ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Πτυχιακή Εργασία

Σταμάτιος Ορφανός

Ενισχυτική Μάθηση για τον Σχεδιασμό Τροχιών

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Επιβλέπων Καθηγητής Πτυχιακής Εργασίας: Γεώργιος Βούρος
Τίτλος Επιβλέποντος: Καθηγητής

ΠΕΙΡΑΙΑΣ 2021

Πτυχιακή Εργασία

Ενισχυτική Μάθηση για τον Σχεδιασμό Τροχιών

Ονοματεπώνυμο: **Σταμάτιος Ορφανός**
Αριθμός Μητρώου: **E17113**

Περίληψη

Στην παρούσα πτυχιακή εργασία, παρουσιάζουμε την αποδοτικότητα και αποτελεσματικότητα πολυπρακτορικών μεθόδων ενισχυτικής μάθησης, όπου οι αποφάσεις διαμορφώνονται σε διάφορα επίπεδα αφαίρεσης, για τη σύνθεση πολιτικών με σκοπό την επίλυση προβλημάτων σχεδιασμού τροχιών τόσο για μονοπρακτορικά όσο και για πολυπρακτορικά στοχαστικά περιβάλλοντα. Συγκεκριμένα, στοχεύουμε στην επίλυση δυο κύριων περιπτώσεων σχεδιασμού τροχιών. Η πρώτη περίπτωση περιλαμβάνει ένα πράκτορα που κινείται σε ένα χώρο ανάμεσα σε σταθερά εμπόδια ενώ στην δεύτερη περίπτωση στοχεύουμε στην επίλυση περιπτώσεων όπου η ζήτηση για χρήση του εναέριου χώρου υπερβαίνει την προσφορά, καθώς έχουμε πολλαπλά αεροδρόμια σε μικρή απόσταση σε σχέση μέγεθος της ευρύτερης περιοχής.

Οι πράκτορες, οι οποίοι αντιπροσωπεύουν πτήσεις, έχουν περιορισμένη πληροφόρηση σχετικά με το περιβάλλον τις αμοιβές και τις προτιμήσεις των υπόλοιπων πρακτόρων. Το πρόβλημα μοντελοποιείται σαν μια Μαρκοβιανή Διαδικασία Αποφάσεων και παρουσιάζουμε μεθόδους ενισχυτικής μάθησης που επιτρέπουν στους πράκτορες να διομορφώνουν τις δικές τους πολιτικές. Παράλληλα εξερευνούμε την αποτελεσματικότητα των μεθόδων ενισχυτικής μάθησης,με βάση τη βέλτιστη συμπεριφορά αποκτώντας εμπειρία μέσω αλληλεπιδράσεων δοκιμής και σφάλματος με το δυναμικό περιβάλλον του πλέγματος. Ωστόσο, η γνώση χτίζεται από το μηδέν και η εκμάθηση της συμπεριφοράς μπορεί να διαρκέσει πολύ, ειδικά με τη στοχαστική φύση του περιβάλλοντος. Για την επίτευξη της ενισχυτικής μάθησης χρησιμοπούμε δυο βασικές μεθόδους, οι οποίες είναι η Ε-Άπλειστη Στρατηγική και μια υβριδική προσέγγιση Monte Carlo Learning και Temporal Difference Learning .

Λέξεις Κλειδιά: Ενισχυτική Μάθηση, Χ Μάθηση, Ε-Άπλειστη Στρατηγική, Monte Carlo Μάθηση, Μάθηση Χρονικών Διαφορών, Διαδικασίες Απόφασης Markov , Συνάρτηση Αναμοιβής, Πολυ-πρακτορικά Συστήματα

Abstract

In this undergraduate thesis, we present the efficiency and effectiveness of multi-agent Reinforcement learning methods, where decisions are made at different levels of abstraction, for policy formulation in order to solve trajectory design problems for both single-agent and multi-agent stochastic environments. Specifically, we aim to solve two main cases of trajectory design. The first case involves an agent moving in a stochastic environment between fixed obstacles, while in the second case we aim to solve cases where multiple agents need to move in the environment with safety, by avoiding coming close to each other.

Agents, who represent flights, have limited information about the environment, rewards and preferences of other agents. The problem is modeled as a Markov Decision Process (MDP) and we present reinforcing learning methods that allow agents to formulate their own policies. At the same time, we explore the effectiveness of reinforcement learning methods, based on optimal behavior, gaining experience through trial and error interactions with the dynamic environment. However, knowledge is built from scratch and learning behavior can take a long time, especially with the stochastic nature of the environment. To achieve reinforcement learning we use two basic algorithms, which are the E-Greedy Strategy and a hybrid approach of Monte Carlo Learning and Temporal Difference Learning.

Key Words: Reinforcement learning, Q Learning, E-Greedy Stradegy, Monte Carlo Leanring, Temporal Difference Leanring, Markov Decision Process, Reward Function, Multi-agent systems

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα εργασία αποτελεί την πτυχιακή εργασία στα πλαίσια του προπτυχιακού προγράμματος σπουδών του τμήματος Ψηφιακών Συστημάτων. Πριν την παρουσίαση των αποτελεσμάτων της παρούσας διπλωματικής εργασίας, αισθάνομαι την υποχρέωση να ευχαριστήσω ορισμένους από τους ανθρώπους που γνώρισα, συνεργάστηκα μαζί τους και έπαιξαν πολύ σημαντικό ρόλο στην πραγματοποίησή της.

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέπων καθηγητή, κύριο Γεώργιο Βούρο, για την εμπιστοσύνη που μου έδειξε εξ' αρχής, αναθέτοντάς μου το συγκεκριμένο θέμα, την επιστημονική του καθοδήγηση, τις υποδείξεις, το αμείωτο ενδιαφέρον, τη συμπαράστασή και τη συνεχή του υποστήριξη καθ΄ όλη τη διάρκεια της πτυχιακής εργασίας.

Περιεχόμενα

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

1 Εισαγωγή

Οι πρόσφατες αλγοριθμικές και θεωρητικές εξελίξεις στο πεδίο της Ενισχυτικής μάθησης[1] έχουν προσελκύσει ενδιαφέρον για τις πιθανές εφαρμογές αυτού του πεδίου Τεχνητής Νοημοσύνης. Τα πολύπλοκα προβλήματα μεγάλων state space με στοιχεία στοχαστικότητας είναι οι βασικές επιλογές και προκλήσεις για την εφαρμογή μεθόφων Reinforcement Learning (RL). Οι μέθοδοι αυτοί προσεγγίζουν τον δυναμικό προγραμματισμό και μπορούν να εκπαιδευτούν σε πραγματικά ή προσομοιωμένα περιβάλλοντα, εστιάζοντας τον υπολογισμό τους σε κάθε κατάσταση ενός state space που επισκέπτονται κατά τη διάρκεια της εκτέλεσης των πειραματικών μελετών, καθιστώντας τους υπολογιστικά εφικτούς σε πολύπλοκα προβλήματα.

Οι περισσότερες από τις επιτυχημένες εφαρμογές RL, βρίσκονται σε παιχνίδια όπως Go και Poker, στη ρομποτική και στην αυτόνομη οδήγηση, εφαρμογές που περιλαμβάνουν τη συμμετοχή περισσότερων από έναν μεμονωμένων πρακτόρων, που εμπίπτει στη σφαίρα των πολλαπλών πρακτόρων RL (MARL),ενός τομέα με μια ενδιαφέρουσα ιστορία. Παράλληλα με την ανάπτυξη μεθόδων RL συστήματα πολλαπλών πρακτόρων έχουν χρησιμοποιηθεί για την αντιμετώπιση προβλημάτων σε διάφορους τομείς, συμπεριλαμβανομένων του κατανεμημένου ελέγχου, των τηλεπικοινωνιών και των οικονομικών. Η πολυπλοκότητα πολλών εργασιών που προκύπτουν σε αυτά τα domain τους καθιστά δύσκολο να επιλυθούν με προγραμματισμένες συμπεριφορές πράκτορα.

Δυστυχώς τα προβλήματα που αναφέραμε παραπάνω ανήκουν σε ένα δύσκολο τομέα που θέτει έναν συνδυασμό προκλήσεων που δεν εμφανίζονται στα περισσότερα μαθησιακά προβλήματα Ενισχυτικής μάθησης. Οι πράκτορες λαμβάνουν συνήθως ένα σήμα ανταμοιβής που εμφανίζεται σε κάθε έναν, λόγω των αποτελεσμάτων των ενεργειών τους ή των ενέργειων των άλλων παραγόντων ή ακόμα και του περιβάλλοντος. Παρά τις πολλές επιπλοκές, οι μέθοδοι RL δείχνουν αποτελέσματα που ξεπερνούν τις καλύτερους ευρετικούς αλγόριθμους αναζήτησης, που γνωρίζουμε.

Το κύριο χαρακτηριστικό αυτής της εργασίας είναι η προσέγγιση του πραγματικού κόσμου μέσω της εισαγωγής στοχαστικότητας στις ενέργειες του πράκτορα. Γνωρίζουμε ότι σε στοχαστικά περιβάλλοντα μια σταθερή ακολουθία ενεργειών δεν επιλύει πάντα το πρόβλημα. Συνεπώς η λύση του προβλήματος θα πρέπει να προσδιορίζει τι θα πρέπει να κάνει ένας πράκτορας σε οποιαδήποτε κατάσταση έχει βρεθεί. Μια τέτοια λύση ονομάζεται Πολιτική[2] $\Pi(S)$ και αν είναι πλήρης τότε ο πράκτορας θα γνωρίζει τι να κάνει σε κάθε κατάσταση ανεξαρτήτως του αποτελέσματος των ενεργειών του. Δεδομένου ότι η στοχαστικότητα είναι σταθερό χαρακτηριστικό του περιβάλλοντος οι τελική ενέργεια του πράκτορα μπορεί αν αλλάξει είτε εκτελώντας την πολιτική που έχει αναπτύξει είτε κινείται τυχαία στο χώρο. Σε αυτό το σημείο μπορούμε να καταλάβουμε ακόμα ένα πλεονέκτημα της Ενισχυτικής Μάθησης, καθώς κατά τον Σχεδιασμό-Planning εκτελούμε μια διαδικασία μάθησης με βάση ένα μοντέλο που παρέχει με λεπτομέρεια τον χώρο αναζήτησης, την συνάρτηση μετάβασης και την συνάρτηση ανταμοιβής ως είσοδο και μια πολιτική για περιβάλλον ως έξοδο. Αντίθετα με την Ενισχυτική Μάθηση ως είσοδο απαιτούμε μόνο ένα σύνολο διακριτών επεισόδιων και τις αντίστοιχες ανταμοιβές για την δημιουργία μιας πολιτικής.

Επίσης ένα ακόμα σημαντικό ζήτημα για την λύση τέτοιου είδους προβλημάτων είναι η μεγαλή προγραμματιστική πολυπλοκότητα, η οποία απαιτείται για την μοντελοποίηση αυτών των προβλημάτων. Έτσι προσεγγίζοντας αυτά τα προβλημάτα με την χρήση RL επιτυγχάνουμε συχνά την μείωση της πολυπλοκότητας με την χρήση State-Action ζευγαριών.

Η Διαχείριση Εναέριας Κυκλοφορίας-Air-Traffic Management (ATM) ήταν το βασικό κίνητρο για την ανάπτυξη αυτής της εργασίας, καθώς ο Υπολογισμός των Βέλτιστων Τροχιών-Trajectory-based Operations (TBO)για τις πτήσεις είναι ένα περίπλοκο πρόβλημα ύψιστης σημασίας για την βελτιστοποίηση της χρήσης του εναέριου χώρου. Η αυξημένη ζήτηση για την μετακίνηση με εναέρια μέσα έχει οδηγήσει τόσο στην αύξηση του αριθμού των αεροδρομίων όσο και των πτήσεων ανά αεροδρόμιο, δημιουργώντας μια ανάγκη για την βελτιστοποίηση των τροχιών των πτήσεων. Προφανώς ο εναέριος χώρος είναι πεπερασμένος και στόχος είναι η βέλτιστη χρήση του, μέσω της μεγιστοποίησης του αριθμού πτήσεων που μπορεί με ασφάλεια να προσφέρει.

Με βάση τις παραπάνω πληροφορίες, η παρούσα πτυχιακή εργασία μοντελοποιεί το παραπάνω πρόβλημα για δυο ξεχωριστές περιπτώσεις χρησιμοποιώντας την Ενισχυτική Μάθηση ως το μέσο για την λύση. Αρχικά οι δυο βασικές περιπτώσεις αφορούν ένα μονοπρακτορικό περιβάλλον με σταθερά εμπόδια τα οποία αντιπροσωπεύουν περιοχές πάνω από τις οποίες δεν επιτρέπονται οι πτήσεις, ενώ η δεύτερη περίπτωση αφορά ένα πολυπρακτορικό περιβάλλον με πολλαπλά σημεία εκκίνησης και τέλους. Για την σχεδίαση του περιβάλλοντος χρησιμοποιήσαμε ένα πλέγμα-grid,το οποίο αντιπροσωπεύει τον διαθέσιμο εναέριο χώρο στον οποίο θα κινούνται οι πράκτορες.

Η Ενισχυτική Μάθηση επιτρέπει σε κάθε πράκτορα να μάθει τη βέλτιστη συμπεριφορά αποκτώντας εμπειρία μέσω αλληλεπιδράσεων δοκιμής και σφάλματος με το δυναμικό περιβάλλον του πλέγματος. Ωστόσο, η γνώση χτίζεται από το μηδέν και η εκμάθηση της συμπεριφοράς μπορεί να διαρκέσει πολύ, ειδικά με τη στοχαστική φύση του περιβάλλοντος. Μέχρι το τέλος της εκπαίδευσης, ο πράκτορας θα πρέπει να έχει μια πολιτική που καθορίζει την ενέργεια με τη μέγιστη δυνατή ανταμοιβή από τον πίνακα Q για κάθε κατάσταση του πλέγματος. Για την επίτευξη της ενισχυτικής μάθησης χρησιμοποιούμε δυο βασικές μεθόδους, οι οποίοι είναι η E-Greedy Strategy και μια υβριδική προσέγγιση Monte Carlo Learning και Temporal Difference Learning .

Κάποια αρχικά πειράματα είχαν θετικά αποτελέσματα, καθώς μετά από ένα λογικό αριθμό επεισοδίων ο πράκτορας συγκλίνει σε μια πολιτική που τον οδηγεί στον στόχο του, το αεροδρόμιο για προσγείωση. Φυσικά διεξάγαμε πολλά πειράματα έτσι ώστε να έχουμε μια ακριβή εικόνα για την απόδοση των δυο μεθόδων ενισχυτικής μάθησης για τις δυο παραπάνω περιπτώσεις. Στην συνέχεια χρησιμοποιήσαμε τεχνικές ανάλυσης δεδομένων για την ερμηνεία των αποτελεσμάτων όπως καμπύλες ανταμοιβών, ιστογράμματα απόδοσης και πάνω από όλα τα heatmaps προκειμένου να γνωρίζουμε την τροχιά του πράκτορα όταν ακολουθεί την πολιτική που έχει δημιουργήσει από την εμπειρία του.

Ενισχυτική Μάθηση για τον Σχεδιασμό Τροχιών

Οι συνεισφορές αυτής της εργασίας σε αυτό το πεδίο είναι οι εξής:

1. Μοντελοποίηση στοχαστικού περιβάλλοντος με παράγοντες όπως ο αέρας που επηρεάζει ταχύτητα και κατεύθυνση.

2. Οι μέθοδοι μάθησης έχουν αξιολογηθεί σε διαφορετικά περιβάλλοντα με τυχαία σημεία αρχής και τέλους.

3. Δημιουργία μοντέλων ενισχυτικής μάθησης για μονοπρακτορικό περιβάλλον.

4. Δημιουργία μοντέλων ενισχυτικής μάθησης για πολυπρακτορικό περιβάλλον, χωρίς ιεραρχία ή επικοινωνία πρακτόρων.

Η δομή αυτής της πτυχιακής έχει την παρακάτω μορφή:

1. Κεφάλαιο 2 παρέχει πληροφορίες για έρευνα και μελέτες στο θέμα του σχεδιασμού τροχιών.
2. Κεφάλαιο 3 παρέχει πληροφορίες για την μοντελοποίηση του προβληματος.
3. Κεφάλαιο 4 παρουσιάζει τις προτεινόμενες λύσεις για το πρόβλημα.
4. Κεφάλαιο 5 παρουσιάζει τα αποτελέσματα των πειραματικών μελετών.
5. Κεφάλαιο 6 περιλαμβάνει το τελικό συμπέρασμα για την αποδοτικότητα των μεθόδων που χρησιμοποιήθηκαν.
6. Κεφάλαιο 7 αναλύει μελλοντική εργασία για την αύξηση της αποδοτικότητας και της λειτουργικότητας της εφαρμογής
7. Κεφάλαιο 8 περιλαμβάνει αναφορές και ακρωνύμια

2 Διατύπωση Προβλήματος

Όπως έχουμε αναφέρει παραπάνω, η παρούσα εργασία εστιάζει στο πρόβλημα της βελτιστοποίησης των τροχιών σε δυο βασικές περιπτώσεις.

Συνεπώς μπορούμε να καταλάβουμε ότι το περιβάλλον έχει μεγάλη σημασία για τα αποτελέσματα των πειραματικών μελετών, καθώς αναφερόμαστε σε ένα πεπερασμένο πόρο για τον οποίο δημιουργούνται συγκρούσεις ζήτησης-προσφοράς.

Σε αυτή την ενότητα παρουσιάζουμε τα βασικά χαρακτηριστικά του περιβάλλοντος, την χρήση της Διαδικασίας Αποφάσεων Markov και την συνάρτηση ανταμοιβής για κάθε περίπτωση που μελετήσαμε σε αυτή την εργασία.

2.1 Περιβάλλον του Προβλήματος

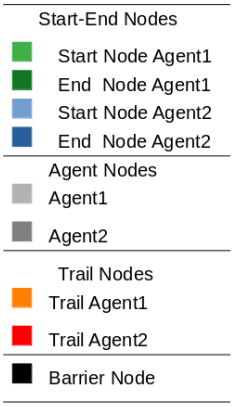
Το περιβάλλον της εφαρμογής αποτελείται από ένα πλέγμα-grid μεγέθους NxN, όπου N είναι το επιθυμητό μέγεθος του πλέγματος. Ένα από τα βασικά χαρακτηριστικά αυτής της εργασίας είναι η δυνατότητα δυναμικής επιλογής του μεγέθους του πλέγματος, γεγονός που μας δίνει πολλές δυνατότητες για πειραματικές μελέτες. Κάθε κόμβος του πλέγματος αντιστοιχεί σε ένα τετραγωνικό χιλιόμετρο, παραδοχή που μας βοηθά ιδιαίτερα με την συνάρτηση ανταμοιβής και κάποιους περιορισμούς απόστασης που θα αναλύσουμε σε επόμενες ενότητες. Σε αυτή την εργασία εξερευνούμε δυο διαφορετικές περιπτώσεις περιβάλλοντος όσον αφορά αριθμό πρακτόρων.

(α′) Grid size 10x10 (β′) Grid size 20x20 (γ′) Grid size 50x50

Εικόνα 1: Είδη Πλέγματος

Η πειραματική μελέτη και τα αντίστοιχα αποτελέσματα έγιναν σε ένα πλέγμα μεγέθους 20x20, καθώς ήταν μια καλή μέση λύση όπου ο πράκτορας μαθαίνει σταδιακά το περιβάλλον σε λογικό αριθμό επεισοδίων. Ανεξαρτήτως του αριθμού των πρακτόρων, το περιβάλλον περιλαμβάνει και άλλα βασικά χαρακτηριστικά όπως η μοντελοπίηση της στοχαστικότητα του πραγματικού κόσμου με αρκετή ακρίβεια στην εφαρμογή.

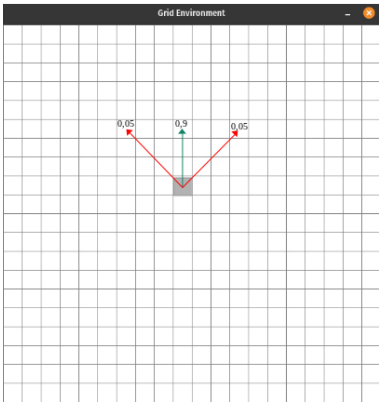
Τα βασικά συστατικά του περιβάλλοντος είναι τα εξής:



Εικόνα 2: Συστατικά Περιβάλλοντος

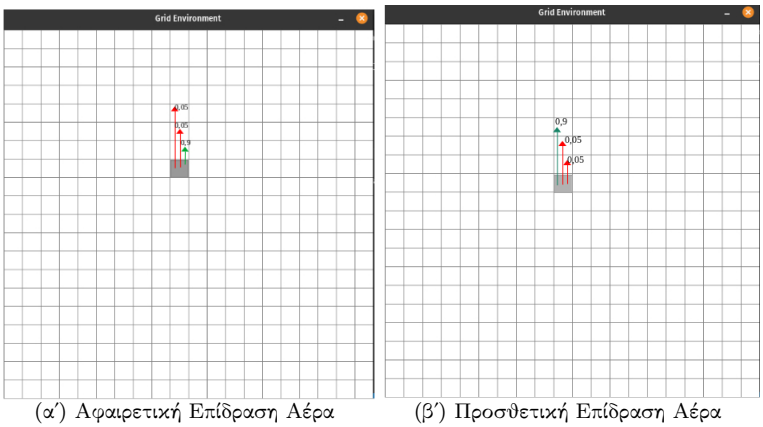
Λαμβάνοντας υπόψη το γεγονός ότι η εργασία αυτή επικεντρώνει την προσοχή της σε ένα κομμάτι της διαχείρισης της εναέριας κυκλοφορίας, είναι απαραίτητο να συμπεριλάβουμε την επίδραση του αέρα στην εφαρμογή μας. Γνωρίζοντας ότι ο αέρας αλλάζει σε κατεύθυνση και ισχύ, χρησιμοποιήσαμε μια συνάρτηση η οποία για κάθε N βήματα του πράκτορα θα αλλάζει την ισχύ του ανέμου. Φυσικά όπως και στον πραγματικό κόσμο η επίδραση μπορεί να είναι προσθετική ή αφαιρετική, επιταχύνοντας ή επιβραδύνοντας την κίνηση του πράκτορα αντίστοιχα. Πιο συγκεκριμένα η επίδραση του αέρα μπορεί να αλλάξει μια ενέργεια με δυο τρόπους. Αρχικά μια ενέργεια αποτελείται από δυο βασικές παραμέτρους, την κατεύθυνση και την ταχύτητα.

Ξεκινώντας με την παράμετρο της κατεύθυνσης η επίδραση του αέρα μπορεί να αλλάξει την τελική πορεία του πράκτορα κατά 45° σε σχέση με την αρχική κατεύθυνση. Παρακάτω παρατηρούμε ότι ο πράκτορας έχει ως στόχο να κινηθεί βόρεια, αλλά το περιβάλλον μπορεί να αλλάξει το αποτελέσματα με πιθανότητα 10% προς την βορειοανατολική ή βορειοδυτική κατεύθυνση.



Εικόνα 3: Στοχαστικότητα Κατεύθυνσης

Όσον αφορά την δεύτερη παράμετρο ο αέρας μπορεί να επιταχύνει ή να επιβραδύνει την ταχύτητα του πράκτορα, με μέγιστη δυνατή τιμή ταχύτητας τρεις κόμβους και ελάχιστη τιμή έναν κόμβο του πλέγματος.

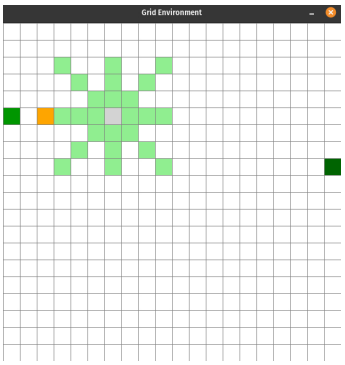


(α′) Αφαιρετική Επίδραση Αέρα (β′) Προσθετική Επίδραση Αέρα

Εικόνα 4: Στοχαστικότητα Ταχύτητας

2.1.1 Δυνατότητες Πρακτόρων

Σε αυτό το σημείο είναι σημαντικό να αναφέρουμε ότι ο πράκτορας είναι το βασικό συστατικό της εφαρμογής. Όσον αφορά τις λειτουργίες στο περιβάλλον ένας πράκτορας μπορεί να κινηθεί σε μια ακτίνα 3 βημάτων στις κατεύθυνσεις Βόρεια, Βορειοανατολικά, Ανατολικά, Νοτιοανατολικά, Νότια, Νοτιοδυτικά, Δυτικά, Βορειοδυτικά, όπως φαίνεται παρακάτω:

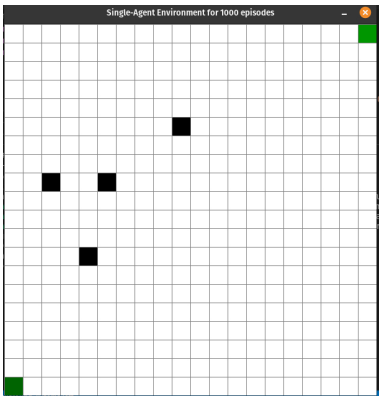


Εικόνα 5: Με ανοιχτό πράσινο χρώμα είναι όλες οι πιθανές ενέργειες του πράκτορα

2.1.2 Μονοπρακτορικό Περιβάλλον

Παραπάνω αναφέραμε το βασικό πλαίσιο για την δημιουργία του περιβάλλοντος, αλλά είναι σημαντικό να περιγράψουμε με λεπτομέρεια τις ιδιαιτερότητες της κάθε περίπτωσης. Στην περίπτωση του μονοπρακτορικού προβλήματος το περιβάλλον αποτελείται από ένα σημείο εκκίνησης και ένα σημείο τερματισμού, όπου στόχος του πράκτορα είναι ξεκινώντας από το σημείο εκκίνησης να φτάσει στο σημείο τερματισμού όσο το δυνατόν γρηγορότερα.

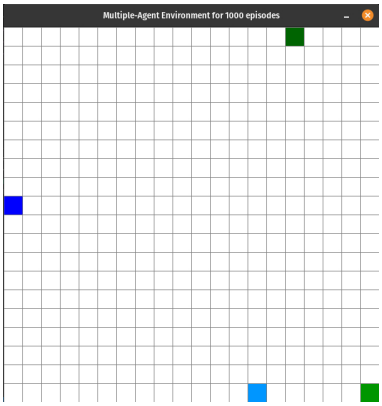
Επιπλέον ένα βασικό χαρακτηριστικό του μονοπρακτορικού περιβάλλοντος είναι τα εμπόδια που υπάρχουν στο πλέγμα, τα οποία είναι τυχαία κατανεμημένα στο εσωτερικό του πλέγματος. Το πλήθος των εμποδίων εξαρτάται από το μέγεθος του πλέγματος, διότι τα εμπόδια θα είναι το κατώφλι της ρίζας του συνόλου των συνολικών γραμμών του πλέγματος. Τα εμπόδια θεωρούνται ως περιοχές πάνω από τις οποίες δεν μπορεί να κινηθεί ο πράκτορας.



Εικόνα 6: Μονοπρακτορικό Περιβάλλον

2.1.3 Πολυπρακτορικό Περιβάλλον

Αντίθετα στην περίπτωση του πολυπρακτορικού προβλήματος το περιβάλλον αποτελείται από δυο σημεία εκκίνησης και τερματισμού, ένα για κάθε πράκτορα. Ειδικότερα στόχος ενός πράκτορα σε αυτό το περιβάλλον είναι να φτάσει όσο το δυνατόν γρηγορότερα στον στόχο, ενώ παράλληλα βρίσκεται σε ασφαλή απόσταση από τον άλλο. Οι πράκτορες δεν έχουν καμία επικοινωνία μεταξύ τους και πρέπει να μάθουν τις περιοχές που πρέπει να αποφεύγουν μέσω του σταδίου της εξερεύνησης του περιβάλλοντος. Σε αυτή την περίπτωση τα εμπόδια είναι το trail-μονοπάτι του άλλου πράκτορα. Ωστόσο μια κύρια διαφορά είναι το γεγονός ότι ένας πράκτορας μπορεί αν κινηθεί πάνω στο μονοπάτι του άλλου, με κάποιες αρνητικές επιπτώσεις που θα συζητήσουμε παρακάτω.



Εικόνα 7: Πολυπρακτορικό Περιβάλλον

2.2 Διαδικασία Αποφάσεων Markov

Το πρόβλημα της εύρεσης του βέλτιστου μονοπατιού μπορεί να μοντελοποιηθεί ως ένα ακολουθιακό πρόβλημα απόφασης, το οποίο σε συνδυασμό με τις προσθετικές ανταμοιβές και το μοντέλο μετάβασης θα μπορούσε να χρησιμοποιηθεί ως το μέσο για την λύση του προβλήματος. Ένα Markov Decision Process περιλαμβάνει τα παρακάτω βασικά συστατικά:

1. Αρχική Κατάσταση: S_0

2. Μοντέλο Μετάβασης: $T(S, a, S')$ είναι ο προσδιορισμός των πιθανοτήτων του αποτελέσματος για κάθε ενέργεια σε κάθε δυνατή κατάσταση.[3]

3. Συνάρτηση Ανταμοιβής: $R(S)$

Ωστόσο για την χρήση ενός MDP είναι απαραίτητο ο πράκτορας να γνωρίζει το περιβάλλον μέσω της συνάρτησης μετάβασης και της συνάρτησης ανταμοιβής, δεδομένα που δεν διαθέτει. Ένα από τα βασικά χαρακτηριστικά των πρακτόρων είναι ότι δεν γνωρίζουν τίποτα για το περιβάλλον και μπορούν να μάθουν το μοντέλο μετάβασης σταδιακά με την εξερεύνηση του περιβάλλοντος.

2.3 Συνάρτηση Ανταμοιβής

Για το TBO πρόβλημα έχουμε δημιουργήσει μια συνάρτηση ανταμοιβής, η οποία λαμβάνει όλες τις παραμέτρους του περιβάλλοντος. Παρακάτω αριθμούμε όλες τις πιθανές ανταμοιβές που θα λάβει ένας πράκτορας:

1. Ανταμοιβή Περιβάλλοντος: -0.1

2. Ανταμοιβή Εμποδίων: -0.5

3. Ανταμοιβή Στόχου: +15

Το περιβάλλον έχει μια μικρή αρνητική ανταμοιβή, έτσι ώστε να ενθαρύνει τον πράκτορα να βρει τον τελικό στόχο ή να βγεί από το πλέγμα. Το πλέγμα της εφαρμογής είναι μεγέθους 20x20 , άρα σε μια μέση περίπτωση όπου ο πράκτορας κάνει 15 βήματα στο πλέγμα τότε η ανταμοιβή για αυτό το επεισόδιο θα είναι -1.5, μια μεγάλη αρνητική ανταμοιβή δεδομένου ότι λαμβάνουμε την βέλτιστη ανταμοιβή πολύ σπάνια στο στάδιο εξερεύνησης.

Ανάλογα με το περιβάλλον η συνάρτηση ανταμοιβής διαφέρει ελαφρώς όσον αφορά την απόδοση των ανταμοιβών των εμποδίων.

2.3.1 Μονοπρακτορικό Περιβάλλον

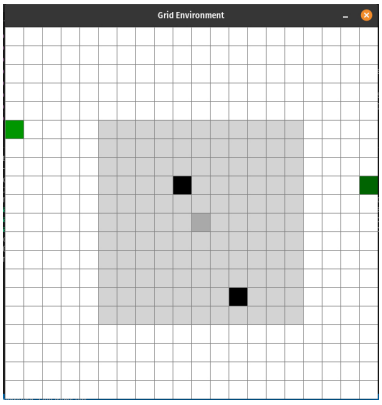
Ο στόχος του πράκτορα στο μονοπρακτορικό περιβάλλον είναι να μετακινείται στο πλέγμα για να φτάσει στο στόχο, ενώ παράλληλα να αποφεύγει τα σταθερά εμπόδια. Σύμφωνα με κανονισμούς της εναέριας κυκλοφορίας απαιτείται μια ελάχιστη απόσταση 5 μιλίων μεταξύ ενός αεροπλάνου και οποιαδήποτε άλλης οντότητας στον χώρο, είτε αεροπλάνο είτε άλλος περιορισμός. Συνεπώς μπορούμε να καταλάβουμε ότι η απόσταση είναι βασικός παράγοντας για το μέγεθος της αρνητικής ανταμοιβής, και μια συνάρτηση απόστασης εμποδίων μαζί με μια συνάρτηση εύρεσης εμποδίων είναι απαραίτητες.

Απόσταση Manhattan: η απόσταση Manhattaneίναι η κατάλληλη συνάρτηση απόστασης για τον παρόν πρόβλημα, καθώς έχουμε ένα πλέγμα ως περιβάλλον.

$$d(p,q) = \sum |p_i - q_i| = |x_1 - x_2| + |y_1 - y_2|$$

, όπου (p, q) είναι δισδιάστατοι πίνακες $p = (x_1, y_1)$, $q = (x_2, y_2)$.

Συνάρτηση Εύρεσης Εμποδίων: η συνάρτηση αυτή έχει ως στόχο να βρει όλα τα εμπόδια σε μια ακτίνα 5 μιλίων.



Εικόνα 8: Παράδειγμα όπου σταθερά εμπόδια βρίσκονται στη restricted περιοχή του πράκτορα, ο οποίος κινείται στο περιβάλλον

Συνάρτηση Ανταμοιβής: η συνάρτηση ανταμοιβής αποτελείται από το άθροισμα της ανταμοιβής του κόμβου και το άθροισμα των ανταμοιβών των εμποδίων εντός των 5 μιλίων διαιρεμένο με την ManhattanΑπόσταση έτσι ώστε να έχουμε μεγαλύτερη αρνητική ανταμοιβή όταν ο πράκτορας πλησιάζει σε εμπόδιο.

$$R(S) = r(S) + \sum Barriers_{reward}$$

όπου,

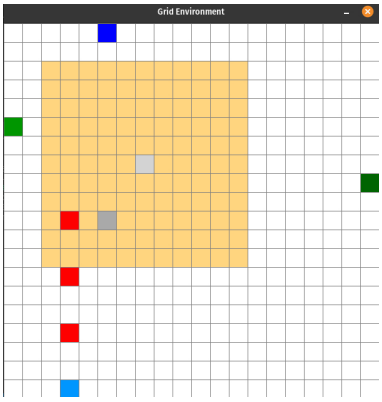
- $r(S)$ είναι η ανταμοιβή περιβάλλοντος ίση με -0.1 όταν ο πράκτορας βρίσκεται σε απλό κόμβο του περιβάλλοντος και +15 όταν βρεθεί στον κόμβο που βρίσκεται ο τελικός στόχος.
- $Barriers_{reward}$ είναι η συνάρτηση που βρίσκει όλα τα εμπόδια εντός της restricted περιοχής και εισάγει σε μια λίστα όλες τις σταθμισμένες ανταμοιβές.

2.3.2 Πολυπρακτορικό Περιβάλλον

Ο στόχος του πράκτορα στο πολυπρακτορικό περιβάλλον είναι να μετακινείται στο πλέγμα για να φτάσει στο στόχο, ενώ παράλληλα να αποφεύγει την κίνηση του άλλου πράκτορα. Τα εμπόδια σε αυτό το περιβάλλον αποτελούνται από δυο βασικά συστατικά:

1. Ο άλλος πράκτορας
2. Τροχιά του άλλου πράκτορα

Συνάρτηση Εύρεσης Εμποδίων: η συνάρτηση αυτή έχει ως στόχο να βρει όλα τα πιθανά εμπόδια σε μια ακτίνα 5 μιλίων. Σε αυτή την συνάρτηση είναι σημαντικό να σταθμίσουμε σωστά τις αρνητικές ανταμοιβές που θα λαμβάνει ένας πράκτορας. Πιο συγκεκριμένα η τροχιά του άλλου πράκτορα αντιμετωπίζεται ως ένα σταθερό εμπόδιο όπου σταθμίζουμε την αρνητική ανταμοιβή με βάση την απόσταση Manhattan της τροχίας από την τρέχουσα θέση του πράκτορα. Ενώ η παρουσία άλλου πράκτορα στην ακτίνα των πέντε μιλίων σημαίνει ότι ο πράκτορας θα λάβει μια μη σταθμισμένη αρνητική ανταμοιβή, καθώς βασικός στόχος είναι η ασφάλεια των πτήσεων.



Εικόνα 9: Παράδειγμα όπου στη restricted περιοχή ενός πράκτορα βρίσκεται τόσο το trail του άλλου πράκτορα όσο και ο άλλος πράκτορας του περιβάλλοντος

Συνάρτηση Ανταμοιβής: η συνάρτηση ανταμοιβής αποτελείται από το άθροισμα της ανταμοιβής του κόμβου και το άθροισμα των ανταμοιβών του άλλου πράκτορα.

$$R(S) = r(S) + \sum Barriers_{reward}$$

όπου,

- $r(S)$ είναι η ανταμοιβή περιβάλλοντος ίση με -0.1 όταν ο πράκτορας βρίσκεται σε απλό κόμβο του περιβάλλοντος και +15 όταν βρεθεί στον κόμβο που βρίσκεται ο τελικός στόχος.
- $Barriers_{reward}$ είναι η συνάρτηση που βρίσκει όλα τους κόμβους trail του άλλου πράκτορα εντός της restricted περιοχής και εισάγει σε μια λίστα όλες τις σταθμισμένες ανταμοιβές. Ομοίως η συνάρτηση θα κάνει έλεγχο για την ύπαρξη άλλου πράκτορα στην γνωστή ακτίνα και θα προσθέσει στην λίστα την κατάλληλη αρνητική ανταμοιβή.

3 Προτεινόμενη Λύση

Όπως έχουμε αναφέρει το βασικό εργαλείο για την λύση αυτού του προβλήματος είναι η ενισχυτική μάθηση, η οποία χωρίζεται σε δυο βασικές κατηγορίες της Παθητικής Ενισχυτικής Μάθησης[4] και της Ενεργητικής Ενισχυτικής Μάθησης[5]. Στην παθητική ενισχυτική μάθηση η πολιτική $\Pi(S)$ του πράκτορα είναι σταθερή και στην κατάσταση S θα εκτελεί πάντα την ενέργεια $\Pi(S)$. Αντιθέτως η ενεργητική ενισχυτική μάθηση χρησιμοποιεί τις διαδικασίες εξερεύνησης του περιβάλλοντος και της εκμετάλλευσης της γνώσης για να βρει την πολιτική και να την εφαρμόσει για να λύσει το πρόβλημα.

Σε αυτή την ενότητα παρουσιάζουμε την χρήση μεθόδων ενισχυτικής μάθησης για την λύση του πολύπλοκου προβλήματος της εργασίας. Αρχικά είναι σημαντικό να αναφέρουμε τους λόγους για τους οποίους επιλέξαμε τις μεθόδους E-Greedy Strategy και ένα υβριδικό μοντέλο των μεθόδων Monte-Carlo - Temporal Difference (TD). Ξεκινώντας μπορούμε να καταλάβουμε ότι η Παθητική Ενισχυτική Μάθηση δεν είναι η κατάλληλη τεχνική για την λύση του προβλήματος, διότι η στοχαστικότητα του περιβάλλοντος δεν θα επέτρεπε σε ένα πράκτορα να βρει τους βέλιστους συνδυασμούς state-action pairs, με βάση μια σταθερή πολιτική, επειδή σε διαφορετικά επεισόδια θα λάμβανε διαφορετική επίδραση του αέρα.

Συνεπώς η χρήση Ενεργητικής Ενισχυτικής Μάθησης ήταν η κατάλληλη κατεύθυνση για την εύρεση της βέλτιστης λύσης του προβλήματος, δεδομένου φυσικά ότι η εκπαίδευση των πρακτόρων στο περιβάλλον είναι επιτυχημένη. Επιπλέον είναι σημαντικό να σημειώσουμε ότι ο πράκτορας δεν γνωριζει τη συνάρτηση μετάβασης, άρα με την εξέλιξη των επεισοδίων ο πράκτορας θα αποκτήσει μια καλή προσέγγιση της λειτουργίας του περιβάλλοντος όσο παιρνούν τα επεισόδια δίνοντας μας ένα ακόμη λόγο για την επιλογή της EEM. Κάποιες από τις βασικές προϋποθέσεις για την επιτυχημένη εκπαίδευση των πρακτόρων στο περιβάλλον είναι οι παρακάτω:

1. Το μέγεθος του πλέγματος
2. Τη στοχαστικότητα του περιβάλλοντος
3. Ο αριθμός των επεισοδίων για την εκπαίδευση/εξερεύνηση
4. Η επιλογή των σταθερών μάθησης όπως λ και γ
5. Η επιλογή των τιμών για κάθε είδος ανταμοιβής

Με βάση τις παραπάνω προϋποθέσεις καταλαβαίνουμε ότι το πρόβλημα της παρούσας εργασίας είναι αρκετά περίπλοκο και απαιτείται αρκετός πειραματισμός για την σωστή παραμετροποίηση όλων των σταθερών. Επίσης είναι γνωστό ότι η γνώση του πράκτορα για το περιβάλλον ξεκινάει από το μηδέν και χρειάζεται χρόνος για να αποκτήσει μια καλή εικόνα του περιβάλλοντος και των κανόνων που υπάρχουν σε αυτό.

Πιο συγκεκριμένα στις παρακάτω ενότητες θα ορίσουμε μια συνάρτηση $Q(s, a)$, η οποία θα αξιολογεί την ανταμοιβή ύπαρξης ενός πράκτορα σε μια συγκεκριμένη κατάσταση και εκτέλεση μιας συγκεκριμένης ενέργειας για την μετάβαση του σε μια νέα κατάσταση. Η εξίσωση για την Q-Learning μάθηση που αφορά τα state-action pairs είναι η εξής:

$$Q(S, A) = R(S) + \gamma \sum T(S, A, S') * maxQ(S', A')$$

όπου,

- $Q(S, A)$ η τιμή του Q-table για την κατάσταση S και την ενέργεια A
- $R(S)$ η ανταμοιβή που λάμβάνει ο πράκτορας στην κατάσταση S
- γ είναι ο παράγοντας προεξόφλησης με τιμές μεταξύ των $0 - 1$ και περιγράφει την προτίμηση του πράκτορα για τρέχουσες ανταμοιβές έναντι των μελλοντικών ανταμοιβών
- $T(S, A, S')$ είναι το μοντέλο μετάβασης
- $maxQ(S', A')$ είναι η μέγιστη τιμή του Q-table για μια κατάσταση S' στην οποία μπορεί να μεταβεί ο πράκτορας από την κατάσταση S μέσω της ενέργειας A

Προτού ξεκινήσουμε με την ανάλυση της μεθόδου της Ε-Άπλειστης Στρατηγικής πρέπει να αναφέρουμε ότι το βασικότερο στοιχείο είναι ο Q-table. Έτσι πρέπει πρώτα να δημιουργήσουμε ένα πίνακα ο οποίος θα αποθηκεύει τις q-values . Το $Q(S, A)$ αντιστοιχεί στο state-action ζευγάρι, όπου S είναι η κατάσταση και A η ενέργεια. Ο Q-table για το περιβάλλον της εφαρμογής θα έχει την παρακάτω μορφή:

Table 1: Q-table

State-Action	((-1,0),1)	((-1,0),2)	((-1,0),3)	((-1,1),1)	...	((-1,-1),3)
(0,1)	0	0	0	0	0	0
(0,2)	0	0	0	0	0	0
(0,3)	0	0	0	0	0	0
(0,4)	0	0	0	0	0	0
(0,5)	0	0	0	0	0	0
....	0	0	0	0	0	0
(19,19)	0	0	0	0	0	0

Στην πρώτη γραμμή του πίνακα μπορούμε να δούμε την κωδικοποίηση των ενεργειών ενός πράκτορα, όπου η πρώτη παρένθεση αφορά την κατεύθυνση της κίνησης του πράκτορα και η τιμή δίπλα αφορά την απόσταση που θα καλύψει ο πράκτορας στην κίνηση του. Αναλυτικότερα χρησιμοποιήσαμε την βιβλιοθήκη pygame για την δημιουργία του πλέγματος, σύμφωνα με την οποία τα δεδομένα ενός πλέγματος σχεδιάζονται ως (x, y) αλλά αποθηκεύονται ως (y, x). Συνεπώς η πρώτη τιμή αφορά τις στήλες του πλέγματος ή την κατεύθυνση προς τα Ανατολικά ή Δυτικά και η δεύτερη τιμή αφορά τις γραμμές και τις κατευθύνσεις Βόρεια ή Νότια. Έτσι η κατεύθυνση μοντελοποιείται ως εξής:

- $(0, -1)$ Βόρεια
- $(1, -1)$ Βορειοανατολικά
- $(1, 0)$ Ανατολικά
- $(1, 1)$ Νοτιοανατολικά
- $(0, 1)$ Νότια
- $(-1, 1)$ Νοτιοδυτικά
- $(-1, 0)$ Δυτικά
- $(-1, -1)$ Βορειοδυτικά

3.1 E-Greedy Strategy

Δεδομένου ότι έχουμε αρχικοποιήσει τον πίνακα Q , μπορούμε να συνεχίσουμε με τον ορισμό της μεθόδου E-Greedy Q-Learning:

Algorithm 1: E-Greedy Strategy Q-Learning

Data: α: learning rate, γ: discount factor, ϵ: number between 0 and 1, Actions: list with all the possible actions, States: list with all the possible satates according to rows

Output: A Q-table containing Q(S,A) pair defining estimated optimal policy Π^*

```
/* Initialization of Q-table                                          */
Q(S, A) = makeQTable(States, Actions)

for each episode do
    Initialize/Reset the environment        // Initialize during the first run, or reset the environment
    for each step in episode do
        if S is not terminal node or out of the gird limits then
            A ← SELECT − ACTION(Q, S, ϵ)
            Take action A at state S, observe reward R and next state S'
            Q(S, A) ← Q(S, A) + a[R + γmax_α Q(S', a) − Q(S, a)]
            S ← S'
```

Στον παραπάνω ψευδοκώδικα ξεκινάμε με την αρχικοποίηση του Q-table , ο οποίος είναι το βασικό συστατικό αυτής της μεθόδου ενισχυτικής μάθησης. Αμέσως μετά συνεχίζουμε με το loop όπου τρέχουμε όλα τα επεισόδια του πειράματος. Στην συνέχεια προχωράμε με την εκτέλεση των βημάτων του πράκτορα στο συγκεκριμένο επεισόδιο, όπου σε κάθε βήμα θα πρέπει ο πράκτορας να επιλέξει μια ενέργεια A. Με την εκτέλεση της ενέργειας ο πράκτορας αλλάζει κατάσταση από την S στην S' και λαμβάνει την ανταμοιβή για την μετάβαση στην κατάσταση S' μέσω της ενέργειας A. Αφού λάβει την ανταμοιβή μπορεί να ενημερώσει την τιμή του πίνακα Q-Table για το state-action ζεύγος (S, A).

Μέχρι τώρα δεν έχουμε αναφέρει τη λειτουργία της συνάρτησης SELECT-ACTION η οποία είναι η βασική λειτουργία της μεθόδου. Για την καλύτερη κατανόηση της μεθόδου επιλογής ενεργειών είναι απαραίτητο να καταλάβουμε την λογική της Εξερεύνησης-Εκμετάλλευσης του περιβάλλοντος.

3.1.1 Εξερεύνηση-Εκμετάλλευση Περιβάλλοντος

Για την καλύτερη κατανόηση της συνάρτησης επιλογής ενέργειας SELECT-ACTION πρέπει να ορίσουμε τις έννοιες εξερεύνηση και εκμετάλλευση του περιβάλλοντος. Στην ενισχυτική μάθηση ένας πράκτορας προσπαθεί να ανακαλύψει το περιβάλλον. Ξεκινώντας με την διαδικασία της εξερεύνησης ο πράκτορας έχει ένα σύνολο από ενέργειες που μπορεί να επιλέξει. Στόχος του πράκτορα σε αυτό το σημείο είναι να επιλέξει όλες τις δυνατές ενέργειες για όλες τις καταστάσεις του πλέγματος έτσι ώστε να γνωρίζει ποιά ενέργεια αποφέρει την μεγαλύτερη ανταμοιβή για κάθε κατάσταση.

Στην συνέχεια έχουμε την εκμετάλλευση του περιβάλλοντος, όπου ο πράκτορας γνωρίζει τις ενέργειες που πιθανόν θα επιφέρουν το μέγιστο κέρδος για κάθε κατάσταση. Αντίστοιχα ο στόχος του πράκτορα σε αυτό το σημείο είναι να μεγιστοποιήσει το πιθανό κέρδος χρησιμοποιώντας την βέλτιστη ενέργεια σε κάθε κατάσταση.

Για την επιτυχία της ενισχυτικής μάθησης είναι πολύ σημαντικό να έχουμε την κατάλληλη ισορροπία μεταξύ των διαδικασιών εξερεύνησης και εκμετάλλευσης. Πιο συγκεκριμένα αν δεν έχουμε ένα καλό αποτέλεσμα από την διαδικασία εξερεύνησης είναι πολύ πιθανό ο πράκτορας να μάθει μια μη-βέλτιστη πολιτική την οποία θα εφαρμόσει στην διαδικασία εκμετάλλευσης της γνώσης για να λύσει το πρόβλημα μας. Στο παρόν πρόβλημα ένας παράγοντας που δυσχαιρένει την ισορροπία αυτή είναι η στοχαστικότητα του περιβάλλοντος, καθώς είναι πιθανό να μην συναντήσουμε όλες τις ενέργειες σε όλες τις καταστάσεις δίνοντας μας μια ανακριβή εικόνα του περιβάλλοντος.

3.1.2 E-Greedy Επιλογή Ενέργεια

Όπως αναφέραμε παραπάνω μια βασική λειτουργία της E-Greedy μεθόδου είναι η επιλογή ενέργειας με βάση την σωστή ισορροπία εξερεύνησης-εκμετάλλευσης.

Η ισορροπία αυτή γίνεται με βάση την παρακάτω μέθοδο:

Algorithm 2: E-Greedy Action Selection

Data: Q: q-table generated thus far, e:exploration rate, S:current state, episode: the number of episode

Output: Selected Action

```
1
2 Function SELECT-ACTION(Q, S, ϵ):
3     n ← uniform random number between 0 and 1
4     if n < ϵ then
5         A ←random action from the actions set
6     else
7         A ← maxQ(S, .)            // With the . symbol, we refer to all the actions for the state S
8     ϵ ← UpdateExplorationRate(ϵ, episode)
9     return A
```

Η μεταβλητή ϵ είναι το exploration threshold με το οποίο ο πράκτορας παίρνει την απόφαση για τη χρήση της εξερεύνησης ή της εκμετάλλευσης, που θα εκτελέσει σε κάθε βήμα. Η μεταβλητή αυτή θα ξεκινάει με τιμή 1, καθώς στα πρώτα επεισόδια στόχος του πράκτορα είναι η εξερεύνηση και με το πέρασμα των επεισοδίων θα μειώνεται η τιμή αυτή έτσι ώστε να μεταβούμε ομαλά στην διαδικασία εκμετάλλευσης της γνώσης.

Algorithm 3: Αλγόριθμος Ενημέρωσης Μεταβλητής Εξερεύνησης

Data: ϵ: exploration rate, episode: the number of episode, EDR:exploration decay rate, MinExpRate: minimum exploration rate, MaxExpRate: maximum exploration rate

Output: Updated exploration rate

```
1
2 Function updateExplorationRate(ϵ, episode):
3     ϵ = MinExpRate + [(MaxExpRate − MinExpRate) ∗ e^(−EDR∗episode)]
4     return ϵ
```

3.2 Monte Carlo and Temporal Difference Hybrid

Η υβριδική μέθοδος Monte Carlo & Temporal Difference δημιουργήθηκε για να συνδυάσει τα πλεονεκτήματα και να μειώσει παράλληλα την επίδραση των μειονεκτήματων των δυο αυτών μεθόδων. Για την κατανόηση της ανάγκης χρήσης αυτής της μεθόδου αναφέρουμε σύντομα παρακάτω τις δυο μεθόδους με τα αντίστοιχα μειονεκτήματα και πλεονεκτήματα.

3.2.1 Monte Carlo

Για την ενισχυτική μάθηση Monte Carlo χρησιμοποιούμε ένα σύνολο από ολοκληρωμένα επεισόδια για να προσεγγίσουμε την τιμή $Q(S, .)$. Ένα επεισόδιο χαρακτηρίζεται ως το συναλικό ταξίδι του πράκτορα από την αρχική κατάσταση μέχρι και την τελική κατάσταση. Η προσέγγιση αυτή δουλεύει βέλτιστα όταν έχουμε μια καλά ορισμένη τελική κατάσταση, γεγονός που δεν ισχύει στην περίπτωση μας. Στο πρόβλημα αυτό ο πράκτορας μπορεί να βγει από οποιδήποτε σημείο του πλέγματος τερματίζοντας έτσι και την κίνηση του. Παρακάτω ορίζουμε την Q-Function ως το αναμενόμενο άθροισμα ανταμοιβών ξεκινώντας από μια δεδομένη κατάσταση και εκτελώντας μια συγκεκριμένη ενέργεια.

$$Q(s, a) = E[R(S_t) + \gamma R(S_t + 1) + \gamma^2 R(S_t + 2) + ...|St = s, At = a]$$

Για να προσεγγίσουμε την μέθοδο Monte Carlo θα αντικαταστήσουμε την αναμενόμενη τιμή E με την μέση τιμή του αθροίσματος των ανταμοιβών από όλα τα επεισοδία. Αυτές οι αθροιστικές ανταμοιβές θεωρούνται συνολικά ως μια επιστροφή, η οποία συμβολίζεται με G_t και είναι η εξής:

$$G_t = R_t + \gamma R_{t+1} + ... + \gamma^{T-1} R_T$$

Δεδομένου ότι γνωρίζουμε το άθροισμα ανταμοιβών, μπορούμε να ορίσουμε την Q-value της μεθόδου και με λίγο αναδιάταξη του αθροίσματος, καταλήγουμε σε έναν ορισμό για το μέσο που μπορούμε να ενημερώσουμε σταδιακά χρησιμοποιώντας τις τρέχουσες τιμές του state-action pair σε συνδυασμό με τις συνολικές κατά προσέγγιση τιμές state-action pairαπό όλα τα προηούμενα επεισόδια:

$$Q(S, A) = \mu_k = \frac{1}{k} \sum_{i=1}^{k} G_i(S, A) \Rightarrow$$

$$\mu_k = \frac{1}{k}[G^k(S, A) \sum_{i=1}^{k-1} G_i(S, A)] \Rightarrow$$

$$\mu_k = \frac{1}{k}[G_k(S, A) + (k - 1)\mu_{k-1}] \Rightarrow$$

$$\mu_k = \mu_{k-1} + \frac{1}{k}[G_k(S, A) - \mu_{k-1}]$$

Έτσι καταλήγουμε στον παρακάτω τύπο της Q-Learning function όπου χρησιμοποιούμε την Q-value και όχι την προσέγγιση μ_k:

$$Q(S, A) = Q(S, A) + a(G_k(S, A) - Q(S, A))$$

Παραπάνω έχουμε αναφέρει ότι ενημερώνουμε την q-value του Q-table με βάση την εμπειρία που έχει αποκτήσει ο πράκτορας από προηγούμενα επεισόδια, καθώς λαμβάνουμε τον μέσο όρο τον τιμών όλων των επεισοδίων. Το πλεονέκτημα αυτού του τρόπου είναι ότι μειώνουμε την μεροληψία του πράκτορα στο ελάχιστο, διότι μια καινούργια τιμή για την κατάσταση θα έχει μικρή τελική επίδραση. Ωστόσο το μειονέκτημα είναι ότι αναφερόμαστε σε ένα άθροισμα από κινήσεις που επηρεάζονται άμεσα από την στοχαστικότητα του περιβάλλοντος.

3.2.2 Temporal Difference

Αρχικά πρέπει να ορίσουμε τον Q-table για τον οποίο θα ορίσουμε αυθαίρετα τις τιμές για κάθε state-action pair με μηδενικές τιμές, αν και θα μπορούσαμε να βάλουμε μια μικρή θετική τιμή για να ενθαρύνουμε τον πράκτορα να εξερευνήσει το περιβάλλον, γνωστό ως αισιοδοξία μπροστά στην αβεβαιότητα. Στην προσέγγιση του Monte Carlo που συζητήθηκε προηγουμένως, προσομοιώσαμε πλήρη επεισόδια και στη συνέχεια υπολογίζαμε εκ νέου την τιμή κάθε state-action pair που εκτελέστηκε κατά τη διάρκεια του επεισοδίου.

Αντιθέτως με την Temporal Difference προσέγγιση ενισχυτικής μάθησης έχουμε την δυνατότητα να ενημερώσουμε την q-value κατά τη διάρκεια του επεισοδίου. Όπως υποδηλώνει το όνομα, ενημερώνουμε τη συνάρτηση τιμής μας εξετάζοντας τη διαφορά της εκτιμώμενης τιμής ενός state-action pair σε δύο διαφορετικά χρονικά σημεία (χρονική διαφορά). Πιο συγκεκριμένα, θα συγκρίνουμε μια τρέχουσα εκτίμηση για την αξία ενός state-action pair με μια πιο ενημερωμένη μελλοντική εκτίμηση αφού ο πράκτορας αποκτήσει περισσότερη εμπειρία στο περιβάλλον.

Για παράδειγμα, όταν φτάσουμε σε κάποια κατάσταση, θα εξετάσουμε την ανταμοιβή που συλλέγει αμέσως ο πράκτορας συν τις αναμενόμενες μελλοντικές αποδόσεις κατά την είσοδο σε μια επόμενη κατάσταση S', μέσω της εκτέλεσης κάποιας ενέργειας και θα το συγκρίνουμε με την προηγούμενη εκτίμησή μας για την αξία του state-action ζεύγους.

$$Q(S, A) = Q(s, a) + \alpha[(R_t + \gamma Q(S', A')) - Q(S, A)]$$

Η έκφραση $R_t + \gamma Q(S', A')$ είναι γνωστή ως TD target και αντιπροσωπεύει μια καλύτερη προσέγγιση του $Q(S, A)$ δεδομένου του γεγονότος ότι αντικαθιστά την εκτιμώμενη ανταμοιβή που λαμβάνεται από την υπάρχουσα κατάσταση S με την πραγματική ανταμοιβή. Σε αυτήν την περίπτωση, το TD target είναι ένα βήμα μπροστά από την τιμή του ζεύγους κατάστασης-δράσης που ενημερώνουμε, αλλά μπορεί να επεκταθεί ώστε να φαίνεται ν βήματα μπροστά κατά την προσέγγιση των $Q(S, A)$.

Η διαφόρα του TD target στην χρονική στιγμή t με την q-value για το τρέχον state-action ζεύγους $Q(S, A)$ ονομάζεται TD error , συμβολίζεται με δ_t και έχει σημασία για τον πράκτορα καθώς στόχος του είναι να επιλέξει την ενέργεια που ελαχιστοποιεί το TD error .

$$\delta_t = [(R_t + \gamma Q(S', A')) - Q(S, A)]$$

Σε αντίθεση με την προσέγγιση Monte Carlo, όταν ενημερώνουμε την state-action συνάρτηση με την μέθοδο TD χρησιμοποιούμε ένα μόνο βήμα του τρέχοντος επεισοδίου (S, A, S') σε συνδυασμό με την υπάρχουσα εμπειρία $Q(S, A)$ για την τιμή αυτή για να υπολογίσουμε την νέα τιμή της state-action συνάρτηση. Με αυτόν τον τρόπο δεν απαιτείται η ολοκλήρωση ενός επεισοδίου και η ύπαρξη μιας συγκεκριμένης τελικής κατάστασης, απαιτήσεις που δεν ικανοποιούνται από το περιβάλλον της εφαρμογής. Φυσικά επιλέγουμε την ενέργεια που θα επιλέξουμε με βάση τις μελλοντικές ανταμοιβές από εκείνο το σημείο και ύστερα, το οποίο προσθέτει μεροληψία στην απόφαση του πράκτορα. Συνεπώς μπορούμε να καταλάβουμε ότι η κάθε μέθοδος περιλαμβάνει κάποια θετικά και αρνητικά στοιχεία:

Monte Carlo

1. Απουσία Μεροληψίας
2. Υψηλή Διακύμναση τιμών
3. Καλή απόδοση μόνο σε συγκεκριμένα περιβάλλοντα

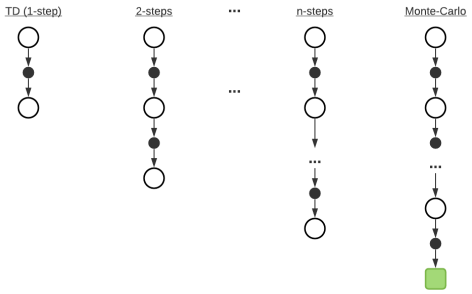
Temporal Difference

1. Παρουσία Μεροληψίας
2. Χαμηλή Διακύμναση τιμών
3. Καλή απόδοση σε μη-συγκεκριμένα περιβάλλοντα

3.2.3 Monte Carlo and Temporal Difference Hybrid

Προκειμένου να πετύχουμε ένα καλύτερο αποτέλεσμα ενισχυτικής μάθησης χρησιμοποιούμε μια προσέγγιση που συνδυάζει τις δυο παραπάνω τεχνικές. Στόχος σε αυτή την περίπτωση είναι να επεκτείνουμε την TD έκφραση για να περιλαμβάνει παραπάνω από ένα βήματα στην διαδικασία ενημέρωσης του TD target . Μέχρι αυτό το σημείο γνωρίζουμε ότι το TD target αντιπροσωπεύει μια πληροφορημένη προσέγγιση της q-value για ένα state-action ζεύγος μέσω της χρήσης της τρέχουσας ανταμοιβής και της αναμενόμενης ανταμοιβής από την τρέχουσα κατάσταση και μετά.

Για $n = 1$ παρατηρούμε την ανταμοιβή για το επόμενο βήμα και την αντίστοιχη εκτιμώμενη ανταμοιβή από εκείνο το σημείο και έπειτα, και το άθροισμα αυτών των ανταμοιβών αυτών εκχωρείται ως το TD target της αρχικής κατάστασης. Ομοίως για $n = 2$ θα πάρουμε τις αντίστοιχες ανταμοιβές για τα επόμενα δυο βήματα και θα τα εκχωρήσουμε στο TD target της αρχικής κατάστασης.



Εικόνα 10: Αριθμός Βημάτων ανά Μέθοδο Ενισχυτικής Μάθησης

Γενικά γνωρίζουμε ότι το TD target για το $TD(n)$ βρίσκεται σύμφωνα με την παρακάτω συνάρτηση:

$$R_t + \gamma R_{t+1} + ... + \gamma^{n-1} R_{t+n-1} + \gamma^n Q(S', A')$$

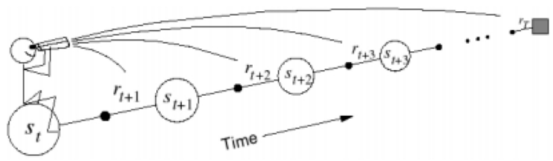
Αν είχαμε την δυνατότητα να φτάναμε για $n = \infty$ βήματα θα φτάναμε στην Monte Carlo προσέγγιση που αναφέραμε παραπάνω. Όσο αυξάνουμε το n περιλαμβάνουμε περισσότερες παρατηρήσεις του περιβάλλοντος, μειώνοντας την μεροληψία στην εκτίμηση μας, ωστόσο επειδή κάθε νέα κατάσταση είναι αποτέλεσμα μιας στοχαστικής διαδικασίας η διακύμανση αυξάνεται με μεγάλο ρυθμό. Σε ένα στοχαστικό περιβάλλον θα ήταν η βέλτιστη τεχνική να εξετάσουμε όλες τις πιθανές τιμές n για να πετύχουμε την βέλτιστη απόδοση. Η προσέγγιση αυτή αναφέρεται ως προσέγγιση $TD(\lambda)$ που συνδυάζει όλες τις αποδόσεις $n - step$ ως γεωμετρικό σταθμισμένο άθροισμα, που μειώνεται κατά συντελεστή λ, όπως φαίνεται στο παρακάτω διάγραμμα. Επιλέγουμε συγκεκριμένα να χρησιμοποιήσουμε ένα γεωμετρικό άθροισμα για λόγους υπολογιστικής αποτελεσματικότητας.

Λαμβάνοντας ως τιμή $\lambda = 0$, μόνο η απόδοση ενός σταδίου έχει μη μηδενικό βάρος και επομένως αυτό ισοδυναμεί με τη μέθοδο $TD(0)$. Αντίστοιχα λαμβάνοντας ως τιμή το $\lambda = 1$, μόνο η πλήρης ακολουθία επιστροφής έχει μη μηδενικό βάρος και έτσι αυτό ισοδυναμεί με τη μέθοδο του Monte Carlo.

Αυτή την φορά ο ορισμός τόσο για το $TD(\lambda)$, G_t^λ προσαρμόζονται έτσι ώστε να λαμβάνουν υπόψη το άθροισμα των επιστροφών για n βήματα.

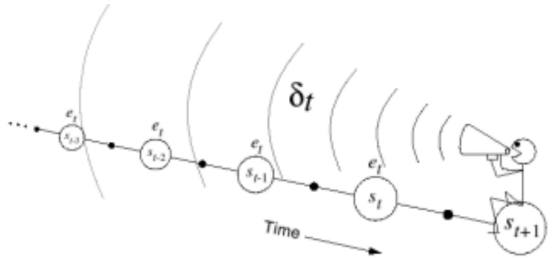
$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)}$$

όπου $G_t^{(n)}$ αναπαριστά την επιστροφή για το $n-step$. Μέχρι αυτό το σημείο έχουμε αναλύσει την forward-view προσέγγιση όπου αναθέτουμε q-value σε ένα state-action ζεύγος με βάση τις επιστροφές από τα επόμενα βήματα. Ωστόσο για να εφαρμόσουμε την forward-view προσέγγιση πρέπει να περιμένουμε μέχρι το τέλος του επεισοδίου για κάποιες περιπτώσεις.



Εικόνα 11: Forward-view Προσέγγιση[6]

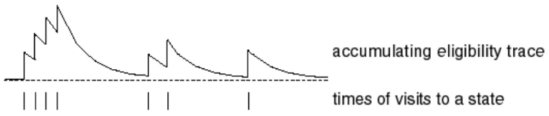
Ευτυχώς, μπορούμε να αναδιαμορφώσουμε αυτήν την προοπτική για να δούμε μια backward-view που θα μας επέτρεπε να ενημερώσουμε τις καταστάσεις που είχαμε επισκεφθεί προηγουμένως μετά από επαρκή εξερεύνηση. Για παράδειγμα, στην περίπτωση της μάθησης $TD(n)$, κάθε βήμα που κάνουμε σε μια τροχιά μας επιτρέπει να ενημερώνουμε την τιμή της κατάστασης n βήματα πριν από την τρέχουσα κατάστασή μας.



Εικόνα 12: Backward-view Προσέγγιση[7]

Με αυτόν τον τρόπο, μπορούμε να στείλουμε πίσω τις μελλοντικές ανταμοιβές των ενεργειών σε προηγούμενες καταστάσεις. Ωστόσο, αυτή η προσέγγιση εισάγει το Πρόβλημα Εκχώρησης Πίστωσης-Credit Assignment Problem (CAP). Όταν εξετάζουμε ενημερώσεις για n-βήματα, είναι δύσκολο να διαπιστώσουμε τις προηγούμενες ενέργειες που είχαν επιρροή στην ανταμοιβή που συσσωρεύτηκε στο τρέχον χρονικό βήμα. Για να λύσουμε το CAP πρόβλημα θα μπορούσαμε να χρησιμοποιήσουμε μια ευρετική συνάρτηση συχνότητας με την οποία θα αναθέταμε περισσότερο βάρος στις καταστάσεις του περιβάλλοντος που επισκέπτεται ο πράκτορας συχνότερα. Σε αυτό το σημείο υπόθετουμε ότι τα τρέχοντα αποτελέσματα είναι πιθανότερο να οφείλονται στις καταστάσεις

που πέρασε ο πράκτορας πιο πρόσφατα. Έτσι προκειμένου να αποφύγουμε μια τέτοια υπόθεση χρησιμοποιούμε ένα ίχνος καταλληλότητας το οποίο συνδυάζει και τις δύο αυτές ευρετικές, αυξάνοντας το credit που εκχωρείται σε μια κατάσταση κατά ένα σταθερό ποσό κάθε φορά που επισκέπτεται την κατάσταση, και με σταθερό μικρό ρυθμό μειώνει αυτό το credit με την πάροδο του χρόνου, όπως φαίνεται στην παρακάτω φωτογραφία.



Εικόνα 13: Elegibility Trace[8]

Καθώς ο πράκτορας μας εξερευνά το περιβάλλον, ενημερώνουμε τη state-action function προς την κατεύθυνση που ελαχιστοποιεί το TD error, κλιμακωμένο από το ίχνος καταλληλότητας για κάθε ενημερωμένη κατάσταση. Αυτό μας επιτρέπει να εστιάσουμε τις ενημερώσεις μας στις καταστάσεις που πιστεύουμε ότι οδήγησαν σε μελλοντικές αποδόσεις. Σε αυτή τη μέθοδο εκτός από τον Q-table έχουμε και τον αντίστοιχο Πίνακα Καταλληλότητας, το οποίο αρχικοποιούμε με μηδενικές τιμές.

$$E_0(S, A) = 0$$

Table 2: Elegibility-table

State-Action	((-1,0),1)	((-1,0),2)	((-1,0),3)	((-1,1),1)	...	((-1,-1),3)
(0,1)	0	0	0	0	0	0
(0,2)	0	0	0	0	0	0
(0,3)	0	0	0	0	0	0
(0,4)	0	0	0	0	0	0
(0,5)	0	0	0	0	0	0
....	0	0	0	0	0	0
(19,19)	0	0	0	0	0	0

Ομοίως με τον Q-table στη πρώτη γραμμή του πίνακα μπορούμε να δούμε την κωδικοποίηση των ενεργειών ενός πράκτορα, όπου η πρώτη παρένθεση αφορά την κατεύθυνση της κίνησης του πράκτορα και η τιμή δίπλα αφορά την απόσταση που θα καλύψει ο πράκτορας στην κίνηση του. Ξανά η πρώτη τιμή αφορά τις στήλες του πλέγματος ή την κατεύθυνση προς τα Ανατολικά ή Δυτικά και η δεύτερη τιμή αφορά τις γραμμές και τις κατευθύνσεις Βόρεια ή Νότια.

Σε κάθε χρονικό βήμα, όλα τα elegibility-traces για κάθε state-action ζεύγος μειώνονται με έναν συντελεστή γ, ενώ αυξάνουμε την τιμή elegibility κάθε φορά που ο πράκτορας εκτελεί την ίδια ενέργεια για την ίδια κατάσταση.

$$E_t(S, A) = \gamma\lambda E_{t-1}(S, A) + 1$$

Αυτό μας επιτρέπει να πραγματοποιούμε ενημερώσεις σε ζεύγη state-action , σε κάθε χρονικό βήμα, ανάλογα με την καταλληλότητά τους, $E_t(S, A)$ και το TD error και δ_t.

$$\delta_t = R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)$$

$$Q(S, A) = Q(S, A) + \alpha\delta_t E_t(S, A)$$

Ενώ το TD error συγκρίνει μόνο ένα βήμα προς την τρέχουσα εκτίμησή μας, έχουμε τη δυνατότητα να στείλουμε πίσω σφάλμα σε οποιοδήποτε state-action ζεύγος με μη μηδενική elegibility.

4 Αποτελέσματα Πειραματικών Μελετών

Στόχος της εργασίας είναι ο Υπολογισμός των Βέλτιστων Τροχιών-Trajectory Optimization για την λύση του πολύπλοκου προβήματος ATM . Όπως έχουμε ήδη αναφέρει η εργασία περιλαμβάνει δυο κύριες περιπτώσεις, το μονοπρακτορικό και το πολυπρακτορικό περιβάλλον, όπου χρησιμοποιούμε δυο διαφορετικές μεθόδους ενισχυτικής μάθησης τον E-Greedy Strategy και Monte-Carlo-Temporal Difference Hybrid.

4.1 Κριτήρια Αξιολόγησης

Για την αξιολόγηση της αποδοτικότητας και τις αποτελεσματικότητας των πειραματικών μελετών παρέχουμε τις εξής πληροφορίες για την απόδοση των μεθόδων σε κάθε περίπτωση:

1. **Καμπύλες Μάθησης**: Οι καμπύλες αυτές δείχνουν τις ανταμοιβές που λαμβάνουν ο πράκτορας ή οι πράκτορες κατά τη διάρκεια των επεισοδίων όσο μαθαίνουν και εφαρμόζουν την πολιτική τους για το περιβάλλον. Αρχικά οι ανταμοιβές θα είναι αρνητικές, καθώς ο πράκτορας θα εξερευνεί το περιβάλλον και αργότερα θα αυξάνονται σταδιακά μέχρι να τείνουν να σταθεροποιηθούν. Επομένως η ταχύτητα επίτευξης αυτού του σημείου (σημείο σταθεροποίησης) και η τιμή του θα μας δώσουν τόσο την αποτελεσματικότητα όσο και την αποδοτικότητα της λύσης. Σε περίπτωση που μια μέθοδος δεν επιτύχει κάποια επιθυμητή λύση τότε η πολιτική που έχει σχηματιστεί είναι μια μη-βέλτιστη πολιτική.

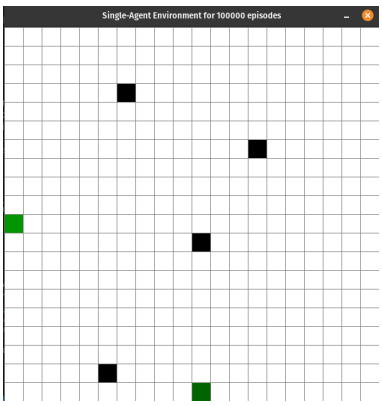
2. **Heatmap**: είναι ένας χάρτης του περιβάλλοντος, ο οποίος για κάθε κόμβου του πλέγματος έχει μια τιμή. Η τιμή αυτή αντιπροσωπεύει πόσες φορές έχει βρεθεί σε αυτό τον κόμβο ο πράκτορας εκτελώντας μια ενέργεια με βάση την πολιτική, που έχει αναπτύξει στο στάδιο της εξερεύνησης.

Τα αποτελέσματα παρακάτω είναι οι μέσοι όροι αποτελεσμάτων 10 ανεξάρτητων πειραμάτων για κάθε μέθοδο και περίπτωση.

4.2 Μονοπρακτορικό Περιβάλλον

Για να έχουμε μια δίκαιη σύγκριση μεταξύ των μεθόδων τα σημεία εκκίνησης και τερματισμού θα παραμένουν σταθερά για κάθε πείραμα σε κάθε μια από τις παρακάτω περιπτώσεις μονοπρακτορικού περιβάλλοντος. Πιο συγκεκριμένα το σημείο εκκίνησης είναι το $(0, 10)$ και το σημείο τερματισμού είναι το $(10, 19)$, λόγω της χρήσης της βιβλιοθήκης pygame οι στήλες και οι γραμμές δίνοται ως εξής (y, x). Παράλληλα θα έχουμε τρεις περιπτώσεις όσον αφορά την συχνότητα κίνησης των εμποδίων στο περιβάλλον.

1. Τα εμπόδια θα είναι σταθερά
2. Τα εμπόδια θα μετακινούνται κάθε 1000 επεισόδια
3. Τα εμπόδια θα μετακινούνται κάθε 10000 επεισόδια



Εικόνα 14: Μονοπρακτορικό Περιβάλλον Πειραματικών Μελετών

Για την εκτέλεση των πειραματικών μελετών χρησιμοποιήσαμε τις παρακάτω μεταβλητές μάθησης:

1. Ρυθμός μάθησης $\alpha = 0.001$
2. Παράγοντας Προεξόφλησης $\gamma = 1$

4.2.1 Σταθερά Εμπόδια

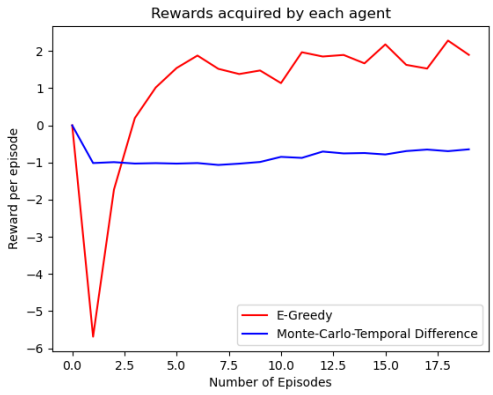
Σύμφωνα με τον τίτλο αυτής της ενότητας τα εμπόδια σε αυτή την περίπτωση θα είναι σταθερά για κάθε πείραμα και ο πράκτορας θα πρέπει να βρίσκει το στόχο, ενώ παράλληλα τηρεί αποστάσεις μεγαλύτερες των πέντε μιλίων ή πέντε κόμβων για μέγιστη απόδοση.

Καμπύλες Μάθησης

Οι καμπύλες μάθησης που παρουσιάζονται στην εικόνα 15 απεικονίζουν την ταχύτητα και την ακρίβεια με την οποία ο πράκτορας μαθαίνει το περιβάλλον με την χρήση της κάθε μεθόδου. Για την καλύτερη κατανόηση της εξέλιξης της διαδικασίας έχουμε ομαδοποιήσει τα αποτελέσματα ανά 5.000 αποτελέσματα, όπου η κόκκινη καμπύλη αφορά τη μέθοδο E-Greedy Strategy και η μπλέ καμπύλη αφορά τη μέθοδο Monte-Carlo-Temporal Difference Hybrid.

Αρχικά η μέθοδος E-Greedy Strategy λαμβάνει μια μεγάλη αρνητική ανταμοιβή καθώς εκτελεί την εξερεύνηση του περιβάλλοντος, αλλά βελτιώνει την ανταμοιβή του λίγο μετά τα 10.000 επεισόδια και συνεχίζει την σταδιακή βελτίωση μέχρι και τα 100.000 επεισόδια. Όπως φαίνεται ο πράκτορας φτάνει τον στόχο του με μια καλή συχνότητα, διότι στο μεγάλο περιβάλλον με στοχαστικότητα καταφέρνει να λαμβάνει θετική ανταμοιβή.

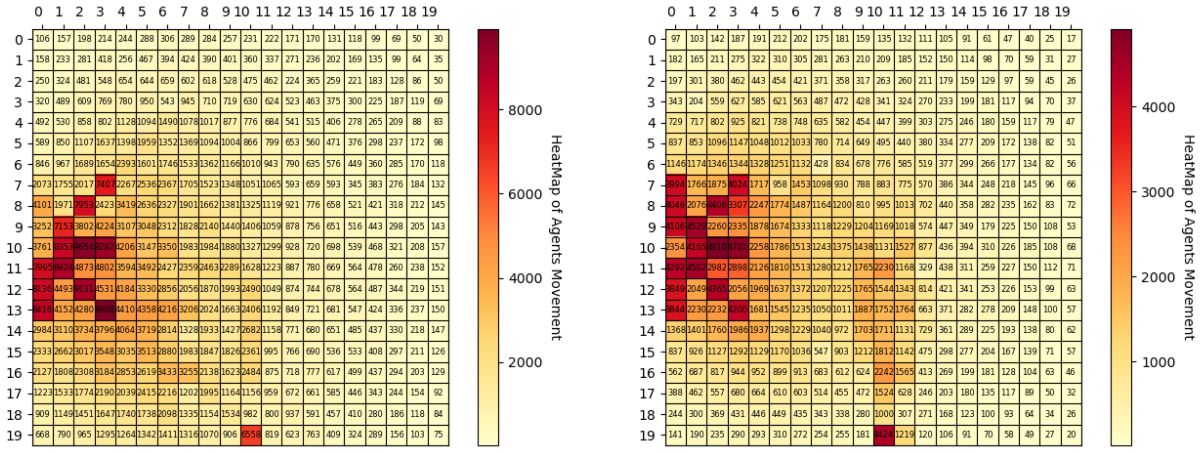
Αντίθετα με την χρήση της μεθόδου Monte-Carlo-Temporal Difference Hybrid ο πράκτορας εξερευνεί το περιβάλλον για 50.000 επεισόδια με μικρή επιτυχία κατά μέσο όρο στα 10 πειράματα, ενώ στην συνέχεια χρησιμοποιεί την εμπειρία που έχει συλλέξει για να πετύχει τον στόχο του προβλήματος. Παρατηρούμε μια μικρή βελτίωση, μετά τα 50.000 επεισόδια, αλλά δεν θεωρείται ως μια ουσιαστική βελτίωση που μαρτυρά την λύση του προβλήματος. Στα επτά από τα δέκα πειράματα η γρηγορότερη μετάδοση της γνώσης είχε αρνητικά αποτελέσματα για δύο λόγους. Αρχικά ο πράκτορας μπορεί να βγεί από το περιβάλλον από οποιοδήποτε σημείο στα όρια του περιβάλλοντος λαμβάνοντας έτσι μια αρνητική ανταμοιβή, η οποία θα μεταδοθεί γρηγορότερα σε περισσότερες καταστάσεις δημιουργώντας μια πολιτική που αποθαρρύνει τον πράκτορα να βρεί τον κόμβο στόχο. Επίσης σε κάποιες περιπτώσεις τα εμπόδια βρίσκονταν κοντά στον κόμβο στόχο, δημιουργώντας διπλό πρόβλημα όπου ο πράκτορας δυσκλευόταν να φτάσει στον στόχο ενώ παράλληλα λαμβάνε μεγάλες αρνητικές ανταμοιβές. Στα τρία από τα δέκα πειράματα είχαμε θετικά αποτελέσματα, τα οποία είχαν πολύ θετικά αποτελέσματα. Πιο συγκεκριμένα ο πράκτορας κατέφερε να προσεγγίσει τον κόμβο στόχο 12.000 φορές κατά μέσο όρο σε αυτά τα τρία πειράματα, ξεπερνώντας τα καλύτερα αποτελέσματα της μεθόδου E-Greedy Strategy, στα οποία ο πράκτορας έφτασε στον στόχο περίπου 7.000 φορές.



Εικόνα 15: Ανταμοιβές πράκτορα ανάλογα με τη μέθοδο ενισχυτικής μάθησης

Heatmap

Τα παρακάτω heatmaps παρέχουν χρήσιμη πληροφορία, η οποία μπορεί να εξηγήσει καλύτερα τα αποτελέσματα των καμπυλών μάθησης. Ξεκινώντας με τη μέθοδο E-Greedy Strategy καταλαβαίνουμε ότι ο πράκτορας έχει βρει την τοποθεσία του κόμβου στόχου και λαμβάνει πολλές διαφορετικές διαδρομές για να φτάσει. Το γεγονός ότι δεν έχουμε μια διαδρομή προς τον στόχο εξηγείται από την μεταβλητότητα των θέσεων των εμποδίων ανα πείραμα και επιβεβαιώνει την αποδοτικότητα του πράκτορα για τη μάθηση του εκάστοτε περιβάλλοντος. Αντίθετα η χρήση της μεθόδου Monte-Carlo-Temporal Difference Hybrid αποτυπώνει μια διαφορετική κατάσταση για την διαδρομή του πράκτορα προς τον κόμβο στόχο. Πιο συγκεκριμένα φαίνεται ότι ο πράκτορας χρησιμοποιεί ένα κύριο μονοπάτι για να επιτύχει το ζητούμενο, γεγονός που μας προβληματίζει για την αποδοτικότητα του πράκτορα σε διάφορα περιβάλλοντα. Είναι γεγονός ότι η επιτυχία του πράκτορα οφείλεται κυρίως σε ένα υποσύνολο πειραμάτων όπου τα εμπόδια βρίσκονταν μακριά από την κατάσταση στόχου και ήταν εύκολο για τον πράκτορα να βρει την λύση του προβλήματος. Συνεπώς τα θετικά αυτά πειράματα επηρέασαν σημαντικά τόσο τα heatmaps όσο και τις καμπύλες μάθησης.



(α′) Heatmap για τον πράκτορα με χρήση της E-Greedy Strategy μεθόδου

(β′) Heatmap για τον πράκτορα με χρήση της Monte-Carlo-Temporal Difference Hybrid μεθόδου

Εικόνα 16: Heatmap πρακτόρων ανάλογα με τη μέθοδο ενισχυτικής μάθησης

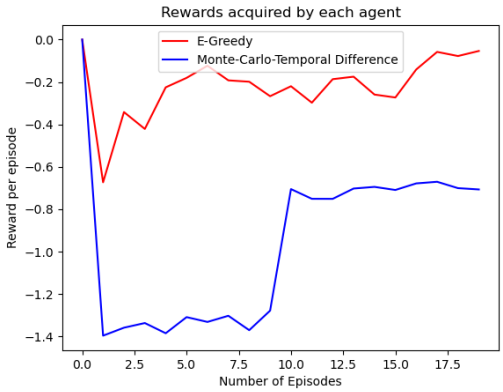
4.2.2 Εμπόδια που μετακινούνται ανά 1.000 επεισόδια

Σε αυτή την περίπτωση τα εμπόδια στο περιβάλλον αλλάζουν κάθε 1000 επεισόδια, όπου ο πράκτορας δεν διαθέτει αρκετό χρόνο για να εξερευνήσει επρακώς το περιβάλλον. Φυσικά υπάρχει το πλεονέκτημα ότι η πιθανότητα τα εμπόδια να βρίσκονται κοντά στον στόχο για μεγάλο διάστημα της διαδικασίας εξερεύνησης μειώνεται σημαντικά. Ομοίως ο στόχος του πράκτορα είναι να φτάσει στο σημείο τερματισμού προσπαθώντας να αποφεύγει παράλλληλα τα εμπόδια.

Καμπύλες Μάθησης

Ομοίως στην εικόνα 17 βρίσκουμε τα αποτελέσματα των πειραματικών μελετών για αυτή την περίπτωση. Για την καλύτερη κατανόηση της εξέλιξης της διαδικασίας έχουμε ομαδοποιήσει τα αποτελέσματα ανά 5.000 αποτελέσματα, όπου η κόκκινη καμπύλη αφορά τη μέθοδο E-Greedy Strategy και η μπλέ καμπύλη αφορά τη μέθοδο Monte-Carlo-Temporal Difference Hybrid.

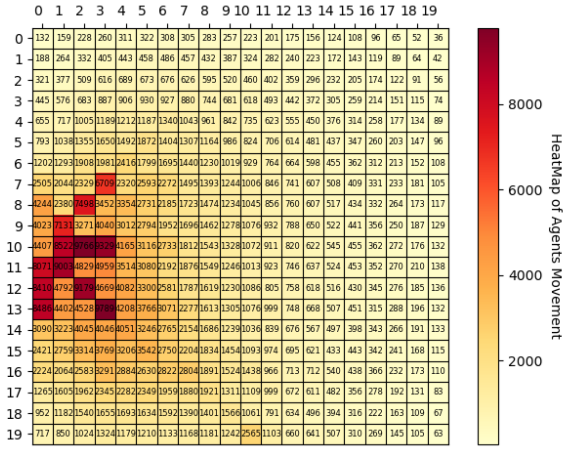
Αυτή τη φορά η μέθοδος E-Greedy Strategy λαμβάνει μια μικρότερη αρνητική ανταμοιβή καθώς εκτελεί την εξερεύνηση του περιβάλλοντος, αλλά βελτιώνει την ανταμοιβή του λίγο μετά τα 10.000 επεισόδια και συνεχίζει την σταδιακή βελτίωση μέχρι και τα 100.000 επεισόδια. Όπως φαίνεται ο πράκτορας φτάνει τον στόχο του με μια μέτρια συχνότητα, διότι στο μεγάλο περιβάλλον με στοχαστικότητα καταφέρνει να λαμβάνει οριακά θετική ανταμοιβή. Αντίθετα με την χρήση της μεθόδου Monte-Carlo-Temporal Difference Hybrid ο πράκτορας εξερευνεί το περιβάλλον για 50.000 επεισόδια. Στις συγκεκριμένες μελέτες η χρήση της μεθόδου Monte-Carlo-Temporal Difference Hybrid επιβεβαιώνει την διαφορά μεταξύ της διαδικασίας εξερεύνησης και εκμετάλλευσης. Παρατηρούμε ότι μετά τα 50.000 επεισόδια η απόδοση αυξάνεται απότομα αλλά όχι αρκετά έτσι ώστε να έχουμε θετικές ανταμοιβές, το οποίο υποδηλώνει ότι ο πράκτορας έχει μια μη-βέλτιστη πολιτική.



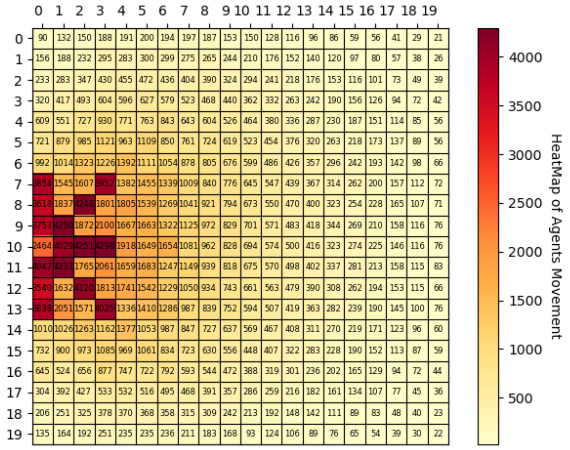
Εικόνα 17: Ανταμοιβές πράκτορα ανάλογα με τη μέθοδο ενισχυτικής μάθησης

Heatmap

Ομοίως τα παρακάτω heatmaps παρέχουν χρήσιμη πληροφορία, η οποία μπορεί να εξηγήσει καλύτερα τα αποτελέσματα των καμπυλών μάθησης. Ξεκινώντας με τη μέθοδο E-Greedy Strategy καταλαβαίνουμε ότι ο πράκτορας έχει βρει την τοποθεσία του κόμβου στόχου, ωστόσο με μικρότερη συχνότητα. Αντίθετα η χρήση της μεθόδου Monte-Carlo-Temporal Difference Hybrid αποτυπώνει μια διαφορετική κατάσταση όπου ο πράκτορας δεν έχει εντοπίσει ακριβώς την τοποθεσία του κόμβου στόχου και προτιμεί να βγει από το πλέγμα για να μην λάβει μεγάλη αρνητική τιμή. Το γεγονός αυτό επιβεβαιώνει την καμπύλη μάθησης, όπου παρατηρούμε μια βελτίωση, η οποία επιφέρει μικρότερες αρνητικές ανταμοιβές αλλά δεν επιφέρει θετικά αποτελέσματα.



(α′) Heatmap για τον πράκτορα με την χρήση της E-Greedy Strategy μεθόδου



(β′) Heatmap για τον πράκτορα με χρήση της Monte-Carlo-Temporal Difference Hybrid μεθόδου

Εικόνα 18: Heatmap πρακτόρων ανάλογα με τη μέδοδο ενισχυτικής μάθησης

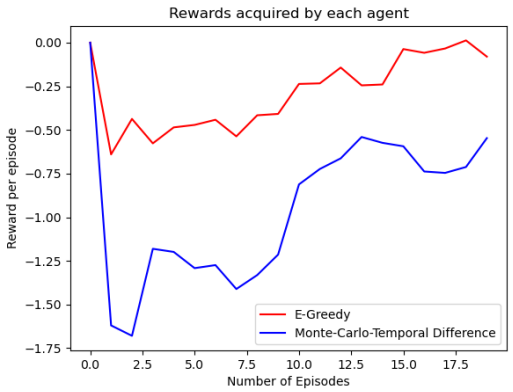
4.2.3 Εμπόδια που μετακινούνται ανά 10.000 επεισόδια

Στην τελευταία περίπτωση μονοπρακτορικού περιβάλλοντος τα εμπόδια θα αλλάζουν θέση κάθε 10.000 επεισόδια, όπου ο πράκτορας αντιμετωπίζει το ίδιο πρόβλημα περιορισμένου χρόνου για να μάθει αποτελεσματικά τα περιβάλλον. Επιπλέον η πιθανότητα ύπαρξης εμποδίων κοντά στο σημείο τερματισμού αυξάνεται καθώς έχουμε κίνηση των εμποδίων και τα εμπόδια παραμένουν σε αυτά τα σημεία για περισσότερα επεισόδια.

Καμπύλες Μάθησης

Στην εικόνα 19 έχουμε τις καμπύλες μάθησης για τις πειραματικές μελέτες για αυτή την περίπτωση. Για την καλύτερη κατανόηση της εξέλιξης της διαδικασίας έχουμε ομαδοποιήσει τα αποτελέσματα ανά 5.000 αποτελέσματα, όπου η κόκκινη καμπύλη αφορά τη μέθοδο E-Greedy Strategy και η μπλέ καμπύλη αφορά τη μέθοδο Monte-Carlo-Temporal Difference Hybrid.

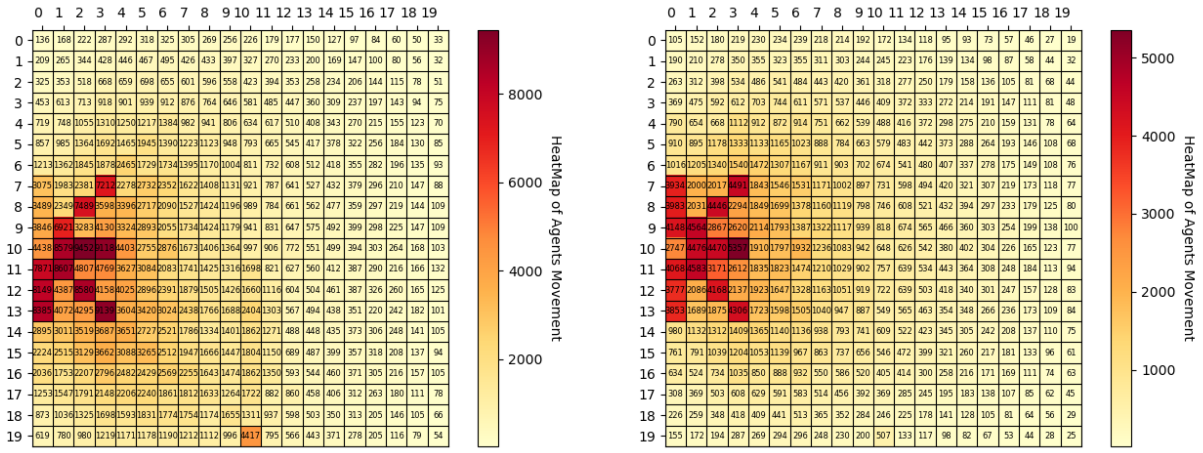
Στην τελευταία περίπτωση η μέθοδος E-Greedy Strategy λαμβάνει μια μικρή αρνητική ανταμοιβή αρχικά κατά τη διάρκεια της εξερεύνηση του περιβάλλοντος, αλλά βελτιώνει την ανταμοιβή του λίγο μετά τα 8.000 επεισόδια και συνεχίζει την σταδιακή βελτίωση μέχρι και τα 100.000 επεισόδια. Όπως φαίνεται ο πράκτορας φτάνει τον στόχο του με μια καλή συχνότητα, το οποίο αποδεικνύει την λύση του προβήματος από τον πράκτορα. Αντίθετα με την χρήση της μεθόδου Monte-Carlo-Temporal Difference Hybrid ο πράκτορας εξερευνεί το περιβάλλον για 50.000 επεισόδια. Ομοίως παρατηρούμε την διαφορά μεταξύ της διαδικασίας εξερεύνησης και εκμετάλλευσης. Παρατηρούμε ότι μετά τα 50.000 επεισόδια η απόδοση αυξάνεται απότομα αλλά όχι αρκετά έτσι ώστε να έχουμε θετικές ανταμοιβές, το οποίο υποδηλώνει ότι ο πράκτορας έχει μια μη-βέλτιστη πολιτική.



Εικόνα 19: Ανταμοιβές πράκτορα ανάλογα με τη μέθοδο ενισχυτικής μάθησης

Heatmap

Ξεκινώντας με τη μέθοδο E-Greedy Strategy καταλαβαίνουμε ότι ο πράκτορας έχει βρει με ακρίβεια την τοποθεσία του κόμβου στόχου. Αντίθετα η χρήση της μεθόδου Monte-Carlo-Temporal Difference Hybrid παρέχει μια διαφορετική κατάσταση όπου ο πράκτορας δεν έχει εντοπίσει ακριβώς την τοποθεσία του κόμβου στόχου και προτιμεί να βγει από το πλέγμα για να μην λάβει μεγάλη αρνητική τιμή. Το γεγονός αυτό επιβεβαιώνει την καμπύλη μάθησης, όπου παρατηρούμε μια βελτίωση η οποία επιφέρει μικρότερες αρνητικές ανταμοιβές αλλά δεν επιφέρει θετικά αποτελέσματα.



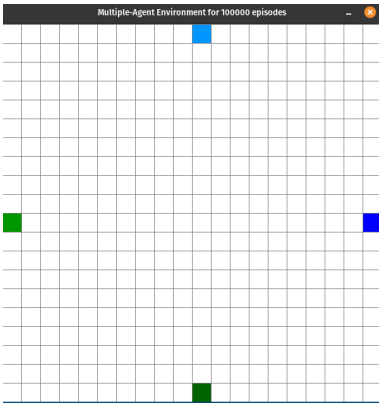
(α′) Heatmap για τον πράκτορα με χρήση της E-Greedy Strategy μεθόδου

(β′) Heatmap για τον πράκτορα με χρήση της Monte-Carlo-Temporal Difference Hybrid μεθόδου

Εικόνα 20: Heatmap πρακτόρων ανάλογα με τη μέθοδο ενισχυτικής μάθησης

4.3 Πολυπρακτορικό Περιβάλλον

Για να έχουμε μια δίκαιη σύγκριση μεταξύ των μεθόδων τα σημεία εκκίνησης και τερματισμού των δυο πρακτόρων θα παραμένουν σταθερά για κάθε πείραμα και θα έχουμε τον ίδιο αριθμό επεισοδίων για κάθε πείραμα. Πιο συγκεκριμένα τα σημεία εκκίνησης και τερματισμού για τον πρώτο πράκτορα είναι τα $(0, 10)$ και $(10, 19)$ αντίστοιχα, ενώ για τον δεύτερο είναι $(10, 0)$ και $(19, 10)$, γνωρίζοντας φυσικά ότι λόγω της χρήσης της βιβλιοθήκης pygame οι στήλες και οι γραμμές δίνοται ως εξής (y, x).



Εικόνα 21: Πολυπρακτορικό Περιβάλλον Πειραματικών Μελετών

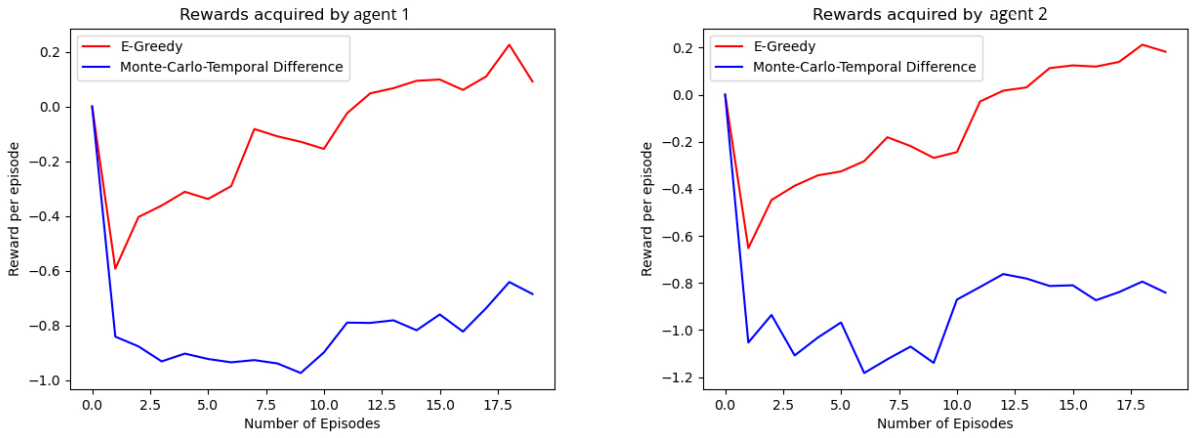
Για την εκτέλεση των πειραματικών μελετών χρησιμοποιήσαμε τις παρακάτω μεταβλητές μάθησης:

1. Ρυθμός μάθησης $\alpha = 0.001$
2. Παράγοντας Προεξόφλησης $\gamma = 1$
3. Παράμετρος $\lambda = 0.65$

Καμπύλες Μάθησης

Οι καμπύλες μάθησης που παρουσιάζονται στην εικόνα 22 απεικονίζουν πώς οι πράκτορες σε 100.000 επεισόδια καταφέρνουν να σχηματίσουν πολιτικές και να τις εφαρμόσουν αργότερα για την επίλυση του προβλήματος της βελτιστοποίησης τροχιών. Για την καλύτερη κατανόηση της εξέλιξης της διαδικασίας έχουμε ομαδοποιήσει τα αποτελέσματα ανά 5.000 αποτελέσματα, όπου η κόκκινη καμπύλη αφορά τη μέθοδο E-Greedy Strategy και η μπλέ καμπύλη αφορά τη μέθοδο Monte-Carlo-Temporal Difference Hybrid. Όπως μπορούμε να δούμε η μέθοδος E-Greedy Strategy βελτιώνει την ανταμοιβή του λίγο μετά τα 10.000 επεισόδια, ενώ η Monte-Carlo-Temporal Difference Hybrid έχει την ίδια βελτίωση μετά τα 50.000 επεισόδια. Η διαφορά αυτή είναι προφανής καθώς η μέθοδος E-Greedy Strategy επιτρέπει στον πράκτορα να κάνει χρήση της γνώσης του για το περιβάλλον αρκετά νωρίτερα, αφού η SELECT-ACTION συνάρτηση λειτουργεί με μια πιθανότητα εξερεύνησης, η οποία μειώνεται σταδιακά επιτρέποντας την συνύπαρξη των τυχαίων ενεργειών και των ενεργειών εκμετάλλευσης της γνώσης. Αντίθετα η μέθοδος Monte-Carlo-Temporal Difference Hybrid θα εκτελεί μόνο τυχαίες ενέργειες για τα πρώτα 50.000 επεισόδια, ενώ θα χρησιμοποιήσει την γνώση του για το περιβάλλον από τα 50.000 επεισόδια και μετά. Με αυτόν τον τρόπο καταλαβαίνουμε τα πλεονεκτήματα που προσφέρει η προσέγγιση του συνδυασμού τυχαίων ενεργειών και χρήση της πολιτικής έτσι ώστε ο πράκτορας να μάθει την βέλτιστη ενέργεια για κάθε κατάσταση.

Συνεπώς οι καμπύλες μάθησης μας παρέχουν την κατάλληλη πληροφορία ως προς την απόδοση των δύο μεθόδων, καθώς ο E-Greedy Strategy μαθαίνει γρηγορότερα και καλύτερα το περιβάλλον σε σχέση με τη μέθοδο Monte-Carlo-Temporal Difference Hybrid, διότι λαμβάνει καλύτερες ανταμοιβές κατά τη διάρκεια των πειραμάτων.



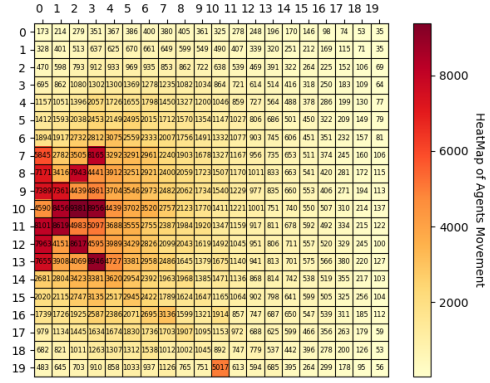
(α′) Ανταμοιβές του Πράκτορα 1

(β′) Ανταμοιβές του Πράκτορα 2

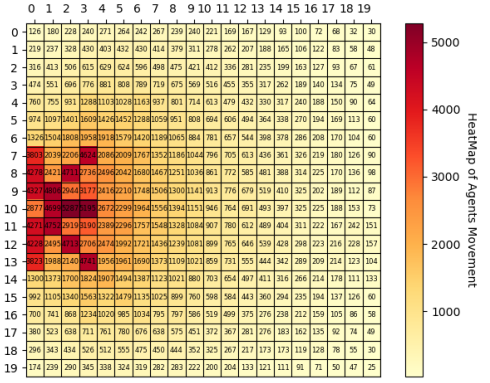
Εικόνα 22: Ανταμοιβές πρακτόρων ανάλογα με τη μέθοδο ενισχυτικής μάθησης

Heatmap

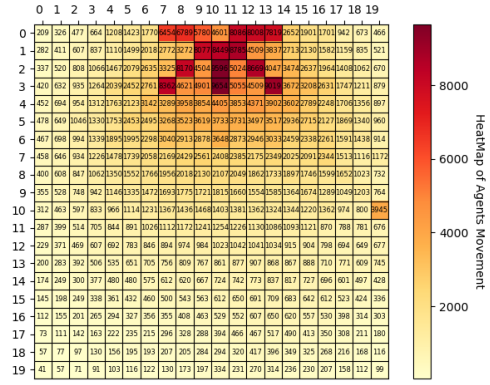
Τα heatmaps των δυο πρακτόρων επιβεβαιώνουν τα παραπάνω αποτελέσματα από τις καμπύλες μάθησης, διότι τόσο ο πράκτορας 1 όσο και ο πράκτορας 2 με τη E-Greedy Strategy μέθοδο έχουν μια ακριβή εικόνα για την τοποθεσία του κόμβου τερματισμού, ενώ η μέθοδος Monte-Carlo-Temporal Difference Hybrid φαίνεται να έχει μια λανθασμένη πολιτική σύμφωνα με την οποία είναι χρησιμότερο για τον πράκτορα να βγαίνει άμεσα από το περιβάλλον με μια μικρή αρνητική ανταμοιβή.



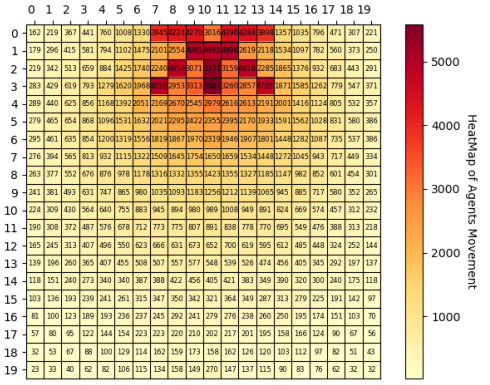
Heatmap για τον πράκτορα 1 με E-Greedy Strategy



Heatmap για τον πράκτορα 1 με Monte-Carlo-Temporal Difference



Heatmap για τον πράκτορα 2 με E-Greedy Strategy



Heatmap για τον πράκτορα 2 με Monte-Carlo-Temporal Difference

Σχήμα 23: Heatmap των πρακτόρων για κάθε μέθοδο

5 Συμπεράσματα

5.1 Μονοπρακτορικό Περιβάλλον

Ο βασικός στόχος της εργασίας για το μονοπρακτορικό περιβάλλον ήταν η εύρεση του βέλτιστου μονοπατιού από τον πράκτορα σε ένα πεπερασμένο σύνολο επεισοδίων. Τα πρώτα πειράματα που εκτελέσαμε αφορούσαν τα σταθερά εμπόδια όπου εξετάσαμε την αποτελεσματικότητα των μεθόδων E-Greedy Strategy και Monte-Carlo-Temporal Difference Hybrid. Πιο συγκεκριμένα η μέθοδος E-Greedy Strategy βρήκε τη λύση σε κάθε πείραμα και είχαμε σταθερά μικρά θετικά αποτελέσματα, ενώ η μέθοδος Monte-Carlo-Temporal Difference Hybrid βρήκε τη λύση στα τρία πειράματα, όπου έλαβε πολύ μεγάλες θετικές ανταμοιβές.

Στην συνέχεια εκτελέσαμε πειράματα όπου τα εμπόδια άλλαζαν θέση στο πλέγμα κάθε 1.000 επεισόδια. Σε αυτή την περίπτωση η μέθοδος E-Greedy Strategy βρήκε την λύση του προβλήματος έξι φορές λαμβάνοντας μικρές θετικές ανταμοιβές ξεπερνώντας τις προκλήσεις του μεταβατικού χαρακτήρα του περιβάλλοντος. Αντιθέτως η μέθοδος MC-TD Hybrid δεν κατόρθωσε να βρει τα αναμενόμενα αποτελέσματα, καθώς το περιβάλλον στο οποίο εκτέλεσε την διαδικασία εξερεύνησης ήταν διαφορετικό από αυτό της διαδικασίας εκμετάλευσης της γνώσης.

Στην τελευταία περίπτωση τα εμπόδια άλλαζαν θέση στο πλέγμα κάθε 10.000 επεισόδια. Το γεγονός αυτό επέτρεψε στη μέθοδο E-Greedy Strategy να λάβει με μεγαλύτερη ακρίβεια την τοποθεσία του κόμβου στόχου και να βρει το βέλτιστο μονοπάτι. Ομοίως με την προηγούμενη περίπτωση η μέθοδος MC-TD Hybrid δεν έλαβε θετικά αποτελέσματα, λόγω της αλλαγής μεταξύ του περιβάλλοντος εξερεύνησης και εκμετάλευσης.

Στα αποτελέσματα των πειραματικών μελετών θα περιμέναμε η μέθοδος MC-TD Hybrid να είχε θετικά αποτελέσματα με μεγάλες ανταμοιβές σε κάθε περίπτωση, λόγω της ταχύτερης μετάδοσης της γνώσης για το περιβάλλον. Ο στόχος αυτός δεν επιτεύχθηκε, διότι το περιβάλλον δεν είχε ένα καλά ορισμένο τέλος. Στο πρόβλημα της εργασίας ο πράκτορας θα μπορούσε να τελειώσει ένα επεισόδιο με την έξοδο του από οποιοδήποτε σημείο στα όρια του πλέγματος λαμβάνοντας ως ανταμοιβή του επεισοδίου τις αρνητικές ανταμοιβές κίνησης μέσα στο περιβάλλον. Συνεπώς το πλεονέκτημα της γρήγορης μετάδοσης γνώσης λειτούργησε ως το βασικό μειονέκτημα της μεθόδου, καθώς πολύ γρήγορα ο πράκτορας έλαβε πολλές αρνητικές ανταμοιβές δημιουργώντας μια πολιτική που δεν ενθαρρύνει την κίνηση μέσα στο περιβάλλον. Στα πειράματα όπου η μέθοδος είχε επιτυχία ο πράκτορας βρήκε τον κόμβο στόχο περίπου 12.000 φορές, τιμή διπλάσια σε σχέση με την αντίστοιχη για την μέθοδο E-Greedy Strategy.

Συμπερασματικά η μέθοδος Monte-Carlo-Temporal Difference Hybrid είχε επιτυχία όταν το περιβάλλον δεν άλλαζε κατά τη διάρκεια του πειράματος με μεγάλες θετικές ανταμοιβές σε ορισμένα πειράματα, ενώ η E-Greedy Strategy με τον συνδυασμό εξερεύνησης-εκμετάλευσης στο ίδιο επεισόδιο έχει επιτυχία σε όλες τις περιπτώσεις με μικρές θετικές ανταμοιβές.

5.2 Πολυπρακτορικό Περιβάλλον

Ο βασικός στόχος της εργασίας για το πολυπρακτορικό περιβάλλον ήταν η εύρεση του βέλτιστου μονοπατιού από τους δυο πράκτορας σε ένα πεπερασμένο σύνολο επεισοδίων. Η μέθοδος E-Greedy Strategy ήταν αποτελεσματική για τους δυο πράκτορες, οι οποίοι κατάφεραν να βρουν ένα σχεδόν βέλτιστο μονοπάτι, αφού έλαβαν κατά μέσο όρο μικρές θετικές ανταμοιβές. Η μέθοδος Monte-Carlo-Temporal Difference Hybrid δεν κατάφερε να λύσει το πρόβλημα. Σε αυτή την περίπτωση η γρηγορότερη μετάδοση της γνώσης για το περιβάλλον αποτέλεσε πρόβλημα, διότι από τα πρώτα επεισόδια η μεγάλη αρνητική ανταμοιβή είχε αποτυπωθεί στον πίνακα Q-table, λόγω της συνύπαρξης των δυο πρακτόρων σε κοντινή περιοχή.

Συνεπώς τα αποτελέσματα του πολυπρακτορικού περιβάλλοντος προσομοιάζουν αυτά του μονοπρακτορικού περιβάλλοντος, καθώς η μέθοδος Monte-Carlo-Temporal Difference Hybrid οδήγησε σε μια πολιτική εξόδου από το περιβάλλον, ενώ η E-Greedy Strategy με την σταδιακή αλλαγή από την εξερεύνηση προς την εκμετάλλευση κατάφερε καλύτερα αποτελέσματα.

6 Μελλοντική Δουλειά

Παραπάνω αναφέραμε τα αποτελέσματα των πειραματικών μελετών της εφαρμογής, όπου είχαμε θετικά αποτελέσματα. Φυσικά πάντα υπάρχει η δυνατότητα βελτίωσης της απόδοσης μια εφαρμογής είτε με την βελτίωση της λειτουργικότητας της είτε προσθέτοντας νέες δυνατότητες. Παρακάτω αναφέρουμε ορισμένες από τις δυνατές βελτιώσεις, οι οποίες θα μπορούσαν να αυξήσουν σημαντικά την χρησιμότητα της παρούσας εφαρμογής.

Μια προφανής βελτίωση της εφαρμογής θα ήταν η προσθήκη σταθερών εμποδιών στο πολυπρακτορικό περιβάλλον, έχοντας έτσι τον συνδυασμό των δυο καταστάσεων. Έτσι ένα πολυπρακτορικό περιβάλλον με σταθερά εμπόδια θα ήταν σίγουρα μια πληρέστερη αναπαράσταση του πραγματικού κόσμου. Δυστυχώς όμως η εκπαίδευση πρακτόρων σε ένα τέτοιο περιβάλλον αποδείχθηκε ως ένα απαιτητικό υπολογιστικά πρόβλημα, καθώς περιελάμβανε πολλές διαφορετικές παραμέτρους που έπρεπε να ληφθούν υπόψη. Αναλυτικότερα ένας πράκτορας θα έπρεπε να μάθει να πετυχαίνει τρεις στόχους ταυτόχρονα, επειδή θα έπρεπε να μάθει που βρίσκονται τα σταθερά εμπόδια, την τροχιά του άλλου πράκτορα και να αντιστοιχίσει με σωστά βάρη τις αρνητικές ανταμοιβές σε αυτές τις παραμέτρους όσο αναζητεί τον κόμβο τερματισμού για να πετύχει τον επιθυμητό στόχο. Για την επίτευξη όλων των αυτών των στόχων, ήταν αναγκαία η εκπαίδευση των πρακτόρων για ένα μεγάλο αριθμό επεισοδίων που προσέγγιζε τα 150.000 έτσι ώστε να έχουμε μια αποδεκτή γνώση του περιβάλλοντος και των μηχανισμών του. Επιπροσθέτως πρέπει να σημειώσουμε ότι τα αποτελέσματα για εκπαίδευση 90.000-100.000 επεισοδίων είχε μεγάλη διακύμανση όσον αφορά το τελικό αποτέλεσμα, γεγονός που μαρτυρά ότι η επιτυχία της εφαρμογής βασιζόταν σε μεγάλο βαθμό σε τύχη.

Με στόχο την βελτίωση της απόδοσης των μεθόδων ενισχυτικής μάθησης σε πολύπλοκα προβλήματα, χρησιμοποιούμε μια τακτική συνδυασμού μεθόδων ενισχυτής μάθησης με μοντέλα της μηχανικής μάθησης. Με την χρήση ενός μοντέλου μηχανικής μάθησης όπως ένα νευρωνικό δίκτυοθα έχουμε την δυνατότητα να λάβουμε υπόψη με μεγαλύτερη ακρίβεια τα δεδομένα του περιβάλλοντος. Με αυτό τον τρόπο για κάθε state-action ζεύγος θα μπορούμε να εισάγουμε ως παραμέτρους στο νευρωνικό δίκτυο τον αριθμό σταθερών εμποδίων, την απόσταση από τον άλλο πράκτορα, την τιμή του πίνακα Q-Learning κ.ά. προσφέροντας έτσι στον πράκτορα ακόμα περισσότερη πληροφορία για το περιβάλλον. Παράλληλα η καλύτερη κατανόηση του περιβάλλοντος είναι πιθανό να μειώσει και τον αριθμό επεισοδίων που είναι απαραίτητα για την πλήρη εκπαίδευση των πρακτόρων. Συνεπώς η χρήση του Deep Q-Learning μπορεί να προσφέρει ακόμα καλύτερα αποτελέσματα αν χρησιμοποιηθεί με τον σωστό τρόπο.

Επίσης βελτιώσεις μπορούν να γίνουν μέσω της προσθήκης νέων δυνατοτήτων στην παρούσα εφαρμογή. Προς το παρόν η εφαρμογή λειτουργεί σε ένα 2-D περιβάλλον, στο οποίο θέλουμε κάθε πράκτορας να έχει την ελάχιστη ασφαλή απόσταση από άλλες οντότητες όπως εμπόδια ή άλλοι πράκτορες. Έτσι υπάρχει η πιθανότητα σε ένα περιβάλλον που προσεγγίζει ακόμα καλύτερα την πραγματικότητα να είχαμε καλύτερη απόδοση και ακρίβεια, καθώς με την χρήση ενός 3-D περιβάλλοντος έχουμε χώρο για την κίνηση περισσότερων πρακτόρων. Έτσι ο περιορισμός των 5 μιλιών θα μπορούσε να έχει μικρότερη επίδραση, καθώς δυο πράκτορες μπορούν να βρίσκονται στην ίδια τροχιά με διαφορά όμως το ύψος στο οποίο πετούν. Φυσικά με την προσθήκη περισσότερων πρακτόρων υπάρχει η πιθανότητα να χρειαστεί η ένταξη αλγορίθμων επικοινωνίας και ιεραρχίας μεταξύ των πρακτόρων, δηλαδή θα είναι πλέον αναγκαίο ένα μοντέλο προτεραιότητας και επικοινωνίας για την λύση τέτοιων προβλημάτων. Μια ακόμη βελτίωση είναι η καλύτερη μοντελοποίηση της συνάρτησης ανταμοιβής όπου εκτός από τον παράγοντα καυσίμου να είχαμε ως παράμετρο το ύψος στο οποίο θα πετάει ο πράκτορας στον 3-D κόσμο. Γνωρίζουμε ότι σε μεγάλα ύψη υπάρχει μικρότερη τριβή από τον αέρα, που θα ήταν χρήσιμο να λαμβάνει υπόψη ο πράκτορας έτσι ώστε να παραμένει σε μεγάλο ύψος μέχρι να πλησιάσει στο αεροδρόμιο όπου και θα πρέπει να μειώσει το ύψος πτήσης του.

Τέλος μια ακόμη βελτίωση θα μπορούσε να είναι η δυνατότητα χαρτογράφησης μιας περιοχής, όπου υπάρχουν πολλαπλά αεροδρόμια και οι αντίστοιχοι περιορισμοί από ουρανοξύστες ή άλλα σταθερά εμπόδια, όπως για παράδειγμα η περιοχή της Νέας Υόρκης και του New Jersey όπου σε μικρή απόσταση έχουμε πολλαπλά αεροδρόμια με υψηλή κίνηση και αρκετούς περιορισμούς ως προς την κίνηση.

7 Αναφορές

Λίστα Συντομογραφιών

AI	Artificial Intelligence
RL	Reinforcement Learning
MARL	Multi-agent Reinforcement Learning
MDP	Markov Decision Process
ATM	Air-Traffic Managemen
TBO	Trajectory-based Operations
CAP	Credit-Assignment Problem
MC-TD	Monte-Carlo-Temporal Difference

References

[1] Peter Norvig Stuart Russel. *Artificial Intelligence, A modern approach*, chapter 21, pages 852–853. Second edition, 2002.

[2] Peter Norvig Stuart Russel. *Artificial Intelligence, A modern approach*, chapter 17, pages 693–694. Second edition, 2002.

[3] Peter Norvig Stuart Russel. *Artificial Intelligence, A modern approach*, chapter 17, pages 693–694. Second edition, 2002.

[4] Peter Norvig Stuart Russel. *Artificial Intelligence, A modern approach*, chapter 21, pages 854–855. Second edition, 2002.

[5] Peter Norvig Stuart Russel. *Artificial Intelligence, A modern approach*, chapter 21, pages 860–861. Second edition, 2002.

[6] Richard Sutton. *Reinforcement Learning: An Introduction*. Second edition, 2018.

[7] Richard Sutton. *Reinforcement Learning: An Introduction*. Second edition, 2018.

[8] Richard Sutton. *Reinforcement Learning: An Introduction*. Second edition, 2018.