8, p. 1539-1542.

FARAHANI, Reza Zanjirani et al. (2014). “Survey : Hierarchical Facility Location Problem : Models, Classifications, Techniques, and Applications”. In : *Comput. Ind. Eng.* 68, p. 104-117.

FRANGOUDIS, P. A., L. YALA et A. KSENTINI (2017). “CDN-As-a-Service Provision Over a Telecom Operator’s Cloud”. In : *IEEE Transactions on Network and Service Management* 14.3, p. 702-716.

GHALI, Cesar (2015). *Topology generator*. https://github.com/cesarghali/topology-generator.

GHAZNAVI, M., A. KHAN et al. (2015). “Elastic virtual network function placement”. In : *IEEE 4th International Conference on Cloud Networking (CloudNet)*, p. 255-260.

GHAZNAVI, M., N. SHAHRIAR et al. (2016). “Service function chaining simplified”. In : *arXiv preprint arXiv :1601.00751*.

GIOTIS, K., Y. KRYFTIS et V. MAGLARIS (2015). “Policy-based orchestration of NFV services in Software-Defined Networks”. In : *IEEE Conference on Network Softwarization (NetSoft)*, p. 1-5.

HAN, B. et al. (2015). “Network function virtualization : Challenges and opportunities for innovations”. In : *IEEE Communications Magazine* 53.2, p. 90-97.

HERBAUT, N., D. NEGRU, D. DIETRICH et al. (2017). “Service chain modeling and embedding for NFV-based content delivery”. In : *2017 IEEE International Conference on Communications (ICC)*, p. 1-7.

HERBAUT, N., D. NEGRU, G. XILOURIS et al. (2015). “Migrating to a NFV-based Home Gateway : Introducing a Surrogate vNF approach”. In : *6th International Conference on the Network of the Future (NOF)*, p. 1-7.

IBN-KHEDHER, Hatem et al. (2017). “OPAC : An optimal placement algorithm for virtual CDN”. In : *Computer Networks* 120.Supplement C, p. 12-27.

JAHROMI, N. et al. (2017). “NFV and SDN-based Cost-efficient and Agile Value-added Video Services Provisioning in Content Delivery Networks”. In : *IEEE Consumer Communications & Networking Conference (CCNC)*. Las Vegas, NV, USA.

LAWRENCE, Page et al. (1998). *The PageRank Citation Ranking : Bringing Order to the Web*. Technical Report. Stanford University.

LIN, T. et al. (2016). "Demand-Aware Network Function Placement". In : *Journal of Lightwave Technology* 34.11, p. 2590-2600.

LUIZELLI, M. C. et al. (2015). "Piecing together the NFV provisioning puzzle : Efficient placement and chaining of virtual network functions". In : *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, p. 98-106.

MARTINS, Joao et al. (2014). "ClickOS and the Art of Network Function Virtualization". In : *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*. Nsdi'14. Seattle, WA, p. 459-473.

MECHTRI, M., C. GHRIBI et D. ZEGHLACHE (2016). "A Scalable Algorithm for the Placement of Service Function Chains". In : *IEEE Transactions on Network and Service Management* 13.3, p. 533-546.

MIJUMBI, R. et al. (2016). "Network Function Virtualization : State-of-the-Art and Research Challenges". In : *IEEE Communications Surveys Tutorials* 18.1, p. 236-262.

MOENS, H. et F. D. TURCK (2014). "VNF-P : A model for efficient placement of virtualized network functions". In : *10th International Conference on Network and Service Management (CNSM) and Workshop*, p. 418-423.

MONACI, Michele et Paolo TOTH (2006). "A Set-Covering-Based Heuristic Approach for Bin-Packing Problems". In : *INFORMS J. on Comput.* 18.1, p. 71-85.

Mouradian, C. et al. (2015). "NFV based gateways for virtualized wireless sensor networks : A case study". In : *IEEE International Conference on Communication Workshop (ICCW)*, p. 1883-1888.

Muñoz, R. et al. (2015). "SDN/NFV orchestration for dynamic deployment of virtual SDN controllers as VNF for multi-tenant optical networks". In : *Optical Fiber Communications Conference and Exhibition (OFC)*, p. 1-3.

Pallis, George et Athena Vakali (2006). "Insight and Perspectives for Content Delivery Networks". In : *Commun. ACM* 49.1, p. 101-106.

Pathan, M., R. K. Sitaraman et D. Robinson (2014). "Advanced Content Delivery, Streaming, and Cloud Services". In : *Content Delivery Networks*. Sous la dir. de Rajkumar Buyya, Mukaddim Pathan et Athena Vakali. Wiley Publishing.

Pathan, Mukaddim, Rajkumar Buyya et Athena Vakali (2008). "Content Delivery Networks : State of the Art, Insights, and Imperatives". In : *Content Delivery Networks*. Sous la dir. de Rajkumar Buyya, Mukaddim Pathan et Athena Vakali. Berlin, Heidelberg : Springer Berlin Heidelberg, p. 3-32.

Qu, L. et al. (2016). "Reliability-Aware Service Provisioning in NFV-enabled Enterprise Datacenter Networks". In : *IFIP/IEEE 12th International Conference on Network and Service Management*.

Riggio, R., A. Bradai et al. (2016). "Scheduling Wireless Virtual Networks Functions". In : *IEEE Transactions on Network and Service Management* 13.2, p. 240-252.

Riggio, R., T. Rasheed et R. Narayanan (2015). “Virtual network functions orchestration in enterprise WLANs”. In : *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, p. 1220-1225.

Son, S-W et al. (2012). “PageRank and rank-reversal dependence on the damping factor”. In : *Physical Review E* 86.6, p. 066104.

Sun, Q. et al. (2016). “Forecast-Assisted NFV Service Chain Deployment based on Affiliation-Aware vNF Placement”. In : *IEEE Globecom*.

Teixeira, J. C. et A. P. Antunes (2008). “A hierarchical location model for public facility planning”. In : *European Journal of Operational Research* 185.1, p. 92-104.

Trefis (2015). *Akamai Q2 Earnings Preview : Value-Added Services Will Drive Growth, But Currency Headwinds Likely Played The Minor Spoilsport*. url : http://www.trefis.com/.

Xia, M. et al. (2015). “Network Function Placement for NFV Chaining in Packet/Optical Datacenters”. In : *Journal of Lightwave Technology* 33.8, p. 1565-1570.

Zeng, M., W. Fang et Z. Zhu (2016). “Orchestrating Tree-Type VNF Forwarding Graphs in Inter-DC Elastic Optical Networks”. In : *Journal of Lightwave Technology* 34.14, p. 3330-3341.

CHAPITRE IV

ARTICLE 2 - TOWARDS RELIABLE REMOTE HEALTH MONITORING IN FOG COMPUTING NETWORKS

Mouhamad Dieye
Université du Québec à Montréal
Montréal (Québec), Canada

Amina Mseddi
Université du Québec à Montréal
Montréal (Québec), Canada

Wael Jaafar, Professeur
École de Technologie Supérieure
Montréal (Québec), Canada

Halima Elbiaze, Professeur titulaire
Université du Québec à Montréal
Montréal (Québec), Canada

AVANT-PROPOS À L'ARTICLE 2

Le deuxième article de la thèse explore la problématique de l'approvisionnement de services critiques, donc caractérisés par des exigences strictes en matière de qualité de service, dans un environnement réseau non fiable tels que les réseaux de *fog computing*. En particulier, nous étudions la mise en place d'une plateforme de suivi médical à distance, dans le contexte de la pandémie COVID-19.

L'article précédent nous a poussé à remettre en question l'hypothèse mono-domaine qui prévaut dans la littérature, qui impliquerait que tous les nœuds de traitement entre les sources de données et les utilisateurs finaux seraient sous l'autorité d'un seul opérateur CDN tel qu'Akamai. Compte tenu de la dispersion et de la mobilité des utilisateurs finaux, l'hypothèse mono-domaine est peu plausible pour plusieurs raisons (pratique, technique, financière, réglementaire, etc.). Le rejet de l'hypothèse mono-domaine, c'est-à-dire que certains éléments du réseau échappent au contrôle direct de l'opérateur, suggère que le respect des exigences QoS pour les utilisateurs finaux n'est pas assuré, en particulier pour les applications de la prochaine génération dont les besoins de latence sont inférieurs à 10 ms. En outre, nos simulations avec des réseaux à grande échelle révèlent les limites des approches heuristiques standard en ce qui concerne l'évolutivité, l'intelligence du réseau, la flexibilité et le temps de réponse aux décisions face à l'échelle sans précédent des réseaux de nouvelle génération. À la lumière de ces éléments, nous nous orientons vers les approches d'apprentissage automatique, en particulier l'apprentissage par renforcement profond.

Cet article a été publié dans la revue **IEEE Transactions on Network and Service Management (IEEE TNSM), Special issue on Recent Advances in the Design and Management of Reliable Communication Networks, Volume 19, Issue 3, July 2022**. IEEE TNSM est classé Q1 dans la catégorie

"Computer Networks and Communications" (SJR 2021) avec un facteur d'impact sur 5 ans de 6.048 (H-Index : 57).

- Dieye, M., Mseddi, A., Jaafar, W., & Elbiaze, H. (2022). **Towards Reliable Remote Health Monitoring in Fog Computing Networks.** *IEEE Transactions on Network and Service Management, 15(2), 774-786.* DOI: 10.1109/TNSM.2022.3194806

Les articles listés ci-dessous sont connexes à l'article présenté ci-haut.

- Bourhim, E. H., Elbiaze, H., & Dieye, M. (2019, October). **Inter-container communication aware container placement in fog computing.** In *2019 15th International Conference on Network and Service Management (CNSM) (pp. 1-6).* DOI: 10.23919/CNSM46954.2019.9012671
- Abbasi, U., Bourhim, E. H., Dieye, M., & Elbiaze, H. (2019). **A performance comparison of container networking alternatives.** *IEEE Network, 33(4), 178-185.* DOI: 10.1109/MNET.2019.1800141

Les articles IEEE TNSM, IEEE CNSM et IEEE Network sont examinés par des pairs conformément aux exigences énoncées dans le manuel d'opérations de l'IEEE PSPB (sections 8.2.1.C & 8.2.2.A). Chaque article publié a été révisé par un minimum de deux réviseurs indépendants utilisant un processus de révision par les pairs en simple aveugle, où l'identité des réviseurs n'est pas connue des auteurs, mais les réviseurs connaissent l'identité des auteurs. Les articles sont soumis à un contrôle de plagiat avant d'être acceptés.

ABSTRACT

As the World is still facing the COVID-19 pandemic, several researchers and industry players have proposed technological solutions to help fight the pandemic and pave the way for post-pandemic era precautions.

In this matter, the potential benefits of remote health monitoring have been brought back to the spotlight. Indeed, with current advances in wireless communications, core network virtualization, and computing architectures as enablers, consistently guaranteeing the stringent quality-of-service (QoS) requirements of remote health monitoring, e.g., ultra-low latency, may be achievable. Notably, the fog computing (FC) paradigm has been advocated as a potential solution for remote health monitoring.

However, the unreliability of fog nodes in FC networks is a critical aspect often overlooked despite its significant impact on vital latency requirements.

This paper proposes a reliable fog-based remote health monitoring framework operating under uncertain fog computing conditions. Specifically, we formulate the problem of assigning tasks of remote sensors attached to patients to their adequate applications deployed in fog nodes aiming to maximize the number of satisfied tasks with respect to the fog nodes' availability and communication latency constraints.

Due to the problem's NP-hardness, we leverage a differential evolution-based algorithm enhanced by reinforcement learning to deploy applications in fog nodes. Numerical results demonstrate the superior reliability performance of our proposed solution, in terms of the average success ratio of tasks, compared to benchmarks.

Specifically, our simulations show up to 60 % performance improvement compared to benchmarks in specific scenarios. Moreover, by investigating the impact of several key parameters, we identify a design trade-off between the number of fog nodes and the latter's intrinsic failure rates.

Keywords: Medical services, Remote health monitoring, Reliability, COVID-19,

Cloud computing, Quality of service, fog computing, containers, sensors

4.1 Introduction

The World is currently facing new waves of the pandemic outbreak of COVID-19, a.k.a., SARS-CoV-2. The number of confirmed infection cases has reached over 250 million, with over 5 million reported deaths worldwide. Due to its contagious nature, preventive measures have been advocated to reduce community transmission (A. Remuzzi et G. Remuzzi, 2020; Cascella et al., 2020).

Unfortunately, the rapid growth rate in some countries has stretched healthcare systems to maximum capacity, forcing emergency room restrictions and shortages in testing kits and essential medical equipment. This chaotic situation has partially led to an increased COVID-19 death rate.

Consequently, population confinement, social isolation, and lockdown measures have been chosen as the appropriate worldwide response policy. In fact, it is acknowledged that stringent isolation measures have significantly reduced the frequency of contaminated cases in certain countries. Such practices, known in epidemiology as "flattening the curve" serve to slow the infection rate, thus allowing enough medical resources to be available and ready to treat the sick.

In practice, however, the application of these measures is found to be challenging. Indeed, initial symptoms associated with COVID-19 are ambiguous and similar to those of the common cold or influenza. Furthermore, recent findings indicate a possible disease transmission from pre-symptomatic and asymptomatic individuals (Rothan et Byrareddy, 2020; Mizumoto et al., 2020).

Fear of contamination in clinical environments has drastically curtailed on-site elective care to prevent risking the health of medical staff and patients, despite strict precautions, especially in light of the recent development of variants like the Delta variant. Since the clinical spectrum of SARS-CoV-2 pneumonia requires early detection and monitoring within a clinical environment, critical cases are supported in hospitals while mild cases are relegated to remote care (Behar et al., 2020).

Moreover, the expense and complexity of managing mass face-to-face, unscheduled, or quickly arranged visits pose a challenge for most healthcare systems (Watson et al., 2020). Hence, monitoring non-severe COVID-19 patients continuously, either from their quarantine site at home or dedicated venues such as hotels, becomes an attractive alternative (Behar et al., 2020).

To complement the aforementioned public health efforts and reduce risks for the medical staff, we advocate for a real-time fog-based healthcare monitoring system to collect, transfer, and analyze physiological data from sensors attached to confined patients with the aim of 1) identifying and emitting alarms for urgent cases, particularly in elderly housings, 2) detecting local community transmissions, 3) adequately managing the planning and deployment of scarce supplies such as testing kits, respiratory equipment, and beds to decrease the workload of medical practitioners and reduce healthcare costs ; and 4) tracking the non-observation of confinement measures. In addition to real-time patient monitoring, a remote health monitoring deployment would reduce the risk of disease exposure and transmission by preventing unnecessary hospital visits.

Current cloud-based healthcare solutions provide numerous advantages, including the ability to transfer patient data in bulk for processing by reliable, cost-effective, and on-demand computing and storage resources. Data examples include measurements and analyses of health-related metrics such as heart rate, cough recordings, blood pressure, blood oxygen saturation, and temperature, which health authorities use to make appropriate public health decisions. Despite this, cloud-based processes are often not performed in real-time and may have delays due to numerous network variables. With the increased potential of future outbreaks and pandemics, health officials must make excellent medical judgments swiftly, primarily since delays may raise infection rates and community dissemination.

Given the stringent quality-of-service (QoS) requirements for patient data, particularly in terms of response time and availability, fog computing (FC) networks have gained popularity in recent literature for time-sensitive tasks, such as those for healthcare applications (Kraemer et al., 2017; Bonomi et al., 2012).

Indeed, FC has emerged as a potential paradigm for addressing the issues posed by the exponential expansion of the Internet of Things (IoT) devices and network edge sensors. It expands the concept of cloud computing by bringing resources and services closer to IoT/end-user devices, such as access points, routers, mobile base stations, and cars (He et al., 2018). Fog nodes are typically characterized by restricted computing, communication, storage, and control capabilities, in addition to additional physical constraints like battery life, on-and-off switching, and wireless interference (A. Mseddi et al., 2019; Amina Mseddi et al., 2019). Introducing an extra computational layer via fog nodes enables time-sensitive, mission-critical analytic jobs to be conducted close to end-users, decreasing response times.

In the current pandemic situation, several benefits of FC can be critically exploited. Firstly, fog devices can sort tasks based on multiple evaluation criteria and reroute less critical tasks and those requiring large-scale analytical work or long-term storage to the cloud. Additionally, the location awareness of FC networks facilitates the detection of local community transmissions by considering the geographical locations of fog nodes while placing services and tasks.

Moreover, by restricting the propagation of data, data privacy and security can be enhanced by, for instance, processing sensitive data on local devices rather than in the cloud, which is beyond the control of the user (Vaquero et Rodero-

Merino, 2014 ; Kraemer et al., 2017). In addition, strategies such as differential privacy can be implemented at the fog level to improve confidentiality (Piao et al., 2019).

FC networks must handle various obstacles despite the advantages mentioned above. Among these are the heterogeneity and reliability of fog devices in comparison to their cloud-based counterparts (Madsen et al., 2013). The literature mostly ignores the reliability and availability of fog (including IoT) devices, as they are primarily viewed as black boxes when incorporated into critical and large-scale systems (Mavrogiorgou et al., 2018).

Unlike cloud computing systems, where fault tolerance solutions are more readily available, maintaining reliability and availability in fog computing remains an unanswered question (Hu et al., 2017 ; Yi et al., 2015 ; Madsen et al., 2013). Note that this has significant ramifications for the satisfaction of QoS requirements, particularly for remote health monitoring, where information delay or loss can have catastrophic repercussions (Kraemer et al., 2017 ; Dhanvijay et Patil, 2019).

A typical fault tolerance strategy used to assure service continuation in the event of failure is replication, which can also be utilized to lower the overall response time by distributing the workload across compute nodes. Due to the heterogeneity of fog nodes in terms of failure rate, i.e., the frequency with which they become unavailable, and their limited storage capacity, providing real-time critical healthcare services with stringent QoS standards is a challenging and pertinent endeavor.

Consequently, we aim to address the challenge of providing healthcare services in unreliable fog computing environments. To the best of our knowledge, this work is the first to consider heterogeneous node reliability and availability for intelligent healthcare monitoring service placement in FC networks[1]. The main contributions of this paper are summarized as follows :

- We provide an FC framework capable of identifying and estimating the availability of known and unknown models of fog/IoT devices.
- we formulate the problem of deploying applications in fog nodes and assigning tasks of remote sensors to them, aiming to maximize the number of satisfied tasks, given the fog nodes' availability and communication latency constraints.
- Due to the NP-hardness of the problem, we propose to solve it through a novel differential evolution algorithm on which we apply a deep reinforcement learning (DRL)-based adaptive operator selection (AOS) mechanism
- Numerical results prove the superior reliability of our approach compared to baseline solutions. Moreover, by investigating key parameters, we identify a design trade-off between the number of fog nodes and their intrinsic failure rates.

The remaining of the paper is organized as follows. Section II reviews the related work. Section III describes a use case of our remote health monitoring model. In Section IV, we mathematically formulate the joint application deployment and task assignment problem. Then, we present our proposed algorithmic solution in Section V, prior to exposing the simulation results in Section VI. Finally, Section VII concludes the paper.

---

[1]Although this article is written in the context of COVID-19, its scope can be extended to other health crisis events.

4.2 Related Work

Remote health monitoring refers to individuals' diagnosis and monitoring outside the clinical environment, typically in their home and through mobile clinics, to monitor a patient's well-being, diagnosis of medical conditions, and tracking changes over time, which may require medical attention (Behar et al., 2020).

4.2.1 Cloud-based remote health monitoring

In the literature, numerous publications have examined the benefits of cloud-based remote health monitoring solutions. For instance, the authors of (Pham et al., 2018) introduced CoSHE, a cloud-based smart home environment capable of collecting physiological, motion, and audio signals from residents using non-invasive wearable sensors to offer information on residents' daily activities and location within the home. The collected data are processed by a smart home gateway and then transmitted to a private cloud, giving remote caregivers real-time access to the data. Similarly, the authors of (Hossain, 2016) advocated for a remote healthcare solution to meet the continuing care demands of the elderly. The suggested system continuously captures patients' videos and speech through sensors from video cameras and microphones implanted in their homes, then transmits them to a specialized cloud for processing and scoring. Physicians administer medications and provide services via voice or video messaging, depending on the classification score. Finally, (Dang et al., 2019) provided an overview of recent works in cloud-based remote healthcare.

4.2.2 Fog Computing-based Remote Health Monitoring

The usage of fog computing for the Internet of Things in healthcare (H-IoT) is not new. The emergence of H-IoT, particularly remote patient monitoring, is the consequence of a strong push for an information-centric healthcare vision, which intends to lower costs while providing better accessibility, efficiency, continuity, and quality of patient care (Kraemer et al., 2017 ; Mandellos et al., 2009). Its importance within any nation's public health strategy has been accentuated with the advent of COVID-19.

Extensive surveys of recent work on H-IoT in fog computing networks are presented in (Dhanvijay et Patil, 2019 ; Kraemer et al., 2017). These works highlighted the immediate positive impact of an effective remote monitoring system, including reducing the workload on healthcare staff (who can provide better care) by automating manual measurements and supervision tasks, besides keeping non-critical patients out of the hospital for as long as possible.

Moreover, Bertini *et al.* compared in (Bertini et al., 2016) remote monitoring to in-hospital follow-ups, showing the superior benefits of remote monitoring, which even includes a positive impact on the survival rate. Similarly, Topol underlined in (Topol, 2012) that using patient-attached sensors enables health authorities to collect a more precise stream of biometric data to aid, via analytical approaches, early detection, diagnosis, medication, and the discovery of new therapeutic avenues.

Most published works examined patient data to activate alarms when urgent circumstances were detected. For instance, the authors of (Cao et al., 2015) developed a fall detection system based on the local analysis of acceleration magnitude measurements. To detect false positives, filtering and non-linear time series analysis techniques were also employed. This system uses fog computing to spread analytical workloads over the network by dividing the detection task across fog nodes and cloud servers.

Within the same context, Craciunescu *et al.* suggested in (Craciunescu et al., 2015) a remote monitoring technique that includes fall detection via real-time analysis of environmental and patient-attached sensor data. Also, Aazam *et al.* detailed in (Aazam et Huh, 2015) the deployment of an emergency warning service called E-HAMC for offloading and preprocessing in an FC network. In an emergency, this service transmits the incident's location and automatically contacts the appropriate emergency department.

The authors of (Gia et al., 2015) analyzed a case study of Electrocardiogram (ECG) feature extraction to aid in diagnosing heart disorders using a lightweight wavelet transform mechanism. In a broader framework, the authors of (López et al., 2010 ; Huang et al., 2009) advocated monitoring the patient's vital signs by remotely recording his position and activity data.

Finally, precise remote control of actuators attached to a patient has been investigated. For instance, Masip-Bruin *et al.* proposed in (Masip-Bruin et al., 2016) a scenario in which they collect and assess the patient's blood oxygen level by considering ambient data. Based on this data, they altered the patient's oxygen dose appropriately in real-time.

4.2.3 COVID-19 Centric Remote Health Monitoring

Within the COVID-19 context, an increasing number of works have become available in the literature, primarily focusing on architectural considerations and potential benefits in the fight against the pandemic (Swayamsiddha et Mohanty, 2020 ; Annis et al., 2020 ; Mirjalali et al., 2021 ; Gordon et al., 2020 ; Watson et al., 2020 ; Behar et al., 2020). For instance, (Watson et al.,

2020) expands on the value and requirements of a remote monitoring solution to immediately fight COVID-19 given its unprecedented impact from a societal and financial standpoint.

In their survey, Behar *et al.* presented a review of remote healthcare initiatives applied during the pandemic in 20 countries (Behar et al., 2020). The initiatives were categorized into four groups based on their respective objectives : (1) facilitate contact tracing and reconstruction of exposure to potentially infected COVID-19 patients through proximity and location tracing ; (2) perform remote monitoring by tools such as video chat and connected medical sensors to diagnose and manage patients ; (3) in addition to mobile clinics ; and (4) enable point-of-contact screening to rapidly identify potential patients in crowded locations.

In the context of H-IoT, we argue that reliability and availability are crucial QoS requirements in the FC framework, as is the latency for service placement in fog nodes. Indeed, defects in these QoS requirements may cause service failures ranging from minor inconvenience to life-threatening (Kraemer et al., 2017). Nevertheless, all the works mentioned above failed to consider this aspect, although FC nodes are subject to higher failure rates and lower redundancy than cloud nodes (Madsen et al., 2013 ; Mavrogiorgou et al., 2018).

Indeed, (Madsen et al., 2013 ; Mavrogiorgou et al., 2018) are among the rare studies to consider such aspects in fog computing. Madsen *et al.* described in (Madsen et al., 2013) the most critical characteristics regarding the reliability of FC networks, such as the IoT device and service failures, whereas in (Mavrogiorgou et al., 2018), based on the available or estimated reliability metrics, the authors proposed a framework to identify and assess the reliability of IoT devices.

4.2.4 Reliable Service Placement in FC Networks

Reliability represents a critical feature for service provisioning in FC networks, mainly as low reliability is generally synonymous with frequent service failures and unfulfilled QoS requirements (Yao et Ansari, 2018; Yao et Ansari, 2019). Typical methods leveraged to improve reliability include periodical checkpointing, rescheduling failed tasks, or replicating to exploit the power of parallelization (Madsen et al., 2013).

However, checkpointing and rescheduling in dynamic environments such as FC networks may be inappropriate due to system overheads, added latency, and low flexibility (Madsen et al., 2013; Yao et Ansari, 2018; Yao et Ansari, 2019). Despite its effectiveness for fault tolerance purposes, replication also induces high costs as replicas must be stored in different locations and require a coordination mechanism to ensure replica synchronization.

In this context, recent works have focused on improving the reliability of service placement for FC networks. For instance, the authors in (Yao et Ansari, 2018) focused on the resource provisioning of deadline-driven IoT applications. They present a provisioning algorithm, formulated as a Mixed-Integer Non-Linear(MINL) problem, for reliable virtual machine allocation in IoT networks. By offloading computationally intensive, deadline-driven tasks to fog nodes, their goal is to minimize costs while ensuring reliability requirements are met. As an extension to (Yao et Ansari, 2018), the authors in (Yao et Ansari, 2019) evaluated the trade-off between maximizing the reliability and minimizing the overall system cost. They described a multi-objective fog resource provisioning problem with a failure, recovery, and reliability model through a highly computationally complex ILP model. As such, they proposed the MBFD algorithm to obtain near-optimal solutions with lower computational complexity. In a broader context, extensive surveys of recent work on dependability in fog computing networks are presented in (Madsen et al., 2013; Bakhshi et al., 2019).

4.3 Fog-based remote health monitoring model

4.3.1 Use Case Description

We hereafter present our reliable fog-based remote health monitoring (F-RHM) system, shown in Fig.4.1. It is designed to prioritize high-risk and vulnerable populations, including those under preventive quarantine, over the age of 60, immuno-compromised, of frail constitution, or suffering from a chronic disease.

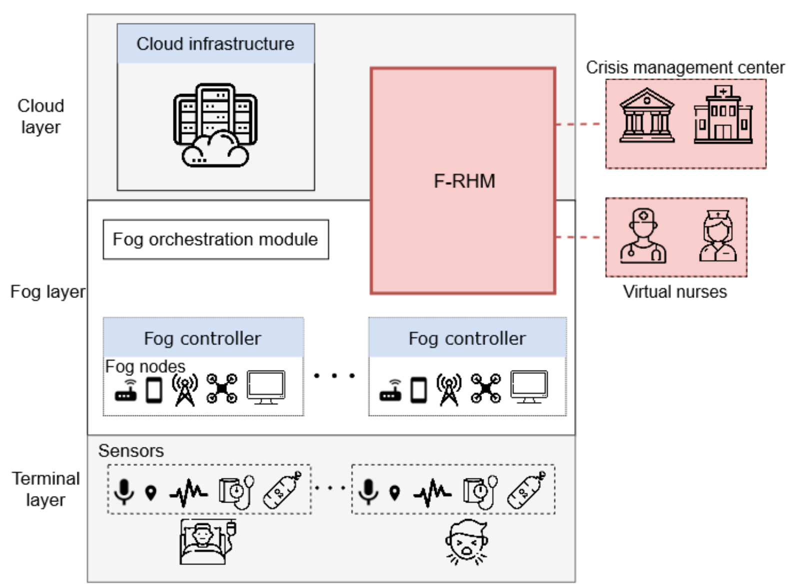

FIGURE 4.1 F-RHM system model.

Our system model identifies the Crisis management center (CMC) as any health authority assuming the following responsibilities :

- designing, implementing, and coordinating the emergency response,

- monitoring, assessing, and allocating medical resources,
- collecting, analyzing, and disseminating information,
- accommodating, informing, and taking direction from other national crisis management structures,
- preventing and preparing for future emergencies.

Our model also employs a 24/7 hotline headed by registered nurses to assess the symptoms of monitored patients and, if deemed necessary, escalate to virtual or on-site care. We define the following requirements for F-RHM inspired by (Behar et al., 2020 ; Watson et al., 2020) :

- able to survive uncertain network conditions ;
- asynchronous to enable scale ;
- allows for interactive engagement with patients through educational videos and messages, and continuous encouragement to respect health guidelines. Furthermore, enable patients to ask questions to a care team regarding their symptoms and pain control, receive updated information and reassurances ;
- a patient can always request a callback if needed, whether alarms are triggered or not ;
- ability to either extract data from connected medical sensors or self-enter data. If the latter method is used, patient-reported outcomes must come from designed survey questions ;
- allows for a secure live video visit, phone call, and chat messaging to enable virtual care ;

- ability to coordinate with CMC when on-site care is required for a patient ;
- ability to report to CMC in real-time in case of significant events such as new disease information, updated symptom questions, detected patterns of local transmission ;
- facilitate contact tracing and reconstruction of exposure to potentially infected COVID-19 patients using proximity and location tracing ;

We illustrate the flow diagram of F-RHM in Fig. 4.2. It begins with an onboarding phase in which potential patients are screened for COVID-19 symptoms through a telephone, chat messaging, or video-conference consultation. Patients with suspected COVID-19 symptoms are enrolled in the remote monitoring program. They are provided with instructional materials, reminders of health guidelines, and information on pain management, diet, and self-care, among other topics. Additionally, a kit containing various measuring sensors (e.g., temperature, blood pressure, respiratory rate, pulse oximeter, etc.) and IoT devices such as a cough recorder, are delivered to the patient's home, and support is provided until they can proficiently operate the sensors. The precise qualitative measures supplied by the sensors enable early disease detection and effective disease progression tracking. For instance, according to the World Health Organization (WHO), the beginning of severe pneumonia (a prominent symptom of COVID-19) occurs when the blood oxygen level falls below 93 % (Organization, 2020). Patients remain in the remote monitoring program until their symptoms lessen after the quarantine period, are escalated to on-site care, or choose to leave the program.

F-RHM intervenes when early signs of symptom deterioration and critical situations are detected by triggering alarms, and reviewed by the registered nurse team to filter false positives. Afterward, the patient is monitored closely. If symptoms worsen, the system alerts the virtual nurses to take additional actions such as

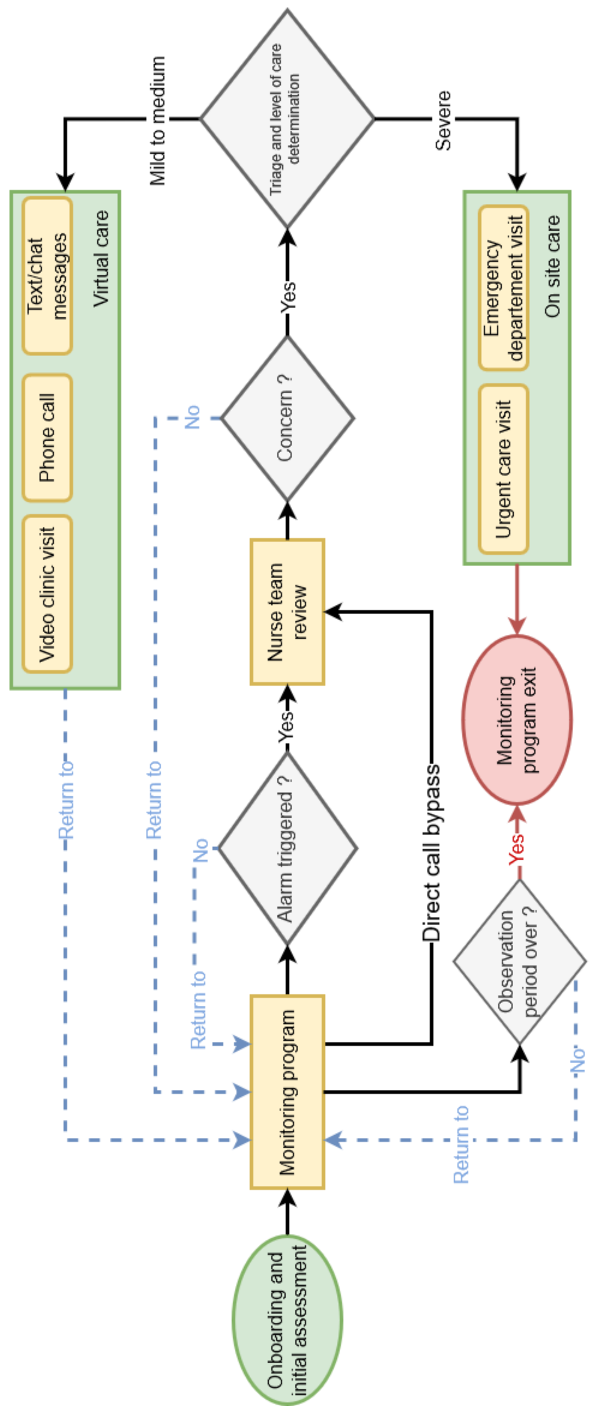

FIGURE 4.2 F-RHM flow diagram.

contacting patients through phone or video conference to assess and provide, if possible, virtual remote care. In the worst-case scenario, hospitalization is decided. Patients with severe symptoms are scheduled and directed to the appropriate level of local care for testing and hospitalized if confirmed for COVID-19. The CMC is notified during this triage process, thus allowing it to plan for medical resources and notify the hospital staff of the COVID-19 patients' arrival. Finally, F-RHM leverages a "cloud management module" for advanced computation and analysis in the cloud, which enables the CMC to gain new insights on the disease (e.g., the discovery of new symptoms) and be alerted to any event that requires its attention, such as admission, discharge, detection of local community transmission, tracing if a monitoree visits crowded areas, etc.

4.3.2 F-RHM Framework and Modules

We design the F-RHM framework to collaboratively operate within the three-layered hierarchical fog computing architecture composed of the terminal, fog layer, and cloud layer, as illustrated in Fig. 4.1.

Typically, the terminal layer comprises various IoT devices used for data sensing and is characterized by a widespread geographical distribution. In our scenario, these devices can be implantable or wearable medical sensors integrated into a wireless module that sends encrypted data, such as temperature, location, oxygen saturation, and ECG. The collected data is transmitted to the upper layers (notably the fog layer) for processing and storage purposes. The fog layer comprises fog nodes, which consist of various static and mobile smart devices (i.e., routers, switches, gateways, access points, etc.) that can process, route, and store data received from the terminal layer. For management purposes, fog nodes are typically clustered into domains placed under the authority of a fog controller. The latter is responsible for

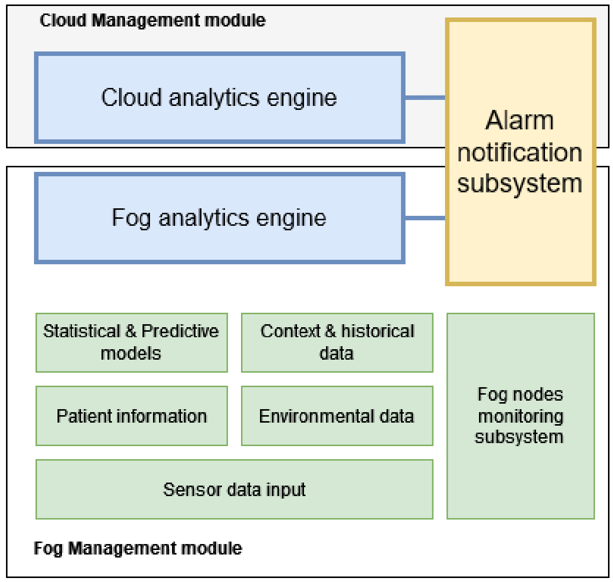

FIGURE 4.3 Components of F-RHM modules.

mapping end-user requests to the most appropriate resources at the network edge, using information from lower-layer resources. However, fog nodes are inherently resource-constrained in terms of processing power, memory, availability, etc., so the fog layer may not be suitable for highly demanding tasks.

Consequently, these tasks are offloaded to the cloud layer, consisting of data centers with high-performance computing and storage resources. The latter can handle the long-term/resource-heavy tasks at the expense of higher communication delays. Finally, the "fog orchestration module" coordinates service management between the cloud resources and fog controllers.

In Fig. 4.3, we provide a detailed overview of the F-RHM modules. F-RHM is designed to integrate patients, health professionals, and CMCs into a framework where data collected from patient-attached sensors can be thoroughly analyzed to obtain more accurate diagnoses and alerts. Note that it would be conceivable for these sensors to be equipped with actuators that receive commands and perform actions from health professionals (Mohsin et al., 2018). As such, strict authentication and privacy measures must be followed.

The F-RHM "fog management module" operates at the fog layer, assuming its deployment on trusted fog nodes with adequate security measures. Its primary role is to combine patient-attached sensor data with contextual knowledge, such as patient information, environmental data, and medical databases, as inputs to the "fog analytics engine".

Note that although some contextual data, e.g., environmental data, could be cached near end-users to reduce latency, the location of data, such as patient information, may be restricted as they are subject to privacy and security regulations. The function of the fog analytics engine is to determine the analytical tasks that need to be performed, e.g., ECG extraction.

Once the analytical tasks are determined, they are communicated to the fog orchestration module, which communicates with the fog controllers to find the appropriate placement of resources to execute the tasks. The results are then returned to the fog analytics engine to determine whether an alarm should be triggered or not. To avoid alarm fatigue, the fog analytics engine is configured to only trigger alarms based on sustained variations over time and across multiple patient-attached sensors.

In our context, alarms triggered at this level are considered time-sensitive and thus are redirected to local health authorities, including the registered nurses, who determine the subsequent actions to be taken.

The analytical task results are also transferred to the F-RHM cloud management module for data analysis based on aggregated data (with high resource demand) and long-term storage. The "cloud analytics engine" leverages the cloud infrastructure to extract global insights and trigger escalation alarms to the CMC if needed, for instance, when detecting COVID-19 hotspots or outbreaks. Moreover, the CMC can use the cloud analytics engine to improve operational efficiency by tracking, for instance, the efficacy of interventions following triggered alarms. This data can be reviewed to reduce false positives or negatives and serve as feedback to the fog analytics engine to decrease alarm fatigue or plan group testing at specific locations.

As previously mentioned, ensuring the reliability and availability of fog devices is critical to satisfying QoS requirements. Given the strong correlation between reliability and availability, we hereafter focus on the availability of fog devices. Hence, a "fog node monitoring subsystem" is proposed for F-RHM to assess the availability of fog nodes based on estimated/computed metrics, which are inspired from (Mavrogiorgou et al., 2018). The result is fed back to the fog orchestration module for adequate task placement.

Our fog node monitoring subsystem is built through a three-step process. First, we identify the baseline availability-related specifications for every new fog node by relying on a database populated with data published by the device model manufacturer. Table 4.1 provides an excerpt of the database.

Although the credibility of the manufacturer's data is generally in doubt, due in part to the lack of standard methods for the computation of failure-related metrics (O'Neill et Tolley, 2020; Krasich, 2009), such data is leveraged in the second step of our process to obtain baselines that we adjust over time to their true values. If an entry in the database corresponds to a new fog node model, we monitor the latter's failure data for a predetermined observation period. Then, we match it to a device model with the most similar failure behavior.

The final step consists of the feedback exchange between the fog node monitoring

TABLE 4.1 Fog node monitoring subsystem database excerpt
Source : Manufacturer datasheet, unless otherwise specified.
* Blancco Q4 2017 State of Mobile Device Repair & Security report.

#	Name	Device Type	Annualized Failure Rate AFR (%)
1	Cisco C812G-CIFI	Router	1.11
2	Meraki MX67W	Security Appliance	1.08
3*	Xiaomi Redmi	Mobile	9
4*	iPhone 6	Mobile	20
5	Cisco WS-C3850-12S	Switch	1.07
6	Cisco C1000-48T-4X-L	Switch	1.01
7	Aruba AP-504	Access point	1.01
8	Allied Telesis TQ4400e	Access point	1.08

subsystem and the fog orchestration module to improve our model's accuracy. Specifically, the orchestration module notifies the monitoring subsystem of a fog node's failure, which is used to reassess its availability.

4.4 Problem Formulation

In the following section, we formulate our real-time remote health monitoring platform in FC networks.

We assume that time is discretized according to the time system $\mathcal{T}$. Also, let $\mathcal{K}$ be the set of fog domains under the control of fog controllers, and $\mathcal{F}_k$ represents the set of heterogeneous fog nodes within the fog domain operated by fog controller k. Given the limited capacity of fog nodes, lightweight virtualization techniques such as containerization are typically used as deployment medium to improve isolation, portability, and security of applications (YANNUZZI et al., 2014). As such, we assume that each fog node $f \in \mathcal{F}_k$ embeds a containerization platform to manage its assigned containers. For simplicity, we focus our study on a system with a single fog domain. Moreover, we respectively omit the subscript $k \in \mathcal{K}$ and $t \in \mathcal{T}$ to improve readability.

We consider a scenario where patient-attached remote sensors send tasks to the FC platform for processing. At the fog layer, the fog controller deploys a resource allocation strategy to maximize the number of satisfied patient requests while taking into account the locations of fog nodes and patients, in addition to the fog nodes' failure rate and amount of available resources. Once a resource allocation strategy is determined, container deployment requests are sent to the designated fog nodes with patient tasks redirected to them. If a patient task's requirement is unmet, the fog controller must take corrective measures, such as replica creation or container migration, to ensure continued service. For resource efficiency reasons, a given container can execute compatible tasks from multiple patients.

Let $\mathcal{F}$ be the fog nodes set. We respectively define the vectors $\mathbf{l}_{|\mathcal{F}|}$, $\mathbf{c}_{|\mathcal{F}|}$ and $\mathbf{h}_{|\mathcal{F}|}$ of size $|\mathcal{F}|$, where $|\cdot|$ is the cardinality operator. The vector $\mathbf{l} = [l^f]$ expresses the fog nodes' locations. We use $\mathbf{c} = [c^f]$ to denote the remaining load capacity of the fog nodes. Note that the fog node's resources are not entirely available to the FC platform, as they are allocated in priority to the fog node's dedicated functions, e.g., a router must route network packets in priority. As the load incurred by dedicated functions varies over time, the amount of resources available to the FC platform fluctuates accordingly (A. MSEDDI et al., 2019). Hence, we express through the vector $\mathbf{z} = [z^f]$ the fog nodes' utilization, proportional to the number of resources dedicated to the primary functions. Finally, the vector $\mathbf{h} = [h^f]$ represents the fog nodes' failure rate, which can be estimated or extracted from the manufacturer's data.

We define by $\mathcal{U}$ the set of patients in the system with $\mathbf{p} = [p^u]_{|\mathcal{U}|}$ the vector denoting the locations of patients. We denote by $\mathcal{Y}$ the set of application types, i.e., ECG, body temperature, oxygen saturation, etc., which are available in the system. Let $\mathbf{Q} = [Q^{u,y}]_{|\mathcal{U}|\times|\mathcal{Y}|}$ be the matrix that defines applications' task load, $\mathbf{T} = [T^{u,y}]_{|\mathcal{U}|\times|\mathcal{Y}|}$ the matrix that defines applications' threshold latency requirements, and $\mathbf{A} = [A^{u,y}]_{|\mathcal{U}|\times|\mathcal{Y}|}$ the matrix that defines applications' minimum availability requirements. As previously noted, we mainly use replication to ensure that availability requirements are met. It can also be used to split the task load of applications. For any application, we define a maximum number R of replicas given that a large R would induce a significant replica synchronization delay. As the replicas may be hosted by multiple fog nodes, let the binary matrix $\mathbf{W} = [W^{r,u,y,f}]_{R\times|\mathcal{U}|\times|\mathcal{Y}|\times|\mathcal{F}|}$ indicate whether an application replica r of application task load u of type y is mapped to a fog node f or not.

Since replicas can be used for load sharing, we define by $\tilde{\mathbf{Q}} = [\tilde{Q}^{r,u,y,f}]_{R\times|\mathcal{U}|\times|\mathcal{Y}|\times|\mathcal{F}|}$ the matrix that represents the distribution of applications' task loads in fog nodes, where $\tilde{Q}^{r,u,y,f}$ denotes a partial load of $Q^{u,y}$ redirected to the container replica r that is placed in fog node f. We summarize the defined variables in Table 4.2.

4.4.1 QoS Requirements Expressions

In what follows, we determine the application availability and delay expressions to be used in formulating the optimization problem.

4.4.1.1 Availability

For reasons of simplicity, assume the application's availability to be identical to that of the hosting fog node. This assumption is reasonable since hardware failures are typically the root cause of service disruptions at the network edge, in addition to being the slowest to mitigate (Lloyd's, 2018; Gill et al., 2011; Dieye et al., 2017). Further assume a constant repair rate μ, which is reasonable under a scenario where nodes that are not repaired within a time frame are immediately replaced. The availability λ^f of a given fog node f is therefore expressed as

$$\lambda^f = \frac{\mu}{\mu + h^f}, \quad \forall f \in \mathcal{F}. \tag{1}$$

For an application to be considered available, at least one of its replicas must be accessible. Hence, the availability $A^{u,y}$ of application y associated with patient u can be written as

$$A^{u,y} = 1 - \prod_{f \in \mathcal{F}} (1 - \sum_{r=1}^{R} \lambda^f \cdot W^{r,u,y,f}), \forall u \in \mathcal{U}, y \in \mathcal{Y}. \tag{2}$$

4.4.1.2 Latency

The latency $\delta^{u,y}$ for the tasks of a given application y associated with patient u is composed of two parts : 1) the round-trip communication delay between fog node and patient, and 2) the queuing and computation delay. Hence, the application latency can be written as

$$\delta^{u,y} = \delta^{u,y}_{\text{cmp}} + \delta^{u,y}_{\text{com}}, \quad \forall u \in \mathcal{U}, y \in \mathcal{Y}. \tag{3}$$

where $\delta^{u,y}_{\text{cmp}}$ is the queuing and computation delay, while $\delta^{u,y}_{\text{com}}$ is the communication delay. Recall that the application's task load is split across replicas, hence the queuing and computation delay $\delta^{u,y}_{\text{cmp}}$ for application y of patient u is equal to the longest queuing and computation delay $\delta^{r,u,y,f}_{\text{cmp}}$ of its replicas. Moreover, if the node is unavailable, we associate a high computation delay value, such that

$$\delta^{u,y}_{\text{cmp}} = \max_{\substack{r=1,\ldots,R \\ f \in \mathcal{F}}} (\delta^{r,u,y,f}_{\text{cmp}} \cdot W^{r,u,y,f}), \quad \forall u \in \mathcal{U}, y \in \mathcal{Y}. \tag{4}$$

Table 4.2 Table of notations

$\mathcal{T}$	Time system.
$\mathcal{K}$	Set of fog domains.
$\mathcal{F}$	Set of fog nodes.
$\mathcal{F}_k$	Set of fog nodes in fog domain $k \in \mathcal{K}$.
$\mathcal{U}$	Set of monitored patients.
$\mathcal{Y}$	Set of supported application types (i.e., ECG, temperature, etc.).
$\mathbf{l}$	$\|\mathcal{F}\|$ vector representing the locations of the fog nodes.
$\mathbf{c}$	$\|\mathcal{F}\|$ vector representing the remaining load capacity of fog nodes.
$\mathbf{h}$	$\|\mathcal{F}\|$ vector representing the failure rates of fog nodes.
$\mathbf{p}$	$\|\mathcal{U}\|$ vector representing the location of monitored patients.
$\mathbf{Q}$	$\|\mathcal{U}\| \times \|\mathcal{Y}\|$ matrix representing the task load of applications.
$\mathbf{T}$	$\|\mathcal{U}\| \times \|\mathcal{Y}\|$ matrix denoting the threshold latency requirements for applications.
$\mathbf{A}$	$\|\mathcal{U}\| \times \|\mathcal{Y}\|$ matrix representing the min. availability requirements for applications.
$\mathbf{W}$	$R \times \|\mathcal{U}\| \times \|\mathcal{Y}\| \times \|\mathcal{F}\|$ matrix representing whether an application replica r of application task load u of type y is mapped to a fog node f or not.
$\tilde{\mathbf{Q}}$	$R \times \|\mathcal{U}\| \times \|\mathcal{Y}\| \times \|\mathcal{F}\|$ matrix representing the distribution of applications' task loads, where $\tilde{Q}^{r,u,y,f}$ denotes a partial load of $Q^{u,y}$ redirected to a replica r that is placed in fog node f.
R	Max. number of replicas allowed for an application.

For simplicity's sake, we derive $\delta_{\text{cmp}}^{r,u,y,f}$ using the standard M/M/1 queuing model. To be noted that our framework can be easily adapted to other queuing models. Hence, $\delta_{\text{cmp}}^{r,u,y,f}$ is computed as

$$\delta_{\text{cmp}}^{r,u,y,f} = \frac{1}{\rho^f - \tilde{Q}^{r,u,y,f}}, \tag{5}$$

where ρ^f is the service rate of fog node f.

Similarly, the communication delay associated to user u for application y is upper-bounded by the highest communication delay of its replicas among fog nodes, denoted $\delta_{\text{com}}^{r,u,y,f}$, such that

$$\delta_{\text{com}}^{u,y} = \max(\delta_{\text{com}}^{r,u,y,f} \cdot W^{r,u,y,f}), \quad \forall u \in \mathcal{U}, y \in \mathcal{Y}. \tag{6}$$

The replica communication delay $\delta_{\text{com}}^{r,u,y,f}$ is further decomposed as follows :

$$\delta_{\text{com}}^{r,u,y,f} = \delta_{\text{com,u}\longrightarrow\text{f}}^{r,u,y,f} + \delta_{\text{com,f}\longrightarrow\text{u}}^{r,f,u,y} \tag{7}$$

where $\delta_{\text{com,u}\longrightarrow\text{f}}^{r,u,y,f}$ (resp. $\delta_{\text{com,f}\longrightarrow\text{u}}^{r,u,y,f}$) denotes the transmission delay of the wireless link from patient u towards fog node f (resp. from fog node f to patient u), expressed by

$$\delta_{\text{com,u}\longrightarrow\text{f}}^{r,u,y,f} = \frac{\tilde{Q}^{r,u,y,f}}{B \cdot \log_2\left(1 + \frac{\eta^u \cdot \|p^u - l^f\|^{-\beta}}{N_0}\right)}, \tag{8}$$

where B is the channel's bandwidth, η^u is the patient's transmit power, N_0 is the noise power, $\beta \geq 2$ is the path-loss exponent, and $\|\cdot\|$ denotes the Euclidean norm.

4.4.2 Problem Formulation

Given the defined system model above, our objective is to determine the optimal placement of applications replicas' containers and load provisioning that maximize the number of satisfied patients' tasks with respect to the applications' delay thresholds and the number of applications meeting the required availability, under constraints of fog nodes' availability and abiding by the fog nodes resources' capacity.

Thus, we start by defining the constraint for satisfied patients' tasks within the delay threshold given by

$$T^{u,y} - \delta^{u,y} \geq G \cdot (S^{u,y} - 1), \quad \forall u \in \mathcal{U}, y \in \mathcal{Y}, \tag{9}$$

where G is a large arbitrarily selected positive number used for linearization purposes (Coelho, s. d.), and $S^{u,y}$ is a binary variable indicating whether the delay $\delta^{u,y}$ is below the application's delay threshold $T^{u,y}$ or not.

The next constraint ensures that the satisfaction of a patient u request for application y is conditioned on the satisfaction of all the required application task loads of its replicas from the delay requirement perspective. It is written as

$$S^{u,y} \cdot Q^{u,y} \leq \sum_{r=1}^{R} \sum_{f \in \mathcal{F}} \tilde{Q}^{r,u,y,f}, \quad \forall u \in \mathcal{U}, y \in \mathcal{Y}. \tag{10}$$

Similarly to (9)–(10), the availability constraint for the application y of user u can be expressed by

$$A^{u,y} - \alpha^{u,y} \leq G \cdot (S_A^{u,y} - 1), \quad \forall u \in \mathcal{U}, y \in \mathcal{Y}, \tag{11}$$

where $S_A^{u,y}$ is the binary variable indicating whether the application availability $\alpha^{u,y}$ is above the required minimum availability $A^{u,y}$ or not. From an availability perspective, the satisfaction of application y for patient u is conditioned on ensuring that the required application task loads are adequately distributed to the replicas such that

$$S_A^{u,y} \cdot Q^{u,y} \leq \sum_{r=1}^{R} \sum_{f \in F} \tilde{Q}^{r,u,y,f}, \quad \forall u \in \mathcal{U}, y \in \mathcal{Y}. \tag{12}$$

We also guarantee that the total load allocated in each fog node does not exceed the node's capabilities, expressed by

$$c^f \geq \sum_{u \in \mathcal{U}} \sum_{y \in \mathcal{Y}} \sum_{r=1}^{R} \tilde{Q}^{r,u,y,f}, \quad \forall f \in \mathcal{F}. \tag{13}$$

Moreover, two replicas of the same container of a patient's application y should not be hosted in the same fog node, i.e.,

$$\sum_{r=1}^{R} W^{r,u,y,f} \leq 1, \quad \forall u \in \mathcal{U}, y \in \mathcal{Y}, f \in \mathcal{F}, \tag{14}$$

and a replica can only be hosted in one fog node, thus

$$\sum_{f \in \mathcal{F}} W^{r,u,y,f} \leq 1, \quad \forall 1 \leq r \leq R, u \in \mathcal{U}, y \in \mathcal{Y}. \tag{15}$$

Also, a container's replica cannot be placed in a fog node unless there is a portion of a task load that is executed within this fog node. Hence,

$$\tilde{Q}^{r,u,y,f} \leq G \cdot W^{r,u,y,f}, \quad \forall 1 \leq r \leq R, u \in \mathcal{U}, y \in \mathcal{Y}, f \in \mathcal{F}. \tag{16}$$

As the latency of patients' tasks may vary due to the sporadic availability and locations of fog nodes, we aim in this problem to jointly maximize the number of deployed applications meeting the availability requirements and the specified delay thresholds. Hence, the problem can be formulated as follows :

$$\max_{\mathbf{W},\tilde{\mathbf{Q}}} \quad \sum_{u\in\mathcal{U}} \sum_{y\in\mathcal{Y}} (S^{u,y} + S_A^{u,y}) \tag{P1}$$

$$\text{s.t.} \quad (9) - (16).$$

Problem (P1) is NP-Hard. To illustrate this, consider for simplicity purposes the single replica problem aiming to place containerized applications in fog nodes to maximize the number of patients' applications satisfying their latency requirements. Moreover, let us focus on a scenario where the availability requirements of applications are already satisfied. This particular instance of problem can trivially be comprehended as the well-known Generalized Assignment Problem (GAP) (Ross et Soland, 1975). Indeed, the goal in GAP is to find a maximum total profit mapping between m agents and n heterogeneous tasks. Each task is assigned, to exactly, one agent given it has sufficient capacity. Moreover, each task may have a different profit based on the assigned agent.

By analogy to our problem, agents can be assimilated to fog nodes and containerized applications to tasks. The task profit is the number of answered patients' application requests within the latency requirements. Since GAP is known to be NP-hard, then, by restriction, our simplified problem is also NP-hard.

4.5 Proposed Smart Solution : Differential Evolution-Deep-Q-Networks (DE-DQN)

We present in this section our approach to solving the problem (P1), called DE-DQN. The main idea is to use differential evolution (DE), which we enhance with a DRL-driven deep-Q-learning (DQN) algorithm. The goal of DQN is to serve as the AOS mechanism (Fialho, 2010) to optimally control the DE parameters during execution.

4.5.1 Background on Differential Evolution

DE is an efficient population-based search algorithm. It is much more straightforward to implement, requiring fewer control parameters while performing better in a variety of fields (Das et Suganthan, 2010) compared to other evolutionary algorithms (EAs). In DE terminology, a "parent" from the current generation is called "target", a mutant obtained through the mutation operation is called "donor", and finally, an offspring formed by recombining the donor with the target is called "trial". Like conventional EAs, DE has four main stages : initialization, mutation, crossover, and selection. It starts with a generation of NP individuals (initialization stage). Unlike other EAs, DE generates the next generation by perturbing the scaled differences of selected and distinct population members (mutation stage). Each mutation strategy has its exploration/exploitation trade-offs (Sharma et al., 2019 ; Das et Suganthan, 2010).

To clarify the DE operations, assume that o_j and v_j are the j^{th} offspring and parent at the current generation, respectively. Then, the following mutation strategies are defined :

$$\begin{aligned}
&\textbf{DE/rand/1}: o_j = v_{b_1} + M(v_{b_2} - v_{b_3})\\
&\textbf{DE/rand/2}: o_j = v_{b_1} + M(v_{b_2} - v_{b_3}) + M(v_{b_4} - v_{b_5})\\
&\textbf{DE/best/1}: o_j = v_{\text{best}} + M(v_{b_1} - v_{b_2})\\
&\textbf{DE/best/2}: o_j = v_{\text{best}} + M(v_{b_1} - v_{b_2}) + M(v_{b_3} - v_{b_4})\\
&\textbf{DE/target-to-best/1}: o_j = v_j + M(v_{\text{best}} - v_j)\\
&\qquad\qquad\qquad\qquad\qquad + M(v_{b_1} - v_{b_2})\\
&\textbf{DE/rand-to-best/2}: o_j = v_{b_1} + M(v_{\text{best}} - v_{b_1})\\
&\qquad\qquad\qquad\qquad + M(v_{b_2} - v_{b_3}) + M(v_{b_4} - v_{b_5}),
\end{aligned}$$

where M is a scaling factor, v_{best} is the best parent in the population, and b_i ($i = 1, \ldots, 5$) are randomly generated indexes within the range $[1, NP]$. Crossover is realized over a target individual $\theta_j = \{\theta_{j,1}, \ldots, \theta_{j,J'}\}$ and a donor individual $\psi_j = \{\psi_{j,1}, \ldots, \psi_{j,J'}\}$ to generate a trial $\phi_j = \{\phi_{j,1}, \ldots, \phi_{j,J'}\}$, such that, $\forall j' = 1, \ldots, J'$,

$$\phi_{j,j'} = \begin{cases} \psi_{j,j'} & \text{if } rand_{j,j'}[0,1] \leq C \text{ or } j' = j_{rand} \\ \theta_{j,j'} & \text{otherwise} \end{cases} \tag{18}$$

where $C \in [0, 1]$ is the crossover parameter that determines whether the mutation strategy is applied or not. The function $rand_{j,j'}[0, 1]$ generates a uniformly distributed random number between $[0, 1]$ and j_{rand} is a randomly chosen index that ensures at least one component of the donor individual ψ_j is inherited in the trial ϕ_j.

Following the crossover phase, a selection phase is usually applied to maintain a stable population over the next generations. The selection operation determines whether a target or a trial offspring should be carried onto the next generation through the following operation :

$$\theta_j^+ = \begin{cases} \phi_j & \text{if } \xi(\phi_j) \leq \xi(\theta_j) \\ \theta_j & \text{if } \xi(\phi_j) > \xi(\theta_j), \end{cases} \tag{19}$$

where θ_j^+ represents the next generation target, and $\xi(\cdot)$ is an evaluation function that needs to be maximized. [2] This ensures that the population quality never deteriorates over time.

Experimental analysis has shown that different operators, e.g., encoding schemes, parameter settings, mutation, crossover strategies, etc., perform better for specific optimization problems. For instance, authors of (SHARMA et al., 2019; DAS et SUGANTHAN, 2010) have shown that adequately choosing the mutation strategy at specific stages can further improve the algorithmic performance.

In this context, AOS guides the choice of the genetic operator at each generation according to a reward function. This can be very challenging in a dynamic, large-scale, and heterogeneous environment such as ours. Moreover, a significant drawback of EAs is their unstable convergence as it depends on the initial population. We consequently leverage a DRL-based method to capture the algorithm's state at each generation, select an operator to be applied, and subsequently compute a reward to determine the optimal actions that enable the DE algorithm to increase its convergence rate.

Inspired by (SHARMA et al., 2019), we adopt DQN as the AOS technique in this work to select the adequate mutation strategy at each generation. We denote our approach by DE-DQN. In what follows, we first describe the adaptations done to the DE algorithm to solve our problem. Then, we detail our integration approach of DQN as the AOS method.

---

[2] Further details about $\xi(\cdot)$ will be provided in the next subsection.

4.5.2 Adaptation of Differential Evolution

Since DE is designed to solve optimization problems with real number parameters, it cannot be used directly to solve our formulated problem. This subsection focuses on the adaptation procedures for this specific problem, including the individual encoding scheme and initialization.

4.5.2.1 Individual encoding scheme

Within a population, we define at first the maximum number of replicas R for a given application request. We greedily determine R by sorting through fog nodes according to their estimated failure rates, i.e., by appending the worst fog nodes progressively into a set $\mathcal{F}'$, such that

$$1 - \prod_{f \in \mathcal{F}'} \left(1 - \sum_{r=1}^{R} \lambda^f\right) \geq A^{u,y}, \quad \forall u \in \mathcal{U}, y \in \mathcal{Y}, \tag{20}$$

where $\mathcal{F}' \neq \emptyset$. To prevent the single point of failure scenario, i.e., when $A^{u,y} \to 0$ and thus $T^{u,y} \to \infty$, we ensure the minimum number of replicas for an application $R \geq 2$.

Let θ_j be a target offspring of dimension R representing a potential solution to our optimization problem and let $\{\theta_{j,1}, \theta_{j,2}, \ldots, \theta_{j,j'}, \ldots, \theta_{j,R}\}$ denote its fog node components. Although the target offspring has a fixed dimension, the use of dynamic-sized offsprings remains possible in our model by adding filler (virtual) fog nodes, which have no impact on the system. Furthermore, an internal routine maintains the correspondence between an index j' and its associated fog node f for decoding purposes. To ensure constraints (14)–(15) are respected, we validate that two components of θ_j do not point towards the same fog node f.

Then, the evaluation function $\xi(\cdot)$ is defined such that, for an offspring θ_j, it is expressed by

$$\xi(\theta_j) = -\frac{(A^{u,y}_{\theta_j} - A^{u,y})^2}{\varkappa^A} - \frac{(T^{u,y}_{\theta_j} - T^{u,y})^2}{\varkappa^T}, \tag{21}$$

where $A^{u,y}_{\theta_j}$ and $T^{u,y}_{\theta_j}$ are the current availability and delay provided through the components of θ_j, respectively. The variables $\varkappa^A$ and $\varkappa^T$ are scaling penalty factors that discourage both under-provisioning and over-provisioning. They are defined as

$$\varkappa^A = \begin{cases} \varkappa^A_1, & \text{if } A^{u,y}_{\theta_j} - A^{u,y} > 0 \\ \varkappa^A_2, & \text{otherwise,} \end{cases} \tag{22}$$

$$\varkappa^T = \begin{cases} \varkappa^T_1, & \text{if } T^{u,y}_{\theta_j} - T^{u,y} > 0 \\ \varkappa^T_2, & \text{otherwise,} \end{cases} \tag{23}$$

where $(\varkappa^A_1, \varkappa^A_2, \varkappa^T_1, \varkappa^T_2 \in \mathbb{R}^*)$ with $\varkappa^A_2 < \varkappa^A_1$ and $\varkappa^T_2 < \varkappa^A_1$. Under-provisioning induces non-satisfaction of the QoS requirements, whereas over-provisioning may endanger future request placement.

To adapt the mutation process, we start by characterizing, for any offspring θ_j, the inherent contribution $\sigma(\theta_{j,j'})$ of a composing fog node $\theta_{j,j'}$ towards the satisfaction of the QoS requirements as follows :

$$\sigma(\theta_{j,j'}) = \varphi^A \cdot (\lambda_{j'} - A^{u,y}) + \left(\frac{\varphi^T}{(\delta^{r,u,y,j'}_{\text{cmp}} + \delta^{r,u,y,j'}_{\text{com}}) - T^{u,y}} \right), \tag{24}$$

where $(\varphi^A, \varphi^T \in \mathbb{R} \mid \varphi^A + \varphi^T = 1)$ are weights indicating the perceived priority of the availability and latency QoS requirements. Recall that we prioritize availability over latency, hence $\varphi^A > \varphi^T$. Afterwards, we define the relative contribution $\tilde{\sigma}(\theta_{j,j'})$ of a composing fog node $\theta_{j,j'}$ in comparison to its peers, written as

$$\tilde{\sigma}(\theta_{j,j'}) = (NP - 1) \cdot \frac{\sigma(\theta_{j,j'})}{\sum\limits_{\substack{j''=1 \\ j'' \neq j'}}^{R} \sigma(\theta_{j,j''})}, \tag{25}$$

where NP is the population size, and whereupon the composing fog nodes are accordingly sorted. We then adjust the mutation process in the following manner.

Let θ_1 and θ_2 be two offsprings involved in the mutation process and o the resulting mutant. Then, the offsprings are redefined according to the relations :

$$\theta_1 - \theta_2 = \begin{bmatrix} \sigma(\theta_{1,1}) - \sigma(\theta_{2,1}) \\ \sigma(\theta_{1,2}) - \sigma(\theta_{2,2}) \\ \cdots \\ \sigma(\theta_{1,R}) - \sigma(\theta_{2,R}) \end{bmatrix},$$

and

$$\theta_1 + \theta_2 = \begin{bmatrix} \sigma(\theta_{1,1}) + \sigma(\theta_{2,1}) \\ \sigma(\theta_{1,2}) + \sigma(\theta_{2,2}) \\ \cdots \\ \sigma(\theta_{1,R}) + \sigma(\theta_{2,R}) \end{bmatrix}.$$

We retain a similar use to the classic mutation scale factor M as a search exploration/exploitation adjustment parameter. In our context, we assume a rule enforced for M components such that, when $o = \sigma(\theta_{1,j'}) - \sigma(\theta_{2,j'}) > 0$, we retain in the mutant offspring $\sigma(\theta_{1,j'})$ at the expense of $\sigma(\theta_{2,j'})$. The aim is to create a mutant offspring with components characterized by a strong inherent contribution to the QoS requirements. For $o = \sigma(\theta_{1,j'}) + \sigma(\theta_{2,j'})$, we leverage a greedy algorithm to determine which components of $\sigma(\theta_{1,j'})$ should be transferred into $\sigma(\theta_{2,j'})$ in order to maximize $\xi(o)$, which implies that the most-contributing (resp. least-contributing) components of $\sigma(\theta_{1,j'})$ may be retained to avoid under-provisioning (resp. over-provisioning).

4.5.3 DQN State and Reward Definitions

The execution of the adapted DE algorithm for a maximum number of I iterations constitutes the environment of the DQN algorithm. A state s_t is a collection of features at step t. The potential actions taken by DQN are the set of mutation strategies available, and the action a_t is the selected strategy applied at step t.

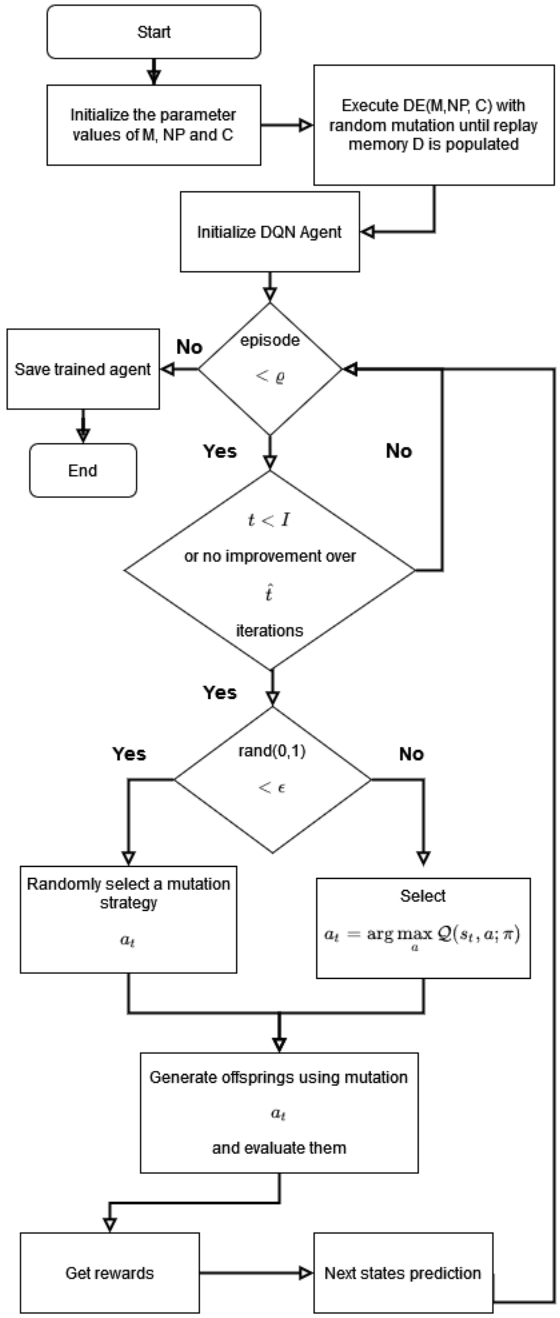

FIGURE 4.4 Flowchart of the DE-DQN training algorithm

Once a mutation strategy is applied, an immediate reward r_t is obtained before transiting to a new state s_{t+1}. In the training phase, DQN uses a deep neural

Algorithm 2: DE-DQN algorithm

Initialize parameter values of DE(M, NP, C) where M : mutation scaling factor, NP : population size and C : crossover rate;
Execute DE with a randomly selected mutation strategy to populate the replay memory $\mathcal{D}$;
Initialize action-value function $\mathcal{Q}$ with random weights π;
Initialize target action-value function $\hat{\mathcal{Q}}$ with weights π^-;
for *episode = 1 to* ϱ **do**
  $t = 0$;
  while $t < I$ *or no improvement over* $\hat{t}$ *iterations* **do**
    for i = *1 to NP* **do**
      if *rand(0,1)* $< \epsilon$ **then**
        Randomly select a mutation strategy a_t;
      else
        Select $a_t = \arg\max_a \mathcal{Q}(s_t, a; \pi)$
      end
      Generate trial offspring o_i from parent v_i using mutation a_t;
      Evaluate trial offspring and keep the best between o_i and v_j;
      Store transition $(s_t, a_t, \mathcal{R}_t, s_{t+1})$ in $\mathcal{D}$;
      Sample a random transition mini-batch from $\mathcal{D}$;
      Set $\mathcal{R}_j = \begin{cases} \mathcal{R}_t, \text{ if episode terminates} \\ \mathcal{R}_t + \gamma \max_{a'} \hat{\mathcal{Q}}(s_{t+1}, a', \pi^-), \text{ otherwise;} \end{cases}$ Perform a gradient descent step on $[\mathcal{R}_j - \mathcal{Q}(s_j, a_j, \pi)]^2$ w.r.t π;
      In every $\mathcal{L}$ steps, reset $\hat{\mathcal{Q}} = \mathcal{Q}$;
      $t = t + 1$;
    end
  end
end

network whose role is to predict the Q-values $\mathcal{Q}(s_t, a_t, \pi)$ used to select an action a, given the state s_t with weights π. The training process, shown in Algorithm 2, starts by running DE with a randomly selected mutation strategy for a fixed number of steps, thus generating enough observations to populate a fixed-size replay memory. Then, we execute DE for ϱ number of iterations. The stopping criterion for each execution is either reaching the maximum number of iterations I or no fitness function improvement over predefined time steps.

The DQN state consists of various normalized features with values in the interval

$[0, 1]$, and are summarized in Table 4.3 to encode key properties of the current solution environment, inspired by (SHARMA et al., 2019). Assume that o_j and v_j are the j^{th} offspring and parent, respectively. Also, let v_{best} be the best parent in the population and b_i $(i = 1, \ldots, 5)$ randomly generated indexes within the range $\{1, \ldots, NP\}$. Given that Z is the set of available mutation strategies, we provide below brief explanations of the features retained :

- The fitness of current parent $\xi(v_j)$,
- The current population mean fitness,
- The standard deviation of fitness values in the population
- The current generation count,
- The stagnation count to indicate the last time an improvement of the best fitness was achieved,
- The difference between $\xi(v_{\text{best}})$ and $\xi(v_{\text{b}_\text{i}}), \forall i \in \{1, \ldots, 5\}$,
- The level of similarity between components used in v_{best} and $v_{\text{b}_\text{i}}, \forall i \in \{1, \ldots, 5\}$ to assist in strategy learning that best combines these solutions.

We also leverage historical information to enrich the environment description further. To this end, we consider the stagnation count history and the following metrics :

- $\zeta_1(i, z) = \xi(v_j) - \xi(o_j)$ is the i^{th} fitness improvement entry for offspring o_j over parent v_j, when applying mutation strategy $z \in Z$,
- $\zeta_2(i, z) = \xi(v_{\text{best}}) - \xi(o_j)$, the i^{th} fitness improvement entry for offspring o_j over the best parent v_{best} in the population, when applying mutation strategy $z \in Z$,

- $\zeta_3(i, z)$, the i^{th} fitness improvement entry for offspring o_j over the median fitness of the parent population, when applying mutation strategy $z \in Z$.

Finally, DQN takes the fitness improvement from a parent to an offspring, expressed by $\mathcal{R} = \max\{\xi(v_j) - \xi(o_j), 0\}$, as the reward function.

TABLE 4.3 Summary of state features (SHARMA et al., 2019)

<table>
<tr><th>Feature</th><th>Description</th></tr>
<tr><td>$\frac{\xi(v_j)-\xi(v_{\text{best}})}{\xi(v_{\text{worst}})-\xi(v_{\text{best}})}$</td><td>Fitness of current parent v_j</td></tr>
<tr><td colspan="2">v_j denotes the j^{th} parent solution of the population and $\xi(v_j)$ its fitness, $\xi(v_{\text{best}})$ and $\xi(v_{\text{worst}})$ are the parents with the best and worst fitness within the episode, respectively.</td></tr>
<tr><td>$\frac{\frac{\sum_j^{NP}\xi(v_j)}{NP}-\xi(v_{\text{best}})}{\xi(v_{\text{worst}})-\xi(v_{\text{best}})}$</td><td>Mean fitness of population</td></tr>
<tr><td colspan="2">NP denotes the population size</td></tr>
<tr><td>$\frac{std(\xi(v_j))}{std_{\max}}, \forall j = \{1, \dots, NP\}$</td><td>Standard deviation of population members' fitness values</td></tr>
<tr><td colspan="2">$std(\cdot)$ denotes the standard deviation and $std_{\max}$ is the value when $\frac{NP}{2}$ solutions have $\xi(v_{\text{best}})$ and the remaining have $\xi(v_{\text{worst}})$</td></tr>
<tr><td>$\frac{\mathcal{G}}{\mathcal{G}_{\max}}$</td><td>Current generation count</td></tr>
<tr><td colspan="2">$\mathcal{G}$ and $\mathcal{G}_{\max}$ are the current and max. number of generations, respectively.</td></tr>
<tr><td>$\frac{\mathcal{S}}{\mathcal{G}_{\max}}$</td><td>Stagnation count</td></tr>
<tr><td colspan="2">$\mathcal{S}$ denotes the stagnation counter</td></tr>
<tr><td>$\frac{\xi(v_{b_i})-\xi(v_{\text{best}})}{\xi(v_{\text{worst}})-\xi(v_{\text{best}})}, \forall i \in \{1, \dots, 5\}$</td><td>Fitness difference to best parent</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td>$\frac{|v_{\text{best}} \cap v_{b_i}|}{R}, \forall i \in \{1, \dots, 5\}$</td><td>Similarity between v_{best} and v_{b_i} components</td></tr>
<tr><td colspan="2">$|\cdot|$ denotes the set cardinality and R is the maximum number of allowed components.</td></tr>
</table>

TABLE 4.4 Hyperparameters of DE-DQN

Hidden layers	4
Hidden nodes per hidden layer	128
Batch size	256
Epsilon greedy probability ϵ	0.5
Discount factor	0.9
Learning rate	10^{-4}

4.5.4 Complexity Analysis

This section presents the time complexity of DE-DQN during the training phase. The first step for the DE portion of DE-DQN is to initialize the population of NP solutions. Each solution has a length of R. Subsequently, we apply selection, crossover, and mutation operations to generate J offsprings, which are evaluated to verify their fitness. This process is repeated over $\mathcal{G}_{\max}$ number of generations. We hence obtain for the DE portion of DE-DQN a time complexity of $\mathcal{O}(\mathcal{G}_{\max} \times NP \times R \times J)$ (KUMAR et A. K. SINGH, 2018).

For DQN training, the computational complexity mainly involves collecting transitions and backpropagation operations. With $\mathcal{D}$ as the replay memory size, I as the number of iterations, $\tilde{d}$ as the number of stored transitions from replay memory, and assuming θ and ϖ respectively as the number of layers and layer units, the training complexity is $\mathcal{O}(\theta \times \varpi \times I \times \tilde{d})$

Consequently, the complexity of DE-DQN can be estimated as $\mathcal{O}(\mathcal{G}_{\max} \times NP \times R \times J)$ + $\mathcal{O}(\theta \times \varpi \times I \times \tilde{d})$.

4.6 Experimental evaluation

We hereafter describe our experimental setting to evaluate the merits of our proposal in environments with unreliable components such as FC networks. We implement scenarios to evaluate the performances of the following algorithms :

- *Random :* A placement strategy where random fog nodes are selected to host patient tasks until QoS requirements are met
- *Greedy :* We select a random subset of fog nodes over which we search for the best task placement. This process is repeated three times to allow for sufficient exploration of the solution space
- *Adapted-DE :* It is the AOS-less variant of our DE-DQN proposal. We consider 6 mutation strategies in our DE implementations : DE/rand/1, DE/rand/2, DE/best/1, DE/best/2, DE/target-to-best/1, DE/rand-to-best/2, as described in Subsection 4.5.1, and corresponding respectively to DE-1, DE-2, DE-3, DE-4, DE-5 and DE-6 in our illustrations.

At each time slot, the fog controller takes task provisioning decisions regarding the placement of application requests onto fog nodes. In our evaluation, we adopt a strict placement policy where all QoS requirements (in terms of latency and availability) must be satisfied to declare a successful request placement.

Unless stated otherwise, we assume a scenario where 100 fog nodes and 50 patients are uniformly spread within a 1 km radius. All nodes are under the control of a single fog domain controller. Also, we assume the arrival of 50 new application requests in each time slot, which must be placed within the FC network. We assign to each application a QoS requirement tuple made through random selection from the sets of availability requirements $\{0.99, 0.999, 0.99999\}$ and of latency

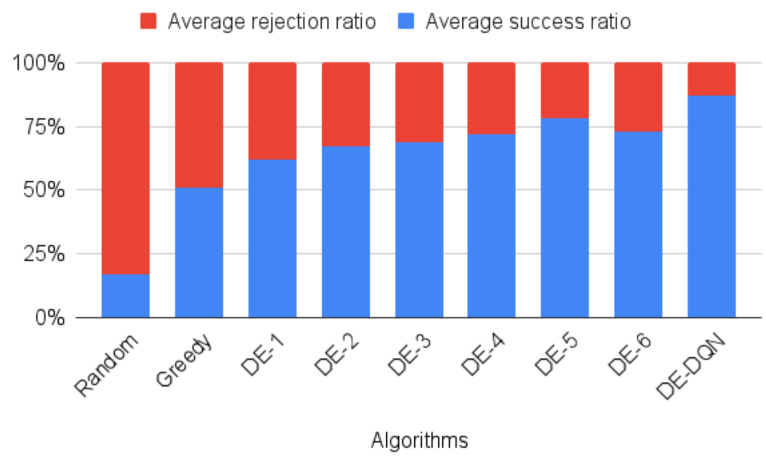

FIGURE 4.5 Performance comparison of proposed and benchmark algorithms.

requirements $\{30 \text{ ms}, 50 \text{ ms}, 100 \text{ ms}\}$. The annualized failure rate (AFR) value of each fog node is uniformly sampled from the range $[0.1\%, 15\%]$ and a load capacity value from the interval $[0.5 \text{ GB}, 3 \text{ GB}]$. Moreover, we consider the available fog node's RAM memory to be within the range $[1 \text{ GB}, 4 \text{ GB}]$. Any node within the network is assumed to transmit packets of length $L = 1500$ bits over a wireless channel with bandwidth $B = 20$ MHz. We assume that wireless links experience path-loss with coefficient $\beta = 3$ and noise power $N_0 = 0.02$ Watts (XIAO, 2005; A. MSEDDI et al., 2019). The maximum transmit power p^u is set to 23dBm (CHANG et al., 2017).

We present the values of relevant hyperparameters for our deep neural network implementation in Table 4.4. We specifically leverage a multi-layer perceptron neural network with rectified linear unit (ReLU) activation. Our DQN implementation is based on the Stable Baselines implementation with a custom policy network to fit our solution (RAFFIN et al., 2019). We set the mutation scaling factor M to 0.5 and the crossover rate parameter C to 0.8. The population size NP is set to 50 and the maximum number of generations $\mathcal{G}_{\max}$ to 10, while the maximum number

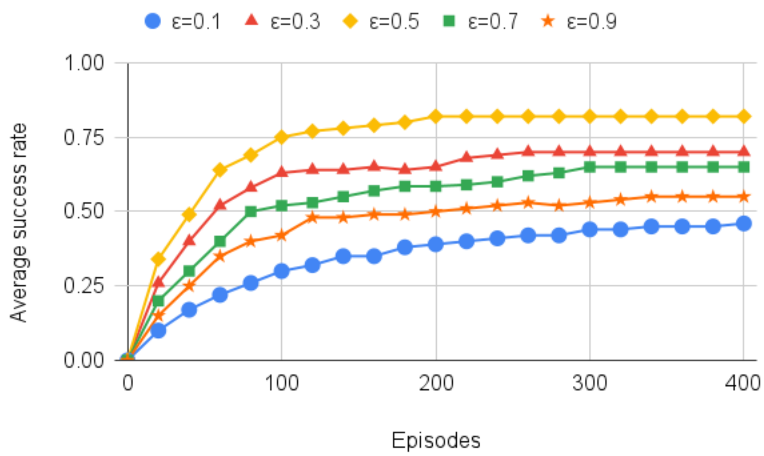

FIGURE 4.6 Training convergence behavior of DE-DQN algorithm.

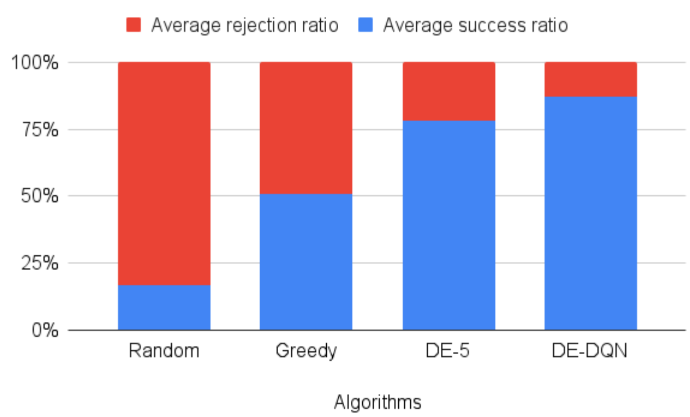

FIGURE 4.7 Success ratio versus the number of fog nodes - 100 fog nodes

of replicas allowed R is 4.

In Fig. 4.5, we evaluate and compare the success ratio performances of the baseline and proposed algorithms. The success ratio is defined as the ratio of the tasks successfully executed within the FC framework to the total number of tasks. Random and greedy-based algorithms perform worse than DE-based ones, while DE-DQN outperforms all approaches, including the Adapted-DE ones (i.e., DE-1

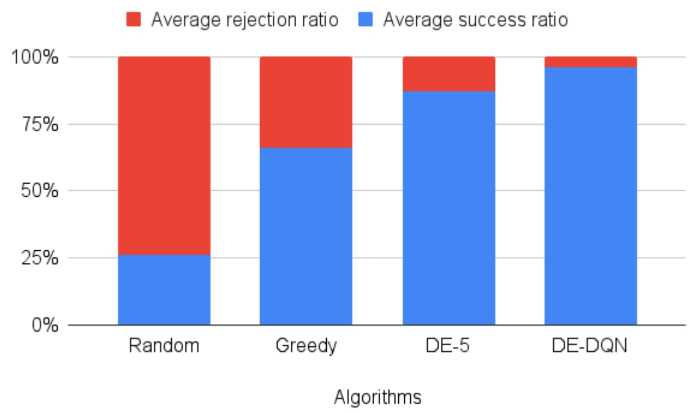

FIGURE 4.8 Success ratio versus the number of fog nodes - 150 fog nodes

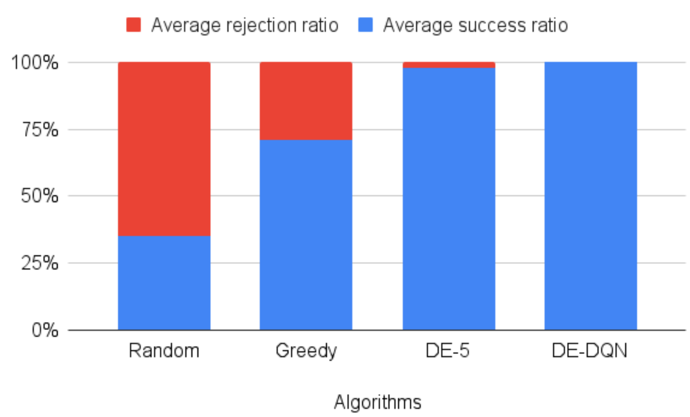

FIGURE 4.9 Success ratio versus the number of fog nodes - 200 fog nodes

to DE-6). Indeed, due to the dependence of population-based search algorithms on population initialization, algorithms DE-1 to DE-6 may perform poorly. In contrast, due to its adaptive selection of mutation strategies, DE-DQN can explore the solution space more efficiently, thus leading to optimal resource allocation.

Based on the same setup as for Fig. 4.5, we illustrate in Fig. 4.6 the average success ratio of our proposed DE-DQN algorithm as a function of the training episodes and for different ϵ probability values. For any ϵ, the performance improves with

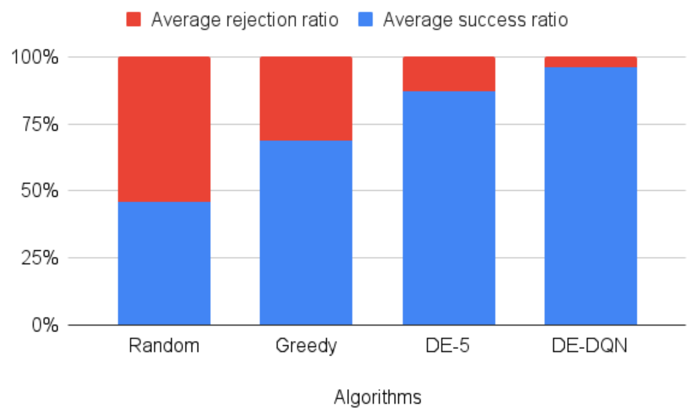

FIGURE 4.10 Success ratio versus the fog nodes' AFR - Fog nodes' AFR less than 3%

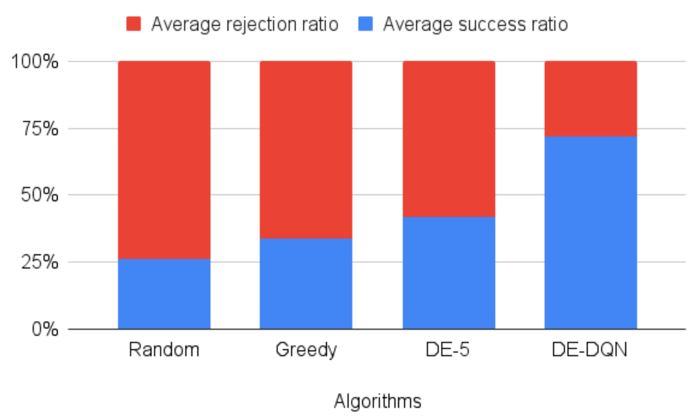

FIGURE 4.11 Success ratio versus the fog nodes' AFR - Fog nodes' AFR in range [3%-15%]

the number of episodes. This is expected since training updates the neural network weights to better values. Also, we notice that $\epsilon = 0.5$ provides the best performance. Indeed, this value allows for balancing between exploring new mutation strategies and adopting the up-to-date best strategy. When $\epsilon < 0.5$, DE-DQN limits its search in the solution space, while for $\epsilon > 0.5$ the algorithm tends to behave

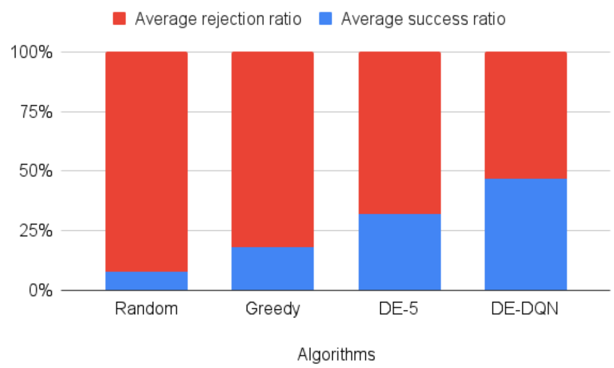

Figure 4.12 Success ratio versus the fog nodes' AFR - Fog nodes' AFR over 15%

randomly more often. Finally, DE-DQN converges the fastest for $\epsilon = 0.5$ (around 200 episodes), thus making it the most practical approach.

In Fig. 4.7, 4.8 and 4.9, we compare the success ratio performances of Random, Greedy, DE-5, and DE-DQN for different numbers of fog nodes. As available fog nodes increase from 100 to 200, all algorithms present improved performances. Indeed, a higher number of fog nodes facilitates the satisfaction of the QoS requirements and thus responds to a higher number of application requests. Nevertheless, the random and greedy-based solutions achieve a small gain compared to DE-5 and DE-DQN since they cannot efficiently satisfy the latency requirements. Comparatively, DE-DQN performs better than DE-5, with DE-DQN being able to place all application requests in a 200 fog nodes' FC network.

We examine in Fig. 4.10, 4.10 and 4.12, the impact of unreliability on application request placement by varying the fog nodes' AFR distribution. Specifically, we define three scenarios comprising 100 fog nodes and where we set the fog nodes' AFR to be either under 3% (Fig. 4.10), in the range $[3\%, 15\%]$ (Fig. 4.11), or over 15% (Fig. 4.12). The best success ratios are obtained for AFR$< 3\%$, where DE-DQN achieves the highest performance of 96%. DE-DQN success ratio drops to 72% for AFR $\in [3\%, 15\%]$ and 47% for AFR $> 15\%$. This is expected given our strict placement policy, which requires complete satisfaction of the QoS requirements. First, in high AFR environments, the number of replicas is related to AFR through (2) and (11).

Furthermore, from (3)–(8), it is clear that additional replicas (depending on distance l^f to the patient) increase the probability of a latency requirement violation, in particular for later requests. For simplicity, we define a maximum number of replicas $R = 4$ for any given application request, which aligns with the consistency requirement for replication due to added synchronization latency. This extra synchronization time depends on the actual replication pattern, used consistency model, number of replicas to reconcile, and the link quality between replicas.

Consequently, deploying high AFR fog nodes remains highly detrimental to latency requirements regardless of the improvements in terms of application availability. Indeed, by observing the performance degradation rate in Figs. 4.7, 4.8, 4.9, 4.10, 4.10 and 4.12, it seems that the proportion of high AFR fog nodes within the FC network is a more critical parameter than the total number of fog nodes. In contrast, low AFR environments, i.e., with a high proportion of low AFR fog nodes, seem to significantly improve the probability of satisfying both the availability and latency requirements. However, it can be argued that the cost of low AFR fog nodes would be substantially higher than their high AFR counterparts.

Furthermore, the geographic distribution of fog nodes in real environments, with regard to patients, is a non-trivial aspect given the transmission delay requirements (8) and power limitation of IoT devices (Kartakis et al., 2016; Raj et Steingart, 2018). Finally, according to these results, the design, deployment, and use of fog nodes for critical applications, such as remote health monitoring, require careful planning that trades-off between the number of fog nodes, the proportions of high and low AFR fog nodes, and their locations, hence, sustaining reliable FC services.

4.7 Conclusion

To enable the practicality of remote health monitoring as a means to counter the health impacts of the pandemic, stringent QoS requirements must be met. Despite the fact that fog computing is being touted for time-sensitive tasks such as healthcare applications, the impact of unreliable fog nodes is often overlooked.

This paper focused on this aspect by formulating a remote health monitoring problem as an FC placement problem within unreliable fog nodes. Such a problem is proven to be NP-hard. Consequently, we proposed DE-DQN, a novel DE algorithm enhanced by a DQN-based operation selection mechanism, to optimize the task placement strategy.

Through experimental evaluation, we proved the superior reliability performance of our proposed solution, in terms of average success ratio, compared to benchmarks. Moreover, by investigating the impact of several key parameters, we identified a design trade-off between the number of fog nodes and the latter's AFRs.

This work aimed to highlight the implications of the inherent unreliability in FC networks for sensitive applications such as remote health monitoring. By taking into account devices' AFR, the placement of applications with strict QoS requirements can be noticeably improved, as shown in our simulations. However, replacing most devices with high AFR in the FC networks may not be realistic due to the costs. In future work, we plan to develop a comprehensive testbed to gather more insights. Moreover, we intend to investigate the merits of alternative fault tolerance methods such as checkpointing.

REFERENCES

AAZAM, Mohammad et Eui-Nam HUH (2015). “E-HAMC : Leveraging Fog computing for emergency alert service”. In : *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Wrkshps. (PerCom Wrkshps.)* P. 518-523.

ANNIS, Tucker et al. (2020). “Rapid implementation of a COVID-19 remote patient monitoring program”. In : *J. American Medical Informatics Asso.* 27.8, p. 1326-1330.

BAKHSHI, Zeinab, Guillermo RODRIGUEZ-NAVAS et Hans HANSSON (2019). “Dependable fog computing : A systematic literature review”. In : *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. Ieee, p. 395-403.

BEHAR, Joachim A et al. (2020). “Remote health monitoring in the time of COVID-19”. In : *Physiological measurement* 41 (10), p. 1361-6579.

BERTINI, Matteo et al. (2016). “Remote monitoring of implantable devices : Should we continue to ignore it ?” In : *Int. J. of Cardiology* 202, p. 368-377.

BONOMI, Flavio et al. (2012). “Fog computing and its role in the Internet of Things”. In : *Proc. 1st Ed. MCC Wrkshp. Mob. Cloud Comput.* P. 13-16.

CAO, Xiaoyuan et al. (2015). “Heterogeneous Multi-domain Network Virtualization with End-to-End Differentiated Service Provisioning and Virtual Network

Organization". In : *Optical Fiber Commun. Conf.* Optical Society of America, Th4G-3.

Cascella, Marco et al. (2020). "Features, evaluation and treatment coronavirus (COVID-19)". In : *Statpearls*. StatPearls Publishing.

Chang, Zheng et al. (2017). "Energy efficient optimization for computation offloading in fog computing system". In : *IEEE Global Commun. Conf. (GLOBECOM)*, p. 1-6.

Coelho, Leandro (s. d.). *Linearizing the product of a binary and a continuous variable*. https://www.leandro-coelho.com/linearization-product-variables/. [Online ; accessed March-2022].

Craciunescu, Razvan et al. (2015). "Implementation of Fog computing for reliable E-health applications". In : *Proc. IEEE Asilomar Conf. Sig., Syst. Comput. (ACSSC)*, p. 459-463.

Dang, L Minh et al. (2019). "A survey on Internet of Things and cloud computing for healthcare". In : *Electronics* 8.7, p. 768.

Das, Swagatam et Ponnuthurai Nagaratnam Suganthan (2010). "Differential evolution : A survey of the state-of-the-art". In : *IEEE Trans. Evolutionary Comput.* 15.1, p. 4-31.

Dhanvijay, Mrinai M et Shailaja C Patil (2019). "Internet of Things : A survey of enabling technologies in healthcare and its applications". In : *Comput. Netw.*

DIEYE, Mouhamad, Mohamed Faten ZHANI et Halima ELBIAZE (2017). “On achieving high data availability in heterogeneous cloud storage systems”. In : *Proc. IFIP/IEEE Symp. Integrated Netw. Serv. Mgmt. (IM)*. Ieee, p. 326-334.

ELSAYED, E.A. (1996). *Reliability Engineering*. Reliability Engineering v. 1. Addison Wesley Longman. ISBN : 9780201634815.

FIALHO, Álvaro (2010). “Adaptive operator selection for optimization”. Thèse de doct. Université Paris Sud - Paris XI.

GIA, Tuan Nguyen et al. (2015). “Fog computing in healthcare Internet of Things : A case study on ECG feature extraction”. In : *Proc. IEEE Int. Conf. Comp. Info. Techno. ; Ubiquitous Comput. Commun. ; Dependable, Autonomic Secure Comput. ; Pervas. Intellig. Comput.* P. 356-363.

GILL, Phillipa, Navendu JAIN et Nachiappan NAGAPPAN (2011). “Understanding network failures in data centers : measurement, analysis, and implications”. In : *Proc. ACM SIGCOMM Conf.* P. 350-361.

GORDON, William J et al. (2020). “Remote patient monitoring program for hospital discharged COVID-19 patients”. In : *Applied Clinical Informatics* 11.05, p. 792-801.

HE, Jianhua et al. (2018). “Multitier fog computing with large-scale IoT data analytics for smart cities”. In : *IEEE Internet of Things J.* 5.2, p. 677-686.

HOSSAIN, M Shamim (2016). “Patient status monitoring for smart home healthcare”. In : *Proc. IEEE Int. Conf. Multimedia & Expo Wrkshps. (ICMEW)*. Ieee, p. 1-6.

HU, Pengfei et al. (2017). “Survey on fog computing : Architecture, key technologies, applications and open issues”. In : *J. Netw. Comp. Apps.* 98, p. 27-42.

HUANG, Yueh-Min et al. (2009). “Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architecture in wireless heterogeneous networks”. In : *IEEE J. Select. Areas Commun.* 27.4, p. 400-411.

KARTAKIS, Sokratis et al. (2016). “Demystifying low-power wide-area communications for city IoT applications”. In : *Proc. ACM Int. Wrkshp. Wireless Netw. Testbeds, Experimental Eval., Characteriz. (WNTEEC)*, p. 2-8.

KRAEMER, Frank Alexander et al. (2017). “Fog computing in healthcare-A review and discussion”. In : *IEEE Access* 5, p. 9206-9222.

KRASICH, Milena (2009). “How to estimate and use MTTF/MTBF would the real MTBF please stand up ?” In : *Proc. IEEE Ann. Reliab. Maintain. Symp. (RAMS)*, p. 353-359.

KUMAR, Jitendra et Ashutosh Kumar SINGH (2018). “Workload prediction in cloud using artificial neural network and adaptive differential evolution”. In : *Future Generation Computer Systems* 81, p. 41-52.

LLOYD’S (2018). *Lloyd’s Emerging Risk Report.* https://www.lloyds.com/~/media/files/news-and-insight/risk-insight/2018/cloud-down/aircyberlloydspublic2018final.pdf. [Online ; accessed 28-July-2020].

LÓPEZ, Gregorio, Viéctor CUSTODIO et José Ignacio MORENO (2010). “LOBIN : E-textile and wireless-sensor-network-based platform for healthcare monitoring in future hospital environments”. In : *IEEE Trans. Info. Techno. Biomed.* 14.6, p. 1446-1458.

MADSEN, Henrik et al. (2013). “Reliability in the utility computing era : Towards reliable fog computing”. In : *Proc. IEEE Int. Conf. Syst., Sig. Image Process. (IWSSIP)*, p. 43-46.

MANDELLOS, George J et al. (2009). *Requirements and Solutions for Advanced Telemedicine Applications*. INTECH Open Access Publisher.

MASIP-BRUIN, Xavier et al. (2016). “Fog-to-cloud computing (F2C) : The key technology enabler for dependable E-health services deployment”. In : *Proc. IEEE Mediter. Ad Hoc Network. Wrkshp. (Med-Hoc-Net)*, p. 1-5.

MAVROGIORGOU, Argyro et al. (2018). “Capturing the reliability of unknown devices in the IoT world”. In : *Proc. IEEE Int. Conf. IoT : Syst., Mgmt. Secu. (IoTSMS)*, p. 62-69.

MIRJALALI, Sheyda et al. (2021). “Wearable Sensors for Remote Health Monitoring : Potential Applications for Early Diagnosis of COVID-19”. In : *Adv. Materials Technol.*, p. 2100545.

MIZUMOTO, Kenji et al. (2020). “Estimating the asymptomatic proportion of coronavirus disease 2019 (COVID-19) cases on board the Diamond Princess cruise ship, Yokohama, Japan, 2020”. In : *Eurosurveillance* 25.10, p. 2000180.

MOHSIN, AH et al. (2018). “Real-time remote health monitoring systems using body sensor information and finger vein biometric verification : A multi-layer systematic review”. In : *J. Medical Syst.* 42.12, p. 238.

MSEDDI, A. et al. (2019). “Joint Container Placement and Task Provisioning in Dynamic Fog Computing”. In : *IEEE Internet of Things J.* 6.6, p. 10028-10040. DOI : 10.1109/jiot.2019.2935056.

MSEDDI, Amina et al. (2019). “Intelligent Resource Allocation in Dynamic Fog Computing Environments”. In : *Proc. IEEE Int. Conf. Cloud Network. (CloudNet)*, p. 1-7. DOI : 10.1109/CloudNet47604.2019.9064110.

O'NEILL, Graham et Neil S TOLLEY (2020). “Cochlear Implant Reliability : Reporting of Device Failures”. In : *Indian J. Otolaryngology and Head & Neck Surgery*, p. 1-3.

ORGANIZATION, World Health (2020). *Algorithm for COVID-19 triage and referral : Patient triage and referral for resource-limited settings during community transmission.*

PHAM, Minh et al. (2018). “Delivering home healthcare through a cloud-based smart home environment (CoSHE)”. In : *Future Gen. Comput. Syst.* 81, p. 129-140.

PIAO, Chunhui et al. (2019). “Privacy-preserving governmental data publishing : A fog-computing-based differential privacy approach”. In : *Future Gen. Comput. Syst.* 90, p. 158-174.

RAFFIN, Antonin et al. (2019). *Stable Baselines3*. https://github.com/DLR-RM/stable-baselines3.

RAJ, Abhi et Dan STEINGART (2018). “Power sources for the Internet of Things”. In : *J. Electrochemical Soc.* 165.8, B3130.

REMUZZI, Andrea et Giuseppe REMUZZI (2020). “COVID-19 and Italy : What next ?” In : *The Lancet*.

Ross, G Terry et Richard M Soland (1975). “A branch and bound algorithm for the generalized assignment problem”. In : *Mathematical Programming* 8.1, p. 91-103.

Rothan, Hussin A et Siddappa N Byrareddy (2020). “The epidemiology and pathogenesis of coronavirus disease (COVID-19) outbreak”. In : *J. of Autoimmunity*, p. 102433.

Sharma, Mudita et al. (2019). “Deep reinforcement learning based parameter control in differential evolution”. In : *Proc. Genetic Evolution. Comput. Conf. (GECCO)*, p. 709-717.

Swayamsiddha, Swati et Chandana Mohanty (2020). “Application of cognitive Internet of Medical Things for COVID-19 pandemic”. In : *Diabetes & Metabolic Syndrome : Clinical Research & Rev.* 14.5, p. 911-915.

Topol, Eric (2012). *The Creative Destruction of Medicine : How the Digital Revolution will Create Better Health Care*. Basic Books.

Vaquero, Luis M et Luis Rodero-Merino (2014). “Finding your way in the fog : Towards a comprehensive definition of fog computing”. In : *ACM SIGCOMM Comp. Commun. Review* 44.5, p. 27-32.

Watson, Andrew R, Robert Wah et Ritu Thamman (2020). “The value of remote monitoring for the COVID-19 pandemic”. In : *Telemedicine and e-Health* 26.9, p. 1110-1112.

Xiao, Yang (2005). “IEEE 802.11n : Enhancements for higher throughput in wireless LANs”. In : *IEEE Wireless Commun.* 12.6, p. 82-91.

YANNUZZI, Marcelo et al. (2014). “Key ingredients in an IoT recipe : Fog Computing, Cloud computing, and more Fog Computing”. In : *Proc. IEEE Int. Wrkshp. Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, p. 325-329.

YAO, Jingjing et Nirwan ANSARI (2018). “Reliability-aware fog resource provisioning for deadline-driven IoT services”. In : *2018 IEEE global communications conference (GLOBECOM)*. Ieee, p. 1-6.

– (2019). “Fog resource provisioning in reliability-aware IoT networks”. In : *IEEE Internet of Things Journal* 6.5, p. 8262-8269.

YI, Shanhe, Cheng LI et Qun LI (2015). “A survey of fog computing : concepts, applications and issues”. In : *Proc. Mob. Big Data Wrkshp. (MOBISYS)*, p. 37-42.

ANNEXE A

- Dans la sous-section 4.4.1.2 de ce chapitre, nous définissons, pour des raisons de simplicité, le délai de transmission sans fil entre un patient u et un nœud fog f de la manière suivante :

$$\delta^{r,u,y,f}_{\mathrm{com,u \longrightarrow f}} = \frac{\tilde{Q}^{r,u,y,f}}{B \cdot \log_2 \left(1 + \frac{\eta^u \cdot \|p^u - l^f\|^{-\beta}}{N_0}\right)}, \tag{8}$$

où B représente la bande passante du canal, η^u est la puissance d'émission du patient, N_0 est la puissance du bruit, $\beta \geq 2$ est l'exposant de l'affaiblissement du trajet, et $\| \cdot \|$ représente la norme euclidienne.

Cependant, cette formulation ne prend pas en compte des facteurs tels que l'affaiblissement de canal et les interférences, qui peuvent être inclus à la formule précédente de la manière suivante :

$$\delta^{r,u,y,f}_{\mathrm{com,u \longrightarrow f}} = \frac{\tilde{Q}^{r,u,y,f}}{B \cdot \log_2 \left(1 + \frac{\eta^u \cdot \mathcal{H} \cdot \|p^u - l^f\|^{-\beta}}{N_0 + \mathcal{I}}\right)}, \tag{9}$$

où $\mathcal{H}$ est une variable aléatoire représentant l'affaiblissement du canal (souvent supposée suivre une distribution de Rayleigh ou de Rice), et $\mathcal{I}$ représente la puissance des signaux interférents. Notez que dans un système de communication réel, ces quantités pourraient varier en fonction du temps et même en fonction de l'emplacement des interlocuteurs.

- Dans la section 4.4.2 de ce chapitre, nous explicitons l'objectif de la fonction. Celle-ci cherche à maximiser le nombre d'applications qui répondent aux exigences de disponibilité et de latence, comme indiqué ci-dessous :

$$\max_{\mathbf{W},\tilde{\mathbf{Q}}} \quad \sum_{u\in\mathcal{U}}\sum_{y\in\mathcal{Y}}(S^{u,y}+S_A^{u,y}) \text{ s.t.} \qquad (9)-(16). \tag{P1}$$

Il est crucial de remarquer que, bien que la maximisation de la fonction objectif dans la formulation relaxée ci-dessus permette d'atteindre les objectifs mentionnés, elle n'assure pas une adhérence absolue en tout temps aux exigences de disponibilité et de latence pour une application déterminée.

Pour pallier à cela, nous proposons une redéfinition de $S^{u,y}$ afin de signaler la satisfaction simultanée des exigences de délai et de disponibilité, comme illustré ci-après :

$$S^{u,y} = \begin{cases} 1, & \text{si } T^{u,y}-\delta^{u,y}\geq 0 \text{ et } \alpha^{u,y}-A^{u,y}\geq 0, \\ 0, & \text{autrement} \end{cases} \quad \forall u\in\mathcal{U}, y\in\mathcal{Y}. \tag{A.2}$$

Par la suite, nous recommandons l'élimination des contraintes (11) et (12) et la redéfinition de la fonction objectif de la manière suivante :

$$\max_{\mathbf{W},\tilde{\mathbf{Q}}} \quad \sum_{u\in\mathcal{U}}\sum_{y\in\mathcal{Y}}S^{u,y} \text{ s.t.} \qquad (9)-(16). \tag{P1}$$

ANNEXE B

Dans la sous-section 4.4.1.1 de ce chapitre, nous formulons la disponibilité. Cette annexe met en lumière l'interconnexion entre fiabilité et disponibilité.

La disponibilité, probabilité qu'un système soit opérationnel lorsqu'il est sollicité, est un critère de performance majeur, relié aux notions de fiabilité (durée de vie du système) et de maintenabilité (nombre prévu de maintenances).

B.1 Fiabilité

La fiabilité représente la probabilité qu'un produit ou un service fonctionne sans défaillance sur une période donnée.

Mathématiquement, la fiabilité est définie par une variable aléatoire $X \geq 0$, représentant le temps avant la défaillance d'un équipement. Ainsi, la fiabilité à un moment t, $R(t)$, est une fonction de répartition cumulative, définie par :

$$R(t) = P\{X > t\}, \quad t > 0 \tag{B.1}$$

Cette équation démontre la probabilité qu'un équipement fonctionne correctement après un moment t donné.

L'état de défaillance d'un équipement est aussi représenté par une fonction de répartition cumulative des défaillances $F(t)$, qui est mutuellement exclusive avec $R(t)$:

$$F(t) + R(t) = 1, \quad t > 0 \tag{B.2}$$

Cela permet de définir la densité de probabilité des défaillances $f(t)$ et de dériver les formules suivantes :

$$R(t) = 1 - F(t) = 1 - \int_0^t f(x)dx \tag{B.3}$$

$$\frac{dR(t)}{dt} = -f(t) \tag{B.4}$$

Ces équations mettent en avant le lien entre la fiabilité et le taux de défaillance d'un équipement.

B.2 Taux de pannes

Le taux de pannes $\lambda(t)$ donne la fréquence de pannes pour l'équipement considéré et un intervalle de temps $[t,\ t + \Delta t]$.

$$\lambda(t) = \frac{P\{t < X \leq t + \Delta t \mid X > t\}}{\Delta t} \tag{B.5}$$

$$\lambda(t) = \frac{P\{t < X \leq t + \Delta t\}}{\Delta t \ \cdot \ P\{X > t\}} \tag{B.6}$$

$$\lambda(t) = \frac{1}{\Delta t \ \cdot \ R(t)} \int_t^{t+\Delta t} f(t)\ dt \tag{B.7}$$

$$\lambda(t) = \frac{R(t) - R(t + \Delta t)}{\Delta t\ R(t)} \tag{B.8}$$

Le taux de panne instantané, à un moment précis t, est alors déduit :

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{R(t) - R(t+\Delta t)}{\Delta t \; R(t)} = \frac{1}{R(t)} \left[-\frac{dR(t)}{dt} \right] \tag{B.9}$$

Ceci nous permet d'établir les relations entre la fiabilité, le taux de pannes et la probabilité de pannes :

$$\lambda(t) = \frac{f(t)}{R(t)} = \frac{1}{R(t)} \left[\frac{dF(t)}{dt} \right] \tag{B.10}$$

$$R(t) = e^{-\int_0^t \lambda(x) \; dx} = 1 - \int_0^t \lambda(x) \; dx \tag{B.11}$$

B.3 Maintenabilité

La maintenabilité se réfère à la capacité de maintenir l'opérabilité ou de restaurer un équipement à son état opérationnel en conformité avec les conditions spécifiées de fonctionnement et dans un intervalle de temps défini.

Parallèlement à la fiabilité, considérons une variable aléatoire $Y \geq 0$ qui représente le temps requis pour réparer un équipement ou un service. Nous définissons alors la fonction de répartition cumulative du temps de réparation $M(t)$ comme suit :

$$M(t) = P\{Y \leq t\}, \quad t > 0 \tag{B.12}$$

En nous basant sur la densité de probabilité du temps de réparation $m(t)$ d'un équipement dans un intervalle de temps $[t, \; t + \Delta t]$, nous pouvons dériver les relations suivantes :

$$M(t) = \int_0^t m(x)dx \tag{B.13}$$

$$\int_{t}^{t+\Delta t} m(x)dx = M(t+\Delta t) - M(t) \tag{B.14}$$

Le taux de réparations $\mu(t)$ peut être défini comme étant la probabilité qu'un équipement ou un service, ayant subi une défaillance à partir de l'instant t, soit réparé avant $t + \Delta t$:

$$\mu(t) = \lim_{\Delta t \to 0} \frac{P\{t < Y \leq t + \Delta t \mid Y > t\}}{\Delta t} = \frac{m(t)}{1 - M(t)} \tag{B.15}$$

B.4 Modélisation de la disponibilité

La préservation de la disponibilité d'un équipement se décompose en deux aspects distincts : Le premier aspect vise à prévenir la survenance d'une panne pour une durée aussi longue que possible. Le second aspect est axé sur la réalisation d'une réparation aussi rapidement que possible en cas de panne.

Une approche couramment adoptée pour modéliser la disponibilité est le processus de renouvellement ELSAYED, 1996. De manière succincte, un processus de renouvellement vise à caractériser la fréquence d'apparition d'un phénomène (par exemple, une défaillance) au sein d'un système, et peut être décrit comme une alternance de séquences d'états (par exemple, état de défaillance et état de fonctionnement).

Formellement, un processus de renouvellement est défini comme une suite infinie de variables aléatoires positives, indépendantes et identiquement distribuées, où le système est supposé être réparé à neuf (renouvelé) entre chaque séquence.

Dans le contexte de la disponibilité, nous définissons T_i comme la i^{me} période de fonctionnement et D_i comme la i^{me} période de défaillance, pendant laquelle des opérations de réparation sont menées en parallèle. Pour chaque période $i = \{1, 2..., t\}$, nous avons une séquence de variables aléatoires $X_i = T_i + D_i$.

Considérons maintenant T_i avec $i = \{1, 2, ..., t\}$, une variable aléatoire positive, indépendante et identiquement distribuée avec une fonction de répartition cumulative $W(t)$ et une densité de probabilité $w(t)$. De manière analogue, nous définissons pour $D_i \mid i = \{1, 2, ..., t\}$, une variable aléatoire positive, indépendante et identiquement distribuée avec une fonction de répartition cumulative $G(t)$ et une densité de probabilité $g(t)$. Par transitivité, X_i avec $i = \{1, 2, ..., t\}$ est également positive, indépendante et identiquement distribuée. Étant donné que X_i avec $i = \{1, 2, ..., t\}$ est la somme de deux variables aléatoires indépendantes, nous avons, par le théorème de la convolution, que $f(t)$, la densité de probabilité associée à X_i, est la convolution de w et g, telle que :

$$\hat{f}(s) = \hat{w}(s)\ \hat{g}(s) \tag{B.16}$$

Il est alors possible de catégoriser l'état de fonctionnement d'un équipement en deux cas de figure Elsayed, 1996 :

- Dans le premier cas, l'équipement n'a jamais connu de défaillance dans l'intervalle $[0, t]$, ce qui correspond alors à la définition de la fiabilité $R(t)$.
- Le second cas correspond à celui où l'équipement est de nouveau opérationnel après avoir été réparé à un moment $x \mid 0 < x < t$. La disponibilité est alors équivalente à la probabilité $\int_0^t R(t-x)\ m(x)\ dx$.

En combinant ces deux probabilités, la disponibilité est alors définie par la relation suivante Elsayed, 1996 :

$$A(t) = R(t) + \int_0^t R(t-x)\ m(x)\ dx \tag{B.17}$$

où $m(x)$ représente la fonction de densité de probabilité du processus de renouvellement du système.

Cette équation peut être résolue en utilisant la transformation de Laplace Elsayed, 1996 :

$$\hat{A}(s) = \hat{R}(s)\ [1 + \hat{m}(s)] \tag{B.18}$$

$$\hat{A}(s) = \frac{\hat{R}(s)}{1 - \hat{w}(s)\hat{g}(s)} \tag{B.19}$$

où

$$\hat{m}(s) = \frac{\hat{w}(s)\hat{g}(s)}{1 - \hat{w}(s)\hat{g}(s)} \tag{B.20}$$

Rappelons que $W(t)$ représente la fonction de répartition cumulative qui dénote les périodes d'états de fonctionnement du système. Nous avons donc par définition :

$$R(t) = 1 - W(t) \tag{B.21}$$

En appliquant la transformation de Laplace, nous obtenons alors :

$$\hat{R}(s) = \frac{1 - \hat{w}(s)}{s} \tag{B.22}$$

En réécrivant l'équation B.19 en prenant en compte l'équation B.22, nous obtenons :

$$\hat{A}(s) = \frac{1 - \hat{w}(s)}{s[1 - \hat{w}(s)\hat{g}(s)]} \tag{B.23}$$

Nous aurions obtenu la disponibilité instantanée $A(t)$ en effectuant une transformation de Laplace inverse. Cependant, une expression analytique de la transformation de Laplace inverse est difficile à obtenir.

Dans ce contexte, on définit la disponibilité asymptotique comme étant la probabilité qu'un système soit disponible en considérant une période de temps extrêmement longue Elsayed, 1996. Le principal avantage de la disponibilité asymptotique réside dans le fait qu'elle permet d'obtenir une solution analytique avec aisance, ce qui explique pourquoi elle est fréquemment utilisée dans l'analyse de la disponibilité de nombreux systèmes.

En effet, à partir de l'équation B.23, nous pouvons obtenir la disponibilité asymptotique A_s en posant :

$$A_s = \lim_{t \to \infty} A(t) \tag{B.24}$$

En transposant dans le domaine laplacien et en appliquant le théorème de la valeur finale, nous obtenons :

$$A_s = \lim_{s \to 0} s\hat{A}(s) \tag{B.25}$$

Selon la définition de la transformation de Laplace, nous avons :

$$\hat{w}(s) = \int_0^\infty e^{-st}\ w(t)\ dt \tag{B.26}$$

En considérant que s est très petit (ou, en d'autres termes, que t est très grand), alors nous avons $e^{-st} \cong 1 - st$, ce qui donne :

$$\hat{w}(s) = \int_0^\infty (1 - st)\ w(t)\ dt \tag{B.27}$$

Si nous considérons une distribution exponentielle pour $w(t)$ et $g(t)$, nous obtenons alors :

$$\hat{w}(s) = 1 - \frac{s}{\lambda} \tag{B.28}$$

où λ représente le taux de défaillance du système. De manière similaire, nous avons :

$$\hat{g}(s) = 1 - \frac{s}{\mu} \tag{B.29}$$

où μ représente le taux de réparation du système.

Dès lors, nous pouvons reformuler la disponibilité asymptotique de la manière suivante :

$$A_s = \frac{\frac{1}{\lambda}}{\frac{1}{\lambda} + \frac{1}{\mu}} \tag{B.30}$$

$$A_s = \frac{\mu}{\lambda + \mu} \tag{B.31}$$

En notant $\frac{1}{\lambda}$ par le temps moyen entre les défaillances (MTTF pour *mean time to failure*) et $\frac{1}{\mu}$ par le temps moyen de réparation (MTTR pour *mean time to repair*), nous retrouvons la formulation habituelle de la disponibilité asymptotique :

$$A_s = \frac{MTTF}{MTTF + MTTR} \tag{B.32}$$

Cette équation indique clairement que la disponibilité asymptotique d'un système est directement proportionnelle au temps moyen entre les défaillances et inversement proportionnelle à la somme du temps moyen entre les défaillances et du temps moyen de réparation. Cela reflète le fait que plus un système fonctionne longtemps sans défaillance et plus les réparations sont rapides, plus la disponibilité du système est élevée.

CHAPITRE III

ARTICLE 3 - DRL-BASED GREEN RESOURCE PROVISIONING FOR 5G NETWORKS

Mouhamad Dieye
Université du Québec à Montréal
Montréal (Québec), Canada

Wael Jaafar, Professeur
École de Technologie Supérieure
Montréal (Québec), Canada

Halima Elbiaze, Professeur titulaire
Université du Québec à Montréal
Montréal (Québec), Canada

Roch Glitho, Professeur titulaire
Concordia University
Montréal (Québec), Canada

AVANT-PROPOS À L'ARTICLE 3

Le troisième chapitre de cette thèse se concentre sur l'allocation à grande échelle de ressources virtuelles dans les réseaux mobiles et fixes, en tenant compte des contraintes de qualité de service. Ce chapitre porte une attention particulière à la capacité de généralisation des algorithmes d'apprentissage dans un environnement de réseau caractérisé par une dynamique et une hétérogénéité notables.

Dans le cadre spécifique du DRL, la généralisation est définie comme la capacité d'un agent à s'adapter à des situations, des états ou des tâches inédits, en se fondant sur ses expériences d'apprentissage antérieures. L'objectif est d'élaborer des politiques efficaces non seulement pour les états spécifiques rencontrés lors de l'entraînement, mais également pour une gamme plus large d'états qui n'ont pas été directement expérimentés.

Cette capacité revêt une importance cruciale dans les environnements à grands espaces d'états ou à espaces d'états continus, où il est irréaliste ou impossible d'apprendre une politique pour chaque état individuel. L'agent doit être capable de généraliser à partir de ses expériences pour prendre des décisions éclairées dans de nouvelles situations ou des conditions légèrement modifiées.

Néanmoins, obtenir une bonne généralisation est notoirement difficile, particulièrement dans le domaine du DRL. Cela est dû à plusieurs défis tels que le surapprentissage, où un agent apprend une politique qui est trop spécialisée dans les particularités de son environnement d'entraînement et qui performe mal dans des paramètres légèrement modifiés. Un autre obstacle est le "décalage de distribution", où l'expérience de l'agent lors de l'entraînement peut ne pas couvrir de manière adéquate la gamme de situations qu'il pourrait rencontrer dans des conditions réelles, rendant difficile pour l'agent de généraliser efficacement.

Cette difficulté est exacerbée dans des environnements aux dynamiques complexes comme les réseaux, particulièrement ceux sans fil en raison de leur nature hautement dynamique, stochastique et multidimensionnelle. Les défis dans la conception de la fonction de récompense et leur impact conséquent sur la généralisation peuvent être résumés comme suit :

- **Dynamiques complexes :** Dans les réseaux sans fil, de nombreux facteurs tels que l'affaiblissement, l'interférence, la demande de trafic, la mobilité des utilisateurs, et d'autres, contribuent à l'état du réseau à un instant donné. Ces facteurs peuvent changer rapidement et de manière imprévisible, ce qui donne lieu à des environnements non stationnaires qui rendent le processus d'apprentissage ardu. L'agent doit non seulement apprendre une politique qui fonctionne efficacement dans les conditions actuelles, mais aussi une politique qui se généralise bien aux conditions futures qui pourraient être radicalement différentes.

- **Récompenses différées :** Dans de nombreux scénarios, les conséquences des actions pourraient ne pas être immédiates. Par exemple, une décision d'attribuer un canal spécifique à un utilisateur pourrait entraîner une interférence qui n'affecte la qualité du service que plus tard. Ce phénomène introduit le problème d'attribution de crédit temporel, où il devient difficile d'attribuer la récompense finale (ou la pénalité) aux actions spécifiques qui l'ont causée. Cela complique l'apprentissage d'une politique optimale et sa généralisation à de nouvelles situations.

- **Récompenses multi-objectifs :** Souvent, il y a plusieurs objectifs à considérer dans les réseaux sans fil, tels que maximiser le débit, minimiser le délai, maintenir l'équité entre les utilisateurs, etc. Concevoir une fonction de récompense qui équilibre ces objectifs n'est pas trivial, et de petits changements dans la conception de la récompense peuvent conduire à des politiques apprises radicalement différentes. De plus, une politique qui fonctionne bien sous une fonction de récompense donnée peut ne pas bien se généraliser à des situations où les objectifs ou leur importance relative changent.

- **Grand espace d'états-actions :** L'espace d'états-actions dans les réseaux sans fil peut être extrêmement vaste en raison du grand nombre de paramètres et de variables impliqués. Cela augmente le risque de surapprentissage, où l'agent apprend une politique qui fonctionne bien sur les états spécifiques et les actions qu'il a expérimentées, mais échoue à se généraliser à de nouveaux.
- **Politique optimale inconnue :** Dans des environnements complexes tels que les réseaux sans fil, il est souvent difficile de savoir à quoi ressemble même la politique optimale. Cela rend difficile la guidance du processus d'apprentissage et l'évaluation de la généralisation de la politique apprise.

L'article précédent emploie DRL en complément à DE afin (1) d'optimiser les décisions de placement de tâches et d'approvisionnement en ressources dans un environnement instable via une sélection améliorée de la stratégie de mutation, et (2) d'accélérer le temps de convergence. Cependant, la nécessité de'effectuer une suringénierie de la modélisation de l'agent DRL pour obtenir une haute performance d'apprentissage a été notée comme le principal inconvénient, car étant un obstacle à la généralisation de l'apprentissage. En réalité, l'article s'appuie fortement sur la bonne connaissance de l'environnement pour guider l'apprentissage de l'agent, ce qui devient peu pratique lorsque l'environnement subit des changements fréquents et importants. Le présent article intègre des considérations environnementales, devenues des composantes essentielles dans la gestion des réseaux modernes. Il continue également l'étude de la viabilité du DRL comme stratégie d'allocation des ressources dans les réseaux dynamiques à grande échelle.

Dans le présent article, nous recourons à des algorithmes parallélisables en vue d'atténuer la complexité des dynamiques environnementales. Cette stratégie nous permet de découvrir un mécanisme de récompense exhaustif, favorisant ainsi une meilleure généralisation de l'apprentissage.

Cet article a été publié dans la revue **IEEE Transactions on Green Communications and Networking (IEEE TGCN), Early Access, July 2023**. IEEE TGCN est classé Q1 dans la catégorie "Computer Networks and Communications" (SJR 2022) avec un facteur d'impact de 4.82 (H-Index : 38).

- Dieye, M., Jaafar, W., Elbiaze, H., & Glitho, R. H. (2023). **DRL-based Green Resource Provisioning for 5G and Beyond Networks.** *Early Access.*
  DOI: 10.1109/TGCN.2023.3296646

ABSTRACT

Networks are indisputably central to daily human activities, as our reliance on their capacity to efficiently deliver stringent multimedia services for applications such as remote health monitoring and remote education grows.

Therefore, network operators are tasked with supporting innovative services through efficient resource provisioning, while ensuring the satisfaction of profit maximization and environmental goals. Machine learning-based (particularly deep reinforcement learning (DRL)-based) resource allocation strategies are gaining popularity due to their ability to make near-optimal decisions directly from observed data. The environment inherently influences the performance of DRL agents.

Consequently, the reward design mechanism of DRL agents is considerably more complex for the resource provisioning problem in large-scale, dynamic environments such as wireless networks, where multi-objective optimization constraints are almost always present. In order to avoid intractability, reduce computational complexity, and improve the efficacy of DRL-based approaches to resource provisioning with environmentally-friendly goals, this study investigates the feasibility of employing parallelization approaches to generate network environments that enable efficient and generalizable learning for DRL algorithms. We propose two notable solutions, DRL-MCTS, and DRL-VNF, for the wireless resource allocation and wired network virtual network function (VNF) service provisioning problems, respectively. We optimize the search space of associated DRL algorithms using Monte Carlo tree search (MCTS) and Floyd-Warshall algorithms for admission control and path assessment.

Through experimental evaluations, we demonstrate the efficacy of the proposed approaches in reducing the network's environmental footprint while meeting end-users QoS requirements. In addition, our findings reveal model drift issues with DRL agents, which are discussed.

Keywords: Service Function Chaining, virtual network functions, VNF, network slicing, wireless resource allocation, carbon footprint reduction, deep reinforcement learning, resource provisioning, energy-efficient networks, admission control, QoS.

3.1 Introduction

Recent years have seen networks cope with unparalleled data traffic growth stemming from the advent of innovative multimedia services, such as HD video streaming and augmented reality, besides the rapid development of Internet of things (IoT) applications. However, due to resource scarcity and the current inefficiency in network and service management, critical requirements of future services may not be achieved by current infrastructures (Zhani et ElBakoury, 2019). To avoid such concerns, 5G (and beyond) networks are expected to provide, through several enabling technologies, higher data rates, improved spectrum and energy consumption, and reduced end-to-end latency (Hossain et M. Hasan, 2015 ; Zhu et Hossain, 2016 ; Andrews et al., 2014).

Despite substantial investments to enhance network capacity, particularly for mobile networks, revenues are still below expectations due to fierce competition from third-party operators (Kiiski, 2006). This has led to the emergence of two main stakeholders : (1) mobile network operators (MNOs) as owners of physical network resources such as base stations and antennas that are virtualized into slices and leased to different mobile virtual network operators (MVNOs). Hence allowing several MVNOs to coexist under the same MNO and contribute towards its expenditure savings. (2) MVNOs that leverage virtual resources to offer various services, attract more customers, and generate profits (Kiiski, 2006).

Several works in the literature have investigated problems related to virtualization, particularly network resource slicing ((Zhu et Hossain, 2016 ; Jiang et al., 2016) and references therein).

Nonetheless, due to the diverse nature of QoS requirements and network resource scarcity, the key challenge is to efficiently allocate these resources to slices while maintaining the necessary flexibility to allow enhanced multimedia services creation and deployment (JIANG et al., 2016; ZHU et HOSSAIN, 2016).

Additionally, MVNOs increasingly seek active involvement in resource allocation to guarantee their subscribers' varying requirements and maximize their revenues (ZHU et HOSSAIN, 2016; JIANG et al., 2016). Indeed, proper assessment of the required priority and QoS requirements of user requests for management purposes, such as scheduling decisions, can be a tedious task for MNOs (FU et KOZAT, 2013; ZHU et HOSSAIN, 2016), given the incurring high computational complexity and hindered customization.

Consequently, MVNOs must determine their own needs (e.g., data rates, latency, etc.), emit resource requests, and subsequently apply custom resource scheduling policies on the slices allocated by the MNO. However, in mobile networks where user mobility must also be handled, the achieved data rates by slices allocated to MVNOs vary depending on the physical channel quality between subscribers and MNOs' base stations (BS). This puts MNOs at risk of resource under-utilization or added costs to meet MVNOs' QoS requirements.

In other words, MNOs must determine how to allocate their limited resources to competing MVNOs (also individually seeking to maximize their profits) while considering profitability and service priority.

In addition to the above challenges, environmental-friendly network service provisioning has become an essential aspect of future networks due to the significant effect that carbon footprint has on firm-value (MATSUMURA et al., 2014). As such, substantial efforts toward greener mobile network infrastructure and service provisioning have been observed in recent years, driven primarily by government restrictions and marketing concerns (AMOKRANE et al., 2015). In this respect, MNOs are mandated to reduce their carbon footprint through energy-efficient service provisioning approaches and the increased deployment of renewable energy sources within the network infrastructures (Amokrane et al., 2015).

Recent works have, for instance, advocated for defining green service license agreements (SLAs) (Amokrane et al., 2015), while others investigated joint energy source selection and power control to optimize consumed network energy (Han et Ansari, 2013 ; Han et Ansari, 2014). However, these proposals did not investigate the virtual networking function allocation and provisioning issues.

This paper aims to address the resource provisioning problem of enhanced multimedia services through VNF deployment while considering QoS, profitability, and energy-efficiency stakes. Notably, our resource provisioning decision-making process considers network devices' energy consumption and carbon footprint.

In large-scale networks, operators typically rely on various heuristic algorithms to provide sub-optimal solutions for the resource provisioning problem mainly due to the exponential computational complexity growth with network scale (Blenk et al., 2016 ; Boutaba et al., 2018). As a result, machine learning (ML)-driven solutions have gained traction as an alternative or complement for network-related combinatorial problems ((Boutaba et al., 2018 ; Bengio et al., 2018) and references therein).

In particular, reinforcement learning (RL)-based solutions have gained substantial momentum in the literature due to the ability of RL agents to autonomously interact with their environment and eventually acquire optimal behaviors over time. Nevertheless, the majority of existing work suffers from practical limitations as they either : 1) involve inherently atomic agents, i.e., they perform a simple periodic skill and rarely involve complex multi-level reasoning or constraints (Nachum et al., 2018), or 2) are hampered by their lack of generalization, since they require manual task-specific design, in addition to costly on-policy training that is unable to benefit from advances in off-policy RL, in terms of drastically reducing sample complexity requirements (Nachum et al., 2018).

In fact, the network resource provisioning problem is almost always multi-objective, particularly for wireless networks. Moreover, because of the fluctuations in network dynamics (e.g., frequent virtual and physical topological changes) and user request characteristics, the generalization and scalability of RL algorithms are critical and understated factors. In this work, we investigate the potential of parallelizable heuristics that can provide an environment conducive to efficient and generalizable learning.

To our knowledge, this is the first work to jointly investigate these issues using ML-based methods. We propose two parallelizable RL-based methods, called DRL-MCTS and DRL-VNF, to respectively solve, from the MNO's perspective, the resource provisioning problem both in wireless and wired large-scale networks. For readability purposes, the MVNO's resource provisioning problem is out of the scope of this paper.

The main contributions are summarized as follows :

1. We model the MNO's green resource allocation problem for wireless networks considering massive multiple-input multiple-output (MIMO) base stations and imperfect channel state information (CSI) acquisition. End-users' data rate requirements, resource utilization, and profit considerations are also taken into account in the system model.

2. We also formulate the MNO's green VNF resource provisioning problem in large-scale wired networks. We jointly account for the resource provisioning-induced environmental impacts, end-users QoS requirements, and resource utilization.

3. Two RL-driven solutions are proposed to solve the aforementioned green wireless and VNF resource provisioning problems. The developed solutions are designed to be parallelizable for large-scale systems.

4. Through realistic experimental evaluations, we illustrate the efficiency of the proposed approaches in reducing the network's environmental footprint while satisfying end-users QoS requirements.

The rest of the paper is organized as follows. Section II provides a summary of the related work. Section III presents the system model. Section IV provides a mathematical formulation of the enhanced multimedia services provisioning problem in 5G networks. We detail our proposed algorithms in Section V, whereas Section VI presents the experimental evaluations and results. Section VII discusses our approaches' limitations, and finally, we conclude the paper in Section VIII.

3.2 Related Work

This section highlights recent research utilizing ML-driven approaches to solve the network resource provisioning problem. Network resource provisioning is a decision problem that entails managing network resources to maximize objectives such as resource utilization or revenue. With network function virtualization (NFV), another dimension of complexity is introduced, given the need to embed a set of virtual network resources on top of physical resources to provide innovative services to end-users (Boutaba et al., 2018).

Furthermore, guaranteeing proper satisfaction of end-users QoS requirements and operator profit maximization entails ensuring that virtual network resources are dynamically and optimally provisioned, as service demands vary over time. A two-step process is generally followed to do so, which involves admission control and resource allocation/migration decisions. In admission control, the challenge is to accept or deny incoming service requests based on multiple criteria, such as the estimated profits, the presently available resources, the QoS requirements of the request, and its estimated impact on existing services deployed within the network. Once a service request is accepted, we focus on the resource allocation step, where the adequate locations of NFV service components are determined (Boutaba et al., 2018).

3.2.1 ML-driven Admission Control

Techniques employing machine learning for admission control have been extensively studied, especially for wireless and mobile networks (see (Boutaba et al., 2018) and references therein).

The majority of research has examined admission control using the supervised learning paradigm (Piamrat et al., 2008; Blenk et al., 2016). For instance, in (Piamrat et al., 2008), the authors advocated for an admission control mechanism based on the Quality-of-Experience (QoE) perceived by end-users who submit an average opinion score. In turn, the mean opinion scores are utilized to build the training and testing datasets of a random neural network (RandNN), which generates a real-time prediction of mean opinion scores.

Blenk *et al.* (Blenk et al., 2016) explored the online virtual network embedding problem with a recurrent neural network (RNN) for admission control. The RNN is trained based on the substrate network topological and resource parameters, including the number of nodes, load and the average number of node degrees, and virtual request characteristics. The RNN then predicts whether a request will be granted or rejected based on the incoming request's requirements and the existing state of the substrate network.

In contrast to the previous work, Ahn *et al.* (Ahn et Ramakrishna, 2004) proposed an unsupervised learning-based approach that uses a Hopfield neural network to grant admission to an incoming request only when the bandwidth of corresponding wireless cells is sufficient to meet the request's bandwidth requirements. Notably, their method does not necessitate training and can quickly adapt to dynamic network conditions.

In the literature, RL-based admission control systems have also been examined. Several authors employed Q-learning for admission control (Mignanti et al., 2009 ; Wang et Y. Qiu, 2013 ; Tong et Brown, 2000), while others leveraged temporal difference-based RL (Marbach et al., 2000). For instance, Mignanti *et al.* used in (Mignanti et al., 2009) Q-learning, where Q-values are computed to approve or deny incoming requests. In addition, they used Q-learning to derive the optimal policy for guard channel allocation to minimize the cumulative blocking probability.

Marbach *et al.* (Marbach et al., 2000) proposed an approximation design in which the NN weights are tuned using tabular temporal difference. To avoid the latter's slow convergence rate, they utilized a novel approach in which the network is fragmented into a set of link processes to reduce training time.

3.2.2 ML-driven Resource Allocation

In a dynamic network environment where demands can fluctuate unpredictably over time, ML-driven algorithms benefit from their capability to learn pertinent indicators and policies for optimal resource allocation. In this domain, RL-based solutions have received significant attention in the literature since they can be deployed without any initial policy and can learn and adapt through interactions with the environment.

In this context, the authors of (Tesauro et al., 2005) employed decompositional RL to dynamically assign server resources for distinct workload types. Since the impact of the resource allocation choice is Markovian, they argue that it can be solved using a Markov Decision Process (MDP)-based formulation.

Given that the state and action space of an MDP expands exponentially, they proposed a decompositional RL formula for MDPs that leverages state-action - reward-state-action (SARSA) to learn the local value function based on the request's local state and resource allocation. Meanwhile, authors in (Pietrabissa et al., 2017) advocated for a scalable RL-based resource allocation solution that uses the policy reduction mechanism and state aggregation to reduce the state dimension.

Other works have investigated the virtual resources allocation problem. Mijumbi *et al.* introduced in (Mijumbi et al., 2014) a distributed RL-based resource management scheme that dynamically assigns virtual nodes and links. They modeled a substrate network where multiple agents used Q-learning on each physical node and link to learn the optimal policy for virtual-to-substrate resource allocation.

Additionally, they introduced a biased initialization phase in which every potential state-action value is computed and provided via a Q-table, further improving the Q-learning convergence rate. Similarly, authors of (Dutreilh et al., 2011) presented an RL-based resource allocation algorithm in which statistical estimators were drawn from measurements to accelerate the convergence rate.

In (Miyazawa et al., 2017), Miyazawa *et al.* investigated the dynamic resource reallocation mechanism for virtual networks, particularly in cases of urgent contingency. They proposed an RL-based resource migration strategy to determine alternate resources when QoS requirements are no longer met. Finally, Li *et al.* (R. Li et al., 2018) proposed a deep reinforcement learning (DRL) approach for network slicing resource management. They analyzed several use cases, including radio resource slicing and priority-based slicing of core network resources.

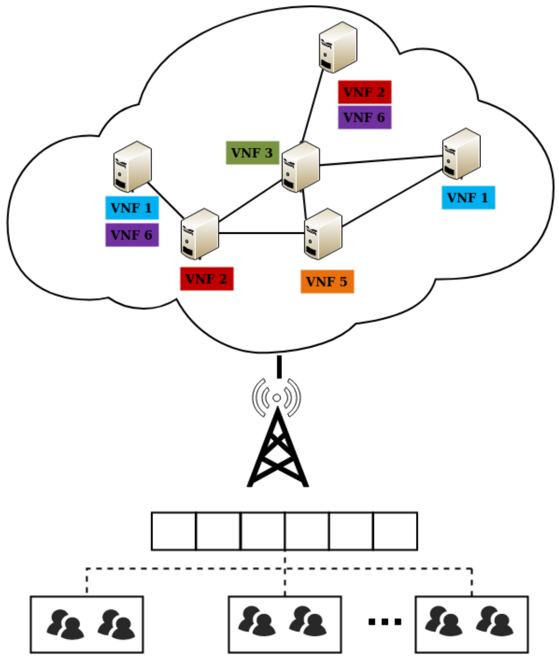

FIGURE 3.1 System model

3.3 System Model

We assume an MNO that owns many BSs and provides cellular coverage and resources to several competing MVNOs, as depicted in Fig.3.1. For the sake of simplicity, the MNO is also assumed to own the NFV infrastructure (NFVI). MVNOs provide various multimedia services to their subscribers by autonomously subdividing allocated virtual slices. These services are represented by VNF chains (i.e., sequences of VNFs) running on top of the NFVI, as seen in Fig. 3.1.

Let time be discretized as $\mathcal{T} = \{1, 2, ..., T\}$, with period $t \in \mathcal{T}$ corresponding to the total uplink frame time and T corresponding to total observation time.

Then, to ensure the proper delivery of services to its subscribers, each MVNO sends the MNO, resource requests as VNF chains with QoS requirements such as minimum data rate and budget offer, at period t.

Once received, the MNO determines through an admission control process whether the request for network resources can be accepted, depending on several factors, such as available resources, stringent QoS requirements, power and spectrum limitations, or simply low profitability to the network.

There are multiple stages of negotiation in the admission control process where each MVNO can either retain or change their offer.

An MVNO can lower the likelihood of rejection by boosting its budget, relaxing its QoS requirements, or postponing service for part of its subscribers in exchange for a penalty. When a request is considered appropriate, the MNO deploys the necessary VNFs while keeping energy efficiency and placement costs in mind.

3.4 Problem formulation

This section describes the MNO's wireless resource allocation and VNF provisioning problems. For wireless resource allocation, the MNO must ensure that its resources, such as BS antennas and frequency band subcarriers, are optimally assigned to users so as to maximize its wireless communication profit. In the VNF provisioning process, the MNO has to achieve the best VNFs service placement policy within its core network to reduce deployment costs and meet users' QoS requirements.

3.4.1 MNO's Green Wireless Resource Allocation Problem

We consider here the uplink of an orthogonal frequency division multiple access (OFDMA) cellular network. We assume a massive MIMO BS $\mathcal{B}$ with A antennas connected via backhaul to a core network $\mathcal{G}$ where enhanced multimedia services are deployed. The BS provides cellular coverage for the users of a set of MVNOs

$\mathcal{M} = \{1, 2, ..., M\}$. Each MVNO $m \in \mathcal{M}$ hosts at time t a set of slices $\mathcal{S}_m^t = \{S_{m,1}, S_{m,2}, \cdots, S_{m,|\mathcal{S}_m^t|}\}$ to which users are subscribed. Let $\mathcal{K}$ be the set of all users, such that $|\mathcal{K}| \ll A$. We express servicing user k by the slice s of MVNO m at instant t by the binary variable $\alpha_{s,m}^{k,t}$, such that

$$\alpha_{s,m}^{k,t} = \begin{cases} 1, & \text{if } k \in \mathcal{K} \text{ is subscribed to } s \in \mathcal{S}_m^t, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

We further consider that each user may have at time t a service request requirement, for instance $\sigma^{k,t} = \{u^{k,t}, g^{k,t}\}$, with $u^{k,t}$ denoting the minimum required data rate (bps/Hz) and $g^{k,t}$ the unitary price that can be paid to the MNO to access the resources. Let $\mathcal{C}^t = \{1, 2, ..., |\mathcal{C}^t|\}$ be the set of subcarriers available at the BS. The assignment of subcarrier $c \in \mathcal{C}^t$ to user k at time t can be defined by the binary variable $\beta_c^{k,t}$ where

$$\beta_c^{k,t} = \begin{cases} 1, & \text{if } c \in \mathcal{C}^t \text{ is assigned to } k \in \mathcal{K}, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

To reduce intra-cell interference during transmission, the exclusive OFDMA subcarrier assignment is constrained by the following condition :

$$\sum_{m \in \mathcal{M}} \sum_{s \in \mathcal{S}_m^t} \sum_{k \in \mathcal{K}} \alpha_{s,m}^{k,t} \cdot \beta_c^{k,t} \leq 1, \quad \forall c \in \mathcal{C}^t. \tag{3}$$

Let $a^{k,t} \in [0, A]$ be the number of the BS's antennas used to receive the signal of user k at time t. Hence, the new variable $\gamma_{s,m}^{k,c,t} = a^{k,t} \cdot \alpha_{s,m}^{k,t} \cdot \beta_c^{k,t}$ denotes the number of antennas assigned to user $k \in \mathcal{K}$ subscribed to slice $s \in \mathcal{S}_m$ used on subcarrier $c \in \mathcal{C}^t$. Consequently, the BS's antenna assignment needs to satisfy the following constraint in order to avoid antennas over-allocation :

$$\sum_{m \in \mathcal{M}} \sum_{s \in \mathcal{S}_m^t} \sum_{k \in \mathcal{K}} \sum_{c \in \mathcal{C}^t} \gamma_{s,m}^{k,c,t} \leq A. \tag{4}$$

To enjoy the full benefits of massive MIMO, an exact CSI is required. However, due to interference, this assumption is impractical (Tho Le-Ngoc et Derakhshani, 2017).

Hence, we assume in our model that CSI is imperfect, where the impact of non-orthogonal pilots are considered in signal reception. Thus, similarly to (Tho Le-Ngoc et Derakhshani, 2017), we denote by $\tilde{\mathbf{h}}_c^{k,t} \in \mathbb{C}^{1\times a^{k,t}}$ the estimated channel vector between user k and the BS at time t, by $\mathbf{f}_c^{k,t} \in \mathbb{C}^{a^{k,t}\times 1}$ the precoding vector for user $k \in \mathcal{K}$, subscribed to slice s of MVNO m, on subcarrier c at time t, and by $\mathbf{z}_c^{k,t} \in \mathbb{C}^{1\times a^{k,t}}$ the additive white Gaussian noise (AWGN) with zero mean and variance fixed to 1 for simplicity. Then, the received signal at the BS from user $k \in \mathcal{K}$ can be expressed by

$$\begin{aligned} y_{s,m}^{k,c,t} &= \alpha_{s,m}^{k,t}\beta_c^{k,t}\left[\tilde{\mathbf{h}}_c^{k,t}\mathbf{f}_c^{k,t}\sqrt{P_c^{k,t}}\varkappa_c^{k,t} - \mathbf{e}_c^{k,t}\mathbf{f}_c^{k,t}\sqrt{P_c^{k,t}}\varkappa_c^{k} + \mathbf{z}_c^{k,t}\mathbf{f}_c^{k,t}\right] \\ &+ \sum_{m'\in\mathcal{M}}\sum_{s'\in\mathcal{S}_{m'}^t}\sum_{\substack{k'\in\mathcal{K}\\k'\neq k}}\alpha_{s',m'}^{k',t}\beta_c^{k',t}\tilde{\mathbf{h}}_c^{k',t}\mathbf{f}_c^{k',t}\sqrt{P_c^{k',t}}\varkappa_c^{k',t} \end{aligned} \tag{5}$$

where $P_c^{k,t}$ and $\varkappa_c^{k,t}$ are the transmit power and transmitted signal of user k on subcarrier c at time t respectively, and $\mathbf{e}_c^{k,t} = \tilde{\mathbf{h}}_c^{k,t} - \mathbf{h}_c^{k,t}$ is the error in the CSI, with $\mathbf{h}_c^{k,t} = \left[h_{c,i}^{k,t}\right]_{1\times a^{k,t}}$ being the real channel vector at time t, channel coefficient $h_{c,i}^{k,t} = \chi_{c,i}^{k,t}\sqrt{w_c^{k,t}}$, $\chi_{c,i}^{k,t}$ captures the small-scale fading effect between user k and the i^{th} antenna among BS's $a^{k,t}$ antennas assigned, and $w_c^{k,t}$ is the large-scale link power attenuation due to path-loss and shadowing.

The elements of $\mathbf{e}_c^{k,t}$ are random variables (RVs) with zero mean and variance $\frac{w_c^{k,t}}{P_c^{k,t,\mathrm{plt}}w_c^{k,t}+1}$, where $P_c^{k,t,\mathrm{plt}} = \tau_c^{k,t}P_c^{k,t}$ is the power used for the pilot signal by user k on subcarrier c and $\tau_c^{k,t} \in [0,1]$ is the power fraction used for pilots signaling.

The above expression (5) includes : 1) the received signal from user $k \in \mathcal{K}$ on subcarrier c with precoding ; 2) the error in CSI estimation due to pilots contamination effect ; and 3) the external interference from other users.

Taking into account the properties of the Minimum Mean-Square Error (MMSE)-based channel estimation, $\mathbf{e}_c^{k,t}$ is independent of $\tilde{\mathbf{h}}_c^{k,t}$. Then, using Maximum Ratio Combining (MRC) precoding with $\mathbf{f}_c^{k,t} = \left(\tilde{\mathbf{h}}_c^{k,t}\right)^H$, the received Signal-to-Interference-plus-Noise-Ratio (SINR) at the BS from user k serviced by slice s of MVNO m on subcarrier c, denoted $\eta_{s,m}^{k,c,t}$, can be given by (Tho Le-Ngoc et Derakhshani, 2017)

$$\eta_{s,m}^{k,c,t} = (\alpha_{s,m}^{k,t})^2(\beta_c^{k,t})^2 P_c^{k,t}\left\|\tilde{\mathbf{h}}_c^{k,t}\right\|^4 / \Bigg[\left\|\mathcal{I}_c^{k,t}\right\|^2 + (\alpha_{s,m}^{k,t})^2 (\beta_c^{k,t})^2\left(P_c^{k,t}\left\|\mathbf{e}_c^{k,t}\left(\tilde{\mathbf{h}}_c^{k,t}\right)^H\right\|^2 + \left\|\tilde{\mathbf{h}}_c^{k,t}\right\|^2\right)\Bigg],$$

where

$$\mathcal{I}_c^{k,t} = \sum_{m'\in\mathcal{M}}\sum_{s'\in\mathcal{S}_{m'}^t}\sum_{\substack{k'\in\mathcal{K}\\ k'\neq k}} \alpha_{s',m'}^{k',t}\beta_c^{k',t}\tilde{\mathbf{h}}_c^{k',t}\left(\tilde{\mathbf{h}}_c^{k',t}\right)^H\sqrt{P_c^{k',t}}\,\varkappa_c^{k',t}, \tag{6}$$

denotes the interference received on subcarrier c by user k from other transmitters, $\|\cdot\|$ is the Euclidean norm, and $(\cdot)^H$ is the Hermitian operator. From the law of large numbers, given a large $a^{k,t}$, $\forall k \in \mathcal{K}$, $\left\|\mathcal{I}_c^{k,t}\right\| \to 0$ (Tho Le-Ngoc et Derakhshani, 2017). Hence, the SINR expression simplifies into

$$\eta_{s,m}^{k,c,t} = \eta^{k,c,t} = \frac{P_c^{k,t}\left\|\tilde{\mathbf{h}}_c^{k,t}\right\|^4}{P_c^{k,t}\left\|\mathbf{e}_c^{k,t}\left(\tilde{\mathbf{h}}_c^{k,t}\right)^H\right\|^2 + \left\|\tilde{\mathbf{h}}_c^{k,t}\right\|^2}. \tag{7}$$

By substituting the variance of $\tilde{\mathbf{h}}_c^{k,t}$ by $\frac{P_c^{k,t,\mathrm{plt}}(w_c^{k,t})^2}{P_c^{k,t,\mathrm{plt}}w_c^{k,t}+1}$ (Tho Le-Ngoc et Derakhshani, 2017), $P_c^{k,t,\mathrm{plt}}$ by $\tau_c^{k,t}P_c^{k,t}$, and $P_c^{k,t}$ by $\rho^{k,t}/\sqrt{a^{k,t}}$ in (7), where $\rho^{k,t}$ is the total power of user k, we get

$$\eta^{k,c,t} = \frac{\tau_c^{k,t}(\rho^{k,t})^2(w_c^{k,t})^2}{1 + (1+\tau_c^{k,t})\rho^{k,t}w_c^{k,t}/\sqrt{a^{k,t}}}. \tag{8}$$

Assuming the asymptotic case of massive MIMO where $A \to \infty$, and a unitary uplink frame time (Tho Le-Ngoc et Derakhshani, 2017), then the data rate of user k on subcarrier c, denoted $r_c^{k,t}$, can be given by

$$r_c^{k,t} = \left(1 - \tau_c^{k,t}\right) \log_2 \left(1 + a^{k,t} \tau_c^{k,t} (P_c^{k,t} \ w_c^{k,t})^2\right). \tag{9}$$

Accordingly, we derive the total data rate $R^{k,t}$ of user k as the following

$$R^{k,t} = \sum_{c \in \mathcal{C}^t} \beta_c^{k,t} \cdot r_c^{k,t}, \tag{10}$$

where $R^{k,t}$ is expressed as a function of $\beta_c^{k,t}$, $\tau_c^{k,t}$, $a^{k,t}$ and $P_c^{k,t}$. For the wireless resource allocation of the MNO, we define the following objective function :

$$\max_{\beta_c^{k,t}, a^{k,t}} \sum_{k \in \mathcal{K}} \left[g^{k,t} u^{k,t} - \mathfrak{C}_1 \left(P_\pi a^{k,t} + P_0 + \frac{1}{\nu} \sum_{c \in \mathcal{C}^t} P_c^{k,t} \beta_c^{k,t} \right) \right], \tag{11}$$

where $\mathfrak{C}_1(\cdot)$ denotes the energy consumption cost by active antennas and subcarriers. Note that we focus mainly on the BS's power consumption since it has been identified as the dominant energy consumer in wireless networks (Z. Hasan et al., 2011).

Hence, similarly to (Xu et L. Qiu, 2013 ; C. Li et al., 2014), we consider a power model where ν denotes the power amplifier efficiency, P_0 is the static power independent of the number of active antennas and transmission power (e.g., battery units) and P_π is the dynamic power consumption proportional to the number of active antennas (e.g., circuit power of corresponding radio frequency (RF) chains and processing power) (Xu et L. Qiu, 2013 ; C. Li et al., 2014). The objective here is to maximize the MNO's profit for serving users' requests. The profit is issued from the revenue obtained when satisfying users' requests minus the costs of allocating the BS's antennas and subcarriers to users. In addition to constraints (3)–(4), we also enforce the following constraints :

$$\alpha_{s,m}^{k,t} \sum_{c \in \mathcal{C}^t} \beta_c^{k,t} \cdot P_c^{k,t} \leq P_{\max}^{k,t}, \ \forall \ m \in \mathcal{M}, \ s \in \mathcal{S}_m^t, \ k \in \mathcal{K}, \tag{12}$$

$$\alpha_{s,m}^{k,t} R^{k,t} \geq u^{k,t}, \ \forall \ m \in \mathcal{M}, \ s \in \mathcal{S}_m^t, \ k \in \mathcal{K}, \tag{13}$$

where constraint (12) ensures that a maximum power budget is respected by each user, while (13) guarantees that data rate requirements are satisfied for the users served by the MNO.

The resulting problem (P1), defined by (3)–(4) and (11)–(13), is NP-Hard. To prove it, we leverage the NP-hardness reduction approach in which we consider a simplified single time slot problem where subcarriers are assumed to operate without any interference. We also assume an identical data rate requirement for all users.

This particular problem can be easily comprehended as the well-known Generalized Assignment Problem (GAP) (Cohen et al., 2006). In GAP, the objective is to find a maximum overall profit assignment of tasks with different amounts of resources to agents, such that each task is assigned to precisely one agent, subject to the agents' capacity. Also, each task has a different profit, depending on the assigned agent.

In our problem, agents can be assimilated to subcarriers and end-user requests to tasks, where task profit is linked to the profits made. Since the GAP is known to be NP-hard (Katoh et Ibaraki, 1998; Cohen et al., 2006), then, by restriction, our simplified version of the problem is also NP-hard.

3.4.2 MNO's Green VNF Resource Provisioning Problem

To provision the enhanced multimedia services requested by MVNO users, the MNO shall deploy the required VNF chains of these services within its core network.

Here, the MNO's objective is to minimize VNF placement costs, including VNF instances deployment, computing, and communication costs, while reducing its environmental impact in terms of carbon footprint. The physical topology of the

TABLE 3.1 Notations (Wireless resource allocation)

$\mathcal{C}^t$	Set of subcarriers available at BS
$\mathcal{K}$	Set of users
$\mathcal{M}$	Set of MVNOs
$\mathcal{S}_m^t$	Set of slices owned by MVNO m
$\alpha_{s,m}^{k,t}$	Binary indicator of user k's association to slice $s \in \mathcal{S}_m^t$
$\beta_c^{k,t}$	Binary indicator of subcarrier c's allocation to user k
$\rho^{k,t}$	Total power of user k
$\tau_c^{k,t}$	Power fraction for pilots signaling by user k in subcarrier c
$\chi_{c,i}^{k,t}$	Small-scale fading coefficient between BS's i^{th} antenna and user k on subcarrier c
A	Total number of BS antennas
$a^{k,t}$	Number of antennas assigned to user k
$\sigma^{k,t}$	Service request sent by user k
$u^{k,t}$	Data rate requirement of user k
$g^{k,t}$	Available budget to serve user k's request
$P_c^{k,t}$	Transmit power of user k on subcarrier c
$P_{\max}^k$	Maximum transmit power of user k
P_0	static antenna power
P_π	dynamic antenna power
ν	Power amplifier efficiency
$r_c^{k,t}$	Data rate of user k on subcarrier c
$R^{k,t}$	Total data rate for user k
$\varkappa_c^{k,t}$	transmitted signal of user k on subcarrier c
$w_c^{k,t}$	large-scale power attenuation
$\mathbf{e}_c^{k,t} \in \mathbb{C}^{1\times a^{k,t}}$	CSI error
$\mathbf{f}_c^{k,t} \in \mathbb{C}^{a^{k,t}\times 1}$	Precoding vector for user k's signal using subcarrier c
$\mathbf{h}_c^{k} \in \mathbb{C}^{1\times a^{k,t}}$	Uplink real channel vector of user k
$\tilde{\mathbf{h}}_c^{k,t} \in \mathbb{C}^{1\times a^{k,t}}$	Uplink estimated channel vector of user k
$\mathbf{z}_c^{k,t} \in \mathbb{C}^{1\times a^{k,t}}$	Additive white Gaussian noise
$\mathcal{I}_c^{k,t} \in \mathbb{C}^{1\times a^{k,t}}$	Interference received on subcarrier c by user k from users $k' \neq k$

core network is modeled as a directed graph $\mathcal{G}(\mathcal{N}, \mathcal{B}, \mathcal{E})$ with $\mathcal{N}$ representing the set of nodes, e.g., physical servers, connected through a set of directional edges $\mathcal{E}$, e.g., communication links, and $\mathcal{B}$ representing the BS.

Let $\mathcal{V} = \{1, 2, ..., |\mathcal{V}|\}$ be the set of VNF types, where each VNF type $v \in \mathcal{V}$ is characterized by a tuple $\{\kappa^v, \varrho^v\}$, with κ^v and ϱ^v the required number of vCPUs and minimum I/O bandwidth (in Mbps), respectively. We assume that the deployment of a VNF onto a node is done using a virtual machine (VM). Thus, we define $\mathcal{X} = \{1, 2, ..., |\mathcal{X}|\}$ as the set of VM types, each characterized by a tuple $\{\varsigma_x, \varrho_x\}$, with ς_x and ϱ_x the available number of vCPUs and I/O bandwidth (in Mbps), respectively. We also consider a licensing model where the number of instances for VNF type v, denoted I^v, is limited by a maximum value $I^v_{\max}$ at any given time. Also, deploying an instance of VNF type v incurs a software license fee, denoted ϵ^v. From the perspective of the core network, a user service request is seen as a VNF chain to execute an enhanced multimedia service. Let $q^k = \{l^k, d^k, \mathcal{G}(\mathcal{V}^k, \mathcal{E}^k)\}$ be user k's service request, characterized by a multimedia content load l^k, a delay requirement d^k, a VNF chain modeled as a Directed Acyclic Graph (DAG) of VNFs, where $\mathcal{V}^k$ and $\mathcal{E}^k$ are the set of VNFs in the service's chain and the set of virtual edges linking them, respectively.

Let $b^v_{n,x,i}$ be the binary indicator of deploying the i^{th} instance of VNF type v in node n through VM type x, i.e.,

$$b^v_{n,x,i} = \begin{cases} 1, & \text{if instance } i \text{ of VNF type } v \text{ is deployed in} \\ & \text{node } n \text{ using VM type } x, \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

where $i \leq I^v_{\max}$, $v \in \mathcal{V}$, and $n \in \mathcal{N}$. Also, let $\phi_n \in \{0, 1\}$ designate the availability of a server node. Indeed, a node can be unavailable due to a physical/software fault or due to resources saturation.

Similarly to (KHOSRAVI et al., 2013; MÖBIUS et al., 2013), we express the power consumption P_n of a node n, hosting multiple VMs by

$$P_n = P_n^{idle} + \sum_{v \in \mathcal{V}} \sum_{i=1}^{I^v} \sum_{x \in \mathcal{X}} b_{n,x,i}^v \cdot P_{x,i}^v, \tag{15}$$

where P_n^{idle} denotes the idle power consumption of node n and P_x is the power consumption of a VM of type x. The latter parameter is defined by the following (KHOSRAVI et al., 2013; MÖBIUS et al., 2013) :

$$P_{x,i}^v = \kappa^v \cdot P_{x,i}^{v,cpu} + P_{x,i}^{v,cache} + P_{x,i}^{v,mem} + P_{x,i}^{v,disk}, \tag{16}$$

where $P_{x,i}^{v,cpu}$, $P_{x,i}^{v,cache}$, $P_{x,i}^{v,mem}$ and $P_{x,i}^{v,disk}$ denote the CPU, cache, memory and disk power consumption after deploying VNF instance i, of type v in VM type x, respectively. Note that we consider a 1 :1 vCPU to physical CPU ratio.

Consequently, the operational cost of node $n \in \mathcal{N}$ can be defined by

$$\psi_n = \phi_n \cdot \mathfrak{C}_2 \left(\mu_n P_n \right), \tag{17}$$

where $\mathfrak{C}_2 (\cdot)$ is the power consumption cost function. In this paper, we assume that nodes may be powered by different electrical energy sources with varying environmental impacts. On the one hand, we may have fossil-based energy sources that are considered harmful due to the greenhouse effect and causing air pollution. On the other hand, we have green energy sources, expected to become more mainstream in future networks.

Hence, the defined parameter μ_n in (17) represents the mean CO_2 emissions (in gCO_2/W), while P_n is the consumed power by node n (in W). A reference of various mean CO_2 emissions (in gCO_2/kWh) per energy source is provided in Table 3.2 below. Besides the operational cost of node $n \in \mathcal{N}$, instantiating a VNF chain $\mathcal{G}\left(\mathcal{V}^k, \mathcal{E}^k\right)$ generates a communication cost $\gamma_{n,n'}^{k,v,v'}$, which denotes the bandwidth costs among nodes hosting different instances of required VNFs for the chain in service request q^k. Let $e_{n,n'} \in \mathcal{E}$ and $e^k_{v,v'} \in \mathcal{E}^k$ be the edges between nodes n and n', and between VNFs instances v and v' of user k's request, respectively.

Then, we define by $y^{k,v,v'}_{n,n'}$ the binary indicator of the correlation between the physical communication link between nodes n and n', i.e., edge $e_{n,n'}$, and instantiating VNF chain's virtual edge $e^k_{v,v'}$. It is given by

$$y^{k,v,v'}_{n,n'} = \begin{cases} 1, & \text{if edge } e_{n,n'} \in \mathcal{E} \text{ hosts VNF edge } e^k_{v,v'} \in \mathcal{E}^k \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

The communication link between nodes n and n' includes also a number of routers and switches. To this end, assume $\widetilde{\mathcal{N}}^{k,v,v'}_{n,n'}$ as the set of switches and routers within edge $e_{n,n'}$. For $\widetilde{n} \in \widetilde{\mathcal{N}}^{k,v,v'}_{n,n'}$, we define the incurred power consumption $P_{\widetilde{n}}$ as the following (VISHWANATH et al., 2014) :

$$P_{\widetilde{n}} = P^{idle}_{\widetilde{n}} + R^{byte}\left(\frac{P^{proc}_{\widetilde{n}}}{l^k} + P^{store}_{\widetilde{n}}\right), \tag{19}$$

where $P^{idle}_{\widetilde{n}}$ represents the idle power consumption of the switch or router, R^{byte} denotes the input byte rate, $P^{proc}_{\widetilde{n}}$ is the per-byte processing energy and $P^{store}_{\widetilde{n}}$ is the store and forward energy (VISHWANATH et al., 2014).

We define by $P^{v,v'}_{n,n'}$ the total carbon footprint of an edge $e_{n,n'}$ by

$$P^{v,v'}_{n,n'} = \sum_{\widetilde{n} \in \widetilde{\mathcal{N}}^{k,v,v'}_{n,n'}} \mu_{\widetilde{n}} \cdot P_{\widetilde{n}}, \tag{20}$$

where $\mu_{\widetilde{n}}$ represents the mean CO2 emissions (in gCO_2/W) of a node $\widetilde{n}$.

Consequently, $\lambda^{k,v,v'}_{n,n'}$ is the total cost of edge $e_{n,n'}$ when $y^{k,v,v'}_{n,n'} = 1$, expressed by

$$\lambda^{k,v,v'}_{n,n'} = \gamma^{k,v,v'}_{n,n'} + \mathfrak{C}_3(P^{v,v'}_{n,n'}) \quad \forall n \neq n', \tag{21}$$

where $\mathfrak{C}_3(P^{k,v,v'}_{n,n'})$ is the incurred power cost of communication between n and n'.

TABLE 3.2 CO_2 emissions (RICCIARDI et al., 2013)

Energy source	**Renewable**	**Mean value (gCO_2/kWh)**
Solar, wind, hydro-electric	Yes	0
Nuclear	No	20
Geothermal	Yes	107
Biomasses	Yes	180
Natural gas	No	370
Fuel	No	880
Coal	No	980

We denote by $u_{n,x,i}^{k,v}$ the binary parameter indicating whether a VNF belonging to a chain in user k's request is mapped to instance i deployed in node n using VM type x, i.e.,

$$u_{n,x,i}^{k,v} = \begin{cases} 1, & \text{if instance } i \text{ of } v \in \mathcal{V}^k \text{ is deployed in} \\ & \text{node } n \in \mathcal{N} \text{ using VM type } x \text{, for request } q^k, \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

From (22), it is clear that parameter $b_{n,x,i}^{v}$ defined in (14) can be expressed as $b_{n,x,i}^{v} = \max_{k \in \mathcal{K}} u_{n,x,i}^{k,v}$.

To guarantee the servicing of user request q^k, it is required that for each $v \in \mathcal{V}^k$ at least one instance is placed within the core network, i.e.,

$$\sum_{n \in \mathcal{N}} \sum_{i=1}^{I^v} u_{n,x,i}^{k,v} \geq 1, \quad \forall k \in \mathcal{K}, v \in \mathcal{V}^k. \tag{23}$$

In addition, the mapping of VNFs and their virtual edges to physical nodes and edges must satisfy the following :

$$y_{n,n'}^{k,v,v'} = \left(\sum_{i=1}^{I^v} u_{n,x,i}^{k,v}\right) \cdot \left(\sum_{i'=1}^{I^v} u_{n',x',i'}^{k,v'}\right),$$
$$\forall k \in \mathcal{K}, e_{v,v'}^{k} \in \mathcal{E}^k, e_{n,n'} \in \mathcal{E}. \quad (24)$$

Finally, each virtual edge of a VNF chain requested by user k has to be mapped to only one pair of physical nodes (n, n'), i.e.,

$$\sum_{e_{n,n'} \in \mathcal{E}} y_{n,n'}^{k,v,v'} = 1, \forall k \in \mathcal{K}, e_{v,v'}^{k} \in \mathcal{E}^k. \quad (25)$$

Obviously, before placing any VNF, we must ensure the availability of sufficient processing and resources capacity. We notably ensure that a VM has enough resources to host the VNF and in turn that a node can host the VM. Hence, considering $d_{n,n'}$ as the delay over edge $e_{n,n'} \in \mathcal{E}$ per load unit, these constraints can be expressed by

$$\sum_{k \in \mathcal{K}} \frac{l^k y_{n,n'}^{k,v,v'}}{d_{n,n'}} \leq \varrho^v b_{n,x,i}^{v}, \forall (v, v') \in \mathcal{V}^2, (n, n') \in \mathcal{N}^2,$$
$$i = 1, \ldots, I^v, x \in \mathcal{X}, \quad (26)$$

and

$$\varrho^v b_{n,x,i}^{v} \leq \varrho_x, \forall n \in \mathcal{N}, i = 1, \ldots, I^v, x \in \mathcal{X}. \quad (27)$$

Also, we ensure that we do not overload a VNF instance by assigning too many user requests to it or by deploying VNF instances beyond node capacity ς_n (in vCPUs) as follows :

$$\sum_{v \in \mathcal{V}} \sum_{i=1}^{I^v} \kappa^v b_{n,i}^{v} \leq \varsigma_x, \forall x \in \mathcal{X}, \quad (28)$$

and

$$\varsigma_x \leq \phi_n \varsigma_n, \forall n \in \mathcal{N}, x \in \mathcal{X}. \tag{29}$$

$\delta_{v,n,x}$ be the processing delay of VNF v in node n using VM type x per load unit. For a timely delivery of service to user k, the following condition has to be respected :

$$\sum_{\substack{k\in\mathcal{K}\\ v\in\mathcal{V}^k\\ i=1,\ldots,I^v}} l^k\, u^{k,v}_{n_0,x,i}\, d_{\mathcal{B},n_0} + \sum_{\substack{k\in\mathcal{K}\\ v\in\mathcal{V}^k\\ i=1,\ldots,I^v}} l^k\, u^{k,v}_{n_1,x,i}\, d_{n_1,\mathcal{B}} + \sum_{\substack{k\in\mathcal{K}\\ e^k_{v,v'}\in\mathcal{E}^k\\ e_{n,n'}\in\mathcal{E}}} l^k\, y^{k,v,v'}_{n,n'}\, d_{n,n'} + \sum_{\substack{k\in\mathcal{K}\\ n\in\mathcal{N}\\ v\in\mathcal{V}^k\\ i=1,\ldots,I^v}} l^k\, u^{k,v}_{n,x,i}\, \delta_{v,n,x} \leq d^k, \tag{30}$$

where n_0 and n_1 denote the nodes hosting the first and last VNFs of the required VNF chain, respectively. Hence, we account for the communication delay between : 1) the base station and the node hosting the first VNF of the VNF chain request ; 2) the node hosting the last VNF and the base station ; 3) each pair of nodes involved in hosting VNFs' instances of the chain ; and for the processing delays within these nodes.

Similarly, the load on a given physical edge should not exceed the available bandwidth capacity. Given that $\Delta_{n,n'}$ is the available bandwidth (in Mbps) on edge $e_{n,n'} \in \mathcal{E}$, the following constraint is enforced :

$$\sum_{\substack{k\in\mathcal{K}\\ v\in\mathcal{V}^k\\ v'\in\mathcal{V}^k}} \sum_{e^k_{v,v'}\in\mathcal{E}^k} l^k y^{k,v,v'}_{n,n'} d_{n,n'} \leq \Delta_{n,n'}, \forall e_{n,n'} \in \mathcal{E}. \tag{31}$$

In this problem, the objective is to minimize the operational and communication costs of nodes, as well as VNFs deployment costs among them. It is expressed by

$$\min_{\substack{u^{k,v}_{n,x,i},\\ y^{k,v,v'}_{n,n'}}} \Bigg[\sum_{\substack{v\in\mathcal{V}\\ n\in\mathcal{N}\\ x\in\mathcal{X}}} \sum_{i=1}^{I^v} \epsilon^v b^v_{n,x,i} + \sum_{n\in\mathcal{N}} \psi_n + \sum_{\substack{k\in\mathcal{K}\\ n\in\mathcal{N}\\ e^k_{v,v'}\in\mathcal{E}^k\\ e_{n,n'}\in\mathcal{E}}} \lambda^{k,v,v'}_{n,n'}\, y^{k,v,v'}_{n,n'} \Bigg]. \tag{32}$$

Similarly to the previous MNO's formulated problem, the problem given by (23)–(32), denoted (P2), is also NP-Hard. We illustrate this by considering a simplified single time slot problem where VNFs are placed in nodes to minimize costs while guaranteeing end-user QoS requirements.

We further focus on the particular case of homogeneous nodes (same capacity), while VNFs may have different resource requirements. The defined problem can be seen as a GAP where the agents are assimilated to nodes, VNFs to tasks, and task profit as cost savings. Since GAP is known to be NP-hard, then our simplified problem is also NP-hard.

TABLE 3.3: Notations (Green VNF resource provisioning)

Symbol	**Meaning**
$\mathcal{B}$	MNO's BS
$\mathcal{N}$	Set of core network nodes
$\mathcal{E}$	Set of core network edges
$\mathcal{E}^k$	Virtual edges (links) between VNFs in service request q^k
$\mathcal{V}$	Set of VNF types
$\mathcal{V}^k$	VNF chain of service request q^k
$\mathcal{X}$	Set of VM types
ς_x	Capacity (in vCPUs) of VM type x
ϱ_x	Available I/O bandwidth (in Mbps) in VM type x
ϵ^v	Software license fee for an instance of VNF v
q^k	Service request of user k
I^v	Number of instances for VNF type v

Continued on next page

Table 3.3 – continued from previous page

Symbol	Meaning
I^v_{max}	Maximum number of instances for $v \in \mathcal{V}, \forall t \in \mathcal{T}$
l^k	Load of user k
d^k	Delay requirement for user k's service request
$b^v_{n,x,i}$	Binary indicator of an instance i of VNF v's deployment in node n using VM type x
$P^{k,v,v'}_{n,n'}$	Incurred CO_2 emission for $e^k_{v,v'}$ deployed between n and n'
$u^{k,v}_{n,x,i}$	Binary indicator of an instance i of VNF v's deployment on node n using VM type x for service request q^k
κ^v	Required number of vCPUs for VNF v
ϱ^v	Minimum I/O bandwidth (in Mbps) for VNF v
ϕ_n	Binary indicator of a node n's availability
ψ_n	Operational cost of node n
$\lambda^{k,v,v'}_{n,n'}$	Total cost of edge $e_{n,n'}$ for $e^k_{v,v'}$ deployed between n and n'
$\gamma^{k,v,v'}_{n,n'}$	Communication cost for $e^k_{v,v'}$ deployed between n and n'
ς_n	Capacity (in vCPUs) of node n
$\delta_{v,n,x}$	Processing delay of VNF v in node n using VM type x per load unit
$\Delta_{n,n'}$	Available bandwidth (in Mbps) on edge $e_{n,n'}$

3.5 Proposed Solutions

In this section, we begin by detailing the architectural framework assumptions for the use of machine learning in next-generation networks. Then, we describe the proposed heuristic DRL-MCTS, which leverages DRL and Monte Carlo Tree Search (MCTS) to solve the MNO's green wireless resource allocation problem. Finally, we present our proposed DRL-VNF algorithm, a DRL-based approach for solving the MNO's green VNF resource provisioning problem.

3.5.1 The Architectural Framework for Machine Learning in Future Networks

This paper follows the ITU-T Y.3172 architectural framework recommendations for machine learning in future networks (Union-T, 2020). Expressly, we assume the existence of an isolated machine learning sandbox domain that enables MNOs to train, test, and evaluate their machine learning models in parallel and on dedicated nodes before their deployment in live networks.

This allows the MNO to proactively and fully evaluate from different perspectives, i.e., performance, profitability, etc., the impact of a resource management decision. To increase the deployed ML training model efficiency, we employ data parallelism consisting of splitting data into multiple partitions and, subsequently, train copies of the model in parallel with their own allocated data samples.

3.5.2 DRL-MCTS : A Hybrid DRL and MCTS-based Wireless Resource Allocation Strategy

We present our proposed algorithm to solve the MNO's wireless resource allocation problem. DRL-MCTS solves this problem by jointly determining the optimal subcarrier(s)-end-user associations (hereafter abbreviated as S-E), and the number

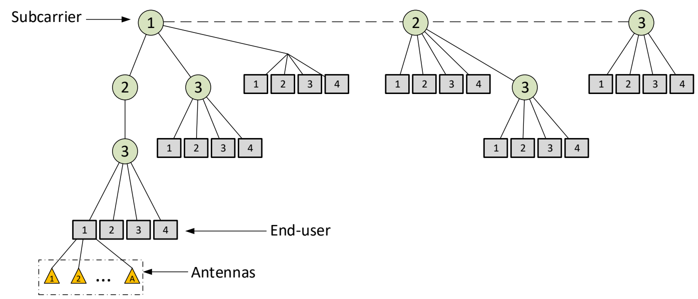

FIGURE 3.2 Potential S-E associations.

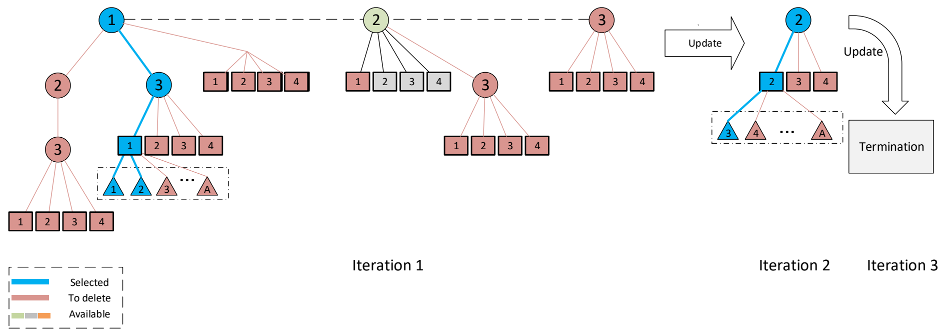

FIGURE 3.3 DRL-MCTS iterative run.

of antennas assigned for each S-E, to maximize the MNO's profit while ensuring the end-users QoS requirements.

3.5.2.1 DRL-MCTS Algorithm Description

As it can be seen in problem (P1), defined by (3)–(4) and (11)–(13), the MNO is essentially tasked to decide when and how many subcarriers and antennas it needs to assign to user k in order to guarantee the data rate requirement $u^{k,t}$, and receive a profitable payoff $g^{k,t}$, at the expense of other users.

Obviously, given the large number of subcarriers, antennas, and users, a naive search would be too inefficient and computationally complex. To tackle this problem efficiently, we instead leverage the notion of powerset, where, given a set Ξ, the powerset $\wp(\Xi)$ is defined as the set of all subsets of Ξ including the empty set and the set Ξ itself.

By combining the powerset of subcarriers $\wp(\mathcal{C}^t)$ with that of users set $\mathcal{K}$, we obtain a tree representation of all possible S-E associations. Our goal is to efficiently generate all the possible S-E associations, which are subsequently filtered by their profitability to the MNO. In this regard, it is worth noting that the generation of powersets is parallelizable (Goodwin, 2019). In addition, a tree representation enables us to leverage efficient parallel algorithms for traversal and compression operations.

For illustration purposes, we present in Fig. 3.2 the S-E associations with $\mathcal{C}^t = \{1, 2, 3\}$ and $\mathcal{K} = \{1, 2, 3, 4\}$ as powerset disjoint trees with subcarriers (circles), users (squares) and the associated number of antennas (triangles), representing the tree nodes and leaves respectively.

Evidently, it must be noted that the computation complexity for the tree traversal is mainly dependent on its height $\top$. However, since the problem's objective includes profit maximization, the worst-case tree height $\top_{\max}$ can be preemptively bounded to reduce computation complexity. Indeed, let $\nabla = \max_{k\in\mathcal{K}}(g^{k,t})$ be the maximum payoff among users, and ∇_c, the cost of the cheapest subcarrier. we argue that the maximum number of subcarriers (i.e maximum tree height) $\top_{\max}$ that can be allocated to a single user while maintaining positive profits is derived by $\top_{\max} = \nabla/\nabla_c$.

Fig. 3.3 shows an example run of our proposal for the previous scenario, where we also consider the following user requirements tuples $\{(20, 100), (10, 50), (10, 20), (10, 10)\}$, with the first and second elements of the tuple representing the data rate and payoff requirements, respectively.

For simplicity's sake, we assume that each subcarrier can provide a data rate of 10 for any user using one antenna and costs a price of 2. Hence, to satisfy a data rate requirement of 20, the use of two subcarriers and two antennas is required. In such a scenario, the optimal S-E association for user 1 is either aggregated subcarriers 1 and 3 or 2 and 3. If subcarriers 1 and 3 are selected at the first DRL-MCTS iteration (shown in bold blue lines), the remaining trees are updated by removing the selected nodes and children.

In the next iteration, the optimal S-E association is subcarrier 2 for user 2 with one antenna. After this iteration, no more subcarriers are left, and the DRL-MCTS algorithm terminates. Also, note that if the search starts from the most-left tree and no S-E association can be found for the concerned user due, for instance, to a high data rate requirement, then the latter can be safely removed and truncated from all trees. By doing so, after each iteration where an S-E association is made, both the height and breadth of trees are reduced.

The proposed DRL-MCTS algorithm follows a two-step process summarized in Algorithm 3. The first step, DRL, uses a multi-agent hierarchical DRL framework (Kumar et al., 2017) that takes as inputs end-user requests and environmental characteristics, while as outputs, each end-user request is associated with estimates of the number of subcarriers, number of antennas, and power allocation among antennas, required to satisfy the QoS requirements. Subsequently, the outputs are appended in turn to the subcarrier powerset to form potential S-E associations, as shown in Fig. 3.2.

The use of the multi-agent hierarchical DRL framework is motivated by the following. On the one hand, while multi-agent RL can be used to find decentralized policies that jointly optimize the private value functions of participating agents, it scales poorly with the problem's size (Kumar et al., 2017). Also, the communication overhead increases exponentially with the number of agents such that a substantial combined action space must be explored before receiving feedback from the environment. On the other hand, hierarchical reinforcement learning (HRL) enables learning goal-directed behavior from sparse feedback in complex environments by dividing the overall task into independent sub-tasks that can be solved sequentially. In HRL, a meta-controller is trained to pick the next goal, while a controller learns to reach individual goals.

The multi-agent hierarchical DRL framework consists of a meta-controller and controllers illustrated in Fig. 3.4, which represent hierarchically organized DRL modules operating at separate time scales. In our case, a controller characterizes a decentralized agent, which receives a partial view of the state and chooses actions that maximize its private value function.

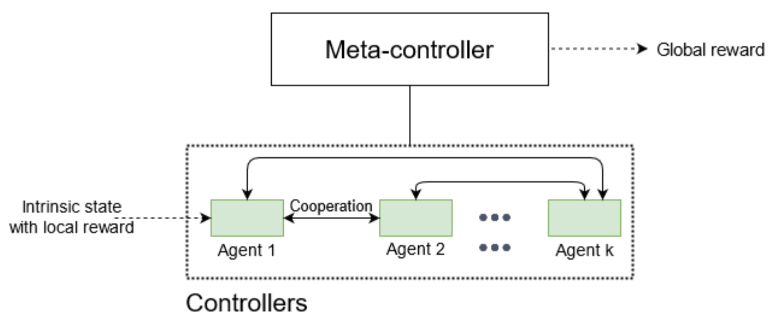

FIGURE 3.4 Overview of the multi-agent hierarchical DRL framework, inspired by (KUMAR et al., 2017).

We opt for an approach where neural networks are distributively executed at the agents, whereas training is centralized in the machine learning sandbox, in order to ease implementation and improve learning stability. Each agent has the same neural network parameters at time t.

Despite synchronized neural parameters, agents may still behave differently due to their individual state inputs. In interfered wireless networks, an agent is by default incentivized to increase its transmit power to improve its own data rate at the expense of other transmitters. In our context, the agent is first tasked to individually estimate the required number of subcarriers and antennas to satisfy its QoS requirements, assuming a fixed antenna power. Afterward, agents are intrinsically motivated to negotiate with each other and cooperatively determine the adequate antenna power to jointly satisfy their QoS requirements and contribute to the maximization of the objective function defined in (11). The pair of negotiating agents are chosen by the meta-controller. For scalability purposes, these agents are clustered based on their locations given the inverse correlation between interference and distance.

To find a globally coherent set of agent actions, the meta-controller is rewarded by the environment through the following reward function

$$\mathcal{R}(t) = \sum_{k \in \mathcal{K}} u^{k,t}. \tag{33}$$

Since an agent needs to communicate with a single controller at any step, its policy can be trained separately via self-play to maximize expected future intrinsic reward with gradient descent. As the details of individual negotiations are abstracted away from the meta-controller, the latter can efficiently explore large-sized spaces of agent pairs. The agent manages two DRL modules using two deep neural networks (DNNs), denoted $\mathcal{J}^1_\theta$ and $\mathcal{J}^2_{\theta'}$ respectively, where θ and θ' are their associated neural parameters, respectively. The first neural network $\mathcal{J}^1_\theta$ takes as inputs the MVNO end-user's request σ^k along with the SINR $\eta^{k,c,t}$. It is trained using DRL to estimate the number of subcarriers and antennas necessary for the agent to satisfy the end-user's QoS requirements. Then, it outputs a tuple $(\tilde{c}, \tilde{a})$, where $\tilde{c}$ and $\tilde{a}$ are the estimated numbers of subcarriers and antennas, respectively. The action set is defined as $\{(\tilde{c}, \tilde{a}) \mid \tilde{c} \in \mathcal{C}^t,\ \tilde{a} \in [1, A]\}$. The reward mechanism is defined by

$-\frac{(R^{k,t}-u^{k,t})^2}{\Gamma}$ where

$$\Gamma = \begin{cases} \Gamma_1, & \text{if } R^{k,t} - u^{k,t} > 0 \\ \Gamma_2, & \text{otherwise,} \end{cases} \tag{34}$$

$(\Gamma_1, \Gamma_2) \in \mathbb{R}^*$, and $\Gamma_2 < \Gamma_1$ are scaling factors to discourage both over and under-provisioning. Obviously, under-provisioning is prohibited to ensure QoS satisfaction and over-provisioning is avoided for green energy-efficiency purposes.

Once the neural network $\mathcal{J}^1_\theta$ is adequately trained, we leverage its output as part of the inputs for $\mathcal{J}^2_{\theta'}$. Also, we denote by $\{\sigma^k, \eta^{k,\tilde{c}}, P^k_{\max}, \tilde{c}, \tilde{a}, \mathcal{R}^k(t-1)\}$ the tuple of the request, SINR, maximum power constraint, the number of subcarriers and antennas for end-user k at time t, and the previous reward of end-user k at time $t-1$, respectively. Similarly, the tuple $\{\sigma^{k'}, \eta^{k',\tilde{c}}, P^{k'}_{\max}, \tilde{c}', \tilde{a}', \mathcal{R}^{k'}(t-1)\}$ is defined for end-user k'. For the pair (k, k') of one cluster, a pair of negotiating agents is assigned by the meta-controller. Note that paired agents are synchronized and take their actions at the same time. Prior to taking an action, agent of end-user k observes the effect of its paired agent's past actions on its current state, but has no knowledge of the current action taken by the paired agent k'. Accordingly, an agent k can estimate the impact of its own actions on the paired agent, which can be used to determine a policy that maximizes future rewards. With the preceding tuples as inputs, the neural network $\mathcal{J}^2_{\theta'}$ outputs the power allocation for the BS's antennas to satisfy agents' QoS requirements while accounting for interference. The action set is defined as $\Omega^{k,t} = \{0, P^k_1, \ldots, P^k_{\max}\}$ where at a given i-th index, $P^k_i = P^k_{i-1} + \frac{P^k_{\max}}{|\Omega^{k,t}|}$ where $|\Omega^{k,t}|$ is the number of power levels. The reward mechanism is defined by

$$\mathcal{R}^k(t) = -\frac{(R^{k,t} - u^{k,t})^2}{\Gamma} - \frac{(R^{k'} - u^{k',t})^2}{\Gamma} - \varphi, \tag{35}$$

where

$$\Gamma = \begin{cases} \Gamma_1, & \text{if } R^{k,t} - u^{k,t} \text{ (or } R^{k'} - u^{k',t}) > 0 \\ \Gamma_2, & \text{otherwise} \end{cases} \tag{36}$$

and φ is a small number such that

$$\varphi = \begin{cases} \varphi, & \text{if } R^{k,t} = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{37}$$

The above reward function is designed to be maximized when both negotiating agents achieve their QoS requirements. When the agents' QoS cannot be simultaneously satisfied, the following maximum reward occurs when one agent does not transmit in favor of the other agent (i.e., $P^k = 0$ or $P^{k'} = 0$).

The second step of DRL-MCTS uses MCTS to search for the potential S-E associations and find the optimal global solution. MCTS combines a tree search approach with Monte Carlo simulations (also known as rollouts) and uses the outcome of these simulations to evaluate states in a look-ahead tree. It requires little to no domain knowledge and constitutes an attractive approach for our problem since its execution can be parallelized. For readability purposes, we omit the MCTS details, which can be found in (Browne et al., 2012) and references therein.

In the MCTS process, we calculate the profit of each S-E pair as

$$g^{k,t} - \mathfrak{C}_1 \left(P_\pi a^{k,t} + P_0 + \frac{1}{\nu} \sum_{c \in \mathcal{C}^t} P_c^{k,t} \beta_c^{k,t} \right). \tag{38}$$

After each MCTS selection phase, where the most profitable S-E is selected, we update the powerset as shown in Fig. 3.3 until no further S-E associations are possible.

3.5.3 DRL-VNF : A Green DRL-based VNF Provisioning Strategy

This section describes our proposal for VNF provisioning with green considerations summarized in Algorithm 4. The MNO is tasked with placing end-user service requests at minimal costs within its core network. If unable to do so, for instance,

Algorithm 3: DRL-MCTS algorithm

Input : $\mathcal{C}^t$, $\mathcal{K}$, A, $\{\sigma^k$ for $k \in \mathcal{K}\}$

Generate $\wp = \wp(\mathcal{C}^t) \times \wp(\mathcal{K})$

for k = 1 to $\mathcal{K}$ **do**

    // Using DQN (Mnih et al., 2015)

    $(\tilde{c}, \tilde{a}, \mathcal{R}^k(t-1)) = \mathcal{J}^1_\theta(\sigma^k, \{\eta^{k,c,t}$ for $c \in \mathcal{C}^t\})$;

end

Init. experience replay buffers respectively for meta-controller and agents D_{meta}, D_{agent}

Init. meta-controller's Q-network $\mathcal{J}^{meta}_\theta$ and agents' $\mathcal{J}^2_\theta$ with random weights

for *episode = 1 to numEpisodes* **do**

    Obtain pairs of negotiating agents (k, k') from meta-controller;

    // state_meta : meta-controller state

    while *state_meta is not terminal* **do**

        meta-controller generates agent pairs (k, k')

        $\text{state}_k = \{\sigma^k, \eta^{k,\tilde{c}}, P^k_{\max}, \tilde{c}, \tilde{a}, \mathcal{R}^k(t-1)\}$

        $\text{state}_{k'} = \{\sigma^{k'}, \eta^{k',\tilde{c}}, P^{k'}_{\max}, \tilde{c}', \tilde{a}', \mathcal{R}^{k'}(t-1)\}$

        for *1 to num_negotiation_rounds* **do**

            Select P^k (and $P^{k'}$) from $\Omega^{k,t}$ (from $\Omega^{k',t}$) using epsilon-greedy.

            $\text{next\_state}_k = \{\sigma^k, \eta^{k,\tilde{c}}, P^k_{\max}, \tilde{c}, \tilde{a}, \mathcal{R}^k(t-1), P^k\}$

            $\text{next\_state}_{k'} = \{\sigma^{k'}, \eta^{k',\tilde{c}}, P^{k'}_{\max}, \tilde{c}', \tilde{a}', \mathcal{R}^{k'}(t-1), P^{k'}\}$

            Compute rewards $\mathcal{R}^k(t)$ and $\mathcal{R}^{k'(t)}$

            Store (state_k, P^k, $\mathcal{R}^k(t)$, next_state_k) in D_{agent}

            Store ($\text{state}_{k'}$, $P^{k'}$, $\mathcal{R}^{k'(t)}$, $\text{next\_state}_{k'}$) in D_{agent}

            $\text{state}_k = \text{next\_state}_k$

            $\text{state}_{k'} = \text{next\_state}_{k'}$

        end

        Sample mini-batch from D_{agent} and update $\mathcal{J}^2_\theta$ weights.

        Append in state_meta $\{(k, k'), (P^k, P^{k'}), (\mathcal{R}^k(t), \mathcal{R}^{k'(t)})\}$

        Compute meta-controller reward and update weights

    end

    Use parallel MCTS to search for most profitable S-E associations (Chaslot et al., 2008)

end

due to a bad placement strategy, a penalty fee proportional to the end-user payoff must be paid. The MNO's green VNF resource provisioning problem may be described as follows : We are required to find physical paths such that the added VNF instances processing delay does not exceed a given delay threshold (eq. (30)), and upon which we map VNF instances to nodes within the selected physical paths (eqs. (23)–(25)). Among the VNF instances to physical paths mapping possibilities, the optimal mapping should satisfy the resources and processing capacities (eq. (26)–(29)) while minimizing the mapping costs (eq. (32)), which also includes the environmental costs. Cost minimization objectives incentivize reusing and co-hosting VNF instances when delay threshold constraints are not violated. In our context, due to the potential penalty fee for unserved service requests, it is clear that service requests with the highest payoffs must be prioritized for provisioning unless the mapping costs exceed the payoff. Hence, when end-users service requests are received, they are first sorted and processed according to their payoff potential. After every mapping, the capabilities of nodes and links on a physical path must be updated, thus inducing continuous virtual topological changes in the network and necessitating careful consideration of the impact of mappings on future service provisioning tasks.

Despite recent advances in graph neural networks, applying deep learning algorithms on high-dimensional time-varying graphs constitutes a very challenging task, especially in real-time environments, as learned policies may fail to generalize (sometimes slightly) on different graphs compared to the ones on which they were trained. Also, the latter remains a manual and computationally expensive process according to (ZHOU et al., 2018). Consequently, we aim to decompose the time-evolving network graph into elementary subgraphs, from which consistent and efficient learning can be made. To do so, we first generate the Υ-shortest paths between physical node pairs using Floyd-Warshall's algorithm, where $\Upsilon \in \mathbb{Z}^{+}$.

Note that the nodes within a path can be obtained using path reconstruction techniques. Given the initial graph $\mathcal{G}^{(0)}$, the adjacency matrix of the graph $\mathcal{G}(\mathcal{N},\mathcal{E})$, and leveraging a pipelined 2-D block mapping parallelization (we refer interested readers to (Grama et al., 2003; Harish et Narayanan, 2007) and references therein for further details), we calculate the overall complexity as $\mathcal{O}\big(\frac{|\mathcal{N}|^3}{\mathcal{F}}\big)+\mathcal{O}(|\mathcal{N}|)$ where $|\mathcal{N}|$ denotes the number of nodes in the graph and $\mathcal{F}$ is a parallelization factor (i.e., the number of threads). The corresponding isoefficiency, a scalability measure of a parallelized algorithm, is $\Theta(\mathcal{F}^{1.5})$ (Grama et al., 2003). Recent studies demonstrated efficient implementations of parallel Floyd-Warshall for large-scale graphs using graphics processing units (GPUs), hence achieving significant computation speed-ups (Djidjev et al., 2014). Motivated by the aforementioned analysis, it is possible to incrementally update the shortest paths as VNF chains are progressively placed within the network.

Assuming a fixed number of VNF types, a VNF chain $\mathcal{G}(\mathcal{V}^k,\mathcal{E}^k)$ can be viewed as a specific permutation of VNF instances from the set or subsets of $\mathcal{V}$. Note that in practice, both the VNF chain length and the number of VNF types are relatively small (typically less than 10). Thus we reasonably assume that the pre-computation of the set $\mathcal{V}$ and its subsets are efficiently executed. We consider for each permutation a set of pre-filtered paths. Subsequently, the goal is to determine the optimal path and node placement based on the requirements of the VNF chain as illustrated in Fig. 3.5.

Let $\mathcal{H}^{\Upsilon}(\mathcal{G}^{(0)})$ be the set of Υ-shortest paths matrices, which is computed for an adjacency matrix $\mathcal{G}^{(0)}$ and a given load. The applied path pre-filtering mechanism aims to reduce the search space by removing obvious infeasible paths.

Hence, we remove from $\mathcal{H}^{\Upsilon}(\mathcal{G}^{(0)})$ any path with a latency over the required threshold or where the end-user is not the destination. This ensures the respect of constraints

Algorithm 4: DRL-VNF algorithm

Input : $\mathcal{G}^{(0)}$, $\{q^k \text{ for } k \in \mathcal{K}\}$, Υ

// Use parallel Floyd-Warshall (PFW) to compute shortest paths (Djidjev et al., 2014).

$\mathcal{H}^{\Upsilon}(\mathcal{G}^{(0)}) = PFW(\mathcal{G}^{(0)})$

for *k = 1 to* $\mathcal{K}$ **do**

  // Create a local version of $\mathcal{H}^{\Upsilon}(\mathcal{G}^{(0)})$

  $\mathcal{H}^{\Upsilon,k}(\mathcal{G}^{(0)}) = \mathcal{H}^{\Upsilon}(\mathcal{G}^{(0)})$

  // Filtering for search space reduction

  for $\upsilon = 1$ *to* Υ **do**

    if *end-points of* $\mathcal{G}(\mathcal{V}^k, \mathcal{E}^k)$ *not in* $\mathcal{H}^{\upsilon,k}(\mathcal{G}^{(0)})$ **then**

      remove $\mathcal{H}^{\upsilon,k}(\mathcal{G}^{(0)})$ from $\mathcal{H}^{\Upsilon,k}(\mathcal{G}^{(0)})$

    end

    if *delay constraint specified in eq. (30) violated* **then**

      remove $\mathcal{H}^{\upsilon,k}(\mathcal{G}^{(0)})$ from $\mathcal{H}^{\Upsilon,k}(\mathcal{G}^{(0)})$

    end

    if *all nodes in* $\mathcal{H}^{\upsilon,k}(\mathcal{G}^{(0)})$ *violates constraints from eq. (26) - (29) and (31)* **then**

      remove $\mathcal{H}^{\upsilon,k}(\mathcal{G}^{(0)})$ from $\mathcal{H}^{\Upsilon,k}(\mathcal{G}^{(0)})$

    end

  end

  Use DQN (Mnih et al., 2015) for path selection learning with rewards computed in eq. (39) using ILP.

  Select path with highest reward.

  In $\mathcal{H}^{\Upsilon}(\mathcal{G}^{(0)})$, update node and path properties (available bandwidth, resource capacities, etc) on selected path.

end

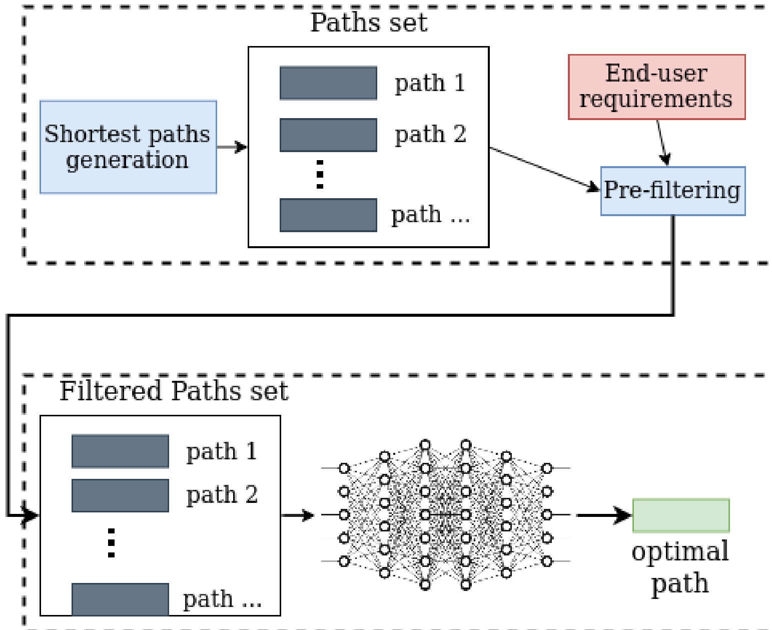

FIGURE 3.5 Overview of the DRL-VNF process.

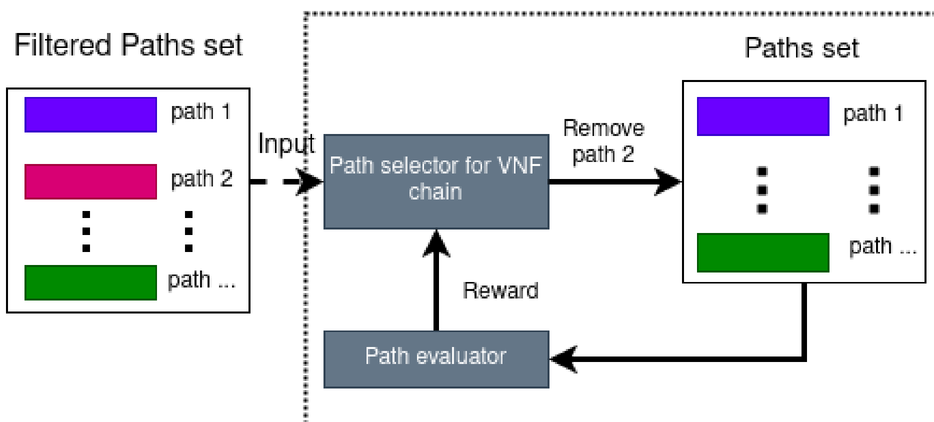

FIGURE 3.6 Overview of the interactions between path selector and path evaluator.

(24) and (30). Also, we remove a path if every node and edge within the path do not satisfy the bandwidth constraint (31) and the processing and resources limitations (26)–(29).

Once $\mathcal{H}^{\Upsilon}(\mathcal{G}^{(0)})$ is properly filtered according to end-user requirements, we leverage a framework based on two modules, namely the DRL-driven *path selector* and the *path evaluator*, as illustrated in Fig. 3.6. The path selector agent aims to determine whether a given path should be retained or removed from $\mathcal{H}^{\Upsilon}(\mathcal{G}^{(0)})$.

Each path has a state vector that encodes the feature vector of the current path along with the average feature vector of previously selected paths. In this work, we consider the following path features : 1) number of nodes, 2) average bandwidth capacity, 3) average power consumed by nodes, 4) average amount of CO_2 emitted by links, 5) remaining bandwidth between edges, 6) remaining capacity of nodes, and 7) whether or not a VNF instance has already been placed on the path for resources reuse.

We define a binary action set $\{0, 1\}$ to represent retention or removal of a path, respectively. The path selector receives, for each selected path, a delayed reward from the path evaluator. Indeed, the latter returns the VNFs-to-nodes mapping costs of the path. Based on the received rewards, the path selector updates its parameters.

Since we assume the existence of a sandbox domain for training, and that the number of nodes within a pre-filtered path can be reasonably small, then the path evaluator can efficiently leverage integer linear programming (ILP) to optimally solve the VNF-to-node placement problem (P2). We thus consider the reward as

$$\mathcal{R}(t) = \Phi - \sum_{\substack{v\in\mathcal{V}\\ n\in\mathcal{N}\\ x\in\mathcal{X}}} \sum_{i=1}^{I^v} \epsilon^v b^v_{n,x,i} + \sum_{n\in\mathcal{N}} \psi_n + \sum_{\substack{k\in\mathcal{K}\\ n\in\mathcal{N}\\ e^k_{v,v'}\in\mathcal{E}^k\\ e_{n,n'}\in\mathcal{E}}} \lambda^{k,v,v'}_{n,n'} y^{k,v,v'}_{n,n'}, \tag{39}$$

where Φ is a large number.

3.5.4 Complexity analysis

We remind the notions of sample and computational complexity. The sample complexity denotes the amount of data to collect in order to achieve adequate learning, whereas computational complexity determines the amount of computation required in using the collected data to achieve adequate learning (Kakade, 2003; Jin et al., 2018).

Model-free DRL algorithms such as DQN have multiple benefits. They are online, require less space and are expressive as specifying the value functions or policies is often more flexible than specifying the model for the environment particularly in the context of large scale networks. Unfortunately, it is empirically proven that model-free algorithms may suffer from high sample complexity (Jin et al., 2018).

In our proposed approaches, we make use of "simulators" to guide the exploration process in order to reduce the state space dimension and improve sample efficiency. Hence for a DQN-based RL algorithm, assume $\bar{S}$ the state dimension, $\bar{A}$ the number of actions and $\bar{T} = \bar{K} \times \bar{H}$ is the total number of steps with $\bar{K}$ the number of episodes and $\bar{H}$ the number of steps per episodes. The computational complexity is then $\mathcal{O}(\bar{T})$ and the regret bound is derived as $\Omega(\min\{\bar{T}, \bar{A}^{\frac{H}{2}}\})$ (Jin et al., 2018; Kakade, 2003).

3.6 Experimental evaluation

We present in this section, the experimental evaluations of our proposed approaches. Note that in order to improve our training model efficiency, we leverage data parallelism by splitting the training data into 5 partitions and feed them in parallel to 5 duplicates of the model. The latter are trained in parallel and afterwards, their results are aggregated.

3.6.1 MNO's Wireless Resource Allocation

3.6.1.1 Experimental settings

For DRL-MCTS, we consider agents with two fully-connected DNNs, $\mathfrak{J}_{\theta}^{1}$ and $\mathfrak{J}_{\theta'}^{2}$, with rectifier linear unit (ReLU) activation implemented using Tensorflow. We leverage DQN to train $\mathfrak{J}_{\theta'}^{1}$. We use hierarchical deep Q-network (H-DQN) (Kulkarni et al., 2016) with experience-replay memory as our DRL method for the meta-controller and $\mathfrak{J}_{\theta'}^{2}$. For each scenario, we experimentally determine the adequate DNN parameters, e.g., number of hidden layers. Initially, the DNN learning rate is set to 0.001 and the batch size to 1000. We leverage RMSProp with adaptive learning rate as our DNN optimization algorithm (Keras, s. d.).

In our system, we set $|\mathcal{K}| \in \{10, 20, 30\}$. For an end-user k, the required data rate $u^{k,t}$ is randomly taken from the range $[2, 8]$ bps/Hz, and the available budget $g^{k,t}$ is randomly fixed from the normalized interval $[50, 100]$. Also, the total number of subcarriers in $\mathcal{C}^t$ is set to 30, while the total number of antennas at the base station is $A = 150$. The maximum transmit power of user k is $P_{\max}^{k} = 38$ dBm (Gao et al., 2017), over a 10 MHz bandwidth. Based on (Commscope, s. d.), we set P_0 and P_π to $2W$ and $13W$, respectively. The value of ν is taken as 0.38 (Xu et L. Qiu, 2013). We assume a fixed pilot signaling fraction $\tau_c^{k,t} = 0.3$.

Considering average electricity costs, we assume $\mathfrak{C}_1$ as a linear cost function with the scalar coefficient 0.09. Moreover, we simulate user mobility by extracting the normalized distance $[0.1, 1]$ at time t between an end-user and the BS by leveraging the random waypoint mobility model (Mao, 2010). The log-normal shadowing and the path loss exponent are fixed to 8 dB and 3 dB respectively, whereas the number of transmit power levels $|\Omega^{k,t}| = 10$. In addition, agents paired by the meta-controller negotiate for 10 rounds, after which they are rewarded according to (33)–(37).

We leverage the tuple $(\tilde{c}, \tilde{a})$ output from $\mathcal{J}_\theta^1$ to evaluate the performance of the DRL-MCTS and compare them to benchmarks with the following baseline power allocation schemes : 1) Weighted minimum mean square error (WMMSE) (SHI et al., 2011), 2) Random, which generates a random value between 0 and $P_{\max}^k$, and 3) Max. power that allocates full power $P_{\max}^k$ to the agent. In case of infeasibility, we default the sum rate to zero.

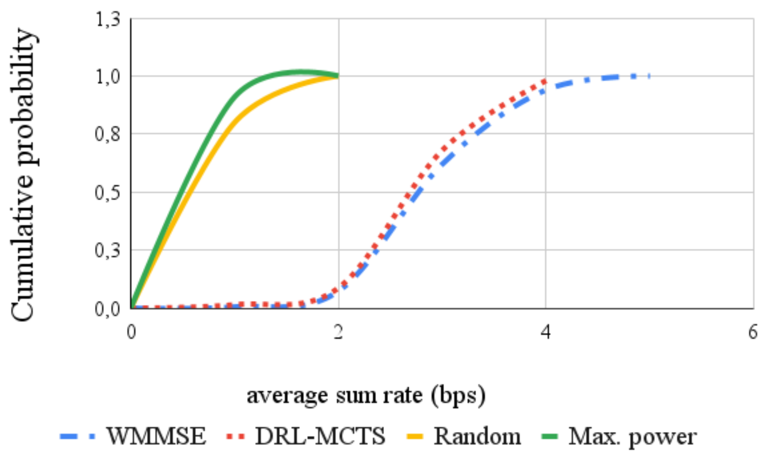

FIGURE 3.7 Sum-rate per number of users - Empirical CDF for 10 users.

3.6.1.2 Results

In Fig. 3.7, we evaluate the sum-rate obtained for each algorithm in a scenario where $|\mathcal{K}| = 10$. The cumulative distribution function (CDF) that describes the sum-rates achieved by different algorithms is built through the random generation of 500 channel realizations. It is clear that Random and Max. power resource allocations are outperformed by WMMSE and DRL-MCTS.

In this small-scale environment scenario, DRL-MCTS performance rivals with that

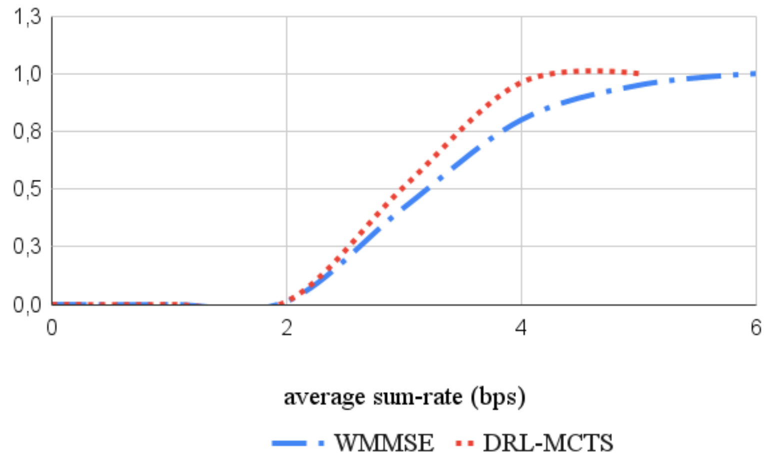

FIGURE 3.8 Sum-rate per number of users - Empirical CDF for 20 users.

of WMMSE. In large-scale systems, illustrated in Figs. 3.8–3.9 (where $|\mathcal{K}| = 20$ and $|\mathcal{K}| = 30$ respectively), we observe an increasing performance gap between DRL-MCTS and WMMSE. This is mainly due to the limitation of DRL-MCTS in accurately handling larger systems with a small deep network architecture.

Nevertheless, since WMMSE is known to incur high computational costs due to complex operations, such as high-rank matrix inversion, its use for resource allocation may be limited in dynamic network conditions (SUN et al., 2017). As such, assuming sufficient model training, DRL-MCTS is able to provide an efficient and low-complex approximation mechanism for resource allocation. As a future work, we aim to reduce the performance gap between WMMSE and DRL-MCTS through accurate hyperparameters fine-tuning.

Fig. 3.11 shows the achieved profits by each algorithm, where they are accounted only when all QoS requirements are met. Benchmark approaches, namely Random and Max. power, realize low profits due to unmet rate requirements, caused mainly by inefficient interference management. In accordance with the results of Fig. 3.7,3.8

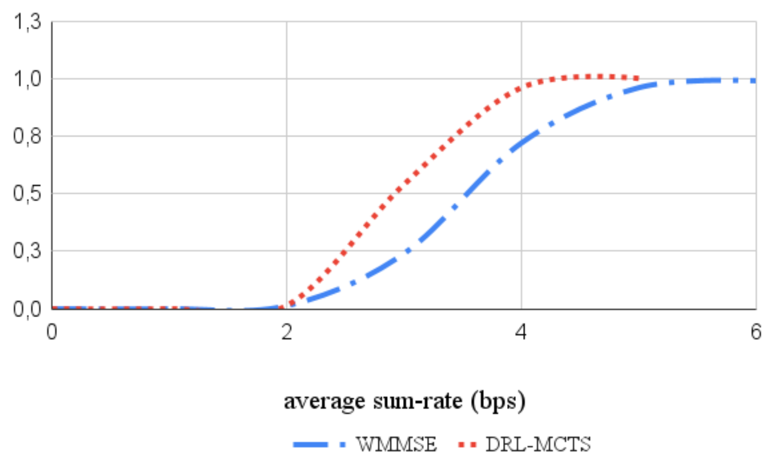

FIGURE 3.9 Sum-rate per number of users - Empirical CDF for 30 users.

and 3.9, WMMSE and DRL-MCTS achieve high profits, where the performance gap between them is widening as the number of end-users increases.

3.6.2 MNO's VNF Resource Provisioning

3.6.2.1 Experimental settings

For DRL-VNF, we adopt a path selector agent with a DNN architecture. It is composed of five fully-connected layers, and leverages ReLU activation function. Each layer is formed by 128 neurons. For our experiments, we initiate the learning rate and discount factor to 10^{-4} and 0.9, respectively.

In our system, we assume that any node is connected to 1 to 4 other nodes through communication links. These links can be seen as directed edges between nodes, where a random bandwidth capacity, selected from the set $\{500, 1000, 10000\}$ Mbps, is associated to each edge. Furthermore, we assume four types of energy sources to power the edges with the following distribution throughout the network : 60% fuel, 15% coal, 15% natural gas, and 10% solar/wind. The values retained for

FIGURE 3.10 Average cumulative rewards for DRL-MCTS and WMMSE.

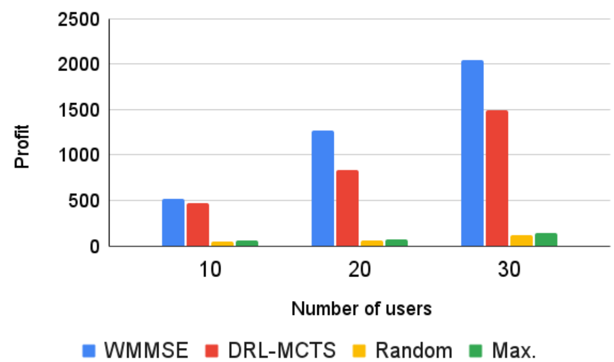

FIGURE 3.11 Profit vs. number of users.

μ_n and $\mu_{\widetilde{n}}$, are presented in Table 3.2. We further consider two types of power consumption profiles for nodes based on (ZHANG et al., 2013). The considered values are highlighted in Table 3.4 below. Also, we assume for each link a random number of routers and switches, taken from the range $[2, 5]$. The values of $P_{\widetilde{n}}^{idle}$, $P_{\widetilde{n}}^{proc}$ and $P_{\widetilde{n}}^{store}$ are taken from (VISHWANATH et al., 2014) and summarized in Table 3.5.

For each service request, we set a delay threshold requirement d^k between 50 and 200 ms. Moreover, we choose $\mathfrak{C}_2(\cdot)$ and $\mathfrak{C}_3(\cdot)$ as linear cost functions with the scalar coefficient 0.09, while the number of VNF types is set to $|\mathcal{V}| = 4$. We further set the number of VM types to $|\mathcal{X}| = 5$. Specifically, we model five types of Amazon EC2 instances : a1.medium, a1.4xlarge, m3.xlarge, t4g.2xlarge, and c6g.large. The bandwidth, vCPU, and power consumption values for each VM type are extracted from the datasets available in references (AWS, s. d.)–(Davy, s. d.[a]). We assume a number of end-user requests in the range $[30, 50]$. For each request, the end-user load l^k is selected from the range $[10, 40]$ Mbps. Each node is characterized by a vCPU capacity $\varsigma_n \in [12, 128]$. The software license cost is set to $\epsilon^v = 10$, whereas the operational cost per node (in dollars/vCPU) is in the range $[5, 10]$. Finally, the communication cost per-link (dollars/Mbps) are chosen in the interval $[1, 2]$.

Table 3.4 Node Power consumption profile (Zhang et al., 2013)

Profile	Idle (W)	Max (W)	Avg. (W)
Profile1	54.1	243	139.5
Profile2	99.3	337	194.3

Table 3.5 Power consumption measurements of switches/routers (Vishwanath et al., 2014)

Device	$P_{\tilde{n}}^{idle}$ (W)	$P_{\tilde{n}}^{proc}$ **(nJ/pkt)**	$P_{\tilde{n}}^{store}$ **(nJ/byte)**
Entreprise Ethernet Switch	36.2	40	0.28
Edge Ethernet Switch	631	1571	9.4
Metro IP router	352	1375	14.4
Edge IP router	576	1707	10.2

In our simulations, we compare DRL-VNF to the following approaches :

- **ILP** : The problem is modeled using AMPL and solved using the CPLEX solver.

- **Greedy** : In this method, filtered paths set is sorted based on the number of nodes, the average bandwidth capacity, and the average nodes' power consumption. Afterwards, the best available 10 paths are retained, where each one is iterated 10 times. In each iteration, we greedily place VNFs onto the nodes within the path aiming to minimize the costs. Once the iterations end, the path with the lowest cost is retained.

- **Random** : For added fairness, this approach still leverages the best 10 filtered path. Adversely, VNFs are randomly placed on nodes. Then, the path with the lowest cost is ultimately selected.

3.6.2.2 Results

First, we evaluate the learning performance of DRL-VNF in Fig. 3.12, given different scenarios.

The cumulative reward is presented for a fixed number of end-users requests equal to 30 and a varying number of nodes $\mathcal{N} \in \{30, 40, 50\}$. We notice that the reward convergence is strongly correlated to the size of the problem. Indeed, as the number of nodes increases, DRL-VNF converges slower.

We argue, however, that DRL-VNF is still advantageous even in the scenario with a high number of nodes. In fact, the problem size is mainly expanded by the number of added potential paths, and to a lesser degree by the presence of longer paths. Since DRL-VNF leverages paralellizable methods to generate and filter paths, we believe that it can handle efficiently the impact of an increased number of nodes. Finally, as the size of the system gets larger, higher rewards are achieved. This is due to the availability of more resources that can be used to place the VNF chains with less costs.

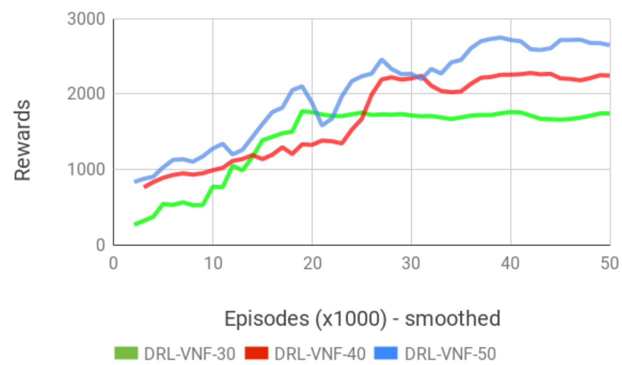

FIGURE 3.12 Average cumulative rewards vs. number of nodes.

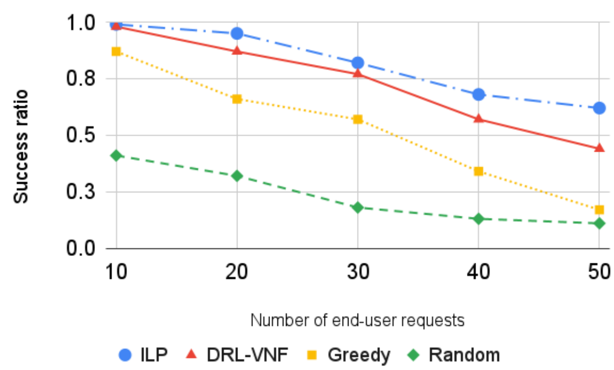

FIGURE 3.13 Success ratio.

In Fig. 3.13, we evaluate the VNF placement success ratios as a function of the number of end-user requests, given a fixed number of nodes $\mathcal{N} = 30$. A service request is successfully placed if its QoS requirements are met and every VNF in the request has been allocated to a physical node. As it can be seen, DRL-VNF achieves a near-optimal performance, unlike the greedy and random approaches that experience quick drops in their performances as the number of requests increases. Understandably, a higher number of requests implies an additional load, which requires effective management to ensure continued QoS satisfaction.

Subsequently, we present in Fig. 3.14, 3.15 and 3.16 the induced costs from VNF

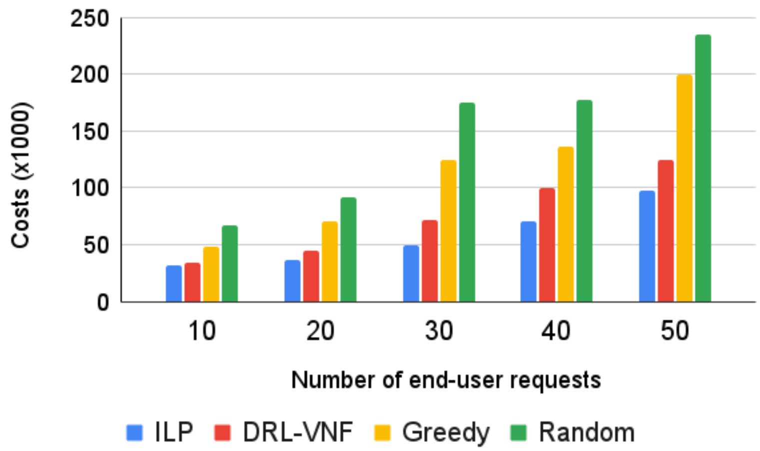

FIGURE 3.14 Server costs

placements as a function of the number of end-user requests, given $\mathcal{N} = 30$. Note that our cost evaluation includes the carbon footprint penalties $\mu_{\tilde{n}}$ and μ_n to assess the ecological impact of service request placements. In Figs. 3.14–3.15, we see that ILP and DRL-VNF have small server and communication costs, compared to greedy and random approaches.

Indeed, ILP and DRL-MCTS are able to carefully address the tradeoff between QoS satisfaction and green consideration (illustrated by the ecological impact of the energy sources that power the network) when selecting a path and its associated nodes for VNF placement.

Finally, Fig. 3.16 shows the total costs for each algorithm. We notice that the communication costs represent the largest share of the overall costs (up to 80%), and thus the total costs evolve in accordance with the communication costs of Fig. 3.15. Consequently, green VNF resource provisioning requires focused efforts to reduce communications within the core network.

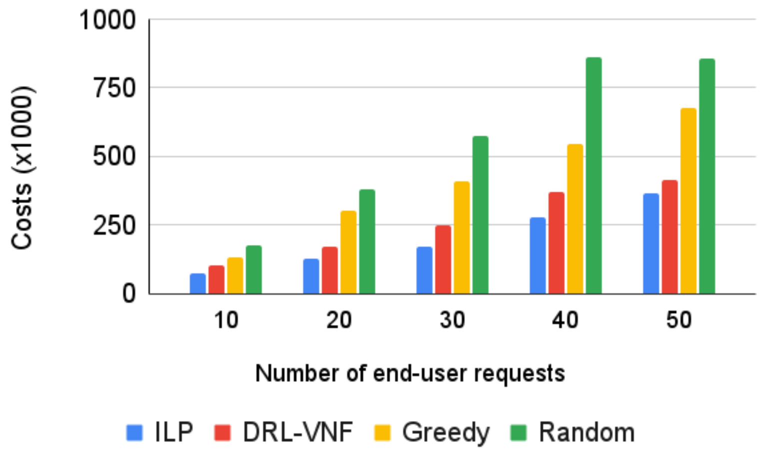

FIGURE 3.15 Communication costs

3.7 Discussion

Future network applications are expected to generate and consume massive amounts of data, requiring the implementation of various network functions such as unicast, multicast, and interactive communication mechanisms. End-to-end quality control is also needed, as is the ability to manage run-time processing before and after transmission and, in some cases, on their way to the end-user (ZHANI et ELBAKOURY, 2019).

In this context, next-generation networks strive for ultra-fast data rates, low latency, high reliability, and flexibility through novel architectural frameworks and a plethora of enabling technologies including, artificial intelligence (AI), edge intelligence, network softwarization, and data-plane programmability. These frameworks aim to enable the emergence of intelligent and autonomous services and applications. At the same time, operators are under increasing pressure to become more energy-efficient and achieve zero-carbon networks (MATSUMURA et al., 2014; AMOKRANE et al., 2015).

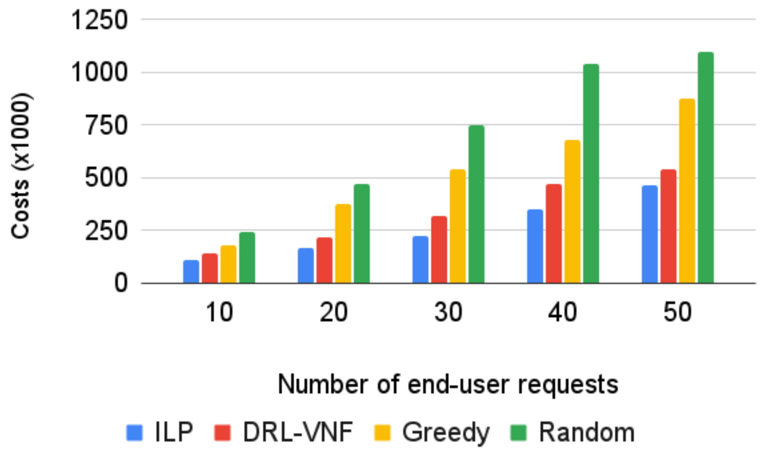

FIGURE 3.16 Total costs

To support the dynamic nature of network configuration, enable end-to-end system automation, and deliver more dependable and efficient applications to end-users, large-scale network resource provisioning necessitates cooperative and distributed learning-based management approaches. Undoubtedly, AI and ML-driven network and resource management methods, particularly DRL-based ones, are on track to become the norm. In this paper, we propose two solutions : DRL-MCTS and DRL-VNF, to resolve the resource provisioning problem in wireless and wired large-scale networks from the perspective of environmentally-conscious MNOs. Nonetheless, these solutions have limitations, thus creating opportunities for future improvements.

For instance, the rising performance gap illustrated in Fig. 3.7,3.8 and 3.9 reflects the difficulty in sustaining optimal learning in complex environments such as wireless networks. Since ML models are based on data patterns, an environment defined by mobility and interference control requirements inevitably results in model drifting (TSYMBAL, 2004). To address this issue, continuous monitoring of the model's performance is needed to trigger corrective actions such as hyperparameter optimization (HPO) or model re-training, despite the arduous parametrization required for highly dynamic environments (Hutter et al., 2019). Nonetheless, HPO or model re-training can have a substantial impact on DRL performance as noted by (Henderson et al., 2018).

As stated in Subsection V-A, this study assumes an ITU-T Y.3172-based architectural framework with an isolated ML sandbox environment for training, testing, and assessing ML models in parallel and on dedicated nodes, before deployment in real networks. It also facilitates effective hyperparameter searching, hence enhancing model performance and validity, given that hyperparameters can have dramatically varied effects across algorithms and environments, especially for DRL-based methods. Indeed, according to (Henderson et al., 2018) : "*... due to the high variance across trials and random seeds of reinforcement learning algorithms, many trials must be run with different random seeds when comparing performance. Unless random seed selection is explicitly part of the algorithm, averaging multiple runs over different random seeds gives insight into the population distribution of the algorithm performance on an environment.*"

Note, however, that HPO has some practical drawbacks, particularly for wireless networks (Hutter et al., 2019) :

- For large models or datasets, function evaluation may be prohibitively costly.
- The configuration space is complex and high dimensional.
- Due to the limited size of training datasets, direct model generalization is not possible. To circumvent this problem, an alternative consists on the usage of stable optima, where the function value around an optimum does not change for tiny perturbations of the hyperparameter values, thus resulting in a superior generalization with fresh and unknown data samples (Dai

NGUYEN et al., 2017 ; HUTTER et al., 2019).

The sandbox can also be used to predict/detect model drift and train ML models proactively, based on simulated network conditions. However, highly dynamic networks with cyclical model drift could result in extremely high energy consumption, which nullifies the energy-saving benefits of our proposed approaches.

Furthermore, model drift prediction is hampered by the complexity of uncertainty estimation for DDNs. Solving this issue is a hot research topic due to its impact on production deployment. In high-risk situations, reliable DDN uncertainty assessment may play a key role in deciding whether to abandon model training/prediction and, for example, assign decision-making to a human operator, or to opt for a more deterministic method.

Furthermore, it may improve the training and runtime efficiency of deep learning approaches (LAI et al., 2021). The main difficulty in quantifying DDN uncertainty is computational intractability, which is typically overcome with limited accuracy using Bayesian or Frequentist techniques (TAGASOVSKA et LOPEZ-PAZ, 2019). Moreover, for RL-based algorithms, the agent-environment interaction does not satisfy the classical assumptions commonly used to compute uncertainty (M. WHITE et A. M. WHITE, 2010).

According to (LAI et al., 2021), the following explains the DDN uncertainty estimate difficulty and the common use of loss function as an alternative : "*The precise prediction intervals of neural network is a difficult task to solve, and its development is closely related to uncertainty estimation. The prediction interval (PI) directly reflects the uncertainty. Many literature have proposed that the loss function can be modified to implicit learning prediction intervals ... These methods have obvious disadvantages, such as loss function can not be optimized by gradient descent-based algorithm ... or the accuracy of PIs is low. Although there are* *many confidence interval construction methods with good performance in statistical literature, these methods can not be effectively combined with deep learning.*"

3.8 Conclusion

Network operators are under increasing pressure to employ eco-friendly practices due to the substantial impact of the carbon footprint on firm value. In addition to expanding network scale and dynamism, end-users are now globally dispersed and have varying needs. Consequently, network disruptions have a greater economic impact. RL-based techniques have gained appeal due to the computing limits of heuristic algorithms and the desire to bring more autonomous decision-making into networks.

Unfortunately, the complexity and dynamism of network environments, especially wireless networks, have made designing RL agents effectively, difficult. Consequently, this study studied the prospect of leveraging parallelizable approaches while considering green objectives to induce more effective and generalized learning for green resource provisioning NP-hard problems in heterogeneous networks. Specifically, the allocation of wireless resources in radio access networks (DRL-MCTS) and the placement of VNF chains in wired core networks (DRL-VNF). For DRL-MCTS, the parallelizable MCTS algorithm is used to pre-filter end-user requests based on their estimated requirements and payoffs prior to leveraging a multi-agent hierarchical DRL framework to create optimized end-user-to-resource pairings with interference constraints primarily considered. After employing Floyd-Warshall's algorithm to generate a set of shortest paths for DRL-VNF, we selected and assessed the suitability of an end-user VNF chain request for a given network path using a DRL-based approach.

Despite the fact that our experimental evaluations demonstrated the advantages of these approaches in terms of reducing the network's environmental footprints, meeting end-users' QoS requirements, and reducing computational complexity, we must also note limitations concerning the learning drift of our DRL agents. Primarily due to the network's dynamism, it might be impossible to generate sample datasets in advance, and learning convergence might be time-consuming. In addition, solutions based on a sandbox framework may result in excessive energy consumption.

Future research will therefore focus on narrowing the learning performance gap using techniques like continuous monitoring of learning model performances with HPO and periodic model retraining.

REFERENCES

AHN, Chang Wook et Rudrapatna S RAMAKRISHNA (2004). "QoS provisioning dynamic connection-admission control for multimedia wireless networks using a Hopfield neural network". In : *IEEE Trans. Veh. Tech.* 53.1, p. 106-117.

AMOKRANE, A. et al. (2015). "Greenslater : On Satisfying Green SLAs in Distributed Clouds". In : *IEEE Trans. Netw. and Serv. Management* 12.3, p. 363-376. ISSN : 1932-4537. DOI : 10.1109/tnsm.2015.2440423.

ANDREWS, J. G. et al. (2014). "What Will 5G Be ?" In : *IEEE J. Sel. Areas in Commun.* 32.6, p. 1065-1082. ISSN : 0733-8716. DOI : 10.1109/jsac.2014.2328098.

AWS, Amazon (s. d.). *Amazon EC2 Instance Types*. [Online ; Accessed on Nov. 15, 2021]. URL : https://aws.amazon.com/ec2/instance-types/.

BENGIO, Yoshua, Andrea LODI et Antoine PROUVOST (2018). "Machine Learning for Combinatorial Optimization : A Methodological Tour d'Horizon". In : *arXiv preprint arXiv :1811.06128*.

BLENK, Andreas et al. (2016). "Boost online virtual network embedding : Using neural networks for admission control". In : *Proc. IEEE 12th Int. Conf. Netw. and Serv. Management (CNSM)*, p. 10-18.

Boutaba, Raouf et al. (2018). “A comprehensive survey on machine learning for networking : Evolution, applications and research opportunities”. In : *J. Internet Serv. and Apps.* 9.1, p. 16.

Browne, Cameron B et al. (2012). “A survey of monte carlo tree search methods”. In : *IEEE Trans. Computa. Intellig. and AI in Games* 4.1, p. 1-43.

Chaslot, Guillaume M. J. -B., Mark H. M. Winands et H. Jaap van den Herik (2008). “Parallel Monte-Carlo tree search”. In : *International Conference on Computers and Games*. Sous la dir. de H. Jaap van den Herik et al. Springer Berlin Heidelberg, p. 60-71. isbn : 978-3-540-87608-3.

Cohen, Reuven, Liran Katzir et Danny Raz (2006). “An efficient approximation for the generalized assignment problem”. In : *Info. Process. Lett.* 100.4, p. 162-166.

Commscope (s. d.). *Base Station Antennas - Reference data*. [Online ; Accessed on Nov. 15, 2021]. url : https://www.commscope.com/product-type/antennas/base-station-antennas-equipment/base-station-antennas/.

Dai Nguyen, Thanh et al. (2017). “Stable bayesian optimization”. In : *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, p. 578-591.

Davy, Benjamin (s. d.[a]). *AWS EC2 Carbon Footprint Dataset*. [Online ; Accessed on Nov. 15, 2021]. url : https://docs.google.com/spreadsheets/d/1DqYgQnEDLQVQm5acMAhLgHLD8xXCG9BIrk-%5C%5FNv6jF3k/.

– (s. d.[b]). *Estimating AWS EC2 Instances Power Consumption*. [Online ; Accessed on Nov. 15, 2021]. url : https://medium.com/teads-engineering/estimating-aws-ec2-instances-power-consumption-c9745e347959.

DJIDJEV, Hristo et al. (2014). “Efficient multi-GPU computation of all-pairs shortest paths”. In : *Proc. IEEE 28th Int. Parallel and Distrib. Process. Symp.* P. 360-369.

DUTREILH, Xavier et al. (2011). “Using reinforcement learning for autonomic resource allocation in clouds : Towards a fully automated workflow”. In : *Proc. 7th Int. Conf. Automatic Autonomous Syst. (ICAS)*, p. 67-74.

FU, F. et U. C. KOZAT (2013). “Stochastic Game for Wireless Network Virtualization”. In : *IEEE/ACM Trans. Network.* 21.1, p. 84-97. ISSN : 1063-6692. DOI : 10.1109/tnet.2012.2190419.

GAO, Yuan et al. (2017). “Energy Oriented Resource Allocation in Heterogeneous 5G Networks”. In : *International Conference in Communications, Signal Processing, and Systems.* Springer, p. 208-217.

GOODWIN, Roger L (2019). “Efficiently Computing the Power Set in a Parallel Environment”. In : *Proc. IEEE Int. Parallel and Distrib. Process. Symp. Wrkshps. (IPDPSW)*, p. 573-579.

GRAMA, Ananth et al. (2003). *Introduction to parallel computing.* Pearson Education.

HAN, T. et N. ANSARI (2013). “On Optimizing Green Energy Utilization for Cellular Networks with Hybrid Energy Supplies”. In : *IEEE Trans. Wireless Commun.* 12.8, p. 3872-3882. ISSN : 1536-1276. DOI : 10.1109/tcomm.2013.051313.121249.

– (2014). “Powering mobile networks with green energy”. In : *IEEE Wireless Commun.* 21.1, p. 90-96. ISSN : 1536-1284. DOI : 10.1109/mwc.2014.6757901.

HARISH, Pawan et PJ NARAYANAN (2007). “Accelerating large graph algorithms on the GPU using CUDA”. In : *Int. Conf. High-perform. Comput.* Springer, p. 197-208.

HASAN, Ziaul, Hamidreza BOOSTANIMEHR et Vijay K BHARGAVA (2011). “Green cellular networks : A survey, some research issues and challenges”. In : *IEEE Communications surveys & tutorials* 13.4, p. 524-540.

HENDERSON, Peter et al. (2018). “Deep reinforcement learning that matters”. In : *Proceedings of the AAAI conference on artificial intelligence*. T. 32. 1.

HOSSAIN, E. et M. HASAN (2015). “5G Cellular : Key Enabling Technologies and Research Challenges”. In : *IEEE Instrumentation Measurement Mag.* 18.3, p. 11-21. ISSN : 1094-6969. DOI : 10.1109/mim.2015.7108393.

HUTTER, Frank, Lars KOTTHOFF et Joaquin VANSCHOREN (2019). *Automated machine learning : methods, systems, challenges*. Springer Nature.

JIANG, Menglan, Massimo CONDOLUCI et Toktam MAHMOODI (2016). “Network slicing management & prioritization in 5G mobile systems”. In : *Proc. VDE 22nd Europ. Wireless Conf.* P. 1-6.

JIN, Junqi et al. (2018). “Real-Time Bidding with Multi-Agent Reinforcement Learning in Display Advertising”. In : Cikm ’18, p. 2193-2201. DOI : 10.1145/3269206.3272021. URL : http://doi.acm.org/10.1145/3269206.3272021.

KAKADE, Sham Machandranath (2003). *On the sample complexity of reinforcement learning*. University of London, University College London (United Kingdom).

Katoh, Naoki et Toshihide Ibaraki (1998). “Resource allocation problems”. In : *Handbook of combinatorial optimization*. Springer, p. 905-1006.

Keras (s. d.). *RMSprop*. [Online ; Accessed on Nov. 15, 2021]. url : https://keras.io/api/optimizers/rmsprop/.

Khosravi, Atefeh, Saurabh Kumar Garg et Rajkumar Buyya (2013). “Energy and carbon-efficient placement of virtual machines in distributed cloud data centers”. In : *European Conference on Parallel Processing*. Springer, p. 317-328.

Kiiski, Annukka (2006). “Impacts of MVNOs on Mobile Data Service Market”. In : *Proc. 17th European Regional ITS Conf.*

Kulkarni, Tejas D et al. (2016). “Hierarchical deep reinforcement learning : Integrating temporal abstraction and intrinsic motivation”. In : *Proc. Adv. Neural Info. Process. Syst.* 29, p. 3675-3683.

Kumar, Saurabh et al. (2017). “Federated control with hierarchical multi-agent deep reinforcement learning”. In : *arXiv preprint arXiv :1712.08266*.

Lai, Yuandu et al. (2021). “Exploring Uncertainty in Deep Learning for Construction of Prediction Intervals”. In : *arXiv preprint arXiv :2104.12953*.

Li, Chang, Jun Zhang et Khaled B Letaief (2014). “Throughput and energy efficiency analysis of small cell networks with multi-antenna base stations”. In : *IEEE Transactions on Wireless Communications* 13.5, p. 2505-2517.

Li, Rongpeng et al. (2018). “Deep reinforcement learning for resource management in network slicing”. In : *IEEE Access* 6, p. 74429-74441.

Mao, Shiwen (2010). “Chapter 8 - Fundamentals of communication networks”. In : *Cognitive Radio Communications and Networks*. Sous la dir. d’Alexander M. Wyglinski, Maziar Nekovee et Y. Thomas Hou. Oxford : Academic Press, p. 201-234. isbn : 978-0-12-374715-0. doi : https://doi.org/10.1016/B978-0-12-374715-0.00008-3. url : https://www.sciencedirect.com/science/article/pii/B9780123747150000083.

Marbach, Peter, Oliver Mihatsch et John N Tsitsiklis (2000). “Call admission control and routing in integrated services networks using neuro-dynamic programming”. In : *IEEE J. Select. Areas Commun.* 18.2, p. 197-208.

Matsumura, Ella Mae, Rachna Prakash et Sandra C Vera-Muñoz (2014). “Firm-value effects of carbon emissions and carbon disclosures”. In : *The Account. Review* 89.2, p. 695-724.

Mignanti, Silvano, Alessandro Di Giorgio et Vincenzo Suraci (2009). “A model based RL admission control algorithm for next generation networks”. In : *Proc. IEEE 8th Int. Conf. Netw.* P. 191-196.

Mijumbi, Rashid et al. (2014). “Design and evaluation of learning algorithms for dynamic resource management in virtual networks”. In : *Proc. IEEE Netw. Op. and Management Symp. (NOMS)*, p. 1-9.

Miyazawa, Takaya, Ved P Kafle et Hiroaki Harai (2017). “Reinforcement learning based dynamic resource migration for virtual networks”. In : *Proc. IFIP/IEEE Symp. Integrat. Netw. and Serv. Management (IM)*, p. 428-434.

Mnih, Volodymyr et al. (2015). “Human-level Control Through Deep Reinforcement Learning”. In : *Nature* 518.7540, p. 529.

Möbius, Christoph, Waltenegus Dargie et Alexander Schill (2013). “Power consumption estimation models for processors, virtual machines, and servers”. In : *IEEE Transactions on Parallel and Distributed Systems* 25.6, p. 1600-1614.

Nachum, Ofir et al. (2018). “Data-efficient hierarchical reinforcement learning”. In : *Proc. Adv. in Neural Info. Process. Syst.* P. 3303-3313.

Piamrat, Kandaraj et al. (2008). “QoE-aware admission control for multimedia applications in IEEE 802.11 wireless networks”. In : *Proc. IEEE 68th Veh. Tech. Conf. (VTC)*, p. 1-5.

Pietrabissa, Antonio et al. (2017). “An approximate dynamic programming approach to resource management in multi-cloud scenarios”. In : *Int. J. Control* 90.3, p. 492-503.

Ricciardi, Sergio et al. (2013). “Modeling energy consumption in next-generation wireless access-over-WDM networks with hybrid power sources”. In : *Math. and Comput. Model.* 58.5–6, p. 1389-1404. issn : 0895-7177. doi : http://dx.doi.org/10.1016/j.mcm.2012.12.004.

Shi, Qingjiang et al. (2011). “An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel”. In : *IEEE Trans. Sig. Process.* 59.9, p. 4331-4340.

Sun, Haoran et al. (2017). “Learning to optimize : Training deep neural networks for wireless resource management”. In : *Proc. IEEE 18th Int. Wrkshp. Sig. Process. Adv. Wireless Commun. (SPAWC)*, p. 1-6.

TAGASOVSKA, Natasa et David LOPEZ-PAZ (2019). “Single-model uncertainties for deep learning”. In : *Advances in Neural Information Processing Systems* 32, p. 6417-6428.

TESAURO, Gerald et al. (2005). “Online resource allocation using decompositional reinforcement learning”. In : *Aaai*. T. 5, p. 886-891.

THO LE-NGOC Rajesh Dawadi, Saeedeh Parsaeefard et Mahsa DERAKHSHANI (2017). *Virtualized Wireless Networks : User Association and Resource Allocation*. First. SpringerLink.

TONG, Hui et Timothy X BROWN (2000). “Adaptive call admission control under quality of service constraints : A reinforcement learning solution”. In : *IEEE J. Select. Areas Commun.* 18.2, p. 209-221.

TSYMBAL, Alexey (2004). “The problem of concept drift : definitions and related work”. In : *Computer Science Department, Trinity College Dublin* 106.2, p. 58.

UNION-T, International Telecommunication (2020). *Architectural framework for machine learning in future networks including IMT-2020*. URL : https://www.itu.int/rec/T-REC-Y.3172-201906-I/en.

VISHWANATH, Arun et al. (2014). “Modeling energy consumption in high-capacity routers and switches”. In : *IEEE Journal on Selected Areas in Communications* 32.8, p. 1524-1532.

WANG, Jie et Yangfan QIU (2013). “A new call admission control strategy for LTE femtocell networks”. In : *Proc. 2nd Int. Conf. Adv. Comput. Sci. Eng. (CSE)*. Atlantis Press.

WHITE, Martha et Adam M WHITE (2010). "Interval Estimation for Reinforcement-Learning Algorithms in Continuous-State Domains." In : *Nips*, p. 2433-2441.

XU, Jie et Ling QIU (2013). "Energy efficiency optimization for MIMO broadcast channels". In : *IEEE Transactions on Wireless Communications* 12.2, p. 690-701.

ZHANG, Yang et al. (2013). "Auction Approaches for Resource Allocation in Wireless Systems : A Survey". In : t. 15. 3. Ieee, p. 1020-1041.

ZHANI, Mohamed Faten et Hesham ELBAKOURY (2019). "FlexNGIA : A flexible Internet architecture for the next-generation tactile Internet". In : *arXiv preprint arXiv :1905.07137*.

ZHOU, Jie et al. (2018). "Graph neural networks : A review of methods and applications". In : *arXiv preprint arXiv :1812.08434*.

ZHU, K. et E. HOSSAIN (2016). "Virtualization of 5G Cellular Networks as a Hierarchical Combinatorial Auction". In : *IEEE Trans. Mob. Comput.* 15.10, p. 2640-2654.

CHAPITRE IV

ARTICLE 4 - MARKET DRIVEN MULTI-DOMAIN NETWORK SERVICE ORCHESTRATION IN 5G NETWORKS

Mouhamad Dieye
Université du Québec à Montréal
Montréal (Québec), Canada

Wael Jaafar, Professeur
École de Technologie Supérieure
Montréal (Québec), Canada

Halima Elbiaze, Professeur titulaire
Université du Québec à Montréal
Montréal (Québec), Canada

Roch Glitho, Professeur titulaire
Concordia University
Montréal (Québec), Canada

AVANT-PROPOS À L'ARTICLE 4

Le quatrième article de la thèse vise à explorer l'allocation de ressources virtuelles dans les réseaux multi-domaines, en se basant sur des mécanismes de marché afin de garantir conjointement le respect des exigences de qualité de service ainsi que la compétitivité entre opérateurs. Cet article examine l'influence de la profitabilité sur la collaboration et la prise de décision concernant l'orchestration des ressources.

Cet article a été accepté pour une publication dans la revue (journal) scientifique **IEEE Journal on Selected Areas in Communications (IEEE JSAC), Special issue on Softwarization & Enablers, Volume 38, Issue 7, July 2020.**. IEEE TNSM est classé Q1 dans la catégorie "Computer Networks and Communications" (SJR 2021) avec un facteur d'impact sur 5 ans de 14.2 (H-Index : 242).

- Dieye, M., Jaafar, W., Elbiaze, H., & Glitho, R. H. (2020). **Market driven multidomain network service orchestration in 5g networks**. *IEEE Journal on Selected Areas in Communications, 38(7), 1417-1431.* DOI: 10.1109/JSAC.2020.2986692

Les articles IEEE JSAC sont examinés par des pairs conformément aux exigences énoncées dans le manuel d'opérations de l'IEEE PSPB (sections 8.2.1.C & 8.2.2.A). Chaque article publié a été révisé par un minimum de deux réviseurs indépendants utilisant un processus de révision par les pairs en simple aveugle, où l'identité des réviseurs n'est pas connue des auteurs, mais les réviseurs connaissent l'identité des auteurs. Les articles sont soumis à un contrôle de plagiat avant d'être acceptés.

ABSTRACT

The advent of a new breed of enhanced multimedia services has put network operators into a position where they must support innovative services while ensuring both end-to-end Quality of Service requirements and profitability.

Recently, Network Function Virtualization (NFV) has been touted as a cost-effective underlying technology in 5G networks to efficiently provision novel services. These NFV-based services have been increasingly associated with multi-domain networks.

However, several orchestration issues, linked to cross-domain interactions and emphasized by the heterogeneity of underlying technologies and administrative authorities, present an important challenge.

In this paper, we tackle the cross-domain interaction issue by proposing an intelligent and profitable auction-based approach to allow inter-domains resource allocation.

Keywords: End-to-end Quality of Service requirements, cross-domain interactions, NFV-based services, Network Function Virtualization, profitability, network operators, 5G networks, market driven multi-domains, Network Service orchestration, profitable auction-based approach, multi-domain, resource orchestration, resource allocation, deep reinforcement learning, collusion, coopetition, competition, quality of service.

4.1 Introduction

Recent years have seen the emergence of services such as tactile internet, multi-player gaming, etc. characterized by their innovativeness and stringent quality of service (QoS) requirements, such as ultra low latency (Guerzoni et al., 2017; Osseiran et al., 2014). 5G networks have been envisioned to provide flexible and cost-effective service provisioning through several enabling technologies, while ensuring profitability (Akyildiz et al., 2016; Alliance, 2015).

One of the major goals of 5G networks consist of providing ultra reliable end-to-end service delivery through satisfaction of end-to-end QoS requirements (Itu, 2015). In this regard, Network Function Virtualization (NFV) has been advocated as an effective service provisioning model for network operators to efficiently create, deploy and manage services, thereby allowing efficient and flexible utilization of their limited infrastructures while decreasing capital and operational expenditures.

Through NFV, a network service (NS) is decomposed into a set of chained virtual network functions (VNFs) running atop of a virtualized infrastructure called Network Function Virtualization Infrastructure (NFVI). In order to model and deploy a service, an efficient approach is required to determine the composition/decomposition of the service, and how to automate selection and control (e.g. VNF creation, placement, migration, monitoring, etc.) of underlying physical or virtual resources and services with certain objectives (e.g. QoS requirements, costs, etc.). This process is referred to as network service orchestration (NSO) and provisioning (Mijumbi et al., 2016; De Sousa et al., 2018).

Recently, end-to-end service provisioning has been increasingly associated with multi-domain networks where the resources hosting VNFs are owned and controlled by multiple independent network operators located in different geographical locations (Guerzoni et al., 2017; Rosa et al., 2015). Nonetheless, guaranteeing an end-to-end service delivery with stringent QoS requirements is a challenging task due to several issues related to cross-domain interactions, such as the heterogeneity of underlying 5G technologies including fog and edge computing platforms, various radio access technologies, and different transport and core networks. In addition, as the network infrastructure can be owned and managed by different administrative entities, it also requires taking into account different business models and orchestration approaches (Katsalis, Nikaein et al., 2016; Rosa et al., 2015; De Sousa et al., 2018).

Indeed, a disruption in one resource/domain can carry a detrimental effect to the overall satisfaction of end-to-end QoS requirements. Yet, as of today, no clear consensus on multi-domain service provisioning has been achieved. Accordingly, several studies have noted the lack of support of service provisioning by the ETSI MANO, in the context of multi-domain networks architecture (Katsalis, Nikaein et al., 2016; De Sousa et al., 2018).

Thus, new profitable cooperation and effective service orchestration mechanisms are urgently needed to leverage the resources offered by operators to support end-users' QoS and ensure end-to-end service provisioning. Furthermore, as end-user demands and networks themselves (e.g nodes, routes, etc.) are characterized by high dynamism and heterogeneity, future NSO mechanisms should autonomously adapt to various network environments, topologies, and sizes. Finally, given the market competition from third-party operators, e.g. virtual operators and service providers, operators' profitability has to be taken into account in the resource orchestration process.

Albeit mostly in single domain networks, several approaches have been explored in the literature in the realm of service provisioning related problems, e.g. resource allocation, slicing, scheduling, etc. However, a number of challenges remain to be effectively tackled in the context of large scale 5G multi-domain networks where emphasis is put on automated network resource sharing, multi-tenancy and cooperative resource provisioning.

In this paper, we focus on the multi-domain network provisioning problem, by taking into consideration automated resource sharing and cooperative resource provisioning. For that, we propose an auction-based NFV orchestration method, where inter-operator interactions and exchange of network resources for service orchestration is translated into buyer/seller transactions and market dynamics (Habiba et Hossain, 2018 ; Zhang et al., 2013).

Indeed, auctions as a resource allocation and orchestration mechanism have the advantage of being economically efficient through automatic discovery of service chain market value and assignment of limited resources to the bidders who value them the most. Furthermore, it provides compelling opportunities to modern telco actors, namely Infrastructure Providers/Mobile Network Operators (InPs/MNOs) and Service Providers/Mobile Virtual Network Operators (SPs/MVNOs).

Briefly, InPs own physical/virtual infrastructure and resources, that are leased to different SPs. The SPs are operators who provide 5G services and yet do not have the required physical infrastructure and resources to meet the demands and requirements of their subscribers. Thus, they rent resources from InPs and, if unused, can lease those resources to other SPs. This business model allows SPs to acquire resources to satisfy their end-users QoS requirements and enables them to reduce new service deployment costs, while InPs increase their profits by leasing their unused network resources within a marketplace.

Finally, given a multi-InP marketplace, interested SPs can switch between InPs when market dynamics are not favourable (Habiba et Hossain, 2018 ; Palattella et al., 2016 ; Zhu et Hossain, 2016 ; Jiang et al., 2017).

In practice, self-management through an efficient auction-based framework for large scale and dynamic networks remains a challenging task both from the auctioneer's and bidders' perspective respectively. This is due to the increasing complexity, resulting from the exponential bid space growth, e.g. increased/decreased competition, inflation, new market entry, etc., as in-market resources and market dynamics fluctuate over time (Brero et al., 2018 ; Ausubel et Baranov, 2017).

Coincidentally, the advent in recent years of deep reinforcement learning (DRL) algorithms has enabled solving decision-making problems, previously considered intractable due to high-dimensional state and action spaces. In short, DRL algorithms can be exploited to produce fully autonomous agents (SPs) capable of interacting with their environment (market conditions) to learn optimal behaviours (bidding strategies and resource requests), improving over time through trial and error.

The main contributions of this paper are summarized as follows :

1. First, we emphasize the importance of multi-domain interactions for end-to-end service provisioning, through the Massive Multiplayer Online Gaming (MMOG) use case.

2. Then, we model a market-driven multi-domain interaction framework for resource allocation between an InP and several candidate SPs. We formulate the associated resource allocation problem, where the InP and SPs aim at maximizing their profits by selling/buying in-market resources. We define different marketplace environments, where SPs can compete and/or cooperate in the bidding process, and the InP can decide whether to apply or not fairness among SPs.

3. Due to the complexity of the system with several SPs, we propose a distributed multi-agent deep reinforcement learning approach, where we equip each SP with a learning agent able to perceive the environment and takes strategic actions to win auctions, hence satisfies its QoS requirements and increases its profit. Moreover, agents of different SPs may exchange information in cooperation scenarios to improve their mutual profits.

4. Through experimental setups, we trained SPs' agents and evaluated the performances of InP and SPs in different marketplace scenarios. It has been shown that a fully competitive market (no cooperation) is the most profitable for InP. In contrast, a cooperative market is in average the most lucrative for SPs. Moreover, double-deep-Q-learning agents outperform other learning-based and learning-free agents in terms of SPs' profits. Finally, impact of some parameters are emphasized.

The rest of the paper is organized as follows. Section II presents a motivating use case, highlighting multi-domain interactions. The following Section III details the related work. Section IV provides a system model of a market-driven multi-domain interaction framework. Then, the associated problem is formulated in Section V. Section VI describes the investigated auctions scenarios, while section VII presents our proposed solutions. In Section VIII, experimental results are presented and discussed. Finally, Section IX closes the paper.

4.2 Massive Multiplayer Online Game use case

4.2.1 Overview

Massive Multiplayer Online Gaming (MMOG) has increasingly gained popularity to become one of the most lucrative industries with millions of subscribers worldwide with an estimated 55% of Internet users today to be also online gamers (Lee et K.-T. Chen, 2010). MMOG is described as a gaming environment where a large number of active concurrent players gather around a shared virtual environment. It is also characterized by its real time requirements to ensure an immersive game-play experience. Unfortunately, as the number of connected subscribers increases, the resource load generated by a game server may induce a degradation of game-play experience making the game unplayable to players who eventually quit (Prodan et Iosup, 2016).

Typically, to ensure the QoS requirements of its widely distributed subscribers at all times, MMOG operators maintain a rigid multi-server distributed infrastructure with -often- over-provisioned computational and network capabilities. Such design increases operational costs due to potential resource under-utilization and/or capacity shortages caused by sudden peaks in demands (Nae et al., 2011). Alternatively, cloud based MMOG has been increasingly advocated to solve some of these issues (Prodan et Iosup, 2016).

Also, in order to serve several concurrent players into a unique game session, a current practice is to *parallelize* the game server code and distribute the load across multiple resources, through techniques such as *zoning*, in which the game world is geographically partitioned into disjoint zones that can be assigned to different autonomous computing resources (Prodan et Iosup, 2016).

Similar to (Prodan et Iosup, 2016), we consider an ecosystem consisting of players connected through game providers to game operators who ensure autonomous execution of MMOG sessions by provisioning physical/virtual resources from the worldwide distributed resource providers. Resource providers host the game servers and may serve several independent game operators simultaneously. In this ecosystem, as an incentive to respect the QoS requirements, penalization is due

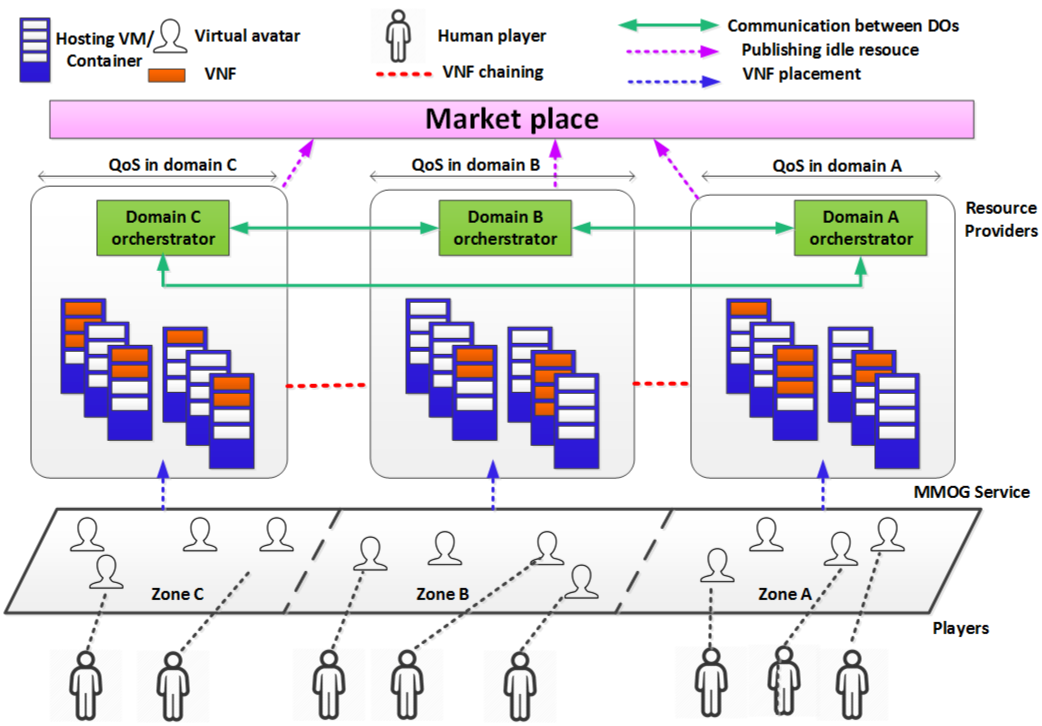

FIGURE 4.1 MMOG scenario.

for any SLA (Service Level Agreement) violation.

Further consider the scenario of Fig. 4.1, where a game operator is in charge of the zoning MMOG slice to which end-users are connected. The game operator is tasked with ensuring QoS requirements regardless of users' demands variation and locations.

To this end, it can strategically deploy a game server service as a VNF chain, with each VNF handling a specific game-play task such as command emulation, character management, game physics/logic responses, graphics rendering, etc. It is worth noting that the VNF chain composition may vary, depending on the capabilities of devices (mobile, desktop, etc.) used to play the game. Indeed, some end-users may not have, for instance, the required codecs to output the gameplay, thus requiring transcoding. Also, other end-users, with limited Internet bandwidth, wouldn't obtain full resolution graphics rendering.

Hence, in order to provide the service with regards to specific QoS requirements, the game operator must strategically embed VNFs by leveraging the virtual/physical resources offered by independent resource operators.

4.2.2 Requirements

Achieving optimal placement of VNFs remains a challenging issue in the MMOG scenario due to the inherent goals associated. First, the game play service must be delivered to the end-users with strict end-to-end QoS requirements, notably end-to-end low latency, as it constitutes the leading reason why end-users drop a game. Thus end-to-end latency is vital from the profitability perspective (Prodan et Iosup, 2016). Second, the chosen service provisioning strategy must be adaptable to both small and large scale scenarios, as the number of players involved is large and time-varying. Third, game operators are to be considered with limited budgets and an emphasis is put on profit maximization, e.g reducing the amount of penalties to pay for SLA violations.

In our scenario, a possible way to deploy a game server service by the game operator, within a single domain, is to leverage a single resource operator to host all required VNFs. However, given the multi-dimensional nature of QoS requirements and the time-varying users' behaviours, it is very hard to provision and ensure timely service delivery, without over-provisioning resources (Buyya et al., 2010). Worse, as there are intermediary networks between the end-users and the resource operators hosting a game server service, ensuring the QoS requirements within a single domain does not guarantee satisfaction of end-to-end QoS requirements.

Alternatively, a game operator could deploy its game server service by efficiently and seamlessly leveraging offered resources by multiple resource operators. Hence, subscribers' heterogeneous QoS requirements can be cooperatively satisfied. Moreover, embedding VNFs closer to customers, as in an edge/fog computing design, would achieve low latency services and simplify delegated monitoring and maintenance tasks (Rosa et al., 2015).

Also, as each type of domains has its own strengths and weaknesses in terms of network characteristics, e.g. bandwidth, latency, and computing power, embedding VNFs across multiple domains may enhance service performances and cost efficiency.

In sum, a game operator can chain VNFs from different operators into a single functional service that needs to operate over heterogeneous network infrastructures owned by different providers. However, a fundamental step to this is to provide a profitable and viable cooperation mechanism to provision and manage VNFs (and services), thereby enabling multi-domain VNF chaining.

In the proposed scenario, we assume resource operators to be available through a market where they publish their resources. These resources can be leveraged by game operators to compose their VNF chains and services. Obviously, the proposed service provisioning mechanism requires not only to be efficient in large scale environments, but also to adapt to varying network and market conditions, such as available resources, workload variation, increased competition, price inflation, etc.

4.3 Related Work

Network operators rely on paradigms such as Software-Defined Networks (SDN) and NFV to efficiently create, deploy and manage services to serve their subscribers. By decoupling the physical and logical network layers, an operator is able to model a service from end-to-end by abstracting and automating the control of physical and virtual resources. This orchestration process comes down to an automatic selection and control of multiple resources, services and controllers, in order to satisfy predefined objectives (De Sousa et al., 2018).

4.3.1 SDN-based Orchestration

Leveraging SDN technology to separate the control and data plane has brought to light several challenges especially for multi-domain systems. A typically important issue consists of determining the placement of SDN controllers in order to achieve optimized control over physical/virtual resources. In this matter, several papers have investigated SDN controller placement in a single-domain network considering various objectives, such as minimizing resource utilization, overall operational costs or network delay (Bari et al., 2013 ; Lange et al., 2015 ; G. Wang et al., 2017).

For multi-domain networks, (Katsalis, Rofoee et al., 2017) investigated SDN implementations over converged wireless-optical networks and proposed abstractions and virtualization techniques to integrate virtual wireless and optical resources in a framework called CONTENT. Authors in (J. Wang et al., 2016) proposed a scalable SDN architecture for multi-domain and multi-vendor networks by implementing a coordinator controller to enable cooperation among different SDN administrative domains.

In (Figueira et Krishnan, 2015), the authors summarized the challenges of SDN multi-domain orchestration and control before proposing a hierarchical SDN framework that aims at simplifying network control and orchestration.

4.3.2 NFV-based Orchestration

From the NFV perspective, numerous research studies ((Mijumbi et al., 2016 ; Bhamare et al., 2016) and references therein) investigated service provisioning through optimizing the VNF chain instances placement over virtual/physical resources of a network (Mijumbi et al., 2016; De Sousa et al., 2018; Abu-Lebdeh et al., 2017). Whereas, ETSI has concentrated its efforts in the standardization of NFV management and orchestration. Indeed, ETSI set the standard's requirements, specifications and architectural framework, called NFV-MANO (NFV Management and Orchestration) (Etsi, 2014). Its role, mainly through the NFVO (NFV Orchestrator) and the VNFM (VNF Manager), is to enable VNF operations, e.g. orchestration, lifecycle management, across computing resources within a single administrative network domain.

Besides VNF placement and resources orchestration, the management of lifecycle operations, such as creation, monitoring, release, and dependencies, and the interactions with other components, e.g. marketplaces to acquire new resources and OSS/BSS (Operation Support System/Business Support Systems), are also important to satisfy end-to-end QoS and realize profitability.

Hence, within the NFV-MANO framework and in the context of multi-domain NFV systems, the number and locations of NFVO and VNFM functional blocks are critical to the system's overall scalability and performances. For instance, given that in order to properly satisfy end-to-end latency requirements, latencies between VNFs, Element Managers (EMs), Virtualized Infrastructure Managers (VIMs), NFVOs and cooperating VNFMs, must all be taken into account. Apart from (Abu-Lebdeh, 2018; Abu-Lebdeh et al., 2017), very few have investigated the MANO functional blocks placement problem.

4.3.2.1 NFV Single-Domain vs. Multi-Domain Orchestration

In order to provision and orchestrate network services with end-to-end QoS requirements, a service operator may consider the use of physical/virtual resources and/or services of other operators. The orchestration process being noticeably different between a single-domain and multi-domain networks. Indeed, a domain orchestrator has only control over resources within the operator's administrative boundaries (De Sousa et al., 2018). Its role covers managing network services' lifecycle (by interacting with other components to control VNFs) and associated service provisioning resources (e.g. computing, storage, communication, etc.).

Typically, a single-domain orchestrator oversees and controls all resources and services within its domain, by leveraging the ETSI NFV-MANO framework. In contrast, orchestration in multi-domain networks is more difficult, given the incomplete knowledge and control of resources offered by independent providers.

From a service composition perspective, multi-domain specification of an end-to-end QoS requirement differs from that in a single-domain network since in the latter a single-domain QoS requirement is provided. Meanwhile, due to the heterogeneity of multi-domain infrastructures and administrative control, partial QoS requirements may be needed for each domain, thus constituting a significantly different set of constraints in comparison to a single domain network orchestration problem (Campbell et Nahrstedt, 2013 ; Abu-Lebdeh et al., 2017).

As of today, no standards exist for multi-domain orchestration (De Sousa et al., 2018). A number of multi-domain orchestration frameworks have been advocated in the literature, including T-NOVA (Kourtis et al., 2017), SONATA (Dräxler et al., 2017) and ONAP (*Open Network Automation Platform* s. d.).

Meanwhile, others proposed and investigated NFV orchestration architectures and use-cases inspired from the single-domain ETSI NFV framework. For instance, Rosa et al. (Rosa et al., 2015) proposed MD2-NFV where three use case scenarios are studied to highlight the benefits of distributed NFV. In (Etsi, 2015 ; Ietf, 2018), distributed MANO orchestration is discussed and three models are identified :

hierarchical, flat (or peer-to-peer) and hybrid.

Most works have adopted hierarchical orchestration in their proposals. ETSI report (Etsi, 2018) proposes a two-layer hierarchical architecture addressing end-to-end network services provisioning across two administrative domains, and the adaptation of the NFV-MANO framework to generalized multi-domain networks. In (Sciancalepore et al., 2016), an extension of the ETSI NFV MANO framework to enable joint orchestration of VNFs and edge computing applications is discussed.

Finally, (Borjigin et al., 2018) proposes a scheme named DARA to distribute network resources through a double auction approach. Their design considers three actors : an NFV broker, customers and SPs. The centralized broker collects SPs' resources to supply customers, while maximizing SPs' profits. Unlike previous works, (Borjigin et al., 2018) is the first in the literature to propose a multi-domain orchestration scheme.

In this work, we focus on the issue of NFV-based orchestration for multi-domain networks. We thus propose an auction-based method in which inter-operator interactions and exchange of network resources for service orchestration are encompassed into buyer/seller transactions and market dynamics. Namely, we turn SPs into buyers and InPs into sellers.

We further investigate different realistic market conditions to ensure profitability of all actors within the networks. To alleviate the scalability burden of such an auction framework, we leverage the benefits of multi-agent DRL in a heterogeneous and dynamic environment. Hence, SPs and InPs are able to engage into an autonomous negotiation/cooperation mechanism, where profitability and partial QoS requirements' satisfaction are taken into account.

Besides (Borjigin et al., 2018), this work is among the pioneers in proposing and investigating auction-based NFV multi-domain orchestration from the algorithmic aspect, rather than the architectural aspect. Moreover, to the best of our knowledge, this is the first work to leverage distributed/collaborative intelligence within SPs to achieve autonomous bidding/resource allocation behaviours within resources' marketplace.

4.4 System Model

For illustrative and simplicity purposes, we hereby describe a 5G mobile network scenario, where two main stakeholders co-exist : (1) an InP as the owner/controller of network resources, including base stations, radio spectrum licences, etc. The latter can be virtualized and leased to (2) SPs, who leverage these resources to offer various innovative and tailored services to attract more users - and profit - to the network (Kiiski, 2006). Since resources can be abstracted and sliced into virtual resources, several SPs can then co-exist under the same InP, hence contributing in the InP's expenditure savings. At the same time, by leveraging the logical isolation between these resources, an SP is free to use its allocated resources from the InP to accommodate the heterogeneous provided services to its own subscribers, with respect to QoS and SLA requirements, priority, etc.

To do so, we consider that the InP publishes its available resources to a free marketplace as shown in Fig 4.2. Then, the SPs compete for accessing these limited resources through a combinatorial auction process, allowing them to define their bids as combinations of discrete sets of resources, required to satisfy their subscribers needs. Naturally, both the InP and SPs aim at maximizing their own profits. For simplicity, we assume that the InP has no subscribers, and hence does not take part in the bidding process. We assume also that each SP is independently

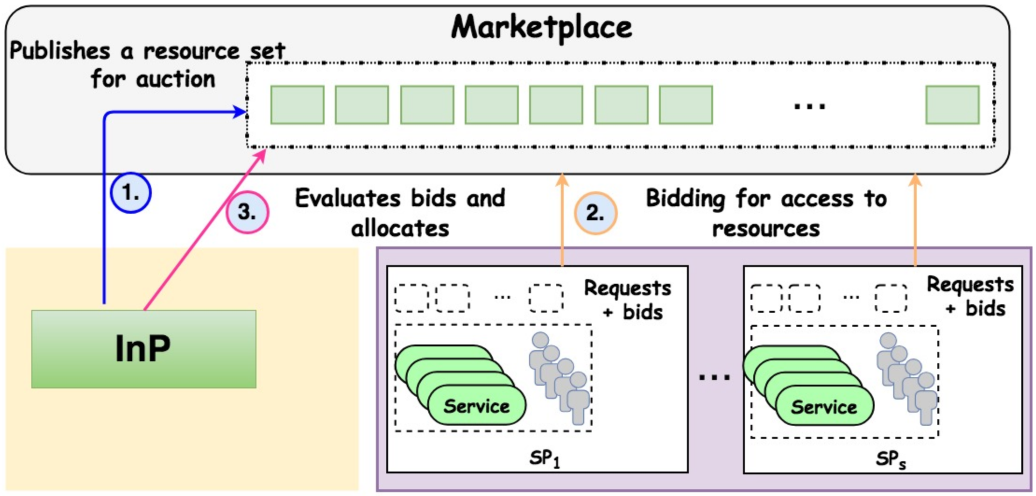

FIGURE 4.2 Bidding process.

operated, self-interested, and bids using an initially fixed budget. When an SP's budget is exhausted, it leaves the market, and therefore cannot participate in the next bidding rounds. In this context, SPs submit their requests for services with QoS requirements and a bidding offer to the InP. The latter evaluates all requests and bids, then selects the SPs to serve for a given time period. We also incentivize SPs to bid truthfully by assuming a penalty fee to pay each time an SP's bid is rejected, given that it fails to serve its subscribers.

In order to provide end-to-end service orchestration with stringent QoS requirements, an SP decomposes the service into VNF chains to deploy within network resources managed by InPs. Typically, in a single domain network setting with a centralized NFVO (ETSI, 2014), QoS requirements are generally specified on a higher granularity level. However, in multi-domain networks with distributed and independently managed orchestrators, specifying a global QoS requirement for a given service is inefficient due to to the heterogeneity of underlying networks in terms of available capacity, and to the ubiquitous locations of end-users. Consequently, being inspired from (DUAN, 2010), we consider a method in which a global QoS requirement (Q) is partitioned into partial QoS requirements, i.e. $Q_{InP_i} = \mathcal{P}(Q)$, with $\mathcal{P}(\cdot)$ denoting the partition function. The partial QoS depends on the serving InP's capacity, e.g. available network resources, connectivity, etc. Once the partial QoS requirements are set, the SP must ensure access to the limited resources by competing against other SPs in auctions. To this end, it must be able to derive an optimal bidding strategy regardless of market conditions.

For simplicity, we assume a single InP. Note that our model can be easily extended to multiple InPs, with SPs being able to switch between serving InPs as market conditions become detrimental for them over time, for instance due to sustained inability to satisfy end-users' requests, i.e. loss of profits.

4.5 Problem formulation

Consider a cellular network consisting of one base station (BS) owned and managed by an InP i. The InP plays the role of an auctioneer who serves several SPs within $\mathcal{S} = \{1, 2, \ldots, |\mathcal{S}|\}$, where $|.|$ is the cardinality operator. Each SP s, playing the role of a bidder, is assumed to be in charge of $\mathcal{N}_s = \{1, 2, \ldots, |\mathcal{N}_s|\}$ subscribers.

We assume that the available radio spectrum resource is of a total bandwidth W, and is divided into $\mathcal{C} = \{1, 2, \ldots, |\mathcal{C}|\}$ orthogonal subcarriers. The SPs compete to get these subcarriers. Also, let $a^t_{c,s} \in \{0, 1\}$ be the binary variable indicating whether subcarrier $c \in \mathcal{C}$ has been allocated at the end of time t to SP s, following the bids evaluation by the InP, or not. We assume that for each subcarrier c, an undisclosed minimum operational price $\lambda^t_{c,min}$ is estimated and accounted by the InP. Moreover, each SP s has a current bidding budget $\mathcal{B}^t_s$, that fluctuates with the bidding rounds where SP s participates.

Time is considered slotted, where a set of observations time periods are defined as $\mathcal{T} = \{1, 2, \ldots |\mathcal{T}|\}$ with $t \in \mathcal{T}$. The end of each time period coincides with the end of the auction and the decision-making process for resources allocation by the InP. Let $y_{s,i}^{t,c} \in \{0,1\}$ be the binary variable indicating if a bid submitted by an SP s is accepted by the InP i at time t or not.

At the beginning of each time period t, SP s sends its request $z_s^t = \{u_s^t, b_s^t\}$, where, for simplicity, u_s^t is the QoS requirement (e.g. minimum data rate), demanding c_s^t subcarriers to satisfy its $|\bar{\mathcal{N}}_s^t|$ subscribers in time t, and b_s^t is the bid that it is willing to pay to access the required resources.

We assume that b_s^t is limited by a maximum bid $b_{s,max}^t$. Moreover, we define the penalty that SP s has to pay when it fails to satisfy its subscribers' demands as :

$$\delta_s^t = |\bar{\mathcal{N}}_s^t| \cdot \delta \quad \forall t \in \mathcal{T},\ s \in \mathcal{S}, \tag{1}$$

where δ represents a unitary fee.

From the InP's perspective, resource allocation through an economic driven optimization model yields two types of benefits :

1. InP resource-centric : better resources utilization, availability and performances (e.g. data rates).
2. SP-centric : Increased profits, reduced costs and fairness.

Let R_i^t be the revenue obtained by InP i at time t, given by :

$$R_i^t = \sum_{s \in \mathcal{S}} y_{s,i}^{t,c} \cdot b_s^t \quad \forall t \in \mathcal{T}, \tag{2}$$

and $R_i^{Total} = \sum_{t \in \mathcal{T}} R_i^t$ is the total revenue garnered by the InP. The latter is deemed to make a profit at time t when $R_i^t \geq \sum_{c \in \mathcal{C}} \lambda_{c,min}^t$, otherwise it loses money.

Hence, the profit maximization problem of the InP can be formulated as follows :

$$\max_{a^t_{c,s}} \quad R_i^{Total} \tag{3a}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}} c^t_s \cdot y^{t,c}_{s,i} \leq |\mathcal{C}| \quad \forall t \in \mathcal{T}, \tag{3b}$$

$$\sum_{s\in\mathcal{S}} c^t_s \cdot y^{t,c}_{s,i} \geq \Omega(u^t_s) \quad \forall\ t \in \mathcal{T} \tag{3c}$$

$$\sum_{s\in\mathcal{S}} a^t_{c,s} \leq 1 \quad \forall t \in \mathcal{T}, c \in \mathcal{C} \tag{3d}$$

where $\Omega(.)$ denotes a function used by the InP to compute the minimum required subcarriers to satisfy a given data rate. Constraints (3b), (3c) and (3d) ensure that the InP does not over-allocate its resources, or allocate the same resource to different SPs, while allocating enough resources to satisfy SPs' QoS requirements. In order to maintain its level of profitability, the InP must also take the necessary steps to ensure that the value of its resources does not depreciate over time.

From a SP's perspective, the objective is to maximize its profit while satisfying its subscribers' demands. Depending on the market conditions and number of subscribers to serve, a variety of bidding behaviours may be suitable, where the SP evaluates the risks associated to under-market bids. We denote by $R_s^{Total} = \sum_{t\in\mathcal{T}} R^t_s$ the total revenue obtained by SP s, where R^t_s is the revenue obtained at time t, written as :

$$R^t_s = (y^{t,c}_{s,i} \cdot (b^t_{s,max} - b^t_s)) - \delta^t_s, \quad \forall t \in \mathcal{T}, b^t_s \leq b^t_{s,max}, \tag{4}$$

The profit problem of SP s is formulated as :

$$\max_{b^t_s} \quad R_s^{Total} \tag{5a}$$

$$\text{s.t.} \quad b^t_s \leq b^t_{s,max} \leq \mathcal{B}^t_s \quad \forall t \in \mathcal{T}, \tag{5b}$$

$$\mathcal{B}^t_s > 0 \quad \forall s \in \mathcal{S},\ t \in \mathcal{T}. \tag{5c}$$

TABLE 4.1 Notations

$\mathcal{C}$	Set of subcarriers
$\mathcal{S}$	Set of SPs
$\mathcal{T}$	Discrete time system
$\mathcal{N}_s$	Set of subscribers for SP s
$a^t_{c,s}$	1 if subcarrier c is allocated to SP s at time t; 0 otherwise
b^t_s	Bid sent by SP s at time t
$b^t_{s,max}$	Maximum bid by SP s at time t
$\mathcal{B}^t_s$	Budget of SP s at time t
$\bar{\mathcal{N}}^t_s$	Set of subscribers of SP s to be served at time t
R^t_i	Revenue obtained by InP i at time t
R^{Total}_i	Total revenue garnered by InP i
R^t_s	Revenue gained by SP s at time t
R^{Total}_s	Total revenue gained by SP s at time t
u^t_s	Minimum data rate for request sent by SP s at t
$y^{t,c}_{s,i}$	1 if bid submitted by SP s at time t is accepted by InP i; 0 otherwise
z^t_s	Request sent by SP s at time t
$\lambda^t_{c,min}$	Minimum reserve price for subcarrier c at time t
δ^t_s	Penalty fee of SP s at time t

All the defined variables are summarized in Table 4.1.

A fundamental aspect to note is that auctions differ not only by the associated rules but also by the auction environment. Hence, auctions can be studied in a wide range of environments, with varying numbers of sellers/buyers, number of resources within the marketplace, exchanged information, etc. (CRAMTON et al., 2006).

In our work, we define several scenarios under different auction settings and market conditions. Moreover, since the InP and SP formulated problems generalize the combinatorial allocation/auction problem, in particular the NP-complete winner determination problem (CRAMTON et al., 2006), heuristic algorithms are needed to find optimal or near-optimal solutions in polynomial times.

In the next sections, we detail the scenarios to investigate. Then, we expose the developed solutions for the InP and SP formulated problems.

4.6 Auction scenarios

We present in this section, the auctions scenarios to be considered and evaluated. In particular, we consider two situations : *bidding war* and *bidding collusion*. In the former, bidders attempt to outbid each other in their pursuit of network resources. In the short term, this would represent the optimal situation for the InP as it generally leads SPs to bid over the real-value price, thus substantially increasing its profit. Adversely, this may be the worst situation for poor SPs, as it can push them out of bidding rounds and give more power to wealthy SPs to control the market.

It has been shown in (Maillé et al., 2009 ; Bhaskar et al., 2002) that oligopolistic/oligopsonic coordination, i.e. *bidding collusion*, is more likely to occur in this situation, aiming at maximizing the collusion-players profits at the expense of other players' and auctioneers' welfare. Bidding collusion in telecommunication networks is a realistic assumption, and has been well-documented in the following works (Maillé et al., 2009 ; Massey et McDowell, 2008). The problem is further amplified by technology-aided price fixing algorithms, for which collusion is much harder to detect.

4.6.1 Scenario 1

We first investigate an auction environment where the allocation policy of the InP is known to all bidders, that is the bidders with the highest bids are always selected, for each time period.

In this scenario, we assume that SPs may realistically have different initial budget powers, and may pursue resources in a conservative or an aggressive way, depending on their initial budgets and growth in number of subscribers. Moreover, as a result of the InP policy, there is a risk of starvation, hence pushing poor SPs to leave the market while allowing wealthy SPs to expand their control on resources' prices.

As a consequence, an oligopsony may be created and the profits of the InP may be threatened (Pla et al., 2015). To avoid this situation, the InP introduces a fairness model where every SP must be served at least once within a fixed number of bidding rounds, denoted τ_{th}. Precisely, the InP keeps track of the number of consecutive wins and losses for each SP, then establishes SPs' priority ranks.

Finally, we assume that the InP analyzes the interest sparked over time by its resources within the market, i.e appreciation or depreciation, and adjusts its minimum asking price $\lambda^t_{c,min}$ accordingly in order to maximize its revenues or rejuvenate interest for its resources.

4.6.2 Scenario 2

Pushing further the first scenario, we analyze the behaviour dynamics of SPs within the marketplace in this second scenario. We assume here that the InP does not apply a fairness mechanism, but rather attempts to detect and punish instances of collusive cooperation among wealthy bidders.

In order to allow poor SPs remain competitive, the InP permits competitive cooperation (or coopetition) among them. *Coopetition* is the strategy in which bidders, with partial congruent goals, cooperate and compete simultaneously by sharing partial information (Brandenburger Adam et Nalebuff Barry, 1996 ; Dagnino, 2009).

In our paper, we call by poor, every SP with an initial budget under a predefined threshold.

4.7 Proposed solutions

In this section, we present the proposed strategies to adopt by the InP and SPs aiming at maximizing their profits.

4.7.1 From the InP's Perspective

Determining the winners in combinatorial auctions is computationally complex due to the problem's NP-completeness. A synopsis of the problem and main approaches to solve it are presented in (Cramton et al., 2006). For our work, we leverage an algorithm adapted from CABOB (Combinatorial Auction Branch On Bids) (Sandholm et al., 2005), that is an optimal tree search algorithm based on a mix of linear programming and branch-and-bound techniques (interested readers are referred to (Sandholm et al., 2005) for further details). This approach is detailed in Algorithm 5.

Its main idea consists of maintaining a bid graph $\mathcal{G}$ where a branch and bound Depth First Search (DFS) method is applied to find the most profitable bids for the InP, i.e. to be selected for service.

Graph nodes represent the SPs' bids that can still be appended to the search path, given that bids do not concern already allocated network resources. Moreover, two nodes are bound by an edge when the corresponding SPs' bids compete for the same resources. In the algorithm, f^* denotes the best solution found and is regularly updated as better bids are found during the search. Revenue from winning bids on the search path is denoted by g. As nodes (selected bids) are removed

Algorithm 5: CABOB*(G, g, $\lambda^t_{c,min}$)

$\lambda_{min} = \lambda^t_{c,min}$;

$\mathcal{F} : \{G_1, G_2, ..., G_{|\mathcal{F}|}\} \leftarrow$ DFS(G) ;

Compute upper bound $\mathcal{U}_f$ **for** f in $\mathcal{F}$;

if $\sum_{i=1}^{|\mathcal{F}|} \mathcal{U}_f \leq \lambda_{min}$ then

  return 0;

end

Compute lower bound $\mathcal{L}_f$ **for** f in $\mathcal{F}$;

if $g + \sum_{i=1}^{|\mathcal{F}|} \mathcal{L}_f > f^*$ then

  $f^* \leftarrow g + \sum_{i=1}^{|\mathcal{F}|} \mathcal{L}_f$; $\lambda_{min} \leftarrow \lambda_{min} + g + \sum_{i=1}^{|\mathcal{F}|} \mathcal{L}_f - f*$;

end

if $|\mathcal{F}| > 1$ then

  $\mathcal{F}^* \leftarrow 0$; $\mathcal{U}' \leftarrow \sum_{i=1}^{|\mathcal{F}|} \mathcal{U}_f$; $\mathcal{L}' \leftarrow \sum_{i=1}^{|\mathcal{F}|} \mathcal{L}_f$;

  for k *in* $\mathcal{F}$ do

    If $\mathcal{F}^* + \mathcal{U}' \leq \lambda_{min}$ **then** return 0;

    $g'_f \leftarrow \mathcal{F}^* + (\mathcal{L}' - \mathcal{L}_f)$; $f^*_{old} \leftarrow f^*$;

    $f^*_k \leftarrow CABOB^*(G_k, g + g'_k, \lambda_{min} - g'_k)$;

    $\lambda_{min} \leftarrow \lambda_{min} + (f^* - f^*_{old})$; $\mathcal{F}^* \leftarrow \mathcal{F}^* + f^*_k$; $\mathcal{U}' \leftarrow \mathcal{U}' - \mathcal{U}_f$; $\mathcal{L}' \leftarrow \mathcal{L}' - \mathcal{L}_f$;

  end

  return $\mathcal{F}^*$;

else

  $\Theta_s \leftarrow \{c : a^t_{c,s} = 1\}$; $\Theta_{s'} \leftarrow \{c : a^t_{c,s'} = 1\}$;

  Select next bid set $b^*_s = \{b^t_s \cup \Theta_s\}$ to branch on ; $G \leftarrow G - b^*_s$;

  for $s' \in \mathcal{S}$ do

    If $s' \neq s$ and $\Theta_s \cap \Theta_{s'} \neq \emptyset$ **then** $G \leftarrow G - b^*_{s'}$;

  end

  $f^*_{old} \leftarrow f^*$; $f_{in} \leftarrow CABOB^*(G, g + b^t_s, \lambda_{min} - b^t_s)$; $\lambda_{min} \leftarrow \lambda_{min} + (f^* - f^*_{old})$;

  for $s' \in \mathcal{S}$ do

    If $s' \neq s$ and $\Theta_s \cap \Theta_{s'} \neq \emptyset$ **then** $G \leftarrow G \cup b^*_{s'}$;

  end

  $f^*_{old} \leftarrow f^*$;

  $f_{out} \leftarrow CABOB(G, g, \lambda_{min})$;

  $\lambda_{min} \leftarrow \lambda_{min} + (f^* - f^*_{old})$; $G \leftarrow G - b^*_s$;

  return $max(f_{in}, f_{out})$;

end

from $\mathcal{G}$ down a search path, their edges are also removed. Similarly, as nodes are reinserted into $\mathcal{G}$ when backtracking, edges are also reinserted. To prune across independent components of $\mathcal{G}$, λ_{min} is used to denote the minimum revenue that the InP expects from a given SP's bid (Sandholm et al., 2005). Furthermore, it evaluates whether a search should be pruned based on the unallocated items, estimated by the upper and lower revenue bounds.

4.7.1.1 Scenario 1

For this scenario, we first upgrade CABOB with a fairness mechanism. To do so, the InP artificially inflates losing SPs' bids at the end of each time period t, using a weight parameter Δ_s^t. The latter is increased after each bid loss. Its value is reinitialized to the smallest value once the SP wins a bid within τ_{th}. We also consider a function to gauge the interest sparked by the InP's resources over time. To this end, we note the expected difference between the InP's desired minimum resource price $\lambda_{c,min}^t$ and SPs' bids, a variant of the metric known in the literature as the bid-ask spread. Obviously, the InP's objective is to minimize the bid-ask spread by encouraging competition in the market. It could be argued that the InP can also reduce temporarily its minimum price (ask) to induce market rally, at the expense of a short profit loss, but subsequent gains would be obtained at a later time. For this, the InP needs to keep historical data from bidding rounds.

4.7.1.2 Scenario 2

In this scenario, it is critical for the InP to accurately distinguish occurrences of *coopetition* and *collusion* given their similarity in the cooperation process. Indeed, actors in a coopetition strategy work to ensure their stability and viability within the market, whereas actors in a collusion strategy impede market's competition in order to increase their profits at the expense of the auctioneer's and other actors' welfare (Pathak et al., 2013). Here, the InP leverages a simple reputation-based approach, where a score $\vartheta_s = \sum_{t \in \mathcal{T}} \vartheta_s^t$ is associated to each SP, based on the latter's behaviour over time and its impact on the InP's and SPs' profits, such that :

$$\vartheta_s^t = \rho_s^t + \sum_{s' \in \mathcal{S} \setminus \{s\}} m_{s,s'} \tag{6}$$

where ρ_s^t is the utility score of SP s in time t, calculated regarding its own profit and the market value of the resources, and $m_{s,s'}$ is the similarity factor between SPs s and s'. ρ_s^t is given by :

$$\rho_s^t = q^{t'/\mathcal{T}'} \cdot |b_s^t - \lambda_t^{c,min}|_a \tag{7}$$

where $|.|_a$ is the absolute value, q denotes an exponential growth parameter with $\mathcal{T}'$, an observation period, and t' is an incremental factor, increased when the bid b_s^t, with a limited utility to SP s, persists from a previous observation period, and resets to 0 otherwise. Whereas, $m_{s,s'}$ can be written as :

$$m_{s,s'} = \sqrt{(b_s^t - b_{s'}^t) + (u_s^t - u_{s'}^t)},\ \forall (s, s') \in \mathcal{S}^2. \tag{8}$$

Hence, the InP maintains a similarity matrix of all SPs as follows :

$$\mathbf{M} = \begin{pmatrix} m_{1,1} & \cdots & m_{1,|\mathcal{S}|} \\ m_{2,1} & \cdots & a_{2,|\mathcal{S}|} \\ \vdots & \ddots & \vdots \\ m_{|\mathcal{S}|,1} & \cdots & m_{|\mathcal{S}|,|\mathcal{S}|} \end{pmatrix}. \tag{9}$$

The above information helps the InP evaluate whether an SP's behaviour is more geared towards its own profit or aims at inhibiting competitors. When the InP is confident about a collusive cooperation between two or more SPs, it attempts to break this partnership by granting access to its resources to only one of the colluding SPs for a given time period. Similarly to Scenario 1, ill-behaved (collusive) SPs are temporarily penalized by artificially deflating their bids' values.

4.7.2 From the SPs' Perspective : A Distributed Multi-agent Deep Reinforcement Learning Approach

From the SPs standpoint, the bidding problem can be modeled as a multi-agent system, where each agent's knowledge and strategic actions are specific to them and dependent on their budget constraints or their perception of the auction environment. We use a multi-agent reinforcement learning approach that can be considered as an extension of Markov Decision Process (MDP), called Markov Game. In this game, $|\mathcal{S}|$ SPs bid for the InP's resources, to satisfy their subscribers' needs. A Markov game is defined by a set of states $\mathcal{X}$ describing the possible status of all bidding agents, a set of actions $\{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_{|\mathcal{S}|}\}$ with $\mathcal{A}_s$ the action space of SP s. At each time period t, each SP s uses a policy $\pi : \mathcal{X}_s \longmapsto \mathcal{A}_s$ to determine an action a_s, where $\mathcal{X}_s$ is the state space of SP s. After the execution of a_s, SP s transfers to the next state according to the state transition function $\mathcal{X} \times \mathcal{A}_1 \times \dots \times \mathcal{A}_{|\mathcal{S}|} \longmapsto \phi(\mathcal{X})$ where $\phi(\mathcal{X})$ indicates the set of probability distributions over the state space. Each SP s obtains a reward based on a function of the state, and all the SPs' actions as $R_s^t : \mathcal{X} \times \mathcal{A}_1 \times \dots \times \mathcal{A}_{|\mathcal{S}|} \longmapsto \mathcal{R}$. Each SP s maximizes its own total expected revenue R_s^{Total} such that :

$$R_s^{Total} = \sum_{t \in \mathcal{T}} \gamma^t \cdot R_s^t, \tag{10}$$

where $\gamma^t \in [0, 1]$ denotes a discount factor meta-parameter that underlines the perceived importance of future rewards. Specifically, a factor of 0 will make the agent short-sighted by only considering current rewards, while a factor approaching 1 will make it strive for a long-term high reward. Hereafter, we identify the states, actions and rewards in our auction environment :

- **State** Our state design aims at letting any SP optimize its bidding decisions based on its perception of the auction environment. We design the agent state to consist of the QoS requirements u_s^t of its submitted request(s), its bid

b_s^t, whether the bid of SP s was accepted or not by the InP $y_{s,i}^{t,c}$, $\theta_{s,s'}^t \in \{0, 1\}$ indicating whether the agent s decides to cooperate with another agent s' or not, the incurred penalty δ_s^t and the current losing streak l_s^t.

Hence, the state for SP s is composed as :

$$\mathcal{X}_s = \{u_s^t, b_s^t, y_{s,i}^{t,c}, \theta_{s,s'}^t, \delta_s^t, l_s^t\}. \tag{11}$$

- **Action** Every SP s has to take a bidding decision at time t in order to ensure its bid is/remains selected by the InP, for the sake of its subscribers' satisfaction and its own profit. Depending on the environment's state, it must decide whether to adjust or not its next bid, while keeping in mind budget constraints. In a cooperative scenario, it must determine how to bid with another agent. An agent's action a_s can be given by :

$$a_s(\mathcal{X}_s) = \{b_s^{t-1} + \mu_s^t\} \cup \{\theta_{s,s'}^t\}, \tag{12}$$

where $\mu_s^t \in \mathbb{Z} : \mu_s^t \in [-b_s^{t-1}, (\mathcal{B}_s^t - b_s^{t-1})]$. It must be noted that, even though constrained, our action space remains very large. For computation efficiency purposes, we discretize the action space by defining for each agent a set of legal bids, denoting a range of bidding behaviour from prudent to aggressive. More specifically, each agent is free to bid a portion or the totality of its maximum bid $b_{s,max}^t$. Bid-action space discretization is implemented through equal-width binning (Dougherty et al., 1995) with a predetermined number of bins.

- **Reward** The agent's reward is formulated as :

$$R_s^t = (y_{s,i}^{t,c} \cdot (b_{s,max}^t - b_s^t)) - \delta_s^t. \tag{13}$$

In this formulation, an agent gains significant profit if it manages to win in the auction by bidding as low as possible. It is worth noting that it is possible for coopeting agents to win in the auction and still incur penalties if a portion of their subscribers are left unserved. In contrast, when an agent takes too much risk by submitting below market bids and as a result loses in the auction, it will be penalized.

A key aspect for the SP, with regards to computational efficiency, consists of its training time. Several works (Shallue et al., 2018 ; Dean et al., 2012 ; Gholami et al., 2018 ; Daniely et al., 2014 ; Bartlett et Ben-David, 2002 ; Livni et al., 2014 ; Kearns, 1990 ; Boob et al., 2018) emphasize the increased scale of parallelism available for neural network training, made possible by hardware advancements. Hence, assuming optimized data parallel systems that spend negligible time synchronizing between processors and leveraging practical training “tricks" (e.g., regularization, over-specification, adequate activation functions, batch size increase, etc.), training time can be reduced and measured in the number of training steps (Shallue et al., 2018 ; Livni et al., 2014). For instance, (Shallue et al., 2018) showed the correlation between reduced training time and increasing batch size without degradation of the solution quality.

4.7.3 Bidding analysis

The objective of the auction framework presented above is two-fold : (1) assign resources to bidding who value them the most while (2) ensuring as well profitable revenues for the InP, particularly in cases where SPs have heterogeneous budgets and in the presence of collusion. While truthfulness is the dominant strategy in Vickrey auctions (Vickrey, 1961), a noted shortcoming consist of its inability to guarantee auctioneer revenues (Ausubel et Cramton, 2004). Our framework counters by allowing the InP to artificially introduce competition in both scenarios, respectively applying a fairness and a collusion detection/punishment mechanism in addition to enabling coopetition for poor agents.

Consider $b^t_{s,max}$ as the true value of resources (which are indivisible) for an SP s. The payoff received is given in eq.(13). Allow $\max\limits_{s' \neq s} b^t_{s'}$ as the highest bidder s' among competing SPs other than SP s. Recall the InP allocation policy, which consists of always selecting the highest bids. Furthermore, an SP s cannot bid above its budget. These are the following outcomes following a bid by SP s for a time period t :

$$\max_{s' \neq s} b^t_{s'} < b^t_{s,max} \tag{14}$$

$$\max_{s' \neq s} b^t_{s'} < b^t_s \mid b^t_s < b^t_{s,max} \tag{15}$$

$$\max_{s' \neq s} b^t_{s'} > b^t_{s,max} \tag{16}$$

$$\max_{s' \neq s} b^t_{s'} > b^t_s \mid b^t_s < b^t_{s,max} \tag{17}$$

$$b^t_s < \max_{s' \neq s} b^t_{s'} > b^t_{s,max} \tag{18}$$

In eq. (14) and (15), SP s wins both with a truthful bid, and underbids with a positive payoff for an underbid compared to a zero penalty paid for truthful bidding. In eq. (16) and (17), the SP loses the bid and incurs a penalty being paid regardless of the employed bidding strategy. The only winning strategy in eq. (18) is truthful bidding with zero payoff while other bidding strategies induce a penalty. Hence, it can be seen that truthful bidding is the dominant strategy in every case except in eq. (15) as underbidding is the dominant strategy.

In our framework, the weak dominance of truthful bidding is designed as an incentive to prolong market competitiveness. It allows for poor SP s to remain within the market given it can adequately adapt its bidding strategy. For richer SPs, underbidding is a risky strategy as it gives poor SPs an opportunity to win the auction. Furthermore, in the scenario where a fairness mechanism is applied, an underbidding SP would probably lose in favour of a poor SP with an artificially inflated bid. Collusive SPs attempt to bid truthfully in the short term, hoping to drive poor SPs out of the auction, and aiming to underbid in the long term. We counteract this set of strategies by detecting collusive SPs at first. Once this is done, we create a cost asymmetry between the colluders by granting access to resources to only one SP, which constitutes an effective barrier to collusion (Ivaldi et al., 2003).

4.8 Experimental evaluation

For our evaluations, we leveraged the Open AI Gym toolkit to implement a custom auction simulation environment, where SPs bid to gain access to resources to satisfy their subscribers' demands. As mentioned in Section 4.7.1, the InP runs a modified version of CABOB, and activates in certain scenarios a fairness mechanism favouring underserved SPs within the market. We leave the case of multiple InPs and its analysis for future work. We consider 8 independent bidding agents (i.e. SPs) who apply specific bidding policies, learned after 1000 episodes of training. Auction environment parameters are summarized in Table 4.2.

Given our action space, we evaluate and compare the following bidding algorithms in similar simulation settings :

- *Incremental* : An agent using this algorithm applies a simple conservative bidding strategy. When it wins in a bidding round, it linearly decreases its bid. In contrast, when the agent suffers an auction loss, it exponentially increases its bid in the following round. Incremental is designed to mimic the behaviour of two algorithms. Similarly to AWESOME (Conitzer et Sandholm, 2007), it assumes rival agents behavior at different stages of the game and tries to maintain hypotheses about an agent's behaviour. Like GIGA-WoLF (Bowling, 2005), it also uses an adaptive step that makes it more or less aggressive in changing its bidding strategy.

- *Random* : In this benchmark strategy, an agent randomly selects an action within the legal action set.

- *DQN (Deep Q-learning) :* DQN is a variant of the well known Q-learning technique using a deep neural network for stable learning (Mnih et al., 2015). It uses the experience replay technique, where random samples of previously stored experiences are taken into account for future learning and action selection.

- *DDQN (Double DQN) :* DDQN aims at reducing the overestimation of Q values, encountered in DQN (Hasselt, 2010). Hence, it allows faster training and a more stable learning. It leverages two policies, one for value evaluation and another for future action decisions.

The baseline approaches presented above aim to highlight specific insights on the behaviour of deep learning-driven agents in dynamic market interactions. We omit multi-agent learning algorithms such as Fictitious Play (Brown, 1951), Bully (Littman et Stone, 2001), AWESOME (Conitzer et Sandholm, 2007), Meta (Powers et Shoham, 2005), Minimax-Q (Littman, 1994), Nash-Q (Hu et Wellman, 2003), Correlated-Q (Greenwald et al., 2003), GiGA-WoLf (Bowling, 2005), $RV_{\sigma(t)}$ (Banerjee et Peng, 2006), and GSA (Spall, 2005), given they have all been shown to be less performing and stable than Q-learning (Zawadzki et al., 2014). Evolutionary-based algorithms are also omitted due to their drawbacks and for fairness purposes. Indeed, evolutionary algorithms cannot guarantee optimality. In addition, the associated solution's quality depends highly on the initialization setting, and it deteriorates as the problem size increases.

Based on the scenarios of section 4.6, three types of auction environments are assumed : 1) Full competition, 2) collusion with fairness and 3) collusion and

Table 4.2 Sim. Parameters

Auction settings	
Number of bidding rounds	10
Number of SPs $\mathcal{S}$	8
Number of InPs	1
Fairness threshold τ_{th}	3
Number of action space discretization bins	10
InP settings	
Number of subcarriers	10
SP settings	
Number of subcarriers required to satisfy QoS	4
Penalty after bid rejection	25
Number of subscribers	3
Number of bidding behaviours	6
Hyperparameters	
Learning rate	0.001
Memory size $\mathcal{S}$	10000
Batch size	128
Probability ϵ	1 - 0.01
Probability ϵ decay	1 - 0.01
Number of episodes	4000
Discount factor γ	0.95

coopetition. In the first, we investigate the market dynamics in which agents are seeking access to the InP's resources without any form of cooperation. In the second, some agents may coordinate their bids to increase their mutual profits. Specifically, we consider 2 colluding and 6 competing agents within the marketplace. In the last case, we assume 2 colluding, 2 coopeting and 4 competing agents within the marketplace. Moreover, we qualify the agents with a starting budget within the range 800 to 1000 as rich agents, within 500 to 800 as middle budget agents, while agents with budgets below 300 are called poor agents. The environment parameters are summarized in Table 4.3.

Table 4.3 Budget Distribution

	Number of agents	Budget range
Full competition		
Competing agents	8	300 - 1000
Colluding agents	0	-
Coopeting agents	0	-
Collusion		
Competing agents	6	300 - 1000
Colluding agents	2	1000
Coopeting agents	0	-
Collusion and coopetition		
Competing agents	2	500 - 1000
Competing agents	2	300 - 500
Colluding agents	2	1000
Coopeting agents	2	300

4.8.1 From the InP perspective

First, we investigate the InP's average profits under different auction environments as shown in Fig. 4.3. According to simulations, the most profitable auction environment is full competition. In this case, DRL-aided SPs rely only on themselves to win in the auction market. We observe a general tendency where SPs typically react by bidding aggressively after an auction loss, thus leading to a profit increase for the InP. Similarly, when an agent achieves sustained auction wins, it seems to regularly lower its bids, in an attempt to increase its own profits. Such behaviour is highlighted, for instance in Fig. 4.7 where a DDQN-driven SP is able to adapt its bidding behaviour quickly after consecutive auction losses. With similar profita-

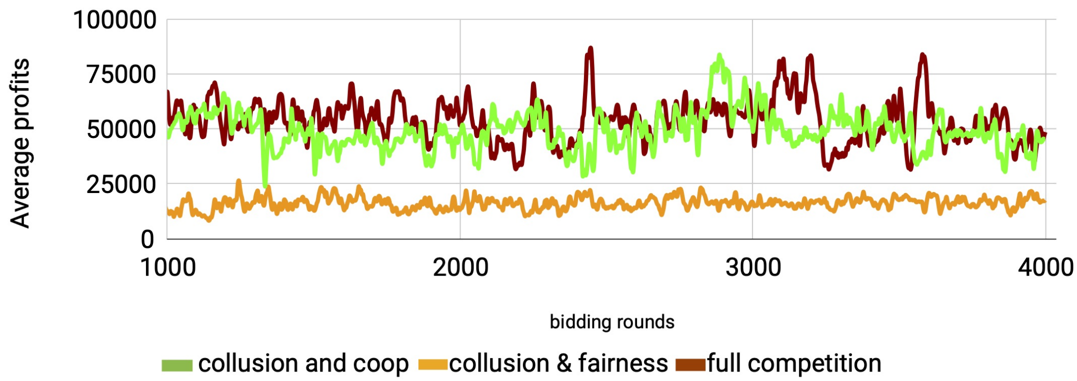

FIGURE 4.3 Average InP profits in different auction environments.

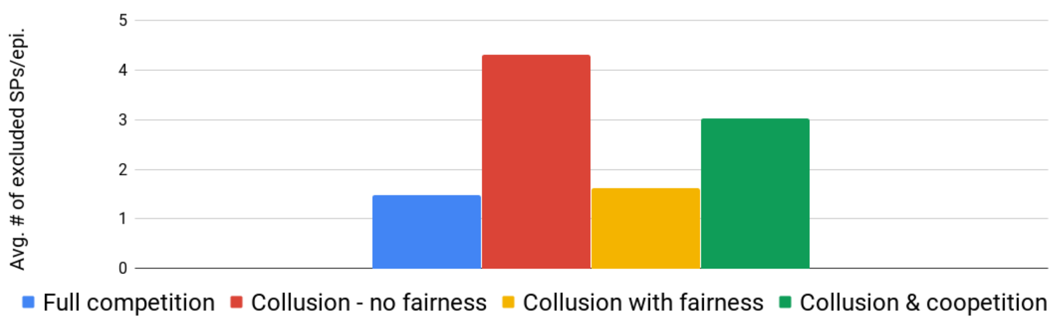

FIGURE 4.4 Average number of excluded SPs per episode.

bility, the collusion and coopetition auction setting can arguably be viewed as a form of full competition. Indeed, the InP counter-attacks collusion by empowering poor agents to jointly bid to access the resources and avoid quitting the market. As a consequence, colluding agents are forced to over-bid and enrich the InP. However, colluding remains able to profit rich agents at the expense of poor ones. In an attempt to remedy to this situation, we proposed a fairness mechanism, where the InP allow SPs to access its network resources within a certain threshold ($\tau_{th} = 3$). This setting is shown to be the least profitable, since enforced fairness comes at the expense of its profits.

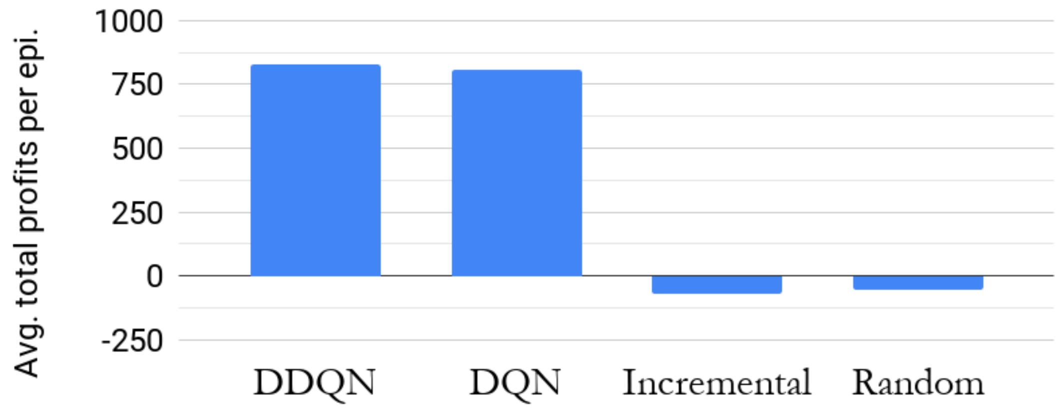

FIGURE 4.5 Average total profits per episode under each algorithm.

In Fig. 4.4, we compare the average number of SPs forced out of market. The highest number of exclusions is in the collusive environment. Indeed, since collusive agents have high budgets, they are able to consistently bid over market value, leading into sustained auction losses to poor agents. Both full competition and the collusion with fairness settings have the best ability to keep poor SPs in the market. Whereas, collusion and coopetition environment presents a non-negligible number of exclusions. This can be due to the amount of time it takes coopeting agents to adjust their bids and beat colluding agents. To be noted that coopetition is alleviated when agents' budgets become above 700 (middle budget agent) due to accumulated profits.

4.8.2 From the SPs perspective

In Fig. 4.5, DDQN, DQN, incremental and random algorithms are compared, in terms of average profits gained by an SP. Both incremental and random algorithms present the worst profits. Whereas, DQN and DDQN achieve very high performances. Indeed, these DRL-based approaches enable SPs to smartly adjust their bidding

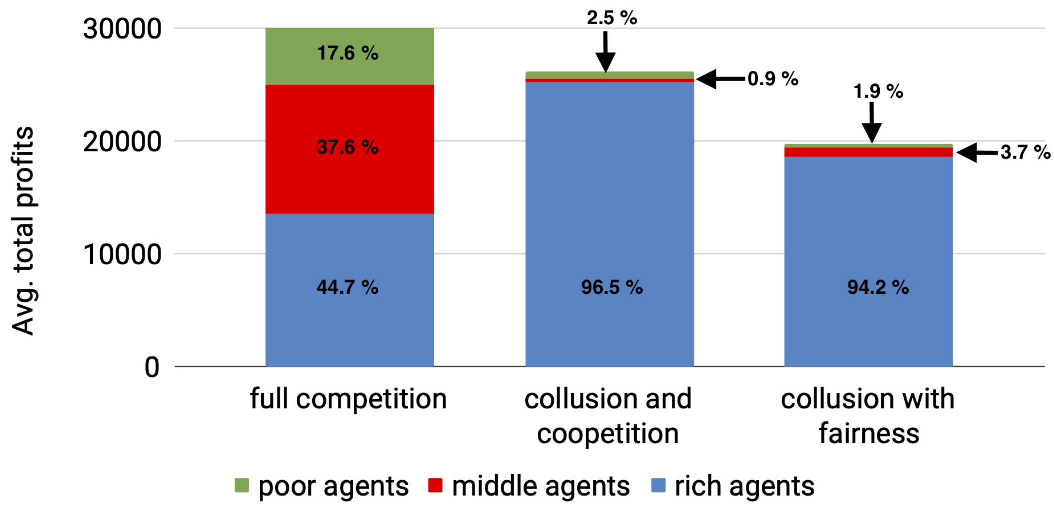

FIGURE 4.6 Average total SP profits under different auction settings.

strategies according to market conditions. Indeed, as it can be seen in Fig. 4.7, the DDQN agent is able to recover after a series of auction losses. In the remaining simulations, only DDQN agents will be considered, for their fast convergence and better performances, compared to the other approaches.

In Fig. 4.6, we illustrate the SPs' profits as a function of auction environments, and for different SPs' profiles. We assume that rich agents are initiated with a budget of 1000, middle agents with a budget of 500 and finally poor agents with a budget of 300. Meanwhile, the same number of subscribers and QoS requirements are given to them. Results show that the full competition environment is the most beneficial to both middle and poor agents. Interestingly, this setting showcases that middle agents and rich agents are able to compete with comparable profit margins despite their budget differences, thus indicating a potential for greater profits if agents are able to adapt more precisely their bidding behaviour. This can be seen in Fig.4.7 where each agent is able to secure wins and most importantly offsets auction losses with profits margins mostly above 0 during the auction process.

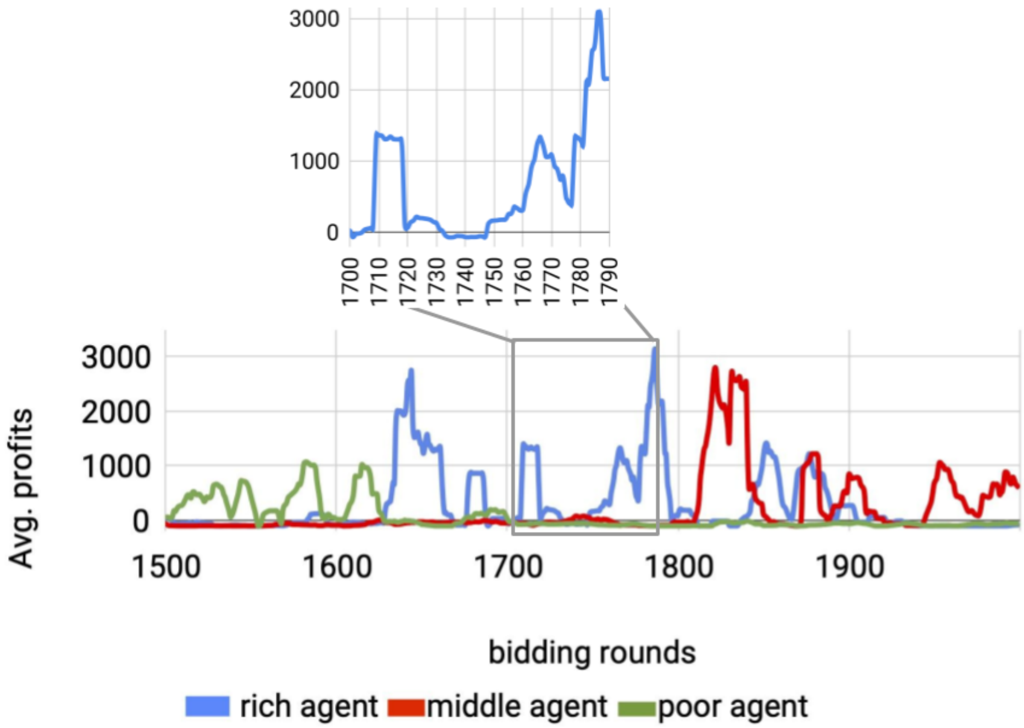

FIGURE 4.7 Average SP profits under full competition auction environment.

Moreover, poor agents' strategy seems to require short aggressive bidding bursts in order to win, with a downside of limited bidding power afterwards. Finally, it is shown that the fairness mechanism, while improves middle agents' performance, has a very small impact on the rich agents' profits. Indeed, SPs' intelligence allows them to detect the fairness mechanism and bypass it, by smartly adjusting their bids. Consequently, colluding agents still get most profits.

The impact of the penalty's value is investigated, by illustrating agents' bidding behaviours in Fig.4.8 where the vertical axis denotes the level of bid aggressiveness adopted by an agent. Here, we compare two agents operating under a different penalty coefficients. As such, we observe a trend where the agent with a lower penalty coefficient seems to adopt a risky bidding behaviour by bidding less aggressively perhaps to conserve its limited budget achieving occasionally high

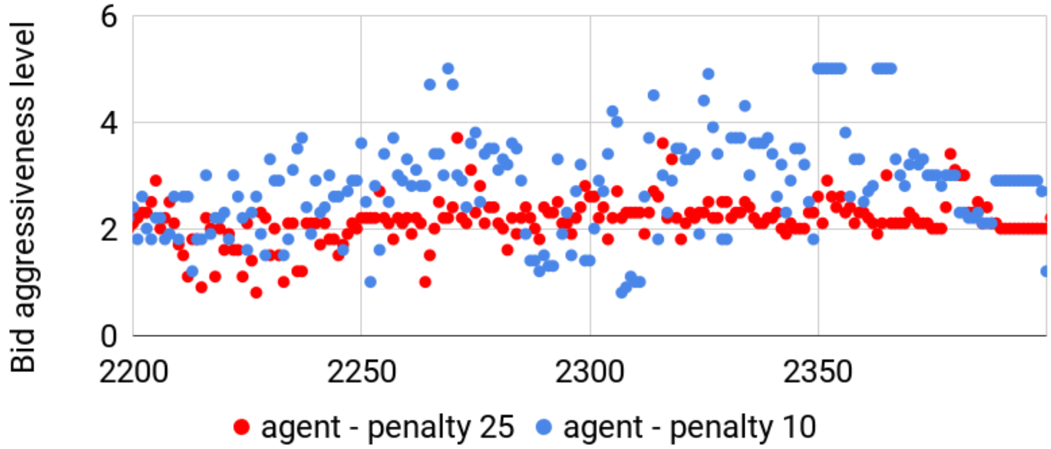

FIGURE 4.8 Impact of penalty function on agent behaviour.

value benefits. In contrast, the agent with a higher penalty coefficient seems more focused on offsetting auction losses.

A critical benefit of the proposed approach is the potential for self management (including self negotiation and self organization) as highlighted in (MARINESCU et al., 2014) when SPs are allowed automatically switch between InPs whenever necessary for orchestration or profitability reasons. Hence, it can be argued that DRL-based agents could self manage VNFs by switching serving InP once the market become detrimental to them (i.e profit loss, presence of collusion, etc.). Sometimes however, there are obstacles toward InP switching. For instance in mobile networks where the number of InPs to switch is limited by cellular coverage. Further, from the simulations above, poor agents in auction settings where collusion is present are the most likely to regularly switch InP in order to ensure their end-users QoS requirements are met. This entails however that the agent must re-train as they are put in a new environment, which may hinder their performance.

4.9 Discussion

4.9.1 Context

It has become common belief that AI will become a focal point, not only for network management, but also for next generation applications that are expected to generate and consume exponential quantities of data, and requiring run-time processing before and after transmission, and in some cases, on their way to the end-user (Zhani et ElBakoury, 2019). This comes as next generation networks are increasingly being described to be independent and decentralized systems on which decisions are taken at different granular levels (Zhani et ElBakoury, 2019 ; Huang et al., 2019), and based on numerous requirements. Recently, the ITU-T Focus Group Technologies for Network 2030 (Fg-net-2030, 2019 ; Li, 2019) have pushed forward the idea of "Manynets", on which they state : *[...] Quite likely there will not be just one, but many public Internets. New technologies further widen the constraints for transmitting packets through the utilization of infrastructure-based wireless, wireless mesh, satellite, fixed line technologies (such as fibre optics), all of which must be accompanied by the fundamental packet transfer solution, while adhering to the underlying ownership relations when traversing those different networks. As a consequence, the end-to-end realization of services across those many internet environments need strong consideration for Network2030 and is an increasing departure from the structures of networks as we see today.*

An example of such requirements, notably for AI-driven applications, is multi-flow synchronization. The latter ensures that generated data from different sources arrive at the destination within a specific interval of time or even at a particular point of time (Zhani et ElBakoury, 2019). For example, paradigms such as federated learning (Konečný et al., 2016), where models are trained on network resources located on edge using local sample patterns and sent to centralized entity to build a shared global model, are leveraged to speed up and improve the distributed learning process (Huang et al., 2019), strengthen security and guarantee privacy.

Several observations motivate the use of auction-based solutions proposed in this work. Firstly, as network resources are distributed on independent domains, it is obvious that a large-scale autonomous cooperation/negotiation mechanism must be in place to ensure access to adequate network resources. Second, with NFV as an enabling technology, the cooperation mechanism must also be able to embed dynamic service requirements in order to achieve multi-domain orchestration. In this regard, we argue that auction and market dynamics are a flexible signal mechanism to identify adequate resource locations, allowing for instance network characteristics (penalty to compensate for low bandwidth, high latency, etc.) to be easily embedded within a bid, aiming to jointly solve the profitability and performance conundrum.

However, given that bidding agents will most likely be powered through AI-enabled algorithms, it is essential to investigate the impact of such agents within the market, and effective methods to counter them. In fact, the Organisation for Economic Co-operation and Development (OECD) reports that there is a particular concern for AI-enabled algorithms to become *a facilitating factor for collusion and may enable new forms of co-ordination that were not observed or even possible before. This is referred to as "algorithmic collusion"* (Oecd, 2017).

4.9.2 Architectural implications for Future Networks

Despite strong auction performance dominance for DRL-based agents compared to other agents after training, two main concerns still need to be addressed. The first concern is to investigate into where the training process will take place. In this regard, the ITU-T Focus Group on Machine Learning for Future Networks has advocated for a sandbox domain, which is an internal operator where machine learning (ML) models can be trained, verified and their effects on the network studied (FG-ML5G, 2019). The second concern relates to the amount of communication between coopeting/colluding agents that may induce further congestion in future networks.

4.9.3 Multiple InPs

While we considered scenarios with a single InP, we hypothesize that considering multiple InPs scenarios may induce several changes to both InPs and SPs strategies. For instance, shifting the level of aggressiveness an SP would display to obtain networking resources given that alternative choices also exist. Naturally, this influences whether and when (in particular, poor) SPs exit the market, as bidding conditions may change favourably or unfavourably. A direct consequence would be that InPs become in a weaker position, especially with the presence of colluding SPs who would in the long run endanger InPs' profits margin. It puts also an increased pressure on the InPs as they must adopt more generous fairness mechanisms to retain self-interested SPs, while severely penalizing colluding SPs. Consequently, this would come at the expense of slightly reduced profits, but still better than a full collusive market. On the other hand, market switching in the context of resource allocation may not be all benefits for SPs. In addition to restarting the process of market price discovery, the possible markets/InPs a given SP can negotiate with may be constrained by its end-user requirements and the incurred penalty's value, assuming it fails to serve its users.

4.10 Conclusion

As one of the major goals of upcoming 5G networks, the need for end-to-end service provisioning has rendered urgent new profitable cooperation and service orchestration mechanisms particularly for multi-domain networks. We thus proposed a market driven in which SPs and InP interact to exchange and orchestrate resources while keeping in mind stringent QoS requirements. Through realistic market scenarios, we analyzed the behaviour of DRL-based agents. Our simulations results have shown DDQN and DQN - driven agents to perform generally well in dynamic network environments, although profitability has been shown to be constrained by budget limitations. From the InP perspective, a competitive auction environment has been shown to be the most profitable. However, in cases where collusion between SPs is present, tradeoffs must be made between ensuring market fairness and harming its profits.

REFERENCES

Abu-Lebdeh, Mohammad (2018). “NFV Management and Orchestration in Large-scale Distributed Systems”. An optional note. Thèse de doct. 1515 Ste-Catherine St. W., EV 7.640, Montreal, QC : Concordia Institute for Information Systems Engineering.

Abu-Lebdeh, Mohammad et al. (2017). “On the Placement of VNF Managers in Large-scale and Distributed NFV Systems”. In : *IEEE Trans. Net. and Service Mgmt.* 14.4, p. 875-889.

Akyildiz, Ian F. et al. (2016). “5G Roadmap : 10 Key Enabling Technologies”. In : *Computer Networks* 106, p. 17-48. issn : 1389-1286. doi : http://dx.doi.org/10.1016/j.comnet.2016.06.010.

Alliance, NGMN (2015). “5G White Paper”. In : *Next Generation Mobile Networks*, p. 1-125.

Ausubel, Lawrence M et Oleg Baranov (2017). “A Practical Guide to The Combinatorial Clock Auction”. In : *The Economic J.* 127.605, F334-f350.

Ausubel, Lawrence M et Peter Cramton (2004). “Vickrey auctions with reserve pricing”. In : *Economic Theory* 23.3, p. 493-505.

Banerjee, Bikramjit et Jing Peng (2006). “RV σ (t) a unifying approach to performance and convergence in online multiagent learning”. In : *Proceedings of* *the fifth international joint conference on Autonomous agents and multiagent systems*, p. 798-800.

Bari, Md Faizul et al. (2013). “Dynamic Controller Provisioning in Software Defined Networks”. In : *Proc. 9th Int. Conf. on Net. and Service Mgmt. (CNSM)*. Ieee, p. 18-25.

Bartlett, Peter L et Shai Ben-David (2002). “Hardness results for neural network approximation problems”. In : *Theoretical Computer Science* 284.1, p. 53-66.

Bhamare, Deval et al. (2016). “A Survey on Service Function Chaining”. In : *J. Net. and Comput. Apps.* 75, p. 138-155.

Bhaskar, Venkataraman, Alan Manning et Ted To (2002). “Oligopsony and Monopsonistic Competition in Labor Markets”. In : *J. of Economic Perspectives* 16.2, p. 155-174.

Boob, Digvijay, Santanu S Dey et Guanghui Lan (2018). “Complexity of training relu neural network”. In : *arXiv preprint arXiv :1809.10787*.

Borjigin, Wuyunzhaola, Kaoru Ota et Mianxiong Dong (2018). “In Broker We Trust : A double-auction approach for resource allocation in NFV markets”. In : *IEEE Transactions on Network and Service Management* 15.4, p. 1322-1333.

Bowling, Michael (2005). “Convergence and no-regret in multiagent learning”. In : *Advances in neural information processing systems*, p. 209-216.

Brandenburger Adam, M et J Nalebuff Barry (1996). *Coopetition.*

Brero, Gianluca, Benjamin Lubin et Sven Seuken (2018). “Combinatorial Auctions via Machine Learning-based Preference Elicitation”. In : *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, p. 128-136.

Brown, George W (1951). “Iterative solution of games by fictitious play”. In : *Activity analysis of production and allocation* 13.1, p. 374-376.

Buyya, Rajkumar, Rajiv Ranjan et Rodrigo N. Calheiros (2010). “InterCloud : Utility-oriented Federation of Cloud Computing Environments for Scaling of Application Services”. In : *Proc. 10th Int. Conf. on Algo. and Archi. for Parallel Process. - Volume Part I*. Busan, Korea : Springer-Verlag, p. 13-31.

Campbell, Andrew T et Klara Nahrstedt (2013). *Building QoS into Distributed Systems : IFIP TC6 WG6. 1 Fifth International Workshop on Quality of Service (IWQOS’97), 21–23 May 1997, New York, USA*. Springer.

Conitzer, Vincent et Tuomas Sandholm (2007). “AWESOME : A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents”. In : *Machine Learning* 67.1-2, p. 23-43.

Cramton, Peter, Yoav Shoham et Richard Steinberg (2006). *Combinatorial Auctions*. The MIT Press. isbn : 0262033429.

Dagnino, Giovanni Battista (2009). “Coopetition Strategy : A New Kind of Interfirm Dynamics for Value Creation”. In : *Coopetition strategy*. Routledge, p. 45-63.

Daniely, Amit, Nati Linial et Shai Shalev-Shwartz (2014). “From average case complexity to improper learning complexity”. In : *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, p. 441-448.

De Sousa, Nathan F Saraiva et al. (2018). "Network Service Orchestration : A Survey". In : *arXiv preprint arXiv :1803.06596*.

Dean, Jeffrey et al. (2012). "Large scale distributed deep networks". In : *Advances in neural information processing systems*, p. 1223-1231.

Dougherty, James, Ron Kohavi et Mehran Sahami (1995). "Supervised and unsupervised discretization of continuous features". In : *Machine Learning Proceedings 1995*. Elsevier, p. 194-202.

Dräxler, Sevil et al. (2017). "SONATA : Service Programming and Orchestration for Virtualized Software Networks". In : *Proc. Int. Conf. on Commun. Wrkshps. (ICC Wrkshps.)* Ieee, p. 973-978.

Duan, Qiang (2010). "Modeling and Analysis of End-to-End Quality of Service Provisioning in Virtualization-based Future Internet". In : *Proc. 19th Int. Conf. on Comput. Commun. and Net. (ICCCN)*. Ieee, p. 1-6.

Etsi (2014). *Network Functions Virtualisation (NFV) : Architectural Framework*. url : https://docbox.etsi.org/ISG/NFV/Open/Publications%5C%5Fpdf/Specs-Reports/NFV%5C%20002v1.2.1%5C%20-%5C%20GS%5C%20-%5C%20NFV%5C%20Architectural%5C%20Framework.pdf.

– (2015). *Network Functions Virtualisation (NFV) Ecosystem : Report on SDN Usage in NFV Architectural Framework*. url : https://www.etsi.org/deliver/etsi%5C%5Fgs/NFV-EVE/001%5C%5F099/005/01.01.01%5C%5F60/gs%5C%5FNFV-EVE005v010101p.pdf.

– (2018). *Network Functions Virtualisation (NFV) Release 3 ; Management and Orchestration ; Report on Architecture Options to Support Multiple Administrative

Domains. URL : https://www.etsi.org/deliver/etsi%5C%5Fgr/NFV-IFA/001%5C%5F099/028/03.01.01%5C%5F60/gr%5C%5FNFV-IFA028v030101p.pdf.

Figueira, Norival et Ram Ramki Krishnan (2015). “SDN Multi-Domain Orchestration and Control : Challenges and Innovative Future Directions”. In : *Proc. Int. Conf. on Comput., Net. and Commun. (ICNC)*. Ieee, p. 406-412.

Gholami, Amir et al. (2018). “Integrated model, batch, and domain parallelism in training neural networks”. In : *Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures*, p. 77-86.

Greenwald, Amy, Keith Hall et Roberto Serrano (2003). “Correlated Q-learning”. In : *Icml*. T. 20. 1, p. 242.

Guerzoni, Riccardo et al. (2017). “Analysis of End-to-End Multi-domain Management and Orchestration Frameworks for Software Defined Infrastructures : An Architectural Survey”. In : *Trans. on Emerging Telecom. Tech.* 28.4.

Habiba, Ummy et Ekram Hossain (2018). “Auction Mechanisms for Virtualization in 5G Cellular Networks : Basics, Trends, and Open Challenges”. In : *IEEE Commun. Surveys & Tuto.*

Hasselt, Hado V (2010). “Double Q-learning”. In : *Advances in Neural Information Processing Systems*, p. 2613-2621.

Hu, Junling et Michael P Wellman (2003). “Nash Q-learning for general-sum stochastic games”. In : *Journal of machine learning research* 4.Nov, p. 1039-1069.

Huang, Tongyi et al. (2019). “A Survey on Green 6G Network : Architecture and Technologies”. In : *IEEE Access* 7, p. 175758-175768.

IETF (2018). *Multi-domain Network Virtualization*. URL : https://tools.ietf.org/html/draft-bernardos-nfvrg-multidomain-04.

ITU (2015). *FG IMT-2020 : Report on Standards Gap Analysis*. Rapp. tech. Fg Imt-2020. International Telecommunication Union. URL : https://www.itu.int/en/ITU-T/focusgroups/imt-2020/Documents/T13-SG13-151130-TD-PLEN-0208!!MSW-E.docx.

IVALDI, Marc et al. (2003). "The economics of tacit collusion". In : *Study Report*.

JIANG, Menglan, Massimo CONDOLUCI et Toktam MAHMOODI (2017). "Network Slicing in 5G : An Auction-based Model". In : *Proc. Int. Conf. on Commun. (ICC)*. Ieee, p. 1-6.

KATSALIS, Kostas, Navid NIKAEIN et Andy EDMONDS (2016). "Multi-domain Orchestration for NFV : Challenges and Research Directions". In : *Proc. Int. Conf. on Ubiquitous Comput. and Commun. and Int. Symp. on Cyberspace and Security (IUCC-CSS)*, Ieee, p. 189-195.

KATSALIS, Kostas, B ROFOEE et al. (2017). "Implementation Experience in Multi-domain SDN : Challenges, Consolidation and Future Directions". In : *Comput. Net.* 129, p. 142-158.

KEARNS, Michael J (1990). *The computational complexity of machine learning*. MIT press.

KIISKI, Annukka (2006). "Impacts of MVNOs on Mobile Data Service Market". In : *Proc. 17th European Regional ITS Conf.*

KONEČNÝ, Jakub et al. (2016). “Federated learning : Strategies for improving communication efficiency”. In : *arXiv preprint arXiv :1610.05492*.

KOURTIS, Michail-Alexandros et al. (2017). “T-NOVA : An Open-Source MANO Stack for NFV Infrastructures”. In : *IEEE Trans. Net. and Service Mgmt.* 14.3, p. 586-602.

LANGE, Stanislav et al. (2015). “Heuristic Approaches to The Controller Placement Problem in Large Scale SDN Networks”. In : *IEEE Trans. Net. and Service Mgmt.* 12.1, p. 4-17.

LEE, Yeng-Ting et Kuan-Ta CHEN (2010). “Is Server Consolidation Beneficial to MMORPG ? A Case Study of World of Warcraft”. In : *3rd Int. Conf. on Cloud Comput. (CLOUD)*. Ieee, p. 435-442.

LI, Richard (2019). *Network 2030 and New IP*. URL : http://www.cnsm-conf.org/2019/files/slides-Richard.pdf.

LITTMAN, Michael L (1994). “Markov Games As A Framework for Multi-agent Reinforcement Learning”. In : *Machine Learning Proc.* Elsevier, p. 157-163.

LITTMAN, Michael L et Peter STONE (2001). “Implicit negotiation in repeated games”. In : *International Workshop on Agent Theories, Architectures, and Languages*. Springer, p. 393-404.

LIVNI, Roi, Shai SHALEV-SHWARTZ et Ohad SHAMIR (2014). “On the computational efficiency of training neural networks”. In : *Advances in neural information processing systems*, p. 855-863.

MAILLÉ, Patrick, Maurizio NALDI et Bruno TUFFIN (2009). “Understanding and Preventing Tacit Collusion Among Telecommunication Operators”. In : *Int. Conf. on Net. Control and Optimization*. Springer, p. 249-263.

MARINESCU, Dan C, Ashkan PAYA et John P MORRISON (2014). “Coalition formation and combinatorial auctions ; applications to self-organization and self-management in utility computing”. In : *arXiv preprint arXiv :1406.7487*.

MASSEY, Patrick et Moore MCDOWELL (2008). *Joint Dominance and Tacit Collusion : An Analysis of The Irish Vodafone/O2 Case and The Implications for Competition and Regulatory Policy*. Rapp. tech. Working Paper Series, UCD Centre for Economic Research.

MIJUMBI, Rashid et al. (2016). “Network Function Virtualization : State-of-the-art and Research Challenges”. In : *IEEE Commun. Surveys & Tuto.* 18.1, p. 236-262.

FG-ML5G (2019). *Unified architecture for machine learning in 5G and future networks*. URL : https://www.itu.int/en/ITU-T/focusgroups/ml5g/Documents/ML5G-delievrables.pdf.

MNIH, Volodymyr et al. (2015). “Human-level Control Through Deep Reinforcement Learning”. In : *Nature* 518.7540, p. 529.

NAE, Vlad et al. (2011). “A New Business Model for Massively Multiplayer Online Games”. In : *SIGSOFT Soft. Engineer. Notes*. T. 36. 5. Acm, p. 271-282.

FG-NET-2030 (2019). *Network 2030 A Blueprint of Technology, Applications and Market Drivers Towards the Year 2030 and Beyond*. URL : https://www.itu.int/en/ITU-T/focusgroups/net2030/Documents/White%5C%5FPaper.pdf.

OECD (2017). *Algorithms and Collusion : Competition Policy in the Digital Age*. URL : www.oecd.org/competition/algorithms-collusion-competition-policy-in-the-digital-age.htm.

Open Network Automation Platform (s. d.). https://www.onap.org/. Accessed : 2018-12-03.

OSSEIRAN, A. et al. (2014). “Scenarios for 5G Mobile and Wireless Communications : The Vision of the METIS Project”. In : *IEEE Commun. Mag.* 52.5, p. 26-35. ISSN : 0163-6804. DOI : 10.1109/mcom.2014.6815890.

PALATTELLA, Maria Rita et al. (2016). “Internet of Things in The 5G Era : Enablers, Architecture, and Business Models”. In : *IEEE J. Sel. Areas in Commun.* 34.3, p. 510-527.

PATHAK, Surya, Mohan P POKHAREL et Sankaran MAHADEVAN (2013). “Hyper-competition, Collusion, Free Riding or Coopetition : Basins of Attraction When Firms Simultaneously Compete and Cooperate”. In : *Nonlinear Dynamics, Psychology, and Life Sciences* 17.1, p. 133-157.

PLA, Albert, Beatriz LOPEZ et Javier MURILLO (2015). “Multi-dimensional Fairness for Auction-based Resource Allocation”. In : *Knowledge-Based Systems* 73, p. 134-148.

POWERS, Rob et Yoav SHOHAM (2005). “New criteria and a new algorithm for learning in multi-agent systems”. In : *Advances in neural information processing systems*, p. 1089-1096.

PRODAN, Radu et Alexandru IOSUP (2016). “Operation Analysis of Massively Multiplayer Online Games on Unreliable Resources”. In : *Peer-to-Peer Net. and Apps.* 9.6, p. 1145-1161.

ROSA, R. V., M. A. S. SANTOS et C. E. ROTHENBERG (2015). “MD2-NFV : The Case for Multi-domain Distributed Network Functions Virtualization”. In : *Int. Conf. and Wrkshps. on Network. Syst. (NetSys)*, p. 1-5. DOI : 10.1109/NetSys.2015.7089059.

SANDHOLM, Tuomas et al. (2005). “CABOB : A Fast Optimal Algorithm for Winner Determination in Combinatorial Auctions”. In : *Management Science* 51.3, p. 374-390.

SCIANCALEPORE, Vincenzo et al. (2016). “A Double-tier MEC-NFV Architecture : Design and Optimisation”. In : *Proc. Conf. on Standards for Commun. and Net. (CSCN),* Ieee, p. 1-6.

SHALLUE, Christopher J et al. (2018). “Measuring the effects of data parallelism on neural network training”. In : *arXiv preprint arXiv :1811.03600.*

SPALL, James C (2005). *Introduction to stochastic search and optimization : estimation, simulation, and control.* T. 65. John Wiley & Sons.

VICKREY, William (1961). “Counterspeculation, Auctions, and Competitive Sealed Tenders”. In : *The Journal of Finance* 16.1, p. 8-37.

WANG, Guodong et al. (2017). “The Controller Placement Problem in Software Defined Networking : A Survey”. In : *IEEE Network* 31.5, p. 21-27.

Wang, Jianglong et al. (2016). “A Multi-domain SDN Scalability Architecture Implementation Based on The Coordinate Controller”. In : *Proc. Int. Conf. on Cyber-Enabled Dist. Comput. and Knowledge Discovery (CyberC)*. Ieee, p. 494-499.

Zawadzki, Erik, Asher Lipson et Kevin Leyton-Brown (2014). “Empirically evaluating multiagent learning algorithms”. In : *arXiv preprint arXiv :1401.8074*.

Zhang, Yang et al. (2013). “Auction Approaches for Resource Allocation in Wireless Systems : A Survey”. In : t. 15. 3. Ieee, p. 1020-1041.

Zhani, Mohamed Faten et Hesham ElBakoury (2019). “FlexNGIA : A flexible Internet architecture for the next-generation tactile Internet”. In : *arXiv preprint arXiv :1905.07137*.

Zhu, Kun et Ekram Hossain (2016). “Virtualization of 5G Cellular Networks As A Hierarchical Combinatorial Auction”. In : *IEEE Trans. on Mob. Comput.* 15.10, p. 2640-2654.

CHAPITRE V

CONCLUSIONS ET TRAVAUX FUTURS

5.1 Conclusions

Cette thèse présente des solutions architecturales et algorithmiques au problème d'approvisionnement des ressources multi-domaines dans les réseaux de nouvelle génération avec comme objectifs principaux l'optimisation des exigences QoS pour les utilisateurs finaux et de la rentabilité pour les opérateurs de réseaux.

Une revue systématique de la littérature pertinente a révélé la complexité du processus d'orchestration des ressources dans les réseaux multi-domaines, en raison des facteurs suivants :

- l'absence d'un consensus architectural guidant les interactions entre des opérateurs ayant des modèles économiques, des infrastructures et mécanismes de gestion de ressources ainsi que des caractéristiques de trafic hétérogènes ;
- l'explosion du trafic de données et l'hétérogénéité des modèles d'accès provoquées par l'émergence d'applications innovantes de nouvelle génération. Il convient de souligner que le rythme de développement des applications est nettement plus rapide que le rythme d'innovation des infrastructures de réseau. En outre, ces applications ont des exigences plus strictes en matière de QoS et de QoE et sont gourmandes en ressources. Il existe donc une marge

d'erreur réduite en termes d'orchestration de ressources ;

- la nature dynamique des applications et des conditions de réseau nécessite l'emploi de stratégies d'orchestration de ressources adaptatives, capables de prendre des décisions optimales à la lumière d'évènements futurs incertains et des préférences des utilisateurs finaux et des opérateurs réseaux. En raison de cette situation, l'optimalité d'une décision d'orchestration de ressources doit être redéfinie.

Dans les réseaux modernes, le déploiement d'applications nécessite la mise en œuvre et la distribution d'une variété de fonctions de réseau, y compris des mécanismes de communication unicast, multicast et interactifs qui exigent un contrôle de qualité de bout en bout, ainsi que la capacité de réguler le traitement des données avant et après la transmission, et dans certains cas pendant leur trajet jusqu'à l'utilisateur final.

Nous pensons que les techniques de gestion de réseau basées sur l'apprentissage coopératif et distribué deviendront la norme pour l'approvisionnement en ressources de réseau à grande échelle afin de prendre en charge la nature dynamique de la configuration du réseau, de permettre l'automatisation du système de bout en bout et de fournir des applications d'utilisateur final plus fiables et plus efficaces.

Dans cette thèse, nous présentons quatre articles de recherche, chacun d'entre eux se concentrant sur un sous-problème distinct du problème d'approvisionnement de ressources dans les réseaux multi-domaines et pour lequel nous proposons des solutions algorithmiques et architecturales efficaces.

Nous examinons d'abord le sujet de l'approvisionnement de services multimédias à valeur ajoutée dans un contexte de réseaux CDNs mono-domaine. Pour ce faire, nous proposons l'article **CPVNF : Cost-Efficient Proactive VNF Placement**

and Chaining for Value-Added Services in Content Delivery Networks dans lequel est fourni :

- une étude de cas illustrant le problème de l'allocation et du chaînage des ressources virtuelles dans un réseau CDN mono-domaine. Nous décrivons les conditions préalables d'une solution potentielle et soulignons en quoi notre travail diffère des articles existants dans le domaine ;
- une modélisation et une formulation ILP du problème ;
- une méthode proactive basée sur le PageRank pour l'allocation et le chaînage des ressources virtuelles en tenant compte des différentes sources de données et des contraintes de QoS ;
- une évaluation expérimentale comparative de notre stratégie et de la solution optimale.

Ce travail nous a conduits à remettre en question l'hypothèse mono-domaine qui prévaut dans la littérature, qui impliquerait que tous les nœuds de traitement entre les sources de données et les utilisateurs finaux seraient sous l'autorité d'un seul opérateur CDN tel qu'Akamai. Compte tenu de la dispersion et de la mobilité des utilisateurs finaux, l'hypothèse mono-domaine est peu plausible pour plusieurs raisons (financière, réglementaire, etc.). Le rejet de l'hypothèse mono-domaine, c'est-à-dire que certains éléments du réseau échappent au contrôle direct de l'opérateur, suggère que le respect des exigences QoS pour les utilisateurs finaux n'est pas assuré, en particulier pour les applications de la prochaine génération dont les besoins de latence sont inférieurs à 10 ms. En outre, nos simulations avec des réseaux à grande échelle révèlent les limites des approches heuristiques standard en ce qui concerne l'évolutivité, l'intelligence du réseau, la flexibilité et le temps de réponse aux décisions face à l'échelle sans précédent des réseaux de nouvelle génération.

À la lumière de ces éléments, nous nous orientons sur les approches d'apprentissage automatique, en particulier l'apprentissage par renforcement profond. Ainsi, le second article intitulé **Towards Reliable Remote Health Monitoring in Fog Computing Networks** combine DRL et DE de manière hybride. En particulier, cet article propose :

- un modèle de plateforme *fog computing* capable d'identifier et d'estimer la disponibilité de modèles d'appareils IoT/*fog*.
- Nous formulons le problème du déploiement d'applications dans les nœuds *fogs* sur lesquels s'exécutent des tâches de traitements provenant des capteurs de patients. L'objectif est de maximiser le nombre de tâches satisfaites, compte tenu des contraintes de latence et en termes de disponibilité pour les nœuds *fogs*.
- En raison de la NP-difficulté du problème, nous proposons un nouvel algorithme d'évolution différentielle auquel nous appliquons un mécanisme de sélection adaptative des opérateurs (AOS) basé sur l'apprentissage par renforcement profond (DRL).
- Les résultats numériques prouvent la supériorité de notre approche en termes de fiabilité par rapport aux solutions de base. De plus, en étudiant l'impact de plusieurs paramètres clés, nous identifions un compromis entre le nombre de nœuds *fog* et les taux de défaillance intrinsèques de ces derniers.

Ce travail emploie DRL en complément à DE afin (1) d'optimiser les décisions de placement de tâches et d'approvisionnement en ressources dans un environnement instable via une sélection améliorée de la stratégie de mutation, et (2) d'accélérer le temps de convergence. La nécessité de suringénierie de la modélisation de l'agent DRL pour obtenir une haute performance d'apprentissage est notée comme le principal inconvénient de cette étude, car étant un obstacle à la généralisation de l'apprentissage. En réalité, nous nous appuyons fortement sur notre connaissance de l'environnement pour guider l'apprentissage de l'agent pour ce travail, ce qui devient peu pratique lorsque l'environnement subit des changements fréquents et importants.

Dès lors, nous proposons dans le troisième article intitulé **DRL-based Green Resource Provisioning for 5G Networks** :

- une modélisation du problème d'allocation de ressources, de façon verte, pour les réseaux sans fil du point de vue des opérateurs de réseaux mobiles en considérant des stations de base MIMO massive et une acquisition imparfaite des informations sur l'état du canal. Notre modèle de système prend également en compte les exigences en termes de débit de données des utilisateurs finaux, l'utilisation des ressources et la maximisation des profits.

- Une formulation du problème d'approvisionnement "verte" de ressources VNF dans les réseaux câblés, à grande échelle, du point de vue des opérateurs de réseaux mobiles. Nous prenons conjointement en compte les impacts environnementaux induits par l'approvisionnement des ressources, le respect des exigences QoS des utilisateurs finaux et l'utilisation des ressources.

- Deux solutions basées sur l'apprentissage par renforcement profond sont proposées pour résoudre les problèmes d'approvisionnement "verte" des ressources VNF et sans fil. Les solutions développées sont conçues pour être parallélisables pour les systèmes à grande échelle.

- Grâce à des évaluations expérimentales réalistes, nous illustrons l'efficacité des approches proposées pour réduire l'empreinte environnementale du réseau tout en satisfaisant les exigences de qualité de service des utilisateurs finaux.

Cette étude intègre des considérations environnementales, devenues des composantes essentielles dans la gestion des réseaux modernes. Nous continuons également à étudier la viabilité du DRL comme stratégie d'allocation des ressources dans les réseaux dynamiques à grande échelle. En particulier, nous nous concentrons sur les contraintes pratiques qui empêchent généralement la généralisation des agents DRL, à savoir la nécessité (1) d'un agent conçu manuellement pour la tâche à accomplir et (2) d'un entraînement coûteux sur-politique (*on-policy*), ne bénéficiant donc pas des progrès substantiels réalisés sur l'entraînement hors politique (*off-policy*). D'autant plus qu'en pratique, les problèmes d'approvisionnement en ressources de réseau, en particulier dans les réseaux sans fil, sont presque toujours multiobjectifs et extrêmement dynamiques (fréquents changements topologiques virtuels et réels, variations du rythme et du volume des demandes des utilisateurs, etc.). Dans ce contexte, nous employons pour ce travail des heuristiques parallélisables en complément du DRL afin de créer un environnement propice à un apprentissage efficace et généralisable.

Le dernier article, **Market Driven Multi-domain Network Service Orchestration in 5G Networks** examine l'influence de la profitabilité sur la collaboration et la prise de décision concernant l'orchestration des ressources dans un scénario de réseaux multi-domaines. À cette fin, nous explorons plusieurs mécanismes de marché libre et analysons les dynamiques de coopération et de rivalité entre les opérateurs réseau qui cherchent à satisfaire les exigences QoS de leurs utilisateurs finaux. L'article propose :

- un exposé notant l'importance des interactions multi-domaines pour l'approvisionnement de services de bout en bout, à travers le cas d'utilisation MMOG (Massive Multiplayer Online Gaming).

- la modélisation d'une plateforme d'interaction multi-domaine, basée sur des mécanismes de libre marché, pour l'allocation de ressources entre un fournisseur d'infrastructures et plusieurs fournisseurs de services candidats. Nous formulons le problème d'allocation de ressources associé, où le fournisseur d'infrastructure et les fournisseurs de service visent à maximiser leurs profits en vendant/achetant des ressources sur le marché. Nous définissons différents environnements de marché, où les fournisseurs de services peuvent être en concurrence et/ou coopérer dans le processus d'appel d'offres, et où le fournisseur d'infrastructure peut décider d'appliquer, ou non, l'équité entre les fournisseurs de services.

- En raison de la complexité du système considérant avec plusieurs fournisseurs de services, nous proposons une approche d'apprentissage par renforcement profond multiagent distribué, où nous équipons chaque fournisseur de services d'un agent d'apprentissage capable de percevoir l'environnement et de prendre des actions stratégiques pour gagner des enchères, donc de satisfaire ses exigences de qualité de service et d'augmenter son profit. De plus, les agents de différents fournisseurs de services peuvent échanger des informations dans des scénarios de coopération afin d'améliorer leurs profits mutuels.

- À travers des scénarios d'expérimentation, nous avons formé des agents de fournisseurs de services et évalué les performances du fournisseur d'infrastructure et des fournisseurs de services dans différents scénarios de marché. Il a été démontré qu'un marché entièrement concurrentiel (sans coopération) est le plus rentable pour le fournisseur d'infrastructure. En revanche, un marché coopératif est en moyenne le plus lucratif pour les fournisseurs de services. De plus, les agents à apprentissage double-deep-Q surpassent les autres agents à apprentissage et sans apprentissage en termes de profits pour les fournisseurs de services. Enfin, l'impact de certains paramètres est souligné.

5.2 Limites et travaux de recherche futurs

Certaines approches proposées dans cette thèse souffrent de quelques limites, ce qui ouvre la porte à plusieurs améliorations et pistes de recherche que nous détaillons ci-dessous.

- Les méthodes d'apprentissage centralisées souvent utilisées dans cette thèse offrent l'avantage de la simplicité, mais elles présentent un problème d'évolutivité en termes de performances d'apprentissage lorsque la taille du réseau et le nombre de dispositifs et d'utilisateurs finaux augmentent. Dans le contexte des réseaux multi-domaines, l'adoption d'une technique d'apprentissage centralisée crée des problèmes de sécurité et de confidentialité. Plus précisément, parce qu'elle risque de révéler des informations internes cruciales des domaines réseau gérés par des opérateurs concurrents, expliquant leur réticence. En outre, la précision d'un modèle d'apprentissage global est très variable, surtout si l'on tient compte de l'hétérogénéité des conditions réseau dans chaque domaine. Comme piste de recherche, nous envisageons l'utilisation de méthodes d'apprentissage décentralisées à large échelle, en particulier l'apprentissage par fédération avec des modèles locaux hétérogènes.
- Pour l'article **DRL-based Green Resource Provisioning for 5G Networks**, nous notons les écarts de performances en fonction de l'échelle du réseau, reflétant la difficulté à maintenir un apprentissage optimal dans des environnements complexes tels que les réseaux sans fil. Puisque les modèles ML sont basés sur des modèles de données, un environnement défini par des exigences de contrôle de la mobilité et des interférences entraîne inévi-

tablement une dérive du modèle qui requiert une surveillance continue des performances du modèle pour déclencher des actions correctives telles que l'optimisation des hyperparamètres (HPO) ou le réentraînement du modèle. Nous souhaitons explorer le potentiel de deux solutions précédentes dans nos travaux futurs.

BIBLIOGRAPHIE

AGARWAL, Rishabh et al. (2021). “Deep Reinforcement Learning at the Edge of the Statistical Precipice”. In : *Advances in Neural Information Processing Systems.*

ALAM, Iqbal et al. (2020). “A survey of network virtualization techniques for internet of things using sdn and nfv”. In : *ACM Computing Surveys (CSUR)* 53.2, p. 1-40.

AREZOUMAND, Saeed et al. (2017). “MD-IDN : Multi-domain intent-driven networking in software-defined infrastructures”. In : *2017 13th International Conference on Network and Service Management (CNSM)*. IEEE, p. 1-7.

ARULKUMARAN, Kai et al. (2017a). “Deep reinforcement learning : A brief survey”. In : *IEEE Signal Processing Magazine* 34.6, p. 26-38.

– (2017b). “Deep reinforcement learning : A brief survey”. In : *IEEE Signal Processing Magazine* 34.6, p. 26-38.

BARAKABITZE, Alcardo Alex et al. (2020). “5G network slicing using SDN and NFV : A survey of taxonomy, architectures and future challenges”. In : *Computer Networks* 167, p. 106984.

BARANDA, Jorge et al. (2018). “Orchestration of end-to-end network services in the 5G-crosshaul multi-domain multi-technology transport network”. In : *IEEE Communications Magazine* 56.7, p. 184-191.

BELBEKKOUCHE, Abdeltouab, Md. Mahmud HASAN et Ahmed KARMOUCH (2012). “Resource Discovery and Allocation in Network Virtualization”. In : *IEEE Communications Surveys & Tutorials* 14.4, p. 1114-1128. ISSN : 1553-877X. DOI : 10.1109/SURV.2011.122811.00060.

BENZEKKI, Kamal, Abdeslam EL FERGOUGUI et Abdelbaki ELBELRHITI ELALAOUI (2016). “Software-defined networking (SDN) : a survey”. In : *Security and communication networks* 9.18, p. 5803-5833.

BERKHIN, Pavel (2005). “A survey on PageRank computing”. In : *Internet mathematics* 2.1, p. 73-120.

BERNARDOS, Carlos J. et al. (avr. 2019). *Network Virtualization Research Challenges*. Request for Comments RFC 8568. Internet Engineering Task Force. 42 p. DOI : 10.17487/RFC8568. URL : https://datatracker.ietf.org/doc/rfc8568 (visité le 10/12/2022).

BIANCHINI, Monica, Marco GORI et Franco SCARSELLI (2005). “Inside pagerank”. In : *ACM Transactions on Internet Technology (TOIT)* 5.1, p. 92-128.

BOUCADAIR, Mohamed, Christian JACQUENET et Luis M. CONTRERAS (13 fév. 2015). *Requirements for Automated (Configuration) Management*. Internet Draft draft-boucadair-network-automation-requirements-05. Internet Engineering Task Force. 19 p. URL : https://datatracker.ietf.org/doc/draft-boucadair-network-automation-requirements (visité le 10/12/2022).

BRADLEY, Stephen P, Arnoldo C HAX et Thomas L MAGNANTI (1977). *Applied mathematical programming*. Addison-Wesley.

BRIN, Sergey et Lawrence PAGE (1998). “The anatomy of a large-scale hypertextual Web search engine”. In : *Computer Netw. and ISDN Syst.* 30.1, p. 107-117.

CHAN, Stephanie CY et al. (2020). “Measuring the Reliability of Reinforcement Learning Algorithms”. In : *International Conference on Learning Representations, Addis Ababa, Ethiopia.*

CHUNG, Fan (2014). “A Brief Survey of PageRank Algorithms.” In : *IEEE Trans. Netw. Sci. Eng.* 1.1, p. 38-42.

CLAYMAN, Stuart et al. (2012). “Stability in Dynamic Networks”. In : p. 6.

DAS, Swagatam et Ponnuthurai Nagaratnam SUGANTHAN (2010). “Differential evolution : A survey of the state-of-the-art”. In : *IEEE Trans. Evolutionary Comput.* 15.1, p. 4-31.

DE SOUSA, Nathan F Saraiva et al. (2018). “Network Service Orchestration : A Survey”. In : *arXiv preprint arXiv :1803.06596.*

DEVLIC, Alisa et al. (2017). “NESMO : Network slicing management and orchestration framework”. In : *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, p. 1202-1208.

FISCHER, Andreas et al. (hiver 2013). “Virtual Network Embedding : A Survey”. In : *IEEE Communications Surveys & Tutorials* 15.4, p. 1888-1906. ISSN : 1553-877X. DOI : 10.1109/SURV.2013.013013.00155. URL : http://ieeexplore.ieee.org/document/6463372/ (visité le 11/12/2022).

François-Lavet, Vincent et al. (2018). “An introduction to deep reinforcement learning”. In : *Foundations and Trends® in Machine Learning* 11.3-4, p. 219-354.

Grzybowski, Lukasz et Chiraz Karamti (2010). “Competition in mobile telephony in France and Germany”. In : *The Manchester School* 78.6, p. 702-724.

Haveliwala, Taher (1999). *Efficient computation of PageRank*. Rapp. tech. Stanford.

Hawilo, Hassan et al. (2014). “NFV : state of the art, challenges, and implementation in next generation mobile networks (vEPC)”. In : *IEEE network* 28.6, p. 18-26.

Henderson, Peter et al. (2018). “Deep reinforcement learning that matters”. In : *Proceedings of the AAAI conference on artificial intelligence*. T. 32. 1.

Herrera, Juliver Gil et Juan Felipe Botero (2016). “Resource allocation in NFV : A comprehensive survey”. In : *IEEE Transactions on Network and Service Management* 13.3, p. 518-532.

Hosseini, Mohammad-Parsa et al. (2020). “Deep learning architectures”. In : *Deep learning : concepts and architectures*. Springer, p. 1-24.

IBM (2022). *IBM ILOG CPLEX Optimizer*. url : https://www.ibm.com/analytics/cplex-optimizer (visité le 28/06/2022).

Ietf (2018). *Multi-domain Network Virtualization*. url : https://tools.ietf.org/html/draft-bernardos-nfvrg-multidomain-04.

Ismail, Muhammad et al. (2015). “A Survey on Green Mobile Networking : From The Perspectives of Network Operators and Mobile Users”. In : *IEEE Communications Surveys & Tutorials* 17.3, p. 1535-1556. issn : 1553-877X. doi : 10.1109/COMST.2014.2367592.

Jordan, Michael I et Tom M Mitchell (2015). “Machine learning : Trends, perspectives, and prospects”. In : *Science* 349.6245, p. 255-260.

JuMP (2022). *JuMP*. url : https://jump.dev/ (visité le 28/06/2022).

Kaelbling, Leslie Pack, Michael L Littman et Andrew W Moore (1996). “Reinforcement learning : A survey”. In : *Journal of artificial intelligence research* 4, p. 237-285.

Kiiski, Annukka (2006). “Impacts of MVNOs on Mobile Data Service Market”. In : *Proc. 17th European Regional ITS Conf.*

Krämer, Jan, Lukas Wiewiorra et Christof Weinhardt (2013). “Net neutrality : A progress report”. In : *Telecommunications Policy* 37.9, p. 794-813.

LeCun, Yann, Yoshua Bengio et Geoffrey Hinton (2015). “Deep learning”. In : *nature* 521.7553, p. 436-444.

Li, Yuxi (2017). “Deep reinforcement learning : An overview”. In : *arXiv preprint arXiv :1701.07274*.

Liu, Weibo et al. (2017). “A survey of deep neural network architectures and their applications”. In : *Neurocomputing* 234, p. 11-26.

LP_solve (2022). *LP_solve*. url : http://lpsolve.sourceforge.net/5.5/ (visité le 28/06/2022).

MIJUMBI, Rashid et al. (2016). “Management and Orchestration Challenges in Network Functions Virtualization”. In : *IEEE Commun. Mag.* 54.1, p. 98-105.

MOUSAVI, Seyed Sajad, Michael SCHUKAT et Enda HOWLEY (2016). “Deep reinforcement learning : an overview”. In : *Proceedings of SAI Intelligent Systems Conference*. Springer, p. 426-440.

ODLYZKO, Andrew (2009). “Network neutrality, search neutrality, and the never-ending conflict between efficiency and fairness in markets”. In : *Review of Network Economics* 8.1.

PANWAR, Shivendra (2020). “Breaking the millisecond barrier : Robots and self-driving cars will need completely reengineered networks”. In : *IEEE Spectrum* 57.11, p. 44-49.

PULP (2022). *Optimization with PuLP*. URL : https://coin-or.github.io/pulp/ (visité le 28/06/2022).

SAAD, Walid, Mehdi BENNIS et Mingzhe CHEN (mai 2020). “A Vision of 6G Wireless Systems : Applications, Trends, Technologies, and Open Research Problems”. In : *IEEE Network* 34.3, p. 134-142. ISSN : 1558-156X. DOI : 10.1109/MNET.001.1900287.

SHAFIQUE, Muhammad et al. (2018). “An overview of next-generation architectures for machine learning : Roadmap, opportunities and challenges in the IoT era”. In : *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, p. 827-832.

Sharma, Mudita et al. (2019). “Deep reinforcement learning based parameter control in differential evolution”. In : *Proc. Genetic Evolution. Comput. Conf. (GECCO)*, p. 709-717.

Shen, Xuemin et al. (2020). “AI-Assisted Network-Slicing Based Next-Generation Wireless Networks”. In : *IEEE Open Journal of Vehicular Technology* 1, p. 45-66. issn : 2644-1330. doi : 10.1109/OJVT.2020.2965100.

Storn, Rainer et Kenneth Price (1997). “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces”. In : *Journal of global optimization* 11.4, p. 341-359.

Sutton, Richard S et Andrew G Barto (2018). *Reinforcement learning : An introduction*. MIT press.

Taleb, Tarik et al. (2019). “On multi-domain network slicing orchestration architecture and federated resource control”. In : *IEEE Network* 33.5, p. 242-252.

Toumi, Nassima et al. (mars 2021). “On Cross-Domain Service Function Chain Orchestration : An Architectural Framework”. In : *Computer Networks* 187, p. 107806. issn : 13891286. doi : 10.1016/j.comnet.2021.107806. url : https://linkinghub.elsevier.com/retrieve/pii/S1389128621000013 (visité le 10/12/2022).

Velasco, Luis et al. (oct. 2021). “End-to-End Intent-Based Networking”. In : *IEEE Communications Magazine* 59.10, p. 106-112. issn : 1558-1896. doi : 10.1109/MCOM.101.2100141.

World Economic Forum (2022). World Economic Forum. URL : https://www.weforum.org/reports/global-information-technology-report-2012/ (visité le 12/12/2022).

ZHANG, Chuangchuang et al. (2019). “Cost Efficient and Low-Latency Network Service Chain Deployment Across Multiple Domains for SDN”. In : *IEEE Access* 7, p. 143454-143470. ISSN : 2169-3536. DOI : 10.1109/ACCESS.2019.2944874.

ZHANG, Jingjing et Nirwan ANSARI (2011). “On assuring end-to-end QoE in next generation networks : challenges and a possible solution”. In : *IEEE Communications Magazine* 49.7, p. 185-191.