Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

Βελτιστοποίηση Μέσω Βαθιάς Ενισχυτικής Μάθησης για Ασφαλή Εκφόρτωση Δεδομένων σε Ασύρματα Δίκτυα

Διπλωματική Εργασία

του

ΜΙΧΑΗΛ ΒΑΣΙΛΑΚΟΥ

Επιβλέπων: Συμεών Παπαβασιλείου

Καθηγητής Ε.Μ.Π.

Αθήνα, Μάιος 2024

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

Βελτιστοποίηση Μέσω Βαθιάς Ενισχυτικής Μάθησης για Ασφαλή Εκφόρτωση Δεδομένων σε Ασύρματα Δίκτυα

Διπλωματική Εργασία

του

ΜΙΧΑΗΛ ΒΑΣΙΛΑΚΟΥ

Επιβλέπων: Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 16η Μαΐου 2024.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Συμεών Παπαβασιλείου	Ιωάννα Ρουσσάκη	Ελένη Στάη
Καθηγητής Ε.Μ.Π.	Αναπλ. Καθηγήτρια Ε.Μ.Π.	Επίκ. Καθηγήτρια Ε.Μ.Π.

Αθήνα, Μάιος 2024

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

(Υπογραφή)

.....

Μιχαήλ Βασιλάκος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η ευρεία διάδοση κινητών συσκευών καθώς και άλλων ασυρμάτων συσκευών περιορισμένης υπολογιστικής ισχύος έχει φέρει και την απαίτηση για εκτέλεση υπολογιστικά απαιτητικών εφαρμογών σε αυτές. Σε αυτό το πρόβλημα, τα Mobile Edge Computing (MEC) και Rate Splitting Multiple Access (RSMA) είναι τεχνικές που πιθανώς να ικανοποιήσουν τις υψηλές απαιτήσεις των εφαρμογών αυτών χωρίς να απαιτείται η χρήση νέων, ισχυρότερων συσκευών. Με βάση αυτές τις τεχνικές προτείνουμε ένα σύστημα MEC που χρησιμοποιεί RSMA όπου κινητοί χρήστες (Mobile Users - MUs) μπορούν να εκφορτώσουν (offload) ολόκληρες τις εργασίες τους ή μέρος αυτών στον edge server για εκτέλεση. Για το σκοπό αυτό απαιτείται η κατάλληλη επιλογή παραμέτρων του συστήματος, όπως το ποσοστό της εργασίας που θα εκφορτωθεί (splitting ratio), η ισχύς εκπομπής (transmit power) των δεδομένων αυτών και η σειρά αποκωδικοποίησης (decoding order) των μηνυμάτων που τα περιέχουν, με στόχο την ελαχιστοποίηση της κατανάλωσης ενέργειας και του χρόνου απόκρισης του συστήματος. Το παραπάνω πρόβλημα βελτιστοποίησης μπορεί να εκφραστεί ως Διαδικασία Απόφασης Markov (Markov Decision Process - MDP) και έτσι να μεταφερθεί σε μοντέλο Ενισχυτικής Μάθησης (Reinforcement Learning) και να λυθεί με τη μέθοδο Deep Deterministic Policy Gradient για πολλαπλούς πράκτορες (MADDPG). Για να βελτιώσουμε την εξερεύνηση στο στάδιο της εκμάθησης κάναμε χρήση προστιθέμενου θορύβου στη λήψη αποφάσεων.

Λέξεις Κλειδιά

Ασφάλεια Φυσικού Επιπέδου, Βαθιά Ενισχυτική Μάθηση, Διαδικασία Απόφασης Markov, Διαδοχική Ακύρωση Παρεμβολών, Δράστης-Κριτής, Εκφόρτωση, Ενισχυτική Μάθηση, Ενισχυτική Μάθηση Πολλαπλών Πρακτόρων, Κατανομή Πόρων, Κινητή Υπολογιστική Άκρων, Πολλαπλή Πρόσβαση Διαίρεσης Ρυθμού, Σύστημα Πολλαπλών Πρακτόρων, Πρόβλημα Μικτών Ακεραίων, Σειρά Αποκωδικοποίησης, Συνεργατικές Παρεμβολές, Συνεργατικό Πρόβλημα, Υποκλοπέας, MADDPG

Abstract

The widespread use of mobile devices as well as other wireless devices with limited computing power has created the requirement for computationally demanding applications to be executed on them. In this problem, Mobile Edge Computing (MEC) and Rate Splitting Multiple Access (RSMA) are paradigms that are likely to meet the high demands of these applications without requiring the use of newer, more powerful devices. Based on these techniques, we propose a MEC system which uses RSMA where Mobile Users (MUs) can offload all or part of their tasks to the edge server for execution. This requires the appropriate selection of system parameters such as the splitting ratio of the offloaded task, the transmit power for offloading this data and the decoding order of the messages containing it, in order to minimize the energy consumption and the system response time. The above optimization problem can be expressed as a Markov Decision Process (MDP) and thus can be transferred to a Reinforcement Learning model and solved by the Multi Agent Deep Deterministic Policy Gradient (MADDPG) method. To improve the exploration in the learning stage we made use of additive noise in the decision making.

Keywords

Physical Layer Security, Deep Reinforcement Learning, Markov Decision Process, Successive Interference Cancellation, Actor-Critic, Offloading, Reinforcement Learning, Multi Agent Reinforcement Learning, Resource Allocation, Mobile Edge Computing, Rate Splitting Multiple Access, Multi Agent System, Mixed Integer Problem, Decoding Order, Cooperative Jamming, Cooperative Problem, Eavesdropper, MADDPG

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ.Συμεών Παπαβασιλείου, για την ευκαιρία που μου έδωσε να πραγματοποιήσω τη διπλωματική μου εργασία στο εργαστήριό του, καθώς και για την άριστη συνεργασία μας.
Ιδιαίτερες ευχαριστίες θέλω να δώσω και στη διδάκτορα Μαρία Διαμάντη για το ενδιαφέρον της και την καθοδήγησή της καθ' όλη τη διάρκεια της συγγραφής της εργασίας αυτής.
Στη συνέχεια, θα ήθελα να ευχαριστήσω μέσα από την καρδιά μου τους φίλους και συμφοιτητές μου Παναγιώτη, Παντελή και Κάλλια, για τις υπέροχες αναμνήσεις αλλά και την υποστήριξη όλα αυτά τα χρόνια.
Κυρίως όμως, θέλω να ευχαριστήσω την οικογένειά μου, τους γονείς μου, Δημήτρη και Κανέλλα, τα αδέρφια μου, Γεωργία και Χρήστο, αλλά και τους Μιχάλη και Δημήτρη για την αγάπη και τη στήριξή τους.

Βασιλάκος Μιχαήλ
Μάιος 2024

Περιεχόμενα

Κατάλογος σχημάτων

1 Εισαγωγή

1.1 Πρόλογος

Οι τελευταίες εξελίξεις στο πεδίο των κινητών δικτύων, καθώς και ο ολοένα αυξανόμενος όγκος δεδομένων που οι κινητές συσκευές καλούνται να επεξεργαστούν έχουν οδηγήσει στην ανάπτυξη νέων επαναστατικών τεχνολογιών επικοινωνίας και επεξεργασίας δεδομένων.

Τα καινούρια κινητά δίκτυα χρειάζεται να εξυπηρετήσουν ένα τεράστιο πλήθος χρηστών, το οποίο δημιουργεί την ανάγκη βέλτιστης αξιοποίησης των πόρων του δικτύου. Οι τεχνολογίες Ορθογωνικής Πολλαπλής Πρόσβασης (OMA) που χρησιμοποιούνται σε παλαιότερες γενιές δικτύων έχουν ήδη αρχίσει να παραχωρούν τη θέση τους σε τεχνολογίες NOMA (Μη Ορθογωνικής Πολλαπλής Πρόσβασης) στα δίκτυα πέμπτης γενιάς (5G), ενώ ήδη έχουν προταθεί νέες τεχνολογίες, όπως η Πολλαπλή Πρόσβαση Διαίρεσης Ρυθμού (RSMA) η οποία υπόσχεται βελτιωμένες επιδόσεις και χαμηλότερο κόστος υλοποίησης στους δέκτες [29].

Οι νέες τεχνολογίες δικτύων έχουν δώσει ώθηση στην ανάπτυξη μέχρι πρότεινος αδύνατων εφαρμογών, όπως εφαρμογές IoT σε τομείς από τη μετεωρολογία και τη γεωργία μέχρι έξυπνα σπίτια (smart homes), supply chain management ή και εφαρμογές Augmented Reality (AR) και Virtual Reality (VR) για κινητές συσκευές [2] [59]. Οι εφαρμογές αυτές έχουν υψηλές υπολογιστικές απαιτήσεις από τις συσκευές στις οποίες εκτελούνται, ενώ παράλληλα είναι ενεργειακά δαπανηρές. Αυτό φτάνει τις κινητές συσκευές στα όριά τους, αφού κατά κανόνα έχουν περιορισμένους πόρους, τόσο υπολογιστικούς όσο και ενεργειακούς.

Για να αντιμετωπιστούν οι παραπάνω περιορισμοί, έχουν προταθεί παραδείγματα "νεφοποίησης" των κινητών συσκευών (mobile cloudification), αυτό όμως εισάγει μία σημαντική καθυστέρηση (latency) στις εφαρμογές που εκτελούνται στο cloud. Ένα άλλο παράδειγμα που έχει προταθεί είναι το mobile edge computing (MEC). Το MEC ενισχύει το cloud computing φέρνοντας υπηρεσίες cloud στην "άκρη" (edge) του δικτύου, δηλαδή κοντά στις συσκευές, ώστε να είναι διαθέσιμοι ισχυροί υπολογιστικοί πόροι στους χρήστες με χαμηλό latency [1]. Οι κινητοί χρήστες μπορούν έτσι να εκφορτώσουν μέρος της εργασίας τους στους κοντινούς MEC servers για να γίνει σε εκείνους η επεξεργασία αντί αυτό να γίνεται σε μακρινούς cloud servers. Με αυτόν τον τρόπο μειώνεται η χρονική καθυστέρηση που εισάγουν οι cloud servers.

Η χρήση MEC εισάγει κάποιες προκλήσεις στον σχεδιασμό τους. Για παράδειγμα, κάποιες

εργασίες που εκτελούνται σε αυτά μπορεί να έχουν αυστηρές χρονικές απαιτήσεις. Επιπλέον παρουσιάζεται το πρόβλημα της κατανομής πόρων, αφού η εκφόρτωση δεδομένων στους edge servers είναι ενεργειακά ακριβή. Τέλος, τα κινητά δίκτυα είναι εξ ορισμού δυναμικά και το περιβάλλον τους (π.χ. κατάσταση καναλιού) είναι μεταβαλλόμενο, με συνέπεια η πολιτική του MEC να πρέπει να ανταποκριθεί σε αυτές τις αλλαγές. Παράλληλα, η ασύρματη εκφόρτωση δεδομένων δίνει τη δυνατότητα σε κακόβουλους χρήστες να κρυφακούν (eavesdropper) τα κανάλια επικοινωνίας για να υποκλέψουν δεδομένα από τους χρήστες του MEC. Αυτό μπορεί να περιοριστεί στα σχήματα μη ορθογωνικής παράλληλης πρόσβασης, όπως NOMA και RSMA με χρήση τεχνικών cooperative jamming, κατά τις οποίες ο ρυθμός αποκωδικοποίησης του eavesdropper είναι μικρότερος του ρυθμού αποκωδικοποίησης στον σταθμό βάσης - edge server, το οποίο επιτρέπει την ασφαλή μεταφορά δεδομένων [54].

Από την περιγραφή του προβλήματος προκύπτει ένα μη κυρτό πρόβλημα βελτιστοποίησης [2]. Οι μεταβαλλόμενες συνθήκες του προβλήματος, η ανάγκη για προσαρμογή της λύσης στις μεταβολές του περιβάλλοντος και η δυσκολία επίλυσής του έχουν οδηγήσει στην υιοθέτηση τεχνικών βαθιάς ενισχυτικής μάθησης.

1.2 Βιβλιογραφική Επισκόπηση

Οι τεχνολογίες πίσω από τα ασύρματα κυψελωτά δίκτυα νέας γενιάς έχουν μελετηθεί εκτενώς τα τελευταία χρόνια εξαιτίας της ραγδαίας αύξησης των κινητών δικτύων. Οι τεχνικές OMA [20] και SDMA [5] έχουν αρχίσει να υποχωρούν έναντι των NOMA [9, 18, 20] και RSMA [29, 36, 44] στα δίκτυα τελευταίας γενιάς. Στην εργασία αυτή θα χρησιμοποιηθεί κυρίως η τεχνική RSMA, ενώ θα χρησιμοποιηθεί και η NOMA για να συγκριθούν οι επιδόσεις τους. Η μοντελοποίηση των ασυρμάτων καναλιών επικοινωνίας θα γίνει υιοθετώντας το block fading model [12], ενώ ο παράγοντας Rayleigh fading που αναπαριστά τις χρονικά μεταβαλλόμενες ιδιότητες του καναλιού σε σύντομα χρονικά διαστήματα γίνεται με το Jake's model [10].

Το πρόβλημα βελτιστοποίησης στο οποίο ανάγεται το πρόβλημα κατανομής πόρων σε MEC μοντελοποιείται εύκολα με Markov Decision Process (MDP). Η επίλυση MDP με χρήση Reinforcement Learning (RL) αναλύεται στο [50], ενώ γενικότερες έννοιες της RL στο [41]. Η διατριβή [45] συγκρίνει τη συνεργατική με την ανεξάρτητη λειτουργία πρακτόρων στην RL πολλών πρακτόρων. Οι εξελίξεις στον τομέα της τεχνητής νοημοσύνης έχουν καταστήσει δυνατή τη χρήση βαθιάς μηχανικής μάθησης στην ενισχυτική μάθηση και την ανάπτυξη της βαθιάς ενισχυτικής μάθησης [4, 23, 35, 60]. Για τη λύση συστημάτων όπου πολλοί, ανεξάρτητοι μεταξύ τους χρήστες παίρνουν αποφάσεις οι οποίες επηρεάζουν το κοινό τους περιβάλλον έχει αναπτυχθεί η θεωρία της ενισχυτικής μάθησης πολλαπλών πρακτόρων (MARL) [14, 33]. Στο [8] αναπτύσσεται μία actor-critic προσέγγιση για συνεργασία πολλών πρακτόρων και διαμοιρασμό παραμέτρων (parameter sharing) για τη μείωση της πολυπλοκότητας του συστήματος. Ως συνεργατικό ορίζεται ένα πρόβλημα στο οποίο οι χρήστες καλούνται να το λύσουν με τρόπο ώστε να έχουν το μέγιστο δυνατό όφελος χωρίς να ζημιώνουν σημαντικά την επίδοση των υπολοίπων χρηστών. Στο [28] ο κάθε χρήστης σε ένα συνεργατικό (cooperative) πρόβλημα μοντελοποιεί τη συμπεριφορά των υπόλοιπων χρηστών. Το [26] επεκτείνει αυτή τη προσέγγιση

για μικτά συνεργατικά/ανταγωνιστικά (cooperative/competitive) προβλήματα. Στο [42] αναλύονται οι διαφορές μεταξύ cooperative και competitive συστημάτων, δείχνοντας πως η μεταβολή του κινήτρου των πρακτόρων στο γνωστό παιχνίδι Pong συντελεί στην αντιμετώπιση του ίδιου προβλήματος και με τους δύο τρόπους. Στο ίδιο γίνεται σύγκριση αλγορίθμων με προσαρμοζόμενους πράκτορες (adaptive agents). Τέλος, στο [30] βρίσκεται ένα review της τεχνικής independent learners σε συνεργατικά προβλήματα.

Οι έρευνες [27, 1] πραγματεύονται την ανάγκη για χρήση mobile edge computing (MEC) για την βελτίωση των επιδόσεων των συσκευών των χρηστών σε σύγχρονα κινητά δίκτυα. Προβλήματα resource allocation σε MEC αντιμετωπίζονται χωρίς τη χρήση DRL στα [3, 58, 24]. Στα [58, 24] αντιμετωπίζεται ένα πρόβλημα παρόμοιο με το δικό μας, καθώς γίνεται αναζήτηση βέλτιστης τιμής για συνεχείς (ισχύς) όσο και διακριτές μεταβλητές (σειρά αποκωδικοποίησης), πρόκειται δηλαδή για mixed integer problems.

Στη βιβλιογραφία είναι αρκετά διαδεδομένη η χρήση βαθιάς ενισχυτικής μάθησης (DRL) για την επίλυση του προβλήματος κατανομής πόρων και της εκφόρτωσης εργασιών σε MEC [47, 17, 43, 51]. Στο [2] γίνεται εκφόρτωση εργασιών σε IoT περιβάλλον, ενώ στο [39] χρησιμοποιείται hierarchical DRL για την επίλυση του mixed integer problem όπου το συνεχές και το διακριτό μέρος του προβλήματος λύνονται από διαφορετικά δίκτυα και αυτά συνδυάζονται για να βρεθεί η βέλτιστη λύση.

Με την ανάπτυξη της θεωρίας για multi agent reinforcement learning έχει επεκταθεί η χρήση MARL για την επίλυση προβλημάτων σε MEC. Σε αυτά, κάθε mobile user εκπροσωπείται από έναν agent, οι οποίοι προσπαθούν συνεργατικά να βρουν μία βέλτιστη λύση. Στο paper [6] χρησιμοποιείται η πιο απλή επέκταση της θεωρίας της ενισχυτικής μάθησης σε multi user συστήματα όπου ο κάθε πράκτορας αγνοεί την ύπαρξη των άλλων πρακτόρων (independent learners). Στο [34] χρησιμοποιούνται αλγόριθμοι για διακριτά προβλήματα για resource management σε very high throughput satelite (VHTS) systems. Η εργασία [59] χρησιμοποιεί έναν συνδυασμό DRL και federated learning για αποκεντρωμένη (decentralized) επίλυση του προβλήματος. Τέλος, το [7] παρουσιάζει έναν αλγόριθμο MADRL για συνεχή προβλήματα σε NOMA.

Σε συνδυασμό με τα παραπάνω προβλήματα μερικές εργασίες αντιμετωπίζουν το πρόβλημα της ασφάλειας στο physical layer (physical layer security - PLS) για την αποτροπή της παρακολούθησης των μηνυμάτων μεταξύ mobile users και edge server με χρήση artificial jamming. Το [53] φτάνει σε μία αναλυτική λύση σε NOMA, ικανοποιώντας παράλληλα κάποιους περιορισμούς στον ρυθμό εκφόρτωσης. Όμοια, το [54] φτάνει σε μία λύση με χρήση προσεγγιστικών μεθόδων. Σε ένα λίγο διαφορετικό πρόβλημα, τα papers [56, 61] κάνουν χρήση DRL για να εξασφαλίσουν ασφαλή αποστολή δεδομένων μέσω beamforming ελέγχοντας συσκευές intelligent reflective surface (IRS). Στο [21] με χρήση independent learners γίνεται ασφαλής εκφόρτωση δεδομένων σε MEC σε περιβάλλον πολλών χρηστών, όπου οι χρήστες δεν είναι κινητές συσκευές που επικοινωνούμε με κάποιο multiple access πρωτόκολλο αλλά είναι οχήματα που επικοινωνούν μέσω vehicle to vehicle (V2V) πρωτοκόλλου.

Τέλος, βλέπουμε να χρησιμοποιούνται τεχνικές MADRL για την επίλυση διαφορετικών προβλημάτων, όπως ένα supply chain management system [15] ή ο έλεγχος παρεμβολών σε radio networks [16].

1.3 Σκοπός Διπλωματικής Εργασίας

Στην εργασία αυτή καλούμαστε να λύσουμε το πρόβλημα βελτιστοποίησης που προκύπτει σε ένα σύστημα mobile edge computing (MEC), το οποίο χρησιμοποιεί rate splitting multiple access (RSMA) για την ασύρματη επικοινωνία μεταξύ κινητών χρηστών (mobile users - MUs) και edge server. Η λύση πρέπει να αντιμετωπίζει τους χρονικούς περιορισμούς που θέτουμε στο σύστημα, καθώς και το θέμα της ασφάλειας κατά τη μεταφορά δεδομένων από τους χρήστες στον edge server. Ο κάθε χρήστης καλείται να μάθει τη βέλτιστη συμπεριφορά, ρυθμίζοντας το ποσοστό της εργασίας του που θα εκφορτώσει στον edge server, καθώς και την ισχύ του σήματος μέσω του οποίου το κατορθώνει αυτό. Παράλληλα, συνεργάζεται με τους υπόλοιπους χρήστες για να επιτύχουν συνολικά τις καλύτερες επιδόσεις. Η εκμάθηση της κατάλληλης πολιτικής θα γίνει με τη βοήθεια βαθιάς ενισχυτικής μάθησης (deep reinforcement learning - DRL) και συγκεκριμένα με τον αλγόριθμο MADDPG (multi agent deep deterministic policy gradient).

1.4 Διάρθρωση Διπλωματικής Εργασίας

Η διπλωματική εργασία οργανώνεται ως εξής. Στο Κεφάλαιο 1 γίνεται μία σύντομη εισαγωγή στο πρόβλημα, αναλύεται η βιβλιογραφία που χρησιμοποιήθηκε και εξηγείται ο σκοπός της εργασίας. Το Κεφάλαιο 2 αναλύει σύντομα τις απαραίτητες έννοιες για την κατανόηση του προβλήματος και της μεθόδου επίλυσής του. Στο Κεφάλαιο 3 γίνεται η μοντελοποίηση του συστήματος και η διατύπωση του προβλήματος που καλούμαστε να λύσουμε. Στο Κεφάλαιο 4 περιγράφεται ο τρόπος επίλυσης του προβλήματος που διατυπώσαμε στο προηγούμενο κεφάλαιο. Το Κεφάλαιο 5 περιέχει τα αποτελέσματα των πειραμάτων που πραγματοποιήσαμε και αξιολογείται η λύση μέσα από συγκρίσεις με άλλα συστήματα. Τέλος, στο Κεφάλαιο 6 γίνεται μία σύνοψη των αποτελεσμάτων και εξάγονται συμπεράσματα από αυτά, καθώς και προτείνονται κατευθύνσεις για μελλοντική δουλειά ως συνέχεια αυτής της εργασίας.

2

Θεωρητικό Υπόβαθρο

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο αναλύονται έννοιες σχετικές με τις τεχνικές πολλαπλής πρόσβασης και πιο συγκεκριμένα με τις τεχνικές Rate Splitting Multiple Access (RSMA) και Non Orthogonal Multiple Access (NOMA). Στη συνέχειαγίνεται μία εισαγωγή στις Διαδικασίες Απόφασης Markov (Markov Decision Process - MDP). Θα αναλύσουμε κάποιες σημαντικές έννοιες γύρω από την Ενισχυτική Μάθηση (Reinforcement Learning - DRL), τη Βαθιά Ενισχυτική Μάθηση (Deep Reinforcement Learning - DRL) και την Ενισχυτική Μάθηση Πολλών Πρακτόρων (Multi-Agent Reinforcement Learning - MARL). Τέλος, θα περιγράψουμε τον αλγόριθμο DRL που θα χρησιμοποιηθεί (Multi-Agent Deep Deterministic Policy Gradient - MADDPG) καθώς και τον τρόπο λειτουργίας του.

2.2 Τεχνικές Πολλαπλής Πρόσβασης (MA)

Η εξέλιξη των τεχνικών πολλαπλής πρόσβασης είναι συνυφασμένη με την εξέλιξη των ασύρματων δικτύων, από τις πρώτες μέρες των αναλογικών συστημάτων μέχρι τα σύγχρονα πολύπλοκα ψηφιακά δίκτυα. Η ευρεία υιοθέτηση των ασύρματων δικτύων και ο ραγδαία αυξανόμενος αριθμός χρηστών έχουν φτάσει στα όρια την αξιοποίηση του διαθέσιμου φάσματος και προς αυτό έχουν αναπτυχθεί δίαφορες τεχνικές πολλαπλής πρόσβασης. Οι τεχνικες MA (Multiple Access) μπορούν να χωριστούν σε δύο βασικές προσεγγίσεις, την Ορθογωνική Πολλαπλή Πρόσβαση (Orthogonal Multiple Access - OMA) και τη Μη Ορθογωνική Πολλαπλή Πρόσβαση (Non Orthogonal Multiple Access - NOMA).

2.2.1 Ορθογωνική Πολλαπλή Πρόσβαση (OMA)

Η βασική αρχή της ορθογωνιότητας είναι ο διαμοιρασμός ορθογώνιων μεταξύ τους πόρων στους χρήστες με σκοπό την αποφυγή παρεμβολών στην επικοινωνία τους. Ένα ορθογωνικό σύστημα επιτρέπει σε έναν τέλειο δέκτη να διαχωρίζει απόλυτα τα επιθυμητά σήματα από τα ανεπιθύμητα. Αυτό συνεπάγεται πως εξασφαλίζεται η ορθογωνιότητα των σημάτων των διαφο-

ρετικών χρηστών. Μερικές από αυτές τις τεχνικές είναι η Frequency Division Multiple Access (FDMA), η Time Division Multiple Access (TDMA) και η Code Division Multiple Access [20]. Στο TDMA το ίδιο κανάλι συχνότητας μοιράζεται στους χρήστες στο πεδίο του χρόνου. Οι χρήστες επικοινωνούν διαδοχικά, κάνοντας χρήση του χρονικού διαστήματους (time slot) που τους διατέθηκε. Στο FDMA το διαθέσιμο φάσμα χωρίζεται σε ανεξάρτητα και ορθογώνια μεταξύ τους μέρη και στον κάθε χρήστη δίνεται από ένα. Έτσι, ο κάθε χρήστης μπορεί να χρησιμοποιεί το κανάλι του για επικοινωνία χωρίς να πρέπει να το μοιραστεί με κανέναν άλλο και αποφεύγοντας έτσι τις παρεμβολές. Στο CDMA ένα πλήθος χρηστών μπορεί να στέλνει μηνύματα πάνω στο ίδιο κανάλι επικοινωνίας. Στους χρήστες αντιστοιχίζονται ορθογώνιοι μεταξύ τους κώδικες με τους οποίους μπορούν να μοιράζονται τους ίδιους πόρους συχνότητας-χρόνου. Οι ορθογωνικές τεχνικές εξασφαλίζουν την αποφυγή παρεμβολών στα επιθυμητά σήματα χωρίς τη χρήση δαπανηρών και πολύπλοκων εξοπλισμών.

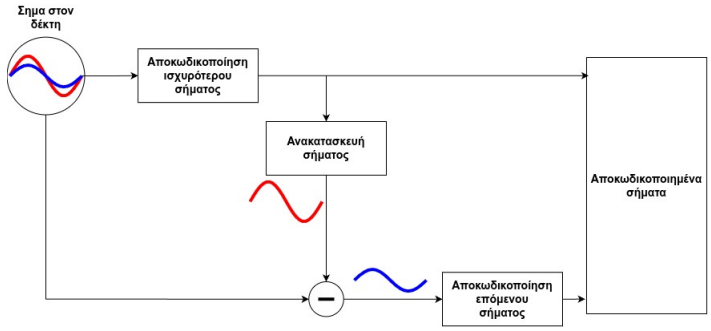
2.2.2 Πολλαπλή Πρόσβαση Διαίρεσης Χώρου (SDMA)

Η ιδέα πίσω από την SDMA είναι εμπνευσμένη από το CDMA. Στα 4G και 5G δίκτυα η ανάγκη για ασύρματη χωρητικότητα και η σπανιότητα των φασματικών πόρων έδωσε ώθηση στην υιοθέτηση επικοινωνιών multiple-input multiple-output (MIMO), στις οποίες το κάθε σημείο πρόσβασης (access point/base station - BS) διαθέτει πολλαπλές κεραίες. Τα δίκτυα MIMO δημιουργούν μια χωρική συνιστώσα στα συστήματα επικοινωνιών, το οποίο επιτρέπει την ανάπτυξη της τεχνικής space division multiple access (SDMA) [5]. Με την κατάλληλη αξιοποίηση των χωρικών πόρων και των πολλαπλών κεραιών μπορούν να εξυπηρετηθούν πολλαπλοί χρήστες στους ίδιους πόρους συχνότητας-χρόνου.

2.2.3 Μη Ορθογωνική Πολλαπλή Πρόσβαση (NOMA)

Το μειονέκτημα της OMA είναι πως το πλήθος των χρηστών που μπορούν να εξυπηρετηθούν περιορίζεται από το πλήθος των πόρων και το πόσο αυτοί μπορούν να διαιρεθούν. Αντίθετα με τις παραδοσιακές τεχνικές OMA, η Μη Ορθογωνική Πολλαπλή Πρόσβαση (Non Orthogonal Multiple Access - NOMA) επιτρέπει τον ταυτόχρονο διαμοιρασμό ενός καναλιού συχνότητας σε πολλαπλούς χρήστες εντός του ίδιου κελιού. Αυτό μας δίνει διάφορα πλεονεκτήματα, όπως την αύξηση του throughput μεταξύ κελιού και server, βελτιωμένο spectral efficiency, μεγαλύτερο πλήθος υποστηριζόμενων χρηστών και χαμηλότερο transmission latency καθώς δεν απαιτούνται scheduling requests από τους users στο base station [18].

Οι τεχνικές NOMA μπορούν να χωριστούν σε δύο βασικές κατηγορίες: power domain multiplexing και code domain multiplexing [9]. Το πρώτο σημαίνει πως σε διαφορετικούς χρήστες ανατίθενται διαφορετικές στάθμες ισχύος ανάλογα με τις συνθήκες του καναλιού τους, ώστε να επιτευχθεί βέλτιστη λειτουργία του συστήματος. Η ανάθεση ισχύος αυτή είναι χρήσιμη για το διαχωρισμό των διαφόρων χρηστών μέσω της τεχνικής Successive Interference Cancellation (SIC) [38] που χρησιμοποιείται για την αφαίρεση των παρεμβολών σε συστήματα πολλαπλών χρηστών. Στο code domain multiplexing ανατίθενται διαφορετικοί κώδικες (codes) σε διαφορετικούς χρήστες πάνω στους ίδιους πόρους χρόνου-συχνότητας.



Σχήμα 2.1: Τρόπος λειτουργίας SIC: διαδοχική αποκωδικοποίηση με σειρά ισχύος των σημάτων

Στην παρούσα εργασία θα ασχοληθούμε αποκλειστικά με το uplink, την αποστολή μηνυμάτων δηλαδή από τους χρήστες στον Base Station (BS), τον κεντρικό σταθμό του κελιού. Ο κάθε χρήστης στέλνει στον BS το σήμα του ταυτόχρονα με όλους τους υπόλοιπους. Ο BS αποκωδικοποιεί διαδοχικά το σήμα από κάθε χρήστη με την τεχνική SIC, όπως φαίνεται και στην εικόνα 2.1. Αποκωδικοποιεί λοιπόν το κάθε σήμα θεωρώντας ως παρεμβολή όλα τα υπόλοιπα σήματα και αφαιρεί κάθε αποκωδικοποιημένο σήμα από το σύνολο τον σημάτων που δέχεται. Έτσι, με κάθε σήμα που αφαιρείται μπορεί με μεγαλύτερη ευκολία να αποκωδικοποιήσει το επόμενο.

Για την ορθή διαδοχική ακύρωση των παρεμβολών απαιτούνται πολύπλοκοι δέκτες. Για κάθε χρήστη που περιέχεται στην ομάδα είναι απαραίτητο ένα παραπάνω επίπεδο στον δέκτη. Αυτό σημαίνει πως το κόστος του δέκτη αυξάνεται με την αύξηση των χρηστών που εξυπηρετούνται καθώς και πως σε περίπτωση αλλαγής των απαιτήσεων του συστήματος απαιτείται αλλαγή του δέκτη.

2.2.4 Uplink NOMA

Στην εργασία αυτή θα γίνει χρήση του σχήματος NOMA για την εκφόρτωση δεδομένων προς τον edge server από τους mobile users. Αυτό θα γίνει για να συγκριθεί η επίδοση της τεχνικής RSMA με μια πιο ευρέως διαδεδομένη τεχνική μη ορθογωνικής πρόσβασης.

Έστω K χρήστες και ένας base station (BS). Το σήμα που εκπέμπει ένας χρήστης k προς τον BS είναι:

$$x_k = \sqrt{P_k} s_k, \forall k \in K \tag{2.1}$$

όπου $x_k \in \mathbb{C}$ το σήμα, P_k η ισχύς εκπομπής και s_k η ροή δεδομένων (data stream) προς εκφόρτωση. Μετά την υπέρθεση των σημάτων που εκπέμπονται, ο σταθμός βάσης (base station - BS) λαμβάνει το σήμα [7]:

$$y = \sum_{k=1}^{K} \sqrt{G_k} x_k + n = \sum_{k=1}^{K} \sqrt{G_k P_k} s_k + n \tag{2.2}$$

όπου $G_k \in \mathbb{C}$ είναι το κέρδος καναλιού (channel gain) μεταξύ του χρήστη k και του server (BS) και $n \sim \mathcal{CN}(0, \sigma^2)$ είναι ο Additive White Gaussian Noise (AWGN).

Χωρίς βλάβη της γενικότητας μπορούμε να θεωρήσουμε πως τα μηνύματα των χρηστών αποκωδικοποιούνται με τη σειρά από τον τελευταίο χρήστη έως τον πρώτο, δηλαδή πρώτα

αποκωδικοποιείται η ροή δεδομένων του χρήστη $K-1$, έπειτα εκείνη του $K-2$ και τελευταία εκείνη του 0. Ο ρυθμός μετάδοσης δεδομένων από έναν χρήστη n προς τον BS που επιτυγχάνεται με χρήση του σχήματος NOMA στο uplink δίνεται από την παρακάτω σχέση [11]:

$$r_n = Blog_2\left(1+\frac{G_nP_n}{\sum_{i=1}^{n-1}G_iP_i+\sigma^2B}\right)\ [bps] \tag{2.3}$$

όπου B είναι το bandwidth που έχει στη διάθεσή του το σχήμα, G_i είναι το channel gain του χρήστη i προς τον BS, P_i είναι η ισχύς που χρησιμοποιεί ο χρήστης i για την εκπομπή της ροής δεδομένων του και σ^2 είναι η φασματική πυκνότητα ισχύος του AWGN μηδενικού μέσου.

2.2.5 Πολλαπλή Πρόσβαση Διαίρεσης Ρυθμού (RSMA)

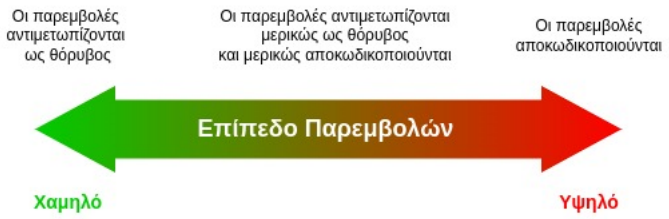
Το RSMA πρόκειται για μια τεχνική που γεφυρώνει τις δύο ακραίες στρατηγικές αντιμετώπισης παρεμβολών που είδαμε στα SDMA και NOMA. Η τεχνική αυτή αποτελεί γενίκευση τεσσάρων σχημάτων πολλαπλής πρόσβασης, των OMA, SDMA, physical layer multicasting και NOMA [29]. Η ιδέα πίσω από το RSMA είναι η διαίρεση των μηνυμάτων σε κοινά (common) και ιδιωτικά (private) μέρη και η δυνατότητα οι παρεμβολές μερικώς να αποκωδικοποιούνται και μερικώς να αντιμετωπίζονται ως θόρυβος, μια γενικότερη και λιγότερο αυστηρή προσέγγιση από όσες έχουμε δει μέχρι τώρα. Το RSMA είναι ευέλικτο στο επίπεδο των παρεμβολών και συμπεριφέρεται ως SDMA ή NOMA σε περίπτωση πολύ αδύναμων ή πολύ ισχυρών παρεμβολών, ρυθμίζοντας τις ισχύς και τα περιεχόμενα των μηνυμάτων του.

Για να εξηγήσουμε τη λειτουργία του RSMA θα θεωρήσουμε έναν BS με M κεραίες ο οποίος εξυπηρετεί 2 χρήστες. Στην περίπτωση του downlink ο πομπός χωρίζει το μήνυμά του W_k σε ένα κοινό μέρος $W_{c,k}$ και ένα ιδιωτικό μέρος $W_{p,k}$ και ενώνει τα $W_{c,1}$ και $W_{c,2}$ σε ένα μήνυμα W_c. Τα μηνύματα W_c, $W_{p,1}$ και $W_{p,2}$ κωδικοποιούνται ανεξάρτητα και προκωδικοποιούνται γραμμικά στον πομπό. Έπειτα, ο κάθε χρήστης αποκωδικοποιεί το κοινό stream s_c μεταχειρίζοντας τα ιδιωτικά streams ως θόρυβο. Το κοινό μήνυμα αφαιρείται από το εισερχόμενο σήμα και το ιδιωτικό stream που προορίζεται για εκείνον αποκωδικοποιείται, με το private stream των άλλων χρηστών να θεωρείται θόρυβος.

Το RSMA έρχεται ως απάντηση στο όραμα για τα σχήματα πολλαπλής πρόσβασης επόμενης γενιάς, στο οποίο τα σχήματα αυτά προσαρμόζονται στο επίπεδο παρεμβολών. Σε χαμηλά επίπεδα παρεμβολών αυτές αντιμετωπίζονται ως θόρυβος και σε υψηλά επίπεδα αποκωδικοποιούνται πλήρως, ενώ σε μεσαία επίπεδα αυτές αντιμετωπίζονται μερικώς ως θόρυβος και μερικώς αποκωδικοποιούνται, όπως φαίνεται και στο Σχήμα 2.2. Ένα από τα σημαντικότερα πλεονεκτήματα του RSMA έναντι του NOMA είναι η χαμηλότερη πολυπλοκότητα του δέκτη και συνεπώς το χαμηλότερο κόστος και η πιο εύκολη κλιμάκωσή του [29].

2.2.6 Uplink RSMA

Στην περίπτωση του uplink με την οποία θα ασχοληθούμε εμείς παρατηρείται καλύτερο throughput συγκριτικά με σχήματα OMA ή σχήματα που βασίζονται αποκλειστικά σε SIC, όπως το NOMA [44]. Στο uplink RSMA ο χρήστης k χωρίζει το μήνυμά του W_k σε δύο μέρη, $W_{k,1}$ και



Σχήμα 2.2: Το όραμα για τα σχήματα MA: καθώς το επίπεδο των παρεμβολών αλλάζει μεταβάλλεται και η συμπεριφορά του

$W_{k,2}$ και στη συνέχεια ο BS τα αποκωδικοποιεί με χρήση του SIC [36]. Το κάθε μήνυμα γίνεται encode στα streams $s_{k,1}$ και $s_{k,2}$ με τον χρήστη k να αναθέτει ισχύ $P_{k,1}$ και $P_{k,2}$ αντίστοιχα. Ας θεωρήσουμε ϕ μία διάταξη όλων των μηνυμάτων των edge users βάση της οποίας γίνεται η αποκωδικοποίησή τους από τον BS. Η σειρά του μηνύματος $s_{k,l}$ είναι $\phi_{k,l}$ και έτσι αν $\phi_{i,j} < \phi_{k,l}$, τότε το μήνυμα $s_{k,l}$ αποκωδικοποιείται μετά από το $s_{i,j}$.

Ο χρήστης k τότε εκπέμπει το σήμα [57]

$$x_k = \sqrt{P_{k,1}} s_{k,1} + \sqrt{P_{k,2}} s_{k,2}, \forall k \in K \tag{2.4}$$

όπου $x_k \in \mathbb{C}$. Το σήμα που λαμβάνει ο BS είναι [13]

$$y = \sum_{k=1}^{K} \sqrt{G_k} x_k + n = \sum_{k=1}^{K} \sum_{j=1}^{2} \sqrt{G_k P_{k,j}} s_{k,j} + n \tag{2.5}$$

όπου $G_k \in \mathbb{C}$ είναι το κέρδος καναλιού (channel gain) μεταξύ του χρήστη k και του BS και $n \sim \mathcal{CN}(0, \sigma^2)$ είναι ο Additive White Gaussian Noise (AWGN).

Από αυτά προκύπτει πως ο ρυθμός αποκωδικοποίησης του stream $s_{k,j}$ που μπορεί να επιτευχθεί είναι:

$$r_{k,j} = B log_2 \left(1 + \frac{G_k P_{k,j}}{\sum_{(l \in \mathcal{K}, m \in \mathcal{J}) | \phi_{l,m} > \phi_{k,j}} G_l P_{l,m} + \sigma^2 B} \right) \ [bps] \tag{2.6}$$

όπου B το bandwidth του BS, σ^2 το power spectral density του γκαουσιανού θορύβου, G_k το channel gain του edge user k προς τον BS, $\mathcal{K}$ το σύνολο των χρηστών και $\mathcal{J}$ το σύνολο των streams του κάθε χρήστη. Το σύνολο $(l \in \mathcal{K}, m \in \mathcal{J}) | \phi_{l,m} > \phi_{k,j}$ αντιπροσωπεύει τα μηνύματα που αποκωδικοποιούνται μετά από το μήνυμα $s_{k,j}$. [57]

Όπως αναφέραμε πιο πάνω, ο κάθε χρήστης k έχει δύο data streams, τα $s_{k,1}$ και $s_{k,2}$. Έτσι προκύπτει ο συνολικός ρυθμός αποκωδικοποίησης του χρήστη:

$$r_k = \sum_{j=1}^{2} r_{k,j} \tag{2.7}$$

2.2.7 Κέρδος Καναλιού (Channel Gain)

Σχετικά με το κέρδος καναλιού, στην εργασία αυτή υιοθετούμε το μοντέλο block fading [12] ως εξής:

$$G_k = |h_k|^2 \beta_k, G_k \in \mathbb{R} \tag{2.8}$$

όπου $\beta_k \in \mathbb{R}$ είναι ο παράγοντας μεγάλης κλίμακας του block fading ο οποίος μένει ίδιος για πολλαπλές χρονικές στιγμές και $h_k \in \mathbb{C}$ είναι ο παράγοντας μικρής κλίμακας που αναπαριστά το Rayleigh fading. Ο παράγοντας μικρής κλίμακας εκφράζει τις χρονικά μεταβαλλόμενες ιδιότητες του καναλιού και τον υπολογίζουμε κάνοντας χρήση του Jake's model [10] με τον παρακάτω τρόπο:

$$h_k^{(t)} = \rho h_k^{(t-1)} + \sqrt{1-\rho^2}\zeta_k^{(t)} \tag{2.9}$$

όπου $\zeta_k^{(t)} \sim \mathcal{CN}(0,1)$. Το ρ είναι η παράμετρος συσχέτισης και υπολογίζεται ως:

$$\rho = J_0(2\pi f_d T) \tag{2.10}$$

όπου J_0 είναι η μηδενικής τάξης συνάρτηση Bessel, f_d η μέγιστη συχνότητα Doppler και T ο χρόνος μεταξύ των χρονικών στιγμών που υπολογίζουμε το $h_k^{(t)}$. Η τιμή για την πρώτη χρονική στιγμή ακολουθεί την κατανομή $h_k^{(0)} \sim \mathcal{CN}(0,1)$.

2.3 Διαδικασία Απόφασης Markov (Markov Decision Process)

Τα Markov Decision Processes (MDP) είναι μία κεντρική και αναπόσπαστη έννοια για την ενισχυτική μάθηση [50]. Ένα MDP είναι μία στοχαστική διαδικασία διακριτού χρόνου, η οποία μας παρέχει ένα μαθηματικό υπόβαθρο για τη μοντελοποίηση προβλημάτων στα οποία το αποτέλεσμα είναι μερικώς τυχαίο και μερικώς επηρεάζεται από κάποια απόφαση που παίρνει εκείνος ο οποίος καλείται να το λύσει (decision maker). Τα MDP είναι εξαιρετικά χρήσιμα για τη μελέτη προβλημάτων βελτιστοποίησης.

Σε κάθε βήμα, η διαδικασία (process) βρίσκεται σε μία κατάσταση (state) και ο decision maker μπορεί να επιλέξει κάποια από τις ενέργειες που του είναι διαθέσιμες στη συγκεκριμένη κατάσταση. Η διαδικασία στο επόμενο βήμα αντιδρά μεταβαίνοντας τυχαία σε κάποια κατάσταση με πιθανότητα που επηρεάζεται από την ενέργεια που επιλέχθηκε. Έτσι η κατάσταση στο επόμενο βήμα εξαρτάται από την κατάσταση στο τρέχον βήμα και την ενέργεια που θα επιλεχθεί, ενώ είναι ανεξάρτητη από όλες τις προηγούμενες καταστάσεις - ικανοποιεί δηλαδή την ιδιότητα Markov. Τα MDP αποτελούνται από states, actions, transitions μεταξύ states και ένα reward function.

2.3.1 Καταστάσεις (States)

Το σύνολο των καταστάσεων του προβλήματος $\mathcal{S}$ ορίζεται ως το πεπερασμένο σύνολο $\{s^1, ..., s^N\}$, όπου το μέγεθος του χώρου καταστάσεων (state space) είναι N. Μία κατάσταση είναι μία μοναδική περιγραφή όλων των σημαντικών πληροφοριών για τη μοντελοποίηση του συστήματος κάποια δεδομένη χρονική στιγμή.

2.3.2 Ενέργειες (Actions)

Το σύνολων ενεργειών $\mathcal{A}$ ορίζεται ως το πεπερασμένο σύνολο $\{a^1, ..., a^K\}$, όπου το μέγεθος του χώρου δράσης (action space) είναι K. Οι ενέργειες μπορούν να χρησιμοποιηθούν για

τον έλεγχο της κατάστασης του συστήματος. Το σύνολο των ενεργειών που μπορούν να εκτελεστούν σε μία κατάσταση $s \in \mathcal{S}$ συμβολίζεται ως $\mathcal{A}(s)$, όπου $\mathcal{A}(s) \subseteq \mathcal{A}$. Για το πρόβλημά μας μπορούμε να θεωρήσουμε $\mathcal{A}(s) = \mathcal{A},\ \forall s \in \mathcal{S}$.

2.3.3 Συνάρτηση Μετάβασης (Transition Function)

Εφαρμόζοντας ένα action $a \in \mathcal{A}$ σε κάποιο state $s \in \mathcal{S}$, το σύστημα μεταβαίνει από την s σε μία νέα κατάσταση $s' \in \mathcal{S}$, η οποία βασίζεται σε μία πιθανοτική κατανομή πάνω στο σύνολο όλων των δυνατών μεταβάσεων. Η συνάρτηση μετάβασης $\mathcal{T}$ ορίζεται ως $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$. Όπως είναι γνωστό από τη θεωρία πιθανοτήτων, $0 \leq \mathcal{T}(s, a, s') \leq 1, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}, \forall s' \in \mathcal{S}$ και $\sum_{s' \in \mathcal{S}} \mathcal{T}(s, a, s') = 1, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$. Για να διατάξουμε τις καταστάσεις ανάλογα με τη σειρά εμφάνισης των καταστάσεων μπορούμε να ορίσουμε ένα διακριτό ρολόι $t = 1, 2, ...$ έτσι ώστε η κατάσταση s_{t+1} να είναι η επόμενη της s_t, αφού η πρώτη είναι η κατάσταση του συστήματος τη χρονική στιγμή $t + 1$ και η δεύτερη τη χρονική στιγμή t.

Το σύστημα ονομάζεται μαρκοβιανό (Markovian) αν το αποτέλεσμα μίας ενέργειας δεν εξαρτάται από τις προηγούμενες καταστάσεις στις οποίες βρέθηκε το σύστημα αλλά μόνο στην τρέχουσα κατάσταση, δηλαδή:

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, ...) = P(s_{t+1}|s_t, a_t) = \mathcal{T}(s_t, a_t, s_{t+1}) \tag{2.11}$$

Αυτό μας εξασφαλίζει πως οι πληροφορίες που έχουμε σε κάθε κατάσταση αρκούν για να λάβουμε τη βέλτιστη απόφαση σε αυτή.

2.3.4 Συνάρτηση Ανταμοιβής (Reward Function)

Η συνάρτηση ανταμοιβής εκφράζει την αξία του να βρίσκεται το σύστημα σε ένα state ή να πραγματοποιεί μία ενέργεια. Η συνάρτηση ανταμοιβής που θα χρησιμοποιήσουμε στο πρόβλημά μας ορίζεται ως $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, χωρίς όμως να είναι ο μοναδικός ορισμός reward function που υπάρχει. Η συγκεκριμένη συνάρτηση δίνει ανταμοιβή (reward) για την εκτέλεση κάποιου action σε κάποιο state. Το reward function είναι σημαντικό στα MDP γιατί δηλώνει έμμεσα τον σκοπό της εκπαίδευσης, δηλαδή τη μεγιστοποίηση του reward.

2.3.5 Markov Decision Process

Συνδυάζοντας τα παραπάνω στοιχεία έχουμε τον ορισμό ενός MDP, τα οποία είναι η βάση της μεθόδου επίλυσης πολλών προβλημάτων βελτιστοποίησης με RL.

*Ένα **Markov Decision Process** είναι μία πλειάδα (tuple)* $< \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} >$ *όπου* $\mathcal{S}$ *ένα πεπερασμένο σύνολο καταστάσεων,* $\mathcal{A}$ *ένα πεπερασμένο σύνολο ενεργειών,* $\mathcal{T}$ *μία συνάρτηση μετάβασης που ορίζεται ως* $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ *και* $\mathcal{R}$ *μία συνάρτηση ανταμοιβής που ορίζεται ως* $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

2.4 Ενισχυτική Μάθηση

Η Ενισχυτική Μάθηση (Reinforcement Learning - RL) συνδυάζει τους τομείς της τεχνητής νοημοσύνης και της λήψης αποφάσεων, κατασκευάζοντας ευφυείς πράκτορες οι οποίοι μαθαίνουν μέσω της εμπειρίας. Σε αντίθεση με την επιβλεπόμενη μάθηση (supervised learning), όπου τα μοντέλα εκπαιδεύονται σε επισημασμένα σύνολα δεδομένων, ή τη μη επιβλεπόμενη μάθηση, όπου προσπαθούμε να βρούμε δομές και σχέσεις μέσα σε μη επισημασμένα δεδομένα, η RL επικεντρώνεται στην εκμάθηση βέλτιστων συμπεριφορών μέσω αλληλεπιδράσεων με το περιβάλλον.

Η βάση της ενισχυτικής μάθησης είναι ένας πράκτορας και το περιβάλλον μέσα στο οποίο δρα αυτός. Ο πράκτορας λαμβάνει αποφάσεις παρατηρώντας το περιβάλλον, ενώ εκείνο ανταποκρίνεται σε αυτές τις ενέργειες παρέχοντας ανατροφοδότηση (feedback) με τη μορφή ανταμοιβών (reward) ή ποινών (penalty). Μέσω αυτής της επαναληπτικής διαδικασίας εξερεύνησης (exploration) και εκμετάλλευσης (exploitation), οι αλγόριθμοι ενισχυτικής μάθησης επιτρέπουν στους πράκτορες να μαθαίνουν αποτελεσματικές στρατηγικές για τη μεγιστοποίηση της ανταμοιβής τους.

2.4.1 Model Free / Model Based

Σε ένα σύστημα ενισχυτικής μάθησης μπορούν να υιοθετηθούν δύο προσεγγίσεις: model-free ή model-based. Η model-based προσέγγιση βασίζεται κυρίως στο σχεδιασμό και η model-free στην εκμάθηση. Η model-based RL επιδιώκει την εκμάθηση ενός μοντέλου της δυναμικής του περιβάλλοντος, των πιθανοτήτων μετάβασης από το ένα state στο άλλο και των συναρτήσεων ανταμοιβής. Το μοντέλο αυτό ύστερα χρησιμοποιείται από τον πράκτορα για τη λήψη της βέλτιστης απόφασης. Αυτού του είδους η προσέγγιση μπορεί να έχει καλύτερα αποτελέσματα και μεγαλύτερη ανθεκτικότητα στον θόρυβο, καθώς και να κάνει αποδοτικότερη χρήση δεδομένων. Στη model-free RL ο πράκτορας μαθαίνει απευθείας μέσω της αλληλεπίδρασης με το περιβάλλον χωρίς να μοντελοποιεί τη δυναμική του περιβάλλοντος. Αντίθετα, εστιάζει στην εκμάθηση μίας πολιτικής (policy) ή μίας συνάρτησης τιμής (value function) για τη λήψη αποφάσεων. Η προσέγγιση αυτή χρησιμοποιείται κυρίως σε πιο περίπλοκα συστήματα όπου η δυναμική του περιβάλλοντος είναι δύσκολο να μοντελοποιηθεί, καθώς είναι πιο ευέλικτη και προσαρμόζεται εύκολα σε διάφορα περιβάλλοντα [41].

2.4.2 Value Based / Policy Based / Actor-Critic

Η **value based** ενισχυτική μάθηση είναι μία προσέγγιση στην οποία ένας πράκτορας μαθαίνει να κάνει σωστή λήψη αποφάσεων εκτιμώντας την αξία των διάφορων καταστάσεων στις οποίες βρίσκεται ή/και των διαθέσιμων σε εκείνον ενεργειών. Βασική ιδέα της value based RL είναι η συνάρτηση αξίας (value function) η οποία προβλέπει την αναμενόμενη απόδοση (expected return) ή τη συσσωρευτική απόδοση (cumulative return) δεδομένης της κατάστασης (state) του συστήματος και της ενέργειας που είναι διαθέσιμη στον πράκτορα σε αυτή. Η συνάρτηση αυτή παίρνει συνήθως τη μία από τις δύο παρακάτω μορφές. Η πρώτη μορφή είναι εκείνη της συνάρ-

τησης αξίας κατάστασης (state value function), η οποία συνήθως συμβολίζεται με $V^{\pi}(s)$, όπου ως s συμβολίζεται η κατάσταση του συστήματος. Η συνάρτηση αυτή αξιολογεί την κατάσταση στην οποία βρίσκεται το σύστημα ακολουθώντας κάποια πολιτική (policy) π. Το πρόβλημα με αυτή τη συνάρτηση είναι πως συνήθως δεν γνωρίζουμε τη βέλτιστη πολιτική π^*, ούτε είναι σταθερή η μηχανική μετάβασης από τη μία κατάσταση στην άλλη ώστε να βρούμε τον τρόπο να οδηγηθούμε στη βέλτιστη κατάσταση. Η δεύτερη μορφή αντιμετωπίζει το παραπάνω πρόβλημα. Η μορφή που παίρνει η συνάρτηση αξίας είναι η συνάρτηση αξίας κατάστασης-ενέργειας (state-action value function), που συμβολίζεται με $Q^{\pi}(s, a)$. Με s συμβολίζεται ξανά το state και με a μία ενέργεια διαθέσιμη στον πράκτορα στο state αυτό. Η συνάρτηση εκφράζει την αξία της κατάστασης δεδομένης της ενέργειας a και έτσι είναι εφικτή η εύρεση της βέλτιστης πολιτικής ακολουθώντας πάντα την ενέργεια για την οποία η συνάρτηση παίρνει τη μεγαλύτερη τιμή. Ο πράκτορας αλληλεπιδρώντας με το περιβάλλον βελτιστοποιεί τις εκτιμήσεις των συναρτήσεων αυτών [4]. Μερικοί value based αλγόριθμοι είναι ο Q-Learning [52] και ο DQL [32].

Στη **policy based** ενισχυτική μάθηση ο πράκτορας μαθαίνει μία πολιτική (policy) απευθείας από το περιβάλλον χωρίς να κατασκευάζει συναρτήσεις αξίας. Η πολιτική είναι μια αντιστοιχία states και actions, όπου ο πράκτορας ακολουθεί κάθε φορά το action που αντιστοιχεί στο state στο οποίο βρίσκεται. Αντίθετα με την προσέγγιση της value based ενισχυτικής μάθησης, ο πράκτορας επικεντρώνεται στη μεγιστοποίηση του cumulative reward. Αυτό επιτυγχάνεται με τεχνικές όπως το gradient ascent, το οποίο προσαρμόζει την πολιτική επιζητώντας το μακροπρόθεσμο κέρδος. Μερικοί policy based αλγόριθμοι είναι ο REINFORCE [48] και ο PPO [37]. Οι policy based μέθοδοι επιτρέπουν την αντιμετώπιση προβλημάτων σε συνεχείς χώρους. [4]

Μία πολύ δημοφιλής μέθοδος είναι η συγχώνευση των δύο παραπάνω προσεγγίσεων. Η ιδέα αυτή γέννησε τη μέθοδο **actor critic**, η οποία χρησιμοποιεί τις μεθόδους των value based αλγορίθμων για να βρει μία συνάρτηση τιμής και τις μεθόδους των policy based αλγορίθμων για να βρει τη συνάρτηση πολιτικής (policy function). Η μέθοδος actor critic αποτελεί βελτίωση της policy based μεθόδου καθώς αυξάνει το sample efficiency και επιταχύνει την εκπαίδευση αλλά και της value based μεθόδου αφού επιτρέπει τη χρήση τους σε συνεχή περιβάλλοντα. Μαζί με τα πλεονεκτήματα των δύο άλλων μεθόδων όμως κληρονομεί και αρκετά από τα προβλήματά τους, όπως το overestimation του critic ή την ανεπαρκή εξερεύνηση του actor [60]. Μερικοί αλγόριθμοι που βασίζονται στη μέθοδο actor critic είναι ο DDPG [25] και ο A3C [31].

2.4.3 On Policy / Off Policy

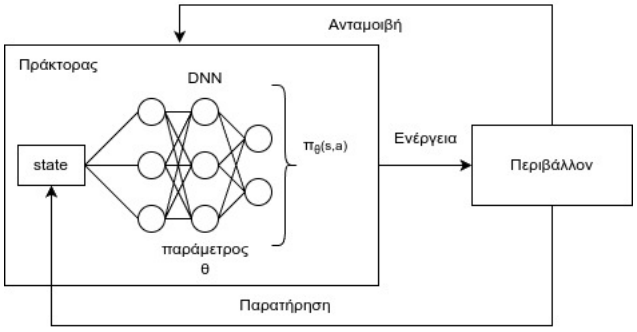
Μία από τις σημαντικότερες αποφάσεις για ένα σύστημα ενισχυτικής μάθησης είναι η επιλογή δειγμάτων που θα δοθούν σε αυτό για εκπαίδευση. Ως προς αυτό υπάρχουν δύο μέθοδοι, on-policy και off-policy. Ο πράκτορας αλληλεπιδρά με το περιβάλλον κάθε φορά που λαμβάνει μία απόφαση και εκτελεί κάποια ενέργεια και στη συνέχεια βελτιώνει τη συμπεριφορά του χρησιμοποιώντας τα δεδομένα που έχει συλλέξει από τις αλληλεπιδράσεις του. Βλέπουμε λοιπόν πως προκύπτουν δύο ξεχωριστές διαδικασίες, η συλλογή δεδομένων από τις αλληλεπιδράσεις του πράκτορα με το περιβάλλον και η εκπαίδευσή του πάνω σε αυτά τα δεδομένα. Στους αλγορίθμους που χρησιμοποιούν τη μέθοδο on-policy η εκπαίδευση του πράκτορα γίνεται πάνω

στα δεδομένα που μόλις εκείνος συνέλεξε. Αντίθετα, η off-policy μεθοδος δίνει μεγαλύτερη ευελιξία στα δεδομένα που μπορούν να χρησιμοποιηθούν για την εκπαίδευση του πράκτορα, αφού μπορούν να χρησιμοποιηθούν δείγματα από παρελθοντικές αλληλεπιδράσεις του χρήστη με το περιβάλλον. Το πλεονέκτημα της off-policy μεθόδου είναι η εντατικότερη εξερεύνηση η οποία δίνει τη δυνατότητα στον πράκτορα να δοκιμάσει περισσότερες ενέργειες σε κάθε κατάσταση. Οι off policy αλγόριθμοι ενθαρρύνουν τους πράκτορες να εξερευνήσουν το περιβάλλον πληρέστερα [14].

2.5 Βαθιά Ενισχυτική Μάθηση

Τα πρόσφατα επιτεύγματα της βαθιάς μάθησης (deep learning), η οποία επωφελήθηκε από τα big data, τη ραγδαία αύξηση της υπολογιστικής ισχύος, νέες αλγοριθμικές τεχνικές και αρχιτεκτονικές και πακέτα λογισμικού έδωσαν την ώθηση για την υιοθέτησή τους σε πλήθος επιστημονικών πεδίων, ανάμεσά τους και η ενισχυτική μάθηση.

Η βαθιά μάθηση χρησιμοποιείται στην RL συνήθως για την προσέγγιση συναρτήσεων (π.χ. value function). Στη βαθιά μάθηση κατασκευάζονται νευρωνικά δίκτυα (neural networks) που αποτελούνται από πολλαπλά επίπεδα. Το κάθε επίπεδο συνδέεται με τους νευρώνες του προηγούμενου, ενώ η ισχύς της σύνδεσής τους αλλάζει όσο αυτό εκπαιδεύεται ώστε να προσεγγίζει μία επιθυμητή συνάρτηση (function approximation). Τα νευρωνικά που χρησιμοποιούνται έχουν την ικανότητα να μαθαίνουν από "raw" δεδομένα, δηλαδή δεδομένα που έχουν υποστεί ελάχιστη επεξεργασία, αντιμετωπίζοντας έτσι το λεγόμενο "curse of dimensionality" σύμφωνα με το οποίο γραμμική αύξηση της διαστατικότητας του προβλήματος οδηγεί σε εκθετικά πολυπλοκότερα συστήματα [23].



Σχήμα 2.3: Βαθιά Ενισχυτική Μάθηση

2.6 Ενισχυτική Μάθηση Πολλαπλών Πρακτόρων

Η ενισχυτική μάθηση πολλαπλών πρακτόρων (Multi-Agent RL - MARL) συνδυάζει τα συστήματα πολλαπλών πρακτόρων με την RL, βρίσκεται έτσι στην τομή των πεδίων της τεχνητής νοημοσύνης με το πεδίο της θεωρίας παιγνίων [23].

Συνήθως η RL θεωρεί μονούς πράκτορες (single-agent) σε στάσιμα (stationary) περιβάλλοντα. Σε αντίθεση με αυτό, η MARL θεωρεί πολλαπλούς χρήστες οι οποίοι μαθαίνουν μέσω της ενισχυτικής μάθησης σε συχνά μη στάσιμα (non stationary) περιβάλλοντα λόγω της επιρροής των άλλων πρακτόρων στη δυναμική του περιβάλλοντος. Για την αντιμετώπιση αυτού του προβλήματος έχουν προταθεί λύσεις όπως η προσθήκη καναλιών επικοινωνίας μεταξύ των πρακτόρων [4].

2.6.1 Συγκεντρωποίηση - Αποκέντρωση

Στην αναζήτηση τεχνικών για την αντιμετώπιση του προβλήματος της αστάθειας του περιβάλλοντος ερχόμαστε αντιμέτωποι με μία από τις σημαντικότερες σχεδιαστικές επιλογές που καλούμαστε να κάνουμε στα MARL, αυτή της **συγκεντρωποίησης (centralization)** και της **αποκέντρωσης (decentralization)**.

Η πρώτη προσέγγιση είναι η **fully decentralized** με ένα χαρακτηριστικό της παράδειγμα την τεχνική independent learning (IL). Πρόκειται για την πιο απλή επέκταση της single-agent RL σε multi-agent προβλήματα, στην οποία ο κάθε πράκτορας βελτιστοποιεί το policy του ανεξάρτητα από τους υπόλοιπους, αγνοώντας το πρόβλημα της αστάθειας και της συνεργασίας χωρίς όμως αυτό να σημαίνει πως δεν επιτυγχάνουν ικανοποιητικά αποτελέσματα.

Η δεύτερη προσέγγιση είναι η **fully centralized**. Σε αυτή θεωρούμε την ύπαρξη μίας κεντρικής μονάδας η οποία συγκεντρώνει πληροφορίες για όλους τους πράκτορες. Αυτή η τεχνική αντιμετωπίζει μερικώς τα προβλήματα της αστάθειας και της μερικής παρατηρησιμότητας του περιβάλλοντος από τους πράκτορες, όμως απαιτεί σημαντικούς πόρους και χρόνο καθώς και κάνει εφικτή την υποκλοπή των δεδομένων των χρηστών από κακόβουλους χρήστες.

Η τελευταία προσέγγιση είναι η **centralized training and decentralized execution**. Σε αυτή, μία κεντρική μονάδα συλλέγει πληροφορίες για τους πράκτορες κατά τη διάρκεια της εκπαίδευσής τους, όμως οι πολιτικές που μαθαίνουν αποκεντρώνονται και χρησιμοποιούνται για τοπική εκτέλεση από τους πράκτορες με τις τοπικές τους πληροφορίες. Αυτό επιτρέπει τη μερική αντιμετώπιση του προβλήματος της αστάθειας χωρίς σημαντική επιβάρυνση στην υπολογιστική ισχύ που απαιτείται ή στο χρόνο κατά την εκτέλεση, όπως και στην ασφάλεια του συστήματος. [14]

2.7 DDPG

Ο αλγόριθμος DDPG (Deep Deterministic Policy Gradient) [25] είναι ένας actor-critic, model-free αλγόριθμος βασισμένος στο ντετερμινιστικό policy gradient που μπορεί να λειτουργήσει σε συνεχή action spaces. Βασίζεται στον αλγόριθμο Deep Q-Learning [32] και μπορεί να βρει policies των οποίων η επίδοση είναι συγκρίσιμη με planning αλγορίθμους που έχουν πλήρη πρόσβαση στη δυναμική του περιβάλλοντος.

2.7.1 Θεωρητική Θεμελίωση

Θεωρούμε ένα σύνηθες σύστημα RL με έναν πράκτορα ο οποίος αλληλεπιδρά με ένα περιβάλλον $\mathcal{E}$ σε διακριτές χρονικές στιγμές (timesteps). Στο timestep t ο πράκτορας λαμβάνει την παρατήρηση x_t, εκτελεί την ενέργεια a_t και παίρνει την επιβράβευση r_t. Οι ενέργειες είναι πραγματικοί αριθμοί και μπορούν να ανήκουν σε ένα συνεχές action space. Θεωρούμε πως το περιβάλλον είναι πλήρως παρατηρήσιμο, δηλαδή $s_t = x_t$.

Η συμπεριφορά του πράκτορα καθορίζεται από την πολιτική π που ακολουθεί, η οποία αντιστοιχεί σε μία κατανομή πιθανότητας $\pi : \mathcal{S} \rightarrow \mathcal{P}(\mathcal{A})$. Το περιβάλλον $\mathcal{E}$ μπορεί να είναι επίσης στοχαστικό. Το μοντελοποιούμε ως ένα MDP με state space $\mathcal{S}$, action space $\mathcal{A} = \mathbb{R}^N$, μία αρχική κατανομή καταστάσεων $p(s_1)$ και δυναμικές μετάβασης $p(s_{t+1}|s_t, a_t)$ και reward function $r(s_t, a_t)$.

Η τιμή που επιστρέφεται από κάποια κατάσταση είναι $\mathcal{R}_t = \sum_{i=t}^{T} \gamma^{(i-t)} r(s_i, a_i)$ με εκπτωτικό συντελεστή $\gamma \in [0, 1]$. Στόχος είναι η εκμάθηση ενός policy το οποίο να μεγιστοποιεί την αναμενόμενη επιστροφή από την αρχική κατανομή $J = \mathbb{E}_{r_i, s_i \sim E, a_i \sim \pi}[R_1]$. Ως ρ^π συμβολίζεται το discounted state visitation distribution για την πολιτική π.

Η συνάρτηση action-value που περιγράφει την αναμενόμενη επιστροφή στην κατάσταση s_t εκτελώντας την ενέργεια a_t ακολουθώντας την πολιτική π είναι:

$$Q^\pi(s_t, a_t) = \mathbb{E}_{r_{i \geq t}, s_{i>t} \sim E, a_{i>t} \sim \pi}[\mathcal{R}_t | s_t, a_t] \tag{2.12}$$

Κάνοντας χρήση της αναδρομικής σχέσης γνωστής ως Bellman equation:

$$Q^\pi(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}[r(s_t, a_t) + \gamma \mathbb{E}_{a_{t+1} \sim \pi}[Q^\pi(s_{t+1}, a_{t+1})]] \tag{2.13}$$

Αν η πολιτική στόχος (target policy) είναι ντετερμινιστική μπορούμε να την περιγράψουμε ως μία συνάρτηση $\mu : \mathcal{S} \leftarrow \mathcal{A}$:

$$Q^\mu(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}[r(s_t, a_t) + \gamma Q^\mu(s_{t+1}, \mu(s_{t+1}))] \tag{2.14}$$

Αυτό σημαίνει πως είναι δυνατό να μάθουμε το Q^μ off-policy, με χρήση transitions από μία διαφορετική στοχαστική πολιτική β.

Θεωρούμε function approximators που χαρακτηρίζονται από το θ^Q, το οποίο βελτιστοποιούμε ελαχιστοποιώντας το loss:

$$L(\theta^Q) = \mathbb{E}_{s_t \sim \rho^\beta, a_t \sim \beta, r_t \sim E}[(Q(s_t, a_t|\theta^Q) - y_t)^2] \tag{2.15}$$

όπου

$$y_t = r(s_t, a_t) + \gamma Q(s_{t+1}, \mu(s_{t+1})|\theta^Q) \tag{2.16}$$

2.7.2 Αλγόριθμος

Λόγω της δυσκολίας εφαρμογής του Q-Learning σε συνεχή action spaces, ο αλγόριθμος αυτός κάνει χρήση της προσέγγισης actor-critic βασισμένης στον αλγόριθμο DPG [40]. Ο αλγόριθμος DPG αξιοποιεί μια παραμετροποιημένη συνάρτηση actor $\mu(s|\theta^\mu)$ η οποία καθορίζει

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode=1,M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t=1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t and new state s_{t+1}
    Store transition $(s_t, a_t, r_t, s_{t+1}$ in R
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i + \gamma Q' \left(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}\right)$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i \left(y_i - Q\left(s_i, a_i|\theta^Q\right)\right)^2$
    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i} \tag{2.17}$$

    Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \tag{2.18}$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'} \tag{2.19}$$

  end for
end for

την πολιτική με μία ντετερμινιστική αντιστοίχιση καταστάσεων σε συγκεκριμένες ενέργειες. Ο critic $Q(s,a)$ μαθαίνεται με την εξίσωση Bellman όπως στο Q-Learning. Ο actor ενημερώνεται με εφαρμογή του νόμου αλυσίδας (chain rule) στην αναμενόμενη επιστροφή της αρχικής κατανομής J:

$$\begin{aligned}\nabla_{\theta^\mu} J &\approx \mathbb{E}_{s_t\sim\rho^\beta}\left[\nabla_{\theta^\mu} Q(s,a|\theta^Q)|_{s=s_t,a=\mu(s_t|\theta^\mu)}\right] \\ &= \mathbb{E}_{s_t\sim\rho^\beta}\left[\nabla_a Q(s,a|\theta^Q)|_{s=s_t,a=\mu(s_t)}\nabla_{\theta_\mu}\mu(s|\theta^\mu)|_{s=s_t}\right]\end{aligned} \tag{2.20}$$

Αποδεικνύεται πως αυτό είναι το *policy gradient* [40], η κλίση της επίδοσης του policy.

Όπως και στο Q-Learning, η χρήση μη γραμμικών function approximators είναι αναγκαία για μεγάλα state spaces, όμως αυτό κάνει τη σύγκλιση μη εγγυημένη. Ο αλγόριθμος αυτός χρησιμοποιεί νευρωνικά δίκτυα ως function approximators. Η χρήση νευρωνικών δικτύων προϋποθέτει ανεξάρτητα και όμοια κατανεμημένα δείγματα, το οποίο είναι πρόκληση δεδομένου πως η εξερεύνηση γίνεται σειριακά στο περιβάλλον. Έτσι, όπως και στον DQN, ο DDPG κάνει χρήση replay buffer. Πρόκειται για μία πεπερασμένη μνήμη (cache) $\mathcal{B}$. Οι μεταβάσεις που παίρνονται ως δείγματα από την εξερεύνηση στο περιβάλλον αποθηκεύονται στον replay buffer, με τα παλαιότερα δείγματα να αντικαθιστώνται πρώτα όταν αυτός γεμίσει (FIFO). Σε κάθε χρονική στιγμή, ο actor και ο critic εκπαιδεύονται σε ένα sample batch (συλλογή δειγμάτων) που εκλέγεται ομοιόμορφα από τον buffer.

Καθώς το νευρωνικό δίκτυο $Q(s,a|\theta^Q)$ που ενημερώνεται χρησιμοποιείται για τον καθορισμό της τιμής στόχου (target value) (2.16), η απευθείας εφαρμογή του Q-Learning (2.15) γίνεται ασταθής σε πολλά περιβάλλοντα και τείνει να αποκλίνει. Η λύση μοιάζει με τα target network του DQL [32], τροποποιημένο για actor-critic συστήματα και χρησιμοποιώντας "soft" target updates. Δημιουργούμε ένα αντίγραφο για τα actor και critic networks, $Q'(s,a|\theta^{Q'})$ και $\mu'(s|\theta^{\mu'})$ που χρησιμοποιούνται για να υπολογίζουμε τα target values. Τα βάρη των networks ενημερώνονται ακολουθώντας τα networks που εκπαιδεύονται με: $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$, με $\tau \ll 1$, ώστε τα target networks να αλλάζουν αργά, βελτιώνοντας τη σταθερότητα της εκμάθησης.

Ένα ακόμα από τα προβλήματα στα συνεχή action spaces είναι η εξερεύνηση. Ένα από τα πλεονεκτήματα των off-policy αλγορίθμων είναι πως μπορούν να χειριστούν το πρόβλημα αυτό ανεξάρτητα από τον αλγόριθμο εκπαίδευσης. Για τον λόγο αυτό γίνεται χρήση μιας πολιτικής εξερεύνησης μ', η οποία προσθέτει θόρυβο $\mathcal{N}$ στο actor policy, ώστε να εξερευνήσει καλύτερα τον χώρο δράσης. Η νέα πολιτική ορίζεται ως:

$$\mu'(s_t) = \mu(s_t|\theta^\mu_\tau) + \mathcal{N} \tag{2.21}$$

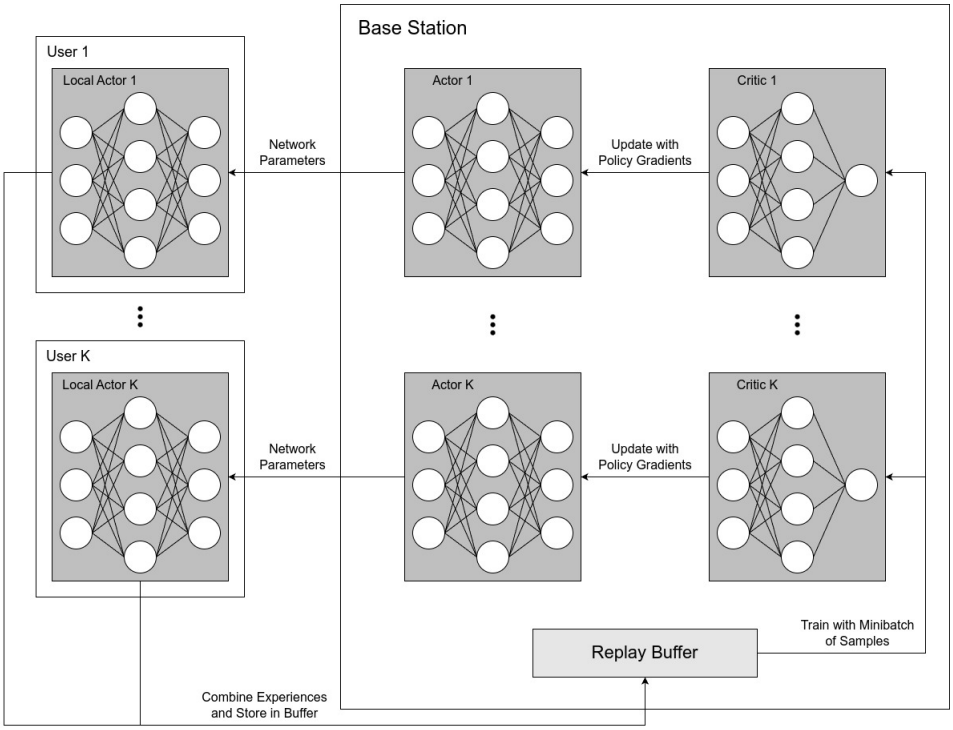
όπου το $\mathcal{N}$ επιλέγεται ώστε να ταιριάζει στο περιβάλλον.

2.8 MADDPG

Πολλά προβλήματα που καλούμαστε να αντιμετωπίσουμε με χρήση RL χαρακτηρίζονται από ένα πλήθος χρηστών που πρέπει να δουλέψουν συνεργατικά για την επίτευξη ενός βέλτιστου αποτελέσματος για όλους. Η ανάγκη αυτή οδήγησε στην ανάπτυξη της θεωρίας της ενισχυτικής μάθησης πολλών πρακτόρων (MARL). Ο αλγόριθμος MADDPG [7] αποτελεί επέκταση του DDPG [25] στα MARL.

2.8.1 Θεωρητική Θεμελίωση

Ο MADDPG ανήκει στην κατηγορία centralized training decentralized execution[33]. Σε αντίθεση με αλγορίθμους DDPG βασισμένους στην τεχνική independent learner, όπως για παράδειγμα στο [6], ο MADDPG κάνει χρήση ενός πλήρως παρατηρίσιμου critic για τον κάθε χρήστη της τοπολογίας, τοποθετημένο στο BS. Μαζεύοντας τις τοπικές παρατηρήσεις και ενέργειες όλων των χρηστών στον replay buffer μπορούμε να κάνουμε το περιβάλλον σταθερό (stationary) ανεξαρτήτως των policies των άλλων agents.



Σχήμα 2.4: Η αρχιτεκτονική του αλγορίθμου MADDPG: Κ χρήστες με έναν local actor ο καθένας, 1 Base Station με ένα ζευγάρι actor-critic για κάθε χρήστη και τον Replay Buffer

Η διαδικασία centralized training decentralized execution φαίνεται και στην εικόνα 2.4. Ο BS περιέχει K fully observable critics, έναν για κάθε έναν από τους χρήστες της τοπολογίας του προβλήματος, οι οποίοι θα χρησιμοποιηθούν για να εκπαιδευτούν οι K actors που βρίσκονται επίσης στον BS. Ο κάθε χρήστης έχει ένα τοπικό αντίγραφο μόνο για τον actor του, αφού είναι το μόνο που χρειάζεται για να παίρνει τις αποφάσεις. Το experience του, δηλαδή οι παρατηρήσεις του state και οι ενέργειες που εκτέλεσε σε κάθε timeslot t, δεν αποθηκεύονται τοπικά, αλλά αποστέλλονται στον BS για να συνενωθούν σε ένα δείγμα συνολικής εμπειρίας (global experience).

Οι critics εκπαιδεύονται σε δείγματα global experience για να έχουμε το επιθυμητό αποτέλεσμα του stationary περιβάλλοντος, ενώ οι actors παίρνουν αποφάσεις χρησιμοποιώντας τις περιορισμένες παρατηρήσεις που μπορεί να κάνει ο κάθε user μόνος του. Κατά την εκπαίδευση, κάθε χρονική στιγμή t ένα batch δειγμάτων global experience επιλέγεται τυχαία από τον replay

buffer $\mathcal{B}$ για την ενημέρωση του critic k ελαχιστοποιώντας το loss:

$$L_k = \mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{r},\mathbf{s}')\sim U(\mathcal{B})}\left[\left(y_k - Q(s,a|\theta^{Q_k})\right)^2\right] \tag{2.22}$$

όπου το target value για τον critic υπολογίζεται ως:

$$y_k = r_k + \gamma Q\left(\mathbf{s}', \mathbf{a}'|\theta^{Q_k}\right)|\mathbf{a}' = \{\mu(s'_j|\theta^{\mu_j})\}_{j=1}^{K} \tag{2.23}$$

Ως $(\mathbf{s}, \mathbf{a}, \mathbf{r}, \mathbf{s}')$ συμβολίζονται τα state, action, reward και next state των συνολικών δειγμάτων εμπειρίας, ενώ r_k είναι το reward του χρήστη k. Αυτό μπορεί να είναι ίδιο για όλους τους χρήστες και συνεπώς ίδιο με το $\mathbf{r}$, ή και ο κάθε χρήστης να έχει δικό του reward και μέσα από τον συνδυασμό όλων των reward να υπολογίζεται το $\mathbf{r}$. Στην παραπάνω εξίσωση, με $\mathbf{a}' = (a'_1, a'_2, ..., a'_K)$ συμβολίζεται η εκτιμώμενη ενέργεια για το επόμενο timeslot, η οποία υπολογίζεται με $a'_j = \mu(s'_j|\theta^{\mu_j})$, όπου $\mu(s'_j|\theta^{\mu_j})$ το actor function του χρήστη $j, \forall j \in \mathcal{K}$. Το actor function ενημερώνεται με το παρακάτω policy gradient:

$$\nabla_{\theta^{\mu_k}} J_k \approx \mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{r},\mathbf{s}')\sim U(\mathcal{B})}\left[\nabla_{a_k} Q\left(\mathbf{s}, \mathbf{a}|\theta^{Q_k}\right)|_{\mathbf{a}=\{\mu(s_j|\theta^{\mu_j})\}_{j=1}^{K}} \times \nabla_{\theta^{\mu_k}}\mu(s_k|\theta^{\mu_k})\right] \tag{2.24}$$

Για τη σωστή λειτουργία του αλγορίθμου απαιτείται η χρήση soft updating. Αυτό σημαίνει πως τα target networks των actor και critic ($\theta^{Q'_k}$ και $\theta^{\mu'_k}$ αντίστοιχα) θα ακολουθούν αργά τις συμπεριφορές των δικτύων των actor και critic που εκπαιδεύονται με χρήση των παραπάνω εξισώσεων.

Ο κάθε χρήστης k περιέχει έναν local actor, ο οποίος ενημερώνεται κάθε β χρονικές στιγμές από τον actor k του base station. Αυτό επιτρέπει τη μείωση του overhead της επικοινωνίας μεταξύ χρηστών και BS και επιτρέπει την ανεξάρτητη από τον BS και τους υπόλοιπους χρήστες λήψη αποφάσεων από τον k. Έτσι έχουμε αναπτύξει την τεχνική centralized training decentralized execution, η οποία είναι η βάση του αλγορίθμου.

2.8.2 Αλγόριθμος

Η πλήρης περιγραφή του αλγορίθμου σε ψευδογλώσσα γίνεται στο 2.

Algorithm 2 MADDPG algorithm

BS randomly initialized the actor network $\mu(s|\theta^{\mu_k})$, the critic network $Q(s,a|\theta^{Q_k})$ and the associated target networks with weights $\theta^{\mu'_k} \leftarrow \theta^{\mu_k}$ and $\theta^{Q'_k} \leftarrow \theta^{Q_k}, \forall k \in \mathcal{K}$;

BS initializes the experience replay buffer $\mathcal{B}$;

Randomly initialize the local actor network $\mu(s|\theta^{\mu''_k}), \forall k \in \mathcal{K}$;

for each episode $m = 1, 2, ..., M_{max}$ **do**

Reset simulation parameters for the NOMA-based MEC model environment;

Randomly generate an initial state s, with s_k being the observation of each user $k \in \mathcal{K}$;

for each timeslot $t = 1, 2, ..., T_{max}$ **do**

for each user $k \in \mathcal{K}$ **do**

Select an action $a_{k,t} = \mu(s_{k,t}|\theta^{\mu''_k}) + \Delta\mu$ by using the local policy network $\theta^{\mu''_k}$ and exploration noise $\Delta\mu$;

Execute the action $a_{k,t}$ independently at the user agent, observe the next state $s_{k,t+1}$ from the environment simulator and receive the reward $r_{k,t}$;

Send the tuple $(s_{k,t}, a_{k,t}, r_{k,t}, s_{k,t+1})$ to BS;

end for

BS combines the tuples from all users as $(\mathbf{s_t}, \mathbf{a_t}, \mathbf{r_t}, \mathbf{s_{t+1}})$ and stores it into $\mathcal{B}$;

Randomly sample a minibatch of I tuples $\{(\mathbf{s_i}, \mathbf{a_i}, \mathbf{r_i}, \mathbf{s'_i})\}_{i=1}^{I}$ from $\mathcal{B}$;

for each actor-critic network $k \in \mathcal{K}$ **do**

Update the critic network $Q(\mathbf{s}, \mathbf{a}|\theta^{Q_k})$ by minimizing the loss L_k:

$$L_k = \frac{1}{I}\sum_{i=1}^{I}\left[\left(y_{i,k} - Q(\mathbf{s_i}, \mathbf{a_i}|\theta^{Q_k})\right)^2\right], \tag{2.25}$$

where $y_{i,k} = r_{i,k} + \gamma Q\left(\mathbf{s'_i}, \mathbf{a'_i}|\theta^{Q'_k}\right)$ with setting $\mathbf{a'_i} = \{\mu(s'_{i,j}|\theta^{\mu_j})\}_{j=1}^{K}$;

Update the actor network $\mu s|\theta^{\mu_k}$ by using the following policy gradient:

$$\nabla_{\theta^{\mu_k}} J_k = \frac{1}{I}\sum_{i=1}^{I}\left[\nabla_{a_k} Q\left(\mathbf{s_i}, \hat{\mathbf{a}}_\mathbf{i}|\theta^{Q_k}\right)|_{\hat{\mathbf{a}}_\mathbf{i}=\{\mu(s_{i,j}|\theta^{\mu_j})\}_{j=1}^{K}} \nabla_{\theta^{\mu_k}}\mu(s_{i,k}|\theta^{\mu_k})\right] \tag{2.26}$$

end for

Update the target networks $\theta^{Q'_k} \leftarrow \tau\theta^{Q_k} + (1-\tau)\theta^{Q'_k}$ and $\theta^{\mu'_k} \leftarrow \tau\theta^{\mu_k} + (1-\tau)\theta^{\mu'_k}$;

Update the local actor network $\theta^{\mu''_k} \leftarrow \theta^{\mu_k}$ every β timeslots for each user $k \in \mathcal{K}$;

end for

end for

3

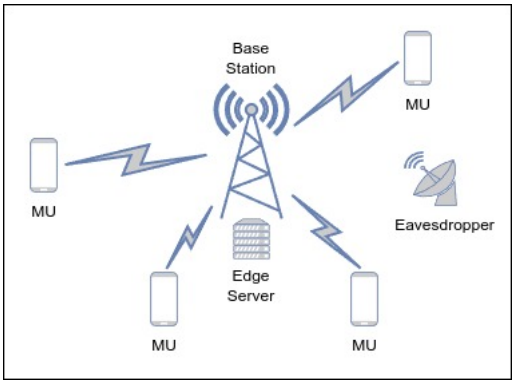
Μοντέλο Συστήματος - Διατύπωση Προβλήματος

Ορίζουμε ένα σύνολο $\mathcal{K} = \{1, 2, ..., K\}$ κινητών χρηστών (mobile users - MUs), 1 σταθμό βάσης (base station - BS) με 1 edge server (ES) και 1 κακόβουλο χρήστη (eavesdropper) και ορίζουμε χρονικά διαστήματα $t = \{1, ..., T\}$ διαρκείας Δ δευτερολέπτων. Σε κάθε χρονικό διάστημα t στον χρήστη k δημιουργείται μία εργασία (task) μεγέθους $S_k^{tot}[t]$ [CPU cycles]. Η διάρκεια των χρονικών διαστημάτων είναι σταθερή και ίση με Δ. Οι χρήστες έχουν τη δυνατότητα να εκφορτώσουν (offload) ένα μέρος της εργασίας τους στον server για εκτέλεση.

Για την επίλυση του συστήματος θα χρησιμοποιήσουμε την προσέγγιση πολλαπλών πρακτόρων. Ο κάθε κινητός χρήστης του συστήματος μπορεί να θεωρηθεί ως ένας πράκτορας, με τον κάθε πράκτορα να μπορεί να λάβει αποφάσεις ανεξάρτητα από τους υπόλοιπους. Χρησιμοποιούμε την τεχνική της κεντρικής εκπαίδευσης και αποκεντρωμένης εκτέλεσης, κατά την οποία οι πράκτορες λαμβάνουν αποφάσεις ανεξάρτητα με βάση τις τοπικές τους παρατηρήσεις, ενώ η εκπαίδευσή τους γίνεται κεντρικά, αφού συλλεχθούν και ενοποιηθούν όλες οι τοπικές παρατηρήσεις.

3.1 Μοντέλο Συστήματος

3.1.1 Δίκτυο



Σχήμα 3.1: Τοπολογία δικτύου

Οι MUs βρίσκονται σε ένα ασύρματο κινητό δίκτυο, μέσα στο οποίο επικοινωνούν με χρήση RSMA. Μέσω του δικτύου έχουν τη δυνατότητα να επικοινωνούν μεταξύ τους, καθώς και με τον σταθμό βάσης. Η μεταξύ τους επικοινωνία δεν θα μας απασχολήσει στην παρούσα εργασία. Οι χρήστες τοποθετούνται τυχαία σε ένα κελί, το οποίο ορίζουμε ως ένα ορθογώνιο παραλληλόγραμμο διαστάσεων $(X \times Y)$, στο κέντρο του οποίου βρίσκεται ο BS. Ο edge server τοποθετείται στον BS και θεωρούμε την επικοινωνία μεταξύ τους άμεση, ασφαλή και με μηδενική καθυστέρηση, ώστε να εστιάσουμε στην επικοινωνία μεταξύ των MUs και του BS. Ο eavesdropper τοποθετείται και αυτός τυχαία στον ίδιο χώρο που τοποθετούνται και οι MUs.

Σε κάθε χρονική στιγμή t, ο κάθε MU k αναλαμβάνει την εκτέλεση μίας εργασίας μεγέθους $S_k^{tot}[t]$, την οποία μπορεί να εκτελέσει τοπικά ή να εκφορτώσει στον edge server μέσω του BS. Θα θεωρήσουμε τη γενικότερη περίπτωση στην οποία ο χρήστης μπορεί να εκφορτώσει μέρος της εργασίας, συμβολίζοντας έτσι ως $o_k[t] \in [0, 1]$ το ποσοστό της εργασίας που θα εκφορτώσει. Έτσι, $o_k[t]$ είναι το ποσοστό του $S_k^{tot}[t]$ που κάνει offload ο χρήστης k στον ES και $(1 - o_k[t])$ το ποσοστό του $S_k^{tot}[t]$ που εκτελείται τοπικά.

3.1.2 Μοντέλο Επικοινωνίας

Για τη μοντελοποίηση του ασύρματου καναλιού που χρησιμοποιείται για την επικοινωνία του χρήστη k με τον BS κάνουμε χρήση του μοντέλου block fading [12]. Όπως είδαμε και στο θεωρητικό μέρος, το κέρδος του καναλιού τη χρονική στιγμή t δίνεται από τη σχέση:

$$G_k^{(t)} = |h_k^{(t)}|^2 \beta_k, G_k^{(t)} \in \mathbb{R} \tag{3.1}$$

όπου β_k ο παράγοντας μεγάλης κλίμακας block fading και $h_k^{(t)}$ ο παράγοντας μικρής κλίμακας Rayleigh fading. Το $h_k^{(t)}$ δείξαμε ότι υπολογίζεται με βάση το Jake's model [10] ως εξής:

$$h_k^{(t)} = \rho h_k^{(t-1)} + \sqrt{1 - \rho^2}\zeta_k^{(t)}, h_k^{(t)} \in \mathbb{C} \tag{3.2}$$

όπου $\zeta_k^{(t)} \sim \mathcal{CN}(0, 1)$ και $\rho = J_0(2\pi f_d T)$, με f_d τη μέγιστη συχνότητα Doppler και $T = \Delta$ την απόσταση μεταξύ δύο διαδοχικών χρονικών στιγμών. Ο παράγοντας block fading υπολογίζεται με βάση την απόσταση με χρήση του path loss model [13]:

$$PL_k = 128.1 + 37.6log_{10}(d) \pm n_k \text{ [dB]} \tag{3.3}$$

όπου $d = \sqrt{x_k^2 + y_k^2}$ η απόσταση του χρήστη k από τον BS σε km (θεωρώντας πως ο BS βρίσκεται στη θέση $(0, 0)$ και ο k στη θέση (x_k, y_k)) και $n_k \sim \mathcal{N}(0, \sigma_2)$ ο παράγοντας shadow fading [46, 13]. Το PL_k υπολογίζεται σε μονάδες dB. Το β_k υπολογίζεται από το PL_k ως:

$$\beta_k = \frac{1}{PL_k} \tag{3.4}$$

Για να το μετατρέψουμε από dB σε καθαρές μονάδες μπορούμε να χρησιμοποιήσουμε τη σχέση

$$\beta_k = 10^{-\frac{PL_k}{10}} \tag{3.5}$$

Όμοια υπολογίζεται και το κέρδος καναλιού μεταξύ του χρήστη k και του eavesdropper $G_{k,e}^{(t)}$.

Με $s_k[t]$ συμβολίζουμε το stream των δεδομένων που κάνει offload ο MU k τη χρονική στιγμή t. Τα δεδομένα αυτά χωρίζονται σε K υπομηνύματα. Στην εργασία αυτή θεωρούμε $K = 2$, οπότε τα μηνύματα που στέλνει ο k στον BS είναι τα $s_{k,1}[t]$ και $s_{k,2}[t]$. Το σήμα που εκπέμπει ο k είναι:

$$x_k^{BS}[t] = \sqrt{P_{k,1}[t]}s_{k,1}[t] + \sqrt{P_{k,2}[t]}s_{k,2}[t] \tag{3.6}$$

και έτσι το σήμα που λαμβάνει ο BS είναι:

$$y^{BS}[t] = \sum_{k=1}^{K} \sqrt{G_k[t]}x_k^{BS}[t] + n = \sum_{k=1}^{K}\sum_{j=1}^{2} \sqrt{G_k[t]P_{k,j}[t]}s_{k,j}[t] + n \tag{3.7}$$

όπου $x, y \in \mathbb{C}$ και $n \sim \mathcal{CN}(0, \sigma^2)$ είναι ο Additive White Gaussian Noise (AWGN). Το σήμα που φτάνει στην κεραία του eavesdropper είναι αντίστοιχα:

$$y^e[t] = \sum_{k=1}^{K} \sqrt{G_{k,e}[t]}x_k[t] + n = \sum_{k=1}^{K}\sum_{j=1}^{2} \sqrt{G_{k,e}[t]P_{k,j}[t]}s_{k,j}[t] + n \tag{3.8}$$

Κάθε χρήστης k έχει ένα μέγιστο όριο P_k^{max} [Watt] στην ισχύ εκπομπής του, τέτοιο ώστε $\sum_{j=1}^{2} P_{k,j}[t] \leq P_k^{max}$.

Για να αποκωδικοποιήσει ο BS τα μηνύματα όλων των χρηστών από το μήνυμα $y[t]$ που λαμβάνει πραγματοποιεί successive interference cancellation (SIC). Καθώς υπάρχουν $2|K|$ μηνύματα για αποκωδικοποίηση, αυτό σημαίνει πως υπάρχουν $2|K|!$ διαφορετικές διατάξεις στη σειρά των μηνυμάτων προς αποκωδικοποίηση. Η σειρά αποκωδικοποίησης ϕ είναι μία διάταξη όλων των μηνυμάτων των mobile users βάση της οποίας γίνεται η αποκωδικοποίησή τους από τον BS, τέτοια ώστε αν $\phi_{i,j} < \phi_{k,l}$, τότε το μήνυμα $s_{k,l}$ αποκωδικοποιείται μετά από το $s_{i,j}$.

Δεδομένης μιας σειράς ϕ ο ρυθμός αποκωδικοποίησης του μηνύματος $s_{k,j}$ δίνεται από την εξίσωση:

$$r_{k,j}^{BS} = Blog_2\left(1 + \frac{G_k P_{k,j}}{\sum_{(l\in\mathcal{K}, m\in\mathcal{J})|\phi_{l,m}>\phi_{k,j}} G_l P_{l,m} + \sigma^2 B}\right) \text{ [bps]} \tag{3.9}$$

Ο συνολικός ρυθμός αποκωδικοποίησης για τα μηνύματα του χρήστη k από τον BS είναι:

$$r_k^{BS} = \sum_{j=1}^{2} r_{k,j}^{BS} \text{ [bps]} \tag{3.10}$$

Στην εργασία αυτή θεωρούμε πως ο κακόβουλος χρήστης (eavesdropper) δεν έχει καμία πληροφορία για το ϕ και επομένως δεν είναι ικανός να πραγματοποιήσει SIC. Για τον λόγο αυτό, όταν ο eavesdropper αποκωδικοποιεί ένα μήνυμα $s_{k,j}$, αντιμετωπίζει τα υπόλοιπα $(2|K| - 1)$ μηνύματα ως παρεμβολές. Σαν αποτέλεσμα, ο ρυθμός αποκωδικοποίησης του μηνύματος $s_{k,j}$ του χρήστη k στον eavesdropper είναι:

$$r_{k,j}^{e} = Blog_2\left(1 + \frac{G_{k,e} P_{k,j}}{\sum_{(l\in\mathcal{K}, m\in\mathcal{J})|(l,m)\neq(k,j)} G_{l,e} P_{l,m} + \sigma^2 B}\right) \text{ [bps]} \tag{3.11}$$

Με βάση τις δύο αυτές σχέσεις για τους ρυθμούς αποκωδικοποίησης στον BS και τον eavesdropper μπορούμε να χρησιμοποιήσουμε το Wyner's secrecy encoding scheme [55] για να υπολογίσουμε

το secure data rate της αποκωδικοποίησης του μηνύματος $s_{k,j}$:

$$r_{k,j}^{sec} = [r_{k,j}^{BS} - r_{k,j}^{e}]^{+} \text{ [bps]} \tag{3.12}$$

όπου $[x]^{+} = max(0, x)$.

3.1.3 Μοντέλο Τοπικής Εκτέλεσης

Ο χρήστης k αναλαμβάνει τη χρονική στιγμή t να εκτελέσει μία εργασία μεγέθους $S_k^{tot}[t]$. Έχοντας εκφορτώσει ένα ποσοστό $o_k[t]$ της εργασίας αυτής, τα δεδομένα που καλείται να επεξεργαστεί τοπικά έχουν μέγεθος $(1 - o_k[t])S_k^{tot}[t]$ [CPU cycles]. Ο χρόνος που θα χρειαστεί ο MU k για την τοπική εκτέλεση του μέρους της εργασίας που του απομένει μετά την εκφόρτωση μπορεί να υπολογιστεί ως:

$$T_k^{exec}[t] = \frac{(1 - o_k[t])S_k^{tot}[t]}{f_k} \text{ [sec]} \tag{3.13}$$

όπου f_k η συχνότητα της CPU του χρήστη k. Η ενεργειακή κατανάλωση για την τοπική εκτέλεση υπολογίζεται ως εξής:

$$E_k^{exec}[t] = \rho f_k^2 (1 - o_k[t])S_k^{tot}[t] \text{ [Joule]} \tag{3.14}$$

όπου ρ ο συντελεστής αποτελεσματικής χωρητικότητας (effective capacitance coefficient) που εξαρτάται από τη συσκευή του χρήστη k.

3.1.4 Μοντέλο Εκφόρτωσης

Τη χρονική στιγμή t ο MU k κάνει offload $o_k[t]S_k^{tot}[t]$ [CPU cycles] δεδομένων. Θεωρούμε πως ο χρόνος που χρειάζεται για την επεξεργασία των δεδομένων στον edge server είναι μηδενικός, όπως και ο χρόνος που απαιτείται για την αποστολή του αποτελέσματος της επεξεργασίας στον MU, καθώς είναι πέρα από τα πλαίσια αυτής της εργασίας. Δεδομένου ότι ο ρυθμός ασφαλούς αποκωδικοποίησης δεδομένων έχει μονάδες [bps] θα μετατρέψουμε τα [CPU cycles] σε bit με τον συντελεστή c_k [CPU cycles / bit], ο οποίος εκφράζει το πλήθος κύκλων CPU που χρειάζονται για την επεξεργασία ενός bit δεδομένων. Έτσι, ο χρόνος που απαιτείται για την εκφόρτωση των δεδομένων από τον χρήστη k στον BS είναι:

$$T_k^{off} = \frac{o_k[t]S_k^{tot}[t]}{c_k r_k^{sec}} \text{ [sec]} \tag{3.15}$$

Η ενέργεια που καταναλώνει ο MU k για την εκπομπή των δεδομένων υπολογίζεται από την εξίσωση:

$$E_k^{off}[t] = \sum_{j=1}^{2} P_{k,j} T_k^{off} \text{ [Joule]} \tag{3.16}$$

3.2 Διατύπωση Προβλήματος

3.2.1 Συνάρτηση Κόστους

Καθώς το σύστημα επιτρέπει τη μερική εκφόρτωση της εργασίας, ο χρόνος που θα απαιτηθεί για την ολοκλήρωση της επεξεργασίας της εργασίας τη χρονική στιγμή t θα είναι:

$$T_k^{max}[t] = max\left(T_k^{exec}[t], T_k^{off}[t]\right) \tag{3.17}$$

ενώ η συνολική ενέργεια που απαιτείται για την ολοκλήρωση της εργασίας είναι:

$$E_k[t] = E_k^{exec}[t] + E_k^{off}[t] \tag{3.18}$$

Σκοπός του συστήματος είναι η ελαχιστοποίηση της καταναλισκόμενης ενέργειας κάθε χρονική στιγμή, ενώ παράλληλα όλες οι εργασίες να ολοκληρώνονται εντός του χρονικού διαστήματος Δ. Η συνθήκη αυτή εκφράζεται ως εξής:

$$T_k^{max} \leq \Delta \tag{3.19}$$

Με βάση τις εξισώσεις 3.18 και 3.19 μπορούμε να κατασκευάσουμε μία συνάρτηση κόστους για το σύστημα, ώστε να τη χρησιμοποιήσουμε ως μέτρο για την επίδοσή του. Μία τέτοια συνάρτηση δίνεται παρακάτω:

$$\mathcal{C}[t] = \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} wE_k[t] + \lambda[t] \tag{3.20}$$

όπου $|\mathcal{K}|$ το μέγεθος του συνόλου των κινητών χρηστών, w η παράμετρος βάρους της ενέργειας (βρίσκεται πειραματικά ώστε $\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} wE_k[t] \approx 1$) και $\lambda[t]$ η συνάρτηση τιμωρίας η οποία ελαχιστοποιείται όταν ικανοποιείται ο χρονικός περιορισμός 3.19. Η συνάρτηση τιμωρίας ορίζεται ως:

$$\lambda[t] = \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} \Xi\,\xi_k[t] \tag{3.21}$$

όπου $\Xi > 0$ η τιμή της τιμωρίας και

$$\xi_k[t] = \begin{cases} 1, \text{ if constraint 3.19 is not satisfied,} \\ 0, \text{ else} \end{cases} \tag{3.22}$$

3.2.2 Πρόβλημα Βελτιστοποίησης

Στόχος της εργασίας είναι η ελαχιστοποίηση της ενέργειας που καταναλώνουν αθροιστικά οι χρήστες, προσαρμόζοντας το ποσοστό εκφόρτωσης της εργασίας και το επίπεδο της ισχύος εκπομπής σε κάθε κανάλι για κάθε MU του συστήματος, καθώς και τη σειρά αποκωδικοποίησης των μηνυμάτων των χρηστών από τον σταθμό βάσης. Αυτό εκφράζεται από το παρακάτω πρόβλημα βελτιστοποίησης:

$$\begin{aligned} \min_{\phi, P_{k,j}, o_k, \forall k \in \mathcal{K}, \forall j \in \{1,2\}} \quad & \sum_{k=1}^{|\mathcal{K}|} E_k \\ \text{s.t.} \quad & \sum_{j=1}^{2} P_{k,j} \leq P_k^{max}, \ \forall k \in \mathcal{K} \\ & T_k^{max} \leq \Delta, \ \forall k \in \mathcal{K} \\ & P_{k,j} \geq 0, \ \forall k \in \mathcal{K}, \forall j \in \{1,2\} \\ & 0 \leq o_k \leq 1, \ \forall k \in \mathcal{K} \\ & \phi \in \Phi \end{aligned} \tag{3.23}$$

όπου Φ το σύνολο όλων των μεταθέσεων των ϕ.

Οι περιορισμοί για τα $P_{k,j}$ και o_k μπορούν να ικανοποιηθούν εύκολα, καθώς πρόκειται για τις μεταβλητές που ελέγχουμε στο σύστημα, όμως για τον περιορισμό $T_k^{max} \leq \Delta$ δεν είναι τόσο εύκολο καθώς εξαρτάται από την κατάσταση του συστήματος και υπολογίζεται μόνο μετά τη λήψη της απόφασης. Για τον λόγο αυτό μπορούμε να αφαιρέσουμε αυτόν τον περιορισμό από το πρόβλημα βελτιστοποίησης, αντικαθιστώντας το άθροισμα της ενέργειας με τη συνάρτηση κόστους 3.20, η οποία τον ενσωματώνει. Έτσι, προκύπτει το πρόβλημα:

$$\begin{aligned} \min_{\phi, P_{k,j}, o_k, \forall k \in \mathcal{K}, \forall j \in \{1,2\}} \quad & \mathcal{C}[t] \\ \text{s.t.} \quad & \sum_{j=1}^{2} P_{k,j} \leq P_k^{max}, \ \forall k \in \mathcal{K} \\ & P_{k,j} \geq 0, \ \forall k \in \mathcal{K}, \forall j \in \{1,2\} \\ & 0 \leq o_k \leq 1, \ \forall k \in \mathcal{K} \\ & \phi \in \Phi \end{aligned} \tag{3.24}$$

Στο παραπάνω πρόβλημα βελτιστοποίησης, όλοι οι περιορισμοί είναι για τις μεταβλητές ελέχου του συστήματος, τις οποίες έχουμε υπό τον έλεγχό μας και επομένως μπορούμε να εξασφαλίσουμε την ικανοποίησή τους. Αυτή η μορφή του προβλήματος βελτιστοποίησης θα χρησιμοποιηθεί για την εκπαίδευση ευφυών πρακτόρων μέσω ενισχυτικής μάθησης. Για να γίνει αυτό όμως, πρέπει το πρόβλημα 3.24 να εκφραστεί με τη μορφή Markov Decision Process (MDP).

4

Περιγραφή Μεθόδου Επίλυσης

Η εργασία αυτή επικεντρώνεται στην μακροπρόθεσμα βέλτιστη επίδοση του συστήματος MEC, ενώ στόχος είναι η ελαχιστοποίηση της ενέργειας που καταναλώνουν αθροιστικά οι κινητοί χρήστες του συστήματος (MUs). Για να επιτύχουμε τον στόχο μας θα εκπαιδεύσουμε ευφυείς πράκτορες, οι οποίοι αλληλεπιδρώντας με το περιβάλλον πλησιάζουν στη βέλτιστη πολιτική λήψης αποφάσεων. Οι πράκτορες αυτοί θα δημιουργηθούν και θα εκπαιδευτούν με χρήση βαθιάς ενισχυτικής μάθησης. Αυτό απαιτεί τη διατύπωση του προβλήματος βελτιστοποίησης 3.24 σε μορφή Markov Decision Process (MDP).

4.1 Μοντελοποίηση Προβλήματος ως MDP

Τα MDP είναι διαδικασίες διακριτού χρόνου, στις οποίες οι κατάσταση του περιβάλλοντος μεταβάλλεται στοχαστικά, ενώ η στοχαστική αυτή διαδικασία επηρεάζεται από τις ενέργειες που εκτελούμε μέσα στο περιβάλλον. Το μοντέλο του προβλήματος που κατασκευάσαμε είναι μοντέλο διακριτού χρόνου, στο οποίο κάθε χρονική στιγμή καλούμαστε να λάβουμε τη βέλτιστη απόφαση για την ελαχιστοποίηση της συνολικής ενέργειας που καταναλώνουν οι χρήστες. Χωρίζουμε τις διακριτές χρονικές στιγμές σε επεισόδια (episodes) $\mathcal{E}$ και χρονικά βήματα (timesteps) $\mathcal{T}$, με κάθε επεισόδιο να περιλαμβάνει ένα συγκεκριμένο πλήθος χρονικών βημάτων $|\mathcal{T}|$. Σε κάθε επεισόδιο οι θέσεις των MUs και του eavesdropper αλλάζουν, ενώ τοποθετούνται τυχαία στο οριοθετημένο ορθογώνιο. Μεταξύ των χρονικών βημάτων τα κέρδη καναλιών μεταβάλλονται με βάση το block fading model (εξίσωση 2.8).

Ένα MDP χαρακτηρίζεται από τον χώρο καταστάσεων (state space), τον χώρο δράσης (action space), μία συνάρτηση μετάβασης (transition function) και μία συνάρτηση ανταμοιβής (reward function), ενώ στόχος είναι η μεγιστοποίηση της ανταμοιβής. Καθώς το σύστημα με το οποίο θα ασχοληθούμε σε αυτή την εργασία πρόκειται για σύστημα πολλαπλών πρακτόρων, υπάρχουν τόσο συλλογικές όσο και ατομικές τιμές για κάθε ένα από τα χαρακτηριστικα των MDP.

4.1.1 Χώρος Καταστάσεων

Ο χώρος καταστάσεων (state space) $\mathcal{S}$ του MDP αποτελείται από τις σημαντικές πληροφορίες που έχουμε για το σύστημα. Στο δικό μας πρόβλημα, οι πληροφορίες για το περιβάλλον που επηρεάζουν τον χρόνο τοπικής εκτέλεσης T_K^{exec} και τον χρόνο εκφόρτωσης T_k^{off} και συνεπώς και το άθροισμα της ενέργειας $\sum_{k\in\mathcal{K}} E_k$, είναι οι τιμές του κέρδους καναλιού για κάθε ένα από τα κανάλια που μας απασχολούν (MU-BS, MU-eavesdropper) καθώς και το μέγεθος του task που καλείται ο κάθε χρήστης να επεξεργαστεί. Έτσι, προκύπτει ο παρακάτω χώρος καταστάσεων:

$$\mathcal{S} = \{G_1, G_2, ..., G_K, G_{1,e}, G_{2,e}, ..., G_{K,e}, S_1^{tot}, S_2^{tot}, ..., S_K^{tot}\} \tag{4.1}$$

όπου G_k το κέρδος καναλιού μεταξύ του χρήστη k και του BS, $G_{k,e}$ το κέρδος καναλιού μεταξύ του k και του eavesdropper και S_k^{tot} το μέγεθος της εργασίας που αναλαμβάνει να εκτελέσει ο k. Όπως προκύπτει από την παραπάνω σχέση, το μέγεθος του χώρου καταστάσεων είναι $|S| = 3K$, όπου K το πλήθος των χρηστών του συστήματος.

Ο ατομικός χώρος καταστάσεων για τον χρήστη k αποτελείται από τις πληροφορίες του χώρου καταστάσεων που του είναι διαθέσιμες μέσω τοπικών παρατηρήσεων. Έτσι, ορίζεται ο ατομικός χώρος καταστάσεων:

$$\mathcal{S}_k = \{G_k, G_{k,e}, S_k^{tot}\} \tag{4.2}$$

ο οποίος έχει μέγεθος $|\mathcal{S}_k| = 3$.

4.1.2 Χώρος Δράσης

Ο χώρος δράσης (action space) $\mathcal{A}$ περιέχει τις ενέργειες που μπορούν να επηρεάσουν την πιθανότητα μετάβασης του συστήματος σε μία επόμενη κατάσταση και την ανταμοιβή του συστήματος. Στο δικό μας πρόβλημα, ο χώρος δράσης είναι:

$$\mathcal{A} = \{P_{1,1}, P_{2,1}, .., P_{K,1}, P_{1,2}, P_{2,2}, .., P_{K,2}, o_1, o_2, ..., o_K, \phi\} \tag{4.3}$$

όπου $P_{k,j}$ η ισχύς εκπομπής του χρήστη k για το μήνυμα j και o_k το ποσοστό του task που κάνει offload ο k. Το πεδίο τιμών των ενεργειών μπορεί να εξαρτάται από την κατάσταση στην οποία βρίσκεται το σύστημα. Στη δική μας περίπτωση δεν ισχύει κάτι τέτοιο, οπότε οι τιμές που μπορούν να πάρουν οι ενέργειες ορίζονται ως εξής:

$$\begin{aligned} &P_{k,1} \in [0, P_k^{max}],\ \forall k \in \mathcal{K} \\ &P_{k,2} \in [0, (P_k^{max} - P_{k,1})],\ \forall k \in \mathcal{K} \\ &o_k \in [0, 1],\ \forall k \in \mathcal{K} \end{aligned} \tag{4.4}$$

Ο περιορισμός στην ισχύ εκπομπής είναι ο ίδιος περιορισμός που είχαμε στο πρόβλημα βελτιστοποίησης 3.24 και εξασφαλίζει πως η συνολική ισχύς εκπομπής του χρήστη k είναι το πολύ ίση με τη μέγιστη ισχύ εκπομπής P_k^{max}. Το o_k εκφράζει ένα ποσοστό του task, οπότε είναι 0 αν αυτό εκτελείται μόνο τοπικά και 1 αν εκφορτώνεται πλήρως, ενώ μπορεί να πάρει και οποιαδήποτε ενδιάμεση τιμή. Ο χώρος δράσης έχει μέγεθος $|\mathcal{A}| = 3K + 1$.

Όπως και με τον χώρο καταστάσεων μπορούμε να ορίσουμε τον ατομικό χώρο δράσης $\mathcal{A}_k$ για τον χρήστη k από τις ενέργειες του χώρου δράσης τις οποίες αποφασίζει ως εξής:

$$\mathcal{A}_k = \{P_{k,1}, P_{k,2}, o_k\} \tag{4.5}$$

Παρατηρούμε πως το μέγεθος του ατομικού χώρου δράσης είναι $|\mathcal{A}_k| = 3$, το οποίο σημαίνει πως υπάρχει μία ενέργεια που δεν επηρεάζεται από τους πράκτορες του συστήματος. Η ενέργεια αυτή είναι η σειρά αποκωδικοποίησης των μηνυμάτων στο σταθμό βάσης. Αυτό μπορούμε να το αντιμετωπίσουμε ξεχωρίζοντας αυτή την ενέργεια από το υπόλοιπο πρόβλημα και λύνοντας το νέο πρόβλημα στον σταθμό βάσης μετά τη λήψη των αποφάσεων από όλους τους πράκτορες του συστήματος, δηλαδή μετά τη ολοκλήρωση των εκπομπών όλων των μηνυμάτων. Έτσι, ο νέος χώρος δράσης ορίζεται ως:

$$\mathcal{A} = \{P_{1,1}, P_{2,1}, .., P_{K,1}, P_{1,2}, P_{2,2}, .., P_{K,2}, o_1, o_2, ..., o_K\} \tag{4.6}$$

Το πρόβλημα της εύρεσης της βέλτιστης σειράς αποκωδικοποίησης ϕ θα αναλυθεί παρακάτω.

4.1.3 Συνάρτηση Μετάβασης

Η συνάρτηση μετάβασης (transition function) $\mathcal{T}(s, a, s')$ εκφράζει την πιθανότητα μετάβασης από την κατάσταση s στην κατάσταση s' μετά την εκτέλεση της ενέργειας a. Όπως αναφέραμε και στην ενότητα 2.4.1, υπάρχουν δύο είδη προβλημάτων: τα model-based, στα οποία η δυναμική του περιβάλλοντος είναι γνωστή στον πράκτορα και τα model-free, τα οποία δεν χρησιμοποιούν κάποιο μοντέλο για τη δυναμική του περιβάλλοντος, αλλά ο πράκτορας μαθαίνει απευθείας μέσω της αλληλεπίδρασης με αυτό. Το δικό μας πρόβλημα είναι model-free και για αυτό δεν ορίζεται συνάρτηση μετάβασης $\mathcal{T}$.

4.1.4 Συνάρτηση Ανταμοιβής

Στα MDP ορίζονται διάφορα είδη συναρτήσεων ανταμοιβής (reward function) $\mathcal{R}$. Αυτή που θα χρησιμοποιήσουμε εμείς αξιολογεί μία ενέργεια a όταν αυτή εκτελείται στην κατάσταση s, ορίζεται δηλαδή ως $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Ο στόχος σε ένα MDP είναι η μεγιστοποίηση της ανταμοιβής. Όπως μπορούμε να παρατηρήσουμε, πρόκειται για την αντίστροφη διαδικασία από την ελαχιστοποίηση του κόστους στο πρόβλημα βελτιστοποίησης 3.24. Καθώς το κόστος 3.20 είναι πάντοτε μη αρνητικό, μπορούμε να ορίσουμε τη συνάρτηση ανταμοιβής ως το αντίθετο της συνάρτησης κόστους που κατασκευάσαμε πριν:

$$\mathcal{R}[t] = -\mathcal{C}[t] = -\left(\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} wE_k[t] + \lambda[t]\right) \tag{4.7}$$

Στο πρόβλημά μας ορίζουμε την ατομική συνάρτηση ανταμοιβής ίδια με τη συλλογική συνάρτηση ανταμοιβής, καθώς επιθυμούμε οι πράκτορες να προσπαθούν για την ελαχιστοποίηση

της συνολικής κατανάλωσης ενέργειας και την ικανοποίηση των χρονικών περιορισμών από όλους τους πράκτορες. Ορίζοντας ξεχωριστές συναρτήσεις ανταμοιβής για κάθε πράκτορα ίσως παροτρύναμε τους πράκτορες να μεγιστοποιήσουν τη δική τους ανταμοιβή χωρίς να ενδιαφέρονται για το αντίκτυπο στην επίδοση των υπολοίπων πρακτόρων και έτσι να αποθαρρύναμε τη συνεργασία μεταξύ τους.

4.2 Σειρά αποκωδικοποίησης

Η κατάλληλη σειρά αποκωδικοποίησης μπορεί να ελαττώσει σημαντικά τις παρεμβολές των άλλων υπομηνυμάτων κατά τη διάρκεια του SIC και έτσι να βελτιώσει τη συνολική επίδοση του συστήματος. Για τον λόγο αυτό, η εύρεση της βέλτιστης σειράς αποκωδικοποίησης αποτελεί έναν από τους καθοριστικότερους παράγοντες για την αύξηση της ανταμοιβής του συστήματος.

Έχοντας διαχωρίσει το πρόβλημα αυτό από το πρόβλημα της εύρεσης της βέλτιστης ισχύος εκπομπής και του βέλτιστου ποσοστού εκφόρτωσης, μπορούμε να θεωρήσουμε πως η απόφαση της σειράς αποκωδικοποίησης πραγματοποιείται μετά την ολοκλήρωση των ενεργειών των πρακτόρων. Έτσι, στον BS φτάνει η συμβολή των σημάτων που εκπέμπουν όλοι οι MUs και τότε αυτός καλείται να αποφασίσει τη σειρά αποκωδικοποίησης των μηνυμάτων ϕ. Το πρόβλημα εύρεσης της βέλτιστης σειράς αποκωδικοποίησης εκφράζεται ως εξής:

$$\begin{aligned} \max_{\phi} \quad & \sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{J}} r_{k,j}^{sec} \\ \text{s.t.} \quad & \phi \in \Phi \end{aligned} \tag{4.8}$$

όπου Φ το σύνολο όλων των δυνατών μεταθέσεων των σειρών αποκωδικοποίησης όλων των μηνυμάτων και ϕ μία τέτοια σειρά αποκωδικοποίησης.

Μία προσέγγιση για την επίλυση του παραπάνω προβλήματος βελτιστοποίησης είναι η υιοθέτηση μίας σταθερής σειράς αποκωδικοποίησης βασισμένης στη μέση τιμή του κέρδους καναλιού του κάθε χρήστη προς τον σταθμό βάσης (ή αντίστοιχα της απόστασής του από εκείνον), όπως γίνεται για παράδειγμα στο [7]. Αυτή η προσέγγιση, αν και δεν υπόσχεται την καλύτερη λύση, υλοποιείται εύκολα και μειώνει το overhead των υπολογισμών. Ένας άλλος τρόπος επίλυσης είναι η εξαντλητική αναζήτηση, όπως γίνεται στο [58]. Αν και είναι σίγουρη η εύρεση της βέλτιστης λύσης, το επιπλέον κόστος υπολογισμού κρίνεται απαγορευτικό σε μεγάλα συστήματα.

Η λύση που θα ακολουθήσουμε δίνεται στην εργασία [24]. Σε αυτή, μία καλή σειρά αποκωδικοποίησης δίνεται από τη σχέση:

$$\eta_{k,j} \triangleq |G_k|^2 \left(1 + \frac{1}{SINR_{k,j}}\right) \tag{4.9}$$

Η σειρά αποκωδικοποίησης ϕ των μηνυμάτων $s_{k,j}$ υπολογίζεται ως η φθίνουσα σειρά των $\eta_{k,j}$. Σύμφωνα με το [22], αντί για τη σχέση 4.9 μπορούμε να χρησιμοποιήσουμε την παρακάτω:

$$\eta_{k,j} = |G_k|^2 \left(1 + \frac{1}{2^{rate_k^{min}}}\right) \tag{4.10}$$

όπου $rate_k^{min}$ ο ελάχιστος ρυθμός για την εκφόρτωση του μηνύματος του k. Αυτός υπολογίζεται ως:

$$rate_k^{min} = o_k \frac{S_k^{max}}{c_k \Delta} \tag{4.11}$$

όπου o_k το ποσοστό της εργασίας που εκφορτώνεται, S_k^{max} το μέγιστο μέγεθος εργασίας που μπορεί να αναλάβει ο MU k, c_k το CPU cycles/bit του k και Δ η διάρκεια ενός χρονικού διαστήματος.

4.3 Εφαρμογή Αλγορίθμου MADDPG

Ο αλγόριθμος βαθιάς ενισχυτικής μάθησης πολλαπλών πρακτόρων που θα χρησιμοποιήσουμε βασίζεται στον αλγόριθμο MADDPG ο οποίος προτάθηκε στο [7]. Πρόκειται για έναν actor-critic αλγόριθμο, ο οποίος χειρίζεται πολλαπλούς πράκτορες και συνεχή action spaces για να πραγματοποιήσει ενισχυτική μάθηση.

Ο MADDPG δημιουργεί έναν πράκτορα για κάθε έναν από τους κινητούς χρήστες. Ο πράκτορας αυτός στον χρήστη k περιλαμβάνει ένα actor network $\mu_k(s_k|\theta_k^\mu)$ με παράμετρο θ_k^μ, η οποία καθορίζει την πολιτική του πράκτορα. Ο πράκτορας, με χρήση του actor network, αποφασίζει τις ενέργειες a_k του k σε κάθε χρονική στιγμή, χρησιμοποιώντας ως είσοδο τις τοπικές παρατηρήσεις του χρήστη για την κατάσταση του συστήματος s_k. Ο πράκτορας περιλαμβάνει επίσης και ένα critic network $Q_k(s, a_k|\theta_k^Q)$ με παράμετρο θ_k^Q που αξιολογεί τις ενέργειες a_k του χρήστη στην κατάσταση s.

Η εκπαίδευση αυτών των δικτύων γίνεται κεντρικά στον ES με χρήση target actor network $\mu'_k(s_k|\theta_k^\mu)$ με παράμετρο θ_k^μ και target critic network $Q'_k(s|\theta_k^Q)$ με παράμετρο θ_k^Q. Η χρήση target networks γίνεται για να βελτιωθεί η σταθερότητα του αλγορίθμου κατά τη διάρκεια της εκπαίδευσης. Ένα αντίγραφο του actor network τοποθετείται στον χρήστη ώστε να έχει τη δυνατότητα αποκεντρωμένης λήψης αποφάσεων.

Σε κάθε χρονικό βήμα ο πράκτορας k συγκεντρώνει τις τοπικές παρατηρήσεις του για την κατάσταση του συστήματος σε μία μερική κατάσταση s_k και χρησιμοποιώντας τη ως είσοδο στο actor network του, αποφασίζει την ενέργεια a_k που θα εκτελέσει. Αφού εκτελεστούν οι ενέργειες από όλους του χρήστες, το περιβάλλον μεταφέρεται σε μία νέα κατάσταση s_{k+1} και οι χρήστες μαθαίνουν την ανταμοιβή τους για τη χρονική στιγμή που πέρασε. Ο ES συγκεντρώνει τις τοπικές παρατηρήσεις, ενέργειες και ανταμοιβές από όλους τους πράκτορες και τις αποθηκεύει σε ένα δείγμα global experience.

Σε κάθε βήμα της εκπαίδευσης, οι πράκτορες μαθαίνουν σε batches $\mathbf{B}$ αποθηκευμένων εμπειριών μεγέθους $|\mathbf{B}| = B$. Η παράμετρος θ_k^Q του critic network k ενημερώνεται ελαχιστοποιώντας τη διαφορά (loss) μεταξύ της συνάρτησης action-value $Q_k(s_i, a_i|\theta_k^Q)$ και του target value $y_{i,k}$, η οποία υπολογίζεται από τη σχέση:

$$L_k = \frac{1}{B}\sum_{i=1}^{B}\left(y_{i,k} - Q_k\left(s_i, a_i|\theta^Q\right)\right)^2 \tag{4.12}$$

όπου a_i και s_i η ενέργεια και η κατάσταση στο δείγμα i του batch $\mathcal{B}$. Το target value $y_{i,k}$ του δείγματος i υπολογίζεται ως:

$$y_{i,k} = r_i + \gamma Q'_k\left(s_{i+1}, \mu'_k(s_{i+1}|\theta^{\mu'_k})|\theta^{Q'_k}\right) \tag{4.13}$$

όπου r_i το reward του δείγματος i, που όπως είδαμε παραπάνω είναι ίδιο για όλους τους πράκτορες, s'_{i+1} η επόμενη κατάσταση του δείγματος και γ το discounting factor που σταθμίζει την τωρινή ανταμοιβή του συστήματος με την αναμενόμενη μελλοντική του ανταμοιβή.

Η παράμετρος θ^μ_k του actor network του πράκτορα k ενημερώνεται από το policy gradient ως εξής:

$$\nabla_{\theta^\mu} J_k \approx \frac{1}{B}\sum_{i=1}^{B} \nabla_a Q_k(s, a|\theta^Q_k)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu_k(s|\theta^\mu_k)|_{s_i} \tag{4.14}$$

Τέλος, οι παράμετροι των target networks ενημερώνονται με soft update με μία μικρή θετική τιμή $\tau \ll 1$:

$$\begin{aligned} \theta^{\mu'}_k &\leftarrow \tau\theta^\mu_k + (1-\tau)\theta^{\mu'}_k, \\ \theta^{Q'}_k &\leftarrow \tau\theta^Q_k + (1-\tau)\theta^{Q'}_k \end{aligned} \tag{4.15}$$

Για να εξασφαλίσει επαρκή εξερεύνηση στο χώρο δράσης, ο MADDPG προσθέτει θόρυβο στην πολιτική του actor κατά την εκπαίδευση. Κατά αυτόν τον τρόπο ο πράκτορας δοκιμάζει διαφορετικές ενέργειες και περιπλανάται περισσότερο στον χώρο δράσης. Έτσι, τη χρονική στιγμή t, η ενέργεια του χρήστη k προκύπτει ως εξής:

$$a_k[t] = \mu_k(s_k[t]|\theta^\mu_k) + n[t] \tag{4.16}$$

όπου $n[t]$ μία στοχαστική διαδικασία θορύβου. Οι πιο συνηθισμένες διαδικασίες [19] είναι η Gaussian και η Ornstein-Uhlenbeck [49]. Στην εργασία αυτή επιλέχθηκε η χρήση Gaussian θορύβου.

5

Πειράματα - Αποτελέσματα

5.1 Παράμετροι Προσομοίωσης

Προχωρώντας στο πειραματικό μέρος της εργασίας, θα αξιολογήσουμε την επίδοση του συστήματος προσομοιώνοντας ένα περιβάλλον MEC με 4 mobile users, 1 BS με 1 ES και 1 eavesdropper. Το κελί στο οποίο τοποθετούνται τυχαία οι MUs και ο eavesdropper έχει διαστάσεις (500×500) και είναι τετράγωνο στο σχήμα, ενώ ο σταθμός βάσης τοποθετείται στο κέντρο του τετραγώνου.

Το δίκτυο του actor έχει ως είσοδο τη μερική κατάσταση s_i μεγέθους $|s_i| = 3$ που προκύπτει από τις τοπικές παρατηρήσεις του πράκτορα, έχει ως έξοδο την ενέργεια a_i του πράκτορα, μεγέθους $|a_i| = 3$ και περιέχει δύο κρυφά επίπεδα, το πρώτο με 128 και το δεύτερο με 256 νευρώνες. Όλα τα επίπεδα είναι fully connected (FC).

Το δίκτυο του critic έχει για είσοδο την πλήρη κατάσταση του συστήματος s με $|s| = 3K$ και έχει για έξοδο έναν νευρώνα που υπολογίζει το loss του critic. Αποτελείται και αυτό από δύο κρυφά επίπεδα, το πρώτο εκ των οποίων γίνεται concatenate με τις ενέργειες a όλων των πρακτόρων και έχει $512 + |a|$ νευρώνες, όπου $|a| = 3K$ οι ενέργειες όλων των πρακτόρων, ενώ το δεύτερο έχει 1024 νευρώνες. Όπως και στο actor network, έτσι και σε αυτό όλα τα επίπεδα είναι FC.

Στις εισόδους των δικτύων γίνεται κανονικοποίηση για να αποφευχθούν οι μεγάλες ταλαντώσεις των βαρών των νευρωνικών. Οι παράμετροι του μοντέλου, καθώς και οι παράμετροι του περιβάλλοντος, δίνονται στον πίνακα 5.1. Για να αξιολογηθεί η επίδοση του συστήματος θα συγκριθεί με τις παρακάτω υλοποιήσεις:

- ***NOMA***: Σε αυτό το πείραμα, η επικοινωνία μεταξύ mobile users και edge server γίνεται χρησιμοποιώντας την τεχνική non-orthogonal multiple access (NOMA) ώστε να συγκριθεί με την επίδοση της RSMA. Σε αυτό το σύστημα θα χρησιμοποιηθεί ο αλγόριθμος MADDPG χωρίς αλλαγές.
- ***Random***: Ο αλγόριθμος MADDPG θα χρησιμοποιηθεί μόνο για τον υπολογισμό της βέλτιστης ισχύος εκπομπής, ενώ το offloading ratio o_k δίνεται τυχαία από μία κανονική και μία ομοιόμορφη κατανομή.

- ***Full offloading***: Η εργασία εκφορτώνεται πλήρως. Χρησιμοποιείται ο αλγόριθμος MADDPG για να υπολογιστούν οι βέλτιστες ισχύς εκπομπής.

Πίνακας 5.1: Παράμετροι Προσομοίωσης

Παράμετρος	**Τιμή**
Μέγεθος εργασίας, $S_k^{tot}[t]$	$[1 - 1.5] \cdot 10^6$ CPU cycles
Απαιτούμενοι υπολογιστικοί πόροι, c_k	50 CPU cycles / bit
Διαστάσεις κελιού, $(X \times Y)$	(500×500) m
Πλήθος MU, $\mathcal{K}$	4
Διάρκεια χρονικής στιγμής, Δ	0.1 sec
Bandwidth, B	1 MHz
Μέγιστη ισχύς μετάδοσης, P_k^{max}	24 dBm
Συχνότητα CPU κινητών συσκευών, f	6 MHz
Effective capacitance coefficient, ρ	10^{-28}
Μέγιστη συχνότητα Doppler, f_d	10 Hz
Ισχύς θορύβου, σ^2	-174 dBm/Hz
Shadow fading, σ_2^2	4 dB
Επεισόδια, $\mathcal{E}$	3000
Χρονικά βήματα, $\mathcal{T}$	300
Μέγεθος μνήμης, $\mathcal{R}$	$2 \cdot 10^6$
Μέγεθος batch, $\mathcal{B}$	32
Discount factor, γ	0.99
Σταθερά soft update, τ	10^{-5}
Πλήθος βημάτων για ενημέρωση local actors, β	10
Θόρυβος πολιτικής actor, $n[t]$	$n[t] \sim \mathcal{N}(0, 0.1)$
Actor learning rate, LR_{actor}	10^{-5}
Critic learning rate, LR_{critic}	10^{-4}
Παράμετρος βάρους ενέργειας, w	200
Τιμή τιμωρίας, Ξ	5

5.2 Ανάλυση Αποτελεσμάτων

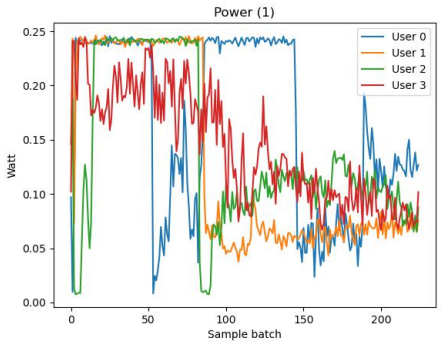
Για την παρακολούθηση της διαδικασίας της εκπαίδευσης κρατάμε ένα δείγμα κάθε 100 timesteps. Καθώς η εκπαίδευση αποτελείται από 3000 επεισόδια, καθένα από τα οποία περιέχει 300 timesteps, έχουμε συνολικά 9000 δείγματα από την εκπαίδευση του συστήματος. Στο κάθε δείγμα κρατάμε το secure data rate για κάθε μήνυμα ξεχωριστά αλλά και συνολικά για κάθε χρήστη, τον χρόνο που απαιτείται για το offload, τον χρόνο που χρειάζεται ο MU για την τοπική εκτέλεση, τον χρόνο που χρειάζεται συνολικά για την εκτέλεση της εργασίας του MU, τις ισχύς εκπομπής για τα μηνύματα 1 και 2 αλλά και τη συνολική ισχύ εκπομπής, το ποσοστό του offload (split), τις ενέργειες εκφόρτωσης και εκτέλεσης καθώς και τη συνολική ενέργεια για

κάθε χρήστη και τέλος την ανταμοιβή του συστήματος (reward), η οποία είναι ίδια για όλους τους χρήστες.

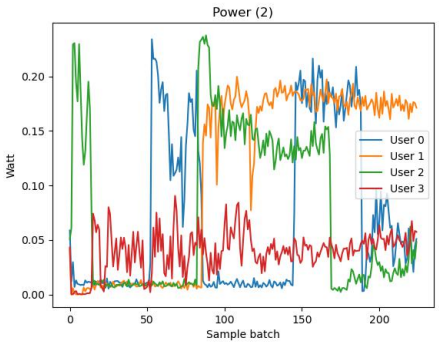
Καθώς ο όγκος των δεδομένων είναι μεγάλος και παρατηρείται ταλάντωση στις τιμές (κυρίως σε αυτές που επηρεάζονται από το channel gain αφού αυτό μεταβάλλεται σημαντικά σε κάθε χρονική στιγμή) και για να φαίνεται πιο ξεκάθαρα η μέση συμπεριφορά του συστήματος, αποφασίσαμε να παρουσιάσουμε στις γραφικές παραστάσεις τη μέση τιμή ανά 40 δείγματα σε κάθε μετρική. Αυτό σημαίνει πως ομαδοποιούμε τα δείγματα σε batches των 40 δειγμάτων και εμφανίζουμε σε κάθε γραφική παράσταση το μέσο όρο για κάθε batch.

5.2.1 Βασική Περίπτωση

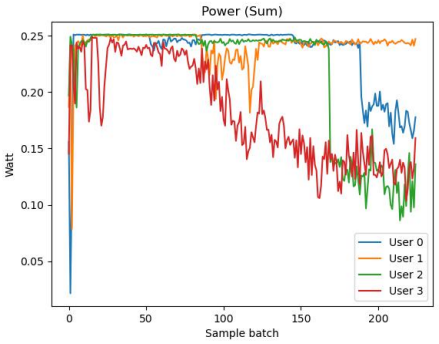
Παρακάτω φαίνονται τα αποτελέσματα του πειράματος που περιγράψαμε παραπάνω. Οι τιμές των παραμέτρων είναι ίδιες με τις τιμές που περιγράφονται στον πίνακα 5.1.



(α΄) Η ισχύς εκπομπής του μηνύματος 1 στη βασική περίπτωση



(β΄) Η ισχύς εκπομπής του μηνύματος 2 στη βασική περίπτωση

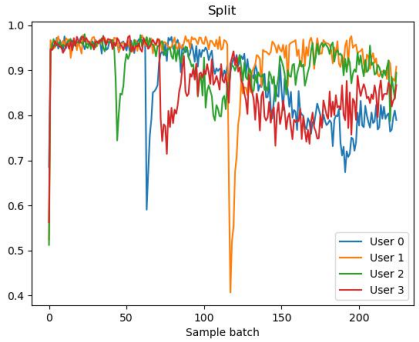


(γ΄) Η συνολική ισχύς εκπομπής στη βασική περίπτωση

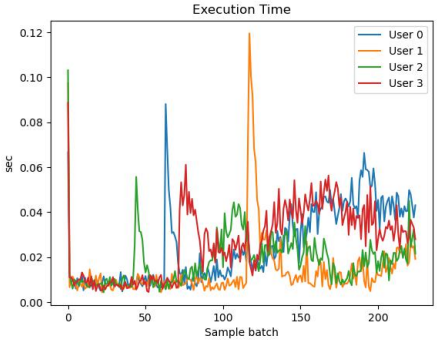
Σχήμα 5.1: Ισχύς μηνυμάτων στη βασική περίπτωση

Παρατηρούμε από τις γραφικές παραστάσεις 5.1 πως οι χρήστες επικοινωνούν με τον σταθμό βάσης με χρήση και των δύο μηνυμάτων. Οι χρήστες σταδιακά μειώνουν την ισχύ που χρησιμοποιούν για να εκπέμπουν τα μηνύματά τους, καθώς βρίσκουν μία ισορροπία στην οποία η εκπομπή του μηνύματός τους δεν επηρεάζει αρνητικά τον ρυθμό εκπομπής των άλλων χρηστών. Παρατηρούμε επίσης πως κατά τη διάρκεια της εκπαίδευσης οι χρήστες δεν μειώνουν ταυτό-

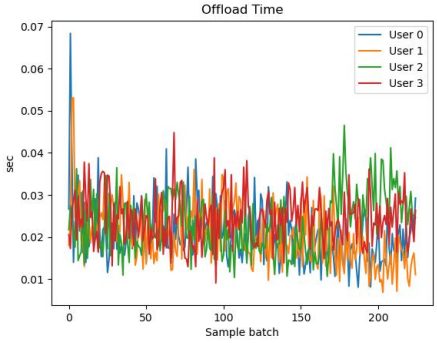
χρονα την ισχύ τους, αλλά ο κάθε χρήστης φτάνει σε ένα σημείο ισορροπίας πριν αρχίσει ο επόμενος να μειώνει τη συνολική του ισχύ. Αυτό δεν γίνεται από σχέδιο, καθώς οι χρήστες δεν επικοινωνούν άμεσα μεταξύ τους, αλλά ανακύπτει από τη δυναμική του συστήματος.



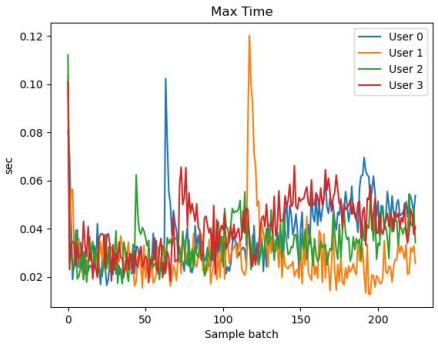
Σχήμα 5.2: Ποσοστό offload (split) στη βασική περίπτωση



(α΄) Ο χρόνος τοπικής εκτέλεσης στη βασική περίπτωση



(β΄) Ο χρόνος εκφόρτωσης στη βασική περίπτωση

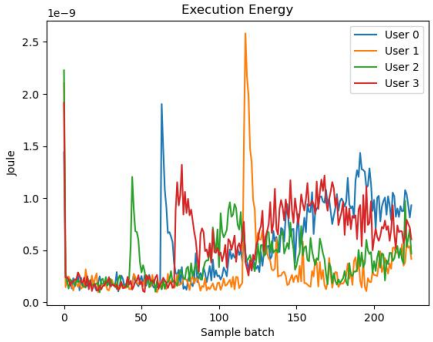


(γ΄) Ο χρόνος για την ολοκλήρωση της επεξεργασίας στη βασική περίπτωση

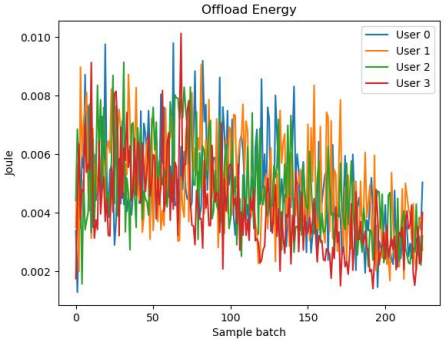
Σχήμα 5.3: Χρόνος επεξεργασίας εργασιών στη βασική περίπτωση

Στο 5.2 βλέπουμε το split για κάθε έναν από τους κινητούς χρήστες. Μεγάλες τιμές στο split κρατάνε ελάχιστο από το task για τοπική εκτέλεση αυξάνοντας την ενέργεια που καταναλώνει το σύστημα, αλλά ελαχιστοποιώντας τον απαιτούμενο για την επεξεργασία της εργασίας

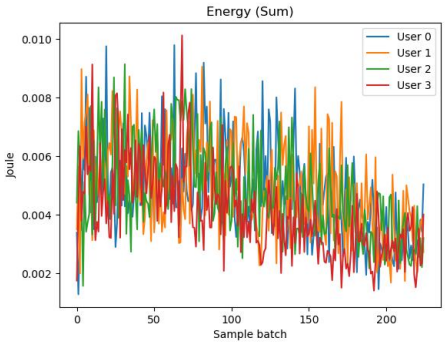
χρόνο. Μικρές τιμές ελαχιστοποιούν την ενέργεια αλλά αυξάνουν τον απαιτούμενο χρόνο. Παρατηρούμε πως μετά από μία περίοδο εξερεύνησης καταλήγουν όλοι τους σε μία τιμή μεταξύ του 0.8 και του 0.9. Το split μειώνεται κατά την εκπαίδευση σε πολύ χαμηλά επίπεδα, το οποίο έχει ως αποτέλεσμα η τοπική εκτέλεση να καθυστερεί περισσότερο από το χρονικό όριο των 0.1 δευτερολέπτων και να ενεργοποιείται η ποινή στη συνάρτηση ανταμοιβής του αλγορίθμου ενισχυτικής μάθησης. Καθώς η τιμή αυτή είναι αρκετά μεγαλύτερη της συνάρτησης ανταμοιβής, ο πράκτορας διορθώνει ταχύτατα την απόφασή του, επαναφέροντας το split σε μεγαλύτερες τιμές. Στη συνέχεια, με μικρότερες μεταβολές, οι πράκτορες συγκλίνουν σε μία τιμή η οποία εξασφαλίζει την ικανοποίηση του χρονικού περιορισμού του προβλήματος, ενώ παράλληλα πραγματοποιεί εκφόρτωση μέρους της εργασίας για μείωση της καταναλισκόμενης ενέργειας.



(α΄) Η ενέργεια τοπικής εκτέλεσης στη βασική περίπτωση



(β΄) Η ενέργεια εκφόρτωσης στη βασική περίπτωση

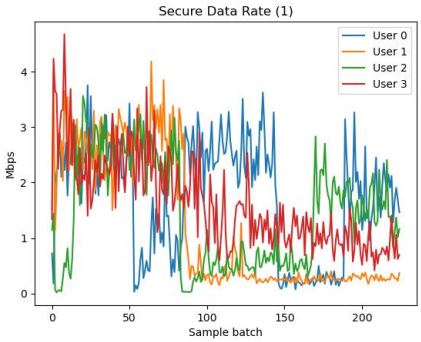


(γ΄) Η συνολική ενέργεια στη βασική περίπτωση

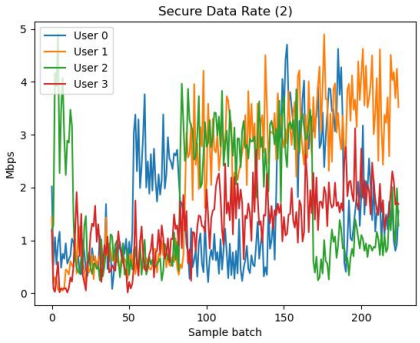
Σχήμα 5.4: Καταναλισκόμενη ενέργεια στη βασική περίπτωση

Οι γραφικές παραστάσεις 5.3 δείχνουν τον χρόνο που απαιτείται για την τοπική εκτέλεση του μέρους της εργασίας που δεν εκφορτώνεται, τον χρόνο που χρειάζεται για την εκφόρτωση του μέρους της εργασίας που εκφορτώνεται και τον συνολικό χρόνο για την επεξεργασία της εργασίας του κάθε χρήστη. Παρατηρούμε πως ο συνολικός χρόνος επεξεργασίας αυξομειώνεται και ξεπερνά το όριο των 0.1 sec κατά την εκπαίδευση, συγκλίνει όμως τελικά σε τιμές αρκετά κάτω του ορίου. Τα "καρφία" που παρατηρούμε στον χρόνο τοπικής εκτέλεσης τα οποία ξεπερνάνε το χρονικό όριο των 0.1 sec βλέπουμε πως συσχετίζονται απόλυτα με τα ανάποδα "καρφιά" στο split, το οποίο και εξηγεί γιατί διορθώνονται τόσο γρήγορα όπως εξηγήσαμε και

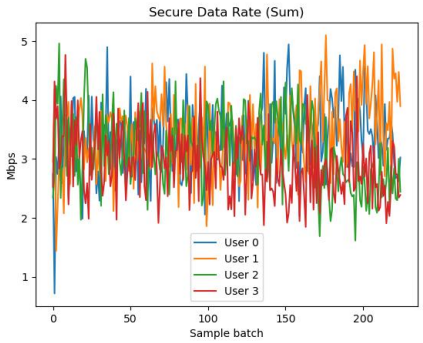
στην αντίστοιχη παράγραφο. Ο χρόνος εκφόρτωσης είναι πολύ λιγότερο σταθερός από τον χρόνο τοπικής εκτέλεσης καθώς εξαρτάται από το secure data rate, το οποίο με τη σειρά του εξαρτάται από το channel gain. Το channel gain όπως είπαμε νωρίτερα μεταβάλλεται σημαντικά ανάμεσα στις χρονικές στιγμές, το οποίο και εξηγεί αυτή τη συμπεριφορά.



(α΄) Το secure data rate για το μήνυμα 1 στη βασική περίπτωση



(β΄) Το secure data rate για το μήνυμα 2 στη βασική περίπτωση



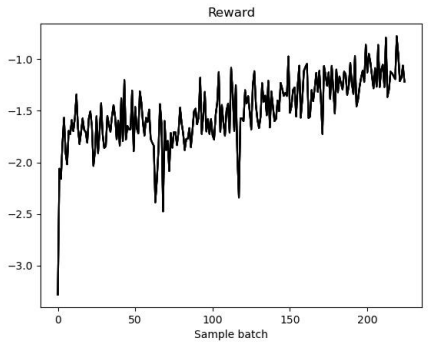
(γ΄) Το συνολικό secure data rate στη βασική περίπτωση

Σχήμα 5.5: Secure data rate στη βασική περίπτωση

Οι γραφικές παραστάσεις 5.4 μας δείχνουν την ενέργεια που καταναλώνει ο κάθε χρήστης ξεχωριστά για την τοπική εκτέλεση του μέρους της εργασίας που δεν κάνει offload και για την εκφόρτωση της υπόλοιπης εργασίας, καθώς και το άθροισμά τους. Συγκρίνοντας τη γραφική παράσταση της ενέργειας τοπικής εκτέλεσης με τον χρόνο τοπικής εκτέλεσης βλέπουμε πως τα δύο αυτά μεγέθη είναι εντελώς ανάλογα, όπως άλλωστε περιμέναμε. Η ενέργεια εκφόρτωσης εμφανίζει τις ίδιες διακυμάνσεις με τον χρόνο εκφόρτωσης. Η συμπεριφορά αυτή είναι επίσης αναμενόμενη, καθώς η ενέργεια εκφόρτωσης εκφράζεται ως το γινόμενο του χρόνου εκφόρτωσης επί τη συνολική ισχύ εκπομπής. Καθώς -όπως παρατηρήσαμε και στην αντίστοιχη γραφική παράσταση- η συνολική ισχύς εκπομπής μειώνεται κατά την εκπαίδευση, παρατηρείται μία ξεκάθαρη πτωτική τάση και στην ενέργεια που καταναλώνεται για την εκπομπή των μηνυμάτων προς τον σταθμό βάσης. Η ενέργεια εκφόρτωσης είναι μεγαλύτερη από την ενέργεια τοπικής εκτέλεσης κατά αρκετές τάξεις μεγέθους, το οποίο σημαίνει πως προς το τέλος της εκπαίδευσης η συνολική καταναλισκόμενη ενέργεια του συστήματος μειώνεται σημαντικά. Η πτωτική τάση της ενέργειας είναι απόδειξη της ορθής λειτουργίας του συστήματος, καθώς είναι ένας από

τους δύο στόχους του αλγορίθμου που έχουμε εφαρμόσει, με τον δεύτερο να είναι η ικανοποίηση των χρονικών περιορισμών την οποία έχουμε επίσης επιτύχει σε σημαντικό βαθμό.

Στο 5.5 βλέπουμε το secure data rate για κάθε ένα από τα μηνύματα για όλους τους χρήστες. Παρατηρούμε και εδώ σημαντικές αυξομειώσεις, που όπως και πριν οφείλονται στις αυξομειώσεις των channel gain μεταξύ των χρονικών στιγμών. Βλέπουμε επίσης πως ο χρήστης MU_1 ο οποίος εκπέμπει με τη μεγαλύτερη συνολική ισχύ επιτυγχάνει και το μεγαλύτερο secure data rate ανάμεσα σε όλους τους χρήστες, ενώ ο χρήστης MU_0 ο οποίος έχει τη δεύτερη μεγαλύτερη ισχύ, έχει το δεύτερο μεγαλύτερο secure data rate. Αυτό συμβαίνει διότι το secure data rate είναι ανάλογο της ισχύος εκπομπής των μηνυμάτων προς τον BS. Ωστόσο, παρατηρούμε πως αυτό δεν είναι το μοναδικό κριτήριο, αφού οι παρεμβολές στα μηνύματα ενός χρήστη από τα μηνύματα των άλλων χρηστών μπορεί να είναι αρκετά ισχυρές και να ρίχνουν το secure data rate σε επίπεδα χαμηλότερα από εκείνα των άλλων χρηστών ακόμα και αν η ισχύς εκπομπής είναι πάντα η μεγαλύτερη για τον χρήστη αυτόν.



Σχήμα 5.6: Ανταμοιβή (reward) στη βασική περίπτωση

Τέλος, στο 5.6 βλέπουμε το reward του συστήματος στο πείραμά μας. Στην αρχή της εκπαίδευσης ο συνολικός χρόνος που απαιτείται για την επεξεργασία μίας εργασίας είναι αρκετά μεγαλύτερος από το χρονικό όριο που έχουμε βάλει στο σύστημά μας, το οποίο και αντικατοπτρίζεται στην πολύ χαμηλή τιμή του reward στην αρχή της γραφικής παράστασης. Καθώς το σύστημα εκπαιδεύεται, βλέπουμε πως ο χρονικός περιορισμός, ο οποίος είναι και ο πιο αυστηρός και μειώνει σημαντικά την ανταμοιβή των χρηστών, ικανοποιείται γρήγορα. Στη συνέχεια το σύστημα αναζητά τη βέλτιστη τιμή για τις μεταβλητές ελέγχου ώστε να ελαχιστοποιήσει την καταναλισκόμενη ενέργεια και έτσι να μεγιστοποιήσει το reward. Στη διαδικασία εξερεύνησης και εκμάθησης δοκιμάζει τιμές οι οποίες οδηγούν το σύστημα σε παραβίαση των χρονικών περιορισμών, πράγμα το οποίο είδαμε και παραπάνω στις γραφικές παραστάσεις του split και του χρόνου εκτέλεσης. Αυτό φαίνεται ξεκάθαρα και σε αυτή τη γραφική, καθώς παρατηρούνται αρνητικά "καρφιά" κοντά στη μέση της διαδικασίας εκπαίδευσης. Καθώς το σύστημα εκπαιδεύεται όλο και περισσότερο, οι περιορισμοί ικανοποιούνται όλο και περισσότερο και η ενέργεια που καταναλώνει το σύστημα ολοένα και μειώνεται. Αυτό φαίνεται από την ανοδική πορεία της ανταμοιβής του συστήματος κατά τη διάρκεια της εκπαίδευσης.

5.2.2 Σύγκριση Επιδόσεων

Για να αξιολογήσουμε τις επιδόσεις του συστήματος θα προχωρήσουμε σε μία σειρά από συγκρίσεις. Για αρχή θα αξιολογήσουμε την τεχνική πολλαπλής πρόσβασης RSMA συγκρίνοντας τις επιδόσεις της με την τεχνική NOMA. Στη συνέχεια θα συγκρίνουμε την ικανότητα λήψης απόφασης εκφόρτωσης του συστήματος, συγκρίνοντάς το με ένα σύστημα το οποίο βελτιστοποιεί μόνο τις ισχύς εκπομπής και λαμβάνει μία τυχαία απόφαση εκφόρτωσης. Τέλος, θα συγκρίνουμε το σύστημα με ένα σύστημα το οποίο εκφορτώνει πλήρως τις εργασίες του στον edge server, ενώ και αυτό βελτιστοποιεί τις ισχύς εκπομπής, ώστε να αξιολογήσουμε αν η μερική εκφόρτωση εργασιών είναι οφέλιμη.

Για να συγκρίνουμε την τεχνική RSMA με εκείνη του NOMA θα κατασκευάσουμε ένα νέο σύστημα, το οποίο κάνει χρήση του NOMA και χρησιμοποιεί την ίδια τεχνική βαθιάς ενισχυτικής μάθησης για την εκπαίδευσή του. Το σύστημα αυτό χρησιμοποιεί τις ίδιες παραμέτρους με το σύστημα που μελετήσαμε παραπάνω, όπως αυτές φαίνονται στον πίνακα 5.1. Η διαφορά του νέου συστήματος είναι πως στο NOMA υπάρχει ένα μόνο κανάλι επικοινωνίας με τον σταθμό βάσης. Για τον λόγο αυτό, το action space έχει διαστάσεις $|A_k| = 2$ για κάθε πράκτορα, αφού οι μεταβλητές απόφασης είναι μόνο 2, το power και το split. Αυτό κάνει τα νευρωνικά που χρησιμοποιούνται από τον MADDPG αρκετά απλούστερα, αφού έχουν λιγότερους νευρώνες.

Για τη σύγκριση με το random σύστημα θα κατασκευάσουμε ένα σύστημα ίδιο με το αρχικό, με τη μόνη διαφορά πως η μεταβλητή ελέγχου split θα επιλέγεται τυχαία. Έτσι, δίνουμε τη δυνατότητα στο σύστημα να επιλέγει τις βέλτιστες τιμές ισχύος για τα κανάλια του ώστε να ελαχιστοποιεί την κατανάλωση ενέργειας του συστήματος. Για τη μεταβλητή split θα χρησιμοποιήσουμε δύο προσεγγίσεις. Στην πρώτη, η μεταβλητή θα επιλέγεται από μία ομοιόμορφη κατανομή $\mathcal{U}[0, 1]$, ενώ στη δεύτερη από μία κανονική κατανομή $\mathcal{N}(0.5, 0.25)$.

Τέλος, το σύστημα full offloading θα είναι ίδιο με το random σύστημα με τη διαφορά πως το split θα είναι πάντα ίσο με 1.

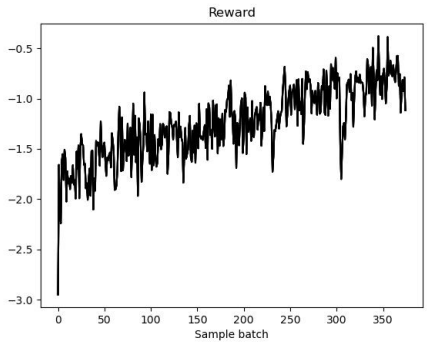
Για να συγκρίνουμε τα συστήματα, πρώτα θα τα εκπαιδεύσουμε με τη διαδικασία που εκπαιδεύσαμε και το αρχικό μας σύστημα και στη συνέχεια θα δοκιμάσουμε τις επιδόσεις σε 100 διαδοχικά επεισόδια. Κατά την επιλογή ενεργειών στην αξιολόγηση απενεργοποιούμε τον προστιθέμενο θόρυβο που χρησιμοποιήσαμε για την εξερεύνηση του χώρου δράσης κατά την εκπαίδευση. Από τα αποτελέσματα θα υπολογίσουμε τον μέσο όρο κάθε μετρικής για όλους τους χρήστες μαζί, ώστε να αξιολογήσουμε τη μέση επίδοση για κάθε χρήστη.

	RSMA	**NOMA**	**Random** ($\mathcal{U}$)	**Random** ($\mathcal{N}$)	**Full Offload**
r^{sec} $(Mbps)$	3.09	2.81	3.07	3.28	3.25
T_{off} (ms)	20.10	24.69	13.14	13.04	24.01
T_{exec} (ms)	29.92	20.36	101.99	104.08	0.00
T_{max} (ms)	37.77	32.15	104.06	105.84	24.01
P (W)	0.167	0.069	0.110	0.221	0.194
Split	0.856	0.902	0.509	0.500	1.00
E_{off} (J)	$3.12 \cdot 10^{-3}$	$1.47 \cdot 10^{-3}$	$1.18 \cdot 10^{-3}$	$2.74 \cdot 10^{-3}$	$4.13 \cdot 10^{-3}$
E_{exec} (J)	$6.46 \cdot 10^{-10}$	$4.40 \cdot 10^{-10}$	$2.20 \cdot 10^{-9}$	$2.25 \cdot 10^{-9}$	0.00
E_{tot} (J)	$3.12 \cdot 10^{-3}$	$1.47 \cdot 10^{-3}$	$1.18 \cdot 10^{-3}$	$2.74 \cdot 10^{-3}$	$4.13 \cdot 10^{-3}$
Reward	-0.990	-0.927	-2.84	-3.31	-1.42

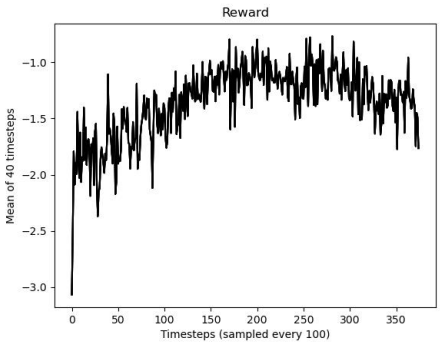
Στον πίνακα 5.2.2 βλέπουμε πως το RSMA επιτυγχάνει μεγαλύτερο secure data rate από το NOMA, ενώ έχει ελαφρώς μεγαλύτερο συνολικό χρόνο εκτέλεσης αλλά και συνολική ενέργεια. Καθώς η συνολική κατανάλωση ενέργειας στο NOMA είναι μικρότερη και οι χρονικές απαιτήσεις ικανοποιούνται σε μεγάλο βαθμό και στις δύο περιπτώσεις προκύπτει λίγο καλύτερο reward για το σύστημα NOMA. Αυτό ίσως οφείλεται στο γεγονός πως τα δίκτυα του NOMA είναι απλούστερα και πιθανόν η εκπαίδευση να γίνεται ταχύτερα από ότι στο σύστημα που χρησιμοποιεί RSMA.

Αυξάνοντας τα επεισόδια εκπαίδευσης των δύο συστημάτων από τα 3000 στα 5000, τα αποτελέσματα υποστηρίζουν αυτή τη θεωρία. Αυτό φαίνεται στον πίνακα 5.2.2, καθώς σε μεγαλύτερη διάρκεια εκπαίδευσης το σύστημα που χρησιμοποιεί RSMA βελτιώνει την επίδοσή του, ενώ το σύστημα NOMA το αντίθετο. Το NOMA, έχοντας απλούστερα νευρωνικά, φτάνει ταχύτερα στη βέλτιστη λύση, ενώ συνεχίζοντας την εκπαίδευση πέφτουμε στην παγίδα του overfitting και η επίδοση του συστήματος χειροτερεύει. Αντίθετα, βλέπουμε πως το πιο πολύπλοκο από άποψη νευρωνικών σύστημα του RSMA βελτιώνει για αρκετές ακόμα εποχές την επίδοσή του. Αυτό φαίνεται και στο σχήμα 5.7 Η χαμηλότερη πολυπλοκότητα και το μικρότερο κόστος στους δέκτες του RSMA επιτρέπει την υλοποίηση συστημάτων με αυτό το πρωτόκολλο πολλαπλής πρόσβασης χωρίς να θυσιάζονται οι επιδόσεις του συστήματος.

	RSMA (5000 episodes)	**NOMA** (5000 episodes)
r^{sec} $(Mbps)$	2.94	2.51
T_{off} (ms)	16.71	33.23
T_{exec} (ms)	37.14	31.91
T_{max} (ms)	42.06	50.30
P (W)	0.090	0.049
Split	0.818	0.845
E_{off} (J)	$1.38 \cdot 10^{-3}$	$1.03 \cdot 10^{-3}$
E_{exec} (J)	$8.02 \cdot 10^{-10}$	$6.89 \cdot 10^{-10}$
E_{tot} (J)	$1.38 \cdot 10^{-3}$	$1.03 \cdot 10^{-3}$
Reward	-0.509	-1.378



(α΄) Το reward του συστήματος RSMA σε 5000 εποχές



(β΄) Το reward του συστήματος NOMA σε 5000 εποχές

Σχήμα 5.7: Σύγκριση των συστημάτων RSMA και NOMA σε μεγαλύτερη διάρκεια εκπαίδευσης

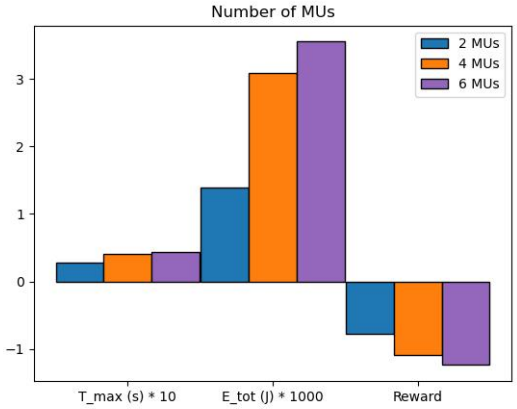
Στα δύο συστήματα που χρησιμοποιούν random processes για να επιλέξουν το ποσοστό εκφόρτωσης παρατηρούμε πως ο χρόνος της τοπικής εκτέλεσης είναι κατά μέσο όρο μεγαλύτερος του χρονικού ορίου που έχουμε θέσει για το σύστημα. Αυτό έχει ως αποτέλεσμα το reward να μειώνεται σημαντικά. Παρατηρείται φυσικά μείωση στην ενέργεια που καταναλώνει το σύστημα, καθώς μικρότερο μέρος του task υποβάλλεται στην ενεργειακά ακριβή διαδικασία της εκφόρτωσης, η οποία όμως δεν είναι αρκετή για να αντισταθμίσει την ποινή από την παραβίαση των χρονικών περιορισμών του συστήματος. Προκύπτει λοιπόν πως η τιμή του ποσοστού εκφόρτωσης που επιλέγει το αρχικό σύστημα είναι καλύτερη από τυχαία.

Τέλος, συγκρίνοντας με την περίπτωση του full offload, βλέπουμε πως έχουμε μεγαλύτερο συνολικό χρόνο επεξεργασίας στην αρχική περίπτωση. Αυτό όμως δεν είναι πρόβλημα, καθώς το σύστημά μας δεν έχει σκοπό την ελαχιστοποίηση του χρόνου επεξεργασίας των εργασιών, παρά μόνο την ικανοποίηση των χρονικών περιορισμών που του θέσαμε. Καθώς και στα δύο συστήματα ικανοποιούνται κατά μέσο όρο οι χρονικοί περιορισμοί, η μετρική που έχει σημασία είναι εκείνη της συνολικής ενέργειας που καταναλώνει το σύστημα. Το αρχικό σύστημα έχει χαμηλότερη ενεργειακή κατανάλωση από το σύστημα full offload, γεγονός που σημαίνει πως οι επιδόσεις του δικαιολογούν την επιλογή μερικής μόνο εκφόρτωσης των εργασιών.

5.2.3 Ανάλυση Υπερπαραμέτρων

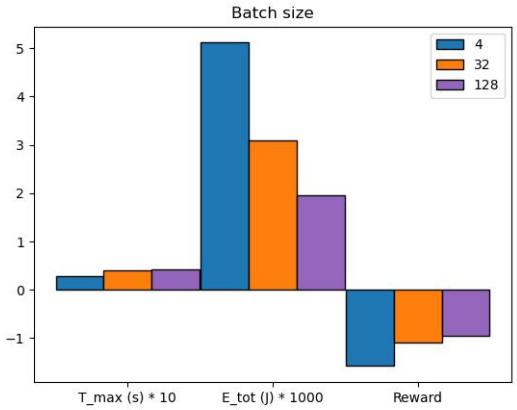
Ένας σημαντικός παράγοντας για την αποτελεσματικότητα της εκπαίδευσης του συστήματος είναι οι υπερπαράμετροι που έχουμε επιλέξει για το σύστημα. Παρακάτω θα αναλύσουμε μερικές από αυτές, καθώς και θα συγκρίνουμε την αποτελεσματικότητα του συστήματος καθώς αυτές μεταβάλλονται αλλά και τις επιπτώσεις που έχει αυτή η μεταβολή στον χρόνο εκπαίδευσης.

Για αρχή θα μελετήσουμε τις επιδόσεις του συστήματος ανάλογα με το πλήθος των MUs σε αυτό. Ακολουθούμε την ίδια διαδικασία με το αρχικό σύστημα και εκπαιδεύουμε συστήματα 2, 4 και 6 χρηστών. Όπως βλέπουμε στο 5.8, καθώς οι χρήστες μέσα στο σύστημα αυξάνονται, τόσο ο χρόνος που απαιτείται συνολικά για την εκτέλεση των εργασιών τους όσο και η ενέργεια που



Σχήμα 5.8: Επιδόσεις συστήματος για διαφορετικό πλήθος χρηστών

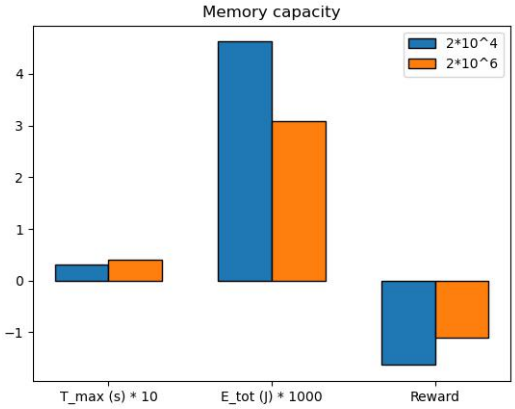
καταναλώνουν για αυτόν τον σκοπό αυξάνονται. Αυτό έχει ως συνέπεια μείωση του reward του συστήματος. Ωστόσο, βλέπουμε πως η αύξηση αυτή στο κόστος δεν είναι πολύ μεγάλη, ειδικά μεταξύ των συστημάτων με τους 4 και τους 6 χρήστες. Αυτή η συμπεριφορά εξηγείται από την ικανότητα του RSMA να κλιμακώνεται αποτελεσματικά.



Σχήμα 5.9: Επιδόσεις συστήματος για διαφορετικό batch size

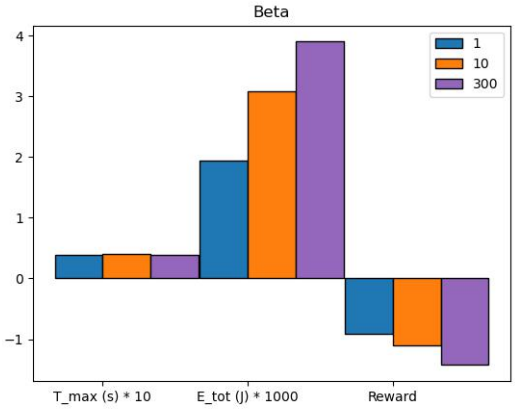
Στο σχήμα 5.9 παρατηρούμε την επίδοση του συστήματος για διαφορετικά μεγέθη batch size. Καθώς το batch size μεγαλώνει, η συνολική ενέργεια που καταναλώνει το σύστημα ελαττώνεται και το reward αυξάνεται. Ωστόσο, η διαφορά στο reward μεταξύ των batch size μεγέθους 32 και 128 είναι πολύ μικρή, ενώ ο χρόνος που απαιτείται για την εκπαίδευση αυξάνεται εξαιρετικά. Για τον λόγο αυτό κρίνουμε πως η βέλτιστη τιμή για το batch size είναι ίση με 32.

Το σύστημα αποθηκεύει τις εμπειρίες στη μνήμη του. Μία μεγάλη μνήμη απαιτεί πολλούς πόρους από τον υπολογιστή στον οποίο θα γίνει η εκπαίδευση. Από την άλλη, μία μικρή μνήμη χάνει γρήγορα τις παλιές εμπειρίες και δείχνει προτίμηση στις πιο πρόσφατες με κίνδυνο να μην εξερευνεί αρκετά τον χώρο δράσης. Παρατηρούμε στο 5.10 πως αυτό συμβαίνει για τη μνήμη μεγέθους $2 \cdot 10^4$, καθώς οι επιδόσεις του συστήματος είναι κατώτερες του αρχικού συστήματος.



Σχήμα 5.10: Επιδόσεις συστήματος για διαφορετικό μέγεθος μνήμης

Η υπερπαράμετρος β ορίζει τον αριθμό των βημάτων πριν την ενημέρωση των actor networks στους MUs. Στο 5.11 συγκρίνεται η ενημέρωση σε κάθε βήμα ($\beta = 1$), η ενημέρωση κάθε 10 βήματα ($\beta = 10$) και η ενημέρωση σε κάθε epoch ($\beta = 300$). Παρατηρούμε πως όσο πιο συχνή η ενημέρωση τόσο καλύτερη η επίδοση του συστήματος. Ωστόσο, η ενημέρωση σε κάθε βήμα εισάγει σημαντικό overhead στην εκπαίδευση του συστήματος και την καθυστερεί σημαντικά. Για τον λόγο αυτό προτιμούμε την τιμή $\beta = 10$ η οποία επιτρέπει και καλά αποτελέσματα αλλά και μικρότερο overhead και ταχύτερη εκπαίδευση.



Σχήμα 5.11: Επιδόσεις συστήματος για διαφορετικό β

6 Σύνοψη - Συμπεράσματα

6.1 Σύνοψη

Σε αυτή την εργασία ερευνήσαμε ένα σύστημα MEC το οποίο χρησιμοποιεί uplink RSMA για την επικοινωνία μεταξύ mobile users και edge server. Διατυπώσαμε ένα πρόβλημα βελτιστοποίησης, το οποίο περιλαμβάνει την απόφαση εκφόρτωσης, το ποσοστό της εργασίας που θα εκφορτωθεί, την ισχύ εκπομπής και τη σειρά αποκωδικοποίησης με σκοπό να ελαχιστοποιήσουμε το κόστος επεξεργασίας των εργασιών των χρηστών ενώ παράλληλα ικανοποιούνται κάποιοι χρονικοί περιορισμοί. Για να αντιμετωπίσουμε αυτό το πρόβλημα, κατασκευάσαμε ένα μοντέλο βαθιάς ενισχυτικής μάθησης, το οποίο εκπαιδεύει ένα πλήθος πρακτόρων που ενεργούν συνεργατικά με τη χρήση του αλγορίθμου MADDPG. Στις ενέργειες τις οποίες αποφασίζει το σύστημα προσθέσαμε θόρυβο που ακολουθεί την κανονική κατανομή ώστε να βελτιώσουμε την ικανότητα εξερεύνησης του συστήματος σε συνεχές action space.

6.2 Συμπεράσματα

Τα αριθμητικά αποτελέσματα που προέκυψαν από πειράματα σε προσομοιωμένο περιβάλλον έδειξαν σαφή βελτίωση της επίδοσης του συστήματος κατά την εκπαίδευση των πρακτόρων. Οι πράκτορες δρουν συνεργατικά ώστε να βελτιώσουν το secure data rate συνολικά και να μειώσουν την ενέργεια που καταναλώνει όλο το σύστημα για την επεξεργασία των εργασιών ικανοποιώντας τους χρονικούς περιορισμούς που τους έχουμε επιβάλει.

Ύστερα από σύγκριση με συστήματα που χρησιμοποιούν NOMA για την επικοινωνία των χρηστών με τον BS παρατηρούμε πως ενώ το σύστημα NOMA εκπαιδεύεται ταχύτερα εξαιτίας των λιγότερο πολύπλοκων νευρωνικών που περιέχει, το RSMA σύστημα έχει καλύτερες δυνατότητες και επιτυγχάνει υψηλότερο secure data rate και χαμηλότερους χρόνους επεξεργασίας, ενώ παράλληλα διατηρεί την ενεργειακή κατανάλωση σε αντίστοιχα με το NOMA επίπεδα.

Συγκρίνοντας το σύστημα με αντίστοιχα συστήματα στα οποία η απόφαση για εκφόρτωση επιλέγεται τυχαία με διάφορες κατανομές, βλέπουμε πως η επίδοση του συστήματος είναι καλύτερη από τυχαία και επομένως δικαιολογεί τη χρήση ενισχυτικής μάθησης για τη λήψη της απόφασης.

Τέλος, ύστερα από σύγκριση με ένα παρόμοιο σύστημα το οποίο όμως εκφορτώνει πάντοτε πλήρως τις εργασίες του, παρατηρούμε μικρότερη ενεργειακή κατανάλωση στο δικό μας σύστημα και επομένως καλύτερες επιδόσεις συνολικά.

Στο τελευταίο βήμα της εργασίας αναλύσαμε τις διάφορες υπερπαραμέτρους του συστήματος και εξηγήσαμε τον αντίκτυπό τους στην επίδοσή του.

6.3 Μελλοντική Δουλειά

Ως μελλοντική δουλειά, μπορεί να δοθεί η δυνατότητα στους κινητούς χρήστες να εκφορτώνουν μέρος της εργασίας τους σε άλλους κινητούς χρήστες με μικρότερο φόρτο εργασίας, εκμεταλλευόμενοι τις κοντινές τους αποστάσεις για μικρότερη κατανάλωση ενέργειας κατά την εκφόρτωση. Η χρήση πολλαπλών edge servers είναι αρκετά συνηθισμένη σε συστήματα MEC και μπορεί και αυτή να αντιμετωπιστεί με αντίστοιχες μεθόδους. Στην παρούσα εργασία θεωρήσαμε αμελητέο τον χρόνο επεξεργασίας της εργασίας στον edge server, αλλά και τον χρόνο που απαιτείται για την επιστροφή του αποτελέσματος στους χρήστες. Το επιπλέον αυτό κόστος μπορεί να μελετηθεί σε κάποια άλλη εργασία. Ενδιαφέρον έχει και η προσπάθεια δικαιότερης κατανομής του ενεργειακού κόστους. Τέλος, μπορεί να γίνει προσπάθεια να αντιμετωπιστεί και το πρόβλημα της σειράς αποκωδικοποίησης με χρήση ενισχυτικής μάθησης.

Βιβλιογραφία

[1] Nasir Abbas, Yan Zhang, Amir Taherkordi, and Tor Skeie. "Mobile edge computing: A survey." English. In: *IEEE Internet of Things Journal* 5.1 (2017), pp. 450–465.

[2] Laha Ale, Ning Zhang, Xiaojie Fang, Xianfu Chen, Shaohua Wu, and Longzhuang Li. "Delay-aware and energy-efficient computation offloading in mobile-edge computing using deep reinforcement learning." English. In: *IEEE Transactions on Cognitive Communications and Networking* 7.3 (2021), pp. 881–892.

[3] Md Shipon Ali, Hina Tabassum, and Ekram Hossain. "Dynamic user clustering and power allocation for uplink and downlink non-orthogonal multiple access (NOMA) systems." English. In: *IEEE access* 4 (2016), pp. 6325–6343.

[4] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. "Deep reinforcement learning: A brief survey." English. In: *IEEE Signal Processing Magazine* 34.6 (2017), pp. 26–38.

[5] Soheila V Bana and Pravin Varaiya. "Space division multiple access (SDMA) for robust ad hoc vehicle communication networks." English. In: *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*. IEEE. 2001, pp. 962–967.

[6] Zhao Chen and Xiaodong Wang. "Decentralized computation offloading for multi-user mobile edge computing: A deep reinforcement learning approach." English. In: *EURASIP Journal on Wireless Communications and Networking* 2020.1 (2020), p. 188.

[7] Zhao Chen, Lei Zhang, Yukui Pei, Chunxiao Jiang, and Liuguo Yin. "NOMA-based multi-user mobile edge computation offloading via cooperative multi-agent deep reinforcement learning." English. In: *IEEE Transactions on Cognitive Communications and Networking* 8.1 (2021), pp. 350–364.

[8] Xiangxiang Chu and Hangjun Ye. "Parameter sharing deep deterministic policy gradient for cooperative multi-agent reinforcement learning." English. In: *arXiv preprint arXiv:1710.00336* (2017).

[9] Linglong Dai, Bichai Wang, Yifei Yuan, Shuangfeng Han, I Chih-Lin, and Zhaocheng Wang. "Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends." English. In: *IEEE Communications Magazine* 53.9 (2015), pp. 74–81.

[10] Paul Dent, Gregory E Bottomley, and T Croft. "Jakes fading model revisited." English. In: *Electronics letters* 13.29 (1993), pp. 1162–1163.

[11] Maria Diamanti, Panagiotis Charatsaris, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. "Incentive mechanism and resource allocation for edge-fog networks driven by multi-dimensional contract and game theories." English. In: *IEEE Open Journal of the Communications Society* 3 (2022), pp. 435–452.

[12] Maria Diamanti, Georgios Kapsalis, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. "Energy-Efficient Rate-Splitting Multiple Access: A Deep Reinforcement Learning-Based Framework." English. In: *IEEE Open Journal of the Communications Society* 4 (2023), pp. 2397–2409. DOI: 10.1109/OJCOMS.2023.3322047.

[13] Maria Diamanti, Christos Pelekis, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. "Delay minimization for rate-splitting multiple access-based multi-server mec offloading." English. In: *IEEE/ACM Transactions on Networking* (2023).

[14] Amal Feriani and Ekram Hossain. "Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: A tutorial." English. In: *IEEE Communications Surveys & Tutorials* 23.2 (2021), pp. 1226–1252.

[15] Taiki Fuji, Kiyoto Ito, Kohsei Matsumoto, and Kazuo Yano. "Deep multi-agent reinforcement learning using dnn-weight evolution to optimize supply chain performance." English. In: (2018).

[16] Ana Galindo-Serrano and Lorenza Giupponi. "Distributed Q-learning for aggregated interference control in cognitive radio networks." English. In: *IEEE Transactions on Vehicular Technology* 59.4 (2010), pp. 1823–1834.

[17] Nguyen Quang Hieu, Dinh Thai Hoang, Dusit Niyato, and Dong In Kim. "Optimal power allocation for rate splitting communications with deep reinforcement learning." English. In: *IEEE wireless communications letters* 10.12 (2021), pp. 2820–2823.

[18] Kenichi Higuchi and Anass Benjebbour. "Non-orthogonal multiple access (NOMA) with successive interference cancellation for future radio access." English. In: *IEICE Transactions on Communications* 98.3 (2015), pp. 403–414.

[19] Jakob Hollenstein, Sayantan Auddy, Matteo Saveriano, Erwan Renaudo, and Justus Piater. "Action noise in off-policy deep reinforcement learning: Impact on exploration and performance." English. In: *arXiv preprint arXiv:2206.03787* (2022).

[20] SM Riazul Islam, Nurilla Avazov, Octavia A Dobre, and Kyung-Sup Kwak. "Power-domain non-orthogonal multiple access (NOMA) in 5G systems: Potentials and challenges." English. In: *IEEE Communications Surveys & Tutorials* 19.2 (2016), pp. 721–742.

[21] Ying Ju, Yuchao Chen, Zhiwei Cao, Lei Liu, Qingqi Pei, Ming Xiao, Kaoru Ota, Mianxiong Dong, and Victor CM Leung. "Joint secure offloading and resource allocation for vehicular edge computing network: A multi-agent deep reinforcement learning approach." English. In: *IEEE Transactions on Intelligent Transportation Systems* (2023).

[22] Mayur Katwe, Keshav Singh, Bruno Clerckx, and Chih-Peng Li. "Rate splitting multiple access for sum-rate maximization in IRS aided uplink communications." English. In: *IEEE Transactions on Wireless Communications* (2022).

[23] Yuxi Li. "Deep reinforcement learning: An overview." English. In: *arXiv preprint arXiv:1701.07274* (2017).

[24] Zhe Li, Pengcheng Chen, Jie Jiang, Bin Lyu, Haiyan Guo, and Zhen Yang. "Sum Computational Bits Maximization for Active RIS-assisted MEC system with RSMA." English. In: *2023 International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE. 2023, pp. 1085–1090.

[25] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. "Continuous control with deep reinforcement learning." English. In: *arXiv preprint arXiv:1509.02971* (2015).

[26] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. "Multi-agent actor-critic for mixed cooperative-competitive environments." English. In: *Advances in neural information processing systems* 30 (2017).

[27] Pavel Mach and Zdenek Becvar. "Mobile edge computing: A survey on architecture and computation offloading." English. In: *IEEE communications surveys & tutorials* 19.3 (2017), pp. 1628–1656.

[28] Hangyu Mao, Zhengchao Zhang, Zhen Xiao, and Zhibo Gong. "Modelling the dynamic joint policy of teammates with attention multi-agent DDPG." English. In: *arXiv preprint arXiv:1811.07029* (2018).

[29] Yijie Mao, Onur Dizdar, Bruno Clerckx, Robert Schober, Petar Popovski, and H. Vincent Poor. "Rate-Splitting Multiple Access: Fundamentals, Survey, and Future Research Trends." English. In: *IEEE Communications Surveys & Tutorials* 24.4 (2022), pp. 2073–2126. ISSN: 2373-745X. DOI: 10.1109/comst.2022.3191937. URL: http://dx.doi.org/10.1109/COMST.2022.3191937.

[30] Laetitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. "Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems." English. In: *The Knowledge Engineering Review* 27.1 (2012), pp. 1–31.

[31] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. "Asynchronous methods for deep reinforcement learning." English. In: *International conference on machine learning*. PMLR. 2016, pp. 1928–1937.

[32] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. "Playing atari with deep reinforcement learning." English. In: *arXiv preprint arXiv:1312.5602* (2013).

[33] Afshin Oroojlooy and Davood Hajinezhad. "A review of cooperative multi-agent deep reinforcement learning." English. In: *Applied Intelligence* 53.11 (2023), pp. 13677–13722.

[34] Flor G Ortiz-Gomez, Daniele Tarchi, Ramón Martínez, Alessandro Vanelli-Coralli, Miguel A Salas-Natera, and Salvador Landeros-Ayala. "Cooperative multi-agent deep reinforcement learning for resource management in full flexible VHTS systems." English. In: *IEEE Transactions on Cognitive Communications and Networking* 8.1 (2021), pp. 335–349.

[35] Aske Plaat. *Deep reinforcement learning*. English. Vol. 10. Springer, 2022.

[36] Bixio Rimoldi and Rüdiger Urbanke. "A rate-splitting approach to the Gaussian multiple-access channel." English. In: *IEEE Transactions on Information Theory* 42.2 (1996), pp. 364–375.

[37] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms." English. In: *arXiv preprint arXiv:1707.06347* (2017).

[38] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. "Successive interference cancellation: A back-of-the-envelope perspective." English. In: *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. 2010, pp. 1–6.

[39] Zhaoyuan Shi, Xianzhong Xie, Huabing Lu, Helin Yang, and Jun Cai. "Deep reinforcement learning based dynamic user access and decode order selection for uplink NOMA system with imperfect SIC." English. In: *IEEE Wireless Communications Letters* 10.4 (2020), pp. 710–714.

[40] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. "Deterministic policy gradient algorithms." English. In: *International conference on machine learning*. Pmlr. 2014, pp. 387–395.

[41] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. English. MIT press, 2018.

[42] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. "Multiagent cooperation and competition with deep reinforcement learning." English. In: *PloS one* 12.4 (2017), e0172395.

[43] Ming Tang and Vincent WS Wong. "Deep reinforcement learning for task offloading in mobile edge computing systems." English. In: *IEEE Transactions on Mobile Computing* 21.6 (2020), pp. 1985–1997.

[44] Sotiris A Tegos, Panagiotis D Diamantoulakis, and George K Karagiannidis. "On the performance of uplink rate-splitting multiple access." English. In: *IEEE Communications Letters* 26.3 (2022), pp. 523–527.

[45] T Thorpe. “Multi-agent reinforcement learning: Independent vs. cooperative agents.” English. PhD thesis. Master’s thesis, Department of Computer Science, Colorado State University, 1997.

[46] Tuyen X Tran and Dario Pompili. “Joint task offloading and resource allocation for multi-server mobile-edge computing networks.” English. In: *IEEE Transactions on Vehicular Technology* 68.1 (2018), pp. 856–868.

[47] Thanh Phung Truong, Nhu-Ngoc Dao, and Sungrae Cho. “HAMEC-RSMA: Enhanced aerial computing systems with rate splitting multiple access.” English. In: *IEEE Access* 10 (2022), pp. 52398–52409.

[48] George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. “Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models.” English. In: *Advances in Neural Information Processing Systems* 30 (2017).

[49] George E Uhlenbeck and Leonard S Ornstein. “On the theory of the Brownian motion.” English. In: *Physical review* 36.5 (1930), p. 823.

[50] Martijn Van Otterlo and Marco Wiering. “Reinforcement learning and markov decision processes.” English. In: *Reinforcement learning: State-of-the-art*. Springer, 2012, pp. 3–42.

[51] Jian Wang, Hongchang Ke, Xuejie Liu, and Hui Wang. “Optimization for computational offloading in multi-access edge computing: A deep reinforcement learning scheme.” English. In: *Computer Networks* 204 (2022), p. 108690.

[52] Christopher JCH Watkins and Peter Dayan. “Q-learning.” English. In: *Machine learning* 8 (1992), pp. 279–292.

[53] Wei Wu, Fuhui Zhou, Rose Qingyang Hu, and Baoyun Wang. “Energy-efficient resource allocation for secure NOMA-enabled mobile edge computing networks.” English. In: *IEEE Transactions on Communications* 68.1 (2019), pp. 493–505.

[54] Yuan Wu, Guangyuan Ji, Tianshun Wang, Liping Qian, Bin Lin, and Xuemin Shen. “Non-orthogonal multiple access assisted secure computation offloading via cooperative jamming.” English. In: *IEEE Transactions on Vehicular Technology* 71.7 (2022), pp. 7751–7768.

[55] Aaron D Wyner. “The wire-tap channel.” English. In: *Bell system technical journal* 54.8 (1975), pp. 1355–1387.

[56] Helin Yang, Zehui Xiong, Jun Zhao, Dusit Niyato, Liang Xiao, and Qingqing Wu. “Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications.” English. In: *IEEE Transactions on Wireless Communications* 20.1 (2020), pp. 375–388.

[57] Zhaohui Yang, Mingzhe Chen, Walid Saad, Wei Xu, and Mohammad Shikh-Bahaei. “Sum-Rate Maximization of Uplink Rate Splitting Multiple Access (RSMA) Communication.” English. In: *IEEE Transactions on Mobile Computing* 21.7 (2022), pp. 2596–2609. DOI: 10.1109/TMC.2020.3037374.

[58] Zhaohui Yang, Mingzhe Chen, Walid Saad, Wei Xu, and Mohammad Shikh-Bahaei. "Sum-rate maximization of uplink rate splitting multiple access (RSMA) communication." English. In: *IEEE Transactions on Mobile Computing* 21.7 (2020), pp. 2596–2609.

[59] Lianqi Zang, Xin Zhang, and Boren Guo. "Federated deep reinforcement learning for online task offloading and resource allocation in WPC-MEC networks." English. In: *IEEE Access* 10 (2022), pp. 9856–9867.

[60] Hongming Zhang and Tianyang Yu. "Taxonomy of reinforcement learning algorithms." English. In: *Deep reinforcement learning: Fundamentals, research and applications* (2020), pp. 125–133.

[61] Zhibo Zhang and Li Li. "Distributed RISs-Aided Secure Wireless Network Under Practical Phase Shift Model: A Deep Reinforcement Learning Approach." English. In: *2022 14th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE. 2022, pp. 1–5.

Γλωσσάριο

Απόδοση	Ξενόγλωσσος όρος
Αίτημα Προγραμματισμού	Scheduling Request
Αναμενόμενη Απόδοση	Expected Return
Ανατροφοδότηση	Feedback
Ανεξάρτητος Μαθητής	Independent Learner (IL)
Ανεπεξέργαστα Δεδομένα	Raw Data
Ανερχόμενη Ζεύξη	Uplink
Ανταγωνιστικό Πρόβλημα	Competitive Problem
Ανταμοιβή	Reward
Αποκέντρωση	Decentralization
Αποτελεσματικότητα Δείγματος	Sample Efficiency
Αποφασίζων	Decision Maker
Απώλεια	Loss
Απώλεια Διαδρομής	Path Loss
Ασφάλεια Φυσικού Επιπέδου	Physical Layer Security (PLS)
Ασφαλής Ρυθμός Δεδομένων	Secure Data Rate
Βαθιά Ενισχυτική Μάθηση	Deep Reinforcement Learning (DRL)
Βαθιά Μάθηση	Deep Learning
Βαθμιδωτή Άνοδος	Gradient Ascent
Βασισμένο Σε Μοντέλο	Model-Based
Βασισμένο Σε Πολιτική	Policy Based
Βασισμένο Σε Τιμή	Value Based
Διαδικασία Απόφασης Markov	Markov Decision Process (MDP)
Διαδοχική Ακύρωση Παρεμβολών	Successive Interference Cancellation (SIC)
Διαμοιρασμός Παραμέτρων	Parameter Sharing
Διαμόρφωση Ακτίνας	Beamforming
Διαχείριση Εφοδιαστικής Αλυσίδας	Supply Chain Management
Δίκτυο Δράστη	Actor Network

Δίκτυο Κριτή	Critic Network
Δίκτυο Στόχος	Target Network
Δράστης-Κριτής	Actor-Critic
Εικονική Πραγματικότητα	Virtual Reality (VR)
Εκμετάλλευση	Exploitation
Εκπομπή	Transmission
Εκπτωτικός Συντελεστής	Discounting Factor
Εκτός Πολιτικής	Off Policy
Εκφόρτωση	Offloading
Εμπειρία	Experience
Ενέργεια	Action
Ενισχυτική Μάθηση	Reinforcement Learning (RL)
Ενισχυτική Μάθηση Πολλαπλών Πρακτόρων	Multi Agent Reinforcement Learning (MARL)
Εντός Πολιτικής	On Policy
Εξασθένιση Μπλοκ	Block Fading
Εξασθένιση Σκιάς	Shadow Fading
Εξασθένιση Rayleigh	Rayleigh Fading
Εξερεύνηση	Exploration
Εξίσωση Bellman	Bellman Equation
Εξυπηρετητής	Server
Εξυπηρετητής Άκρου	Edge Server (ES)
Εξυπηρετητής Νέφους	Cloud Server
Έξυπνα Σπίτια	Smart Homes
Έξυπνη Ανακλαστική Επιφάνεια	Intelligent Reflective Surface (IRS)
Επαυξημένη Πραγματικότητα	Augmented Reality (AR)
Επιβλεπόμενη Μάθηση	Supervised Learning
Επιπλέον Κόστος	Overhead
Επισόδειο	Episode
Εργασία	Task
Εύρος Ζώνης	Bandwidth
Ήπια Ενημέρωση	Soft Update
Ιδιωτική Ροή	Private Stream
Ισχύς Εκπομπής	Transmit Power
Καθυστέρηση	Latency
Κατανομή Πόρων	Resource Allocation
Κατάρα Της Διαστατικότητας	Curse Of Dimensionality
Κατάσταση	State
Κατερχόμενη Ζεύξη	Downlink
Κέρδος Καναλιού	Channel Gain
Κινητή Υπολογιστική Άκρων	Mobile Edge Computing (MEC)

Κινητός Χρήστης	Mobile User
Κοινή Ροή	Common Stream
Κωδικοποίηση	Encoding
Πολιτική	Policy
Πολιτική Στόχος	Target Policy
Προστιθέμενος Λευκός Γκαουσιανός Θόρυβος	Additive White Gaussian Noise (AWGN)
Μεγάλα Δεδομένα	Big Data
Μεμονωμένου Πράκτορα	Single Agent
Μετάβαση	Transition
Μη Ορθογωνική Πολλαπλή Πρόσβαση	Non Orthogonal Multiple Access (NOMA)
Μοντέλο του Jake	Jake's Model
Νεφοποίηση Κινητών Συσκευών	Mobile Cloudification
Νέφος	Cloud
Ορθογωνική Πολλαπλή Πρόσβαση	Orthogonal Multiple Access (OMA)
Περιβάλλον	Environment
Πλειάδα	Tuple
Πλήρως Αποκεντρωμένο	Fully Decentralized
Πλήρως Παρατηρίσιμος Κριτής	Fully Observable Critic
Πλήρως Συγκετρωποιημένο	Fully Centralized
Πλήρως Συνδεδεμένα	Fully Connected
Πράκτορας	Agent
Ποινή	Penalty
Πολλαπλή Διανομή Φυσικού Επιπέδου	Physical Layer Multicasting
Πολλαπλή Είσοδος - Πολλαπλή Έξοδος	Multiple Input - Multiple Output (MIMO)
Πολλαπλή Πρόσβαση	Multiple Access (MA)
Πολλαπλή Πρόσβαση Διαίρεσης Κώδικα	Code Division Multiple Access (CDMA)
Πολλαπλή Πρόσβαση Διαίρεσης Ρυθμού	Rate Splitting Multiple Access (RSMA)
Πολλαπλή Πρόσβαση Διαίρεσης Συχνότητας	Frequency Division Multiple Access (FDMA)
Πολλαπλή Πρόσβαση Διαίρεσης Χρόνου	Time Division Multiple Access (TDMA)
Πολλαπλή Πρόσβαση Διαίρεσης Χώρου	Space Division Multiple Access (SDMA)
Πολλαπλών Πρακτόρων	Multi Agent
Πολυπλεξία Πεδίου Κώδικα	Code Domain Multiplexing
Πολυπλεξία Πεδίου Ισχύος	Power Domain Multiplexing
Ποσοστό Εκφόρτωσης	Splitting Ratio
Πρόβλημα Μικτών Ακεραίων	Mixed Integer Problem
Προσαρμοζόμενος Πράκτορας	Adaptive Agent
Προσέγγιση Συνάρτησης	Fuction Approximation
Προσωρινή Μνήμη Επανάληψης	Replay Buffer
Ροή Δεδομένων	Data Stream

Ρυθμός Μετάδοσης	Throughput
Σειρά Αποκωδικοποίησης	Decoding Order
Σημείο Πρόσβασης	Access Point (AP)
Σκληρή Ενημέρωση	Hard Update
Σταθμός Βάσης	Base Station (BS)
Στασιμότητα	Stationarity
Συγκεντροποιημένη Εκπαίδευση - Αποκεντρωμένη Εκτέλεση	Cenralized Training - Decentralized Execution
Συγκεντροποίηση	Centralization
Συλλογή Δειγμάτων	Sample Batch
Συνάρτηση Ανταμοιβής	Reward Function
Συνάρτηση Μετάβασης	Transition Function
Συνάρτηση Αξίας	Value Function
Συνάρτηση Αξίας Κατάστασης	State Value Function
Συνάρτηση Αξίας Κατάστασης-Ενέργειας	State-Action Value Function
Συνάρτηση Πολιτικής	Policy Function
Συνενώνω	Concatenate
Συνεργατικές Παρεμβολές	Cooperative Jamming
Συνεργατικό Πρόβλημα	Cooperative Problem
Συνολική Εμπειρία	Global Experience
Συσσωρευτική Απόδοση	Cummulative Return
Συντελεστής Αποτελεσματικής Χωρητικότητας	Effective Capacitance Coefficient
Σχεδιαστικός Αλγόριθμος	Planning Algorithm
Σχήμα Κωδικοποίησης Μυστικότητας	Secrecy Encoding Scheme
Τεχνητές Παρεμβολές	Artificial Jamming
Τιμή Στόχος	Target Value
Τοπικός Δράστης	Local Actor
Υπερεκτίμηση	Overestimation
Υποκλοπέας	Eavesdropper
Υπολογιστική Νέφους	Cloud Computing
Φασματική Απόδοση	Spectral Efficiency
Φασματική Πυκνότητα Ισχύος	Power Spectral Density
Φυσικό Επίπεδο	Physical Layer
Χρονικό Βήμα	Timestep
Χρονικό Διάστημα	Time Slot
Χωρίς Μοντέλο	Model-Free
Χώρος Δράσης	Action Space
Χώρος Καταστάσεων	State Space