Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

Κατανεμημένη Ενισχυτική Μάθηση για τη Βέλτιστη Διαχείριση Κατανεμημένων Μπαταριών σε Smart Grids

Διπλωματικη Εργασια

της

ΠΑΠΑΔΟΣΤΕΦΑΝΑΚΗ ΜΑΡΙΑΝΝΑΣ

Επιβλέπων: Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2024

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

Κατανεμημένη Ενισχυτική Μάθηση για τη Βέλτιστη Διαχείριση Κατανεμημένων Μπαταριών σε Smart Grids

Διπλωματικη Εργασια

της

ΠΑΠΑΔΟΣΤΕΦΑΝΑΚΗ ΜΑΡΙΑΝΝΑΣ

Επιβλέπων: Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 5η Μαρτίου 2024.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Συμεών Παπαβασιλείου	Ιωάννα Ρουσσάκη	Βασίλειος Καρυώτης
Καθηγητής Ε.Μ.Π.	Αναπληρώτρια Καθηγήτρια Ε.Μ.Π.	Αναπληρωτής Καθηγητής Ι.Π.

Αθήνα, Μάρτιος 2024

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

(Υπογραφή)

.....
Μαριάννα
Παπαδοστεφανάκη

5 Μαρτίου 2024

Περίληψη

Τα τελευταία χρόνια, η ενσωμάτωση ανανεώσιμων πηγών ενέργειας στα έξυπνα δίκτυα ενέργειας παρουσιάζει αξιοσημείωτη αύξηση. Ωστόσο, η ζήτηση σε ενέργεια πρέπει να είναι πάντα σε ισορροπία με την προσφορά. Για να επιτευχθεί αυτό, εξαιτίας της αβεβαιότητας που διέπει την παραγωγή ενέργειας από ανανεώσιμες πηγές, απαιτείται αυξημένη ποσότητα αποθεμάτων, τα οποία όμως είναι δαπανηρά και αυξάνουν τα λειτουργικά και επενδυτικά κόστη. Μια στρατηγική για τη μείωση των απαιτούμενων αποθεμάτων, είναι η χρήση συστημάτων αποθήκευσης ενέργειας, όπως μπαταρίες. Οι μπαταρίες μπορούν να καταναλώνουν ενέργεια όταν υπάρχει υπερ-παραγωγή από τις ανανεώσιμες πηγές και να την απελευθερώνουν όταν η ζήτηση υπερβαίνει την παραγωγή. Προκειμένου όμως, να αξιοποιούνται στο έπακρο οι δυνατότητες των μπαταριών, είναι απαραίτητος ο βέλτιστος έλεγχος της λειτουργίας τους.

Ο έλεγχος πολλαπλών μπαταριών, λαμβάνοντας υπόψιν τους περιορισμούς καθεμίας εξ αυτών, μπορεί να μοντελοποιηθεί ως ένα πρόβλημα βελτιστοποίησης πολλαπλών σταδίων. Αυτό το είδος προβλημάτων επιλύεται τυπικά μέσω του Model Predictive Control. Στην παρούσα διπλωματική εργασία μελετάμε τον βέλτιστο σε πραγματικό χρόνο έλεγχο πολλαπλών μπαταριών στα έξυπνα δίκτυα, χρησιμοποιώντας ενισχυτική μάθηση πολλαπλών πρακτόρων. Ο έλεγχος γίνεται με κατανεμημένο τρόπο για να αποφευχθούν τα ζητήματα πολυπλοκότητας και δυνατότητας εφαρμογής που ανακύπτουν όταν ένας κεντρικός πράκτορας λαμβάνει αποφάσεις για όλους τους άλλους πράκτορες.

Σε αυτό το έργο λοιπόν, αναπτύσσουμε δύο συστήματα κατανεμημένου ελέγχου μπαταριών, εφαρμόζοντας τον αλγόριθμο MADDPG. Μοντελοποιούμε το πρόβλημα ως Μαρκοβιανή Διαδικασία Απόφασης και αναθέτουμε σε κάθε μπαταρία έναν διαφορετικό πράκτορα για τη λήψη των αποφάσεων. Στο πρώτο σύστημα συνδυάζουμε τον MADDPG με μια παραδοσιακή τεχνική βελτιστοποίησης, τη Lagrangian Decomposition, που διασπά το αρχικό πρόβλημα σε υποπροβλήματα. Η συνεργασία και ο συντονισμός μεταξύ των πρακτόρων επιτυγχάνεται μέσω του κοινού πολλαπλασιαστή Lagrange. Στο δεύτερο σύστημα, εφαρμόζουμε τον αλγόριθμο MADDPG στο συνολικό πρόβλημα, χωρίς να χρησιμοποιήσουμε τη Lagrangian Decomposition. Τα δύο συστήματα συγκρίνονται μεταξύ τους και με τη μέθοδο Model Predictive Control, ως προς την επίδοση και την υπολογιστική τους πολυπλοκότητα.

Λέξεις Κλειδιά

έξυπνα δίκτυα ενέργειας, συστήματα αποθήκευσης ενέργειας με μπαταρίες, κατανεμημένος έλεγχος μπαταριών, κυρτή βελτιστοποίηση, ενισχυτική μάθηση πολλαπλών πρακτόρων, Lagrangian decompositon

Abstract

In recent years, smart grids have been characterized by an increasing penetration of renewable energy sources. However, in smart grids, the energy demand should always be in balance with the supply. To achieve this, due to the uncertainty of renewable energy generation, the need for an increasing amount of reserves emerges. However, the reserves are costly and lead to increased operational and investment costs. One potential strategy to reduce the amount of required reserves is to deploy energy storage systems, such as batteries. The batteries can charge to consume energy when there is overgeneration of energy by the renewable energy sources and discharge to provide extra energy when the demand is higher than the supply. In order to exploit the full potential of the batteries, it is necessary to optimally control them.

Controlling multiple batteries while taking into account the lookahead constraints of each one of them can be modeled as a multi-step optimization problem. This type of problem is commonly solved using Model Predictive Control. In this diploma thesis, we study the optimal real-time control of multiple batteries in smart grids using multi-agent reinforcement learning. The control is distributed among the batteries to avoid the complexity and applicability issues that arise when a single agent takes central decisions for all batteries in the smart grid.

In particular, here we develop two distributed battery control schemes using the MADDPG algorithm. We formulate the problem as a Markov Decision Process and we assign a different decision agent per battery. In the first system, we integrate the MADDPG with a traditional convex optimization technique, Lagrangian Decomposition, that decomposes the original problem into subproblems. Cooperation and coordination among the different agents are achieved via the common Lagrange multiplier. In the second system, we implement the MADDPG algorithm into the original problem without using Lagrangian Decomposition. The two systems are compared with each other as well as with the Model Predictive Control in terms of performance and computational complexity.

Keywords

smart grids, battery energy storage systems, distributed batteries control, convex optimization, multi-agent reinforcement learning, Lagrangian decomposition

Αφιερώνεται στον αδελφό μου, Χρήστο

Ευχαριστίες

Θα ήθελα κατ' αρχάς να ευχαριστήσω τον καθηγητή κύριο Συμεών Παπαβασιλείου που μου έδωσε την ευκαιρία να εκπονήσω τη διπλωματική μου εργασία στο εργαστήριό του και να ασχοληθώ με ένα τόσο ενδιαφέρον ερευνητικό θέμα. Επίσης, οφείλω ένα τεράστιο ευχαριστώ στην επίκουρη καθηγήτρια κυρία Ελένη Στάη για τις συμβουλές, τη συνεχή καθοδήγηση και την πολύτιμη βοήθειά της. Είμαι ευγνώμων για τη συνεργασία μας. Τέλος, ευχαριστώ την οικογένεια και τους φίλους μου για την αμέριστη υποστήριξή τους καθ' όλη τη διάρκεια των σπουδών μου.

Αθήνα, Μάρτιος 2024

Μαριάννα Παπαδοστεφανάκη

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Μια από τις μεγαλύτερες προκλήσεις με την οποία έρχεται σήμερα αντιμέτωπη η ανθρωπότητα είναι η κλιματική αλλαγή. Τα τελευταία χρόνια η θερμοκρασία της γης αυξάνεται με πρωτοφανείς ρυθμούς, με αποτέλεσμα να εκδηλώνονται ακραία καιρικά φαινόμενα που διαταράσσουν την ισορροπία των οικοσυστημάτων και απειλούν τη βιωσιμότητα των κοινωνιών. Βασικότερη αιτία της υπερθέρμανσης του πλανήτη είναι οι αυξημένες εκπομπές αερίων του θερμοκηπίου και κυρίως του διοξειδίου του άνθρακα (CO_2), το οποίο παράγεται ως επί το πλείστον από την καύση ορυκτών καυσίμων για την παραγωγή ηλεκτρικής ενέργειας.

Η αντιμετώπιση της κλιματικής κρίσης αποτελεί επιτακτική ανάγκη και απαιτεί έναν θεμελιώδη ανασχηματισμό του τρόπου παραγωγής και διαχείρισης της ενέργειας, με γνώμονα την απεξάρτηση από τα ρυπογόνα ορυκτά καύσιμα. Με τη Συμφωνία του Παρισιού, που υπογράφηκε το 2015 από τα κράτη μέλη της Ευρωπαϊκής Ένωσης και άλλα 196 κράτη, ξεκινά και επίσημα μια οικουμενική προσπάθεια αναχαίτισης της ανόδου της θερμοκρασίας, με μακροπρόθεσμο στόχο τη σταθεροποίησή της. Μια από τις βασικές δεσμεύσεις που θέτει η συμφωνία αυτή είναι ο σχεδιασμός στρατηγικών για τη μείωση των εκπομπών αερίων του θερμοκηπίου.

Οι Ανανεώσιμες Πηγές Ενέργειας (ΑΠΕ) και οι σύγχρονες τεχνολογίες κατέχουν κεντρικό ρόλο σε αυτή την προσπάθεια. Οι ΑΠΕ αποτελούν καθαρές και βιώσιμες πηγές ενέργειας, οπότε η πλήρης κάλυψη των ενεργειακών μας αναγκών από αυτές μπορεί να συμβάλλει καθοριστικά στην επίτευξη των στόχων βιωσιμότητας. Παράλληλα, οι αλματώδεις τεχνολογικές εξελίξεις συντελούν στην ανάπτυξη Έξυπνων Δικτύων ηλεκτρικής ενέργειας (Smart Grids), τα οποία αναμένεται να διαδραματίσουν σημαντικό ρόλο στην ενεργειακή μετάβαση και την αποδοτική αξιοποίηση των ΑΠΕ και της παραγόμενης ενέργειας. Ωστόσο, η ενσωμάτωση των ΑΠΕ στα ηλεκτρικά δίκτυα είναι απαιτητική όσον αφορά τη διαχείριση της παραγόμενης ενέργειας και τη διασφάλιση της ομαλής λειτουργίας του δικτύου.

1.1 Έξυπνα Δίκτυα και Συστήματα Αποθήκευσης Ενέργειας

Ένα δίκτυο ηλεκτρικής ενέργειας χαρακτηρίζεται έξυπνο (smart grid) όταν συνδυάζει την βασική υλικοτεχνική υποδομή ενός παραδοσιακού δικτύου με προηγμένες τεχνολογίες πληροφορικής και επικοινωνιών, προκειμένου να βελτιστοποιήσει τις διαδικασίες της πα-

ραγωγής, της διανομής και της κατανάλωσης ηλεκτρικής ενέργειας. Ένα έξυπνο δίκτυο περιλαμβάνει τη χρήση λογισμικού και τεχνολογίες όπως έξυπνους μετρητές, αισθητήρες και έξυπνα συστήματα αυτοματισμού, μέσω των οποίων αυτοματοποιείται ο έλεγχος της παραγωγής και η διαχείριση του φορτίου, ανταλλάσσονται δεδομένα μεταξύ των συνιστωσών του δικτύου και ευνοείται η ορθολογική διαχείριση της ενέργειας από την πλευρά των καταναλωτών. Επιπλέον, χρησιμοποιούνται τεχνολογίες αμφίδρομης ροής πληροφοριών που δίνουν τη δυνατότητα στους καταναλωτές να συμμετέχουν ενεργά ως παραγωγοί ενέργειας.

Οι ψηφιακές τεχνολογίες των έξυπνων δικτύων βελτιώνουν την αποδοτικότητα και την ασφάλεια των συστημάτων ηλεκτρικής ενέργειας και δημιουργούν το κατάλληλο πλαίσιο για την πλήρη ενσωμάτωση των ΑΠΕ στον τομέα της παραγωγής. Ωστόσο, ΑΠΕ όπως η ηλιακή και η αιολική εξαρτώνται σημαντικά από τις καιρικές συνθήκες. Αυτό σημαίνει ότι η παραγωγή τους παρουσιάζει διακυμάνσεις και ενδεχομένως να μην επαρκεί πάντα για να καλύψει τη ζήτηση. Το γεγονός αυτό, σε συνδυασμό με την αδυναμία ακριβούς πρόβλεψης τόσο της προσφοράς όσο και της ζήτησης, καθιστά ένα τέτοιο δίκτυο ασταθές, υπονομεύοντας την αξιοπιστία του.

Μια ελκυστική λύση σε αυτό το πρόβλημα είναι τα συστήματα αποθήκευσης ενέργειας και συγκεκριμένα οι μπαταρίες. Τα συστήματα αυτά έχουν τη δυνατότητα να αποθηκεύουν το πλεόνασμα της ηλεκτρικής ενέργειας που έχει παραχθεί και να το διοχετεύουν στο δίκτυο όταν υπάρχει έλλειψη. Έτσι, παρέχουν στο δίκτυο την ικανότητα να ανταποκρίνεται άμεσα στις μεταβαλλόμενες ενεργειακές ανάγκες και να εξισορροπεί τυχόν αποκλίσεις στο ισοζύγιο ισχύος. Επομένως, τα συστήματα αποθήκευσης μπορούν να συνδράμουν καθοριστικά στην αποτελεσματική ενσωμάτωση των ΑΠΕ στα έξυπνα δίκτυα, διατηρώντας την αξιοπιστία και ευνοώντας την ευελιξία τους.

Μεταξύ των μέσων αποθήκευσης ενέργειας, οι μπαταρίες φαίνεται να παρουσιάζουν σημαντικά πλεονεκτήματα. Κατ' αρχάς, μπορούν να ανταποκρίνονται άμεσα στις μεταβολές των ενεργειακών αναγκών, παρέχοντας την ενέργεια που απαιτείται χωρίς καθυστερήσεις. Επίσης, σε ένα δίκτυο μπορούν να προστεθούν όσες μπαταρίες χρειάζονται, ώστε να αυξηθεί η χωρητικότητά του στα επιθυμητά επίπεδα. Ένα επιπλέον πλεονέκτημα των μπαταριών είναι ότι η αποδοτικότητά τους αυξάνεται με την πάροδο των ετών, ενώ το κόστος για την εγκατάσταση και τη λειτουργία τους παρουσιάζει μείωση και αναμένεται συνεχίσει να μειώνεται τα επόμενα χρόνια [1].

Ωστόσο, κάθε μπαταρία έχει διαφορετικά φυσικά χαρακτηριστικά και ιδιότητες, όπως είναι η μέγιστη ισχύς φόρτισης/εκφόρτισης και η μέγιστη και η ελάχιστη Κατάσταση Ενέργειας (State of Energy - SoE), που αναφέρεται στο ποσοστό της ενέργειας που περιέχεται στην μπαταρία σε σχέση με τη μέγιστη χωρητικότητά της. Επομένως, είναι απαραίτητη η δημιουργία ευέλικτων και αποδοτικών αλγορίθμων που θα ελέγχουν τη συμπεριφορά των μπαταριών ενός δικτύου σύμφωνα με τις τεχνικές προδιαγραφές και τις δυνατότητές τους.

1.2 Σχετική Βιβλιογραφία

Το πρόβλημα του κατανεμημένου ελέγχου των μπαταριών ενός συστήματος διανομής με ανανεώσιμες πηγές ενέργειας αποτελεί αντικείμενο εκτενούς μελέτης τα τελευταία χρόνια, στο πλαίσιο της οποίας αναδεικνύεται η συμβολή της μηχανικής μάθησης στον τομέα αυτό,

έναντι παραδοσιακών τεχνικών που βασίζονται στη βελτιστοποίηση, όπως η Model Predictive Control. Στο έργο [2], αναπτύχθηκε ένα μοντέλο ελέγχου πραγματικού-χρόνου, που χρησιμοποιεί τον αλγόριθμο ενισχυτικής μάθησης Deep Deterministic Policy Gradients (DDPG). Ωστόσο, η μέθοδος αυτή ήταν συγκεντρωτική και συνεπώς καταλληλότερη για τον έλεγχο μίας μόνο μπαταρίας και όχι πολλαπλών.

Για τον έλεγχο πολλαπλών μπαταριών, οι κατανεμημένες μέθοδοι ενισχυτικής μάθησης όπως η ενισχυτική μάθηση πολλαπλών πρακτόρων (Multi-Agent Reinforcement Learning - MARL) επιστρατεύονται με στόχο να βελτιώσουν τη δυνατότητα κλιμάκωσης. Στο [3] το ζήτημα της κλιμάκωσης που προκύπτει κατά την ταυτόχρονη μάθηση πολλαπλών ανεξάρτητων πρακτόρων εντός ενός μερικώς παρατηρήσιμου στοχαστικού περιβάλλοντος, αντιμετωπίζεται μέσω του συνδυασμού της MARL και της μάθησης από off-line βελτιστοποιήσεις που βασίζονται σε δεδομένα που έχουν συλλεγεί εκ των προτέρων και αποτυπώνουν την εμπειρία των πρακτόρων. Στην εν λόγω μελέτη, αντιπαρατίθενται η κατανεμημένη (distributed) και η συγκεντρωτική (centralized) εκπαίδευση και αναδεικνύονται τα πλεονεκτήματα της κατανεμημένης. Συγκεκριμένα, οι πράκτορες είναι ανεξάρτητοι μεταξύ τους και ο καθένας ενημερώνει τη δική του Q-function μέσω ενός πίνακα σταθερού μεγέθους, λαμβάνοντας υπόψιν μόνο τις δικές τους πληροφορίες, χωρίς να επικοινωνεί με τους υπόλοιπους πράκτορες.

Ωστόσο, η πραγματοποίηση της εκπαίδευσης και του ελέγχου με κατανεμημένο τρόπο, με τους πράκτορες να λαμβάνουν αποφάσεις βασιζόμενοι αποκλειστικά στις παρατηρήσεις από το δικό τους ατομικό περιβάλλον, μπορεί να δυσκολέψει την επίτευξη του στόχου για μια καθολικά βέλτιστη λύση. Επομένως, είναι σημαντικό να εξισορροπήσουμε τα οφέλη της αυτονομίας με την ανάγκη για συνεργασία και συντονισμένη δράση. Ένα επιπλέον πρόβλημα, κυρίως των Q-learning τεχνικών, που προκύπτει κατά τη δράση πολλαπλών πρακτόρων στο ίδιο περιβάλλον, είναι η αστάθεια που παρουσιάζει το περιβάλλον από την οπτική γωνία του κάθε πράκτορα. Στο [4] προτείνεται μια μέθοδος για την αντιμετώπιση αυτών των προβλημάτων, η οποία επεκτείνοντας τις μεθόδους δράστη-κριτή (actor-critic), συνδυάζει τη συγκεντρωτική εκπαίδευση με την αποκεντρωμένη εκτέλεση (Centralized Training with Decentralized Execution). Στο έργο αυτό, αναπτύσσεται ο αλγόριθμος Multi-Agent Deep Deterministic Policy Gradients (MADDPG) ο οποίος επιτρέπει στο δίκτυο κριτή να έχει πρόσβαση στις πληροφορίες όλων των πρακτόρων, ενώ το δίκτυο δράστη λαμβάνει μόνο τις τοπικές πληροφορίες κάθε πράκτορα.

Στο [5] εξετάζονται τρεις διαφορετικές προσεγγίσεις ενισχυτικής μάθησης πολλαπλών πρακτόρων για την διαχείριση της ενέργειας που προέρχεται από κατανεμημένες πηγές, συμπεριλαμβανομένων των συστημάτων αποθήκευσης, στο πλαίσιο της αδιαμεσολάβητης ανταλλαγής και αγοραπωλησίας ενέργειας μεταξύ παραγωγών και καταναλωτών. Διαπιστώνεται ότι η τεχνική Centralized Training with Decentralized Execution υπερτερεί έναντι των άλλων. Συγκεκριμένα με την εφαρμογή του αλγορίθμου MADDPG σε συνδυασμό με την τεχνική Διαμοιρασμού Παραμέτρων (Parameter Sharing), επιταχύνεται η εκπαίδευση του μοντέλου, βελτιώνεται η δυνατότητα γενίκευσής του και μειώνεται το υπολογιστικό κόστος, χωρίς ωστόσο να λύνεται το πρόβλημα του "curse of dimensionality".

Στο προηγούμενο του παρόντος ερευνητικό έργο [6], ο έλεγχος των μπαταριών ενός δικτύου διανομής μοντελοποιείται ως πρόβλημα κυρτής βελτιστοποίησης και επιλύεται με κατανεμημένο τρόπο με χρήση ενισχυτικής μάθησης. Συγκεκριμένα, μέσω της μεθόδου La-

grangian Decomposition, το αρχικό πρόβλημα αποσυντίθεται σε υποπροβλήματα, ένα για κάθε μπαταρία και ένα για το DSO. Τον έλεγχο κάθε μπαταρίας αναλαμβάνει ένας διαφορετικός πράκτορας, ο οποίος εκπαιδεύεται off-line και ανεξάρτητα από τους άλλους μέσω της μεθόδου Q-Learning [7]. Η εκπαίδευση όλων των πρακτόρων πραγματοποιείται παράλληλα, θεωρώντας καθορισμένα διακριτά σύνολα καταστάσεων και ενεργειών και η επικοινωνία μεταξύ των πρακτόρων επιτυγχάνεται μέσω του κοινού πολλαπλασιαστή Lagrange. Επίσης, η εκπαίδευση γίνεται σε ένα συγκεκριμένο σύνολο διακριτών τιμών για τους πολλαπλασιαστές Lagrange που αποκτάται μέσω μιας τεχνικής βελτιστοποίησης που εκτελείται off-line.

1.3 Αντικείμενο της διπλωματικής

Στην παρούσα διπλωματική εργασία θα δημιουργήσουμε δύο συστήματα για τον κατανεμημένο έλεγχο των μπαταριών ενός έξυπνου δικτύου διανομής ηλεκτρικής ενέργειας, λαμβάνοντας υπόψιν τους περιορισμούς κάθε μπαταρίας για την κατάσταση της ενέργειάς της, τόσο για την τρέχουσα χρονική στιγμή όσο και μελλοντικά, καθώς και τα επιτρεπόμενα όρια ισχύος φόρτισης/εκφόρτισης της. Συγκεκριμένα, θα υλοποιήσουμε τον αλγόριθμο MADDPG που παρουσιάζεται εδώ [4]. Πρόκειται για έναν αλγόριθμο ενισχυτικής μάθησης πολλαπλών πρακτόρων που αξιοποιεί τις δυνατότητες των νευρωνικών δικτύων για την εκμάθηση μιας εξατομικευμένης για κάθε πράκτορα πολιτικής. Η πολιτική που προκύπτει για κάθε πράκτορα έχει διαμορφωθεί με βάση την εμπειρία όλων των πρακτόρων, αλλά κατά την εφαρμογή της ο πράκτορας την τροφοδοτεί αποκλειστικά με τις δικές του πληροφορίες.

Θα αναθέσουμε τον έλεγχο κάθε μπαταρίας σε έναν διαφορετικό πράκτορα, ο οποίος θα παίρνει αποφάσεις για την ισχύ φόρτισης/εκφόρτισης της λαμβάνοντας υπόψιν τα χαρακτηριστικά της. Επίσης, οι πράκτορες των μπαταριών θα ανταλλάσσουν πληροφορίες με τον πράκτορα του Distributed System Operator (DSO) ο οποίος διαχειρίζεται αυτόνομα την ισχύ στο Point of Common Coupling (PCC), δηλαδή στο σημείο σύνδεσης του δικτύου διανομής με το δίκτυο μεταφοράς.

Στο πρώτο σύστημα, θα ακολουθήσουμε την τεχνική που ακολουθείται στο έργο [6] όσον αφορά στην χρήση της Lagrangian Decomposition για τον διαχωρισμό του προβλήματος σε διαφορετικά υποπροβλήματα, αλλά αντί για τη μέθοδο Q-Learning, θα εφαρμόσουμε τον MADDPG, το οποίο εκτός των άλλων, θα επιτρέψει την online εκπαίδευση των πρακτόρων, την μεταξύ τους αλληλεπίδραση και τη χρήση συνεχών συνόλων καταστάσεων, δράσεων και τιμών για τους πολλαπλασιαστές Lagrange κατά την εκπαίδευση. Θα αποδείξουμε ότι το πρόβλημα του DSO μπορεί να λυθεί ανεξάρτητα για κάθε χρονική στιγμή και θα επιλύσουμε το πρόβλημα των μπαταριών με τον MADDPG. Στο δεύτερο σύστημα, δε θα χρησιμοποιήσουμε τη μέθοδο Lagrangian Decomposition, αλλά θα εφαρμόσουμε τον MADDPG προκειμένου να λύσουμε το συνολικό πρόβλημα, εκμεταλλευόμενοι το γεγονός ότι οι μπαταρίες έχουν την πλήρη εποπτεία του συστήματος κατά την εκπαίδευσή τους.

Τέλος, θα εξετάσουμε την αποδοτικότητά των δύο συστημάτων με κριτήρια την κάλυψη του ενεργειακού ισοζυγίου ισχύος, τον σεβασμό των περιορισμών των μπαταριών, τον χρόνο απόκρισης, την υπολογιστική πολυπλοκότητα και την σύγκρισή τους με μια κλασική μέθοδο αντιμετώπισης τέτοιου είδους προβλημάτων, την Model Predictive Control.

1.4 Οργάνωση του τόμου

Η εργασία αυτή είναι οργανωμένη σε τέσσερα κεφάλαια:

- Στο Κεφάλαιο 2 δίνεται το απαραίτητο θεωρητικό υπόβαθρο των τεχνολογιών που χρησιμοποιήθηκαν για τη διπλωματική. Συγκεκριμένα, γίνεται αναφορά στα προβλήματα βελτιστοποίησης, τη μηχανική μάθηση, τα νευρωνικά δίκτυα, τις Μαρκοβιανές Διαδικασίες Απόφασης, την ενισχυτική μάθηση, τον αλγόριθμο MADDPG και τη Model Predictive Control.
- Στο Κεφάλαιο 3 περιγράφεται η μεθοδολογία και οι αλγόριθμοι που εφαρμόστηκαν.
- Στο Κεφάλαιο 4, αρχικά αναλύονται οι τεχνικές λεπτομέρειες και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση των συστημάτων και στη συνέχεια, παρουσιάζονται τα πειράματα που έγιναν και τα αποτελέσματα που προέκυψαν.
- Στο Κεφάλαιο 5, συνοψίζονται τα αποτελέσματα και δίνονται τα τελικά συμπεράσματα όσον αφορά τη συνεισφορά αυτής της διπλωματικής εργασίας, καθώς και πιθανές μελλοντικές επεκτάσεις.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό παρουσιάζεται το απαραίτητο θεωρητικό υπόβαθρο των τεχνολογιών που χρησιμοποιήθηκαν για την εκπόνηση της εργασίας. Στην πρώτη ενότητα γίνεται αναφορά στο πεδίο της Μαθηματικής Βελτιστοποίησης και σε δύο θεμελιώδεις τεχνικές που χρησιμοποιούνται στο πεδίο αυτό, την Lagrangian Relaxation και την Lagrangian Decomposition. Στη δεύτερη ενότητα γίνεται μια εισαγωγή στη Μηχανική Μάθηση (Machine Learning) και στην τρίτη καλύπτονται βασικές έννοιες στον τομέα των Νευρωνικών Δικτύων (Neural Networks). Στην τέταρτη ενότητα περιγράφονται οι Μαρκοβιανές Διαδικασίες Απόφασης (Markov Decision Processes). Στην πέμπτη και την έκτη ενότητα αναλύονται η Ενισχυτική Μάθηση (Reinforcement Learning) και η Ενισχυτική Μάθηση Πολλαπλών Πρακτόρων (Multi-Agent Reinforcement Learning), αντίστοιχα. Τέλος, στην έβδομη ενότητα περιγράφεται ο αλγόριθμος MADDPG και στην όγδοη η μέθοδος Model Predictive Condtrol (MPC).

2.1 Μαθηματική Βελτιστοποίηση

Το πεδίο της Μαθηματικής Βελτιστοποίησης αφορά προβλήματα για τα οποία αναζητείται η βέλτιστη δυνατή λύση υπό δεδομένους περιορισμούς και πεπερασμένους πόρους [8]. Τα προβλήματα αυτά μπορεί να έχουν πολλές διαφορετικές εφικτές λύσεις και το ζητούμενο είναι να βρεθεί η βέλτιστη εξ αυτών.

Ένα πρόβλημα βελτιστοποίησης αναπαρίσταται από ένα μαθηματικό μοντέλο που περιλαμβάνει τρία βασικά στοιχεία: τις μεταβλητές απόφασης (decision variables), δηλαδή τους αγνώστους που πρέπει να καθοριστούν, τους περιορισμούς (constraints), που καθορίζουν τις επιτρεπτές ή εφικτές τιμές των μεταβλητών απόφασης και την αντικειμενική συνάρτηση (objective function), που αποτελεί συνάρτηση των μεταβλητών απόφασης και είναι το μέτρο της επίδοσης του μοντέλου. Στόχος είναι η βελτιστοποίηση του μαθηματικού μοντέλου, δηλαδή, η εύρεση των τιμών των μεταβλητών απόφασης για τις οποίες ελαχιστοποιείται ή μεγιστοποιείται η τιμή της αντικειμενικής συνάρτησης, χωρίς να παραβιάζονται οι περιορισμοί.

Η γενική μαθηματική διατύπωση ενός προβλήματος ελαχιστοποίησης είναι η εξής:

$$\begin{aligned} &\min_{x} f(x) \\ \text{s.t. } &g_i(x) \leq 0, \quad i \in \{1, \ldots, p\} \\ &h_j(x) = 0, \quad j \in \{1, \ldots, q\} \end{aligned} \tag{2.1}$$

όπου $x = (x_1, x_2, \ldots, x_n)$ είναι οι μεταβλητές απόφασης, $f(x)$ η αντικειμενική συνάρτηση και $g_i(x) \leq 0$, $h_j(x) = 0$ οι περιορισμοί.

Αν οι συναρτήσεις $f(x)$ και $g_i(x)$ είναι γραμμικές, τότε πρόκειται για πρόβλημα γραμμικής βελτιστοποίησης, ενώ αν είναι μη γραμμικές, πρόκειται για πρόβλημα μη γραμμικής βελτιστοποίησης. Επίσης, μια σημαντική κατηγορία προβλημάτων βελτιστοποίησης είναι τα Προβλήματα Κυρτής Βελτιστοποίησης, (Convex Optimization Problems) [9], στα οποία οι συναρτήσεις $f(x)$ και $g_i(x)$ είναι κυρτές και οι $h_j(x)$ γραμμικές.

Όταν ένα πρόβλημα βελτιστοποίησης είναι σύνθετο και περιέχει περίπλοκους περιορισμούς, η επίλυσή του έχει υψηλό υπολογιστικό κόστος και είναι αρκετά χρονοβόρα. Δύο τεχνικές που χρησιμοποιούνται ευρέως για τη διαχείριση τέτοιου είδους προβλημάτων είναι η Lagrangian Relaxation και η Lagrangian Decomposition.

2.1.1 Lagrangian Relaxation

Η Lagrangian Relaxation [9] είναι μια τεχνική που καθιστά ευκολότερη την επίλυση ενός προβλήματος βελτιστοποίησης, επιτρέποντας την χαλάρωση περιορισμών που συνήθως εμπλέκουν πολλές διαφορετικές μεταβλητές, μέσω της ενσωμάτωσής τους στην αντικειμενική συνάρτηση. Για κάθε τέτοιο περιορισμό εισάγεται μια νέα μεταβλητή, γνωστή ως πολλαπλασιαστής Lagrange (Lagrange multiplier), η οποία αποτελεί το βάρος με το οποίο ο αντίστοιχος περιορισμός προστίθεται στην αντικειμενική συνάρτηση, επιβάλλοντας ένα πρόσθετο κόστος ως «τιμωρία». Έτσι, δημιουργείται μια απλούστερη μορφή του αρχικού προβλήματος, πιο ευέλικτη όσον αφορά την ικανοποίηση πολύπλοκων περιορισμών και συνεπώς, πιο εύκολα επιλύσιμη. Ωστόσο, η λύση που προκύπτει αποτελεί προσέγγιση της λύσης του αρχικού προβλήματος.

Για το προηγούμενο πρόβλημα ελαχιστοποίησης, η αντικειμενική συνάρτηση που προκύπτει με εφαρμογή της Lagrangian Relaxation σε όλους τους περιορισμούς είναι η εξής:

$$L = (x, \lambda, \mu) = f(x) + \sum_{i=1}^{p} \lambda_i g_i(x) + \sum_{j=1}^{q} \mu_j h_j(x) \tag{2.2}$$

όπου λ_i, μ_j οι πολλαπλασιαστές Lagrange για τους περιορισμούς ανισότητας και ισότητας, αντίστοιχα.

2.1.2 Lagrangian Decomposition

Η Lagrangian Decomposition [9] είναι μια τεχνική διάσπασης ενός σύνθετου προβλήματος βελτιστοποίησης σε υποπροβλήματα μικρότερης υπολογιστικής πολυπλοκότητας σε σχέση με το αρχικό πρόβλημα. Γενικά κάθε υποπρόβλημα αντιστοιχεί σε ένα υποσύνολο των μεταβλητών και των περιορισμών του αρχικού προβλήματος και επιλύεται ανεξάρτητα από τα

υπόλοιπα. Οι λύσεις των υποπροβλημάτων τελικά συνδυάζονται κατάλληλα για να δώσουν τη λύση του συνολικού προβλήματος. Σε κάθε υποπρόβλημα ανατίθενται συγκεκριμένοι πολλαπλασιαστές Lagrange, οι οποίοι αποτελούν το μέσο «επικοινωνίας» των υποπροβλημάτων, ώστε να μπορούν να συντονίζουν τις λύσεις τους με γνώμονα την επίλυση του συνολικού προβλήματος.

Έστω το ακόλουθο πρόβλημα ελαχιστοποίησης:

$$\begin{aligned} &\min_{x_1,\dots,x_n} \sum_{p=1}^{n} f_p(x_p) \\ \text{s.t. } & g_p(x_p) \leq 0, \quad p \in \{1,\dots,n\} \\ & h_p(x_p) = 0, \quad p \in \{1,\dots,n\} \\ & \sum_{p=1}^{n} g_{c,p}(x_p) \leq 0 \\ & \sum_{p=1}^{n} h_{c,p}(x_p) = 0 \end{aligned} \tag{2.3}$$

Με εφαρμογή της Lagrangian Decomposition προκύπτει:

$$\begin{aligned} &\min_{x_1,\dots,x_n} \sum_{p=1}^{n} f_p(x_p) + \bar{\lambda}_c^T \sum_{p=1}^{n} g_{c,p}(x_p) + \bar{\mu}_c^T \sum_{p=1}^{n} h_{c,p}(x_p) \\ \text{s.t. } & g_p(x_p) \leq 0, \quad p \in \{1,\dots,n\} \\ & h_p(x_p) = 0, \quad p \in \{1,\dots,p\} \end{aligned} \tag{2.4}$$

Έτσι το πρόβλημα μπορεί να διασπαστεί n υποπροβλήματα p της μορφής:

$$\begin{aligned} &\min_{x_1,\dots,x_n} f_p(x_p) + \bar{\lambda}_c^T g_{c,p}(x_p) + \bar{\mu}_c^T h_{c,p}(x_p) \\ \text{s.t. } & g_p(x_p) \leq 0 \\ & h_p(x_p) = 0 \end{aligned} \tag{2.5}$$

2.2 Μηχανική Μάθηση

Η Μηχανική Μάθηση αποτελεί έναν κλάδο της Τεχνητής Νοημοσύνης που εστιάζει στη μελέτη και ανάπτυξη αλγορίθμων με την ικανότητα να «μαθαίνουν» από ένα σύνολο πειραματικών δεδομένων και να δημιουργούν μοντέλα που κάνουν προβλέψεις ή λαμβάνουν αποφάσεις με βάση τα δεδομένα αυτά. Ο Arthur Samuel, πρωτοπόρος της τεχνητής νοημοσύνης, όρισε το 1959 τη μηχανική μάθηση ως «το πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν χωρίς να έχουν προγραμματιστεί ρητά» [10].

Η έννοια της μάθησης ενός αλγορίθμου συνίσταται στη βελτίωση της απόδοσής του στην εργασία που του έχει ανατεθεί, αξιοποιώντας την πρότερη γνώση και εμπειρία του, χωρίς να απαιτείται εκ νέου προγραμματισμός του. Οι διαδικασίες μηχανικής μάθησης συνήθως κατατάσσονται σε τρεις κατηγορίες, οι οποίες καθορίζονται κατά κύριο λόγο από το είδος της ανατροφοδότησης που συνοδεύει την είσοδο και είναι η Επιβλεπόμενη Μάθηση, η Μη

Επιβλεπόμενη Μάθηση και η Ενισχυτική Μάθηση.

Επιβλεπόμενη Μάθηση

Στην Επιβλεπόμενη Μάθηση (Supervised Learning) ο αλγόριθμος, κατά την εκπαίδευσή του, δέχεται ένα δείγμα ζευγών εισόδου-εξόδου (σύνολο εκπαίδευσης) και προσπαθεί να μάθει μια συνάρτηση, η οποία αντιστοιχίζει τις εισόδους στις εξόδους [11]. Κάθε τέτοια έξοδος ονομάζεται ετικέτα (label). Κατά τη διαδικασία της εκπαίδευσης, ο αλγόριθμος εκτελεί προβλέψεις και γνωρίζοντας τις σωστές αντιστοιχίσεις, υπολογίζει το σφάλμα και προβαίνει σε διορθώσεις ώστε να βελτιώσει την επίδοσή του. Το ζητούμενο είναι, μετά το πέρας της εκπαίδευσης, το μοντέλο που δημιουργήθηκε να έχει την ικανότητα να προβλέψει σωστά την έξοδο για ανεπεξέργαστα δεδομένα εισόδου, που δεν περιέχονταν στο σύνολο εκπαίδευσης.

Τα προβλήματα επιβλεπόμενης μάθησης διακρίνονται σε δύο βασικές κατηγορίες, τα προβλήματα ταξινόμησης (classification) και τα προβλήματα παλινδρόμησης (regression). Στα προβλήματα ταξινόμησης η έξοδος είναι μια τιμή που λαμβάνεται από ένα πεπερασμένο σύνολο τιμών και αντιστοιχεί σε κάποια κλάση. Ο αλγόριθμος εκπαιδεύεται πάνω σε δείγματα εισόδου γνωστής κλάσης με απώτερο στόχο να είναι σε θέση να ταξινομήσει ορθά κάθε νέο, άγνωστο δεδομένο. Στα προβλήματα παλινδρόμησης, η έξοδος είναι ένας αριθμός, ακέραιος ή πραγματικός και στόχος του αλγορίθμου είναι να εντοπίσει το είδος της συσχέτισης μεταξύ των δεδομένων εισόδου με τα δεδομένα εξόδου, ώστε με βάση αυτή, να μπορεί να προβλέψει την τιμή για κάθε νέα, άγνωστη είσοδο.

Μη Επιβλεπόμενη Μάθηση

Στην Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning) ο αλγόριθμος δέχεται δείγματα εισόδου χωρίς ετικέτα, δηλαδή χωρίς ρητή αντιστοίχιση σε κάποια έξοδο και στη συνέχεια, παρατηρεί τη δομή τους, εξάγει πληροφορίες και διακρίνει πιθανές αναπαραστάσεις για αυτά [11]. Το γεγονός ότι οι τεχνικές επιβλεπόμενης μάθησης δύνανται να διαχειριστούν μη επισημασμένα δεδομένα, να διακρίνουν ομοιότητες μεταξύ τους και να ανακαλύψουν κρυφά πρότυπα σε αυτά, τις καθιστά ιδανικές για την επίλυση πληθώρας προβλημάτων, ιδιαίτερα στον τομέα της διερευνητικής ανάλυσης δεδομένων (Exploratory Data Analysis) [12].

Η πιο συνηθισμένη μορφή προβλημάτων αυτού του είδους μάθησης είναι η ομαδοποίηση (clustering), δηλαδή ο διαχωρισμός των δεδομένων σε ομάδες με τέτοιο τρόπο ώστε τα μέλη της ίδιας ομάδας να να έχουν μεγάλο βαθμό ομοιότητας μεταξύ τους και μικρό βαθμό ομοιότητας με τα μέλη των άλλων ομάδων. Η μέθοδος αυτή έχει μεγάλο εύρος εφαρμογών σε τομείς όπως η ιατρική [13], η ανάλυση κοινωνικών δικτύων [14] και η εξόρυξη δεδομένων [15]. Επιπλέον, σε αυτήν την κατηγορία μάθησης υπάγονται τεχνικές μείωσης της διαστατικότητας [16] και η ανίχνευση ανωμαλιών [17].

Ενισχυτική Μάθηση

Στην Ενισχυτική Μάθηση (Reinforcement Learning), ένας πράκτορας (agent) αλληλεπιδρά με το περιβάλλον του και μαθαίνει να ενεργεί σε αυτό με βάση την εμπειρία του, χωρίς να δέχεται καθοδήγηση από έναν εξωτερικό επιτηρητή [11]. Ο πράκτορας παρατηρεί το περιβάλλον, δρα, λαμβάνει είτε ανταμοιβές είτε ποινές εξ αιτίας της δράσης του και προσαρμόζει

ανάλογα την συμπεριφορά του. Στόχος του είναι να ανακαλύψει τις δράσεις που θα του αποφέρουν τη μέγιστη δυνατή συνολική ανταμοιβή. Αξιοσημείωτοι τομείς εφαρμογής της ενισχυτικής μάθησης αποτελούν η ρομποτική [18], η μάθηση επιτραπέζιων παιχνιδιών [19] και οι βιομηχανικές διεργασίες [20].

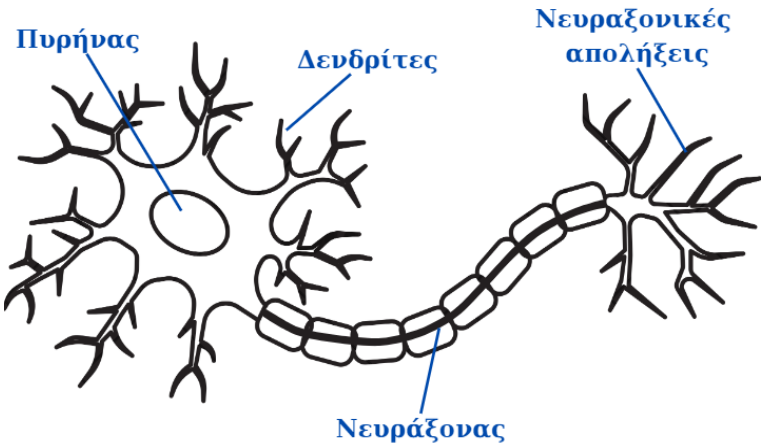
2.3 Νευρωνικά Δίκτυα

Τα Νευρωνικά Δίκτυα (Neural networks) αποτελούν θεμελιώδες εργαλείο της Μηχανικής Μάθησης. Πρόκειται για υπολογιστικές δομές εμπνευσμένες από τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου, τον οποίο προσπαθούν να προσομοιώσουν με στόχο να επιλύσουν κάποιο υπολογιστικό πρόβλημα [21].

2.3.1 Δομή Νευρωνικών Δικτύων

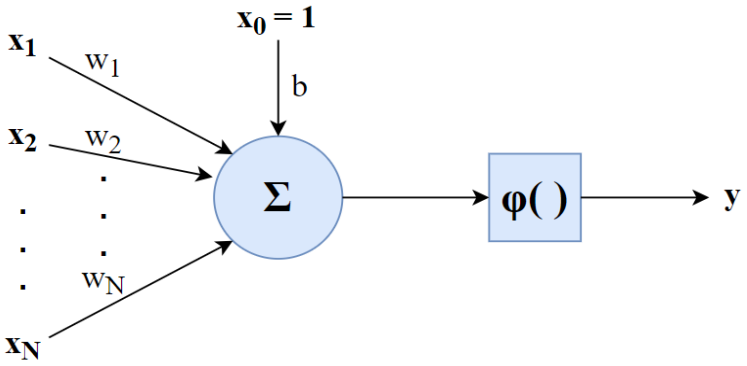
Ο ανθρώπινος εγκέφαλος αποτελείται κυρίως από ένα ευρύ φάσμα νευρώνων, μαζικά διασυνδεδεμένων μεταξύ τους. Οι νευρώνες είναι εξειδικευμένα κύτταρα που αλληλεπιδρούν μεταδίδοντας ηλεκτροχημικά σήματα. Κάθε νευρώνας αποτελείται από ένα κυτταρικό σώμα, στο οποίο βρίσκεται ο πυρήνας του, διακλαδιζόμενες κυτταρικές προεξοχές που συλλέγουν τα σήματα, τους δενδρίτες και μια μακριά λεπτή ίνα που μεταφέρει τα σήματα, τον νευράξονα.

Ο νευράξονας καταλήγει σε πολλαπλές διακλαδώσεις, τις νευραξονικές απολήξεις, οι οποίες συνδέονται με τους δενδρίτες ενός άλλου νευρώνα μέσω συνάψεων, επιτρέποντας την διάδοση σημάτων μεταξύ τους. Αν το σήμα που λαμβάνει ο νευρώνας υπερβαίνει ένα συγκεκριμένο δυναμικό ενέργειας, τον ουδό πυροδότησης, τότε ενεργοποιείται και πυροδοτεί το σήμα αυτό κατά μήκος του νευράξονά του.



Σχήμα 2.1: *Νευρώνας*

Τα δομικά στοιχεία ενός νευρωνικού δικτύου είναι οι τεχνητοί νευρώνες, δηλαδή υπολογιστικές μονάδες που, όπως οι βιολογικοί νευρώνες, αλληλεπιδρούν μεταξύ τους μέσω συνάψεων. Κάθε νευρώνας δέχεται πολλαπλά δεδομένα εισόδου, τόσο από το περιβάλλον όσο και από άλλους νευρώνες και αφού τα επεξεργαστεί, παράγει μια έξοδο την οποία μπορεί είτε να διοχετεύσει στο περιβάλλον είτε να προωθήσει στην είσοδο άλλων νευρώνων. Σε κάθε σύναψη ανάμεσα σε δύο νευρώνες αποδίδεται ένα βάρος που αποτυπώνει την «ισχύ»



Σχήμα 2.2: *Τεχνητός Νευρώνας με N εισόδους*

της. Ο νευρώνας πολλαπλασιάζει καθεμία από τις εισόδους του με το αντίστοιχο συναπτικό βάρος και τροφοδοτεί το άθροισμα αυτών των γινομένων ως είσοδο σε μια συνάρτηση που καλείται συνάρτηση ενεργοποίησης. Η συνάρτηση ενεργοποίησης υλοποιείται εσωτερικά από τον νευρώνα και η τιμή που επιστρέφει αποτελεί την έξοδό του.

Έστω x_{ki} η i-οστή είσοδος του νευρώνα k, w_{ki} το i-οστό συναπτικό βάρος του νευρώνα k, b_k η εξωτερικά εφαρμοζόμενη πόλωση στον νευρώνα k και φ, η συνάρτηση ενεργοποίησης. Τότε, το σήμα εξόδου, y_k, του νευρώνα k υπολογίζεται ως εξής:

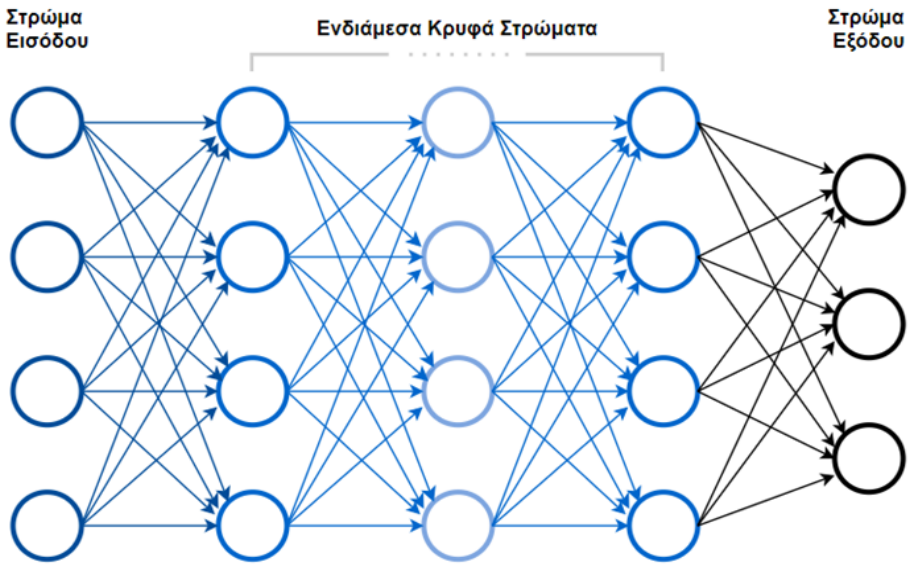
$$y_k = \varphi(b_k + \sum_{i=1}^{N} w_{ki}x_{ki}) \tag{2.6}$$

Οι νευρώνες πρακτικά οργανώνονται σε διαδοχικά στρώματα, που περιλαμβάνουν ένα στρώμα εισόδου, ένα στρώμα εξόδου και κανένα, ένα ή πολλαπλά ενδιάμεσα «κρυφά» στρώματα. Οι νευρώνες του στρώματος εισόδου δέχονται μια είσοδο από το περιβάλλον την οποία μεταβιβάζουν αυτούσια στους νευρώνες του επόμενου στρώματος, χωρίς να την επεξεργαστούν. Κάθε νευρώνας των κρυφών στρωμάτων επεξεργάζεται και μεταβιβάζει την πληροφορία που δέχεται από τις εισόδους του, όπως αναλύθηκε προηγουμένως. Οι νευρώνες του στρώματος εξόδου διοχετεύουν στο περιβάλλον τις τελικές εξόδους που παρήγαγε το δίκτυο ως λύση του προβλήματος.

2.3.2 Βασικές Κατηγορίες Νευρωνικών Δικτύων

Εν γένει, οι νευρώνες ενός στρώματος μπορούν να συνδεθούν μόνο με τους νευρώνες του αμέσως επόμενου και του αμέσως προηγούμενου στρώματος. Ανάλογα με τον τρόπο διασύνδεσης των νευρώνων των διαφορετικών στρωμάτων, τα νευρωνικά δίκτυα διακρίνονται σε δύο βασικές κατηγορίες: τα δίκτυα πρόσθιας τροφοδότησης και τα αναδρομικά δίκτυα.

Σε ένα δίκτυο πρόσθιας τροφοδότησης (feedforward network), οι νευρώνες ενός στρώματος συνδέονται με όλους τους νευρώνες του αμέσως επόμενου στρώματος και η ροή της πληροφορίας συντελείται μόνο «προς τα μπρος», προσιδιάζοντας στην τοπολογία ενός κατευθυνόμενου ακυκλικού γράφου. Μια ειδική κατηγορία forward δικτύου αποτελούν τα συνελικτικά δίκτυα (convolutional networks), στα οποία είναι δυνατόν ένα σύνολο νευρώνων ενός στρώματος να συνδέεται σε έναν μόνο νευρώνα του επόμενου στρώματος (pooling). Στα



Σχήμα 2.3: *Νευρωνικό Δίκτυο*

αναδρομικά δίκτυα επιτρέπεται η σύνδεση ενός νευρώνα με τους νευρώνες του επόμενου, του προηγούμενου και του ίδιου στρώματος.

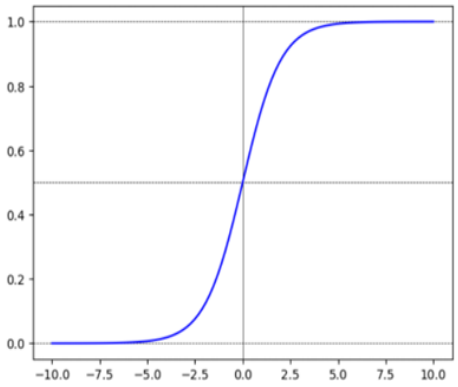
2.3.3 Συνάρτηση Ενεργοποίησης

Η συνάρτηση ενεργοποίησης (activation function) προσδιορίζει την έξοδο ενός νευρώνα και ενδεχομένως περιορίζει τις τιμές της σε ένα συγκεκριμένο εύρος. Η απλούστερη μορφή συνάρτησης ενεργοποίησης είναι η γραμμική συνάρτηση. Ωστόσο, ένα δίκτυο που περιέχει μόνο γραμμικές συναρτήσεις ενεργοποίησης ουσιαστικά εκφυλίζεται σε μία μόνο γραμμική συνάρτηση, μέσω της οποίας είναι αδύνατον να αναπαρασταθούν αυθαίρετα μη γραμμικά συστήματα του φυσικού κόσμου. Συνεπώς, είναι απαραίτητη η εισαγωγή μη γραμμικών συναρτήσεων ενεργοποίησης. Σύμφωνα με το θεώρημα καθολικής προσέγγισης, ένα δίκτυο που αποτελείται από ένα γραμμικό και ένα μη γραμμικό επίπεδο, έχει τη δυνατότητα να προσεγγίσει οποιαδήποτε συνεχή συνάρτηση με έναν αυθαίρετο βαθμό ακρίβειας. Οι πιο ευρέως χρησιμοποιούμενες μη γραμμικές συναρτήσεις ενεργοποίησης είναι:

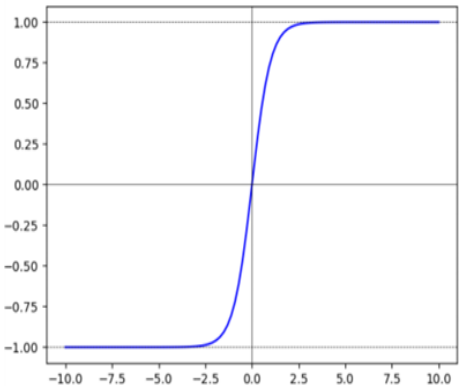
- Οι σιγμοειδείς συναρτήσεις, των οποίων η καμπύλη έχει τη μορφή «S» και είναι συνεχείς και διαφορίσιμες σε όλο το πεδίο ορισμού τους. Δύο τυπικές σιγμοειδείς συναρτήσεις είναι η λογιστική σιγμοειδής $\sigma(x) = \frac{1}{1+e^{-x}}$, με εύρος τιμών $(0, 1)$ και η υπερβολική εφαπτομένη $tanh(x)$, με εύρος τιμών $(-1, 1)$.

- Η συνάρτηση ReLU (Rectified Linear Unit - Ανορθωμένη Γραμμική Μονάδα), η οποία ορίζεται ως $\text{ReLU}(x) = \max(0, x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}$ και το εύρος τιμών της είναι το $[0, \infty)$.

- Η συνάρτηση softplus που είναι μια ομαλή εκδοχή της ReLU και ορίζεται ως $softmax(x) = log(1 + e^x)$, με εύρος τιμών $(0, \infty)$.

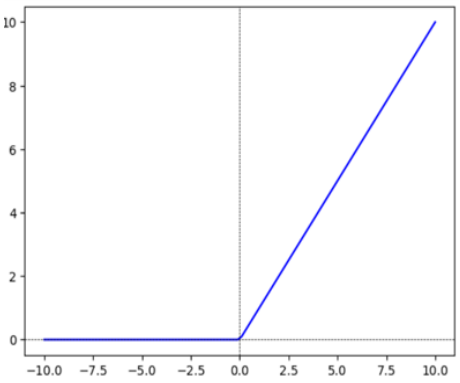


(α') *Λογιστική σιγμοειδής συνάρτηση σ(x)*



(β') *Συνάρτηση tanh(x)*

Σχήμα 2.4: *Σιγμοιειεδείς Συναρτήσεις*

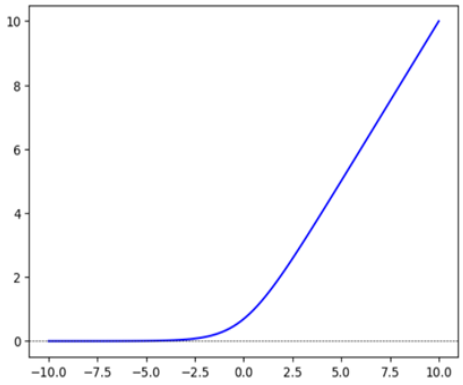


Σχήμα 2.5: *Συνάρτηση ReLU(x)*

2.3.4 Εκπαίδευση Νευρωνικών Δικτύων

Ένα από τα βασικότερα χαρακτηριστικά των νευρωνικών δικτύων είναι η ικανότητά τους να μαθαίνουν να επιλύουν κάποιο πρόβλημα. Η εκπαίδευσή ενός νευρωνικού δικτύου συνιστά μια επαναληπτική διαδικασία κατά την οποία οι παράμετροί του, δηλαδή τα συναπτικά βάρη και οι πολώσεις του, τροποποιούνται κατάλληλα ώστε να βελτιώνεται η επίδοση του. Μετά το πέρας της εκπαίδευσης οι τιμές των παραμέτρων που προέκυψαν χρησιμοποιούνται ως σταθερές ώστε το δίκτυο να επιτελέσει τη λειτουργία του. Το ζητούμενο είναι το δίκτυο να έχει την ικανότητα να δίνει ορθές εξόδους για πρωτόγνωρες εισόδους που δε χρησιμοποιήθηκαν για την εκπαίδευσή του.

Η εκπαίδευση ενός νευρωνικού δικτύου μπορεί να είναι επιβλεπόμενη, μη επιβλεπόμενη ή ενισχυτική. Στην επιβλεπόμενη εκπαίδευση, η απόκριση του δικτύου είναι γνωστή για δεδομένα δείγματα εισόδου, οπότε η προσαρμογή των παραμέτρων του βασίζεται στο υπολογιζόμενο σφάλμα, δηλαδή στη διαφορά μεταξύ της πραγματικής και της επιθυμητής εξόδου. Στη μη-επιβλεπόμενη εκπαίδευση δεν απαιτείται η καθοδήγηση από έναν εξωτερικό επιβλέποντα. Η ποιότητα της ζητούμενης αναπαράστασης αποτυπώνεται σε κάποια μετρική και οι παράμετροι του δικτύου τροποποιούνται με βάση αυτή, ώστε να τη βελτιστοποιήσουν. Η ενισχυτική εκπαίδευση βασίζεται στη συνεχή αλληλεπίδραση ενός πράκτορα με το περιβάλλον, με στόχο την ελαχιστοποίηση μιας συνάρτησης που υπολογίζει την αναμενόμενη



Σχήμα 2.6: *Συνάρτηση softplus(x)*

συνολική ανταμοιβή για την αλληλουχία των αποφάσεων που παίρνει ως προς τη δράση του.

Η ενημέρωση των παραμέτρων του δικτύου, δηλαδή των συναπτικών βαρών μπορεί να πραγματοποιηθεί με διάφορες μεθόδους, μεταξύ των οποίων η κατάβαση πλαγιάς, η στοχαστική κατάβαση πλαγιάς και ο βελτιστοποιητής Adam.

Κατάβαση Πλαγιάς

Η μέθοδος της κατάβασης πλαγιάς (Gradient Descent) χρησιμοποιείται ευρέως στην εκπαίδευση των νευρωνικών δικτύων για τη σταδιακή προσαρμογή των παραμέτρων τους, ώστε να ελαττώνεται το εκάστοτε σφάλμα. Το σφάλμα ορίζεται ως συνάρτηση των βαρών του δικτύου, η οποία αναπαριστά μια πολυδιάστατη επιφάνεια σφάλματος με συντεταγμένες τα βάρη. Υπολογίζοντας την κλίση της επιφάνειας σφάλματος σε κάποιο σημείο καταδεικνύεται η κατεύθυνση προς την οποία πρέπει να κινηθούν τα βάρη προκειμένου να οδηγηθούν σε κάποιο ελάχιστο σημείο αυτής (τοπικό ή ολικό).

Σε κάθε επανάληψη της εκπαιδευτικής διαδικασίας υπολογίζονται οι μερικές παράγωγοι της συνάρτησης σφάλματος ως προς τα ζητούμενα βάρη, με χρήση μεθόδων όπως η οπίσθια διάδοση (backpropagation). Η διόρθωση που εφαρμόζεται σε κάθε βάρος είναι ανάλογη της μερικής παραγώγου της συνάρτησης σφάλματος ως προς αυτό.

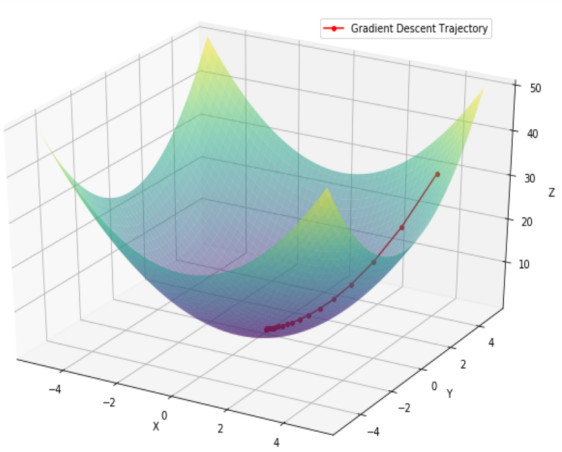
Έστω w το διάνυσμα βαρών, w_{ij} το βάρος της σύναψης μεταξύ των νευρώνων i και j και $J(w)$ η συνάρτηση σφάλματος. Η ενημέρωση του βάρους w_{ij} πραγματοποιείται ως εξής:

$$w'_{ij} = w_{ij} - \eta \frac{\partial J(w)}{\partial w_{ij}} \tag{2.7}$$

όπου w'_{ij} είναι η επικαιροποιημένη τιμή του βάρους και η ο ρυθμός μάθησης, δηλαδή ο ρυθμός μεταβολής των βαρών. Το αρνητικό πρόσημο σηματοδοτεί την κατάβαση στον χώρο των βαρών, δηλαδή την κατεύθυνση προς την οποία η μεταβολή θα επιφέρει μείωση του σφάλματος.

Οπίσθια Διάδοση

Η μέθοδος της οπίσθιας διάδοσης (backpropagation) αποτελεί έναν συστηματικό τρόπο υπολογισμού της κλίσης της επιφάνειας σφάλματος. Υπολογίζει τις μερικές παραγώγους της συνάρτησης κόστους ως προς τα ζητούμενα βάρη, χρησιμοποιώντας τον κανόνα της αλυ-



Σχήμα 2.7: *Τριδιάστατη επιφάνεια σφάλματος και τροχιά της Κατάβασης Πλαγιάς*

σίδας, ξεκινώντας από το στρώμα εξόδου και προχωρώντας προς τα πίσω. Η διαδικασία αυτή πραγματοποιείται μέσω δυναμικού προγραμματισμού [22], αποθηκεύοντας τα ενδιάμεσα αποτελέσματα.

Στοχαστική Κατάβαση Πλαγιάς

Μια παραλλαγή της μεθόδου Gradient Descent είναι η Στοχαστική Κατάβαση Πλαγιάς (Stochastic Gradient Descent – SGD), κατά την οποία σε κάθε βήμα της εκπαίδευσης επιλέγεται ένα μικρό πλήθος τυχαίων δειγμάτων για τον υπολογισμό της κλίσης της συνάρτησης κόστους. Σημειώνεται ότι η μέθοδος Gradient Descent χρησιμοποιεί ολόκληρο το σύνολο των δειγμάτων εκπαίδευσης και υπολογίζει την πραγματική κλήση. Η SGD ουσιαστικά, υπολογίζει μια προσέγγιση της πραγματικής κλίσης. Η μέθοδος αυτή συγκλίνει γρηγορότερα σε κάποιο ελάχιστο σε σχέση με την Gradient Descent, ωστόσο η σύγκλησή της στο ολικό ελάχιστο δεν είναι απόλυτα εγγυημένη, καθώς μπορεί να ταλαντώνεται γύρω του χωρίς να κατασταλάζει σε αυτό.

Βελτιστοποιητής Adam

Στην κάθοδο κλίσης και τη στοχαστική κάθοδο κλίσης χρησιμοποιείται ο ίδιος ρυθμός μάθησης για τις ενημερώσεις όλων των παραμέτρων και παραμένει σταθερός καθόλη τη διάρκεια της εκπαίδευσης. Αντιθέτως, ο Adam είναι ένας βελτιστοποιητής που διατηρεί έναν διαφορετικό ρυθμό μάθησης για κάθε παράμετρο, τον οποίο προσαρμόζει ξεχωριστά για την καθεμία καθώς εξελίσσεται η εκπαίδευση.

2.3.5 Παράμετροι Νευρωνικών Δικτύων

Οι υπερπαράμετροι του δικτύου είναι σταθερές παράμετροι που προσδιορίζονται πριν την έναρξη της διαδικασίας εκπαίδευσης και επηρεάζουν σημαντικά την ποιότητά της. Παραδείγματα υπερπαραμέτρων είναι ο το πλήθος των κρυφών στρωμάτων του δικτύου και των νευρώνων κάθε στρώματος, ο ρυθμός μάθησης, το πλήθος των επαναλήψεων καθώς και το μέγεθος των πακέτων δειγμάτων που θα χρησιμοποιηθούν σε κάθε επανάληψη.

Στρώματα και νευρώνες

Το πλήθος των στρωμάτων του νευρωνικού δικτύου καθώς και το πλήθος των νευρώνων κάθε στρώματος επηρεάζουν καθοριστικά τόσο την υπολογιστική του πολυπλοκότητα όσο και την επίδοσή του. Αυξάνοντας το πλήθος των στρωμάτων και των νευρώνων που τα απαρτίζουν αυξάνεται σημαντικά η πολυπλοκότητα του συστήματος και συνεπώς ο χρόνος που απαιτείται για την εκπαίδευσή του. Ωστόσο, μειώνοντας αυτό το πλήθος, ενδέχεται να μειωθεί η απόδοση.

Ρυθμός μάθησης

Ο ρυθμός μάθησης διαδραματίζει καθοριστικό ρόλο στην επίδοση της εκπαίδευσης. Μικρός ρυθμός μάθησης συνεπάγεται μικρές μεταβολές των παραμέτρων και κατ' επέκταση πιο ομαλή αλλά και πιο αργή διαδικασία εκπαίδευσης. Επίσης, υπάρχει κίνδυνος εγκλωβισμού σε κάποιο τοπικό ελάχιστο. Αντιθέτως, αν ο ρυθμός μάθησης είναι σχετικά μεγάλος, συντελούνται μεγάλες μεταβολές στις παραμέτρους με αποτέλεσμα να επιταχύνεται η διαδικασία εκπαίδευσης, αλλά και να υπάρχει κίνδυνος αστάθειας και δυσκολίας σύγκλισης του δικτύου λόγω ταλαντώσεων. Συνεπώς, είναι προτιμότερο η τιμή του ρυθμού μετάβασης να μεταβάλλεται κατάλληλα από τη μια επανάληψη στην άλλη.

2.4 Διαδικασίες Λήψης Αποφάσεων Markov

Ένα πρόβλημα λήψης αποφάσεων μοντελοποιείται συνήθως ως μία Μαρκοβιανή Διαδικασία Απόφασης (ΜΔΑ) (Markov Decision Process-MDP), η οποία αποτελεί μία στοχαστική διαδικασία λήψης αποφάσεων για διακριτές τιμές χρόνου [23]. Η Μαρκοβιανή διαδικασία ή Μαρκοβιανή αλυσίδα είναι ένα στοχαστικό μοντέλο που περιγράφει μια αλληλουχία καταστάσεων, στην οποία η επόμενη κατάσταση εξαρτάται αποκλειστικά από την τρέχουσα κατάσταση και όχι από τις προηγούμενες [24].

Η ΜΔΑ αποτελεί επέκταση της Μαρκοβιανής αλυσίδας, εισάγοντας τις έννοιες της δράσης και της ανταμοιβής. Συγκεκριμένα, μια ΜΔΑ ορίζεται ως μια πλειάδα $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ όπου:

- $\mathcal{S}$: το σύνολο όλων των δυνατών καταστάσεων
- $\mathcal{A}$: το σύνολο όλων των δυνατών δράσεων
- $\mathcal{P}$: το μοντέλο μετάβασης
- $\mathcal{R}$: η άμεση ανταμοιβή λόγω μετάβασης

2.4.1 Βασικές έννοιες

Για να μοντελοποιηθεί ένα πρόβλημα ως διαδικασία λήψης αποφάσεων, θα πρέπει να προσδιοριστούν οι ακόλουθες έννοιες [11, 25]:

Πράκτορας

Ως πράκτορας νοείται οποιοδήποτε υπολογιστικό σύστημα έχει την ικανότητα να αντιλαμβάνεται το περιβάλλον του και να επενεργεί σε αυτό με βάση τα ερεθίσματα που δέχεται.

Περιβάλλον

Το περιβάλλον αποτελεί ουσιαστικά τον κόσμο στον οποίο ζει ο πράκτορας και επηρεάζεται από τη δράση του πράκτορα σε αυτό και από άλλους ενδεχομένως παράγοντες. Ανάλογα με τις μεταβολές που υφίσταται η κατάστασή του αποδίδει στον πράκτορα ανταμοιβές ή ποινές.

Κατάσταση

Η έννοια της κατάστασης μπορεί να θεωρηθεί ως μια αναπαράσταση του περιβάλλοντος. Συνιστά ένα σύνολο πληροφοριών που περιγράφουν το περιβάλλον σε μια δεδομένη χρονική στιγμή. Ο πράκτορας παρατηρώντας την κατάσταση του περιβάλλοντος, λαμβάνει πληροφορίες τις οποίες αξιοποιεί για να αποφασίσει την δράση του.

Αν ο πράκτορας έχει πρόσβαση στην πλήρη κατάσταση του περιβάλλοντος κάθε χρονική στιγμή, τότε το περιβάλλον είναι πλήρως παρατηρήσιμο, ενώ αν έχει πρόσβαση σε μια μερική αναπαράσταση της κατάστασης του περιβάλλοντος, τότε το περιβάλλον είναι μερικώς παρατηρήσιμο. Υπάρχουν όμως και περιπτώσεις που ο πράκτορας δεν έχει καθόλου πρόσβαση στην κατάσταση του περιβάλλοντος, οπότε το περιβάλλον είναι μη παρατηρήσιμο.

Ανταμοιβή

Για κάθε μετάβαση από την κατάσταση s στην κατάσταση s' λόγω της δράσης a, ο πράκτορας λαμβάνει μια άμεση ανταμοιβή, η οποία υπολογίζεται μέσω μιας συνάρτησης $R_s^a = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$. Στόχος του πράκτορα είναι να ανακαλύψει την αλληλουχία δράσεων που μεγιστοποιεί την αναμενόμενη συνολική ανταμοιβή.

Αν το πρόβλημα είναι πεπερασμένου χρονικού ορίζοντα, δηλαδή έχει πεπερασμένη χρονική διάρκεια, η συνολική ανταμοιβή αποτελεί απλώς το άθροισμα των άμεσων ανταμοιβών, δηλαδή $G_t = R_{t+1} + R_{t+2} + \cdots + R_T$. Αν όμως το πρόβλημα είναι άπειρου χρονικού ορίζοντα, εισάγεται ένας παράγοντας μείωσης γ με τον οποίο σταθμίζονται οι ανταμοιβές και η συνολική ανταμοιβή εκφράζεται ως το άθροισμα των σταθμισμένων άμεσων ανταμοιβών, δηλαδή $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$. Σε κάθε περίπτωση το ζητούμενο είναι η μεγιστοποίηση της αναμενόμενης τιμής της ποσότητας G_t.

Παράγοντας μείωσης γ

Ο παράγοντας μείωσης (discount factor) γ είναι ένας αριθμός μεταξύ 0 και 1 που αποτυπώνει τον βαθμό της προτίμησης του πράκτορα για τις τρέχουσες ανταμοιβές έναντι των μεταγενέστερων ανταμοιβών. Όσο πιο κοντά στο 0 είναι η τιμή του, τόσο πιο αμελητέες θεωρούνται οι μεταγενέστερες ανταμοιβές, ενώ όσο πιο κοντά στο 1 είναι η τιμή του, τόσο πιο πολύ λαμβάνονται υπόψιν.

Μοντέλο Μετάβασης

Το μοντέλο μετάβασης του περιβάλλοντος περιγράφει το αποτέλεσμα κάθε δράσης σε κάθε κατάσταση και βοηθάει τον πράκτορα να επιλέξει τις δράσεις του. Αποτελεί μια συνάρτηση πυκνότητας πιθανότητας μέσω της οποίας υπολογίζεται η πιθανότητα μετάβασης σε κάποια επόμενη κατάσταση δεδομένης της τρέχουσας κατάστασης και της δράσης που εκτελείται. Συγκεκριμένα, ορίζεται μια συνάρτηση, $P_{SS'}^a = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$ που υπολογίζει

την πιθανότητα μετάβασης στην κατάσταση $S_{t+1} = s'$ δεδομένου ότι η τρέχουσα κατάσταση είναι η $S_t = s$ και η δράση που επιλέχθηκε είναι η $A_t = a$.

Πολιτική

Η πολιτική καθορίζει πλήρως τη συμπεριφορά του πράκτορα, καθώς αποτελεί τον κανόνα που χρησιμοποιεί για να αποφασίσει ποια δράση θα εκτελέσει. Ουσιαστικά είναι μια συνάρτηση που εκφράζει την κατανομή πιθανότητας κάθε δράσης που είναι δυνατόν να επιλεγεί δεδομένης μιας κατάστασης. Επομένως, αν ο πράκτορας ακολουθεί την πολιτική π και τη χρονική στιγμή t βρίσκεται στην κατάσταση s, τότε η $\pi(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s]$ δίνει την πιθανότητα να επιλέξει την δράση a. Στην πράξη, η πολιτική καταδεικνύει ποια δράση θα εκτελεστεί σε κάθε κατάσταση. Η πολιτική μέσω της οποίας προκύπτει η μέγιστη συνολική ανταμοιβή αντιστοιχεί στην βέλτιστη πολιτική.

Στην περίπτωση πεπερασμένου ορίζοντα, η πολιτική είναι μη στάσιμη, δηλαδή εξαρτάται από τον χρόνο, καθώς σε μια δεδομένη κατάσταση η βέλτιστη δράση μπορεί να είναι διαφορετική ανάλογα με τη χρονική στιγμή. Αντιθέτως, στην περίπτωση άπειρου ορίζοντα, η πολιτική είναι στάσιμη, καθώς για μια δεδομένη κατάσταση η βέλτιστη δράση είναι πάντα η ίδια, ανεξάρτητα από την χρονική στιγμή.

Συνοψίζοντας, κάθε χρονική στιγμή $t = 0, 1, 2, \ldots$ το περιβάλλον βρίσκεται σε κάποια κατάσταση S_t. Ο πράκτορας παρατηρεί αυτήν την κατάσταση και επιλέγει μια δράση A_t σύμφωνα με την πολιτική του. Ως αποτέλεσμα, το περιβάλλον μεταβαίνει σε μια νέα κατάσταση S_{t+1}, με πιθανότητα που υπαγορεύεται από το μοντέλο μετάβασης και παράλληλα στέλνει στον πράκτορα την αντίστοιχη αναμενόμενη ανταμοιβή R_s^a. Έτσι, προκύπτει μια αλληλουχία καταστάσεων, δράσεων και ανταμοιβών: $S_0 A_0 R_0 S_1 A_1 R_1 S_2, \ldots$ Στόχος είναι η εύρεση μιας πολιτικής επιλογής δράσεων που μεγιστοποιεί την μακροπρόθεσμη συνολική ανταμοιβή.

2.4.2 Συνάρτηση Αξίας

Ο υπολογισμός της αναμενόμενης συνολικής ανταμοιβής αν αφετηρία είναι η τρέχουσα κατάσταση και εφαρμόζεται μια συγκεκριμένη πολιτική, πραγματοποιείται από τις λεγόμενες συναρτήσεις αξίας (value-functions), οι οποίες διακρίνονται στις συναρτήσεις κατάστασης-αξίας (state-value functions) και στις συναρτήσεις δράσης-αξίας (action-value functions).

State-value functions

Οι state-value functions συμβολίζονται ως $v^\pi(s)$ και υπολογίζουν την αναμενόμενη συνολική ανταμοιβή που θα ληφθεί ξεκινώντας την κατάσταση s και ακολουθώντας την πολιτική π. Δηλαδή, για κάθε $s \in \mathcal{S}$:

$$v^\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s] \tag{2.8}$$

Action-value functions

Οι action-value functions ή συναρτήσεις-Q (Q-functions) συμβολίζονται ως $q^\pi(s, a)$ και υπολογίζουν την αναμενόμενη συνολική ανταμοιβή που θα ληφθεί αν ξεκινώντας από την

κατάσταση s, εκτελεστεί η δράση a και στη συνέχεια εφαρμοστεί η πολιτική π.

$$q^{\pi}(s,a) = \mathbb{E}_{\pi}[G_t \mid S_t = s, A_t = a] \tag{2.9}$$

2.4.3 Εξίσωση Bellman

Η state-value function μπορεί να αναλυθεί σε δύο μέρη, στην άμεση αναμενόμενη ανταμοιβή για την μετάβαση στην επόμενη κατάσταση, R_{t+1} και την αξία της επόμενης κατάστασης πολλαπλασιασμένη με τον παράγοντα μείωσης, $\gamma v^{\pi}(S_{t+1})$ με την παραδοχή ότι ο πράκτορας εφαρμόζει μια συγκεκριμένη πολιτική. Επομένως, ορίζεται η εξίσωση Bellman για την state-value function ως εξής:

$$\begin{aligned} v^{\pi}(s) &= E_{\pi}[R_{t+1} + \gamma v^{\pi}(S_{t+1}) \mid S_t = s] \\ &= \sum_{a \in \mathcal{A}} \pi(a \mid s)(R_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v^{\pi}(s')) \end{aligned} \tag{2.10}$$

Ομοίως, ορίζεται η εξίσωση Bellman για την action-value function ως εξής:

$$\begin{aligned} q^{\pi}(s,a) &= E_{\pi}[R_{t+1} + \gamma q^{\pi}(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a] \\ &= R_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a' \mid s') q_{\pi}(s', a') \end{aligned} \tag{2.11}$$

Η εξισώσεις Bellman ουσιαστικά επιτρέπουν τον υπολογισμό της αξίας κάθε κατάστασης με βάση την αξία μελλοντικών καταστάσεων. Γνωρίζοντας την αξία μεταγενέστερων καταστάσεων και δεδομένης μιας τυχαίας αρχικοποίησης, η value-function μπορεί να υπολογιστεί αναδρομικά για κάθε δυνατή κατάσταση, σύμφωνα με την αντίστοιχη εξίσωση Bellman.

Οι εξισώσεις αυτές κατέχουν κεντρικό ρόλο στις Μαρκοβιανές διαδικασίες λήψης αποφάσεων και αποτελούν τη βάση για αλγορίθμους στους τομείς του δυναμικού προγραμματισμού και της ενισχυτικής μάθησης.

2.4.4 Βέλτιστη Πολιτική

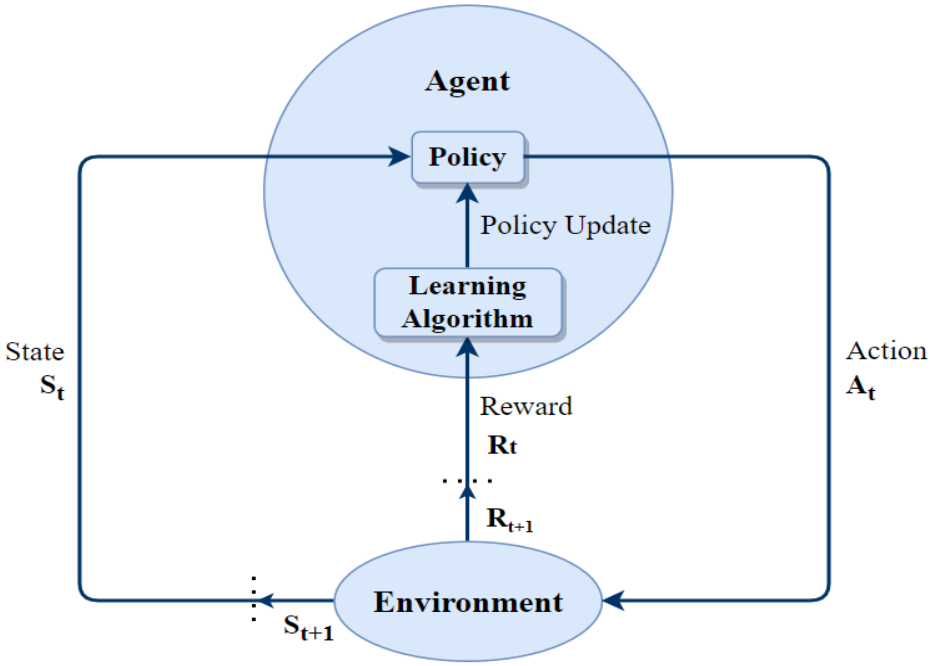
Η βέλτιστη πολιτική π^* ικανοποιεί την εξίσωση Bellman και ισχύει ότι:

$$\pi^* \geq \pi, \forall \pi \tag{2.12}$$

Επιπλέον, η βέλτιστη, δηλαδή η μέγιστη τιμή των value-functions προκύπτει αν εφαρμοστεί η βέλτιστη πολιτική. Συγκεκριμένα ισχύουν τα εξής:

$$v_{\pi^*}(s) = v^*(s) \tag{2.13}$$

$$q_{\pi^*}(s,a) = q^*(s,a) \tag{2.14}$$



Σχήμα 2.8: *Ενισχυτική Μάθηση*

Κατά συνέπεια, ο στόχος της εύρεσης της βέλτιστης πολιτικής αντικατοπτρίζεται στην εύρεση της βέλτιστης action-value function

$$\pi^*(a \mid s) = \begin{cases} 1, & \text{if } a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, q^*(s, a) \\ 0, & \text{otherwise} \end{cases} \tag{2.15}$$

2.5 Ενισχυτική Μάθηση

Η ενισχυτική μάθηση αποτελεί έναν κλάδο της μηχανικής μάθησης στον οποίο μια αυτόνομη οντότητα που καλείται πράκτορας, αλληλεπιδρά με το περιβάλλον της, προσπαθώντας να επιτύχει κάποιο στόχο [11, 23]. Παρόμοια με τις ΜΔΑ, κατά την αλληλεπίδραση πράκτορα-περιβάλλοντος, ο πράκτορας παρατηρεί το περιβάλλον και δρα σε αυτό. Ακολούθως, το περιβάλλον μεταβάλλει ανάλογα την κατάστασή του και αποδίδει στον πράκτορα μια ανταμοιβή που αντικατοπτρίζει την «αξία» της κατάστασης στην οποία μετέβη. Σκοπός του πράκτορα είναι να βρει τη βέλτιστη πολιτική που πρέπει να ακολουθήσει όσον αφορά τον τρόπο που δρα, ώστε να επιτύχει τη μέγιστη συνολική ανταμοιβή. Ωστόσο, σε αντίθεση με τις ΜΔΑ, στην ενισχυτική μάθηση ενδέχεται να είναι άγνωστο το μοντέλο μετάβασης ή η συνάρτηση ανταμοιβής.

2.5.1 Βασικές Έννοιες

Model-based και model-free

Το μοντέλο μετάβασης του περιβάλλοντος δεν είναι πάντα γνωστό. Με βάση αυτή τη συνθήκη, οι μέθοδοι ενισχυτικής μάθησης διακρίνονται σε μεθόδους βασισμένους σε μοντέλο (model-based) και σε μεθόδους ελεύθερου μοντέλου (model-free).

Στην model-based ενισχυτική μάθηση ο πράκτορας αξιοποιεί το μοντέλο μετάβασης του περιβάλλοντος, το οποίο είτε είναι εξ αρχής γνωστό είτε χρειάζεται να το μάθει σταδιακά αξιολογώντας την επίδραση της δράσης του. Στην model-free ενισχυτική μάθηση ο πράκτορας δε γνωρίζει και ούτε δύναται να μάθει το μοντέλο μετάβασης του περιβάλλοντος. Επομένως, καλείται να μάθει έμμεσα έναν τρόπο συμπεριφοράς μέσω της δοκιμής και του σφάλματος. Δύο χαρακτηριστικές μέθοδοι μάθησης αυτού του είδους είναι η policy iteration και η value iteration, στην οποία υπάγεται η ευρέως χρησιμοποιούμενη μάθηση-Q (Q-learning). Συνήθως, το μοντέλο μετάβασης δεν είναι γνωστό.

On-policy και Off-policy

Η μέθοδος που χρησιμοποιείται για την εύρεση της βέλτιστης πολιτικής μπορεί να είναι είτε on-policy είτε off-policy. Στην περίπτωση μιας on-policy τεχνικής, ο πράκτορας προσπαθεί να εκτιμήσει και να βελτιώσει την ίδια πολιτική που χρησιμοποιεί και για να επιλέξει τις δράσεις του. Αντιθέτως, σε μια off-policy τεχνική, η πολιτική που μαθαίνει και βελτιστοποιεί ο πράκτορας καλείται target policy και είναι διαφορετική από αυτή με την οποία επιλέγει τις δράσεις του, η οποία καλείται behaviour policy.

Online και Offline learning

Στην online εκπαίδευση, ο πράκτορας μαθαίνει σε πραγματικό χρόνο, δηλαδή ενημερώνει την πολιτική του με βάση τις πληροφορίες που αποκτά καθώς αλληλεπιδρά με το περιβάλλον του. Αυτή η προσέγγιση επιτρέπει στον πράκτορα να εξερευνά το περιβάλλον και να προσαρμόζεται άμεσα στις μεταβολές του, τροποποιώντας την πολιτική του σύμφωνα με τα νέα δεδομένα που λαμβάνει. Ωστόσο, η συνεχής αλληλεπίδραση με το περιβάλλον μπορεί να επιφέρει αστάθεια στην εκπαίδευση.

Στην offline εκπαίδευση, ο πράκτορας δεν αλληλεπιδρά ενεργά με το περιβάλλον του, αλλά μαθαίνει χρησιμοποιώντας ένα συγκεκριμένο σύνολο δεδομένων, τα οποία έχουν συλλεγεί εκ των προτέρων. Το γεγονός ότι η εκπαίδευση βασίζεται σε προκαθορισμένες πληροφορίες, συνεπάγεται σταθερότητα και αξιοπιστία. Ωστόσο, ο πράκτορας δεν μπορεί να ανταποκριθεί σε τυχόν μεταβολές του περιβάλλοντος και ενδεχομένως να μην έχει στη διάθεσή του επικαιροποιημένες πληροφορίες.

2.5.2 Βασικές Μέθοδοι Ενισχυτικής Μάθησης

Δυναμικός προγραμματισμός

Ο Δυναμικός Προγραμματισμός (Dynamic Programming - DP) αποτελεί μια υπολογιστική μέθοδο επίλυσης προβλημάτων μέσω του αναδρομικού διαχωρισμού τους σε αλληλο-

εξαρτώμενα υποπροβλήματα [22]. Κάθε υποπρόβλημα επιλύεται μόνο μια φορά και η λύση του αποθηκεύεται στη μνήμη ώστε να αποφευχθεί η εκ νέου επίλυσή του κάθε φορά που χρειάζεται η λύση του. Η θεωρία του δυναμικού προγραμματισμού εδράζεται στην αρχή της βελτιστότητας που εισήγαγε ο Bellman, σύμφωνα με την οποία κάθε τμήμα της βέλτιστης λύσης αποτελεί βέλτιστη λύση για το αντίστοιχο υποπρόβλημα.

Η τεχνική αυτή εφαρμόζεται για την εύρεση της βέλτιστης πολιτικής σε μια ΜΔΑ και γενικότερα σε προβλήματα βελτιστοποίησης που υπακούουν στη Μαρκοβιανή ιδιότητα. Γενικά, οι αποφάσεις πρέπει να λαμβάνονται σε διακριτά στάδια, με την έκβασή τους να είναι σε ένα βαθμό προβλέψιμη. Επίσης, ενώ θεωρητικά η χρήση αυτής της μεθόδου μπορεί να γίνει τόσο σε διακριτούς όσο και σε συνεχείς χώρους καταστάσεων και δράσεων, στην πράξη, σε συνεχείς χώρους είναι εφικτή υπό προϋποθέσεις. Επομένως, συνήθως οι συνεχείς χώροι διακριτοποιούνται πριν την εφαρμογή δυναμικού προγραμματισμού.

Θεμέλιο των αλγορίθμων δυναμικού προγραμματισμού είναι οι εξισώσεις Bellman, οι οποίες εκφράζουν την αναδρομική σχέση που συνδέει τη βέλτιστη λύση ενός προβλήματος με τη βέλτιστη λύση των υποπροβλημάτων του. Σε μια ΜΔΑ και στην Ενιχυτική Μάθηση, ο δυναμικός προγραμματισμός χρησιμοποιεί τις value-functions προκειμένου να οργανώσει και να δομήσει τη στρατηγική εύρεσης της βέλτιστης πολιτικής. Ουσιαστικά, γνωρίζοντας τις βέλτιστες τιμές των value-functions, η βέλτιστη πολιτική υπολογίζεται με επαναληπτική εφαρμογή των εξισώσεων Bellman. Οι τιμές των value-functions ενημερώνονται σύμφωνα με την εξίσωση Bellman και με βάση τις νέες αυτές τιμές ανανεώνεται η πολιτική. Η διαδικασία επαναλαμβάνεται μέχρι να επιτευχθεί σύγκλιση στη βέλτιστη πολιτική:

$$v^*(s) = \max_{a \in \mathcal{A}} R_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v^*(s') \tag{2.16}$$

$$q^*(s,a) = R_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a \max_{a' \in \mathcal{A}} q^*(s',a') \tag{2.17}$$

Δύο βασικοί αλγόριθμοι για την επίλυση της εξίσωσης Bellman με στόχο τον υπολογισμό της βέλτιστης πολιτικής είναι η επανάληψη πολιτικής (policy-iteration) και η επανάληψη τιμής (value-iteration).

Policy Iteration

Ο αλγόριθμος Policy Iteration εκτελεί εναλλάξ τα ακόλουθα δύο βήματα:

1. Αποτίμηση πολιτικής: Δοθείσης της τρέχουσας πολιτικής π_i, υπολογίζεται η state-value function για κάθε κατάσταση v_{π_i}.

2. Βελτιστοποίηση πολιτικής: Υπολογίζεται μια νέα πολιτική χρησιμοποιώντας τη v_{π_i}.

Η διαδικασία αυτή τερματίζεται όταν η πολιτική παύει να μεταβάλλεται, δηλαδή όταν το βήμα της βελτίωσης δεν μεταβάλλει την τιμή της αξίας. Δεδομένου ότι ΜΔΑ αποτελείται από πεπερασμένο αριθμό καταστάσεων, αυτό συμβαίνει μετά από πεπερασμένο αριθμό επαναλήψεων. Η τελική value-function αποτελεί λύση της εξίσωσης Bellman και η αντίστοιχη

πολιτική είναι η βέλτιστη.

$$v(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P_{ss'}^{\pi(s)} \left(R_s^a + \gamma v(s')\right) \tag{2.18}$$

$$\pi(s) = \operatorname*{argmax}_{a' \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P_{ss'}^{a} \left(R_s^a + \gamma v(s')\right) \tag{2.19}$$

Value-Iteration

Στον αλγόριθμο Value-Iteration σε κάθε κατάσταση αντιστοιχεί μια εξίσωση Bellman που υπολογίζει τη βέλτιστη τιμή της state-value function. Για την επίλυση αυτών των μη-γραμμικών εξισώσεων, η αξία κάθε κατάστασης αρχικοποιείται σε κάποια αυθαίρετη τιμή και στη συνέχεια, ενημερώνεται επαναληπτικά με βάση την αξία των γειτονικών καταστάσεων. Η ενημέρωση αυτή εκτελείται ταυτόχρονα για όλες τις καταστάσεις και επαναλαμβάνεται μέχρι να επιτευχθεί σύγκλιση.

$$v_{i+1}(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P_{ss'}^{a} \left(R_s^a + \gamma v_i(s')\right) \tag{2.20}$$

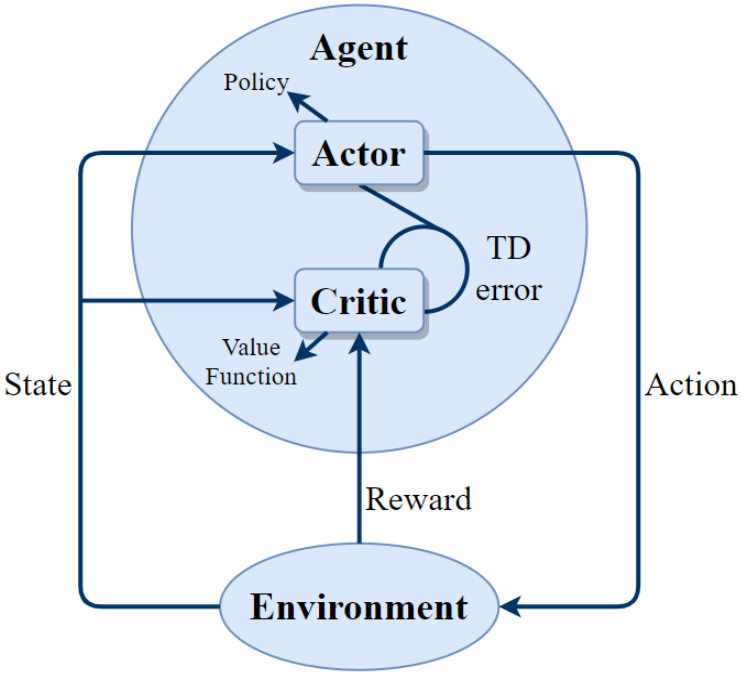
Απαραίτητη προϋπόθεση για την εφαρμογή δυναμικού προγραμματισμού σε προβλήματα ενισχυτικής μάθησης είναι η πλήρης παρατηρησιμότητα του περιβάλλοντος, δηλαδή η πλήρης γνώση του μοντέλου μετάβασης. Ωστόσο, η αυστηρή αυτή προϋπόθεση συνήθως παραβιάζεται σε πολλές πρακτικές περιπτώσεις. Το γεγονός αυτό, σε συνδυασμό με τις υψηλές απαιτήσεις σε μνήμη, ακόμα και για μέτριου μεγέθους χώρο καταστάσεων, αλλά και σε χρόνο, όταν πρόκειται για προβλήματα μεγάλης κλίμακας, καθιστά περιορισμένη τη χρήση του στην ενισχυτική μάθηση. Παρ' όλα αυτά ο δυναμικός προγραμματισμός αποτελεί βάση για την ανάπτυξη αλγορίθμων που μπορούν να εφαρμοστούν σε ένα ευρύ φάσμα προβλημάτων ενισχυτικής μάθησης.

2.5.3 Βαθιά Ενισχυτική Μάθηση

Η Βαθιά Ενισχυτική Μάθηση (Deep Reinforcement Learning - DRL) συνδυάζει την ενισχυτική μάθηση με τα νευρωνικά δίκτυα. Συγκεκριμένα, τα νευρωνικά δίκτυα εκπαιδεύονται προκειμένου να μάθουν να υπολογίζουν τις value-functions. Η χρήση νευρωνικών δικτύων επιτρέπει την εφαρμογή μεθόδων ενισχυτικής μάθησης σε περιπτώσεις που δεν είναι εφικτή ή πρακτική η χρήση δυναμικού προγραμματισμού, όταν για παράδειγμα δεν είναι γνωστό το μοντέλο μετάβασης του περιβάλλοντος, ο χώρος καταστάσεων είναι μη πεπερασμένος ή εξαιτίας υψηλού υπολογιστικού κόστους. Επιπλέον, τα νευρωνικά δίκτυα μπορούν να υπολογίζουν περίπλοκες αναπαραστάσεις και παρέχουν τη δυνατότητα γενίκευσης, δηλαδή πρόβλεψης της value-function πρωτόγνωρων καταστάσεων.

Μέθοδοι Actor-Critic

Η αρχιτεκτονική actor-critic συνδυάζει τις βασισμένες στην πολιτική και τις βασισμένες στην τιμή τεχνικές. Περιλαμβάνει δύο μοντέλα, τον δράστη (actor), που καθορίζει τη συ-



Σχήμα 2.9: *Αρχιτεκτονική actor-critic*

μπεριφορά του πράκτορα και τον κριτή (critic), που αποτιμά την πολιτική του δράστη. Συγκεκριμένα, ο δράστης είναι υπεύθυνος για την επιλογή της δράσης του πράκτορα και τη βελτίωση της πολιτικής σύμφωνα με τις οδηγίες του κριτή. Ο κριτής αποτιμά τις δράσεις που επιλέγει ο δράστης υπολογίζοντας τη value-function και τον καθοδηγεί προκειμένου να βελτιώσει την πολιτική του. Συνήθως, τα μοντέλα του δράστη και του κριτή υλοποιούνται μέσω νευρωνικών δικτύων.

2.5.4 Ο αλγόριθμος DDPG

Ο αλγόριθμος Deep Deterministic Policy Gradients (DDPG) είναι ένας online, off-policy, model-free αλγόριθμος βαθιάς ενισχυτικής μάθησης που βασίζεται στην actor-critic αρχιτεκτονική [26]. Πρόκειται για έναν αλγόριθμο που χρησιμοποιεί την εξίσωση Bellman και δεδομένα που συλλέγει off-policy για να υπολογίσει την action-value function (Q-function) και στη συνέχεια, χρησιμοποιεί την Q-function προκειμένου να μάθει τη βέλτιστη πολιτική.

Ο αλγόριθμος DDPG είναι ειδικά σχεδιασμένος για συνεχείς χώρους δράσεων. Ως γνωστόν, με δεδομένη σε κάθε κατάσταση τη βέλτιστη Q-function, $Q^*(s,a)$, η βέλτιστη δράση προκύπτει ως $a^*(s) = \text{argmax}_a Q^*(a,s)$. Σε πεπερασμένους χώρους διακριτών δράσεων, η ανάδειξη της καλύτερης δράσης μεταξύ όλων γίνεται εύκολα, καθώς η Q-function μπορεί να υπολογιστεί για κάθε δράση ξεχωριστά. Σε συνεχείς χώρους δράσεων αυτό δεν είναι εφικτό, οπότε η βέλτιστη Q-function μπορεί να υπολογιστεί μόνο κατά προσέγγιση.

Η συνέχεια στον χώρο των δράσεων συνεπάγεται τη διαφορισιμότητα της Q-function ως προς την δράση. Ο αλγόριθμος DDPG παρέχει μια μέθοδο εκμάθησης που αξιοποιεί τη δυνατότητα αυτή της διαφορισιμότητας για τον προσεγγιστικό υπολογισμό της βέλτιστης Q-function και κατ' επέκταση, για την προσέγγιση της βέλτιστης πολιτικής. Οι υπολογισμοί αυτοί γίνονται μέσω νευρωνικών δικτύων.

Αρχικός στόχος του αλγορίθμου είναι η εκπαίδευση ενός νευρωνικού δικτύου $Q_\phi(s,a)$, με παραμέτρους ϕ, ώστε να μπορεί να υπολογίζει μια προσέγγιση της Q-function που να ικανοποιεί όσο το δυνατόν περισσότερο την εξίσωση Bellman. Η βέλτιστη Q-function υπολογίζεται σύμφωνα με την εξίσωση Bellman ως εξής:

$$Q^*(s,a) = \mathbb{E}_{s' \sim P}\left[r(s,a) + \gamma \max_{a'} Q^*(s',a')\right] \tag{2.21}$$

όπου γ ο παράγοντας μείωσης. Επομένως, πρέπει να ελαχιστοποιηθεί το σφάλμα Mean Squared Bellman Error (MSBE):

$$L(\phi,\mathcal{D}) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\phi(s,a) - \left(r(s,a) + \gamma(1-d)\max_{a'} Q_\phi(s',a')\right)\right)^2\right] \tag{2.22}$$

όπου $\mathcal{D}$ είναι ο Replay Buffer, μια δομή δεδομένων στην οποία, κατά τη διάρκεια της εκπαίδευσης, αποθηκεύεται η εμπειρία του πράκτορα ως ένα σύνολο μεταβάσεων (s,a,r,s',d) και το d δηλώνει αν η s′ είναι τερματική ή όχι.

Ο όρος $y = r(s,a)+\gamma(1-d)\max_{a'} Q_\phi(s',a')$ καλείται στόχος (target), καθώς το ζητούμενο είναι η $Q_\phi(s,a)$ να τον πλησιάσει όσο το δυνατόν περισσότερο, ώστε να ελαχιστοποιηθεί το σφάλμα. Ωστόσο, ο όρος αυτός εξαρτάται από τις παραμέτρους φ, το οποίο σημαίνει ότι είναι μεταβαλλόμενος και συνεπώς καθιστά ασταθή την ελαχιστοποίηση. Για τον λόγο αυτό χρησιμοποιείται ένα επιπλέον δίκτυο που λειτουργεί ως δίκτυο-στόχος (target-critic network), με παραμέτρους ϕ_{targ}, οι οποίες υστερούν ελάχιστα χρονικά από αυτές του κανονικού δικτύου:

$$\phi_{targ} \leftarrow \tau\phi_{targ} + (1-\tau)\phi \tag{2.23}$$

όπου τ μια πραγματική σταθερά στο διάστημα $[0,1]$. Επομένως, το σφάλμα διαμορφώνεται ως εξής:

$$L(\phi,\mathcal{D}) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\phi(s,a) - \left(r(s,a) + \gamma(1-d)\max_{a'} Q_{\phi_{targ}}(s',a')\right)\right)^2\right] \tag{2.24}$$

Δεδομένου όμως ότι το $\max_{a'} Q_{\phi_{targ}}(s',a')$ δεν μπορεί να υπολογιστεί σε συνεχείς χώρους καταστάσεων, εισάγεται ένα επιπλέον δίκτυο $\mu_{\theta_{targ}}(s)$, με παραμέτρους θ_{targ}, το οποίο εκφράζει την πολιτική-στόχο (target-actor network) και υπολογίζει την δράση που προσεγγιστικά μεγιστοποιεί το $Q_{\phi_{targ}}$. Επομένως, τελικά, πρέπει να ελαχιστοποιηθεί το εξής σφάλμα:

$$L(\phi,\mathcal{D}) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\phi(s,a) - \left(r(s,a) + \gamma(1-d)\max_{a'} Q_{\phi_{targ}}\left(s',\mu_{\theta_{targ}}(s')\right)\right)\right)^2\right] \tag{2.25}$$

Η ελαχιστοποίηση της $L(\phi,\mathcal{D})$ πραγματοποιείται με την μέθοδο της κατάβασης πλαγιάς (gradient-descent), υπολογίζοντας το gradient ως προς ϕ.

Στη συνέχεια, στόχος του αλγορίθμου είναι να βρεθεί η πολιτική που δίνει την δράση που μεγιστοποιεί το $Q_\phi(s,a)$. Το δίκτυο πολιτικής (actor network), $\mu_\theta(s)$, με παραμέτρους θ, μεγιστοποιεί προσεγγιστικά το Q_ϕ:

$$J(\theta) = \max_{\theta} \mathbb{E}_{s\sim\mathcal{D}} Q_\phi(s,\mu_\theta(s)) \tag{2.26}$$

Η μεγιστοποίηση της συνάρτησης $J(\vartheta)$ πραγματοποιείται μέσω της μεθόδου της Ανάβασης Πλαγιάς (gradient ascent), υπολογίζοντας το gradient ως προς ϑ.

Οι παράμετροι των δικτύων actor και target-actor συνδέονται μεταξύ τους μέσω της σχέσης:

$$\vartheta_{targ} \leftarrow \tau\vartheta_{targ} + (1-\tau)\vartheta \tag{2.27}$$

2.6 Ενισχυτική Μάθηση Πολλαπλών Πρακτόρων

Η Ενισχυτική Μάθηση Πολλαπλών Πρακτόρων (Multi-agent Reinforcement Learning - MARL) αποτελεί ένα υποπεδίο της Ενισχυτικής Μάθησης που περιλαμβάνει τις περιπτώσεις που πολλαπλοί πράκτορες συνυπάρχουν και συντονίζουν τη δράση τους σε ένα κοινό περιβάλλον, προς την επίτευξη ενός συλλογικού ή ατομικού στόχου. Τέτοιου είδους προβλήματα παρουσιάζουν υψηλή πολυπλοκότητα, η οποία συνήθως αυξάνεται με την αύξηση του πλήθους των πρακτόρων. Συνεπώς, η ανάπτυξη μεθόδων μάθησης καθίσταται ιδιαίτερα απαιτητική.

Ένα επιπλέον ζήτημα που καθιστά απαιτητική την ανάπτυξη μεθόδων μάθησης είναι η μη στασιμότητα (non-stationarity) του περιβάλλοντος, η οποία οφείλεται στο γεγονός ότι πολλοί πράκτορες αλληλεπιδρούν μεταξύ τους και μαθαίνουν, δηλαδή μεταβάλλουν την συμπεριφορά τους, ταυτόχρονα στο ίδιο περιβάλλον. Ως εκ τούτου, από την οπτική γωνία κάθε πράκτορα, το περιβάλλον παύει να έχει τη μαρκοβιανή ιδιότητα και ο καθένας πρέπει να λαμβάνει υπόψιν του εκτός από τα αποτελέσματα των δικών του δράσεων, τη συμπεριφορά των υπολοίπων πρακτόρων και τον τρόπο που οι δικές του δράσεις επηρεάζουν και επηρεάζονται από τις δράσεις των άλλων.

Σε αυτό το πλαίσιο, είναι καθοριστικής σημασίας ο προσδιορισμός του στόχου των πρακτόρων. Σε περίπτωση που οι πράκτορες επιδιώκουν έναν κοινό στόχο, οφείλουν να συνεργάζονται μεταξύ τους προς τη βελτίωση της συλλογικής επίδοσης, δηλαδή να προσέχουν οι δράσεις τους να μην έχουν αρνητικό αντίκτυπο στη δράση των υπολοίπων. Αντιθέτως, αν οι πράκτορες έχουν αντικρουόμενους στόχους, τότε δρουν με γνώμονα το προσωπικό τους συμφέρον και ανταγωνίζονται μεταξύ τους. Επίσης, υπάρχει το ενδεχόμενο ένα πρόβλημα να συνδυάζει και τις δύο περιπτώσεις, περιλαμβάνοντας πράκτορες που επιδιώκουν να βελτιώσουν την προσωπική τους επίδοση, έχοντας όμως παράλληλα έναν κοινό στόχο.

2.6.1 ΜΔΑ για την Ενισχυτική Μάθηση Πολλαπλών Πρακτόρων

Παρόμοια με την ενισχυτική μάθηση ενός πράκτορα, η πολυπρακτορική ενισχυτική μάθηση μπορεί να μοντελοποιηθεί ως μια Μαρκοβιανή Διαδικασία Απόφασης. Βάση των μοντέλων ενισχυτικής μάθησης πολλαπλών πρακτόρων αποτελεί το Παίγνιο Markov (Markov Game), το οποίο επεκτείνει την ΜΔΑ για τις περιπτώσεις πολλών πρακτόρων [27]. Το Παίγνιο Markov ορίζεται ως μια πλειάδα $(\mathcal{N}, \mathcal{S}, \mathcal{A}, T, R_i)$, όπου:

- $\mathcal{N}$: είναι το σύνολο των n πρακτόρων, $\mathcal{N} = \{1, \ldots, n\}$
- $\mathcal{S}$: το σύνολο των καταστάσεων

- $\mathcal{A} = \{A_1, \ldots, A_N\}$ το σύνολο των δράσεων όλων των πρακτόρων, όπου A_i το σύνολο των δράσεων του πράκτορα $i \in \mathcal{N}$
- $T : S \times A_1 \times, \ldots, \times A_n \to [0, 1]$ η συνάρτηση μετάβασης του περιβάλλοντος
- $R_i : S \times A_1 \times, \ldots, \times A_n \to \mathbb{R}$ η συνάρτηση ανταμοιβής του πράκτορα i

Κάθε χρονική στιγμή t κάθε πράκτορας i παρατηρεί την κατάσταση του περιβάλλοντος s_t και εκτελεί την δράση a_{it} που υπαγορεύει η πολιτική του $\pi_i : S \times A_i \to [0, 1]$. Στη συνέχεια, το περιβάλλον αποδίδει στον πράκτορα μια ανταμοιβή r_i και μεταβαίνει στην επόμενη κατάσταση s_{t+1} με την πιθανότητα που υπολογίζει η συνάρτηση μετάβασης T, δεδομένης της τρέχουσας κατάστασης s_t και των δράσεων όλων των πρακτόρων $a_{1t}, \ldots, a_{nt}$. Στόχος κάθε πράκτορα είναι η μεγιστοποίηση της αναμενόμενης συνολικής ανταμοιβής του.

2.6.2 Συγκεντρωτικός και Αποκεντρωμένος σχεδιασμός

Οι δύο βασικές προσεγγίσεις για τον σχεδιασμό των μεθόδων ενισχυτικής μάθησης είναι η συγκεντρωτική (centralized) και η αποκεντρωμένη (decentralized) εκπαίδευση. Στην centralized εκπαίδευση, υπάρχει ένας κεντρικός πράκτορας (central agent) επιφορτισμένος με τη λήψη των αποφάσεων, ο οποίος συγκεντρώνει πληροφορίες από όλους τους πράκτορες και υπαγορεύει τις δράσεις του καθενός. Στην decentralized εκπαίδευση, οι πράκτορες μαθαίνουν ανεξάρτητα ο ένας από τον άλλον, χωρίς ρητές επικοινωνιακές ενέργειες για ανταλλαγή πληροφοριών και λαμβάνουν μόνοι τους τις αποφάσεις για τις δικές τους δράσεις. Με βάση αυτές τις προσεγγίσεις, έχουν αναπτυχθεί οι ακόλουθες τεχνικές [5]:

Centralized Training with Centralized Execution

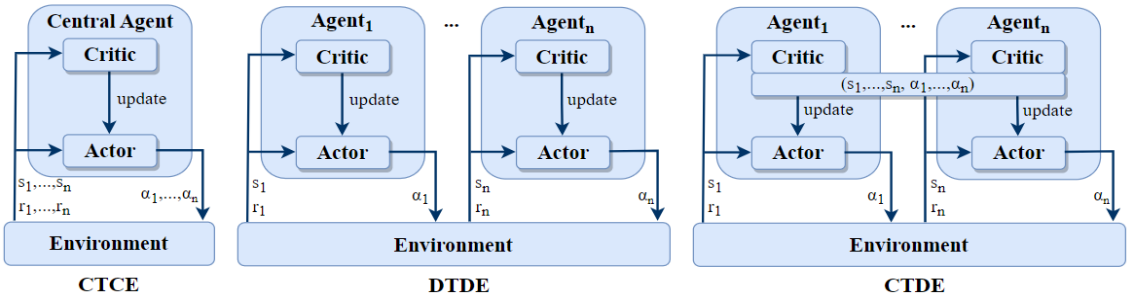
Στην τεχνική που περιλαμβάνει centralized training και centralized execution (CTCE) υπάρχει ένας κεντρικός πράκτορας που συγκεντρώνει τις παρατηρήσεις όλων των πρακτόρων, συνθέτει μια ενιαία παρατήρηση και διαμορφώνει μια κοινή πολιτική για όλους τους πράκτορες. Η πολιτική αυτή αντιστοιχίζει την ενιαία παρατήρηση σε ένα σύνολο κατανομών πιθανότητας για τις ατομικές δράσεις κάθε πράκτορα.

Decentralized Training with Centralized Execution

Στην τεχνική που περιλαμβάνει decentralized training και decentralized execution (DTCE) κάθε πράκτορας μαθαίνει ανεξάρτητα από τους υπόλοιπους και αναπτύσσει μια προσωπική πολιτική, η οποία αντιστοιχίζει τις δικές του παρατηρήσεις σε μια κατανομή πιθανότητας που αφορά τις δικές του δράσεις. Οι πράκτορες δεν ανταλλάσσουν πληροφορίες μεταξύ τους και δεν αντιλαμβάνονται τη δράση των υπολοίπων.

Centralized Training with Decentralized Execution

Μια ευρέως χρησιμοποιούμενη τεχνική είναι αυτή που συνδυάζει centralized training με decentralized execution (CTDE). Κάθε πράκτορας αναπτύσσει τη δική του ατομική πολιτική, η οποία αντιστοιχίζει τις παρατηρήσεις του με μια κατανομή πιθανότητας που αφορά τις δικές του πιθανές δράσεις. Ωστόσο, κατά τη διάρκεια της εκπαίδευσης, οι πράκτορες έχουν στη διάθεσή τους επιπλέον πληροφορίες, που περιλαμβάνουν τις παρατηρήσεις και τις δράσεις



Σχήμα 2.10: *Αρχιτεκτονικές Ενισχυτικής Μάθησης Πολλαπλών Πρακτόρων*

όλων των πρακτόρων. Ως εκ τούτου, διαμορφώνεται μια συλλογική εκτίμηση, η οποία όμως αξιοποιείται αποκλειστικά για την εκπαίδευση.

Η φάση της εκτέλεσης συντελείται ανεξάρτητα για κάθε πράκτορα και οι πληροφορίες που λαμβάνονται υπόψιν από τον καθένα περιλαμβάνουν αυστηρά μόνο τις δικές του παρατηρήσεις. Ο πλέον χαρακτηριστικός αλγόριθμος της κατηγορίας αυτής είναι ο αλγόριθμος MADDPG.

2.7 Ο αλγόριθμος MADDPG

Ο αλγόριθμος Multi-Agent Deep Deterministic Policy Gradients (MADDPG) είναι μια actor-critic μέθοδος, που επεκτείνει τον αλγόριθμο DDPG για την επίλυση προβλημάτων πολλαπλών πρακτόρων και ανήκει στην κατηγορία τεχνικών Centralized Training with Decentralized Execution (CTDE) [4].

Όπως και στον αλγόριθμο DDPG, κάθε πράκτορας διαθέτει ένα δίκτυο κριτή (critic) για τον υπολογισμό της Q-function και ένα δίκτυο δράστη (actor) για τον υπολογισμό της πολιτικής. Παράλληλα, χρησιμοποιούνται ένα δίκτυο-στόχος κριτή (target-critic) και ένα δίκτυο-στόχος δράστη (target-actor), καθώς και ένας replay buffer, στον οποίο αποθηκεύεται η εμπειρία των πρακτόρων με τη μορφή μεταβάσεων. Το critic δίκτυο χρησιμοποιείται μόνο κατά τη διάρκεια της εκπαίδευσης, λειτουργώντας επικουρικά στην εκμάθηση της πολιτικής του πράκτορα. Κατά την πραγματική εκτέλεση, για την επιλογή της δράσης χρησιμοποιείται αποκλειστικά το actor δίκτυο.

Το critic δίκτυο είναι centralized, δηλαδή για να υπολογίσει τη Q-function του πράκτορα λαμβάνει υπόψιν πληροφορίες που προέρχονται από όλους τους πράκτορες και όχι μόνο από τον ίδιο. Έτσι, η πολιτική κάθε πράκτορα αξιολογείται σύμφωνα με τις παρατηρήσεις και τις δράσεις όλων των πρακτόρων. Αντιθέτως, το actor δίκτυο είναι decentralized, δηλαδή έχει πρόσβαση μόνο στις ατομικές πληροφορίες του πράκτορα και αποφασίζει αυτόνομα τις δράσεις που θα εκτελεστούν, εφαρμόζοντας την εξατομικευμένη πολιτική του.

Έστω ένα περιβάλλον με N πράκτορες. Κάθε πράκτορας i διαμορφώνει τη δική του πολιτική μ_i, μέσω του actor, το οποίο δέχεται ως είσοδο την παρατήρηση-κατάσταση του πράκτορα, s_i και επιστρέφει την προς εκτέλεση δράση, $a_i = \mu_i(s_i; \theta_i)$, όπου θ_i οι παράμετροι του δικτύου. Το critic κάθε πράκτορα είναι παραμετροποιημένο ως προς ϕ_i και με είσοδο την συνολική κατάσταση $\mathbf{S} = (s_1, \ldots, s_N)$ και τις δράσεις όλων των πρακτόρων, υπολογίζει την Q-function, $Q_i(\mathbf{S}, a_1, \ldots, a_N; \phi_i)$ του πράκτορα. Επιπλέον, κάθε πράκτορας έχει τη δική

του συνάρτηση ανταμοιβής, r_i, η οποία επιστρέφει την άμεση ανταμοιβή κάθε μετάβασης.

Εκπαίδευση

Κατά την εκπαίδευση, κάθε πράκτορας i επιλέγει μια δράση $a_i = \mu_i(s_i; \theta_i)$ και υπολογίζει την αξία της, $Q_i(\mathbf{S}, a_1 \ldots, a_N; \phi_i)$. Ακολούθως, το περιβάλλον μεταβαίνει στην κατάσταση $\mathbf{S'}$ και αποδίδει στον πράκτορα την άμεση ανταμοιβή του r_i. Η πλειάδα $(\mathbf{S}, \mathbf{A}, \mathbf{R}, \mathbf{S'}, \mathbf{d})$ αποθηκεύεται στον replay buffer $\mathcal{D}$, όπου $\mathbf{A} = \{a_1, \ldots, a_N\}$, $\mathbf{R} = \{r_1, \ldots, r_N\}$ και το $\mathbf{d}$ είναι ένα διάνυσμα που δηλώνει αν η $\mathbf{S'}$ είναι τερματική ή όχι.

Στη συνέχεια, για έναν ορισμένο αριθμό επαναλήψεων, κάθε πράκτορας επιλέγει τυχαία ένα σύνολο M μεταβάσεων $(\mathbf{S}^j, \mathbf{A}^j, \mathbf{R}^j, \mathbf{S'}^j)$ από τον replay buffer $\mathcal{D}$ και ενημερώνει τα actor και critic δίκτυα.

Ενημέρωση του critic

Η εκπαίδευση του critic πραγματοποιείται με στόχο τη μείωση της αναμενόμενης τιμής του σφάλματος ανάμεσα στις υπολογισθείσες από το δίκτυο Q-τιμές και την επιδιωκόμενη τιμή στόχο $y_i^j = r_i^j + \gamma(1-d)Q_i^-(\mathbf{s'}^j, a'_1, \ldots, a'_N|_{a'_k=\mu_k^-(s_k'^j;\theta_k^-)}; \phi_i^-)$, όπου Q_i^- το δίκτυο target-critic, παραμετροποιημένο ως προς ϕ_i^-, μ_k^- η πολιτική-στόχος του k-οστού πράκτορα, παραμετροποιημένη ως προς θ_k^- και γ ο παράγοντας μείωσης:

$$L(\phi_i) = \mathbb{E}_{\mathbf{s},a,\mathbf{r},\mathbf{s'}}\left[\left(y_i^j - Q_i\left(\mathbf{s}^j, a_1, \ldots, a_N; \phi_i\right)\right)^2\right] = \frac{1}{M}\sum_j \left(y_i^j - Q_i\left(\mathbf{s}^j, a_1, \ldots, a_N; \phi_i\right)\right)^2 \quad (2.28)$$

Οι παράμετροι ϕ_i ενημερώνονται με τη μέθοδο του gradient descent προς την κατεύθυνση που μειώνει το σφάλμα.

Ενημέρωση του actor

Η εκπαίδευση του actor πραγματοποιείται με στόχο να βρεθεί η πολιτική που μεγιστοποιεί την συνολική αναμενόμενη ανταμοιβή:

$$J(\theta_i) = \mathbb{E}_{\mathbf{s},a\sim\mathcal{D}}\left[Q_i(\mathbf{s}^j, a_1^j, \ldots, a_i, \ldots, a_N^j|_{a_i=\mu_i(s_i^j;\theta_i)}; \phi_i)\right] \quad (2.29)$$

Οι παράμετροι θ_i ενημερώνονται με τη μέθοδο του gradient ascent προς την κατεύθυνση που αυξάνει την αναμενόμενη ανταμοιβή. Κατά τη διαδικασία αυτή, οι δράσεις των υπολοίπων πρακτόρων θεωρούνται σταθερές και λαμβάνονται από τον replay buffer, $\mathcal{D}$. Επομένως, υπολογίζεται το gradient:

$$\nabla_{\theta_i} J(\theta_i) \approx \frac{1}{M}\sum_j \nabla_{\theta_i}\mu_i(s_i^j)\nabla_{a_i} Q_i(\mathbf{s}^j, a_1^j, \ldots, a_i, \ldots, a_N^j|_{a_i=\mu_i(s_i^j;\theta_i)}; \phi_i) \quad (2.30)$$

Ενημέρωση των παραμέτρων των target δικτύων

Τέλος, οι παράμετροι ϕ_i^- και θ_i^- των δικτύων target-critic και target-actor, αντίστοιχα, ενημερώνονται ως εξής:

$$\phi_i^- \leftarrow \tau\phi_i^- + (1-\tau)\phi_i \quad (2.31)$$

$$\theta_i^- \leftarrow \tau\theta_i^- + (1-\tau)\theta_i \quad (2.32)$$

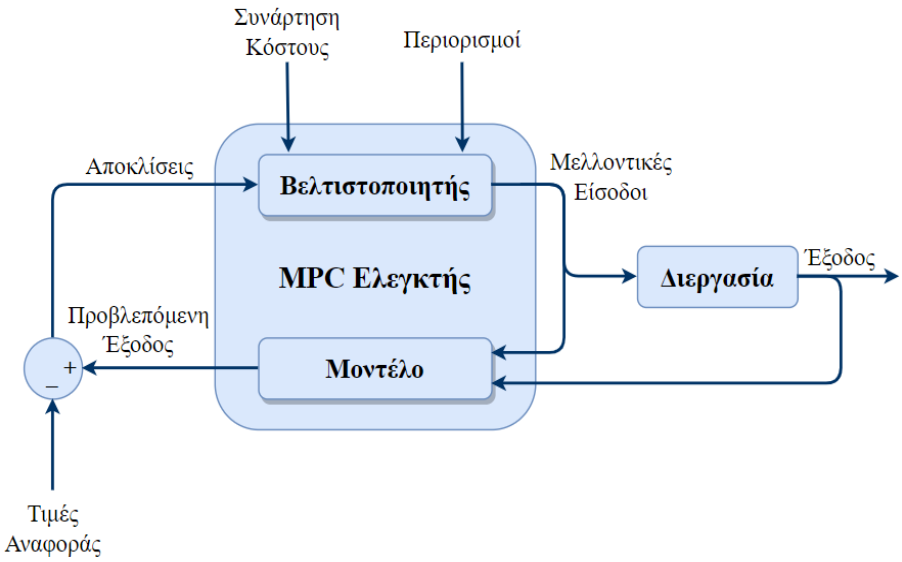
όπου τ μια πραγματική σταθερά στο διάστημα $[0, 1]$.

Στο Α΄.1 του παραρτήματος ο αλγόριθμος MADDPG παρουσιάζεται με τη μορφή ψευδοκώδικα.

2.8 Model Predictive Control

Η Model Predictive Control (MPC) περιλαμβάνει ένα μεγάλο εύρος μεθόδων που βασίζονται στον σχεδιασμό μιας αλληλουχίας ενεργειών ελέγχου ώστε να επιτευχθεί η επιθυμητή συμπεριφορά μιας διεργασίας εντός ενός πεπερασμένου χρονικού ορίζοντα [28]. Οι ενέργειες ελέγχου επανασχεδιάζονται κάθε χρονική στιγμή σύμφωνα με την εκάστοτε κατάσταση του συστήματος. Αυτές οι μέθοδοι μπορούν να διαχειριστούν μη-γραμμικές σχέσεις και περιορισμούς που μεταβάλλονται δυναμικά με τον χρόνο. Είναι ευπροσάρμοστες σε τυχόν αλλαγές των κριτηρίων επίδοσης και λειτουργικές ακόμα κι αν το αρχικό σύστημα παρουσιάζει αβεβαιότητα.

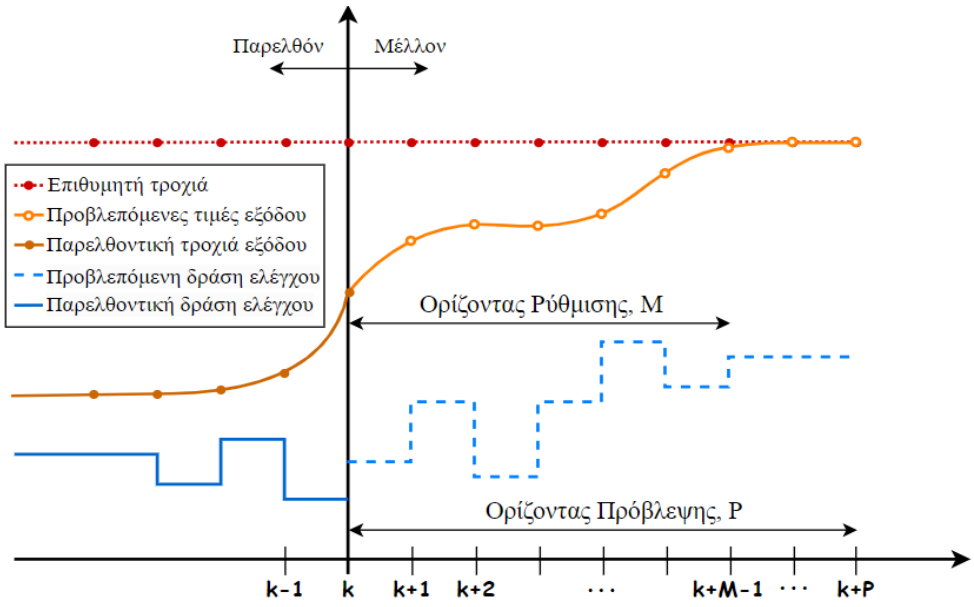
Ένας αλγόριθμος MPC χρησιμοποιεί ένα μοντέλο του συστήματος προκειμένου να προβλέψει τη μελλοντική συμπεριφορά του εντός ενός πεπερασμένου μελλοντικού ορίζοντα, η οποία εκφράζεται ως τροχιά αναφοράς. Η απόκλιση από την επιθυμητή συμπεριφορά υπολογίζεται από μια κατάλληλα ορισμένη συνάρτηση κόστους. Στόχος είναι να υπολογιστεί μια ακολουθία ενεργειών ελέγχου εισόδου, ώστε η παραγόμενη έξοδος να ακολουθεί την επιθυμητή τροχιά αναφοράς όσο το δυνατόν πιο πιστά, δηλαδή να ελαχιστοποιείται η συνάρτηση κόστους.



Σχήμα 2.11: *Διάγραμμα Ροής Λειτουργίας MPC*

Κάθε χρονική στιγμή, ο ελεγκτής λαμβάνει ή υπολογίζει την τρέχουσα κατάσταση του συστήματος και επιλύει το πρόβλημα βελτιστοποίησης ούτως ώστε να βρει τη βέλτιστη αλληλουχία ενεργειών ελέγχου κατά μήκος ενός πεπερασμένου χρονικού άξονα, που ονομάζεται ορίζοντας ρύθμισης. Κατά τη διαδικασία αυτή, εξερευνά όλες τις πιθανές μελλοντικές τροχιές, προβλέποντας τις αποκρίσεις για κάθε διακριτή χρονική στιγμή εντός του χρονικού

ορίζοντα πρόβλεψης. Οι προβλέψεις αυτές γίνονται σύμφωνα με το μοντέλο του συστήματος και την τρέχουσα κατάσταση. Επιλέγεται εκείνη η τροχιά που ελαχιστοποιεί τη συνάρτηση κόστους και στη συνέχεια, από την βέλτιστη αλληλουχία ενεργειών ελέγχου εφαρμόζεται μόνο η πρώτη κατά σειρά, ενώ οι υπόλοιπες απορρίπτονται. Η διαδικασία επαναλαμβάνεται, εκκινώντας από την κατάσταση στην οποία μετέβη το σύστημα ως συνέπεια της ενέργειας αυτής και με μετατόπιση του χρονικού ορίζοντα κατά ένα βήμα.



Σχήμα 2.12: *Σχηματική Αναπαράσταση Λειτουργίας MPC*

Κεφάλαιο 3

Μεθοδολογία

Στο κεφάλαιο αυτό αναλύεται η μεθοδολογία που εφαρμόσαμε για τη δημιουργία δύο κατανεμημένων συστημάτων ελέγχου της φόρτισης και της εκφόρτισης των μπαταριών ενός έξυπνου ηλεκτρικού δικτύου διανομής, με απαραίτητη προϋπόθεση την τήρηση των τεχνικών προδιαγραφών τους. Στόχος μας είναι να ικανοποιούνται οι ενεργειακές ανάγκες του δικτύου και παράλληλα η ισχύς στο PCC να ακολουθεί το προκαθορισμένο σχέδιο-διανομής (dispatch-plan). Το dispatch-plan ορίζει τον τρόπο παραγωγής και διανομής της ηλεκτρικής ενέργειας σε ένα δίκτυο, ώστε να επιτυγχάνεται η ασφαλής και οικονομικά αποδοτικότερη λειτουργία του. Καθορίζεται καθημερινά και συνήθως καλύπτει ένα διάστημα από 1 ώρα έως και 15 λεπτά.

Στην πρώτη ενότητα, διατυπώνεται η μαθηματική μοντελοποίηση του προβλήματος. Στην δεύτερη ενότητα παρουσιάζεται η μέθοδος που αξιοποιεί τη Lagrangian Decomposition προκειμένου το πρόβλημα να διαχωριστεί σε απλούστερα υποπροβλήματα και περιγράφεται η μοντελοποίηση του κάθε υποπροβλήματος ως ΜΔΑ και ο τρόπος υλοποίησης του MADDPG. Τέλος, στην τρίτη ενότητα παρουσιάζεται η επίλυση του προβλήματος χωρίς τη Lagrangian Decomposition, με απλή εφαρμογή του MADDPG.

3.1 Μοντελοποίηση του Προβλήματος

Τα δίκτυα που εξετάζουμε στην παρούσα εργασία αποτελούνται από N_B μπαταρίες και ένα μοναδικό PCC για τη σύνδεση με το δίκτυο μεταφοράς. Επίσης, θεωρούμε T διακριτά χρονικά βήματα $t \in \{1, \dots, T\}$, διάρκειας Δt το καθένα. Για τη μαθηματική διατύπωση του προβλήματος χρησιμοποιούνται οι παρακάτω συμβολισμοί, όπου το t εκφράζει χρονική εξάρτηση.

- $P_{disp}(t)$: Ισχύς αναφοράς στο PCC
- $P(t)$: Πραγματική ισχύς του δικτύου διανομής στο PCC
- $L(t)$: Συνολικό ενεργό φορτίο του δικτύου, χωρίς να συνυπολογίζεται η συνεισφορά των μπαταριών στην ισχύ.
- $B(t)$: Ενεργός ισχύς της μπαταρίας.
- $SoE(t)$: Η κατάσταση της ενέργειας της μπαταρίας.

- B^{min}: Ελάχιστη επιτρεπόμενη τιμή της ενεργού ισχύος της μπαταρίας, $B^{min} \leq 0$.
- B^{max}: Μέγιστη επιτρεπόμενη τιμή της ενεργού ισχύος της μπαταρίας, $B^{max} \geq 0$.
- SoE^{min}: Ελάχιστο όριο για τη *SoE* της μπαταρίας.
- SoE^{max}: Μέγιστο όριο για τη *SoE* της μπαταρίας.
- $SoE^{min,T}$: Ελάχιστο όριο για τη *SoE* της μπαταρίας, την τελική χρονική στιγμή, T.
- $SoE^{max,T}$: Μέγιστο όριο για τη *SoE* της μπαταρίας, την τελική χρονική στιγμή, T.

Για τα όρια της SoE της μπαταρίας ισχύει η σχέση $SoE^{min} \leq SoE^{min,T} \leq SoE^{max,T} \leq SoE^{max}$. Επίσης, οι ισχύες $P_{disp}(t)$ και $P(t)$ είναι θετικές όταν η ισχύς ρέει από το δίκτυο μεταφοράς (transmission grid) προς το δίκτυο διανομής.

Το πρόβλημα ανήκει στην κατηγορία των προβλημάτων κυρτής βελτιστοποίησης και για κάθε χρονική στιγμή t, εκφράζεται μαθηματικά ως εξής:

$$\min_{P(t),B_i(t),\ldots,B_{N_B}(t)} \sum_{t=1}^{T} (P(t) - P_{disp}(t))^2 \tag{3.1}$$

$$\text{τ.ω. } P(t) - \sum_{i=1}^{N_B} B_i(t) - L(t) = 0 \tag{3.2}$$

$$SoE_i(t+1) = SoE_i(t) + B_i(t)\Delta t \tag{3.3}$$

$$B_i^{min} \leq B_i(t) \leq B_i^{max} \tag{3.4}$$

$$SoE_i^{min} \leq SoE_i(t) \leq SoE_i^{max} \tag{3.5}$$

$$SoE_i^{min,T} \leq SoE_i(T) \leq SoE_i^{max,T} \tag{3.6}$$

$$\text{για } i \in \{1,\ldots,N_B\}, t \in \{1,\ldots,T-1\} \tag{3.7}$$

Στην εξίσωση 3.1 αποτυπώνεται ο στόχος εύρεσης των κατάλληλων τιμών για τις μεταβλητές απόφασης $P(t)$ και $B_i(t)$ ώστε να επιτυγχάνεται η ελαχιστοποίηση της αντικειμενικής συνάρτησης $\sum_{t=1}^{T}(P(t) - P_{disp}(t))^2$, δηλαδή του αθροίσματος των αποκλίσεων της ισχύος που μετράται στο PCC κάθε χρονική στιγμή από την αντίστοιχη ισχύ που υπαγορεύει το dispatch-plan. Η εξίσωση 3.2 αποτελεί την εξίσωση του ισοζυγίου ισχύος. Η εξίσωση 3.3 περιγράφει τον τρόπο με τον οποίο ενημερώνεται η τιμή της SoE κάθε χρονική στιγμή, δεδομένου ότι είναι γνωστή η αρχική τιμή της. Ο περιορισμός 3.4 οριοθετεί την ισχύ φόρτισης/εκφόρτισης κάθε μπαταρίας και ο περιορισμός 3.5 υποδεικνύει το επιτρεπόμενο εύρος για την SoE της. Τέλος, με τον περιορισμό 3.6 οριοθετείται το φορτίο που δύναται να έχει η μπαταρία στο τέλος της περιόδου ελέγχου προκειμένου να μπορεί να χρησιμοποιηθεί αποτελεσματικά κατά την επόμενη περίοδο.

3.2 Εφαρμογή της μεθόδου Lagrangian Decomposition

Το ισοζύγιο ισχύος που εκφράζεται στην εξίσωση 3.2, περιέχει αλληλεξαρτήσεις μεταξύ των μεταβλητών απόφασης, $P(t)$ και $B_i(t)$ και συνιστά τον μοναδικό περίπλοκο περιορισμό του προβλήματος. Προκειμένου να «χαλαρώσουμε» τον περιορισμό αυτό και να μπορέσουμε

να αποσυνθέσουμε το πρόβλημα σε μικρότερα υποπροβλήματα, εφαρμόζουμε την τεχνική Lagrangian Decomposition, όπως περιγράφεται στην ενότητα 2.1. Επομένως, το πρόβλημα αναδιατυπώνεται ως εξής:

$$\begin{aligned}
\min_{P(t),B_i(t),\dots,B_{N_B}(t)} & \sum_{t=1}^{T}\left[\Big(P(t)-P_{disp}(t)\Big)^2+\lambda(t)(P(t)-\sum_{i=1}^{N_B}B_i(t)-L(t))\right] \\
\text{τ.ω. } & SoE_i(t+1)=SoE_i(t)+B_i(t)\Delta t \\
& B_i^{min}\le B_i(t)\le B_i^{max} \\
& SoE_i^{min}\le SoE_i(t)\le SoE_i^{max} \\
& SoE_i^{min,T}\le SoE_i(T)\le SoE_i^{max,T} \\
& \text{για } i\in\{1,\dots,N_B\}, t\in\{1,\dots,T-1\}
\end{aligned} \tag{3.8}$$

Αυτή η μορφή του προβλήματος επιτρέπει την αποσύνθεσή του ως προς τις μεταβλητές απόφασης σε N_B+1 υποπροβλήματα, ένα για καθεμία από τις N_B μπαταρίες και ένα για το DSO. Όλα τα υποπροβλήματα συνδέονται μεταξύ τους μέσω του πολλαπλασιαστή Lagrange, ο οποίος είναι κοινός για όλες τις μπαταρίες και για το DSO.

Υποπρόβλημα DSO

Το DSO καλείται κάθε χρονική στιγμή t να υπολογίσει την ισχύ $P(t)$ στο PCC ώστε να ελαχιστοποιείται η απόκλιση από το dispatch-plan. Το υποπρόβλημα για το DSO τη χρονική στιγμή t ορίζεται ως εξής:

$$\min_{P(t)} \sum_{t=1}^{T}\left[\Big(P(t)-P_{disp}(t)\Big)^2+\lambda(t)(P(t)-L(t))\right] \tag{3.9}$$

Υποπρόβλημα Μπαταριών

Κάθε μπαταρία i καλείται κάθε χρονική στιγμή να αποφασίσει την βέλτιστη ισχύ φόρτισης/εκφόρτισης για την ίδια, $B_i(t)$. Το υποπρόβλημα κάθε μπαταρίας $i\in\{1,\dots,N_B\}$ τη χρονική στιγμή t ορίζεται ως εξής:

$$\min_{B_i(t)} -\sum_{t=1}^{T}\lambda(t)B_i(t) \tag{3.10}$$

$$\begin{aligned}
\text{τ.ω. } & SoE_i(t+1)=SoE_i(t)+B_i(t)\Delta t \\
& B_i^{min}\le B_i(t)\le B_i^{max} \\
& SoE_i^{min}\le SoE_i(t)\le SoE_i^{max} \\
& SoE_i^{min,T}\le SoE_i(T)\le SoE_i^{max,T} \\
& \text{για } t\in\{1,\dots,T-1\}
\end{aligned} \tag{3.11}$$

Πολλαπλασιαστής Lagrange

Ο πολλαπλασιαστής Lagrange ανταλλάσσεται κάθε χρονική στιγμή μεταξύ του DSO και των μπαταριών, αποτελώντας το μέσο σύνδεσης και συντονισμού των διαφορετικών υποπροβλημάτων. Τη χρονική στιγμή t το DSO λαμβάνει τις ισχύς φόρτισης/εκφόρτισης όλων των

μπαταριών και με δεδομένη την τιμή του συνολικού φορτίου $L(t)$ ανανεώνει την τιμή του πολλαπλασιαστή Lagrange, εφαρμόζοντας επαναληπτικά την παρακάτω εξίσωση:

$$\lambda^{(\eta+1)}(t) = \lambda^{(\eta)}(t) + a^{(\eta)}(P^{(\eta)}(t) - \sum_{i=1}^{N_B} B_i^{(\eta)}(t) - L(t)) \tag{3.12}$$

όπου $\eta \in \{0, 1, \ldots, H_{max}\}$ το βήμα της επανάληψης, με H_{max} το συνολικό πλήθος των επαναλήψεων και a^{η} το βήμα ενημέρωσης, δηλαδή ο ρυθμός με τον οποίο μεταβάλλεται η τιμή του λ. Βασιζόμενοι στη μελέτη [6], θέσαμε $a^{\eta} = \frac{1}{\sqrt{\eta+1}}$. Σε κάθε επανάληψη, η ισχύς στο PCC και οι ισχύες φόρτισης/εκφόρτισης των μπαταριών υπολογίζονται εκ νέου με βάση την νέα τιμή του λ.

Η τιμή που έχει ο πολλαπλασιαστής Lagrange στην αρχή κάθε χρονικού βήματος και πριν την ενημέρωσή του είναι η τιμή που είχε στο αμέσως προηγούμενο χρονικό βήμα, δηλαδή $\lambda^0(t+1) = \lambda^{H_{max}}(t)$.

3.2.1 Επίλυση του Προβλήματος του DSO

Το πρόβλημα του DSO είναι απαλλαγμένο από χρονικές εξαρτήσεις, οπότε μπορεί να λυθεί ανεξάρτητα για κάθε χρονική στιγμή. Επομένως, η βέλτιστη τιμή της ισχύος στο PCC τη χρονική στιγμή t, $P^*(t)$ υπολογίζεται ως εξής:

$$P^*(t) = \underset{P(t)}{\operatorname{argmin}}\{(P(t) - P_{disp}(t))^2 + \lambda(t)(P(t) - \widetilde{L}(t))\} \tag{3.13}$$

όπου το $\widetilde{L}(t)$ είναι μια εκτίμηση του πραγματικού φορτίου $L(t)$, η οποία γίνεται διότι το πραγματικό φορτίο δεν είναι γνωστό εκ των προτέρων.

3.2.2 Επίλυση του Προβλήματος των Μπαταριών

Στόχος μας είναι να λύσουμε το πρόβλημα των μπαταριών με χρήση ενισχυτικής μάθησης πολλαπλών πρακτόρων, αντιστοιχίζοντας κάθε μπαταρία σε έναν διαφορετικό πράκτορα. Συγκεκριμένα, θέλουμε να εφαρμόσουμε τον αλγόριθμο MADDPG. Για να είναι όμως εφικτή η χρήση ενισχυτικής μάθησης, θα πρέπει πρώτα να μοντελοποιήσουμε κάθε πρόβλημα βελτιστοποίησης ως ΜΔΑ. Αυτό σημαίνει ότι θα πρέπει να προσδιορίσουμε τις έννοιες της κατάστασης, της δράσης, της ανταμοιβής και του μοντέλου μετάβασης, που στην προκειμένη περίπτωση είναι γνωστό. Επιπλέον, είναι απαραίτητο να δομηθεί κατάλληλα το περιβάλλον που θα αποτελεί τον χώρο δράσης και αλληλεπίδρασης των μπαταριών τόσο μεταξύ τους όσο και με το DSO.

3.2.3 Μοντελοποίηση ως Μαρκοβιανή Διαδικασία Απόφασης

Για την μοντελοποίηση του προβλήματος βελτιστοποίησης της μπαταρίας i ως ΜΔΑ, κατ' αρχάς, ορίζεται η κατάστασή της την χρονική στιγμή t, ως $s_i(t) = (SoE_i(t), \lambda(t), t)$, περιλαμβάνοντας την SoE της, τον πολλαπλασιαστή Lagrange και την εκάστοτε χρονική στιγμή. Η δράση που επιλέγεται να εκτελεστεί την χρονική στιγμή t εκφράζει την ποσότητα ισχύος φόρτισης ή εκφόρτισης της μπαταρίας, δηλαδή αντιστοιχεί στην $B_i(t)$. Όταν είναι $B_i(t) > 0$,

η μπαταρία φορτίζει, ενώ όταν είναι $B_i(t) < 0$, η μπαταρία εκφορτίζει. Η ανταμοιβή που αποδίδεται στην μπαταρία υπολογίζεται ως $r_i(t) = \lambda(t)B_i(t) - \Lambda_i(t)$, όπου $\Lambda_i(t)$ μια συνάρτηση ποινής που επιβάλλεται σε περίπτωση παραβίασης κάποιου από τους περιορισμούς της 3.11. Τέλος, η επόμενη κατάσταση της μπαταρίας είναι η $s_i(t+1) = (SoE_i(t+1), \lambda(t+1), t+1)$. Η $SoE_i(t+1)$ υπολογίζεται σύμφωνα με την εξίσωση 3.3, αφού όμως πρώτα η τιμή του $\lambda(t)$ έχει οριστικοποιηθεί μέσω της 3.12 και το $B_i(t)$ έχει υπολογισθεί βάσει αυτής της τιμής. Το $\lambda(t+1)$ είναι αρχικά ίσο με $\lambda(t)$ και στη συνέχεια ενημερώνεται η τιμή του.

3.2.4 Εφαρμογή του αλγορίθμου MADDPG

Κάθε μπαταρία διαθέτει τον δικό της πράκτορα, ο οποίος παρατηρεί την κατάσταση του περιβάλλοντός της και αποφασίζει την δράση που θα εκτελέσει, δηλαδή την ισχύ με την οποία θα φορτίσει ή θα εκφορτίσει, ακολουθώντας την εξατομικευμένη πολιτική που διαμορφώθηκε κατά την εκπαίδευση του συστήματος. Όπως περιγράφεται στην ενότητα 2.7, κάθε πράκτορας έχει το δικό του decentralized actor δίκτυο, που χρησιμοποιείται για την επιλογή της δράσης του και το δικό του centralized critic δίκτυο το οποίο χρησιμοποιείται μόνο κατά τη διάρκεια της εκπαίδευσης, βοηθώντας το actor στην διαμόρφωση της πολιτικής.

Εκπαίδευση δικτύων

Κατά τη διαδικασία της εκπαίδευσης, ο πράκτορας κάθε μπαταρίας i παρατηρεί την κατάσταση του περιβάλλοντός της τη χρονική στιγμή t, $s_i(t) = (SoE_i(t), \lambda(t), t)$ και τη δίνει ως είσοδο στο actor δίκτυό του, το οποίο επιστρέφει την προς εκτέλεση δράση, σύμφωνα με την πολιτική του, $B_i(t) = \mu_i(s_i(t); \theta_i)$, όπου θ_i οι παράμετροι του δικτύου. Στη συνέχεια, πραγματοποιείται η διαδικασία ενημέρωσης του $\lambda(t)$. Ακολούθως, το περιβάλλον της μπαταρίας αποδίδει στον πράκτορα την ανταμοιβή για τη δράση του, $r_i(t) = \lambda(t)B_i(t) - \Lambda_i(t)$ και μεταβαίνει στην επόμενη κατάσταση, $s_i(t+1) = (SoE_i(t+1), \lambda(t+1), t+1)$. Στην παρούσα φάση ισχύει $\lambda(t+1) = \lambda(t)$.

Οι καταστάσεις, οι δράσεις, οι ανταμοιβές και οι επόμενες καταστάσεις όλων των πρακτόρων συγχωνεύονται και διαμορφώνουν μια πλειάδα, που συνιστά τη συλλογική μετάβαση, $(\mathbf{S}(t), \mathbf{B}(t), \mathbf{R}(t), \mathbf{S}(t+1)) = (\mathbf{S}, \mathbf{B}, \mathbf{R}, \mathbf{S}')$, όπου $\mathbf{S} = \{s_1(t), \ldots, s_{N_B}(t)\}$, $\mathbf{B} = \{B_1(t), \ldots, B_{N_B}(t)\}$, $\mathbf{R} = \{r_1(t), \ldots, r_{N_B}(t)\}$ και $\mathbf{S}' = \{s_1(t+1), \ldots, s_{N_B}(t+1)\}$, η οποία αποθηκεύεται στον κοινό για όλους τους πράκτορες Replay Buffer, $\mathcal{D}$.

Ακολούθως, κάθε πράκτορας επιλέγει τυχαία M μεταβάσεις από τον $\mathcal{D}$ και για κάθε μετάβαση j δίνει την καθολική κατάσταση $\mathbf{S}^j$ και τις δράσεις όλων των πρακτόρων $B_1^j, \ldots, B_{N_B}^j$ ως είσοδο στο critic δίκτυό του, ώστε να υπολογίσει την Q-function, $Q_i^j(\mathbf{S}^j, B_1^j, \ldots, B_{N_B}^j; \phi_i)$, όπου ϕ_i οι παράμετροι του δικτύου.

Ωστόσο, αν χρησιμοποιηθεί ένας μόνο replay buffer, οι πλειάδες που αφορούν τις τελικές μεταβάσεις θα επιλέγονται πολύ σπάνια κατά τη δειγματοληψία, καθώς είναι πολύ λιγότερες αναλογικά με τις υπόλοιπες. Το γεγονός αυτό επηρεάζει καθοριστικά την εκπαίδευση, καθώς οι μεταβάσεις αυτές περιέχουν τις ανταμοιβές που σχετίζονται με τους τελικούς περιορισμούς για την SoE των μπαταριών και η μη επιλογή τους παρεμποδίζει την εκμάθηση της επιθυμητής συμπεριφοράς. Για να αντιμετωπίσουμε αυτό το πρόβλημα, επιλέγουμε να χρησιμοποιήσουμε δύο replay buffers αντί για έναν, τον Replay Buffer $\mathcal{D}$ για την αποθήκευ-

ση των μη-τερματικών μεταβάσεων και τον Terminal Replay Buffer $\mathcal{D}_T$ για την αποθήκευση μόνο των τερματικών μεταβάσεων. Με αυτόν τον τρόπο, κατά τη δειγματοληψία θα μπορούμε να επιλέγουμε τι ποσοστό δειγμάτων θα προέρχεται από κάθε replay buffer εξασφαλίζοντας ότι θα υπάρχουν επαρκή δείγματα εμπειρίας για τις τελικές μεταβάσεις και επιταχύνοντας τη σύγκλιση.

Ενημέρωση παραμέτρων του critic

Οι παράμετροι του critic δικτύου κάθε πράκτορα i ενημερώνονται με τη μέθοδο gradient descent προς την κατεύθυνση που ελαχιστοποιεί το σφάλμα MSE:

$$L(\phi_i) = \frac{1}{M}\sum_{j=1}^{M}\left(y_i^j - \mathrm{Q}_i(\mathbf{S}^j, B_1^j, \ldots, B_{N_B}^j; \phi_i)\right)^2 \tag{3.14}$$

όπου $y_i^j = r_i^j + \gamma(1-d_i)\mathrm{Q}_i^-\left(\mathbf{S}'^j, B'_1, \ldots, B'_{N_B}\,|_{B'_i=\mu_i^-(s_i'^j;\partial_i^-)}; \phi_i^-\right)$ ο όρος-στόχος (target), στον οποίο Q_i^- είναι το target-critic δίκτυο, με παραμέτρους ϕ_i^-, μ_i^- είναι το target-actor δίκτυο, με παραμέτρους ∂_i^-, γ είναι ο παράγοντας μείωσης και d_i το σήμα που δηλώνει αν η $s_i'^j$ είναι τερματική ή όχι.

Ενημέρωση παραμέτρων του actor δικτύου

Οι παράμετροι του actor δικτύου κάθε πράκτορα i ενημερώνονται με τη μέθοδο gradient ascent προς την κατεύθυνση που μεγιστοποιεί την αναμενόμενη συνολική ανταμοιβή:

$$J(\partial_i) = \mathbb{E}_{\mathbf{S},\mathbf{B}\sim\mathcal{D}}\left[\mathrm{Q}_i(\mathbf{S}^j, B_1^j, \ldots, B_i, \ldots, B_N^j\,|_{B_i=\mu_i(s_i^j;\partial_i)}; \phi_i)\right] \tag{3.15}$$

Η δράση του πράκτορα i υπολογίζεται από την τρέχουσα πολιτική του, ενώ οι δράσεις των υπόλοιπων πρακτόρων λαμβάνονται από τον Replay Buffer, $\mathcal{D}$. Το gradient ως προς ∂_i υπολογίζεται ως εξής:

$$\nabla_{\partial_i} J(\partial_i) \approx \frac{1}{M}\sum_{j=1}^{M}\nabla_{\partial_i}\mu_i(s_i^j)\nabla_{B_i}\mathrm{Q}_i(\mathbf{S}^j, B_1^j, \ldots, B_i, \ldots, B_N^j\,|_{B_i=\mu_i(s_i^j;\partial_i)}; \phi_i) \tag{3.16}$$

Ενημέρωση παραμέτρων των target δικτύων

Οι παράμετροι των target-actor και target-critic δικτύων του πράκτορα i ενημερώνονται αντίστοιχα ως εξής:

$$\phi_i^- \leftarrow \tau\phi_i^- + (1-\tau)\phi_i \tag{3.17}$$

$$\partial_i^- \leftarrow \tau\partial_i^- + (1-\tau)\partial_i \tag{3.18}$$

Όπου τ είναι μια πραγματική σταθερά στο διάστημα $[0, 1]$.

3.2.5 Συνολικός Αλγόριθμος

Ο συνολικός αλγόριθμος ελέγχου διαμορφώνεται ως εξής: Κάθε χρονική στιγμή, το πρόβλημα του DSO επιλύεται αυτόνομα σύμφωνα με την εξίσωση 3.13, υπολογίζοντας τη βέλτιστη ισχύ στο PCC. Επίσης, ο πράκτορας κάθε μπαταρίας λαμβάνει από το περιβάλλον

της την τρέχουσα κατάστασή της και επιλέγει μια δράση βάσει αυτής. Ακολουθεί η ενημέρωση του πολλαπλασιαστή Lagrange μέσω του επαναληπτικού υπολογισμού της σχέσης 3.12.

Με βάση την ανανεωμένη τιμή του πολλαπλασιαστή lagrange, επανυπολογίζεται η βέλτιστη τιμή στο PCC και ενημερώνονται οι καταστάσεις και οι δράσεις των μπαταριών. Το περιβάλλον κάθε μπαταρίας λαμβάνοντας την τελική δράση και την τιμή του πολλαπλασιαστή Lagrange, υπολογίζει την ανταμοιβή της και μεταβαίνει στην επόμενη κατάσταση. Στη συνέχεια, οι καταστάσεις, οι δράσεις, οι ανταμοιβές και οι επόμενες καταστάσεις όλων των πρακτόρων συγχωνεύονται και αποθηκεύονται είτε στον Replay Buffer είτε στον Terminal Replay Buffer, ανάλογα με την χρονική στιγμή.

Στο σημείο αυτό, πραγματοποιείται η εκπαίδευση των actor και critic δικτύων των πρακτόρων, δηλαδή η προσαρμογή των παραμέτρων τους προς την επίτευξη της βέλτιστης απόδοσης του συστήματος, όπως περιγράφηκε στην προηγούμενη ενότητα. Ο αλγόριθμος επαναλαμβάνεται μέχρι να επιτευχθεί σύγκλιση.

Μετά το πέρας της εκπαίδευσης, το μοντέλο είναι έτοιμο για χρήση, εκτελώντας ουσιαστικά τον ίδιο αλγόριθμο με εξαίρεση τη φάση της εκπαίδευσης των δικτύων. Οι παράμετροι του actor δικτύου κάθε μπαταρίας έχουν παγιωθεί πλέον στις βέλτιστες τιμές και αποτυπώνουν την πολιτική βάση της οποίας δρα κάθε χρονική στιγμή.

Οι αλγόριθμοι για την εκπαίδευση και την αξιολόγηση του μοντέλου παρουσιάζονται αντίστοιχα με τη μορφή ψευδοκώδικα στα Α΄.2 και Α΄.3, του παραρτήματος.

3.3 Επίλυση χωρίς Lagrangian Decomposition

Στη συνέχεια, εξετάζουμε τη δυνατότητα επίλυσης του προβλήματος με απλή εφαρμογή του αλγορίθμου MADDPG, χωρίς την πρότερη χρήση της μεθόδου Lagrangian Decomposition. Εφαρμόζοντας τον MADDPG, δεδομένου ότι το critic κάθε μπαταρίας είναι centralized και υπολογίζει τη συνάρτηση χρησιμότητας με βάση τις καταστάσεις και τις δράσεις όλων των μπαταριών, θεωρούμε ότι οι μπαταρίες ανταλλάσσουν όλες τις απαραίτητες πληροφορίες μεταξύ τους κατά τη διαδικασία της εκπαίδευσής τους και έχουν πλήρη εικόνα του συστήματος.

Το συνολικό πρόβλημα μοντελοποιείται αρχικά ως ΜΔΑ όπως περιγράφεται στην ενότητα 3.2.3, με τη διαφορά ότι η κατάσταση κάθε μπαταρίας i τη χρονική στιγμή t περιλαμβάνει μόνο τη SoE της και την χρονική στιγμή, δηλαδή $s_i(t) = (SoE_i(t), t)$. Επίσης, πλέον η εξίσωση 3.2, που εκφράζει την ικανοποίηση του ισοζυγίου ισχύος, ενσωματώνεται στην ανταμοιβή κάθε μπαταρίας, η οποία διαμορφώνεται ως εξής:

$$r_i(t) = -(P(t) - P_{disp}(t))^2 - (P(t) - \sum_{i=1}^{N_B} B_i(t) - \widetilde{L}(t))^2 - \Lambda_i(t) \tag{3.19}$$

Για τον υπολογισμό της ισχύος στο PCC, το DSO λαμβάνει επιπλέον υπόψιν τις δράσεις όλων των μπαταριών. Επομένως, κάθε χρονική στιγμή t, η βέλτιστη ισχύς στο PCC, $P^*(t)$,

υπολογίζεται από το DSO ως εξής:

$$P^*(t) = \underset{P(t)}{\operatorname{argmin}}\{(P(t) - P_{disp}(t))^2 + w \cdot (P(t) - \sum_{i=1}^{N_B} B_i(t) - \widetilde{L}(t))^2\} \tag{3.20}$$

όπου w ένα θετικό βάρος.

Ο αλγόριθμος MADDPG εφαρμόζεται όπως περιγράφεται στην ενότητα 3.2.4, με εξαίρεση τις διαδικασίες που αφορούν τον πολλαπλασιαστή Lagrange. Επομένως, ο συνολικός αλγόριθμος ελέγχου διαμορφώνεται ως εξής: Κάθε χρονική στιγμή, το πρόβλημα του DSO επιλύεται σύμφωνα με την εξίσωση 3.20, υπολογίζοντας τη βέλτιστη ισχύ στο PCC. Επίσης, ο πράκτορας κάθε μπαταρίας λαμβάνει από το περιβάλλον της την τρέχουσα κατάστασή της και επιλέγει μια δράση βάσει αυτής. Στη συνέχεια, το περιβάλλον κάθε μπαταρίας υπολογίζει την ανταμοιβή της σύμφωνα με την 3.19 και μεταβαίνει στην επόμενη κατάσταση. Ακολούθως, οι καταστάσεις, οι δράσεις, οι ανταμοιβές και οι επόμενες καταστάσεις όλων των πρακτόρων συγχωνεύονται και αποθηκεύονται είτε στον Replay Buffer είτε στον Terminal Replay Buffer, ανάλογα με την χρονική στιγμή.

Στο σημείο αυτό, πραγματοποιείται η εκπαίδευση των actor και critic δικτύων των πρακτόρων και ο αλγόριθμος επαναλαμβάνεται μέχρι να επιτευχθεί σύγκλιση. Μετά το πέρας της εκπαίδευσης, το μοντέλο είναι έτοιμο για χρήση, εκτελώντας ουσιαστικά τον ίδιο αλγόριθμο με εξαίρεση τη φάση της εκπαίδευσης των δικτύων. Οι συνολικοί αλγόριθμοι για την εκπαίδευση και την αξιολόγηση του μοντέλου παρουσιάζονται αντίστοιχα με τη μορφή ψευδοκώδικα στα Α΄.4 και Α΄.5 του παραρτήματος.

Κεφάλαιο 4

Υλοποίηση

Στο κεφάλαιο αυτό παρουσιάζεται ο τρόπος υλοποίησης και η πειραματική εφαρμογή των μεθόδων που αναλύθηκαν στο κεφάλαιο 3. Τα πειράματα έγιναν για δύο περιπτώσεις δικτύων διανομής με 2 και 4 μπαταρίες, αντίστοιχα. Η απόδοση των μοντέλων αξιολογείται με κριτήρια το συνολικό κόστος, τον σεβασμό των περιορισμών για την τελική SoE των μπαταριών, τον χρόνο απόκρισης και τον χρόνο εκπαίδευσης. Επιπλέον, εφαρμόζεται η μέθοδος (MPC) και πραγματοποιείται σύγκριση των επιδόσεων των τριών προσεγγίσεων ως προς τις προηγούμενες μετρικές.

Στη πρώτη ενότητα αναφέρονται τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του κώδικα και στην δεύτερη ενότητα, περιγράφεται ο τρόπος επιλογής των παραμέτρων. Στην τρίτη ενότητα παρουσιάζονται τα πειράματα που διεξήχθησαν και τα αποτελέσματα που προέκυψαν. Τέλος, στην τέταρτη ενότητα πραγματοποιείται η σύγκριση της επίδοσης των μοντέλων μας με την επίδοση της MPC.

4.1 Υλοποίηση του κώδικα

Ο κώδικας για την υλοποίηση της μεθόδου αναπτύχθηκε στην γλώσσα προγραμματισμού Python3. Οι βασικές βιβλιοθήκες που χρησιμοποιήθηκαν είναι οι εξής:

- **NumPy** [29]: Για τη διαχείριση των πινάκων.
- **Pandas** [30]: Για την επεξεργασία των δεδομένων.
- **SciPy** [31]: Για την επίλυση του προβλήματος βελτιστοποίησης του DSO.
- **TensorFlow** [32]: Για την ανάπτυξη και την εκπαίδευση του μοντέλου ενισχυτικής μάθησης.
- **Matplotlib** [33]: Για τη δημιουργία των γραφημάτων.

Για την υλοποίηση του αλγορίθμου MADDPG χρησιμοποιήσαμε ως βάση τον κώδικα που βρίσκεται εδώ [34], κάνοντας τις κατάλληλες τροποποιήσεις ώστε να λειτουργεί στο πλαίσιο της βιβλιοθήκης TensorFlow και όχι της PyTorch. Επίσης, το περιβάλλον της μπαταρίας σχεδιάστηκε σύμφωνα με τα πρότυπα του πακέτου Gym της OpenAI [35].

4.2 Επιλογή Παραμέτρων

Χαρακτηριστικά δικτύου διανομής

Θεωρήσαμε ότι το δίκτυο διανομής έχει ένα μοναδικό *PCC* για τη σύνδεση με το δίκτυο μεταφοράς και θέσαμε την ισχύ του dispatch-plan, $P_{disp}(t)$, ίση με 0 για όλες τις χρονικές στιγμές. Με αυτόν τον τρόπο, στοχεύουμε το δίκτυο διανομής να λειτουργεί αυτόνομα και ανεξάρτητα από το transmission δίκτυο ως προς την παροχή ισχύος, ώστε να βελτιώσουμε την σταθερότητα και την αξιοπιστία του.

Διάστημα λειτουργίας

Θεωρήσαμε διάστημα λειτουργίας $T = 90$ χρονικών βημάτων, διάρκειας $\Delta t = 10sec$ το καθένα. Το μήκος του χρονικού βήματος επιλέγεται ανάλογα με τον χρόνο που απαιτείται για τον υπολογισμό της δράσης της μπαταρίας σε κάθε βήμα.

Δομή Νευρωνικών Δικτύων

Τα actor δίκτυα αποτελούνταν από:

- Το επίπεδο εισόδου με έναν νευρώνα για κάθε παράμετρο της κατάστασης.
- 2 κρυφά επίπεδα με 64 νευρώνες το καθένα και με συνάρτηση ενεργοποίησης την ReLU.
- Το επίπεδο εξόδου με 1 νευρώνα και συνάρτηση ενεργοποίησης την tanh.

Το actor δεχόταν επιπλέον ως παραμέτρους την ελάχιστη και τη μέγιστη επιτρεπόμενη τιμή της ισχύος φόρτισης/εκφόρτισης της μπαταρίας, B^{min} και B^{max}, αντίστοιχα, οι οποίες χρησιμοποιούνταν για την κλιμάκωση της εξόδου από μια τιμή στο διάστημα $[-1, 1]$ σε μια τιμή στο διάστημα $[B^{min}, B^{max}]$. Αυτό έγινε προκειμένου να διασφαλιστεί ότι η τελική έξοδος του δικτύου θα βρίσκεται πάντα εντός των επιτρεπόμενων ορίων ισχύος της μπαταρίας.

Τα critic δίκτυα αποτελούνταν από:

- Το επίπεδο εισόδου με έναν νευρώνα για κάθε παράμετρο της κατάστασης όλων των μπαταριών και έναν για την δράση κάθε μπαταρίας.
- 3 κρυφά επίπεδα με 64 νευρώνες το καθένα και με συνάρτηση ενεργοποίησης την ReLU.
- Το επίπεδο εξόδου με 1 νευρώνα και γραμμική συνάρτηση ενεργοποίησης.

Για όλα τα δίκτυα χρησιμοποιήθηκε η συνάρτηση βελτιστοποίησης Adam (βλπ ενότητα 2.3.4). Επίσης, ο ρυθμός μάθησης αρχικοποιήθηκε στην τιμή 0.00001 για τα actor δίκτυα και στην τιμή 0.0001 για τα critic δίκτυα.

Πολλαπλασιαστής Lagrange

Για την αρχικοποίηση του πολλαπλασιαστή Lagrange λάβαμε υπόψιν την πειραματική μελέτη που έγινε στο έργο [6] και ουσιαστικά επιλέξαμε την τιμή στην οποία, σύμφωνα με την εν λόγω μελέτη, συνέκλινε ο πολλαπλασιαστής Lagrange, δηλαδή −0.212. Επίσης, το πλήθος των επαναλήψεων ενημέρωσης ήταν $H_{max} = 100$.

Εκτίμηση του φορτίου

Για την επίλυση του προβλήματος του DSO σύμφωνα με την εξίσωση 3.13 είναι απαραίτητο να κάνουμε μια εκτίμηση $\widetilde{L}(t)$ του συνολικού φορτίου τη χρονική στιγμή t. Το φορτίο ενός ηλεκτρικού δικτύου είναι δύσκολο να προβλεφθεί με ακρίβεια, καθώς επηρεάζεται από διάφορους παράγοντες που διέπονται από τυχαιότητα, όπως οι ανάγκες των καταναλωτών, οι καιρικές συνθήκες κ.λπ. Ωστόσο, μπορούμε να υποθέσουμε ότι το φορτίο δεν μεταβάλλεται σημαντικά μεταξύ δύο διαδοχικών χρονικών στιγμών και ως εκ τούτου, να θέσουμε ως εκτίμησή του την τιμή του πραγματικού φορτίου που παρατηρήθηκε την αμέσως προηγούμενη χρονική στιγμή. Επομένως, θεωρήσαμε $\widetilde{L}(t) = L(t-1)$.

Παράμετρος γ

Ο παράγοντας μείωσης γ επηρεάζει σημαντικά την εκπαίδευση του μοντέλου. Όπως περιγράφεται στην ενότητα 2.4.1, ο παράγοντας μείωσης καθορίζει τον βαθμό στον οποίο επηρεάζουν τις αποφάσεις του πράκτορα οι μελλοντικές ανταμοιβές έναντι των πιο πρόσφατων. Επομένως, στην περίπτωσή μας, αν η τιμή του γ είναι υπερβολικά χαμηλή, οι τελικές τιμές $SoE(T)$ μπορεί να βρεθούν εκτός των επιτρεπόμενων ορίων. Αντιθέτως, αν η τιμή του γ είναι υπερβολικά υψηλή, ενδέχεται οι μπαταρίες να μην εκφορτίζουν για να αποφευχθεί ο κίνδυνος η $SoE(T)$ να βγει εκτός ορίων. Τελικά, επιλέξαμε $\gamma = 0.4$.

Συνάρτηση ποινής

Η ανταμοιβή που επιστρέφεται στον πράκτορα από το περιβάλλον κάθε χρονική στιγμή αποτελεί επίσης καθοριστικό παράγοντα για την ανάπτυξη της πολιτικής του, αλλά και για την εξασφάλιση της ορθής λειτουργίας της μπαταρίας. Προκειμένου να ενθαρρύνουμε τους πράκτορες να αναπτύσσουν πολιτικές που σέβονται τους περιορισμούς της 3.11, επιβάλλαμε μια ποινή για την παραβίαση καθενός εξ αυτών, προσθέτοντας έναν επιπλέον όρο-ποινή (penalty) στην συνάρτηση ανταμοιβής, ο οποίος μειώνει την τιμή της.

Επιπλέον, προκειμένου να διασφαλίσουμε την ικανοποίηση των τελικών περιορισμών, κάθε χρονική στιγμή υπολογίσουμε την αναμενόμενη τελική SoE της μπαταρίας, με δεδομένο τον χρόνο που απομένει μέχρι τη λήξη της περιόδου λειτουργίας και τη μέγιστη ισχύ φόρτισης/εκφόρτισης της και επιβάλλαμε μια ποινή σε περίπτωση που η τιμή αυτή βρίσκεται εκτός των τελικών ορίων. Στην ουσία, ελέγχαμε αν ο διαθέσιμος χρόνος επαρκεί για να καλύψει τον χρόνο που χρειάζεται η μπαταρία για να φορτίσει/εκφορτίσει μέχρι η SoE της να ικανοποιεί τους τελικούς περιορισμούς. Επομένως, εφαρμόσαμε την ακόλουθη συνάρτηση

ποινής:

$$\Lambda_i(t) = \begin{cases} -1, & \text{Αν } SoE_i(t) < SoE_i^{min} \text{ ή } SoE_i(t) > SoE_i^{max}(t) \\ -1, & \text{Αν } SoE_i(t) < SoE_i^{min,T} \text{ ή } SoE_i(t) > SoE_i^{max,T} \text{ και } t = T \\ -1, & \text{Αν } (SoE_i^{min,T} - SoE_i(t+1)) > B_i^{max} \cdot \Delta T \\ -1, & \text{Αν } (SoE_i(t+1) - SoE_i^{max,T}) > |B_i^{min}| \cdot \Delta T \\ 0, & \text{Αλλιώς} \end{cases} \tag{4.1}$$

όπου $\Delta T = |T - t - 5| \cdot \Delta t$.

Αξιολόγηση επίδοσης

Τέλος, για την αξιολόγηση της επίδοσης των μοντέλων με γνώμονα το κατά πόσο η λύση που βρέθηκε ικανοποιεί βέλτιστα το ισοζύγιο ισχύος, υπολογίσαμε το ακόλουθο συνολικό κόστος, λαμβάνοντας υπόψιν την πραγματική τιμή του φορτίου:

$$C_{total} = \sum_{t=0}^{T-1} \Big(P(t) - P_{disp}(t)\Big)^2 = \sum_{t=0}^{T-1} \Big(\sum_{i=1}^{N_B} B_i(t) + L(t) - P_{disp}(t)\Big)^2 \tag{4.2}$$

4.3 Πειράματα

Στην ενότητα αυτή παρουσιάζονται τα πειράματα που εκτελέστηκαν για την αξιολόγηση των μεθόδων. Συγκεκριμένα προσομοιώθηκαν δύο δίκτυα διανομής, με 2 και με 4 μπαταρίες αντίστοιχα. Στο σχήμα 4.1 παρουσιάζεται το φορτίο που χρησιμοποιήθηκε για τις προσομοιώσεις. Πρόκειται για ένα ημιτονοειδές σήμα της πρώτης μισής περιόδου και πλάτους 0.3 ανά μονάδα που ορίζεται για 90 διακριτές χρονικές στιγμές. Στον πίνακα Β΄.1 του παραρτήματος συνοψίζονται οι τιμές των υπερπαραμέτρων που χρησιμοποιήθηκαν για την εκπαίδευση.

Δεδομένου ότι η εκπαίδευση ενός μοντέλου στο πλαίσιο του αλγορίθμου MADDPG δε δίνει πάντα το ίδιο αποτέλεσμα, χρειάστηκε να εκτελέσουμε αρκετές εκπαιδεύσεις και τελικά να επιλέξουμε τα μοντέλα που παρουσίασαν την καλύτερη επίδοση κατά την αξιολόγηση.

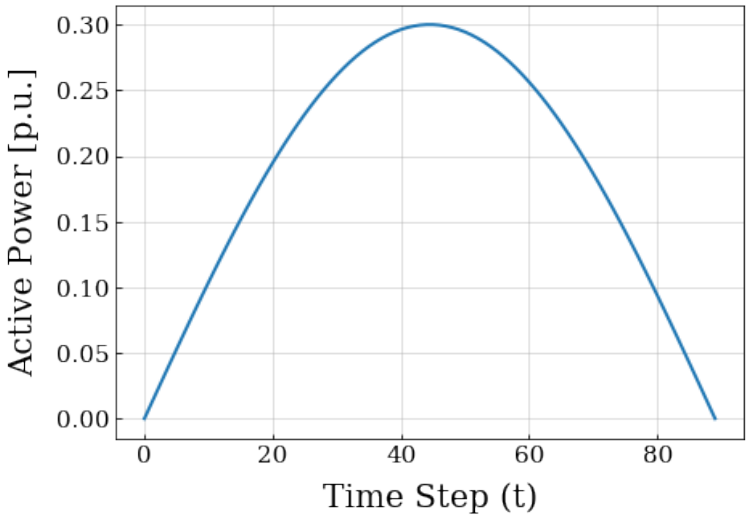
4.3.1 Αποτελέσματα μεθόδου με Lagrangian Decomposition

Προσομοίωση με 2 μπαταρίες

Στον πίνακα 4.1 παρουσιάζονται οι τεχνικές προδιαγραφές των 2 μπαταριών. Η εκπαίδευση χρειάστηκε περίπου 48 ώρες για να ολοκληρωθεί.

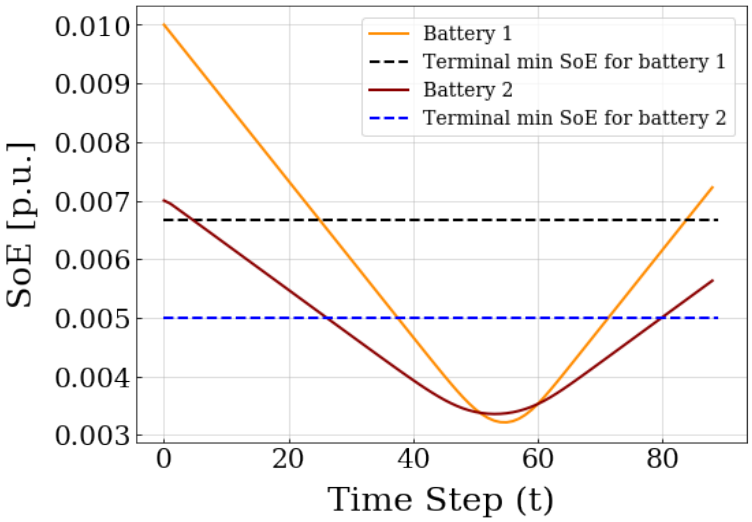
Πίνακας 4.1: *Παράμετροι μπαταριών σε ανά μονάδα τιμές (p.u.)*

ID	**SoE**			**B**		**SoE(T)**	
	min	max	init	min	max	min	max
1	0	0.0334	0.01	-0.048	0.048	0.0066	0.0334
2	0	0.0334	0.007	-0.028	0.028	0.005	0.0334



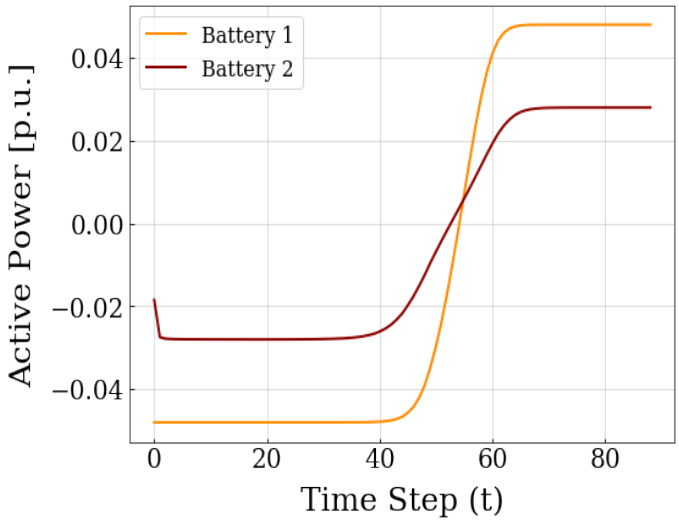
Σχήμα 4.1: *Φορτίο*

Στα σχήματα 4.2-4.4 φαίνονται τα αποτελέσματα που προέκυψαν. Στο σχήμα 4.2 απεικονίζεται η εξέλιξη της κατάστασης της ενέργειας των μπαταριών SoE και στο σχήμα 4.3 η ισχύς φόρτισης/εκφόρτισης τους κάθε χρονική στιγμή. Στο σχήμα 4.4 παρουσιάζεται το φορτίο, η συνολική ενεργός ισχύς του φορτίου και οι αποκλίσεις στο ισοζύγιο ισχύος κάθε χρονική στιγμή.



Σχήμα 4.2: *Προσομοίωση δικτύου διανομής με 2 μπαταρίες: State of Energy Μπαταριών*

Η πρώτη σημαντική παρατήρηση είναι ότι ικανοποιούνται οι περιορισμοί για την τελική τιμή της SoE και για τις δύο μπαταρίες. Παρατηρούμε ότι οι μπαταρίες εκφορτίζουν περίπου μέχρι τη χρονική στιγμή $t = 55$ προκειμένου να ικανοποιηθεί το ενεργειακό ισοζύγιο ισχύος, δηλαδή η συνολική ισχύς να έρθει όσο τον δυνατόν πιο κοντά στην τιμή της ισχύος αναφοράς $P_{disp} = 0$. Στη συνέχεια, αρχίζουν να φορτίζουν ώστε η τελική SoE τους να καταφέρει να υπερβεί την ελάχιστη επιτρεπόμενη τιμή. Ο χρόνος απόκρισης του συστήματος ήταν 1.507 sec και το συνολικό κόστος που προέκυψε σύμφωνα με τη σχέση 4.2 είναι $C_{tot} = 3.321$.



Σχήμα 4.3: *Προσομοίωση δικτύου διανομής με 2 μπαταρίες: Ισχύες Μπαταριών*

Προσομοίωση με 4 μπαταρίες

Στον πίνακα 4.2 παρουσιάζονται οι τεχνικές προδιαγραφές των 4 μπαταριών. Η εκπαίδευση χρειάστηκε περίπου 92 ώρες για να ολοκληρωθεί, δηλαδή σχεδόν διπλάσιο χρόνο σε σχέση με την περίπτωση των 2 μπαταριών.

Πίνακας 4.2: *Παράμετροι μπαταριών σε ανά μονάδα τιμές (p.u.)*

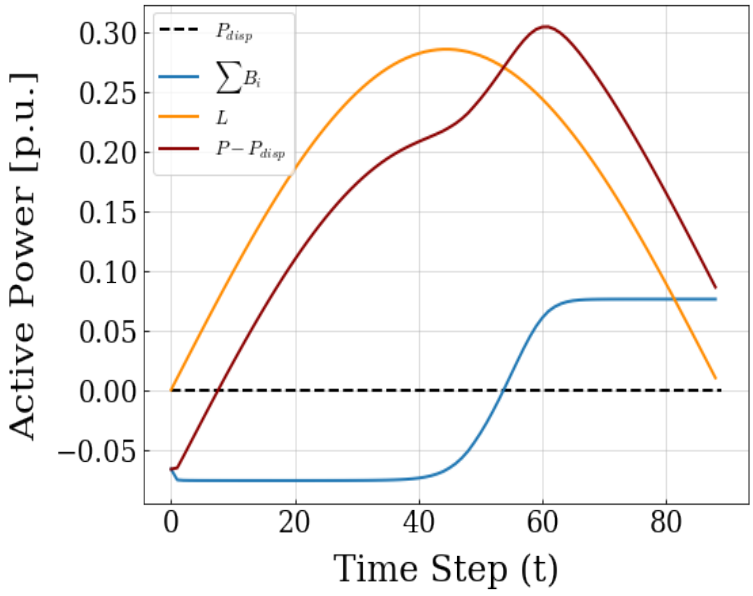
ID	**SoE**			**B**		**SoE(T)**	
	min	max	init	min	max	min	max
1	0	0.0334	0.01	-0.048	0.048	0.0066	0.0334
2	0	0.0334	0.007	-0.028	0.028	0.005	0.0334
3	0	0.0334	0.003	-0.015	0.015	0.002	0.0334
4	0	0.0334	0.02	-0.048	0.048	0.0133	0.0334

Στα σχήματα 4.5, 4.6 και 4.7 φαίνονται τα αποτελέσματα που προέκυψαν. Όπως και στην περίπτωση των 2 μπαταριών, παρατηρούμε ότι όλες οι μπαταρίες εκφορτίζουν μέχρι κάποια χρονική στιγμή, διαφορετική για την καθεμία, προκειμένου να ικανοποιηθούν οι ενεργειακές απαιτήσεις και στη συνέχεια φορτίζουν ώστε η τελική SoE τους να φτάσει στην ελάχιστη επιτρεπόμενη τιμή. Ο χρόνος απόκρισης του συστήματος ήταν 3.135 sec, μεγαλύτερος αυτού του συστήματος των 2 μπαταριών ενώ το συνολικό κόστος ελαττώθηκε στο $C_{tot} = 3.224$.

Παρατηρούμε λοιπόν, ότι με την αύξηση του αριθμού των μπαταριών, βελτιώνεται η επίδοση του μοντέλου, αλλά αυξάνεται σημαντικά η υπολογιστική του πολυπλοκότητα, τόσο ως προς τον χρόνο εκπαίδευσης όσο και ως προς τις απαιτήσεις σε μνήμη.

4.3.2 Αποτελέσματα μεθόδου χωρίς Lagrangian Decomposition

Εφαρμόσαμε τον αλγόριθμο MADDPG χωρίς τη πρότερη χρήση της Lagrangian Decomposition στα δύο προηγούμενα δίκτυα διανομής με τις 2 και τις 4 μπαταρίες, για το ίδιο φορτίο. Χρησιμοποιήσαμε τις τιμές των υπερπαραμέτρων που αναγράφονται στον πίνακα



Σχήμα 4.4: *Προσομοίωση δικτύου διανομής με 2 μπαταρίες: Φορτίο και Συνολική ενεργός ισχύς του δικτύου*

Β΄.1 του παραρτήματος και θέσαμε $w = 1$. Ο χρόνος που χρειάστηκε για την εκπαίδευση των δικτύων ήταν περίπου 4 και 8 ώρες, αντίστοιχα, λιγότερος κατά πολύ σε σχέση με την προηγούμενη μέθοδο.

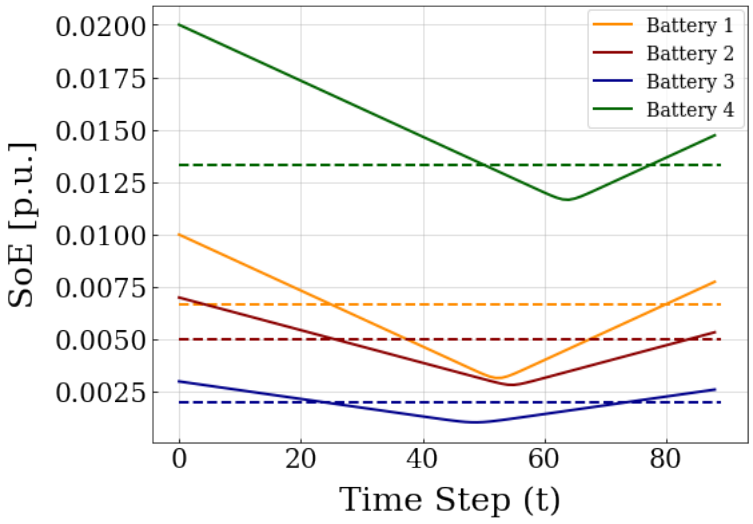
Στα σχήματα 4.8, 4.9 και 4.10 παρουσιάζονται τα αποτελέσματα που προέκυψαν για το δίκτυο με τις 2 μπαταρίες. Παρατηρούμε ότι οι μπαταρίες παρουσιάζουν παρόμοια συμπεριφορά με αυτή της μεθόδου που περιλαμβάνει τη Lagrangian Decomposition, αλλά με μικρότερο συνολικό κόστος, $C_{tot} = 3.167$. Επιπλέον, μειώθηκε και ο χρόνος απόκρισης του συστήματος στα 0.332 sec.

Στα σχήματα 4.11, 4.12 και 4.13 παρουσιάζονται τα αποτελέσματα που προέκυψαν για το δίκτυο με τις 4 μπαταρίες. Παρόμοια με την περίπτωση των 2 μπαταριών, παρατηρούμε ότι οι μπαταρίες παρουσιάζουν την επιθυμητή συμπεριφορά, με μικρότερο συνολικό κόστος σε σχέση με αυτό της μεθόδου που περιλαμβάνει τη Lagrangian Decomposition, $C_{tot} = 2.892$, καθώς και με μικρότερο χρόνο απόκρισης, 0.679 sec. Επίσης, το συνολικό κόστος του συστήματος των 4 μπαταριών είναι μικρότερο από αυτό του συστήματος των 2 μπαταριών, ενώ ο χρόνος απόκρισης διπλασιάστηκε.

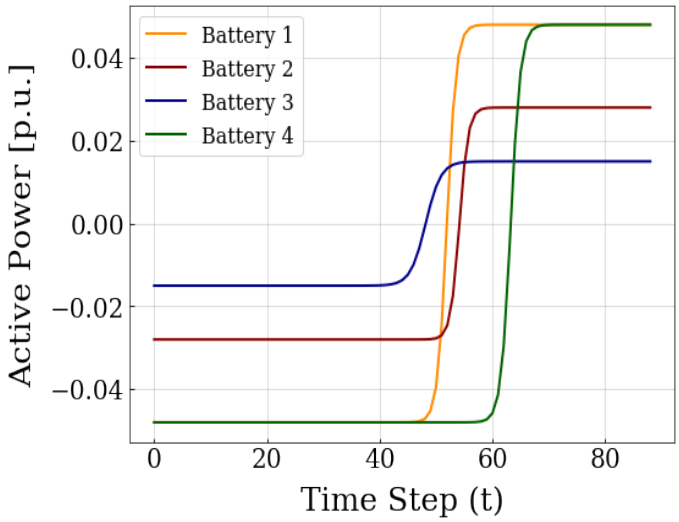
Συμπεραίνουμε λοιπόν, ότι όπως και με την προηγούμενη μέθοδο, η αύξηση του αριθμού των μπαταριών βελτιώνει την απόδοση του συστήματος, αλλά αυξάνει σημαντικά τον απαιτούμενο χρόνο εκπαίδευσης. Επιπλέον όμως, έναντι του προηγούμενου μοντέλου, το μοντέλο αυτό παρουσιάζει μικρότερο κόστος, μικρότερο χρόνο απόκρισης και παράλληλα χρειάζεται πολύ λιγότερο χρόνο για την εκπαίδευσή του.

4.4 Αποτελέσματα της MPC

Στην ενότητα αυτή, συγκρίνουμε τα αποτελέσματα των δύο συστημάτων που αναπτύξαμε με αυτά που προέκυψαν από προσομοιώσεις βασισμένες στη μέθοδο MPC. Συγκεκριμένα,



Σχήμα 4.5: *Προσομοίωση δικτύου διανομής με 4 μπαταρίες: State of Energy Μπαταριών*

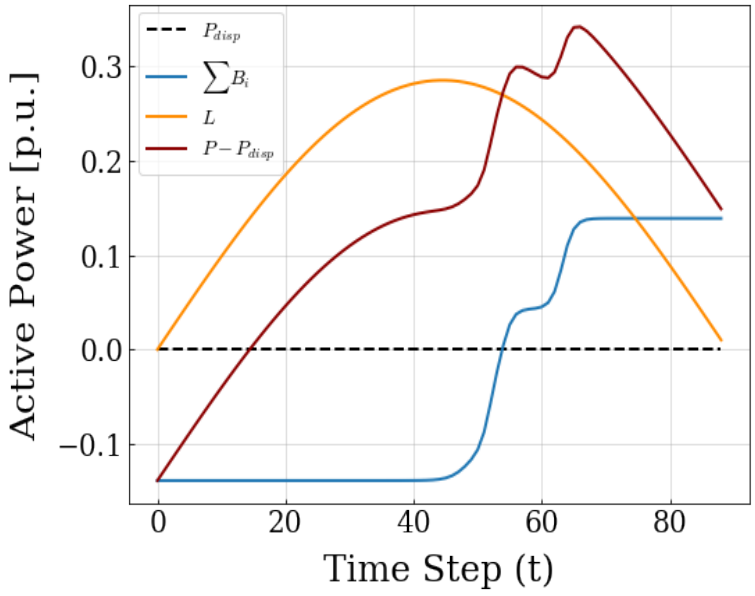


Σχήμα 4.6: *Προσομοίωση δικτύου διανομής με 4 μπαταρίες: Ισχύες Μπαταριών*

έχοντας εφαρμόσει την MPC, για να επιλύσουμε το αρχικό πρόβλημα (3.10-3.7) στα δύο προηγούμενα δίκτυα διανομής, συγκρίνουμε την απόδοση των δικών μας προσεγγίσεων με αυτήν, ως προς το κόστος και τον χρόνο απόκρισης. Στον πίνακα 4.3 παρουσιάζεται το συνολικό κόστος κάθε μεθόδου και στον πίνακα 4.4 ο χρόνος απόκρισης.

Παρατηρούμε κατ' αρχάς, ότι το κόστος της MPC είναι μικρότερο από αυτό των δικών μας μοντέλων, χωρίς ωστόσο να υπάρχει μεγάλη διαφορά. Επίσης, είναι σημαντικό να σημειωθεί ότι για τη σύγκριση του κόστους θα πρέπει να λάβουμε υπόψιν ότι η MPC βασίστηκε σε μια ακριβή πρόβλεψη του φορτίου, ενώ τα δικά μας μοντέλα αξιολογήθηκαν με ένα ελαφρώς διαφορετικό φορτίο από αυτό με το οποίο εκπαιδεύτηκαν. Ως εκ τούτου, μπορούμε να αξιολογήσουμε την επίδοση των μοντέλων μας ως αρκετά ικανοποιητική.

Όσον αφορά τον χρόνο απόκρισης, η MPC χρειάστηκε περίπου 4 λεπτά στην περίπτωση των 2 μπαταριών και περίπου 12 λεπτά, στην περίπτωση 4 μπαταριών, για να ολοκληρώσει τους υπολογισμούς, ενώ τα δύο μοντέλα που σχεδιάσαμε αποκρίνονται άμεσα. Επιπλέον,

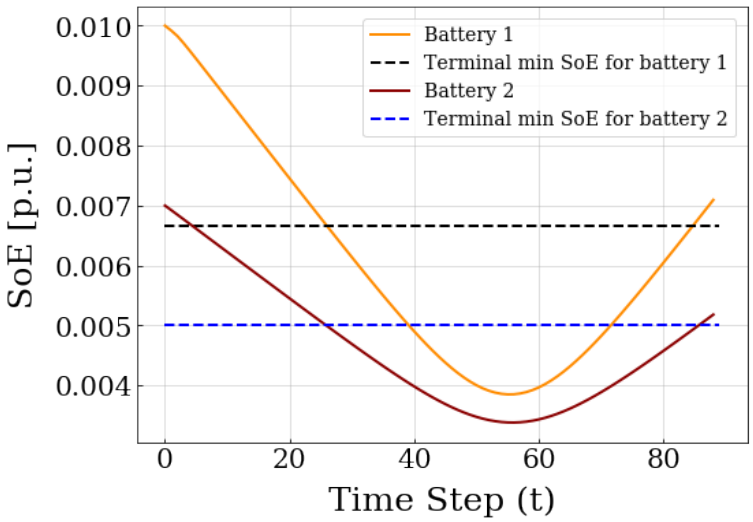


Σχήμα 4.7: *Προσομοίωση δικτύου διανομής με 4 μπαταρίες: Φορτίο και Συνολική ενεργός ισχύς του δικτύου*

Πίνακας 4.3: *Σύγκριση του συνολικού κόστους*

#Batteries	**MPC**	**MADDPG with LD**	**MADDPG**
2	2.413	3.321	3.167
4	1.531	3.224	2.892

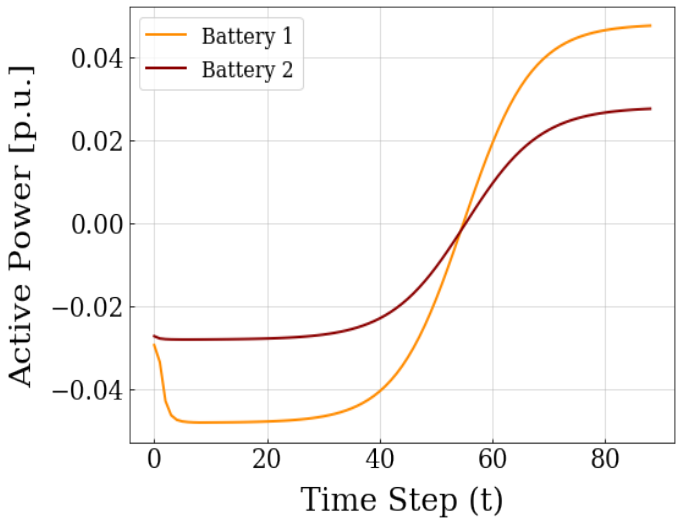
φαίνεται πως ο χρόνος απόκρισης της MPC παρουσιάζει σημαντική κλιμάκωση με την αύξηση των μπαταριών, ενώ στην περίπτωση των μοντέλων μας η κλιμάκωση είναι αμελητέα.



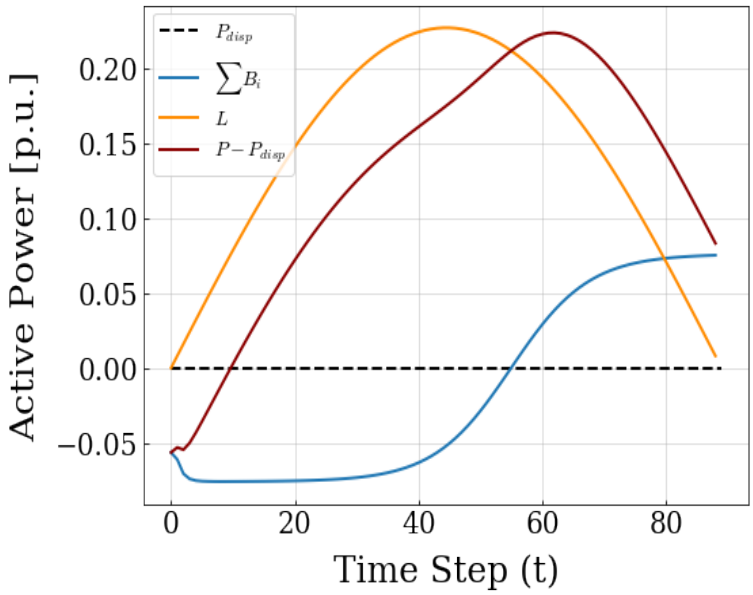
Σχήμα 4.8: *Προσομοίωση δικτύου διανομής με 2 μπαταρίες, χωρίς Lagrangian Decomposition: State of Energy Μπαταριών*

Πίνακας 4.4: *Σύγκριση του χρόνου απόκρισης (sec)*

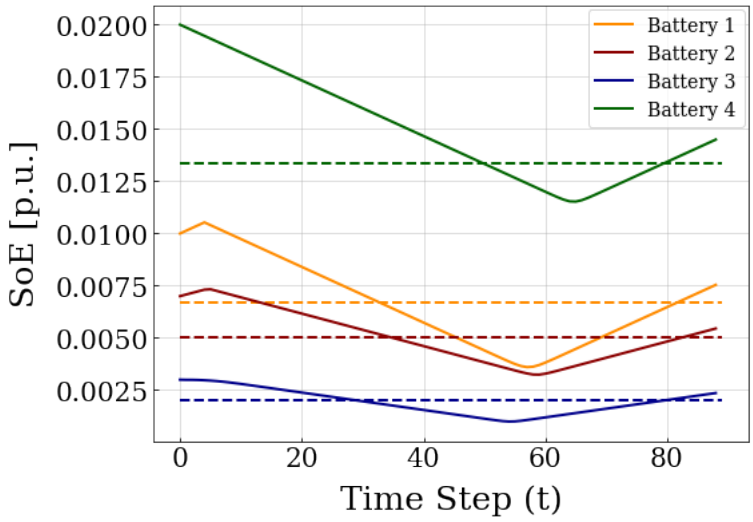
#Batteries	MPC	MADDPG with LD	MADDPG
2	253.784	1.507	0.332
4	702.890	3.135	0.679



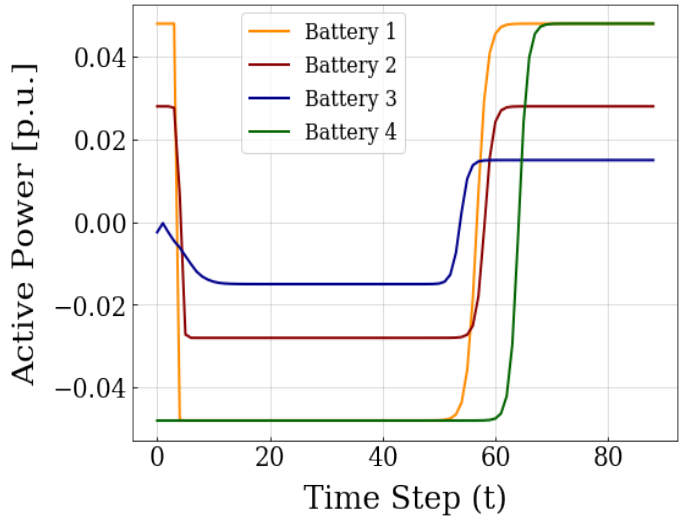
Σχήμα 4.9: *Προσομοίωση δικτύου διανομής με 2 μπαταρίες, χωρίς Lagrangian Decomposition: Ισχύες Μπαταριών*



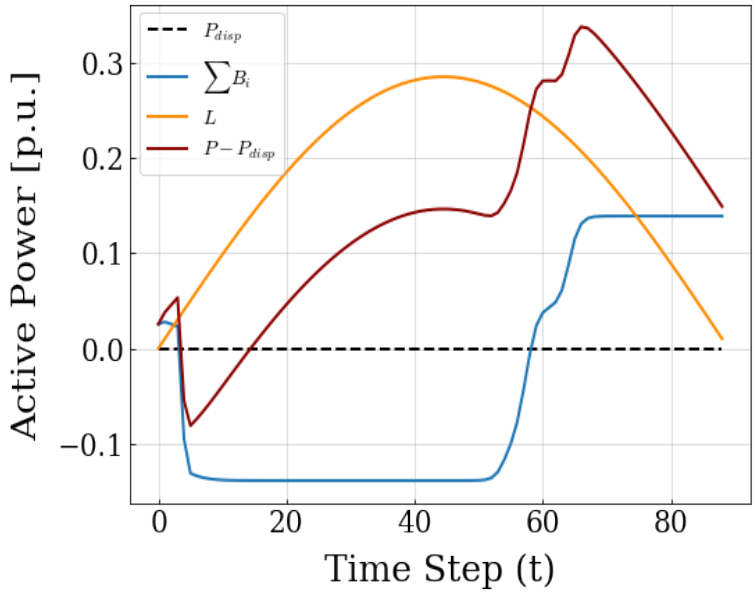
Σχήμα 4.10: *Προσομοίωση δικτύου διανομής με 2 μπαταρίες, χωρίς Lagrangian Decomposition: Φορτίο και Συνολική ενεργός ισχύς του δικτύου*



Σχήμα 4.11: *Προσομοίωση δικτύου διανομής με 4 μπαταρίες, χωρίς Lagrangian Decomposition: State of Energy Μπαταριών*



Σχήμα 4.12: *Προσομοίωση δικτύου διανομής με 4 μπαταρίες, χωρίς Lagrangian Decomposition: Ισχύες Μπαταριών*



Σχήμα 4.13: *Προσομοίωση δικτύου διανομής με 4 μπαταρίες, χωρίς Lagrangian Decomposition: Φορτίο και Συνολική ενεργός ισχύς του δικτύου*

Κεφάλαιο 5

Επίλογος

5.1 Συμπεράσματα

Σε αυτή τη διπλωματική αναπτύχθηκαν δύο συστήματα για τον κατανεμημένο έλεγχο των μπαταριών ενός smart grid, με χρήση ενισχυτικής μάθησης πολλαπλών πρακτόρων. Συγκεκριμένα, εφαρμόστηκε ο αλγόριθμος MADDPG, ο οποίος είναι ένας online, off-policy και model-free αλγόριθμος που συνδυάζει την centralized εκπαίδευση, κατά την οποία κάθε πράκτορας μαθαίνει αξιοποιώντας πληροφορίες που προέρχονται από όλους τους πράκτορες, με την decentralized εκτέλεση, κατά την οποία κάθε πράκτορας εφαρμόζει την εξατομικευμένη πολιτική του με βάση τις δικές του μόνο παρατηρήσεις.

Στο πρώτο σύστημα, έγινε χρήση της μεθόδου Lagrangian Decomposition και μέσω των πολλαπλασιαστών Lagrange ανταλλασσόταν ένα σήμα μεταξύ μπαταριών και του DSO, το οποίο ενημέρωνε τους πράκτορες σχετικά με την ικανοποίηση του ισοζυγίου ισχύος. Στο δεύτερο σύστημα δε χρησιμοποιήθηκε η Lagrangian Decomposition, αλλά εφαρμόστηκε απλώς ο αλγόριθμος MADDPG, με δεδομένο ότι οι μπαταρίες έχουν πλήρη εποπτεία του συστήματος κατά την εκπαίδευσή τους.

Από τα πειράματα που εκτελέστηκαν προέκυψε κατ' αρχάς, ότι συγκριτικά με τη συμβατική μέθοδο MPC, τα συστήματα που αναπτύχθηκαν μολονότι εμφανίζουν μεγαλύτερο κόστος, παρουσιάζουν ικανοποιητικές επιδόσεις και επιπλέον, υπερτερούν σημαντικά όσον αφορά τον χρόνο απόκρισης. Ωστόσο, ένα μειονέκτημα των δικών μας μοντέλων είναι ότι ο χρόνος που απαιτείται για την εκπαίδευση τους κλιμακώνεται έντονα με την αύξηση του πλήθους των μπαταριών στο δίκτυο.

Μεταξύ των δύο συστημάτων που αναπτύξαμε, αυτό που δεν εφαρμόζει την Lagrangian Decomposition έχει καλύτερη απόδοση και μικρότερο χρόνο απόκρισης. Επίσης, έχει μικρότερη πολυπλοκότητα, είναι απλούστερο στην υλοποίηση και χρειάζεται πολύ λιγότερο χρόνο για την εκπαίδευσή του.

5.2 Μελλοντικές Επεκτάσεις

Η ανάπτυξη αυτού του συστήματος ελέγχου θα μπορούσε να επεκταθεί με τη χρήση συνδυαστικών περιορισμών που αφορούν το άθροισμα των τελικών SoE των μπαταριών, ώστε να αξιοποιείται αποδοτικότερα η ενέργειά τους. Για παράδειγμα, το πρόβλημα μπορεί να

μοντελοποιηθεί ως εξής:

$$\min_{P(t),B_i(t),\ldots,B_{N_B}(t)} \sum_{t=1}^{T} (P(t) - P_{disp}(t))^2 \tag{5.1}$$

$$\text{τ.ω. } P(t) - \sum_{i=1}^{N_B} B_i(t) - L(t) = 0 \tag{5.2}$$

$$SoE_i(t+1) = SoE_i(t) + B_i(t)\Delta t \tag{5.3}$$

$$B_i^{min} \leq B_i(t) \leq B_i^{max} \tag{5.4}$$

$$SoE_i^{min} \leq SoE_i(t) \leq SoE_i^{max} \tag{5.5}$$

$$SoE_i^{min,T} \leq SoE_i(T) \leq SoE_i^{max,T} \tag{5.6}$$

$$SoE_j^{min,T} \leq \sum_i a_{ji} SoE_i(T) \leq SoE_j^{max,T} \tag{5.7}$$

$$\text{για } i \in \{1,\ldots,N_B\}, t \in \{1,\ldots,T-1\}, j \in \{1,\ldots,G\} \tag{5.8}$$

Όπου G το πλήθος των περιορισμών που συνδυάζουν τις SoE διαφορετικών μπαταριών. Στο σύστημα αυτό, για τις μπαταρίες που δε συμμετέχουν σε περιορισμούς αθροίσματος, ισχύουν οι περιορισμοί για τις δικές τους τελικές SoE.

Με αυτόν τον τρόπο, θα μπορούσε να επιτευχθεί μια πιο ισορροπημένη διαχείριση της ενέργειας στο σύστημα των μπαταριών, αξιοποιώντας πλήρως τις δυνατότητες κάθε μπαταρίας. Για παράδειγμα, ορισμένες μπαταρίες θα είχαν τη δυνατότητα να διατηρήσουν υψηλότερα επίπεδα ενέργειας, ενώ άλλες θα συμμετείχαν πιο ενεργά στην κάλυψη του ισοζυγίου ισχύος. Επομένως, με τη χρήση συνδυαστικών περιορισμών θα ήταν δυνατόν να αναπτυχθεί ένα πιο αποδοτικό και ευέλικτο σύστημα ελέγχου των μπαταριών, που θα αξιοποιούσε αποτελεσματικότερα τη διαθέσιμη ενέργεια, θα προσαρμοζόταν ευκολότερα σε διαφορετικές συνθήκες και θα κάλυπτε αποδοτικότερα τις μελλοντικές ανάγκες του δικτύου.

Παραρτήματα

Κεφάλαιο Α΄

Αλγόριθμοι

Αλγοριθμος Α'.1: *Αλγόριθμος MADDPG*

for *each agent i* **do**
  Initialize critic $Q_i(\mathbf{S}, \mathbf{A}; \phi_i)$ and actor $\mu_i(s_i; \vartheta_i)$ network with weights ϕ_i and ϑ_i
  Initialize target critic Q_i^- and target actor μ_i^- network with $\phi_i^- \leftarrow \phi_i$, and $\vartheta_i^- \leftarrow \vartheta_i$
end
Initialize Replay Buffer $\mathcal{D}$ as empty
for *each iteration until convergence* **do**
  for *each agent i* **do**
    Observe state s_i
    Select action $a_i = clip(\mu_i(s_i; \vartheta_i), a_i^{min}, a_i^{max})$
    Execute a_i in the environment
    Observe next state s_i', reward r_i and done signal d_i to indicate whether s_i' is terminal
  end
  Store $(\mathbf{S}, \mathbf{A}, \mathbf{R}, \mathbf{S}', \mathbf{d})$ in Replay Buffer $\mathcal{D}$, where $\mathbf{S} = (s_1, \ldots, s_N)$, $\mathbf{A} = (a_1, \ldots, a_N)$, $\mathbf{R} = (r_1, \ldots, r_N)$, $\mathbf{S}' = (s_1', \ldots, s_N')$ and $\mathbf{d} = (d_1, \ldots, d_N)$
  if S' is terminal **then** reset environment state
  if *it's time to update* **then**
    for *each update* **do**
      Randomly sample a batch of M transitions $(\mathbf{S}, \mathbf{A}, \mathbf{R}, \mathbf{S}', \mathbf{d})$ from $\mathcal{D}$
      for *each transition j in the minibatch* **do**
        Compute the target

$$y_i^j = r_i^j + \gamma(1 - d_i^j)Q_i^-(\mathbf{S}'^j, a_1'^j, ..., a_N'^j|_{a_i'^j = \mu_i^-(s_i'^j|\vartheta_i^-)}; \phi_i^-)$$

      end
      Update the agent's critic (Q-function) by minimizing the following loss, using one step gradient descent:

$$L(\phi_i) = \frac{1}{M}\sum_{j=1}^{M}\left(y_i^j - Q_i(\mathbf{S}^j, a_1^j, \ldots, a_N^j; \phi_i)\right)^2$$

      Update the agent's actor (policy) by one step of gradient ascent using:

$$\begin{aligned}\nabla_{\vartheta_i} J(\vartheta_i) &= \nabla_{\vartheta_i}\mathbb{E}_{\mathbf{S},\mathbf{A}\sim\mathcal{D}}\left[Q_i(\mathbf{S}^j, a_1^j, \ldots, a_i, \ldots, a_N^j|_{a_i=\mu_i(s_i^j;\vartheta_i)}; \phi_i)\right] \\ &= \frac{1}{M}\sum_{j=1}^{M}\nabla_{\vartheta_i}\mu_i(s_i;\vartheta_i)\nabla_{a_i}Q_i(\mathbf{S}^j, a_1^j, \ldots, a_i, \ldots, a_N^j|_{a_i=\mu_i(s_i^j;\vartheta_i)}; \phi_i)\end{aligned}$$

      Update the weights of its target critic and target actor networks:

$$\phi_i^- \leftarrow \tau\phi_i^- + (1+\tau)\phi_i$$
$$\vartheta_i^- \leftarrow \tau\vartheta_i^- + (1+\tau)\vartheta_i$$

    end
  end
end

Αλγοριθμος Α΄.2: *Εκπαίδευση με Lagrangian Decomposition*

for *each agent* i **do**
  Initialize critic $Q_i(\mathbf{S}, \mathbf{B}; \phi_i)$ and actor $\mu_i(s_i; \vartheta_i)$ network with weights ϕ_i and ϑ_i
  Initialize target critic Q_i^- and target actor μ_i^- network with $\phi_i^- \leftarrow \phi_i$, and $\vartheta_i^- \leftarrow \vartheta_i$
end
Initialize the Replay Buffer $\mathcal{D}$ and the Terminal Replay Buffer $\mathcal{D}_T$ as empty
Define P_{disp} and L^{real} for all timesteps
best_score $\leftarrow -\infty$
for *each iteration until convergence* **do**
  score $\leftarrow 0$
  Rendomly decide whether to initialize $\lambda^{(0)}(0)$ or not
  for each battery i **do** receive initial state observation $s_i(0) = (SoE_i(0), \lambda(0), 0)$ **end**
  for $t = 0 : 1 : T - 1$ **do**
    if $t = 0$ **then**
      $\widetilde{L}(t) = L^{real}(t)$
    else
      $\widetilde{L}(t) = L^{real}(t-1)$
    end
    Initialize $a^{(0)}, \eta \leftarrow 0$
    while $\eta \leq H_{max} \wedge |\lambda^{(\eta)} - \lambda^{(\eta-1)}| \geq \epsilon$ **do**
      Compute $P^{(\eta)}(t)$ using 3.13
      for *each battery* i **do**
        Select action $B_i^{(\eta)}(t) = B_i^{(\eta)} = clip(\mu_i(s_i^{(\eta)}(t); \vartheta_i), B_i^{min}, B_i^{max})$,
        where $s_i^{(\eta)}(t) = (SoE_i(t), \lambda^{(\eta)}(t), t)$
      end
      Compute $\lambda^{(\eta+1)}$ using 3.12 and update the state of each battery with it
      Compute $a^{(\eta+1)}$
      $\eta \leftarrow \eta + 1$
    end
    for *each battery* i **do**
      Execute $B_i^{(\eta)}$ in the environment
      Observe next state $s_i^{(\eta)}(t+1) = s_i'^{(\eta)} = (SoE_i^{(\eta)}(t+1), \lambda^{(\eta)}(t), t+1)$,
      where $SoE_i^{(\eta)}(t+1) = SoE_i(t) + B_i^{(\eta)}(t)\Delta t$
      Observe reward $r_i^{(\eta)}(t) = r_i^{(\eta)} = \lambda^{(\eta)}(t) B_i^{(\eta)}(t) - \Lambda_i(t)$ and done signal $d_i(t)$
    end
    if $t = T - 1$ **then**
      Store transition $(\mathbf{S}, \mathbf{B}, \mathbf{S}', \mathbf{R}, \mathbf{b})$ in Terminal Replay Buffer $\mathcal{D}_T$
    else
      Store transition $(\mathbf{S}, \mathbf{B}, \mathbf{S}', \mathbf{R}, \mathbf{b})$ in Replay Buffer $\mathcal{D}$
    end
    where $\mathbf{S} = \{s_1^{(\eta)}, \ldots, s_{N_B}^{(\eta)}\}, \mathbf{B} = \{B_1^{(\eta)}, \ldots, B_{N_B}^{(\eta)}\}, \mathbf{S}' = \{s_1'^{(\eta)}, \ldots, s_{N_B}'^{(\eta)}\}$,
    $\mathbf{R} = \{r_1^{(\eta)}, \ldots, r_{N_B}^{(\eta)}\}$ and $\mathbf{d} = \{d_1(t), \ldots, d_{N_B}(t)\}$
    Run the MADDPG Training Algorithm algorithm to train the agents
    Update the state of each battery i: $s_i(t) \leftarrow s_i^{(\eta)}(t+1)$
    score $\leftarrow$ *score* $+ \sum_{i=1}^{N_B} r_i^{(\eta)}$
  end
  if *score* $>$ *best_score* **then**
    Save the model
    best_score $\leftarrow$ *score*
  end
end

Αλγοριθμος Α'.3: *Αξιολόγηση με Lagrangian Decomposition*

Define P_{disp} and L^{real} for all timesteps
for *each battery* i **do**
  Receive initial state observations $s_i(t) = s_i = (SoE_i(t), \lambda(t), t)$
end
Initialize $\lambda^{(0)}(0)$
$cost \leftarrow 0$ **for** $t = 0 : 1 : T - 1$ **do**
  if $t = 0$ **then**
    $\widetilde{L}(t) = L^{real}(t)$
  else
    $\widetilde{L}(t) = L^{real}(t-1)$
  end
  Initialize $a^{(0)}, \eta \leftarrow 0$
  while $\eta \leq H_{max} \wedge |\lambda^{(\eta)} - \lambda^{(\eta-1)}| \geq \epsilon$ **do**
    Compute $P^{(\eta)}(t)$ using3.13
    for *each battery* i **do**
      Select action $B_i^{(\eta)}(t) = B_i^{(\eta)} = clip(\mu_i(s_i^{(\eta)}(t); \vartheta_i), B_i^{min}, B_i^{max})$,
      where $s_i^{(\eta)}(t) = (SoE_i(t), \lambda^{(\eta)}(t), t)$
    end
    Compute $\lambda^{(\eta+1)}$ using 3.12 and update the state of each battery with it
    Compute $a^{(\eta+1)}$
    $\eta \leftarrow \eta + 1$
  end
  for *each battery* i **do**
    Execute B_i in the environment
    Observe next state $s_i(t+1) = s_i' = (SoE_i^{(\eta)}(t+1), \lambda^{(\eta)}(t), t+1)$,
    where $SoE_i^{(\eta)}(t+1) = SoE_i(t) + B_i^{(\eta)}(t)\Delta t$
    Update its state: $s_i(t) \leftarrow s_i^{(\eta)}(t+1)$
  end
  $cost \leftarrow cost + (\sum_{i=1}^{N_B} B_i(t) + L^{real}(t) - P_{disp}(t))^2$
end

Αλγοριθμος Α'.4: *Εκπαίδευση χωρίς Lagrangian Decomposition*

for *each agent* i **do**
  Initialize critic $Q_i(\mathbf{S}, \mathbf{B}; \phi_i)$ and actor $\mu_i(s_i; \vartheta_i)$ network with weights ϕ_i and ϑ_i
  Initialize target critic Q_i^- and target actor μ_i^- network with $\phi_i^- \leftarrow \phi_i$, and $\vartheta_i^- \leftarrow \vartheta_i$
end
Initialize the Replay Buffer $\mathcal{D}$ and the Terminal Replay Buffer $\mathcal{D}_T$ as empty
Define P_{disp} and L^{real} for all timesteps
$best\_score \leftarrow -\infty$
for *each iteration until convergence* **do**
  $score \leftarrow 0$
  for *each battery* i **do**
    Receive initial state observation $s_i(0) = (SoE_i(0), 0)$
  end
  for $t = 0 : 1 : T - 1$ **do**
    if $t = 0$ **then**
      $\widetilde{L}(t) = L^{real}(t)$
    else
      $\widetilde{L}(t) = L^{real}(t-1)$
    end
    for *each battery* i **do**
      Select action $B_i(t) = B_i = clip(\mu_i(s_i(t); \vartheta_i), B_i^{min}, B_i^{max})$,
      where $s_i(t) = (SoE_i(t), t)$
    end
    Compute $P(t)$ using 3.20
    for *each battery* i **do**
      Execute $B_i(t)$ in the environment
      Observe next state $s_i(t+1) = s_i' = (SoE_i(t+1), t+1)$,
      where $SoE_i(t+1) = SoE_i(t) + B_i(t)\Delta t$
      Observe reward $r_i(t)$ based on 3.19 and done signal $d_i(t)$
    end
    if $t = T - 1$ **then**
      Store transition $(\mathbf{S}, \mathbf{B}, \mathbf{S}', \mathbf{R}, \mathbf{b})$ in Terminal Replay Buffer $\mathcal{D}_T$
    else
      Store transition $(\mathbf{S}, \mathbf{B}, \mathbf{S}', \mathbf{R}, \mathbf{b})$ in Replay Buffer $\mathcal{D}$
    end
    where $\mathbf{S} = \{s_1, \ldots, s_{N_B}\}$, $\mathbf{B} = \{B_1, \ldots, B_{N_B}\}$, $\mathbf{S}' = \{s_1', \ldots, s_{N_B}'\}$,
    $\mathbf{R} = \{r_1, \ldots, r_{N_B}\}$ and $\mathbf{d} = \{d_1(t), \ldots, d_{N_B}(t)\}$
    Run the MADDPG Training Algorithm algorithm to train the agents
    Update the state of each battery i: $s_i(t) \leftarrow s_i(t+1)$
    $score \leftarrow score + \sum_{i=1}^{N_B} r_i$
  end
  if $score > best\_score$ **then**
    Save the model
    $best\_score \leftarrow score$
  end
end

Αλγοριθμος Α'.5: *Αξιολόγηση χωρίς Lagrangian Decomposition*

Define P_{disp} and L^{real} for all timesteps
for *each battery* i **do**
  Receive initial state observations $s_i(0) = (SoE_i(0), 0)$
end
$cost \leftarrow 0$ **for** $t = 0 : 1 : T - 1$ **do**
  if $t = 0$ **then**
    $\widetilde{L}(t) = L^{real}(t)$
  else
    $\widetilde{L}(t) = L^{real}(t-1)$
  end
  for *each battery* i **do**
    Select action $B_i(t) = B_i = clip(\mu_i(s_i(t); \vartheta_i), B_i^{min}, B_i^{max})$,
    where $s_i(t) = (SoE_i(t), t)$
  end
  Compute $P(t)$ using 3.20
  for *each battery* i **do**
    Execute $B_i(t)$ in the environment
    Observe next state $s_i(t+1) = s_i' = (SoE_i(t+1), t+1)$,
    where $SoE_i(t+1) = SoE_i(t) + B_i(t)\Delta t$
    Update its state: $s_i(t) \leftarrow s_i(t+1)$
  end
  $cost \leftarrow cost + (\sum_{i=1}^{N_B} B_i(t) + L^{real}(t) - P_{disp}(t))^2$
end

Αλγοριθμος Α΄.6: *MADDPG training*

for *each agent i* **do**

Randomly sample a batch of M^D transitions, from $\mathcal{D}$

Randomly sample a minibatch of M^{D_T} transitions, from $\mathcal{D}_T$

where $M^D + M^{D_T} = M$

Concatenate the transitions of the two buffers

for *each transition j in the minibatch* **do**

Compute the target

$$y_i^j = r_i^j + \gamma(1 - d_i^j)\mathsf{Q}_i^-(\mathbf{S}'^j, B_1'^j, ..., B_{N_B}'^j|_{B_i'^j=\mu_i^-(s_i'^j|\partial_i^-)}; \varphi_i^-)$$

end

Update the agent's critic (Q-function) by minimizing the following loss, using one step gradient descent:

$$L(\varphi_i) = \frac{1}{M}\sum_{j=1}^{M}\left(y_i^j - \mathsf{Q}_i(\mathbf{S}^j, B_1^j, \ldots, B_{N_B}^j; \varphi_i)\right)^2$$

Update the agent's actor (policy) by one step of gradient ascent using:

$$\begin{aligned}\nabla_{\partial_i} J(\partial_i) &= \nabla_{\partial_i}\mathbb{E}_{\mathbf{S},\mathbf{B}\sim\mathcal{D}}\left[\mathsf{Q}_i(\mathbf{S}^j, B_1^j, \ldots, B_i, \ldots, B_{N_B}^j|_{B_i=\mu_i(s_i^j;\partial_i)}; \varphi_i)\right] \\ &= \frac{1}{M}\sum_{j=1}^{M}\nabla_{\partial_i}\mu_i(s_i;\partial_i)\nabla_{B_i}\mathsf{Q}_i(\mathbf{S}^j, B_1^j, \ldots, B_i, \ldots, B_{N_B}^j|_{B_i=\mu_i(s_i^j;\partial_i)}; \varphi_i)\end{aligned}$$

Update the weights of its target critic and target actor networks:

$$\varphi_i^- \leftarrow \tau\varphi_i^- + (1+\tau)\varphi_i$$
$$\partial_i^- \leftarrow \tau\partial_i^- + (1+\tau)\partial_i$$

end

Κεφάλαιο Β΄

Υπερπαράμετροι

Πίνακας Β΄.1: *Τιμές υπερπαραμέτρων*

Υπερπαράμετρος	**Σύμβολο**	**Τιμή**
Μέγεθος Replay Buffer	$\mathcal{D}$	1000000
Μέγεθος Terminal Buffer	$\mathcal{D}_T$	1000000
Πλήθος χρονικών βημάτων	T	90
Μέγεθος χρονικού βήματος	Δt	10s
Μέγεθος δείγματος (batch size)	M	200
Δείγματα από τον Replay Buffer	M^D	140 (70%)
Δείγματα από τον Terminal Replay Buffer	M^{D_T}	60 (30%)
Ρυθμός μάθησης actor	-	0.00001
Ρυθμός μάθησης critic	-	0.0001
Μέγεθος actor με LD	-	$3, 64, 64, 1$
Μέγεθος critic με LD	-	$4N_B, 64, 64, 64, 1$
Μέγεθος actor χωρίς LD	-	$2, 64, 64, 1$
Μέγεθος critic χωρίς LD	-	$3N_B, 64, 64, 64, 1$
Συνάρτηση βελτιστοποίησης	-	Adam
Σταθερά ενημέρωσης των target-networks	τ	0.001
Παράμετρος γ	γ	0.4

Βιβλιογραφία

[1] E.A. Committee. *Securing the 21st-Century Grid: The Potential Role of Storage in Providing Resilience, Reliability, and Security Services*. Recommendations for the U.S. Department of Energy, Τεχνική Αναφορά, 2018.

[2] J.da Silva André, E. Stai, O. Stanojev και G. Hug. *Battery Control with Lookahead Constraints in Distribution Grids using Reinforcement Learning*. *Electric Power Systems Research*, 211:108551, 2022.

[3] F. Charbonnier, T. Morstyn, M. D. McCulloch και Scalable Multi-agent. *Reinforcement Learning for Distributed Control of Residential Energy Flexibility*. *Applied Energy*, 314:118825, 2022.

[4] R. Lowe, Y. Wu, A. Tamar, P. Abbeel J. Harb και I. Mordatch. *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments*. *NIPS'17*, σελίδα 6382–6393, 2017.

[5] Dawei Qiu, Yujian Ye, Dimitrios Papadaskalopoulos και Goran Strbac. *Scalable coordinated management of peer-to-peer energy trading: A multi-cluster deep reinforcement learning approach*. *Applied Energy*, 292:116940, 2021.

[6] E. Stai, O. Stanojev, R. De Nardis Di Prata και G. Hug. *Distributed Reinforcement Learning for Real-Time Batteries Control Using Lagrangian Decomposition*. *International Conference on Smart Energy Systems and Technologies (SEST), Eindhoven, Netherlands*, σελίδες 1–6, 2022.

[7] A. Oroojlooy και D. Hajinezhad. *A Review of Cooperative Multi-Agent Deep Reinforcement Learning*. *Applied Intelligence*, 2019.

[8] Ι. Μαρινάκης, Α. Μυγδαλάς. *Συνδυαστική Βελτιστοποίηση*. Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1η έκδοση, 2016.

[9] D. P. Bertsekas. *Convex Optimization Theory*. Athena Scientific, 2009.

[10] A. L. Samuel. *Some studies in machine learning using the game of checkers*. *IBM Journal of research and development*, 3(3):210–229, 1959.

[11] Stuart J. Russell and Peter Norving. *Τεχνητή Νοημοσύνη: Μια σύγχρονη προσέγγιση*. Κλειδάριθμος, Αθήνα, 4η έκδοση, 2021.

[12] Nico Verbeeck, Richard M. Caprioli και Raf Vande Plas. *Unsupervised machine learning for exploratory data analysis in imaging mass spectrometry*. *Mass Spectrometry Reviews*, 39:245–291, 2020.

[13] H. Alashwal, M. El Halaby, J. Crouse, A. Abdalla και A. Moustafa. *The Application of Unsupervised Clustering Methods to Alzheimer's Disease*. *Frontiers in Computational Neuroscience*, 13, 2019.

[14] Gong Yu. *Social Network Analysis Based on BSP Clustering Algorithm*. *Communications of the IIMA*, 7, 2007.

[15] M. Faizan, M. F. A. Zuhairi, S. Ismail και S. Sultan. *Applications of Clustering Techniques in Data Mining: A Comparative Study*. *International Journal of Advanced Computer Science and Applications*, 11, 2020.

[16] Laurens Van Der Maaten, Eric O Postma και van H Jaap den Herik et al. *Dimensionality reduction: A comparative review*. *Journal of Machine Learning Research*, 10(66-71):13, 2009.

[17] Salima Omar, Asri Ngadi και Hamid H Jebur. *Machine learning techniques for anomaly detection: an overview*. *International Journal of Computer Applications*, 79(2), 2013.

[18] P. Kormushev, S. Calinon και D. G. Caldwell. *Reinforcement Learning in Robotics: Applications and Real-World Challenges*. *Robotics*, 2(3):122–148, 2013.

[19] X. Konstantia, G. Chalkiadakis και S. Afantenos. *Deep Reinforcement Learning in Strategic Board Game Environments*. *16th European Conference on Multi-Agent Systems (EUMAS 2018), Bergen, Norway*, 2018.

[20] Rui Nian, Jinfeng Liu και Biao Huang. *A review On reinforcement learning: Introduction and applications in industrial process control*. *Computers & Chemical Engineering*, 139:106886, 2020.

[21] Simon Haykin. *Νευρωνικά Δίκτυα και Μηχανική Μάθηση*. Παπασωτηρίου, Αθήνα, 3η έκδοση, 2010.

[22] T. H. Cormen, C. E. Leiserson, R. L. Rivest και C. Stein. *Εισαγωγή στους Αλγορίθμους*. Πανεπιστημιακές Εκδόσεις Κρήτης, Ηράκλειο, 3η έκδοση, 2018.

[23] Richard S. Sutton και Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, Massachusetts, 3η έκδοση, 2018.

[24] Μ. Λουλάκης. *Στοχαστικές Διαδικασίες*. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις, Αθήνα, 2015.

[25] David Silver. *Lectures on Reinforcement Learning*. https://www.davidsilver.uk/teaching/, 2015.

[26] *OpenAI Spinning Up*. https://spinningup.openai.com/en/latest/algorithms/ddpg.html.

[27] Michael L. Littman. *Markov Games as a Framework for Multi-Agent Reinforcement Learning*. *International Conference on Machine Learning*, 1994.

[28] J. B. Rawlings, D. Q. Mayne και M. M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2η έκδοση, 2020.

[29] C. R. Harris, K. J. Millman και S. J.van der Walt et al. *Array programming with NumPy*. *Nature*, 585(7825):357–362, 2020.

[30] *pandas - Python Data Analysis Library*. https://pandas.pydata.org. Ημερομηνία πρόσβασης: 23-1-2024.

[31] P. Virtanen, R. Gommers και T. E. Oliphantet al. *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*. *Nature Methods*, 17:261–272, 2020.

[32] M. Abadi, A. Agarwal και P. Barhamet al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. https://www.tensorflow.org/, 2015. Software available from tensorflow.org.

[33] J. D. Hunter. *Matplotlib: A 2D Graphics Environment*. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[34] *Multi-Agent-Deep-Deterministic-Policy-Gradients*. https://github.com/philtabor/Multi-Agent-Deep-Deterministic-Policy-Gradients.

[35] Greg Brockma, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang και Wojciech Zaremba. *OpenAI Gym*. *arXiv preprint arXiv:1606.01540*, 2016.

Συντομογραφίες - Αρκτικόλεξα - Ακρωνύμια

βλπ	βλέπε
κ.λπ.	και λοιπά
ΑΠΕ	Ανανεώσιμες Πηγές Ενέργειας
SoE	State of Energy
DSO	Distribution System Operator
PCC	Point of Common Coupling
DDPG	Deep Deterministic Policy Gradients
MARL	Multi-Agent Reinforcement Learning
MADDPG	Multi-Agent Deep Deterministic Policy Gradients
SGD	Stochastic Gradient Descent
ΜΔΑ	Μαρκοβιανή Διαδικασία Απόφασης
MPC	Model Predictive Control
LD	Lagrangian Decomposition

Απόδοση ξενόγλωσσων όρων

Απόδοση	Ξενόγλωσσος όρος
έξυπνο δίκτυο	smart grid
κατανεμημένος	distributed
συνάρτηση-Q	Q-function
στρώμα	layer
πράκτορας	agent
περιβάλλον	environment
παρατήρηση	observation
κατάσταση	state
δράση	action
ανταμοιβή	reward
παράγοντας μείωσης	discount factor
μοντέλο μετάβασης	transition matrix
πολιτική	policy
συνάρτηση χρησιμότητας	value function
δίκτυο δράστη	actor network
δίκτυο κριτή	critic network
δίκτυο στόχος	target network
πολλαπλασιαστής Lagrange	Lagrange multiplier
συγκεντρωτική	centralized
αποκεντρωμένη	decentralized
περιορισμός	constraint