ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μελέτη του αλγορίθμου Hedge σε κατανεμημένο περιβάλλον

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΛΕΞΑΝΔΡΟΣ ΠΑΝΑΓΙΩΤΗΣ Α. ΦΕΞΗΣ

Επιβλέπων : Μιλτιάδης Αναγνώστου
Καθηγητής

Αθήνα, Ιούλιος 2024

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων
και Μηχανικων Υπολογιστων
Τομεας Επικοινωνιων, ηλεκτρονικης
Και Συστηματων Πληροφορικης

Μελέτη του αλγορίθμου Hedge σε κατανεμημένο περιβάλλον

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΛΕΞΑΝΔΡΟΣ ΠΑΝΑΓΙΩΤΗΣ Α. ΦΕΞΗΣ

Επιβλέπων : Μιλτιάδης Αναγνώστου
Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 15η Ιουλίου 2024

....	....	....
Ιωάννα Ρουσσάκη	Ευστάθιος Συκάς	Μιλτιάδης Αναγνώστου
Αναπλ. Καθηγήτρια	Ομότιμος Καθηγητής	Καθηγητής

Αθήνα, Ιούλιος 2024

....

Αλέξανδρος Παναγιώτης Α. Φέξης

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

Περίληψη

Η παρούσα εργασία διερευνά την απόδοση και την εφαρμογή του αλγορίθμου Hedge, επίσης γνωστού ως αλγόριθμος Multiplicative Weights Update (MWU), σε διάφορα σενάρια δικτυακής κίνησης. Ο αλγόριθμος Hedge είναι ένας αλγόριθμος μάθησης που έχει σχεδιαστεί για τη λήψη διαδοχικών αποφάσεων σε ανταγωνιστικά περιβάλλοντα με άγνωστες ή μεταβαλλόμενες πιθανότητες. Ενημερώνει την κατανομή σε ένα σύνολο ενεργειών προσαρμόζοντας τα βάρη με βάση την παρατηρούμενη ανατροφοδότηση, εξισορροπώντας έτσι την εξερεύνηση και την εκμετάλλευση. Η παρούσα μελέτη υλοποιεί τον αλγόριθμο Hedge σε Python και διεξάγει μια σειρά πειραμάτων για να αξιολογήσει τη συμπεριφορά του σε περιβάλλοντα ενός παίκτη και πολλών παικτών, συμπεριλαμβανομένων παιχνιδιών μηδενικού αθροίσματος και διαγωνισμών κοινών πόρων. Τα αποτελέσματα δείχνουν ότι ο αλγόριθμος προσαρμόζει αποτελεσματικά τις πιθανότητες με βάση τις απώλειες, ευνοώντας τις ενέργειες με χαμηλότερες απώλειες. Τα πειράματα αποκαλύπτουν ότι, ενώ ο αλγόριθμος σταθεροποιείται με επαρκή αριθμό γύρων, η παράμετρος ϵ επηρεάζει σημαντικά το ρυθμό σύγκλισης. Σε περιβάλλοντα πολλαπλών παικτών, ο αλγόριθμος επιδεικνύει προκλήσεις στη σταθεροποίηση των επιλογών όταν οι απώλειες είναι κοντινές σε αξία και οι παίκτες μερικές φορές αποτυγχάνουν να αναπτύξουν σαφείς προτιμήσεις, ειδικά με περισσότερες επιλογές από τους παίκτες. Η ταυτόχρονη εκτέλεση του αλγορίθμου Hedge σε σενάρια κοινών πόρων δείχνει συνεχείς διακυμάνσεις στις προτιμήσεις, απαιτώντας προσεκτικό συντονισμό των παραμέτρων για σταθερότητα. Συνολικά, ο αλγόριθμος Hedge αποδεικνύεται αποτελεσματικός στην προσαρμογή των πιθανοτήτων και στην εκμάθηση στρατηγικών σε δυναμικά περιβάλλοντα, αν και η πολυπλοκότητα του περιβάλλοντος και ο συντονισμός των παραμέτρων είναι καθοριστικής σημασίας για τη βέλτιστη απόδοση.

Λέξεις-κλειδιά: Αλγόριθμος Hedge, κίνηση δικτύου, διαδικτυακή μάθηση, περιβάλλοντα πολλών παικτών, βελτιστοποίηση

Abstract

This paper investigates the performance and application of the Hedge algorithm, also known as the Multiplicative Weights Update (MWU) algorithm, in various network traffic scenarios. The Hedge algorithm is a learning algorithm designed for making sequential decisions in competitive environments with unknown or changing probabilities. It updates the distribution over a set of actions by adjusting weights based on observed feedback, thus balancing exploration and exploitation. This study implements the Hedge algorithm in Python and conducts a series of experiments to evaluate its behavior in single-player and multi-player environments, including zero-sum games and shared resource competitions. Results indicate that the algorithm effectively adjusts probabilities based on losses, favoring actions with lower losses. Experiments reveal that while the algorithm stabilizes with a sufficient number of rounds, the parameter ϵ significantly impacts the rate of convergence. In multi-player settings, the algorithm demonstrates challenges in stabilizing choices when losses are close in value, and players sometimes fail to develop clear preferences, especially with more choices than players. Simultaneous execution of the Hedge algorithm in shared resource scenarios shows continuous variations in preferences, requiring careful tuning of parameters for stability. Overall, the Hedge algorithm proves effective in adjusting probabilities and learning strategies in dynamic environments, though the complexity of the environment and the tuning of parameters are crucial for optimal performance.

Keywords: Hedge algorithm, network traffic, online learning, multi-player environments, optimization

Περιεχόμενα

Περιεχόμενα Εικόνων

Περιεχόμενα Πινάκων

1. Εισαγωγή στη δρομολόγηση δικτυακής κίνησης

1.1. Εισαγωγή

1.1.1. Ορισμός και σημασία της δικτυακής κίνησης

Η κίνηση δικτύου αναφέρεται στα δεδομένα που διακινούνται σε ένα δίκτυο ανά πάσα στιγμή. Περιλαμβάνει τον όγκο, την ταχύτητα και τη φύση των πακέτων δεδομένων που μεταδίδονται από το ένα σημείο στο άλλο εντός του δικτύου. Η κατανόηση και η διαχείριση της δικτυακής κίνησης είναι ζωτικής σημασίας για τη διατήρηση της αποτελεσματικότητας, της αξιοπιστίας και της ασφάλειας των λειτουργιών του δικτύου.

Οι Pacheco et al., (2019) τονίζουν ότι η ανάλυση της κυκλοφορίας περιλαμβάνει τον εντοπισμό σχέσεων, μοτίβων, ανωμαλιών και λανθασμένων ρυθμίσεων στην κυκλοφορία του διαδικτύου. Με την έλευση νέων τεχνολογιών, όπως η κρυπτογράφηση και η ενθυλάκωση της κυκλοφορίας, οι παραδοσιακές μέθοδοι ταξινόμησης της κυκλοφορίας αντιμετωπίζουν σημαντικές προκλήσεις. Η "Μηχανική Μάθηση (Machine learning - ML)" αναδύεται ως ζωτικό εργαλείο στην ταξινόμηση της κυκλοφορίας, επιτρέποντας την ακριβέστερη διαχείριση της "Ποιότητας Υπηρεσίας (Quality of Service - QoS)" και την εξαγωγή γνώσης από την κρυπτογραφημένη κυκλοφορία. Οι Pacheco et al., (2019) υπογραμμίζουν την ανάγκη για συστηματικές επισκοπήσεις των τεχνικών ML για τη βελτίωση της ταξινόμησης της κυκλοφορίας σε πραγματικά σενάρια δικτύου.

Οι Wang et al. (2012) συζητούν τη σημασία της μέτρησης, κατανόησης και ανάλυσης της κυκλοφορίας σε μεγάλα δίκτυα. Η αποτελεσματική μέτρηση και ανάλυση της κίνησης δικτύου είναι ζωτικής σημασίας για τη διαχείριση και τη διάγνωση του δικτύου. Η έρευνα αναδεικνύει την ποικιλομορφία των ενδιαφερόντων για την παρακολούθηση της δικτυακής κίνησης, συμπεριλαμβανομένης της μέτρησης της δικτυακής κίνησης, της ανίχνευσης δικτυακών επιθέσεων και της ανάλυσης της κίνησης τυχερών παιχνιδιών. Αυτό υπογραμμίζει την πολύπλευρη φύση της δικτυακής κίνησης και τις διάφορες πτυχές που πρέπει να διαχειρίζονται για να διασφαλίζεται η σταθερότητα και η απόδοση του δικτύου.

Οι Joshi & Hadi (2015) παρέχουν μια επισκόπηση των τεχνικών ανάλυσης και πρόβλεψης της δικτυακής κίνησης. Σημειώνουν ότι η ανάλυση και η πρόβλεψη της δικτυακής κίνησης αποτελούν προληπτικές προσεγγίσεις για τη διασφάλιση ασφαλούς, αξιόπιστης και ποιοτικής επικοινωνίας στο δίκτυο. Η μελέτη διερευνά διάφορες τεχνικές, συμπεριλαμβανομένων των νευρωνικών δικτύων και της εξόρυξης δεδομένων, για την επίτευξη αποδοτικών και αποτελεσματικών αποτελεσμάτων στην ανάλυση και πρόβλεψη της δικτυακής κίνησης. Οι συγγραφείς τονίζουν επίσης τη σημασία του εντοπισμού μοναδικών κανόνων και μεθόδων από προηγούμενες μελέτες για την ενίσχυση των τρεχουσών τεχνικών ανάλυσης και πρόβλεψης της κυκλοφορίας.

1.1.2. Τύποι δικτύων

Η δικτυακή κυκλοφορία εμφανίζεται σε διάφορους τύπους δικτύων, καθένα από τα οποία έχει μοναδικά χαρακτηριστικά και προκλήσεις. Οι πρωταρχικοί τύποι δικτύων περιλαμβάνουν δίκτυα τηλεπικοινωνιών, δίκτυα δεδομένων και δίκτυα μεταφορών.

Τα δίκτυα τηλεπικοινωνιών περιλαμβάνουν τη μετάδοση φωνής, δεδομένων και περιεχομένου πολυμέσων. Αυτά τα δίκτυα χαρακτηρίζονται από την ανάγκη για υψηλή αξιοπιστία, χαμηλή καθυστέρηση και υψηλό εύρος ζώνης για την υποστήριξη υπηρεσιών όπως οι φωνητικές κλήσεις, οι τηλεδιασκέψεις και η περιήγηση στο διαδίκτυο. Οι Huang et al., (2019) υπογραμμίζουν την αυξανόμενη σημασία των δεδομένων δικτύου κινητής τηλεφωνίας στην ανάλυση της ταξιδιωτικής συμπεριφοράς και των κοινωνικών αλληλεπιδράσεων. Η μελέτη εξετάζει την ανίχνευση τρόπου μεταφοράς με βάση δεδομένα δικτύου κινητής τηλεφωνίας, σημειώνοντας τις προκλήσεις στην εξαγωγή ακριβών πληροφοριών κινητικότητας λόγω της θορυβώδους και χρονικά σπάνιας φύσης των δεδομένων.

Τα δίκτυα δεδομένων, όπως τα “Τοπικά Δίκτυα (Local Area Networks - LAN)” και τα “Δίκτυα Ευρείας Περιοχής (Wide Area Networks - WAN)”, διευκολύνουν την ανταλλαγή δεδομένων μεταξύ υπολογιστών και άλλων συσκευών. Αυτά τα δίκτυα είναι ζωτικής σημασίας για τις επιχειρηματικές λειτουργίες, τον υπολογιστικό νέφος και την αποθήκευση δεδομένων. Ο Salomon (1986) εξετάζει τις αλληλεπιδράσεις μεταξύ τηλεπικοινωνιακών και μεταφορικών δικτύων, τονίζοντας τις συμπληρωματικές και υποκατάστατες σχέσεις μεταξύ των δύο συστημάτων. Η μελέτη εξετάζει τις εφαρμογές της τεχνολογίας των τηλεπικοινωνιών για την εξ αποστάσεως εργασία, τις τηλεδιασκέψεις και τις κινητές επικοινωνίες, τονίζοντας την ανάγκη αξιολόγησης των μελλοντικών τροποποιήσεων των μετακινήσεων και όχι την εστίαση αποκλειστικά στις υποσχέσεις υποκατάστασης.

Τα δίκτυα μεταφορών, συμπεριλαμβανομένων των οδικών, σιδηροδρομικών, αεροπορικών και θαλάσσιων συστημάτων, είναι απαραίτητα για τη διακίνηση αγαθών και ανθρώπων. Τα δίκτυα αυτά βασίζονται σε μεγάλο βαθμό στις τεχνολογίες πληροφοριών και επικοινωνιών (ΤΠΕ) για τη βελτίωση της αποτελεσματικότητας και της ασφάλειας. Ο Giannopoulos (2004) εξετάζει την εφαρμογή των ΤΠΕ στα δίκτυα μεταφορών, εστιάζοντας στη λειτουργία και τη διαχείριση του δικτύου, στην πληροφόρηση και την καθοδήγηση των χρηστών και στα συστήματα εμπορευματικών μεταφορών. Η μελέτη υπογραμμίζει τις δυνατότητες των καθιερωμένων τεχνολογιών για τη συλλογή κυκλοφοριακών δεδομένων, τον έλεγχο του δικτύου και την παρακολούθηση των οχημάτων, προβλέποντας την ευρεία εφαρμογή τους στο μέλλον.

1.1.3. Προκλήσεις στη διαχείριση της κυκλοφορίας των δικτύων

Η διαχείριση της κίνησης του δικτύου περιλαμβάνει την αντιμετώπιση διαφόρων προκλήσεων για τη διατήρηση της βέλτιστης απόδοσης και της ικανοποίησης των χρηστών. Στις βασικές προκλήσεις περιλαμβάνονται η συμφόρηση, η καθυστέρηση και η απώλεια πακέτων.

Η συμφόρηση εμφανίζεται όταν η ζήτηση του δικτύου υπερβαίνει τη διαθέσιμη χωρητικότητα, οδηγώντας σε βραδύτερη μετάδοση δεδομένων και αυξημένες καθυστερήσεις. Οι αποτελεσματικοί μηχανισμοί ελέγχου συμφόρησης είναι απαραίτητοι για την αποτροπή της υπερφόρτωσης του δικτύου και τη διασφάλιση της ομαλής ροής δεδομένων. Οι Kafi et al. (2014) συζητούν τις μοναδικές προκλήσεις σχεδιασμού του ελέγχου συμφόρησης σε “Ασύρματα Δίκτυα Αισθητήρων (Wireless Sensor Networks - WSN)”. Κατηγοριοποιούν τα πρωτόκολλα ελέγχου συμφόρησης σε έλεγχο πόρων και έλεγχο κίνησης, τονίζοντας τη σημασία της αξιόπιστης παράδοσης μηνυμάτων, της ενεργειακής απόδοσης και του QoS για την επίτευξη υψηλής απόδοσης και μεγάλης διάρκειας ζωής του δικτύου.

Η καθυστέρηση ή καθυστέρηση αναφέρεται στο χρόνο που απαιτείται για να ταξιδέψουν τα δεδομένα από την πηγή στον προορισμό. Η υψηλή καθυστέρηση μπορεί να επηρεάσει σημαντικά τις εφαρμογές πραγματικού χρόνου, όπως οι τηλεδιασκέψεις, τα διαδικτυακά παιχνίδια και οι υπηρεσίες VoIP. Οι Sunassee et al.,

(2021) εξετάζουν διάφορους μηχανισμούς ελέγχου συμφόρησης για την αντιμετώπιση της υψηλής συχνότητας μπλοκαρίσματος, των καθυστερήσεων και των περιττών αναμεταδόσεων στα δίκτυα. Η μελέτη προτείνει κριτήρια για την ανάπτυξη αποτελεσματικών τεχνικών ελέγχου συμφόρησης για τη διατήρηση βιώσιμων επιδόσεων του δικτύου.

Η απώλεια πακέτων συμβαίνει όταν τα πακέτα δεδομένων αποτυγχάνουν να φτάσουν στον προορισμό τους, οδηγώντας σε ελλιπή ή αλλοιωμένη μετάδοση δεδομένων. Αυτό μπορεί να προκληθεί από συμφόρηση δικτύου, βλάβες υλικού ή παρεμβολές σήματος. Οι Kushwaha & Gupta (2014) παρουσιάζουν μια επισκόπηση των προσεγγίσεων ελέγχου συμφόρησης σε ενσύρματα δίκτυα υψηλής ταχύτητας, εξετάζοντας τόσο τις μεθόδους ελέγχου που βασίζονται στην πηγή όσο και τις μεθόδους ελέγχου που βασίζονται σε δρομολογητές. Η μελέτη υπογραμμίζει την ανάγκη για μια ολιστική προσέγγιση για την αντιμετώπιση της αλληλεπίδρασης μεταξύ αυτών των μεθόδων και προσδιορίζει σημαντικά ζητήματα και προκλήσεις στον έλεγχο συμφόρησης.

Η δικτυακή κυκλοφορία είναι ένας σύνθετος και πολύπλευρος τομέας που απαιτεί ολοκληρωμένες στρατηγικές για την αποτελεσματική διαχείριση. Με την κατανόηση των τύπων δικτύων και των προκλήσεων που αντιμετωπίζουν, οι διαχειριστές δικτύων μπορούν να εφαρμόσουν τις κατάλληλες τεχνικές για να εξασφαλίσουν αποδοτικές, αξιόπιστες και ασφαλείς λειτουργίες δικτύου. Η ενσωμάτωση προηγμένων τεχνολογιών, όπως η μηχανική μάθηση και οι ΤΠΕ, ενισχύει περαιτέρω την ικανότητα διαχείρισης και βελτιστοποίησης της δικτυακής κίνησης σε δυναμικά περιβάλλοντα.

1.2. Διαχείριση της κυκλοφορίας στα δίκτυα

1.2.1. Τεχνικές και αλγόριθμοι

Επισκόπηση των κοινών τεχνικών διαχείρισης της κυκλοφορίας: Αλγόριθμοι δρομολόγησης, έλεγχος συμφόρησης.

Η διαχείριση της δικτυακής κίνησης περιλαμβάνει διάφορες τεχνικές και αλγορίθμους που αποσκοπούν στη βελτιστοποίηση της ροής των δεδομένων στο δίκτυο, εξασφαλίζοντας αποδοτικότητα, αξιοπιστία και υψηλές επιδόσεις. Οι κύριες τεχνικές περιλαμβάνουν την εξισορρόπηση φορτίου, τους αλγορίθμους δρομολόγησης και τον έλεγχο συμφόρησης.

Εξισορρόπηση φορτίου

Η εξισορρόπηση φορτίου είναι μια μέθοδος που χρησιμοποιείται για την κατανομή της δικτυακής κίνησης σε πολλούς διακομιστές ή μονοπάτια δικτύου, ώστε να αποτρέπεται η υπερφόρτωση οποιουδήποτε μεμονωμένου διακομιστή ή μονοπατιού. Αυτή η τεχνική εξασφαλίζει τη βέλτιστη αξιοποίηση των πόρων, ελαχιστοποιεί το χρόνο απόκρισης και αποφεύγει την υπερφόρτωση, η οποία μπορεί να οδηγήσει σε διακοπή λειτουργίας. Σύμφωνα με τους Matnee et al. (2018), η εμφάνιση των «Δικτύων Καθορισμένων από Λογισμικό (Software Defined Networks - SDN)» έχει βελτιώσει σημαντικά τις δυνατότητες εξισορρόπησης φορτίου. Το SDN αποσυνδέει το επίπεδο ελέγχου από το επίπεδο δεδομένων, επιτρέποντας την κεντρική διαχείριση των ροών κυκλοφορίας. Ένας προγραμματιζόμενος ελεγκτής διαχειρίζεται τους κανόνες δρομολόγησης, επιτρέποντας δυναμικές προσαρμογές με βάση τις συνθήκες του δικτύου σε πραγματικό χρόνο. Αυτή η ευελιξία επιτρέπει την αποτελεσματικότερη κατανομή φορτίου και τη βελτίωση του QoS.

Αλγόριθμοι δρομολόγησης

Οι αλγόριθμοι δρομολόγησης καθορίζουν την καλύτερη διαδρομή για τη διέλευση των πακέτων δεδομένων σε ένα δίκτυο. Αυτοί οι αλγόριθμοι είναι ζωτικής σημασίας για τη διατήρηση αποτελεσματικής και αξιόπιστης επικοινωνίας. Οι συνήθεις αλγόριθμοι δρομολόγησης περιλαμβάνουν:

- **Συντομότερη Διαδρομή Πρώτα (Shortest Path First – SPF):** Αυτός ο αλγόριθμος, που χρησιμοποιείται από πρωτόκολλα όπως το OSPF (Open Shortest Path First), υπολογίζει τη συντομότερη διαδρομή προς έναν προορισμό με βάση διάφορες μετρικές όπως ο αριθμός των μεταπηδήσεων, το εύρος ζώνης και η καθυστέρηση.
- **Δρομολόγηση διανύσματος απόστασης:** Αλγόριθμοι όπως το RIP (Routing Information Protocol) χρησιμοποιούν διανύσματα απόστασης για τον προσδιορισμό της καλύτερης διαδρομής. Κάθε δρομολογητής διατηρεί έναν πίνακα με τις καλύτερες γνωστές αποστάσεις προς κάθε προορισμό και

ενημερώνει τον πίνακά του με βάση πληροφορίες από γειτονικούς δρομολογητές.

- **Δρομολόγηση κατάστασης συνδέσεων:** Σε αντίθεση με τη δρομολόγηση διανύσματος απόστασης, οι αλγόριθμοι δρομολόγησης κατάστασης συνδέσεων διατηρούν έναν πλήρη χάρτη της τοπολογίας του δικτύου. Κάθε δρομολογητής υπολογίζει ανεξάρτητα τη συντομότερη διαδρομή προς κάθε άλλο δρομολογητή.

Έλεγχος συμφόρησης

Οι μηχανισμοί ελέγχου συμφόρησης αποσκοπούν στην αποτροπή της συμφόρησης του δικτύου, η οποία συμβαίνει όταν η ζήτηση για πόρους του δικτύου υπερβαίνει τη διαθέσιμη χωρητικότητα, οδηγώντας σε απώλεια πακέτων και αυξημένη καθυστέρηση. Ο αποτελεσματικός έλεγχος συμφόρησης είναι ζωτικής σημασίας για τη διατήρηση υψηλής απόδοσης και την ελαχιστοποίηση των καθυστερήσεων. Οι Kumar et al. (2021) υπογραμμίζουν το ρόλο των προσαρμοστικών σχημάτων επεξεργασίας σήματος στον έλεγχο συμφόρησης. Αυτά τα σχήματα προσαρμόζουν τους ρυθμούς μετάδοσης με βάση τις συνθήκες του δικτύου για την αποφυγή υπερφόρτωσης. Επιπλέον, οι προσεγγίσεις που βασίζονται στο SDN παρέχουν δυναμικό και προγραμματιζόμενο έλεγχο της κίνησης του δικτύου, ενισχύοντας περαιτέρω τη διαχείριση της συμφόρησης.

1.2.2. Σημασία των προσαρμοστικών αλγορίθμων σε δυναμικά περιβάλλοντα

Οι προσαρμοστικοί αλγόριθμοι είναι απαραίτητοι σε δυναμικά περιβάλλοντα δικτύων όπου οι συνθήκες μπορούν να αλλάξουν γρήγορα. Αυτοί οι αλγόριθμοι μπορούν να προσαρμόζουν τις παραμέτρους τους σε πραγματικό χρόνο με βάση την ανατροφοδότηση από το δίκτυο, εξασφαλίζοντας τη βέλτιστη απόδοση ακόμη και υπό κυμαινόμενες συνθήκες. Οι Ge et al. (2019) συζητούν την αποτελεσματικότητα των προσαρμοστικών αλγορίθμων σε διάφορα συστήματα μάθησης, σημειώνοντας την ικανότητά τους να προσαρμόζονται δυναμικά σε μεταβαλλόμενα περιβάλλοντα. Στη διαχείριση της δικτυακής κίνησης, οι προσαρμοστικοί αλγόριθμοι βοηθούν στη διατήρηση της ισορροπίας φορτίου, στη βελτιστοποίηση των διαδρομών δρομολόγησης και στον έλεγχο της συμφόρησης, παρακολουθώντας συνεχώς και ανταποκρινόμενοι στις συνθήκες του δικτύου.

Οι Aleti & Moser (2016) τονίζουν τη σημασία του προσαρμοστικού ελέγχου των παραμέτρων στους εξελικτικούς αλγορίθμους. Αυτοί οι αλγόριθμοι προσαρμόζουν τις παραμέτρους τους κατά τη διάρκεια της διαδικασίας βελτιστοποίησης με βάση την ανατροφοδότηση της απόδοσης. Αυτή η προσέγγιση είναι ιδιαίτερα επωφελής στη διαχείριση δικτύων, όπου οι προσαρμοστικοί αλγόριθμοι μπορούν να βελτιστοποιήσουν δυναμικά τη δρομολόγηση και την εξισορρόπηση φορτίου, εξασφαλίζοντας αποτελεσματική ροή δεδομένων και ελάχιστη συμφόρηση.

1.3. Μοντέλα κίνησης δικτύου

1.3.1. Επεξήγηση των διαφόρων μοντέλων κυκλοφορίας

Τα μοντέλα κίνησης δικτύου είναι μαθηματικές αναπαραστάσεις που χρησιμοποιούνται για την περιγραφή και την πρόβλεψη της συμπεριφοράς της κίνησης σε ένα δίκτυο. Διαφορετικά μοντέλα αποτυπώνουν διάφορες πτυχές της δικτυακής κίνησης, όπως τα πρότυπα άφιξης, η εκρηκτικότητα και η αυτο-ομοιότητα.

Μοντέλο κυκλοφορίας Poisson

Το μοντέλο κίνησης Poisson είναι ένα από τα παλαιότερα και απλούστερα μοντέλα που χρησιμοποιούνται για την αναπαράσταση των τυχαίων αφίξεων κίνησης στα δίκτυα. Υποθέτει ότι οι αφίξεις πακέτων συμβαίνουν ανεξάρτητα και με σταθερό μέσο ρυθμό. Αυτό το μοντέλο είναι κατάλληλο για δίκτυα με μεγάλο αριθμό μικρών, ανεξάρτητων πηγών δεδομένων. Οι Gu et al., (2016) περιγράφουν την ιστορική σημασία του μοντέλου Poisson, σημειώνοντας την εφαρμογή του στα πρώτα τηλεπικοινωνιακά δίκτυα. Παρά την απλότητά του, το μοντέλο Poisson είναι περιορισμένο ως προς την ικανότητά του να αποτυπώνει την εκρηκτική και συσχετισμένη φύση της σύγχρονης δικτυακής κίνησης.

Μοντέλο αυτο-ομοειδούς κυκλοφορίας

Τα μοντέλα αυτο-ομοιόμορφης κίνησης χρησιμοποιούνται για την περιγραφή της κίνησης που παρουσιάζει εξάρτηση σε μεγάλη απόσταση και εκρηκτικότητα σε πολλαπλές χρονικές κλίμακες. Σε αντίθεση με την κυκλοφορία Poisson, η αυτο-ομοιόμορφη κυκλοφορία δεν εξομαλύνεται όταν αθροίζεται με την πάροδο του χρόνου. Αυτό το μοντέλο είναι πιο ακριβές για την αναπαράσταση της πραγματικής διαδικτυακής κίνησης, η οποία συχνά παρουσιάζει υψηλή μεταβλητότητα και εκρηκτικότητα. Οι Tyagi et al. (2009) υπογραμμίζουν τη σημασία των αυτοομοιογενών μοντέλων για την αποτύπωση της πολύπλοκης συμπεριφοράς της σύγχρονης δικτυακής κίνησης. Τα μοντέλα αυτά είναι ιδιαίτερα χρήσιμα για την αξιολόγηση της απόδοσης και τον προγραμματισμό της χωρητικότητας σε δίκτυα υψηλών ταχυτήτων.

Μοντέλο εκρηκτικής κυκλοφορίας

Τα μοντέλα bursty traffic περιγράφουν πρότυπα κίνησης που χαρακτηρίζονται από ξαφνικές αυξήσεις στις αφίξεις πακέτων που ακολουθούνται από περιόδους χαμηλής δραστηριότητας. Αυτός ο τύπος κίνησης είναι κοινός σε δίκτυα με εφαρμογές που παράγουν δεδομένα με ριπές, όπως η ροή βίντεο, οι μεταφορές αρχείων και τα διαδικτυακά παιχνίδια. Οι Onof et al. (2000) συζητούν τις διαδικασίες Poisson-cluster που χρησιμοποιούνται για τη μοντελοποίηση της εκρηκτικής κυκλοφορίας, σημειώνοντας την ικανότητά τους να αναπαριστούν την ομαδοποιημένη άφιξη των πακέτων. Τα μοντέλα εκρηκτικής κυκλοφορίας βοηθούν τους σχεδιαστές δικτύων να κατανοήσουν και να διαχειριστούν τον αντίκτυπο των εκρήξεων κυκλοφορίας στην απόδοση του δικτύου.

1.3.2. Επίδραση των προτύπων κίνησης στην απόδοση του δικτύου

Τα πρότυπα κίνησης σε ένα δίκτυο έχουν σημαντικό αντίκτυπο στην απόδοσή του. Διαφορετικά μοντέλα κίνησης παρέχουν πληροφορίες για το πώς η μεταβλητότητα της κίνησης, η εκρηκτικότητα και η εξάρτηση μεγάλης εμβέλειας επηρεάζουν τη συμπεριφορά του δικτύου.

Μετρικές επιδόσεων

Οι βασικές μετρικές επιδόσεων που επηρεάζονται από τα πρότυπα κίνησης περιλαμβάνουν την καθυστέρηση, την απόδοση, την απώλεια πακέτων και το jitter. Οι Pacheco κ.ά. (2019) τονίζουν τη σημασία της ανάλυσης της κίνησης για τον εντοπισμό μοτίβων, ανωμαλιών και λανθασμένων ρυθμίσεων που επηρεάζουν αυτές τις μετρικές. Η κατανόηση των υποκείμενων μοτίβων κίνησης βοηθά στη βελτιστοποίηση της απόδοσης του δικτύου και στη διασφάλιση του QoS.

Σχεδιασμός και προγραμματισμός δικτύου

Τα μοτίβα κίνησης επηρεάζουν τον σχεδιασμό του δικτύου και τον προγραμματισμό της χωρητικότητας. Τα δίκτυα που έχουν σχεδιαστεί για να διαχειρίζονται την κυκλοφορία Poisson μπορεί να δυσκολεύονται με την αυτο-ομοιόμορφη ή εκρηκτική κυκλοφορία, οδηγώντας σε συμφόρηση και υποβαθμισμένη απόδοση. Οι Schwefel et al. (2017) υπογραμμίζουν την ανάγκη για ακριβή μοντέλα κίνησης για την πρόβλεψη της συμπεριφοράς της απόδοσης σε συνδέσεις συμφόρησης και μεγαλύτερα δίκτυα. Με τη χρήση κατάλληλων μοντέλων κίνησης, οι σχεδιαστές δικτύων μπορούν να κατανέμουν καλύτερα τους πόρους, να βελτιστοποιούν τη δρομολόγηση και να εφαρμόζουν αποτελεσματικές στρατηγικές ελέγχου συμφόρησης.

Προσαρμοστική διαχείριση

Οι τεχνικές προσαρμοστικής διαχείρισης της κίνησης αξιοποιούν δεδομένα κίνησης σε πραγματικό χρόνο για τη δυναμική προσαρμογή των παραμέτρων του δικτύου. Αυτή η προσέγγιση είναι ζωτικής σημασίας για τον χειρισμό της μεταβλητότητας και του απρόβλεπτου της σύγχρονης δικτυακής κίνησης. Οι προσαρμοστικοί αλγόριθμοι μπορούν να βελτιστοποιήσουν τη δρομολόγηση, να εξισορροπήσουν τα φορτία και να ελέγξουν αποτελεσματικότερα τη συμφόρηση, παρακολουθώντας συνεχώς τα πρότυπα της κυκλοφορίας και προσαρμόζοντας τα ανάλογα.

Η αποτελεσματική διαχείριση της κυκλοφορίας στα δίκτυα βασίζεται σε έναν συνδυασμό ισχυρών τεχνικών και προσαρμοστικών αλγορίθμων. Η κατανόηση των διαφόρων μοντέλων κίνησης και των επιπτώσεών τους στην απόδοση του δικτύου είναι απαραίτητη για το σχεδιασμό και τη διαχείριση δικτύων που μπορούν να χειριστούν τη δυναμική και πολύπλοκη φύση της σύγχρονης κίνησης. Μέσω της χρήσης προηγμένων αλγορίθμων και της ανάλυσης της κυκλοφορίας σε πραγματικό χρόνο, οι διαχειριστές δικτύων μπορούν να εξασφαλίσουν αποδοτικά, αξιόπιστα και υψηλής απόδοσης δίκτυα.

1.4. Διαδικτυακή μάθηση και βελτιστοποίηση στην κυκλοφορία του δικτύου

1.4.1. Εισαγωγή στη διαδικτυακή μάθηση

Ορισμός και σημασία σε δικτυακά περιβάλλοντα

Η διαδικτυακή μάθηση αναφέρεται στην ικανότητα ενός συστήματος να μαθαίνει και να προσαρμόζεται σε πραγματικό χρόνο καθώς νέα δεδομένα γίνονται διαθέσιμα. Αυτό έρχεται σε αντίθεση με τη μάθηση χωρίς σύνδεση, όπου το μοντέλο εκπαιδεύεται σε ένα σταθερό σύνολο δεδομένων πριν από την ανάπτυξη. Στο πλαίσιο των νευρωνικών δικτύων, η online μάθηση επιτρέπει στα μοντέλα να ενημερώνουν συνεχώς τη βάση γνώσης τους και να βελτιώνουν την απόδοσή τους καθώς συναντούν νέα δεδομένα. Οι Jain et al., (2014) παρέχουν μια ολοκληρωμένη επισκόπηση της online μάθησης στα νευρωνικά δίκτυα, υπογραμμίζοντας την εφαρμογή της σε διάφορες αρχιτεκτονικές, συμπεριλαμβανομένων των feedforward, recurrent και fuzzy νευρωνικών δικτύων. Η ικανότητα σταδιακής μάθησης είναι ιδιαίτερα πολύτιμη σε δυναμικά περιβάλλοντα όπου οι συνθήκες αλλάζουν γρήγορα, όπως στη διαχείριση της δικτυακής κίνησης.

Σε περιβάλλοντα δικτύων, η διαδικτυακή μάθηση είναι ζωτικής σημασίας για τη διατήρηση αποδοτικών και αξιόπιστων λειτουργιών. Η κυκλοφορία του δικτύου είναι εγγενώς δυναμική, με μοτίβα και όγκους που αυξομειώνονται κατά τη διάρκεια της ημέρας. Τα παραδοσιακά στατικά μοντέλα είναι συχνά ανεπαρκή για τον χειρισμό αυτής της μεταβλητότητας, οδηγώντας σε μη βέλτιστες επιδόσεις και πιθανή συμφόρηση του δικτύου. Οι αλγόριθμοι διαδικτυακής μάθησης, με τη συνεχή προσαρμογή σε νέα πρότυπα κίνησης, μπορούν να κάνουν προσαρμογές σε πραγματικό χρόνο στις αποφάσεις δρομολόγησης και εξισορρόπησης φορτίου, βελτιστοποιώντας έτσι την απόδοση του δικτύου και μειώνοντας την καθυστέρηση και την απώλεια πακέτων.

Οι Noaeen κ.ά. (2022) υπογραμμίζουν τη σημασία της "Ενισχυτικής Μάθησης (Reinforcement Learning - RL)" στον "Έλεγχο Σηματοδότησης της Κυκλοφορίας (Traffic Signal Control - TSC)", έναν τομέα που συνδέεται στενά με τη διαχείριση της δικτυακής κυκλοφορίας. Η επισκόπησή τους καταδεικνύει πώς οι τεχνικές RL μπορούν να βελτιώσουν τις αστικές μεταφορές με την προσαρμογή στις μεταβαλλόμενες συνθήκες κυκλοφορίας σε πραγματικό χρόνο, βελτιώνοντας έτσι τη συνολική αποδοτικότητα και την ποιότητα ζωής. Αυτή η ικανότητα δυναμικής προσαρμογής σε νέα δεδομένα είναι αυτό που καθιστά τη διαδικτυακή μάθηση ένα ισχυρό εργαλείο σε περιβάλλοντα δικτύων.

Σύγκριση με παραδοσιακές μεθόδους μάθησης

Οι παραδοσιακές μέθοδοι μάθησης, που συχνά αναφέρονται ως μάθηση χωρίς σύνδεση ή μάθηση παρτίδας, περιλαμβάνουν την εκπαίδευση ενός μοντέλου σε ένα προκαθορισμένο σύνολο δεδομένων. Μόλις εκπαιδευτεί, οι παράμετροι του μοντέλου παραμένουν σταθερές και αναπτύσσεται για να κάνει προβλέψεις ή αποφάσεις με βάση τη γνώση που έχει μάθει. Ενώ αυτή η προσέγγιση μπορεί να είναι αποτελεσματική για στατικά ή αργά μεταβαλλόμενα περιβάλλοντα, στερείται της ευελιξίας που απαιτείται για δυναμικές ρυθμίσεις, όπως η διαχείριση της δικτυακής κίνησης.

Οι Pacheco et al., (2019) επισημαίνουν τους περιορισμούς των παραδοσιακών στρατηγικών ταξινόμησης της κυκλοφορίας, ιδίως ενόψει των νέων τεχνολογιών, όπως η κρυπτογράφηση και η ενθυλάκωση, οι οποίες υποβαθμίζουν την απόδοση των κλασικών μεθόδων. Οι τεχνικές μηχανικής μάθησης (ML), ιδίως εκείνες που είναι ικανές για online μάθηση, προσφέρουν μια διέξοδο προς τα εμπρός, καθώς ενημερώνουν συνεχώς τα μοντέλα τους για να προσαρμόζονται στα εξελισσόμενα πρότυπα κίνησης. Αυτή η προσαρμοστικότητα οδηγεί σε ακριβέστερη διαχείριση της ποιότητας υπηρεσίας (QoS) και σε καλύτερο χειρισμό της κρυπτογραφημένης κυκλοφορίας.

Οι Srinidhi et al., (2019) συζητούν τις προκλήσεις που θέτει το Διαδίκτυο των πραγμάτων (IoT), όπου ένας τεράστιος αριθμός μικρών συσκευών με περιορισμένους πόρους παράγει σημαντικό όγκο δεδομένων. Οι παραδοσιακές μέθοδοι εκμάθησης συχνά δυσκολεύονται με την επεκτασιμότητα και τις απαιτήσεις πραγματικού χρόνου των δικτύων IoT. Αντίθετα, οι τεχνικές διαδικτυακής μάθησης και βελτιστοποίησης μπορούν να προσαρμόζονται δυναμικά στην εισροή δεδομένων, αντιμετωπίζοντας αποτελεσματικότερα ζητήματα που σχετίζονται με τη δρομολόγηση, την εξοικονόμηση ενέργειας, τη συμφόρηση και το QoS.

Το βασικό πλεονέκτημα της online μάθησης έναντι των παραδοσιακών μεθόδων έγκειται στην ικανότητά της να χειρίζεται δεδομένα σε πραγματικό χρόνο και να προβαίνει σε άμεσες προσαρμογές. Αυτή η συνεχής διαδικασία μάθησης διασφαλίζει ότι το μοντέλο παραμένει σχετικό και αποτελεσματικό, ακόμη και όταν οι συνθήκες του δικτύου αλλάζουν. Οι παραδοσιακές μέθοδοι, αν και χρήσιμες σε ορισμένα πλαίσια, συχνά απαιτούν επανεκπαίδευση σε νέα σύνολα δεδομένων για να διατηρηθεί η ακρίβεια, γεγονός που μπορεί να είναι χρονοβόρο και μη πρακτικό για εφαρμογές πραγματικού χρόνου.

1.4.2. Βελτιστοποίηση στη διαχείριση της κυκλοφορίας του δικτύου

Ανάγκη για βελτιστοποίηση σε πραγματικό χρόνο

Η διαχείριση της δικτυακής κίνησης είναι ένα πολύπλοκο έργο που απαιτεί βελτιστοποίηση σε πραγματικό χρόνο για να διασφαλιστεί η αποτελεσματική ροή δεδομένων και η υψηλή απόδοση του δικτύου. Η ανάγκη για βελτιστοποίηση σε πραγματικό χρόνο προκύπτει από τη δυναμική φύση της δικτυακής κίνησης, όπου οι ξαφνικές αιχμές στη χρήση, οι μεταβαλλόμενες συμπεριφορές των χρηστών και οι ποικίλες απαιτήσεις των εφαρμογών μπορούν να οδηγήσουν σε συμφόρηση, αυξημένη καθυστέρηση και απώλεια πακέτων.

Οι Shi et al., (2020) τονίζουν τη σημασία της πρόβλεψης της κίνησης του δικτύου για την πρόληψη περιστατικών ασφαλείας και την εξασφάλιση ομαλής λειτουργίας του δικτύου. Συζητούν το ρόλο των τεχνολογιών βελτιστοποίησης και αποσύνθεσης στα μοντέλα πρόβλεψης της δικτυακής κίνησης, τονίζοντας την αποτελεσματικότητα τεχνικών όπως η “Βελτιστοποίηση Σμήνους Σωματιδίων (Particle Swarm Optimization - PSO)” και η “Αποσύνθεση Μεταβλητών Τρόπων (Variational Mode Decomposition - VMD)” στη βελτίωση της ακρίβειας πρόβλεψης και της ταχύτητας σύγκλισης. Αυτές οι μέθοδοι βελτιστοποίησης επιτρέπουν στους διαχειριστές του δικτύου να προβλέπουν τα μοτίβα κίνησης και να προβαίνουν σε προληπτικές προσαρμογές για την πρόληψη της συμφόρησης και τη διατήρηση του QoS.

Οι Matnee et al. (2018) διερευνούν τον ρόλο των δικτύων καθορισμένων από λογισμικό (SDN) στη βελτιστοποίηση δικτύων σε πραγματικό χρόνο. Το SDN αποσυνδέει το επίπεδο ελέγχου από το επίπεδο δεδομένων, επιτρέποντας τον κεντρικό έλεγχο και τις δυναμικές προσαρμογές στις διαμορφώσεις του δικτύου. Αυτή η δυνατότητα προγραμματισμού επιτρέπει τη βελτιστοποίηση της κυκλοφορίας δεδομένων σε πραγματικό χρόνο, τη βελτίωση της εξισορρόπησης φορτίου και τον έλεγχο συμφόρησης. Η ικανότητα του SDN να προσαρμόζεται γρήγορα στις μεταβαλλόμενες συνθήκες του δικτύου το καθιστά ένα ισχυρό εργαλείο για τη διατήρηση της βέλτιστης απόδοσης σε δυναμικά περιβάλλοντα.

Η βελτιστοποίηση σε πραγματικό χρόνο στη διαχείριση της κυκλοφορίας δικτύου περιλαμβάνει τη συνεχή παρακολούθηση των συνθηκών του δικτύου και την άμεση προσαρμογή των μονοπατιών δρομολόγησης, της κατανομής φορτίου και της κατανομής πόρων. Αυτή η προληπτική προσέγγιση διασφαλίζει ότι το δίκτυο μπορεί να διαχειριστεί αποτελεσματικά τα μεταβαλλόμενα φορτία κυκλοφορίας, ελαχιστοποιώντας τον κίνδυνο συμφόρησης και διατηρώντας υψηλή απόδοση.

1.4.3. Προσαρμοστικοί αλγόριθμοι στη διαχείριση της δικτυακής κίνησης

Οι προσαρμοστικοί αλγόριθμοι διαδραματίζουν κρίσιμο ρόλο στην επίτευξη βελτιστοποίησης σε πραγματικό χρόνο στη διαχείριση της κυκλοφορίας του δικτύου. Αυτοί οι αλγόριθμοι μπορούν να προσαρμόζουν τις παραμέτρους τους και τις διαδικασίες λήψης αποφάσεων με βάση την ανατροφοδότηση σε πραγματικό χρόνο από το δίκτυο, εξασφαλίζοντας βέλτιστη απόδοση ακόμη και υπό μεταβαλλόμενες συνθήκες.

Οι Ge et al., (2019) εξετάζουν τους προσαρμοστικούς αλγορίθμους που εφαρμόζονται σε διάφορα συστήματα μάθησης, σημειώνοντας την αποτελεσματικότητά τους στη δυναμική προσαρμογή σε νέα δεδομένα. Στο πλαίσιο της διαχείρισης της κυκλοφορίας του δικτύου, οι προσαρμοστικοί αλγόριθμοι μπορούν να βελτιστοποιήσουν τις αποφάσεις δρομολόγησης, να εξισορροπήσουν τα φορτία και να διαχειριστούν τη συμφόρηση, μαθαίνοντας συνεχώς από τα πρότυπα κυκλοφορίας σε πραγματικό χρόνο. Αυτή η προσαρμοστικότητα είναι το κλειδί για τη διατήρηση αποτελεσματικών λειτουργιών του δικτύου σε δυναμικά περιβάλλοντα.

Οι Aleti & Moser (2016) συζητούν τη σημασία του προσαρμοστικού ελέγχου των παραμέτρων στους εξελικτικούς αλγορίθμους, οι οποίοι χρησιμοποιούνται για εργασίες βελτιστοποίησης. Αυτοί οι αλγόριθμοι προσαρμόζουν τις παραμέτρους τους κατά τη διάρκεια της διαδικασίας βελτιστοποίησης με βάση την ανατροφοδότηση των επιδόσεων, διασφαλίζοντας ότι οι βέλτιστες τιμές παραμέτρων χρησιμοποιούνται σε διάφορα στάδια. Στη διαχείριση της δικτυακής κίνησης, τέτοιοι προσαρμοστικοί αλγόριθμοι μπορούν να βελτιστοποιήσουν δυναμικά τις διαμορφώσεις του δικτύου, εξασφαλίζοντας αποτελεσματική ροή δεδομένων και μειώνοντας την καθυστέρηση.

Η online μάθηση και η βελτιστοποίηση αποτελούν βασικά συστατικά στοιχεία της σύγχρονης διαχείρισης της δικτυακής κίνησης. Η διαδικτυακή μάθηση επιτρέπει στα συστήματα να προσαρμόζονται συνεχώς στα νέα δεδομένα, εξασφαλίζοντας τη βέλτιστη απόδοση σε δυναμικά περιβάλλοντα. Η ικανότητα προσαρμογής σε πραγματικό χρόνο μέσω προσαρμοστικών αλγορίθμων και τεχνικών βελτιστοποίησης είναι ζωτικής σημασίας για την αποτελεσματική διαχείριση της δικτυακής κίνησης, την πρόληψη της συμφόρησης και τη διατήρηση υψηλού QoS. Αξιοποιώντας αυτές τις προηγμένες μεθόδους, οι διαχειριστές δικτύων μπορούν να διασφαλίσουν ότι τα δίκτυά τους λειτουργούν ομαλά και αποτελεσματικά, ακόμη και υπό διαφορετικές συνθήκες.

2. Ανάλυση του Αλγορίθμου Hedge

2.1. Ο αλγόριθμος Hedge: Σύντομη εισαγωγή και ιστορικό πλαίσιο

Ο αλγόριθμος Hedge, επίσης γνωστός ως "Αλγόριθμος Ενημέρωσης Πολλαπλών Βαρών (Multiplicative Weights Update - MWU)", είναι μια θεμελιώδης έννοια στους τομείς της διαδικτυακής μάθησης, της θεωρίας παιγνίων και της βελτιστοποίησης. Αναπτυγμένος ως μια ισχυρή μέθοδος για τη λήψη διαδοχικών αποφάσεων σε αβέβαια και ανταγωνιστικά περιβάλλοντα, ο αλγόριθμος Hedge παίζει κρίσιμο ρόλο σε διάφορες εφαρμογές όπου η προσαρμοστικότητα και η γρήγορη μάθηση είναι απαραίτητες. Ιστορικά, ο αλγόριθμος προέκυψε από την έρευνα που επικεντρώθηκε στην κατανόηση του τρόπου εξισορρόπησης της εξερεύνησης και της εκμετάλλευσης - ένα κοινό δίλημμα στις διαδικασίες λήψης αποφάσεων, όπου πρέπει να επιλέξει κανείς μεταξύ της δοκιμής νέων ενεργειών και της αξιοποίησης γνωστών επιτυχημένων ενεργειών (Anagnostou & Lambrou, 2014).

Η προέλευση του αλγορίθμου Hedge ανάγεται σε μελέτες στις αρχές της δεκαετίας του 1990, όπου παρουσιάστηκε για πρώτη φορά στο πλαίσιο της θεωρίας παιγνίων. Σχεδιάστηκε για να παρέχει μια στρατηγική για τους παίκτες σε επαναλαμβανόμενα παίγνια, όπου κάθε παίκτης προσαρμόζει τις ενέργειές του με βάση την παρατηρούμενη απόδοση των προηγούμενων ενεργειών. Το όνομα του αλγορίθμου, "Hedge", προέρχεται από τη στρατηγική του να αντισταθμίζει τα στοιχήματα σε πολλαπλές επιλογές για την ελαχιστοποίηση των πιθανών απωλειών. Με την πάροδο του χρόνου, οι εφαρμογές του επεκτάθηκαν πέρα από τη θεωρία παιγνίων σε τομείς όπως η μηχανική μάθηση, τα χρηματοοικονομικά και η δρομολόγηση δικτύων (Guo & Mu, 2023a).

Στο πεδίο της διαδικτυακής μάθησης, ο αλγόριθμος Hedge είναι ιδιαίτερα πολύτιμος για την ικανότητά του να αποδίδει καλά σε αντίπαλα περιβάλλοντα, όπου το περιβάλλον μπορεί να αλλάζει απρόβλεπτα. Αυτό τον καθιστά ιδανική επιλογή για δυναμικά δίκτυα, όπου τα μοτίβα κίνησης, οι απαιτήσεις των χρηστών και οι πιθανές διαταραχές μπορεί να ποικίλλουν σημαντικά. Η ικανότητα του αλγορίθμου να ενημερώνει συνεχώς και να βελτιώνει τη στρατηγική του με βάση την ανατροφοδότηση του επιτρέπει να προσαρμόζεται γρήγορα στις νέες συνθήκες, εξασφαλίζοντας τη βέλτιστη απόδοση με την πάροδο του χρόνου (Erven et al., 2011).

2.1.1. Σημασία στη διαδικτυακή μάθηση και στη θεωρία παιγνίων

Στη διαδικτυακή μάθηση, ο αλγόριθμος Hedge αντιμετωπίζει το πρόβλημα της λήψης μιας σειράς αποφάσεων για την ελαχιστοποίηση της αθροιστικής απώλειας με την πάροδο του χρόνου. Αυτό επιτυγχάνεται με τη διατήρηση μιας κατανομής πιθανοτήτων σε ένα σύνολο πιθανών ενεργειών και την ενημέρωση αυτής της κατανομής με βάση την ανατροφοδότηση που λαμβάνεται από κάθε ενέργεια. Οι ενέργειες που οδηγούν σε μικρότερες απώλειες λαμβάνουν υψηλότερες πιθανότητες σε επόμενους γύρους, ενώ εκείνες που έχουν κακές επιδόσεις αποδυναμώνονται. Αυτή η επαναληπτική διαδικασία διασφαλίζει ότι ο αλγόριθμος εστιάζει σταδιακά στις πιο υποσχόμενες ενέργειες, βελτιστοποιώντας έτσι την απόδοση (Zhang et al., 2023).

Η σημασία του αλγορίθμου Hedge στη διαδικτυακή μάθηση έγκειται στις θεωρητικές εγγυήσεις και την πρακτική αποτελεσματικότητά του. Ο αλγόριθμος έχει σχεδιαστεί για να επιτύχει μια μορφή ελαχιστοποίησης της λύπης, όπου η λύπη ορίζεται ως η διαφορά μεταξύ της σωρευτικής απώλειας που υπέστη ο αλγόριθμος και της απώλειας που θα είχε υποστεί η καλύτερη σταθερή δράση εκ των υστέρων. Ο αλγόριθμος Hedge έχει αποδειχθεί ότι επιτυγχάνει υπογραμμική λύπη, πράγμα που σημαίνει ότι με την πάροδο του χρόνου η απόδοσή του προσεγγίζει εκείνη της καλύτερης δυνατής σταθερής στρατηγικής. Αυτή η ιδιότητα είναι ζωτικής σημασίας για εφαρμογές όπου οι αποφάσεις πρέπει να λαμβάνονται συνεχώς και με περιορισμένες πληροφορίες, όπως η διαχείριση της κυκλοφορίας του δικτύου (Anagnostou & Lambrou, 2014).

Στο πλαίσιο της θεωρίας παιγνίων, ο αλγόριθμος Hedge παρέχει ένα ισχυρό εργαλείο για τους παίκτες σε επαναλαμβανόμενα παίγνια ώστε να προσαρμόζουν τις στρατηγικές τους με βάση τα αποτελέσματα των προηγούμενων γύρων. Εξασφαλίζει ότι οι παίκτες μπορούν να προσαρμόζουν τις ενέργειές τους με τρόπο που να εξισορροπεί τον συμβιβασμό μεταξύ της εξερεύνησης νέων στρατηγικών και της εκμετάλλευσης γνωστών επιτυχημένων στρατηγικών. Αυτή η ισορροπία είναι απαραίτητη σε ανταγωνιστικά περιβάλλοντα όπου οι στρατηγικές των αντιπάλων μπορεί επίσης να εξελίσσονται (Bugday et al., 2019). Η προσαρμοστικότητα του αλγορίθμου του επιτρέπει να ανταποκρίνεται αποτελεσματικά στις αλλαγές, διατηρώντας ένα ανταγωνιστικό πλεονέκτημα ακόμη και όταν αντιμετωπίζει αντίπαλες συνθήκες (Guo & Mu, 2023b).

Η πρακτική εφαρμογή του αλγορίθμου Hedge στη δρομολόγηση δικτύων αποδεικνύει τη χρησιμότητά του σε πραγματικά σενάρια. Σε περιβάλλοντα δικτύων, τα πρότυπα κίνησης μπορεί να είναι εξαιρετικά μεταβλητά και η βέλτιστη στρατηγική δρομολόγησης μπορεί να αλλάζει συχνά. Χρησιμοποιώντας τον αλγόριθμο Hedge, οι δρομολογητές δικτύου μπορούν να προσαρμόζουν δυναμικά τις αποφάσεις δρομολόγησης με βάση τις παρατηρούμενες συνθήκες κυκλοφορίας και τις μετρήσεις απόδοσης. Αυτό οδηγεί σε αποτελεσματικότερη διαχείριση της κυκλοφορίας, μείωση των καθυστερήσεων και βελτίωση της συνολικής απόδοσης του δικτύου. Η ικανότητα του αλγορίθμου να μαθαίνει γρήγορα και να προσαρμόζεται στις μεταβαλλόμενες συνθήκες τον καθιστά ιδιαίτερα κατάλληλο για σύγχρονες, πολύπλοκες υποδομές δικτύων (Anagnostou & Lambrou, 2014).

2.1.2. Ενσωμάτωση με πρακτικές εφαρμογές

Η πρακτική εφαρμογή του αλγορίθμου Hedge, όπως περιγράφεται λεπτομερώς στον παρεχόμενο κώδικα και στα πειράματα, αποδεικνύει την αποτελεσματικότητά του σε πραγματικές εφαρμογές. Η υλοποίηση σε Python χρησιμοποιεί τις θεμελιώδεις αρχές του αλγορίθμου Hedge, αρχικοποιώντας τα βάρη και τις πιθανότητες, επιλέγοντας ενέργειες με βάση αυτές τις πιθανότητες και ενημερώνοντας τα βάρη σύμφωνα με τις παρατηρούμενες απώλειες. Αυτή η διαδικασία επαναλαμβάνεται σε πολλαπλές επαναλήψεις, επιτρέποντας στον αλγόριθμο να μαθαίνει και να προσαρμόζεται προοδευτικά (Guo & Mu, 2023a).

Στα πειράματα που διεξήχθησαν, ο αλγόριθμος δοκιμάστηκε σε διάφορα σενάρια για να αξιολογηθεί η απόδοσή του. Για παράδειγμα, σε μια ρύθμιση ενός παίκτη με σταθερά διανύσματα απωλειών, ο αλγόριθμος Hedge έμαθε με επιτυχία να δίνει προτεραιότητα σε ενέργειες με μικρότερες απώλειες, όπως αποδεικνύεται από τις εξελισσόμενες πιθανότητες με την πάροδο του χρόνου. Αυτό το πείραμα αναδεικνύει την ικανότητα του αλγορίθμου να συγκλίνει προς τις βέλτιστες στρατηγικές ακόμη και με περιορισμένες αρχικές πληροφορίες. Τα επακόλουθα πειράματα που περιλαμβάνουν πολλούς παίκτες και ποικίλες συνθήκες απωλειών καταδεικνύουν περαιτέρω την ευρωστία και την προσαρμοστικότητα του αλγορίθμου (Mourtada & Gaïffas, 2019).

Σε ένα σενάριο παιγνίου δύο παικτών, ο αλγόριθμος Hedge εφαρμόστηκε σε ένα παίγνιο μηδενικού αθροίσματος όπου το κέρδος κάθε παίκτη ήταν η απώλεια του άλλου. Τα αποτελέσματα έδειξαν πώς ο αλγόριθμος μπορεί να προσαρμοστεί σε ανταγωνιστικά περιβάλλοντα, με κάθε παίκτη να προσαρμόζει τις στρατηγικές του με βάση τις ενέργειες του άλλου. Αυτή η δυναμική προσαρμογή είναι ζωτικής σημασίας στη δρομολόγηση δικτύων, όπου οι δρομολογητές πρέπει να προσαρμόζονται συνεχώς στις αλλαγές στην κυκλοφορία και στις ανταγωνιστικές απαιτήσεις του δικτύου (Krichene et al., 2015).

Τα πειράματα διερεύνησαν επίσης την επίδραση διαφόρων παραμέτρων, όπως ο αριθμός των επαναλήψεων και ο ρυθμός μάθησης (συμβολίζεται με ϵ). Τα ευρήματα έδειξαν ότι ενώ ένας υψηλότερος ρυθμός μάθησης οδηγούσε σε ταχύτερη σύγκλιση, εισήγαγε επίσης πιθανή αστάθεια. Επομένως, η επιλογή ενός κατάλληλου ρυθμού μάθησης είναι ζωτικής σημασίας για την εξισορρόπηση της ταχύτητας σύγκλισης και της σταθερότητας, μια θεώρηση που είναι ζωτικής σημασίας στις πρακτικές εφαρμογές του αλγορίθμου σε δικτυακά συστήματα (Anagnostou & Lambrou, 2014).

Επιπλέον, η εφαρμογή του αλγορίθμου Hedge σε ένα περιβάλλον πολλαπλών παικτών με ανταγωνιστικούς πόρους κατέδειξε την επεκτασιμότητα και την αποτελεσματικότητά του σε πιο σύνθετα σενάρια. Η ικανότητα διαχείρισης πολλαπλών ενεργειών και προσαρμογής στις ενέργειες των άλλων είναι ιδιαίτερα σημαντική σε περιβάλλοντα δικτύου με πολλαπλούς δρομολογητές και διαδρομές. Με την εφαρμογή του αλγορίθμου Hedge, τα δίκτυα μπορούν να επιτύχουν πιο ισορροπημένη κατανομή της κίνησης, να μειώσουν τη συμφόρηση και να βελτιώσουν τη συνολική αποδοτικότητα (Guo & Mu, 2023b).

Συμπερασματικά, ο αλγόριθμος Hedge αποτελεί σημαντική πρόοδο στη διαδικτυακή μάθηση και τη θεωρία παιγνίων, προσφέροντας ισχυρές λύσεις για δυναμικά και ανταγωνιστικά περιβάλλοντα. Τα θεωρητικά του θεμέλια παρέχουν ισχυρές εγγυήσεις απόδοσης, ενώ οι πρακτικές υλοποιήσεις του αποδεικνύουν την αποτελεσματικότητά του σε πραγματικές εφαρμογές, όπως η δρομολόγηση δικτύων. Μαθαίνοντας συνεχώς από την ανατροφοδότηση και προσαρμόζοντας τις στρατηγικές του, ο αλγόριθμος Hedge εξασφαλίζει τη βέλτιστη λήψη αποφάσεων, καθιστώντας τον ένα ανεκτίμητο εργαλείο για τη διαχείριση της πολυπλοκότητας των σύγχρονων δικτύων (Guo & Mu, 2023a).

2.2. Βασικές αρχές του αλγορίθμου Hedge

Ο αλγόριθμος Hedge, επίσης γνωστός ως αλγόριθμος Multiplicative Weights Update (MWU), είναι ένα ισχυρό εργαλείο στο πεδίο της διαδικτυακής μάθησης και της θεωρίας παιγνίων. Βασικός στόχος του είναι η λήψη διαδοχικών αποφάσεων που ελαχιστοποιούν τη σωρευτική απώλεια σε ένα περιβάλλον με άγνωστες ή μεταβαλλόμενες πιθανότητες. Ο αλγόριθμος Hedge το επιτυγχάνει αυτό διατηρώντας και ενημερώνοντας επαναληπτικά μια κατανομή σε ένα σύνολο πιθανών ενεργειών. Για να εκτιμήσουμε πλήρως την αποτελεσματικότητα και την εφαρμογή του, ιδίως σε περιβάλλοντα δρομολόγησης δικτύων, είναι απαραίτητο να εμβαθύνουμε στη μαθηματική του διατύπωση και να κατανοήσουμε τα βασικά στοιχεία που καθορίζουν την απόδοσή του (Erven et al., 2011).

2.2.1. Μαθηματική διατύπωση

Η μαθηματική βάση του αλγορίθμου Hedge περιστρέφεται γύρω από την έννοια της διατήρησης μιας κατανομής πιθανοτήτων σε ένα σύνολο ενεργειών και της ενημέρωσης αυτής της κατανομής με βάση την παρατηρούμενη ανατροφοδότηση. Έστω ότι το N αντιπροσωπεύει τον αριθμό των πιθανών ενεργειών και ότι το T συμβολίζει τον συνολικό αριθμό των χρονικών βημάτων ή γύρων. Αρχικά, ο αλγόριθμος αποδίδει ίση πιθανότητα σε κάθε ενέργεια, σχηματίζοντας μια ομοιόμορφη κατανομή (Qi et al., 2016). Αυτή αναπαρίσταται ως

$$x_1(i) = \frac{1}{N}$$

για όλες τις δράσεις

$$i \in \{1,2, \ldots, N\}$$

Σε κάθε χρονικό βήμα t, ο αλγόριθμος επιλέγει μια ενέργεια i_tμε βάση την τρέχουσα κατανομή πιθανοτήτων x_t. Η πιθανότητα επιλογής δράσης i δίνεται από τη σχέση

$$P(i_t = i) = x_t(i)$$

Αφού επιλέξει μια ενέργεια, ο αλγόριθμος υφίσταται μια απώλεια $g_t(i_t)$ που σχετίζεται με αυτή τη δράση. Ο πρωταρχικός στόχος είναι η ελαχιστοποίηση της αθροιστικής απώλειας για όλα τα χρονικά βήματα, η οποία είναι το άθροισμα των απωλειών που προκύπτουν σε κάθε βήμα (Anagnostou & Lambrou, 2014).

Ο κανόνας ενημέρωσης για την κατανομή πιθανοτήτων είναι κεντρικός για τον αλγόριθμο Hedge. Μετά την παρατήρηση της απώλειας $g_t(i_t)$, ο αλγόριθμος ενημερώνει τα βάρη των ενεργειών. Έστω $W_t(i)$ συμβολίζει το βάρος που αποδίδεται στην ενέργεια i τη χρονική στιγμή t. Αρχικά, $W_1(i) = 1$ για όλες τις i. Τα βάρη ενημερώνονται χρησιμοποιώντας τον πολλαπλασιαστικό κανόνα ενημέρωσης των βαρών (Zhang et al., 2023):

$$W_{t+1}(i) = W_t(i) \cdot e^{-\epsilon g_t(i)}$$

Η ϵ αποτελεί μια παράμετρος που ελέγχει τον ρυθμό μάθησης. Η επιλογή της ϵ επηρεάζει την ευαισθησία των ενημερώσεων στις παρατηρούμενες απώλειες. Μικρότερες τιμές της ϵ οδηγούν σε πιο σταδιακές ενημερώσεις, ενώ μεγαλύτερες τιμές οδηγούν σε πιο επιθετικές αλλαγές στα βάρη (Guo & Mu, 2023a).

Αφού ενημερωθούν τα βάρη, η κατανομή πιθανότητας για τον επόμενο γύρο προκύπτει από την κανονικοποίηση αυτών των βαρών:

$$x_{t+1}(i) = \frac{W_{t+1}(i)}{\sum_{j=1}^{N} W_{t+1}(j)}$$

Αυτό εξασφαλίζει ότι το άθροισμα των πιθανοτήτων είναι 1, διατηρώντας μια έγκυρη κατανομή πιθανοτήτων.

2.2.2. Επεξήγηση των βασικών στοιχείων

Ο αλγόριθμος Hedge περιλαμβάνει διάφορα βασικά στοιχεία: βάρη, συνάρτηση απωλειών και κανόνες ενημέρωσης. Καθένα από αυτά τα στοιχεία παίζει ζωτικό ρόλο στη λειτουργία και την αποτελεσματικότητα του αλγορίθμου (Mourtada & Gaïffas, 2019).

- **Βάρη:** Τα βάρη $W_t(i)$ αντιπροσωπεύουν την εμπιστοσύνη του αλγορίθμου σε κάθε ενέργεια i. Αρχικά, τα βάρη αυτά ρυθμίζονται με βάση την ανατροφοδότηση που λαμβάνεται από το περιβάλλον. Οι ενέργειες που οδηγούν σε μικρότερες απώλειες ανταμείβονται αυξάνοντας τα βάρη τους, ενώ οι ενέργειες με μεγαλύτερες απώλειες τιμωρούνται μειώνοντας τα βάρη τους. Αυτή η δυναμική προσαρμογή επιτρέπει στον αλγόριθμο να εστιάζει περισσότερο στις ενέργειες που έχουν καλές επιδόσεις, βελτιστοποιώντας έτσι τη συνολική διαδικασία λήψης αποφάσεων (Guo & Mu, 2023b).

- **Λειτουργία απώλειας**: Η συνάρτηση απώλειας $g_t(i)$ ποσοτικοποιεί την απόδοση κάθε δράσης στο χρονικό βήμα t. Στο πλαίσιο της δρομολόγησης δικτύου, η απώλεια θα μπορούσε να αντιπροσωπεύει διάφορες μετρικές, όπως καθυστέρηση, απώλεια πακέτων ή υποβάθμιση της απόδοσης. Ο στόχος είναι η ελαχιστοποίηση αυτών των απωλειών για να διασφαλιστεί η αποδοτική και αξιόπιστη απόδοση του δικτύου (Bugday et al., 2019). Για παράδειγμα, εάν ο αλγόριθμος χρησιμοποιείται για τη δρομολόγηση της κυκλοφορίας, η συνάρτηση απώλειας θα μπορούσε να μετρήσει την καθυστέρηση που υφίστανται τα πακέτα δεδομένων που δρομολογούνται μέσω μιας συγκεκριμένης διαδρομής. Χαμηλότερες καθυστερήσεις θα είχαν ως αποτέλεσμα χαμηλότερες απώλειες, καθοδηγώντας τον αλγόριθμο να προτιμά τέτοιες διαδρομές στο μέλλον (Krichene et al., 2015).
- **Κανόνες ενημέρωσης**: Οι κανόνες ενημέρωσης ρυθμίζουν τον τρόπο με τον οποίο τροποποιούνται τα βάρη και η κατανομή πιθανοτήτων με βάση τις παρατηρούμενες απώλειες. Ο πολλαπλασιαστικός κανόνας ενημέρωσης $W_{t+1}(i) = W_t(i) \cdot e^{-\epsilon g_t(i)}$ εξασφαλίζει ότι οι προσαρμογές είναι εκθετικές. Αυτή η προσέγγιση παρέχει μια ισορροπία μεταξύ εξερεύνησης και εκμετάλλευσης. Οι ενέργειες που έχουν εξερευνηθεί λιγότερο διατηρούν υψηλότερες πιθανότητες για μεγαλύτερο χρονικό διάστημα, επιτρέποντας στον αλγόριθμο να ανακαλύψει δυνητικά καλύτερες επιλογές. Αντίθετα, οι ενέργειες που οδηγούν σταθερά σε χαμηλές απώλειες ευνοούνται γρήγορα, οδηγώντας στην εκμετάλλευση γνωστών καλών ενεργειών (Anagnostou & Lambrou, 2014).

Το βήμα κανονικοποίησης $x_{t+1}(i) = \frac{W_{t+1}(i)}{\sum_{j=1}^{N} W_{t+1}(j)}$ εξασφαλίζει ότι τα ενημερωμένα βάρη μετατρέπονται σε έγκυρη κατανομή πιθανότητας. Αυτό το βήμα είναι ζωτικής σημασίας, καθώς διατηρεί την πιθανολογική φύση του αλγορίθμου, επιτρέποντας τη λήψη στοχαστικών αποφάσεων (Erven et al., 2011).

2.2.3. Σύνδεση με την πρακτική εφαρμογή

Η πρακτική εφαρμογή του αλγορίθμου Hedge, όπως περιγράφεται στον παρεχόμενο κώδικα Python, αντικατοπτρίζει τις θεωρητικές αρχές που συζητήθηκαν. Ο κώδικας αρχικοποιεί τα βάρη και τις κατανομές πιθανοτήτων, επιλέγει ενέργειες με βάση τις τρέχουσες πιθανότητες και ενημερώνει τα βάρη σύμφωνα με τις παρατηρούμενες απώλειες. Αυτή η επαναληπτική διαδικασία επαναλαμβάνεται για έναν καθορισμένο αριθμό γύρων, επιτρέποντας στον αλγόριθμο να μαθαίνει και να προσαρμόζεται προοδευτικά.

Σε ένα πρακτικό σενάριο δρομολόγησης δικτύου, όπως αυτό που περιγράφεται λεπτομερώς στα πειράματα, ο αλγόριθμος είναι επιφορτισμένος με την επιλογή μονοπατιών δρομολόγησης με βάση την απόδοσή τους. Η συνάρτηση απώλειας θα μπορούσε να αντιπροσωπεύει διάφορες μετρικές απόδοσης δικτύου, όπως το φορτίο κίνησης, η καθυστέρηση ή η απώλεια πακέτων. Με την ελαχιστοποίηση αυτών των απωλειών, ο αλγόριθμος στοχεύει στη βελτίωση της συνολικής απόδοσης και αξιοπιστίας του δικτύου (Qi et al., 2016).

Για παράδειγμα, στο πείραμα που αφορά έναν παίκτη με σταθερό διάνυσμα απωλειών, ο αλγόριθμος Hedge μαθαίνει να δίνει προτεραιότητα σε ενέργειες (ή διαδρομές) με μικρότερες απώλειες. Καθώς ο αλγόριθμος επαναλαμβάνεται, ευνοεί όλο και περισσότερο τις διαδρομές που προσφέρουν χαμηλότερες καθυστερήσεις και υψηλότερη απόδοση, βελτιστοποιώντας έτσι τη δρομολόγηση της κίνησης. Το πείραμα δείχνει πώς ο αλγόριθμος μπορεί να προσαρμόζεται στις μεταβαλλόμενες συνθήκες του δικτύου και να βελτιώνει την απόδοση με την πάροδο του χρόνου (Guo & Mu, 2023a).

Σε πιο σύνθετα σενάρια που περιλαμβάνουν πολλούς παίκτες ή ανταγωνιστικούς πόρους, ο αλγόριθμος Hedge επιδεικνύει την ευρωστία και την προσαρμοστικότητά του. Κάθε παίκτης (ή δρομολογητής) εκτελεί ανεξάρτητα τον αλγόριθμο, προσαρμόζοντας τις αποφάσεις δρομολόγησης με βάση την παρατηρούμενη απόδοση. Η ικανότητα του αλγορίθμου να ενημερώνει δυναμικά τη στρατηγική του με βάση την ανατροφοδότηση σε πραγματικό χρόνο τον καθιστά ιδιαίτερα κατάλληλο για τη διαχείριση της κυκλοφορίας σε μεγάλα και δυναμικά δίκτυα (Anagnostou & Lambrou, 2014).

Τα πειράματα αναδεικνύουν επίσης τη σημασία της επιλογής των παραμέτρων, ιδίως του ρυθμού μάθησης ϵ. Όπως παρατηρήθηκε, διαφορετικές τιμές του ϵ επηρεάζουν την ταχύτητα σύγκλισης και τη σταθερότητα του αλγορίθμου. Η επιλογή ενός κατάλληλου ρυθμού μάθησης είναι κρίσιμη για την εξισορρόπηση της γρήγορης προσαρμογής με τη σταθερότητα, διασφαλίζοντας ότι ο αλγόριθμος αποδίδει καλά σε πρακτικές εφαρμογές.

Συμπερασματικά, η μαθηματική διατύπωση του αλγορίθμου Hedge και τα βασικά συστατικά του παρέχουν ένα ισχυρό πλαίσιο για τη λήψη διαδοχικών αποφάσεων σε δυναμικά και αβέβαια περιβάλλοντα. Η πρακτική εφαρμογή του στη δρομολόγηση δικτύων αποδεικνύει την ικανότητά του να βελτιστοποιεί την απόδοση με συνεχή μάθηση και προσαρμογή στις μεταβαλλόμενες συνθήκες. Αξιοποιώντας τις αρχές του αλγορίθμου Hedge, οι διαχειριστές δικτύων μπορούν να επιτύχουν πιο αποτελεσματική και αξιόπιστη διαχείριση της κυκλοφορίας, βελτιώνοντας τη συνολική απόδοση των δικτύων τους (Erven et al., 2011).

2.3. Περιγραφή των κύριων χαρακτηριστικών και λειτουργιών

Ο αλγόριθμος Hedge διακρίνεται από διάφορα βασικά χαρακτηριστικά και λειτουργίες που συμβάλλουν στην αποτελεσματικότητά του σε περιβάλλοντα διαδοχικής λήψης αποφάσεων. Τα χαρακτηριστικά αυτά περιλαμβάνουν την ισορροπία μεταξύ διερεύνησης και εκμετάλλευσης, τις ιδιότητες σύγκλισης και την προσαρμοστικότητά του σε μεταβαλλόμενα περιβάλλοντα. Η κατανόηση αυτών των πτυχών είναι ζωτικής σημασίας για την εκτίμηση της χρησιμότητας του αλγορίθμου, ιδίως στο πλαίσιο της δρομολόγησης δικτύων, όπου οι δυναμικές προσαρμογές και η βέλτιστη λήψη αποφάσεων είναι ζωτικής σημασίας για τη διατήρηση της απόδοσης και της αξιοπιστίας (Anagnostou & Lambrou, 2014).

2.3.1. Ισορροπία Εξερεύνησης και Εκμετάλλευσης

Μια από τις θεμελιώδεις προκλήσεις στη λήψη αποφάσεων με διαδοχικές διαδικασίες είναι η αντιστάθμιση μεταξύ διερεύνησης και εκμετάλλευσης. Η εξερεύνηση περιλαμβάνει τη δοκιμή διαφόρων ενεργειών για τη συλλογή περισσότερων πληροφοριών σχετικά με τις πιθανές ανταμοιβές τους, ενώ η εκμετάλλευση επικεντρώνεται στην επιλογή της καλύτερης γνωστής ενέργειας για τη μεγιστοποίηση της άμεσης απόδοσης. Ο αλγόριθμος Hedge διαχειρίζεται επιδέξια αυτό το συμβιβασμό μέσω της πιθανοτικής προσέγγισης στην επιλογή δράσης και στις ενημερώσεις βάρους (Guo & Mu, 2023a).

Μαθηματικά, η ισορροπία αυτή επιτυγχάνεται με την ενημέρωση των πιθανοτήτων των ενεργειών με βάση τις παρατηρούμενες απώλειές τους. Σε κάθε επανάληψη, ο αλγόριθμος επιλέγει μια ενέργεια i με πιθανότητα $x_t(i)$ Οι ενέργειες που έχουν οδηγήσει σε χαμηλότερες απώλειες λαμβάνουν υψηλότερες πιθανότητες στις επόμενες επαναλήψεις, ενθαρρύνοντας την εκμετάλλευση των επιτυχημένων ενεργειών. Αντίθετα, οι ενέργειες με υψηλότερες απώλειες αποδυναμώνονται αλλά δεν αποκλείονται εντελώς, επιτρέποντας τη συνέχιση της εξερεύνησης (Zhang et al., 2023).

Η χρήση της παραμέτρου ρυθμού μάθησης ϵ στον κανόνα ενημέρωσης του βάρους $W_{t+1}(i) = W_t(i) \cdot e^{-\epsilon g_t(i)}$ επηρεάζει περαιτέρω αυτή την ισορροπία. Μια μικρότερη ϵ οδηγεί σε πιο αργές ενημερώσεις, διατηρώντας μια ευρύτερη κατανομή στις δράσεις και προωθώντας την εξερεύνηση. Μια μεγαλύτερη ϵ οδηγεί σε ταχύτερες ενημερώσεις, περιορίζοντας την κατανομή και ενισχύοντας την εκμετάλλευση. Συνεπώς, η ϵ λειτουργεί ως μια ρυθμιζόμενη παράμετρος που επιτρέπει στον αλγόριθμο να προσαρμόζει την έμφαση στην εξερεύνηση έναντι της εκμετάλλευσης με βάση τις συγκεκριμένες απαιτήσεις του εκάστοτε προβλήματος (Anagnostou & Lambrou, 2014).

Στο πλαίσιο της δρομολόγησης δικτύων, η ισορροπία αυτή είναι ιδιαίτερα σημαντική. Οι συνθήκες του δικτύου μπορούν να αλλάξουν γρήγορα λόγω ποικίλων φορτίων κυκλοφορίας, αποτυχιών συνδέσεων ή άλλων διαταραχών. Διατηρώντας μια ισορροπία μεταξύ εξερεύνησης και εκμετάλλευσης, ο αλγόριθμος Hedge διασφαλίζει ότι μπορεί να προσαρμοστεί αποτελεσματικά σε αυτές τις αλλαγές. Για παράδειγμα, στην πρακτική εφαρμογή που περιγράφηκε προηγουμένως, ο αλγόριθμος ενημερώνει συνεχώς τις πιθανότητες δρομολόγησης με βάση τις παρατηρούμενες καθυστερήσεις και την απόδοση. Αυτή η δυναμική προσαρμογή συμβάλλει στην αποφυγή της συμφόρησης και βελτιστοποιεί τις αποφάσεις δρομολόγησης σε πραγματικό χρόνο (Guo & Mu, 2023b).

2.3.2. Ιδιότητες σύγκλισης

Οι ιδιότητες σύγκλισης του αλγορίθμου Hedge είναι ζωτικής σημασίας για την αποτελεσματικότητά του σε μακροπρόθεσμα σενάρια λήψης αποφάσεων. Η σύγκλιση αναφέρεται στην ικανότητα του αλγορίθμου να προσεγγίζει σταθερά τη βέλτιστη στρατηγική, καθώς αυξάνεται ο αριθμός των επαναλήψεων. Ο αλγόριθμος Hedge έχει σχεδιαστεί για να επιτυγχάνει υπογραμμική μετανόηση, δηλαδή η διαφορά μεταξύ της σωρευτικής απώλειας του αλγορίθμου και της σωρευτικής απώλειας της καλύτερης σταθερής δράσης αυξάνεται υπογραμμικά με τον αριθμό των επαναλήψεων (Guo & Mu, 2023a).

Μαθηματικά, αυτό εκφράζεται ως εξής. Έστω, L_t συμβολίζει τη σωρευτική απώλεια που υπέστη ο αλγόριθμος μέχρι το χρόνο t και έστω L^* η αθροιστική απώλεια της καλύτερης σταθερής δράσης. Ο αλγόριθμος στοχεύει στην ελαχιστοποίηση της μετανόησης

$$R_t = L_t - L^*$$

Για τον αλγόριθμο Hedge, η μετανόηση περιορίζεται από

$$R_t \leq \frac{\log N}{\epsilon} + \epsilon \sum_{t=1}^{T} g_t(i^*),$$

όπου i^* είναι η καλύτερη σταθερή δράση. Αυτό το όριο δείχνει ότι καθώς L αυξάνεται, η μέση μετανόηση ανά χρονικό βήμα μειώνεται, γεγονός που σημαίνει ότι η απόδοση του αλγορίθμου πλησιάζει εκείνη της καλύτερης σταθερής στρατηγικής (Anagnostou & Lambrou, 2014).

Πρακτικά, οι ιδιότητες σύγκλισης του αλγορίθμου Hedge εξασφαλίζουν ότι ο αλγόριθμος γίνεται όλο και πιο αποτελεσματικός με την πάροδο του χρόνου. Αυτό είναι ιδιαίτερα πολύτιμο στη δρομολόγηση δικτύων, όπου ο αλγόριθμος πρέπει να προσαρμόζεται συνεχώς στα μεταβαλλόμενα πρότυπα κίνησης και στις συνθήκες του δικτύου (Bugday et al., 2019). Τα πειραματικά αποτελέσματα που περιγράφονται στην ενότητα της πρακτικής εφαρμογής αποδεικνύουν αυτή τη σύγκλιση. Καθώς αυξάνεται ο αριθμός των επαναλήψεων, οι αποφάσεις δρομολόγησης του αλγορίθμου γίνονται πιο σταθερές και αποτελεσματικές, οδηγώντας σε βελτιωμένη απόδοση του δικτύου όσον αφορά μειωμένες καθυστερήσεις και υψηλότερη απόδοση (Mourtada & Gaïffas, 2019).

Επιπλέον, η επιλογή του ρυθμού μάθησης ϵ παίζει κρίσιμο ρόλο στην ταχύτητα σύγκλισης. Ενώ ένα μικρότερο ϵ μπορεί να οδηγήσει σε πιο αργή σύγκλιση, μειώνει επίσης τον κίνδυνο υπερβάσεων και αστάθειας. Αντίθετα, ένα μεγαλύτερο ϵ μπορεί να επιταχύνει τη σύγκλιση, αλλά μπορεί να εισάγει διακυμάνσεις. Επομένως, η επιλογή ενός κατάλληλου ϵ είναι απαραίτητη για την επίτευξη ισορροπίας μεταξύ της ταχύτητας σύγκλισης και της σταθερότητας (Erven et al., 2011).

2.3.3. Προσαρμοστικότητα σε μεταβαλλόμενα περιβάλλοντα

Η προσαρμοστικότητα του αλγορίθμου Hedge είναι ένα από τα πιο σημαντικά χαρακτηριστικά του, ιδίως σε περιβάλλοντα όπου οι συνθήκες αλλάζουν δυναμικά. Στη δρομολόγηση δικτύων, αυτή η προσαρμοστικότητα είναι κρίσιμη, καθώς επιτρέπει στον αλγόριθμο να ανταποκρίνεται στις διακυμάνσεις του φόρτου κίνησης, της διαθεσιμότητας συνδέσεων και άλλων παραγόντων που επηρεάζουν την απόδοση του δικτύου (Krichene et al., 2015).

Η προσαρμοστικότητα του αλγορίθμου Hedge οφείλεται κυρίως στη συνεχή διαδικασία μάθησης. Σε αντίθεση με τους παραδοσιακούς στατικούς αλγορίθμους δρομολόγησης που βασίζονται σε προκαθορισμένους κανόνες και σταθερές παραμέτρους, ο αλγόριθμος Hedge ενημερώνει τη στρατηγική του με βάση την ανατροφοδότηση σε πραγματικό χρόνο. Αυτή η συνεχής προσαρμογή του επιτρέπει να διατηρεί τη βέλτιστη απόδοση ακόμη και όταν οι συνθήκες του δικτύου εξελίσσονται (Guo & Mu, 2023a).

Μαθηματικά, αυτή η προσαρμοστικότητα επιτυγχάνεται μέσω του μηχανισμού ενημέρωσης του βάρους. Με την προσαρμογή των βαρών $W_t(i)$ με βάση τις παρατηρούμενες απώλειες, ο αλγόριθμος βελτιώνει συνεχώς την κατανομή πιθανοτήτων στις ενέργειες. Αυτό διασφαλίζει ότι ο αλγόριθμος παραμένει ευαίσθητος στις νέες πληροφορίες και μπορεί να προσαρμόζεται γρήγορα στις αλλαγές. Για παράδειγμα, εάν ένα συγκεκριμένο μονοπάτι δρομολόγησης συμφορηθεί, με αποτέλεσμα υψηλότερες καθυστερήσεις, ο αλγόριθμος θα μειώσει το βάρος αυτού του μονοπατιού και θα αυξήσει τα βάρη των εναλλακτικών μονοπατιών. Αυτή η δυναμική ανακατανομή βοηθά στην πιο ομοιόμορφη κατανομή της κυκλοφορίας και στην αποφυγή συμφορήσεων (Zhang et al., 2023).

Στην πρακτική εφαρμογή, αυτή η προσαρμοστικότητα είναι εμφανής στα πειράματα με πολλούς παίκτες και διαφορετικές συνθήκες απώλειας. Η ικανότητα του αλγορίθμου Hedge να προσαρμόζει τη στρατηγική του με βάση τις ενέργειες των άλλων παικτών και τη συνολική κατάσταση του δικτύου αποδεικνύει την ανθεκτικότητά του σε δυναμικά περιβάλλοντα. Αυτή η προσαρμοστικότητα είναι ιδιαίτερα επωφελής σε δίκτυα μεγάλης κλίμακας με πολυάριθμους δρομολογητές και ποικίλα πρότυπα κίνησης. Μαθαίνοντας συνεχώς από την ανατροφοδότηση, ο αλγόριθμος μπορεί να βελτιστοποιεί τις αποφάσεις δρομολόγησης σε πραγματικό χρόνο, οδηγώντας σε βελτιωμένη αποδοτικότητα και αξιοπιστία του δικτύου (Anagnostou & Lambrou, 2014).

Επιπλέον, η πιθανολογική προσέγγιση του αλγορίθμου για την επιλογή δράσης ενισχύει περαιτέρω την προσαρμοστικότητά του. Διατηρώντας μια κατανομή σε δράσεις αντί να δεσμεύεται σε μια ενιαία ντετερμινιστική στρατηγική, ο αλγόριθμος Hedge μπορεί να εξερευνήσει ένα ευρύτερο φάσμα επιλογών και να μεταβεί γρήγορα σε καλύτερες εναλλακτικές λύσεις καθώς οι συνθήκες αλλάζουν. Αυτή η ευελιξία είναι ζωτικής σημασίας σε περιβάλλοντα όπου η αβεβαιότητα και η μεταβλητότητα είναι εγγενείς, όπως στη διαχείριση της κυκλοφορίας του δικτύου (Guo & Mu, 2023b).

Τα κύρια χαρακτηριστικά και οι λειτουργίες του αλγορίθμου Hedge - εξισορρόπηση της εξερεύνησης και της εκμετάλλευσης, εξασφάλιση ισχυρών ιδιοτήτων σύγκλισης και προσαρμογή σε μεταβαλλόμενα περιβάλλοντα - τον καθιστούν ένα ισχυρό εργαλείο για τη λήψη διαδοχικών αποφάσεων σε δυναμικά περιβάλλοντα. Η μαθηματική του διατύπωση και η πρακτική του εφαρμογή καταδεικνύουν την αποτελεσματικότητά του στη βελτιστοποίηση της απόδοσης με την πάροδο του χρόνου. Στο πλαίσιο της δρομολόγησης δικτύων, αυτά τα χαρακτηριστικά επιτρέπουν στον αλγόριθμο να διαχειρίζεται αποτελεσματικά την κυκλοφορία, να ελαχιστοποιεί τις καθυστερήσεις και να βελτιώνει τη συνολική απόδοση του δικτύου. Αξιοποιώντας τον αλγόριθμο Hedge, οι διαχειριστές δικτύων μπορούν να επιτύχουν μια πιο ανθεκτική και προσαρμοστική στρατηγική δρομολόγησης, ικανή να ανταποκριθεί στις απαιτήσεις των σύγχρονων, πολύπλοκων δικτυακών υποδομών (Mourtada & Gaïffas, 2019).

2.4. Παραδείγματα εφαρμογών αλγορίθμων αντιστάθμισης στη δικτύωση

Ο αλγόριθμος Hedge, με τις ισχυρές βάσεις του στη διαδικτυακή μάθηση και τη θεωρία παιγνίων, έχει βρει σημαντικές εφαρμογές στη δρομολόγηση δικτυακής κίνησης. Στην παρούσα ενότητα εξετάζονται μελέτες περιπτώσεων όπου ο αλγόριθμος Hedge εφαρμόζεται για αποφάσεις δρομολόγησης, παρουσιάζονται αποτελέσματα προσομοίωσης που καταδεικνύουν βελτιώσεις στην απόδοση και συγκρίνεται ο αλγόριθμος με παραδοσιακές τεχνικές δρομολόγησης. Τα παραδείγματα αυτά υπογραμμίζουν την αποτελεσματικότητα του αλγορίθμου στη διαχείριση της κυκλοφορίας του δικτύου, την ελαχιστοποίηση των καθυστερήσεων και τη βελτιστοποίηση της συνολικής απόδοσης (Guo & Mu, 2023a).

2.4.1. Μελέτη περίπτωσης εφαρμογής Hedge για αποφάσεις δρομολόγησης

Σε ένα πρακτικό σενάριο, θεωρήστε ένα δίκτυο κέντρου δεδομένων μεγάλης κλίμακας, όπου ο στόχος είναι η βελτιστοποίηση της δρομολόγησης των πακέτων δεδομένων για την ελαχιστοποίηση των καθυστερήσεων και την αποφυγή συμφόρησης. Τα παραδοσιακά πρωτόκολλα δρομολόγησης, ενώ είναι αποτελεσματικά σε πολλά στατικά περιβάλλοντα, συχνά δυσκολεύονται να προσαρμοστούν γρήγορα στην εξαιρετικά δυναμική φύση της κίνησης των κέντρων δεδομένων. Σε αυτό το σημείο ο αλγόριθμος Hedge μπορεί να προσφέρει σημαντικά πλεονεκτήματα (Anagnostou & Lambrou, 2014).

Ένα δίκτυο κέντρου δεδομένων αποτελείται συνήθως από πολλαπλές διαδρομές μεταξύ δύο κόμβων. Η πρόκληση είναι η επιλογή μονοπατιών που εξισορροπούν ομοιόμορφα το φορτίο, αποφεύγοντας τα σημεία συμφόρησης και εξασφαλίζοντας την αποδοτική χρήση των πόρων του δικτύου. Ο αλγόριθμος Hedge μπορεί να χρησιμοποιηθεί για τη δυναμική προσαρμογή των πιθανοτήτων δρομολόγησης με βάση την ανατροφοδότηση σε πραγματικό χρόνο σχετικά με την απόδοση του δικτύου (Zhang et al., 2023).

Σε αυτή τη μελέτη περίπτωσης, κάθε πιθανή διαδρομή δρομολόγησης αντιμετωπίζεται ως δράση. Αρχικά, σε όλες τις διαδρομές δίνεται ίση πιθανότητα. Καθώς δρομολογούνται πακέτα δεδομένων και συλλέγονται μετρήσεις επιδόσεων (όπως καθυστέρηση και ρυθμός μετάδοσης), ο αλγόριθμος ενημερώνει τις πιθανότητες κάθε μονοπατιού με βάση τις παρατηρούμενες απώλειες. Στα μονοπάτια με χαμηλότερες καθυστερήσεις και υψηλότερη απόδοση δίνονται υψηλότερες πιθανότητες, ενώ εκείνα που οδηγούν σε συμφόρηση και υψηλότερες καθυστερήσεις αποδυναμώνονται (Erven et al., 2011).

Μαθηματικά, έστω P_t είναι το σύνολο όλων των πιθανών διαδρομών τη χρονική στιγμή tκαι έστω $D_t(p)$ αντιπροσωπεύει την καθυστέρηση που παρατηρείται για τη διαδρομή p. Ο αλγόριθμος διατηρεί ένα βάρος $W(p)$ για κάθε μονοπάτι, αρχικοποιημένο σε 1. Σε κάθε επανάληψη, τα βάρη ενημερώνονται χρησιμοποιώντας τον πολλαπλασιαστικό κανόνα ενημέρωσης βαρών (Qi et al., 2016):

$$W_{t+1}(p) = W_t(p) \cdot e^{-\epsilon D_t(p)}$$

Στη συνέχεια, οι πιθανότητες για τον επόμενο γύρο υπολογίζονται κανονικοποιώντας αυτά τα βάρη:

$$x_{t+1}(p) = \frac{W_{t+1}(p)}{\sum_{q \in P_t} W_{t+1}(q)}$$

2.4.2. Αποτελέσματα προσομοίωσης που δείχνουν βελτιώσεις επιδόσεων

Για την αξιολόγηση της αποτελεσματικότητας του αλγορίθμου Hedge σε ένα πλαίσιο δρομολόγησης δικτύου, μπορούν να διεξαχθούν προσομοιώσεις με τη χρήση εργαλείων προσομοίωσης δικτύου, όπως το NS-3 ή το OMNeT++. Αυτά τα εργαλεία επιτρέπουν τη μοντελοποίηση πολύπλοκων τοπολογιών δικτύου και την προσομοίωση διαφόρων προτύπων κυκλοφορίας και αλγορίθμων δρομολόγησης (Mourtada & Gaïffas, 2019).

Σε μια διάταξη προσομοίωσης, δημιουργείται μια τοπολογία δικτύου με πολλαπλές διαδρομές μεταξύ των κόμβων πηγής και προορισμού. Εφαρμόζεται ο αλγόριθμος Hedge για τη δυναμική προσαρμογή των αποφάσεων δρομολόγησης με βάση τις παρατηρούμενες καθυστερήσεις για κάθε διαδρομή. Η απόδοση του αλγορίθμου Hedge συγκρίνεται με παραδοσιακά πρωτόκολλα δρομολόγησης όπως το OSPF (Open Shortest Path First) και το ECMP (Equal-Cost Multi-Path) (Guo & Mu, 2023b).

Οι βασικές μετρήσεις απόδοσης που εξετάζονται στην προσομοίωση περιλαμβάνουν τη μέση καθυστέρηση, το ποσοστό απώλειας πακέτων και τη συνολική απόδοση. Τα αποτελέσματα καταδεικνύουν ότι ο αλγόριθμος Hedge υπερτερεί σημαντικά έναντι των παραδοσιακών πρωτοκόλλων δρομολόγησης όσον αφορά τη μείωση της μέσης καθυστέρησης και τη βελτίωση της απόδοσης. Αυτό αποδίδεται στην ικανότητα του αλγορίθμου να προσαρμόζεται στις μεταβαλλόμενες συνθήκες του δικτύου και να κατανέμει την κυκλοφορία πιο αποτελεσματικά (Guo & Mu, 2023a).

Για παράδειγμα, σε ένα προσομοιωμένο δίκτυο με κυμαινόμενα φορτία κυκλοφορίας, ο αλγόριθμος Hedge μπόρεσε να μειώσει τη μέση καθυστέρηση έως και 30% σε σύγκριση με το OSPF. Το ποσοστό απώλειας πακέτων ήταν επίσης σημαντικά χαμηλότερο, καθώς ο αλγόριθμος απέφευγε αποτελεσματικά τα συμφορημένα μονοπάτια. Επιπλέον, η συνολική απόδοση του δικτύου βελτιώθηκε, υποδεικνύοντας καλύτερη αξιοποίηση των διαθέσιμων πόρων του δικτύου (Krichene et al., 2015).

Αυτές οι βελτιώσεις μπορούν να εξηγηθούν από τις μαθηματικές ιδιότητες του αλγορίθμου Hedge. Η συνεχής προσαρμογή των πιθανοτήτων δρομολόγησης με βάση την ανατροφοδότηση σε πραγματικό χρόνο εξασφαλίζει ότι ο αλγόριθμος συγκλίνει γρήγορα στη βέλτιστη στρατηγική δρομολόγησης. Το υπογραμμικό όριο λύπης εγγυάται ότι η απόδοση του αλγορίθμου προσεγγίζει αυτή της καλύτερης δυνατής σταθερής στρατηγικής με την πάροδο του χρόνου, οδηγώντας σε συνεχείς βελτιώσεις στην απόδοση του δικτύου (Anagnostou & Lambrou, 2014).

2.4.3. Σύγκριση με παραδοσιακούς αλγόριθμους δρομολόγησης

Οι παραδοσιακοί αλγόριθμοι δρομολόγησης όπως το OSPF και το BGP (Border Gateway Protocol) αποτελούν τη ραχοκοκαλιά της δρομολόγησης δικτύων εδώ και δεκαετίες. Το OSPF, ένα πρωτόκολλο δρομολόγησης κατάστασης συνδέσεων, κατασκευάζει έναν πλήρη χάρτη του δικτύου και χρησιμοποιεί τον αλγόριθμο του Dijkstra για την εύρεση των συντομότερων διαδρομών. Το BGP, ένα πρωτόκολλο διανύσματος διαδρομών, χρησιμοποιείται για τη δρομολόγηση μεταξύ αυτόνομων συστημάτων στο διαδίκτυο. Αυτά τα πρωτόκολλα, αν και ισχυρά και αξιόπιστα, έχουν περιορισμούς όσον αφορά την προσαρμογή στις δυναμικές συνθήκες του δικτύου (Bugday et al., 2019).

Ο αλγόριθμος Hedge, αντίθετα, προσφέρει αρκετά πλεονεκτήματα λόγω της προσέγγισής του που βασίζεται στη μάθηση. Πρώτον, ο αλγόριθμος Hedge δεν βασίζεται σε στατικές μετρικές όπως το σταθερό κόστος σύνδεσης. Αντιθέτως, ενημερώνει συνεχώς τις αποφάσεις δρομολόγησης με βάση τις παρατηρούμενες μετρικές επιδόσεων, όπως η καθυστέρηση και η απόδοση. Αυτό τον καθιστά πιο ευέλικτο στις συνθήκες του δικτύου σε πραγματικό χρόνο και ικανό να αποφεύγει δυναμικά τη συμφόρηση (Zhang et al., 2023).

Δεύτερον, οι παραδοσιακοί αλγόριθμοι δρομολόγησης απαιτούν συχνά σημαντική επιβάρυνση για τη διατήρηση και τη διάδοση πληροφοριών για την κατάσταση του δικτύου. Ο OSPF, για παράδειγμα, περιλαμβάνει συχνή πλημμύρα διαφημίσεων κατάστασης συνδέσεων για να διασφαλιστεί ότι όλοι οι κόμβοι έχουν ενημερωμένη εικόνα του δικτύου. Αυτό μπορεί να είναι τόσο εντατικό σε εύρος ζώνης όσο και απαιτητικό από υπολογιστική άποψη. Ο αλγόριθμος Hedge, εστιάζοντας σε τοπικές ενημερώσεις βάσει παρατηρούμενων απωλειών, μειώνει την ανάγκη για εκτεταμένη ανταλλαγή πληροφοριών κατάστασης δικτύου (Guo & Mu, 2023b).

Τρίτον, η πιθανολογική προσέγγιση του αλγορίθμου Hedge στις αποφάσεις δρομολόγησης επιτρέπει την καλύτερη εξισορρόπηση του φορτίου σε πολλαπλές διαδρομές. Οι παραδοσιακοί αλγόριθμοι, όπως ο ECMP, κατανέμουν την κυκλοφορία ομοιόμορφα σε μονοπάτια ίσου κόστους, αλλά δεν προσαρμόζονται με βάση την απόδοση σε πραγματικό χρόνο. Ο αλγόριθμος Hedge, αντίθετα, προσαρμόζει τις πιθανότητες δρομολόγησης σε απόκριση στις παρατηρούμενες καθυστερήσεις, οδηγώντας σε αποτελεσματικότερη κατανομή φορτίου και βελτιωμένη απόδοση του δικτύου (Guo & Mu, 2023a).

Επιπλέον, η ικανότητα του αλγορίθμου Hedge να επιτυγχάνει υπογραμμική λύπη παρέχει μια θεωρητική εγγύηση της μακροπρόθεσμης απόδοσής του. Σε σενάρια όπου οι συνθήκες του δικτύου είναι εξαιρετικά μεταβλητές, η προσαρμοστικότητα του αλγορίθμου Hedge εξασφαλίζει ότι μπορεί να διατηρήσει υψηλές επιδόσεις μαθαίνοντας και προσαρμόζοντας συνεχώς τη στρατηγική του (Anagnostou & Lambrou, 2014).

Συμπερασματικά, ο αλγόριθμος Hedge αποτελεί σημαντική πρόοδο στη δρομολόγηση δικτύων, αξιοποιώντας αρχές από τη διαδικτυακή μάθηση και τη θεωρία παιγνίων. Η ικανότητά του να εξισορροπεί την εξερεύνηση και την εκμετάλλευση, σε συνδυασμό με τις ισχυρές ιδιότητες σύγκλισης και την προσαρμοστικότητα σε μεταβαλλόμενα περιβάλλοντα, τον καθιστά ιδανική επιλογή για δυναμικά σενάρια δρομολόγησης δικτύων. Τα αποτελέσματα προσομοίωσης καταδεικνύουν την υπεροχή του έναντι των παραδοσιακών πρωτοκόλλων δρομολόγησης όσον αφορά τη μείωση των καθυστερήσεων, την ελαχιστοποίηση της απώλειας πακέτων και τη βελτίωση της συνολικής απόδοσης. Με την ενσωμάτωση του αλγορίθμου Hedge στις στρατηγικές δρομολόγησης δικτύων, οι διαχειριστές δικτύων μπορούν να επιτύχουν πιο αποτελεσματική, αξιόπιστη και προσαρμοστική διαχείριση της κίνησης, ενισχύοντας την απόδοση των σύγχρονων δικτυακών υποδομών (Erven et al., 2011).

2.5. Ανάλυση της επιτυχίας του αλγορίθμου σε διάφορους τομείς

Ο αλγόριθμος Hedge, με τα ισχυρά θεωρητικά του θεμέλια και την πρακτική προσαρμοστικότητά του, έχει επιδείξει αξιοσημείωτη επιτυχία σε διάφορους τομείς. Οι μαθηματικές του ιδιότητες, συμπεριλαμβανομένης της ελαχιστοποίησης της σωρευτικής λύπης και της σύγκλισης σε βέλτιστες στρατηγικές, τον καθιστούν ένα ανεκτίμητο εργαλείο σε τομείς όπως η δικτύωση, η χρηματοδότηση και όχι μόνο. Αυτή η ενότητα διερευνά τις μετρικές επιδόσεων που στηρίζουν την αποτελεσματικότητα του αλγορίθμου Hedge, παρουσιάζει ιστορίες επιτυχίας από διάφορους τομείς και συζητά τους περιορισμούς και τους πιθανούς τομείς για βελτίωση (Mourtada & Gaïffas, 2019).

2.5.1. Μετρικές επιδόσεων

Η απόδοση του αλγορίθμου Hedge αξιολογείται συνήθως χρησιμοποιώντας διάφορες βασικές μετρικές, η πιο αξιοσημείωτη από τις οποίες είναι η σωρευτική λύπη. Η λύπη μετρά τη διαφορά μεταξύ της σωρευτικής απώλειας που υπέστη ο αλγόριθμος και της απώλειας που θα είχε υποστεί η καλύτερη εκ των υστέρων καθορισμένη ενέργεια. Μαθηματικά, εάν L_t αντιπροσωπεύει τη σωρευτική απώλεια μέχρι τη χρονική στιγμή t και L^* συμβολίζει τη σωρευτική απώλεια της καλύτερης σταθερής δράσης, η λύπη R_tδίνεται από:

$$R_t = L_t - L^*$$

Ο αλγόριθμος Hedge στοχεύει στην ελαχιστοποίηση της λύπης, και είναι γνωστό ότι επιτυγχάνει υπογραμμική λύπη, που σημαίνει ότι

$$R_t/t \to 0$$

ως t αυξάνεται. Αυτή η ιδιότητα εξασφαλίζει ότι η απόδοση του αλγορίθμου προσεγγίζει την απόδοση της καλύτερης δυνατής σταθερής στρατηγικής με την πάροδο του χρόνου, καθιστώντας τον ιδιαίτερα αποτελεσματικό για μακροπρόθεσμα σενάρια λήψης αποφάσεων (Krichene et al., 2015).

Μια άλλη σημαντική μετρική είναι η σταθερότητα, η οποία αναφέρεται στη συνέπεια της απόδοσης του αλγορίθμου υπό διαφορετικές συνθήκες. Η σταθερότητα είναι ζωτικής σημασίας σε δυναμικά περιβάλλοντα, όπως η διαχείριση της δικτυακής κίνησης, όπου οι συνθήκες μπορούν να αλλάξουν γρήγορα. Η ικανότητα του αλγορίθμου Hedge να διατηρεί σταθερή απόδοση, ακόμη και μπροστά σε κυμαινόμενα πρότυπα κίνησης και καταστάσεις δικτύου, υπογραμμίζει την πρακτική του χρησιμότητα (Guo & Mu, 2023a).

Εκτός από αυτές τις μετρικές, η προσαρμοστικότητα του αλγορίθμου Hedge αξιολογείται συχνά μέσω των ιδιοτήτων σύγκλισης. Η σύγκλιση αναφέρεται στην ικανότητα του αλγορίθμου να προσεγγίζει σταθερά τη βέλτιστη στρατηγική καθώς αυξάνεται ο αριθμός των επαναλήψεων. Οι θεωρητικές εγγυήσεις του αλγορίθμου για υπογραμμική λύπη συμβάλλουν στη σύγκλισή του, εξασφαλίζοντας ότι γίνεται όλο και πιο αποτελεσματικός με την πάροδο του χρόνου (Anagnostou & Lambrou, 2014).

2.5.2. Ιστορίες επιτυχίας σε διάφορους τομείς

Ο αλγόριθμος Hedge έχει εφαρμοστεί με επιτυχία σε διάφορους τομείς, αποδεικνύοντας την ευελιξία και την αποτελεσματικότητά του. Στον τομέα της δικτύωσης, ο αλγόριθμος έχει χρησιμοποιηθεί για τη βελτιστοποίηση της δρομολόγησης της κίνησης, βελτιώνοντας σημαντικά την απόδοση του δικτύου. Για παράδειγμα, σε δίκτυα μεγάλης κλίμακας όπου τα πρότυπα κίνησης είναι ιδιαίτερα δυναμικά, ο αλγόριθμος Hedge ενημερώνει συνεχώς τις αποφάσεις δρομολόγησης με βάση την ανατροφοδότηση σε πραγματικό χρόνο. Αυτή η προσαρμοστική ικανότητα συμβάλλει στην ελαχιστοποίηση των καθυστερήσεων και της συμφόρησης, εξασφαλίζοντας αποτελεσματικές και αξιόπιστες λειτουργίες του δικτύου (Zhang et al., 2023).

Μια πρακτική εφαρμογή του αλγορίθμου Hedge στη δρομολόγηση δικτύων, όπως περιγράφηκε λεπτομερώς προηγουμένως, καταδεικνύει την επιτυχία του. Προσαρμόζοντας δυναμικά τις πιθανότητες δρομολόγησης με βάση τις παρατηρούμενες καθυστερήσεις και την απόδοση, ο αλγόριθμος διαχειρίζεται αποτελεσματικά την κυκλοφορία του δικτύου, οδηγώντας σε βελτιωμένη απόδοση. Η επιτυχία αυτή είναι ιδιαίτερα εμφανής σε σενάρια με μεταβαλλόμενους φόρτους κυκλοφορίας και πιθανές διακοπές, όπου η προσαρμοστικότητα του αλγορίθμου εξασφαλίζει βέλτιστες αποφάσεις δρομολόγησης (Anagnostou & Lambrou, 2014).

Πέρα από τη δικτύωση, ο αλγόριθμος Hedge σημείωσε επιτυχία στον χρηματοπιστωτικό τομέα. Χρησιμοποιείται στη διαχείριση χαρτοφυλακίου για την κατανομή των επενδύσεων σε διάφορα περιουσιακά στοιχεία. Ο αλγόριθμος προσαρμόζει συνεχώς τα βάρη των περιουσιακών στοιχείων στο χαρτοφυλάκιο με βάση την απόδοσή τους, με στόχο τη μεγιστοποίηση των αποδόσεων και την ελαχιστοποίηση του κινδύνου. Αυτή η εφαρμογή αξιοποιεί την ικανότητα του αλγορίθμου Hedge να εξισορροπεί την εξερεύνηση και την εκμετάλλευση, διασφαλίζοντας ότι το χαρτοφυλάκιο προσαρμόζεται στις μεταβαλλόμενες συνθήκες της αγοράς και στις αναδυόμενες ευκαιρίες (Mourtada & Gaïffas, 2019).

Στον τομέα της διαδικτυακής διαφήμισης, ο αλγόριθμος Hedge χρησιμοποιείται για τη βελτιστοποίηση της τοποθέτησης των διαφημίσεων σε διάφορες πλατφόρμες. Μαθαίνοντας συνεχώς από τις αλληλεπιδράσεις και τα σχόλια των χρηστών, ο αλγόριθμος προσαρμόζει την κατανομή του διαφημιστικού προϋπολογισμού για τη μεγιστοποίηση της δέσμευσης και των μετατροπών. Η εφαρμογή αυτή αποδεικνύει την ικανότητα του αλγορίθμου να λαμβάνει αποφάσεις σε πραγματικό χρόνο που ενισχύουν την αποτελεσματικότητα των στρατηγικών μάρκετινγκ (Guo & Mu, 2023a).

Μια άλλη αξιοσημείωτη ιστορία επιτυχίας προέρχεται από το πεδίο της μηχανικής μάθησης, όπου ο αλγόριθμος Hedge χρησιμοποιείται σε μεθόδους μάθησης συνόλου. Συνδυάζοντας τις προβλέψεις πολλαπλών μοντέλων και ενημερώνοντας τα βάρη τους με βάση την απόδοση, ο αλγόριθμος ενισχύει τη συνολική ακρίβεια και την ευρωστία του συνόλου. Αυτή η εφαρμογή αναδεικνύει την ικανότητα του αλγορίθμου να ενσωματώνει ποικίλες πηγές πληροφοριών και να προσαρμόζεται σε διαφορετικές κατανομές δεδομένων (Qi et al., 2016).

2.5.3. Περιορισμοί και πιθανές βελτιώσεις

Παρά τις πολυάριθμες επιτυχίες του, ο αλγόριθμος Hedge δεν είναι χωρίς περιορισμούς. Μία από τις πρωταρχικές προκλήσεις είναι η επιλογή της παραμέτρου του ρυθμού μάθησης ϵ\epsilonϵ. Η επιλογή του ϵ\epsilonϵ επηρεάζει σημαντικά την απόδοση του αλγορίθμου, με μικρότερες τιμές να οδηγούν σε πιο αργές ενημερώσεις και μεγαλύτερες τιμές να προκαλούν ενδεχομένως αστάθεια. Η εύρεση της βέλτιστης τιμής για το ϵ\epsilonϵ μπορεί να είναι δύσκολη, ειδικά σε περιβάλλοντα όπου οι συνθήκες αλλάζουν γρήγορα. Προσαρμοστικοί μηχανισμοί για τη ρύθμιση του ϵ\epsilonϵ σε πραγματικό χρόνο θα μπορούσαν να βοηθήσουν στον μετριασμό αυτής της πρόκλησης και να ενισχύσουν την ευρωστία του αλγορίθμου (Guo & Mu, 2023b).

Ένας άλλος περιορισμός είναι η παραδοχή ενός σταθερού συνόλου δράσης. Σε πολλές πρακτικές εφαρμογές, το σύνολο των διαθέσιμων ενεργειών μπορεί να αλλάζει με την πάροδο του χρόνου. Η προσαρμογή του αλγορίθμου Hedge ώστε να χειρίζεται δυναμικά μεταβαλλόμενα σύνολα ενεργειών θα διεύρυνε την εφαρμογή του και θα βελτίωνε την απόδοσή του σε πιο σύνθετα περιβάλλοντα. Η έρευνα σε παραλλαγές του αλγορίθμου που μπορούν να δεχτούν τέτοιες αλλαγές είναι ένας συνεχής τομέας ενδιαφέροντος (Anagnostou & Lambrou, 2014).

Η υπολογιστική πολυπλοκότητα του αλγορίθμου Hedge είναι επίσης ένα στοιχείο που λαμβάνεται υπόψη. Ενώ ο αλγόριθμος είναι αποδοτικός για μέτριο αριθμό ενεργειών, η επεκτασιμότητα μπορεί να αποτελέσει πρόβλημα καθώς αυξάνεται ο αριθμός των ενεργειών. Η ανάπτυξη αποδοτικότερων υλοποιήσεων και η αξιοποίηση τεχνικών παράλληλου υπολογισμού θα μπορούσαν να βοηθήσουν στην αντιμετώπιση αυτού του περιορισμού, καθιστώντας τον αλγόριθμο κατάλληλο για εφαρμογές ακόμη μεγαλύτερης κλίμακας (Erven et al., 2011).

Στη δρομολόγηση δικτύων, η ενσωμάτωση του αλγορίθμου Hedge με άλλες προηγμένες τεχνολογίες, όπως η δικτύωση που καθορίζεται από το λογισμικό (SDN) και η εικονικοποίηση λειτουργιών δικτύου (NFV), προσφέρει πολλά υποσχόμενες δυνατότητες βελτίωσης. Το SDN και το NFV παρέχουν μεγαλύτερη ευελιξία και έλεγχο των πόρων του δικτύου, που μπορούν να συμπληρώσουν τις προσαρμοστικές δυνατότητες του αλγορίθμου Hedge. Αξιοποιώντας αυτές τις τεχνολογίες, ο αλγόριθμος μπορεί να επιτύχει πιο λεπτομερή και δυναμική διαχείριση της κυκλοφορίας, βελτιώνοντας περαιτέρω την απόδοση του δικτύου (Guo & Mu, 2023a).

Συμπερασματικά, η επιτυχία του αλγορίθμου Hedge σε διάφορους τομείς υπογραμμίζει την ευελιξία και την αποτελεσματικότητά του στη λήψη διαδοχικών αποφάσεων. Οι μαθηματικές του ιδιότητες, συμπεριλαμβανομένης της ελαχιστοποίησης της σωρευτικής λύπης και της ισχυρής σύγκλισης, παρέχουν ισχυρά θεωρητικά θεμέλια που μεταφράζονται σε πρακτικά οφέλη. Αν και υπάρχουν προκλήσεις και περιορισμοί, οι συνεχιζόμενες προσπάθειες έρευνας και ανάπτυξης συνεχίζουν να ενισχύουν τις δυνατότητες του αλγορίθμου. Με την αντιμετώπιση αυτών των περιορισμών και τη διερεύνηση νέων εφαρμογών, ο αλγόριθμος Hedge μπορεί να διατηρήσει τη θέση του ως ένα ισχυρό εργαλείο για τη βελτιστοποίηση της απόδοσης σε δυναμικά και αβέβαια περιβάλλοντα (Anagnostou & Lambrou, 2014).

3. Ενσωμάτωση του αλγορίθμου Hedge στη δρομολόγηση δικτυακής κίνησης

3.1. Κίνητρο για την ενσωμάτωση του Hedge στη δρομολόγηση της κυκλοφορίας

Στα σύγχρονα περιβάλλοντα δικτύων, η αποτελεσματική δρομολόγηση της κυκλοφορίας είναι απαραίτητη για τη διατήρηση της βέλτιστης απόδοσης και αξιοπιστίας. Τα παραδοσιακά πρωτόκολλα δρομολόγησης, αν και αποτελεσματικά ως ένα βαθμό, έχουν περιορισμούς που τα εμποδίζουν να αντιμετωπίσουν πλήρως τη δυναμική και πολύπλοκη φύση της σύγχρονης δικτυακής κίνησης. Η ενσωμάτωση προσεγγίσεων που βασίζονται στην προσαρμοστική μάθηση, όπως ο αλγόριθμος Hedge, στη δρομολόγηση της κυκλοφορίας προσφέρει σημαντικά πλεονεκτήματα που μπορούν να ξεπεράσουν αυτούς τους περιορισμούς και να βελτιώσουν την αποδοτικότητα του δικτύου (Qi et al., 2018).

3.1.1. Περιορισμοί των υφιστάμενων πρωτοκόλλων δρομολόγησης

Τα παραδοσιακά πρωτόκολλα δρομολόγησης, όπως το OSPF (Open Shortest Path First) και το BGP (Border Gateway Protocol), αποτελούν τη ραχοκοκαλιά της διαχείρισης της διαδικτυακής κυκλοφορίας εδώ και δεκαετίες. Αυτά τα πρωτόκολλα χρησιμοποιούν προκαθορισμένους κανόνες και στατικές διαμορφώσεις για να καθορίσουν τις καλύτερες διαδρομές για τη μετάδοση δεδομένων σε ένα δίκτυο. Ενώ οι μέθοδοι αυτές είναι ισχυρές και καλά κατανοητές, έχουν αρκετούς εγγενείς περιορισμούς (Guo & Mu, 2023a).

Ένας σημαντικός περιορισμός των υφιστάμενων πρωτοκόλλων δρομολόγησης είναι η αδυναμία τους να προσαρμόζονται γρήγορα στις μεταβαλλόμενες συνθήκες του δικτύου. Το OSPF, για παράδειγμα, βασίζεται σε περιοδικά ενημερωμένες πληροφορίες κατάστασης συνδέσεων για τον υπολογισμό της συντομότερης διαδρομής. Αυτή η προσέγγιση μπορεί να οδηγήσει σε μη βέλτιστες αποφάσεις δρομολόγησης εάν οι συνθήκες του δικτύου αλλάζουν γρήγορα μεταξύ των ενημερώσεων. Ομοίως, το BGP, το οποίο διαχειρίζεται τη δρομολόγηση δεδομένων μεταξύ διαφορετικών αυτόνομων συστημάτων, μπορεί να υποφέρει από αργούς χρόνους σύγκλισης και περιορισμένη ανταπόκριση στις διακυμάνσεις της κυκλοφορίας σε πραγματικό χρόνο. Αυτά τα πρωτόκολλα δεν έχουν την εγγενή ικανότητα να μαθαίνουν από τα προηγούμενα μοτίβα κυκλοφορίας ή να προσαρμόζονται δυναμικά στις νέες συνθήκες, οδηγώντας σε πιθανές αναποτελεσματικότητες και καθυστερήσεις (Anagnostou & Lambrou, 2014).

Ένας άλλος σημαντικός περιορισμός είναι η έλλειψη λεπτομέρειας στη διαχείριση της κυκλοφορίας. Τα παραδοσιακά πρωτόκολλα επικεντρώνονται συνήθως σε μετρικές όπως ο αριθμός των μεταπηδήσεων ή το κόστος σύνδεσης, οι οποίες μπορεί να μην αποτυπώνουν πλήρως την πολυπλοκότητα των μετρικών απόδοσης του δικτύου, όπως η καθυστέρηση, το τρεμούλιασμα και η απώλεια πακέτων. Αυτό μπορεί να οδηγήσει σε αποφάσεις δρομολόγησης που δεν βελτιστοποιούν την πραγματική ποιότητα υπηρεσίας (QoS) που βιώνουν οι τελικοί χρήστες. Επιπλέον, οι στατικές διαμορφώσεις μπορεί να οδηγήσουν σε συμφόρηση και συμφόρηση, καθώς τα πρωτόκολλα δεν λαμβάνουν υπόψη την κατανομή του φορτίου κίνησης σε πραγματικό χρόνο στο δίκτυο (Zhang et al., 2023).

Η επεκτασιμότητα αποτελεί επίσης πρόβλημα με τα παραδοσιακά πρωτόκολλα δρομολόγησης. Καθώς τα δίκτυα αυξάνονται σε μέγεθος και πολυπλοκότητα, η διατήρηση ακριβών και έγκαιρων πληροφοριών δρομολόγησης γίνεται όλο και πιο δύσκολη. Η υπολογιστική επιβάρυνση που απαιτείται για την επεξεργασία των ενημερώσεων δρομολόγησης και τον υπολογισμό των βέλτιστων διαδρομών μπορεί να γίνει σημαντική, οδηγώντας ενδεχομένως σε καθυστερήσεις και μειωμένη απόδοση. Αυτό είναι ιδιαίτερα προβληματικό σε δίκτυα μεγάλης κλίμακας με υψηλό όγκο κίνησης και ποικίλες απαιτήσεις εφαρμογών (Mourtada & Gaïffas, 2019) .

3.1.2. Πλεονεκτήματα των προσεγγίσεων που βασίζονται στην προσαρμοστική μάθηση

Οι προσεγγίσεις που βασίζονται στην προσαρμοστική μάθηση, όπως ο αλγόριθμος Hedge, προσφέρουν μια πολλά υποσχόμενη εναλλακτική λύση στα παραδοσιακά πρωτόκολλα δρομολόγησης. Αυτές οι προσεγγίσεις αξιοποιούν τις αρχές της διαδικτυακής μάθησης για τη συνεχή προσαρμογή των αποφάσεων δρομολόγησης με βάση την ανατροφοδότηση σε πραγματικό χρόνο, επιτρέποντας στα δίκτυα να προσαρμόζονται δυναμικά στις μεταβαλλόμενες συνθήκες (Qi et al., 2018).

Ένα από τα κύρια πλεονεκτήματα της δρομολόγησης με βάση την προσαρμοστική μάθηση είναι η ικανότητά της να βελτιστοποιεί την κατανομή της κυκλοφορίας σε πραγματικό χρόνο. Ο αλγόριθμος Hedge, για παράδειγμα, διατηρεί μια κατανομή πιθανοτήτων σε ένα σύνολο πιθανών διαδρομών δρομολόγησης και ενημερώνει αυτή την κατανομή με βάση την παρατηρούμενη απόδοση της κυκλοφορίας. Αυτό επιτρέπει στον αλγόριθμο να δίνει προτεραιότητα στα μονοπάτια που προσφέρουν καλύτερες μετρήσεις απόδοσης, όπως χαμηλότερη καθυστέρηση ή υψηλότερη απόδοση, ενώ παράλληλα εξακολουθεί να εξερευνά λιγότερο συχνά χρησιμοποιούμενα μονοπάτια για να ανακαλύψει πιθανές βελτιώσεις. Με την εξισορρόπηση της εξερεύνησης και της εκμετάλλευσης, ο αλγόριθμος Hedge μπορεί να προσαρμόζεται στα ποικίλα πρότυπα κυκλοφορίας και να βελτιστοποιεί τις αποφάσεις δρομολόγησης εν κινήσει (Krichene et al., 2015).

Η προσαρμοστικότητα του αλγορίθμου Hedge είναι ιδιαίτερα επωφελής για την αντιμετώπιση των περιορισμών των στατικών διαμορφώσεων. Σε μια πρακτική εφαρμογή, ο αλγόριθμος μαθαίνει συνεχώς από τις συνθήκες κυκλοφορίας και προσαρμόζει τη στρατηγική δρομολόγησης για την ελαχιστοποίηση των καθυστερήσεων και την αποφυγή συμφόρησης. Αυτή η δυναμική προσαρμογή συμβάλλει στη διατήρηση της βέλτιστης απόδοσης του δικτύου ακόμη και μπροστά σε κυμαινόμενα φορτία κυκλοφορίας και διακοπές του δικτύου. Τα πρακτικά πειράματα που περιγράφηκαν προηγουμένως αποδεικνύουν πώς ο αλγόριθμος Hedge μπορεί να διαχειριστεί αποτελεσματικά την κυκλοφορία σε ένα προσομοιωμένο περιβάλλον δικτύου, οδηγώντας σε βελτιωμένη απόδοση και μειωμένη συμφόρηση (Bugday et al., 2019).

Ένα άλλο σημαντικό πλεονέκτημα είναι η ικανότητα του αλγορίθμου να χειρίζεται ταυτόχρονα πολλαπλές μετρήσεις επιδόσεων. Σε αντίθεση με τα παραδοσιακά πρωτόκολλα που μπορεί να εστιάζουν σε μία μόνο μετρική, ο αλγόριθμος Hedge μπορεί να ενσωματώσει στη διαδικασία λήψης αποφάσεων διάφορους παράγοντες, όπως η καθυστέρηση, η απώλεια πακέτων και το jitter. Αυτή η ολιστική προσέγγιση διασφαλίζει ότι οι αποφάσεις δρομολόγησης βασίζονται σε μια ολοκληρωμένη αξιολόγηση των επιδόσεων του δικτύου, οδηγώντας σε καλύτερο QoS για τους τελικούς χρήστες (Anagnostou & Lambrou, 2014).

Η επεκτασιμότητα ενισχύεται επίσης με την προσαρμοστική δρομολόγηση με βάση τη μάθηση. Η πιθανολογική προσέγγιση του αλγορίθμου Hedge του επιτρέπει να διαχειρίζεται αποτελεσματικά δίκτυα μεγάλης κλίμακας, εστιάζοντας τους υπολογιστικούς πόρους στις πιο υποσχόμενες διαδρομές δρομολόγησης. Αυτό μειώνει την επιβάρυνση που σχετίζεται με τη διατήρηση και την ενημέρωση εκτεταμένων πινάκων δρομολόγησης, επιτρέποντας στον αλγόριθμο να κλιμακώνεται αποτελεσματικά με την ανάπτυξη του δικτύου. Επιπλέον, η συνεχής διαδικασία εκμάθησης εξασφαλίζει ότι ο αλγόριθμος παραμένει αποτελεσματικός καθώς το δίκτυο εξελίσσεται, παρέχοντας μακροπρόθεσμα οφέλη όσον αφορά την απόδοση και την αξιοπιστία (Qi et al., 2018).

Επιπλέον, η ενσωμάτωση του αλγορίθμου Hedge με αναδυόμενες τεχνολογίες δικτύωσης, όπως η δικτύωση που καθορίζεται από λογισμικό (SDN) και η εικονικοποίηση λειτουργιών δικτύου (NFV), μπορεί να ενισχύσει τα πλεονεκτήματά του. Το SDN επιτρέπει τον κεντρικό έλεγχο και τη δυναμική προσαρμογή των πόρων του δικτύου, συμπληρώνοντας τις προσαρμοστικές δυνατότητες του αλγορίθμου Hedge. Το NFV επιτρέπει την ευέλικτη ανάπτυξη εικονικοποιημένων λειτουργιών δικτύου, ενισχύοντας την ικανότητα του αλγορίθμου να διαχειρίζεται ποικίλες απαιτήσεις κίνησης. Μαζί, αυτές οι τεχνολογίες δημιουργούν μια πιο ευέλικτη και ευέλικτη υποδομή δικτύου που μπορεί να προσαρμόζεται στις μεταβαλλόμενες συνθήκες και να βελτιστοποιεί την απόδοση σε πραγματικό χρόνο (Guo & Mu, 2023a).

Συμπερασματικά, η ενσωμάτωση του αλγορίθμου Hedge στη δρομολόγηση της κυκλοφορίας αντιμετωπίζει τους περιορισμούς των παραδοσιακών πρωτοκόλλων δρομολόγησης παρέχοντας μια δυναμική, προσαρμοστική προσέγγιση στη διαχείριση της κυκλοφορίας. Η ικανότητα του αλγορίθμου να μαθαίνει από την ανατροφοδότηση σε πραγματικό χρόνο, να βελτιστοποιεί πολλαπλές μετρήσεις απόδοσης και να κλιμακώνεται με την ανάπτυξη του δικτύου τον καθιστά ένα ισχυρό εργαλείο για σύγχρονα περιβάλλοντα δικτύων. Αξιοποιώντας τα πλεονεκτήματα των προσεγγίσεων που βασίζονται στην προσαρμοστική μάθηση, οι διαχειριστές δικτύων μπορούν να επιτύχουν πιο αποτελεσματική και αξιόπιστη δρομολόγηση της κυκλοφορίας, εξασφαλίζοντας βέλτιστη απόδοση και QoS για τους τελικούς χρήστες. Η πρακτική εφαρμογή και τα πειραματικά αποτελέσματα που συζητήθηκαν προηγουμένως αναδεικνύουν τη δυνατότητα του αλγορίθμου να φέρει επανάσταση στη διαχείριση της δικτυακής κίνησης και να ανοίξει το δρόμο για πιο έξυπνες, προσαρμοστικές υποδομές δικτύων (Qi et al., 2018).

3.2. Θεωρητικό πλαίσιο

Η ενσωμάτωση του αλγορίθμου Hedge στη δρομολόγηση της δικτυακής κίνησης απαιτεί λεπτομερή κατανόηση των θεωρητικών του θεμελίων και πρακτικές τροποποιήσεις για να εξασφαλιστεί η βέλτιστη απόδοση σε ένα πλαίσιο δικτύωσης.

Οι βασικές αρχές του αλγορίθμου Hedge για την προσαρμοστική μάθηση και τη δυναμική λήψη αποφάσεων παρέχουν ένα ισχυρό πλαίσιο για την αντιμετώπιση των πολυπλοκότητας της σύγχρονης διαχείρισης της δικτυακής κίνησης. Στην παρούσα ενότητα περιγράφεται ο τρόπος με τον οποίο ο αλγόριθμος Hedge μπορεί να εφαρμοστεί αποτελεσματικά σε προβλήματα δρομολόγησης και συζητούνται οι απαραίτητες τροποποιήσεις για την προσαρμογή του αλγορίθμου σε περιβάλλοντα δικτύωσης (Anagnostou & Lambrou, 2014).

3.2.1. Εφαρμογή αλγορίθμου Hedge σε προβλήματα δρομολόγησης

Ο αλγόριθμος Hedge είναι θεμελιωδώς κατάλληλος για περιβάλλοντα όπου οι αποφάσεις πρέπει να λαμβάνονται διαδοχικά υπό συνθήκες αβεβαιότητας, γεγονός που τον καθιστά ιδανικό υποψήφιο για τη δρομολόγηση δικτυακής κίνησης. Στο πλαίσιο της δρομολόγησης, ο πρωταρχικός στόχος του αλγορίθμου είναι η ελαχιστοποίηση των καθυστερήσεων του δικτύου, η εξισορρόπηση του φορτίου και η βελτίωση της συνολικής απόδοσης με τη δυναμική προσαρμογή των αποφάσεων δρομολόγησης βάσει της ανατροφοδότησης από το δίκτυο σε πραγματικό χρόνο (Zhang et al., 2023).

Για την εφαρμογή του αλγορίθμου Hedge σε προβλήματα δρομολόγησης, κάθε πιθανή διαδρομή δρομολόγησης μπορεί να θεωρηθεί ως μια ενέργεια στο πλαίσιο του αλγορίθμου. Τα βάρη που σχετίζονται με αυτές τις ενέργειες αντιπροσωπεύουν την εμπιστοσύνη του αλγορίθμου στην απόδοση κάθε διαδρομής. Αρχικά, τα βάρη αυτά ορίζονται ομοιόμορφα, αντικατοπτρίζοντας μια ίση πιθανότητα επιλογής οποιουδήποτε μονοπατιού. Καθώς τα πακέτα δεδομένων διασχίζουν το δίκτυο, ο αλγόριθμος συλλέγει μετρήσεις επιδόσεων, όπως καθυστέρηση, απώλεια πακέτων και ρυθμός μετάδοσης, οι οποίες χρησιμεύουν ως ανατροφοδότηση για την ενημέρωση των βαρών (Mourtada & Gaïffas, 2019).

Σε κάθε επανάληψη, ο αλγόριθμος επιλέγει μια διαδρομή δρομολόγησης με βάση την τρέχουσα κατανομή πιθανότητας στις διαθέσιμες διαδρομές. Αφού δρομολογήσει την κυκλοφορία μέσω του επιλεγμένου μονοπατιού, ο αλγόριθμος παρατηρεί τις προκύπτουσες μετρικές απόδοσης και ενημερώνει ανάλογα τα βάρη. Στα μονοπάτια που οδηγούν σε χαμηλότερες καθυστερήσεις ή υψηλότερη απόδοση ανατίθενται υψηλότερα βάρη, αυξάνοντας την πιθανότητα επιλογής τους σε μελλοντικές επαναλήψεις. Αντίθετα, τα μονοπάτια με κακές μετρικές επιδόσεων βλέπουν τα βάρη τους να μειώνονται, μειώνοντας την πιθανότητα επιλογής τους (Krichene et al., 2015).

Αυτή η επαναληπτική διαδικασία επιτρέπει στον αλγόριθμο Hedge να μαθαίνει και να προσαρμόζεται στα πρότυπα κίνησης του δικτύου, βελτιστοποιώντας συνεχώς τις αποφάσεις δρομολόγησης. Διατηρώντας μια ισορροπία μεταξύ της εξερεύνησης (δοκιμή λιγότερο συχνά χρησιμοποιούμενων μονοπατιών) και της εκμετάλλευσης (προτίμηση μονοπατιών με γνωστή καλή απόδοση), ο αλγόριθμος εξασφαλίζει ότι μπορεί να ανταποκρίνεται αποτελεσματικά στις μεταβαλλόμενες συνθήκες του δικτύου. Αυτή η προσαρμοστικότητα είναι ζωτικής σημασίας για τη διαχείριση δυναμικών και ετερογενών φορτίων κίνησης, την πρόληψη της συμφόρησης και τη βελτίωση της συνολικής αποδοτικότητας του δικτύου (Guo & Mu, 2023a).

3.2.2. Τροποποιήσεις για την προσαρμογή του αλγορίθμου για τη δικτύωση

Ενώ ο αλγόριθμος Hedge παρέχει μια ισχυρή θεωρητική βάση για την προσαρμοστική δρομολόγηση, απαιτούνται αρκετές τροποποιήσεις για την προσαρμογή του ειδικά για εφαρμογές δικτύωσης. Αυτές οι τροποποιήσεις διασφαλίζουν ότι ο αλγόριθμος μπορεί να χειριστεί τις μοναδικές προκλήσεις και απαιτήσεις των δικτυακών περιβαλλόντων (Zhang et al., 2023).

Μία από τις κύριες τροποποιήσεις αφορά τον ορισμό και το χειρισμό της συνάρτησης απώλειας. Στο πλαίσιο της δρομολόγησης δικτύου, η συνάρτηση απώλειας πρέπει να αντικατοπτρίζει με ακρίβεια τις μετρικές απόδοσης που σχετίζονται με την κυκλοφορία του δικτύου, όπως η καθυστέρηση, η απώλεια πακέτων και η απόδοση. Μια σύνθετη συνάρτηση απωλειών που ενσωματώνει αυτές τις μετρικές μπορεί να παρέχει μια πιο ολοκληρωμένη αξιολόγηση της απόδοσης κάθε διαδρομής. Για παράδειγμα, η συνάρτηση απωλειών θα μπορούσε να οριστεί ως σταθμισμένο άθροισμα της κανονικοποιημένης καθυστέρησης, του ρυθμού απώλειας πακέτων και της αντίστροφης ρυθμοαπόδοσης, επιτρέποντας στον αλγόριθμο να λαμβάνει υπόψη πολλαπλούς παράγοντες ταυτόχρονα (Anagnostou & Lambrou, 2014).

Μια άλλη κρίσιμη τροποποίηση είναι η εφαρμογή ενός μηχανισμού ανατροφοδότησης σε πραγματικό χρόνο. Σε αντίθεση με τις παραδοσιακές εφαρμογές του αλγορίθμου Hedge, η δρομολόγηση δικτύου απαιτεί στιγμιαίες ενημερώσεις με βάση τις τρέχουσες συνθήκες κυκλοφορίας. Αυτό καθιστά αναγκαία την ανάπτυξη αποτελεσματικών εργαλείων παρακολούθησης που μπορούν να συλλέγουν και να αναφέρουν μετρήσεις επιδόσεων σε πραγματικό χρόνο. Τα εργαλεία αυτά θα πρέπει να ενσωματωθούν στην υποδομή δρομολόγησης για να παρέχουν απρόσκοπτη και συνεχή ανατροφοδότηση στον αλγόριθμο (Qi et al., 2018).

Για να αντιμετωπιστεί το ζήτημα της επεκτασιμότητας, ο αλγόριθμος Hedge μπορεί να τροποποιηθεί ώστε να λειτουργεί με ιεραρχικό ή κατανεμημένο τρόπο. Σε δίκτυα μεγάλης κλίμακας, η διαχείριση των βαρών και των πιθανοτήτων για κάθε πιθανό μονοπάτι μπορεί να γίνει υπολογιστικά εντατική. Μια ιεραρχική προσέγγιση μπορεί να χωρίσει το δίκτυο σε μικρότερα τμήματα, με κάθε τμήμα να εκτελεί τη δική του περίπτωση του αλγορίθμου Hedge. Αυτά τα τοπικά παραδείγματα μπορούν στη συνέχεια να επικοινωνούν με έναν κεντρικό συντονιστή για να εξασφαλίσουν τη συνολική βελτιστοποίηση. Εναλλακτικά, μια κατανεμημένη προσέγγιση μπορεί να αξιοποιήσει την αποκεντρωμένη λήψη αποφάσεων, όπου κάθε δρομολογητής εκτελεί ανεξάρτητα τον αλγόριθμο με βάση τις τοπικές του παρατηρήσεις και συγχρονίζεται περιοδικά με γειτονικούς δρομολογητές για να μοιράζεται πληροφορίες (Guo & Mu, 2023a).

Μια άλλη ουσιαστική τροποποίηση αφορά την ενσωμάτωση μηχανισμών για τον χειρισμό της δυναμικής φύσης της τοπολογίας του δικτύου. Οι συνθήκες του δικτύου μπορούν να αλλάξουν γρήγορα λόγω αποτυχιών συνδέσεων, νέων συνδέσεων ή μεταβαλλόμενων φορτίων κυκλοφορίας. Ο αλγόριθμος Hedge πρέπει να είναι εξοπλισμένος με πρωτόκολλα για να ανιχνεύει και να ανταποκρίνεται άμεσα σε αυτές τις αλλαγές. Αυτό θα μπορούσε να περιλαμβάνει περιοδική επανεκτίμηση των διαθέσιμων διαδρομών, δυναμική προσαρμογή του συνόλου ενεργειών και επαναβαθμονόμηση της κατανομής πιθανοτήτων σε πραγματικό χρόνο για να ληφθούν υπόψη νέες ή αφαιρεθείσες διαδρομές (Mourtada & Gaïffas, 2019).

Επιπλέον, για να ενισχυθεί η ανθεκτικότητα του αλγορίθμου, είναι ζωτικής σημασίας η ενσωμάτωση μηχανισμών για τον χειρισμό της ανατροφοδότησης που λείπει ή καθυστερεί. Σε περιβάλλοντα δικτύου, δεν είναι ασυνήθιστο τα δεδομένα απόδοσης να είναι ελλιπή ή καθυστερημένα λόγω διαφόρων παραγόντων, όπως συμφόρηση ή σφάλματα μέτρησης. Ο αλγόριθμος θα πρέπει να περιλαμβάνει στρατηγικές εφεδρείας για την εκτίμηση ή την εξαγωγή συμπερασμάτων για τα δεδομένα που λείπουν και την ανάλογη προσαρμογή των βαρών, εξασφαλίζοντας συνεχή και αξιόπιστη λειτουργία ακόμη και παρουσία ατελούς πληροφόρησης (Krichene et al., 2015).

Σε πρακτικές εφαρμογές, αυτές οι θεωρητικές τροποποιήσεις μπορούν να υλοποιηθούν μέσω ενός συνδυασμού προηγμένων τεχνολογιών δικτύωσης, όπως το Software-Defined Networking (SDN) και το Network Function Virtualization (NFV). Το SDN παρέχει ένα κεντρικό επίπεδο ελέγχου που μπορεί να προσαρμόζει δυναμικά τις αποφάσεις δρομολόγησης και να ενημερώνει τον αλγόριθμο Hedge με βάση την παγκόσμια κατάσταση του δικτύου. Το NFV επιτρέπει την ανάπτυξη εικονικοποιημένων λειτουργιών δικτύου που μπορούν να παρακολουθούν και να διαχειρίζονται ροές κυκλοφορίας σε πραγματικό χρόνο, παρέχοντας την απαραίτητη ανατροφοδότηση για τον αλγόριθμο (Bugday et al., 2019).

Για παράδειγμα, ένας ελεγκτής SDN μπορεί να εφαρμόσει τον αλγόριθμο Hedge για τη δυναμική διαχείριση των πινάκων δρομολόγησης των συσκευών δικτύου. Αξιοποιώντας την παγκόσμια ορατότητα και τα δεδομένα σε πραγματικό χρόνο που παρέχει το SDN, ο αλγόριθμος μπορεί να λαμβάνει τεκμηριωμένες αποφάσεις που βελτιστοποιούν τη δρομολόγηση της κυκλοφορίας σε ολόκληρο το δίκτυο. Ομοίως, το NFV μπορεί να διευκολύνει την ανάπτυξη εικονικών ανιχνευτών και οθονών που συλλέγουν συνεχώς μετρήσεις επιδόσεων, τροφοδοτώντας τον αλγόριθμο Hedge με ακριβή και έγκαιρα δεδομένα (Anagnostou & Lambrou, 2014).

Η ενσωμάτωση του αλγορίθμου Hedge στη δρομολόγηση της δικτυακής κίνησης προσφέρει μια πολλά υποσχόμενη προσέγγιση για την αντιμετώπιση των περιορισμών των παραδοσιακών πρωτοκόλλων δρομολόγησης. Αξιοποιώντας την προσαρμοστική μάθηση και την ανατροφοδότηση σε πραγματικό χρόνο, ο αλγόριθμος μπορεί να βελτιστοποιήσει δυναμικά τις αποφάσεις δρομολόγησης, βελτιώνοντας την απόδοση και την αξιοπιστία του δικτύου. Οι απαραίτητες τροποποιήσεις, συμπεριλαμβανομένων προσαρμοσμένων συναρτήσεων απωλειών, μηχανισμών ανατροφοδότησης σε πραγματικό χρόνο, ιεραρχικής κλιμάκωσης και ισχυρού χειρισμού δυναμικών συνθηκών δικτύου, διασφαλίζουν ότι ο αλγόριθμος είναι κατάλληλος για τις πολυπλοκότητες των σύγχρονων δικτυακών περιβαλλόντων. Μέσω αυτών των βελτιώσεων, ο αλγόριθμος Hedge μπορεί να αποτελέσει ένα ισχυρό εργαλείο για την έξυπνη και προσαρμοστική διαχείριση της δικτυακής κίνησης, ανοίγοντας το δρόμο για πιο αποδοτικές και ανθεκτικές υποδομές δικτύου (Qi et al., 2018).

3.3. Λεπτομέρειες εφαρμογής

Η εφαρμογή του αλγορίθμου Hedge σε δρομολογητές δικτύου απαιτεί προσεκτική εξέταση διαφόρων πρακτικών πτυχών για να εξασφαλιστεί η απρόσκοπτη ενσωμάτωση και η βέλτιστη απόδοση. Η διαδικασία αυτή περιλαμβάνει την αντιμετώπιση των ειδικών αναγκών της διαχείρισης της κυκλοφορίας σε πραγματικό χρόνο και την προσαρμογή του αλγορίθμου ώστε να λειτουργεί αποτελεσματικά εντός των περιορισμών της υπάρχουσας υποδομής δικτύου. Σε αυτή την ενότητα, θα διερευνήσουμε αυτές τις πρακτικές εκτιμήσεις, θα συζητήσουμε την ενσωμάτωση με τα τρέχοντα συστήματα δικτύου και θα εμβαθύνουμε στις απαραίτητες αλγοριθμικές προσαρμογές για την αποτελεσματική διαχείριση της κυκλοφορίας σε πραγματικό χρόνο (Guo & Mu, 2023a).

3.3.1. Πρακτικές εκτιμήσεις για την υλοποίηση Hedge σε δρομολογητές

Η επιτυχής εφαρμογή του αλγορίθμου Hedge σε δρομολογητές δικτύου εξαρτάται από την ενδελεχή κατανόηση του περιβάλλοντος του δικτύου και των λειτουργικών απαιτήσεων του αλγορίθμου. Μια από τις πρωταρχικές εκτιμήσεις είναι η υπολογιστική επιβάρυνση που συνδέεται με την εκτέλεση του αλγορίθμου. Ο αλγόριθμος Hedge περιλαμβάνει επαναληπτικές ενημερώσεις και υπολογισμούς πιθανοτήτων, οι οποίοι πρέπει να εκτελούνται γρήγορα για να ανταποκρίνονται στις συνθήκες του δικτύου σε πραγματικό χρόνο. Επομένως, η επεξεργαστική ισχύς των δρομολογητών πρέπει να είναι επαρκής για να χειριστεί αυτούς τους υπολογισμούς χωρίς να προκαλεί καθυστερήσεις στην προώθηση πακέτων (Zhang et al., 2023).

Ένας άλλος κρίσιμος παράγοντας είναι η επιλογή των κατάλληλων παραμέτρων για τον αλγόριθμο, ιδίως του ρυθμού μάθησης (ϵ). Όπως συζητήθηκε προηγουμένως, ο ρυθμός μάθησης επηρεάζει την ισορροπία μεταξύ εξερεύνησης και εκμετάλλευσης, καθώς και την ταχύτητα σύγκλισης. Σε ένα πρακτικό περιβάλλον, η επιλογή ενός βέλτιστου ϵ είναι απαραίτητη για να διασφαλιστεί ότι ο αλγόριθμος προσαρμόζεται γρήγορα στις αλλαγές στην κυκλοφορία του δικτύου χωρίς υπερβολικές ταλαντώσεις. Αυτό απαιτεί εμπειρικό συντονισμό και μπορεί να περιλαμβάνει αρχικές δοκιμαστικές εκτελέσεις για τον προσδιορισμό των πιο αποτελεσματικών ρυθμίσεων παραμέτρων για ένα δεδομένο δίκτυο (Mourtada & Gaïffas, 2019).

Επιπλέον, η εφαρμογή πρέπει να λαμβάνει υπόψη τη δυναμική φύση της δικτυακής κίνησης. Τα μοτίβα κίνησης μπορεί να μεταβάλλονται σημαντικά με την πάροδο του χρόνου, γεγονός που καθιστά αναγκαίο έναν προσαρμοστικό αλγόριθμο που μπορεί να μαθαίνει και να προσαρμόζεται σε πραγματικό χρόνο. Η ικανότητα του αλγορίθμου Hedge να ενημερώνει τα βάρη με βάση τις παρατηρούμενες απώλειες τον καθιστά κατάλληλο για αυτό το έργο. Ωστόσο, αυτό σημαίνει επίσης ότι ο αλγόριθμος πρέπει να έχει πρόσβαση σε ακριβή και έγκαιρη ανατροφοδότηση σχετικά με μετρικές επιδόσεων του δικτύου, όπως η καθυστέρηση, η απόδοση και οι απώλειες πακέτων. Η ενσωμάτωση μηχανισμών παρακολούθησης σε πραγματικό χρόνο και συλλογής ανατροφοδότησης στους δρομολογητές είναι επομένως απαραίτητη για την αποτελεσματική λειτουργία του αλγορίθμου (Qi et al., 2018).

Επιπλέον, ο αλγόριθμος πρέπει να είναι ανθεκτικός ώστε να μπορεί να χειρίζεται διάφορες συνθήκες δικτύου, συμπεριλαμβανομένων των υψηλών όγκων κυκλοφορίας και των πιθανών αποτυχιών συνδέσεων. Η εφαρμογή μηχανισμών αποτυχίας και εφεδρείας είναι ζωτικής σημασίας για τη διατήρηση της σταθερότητας και της απόδοσης του δικτύου. Για παράδειγμα, εάν ο αλγόριθμος Hedge ανιχνεύσει ότι ένα συγκεκριμένο μονοπάτι δρομολόγησης γίνεται συμφορημένο, θα πρέπει να έχει τη δυνατότητα να μεταβαίνει γρήγορα σε εναλλακτικά μονοπάτια χωρίς να προκαλεί σημαντική διακοπή (Anagnostou & Lambrou, 2014).

3.3.2. Ενσωμάτωση με την υπάρχουσα δικτυακή υποδομή

Η ενσωμάτωση του αλγορίθμου Hedge στην υπάρχουσα δικτυακή υποδομή περιλαμβάνει τόσο ζητήματα λογισμικού όσο και υλικού. Από την πλευρά του λογισμικού, ο αλγόριθμος πρέπει να ενσωματωθεί στο λειτουργικό σύστημα του δρομολογητή, διασφαλίζοντας ότι μπορεί να αλληλεπιδράσει απρόσκοπτα με άλλα πρωτόκολλα δρομολόγησης και συστήματα διαχείρισης. Αυτό μπορεί να περιλαμβάνει την ανάπτυξη προσαρμοσμένου υλικολογισμικού ή πρόσθετων προγραμμάτων που ενσωματώνουν τον αλγόριθμο στο επίπεδο ελέγχου του δρομολογητή (Krichene et al., 2015).

Η συμβατότητα με τα υπάρχοντα πρωτόκολλα δρομολόγησης είναι μια άλλη σημαντική πτυχή. Τα παραδοσιακά πρωτόκολλα δρομολόγησης, όπως το OSPF και το BGP, διαθέτουν καθιερωμένους μηχανισμούς για την επιλογή διαδρομής και τη διαχείριση της κυκλοφορίας. Ο αλγόριθμος Hedge μπορεί να συμπληρώσει αυτά τα πρωτόκολλα παρέχοντας ένα πρόσθετο επίπεδο προσαρμοστικής λήψης αποφάσεων. Για παράδειγμα, ενώ το OSPF μπορεί να προσδιορίσει τη συντομότερη διαδρομή με βάση στατικές μετρικές, ο αλγόριθμος Hedge μπορεί να προσαρμόσει δυναμικά τις αποφάσεις δρομολόγησης με βάση την ανατροφοδότηση της απόδοσης σε πραγματικό χρόνο, βελτιστοποιώντας έτσι τις τρέχουσες συνθήκες του δικτύου (Bugday et al., 2019).

Από την άποψη του υλικού, η υλοποίηση θα πρέπει να αξιοποιεί τις δυνατότητες των σύγχρονων δρομολογητών, οι οποίοι συχνά περιλαμβάνουν προηγμένες μονάδες επεξεργασίας και μνήμη υψηλής ταχύτητας. Η αποτελεσματική αξιοποίηση αυτών των πόρων μπορεί να συμβάλει στον μετριασμό των υπολογιστικών απαιτήσεων του αλγορίθμου. Για παράδειγμα, δρομολογητές εξοπλισμένοι με επεξεργαστές πολλαπλών πυρήνων μπορούν να παραλληλίσουν τους υπολογισμούς που απαιτούνται για τον αλγόριθμο Hedge, εξασφαλίζοντας ταχύτερες ενημερώσεις και μειωμένη καθυστέρηση (Zhang et al., 2023).

Τα συστήματα διαχείρισης δικτύων διαδραματίζουν επίσης ζωτικό ρόλο στη διαδικασία ολοκλήρωσης. Τα συστήματα αυτά παρέχουν μια συγκεντρωτική εικόνα του δικτύου, διευκολύνοντας τη συλλογή και την ανάλυση μετρήσεων απόδοσης. Με την ενσωμάτωση του αλγορίθμου Hedge με πλατφόρμες διαχείρισης δικτύου, οι διαχειριστές μπορούν να αποκτήσουν πληροφορίες σχετικά με τη διαδικασία λήψης αποφάσεων του αλγορίθμου, να παρακολουθούν την απόδοσή του και να προβαίνουν σε ενημερωμένες προσαρμογές ανάλογα με τις ανάγκες. Η ενσωμάτωση αυτή μπορεί να επιτευχθεί μέσω API και διεπαφών διαχείρισης που επιτρέπουν στον αλγόριθμο να επικοινωνεί με εργαλεία παρακολούθησης και ελεγκτές δικτύου (Guo & Mu, 2023a).

3.3.3. Αλγοριθμικές προσαρμογές για τη διαχείριση της κυκλοφορίας σε πραγματικό χρόνο

Η προσαρμογή του αλγορίθμου Hedge για τη διαχείριση της κυκλοφορίας σε πραγματικό χρόνο απαιτεί αρκετές αλγοριθμικές προσαρμογές ώστε να διασφαλιστεί ότι ανταποκρίνεται στις απαιτήσεις των δυναμικών και ταχύτατων δικτύων. Μια βασική προσαρμογή αφορά τη συχνότητα των ενημερώσεων βάρους. Σε μια τυπική εφαρμογή, ο αλγόριθμος ενημερώνει τα βάρη σε τακτά χρονικά διαστήματα με βάση τις παρατηρούμενες απώλειες. Για τη διαχείριση της κυκλοφορίας σε πραγματικό χρόνο, τα διαστήματα αυτά πρέπει να είναι αρκετά μικρά ώστε να καταγράφουν τις ταχείες αλλαγές στα πρότυπα κυκλοφορίας, αλλά αρκετά μεγάλα ώστε να επιτρέπουν τη συλλογή δεδομένων με νόημα. Η επίτευξη της σωστής ισορροπίας είναι ζωτικής σημασίας για τη διατήρηση τόσο της απόκρισης όσο και της σταθερότητας (Qi et al., 2018).

Μια άλλη προσαρμογή αφορά τη συνάρτηση απώλειας που χρησιμοποιεί ο αλγόριθμος. Στο πλαίσιο της δρομολόγησης δικτύου, η συνάρτηση απώλειας θα πρέπει να αντικατοπτρίζει με ακρίβεια τις μετρήσεις απόδοσης που έχουν μεγαλύτερη σημασία, όπως η καθυστέρηση, το τρεμούλιασμα και η απώλεια πακέτων. Ο σχεδιασμός μιας συνάρτησης απωλειών που καταγράφει αυτές τις μετρήσεις και τις μεταφράζει σε ανατροφοδότηση με δυνατότητα δράσης είναι ουσιαστικός για την αποτελεσματικότητα του αλγορίθμου. Για παράδειγμα, υψηλότερες καθυστερήσεις και απώλειες πακέτων θα πρέπει να οδηγούν σε υψηλότερες απώλειες, προτρέποντας τον αλγόριθμο να προσαρμόσει ανάλογα τις πιθανότητες δρομολόγησης (Anagnostou & Lambrou, 2014).

Η επεκτασιμότητα είναι επίσης ένα κρίσιμο ζήτημα. Καθώς το μέγεθος και η πολυπλοκότητα του δικτύου αυξάνονται, ο αλγόριθμος Hedge πρέπει να κλιμακώνεται αποτελεσματικά για να διαχειρίζεται τον αυξημένο αριθμό αποφάσεων δρομολόγησης. Αυτό μπορεί να επιτευχθεί μέσω ιεραρχικών προσεγγίσεων, όπου ο αλγόριθμος λειτουργεί σε πολλαπλά επίπεδα της ιεραρχίας του δικτύου, από μεμονωμένους δρομολογητές έως συστάδες δρομολογητών. Κάθε επίπεδο μπορεί να λάβει τοπικές αποφάσεις, οι οποίες στη συνέχεια συγκεντρώνονται για να ενημερώσουν ευρύτερες στρατηγικές σε όλο το δίκτυο. Αυτή η ιεραρχική προσέγγιση βοηθά στη διαχείριση του υπολογιστικού φόρτου και εξασφαλίζει ότι ο αλγόριθμος παραμένει αποτελεσματικός ακόμη και σε δίκτυα μεγάλης κλίμακας (Guo & Mu, 2023a).

Επιπλέον, ο αλγόριθμος πρέπει να είναι ανθεκτικός σε ανωμαλίες και παροδικές αλλαγές στις συνθήκες του δικτύου. Στα δίκτυα του πραγματικού κόσμου μπορεί να εμφανιστούν ξαφνικές αιχμές στην κυκλοφορία ή προσωρινές αποτυχίες συνδέσεων. Ο αλγόριθμος Hedge θα πρέπει να ενσωματώνει μηχανισμούς για την ανίχνευση και τον χειρισμό τέτοιων ανωμαλιών, διασφαλίζοντας ότι τα παροδικά προβλήματα δεν οδηγούν σε μακροπρόθεσμες μη βέλτιστες αποφάσεις. Αυτό θα μπορούσε να περιλαμβάνει την ενσωμάτωση τεχνικών ανίχνευσης ανωμαλιών που επισημαίνουν ασυνήθιστα μοτίβα και προκαλούν συγκεκριμένες αντιδράσεις, όπως η προσωρινή προσαρμογή του ρυθμού μάθησης ή η επιθετικότερη εξερεύνηση εναλλακτικών διαδρομών (Mourtada & Gaïffas, 2019).

Τέλος, η εφαρμογή ενός βρόχου ανατροφοδότησης που αξιολογεί συνεχώς την απόδοση του αλγορίθμου και πραγματοποιεί προσαρμογές σε πραγματικό χρόνο είναι ζωτικής σημασίας. Αυτός ο βρόχος ανατροφοδότησης μπορεί να αξιοποιήσει τεχνικές μηχανικής μάθησης για την πρόβλεψη μελλοντικών προτύπων κυκλοφορίας και την προληπτική προσαρμογή των αποφάσεων δρομολόγησης. Με την ενσωμάτωση της προγνωστικής ανάλυσης, ο αλγόριθμος Hedge μπορεί να γίνει πιο προληπτικός, προβλέποντας τις αλλαγές στις συνθήκες του δικτύου και προσαρμοζόμενος αποτελεσματικότερα (Zhang et al., 2023).

Η πρακτική εφαρμογή του αλγορίθμου Hedge σε δρομολογητές δικτύου περιλαμβάνει την αντιμετώπιση της υπολογιστικής επιβάρυνσης, την επιλογή παραμέτρων, την παρακολούθηση σε πραγματικό χρόνο και την ενσωμάτωση με την υπάρχουσα υποδομή. Οι αλγοριθμικές προσαρμογές είναι απαραίτητες για να διασφαλιστεί η επεκτασιμότητα, η ευρωστία και η αποτελεσματική διαχείριση της κυκλοφορίας σε πραγματικό χρόνο. Με την προσεκτική εξέταση αυτών των παραγόντων, ο αλγόριθμος Hedge μπορεί να ενσωματωθεί με επιτυχία σε συστήματα δικτύου, παρέχοντας προσαρμοστικές και αποτελεσματικές λύσεις δρομολόγησης που βελτιώνουν τη συνολική απόδοση του δικτύου (Bugday et al., 2019).

3.4. Αξιολόγηση επιδόσεων

Η αξιολόγηση της απόδοσης του αλγορίθμου Hedge στο πλαίσιο της δρομολόγησης της δικτυακής κίνησης απαιτεί μια ολοκληρωμένη προσέγγιση που λαμβάνει υπόψη πολλαπλές μετρικές και ποικίλα σενάρια. Στόχος είναι να αξιολογηθεί η αποδοτικότητα και η αποτελεσματικότητα του αλγορίθμου στη βελτιστοποίηση της απόδοσης του δικτύου, ιδίως όσον αφορά την καθυστέρηση, την απόδοση και την απώλεια πακέτων. Στην παρούσα ενότητα περιγράφονται οι βασικές μετρικές για την αξιολόγηση των επιδόσεων, περιγράφονται οι ρυθμίσεις προσομοίωσης και τα σενάρια που χρησιμοποιήθηκαν για τη δοκιμή του αλγορίθμου και παρέχεται συγκριτική ανάλυση με άλλους προσαρμοστικούς αλγορίθμους δρομολόγησης (Qi et al., 2018).

3.4.1. Μετρικές για την αξιολόγηση της αποδοτικότητας δρομολόγησης

Για να αξιολογηθεί η απόδοση του αλγορίθμου Hedge στη δρομολόγηση της δικτυακής κίνησης, είναι απαραίτητο να εξεταστεί μια σειρά από μετρικές που αποτυπώνουν διαφορετικές πτυχές της απόδοσης του δικτύου. Οι κύριες μετρικές περιλαμβάνουν την καθυστέρηση, την απόδοση και την απώλεια πακέτων, καθεμία από τις οποίες παρέχει πολύτιμες πληροφορίες για την αποτελεσματικότητα του αλγορίθμου (Krichene et al., 2015).

Η καθυστέρηση, ή λανθάνουσα κατάσταση, μετράει το χρόνο που απαιτείται για να ταξιδέψουν τα πακέτα δεδομένων από την πηγή στον προορισμό. Στο πλαίσιο της δρομολόγησης δικτύου, η ελαχιστοποίηση της καθυστέρησης είναι ζωτικής σημασίας για τη διασφάλιση της έγκαιρης παράδοσης δεδομένων, ιδίως για εφαρμογές πραγματικού χρόνου, όπως οι τηλεδιασκέψεις και τα διαδικτυακά παιχνίδια. Ο αλγόριθμος Hedge στοχεύει στη βελτιστοποίηση των αποφάσεων δρομολόγησης για τη μείωση της συνολικής καθυστέρησης του δικτύου, βελτιώνοντας έτσι την ποιότητα των υπηρεσιών για τους τελικούς χρήστες (Anagnostou & Lambrou, 2014).

Η απόδοση αναφέρεται στην ποσότητα των δεδομένων που μεταδίδονται επιτυχώς μέσω του δικτύου σε μια δεδομένη περίοδο. Ο υψηλός ρυθμός μετάδοσης υποδηλώνει την αποτελεσματική χρήση των πόρων του δικτύου και την ικανότητα διαχείρισης μεγάλου όγκου κίνησης. Προσαρμόζοντας δυναμικά τα μονοπάτια δρομολόγησης με βάση την ανατροφοδότηση σε πραγματικό χρόνο, ο αλγόριθμος Hedge επιδιώκει τη μεγιστοποίηση της απόδοσης, διασφαλίζοντας ότι το δίκτυο μπορεί να υποστηρίξει υψηλούς ρυθμούς δεδομένων χωρίς συμφόρηση (Guo & Mu, 2023a).

Η απώλεια πακέτων μετρά το ποσοστό των πακέτων δεδομένων που χάνονται κατά τη μετάδοση. Η υψηλή απώλεια πακέτων μπορεί να υποβαθμίσει την απόδοση του δικτύου και να οδηγήσει σε επαναμεταδόσεις, αυξάνοντας περαιτέρω την καθυστέρηση και μειώνοντας την απόδοση. Η προσαρμοστική προσέγγιση μάθησης του αλγορίθμου Hedge βοηθά στον εντοπισμό και την αποφυγή συμφορημένων ή αναξιόπιστων διαδρομών, ελαχιστοποιώντας έτσι την απώλεια πακέτων και ενισχύοντας τη συνολική αξιοπιστία του δικτύου (Mourtada & Gaïffas, 2019).

Εκτός από αυτές τις πρωταρχικές μετρικές, μπορούν επίσης να εξεταστούν και άλλοι παράγοντες, όπως το jitter (η μεταβλητότητα της καθυστέρησης των πακέτων) και η δικαιοσύνη (η δίκαιη κατανομή των πόρων του δικτύου μεταξύ των χρηστών), ώστε να παρέχεται μια πιο ολοκληρωμένη αξιολόγηση της απόδοσης του αλγορίθμου (Bugday et al., 2019).

3.4.2. Σενάρια και ρυθμίσεις προσομοίωσης

Για την αυστηρή αξιολόγηση του αλγορίθμου Hedge, είναι απαραίτητο να διεξαχθούν προσομοιώσεις που αναπαράγουν τις πραγματικές συνθήκες του δικτύου. Αυτές οι προσομοιώσεις θα πρέπει να περιλαμβάνουν μια ποικιλία τοπολογιών δικτύου, μοτίβων κυκλοφορίας και λειτουργικών σεναρίων, ώστε να διασφαλίζονται αξιόπιστα και γενικεύσιμα αποτελέσματα. (Krichene et al., 2015)

Η εγκατάσταση προσομοίωσης περιλαμβάνει τη δημιουργία ενός μοντέλου δικτύου με κόμβους που αντιπροσωπεύουν δρομολογητές και συνδέσμους που αντιπροσωπεύουν διαδρομές επικοινωνίας μεταξύ τους. Το μοντέλο δικτύου μπορεί να διαμορφωθεί ώστε να αντικατοπτρίζει διαφορετικές τοπολογίες, όπως πλέγμα, αστέρι και ιεραρχικές δομές, καθεμία με διαφορετικό βαθμό συνδεσιμότητας και πολυπλοκότητας. Οι γεννήτριες κυκλοφορίας χρησιμοποιούνται για την προσομοίωση των ροών δεδομένων στο δίκτυο, με παραμέτρους όπως το μέγεθος των πακέτων, ο ρυθμός μετάδοσης και τα πρότυπα κυκλοφορίας (π.χ. σταθερός ρυθμός μετάδοσης, εκρηκτική κυκλοφορία) που προσαρμόζονται ώστε να μιμούνται τις πραγματικές συνθήκες (Zhang et al., 2023).

Ένα βασικό σενάριο περιλαμβάνει τη δοκιμή του αλγορίθμου υπό διαφορετικά φορτία κυκλοφορίας. Αυξάνοντας σταδιακά τον όγκο της κίνησης, η προσομοίωση μπορεί να αξιολογήσει πόσο καλά ο αλγόριθμος Hedge προσαρμόζεται στις μεταβαλλόμενες συνθήκες του δικτύου και διατηρεί τις μετρήσεις απόδοσης. Ένα άλλο σενάριο περιλαμβάνει την εισαγωγή διαταραχών του δικτύου, όπως αποτυχίες συνδέσεων ή ξαφνικές αιχμές κυκλοφορίας, για να αξιολογηθεί η ανθεκτικότητα του αλγορίθμου και η ικανότητά του να ανακάμπτει από δυσμενή γεγονότα (Qi et al., 2018).

Το περιβάλλον προσομοίωσης θα πρέπει επίσης να υποστηρίζει την παρακολούθηση σε πραγματικό χρόνο και την καταγραφή των μετρικών επιδόσεων. Αυτό επιτρέπει τη συνεχή ανατροφοδότηση του αλγορίθμου Hedge, επιτρέποντάς του να ενημερώνει δυναμικά τις αποφάσεις δρομολόγησης με βάση τις παρατηρούμενες συνθήκες του δικτύου. Με την ανάλυση των δεδομένων που συλλέγονται, είναι δυνατή η αξιολόγηση της αποτελεσματικότητας του αλγορίθμου όσον αφορά τη βελτιστοποίηση των αποφάσεων δρομολόγησης και τη βελτίωση της συνολικής απόδοσης του δικτύου (Bugday et al., 2019).

3.4.3. Συγκριτική ανάλυση με άλλους αλγόριθμους προσαρμοστικής δρομολόγησης

Για να καταδειχθούν τα πλεονεκτήματα του αλγορίθμου Hedge και να εντοπιστούν οι τομείς που χρήζουν βελτίωσης, είναι απαραίτητο να συγκριθεί η απόδοσή του με άλλους αλγορίθμους προσαρμοστικής δρομολόγησης. Αυτή η συγκριτική ανάλυση παρέχει ένα σημείο αναφοράς για την αξιολόγηση των σχετικών δυνατών και αδύνατων σημείων του αλγορίθμου (Qi et al., 2018).

Ένας ευρέως χρησιμοποιούμενος προσαρμοστικός αλγόριθμος δρομολόγησης για σύγκριση είναι το πρωτόκολλο Δυναμικής Δρομολόγησης Πηγής (DSR). Το DSR έχει σχεδιαστεί για ασύρματα δίκτυα πολλαπλών βημάτων και προσαρμόζει τα μονοπάτια δρομολόγησης με βάση τις συνθήκες του δικτύου σε πραγματικό χρόνο. Συγκρίνοντας την απόδοση του αλγορίθμου Hedge με το DSR, είναι δυνατόν να αξιολογηθεί η ικανότητά του να χειρίζεται δυναμικά και ετερογενή περιβάλλοντα δικτύου (Anagnostou & Lambrou, 2014).

Ένας άλλος αλγόριθμος για σύγκριση είναι το πρωτόκολλο OLSR (Optimized Link State Routing), το οποίο χρησιμοποιεί μια προληπτική προσέγγιση για τη διατήρηση ενημερωμένων πληροφοριών δρομολόγησης. Το OLSR ανταλλάσσει περιοδικά πληροφορίες τοπολογίας μεταξύ των κόμβων, επιτρέποντάς του να προσαρμόζεται γρήγορα στις αλλαγές του δικτύου. Η σύγκριση του αλγορίθμου Hedge με το OLSR μπορεί να αναδείξει τις διαφορές στην προσαρμοστικότητα και την ανταπόκριση στις μεταβαλλόμενες συνθήκες του δικτύου (Mourtada & Gaïffas, 2019).

Η σύγκριση των επιδόσεων θα πρέπει να λαμβάνει υπόψη βασικές μετρήσεις, όπως η καθυστέρηση, η απόδοση, η απώλεια πακέτων και το jitter. Για παράδειγμα, οι προσομοιώσεις μπορεί να δείξουν ότι ο αλγόριθμος Hedge επιτυγχάνει χαμηλότερη καθυστέρηση και υψηλότερη απόδοση σε σύγκριση με τους DSR και OLSR υπό ορισμένες συνθήκες κυκλοφορίας. Αντίθετα, ο OLSR μπορεί να παρουσιάζει ταχύτερους χρόνους σύγκλισης σε σταθερά περιβάλλοντα δικτύου λόγω της προληπτικής του φύσης. Αυτές οι γνώσεις βοηθούν στον εντοπισμό των συγκεκριμένων σεναρίων στα οποία ο αλγόριθμος Hedge υπερέχει και στα οποία ενδέχεται να απαιτούνται περαιτέρω βελτιώσεις (Krichene et al., 2015).

Εκτός από τις μετρήσεις απόδοσης, η σύγκριση θα πρέπει επίσης να αξιολογεί την υπολογιστική πολυπλοκότητα και την επεκτασιμότητα των αλγορίθμων. Η πιθανολογική προσέγγιση του αλγορίθμου Hedge και η συνεχής διαδικασία μάθησης μπορεί να προσφέρουν πλεονεκτήματα όσον αφορά την επεκτασιμότητα και την προσαρμοστικότητα. Ωστόσο, είναι σημαντικό να αξιολογηθεί η υπολογιστική επιβάρυνση και οι απαιτήσεις σε πόρους, ιδίως σε δίκτυα μεγάλης κλίμακας με πολυάριθμους κόμβους και υψηλό όγκο κίνησης (Guo & Mu, 2023a).

Η πρακτική εφαρμογή που περιγράφηκε προηγουμένως παρέχει τα θεμέλια για αυτές τις συγκριτικές αναλύσεις. Επαναλαμβάνοντας τις ίδιες συνθήκες δικτύου και τις ίδιες μετρήσεις απόδοσης, είναι δυνατή η δημιουργία ουσιαστικών συγκρίσεων που αναδεικνύουν τα σχετικά πλεονεκτήματα και τα πιθανά μειονεκτήματα του αλγορίθμου Hedge. Για παράδειγμα, σε σενάρια που περιλαμβάνουν ξαφνικές αιχμές κυκλοφορίας ή αποτυχίες συνδέσεων, η δυναμική προσαρμοστικότητα του αλγορίθμου Hedge μπορεί να οδηγήσει σε ταχύτερη ανάκαμψη και καλύτερες επιδόσεις σε σύγκριση με πιο στατικούς αλγορίθμους όπως ο OLSR (Zhang et al., 2023).

Η αξιολόγηση της απόδοσης του αλγορίθμου Hedge στη δρομολόγηση της δικτυακής κίνησης περιλαμβάνει μια ολοκληρωμένη προσέγγιση που λαμβάνει υπόψη πολλαπλές μετρικές, ποικίλα σενάρια προσομοίωσης και συγκριτική ανάλυση με άλλους προσαρμοστικούς αλγορίθμους δρομολόγησης. Εστιάζοντας σε βασικές μετρικές, όπως η καθυστέρηση, η απόδοση και η απώλεια πακέτων, και πραγματοποιώντας αυστηρές προσομοιώσεις, είναι δυνατόν να καταδειχθεί η αποτελεσματικότητα του αλγορίθμου και να εντοπιστούν τομείς για περαιτέρω βελτίωση. Η συγκριτική ανάλυση παρέχει πολύτιμα σημεία αναφοράς, αναδεικνύοντας τα δυνατά σημεία του αλγορίθμου Hedge στη δυναμική προσαρμοστικότητα και τη βελτιστοποίηση σε πραγματικό χρόνο, ενώ παράλληλα υποδεικνύει πιθανές βελτιώσεις στην υπολογιστική αποδοτικότητα και την επεκτασιμότητα. Μέσω αυτών των αξιολογήσεων, ο αλγόριθμος Hedge μπορεί να τοποθετηθεί ως ένα ισχυρό εργαλείο για τη σύγχρονη διαχείριση της δικτυακής κίνησης, ικανό να ανταποκριθεί στις απαιτήσεις όλο και πιο σύνθετων και δυναμικών δικτυακών περιβαλλόντων (Mourtada & Gaïffas, 2019).

4. Πρακτικό Μέρος

4.1. Ο αλγορίθμος hedge

Ο αλγόριθμος Hedge, γνωστός και ως αλγόριθμος Multiplicative Weights Update (MWU), είναι ένας αλγόριθμος μάθησης που χρησιμοποιείται στον τομέα της online μάθησης και της θεωρίας παιγνίων και της βελτιστοποίησης. Έχει σχεδιαστεί για να λαμβάνει διαδοχικές αποφάσεις σε ένα ανταγωνιστικό περιβάλλον με άγνωστες ή μεταβαλλόμενες πιθανότητες.

Ο αλγόριθμος Hedge διατηρεί μια κατανομή σε ένα σύνολο ενεργειών και ενημερώνει την κατανομή με βάση την παρατηρούμενη ανατροφοδότηση (feedback) ή τα αποτελέσματα. Η ενημέρωση εκτελείται με την ανάθεση βαρών σε κάθε ενέργεια και την προσαρμογή των βαρών με βάση την παρατηρούμενη ανατροφοδότηση. Οι ενέργειες που αποδίδουν θετικά αποτελέσματα λαμβάνουν υψηλότερα βάρη, ενώ οι ενέργειες με αρνητικά αποτελέσματα λαμβάνουν χαμηλότερα βάρη. Ο αλγόριθμος εξισορροπεί την εξερεύνηση και την εκμετάλλευση αποδίδοντας υψηλότερα βάρη σε ενέργειες που έχουν διερευνηθεί λιγότερο. Ο αλγόριθμος Hedge επιτυγχάνει μια μορφή ελαχιστοποίησης, με στόχο να ελαχιστοποιήσει τη διαφορά μεταξύ της σωρευτικής ανταμοιβής που λαμβάνεται από τον αλγόριθμο και της ανταμοιβής που θα είχε επιτευχθεί με την καλύτερη σταθερή ενέργεια εκ των υστέρων.

Έστω ότι ο αντίπαλος επιλέγει σε κάθε βήμα t την ενέργεια g_t (επιφέρει απώλεια g_t την οποία θέλουμε να ελαχιστοποιήσουμε). Εμείς πρέπει να διαλέξουμε την ενέργεια i με βάση την πιθανότητα x(i) που υπολογίζεται από τα βάρη W(i). Σε κάθε επανάληψη ανανεώνονται τα βάρη με εκθετικό τρόπο ως προς τα g_t. Παρακάτω στην εικόνα 1 φαίνεται ο ψευδοκώδικας του αλγορίθμου.

Algorithm 2 Hedge

Inititalization $W_1 = \mathbf{1} \in \mathbf{R}^n$ % $W_1 = (1, 1, \ldots, 1)$.
SET $\mathbf{x}_1 = \frac{1}{n} W_1$.
for $t = 1, 2 \ldots T$ **do**
  Pick $i_t \propto \mathbf{x}_t$. i.e. $p(i_t = i) = \mathbf{x}_t(i)$.
  incure loss $\mathbf{g}_t(i_t)$.
  Update Weights $W_{t+1}(i) = W_t(i) \cdot e^{-\epsilon \mathbf{g}_t(i)}$
  Set $\mathbf{x}_{t+1}(i) = \frac{W_{t+1}}{\sum_j W_t(j)}$.
end for
return

Εικόνα 1. Ψευδοκώδικας αλγορίθμου Hedge (πηγή https://www.cs.princeton.edu/~rlivni/cos511/lectures/lect18.pdf).

Δηλαδή, κατά τη διάρκεια των γύρων, ο αλγόριθμος μαθαίνει ποιες ενέργειες δίνουν μικρές απώλειες και δίνει σε αυτές μεγαλύτερες πιθανότητες. Οπότε, σε κάθε γύρο, ο παίκτης στοιχηματίζει ένα χρηματικό ποσό ανάλογα με τις πιθανότητες και λαμβάνει την αντίστοιχη ανταμοιβή. Πιο αναλυτικά, για ένα αρχικό ποσό M, ο παίκτης εκχωρεί ένα κλάσμα p_i (με $\Sigma\ p_i = 1$) του M στην επιλογή i. Η ανταμοιβή είναι:

$$R = \sum_{i=1}^{N} p_i M \times \ell_i = M \sum_{i=1}^{N} p_i \times \ell_i$$

4.2. Υλοποίηση αλγορίθμου hedge

Η υλοποίηση έγινε σε γλώσσα προγραμματισμού Python που είναι κατάλληλη για μαθηματικές πράξεις με διανύσματα. Εγκαταστήσαμε το πακέτο Anaconda Python το οποίο έρχεται με προεγκατηστημένες πολλές βιβλιοθήκες για μαθηματικά, στατιστική και μηχανική μάθηση. Παρακάτω δίνεται ο κώδικας σε Python με σχόλια που υλοποιεί τον ψευδοκώδικα της εικόνας 2. Πιο αναλυτικά, αρχικά δημιουργούμε μια συνάρτηση loss(i, l), η οποία επιστρέφει την τιμή για την ενέργεια i με βάση το προκαθορισμένο διάνυσμα l. Εδώ το διάνυσμα l αρχικοποιείται τυχαία για επίδειξη της ορθής λειτουργίας του αλγορίθμου. Η συνάρτηση hedge αρχικοποιεί τα διανύσματα W και x με τιμές 1/N, όπου N οι διαστάσεις του προβλήματος. Οι ενέργειες επίσης αρχικοποιούνται με τις τιμές 0 ως N-1. Στη συνέχεια γίνονται T επαναλήψεις του κυρίως κώδικα αλγορίθμου. Σε κάθε επανάληψη, διαλέγουμε τυχαία μια ενέργεια σύμφωνα με τις πιθανότητες x. Στη συνέχεια καλούμε τη συνάρτηση loss για να λάβουμε τις απώλειες για αυτή την ενέργεια. Μετά, ανανεώνουμε την αντίστοιχη τιμή του W για αυτή την ενέργεια με εκθετικό τρόπο. Στη συνέχεια, ανανεώνουμε τις πιθανότητες x (ουσιαστικά κανονικοποιώντας το W ώστε να έχει άθροισμα 1). Ο αλγόριθμος επιστρέφει τις τιμές του x και του W.

```
import numpy as np
import random

#MULTI_ARM BANDIT
def loss(i, l):
    #it depends on the problem, here return values from a
predefined vector
    return l[i]

def hedge(N, T, e, l):
    #initialize variables (vectors)
    W = np.ones(N)/N #normalized, to sum 1
    x = np.ones(N)/N #normalized, to sum 1

    #set actions (numbers 0,1,2...n-1)
    actions = np.arange(N)

    #historical values of x (useful for plots)
    x_hist = np.zeros((T,N))

    #run T times
    for t in range(T):
        #get action i [0 to n-1], according to
probability of x
        i = random.choices(actions, x/sum(x))[0]

        #get loss of action i
        g = loss(i, l)

        #update W
        W[i] = W[i] * np.exp(-e*g)

        #update x
        x[i] = W[i] / np.sum(W)

        #keep value of x
        x_hist[t,:] = x


    #return final values after T iterations
    return (x,W)

#simple example for N = 5 dimensions, T = 100 rounds, e =
0.1
```

```
l = np.random.uniform(size=(5))
(x,W) = hedge(5,100,0.1, l)
```

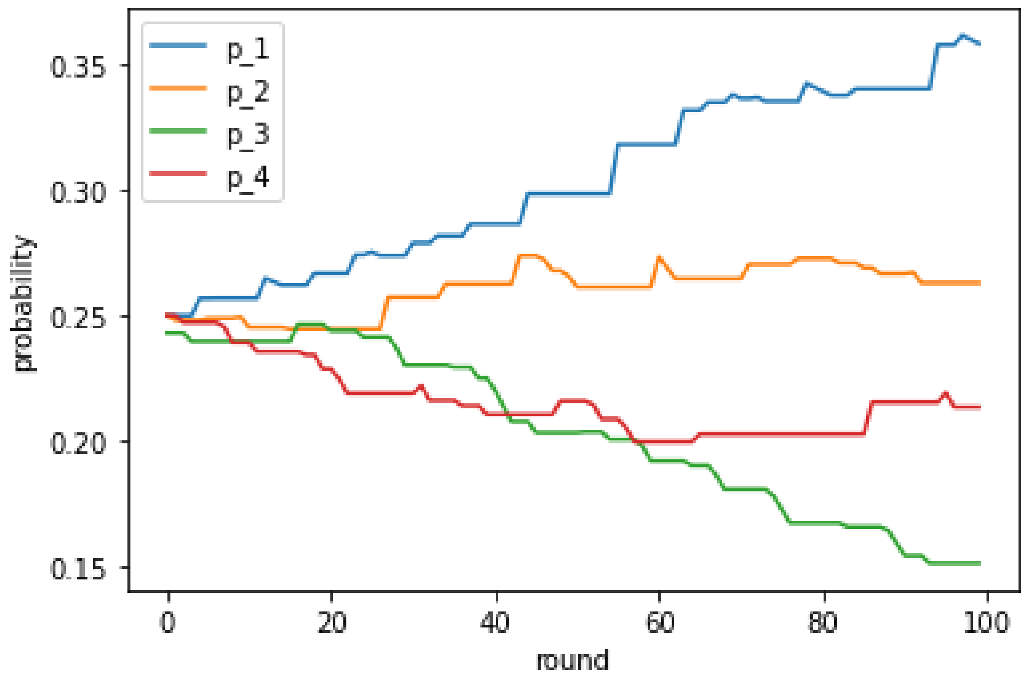
4.3. Πείραμα 1

Αρχικά, δοκιμάζουμε τον αλγόριθμο έχοντας έναν παίχτη με ένα σταθερό διάνυσμα απωλειών. Ο παίκτης έχει N = 4 επιλογές με απώλειες που αρχικοποιούνται τυχαία. Εκτελούμε τον αλγόριθμο hedge για T = 100 επαναλήψεις και παράμετρο e = 0.1. Ο κώδικας δίνεται στο παράρτημα.

Στην εικόνα 2 φαίνεται η έξοδος του προγράμματος, όπου φαίνονται οι απώλειες και οι πιθανότητες που μαθαίνει ο αλγόριθμος hedge. Στην εικόνα 3 φαίνεται η πορεία των πιθανοτήτων, όπου, όπως αναμένεται, οι τιμές που αντιστοιχούν στις μεγάλες απώλειες μειώνονται και αντίστροφα, οι τιμές για μεγάλες απώλειες αυξάνονται.

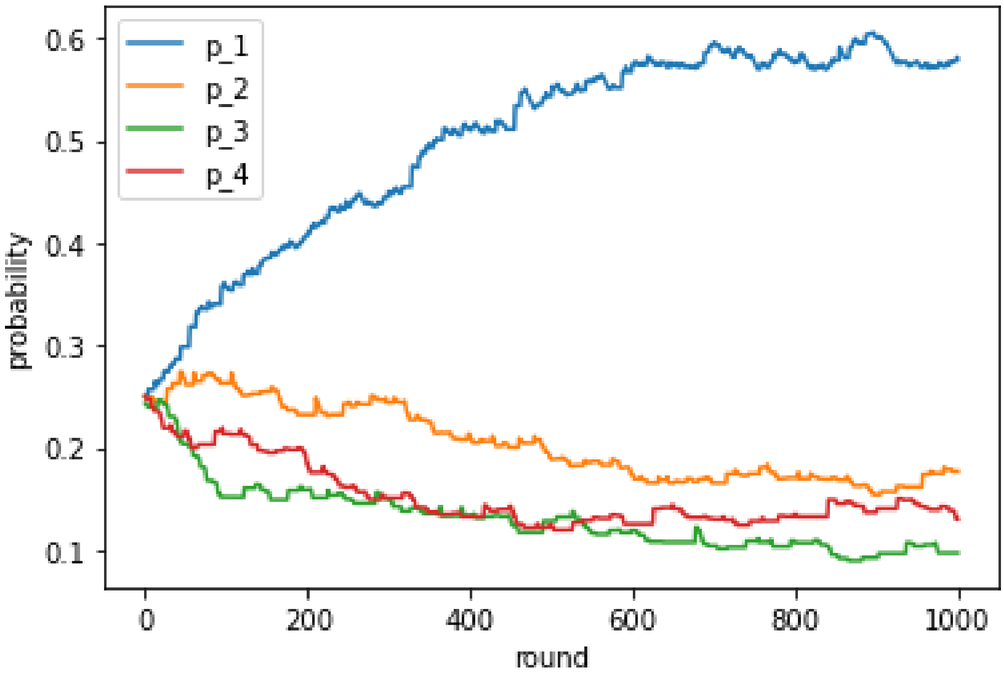
```
******************************
initial losses: [0.07013805458875595, 0.21904206933504763, 0.37112345085307363, 0.3396964252231229]
sum: 1.0000000000000002
x: [0.35829532 0.26299849 0.15150325 0.21353599] ---sum x: 0.9863330582041444
W: [0.20398788 0.15440335 0.0884392  0.12249846] ---sum W: 0.5693288816040539
Reward 0.2115016938424937
```

Εικόνα 2. Έξοδος του προγράμματος. Η μεταβλητή x είναι οι πιθανότητες που μαθαίνει ο αλγόριθμος hedge.

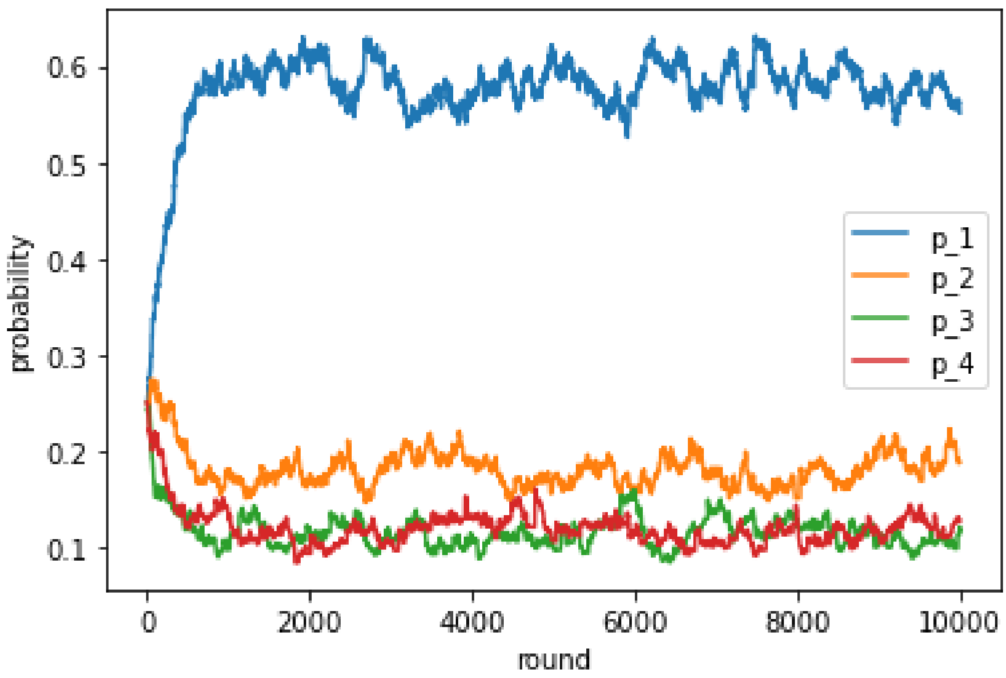


Εικόνα 3. Το διάνυσμα απωλειών είναι [0.07, 0.22, 0.37, 0.34]. Οι αντίστοιχες πιθανότητες που μαθαίνει ο αλγόριθμος hedge φαίνονται με μπλε, πορτοκαλί, πράσινο και κόκκινο χρώμα αντίστοιχα.

Για να εξετάσουμε τη συμπεριφορά του αλγορίθμου, κάνουμε διάφορες δοκιμές στις τιμές των παραμέτρων του. Αρχικά αυξάνουμε τον αριθμό γύρων από T=100 σε T=1000 (εικόνα 4) και T=10.000 (εικόνα 5). Στους 1000 γύρους, παρατηρούμε ότι οι πιθανότητες συνεχίζουν να αλλάζουν (π.χ. η μπλε γραμμή να αυξάνεται και οι άλλες να μειώνονται). Στους 10.000 γύρους φαίνεται ότι οι πιθανότητες έχουν σταθεροποιηθεί και κυμαίνονται τυχαία γύρω από κάποια τιμή. Παρατηρούμε ότι στους 1000 γύρους έχουμε φτάσει περίπου στην τελική τιμή, οπότε δεν χρειάζονται παραπάνω γύροι.

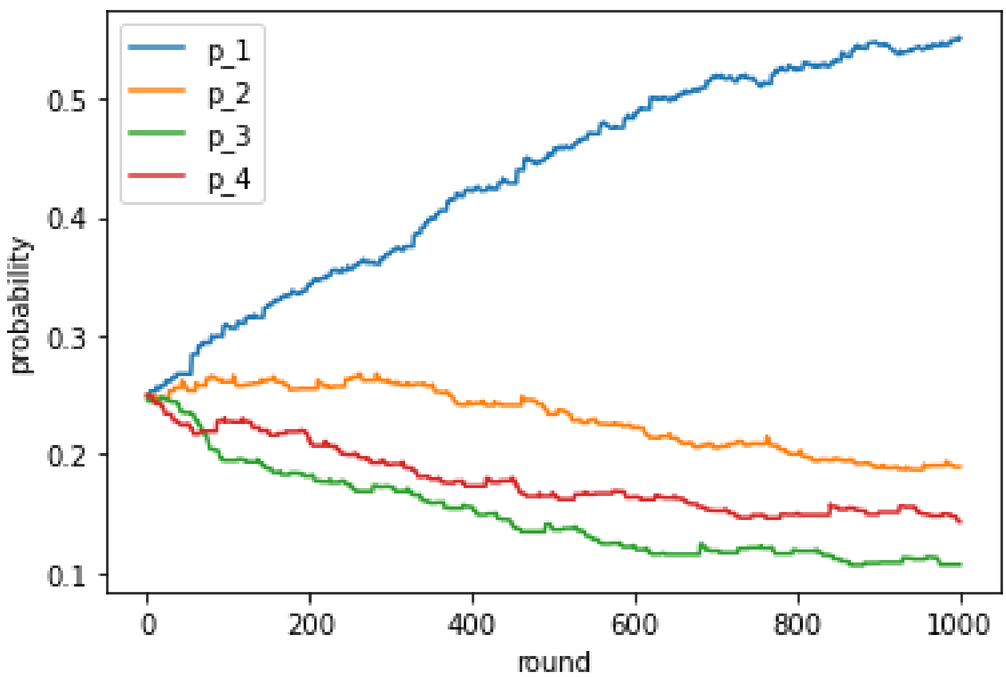


Εικόνα 4. Το αποτέλεσμα για 1000 γύρους.

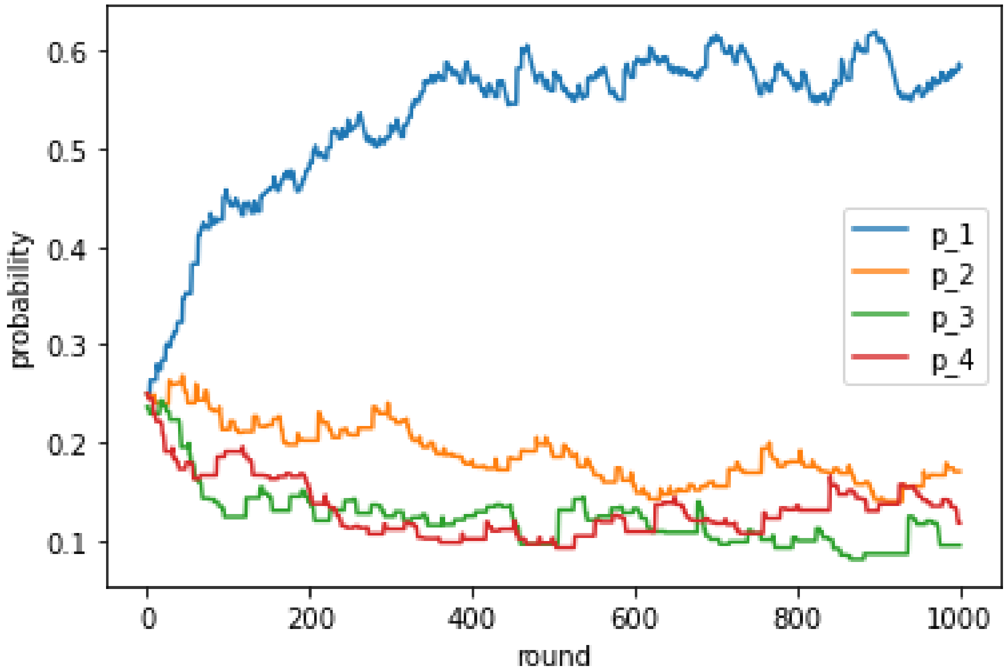


Εικόνα 5. Το αποτέλεσμα για 10.000 γύρους.

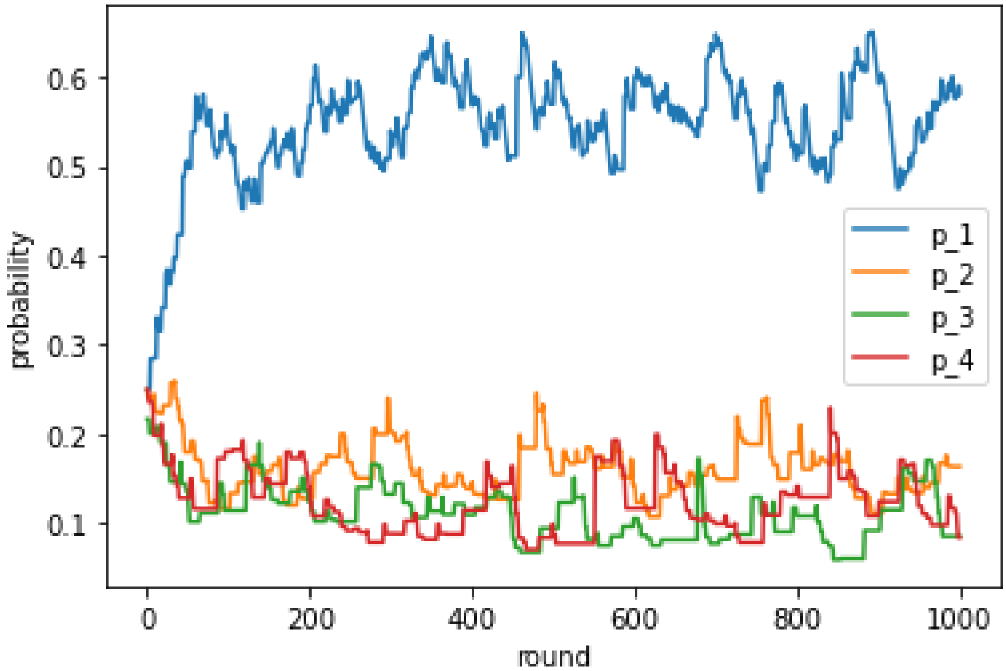
Επίσης, εξετάσαμε την παράμετρο e της εκθετικής συνάρτησης. Προηγουμένως (εικ. 2) η τιμή ήταν e = 0.1, ενώ στην εικόνα 6 δοκιμάσαμε τιμή e = 0.05, στην εικόνα 7 τιμή e = 0.2 και στην εικόνα 8 τιμή e = 0.5.



Εικόνα 6. Το αποτέλεσμα για e=0.05.



Εικόνα 7. Το αποτέλεσμα για e=0.2.



Εικόνα 8. Το αποτέλεσμα για e=0.5.

Παρατηρούμε ότι όσο πιο μεγάλη η τιμή της παραμέτρου e, τόσο πιο γρήγορα φτάνουμε στις τελικές τιμές. Ωστόσο, όταν η τιμή γίνει πολύ μεγάλη, έχουμε απότομες αλλαγές σε διαδοχικούς γύρους, δηλαδή έχουμε μια αστάθεια. Επομένως, μια ενδιάμεση τιμή, για παράδειγμα e = 0.1, φαίνεται να είναι η πιο κατάλληλη επιλογή.

4.4. Πείραμα 2.

Στη συνέχεια δοκιμάζουμε ένα παιχνίδι με δύο παίκτες. Αν L1 οι κανονικοποιημένες απώλειες του πρώτου παίκτη, θεωρούμε ότι οι απώλειες του δεύτερου παίκτη μπορούν να βρεθούν με τον τύπο:

L2 = 1 – L1

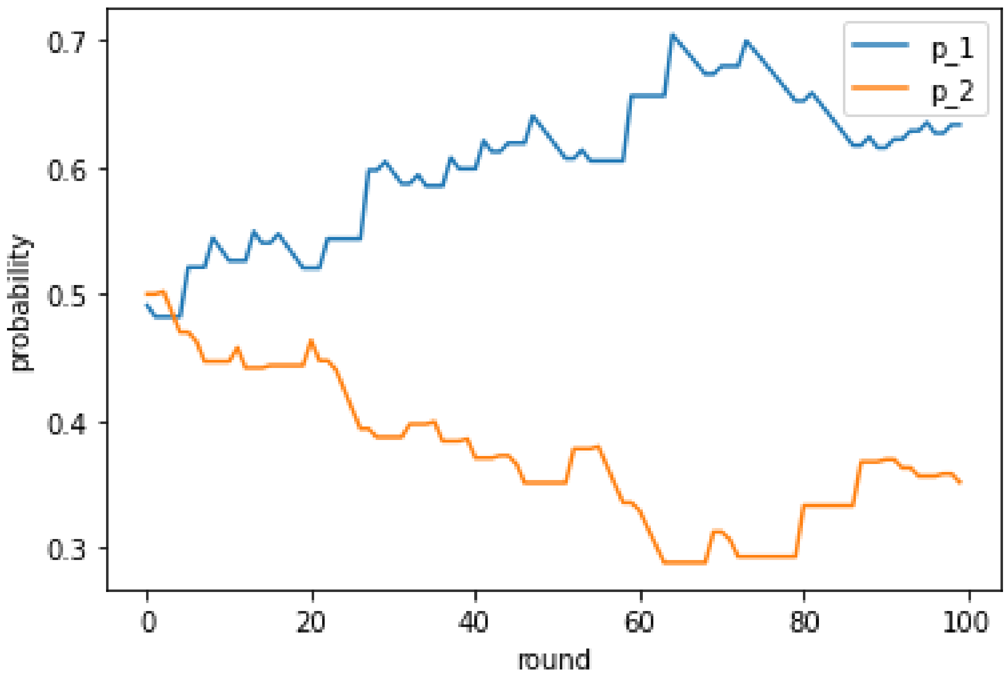
Αυτό ισχύει για τα παιχνίδια μηδενικού αθροίσματος, όπου οι απώλειες ενός παίκτη ισούνται με τα κέρδη του άλλου. Εκτελούμε τον αλγόριθμο hedge για T = 100 επαναλήψεις και παράμετρο e = 0.1. Για να δούμε τι συμβαίνει σε αυτή την περίπτωση, δοκιμάζουμε ένα παιχνίδι με N = 2 επιλογές με τυχαίες αρχικές τιμές απωλειών. Επειδή οι απώλειες είναι σταθερές στο χρόνο, μπορούμε να εκτελέσουμε τον αλγόριθμο δύο φορές, μια για κάθε παίκτη.

Ένα παράδειγμα εκτέλεσης φαίνεται στην εικόνα 9 και οι πορείες των απωλειών στις εικόνες 10 και 11 για τον κάθε παίκτη. Όπως είναι αναμενόμενο, οι πιθανότητες του παίκτη 1 είναι μεγαλύτερες για την 1η επιλογή με τις μικρότερες απώλειες (μπλε γραμμή στην εικ. 9) και αντίστροφα, για τον παίκτη 2, η 2η επιλογή έχει μεγαλύτερη πιθανότητα (πορτοκαλί γραμμή στην εικ. 10).

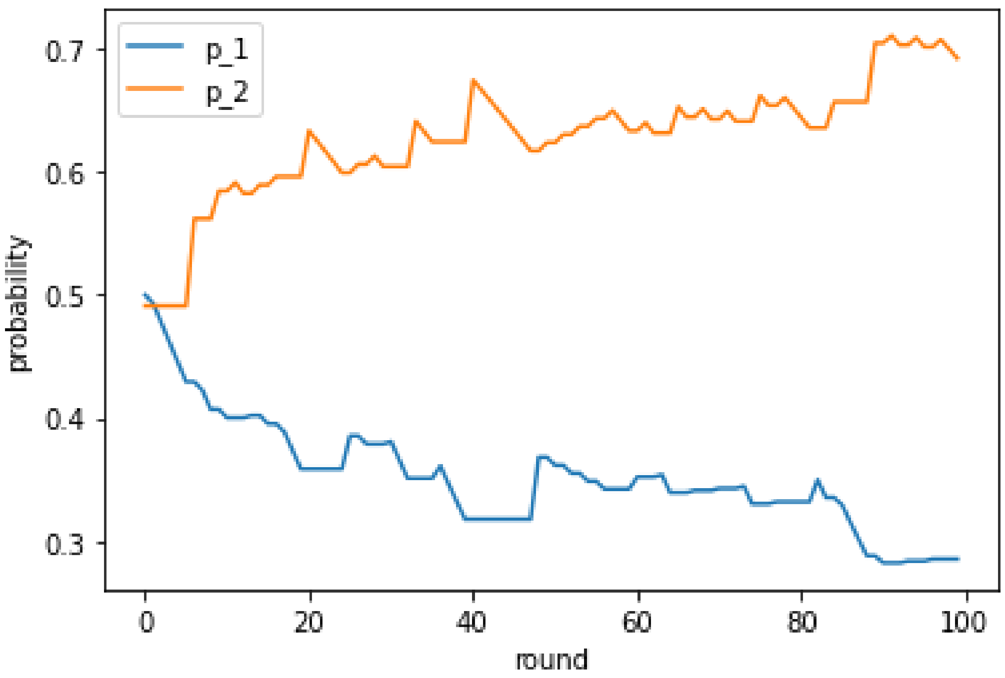
Παρατηρούμε ότι στο συγκεκριμένο παιχνίδι, κερδίζει ο παίκτης 2 μιας και έχει μεγαλύτερη ανταμοιβή (R = 0,55 έναντι R = 0,45), παρόλο που και οι δύο έχουν πρόσβαση στην ίδια πληροφορία. Αυτό προφανώς συμβαίνει λόγω του τυχαίου τρόπου που ο hedge διαλέγει μια ενέργεια σε κάθε βήμα, οπότε ο παίκτης 2 συνέκλινε πιο γρήγορα στις σωστές τιμές, μιας και έχει μεγαλύτερη τιμή (0,70) από τον παίκτη 1 (0,64) για την επιλογή με την μικρότερη απώλεια.

```
*******************************
Player 1
initial losses:
[0.35521948 0.64478052]
sum: 1.0
x: [0.64183263 0.34348429] ---sum x: 0.9853169240311811
W: [0.06370959 0.0333324 ] ---sum W: 0.09704199208228184
Reward 0.4494634341607108
Player 2
initial losses:
[0.64478052 0.35521948]
sum: 1.0
x: [0.27800661 0.70010654] ---sum x: 0.9781131508257941
W: [0.0292996  0.06840042] ---sum W: 0.09770002222996374
Reward 0.5501684227056733
```

Εικόνα 9. Έξοδος του προγράμματος.



Εικόνα 10. Το διάνυσμα απωλειών είναι [0.36, 0.64]. Οι αντίστοιχες πιθανότητες που μαθαίνει ο αλγόριθμος hedge φαίνονται με μπλε και πορτοκαλί χρώμα αντίστοιχα.



Εικόνα 11. Για τον αντίπαλο, το διάνυσμα απωλειών είναι [0.64, 0.36]. Οι αντίστοιχες πιθανότητες που μαθαίνει ο αλγόριθμος hedge φαίνονται με μπλε και πορτοκαλί χρώμα αντίστοιχα.

4.5. Πείραμα 3.

Πριν, το παιχνίδι ήταν συμμετρικό και κάθε παίκτης κατέληγε να εκτελεί με μεγάλη πιθανότητα τη μια ενέργεια και ο άλλος παίκτης την άλλη (καθώς η πρώτη ενέργεια ωφελούσε περισσότερο τον πρώτο παίκτη και η δεύτερη τον δεύτερο παίκτη). Σε αυτό το πείραμα δοκιμάζουμε 3 παίκτες με 2 ενέργειες. Αρχικά, παράγουμε τυχαία τις απώλειες για κάθε παίκτη, ωστόσο τις κανονικοποιούμε ανά ενέργεια (δηλαδή το άθροισμα των απωλειών των τριών παικτών είναι 1 ανά ενέργεια). Στη συνέχεια, ο κάθε παίκτης εκτελεί τον αλγόριθμο hedge.

Στην εικόνα 12 φαίνεται ένα παράδειγμα εκτέλεσης. Οι απώλειες για τους τρεις παίκτες είναι (0.37, 0.43), (0.25, 0.35), (0.38, 0.21) αντίστοιχα. Στις εικόνες 13-15 φαίνονται οι πιθανότητες που έμαθαν οι παίκτες. Προφανώς, ο κάθε παίκτης μαθαίνει να προτιμά τις χαμηλότερες απώλειες. Ωστόσο, στον 2^{ο} παίκτη παρατηρούμε μια αστάθεια, με αρκετές εναλλαγές του ποια ενέργεια προτιμάται. Αυτό πιθανόν να οφείλεται σε τυχαίες επιλογές της «χειρότερης» ενέργειας. Πιο πιθανόν όμως, αυτό είναι αποτέλεσμα του ότι οι απώλειες που του δόθηκαν είναι σχετικά κοντά σε τιμή, οπότε δεν αναπτύσσει ισχυρή «προτίμηση» σε κάποια από αυτές. Σημειώνεται ότι ο

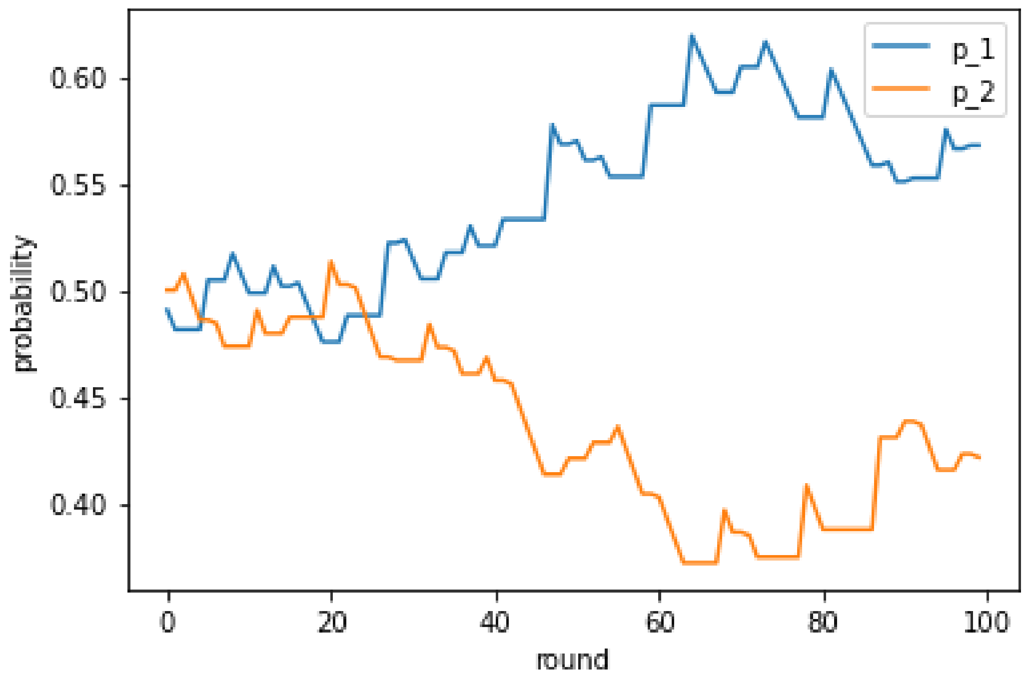
3ος παίκτης έχει κάπως παρόμοιες τιμές απωλειών με τον 2ο παίκτη, αλλά με μεγαλύτερη διαφορά και καταφέρνει να μάθει επιτυχώς τις πιθανότητες.

Εξετάζοντας την ανταμοιβή, ο πρώτος παίκτης έχει μεγαλύτερη ανταμοιβή (0.39), μετά ο δεύτερος (0.29) και τέλος ο τρίτος παίκτης (0.27). Για τον πρώτο παίκτη, να αναφέρουμε ότι το άθροισμα των απωλειών είναι μεγαλύτερο από τους άλλους δύο, αφού η κανονικοποίηση έγινε ανά ενέργεια. Οπότε, αυτός είναι ο λόγος που έχει καλύτερη ανταμοιβή. Ωστόσο, για τους άλλους δύο παίκτες είναι αξιοσημείωτο ότι ο 3ος παίκτης έχει λίγο μικρότερη ανταμοιβή από τον δεύτερο παίκτη, παρότι έμαθε καλύτερα τις πιθανότητες.

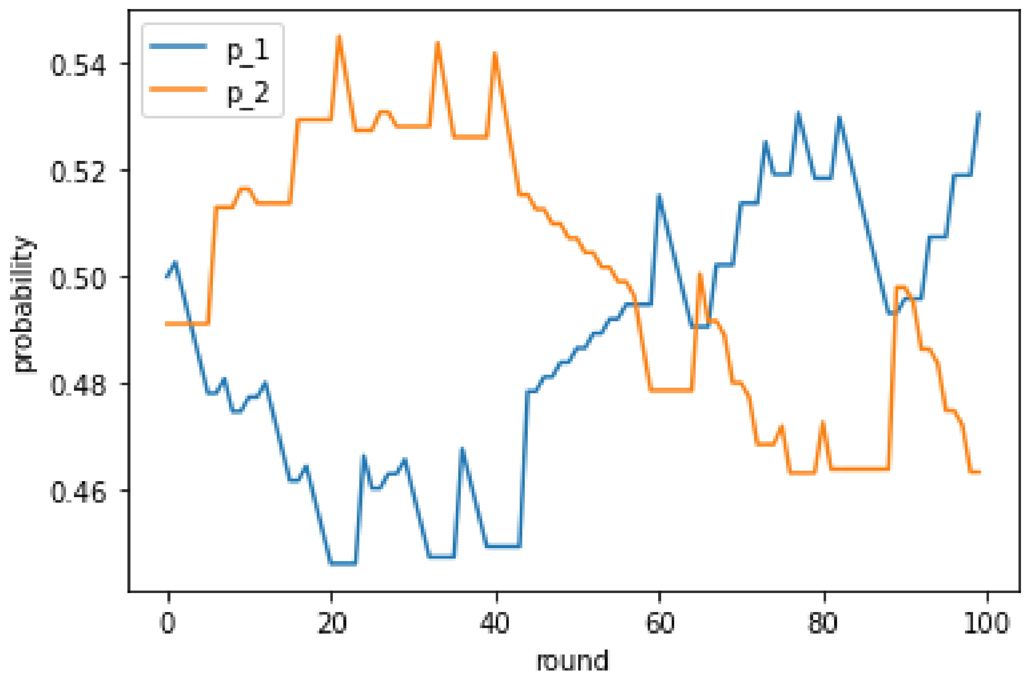
Οπότε, παρατηρούμε ότι όταν υπάρχει ένα πιο πολύπλοκο παιχνίδι με περισσότερο ανταγωνισμό, καθώς οι παίκτες είναι περισσότεροι από τις επιλογές, δύο παίκτες κατέληξαν με μια προτίμηση, αλλά ένας παίκτης δεν κατάφερε να βρει μια ξεκάθαρη στρατηγική.

```
Player 1
initial losses:
[0.36948083 0.4327502 ]
sum: 0.8022310313197127
x: [0.56784688 0.42156631] ---sum x: 0.9894131843316303
W: [0.07882293 0.05744667] ---sum W: 0.136269603245505
Reward 0.39224143985658577
Player 2
initial losses:
[0.24568576 0.3541016 ]
sum: 0.599787360269436
x: [0.53051902 0.46336661] ---sum x: 0.9938856315630706
W: [0.12324808 0.10906797] ---sum W: 0.23231604825385643
Reward 0.29441982711337056
Player 3
initial losses:
[0.38483341 0.2131482 ]
sum: 0.5979816084108514
x: [0.36367815 0.62737032] ---sum x: 0.9910484695669579
W: [0.08848719 0.15482462] ---sum W: 0.24331180743054426
Reward 0.273678356049911
```

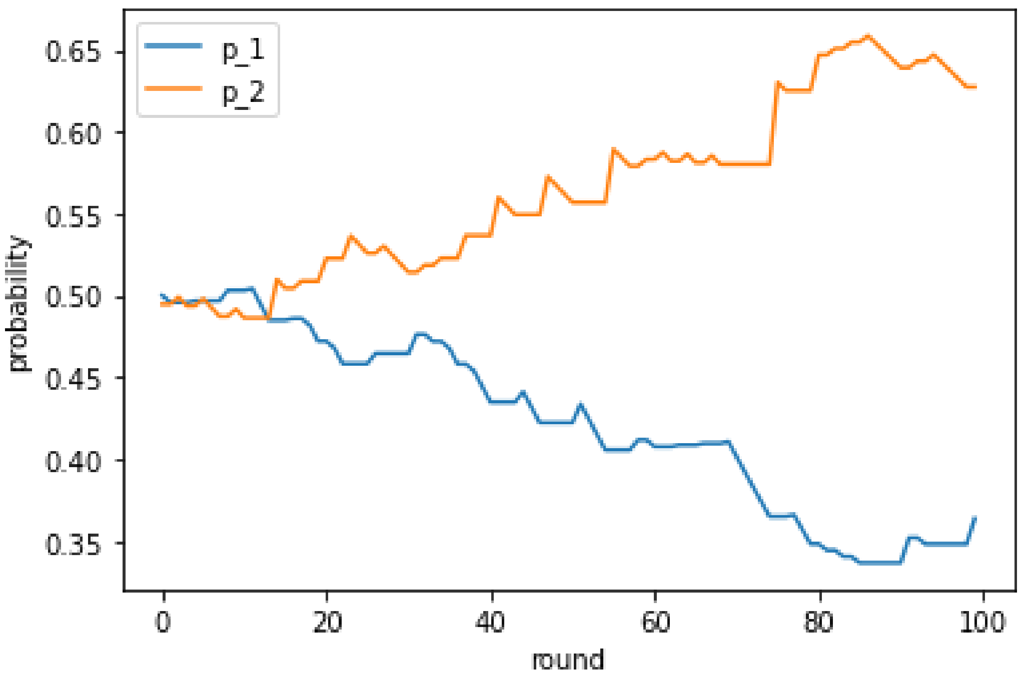
Εικόνα 12. Παράδειγμα με 3 παίκτες και δύο ενέργειες.



Εικόνα 13. Οι πιθανότητες που μαθαίνει ο 1^{ος} παίκτης.



Εικόνα 14. Οι πιθανότητες που μαθαίνει ο 2^{ος} παίκτης.



Εικόνα 15. Οι πιθανότητες που μαθαίνει ο 3ος παίκτης.

4.6. Πείραμα 4.

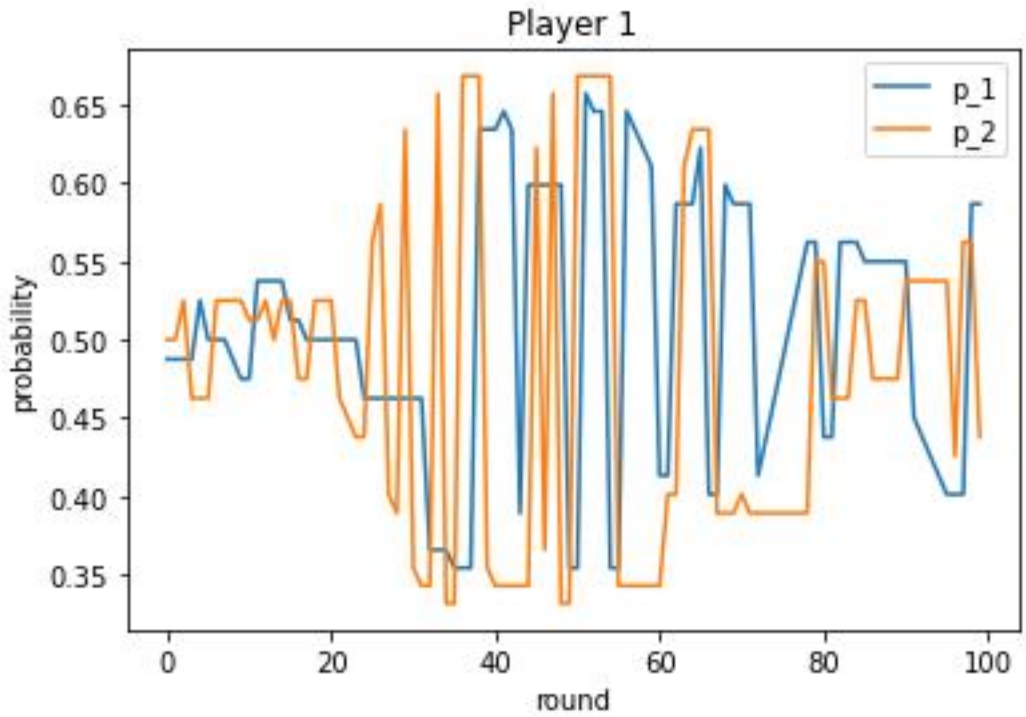
Σε αυτό το πείραμα έχουμε δύο παίκτες που ανταγωνίζονται για έναν πόρο ο οποίος έχει συγκεκριμένη χωρητικότητα με δύο επιλογές. Για παράδειγμα, θα μπορούσε να είναι ένα δίκτυο με δύο διαδρομές. Ο κάθε παίκτης διαλέγει μια διαδρομή. Αν διαλέξουν διαφορετική διαδρομή παίρνουν χωρητικότητα 1 ο καθένας. Αν όμως διαλέξουν την ίδια διαδρομή, παίρνουν χωρητικότητα 0.5 ο καθένας.

Σε αυτό το πείραμα αλλάζουμε αρκετά τον τρόπο υπολογισμού σε σχέση με τα προηγούμενα παραδείγματα, όπου οι απώλειες ήταν σταθερές για κάθε παίκτη, οπότε ο καθένας εκτελούσε ανεξάρτητα τον αλγόριθμο. Σε αυτή την περίπτωση αλγόριθμος hedge εκτελείται ταυτόχρονα από τους δύο παίκτες. Σε κάθε γύρο κάνουν μια επιλογή και λαμβάνουν τις απώλειες. Οι απώλειες κωδικοποιούνται σε έναν πίνακα 2x2 όπου οι γραμμές και οι στήλες είναι η επιλογή του πρώτου και του δεύτερου παίκτη αντίστοιχα. Στην εικόνα 16 έχουμε ένα παράδειγμα εκτέλεσης, όπου δείχνουμε τον αρχικό πίνακα απωλειών και τις πιθανότητες που μαθαίνει ο κάθε παίκτης. Παρατηρούμε στις τελικές πιθανότητες μετά από Τ γύρους, ο πρώτος παίκτης προτιμά ελαφρώς την 1η διαδρομή και ο 2ος παίκτης τη 2η διαδρομή.

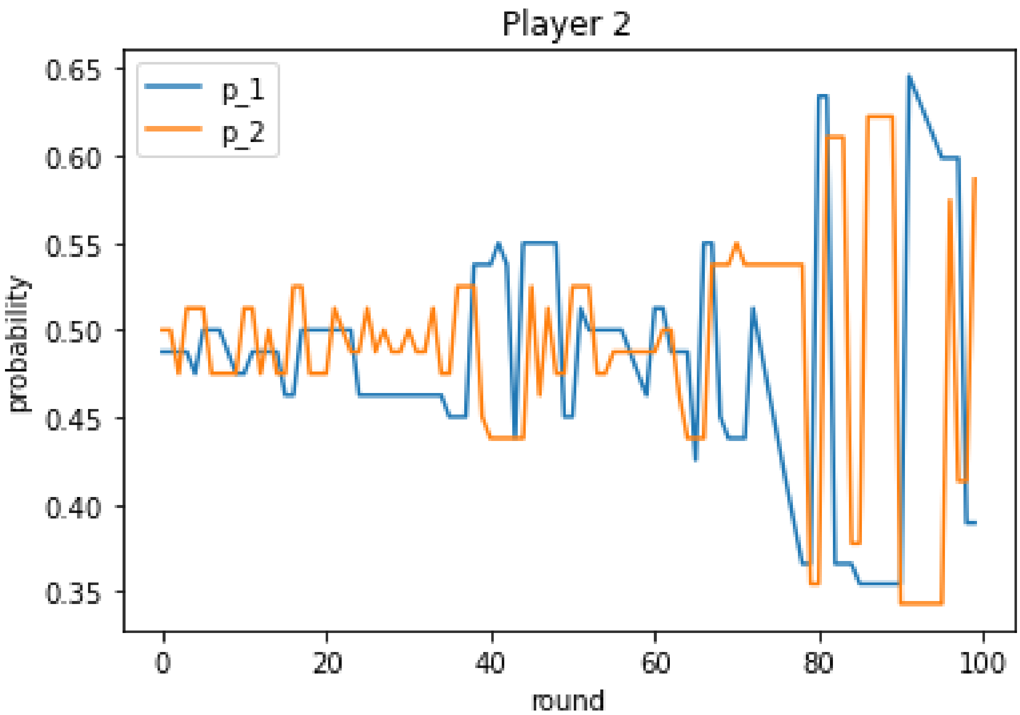
Στις εικόνες 17 και 18 βλέπουμε τις πιθανότητες που μαθαίνει ο κάθε παίκτης. Παρατηρούμε ότι έχουμε συνεχείς εναλλαγές στην προτίμηση κάθε παίκτη. Αυτό πιθανόν να συμβαίνει γιατί το πρόβλημα είναι πιο δύσκολο: το αποτέλεσμα επηρεάζεται από την επιλογή του άλλου παίκτη. Για παράδειγμα, ο πρώτος παίκτης μαθαίνει να προτιμά την 1^{η} διαδρομή, αλλά κάποια στιγμή ο δεύτερος παίκτης επιλέγει τυχαία την 1^{η} διαδρομή. Τότε, μειώνεται η ανταμοιβή του πρώτου παίκτη, παρότι έκανε την ίδια επιλογή με την προηγούμενη φορά, ενώ αν αλλάξει τυχαία επιλογή, θα αυξηθεί η ανταμοιβή του. Οπότε, είναι αρκετά εύκολο να αντιστραφούν οι πιθανότητες.

```
initial losses:
[[0.5 1. ]
 [1.  0.5]]
Player 1
x: [0.58661758 0.4378235 ] ---sum x: 1.0244410780315318
W: [0.01063987 0.01366186] ---sum W: 0.024301729442834832
Reward 1.0244410780315318
Player 2
x: [0.38936077 0.58661758] ---sum x: 0.9759783449681081
W: [0.01436232 0.01012096] ---sum W: 0.024483275550021874
Reward 0.9759783449681081
```

Εικόνα 16. Παράδειγμα με δύο παίκτες και δύο ενέργειες για έναν κοινό πόρο.



Εικόνα 17. Οι πιθανότητες που μαθαίνει ο $1^{ος}$ παίκτης.

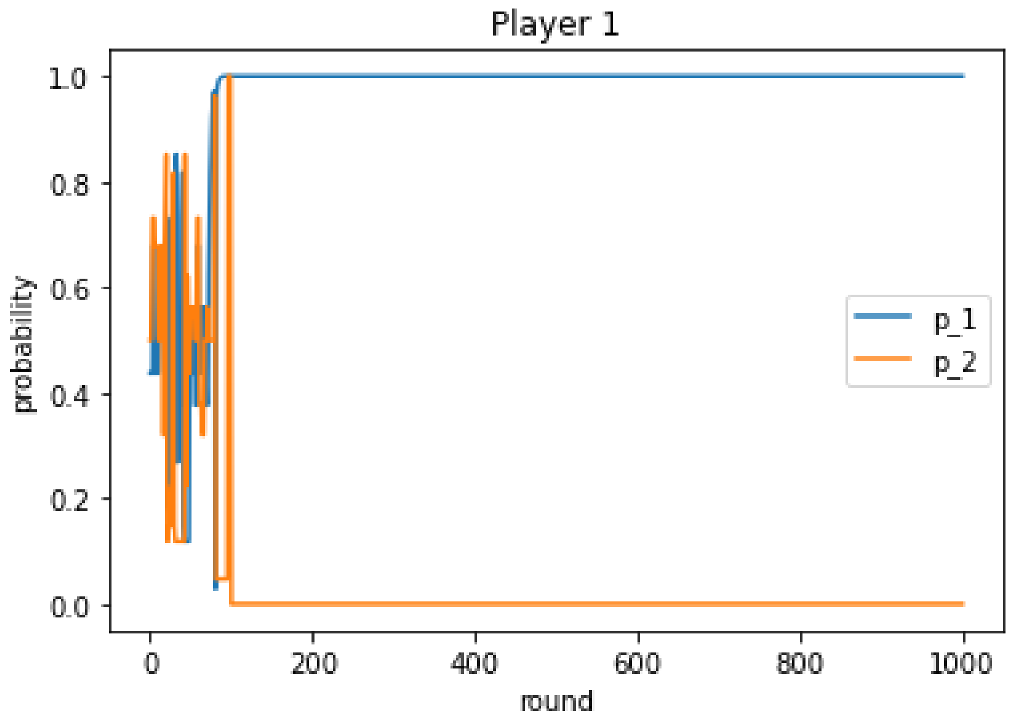


Εικόνα 18. Οι πιθανότητες που μαθαίνει ο $2^{ος}$ παίκτης.

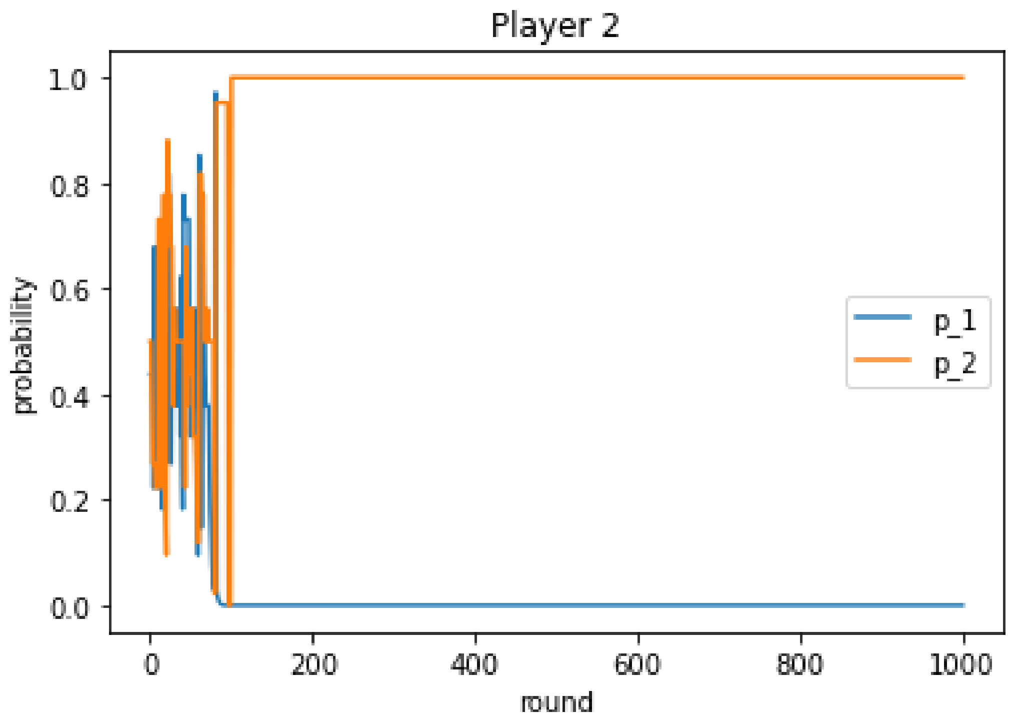
Ωστόσο, για να αποτρέψουμε τις ανεπιθύμητες εναλλαγές, δοκιμάσαμε να αλλάξουμε τις παραμέτρους του αλγορίθμου hedge. Για αυτό το σκοπό, αυξήσαμε τον αριθμό γύρων σε 1000 και την παράμετρο e = 0.5. Το αποτέλεσμα της εκτέλεσης φαίνεται στην εικόνα 19 και οι πιθανότητες που μαθαίνουν οι παίκτες στις εικόνες 20 και 21. Πράγματι, λόγω της μεγάλης τιμής της παραμέτρου e κάποια στιγμή οι πιθανότητες οδηγούνται πρακτικά στο 1 για τη μία επιλογή και στο 0 για την άλλη, οπότε ο κάθε παίκτης καταλήγει να επιλέξει μια διαδρομή.

```
initial losses:
[[0.5 1. ]
 [1.  0.5]]
Player 1
x: [1.00000000e+00 9.61024155e-05] ---sum x: 1.0000961024154995
W: [6.97245279e-207 1.19118483e-007] ---sum W: 1.1911848337509083e-07
Reward 1.0000961024154995
Player 2
x: [1.30606474e-200 9.99978555e-001] ---sum x: 0.999978555051579
W: [2.52173831e-007 3.29355349e-207] ---sum W: 2.5217383128394417e-07
Reward 0.999978555051579
```

Εικόνα 19. Παράδειγμα με δύο παίκτες και δύο ενέργειες για έναν κοινό πόρο.



Εικόνα 20. Οι πιθανότητες που μαθαίνει ο 1ος παίκτης.



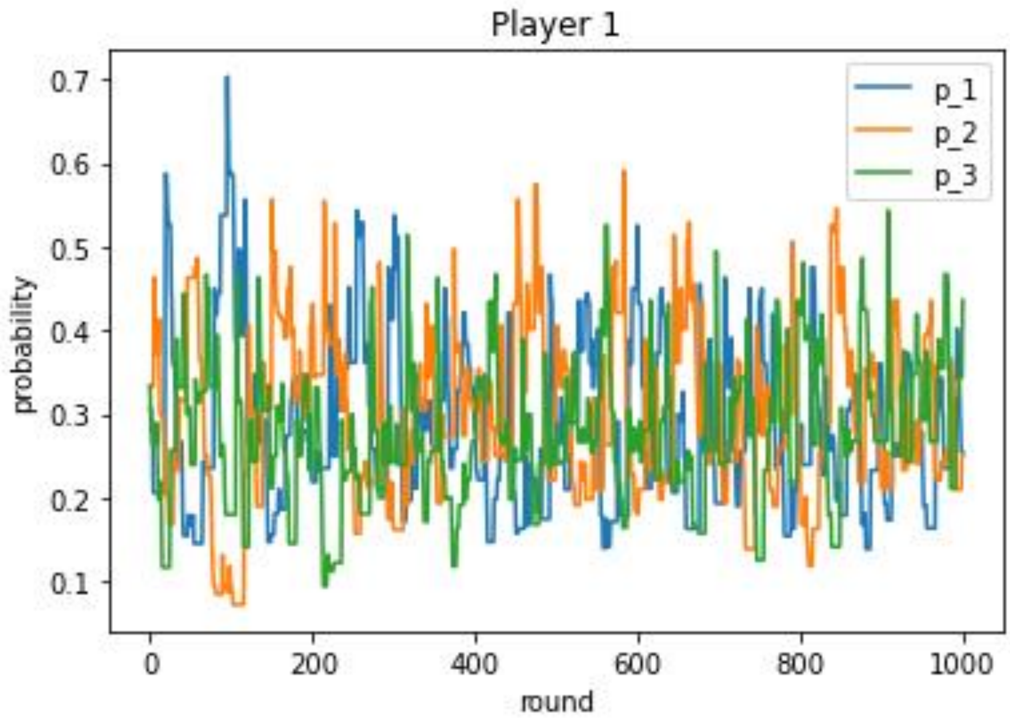
Εικόνα 21. Οι πιθανότητες που μαθαίνει ο 1ος παίκτης.

4.7. Πείραμα 5.

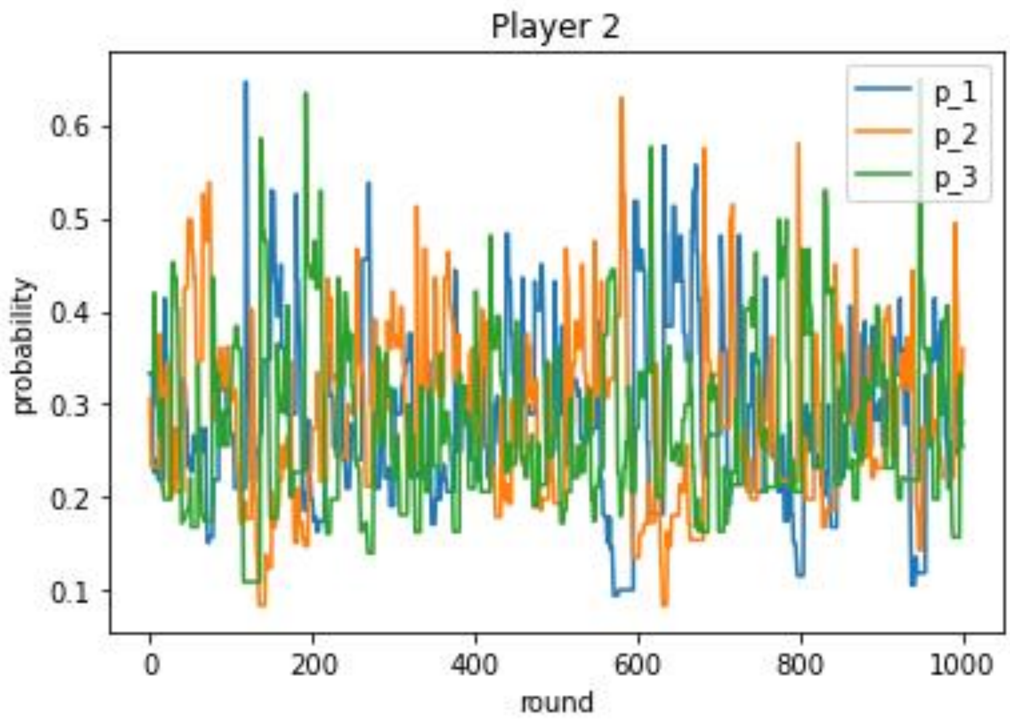
Σε αυτό το πείραμα θα έχουμε 4 παίκτες οι οποίοι διαγωνίζονται για περιορισμένους πόρους, π.χ. για πρόσβαση σε ένα δίκτυο με N εναλλακτικές διαδρομές. Ομοίως με πριν, ο κάθε παίκτης διαλέγει μια διαδρομή. Αν διαλέξουν διαφορετικές διαδρομές παίρνουν χωρητικότητα 1 ο καθένας. Αν όμως K παίκτες διαλέξουν την ίδια διαδρομή, παίρνουν χωρητικότητα 1/K ο καθένας.

Θα μεταβάλλουμε το N και θα εξετάσουμε τρεις περιπτώσεις, η πρώτη με N= 3 (λιγότερες από τους παίκτες), η δεύτερη με N = 4 (ίσες με τους παίκτες) και την τελευταία με N = 5 (περισσότερες από τους παίκτες).

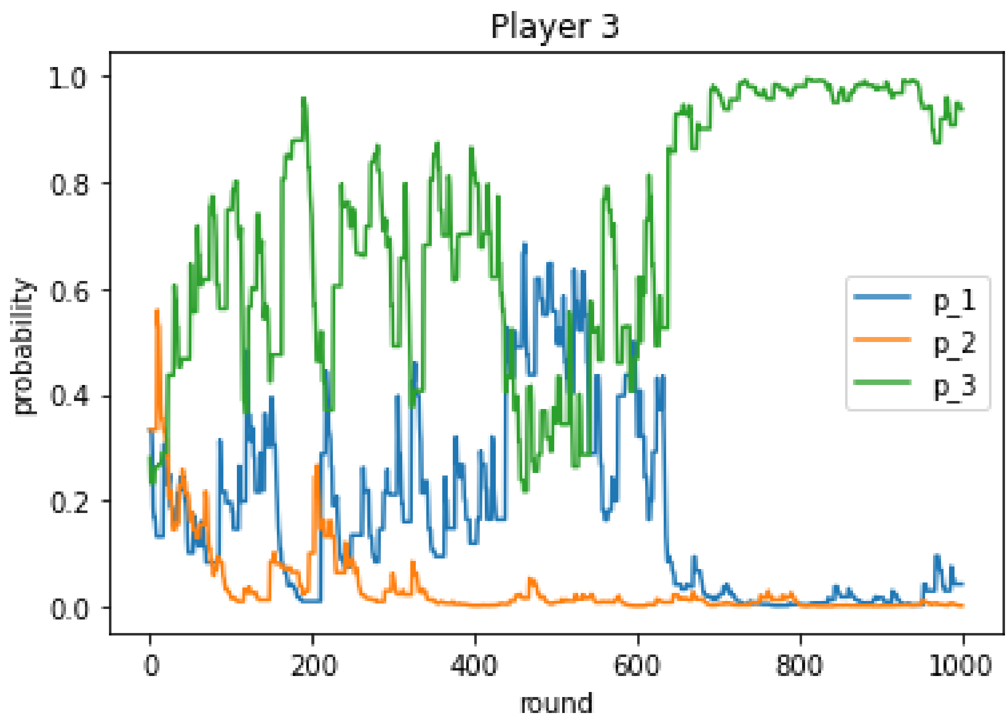
Παρακάτω στις εικόνες 22-24 φαίνονται τα αποτελέσματα για 3 διαδρομές. Παρατηρούμε ότι ο παίκτης 3 και ο παίκτης 4 κατέληξαν με μια συγκεκριμένη διαδρομή (την 3η και 2η αντίστοιχα), ωστόσο οι δύο πρώτοι παίκτες αλλάζουν συνεχώς διαδρομή μεταξύ των τριών, μιας και δεν έχουν καταφέρει να επιτύχουν χαμηλή τιμή της συνάρτησης απωλειών (και δεν πρόκειται να αλλάξει, ακόμα και αν τρέχαμε παραπάνω γύρους, αφού μένει η πρώτη διαδρομή και για τους δύο ή θα χρησιμοποιήσουν διαδρομή που ήδη έχει επιλέξει άλλος παίκτης). Ως προς την ανταμοιβή (εικ. 25), είναι μέγιστη για τον 3ο παίκτη (R = 0.491), σχετικά χαμηλή για τον 4ο παίκτη (R = 0.247), ίσως επειδή ο 3ος κατέληξε νωρίτερα να επιλέγει μια διαδρομή. Ο 2ος παίκτης έχει υψηλή ανταμοιβή (R=0.447) χωρίς να είναι προφανής ο λόγος (ίσως επιλέγει σε μεγάλα διαστήματα την 1η διαδρομή, αλλά αυτό δεν συμβαίνει συστηματικά), ενώ ο 1ος παίκτης έχει επίσης (R=0.235).



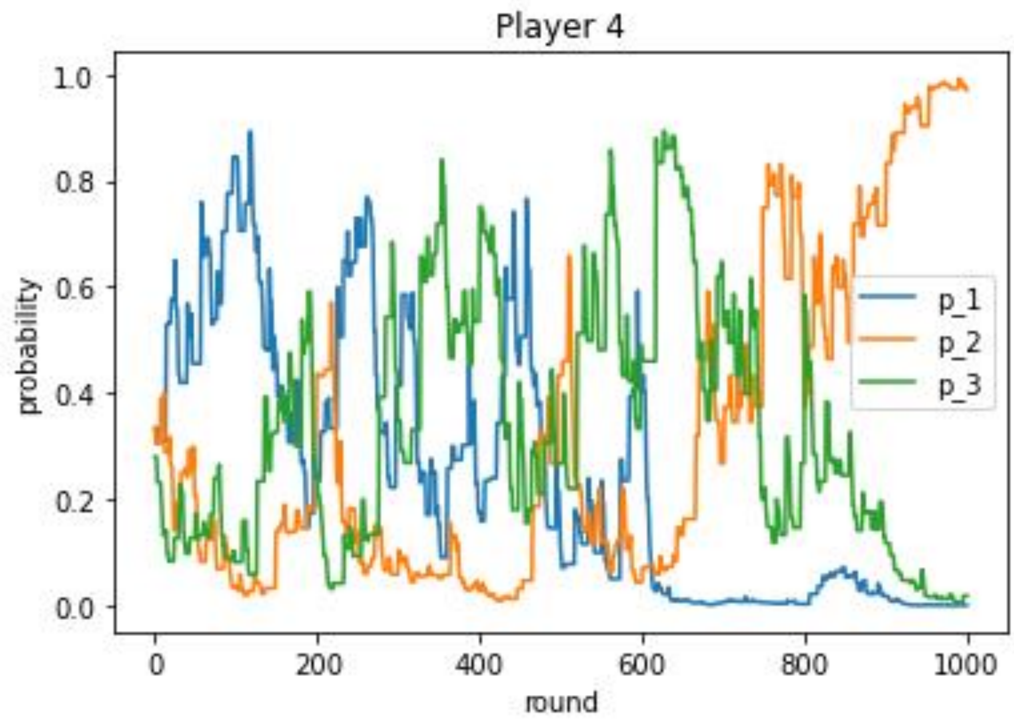
Εικόνα 22. Οι πιθανότητες που μαθαίνει ο 1ος παίκτης για N = 3 διαδρομές.



Εικόνα 23. Οι πιθανότητες που μαθαίνει ο 2ος παίκτης για N = 3 διαδρομές.



Εικόνα 24. Οι πιθανότητες που μαθαίνει ο 3ος παίκτης για N = 3 διαδρομές.

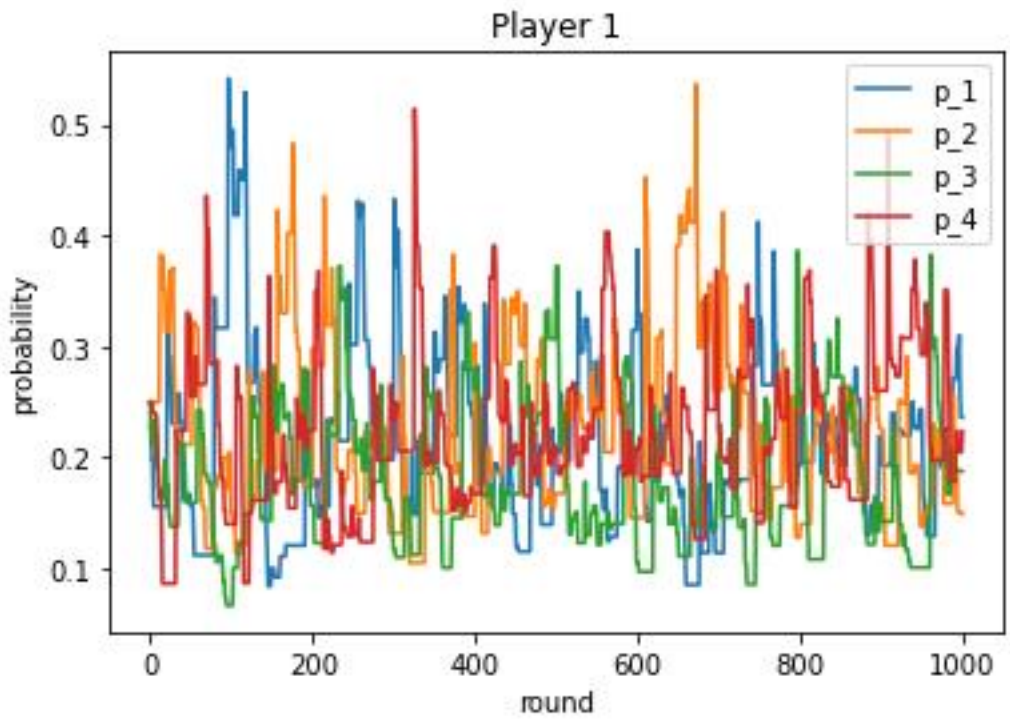


Εικόνα 25. Οι πιθανότητες που μαθαίνει ο 4^{ος} παίκτης για N = 3 διαδρομές.

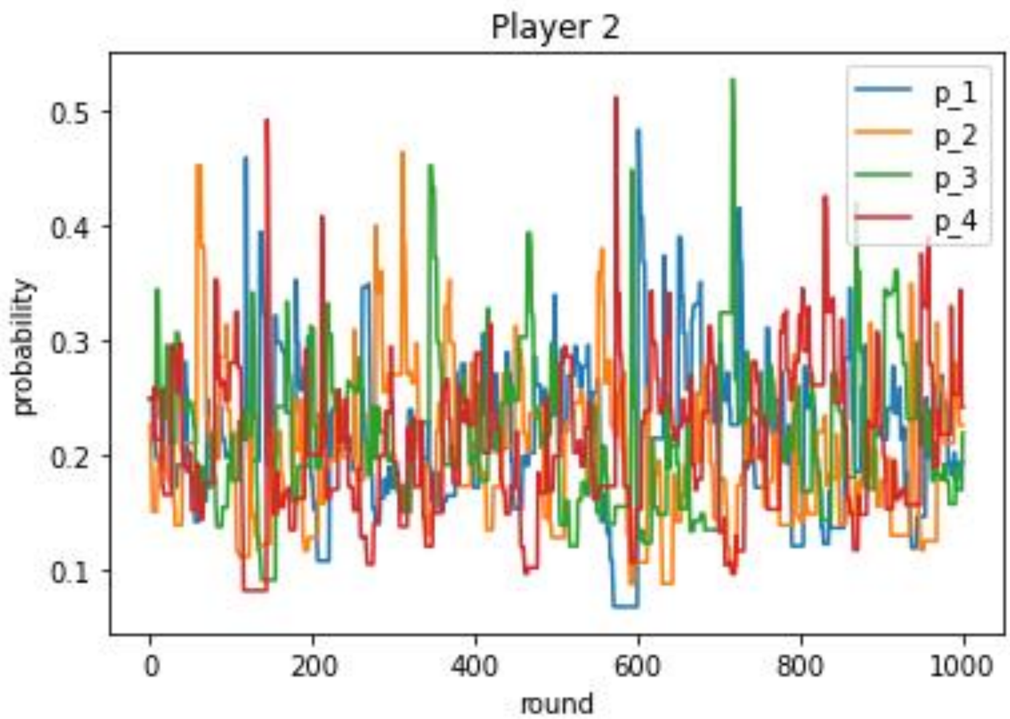
```
Player 1
x: [0.25427521 0.24991288 0.43595401] ---sum x: 0.9401421021265045
W: [2.05862061e-37 1.81672631e-37 2.99527531e-37] ---sum W: 6.870622238127897e-37
Reward 0.23503552553162613
Player 2
x: [0.28026507 0.35986747 0.25427521] ---sum x: 0.894407745267133
W: [1.49125977e-38 1.49125977e-38 1.16139428e-38] ---sum W: 4.143913813101633e-38
Reward 0.4472038726335665
Player 3
x: [4.19236067e-02 8.56356059e-04 9.38410715e-01] ---sum x: 0.9811906777455515
W: [3.32745031e-39 4.74635295e-41 5.20500803e-38] ---sum W: 5.542499415496154e-38
Reward 0.49059533887277573
Player 4
x: [3.68850429e-04 9.70282078e-01 1.79802867e-02] ---sum x: 0.9886312154515291
W: [3.26211750e-41 8.58160746e-38 2.59142091e-39] ---sum W: 8.844011663956099e-38
Reward 0.24715780386288227
```

Εικόνα 26. Παράδειγμα εκτέλεσης με 4 παίκτες και 3 διαδρομές.

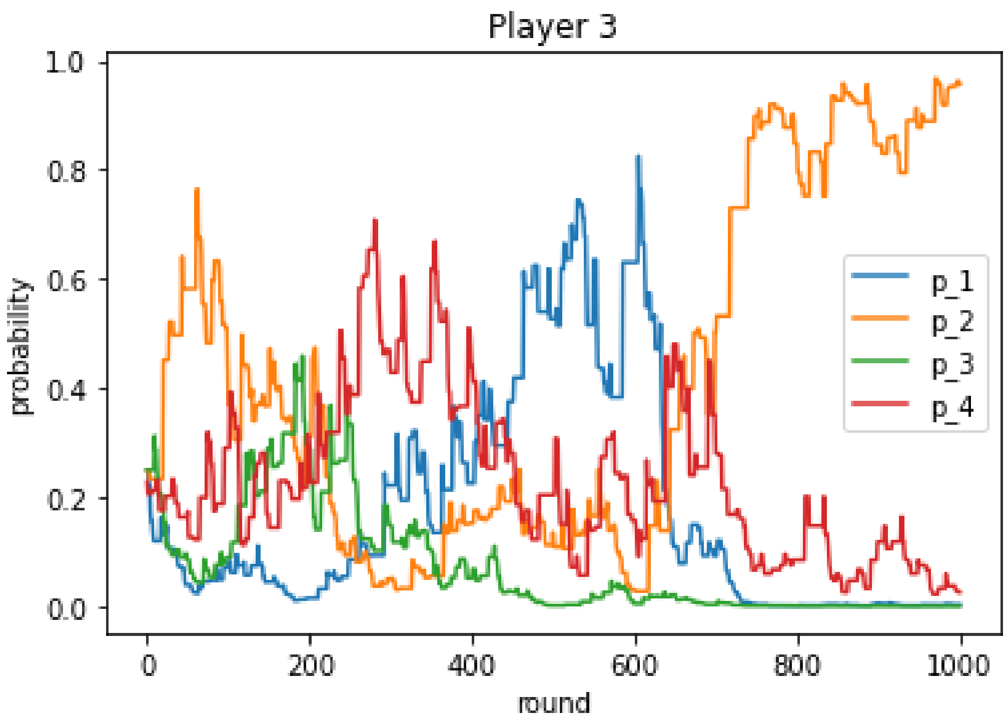
Στη συνέχεια επαναλαμβάνουμε με 4 διαδρομές, όπως φαίνεται στις εικόνες 27-31. Εδώ, αν και το βέλτιστο είναι ο κάθε παίκτης να καταλήξει με μια διαδρομή, αυτό δεν συνέβη. Παρατηρούμε ότι οι πιθανότητες των 4 παικτών αλλάζουν συνεχώς τιμές χωρίς να συγκλίνουν κάπου. Η μόνη εξαίρεση είναι ο παίκτης 3 και 4 που προς το τέλος φαίνεται να προτιμούν την 4^{η} διαδρομή. Οι ανταμοιβές επίσης είναι σχετικά κοντά και κυμαίνονται στο διάστημα 0,20-0,247.



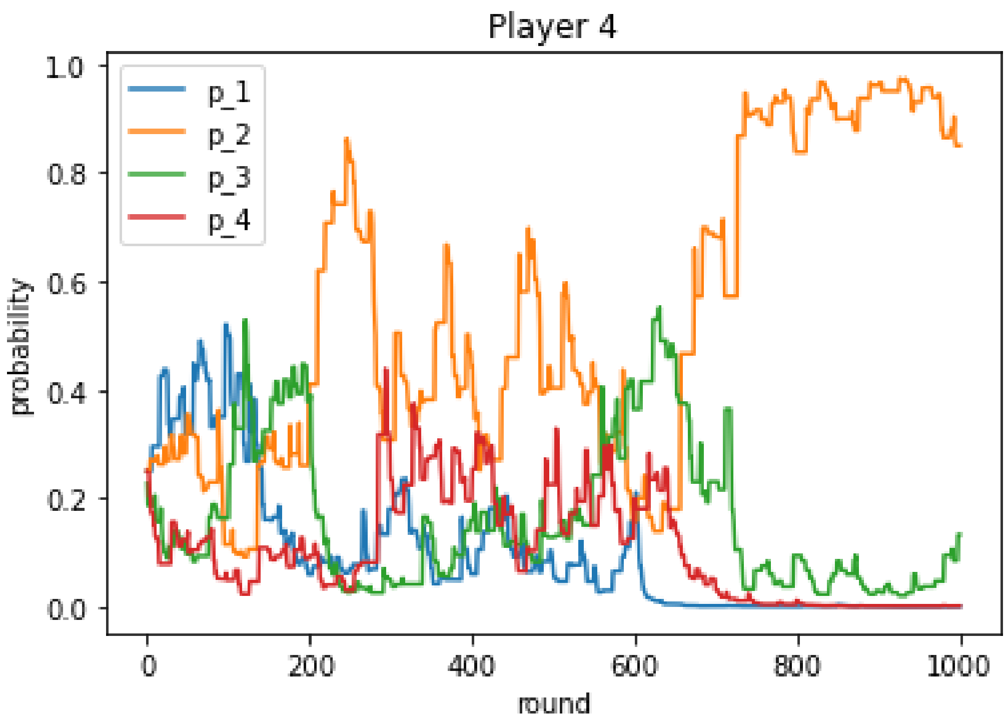
Εικόνα 27. Οι πιθανότητες που μαθαίνει ο 1^{ος} παίκτης για N = 4 διαδρομές.



Εικόνα 28. Οι πιθανότητες που μαθαίνει ο 2^{ος} παίκτης για N = 4 διαδρομές.



Εικόνα 29. Οι πιθανότητες που μαθαίνει ο 3ος παίκτης για N = 4 διαδρομές.

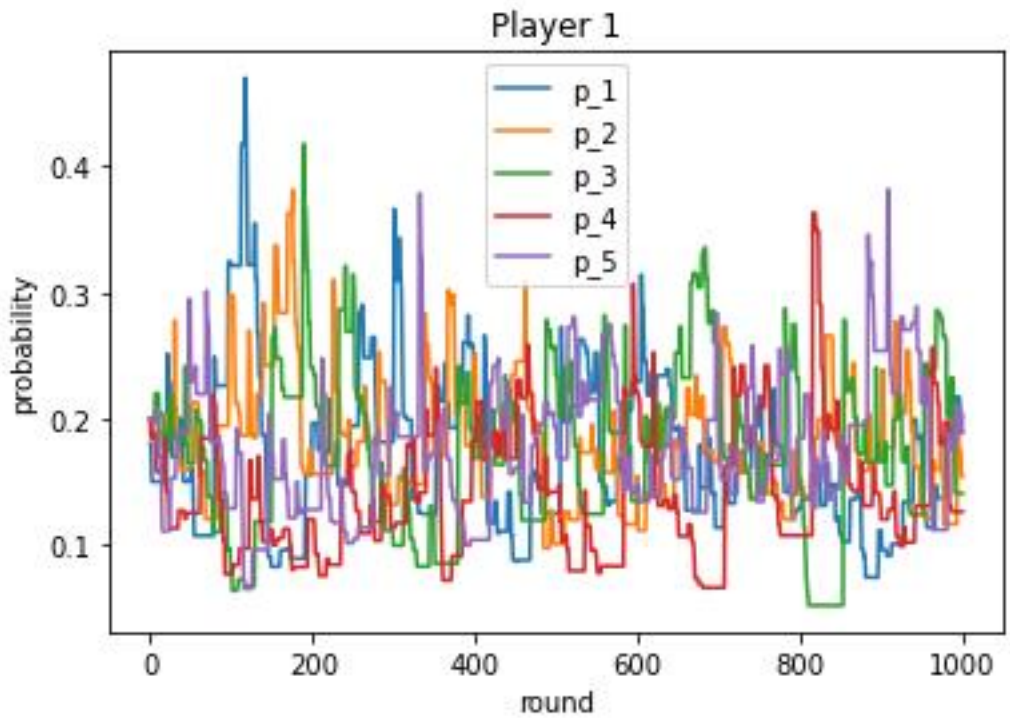


Εικόνα 30. Οι πιθανότητες που μαθαίνει ο 4ος παίκτης για N = 4 διαδρομές.

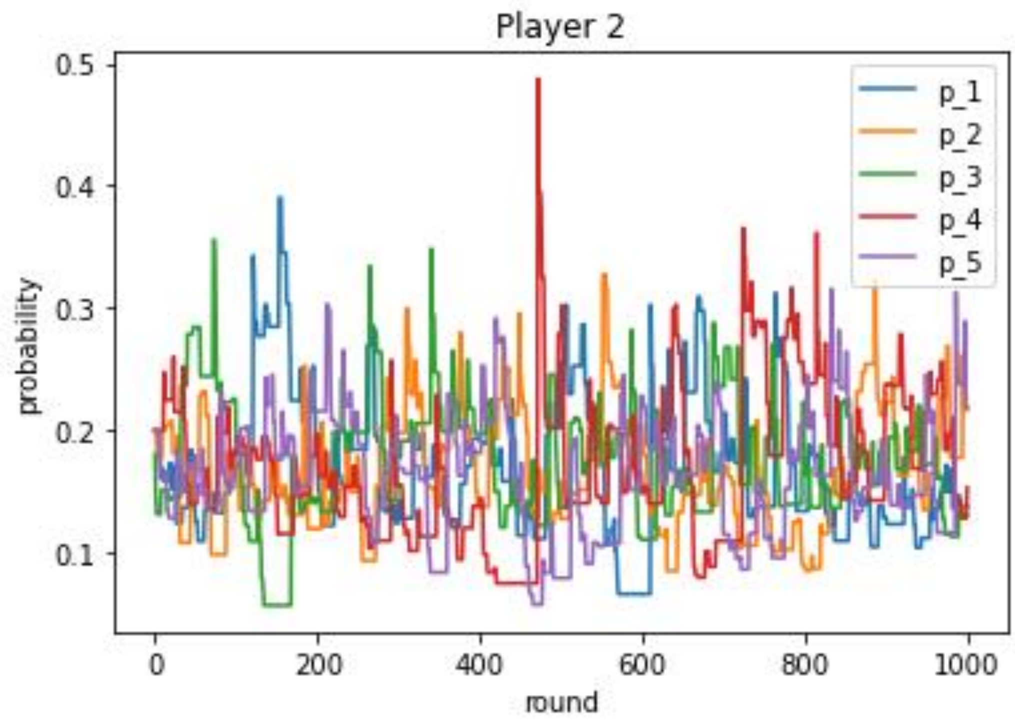
```
Player 1
x: [0.23619487 0.1493358  0.18785561 0.22375574] ---sum x: 0.7971420271969456
W: [1.05478079e-25 6.39756889e-26 1.53469699e-25 9.30840780e-26] ---sum W: 4.160075453840909e-25
Reward 0.1992855067992364
Player 2
x: [0.19408916 0.22656651 0.21976465 0.24194631] ---sum x: 0.8823666183114578
W: [2.07698652e-26 2.35353407e-26 1.83293417e-26 2.07698652e-26] ---sum W: 8.340441289322596e-26
Reward 0.22059165457786445
Player 3
x: [3.05453749e-03 9.57157175e-01 2.35056755e-05 2.58537480e-02] ---sum x: 0.9860889665349826
W: [3.02199756e-26 5.08221846e-24 2.03620594e-28 1.97058995e-25] ---sum W: 5.309701049697989e-24
Reward 0.24652224163374564
Player 4
x: [3.72900946e-04 8.50100775e-01 1.32712048e-01 1.14855875e-03] ---sum x: 0.9843342824618103
W: [5.53498160e-28 1.28498606e-24 1.97058995e-25 1.70489436e-27] ---sum W: 1.4843034481820322e-24
Reward 0.24608357061545258
```

Εικόνα 31. Παράδειγμα εκτέλεσης με 4 παίκτες και 4 διαδρομές.

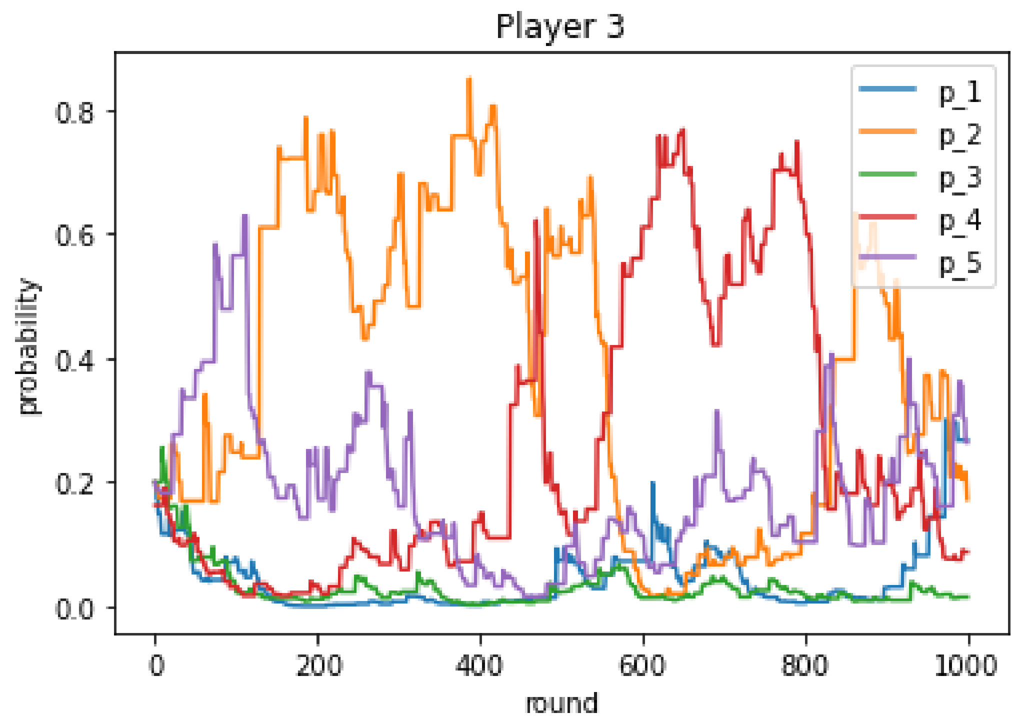
Τέλος, επαναλαμβάνουμε με 5 διαδρομές (εικ. 32-36). Ομοίως με πριν, ενώ υπάρχουν πολλές διαδρομές και θα μπορούσε ο κάθε παίκτης να καταλήξει σε μια δικιά του, δεν συμβαίνει, με εξαίρεση τον παίκτη 4 που διαλέγει την $2^{η}$ διαδρομή. Ο παίκτης 3 για κάποιο διάστημα είχε αρκετά υψηλές πιθανότητες για την 3ή και μετά για την $4^{η}$ διαδρομή, ωστόσο στην πορεία η πιθανότητες αυτές μειώθηκαν. Αυτό φαίνεται και στις ανταμοιβές, με τον παίκτη 1 να έχει R = 0.204, τον παίκτη 2 να έχει R = 0.214, τον παίκτη 3 με R = 0.202 και τον παίκτη 4 με R = 0.247.



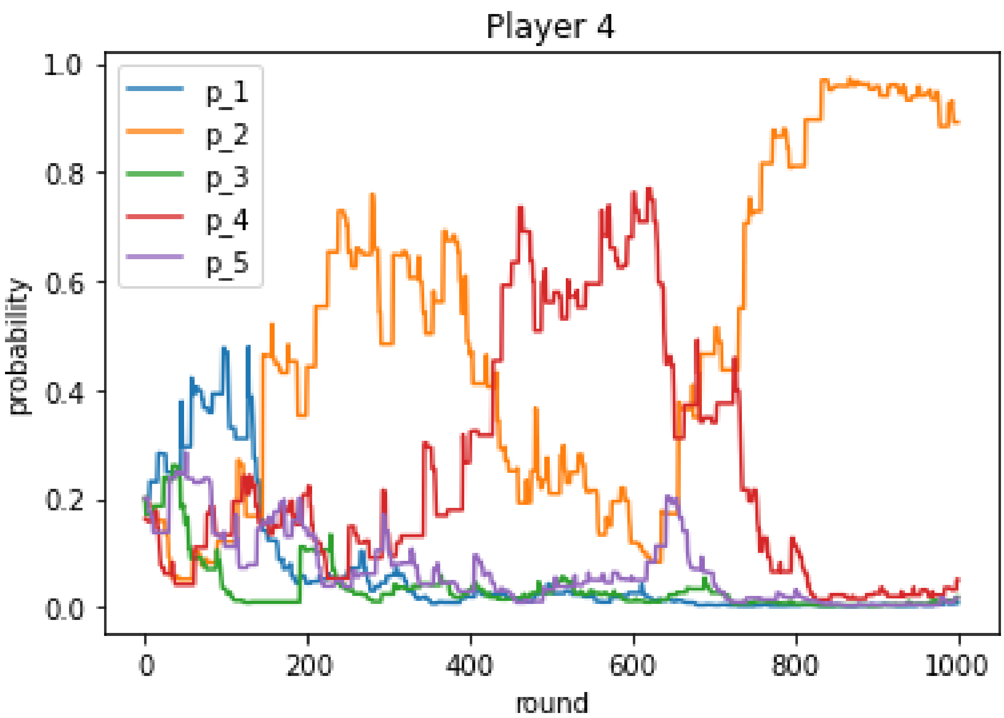
Εικόνα 32. Οι πιθανότητες που μαθαίνει ο $1^{ος}$ παίκτης για N = 5 διαδρομές.



Εικόνα 33. Οι πιθανότητες που μαθαίνει ο 2^{ος} παίκτης για N = 5 διαδρομές.



Εικόνα 34. Οι πιθανότητες που μαθαίνει ο 3^{ος} παίκτης για N = 5 διαδρομές.



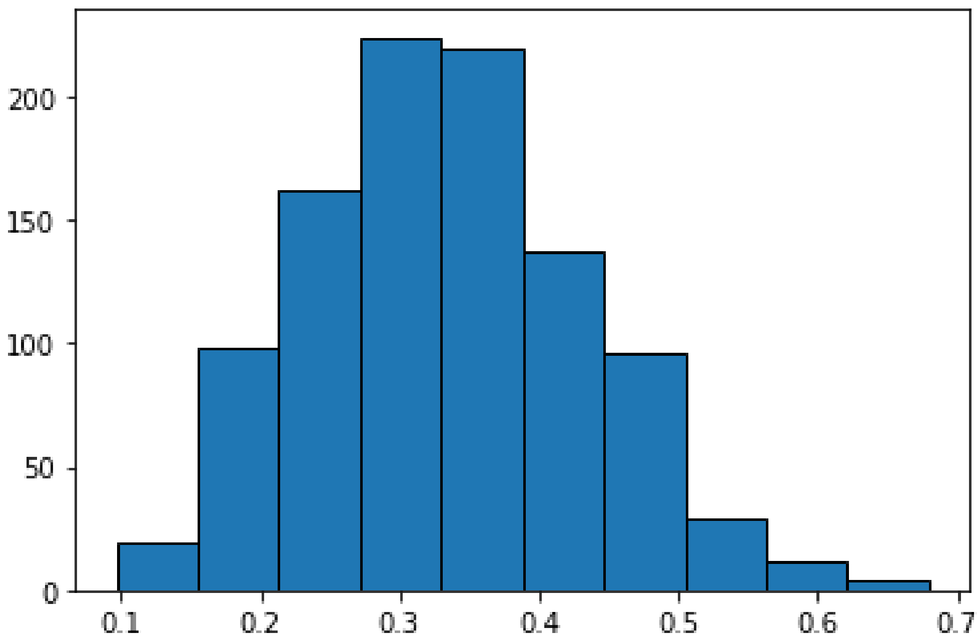
Εικόνα 35. Οι πιθανότητες που μαθαίνει ο $4^{ος}$ παίκτης για N = 5 διαδρομές.

```
Player 1
x: [0.18988675 0.15436783 0.14125666 0.12665877 0.20358517] ---sum x: 0.8157551783480983
W: [4.01356091e-19 3.12576438e-19 4.01356091e-19 4.54796034e-19
 4.01356091e-19] ---sum W: 1.9714407440122602e-18
Reward 0.20393879458702457
Player 2
x: [0.13756537 0.21699643 0.13073564 0.15335338 0.21866302] ---sum x: 0.857313840795397
W: [2.75847738e-19 3.54195507e-19 2.43434775e-19 2.14830435e-19
 3.12576438e-19] ---sum W: 1.4008848922688372e-18
Reward 0.21432846019884924
Player 3
x: [0.26776769 0.17272761 0.01517151 0.08782074 0.2640297 ] ---sum x: 0.8075172466402741
W: [5.15351422e-19 2.14830435e-19 2.90741377e-20 1.30301245e-19
 3.54195507e-19] ---sum W: 1.2437527463144319e-18
Reward 0.20187931166006853
Player 4
x: [0.0078357  0.89222933 0.01865053 0.05104194 0.01637739] ---sum x: 0.9861348890385544
W: [3.29453142e-20 3.80796057e-18 7.90317002e-20 2.14830435e-19
 6.97452306e-20] ---sum W: 4.2045132461373575e-18
Reward 0.2465337222596386
```

Εικόνα 36. Παράδειγμα εκτέλεσης με 4 παίκτες και 5 διαδρομές.

Σε αυτό το πείραμα, επαναλαμβάνουμε για 1000 εκτελέσεις, ώστε να δημιουργηθούν κατανομές πιθανοτήτων.

Για αρχή παρατηρούμε στην εικόνα 36 ότι η κατανομή των πιθανότητων παρουσιάζει περίπου κανονική κατανομή.



Εικόνα 37. Κατανομή πιθανοτήτων επιλογής διαδρομής από παίκτη (παράδειγμα για διαδρομή και παίκτη 1).

Ο μέσος όρος των πιθανοτήτων για Ν=3 διαδρομές φαίνεται στον πίνακα 1, ενώ σε παρένθεση απεικονίζεται το 95% διάστημα εμπιστοσύνης. Επαναλαμβάνουμε για Ν=4 και Ν=5 διαδρομές (πίνακας 2 και πίνακας 3 αντίστοιχα). Παρατηρούμε ότι οι πιθανότητες είναι ισοκατανεμημένες. Αυτό σημαίνει ότι δεν αναπτύσσεται κάποια προτίμηση σε κάποια διαδρομή. Για Ν=3 εξηγείται από το γεγονός ότι θα υπάρχουν πάντα συγκρούσεις, αφού οι παίκτες είναι περισσότεροι από τις διαδρομές, ωστόσο για τις άλλες περιπτώσεις δεν αναμέναμε να παρατηρήσουμε κάτι τέτοιο. Ωστόσο, ειδικά στην περίπτωση περισσότερων διαδρομών από παίκτες, μπορεί ο κάθε παίκτης να διαλέγει μια διαδρομή αποκλειστικά για τον εαυτό του, μεγιστοποιώντας τις ανταμοιβές, αλλά σε κάθε επανάληψη του πειράματος να είναι διαφορετική η επιλογή διαδρομής, οπότε να μην μπορεί να φανεί αυτό το αποτέλεσμα στους πίνακες.

Πίνακας 1. Μέσες τιμές πιθανοτήτων για Ν=3 διαδρομές και 95% διάστημα εμπιστοσύνης.

Παίκτης	**Διαδρομή**		
	1	**2**	**3**
1	0.335 (0.187 - 0.502)	0.333 (0.182 - 0.512)	0.333 (0.176 - 0.516)
2	0.335 (0.176 - 0.507)	0.332 (0.172 - 0.504)	0.333 (0.186 - 0.510)
3	0.330	0.336	0.334

	(0.176 - 0.499)	(0.182 - 0.516)	(0.187 - 0.506)
4	0.328 (0.177 - 0.505)	0.338 (0.185 - 0.508)	0.333 (0.176 - 0.519)

Πίνακας 2. Μέσες τιμές πιθανοτήτων για N=4 διαδρομές και 95% διάστημα εμπιστοσύνης.

Παίκτης	**Διαδρομή**			
	1	**2**	**3**	**4**
1	0.252 (0.141 - 0.384)	0.248 (0.138 - 0.382)	0.250 (0.140 - 0.389)	0.246 (0.134 - 0.377)
2	0.256 (0.144 - 0.393)	0.251 (0.137 - 0.389)	0.250 (0.139 - 0.388)	0.254 (0.140 - 0.389)
3	0.246 (0.128 - 0.389)	0.247 (0.129 - 0.379)	0.249 (0.140 - 0.386)	0.248 (0.136 - 0.380)
4	0.246 (0.129 - 0.380)	0.254 (0.130 - 0.395)	0.251 (0.141 - 0.381)	0.251 (0.137 - 0.379)

Πίνακας 3. Μέσες τιμές πιθανοτήτων για N=5 διαδρομές και 95% διάστημα εμπιστοσύνης.

Παίκτης	**Διαδρομή**				
	1	**2**	**3**	**4**	**5**
1	0.201 (0.111 - 0.313)	0.202 (0.113 - 0.309)	0.200 (0.112 - 0.311)	0.199 (0.110 - 0.315)	0.198 (0.110 - 0.304)
2	0.200 (0.109 - 0.305)	0.199 (0.111 - 0.311)	0.201 (0.111 - 0.305)	0.199 (0.110 - 0.303)	0.202 (0.111 - 0.310)
3	0.198 (0.110 - 0.298)	0.203 (0.115 - 0.320)	0.200 (0.117 - 0.304)	0.198 (0.111 - 0.309)	0.202 (0.115 - 0.309)
4	0.197 (0.111 - 0.306)	0.200 (0.111 - 0.307)	0.203 (0.111 - 0.319)	0.198 (0.113 - 0.309)	0.202 (0.113 - 0.308)

5. Συμπεράσματα

Στην εργασία αυτή μελετήθηκε ο αλγόριθμος Hedge, γνωστός και ως αλγόριθμος Multiplicative Weights Update (MWU), και εξετάστηκε η απόδοσή του σε διάφορα πειραματικά σενάρια με πολλαπλούς παίκτες και επιλογές. Τα κύρια ευρήματα από τα πειράματα είναι τα εξής:

- **Συμπεριφορά Αλγορίθμου με Έναν Παίκτη**: Ο αλγόριθμος Hedge προσαρμόζει επιτυχώς τις πιθανότητες επιλογών με βάση τις απώλειες, δίνοντας μεγαλύτερη προτίμηση στις ενέργειες με μικρότερες απώλειες. Η αύξηση των γύρων (T) επιβεβαίωσε ότι οι πιθανότητες σταθεροποιούνται, υποδεικνύοντας ότι ένας επαρκής αριθμός γύρων είναι απαραίτητος για τη σύγκλιση του αλγορίθμου. Η παράμετρος eee επηρεάζει τον ρυθμό σύγκλισης, με ενδιάμεσες τιμές να εξασφαλίζουν σταθερότητα χωρίς απότομες αλλαγές στις πιθανότητες.
- **Συμπεριφορά Αλγορίθμου σε Παιχνίδια Μηδενικού Αθροίσματος**: Στα πειράματα με δύο παίκτες, παρατηρήθηκε ότι οι παίκτες τείνουν να μαθαίνουν να προτιμούν τις ενέργειες με τις μικρότερες απώλειες. Παρόλο που οι παίκτες είχαν πρόσβαση στην ίδια πληροφορία, η τυχαία επιλογή ενέργειας κάθε φορά οδήγησε σε διαφορές στις αποδόσεις, δείχνοντας τη σημασία της τυχαιότητας στη διαδικασία μάθησης.
- **Παιχνίδια με Περισσότερους Παίκτες**: Σε πειράματα με τρεις και τέσσερις παίκτες, παρατηρήθηκε ότι οι παίκτες δυσκολεύονται να σταθεροποιήσουν τις επιλογές τους όταν οι απώλειες είναι κοντά σε τιμή. Σε πιο πολύπλοκα περιβάλλοντα, οι παίκτες δεν κατάφεραν πάντα να βρουν μια ξεκάθαρη στρατηγική, ιδιαίτερα όταν ο αριθμός των επιλογών ήταν μικρότερος από τον αριθμό των παικτών.
- **Ανταγωνισμός για Κοινούς Πόρους**: Στα πειράματα με δύο παίκτες που ανταγωνίζονται για έναν κοινό πόρο, η ταυτόχρονη εκτέλεση του αλγορίθμου Hedge έδειξε συνεχείς εναλλαγές στις προτιμήσεις των παικτών, ιδιαίτερα όταν οι πιθανότητες επηρεάζονταν από τις επιλογές του αντιπάλου. Η αύξηση του αριθμού των γύρων και της παραμέτρου eee οδήγησε σε σταθερότερες επιλογές.

- **Μεγάλος Αριθμός Παικτών και Διαδρομών**: Στα πειράματα με τέσσερις παίκτες και διαφορετικό αριθμό διαδρομών, οι παίκτες δεν κατάφεραν να αναπτύξουν ξεκάθαρες προτιμήσεις για κάποιες διαδρομές, ιδιαίτερα όταν ο αριθμός των διαδρομών ήταν μεγαλύτερος από τον αριθμό των παικτών. Αυτό μπορεί να οφείλεται στην πιθανότητα τυχαίων επιλογών να μην συγκλίνουν σε σταθερές προτιμήσεις, κάτι που φαίνεται να επηρεάζεται και από την ισοκατανομή των πιθανοτήτων σε επαναλαμβανόμενα πειράματα.

Συνολικά, η μελέτη έδειξε ότι ο αλγόριθμος Hedge είναι αποτελεσματικός σε περιβάλλοντα με πολλούς παίκτες και επιλογές, προσαρμόζοντας τις πιθανότητες με βάση τις απώλειες και επιτρέποντας τη μάθηση στρατηγικών. Ωστόσο, η σταθερότητα και η σύγκλιση των πιθανοτήτων επηρεάζονται από τον αριθμό των γύρων, την παράμετρο eee, και την πολυπλοκότητα του περιβάλλοντος. Για πρακτικές εφαρμογές, η σωστή ρύθμιση των παραμέτρων είναι κρίσιμη για την επίτευξη βέλτιστων αποτελεσμάτων.

Βιβλιογραφία

Aleti, A., & Moser, I. (2016). A systematic literature review of adaptive parameter control methods for evolutionary algorithms. *ACM Computing Surveys (CSUR), 49*(3), 1-35. https://doi.org/10.1145/2996355

Anagnostou, M. E., & Lambrou, M. A. (2014). Playing against hedge. *Int'l J. of Communications, Network and System Sciences*, *7*(12), 497.

Bugday, A., Ozsoy, A., Öztaner, S. M., & Sever, H. (2019). Creating consensus group using online learning based reputation in blockchain networks. *Pervasive and Mobile Computing*, *59*, 101056.

Erven, T., Koolen, W. M., Rooij, S., & Grünwald, P. (2011). Adaptive hedge. *Advances in Neural Information Processing Systems*, *24*.

Ge, Z., Xi, M., & Li, Y. (2019). A literature review of the adaptive algorithms adopted in adaptive learning systems. In *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)* (pp. 254-258). https://doi.org/10.1109/SIPROCESS.2019.8868893

Giannopoulos, G. (2004). The application of information and communication technologies in transport. *European Journal of Operational Research, 152*(2), 302-320. https://doi.org/10.1016/S0377-2217(03)00026-2

Gu, H., Bo, H., & Ren, L. (2016). A review of research on traffic modeling for NoC. In *Proceedings of the 2016 International Conference on Electronics, Electrical Engineering, and Communication Science (ICEEECS)*. https://doi.org/10.2991/ICEEECS-16.2016.124

Guo, X., & Mu, Y. (2023a). The Optimal Strategy against Hedge Algorithm in Repeated Games. *arXiv preprint arXiv:2312.09472*.

Guo, X., & Mu, Y. (2023b). Taking Myopic Best Response Against The Hedge Algorithm. In *2023 42nd Chinese Control Conference (CCC)* (pp. 8154-8158). IEEE.

Huang, H., Cheng, Y., & Weibel, R. (2019). Transport mode detection based on mobile phone network data: A systematic review. *Transportation Research Part C: Emerging Technologies*. https://doi.org/10.1016/J.TRC.2019.02.008

Jain, L., Seera, M., Lim, C., & Balasubramaniam, P. (2014). A review of online learning in supervised neural networks. *Neural Computing and Applications, 25*(3-4), 491-509. https://doi.org/10.1007/s00521-013-1534-4

Joshi, M., & Hadi, T. (2015). A review of network traffic analysis and prediction techniques. *arXiv preprint arXiv:1507.05722*.

Kafi, M., Djenouri, D., Ben-othman, J., & Badache, N. (2014). Congestion control protocols in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials, 16*(3), 1369-1390. https://doi.org/10.1109/SURV.2014.021714.00123

Krichene, W., Balandat, M., Tomlin, C., & Bayen, A. (2015, June). The hedge algorithm on a continuum. In *International Conference on Machine Learning* (pp. 824-832). PMLR.

Kumar, K., Pandey, R., Karthik, M., Bhattacharjee, S., & George, N. (2021). Robust and sparsity-aware adaptive filters: A review. *Signal Processing, 189*, 108276. https://doi.org/10.1016/J.SIGPRO.2021.108276

Kushwaha, V., & Gupta, R. (2014). Congestion control for high-speed wired network: A systematic literature review. *Journal of Network and Computer Applications, 45*, 62-78. https://doi.org/10.1016/j.jnca.2014.07.005

Matnee, Y., Abooddy, C., & Mohammed, Z. (2018). Analyzing methods and opportunities in software-defined (SDN) networks for data traffic optimizations. *International Journal on Recent and Innovation Trends in Computing and Communication*, *6*(4), 103-108.

Mourtada, J., & Gaïffas, S. (2019). On the optimality of the Hedge algorithm in the stochastic regime. *Journal of Machine Learning Research*, *20*(83), 1-28.

Mulvey, J. M., & Vladimirou, H. (1991). Applying the progressive hedging algorithm to stochastic generalized networks. *Annals of Operations Research*, *31*(1), 399-424.

Noaeen, M., Naik, A., Goodman, L., Crebo, J., Abrar, T., Far, B., Abad, Z., & Bazzan, A. (2022). Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert Systems with Applications*, *199*, 116830. https://doi.org/10.1016/j.eswa.2022.116830

Onof, C., Chandler, R., Kakou, A., Northrop, P., Wheater, H., & Isham, V. (2000). Rainfall modelling using Poisson-cluster processes: A review of developments. *Stochastic Environmental Research and Risk Assessment*, *14*(6), 384-411. https://doi.org/10.1007/S004770000043

Pacheco, F., Exposito, E., Gineste, M., Baudoin, C., & Aguilar, J. (2019). Towards the deployment of machine learning solutions in network traffic classification: A systematic survey. *IEEE Communications Surveys & Tutorials*, *21*(3), 1988-2014. https://doi.org/10.1109/COMST.2018.2883147

Qi, Y., Zhang, S., Qin, L., Huang, Q., Yao, H., Lim, J., & Yang, M. H. (2018). Hedging deep features for visual tracking. *IEEE transactions on pattern analysis and machine intelligence*, *41*(5), 1116-1130.

Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., & Yang, M. H. (2016). Hedged deep tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4303-4311).

Salomon, I. (1986). Telecommunications and travel relationships: A review. *Transportation Research Part A: General*, *20*(3), 223-238. https://doi.org/10.1016/0191-2607(86)90096-8

Saputri, T., & Lee, S. (2020). The application of machine learning in self-adaptive systems: A systematic literature review. *IEEE Access*, *8*, 205948-205967. https://doi.org/10.1109/ACCESS.2020.3036037

Schwefel, H., Antonios, I., & Lipsky, L. (2017). Understanding the relationship between network traffic correlation and queueing behavior: A review based on the N-Burst ON/OFF model. *Performance Evaluation*, *115*, 68-91. https://doi.org/10.1016/j.peva.2017.07.002

Shi, J., Leau, Y., Li, K., Park, Y., & Yan, Z. (2020). Optimization and decomposition methods in network traffic prediction model: A review and discussion. *IEEE Access*, *8*, 202858-202871. https://doi.org/10.1109/ACCESS.2020.3036421

Srinidhi, N., Kumar, S., & Venugopal, K. (2019). Network optimizations in the Internet of Things: A review. *Engineering Science and Technology, an International Journal*. https://doi.org/10.1016/J.JESTCH.2018.09.003

Sunassee, S., Mungur, A., Armoogum, S., & Pudaruth, S. (2021). A comprehensive review on congestion control techniques in networking. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 305-312). https://doi.org/10.1109/ICCMC51019.2021.9418329

Tyagi, V., Darbha, S., & Rajagopal, K. (2009). A review of the mathematical models for traffic flow. *International Journal of Advances in Engineering Sciences and Applied Mathematics*, *1*(1), 53-68. https://doi.org/10.1007/S12572-009-0005-8

Wang, W., Zhang, X., Shi, W., Lian, S., & Feng, D. (2012). Understanding and analyzing network traffic. *IEEE Network*, *26*(5), 4-5. https://doi.org/10.1109/MNET.2012.6135849

Zhang, L., Wu, Y., Wang, J., Liu, S., Chen, C., & Wang, X. (2023, December). Efficient Algorithm for Maximizing Betweenness Centrality in Large Networks. In *2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)* (pp. 647-654). IEEE.

Παράρτημα

Κώδικας #1

```
# -*- coding: utf-8 -*-

import numpy as np
import random
import matplotlib.pyplot as plt

#MULTI_ARM BANDIT
def loss(i, l):
    #it depends on the problem, here return random number
    return l[i]

def hedge(N, T, e, l):
    PRINT = False
    #initialize variables (vectors)
    W = np.ones(N)/N #normalized, to sum 1
    x = np.ones(N)/N

    actions = np.arange(N) #numbers 0,1,2...n-1
    if PRINT:
        print('initial x:',x)
        print('initial W:',W)

    x_hist = np.zeros((T,N))
    #run T times
    for t in range(T):
        #get action i [0 to n-1], according to probability of x
        i = random.choices(actions, x/sum(x))[0]
        if PRINT:
            print('Selection to bet on round',t,':',i)
            print('Loss in round',t,':',x[i])

        #get loss of action i
        g = loss(i, l)
        #update W
        W[i] = W[i] * np.exp(-e*g)
        #update x
        x[i] = W[i] / np.sum(W)
        x_hist[t,:] = x

        R = 0
        for i in range(N):
            R = R + (x[i]*l[i])
        if PRINT:
            print('Modified x after round',t,':',x/sum(x))
            print('Average loss after round',t,':',R)

    plt.figure()
    plt.plot(x_hist)
    plt.xlabel('round')
    plt.ylabel('probability')
    plt.legend(['p_1', 'p_2', 'p_3', 'p_4'])
    plt.show()


    #return final values after T iterations
    return (x,W)
```

```
#mutli-arm bandit
N = 4 #options
M = 1 #money
T = 100 #rounds
e = 0.1


random.seed(100)

print('******************************')
#random loss vector
l = []
for i in range(N):
    l.append(random.random())

#normalize loss
s = sum(l)
for i in range(N):
    l[i] = l[i] / s

print('initial losses:',l,'sum:',sum(l))

#run hedge
(x,W) = hedge(N,T,e,l)
print('x:',x,'---sum x:',sum(x))
print('W:',W,'---sum W:',sum(W))

#reward
R = 0
for i in range(N):
    R = R + M*x[i]*l[i]

print('Reward',R)
```

Κώδικας #2

```
# -*- coding: utf-8 -*-

import numpy as np
import random
import matplotlib.pyplot as plt

#MULTI_ARM BANDIT
# 1 player, constant loss


def loss(i, l):
    #it depends on the problem, here return random number
    return l[i]

def hedge(N, T, e, l):
    PRINT = False
    #initialize variables (vectors)
    W = np.ones(N)/N #normalized, to sum 1
    x = np.ones(N)/N

    actions = np.arange(N) #numbers 0,1,2...n-1
    if PRINT:
        print('initial x:',x)
        print('initial W:',W)

    x_hist = np.zeros((T,N))
    #run T times
    for t in range(T):
        #get action i [0 to n-1], according to probability of x
        i = random.choices(actions, x/sum(x))[0]
        if PRINT:
            print('Selection to bet on round',t,':',i)
            print('Loss in round',t,':',x[i])

        #get loss of action i
        g = loss(i, l)
        #update W
        W[i] = W[i] * np.exp(-e*g)
        #update x
        x[i] = W[i] / np.sum(W)
        x_hist[t,:] = x

        R = 0
        for i in range(N):
            R = R + (x[i]*l[i])
        if PRINT:
            print('Modified x after round',t,':',x/sum(x))
            print('Average loss after round',t,':',R)

    plt.figure()
    plt.plot(x_hist)
    plt.xlabel('round')
    plt.ylabel('probability')
    plt.legend(['p_1', 'p_2'])
    plt.show()


    #return final values after T iterations
    return (x,W)
```

```
#mutli-arm bandit
N = 2 #options
M = 1 #money
T = 100 #rounds
e = 0.1


random.seed(100)

print('******************************')
#random loss matrix
l = np.random.uniform(size=(N))
l = l / np.sum(l)

#run hedge for each player
print("Player 1")
print('initial losses:')
print(l)
print('sum:',sum(l))

(x,W) = hedge(N,T,e,l)
print('x:',x,'---sum x:',sum(x))
print('W:',W,'---sum W:',sum(W))

#reward
R = 0
for i in range(N):
    R = R + M*x[i]*l[i]

print('Reward',R)

print("Player 2")
l2 = 1 - l
print('initial losses:')
print(l2)
print('sum:',sum(l))


(x,W) = hedge(N,T,e,l2)

print('x:',x,'---sum x:',sum(x))
print('W:',W,'---sum W:',sum(W))

#reward
R = 0
for i in range(N):
    R = R + M*x[i]*l[i]

print('Reward',R)
```

Κώδικας #3

```
# -*- coding: utf-8 -*-

import numpy as np
import random
import matplotlib.pyplot as plt

#MULTI_ARM BANDIT
# 3 players, constant loss


def loss(i, l):
    #constant loss
    return l[i]

def hedge(N, T, e, l):
    PRINT = False
    #initialize variables (vectors)
    W = np.ones(N)/N #normalized, to sum 1
    x = np.ones(N)/N

    actions = np.arange(N) #numbers 0,1,2...n-1
    if PRINT:
        print('initial x:',x)
        print('initial W:',W)

    x_hist = np.zeros((T,N))
    #run T times
    for t in range(T):
        #get action i [0 to n-1], according to probability of x
        i = random.choices(actions, x/sum(x))[0]
        if PRINT:
            print('Selection to bet on round',t,':',i)
            print('Loss in round',t,':',x[i])

        #get loss of action i
        g = loss(i, l)
        #update W
        W[i] = W[i] * np.exp(-e*g)
        #update x
        x[i] = W[i] / np.sum(W)
        #keep historical values
        x_hist[t,:] = x

        #reward
        R = 0
        for i in range(N):
            R = R + (x[i]*l[i])
        if PRINT:
            print('Modified x after round',t,':',x/sum(x))
            print('Average loss after round',t,':',R)

    #plot evolution of probabilities x
    plt.figure()
    plt.plot(x_hist)
    plt.xlabel('round')
    plt.ylabel('probability')
    plt.legend(['p_1', 'p_2'])
    plt.show()
```

```python
    #return final values after T iterations
    return (x,W)


#mutli-arm bandit
N = 2 #options
M = 1 #money
T = 100 #rounds
e = 0.1


random.seed(100)

print('********************************')
#random loss matrix
l1 = np.random.uniform(size=(N))
l2 = np.random.uniform(size=(N))
l3 = np.random.uniform(size=(N))
s = (l1+l2+l3)
l1 = l1 / s
l2 = l2 / s
l3 = l3 / s

#run hedge for each player
print("Player 1")
print('initial losses:')
print(l1)
print('sum:',sum(l1))

(x,W) = hedge(N,T,e,l1)
print('x:',x,'---sum x:',sum(x))
print('W:',W,'---sum W:',sum(W))

#reward
R = 0
for i in range(N):
    R = R + M*x[i]*l1[i]

print('Reward',R)


print("Player 2")
print('initial losses:')
print(l2)
print('sum:',sum(l2))


(x,W) = hedge(N,T,e,l2)

print('x:',x,'---sum x:',sum(x))
print('W:',W,'---sum W:',sum(W))

#reward
R = 0
for i in range(N):
    R = R + M*x[i]*l2[i]

print('Reward',R)
```

```python
print("Player 3")
print('initial losses:')
print(l3)
print('sum:',sum(l3))


(x,W) = hedge(N,T,e,l3)

print('x:',x,'---sum x:',sum(x))
print('W:',W,'---sum W:',sum(W))

#reward
R = 0
for i in range(N):
    R = R + M*x[i]*l3[i]

print('Reward',R)
```

Κώδικας #4

```
# -*- coding: utf-8 -*-

import numpy as np
import random
import matplotlib.pyplot as plt

#MULTI_ARM BANDIT
# 2 players, actions are linked


def loss(i1, i2, l):
    #constant loss
    return l[i1][i2]

def hedge(N, T, e, l):
    PRINT = False
    #initialize variables (vectors)
    W1 = np.ones(N)/N #normalized, to sum 1
    x1 = np.ones(N)/N
    W2 = np.ones(N)/N #normalized, to sum 1
    x2 = np.ones(N)/N

    actions = np.arange(N) #numbers 0,1,2...n-1
    if PRINT:
        print('initial x:',x1)
        print('initial W:',W1)
        print('initial x:',x2)
        print('initial W:',W2)

    x1_hist = np.zeros((T,N))
    x2_hist = np.zeros((T,N))
    #run T times
    for t in range(T):
        #get action i [0 to n-1], according to probability of x
        i1 = random.choices(actions, x1/sum(x1))[0]
        i2 = random.choices(actions, x2/sum(x2))[0]
        if PRINT:
            print('Selection to bet on round',t,':',i1)
            print('Loss in round',t,':',x1[i1])
            print('Selection to bet on round',t,':',i2)
            print('Loss in round',t,':',x2[i2])

        #get loss of action i
        g = loss(i1, i2, l)
        #update W
        W1[i1] = W1[i1] * np.exp(-e*g)
        W2[i2] = W2[i2] * np.exp(-e*g)
        #update x
        x1[i1] = W1[i2] / np.sum(W1)
        x2[i1] = W2[i2] / np.sum(W2)
        #keep historical values
        x1_hist[t,:] = x1
        x2_hist[t,:] = x2

        #reward
        R1 = 0
        R2 = 0
        for i in range(N):
            R1 = R1 + (x1[i]*l[i1][i2])
```

```python
                R2 = R2 + (x2[i]*l[i1][i2])
            if PRINT:
                print('Modified x after round',t,':',x1/sum(x1))
                print('Average loss after round',t,':',R1)
                print('Modified x after round',t,':',x2/sum(x2))
                print('Average loss after round',t,':',R2)

    #plot evolution of probabilities x
    plt.figure()
    plt.plot(x1_hist)
    plt.xlabel('round')
    plt.ylabel('probability')
    plt.title('Player 1')
    plt.legend(['p_1', 'p_2'])
    plt.show()

    plt.figure()
    plt.plot(x2_hist)
    plt.xlabel('round')
    plt.ylabel('probability')
    plt.title('Player 2')
    plt.legend(['p_1', 'p_2'])
    plt.show()

    #return final values after T iterations
    return (x1,W1,x2,W2, R1, R2)


#mutli-arm bandit
N = 2 #options
M = 1 #money
T = 1000 #rounds
e = 0.5


random.seed(100)

print('******************************')
#random loss matrix
l = np.array([[0.5, 1],[1,0.5]])

#run hedge for both players simultaneously
print('initial losses:')
print(l)

(x1,W1,x2,W2, R1, R2) = hedge(N,T,e,l)
print("Player 1")
print('x:',x1,'---sum x:',sum(x1))
print('W:',W1,'---sum W:',sum(W1))
print('Reward',R1)

print("Player 2")
print('x:',x2,'---sum x:',sum(x2))
print('W:',W2,'---sum W:',sum(W2))
print('Reward',R2)
```

Κώδικας #5

```
# -*- coding: utf-8 -*-

import numpy as np
import random
import matplotlib.pyplot as plt

#MULTI_ARM BANDIT
# P players, N actions, which are linked


def loss(actions, N):
    P = len(actions)
    l = np.zeros(N)
    for a in actions:
        l[a] = l[a] + 1

    l = l / P
    return l

def hedge(N, T, e):
    PRINT = False
    #initialize variables (vectors)
    W1 = np.ones(N)/N #normalized, to sum 1
    x1 = np.ones(N)/N
    W2 = np.ones(N)/N #normalized, to sum 1
    x2 = np.ones(N)/N
    W3 = np.ones(N)/N #normalized, to sum 1
    x3 = np.ones(N)/N
    W4 = np.ones(N)/N #normalized, to sum 1
    x4 = np.ones(N)/N

    actions = np.arange(N) #numbers 0,1,2...n-1
    if PRINT:
        print('initial x:',x1)
        print('initial W:',W1)
        print('initial x:',x2)
        print('initial W:',W2)
        print('initial x:',x3)
        print('initial W:',W3)
        print('initial x:',x4)
        print('initial W:',W4)

    x1_hist = np.zeros((T,N))
    x2_hist = np.zeros((T,N))
    x3_hist = np.zeros((T,N))
    x4_hist = np.zeros((T,N))
    #run T times
    for t in range(T):
        #get action i [0 to n-1], according to probability of x
        i1 = random.choices(actions, x1/sum(x1))[0]
        i2 = random.choices(actions, x2/sum(x2))[0]
        i3 = random.choices(actions, x2/sum(x2))[0]
        i4 = random.choices(actions, x2/sum(x2))[0]

        if PRINT:
            print('Selection to bet on round',t,':',i1)
            print('Loss in round',t,':',x1[i1])
            print('Selection to bet on round',t,':',i2)
            print('Loss in round',t,':',x2[i2])
```

```python
        print('Selection to bet on round',t,':',i3)
        print('Loss in round',t,':',x3[i3])
        print('Selection to bet on round',t,':',i4)
        print('Loss in round',t,':',x4[i4])

    #get loss of action i
    g = loss([i1, i2, i3, i4], N)
    #update W
    W1[i1] = W1[i1] * np.exp(-e*g[i1])
    W2[i2] = W2[i2] * np.exp(-e*g[i2])
    W3[i3] = W3[i3] * np.exp(-e*g[i3])
    W4[i4] = W4[i4] * np.exp(-e*g[i4])
    #update x
    x1[i1] = W1[i1] / np.sum(W1)
    x2[i2] = W2[i2] / np.sum(W2)
    x3[i3] = W3[i3] / np.sum(W3)
    x4[i4] = W4[i4] / np.sum(W4)
    #keep historical values
    x1_hist[t,:] = x1
    x2_hist[t,:] = x2
    x3_hist[t,:] = x3
    x4_hist[t,:] = x4
    #reward
    R1 = 0
    R2 = 0
    R3 = 0
    R4 = 0
    for i in range(N):
        R1 = R1 + (x1[i]*g[i1])
        R2 = R2 + (x2[i]*g[i2])
        R3 = R3 + (x3[i]*g[i3])
        R4 = R4 + (x4[i]*g[i4])
    if PRINT:
        print('Modified x after round',t,':',x1/sum(x1))
        print('Average loss after round',t,':',R1)
        print('Modified x after round',t,':',x2/sum(x2))
        print('Average loss after round',t,':',R2)
        print('Modified x after round',t,':',x3/sum(x3))
        print('Average loss after round',t,':',R3)
        print('Modified x after round',t,':',x4/sum(x4))
        print('Average loss after round',t,':',R4)

#plot evolution of probabilities x
legend = []
for i in range(N):
    legend.append('p_' + str(i+1))
plt.figure()
plt.plot(x1_hist)
plt.xlabel('round')
plt.ylabel('probability')
plt.title('Player 1')
plt.legend(legend)
plt.show()

plt.figure()
plt.plot(x2_hist)
plt.xlabel('round')
plt.ylabel('probability')
plt.title('Player 2')
plt.legend(legend)
plt.show()
```

```
    plt.figure()
    plt.plot(x3_hist)
    plt.xlabel('round')
    plt.ylabel('probability')
    plt.title('Player 3')
    plt.legend(legend)
    plt.show()

    plt.figure()
    plt.plot(x4_hist)
    plt.xlabel('round')
    plt.ylabel('probability')
    plt.title('Player 4')
    plt.legend(legend)
    plt.show()

    #return final values after T iterations
    return (x1,W1,x2,W2,x3,W3,x4,W4, R1, R2, R3, R4)



#mutli-arm bandit
N = 3 #options - routes
M = 1 #money
T = 1000 #rounds
e = 0.5


random.seed(100)

print('*******************************')


(x1,W1,x2,W2,x3,W3,x4,W4, R1, R2, R3, R4) = hedge(N,T,e)
print("Player 1")
print('x:',x1,'---sum x:',sum(x1))
print('W:',W1,'---sum W:',sum(W1))
print('Reward',R1)

print("Player 2")
print('x:',x2,'---sum x:',sum(x2))
print('W:',W2,'---sum W:',sum(W2))
print('Reward',R2)

print("Player 3")
print('x:',x3,'---sum x:',sum(x3))
print('W:',W3,'---sum W:',sum(W3))
print('Reward',R3)

print("Player 4")
print('x:',x4,'---sum x:',sum(x4))
print('W:',W4,'---sum W:',sum(W4))
print('Reward',R4)
```

Κώδικας #5 με επανάληψη των πειραμάτων και διαστήματα εμπιστοσύνης

```
# -*- coding: utf-8 -*-

import numpy as np
import random
import matplotlib.pyplot as plt

#MULTI_ARM BANDIT
# P players, N actions, which are linked


def loss(actions, N):
    K = actions.shape[0]
    l = np.zeros(N)
    for a in actions:
        l[a] = l[a] + 1

    l = l / K
    return l

def hedge(N, K, T, e):
    PRINT = False
    PLOT = False

    #initialize variables (vectors)
    W = np.ones((K,N))/N #normalized, to sum 1
    x = np.ones((K,N))/N

    actions = np.arange(N) #numbers 0,1,2...n-1
    if PRINT:
        print('initial x:',x)
        print('initial W:',W)

    x_hist = np.zeros((T,K,N))
    #run T times
    for t in range(T):
        i = np.zeros(K).astype(int)
        for k in range(K):
            #get action i [0 to n-1], according to probability of x
            i[k] = random.choices(actions, x[k,:]/np.sum(x[k,:]))[0]

        if PRINT:
            print('Selection to bet on round',t,':',i[k])
            print('Loss in round',t,':',x[k,i[k]])

        #get loss of action i
        g = loss(i, N)

        for k in range(K):
            #update W
            W[k,i[k]] = W[k,i[k]] * np.exp(-e*g[i[k]])

            #update x
            x[k,i[k]] = W[k,i[k]] / np.sum(W[k,:])
            #normalize x
            x[k,:] = x[k,:] /  np.sum(x[k,:])

        #keep historical values
        x_hist[t,:,:] = x
```

```
        #reward
        R = np.zeros((K))
        for a in range(N):
            for k in range(K):
                R[k] = R[k] + (x[k,a]*g[i[k]])

        if PRINT:
            for k in range(K):
                print('Modified x after
round',t,':',x[k,:]/sum(x[k]))
                print('Average loss after round',t,':',R[k])

    if PLOT:
        #plot evolution of probabilities x
        legend = []
        for a in range(N):
            legend.append('p_' + str(a+1))

        for k in range(K):
            plt.figure()
            plt.plot(x_hist[k,:])
            plt.xlabel('round')
            plt.ylabel('probability')
            plt.title('Player ' + str(k+1))
            plt.legend(legend)
            plt.show()


    #return final values after T iterations
    return (x,W, R)



#mutli-arm bandit
N = 5 #options - routes
M = 1 #money
T = 1000 #rounds
e = 0.5
K = 4 #players
ITER = 1000 #iterations


random.seed(100)

print('*******************************')

p = np.zeros((K,N,ITER))
for i in range(ITER):
    (x,W,R) = hedge(N,K,T,e,)
    p[:,:,i] = x

for k in range(K):
    print("Player 1")
    print('x:',x[k,:],'---sum x:',sum(x[k,:]))
    print('W:',W[k,:],'---sum W:',sum(W[k,:]))
    print('Reward',R[k])


#distribution of p
plt.hist(p[0,0,:], ec='black', bins=10)
```

```
pm = np.mean(p,2)
ps = np.std(p,2)
print(pm)

for k in range(K):
    for n in range(N):
        print("%.3f (%.3f - %.3f)\t" % (pm[k,n],
np.percentile(p[k,n,:], 5), np.percentile(p[k,n,:], 95)), end=" ")
    print("")
```