UNIVERSITÉ DU QUÉBEC À MONTRÉAL

ORDONNANCEMENT DES TÂCHES ET ALLOCATION DES RESSOURCES DANS L'INFORMATIQUE DE PÉRIPHÉRIE

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

DAMOULAY IHSANE

MAI 2023

REMERCIEMENTS

Je tiens à exprimer ma gratitude à mon directeur de thèse, Monsieur Driouch Elmahdi. Je le remercie de m'avoir encadré, guidé, aidé, conseillé et soutenu.

J'adresse mes sincères remerciements à tous les professeurs et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions durant mes recherches.

Je remets un grand merci à mes très chers parents qui ont toujours été là pour moi. Je tiens à remercier aussi mon conjoint, ma sœur et mon frère pour leurs encouragements.

Je remercie tous mes amis et particulièrement Benyerbah Hanaa, qui ont toujours été là pour moi. Leur soutien inconditionnel et leurs encouragements m'ont été d'une grande aide.

Enfin, je tiens à exprimer ma reconnaissance à toute ma grande famille et à tous ceux qui ont été toujours là pour moi.

RÉSUMÉ

Les avancées technologiques récentes au niveau de la recherche et du développement des réseaux sans fil ont permis une amélioration considérable des performances liées aux communications entre les différents utilisateurs. En effet, l'utilisation de la technologie émergente de l'informatique de périphérie présente un très grand nombre d'avantages dont celui de rapprocher les services informatiques des utilisateurs finaux afin de minimiser les délais de réponse. Toutefois, afin de profiter pleinement de ces avantages, plusieurs défis doivent surmonter tels que l'allocation efficace des ressources, la priorisation des tâches des différents utilisateurs par le biais d'ordonnancement ainsi que leur traitement avant leurs date d'échéance. Dans ce contexte, ce mémoire vise à maximiser le nombre des utilisateurs qui peuvent décharger leurs tâches à la périphérie du réseau. Plus précisément, nous nous attaquons à deux problématiques, soient (i) le partage dynamique du spectre entre des communications primaires et secondaires grâce à la technologie de la radio cognitive et (ii) l'ordonnancement des tâches au niveau du serveur périphérique tout ben respectant leurs échéances strictes. Basés sur plusieurs approches de conception différentes, nous développons des algorithmes qui tentent de résoudre conjointement les deux problématiques annoncées en proposant différents compromis entre les performances obtenues et la complexité algorithmique. Les résultats des simulations présentés dans ce mémoire comparent les performances des algorithmes proposés à deux benchmarks, dont celui qui permet d'obtenir la solution optimale. Ces résultats prouvent aussi la quasi-optimalité de l'algorithme basé sur la méta-heuristique génétique.

MOTS-CLÉS : l'informatique de périphérie, partage du spectre, ordonnancement, radio cognitive, allocation des ressources.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

Tableau Page

LISTE DES FIGURES

CHAPITRE I

INTRODUCTION GÉNÉRALE

1.1 Mise en contexte

Le développement de l'internet des objets (IoT) a connu une croissance considérable avec le progrès des technologies de l'information et de la communication. Plusieurs dispositifs sans fil (en anglais wireless devices, WDs) sont destinés à être connectés via l'internet pour collecter et échanger des informations afin de fournir de nombreuses applications, telles que les soins de santé, les applications environnementales, la domotique (en anglais Home automation), les villes intelligentes, etc. Cependant, en raison des ressources limitées des WDs, l'implémentation des applications IoT dans les scénarios à forte intensité de calcul et sensibles aux délais reste un défi important.

Plusieurs technologies prometteuses ont été mises au point pour relever ces enjeux, parmi lesquelles on retrouve l'informatique de périphérie (en anglais edge computing). Il s'agit d'une technologie qui fournit un environnement informatique distribué pour l'hébergement d'applications et de services. Elle a également la capacité de stocker et de traiter le contenu à proximité des utilisateurs pour un temps de réponse plus rapide et pour une économie des ressources des appareils de manière significative (He et Dou, 2020).

Alors que la demande d'accès sans fil des WDs continue de croitre, la disponibilité d'une bande passante suffisante reste un défi pour l'informatique de périphérie également. La technologie de la radio cognitive (RC) peut constituer une solution viable pour surmonter le manque de bande passante. En effet, les réseaux à radio cognitive sont capables de gérer d'une façon plus adaptée les ressources radio, dont les bandes fréquentielles, grâce aux techniques d'accès dynamique au spectre et de contrôle de puissance (Akyildiz *et al.*, 2008). Par conséquent, les fonctions de gestion du spectre dans l'informatique de périphérie permettront de relever les défis d'allocation des ressources et d'améliorer les exigences des différentes applications en termes de qualité de service (QdS).

1.2 Motivation et problématique

Le déploiement de la cinquième génération des réseaux sans fil (5G) vise à garantir une QdS élevée aux utilisateurs, grâce à l'amélioration des performances globales du réseau, en termes de débit, de délai et de densité. La 5G se présente, pas seulement comme une nouvelle technologie d'accès, mais aussi comme un concept de réseau centré sur l'utilisateur qui vise à répondre aux exigences des applications (Shah *et al.*, 2018). La proximité des services des utilisateurs, grâce à l'informatique de périphérie, a permis de faciliter le traitement surtout pour les applications qui nécessitent des calculs complexes et des algorithmes gourmands en ressources de calcul.

L'interaction directe entre les appareils mobiles et les serveurs périphériques par le biais de communications sans fil offre la possibilité de prendre en charge des applications à très faible latence, de prolonger la durée de vie de la batterie des appareils et d'améliorer la qualité de vie des utilisateurs (Mao *et al.*, 2017). Le fait d'acheminer les demandes des WDs à la périphérie du réseau permet d'amé-

liorer la disponibilité des services car les dispositifs connectés n'ont pas à attendre qu'une plate-forme hautement centralisée, en l'occurrence le nuage (utilisé souvent sous son appellation anglaise cloud), leur fournisse ces services. De plus, ils ne sont pas non plus restreints par les ressources limitées de l'informatique mobile traditionnelle (Yousefpour *et al.*, 2019).

Pourtant, malgré tous ces avantages, l'informatique de périphérie ne permet pas de résoudre tous les défis des réseaux sans fil. Plusieurs facteurs peuvent impacter d'une façon directe ou indirecte le délai de traitement des tâches des utilisateurs déchargées vers la périphérie du réseau. Dans plusieurs articles de recherche, ces facteurs ont été étudiés et traités de différentes manières, l'objectif étant toujours de minimiser le temps de traitement global de tous les utilisateurs qui souhaitent décharger leurs tâches à la périphérie du réseau. Des critères comme le délai d'échéance, l'interférence, la capacité du serveur périphérique, etc.., constituent des contraintes qui rendent l'optimisation de ce délai de traitement une tâche complexe. De plus, la priorité des utilisateurs par rapport à l'importance de leurs messages est aussi un facteur qui impacte le délai de traitement. Par exemple, un message différé pour une application d'évitement de collision peut être inutile s'il est délivré après que le conducteur ait déjà réagi à la situation (Shah *et al.*, 2018).

Le déchargement de l'ensemble des tâches de calcul sur un serveur périphérique entraine plusieurs enjeux qui sont souvent conflictuels. En effet, le déchargement nécessite des communications supplémentaires qui peuvent entrainer parfois des délais additionnels importants. Par conséquent, certaines tâches se voient obligées d'être exécutées localement. D'un autre côté, la gestion des tâches de calcul déchargées au niveau des serveurs périphériques a un impact direct sur les délais de traitement. Ces serveurs doivent procéder à une sélection et à un ordonnancement des tâches afin de garanti une latence optimale et un respect des échéances.

Un autre enjeu majeur de l'opération de déchargement concerne l'allocation des ressources disponibles au niveau du réseau aux différents utilisateurs qui désirent envoyer leurs tâches. La principale ressource partagée entre ces derniers est la ressource fréquentielle. En effet, l'instant d'arrivée de la tâche déchargée au niveau du serveur dépend fortement de la bande fréquentielle allouée. Par conséquent, l'ordonnancement des tâches par le serveur de périphérie se voit aussi grandement influencé par cette allocation fréquentielle. Il est à noter que la grande majorité des travaux de recherche connexe ignore cette problématique d'ordonnancement en considérant que le traitement de toutes les tâches commence au même instant.

1.3 Contribution

Afin d'assurer une meilleure utilisation des ressources fréquentielles du réseau, nous considérons que les utilisateurs qui détiennent une licence pour l'utilisation du spectre fréquentiel, dits aussi utilisateurs primaires, «acceptent» de partager ce dernier avec d'autres utilisateurs sans licence, appelés aussi utilisateurs secondaires. En effet, les transmissions des secondaires sont contraintes à respecter un seuil d'interférence afin de ne pas déranger les transmissions des primaires. En d'autres termes, les secondaires doivent fonctionner d'une manière quasi-transparente du point de vue des primaires. **L'objectif de ce projet de maîtrise est de maximiser le nombre des utilisateurs secondaires qui déchargeront leurs tâches vers la périphérie du réseau sans que leurs délai de traitement dépassent leur délai d'échéance.** Pour cela, nous étudions l'optimisation des décisions suivantes : (i) le déchargement des tâches, (ii) le partage du spectre et (iii) l'ordonnancement des tâches au niveau du serveur de la périphérie. Nos contributions détaillées peuvent être résumées comme suit :

1. Nous avons élaboré un algorithme heuristique qui maximise le nombre d'uti-

lisateurs secondaires qui vont décharger leurs tâches vers le serveur périphérique, sur la base d'un choix de bande de fréquence qui donne un délai de transmission minimal pour chaque tâche. Nous avons appliqué également l'algorithme hongrois pour résoudre le problème d'affectation des utilisateurs secondaire aux bandes de fréquence. Ces deux heuristiques traitent de manière successive les problèmes de déchargement et d'ordonnancement des tâches.

2. Nous avons conçu une deuxième solution algorithmique basée sur la métaheuristique génétique qui permet de résoudre différemment le compromis performance/complexité algorithmique en explorant intelligemment l'espace de recherche.
3. Nous avons aussi conçu un algorithme optimal qui repose sur la force brute. Enfin, nous avons réalisé plusieurs simulations qui nous ont permis d'évaluer et de comparer les performances de toutes les solutions proposées dans ce mémoire et de montrer le degré d'efficacité qui change d'un algorithme à l'autre.

1.4 Organisation du mémoire

Après la présentation de notre projet de recherche, le chapitre II offre une vue d'ensemble de l'informatique de périphérie, de ses avantages et de certains paradigmes périphériques, et évoque brièvement la technologie de la radio cognitive. Dans le chapitre III, nous exposons les travaux de la littérature qui se sont intéressés aux problématiques de déchargement des tâches vers les serveurs périphériques, de partage du spectre et également d'ordonnancement des tâches. Nous présentons ensuite dans le chapitre IV le modèle du système étudié, ainsi que la formulation du problème de maximisation du nombre d'utilisateurs secondaires qui déchargeront leurs tâches sur le serveur périphérique. Dans ce même chapitre, nous présentons

les algorithmes proposés et évaluons leurs performances tout en et analysant les résultats de cette évaluation. Enfin, le chapitre V conclut ce mémoire.

CHAPITRE II

CONCEPTS PRÉLIMINAIRES

2.1 Informatique de périphérie

2.1.1 Introduction

Malgré son large succès, l'informatique en nuage (en anglais cloud computing) n'est plus une solution universelle. L'un de ces principaux problèmes est la centralisation des ressources. Cela se traduit par une séparation accrue entre les appareils des utilisateurs et les centres de données du nuage, ce qui entraine des niveaux de latence très importants (Roman *et al.*, 2018). Au fur et à mesure de l'émergence d'applications sensibles au délai et à l'emplacement (telles que la surveillance des patients, la fabrication en temps réel, les voitures à conduite autonome, les troupeaux de drones ou l'assistance cognitive), le nuage distant n'est plus en mesure de répondre aux exigences de latence ultra-faible de ces applications, de fournir des services sensibles à l'emplacement ou de s'adapter à l'ampleur des données que ces applications produisent (Yousefpour *et al.*, 2019).

Pour ces raisons, au cours des dernières années, de nouveaux paradigmes ont vu le jour, tels que l'informatique en brouillard (en anglais fog computing), l'informatique en périphérie mobile (en anglais mobile edge computing), et l'informatique en nuage mobile (en anglais mobile cloud computing). Le dénominateur commun

de ces paradigmes est le déploiement de capacités de calcul et de stockage à la périphérie du réseau. La plupart des paradigmes de périphérie suivent la structure illustrée à la figure 2.1. Les centres de données des périphéries, qui sont détenus et déployés par des fournisseurs d'accès, mettent en œuvre une infrastructure de virtualisation multi-locataires. Tout client peut ainsi utiliser les services de ces centres de données. En outre, si les centres de données périphériques peuvent agir de manière autonome et coopérer entre eux, ils ne sont pas autant déconnectés du nuage traditionnel. Il est donc possible de créer une architecture hiérarchique à plusieurs niveaux, interconnectée par un réseau d'infrastructure de réseau (Roman *et al.*, 2018).

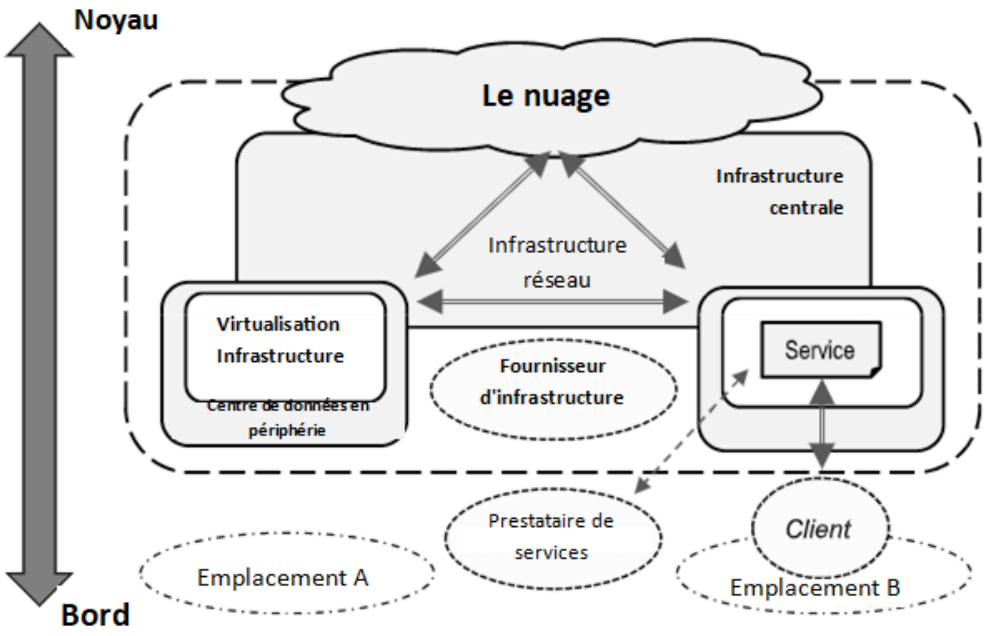

FIGURE 2.1 – Structure fonctionnelle des différents paradigmes de la périphérie (Roman *et al.*, 2018)

2.1.2 Vue d'ensemble des paradigmes de la périphérie

— L'informatique en brouillard

L'informatique en brouillard comble le fossé entre le nuage et les dispositifs finaux en permettant le calcul, le stockage, la mise en réseau et la gestion des données sur des nœuds de réseau à proximité des clients (Yousefpour *et al.*, 2019). Parmi les services, on peut citer les systèmes d'analyse hiérarchique des données massives (en anglais Big Data) et les systèmes de gestion d'infrastructures intelligentes (par exemple, parcs éoliens, feux de circulation, etc.)(Bonomi *et al.*, 2012). Plusieurs études ont examiné comment ce paradigme pourrait être utilisé pour mettre en œuvre d'autres types de services pour les appareils mobiles à ressources limitées, comme la minimisation des interférences et de la latence, et pour les communications véhiculaires, telles que les systèmes de stationnement partagés (Roman *et al.*, 2018).

L'informatique en brouillard est définie par l'OpenFog Consortium [1] comme une architecture horizontale de niveau système qui distribue les fonctions de calcul, de stockage, de contrôle et de mise en réseau plus près des utilisateurs. La plateforme horizontale de l'informatique en brouillard permet de distribuer les fonctions informatiques entre différentes plateformes et industries, alors qu'une plateforme verticale favorise des applications distinctes (Brito *et al.*, 2017). En plus de faciliter une architecture horizontale, l'informatique en brouillard fournit une plateforme flexible pour répondre aux besoins des opérateurs et des utilisateurs en matière de données (Yousefpour *et al.*, 2019).

L'informatique en brouillard peut être intégrée aux technologies mobiles sous la forme de réseaux d'accès radio (en anglais radio access networks ; RAN), pour

1. https ://opcfoundation.org/markets-collaboration/openfog/

former ce que l'on appelle le fog RAN (F-RAN). Les ressources informatiques sur les F-RAN peuvent être utilisées pour la mise en cache à la périphérie du réseau, ce qui permet une récupération plus rapide du contenu et une réduction de la charge sur le réseau principal. Le F-RAN fait notamment partie des technologies importantes liées au déploiement de la 5G (Hung *et al.*, 2015). L'informatique en brouillard ouvre la voie à la prise en charge des dispositifs avec différentes technologies d'accès pour communiquer entre le nœud et l'appareil final (Dolui et Datta, 2017).

— L'informatique mobile (en anglais mobile computing)

Le concept de l'informatique mobile consiste à effectuer des opérations informatiques à l'aide de dispositifs mobiles et portables, tels que des ordinateurs portables, des tablettes ou des téléphones mobiles. L'informatique mobile peut être utilisée pour créer des applications contextuelles omniprésentes, telles que des rappels basés sur l'emplacement. La puissance de l'informatique mobile provient de son architecture informatique distribuée. Les dispositifs bénéficient ainsi de services de stockage et de calcul dans un contexte totalement décentralisé qui ne repose pas sur la présence d'une architecture centralisée pour fonctionner (Yousefpour *et al.*, 2019).

Les technologies de l'informatique mobile ont atteint leur sommet avant l'informatique en nuage, grâce à un environnement caractérisé par une faible puissance de traitement et une connectivité réseau intermittente et réduite. Un grand nombre de défis fondamentaux tels que la mobilité des utilisateurs, l'hétérogénéité des réseaux et la faible largeur de bande de l'informatique mobile ont été évoqués. Ces problèmes ont été résolus par des avancées telles que des matériels et des protocoles de transmission et de mise en cache robustes, ainsi que des algorithmes de compression.

En raison de l'évolution des exigences des appareils connectés des consommateurs, l'informatique mobile seule ne permet pas de relever de nombreux défis informatiques récents(Yousefpour *et al.*, 2019). Parmi les majeurs inconvénients de l'informatique mobile, on retrouve les contraintes reliées aux ressources limitées, l'équilibre entre l'autonomie et l'interdépendance (prévalant dans toutes les architectures distribuées), le délai de communication, et la nécessité pour les clients mobiles de s'adapter efficacement à des environnements changeants.

— L'informatique en périphérie mobile (en anglais Mobile edge computing)

Le terme d'informatique en périphérie mobile a été utilisé pour la première fois pour décrire l'exécution de services à la périphérie du réseau en 2013, lorsqu'IBM et Nokia Siemens Network ont présenté une plateforme qui pouvait exécuter des applications au sein d'une station de base mobile. Ce concept initial n'avait qu'une portée locale, et ne prenait pas en compte d'autres aspects tels que la migration des applications, la coopération, etc. (Roman *et al.*, 2018). L'informatique en périphérie mobile a acquis sa signification actuelle en 2014, lorsque l'European Telecommunications Standards Institute (ETSI) [2] a lancé l'Industry Specification Group (ISG) pour l'informatique en périphérie mobile (Hu *et al.*, 2015).

Dans le cadre de cette spécification, l'informatique en périphérie mobile vise à fournir un environnement de services informatiques et des capacités d'informatique en nuage à la périphérie du réseau (Yousefpour *et al.*, 2019). Il faut par conséquent déployer des serveurs de virtualisation à plusieurs emplacements à la périphérie du réseau mobile. Certains emplacements de déploiement envisagés par l'ISG sont les stations de base LTE/5G (eNodeB ou gNodeB), les contrôleurs de réseau radio 3G (RNC) ou les technologies d'accès radio multiples (3G/LTE/WLAN), qui peuvent

2. https ://www.etsi.org/

être situés à l'intérieur ou à l'extérieur de la périphérie du réseau mobile (Roman *et al.*, 2018).

L'informatique mobile de périphérie désigne un nouveau paradigme de réseau fournissant des services de technologie de l'information (TI), et les capacités de l'informatique en nuage au sein des réseaux d'accès mobiles des utilisateurs (Peng *et al.*, 2018). Une proposition de normalisation de l'informatique en périphérie des réseaux mobiles recommande une nouvelle structure organisationnelle et un nouvel écosystème, qui peuvent utiliser la technologie d'informatique mobile de périphérie pour réduire les tâches intensives impliquant des calculs. L'intention est de faire migrer ces tâches appartenant aux utilisateurs mobiles vers des serveurs de périphérie situés à proximité (Muniswamaiah *et al.*, 2021).

Étant donné que l'informatique de périphérie mobile est située dans le réseau d'accès radio, et à proximité immédiate des utilisateurs mobiles, il peut offrir des bandes passantes plus larges, et assurer une faible latence, pour améliorer principalement la qualité de service et la qualité d'expérience des utilisateurs finaux. L'informatique de périphérie mobile est également une technologie essentielle pour le développement de la 5G, qui permet d'atteindre les exigences élevées de la 5G en termes de flexibilité, de délai et d'évolutivité (Peng *et al.*, 2018).

La structure fonctionnelle de l'informatique de périphérie mobile comprend quatre couches fonctionnelles, soient (i) les dispositifs d'extrémité, soient (ii) le réseau d'accès, (iii) le réseau de périphérie et (vi) l'infrastructure centrale, comme le montre la figure 2.2. La première couche comprend les dispositifs connectés au réseau d'accès, par exemple les dispositifs IoT, les caméras IP et les terminaux mobiles. La deuxième couche, c.-à-d. le réseau d'accès sert de connexion entre les couches fonctionnelles et l'internet. Le réseau de périphérie , qui constitue la troisième couche, combine les concepts de virtualisation des fonctions de réseau et

de l'informatique de périphérie mobile. L'informatique de périphérie mobile peut être déployé par l'intermédiaire de plusieurs réseaux périphériques qui coopèrent en permanence et restent connectés au nuage traditionnel (Ali *et al.*, 2021).

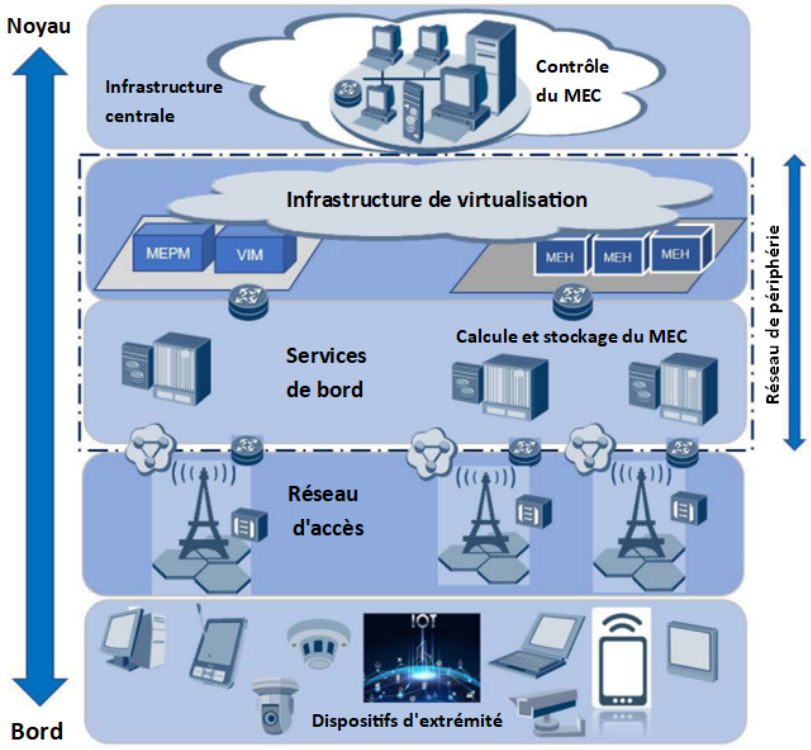

FIGURE 2.2 – Structure fonctionnelle de l'informatique de périphérie mobile (Ali *et al.*, 2021).

— Autres paradigmes informatiques similaires

Les expressions d'informatique en nuage mobile et d'informatique en brouillard sont utilisées de manière interchangeable dans certains articles, car elles ont toutes le terme « bord (en anglais edge)» en commun. Le terme « bord » utilisé par l'industrie des télécommunications fait généralement référence aux stations de base 4G/5G, aux RAN et aux ISP (Internet Service Provider) des réseaux d'accès de la périphérie (Yousefpour *et al.*, 2019). Il est important de noter qu'il existe plusieurs

autres paradigmes de périphérie qui jouent le même rôle, à savoir rapprocher les services du réseau aux utilisateurs, mais qui sont peu évoqués. Dans cette section, nous en mentionnons quelques-uns.

Les cloudlets qui sont parfois appelés micro-centres de données dans certaines études ont été proposés par Microsoft Research en 2015 et sont définis comme une extension des centres de données traditionnels utilisés dans l'informatique en nuage. Un micro-centre de données peut être un nœud de périphérie (ou un cloudlet) qui est déployé entre les dispositifs sans fil et le nuage (Yousefpour *et al.*, 2019).

Un autre paradigme de la périphérie qui est l'informatique mobile en nuage (en anglais mobile cloud computing) se concentre principalement sur la notion de « délégation mobile». En effet, en raison des ressources limitées dont disposent les dispositifs mobiles, ils doivent déléguer le stockage de données en masse et l'exécution de tâches à forte intensité de calcul à des entités distantes (Roman *et al.*, 2018). Lorsque nous parlons d'informatique en nuage, nous parlons principalement des nuages «centraux» ou «distants», qui sont éloignés de l'utilisateur ou des appareils. Les nuages centraux sont plus éloignés des objets connectés et sont responsables des calculs lourds. En revanche, les nuages « périphériques» sont à plus petite échelle que les nuages centraux et sont plus proches des appareils (Yousefpour *et al.*, 2019). Le concept de nuage de périphérie (en anglais edge cloud) est similaire à l'informatique de périphérie, il étend les capacités du nuage en exploitant les nœuds de calcul fournis par l'utilisateur ou l'opérateur à la périphérie du réseau. Comme pour l'informatique en brouillard, les nuages de périphérie permettent d'exécuter une application de manière coordonnée à la fois en périphérie et dans le nuage distant (Chang *et al.*, 2014).

2.1.3 Comparaison des paradigmes

Bien que l'informatique en brouillard et l'informatique de périphérie déplacent toutes les deux le calcul et le stockage à la périphérie du réseau et plus près des nœuds finaux, ces paradigmes ne sont pas identiques. En fait, le Consortium OpenFog déclare que l'informatique de périphérie est souvent appelée incorrectement l'informatique en brouillard ; le Consortium OpenFog fait la distinction suivante : l'informatique en brouillard est hiérarchique et fournit le calcul, la mise en réseau, le stockage, le contrôle et l'accès n'importe où dans le continuum entre nuages et les objets ; tandis que l'informatique de périphérie tend à se limiter à la périphérie (Yousefpour *et al.*, 2019). De plus, l'informatique en brouillard se concentre sur des plateformes horizontales, qui renforcent les fonctions communes vers des domaines d'application et des industries multiples (Muniswamaiah *et al.*, 2021).

De toute évidence, même si tous ces paradigmes ont le même objectif, ils auront des différences implicites dans la façon de les atteindre. Par exemple, l'informatique de périphérie mobile limite le déploiement des plateformes de calcul en périphérie aux infrastructures de réseaux mobiles telles que la 5G. D'autre part, les nœuds de l'informatique de brouillard peuvent également être déployés à d'autres emplacements, tels que des serveurs gérés par l'utilisateur, des points d'accès, des routeurs, passerelles, etc. (Roman *et al.*, 2018). Comparée à l'informatique en périphérie mobile, l'informatique de brouillard englobe le nuage, le réseau d'infrastructure (en anglais core network), le réseau métro politain, la périphérie, les clients et les objets (figure 2.3).

L'informatique de brouillard cherche à réaliser un enchainement continu de services informatiques, du nuage aux objets, plutôt que de traiter les bords du réseau comme des plateformes informatiques isolées telles que l'informatique mobile (en anglais mobile computing) et l'informatique en nuage ad hoc mobile (mobile ad

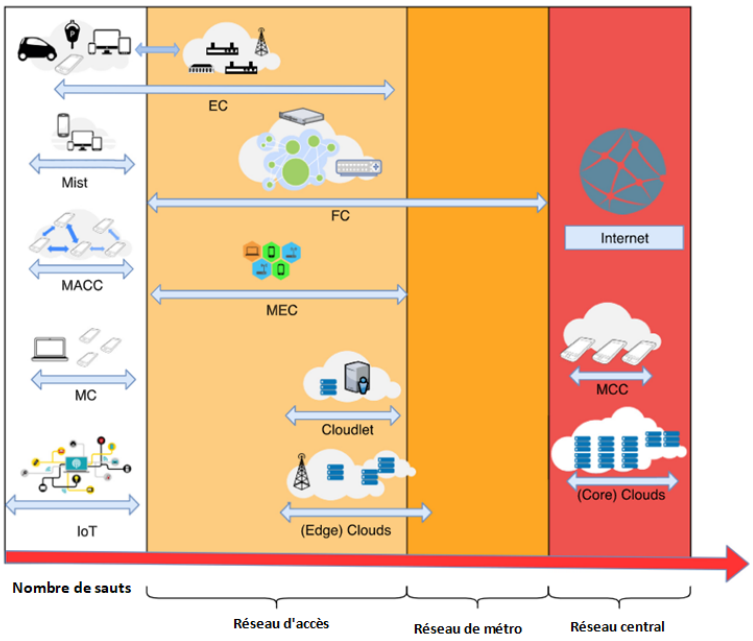

FIGURE 2.3 – Comparaison entre les paradigmes informatiques (Akyildiz *et al.*, 2008).

hoc cloud computing) (Roman *et al.*, 2018). L'informatique en nuage ad hoc mobile diffère également de l'informatique en nuage mobile (en anglais mobile cloud computing) en ce qui concerne le matériel, la méthode d'accès au service et la distance entre les utilisateurs, puisque le calcul est effectué sur des dispositifs mobiles dans le cas de l'informatique en nuage ad hoc mobile, alors qu'il est éloigné des dispositifs mobiles dans le cas de l'informatique en nuage mobile (Yousefpour *et al.*, 2019).

2.1.4 Conclusion

Nous avons exposé la vision et défini les principales caractéristiques du réseau d'informatique de périphérie, une plateforme permettant de fournir une riche gamme de nouveaux services et applications. Grâce à tous les paradigmes développés dans

ce type de réseau, il est devenu plus facile pour les utilisateurs d'utiliser des applications très gourmandes en termes de calcul et de mémoire. Des avantages tels que le calcul et le stockage à proximité des utilisateurs sont des caractéristiques partagées par les différents paradigmes informatiques. Bien que certains d'entre eux soient plus puissants et plus avantageux que d'autres, le traitement des demandes des utilisateurs à proximité de ces derniers reste un grand point fort.

2.2 Partage dynamique du spectre

2.2.1 Introduction

Le spectre est une ressource naturelle précieuse et rare. Les organismes de règlementation attribuent les droits d'utilisation du spectre selon les applications, principalement en délivrant des licences. Avec l'introduction progressive de la libération du spectre, les problèmes de propriété et d'attribution des licences d'utilisation du spectre ont attiré une attention croissante (You *et al.*, 2010). Cette demande accrue du spectre est la cause principale de sa rareté, qui est plus importante au niveau de certaines bandes de fréquence. Toutefois, une grande partie du spectre attribué est utilisée de manière cyclique, ce qui entraine une sous-utilisation d'une partie importante de ce spectre. Pour améliorer l'efficacité du spectre, des techniques avancées de partage sont généralement utilisées.

Parmi les techniques avancées de partage du spectre, on retrouve la radio cognitive, la communication entre appareils (en anglais device-to-device ou D2D), la communication en duplex intégral (en anglais full duplex), l'accès multiple non orthogonal (en anglais non-orthogonal multiple access ou NOMA) (Zhang *et al.*, 2017). Les utilisateurs mobiles de la 5G peuvent être multi-modes, c'est-à-dire, prendre en charge plusieurs techniques de partage du spectre comme montre la figure 2.4. Le choix du mode de partage va être fait en se basant sur les besoins

des utilisateurs.

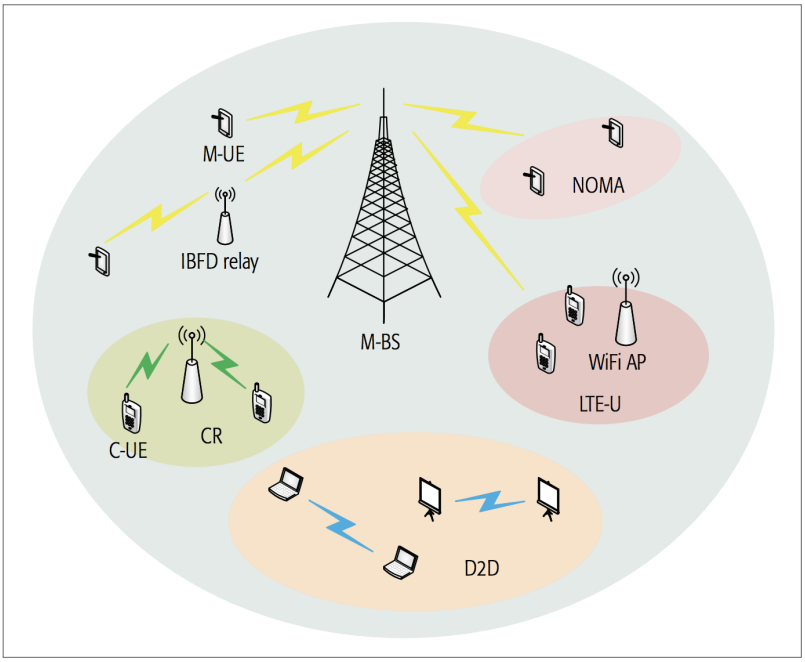

FIGURE 2.4 – Plusieurs techniques avancées de partage du spectre coexistent dans un réseau 5G (Zhang *et al.*, 2017).

Dans le cadre de ce projet de maîtrise, nous considérons la technologie de la radio cognitive (RC) qui offre la possibilité de partager le canal sans fil avec des utilisateurs détenant la licence d'utilisation (appelés primaires) d'une manière opportuniste. Ces dispositifs à radio cognitive recueillent des informations sur les ressources spectrales disponibles et sélectionnent les ressources les plus appropriées tout en évitant les interférences avec les dispositifs sans fil primaires (Hasegawa *et al.*, 2014). Formellement, une radio cognitive est définie comme une radio qui peut modifier les paramètres de son émetteur en fonction de l'interaction avec son environnement.

2.2.2 Architecture du réseau à radio cognitive

L'architecture du réseau à radio cognitive détermine la façon avec laquelle le spectre sous licence est partagé entre les utilisateurs secondaires (non-porteurs de licence). Il existe principalement deux types d'architectures d'accès, à savoir centralisée et distribuée. Dans une architecture centralisée, le contrôle de l'attribution et de l'accès au spectre par les utilisateurs secondaires est effectué par un point central, par exemple une station de base (Pandit, 2017). Le rôle du contrôleur central est de collecter des informations sur l'utilisation du spectre des utilisateurs disposant de la licence (alias primaires) ainsi que des informations sur les besoins en spectre des utilisateurs secondaires. L'inconvénient de cette architecture est que les communications entre le contrôleur et les utilisateurs secondaires entrainent une surcharge, parfois importante, du réseau (Qin *et al.*, 2020).

L'architecture distribuée des réseaux à radio cognitive permet une communication directe entre les utilisateurs secondaires sans avoir besoin d'une station de base ou d'un contrôleur central. Par conséquent, l'utilisateur secondaire peut prendre une décision sur l'accès au spectre de manière indépendante et de façon autonome. Comme chaque utilisateur secondaire doit collecter les informations sur l'environnement radio ambiant et prendre sa décision localement, l'émetteur-récepteur à radio cognitive de chaque utilisateur secondaire nécessite plus de ressources de calcul que celles requises dans le réseau centralisé (Pandit, 2017).

2.2.3 Structure de la gestion du spectre

Afin de répondre aux exigences critiques du réseau mobile de cinquième génération (5G), en particulier la couverture élargie, la capacité massive, la connectivité massive et la faible latence utilisée dans la 5G, le réseau sera étendu à la zone du spectre complet éventuellement de 1 GHz à 100 GHz (Hu *et al.*, 2015). Se-

lon la bande de fréquence utilisée par les utilisateurs secondaires, on peut définir deux catégories de partage spectrales. Le premier partage du spectre est celui sans licence qui permet aux utilisateurs d'accéder au spectre en utilisant ces parties libres, ce qui permet de minimiser les interférences avec les utilisateurs primaires et le second est le partage du spectre sous licence (Sharmila et Dananjayan, 2019).

Pour autoriser les utilisateurs secondaires à accéder au spectre partagé avec les utilisateurs primaires, diverses techniques peuvent être employées dans un réseau à radio cognitive. Lorsque l'utilisateur secondaire accède à des bandes de fréquence qui ne sont pas occupées par l'utilisateur primaire, on parle du partage par entrecroiser (en anglais interweave). Si un utilisateur secondaire et un utilisateur primaire partagent la même bande de fréquence à condition que les transmissions primaires ne soient pas dérangées, on dit qu'il s'agit du partage du spectre par sous-couche (en anglais underlay). La combinaison entre les deux techniques précitées confère plus d'avantages aux utilisateurs secondaires, puisque chaque utilisateur peut soit combler une bande de fréquence libre ou partager avec un utilisateur primaire ; cette technique est appelée le partage hybride. Dans toutes ces techniques, les utilisateurs secondaires sont obligés d'évacuer la bande dès que l'utilisateur primaire reprend sa transmission ou qu'il est dérangé (Sharmila et Dananjayan, 2019).

La disponibilité du spectre varie selon les cas, ce qui entraine une concurrence entre les utilisateurs secondaires. Il est donc nécessaire d'utiliser les morceaux du spectre disponibles (appelés aussi trous spectraux ou spectrum holes en anglais) de manière équitable et efficace. La clé de l'attribution du spectre est de concevoir des algorithmes et des règles d'attribution du spectre, qui peuvent améliorer l'efficacité de l'utilisation du spectre dans le cas d'une minimisation des conflits ou d'une absence de conflits (Hu *et al.*, 2015). Dans un scénario où un utilisateur primaire arrive dans une bande de fréquence occupée par un utilisateur secondaire,

ce dernier devrait idéalement passer à une autre bande de fréquence. Ce processus, souvent appelé transfert (en anglais handover ou handoff) a des configurations distinctes qui concernent son intégration avec la détection du spectre et l'arrivée de l'utilisateur primaire. Parmi ces configurations, on trouve le non-transfer (en anglais non-handoff), qui consiste que l'utilisateur secondaire reste inactif jusqu'à ce que l'utilisateur primaire quitte le canal, puis reprend la transmission de données (Santana *et al.*, 2018).

2.2.4 Conclusion

Les réseaux à radio cognitive sont développés pour résoudre les problèmes actuels des réseaux sans fil provoqués par la limitation du spectre disponible et de l'inefficacité de l'utilisation du spectre. L'un des points forts des réseaux à radio cognitive est d'être attentifs et sensibles aux changements de leur environnement, ce qui facilite la gestion du spectre. Avec sa structure de gestion du spectre, le réseau à radio cognitif diminue l'interférence entre les utilisateurs et garantit une bonne qualité de service pour les utilisateurs prioritaires dans le réseau.

CHAPITRE III

ÉTAT DE L'ART

3.1 Introduction

Le présent chapitre présente un survol des travaux connexes qui ont pour thème la même problématique étudiée dans le cadre de ce projet. Plus précisément, nous allons discuter des articles de recherche qui traitent les trois enjeux suivants : le déchargement des tâches vers l'informatique de périphérie, le partage dynamique du spectre entre les utilisateurs primaires et secondaires et le problème d'ordonnancement au niveau des serveurs de l'informatique de périphérie.

3.2 Déchargement des tâches vers le réseau d'informatique de périphérie

Durant les dernières années, le déchargement des calculs est devenu un moyen efficace pour surmonter les contraintes liées aux ressources des appareils mobiles en déchargeant les tâches d'applications mobiles sensibles aux délais et gourmands en calcul vers des serveurs de l'informatique de périphérie. Cette technique permet ainsi de prolonger la durée de vie des batteries, de réduire la latence et d'améliorer les performances des applications. Cependant, le déchargement des tâches de calcul est affecté par de nombreux facteurs, ce qui rend difficile l'atteinte des objectifs prévus.

L'article (Feng *et al.*, 2019) étudie le déchargement multi-utilisateurs et multi-tâches basé sur le nuage de bord (en anglais edge cloud) non-équilibré (en anglais umbalanced). Les auteurs présentent un nouveau critère pour représenter le coût de déchargement, basé sur un compromis entre le délai, la consommation d'énergie et le coût, conçu pour quantifier l'expérience de l'utilisateur en matière de déchargement de tâches et pour être la cible d'optimisation de la décision de déchargement. L'article s'attaque à deux problèmes d'optimisation, à savoir la minimisation de la somme des coûts de déchargement pour tous les utilisateurs (basé sur l'efficacité) et la minimisation du coût de déchargement maximal par utilisateur (basé sur l'équité). Les utilisateurs avec un coût de déchargement élevé ne déchargent pas leurs tâches, ce qui va minimiser le nombre global des utilisateurs.

Dans le déchargement des tâches, la distance entre un utilisateur et un serveur affecte considérablement la latence causée par l'opération de déchargement. Il est donc crucial de déterminer (1) où placer les serveurs, (2) combien de serveurs installer et (3) quel serveur sélectionner. L'article (Song *et al.*, 2022) optimise conjointement les décisions de déchargement et le déploiement des serveurs de calcul, minimiser le délai de traitement moyen dans le réseau périphérique sans fil tout en satisfaisant les exigences de délai individuels des utilisateurs. Les auteurs présentent une solution basée sur l'algorithme génétique classique, dans le but d'optimiser le délai de traitement des utilisateurs.

Dans l'article (Alfakih *et al.*, 2020), les dispositifs mobiles traitent leurs tâches localement ou à distance. Il existe trois options de déchargement : le serveur de périphérie le plus proche, le serveur de périphérie adjacent et le nuage distant. Les auteurs proposent un algorithme basé sur l'apprentissage par renforcement pour résoudre le problème de la gestion des ressources dans le serveur de périphérie, et prendre la décision de déchargement optimale pour minimiser un coût formulé en fonction de la consommation d'énergie et du délai de calcul. Relativement à

notre sujet de recherche, la priorité des utilisateurs est une contrainte qui reste à étudier dans cet article.

De même, dans l'article (Nath et Wu, 2020), les utilisateurs mobiles exécutent leurs tâches soit localement soit à distance dans un ou plusieurs serveurs périphériques mobiles. Les auteurs visent à minimiser le coût moyen à long terme du système, en optimisant la consommation d'énergie, le délai et le coût de récupération du cache ; et en tenant compte des contraintes des ressources limitées de stockage et de calcul au niveau du serveur périphérique mobile. Malgré cela, l'approche ne garantit pas que chaque tâche déchargée sera traitée avant son délai d'échéance.

L'article (Jošilo et Dán, 2021) analyse les décisions relatives au déchargement des calculs des dispositifs dans un système informatique mobile en périphérie où les ressources sans fil et informatiques sont gérées conjointement par un opérateur. Chaque dispositif prend la décision de décharger la tâche ou de la traiter localement, afin de minimiser son propre délai de traitement. En utilisant la théorie des jeux, les auteurs développent une solution illustrant l'interaction entre les dispositifs et l'opérateur en vue de minimiser les coûts et de répartir les ressources de l'opérateur. Avec l'augmentation du nombre de dispositifs, la solution offre constamment un délai de traitement optimal, mais ceci sans prendre en compte les interférences entre les dispositifs.

Dans un contexte réaliste, plusieurs sous-réseaux autonomes de l'informatique de périphérie peuvent co-exister dans des zones adjacentes, ce qui entraine la possibilité d'un déséquilibre de la charge des serveurs entre les sous-réseaux autonomes pendant la période de pointe de la demande des tâches. Pour relever ces défis, et en tenant compte des tâches multiples, de l'hétérogénéité du sous-réseau périphérique et de la mobilité des dispositifs périphériques, l'article (Li *et al.*, 2020) propose un algorithme d'apprentissage par renforcement profond qui peut connaitre l'en-

vironnement du réseau et générer la décision de déchargement des calculs pour minimiser le délai de la tâche.

Dans les réseaux informatiques de périphérie, il est souvent difficile de faire le choix entre le déchargement des calculs à la périphérie ou dans un nuage distant. Chaque utilisateur veut que sa tâche soit traitée dans les plus brefs délais. Par conséquent, même si l'informatique de périphérie peut offrir un service de proximité qui peut garantir un délai de traitement réduit, il y a toujours des cas où il est préférable que la tâche soit traitée dans le nuage distant grâce à ses ressources quasi-illimitées. À cet effet, l'article (Xu *et al.*, 2019) offre aux utilisateurs le choix de décharger leurs tâches au niveau du nuage ou des nœuds périphériques. Les auteurs proposent une méthode de déchargement heuristique afin de minimiser le délai total du traitement.

Afin d'améliorer l'efficacité du traitement des tâches à la périphérie du réseau tout en respectant la limitation des ressources de calcul et de communication, l'article (Kai *et al.*, 2021) développe un système de calcul collaboratif dans lequel les tâches des dispositifs mobiles peuvent être partiellement traitées au niveau des terminaux, des nœuds périphériques et du centre de données dans le nuage. L'approche permet de minimiser la latence de tous les dispositifs mobiles en tenant compte de la stratégie de déchargement, des ressources de calcul, et de l'allocation d'énergie. Le partage des bandes de fréquence entre les dispositifs mobiles est un facteur qui a un impact sur la minimisation de la latence, mais qui n'est pas traité dans cet article.

De même, l'article (Bi *et al.*, 2021) propose une méthode de déchargement partiel des calculs pour minimiser l'énergie totale consommée par les dispositifs mobiles intelligents et les serveurs périphériques, et cela en optimisant conjointement le taux de déchargement des tâches, les vitesses des processeurs, et la puissance de

transmission des dispositifs mobiles intelligents, et la bande passante des canaux disponibles. Un planificateur de calcul périphérique divise les données de tous les dispositifs mobiles intelligents en plusieurs parties indépendantes qui peuvent être exécutées en parallèle, ce qui permet de minimiser la consommation d'énergie. Cette approche est performante uniquement dans le cas où le partage du spectre entre les dispositifs mobiles intelligents est inexistant.

3.3 Ordonnancement des tâches au niveau des serveurs périphérique

Comme nous l'avons mentionné, le fait de décharger des tâches sur des dispositifs situés à la périphérie du réseau permet de réduire les délais de service et la consommation en bande passante. Néanmoins, ces dispositifs ne sont pas forcément dotés de ressources suffisantes pour traiter toutes les demandes des utilisateurs connectés au réseau. L'ordonnancement des tâches au niveau des serveurs de la périphérie est une solution qui permet de considérer les utilisateurs selon leur priorité, utiliser efficacement les ressources au niveau des serveurs qui disposent de ressources limitées et aussi réduire le temps d'exécution des tâches.

L'ordonnancement au niveau des systèmes sans fil était étudié depuis plus de vingt ans, mais n'a pas été mis en œuvre dans des systèmes réels en raison de sa grande complexité et de ces exigences difficilement réalisables (Asadi et Mancuso, 2013). De nombreux algorithmes heuristiques d'ordonnancement ont été proposés dans divers articles de recherche, mais des améliorations sont nécessaires pour rendre le système plus rapide et plus réactif. Les algorithmes d'ordonnancement traditionnels tels que premier arrivé, premier servi (en anglais First Come First Serve ; FCFS), plus courte tâche en premier (en anglais Shortest Job First ; SJF), Min-Min et Max-Min ne représentent pas les meilleures solutions aux problèmes d'ordonnancement (Wadhonkar et Theng, 2016) au vu de l'évolution actuelle des

réseaux sans fil.

Dans l'informatique en nuage, plusieurs algorithmes d'ordonnancement ont été mis en place dans le but d'améliorer les performances offertes aux utilisateurs. L'article (Jian *et al.*, 2019) présente une approche qui réduit le délai de traitement et la charge non uniforme causés par un ordonnancement inadéquat. La solution proposée permet d'améliorer le compromis entre la complexité algorithmique et les performances. Cependant, si le délai d'échéance de la tâche est extrêmement limité, la solution ne peut pas garantir un délai optimal aux utilisateurs.

L'affectation des tâches aux serveurs de périphérie de manière à minimiser le délai d'exécution est un des défis principaux du mécanisme d'ordonnancement. Afin de minimiser le temps de réponse total pour traiter toutes les tâches des utilisateurs au niveau des serveurs périphériques, l'article (Fang *et al.*, 2020) considère l'ordonnancement des tâches avec différents instants de disponibilité, de temps de transmission des données et de retour des résultats et également des serveurs avec différentes vitesses de traitement. Néanmoins, le traitement non-préemptif des tâches au niveau du serveur a pour conséquence que le nombre de celles-ci ne peut être maximisé.

Pour améliorer l'efficacité d'exécution d'une application dans un serveur périphérique, l'ordonnancement des tâches et la conteneurisation doivent être considérés conjointement. À cette fin, une technique d'optimisation conjointe a été développée dans (Zhang *et al.*, 2021). Sans tenir compte de la conteneurisation, l'ordonnancement initial des tâches a eu lieu, pour ensuite faire correspondre les tâches aux conteneurs grâce à un algorithme spécialement conçu à cette fin. Cette méthodologie démontre qu'il est possible de réduire les opérations inefficaces des conteneurs et améliorer l'efficacité de l'exécution des tâches.

L'article (Kim *et al.*, 2020) propose un nouveau concept d'informatique de péri-

phérie qui utilise les ressources inactives consolidées des dispositifs IoT pour les services de périphérie en déchargeant les tâches de périphérie sur les appareils IoT à proximité. Étant donné que les dispositifs IoT modernes n'utilisent pas pleinement leurs ressources informatiques, il est possible que les tâches de périphérie soient déchargées sur des dispositifs IoT tant qu'elles n'interfèrent pas avec l'exécution normale des tâches IoT locales. Les résultats de cette initiative montrent que la technique proposée permet non seulement d'atteindre un débit de tâches quasi-optimal par rapport aux autres algorithmes d'ordonnancement en termes de taux de satisfaction des échéances des tâches critiques, mais aussi de maintenir les échéances des tâches locales dans les dispositifs IoT.

L'article (Sorkhoh *et al.*, 2019) fait état d'approches d'ordonnancement basées sur des techniques de relaxation lagrangienne afin de maximiser le nombre des tâches autorisées au sein du serveur périphérique. Le prétraitement et le filtrage des tâches constituent les premières étapes effectuées au début de chaque époque d'ordonnancement. Les auteurs développent un système qui permet l'ordonnancement de deux ensembles de tâches : celles qui sont déjà dans les files d'attente des serveurs (tâches pré-admises) et celles récemment arrivées. La solution converge vers celle optimale lorsque le nombre de serveurs périphériques augmente, car plus de ressources sont disponibles pour le traitement des tâches. Cependant, lorsqu'on ne dispose que d'un seul serveur périphérique, cette solution risque de ne plus être efficace pour le problème d'ordonnancement.

3.4 Partage dynamique du spectre

La thématique du partage du spectre se situe depuis plusieurs années au centre des préoccupations des recherches dans les domaines de communications et de réseaux sans fil. Plusieurs travaux se sont intéressés notamment au partage dynamique de

spectre dans les réseaux cellulaires, les réseaux véhiculaires, l'internet des objets et bien plus encore. Dans la présente section, nous nous concentrons davantage sur les recherches les plus récentes et également sur celles qui abordent la problématique du partage du spectre dans l'informatique de périphérie. Avec l'augmentation de la charge de trafic, la rareté des ressources spectrales pose un sérieux problème de gestion efficace du spectre.

Dans les réseaux à radio cognitive, les utilisateurs secondaires sont censés opérer dans le spectre des utilisateurs primaires sans leur causer d'interférence. L'article (Homayouni *et al.*, 2011) étudie une technique de partage opportuniste pour un cas particulier dans lequel deux utilisateurs secondaires sont autorisés à co-exister dans la même sous-bande. Quand les utilisateurs primaires et secondaires occupent tous les bandes de fréquence disponibles, l'approche consiste à diviser les canaux des utilisateurs secondaires en deux sous-bandes, et deux utilisateurs secondaires peuvent utiliser une sous-bande simultanément. Les auteurs montrent que la proposition réduit considérablement la probabilité de blocage et d'abandon des utilisateurs secondaires.

Le partage du spectre entre différentes communications entraine généralement une forte dégradation des performances, comme dans l'article (Zhai *et al.*, 2014), où le partage se fait entre les communications cellulaires et les communications ad-hoc. Pour cette raison, une proposition qui permet aux utilisateurs secondaires d'utiliser le spectre sans avoir un impact sur la communication des utilisateurs primaires a été développée. L'objectif est de libérer une fraction de la bande passante pour permettre la transmission des données des utilisateurs secondaires.

Pour allouer les ressources de communication et de calcul aux utilisateurs, plusieurs solutions ont été étudiées dans divers articles, le défi étant toujours de partager le spectre entre ces utilisateurs de manière à répondre à leurs exigences.

L'article (Zeng et Fodor, 2020) propose un partage dynamique du spectre pour relever ce défi. Cette approche équilibre la charge de communication et de calcul entre les stations de base, ce qui permet de minimiser la consommation d'énergie. Ce travail considère que chaque utilisateur doit traiter sa tâche avant une d'échéance stricte. Il faut noter que comme le serveur périphérique traite une tâche à la fois, un délai d'attente doit être ajouté au temps de traitement global de la tâche. Toutefois, cet aspect précis n'a pas été traité dans cet article.

De même, l'article (Biswas *et al.*, 2021) étudie le partage du spectre entre les réseaux cellulaires et ceux IoT, afin de minimiser l'énergie globale de transmission. Les auteurs proposent une méthode d'optimisation conjointe centralisée, pour un équilibre de partage entre les deux réseaux. Les réseaux primaires (cellulaires) et secondaires (IoT) profitent des avantages de la coopération, ce qui permet de garantir une bonne qualité de service ainsi qu'une optimisation d'énergie pour les dispositifs IoT. Cette solution est plus efficace que l'allocation égale de ressources au serveur périphérique.

En outre, l'article (Liu *et al.*, 2016) propose une technique de partage partiel du spectre utilisant la technologie de la radio cognitive. La division des ressources radio en une partie partagée et une partie non partagée permet non seulement de résoudre le problème des interférences à deux niveaux, mais aussi d'améliorer la qualité de service. Cette approche se concentre sur la minimisation de la puissance d'interférences, mais elle ne prend pas en considération les autres contraintes pour optimiser les délais de traitement des utilisateurs tel que les échéances des tâches.

3.5 Exemple d'applications

À travers la littérature, de nombreux exemples d'application ont été étudiés concernant les mêmes problèmes que ceux de notre projet de thèse. Dans cette section,

nous discuterons davantage les réseaux véhiculaires puisque dans ce type de réseau les applications sont très variées.

3.5.1 Réseaux véhiculaires

Les véhicules d'aujourd'hui exigent des communications qui nécessitent une bonne connectivité et une transmission de données à très faible latence (Shah *et al.*, 2018). Afin d'atteindre ces objectifs, plusieurs recherches ont été menées pour proposer des solutions permettant de relever ces défis. La 5G promet d'augmenter la performance des services, surtout avec des réseaux d'accès qui traitent les tâches de calcul des utilisateurs le plus proche d'eux. Le déchargement des tâches véhiculaires sur les serveurs périphériques est une solution qui a fait l'objet de plusieurs travaux vu qu'elle permet d'offrir un temps de traitement optimal pour les demandes des utilisateurs.

L'article (Aissioui *et al.*, 2018) propose une approche qui assure un délai de bout en bout ultra-faible entre l'utilisateur véhiculaire final et les services hébergés. Cette solution considère la combinaison de deux concepts, l'informatique de périphérie mobile et la convergence fixe-mobile (en anglais Follow Me Cloud ; FMC) qui permet une gestion transparente de la mobilité des utilisateurs. L'intégration de ces deux concepts dans les communications entre véhicules et infrastructures (V2I) permet de répondre aux exigences de qualité de service de la conduite autonome dans le réseau 5G.

Dans l'article (Zhao *et al.*, 2019), les auteurs présentent une approche qui utilise le serveur MEC en parallèle avec l'informatique en nuage pour traiter les tâches déchargées des utilisateurs véhiculaires. Cette approche consiste à optimiser conjointement la décision de déchargement des calculs et l'allocation des ressources au niveau des serveurs. La solution permet de réduire considérable-

ment la complexité du système sans perte de performance, grâce à la gestion des ressources informatiques des serveurs MEC.

Afin de décharger les tâches informatiques des utilisateurs sur le serveur MEC, il est essentiel d'assurer une bonne performance pour la transmission des données entre l'utilisateur et la station de base liée au serveur. Les interférences entre les utilisateurs font partie des problèmes que l'on peut rencontrer dans ce cas. L'article (Wang *et al.*, 2017) propose une approche d'allocation des ressources pour la transmission des données des utilisateurs afin de diminuer cette interférence. L'utilisation du mécanisme de gestion des ressources dans le serveur MEC permit d'augmenter les performances du système. Cette combinaison permet de diminuer le délai de traitement total des tâches des utilisateurs ainsi que la puissance d'interférence.

Un utilisateur véhiculaire a le choix entre le déchargement de ces tâches vers un serveur MEC, vers un serveur du nuage distant ou bien de les traiter localement. Un choix adéquat du mode d'exécution (c.-à-d. déchargement ou traitement local) permet d'optimiser le délai de traitement. La proposition dans (Zhang *et al.*, 2021) cherche à déterminer ce type de choix. La minimisation du délai de traitement s'effectue par la comparaison du délai de traitement de la tâche selon les modes d'exécution, peu importe le mode de communication considéré (V2V ou V2I). Cette solution permet d'une part l'équilibrage de la charge au niveau des serveurs MEC, et d'autre part la minimisation du délai total de traitement des tâches des utilisateurs.

Comme nous l'avons vu dans les sections précédentes, le partage du spectre reste un domaine de recherche important, quel que soit le type de réseau ou d'utilisateur en question. Dans le réseau véhiculaire, le partage du spectre entre les communications V2V, V2I ou V2X demeure un enjeu qui a un effet direct sur le temps de

traitement des tâches des utilisateurs et impacte ainsi les performances globales du système. La réutilisation du canal de l'utilisateur primaire est une approche qui a été étudiée différemment dans divers articles.

L'article (Qian *et al.*, 2021) étudie le partage du spectre entre les utilisateurs cellulaires (utilisateurs primaires) et les utilisateurs véhiculaires. Les utilisateurs véhiculaires peuvent décharger leurs tâches sur les serveurs périphériques en réutilisant le canal des utilisateurs cellulaires. Une optimisation conjointe du déchargement partiel des utilisateurs véhiculaires et de l'allocation des ressources informatiques a été développée afin de minimiser le retard des utilisateurs véhiculaires tout en limitant l'augmentation du retard des utilisateurs cellulaires. Les auteurs utilisent l'accès multiple non orthogonal (NOMA) pour former des groupes d'utilisateurs véhiculaires qui déchargent leurs calculs à la périphérie.

L'article (Liang *et al.*, 2019) propose une solution pour optimiser le délai de traitement et l'allocation des ressources conjointement des serveurs MEC et l'informatique en nuage. Cette approche permet de traiter les demandes des utilisateurs qui nécessitent beaucoup de ressources, grâce à la réutilisation du spectre. La solution proposée se base sur l'apprentissage par renforcement multi-agents et permet de garantir une bonne communication véhicule à infrastructure et en même temps d'augmenter le nombre des communications véhicule à véhicule qui vont utiliser le spectre pour décharger leurs calculs.

3.5.2 Les véhicules aériens sans pilote

Les véhicules aériens sans pilote (connus communément sous le nom de drones) ont été initialement développés pour des tâches militaires de contrôle et de surveillance, mais ils ont trouvé plusieurs applications intéressantes dans le domaine civil tel que les stations de base cellulaires aériennes qui permettent d'étendre la

couverture des communications sans fil à la demande (Koulali *et al.*, 2016). Ces applications fonctionnant dans des bandes sans licence connaissent une croissance considérable avec la consolidation de l'IoT. Cependant, ces bandes sont devenues surchargées, car les systèmes qui les utilisent sont en constante augmentation (Santana *et al.*, 2018). L'utilisation de la technologie de la radio cognitive et le partage dynamique du spectre ont permis de faire face à ce défi.

L'utilisation des véhicules aériens sans pilote pour assurer la couverture des dispositifs IoT lors du déchargement des tâches vers la périphérie du réseau a permis de résoudre les problèmes d'interférences entre les utilisateurs et aussi d'optimiser le problème de l'allocation des ressources. L'article (Al-Hourani, 2020) présente une étude du comportement d'interférence observé (occupation du spectre) à partir d'un drone à basse altitude qui permet de détecter l'état du spectre, ainsi que le taux d'interférence lors du partage de spectre entre les dispositifs IoT. Cette technique est plus avantageuse que les analyseurs de spectre classiques en raison de son taux d'échantillonnage élevé.

Une autre façon permettant de se servir des véhicules aériens sans pilote a été étudiée dans l'article (Nguyen *et al.*, 2021). Pour faire face aux effets de la destruction du réseau lors d'une catastrophe naturelle, une proposition d'optimisation en temps réel pour l'allocation des ressources (par exemple, la puissance et le nombre de drones) pour les réseaux à radio cognitive a été développée. Le but de cette étude est que les véhicules aériens sans pilote peuvent servir comme des stations de base volante pour fournir une couverture réseau étendue à la zone touchée. La technique d'optimisation assistée par l'apprentissage pour l'allocation optimale des ressources radio des réseaux considérés sous la contrainte stricte de l'interférence maximale tolérable était la solution qui a permis de desservir les utilisateurs secondaires rapidement, et aussi d'optimiser le débit des réseaux primaires.

Les véhicules aériens sans pilote peuvent aussi servir comme nœuds relais pour le déchargement des données des utilisateurs. L'article (Dai *et al.*, 2022), présente une étude du déchargement de données assisté par un véhicule aérien sans pilote pour les conteneurs intelligents dans les communications maritimes. En effet, chaque drone est considéré comme un intermédiaire entre les conteneurs intelligents et la station de base terrestre. Les auteurs proposent un algorithme de déchargement des données pour réduire le délai moyen de déchargement en tenant compte de la taille des données et les contraintes d'énergie disponible des conteneurs intelligents.

3.6 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art en relation avec la problématique étudiée dans le cadre de ce mémoire qui est l'optimisation du délai de traitement des tâches déchargées vers le serveur périphérique et l'allocation des ressources. Cet état de l'art présente des propositions et solutions qui s'attaquent à de réduire tous ces défis dans les réseaux sans fil, et ce indépendamment du type des applications. Au mieux de notre investigation de la littérature, nous n'avons pas trouvé de travaux de recherche qui étudient la même fonction objectif (soit la maximisation du nombre de tâches déchargées) dans un contexte de partage dynamique du spectre. Dans notre travail, nous proposons une approche qui maximise le nombre de tâches déchargées vers le serveur de périphérie, tout en optimisant le délai de traitement pour chaque utilisateur individuellement et l'allocation des ressources au niveau du serveur périphérique. Dans le chapitre suivant, nous présentons les détails de notre contribution ainsi que sa validation par des résultats de simulation.

CHAPITRE IV

ORDONNANCEMENT DES TÂCHES ET ALLOCATION DES RESSOURCES DANS L'INFORMATIQUE DE PÉRIPHÉRIE

Ce chapitre présente la modélisation du système étudié. Il commence par la définition de l'architecture du système. Ensuite, il formule mathématiquement le problème de maximisation du nombre d'utilisateurs secondaires qui déchargent leurs tâches vers le serveur périphérique. Ce chapitre présente par la suite plusieurs solutions algorithmiques pour résoudre le problème formulé. Finalement, il présente une évaluation des performances des solutions proposées en les comparant à des banchmarks dans un algorithme de force brute qui permet de retrouver la solution optimale du problème.

4.1 Modèle du système

4.1.1 Modèle du réseau

Nous considérons un système MEC multi-utilisateurs comprenant une station de base (BS) ayant accès à un serveur MEC. La station de base dispose d'un ensemble de bandes de fréquence désignées par $\mathcal{N} = \{1, ..., N\}$ où toutes les bandes possèdent la même largeur. Le serveur MEC dispose de ressources de calcul modélisées par la fréquence f de son processeur (donnée en cycles de CPU par seconde). La station de base est en train de servir un ensemble de N utilisateurs primaires

(PUs) qui détiennent une licence pour l'utilisation de la bande considérée, en parallèle avec M utilisateurs secondaires (SUs) qui ne dispose d'aucune licence pour accéder au spectre et qui sont définis par l'ensemble $\mathcal{M} = \{1, ..., M\}$.

Nous supposons que la communication entre la BS et tout utilisateur primaire ($n \in \mathcal{N}$) s'effectue sur une seule bande de fréquence, soit la bande n. Chaque utilisateur primaire partage la bande de fréquence avec au maximum un utilisateur secondaire qui désire décharger sa tâche vers le serveur MEC en passant par la BS. Le modèle du système est illustré à la figure 4.1.

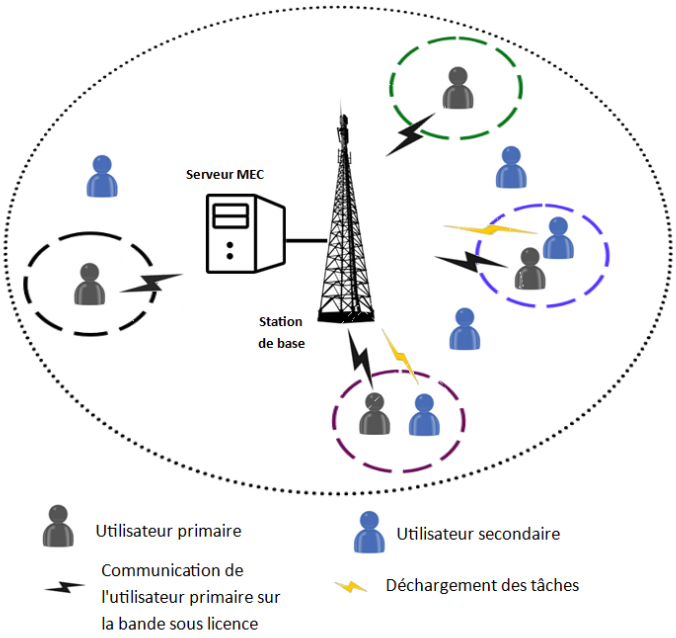

FIGURE 4.1 – Modèle du système étudié

Chaque utilisateur secondaire désire exécuter une tâche à forte intensité de calcul, décrite comme suit : $T_m = \{b_m, c_m, d_m\}$, $m \in \mathcal{M}$, où b_m (en kilo-octets, Ko) indique la taille des données d'entrée, c_m (en mégacycles) représente le nombre total de cycles de CPU nécessaires à l'exécution de la tâche et d_m désigne l'échéance (ou deadline en anglais) à ne pas dépasser pour l'exécution de la tâche.

4.1.2 Modèle de communication

Le temps étant divisé en plusieurs intervalles (appelés aussi slots de temps), au début de chaque intervalle, la station de base, diffuse un message (par exemple une trame d'annonce de type «beacon») à tous les SUs qui y sont associés pour connaitre les utilisateurs qui désirent décharger leurs tâches en utilisant les N bandes de fréquence. Chaque utilisateur qui souhaite décharger une tâche doit se manifester, en envoyant les informations sur la tâche T_m à la BS. Celle-ci va choisir les utilisateurs et leur communiquer la décision en spécifiant la bande qu'ils doivent utiliser. Ce fonctionnement est utilisé dans plusieurs standards connus comme IEEE802.11 et IEEE802.15.4, mais pour céduler des communications sans contention. Les SUs qui n'ont pas réussi à décharger durant l'intervalle du temps courant ou qui ont d'autres tâches peuvent attendre le prochain message de diffusion. Pendant une période de cohérence, le gain de puissance du canal, entre le SU-m et la station de base sur la n ième bande de fréquence est donnée comme suit (Goldsmith, 2005) :

$$g_m^n = \alpha h_m^n, \tag{4.1}$$

où h_m^n est la composante de puissance d'évanouissement à petite échelle dépendant de la fréquence et supposée être distribuée de façon exponentielle avec une moyenne unitaire, et α représente l'effet d'évanouissement à grande échelle, y compris l'affaiblissement sur le trajet et l'ombrage (en anglais shadowing), censé être indépendant de la fréquence.

Lorsque le SU-m utilise la bande de fréquence n le taux de transmission des données est donné par la formule de Shannon-Hartley et peut s'écrire comme suit :

$$R_m^n = \omega \ log_2(1 + \frac{p_m^n g_m^n}{n_0 + p_n g_n}), \tag{4.2}$$

où ω est la largeur de bande du canal, p_m et p_n sont respectivement la puissance de transmission du SU-m et la puissance e transmission du PU-n. De plus, g_n est le gain du canal entre PU-n et la BS sur le canal n. Enfin, n_0 désigne la puissance du bruit blanc additif gaussien (en anglais additive white Gaussian noise ou AWGN) de moyenne nulle et de variance unitaire.

4.1.3 Modèle de calcul

Comme illustré dans la figure 4.1, chaque utilisateur secondaire peut être choisi ou non par le serveur MEC pour décharger sa tâche T_m. Les utilisateurs qui ne seront pas sélectionnés par le serveur MEC pour décharger leurs tâches peuvent attendre le tour suivant en révisant leur échéance ou peuvent exécuter leurs tâches localement. Durant le temps de transmission de ses données, le SU qui déchargera sa tâche utilisera une seule bande de fréquence. Dans ce cas, nous pouvons calculer le délai de transmission de la tâche T_m (celle du SU_m) sur la nième bande comme suit :

$$t_{m,n}^{trans} = \frac{b_m}{R_m^n}. \tag{4.3}$$

Nous pouvons aussi calculer le temps d'exécution de la tâche T_m au niveau du serveur MEC comme suit :

$$t_m^{mec} = \frac{c_m}{f}, \tag{4.4}$$

Étant donné que le serveur MEC n'a pas à traiter une tâche dès son arrivée,

mais plutôt à ordonnancer les tâches, car il y en a plusieurs qui se concurrencent pour les ressources, un temps d'attente t_m^w, difficile à estimer et dépendant de l'ordonnancement des tâches, sera ajouté au temps de traitement des tâches pour chaque utilisateur secondaire.

Sur la base des équations ci-dessus, le délai de total de déchargement lorsque le SU-n décharge sa tâche de calcul sur le serveur MEC en utilisant la bande de fréquence m peut être donné comme suit :

$$t_m^{off} = t_{m,n}^{trans} + t_m^w + t_m^{mec} \tag{4.5}$$

4.2 Formulation du problème

Dans cette section, nous formulons le problème de la maximisation du nombre d'utilisateurs secondaires qui déchargeront leurs tâches sur le serveur MEC. Comme nous l'avons mentionné précédemment, le serveur MEC sélectionne dans chaque intervalle de temps les utilisateurs qui peuvent décharger leurs tâches en respectons leurs échéances. Le partage du spectre entre les PUs et les SUs provoque des interférences qui affectent directement la qualité de service de toutes les transmissions et par conséquent impactent les temps de début et de fin de traitement des tâches déchargées par les SUs.

L'ordonnancement des tâches au niveau du serveur MEC doit respecter l'échéance rigide imposée par chaque tâche déchargée afin qu'elle ne soit pas retardée. Sur la base du modèle du système, nous formulons un problème conjoint de déchargement, de communication et d'allocation des ressources de calcul qui a pour objectif de maximiser le nombre des tâches des SUs traitées au niveau du MEC comme suit :

$$\textbf{(P1) :} \qquad \max_{\mathbf{A},\mathbf{P}} \sum_{n=1}^{N} \sum_{m=1}^{M} \alpha_m^n \tag{4.6}$$

$$s.c.: \quad C1 : t_m^{off} \geq d_m, \ \forall m \in \mathcal{M}, \tag{4.7}$$

$$C2 : \frac{p_n g_n}{n_0 + \sum_{m=1}^{M} \alpha_m^n p_m^n g_m^n} \geq \gamma_0, \ \forall n \in \mathcal{N}, \tag{4.8}$$

$$C3 : \sum_{m=1}^{M} \alpha_m^n \leq 1, \forall n \in \mathcal{N}. \tag{4.9}$$

$$C4 : p_m \leq P_m, \ \forall m \in \mathcal{M}, \tag{4.10}$$

$$C5 : \alpha_m^n = \{0, 1\}, \forall n \in \mathcal{N} et \ \forall m \in \mathcal{M}. \tag{4.11}$$

Avec $\mathbf{A} = \{\alpha_m^n | m \in \mathcal{M}, n \in \mathcal{N}\}$ le vecteur de décision du déchargement, et $\mathbf{P}$ l'ensemble des puissances de transmission des tâches. Les contraintes C1 imposent que le temps de traitement de chaque tâche ne dépasse pas l'échéance d_m. Les contraintes C2 exigent que la puissance de chaque utilisateur secondaire ne dépasse pas la puissance maximale P_m. Afin que les transmissions secondaires soient transparentes pour celles primaires, un niveau de qualité de service (QoS) minimale γ_0 doit obligatoirement être garantie, ce qui est assuré par les contraintes C3. Les contraintes C4 et C5 définissent que chaque utilisateur secondaire doit se voir allouer une seule bande de fréquence et que pour chaque bande de fréquence, au plus un utilisateur secondaire y est servi.

4.3 Solutions proposées

Dans la section précédente, le problème (P1) est formulé sous la forme d'un problème d'optimisation en nombres entiers partiel (en anglais mixed integer problem ou MIP). De façon générale, ce genre de problème est difficile à résoudre [1]. D'après la formulation mathématique du problème (P1), on remarque qu'il tire sa difficulté de la contrainte 4.7, puisqu'il est compliqué de déterminer le temps d'attente de la tâche avant qu'elle soit traitée par le serveur MEC. Toutefois, nous n'avons pas pu développer une preuve formelle qui tranche par rapport à la NP-dureté du problème (P1). Par conséquent, nous proposons dans cette section deux algorithmes heuristiques qui obtiennent des solutions sous-optimales (P1) en résolvant de façons différentes le compromis entre la complexité algorithmique et la performance en termes du nombre de tâches déchargées.

Pour les deux algorithmes proposés, nous suggérons de procéder en deux étapes distinctes et successives. La première étape assurera le déchargement des tâches vers le serveur MEC et le partage du spectre, alors que la seconde partie concerne l'optimisation de l'ordonnancement des tâches au niveau du serveur MEC. Comme illustré à la figure 4.2, la première partie de notre solution est l'une des trois approches différentes de l'assignation des utilisateurs secondaires aux bandes de fréquence. À la différence des deux solutions MTD et Hongroise qui fournissent une décision définitive à l'étape 2, la solution génétique fournit quant à elle plusieurs décisions de déchargement/partage de spectre à l'étape 2 d'une manière itérative.

---

1. https ://fr.wikipedia.org/wiki/Problème_NP-complet

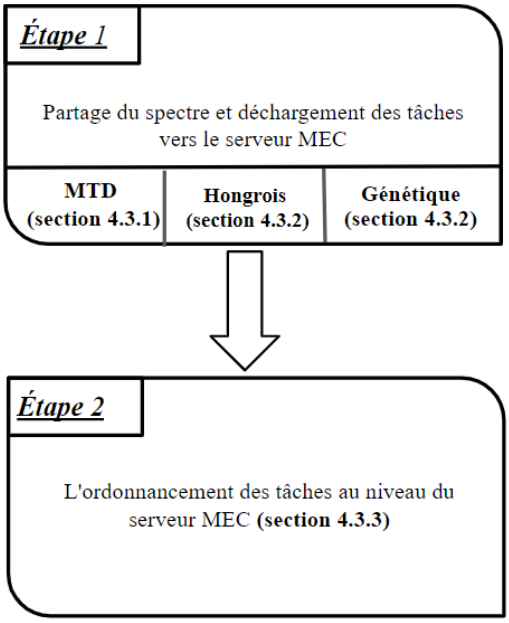

FIGURE 4.2 – Structure de la solution proposée

4.3.1 Algorithme délai de transmission minimal (MTD)

Dans le but de décider des tâches à décharger vers le serveur MEC ainsi que des bandes de fréquence qu'ils doivent utiliser, nous présentons une solution heuristique qui tente de minimiser le délai de transmission de chaque tâche. Comme chaque tâche doit être affectée à une seule bande de fréquence et que le délai de transmission dépend fortement de la bande , le choix de cette dernière est d'une grande importance. À cette fin, notre approche tente d'allouer d'une manière opportuniste la meilleure bande de fréquence à chaque utilisateur secondaire qui souhaite décharger sa tâche vers le serveur MEC.

En partageant la même bande de fréquence entre les utilisateurs primaires et secondaires, chaque SU doit transmettre sa tâche sans déranger l'utilisateur primaire. Pour ce faire, il est nécessaire de déterminer la puissance de transmission

qui permet à chaque utilisateur secondaire de transmettre sa tâche en gardant la puissance de l'interférence causée aux PUs au deçà du seuil permis. En satisfaisant les contraintes C2 par égalité, la puissance de transmission du SU_m sur la bande n peut être calculée comme suit :

$$p_m^n = \frac{\frac{p_n g_n}{\gamma_0} n_0}{g_m}, n \in \mathcal{N} \; \forall m \in \mathcal{M}. \tag{4.12}$$

En nous appuyant sur l'équation (4.2), on peut calculer le taux de transmission de chaque utilisateur secondaire sur toutes les bandes de fréquence. Chaque SU sera affecté à la bande de fréquence qui lui permet d'atteindre le taux de transmission maximal, ce qui se traduit par un délai de transmission minimal. L'ordre d'attribution des SUs aux bandes de fréquence est un facteur important, car si un SU occupe une bande, cette dernière ne peut plus être choisie, même si elle peut être meilleure pour un autre SU. Ces diverses opérations sont décrites dans Algorithme 1.

Conformément à la décomposition de notre problème, Algorithme 1 nous permet de calculer le délai de transmission pour chaque utilisateur secondaire qui servira dans la deuxième partie de notre solution pour ordonnancer les tâches au niveau du serveur MEC. La complexité de l'algorithme est celle de la boucle de la ligne 2 à 5 qui est égale à $\mathcal{O}(MN)$, puisque la complexité des lignes 6 est $\mathcal{O}(M \log M)$ et de 8 à 11 est $\mathcal{O}(M)$.

4.3.2 Algorithme Hongrois (AH)

Il est facile de remarquer que le problème de déchargement/partage de spectre ressemble beaucoup au problème d'affectation, malgré que les deux diffèrent dans leurs fonctions objectifs. Par conséquent, nous proposons d'adapter dans cette

Algorithme 1 : Délai de transmission minimal (MTD)

Entrées : $p_n, g_n, g_m, n_0, \gamma_0, \tau_m, b_m$

Sorties : p_m^n, t_m^{trans}

Initialiser $p_m^n = 0$, *pour tout* m, n

pour *chaque utilisateur secondaire* ***faire***

    pour *chaque bande de fréquence* ***faire***

        Calculer p_m^n en utilisant l'équation (7)

        Calculer R_m^n en utilisant l'équation (2)

        Calculer $t_{m,n}^{trans}$ en utilisant l'équation (3)

Trier les délais d'échéances par ordre croissant.

Commencer par l'utilisateur ayant le délai d'échéance minimum

tantque il y a des bandes de fréquence disponibles

    pour *chaque utilisateur secondaire* ***faire***

        Choisir la bande de fréquence avec le minimum $t_{m,n}^{trans}$

        Supprimer la bande de fréquence choisie de la liste des bandes de fréquence disponible

section l'algorithme hongrois comme seconde solution heuristique. L'algorithme hongrois est un algorithme très connu dans la théorie de l'optimisation et qui permet de trouver la solution optimale au problème d'affectation. De manière similaire à l'algorithme 1 et après avoir calculé les taux de transmission pour chaque SU sur toutes les bandes de fréquence, nous procédons au calcul des délais de transmission. La matrice du délai de transmission A avec N lignes et M colonnes sera l'entrée de l'algorithme hongrois (la ligne 6 d'Algorithme 1). Le but est d'associer pour chaque SU la meilleure bande de fréquence possible pour transmettre sa tâche vers le serveur MEC avec l'objectif de minimiser la somme des délais de transmission. Différemment à notre première solution, un utilisateur secondaire n'est pas forcément affecté à la bande de fréquence qui lui donne un délai de traitement minimal.

Comme décrit dans Algorithme 2, tant que des tâches des utilisateurs secondaires ne sont pas attribuées, nous recherchons des bandes de fréquence qualifiées, c'est-à-dire les entrées zéro. Si une bande de fréquence est déjà affectée à une tâche, mais qu'il est également qualifié pour une autre, alors nous mettons en place l'alternative et continuons avec la prochaine bande de fréquence qualifiée, mais si c'est la seule tâche pour laquelle la bande de fréquence est qualifiée, alors nous tenons à réaffecter toute autre bande de fréquence déjà affectée à cette tâche. Après avoir assigné autant de bandes de fréquence que nécessaire, nous déduisons ensuite le coût le plus bas pour créer une nouvelle bande de fréquence qualifiée. Ainsi, à chaque itération, nous sommes assurés de progresser vers notre objectif de trouver une affectation optimale.

Les zéros « étoilés » dans l'algorithme représentent des affectations des bandes de fréquence aux tâches des utilisateurs secondaires, et les zéros « amorcés » sont des affectations alternatives (antimatroid, 2015). En inversant les bits du chemin, nous réassignons les bandes de fréquence à leurs tâches alternatives tout

en garantissant que l'affectation continue d'être minimale. La complexité de ce deuxième algorithme est $\mathcal{O}(M^2N)$.

4.3.3 Algorithme génétique

Afin d'améliorer les performances de nos deux premières solutions heuristiques, nous proposons dans cette section une autre approche basée sur la métaheuristique génétique. Toutefois, cette amélioration, qui sera quantifiée à l'aide de simulations plus loin dans ce chapitre, viendra au coût d'une augmentation de la complexité algorithmique.

L'algorithme génétique imite les règles du principe de la survie du plus apte parmi un ensemble d'individus qui évoluent au cours de générations consécutives (époques). C'est un algorithme itératif qui s'exécute pendant un nombre prédéterminé de générations et qui se termine si la solution souhaitable est trouvée. Les éléments de base de notre algorithme génétique (détaillé dans l'Algorithme 3) sont décrits comme suite :

a) **Initialisation :** nous générons aléatoirement la population de chromosomes initiale avec des vecteurs où les indices sont l'ensemble des utilisateurs secondaires et les valeurs sont tirées de l'ensemble des bandes de fréquence. Avant de commencer notre étape d'évolution, il est nécessaire de vérifier la contrainte C3 puisque chaque utilisateur secondaire doit être affecté au plus à une seule bande de fréquence. Durant la phase d'évolution, nous évaluons les individus de la population de la génération actuelle en calculant leur fonction d'évaluation (en anglais fitness). Cette dernière représente le nombre de tâches déchargées, similaire à la fonction (4.6).

b) **Sélection :** l'opérateur de sélection est utilisé pour sélectionner des parents potentiellement bons pour le croisement afin de produire des descendants.

Algorithme 2 : Algorithme hongrois

Entrée : A

1 $A_{i,j} \leftarrow A_{i,j} - min_j A_{i,j}$ **pour** $i = 0...N-1$

2 $A_{i,j} \leftarrow A_{i,j} - min_i A_{i,j}$ **pour** $i = 0...M-1$

3 Commencer avec le premier zéro non couvert de la ligne i et couvrir la colonne correspondante j **pour** $i = 0...n-1$

4 **tantque** toutes les colonnes non couvertes

5 **tantque** zéros non couverts

6 Primer le courant découvert à zéro

7 **si** *Il y a un zéro étoilé dans cette rangée* ***alors***

Découvrir la colonne du zéro étoilé et couvrir la ligne

8 **sinon**

9 Trouver un chemin alternatif augmenté à partir du zéro primé

10 Retirer les zéros étoilés de la trajectoire et étoiler les zéros amorcés de la trajectoire

11 Enlever toutes les marques de mise en place et recouvrez tous les zéros étoilés

12 **sortir**

13 **fin si**

14 **fin tantque**

15 **si** chemin trouvé **alors**

16 Continue

17 **fin si**

18 $A^* = minA_{i,j}$ sur tous les i et j non couverts

19 $A_{i,j} = A_{i,j} - A^*$ pour toutes les colonnes j non couvertes

20 $A_{i,j} = A_{i,j} + A^*$ pour toutes les rangées i couvertes

21 **fin tantque**

22 **retourner** zéros étoilés // Ce sont toutes les attributions

L'idée derrière une telle sélection est que les meilleurs parents peuvent produire une meilleure descendance (offspring en anglais). Nous effectuons la sélection des parents en utilisant la fonction de sélection de la roulette, qui attribue une probabilité de sélection à chaque individu ou chromosome en fonction de son *fitness*. Ainsi, les chromosomes ayant la meilleure *fitness* ont plus de chances d'être sélectionnés comme parents.

c) **Croisement :** on combine deux individus sélectionnés, appelés parents, pour produire deux nouveaux individus pour la génération suivante. Nous sélectionnons aléatoirement un emplacement comme point de croisement et nous produisons deux descendants différents. Nous utilisons le croisement à point unique qui recombine deux chromosomes parents autour d'un point de croisement pour générer de nouveaux chromosomes descendants.

d) **Mutation :** on consiste à changer la bande de fréquence affectée à SU pour certains individus de la population actuelle de façon aléatoire afin de produire de nouveaux individus pour la population de la génération suivante. Après avoir produit les descendances, nous effectuons une mutation locale d'un seul bit à chaque gène du chromosome candidat choisi au hasard pour la mutation afin d'obtenir une plus grande diversification.

e) **Clôture :** à la fin, on fait une filtration des chromosomes identiques (jumeaux) de la population pour assurer une quantité appropriée d'individus variés dans la population.

4.3.4 Algorithme d'ordonnancement

Après le choix de la meilleure bande de fréquence pour chaque utilisateur secondaire, le serveur MEC commence à recevoir les tâches selon leurs délai de transmission. Comme chaque tâche ne doit pas dépasser son délai d'échéance, le

Algorithme 3 : Algorithme génétique

Entrées : $\mathcal{M}, \mathcal{N}, p_n, g_n, g_m, n_0, \gamma_0, d_m, b_m$

Sorties : La meilleure solution de déchargement/partage de spectre

Initialisation : générer aléatoirement la population initiale avec des vecteurs où les indices sont l'ensemble de M et les variables sont l'ensemble de N et définir l'itération de l'algorithme Iteration = 1

Vérifier et modifier la population initiale si les contraintes ne sont pas satisfaites.

tantque *Iteration* $\leq MaxIteration$ ***faire***

    pour *chaque individu de la population* ***faire***

        Calculez le fitness : fit = m

    tantque *il existe des individus dans la population* ***faire***

        Sélectionnez $parent_1$ *avec* $fit = max(m)$

        Sélectionnez $parent_2$ *au hasard dans toute la population*

        pour *i = 1 à n* ***faire***

            Générer point du croisement aléatoire p_{cross}

            $enfant_1 = parent_1[1, 2, ..., p_{cross}] + parent_2[p_{cross} + 1, ..., n]$

            $enfant_2 = parent_2[1, 2, .., p_{cross}] + parent_1[p_{cross} + 1, .., n]$

        pour *tous les descendants générés* ***faire***

            si *la probabilité de mutation s'applique* ***alors***

                Mutation des descendants actuels

Contrôler et modifier la génération suivante

Effectuer la sélection des individus les plus aptes en utilisant la procédure de sélection par roulette.

Iterartion = Iteration +1

serveur MEC doit les traiter avant ce délai. Le serveur MEC traite une tâche à la fois, lorsqu'il reçoit en premier une tâche dont le délai d'échéance est élevé, le temps de traitement des utilisateurs secondaires peut augmenter. Pour remédier à cette situation, nous avons opté pour la méthode d'ordonnancement préemptif des tâches au niveau du serveur MEC. En effet, même si une tâche arrive en premier et que son délai d'échéance est supérieur à celui d'une tâche qui arrive après, le serveur suspend la première tâche pour exécuter la seconde.

L'article (Baptiste, 1999), étudie le problème d'ordonnancement préemptif sur une seule machine avec l'objectif de minimiser le nombre des tâches en retard sur une seule machine en prenant compte leurs temps d'arrivée $(1|pmtn, r_j|\sum U_j)$. En se basant sur l'approche de (Baptiste, 1999), nous avons développé un algorithme d'ordonnancement des tâches au niveau du serveur MEC pour notre problème tout en garantissant une maximisation des tâches traitées pour atteindre l'objectif annoncé qui est de maximiser le nombre d'utilisateurs secondaires qui déchargeront leurs tâches. La minimisation du nombre des tâches en retard revient à trouver un ordonnancement sur lequel le nombre de tâches programmées avant leur date d'échéance est maximal. Partant de ce principe, l'article (Baptiste, 1999) a proposé une solution en programmation dynamique avec une complexité $\mathcal{O}(n^4)$ (n est le nombre des tâches à ordonnancer) qui est inférieure par rapport à d'autres travaux.

Nous considérons un ensemble de tâches $P = \{T_1, ..., T_m\}$, chaque tâche étant décrite par un temps d'arrivée $t_{i,n}^{trans}$, un temps d'échéance d_i et un temps d'exécution au niveau du serveur MEC t_i^{mec} (avec $t_{i,n}^{trans} + t_i^{mec} \leq d_i$). Dans le cadre du problème à résoudre, il est nécessaire de trouver un sous-ensemble maximal de tâches qui soit réalisable et qui puisse être ordonnancé de manière préemptive sur le serveur MEC. Pour cela, nous envisageons un sous-ensemble réalisable O de tâches et nous allons utiliser la méthode d'ordonnancement préemptif de Jackson

(Carlier et Pinson, 2004) pour O (T_{PS_O}), c'est-à-dire que chaque fois que le serveur est libre et qu'une tâche dans O est disponible, on va ordonnancer la tâche $T_i \in O$ pour laquelle d_i est le plus petit afin de calculer le délai de traitement des tâches des utilisateurs secondaires au niveau du MEC tout en incluant le délai d'attente avant que la tâche ne soit traitée.

Comme nous l'avons mentionné ci-dessus, puisqu'on considère un ordonnancement préemptif pour traiter les tâches au niveau du serveur MEC, si une tâche T_j devient disponible pendant que T_i est en cours de traitement, T_i sera arrêtée et T_j lancée si bien sûr d_j est strictement inférieur à d_i, sinon la tâche T_i va continuer son traitement. La condition pour laquelle nous pouvons affirmer que le sous-ensemble O est réalisable consiste à ce que toutes les tâches de T_{PS_O} soient planifiées avant leurs délais d'échéance.

En vue de répondre à notre problème de planification, on suppose que toutes les tâches sont triées par ordre croissant de leur délai d'échéance et pour tout entier a et toute tâche T_k, $S_k(a)$ est l'ensemble des tâches $\{T_i\}$ tels que $a \leq t_{i,n}^{trans}$ et $i \leq k$. Dans le but de traiter cette problématique, on considère a et b comme des valeurs quelconques telles que $a \leq b$ et on peut définir $C_k(a;m)$ comme le temps minimal pour lequel m tâches dans $S_k(a)$ peuvent être traitées, et $\pi_k(a;b)$ le nombre maximal des tâches dans $S_{k-1}(a)$ qui peuvent être planifiées avant b, et $\mu_k(a;b)$ le plus grand relâchement (slack) possible sur l'intervalle $[r_k;b]$ parmi les ensembles qui réalisent $\pi_k(a;b)$.

En effet, nous pouvons formuler le problème de la maximisation des tâches traitées au niveau du serveur MEC sans dépasser leur délai d'échéance et en se basant toujours sur l'article (Baptiste, 1999) de la manière suivante :

— Si $T_k \notin S_k(a)$ donc $C_k(a;m) = C_{k-1}(a;m)$, et si $T_k \in S_k(a)$ donc $C_k(a;m) = f_k(\min(C_{k-1}(a;m); \max(r_k; C_{k-1}(a;m1)) + t_k^{mec}$;

$\min_{t_{u,n}^{trans} \leq t_{k,n}^{trans}}(C_{k-1}(t_{u,n}^{trans}; m1\pi_k(a; t_{u,n}^{trans})) + \max(0; t_{k,n}^{trans}\mu_k(a; t_{u,n}^{trans}))))$

— Pour tout $a \leq b$, $\pi_k(a; b) = \max\{m | C_{k-1}(a; m) \leq b\}$.

— $\forall a, b$ et $\forall J_k$ de sorte que $a \leq t_{k,n}^{trans} \leq b$,
$\mu_k(a; b) = \max(b - \max(C_{k-1}(a; \pi_k(a; b)); t_{k,n}^{trans})$;
$\max(\mu_k(a; t_{v,n}^{trans})$+b - $C_{k-1}(t_{v,n}^{trans}; \pi_k(t_{v,n}^{trans}; b)))))$ avec $t_{k,n}^{trans} \leq t_{v,n}^{trans} < b$.

Nous cherchons à déterminer la plus grande valeur de m telle que $C_m(\min t_{i,n}^{trans}; m)$ soit finie. Notre première étape comme montrée dans l'algorithme 4 est de calculer la valeur $C_1(t_{j,n}^{trans}, m)$ (ligne 1 à 3) pour tous les $t_{j,n}^{trans}$ et pour toutes les valeurs de m dans $[1, m]$. Après que plusieurs tâches soient arrivées au serveur MEC, nous pouvons, grâce aux trois notions décrites ci-avant, calculer $C_1(t_{j,n}^{trans}; m)$ avec $k \in [2, m]$.

Algorithme 4 : Ordonnancement (Baptiste, 1999)

Entrées : d_m, t_j^{trans}, f_m

Sorties : m

1 **pour** *chaque* t_j^{trans} *avec* $j \in \mathcal{M}$ ***faire***
2 **pour** *chaque nombre de tâches* m ***faire***
3 Calculer $C_1(\min t_j^{trans}, m)$
4 **pour** *pour* $t_j^{trans} \leq t_u^{trans}$ *avec* $j, u \in \mathcal{M}$ ***faire***
5 Calculer $\pi_k(t_j^{trans}, t_u^{trans})$
6 Calculer $\mu_k(t_j^{trans}, t_u^{trans})$
7 **pour** *chaque* t_j^{trans}
8 **pour** *chaque nombre de tâches* m ***faire***
9 *Calculer* $C_k(min\ t_j^{trans},\ m)$
10

TABLEAU 4.1 – Paramètres des simulations

Notation	***Description***	***Valeur***
N	Nombre des bandes de fréquence	[2, 20]
M	Nombre des utilisateurs secondaires	[2, 20]
w	Largeur de la bande de fréquence	20 MHz
f_m	Capacité du serveur	4 GHz
b_m	Taille des données des tâches	[100, 400] KB
d_m	Délai d'échéance des tâches	[5, 25] s
n_0	Puissance du bruit	-100 dBm
r_c	Taux de croisement	0.8(80%)
r_m	Taux de mutation	0.05(5%)

4.4 Évaluation des performances

4.4.1 Paramètres des simulations

Dans nos évaluations, nous considérons un scénario où tous les utilisateurs secondaires connectés au réseau transmettent leurs tâches simultanément. Nous étudions l'impact du nombre des utilisateurs secondaires, de la taille des données déchargées, du nombre de bandes de fréquence et du délai d'échéance des tâches sur les performances des différents algorithmes proposés. Nos principaux paramètres de simulation sont résumés dans le tableau 4.1.

Pour notre solution d'algorithme génétique, nous avons effectué différentes simulations pour définir deux paramètres importants, à savoir le nombre maximal d'itérations maxIter et la taille de la population. La figure 4.3 présente l'effet de la taille de la population sur les performances de l'algorithme génétique. La figure

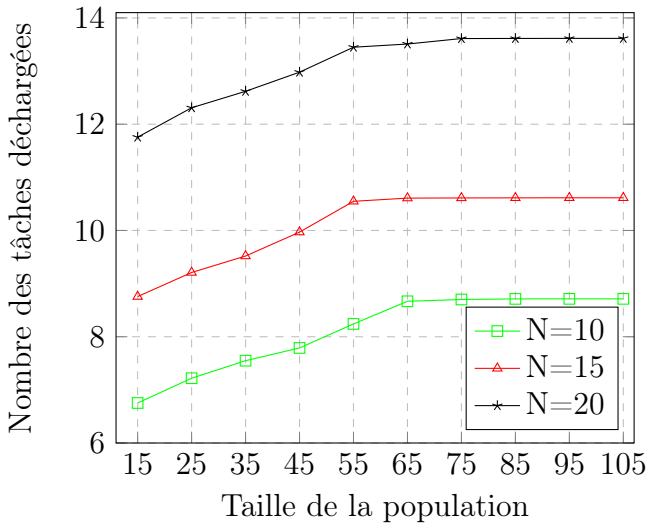

FIGURE 4.3 – Nombres des tâches déchargées vs. la taille de la population et avec $M = 15$.

sert aussi à retrouver la valeur la plus intéressante pour ce paramètre important, et qui sera utilisé dans les simulations subséquentes. Pour différentes valeurs de N et avec un nombre fixe d'utilisateurs secondaires, le nombre de tâches déchargées vers le serveur périphérique tend vers le nombre maximal de bandes de fréquence. Lorsque la taille de la population augmente au-delà de 65, le nombre de tâches déchargées sature. Par conséquent, dans les prochaines simulations, nous fixons la taille de la population à 65 chromosomes.

Dans la figure 4.4, le nombre des tâches déchargée est représenté par rapport au nombre d'itérations (alias le nombre de générations) pour différentes valeurs de N. Comme prévu, les performances de l'algorithme génétique augmentent avec le nombre d'itérations. Cependant, l'amélioration des performances devient négligeable lorsque le nombre des itérations dépasse 900 itérations. Par conséquent, dans les prochaines simulations, nous fixons le nombre d'itérations à 900. Il faut noter que l'algorithme génétique dispose d'une flexibilité puisqu'il peut être arrêté

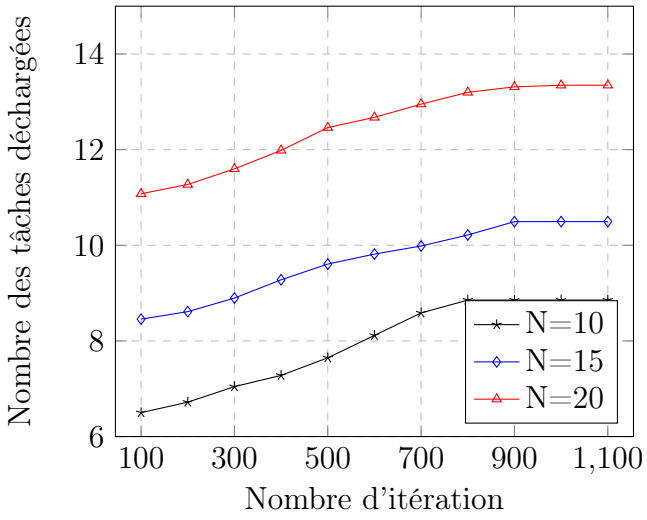

FIGURE 4.4 – Nombres des tâches déchargées vs. le nombre d'itérations avec $M = 15$.

à tout moment et le meilleur chromosome peut être sélectionné et choisi comme solution.

4.4.2 Étude comparative

Avant de détailler les résultats de nos simulations, nous tenons à souligner qu'il est possible de calculer le nombre des tâches des utilisateurs secondaires qui vont être déchargées sur le serveur MEC avec d'autres méthodes différentes. Comme nous l'avons évoqué au début de cette section, le choix de la bande de fréquence qui sera attribuée à chaque utilisateur secondaire reste un défi difficile à relever. Malgré les trois solutions que nous avons proposées, il se peut que la bande de fréquence attribuée à l'utilisateur secondaire ne soit pas le meilleur choix. Pour contourner ce risque, le calcul du nombre de tâches déchargées en utilisant toutes les combinaisons de bandes de fréquence possibles grâce à la force brute nous permet d'obtenir la solution optimale. Cette dernière souffre malheureusement

d'une énorme complexité de calcul.

De plus, nous comparons les solutions proposées avec une technique naïve décrite dans l'algorithme 4. Cette technique calcule le délai de transmission pour chaque utilisateur secondaire avec un choix aléatoire de la bande de fréquence. Pour l'ordonnancement des tâches au niveau du serveur MEC, elle utilise la méthode premier arrivé, premier servi ou PAPS (en anglais first in first out ; FIFO). Toutefois, le temps d'attente au tampon du serveur MEC oblige certains utilisateurs secondaires à manquer leur délai d'échéance, ce qui ne garantit pas une maximisation des tâches déchargées sur le serveur MEC et ce qui est loin de notre objectif.

4.4.3 Analyse des résultats de simulation

La figure 4.5 compare les performances des cinq algorithmes en termes du nombre des tâches déchargées en variant le nombre d'utilisateurs secondaires présents dans le système. On peut observer que notre approche (algorithme génétique) converge vers la solution optimale. Comme prévu, le nombre des tâches déchargées en respectant leur délai d'échéance augmente avec l'augmentation du nombre d'utilisateurs secondaire, ceci est dû à une forme de diversité de sélection dont peuvent profiter tous les algorithmes, mais à des degrés différents. Lorsque le nombre des utilisateurs secondaires est égale à 2, tous les algorithmes donnent le même nombre des tâches déchargées, ce qui est dû au grand nombre des bandes de fréquence. Cependant, lorsque le nombre des utilisateurs secondaires augmente, il y a une divergence claire entre les résultats des algorithmes puisqu'ils explorent d'une manière différente l'espace de recherche.

En raison de la très grande complexité de calcul de l'algorithme de force brute, il a été impossible de trouver une performance optimale pour un nombre d'utilisa-

Algorithme 5 : Maximiser le nombre d'utilisateurs

Entrées : $\mathrm{p}_n, g_n, g_m, n_0, \gamma_0, \tau_m, b_m, d_m, f_m$

Initialiser $p_m^n = 0$
Sélectionner les utilisateurs de manière aléatoire
pour *chaque utilisateur secondaire* ***faire***
    pour *chaque bande de fréquence* ***faire***
        Calculer p_m^n en utilisant eq (7)
        Calculer R_m^n en utilisant eq (2)
        Calculer $t_{m,n}^{trans}$ en utilisant eq (3)
pour *chaque utilisateur secondaire* ***faire***
    Choisissez la bande de fréquence au hasard
Utiliser la méthode FIFO pour ordonner les tâches au niveau du serveur MEC.
pour *chaque utilisateur secondaire* ***faire***
    Calculer $T_{m,n}^{off}$
    si $T_{m,n}^{off} \leq d_m$ ***faire***
        La tâche peut être déchargée

Sorties : m

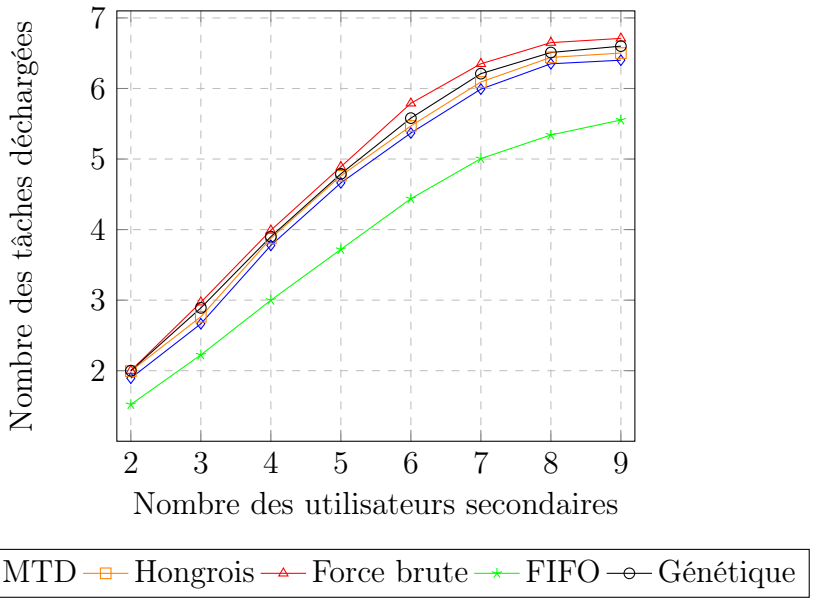

FIGURE 4.5 – Nombres des tâches déchargées vers le serveur MEC avec $N = 9$.

teurs secondaires supérieur à neuf. Par conséquent, nous traçons dans la figure 4.5 des performances similaires à celles de la figure 4.6, mais pour des réseaux encore plus denses avec jusqu'à 19 utilisateurs secondaires en présence de 15 utilisateurs primaires. À cause de l'abondance de bandes de fréquence, l'algorithme génétique peut décharger presque la totalité des tâches des utilisateurs secondaires quand ils sont entre deux et douze. Pour les deux algorithmes heuristiques (MTD et hongrois), plus qu'on se rapproche au nombre maximal des bandes de fréquence, la solution basée sur l'algorithme hongrois commence à creuser un écart de performance sensible par rapport à la solution MTD.

Une autre simulation qu'on va analyser pour étudier les performances des solutions proposées dans cette thèse concerne l'impact de la variation du nombre des bandes de fréquence sur le nombre des tâches déchargées. Comme le montre la figure 4.7, pour un nombre fixe d'utilisateurs secondaire ($M = 4$ et $M = 8$) et avec l'augmentation du nombre des bandes de fréquence on peut voir que toutes

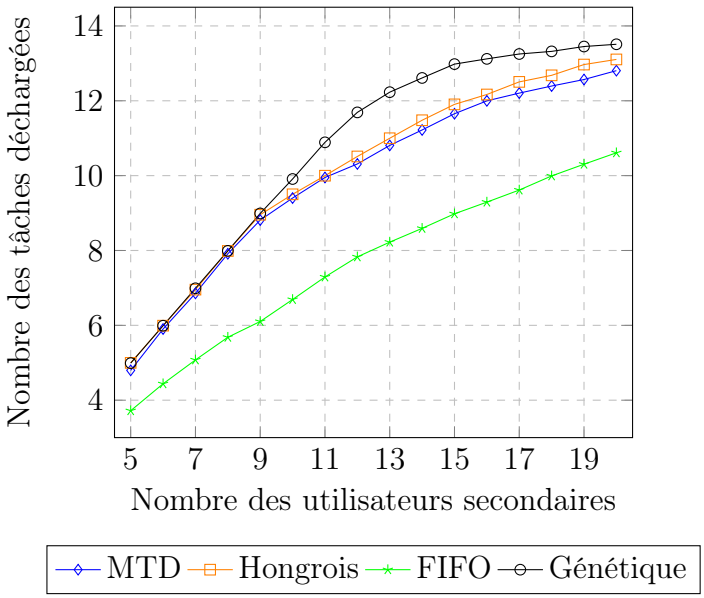

FIGURE 4.6 – Nombres des tâches détachées vers le serveur MEC avec $N = 15$.

nos solutions convergent vers le nombre maximal des utilisateurs secondaires, mais avec des rythmes différents. La différence entre les résultats des solutions basées sur la force brute et l'algorithme génétique sont très proches quand le nombre des utilisateurs secondaires et plus petits par rapport au nombre de bandes de fréquence (figure 4.7 (a)). Par contre avec l'augmentation du nombre des utilisateurs secondaires, on peut constater que l'écart de performance entre les deux solutions.

La figure 4.8 évalue les performances des algorithmes étudiés en utilisant une nouvelle métrique qui est le temps moyen entre la terminaison des tâches et leurs échéances (EMT). Comme prévu, on remarque que la solution basée sur la méthode FIFO traite les tâches bien avant leurs délais d'échéances, ce qui explique en grande partie les mauvaises de cette solution. D'autre part, l'algorithme optimal essaye de profiter de chaque fraction de temps afin d'ordonnancer le maximum de tâches. Dans ce même esprit, le comportement de l'algorithme génétique proposé imite celui de la force brute afin d'approcher ses performances en termes

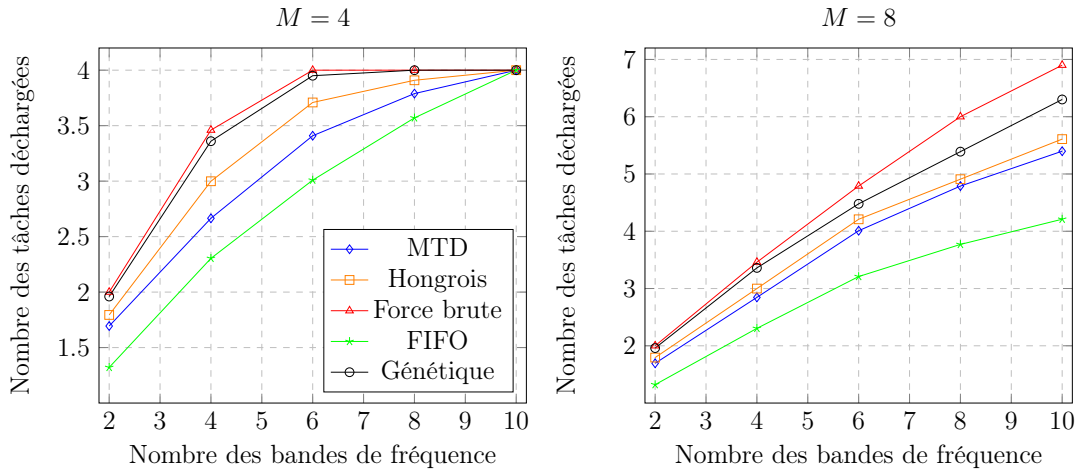

FIGURE 4.7 – Nombres des tâches déchargées vers le serveur MEC avec une variation des bandes de fréquence.

de nombre de tâches déchargées. Par rapport à la solution génétique et de force brute qui la traite plus tard. Toutefois, on remarque qu'en augmentant le nombre d'utilisateurs secondaires, la métrique étudiée EMT augmente également ce qui se traduit par la saturation des performances dans les figures précédentes.

4.5 Conclusion

Nous rappelons que l'objectif de ce chapitre était de présenter notre approche pour résoudre le problème de la maximisation du nombre de tâches des utilisateurs secondaires déchargées vers le serveur MEC. L'approche proposée procède en deux phases. Dans la première étape, nous développons trois solutions qui permettent de choisir les tâches à décharger et d'optimiser le choix d'affectation des tâches aux bandes de fréquence disponible. Dans la deuxième étape, nous élaborons une solution existante pour la résolution de problème d'ordonnancement au niveau du serveur MEC. Nous avons aussi évalué les performances des solutions proposées par le biais d'une série de simulations. Nous avons montré que l'algo-

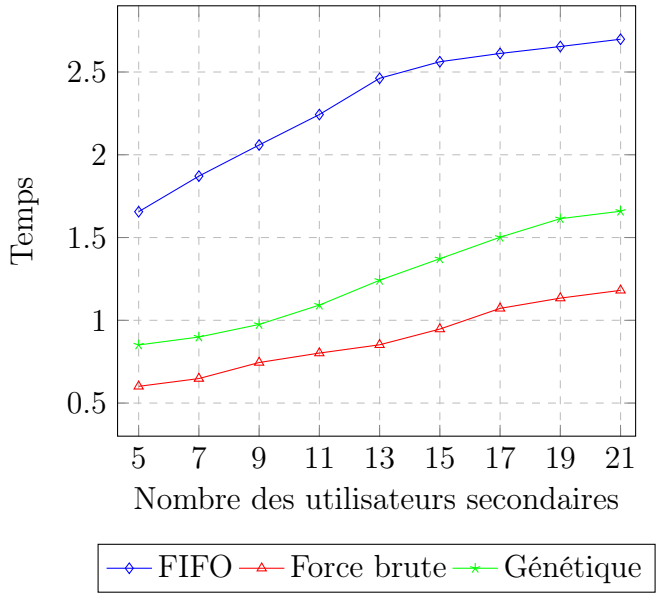

FIGURE 4.8 – La moyenne des échéances-moins-terminaisons ou EMT vs. le nombre des utilisateurs secondaires.

rithme génétique permet d'atteindre des performances quasi-optimales. De plus, grâce à leurs complexités algorithmiques très réduites et leurs performances acceptables, les deux autres heuristiques proposées peuvent aussi être intéressantes pour des implémentations pratiques où le temps de réponse pour prendre une décision constitue un paramètre important.

CHAPITRE V

CONCLUSION

Dans ce mémoire, nous avons étudié le problème de la maximisation du nombre des utilisateurs secondaires qui pourront décharger leurs tâches vers le serveur périphérique, et ce, tout en respectant leurs délais d'échéance. Le travail que nous avons réalisé est un cas général qui peut être adapté à différents cas de figure. Le fait de traiter individuellement les utilisateurs secondaires est une proposition que l'on ne retrouve pas souvent dans la littérature, pour cela, nous avons développé différentes solutions à notre problématique afin de faire une étude comparative des performances et ainsi connaître l'efficacité de nos approches pour atteindre notre objectif.

Après la présentation du modèle du système et la formulation de notre problématique, nous avons développé deux solutions heuristiques qui se caractérisent par une complexité algorithmique très réduite. Étant donné que chaque utilisateur secondaire partage la même bande de fréquence avec un utilisateur primaire, le choix de la bande impacte fortement le délai de traitement général de la tâche de l'utilisateur secondaire. Pour cela, un choix de bande de fréquence qui garantit un délai de transmission minimal était développé dans notre première solution (MTD), et un choix basé sur l'algorithme hongrois était notre deuxième proposition. Pour la partie concernant l'ordonnancement des tâches au niveau du serveur

périphérique, on s'est basé sur l'article (Baptiste, 1999) qui permet de prendre en compte le délai d'attente des tâches avant qu'elles soient traitées au niveau du serveur périphérique.

Comme nous avons discuté dans notre rapport, la partie la plus difficile au niveau de cette problématique concerne l'attribution de la bande de fréquence aux tâches des utilisateurs secondaires. Pour améliorer davantage cette décision (comparé aux deux heuristiques précédentes), nous avons développé une solution basée sur l'algorithme génétique qui nous permet de faire un choix de bande de fréquence plus intelligent et qui nous permet ainsi de maximiser le nombre d'utilisateurs secondaires. Afin d'évaluer les performances des trois algorithmes proposés, nous les avons comparés à deux benchmarks. Le premier benchmark est un algorithme de force brute qui permet d'atteindre les performances optimales mais qui souffre d'une complexité algorithmique exorbitante. Il constitue ainsi une borne supérieure qui a permis de démontrer la quasi-optimalité de l'algorithme génétique et l'intérêt des deux autres heuristques, MTD et AH. Le deuxième benchmark est un algorithme naïf qui combine une allocation aléatoire des bandes de fréquence et un ordonnancement PAPS. Il a constitué une borne inférieure aux performances des algorithmes proposés. Ces derniers ont surpassé grandement le deuxième benchmark.

Cette problématique est sujette d'amélioration. Sur le long terme, une étude sera menée sur les trois points suivants :

— développer des solutions plus intelligentes basées sur des métaheuristiques plus élaborées ou sur l'apprentissage automatique ;
— considérer plusieurs serveurs qui peuvent coopérer ou non pour maximiser le nombre des utilisateurs secondaires qui peuvent décharger leurs tâches ;
— appliquer notre solution sur un cas d'utilisation pratique.

RÉFÉRENCES

Aissioui, A., Ksentini, A., Gueroui, A. M. et Taleb, T. (2018). On enabling 5g automotive systems using follow me edge-cloud concept. *IEEE Transactions on Vehicular Technology*, *67*(6), 5302–5316.

Akyildiz, I. F., Lee, W.-Y., Vuran, M. C. et Mohanty, S. (2008). A survey on spectrum management in cognitive radio networks. *IEEE Communications magazine*, *46*(4), 40–48.

Al-Hourani, A. (2020). Interference modeling in low-altitude unmanned aerial vehicles. *IEEE Wireless Communications Letters*, *9*(11), 1952–1955.

Alfakih, T., Hassan, M. M., Gumaei, A., Savaglio, C. et Fortino, G. (2020). Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on sarsa. *IEEE Access*, *8*, 54074–54084.

Ali, B., Gregory, M. A. et Li, S. (2021). Multi-access edge computing architecture, data security and privacy : A review. *IEEE Access*, *9*, 18706–18721.

antimatroid, T. T. (2015). A greedy approximation algorithm for the linear assignment problem. disponible sur : `https://antimatroid.wordpress.com/`. Accès le : 2022-12-21.

Asadi, A. et Mancuso, V. (2013). A survey on opportunistic scheduling in wireless communications. *IEEE Communications Surveys Tutorials*, *15*(4), 1671–1688.

Baptiste, P. (1999). An o(n4) algorithm for preemptive scheduling of a single machine to minimize the number of late jobs. *Operations Research Letters*, *24*(4), 175–180.

Bi, J., Yuan, H., Duanmu, S., Zhou, M. et Abusorrah, A. (2021). Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization. *IEEE Internet of Things Journal*, *8*(5), 3774–3785.

Biswas, N., Mirghasemi, H. et Vandendorpe, L. (2021). Sharing is caring : A mobile edge computing perspective. Dans *les actes de IEEE International*

Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 1298–1303.

Bonomi, F., Milito, R., Zhu, J. et Addepalli, S. (2012). Fog computing and its role in the internet of things. Dans *les actes de Mobile Cloud Computing*, MCC '12, p. 13–16.

Brito, M. S., Hoque, S., Magedanz, T., Steinke, R., Willner, A., Nehls, D., Keils, O. et Schreiner, F. (2017). A service orchestration architecture for fog-enabled infrastructures. Dans *les actes de 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, 127–132.

Carlier, J. et Pinson, E. (2004). Jackson's pseudo-preemptive schedule and cumulative scheduling problems. *Discrete Applied Mathematics*, *145*(1), 80–94. Graph Optimization IV.

Chang, H., Hari, A., Mukherjee, S. et Lakshman, T. V. (2014). Bringing the cloud to the edge. Dans *les actes de IEEE Conference on Computer Communications Workshops*, 346–351.

Dai, Y., Lin, B., Che, Y. et Lyu, L. (2022). Uav-assisted data offloading for smart container in offshore maritime communications. *China Communications*, *19*(1), 153–165.

Dolui, K. et Datta, S. K. (2017). Comparison of edge computing implementations : Fog computing, cloudlet and mobile edge computing. Dans *les actes de Global Internet of Things Summit (GIoTS)*, 1–6.

Fang, X., Cai, Z., Tang, W., Luo, G., Luo, J., Bi, R. et Gao, H. (2020). Job scheduling to minimize total completion time on multiple edge servers. *IEEE Transactions on Network Science and Engineering*, *7*(4), 2245–2255.

Feng, W.-J., Yang, C.-H. et Zhou, X.-S. (2019). Multi-user and multi-task offloading decision algorithms based on imbalanced edge cloud. *IEEE Access*, *7*, 95970–95977.

Goldsmith, A. (2005). *Wireless communications*. Cambridge university press.

Hasegawa, M., Hirai, H., Nagano, K., Harada, H. et Aihara, K. (2014). Optimization for centralized and decentralized cognitive radio networks. *Proceedings of the IEEE*, *102*(4), 574–584.

He, X. et Dou, W. (2020). Offloading deadline-aware task in edge computing. Dans *les actes de IEEE International Conference on Cloud Computing (CLOUD)*, 28–30.

Homayouni, S., Ghorashi, S. A. et Azizzadeh, F. (2011). Spectrum sharing by sub-banding in cognitive radio networks. Dans *les actes de International eConference on Computer and Knowledge Engineering (ICCKE)*, 322–326.

Hu, Y. C., Patel, M., Sabella, D., Sprecher, N. et Young, V. (2015). Mobile edge computing—a key technology towards 5g. *ETSI white paper*, *11*(11), 1–16.

Hung, S.-C., Hsu, H., Lien, S.-Y. et Chen, K.-C. (2015). Architecture harmonization between cloud radio access networks and fog networks. *IEEE Access*, *3*, 3019–3034.

Jian, C., Chen, J., Ping, J. et Zhang, M. (2019). An improved chaotic bat swarm scheduling learning model on edge computing. *IEEE Access*, *7*, 58602–58610.

Jošilo, S. et Dán, G. (2021). Joint management of wireless and computing resources for computation offloading in mobile edge clouds. *IEEE Transactions on Cloud Computing*, *9*(4), 1507–1520.

Kai, C., Zhou, H., Yi, Y. et Huang, W. (2021). Collaborative cloud-edge-end task offloading in mobile-edge computing networks with limited communication capability. *IEEE Transactions on Cognitive Communications and Networking*, *7*(2), 624–634.

Kim, Y., Song, C., Han, H., Jung, H. et Kang, S. (2020). Collaborative task scheduling for iot-assisted edge computing. *IEEE Access*, *8*, 216593–216606.

Koulali, S., Sabir, E., Taleb, T. et Azizi, M. (2016). A green strategic activity scheduling for uav networks : A sub-modular game perspective. *IEEE Communications Magazine*, *54*(5), 58–64.

Li, Y., Qi, F., Wang, Z., Yu, X. et Shao, S. (2020). Distributed edge computing offloading algorithm based on deep reinforcement learning. *IEEE Access*, *8*, 85204–85215.

Liang, L., Ye, H. et Li, G. Y. (2019). Spectrum sharing in vehicular networks based on multi-agent reinforcement learning. *IEEE Journal on Selected Areas in Communications*, *37*(10), 2282–2292.

Liu, W.-J., Hu, B.-J., Wei, Z.-H., Lv, J.-M., Tan, T. et Liu, X. (2016). A csgc algorithm based partial spectrum sharing scheme in cognitive lte-a two-tier heterogeneous networks. Dans *les actes de IEEE MTT-S International Wireless Symposium (IWS)*, 1–4.

Mao, Y., You, C., Zhang, J., Huang, K. et Letaief, K. B. (2017). A survey on

mobile edge computing : The communication perspective. *IEEE Communications Surveys Tutorials*, *19*(4), 2322–2358.

Muniswamaiah, M., Agerwala, T. et Tappert, C. C. (2021). A survey on cloudlets, mobile edge, and fog computing. Dans *les actes de IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/ IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, 139–142. IEEE.

Nath, S. et Wu, J. (2020). Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems. *Intelligent and Converged Networks*, *1*(2), 181–198.

Nguyen, M.-H. T., Garcia-Palacios, E., Do-Duy, T., Nguyen, L. D., Mai, S. T. et Duong, T. Q. (2021). Spectrum-sharing uav-assisted mission-critical communication : Learning-aided real-time optimisation. *IEEE Access*, *9*, 11622–11632.

Pandit, Shweta, S. G. (2017). n overview of spectrum sharing techniques in cognitive radio communication system. *Wireless Networks*, *23*, 1572–8196.

Peng, K., Leung, V., Xu, X., Zheng, L., Wang, J. et Huang, Q. (2018). A survey on mobile edge computing : Focusing on service adoption and provision. *Wireless Communications and Mobile Computing*, *2018*.

Qian, L., Wu, Y., Yu, N., Jiang, F., Zhou, H. et Quek, T. Q. (2021). Learning driven noma assisted vehicular edge computing via underlay spectrum sharing. *IEEE Transactions on Vehicular Technology*, *70*(1), 977–992.

Qin, Z., Zhou, X., Zhang, L., Gao, Y., Liang, Y.-C. et Li, G. Y. (2020). 20 years of evolution from cognitive to intelligent communications. *IEEE Transactions on Cognitive Communications and Networking*, *6*(1), 6–20.

Roman, R., Lopez, J. et Mambo, M. (2018). Mobile edge computing, fog et al. : A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, *78*, 680–698.

Santana, G. M. D., Cristo, R. S., Dezan, C., Diguet, J.-P., Osorio, D. P. M. et Branco, K. R. L. J. C. (2018). Cognitive radio for uav communications : Opportunities and future challenges. Dans *les actes de International Conference on Unmanned Aircraft Systems (ICUAS)*, 760–768.

Shah, S. A. A., Ahmed, E., Imran, M. et Zeadally, S. (2018). 5g for vehicular communications. *IEEE Communications Magazine*, *56*(1), 111–117.

Sharmila, A. et Dananjayan, P. (2019). Spectrum sharing techniques in

cognitive radio networks – a survey. Dans *les actes de 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, 1–4.

Song, H., Gu, B., Son, K. et Choi, W. (2022). Joint optimization of edge computing server deployment and user offloading associations in wireless edge network via a genetic algorithm. *IEEE Transactions on Network Science and Engineering*, *9*(4), 2535–2548.

Sorkhoh, I., Ebrahimi, D., Atallah, R. et Assi, C. (2019). Workload scheduling in vehicular networks with edge cloud capabilities. *IEEE Transactions on Vehicular Technology*, *68*(9), 8472–8486.

Wadhonkar, A. et Theng, D. (2016). A survey on different scheduling algorithms in cloud computing. Dans *les actes de International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, 665–669.

Wang, C., Yu, F. R., Liang, C., Chen, Q. et Tang, L. (2017). Joint computation offloading and interference management in wireless cellular networks with mobile edge computing. *IEEE Transactions on Vehicular Technology*, *66*(8), 7432–7445.

Xu, X., Li, D., Dai, Z., Li, S. et Chen, X. (2019). A heuristic offloading method for deep learning edge services in 5g networks. *IEEE Access*, *7*, 67734–67744.

You, M., Song, Q., Wang, Q., Lv, T. et Du, H. (2010). Spectrum license assignment. Dans *les actes de 2010 2nd IEEE International Conference on Information Management and Engineering*, 378–380.

Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J. et Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms : A complete survey. *Journal of Systems Architecture*, *98*, 289–330.

Zeng, M. et Fodor, V. (2020). Dynamic spectrum sharing for load balancing in multi-cell mobile edge computing. *IEEE Wireless Communications Letters*, *9*(2), 189–193.

Zhai, C., Zhang, W. et Mao, G. (2014). Cooperative spectrum sharing between cellular and ad-hoc networks. *IEEE Transactions on Wireless Communications*, *13*(7), 4025–4037.

Zhang, J., Zhou, X., Ge, T., Wang, X. et Hwang, T. (2021). Joint task

scheduling and containerizing for efficient edge computing. *IEEE Transactions on Parallel and Distributed Systems*, *32*(8), 2086–2100.

Zhang, L., Xiao, M., Wu, G., Alam, M., Liang, Y.-C. et Li, S. (2017). A survey of advanced techniques for spectrum sharing in 5g networks. *IEEE Wireless Communications*, *24*(5), 44–51.

Zhao, J., Li, Q., Gong, Y. et Zhang, K. (2019). Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks. *IEEE Transactions on Vehicular Technology*, *68*(8), 7944–7956.