ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΚΑΙ ΜΑΘΗΣΗΣ

Τεχνητή Νοημοσύνη για Παιχνίδια: Ανάπτυξη Πρακτόρων με χρήση Βαθιάς Μάθησης

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

του

Αναστάσιου Παπαγιάννη

Αθήνα, Απρίλιος 2024

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Τεχνητή Νοημοσύνη για Παιχνίδια: Ανάπτυξη Πρακτόρων με χρήση Βαθιάς Μάθησης

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

του

Αναστάσιου Παπαγιάννη

Συμβουλευτική Επιτροπή: Ανδρέας - Γεώργιος Σταφυλοπάτης
Γεώργιος Στάμου
Παναγιώτης Τσανάκας

Εγκρίθηκε από την επταμελή επιτροπή την 26η Απριλίου 2024

.....	.....	.....
Ανδρέας - Γεώργιος Σταφυλοπάτης Καθηγητής Ε.Μ.Π.	Γεώργιος Στάμου Καθηγητής Ε.Μ.Π.	Παναγιώτης Τσανάκας Καθηγητής Ε.Μ.Π.
.....	.....	.....
Αθανάσιος Βουλόδημος Επ. Καθηγητής Ε.Μ.Π.	Αθανάσιος Ροντογιάννης Αν. Καθηγητής Ε.Μ.Π.	Γεώργιος Καρυδάκης Αν. Καθηγητής Παν.Αιγαίου

.....
Γεώργιος Αλεξανδρίδης
Επ. Καθηγητής Ε.Κ.Π.Α.

Αθήνα, Απρίλιος 2024

.....

Αναστάσιος Δ. Παπαγιάννης
Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η τεχνητή νοημοσύνη εξελίσσεται την τελευταία δεκαετία με ραγδαίο ρυθμό, διεισδύοντας σε ολοένα και περισσότερους επιστημονικούς κλάδους όπως η πληροφορική, η ιατρική, ακόμα και η εκπαίδευση. Η συνεχώς αυξανόμενη εφαρμογή της σε διαφορετικούς τομείς, αναμενόμενα συμβάλλει αφενός στη διαρκή ανάπτυξη σύγχρονων τεχνικών και αλγορίθμων, αφετέρου στον προσδιορισμό επιπλέον απαιτήσεων και στη δημιουργία νέων προκλήσεων για τον ευρύτερο κλάδο της τεχνητής νοημοσύνης. Ένα πεδίο το οποίο παρουσιάζει ιδιαίτερο ενδιαφέρον αφορά την εφαρμογή τέτοιων αλγορίθμων στο πλαίσιο των παιχνιδιών, τα περιβάλλοντα των οποίων προσφέρουν άμεση ανατροφοδότηση και συγχρόνως παρουσιάζουν διαφορετικές δυσκολίες και περιορισμούς. Στόχος της παρούσας διατριβής είναι η ανάπτυξη ευφυών πρακτόρων για ηλεκτρονικά παιχνίδια με τη χρήση τεχνητής νοημοσύνης και τεχνικών μηχανικής μάθησης. Υπό αυτό το πρίσμα, εξετάζονται οι επικρατέστερες επί του παρόντος τεχνικές, προτείνονται αλγόριθμοι και παρουσιάζονται μέθοδοι προκειμένου να αντιμετωπιστούν ορισμένες από τις κυριότερες προκλήσεις που εμφανίζονται.

Στο πρώτο στάδιο της διατριβής εξετάστηκε η υλοποίηση πρακτόρων με βάση τους γενετικούς αλγορίθμους. Συγκεκριμένα, διερευνήθηκε η δυνατότητα να εφαρμοστούν ως αυτούσια τεχνική για την κωδικοποίηση των καταστάσεων του περιβάλλοντος και την τελική λήψη αποφάσεων από τον πράκτορα. Σε αυτό το πλαίσιο σχεδιάστηκε και μία νέα μέθοδος αναπαράστασης των καταστάσεων προκειμένου να μειωθεί ο χώρος καταστάσεων και να είναι εφικτή η υλοποίηση της προτεινόμενης προσέγγισης. Η λογική της αναπαράστασης βασίστηκε σε ένα μοτίβο N-πλειάδων από συντεταγμένες του χώρου προκειμένου να κωδικοποιηθούν οι καταστάσεις χρησιμοποιώντας τη λιγότερη δυνατή πληροφορία. Τα πειράματα που διενεργήθηκαν ανέδειξαν τη λειτουργικότητα της συγκεκριμένης τεχνικής κατατάσσοντάς την υψηλότερα από αντίστοιχες μεθόδους διαφορετικής προσέγγισης των εξελικτικών αλγορίθμων.

Στη συνέχεια μελετήθηκε η συμπεριφορά ενός ευφυούς πράκτορα σε στοχαστικά περιβάλλοντα. Σε αυτήν την περίπτωση, ερευνήθηκε κατά κύριο λόγο ο αλγόριθμος δενδρικής αναζήτησης Μόντε Κάρλο που αποτελεί την προσέγγιση αιχμής για ένα μεγάλο υποσύνολο του ευρύτερου πεδίου της τεχνητής νοημοσύνης για παιχνίδια. Το πρώτο τμήμα που εξετάστηκε ήταν η βελτιστοποίηση του σταδίου κατά το οποίο γίνεται η αξιολόγηση των καταστάσεων που χρησιμοποιούνται στη συνέχεια από τον αλγόριθμο. Για το σκοπό αυτό, ένας ταξινομητής ακραίας ενίσχυσης κλίσης εκπαιδευμένος σε ειδικά σχεδιασμένο σύνολο δεδομένων, ενσωματώθηκε στον αλγόριθμο αυξάνοντας σε σημαντικό βαθμό την ακρίβεια αποτίμησης της αξίας καταστάσεων. Επιπλέον υλοποιήθηκε μία διαδικασία στοχαστικής αξιολόγησης των κόμβων του δέντρου αναζήτησης με στόχο την προσαρμογή του μοντέλου ανάλογα με το συντελεστή διακλάδωσης. Για την αποδοτικότερη εκμετάλλευση της πληροφορίας με βάση το βάθος των δέντρων, εφαρμόστηκε και μία τεχνική πρώιμης προσομοίωσης στα αρχικό στάδιο του αλγορίθμου που οδήγησε σε ισχυρότερες προβλέψεις του ταξινομητή. Ο συνδυασμός των παραπάνω μεθόδων οδήγησε σε μεγάλη αύξηση της απόδοσης του πράκτορα, που ξεπέρασε τον βέλτιστο αλγόριθμο που

παρέχεται από το περιβάλλον στο οποίο δοκιμάστηκε.

Έπειτα, αξιολογήθηκε η βελτιστοποίηση του σταδίου επιλογής του αλγορίθμου. Σε αυτό το πλαίσιο, η παραπάνω μεθοδολογία ενισχύθηκε επιπλέον με μία πρωτότυπη τεχνική κλαδέματος βασισμένη σε χρήση τεχνητών νευρωνικών δικτύων, προκειμένου να μειωθεί ο χώρος αναζήτησης. Στόχος είναι κατά τη διάρκεια του αλγορίθμου να αφαιρούνται από το σύνολο των ενεργειών προς εξέταση οι ενέργειες που δεν αναμένεται να έχουν υψηλή αξία και οι εναπομείναντες υπολογιστικοί πόροι να αξιοποιούνται για την ακριβέστερη αξιολόγηση των υπόλοιπων ενεργειών. Για την υλοποίηση αυτής της μεθόδου εκπαιδεύτηκαν δύο διαφορετικά νευρωνικά δίκτυα τα οποία χρησιμοποιήθηκαν συνεργατικά. Με το συνδυασμό των εξόδων των δύο δικτύων προσδιορίζεται το βέλτιστο ζεύγος επαναλήψεων και πλήθους ενεργειών προς αφαίρεση και μειώνεται επαναληπτικά ο χώρος αναζήτησης μέχρι την ολοκλήρωση του αλγορίθμου. Η εκπαίδευση των δικτύων έγινε σε συνθετικά δεδομένα εκπαίδευσης τα οποία προέκυψαν από ειδικό περιβάλλον προσομοίωσης που υλοποιήθηκε για αυτό το σκοπό. Η τεχνική κλαδέματος χρησιμοποιήθηκε τόσο αυτούσια όσο και σε συνδυασμό με τον ταξινομητή ενίσχυσης κλίσης οδηγώντας σε περαιτέρω βελτίωση της απόδοσης του αλγορίθμου.

Στο επόμενο μέρος, εξετάστηκε η ενίσχυση της φάσης επιλογής στη δενδρική αναζήτηση Μόντε Κάρλο χωρίς την εισαγωγή γνώσης πεδίου. Η βασική ιδέα σε αυτή την περίπτωση αφορά στην αντιστοίχιση παρόμοιων κόμβων προκειμένου να γίνεται – κατά τη διαδικασία της επιλογής – χρήση στατιστικών των κόμβων που βρίσκονται σε υψηλότερο επίπεδο του δέντρου αναζήτησης και συνεπώς έχουν ακριβέστερα δεδομένα (καθώς έχουν επισκεφθεί περισσότερες φορές κατά τη διάρκεια της αναζήτησης). Υπό αυτό το πρίσμα υλοποιήθηκαν δύο διαφορετικές μεθοδολογίες αντιστοίχισης καταστάσεων με βάση την ακολουθία των ενεργειών που προηγήθηκαν. Στην πρώτη περίπτωση ο προσδιορισμός της ομοιότητας των κόμβων έγινε με κριτήριο το μήκος των πανομοιότυπων N-γραμμών από τα οποία προέκυψαν ενώ στη δεύτερη έγινε με βάση μία ειδικά σχεδιασμένη αναπαράσταση της ομοιότητας των ενεργειών που εκτελέστηκαν. Οι προτεινόμενες τεχνικές εφαρμόστηκαν σε περιβάλλοντα γενικών ευφυών πρακτόρων, πετυχαίνοντας υψηλότερη απόδοση από τις αντίστοιχες προσεγγίσεις που αφορούν το στάδιο επιλογής του αλγορίθμου στην πλειοψηφία των περιπτώσεων.

Στο τελευταίο σκέλος της διατριβής, διερευνήθηκε το πεδίο της ενισχυτικής μάθησης. Ιδιαίτερα, εξετάστηκε η εισαγωγή μίας τεχνικής επαύξησης δεδομένων με γεννητικά μοντέλα για τη δημιουργία νέων, συνθετικών καταστάσεων με στόχο την αποτελεσματικότερη εκπαίδευση του πράκτορα. Συμπληρωματικά, σχεδιάστηκε ένα μοντέλο για την πρόβλεψη της ενέργειας που εκτελείται μεταξύ δύο διαδοχικών καταστάσεων προκειμένου να είναι εφικτή η σύνθεση ολοκληρωμένων δειγμάτων στη μορφή που απαιτείται για την επίλυση προβλημάτων ενισχυτικής μάθησης. Τα επιμέρους μοντέλα χρησιμοποιήθηκαν συνδυαστικά για τη δημιουργία συνθετικών δεδομένων με υψηλή και χαμηλή άμεση ανταμοιβή, τα οποία συναντώνται λιγότερο συχνά κατά τη διάρκεια της αλληλεπίδρασης του πράκτορα με το περιβάλλον. Η προτεινόμενη μεθοδολογία, στην οποία τα συνθετικά δείγματα αναμειγνύονται με τα πραγματικά δεδομένα κατά τη διαδικασία της εκπαίδευσης, αξιολογήθηκε σε διαφορετικά, ετερογενή περιβάλλοντα επιτυγχάνοντας αύξηση της συνολικής ανταμοιβής του πράκτορα συγκριτικά με αντίστοιχες κλασσικές τεχνικές επαύξησης εικόνας.

Λέξεις Κλειδιά

Τεχνητή Νοημοσύνη, Βαθιά Μηχανική Μάθηση, Ευφυείς Πράκτορες, Νευρωνικά Δίκτυα, Γενετικοί Αλγόριθμοι, Δενδρική Αναζήτηση Μόντε Κάρλο, Τεχνικές Κλαδέματος, Ενισχυτική Μάθηση, Γεννητική Επαύξηση Δεδομένων, Ηλεκτρονικά Παιχνίδια

Abstract

Artificial intelligence has been evolving at a rapid pace in the last decade, being part of many scientific fields such as IT, medicine and even education. Its ever-increasing application in different fields contributes on the one hand to the continuous development of modern techniques and algorithms, on the other hand leads to additional requirements and new challenges for the wider field of artificial intelligence. A field of particular interest concerns the application of such algorithms in the context of games, the environments of which offer immediate feedback and at the same time introduce several difficulties and limitations. The aim of this thesis is the development of intelligent agents for video games using artificial intelligence and machine learning techniques. Under this scope, the currently prevailing techniques are examined and new algorithms and methods are proposed in order to deal with the main appearing challenges.

In the first stage of the thesis, the implementation of agents based on genetic algorithms was examined. In particular, the possibility of applying a genetic algorithm as a standalone technique for encoding the states of the environment and the final decision-making by the agent was examined. In this context, a novel method for the states' representation was designed in order to reduce the state space and make feasible the implementation of the proposed approach. The representation logic was based on N-groups of blocks in order to encode states using the least possible information. The experiments carried out highlighted the functionality of this particular technique by ranking it higher than corresponding methods of different evolutionary-based approaches.

Afterwards the behavior of an intelligent agent in stochastic environments was studied. In this case, the Monte Carlo tree search algorithm was mainly studied as it is the state-of-the-art approach to a large subset of the broader field of artificial intelligence for games. The first section that was considered was the optimization of the stage in which the states which are subsequently used by the algorithm are evaluated. To that end, a gradient boosting classifier trained on a specially designed dataset, was incorporated into the algorithm significantly increasing its accuracy in estimating the states' values. In addition, a stochastic evaluation process of the tree nodes was implemented aiming to better adapt the model to the branching factor. In order to exploit information more efficiently based on the trees' depth, an early simulation technique was also applied in the early stages of the algorithm leading to stronger classifier predictions. The combination of the above methods led to a great increase of the agent's performance which surpassed the optimal algorithm provided by the used framework.

Then, the optimization of the selection step of the algorithm was evaluated. In this respect, the above methodology was additionally enhanced with a novel pruning technique based on the use of artificial neural networks, in order to reduce the search space. The goal is to remove the actions with low expected value from the set of possible actions during the execution of the algorithm and utilize the remaining computing resources to more accurately evaluate the remaining actions.

To implement this technique, two separate neural networks were trained and used collaboratively. By combining the outputs of the two networks, the optimal pair of iterations and number of actions to be pruned is determined and the search space is being reduced iteratively until the algorithm is finished. The networks were trained on synthetic training data which were derived from a special simulation environment implemented for this purpose. The pruning technique was tested individually as well as in conjunction with the gradient boosting classifier leading to further improvement of the algorithm's performance.

In the next section, the enhancement of the selection phase of the Monte Carlo tree search algorithm without domain knowledge was examined. In this case, the core idea is to match similar nodes in order to use – during the selection phase – statistics of nodes located at a higher level of the search tree which therefore have more accurate data (as they have been visited more times during the search). In this light, two different state matching methodologies were implemented based on the sequence of actions that preceded. In the first case, the nodes' similarity was based on the length of identical N-grams from which they were derived, while in the second case it was based on a specifically designed representation of the similarity of executed actions. The proposed techniques were applied to general intelligent agents' environments achieving higher performance than relevant approaches concerning the selection phase of the algorithm in the majority of cases.

In the last part of the thesis, the field of reinforcement learning was investigated. In particular, the introduction of a data augmentation technique based on generative models for the creation of new, synthetic states was examined, aiming to make the training of the agent more effective. Additionally, a model was designed to predict the action performed between two successive states in order to be able to produce complete samples in the form needed to solve reinforcement learning problems. These models are combined to create synthetic data for cases of high and low immediate rewards, which are encountered less frequently during the agent's interaction with the environment. The proposed methodology, in which the synthetic samples are mixed with the actually observed data during the training process, was evaluated in different, heterogeneous environments achieving an increase in the agent's total obtained reward compared to traditional image augmentation techniques.

Keywords

Artificial Intelligence, Deep Learning, Intelligent Agents, Neural Networks, Genetic Algorithms, Monte Carlo Tree Search, Pruning Techniques, Reinforcement Learning, Generative Data Augmentation, Video Games

Ευχαριστίες

Για την εκπόνηση της διδακτορικής διατριβής θα ήθελα κατ΄ αρχάς να ευχαριστήσω τον κ. Ανδρέα-Γεώργιο Σταφυλοπάτη, καθηγητή Ε.Μ.Π. και υπεύθυνο του εργαστηρίου Τεχνητής Νοημοσύνης και Συστημάτων Μάθησης της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών. Ο κ. Σταφυλοπάτης με καθοδήγησε από το 2017, οπότε και ξεκίνησα τις διδακτορικές μου σπουδές και συνέβαλε τα μέγιστα στην εξέλιξη και την ολοκλήρωση του ερευνητικού μου έργου. Υπό τη συνεχή επιστημονική του επίβλεψη αλλά και τη στήριξή του σε προσωπικό επίπεδο καθ΄ όλη τη διάρκεια της παρουσίας μου στο εργαστήριο, κατέστη δυνατή η διεκπεραίωση της παρούσας έρευνας.

Ιδιαίτερες ευχαριστίες οφείλω στον κ. Γεώργιο Στάμου και στον κ. Παναγιώτη Τσανάκα που ως μέλη της τριμελούς επιτροπής είχαν καθοριστικό ρόλο σε αυτή την πορεία. Εξίσου θα ήθελα να ευχαριστήσω τον κ. Γεώργιο Αλεξανδρίδη για την ιδιαίτερη συμβολή του και την εξαιρετική συνεργασία που είχαμε όλα αυτά τα χρόνια και τους κυρίους Αθανάσιο Βουλόδημο, Αθανάσιο Ροντογιάννη και Γεώργιο Καρυδάκη για τη συμμετοχή τους στην επταμελή επιτροπή και τις πολύτιμες συμβουλές και παρατηρήσεις τους.

Επίσης θα ήθελα να αναφερθώ ξεχωριστά στα μέλη του εργαστηρίου με τα οποία εκτός από τη συνεργασία που είχαμε σε ακαδημαϊκό επίπεδο, αναπτύξαμε εξαιρετικές σχέσεις φιλίας. Συγκεκριμένα θα ήθελα να ευχαριστήσω τους Γιώργο Ιωάννου, Θάνο Τασάκο, Θάνο Τάγαρη, Άρη Λαναρίδη, Ελένη Βάθη, Μάρα Σδράκα και Πάνο Κουρή για την πολύτιμη βοήθεια και την υποστήριξή τους. Τέλος, ευχαριστώ τους γονείς μου που με τη στήριξη τους σε όλη αυτή την πορεία συνέβαλαν στην ομαλή ενασχόληση μου και τελικά την περάτωση αυτής της διδακτορικής διατριβής.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

1.1 Τεχνητή Νοημοσύνη

Η *Τεχνητή Νοημοσύνη* (ΤΝ) (Artificial Intelligence – AI) είναι ο κλάδος της πληροφορικής που έχει ως αντικείμενο την ανάπτυξη ευφυών συμπεριφορών παρόμοιων με την ανθρώπινη [18]. Με τον όρο "ευφυία" εννοείται η δυνατότητα μάθησης, εκπαίδευσης, εξαγωγής λογικών συμπερασμάτων και γενίκευσης. Υπαχουν τρεις βασικές κατηγορίες τεχνητής νοημοσήνης:

Εξειδικευμένη Τεχνητή Νοημοσύνη (Narrow AI) Πρόκειται για συστήματα τεχνητής νοημοσύνης που έχουν σχεδιαστεί για να επιτελούν συγκεκριμένες εργασίες ή εξειδικεύονται σε έναν τομέα. Τέτοια συστήματα υπερέχουν στην επίλυση ειδικών προβλημάτων, αλλά υστερούν όσον αφορά τη δυνατότητα γενίκευσης. Παραδείγματα εξειδικευμένης τεχνητής νοημοσύνης περιλαμβάνουν συστήματα αναγνώρισης εικόνων, συστήματα συστάσεων, κ.λπ.

Γενικευμένη Τεχνητή Νοημοσύνη (General AI) Η γενικευμένη τεχνητή νοημοσύνη, επίσης γνωστή ως Ισχυρή τεχνητή νοημοσύνη (Strong AI) αναφέρεται σε συστήματα που διαθέτουν νοημοσύνη αντίστοιχη της ανθρώπινης και είναι ικανά να εκτελούν οποιαδήποτε νοητική εργασία μπορεί να κάνει ένας άνθρωπος. Τα συστήματα γενικευμένης τεχνητής νοημοσύνης μπορούν να κατανοήσουν, να μάθουν και να εφαρμόσουν τη γνώση σε διάφορους τομείς.

Εφαρμοσμένη Τεχνητή Νοημοσύνη (Applied AI) Η εφαρμοσμένη τεχνητή νοημοσύνη αφορά τη χρήση τεχνολογιών και μεθόδων τεχνητής νοημοσύνης για την επίλυση συγκεκριμένων προβλημάτων του πραγματικού κόσμου ή την εκτέλεση συγκεκριμένων εργασιών σε διάφορους τομείς. Περιλαμβάνει την υλοποίηση συστημάτων και την ενσωμάτωση δυνατοτήτων τεχνητής νοημοσύνης σε υπάρχουσες εφαρμογές και συστήματα. Σύγχρονα παραδείγματα χρήσης τέτοιων μεθόδων περιλαμβάνουν εφαρμογές τεχνητής νοημοσύνης στα οικονομικά, στην εξυπηρέτηση πελατών και στα αυτόνομα οχήματα.

Με βάση τα χαρακτηριστικά λειτουργίας, η τεχνητή νοημοσύνη μπορεί να αναλυθεί σε δύο επιμέρους κατηγορίες [72]:

Συμβολική Τεχνητή Νοημοσύνη (Symbolic AI) Σε αυτή την κατηγορία ανήκουν οι προσεγγίσεις που βασίζονται στην υπόθεση του συστήματος φυσικών συμβόλων των Newell και Simon και συνιστούν αυτό που καλείται "κλασική" τεχνητή νοημοσύνη. Βασικό τους χαρακτηριστικό είναι η εφαρμογή λογικών τελεστών σε βάσεις καθορισμένης γνώσης. Σε αυτή την περίπτωση, η γνώση σχετικά με έναν τομέα προβλημάτων αντιπροσωπεύεται από δηλωτικές προτάσεις, βασισμένες σε (ή ισοδύναμες με) προτάσεις *λογικής πρώτης τάξης* (first-order logic). Για την εξαγωγή συμπερασμάτων από τη γνώση χρησιμοποιούνται μέθοδοι συλλογιστικής. Αυτές οι προσεγγίσεις χαρακτηρίζονται ως *βασισμένες σε γνώση* (knowledge-based) καθώς η εφαρμογή τους σε πραγματικά προβλήματα προϋποθέτει γνώση του εκάστοτε πεδίου.

Στις περισσότερες προσεγγίσεις συμβολικής τεχνητής νοημοσύνης, υπάρχουν διάφορα επίπεδα ανάλυσης της επιθυμητής συμπεριφοράς. Συνήθως ακολουθείται μία σχεδιαστική μέθοδος *από πάνω προς τα κάτω* (top-down) ξεκινώντας από το επίπεδο γνώσης έως τα επίπεδα υλοποίησης. Στην κορυφή βρίσκεται το επίπεδο γνώσης [70] στο οποίο καθορίζεται η γνώση που χρειάζεται το μηχάνημα. Στη συνέχεια βρίσκεται το επίπεδο συμβόλων, όπου η γνώση αναπαριστάται με συμβολικές δομές όπως λίστες και καθορίζονται οι λειτουργίες που μπορούν να εφαρμοστούν σε αυτές τις δομές. Έπειτα, υπάρχουν χαμηλότερα επίπεδα στα οποία υλοποιούνται οι λειτουργίες επεξεργασίας συμβόλων.

Υποσυμβολική Τεχνητή Νοημοσύνη (Subsymbolic AI) Οι προσεγγίσεις υποσυμβολικής τεχνητής νοημοσύνης ακολουθούν τη μέθοδο *από κάτω προς τα πάνω* (bottom-up), ξεκινώντας από το χαμηλότερο επίπεδο και προχωρώντας σταδιακά προς τα υψηλότερα. Μία εκ των σημαντικότερων υποσυμβολικών προσεγγίσεων είναι η μέθοδος *animat*. Η κύρια ιδέα είναι ότι προκειμένου να υλοποιηθεί μία μηχανή με "υψηλή νοημοσύνη", θα πρέπει να ακολουθηθούν πολλά εξελικτικά βήματα. Συνεπώς θα πρέπει αρχικά να μάθει να προσομοιώνει τη συμπεριφορά και τις ικανότητες απλούστερων οργανισμών (πχ. εντόμων) μέχρι να φτάσει σταδιακά σε υψηλότερα επίπεδα ευφυούς συμπεριφοράς παρόμοιας με την ανθρώπινη.

Το πιο γνωστό παράδειγμα τεχνικών που προέρχονται από την κατηγορία της υποσυμβολικής τεχνητής νοημοσύνης είναι τα νευρωνικά δίκτυα. Αυτά τα μοντέλα είναι εμπνευσμένα από βιολογικά μοντέλα του ανθρώπινου εγκεφάλου και η λειτουργία τους βασίζεται στην προσομοίωση της λειτουργίας των βιολογικών νευρωνικών δικτύων με βάση κάποιο μαθηματικό μοντέλο τους. Επιπλέον μεθοδολογίες έχουν προκύψει επίσης από διαδικασίες που προσομοιώνουν τη βιολογική εξέλιξη συμπεριλαμβανομένης της διασταύρωσης, της μετάλλαξης και της αναπαραγωγής με βάση την ποιότητα των ατόμων. Άλλες προσεγγίσεις από κάτω προς τα πάνω, τύπου animat, βασίζονται στη θεωρία ελέγχου και στην ανάλυση των δυναμικών συστημάτων [6].

1.2 Τεχνητή Νοημοσύνη σε Παιχνίδια

Η εφαρμογή της τεχνητής νοημοσύνης στον τομέα των ηλεκτρονικών παιχνιδιών (*Game AI*), αποτελεί υποκατηγορία του ευρύτερου πεδίου της, παρότι υπήρχε αρχικά η άποψη ότι πρόκειται για δύο ξεχωριστούς κλάδους με ορισμένα κοινά σημεία [101]. Αυτή η άποψη στηρίχθηκε στο γεγονός ότι στις πρώτες απόπειρες εισαγωγής τεχνητής νοημοσύνης σε παιχνίδια, χρησιμοποιούνταν κυρίως προκαθορισμένες συμπεριφορές ανάλογα με τις ενέργειες των παιχτών και κάποιες απλές ευριστικές μέθοδοι, που απείχαν αρκετά από την έννοια της πραγματικής τεχνητής νοημοσύνης. Ωστόσο, τα τελευταία χρόνια έχει υπάρξει σημαντικό ενδιαφέρον από ερευνητές προς αυτή την κατεύθυνση, καθιστώντας πιο ευδιάκριτη την σύνδεση μεταξύ των δύο κλάδων.

Από τις αρχές της δεκαετίας του 1990 άρχισαν να χρησιμοποιούνται κλασικές μέθοδοι της τεχνητής νοημοσύνης όπως *μηχανές πεπερασμένων καταστάσεων* (finite state machines), καθώς η τεχνική που χρησιμοποιούνταν κατά κύριο λόγο μέχρι τότε και βασιζόταν σε προκαθορισμένα *πρότυπα* (patterns) δεν ήταν αποτελεσματική στα πιο πολύπλοκα προβλήματα που παρουσίαζαν τα νεότερα παιχνίδια. Καθώς οι απαιτήσεις για ευφυή συμπεριφορά στα παιχνίδια αυξάνονται συνεχώς, στα σύγχρονα παιχνίδια έχουν εφαρμοστεί πολλές εξελιγμένες τεχνικές τεχνητής νοημοσύνης, όπως δέντρα αναζήτησης, εξελικτικοί αλγόριθμοι, νευρωνικά δίκτυα κ.ά., προσαρμοσμένες κατάλληλα ώστε να ανταποκρίνονται στις απαιτήσεις του κλάδου των παιχνιδιών. Στις μελέτες που έχουν γίνει για την ανάπτυξη ευφυών πρακτόρων στον συγκεκριμένο τομέα, η εφαρμογή αυτών των τεχνικών ξεπερνά το πλαίσιο δοκιμαστικού χαρακτήρα και αποκτά συγκεκριμένους θεωρητικούς και πρακτικούς στόχους. Αντιστρόφως, τεχνικές μηχανικής μάθησης έχουν υλοποιηθεί αρχικά με στόχο την ανάπτυξη ευφυούς συμπεριφοράς σε περιβάλλοντα παιχνιδιών και έχουν επεκταθεί στη συνέχεια σε διαφορετικούς κλάδους αποκτώντας γενικότερη διάσταση.

Ωστόσο, ακόμα υπάρχουν αρκετά περιθώρια βελτίωσης, τα οποία μάλιστα με την πάροδο του χρόνου αυξάνονται μαζί με τις απαιτήσεις των παιχνιδιών για περισσότερες δυνατότητες. Αυτή η εκτίμηση επιβεβαιώνεται και από το γεγονός ότι σε πολλά παιχνίδια, προκειμένου να είναι ανταγωνιστικά για τον παίχτη, χρησιμοποιούνται τεχνικές που "κλέβουν" (*cheating*) [91]. Με αυτό τον όρο περιγράφεται η πρόσβαση σε πληροφορίες που στην πραγματικότητα δεν θα έπρεπε να είναι διαθέσιμες σε έναν πράκτορα, όπως για παράδειγμα η θέση του παίχτη στον κόσμο του παιχνιδιού όταν αυτός είναι κρυμμένος. Άλλες τέτοιες τεχνικές είναι η αύξηση των ικανοτήτων τους (π.χ. ταχύτητα) πέρα από τα φυσιολογικά όρια και η εμφάνισή τους σε ευνοϊκά σημεία της πίστας. Όλα αυτά προκύπτουν από την αδυναμία των γνωστών έως τώρα μεθόδων να δημιουργήσουν ανταγωνιστικούς πράκτορες για τους ανθρώπους, χωρίς τη χρήση επιπλέον πληροφοριών.

Σήμερα, εκτός του ελέγχου των πρακτόρων υπάρχουν πολλές άλλες προκλήσεις για την τεχνητή νοημοσύνη στα ηλεκτρονικά παιχνίδια. Οι βασικότερες είναι η *πλοήγηση* (navigation) και η εύρεση διαδρομής, η προσαρμογή του επιπέδου δυσκολίας στις ικανότητες του χρήστη-παίχτη σε πραγματικό χρόνο και η συλλογή χρήσιμων δεδομένων από τη συμπεριφορά των παιχτών σχετικά με τα σημεία του παιχνιδιού που απολαμβάνουν περισσότερο και τους λόγους για τους οποίους μπορεί να δυσαρεστηθούν ή να βαρεθούν ένα παιχνίδι. Τελευταία, ιδιαίτερο ενδιαφέρον συγκεντρώνει επίσης, η ιδέα ανάπτυξης γεννητριών περιεχομένου υπό-συνθήκη. Αυτό συνεπάγεται τη συλλογή δεδομένων από την πορεία του χρήστη στο παιχνίδι και τη δημιουργία νέων στόχων ανάλογα με τη συμπεριφορά του, δηλαδή την προσαρμογή της εξέλιξης του παιχνιδιού στον κάθε χρήστη.

1.3 Συνεισφορά της Διατριβής

Η έρευνα στην παρούσα διατριβή επικεντρώνεται στη μελέτη και την υλοποίηση ευφυών πρακτόρων με εφαρμογή σε περιβάλλοντα παιχνιδιών. Η συνεισφορά της διατριβής περιλαμβάνει τέσσερις κύριους άξονες: (i) την υλοποίηση πρακτόρων βασισμένων αποκλειστικά στη χρήση γενετικών αλγορίθμων και τον σχεδιασμό κατάλληλης αναπαράστασης των καταστάσεων σε αυτό το πλαίσιο, (ii) την ενίσχυση της φάσης προσομοίωσης του αλγορίθμου δενδρικής αναζήτησης Μόντε Κάρλο με την ενσωμάτωση ταξινομητών ακραίας ενίσχυσης κλίσης και τη στοχαστική αξιολόγηση των κόμβων, (iii) τη βελτιστοποίηση της φάσης επιλογής του ίδιου αλγορίθμου με μία μεθοδολογία κλαδέματος για μείωση του χώρου καταστάσεων καθώς και με τεχνικές που δεν απαιτούν εισαγωγή γνώσης πεδίου και (iv) την εφαρμογή μίας τεχνικής επαύξησης δεδομένων βασισμένης σε γεννητικά μοντέλα, σε αλγορίθμους

ενισχυτικής μάθησης με στόχο την σταθεροποίηση της εκπαίδευσης και την ταχύτερη σύγκλιση των μοντέλων.

Στο πρώτο σκέλος της έρευνας (Κεφάλαιο 3), εξετάζεται μία προσέγγιση χρήσης γενετικών αλγορίθμων ως αυτόνομη τεχνική για την ανάπτυξη ευφυών πρακτόρων. Σε αυτή την περίπτωση υπογραμμίζονται τα πλεονεκτήματα αλλά και οι αδυναμίες της αντιμετώπισης του προβλήματος αμιγώς με γενετικούς αλγορίθμους, συγκριτικά με κλασικές τεχνικές στις οποίες συνήθως λειτουργούν επικουρικά σε συνδυασμό με άλλες μεθόδους. Για να είναι εφικτή η μεταφορά γνώσης σε διαφορετικά περιβάλλοντα, προτείνεται μία μέθοδος επαναχρησιμοποιήσης τμήματος του γενότυπου χωρίς να απαιτείται εκ νέου εκπαίδευση των συγκεκριμένων χρωμοσωμάτων του αλγορίθμου σε κάθε εφαρμογή του. Παράλληλα, παρουσιάζεται μία πρωτότυπη μέθοδος αναπαράστασης των καταστάσεων που βασίζεται σε N-πλειάδες ώστε να αξιοποιείται η διαθέσιμη πληροφορία από τον προτεινόμενο αλγόριθμο με αποδοτικότερο τρόπο. Η πειραματική διαδικασία αναδεικνύει την αποτελεσματικότητα της προτεινόμενης μεθοδολογίας και τη βελτίωση της ταχύτητας σύγκλισης του αλγορίθμου με την εισαγωγή της επαναχρησιμοποίησης τμήματος της αρχικής λύσης.

Στη συνέχεια, η παρούσα εργασία εστιάζει στον αλγόριθμο δενδρικής αναζήτησης Μοντε Κάρλο και συγκεκριμένα στην ενίσχυση των επιμέρους σταδίων του. Αρχικά, στο Κεφάλαιο 4 προτείνονται τεχνικές που αφορούν στη βελτίωση της φάσης προσομοίωσης του αλγορίθμου με στόχο, κατά κύριο λόγο, τη μείωση της διακύμανσης που εισάγεται από την εκτέλεση τυχαίων προσομοιώσεων. Προς αυτή την κατεύθυνση ενσωματώνεται στο δέντρο αναζήτησης ένας ταξινομητής, ο οποίος εκπαιδεύεται σε δεδομένα που συλλέγονται από πραγματικές παρτίδες και τα αποτελέσματα των προσομοιώσεων αντικαθίστανται από τις προβλέψεις του μοντέλου. Επιπλέον, εισάγεται μία μεθοδολογία πρώιμης προσομοίωσης για κόμβους που βρίσκονται στα υψηλότερα επίπεδα του δέντρου προκειμένου να είναι πιο ακριβή τα δεδομένα που χρησιμοποιούνται τελικά ως είσοδος του ταξινομητή. Λαμβάνοντας υπόψη ότι το μέγεθος του χώρου ενεργειών μπορεί να επηρεάσει τη συμπεριφορά του αλγορίθμου, μία στοχαστική μέθοδος αξιολόγησης των κόμβων εξετάζεται επίσης, ανάλογα με το πλήθος των διαθέσιμων ενεργειών ανά περίπτωση. Οι επιμέρους τροποποιήσεις μελετήθηκαν ξεχωριστά αλλά και συνδυαστικά, οδηγώντας σε σημαντική αύξηση της απόδοσης του αλγορίθμου στα πειράματα που διενεργήθηκαν.

Στο επόμενο μέρος, η έρευνα επικεντρώνεται στη φάση επιλογής της δενδρικής αναζήτησης Μόντε Κάρλο (Κεφάλαιο 5). Στο πρώτο στάδιο, προτείνεται μία μεθοδολογία κλαδέματος προκειμένου η αναζήτηση να εστιάζει από νωρίς στις ενέργειες με τη μεγαλύτερη εκτιμώμενη αξία και να αξιοποιούνται βέλτιστα οι διαθέσιμοι υπολογιστικοί πόροι. Η συγκεκριμένη προσέγγιση βασίζεται σε δύο νευρωνικά δίκτυα τα οποία λειτουργούν συνδυαστικά ώστε να προσδιορίσουν το βέλτιστο υποσύνολο ενεργειών που μπορούν να απορριφθούν από το δέντρο αναζήτησης χωρίς να επηρεαστεί η βέλτιστη λύση καθώς και το βέλτιστο χρονικό σημείο στο οποίο πρέπει να εκτελεστεί το κλάδεμα. Καθώς σε περιβάλλοντα παιχνιδιών η βέλτιστη ενέργεια ανά κατάσταση δεν είναι γνωστή ούτε μετά την ολοκλήρωση μίας παρτίδας, στην παρούσα εργασία παρουσιάζεται επίσης ένα ειδικά σχεδιασμένο περιβάλλον μέσω του οποίου παράγονται τα κατάλληλα δεδομένα για την εκπαίδευση των δικτύων.

Στο ίδιο πλαίσιο, διερευνάται και η δυνατότητα ενίσχυσης του σταδίου επιλογής χωρίς την εισαγωγή *γνώσης πεδίου* (domain knowledge). Σε αυτή την περίπτωση ο στόχος είναι η ταχύτερη εκτίμηση της αξίας των ενεργειών με χρήση των στατιστικών παρόμοιων κόμβων οι οποίοι βρίσκονται σε υψηλότερα επίπεδα του δέντρου αναζήτησης. Για τον προσδιορισμό της ομοιότητας των καταστάσεων εφαρμόζονται δύο διαφορετικές μέθοδοι, οι οποίες έχουν σαν κοινή βάση τη χρήση της ακολουθίας των ενεργειών που οδηγεί σε μία κατάσταση του παιχνιδιού. Δεδομένου ότι στην πλειοψηφία των περιπτώσεων η συσχέτιση μεταξύ των ενεργειών που εκτελούνται και των καταστάσεων που προκύπτουν είναι μεγάλη, η επιλογή των ακολουθιών των ενεργειών ως κριτήριο ομοιότητας καταστάσεων απο-

δεικνύεται ότι μπορεί να συμβάλει στον προσδιορισμό καταστάσεων με παρεμφερή χαρακτηριστικά και κατ΄ επέκταση στην καλύτερη αξιοποίηση των στατιστικών από διαφορετικούς κόμβους του δέντρου αναζήτησης.

Στο τελευταίο τμήμα της διατριβής (Κεφάλαιο 6) αξιολογείται η επίδραση των τεχνικών επαύξησης δεδομένων στο πεδίο της ενισχυτικής μάθησης. Συγκεκριμένα, παρουσιάζεται μία μεθοδολογία γεννητικής επαύξησης κατά την οποία παράγονται εξ' ολοκλήρου νέα, συνθετικά δεδομένα σε αντιδιαστολή με τις κλασικές μεθόδους στις οποίες τα νέα δείγματα προκύπτουν μέσω επεξεργασίας των υπάρχοντων δεδομένων. Το συνολικό προτεινόμενο σύστημα αποτελείται από δύο ξεχωριστά δίκτυα διάχυσης τα οποία χρησιμοποιούνται για τη σύνθεση καταστάσεων με υψηλή και χαμηλή άμεση ανταμοιβή αντίστοιχα και από ένα μοντέλο αντίστροφης δυναμικής, ικανό να προβλέπει την ενέργεια που εκτλέστηκε μεταξύ δύο διαδοχικών καταστάσεων. Με αυτό τον τρόπο δημιουργούνται πλήρη δείγματα εκπαίδευσης τα οποία χρησιμοποιούνται σε συνδυασμό με τα πραγματικά δεδομένα, οδηγώντας σε αύξηση του ρυθμού βελτίωσης του πράκτορα κατά την διάρκεια της αλληλεπίδρασης του με το περιβάλλον.

Όλες οι μεθοδολογίες που παρουσιάζονται στην παρούσα εργασία, αναλύονται αρχικά σε θεωρητικό επίπεδο και στη συνέχεια υλοποιούνται και αξιολογούνται σε κατάλληλα περιβάλλοντα παιχνιδιών. Μέσω της πειραματικής διαδικασίας εξετάζονται και προσδιορίζονται οι βέλτιστες τιμές των σημαντικότερων υπερπαραμέτρων και οι τελικοί πράκτορες σε κάθε περίπτωση συγκρίνονται με αντίστοιχους αλγορίθμους που απαντώνται στη σχετική βιβλιογραφία. Τα αποτελέσματα κάθε ενότητας υπογραμμίζουν τη συμβολή των προτεινόμενων τεχνικών στην ερευνητική περιοχή που μελετήθηκε και μπορούν να λειτουργήσουν ως αφετηρία για περαιτέρω διερεύνηση του ευρύτερου πεδίου της εφαρμογής τεχνητής νοημοσύνης στην ανάπτυξη ευφυών πρακτόρων.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Ευφυείς Πράκτορες

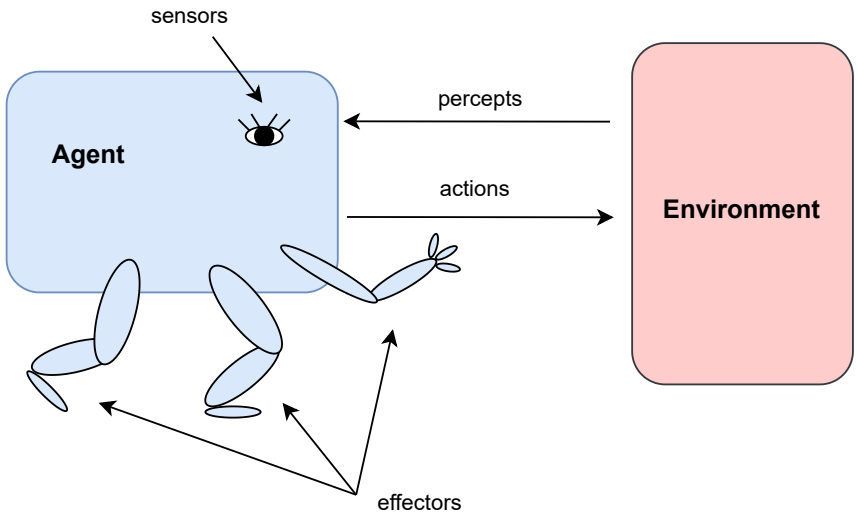
2.1.1 Εισαγωγή

Στο πεδίο της τεχνητής νοημοσύνης, ευφυές σύστημα είναι αυτό που επεξεργάζεται εσωτερική πληροφορία, προκειμένου να εκπληρώσει έναν στόχο [63]. Τέτοια μπορεί να είναι άνθρωποι ή ζώα αλλά και τεχνητά συστήματα όπως αισθητήρες και ρομπότ. Τα τελευταία χρόνια έχει αναπτυχθεί ιδιαίτερο ενδιαφέρον στην κατεύθυνση των τεχνητών ευφυών συστημάτων καθώς αποτελούν ένα πεδίο μελέτης με πολλές ανεξερεύνητες διαστάσεις. Μία κατηγορία ευφυών συστημάτων με πολύ σημαντική απήχηση είναι αυτή των *Αυτόνομων Τεχνητών Ευφυών Πρακτόρων* (Artificial Autonomous Intelligent Agents), οι οποίοι παρουσιάζουν πληθώρα εφαρμογών.

Στην ορολογία της τεχνητής νοημοσύνης, ο *Πράκτορας* (Agent) ορίζεται ως οποιαδήποτε οντότητα είναι ικανή να αντιλαμβάνεται το περιβάλλον μέσω *αισθητήρων* (sensors) και να επιδρά σε αυτό μέσω *ενεργοποιητών* (effectors) [86] (Σχήμα 2.1). *Ευφυής Πράκτορας* (ΕΠ) (Intelligent Agent - IA) είναι ένας πράκτορας ικανός να αποφασίσει ποιες ενέργειες πρέπει να κάνει, βασιζόμενος στην εμπειρία που έχει αποκτήσει [63]. Η *είσοδος* (input) που συλλέγει ο πράκτορας με τη χρήση των αισθητήρων του σε μία δεδομένη στιγμή καλείται *αντίληψη* (percept) του περιβάλλοντος. Για την συνολική πληροφορία που έχει συγκεντρώσει, χρησιμοποιείται ο όρος *αντιληπτική ακολουθία* (percept sequence) και είναι πρακτικά ό,τι χρειάζεται για να αποφασίσει τις επόμενες ενέργειες του.

Συνήθως ο πράκτορας αρχικοποιείται με μία προκαθορισμένη βασική γνώση για το περιβάλλον, η οποία θα τον βοηθήσει στα πρώτα στάδια που η αντιληπτική ακολουθία είναι σχετικά μικρή. Όσο όμως η πληροφορία που αντλεί από το περιβάλλον αυξάνεται θα πρέπει να είναι σε θέση να την αξιοποιεί για τις αποφάσεις του και να μη στηρίζεται μόνο στην αρχική γνώση που είχε. Αυτή είναι η έννοια της *αυτονομίας* (autonomy), η οποία με την πάροδο του χρόνου ενισχύεται καθώς λαμβάνει περισσότερες εισόδους.

Έχοντας κωδικοποιήσει την πληροφορία, ο πράκτορας χρειάζεται έναν τρόπο να την επεξεργαστεί ώστε να επιλέξει την κατάλληλη δράση. Μία τεχνική είναι ο σχηματισμός ενός πίνακα που θα περιγράφει την επιθυμητή ενέργεια για κάθε πιθανή αντιληπτική ακολουθία. Κάτι τέτοιο, προφανώς δεν είναι αποδοτικό για συστήματα με πολλές διαφορετικές εισόδους και καθώς η αντιληπτική ακολουθία αυξάνεται με την πάροδο του χρόνου, καθίσταται αδύνατο σε ένα πραγματικό σύστημα. Στη γενική



Σχήμα 2.1: Σχηματική αναπαράσταση ενός πράκτορα

περίπτωση, χρησιμοποιείται μία μέθοδος, η οποία μπορεί να αντιστοιχίσει μία οποιαδήποτε αντιληπτική ακολουθία σε μία *ενέργεια* (action). Αυτή η μέθοδος καλείται *συνάρτηση πράκτορα* (agent function) και είναι αυτή που ορίζει τη συμπεριφορά του στο περιβάλλον [86]. Η συνάρτηση πράκτορα είναι μία μαθηματική περιγραφή της αντιστοίχισης και στην πράξη υλοποιείται με ένα *πρόγραμμα πράκτορα* (agent program) που αποτελεί συστατικό του στοιχείο. Το πρόγραμμα είναι το ένα από τα δύο βασικά συστατικά της υλοποίησης ενός πράκτορα και εκτελείται συνεχώς σε κάποια υπολογιστική συσκευή. Η υπολογιστική συσκευή μαζί με το μηχανικό μέρος, που αποτελείται πρακτικά από τους αισθητήρες και τους ενεργοποιητές, συνιστούν την αρχιτεκτονική του η οποία είναι το δεύτερο βασικό συστατικό.

Ο στόχος ενός ευφυούς πράκτορα είναι να μπορεί να συμπεριφέρεται με τέτοιο τρόπο, ώστε να προκαλεί τις επιθυμητές αλλαγές στο περιβάλλον του. Ένας πράκτορας ο οποίος επιδρά επιτυχώς στο περιβάλλον κατ' αυτόν τον τρόπο ονομάζεται *ορθολογιστικός* (rational).

Η *λογικότητα* (rationality) ενός πράκτορα εξαρτάται από τις εξής παραμέτρους [86]:

1. Το βαθμό επίδοσής του
2. Το σύνολο των δυνατών ενεργειών του
3. Την αντιληπτική ακολουθία
4. Την αρχικοποιημένη γνώση του για το περιβάλλον

Ο βαθμός επίδοσης μπορεί να προκύψει από ένα μέτρο επίδοσης για την αξιολόγηση των καταστάσεων στις οποίες μεταβαίνει το περιβάλλον έπειτα από κάθε ενέργεια. Το μέτρο, φυσικά, διαφέρει ανάλογα με τα χαρακτηριστικά του περιβάλλοντος και είναι ευθύνη του σχεδιαστή να το ορίσει κατάλληλα. Ένας πράκτορας ο οποίος για κάθε πιθανή αντιληπτική ακολουθία επιλέγει την ενέργεια που μεγιστοποιεί το βαθμό επίδοσής του, στηριζόμενος στην αρχική του γνώση και την πληροφορία που έχει μέσω της αντίληψης καλείται *Ιδανικός Ορθολογιστικός Πράκτορας* (Ideal Rational Agent).

Για την πλήρη περιγραφή ενός πράκτορα απαιτείται ο προσδιορισμός πέντε παραμέτρων [28].

2.1.1.1 Περιβάλλον (Environment) Για την περιγραφή του περιβάλλοντος χρειάζεται να οριστούν όλες οι πιθανές καταστάσεις που μπορούν να υπάρξουν και η μέθοδος με την οποία γίνονται οι μεταβάσεις μεταξύ των καταστάσεων με την πάροδο του χρόνου. Επίσης πρέπει να οριστεί η *αρχική κατάσταση* (initial state).

2.1.1.2 Αισθητήριες Δυνατότητες (Sensing Capabilities) Περιγράφονται οι αισθητήρες του πράκτορα και οι αντίστοιχες είσοδοι από το περιβάλλον. Οι είσοδοι έχουν συνήθως τη μορφή διανύσματος και καθεμία αντιστοιχίζεται σε μία από τις διαστάσεις του περιβάλλοντος.

2.1.1.3 Ενέργειες (Actions) Η κάθε ενέργεια προσδιορίζεται από την αλλαγή που προκαλεί στην κατάσταση του περιβάλλοντος όταν εφαρμόζεται. Πολύπλοκες ενέργειες μπορούν να προκύψουν ως ακολουθίες των μεμονωμένων βασικών ενεργειών που υποστηρίζει ο πράκτορας.

2.1.1.4 Αρχική Γνώση (Drives – Preferences) Είναι η ενσωματωμένη "εμπειρία" που έχει ο πράκτορας όταν τοποθετείται στο περιβάλλον. Η αλληλεπίδραση των κανόνων που προκύπτουν από αυτή τη γνώση εξαρτάται από την κατάσταση του περιβάλλοντος αλλά και του ίδιου του πράκτορα, συνεπώς μπορεί να αλληλοσυμπληρώνονται αλλά και να συγκρούονται μεταξύ τους.

2.1.1.5 Μέθοδος Επιλογής Ενεργειών (Action Selection Architecture) Περιγράφει τον τρόπο με τον οποίο αποφασίζεται ποια ενέργεια θα εκτελεστεί. Η μέθοδος επιλογής λαμβάνει υπόψη τόσο την αρχική γνώση, όσο και την γνώση που έχει αποκτήσει ο πράκτορας από την εμπειρία του και είναι ιδιαίτερα σημαντική για την απόδοσή του.

Ο σωστός σχεδιασμός ενός πράκτορα προϋποθέτει την μελέτη και την ορθή κατανόηση αυτών των παραμέτρων. Το περιβάλλον και η μέθοδος επιλογής ενεργειών είναι αυτά που επηρεάζουν περισσότερο την απόδοση του, καθώς παρουσιάζουν τη μεγαλύτερη ποικιλία και θα εξεταστούν αναλυτικά στη συνέχεια.

2.1.2 Περιβάλλοντα Πρακτόρων

Το περιβάλλον περιλαμβάνει όλα τα στοιχεία με τα οποία μπορεί να αλληλεπιδράσει ο πράκτορας. Οι είσοδοι λαμβάνονται από το περιβάλλον με τη χρήση των αισθητήρων και οι ενεργοποιητές δρουν σε αυτό αλλάζοντας την κατάστασή του. Όλα τα περιβάλλοντα έχουν ορισμένες διαστάσεις/ιδιότητες με βάση τις οποίες μπορούν να κατηγοριοποιηθούν. Η υλοποίηση των πρακτόρων πρέπει να είναι προσαρμοσμένη σε αυτές τις ιδιότητες για την επίτευξη της καλύτερης δυνατής απόδοσης. Η κατηγοριοποίηση των περιβαλλόντων γίνεται με βάση τις παρακάτω ιδιότητες [86].

2.1.2.1 Παρατηρησιμότητα (Observability) Με τον όρο παρατηρησιμότητα ορίζεται η δυνατότητα να λαμβάνονται οι είσοδοι του περιβάλλοντος από τον πράκτορα. Με βάση αυτή τη δυνατότητα τα περιβάλλοντα διακρίνονται σε *πλήρως παρατηρήσιμα* (fully observable) και *μερικώς παρατηρήσιμα* (partially observable). Σε ένα πλήρως παρατηρήσιμο περιβάλλον ο πράκτορας έχει πρόσβαση σε όλες τις παραμέτρους/εισόδους που χρειάζεται για να πάρει μία απόφαση ενώ σε ένα μερικώς παρατηρήσιμο δεν είναι διαθέσιμη ολόκληρη η κατάσταση του περιβάλλοντος. Αυτό μπορεί να οφείλεται είτε σε θόρυβο, ο οποίος μειώνει την αντίληψη του πράκτορα, είτε σε αρχιτεκτονικούς

περιορισμούς, σε περίπτωση για παράδειγμα που δεν υπάρχουν όλοι οι απαραίτητοι αισθητήρες ή κάποιοι δεν λειτουργούν σωστά. Πιο σπάνια συναντάται και η περίπτωση μη παρατηρήσιμων περιβαλλόντων στα οποία ο πράκτορας δεν έχει τη δυνατότητα να λάβει καμία είσοδο, χωρίς ωστόσο αυτό να συνεπάγεται απαραίτητα αδυναμία επίτευξης του στόχου.

2.1.2.2 Πλήθος Πρακτόρων Ανάλογα με τον αριθμό των πρακτόρων που συνυπάρχουν σε ένα περιβάλλον, αυτό μπορεί να είναι *ενός-πράκτορα* (single-agent) ή *πολλών-πρακτόρων* (multi-agent). Για να είναι ξεκάθαρος αυτός ο διαχωρισμός πρέπει να είναι δυνατή η διάκριση των πρακτόρων από τα απλά αντικείμενα του περιβάλλοντος, διαφορετικά μπορεί να υπάρξει σύγχυση. Στα περιβάλλοντα πολλών-πρακτόρων οι στόχοι τους είναι δυνατόν να είναι κοινοί ή αλληλοσυγκρουόμενοι, οπότε ανάλογα με τη συμπεριφορά τους τα περιβάλλοντα μπορούν να διακριθούν επιπλέον σε *συνεργατικά* (cooperative) και *ανταγωνιστικά* (competitive) αντίστοιχα.

2.1.2.3 Προβλεψιμότητα Με την έννοια της προβλεψιμότητας ορίζεται η εξάρτηση της μετάβασης του περιβάλλοντος σε μία κατάσταση από οποιονδήποτε παράγοντα πέραν της τρέχουσας κατάστασης και της ενέργειας που εκτελεί ο πράκτορας. Ένα περιβάλλον του οποίου οι καταστάσεις δεν εξαρτώνται από κανέναν τέτοιο παράγοντα ονομάζεται *ντετερμινιστικό* (deterministic), ενώ σε αντίθετη περίπτωση *στοχαστικό* (stochastic). Αν ένα περιβάλλον δεν είναι πλήρως παρατηρήσιμο ή ντετερμινιστικό καλείται *αβέβαιο* (uncertain). Στη γενική περίπτωση ένα πραγματικό περιβάλλον πρέπει να αντιμετωπίζεται ως στοχαστικό γιατί υπάρχουν πολλές παράμετροι με απρόβλεπτη συμπεριφορά που μπορούν να επηρεάσουν τη μετάβαση των καταστάσεων.

2.1.2.4 Μνήμη Τα περιβάλλοντα που έχουν την ιδιότητα της μνήμης, δηλαδή αυτά στα οποία μία μελλοντική κατάσταση επηρεάζεται από τις προηγούμενες, ονομάζονται *ακολουθιακά* (sequential). Για παράδειγμα σε ένα παιχνίδι στρατηγικής, μία ενέργεια του πράκτορα μπορεί να είναι καθοριστική για την εξέλιξη της πορείας του παιχνιδιού. Στα περιβάλλοντα χωρίς μνήμη ο πράκτορας δε χρειάζεται να αποθηκεύει την αντιληπτική ακολουθία. Η αντίληψη των εισόδων γίνεται ανά επεισόδια, τα οποία είναι ανεξάρτητα μεταξύ τους, και γι' αυτό ονομάζονται *επεισοδιακά* (episodic) περιβάλλοντα. Τα επεισοδιακά είναι σαφώς απλούστερα από τα ακολουθιακά καθώς απαιτούν σημαντικά λιγότερους υπολογισμούς για την επιλογή της ενέργειας του πράκτορα.

2.1.2.5 Δυναμικότητα Ένα περιβάλλον χαρακτηρίζεται *δυναμικό* (dynamic) αν μπορεί να αλλάζει καταστάσεις κατά τη διάρκεια που ο πράκτορας εκτελεί τους απαραίτητους υπολογισμούς για να πάρει μία απόφαση. Αντίστοιχα καλείται *στατικό* (static) όταν προϋπόθεση για να αλλάξει κατάσταση είναι η εκτέλεση κάποιας ενέργειας από τον πράκτορα. Τα δυναμικά περιβάλλοντα είναι πιο πολύπλοκα, αφού ο πράκτορας πρέπει συνεχώς να λαμβάνει υπόψη τις διάφορες αλλαγές που συμβαίνουν.

2.1.2.6 Συνέχεια Η συνέχεια είναι μία ιδιότητα που χαρακτηρίζει ένα περιβάλλον με βάση τον τρόπο διαχείρισης του χρόνου. Έτσι, τα περιβάλλοντα κατατάσσονται σε *διακριτά* (discrete) και *συνεχή* (continuous). Διακριτά είναι αυτά στα οποία μπορεί να οριστεί ξεκάθαρα ένα χρονικό διάστημα για κάθε μία κατάσταση μεμονωμένα. Αντίθετα σε ένα συνεχές περιβάλλον δεν υπάρχει σαφής αριθμός καταστάσεων καθώς αυτές μεταβάλλονται αδιάλειπτα. Ένας πράκτορας για παράδειγμα ο οποίος λαμβάνει με τους αισθητήρες του ένα ηχητικό σήμα για ένα χρονικό διάστημα ανήκει σε συνεχές περιβάλλον, ενώ ένας ο οποίος αποφασίζει τις κινήσεις σε μία παρτίδα πόκερ σε διακριτό.

2.1.2.7 Γνώση Κανόνων Η γνώση των κανόνων είναι μία ιδιότητα που επηρεάζει έμμεσα το περιβάλλον. Πρόκειται στην πραγματικότητα για την αρχικοποιημένη γνώση που έχουν οι πράκτορες σχετικά με τις θεμελιώδεις αρχές του περιβάλλοντος. Αν οι πράκτορες γνωρίζουν τους βασικούς κανόνες που το διέπουν, τότε το περιβάλλον είναι *γνωστό* (known) διαφορετικά είναι *άγνωστο* (unknown). Σε ένα άγνωστο περιβάλλον, ο πράκτορας μαθαίνει τελικά τους κανόνες μέσω της εμπειρίας που αποκτά, οπότε η διαφορά του με ένα γνωστό είναι πρακτικά η αρχική κατάσταση του πράκτορα.

2.1.3 Κατηγορίες Πρακτόρων

Ο δεύτερος παράγοντας που επηρεάζει την αποτελεσματικότητα ενός πράκτορα εξίσου σημαντικά με το περιβάλλον του είναι η συνάρτηση επιλογής ενεργειών. Ιδανικά η συνάρτηση επιλογής πρέπει να μπορεί να περιγράψει ένα γενικό σχήμα που να καλύπτει όλες τις πιθανές αντιληπτικές ακολουθίες με όσο το δυνατόν απλούστερη λογική και λιγότερους υπολογισμούς. Με βάση τον τύπο του προγράμματος που υλοποιεί αυτή τη συνάρτηση οι πράκτορες μπορούν να καταταχθούν σε τέσσερις κατηγορίες [86].

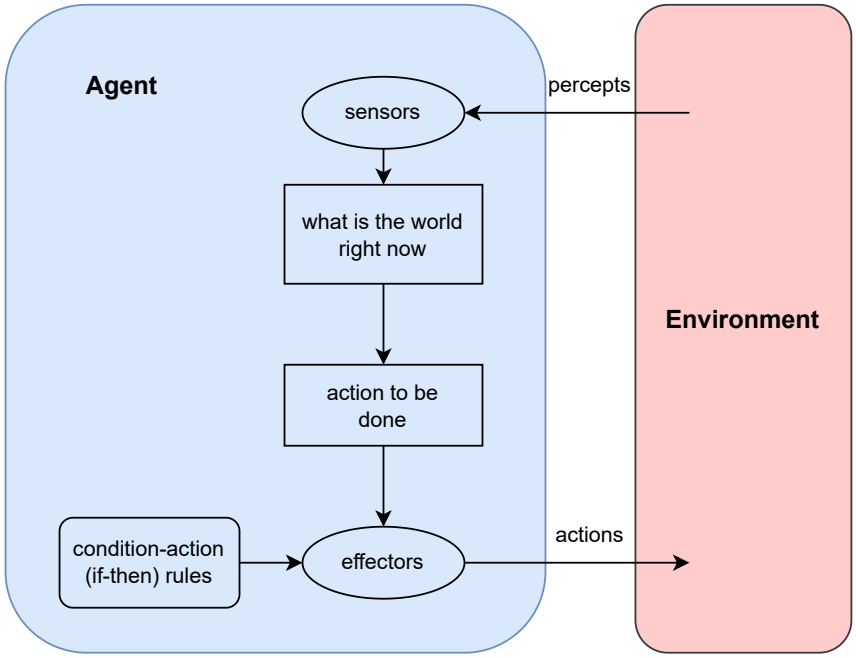
2.1.3.1 Πράκτορες Απλής Αντανάκλασης (Simple Reflex Agents)

Σε αυτή την κατηγορία ανήκουν οι πράκτορες που στηρίζουν κάθε φορά την επιλογή τους μόνο στην αντίληψη της δεδομένης κατάστασης. Για το σκοπό αυτό λειτουργούν με κανόνες που αντιστοιχίζουν κάποιες συνθήκες σε προκαθορισμένες ενέργειες. Οι κανόνες αυτοί ονομάζονται *κανόνες συνθήκης-ενέργειας* (condition-action rules).

Αρχικά ένας πράκτορας απλής αντανάκλασης λαμβάνει τις εισόδους από το περιβάλλον με τη χρήση των αισθητήρων του. Έπειτα το πρόγραμμα πράκτορα διατρέχει το σύνολο των κανόνων και ελέγχει τις συνθήκες που ορίζονται σε αυτό. Για την πρώτη που ικανοποιείται, αν υπάρχει τέτοια, εκτελεί την αντίστοιχη – σύμφωνα με τον κανόνα – ενέργεια, μέσω των ενεργοποιητών. Αυτή η λειτουργία απεικονίζεται στο Σχήμα 2.2.

Οι πράκτορες αυτού του τύπου είναι εξαιρετικά απλοί στην υλοποίηση τους, καθώς δεν ενδιαφέρονται για τις καταστάσεις του παρελθόντος και συνεπώς δεν χρειάζεται να αποθηκεύουν και να επεξεργάζονται ολόκληρη την αντιληπτική ακολουθία, παρά μόνο την τρέχουσα αντίληψη. Αυτό, ωστόσο, έχει αρνητικές συνέπειες στην αποτελεσματικότητά τους, που οφείλονται στη μη αξιοποίηση χρήσιμων πληροφοριών από τις προηγούμενες καταστάσεις.

Βασική προϋπόθεση για να έχει ένας πράκτορας απλής αντανάκλασης την επιθυμητή συμπεριφορά είναι να βρίσκεται σε πλήρως παρατηρήσιμο περιβάλλον. Αν το περιβάλλον είναι μερικώς παρατηρήσιμο, είναι πιθανό η τρέχουσα αντίληψη να μην αρκεί για την ανίχνευση μίας συνθήκης κάποιου κανόνα η οποία στην πραγματικότητα ικανοποιείται. Ένας άλλος κίνδυνος που μπορεί να προκύψει σε τέτοιο περιβάλλον είναι να εγκλωβιστεί ο πράκτορας σε έναν ατέρμονο βρόχο. Αυτό μπορεί να αντιμετωπιστεί με την εισαγωγή τυχαιότητας στις κινήσεις του πράκτορα, όμως δεν ενδείκνυται γιατί στην πλειοψηφία των περιπτώσεων οι πράκτορες που εφαρμόζουν αυτή την τεχνική δεν είναι ορθολογιστικοί.



Σχήμα 2.2: Πράκτορας απλής αντανάκλασης

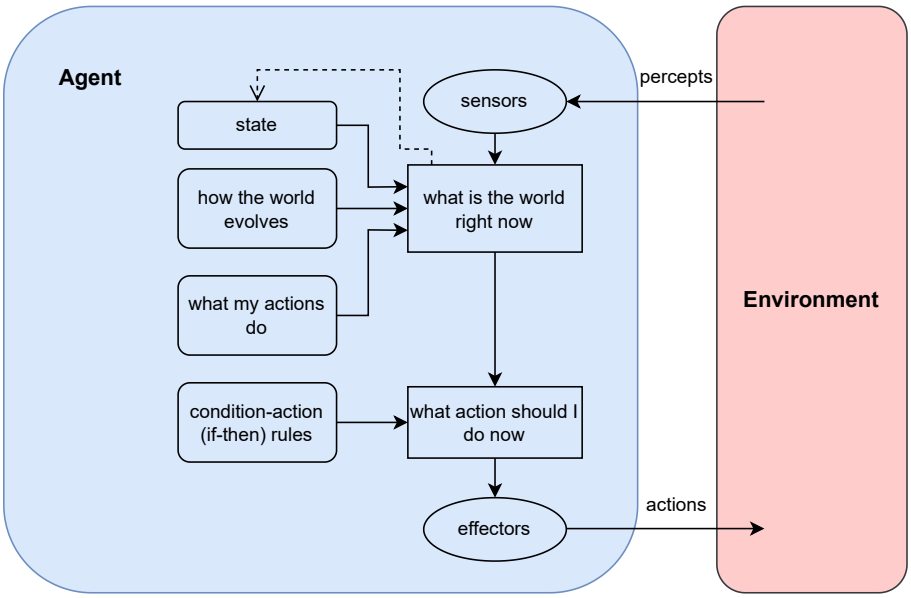
2.1.3.2 Πράκτορες Αντανάκλασης βασισμένοι σε Μοντέλο (Model-based Reflex Agents)

Αυτοί οι πράκτορες αποτελούν επέκταση των πρακτόρων απλής αντανάκλασης, έτσι ώστε να είναι ικανοί να παίρνουν τη σωστή απόφαση και σε μερικώς παρατηρήσιμα περιβάλλοντα. Για να το πετύχουν αυτό διατηρούν ανά πάσα στιγμή μία *εσωτερική κατάσταση* (internal state), που βασίζεται και σε προηγούμενες εισόδους της αντιληπτικής ακολουθίας, προκειμένου να αντισταθμίσουν έτσι την απουσία κάποιων από τις εισόδους της τρέχουσας κατάστασης.

Η εσωτερική κατάσταση ανανεώνεται με κάθε νέα αντίληψη του περιβάλλοντος από τον πράκτορα. Για να μπορεί να γίνεται η ανανέωση επιτυχώς, απαιτείται επιπλέον γνώση για το περιβάλλον σε δύο βασικές κατευθύνσεις. Η πρώτη αφορά στην εξέλιξη του περιβάλλοντος ανεξάρτητα από την ύπαρξη και τη δραστηριότητα του πράκτορα και προϋποθέτει τη γνώση των βασικών κανόνων-νόμων που ισχύουν και δρουν μόνιμα σε αυτό. Η δεύτερη απαραίτητη πληροφορία σχετίζεται με την αλλαγή της κατάστασης του περιβάλλοντος σε συνάρτηση με τις ενέργειες του πράκτορα.

Η γνώση για τον τρόπο με τον οποίο εξελίσσεται ο "κόσμος" στον οποίο λειτουργεί ο πράκτορας όπως περιγράφηκε ονομάζεται μοντέλο και οι πράκτορες που την αξιοποιούν για να ενεργήσουν κατάλληλα, *πράκτορες βασισμένοι σε μοντέλο* (model-based agents). Οι πράκτορες αντανάκλασης βασισμένοι σε μοντέλο συνδυάζουν την αντίληψη με την εσωτερική κατάσταση και τη γνώση για την λειτουργία του περιβάλλοντος τόσο ανεξάρτητα όσο και υπό την επίδραση των ενεργειών τους για να σχηματίσουν την νέα κατάσταση. Στη συνέχεια λειτουργούν όπως οι πράκτορες απλής αντανάκλασης χρησιμοποιώντας αυτή την κατάσταση.

Παρακάτω παρουσιάζεται αυτή η διαδικασία με τη μορφή διαγράμματος (Σχήμα 2.3).



Σχήμα 2.3: Πράκτορας αντανάκλασης βασισμένος σε μοντέλο

2.1.3.3 Πράκτορες βασισμένοι σε Στόχους (Goal-based Agents)

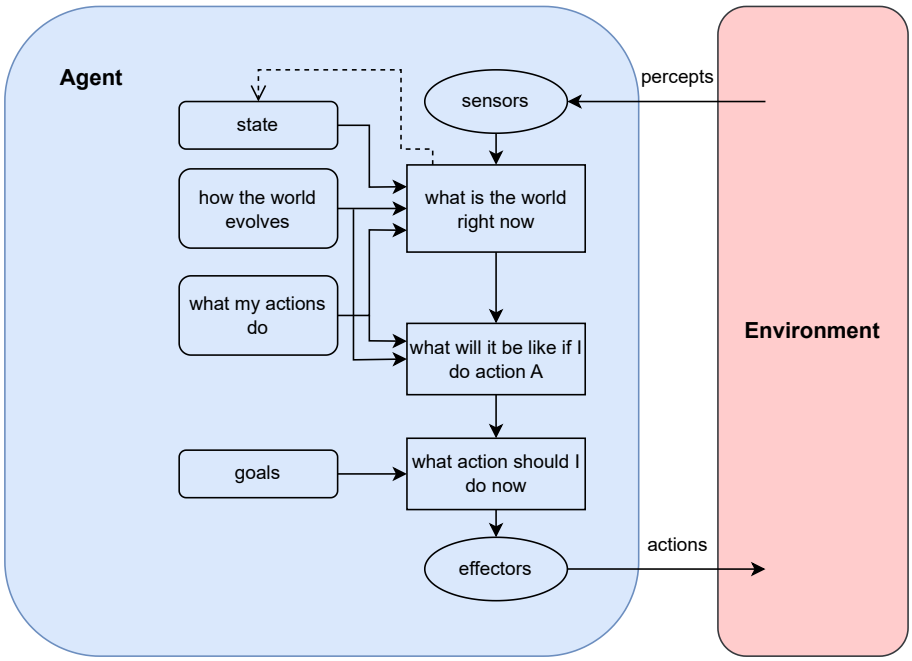
Οι πράκτορες που βασίζονται σε στόχους επιλέγουν κάθε φορά την κίνηση που σύμφωνα με τους υπολογισμούς τους θα τους φέρει πιο κοντά στην επίτευξη αυτών. Κάποιες φορές ενδέχεται μία ενέργεια να αρκεί για την πραγματοποίηση του στόχου αλλά συνήθως απαιτείται μία ακολουθία ενεργειών (είναι πιθανόν να υπάρχουν περισσότερες από μία τέτοιες διαφορετικές ακολουθίες). Αυτό σημαίνει ότι ο πράκτορας πρέπει να είναι σε θέση να κάνει έναν σχεδιασμό αυτής της ακολουθίας και να προαποφασίσει τουλάχιστον κάποιες από τις μελλοντικές του ενέργειες. Για τον σχεδιασμό χρησιμοποιεί το μοντέλο του "κόσμου" ώστε να μπορεί να προβλέψει και να αξιολογήσει τις καταστάσεις που θα προκύψουν από τις πιθανές ενέργειές του και να επιλέξει αυτές που εξυπηρετούν τους στόχους του.

Σε σύγκριση με τους πράκτορες αντανάκλασης, οι οποίοι αντιστοιχίζουν μία κατάσταση σε κάποια ενέργεια, οι πράκτορες που βασίζονται σε στόχους είναι αρκετά πιο πολύπλοκοι καθώς ακόμα και σε περιπτώσεις που τελικά καταλήγουν στην ίδια απόφαση έχουν χρησιμοποιήσει πολύ περισσότερους υπολογιστικούς πόρους. Όμως έχουν το πλεονέκτημα ότι μπορούν και προσαρμόζονται στις αλλαγές του περιβάλλοντος, κάτι που δεν ισχύει για τους πράκτορες αντανάκλασης εφόσον οι ενέργειές τους είναι προκαθορισμένες ανάλογα με την κατάσταση.

Επίσης είναι πολύ εύκολο για έναν βασισμένο σε στόχους πράκτορα να χρησιμοποιηθεί για την επίλυση διαφορετικών προβλημάτων. Το γεγονός ότι μπορεί να σχεδιάσει το πλάνο του χρησιμοποιώντας κάποια "λογική" του επιτρέπει να αναπροσαρμόζεται σε ένα πρόβλημα αρκεί να ορίζονται κατάλληλα οι εκάστοτε στόχοι. Αντίθετα σε έναν πράκτορα αντανάκλασης θα έπρεπε να οριστούν από την αρχή όλοι οι κανόνες και σε κάθε συνθήκη να αντιστοιχιστεί η νέα επιθυμητή ενέργεια.

2.1.3.4 Πράκτορες βασισμένοι στην Ωφελιμότητα (Utility-based Agents)

Οι πράκτορες που βασίζονται σε στόχους μπορούν πολλές φορές να τους πετύχουν με αρκετούς διαφορετικούς τρόπους. Στην περίπτωση που ελέγχεται μόνο η επίτευξή τους, όλες οι ακολουθίες

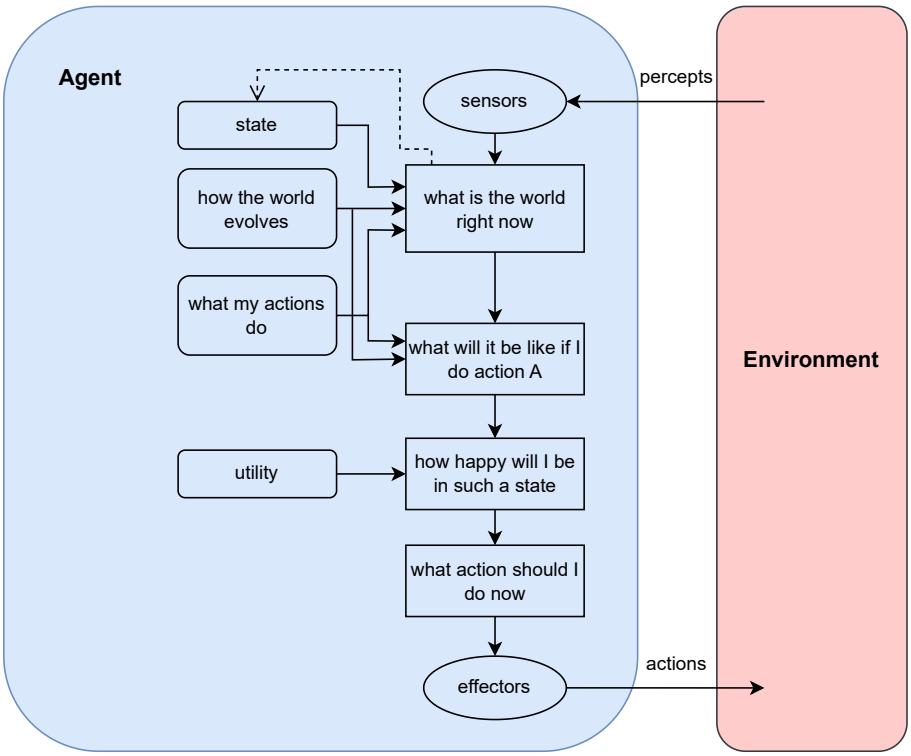


Σχήμα 2.4: Πράκτορας βασισμένος σε στόχους

ενεργειών που το καταφέρνουν κρίνονται επιτυχείς, αυτό όμως δεν σημαίνει ότι είναι όλες το ίδιο αποδοτικές. Αν εξεταστούν και άλλες παράμετροι, όπως για παράδειγμα ο απαιτούμενος χρόνος για την ολοκλήρωση της διαδικασίας, τότε κάποιες ακολουθίες είναι "καλύτερες" από κάποιες άλλες. Αυτή η διαβάθμιση μεταξύ των διαφορετικών ακολουθιών που οδηγούν στην επίτευξη των στόχων εκφράζεται με την έννοια της ωφελιμότητας.

Οι πράκτορες που βασίζονται στην ωφελιμότητα χρειάζονται ένα μέτρο ώστε να μπορούν να επιλέξουν την καλύτερη δυνατή ακολουθία. Για το σκοπό αυτό χρησιμοποιείται μία μέθοδος η οποία καλείται συνάρτηση ωφελιμότητας και επιστρέφει ένα τέτοιο μέτρο για κάθε πιθανή ενέργεια. Για να είναι ορθολογιστικός ο πράκτορας θα πρέπει η συνάρτηση ωφελιμότητας να συμβαδίζει με το μέτρο επίδοσης του πράκτορα, δηλαδή οι ενέργειες που επιλέγονται να οδηγούν σε καταστάσεις που μεγιστοποιούν το βαθμό επίδοσης.

Ένας πράκτορας ωφελιμότητας ακολουθεί όλα τα βήματα που κάνει και ένας πράκτορας που βασίζεται σε στόχους και ελέγχει επιπλέον την ωφελιμότητα της κάθε ενέργειας προτού αποφασίσει. Στην πραγματικότητα, επειδή τα περισσότερα περιβάλλοντα είναι στοχαστικά, ελέγχει την προσδοκώμενη ωφελιμότητα αφού δεν μπορεί να προσδιοριστεί με βεβαιότητα. Εκτός από το ότι είναι πιο αποδοτικοί, οι πράκτορες που βασίζονται στην ωφελιμότητα έχουν δύο επιπλέον πλεονεκτήματα. Το πρώτο αφορά στους αλληλοσυγκρουόμενους στόχους. Τέτοιοι στόχοι είναι πιο συχνοί σε περιβάλλοντα πολλών πρακτόρων, όμως μπορεί να υπάρχουν και σε περιβάλλοντα με έναν μοναδικό πράκτορα. Σε αυτές τις περιπτώσεις ο πράκτορας μπορεί εύκολα να αποφασίσει, με τη χρήση της συνάρτησης ωφελιμότητας, ποιοι στόχοι είναι πιο σημαντικοί και πρέπει να προτιμηθούν έναντι άλλων οι οποίοι θα πρέπει να απορριφθούν. Το δεύτερο πλεονέκτημά τους σχετίζεται επίσης με περιβάλλοντα στα οποία υπάρχουν περισσότεροι από έναν στόχοι, χωρίς όμως απαραίτητα να είναι αντικρουόμενοι. Όταν δεν υπάρχει βεβαιότητα ότι κάποιος από τους στόχους μπορεί να επιτευχθεί, η ωφελιμότητα των στόχων μπορεί να "υποδείξει" αυτούς στους οποίους πρέπει να δοθεί προτεραιότητα με βάση την πιθανότητά τους να πραγματοποιηθούν.



Σχήμα 2.5: Πράκτορας βασισμένος στην ωφελιμότητα

2.1.3.5 Πράκτορες Εκμάθησης (Learning Agents)

Η ανάπτυξη του προγράμματος ενός πράκτορα "χειροκίνητα" είναι μία πολύ απαιτητική και καθόλου αποδοτική διαδικασία. Αφενός είναι ιδιαίτερα χρονοβόρα, αφετέρου είναι δύσκολο να προβλεφθούν όλες οι πιθανές καταστάσεις του "κόσμου" και είναι πιθανό να υπάρχουν κενά στην περιγραφή του προγράμματος. Η τεχνική που χρησιμοποιείται κατά κύριο λόγο σήμερα είναι η δημιουργία πρακτόρων ικανών να εκπαιδευτούν και να προσαρμοστούν αυτόματα στις συνθήκες του περιβάλλοντος. Αυτοί οι πράκτορες ονομάζονται *πράκτορες εκμάθησης* (learning agents).

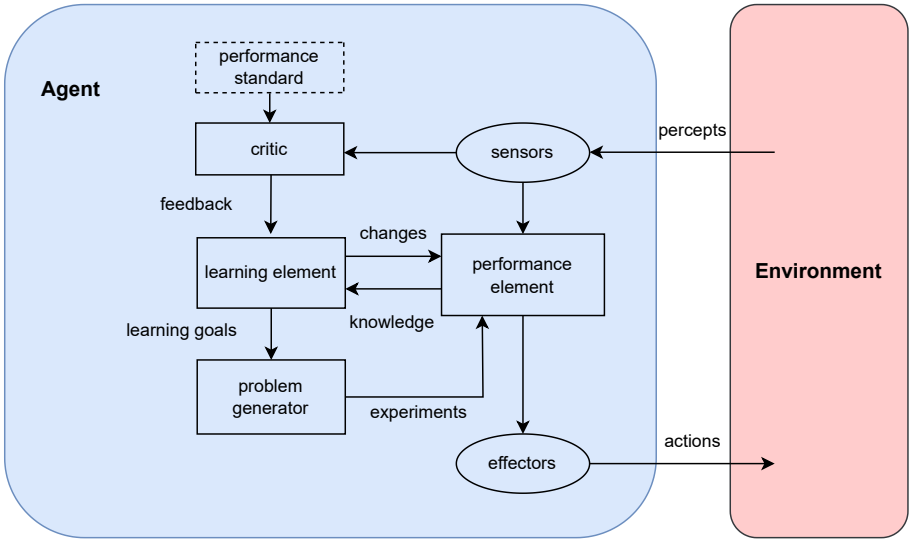
Η δομή ενός πράκτορα εκμάθησης φαίνεται στο Σχήμα 2.6. Αποτελείται από τέσσερα βασικά στοιχεία [86], τα οποία είναι:

2.1.3.5.1 Στοιχείο-Κριτής (Critic) Αυτό το στοιχείο αξιολογεί την επίδοση του πράκτορα σύμφωνα με κάποιο *πρότυπο επίδοσης* (performance standard) και ενημερώνει ανάλογα το στοιχείο εκμάθησης. Είναι απαραίτητο γιατί η αντίληψη που λαμβάνει ο πράκτορας μέσω των αισθητήρων περιγράφει μεν την τρέχουσα κατάσταση, χωρίς όμως να παρέχει κάποια πληροφορία για το αν η κατάσταση αυτή είναι επιθυμητή ή όχι.

2.1.3.5.2 Στοιχείο Εκμάθησης (Learning Element) Το στοιχείο εκμάθησης είναι υπεύθυνο για την προσαρμογή του πράκτορα στα νέα δεδομένα, με βάση την *ανάδραση* (feedback) που παίρνει από το στοιχείο-κριτή. Ουσιαστικά υλοποιεί την λογική της βελτίωσης του μοντέλου που χρησιμοποιεί ο πράκτορας για να πάρει τις αποφάσεις του.

2.1.3.5.3 Στοιχείο Επίδοσης (Performance Element) Πρόκειται για το τμήμα του πράκτορα που υλοποιεί τη συνάρτηση επιλογής των ενεργειών και δρα στο περιβάλλον μέσω των ενεργοποιητών.

2.1.3.5.4 Γεννήτρια Προβλημάτων (Problem Generator) Με τα μέχρι στιγμής ορισμένα στοιχεία, ο πράκτορας επιλέγει κάθε φορά την ενέργεια που του υποδεικνύει το στοιχείο επίδοσης, έτσι όπως έχει διαμορφωθεί από την αλληλεπίδραση του με το στοιχείο εκμάθησης. Όμως στο στάδιο της εκπαίδευσης, στόχος είναι ο πράκτορας να ανακαλύψει όσο το δυνατόν περισσότερες πιθανές καταστάσεις και να τις κωδικοποιήσει κατάλληλα για να μπορεί να τις αναγνωρίσει στο μέλλον. Αυτό το ρόλο αναλαμβάνει η γεννήτρια προβλημάτων που συνδυάζεται με το στοιχείο εκμάθησης και προτείνει ενέργειες που θα οδηγήσουν σε νέες καταστάσεις οι οποίες δεν έχουν εξερευνηθεί ακόμα, επιδιώκοντας μακροπρόθεσμα τη βελτίωση της συμπεριφοράς του πράκτορα.



Σχήμα 2.6: Πράκτορας εκμάθησης

Προηγουμένως θεωρήσαμε ότι το στοιχείο-κριτής χρησιμοποιεί κάποια μέθοδο με την οποία η αντίληψη μίας κατάστασης εκλαμβάνεται ως *ανταμοιβή* (reward) ή *ποινή* (penalty) σύμφωνα με το πρότυπο επίδοσης. Η υπόθεση αυτή όμως, σε ένα πραγματικό περιβάλλον δεν ισχύει για δύο λόγους. Πρώτον, ο προσδιορισμός μίας τέτοιας *συνάρτησης ανταμοιβής* (reward function) γίνεται εμπειρικά και μπορεί να αποδειχθεί λανθασμένος και δεύτερον, στις περισσότερες περιπτώσεις απαιτεί το συνδυασμό πολλών παραμέτρων, των οποίων η βαρύτητα δεν είναι δυνατόν να οριστεί εκ των προτέρων [85].

Συνεπώς για να είναι δυνατή η υλοποίηση ενός πράκτορα εκμάθησης είναι αναγκαίο να μπορεί να καθοριστεί η συνάρτηση ανταμοιβής με τη χρήση:

- μέτρων της συμπεριφοράς του πράκτορα υπό διάφορες συνθήκες
- μέτρων των εισόδων του πράκτορα από τους αισθητήρες του
- του μοντέλου του περιβάλλοντος στο οποίο λειτουργεί

Αυτή η διαδικασία είναι γνωστή ως *Αντίστροφη Ενισχυτική Μάθηση* (Inverse Reinforcement Learning) και αποτελεί μέχρι στιγμής πεδίο έρευνας όσον αφορά τις προϋποθέσεις, την πολυπλοκότητα και τους πιθανούς αλγορίθμους επίλυσης.

2.2 Μηχανική Μάθηση

2.2.1 Εισαγωγή

Η *Μηχανική Μάθηση* (Machine Learning) ορίζεται ως η μελέτη αλγορίθμων υπολογιστών που μπορούν να βελτιωθούν αυτόματα μέσω της εμπειρίας και με τη χρήση δεδομένων [65]. Αυτό σημαίνει ότι δε χρειάζεται να είναι ρητά προγραμματισμένες οι ενέργειες που θα πρέπει να εκτελεστούν αλλά το σύστημα μπορεί να "εκπαιδευτεί" πάνω σε ένα συγκεκριμένο πρόβλημα και στη συνέχεια να παίρνει αποφάσεις ή (συνήθως) να κάνει προβλέψεις με βάση προηγούμενα δείγματα.

Οι αλγόριθμοι μηχανικής μάθησης αρχικά τροφοδοτούνται με δεδομένα τα οποία χρησιμοποιούν προκειμένου να "αναγνωρίσουν" συσχετίσεις μεταξύ των διαφόρων μεταβλητών. Το στάδιο αυτό καλείται φάση εκπαίδευσης και τα δεδομένα που χρησιμοποιούνται για αυτό το σκοπό *δεδομένα εκπαίδευσης* (training data). Μέσω αυτής της διαδικασίας ένα μοντέλο μηχανικής μάθησης επεξεργάζεται επαναληπτικά τα διαθέσιμα δεδομένα και επαναξιολογείται ανά χρονικά διαστήματα μέχρι να σταματήσει να βελτιώνεται (ή να φτάσει σε ένα προκαθορισμένο επιθυμητό επίπεδο), οπότε και ολοκληρώνεται η εκπαίδευση. Στη συνέχεια, ακολουθεί μία φάση αξιολόγησης κατά την οποία το μοντέλο ελέγχεται σε "άγνωστα" δεδομένα – τα οποία είναι διαφορετικά από αυτά στα οποία είχε εκπαιδευτεί – και ονομάζονται *δεδομένα ελέγχου* (test data) και υπολογίζονται οι κατάλληλες μετρικές ώστε να μπορεί να ποσοτικοποιηθεί η απόδοση του. Αφού ολοκληρωθεί και το στάδιο αξιολόγησης, το μοντέλο επανεκπαιδεύεται στο σύνολο των δεδομένων (δεδομένα εκπαίδευσης και ελέγχου) και διατίθεται προς χρήση.

Συνήθως, οι αλγόριθμοι μηχανικής μάθησης είναι ιδιαίτερα χρήσιμοι σε προβλήματα για τα οποία δεν υπάρχει γνωστός, βέλτιστος τρόπος επίλυσης για όλες τις περιπτώσεις ή υπάρχει αλλά είναι αδύνατο ή μη αποδοτικό να υλοποιηθεί στην πράξη (λόγω υπολογιστικού κόστους και χρονικών περιορισμών). Οι τεχνικές μηχανικής μάθησης εκτείνονται σε ένα πάρα πολύ ευρύ πεδίο κλάδων πέραν της επιστήμης των υπολογιστών όπως ιατρική, τηλεπικοινωνίες, οικονομικές επιστήμες κ.λπ. με εφαρμογές σε αναγνώριση εικόνας, πρόβλεψη χρονοσειρών, επεξεργασία φυσικής γλώσσας, συστήματα συστάσεων, μοντελοποίηση χρηστών κ.α.

2.2.2 Κατηγορίες Μεθόδων Μηχανικής Μάθησης

Ανάλογα με τη φύση του προβλήματος, τον τύπο των δεδομένων και την προσέγγιση της διαδικασίας εκπαίδευσης των μοντέλων, οι τεχνικές μηχανικής μάθησης μπορούν να ταξινομηθούν σε τρεις κύριες κατηγορίες· στις τεχνικές *Επιβλεπόμενης Μάθησης* (Supervised Learning), *Μη Επιβλεπόμενης Μάθησης* (Unsupervised Learning) και *Ενισχυτικής Μάθησης* (Reinforcement Learning - RL).

2.2.2.1 Επιβλεπόμενη Μάθηση

Στην επιβλεπόμενη μάθηση, ανήκουν οι περιπτώσεις στις οποίες το διαθέσιμο σύνολο δεδομένων αποτελείται από ζεύγη εισόδων – επιθυμητής εξόδου [88]. Κάθε είσοδος είναι ουσιαστικά ένα σύνολο από *μεταβλητές/χαρακτηριστικά* (features) και κάθε ζεύγος εισόδου – εξόδου καλείται *δείγμα* (sample), ενώ οι έξοδοι καλούνται και *ετικέτες* (labels). Στα προβλήματα επιβλεπόμενης μάθησης υπάρχουν κάποια (παρελθοντικά) δείγματα για τα οποία είναι γνωστή η έξοδος που προήλθε από κάθε είσοδο

χωρίς ωστόσο να υπάρχει πληροφορία για τον τρόπο με τον οποίο σχετίζονται μεταξύ τους. Αυτή την αντιστοίχιση καλείται να "ανακαλύψει" ένας αλγόριθμος επιβλεπόμενης μάθησης χρησιμοποιώντας τα διαθέσιμα δεδομένα προκειμένου να γενικεύσει τις μεταξύ τους συσχετίσεις και να "κατασκευάσει" μία συνάρτηση με την οποία θα μπορεί να προβλέψει την επιθυμητή έξοδο για οποιαδήποτε νέα είσοδο [84].

Ένας αλγόριθμος επιβλεπόμενης μάθησης αρχικοποιεί κατά βάση με τυχαίο τρόπο τις εσωτερικές παραμέτρους – βάρη του μοντέλου και ξεκινάει να κάνει προβλέψεις για τις εισόδους των διαθέσιμων δεδομένων. Στη συνέχεια οι προβλέψεις αυτές συγκρίνονται με τις πραγματικές (γνωστές) εξόδους και υπολογίζεται ένα σφάλμα εκπαίδευσης. Το σφάλμα εκπαίδευσης χρησιμοποιείται έπειτα για την ανανέωση των παραμέτρων με στόχο τη βελτίωση του συστήματος και η διαδικασία επαναλαμβάνεται. Με ορθή χρήση των κατάλληλων αλγορίθμων το σφάλμα εκπαίδευσης μειώνεται σταδιακά και τελικά συγκλίνει, οπότε και η εκπαίδευση σταματάει.

Η πλειοψηφία των προβλημάτων μηχανικής μάθησης εμπίπτουν στην κατηγορία της επιβλεπόμενης μάθησης, γύρω από την οποία επικεντρώνεται και το μεγαλύτερο μέρος της ακαδημαϊκής έρευνας. Συγκεκριμένα υπάρχουν δύο βασικές κατηγορίες προβλημάτων επιβλεπόμενης μάθησης, τα προβλήματα *ταξινόμησης* (classification) και *παλινδρόμησης* (regression). Η διαφορά τους έγκειται στον τύπο των δεδομένων εξόδου. Στα προβλήματα ταξινόμησης η έξοδος παίρνει διακριτές τιμές και στόχος είναι η πρόβλεψη της κατηγορίας στην οποία ανήκει η εκάστοτε είσοδος ενώ στα προβλήματα παλινδρόμησης οι ετικέτες (έξοδοι) μπορούν να πάρουν συνεχείς τιμές και στόχος είναι η ακριβέστερη δυνατή προσαρμογή της καμπύλης που τις προσεγγίζει.

2.2.2.2 Μη επιβλεπόμενη μάθηση

Η μη επιβλεπόμενη μάθηση, περιλαμβάνει τα προβλήματα μηχανικής μάθησης στα οποία τα δεδομένα που έχουμε στη διάθεση μας δεν περιλαμβάνουν ετικέτες (εξόδους) [41]. Σε αντίθεση με την επιβλεπόμενη μάθηση όπου η έξοδος που θέλουμε να προβλέψουμε είναι γνωστή για τα δεδομένα που έχουμε, σε αυτού του τύπου τα προβλήματα δεν υπάρχει σαφής στόχος με την έννοια ότι δεν απαιτείται η πρόβλεψη κάποιας συγκεκριμένης τιμής. Συνήθως, τα μοντέλα μη επιβλεπόμενης μάθησης καλούνται να προσαρμοστούν στις στατιστικές ιδιότητες των δεδομένων εκπαίδευσης και να εντοπίσουν πιθανά μοτίβα [39]. Με αυτόν το τον τρόπο σχηματίζουν εσωτερικές αναπαραστάσεις με τις οποίες κωδικοποιούν τα δεδομένα και εντοπίζουν κοινά χαρακτηριστικά και ιδιότητες.

Ένα πλεονέκτημα των αλγορίθμων μη επιβλεπόμενης μάθησης είναι ότι απαιτείται μικρότερος βαθμός προεπεξεργασίας των δεδομένων σε σύγκριση με τους αλγορίθμους επιβλεπόμενης μάθησης. Αυτό οφείλεται εν μέρει και στο γεγονός ότι η σύνθεση των ετικετών απαιτεί ανθρώπινη παρέμβαση καθώς τις περισσότερες φορές δεν είναι δυνατή η αυτοματοποίηση της συγκεκριμένης διαδικασίας. Ωστόσο, στη μη επιβλεπόμενη μάθηση χρειάζεται μεγαλύτερος όγκος δεδομένων κατά την εκπαίδευση προκειμένου να φτάσει ένα μοντέλο σε ικανοποιητικά επίπεδα απόδοσης. Επιπλέον, υπάρχουν μεγαλύτερες απαιτήσεις σε αποθηκευτικούς και υπολογιστικούς πόρους καθώς και μεγαλύτερη ευαισθησία σε ανωμαλίες των δεδομένων (που στην περίπτωση της επιβλεπόμενης μάθησης θα μπορούσαν να αναγνωριστούν εύκολα και να υποστούν την κατάλληλη επεξεργασία πριν τη φάση της εκπαίδευσης).

Η πιο συνηθισμένη εφαρμογή μη επιβλεπόμενης μάθησης είναι η συσταδοποίηση, δηλαδή η διαδικασία διαχωρισμού των δεδομένων σε ομάδες. Εν αντιθέσει, όμως, με τα προβλήματα ταξινόμησης (που ανήκουν στην κατηγορία της επιβλεπόμενης μάθησης) δεν υπάρχουν προκαθορισμένες κλάσεις (ομάδες) στις οποίες πρέπει να ταξινομηθούν τα δεδομένα, αλλά αυτές καθορίζονται δυναμικά από τον

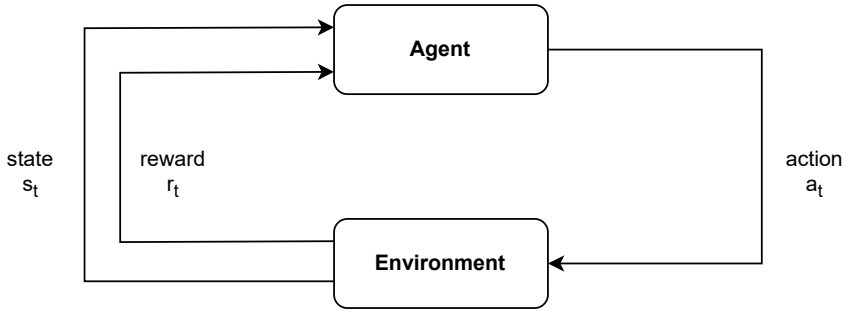
αλγόριθμο με βάση ομοιότητες που εντοπίζονται από τις αναπαραστάσεις των δεδομένων.

Επιπροσθέτως, αλγόριθμοι μη επιβλεπόμενης μάθησης χρησιμοποιούνται κατά κόρον για τη *μείωση της διαστατικότητας* (dimensionality reduction) των δεδομένων. Σε πολλές περιπτώσεις το πλήθος των χαρακτηριστικών των δειγμάτων είναι υπερβολικά μεγάλο, καθιστώντας ιδιαίτερα χρονοβόρα και αναποτελεσματική την εκπαίδευση των μοντέλων μηχανικής μάθησης. Μέσω της μη επιβλεπόμενης μάθησης μπορούν να εντοπιστούν τα χαρακτηριστικά (ή συνδυασμός αυτών) που περιέχουν τη "σημαντικότερη" πληροφορία και να χρησιμοποιηθούν μόνο αυτά κατά την εκπαίδευση. Με αυτόν τον τρόπο ελαττώνεται η διαστατικότητα (και κατά συνέπεια ο χρόνος εκπαίδευσης) με το ελάχιστο δυνατό κόστος καθώς απορρίπτονται τα χαρακτηριστικά με τη μικρότερη συνεισφορά στην εξαγωγή συμπερασμάτων. Τέτοιου είδους εφαρμογές υλοποιούνται συχνά στο στάδιο της προεπεξεργασίας των δεδομένων, προτού αυτά χρησιμοποιηθούν από αλγορίθμους επιβλεπόμενης ή μη επιβλεπόμενης μάθησης κατά την εκπαίδευση.

2.2.2.3 Ενισχυτική Μάθηση

Η ενισχυτική μάθηση είναι η μελέτη αλγορίθμων μηχανικής μάθησης που έχουν στόχο να παίρνουν αποφάσεις με τέτοιο τρόπο ώστε να μεγιστοποιούν το μακροπρόθεσμο κέρδος, ανάλογα με την τρέχουσα κατάσταση σε ένα περιβάλλον [68]. Σε αντίθεση με τα προβλήματα επιβλεπόμενης και μη επιβλεπόμενης μάθησης δεν υπάρχει διαθέσιμο σύνολο δεδομένων για εκπαίδευση αλλά η εκπαίδευση βασίζεται στην αλληλεπίδραση ενός ευφυούς πράκτορα με το περιβάλλον του. Πιο συγκεκριμένα, ένα πρόβλημα ενισχυτικής μάθησης αποτελείται από τα εξής δομικά στοιχεία:

- Ένα περιβάλλον το οποίο περιλαμβάνει ένα σύνολο από δυνατές καταστάσεις στις οποίες μπορεί να βρεθεί.
- Ένα σύνολο ενεργειών οι οποίες προκαλούν μεταβάσεις μεταξύ των καταστάσεων.
- Έναν ευφυή πράκτορα που αποφασίζει και εκτελεί ενέργειες ανάλογα με την τρέχουσα κατάσταση.
- Ένα μοντέλο μετάβασης που καθορίζει την επόμενη κατάσταση με βάση την τρέχουσα κατάσταση και την ενέργεια που εκτελείται.
- Ένα μοντέλο ανταμοιβών που καθορίζει την άμεση ανταμοιβή που προκύπτει από την εκτέλεση μίας ενέργειας σε μια συγκεκριμένη κατάσταση και την αντίστοιχη μετάβαση.



Σχήμα 2.7: Μοντελοποίηση προβλήματος ενισχυτικής μάθησης

Κατά τη διαδικασία της εκπαίδευσης ο πράκτορας, ξεκινώντας με μία τυχαία στρατηγική και αλληλεπιδρώντας σταδιακά με το περιβάλλον, καλείται να "μάθει" μια βέλτιστη στρατηγική, δηλαδή μία αντιστοίχιση καταστάσεων-ενεργειών η οποία θα οδηγεί στη μέγιστη συνολική ανταμοιβή. Μία από τις βασικότερες προκλήσεις σε τέτοιου τύπου προβλήματα είναι η εξισορρόπηση μεταξύ *εξερεύνησης-εκμετάλλευσης* (exploration-exploitation dilemma), δηλαδή η μέθοδος με την οποία ο πράκτορας αποφασίζει αν πρέπει να επιλέξει μία ενέργεια με υψηλή μέχρι στιγμής αναμενόμενη ανταμοιβή ή μία ενέργεια για την οποία δεν έχει συλλέξει ακόμα αρκετή πληροφορία (και ενδεχομένως να αποδειχθεί πιο αποδοτική). Αναλυτική παριγραφή των κυριότερων μεθόδων εκπαίδευσης ενός πράκτορα σε προβλήματα ενισχυτικής μάθησης γίνεται στην Ενότητα 2.2.4.

Η ενισχυτική μάθηση είναι κατάλληλη για προβλήματα στα οποία υπάρχει η δυνατότητα προσομοίωσης ενός περιβάλλοντος (και συλλογής πληροφορίας για τις καταστάσεις, η οποία μεταφράζεται σε ανταμοιβές ή ποινές) αλλά δεν υπάρχει καθαυτή γνώση για τον τρόπο λειτουργίας του. Συνεπώς ο μόνος τρόπος εκμάθησης είναι μέσω αλληλεπίδρασης με αυτό και συλλογής δειγμάτων κατάστασης-ενέργειας-ανταμοιβής. Ένα πεδίο στο οποίο βρίσκει ευρεία εφαρμογή είναι αυτό της αυτόνομης οδήγησης. Συγκεκριμένα δεδομένου ενός περιβάλλοντος προσομοίωσης, είναι δυνατή η εκπαίδευση πρακτόρων ελέγχου οχημάτων οι οποίοι μέσω των ανταμοιβών που λαμβάνουν (π.χ. για την αποφυγή εμποδίων, την ασφαλή μετάβαση στον προορισμό κ.α.) μπορούν να αναπτύξουν την κατάλληλη στρατηγική οδήγησης χωρίς να έχουν προγραμματιστεί ρητά για κάθε πιθανή περίπτωση. Σε αυτή την περίπτωση δεν ορίζεται εξ' αρχής ο τρόπος λειτουργίας του αυτό-οδηγούμενου οχήματος, πρέπει ωστόσο να σχεδιαστεί το μοντέλο ανταμοιβών προκειμένου να μπορέσει ο πράκτορας να αντιστοιχίσει αποτελεσματικά τη βέλτιστη ενέργεια στην εκάστοτε είσοδο που λαμβάνει. Άλλες εφαρμογές της ενισχυτικής μάθησης αφορούν κατά κύριο λόγο προβλήματα *σχεδιασμού* (planning), ρομποτικής καθώς και τον κλάδο των παιχνιδιών.

2.2.3 Τεχνητά Νευρωνικά Δίκτυα

Τα *Τεχνητά Νευρωνικά Δίκτυα* (ΤΝΔ) (Artificial Neural Networks - ANNs) είναι υπολογιστικά μοντέλα εμπνευσμένα από τα βιολογικά νευρωνικά δίκτυα και συγκεκριμένα από τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου [8]. Ουσιαστικά πρόκειται για μια σειρά αλγορίθμων που έχουν στόχο να αναγνωρίσουν συσχετίσεις μεταξύ των δεδομένων και να προσαρμόζονται σε αυτά, μιμούμενοι βιολογικά μοντέλα. Αποτελούν την πιο διαδεδομένη τεχνική στον κλάδο της μηχανικής μάθησης καθώς έχουν αναπτυχθεί αρχιτεκτονικές κατάλληλες για όλους τους τύπους προβλημάτων (όπως περιγράφηκαν παραπάνω) και πλέον πετυχαίνουν υψηλότερες επιδόσεις από τους κλασικούς αλγορίθμους στην πλειοψηφία των περιπτώσεων. Βασικό τους πλεονέκτημα είναι ότι μπορούν να προσεγγίσουν εξαιρετικά μη γραμμικές συναρτήσεις, γεγονός που τα καθιστά ιδιαίτερα αποτελεσματικά σε προβλήματα για τα οποία δεν υπάρχει γνωστή αναλυτική μέθοδος επίλυσης. Όπως σε όλες σχεδόν τις μεθόδους μηχανικής μάθησης υπάρχουν δύο βασικά στάδια, το στάδιο εκπαίδευσης όπου το νευρωνικό δίκτυο τροφοδοτείται επαναληπτικά με δεδομένα και ενημερώνει τα εσωτερικά του βάρη και το στάδιο συμπερασμάτων κατά το οποίο το δίκτυο μπορεί να χρησιμοποιηθεί για να κάνει προβλέψεις σε νέα δεδομένα.

2.2.3.1 Δομικές Μονάδες

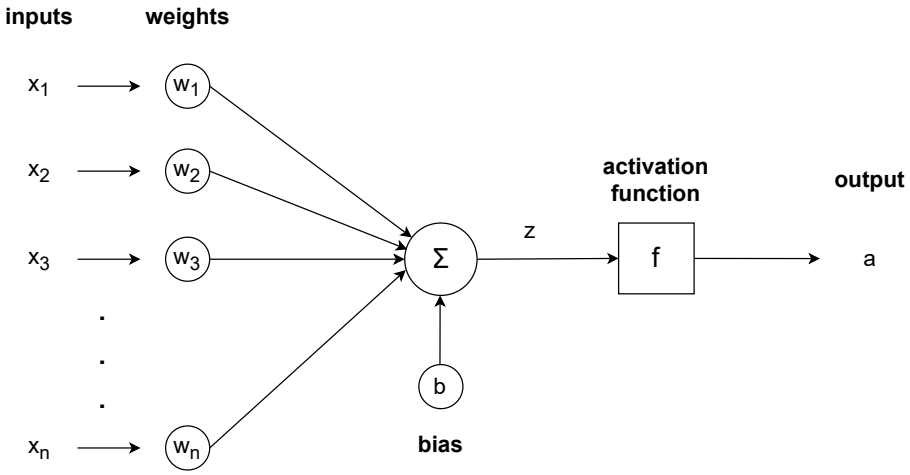
Τεχνητός Νευρώνας (Artificial Neuron) Ένα τεχνητό νευρωνικό δίκτυο αποτελείται από μία συλλογή τεχνητών νευρώνων που συνδέονται μεταξύ τους. Κάθε νευρώνας υλοποιείται ως

ένας κόμβος που επικοινωνεί με τους υπόλοιπους μέσω συνδέσεων οι οποίες είναι εμπνευσμένες από τις βιολογικές συνάψεις. Οι τεχνητοί νευρώνες αποτελούν δομικές μονάδες του δικτύου που δέχονται μία ή περισσότερες εισόδους και παράγουν μία έξοδο την οποία μπορούν να μεταβιβάσουν σε άλλους νευρώνες. Οι είσοδοι μπορεί ανάλογα με την αρχιτεκτονική του δικτύου και τη θέση των νευρώνων να είναι είτε τα χαρακτηριστικά ενός δείγματος από το σύνολο δεδομένων είτε οι έξοδοι άλλων νευρώνων. Όλες οι συνάψεις εισόδου έχουν ένα *βάρος* (weight), δηλαδή μια αριθμητική τιμή που αντικατοπτρίζει τη σημαντικότητα της κάθε εισόδου και σε κάθε νευρώνα ανατίθεται μία επιπλέον τιμή που ονομάζεται *πόλωση* (bias).

Προκειμένου να οριστεί η έξοδος ενός νευρώνα αρχικά υπολογίζεται το εσωτερικό γινόμενο του διανύσματος εισόδου x με το αντίστοιχο διάνυσμα βαρών w. Εν συνεχεία προστίθεται η πόλωση b του νευρώνα και το αποτέλεσμα το οποίο προκύπτει καλείται *ενεργοποίηση* (activation). Έπειτα η ενεργοποίηση δίνεται ως είσοδος σε μία (κατά βάση) μη γραμμική συνάρτηση η οποία καλείται *συνάρτηση ενεργοποίησης* (activation function) και προκύπτει η τελική έξοδος a.

$$a = f(w \cdot x + b) \tag{2.1}$$

Η δομή ενός τυπικού τεχνητού νευρώνα απεικονίζεται διαγραμματικά στο Σχήμα 2.8.



Σχήμα 2.8: Δομή τεχνητού νευρώνα

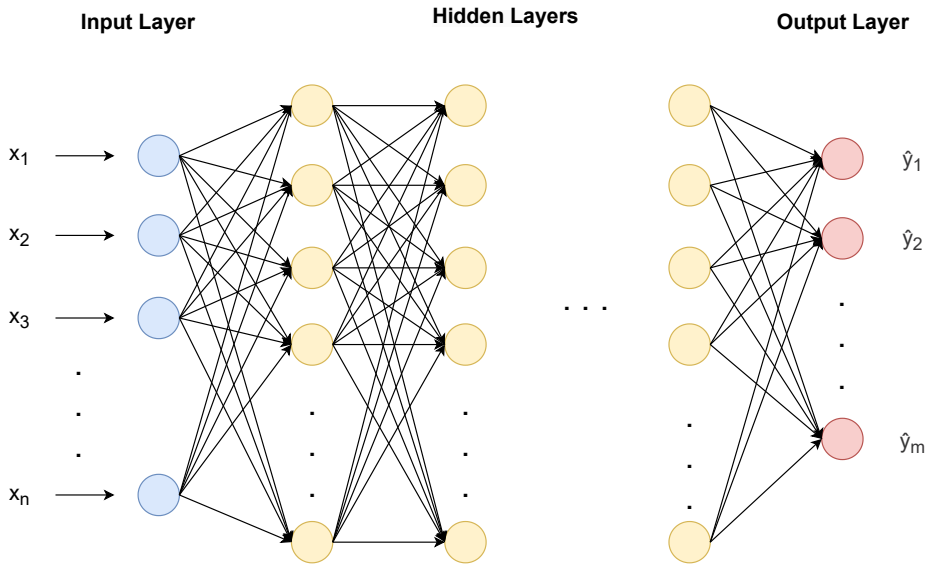
Επίπεδο (Layer) Ο τρόπος με τον οποίο οι νευρώνες οργανώνονται και συνδέονται μεταξύ τους ορίζει την αρχιτεκτονική του δικτύου. Στην πλειοψηφία των αρχιτεκτονικών οι νευρώνες είναι τοποθετημένοι σε επίπεδα, δηλαδή είναι ουσιαστικά χωρισμένοι σε ομάδες και κατά βάση οι νευρώνες ενός επιπέδου δεν επικοινωνούν μεταξύ τους αλλά με τους νευρώνες των υπόλοιπων επιπέδων. Τα επίπεδα ενός νευρωνικού δικτύου μπορούν ανάλογα με τη θέση και το σκοπό που επιτελούν να χωριστούν σε τρεις βασικές κατηγορίες:

- *Επίπεδο εισόδου* (input layer): είναι το πρώτο επίπεδο του δικτύου το οποίο λαμβάνει την είσοδο. Ο αριθμός των νευρώνων του επιπέδου εισόδου είναι ίσος με το πλήθος των χαρακτηριστικών των δειγμάτων καθώς κάθε ένα από αυτά τροφοδοτείται σε έναν ξεχωριστό νευρώνα.
- *Επίπεδο εξόδου* (output layer): είναι το τελευταίο επίπεδο το οποίο παράγει την τελική έξοδο δηλαδή την πρόβλεψη για την αντίστοιχη είσοδο. Και σε αυτή την περίπτωση ο αριθμός των

νευρώνων είναι προκαθορισμένος και ίσος με τις εξόδους που προκύπτουν (π.χ. σε ένα πρόβλημα ταξινόμησης, το επίπεδο εξόδου θα έχει τόσους νευρώνες όσες είναι και οι πιθανές κλάσεις).

- *Κρυφά επίπεδα* (hidden layers): όλα τα ενδιάμεσα επίπεδα μεταξύ εισόδου και εξόδου. Ο αριθμός τους μπορεί να ποικίλλει και συνήθως χρησιμοποιούνται περισσότερα κρυφά επίπεδα ανάλογα με την πολυπλοκότητα του προβλήματος. Το ίδιο ισχύει και για τον αριθμό νευρώνων του κάθε επιπέδου που δεν είναι προκαθορισμένος και αποτελεί σχεδιαστική επιλογή του δικτύου. Το πλήθος των κρυφών επιπέδων και των αντίστοιχων νευρώνων δεν μπορεί να αποφασιστεί εκ των προτέρων παρά μόνο μετά από δοκιμές καθώς η επιλογή τους εξαρτάται σε σημαντικό βαθμό από τη φύση του προβλήματος και τον τύπο των δεδομένων.

Τα νευρωνικά δίκτυα στα οποία οι νευρώνες κάθε επιπέδου μεταβιβάζουν τις εξόδους τους μόνο στους νευρώνες του επόμενου επιπέδου ονομάζονται δίκτυα *πρόσθιας τροφοδότησης* (feed-forward). Σε αυτή την περίπτωση η ροή της πληροφορίας έχει μόνο μία κατεύθυνση (από την είσοδο προς την έξοδο) σχηματίζοντας έναν *κατευθυνόμενο ακυκλικό γράφο* (directed acyclic graph). Συνήθως κάθε νευρώνας συνδέεται με όλους τους νευρώνες του αμέσως επόμενου επιπέδου, συνθέτοντας ένα *πλήρως διασυνδεδεμένο* (fully connected) νευρωνικό δίκτυο. Ωστόσο, ανάλογα με την αρχιτεκτονική είναι δυνατόν να υπάρξουν διαφόρων τύπων παραλλαγές (π.χ η εξόδος ενός νευρώνα να προωθείται μόνο σε ένα υποσύνολο των νευρώνων του επόμενου επιπέδου ή και σε κάποιο από τα επόμενα επίπεδα). Ένα τυπικό πλήρως διασυνδεδεμένο νευρωνικό δίκτυο απεικονίζεται στο Σχήμα 2.9.



Σχήμα 2.9: Τυπική μορφή πλήρως διασυνδεδεμένου τεχνητού νευρωνικού δικτύου πρόσθιας τροφοδότησης

Συνάρτηση Ενεργοποίησης (Activation Function) Οι συναρτήσεις ενεργοποίησης είναι συνήθως μη γραμμικές συναρτήσεις οι οποίες καθορίζουν την τελική έξοδο των νευρώνων. Είναι ιδιαίτερα σημαντικές καθώς συμβάλλουν στην προσέγγιση πολύπλοκων, μη γραμμικών συσχετίσεων μεταξύ εισόδων και εξόδων στα νευρωνικά δίκτυα. Ανάλογα με την αρχιτεκτονική του δικτύου και το επίπεδο στο οποίο βρίσκονται οι νευρώνες ενδεχομένως να ενδείκνυνται διαφορετικές συναρτήσεις ενεργοποίησης· παρόλα αυτά η συνάρτηση ενεργοποίησης αποτελεί παράμετρο του συστήματος και δεν μπορεί να υπάρξει εκ των προτέρων βέλτιστη επιλογή.

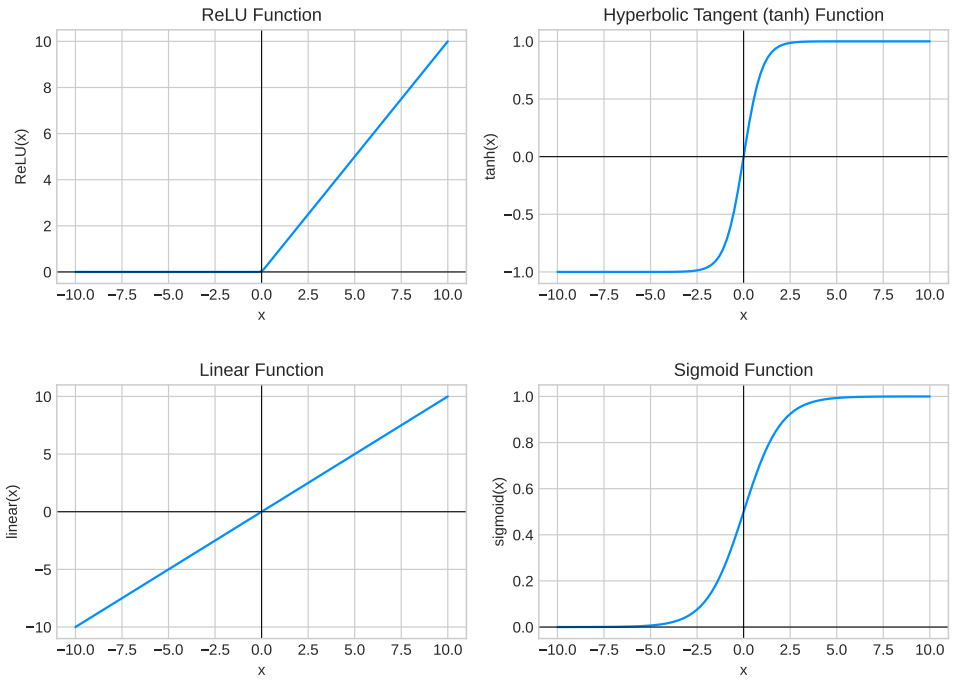
Στη συνέχεια παρουσιάζονται οι συνηθέστερες επιλογές ανάλογα με το επίπεδο του δικτύου στο οποίο εφαρμόζονται. Στα κρυφά επίπεδα οι πιο ευρέως χρησιμοποιούμενες συναρτήσεις ενεργοποίησης είναι οι:

- Διορθωμένη Γραμμική (Rectified Linear Unit or ReLU): $f(x) = max(0, x)$
- Υπερβολική Εφαπτομένη (Hyperbolic Tangent): $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Στο επίπεδο εξόδου, η τιμή μπορεί να χρειάζεται να είναι σε κάποιο συγκεκριμένο εύρος (π.χ. $[0, 1]$ σε περίπτωση ταξινόμησης με δύο κλάσεις, οπότε η έξοδος του δικτύου αναπαριστά ουσιαστικά την πιθανότητα ένα δείγμα να ανήκει σε μία κλάση) ή μπορεί να ανήκει σε όλο το $\mathbb{R}$ (συνήθως σε προβλήματα παλινδρόμησης). Συνήθεις συναρτήσεις ενεργοποίησης στο επίπεδο εξόδου είναι οι:

- Γραμμική (Linear, Identity): $f(x) = x$
- Σιγμοειδής (Sigmoid, Logistic): $f(x) = \frac{1}{1+e^{-x}}$
- Softmax: $f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{C} e^{x_j}}$

Η μορφή των παραπάνω συναρτήσεων φαίνεται γραφικά στο Σχήμα 2.10.



Σχήμα 2.10: Βασικές συναρτήσεις ενεργοποίησης

2.2.3.2 Εκπαίδευση

Κατά τη διαδικασία της εκπαίδευσης, το νευρωνικό δίκτυο τροφοδοτείται επαναληπτικά με δείγματα από το σύνολο δεδομένων εκπαίδευσης και ανανεώνει τα βάρη των νευρώνων με στόχο την ελαχιστοποίηση του σφάλματος μεταξύ των προβλέψεων του μοντέλου και των *πραγματικών τιμών*

(groundtruth). Ουσιαστικά, αναζητώνται οι βέλτιστες τιμές των παραμέτρων W^* (βάρη) και b^* (πολώσεις) του δικτύου οι οποίες οδηγούν στις πλησιέστερες δυνατές προβλέψεις στις πραγματικές τιμές. Όταν αυτό το σφάλμα σταθεροποιείται (πρακτικά αυξομειώνεται με πολύ μικρό ρυθμό οπότε θεωρείται ότι δεν μπορεί να βελτιωθεί περαιτέρω), η εκπαίδευση ολοκληρώνεται και οι τελικές παράμετροι που έχουν υπολογιστεί εκείνη τη στιγμή χρησιμοποιούνται για την πρόβλεψη των νέων τιμών στο στάδιο συμπερασμάτων.

Συνάρτηση Κόστους (Cost Function) Προκειμένου να είναι δυνατή η αξιολόγηση της απόδοσης του δικτύου κατά τη διάρκεια της εκπαίδευσης χρειάζεται ένας τρόπος μέτρησης του σφάλματος των προβλέψεων. Αυτό το ρόλο τον αναλαμβάνει η συνάρτηση κόστους. Η συνάρτηση κόστους δίνει μία εικόνα της αποτελεσματικότητας του μοντέλου σε κάθε βήμα της εκπαίδευσης (δηλαδή για τις αντίστοιχες τιμές των παραμέτρων του δικτύου που χρησιμοποιούνται εκείνη τη στιγμή) και κατευθύνει τον επαναπροσδιορισμό των παραμέτρων με στόχο τη βελτίωση των προβλέψεων σε κάθε επανάληψη.

Ανάλογα με τη φύση του προβλήματος, η μορφή της συνάρτησης κόστους μπορεί να διαφέρει. Σε κάθε περίπτωση, ο στόχος είναι να αποτυπώσει με τον καλύτερο δυνατό τρόπο την απόκλιση μεταξύ των πραγματικών τιμών και των τιμών που προκύπτουν στην έξοδο του δικτύου. Ορισμένες από τις πιο ευρέως χρησιμοποιούμενες συναρτήσεις κόστους για προβλήματα ταξινόμησης και παλινδρόμησης παρουσιάζονται στη συνέχεια. Οι πραγματικές τιμές αναπαρίστανται ως y, οι προβλέψεις του δικτύου ως $\hat{y}$, ο αριθμός των κλάσεων (για προβλήματα ταξινόμησης) ως C και ο αριθμός των δειγμάτων από το σύνολο δεδομένων που χρησιμοποιούνται για τον υπολογισμό του κόστους ως N.

- **Σταυροειδής Εντροπία (Cross-Entropy)**: είναι η πιο συνηθισμένη συνάρτηση κόστους για προβλήματα ταξινόμησης. Ουσιαστικά η ελαχιστοποίηση της σημαίνει την ελαχιστοποίηση της απόκλισης μεταξύ της κατανομής των πραγματικών τιμών y και της κατανομής των προβλέψεων $\hat{y}$. Η μορφή της φαίνεται στην Εξίσωση 2.2.

$$J(y, \hat{y}) = \sum_{i=1}^{N} \sum_{j=1}^{C} -y_i^j \log \hat{y}_i^j \tag{2.2}$$

- **Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error - MSE)**: χρησιμοποιείται σε προβλήματα παλινδρόμησης. Το γεγονός ότι υπολογίζεται το τετράγωνο της διαφοράς, καθιστά τη συνάρτηση πιο "αυστηρή" όσον αφορά τα μεγαλύτερα σφάλματα ώστε να οδηγεί σε μεγαλύτερη τροποποίηση των παραμέτρων. Η μορφή της φαίνεται στην Εξίσωση 2.3.

$$J(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{2.3}$$

- **Μέσο Απόλυτο Σφάλμα (Mean Absolute Error - MAE)**: Χρησιμοποιείται επίσης σε προβλήματα παλινδρόμησης. Ενδείκνυται σε περιπτώσεις όπου το σύνολο δεδομένων περιλαμβάνει πολλές ακραίες τιμές (outliers). Η μορφή της φαίνεται στην Εξίσωση 2.4.

$$J(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{2.4}$$

- **Τετραγωνικό R (R-Squared - R^2)**: Αφορά προβλήματα παλινδρόμησης και εκφράζει το ποσοστό της διακύμανσης της εξαρτημένης μεταβλητής που μπορεί να εξηγηθεί από το μοντέλο. Η μορφή της φαίνεται στην Εξίσωση 2.5.

$$J(y, \hat{y}) = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y}_i)^2} \tag{2.5}$$

Ανάλογα με τις ιδιότητες των δεδομένων και το σκοπό που θέλουμε να πετύχουμε είναι δυνατή και η χρήση συναρτήσεων κόστους ειδικά σχεδιασμένων για το εκάστοτε πρόβλημα. Για παράδειγμα, είναι πιθανό να υπάρχουν συγκεκριμένοι περιορισμοί και να απαιτείται να δοθεί διαφορετική βαρύτητα σε διαφορετικούς τύπους σφάλματος. Σε τέτοιες περιπτώσεις, μία συνάρτηση κόστους σχεδιασμένη αποκλειστικά για το συγκεκριμένο πρόβλημα και σύνολο δεδομένων θα μπορούσε να οδηγήσει στην επιθυμητή συμπεριφορά του δικτύου με μεγαλύτερη αποτελεσματικότητα από ότι αν χρησιμοποιούταν κάποια από τις κλασικές συναρτήσεις κόστους για την εκπαίδευσή του.

Στοχαστική Κάθοδος Κλίσης (Stochastic Gradient Descent - SGD) Εφόσον έχει οριστεί η συνάρτηση κόστους που θα χρησιμοποιηθεί για να κατευθύνει την εκπαίδευση, χρειάζεται ένας αλγόριθμος προκειμένου να προσδιοριστούν οι βέλτιστες παράμετροι του δικτύου, δηλαδή οι παράμετροι με τις οποίες ελαχιστοποιείται η τιμή της συνάρτησης κόστους για το σύνολο δεδομένων. Ένας τέτοιος αλγόριθμος καλείται αλγόριθμος βελτιστοποίησης. Ένας από τους πιο διαδεδομένους αλγορίθμους βελτιστοποίησης είναι η *κάθοδος κλίσης* (gradient descent). Η κάθοδος κλίσης είναι ένας επαναληπτικός αλγόριθμος κατά την εκτέλεση του οποίου ανανεώνονται σταδιακά τα βάρη του δικτύου κάνοντας χρήση της παραγώγου της συνάρτησης κόστους.

Έστω θ το σύνολο των παραμέτρων του δικτύου, δηλαδή τα βάρη W και οι πολώσεις b των νευρώνων. Αναζητώνται οι βέλτιστες παράμετροι θ^* για τις οποίες:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\ J(y, \hat{y}; \theta) \tag{2.6}$$

Ξεκινώντας από ένα τυχαίο σύνολο παραμέτρων θ, ο αλγόριθμος υπολογίζει για κάθε παράμετρο του δικτύου τη μερική παράγωγο της συνάρτησης κόστους ως προς αυτή την παράμετρο και στη συνέχεια ανανεώνει την τιμή της με βάση την παράγωγο. Η βασική αρχή είναι ότι η παράγωγος σε ένα σημείο υποδηλώνει τη μονοτονία της συνάρτησης και επομένως μπορεί να χρησιμοποιηθεί για τη μεταβολή των παραμέτρων προς την κατεύθυνση που μειώνεται η τιμή της συνάρτησης κόστους. Αυτή η μεταβολή για μία παράμετρο w^i φαίνεται στην Εξίσωση 2.7 και σε διανυσματική μορφή στην Εξίσωση 2.8 (το n υποδηλώνει τη n-οστή επανάληψη της διαδικασίας).

$$w^i_{n+1} \leftarrow w^i_n - \eta \cdot \frac{\partial J(y, \hat{y}; \theta)}{\partial w^i} \tag{2.7}$$

$$\theta_{n+1} \leftarrow \theta_n - \eta \cdot \nabla_\theta J(y, \hat{y}; \theta) \tag{2.8}$$

Η παράμετρος η καλείται *ρυθμός μάθησης* (learning rate) και καθορίζει το ρυθμό ανανέωσης των βαρών σε κάθε επανάληψη. Είναι ιδιαίτερα σημαντική για την εκπαίδευση καθώς πολύ μικρές τιμές μπορεί να καθυστερήσουν σημαντικά την εκπαίδευση του δικτύου ενώ πολύ μεγάλες ενδέχεται να την αποσταθεροποιήσουν.

Για τον υπολογισμό της μερικής παραγώγου ως προς παραμέτρους που βρίσκονται πιο πίσω στην αρχιτεκτονική του δικτύου (δηλαδή πιο μακριά από το επίπεδο εξόδου) χρησιμοποιείται ο *κανόνας της αλυσίδας* (chain rule). Στην Εξίσωση 2.9 φαίνεται ο υπολογισμός της μερικής παραγώγου της συνάρτησης κόστους J ως προς την παράμετρο w^i που βρίσκεται στο k-οστό επίπεδο ενός δικτύου με L συνολικά επίπεδα. Με z αναπαριστάται η έξοδος του νευρώνα και με a η έξοδος που προκύπτει από τη συνάρτηση ενεργοποίησης.

$$\frac{\partial J(y, \hat{y}; \theta)}{\partial w^{i[k]}} = \frac{\partial J(y, \hat{y}; \theta)}{\partial a^{[L]}} \cdot \frac{\partial a^{[L]}}{\partial z^{[L]}} \cdot \frac{\partial z^{[L]}}{\partial a^{[L-1]}} \cdot ... \cdot \frac{\partial z^{[k]}}{\partial w^{i[k]}} \tag{2.9}$$

Όπως προκύπτει από την παραπάνω εξίσωση, ορισμένες παράγωγοι είναι απαραίτητες για αρκετούς υπολογισμούς που αφορούν παραμέτρους που βρίσκονται σε πιο πίσω επίπεδα. Για τη μεγαλύτερη δυνατή αποδοτικότητα, προκειμένου να μην υπολογίζονται οι παράγωγοι κάθε φορά εκ νέου, οι παράμετροι του δικτύου ανανεώνονται ξεκινώντας από το τελευταίο επίπεδο προς τα πίσω. Με αυτό τον τρόπο, όλοι οι υπολογισμοί που αφορούν προηγούμενα επίπεδα έχουν ήδη γίνει σε προηγούμενα βήματα (που αφορούν τα επίπεδα πιο κοντά στην έξοδο του δικτύου) και μπορούν να επαναχρησιμοποιηθούν χωρίς επιπλέον υπολογιστικό κόστος. Η συγκεκριμένη μέθοδος καλείται *οπισθοδιάδοση σφάλματος* (error backpropagation).

Άλλο ένα υπολογιστικό πρόβλημα που προκύπτει στην διαδικασία της εκπαίδευσης αφορά το συνολικό σφάλμα του δικτύου σε κάθε επανάληψη. Στην πράξη το σύνολο δεδομένων μπορεί να περιλαμβάνει έναν πολύ μεγάλο αριθμό δειγμάτων. Ως εκ τούτου, η ανανέωση των βαρών με βάση το σφάλμα όλων των δειγμάτων είναι ανέφικτη. Για να αντιμετωπιστεί αυτό το πρόβλημα χρησιμοποιείται μία παραλλαγή του αλγορίθμου καθόδου κλίσης, η *Στοχαστική Κάθοδος Κλίσης* (Stochastic Gradient Descent - SGD), κατά την οποία η ανανέωση των βαρών γίνεται σε *παρτίδες* (batches). Σε αυτή την περίπτωση, το σύνολο δεδομένων χωρίζεται σε ίσα τμήματα (παρτίδες) και ο υπολογισμός του σφάλματος γίνεται σε κάθε παρτίδα ξεχωριστά. Το πλήθος των δειγμάτων κάθε παρτίδας ονομάζεται *μέγεθος παρτίδας* (batch size) και αποτελεί υπερπαράμετρο του συστήματος. Στην κλασική μέθοδο, κάθε παρτίδα χρησιμοποιείται μία φορά προκειμένου να εξασφαλιστεί ότι όλα τα δεδομένα συνεισφέρουν στον ίδιο βαθμό. Μία πλήρης επανάληψη κατά την οποία όλες οι παρτίδες έχουν χρησιμοποιηθεί μία φορά καλείται *εποχή* (epoch). Προκειμένου το δίκτυο να εκπαιδευτεί επαρκώς (σε σημείο που δεν παρουσιάζει πλέον βελτίωση) συνήθως απαιτούνται αρκετές εποχές.

2.2.3.3 Πιθανοτικά Μοντέλα Διάχυσης Αποθορύφωσης

Τα *Πιθανοτικά Μοντέλα Διάχυσης Αποθορύφωσης* (Denoising Diffusion Probabilistic Models - DDPMs) [44] είναι μία κατηγορία γεννητικών μοντέλων που χρησιμοποιούνται για την παραγωγή νέων δειγμάτων μαθαίνοντας την κατανομή δεδομένων εκπαίδευσης. Η βασική ιδέα της λειτουργίας τους είναι η παραποίηση των δεδομένων με σταδιακή προσθήκη θορύβου στα δείγματα και στη συνέχεια η εκπαίδευση του μοντέλου κατά τέτοιο τρόπο ώστε να προβλέπει τη συνάρτηση πυκνότητας πιθανότητας του θορύβου προκειμένου να τον αφαιρεί αποτελεσματικά και να αναπαράγει τα αρχικά δεδομένα. Η διαδικασία εκαπίδευσης του δικτύου χωρίζεται σε δύο φάσεις. Αρχικά, στην *πρόσθια διαδικασία διάχυσης* (forward diffusion process), θόρυβος ο οποίος δειγματοληπτείται από μια κατανομή Gauss διαχέεται επαναληπτικά στα δεδομένα μέχρι να αντιστοιχιστούν πλήρως σε μία νέα, πιο απλή κατανομή. Στη συνέχεια, ένα νευρωνικό δίκτυο μοντελοποιεί την *αντίστροφη διαδικασία* (reverse diffusion process) με την οποία μπορεί να παράγει νέα δείγματα, ξεκινώντας από τυχαίο θόρυβο και αντιστοιχίζοντάς τον βήμα-βήμα στην κατανομή των δεδομένων εκπαίδευσης.

Τυπικά, έστω $q(x)$ η κατανομή των δεδομένων εκπαίδευσης και x_0 ένα δείγμα της κατανομής ($x_0 \sim q(x)$). Η κατανομή της πρόσθιας διαδικασίας διάχυσης ορίζεται στις Εξισώσεις 2.10 και 2.11. Κατά τη διαδικασία της διάχυσης, σε κάθε βήμα t, μια νέα λανθάνουσα μεταβλητή x_t παράγεται προσθέτοντας γκαουσιανό (gaussian) θόρυβο διακύμανσης b_t στην προηγούμενη μεταβλητή x_{t-1}, όπως περιγράφεται στην Εξίσωση 2.12, όπου $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \mu_t = x_{t-1}\sqrt{1-\beta_t}, \Sigma_t = \beta_t \mathbf{I}) \tag{2.10}$$

$$q(x_{1:T}) = q(x_1|x_0)q(x_2|x_1)\dots q(x_T|x_{T-1}) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{2.11}$$

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_t \tag{2.12}$$

Θέτοντας $a_t = 1 - \beta_t$, $\bar{a}_t = \prod_{i=1}^{t} a_i$ και χρησιμοποιώντας το τέχνασμα επαναπαραμετροποίησης [52], οποιοδήποτε δείγμα x_t μπορεί να δειγματοληφθεί απευθείας, χωρίς να χρειάζεται να υπολογιστούν όλες οι ενδιάμεσες λανθάνουσες μεταβλητές. Δεδομένου του αρχικού δείγματος x_0, η κατανομή q παίρνει τη μορφή της Εξίσωσης 2.13 και η λανθάνουσα μεταβλητή μετά από T χρονικά βήματα μπορεί να υπολογιστεί σύμφωνα με την Εξίσωση 2.14.

$$q(x_t|x_0) = \mathcal{N}(x_t; \mu_t = x_0\sqrt{\bar{a}_t}, \Sigma_t = (1-\bar{a}_t)\mathbf{I}) \tag{2.13}$$

$$x_t = \sqrt{\bar{a}_t}x_0 + \sqrt{1-\bar{a}_t}\epsilon_0 \tag{2.14}$$

Για την αντίστροφη διαδικασία, χρησιμοποιείται ένα νευρωνικό δίκτυο για τη μοντελοποίηση της κατανομής p (δηλαδή των παραμέτρων μίας άλλης γκαουσιανής κατανομής) που αφαιρεί προοδευτικά τον θόρυβο και αντιστοιχίζει οποιοδήποτε τυχαίο δείγμα στην αρχική κατανομή. Στόχος είναι η ελαχιστοποίηση της απόστασης (απόκλιση Kullback–Leibler) μεταξύ της κατανομής p_θ και της πρόσθιας *δεσμευμένης* (posterior) κατανομής $q(x_{t-1}|x_t, x_0)$, που ορίζεται στην Εξίσωση 2.15.

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t \mathbf{I}) \tag{2.15}$$

όπου $\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{a}_{t-1}}\beta_t}{1-\bar{a}_t}x_0 + \frac{\sqrt{a_t}(1-\bar{a}_{t-1})}{1-\bar{a}_t}x_t$ και $\tilde{\beta}_t = \frac{1-\bar{a}_{t-1}}{1-\bar{a}_t}\beta_t$.

Θέτοντας τη διακύμανση ίση με μια καθορισμένη σταθερά β_t και χρησιμοποιώντας το τέχνασμα επαναπαραμετροποίησης για να εκφράσουμε το x_0 συναρτήσει του x_t, το $\tilde{\mu}_t$ ορίζεται ως εξής:

$$\tilde{\mu}_t(x_t) = \frac{1}{\sqrt{\bar{x}_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{a}_t}})\epsilon_t \tag{2.16}$$

Συνεπώς, το νευρωνικό δίκτυο αρκεί να προβλέψει τον θόρυβο $\epsilon_{\theta(x_t,t)}$ σε κάθε χρονικό βήμα, προκειμένου να ελαχιστοποιήσει την ακόλουθη συνάρτηση κόστους:

$$L(\theta) = \mathbb{E}_{t,x_0,\epsilon}[\|\epsilon - \epsilon_\theta(x_t, t)\|^2] \tag{2.17}$$

Όσον αφορά την αρχιτεκτονική του μοντέλου, παρότι ο μόνος περιορισμός είναι η είσοδος και η έξοδος του δικτύου να έχουν την ίδια διαστατικότητα, η πιο κοινή προσέγγιση είναι ένα νευρωνικό δίκτυο βασισμένο στην αρχιτεκτονική U-net [83] με *υπολειμματικά* (residual) [40] μπλοκ και μπλοκ

αυτοπροσοχής (self-attention) [99]. Το μοντέλο δέχεται ως είσοδο το χρονικό βήμα και το αντίστοιχο δείγμα (εικόνα) και προβλέπει τον θόρυβο σε αυτό το σημείο. Με αυτόν τον τρόπο, μετά την ολοκλήρωση της εκπαίδευσης, κάθε δείγμα τυχαίου θορύβου (των ίδιων διαστάσεων με τα αρχικά δεδομένα) μπορεί να τροφοδοτηθεί στο μοντέλο και να αντιστοιχιστεί βήμα προς βήμα στην κατανομή των δεδομένων εκπαίδευσης, παράγοντας ένα εντελώς νέο δείγμα.

2.2.4 Ενισχυτική Μάθηση

2.2.4.1 Ορισμός και Μοντελοποίηση

Ένα πρόβλημα ενισχυτικής μάθησης όπου, σε μια ακολουθία διακριτών χρονικών βημάτων t, ένας ευφυής πράκτορας λαμβάνει αποφάσεις με βάση την αλληλεπίδρασή του με το περιβάλλον, μπορεί να περιγραφεί ως μία *Μαρκοβιανή Διαδικασία Αποφάσεων* (ΜΔΑ) (Markov Decision Process - MDP) [68]. Τυπικά, μία ΜΔΑ περιγράφεται με μία πλειάδα $\langle S, A, P, R, \gamma \rangle$ όπου [68] :

- **S**: είναι ο χώρος καταστάσεων, δηλαδή το σύνολο των δυνατών καταστάσεων
- **A**: είναι ο χώρος ενεργειών, δηλαδή το σύνολο των δυνατών ενεργειών
- $\mathbf{P_a(s, s')} : S \times A \times S \to [0, 1]$: είναι η συνάρτηση μετάβασης η οποία καθορίζει, δεδομένης της τρέχουσας κατάστασης s και μίας ενέργειας a, την πιθανότητα η επόμενη κατάσταση να είναι η s'. Συνεπώς, $P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$
- $\mathbf{R_a(s)} : S \times A \to \mathbb{R}$: είναι η συνάρτηση ανταμοιβής η οποία καθορίζει την άμεση ανταμοιβή που λαμβάνεται μετά την εκτέλεση της ενέργειας a στην κατάσταση s
- $\boldsymbol{\gamma} \in [0, 1]$: είναι ένας *συντελεστής μείωσης* (discount factor) που χρησιμοποιείται για τον υπολογισμό της αναμενόμενης επιστροφής $\mathbb{E}[G_t]$

Σε αυτό το πλαίσιο, ικανοποιείται η μαρκοβιανή ιδιότητα καθώς η μετάβαση από μια κατάσταση s σε μια κατάσταση s' και η άμεση ανταμοιβή r μετά την εκτέλεση μιας ενέργειας a, εξαρτώνται αποκλειστικά από αυτήν την ενέργεια και την κατάσταση s (Εξίσωση 2.18). Από αυτή την άποψη, η αναπαράσταση της κατάστασης σε κάθε χρονικό βήμα t περιγράφει πλήρως τις ιδιότητες της κατάστασης και είναι ανεξάρτητη από τη διαδρομή που οδήγησε σε αυτήν.

$$P(s_{t+1} = s', r_{t+1} = r | s_t, a_t) = P(s_{t+1} = s', r_{t+1} = r | s_t, a_t, r_t, s_{t-1}, \ldots, r_1, s_0, a_0) \tag{2.18}$$

Σε κάθε διακριτό χρονικό βήμα t, ο πράκτορας επιλέγει μια ενέργεια $a_t \in A$, ενώ βρίσκεται σε μία κατάσταση $s_t \in S$ και μεταβαίνει σε μια νέα κατάσταση $s_{t+1} \in S$ σύμφωνα με τη δυναμική του περιβάλλοντος, όπως προσδιορίζεται από τη συνάρτηση μετάβασης P. Κατά τη διάρκεια αυτής της μετάβασης, ο πράκτορας λαμβάνει μια άμεση ανταμοιβή $r_t \in R$ σύμφωνα με τις ιδιότητες του συγκεκριμένου περιβάλλοντος. Για την επιλογή των ενεργειών ο πράκτορας ακολουθεί μία *πολιτική* (policy) $\pi : A \times S \to [0, 1]$, δηλαδή μία αντιστοίχιση των καταστάσεων σε πιθανότητες επιλογής ενεργειών. Ουσιαστικά για κάθε κατάσταση $s \in S$ η πολιτική π καθορίζει την κατανομή πιθανοτήτων για το σύνολο των ενεργειών $a \in A$. Ο στόχος του πράκτορα είναι να καθορίσει μια πολιτική που να μεγιστοποιεί την αναμενόμενη τιμή επιστροφής $\mathbb{E}[G_t]$.

Η επιστροφή στην πιο απλή περίπτωση ορίζεται ως το άθροισμα όλων των ανταμοιβών που συλλέγει ο πράκτορας από το χρονικό βήμα t και έπειτα. Στην Εξίσωση 2.19 φαίνεται η επιστροφή για περιπτώσεις με πεπερασμένο αριθμό βημάτων όπου το τελευταίο βήμα ορίζεται ως T. Σε τέτοιου τύπου προβλήματα μία ακολουθία μεταβάσεων μέχρι μία τελική κατάσταση καλείται επεισόδιο (episode).

$$G_t = r_{t+1} + r_{t+2} + ... + r_T \tag{2.19}$$

Στη γενική περίπτωση, το πρόβλημα μπορεί να μην ολοκληρώνεται μετά από συγκεκριμένο αριθμό βημάτων. Σε τέτοιες εφαρμογές, η επιστροφή όπως προκύπτει από την Εξίσωση 2.19 αυξάνεται συνεχώς τείνοντας στο άπειρο με αποτέλεσμα ο στόχος της μεγιστοποίησής της να μην έχει νόημα. Προκειμένου να είναι εφικτή η μοντελοποίηση του προβλήματος εισάγεται η έννοια του συντελεστή μείωσης γ. Ο ρόλος του είναι να μειώνεται σταδιακά η επίδραση των μελλοντικών ανταμοιβών αλλά να εξακολουθούν να λαμβάνονται υπόψη στην αναμενόμενη ανταμοιβή. Το γ παίρνει τιμές στο εύρος $[0, 1]$. Στην ακραία περίπτωση στην οποία $\gamma = 0$ λαμβάνεται υπόψη μόνο η άμεση ανταμοιβή, ενώ στην περίπτωση που $\gamma = 1$ όλες οι ανταμοιβές έχουν την ίδια επίδραση στη συνολική τιμή, ανεξάρτητα από το πότε λαμβάνονται. Υπό αυτές τις προυποθέσεις, ο τελικός στόχος είναι η μεγιστοποίηση της *μειωμένης επιστροφής* (discounted return):

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} ... = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{2.20}$$

2.2.4.2 Συναρτήσεις Αξίας - Εξίσωση Bellman

Προκειμένου να προσδιοριστεί η βέλτιστη πολιτική του πράκτορα, δηλαδή ο τρόπος με τον οποίο θα γίνεται η επιλογή των ενεργειών, συχνά χρειάζεται μία μέθοδος αξιολόγησης των διαφορετικών καταστάσεων (ή/και ενεργειών) δεδομένης μίας πολιτικής. Για το σκοπό αυτο χρησιμοποιούνται κατά βάση δύο *συναρτήσεις αξίας* (value functions)· η *συνάρτηση αξίας κατάστασης* (state-value function) και η *συνάρτηση αξίας ενέργειας* (action-value function).

Η συνάρτηση αξίας κατάστασης $V_\pi(s)$ ορίζεται ως η αναμενόμενη επιστροφή που θα λάβει ο πράκτορας ξεκινώντας από την κατάσταση s και ακολουθώντας στη συνέχεια την πολιτική π και υποδηλώνει ουσιαστικά πόσο "καλό" είναι να βρίσκεται σε μία κατάσταση δεδομένης μίας συγκεκριμένης πολιτικής.

$$V_\pi(s_t) = \mathbb{E}_\pi[G_t|s_t] = \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t\Big] \tag{2.21}$$

Η συνάρτηση αξίας ενέργειας $Q_\pi(s, a)$ ορίζεται ως η αναμενόμενη επιστροφή που θα λάβει ο πράκτορας ξεκινώντας από την κατάσταση s, εκτελώντας την ενέργεια a και ακολουθώντας στη συνέχεια την πολιτική π. Η συνάρτηση Q δίνει ένα μέτρο αξιολόγησης συγκεκριμένων ενεργειών για μία κατάσταση, δεδομένου ότι ο πράκτορας δρα από το επόμενο χρονικό βήμα και έπειτα βάσει μίας συγκεκριμένης πολιτικής.

$$Q_\pi(s_t, a_t) = \mathbb{E}_\pi[G_t|s_t, a_t] = \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t, a_t\Big] \tag{2.22}$$

Η συνάρτηση αξίας οποιασδήποτε κατάστασης s μπορεί να εκφραστεί σε σχέση με την επόμενη κατάσταση μέσω της εξίσωσης Bellman (Εξίσωση 2.23).

$$\begin{aligned} V_\pi(s_t) &= \mathbb{E}_\pi[G_t|s_t] \\ &= \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t\Big] \\ &= \mathbb{E}_\pi\Big[r_{t+1}+\sum_{k=0}^{\infty}\gamma^{k+1} r_{t+k+2}|s_t\Big] \\ &= \mathbb{E}_\pi\Big[r_{t+1}+\gamma\sum_{k=0}^{\infty}\gamma^{k} r_{t+k+2}|s_t\Big] \\ &= \mathbb{E}_\pi[r_{t+1}+\gamma G_{t+1}] = \mathbb{E}_\pi[r_{t+1}+\gamma\mathbb{E}_\pi[G_{t+1}|s_{t+1}]] \\ &= \sum_{a_t}\pi(a_t|s_t)\sum_{s_{t+1}}\sum_{r_{t+1}}p(s_{t+1},r_{t+1}|s_t,a_t)[r_{t+1}+\gamma V_\pi(s_{t+1})] \end{aligned} \tag{2.23}$$

Ομοίως για τη συνάρτηση αξίας ενέργειας προκύπτει ότι:

$$Q_\pi(s_t,a_t)=\sum_{s_{t+1}}\sum_{r_{t+1}}p(s_{t+1},r_{t+1}|s_t,a_t)\Big[r_{t+1}+\gamma\sum_{a_{t+1}}\pi(a_{t+1}|s_{t+1})Q_\pi(s_{t+1},a_{t+1})\Big] \tag{2.24}$$

Με αυτό τον τρόπο προκύπτει μία αναδρομική σχέση η οποία αποτελεί βασικό στοιχείο για την επίλυση προβλημάτων ενισχυτικής μάθησης. Πολλές μέθοδοι που χρησιμοποιούνται για την προσέγγιση των συναρτήσεων αξίας V_π και Q_π στηρίζονται στην εξίσωση Bellman προκειμένου να οδηγήσουν στον τελικό υπολογισμό της αξίας των καταστάσεων (ή/και ενεργειών).

Μία πολιτική π θεωρείται καλύτερη σε σχέση με μία πολιτική π' αν και μόνο αν σε οποιαδήποτε κατάσταση s, η αναμενόμενη επιστροφή ακολουθώντας την πολιτική π είναι υψηλότερη ή ίση με την αναμενόμενη επιστροφή που προκύπτει ακολουθώντας την πολιτική π', δηλαδή $\pi \geq \pi'$ αν και μόνο αν $V_\pi(s) \geq V_{\pi'}(s) \forall s \in S$. Ως εκ τούτου, *βέλτιστη πολιτική* (optimal policy) π^* καλείται μία πολιτική η οποία είναι καλύτερη από οποιαδήποτε πολιτική π δηλαδή $\pi^* \geq \pi$ για κάθε πιθανή πολιτική π. Όσον αφορά τη συνάρτηση αξίας κατάστασης και τη συνάρτηση αξίας ενέργειας της βέλτιστης πολιτικής, οι οποίες καλούνται *βέλτιστη συνάρτηση αξίας κατάστασης* (optimal state-value function) και *βέλτιστη συνάρτηση αξίας ενέργειας* (optimal action-value function) αντίστοιχα, ισχύει:

$$V_*(s)=\max_\pi V_\pi(s) \qquad \forall s\in S \tag{2.25}$$

$$Q_*(s,a)=\max_\pi Q_\pi(s,a) \qquad \forall s\in S,\ a\in A \tag{2.26}$$

Οι συναρτήσεις αξίας κατάστασης και ενέργειας συνδέονται με τη σχέση:

$$V_\pi(s_t)=\sum_{a_t}\pi(a_t|s_t)Q_\pi(s_t,a_t) \tag{2.27}$$

Δεδομένης γνωστής συνάρτησης Q_*, μία ντετερμινιστική πολιτική με βάση την οποία επιλέγεται σε κάθε κατάσταση η ενέργεια με τη μεγαλύτερη αξία θα είναι βέλτιστη. Για μία τέτοια πολιτική π^*:

$$\pi^*(a_t|s_t) = \begin{cases} 1, & \text{αν } a_t = \underset{a}{\operatorname{argmax}}\, Q_*(s_t, a_t) \\ 0, & \text{αλλιώς} \end{cases} \tag{2.28}$$

θα ισχύει:

$$\begin{aligned} V_*(s_t) &= \max_a Q_*(s_t, a_t) \\ &= \max_a \mathbb{E}_{\pi^*}[G_t|s_t, a_t] \\ &= \max_a \mathbb{E}_{\pi^*}\Big[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t, a_t\Big] \\ &= \max_a \mathbb{E}_{\pi^*}\Big[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|s_t, a_t\Big] \\ &= \max_a \mathbb{E}[r_{t+1} + \gamma V_*(s_{t+1})|s_t, a_t] \\ &= \max_a \sum_{s_{t+1}} \sum_{r_{t+1}} p(s_{t+1}, r_{t+1}|s_t, a_t)[r_{t+1} + \gamma V_*(s_{t+1})] \end{aligned} \tag{2.29}$$

Η Εξίσωση 2.29 ονομάζεται *εξίσωση βελτιστοποίησης Bellman* (Bellman optimality equation) και εκφράζει ότι η αξία μίας κατάστασης, δεδομένου ότι ακολουθείται μία βέλτιστη πολιτική, ισούται με την αξία της βέλτιστης ενέργειας. Η εξίσωση βελτιστοποίησης Bellman ως προς τη συνάρτηση αξίας ενέργειας φαίνεται στην Εξίσωση 2.30.

$$\begin{aligned} Q_*(s_t, a_t) &= \mathbb{E}[r_{t+1} + \gamma \max_{a_{t+1}} Q_*(s_{t+1}, a_{t+1})|s_t, a_t] \\ &= \sum_{s_{t+1}} \sum_{r_{t+1}} p(s_{t+1}, r_{t+1}|s_t, a_t)[r_{t+1} + \gamma \max_{a_{t+1}} Q_*(s_{t+1}, a_{t+1})] \end{aligned} \tag{2.30}$$

Η εξίσωση βελτιστοποίησης Bellman έχει μοναδική λύση για πεπερασμένη ΜΔΑ, παρόλο που είναι πιθανό να υπάρχουν περισσότερες από μία βέλτιστες πολιτικές. Με γνωστή τη συνάρτηση μετάβασης P του περιβάλλοντος, η βέλτιστη συνάρτηση αξίας κατάστασης V_* και η βέλτιστη συνάρτηση αξίας ενέργειας Q_* μπορούν να προκύψουν ως λύσεις των εξισώσεων 2.29 και 2.30. Εφόσον βρεθεί η βέλτιση συνάρτηση αξίας κατάστασης V_*, είναι εφικτό να οριστεί μία βέλτιστη πολιτική με βάση την οποία σε κάθε κατάσταση s_t θα επιλέγεται η ενέργεια που οδηγεί στην επόμενη κατάσταση s_{t+1} με τη μέγιστη βέλτιστη αξία. Μία τέτοια *άπληστη* (greedy) πολιτική ως προς την αξία της κατάστασης εκτελούμενη επαναληπτικά κοιτώντας κάθε φορά ένα βήμα μπροστά, οδηγεί στην επιλογή των μακροπρόθεσμα βέλτιστων ενεργειών καθώς η αξία των καταστάσεων εκφράζει τη συνολική αναμενόμενη επιστροφή. Αντίστοιχα, δεδομένης της βέλτιστης συνάρτησης αξίας ενεργειών Q_*, μία βέλτιστη πολιτική μπορεί να προκύψει απευθείας επιλέγοντας την ενέργεια με τη βέλτιστη αξία χωρίς να απαιτείται το ενδιάμεσο βήμα υπολογισμού της αξίας κάθε κατάστασης, εφόσον η συνάρτηση αξίας ενέργειας εκφράζει την αξία κάθε ενέργειας ξεχωριστά για μία δεδομένη κατάσταση.

2.2.4.3 Δυναμικός Προγραμματισμός

Ο *Δυναμικός Προγραμματισμός* (Dynamic Programming) περιλαμβάνει ένα σύνολο αλγορίθμων, βασική αρχή των οποίων είναι η διάσπαση ενός προβλήματος σε επικαλυπτόμενα υπό-προβλήματα και

η χρήση αναδρομής για την επίλυση του αρχικού προβλήματος [19]. Στο πεδίο της ενισχυτικής μάθησης, μπορεί να χρησιμοποιηθεί για τον προσδιορισμό της βέλτιστης πολιτικής δεδομένου ενός πλήρους μοντέλου του περιβάλλοντος. Συγκεκριμένα, αξιοποιώντας τις εξισώσεις Bellman, αλγόριθμοι δυναμικού προγραμματισμού χρησιμοποιούνται για τη βελτιστοποίηση της συνάρτησης αξίας κατάστασης V_π και κατ' επέκταση της ίδιας της πολιτικής π. Δύο κύριες μέθοδοι δυναμικού προγραμματισμού έχουν αναπτυχθεί με στόχο την εύρεση μίας βέλτιστης πολιτικής, η *Επανάληψη Πολιτικής* (Policy Iteration) και η *Επανάληψη Αξίας* (Value Iteration).

Η επανάληψη πολιτικής αποτελείται από δύο αυτόνομες διαδικασίες που εκτελούνται επαναληπτικά, την *Αξιολόγηση Πολιτικής* (Policy Evaluation) και τη *Βελτίωση Πολιτικής* (Policy Improvement). Ξεκινώντας από μία τυχαία πολιτική υπολογίζεται η συνάρτηση αξίας κατάστασης για όλες τις καταστάσεις (αξιολόγηση πολιτικής) και στη συνέχεια ανανεώνεται η πολιτική ώστε σε κάθε περίπτωση να επιλέγεται η ενέργεια που οδηγεί στην κατάσταση με τη μεγαλύτερη αξία (βελτίωση πολιτικής). Αυτά τα βήματα επαναλαμβάνονται μέχρι η αρχική πολιτική να συγκλίνει στη βέλτιστη [68]. Η διαδικασία περιγράφεται στον Αλγόριθμο 1.

Algorithm 1: Policy Iteration

$V_\pi(s_t) \in \mathbb{R}, \pi(s_t) \in A$
`/* Policy Evaluation */`
repeat
  $\Delta \leftarrow 0$
  for $s_t \in S$ **do**
    $v \leftarrow V_\pi(s_t)$
    $V_\pi(s_t) \leftarrow \sum_{s_{t+1}} \sum_{r_{t+1}} p(s_{t+1}, r_{t+1}|s_t, \pi(s_t))[r_{t+1} + \gamma V_\pi(s_{t+1})]$
    $\Delta \leftarrow \max(\Delta, |v - V_\pi(s_t)|)$
until $\Delta < \theta$ `// θ is a small number`

`/* Policy Improvement */`
$stable\_policy \leftarrow True$
for $s_t \in S$ **do**
  $a_t \leftarrow \pi(s_t)$
  $\pi(s_t) \leftarrow \operatorname{argmax}_{a_t} \sum_{s_{t+1}} \sum_{r_{t+1}} p(s_{t+1}, r_{t+1}|s_t, a_t)[r_{t+1} + \gamma V_\pi(s_{t+1})]$
  if $a_t \neq \pi s_t$ **then**
    $stable\_policy \leftarrow False$
if *stable_policy* **then**
  return V_π, π
else
  go to 2

Η σύγκλιση της συνάρτησης αξίας στο στάδιο της αξιολόγησης πολιτικής που εκτελείται στην επανάληψη αξίας μπορεί να απαιτεί πολλές επαναλήψεις και να καθυστερεί τον αλγόριθμο. Ωστόσο, στην πράξη η ανανεωμένη πολιτική που προκύπτει συνήθως συγκλίνει αρκετά πιο γρήγορα, δηλαδή είναι δυνατόν η πολιτική να έχει συγκλίνει μετά από σχετικά μικρό αριθμό επαναλήψεων και η διαδικασία της αξιολόγησης να συνεχίζεται άσκοπα μεχρι τη σύγκλιση της συνάρτησης αξίας. Προκειμένου να αποφευχθεί αυτό, στον αλγόριθμο επανάληψης αξίας ο στόχος είναι ο προσδιορισμός της βέλτι-

στης συνάρτησης αξίας κατάστασης και η απευθείας εξαγωγή της βέλτιστης πολιτικής στο τέλος της διαδικασίας. Για αυτό το σκοπό χρησιμοποιείται η εξίσωση βελτιστοποίησης Bellman (Εξίσωση 2.29) για την επαναληπτική ανανέωση της συνάρτησης αξίας, συνδυάζοντας ουσιαστικά τα βήματα της αξιολόγησης και της βελτίωσης πολιτικής. Συνεπώς, ξεκινώντας από μία τυχαία συνάρτηση αξίας κατάστασης V_π υπολογίζεται επαναληπτικά η βέλτιστη συνάρτηση αξίας κατάστασης V_* και μέσω αυτής μία βέλτιστη ντετερμινιστική πολιτική π^*. Αναλυτικά τα βήματα παρουσιάζονται στον Αλγόριθμο 2.

Algorithm 2: Value Iteration

$V_\pi(s_t) \in \mathbb{R}, \pi(s_t) \in A$
repeat
  $\Delta \leftarrow 0$
  for $s_t \in S$ **do**
    $v \leftarrow V_\pi(s_t)$
    $V_\pi(s_t) \leftarrow \max_{a_t} \sum_{s_{t+1}} \sum_{r_{t+1}} p(s_{t+1}, r_{t+1}|s_t, a_t)[r_{t+1} + \gamma V_\pi(s_{t+1})]$
    $\Delta \leftarrow \max(\Delta, |v - V_\pi(s_t)|)$
until $\Delta < \theta$ `// θ is a small number`

$\pi(s_t) \leftarrow \operatorname{argmax}_{a_t} \sum_{s_{t+1}} \sum_{r_{t+1}} p(s_{t+1}, r_{t+1}|s_t, a_t)[r_{t+1} + \gamma V_\pi(s_{t+1})]$
return π

Τόσο η μέθοδος επανάληψης πολιτικής όσο και η μέθοδος επανάληψης αξίας εγγυώνται τη σύγκλιση σε μία βελτιστη πολιτική. Επιπλέον, και οι δύο βασίζονται στις εξισώσεις Bellman για τον υπολογισμό της συνάρτησης αξίας κατάστασης και τον προσδιορισμό της βέλτιστης πολιτικής. Η μέθοδος επανάληψης αξίας απαιτεί περισσότερους υπολογισμούς καθώς σε κάθε επανάληψη πρέπει να υπολογιστεί η αξία της επόμενης κατάστασης που προκύπτει από κάθε πιθανή ενέργεια. Αντίθετα, στην επανάληψη πολιτικής απαιτούνται περισσότερες επαναλήψεις για την αξιολόγηση της κάθε πολιτικής, όμως εν γένει η πολιτική συγκλίνει γρηγορότερα. Ως εκ τούτου, η επανάληψη πολιτικής οδηγεί συνήθως σε ταχύτερη σύγκλιση και έχει μικρότερο υπολογιστικό κόστος.

2.2.4.4 Μέθοδοι Μόντε Κάρλο - Χρονικής Διαφοράς

Σε πραγματικά προβλήματα ενισχυτικής μάθησης συνήθως δεν είναι γνωστό το μοντέλο του περιβάλλοντος, δηλαδή οι πιθανότητες μετάβασης στις επόμενες καταστάσεις δεδομένων των καταστάσεων και των ενεργειών που εκτελούνται. Συνεπώς, δεν είναι εφικτή η αντιμετώπιση τους με χρήση δυναμικού προγραμματισμού καθώς οι αλγόριθμοι που προαναφέρθηκαν απαιτούν γνώση της συνάρτησης μετάβασης. Στις περισσότερες περιπτώσεις ο πράκτορας καλείται να μάθει μία πολιτική επιλογής ενεργειών μέσω της εμπειρίας του, δηλαδή μέσω δειγμάτων που συλλέγει κατά την αλληλεπίδρασή του με το περιβάλλον. Δύο βασικές κατηγορίες αλγορίθμων που μπορούν να χρησιμοποιήσουν τις παρελθοντικές ακολουθίες καταστάσεων, ενεργειών και ανταμοιβών για την εκμάθηση μίας πολιτικής είναι οι μέθοδοι *Μόντε Κάρλο* (ΜΚ) (Monte Carlo - MC) και οι αλγόριθμοι *Χρονικής Διαφοράς* (ΧΔ) (Temporal Difference - TD).

Οι μέθοδοι Μόντε Κάρλο στηρίζονται στις επιστροφές που λαμβάνει ο πράκτορας για την ανανέωση των συναρτήσεων αξίας. Για αυτό το λόγο μπορούν να χρησιμοποιηθούν σε επεισοδιακά προβλήματα καθώς απαιτείται ο τερματισμός μίας ακολουθίας ενεργειών-καταστάσεων-ανταμοιβών ώστε να προκύψει η τελική επιστροφή για κάθε κατάσταση (ή ζεύγος κατάστασης-ενέργειας). Ως εκ το-

ύτου, η ανανέωση δε μπορεί να γίνει μετά από κάθε βήμα αλλά γίνεται πάντα στο τέλος του επεισοδίου. Συγκεκριμένα, η συνάρτηση αξίας ανανεώνεται με βάση τη μέση τιμή των επιστροφών που έχουν προκύψει για κάθε κατάσταση από όλα τα επεισόδια και συνεπώς πλησιάζει περισσότερο στην πραγματική αξία όσο αυξάνεται ο αριθμός των επεισοδίων.

Όπως γίνεται αντιληπτό, κατά τη διάρκεια ενός επεισοδίου ο πράκτορας μπορεί να βρεθεί περισσότερες από μία φορές στη ίδια κατάσταση. Στην πιο απλή περίπτωση που καλείται μέθοδος *Μόντε Κάρλο πρώτης επίσκεψης* (first-visit MC), λαμβάνεται υπόψη μόνο η πρώτη φορά για τον υπολογισμό της επιστροφής στο τέλος του επεισοδίου ενώ στη μέθοδο *Μόντε Κάρλο κάθε επίσκεψης* (every-visit MC) υπολογίζεται η επιστροφή για κάθε φορά που πέρασε ο πράκτορας από την ίδια κατάσταση ξεχωριστά. Η ανανέωση της συνάρτησης αξίας γίνεται στο τέλος κάθε επεισοδίου με βάση τον παρακάτω κανόνα για την αξία κατάστασης και την αξία ενέργειας αντίστοιχα:

$$V_\pi(s_t) \leftarrow V_\pi(s_t) + \alpha[G_t - V_\pi(s_t)] \tag{2.31}$$

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha[G_t - Q_\pi(s_t, a_t)] \tag{2.32}$$

όπου G_t είναι η επιστροφή από την κατάσταση s_t μέχρι το τέλος του επεισοδίου και α είναι μία παράμετρος που καθορίζει τη συμβολή της κάθε ανανέωσης στην αλλαγή της τρέχουσας εκτίμησης.

Σε αντίθεση με τις μεθόδους Μόντε Κάρλο στις οποίες πρέπει να ολοκληρωθεί ένα επεισόδιο για να ανανεωθεί η συνάρτηση αξίας, η μέθοδος χρονικής διαφοράς επιτρέπει την ανανέωση μετά από κάθε βήμα. Για να επιτευχθεί αυτό, αντί για την επιστροφή G_t σχηματίζεται ένας νέος στόχος που περιλαμβάνει την άμεση ανταμοιβή r_t από την εκτέλεση της ενέργειας και την εκτίμηση $V_\pi(s_{t+1})$ της αξίας της επόμενης κατάστασης. Με αυτόν τον τρόπο δε χρειάζεται να τερματιστεί το επεισόδιο καθώς η εκτίμηση του νέου στόχου ειναι διαθέσιμη μετά από κάθε βήμα. Οι αντίστοιχες ανανεώσεις των συναρτήσεων αξίας στους αλγορίθμους χρονικής διαφοράς γίνονται ως εξής:

$$V_\pi(s_t) \leftarrow V_\pi(s_t) + \alpha[r_{t+1} + \gamma V_\pi(s_{t+1}) - V_\pi(s_t)] \tag{2.33}$$

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha[r_{t+1} + \gamma Q_\pi(s_{t+1}, a_{t+1}) - Q_\pi(s_t, a_t)] \tag{2.34}$$

Στη φόρμουλα ανανέωσης 2.34 περιλαμβάνεται η συνάρτηση αξίας ενέργειας της επόμενης κατάστασης s_{t+1} για μία επόμενη ενέργεια a_{t+1}. Ο τρόπος επιλογής αυτής της ενέργειας είναι κομβικής σημασίας και επηρεάζει σημαντικά τον αλγόριθμο χρονικής διαφοράς. Συγκεκριμένα, στις δύο κυριότερες προσεγγίσεις, η επόμενη ενέργεια μπορεί είτε να προκύπτει από την τρέχουσα πολιτική (δηλαδή να είναι η $\pi(s_{t+1})$) είτε να επιλέγεται ως αυτή με τη μεγαλύτερη αξία Q. Οι δύο αλγόριθμοι που προκύπτουν καλούνται *SARSA* (State-Action-Reward-State-Action) και *Εκμάθηση-Q* (Q-Learning) αντίστοιχα και αποτελούν θεμελιώδεις τεχνικές στο πεδίο της ενισχυτικής μάθησης. Η ανανέωση στον αλγόριθμο Q-Learning φαίνεται ακολούθως.

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a_{t+1}} Q_\pi(s_{t+1}, a_{t+1}) - Q_\pi(s_t, a_t)] \tag{2.35}$$

Ο αλγόριθμος SARSA ουσιαστικά ενημερώνει τη συνάρτηση αξίας ενέργειας της πολιτικής που ακολουθεί ο πράκτορας κατά την περιήγησή του στο πρειβάλλον. Αντιθέτως, στον αλγόριθμο Q-Learning η πολιτική που ενημερώνεται είναι ανεξάρτητη αυτής που χρησιμοποιείται για την επιλογή των ενεργειών. Με βάση αυτή τους την ιδιότητα ο πρώτος καλείται αλγόριθμος *εντός πολιτικής* (on-policy) ενώ ο δεύτερος *εκτός πολιτικής* (off-policy) (βλ. Ενότητα 2.2.4.5).

Στις μεθόδους χρονικής διαφοράς, υπολογίζεται ένας νέος στόχος για τη συνάρτηση αξίας χρησιμοποιώντας την αμέσως επόμενη ανταμοιβή και την εκτίμηση της αξίας της επόμενης κατάστασης. Στη γενική περίπτωση, αυτός ο στόχος μπορεί να περιλαμβάνει περισσότερα ενδιάμεσα βήματα μέχρι τη χρήση της αξίας κάποιας επόμενης κατάστασης. Ορίζοντας την επιστροφή n-βημάτων G_t^{t+n} ως:

$$G_t^{t+n} = r_{t+1} + \gamma r_{t+2} + ... + \gamma^{n-1} r_{t+n} + \gamma^n V_\pi(s_{t+n}) \tag{2.36}$$

η συνάρτηση αξίας μπορεί να ανανεωθεί με βάση τη φόρμουλα:

$$V_\pi(s_t) \leftarrow V_\pi(s_t) + \alpha[G_t^{t+n} - V_\pi(s_t)] \tag{2.37}$$

Οι μέθοδοι χρονικής διαφοράς n-βημάτων συνδυάζουν τις ακραίες περιπτώσεις των μεθόδων Μόντε Κάρλο και χρονικής διαφοράς ενός βήματος. Η μέθοδος χρονικής διαφοράς ενός βήματος αποτελεί ουσιαστικά ειδική περίπτωση της μεθόδου χρονικής διαφοράς n-βημάτων με $n = 1$. Εν αντιθέσει με τις μεθόδους Μόντε Κάρλο, ένα βασικό πλεονέκτημα των μεθόδων χρονικής διαφοράς είναι ότι μπορούν να εφαρμοστούν και σε ακολουθιακά (μη επεισοδιακά) περιβάλλοντα καθώς δεν απαιτείται η συνολική επιστροφή για τον υπολογισμό της συνάρτησης αξίας.

2.2.4.5 Κατηγορίες Αλγορίθμων Επίλυσης

Οι αλγόριθμοι που χρησιμοποιούνται για την επίλυση προβλημάτων ενισχυτικής μάθησης μπορούν να ταξινομηθούν σε κατηγορίες με βάση διαφορετικά κριτήρια. Μία πρώτη βασική κατηγοριοποίηση αφορά στη γνώση του μοντέλου του περιβάλλοντος. Με βάση αυτό το χαρακτηριστικό, οι αλγόριθμοι μπορούν να χωριστούν σε δύο κατηγορίες:

- *Βασισμένοι σε μοντέλο* (Model-based): έχουν πρόσβαση (ή προσεγγίζουν μέσω κάποιας διαδικασίας εκπαίδευσης) στο μοντέλο του περιβάλλοντος, δηλαδή στη συνάρτηση μετάβασης P_a (βλ. παράγραφο 2.2.4.1).
- *Ανεξάρτητοι μοντέλου* (Model-free): δεν έχουν πρόσβαση στο μοντέλο παρά μόνο τη δυνατότητα προσομοίωσης, δηλαδή μπορούν να εκτελούν ενέργειες και να αλληλεπιδρούν με το περιβάλλον λαμβάνοντας ανταμοιβές και μεταβαίνοντας σε νέες καταστάσεις.

Η γνώση της δυναμικής του περιβάλλοντος δίνει τη δυνατότητα του σχεδιασμού εκ των προτέρων, λαμβάνοντας υπόψη τις μελλοντικές καταστάσεις και ανταμοιβές που μπορούν να προκύψουν από την εκτέλεση των πιθανών ενεργειών. Ωστόσο, στην πράξη τις περισσότερες φορές το μοντέλο του περιβάλλοντος δεν είναι διαθέσιμο και ως εκ τούτου πρέπει είτε να το προσεγγίσει ο πράκτορας είτε να καθορίσει την πολιτική του χωρίς γνώση των πιθανοτήτων μετάβασης. Οι πιο διαδεδομένοι αλγόριθμοι είναι ανεξάρτητοι μοντέλου και βασίζονται στην εμπειρία που συλλέγει ο πράκτορας από προηγούμενες ενέργειες, μεταβάσεις και ανταμοιβές προκειμένου να διαμορφώσουν την πολιτική επιλογής ενεργειών.

Αναφορικά με τον στόχο της εκπαίδευσης μπορούν να προκύψουν επιπλέον υποκατηγορίες. Συγκεκριμένα, οι αλγόριθμοι που βασίζονται σε μοντέλο μπορούν να κατηγοριοποιηθούν σε δύο βασικές κλάσεις:

- *Εκμάθησης μοντέλου* (Learn the model): η πιο συχνή περίπτωση στην οποία το μοντέλο του περιβάλλοντος δεν είναι γνωστό και επομένως ο πράκτορας καλείται να προσεγγίσει τη συνάρτηση μετάβασης. Περιλαμβάνει ένα μεγάλο εύρος αλγορίθμων καθώς το μοντέλο μπορεί στη

συνέχεια να χρησιμοποιηθεί με διαφορετικές μεθόδους όπως για το σχεδιασμό μίας ακολουθίας ενεργειών, για την επαύξηση δεδομένων με συνθετικά δείγματα κ.λπ.

- *Δεδομένου μοντέλου* (Given the model): σε αυτή την περίπτωση ο πράκτορας έχει πρόσβαση στη συνάρτηση μετάβασης καταστάσεων και μπορεί να τη χρησιμοποιήσει απευθείας προκειμένου να καθορίσει την επιθυμητή πολιτική. Σε πραγματικά προβλήματα συνήθως δεν ισχύει αυτό καθώς οι ιδιότητες του μοντέλου δεν είναι γνωστές.

Οι αλγόριθμοι που είναι ανεξάρτητοι μοντέλου διακρίνονται ανάλογα με τον τρόπο προσδιορισμού της πολιτικής, δηλαδή με το αν στοχεύουν απευθείας στην εκμάθηση της πολιτικής ή στην εκμάθηση της αξίας των καταστάσεων και των ενεργειών, στις εξής κατηγορίες:

- *Βασισμένοι στην πολιτική* (Policy-based): στόχος είναι η εύρεση της βέλτιστης πολιτικής μεγιστοποιώντας (ή ελαχιστοποιώντας αντίστοιχα) μία αντικειμενική συνάρτηση που εκφράζει ένα μέτρο της απόδοσης του πράκτορα (συνήθως όσον αφορά τη συνολική ανταμοιβή που λαμβάνει ακολουθώντας την πολιτική).
- *Βασισμένοι στην αξία* (Value-based): σε αυτή την περίπτωση ο στόχος του πράκτορα είναι να προσεγγίσει όσο το δυνατόν καλύτερα τη συνάρτηση αξίας ενεργειών. Με αυτό τον τρόπο η βέλτιστη πολιτική μπορεί να προκύψει εμμέσως επιλέγοντας σε κάθε κατάσταση την ενέργεια με τη μεγαλύτερη αξία.

Οι αλγόριθμοι που βασίζονται στην αξία είναι απλούστεροι στην υλοποίηση και πιο αποδοτικοί όσον αφορά τη χρήση των δειγμάτων εκπαίδευσης. Αυτό έχει ως αποτέλεσμα να έχουν συνήθως ταχύτερη σύγκλιση. Ωστόσο, μπορούν να διαχειριστούν μόνο προβλήματα με διακριτό χώρο ενεργειών καθώς πρέπει να μπορεί να υπολογιστεί η αξία κάθε ενέργειας προκειμένου να οριστεί η πολιτική. Επιπλέον, εφόσον η βέλτιστη πολιτική απορρέει από την αξία των ενεργειών, είναι ντετερμινιστική καθώς σε κάθε περίπτωση επιλέγεται η ενέργεια με τη μέγιστη αξία. Αντιθέτως, οι αλγόριθμοι που βασίζονται στην πολιτική μπορούν να οδηγήσουν στην εκμάθηση στοχαστικών πολιτικών. Οι στοχαστικές πολιτικές έχουν το πλεονέκτημα ότι αντιμετωπίζουν εγγενώς το πρόβλημα της εξερεύνησης του χώρου ενεργειών, ενώ στις ντετερμινιστικές πολιτικές πρέπει να οριστεί μία μέθοδος εξερεύνησης.

Προκειμένου να αξιοποιηθούν τα πλεονεκτήματα της κάθε προσέγγισης και να περιοριστούν τα αντίστοιχα μειονεκτήματα, ορισμένοι αλγόριθμοι συνδυάζουν τις δύο μεθοδολογίες. Συνεπώς, πέραν των αμιγώς βασισμένων στην πολιτική ή στην αξία αλγορίθμων, υπάρχουν και υβριδικές τεχνικές που χρησιμοποιούν τη συνάρτηση αξίας για να βελτιώσουν την πολιτική και αντιστρόφως. Παρόλα αυτά, η απόδοση ενός αλγορίθμου εξαρτάται σε μεγάλο βαθμό από τη φύση του προβλήματος και το περιβάλλον στο οποίο εφαρμόζεται και δεν υπάρχει μία μέθοδος που να υπερτερεί των υπολοίπων σε κάθε περίπτωση.

Άλλη μία κατηγοριοποίηση αφορά την πολιτική που χρησιμοποιείται για την ανανέωση της τρέχουσας πολιτικής του πράκτορα. Υπό αυτό το πρίσμα υπάρχουν δύο διαφορετικές προσεγγίσεις:

- *Εντός πολιτικής* (On-policy): η ανανέωση της πολιτικής κατά την εκπαίδευση στηρίζεται αποκλειστικά στις ενέργειες που προκύπτουν από την πολιτική που ακολουθεί εκείνη τη στιγμή ο πράκτορας.
- *Εκτός πολιτικής* (Off-policy): η ανανέωση της πολιτικής βασίζεται σε ενέργειες μίας πολιτικής διαφορετικής από αυτή που ακολουθεί ο πράκτορας εκείνη τη στιγμή (συνήθως η ανανέωση βασίζεται σε μία άπληστη πολιτική ανεξαρτήτως της τρέχουσας πολιτικής).

2.2.4.6 Θεμελιώδεις Αλγόριθμοι Ενισχυτικής Μάθησης

Στα περισσότερα προβλήματα ενισχυτικής μάθησης, όπως έχει ήδη αναφερθεί, το μοντέλο του περιβάλλοντος δεν είναι διαθέσιμο στον πράκτορα. Συνεπώς, οι πιο συχνά χρησιμοποιούμενοι αλγόριθμοι είναι ανεξάρτητοι μοντέλου. Δύο θεμελιώδεις αλγόριθμοι στους οποίους βασίζονται οι περισσότερες σύγχρονες τεχνικές είναι ο αλγόριθμος *Βαθιάς Εκμάθησης-Q* (Deep Q-Learning - DQN) [67] και ο αλγόριθμος *REINFORCE* [95], οι οποίοι είναι βασισμένοι στην αξία και στην πολιτική αντίστοιχα.

Ο DQN ειναι βασισμένος στον αλγόριθμο Q-Learning (βλ. Παράγραφο 2.2.4.4), προσαρμοσμένος κατάλληλα ώστε να μπορεί να εφαρμοστεί σε προβλήματα με μεγάλο χώρο καταστάσεων. Στόχος είναι ο προσδιορισμός της συνάρτησης αξίας ενέργειας Q μέσω της οποίας μπορεί να εξαχθεί μία πολιτική (στη γενική περίπτωση η άπληστη πολιτική με βάση την οποία επιλέγεται σε κάθε κατάσταση η ενέργεια με τη μεγαλύτερη αξία). Ο υπολογισμός της συνάρτησης αξίας ενέργειας σε μορφή πίνακα, όπως γίνεται στον αλγόριθμο Q-Learning, δεν είναι πρακτικός για περιβάλλοντα με πολλές διαφορετικές καταστάσεις. Για αυτό το λόγο, στον DQN χρησιμοποιούνται νευρωνικά δίκτυα για την προσέγγιση της. Με αυτό τον τρόπο αφενός το μοντέλο μπορεί να διαχειριστεί τον χώρο καταστάσεων ανεξάρτητα από το μέγεθος του και αφετέρου μπορεί να αξιοποιήσει διαφορετικές μορφές αναπαράστασης της κατάστασης ως είσοδο όπως διανύσματα, εικόνες κ.λπ.

Συγκεκριμένα, στον αλγόριθμο DQN ένα νευρωνικό δίκτυο χρησιμοποιείται για την προσέγγιση της συνάρτησης αξίας ενέργειας έτσι ώστε $Q(s, a; \theta) \approx Q^*(s, a)$, όπου $s \in S$ είναι η αναπαράσταση μίας κατάστασης του περιβάλλοντος, $a \in A$ είναι μια πιθανή ενέργεια και θ είναι τα βάρη του μοντέλου. Η συνάρτηση κόστους (η οποία προέρχεται από την εξίσωση βελτιστοποίησης Bellman) που χρησιμοποιείται για την εκπαίδευση του νευρωνικού δικτύου είναι η εξής:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim U(D)}[(r + \gamma \max_{a'} Q(s', a'; \theta) - Q(s, a; \theta^-))^2] \tag{2.38}$$

Ο πράκτορας επιλέγει ενέργειες με βάση μία *ϵ-άπληστη* (ϵ-greedy) πολιτική ώστε να εξισορροπήσει την εκμετάλλευση με την εξερεύνηση. Με βάση αυτή την πολιτική επιλέγει με μία μικρή πιθανότητα ϵ μία τυχαία ενέργεια προκειμένου να εξερευνήσει καταστάσεις με ενδεχόμενη μεγαλύτερη επιστροφή και με πιθανότητα $1 - \epsilon$ την ενέργεια με τη μεγαλύτερη τρέχουσα εκτιμώμενη αξία. Για την εκπαίδευση του δικτύου, τα δείγματα (s, a, r, s') αντλούνται από ένα τμήμα *προσωρινής μνήμης* (buffer) D, όπου αποθηκεύονται κατά τη διάρκεια της αλληλεπίδρασης του πράκτορα με το περιβάλλον. Αυτό γίνεται προκειμένου να αποσυσχετιστούν τα δεδομένα και να μη χρησιμοποιούνται με τη σειρά που εμφανίζονται κατά την περιήγηση του πράκτορα. Έτσι, σε κάθε βήμα ο πράκτορας αποθηκεύει τη νέα πλειάδα (s, a, r, s') αλλά χρησιμοποιεί προηγούμενα, τυχαία δείγματα από τη μνήμη για την ανανέωση των βαρών του δικτύου. Επιπλέον, με αυτή τη μέθοδο κάθε δείγμα μπορεί να χρησιμοποιηθεί περισσότερες από μία φορές αυξάνοντας την αποδοτικότητα χρήσης των δεδομένων, ιδίως σε περιβάλλοντα με υψηλό κόστος προσομοίωσης.

Όπως φαίνεται στην Εξίσωση 2.38, ο εκάστοτε στόχος (ετικέτα) εξαρτάται από το δίκτυο που εκπαιδεύεται. Αυτό έχει σαν αποτέλεσμα την αποσταθεροποίηση της εκπαίδευσης καθώς με κάθε ανανέωση του δικτύου αλλάζει και ο στόχος που προσπαθεί να προβλέψει. Προκειμένου να γίνει πιο ομαλή η διαδικασία της εκπαίδευσης, χρησιμοποιείται ένα δεύτερο δίκτυο-στόχος, τα βάρη θ^- του οποίου παραμένουν σταθερά για συγκεκριμένο αριθμό επαναλήψεων. Έτσι, σε κάθε υπολογισμό του κόστους το δίκτυο που εκπαιδεύεται χρησιμοποιείται για την πρόβλεψη της αξίας ενέργειας για το ζεύγος επόμενης κατάστασης-ενέργειας ενώ το σταθερό δίκτυο παράγει την πρόβλεψη-στόχο που αφορά την τρέχουσα κατάσταση. Τα βάρη του δικτύου-στόχου ενημερώνονται περιοδικά ανά συγκε-

κριμένο αριθμό επαναλήψεων ώστε να ακολουθεί την πορεία του βασικού δικτύου και να βελτιώνεται παράλληλα.

Ο αλγόριθμος *REINFORCE* ανήκει στην ευρύτερη κατηγορία των μεθόδων *Κλίσης Πολιτικής* (Policy Gradient) που έχουν ως στόχο τον απευθείας προσδιορισμό της πολιτικής (χωρίς την εκτίμηση της συνάρτησης αξίας κατάστασης ή ενέργειας). Σε αυτή την περίπτωση, ένα νευρωνικό δίκτυο χρησιμοποιείται για την παραμετροποίηση της πολιτικής π_θ που ακολουθεί ο πράκτορας και στόχος είναι η μεγιστοποίηση της αναμενόμενης ανταμοιβής $J(\theta) = \mathbb{E}_{\tau\sim\pi}[r(\tau)]$ που προκύπτει από μία διαδρομή τ ακολουθώντας αυτή την πολιτική. Συνεπώς, είναι ένα κλασικό πρόβλημα μηχανικής μάθησης στο οποίο αναζητούνται οι βέλτιστες παράμετροι θ^* που μεγιστοποιούν τη συνάρτηση J. Μία κλασική μέθοδος αντιμετώπισης είναι η άνοδος κλίσης (αντίστοιχη της καθόδου κλίσης βλ. Παράγραφο 2.2.3.2) κατά την οποία τα βάρη του δικτύου ανανεώνονται σε κάθε επανάληψη ως εξής:

$$\theta_{n+1} \leftarrow \theta_n + \eta \cdot \nabla_\theta J(\theta_n) \tag{2.39}$$

Για την παράγωγο της συνάρτησης $J(\theta)$ που θέλουμε να μεγιστοποιήσουμε ισχύει:

$$\begin{aligned}
&\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_\pi[r(\tau)] \\
&= \mathbb{E}_\pi\big[r(\tau)\nabla_\theta \log \pi(\tau)\big] \\
&= \mathbb{E}_\pi\left[r(\tau)\nabla_\theta \log\Big[P(s_0)\prod_{i=1}^{T}\pi_\theta(a_t|s_t)p(s_{t+1},r_{t+1}|s_t,a_t)\Big]\right] \\
&= \mathbb{E}_\pi\left[r(\tau)\nabla_\theta\Big[\log P(s_0) + \sum_{t=1}^{T}\log\pi_\theta(a_t|s_t) + \sum_{t=1}^{T}\log p(s_{t+1},r_{t+1}|s_t,a_t)\Big]\right] \\
&= \mathbb{E}_\pi\left[r(\tau)\sum_{t=1}^{T}\nabla_\theta\log\pi_\theta(a_t|s_t)\right]
\end{aligned} \tag{2.40}$$

Από την Εξίσωση 2.40 φαίνεται ότι η παράγωγος δεν εξαρτάται από τη συνάρτηση μετάβασης του περιβάλλοντος. Ως εκ τούτου, ο αλγόριθμος είναι ανεξάρτητος μοντέλου και μπορεί να εφαρμοστεί σε προβλήματα ενισχυτικής μάθησης δεδομένου μόνο ενός μοντέλου προσομοίωσης. Συγκεκριμένα, στον αλγόριθμο REINFORCE χρησιμοποιείται η επιστροφή G_t για τον υπολογισμό της παραγώγου. Η ανανέωση των βαρών γίνεται μετά το τέλος κάθε επεισοδίου σύμφωνα με την Εξίσωση 2.41.

$$\theta_{n+1} \leftarrow \theta_n + \eta G_t \nabla_\theta \log \pi_\theta(s_t, a_t) \tag{2.41}$$

Επιπλέον ο αλγόριθμος REINFORCE είναι αλγόριθμος εντός πολιτικής σε αντίθεση με τον αλγόριθμο Βαθιάς Εκμάθησης-Q. Αυτό το γεγονός σε συνδυασμό με το ότι η πολιτική είναι στοχαστική, έχει σαν αποτέλεσμα να μην απαιτείται επιπλέον τροποποίηση για την εξερεύνηση νέων ενεργειών και καταστάσεων καθώς η εξερεύνηση πραγματοποιείται εμμέσως με την επιλογή των ενεργειών κατά την αλληλεπίδραση του πράκτορα με το περιβάλλον. Αναμενόμενα, καθώς εκτελούνται περισσότερα επεισόδια και η εκπαίδευση συνεχίζεται, η πολιτική αρχίζει να συγκλίνει στις ενέργειες με τη μεγαλύτερη αξία και οι υπόλοιπες ενέργειες επιλέγονται με μικρότερη πιθανότητα.

Η χρήση της επιστροφής G_t εισάγει σημαντική διακύμανση στους υπολογισμούς που αφορούν την ανανέωση των βαρών, επιβραδύνοντας τη σύγκλιση τους. Για την αντιμετώπιση αυτού του φαινομένου, μία αποτελεσματική μέθοδος είναι η εισαγωγή ενός *σημείου αναφοράς* (baseline) b τέτοιου ώστε:

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi \left[(G_t - b) \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \right] \tag{2.42}$$

Με αυτό τον τρόπο, χρησιμοποιώντας ένα σημείο αναφοράς ανεξάρτητο από τις παραμέτρους του δικτύου της πολιτικής, είναι δυνατόν να μειωθεί η διακύμανση χωρίς την εισαγωγή επιπλέον μεροληψίας. Αντικαθιστώντας την επιστροφή G_t (ή αντίστοιχα την ποσότητα $G_t - b$) με τη συνάρτηση αξίας ενέργειας Q προκύπτει ο αλγόριθμος *Δράστη-Κριτή* (Actor-Critic) [66]. Σε αυτό τον αλγόριθμο χρησιμοποιούνται δύο ξεχωριστά νευρωνικά δίκτυα, το δίκτυο *Δράστη* (Actor) που παραμετροποιεί την πολιτική του πράκτορα π_θ (ακριβώς όπως στον αλγόριθμο REINFORCE) και το δίκτυο *Κριτή* (Critic) που παραμετροποιεί τη συνάρτηση αξίας κατάστασης Q_w. Σε κάθε βήμα ο πράκτορας επιλέγει μία ενέργεια με βάση την πολιτική του δικτύου-δράστη και ματαβαίνει σε μία νέα κατάσταση λαμβάνοντας την αντίστοιχη ανταμοιβή. Το δίκτυο-κριτής προβλέπει την αξία ενέργειας και χρησιμοποιείται για την ανανέωση των βαρών του δικτύου-δράστη. Στη συνέχεια τα βάρη του δικτύου-κριτή ανανεώνονται με βάση το σφάλμα χρονικής διαφοράς όπως συμβαίνει στον αλγόριθμο Βαθιάς Εκμάθησης-Q. Η ανανέωση των βαρών θ και w των δικτύων δράστη και κριτή αντίστοιχα γίνεται ως εξής:

$$\theta_{n+1} \leftarrow \theta_n + \eta_\theta Q_w(s_t, a_t) \nabla_\theta \log \pi_\theta(s_t, a_t) \tag{2.43}$$

$$w_{n+1} \leftarrow w_n + \eta_w (r_{t+1} + \gamma Q_w(s_{t+1}, a_{t+1}) - Q_w(s_t, a_t)) \nabla_w Q_w(s_t, a_t) \tag{2.44}$$

Ουσιαστικά, αυτή η τεχνική συνδυάζει τους αλγορίθμους βασισμένους στην αξία και την πολιτική με στόχο την αξιοποίηση των πλεονεκτημάτων που προσφέρει η κάθε κατηγορία ξεχωριστά. Αρχικά όπως ήδη αναφέρθηκε, μειώνεται η διακύμανση που προκύπτει από τα δείγματα των διαφορετικών διαδρομών. Επίσης, με τη χρήση της συνάρτησης αξίας ενέργειας δεν είναι αναγκαίο να ολοκληρωθεί ένα επεισόδιο προκειμένου να ανανεωθούν τα βάρη του δικτύου πολιτικής καθώς πλέον η επιστροφή αντικαθίσταται από την εκτίμηση της αξίας ενέργειας. Η καθοδήγηση της πολιτικής από το δίκτυο κριτή, συμβάλλει επιπλέον στην καλύτερη εξερεύνηση αυξάνοντας την αποδοτικότητα των δειγμάτων εκπαίδευσης. Όσον αφορά στις ενέργειες, καθώς η πολιτική προκύπτει από το δίκτυο-δράστη μπορεί να εφαρμοστεί σε συνεχείς χώρους ενεργειών, κάτι που δεν είναι εφικτό στους αλγορίθμους που βασίζονται στην αξία. Συνεπώς, η υβριδική τεχνική δράστη-κριτή (και οι παραλλαγές της) συνδυάζει ως επί το πλείστον τα επιμέρους πλεονεκτήματα των αλγορίθμων που βασίζονται στην αξία και στην πολιτική, εισάγοντας ωστόσο επιπλέον υπολογιστικό κόστος καθώς πρέπει να εκπαιδευτούν παράλληλα δύο διαφορετικά νευρωνικά δίκτυα για την υλοποίησή του αλγορίθμου.

2.3 Γενετικοί Αλγόριθμοι

2.3.1 Εισαγωγή

Οι *Γενετικοί Αλγόριθμοι* (ΓΑ) (Genetic Algorithms - GA) είναι μία κατηγορία αλγορίθμων που στηρίζεται σε βασικές έννοιες της εξελικτικής διαδικασίας στη φύση και ανήκει στην ευρύτερη κατηγορία των *Εξελικτικών Αλγορίθμων* (ΕΑ) (Evolutionary Algorithms - EA). Η κύρια λογική γύρω από την οποία αναπτύχθηκαν είναι αυτή της βιολογικής εξέλιξης, κατά την οποία με την πάροδο του χρόνου σε ένα σύστημα επιβιώνουν και αναπαράγονται τα άτομα με τα πιο "ισχυρά" χαρακτηριστικά σε σημαντικά υψηλότερο ποσοστό από τα "αδύναμα" άτομα. Αυτό έχει ως αποτέλεσμα οι απόγονοι

που θα προκύψουν να κληρονομήσουν σε μεγαλύτερο βαθμό τα συγκεκριμένα χαρακτηριστικά και κατά συνέπεια οι νεότερες γενιές να είναι πιο "ισχυρές" από τις προηγούμενες.

Η εισαγωγή στην έννοια των εξελικτικών αλγορίθμων έγινε τη δεκαετία του 1950. Πρώτος ο Alan Turing χρησιμοποίησε τον όρο "learning machine" για να περιγράψει ένα μοντέλο που θα προσομοίωνε τις βασικές αρχές της εξέλιξης [97]. Στην πράξη αυτή η προσομοίωση της εξέλιξης με τη χρήση υπολογιστή υλοποιήθηκε το 1954 από τον Nils Aall Barricelli και ακολούθησαν αρκετές προσπάθειες ανάπτυξης ΕΑ στις δεκαετίες 1950-1960 τόσο στο πεδίο της βιολογίας όσο και σε αυτό της τεχνητής νοημοσύνης. Το 1965 ο Rachenberg παρουσίασε τις *Στρατηγικές Εξέλιξης* (Evolution Strategies) [90] και ένα χρόνο αργότερα οι Fogel, Walsh και Owens ανέπτυξαν την ιδέα του *Εξελικτικού Προγραμματισμού* (Evolutionary Programming), συστήματα που αποτελούν τις κύριες υποκατηγορίες εξελικτικών αλγορίθμων μαζί με τους γενετικούς.

Οι γενετικοί αλγόριθμοι αναπτύχθηκαν στις αρχές της δεκαετίας του 1970 από τον John Holland και τους συνεργάτες του στο Πανεπιστήμιο του Michigan και διαδόθηκαν ευρέως από το βιβλίο του "Adaptation in Natural and Artificial Systems". Η διαφορά σε σχέση με τις στρατηγικές εξέλιξης και τον εξελικτικό προγραμματισμό είναι ότι ο Holland δεν επεδίωξε να εφαρμόσει τους ΓΑ σε ένα συγκεκριμένο πρόβλημα, αλλά τον ενδιέφερε να εισάγει τον μηχανισμό της εξέλιξης όπως συμβαίνει στη φύση, στον υπολογιστή [64]. Ο ίδιος ανέπτυξε τη μέθοδο με την οποία αναπαράγονται οι γενιές και προκύπτουν οι νέοι πληθυσμοί και όρισε τους βασικούς τελεστές επιλογής, διασταύρωσης και μετάλλαξης που απαιτούνται για αυτή τη διαδικασία.

Το πεδίο εφαρμογής τους αποτελείται κυρίως από προβλήματα βελτιστοποίησης στα οποία δεν υπάρχει αναλυτική μέθοδος αναζήτησης που να εγγυάται την εύρεση επιθυμητής λύσης. Αυτό συμβαίνει συνήθως σε προβλήματα με πολύ μεγάλο αριθμό παραμέτρων/διαστάσεων όπου το μέγεθος του χώρου αναζήτησης δεν επιτρέπει την αξιολόγηση όλων των δυνατών συνδυασμών και το σχηματισμό του βέλτιστου. Η βασική διαφορά των ΓΑ από άλλες μεθόδους επίλυσης αυτού του είδους προβλημάτων είναι ότι διατηρούν ανά πάσα στιγμή έναν πληθυσμό πιθανών λύσεων, ο οποίος τους επιτρέπει να ελέγχουν ταυτόχρονα πολλές κατευθύνσεις του *χώρου αναζήτησης* (search space). Αντίθετα, σε τεχνικές στις οποίες σε κάθε βήμα εξετάζεται μία μεμονωμένη λύση, είναι πιο εύκολο ο αλγόριθμος να περιοριστεί σε μία συγκεκριμένη κατεύθυνση λύσεων και να εγκλωβιστεί σε τοπικά μέγιστα (ή ελάχιστα ανάλογα με τη φύση του προβλήματος).

2.3.2 Βασικά Συστατικά και Υλοποίηση

2.3.2.1 Περιγραφή και Δομή

Σε έναν γενετικό αλγόριθμο στόχος είναι ο προσδιορισμός των τιμών κάποιων παραμέτρων μέσα από μια διαδικασία που ακολουθεί το μοντέλο της βιολογικής εξέλιξης. Αρχικά οι παράμετροι κωδικοποιούνται ώστε να μπορούν να αναπαρασταθούν από μία ακολουθία δυαδικών ψηφίων. Στη συνέχεια δημιουργείται (συνήθως με τυχαίο τρόπο) ένας αρχικός πληθυσμός από άτομα, καθένα από τα οποία περιλαμβάνει μία τέτοια ακολουθία και αποτελεί λύση του προβλήματος. Τα άτομα αξιολογούνται σύμφωνα με μία *συνάρτηση ποιότητας* (fitness function) προσαρμοσμένη στο συγκεκριμένο πρόβλημα και επιβιώνουν αυτά με την υψηλότερη τιμή, τα οποία τελικά αναπαράγονται και δημιουργούν την επόμενη γενιά λύσεων. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να βρεθεί ένα άτομο με ικανοποιητική τιμή της συνάρτησης ποιότητας, το οποίο θα είναι και η τελική λύση του προβλήματος.

Η κωδικοποιημένη αναπαράσταση μίας πιθανής λύσης σε έναν ΓΑ αποτελείται από τα ακόλουθα

δομικά στοιχεία [57]:

Γονίδιο (Gene) Είναι το πιο απλό δομικό συστατικό του γενετικού αλγορίθμου και είναι αυτό που περιέχει την πληροφορία, καθώς κάθε παράμετρος του προβλήματος που θέλουμε να επιλύσουμε κωδικοποιείται κατάλληλα και αντιστοιχίζεται σε ένα γονίδιο. Συνήθως τα γονίδια είναι δυαδικά στοιχεία τα οποία μπορούν να πάρουν τις τιμές 0 ή 1, όμως αυτό δεν είναι απαραίτητο. Ανάλογα με την υλοποίηση του αλγορίθμου τα γονίδια είναι δυνατόν να παίρνουν περισσότερες τιμές ή ακόμα και τιμές άλλου τύπου (πχ πραγματικές), όμως σε κάθε υλοποίηση πρέπει όλα τα γονίδια να είναι του ίδιου τύπου.

Χρωμόσωμα (Chromosome) Είναι μία ακολουθία γονιδίων που αποτελείται από τουλάχιστον ένα γονίδιο. Στη γενική περίπτωση ένας γενετικός αλγόριθμος περιλαμβάνει έναν σημαντικό αριθμό γονιδίων τα οποία ομαδοποιούνται στα χρωμοσώματα. Συνήθως όλη η απαραίτητη πληροφορία περιλαμβάνεται σε ένα χρωμόσωμα, όμως είναι πιθανόν να υπάρχουν περισσότερα από ένα είτε για λόγους οργάνωσης είτε αν υπάρχουν γονίδια με διαφορετικό φάσμα τιμών, τα οποία δεν μπορούν να κωδικοποιηθούν στο ίδιο χρωμόσωμα. Κάθε χρωμόσωμα μπορεί να έχει διαφορετικό μήκος (δηλαδή να αποτελείται από διαφορετικό αριθμό γονιδίων) όπως και διαφορετικό φάσμα τιμών για τα γονίδιά του αλλά όλα πρέπει να περιλαμβάνουν γονίδια του ίδιου τύπου (πχ δυαδικά, ακέραια κλπ).

Γενότυπος (Genotype) Είναι το σύνολο των χρωμοσωμάτων που κωδικοποιούν την πληροφορία και ουσιαστικά κάθε γενότυπος αντιστοιχεί σε μία πιθανή λύση του προβλήματος, με παραμέτρους τις τιμές των χρωμοσωμάτων του.

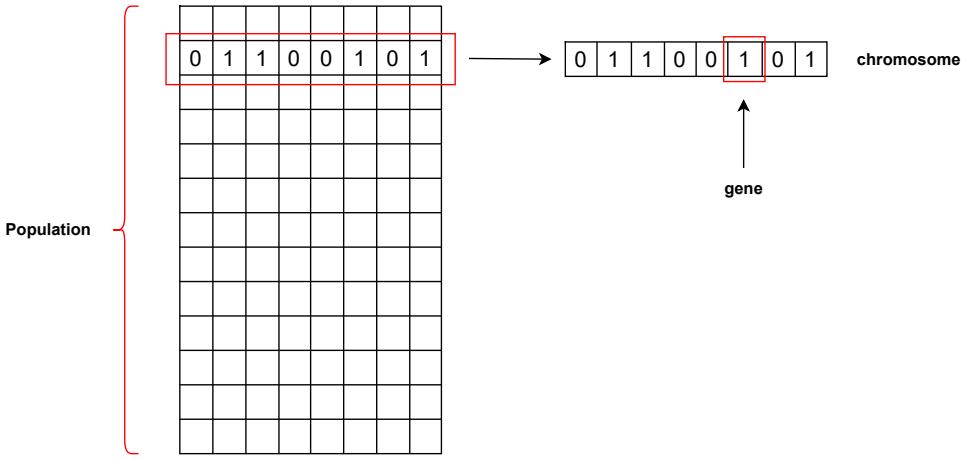
Φαινότυπος (Phenotype) Είναι κάθε συνδυασμός ενός γενοτύπου με την συνάρτηση ποιότητας και αντιστοιχεί στην αποκωδικοποιημένη πληροφορία.

Γενιά (Generation) Αποτελείται από έναν αριθμό ατόμων/λύσεων με κοινούς προγόνους και περίοδο δημιουργίας. Η πρώτη γενιά δημιουργείται με κάποιον ξεχωριστό μηχανισμό (συνήθως τυχαία) και συνεπώς τα άτομα που την απαρτίζουν δεν έχουν προγόνους. Ο αριθμός των γενεών δεν είναι σταθερός και εξαρτάται τόσο από τον ρυθμό της εξέλιξης όσο και από το κριτήριο τερματισμού, με αποτέλεσμα να διαφέρει ακόμα και μεταξύ διαφορετικών στιγμιοτύπων της ίδιας υλοποίησης καθώς πάντα υπάρχει τυχαιότητα που μπορεί να επηρεάσει την ταχύτητα σύγκλισης στην τελική λύση.

2.3.2.2 Βασικά Συστατικά

Όλοι οι γενετικοί αλγόριθμοι, ανεξάρτητα από τις πιθανές παραλλαγές και τις διαφορές που ενδέχεται να παρουσιάζουν μεταξύ τους, έχουν ορισμένα κοινά βασικά συστατικά τα οποία είναι απαραίτητα σε οποιαδήποτε μορφή υλοποίησης. Αυτά είναι [14]:

Πληθυσμός (Population) Είναι το σύνολο των ατόμων ανά γενιά κατά τη διαδικασία της εξέλιξης. Ο πληθυσμός παραμένει σταθερός κατά την εκτέλεση του ΓΑ.



Σχήμα 2.11: Δομή ατόμων ενός πληθυσμού

Συνάρτηση Ποιότητας (Fitness Function) Δίνει ένα μέτρο που υποδεικνύει πόσο "ορθά" προσεγγίζει μία πιθανή λύση την επιθυμητή. Συνήθως είναι κανονικοποιημένη στο διάστημα $[0, 1]$ με τη σύμβαση οι μεγαλύτερες τιμές να υποδηλώνουν αποτελεσματικότερη συμπεριφορά. Η συνάρτηση ποιότητας είναι νευραλγικής σημασίας για την αποδοτικότητα του γενετικού αλγορίθμου γιατί καθορίζει σε σημαντικό βαθμό ποια άτομα θα επιβιώσουν και θα αναπαραχθούν στις επόμενες γενιές.

Επιλογή (Selection) Η επιλογή των ατόμων-γονέων που θα συνδυαστούν για την παραγωγή των απογόνων μπορεί να γίνει με πολλούς διαφορετικούς τρόπους που θα εξεταστούν αναλυτικότερα παρακάτω. Σχεδόν σε όλες τις γνωστές μεθόδους όμως λαμβάνεται υπόψη η συνάρτηση ποιότητας, κάτι που σημαίνει ότι όσο μεγαλύτερος είναι ο *βαθμός καταλληλότητας* (fitness) ενός ατόμου τόσο πιθανότερο είναι να επιλεχθεί ως γονέας.

Διασταύρωση (Crossover) Είναι η διαδικασία κατά την οποία δύο γενότυποι-γονείς συνδυάζονται για την δημιουργία ενός γενοτύπου-απογόνου. Ο απόγονος κληρονομεί σε αυτό το στάδιο γονίδια και από του δύο γονείς με αναλογία που μπορεί να ποικίλλει ανάλογα με τη μέθοδο υλοποίησης. Το ποσοστό των γονιδίων που κληρονομούνται από κάθε γονέα μπορεί να διαφέρει ακόμα και σε διαφορετικές εφαρμογές της ίδιας μεθόδου καθώς συνήθως χρησιμοποιούνται πιθανοτικά μοντέλα. Στην περίπτωση που ο γενότυπος περιλαμβάνει χρωμοσώματα με διαφορετικό μήκος, διασταυρώνονται μεταξύ τους μόνο χρωμοσώματα που έχουν την ίδια σειρά στους γενότυπους-γονείς, δηλαδή τα χρωμοσώματα που κωδικοποιούν ουσιαστικά την ίδια παράμετρο του προβλήματος.

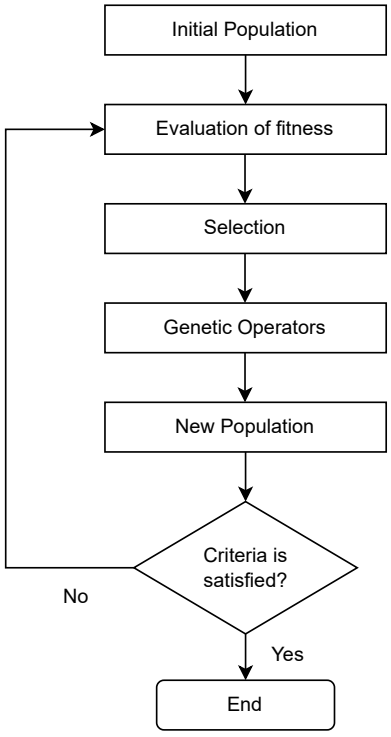
Μετάλλαξη (Mutation) Η μετάλλαξη είναι το τελικό στάδιο της αναπαραγωγής. Εκτελείται τυχαία σε κάποιους από τους απογόνους αμέσως μετά την ολοκλήρωση της διασταύρωσης και αλλάζει κάποια γονίδια τους. Τα γονίδια αυτά επιλέγονται κατά βάση τυχαία, ενώ το ποσοστό τους επί του συνόλου των γονιδίων του χρωμοσώματος ανήκει στο εύρος $[1\% - 10\%]$ έτσι ώστε η τυχαιότητα που εισάγεται να μην επηρεάσει αρνητικά την απόδοση του αλγορίθμου.

2.3.2.3 Υλοποίηση

Αφού κωδικοποιηθεί το πρόβλημα και αντιστοιχιστούν οι προς προσδιορισμό παράμετροι σε γονίδια και χρωμοσώματα, ακολουθείται η εξής διαδικασία [64]:

1. Το πρώτο βήμα είναι η αρχικοποίηση του πληθυσμού. Ο αλγόριθμος δημιουργεί (κατά βάση με τυχαίο τρόπο) την πρώτη γενιά του πληθυσμού αρχικοποιώντας γενότυπους με γονίδια τυχαίων τιμών.
2. Σε κάθε ένα άτομο της γενιάς εφαρμόζεται η συνάρτηση ποιότητας που του αντιστοιχίζει ένα μέτρο ανάλογα με το πόσο προσεγγίζει την επιθυμητή λύση.
3. Όταν εκτιμηθούν τα μέτρα όλων των λύσεων της γενιάς, ξεκινάει η αναπαραγωγή της επόμενης με την εφαρμογή κατά σειρά των τελεστών επιλογής, διασταύρωσης και μετάλλαξης.
4. Τα δύο παραπάνω βήματα επαναλαμβάνονται ώσπου να ικανοποιηθεί το κριτήριο τερματισμού του αλγορίθμου και όταν η διαδικασία ολοκληρωθεί επιστρέφεται η βέλτιστη λύση/γενότυπος που έχει παραχθεί μέχρι εκείνη τη στιγμή.

Η παραπάνω διαδικασία φαίνεται διαγραμματικά στο Σχήμα 2.12.



Σχήμα 2.12: Διαγραμματική υλοποίηση γενετικών αλγορίθμων

2.3.3 Τελεστές

Σε αυτό το τμήμα θα εξεταστούν αναλυτικά οι βασικοί *τελεστές* (operators) των γενετικών αλγορίθμων. Τρεις είναι οι κύριες κατηγορίες τελεστών:

1. Επιλογής (Selectors)
2. Διασταύρωσης (Crossover)
3. Μετάλλαξης (Mutation)

2.3.3.1 Τελεστές Επιλογής (Selectors)

2.3.3.1.1 Επιλογή Γονέων Η επιλογή των ατόμων που θα συνδυαστούν για την αναπαραγωγή των απογόνων είναι καθοριστική για τον γενετικό αλγόριθμο και έχει αποτελέσει πεδίο έρευνας για πολλά χρόνια. Ως εκ τούτου έχουν αναπτυχθεί πολλοί τελεστές επιλογής, χωρίς όμως να υπάρχει κάποιος αποδεδειγμένα πιο αποτελεσματικός για όλα τα προβλήματα. Φυσικά, βασική προϋπόθεση για όλες τις μεθόδους είναι να έχει εκτελεστεί πρώτα η συνάρτηση ποιότητας και να έχει υπολογιστεί ο βαθμός καταλληλότητας των ατόμων. Παρακάτω αναφέρονται οι σημαντικότεροι τελεστές επιλογής [4, 9].

Τελεστής Επιλογής Μόντε Κάρλο (Monte Carlo Selector) Η μέθοδος επιλογής Μόντε Κάρλο είναι η απλούστερη μέθοδος καθώς επιλέγει τα άτομα εντελώς τυχαία. Προφανώς η απόδοσή της είναι πολύ χαμηλή και για αυτό δεν εφαρμόζεται στην πράξη, μπορεί όμως να χρησιμοποιηθεί ως σημείο αναφοράς για τον έλεγχο και την αξιολόγηση άλλων μεθόδων.

Τελεστής Επιλογής Διαγωνισμού (Tournament Selector) Ο τελεστής επιλογής διαγωνισμού ταξινομεί ένα τυχαίο υποσύνολο των ατόμων του πληθυσμού με βάση το βαθμό καταλληλότητάς τους και επιλέγει το άτομο με τον υψηλότερο βαθμό. Έπειτα επαναλαμβάνει αυτή τη διαδικασία $n-1$ φορές ώστε να επιλεγούν τα n άτομα που θα χρησιμοποιηθούν για την αναπαραγωγή. Το μέγεθος των υποσυνόλων μπορεί να ποικίλλει και είναι μία παράμετρος που επίσης απαιτεί μελέτη, καθώς μπορεί να επηρεάσει το ποσοστό των πιο "αδύναμων" ατόμων που θα προκριθούν. Χαρακτηριστικό αυτής της μεθόδου είναι ότι ο γενότυπος με την υψηλότερη ποιότητα θα επιλεγεί σίγουρα και αυτός με τη χαμηλότερη θα απορριφθεί.

Τελεστής Επιλογής Περικοπής (Truncation Selector) Ο τελεστής επιλογής περικοπής έχει την ακόλουθη πολύ απλή λειτουργία. Αρχικά ταξινομεί όλα τα άτομα σε φθίνουσα σειρά με βάση το βαθμό καταλληλότητάς τους και στη συνέχεια επιλέγει τα n πρώτα. Η μέθοδος αυτή οδηγεί γρήγορα σε άτομα με υψηλή ποιότητα, όμως έχει το μειονέκτημα ότι μπορεί να αποκλείσει νωρίς χρήσιμα γονίδια αν τα χρωμοσώματα που τα περιέχουν δεν επιλεγούν και γι' αυτό δε χρησιμοποιείται συχνά.

Τελεστής Επιλογής Ρουλέτας (Roulette-Wheel Selector) Ο τελεστής επιλογής ρουλέτας είναι ένας από τους πιο ευρέως χρησιμοποιούμενους τελεστές επιλογής, ο οποίος ανήκει στην κατηγορία των πιθανοτικών μεθόδων επιλογής. Για κάθε ένα άτομο/λύση i, υπολογίζει την πιθανότητα:

$$P(i) = \frac{f_i}{\sum_{j=1}^{N} f_j} \tag{2.45}$$

όπου f_i είναι ο βαθμός καταλληλότητας του ατόμου i και N ο αριθμός των ατόμων του πληθυσμού και έπειτα δημιουργεί n τυχαίους αριθμούς στο διάστημα $[0, 1]$. Οι πιθανότητες που υπολογίστηκαν χρησιμοποιούνται για το σχηματισμό των εξής διαστημάτων:

$$[0, P(1)], [P(1), P(1) + P(2)], \ldots, [P(1) + P(2) + \ldots + P(n-1), P(1) + P(2) + \ldots + P(n) = 1].$$

Ανάλογα με το διάστημα στο οποίο ανήκει ο κάθε ένας από τους τυχαίους αριθμούς, επιλέγεται το αντίστοιχο άτομο για να αναπαραχθεί στην επόμενη γενιά.

Τελεστής Επιλογής Γραμμικής Κατάταξης (Linear Rank Selector) Ο τελεστής επιλογής γραμμικής κατάταξης είναι ένας, επίσης, πιθανοτικός τελεστής. Αρχικά ταξινομεί τα άτομα με βάση την ποιότητά τους και προσδίδει στο καθένα έναν *βαθμό* (rank) από 1 μέχρι N (όπου N το πλήθος των ατόμων), ξεκινώντας από το μικρότερο προς το μεγαλύτερο. Τελικά η πιθανότητα να επιλεγεί το i άτομο είναι:

$$P(i) = \frac{1}{N}\left(n^- + (n^+ - n^-) * \frac{\text{rank}(i) - 1}{N - 1}\right) \tag{2.46}$$

Η πιθανότητα επιλογής του καλύτερου ατόμου (δηλαδή του ατόμου με τον μεγαλύτερο βαθμό καταλληλότητας) είναι $\frac{n^+}{N}$ και του χειρότερου $\frac{n^-}{N}$. Η υλοποίηση του είναι παρόμοια με αυτή του τελεστή επιλογής ρουλέτας και βασίζεται στη δημιουργία τυχαίων αριθμών με μόνη διαφορά τις διαφορετικές πιθανότητες που αντιστοιχούν στα άτομα/γενότυπους.

Εκθετικός Τελεστής Επιλογής (Exponential Rank Selector) Στον εκθετικό τελεστή επιλογής τα άτομα επίσης ταξινομούνται σε φθίνουσα σειρά και αντιστοιχίζονται σε τιμές από N έως 1. Η πιθανότητα του ατόμου i να επιλεγεί είναι:

$$P(i) = \frac{c^{N-\text{rank}(i)}}{\sum_{j=1}^{N} c^{N-\text{rank}(j)}} \tag{2.47}$$

όπου c παράμετρος που ανήκει στο διάστημα $[0, 1)$ και ορίζεται από τον σχεδιαστή. Όσο η τιμή της παραμέτρου μειώνεται, αυξάνεται η πιθανότητα επιλογής του καλύτερου ατόμου ενώ όσο αυξάνεται, οι πιθανότητες όλων των ατόμων τείνουν να γίνουν ίσες.

Στοχαστικός Καθολικός Τελεστής Επιλογής (Stochastic Universal Selector) Ο στοχαστικός καθολικός τελεστής επιλογής είναι μία παραλλαγή του τελεστή ρουλέτας που στοχεύει στην πιο ομαλή επιλογή των απογόνων για να αποφευχθεί η απότομη εξέλιξη, η οποία έχει μεν ταχύτερη σύγκλιση αλλά αποκλείει γρήγορα λύσεις με χαμηλή ποιότητα που ενδεχομένως περιέχουν κάποια χρήσιμα γονίδια. Για την πραγματοποίηση της επιλογής γίνεται ταξινόμηση των ατόμων σε φθίνουσα σειρά με βάση την ποιότητά τους και σχηματίζονται τα διαστήματα:

$$[0, f(1)], [f(1), f(1) + f(2)], \ldots, [0 + f(1) + \ldots + f(n-1), 0 + f(1) + \ldots + f(n)].$$

Στη συνέχεια υπολογίζεται ο μέσος όρος μ των τιμών ποιότητας και δημιουργείται ένας τυχαίος αριθμός r στο διάστημα $[0, \mu]$. Σε αυτόν τον τυχαίο αριθμό προστίθεται ο μέσος όρος και ανάλογα με το διάστημα στο οποίο ανήκει το άθροισμά τους επιλέγεται το αντίστοιχο άτομο. Έπειτα προστίθεται ο μέσος όρος στο προηγούμενο άθροισμα και αυτή η διαδικασία επαναλαμβάνεται $n-1$ φορές ώστε να επιλεχθούν και τα υπόλοιπα $n-1$ άτομα. Τονίζεται ότι το πρώτο άτομο έχει επιλεχθεί ως αντίστοιχο του διαστήματος που περιλαμβάνει τον τυχαίο αριθμό r και εφόσον $r \leq \mu < f(1)$, το άτομο με το μεγαλύτερο βαθμό καταλληλότητας, δηλαδή το πρώτο άτομο μετά την ταξινόμηση, επιλέγεται πάντα.

2.3.3.1.2 Επιλογή Επιζώντων Κάθε νέα γενιά του ΓΑ αποτελείται από άτομα-απογόνους που προκύπτουν από τον συνδυασμό ατόμων-γονέων της προηγούμενης γενιάς καθώς και από άτομα-επιζώντες τα οποία περνούν αμετάβλητα από την προηγούμενη γενιά στην επόμενη. Ο τρόπος με τον οποίο γίνεται η επιλογή των επιζώντων της κάθε γενιάς μπορεί επίσης να ποικίλλει και είναι εξίσου κομβικός για την αποδοτικότητα του αλγορίθμου. Δύο είναι οι κυριότερες κατηγορίες μεθόδων επιλογής επιζώντων [50]:

Επιλογή βασισμένη στην Ηλικία (Age-based Selection) Με αυτή τη μέθοδο προκρίνονται στην επόμενη γενιά τα άτομα/λύσεις με τη μικρότερη ηλικία, δηλαδή αυτά που έχουν περάσει αυτούσια στις λιγότερες γενιές σε σχέση με τα υπόλοιπα. Ο βαθμός καταλληλότητας των ατόμων δε λαμβάνεται υπόψη.

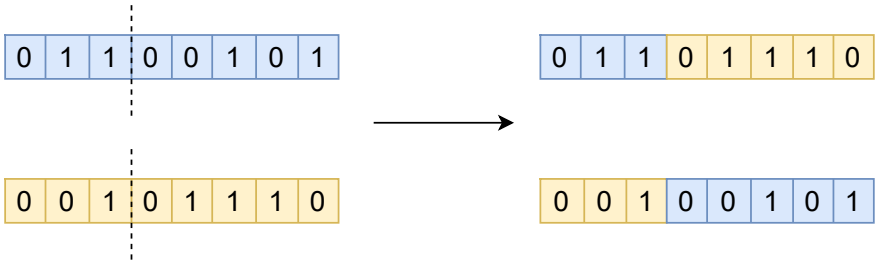
Επιλογή βασισμένη στο Βαθμό Καταλληλότητας (Fitness-based Selection) Είναι η πιο συχνά χρησιμοποιούμενη μέθοδος και στηρίζει την επιλογή των επιζώντων στο βαθμό καταλληλότητάς τους. Συνήθως ακολουθεί την αρχή του ελιτισμού κατά την οποία το άτομο με τον υψηλότερο βαθμό καταλληλότητας επιβιώνει πάντα στην επόμενη γενιά, ώστε να εξασφαλιστεί ότι αυτή δε θα είναι ποτέ χειρότερη από την προηγούμενη. Για την επιλογή των υπόλοιπων ατόμων μπορεί να χρησιμοποιηθεί οποιαδήποτε από τις τεχνικές που περιγράφηκαν παραπάνω και για την επιλογή των γονέων.

2.3.3.2 Τελεστές Διασταύρωσης (Crossover Operators)

Παρακάτω παρουσιάζονται οι πιο σημαντικές τεχνικές διασταύρωσης για χρωμοσώματα με δυαδικά και με αριθμητικά γονίδια [94].

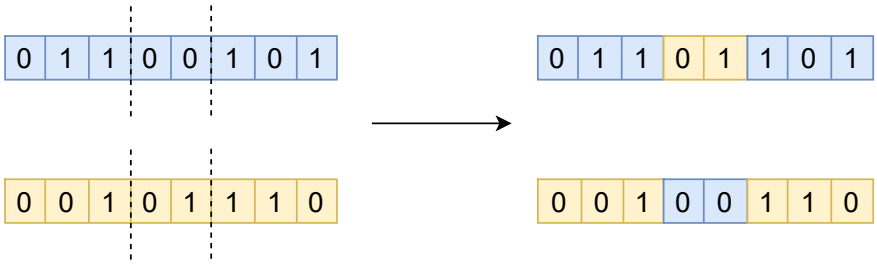
2.3.3.2.1 Δυαδικά Γονίδια

Διασταύρωση Ενός Σημείου (Single-point Crossover) Σε αυτή την τεχνική διασταύρωσης επιλέγεται τυχαία ένα σημείο μέσα στο χρωμόσωμα και το χρωμόσωμα-απόγονος κληρονομεί όλα τα γονίδια του ενός χρωμοσώματος-γονέα μέχρι αυτό το σημείο και όλα τα γονίδια του άλλου γονέα από αυτό το σημείο και έπειτα. Το μειονέκτημα αυτής της τεχνικής είναι ότι καθυστερεί αρκετά την εξέλιξη και πρέπει να αναπτυχθούν πολλές γενιές για να προκύψουν άτομα-λύσεις που συνδυάζουν πολλά διαφορετικά γονίδια.



Σχήμα 2.13: Διασταύρωση ενός σημείου

Διασταύρωση Πολλαπλών Σημείων (Multi-point Crossover) Επιλέγεται ένας αριθμός από σημεία μέσα στο χρωμόσωμα (συνήθως δύο) και στα διαστήματα που ορίζονται από αυτά τα σημεία ο απόγονος κληρονομεί ακολουθίες γονιδίων εναλλάξ από τα δύο χρωμοσώματα-γονείς. Επί της ουσίας, η διασταύρωση ενός σημείου που περιγράφηκε παραπάνω είναι υποπερίπτωση της διασταύρωσης πολλαπλών σημείων στην οποία επιλέγεται ένα μόνο σημείο.



Σχήμα 2.14: Διασταύρωση πολλαπλών σημείων

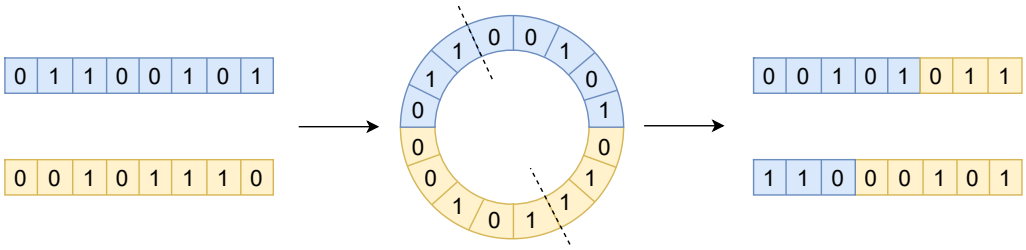
Ομοιόμορφη Διασταύρωση (Uniform Crossover) Είναι μία τεχνική παρόμοια με τη διασταύρωση πολλαπλών σημείων, με τη διαφορά ότι δεν υπάρχουν συγκεκριμένα διαστήματα στα οποία κληρονομούνται γονίδια από τον κάθε γονέα αλλά είναι συγκεκριμένη η αναλογία. Αυτό πρακτικά σημαίνει ότι τα γονίδια των δύο χρωμοσωμάτων-γονέων εξετάζονται ένα προς ένα και κληρονομείται με πιθανότητα p το γονίδιο του ενός γονέα και με πιθανότητα $1-p$ το γονίδιο του άλλου. Η πιθανότητα p είναι τέτοια ώστε να κληρονομείται τελικά το επιθυμητό ποσοστό γονιδίων από τον κάθε γονέα.

Ημιομοιόμορφη Διασταύρωση (Half-uniform Crossover) Αποτελεί υποπερίπτωση της ομοιόμορφης διασταύρωσης στην οποία 50% των γονιδίων κληρονομούνται από τον ένα γονέα και 50% από τον άλλο.

Διασταύρωση Ανάμειξης (Shuffle Crossover) Πρόκειται για μία παραλλαγή της διασταύρωσης ενός σημείου με στόχο να εξαλειφθεί η εξάρτηση από τη θέση του κάθε γονιδίου μέσα στο χρωμόσωμα, η οποία παίζει σημαντικό ρόλο στην κλασική υλοποίηση που περιγράφηκε παραπάνω. Αυτή η τεχνική υλοποιείται σε τρία βήματα. Αρχικά αναδιατάσσει τυχαία τα γονίδια στα χρωμοσώματα των γονέων, πραγματοποιώντας όμως την ίδια αναδιάταξη και στους δύο γονείς ώστε τα γονίδια που βρίσκονται στην ίδια θέση μέσα στο χρωμόσωμα να αντιστοιχούν πάντα στην ίδια κωδικοποιημένη παράμετρο του προβλήματος. Έπειτα εκτελεί στα αναδιατεταγμένα χρωμοσώματα διασταύρωση ενός σημείου και τελικά χρησιμοποιεί την αντίστροφη αναδιάταξη για να επαναφέρει τα γονίδια στην αρχική τους θέση.

Διασταύρωση Μειωμένης Αναπλήρωσης (Reduced Sarrogate Crossover) Άλλη μία παραλλαγή της μεθόδου διασταύρωσης ενός σημείου. Σε αυτή τη μέθοδο στόχος είναι να αποφευχθεί η διασταύρωση σε σημεία που τα γονίδια των δύο γονέων είναι ίδια, καθώς είναι περιττή. Η υλοποίηση της πραγματοποιείται σε δύο στάδια. Στο πρώτο στάδιο γίνεται σύγκριση όλων των γονιδίων των γονέων ένα προς ένα ώστε να αποκλειστούν τα σημεία των χρωμοσωμάτων όπου τα δύο γονίδια είναι ίδια και στο δεύτερο επιλέγεται τυχαία ένα από τα σημεία στα οποία τα γονίδια είναι διαφορετικά και εκτελείται διασταύρωση ενός σημείου.

Διασταύρωση Δακτυλίου (Ring Crossover) Στη διασταύρωση δακτυλίου τα χρωμοσώματα των γονέων αρχικά ενώνονται σχηματίζοντας ένα ενιαίο χρωμόσωμα σε μορφή δακτυλίου, με τέτοιο τρόπο ώστε τα πρώτα γονίδια τους να είναι διαδοχικά όπως και τα τελευταία (Σχήμα 2.15). Ακολούθως, επιλέγεται ένα τυχαίο σημείο του δακτυλίου στο οποίο γίνεται η πρώτη τομή και το αντιδιαμετρικό του για τη δεύτερη, ώστε τα δύο χρωμοσώματα που θα προκύψουν από τον διαχωρισμό να έχουν το ίδιο μήκος, το οποίο είναι φυσικά ίδιο με το μήκος των χρωμοσωμάτων-γονέων.



Σχήμα 2.15: Διασταύρωση δακτυλίου

2.3.3.2.2 Αριθμητικά Γονίδια Σε αυτό το σημείο περιγράφονται οι κλασικές τεχνικές διασταύρωσης για αριθμητικά γονίδια [98, 3].

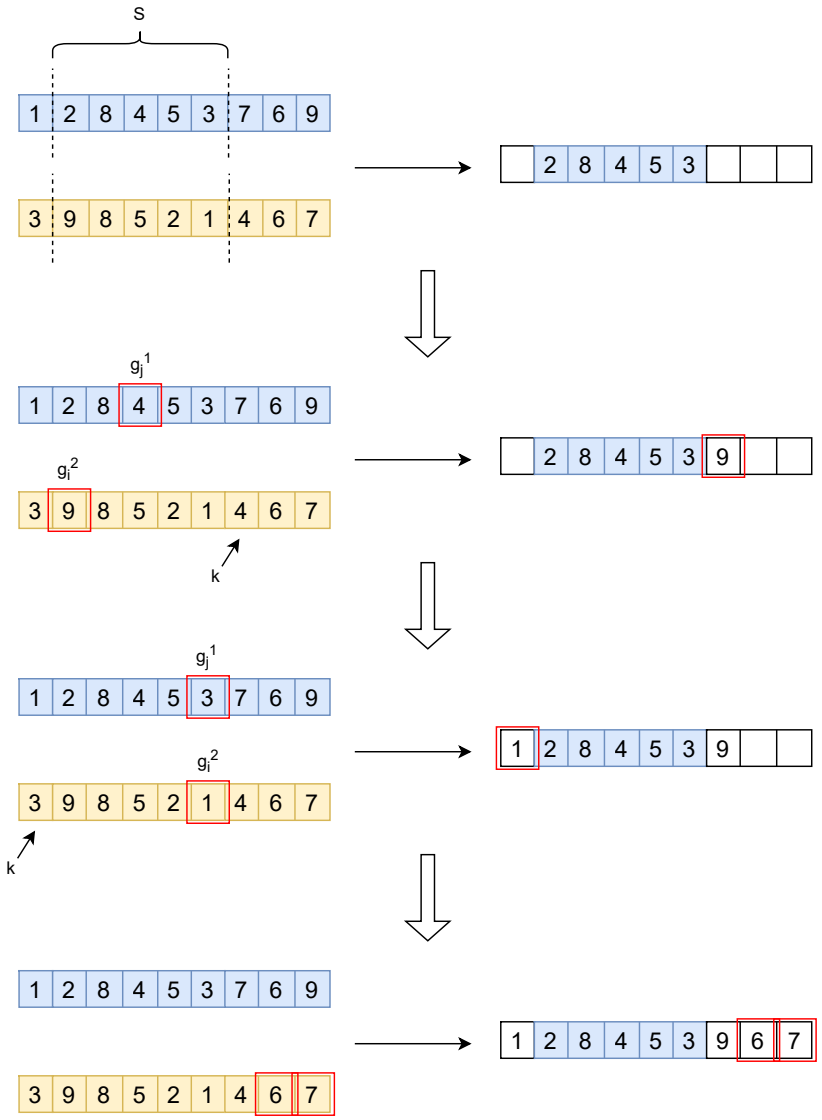
Διασταύρωση Μέσου Όρου (Average Crossover) Είναι μία από τις απλούστερες τεχνικές διασταύρωσης για γενότυπους που περιλαμβάνουν αριθμητικά γονίδια. Στη διασταύρωση μέσου όρου κάθε γονίδιο g_i του απογόνου παίρνει ως τιμή τον μέσο όρο των τιμών των αντίστοιχων γονιδίων g_i^1 και g_i^2 των γονέων του.

Επίπεδη Διασταύρωση (Flat Crossover) Στην τεχνική επίπεδης διασταύρωσης, για κάθε γονίδιο του απογόνου δημιουργείται ένας τυχαίος αριθμός στο διάστημα που ορίζεται από τις τιμές των δύο γονιδίων των γονέων (της αντίστοιχης θέσης μέσα στο χρωμόσωμα).

Διασταύρωση Μερικής Αντιστοίχισης (Partially Matched Crossover - PMX) Είναι μία από τις πιο διαδεδομένες μεθόδους διασταύρωσης για αριθμητικά γονίδια όπου απαιτείται όλα τα γονίδια του κάθε χρωμοσώματος να είναι διαφορετικά μεταξύ τους. Αρχικά, επιλέγονται δύο τυχαία σημεία στα χρωμοσώματα των γονέων και τα γονίδια που βρίσκονται ανάμεσα σε αυτά αντιγράφονται απευθείας από τον πρώτο γονέα στο χρωμόσωμα του απογόνου. Κατόπιν ελέγχονται ένα προς ένα τα γονίδια του δεύτερου γονέα που ανήκουν στο ίδιο διάστημα, έστω S, και για όσα από αυτά δεν έχουν περάσει ήδη στον απόγονο (δηλαδή δεν υπήρχαν γονίδια στο S με αυτή την τιμή στο χρωμόσωμα του πρώτου γονέα) ακολουθείται η παρακάτω διαδικασία.

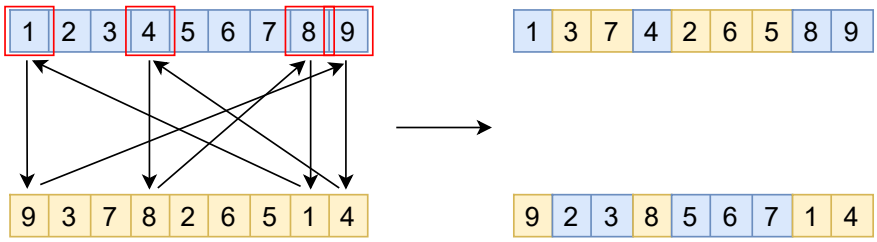
Το γονίδιο g_i^2 του δεύτερου γονέα αντιστοιχίζεται στο πρώτο γονίδιο g_j^1 του πρώτου γονέα που ανήκει στο S στο χρωμόσωμα του πρώτου γονέα αλλά δεν ανήκει στο S στο χρωμόσωμα του δεύτερου. Έπειτα εντοπίζεται η θέση k του γονιδίου του δεύτερου γονέα που έχει την ίδια τιμή με το g_j^1, και αν το k δεν ανήκει στο διάστημα S, η τιμή του γονιδίου g_i^2 αντιγράφεται στη θέση k του χρωμοσώματος του απογόνου. Στην περίπτωση που το k ανήκει στο S, επαναλαμβάνεται η προηγούμενη διαδικασία για το γονίδιο g_k^2 μέχρι να βρεθεί θέση n στο χρωμόσωμα του δεύτερου γονέα η οποία δεν ανήκει στο S. Σημειώνεται ότι όσες φορές και αν χρειαστεί να επαναληφθεί αυτό το βήμα, το γονίδιο που τελικά θα αντιγραφεί στο χρωμόσωμα του απογόνου είναι αυτό που είχε επιλεχθεί αρχικά από το διάστημα

S του δεύτερου γονέα δηλαδή το g_i^2. Αφού ολοκληρωθεί αυτή η διαδικασία, τα υπόλοιπα γονίδια αντιγράφονται αυτούσια στις αντίστοιχες θέσεις γονιδίων του απογόνου.



Σχήμα 2.16: Διασταύρωση μερικής αντιστοίχισης

Κυκλική Διασταύρωση (Cycle Crossover) Η κυκλική διασταύρωση είναι μία τεχνική που επίσης εξασφαλίζει ότι δεν υπάρχουν δύο ίδια γονίδια στο χρωμόσωμα-απόγονο. Η λειτουργία της είναι αρκετά απλή. Ένα γονίδιο g_i^1 του χρωμοσώματος του πρώτου γονέα αντιστοιχίζεται στο γονίδιο του χρωμοσώματος του δεύτερου γονέα που βρίσκεται στην ίδια θέση g_i^2. Το γονίδιο του δεύτερου γονέα g_i^2 αντιστοιχίζεται στο γονίδιο g_j^1 του πρώτου γονέα που έχει την ίδια τιμή. Η διαδικασία αυτή ξεκινάει από το πρώτο γονίδιο του πρώτου γονέα g_1^1 και ολοκληρώνεται όταν βρεθεί το γονίδιο g_n^2 του δεύτερου γονέα με την ίδια τιμή. Τελικά, όσα γονίδια προσπελάστηκαν κατά τη διάρκεια της παραπάνω διαδικασίας αντιγράφονται στον απόγονο από τον πρώτο γονέα, ενώ τα υπόλοιπα από τον δεύτερο.



Σχήμα 2.17: Κυκλική διασταύρωση

Εκτός από αυτές τις τεχνικές, μπορούν να χρησιμοποιηθούν και οι τεχνικές που ισχύουν για τα δυαδικά χρωμοσώματα με την κατάλληλη προσαρμογή.

2.3.3.3 Τελεστές Μετάλλαξης (Mutation Operators)

Στόχος της μετάλλαξης είναι να εισαχθεί το στοιχείο της τυχαιότητας στην επόμενη γενιά ώστε να γίνει εξερεύνηση σε μεγαλύτερο χώρο καταστάσεων. Ωστόσο, η πιθανότητα με την οποία ένα άτομο θα υποστεί μετάλλαξη πρέπει να είναι αρκετά μικρή, διαφορετικά τα άτομα-λύσεις θα καταλήξουν να κινούνται σε έναν τυχαίο χώρο καταστάσεων και η λειτουργία του αλγορίθμου θα εκφυλιστεί. Για τον ίδιο λόγο είναι μικρός και ο αριθμός των γονιδίων του επιλεγμένου χρωμοσώματος που αλλάζουν κάθε φορά.

Για να είναι η διαδικασία της μετάλλαξης αποδεκτή, πρέπει να πληροί ορισμένες προϋποθέσεις που εξασφαλίζουν ότι δε θα αλλοιωθούν οι βασικές αρχές ενός γενετικού αλγορίθμου. Κατ' αρχάς, πρέπει όλες οι πιθανές καταστάσεις του υπό εξερεύνηση χώρου να είναι προσπελάσιμες ανά πάσα στιγμή έστω και με πολύ μικρή πιθανότητα. Σε περίπτωση που η μετάλλαξη εισάγει περιορισμούς που δεν ικανοποιούν αυτή τη συνθήκη, υπάρχει το ενδεχόμενο να μη βρεθεί ποτέ η βέλτιστη λύση καθώς μπορεί να έχει αποκλειστεί από το χώρο αναζήτησης. Η συγκεκριμένη περίπτωση απαιτεί την ορθή κατανόηση του προβλήματος και την επιλογή της κατάλληλης μεθόδου μετάλλαξης από την αρχή καθώς δε γίνεται εύκολα ανιχνεύσιμη κατά την εκτέλεση του αλγορίθμου ή μετά την ολοκλήρωσή του.

Η δεύτερη σημαντική προϋπόθεση είναι η μετάλλαξη να μην ευνοεί την αναζήτηση προς μία συγκεκριμένη κατεύθυνση σε προβλήματα που δεν υπάρχουν αντίστοιχοι περιορισμοί. Αυτό γίνεται πιο εύκολα αντιληπτό αν καθώς εξελίσσεται ο αλγόριθμος προκύπτουν πολλές παρόμοιες λύσεις οι οποίες δεν προσεγγίζουν την επιθυμητή, κάτι που όμως δεν εγγυάται προβληματική μετάλλαξη καθώς μπορεί να οφείλεται και σε άλλους παράγοντες.

Η μετάλλαξη μπορεί να υλοποιηθεί με διαφορετικές τεχνικές ανάλογα με τη φύση του προβλήματος, οι βασικότερες από τις οποίες παρουσιάζονται στη συνέχεια [94, 55, 24].

Μετάλλαξη Αντιστροφής Ψηφίου (Bit Flip Mutation) Είναι μία μέθοδος μετάλλαξης για προβλήματα με δυαδική κωδικοποίηση. Ένας μικρός αριθμός από bits επιλέγεται τυχαία μέσα στο χρωμόσωμα και αλλάζουν στο συμπλήρωμα τους $(0 \Rightarrow 1, 1 \Rightarrow 0)$. Συνήθως το κάθε bit i έχει πιθανότητα να μεταλλαχθεί $P(i) = \frac{1}{N}$ σε ένα χρωμόσωμα με N bits.

Ομοιόμορφη Μετάλλαξη (Uniform Mutation (Random Resseting)) Βασίζεται στη λογική της μετάλλαξης αντιστροφής ψηφίου, την οποία επεκτείνει για προβλήματα με

Σχήμα 2.18: Μετάλλαξη αντιστροφής ψηφίου

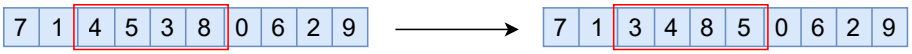
κωδικοποίηση ακέραιων αριθμών. Σε αυτή την περίπτωση, τα επιλεγμένα γονίδια παίρνουν τυχαία μία από τις δυνατές τιμές που έχουν οριστεί κατά την κωδικοποίηση.

Μετάλλαξη Ορίων (Boundary Mutation) Χρησιμοποιείται σε προβλήματα με αριθμητική (ακέραια ή πραγματική) κωδικοποίηση και αντικαθιστά το γονίδιο προς μετάλλαξη είτε με τη μικρότερη είτε με τη μεγαλύτερη δυνατή τιμή τυχαία.

Μετάλλαξη Ανταλλαγής (Swap Mutation) Σε αυτή τη μέθοδο γίνεται ανταλλαγή δύο τυχαίων γονιδίων μέσα στο χρωμόσωμα. Εφαρμόζεται σε προβλήματα με *κωδικοποίηση μετάθεσης* (permutation encoding) όπου αναζητείται η βέλτιστη σειρά των γονιδίων μέσα στο χρωμόσωμα.

Σχήμα 2.19: Μετάλλαξη ανταλλαγής

Μετάλλαξη Αναδιάταξης (Scramble Mutation) Εφαρμόζεται επίσης σε προβλήματα με κωδικοποίηση μετάθεσης και είναι παρόμοια με τη μετάλλαξη ανταλλαγής με τη διαφορά ότι αντί για δύο γονίδια επιλέγεται ένα μπλοκ και τα γονίδια που περιλαμβάνονται σε αυτό αναδιατάσσονται τυχαία.



Σχήμα 2.20: Μετάλλαξη αναδιάταξης

Αντίστροφη Μετάλλαξη (Inverse Mutation) Είναι μία υποπερίπτωση της μετάλλαξης αναδιάταξης, στην οποία η σειρά των γονιδίων του επιλεγμένου μπλοκ δεν αλλάζει τυχαία αλλά αντιστρέφεται.

Μετάλλαξη Εισαγωγής (Insertion Mutation) Επιλέγεται ένα γονίδιο από το χρωμόσωμα και μετατίθεται σε μία άλλη, τυχαία θέση μετακινώντας και όλα τα γονίδια που βρίσκονται μεταξύ της αρχικής και της τελικής του θέσης μία θέση δεξιά ή αριστερά, ανάλογα με την κατεύθυνση της κίνησης του.

Μετάλλαξη Μετατόπισης (Displacement Mutation) Λειτουργεί όπως η μετάλλαξη εισαγωγής μετακινώντας ένα μπλοκ γονιδίων.

2.3.4 Κριτήρια Τερματισμού

Όπως έχει ήδη αναφερθεί, το μέγεθος του πληθυσμού παραμένει σταθερό σε όλες τις γενιές και δε μειώνεται κατά την εξέλιξη ενός γενετικού αλγορίθμου. Αυτό σημαίνει ότι ο αλγόριθμος μπορεί να συνεχίσει να εκτελείται επ' άπειρον αν δεν οριστεί κάποια συνθήκη που όταν ικανοποιηθεί να τον τερματίζει. Αυτές οι συνθήκες ονομάζονται κριτήρια τερματισμού και ανάλογα με το πρόβλημα μπορούν να έχουν διαφορετική υλοποίηση. Επίσης μπορούν να οριστούν περισσότερα από ένα κριτήρια τερματισμού, οπότε ο ΓΑ θα ολοκληρωθεί όταν ικανοποιηθεί το πρώτο από αυτά.

Τα πιο δημοφιλή κριτήρια τερματισμού είναι τα ακόλουθα [48]:

Προκαθορισμένης Γενιάς Είναι το απλούστερο κριτήριο τερματισμού το οποίο ικανοποιείται όταν παραχθεί ένας συγκεκριμένος αριθμός γενιών που έχει οριστεί ως κατώφλι. Συνήθως εφαρμόζεται συνδυαστικά μαζί με κάποιο άλλο κριτήριο, το οποίο ενδέχεται να μη μπορέσει να ικανοποιηθεί, για να εξασφαλιστεί ότι ο αλγόριθμος θα τερματίσει σίγουρα.

Σταθερού Βαθμού Καταλληλότητας Συνήθως στα πρώτα στάδια ενός ΓΑ παρατηρείται σημαντική αύξηση του βαθμού καταλληλότητας των ατόμων/λύσεων από γενιά σε γενιά, η οποία μειώνεται σταδιακά καθώς οι γενιές αυξάνονται. Πολλές φορές, μάλιστα, από ένα σημείο και μετά ο βαθμός καταλληλότητας της βέλτιστης λύσης σταματάει να βελτιώνεται και παραμένει σταθερός. Το κριτήριο σταθερού βαθμού καταλληλότητας αντιμετωπίζει αυτό το φαινόμενο τερματίζοντας τον αλγόριθμο αν ο μέγιστος βαθμός παραμένει σταθερός για έναν συγκεκριμένο αριθμό συνεχόμενων γενιών, θεωρώντας ότι δεν υπάρχουν άλλα περιθώρια βελτίωσης και η καλύτερη δυνατή λύση έχει ήδη βρεθεί.

Χρονικού Ορίου Εξέλιξης Σε αυτή τη μέθοδο ορίζεται ένα χρονικό όριο το οποίο όταν ξεπεραστεί σηματοδοτεί την ολοκλήρωση της εκτέλεσης του ΓΑ. Αυτό το κριτήριο εφαρμόζεται σε χρονικά ευαίσθητα περιβάλλοντα όπου είναι αναγκαία η εύρεση λύσης μέσα σε ένα συγκεκριμένο χρονικό πλαίσιο.

Ορίου Βαθμού Καταλληλότητας Τερματίζει τον αλγόριθμο όταν ξεπεραστεί ένα ελάχιστο ή μέγιστο κατώφλι βαθμού καταλληλότητας, για προβλήματα ελαχιστοποίησης και μεγιστοποίησης αντίστοιχα, το οποίο έχει οριστεί στην αρχή. Για την αποτελεσματική εφαρμογή αυτού του κριτηρίου είναι απαραίτητο να είναι εκ των προτέρων γνωστό το αναμενόμενο διάστημα στο οποίο θα ανήκει η βέλτιστη λύση, διαφορετικά η τελική λύση δε θα προσεγγίζει την επιθυμητή.

Σύγκλισης Πληθυσμού Ο αλγόριθμος ολοκληρώνεται όταν σε μία γενιά ο μέσος όρος του βαθμού καταλληλότητας όλων των ατόμων/λύσεων απέχει από τον μέγιστο, λιγότερο από μία προκαθορισμένη τιμή.

Μέγιστου-Ελαχίστου Είναι μία μέθοδος που επίσης εξετάζει το σύνολο του πληθυσμού της κάθε γενιάς και τερματίζει τον αλγόριθμο αν η διαφορά μεταξύ της καλύτερης και της χειρότερης λύσης είναι μικρότερη από ένα επιλεγμένο κατώφλι.

2.3.5 Βασικά Χαρακτηριστικά – Πλεονεκτήματα

Τα βασικότερα χαρακτηριστικά των γενετικών αλγορίθμων είναι τα εξής:

1. Επιλύουν δύσκολα προβλήματα για τα οποία δεν υπάρχει γνωστή αναλυτική μέθοδος. Οι κυριότερες εφαρμογές των ΓΑ είναι σε πεδία με πολλές διαστάσεις/παραμέτρους που καθιστούν μη αποδοτική ή ακόμα και υπολογιστικά αδύνατη την εφαρμογή πολλών άλλων μεθόδων που χρησιμοποιούνται για την επίλυση τέτοιας φύσης προβλημάτων.
2. Μπορούν να εφαρμοστούν σε πολύ μεγάλο εύρος προβλημάτων. Σχεδόν οποιοδήποτε πρόβλημα μπορεί να προσεγγιστεί από έναν γενετικό αλγόριθμο με την κατάλληλη κωδικοποίηση. Φυσικά αυτό δε σημαίνει ότι οι ΓΑ αποτελούν βέλτιστη επιλογή για κάθε είδος προβλήματος καθώς ανάλογα με τις εκάστοτε απαιτήσεις είναι πιθανό να υπάρχουν πιο αποτελεσματικές μέθοδοι, ωστόσο σε αυτές τις περιπτώσεις μπορούν να χρησιμοποιηθούν ως μέτρο σύγκρισης.
3. Ενδείκνυνται για παράλληλη υλοποίηση. Η φύση των ΓΑ προσφέρει τη δυνατότητα εκμετάλλευσης της παραλληλίας που διαθέτουν οι σύγχρονες υπολογιστικές μηχανές στο μέγιστο, καθώς σε κάθε γενιά εξετάζεται ένα σύνολο ατόμων που είναι ανεξάρτητα μεταξύ τους και συνεπώς μπορούν να εκτελούνται παράλληλα οι απαραίτητοι υπολογισμοί.
4. Εξερευνούν ταυτόχρονα πολλές διαφορετικές κατευθύνσεις του χώρου αναζήτησης. Αυτό είναι ένα σημαντικό πλεονέκτημα των ΓΑ έναντι των περισσότερων μεθόδων επίλυσης παρόμοιων προβλημάτων που εξερευνούν λύσεις προς μία συγκεκριμένη κατεύθυνση σε κάθε στάδιο της εκτέλεσης.
5. Χρησιμοποιούν πιθανοτικές μεθόδους σε διάφορα σημεία της υλοποίησής τους. Με αυτόν τον τρόπο καταλήγουν πιο εύκολα σε πιθανές λύσεις που δε θα ήταν τόσο προφανείς αν βασίζονταν μόνο σε ντετερμινιστικές τεχνικές χωρίς να υπάρχει το στοιχείο της τυχαιότητας.
6. Είναι ευέλικτοι, δηλαδή προσαρμόζονται σχετικά εύκολα στα διάφορα *προγραμματιστικά περιβάλλοντα* (frameworks). Μόνο η συνάρτηση κόστους απαιτεί ένα βαθμό "συγχώνευσης" με το δεδομένο σύστημα προκειμένου να οριστεί κατάλληλα και κατά συνέπεια δεν χρειάζονται ιδιαίτερα σημαντικές αλλαγές σε ένα προσχεδιασμένο περιβάλλον ώστε να εφαρμοστεί ένας ΓΑ.
7. Είναι επεκτάσιμοι. Συνήθως οι ΓΑ επιδέχονται αρκετές αλλαγές ώστε να προσαρμοστούν στα διάφορα προβλήματα χωρίς αυτό να μειώνει την αποτελεσματικότητά τους. Μάλιστα, λόγω της συχνής χρήσης τους σε πληθώρα προβλημάτων, δεκάδες παραλλαγές έχουν προκύψει τόσο ως προς τη μορφή της κωδικοποίησης όσο και ως προς τη διαδικασία που ακολουθείται κατά τη διάρκεια της εξέλιξης.
8. Μπορούν να συνδυαστούν χωρίς προβλήματα με άλλες τεχνικές. Σε προβλήματα για τα οποία υπάρχουν πιο αποτελεσματικές μέθοδοι επίλυσης, οι ΓΑ πολλές φορές χρησιμεύουν έμμεσα, σχηματίζοντας υβριδικά συστήματα. Μία αρκετά συνηθισμένη υβριδική τεχνική αποτελεί ο συνδυασμός γενετικών αλγορίθμων με τεχνητά νευρωνικά δίκτυα, όπου οι ΓΑ εκτελούνται για την εύρεση μίας επιθυμητής λύσης, η οποία χρησιμοποιείται στη συνέχεια για την εκπαίδευση των δικτύων.

2.3.6 Γενετικοί Αλγόριθμοι και Ηλεκτρονικά Παιχνίδια

Η μεγαλύτερη πρόκληση στα *ηλεκτρονικά παιχνίδια* (video games) είναι η δημιουργία ενός ευφυούς συστήματος ελέγχου των χαρακτήρων, εχθρών ή και συμπαικτών, από τον υπολογιστή. Για να

καταστεί αυτό εφικτό, την τελευταία δεκαετία άρχισε να εισάγεται στα ηλεκτρονικά παιχνίδια τεχνητή νοημοσύνη, με τέτοιο ρυθμό μάλιστα ώστε πλέον θεωρείται δεδομένη σε όλα σχεδόν τα παιχνίδια. Σαν αποτέλεσμα, σήμερα υπάρχει πολύ μεγάλη ποικιλία μεθόδων εφαρμογής της προκειμένου να είναι τα παιχνίδια πιο ελκυστικά.

Η εφαρμογή της τεχνητής νοημοσύνης σε ένα ηλεκτρονικό παιχνίδι εστιάζει σε τρεις βασικούς άξονες οι οποίοι είναι η ικανότητα κίνησης των χαρακτήρων, η επιλογή της θέσης της κίνησης και η δυνατότητα στρατηγικής σκέψης. Καθώς η συμπεριφορά των χαρακτήρων εξαρτάται από ένα μεγάλο πλήθος αλληλοσυνδεόμενων παραμέτρων, ο καθορισμός των οποίων δεν μπορεί να γίνει χειροκίνητα, αυτοματοποιημένες μέθοδοι απαιτούνται για τον προσδιορισμό της [62, 13].

Μία τέτοια μέθοδος είναι η υλοποίηση ενός κατάλληλα προσαρμοσμένου γενετικού αλγορίθμου. Οι ΓΑ προσεγγίζουν τις ζητούμενες παραμέτρους με την τεχνική *δοκιμής-λάθους* (trial and error), εξετάζοντας πολλούς πληθυσμούς από πιθανές λύσεις και απορρίπτοντας τις λιγότερο αποτελεσματικές μέχρι να καταλήξουν στη βέλτιστη. Το μεγάλο πλεονέκτημά τους είναι ότι καταλήγουν σε διαφορετική λύση όταν αντιμετωπίζουν διαφορετικούς παίχτες, προσαρμόζοντας ουσιαστικά τη συμπεριφορά των ελεγχόμενων από τον υπολογιστή χαρακτήρων στον τρόπο παιχνιδιού του κάθε παίχτη-ανθρώπου. Επομένως η επιλογή των ΓΑ ενδείκνυται σε παιχνίδια που η συμπεριφορά του πραγματικού παίχτη μπορεί να αλλάζει σε μεγάλο βαθμό, απαιτώντας την αντίστοιχη ανταπόκριση από τον παίχτη-υπολογιστή [11].

Άλλο ένα στοιχείο που όταν υπάρχει ενισχύει την επιλογή γενετικών αλγορίθμων είναι η δυσκολία πρόβλεψης της συμπεριφοράς των παιχτών. Για να είναι ένας χαρακτήρας (συνήθως αντίπαλος) ικανοποιητικός πρέπει να είναι σε θέση να προσχεδιάσει τις κινήσεις του ανάλογα με τις κινήσεις του παίχτη. Σε πολύπλοκα παιχνίδια που δεν είναι δυνατόν να προβλεφθούν όλες οι πιθανές ενέργειες από τους σχεδιαστές, οι ΓΑ αποτελούν μία εναλλακτική μέθοδο.

Φυσικά, οι ΓΑ δε συνιστούν την ιδανικότερη επιλογή για την εισαγωγή τεχνητής νοημοσύνης σε όλα τα παιχνίδια, καθώς ανάλογα με τη φύση του κάθε παιχνιδιού υπάρχουν και αντίστοιχες τεχνικές που είναι αποδοτικότερες. Σε κάθε περίπτωση είναι αναγκαία η ορθή κατανόηση του εκάστοτε περιβάλλοντος, προκειμένου να επιλεχθεί η καταλληλότερη μέθοδος.

2.4 Δενδρική Αναζήτηση Μόντε Κάρλο

Ο αλγόριθμος *Δενδρικής Αναζήτησης Μόντε Κάρλο* (Monte Carlo Tree Search - MCTS) είναι ένας αλγόριθμος αναζήτησης που εφαρμόζεται σε προβλήματα απόφασης, αναπαριστώντας την εκάστοτε δομή δεδομένων ως δέντρο [21]. Στο πεδίο της τεχνητής νοημοσύνης για παιχνίδια συγκεκριμένα, οι ενέργειες του παιχνιδιού αντιστοιχίζονται στις ακμές του δέντρου και οι πιθανές καταστάσεις στους κόμβους. Ο στόχος του αλγορίθμου είναι να επιλέξει τη βέλτιστη ενέργεια (δηλαδή μία από τις ακμές του κόμβου-ρίζας), ενώ το δέντρο του παιχνιδιού μπορεί να επεκταθεί έως τους τελικούς κόμβους-φύλλα, ανάλογα με τους διαθέσιμους υπολογιστικούς πόρους και το χρονικό περιορισμό. Δεδομένου ότι το δέντρο του παιχνιδιού δε χρειάζεται να είναι συμμετρικό, ένα βασικό χαρακτηριστικό του αλγορίθμου είναι ότι δίνεται νωρίς έμφαση στις πιο υποσχόμενες ενέργειες, αναπτύσσοντας ανάλογα τη δομή του δέντρου. Η δημιουργία του δέντρου αναζήτησης και η διαδικασία επιλογής από τον αλγόριθμο μπορεί να αναλυθεί σε τέσσερα βασικά στάδια:

1. **Επιλογή (Selection)** Ξεκινώντας από τον κόμβο-ρίζα του δέντρου επιλέγεται μία από τις πιθανές ενέργειες (ένας κόμβος-παιδί) με βάση κάποια μετρική. Στην παραλλαγή του αλγορίθ-

μου που βασίζεται στην εφαρμογή *Άνω Ορίων Εμπιστοσύνης σε Δέντρα* (Upper Confidence bounds applied to Trees - UCT) [53], χρησιμοποιείται η μετρική *Άνω Ορίου Εμπιστοσύνης* (Upper Confidence Bound - UCB) (Εξίσωση 2.48) και επιλέγεται κάθε φορά ο κόμβος με την υψηλότερη τιμή. Αυτό το βήμα επαναλαμβάνεται μέχρι να βρεθεί ένας μη πλήρως εξερευνημένος κόμβος (δηλαδή ένας κόμβος του οποίου οι κόμβοι-παιδιά δεν έχουν διατρεχθεί τουλάχιστον μία φορά ο καθένας).

$$\mathrm{UCB}(a_i) = Q(s, a_i) + C\sqrt{\frac{\log N}{n_i}} \tag{2.48}$$

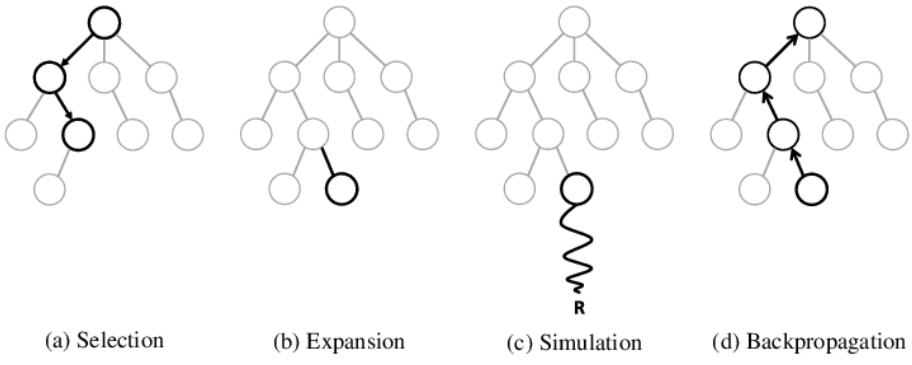
Στην παραπάνω εξίσωση, a_i είναι η ενέργεια που οδηγεί στον κόμβο i, $Q(s, a_i)$ είναι η εκτιμώμενη αξία του κόμβου i, n_i είναι ο αριθμός των φορών που έχει επιλεγεί, N είναι ο αριθμός των φορών που έχει επιλεγεί ο κόμβος-γονέας του και C είναι μία παράμετρος στάθμισης μεταξύ εκμετάλλευσης και εξερεύνησης.

2. **Επέκταση (Expansion)** Όταν εντοπίζεται ένας κόμβος που περιλαμβάνει ενέργειες οι οποίες δεν έχουν εξεταστεί έστω μία φορά, μία από αυτές επιλέγεται τυχαία και προστίθεται στο δέντρο αναζήτησης ένας νέος κόμβος που αναπαριστά την κατάσταση του παιχνιδιού που προκύπτει από την εκτέλεσή της.

3. **Προσομοίωση (Rollout)** Μία πλήρης παρτίδα προσομοιώνεται με αφετηρία το νέο κόμβο που προστέθηκε στο δέντρο, ακολουθώντας μια συγκεκριμένη πολιτική (η οποία είναι διαφορετική από την πολιτική που χρησιμοποιείται στο στάδιο επιλογής του αλγορίθμου). Στην απλή παραλλαγή του αλγορίθμου εκτελούνται τυχαίες κινήσεις μέχρι να προκύψει μία τελική κατάσταση.

4. **Οπισθοδιάδοση (Back-propagation)** Το αποτέλεσμα της προσομοίωσης διαδίδεται προς τα πίσω σε όλους τους κόμβους που διατρέχθηκαν μέχρι τον κόμβο-ρίζα, ανανεώνοντας τα αντίστοιχα στατιστικά των κόμβων.

Τα παραπάνω βήματα εκτελούνται επαναληπτικά, ξεκινώντας από τον κόμβο-ρίζα, για ένα σταθερό αριθμό επαναλήψεων ή μέχρι να ολοκληρωθεί ένα ορισμένο χρονικό διάστημα. Η διαδικασία φαίνεται διαγραμματικά στο Σχήμα 2.21.



Σχήμα 2.21: Στάδια δενδρικής αναζήτησης Μόντε Κάρλο

Για τον προσδιορισμό της βέλτιστης ενέργειας μετά την ολοκλήρωση της διαδικασίας υπάρχουν διάφορες προσεγγίσεις [16], με πιο συνηθισμένη την επιλογή της ενέργειας που έχει διατρεχθεί περισσότερες φορές κατά τη δενδρική αναζήτηση.

- **Μέγιστο παιδί (Max child)**: επιλέγεται η ενέργεια με την υψηλότερη βαθμολογία.
- **Ισχυρό παιδί (Robust child)**: επιλέγεται η ενέργεια με το μεγαλύτερο αριθμό επισκέψεων.
- **Ισχυρό-μέγιστο παιδί (Robust-max child)**: επιλέγεται η ενέργεια που έχει συγχρόνως την υψηλότερη βαθμολογία και το μεγαλύτερο αριθμό επισκέψεων. Αν δεν υπάρχει τέτοια ενέργεια συνεχίζονται οι προσομοιώσεις μέχρι να προκύψει μία ενέργεια που να ικανοποιεί τις δύο συνθήκες.
- **Ασφαλές παιδί (Secure child)**: επιλέγεται η ενέργεια που μεγιστοποιεί ένα κάτω όριο εμπιστοσύνης.

Το κύριο πλεονέκτημα της δενδρικής αναζήτησης Μόντε Κάρλο είναι ότι μπορεί να είναι ιδιαίτερα αποτελεσματική ακόμη και χωρίς προηγούμενη *γνώση του πεδίου* (domain knowledge) στο οποίο εφαρμόζεται· το μόνο προαπαιτούμενο είναι ένα μοντέλο προσομοίωσης του παιχνιδιού το οποίο δεδομένης της τρέχουσας κατάστασης και μίας ενέργειας είναι ικανό να παράγει την επόμενη κατάσταση (ή μία από τις πιθανές επόμενες καταστάσεις σε ένα μη ντετερμινιστικό περιβάλλον). Η ενσωμάτωση γνώσης πεδίου είναι δυνατόν να βελτιώσει την απόδοση του αλγορίθμου, ωστόσο δεν είναι αναγκαία για την ομαλή λειτουργία του. Αυτό το χαρακτηριστικό καθιστά τον αλγόριθμο ιδιαίτερα κατάλληλο για ανάπτυξη γενικών πρακτόρων, καθώς οι μηχανισμοί και οι κανόνες ποικίλλουν μεταξύ των διαφορετικών παιχνιδιών.

Επιπλέον, ο αλγόριθμος μπορεί να διακοπεί οποιαδήποτε στιγμή και να επιστρέψει τη βέλτιστη ενέργεια όπως έχει υπολογιστεί μέχρι εκείνη τη στιγμή, ακόμα και αν δεν έχουν ολοκληρωθεί όλες οι επαναλήψεις. Αυτή η ιδιότητα είναι ιδιαίτερα σημαντική για προβλήματα πραγματικού χρόνου, όπως είναι η εφαρμογή πρακτόρων σε ηλεκτρονικά παιχνίδια όπου ο διαθέσιμος χρόνος εκτέλεσης είναι περιορισμένος.

Κεφάλαιο 3

Κωδικοποίηση Καταστάσεων με χρήση Γενετικών Αλγορίθμων

Σε αυτό το κεφάλαιο παρουσιάζεται μία μεθοδολογία βασισμένη σε γενετικούς αλγορίθμους για την ανάπτυξη ευφυών πρακτόρων για ηλεκτρονικά παιχνίδια. Στο πλαίσιο της τεχνητής νοημοσύνης για παιχνίδια, οι γενετικοί αλγόριθμοι έχουν αξιοποιηθεί μερικώς, κατά κύριο λόγο συνδυαστικά με άλλες μεθόδους ως μέρος υβριδικών τεχνικών. Στην προετεινόμενη προσέγγιση, ο γενετικός αλγόριθμος αποτελεί τη βάση του πράκτορα και μπορεί να λειτουργήσει ως αυτόνομη μέθοδος. Επιπλέον, εξετάζεται η δυνατότητα επαναχρησιμοποίησης μέρους της κωδικοποίησης και η εφαρμογή του πράκτορα σε διαφορετικό περιβάλλον με παρόμοια χαρακτηριστικά. Στις ενότητες που ακολουθούν, γίνεται μία σύντομη ανασκόπηση της σχετικής βιβλιογραφίας, περιγράφεται το περιβάλλον υλοποίησης και παρουσιάζονται ο αλγόριθμος και τα αποτελέσματα από τα σχετικά πειράματα.

3.1 Βιβλιογραφία

Επί του παρόντος, στον κλάδο της τεχνητής νοημοσύνης για παιχνίδια κυριαρχούν οι αλγόριθμοι ενισχυτικής μάθησης και οι παραλλαγές τους. Ωστόσο, οι αλγόριθμοι που ανήκουν στην οικογένεια των εξελικτικών στρατηγικών έχει αποδειχθεί ότι αποτελούν μια αρκετά ανταγωνιστική εναλλακτική από άποψη απόδοσης, ενώ συγχρόνως είναι πολύ πιο απλοί όσον αφορά την υλοποίησή τους [87]. Τα κύριο πλεονεκτήματά τους είναι η δυνατότητα υψηλού βαθμού παραλληλισμού και οι χαμηλές απαιτήσεις μνήμης, που έχουν ως αποτέλεσμα ταχύτερους χρόνους εκπαίδευσης.

Η εφαρμογή των γενετικών αλγορίθμων στο συγκεκριμένο πεδίο έχει συνήθως επικουρικό σκοπό (π.χ. τον προσδιορισμό των υπερπαραμέτρων ενός νευρωνικού δικτύου που επιτελεί το ρόλο του πράκτορα) και δρα συμπληρωματικά με άλλες μεθόδους. Σε αυτό το πλαίσιο, έχουν χρησιμοποιηθεί για την σύνθεση λύσεων (ακολουθιών από ενέργειες) που λειτουργούν εν συνεχεία ως ετικέτες για την εκαπίδευση δικτύων επιβλεπόμενης μάθησης. Αυτό οφείλεται στο ότι ένας γενετικός αλγόριθμος με την κλασική προσέγγιση μπορεί να εκπαιδευτεί ώστε να παράξει μία λύση για ένα συγκεκριμένο επίπεδο (τοπολογία) ενός παιχνιδιού, με αποτέλεσμα η λύση αυτή να μην είναι λειτουργική αν αλλάξει το περιβάλλον για το οποίο εκπαιδεύτηκε και να μη μπορεί να γενικευτεί.

Στην έρευνα που παρουσιάζεται στο [45], το μοντέλο επιλογής ενεργειών βασίζεται στις τιμές

ορισμένων παραμέτρων (οι οποίες αποτελούν ουσιαστικά τις συνθήκες επιλογής) που αφορούν τα χαρακτηριστικά του παίχτη και των αντιπάλων για ένα παιχνίδι στρατηγικής. Ένας γενετικός αλγόριθμος χρησιμοποιείται σε αυτή την περίπτωση για τον προσδιορισμό των καταλληλότερων ορίων στις τιμές των παραμέτρων, προκειμένου να επιλέγεται η βέλτιστη ενέργεια με τη μέγιστη δυνατή συχνότητα. Συνεπώς, τα γονίδια των χρωμοσωμάτων κωδικοποιούν τις αντίστοιχες παραμέτρους και την προτεραιότητα των ενεργειών για τον εκάστοτε συνδυασμό των τιμών τους. Η συγκεκριμένη μέθοδος είναι αποτελεσματική στο περιβάλλον που εφαρμόζεται στο οποίο τέσσερις διαφορετικές ενέργειες είναι διαθέσιμες. Παρόμοια μεθοδολογία εφαρμόστηκε και για το δημοφιλές παιχνίδι Tetris [25]. Και σε αυτή την περίπτωση ο γενετικός αλγόριθμος λειτουργεί εμμέσως, με στόχο τη βελτιστοποίηση των παραμέτρων μίας συνάρτησης αξιολόγησης, η οποία χρησιμοποιείται για την τελική επιλογή των ενεργειών.

Μία προσέγγιση του προβλήματος με χρήση αυτόνομων γενετικών αλγορίθμων σε ηλεκτρονικά παιχνίδια παρουσιάστηκε στο *Πολυπρακτορικό Σύστημα Εξέλιξης Πραγματικού Χρόνου* (On-Line Evolution of Multi-Agent Systems - OLEMAS) [26]. Σε αυτό το σύστημα σχηματίζονται αρχικά ζεύγη ενέργειας-κατάστασης για συγκεκριμένο πλήθος συνδυασμών μέσω μίας εξελικτικής διαδικασίας και κατά τη λειτουργία του πράκτορα υπολογίζεται η κοντινότερη κατάσταση (από το σύνολο των καταστάσεων που έχουν κωδικοποιηθεί) στην πραγματική με τη μέθοδο *Κ-Πλησιέστερων Γειτόνων* (K-Nearest Neighbors - KNN) [27] και επιλέγεται η αντίστοιχη ενέργεια. Η συγκεκριμένη τεχνική έχει το μειονέκτημα ότι αντιστοιχίζει ένα συγκεκριμένο σύνολο καταστάσεων στη βέλτιστη ενέργεια, με αποτέλεσμα να υστερεί σε ακρίβεια όταν απαντώνται καταστάσεις με μεγάλη απόσταση (όσον αφορά τα διανύσματα αναπαράστασής τους) από τις ήδη κωδικοποιημένες καταστάσεις.

Άλλη μία μέθοδος ανάπτυξης ενός ευφυούς πράκτορα βασισμένου απευθείας σε γενετικούς αλγορίθμους προτείνεται στο [80]. Σε αυτή την τεχνική *κυλιόμενου ορίζοντα* (rolling horizon) ο ΓΑ κωδικοποιεί μακρο-ενέργειες, δηλαδή ακολουθίες ενεργειών συγκεκριμένου μήκους οι οποίες αποτελούνται από μία ενέργεια που επαναλαμβάνεται N φορές. Κάθε γονίδιο του χρωμοσώματος κωδικοποιεί μία μακρο-ενέργεια και συνολικά το χρωμόσωμα μία ακολουθία από L μακρο-ενέργειες με συνολικό μήκος $N \times L$. Ο αλγόριθμος εκτελείται σε πραγματικό χρόνο χρησιμοποιώντας ένα μοντέλο προσομοίωσης. Κάθε άτομο (ακολουθία ενεργειών) εφαρμόζεται μέσω αυτού του μοντέλου και αξιολογείται ανάλογα με την τελική κατάσταση που προκύπτει. Το κύριο πρόβλημα αυτής της μεθοδολογίας είναι αφενός ότι πρέπει ο πληθυσμός του ΓΑ να αποτελείται από μικρό αριθμό ατόμων λόγω των χρονικών περιορισμών δεδομένου ότι η εκτέλεση του γίνεται σε πραγματικό χρόνο και αφετέρου η ανάγκη ύπαρξης ενός μοντέλου του περιβάλλοντος ικανού να προσομοιώνει τις υποψήφιες ακολουθίες ενεργειών.

3.2 Περιβάλλον Υλοποίησης

Ο διαγωνισμός *Γενικευμένης Τεχνητής Νοημοσύνης για Ηλεκτρονικά Παιχνίδια* (General Video Game AI - GVGAI) [78], που πραγματοποιήθηκε για πρώτη φορά το 2014, πραγματοποιείται κάθε χρόνο με στόχο την ανάπτυξη γενικών πρακτόρων, ικανών να παίξουν κάθε είδους παιχνίδι. Ένα προγραμματιστικό περιβάλλον που αναλύει και αποκωδικοποιεί την ειδικά σχεδιασμένη *Περιγραφική Γλώσσα Ηλεκτρονικών Παιχνιδιών* (Video Game Description Language - VGDL) [79] παρέχεται στα πλαίσια του διαγωνισμού, προσφέροντας περισσότερα από 100 κλασικά παιχνίδια δύο διαστάσεων για την εκπαίδευση και τη δοκιμή ευφυών πρακτόρων. Η διεπαφή παρέχει επίσης ένα προεπιλεγμένο σύνολο λειτουργιών μέσω των οποίων μπορούν να ανακτώνται πληροφορίες σχετικά με την κατάσταση του κόσμου του παιχνιδιού (όπως οι θέσεις των αντικειμένων, οι διαθέσιμες ενέργειες, το σκορ και το

τέλος του παιχνιδιού). Επιπλέον, υπάρχει η δυνατότητα προσομοίωσης ενεργειών και πρόβλεψης της επόμενης κατάστασης.

Στο πλαίσιο του διαγωνισμού, οι πράκτορες έχουν χρόνικό όριο $40ms$ για την επιλογή κάθε ενέργειας και $1s$ για αρχικοποίηση κατά την έναρξη του παιχνιδιού. Τα παιχνίδια είναι χωρισμένα σε τρία υποσύνολα [79]. Το πρώτο είναι το υποσύνολο εκπαίδευσης, το οποίο παρέχεται ανοιχτά στους διαγωνιζόμενους. Το δεύτερο υποσύνολο είναι κρυφό και οι διαγωνιζόμενοι έχουν πρόσβαση μόνο στα αποτελέσματα των πρακτόρων τους σε αυτά τα παιχνίδια ώστε να ελέγξουν την απόδοσή τους. Τέλος, το τρίτο υποσύνολο είναι αυτό που χρησιμοποιείται για τη δοκιμή και την κατάταξη των πρακτόρων και είναι επίσης κρυφό.

Ο πράκτορας που παρουσιάζεται σε αυτό το κεφάλαιο έχει αναπτυχθεί στην πλατφόρμα που διατίθεται από τον διαγωνισμό GVGAI και έχει δοκιμαστεί στα παιχνίδια *Zelda* και *Portals*. Και τα δύο είναι τυπικά ηλεκτρονικά παιχνίδια τύπου περιπέτειας, στα οποία στόχος του πράκτορα είναι να φτάσει σε μια πόρτα εξόδου επιζώντας παράλληλα από τους εχθρούς. Συγκεκριμένα, στο παιχνίδι Zelda (Σχήματα 3.1 και 3.2α′) ο πράκτορας πρέπει αρχικά να βρει ένα κλειδί και στη συνέχεια να φτάσει στην έξοδο. Συνολικα έξι διαφορετικές ενέργειες είναι διαθέσιμες σε αυτό το παιχνίδι. Στο παιχνίδι Portals (Σχήμα 3.2β′), υπάρχουν διάφορες πύλες διάσπαρτες σε διαφορετικά δωμάτια του κόσμου του παιχνιδιού. Οι πύλες μεταφέρουν τον πράκοτρα σε τυχαία δωμάτια και ο στόχος του είναι να φτάσει στο σωστό δωμάτιο και τελικά να βρει την πόρτα εξόδου. Αυτό το παιχνίδι είναι πιο περίπλοκο και δύσκολο, καθώς υπάρχουν πολλοί εχθροί διαφορετικών τύπων και οι διαθέσιμες ενέργειες είναι πέντε, πράγμα που σημαίνει ότι ο πράκτορας δεν έχει τη δυνατότητα να επιτεθεί και μπορεί μόνο να αποφύγει τους εχθρούς.

3.3 Μεθοδολογία

Στην παρούσα προσέγγιση, υλοποιείται μία αυτόνομη τεχνική βασισμένη σε ΓΑ με εφαρμογή σε ευφυείς πράκτορες για ηλεκτρονικά παιχνίδια. Ο στόχος είναι να βρεθεί, μέσω μιας εξελικτικής διαδικασίας δοκιμής-λάθους, η καλύτερη δυνατή ενέργεια σε κάθε χρονικό βήμα ανάλογα με την κατάσταση του κόσμου του παιχνιδιού, δηλαδή ανάλογα με τη θέση και τον προσανατολισμό των διάφορων *αντικειμένων/χαρακτήρων* (sprites) του παιχνιδιού. Ο προτεινόμενος αλγόριθμος εκπαιδεύεται εκ των προτέρων και το βέλτιστο άτομο-λύση χρησιμοποιείται απευθείας για την επιλογή των ενεργειών κατά τη λειτουργία του πράκτορα χωρίς να υπάρχει ουσιαστική χρονική καθυστέρηση.

Τα χρωμοσώματα κάθε λύσης κωδικοποιούν τη βέλτιστη ενέργεια με βάση την κατάσταση του παιχνιδιού και ως εκ τούτου ο πράκτορας μπορεί εύκολα να γενικευτεί σε περισσότερα επίπεδα του ίδιου παιχνιδιού. Επιπλέον, παρόμοιες καταστάσεις παρατηρούνται σε διαφορετικά παιχνίδια του ίδιου τύπου (π.χ. στρατηγικής), που σημαίνει ότι όταν ο πράκτορας εκπαιδευτεί σε ένα από αυτά μπορεί να προσαρμοστεί και σε άλλα παρόμοια παιχνίδια. Σε αυτό το πλαίσιο, προτείνεται η χρήση διακριτών κωδικοποιήσεων για τις ενέργειες που σχετίζονται με την κίνηση του πράκτορα και τις υπόλοιπες πιο ειδικές ενέργειες (που συνήθως σχετίζονται με την αντιμετώπιση των εχθρών). Η κωδικοποίηση των ενεργειών κίνησης μπορεί στη συνέχεια, εφόσον έχει καθοριστεί μία φορά, να ενσωματωθεί στον γενότυπο άλλων πρακτόρων (που αφορούν διαφορετικά παιχνίδια) με αποτέλεσμα τη μείωση των καταστάσεων που πρέπει να εξερευνηθούν από τον ΓΑ.

Ιδανικά, η αναπαράσταση της κατάστασης θα έπρεπε να περιλαμβάνει όλα τα αντικείμενα στον χάρτη του παιχνιδιού ώστε να αξιοποιηθεί όλη η διαθέσιμη πληροφορία. Ωστόσο, ο χώρος αναζήτησης S αυξάνεται εκθετικά με τον αριθμό των εξεταζόμενων κελιών b του κόσμου ($S(b,t) = t^b$), με

το πλήθος των διαφορετικών τύπων αντικειμένων t να είναι ο *παράγοντας αύξησης* (growth factor). Συνεπώς, είναι πρακτικά αδύνατο να κωδικοποιηθεί κάθε δυνατή κατάσταση παιχνιδιού καθώς για ένα κοινό παιχνίδι δύο διαστάσεων με περίπου 100 διακριτά κελιά και 5 διαφορετικούς τύπους αντικειμένων/χαρακτήρων υπάρχουν 5^{100} πιθανές καταστάσεις παιχνιδιού. Ως εκ τούτου, είναι απαραίτητη μία μέθοδος κωδικοποίησης για την κοινή αντιμετώπιση παρόμοιων καταστάσεων. Στην επόμενη ενότητα παρουσιαζεται μία τεχνική κωδικοποίησης βασισμένη σε *N-πλειάδες* (N-groups) ειδικά σχεδιασμένη για αυτόν τον λόγο.

3.3.1 Κωδικοποίηση Κίνησης

Προκειμένου να επιτευχθεί ισορροπία μεταξύ της διαθέσιμης πληροφορίας και του υπολογιστικού κόστους, για την αναπαράσταση της κατάστασης ένα παράθυρο από M κελιά γύρω από τον πράκτορα λαμβάνεται υπόψη σε κάθε χρονικό βήμα και εν συνεχεία τα κελιά ομαδοποιούνται σε N-πλειάδες. Στη συγκεκριμένη υλοποίηση, έπειτα από δοκιμές το πλήθος M των κελιών έχει οριστεί ίσο με 28 και το N ίσο με 3 (κάθε πλειάδα αποτελείται από τρία μπλοκ) (Σχήμα 3.1). Εφτά τριπλέτες έχουν επιλεχθεί σε κάθε κατεύθυνση (πάνω, κάτω, αριστερά και δεξιά) με στόχο την πληρέστερη δυνατή αναπαράσταση της κατάστασης. Ωστοσο, μόνο οι τριπλέτες που βρίσκονται στην κατεύθυνση του τρέχοντος στόχου (π.χ. πόρτα εξόδου) συμβάλλουν στην απόφαση της αντίστοιχης ενέργειας σε κάθε χρονικό βήμα. Με αυτό τον τρόπο ο συνολικός γενότυπος περιλαμβάνει λιγότερα χρωμοσώματα και μπορεί να εκπαιδευτεί αποτελεσματικότερα.

Οπως φαίνεται στο Σχήμα 3.1, οι τριπλέτες για κάθε κατεύθυνση αποτελούνται κυρίως από κελιά που βρίσκονται στον τρέχοντα προσανατολισμό του πράκτορα, ωστόσο περιλαμβάνουν και κελιά προς τις υπόλοιπες κατευθύνσεις (π.χ. η έβδομη τριπλέτα). Έτσι, επιτυγχάνεται επαρκής αναπαράσταση της κατάστασης του παιχνιδιού παρότι χρησιμοποιούνται λιγότερα χρωμοσώματα για την κωδικοποίηση. Κάθε τριπλέτα κωδικοποιείται σε ένα χρωμόσωμα που αποτελείται από πλήθος γονιδίων ίσο με τον αριθμό των πιθανών διαφορετικών καταστάσεων που μπορεί να έχει η τριπλέτα συνολικά (π.χ. στην περίπτωση που υπάρχουν 3 διαφορετικοί τύποι αντικειμένων, οι δυνατές διαφορετικές καταστάσεις της τριπλέτας είναι $3^3 = 27$). Οι τιμές των γονιδίων υποδεικνύουν τη βέλτιστη ενέργεια με βάση την κατάσταση της τριπλέτας και ως εκ τούτου η τιμή κάθε γονιδίου είναι ένας ακέραιος στο εύρος $[0, 5]$, δεδομένου ότι υπάρχουν έξι διαθέσιμες ενέργειες. Σε κάθε βήμα της εκτέλεσης του αλγορίθμου προσδιορίζεται σε κάθε χρωμόσωμα το γονίδιο που αντιστοιχεί στην τρέχουσα κατάσταση της τριπλέτας και υποδεικνύει τη βέλτιστη ενέργεια. Τέλικά, επιλέγεται η ενέργεια που υποδεικνύεται από τα περισσότερα γονίδια ενώ σε περίπτωση που δύο ή περισσότερες ενέργειες έχουν το ίδιο πλήθος εμφανίσεων, επιλέγεται μία από αυτές τυχαία.

Ακόμη και για N-πλειάδες που αποτελούνται από τρία κελιά, ένας μεγάλος αριθμός δυνατών καταστάσεων μπορεί να προκύψει ανάλογα με το πλήθος των διαφορετικών τύπων αντικειμένων. Περαιτέρω μείωση του χώρου καταστάσεων μπορεί να επιτευχθεί λαμβάνοντας υπόψη λιγότερους τύπους. Προς αυτή την κατεύθυνση, χρησιμοποιείται ξεχωριστή κωδικοποίηση για την επιλογή των ενεργειών του πράκτορα που αφορούν κίνηση και για την επιλογή των υπόλοιπων ενεργειών. Συγκεκριμένα, για την περίπτωση που δεν υπάρχει κάποιος χαρακτήρας κοντά στον πράκτορα (σε απόσταση τουλάχιστον δύο κινήσεων) ορίζονται συγκεκριμένα χρωμοσώματα υπεύθυνα για την κίνηση του πράκτορα, τα οποία κωδικοποιούν τις καταστάσεις στις οποίες δεν περιλαμβάνονται *χαρακτήρες που ελέγχονται από τον υπολογιστή* (Non-Player Characters - NPCs) (δηλαδή κάθε κελί μπορεί να είναι κενό ή να αναπαριστά εμπόδιο). Σε αυτή την περίπτωση, το κάθε χρωμόσωμα κίνησης αποτελείται από οχτώ γονίδια καθώς κάθε κελί της τριπλέτας μπορεί να έχει δύο διαφορετικές καταστάσεις. Επιπλέον, δεδομένου ότι οι

Σχήμα 3.1: Επιλεγμένες τριπλέτες από κελιά στη δεξιά κατεύθυνση του πράκτορα για το παιχνίδι Zelda

N-πλειάδες είναι συμμετρικές ως προς τις τέσσερις κατευθύνσεις, εφτά χρωμοσώματα (ένα για κάθε μία από τις τριπλέτες που φαίνονται στο Σχήμα 3.1) αρκούν για να κωδικοποιήσουν και τις είκοσι οχτώ συνολικά επιλεγμένες N-πλειάδες.

Εκτός από τη μείωση του χώρου καταστάσεων (και κατά συνέπεια την επιτάχυνση της διαδικασίας της εκπαίδευσης) ο διαχωρισμός των χρωμοσωμάτων επιτρέπει την επαναχρησιμοποίηση τμήματος της λογικής του πράκτορα σε παρόμοια περιβάλλοντα. Συγκεκριμένα, η κωδικοποίηση των χρωμοσωμάτων κίνησης μπορεί εύκολα να γενικευτεί σε περισσότερα παιχνίδια, δεδομένου ότι δεν περιλαμβάνει τους πιο συγκεκριμένους τύπους αντικειμένων (που ενδέχεται να διαφέρουν μεταξύ των παιχνιδιών) και να χρησιμοποιηθεί αυτούσια σε διαφορετικά παιχνίδια μετά από εκπαίδευση σε ένα περιβάλλον.

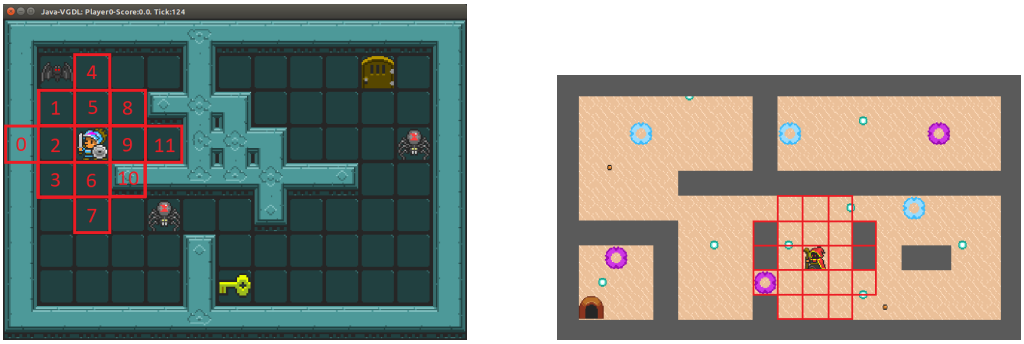
3.3.2 Κωδικοποίηση Χαρακτήρων Υπολογιστή

Όπως περιγράφεται παραπάνω, η κωδικοποίηση μίας κατάστασης είναι διαφορετική όταν υπάρχουν χαρακτήρες υπολογιστή γύρω από τον πράκτορα. Αυτή η σχεδιαστική επιλογή έχει διττό ρόλο· αφενός τη μείωση των δυνατών καταστάσεων των N-πλειάδων και αφετέρου τη δυνατότητα ξεχωριστής επεξεργασίας των περισσότερο κρίσιμων καταστάσεων. Δεδομένου ότι η συμπεριφορά των NPCs δεν είναι ίδια σε διαφορετικά παιχνίδια (ακόμα και της ίδιας κατηγορίας), η κωδικοποίηση πρέπει να είναι ξεχωριστή για κάθε παιχνίδι. Έτσι, σε αντίθεση με τα χρωμοσώματα κίνησης (τα οποία εκπαιδεύονται σε ένα από τα παιχνίδια και στη συνέχεια επαναχρησιμοποιούνται αυτούσια), τα χρωμοσώματα των χαρακτήρων υπολογιστή εκπαιδεύονται ειδικά για κάθε παιχνίδι.

Στο Zelda, υπάρχουν τρία ή τέσσερα NPCs με παρόμοια συμπεριφορά (όπως απεικονίζεται στο Σχήμα 3.1) σε κάθε επίπεδο. Κατά συνέπεια, είναι δυνατόν να κωδικοποιηθούν όλα σε ένα μόνο χρωμόσωμα, αφού ο πράκτορας θα πρέπει να τα αντιμετωπίζει με τον ίδιο τρόπο. Ένα ιδιαίτερο χαρακτηριστικό του συγκεκριμένου παιχνιδιού είναι ότι όταν επιλέγεται μία ενέργεια κίνησης, ο πράκτορας κινείται μόνο στην περίπτωση που η κατεύθυνση της κίνησης ταιριάζει με τον τρέχοντα προσανατολι-

σμό του, διαφορετικά η ενέργεια προκαλεί αλλαγή στον προσανατολισμό αλλά ο πράκτορας παραμένει ακίνητος. Αυτό σημαίνει ότι είναι απαραίτητες δύο διαδοχικές εκτελέσεις μίας ενέργειας κίνησης προκειμένου να κινηθεί σε διαφορετική κατεύθυνση. Ως εκ τούτου, ο προσανατολισμός του πράκτορα σε κάθε χρονικό βήμα πρέπει να λαμβάνεται επίσης υπόψη.

Η ιδιαιτερότητα του περιγραφόμενου πράκτορα καθιστά επίσης απαραίτητη την κωδικοποίηση των χρωμοσωμάτων που αφορούν τα NPCs ανάλογα με την απόστασή τους από τον πράκτορα. Το Σχήμα 3.2α΄ απεικονίζει το παράθυρο των κελιών που ελέγχονται για αυτό το σκοπό. Στην τεχνική που παρουσιάζεται χρησιμοποιούνται δύο χρωμοσώματα· ένα που αφορά την περίπτωση στην οποία βρίσκεται κάποιος χαρακτήρας NPC στην ελάχιστη δυνατή απόσταση από τον πράκτορα (κελιά 2, 5, 6 και 9 στο Σχήμα 3.2α΄) και ένα για τις υπόλοιπες περιπτώσεις (που βρίσκονται σε μεγαλύτερη απόσταση). Τα γονίδια αυτών των χρωμοσωμάτων υποδεικνύουν τη βέλτιστη ενέργεια ανάλογα με τον προσανατολισμό του πράκτορα και τη θέση των χαρακτήρων NPC.



(α΄) Zelda (β΄) Portals

Σχήμα 3.2: Έλεγχος κελιών για χαρακτήρες που ελέγχονται από τον υπολογιστή (NPCs)

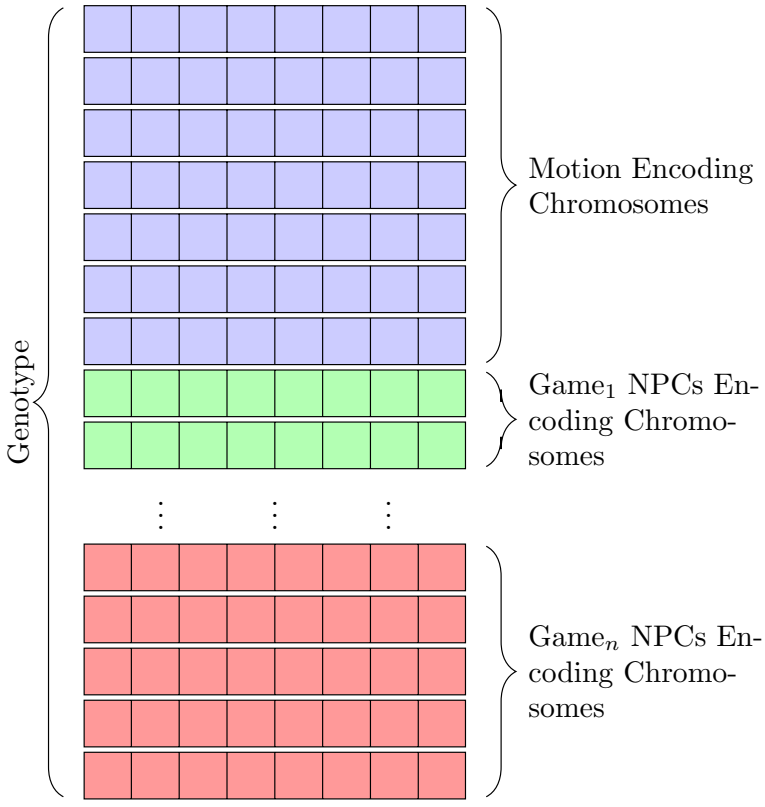
Όσον αφορά στο παιχνίδι Portals, είναι αρκετά πιο περίπλοκο λόγω του μεγάλου πλήθους των NPCs καθώς και του τρόπου με τον οποίο κινούνται. Ανάλογα με την κίνησή τους μπορούν να ταξινομηθούν στις ακόλουθες κατηγορίες:

- Πρώτη κατηγορία: NPCs με κίνηση στον κατακόρυφο άξονα που αλλάζουν κατεύθυνση μόνο μετά από σύγκρουση με τοίχο.
- Δεύτερη κατηγορία: NPCs με κίνηση στον οριζόντιο άξονα που αλλάζουν κατεύθυνση μόνο μετά από σύγκρουση με τοίχο.
- Τρίτη κατηγορία: NPCs που κινούνται άτακτα στο χώρο.

Εκτός από τους διάφορους τύπους, ένα άλλο χαρακτηριστικό των NPCs του παιχνιδιού είναι τα σημεία του χώρου στα οποία μπορούν να μετακινηθούν. Παρότι ο πράκτορας μπορεί να κινηθεί μόνο από ένα κελι σε ένα γειτονικό (όπως ακριβώς στο Zelda), οι χαρακτήρες υπολογιστή μπορούν επίσης να κινηθούν ανάμεσα στα διαφορετικά κελιά. Ειδικότερα, οι NPCs που ανήκουν στις πρώτες δύο κατηγορίες χρειάζονται δύο χρονικά βήματα του παιχνιδιού για να μετακινηθούν στο επόμενο κελί, που σημαίνει ότι αφού ολοκληρωθεί το πρώτο βρίσκονται στο ενδιάμεσο δύο διαδοχικών κελιών (Σχήμα 3.2β΄) ενώ τα NPCs της τρίτης κατηγορίας μπορούν να βρεθούν σε οποιεσδήποτε συντεταγμένες, εφόσον αυτό επιτρέπεται από την τοπολογία του συγκεκριμένου επιπέδου.

Σε αυτήν την περίπτωση, δε θα ήταν αποτελεσματικό να κωδικοποιηθούν όλα τα NPCs σε ένα μόνο χρωμόσωμα, καθώς παρουσιάζουν σημαντικές διαφορές στη συμπεριφορά τους. Ως εκ τούτου, χρησιμοποιείται αριθμός χρωμοσωμάτων ίσος με το πλήθος των διαφορετικών κατηγοριών για την κωδικοποίηση κάθε τύπου ξεχωριστά. Οι δύο πρώτες κατηγορίες που παρουσιάζονται παραπάνω μπορούν να διαχωριστούν περαιτέρω ανάλογα με την κατεύθυνση κίνησης του αντικειμένου, που σημαίνει ότι προκύπτουν δύο νέες υποκατηγορίες από καθεμία, με αποτέλεσμα ένα σύνολο πέντε διαφορετικών τύπων. Όπως και στο Zelda, η συμμετρία του χώρου επιτρέπει την κωδικοποίηση κάθε τύπου για μία κατεύθυνση (χρησιμοποιώντας μόνο ένα χρωμόσωμα) και έπειτα την αντίστοιχη τροποποίηση της επιλεγμένης ενέργειας ώστε να ανταποκρίνεται στην πραγματική κατεύθυνση. Επομένως πέντε χρωμοσώματα ορίζονται στον ΓΑ, ένα για κάθε πιθανή κατηγορία NPC, υποδεικνύοντας την καλύτερη ενέργεια για την αντίστοιχη περίπτωση.

Στο Σχήμα 3.3 φαίνεται η δομή που έχει ο γενότυπος του αλγορίθμου στην προτεινόμενη μεθοδολογία. Οι γραμμές αντιπροσωπεύουν τα διαφορετικά χρωμοσώματα και τα κελιά, τα γονίδια. Τα χρωμοσώματα κίνησης (μπλε) εκπαιδεύονται μία φορά και είναι κοινά για όλα τα παιχνίδια. Τα υπόλοιπα χρωμασώματα εκπαιδεύονται ξεχωριστά για καθε παιχνίδι. Για την επιλογή της ενέργειας, αρχικά γίνεται ένας έλεγχος ώστε να προσδιοριστεί αν θα χρησιμποποιηθούν τα χρωμοσώματα κίνησης ή τα ειδικά χρωμοσώματα του παιχνιδιού. Έπειτα επιλέγεται ένα γονίδιο (κελί) από κάθε χρωμόσωμα (από το αντίστοιχο υποσύνολο) ανάλογα με την τρέχουσα κατάσταση του παιχνιδιού και προκύπτει η τελική ενέργεια ως αυτή με το μεγαλύτερο πλήθος εμφανίσεων στα επιλεγμένα γονίδια.



Σχήμα 3.3: Δομή γενότυπου

Αρχικά, η προτεινόμενη προσέγγιση εφαρμόζεται στο (απλούστερο) παιχνίδι Zelda προκειμένου να προσδιοριστούν τόσο τα χωμοσώματα κίνησης όσο και τα χρωμοσώματα που αφορούν τους χαρακτήρες

που ελέγχονται από τον υπολογιστή. Στη συνέχεια, τα χρωμοσώματα κίνησης που προσδιορίστηκαν επαναχρησιμοποιούνται στο παιχνίδι Portals και ο ΓΑ εκτελείται από την αρχή, προκειμένου να εκπαιδευτεί το τμήμα των επιπλέον χρωμασωμάτων του παιχνιδιού. Αυτή η στρατηγική επιλέχθηκε για δύο λόγους. Πρώτον, στο δεύτερο περιβάλλον υπάρχουν περισσότερες καταστάσεις που περιλαμβάνουν NPCs καθιστώντας πιο δύσκολο και χρονοβόρο τον υπολογισμό των αντίστοιχων χρωμοσωμάτων και δεύτερον, στο παιχνίδι Zelda τα χρωμοσώματα κίνησης χρησιμοποιούνται με μεγαλύτερη συχνότητα (καθώς υπάρχουν πολύ λιγότερα NPCs) και επομένως είναι πιο κατάλληλο για την εκπαίδευση των συγκεκριμένων χρωμοσωμάτων.

3.3.3 Επιλογή Ενέργειας

Η προτεινόμενη τεχνική αποτελείται από μια *εκτός λειτουργίας* (off-line) εκτέλεση του γενετικού αλγορίθμου (φάση εκπαίδευσης) κατά την οποία προσδιορίζονται οι τιμές των γονιδίων στα χρωμοσώματα και εν συνεχεία από την εφαρμογή του σε *πραγματικό χρόνο* (on-line) για την επιλογή της ενέργειας. Στη δεύτερη φάση, η πληροφορία από τα εκπαιδευμένεα χρωμοσώματα συνδυάζεται με έναν επιπλέον έλεγχο πριν τη λήψη της τελικής ενέργειας για την αποτελεσματικότερη λειτουργία του πράκτορα.

Αφού ολοκληρωθεί η φάση εκπαίδευσης, ο πράκτορας είναι έτοιμος να εφαρμοστεί σε πραγματικό χρόνο χρησιμοποιώντας τον γενότυπο που έχει προκύψει από τον ΓΑ. Στη φάση της δοκιμής ενσωματώνεται και το μοντέλο προσομοίωσης στη διαδικασία λήψης αποφάσεων για κάθε ενέργεια. Το μοντέλο προσομοίωσης επιτρέπει στον πράκτορα να μεταβεί σε μία μελλοντική κατάσταση προσομοιώνοντας μία ενέργεια και χρησιμοποιείται σε αυτή την προσέγγιση ως ένας πρόσθετος έλεγχος ασφαλείας. Πιο συγκεκριμένα, σε περιπτώσεις όπου ο χαρακτήρας που ελέγχει ο πράκτορας βρίσκεται σε κίνδυνο, η ενέργεια που προτείνεται από το αντίστοιχο χρωμόσωμα προσομοιώνεται πρώτα ώστε να επιβεβαιωθεί ότι δεν οδηγεί σε τελική κατάσταση ήττας (καθώς αυτός ο έλεγχος αφορά μόνο ένα βήμα, το μοντέλο προσομοιώσης χρησιμοποιείται το πολύ μία φορά σε κάθε χρονικό βήμα και δεν υπερβαίνει τον χρονικό περιορισμό του περιβάλλοντος για τη λήψη της απόφασης). Αυτός ο έλεγχος γίνεται επειδή οι χαρακτήρες που ελέγχονται από τον υπολογιστή κωδικοποιούνται μεμονωμένα (λόγω του μεγέθους του χώρου καταστάσεων) και κατά συνέπεια η προτεινόμενη ενέργεια μπορεί να μην οδηγεί πάντα στην αναμενόμενη κατάσταση (αυτό μπορεί και να συμβεί και λόγω της στοχαστικότητας των παιχνιδιών). Σε περίπτωση που από την προσομοίωση προκύψει τελική κατάσταση του παιχνιδιού με ήττα ή περισσότερες από μία ενέργειες έχουν την ίδια βαθμολογία (το ίδιο πλήθος εμφανίσεων από τα επιλεγμένα γονίδια), μία τυχαία κίνηση ή μία από τις ενέργειες που ισοβαθμούν επιλέγεται αντίστοιχα. Η παραπάνω διαδικασία επιλογής ενέργειας συνοψίζεται στον Αλγόριθμο 3.

3.3.4 Παράμετροι Γενετικού Αλγορίθμου

Η γενική μορφή της συνάρτησης ποιότητας του ΓΑ με την οποία γίνεται η αξιολόγηση των ατόμων σε κάθε γενιά παρουσιάζεται στην Εξίσωση 3.1:

$$fitness = a\left[h\left(D(subgoal), D(subgoal)_{\max}\right)\right] \tag{3.1}$$

όπου a είναι μία παράμετρος ανταμοιβής (αν ο χαρακτήρας που ελέγχεται από τον πράκτορα δεν έχει φτάσει σε τελική κατάσταση ήττας) στο εύρος $[0, 1]$ (καθορίζεται μέσω δοκιμών για κάθε παιχνίδι

Algorithm 3: Action Selection using Chromosomes

```
1  Function Act(StateObservation):
2      initialize actionsValues to zeros
3      if npcInRange() then
4          NpcUpdateValues()
5          bestAction ← actionsValues.getMax()
6          stCopy ← StateObservation.copy()
7          advance(bestAction)
8          if stCopy.isAgentAlive() then
9              return bestAction
10         else
11             return randomAction()
12     else
13         updateValues()
14         bestAction ← actionsValues.getMax()
15         return bestAction
16
```

ξεχωριστά), $D(x)$ είναι η απόσταση του πράκτορα από το αντικείμενο x και h είναι μια ειδικά σχεδιασμένη συνάρτηση που υπολογίζει τη βαθμολογία με βάση την απόσταση του πράκτορα από τον τρέχοντα στόχο (ή τον υποστόχο), ανάλογα με τα χαρακτηριστικά του παιχνιδιού.

Συγκεκριμένα, για το παιχνίδι Zelda η συνάρτηση ποιότητας παίρνει τη μορφή της Εξίσωσης 3.2. Ο κύριος στόχος του πράκοτορα είναι να φτάσει στην πόρτα της εξόδου, αλλά υπάρχει επίσης ένα υποστόχος που έγκειται στην εύρεση του κλειδιού. Λαμβάνοντας υπόψη αυτό το γενονός, η συνάρτηση ποιότητας χωρίζεται σε δύο ισοσταθμισμένα μέρη (με βάρος 0.5) αντίστοιχα με τους δύο υποστόχους. Η συνάρτηση ποιότητας εξαρτάται αρχικά αποκλειστικά από την απόσταση από το κλειδί, με δυνατότητα να φτάσει σε μέγιστη τιμή 0.5 και αφού βρεθεί το κλειδί η τιμή της αυξάνετα ανάλογα με την απόσταση από την πόρτα, οδηγώντας σε μέγιστη συνολική τιμή ίση με 1 όταν ολοκληρωθεί το επίπεδο με επιτυχία όπως φαίνεται στην ακόλουθη εξίσωση:

$$f_{\text{zelda}} = a\left[0.5\left(2 - \frac{D(e)}{D(e)_{\max}}\right)c + 0.5\left(1 - \frac{D(k)}{D(k)_{\max}}\right)(\neg c)\right] \tag{3.2}$$

όπου e είναι η πόρτα εξόδου, k είναι το κλειδί και c είναι μία δυαδική μεταβλητή $(0, 1)$ που υποδηλώνει αν έχει βρεθεί το κλειδί ή όχι. Η παράμετρος ανταμοιβής a (που σχετίζεται με την επιβίωση του πράκτορα) έχει προσδιοριστεί, μετά από πειραματισμό ίση με 1.0 εάν ο πράκτορας είναι ακόμα ζωντανός και 0.7 διαφορετικά.

Η Εξίσωση 3.3 αντικατοπτρίζει τη συνάρτηση ποιότητας για το παιχνίδι Portals. Σε αυτό το περιβάλλον υπάρχουν πολλά διαφορετικά δωμάτια από τα οποία πρέπει να περάσει ο πράκτορας μέχρι να φτάσει στην έξοδο. Για αυτό το λόγο, κάθε νεό δωμάτιο που διατρέχεται αυξάνει τη συνολική τιμή της συνάρτησης ποιότητας ανάλογα με το συνολικό αριθμό των δωματίων στο επίπεδο. Επιπλέον, κατά την παραμονή του πράκτορα σε ένα δωμάτιο η τιμή αυξάνεται επίσης αναλογικά με την απόσταση

από την πύλη που οδηγεί στο επόμενο δωμάτιο. Η συνολική τιμή της συνάρτησης ποιότητας ισούται με 1 όταν ολοκληρωθεί το επίπεδο.

$$f_{\text{portals}} = a \left[r^i_{\max} - \left(r^i_{\max} - r^{i-1}_{\max}\right) \frac{D(g^i)}{D(g^i)_{\max}} \right] \tag{3.3}$$

όπου r^i_{max} είναι η μέγιστη ανταμοιβή που μπορεί να ληφθεί στο δωμάτιο i και g^i είναι η πύλη-στόχος του δωματίου i. Η παράμετρος ανταμοιβής a παίρνει τις ίδιες τιμές με την περίπτωση του πρώτου παιχνιδιού (Εξίσωση 3.2).

Όσον αφορά στις υπόλοιπες υπερπαραμέτρους του ΓΑ, διάφοροι τελεστές επιλογής, διασταύρωσης και μετάλλαξης εξετάστηκαν κατά τη φάση εκπαίδευσης καταλήγοντας σε ένα σύνολο ευρέως χρησιμοποιούμενων προεπιλογών. Συγκεκριμένα επιλέχθηκαν οι εξής τελεστές:

- Τελεστής επιλογής: τελεστής ρουλέτας
- Τελεστής διασταύρωσης: διασταύρωση ενός σημείου
- Τελεστής μετάλλαξης: ομοιόμορφη μετάλλαξη

3.4 Αποτελέσματα

Για την αξιολόγηση του προτεινόμενου αλγορίθμου ο πράκτορας εφαρμόστηκε 500 φορές στο παιχνίδι Zelda (100 φορές σε καθένα από τα 5 διαφορετικά επίπεδα) και 500 φορές στο παιχνίδι Portals. Το ποσοστό νίκης, η μέση βαθμολογία και τα χρονικά βήματα που χρειάστηκαν για την ολοκλήρωση του παιχνιδιού υπολογίστηκαν για κάθε επίπεδο ξεχωριστά, αλλά και συνολικά για το κάθε παιχνίδι προκειμένου να αξιολογηθεί η συνολική απόδοση του πράκτορα (Πίνακες 3.1-3.3). Τα αποτελέσματα συγκρίνονται με μία προσέγγιση ΓΑ κυλιόμενου ορίζοντα (Rolling Horizon Genetic Algorithm - RHGA) που παρέχεται από την πλατφόρμα του διαγωνισμού GVGAI. Σε αυτή την τεχνική, κατά τη διάρκεια κάθε χρονικού βήματος δημιουργείται ένας μικρός πληθυσμός ατόμων-λύσεων κάθε ένα από τα οποία αναπαριστά μία ακολουθία ενεργειών, αξιολογούνται μέσω της συνάρτησης ποιότητας και στη συνέχεια επιλέγεται η πρώτη ενέργεια του ατόμου με την υψηλότερη βαθμολογία.

Στον Πίνακα 3.1 παρουσιάζονται τα ποσοστά νίκης των πρακτόρων για κάθε επίπεδο των δύο παιχνιδιών. Ο προτεινόμενος *Γενετικός Αλγόριθμος με Επαναχρησιμοποίηση της Κωδικοποίησης Κίνησης* (Genetic Algorithm with Motion Encoding Reuse - GAMER) υπερτερεί του RHGA και στα δύο παιχνίδια συνολικά, ωστόσο ο δεύτερος έχει καλύτερη απόδοση σε ορισμένα μεμονωμένα επίπεδα στο παιχνίδι Portals. Πιο συγκεκριμένα, με εξαίρεση το τελευταίο επίπεδο (L4) στο Portals, ο αλγόριθμος GAMER επιτυγχάνει να κερδίσει σε όλα τα υπόλοιπα δοκιμασμένα επίπεδα. Η δυσκολία στο L4 έγκειται στη διαμόρφωση της τοπολογίας στην οποία περιλαμβάνεται μεγάλο πλήθος από NPCs, ωθώντας τον πράκτορα να δίνει συνεχώς προτεραιότητα στην επιβίωση με αποτέλεσμα να αποτυγχάνει να σχεδιάσει αποτελεσματικά μία διαδρομή προς τον τελικό στόχο. Η ιδιαιτερότητα του συγκεκριμένου επιπέδου αποτυπώνεται και στο πολύ χαμηλό ποσοστό νίκης του RHGA. Η συνολική διαφορά στο ποσοστό νίκης μεταξύ των παιχνιδιών Zelda και Portals επιβεβαιώνει τη μεγαλύτερη πολυπλοκότητα του δεύτερου λόγω των περισσότερων διαφορετικών κατηγοριών αντικειμένων, που καθιστούν δυσκολότερο το πρόβλημα της αποδοτικής αναπαράστασης των καταστάσεων. Ωστόσο, εκτός από τις διαφορετικές τοπολογίες στα επίπεδα του ίδιου παιχνιδιού, ο πράκτορας φαίνεται να

προσαρμόζεται στα δύο διαφορετικά παιχνίδια εξίσου αποτελεσματικά χρησιμοποιώντας την ίδια κωδικοποίηση κίνησης.

Όσον αφορά στη μέση βαθμολογία (Πίνακας 3.2), το τελικό σκορ στο παιχνιδι Portals είναι δυαδικό (1.0 σε περίπτωση νίκης, 0.0 σε περίπτωση ήττας) και επομένως η μέση βαθμολογία είναι ουσιαστικά ίση με το ποσοστό νίκης. Αντιθέτως, στο Zelda περισσότερες παράμετροι του παιχνιδιού λαμβάνονται υπόψη (π.χ. επιπλέον πόντοι αποδίδονται στον πράκτορα όταν εξουδετερώνει έναν εχθρό). Δεδομένου ότι οι υπολογισμένες βαθμολογίες βασίζονται σε διαφορετικά κριτήρια, στο πλαίσιο της παρούσας αξιολόγησης αρχικά όλες οι βαθμολογίες κανονικοποιούνται στην κλίμακα $[0, 10]$ προκειμένου να είναι συγκρίσιμες.

Win Percentage (100 games per level)						
	Zelda					
Levels	L0	L1	L2	L3	L4	Overall
RHGA	30.0%	9.0%	7.0%	15.0%	67.0%	25.6%
GAMER	**38.0%**	**11.0%**	**58.0%**	**64.0%**	**77.0%**	**49.6%**
	Portals					
Levels	L0	L1	L2	L3	L4	Overall
RHGA	19.0%	**69.0%**	68.0%	**34.0%**	**3.0%**	38.6%
GAMER	**29.0%**	60.0%	**94.0%**	14.0%	0.0%	**39.4%**

Πίνακας 3.1: Ποσοστό νικών ανά επίπεδο (100 εκτελέσεις) στα παιχνίδα Zelda και Portals για τις δύο προσεγγίσεις (το υψηλότερο ποσοστό σε κάθε επίπεδο παρουσιάζεται εντονότερα)

Normalized Average Scores (100 games per level)												
	Zelda						**Portals**					
Levels	L0	L1	L2	L3	L4	Overall	L0	L1	L2	L3	L4	Overall
RHGA	**8.46**	**6.41**	**6.55**	**7.48**	**8.63**	**7.51**	1.9	**6.9**	6.8	**3.4**	**0.3**	3.86
GAMER	5.44	4.47	6.13	7.43	6.48	5.99	**2.9**	6	**9.4**	1.4	0.0	**3.94**

Πίνακας 3.2: Κανονικοποιημένη μέση βαθμολογία ανά επίπεδο (100 εκτελέσεις) στα παιχνίδα Zelda και Portals για τις δύο προσεγγίσεις (η υψηλότερη βαθμολογία σε κάθε επίπεδο παρουσιάζεται εντονότερα)

Ένα οξύμωρο, εκ πρώτης όψεως, συμπέρασμα που βγαίνει από την αντιστοίχιση του ποσοστού νίκης και της μέσης βαθμολογίας σε κάθε επίπεδο του παιχνιδιού Zelda (Πίνακες 3.1-3.2), είναι ότι δεν φαίνεται να είναι ανάλογα. Αντίθετα, παρόλο που ο αλγόριθμος GAMER πετυχαίνει υψηλότερα ποσοστά νίκης έχει χαμηλότερη μέση βαθμολογία σε όλα τα επίπεδα. Αυτό το φαινόμενο οφείλεται στη δομή της συνάρτησης που υπολογίζει τη βαθμολογία του πράκτορα (η οποία παρέχεται από το περιβάλλον του GVGAI) σε συνδυασμό με τη συνάρτηση ποιότητας του ΓΑ. Από την Εξίσωση 3.2 φαίνεται ότι η βαθμολογία των ατόμων-λύσεων κατά την εκτέλεση του ΓΑ εξαρτάται αποκλειστικά από την εύρεση του κλειδιού και της πόρτας εξόδου. Αυτό έχει σαν αποτέλεσμα ο πράκτορας να μην

επιτίθεται στους εχθρούς αλλά να κατευθύνεται απευθείας στους υποστόχους και η βαθμολογία του να μειώνεται καθώς γίνεται πιο αποτελεσματικός.

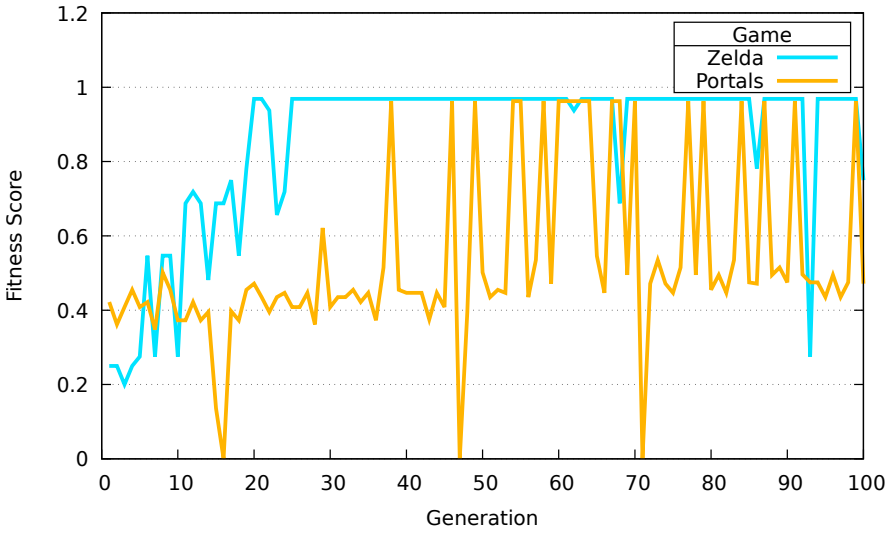
Το γεγονός αυτό επιβεβαιώνεται και από το μέσο πλήθος των βημάτων που εκτελούν οι δύο πράκτορες όπως φαίνεται στον Πίνακα 3.3. Στον προτεινόμενο αλγόριθμο (GAMER) δίνεται έμφαση στην ολοκλήρωση των επιπέδων όσο το δυνατόν γρηγορότερα ενώ στην προσέγγιση του ΓΑ κυλιόμενου ορίζοντα ο πράκτορας αφιερώνει περισσότερο χρόνο στο παιχνίδι με αποτέλεσμα να πετυχαίνει υψηλότερη βαθμολογία (ο χρόνος ολοκλήρωσης των επιπέδων δε λαμβάνεται υπόψη στη συνάρτηση βαθμολόγησης του GVGAI). Όσον αφορά στο παιχνίδι Portals, όπως προαναφέρθηκε η βαθμολογία είναι δυαδική συνεπώς δε μπορούν να εξαχθούν αντίστοιχα συμπεράσματα.

Average Timesteps (100 games per level)						
	Zelda					
Levels	L0	L1	L2	L3	L4	Overall
RHGA	855.2	951.3	918.9	797.8	411.2	786.9
GAMER	**765.7**	**882.5**	**204.3**	**631.0**	**395.2**	**575.7**
	Portals					
	L0	L1	L2	L3	L4	Overall
RHGA	453.8	252.5	597.2	471.5	49.1	364.8
GAMER	**129.0**	**80.9**	**89.7**	**49.9**	**20.9**	**74.1**

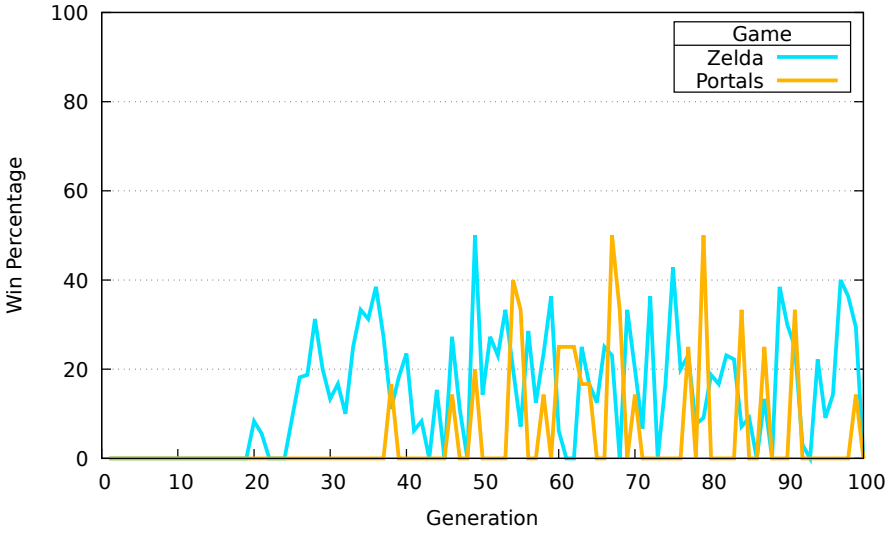
Πίνακας 3.3: Μέσος αριθμός βημάτων ανά επίπεδο (100 εκτελέσεις) στα παιχνίδα Zelda και Portals για τις δύο προσεγγίσεις (ο χαμηλότερος αριθμός βημάτων σε κάθε επίπεδο παρουσιάζεται εντονότερα)

Σχετικά με τη διαδικασία εξέλιξης του ΓΑ, στο Σχήμα 3.4 φαίνεται η ποιότητα του καλύτερου ατόμου-λύσης σε κάθε γενιά και στο Σχήμα 3.5 το ποσοστό νίκης ανά γενιά κατά την εκπαίδευση του ΓΑ για τα δύο παιχνίδια. Αναμενόμενα, η ποιότητα των ατόμων στις πρώτες γενιές είναι υψηλότερη στο παιχνίδι Portals καθώς τα χρωμοσώματα κίνησης που υπολογίστηκαν στο Zelda χρησιμοποιούνται αυτούσια, με αποτέλεσμα ένα μέρος της απαραίτητης πληροφορίας να είναι ήδη γνωστό στο ξεκίνημα της εκαπίδευσης του ΓΑ και τα άτομα-λύσεις να αποδίδουν καλύτερα από τα αντίστοιχα άτομα στο Zelda. Παρ΄ όλα αυτά, ο πρώτος πράκτορας που ολοκληρώνει με επιτυχία το επίπεδο στο Portals εμφανίζεται στην τριακοστή όγδοη γενιά ενώ αντίστοιχα στο Zelda χρειάζονται είκοσι γενιές, επιβεβαιώνοντας το μεγαλύτερο βαθμό δυσκολίας του πρώτου.

Αναφορικά με το ποσοστό νικών (Σχήμα 3.5) φαίνεται πως υπάρχει μία τάση αύξησης καθώς αυξάνεται ο αριθμός των γενιών της εξελικτικής διαδικασίας, η οποία όμως δεν είναι σταθερή. Αυτή η συμπεριφορά είναι αναμενόμενη καθώς ο στόχος του ΓΑ είναι η εύρεση της βέλτισης λύσης χωρίς αυτό να σημαίνει ότι όλα τα άτομα μίας γενιάς θα είναι καλύτερα από τα άτομα της προηγούμενης. Αυτό σε συνδυασμό και με τη στοχαστικότητα του περιβάλλοντος στο οποίο εφαρμόζεται έχει ως αποτελεσμα αυτές τις αυξομειώσεις στο μέσο ποσοστό νίκης ανα γενιά.

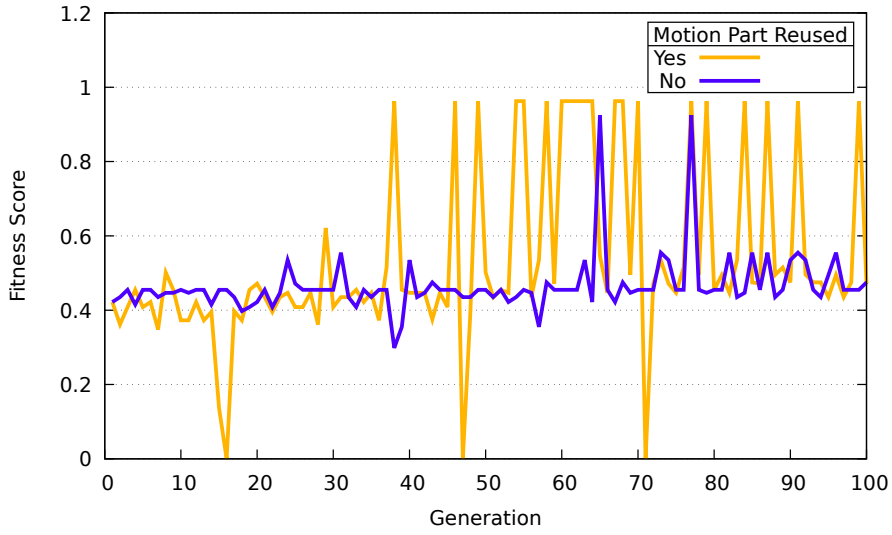


Σχήμα 3.4: Υψηλότερη ποιότητα ατόμου-λύσης ανά γενιά



Σχήμα 3.5: Ποσοστό νίκης ανά γενιά

Στο Σχήμα 3.6 παρουσιάζεται η ποιότητα της βέλτιστης λύσης ανά γενιά για το παιχνίδι Portals στην περίπτωση της επαναχρησιμοποίησης των χρωμοσωμάτων κίνησης και στην περίπτωση που ο ΓΑ εκτελείται εξολοκλήρου από την αρχή. Είναι ξεκάθαρο ότι στην πρώτη περίπτωση ο αλγόριθμος συγκλίνει πολύ πιο γρήγορα και άτομα-λύσεις ικανά να ολοκληρώσουν επιτυχώς το επίπεδο εμφανίζονται με μεγαλύτερη συχνότητα. Συγκεκριμένα, στην περίπτωση που ο ΓΑ εκτελείται χωρίς πρότερη γνώση το πρώτο άτομο που πετυχαίνει νίκη ανήκει στην εξηκοστή όγδοη γενιά. Στην περίπτωση που επαναχρησιμοποιούνται τα χρωμοσώματα κίνησης, το πρώτο νικηφόρο άτομο προκύπτει στην τριακοστή όγδοη γενιά, κάτι που οφείλεται στο μικρότερο αριθμό χρωμοσωμάτων που πρέπει να προσδιοριστούν από τον ΓΑ σε αυτή την περίπτωση και στην ήδη υπάρχουσα πληροφορία που προκύπτει από το επαναχρησιμοποιούμενο τμήμα του γενότυπου.



Σχήμα 3.6: Υψηλότερη ποιότητα ατόμου-λύσης ανά γενιά

Συμπερασματικά, η χρήση του ΓΑ ως αυτόνομης τεχνικής για την ανάπτυξη ενός ευφυούς πράκτορα οδήγησε σε ικανοποιητικά επίπεδα απόδοσης, ιδίως αν λάβουμε υπόψη το χαμηλό υπολογιστικό κόστος σε σύγκριση με άλλες μεθόδους. Σε αυτό συνέβαλε και η αναπαράσταση των καταστάσεων με N-πλειάδες, μειώνοντας το συνολικό πλήθος των γονιδίων προς εκπαίδευση. Από την πειρματική διαδικασία είναι εμφανές ότι η συνάρτηση ποιότητας παίζει καθοριστικό ρόλο στη συνολική συμπεριφορά του πράκτορα, κάτι που αποτυπώνεται και στα ποσοστά νίκης σε συνδυασμό με τη μέση βαθμολογία ανά επίπεδο. Ανάλογα με τη σχεδιαστική επιλογή της συνάρτησης ποιότητας, ο πράκτορας δίνει βαρύτητα σε διαφορετικά στοιχεία του περιβάλλοντος και η εκπαίδευση προσαρμόζεται αντίστοιχα. Τέλος, η επαναχρησιμοποίηση ενός υποσυνόλου των εκπαιδευμένων χρωμοσωμάτων σε ένα διαφορετικό περιβάλλον αποδείχθηκε ιδιαίτερα επιδραστική στην ταχύτητα σύγκλισης του ΓΑ. Ειδικότερα, φαίνεται ότι η εκπαίδευση του κοινού τμήματος του γενότυπου αρχικά σε ένα απλούστερο περιβάλλον και στη συνέχεια η εφαρμογή του σε ένα πολυπλοκότερο, μπορεί να έχει πολλαπλά οφέλη στο πλαίσιο της μεταφοράς μάθησης.

Κεφάλαιο 4

Ενίσχυση Φάσης Προσομοίωσης στη Δενδρική Αναζήτηση Μόντε Κάρλο

Ο αλγόριθμος δενδρικής αναζήτησης Μόντε Κάρλο συγκλίνει στην επιλογή της βέλτισης κίνησης (ως επακόλουθο του ότι οι προσεγγίσεις της αξίας των κόμβων συγκλίνουν στις πραγματικές τους τιμές) καθώς ο αριθμός των προσομοιώσεων τείνει στο άπειρο και δεδομένης άπειρης υπολογιστικής μνήμης [30]. Στην πράξη, ο επιτρεπόμενος αριθμός προσομοιώσεων καθορίζεται από τους διαθέσιμους υπολογιστικούς πόρους αλλά και από χρονικούς περιορισμούς, συνεπώς κάτι τέτοιο δεν είναι εφικτό. Ως εκ τούτου, η ποιότητα της επιλεγμένης ενέργειας εξαρτάται σε μεγάλο βαθμό από την αξιοπιστία των αποτελεσμάτων των προσομοιώσεων του αλγορίθμου. Σε αυτό το κεφάλαιο εξετάζονται μέθοδοι ενίσχυσης της φάσης προσομοίωσης της δενδρικής αναζήτησης Μόντε Κάρλο, προκειμένου να υπολογίζεται η αξία των κόμβων με μεγαλύτερη ακρίβεια εντός του περιορισμένου χρονικού ορίζοντα.

4.1 Περιβάλλον Υλοποίησης

Για την υλοποίηση και την αξιολόγηση των τεχνικών που αναλύονται στη συνέχεια επιλέχθηκε το περιβάλλον *Metastone* [32], το οποίο είναι ένας *ανοιχτού κώδικα* (open source) προσομοιωτής του παιχνιδιού *Hearhstone* [10], ειδικά σχεδιασμένος για την υλοποίηση ευφυών πρακτόρων και την εφαρμογή τους στο περιβάλλον του παιχνιδιού. Συγκεκριμένα, παρέχει τους βασικούς μηχανισμούς που καθορίζουν τη μετάβαση των καταστάσεων δεδομένων των ενεργειών που εκτελούνται καθώς και ένα μοντέλο προσομοίωσης παρτίδας, που είναι απαραίτητο για την υλοποίηση της δενδρικής αναζήτησης Μόντε Κάρλο. Επιπλέον, περιλαμβάνει ένα σύνολο πρακτόρων τεχνητής νοημοσύνης που μπορούν να χρησιμοποιηθούν ως μοντέλα αναφοράς για την τελική αξιολόγηση των αλγορίθμων.

Το Hearhstone, είναι ένα ανταγωνιστικό, συλλεκτικό παιχνίδι καρτών δύο παικτών, το οποίο ενδείκνυται για τη δοκιμή μεθόδων τεχνητής νοημοσύνης και έχει χρησιμοποιηθεί σε πολλές περιπτώσεις στην αντίστοιχη βιβλιογραφία. Σε αυτό συμβάλλουν δύο επιπλέον χαρακτηριστικά που το καθιστούν κατάλληλο για δοκιμές ερευνητικού σκοπού, η στοχαστικότητα και η μερική παρατηρησιμότητα. Η στοχαστικότητα απαντάται σε δύο άξονες· αφενός οι παίκτες τραβούν μία τυχαία κάρτα

από την τράπουλα σε κάθε νέο γύρο του παιχνιδιού και αφετέρου ορισμένες κάρτες μπορεί να έχουν διαφορετική επίδραση στην κατάσταση του παιχνιδιού τη στιγμή που παίζονται. Όσον αφορά στην παρατηρησιμότητα, οι παίκτες δε γνωρίζουν τις κάρτες που έχει ο αντίπαλος τους στο χέρι του, συνεπώς δεν είναι γνωστή η πλήρης κατάσταση του παιχνιδιού ανά πάσα στιγμή. Οι δύο αυτές ιδιότητες προσθέτουν επιπλέον πολυπλοκότητα και επιτρέπουν την ευκολότερη διαβάθμιση των διαφορετικών μεθόδων.

Αναφορικά με τη στρατηγική του παιχνιδιού, υπάρχουν τρία βασικά αρχέτυπα για τις τράπουλες των παιχτών (ένα για κάθε στρατηγική) τα οποία χρησιμοποιούνται κατά κόρον. Οι τρεις στρατηγικές παιχνιδιού ακολουθούν το μοτίβο πέτρα-ψαλίδι-χαρτί, δηλαδή κάθε μία υπερτερεί της μίας και υστερεί της άλλης. Ως εκ τούτου, προκειμένου να μην υπάρχει μεροληψία στην αξιολόγηση των ευφυών πρακτόρων, όλοι οι αλγόριθμοι δοκιμάζονται εναντίον των υπολοίπων χρησιμοποιώντας το ίδιο αρχέτυπο (και για τις τρεις περιπτώσεις).

4.2 Μεθοδολογία

4.2.1 Μοντέλο Αξιολόγησης

Στην απλή εκδοχή της δενδρικής αναζήτησης Μόντε Κάρλο, όταν προστίθεται ένας κόμβος στο δέντρο εκτελείται προσομοίωση μίας πλήρους παρτίδας με αφετηρία το νέο κόμβο και τα στατιστικά του δέντρου ανανεώνονται με βάση το αποτέλεσμα της προσομοίωσης. Καθώς η πολιτική που ακολουθείται κατά τη διάρκεια της προσομοίωσης είναι κατά βάση τυχαία, αναμενόμενα εισάγεται υψηλού βαθμού διακύμανση και απαιτείται πολύ μεγάλος αριθμός επαναλήψεων προκειμένου τα αποτελέσματα να είναι αξιόπιστα όσον αφορά στην αξιολόγηση των κόμβων. Για την αντιμετώπιση αυτού του φαινομένου προτείνεται η ενσωμάτωση στον αλγόριθμο ενός ταξινομητή, ο οποίος θα προβλέπει το αποτέλεσμα της παρτίδας από μία δεδομένη κατάσταση-κόμβο χωρίς να απαιτείται κάθε φορά η εκτέλεση της αντίστοιχης προσομοίωσης.

Για την εκπαίδευση του ταξινομητή χρησιμοποιείται ένα ειδικά σχεδιασμένο σύνολο δεδομένων που δημιουργήθηκε για το συγκεκριμένο περιβάλλον. Συγκεκριμένα, συγκεντρώθηκαν δείγματα της μορφής (*κατάσταση_παιχνιδιού*, *νικητής*) από προσομοιωμένα παιχνίδια στα οποία και οι δύο παίχτες αποτελούνταν από έναν απλό πράκτορα δενδρικής αναζήτησης Μόντε Κάρλο. Για την αναπαράσταση της κατάστασης χρησιμοποιήθηκε ένα *διάνυσμα χαρακτηριστικών* (feature vector) του παιχνιδιού που μπορούν να χωριστούν σε τρεις υποκατηγορίες· χαρακτηριστικά των παιχτών, του τραπεζιού και γενικά χαρακτηριστικά (Πίνακας 4.1). Το σύνολο των δεδομένων αποτελείται από τρία εκατομμύρια δείγματα που συλλέχθηκαν από παρτίδες και με τα τρία διαφορετικά αρχέτυπα.

Το μοντέλο πρόβλεψης είναι ένας ταξινομητής *ακραίας ενίσχυσης κλίσης* (extreme gradient boosting - xgboost). Συνολικά εκπαιδεύτηκαν τέσσερις διαφορετικοί ταξινομητές, ένας για κάθε αρχέτυπο και ένας γενικός χρησιμοποιώντας όλα τα δεδομένα, προκειμένου να εξεταστεί κατά πόσο επηρεάζεται η απόδοση του πράκτορα από τη στοχευμένη εστίαση σε κάποια συγκεκριμένη στρατηγική και να αξιολογηθεί η δυνατότητα υλοποίησης ενός γενικού πράκτορα για όλες τις περιπτώσεις. Οι υπερπαράμετροι του μοντέλου προσδιορίστηκαν έπειτα από *αναζήτηση πλέγματος* (grid search) ως εξής:

- Αριθμός εκτιμητών (δέντρων): 1000
- Μέγιστο βάθος: 15

Κατηγορίες	**Χαρακτηριστικά**
Παίχτες	πόντοι ζωής παίχτη πόντοι επίθεσης παίχτη υπολειπόμενοι πόντοι-mana (μόνο για τον ενεργό παίχτη) αριθμός καρτών στο χέρι αριθμός μαγικών καρτών στο χέρι (μόνο για τον ενεργό παίχτη) αριθμός ενεργών μυστικών
Τραπέζι	αριθμός minions συνολικό κόστος (σε mana) των minions συνολικοί πόντοι επίθεσης των minions συνολικοί πόντοι ζωής των minions αριθμός minions που μπορούν να επιτεθούν συνολικοί πόντοι επίθεσης των minions που μπορούν να επιτεθούν αριθμός minions με taunt (ειδικό χαρκατηριστικό)
Γενικά	γύρος του παιχνιδιού ενεργός παίχτης πρώτος παίχτης

Πίνακας 4.1: Χαρακτηριστικά διανύσματος κατάστασης

- Ελάχιστο βάρος παιδιού: 7
- Γάμμα: 0.1
- Ρυθμός μάθησης: 0.2

Στον Πίνακα 4.2 παρουσιάζεται η ακρίβεια των διαφορετικών μοντέλων ανά σύνολο δεδομένων εκπαίδευσης. Οι κατηγορίες *warlock*, *hunter* και *shaman* αντιστοιχούν στα τρία διαφορετικά αρχέτυπα. Τα μοντέλα που έχουν εκπαιδευτεί σε συγκεκριμένα υποσύνολα αξιολογούνται μόνο σε αυτά (δεν έχει νόημα να αξιολογηθούν στα υπόλοιπα δεδομένα εφόσον η υλοποίηση τους αφορά μόνο το συγκεκριμένο πρόβλημα) ενώ το γενικό μοντέλο αξιολογείται τόσο στο σύνολο των δεδομένων όσο και σε κάθε κατηγορία ξεχωριστά. Παρατηρείται ότι η απόδοση των μοντέλων διαφέρει αρκετά μεταξύ των διαφορετικών κατηγοριών, το οποίο οφείλεται στη μορφή των δεδομένων εκπαίδευσης (ο βαθμός δυσκολίας πρόβλεψης του τελικού αποτελέσματος από μία κατάσταση εξαρτάται από την κατηγορία). Πέραν αυτού, αναμενόμενα το γενικό μοντέλο υστερεί έναντι των ειδικά εκπαιδευμένων ταξινομητών στα αντίστοιχα υποσύνολα δεδομένων, χωρίς ωστόσο οι διαφορές να είναι αποτρεπτικές όσον αφορά τη χρήση του.

δεδομένα εκπαίδευσης	**warlock**	**hunter**	**shaman**	**συνολικά**
warlock	0.798	-	-	-
hunter	-	0.871	-	-
shaman	-	-	0.852	-
συνολικά	0.789	0.844	0.835	0.815

Πίνακας 4.2: Ακρίβεια ταξινομητών ανά σύνολο δεδομένων εκπαίδευσης

Οι ετικέτες του συνόλου δεδομένων παίρνουν τιμές στο {0, 1} (ήττα ή νίκη) καθώς δεν υπάρχει κάποια προφανής αξιόπιστη μετρική με την οποία να βαθμολογείται η τελική κατάσταση σε συνεχές διάστημα (π.χ. στο $[0, 1]$). Ως εκ τούτου, η πρόβλεψη του μοντέλου αξιολόγησης είναι δυαδική. Αυτό έχει σαν αποτέλεσμα παρότι το μοντέλο έχει υψηλή ακρίβεια πρόβλεψης, πολλοί κόμβοι να έχουν την ίδια αξιολόγηση (ειδικά στις πρώτες επαναλήψεις του αλγορίθμου κατά τις οποίες ο κάθε κόμβος έχει διατρεχθεί λίγες φορές). Προκειμένου να μπορούν να διαχωριστούν οι καταστάσεις με παρόμοια βαθμολογία, προτείνεται ο συνδυασμός της πρόβλεψης του μοντέλου με το αποτέλεσμα μίας προσομοίωσης (Εξίσωση 4.1). Η παράμετρος λ παίρνει τιμές στο [0, 1] και καθορίζει τη συνεισφορά του κάθε όρου στη συνολική αξία του κόμβου ($xgb_{pred}(s)$ είναι η πρόβλεψη του μοντέλου για μία δεδομένη κατάσταση παιχνιδιού s ενώ $z(s)$ είναι το αποτέλεσμα μίας τυχαίας προσομοίωσης). Με αυτό τον τρόπο, αφενός κάθε κόμβος έχει μία πιο αντικειμενική βαθμολογία χάρη στη χρήση του ταξινομητή, αφετέρου αξιοποιείται ένα μέρος της τυχαιότητας που εισάγουν οι προσομοιώσεις για την ταχύτερη διαβάθμιση των στατιστικών των διαφορετικών κόμβων.

$$combined\_score(s) = \lambda xgb_{pred}(s) + (1 - \lambda)\, z(s) \tag{4.1}$$

Τα πειράματα που παρουσιάζονται στη συνέχεια εκτελέστηκαν με λ=0.8, ωστόσο δεν παρατηρήθηκαν μεγάλες διακυμάνσεις στην απόδοση για διαφορετικές τιμές της υπερπαραμέτρου.

4.2.2 Πρώιμη Προσομοίωση

Όπως αναφέρθηκε παραπάνω, ένα πρόβλημα που παρουσιάζεται στη δενδρική αναζήτηση Μόντε Κάρλο είναι ο διαχωρισμός των αξιών των καταστάσεων στα πρώτα στάδια του αλγορίθμου που οδηγεί συχνά στην τυχαία επιλογή ενός κόμβου προς επέκταση και μπορεί να καθυστερήσει την αναζήτηση εξερευνώντας κόμβους με χαμηλή αναμενόμενη αξία. Προς αυτή την κατεύθυνση, προτείνεται μία τροποποίηση του αλγορίθμου που βασίζεται σε πρώιμη προσομοίωση.

Σε αυτή την παραλλαγή, αρχικά επιλέγεται ένα κατώφλι t ελάχιστων γύρων του παιχνιδιού που απαιτούνται προκειμένου οι προβλέψεις του μοντέλου να θεωρούνται αξιόπιστες. Στην περίπτωση που η παρτίδα βρίσκεται σε πολύ πρώιμο στάδιο (έχουν παιχτεί λιγότεροι από t γύρους) προσομοιώνουμε πρώτα τυχαίες κινήσεις (και για τους δύο παίχτες) μέχρι το παιχνίδι να φτάσει στον προκαθορισμένο γύρο t και στη συνέχεια χρησιμοποιείται η πρόβλεψη του μοντέλου από τη νέα κατάσταση. Όταν η παρτίδα φτάσει το κατώφλι t, ο αλγόριθμος συνεχίζει να εκτελείται κανονικά. Η βασική διαίσθηση πίσω από την προτεινόμενη τεχνική είναι ότι όσο πιο κοντά βρίσκεται το παιχνίδι σε μία τερματική κατάσταση τόσο πιο ακριβείς θα είναι οι προβλέψεις του μοντέλου, καθώς οι περισσότερες από τις πιθανές καταστάσεις στους πρώτους γύρους είναι πολύ παρόμοιες δυσκολεύοντας τον ταξινομητή να τις διακρίνει. Αυτό έχει ως αποτέλεσμα οι προβλέψεις που γίνονται στα πρώτα στάδια του παιχνιδιού να ενδέχεται να παραπλανήσουν την πολιτική αναζήτησης και να οδηγήσουν σε μη βέλτιστες ενέργειες στους πρώτους γύρους του παιχνιδιού.

4.2.3 Στοχαστική Αξιολόγηση Κόμβων

Ένα άλλο χαρακτηριστικό το οποίο φαίνεται να επηρεάζει αρκετά την αποτελεσματικότητα του αλγορίθμου είναι ο αριθμός των πιθανών ενεργειών σε κάθε επανάληψη. Καθώς ο χώρος ενεργειών μεγαλώνει (που συνεπακόλουθα οδηγεί σε μείωση του αριθμού των διαθέσιμων επαναλήψεων ανά κόμβο), η χρήση τυχαίων προσομοιώσεων στην Εξίσωση 4.1 έχει θετικό αντίκτυπο στην τελική τιμή

της αξίας κάθε κόμβου. Αντίθετα, σε καταστάσεις με λιγότερες διαθέσιμες ενέργειες, καθεμία από αυτές αξιολογείται συχνότερα (και άρα ακριβέστερα) και η εκτέλεση τυχαίων προσομοιώσεων μπορεί να εισάγει επιπλέον διακύμανση επηρεάζοντας δυσμενώς την απόδοση του πράκτορα.

Προκειμένου να αντιμετωπιστεί αυτή η συμπεριφορά, εισάγουμε στον ευφυή πράκτορα τη δυνατότητα στοχαστικής αξιολόγησης των κόμβων λαμβάνοντας υπόψη το εκάστοτε πλήθος των διαθέσιμων ενεργειών. Με βάση αυτή την προσέγγιση, η τελική αξιολόγηση της κατάστασης προκύπτει είτε ως ο συνδυασμός της πρόβλεψης του μοντέλου με το αποτέλεσμα μίας τυχαία εκτελεσμένης προσομοίωσης (Εξίσωση 4.1) είτε ως απλώς η πρόβλεψη του μοντέλου, με πιθανότητα ανάλογη του μεγέθους του χώρου ενεργειών. Η τελική βαθμολογία του κάθε κόμβου υπλογίζεται όπως φαίνεται στην Εξίσωση 4.2.

$$final\_score(s) = \begin{cases} xgb_{pred}(s), \text{if } u \geq action\_space * \mu \\ combined\_score(s), \text{otherwise} \end{cases} \tag{4.2}$$

όπου $u \in [0, 1]$ είναι ένας τυχαίος αριθμός από την ομοιόμορφη κατανομή $U(0, 1)$.

Ο Αλγόριθμος 4 περιγράφει τη διαδικασία της αξιολόγησης των κόμβων κατά τη δενδρική αναζήτηση Μόντε Κάρλο με τις προαναφερθείσες βελτιστοποιήσεις. Αρχικά, ελέγχεται το στάδιο της παρτίδας και σε περίπτωση που δεν έχει φτάσει το ελάχιστο κατώφλι εκτελείται πρώιμη προσομοίωση. Εν συνεχεία, υπολογίζεται ένα κατώφλι πιθανότητας ανάλογο του μεγέθους του χώρου ενεργειών της τρέχουσας κατάστασης και καλείται το μοντέλο ταξινόμησης για την παραγωγή της πρόβλεψης της αξίας. Τέλος, υπολογίζεται στοχαστικά η συνολική αξία του νέου κόμβου.

Algorithm 4: MCTS node evaluation

```
1  Function Rollout(node):
2      state_z ← node.getState()
3      while node.turn ≤ t do
4          new_node ← simulate(random_action)
5      state_xgb ← new_node.getState()
6      u ∼ U(0, 1)
7      prob_threshold ← action_space * μ
8      if u ≥ prob_threshold then
9          return xgb_pred(state_xgb)
10     else
11         return λxgb_pred(state_xgb) + (1 − λ)z(state_z)
12
```

4.2.4 Επαναχρησιμοποίηση Δέντρου

Μία ευρέως χρησιμοποιούμενη τεχνική για τη βελτίωση του αλγορίθμου MCTS είναι η *επαναχρησιμοποίηση δέντρου* (tree reuse). Στην απλή υλοποίηση του αλγορίθμου, καθε φορά που επιλέγεται μία ενέργεια το δέντρο παιχνιδιού που έχει δημιουργηθεί απορρίπτεται και η επόμενη κατάσταση (που

προκύπτει από την εκτέλεση της ενέργειας αυτής) ορίζεται ως ο κόμβος-ρίζα του νέου δέντρου. Στόχος της επαναχρησιμοποίησης δέντρου είναι να διατηρήσει τις τιμές των κόμβων στο υποδέντρο που δημιουργήθηκε κάτω από την επιλεγμένη ενέργεια (δηλαδή το υποδέντρο κάτω από τη ρίζα του νέου δέντρου), έτσι ώστε να εκμεταλλευτεί την υπάρχουσα πληροφορία των ήδη εκτελεσμένων προσομοιώσεων αντί να δημιουργηθεί το νέο δέντρο από την αρχή. Αυτή η ιδέα παρουσιάστηκε αρχικά στο [77] και εφαρμόστηκε για το παιχνίδι *MS Pac-Man*, αποδίδοντας ικανοποιητικά αποτελέσματα.

Το κύριο μειονέκτημα αυτής της τεχνικής έγκειται στο γεγονός ότι απαιτεί πλήρως παρατηρήσιμα, ντετερμινιστικά περιβάλλοντα προκειμένου να μπορεί να εφαρμοστεί σταδιακά κατά τη διάρκεια όλων των μεταβάσεων του παιχνιδιού. Στο παρόν περιβάλλον, οι κάρτες στο χέρι του αντιπάλου (και επομένως το σύνολο των πιθανών ενεργειών) είναι άγνωστες και καθορίζονται στην αρχή της διαδικασίας του αλγορίθμου. Αυτό σημαίνει ότι το σύνολο των πιθανών ενεργειών που λαμβάνονται υπόψη κατά τη δενδρική αναζήτηση δεν ταυτίζεται απαραίτητα με τον πραγματικό χώρο ενεργειών. Ως εκ τούτου, ένα υποσύνολο των πραγματικών ενεργειών του αντιπάλου δεν μπορεί να προσομοιωθεί στο δημιουργημένο δέντρο παιχνιδιού. Προκειμένου να ενσωματωθεί στον προτεινόμενο αλγόριθμο η επαναχρησιμοποίηση δέντρου, τροποποείται κατάλληλα ώστε να εφαρμόζεται στην ακολουθία των ενεργειών που εκτελούνται από τον πράκτορα. Όταν ο γύρος ενεργειών του πράκτορα ολοκληρώνεται, το δέντρο παιχνιδιού απορρίπεται και η τελευταία ενέργεια που επιλέγεται στη σειρά του αντιπάλου χρησιμοποιείται στη συνέχεια ως νέα ρίζα για το επόμενο δέντρο αναζήτησης του πράκτορα.

4.3 Αποτελέσματα

Για την αξιολόγηση του πράκτορα χρησιμοποιήθηκαν ως μέτρο σύγκρισης δύο ευφυείς πράκτορες βασισμένοι στον απλό αλγόριθμο δενδρικής αναζήτησης Μόντε Κάρλο και στον αλγόριθμο *Αξίας Κατάστασης Παιχνιδιού* (Game State Value - GSV) αντίστοιχα. Ο αλγόριθμος αξίας κατάστασης παιχνιδιού παρέχεται από το Metastone και είναι ο πιο αποτελεσματικός αλγόριθμος στο συγκεκριμένο περιβάλλον. Η λειτουργία του βασίζεται στον αλγόριθμο *minimax* [100], ο οποίος ενισχύεται με μία ευριστική συνάρτηση ειδικά σχεδιασμένη για το συγκεκριμένο παίχνιδι προκειμένου να περιοριστεί το βάθος αναζήτησης. Η απόδοση του προτεινόμενου πράκτορα αξιολογείται σε ποσοστό νίκης εναντίον των συγκρινόμενων τεχνικών σε μία σειρά από προσομοιωμένα παιχνίδια.

Όπως αναφέρθηκε παραπάνω, υπάρχουν τρεις βασικές στρατηγικές παιχνιδιού οι οποίες μπορεί να οδηγήσουν σε εκ των προτέρων πλεονέκτημα του ενός πράκτορα έναντι του άλλου. Για αυτό το λόγο, για κάθε ζευγάρι πρακτόρων προσομοιώνονται τρία διαφορετικά σύνολα μεταξύ τους παιχνιδιών (ένα για κάθε πιθανή στρατηγική), σε κάθε ένα εκ των οποίων και οι δύο πράκτορες χρησιμοποιούν το ίδιο σετ καρτών ώστε να εξασφαλιστεί η αντικειμενική αξιολόγησή τους. Εξετάζονται πέντε διαφορετικοί πράκτορες στους οποίους προστίθενται σταδιακά νεές τροποποιήσεις προκειμένου να διερευνήσουμε την επίδραση των διαφορετικών τεχνικών. Αρχικά, εξετάζουμε μόνο την επίδραση του ταξινομητή (MCTS-xgboost), έπειτα την προσθήκη πρώιμης προσομοίωσης (MCTS-xgboostSim), στη συνέχεια τη χρήση συνδυασμένης βαθμολογίας από τον ταξινομητή και μία τυχαία προσομοίωση (MCTS-xgboostComb) και τέλος τη στοχαστική αξιολόγηση (MCTS-xgboostSLE). Τα χαρακτηριστικά του κάθε πράκτορα φαίνονται στον Πίνακα 4.3.

Στους Πίνακες 4.4 και 4.5 παρουσιάζονται τα ποσοστά νίκης (σε σύνολο 200 παιχνιδιών) των διαφορετικών παραλλαγών του προτεινόμενου πράκτορα έναντι των προαναφερθέντων αλγορίθμων αναφοράς. Οι πράκτορες του Πίνακα 4.4 κάνουν χρήση των εξειδικευμένων ταξινομητών ενώ οι πράκτορες στον Πίνακα 4.5 ενσωματώνουν το γενικό μοντέλο που έχει εκπαιδευτεί στο σύνολο των

	Ταξινομητής	Πρώιμη προσ.	Συνδυασμός	Στοχ. αξιολ.
MCTS				
MCTS-xgboost	✓			
MCTS-xgboostSim	✓	✓		
MCTS-xbgboostComb	✓	✓	✓	
MCTS-xgboostSLE	✓	✓	✓	✓

Πίνακας 4.3: Χαρακτηριστικά των διαφορετικών πρακτόρων

δεδομένων. Όλοι οι πράκτορες έχουν περιθώριο εκτέλεσης 500 επαναλήψεων ανά ενέργεια. Αρχικά, από τη σύγκριση του απλού αλγορίθμου δενδρικής αναζήτησης Μόντε Κάρλο με τον αλγόριθμο αξίας κατάστασης παιχνιδιού παρατηρείται ότι όπως και με την εκπαίδευση του ταξινομητή, ο τύπος στρατηγικής (και επακολούθως της τράπουλας) που χρησιμοποιείται παίζει μεγάλο ρόλο στο τελικό αποτέλεσμα. Συγκεκριμένα, παρότι ο MCTS υστερεί και στις τρεις περιπτώσεις, υπάρχει πολύ μεγάλη διακύμανση στο ποσοστό νίκης ανά περίπτωση που φτάνει έως 36.5% (hunter) ενώ στη χειρότερη περίπτωση πετυχαίνει μόλις 4.0% (shaman). Συνεπώς, δε μπορεί να οριστεί κάποιο καθολικό αποδεκτό όριο και κάθε περίπτωση εξετάζεται ξεχωριστά.

Είναι ξεκάθαρο ότι η χρήση του ταξινομητή συμβάλλει σε μεγάλη αύξηση της απόδοσης σε σχέση με την απλή δενδρική αναζήτηση Μόντε Κάρλο. Σε αντίθεση με το τυχαίο αποτέλεσμα μίας απλής προσομοίωσης, η εισαγωγή γνώσης πεδίου μέσω του μοντέλου οδηγεί σε ακριβέστερες εκτιμήσεις της αξίας των καταστάσεων σε μικρότερο αριθμό επαναλήψεων, με αποτέλεσμα η αναζήτηση να επικεντρώνεται γρηγορότερα στους πιο υποσχόμενους κόμβους. Αυτό επιβεβαιώνεται τόσο από τη σύγκριση του ενισχυμένου πράκτορα (MCTS-xgboost) με τον απλό (MCTS), όπου ο πράκτορας με το ενσωματωμένο μοντέλο πρόβλεψης υπερισχύει (με ποσοστό κατά μέσο όρο 58.7% και 59.3% για τους πράκτορες με τα εξειδικευμένα και το γενικό μοντέλο αντίστοιχα) όσο και από τη σύγκριση του πρώτου με τον αλγόριθμο αξίας κατάστασης στην οποία παρουσιάζει ιδιαίτερα αυξημένη απόδοση σε σχέση με τον απλό αλγόριθμο (MCTS). Η βελτίωση είναι σημαντικά μεγαλύτερη στις περιπτώσεις που ο απλός MCTS είχε αρχικά χαμηλό ποσοστό επιτυχίας (warlock και shaman) ενώ στην τρίτη στρατηγική (hunter) που είναι λιγότερο απαιτητική, η διαφορά είναι λιγότερη εμφανής.

	MCTS			**GSV**		
	warlock	hunter	shaman	warlock	hunter	shaman
MCTS	50.0	50.0	50.0	12.5	36.5	4.0
MCTS-xgboost	62.5	62.0	51.5	33.0	41.0	8.0
MCTS-xgboostSim	69.0	57.0	46.5	35.5	45.5	11.5
MCTS-xbgboostComb	**76.5**	63.5	**71.0**	**46.5**	**51.5**	20.0
MCTS-xgboostSLE	**76.5**	**66.0**	66.5	45.0	51.0	**22.0**

Πίνακας 4.4: Ποσοστό νίκης εναντίον MCTS και GSV ανά αρχέτυπο για 500 επαναλήψεις με εξειδικευμένα μοντέλα (το υψηλότερο ποσοστό ανά τύπο τράπουλας και αλγόριθμο αναφοράς παρουσιάζεται εντονότερα)

	MCTS			**GSV**		
	warlock	hunter	shaman	warlock	hunter	shaman
MCTS	50.0	50.0	50.0	12.5	36.5	4.0
MCTS-xgboost	68.0	62.5	47.5	30.5	36.0	12.5
MCTS-xgboostSim	69.5	57.5	62.0	34.0	45.0	13.0
MCTS-xbgboostComb	**77.5**	67.0	74.5	42.5	**53.5**	22.5
MCTS-xgboostSLE	75.5	**68.5**	**75.5**	**43.0**	50.5	**25.0**

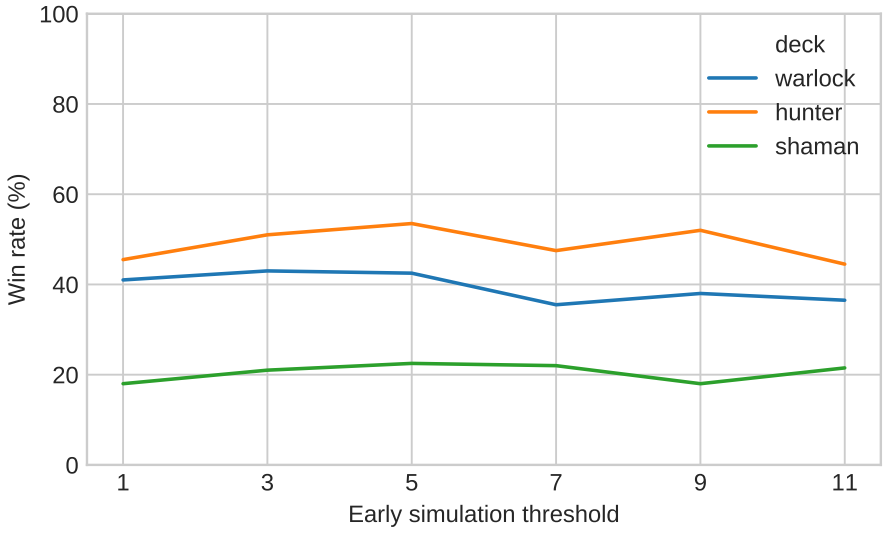
Πίνακας 4.5: Ποσοστό νίκης εναντίον MCTS και GSV ανά αρχέτυπο για 500 επαναλήψεις με το γενικό μοντέλο (το υψηλότερο ποσοστό ανά τύπο τράπουλας και αλγόριθμο αναφοράς παρουσιάζεται εντονότερα)

Σχετικά με την πρώιμη προσομοίωση, η απόδοση του αλγορίθμου βελτιώνεται στις περισσότερες περιπτώσεις, ωστόσο παρατηρούνται και περιπτώσεις όπου ο πράκτορας χωρίς το συγκεκριμένο χαρακτηριστικό (MCTS-xgboost) πετυχαίνει καλύτερα αποτελέσματα απέναντι στον απλό (MCTS). Στη σύγκριση με τον GSV, ο πράκτορας που κάνει χρήση της πρώιμης προσομοίωσης είναι βελτιωμένος σε όλες τις περιπτώσεις. Ιδιαίτερα, σημαντικότερη βελτίωση παρατηρείται για εκείνους τους τύπους στρατηγικής που τείνουν να τελειώνουν το παιχνίδι πιο γρήγορα. Αυτό οφείλεται πιθανότατα στο ότι σε αυτές τις παρτίδες, μετά από την πρώιμη προσομοιώση το παιχνίδι είναι αρκετά κοντά σε μια τερματική κατάσταση και η πρόβλεψη του μοντέλου μπορεί να είναι ακόμα πιο ακριβής, σε αντίθεση με τα παιχνίδια με μεγαλύτερη διάρκεια.

Για τον προσδιορισμό του κατωφλίου που χρησιμοποιείται στην πρώιμη προσομοίωση, εξετάστηκαν διαφορετικές τιμές όπως φαίνεται στο Σχήμα 4.1. Για κάθε τιμή κατωφλίου t, εκπαιδεύτηκε ένα γενικό μοντέλο (χρησιμοποιώντας δεδομένα από όλα τα προσομοιωμένα παιχνίδια) στο αντίστοιχο υποσύνολο δεδομένων που περιλαμβάνει δείγματα από γύρους μεταγενέστερους του κατωφλίου. Για τον τελικό πράκτορα, επιλέχθηκε η πρώιμη προσομοίωση να γίνεται μέχρι τον πέμπτο γύρο της παρτίδας καθώς με αυτή τη ρύθμιση πετυχαίνει τη βέλτιστη απόδοση στους δύο από τους τρεις εξεταζόμενους τύπους τράπουλας και πλησιάζει τη βέλτιστη στην τρίτη περίπτωση.

Στον τέταρτο πράκτορα (MCTS-xgboostComb) εξετάζεται η χρήση της συνδυαστικής βαθμολόγησης των καταστάσεων από το εκπαιδευμένο μοντέλο και το αποτέλεσμα μίας προσομοίωσης τυχαίων κινήσεων. Η συγκεκριμένη μέθοδος οδηγεί σε σημαντική αύξηση της απόδοσης του πράκτορα σε όλα τα πειράματα, ανεξαρτήτως του μοντέλου πρόβλεψης και του τύπου τράπουλας. Συνεπώς, επιβεβαιώνεται ότι παρότι η χρήση του ταξινομητή για την αξιολόγηση των καταστάσεων αποτελεί σημαντική προσθήκη, επιδέχεται επιπλέον βελτίωση καθώς η πρόβλεψη του μοντέλου είναι δυαδική (νίκη/ήττα) και ως εκ τούτου απαιτούνται αρκετές επαναλήψεις προκειμένου να διαχωριστούν παρόμοιες καταστάσεις που μπορεί να οδηγούν στο ίδιο αποτέλεσμα αλλά με διαφορετικό βαθμό βεβαιότητας. Η ενσωμάτωση προσομοιώσεων με πολιτική τυχαίων ενεργειών στην τελική αξιολόγηση συμβάλλει προς αυτή την κατεύθυνση, βοηθώντας τη διαβάθμιση των διαφορετικών ενεργειών/καταστάσεων όπως αποτυπώνεται στο ποσοστό νίκης του πράκτορα στους Πίνακες 4.4 και 4.5.

Όσον αφορά στη στοχαστική αξιολόγηση των κόμβων, η επίδραση της μεθόδου στο πλαίσιο των πρακτόρων με 500 διαθέσιμες επαναλήψεις δεν είναι ξεκάθαρη. Από τους Πίνακες 4.4 και 4.5 φαίνεται ότι έχει σε κάποιες περιτπώσεις θετική επίδραση ενώ σε κάποιες αρνητική, χωρίς να προκύπτει κάποια

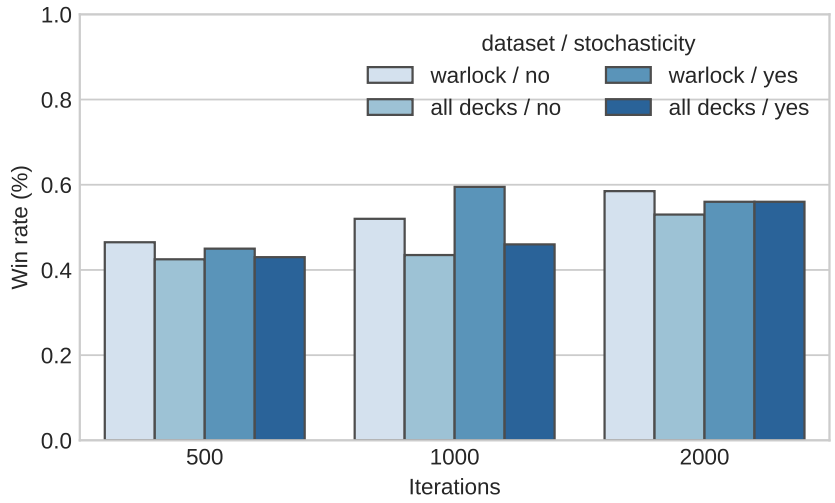


Σχήμα 4.1: Ποσοστό νίκης πράκτορα για διαφορετικές τιμές κατωφλίου t

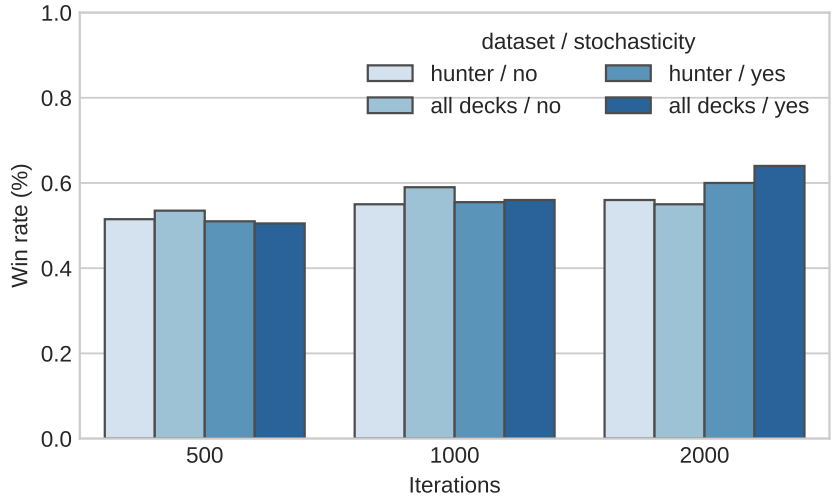
ακριβής συσχέτιση με τον τύπο του μοντέλου ή τη στρατηγική παιχνιδιού στην οποία αξιολογείται. Σε κάθε περίπτωση, η επίδραση στην απόδοση των πρακτόρων είναι μικρότερη από την αντίστοιχη της ενσωμάτωσης του μοντέλου αξιολόγησης και της συνδυαστικής αξιολόγησης καταστάσεων που παρατηρήθηκε στον δεύτερο και στον τέταρτο πράκτορα αντίστοιχα.

Σε όλες τις παραλλαγές του πράκτορα που παρουσιάζονται παραπάνω εκτελούνται 500 επαναλήψεις προτού επιλεχθεί κάθε ενέργεια. Παρότι ο αριθμός επαναλήψεων είναι μικρός δεδομένου του *παράγοντα διακλάδωσης* (branching factor), ο τελικός πράκτορας είναι πολύ ανώτερος από τον απλό MCTS και έχει παρόμοια απόδοση με τον GSV σε δύο από τα τρία είδη τράπουλας. Προκειμένου να εξεταστεί βαθύτερα η τεχνική της στοχαστικής αξιολόγησης κόμβων, στη συνέχεια παρουσιάζεται η απόδοση του τελικού πράκτορα εναντίον του GSV για διαφορετικό αριθμό διαθέσιμων επαναλήψεων. Καθώς μεγαλύτερος αριθμός επαναλήψεων συνεπάγεται μεγαλύτερο υπολογιστικό κόστος, ο μέγιστος αριθμός που εξετάζεται είναι 2.000 ώστε να υπάρχει ισορροπία μεταξύ της ποιότητας της απόφασης του αλγορίθμου και του χρονικού διαστήματος που απαιτείται για την εκτέλεσή του.

Στα Σχήματα 4.2, 4.3 και 4.4 φαίνεται το ποσοστό νίκης των πρακτόρων συναρτήσει του αριθμού διαθέσιμων επαναλήψεων του αλγορίθμου για τους διαφορετικούς συνδυασμούς μοντέλου πρόβλεψης (γενικό/εξειδικευμένο) και στοχαστικής αξιολόγησης (με/χωρίς). Αρχικά, σε όλες τις περιπτώσεις παρατηρείται σταδιακή αύξηση της απόδοσης καθώς αυξάνονται οι διαθέσιμες επαναλήψεις όπως αναμενόταν. Ο ρυθμός αύξησης ώστοσο διαφέρει ανάλογα με τον τύπο παιχνιδιού (στρατηγικής), το οποίο οφείλεται πιθανότατα στον διαφορετικό παράγοντα διακλάδωσης σε κάθε περίπτωση. Επιπλέον, η επιλογή του μοντέλου φαίνεται ότι επίσης εξαρτάται σε μεγάλο βαθμό από τον επιλεγμένο τύπο τράπουλας. Στην πρώτη περίπτωση (warlock) φαίνεται να υπερτερεί το εξειδικευμένο μοντέλο ενώ στις υπόλοιπες η υψηλότερη επίδοση επιτυγχάνεται με το γενικό μοντέλο. Μία εξήγηση για αυτό το φαινόμενο είναι ότι στις δύο τελυταίες περιπτώσεις οι παρτίδες τείνουν να ολοκληρώνονται γρηγορότερα (λόγω της στρατηγικής που ακολουθείται) και το γενικό μοντέλο επωφελείται από τα συνολικά δεδομένα που χρησιμοποιούνται κατά την εκπαίδευση. Αντιθέτως, στην πρώτη περίπτωση όπου το δέντρο αναζήτησης φτάνει σε μεγαλύτερο βάθος, το εξειδικευμένο μοντέλο που εκπαιδεύεται στα αντίστοιχα δεδομένα έχει τη δυνατότητα ακριβέστερης πρόβλεψης στις καταστάσεις που βρίσκονται πιο χαμηλά στο δέντρο σε σχέση με το γενικό μοντέλο.

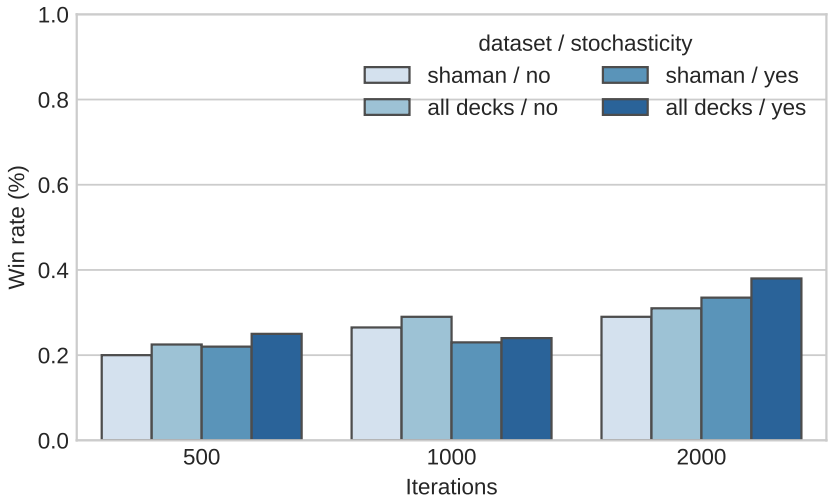


Σχήμα 4.2: Ποσοστό νίκης τελικού πράκτορα εναντίον GSV (warlock)



Σχήμα 4.3: Ποσοστό νίκης τελικού πράκτορα εναντίον GSV (hunter)

Όσον αφορά στη στοχαστική αξιολόγηση, παρότι δεν έχει ιδιαίτερα μεγάλη επίδραση στην περίπτωση των 500 επαναλήψεων, ενισχύει σημαντικά τον αλγόριθμο όταν το διαθέσιμο πλήθος επαναλήψεων αυξάνεται ενώ η ντετερμινιστική εκδοχή του πράκτορα σταθεροποιείται πιο γρήγορα (1000 επαναλήψεις) και παρουσιάζει μικρότερη βελτίωση στη συνέχεια. Συνολικά, η αύξηση του αριθμού των επαναλήψεων βελτιώνει ιδιαίτερα την απόδοση της προτεινόμενης προσέγγισης (έως 16% στην καλύτερη περίπτωση) και ξεπερνάει τον αλγόριθμο GSV στα δύο από τα τρία εξεταζόμενα σύνολα πειραμάτων. Συγκεκριμένα, ο τελικός πράκτορας με την ενσωμάτωση του γενικού ταξινομητή επιτυγχάνει ποσοστά νίκης 56%, 64% και 38% στα αρχέτυπα warlock, hunter και shaman αντίστοιχα και συνολικό ποσοστό νίκης 52,6% έναντι του αλγορίθμου GSV.



Σχήμα 4.4: Ποσοστό νίκης τελικού πράκτορα εναντίον GSV (shaman)

Συνολικά, από την αξιολόγηση των προτεινόμενων πρακτόρων προκύπτει ότι η εισαγωγή γνώσης πεδίου στη φάση προσομοίωσης του αλγορίθμου δενδρικής αναζήτησης Μόντε Κάρλο ενισχύει σημαντικά την αποτελεσματικότητά του. Συγκεκριμένα, η ενσωμάτωση ενός ταξινομητή για την πρόβλεψη του τελικού αποτελέσματος και ο συνδυασμός της πρόβλεψης με το αποτέλεσμα μίας τυχαίας προσομοίωσης έχουν τη μεγαλύτερη επίδραση στη βελτίωση του πράκτορα. Η χρήση του μοντέλου πρόβλεψης μειώνει τη διακύμανση που εισάγεται από τις τυχαίες προσομοιώσεις και οδηγεί σε ακριβέστερη αξιολόγηση των κόμβων του δέντρου αναζήτησης. Η πρώιμη προσομοίωση και η στοχαστική αξιολόγηση των κόμβων συμβάλλουν επίσης στην αύξηση της απόδοσης του αλγορίθμου, σε μικρότερο όμως βαθμό. Η επίδραση της τελευταίας γίνεται πιο ξεκάθαρη όταν αυξάνεται ο αριθμός των διαθέσιμων επαναλήψεων του αλγορίθμου. Σε αυτή την περίπτωση, λαμβάνοντας υπόψη το μέγεθος του χώρου ενεργειών προσαρμόζεται κατάλληλα η μέθοδος αξιολόγησης των κόμβων με αποτέλεσμα την πιο αξιόπιστη εκτίμηση της αξίας τους. Όσον αφορά στα δεδομένα εκπαίδευσης του ταξινομητή, παρότι ο αλγόριθμος αξιολογήθηκε σε τρία διαφορετικά περιβάλλοντα, ο γενικός ταξινομητής που εκπαιδεύτηκε στο σύνολο των δεδομένων οδήγησε σε πράκτορες συγκρίσιμους με αυτούς που ενσωματώνουν εξειδικευμένους ταξινομητές ανά περίπτωση. Καθώς δεν προέκυψε κάποιο εμφανές πλεονέκτημα ενός ταξινομητή (του γενικού ή του ειδικού) σε όλες τις περιπτώσεις, μπορεί με σχετική ασφάλεια να χρησιμοποιηθεί ο γενικός ταξινομητής καθιστώντας πιο ευέλικτο τον αλγόριθμο. Ωστόσο, πρέπει να σημειωθεί ότι η απόδοση του μοντέλου εξαρτάται από τα χαρακτηριστικά του παιχνιδιού από το οποίο συλλέγονται τα δεδομένα εκπαίδευσης και ενδεχομένως σε διαφορετικά περιβάλλοντα ο εξειδικευμένος ταξινομητής να οδηγεί σε μεγαλύτερη βελτίωση του πράκτορα.

Κεφάλαιο 5

Ενίσχυση Φάσης Επιλογής στη Δενδρική Αναζήτηση Μόντε Κάρλο

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, ο αλγόριθμος δενδρικής αναζήτησης Μόντε Κάρλο είναι ιδιαίτερα ευαίσθητος στο πλήθος των διαθέσιμων επαναλήψεων ανά ενέργεια. Αυτό δεν επηρεάζει τη λειτουργικότητα του, δεδομένου ότι μπορεί να σταματήσει οποιαδήποτε στιγμή και να επιστρέψει τη βέλτιστη ενέργεια (όπως έχει οριστεί ως εκείνη τη στιγμή), ωστόσο έχει μεγάλη επίδραση στην αποτελεσματικότητά του. Στο Κεφάλαιο 4 αναπτύχθηκε μία μεθοδολογία βελτίωσης της φάσης προσομοίωσης του αλγορίθμου ώστε να αξιοποιηθούν οι διαθέσιμες επαναλήψεις με αποδοτικότερο τρόπο. Σε αυτό το σημείο παρουσιάζονται δύο νέες μεθοδολογίες που αφορούν στο στάδιο επιλογής του MCTS. Στην πρώτη περίπτωση ο προτεινόμενος αλγόριθμος εστιάζει στο *κλάδεμα* (pruning) του δέντρου αναζήτησης κάνοντας χρήση νευρωνικών δικτύων, προκειμένου να απορρίψει τις ενέργειες με χαμηλή αναμενόμενη ανταμοιβή και να καθοδηγήσει την αναζήτηση στις πιο υποσχόμενες ενέργειες. Με αυτό τον τρόπο, το πλήθος των ενεργειών προς εξέταση στις εναπομείνουσες επαναλήψεις είναι μικρότερο και ως εκ τούτου το δέντρο αναζήτησης μπορεί να φτάσει σε μαγαλύτερο βάθος και να προκύψει ακριβέστερη αξιολόγηση των κόμβων. Η δεύτερη τεχνική που εξετάζεται έχει στόχο την αποδοτικότερη αξιοποίηση των στατιστικών των κόμβων του δέντρου αναζήτησης και στηρίζεται σε μεθόδους εύρεσης ομοιότητας κόμβων διαφορετικών επιπέδων. Εφόσον βρεθεί ένας κόμβος ψηλότερα στο δέντρο με υψηλή ομοιότητα με τον τρέχοντα κόμβο, μπορεί να αξιοποιηθεί η επιπλέον πληροφορία για την παραγωγή ακριβέστερων εκτιμήσεων των μετρικών που χρησιμοποιούνται στο στάδιο επιλογής, αυξάνοντας τη συνολική απόδοση του αλγορίθμου.

5.1 Τεχνικές Κλαδέματος

Η ιδέα του κλαδέματος στον αλγόριθμο δενδρικής αναζήτησης Μόντε Κάρλο έχει εφαρμοστεί με διαφορετικές προσεγγίσεις, τόσο με χρήση τεχνικών που είναι *εξαρτώμενες απο το εκάστοτε πεδίο εφαρμογής* (domain dependent) όσο και με *ανεξάρτητες* (domain independent) τεχνικές. Μία διαδεδομένη μέθοδος, η οποία βασίζεται αμιγώς στα στατιστικά δεδομένα που συλλέγονται κατά την εκτέλεση του αλγορίθμου, ελέγχει κατά πόσο είναι δυνατόν μία ενέργεια να επιλεγεί τελικά ως βέλ-

τιστη δεδομένης της τρέχουσας αξιολόγησής της και του αριθμού επαναλήψεων που απομένουν [46]. Δύο βασικές παραλλαγές της μεθόδου έχουν παρουσιαστεί· στην πρώτη, η οποία καλείται *απόλυτο κλάδεμα* (absolute pruning), ένας κόμβος που προκύπτει από μία ενέργεια (και όλο το υποδέντρο που βρίσκεται κάτω από τον κόμβο) απορρίπτεται από το δέντρο αναζήτησης όταν υπολογιστεί ότι είναι αδύνατο να αξιολογηθεί ως βέλτιστος στις επαναλήψεις που απομένουν ενώ στη δεύτερη, η οποία ονομάζεται *σχετικό κλάδεμα* (relative pruning), υπολογίζεται ένα άνω όριο της αναμενόμενης βαθμολογίας της μετρικής που εξετάζεται για την τελική επιλογή του βέλτιστου κόμβου (π.χ. αριθμός φορών που προσπελάστηκε) και ο κόμβος απορρίπτεται όταν το άνω όριο είναι χαμηλότερο από την υψηλότερη τρέχουσα βαθμολογία. Και οι δύο αυτές προσεγγίσεις ανήκουν στην ευρύτερη κατηγορία του *μόνιμου κλαδέματος* (hard pruning) καθώς οι κόμβοι που απορρίπτονται δεν ξαναλαμβάνονται υπόψη μέχρι το τέλος του αλγορίθμου.

Μία πιο ευέλικτη προσέγγιση, η οποία επιτρέπει την επανεξέταση των κλαδευμένων κινήσεων μετά από συγκεκριμένο χρονικό διάστημα ή αριθμό επαναλήψεων, έχει επίσης διερευνηθεί. Το *προσωρινό κλάδεμα* (soft pruning) διασφαλίζει ότι οι ενέργειες δεν εξαλείφονται οριστικά, προκειμένου να μειωθεί ο κίνδυνος να αποκλειστεί τελείως η βέλτιστη ενέργεια από την αναζήτηση. Προς αυτή την κατεύθυνση, η *προοδευτική αναίρεση κλαδέματος* (progressive unpruning) [16] και η *προοδευτική διεύρυνση* (progressive widening) [22] κάνουν χρήση ευριστικών συναρτήσεων για την αξιολόγηση των κόμβων και κλαδεύουν τις ενέργειες με τη χαμηλότερη αναμενόμενη ανταμοιβή έπειτα από ένα χρονικό κατώφλι. Σταδιακά, οι ενέργειες που εξαιρέθηκαν προστίθενται και πάλι στο δέντρο αναζήτησης και γίνονται διαθέσιμες καθώς αυξάνονται οι επαναλήψεις. Αυτή η προσέγγιση αναπτύσσεται περαιτέρω και προσαρμόζεται ώστε να μπορεί να εφαρμοστεί σε συνεχή στοχαστικά περιβάλλοντα στη μελέτη [20].

Μέθοδοι που ενσωματώνουν γνώση πεδίου έχουν υλοποιηθεί και για μόνιμο κλάδεμα ενεργειών. Στην περίπτωση [92], ευριστικές συναρτήσεις βασισμένες στα χαρακτηριστικά ενός συγκεκριμένου περιβάλλοντος/παιχνιδιού χρησιμοποιήθηκαν για να μειωθεί ο χώρος ενεργειών ενισχύοντας τον αλγόριθμο δενδρικής αναζήτησης Μόντε Κάρλο. Επιπλέον, γνώση πεδίου έχει χρησιμοποιηθεί και για μόνιμο κλάδεμα τετριμμένων ενεργειών, οι οποίες (ανάλογα με το πλήθος τους) μπορούν σε ορισμένες περιπτώσεις να καθυστερήσουν σημαντικά τη διαδικασία αναζήτησης [74].

5.2 Πρόβλημα Ληστή Πολλαπλών Χεριών και Ανώτατο Όριο Εμπιστοσύνης

Το πρόβλημα *Ληστή Πολλαπλών Χεριών* (Multi-Armed Bandit - MAB) [51] είναι ένα πρόβλημα απόφασης που βασίζεται στο *δίλημμα εξερεύνησης-εκμετάλλευσης* (exploration-exploitation dilemma). Ειδικότερα, περιγράφει την κατάσταση κατά την οποία ένας τζογαδόρος επιδιώκει να μεγιστοποιήσει το κέρδος του επιλέγοντας επαναληπτικά μεταξύ διαφορετικών *μηχανών κουλοχέρη* (one-armed bandits) με άγνωστες κατανομές ανταμοιβής. Σε αυτό το πλαίσιο, ο παίκτης πρέπει να λαμβάνει αποφάσεις με τέτοιο τρόπο ώστε αφενός να εκμεταλλεύεται τις πληροφορίες που έχει αποκτήσει από προηγούμενες ανταμοιβές, αφετέρου να διερευνά τις πιο σπάνια επιλεγμένες ενέργειες προκειμένου να επαληθεύσει ποια είναι η μηχανή με την υψηλότερη ανταμοιβή.

Τυπικά, μπορεί να οριστεί ως ένα πρόβλημα K τυχαίων μεταβλητών $R = \{R_1, R_2, \ldots, R_k\}$ με πραγματικές κατανομές $D = \{D_1, D_2, \ldots, D_k\}$, στο οποίο κάθε μεταβλητή αντιπροσωπεύει την ανταμοιβή μίας ενέργειας $x_i \in X = \{x_1, x_2, \ldots, x_k\}$ και κάθε κατανομή την αντίστοιχη κατανομή

ανταμοιβής. Δεδομένου ενός πεπερασμένου πλήθους γύρων T και μίας πολιτικής επιλογής $\pi(t)$, έστω x_i η ενέργεια που επιλέχθηκε στο χρονικό βήμα t και $r_i \sim D_i$ η ανταμοιβή που προέκυψε. Ο στόχος του παίκτη είναι να ελαχιστοποιήσει την τιμή ρ (Εξίσωση 5.1):

$$\rho = T \max_{x_i \in X} \mathbf{E}[R_i|x_i] - \sum_{t=1}^{T} \sum_{i=1}^{K} r_i [\![x_i = \pi(t)]\!] \tag{5.1}$$

δηλαδή τη διαφορά της συνολικής ανταμοιβής που προκύπτει ακολουθώντας την πολιτική π από τη συνολική ανταμοιβή που προκύπτει επιλέγοντας πάντα τη βέλτιστη ενέργεια (δηλαδή την ενέργεια με τη μέγιστη αναμενόμενη ανταμοιβή). Υπό αυτό το πρίσμα, το πρόβλημα ανάγεται σε μία μαρκοβιανή αλυσίδα με συνάρτηση μετάβασης καταστάσεων $P(s, s'|a) = 0 \ \forall \ s' \in \{S - s\}$, όπου S είναι το σύνολο των δυνατών καταστάσεων, δεδομένου ότι η εκτέλεση μίας ενέργειας δεν οδηγεί σε αλλαγή της κατάστασης. Συνεπώς, η αναμενόμενη ανταμοιβή μπορεί να θεωρηθεί ως η αξία ενέργειας $Q(x_i)$, με $Q_*(x_i) = \max_{x_i \in X} Q(x_i)$ να είναι η αξία της βέλτιστης ενέργειας.

Για την επίλυση του MAB έχουν προταθεί διάφορες στρατηγικές. Ο αλγόριθμος *Ανώτατου Ορίου Εμπιστοσύνης* (Upper Confidence Bound - UCB) εστιάζει στη βελτιστοποίηση της στρατηγικής επιλογής εξισορροπώντας την εξερεύνηση και την εκμετάλλευση της τρέχουσας πληροφορίας [5]. Βάση του αλγορίθμου είναι η οπτιμιστική υπόθεση ότι η πραγματική αξία μίας ενέργειας είναι υψηλότερη από την τρέχουσα εκτίμηση της. Συγκεκριμένα, υπολογίζεται ένα ανώτερο όριο για την αξία κάθε ενέργειας αναλογα με την τρέχουσα προσέγγισή της και τον αντίστοιχο βαθμό αβεβαιότητας.

Το άνω όριο της απόκλισης μεταξύ του παρατηρούμενου μέσου όρου ενός συνόλου ανεξάρτητων τυχαίων μεταβλητών (όπως προκύπτει από τα υπάρχοντα δείγματα) και του πραγματικού τους μέσου όρου μπορεί να υπολογιστεί με την *ανισότητα Hoeffding* [43]. Σύμφωνα με την ανισότητα Hoeffding, έστω $X_1, X_2, \ldots, X_n$ *ανεξάρτητες πανομοιότυπα κατανεμημένες* (independent identically distributed - i.i.d.) τυχαίες μεταβλητές στο διάστημα $[0, 1]$ και $\overline{X} = \frac{1}{n}(X_1 + X_2 + \cdots + X_n)$, τότε η πιθανότητα η διαφορά της μέσης τιμής τους από την αναμενόμενη τιμή της να είναι μεγαλύτερη από ένα όριο k οριοθετείται σύμφωνα με την Εξίσωση 5.2:

$$P(\mathbf{E}[X] - \overline{X} \geq k) \leq e^{-2nk^2} \Rightarrow P(\mathbf{E}[X] \geq \overline{X} + k) \leq e^{-2nk^2} \tag{5.2}$$

Στην περίπτωση του MAB, αντικαθιστώντας τις τυχαίες μεταβλητές X_i με τις παρατηρούμενες ανταμοιβές R_i^n της ενέργειας i έπειτα από n βήματα μπορούμε να υπολογίσουμε την πιθανότητα η πραγματική της αξία να είναι μεγαλύτερη από το άνω όριο (Εξίσωση 5.3).

$$P(\mathbf{E}[R_i] \geq \overline{R_i} + k) \leq e^{-2nk^2} \tag{5.3}$$

Καθώς η επιλογή στον αλγόριθμο UCB γίνεται οπτιμιστικά, το άνω όριο της αξίας κάθε ενέργειας πρέπει να είναι όσο το δυνατόν πιο αυστηρό, δηλαδή θα πρέπει να είναι μεγαλύτερο ή ίσο με την αναμενόμενη τιμή της με μεγάλη πιθανότητα. Ως εκ τούτου, η πιθανότητα στην Εξίσωση 5.3 πρέπει να είναι πολύ μικρή. Θέτοντας αυτή την πιθανότητα ίση με μία πολύ μικρή θετική τιμή a, το κατώφλι k μπορεί να προσδιοριστεί σύμφωνα με την Εξίσωση 5.4. Καθώς ο αριθμός των δειγμάτων αυξάνεται, η διακύμανση της παρατηρούμενης μέσης τιμής $\overline{R_i}$ μειώνεται. Συνεπώς, το άνω όριο θα μπορούσε να μειωθεί αναλογικά με τον συνολικό αριθμό των τρεχουσών επαναλήψεων. Στον αλγόριθμο $UCB1$, την πιο συχνά χρησιμοποιούμενη παραλλαγή του αλγορίθμου, η τιμή του a ορίζεται ως N^{-4}.

$$e^{-2nk^2} = a \Rightarrow k = \sqrt{\frac{-\ln a}{2n}} \xrightarrow{a=N^{-4}} k_{\text{UCB1}} = \sqrt{\frac{2\ln N}{n}} \tag{5.4}$$

Με βάση τα παραπάνω, σε κάθε βήμα ο αλγόριθμος επιλέγει την ενέργεια με το υψηλότερο άνω όριο εμπιστοσύνης (Εξίσωση 5.5).

$$\text{UCB}(x_i) = Q(x_i) + C\sqrt{\frac{\log N}{n_i}} \tag{5.5}$$

όπου $Q(x_i)$ είναι η τρέχουσα προσέγγιση της αξίας της ενέργειας x_i, N είναι το συνολικό πλήθος επιλογών που έχουν γίνει μέχρι το βήμα t, n_i είναι το πλήθος των φορών που επιλέχθηκε η ενέργεια x_i και C είναι μία παράμετρος εξερεύνησης.

Ο πρώτος όρος της Εξίσωσης 5.5 αφορά στην εκμετάλλευση της τρέχουσας πληροφορίας (λαμβάνοντας υπόψη την εκτιμώμενη αξία κάθε ενέργειας) ενώ ο δεύτερος όρος αφορά στην εξερεύνηση και εκφράζει την αβεβαιότητα της τρέχουσας εκτίμησης. Όσες περισσότερες φορές έχει αξιολογηθεί μία ενέργεια, τόσο μικρότερη θα πρέπει να είναι η αύξηση του άνω ορίου της αξίας της καθώς η εμπιστοσύνη σε αυτή την τιμή αυξάνεται και το αντίστροφο. Για αυτόν τον λόγο, ο δεύτερος όρος μειώνεται όσο περισσότερο επιλέγεται η συγκεκριμένη ενέργεια και το άνω όριο της αξίας προσεγγίζει την εκτίμηση της αξίας της καθώς αυξάνεται ο αριθμός των επαναλήψεων. Όσον αφορά στην παράμετρο C, καθορίζει τη συνεισφορά του όρου εξερεύνησης στη συνολική τιμή. Στην περίπτωση του $UCB1$ τίθεται ίση με $\sqrt{2}$.

5.3 Κλάδεμα Δέντρου Αναζήτησης με Νευρωνικά Δίκτυα

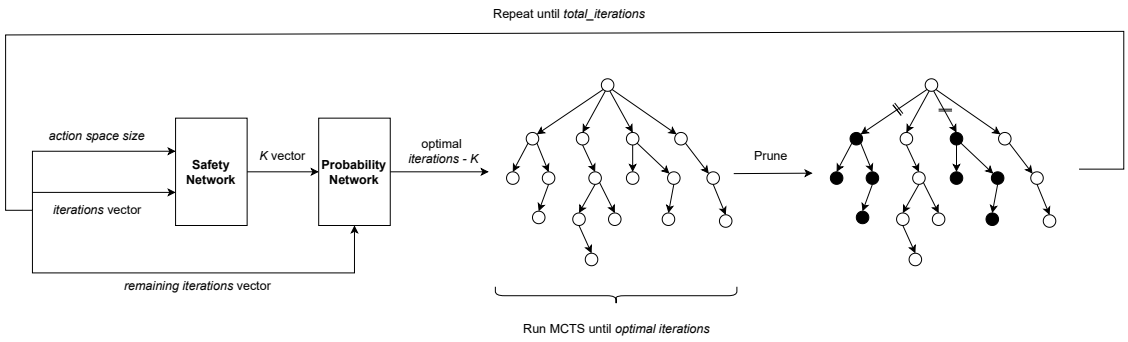
Για να αντιμετωπιστεί το πρόβλημα του περιορισμένου αριθμού επαναλήψεων του αλγορίθμου δενδρικής αναζήτησης Μόντε Κάρλο (ιδιαίτερα στις περιπτώσεις με μεγάλο παράγοντα διακλάδωσης) προτείνεται μία τεχνική κλαδέματος με χρήση τεχνητών νευρωνικών δικτύων. Η συγκεκριμένη μεθοδολογία περιλαμβάνει δύο διαφορετικά δίκτυα με στόχο αρχικά τον προσδιορισμό των ενεργειών που μπορούν να αποκοπούν από το δέντρο αναζήτησης με ασφάλεια (δηλαδή χωρίς να υπάρχει κίνδυνος να απορριφθεί η βέλτιστη ενέργεια) και εν συνεχεία την επιλογή του χρονικού σημείου (σε αριθμό επαναλήψεων από το ξεκίνημα του αλγορίθμου) στο οποίο θα γίνει το κλάδεμα. Για την εκπαίδευση των δικτύων δημιουργήθηκε ένα συνθετικό σύνολο δεδομένων μέσω ενός ειδικά σχεδιασμένου περιβάλλοντος που προσομοιώνει το πρόβλημα ληστή πολλαπλών χεριών. Οι κατανομές των αναμενόμενων αξιών των ενεργειών εξετάστηκαν επίσης και λήφθηκαν υπόψη για τη δημιουργία του συνόλου δεδομένων, με αποτέλεσμα να προκύψουν δύο διαφορετικές παραλλαγές του αλγορίθμου. Η προτεινόμενη μέθοδος υλοποιήθηκε και αξιολογήθηκε στο περιβάλλον που περιγράφεται στο Κεφάλαιο 4.

5.3.1 Μεθοδολογία

Για το κλάδεμα του δέντρου αναζήτησης χρησιμοποιούνται δύο ξεχωριστά νευρωνικά δίκτυα όπως αναφέρθηκε παραπάνω. Στόχος του πρώτου δικτύου (*Δίκτυο Ασφάλειας* (Safety Network)) είναι ο προσδιορισμός του μέγιστου αριθμού ενεργειών που μπορούν να αφαιρεθούν ασφαλώς (δηλαδή του μέγιστου υποσυνόλου ενεργειών με χαμηλή εκτιμώμενη αξία, το οποίο δεν περιλαμβάνει την ενέργεια με τη μέγιστη πραγματική αξία) δεδομένου του αριθμού επαναλήψεων που έχουν εκτελεστεί και του μεγέθους του χώρου ενεργειών. Λόγω της στοχαστικής φύσης του προβλήματος είναι πιθανό η

βέλτιστη ενέργεια αρχικά να αξιολογηθεί (λανθασμένα) ως ενέργεια με χαμηλή αναμενόμενη αξία και η εκτίμηση να αρχίσει να προσεγγίζει την πραγματική της αξία σταδιακά. Το δίκτυο ασφάλειας χρησιμοποιείται ώστε να αποτραπεί η απόρριψη της βέλτιστης ενέργειας σε μία τέτοια περίπτωση. Όπως γίνεται αντιληπτό, το σύνολο των υπολειπόμενων ενεργειών (που έχουν υψηλή εκτιμώμενη αξία και παραμένουν στο δέντρο αναζήτησης) μπορεί να είναι μικρότερο καθώς ο αριθμός επαναλήψεων αυξάνεται. Αυτό συμβαίνει γιατί οι ενέργειες αξιολογούνται ακριβέστερα όσο αυξάνεται το πλήθος των προσομοιώσεων και άρα μπορούν να απορριφθούν περισσότερες ενέργειες με ασφάλεια. Κατά συνέπεια, υπάρχει ένα δίλημμα μεταξύ του υποσυνόλου των ενεργειών που πρέπει να κλαδευτούν και του υπολοιπόμενου αριθμού επαναλήψεων, καθώς όσο περισσότερες επαναλήψεις χρησιμοποιούνται μέχρι το κλάδεμα (εξασφαλίζοντας την απόρριψη περισσότερων ενεργειών), τόσο λιγότερες θα είναι διαθέσιμες στη συνέχεια για την αναζήτηση μεταξύ των εναπομεινουσών ενεργειών.

Για την εύρεση του βέλτιστου συνδυασμού επαναλήψεων και πλήθους ενεργειών για κλάδεμα χρησιμοποιείται το δεύτερο δίκτυο (*Δίκτυο Πιθανότητας* (Probability Network)). Αυτό το δίκτυο εκπαιδεύεται ώστε να προβλέπει την πιθανότητα εύρεσης της βέλτιστης ενεέργειας από τον αλγόριθμο δεδομένου του χώρου ενεργειών και των επαναλήψεων που απομένουν. Συνεπώς, η διαδικασία που ακολουθείται για το κλάδεμα αποτελείται από δύο στάδια. Στο πρώτο βήμα, το μεγαλύτερο υποσύνολο ενεργειών που μπορεί να κλαδευτεί από το δέντρο αναζήτησης μετά από κάθε επανάληψη υπολογίζεται με τη χρήση του δικτύου ασφαλείας. Στο δεύτερο στάδιο προσδιορίζεται ο βέλτιστος συνδυασμός επαναλήψεων που πρέπει να εκτελεστούν πριν το κλάδεμα και πλήθους ενεργειών που πρέπει να αποκοπούν ως ο συνδυασμός με την υψηλότερη πιθανότητα επιλογής της βέλτιστης ενέργειας όπως προκύπτει από το δίκτυο πιθανότητας. Όταν εκτελεστούν οι επαναλήψεις που προσδιορίστηκαν, το αντίστοιχο υποσύνολο ενεργειών κλαδεύεται από το δέντρο και η παραπάνω διαδικασία επαναλαμβάνεται μέχρι την ολοκλήρωση όλων των επαναλήψεων. Η συνολική λειτουργία του αλγορίθμου παρουσιάζεται διαγραμματικά στο Σχήμα 5.1.



Σχήμα 5.1: Διαδικασία κλαδέματος με νευρωνικά δίκτυα για τον αλγόριθμο MCTS (οι κόμβοι που κλαδεύονται παρουσιάζονται με μαύρο χρώμα)

Η παραπάνω διαδικασία κλαδέματος περιγράφεται αναλυτικά στον Αλγόριθμο 5. Αρχικά, ο ελάχιστος αριθμός των ενεργειών (K_i) που μπορούν να παραμείνουν στο δέντρο αναζήτησης υπολογίζεται από το δίκτυο ασφάλειας για κάθε τιμή i των διαθέσιμων επαναλήψεων, δεδομένου του μεγέθους του τρέχοντος χώρου ενεργειών. Καθώς στην αρχή της αναζήτησης δεν υπάρχει καμία πληροφορία για τις αξίες των ενεργειών (δεν έχουν αξιολογηθεί ακόμα), το δίκτυο ασφάλειας χρησιμοποείται για πρώτη φορά όταν ο αριθμός των εκτελεσθέντων επαναλήψεων ξεπεράσει ένα κάτω όριο. Δεδομένου ότι το μέγεθος του χώρου ενεργειών στο συγκεκριμένο περιβάλλον δεν ξεπερνάει το 50 στη μεγάλη πλειοψηφία των κινήσεων, το όριο των επαναλήψεων έχει τεθεί ίσο με 100, με στόχο

η αναμενόμενη αξία όλων των κινήσεων να έχει εκτιμηθεί ως ένα βαθμό τη στιγμή της πρόβλεψης ($i \in [current\_iterations + 100, total\_iterations]$). Στη συνέχεια, το δίκτυο πιθανότητας χρησιμοποιείται για να προβλέψει για κάθε αριθμό επαναλήψεων την πιθανότητα να βρεθεί η βέλτιστη ενέργεια μέσω της δενδρικής αναζήτησης, αν κλαδευτεί σε αυτό το σημείο το υποσύνολο των ενεργειών που υπολογίστηκε από το δίκτυο ασφάλειας. Με βάση την πρόβλεψη του δικτύου προσδιορίζεται ο βέλτιστος συνδυασμός (K_{opt}, $Iter_{opt}$) και η αναζήτηση εκτελείται κανονικά μέχρι να ολοκληρωθούν $Iter_{opt}$ επαναλήψεις ή να τερματιστεί ο αλγόριθμος. Όταν εκτελεστούν $Iter_{opt}$ επαναλήψεις, κλαδεύονται οι ενέργειες με τη χαμηλότερη εκτιμώμενη αξία και η αναζήτηση συνεχίζεται στις K_{opt} ενέργειες. Ο αλγόριθμος συνεχίζεται επαναληπτικά έως ότου ικανοποιηθεί το κριτήριο τερματισμού.

Algorithm 5: MCTS with Pruning Networks.

```
1  Function MCTS():
2      while total_iterations > 0 do
3          probs ← []
4          for i in iterations do
5              K_i ← Safety_Network.predict(action_space, i)
6          for i in iterations do
7              remaining_iterations ← total_iterations − i
8              prob_i ← Probability_Network.predict(K_i, remaining_iterations)
9              probs.append(prob_i)
10         K_opt ← argmax_K probs
11         Iter_opt ← argmax_iter probs
12         for i in min(Iter_opt, total_iterations) do
13             Select()
14             Expand()
15             Rollout()
16             BackPropagate()
17         {actions_to_prune} ← {total_actions} − {K_opt}
18         {action_space}.remove({actions_to_prune})
19         total_iterations ← total_iterations − Iter_opt
20     return best_action
```

Σχετικά με την αρχιτεκτονική των δικτύων ασφάλειας και πιθανότητας, είναι δίκτυα πρόσθιας τροφοδότησης με τρία κρυφά επίπεδα το καθένα. Ο αριθμός νευρώνων σε κάθε επίπεδο είναι 200, 300 και 100 για το δίκτυο ασφάλειας και 300, 500, 200 για το δίκτυο πιθανότητας αντίστοιχα. Η συνάρτηση ενεργοποίησης σε κάθε επίπεδο είναι η διορθωμένη γραμμική (βλ. Ενότητα 2.2.3). Και για τα δύο δίκτυα χρησιμοποιήθηκε ο βελτιστοποιητής *Adam* με στόχο την ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος (η πρόβλεψη του πλήθους ενεργειών για κλάδεμα αντιμετωπίζεται από το δίκτυο ασφάλειας επίσης ως πρόβλημα παλινδρόμησης).

5.3.2 Δημιουργία Συνόλου Δεδομένων

Για την εκπαίδευση των δύο δικτύων που ενσωματώθηκαν στον αλγόριθμο σχεδιάστηκαν αντίστοιχα σύνολα δεδομένων. Ένα πρόβλημα που αφορά στη δημιουργία τους είναι ότι δεν υπάρχουν ετικέτες, δηλαδή η βέλτιστη ενέργεια σε κάθε βήμα δεν είναι γνωστή ακόμα και μετά την ολοκλήρωση του παι-

χνιδιού, καθώς η πραγματική αναμενόμενη αξία των ενεργειών παραμένει άγνωστη. Η αξιολόγηση της απόδοσης των πρακτόρων βασίζεται στα ποσοστά νίκης τους, τα οποία δεν παρέχουν πληροφορία για την επίδραση των επιμέρους ενεργειών στο αποτέλεσμα του παιχνιδιού. Για αυτόν τον λόγο, ένα μοντέλο που υλοποιεί το MAB πρόβλημα χρησιμοποιήθηκε για τη δημιουργία κατάλληλων δεδομένων για το συγκεκριμένο πρόβλημα.

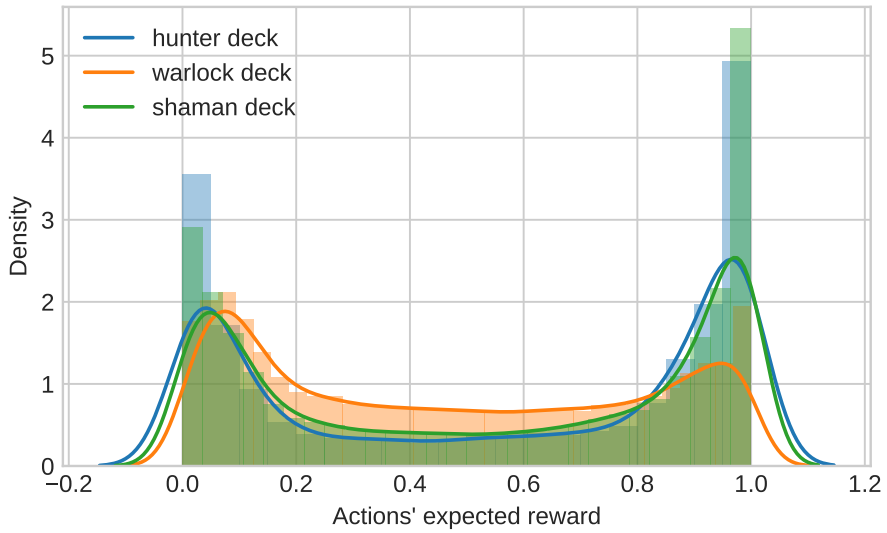
Συγκεκριμένα, σχεδιάστηκε ένα περιβάλλον προσομοίωσης του άνω ορίου εμπιστοσύνης UCB (που αποτελεί ουσιαστικά τη μέθοδο που χρησιμοποιείται στη φάση επιλογής του MCTS) στο πρόβλημα MAB. Προκειμένου να δημιουργηθεί ένα ολοκληρωμένο σύνολο δεδομένων, πραγματοποιήθηκαν διαφορετικές προσομοιώσεις για μέγεθος του χώρου ενεργειών στο εύρος $[3, 50]$. Τα συγκεκριμένα όρια επιλέχθηκαν γιατί αφενός δεν έχει νόημα το κλάδεμα όταν υπάρχουν λιγότερες από τρεις διαθέσιμες ενέργειες και αφετέρου στο περιβάλλον του συγκεκριμένου παιχνιδιού (Hearthstone) που εφαρμόζεται ο πράκτορας, ο συνολικός αριθμός ενεργειών ανά κίνηση σπάνια υπερβαίνει τις πενήντα. Για κάθε ενέργεια, η ανταμοιβή που λαμβάνεται σε κάθε βήμα προκύπτει από μία κατανομή *Bernoulli*. Συνεπώς, η αναμενόμενη ανταμοιβή κάθε ενέργειας εκφράζεται από την παράμετρο Bernoulli p, η οποία επιλέγεται τυχαία για κάθε ενέργεια. Με αυτόν τον τρόπο, η βέλτιστη ενέργεια είναι γνωστή εκ των προτέρων και μπορεί να χρησιμοποιηθεί στη συνέχεια για την εκπαίδευση των δικτύων.

Για την εκπαίδευση του δικτύου ασφάλειας, κάθε δείγμα του συνόλου δεδομένων θα πρέπει να περιλαμβάνει το πλήθος διαθέσιμων ενεργειών και τον αριθμό των επαναλήψεων που έχουν εκτελεστεί (είσοδοι του δικτύου) ενώ η αντίστοιχη ετικέτα θα πρέπει να υποδεικνύει το βέλτιστο (ελάχιστο) υποσύνολο ενεργειών που περιέχει την ενέργεια με την υψηλότερη αναμενόμενη ανταμοιβή. Για το σχηματισμό των δειγμάτων, διαφορετικές τιμές πλήθους επαναλήψεων συνδυάστηκαν με κάθε πιθανή τιμή πλήθους ενεργειών (στο έυρος $[3, 50]$). Ο ελάχιστος αριθμός επαναλήψεων που εξετάστηκαν ήταν 100 καθώς απαιτείται ένα κάτω όριο για να αξιολογήσει ο αλγόριθμος επαρκώς όλες τις ενέργειες (βλ. Ενότητα 5.3.1). Αναφορικά με το άνω όριο, οι πράκτορες που υλοποιήθηκαν έχουν δυνατότητα εκτέλεσης έως 1000 επαναλήψεων, όμως λόγω της επαναχρησιμοποίησης του δέντρου (βλ. Ενότητα 4.2.4) οι συνολικές επαναλήψεις είναι πιθανό να είναι περισσότερες σε ορισμένες περιπτώσεις. Ως εκ τούτου, το τελικό εύρος από το οποίο έγινε δειγματοληψία των επαναλήψεων για τη δημιουργία του συνόλου δεδομένων είναι $[100, 1500]$.

Για τη δημιουργία των αντίστοιχων ετικετών, πραγματοποιήθηκαν 1000 διαφορετικές προσομοιώσεις του MAB για κάθε συνδυασμό χώρου ενεργειών και αριθμού επαναλήψεων και επιλέχθηκε η καλύτερη ενέργεια με βάση το άνω όριο εμπιστοσύνης (Εξίσωση 5.5). Για κάθε προσομοίωση υπολογίστηκε το υποσύνολο ενεργειών (ταξινομημένων με βάση των αριθμό επισκέψεων τους n_i) που περιλάμβανε τη βέλτιστη ενέργεια (δηλαδή την ενέργεια με τη μεγαλύτερη τιμή p). Ο ελάχιστος αριθμός ενεργειών που βρέθηκε ότι περιλαμβάνει τη βέλτιστη ενέργεια σε όλες τις προσομοιώσεις (δηλαδή το μέγιστο μέγεθος των υποσυνόλων που προέκυψαν από τις διαφορετικές προσομοιώσεις) χρησιμοποιήθηκε ως ετικέτα για το αντίστοιχο δείγμα. Μέσω αυτής της διαδικασίας, το τελικό σύνολο δεδομένων για το δίκτυο ασφάλειας αποτελείται από 4704 δείγματα (98 για κάθε μέγεθος χώρου ενεργειών).

Αναφορικά με την εκπαίδευση του δικτύου πιθανότητας, κάθε δείγμα στο σύνολο δεδομένων πρέπει να αποτελείται από τις εναπομείνουσες επαναλήψεις του αλγορίθμου και το εναπομείνον πλήθος ενεργειών (είσοδοι του δικτύου) με την αντίστοιχη ετικέτα να είναι η πιθανότητα να επιλεγεί η βέλτιστη ενέργεια με χρήση του UCB. Ομοίως με το σύνολο δεδομένων του δικτύου ασφάλειας, διαφορετικοί συναδυασμοί επαναλήψεων-ενεργειών σχηματίστηκαν και η πιθανότητα για κάθε δείγμα υπολογίστηκε από 1000 προσομοιώσεις.

Στην παραπάνω διαδικασία, η παράμετρος p στην κατανομή Bernoulli της αξίας κάθε ενέργειας αντλείται από την ομοιόμορφη κατανομή (uniform distribution) $U[0, 1]$, δηλαδή κάθε τιμή στο $[0, 1]$ είναι εξίσου πιθανό να επιλεγεί για να αντιπροσωπεύει την αναμενόμενη αξία μίας ενέργειας. Προκειμένου να διερευνηθεί αν η γνώση της πραγματικής κατανομής μπορεί να οδηγήσει σε πιο αποτελεσματικά δίκτυα κλαδέματος, μία εκτίμηση της κατανομής της αναμενόμενης ανταμοιβής των ενεργειών υπολογίστηκε μέσω δειγμάτων από παιχνίδια στα οποία και οι δύο πράκτορες υλοποιούσαν τον απλό MCTS στο συγκεκριμένο περιβάλλον (Hearthstone). Η εκτίμηση της κατανομής για τους τρεις διαφορετικούς τύπους τράπουλας που εξετάζονται (βλ. Ενότητα 4.1) παρουσιάζεται στο Σχήμα 5.2.



Σχήμα 5.2: Εκτιμώμενη κατανομή αναμενόμενης ανταμοιβής ενεργειών στο Hearthstone

Στη συνέχεια, δύο νέα σύνολα δεδομένων (ένα για κάθε δίκτυο) δημιουργήθηκαν ακολουθώντας ακριβώς την ίδια διαδικασία, χρησιμοποιώντας όμως μία *διτροπική κατανομή* (bimodal distribution) η οποία προσεγγίζει τις κατανομές του Σχήματος 5.2 για την επιλογή της παραμέτρου p καθε ενέργειας. Τα δίκτυα ασφάλειας και πιθανότητας εκπαιδεύτηκαν εκ νέου στα αντίστοιχα σύνολα δεδομένων και ενσωματώθηκαν σε νέο πράκτορα. Η απόδοση των πρακτόρων για τις δύο διαφορετικές προσεγγίσεις όσον αφορά τα δίκτυα κλαδέματος παρουσιάζεται στην επόμενη ενότητα.

5.3.3 Πειραματική Διαδικασία

Για την αξιολόγηση της προτεινόμενης μεθοδολογίας, υλοποιούνται πράκτορες δενδρικής αναζήτησης MCTS που ενσωματώνουν τα δίκτυα κλαδέματος που περιγράφονται στην Ενότητα 5.3.1 στο περιβάλλον του παιχνιδιού Hearthstone (βλ. Ενότητα 4.1). Τρεις διαφορετικές παραλλαγές του αλγορίθμου εξετάζονται, ο απλός αλγόριθμος δενδρικής αναζήτησης Μόντε Κάρλο ενισχυμένος με την τεχνική κλαδέματος (MCTS-PN), ο τροποποιημένος MCTS με ενίσχυση της φάσης προσομοίωσης (MCTS-xgboostSLE) που παρουσιάζεται στο Κεφάλαιο 4 και ο συνδυασμός των δύο (MCTS-xgboostPN). Οι δύο τελευταίοι συγκρίνονται με τον απλό MCTS και με τον GSV (βλ. Ενότητα 4.3) καθώς επίσης και μεταξύ τους. Ο MCTS-PN συγκρίνεται μόνο με τον απλό MCTS καθώς η απουσία μεθόδου αξιολόγησης των καταστάσεων τον καθιστά λιγότερο αποτελεσματικό σε σχέση με τους άλλους πράκτορες. Ωστόσο, η αξιολόγηση της απόδοσης του είναι εξίσου σημαντική καθώς είναι πιο

αποδοτικός από άποψη υπολογιστικού κόστους και χρόνου εκτέλεσης και συνεπώς πιο κατάλληλος για εφαρμογές με αυστηρότερο χρονικό περιορισμό. Αυτό οφείλεται στη συχνότητα χρήσης των δικτύων κλαδέματος, η οποία είναι πολύ μικρότερη από την αντίστοιχη του ταξινομητή καταστάσεων ο οποίος καλείται σε κάθε επανάληψη του αλγορίθμου. Επιπλέον, ο MCTS-PN είναι εντελώς ανεξάρτητος του πεδίου εφαρμογής και μπορεί να γενικευτεί ευκολότερα και να εφαρμοστεί σε οποιοδήποτε πρόβλημα απόφασης.

Οι πίνακες 5.1–5.3 απεικονίζουν την απόδοση των προτεινόμενων πρακτόρων συγκριτικά με τους MCTS, GSV και MCTS-xgboostSLE αντίστοιχα. Σε κάθε πίνακα φαίνονται τα ποσοστά νίκης των διαφορετικών προσεγγίσεων εναντίον ενός συγκεκριμένου αλγορίθμου σε ένα σετ 300 παιχνιδιών. Όλα τα πειράματα πραγματοποιούνται ξεχωριστά για τα τρία διαφορετικά είδη τράπουλας/στρατηγικής ώστε να μην υπερτερεί εκ των προτέρων κάποια προσέγγιση. Επιπλέον, προκειμένου να εξεταστεί η επίδραση των συνολικών διαθέσιμων επαναλήψεων στη δενδρική αναζήτηση, κάθε σετ πειραμάτων έχει πραγματοποιηθεί για 300, 500 και 1000 επαναλήψεις. Ολα τα παιχνίδια παίζονται με τον ίδιο τύπο τράπουλας και τον ίδιο αριθμό επαναλήψεων και για τους δύο παίκτες.

Όπως φαίνεται στον Πίνακα 5.1, η συμπεριφορά του πράκτορα που ενσωματώνει τα δίκτυα κλαδέματος (MCTS-PN) διαφέρει ανάλογα με το σύνολο δεδομένων που χρησιμοποιήθηκε για την εκπαίδευση των δικτύων κλαδέματος. Ο πράκτορας που χρησιμοποιεί τα δίκτυα που βασίζονται στην ομοιόμορφη κατανομή (MCTS-PN(uniform)) αποκτά μεγαλύτερο πλεονέκτημα καθώς αυξάνεται ο αριθμός των επιτρεπόμενων επαναλήψεων, σε αντίθεση με την προσέγγιση που βασίζεται στη διτροπική κατανομή (MCTS-PN(bimodal)), η οποία είναι πιο αποδοτική στις περιπτώσεις με χαμηλό αριθμό επαναλήψεων. Συνολικά, ο MCTS-PN(uniform) υπερνικά τον απλό αλγόριθμο MCTS σε όλες τις περιπτώσεις (με υψηλότερη απόδοση για 1000 διαθέσιμες επαναλήψεις) ενώ ο MCTS-PN(bimodal) πετυχαίνει υψηλότερα ποσοστά νίκης για 300 και 500 διαθέσιμες επαναλήψεις αλλά υστερεί στην περίπτωση των 1000 επαναλήψεων.

Όσον αφορά στις παραλλαγές του MCTS-xgboostSLE, η χρήση του ταξινομητή ενισχύει σημαντικά την απόδοση των πρακτόρων όπως αναμενόταν ενώ ο αριθμός των επαναλήψεων παίζει σημαντικό ρόλο στην επίδραση των δικτύων κλαδέματος και σε αυτήν την περίπτωση. Ο απλός MCTS-xgboostSLE (χωρίς την τεχνική κλαδέματος) επιτυγχάνει οριακά καλύτερα ποσοστά στην περίπτωση των 500 επαναλήψεων από την παραλλαγή κλαδέματος με διτροπική κατανομή (MCTS-xgboostPN(bimodal)) εναντίον του MCTS στα δύο είδη τράπουλας (hunter και shaman) και την ξεπερνά κατά μέσο όρο (από όλα τα είδη) στις περιπτώσεις των 300 και 500 διαθέσιμων επαναλήψεων. Ωστόσο, στα επτά από τα εννέα διαφορετικά πειράματα (συνδυασμούς τράπουλας και επαναλήψεων) τουλάχιστον μία από τις παραλλαγές κλαδέματος εξασφαλίζει καλύτερα αποτελέσματα από τον MCTS-xgboostSLE, ενώ η προσέγγιση που βασίζεται στην ομοιόμορφη κατανομή MCTS-xgboostPN(uniform) πετυχαίνει κατά μέσο όρο (για τις τρεις διαφορετικές στρατηγικές) τα υψηλότερα ποσοστά νίκης εναντίον του MCTS από όλες τις παραλλαγές για κάθε αριθμό επαναλήψεων που εξετάστηκε.

Παρόλο που ο συνδυασμός επιβλεπόμενης μάθησης και δικτύων κλαδέματος οδηγεί σε βελτίωση του MCTS-gxboostSLE (χωρίς κλάδεμα) για 1000 επαναλήψεις (+4.45%), η απόδοση όλων των πρακτόρων μεμονωμένα φαίνεται να μειώνεται καθώς αυξάνεται ο συνολικός αριθμός επαναλήψεων. Αυτό αποδίδεται ωστόσο, στον υψηλότερο ρυθμό βελτίωσης του MCTS και όχι σε μείωση της αποτελεσματικότητας των εξεταζόμενων αλγορίθμων. Οι πράκτορες που κάνουν χρήση του μοντέλου αξιολόγησης των καταστάσεων μπορούν να πετύχουν υψηλή απόδοση με σχετικά μικρό αριθμό επαναλήψεων και ως εκ τούτου σταθεροποιούνται γρηγορότερα. Αντίθετα, η απλή εκδοχή του MCTS επωφελείται σε μεγαλύτερο βαθμο από την αύξηση των υπολογιστικών πόρων (καθώς για μικρό α-

	Iterations			Deck Type
	300	**500**	**1000**	
MCTS-PN(uniform)	52.67	52.0	55.33	**hunter**
MCTS-PN(bimodal)	54.0	50.0	48.67	
MCTS-xgboostSLE	71.33	**68.5**	61.67	
MCTS-xgboostPN (uniform)	**76.0**	68.33	**70.33**	
MCTS-xgboostPN (bimodal)	68.67	67.67	64.0	
MCTS-PN(uniform)	49.67	51.67	54.0	**warlock**
MCTS-PN(bimodal)	47.67	56.33	48.0	
MCTS-xgboostSLE	75.0	75.5	72.33	
MCTS-xgboostPN (uniform)	**78.33**	**80.67**	74.0	
MCTS-xgboostPN (bimodal)	75.67	78.67	**75.67**	
MCTS-PN(uniform)	51.67	55.67	54.0	**shaman**
MCTS-PN(bimodal)	54.33	49.33	50.0	
MCTS-xgboostSLE	74.33	**75.5**	68.33	
MCTS-xgboostPN (uniform)	73.67	74.0	**71.33**	
MCTS-xgboostPN (bimodal)	**75.67**	72.0	70.0	
MCTS-PN(uniform)	51.33	53.11	54.43	**overall**
MCTS-PN(bimodal)	52.0	51.89	48.89	
MCTS-xgboostSLE	73.55	73.17	67.44	
MCTS-xgboostPN (uniform)	**76.0**	**74.33**	**71.89**	
MCTS-xgboostPN (bimodal)	73.33	72.78	69.89	

Πίνακας 5.1: Ποσοστό νίκης ανά αριθμό επαναλήψεων εναντίον του MCTS (το υψηλότερο ποσοστό ανά τύπο τράπουλας και αριθμό επαναλήψεων παρουσιάζεται εντονότερα)

ριθμό επαναλήψεων είναι πιθανότερο να αποτύχει να επιλέξει τη βέλτιστη ενέργεια), με αποτέλεσμα η συνολική διαφορά στα ποσοστά νίκης από τους ενισχυμένους πράκτορες να μειώνεται όσο αυξάνεται ο αριθμός επαναλήψεων.

Στη συνέχεια, οι προσεγγίσεις κλαδέματος με τον ενσωματωμένο ταξινομητή συγκρίνονται με τον αλγόριθμο GSV. Τα αντίστοιχα ποσοστά νίκης φαίνονται στον Πίνακα 5.2. Είναι σαφές ότι κάθε σετ πειραμάτων (ανάλογα με το είδος στρατηγικής που χρησιμοποιείται) έχει διαφορετικό βαθμό δυσκολίας που δε μπορούσε να αποτυπωθεί ξεκάθαρα στη σύγκριση με τον MCTS (Πίνακας 5.1). Συγκεκριμένα, η πρώτη περίπτωση (hunter) φαίνεται να είναι η πιο εύκολη όσον αφορά στην αναζήτηση της καλύτερης ενέργειας, με τους τρεις πράκτορες να ξεπερνούν σε ποσοστό επιτυχίας τον GSV. Από την άλλη πλευρά, η τρίτη περίπτωση (shaman) είναι η πιο σύνθετη, με την καλύτερη παραλλαγή του προτεινόμενου πράκτορα να φτάνει ποσοστό νίκης 26.67% για 1000 επαναλήψεις.

Ομοίως με τις δοκιμές εναντίον του MCTS, ο MCTS-xgboostSLE πετυχαίνει οριακά υψηλότερο ποσοστό νίκης (κατά μέσο όρο για τις τρεις περιπτώσεις) έναντι του GSV από ότι οι παραλλαγές κλαδέματος στην περίπτωση των 500 επαναλήψεων, ενώ υστερεί στις άλλες δύο κατηγορίες. Η παραλλαγή κλαδέματος που βασίζεται στη διτροπική κατανομή υπερτερεί για μικρό αριθμό επαναλήψεων (300) ενώ η προσέγγιση με την ομοιόμορφη κατανομή είναι η βέλτιστη όταν υπάρχουν περισσότεροι διαθέσιμοι υπολογιστικοί πόροι. Όσον αφορά στην κάθε παραλλαγή ξεχωριστά, υπάρχει εμφανής ενίσχυση της

	Iterations			Deck Type
	300	**500**	**1000**	
MCTS-xgboostSLE	46.67	50.5	**56.0**	**hunter**
MCTS-xgboostPN (uniform)	44.33	**55.67**	55.67	
MCTS-xgboostPN (bimodal)	**51.0**	52.33	55.67	
MCTS-xgboostSLE	32.67	**43.0**	46.0	**warlock**
MCTS-xgboostPN (uniform)	**37.67**	40.67	49.0	
MCTS-xgboostPN (bimodal)	34.67	41.33	**49.33**	
MCTS-xgboostSLE	**17.33**	**25.0**	24.0	**shaman**
MCTS-xgboostPN (uniform)	15.33	21.67	**26.67**	
MCTS-xgboostPN (bimodal)	17.0	20.67	25.67	
MCTS-xgboostSLE	32.22	**39.5**	42.0	**overall**
MCTS-xgboostPN (uniform)	32.44	39.34	**43.78**	
MCTS-xgboostPN (bimodal)	**34.22**	38.11	43.56	

Πίνακας 5.2: Ποσοστό νίκης ανά αριθμό επαναλήψεων εναντίον του GSV (το υψηλότερο ποσοστό ανά τύπο τράπουλας και αριθμό επαναλήψεων παρουσιάζεται εντονότερα)

απόδοσης καθώς εκτελούνται περισσότερες επαναλήψεις, επιβεβαιώνοντας ότι η μείωση στα ποσοστά νίκης έναντι του απλού MCTS οφείλεται στον υψηλότερο ρυθμό βελτίωσης του τελευταίου.

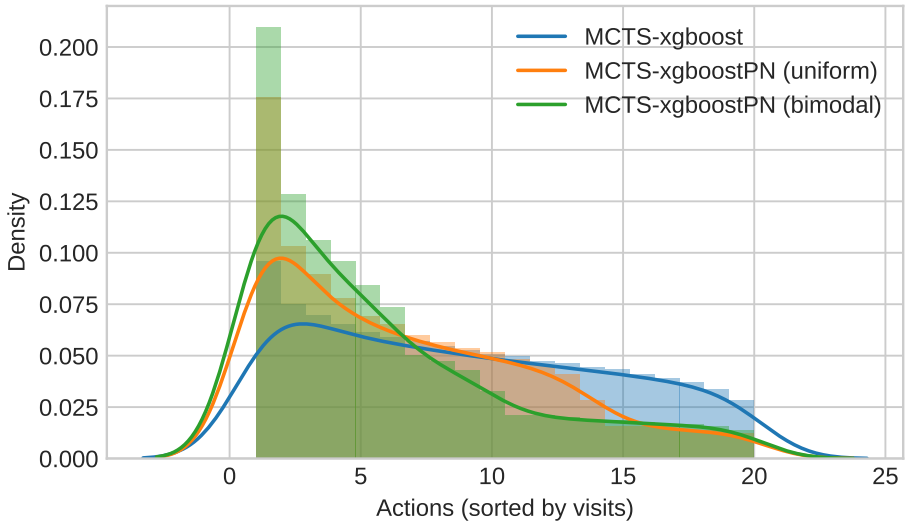
Οι δύο παραλλαγές κλαδέματος (ενισχυμένες με το μοντέλο ταξινόμησης) αξιολογούνται απευθείας σε σχέση με τον MCTS-xgboostSLE με τα αποτελέσματα να παρουσιάζονται στον Πίνακα 5.3. Παρότι σε ορισμένες περιπτώσεις ο MCTS-xgboostSLE πέτυχε υψηλότερα ποσοστά νίκης από ότι ο αλγόριθμος κλαδέματος ομοιόμορφης κατανομής όταν συγκρίθηκαν με τους MCTS και GSV (κυρίως στην περίπτωση των 500 επαναλήψεων), ο MCTS-xgboostPN(uniform) υπερτερεί σε όλες τις περιπτώσεις (με εξαίρεση μία ισοπαλία) στη μεταξύ τους σύγκριση. Αυτό ενδεχομένως οφείλεται στο ότι διαφορετικοί αλγόριθμοι μπορεί να έχουν συγκεκριμένα πλεονεκτήματα ή αδυναμίες έναντι των άλλων, όπως συμβαίνει με τους διαφορετικούς τύπους στρατηγικής που εξετάζονται και ακολουθούν το μοτίβο πέτρα-ψαλίδι-χαρτί. Ο αλγόριθμος κλαδέματος που βασίζεται στη διτροπική κατανομή είναι λιγότερο συνεπής, επιτυγχάνοντας τόσο το υψηλότερο (57.0% στην περίπτωση warlock με 500 επαναλήψεις) όσο και το χαμηλότερο (43.67% στην περίπτωση shaman με 300 επαναλήψεις) ποσοστό νίκης έναντι του MCTS-xgboostSLE. Κατά μέσο όρο για τις τρεις περιπτώσεις, ο MCTS-xgboostPN(uniform) πετυχαίνει την υψηλότερη επίδοση και στις τρεις ρυθμίσεις επαναλήψεων συγκρινόμενος με τον MCTS-xgboostSLE, ενώ ο MCTS-xgboostPN(bimodal) τον ξεπερνά μόνο στην περίπτωση των 500 επαναλήψεων.

Συνολικά, παρατηρείται ότι τα δίκτυα που εκπαιδεύτηκαν σε δεδομένα τα οποία δημιουργήθηκαν με χρήση της διτροπικής κατανομής οδηγούν σε πιο έντονο κλάδεμα (περισσότερες κινήσεις σε μικρότερο χρονικό διάστημα). Αυτό συμβαίνει γιατί τα σύνολα δεδομένων που χρησιμοποιούνται για την εκπαίδευση τους αποτελούνται από ενέργειες που είναι πιο εύκολα διαχωρίσιμες (με βάση την αναμενόμενη αξία τους). Από την άλλη πλευρά, τα δεδομένα που συντίθενται με χρήση της ομοιόμορφης κατανομής αποτελούνται από περισσότερες παρόμοιες (από άποψη αναμενόμενης ανταμοιβής) ενέργειες και οδηγούν τα αντίστοιχα δίκτυα σε πιο συντηρητικό κλάδεμα. Αυτό το φαινόμενο επιβεβαιώνεται από την κατανομή των επισκέψεων των ενεργειών-κόμβων κατά την αναζήτηση σε κάθε περίπτωση.

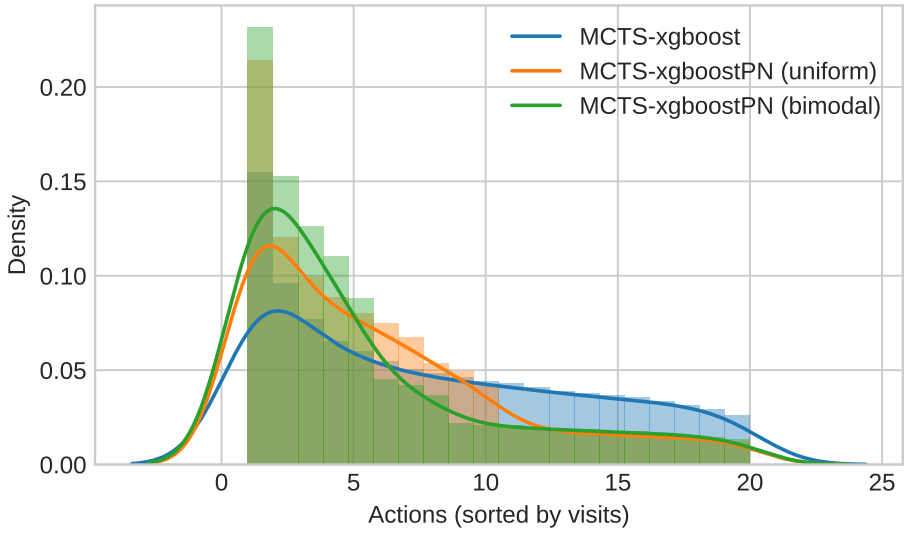
	Iterations			Deck Type
	300	**500**	**1000**	
MCTS-xgboostPN (uniform)	53.0	**54.67**	**52.33**	**hunter**
MCTS-xgboostPN (bimodal)	**54.67**	45.33	44.0	
MCTS-xgboostPN (uniform)	**54.0**	52.67	**54.33**	**warlock**
MCTS-xgboostPN (bimodal)	51.0	**57.0**	47.0	
MCTS-xgboostPN (uniform)	**51.67**	50.0	**53.67**	**shaman**
MCTS-xgboostPN (bimodal)	43.67	**53.67**	48.33	
MCTS-xgboostPN (uniform)	**52.89**	**52.45**	**53.44**	**overall**
MCTS-xgboostPN (bimodal)	49.78	52.0	46.44	

Πίνακας 5.3: Ποσοστό νίκης ανά αριθμό επαναλήψεων εναντίον του MCTS-xgboostSLE (το υψηλότερο ποσοστό ανά τύπο τράπουλας και αριθμό επαναλήψεων παρουσιάζεται εντονότερα)

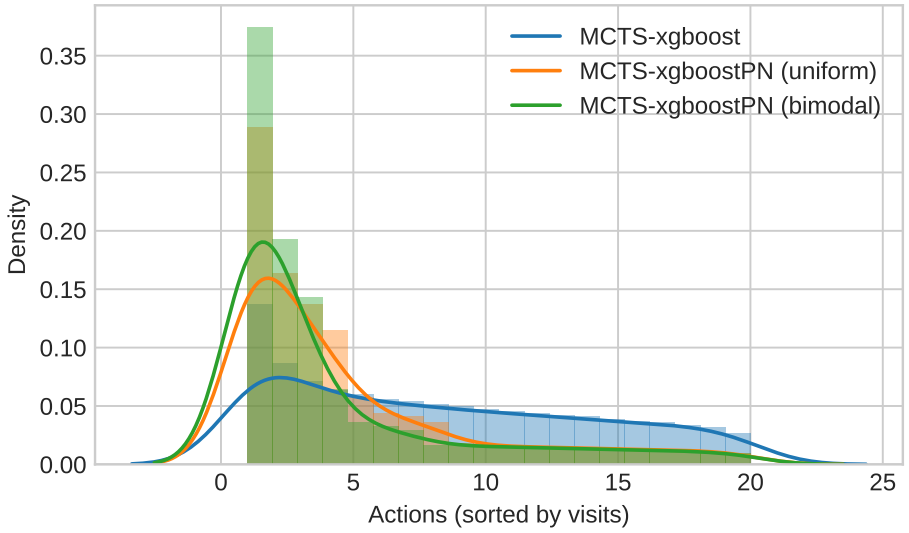
Συγκεκριμένα, οι κατανομές των επισκέψεων των κόμβων όπως προέκυψαν κατά την εκτέλεση των MCTS-xgboost, MCTS-xgboostPN(uniform) και MCTS-xgboostPN(bimodal) για 300, 500 και 1000 διαθέσιμες επαναλήψεις απεικονίζονται στα Σχήματα 5.3–5.5 αντίστοιχα. Οι ενέργειες είναι ταξινομημένες με βάση το πλήθος των επισκέψεων. Τα συγκεκριμένα γραφήματα αφορούν στην ενδιάμεση στρατηγική (shaman) για την περίπτωση που υπάρχουν είκοσι διαθέσιμες ενέργειες, ωστόσο είναι ενδεικτικά της πλειοψηφίας των περιπτώσεων που αφορούν διαφορετικούς τύπους στρατηγικής και χώρους ενεργειών. Στους πράκτορες που χρησιμοποιούν την τεχνική κλαδέματος, οι ενέργειες με χαμηλότερη αξία (όπως καθορίζονται από τον αλγόριθμο) διατρέχονται αναμενόμενα λιγότερες φορές από ότι στον MCTS-xgboostSLE, με αποτέλεσμα μεγαλύτερο ποσοστό των διαθέσιμων επαναλήψεων να αξιοποιείται για την αξιλόγηση των καλύτερων ενεργειών.



Σχήμα 5.3: Κατανομές των επισκέψεων των ενεργειών με και χωρίς δίκτυα κλαδέματος (300 επαναλήψεις).



Σχήμα 5.4: Κατανομές των επισκέψεων των ενεργειών με και χωρίς δίκτυα κλαδέματος (500 επαναλήψεις).



Σχήμα 5.5: Κατανομές των επισκέψεων των ενεργειών με και χωρίς δίκτυα κλαδέματος (1000 επαναλήψεις).

Μεταξύ των δύο προσεγγίσεων κλαδέματος, η κορυφή της κατανομής των επισκέψεων είναι υψηλότερη στην περίπτωση του αλγορίθμου που βασίζεται στη διτροπική κατανομή, που σημαίνει ότι απορρίπτεται ένα μεγαλύτερο σύνολο ενεργειών και η αναζήτηση εστιάζεται στις καλύτερες ενέργειες πιο γρήγορα από ότι συμβαίνει στον MCTS-xgboost(uniform). Αυτή η συμπεριφορά ενέχει τον κίνδυνο να κλαδευτούν υψηλής αξίας ενέργειες στο πρώτο στάδιο της διαδικασίας, αλλά θα μπορούσε να οδηγήσει σε πιο εξειδικευμένη και αποτελεσματική αναζήτηση όταν οι ενέργειες αξιολογούνται σωστά κατά τις πρώτες επισκέψεις. Αυτός είναι ο κύριος λόγος για την ασυνέπεια της παραλλαγής που στηρίζεται στη διτροπική κατανομή, σε αντίθεση με την ομοιόμορφη που είναι περισσότερο σταθερή,

επιτυγχάνοντας τα υψηλότερα συνολικά ποσοστά νίκης μεταξύ των συγκρινόμενων προσεγγίσεων στα εφτά από τα εννιά διαφορετικά σετ πειραμάτων.

Συγκεντρωτικά, η τεχνική κλαδέματος στη φάση επιλογής του MCTS παρατηρείται ότι έχει θετικό αντίκτυπο στην απόδοση του ευφυούς πράκτορα. Το μέγεθος του χώρου ενεργειών επηρεάζει αναμενόμενα την ακρίβεια της εκτίμησης της αξίας των κόμβων και κατά συνέπεια η απόρριψη ενός υποσυνόλου των ενεργειών από το δέντρο αναζήτησης οδηγεί σε συνολική βελτίωση του αλγορίθμου. Καθώς η προτεινόμενη διαδικασία κλαδέματος δεν επηρεάζει τον τρόπο λειτουργίας της δενδρικής αναζήτησης, μπορεί να ενσωματωθεί είτε ως αυτόνομη μέθοδος είτε συνδυαστικά με άλλες τροποποιήσεις. Ιδιαίτερα σημαντική αποδεικνύεται ότι είναι η κατανομή που χρησιμοποιείται για την επιλογή της αναμενόμενης αξίας κάθε ενέργειας κατά τη δημιουργία του συνόλου δεδομένων εκπαίδευσης των δικτύων κλαδέματος. Αυτή η κατανομή προσδιορίζει σε μεγάλο βαθμό τη μορφή των δειγμάτων και κατ΄ επέκταση τη συμπεριφορά των δικτύων και η επιλογή της απαιτεί μελέτη του εκάστοτε περιβάλλοντος προκειμένου να αξιοποιηθεί με το βέλτιστο δυνατό τρόπο η προοτεινόμενη μεθοδολογία.

5.4 Ταχεία Εκτίμηση Αξίας Ενέργειας βασισμένη σε Ομοιότητα Καταστάσεων

Στην προηγούμενη ενότητα παρουσιάστηκε μία μεθοδολογία ενίσχυσης της φάσης επιλογής της δενδρικής αναζήτησης Μόντε Κάρλο με χρήση επιβλεπόμενης μηχανικής μάθησης. Ωστόσο, η δυνατότητα του αλγορίθμου για επί τόπου προγραμματισμό (χωρίς να είναι προεκπαιδευμένος) και ανεξαρτήτως γνώσης πεδίου, τον καθιστούν μία πλήρως αυτόνομη μέθοδο. Υπό αυτό το πρίσμα, έχουν προταθεί διάφορες μέθοδοι για τη βελτίωση του αρχικού αλγορίθμου χωρίς τη χρήση υβριδικών μοντέλων (π.χ. ενσωμάτωση νευρωνικών δικτύων), στοχεύοντας στην καλύτερη αξιοποίηση των στατιστικών τιμών που συγκεντρώνονται κατά τη δενδρική αναζήτηση και δε λαμβάνονται υπόψη από τον αλγόριθμο στην απλή εκδοχή του.

Μία από τις σημαντικότερες προσεγγίσεις για ενίσχυση του σταδίου επιλογής χωρίς την εισαγωγή επιπλέον γνώσης πεδίου είναι η *Ταχεία Εκτίμηση Αξίας Ενέργειας* (Rapid Action Value Estimation - RAVE) [29]. Σε αυτή την παραλλαγή, για κάθε ενέργεια σε έναν κόμβο υπολογίζεται μία επιπλέον τιμή η οποία καλείται *Όλες οι Ενέργειες Σαν Πρώτες* (All Moves As First - AMAF) με βάση τα στατιστικά όλων των προσομοιώσεων στις οποίες η ενέργεια επιλέχθηκε βαθύτερα στο δέντρο αναζήτησης. Για τη βέλτιστη αξιοποίηση αυτών των στατιστικών, σε αυτή την ενότητα προτείνεται μία νέα προσέγγιση για τον προσδιορισμό των καταλληλότερων κόμβων των οποίων η τιμή AMAF μπορεί να ληφθεί υπόψη στο στάδιο επιλογής, η *Ταχεία Εκτίμηση Αξίας Ενέργειας βασισμένη σε Ομοιότητα Καταστάσεων* (State Similarity based Rapid Action Value Estimation). Στόχος είναι να καθοριστεί κατά τη φάση επιλογής του MCTS για κάθε έναν από τους υποψήφιους κόμβους-παιδιά, ο πιο κατάλληλος κόμβος (υψηλότερα στο δέντρο αναζήτησης) για τη χρησιμοποίηση της δικής του τιμής AMAF αντί της τιμής του ίδιου του κόμβου-παιδιού. Με αυτόν τον τρόπο, οι κόμβοι με μικρό αριθμό επισκέψεων κατά τη διαδικασία της αναζήτησης μπορούν να επωφεληθούν από τα στατιστικά που συλλέγονται σε *κόμβους-προγόνους* (ancestor nodes) με παρόμοια συμπεριφορά. Δύο διαφορετικές παραλλαγές εξετάζονται, με κοινό χαρακτηριστικό τον καθορισμό κόμβων που αναπαριστούν παρόμοιες καταστάσεις με βάση τις ενέργειες που εκτελέστηκαν στα αντίστοιχα μονοπάτια κατά τη δενδρική αναζήτηση. Στην πρώτη περίπτωση χρησιμοποιούνται *N-γραμμα* (N-grams) για την εύρεση παρόμοιων μονοπατιών ενώ στη δεύτερη, μία διανυσματική αναπαράσταση των ενεργειών που επιλέχθηκαν.

5.4.1 Θεωρητικό Υπόβαθρο

Σε περιβάλλοντα χωρίς γνώση πεδίου, η αποτελεσματική χρήση των στατιστικών μεταξύ των διαφορετικών κόμβων του δέντρου αναζήτησης και η αλληλεπίδρασή τους είναι κομβική για τη βελτίωση της απόδοσης του αλγορίθμου. Όπως έχει αναφερθεί, στη φάση επιλογής του αλγορίθμου δενδρικής αναζήτησης Μόντε Κάρλο επιλέγεται η ενέργεια με το υψηλότερο άνω όριο εμπιστοσύνης όπως προκύπτει από την Εξίσωση 5.6:

$$\mathrm{UCB}(a_i) = Q(s, a_i) + C\sqrt{\frac{\log N}{n_i}} \tag{5.6}$$

όπου a_i είναι μία ενέργεια που οδηγεί στον κόμβο-παιδί i, $Q(s, a_i)$ είναι η εκτίμηση της αξίας του κόμβου i, n_i είναι το πλήθος των φορών που έχει διατρεχθεί, N είναι το πλήθος των φορών που έχει διατρεχθεί ο *κόμβος-γονέας* (parent node) του και C είναι μία παράμετρος που εξισορροπεί την εκμετάλλευση και την εξερεύνηση. Στην απλή εκδοχή του MCTS η εκτίμηση της αξίας του κάθε κόμβου βασίζεται μόνο στις δικές του στατιστικές τιμές. Συγκεκριμένα, για κάθε ενέργεια η εκτιμώμενη αξία $Q(s, a_i)$ προκύπτει με βάση τα αποτελέσματα z_j των προσομοιώσεων κατά τις οποίες στην κατάσταση s επιλέχθηκε η ενέργεια a_i, σύμφωνα με την Εξίσωση 5.7.

$$\mathrm{Q}(s, a_i) = \frac{1}{n_i}\sum_{j=1}^{N} \mathbb{I}_{\mathrm{j}}(s, a_i) z_j \tag{5.7}$$

Στην παραπάνω εξίσωση, η συνάρτηση $\mathbb{I}_{\mathrm{j}}(s, a_i)$ είναι μία συνάρτηση δείκτης που υποδεικνύει αν κατά την προσομοίωση j εκτελέστηκε η ενέργεια a_i στην κατάσταση s και ισχύει ότι $n_i = \sum_{j=1}^{N} \mathbb{I}_{\mathrm{j}}(s, a_i)$.

Στην παραλλαγή ταχείας εκτίμησης αξίας ενέργειας του MCTS χρησιμοποιείται μία ευρετική που ονομάζεται όλες-οι-ενέργειες-σαν-πρώτες, στην οποία η αξία μίας ενέργειας σε έναν κόμβο ανανεώνεται και στις περιπτώσεις που η ενέργεια εκτελείται σε κάποιον άλλο κόμβο βαθύτερα στο δέντρο αναζήτησης (Εξίσωση 5.8). Για το σκοπό αυτό, τα στατιστικά των ενεργειών που εκτελούνται σε κόμβους-απογόνους αποθηκεύονται επίσης στη μνήμη και διαδίδονται στους κόμβους που βρίσκονται υψηλότερα στο δέντρο αναζήτησης (και ανήκουν στο μονοπάτι που ακολουθήθηκε μέχρι τον τρέχοντα κόμβο) προκειμένου να μειωθεί η διακύμανση της εκτίμησης της αξίας τους. Με αυτό τον τρόπο, οι ενέργειες αξιολογούνται γρηγορότερα και η αναζήτηση κατευθύνεται προς τις πιο υποσχόμενες ενέργειες. Αυτή η προσέγγιση προϋποθέτει ότι η αξία μίας ενέργειας δεν επηρεάζεται από την ακολουθία των υπόλοιπων κινήσεων και το χρονικό σημείο στο οποίο εκτελείται, εισάγοντας μεροληψία αλλά εξασφαλίζοντας ταχύτερη εκτίμηση.

$$\mathrm{Q}_{\mathrm{AMAF}}(s, a_i) = \frac{1}{\tilde{n}_i}\sum_{j=1}^{N} \tilde{\mathbb{I}}_{\mathrm{j}}(s, a_i) z_j \tag{5.8}$$

Η συνάρτηση $\tilde{\mathbb{I}}_{\mathrm{j}}(s, a_i)$ στην Εξίσωση 5.8 επιστρέφει 1 αν η κατάσταση s διατρέχθηκε σε κάποιο βήμα t της προσομοίωσης j και η ενέργεια a_i επιλέχθηκε σε οποιοδήποτε βήμα $u \geq t$ της προσομοίωσης, διαφορετικά επιστρέφει 0. Το πλήθος των προσομοιώσεων οι οποίες χρησιμοποιούνται για τον υπολογισμό της αξίας $Q_{AMAF}(s, a_i)$ είναι $\tilde{n}_i = \sum_{j=1}^{N} \tilde{\mathbb{I}}_{\mathrm{j}}(s, a_i)$.

Στον αλγόριθμο ταχείας εκτίμησης αξίας ενέργειας, η εκτιμώμενη αξία Q της Εξίσωσης 5.6, αντικαθίσταται με έναν σταθμισμένο μέσο όρο της εκτιμώμενης αξίας του κόμβου και της τιμής Q_{AMAF} (Εξίσωση 5.9).

$$Q_{RAVE}(s,a) = (1-\beta(s)) \times Q(s,a) + \beta(s) \times Q_{AMAF}(s,a) \tag{5.9}$$

όπου $\beta(s)$ είναι μία παράμετρος η οποία ρυθμίζει το βαθμό επιρροής της τιμής Q_{AMAF} στη συνολική αξία και ορίζεται ως:

$$\beta(s) = \sqrt{\frac{k}{3N(s)+k}} \tag{5.10}$$

όπου k είναι η παράμετρος που ορίζει το ρυθμό μείωσης της επίδρασης της τιμής Q_{AMAF}.

Ουσιαστικά, η συνεισφορά της τιμής Q_{AMAF} στη συνολική εκτίμηση της αξίας είναι μεγαλύτερη στο αρχικό στάδιο του αλγορίθμου και μειώνεται καθώς εκτελούνται περισσότερες επαναλήψεις. Στόχος είναι να επιταχυνθεί η διαδικασία στις πρώτες επαναλήψεις κατά τις οποίες δεν υπάρχουν αρκετά δείγματα από προσομοιώσεις και στη συνέχεια να δοθεί περισσότερη έμφαση στην αξία $Q(s,a)$ του κάθε κόμβου (Εξίσωση 5.7) η οποία είναι πιο ακριβής.

Παρότι ο αλγόριθμος ταχείας εκτίμησης αξίας ενέργειας χρησιμοποιεί και διαμοιράζει πληροφορία μεταξύ κόμβων σε διαφορετικά υποδέντρα, τόσο οι αξίες $Q(s,a)$ όσο και οι αξίες $Q_{AMAF}(s,a)$ των κόμβων που βρίσκονται χαμηλά στο δέντρο αναζήτησης (κοντά στα φύλλα του δέντρου) εξακολουθούν να βασίζονται σε μικρό αριθμό δειγμάτων και ως εκ τούτου συνεχίζουν να έχουν υψηλή διακύμανση. Για την αντιμετώπιση αυτού του φαινομένου έχει προταθεί μία παραλλαγή της μεθόδου, ο αλγόριθμος *Γενικευμένης Ταχείας Εκτίμησης Αξίας Ενέργειας* (Generalized Rapid Action Value Estimation - GRAVE) [15] με στόχο την καλύτερη εκτίμηση των αξιών αυτών των κόμβων. Στον GRAVE εισάγεται μία παράμετρος κατωφλίου ref, προκειμένου να προσδιοριστούν αρχικά οι κόμβοι που έχουν μικρό αριθμό επισκέψεων και πρέπει να αξιολογηθούν καλύτερα και εν συνεχεία αξιολογούνται με χρήση των στατιστικών από κόμβους που βρίσκονται ψηλότερα στο δέντρο αναζήτησης (και κατά συνέπεια έχουν διατρεχθεί περισσότερες φορές). Συγκεκριμένα, για κάθε κόμβο που έχει διατρεχθεί λιγότερες φορές από το ελάχιστο κατώφλι, προσδιορίζεται ο κοντινότερος κόμβος-πρόγονος στο δέντρο αναζήτησης που ικανοποιεί αυτή τη συνθήκη (δηλαδή έχει τον ελάχιστο απαιτούμενο αριθμό επισκέψεων) και χρησιμοποιείται η δική του αξία $Q_{AMAF}(s,a)$ για τον υπολογισμό της εκτιμώμενης αξίας του κόμβου-απογόνου στην Εξίσωση 5.9. Έτσι, ακόμα και για τους κόμβους που βρίσκονται πιο βαθιά στο δέντρο αναζήτησης οι εκτιμήσεις βασίζονται σε επαρκή αριθμό προσομοιώσεων.

Οπως και στην περίπτωση του RAVE, εισάγεται επιπλέον μεροληψία στις εκτιμήσεις καθώς οι καταστάσεις των κόμβων-προγόνων που χρησιμοποιούνται μπορεί να διαφέρουν σημαντικά από την πραγματική κατάσταση των κόμβων που εξετάζονται. Για αυτό το λόγο, η αξία $Q_{AMAF}(s,a)$ του κόμβου-προγόνου χρησιμοποιείται στις πρώτες επαναλήψεις ώστε να ευνοήσει τις εκτιμήσεις των κόμβων για τους οποίους δεν υπάρχει ικανή διαθέσιμη πληροφορία από τις προσομοιώσεις. Μόλις φτάσει σε ένα συγκεκριμένο σημείο (αριθμό επαναλήψεων) ο αλγόριθμος λειτουργεί ακριβώς όπως ο RAVE, δηλαδή χρησιμοποιούνται οι τιμές των ίδιων των κόμβων που εξετάζονται αντί των προγόνων τους.

Στο ίδιο μοτίβο αναπτύχθηκε μία πιο γενική παραλλαγή του RAVE, που ονομάζεται HRAVE [93], στην οποία χρησιμοποιούνται τα συνολικά στατιστικά που έχουν προκύψει για κάθε ενέργεια. Για αυτό το σκοπό, στον υπολογισμό της αξίας μίας ενέργειας χρησιμοποιείται η τιμή $Q_{AMAF}(s,a)$ της ενέργειας του κόμβου-ρίζας του δέντρου αναζήτησης. Αυτή η τεχνική αξιοποιεί όλη τη διαθέσιμη πληροφορία υπολογίζοντας τον μέσο όρο όλων των προσομοιώσεων που πραγματοποιήθηκαν για κάθε ενέργεια, με κόστος την εισαγωγή επιπλέον μεροληψίας αφού όσο περισσότερο απέχουν μεταξύ τους δύο κόμβοι στο δέντρο είναι πιθανότερο να παρουσιάζουν σημαντικότερες διαφορές στην κατάσταση

τους, με αποτέλεσμα η ίδια ενέργεια να έχει διαφορετική αξία σε κάθε περίπτωση. Στην προκειμένη περίπτωση, η απόσταση των κόμβων είναι η μέγιστη δυνατή καθώς ο κόμβος-ρίζα είναι ο μακρινότερος κόμβος-πρόγονος οποιουδήποτε κόμβου βαθύτερα στο δέντρο αναζήτησης. Συνολικά, ο HRAVE χρησιμοποιεί *καθολική* (global) πληροφορία σε αντίθεση με τον RAVE που αξιοποιεί την *τοπική* (local) πληροφορία, με τον GRAVE να αποτελεί μία ενδιάμεση προσέγγιση.

5.4.2 Ταχεία Εκτίμηση Αξίας Ενέργειας βασισμένη σε N-γραμμα

Σε αυτή την ενότητα παρουσιάζεται μία μέθοδος για την επιλογή του καταλληλότερου κόμβου στο δέντρο αναζήτησης, του οποίου τα στατιστικά μπορούν να χρησιμοποιηθούν για τη βελτίωση της εκτίμησης της αξίας ενός κόμβου-απογόνου. Οι αλγόριθμοι GRAVE και HRAVE που περιγράφηκαν παραπάνω, επιλέγουν έναν κόμβο ψηλότερα στο δέντρο με την προϋπόθεση να έχει διατρεχθεί περισσότερες φορές από ένα συγκεκριμένο κατώφλι (GRAVE) ή απλά να διατηρεί όλα τα στατιστικά που έχουν ληφθεί από τις προσομοιώσεις (HRAVE). Αυτό έχει ως αποτέλεσμα να εισάγεται σημαντικός βαθμός μεροληψίας, καθώς η κατάσταση του κόμβου που επιλέγεται είναι πιθανό να διαφέρει αρκετά από την κατάσταση του κόμβου για τον οποίο γίνεται η εκτίμηση της αξίας της ενέργειας. Προκειμένου να αντιμετωπιστεί αυτό το πρόβλημα προτείνεται ένας αλγόριθμος επιλογής κόμβων με παρόμοια κατάσταση, με βάση τις ενέργειες οι οποίες οδήγησαν στις αντίστοιχες καταστάσεις.

Στο πεδίο της τεχνητής νοημοσύνης για παιχνίδια και στον αλγόριθμο δενδρικής αναζήτησης Μόντε Κάρλο ειδικότερα, τα *N-γραμμα* (N-grams) ορίζονται ως ακολουθίες κινήσεων αποτελούμενες από N συνεχόμενες ενέργειες. Σε αυτό το πλαίσιο, έχουν χρησιμοποιηθεί στη φάση προσομοίωσης του αλγορίθμου για τον υπλογισμό μίας μέσης αξίας των ακολουθιών που απαντώνται κατά την προσομοίωση και ακολούθως την επιλογή της επόμενης ενεργειας στη φάση επέκτασης [96].

Για την ενίσχυση της φάσης επιλογής του αλγορίθμου, η προτεινόμενη τεχνική έχει ως στόχο τη χρήση N-γραμμων για τον προσδιορισμό του κόμβου που έχει την πιο όμοια κατάσταση παιχνιδιού με τον κόμβο που εξετάζεται. Η βασική ιδέα είναι ότι σε κόμβους με παρόμοια κατάσταση, οι αναμενόμενες αξίες της ίδιας ενέργειας θα είναι παραπλήσιες και ως εκ τούτου η αξιοποίηση της πληροφορίας του ενός κόμβου στον άλλο δε θα ενέχει τον κίνδυνο εισαγωγής υψηλού βαθμού μεροληψίας. Σε πολλές περιπτώσεις, οι καταστάσεις οι οποίες προκύπτουν από την ίδια ακολουθία ενεργειών είναι παρόμοιες ανεξάρτητα από την αρχική κατάσταση από την οποία προήλθαν. Υπό αυτό το πρίσμα στόχος είναι για κάθε κόμβο να βρεθεί ο *πιο κατάλληλος κόμβος* (best matching node), όσον αφορά στην κατάσταση του παιχνιδιού, ψηλότερα στο δέντρο αναζήτησης (και ενδεχομένως σε διαφορετικό υποδέντρο) και να χρησιμοποιηθεί η δική του τιμή $Q_{AMAF}(s, a)$ στον υπολογισμό της συνολικής αξίας του εξεταζόμενου κόμβου (Εξίσωση 5.9) αντί για την τιμή ενός κόμβου-προγόνου που απλώς έχει επαρκή αριθμό επισκέψεων.

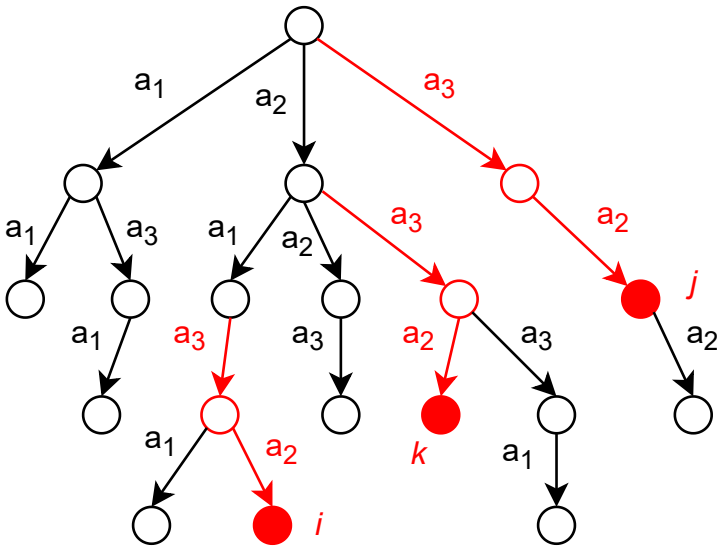
Για την υλοποίηση της προτεινόμενης μεθόδου, η ακολουθία ενεργειών που οδηγεί σε κάθε κόμβο αποθηκεύεται στη μνήμη μαζί με τον αριθμό επισκέψεων και τον αριθμό νικών που προέκυψαν από τις προσομοιώσεις. Στη φάση επιλογής, η ακολουθία κάθε υποψήφιου κόμβου-παιδιού συγκρίνεται με τις αντίστοιχες ακολουθίες των κόμβων που βρίκονται σε υψηλότερο επίπεδο στο δέντρο αναζήτησης. Δεδομένης μίας ακολουθίας ενεργειών $s_i = (a_{1i}, a_{2i}, ..., a_{ki})$ η οποία οδηγεί στον κόμβο-παιδί i και μίας ακολουθίας $s_j = (a_{1j}, a_{2j}, ..., a_{mj})$ η οποία οδηγεί στον κόμβο j υψηλότερα στο δέντρο αναζήτησης, αρχικά υπολογίζεται για κάθε ζεύγος (i, j) το πλήθος N των συνεχόμενων πανομοιότυπων ενεργειών $(a_{ki}, a_{(k-1)i}, ..., a_{(k-N+1)i}) = (a_{mj}, a_{(m-1)j}, ..., a_{(m-N+1)j})$ που εκτελέστηκαν και στα δύο μονοπάτια οδηγώντας στους κόμβους i και j αντίστοιχα. Με βάση αυτό το πλήθος, προσδιορίζεται για κάθε κόμβο-παιδί i ο καταλληλότερος κόμβος-πρόγονος j ως ο κόμβος με το μεγαλύτερο πλήθος

κοινών ενεργειών N, δηλαδή ο κόμβος με το μεγαλύτερο πανομοιότυπο N-γραμμο. Τελικά, η εκτίμηση της αναμενόμενης αξίας ενός κόμβου στον προτεινόμενο αλγόριθμο *Ταχείας Εκτίμησης Αξίας Ενέργειας βασισμένης σε N-γραμμα* (N-grams based Rapid Action Value Estimation - NRAVE) η οποία χρησιμοποιείται στη συνέχεια για τον υπολογισμό του αντίστοχου άνω ορίου εμπιστοσύνης, προκύπτει από την Εξίσωση 5.11:

$$Q_{NRAVE}(s,a) = (1-\beta(s)) \times Q(s,a) + \beta(s) \times Q_{AMAF}(s',a) \tag{5.11}$$

όπου s' είναι η κατάσταση του κόμβου που προσδιορίστηκε με την παραπάνω μέθοδο. Σε περίπτωση που υπάρχουν περισσότεροι από έναν κόμβοι με το ίδιο μέγιστο μήκος ακολουθίας N, εξετάζονται δύο διαφορετικές προσεγγίσεις. Στη μία προσέγγιση ($NRAVE$) επιλέγεται ο κόμβος που βρίσκεται υψηλότερα στο δέντρο καθώς αυτός θα έχει μεγαλύτερο αριθμό επισκέψεων και συνεπώς τα στατιστικά του ενδεχομένως να είναι πιο αξιόπιστα. Στη δεύτερη περίπτωση ($N_{av}RAVE$) υπολογίζεται ο μέσος όρος των στατιστικών από όλους τους υποψήφιους κόμβους. Περάν αυτών, υλοποιήθηκαν δύο ακόμα παραλλαγές για σταθερό μήκος ακολουθίας N ίσο με 1 και 2 (N_1RAVE και N_2RAVE αντίστοιχα), στις οποίες επιλέγονται οι κόμβοι ψηλότερα στο δέντρο αναζήτησης που έχουν πανομοιότυπα N-γραμμα του αντίστοιχου μήκους με τον εξεταζόμενο κόμβο-παιδί. Τα αποτελέσματα για όλους τους πράκτορες παρουσιάζονται στην Ενότητα 5.4.4.

Η διαδικασία εύρεσης του καταλληλότερου κόμβου για χρήση της δικής του αξίας $Q_{AMAF}(s',a)$ φαίνεται στο Σχήμα 5.6. Σε αυτή την περίπτωση, εξετάζεται ο κόμβος i και αναζητείται ο καταλληλότερος κόμβος σε υψηλότερο επίπεδο με βάση τα N-γραμμα. Το μεγαλύτερο N-γραμμο στο δέντρο αναζήτησης που ταυτίζεται με αυτό που οδηγεί στον κόμβο i έχει μήκος 2 και αποτελείται από τις ενέργειες (a_3, a_2). Δύο διαφορετικοί κόμβοι στο δέντρο σε υψηλότερο επίπεδο από τον κόμβο i προκύπτουν από το συγκεκριμένο N-γραμμο (οι κόμβοι j και k). Όπως φαίνεται, ο καταλληλότερος κόμβος δε βρίσκεται απαραίτητα σε μονοπάτι που οδηγεί στον κόμβο υπό εξέταση αλλά μπορεί να βρίσκεται σε διαφορετικό υποδέντρο.



Σχήμα 5.6: Εύρεση καταλληλότερου κόμβου στο δέντρο αναζήτησης με βάση N-γραμμα

5.4.3 Ταχεία Εκτίμηση Αξίας Ενέργειας με Συμμετρική Ακύρωση

Μία εναλλακτική προσέγγιση για τον εντοπισμό ομοιοτήτων μεταξύ των καταστάσεων του παιχνιδιού είναι να εξεταστεί πόσες φορές έχει εκτελεστεί κάθε ενέργεια κατά τη διάρκεια του παιχνιδιού αντί για τις ακολουθίες ενεργειών. Σε πολλές περιπτώσεις, η σειρά με την οποία επιλέγονται οι ενέργειες δεν επηρεάζει την κατάσταση που προκύπτει, που σημαίνει ότι παρόμοιες καταστάσεις μπορούν να επιτευχθούν από διαφορετικά μονοπάτια αν οι ίδιες κινήσεις επιλέγονται σε διαφορετικά χρονικά βήματα. Συνεπώς, για κάθε υποψήφιο κόμβο κατά την επιλογή, ο καλύτερος κόμβος θα μπορούσε να προσδιοριστεί ως αυτός με την υψηλότερη ομοιότητα όσον αφορά στο πλήθος των φορών που εκτελέστηκε κάθε ενέργεια στα αντίστοιχα μονοπάτια τους.

Ένα πρόβλημα που προκύπτει σε αυτή την περίπτωση είναι ότι τα μονοπάτια που οδηγούν σε κόμβους βαθύτερα στο δέντρο αναζήτησης έχουν μεγαλύτερο μήκος. Αυτό σημαίνει ότι οι κόμβοι που βρίσκονται σε διαφορετικά επίπεδα στο δέντρο έχουν σημαντική διαφορά στον αριθμό των εκτελεσθέντων ενεργειών, με αποτέλεσμα η σύγκριση τους όσον αφορά στο πλήθος εμφανίσεων της κάθε ενέργειας να οδηγεί σε πολύ χαμηλή ομοιότητα. Προκειμένου να είναι συγκρίσιμα τα μεγέθη, χρησιμοποιείται μία αναπαράσταση των καταστάσεων βασισμένη στην ακολουθία των ενεργειών και εν συνεχεία υπολογίζεται η ομοιότητα μεταξύ των αναπαραστάσεων. Ειδικότερα, για κάθε ζεύγος συμμετρικών (αντίθετα κατευθυνόμενων) ενεργειών (π.χ. πάνω-κάτω) υπολογίζεται μία συνολική τιμή θεωρώντας ότι αλληλοαναιρούνται και οδηγούν στην αρχική κατάσταση (π.χ. αν σε ένα μονοπάτι έχει επιλεχθεί δύο φορές η ενέργεια κίνησης προς τα κάτω και τρεις φορές η ενέργεια κίνησης προς τα πάνω, η συνολική τιμή για το συγκεκριμένο ζεύγος είναι ένα). Με αυτό τον τρόπο, μπορεί να υπολογιστεί η ομοιότητα μεταξύ δύο κόμβων σε οποιοδήποτε βάθος στο δέντρο αναζήτησης καθώς λαμβάνονται υπόψη οι διαφορές στο πλήθος εμφανίσεων των ενεργειών αντί για τις απόλυτες τιμές.

Για την υλοποίηση της συγκεκριμένης τεχνικής *Ταχείας Εκτίμησης Αξίας Ενέργειας με Συμμετρική Ακύρωση* (Cancel-out Rapid Action Value Estimation - CRAVE), ένα διάνυσμα που περιλαμβάνει το πλήθος των φορών που εκτελέστηκε κάθε ενέργεια (ή το συνολικό πλήθος αν πρόκειται για ζεύγος ενεργειών) χρησιμοποιείται για την αναπαράσταση της κατάστασης κάθε κόμβου και ο καταλληλότερος κόμβος προσδιορίζεται ως αυτός με τη μεγαλύτερη ομοιότητα (το μικρότερο μέσο τετραγωνικό σφάλμα μεταξύ των διανυσμάτων). Επειτα, η τιμή $Q_{AMAF}(s', a)$ του κόμβου που βρέθηκε χρησιμοποιείται αντί των στατιστικών του τρέχοντος κόμβου στη φάση επιλογής πανομοιότυπα με τον αλγόριθμο NRAVE (Εξίσωση 5.11). Ο CRAVE ωστόσο, έχει τον περιορισμό ότι μπορεί να εφαρμοστεί μόνο σε περιβάλλοντα με ζεύγη συμμετρικών κινήσεων (όπως περιγράφηκε παραπάνω) σε αντίθεση με τον αλγόριθμο NRAVE που μπορεί να γενικευτεί σε οποιοδήποτε περιβάλλον.

5.4.4 Αποτελέσματα

Οι αλγόριθμοι που περιγράφονται παραπάνω υλοποιήθηκαν και δοκιμάστηκαν στο περιβάλλον του GVGAI (βλ. Ενότητα 3.2). Η απόδοση των αντίστοιχων πρακτόρων συγκρίνεται με αυτή του απλού αλγορίθμου δενδρικής αναζήτησης Μόντε Κάρλο καθώς και με την απόδοση των παραλλαγών RAVE, GRAVE και HRAVE σε ένα σύνολο δέκα διαφορετικών παιχνιδιών που παρέχονται από το περιβάλλον. Οι Πίνακες 5.4 και 5.5 παρουσιάζουν το ποσοστό νίκης που πέτυχε κάθε πράκτορας σε όλα τα παιχνίδια. Αντίστοιχα, ο Πίνακας 5.6 παρουσιάζει τη μέση βαθμολογία τους. Για κάθε παιχνίδι υπάρχουν πέντε διαφορετικά επίπεδα και κάθε πράκτορας έχει δοκιμαστεί συνολικά πεντακόσιες φορές σε κάθε παιχνίδι (εκατό σε κάθε επίπεδο).

Αναφορικά με την υλοποίηση, όλοι οι πράκτορες έχουν διαθέσιμο χρόνο $40ms$ για την επιλογή

κάθε ενέργειας, που είναι το προεπιλεγμένο χρονικό όριο που διατίθεται στο περιβάλλον του διαγωνισμού GVGAI. Η υπερπαράμετρος C ορίστηκε ίση με $\sqrt{2}$ που είναι η πιο ευρέως χρησιμοποιούμενη τιμή της στην αντίστοιχη βιβλιογραφία και η παράμετρος k παίρνει την τιμή 50. Τέλος, η παράμετρος κατωφλίου ref στον αλγόριθμο GRAVE ορίστηκε ίση με 5 έπειτα από δοκιμές, σε συνδυασμό και με το μέσο αριθμό επισκέψεων του κάθε κόμβου στο χρονικό διάστημα που είναι διαθέσιμο για το συγκεκριμένο περιβάλλον.

Από τους πίνακες που αφορούν τα ποσοστά νίκης προκύπτει ότι οι προτεινόμενες παραλλαγές του αλγορίθμου (Πίνακας 5.5) είναι συγκρίσιμες με τις υπάρχουσες μεθόδους (Πίνακας 5.4). Στις εφτά από τις δέκα περιπτώσεις, το υψηλότερο ποσοστό επιτυγχάνεται από μία εκ των μεθόδων ταχείας εκτίμησης αξίας ενέργειας με ομοιότητα καταστάσεων. Συγκεκριμένα, ο πιο αποτελεσματικός αλγόριθμος είναι ο NRAVE ο οποίος ξεπερνά όλες τις εξεταζόμενες προσεγγίσεις σε τρεις περιπτώσεις και ακολουθεί ο CRAVE με δύο. Αξίζει να τονιστεί ότι σε δύο περιπτώσεις υπερτερεί ο απλός MCTS. Μία πιθανή εξήγηση για αυτό το γεγονός είναι ότι όλες οι παραλλαγές που βασίζονται στην ταχεία εκτίμηση αξίας ενέργειας εισάγουν επιπλέον υπολογιστική πολυπλοκότητα, που σημαίνει ότι οι ακριβέστερες εκτιμήσεις της αναμενόμενης αξίας έχουν κόστος όσον αφορά στο συνολικό αριθμό των επαναλήψεων που εκτελούνται. Ως εκ τούτου, σε κάποιες περιπτώσεις ενδέχεται η αύξηση της ποιότητας των εκτιμήσεων να μην αρκεί ώστε να υπερκαλύψει το όφελος που προκύπτει από το μεγαλύτερο πλήθος των προσομοιώσεων στον MCTS. Επίσης, παρότι όλοι οι αλγόριθμοι φαίνεται ότι είναι αρκετά συνεπείς, κανένας δεν υπερτερεί πλήρως των υπολοίπων όσον αφορά στο ποσοστό νικών σε όλα τα παιχνίδια, κάτι το οποίο αποδίδεται στα διαφορετικά χαρακτηριστικά και στις ιδιαιτερότητες του κάθε παιχνιδιού.

Οι αλγόριθμοι ταχείας εκτίμησης αξίας ενέργειας με N-γραμμα και με συμμετρική ακύρωση είναι εξίσου αποτελεσματικοί και όσον αφορά στη μέση βαθμολογία που πετυχαίνουν. Στον Πίνακα 5.6 παρουσιάζονται τα αντίστοιχα αποτελέσματα των πρακτόρων για κάθε παιχνίδι, με τις προτεινόμενες τεχνικές να πετυχαίνουν την υψηλότερη μέση βαθμολογία σε οχτώ από τα δέκα περιβάλλοντα. Ο αποδοτικότερος αλγόριθμος υπό αυτή την έννοια είναι ο CRAVE υπερτερώντας σε τρία διαφορετικά παιχνίδια. Παρατηρείται ωστόσο, ότι τα υψηλά ποσοστά νίκης δεν συνεπάγονται απαραίτητα υψηλή βαθμολογία και αντιστρόφως, καθώς στις περισσότερες περιπτώσεις το υψηλότερο ποσοστό νίκης και η υψηλότερη μέση βαθμολογία δεν επιτυγχάνονται από τον ίδιο αλγόριθμο. Στη συνάρτηση βαθμο-

	MCTS	**RAVE**	**GRAVE**	**HRAVE**
Angels & Demons	1.2 ± 0.95	1.0 ± 0.87	1.8 ± 1.17	2.2 ± 1.29
Boulderchase	14.2 ± 3.06	$\mathbf{17.8 \pm 3.35}$	14.4 ± 3.08	14.2 ± 3.06
Butterflies	85.2 ± 3.11	84.8 ± 3.15	83.0 ± 3.29	85.4 ± 3.10
Chopper	16.8 ± 3.28	18.4 ± 3.40	17.2 ± 3.31	20.4 ± 3.53
Defender	59.4 ± 4.30	60.8 ± 4.28	63.0 ± 4.23	60.8 ± 4.28
Hungry birds	34.8 ± 4.18	35.6 ± 4.20	34.2 ± 4.16	35.4 ± 4.19
Jaws	$\mathbf{73.0 \pm 3.89}$	71.4 ± 3.96	68.2 ± 4.08	71.6 ± 3.95
Missile command	58.2 ± 4.32	54.8 ± 4.36	57.2 ± 4.34	58.4 ± 4.32
Overload	$\mathbf{18.0 \pm 3.37}$	14.4 ± 3.08	16.2 ± 3.23	13.2 ± 2.97
Plaque attack	80.8 ± 3.45	83.6 ± 3.25	84.0 ± 3.21	80.4 ± 3.48

Πίνακας 5.4: Ποσοστό νίκης πρακτόρων αναφοράς (το υψηλότερο ποσοστό ανά παιχνίδι παρουσιάζεται εντονότερα)

	NRAVE	**N_{av}RAVE**	**N_1RAVE**	**N_2RAVE**	**CRAVE**
Angels & Demons	1.2 ± 0.95	1.8 ± 1.17	1.4 ± 1.03	$\mathbf{2.6 \pm 1.39}$	1.8 ± 1.17
Boulderchase	13.2 ± 2.97	13.0 ± 2.95	16.6 ± 3.26	11.8 ± 2.83	16.2 ± 3.23
Butterflies	$\mathbf{87.2 \pm 2.93}$	86.4 ± 3.00	86.2 ± 3.02	85.2 ± 3.11	84.4 ± 3.18
Chopper	21.6 ± 3.61	20.0 ± 3.51	19.6 ± 3.48	19.6 ± 3.48	$\mathbf{22.4 \pm 3.65}$
Defender	57.8 ± 4.33	60.2 ± 4.29	$\mathbf{63.4 \pm 4.22}$	60.6 ± 4.28	60.2 ± 4.29
Hungry birds	36.4 ± 4.22	32.0 ± 4.09	33.6 ± 4.14	36.8 ± 4.23	$\mathbf{37.2 \pm 4.24}$
Jaws	71.2 ± 3.97	71.4 ± 3.96	68.0 ± 4.09	70.8 ± 3.99	70.2 ± 4.01
Missile command	$\mathbf{59.2 \pm 4.31}$	57.0 ± 4.34	55.4 ± 4.36	55.8 ± 4.35	57.4 ± 4.33
Overload	17.0 ± 3.29	13.6 ± 3.00	14.2 ± 3.06	17.4 ± 3.32	14.6 ± 3.10
Plaque attack	$\mathbf{84.4 \pm 3.18}$	80.6 ± 3.47	82.0 ± 3.37	84.0 ± 3.21	83.6 ± 3.25

Πίνακας 5.5: Ποσοστό νίκης προτεινόμενων πρακτόρων (το υψηλότερο ποσοστό ανά παιχνίδι παρουσιάζεται εντονότερα)

λόγησης λαμβάνονται υπόψη διάφορες παράμετροι του παιχνιδιού (π.χ. συλλογή αντικειμένων κ.λπ.) που δεν είναι αναγκαίο να ικανοποιηθούν σε κάθε περίπτωση προκειμένου να ολοκληρωθεί με επιτυχία το παιχνίδι. Ως εκ τούτου, πράκτορες με διαφορετική συμπεριφορά μπορεί να εξασφαλίζουν υψηλότερη βαθμολογία χωρίς απαραίτητα να πετυχαίνουν υψηλότερα ποσοστά νίκης και αντιστρόφως.

	MCTS	**RAVE**	**GRAVE**	**HRAVE**	**NRAVE**	**N_{av}RAVE**	**N_1RAVE**	**N_2RAVE**	**CRAVE**
Angels & Demons	54.360	48.750	49.906	46.278	54.914	50.790	**55.028**	54.196	51.482
Boulderchase	12.946	13.320	12.678	12.500	12.184	12.168	**13.456**	12.532	13.124
Butterflies	31.016	30.256	30.336	30.864	**31.372**	30.836	30.988	30.804	30.804
Chopper	−3.734	−3.316	−3.898	−3.278	−3.136	−3.434	−3.138	−4.102	**−2.860**
Defender	−26.618	−25.952	**−23.380**	−24.828	−25.346	−25.008	−23.474	−25.646	−25.258
Hungry birds	34.880	35.760	34.280	35.400	36.480	32.160	33.760	36.960	**37.280**
Jaws	59.980	60.080	65.154	59.834	**80.316**	41.026	46.006	59.022	64.700
Missile command	2.844	2.806	3.060	**3.280**	3.016	2.758	2.752	2.924	2.916
Overload	14.444	14.746	14.916	14.640	14.736	14.266	13.978	**15.418**	15.110
Plaque attack	38.106	37.940	38.736	37.532	38.612	37.514	37.950	38.446	**38.840**

Πίνακας 5.6: Μέση βαθμολογία πρακτόρων (η υψηλότερη μέση βαθμολογία ανά παιχνίδι παρουσιάζεται εντονότερα)

Τα αποτελέσματα της πειραματικής διαδικασίας επιβεβαιώνουν ότι ο βαθμός ομοιότητας των καταστάσεων ως κριτήριο για την επιλογή του καταλληλότερου κόμβου στον αλγόριθμο Ταχείας Εκτίμησης Αξίας Ενέργειας οδηγεί σε αποδοτικότερη αξιοποίηση των στατιστικών τιμών των διαφορετικών κόμβων. Και στις δύο προτεινόμενες παραλλαγές, για τον προσδιορισμό της ομοιότητας μεταξύ των κόμβων χρησιμοποιείται η ακολουθία των ενεργειών που οδήγησε στην κάθε κατάσταση αναδεικνύοντας τον υψηλό βαθμό συσχέτισης ενεργειών και καταστάσεων. Η προσέγγιση NRAVE που βασίζεται σε N-γραμμα είναι η πιο αποτελεσματική και συγχρόνως αυτή με τη μεγαλύτερη δυνατότητα γενίκευσης καθώς δεν υπόκειται σε κανέναν περιορισμό όσον αφορά στο περιβάλλον εφαρμογής. Η παραλλαγή CRAVE οδηγεί επίσης σε βελτίωση του αλγορίθμου σε κάποιες περιπτώσεις, αλλά αξιοποιείται στον μέγιστο βαθμό όταν εφαρμόζεται σε περιβάλλοντα στα οποία υπάρχουν ζεύγη αλληλοσυγκρουόμενων ενεργειών. Σε κάθε περίπτωση, είναι ιδιαίτερα σημαντική η βελτιστοποίηση του αλγορίθμου ώστε

το όφελος που προκύπτει σε επίπεδο αξιοποίησης πληροφορίας από διαφορετικούς κόμβους να υπερκαλύπτει το υπολογιστικό κόστος που προστίθεται. Αυτό υπογραμμίζεται και από τα πειράματα σε περιπτώσεις στις οποίες η βελτίωση που προκύπτει από τον διαμοιρασμό των στατιστικών μεταξύ διαφορετικών κόμβων δεν επαρκεί για να αντισταθμίσει το κόστος αναζήτησης του καταλληλότερου κόμβου, με αποτέλεσμα ο απλός MCTS να είναι πιο αποδοτικός.

Κεφάλαιο 6

Ενισχυτική Μάθηση με Γεννητική Επαύξηση Δεδομένων

Η εκπαίδευση και η αποτελεσματικότητα των μοντέλων μηχανικής μάθησης εξαρτάται σε πολύ μεγάλο βαθμό από την ποιότητα αλλά και το μέγεθος του συνόλου των δεδομένων εκπαίδευσης. Στο πεδίο της ενισχυτικής μάθησης ειδικότερα, η ταχύτητα σύγκλισης και η δυνατότητα γενίκευσης των πρακτόρων αποτελούν ανοιχτό πεδίο έρευνας δεδομένης της δυσκολίας αποδοτικής αξιοποίησης των δεδομένων κατά τη διαδικασία της μάθησης. Σε αυτό το κεφάλαιο, προτείνεται μία τεχνική για *επαύξηση των δεδομένων* (data augmentation) εκπαίδευσης (που συλλέγει ο πράκτορας κατά την αλληλεπίδραση με το περιβάλλον του) σε προβλήματα ενισχυτικής μάθησης μέσω γεννητικών μοντέλων. Σε αντίθεση με τις παραδοσιακές μεθόδους επαύξησης κατά τις οποίες τα δείγματα προκύπτουν τροποποιώντας τα υπάρχοντα δεδομένα, στην προτεινόμενη προσέγγιση δημιουργούνται νέα, συνθετικά δείγματα με περισσότερη ποικιλομορφία. Για αυτόν τον σκοπό, χρησιμοποιούνται μοντέλα διάχυσης και ένα μοντέλο *αντίστροφης δυναμικής* (inverse dynamics) προκειμένου να είναι εφικτή η παραγωγή πλήρων δειγμάτων για την εκπαίδευση του ευφυούς πράκτορα. Τα συνθετικά δείγματα χρησιμοποιούνται συνδυαστικά με τα πραγματικά κατά τη διάρκεια της εκπαίδευσης, επιταχύνοντας τη διαδικασία και βελτιώνοντας τη συνολική του απόδοση.

6.1 Βιβλιογραφία

Στο πλαίσιο της βελτίωσης των αλγορίθμων ενισχυτικής μάθησης, έχουν εξεταστεί διαφορετικές μεθοδολογίες επαύξησης δεδομένων. Ειδικότερα σε περιβάλλοντα με οπτικές παρατηρήσεις (στα οποία η αναπαράσταση της κατάστασης είναι εικόνα), τόσο παραδοσιακές τεχνικές επεξεργασίας εικόνων όσο και τεχνικές μηχανικής μάθησης έχουν εφαρμοστεί παρουσιάζοντας υψηλές επιδόσεις και τονίζοντας τη σημασία της ποικιλομορφίας των δειγμάτων εκπαίδευσης. Στη συνέχεια παρουσιάζονται σχετικές έρευνες από τη βιβλιογραφία που αφορούν στην επαύξηση δεδομένων αλλά και στην πρόβλεψη ενεργειών μεταξύ διαδοχικών καταστάσεων σε περιβάλλοντα στα οποία δρουν ευφυείς πράκτορες.

Ένα σύνολο κλασικών τεχνικών επαύξησης εικόνας (όπως αποκοπή, περιστροφή, κ.λπ.) εφαρμόστηκαν σε παρτίδες εικόνων-καταστάσεων πριν δοθούν ως είσοδος για την εκπαίδευση του μοντέλου στο [58]. Αυτή η μέθοδος ενσωματώθηκε στους αλγορίθμους *Ελαστικού Δράστη-Κριτή* (Soft Actor-Critic - SAC) [33] και *Κοντινής Βελτιστοποίησης Πολιτικής* (Proximal Policy Optimization - PPO)

[89] αυξάνοντας σημαντικά την απόδοση των απλών πρακτόρων (χωρίς την επαύξηση δεδομένων). Παρόμοια αποτελέσματα είχε η εφαρμογή της ίδιας μεθοδολογίας και στον ασύγχρονο αλγόριθμο δράστη-κριτή αναδεικνύοντας τη δυνατότητα μεταφοράς της και σε ασύγχρονες προσεγγίσεις. Άλλη μία περίπτωση χρήσης παραδοσιακών τεχνικών επεξεργασίας εικόνας συναντάται στο [54]. Σε αυτή την έρευνα, αρχικά εφαρμόστηκε απλή περικοπή της εικόνας σε συνδυασμό με *παραγέμισμα* (padding) προκειμένου να μετατοπιστούν τυχαία οι αρχικές εικόνες και στη συνέχεια υπολογίστηκε η συνάρτηση αξίας ενέργειας σε ένα σύνολο παρόμοιων δειγμάτων (τα οποία παράγονται από μετασχηματισμούς της ίδιας αρχικής εικόνας). Μία βελτιωμένη, πιο αποδοτική από άποψη χρόνου, έκδοση αυτού του αλγορίθμου παρουσιάζεται στο [102].

Στις παραπάνω προσεγγίσεις, οι τεχνικές επαύξησης επιλέγονται χειροκίνητα και απαιτείται εκτεταμένος πειραματισμός προκειμένου να προσδιοριστεί η βέλτιστη μέθοδος ανά περίπτωση. Αυτό οφείλεται στο γεγονός ότι το αποτέλεσμα ενός μετασχηματισμού μπορεί να οδηγήσει σε διαφορετική συμπεριφορά ανάλογα με τα χαρακτηριστικά της αρχικής εικόνας. Για την αντιμετώπιση αυτού του προβλήματος έχουν διερευνηθεί αρκετές μέθοδοι αυτοματοποίησης της διαδικασίας επιλογής της καταλληλότερης τεχνικής, τόσο σε προβλήματα επιβλεπόμενης όσο και σε προβλήματα ενισχυτικής μάθησης. Στο [23], ένας αλγόριθμος αναζήτησης βασισμένος σε *αναδρομικά νευρωνικά δίκτυα* (Recurrent Neural Networks - RNNs) χρησιμοποιείται για αυτόματη επαύξηση, βελτιώνοντας την ακρίβεια του μοντέλου στην ταξινόμηση εικόνων. Στο ίδιο πλαίσιο, οι συγγραφείς στο [61] παρουσιάζουν μία διαφορίσιμη πολιτική επιλογής μειώνοντας το υπολογιστικό κόστος της αναζήτησης. Όσον αφορά στο πεδίο της ενισχυτικής μάθησης, η έννοια της αυτοματοποίησης της επιλογής των μεθόδων επαύξησης δεδομένων έχει διερευνηθεί κυρίως μέσω της *μετα-μάθησης* (meta-learning) και των ανώτατων ορίων εμπιστοσύνης. Αυτές οι στρατηγικές έχουν χρησιμοποιηθεί για τον καθορισμό της καταλληλότερης προσέγγισης σε αντίστοιχα προβλήματα, επιτυγχάνοντας αξιοσημείωτη βελτίωση της απόδοσης του πράκτορα [31, 82].

Πέραν των κλασικών μεθόδων επαύξησης εικόνων, τεχνικές μηχανικής μάθησης έχουν επίσης εξεταστεί με στόχο την εκπαίδευση σε δεδομένα με μεγαλύτερη ποικιλομορφία. Στο [34], εκπαιδεύεται ένα μοντέλο αντιστοίχισης των καταστάσεων σε έναν λανθάνοντα χώρο, το οποίο εν συνεχεία χρησιμοποιείται για την παραγωγή ακολουθιών ενεργειών-καταστάσεων σε αυτόν τον απλούστερο διανυσματικό χώρο. Σε αυτή την περίπτωση ο πράκτορας εκπαιδεύεται απευθείας σε δείγματα του λανθάνοντα χώρου, τα οποία προέρχονται είτε από πραγματικά δείγματα είτε από *φανταστικά* (imagined). Μία βελτιστοποιημένη παραλλαγή του ίδιου αλγορίθμου παρουσιάζεται στο [36]. Η ιδέα της αντιστοίχισης της αρχικής αναπαράστασης των καταστάσεων σε έναν λανθάνοντα χώρο εξετάζεται και στο [38]. Σε αυτή την περίπτωση, η διαδικασία της επαύξησης αποσυνδέεται από την εκμάθηση της πολιτικής προκειμένου η εκπαίδευση να επωφεληθεί από τη χρήση των νέων δεδομένων χωρίς την εισαγωγή περαιτέρω πολυπλοκότητας.

Μία μεθοδολογία με στόχο τη δημιουργία ανταγωνιστικών ακολουθιών, οι οποίες συνδυάζονται με τις πραγματικές κατά τη διάρκεια της εκπαίδευσης προκειμένου να ενισχυθεί η δυνατότητα γενίκευσης σε διαφορετικά περιβάλλοντα, περιγράφεται στο [105]. Στο ίδιο πλαίσιο, επαυξημένα και πραγματικά δεδομένα χρησιμοποιούνται συγχρόνως για την από κοινού βελτιστοποίηση της συνάρτησης αξίας ενέργειας ως προς μία επαναπροσδιορισμένη αντικειμενική συνάρτηση [37]. Μία διαφορετική προσέγγιση προκειμένου να αποφευχθεί η αστάθεια της εκπαίδευσης ακολουθείται στο [103]. Σε αυτή την έρευνα δίνεται έμφαση στη συσχέτιση των *εικονοστοιχείων* (pixels) με τον εκάστοτε στόχο και προτείνεται μία μεθοδολογία για τον μετασχηματισμό μόνο των λιγότερο σχετικών εικονοστοιχείων ώστε να μην αποπροσανατολίζεται η πολιτική του πράκτορα.

Αναφορικά με την έννοια της πρόβλεψης ενεργειών με βάση δεδομένες ακολουθίες καταστάσε-

ων (καρέ εικόνων), έχει μελετηθεί στο γενικότερο πεδίο της όρασης υπολογιστών καθώς και σε προβλήματα ενισχυτικής μάθησης, κατά βάση μέσω μοντέλων αντίστροφης δυναμικής. Στα [2, 49], οι συγγραφείς πρότειναν ένα μοντέλο με δύο *ροές* (streams) βασισμένο σε *συνελικτικά νευρωνικά δίκτυα* (Convolutional Neural Networks - CNNs) [59] για να προβλέψουν τον μετασχηματισμό μεταξύ δύο εικόνων, ως μεθοδολογία για την εκμάθηση σημαντικών χαρακτηριστικών. Μία παρόμοια τεχνική παρουσιάζεται στο [69], για την πρόβλεψη της ενέργειας μεταξύ των καταστάσεων σε ένα περιβάλλον ενισχυτικής μάθησης όπου οι αντίστοιχες αναπαραστάσεις είναι εικόνες. Το δίκτυο χωρίζεται και σε αυτή την περίπτωση σε δύο διαφορετικές ροές (μία για καθεμία από τις δύο εικόνες εισόδου), οι οποίες στη συνέχεια συνενώνονται και τροφοδοτούνται σε πλήρως συνδεδεμένα επίπεδα για την πρόβλεψη της τελικής εξόδου. Κάθε ροή αποτελείται από συνελικτικά φίλτρα σύμφωνα με την αρχιτεκτονική AlexNet [56] και τα βάρη μοιράζονται από τις δύο ροές.

Η ιδέα της χρήσης ενός μοντέλου πρόβλεψης ενέργειας στο πλαίσιο της ενισχυτικής μάθησης αναπτύχθηκε περαιτέρω στο [76]. Σε αυτή την έρευνα ένα μοντέλο αντίστροφης δυναμικής συνδυάζεται με ένα πλήρως συνδεδεμένο δίκτυο, με στόχο την παραγωγή μίας εσωτερικής ανταμοιβής (ανεξάρτητης από την ανταμοιβή που παρέχεται από το περιβάλλον) η οποία θα συμβάλλει στη βελτίωση της στρατηγικής εξερεύνησης. Τόσο η τρέχουσα όσο και η επόμενη κατάσταση (που αποτελούν τις εισόδους του συστήματος) τροφοδοτούνται αρχικά σε έναν *κωδικοποιητή* (encoder) για την εξαγωγή των σημαντικότερων χαρακτηριστικών σε έναν λανθάνοντα χώρο, τα οποία χρησιμοποιούνται ακολούθως για την πρόβλεψη της ενέργειας. Μία παρόμοια αρχιτεκτονική προτείνεται στο [75], όπου κωδικοποιητές που βασίζονται στο ResNet [40] ακολουθούνται από *δίκτυα μακράς βραχύχρονης μνήμης* (Long Short-Term Memory - LSTMs) [42] για την πρόβλεψη αλληλουχιών από ενέργειες που οδηγούν στην επίτευξη συγκεκριμένων στόχων.

Στο [17] ένα νευρωνικό δίκτυο χρησιμοποιείται επίσης για την πρόβλεψη ενεργειών, με στόχο τη μεταφορά γνώσης από περιβάλλοντα προσομοίωσης σε προβλήματα του πραγματικού κόσμου, σε περιπτώσεις όπου αυτό είναι εφικτό με βάση τις φυσικές ιδιότητες του πραγματικού περιβάλλοντος. Στα περιβάλλοντα που χρησιμοποιήθηκαν σε αυτή τη μελέτη η αναπαράσταση των καταστάσεων γίνεται με διανύσματα χαρακτηριστικών. Ως εκ τούτου, σε αυτή την περίπτωση η είσοδος του μοντέλου είναι μία συνένωση πολλών διανυσμάτων καταστάσεων και ενεργειών (προκειμένου να αναπαρασταθεί μία ακολουθία καταστάσεων και ενεργειών) και το μοντέλο αποτελείται από διαδοχικά πλήρως συνδεδεμένα επίπεδα τα οποία καταλήγουν στο επίπεδο εξόδου.

Πέραν των μοντέλων αντίστροφης δυναμικής, έχουν πραγματοποιηθεί αρκετές μελέτες με στόχο την ανάπτυξη *πρόσθιων μοντέλων* (forward models) ικανών να προβλέπουν-παράγουν απευθείας την επόμενη κατάσταση δεδομένης της τρέχουσας κατάστασης και της ενέργειας προς εκτέλεση. Παρότι στην παρούσα έρευνα δεν χρησιμοποιήθηκαν τέτοιου τύπου μοντέλα, ορισμένες από τις σημαντικότερες σχετικές μελέτες παρατίθενται στη συνέχεια. Καθώς και σε αυτή την περίπτωση η αναπαράσταση της κατάστασης που δίνεται ως είσοδος είναι εικόνα, οι διαφορετικές αρχιτεκτονικές που έχουν προταθεί βασίζονται επίσης σε συνελικτικά νευρωνικά δίκτυα.

Ένα μοντέλο πρόβλεψης της δυναμικής του περιβάλλοντος με στοιχεία ντετερμινιστικής και στοχαστικής μετάβασης παρουσιάζεται στο [35] για σχεδιασμό σε πραγματικό χρόνο. Στο ίδιο πλαίσιο, στο [104] αρχικά εκπαιδευέται ένα νευρωνικό δίκτυο για την πρόβλεψη μελλοντικών καταστάσεων και στη συνέχεια χρησιμοποιείται ένα μοντέλο με αντίστροφη λογική για τη δημιουργία ζευγών προηγούμενων καταστάσεων – ενεργειών, σχηματίζοντας μία κυκλική τροχιά. Εμπνευσμένο από την έννοια της *περιέργειας* (curiosity), ένα πρόσθιο μοντέλο έχει σχεδιαστεί στο [71] με στόχο να ενισχύσει τον κωδικοποιητή εικόνας όσον αφορά στην αξιοποίηση της χρονικής πληροφορίας, παρέχοντας ταυτόχρονα εσωτερικές ανταμοιβές για τη βελτίωση της πολιτικής εξερεύνησης του πράκτορα. Μία αντίστροφη

προσέγγιση όσον αφορά στα μοντέλα δυναμικής και στην επαύξηση δεδομένων παρουσιάζεται στο [47]. Σε αυτήν την περίπτωση, η διαδικασία επαύξησης δεδομένων λαμβάνει χώρα στην αρχική φάση του αλγορίθμου για τη δημιουργία *προβολών* (views) (δηλαδή αναπαραστάσεων σε κάποιον λανθάνοντα χώρο) των καταστάσεων που έχει συλλέξει ο πράκτορας, οι οποίες στη συνέχεια χρησιμοποιούνται για την εκπαίδευση ενός μοντέλου δυναμικής του περιβάλλοντος (σε αυτόν τον νέο χώρο καταστάσεων).

Μία αρχιτεκτονική βασισμένη σε κωδικοποιητή-αποκωδικοποιητή για την πρόβλεψη της επόμενης κατάστασης στα περιβάλλοντα του Gym Atari [12] έχει επίσης προταθεί [73]. Δύο διαφορετικές παραλλαγές δοκιμάστηκαν για το ενδιάμεσο επίπεδο μετασχηματισμού, οδηγώντας σε αρχιτεκτονικές πρόσθιου και αναδρομικού δικτύου αντίστοιχα, παράγοντας ρεαλιστικά καρέ καταστάσεων. Μία επέκταση αυτής της μεθοδολογίας προτείνεται στο [60] με στόχο την πρόβλεψη και της άμεσης ανταμοιβής που λαμβάνεται εκτός από την επόμενη κατάσταση. Σε αυτή την περίπτωση, ένα δεύτερο "κλαδί" του δικτύου που περιλαμβάνει ένα πρόσθιο επίπεδο, συνδέεται με την έξοδο του επιπέδου μετασχηματισμού και δέχεται ως είσοδο τις συνδυασμένες, υψηλού επιπέδου πληροφορίες. Με αυτόν τον τρόπο το τελικό μοντέλο μπορεί να προβλέψει συγχρόνως το επόμενο καρέ και την άμεση ανταμοιβή πολύ αποτελεσματικά. Μία διαφορετική προσέγγιση στην οποία το μοντέλο αποτελείται από πολλά μπλοκ, καθένα από τα οποία περιλαμβάνει συνελικτικά επίπεδα και ένα ειδικά σχεδιασμένο μπλοκ *συμψηφισμού* (pooling) παρουσιάζεται στο [81]. Εδώ, οι συγγραφείς προτείνουν την αναπαράσταση της ενέργειας σε μορφή στοίβας πλαισίων (παρόμοια με την κωδικοποίηση one-hot όπου κάθε ψηφίο αντιστοιχίζεται σε ένα ολόκληρο καρέ) στις ίδιες διαστάσεις με τα καρέ της κατάστασης προτού δοθεί ως είσοδος του δικτύου, ώστε να μπορεί να υποστεί την ίδια επεξεργασία με την εικόνα της τρέχουσας κατάστασης.

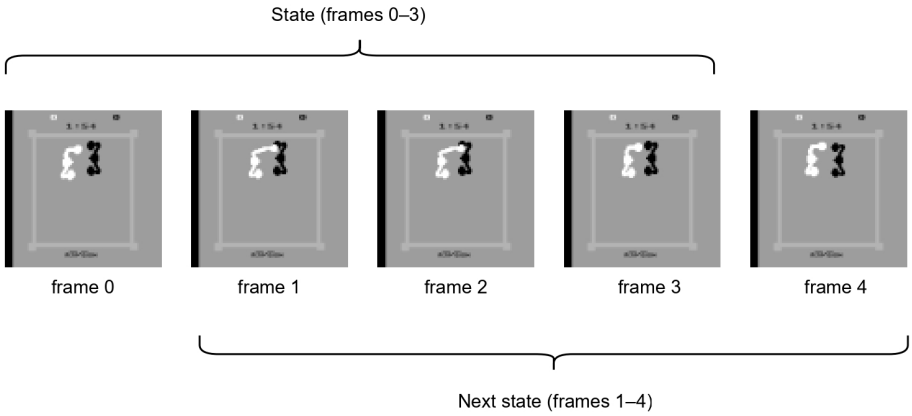
6.2 Περιβάλλον Υλοποίησης

Η μεθοδολογία που περιγράφεται σε αυτό κεφάλαιο υλοποιήθηκε και αξιολογήθηκε στο περιβάλλον *Atari Learning Environment (ALE)* [7]. Το ALE είναι ένα λογισμικό προσομοίωσης που δίνει τη δυνατότητα ανάπτυξης ευφυών πρακτόρων και δοκιμής τους σε περιβάλλοντα παιχνιδιών του κλασικού Atari 2600. Ουσιαστικά μετασχηματίζει κάθε παιχνίδι στην τυπική μορφή ενός προβλήματος ενισχυτικής μάθησης, αποτελούμενο από διακριτές καταστάσεις, ενέργειες, συσσωρευμένη ανταμοιβή και την πληροφορία για την ολοκλήρωση ή μη του παιχνιδιού.

Συγκεκριμένα, επιτρέπει στον χρήστη να στέλνει και να λαμβάνει πληροφορίες όπως την ενέργεια προς εκτέλεση, πληροφορίες που προσδιορίζουν την τρέχουσα κατάσταση του παιχνιδιού κ.λπ., είτε απευθείας μέσω της οθόνης είτε μέσω μνήμης RAM (με τιμές συγκεκριμένων παραμέτρων). Στην περίπτωση των οπτικών παρατηρήσεων, κάθε κατάσταση αποτελείται από ένα καρέ οθόνης παιχνιδιού το οποίο αναπαριστάται με έναν διδιάστατο πίνακα 160×210 εικονοστοιχείων. Όσον αφορά στον χώρο ενεργειών, περιλαμβάνει δεκαοχτώ διακριτές ενέργειες, ωστόσο το υποσύνολο των διαθέσιμων ενεργειών διαφέρει ανά παιχνίδι. Η άμεση ανταμοιβή υπολογίζεται ως η διαφορά μεταξύ της συσσωρευεμένης ανταμοιβής του πράκτορα σε δύο διαδοχικές καταστάσεις ενώ κάθε παιχνίδι που ολοκληρώνεται θεωρείται ως ένα επεισόδιο με συνέπεια το περιβάλλον να επαναορίζεται σε μία αρχική κατάσταση.

Προτού τεθεί σε λειτουργία το περιβάλλον και ξεκινήσει η εκπαίδευση των πρακτόρων που περιγράφονται στη συνέχεια, ακολουθούνται ορισμένα βήματα προεπεξεργασίας τα οποία συναντώνται στην πλειοψηφία των αντίστοιχων ερευνών της πρόσφατης βιβλιογραφίας. Κατ΄ αρχάς, οι οπτικές παρατηρήσεις μετασχηματίζονται ώστε το τελικό τους μέγεθος να είναι 84×84 και συγχρόνως τα εικονοστοιχεία αντιστοιχίζονται στην κλίμακα του γκρι. Επιπλέον, προκειμένου να είναι διαθέσιμη

η πληροφορία της κίνησης, κάθε κατάσταση του παιχνιδιού ορίζεται ως τέσσερα συνεχόμενα καρέ εικόνων. Με αυτόν τον τρόπο δίνεται η δυνατότητα στον πράκτορα να εντοπίσει και να λάβει υπόψη του για παράδειγμα την κατεύθυνση ή την ταχύτητα της κίνησης ενός αντικειμένου, κάτι που δε θα ήταν εφικτό με την επεξεργασία μίας μόνο εικόνας. Ως εκ τούτου, κάθε κατάσταση αναπαριστάται από τέσσερα διαδοχικά καρέ σε κλίμακα του γκρι, τα οποία έχουν περικοπεί σε 84×84 εικονοστοιχεία. Απόρροια αυτού είναι να υπάρχει επικάλυψη μεταξύ διαδοχικών καταστάσεων καθώς τα τρία τελευταία καρέ μίας κατάστασης είναι ουσιαστικά τα ίδια με τα τρία πρώτα καρέ της αμέσως επόμενης (π.χ. αν $s_1 = [f0, f1, f2, f3]$ είναι μία κατάσταση που αποτελείται από τέσσερα καρέ, τότε η αμέσως επόμενη θα είναι $s_2 = [f1, f2, f3, f4]$). Ένα παράδειγμα καταστάσεων που λαμβάνονται από το περιβάλλον Boxing του ALE απεικονίζεται στο Σχήμα 6.1.



Σχήμα 6.1: Αναπαράσταση καταστάσεων στο παιχνίδι Boxing του ALE μετά την προεπεξεργασία

6.3 Μεθοδολογία

Στο πλαίσιο της ενισχυτικής μάθησης, τα δείγματα εκπαίδευσης είναι πλειάδες της μορφής (s, a, r, s'), αποτελούμενες από μία κατάσταση s, μία ενέργεια a, μία άμεση ανταμοιβή r και την προκύπτουσα κατάσταση s'. Η προτεινόμενη διαδικασία δημιουργίας συνθετικών δεδομένων χωρίζεται σε δύο φάσεις. Αρχικά, ένα μοντέλο διάχυσης χρησιμοποιείται για τη δημιουργία της αρχικής κατάστασης s του κάθε δείγματος καθώς και της επόμενης κατάστασης s', μέσω της εκμάθησης της κατανομής των καταστάσεων στις οποίες έχει ήδη βρεθεί ο πράκτορας. Στη συνέχεια, κάθε νέο ζεύγος (s, s') τροφοδοτείται σε ένα μοντέλο αντίστροφης δυναμικής, εκπαιδευμένο για την πρόβλεψη της ενέργειας που οδήγησε σε μια συγκεκριμένη μετάβαση μεταξύ δύο καταστάσεων. Με αυτόν τον τρόπο από τα δύο δίκτυα παράγονται τριπλέτες της μορφής (s, a, s'). Μία εναλλακτική προσέγγιση θα ήταν το δίκτυο διάχυσης να παράγει μόνο την πρώτη κατάσταση s και ακολούθως το δεύτερο δίκτυο να εκπαιδευτεί με στόχο την παραγωγή-πρόβλεψη της επόμενης κατάστασης s', παίρνοντας ως είσοδο την τρέχουσα κατάσταση και μία ενέργεια. Ωστόσο, επιλέξαμε την πρώτη μέθοδο καθώς αποδείχθηκε πιο αποτελεσματικό να παράγονται και οι δύο καταστάσεις χρησιμοποιώντας το δίκτυο διάχυσης και έπειτα να προβλέπεται η ενέργεια που εκτελέστηκε μεταξύ τους (ως πρόβλημα ταξινόμησης), παρά να παράγεται αρχικά μία κατάσταση και το δεύτερο μοντέλο να εκπαιδευτεί ώστε να προβλέπει την επόμενη κατάσταση (εικόνα) με βάση μία συγκεκριμένη ενέργεια.

Όσον αφορά στην ανταμοιβή, δημιουργούνται δύο διαφορετικά σύνολα δεδομένων αποτελούμενα από καταστάσεις και ενέργειες που οδήγησαν είτε σε θετικές είτε σε αρνητικές ανταμοιβές. Στα περισσότερα περιβάλλοντα, η σημαντική πλειοψηφία των ενεργειών δεν οδηγεί σε άμεση ανταμοιβή επιβραδύνοντας τη διαδικασία της εκπαίδευσης. Υπό αυτό το πρίσμα, τα συνθετικά δείγματα μπορούν να προσφέρουν μεγαλύτερη ποικιλομορφία σε περιπτώσεις στις οποίες η άμεση ανταμοιβή είναι μη μηδενική και να βελτιώσουν την ικανότητα γενίκευσης του μοντέλου. Για αυτόν τον σκοπό, δύο ξεχωριστά δίκτυα διάχυσης έχουν εκπαιδευτεί (στα αντίστοιχα σύνολα δεδομένων) για την παραγωγή ζευγών καταστάσεων με υψηλές και χαμηλές άμεσες ανταμοιβές. Έτσι, για κάθε νέο συνθετικό δείγμα η ανταμοιβή είναι γνωστή εκ των προτέρων, ανάλογα με το μοντέλο που χρησιμοποιείται για τη δημιουργία των καταστάσεων και ένα πλήρες δείγμα της μορφής (s, a, r, s') μπορεί να παραχθεί.

6.3.1 Γεννητικά Μοντέλα

Για τη δημιουργία των συνθετικών καταστάσεων s και s' κάθε δείγματος, ένα μοντέλο διάχυσης εκπαιδεύεται σε ένα σύνολο δεδομένων αποτελούμενο από πραγματικές καταστάσεις που συλλέγονται από το περιβάλλον ALE. Το μοντέλο ακολουθεί τη βασική αρχιτεκτονική της υλοποίησης που παρουσιάζεται στο [44]. Όπως προαναφέρθηκε, κάθε κατάσταση στο περιβάλλον που χρησιμοποιούμε αποτελείται από τέσσερις εικόνες σε κλίμακα του γκρι. Καθώς τα τρία τελευταία καρέ της αρχικής κατάστασης επικαλύπτονται με τα τρία πρώτα της επόμενης, εφόσον παραχθεί η αρχική κατάσταση από το μοντέλο, για τη σύνθεση της επόμενης απαιτείται μόνο το τελευταίο καρέ καθώς τα υπόλοιπα μπορούν να ληφθούν απευθείας από την προηγούμενη κατάσταση. Ως εκ τούτου, η είσοδος (και η έξοδος) του δικτύου είναι μια εικόνα 84×84 εικονοστοιχείων με πέντε κανάλια, τα οποία στη συνέχεια χρησιμοποιούνται για να αναπαραστήσουν τις δύο καταστάσεις, όπως φαίνεται στο Σχήμα 6.1.

Προκειμένου να προσαρμοστεί και να μπορεί να λειτουργήσει το δίκτυο για τις συγκεκριμένες διαστάσεις των εικόνων, η αρχιτεκτονική του τροποποιείται ελαφρώς ώστε να αποτελείται από δύο μπλοκ κωδικοποιητών και δύο μπλοκ αποκωδικοποιητών (χωρίς να συμπεριλαμβάνεται το μεσαίο μπλοκ). Έτσι, η εικόνα υποβάλλεται σε υποδειγματοληψία δύο φορές (καταλήγοντας να έχει διαστάσεις $21 \times 21 \times$ πλήθος_συνελικτικών_φίλτρων) με αποτέλεσμα να είναι εφικτή η ανακατασκευή της στο αρχικό μέγεθος. Η κλασική αρχιτεκτονική περιλαμβάνει τρία μπλοκ κωδικοποίησης, τα οποία στη συγκεκριμένη περίπτωση (για τις προαναφερθείσες διαστάσεις των εικόνων) θα καθιστούσαν αδύνατη την παραγωγή μίας εικόνας με τις επιθυμητές διαστάσεις στην έξοδο.

Όπως έχει ήδη συζητηθεί, δύο διαφορετικά δίκτυα διάχυσης εκπαιδεύονται για την παραγωγή δειγμάτων με υψηλές και χαμηλές ανταμοιβές. Στο περιβάλλον ALE που χρησιμοποιείται στην παρούσα έρευνα, όλες οι ανταμοιβές μετά τα βήματα προεπεξεργασίας αντιστοιχίζονται σε μία από τις τιμές του συνόλου $\{-1, 0, 1\}$. Για την εκπαίδευση των μοντέλων δημιουργήθηκαν δύο σύνολα δεδομένων αποτελούμενα από 2000 δείγματα, που λαμβάνονται από την αλληλεπίδραση του πράκτορα με το περιβάλλον. Το πρώτο συνόλο δεδομένων περιλαμβάνει μόνο ζεύγη καταστάσεων (s, s') για τα οποία η άμεση ανταμοιβή ήταν 1, ενώ αντίστοιχα το δεύτερο περιλαμβάνει ζεύγη για τα οποία η άμεση ανταμοιβή ήταν -1. Αρχικά, όλες οι εικόνες κανονικοποιούνται στο διάστημα $[-1, 1]$ και τα δίκτυα εκπαιδεύονται για 140 εποχές με χρήση του βελτιστοποιητή Adam και ρυθμό μάθησης 2×10^{-4}.

6.3.2 Μοντέλο Αντίστροφης Δυναμικής

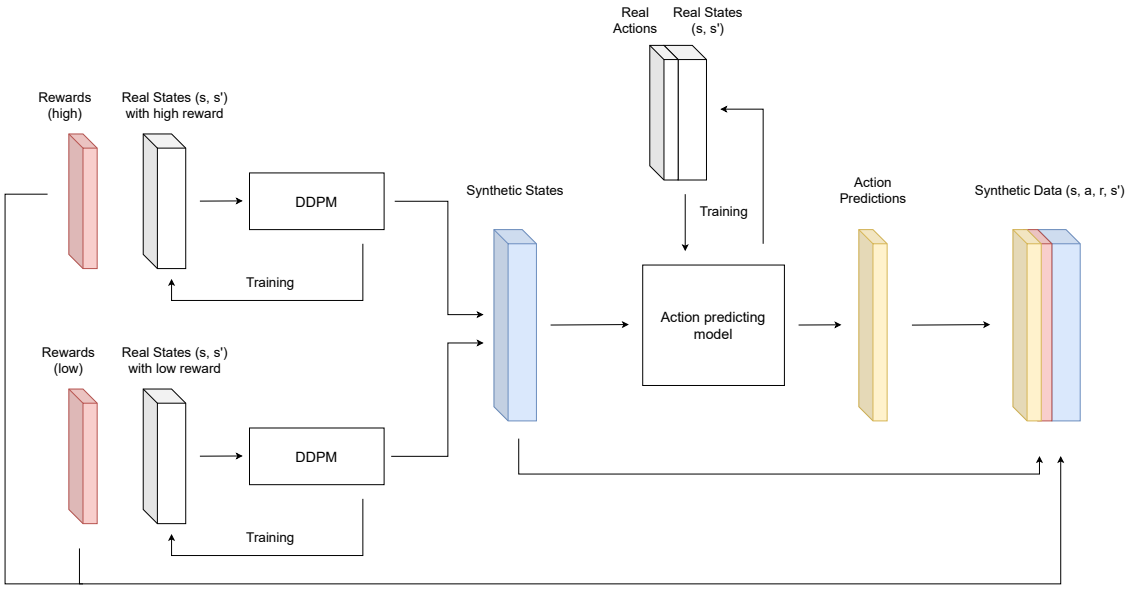
Για την πρόβλεψη των ενεργειών που εκτελούνται μεταξύ δύο καταστάσεων, χρησιμοποιείται ένα μοντέλο βασισμένο σε συνελικτικά επίπεδα με επιπλέον *υπολειμματικές συνδέσεις* (residual connec-

tions). Η είσοδος του μοντέλου είναι μία εικόνα διαστάσεων $84 \times 84 \times 5$ η οποία αναπαριστά δύο διαδοχικές καταστάσεις ενός παιχνιδιού όπως περιγράφεται παραπάνω. Αρχικά η εικόνα διέρχεται από ένα συνελικτικό επίπεδο, το οποίο ακολουθείται από ένα επίπεδο *κανονικοποίησης παρτίδας* (batch normalization) και τη συνάρτηση ενεργοποίησης ReLU. Ακολουθούν τρία υπολειμματικά μπλοκ, καθένα από τα οποία αποτελείται δύο μικρότερα μπλοκ που περιλαμβάνουν ένα διαχωρίσιμο συνελικτικό επίπεδο και ένα επίπεδο κανονικοποίησης παρτίδας. Μετά το δεύτερο επίπεδο κανονικοποίησης παρτίδας εκτελείται *συμψηφισμός μεγίστου* (max pooling). Η αρχική είσοδος τροφοδοτείται επίσης σε ένα συνελικτικό επίπεδο πριν προστεθεί στην έξοδο του υπολειμματικού μπλοκ. Άλλο ένα διαχωρίσιμο συνελικτικό επίπεδο που ακολουθείται από κανονικοποίηση παρτίδας, συνάρτηση ενεργοποίησης ReLU, *καθολικό συμψηφισμό μέσου* (global average pooling) και *απόσυρση* (dropout) χρησιμοποιείται μετά τα υπολειμματικά μπλοκ. Το επίπεδο εξόδου περιλαμβάνει τη συνάρτηση ενεργοποίησης softmax, με πλήθος εξόδων ανάλογο με τον χώρο ενεργειών του κάθε παιχνιδιού. Μία λεπτομερής περιγραφή των επιπέδων και των υπερπαραμέτρων του δικτύου παρέχεται στο Παράρτημα Α′ (Πίνακας Α′.1).

Για την εκπαίδευση του μοντέλου χρησιμοποιείται ο βελτιστοποιητής Adam με ρυθμό μάθησης 10^{-5} και *κατηγορική σταυροειδή εντροπία* (categorical cross entropy) ως συνάρτηση κόστους. Όσον αφορά στο σύνολο δεδομένων που χρησιμοποιήθηκε, αποτελείται από 100.000 δείγματα της μορφής ([τρέχουσα_κατάσταση, επόμενη_κατάσταση], ενέργεια) για κάθε παιχνίδι, τα οποία λήφθηκαν από την αλληλεπίδραση ενός τυχαίου πράκτορα με το αντίστοιχο περιβάλλον.

6.3.3 Σύνθεση Δεδομένων

Η διαδικασία παραγωγής πλήρων συνθετικών δειγμάτων της μορφής (s, a, r, s') απεικονίζεται στο Σχήμα 6.2. Δύο δίκτυα διάχυσης εκπαιδεύονται σε δείγματα ζευγών διαδοχικών καταστάσεων παιχνιδιού προκειμένου να μπορούν να παράγουν μια νέα κατάσταση s και την επόμενη κατάσταση s'. Ο στόχος είναι να δημιουργηθούν δείγματα με είτε πολύ υψηλές είτε πολύ χαμηλές ανταμοιβές, καθώς το



Σχήμα 6.2: Παραγωγή συνθετικών δεδομένων

αρχικό σύνολο δεδομένων που δημιουργείται από τις παρατηρήσεις του πράκτορα έχει μεγάλη ανισορροπία, δεδομένου ότι η άμεση ανταμοιβή στις περισσότερες περιπτώσεις είναι μηδενική. Τα δεδομένα εκπαίδευσης για το πρώτο μοντέλο αποτελούνται από ζεύγη (s, s') με υψηλή άμεση ανταμοιβή, ενώ για το δεύτερο περιέχουν ζεύγη καταστάσεων με χαμηλή ανταμοιβή. Καθώς κάθε μοντέλο εκπαιδεύεται σε συγκεκριμένο τύπο δεδομένων, μπορούμε να υποθέσουμε με σχετική ασφάλεια ότι οι καταστάσεις που δημιουργούνται από το πρώτο μοντέλο αντιστοιχούν σε υψηλές ανταμοιβές ενώ οι καταστάσεις που δημιουργούνται από το δεύτερο μοντέλο αντιστοιχούν σε χαμηλές ανταμοιβές. Με αυτόν τον τρόπο η ανταμοιβή για κάθε συνθετικό δείγμα θεωρείται γνωστή από τη στιγμή της παραγωγής του.

Εφόσον η ενέργεια που εκτελείται από τον πράκτορα δε λαμβάνεται υπόψη κατά την εκπαίδευση των δικτύων διάχυσης, κάθε συνθετικό ζεύγος (s, s') θα μπορούσε να είναι το αποτέλεσμα διαφορετικών ενεργειών. Για την παραγωγή λειτουργικών δειγμάτων που να μπορούν να χρησιμοποιηθούν για την εκπαίδευση του πράκτορα, ένα ξεχωριστό μοντέλο αντίστροφης δυναμικής εκπαιδεύτηκε για την πρόβλεψη της ενέργειας μεταξύ δύο διαδοχικών καταστάσεων. Έτσι για κάθε ζεύγος (s, s') συνθετικών καταστάσεων, η ενέργεια προβλέπεται από αυτό το μοντέλο και η ανταμοιβή ορίζεται αμέσως είτε σε 1 είτε σε -1 ανάλογα με το δίκτυο από το οποίο παράχθηκαν οι συνθετικές καταστάσεις (βλ. 6.3.1), με αποτέλεσμα τη σύνθεση ενός ολοκληρωμένου δείγματος εκπαίδευσης. Αυτή η διαδικασία παρουσιάζεται βήμα προς βήμα στον Αλγόριθμο 6.

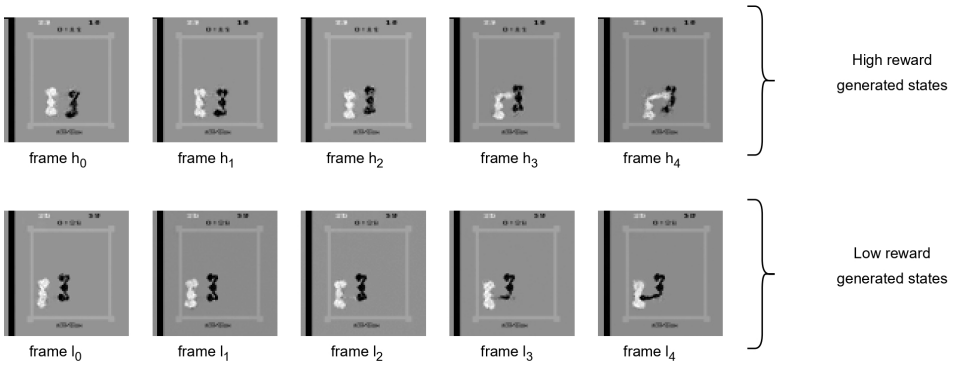
Algorithm 6: Synthetic data generation

Function `GenerateSyntheticData()`:
  $action\_predictor\_buffer, ddpm\_buffer\_high, ddpm\_buffer\_low \leftarrow [], [], []$ **while** $action\_predictor\_buffer.size() < max\_size$ **do**
    $a \leftarrow agent.select\_action(s)$
    $s', r \leftarrow env.execute(a)$
    $action\_predictor\_buffer.append(s, a, r, s')$
    if $r > 0$ **then**
      $ddpm\_buffer\_high.append(s, a, r, s')$
    if $r < 0$ **then**
      $ddpm\_buffer\_low.append(s, a, r, s')$
    $s \leftarrow s'$
  $action\_predictor.train()$
  $ddpm\_high.train()$
  $ddpm\_low.train()$
  $\{s_h, s'_h\} \leftarrow ddpm\_high.generate\_states()$
  $\{s_l, s'_l\} \leftarrow ddpm\_low.generate\_states()$
  $\{a_{pred}\} \leftarrow action\_predictor.predict(\{s_h, s'_h\}, \{s_l, s'_l\})$
  $synthetic\_data.append(\{s_h, a_{pred}, r, s'_h\})$
  $synthetic\_data.append(\{s_l, a_{pred}, r, s'_l\})$
  return $synthetic\_data$

Δείγματα που δημιουργήθηκαν με τη μεθοδολογία που περιγράφεται παραπάνω για το περιβάλλον Boxing του ALE φαίνονται στο Σχήμα 6.3. Συγκεκριμένα, οι εικόνες h_0 - h_4 δημιουργήθηκαν με το μοντέλο διάχυσης που έχει εκπαιδευτεί σε καταστάσεις που οδηγούν σε υψηλές ανταμοιβές,

ενώ οι εικόνες l_0 - l_4 παράχθηκαν από το δεύτερο μοντέλο και αντιπροσωπεύουν δύο διαδοχικές καταστάσεις με χαμηλή άμεση ανταμοιβή. Οι συνθετικές εικόνες τροφοδοτούνται στη συνέχεια στο μοντέλο πρόβλεψης ενέργειας προκειμένου να συνδυαστούν με την πιθανότερη σχετική ενέργεια και να δημιουργηθεί ένα ολοκληρωμένο δείγμα εκπαίδευσης.

Όπως φαίνεται, οι συνθετικές εικόνες που δημιουργούνται είναι αρκετά παρόμοιες με τις πραγματικές όσον αφορά στην ποιότητα. Τα γεννητικά μοντέλα παράγουν επιτυχώς διαδοχικά καρέ καταστάσεων διατηρώντας τα γενικότερα χαρακτηριστικά της εικόνας. Ιδιαίτερα, τα δύο μοντέλα μπορούν να δημιουργήσουν καταστάσεις με λογική συνέχεια, με την έννοια ότι η μετάβαση μεταξύ δύο διαδοχικών εικόνων είναι ομαλή αλλά και να αναπαράγουν τα πιο κρίσιμα μοτίβα που οδηγούν είτε σε θετικές είτε σε αρνητικές ανταμοιβές. Με αυτόν τον τρόπο, οι παραγόμενες καταστάσεις είναι λειτουργικές και δεν υπάρχει κίνδυνος να εκτροχιαστεί η διαδικασία εκπαίδευσης με εισαγωγή μη σχετικών εικόνων στο σύνολο δεδομένων.



Σχήμα 6.3: Συνθετικά δείγματα για το περιβάλλον Boxing του ALE

6.3.4 Ενισχυτική Μάθηση με Συνθετικά Δεδομένα

Προκειμένου να βελτιωθεί η εκπαίδευση του πράκτορα και να επιταχυνθεί η διαδικασία εκμάθησης, τα συνθετικά δεδομένα συνδυάζονται με πραγματικά δείγματα σε επίπεδο παρτίδας. Συγκεκριμένα, τα δεδομένα που δημιουργούνται αποθηκεύονται σε ξεχωριστό τμήμα της προσωρινής μνήμης (buffer) και στη συνέχεια λαμβάνονται δείγματα τόσο από τον πραγματικό όσο και από τον συνθετικό buffer για να διασφαλιστεί ότι κάθε παρτίδα περιέχει δείγματα και από τις δύο κατηγορίες. Με αυτόν τον τρόπο, υπάρχει πλήρης έλεγχος στη διαδικασία εκπαίδευσης καθώς το ακριβές ποσοστό κάθε κατηγορίας σε κάθε παρτίδα μπορεί να προσδιοριστεί ρητά, σε αντίθεση με την τυχαία επιλογή δειγμάτων από έναν κοινό buffer. Επομένως, είναι δυνατόν να βελτιστοποιηθεί περαιτέρω η εκπαίδευση επιλέγοντας την καταλληλότερη αναλογία (βλ. Ενότητα 6.4).

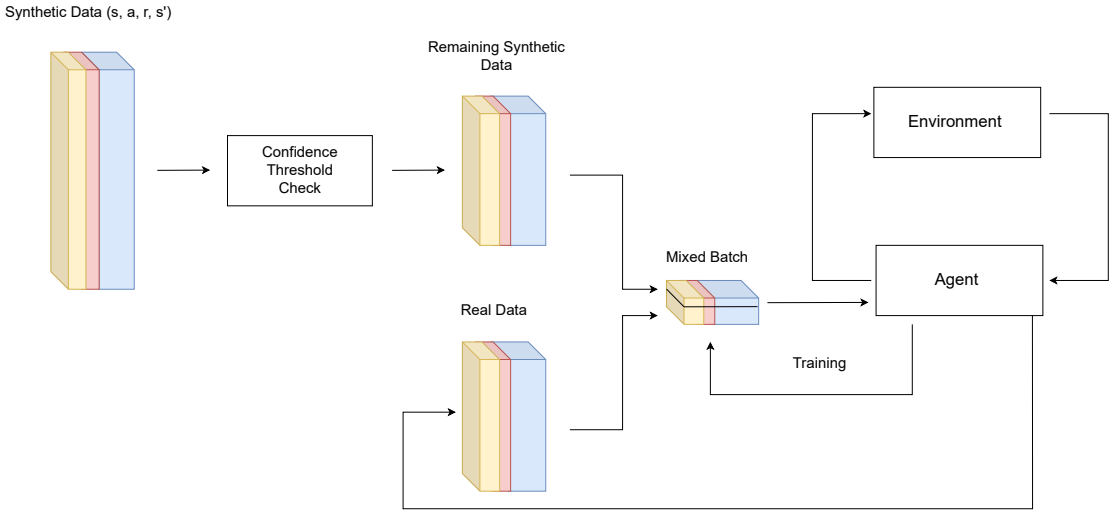
Ένα πολύ σημαντικό τμήμα της προτεινόμενης μεθοδολογίας είναι το μοντέλο πρόβλεψης ενέργειας. Είναι σαφές ότι η ακρίβεια του μοντέλου μπορεί να επηρεάσει σε μεγάλο βαθμό τη συνολική απόδοση του πράκτορα, καθώς επηρεάζει άμεσα το σύνολο των δεδομένων εκπαίδευσης. Η εσφαλμένη πρόβλεψη ενεργειών σε ακολουθίες καταστάσεων συνεπάγεται δείγματα που δεν ανταποκρίνονται στις φυσικές ιδιότητες του περιβάλλοντος και κατ΄ επέκταση μπορεί να οδηγήσει το δίκτυο του πράκτορα σε εκμάθηση λανθασμένων πολιτικών. Καθώς το πρόβλημα της πρόβλεψης της ενέργειας που εκτελείται μεταξύ δύο καταστάσεων μπορεί να είναι αρκετά περίπλοκο και εξαρτάται από το περιβάλλον εκπαίδευσης και τον χώρο δράσης, προτείνεται η χρήση ή μη των συνθετικών δειγμάτων, ανάλογα με

τον βαθμό εμπιστοσύνης των προβλέψεων του μοντέλου αντίστροφης δυναμικής. Για τον σκοπό αυτό, η πιθανότητα που παράγεται από το μοντέλο για την προβλεπόμενη ενέργεια (δηλαδή η μεγαλύτερη από τις πιθανότητες της εξόδου του μοντέλου) λειτουργεί ως βαθμός εμπιστοσύνης και ορίζεται επιπλέον ένα κάτω όριο αναφορικά με τη χρήση του κάθε δείγματος στην εκπαίδευση του πράκτορα. Τα δείγματα στα οποία η πιθανότητα της ενέργειας που προβλέπει το μοντέλο είναι υψηλότερη από το προαναφερθέν *κατώφλι εμπιστοσύνης* (confidence threshold) αποθηκεύονται στην προσωρινή μνήμη συνθετικών δεδομένων, ενώ τα υπόλοιπα απορρίπτονται. Έτσι, μειώνεται ο αριθμός των λανθασμένων συνθετικών δειγμάτων και βελτιώνεται η ποιότητα των δεδομένων εκπαίδευσης.

Η διαδικασία εκπαίδευσης του προτεινόμενου πράκτορα, *Deceiver*, απεικονίζεται στο Σχήμα 6.4 και εξηγείται στον Αλγόριθμο 7. Τόσο το κατώφλι εμπιστοσύνης όσο και το *συνθετικό βάρος* (synthetic weight - sw), δηλαδή η αναλογία των συνθετικών δειγμάτων στην παρτίδα αντιμετωπίζονται ως υπερπαράμετροι του συστήματος (γραμμές 1, 6). Όπως φαίνεται, η διαδικασία παραγωγής συνθετικών δεδομένων είναι ανεξάρτητη από τον αλγόριθμο εκπαίδευσης, που σημαίνει ότι μπορεί να εφαρμοστεί σε διαφορετικές αρχιτεκτονικές δικτύων πρακτόρων. Στην παρούσα έρευνα, η προτεινόμενη μεθοδολογία ενσωματώνεται στον κλασικό αλγόριθμο DQN, συνδυάζοντας συνθετικά δείγματα με πραγματικά δεδομένα που λαμβάνονται από τα περιβάλλοντα των παιχνιδιών.

Algorithm 7: Training on augmented data

```
1  Function PreprocessData(confidence_thres):
2      for (s_i, a_i, r_i, s'_i) in synthetic_buffer do
3          if prob(a_i) < confidence_thres then
4              synthetic_buffer.remove(s_i, a_i, r_i, s'_i)
5      return synthetic_buffer
6  Function TrainAgent(sw):
7      real_samples ← real_buffer.sample_batch(batch_size × (1 − sw))
8      synthetic_samples ← synthetic_buffer.sample_batch(batch_size × sw)
9      batch ← concatenate([real_samples, synthetic_samples]).shuffle()
10     agent.train_on_batch()
11 Function ActAndTrain():
12     PreprocessData()
13     while timesteps < termination_thres do
14         a ← agent.select_action(s)
15         s', r ← env.execute(a)
16         real_buffer.append(s, a, r, s')
17         if timesteps%train_freq = 0 then
18             TrainAgent()
```



Σχήμα 6.4: Εκπαίδευση του πράκτορα σε συνθετικά και πραγματικά δείγματα

6.4 Πειραματική Διαδικασία

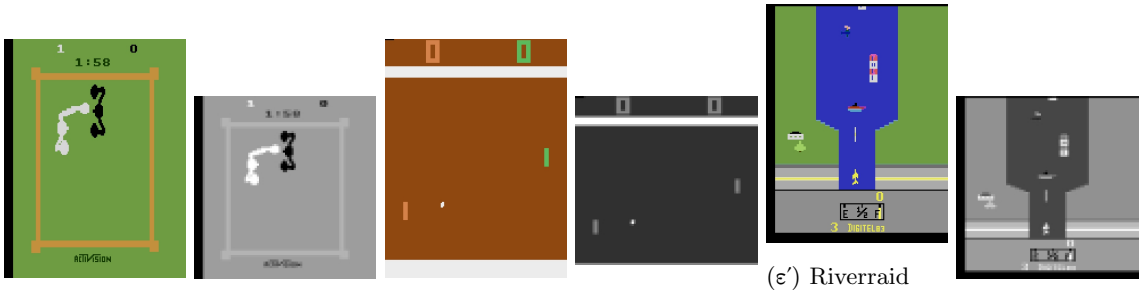
6.4.1 Ρύθμιση Περιβάλλοντος και Υπερπαραμέτρων

Ο προτεινόμενος αλγόριθμος υλοποιείται με χρήση της γλώσσας προγραμματισμού Python. Συγκεκριμένα, για την ανάπτυξη και την εκπαίδευση των μοντέλων μηχανικής μάθησης χρησιμοποιείται η βιβλιοθήκη TensorFlow [1] ενώ η αλληλεπίδραση του πράκτορα με το περιβάλλον γίνεται μέσω του περιβάλλοντος Gym Atari. Η δημιουργία των συνθετικών δεδομένων και η εκπαίδευση του πράκτορα αντιμετωπίζονται ως ανεξάρτητες διαδικασίες και συνδυάζονται ώστε να προκύψει η τελική μεθοδολογία.

Το δίκτυο του πράκτορα, αποτελείται από τρία συνελικτικά επίπεδα, ακολουθούμενα από ένα πλήρως συνδεδεμένο επίπεδο πριν το επίπεδο εξόδου. Το μέγεθος της προσωρινής μνήμης όπου αποθηκεύονται τα πραγματικά δείγματα είναι 100.000, ενώ το τμήμα που περιέχει συνθετικά δείγματα έχει μέγιστη χωρητικότητα 20.000 δειγμάτων. Λεπτομερείς πληροφορίες σχετικά με την αρχιτεκτονική και τις υπερπαραμέτρους του μοντέλου του πράκτορα παρέχονται στους Πίνακες Α΄.2 και Α΄.3. Όσον αφορά στα πειράματα, πραγματοποιούνται σε τρία ετερογενή παιχνίδια του περιβάλλοντος:

- **Boxing:** Ο πράκτορας ελέγχει έναν πυγμάχο σε ένα ρινγκ μάχης και παίρνει θετική ανταμοιβή για κάθε χτύπημα στον αντίπαλο του, ενώ αρνητική ανταμοιβή λαμβάνεται για κάθε χτύπημα που δέχεται. Υπάρχουν δεκαοχτώ πιθανές ενέργειες σε αυτό το παιχνίδι.
- **Pong:** Το κλασικό ηλεκτρονικό παιχνίδι επιτραπέζιας αντισφαίρισης, στο οποίο ο πράκτορας ελέγχει ένα κουπί και προσπαθεί να αποκρούσει τη μπάλα μακριά από το δικό του τέρμα και να τη στείλει στο τέρμα του αντιπάλου του. Η εκάστοτε ανταμοιβή είναι αντίστοιχη του σκορ του παιχνιδιού. Σε αυτό το παιχνίδι, το μέγεθος του χώρου ενεργειών είναι έξι.
- **Riverraid:** ο πράκτορας ελέγχει ένα σκάφος πάνω σε ένα ποτάμι με στόχο να αποφύγει ή να καταστρέψει εχθρικά αντικείμενα. Το μέγεθος του χώρου ενεργειών είναι δεκαοχτώ. Μόνο θετικές ανταμοιβές είναι διαθέσιμες σε αυτό το παιχνίδι.

Οι καταστάσεις που τροφοδοτούνται ως είσοδος στα δίκτυα επαύξησης και στο δίκτυο του πράκτορα υπόκεινται αρχικά σε ορισμένα βήματα προεπεξεργασίας. Αρχικά, κάθε εικόνα μειώνεται σε 84×84 εικονοστοιχεία και "μεταφέρεται" στην κλίμακα του γκρι. Επιπλέον, παρακάμπτονται ορισμένες εικόνες καταστάσεων καθώς η διαφορά μεταξύ δύο διαδοχικών καρέ είναι πολύ μικρή. Όπως προαναφέρθηκε, τέσσερις συνεχόμενες εικόνες λειτουργούν ως αναπαράσταση μίας κατάστασης (βλ. Ενότητα 6.2) ώστε να μη χάνεται η πληροφορία της κίνησης. Στην πραγματικότητα, προκειμένου να υπάρχει ορατή διαφορά, μόνο κάθε τέταρτο καρέ λαμβάνεται υπόψη ενώ τα ενδιάμεσα απορρίπτονται με αποτέλεσμα τα τέσσερα τελικά καρέ της κατάστασης να είναι ουσιαστικά το πρώτο, το πέμπτο, το ένατο και το δέκατο τρίτο κατά σειρά. Τέλος, όλες οι ανταμοιβές αντιστοιχίζονται στις τιμές του συνόλου $\{-1, 0, 1\}$. Ένα παράδειγμα της εικόνας-κατάστασης πριν και μετά την εφαρμογή των βημάτων προεπεξεργασίας για κάθε ένα από τα εξεταζόμενα παιχνίδια απεικονίζεται στο Σχήμα 6.5.



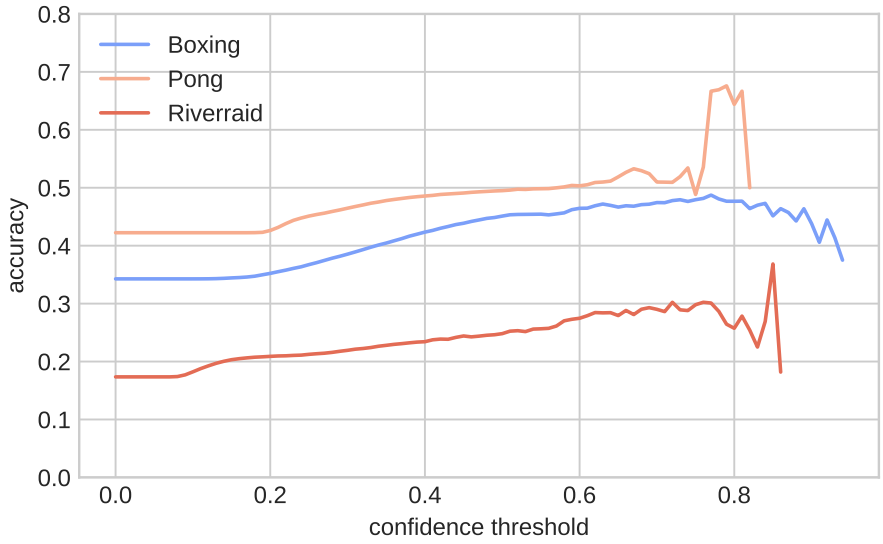
(α′) Boxing χωρίς προεπεξεργασία (β′) Boxing με προεπεξεργασία (γ′) Pong χωρίς προεπεξεργασία (δ′) Pong με προεπεξεργασία (ε′) Riverraid χωρίς προεπεξεργασία (ϛ′) Riverraid με προεπεξεργασία

Σχήμα 6.5: Εικόνες καταστάσεων πριν και μετά την προεπεξεργασία

Όσον αφορά στο κατώφλι εμπιστοσύνης των ενεργειών που πρόβλεπονται από το μοντέλο αντίστροφης δυναμικής και στο βάρος κάθε κατηγορίας δεδομένων (πραγματικά/συνθετικά) ανά παρτίδα, δοκιμάστηκαν διάφορες τιμές. Στο Σχήμα 6.6 απεικονίζεται η ακρίβεια του μοντέλου πρόβλεψης ενέργειας ανάλογα με το κατώφλι εμπιστοσύνης για κάθε παιχνίδι. Καθώς η συνθήκη επιλογής γίνεται πιο αυστηρή, ο αριθμός των δειγμάτων που μπορούν να χρησιμοποιηθούν για την εκπαίδευση του πράκτορα μειώνεται. Αυτό εξηγεί, σε ένα βαθμό, τη μείωση της ακρίβειας του μοντέλου μετά από ένα όριο (η οποία αναμενόταν να αυξάνεται καθώς αυξάνεται το κατώφλι εμπιστοσύνης), αφού σε αυτή την περίπτωση η μέτρηση βασίζεται σε πολύ μικρό αριθμό δειγμάτων. Υπό αυτό το πρίσμα, υπάρχει ένα δίλημμα μεταξύ της ακρίβειας του μοντέλου και του πλήθους των δειγμάτων που απορρίπτονται, καθώς ένας πολύ μικρός αριθμός συνθετικών δειγμάτων μπορεί να προκαλέσει υπερπροσαρμογή και να επηρεάσει αρνητικά την εκπαίδευση του πράκτορα. Ως εκ τούτου, για κάθε παιχνίδι δοκιμάστηκαν αρκετές τιμές κατωφλίου εμπιστοσύνης για το αντίστοιχο μοντέλο πρόβλεψης ενέργειας προκειμένου να προσδιοριστεί η βέλτιστη επιλογή.

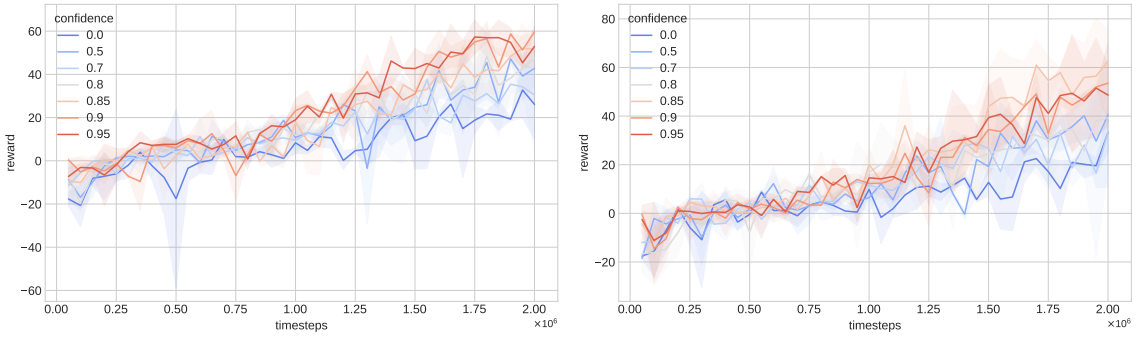
Το Σχήμα 6.7 δείχνει τη μέση ανταμοιβή που επιτυγχάνεται από τον πράκτορα στο παιχνίδι Boxing για διαφορετικές τιμές υπερπαραμέτρων. Συγκεκριμένα, το Σχήμα 6.7α′ απεικονίζει την απόδοση του πράκτορα για διαφορετικές τιμές κατωφλίου εμπιστοσύνης και συνθετικό βάρος ίσο με 0.1, ενώ το Σχήμα 6.7β′ παρουσιάζει τα αντίστοιχα αποτελέσματα για συνθετικό βάρος ίσο με 0.2. Κάθε 50.000 χρονικά βήματα υπολογίζεται η μέση ανταμοιβή από 50 ανεξάρτητες δοκιμές του πράκτορα. Τα αποτελέσματα που παρουσιάζονται αφορούν στη μέση τιμή από δύο διαφορετικές εκτελέσεις (τρεξίματα) του αλγορίθμου.

Όπως ήταν αναμενόμενο, η τιμή του κατωφλίου εμπιστοσύνης επηρεάζει σε μεγάλο βαθμό την



Σχήμα 6.6: Ακρίβεια μοντέλου πρόβλεψης ενέργειας για διαφορετικές τιμές του ορίου εμπιστοσύνης

εκπαίδευση του πράκτορα. Υψηλότερο κατώφλι έχει ως αποτέλεσμα μεγαλύτερη ακρίβεια του μοντέλου πρόβλεψης και χαμηλότερο ποσοστό λανθασμένων συνθετικών δειγμάτων. Από αυτή την άποψη, η αύξηση της απόδοσης παράλληλα με την αύξηση του κατωφλίου εμπιστοσύνης όπως προκύπτει από το Σχήμα 6.7α΄, φαίνεται φυσιολογική. Ωστόσο, όπως αναφέρθηκε παραπάνω, πολύ υψηλές τιμές κατωφλίου αποκλείουν μεγάλο αριθμό δειγμάτων από τον συνθετικό buffer και θα μπορούσαν να έχουν ως αποτέλεσμα το δίκτυο να εκπαιδεύεται στα ίδια δείγματα με υψηλή συχνότητα. Αυτό το φαινόμενο απεικονίζεται πιο έντονα στο Σχήμα 6.7β΄ όπου το συνθετικό βάρος είναι μεγαλύτερο, που σημαίνει ότι εισάγονται περισσότερα συνθετικά δείγματα σε κάθε παρτίδα. Σε αυτή την περίπτωση, μία πολύ υψηλή τιμή κατωφλίου (συγκεκριμένα 0.95) οδηγεί σε πιο αργή εκμάθηση ενώ χαμηλότερες τιμές (0.9 και 0.85) έχουν ως αποτέλεσμα την κορυφαία απόδοση του πράκτορα όσον αφορά στη μέση ανταμοιβή και στα χρονικά βήματα που απαιτούνται για την επίτευξή του. Ως εκ τούτου, γίνεται αντιληπτή η σημασία της ισορροπίας μεταξύ της ακρίβειας του μοντέλου πρόβλεψης ενέργειας και του τελικού μεγέθους του συνθετικού buffer (όπως προκύπτει μετά την εφαρμογή του περιορισμού στην εμπιστοσύνη των προβλέψεων).

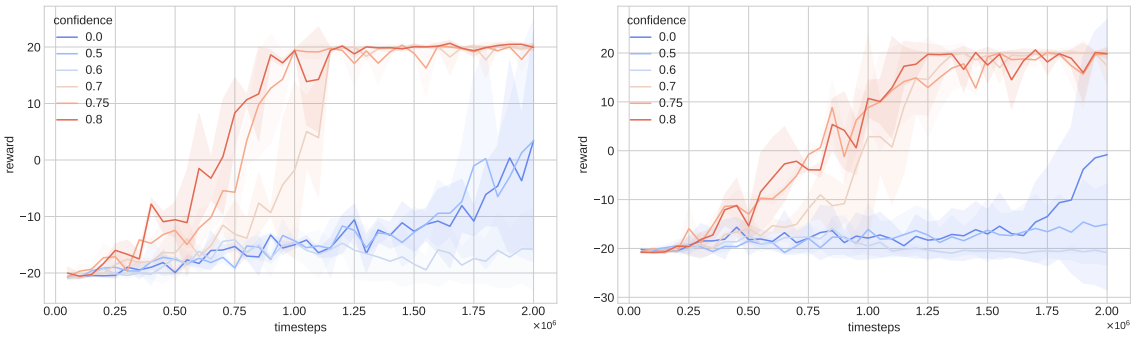


(α΄) συνθετικό βάρος δεδομένων 0.1

(β΄) συνθετικό βάρος δεδομένων 0.2

Σχήμα 6.7: Μέση ανταμοιβή του πράκτορα Deceiver (περιβάλλον Boxing)

Στην περίπτωση του Pong (Σχήμα 6.8), η επίδραση της παραμέτρου εμπιστοσύνης είναι πιο εμφανής. Για πολύ χαμηλές τιμές (έως 0.6) ο πράκτορας μαθαίνει πολύ αργά ή σε ορισμένες περιπτώσεις δε μαθαίνει καθόλου λόγω της χαμηλής ποιότητας των συνθετικών δειγμάτων, καθώς ένα υψηλό ποσοστό αυτών περιέχει λανθασμένες ενέργειες (με την έννοια ότι οι ενέργειες που προβλέπονται από το μοντέλο αντίστροφης δυναμικής δεν οδηγούν στην πραγματικότητα στις μεταβάσεις μεταξύ των καταστάσεων που δημιουργούνται από τα δίκτυα διάχυσης). Σχετικά με το μοντέλο πρόβλεψης ενέργειας, παρόλο που συνολικά επιτυγχάνει υψηλότερη ακρίβεια σε αυτό το περιβάλλον (πιθανώς λόγω του μικρότερου μεγέθους του χώρου ενεργειών), από μία τιμή κατωφλίου εμπιστοσύνης και έπειτα αποκόπτει ολόκληρο το σύνολο συνθετικών δεδομένων (π.χ. δεν υπάρχει συνθετικό δείγμα για το οποίο η πιθανότητα του μοντέλου για την επιλεγμένη ενέργεια να είναι μεγαλύτερη από 0.9, επομένως ο ορισμός του ορίου εμπιστοσύνης στην τιμή 0.9 θα είχε ως αποτέλεσμα έναν κενό συνθετικό buffer). Ως εκ τούτου, η υψηλότερη τιμή εμπιστοσύνης που λαμβάνεται υπόψη για τα πειράματα στο Pong είναι 0.8 καθώς για υψηλότερες τιμές όλα τα συνθετικά δείγματα απορρίπτονται (όπως φαίνεται στο Σχήμα 6.6).

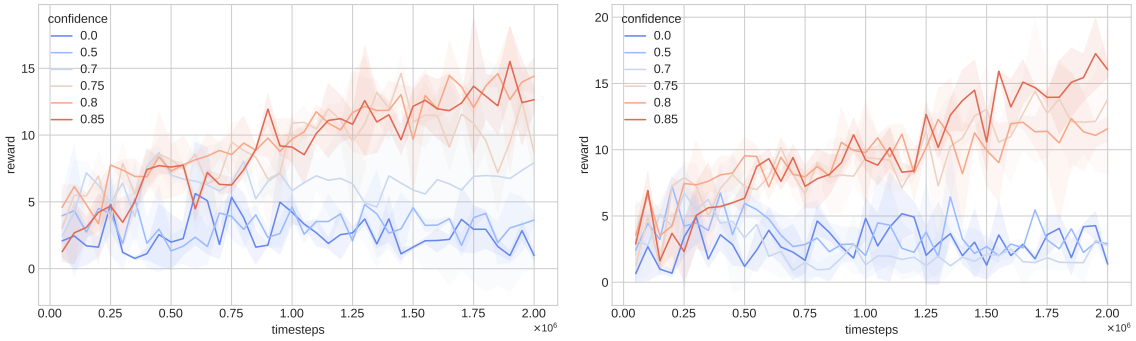


(α′) συνθετικό βάρος δεδομένων 0.1

(β′) συνθετικό βάρος δεδομένων 0.2

Σχήμα 6.8: Μέση ανταμοιβή του πράκτορα Deceiver (περιβάλλον Pong)

Παρόμοια συμπεράσματα μπορούν να εξαχθούν από την απόδοση του πράκτορα στο τρίτο παιχνίδι, το Riverraid (Σχήμα 6.9). Και σε αυτή την περίπτωση δε μπορεί να φτάσει σε υψηλές βαθμολογίες όταν ορίζονται χαμηλές τιμές ορίου εμπιστοσύνης. Το υψηλότερο όριο που επιτρέπεται από το μοντέλο σε αυτό το παιχνίδι (δηλαδή χωρίς να απορρίπτονται όλα τα συνθετικά δείγματα) είναι 0.85. Όσον αφορά στο βάρος συνθετικών δεδομένων, οι πράκτορες με την υψηλότερη τιμή (0.2) φαίνεται ότι επωφελούνται από το μεγαλύτερο ποσοστό συνθετικών δειγμάτων ανά παρτίδα και αποδίδουν καλύτερα, όπως συνέβη και στο πρώτο περιβάλλον που εξετάστηκε (Boxing).



(α′) συνθετικό βάρος δεδομένων 0.1

(β′) συνθετικό βάρος δεδομένων 0.2

Σχήμα 6.9: Μέση ανταμοιβή του πράκτορα Deceiver (περιβάλλον Riverraid)

6.4.2 Αποτελέσματα

Για την αξιολόγηση του προτεινόμενου αλγορίθμου, παρουσιάζεται για κάθε περιβάλλον η απόδοση του καλύτερου πράκτορα (όπως προκύπτει από τους διαφορετικούς συνδυασμούς των δύο βασικών υπερπαραμέτρων, βλ. Ενότητα 6.4.1) σε σύγκριση με πράκτορες στους οποίους έχουν ενσωματωθεί κλασικές τεχνικές επαύξησης δεδομένων. Συγκεκριμένα, οι τεχνικές επαύξησης που υλοποιήθηκαν και εξετάστηκαν στα τρία περιβάλλοντα είναι:

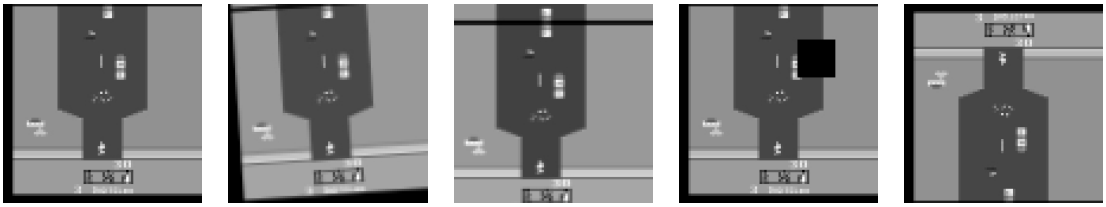
- **Περιστροφή (Rotation):** Εφαρμόζεται τυχαία περιστροφή στην εικόνα είτε δεξιόστροφα είτε αριστερόστροφα. Στα πειράματα που διενεργήθηκαν στην παρούσα έρευνα η εικόνα κάθε κατάστασης περιστρέφεται τυχαία στο εύρος $[-7°, 7°]$.

- **Μετατόπιση (Translation):** Η εικόνα μετατοπίζεται τυχαία οριζόντια και κατακόρυφα κατά w και h εικονοστοιχεία αντίστοιχα. Στην περίπτωσή μας και οι δύο τιμές (δεδομένου ότι οι εικόνες είναι τετράγωνες) είναι στο εύρος $[-0.1\times$ *πλάτος_εικόνας*$, 0.1\times$ *πλάτος_εικόνας*$]$. Τα εικονοστοιχεία που βρίσκονταν έξω από την περιοχή της εικόνας συμπληρώνονται με αντικατοπτρισμό των εικονοστοιχείων που βρίσκονται στην άκρη της εικόνας.

- **Αποκοπή (Cutout):** Επιλέγεται τυχαία ένα μικρό πλαίσιο της εικόνας και όλα τα εικονοστοιχεία του παίρνουν την τιμή 0. Το πλάτος και το ύψος του πλαισίου κυμαίνονται στο εύρος $[10, 20]$ εικονοστοιχείων.

- **Αντιστροφή (Flip):** Η εικόνα αντιστρέφεται τυχαία είτε ως προς τον κατακόρυφο άξονα, είτε ως προς τον οριζόντιο είτε ως προς και τους δύο άξονες.

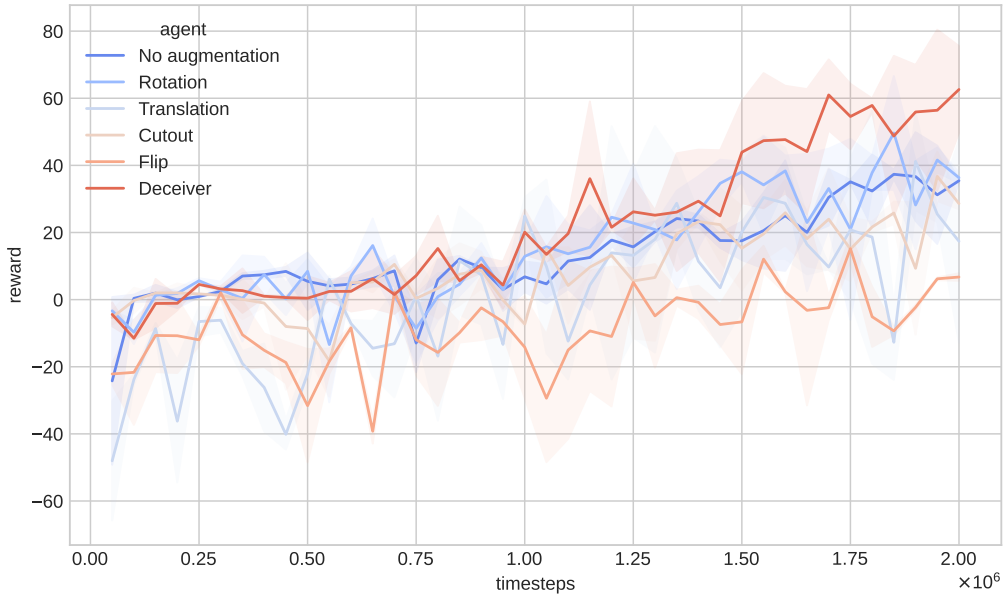
Ένα παράδειγμα μίας κατάστασης παιχνιδιού πριν και μετά την εφαρμογή κάθε τεχνικής επαύξησης στο περιβάλλον Riveraid παρουσιάζεται στο Σχήμα 6.10.



(α′) Χωρίς επαύξηση (β′) Περιστροφή (γ′) Μετατόπιση (δ′) Αποκοπή (ε′) Αντιστροφή

Σχήμα 6.10: Κατάσταση παιχνιδιού πριν και μετά την εφαρμογή τεχνικών επαύξησης

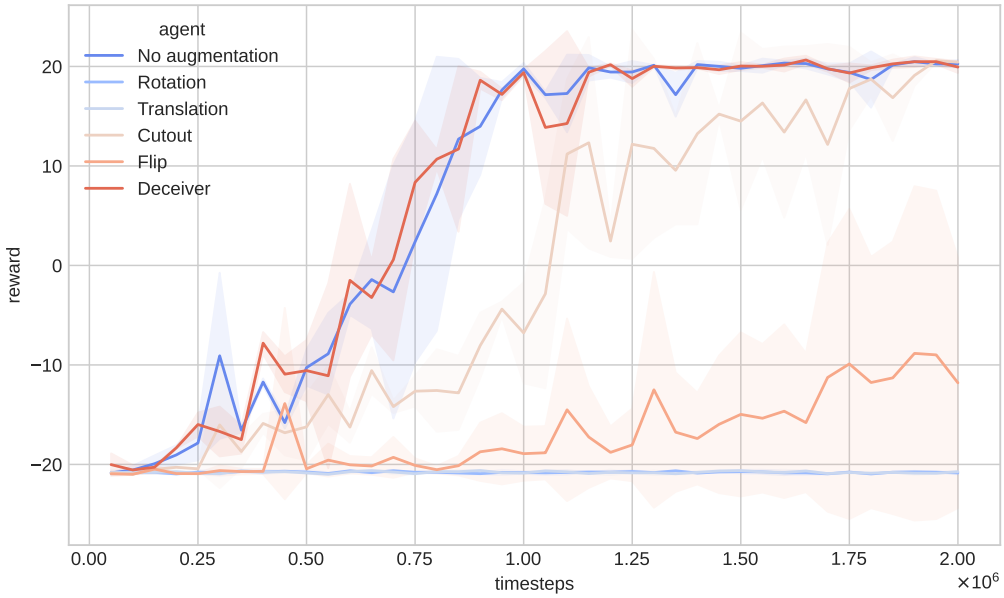
Οι εικόνες 6.11 – 6.13 απεικονίζουν την απόδοση των πρακτόρων στα περιβάλλοντα Boxing, Pong και Riverraid αντίστοιχα. Οι αλγόριθμοι που εξετάζονται είναι ο κλασικός DQN χωρίς επαύξηση δεδομένων, οι τέσσερις πράκτορες ενισχυμένοι με τυχαία περιστροφή, μετατόπιση, αποκοπή και αναστροφή όπως περιγράφονται παραπάνω και ο προτεινόμενος πράκτορας Deceiver (με τις βέλτιστες τιμές υπερπαραμέτρων για κάθε παιχνίδι όπως προκύπτουν από τα Σχήματα 6.7 – 6.9). Όπως φαίνεται, η προτεινόμενη μεθοδολογία υπερτερεί των υπόλοιπων πρακτόρων ή επιτυγχάνει συγκρίσιμα με αυτούς αποτελέσματα σε όλες τις περιπτώσεις. Συγκεκριμένα, στο παιχνίδι Boxing (Σχήμα 6.11) ο αλγόριθμος Deceiver συγκεντρώνει σημαντικά υψηλότερη ανταμοιβή τόσο από τον κλασικό αλγόριθμο, όσο και από τις παραλλαγές του με τις τέσσερις παραδοσιακές τεχνικές επαύξησης. Ιδιαίτερα στο δεύτερο μισό της εκπαίδευσης (μετά από $1M$ χρονικά βήματα), ο πράκτορας φαίνεται να επωφελείται σε μεγάλο βαθμό από τον συνδυασμό των συνθετικών και των πραγματικών δειγμάτων και παρουσιάζει ταχύτερη καμπύλη εκμάθησης.



Σχήμα 6.11: Μέση ανταμοιβή πρακτόρων (περιβάλλον Boxing)

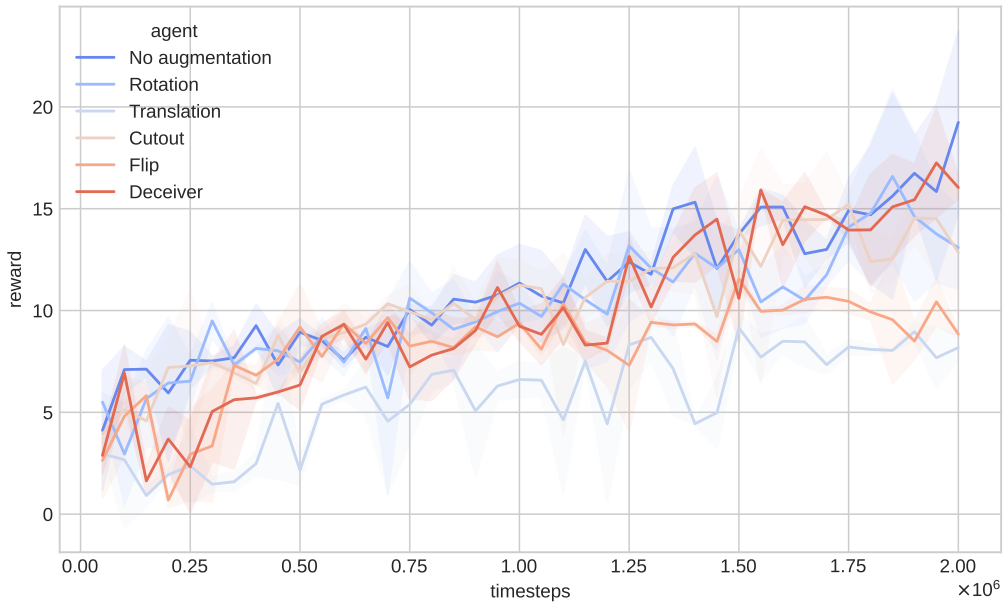
.

Το Pong είναι πιο απλό παιχνίδι, κάτι που επιβεβαιώνεται από το γεγονός ότι τρεις πράκτορες επιτυγχάνουν την υψηλότερη δυνατή ανταμοιβή (στο Σχήμα 6.12 εμφανίζονται οι μέσες τιμές των ανταμοιβών, που σημαίνει ότι εισάγεται διακύμανση στα αποτελέσματα, αλλά από ένα σημείο και έπειτα οι ανταμοιβές είναι σταθερά πολύ κοντά στη μέγιστη δυνατή βαθμολογία για αυτό το παιχνίδι). Με αυτό το δεδομένο, έχει περισσότερο νόημα η σύγκριση απόδοσης όσον αφορά στον αριθμό των χρονικών βημάτων που απαιτούνται για την επίτευξη αυτής της βαθμολογίας. Υπό αυτό το πρίσμα, ο Deceiver πλησιάζει τη μέγιστη ανταμοιβή περίπου $100k$ χρονικά βήματα νωρίτερα και φαίνεται να έχει οριακά πιο γρήγορη καμπύλη εκμάθησης κατά τη διάρκεια της εκπαίδευσης (μέχρι το σημείο που όλοι οι πράκτορες φτάνουν τελικά στη μέγιστη δυνατή βαθμολογία).



Σχήμα 6.12: Μέση ανταμοιβή πρακτόρων (περιβάλλον Pong)

Στο τρίτο παιχνίδι (Σχήμα 6.13), ο προτεινόμενος αλγόριθμος δε φαίνεται να υπερτερεί των υπόλοιπων τεχνικών. Παρόλα αυτά, έχει παρόμοια απόδοση με τις πιο αποτελεσματικές (στο συγκεκριμένο περιβάλλον) παραλλαγές που δοκιμάστηκαν. Αυτό μπορεί να αποδοθεί στην ακρίβεια του μοντέλου πρόβλεψης ενεργειών, η οποία είναι χαμηλότερη στην περίπτωση του Riverraid. Όπως απεικονίζεται στα Σχήματα 6.7 – 6.9, η τιμή κατωφλίου εμπιστοσύνης και κατά συνέπεια η ακρίβεια του μοντέλου έχει μεγάλο αντίκτυπο στη διαδικασία της εκπαίδευσης. Επιπλέον, μία ιδιαιτερότητα αυτού του παιχνιδιού που πιθανώς οδήγησε σε αυτή τη συμπεριφορά είναι ότι δεν υπάρχουν αρνητικές ανταμοιβές. Για αυτόν τον λόγο, σε αυτή την περίπτωση χρησιμοποιήθηκε μόνο ένα μοντέλο διάχυσης για τη δημιουργία συνθετικών καταστάσεων με υψηλή ανταμοιβή, γεγονός που εν μέρει εξηγεί την αδυναμία σημαντικής αύξησης στην απόδοση του Deceiver σε αυτό το περιβάλλον.



Σχήμα 6.13: Μέση ανταμοιβή πρακτόρων (περιβάλλον Riverraid)

Στον Πίνακα 6.1 παρουσιάζονται τα συνολικά αποτελέσματα (μέγιστη ανταμοιβή, χρονικά βήματα, ακρίβεια του μοντέλου αντίστροφης δυναμικής) των πρακτόρων ανά παιχνίδι. Όσον αφορά στον Deceiver παρατίθεται η καλύτερη παραλλαγή του για κάθε μία από τις τιμές συνθετικού βάρους (0.1 και 0.2). Όπως έχει ήδη αναφερθεί, η βέλτιστη τιμή εμπιστοσύνης δεν είναι απαραίτητα η ίδια για τις διαφορετικές τιμές του συνθετικού βάρους, γεγονός που εξηγεί τη διαφορά στην ακρίβεια του μοντέλου πρόβλεψης ενεργειών στις δύο περιπτώσεις για το πρώτο και το τρίτο περιβάλλον.

6.5 Συζήτηση

Σε αυτή την ενότητα παραθέτουμε τις σημαντικότερες παρατηρήσεις και τα συμπεράσματα που προκύπτουν από τη διαδικασία που περιγράφεται παραπάνω. Κατ' αρχάς, τα πειραματικά αποτελέσματα επιβεβαιώνουν την αστάθεια των παραδοσιακών τεχνικών επαύξησης εικόνας στο πεδίο της ενισχυτικής μάθησης. Συγκεκριμένα η τυχαία περιστροφή, η μετατόπιση, η αποκοπή και η κατακόρυφη και οριζόντα αντιστροφή εφαρμόστηκαν για την επαύξηση των δεδομένων εκπαίδευσης και αναδείχθηκε ότι διαφορετικές μεθοδοι μπορεί να έχουν διαφορετικό αποτέλεσμα ανάλογα με το περιβάλλον, και ως εκ τούτου αδυναμία γενίκευσης όπως τονίζεται και στη σχετική βιβλιογραφία [31, 82].

Στα συγκεκριμένα παιχνίδια Atari που χρησιμοποιούνται στην παρούσα έρευνα, είναι εμφανές ότι η επίδραση των διάφορων τεχνικών στην εκπαίδευση του πράκτορα εξαρτάται σε μεγάλο βαθμό από τα χαρακτηριστικά του εκάστοτε περιβάλλοντος. Η περιστροφή και η αποκοπή είναι κατά βάση οι πιο ωφέλιμες, ωστόσο καμία μέθοδος δεν υπερτερεί των υπολοίπων σε όλες τις περιπτώσεις. Για παράδειγμα στο παιχνίδι Pong (Σχήμα 6.12), η τυχαία περιστροφή και η μετατόπιση έχουν αρνητικό αντίκτυπο στην απόδοση του πράκτορα, ο οποίος δε μπορεί να φτάσει σε θετικές ανταμοιβές. Αυτό είναι φυσιολογικό καθώς τα κουπιά (και σε ορισμένες περιπτώσεις η μπάλα) που αποτελούν τα στοιχεία της κατάστασης του παιχνιδιού με τη μεγαλύτερη πληροφορία, βρίσκονται στις άκρες της εικόνας με αποτέλεσμα μετά την εφαρμογή της περιστροφής ή της μετατόπισης να αποκόπτονται από την εικόνα.

	Max Reward	Timesteps	Predictor's Accuracy	Game
No augmentation	37.35	1.85 M	-	
Rotation	49.65	1.85 M	-	
Translation	41.2	1.9 M	-	
Cutout	36.73	1.95 M	-	**Boxing**
Flip	15.21	1.75 M	-	
Deceiver (0.1)	59.74	2.0 M	0.438	
Deceiver (0.2)	**62.61**	2.0 M	0.452	
No augmentation	20.48	1.9 M	-	
Rotation	−20.63	1.35 M	-	
Translation	−20.62	1.5 M	-	
Cutout	20.58	1.95 M	-	**Pong**
Flip	-8.84	1.9 M	-	
Deceiver (0.1)	**20.65**	1.65 M	0.644	
Deceiver (0.2)	20.62	1.7 M	0.644	
No augmentation	**19.24**	2.0 M	-	
Rotation	16.59	1.85 M	-	
Translation	9.15	1.5 M	-	
Cutout	15.21	1.75 M	-	**Riverraid**
Flip	11.56	1.5 M	-	
Deceiver (0.1)	15.51	1.9 M	0.257	
Deceiver (0.2)	17.25	1.95 M	0.368	

Πίνακας 6.1: Απόδοση πρακτόρων με διαφορετικές τεχνικές επαύξησης δεδομένων (η υψηλότερη ανταμοιβή ανά παιχνίδι παρουσιάζεται εντονότερα)

Αναφορικά με την προτεινόμενη μεθοδολογία, στη χειρότερη περίπτωση η απόδοση του πράκτορα είναι συγκρίσιμη με αυτή των αλγορίθμων που ενσωματώνουν τις κλασικές τεχνικές επαύξησης (Πίνακας 6.1), υποδεικνύοντας ότι δεν επηρέαζεται σε μεγάλο βαθμό από τα χαρακτηριστικά των εικόνων. Αυτή η συμπεριφορά μπορεί να αποδοθεί στη διαδικασία παραγωγής συνθετικών δεδομένων που ακολουθήθηκε, κατά την οποία δημιουργούνται εντελώς νέα δείγματα αντί να τροποποιούνται τα υπάρχοντα πραγματικά δεδομένα (που μπορεί να έχει ως αποτέλεσμα την απώλεια βασικών πληροφοριών).

Μία ιδιαίτερα σημαντική πτυχή του αλγορίθμου είναι το μοντέλο που προβλέπει την ενέργεια μεταξύ δύο διαδοχικών καταστάσεων. Οι περισσότερες προσεγγίσεις στη σύγχρονη βιβλιογραφία χρησιμοποιούν μοντέλα πρόσθιας δυναμικής, τα οποία προβλέπουν την επόμενη κατάσταση δεδομένης της τρέχουσας κατάστασης και μίας συγκεκριμένης ενέργειας προς εκτέλεση [35, 104, 71]. Αν και τα αναφερόμενα αποτελέσματα είναι ενθαρρυντικά, τα μοντέλα πρόσθιας δυναμικής που δοκιμάσαμε στα εξεταζόμενα παιχνίδια δεν ήταν ικανοποιητικά και ανέδειξαν τη δυσκολία του συγκεκριμένου εγχειρήματος καθώς, ειδικά σε περιπτώσεις με μεγάλο μέγεθος του χώρου ενεργειών, είναι δύσκολο να προβλεφθούν οι ακριβείς αλλαγές κατάστασης που προκαλούνται από μία συγκεκριμένη ενέργεια. Για αυτόν τον λόγο χρησιμοποιήθηκε ένα μοντέλο αντίστροφης δυναμικής που, δεδομένων δύο καταστάσεων, προβλέπει την εκτελούμενη ενέργεια όπως στα [69, 76, 75]. Αυτή η αντίστροφη προσέγγιση αποδείχθηκε αρκετά πιο απλή και συγχρόνως κατάλληλη για τη δημιουργία συνθετικών δεδομένων,

αφού τόσο οι τρέχουσες όσο και οι επόμενες καταστάσεις στην περίπτωσή μας παράγονται από τα δίκτυα διάχυσης.

Όσον αφορά στις παραμέτρους που χρησιμοποιήθηκαν, το κατώφλι εμπιστοσύνης του μοντέλου πρόβλεψης ενέργειας και η αναλογία πραγματικών και συνθετικών δειγμάτων σε κάθε παρτίδα αποδείχτηκε ότι ήταν οι πιο σημαντικές. Συγκεκριμένα, η αύξηση του ορίου εμπιστοσύνης (και κατά συνέπεια της ακρίβειας του μοντέλου) οδήγησε σε σημαντική βελτίωση της εκπαίδευσης του πράκτορα, παρόλο που η ακρίβεια του μοντέλου είναι μικρότερη από 50% σε δύο από τα τρία περιβάλλοντα που εξετάζονται. Αυτό το αποτέλεσμα αναδεικνύει τα οφέλη της προτεινόμενης μεθοδολογίας επαύξησης δεδομένων καθώς, παρά το γεγονός ότι αρκετά συνθετικά δείγματα δεν περιλαμβάνουν τη σωστή ενέργεια, η συνολική επίδραση της τεχνικής στην απόδοση του πράκτορα είναι θετική. Αυτό πιθανώς υποδηλώνει ότι οι ενέργειες που προβλέπονται από το μοντέλο αντίστροφης δυναμικής στις περιπτώσεις που είναι λανθασμένες, είναι παρόμοιες με τις ενέργειες που θα οδηγούσαν πραγματικά στην εκάστοτε μετάβαση καταστάσεων και ως εκ τούτου δεν δυσχεραίνουν την εκπαίδευση. Παρόλα αυτά, η συμβολή του δικτύου αντίστροφης δυναμικής στην αποτελεσματικότητα της εκπαίδευσης του πράκτορα είναι κομβική και η βελτίωσή του θα μπορούσε να αποτελέσει βασικό παράγοντα για την περαιτέρω ενίσχυση του προτεινόμενου αλγορίθμου. Τέλος, σχετικά με την αναλογία πραγματικών και συνθετικών δειγμάτων ανά παρτίδα, από την πειραματική διαδικασία προέκυψε ότι ένα ποσοστό συνθετικών δειγμάτων $10\% - 20\%$ είναι ιδανικό για την προσθήκη ποικιλομορφίας και την επιτάχυνση της εκπαίδευσης του πράκτορα, ενώ επιπλέον αύξησή του έχει αρνητικό αντίκτυπο στη διαδικασία.

Κεφάλαιο 7

Επίλογος

Η τεχνητή νοημοσύνη ως τμήμα της επιστήμης της πληροφορικής έχει γνωρίσει ιδιαίτερα μεγάλη ανάπτυξη τα τελευταία χρόνια, σε συνδυασμό και με τη βελτίωση του υλικού των ηλεκτρονικών υπολογιστών, καθώς πλέον παρέχεται η δυνατότητα υλοποίησης τεχνικών που μέχρι πρόσφατα δεν ήταν δυνατόν να ξεφύγουν από το θεωρητικό πλαίσιο λόγω υπολογιστικών περιορισμών. Ωστόσο, ανοιχτές παραμένουν προκλήσεις που αφορούν τόσο στην απόδοση και στη χρονική απόκριση των συστημάτων όσο και στη δυνατότητα προσαρμογής και μεταφοράς μάθησης σε διαφορετικά προβλήματα (περιβάλλοντα). Αντικείμενο αυτής της διατριβής είναι ο σχεδιασμός και η υλοποίηση πρωτότυπων αλγορίθμων τεχνητής νοημοσύνης και μηχανικής μάθησης και ο έλεγχος της αποδοτικότητάς τους, με πλαίσιο εφαρμογής την ανάπτυξη πρακτόρων σε περιβάλλοντα που προσομοιώνουν ηλεκτρονικά παιχνίδια τόσο σε οπτικό όσο και σε λειτουργικό επίπεδο. Στη συνέχεια, παρουσιάζονται συνοπτικά τα κυριότερα συμπεράσματα που προέκυψαν από την ερευνητική διαδικασία και προτείνονται ορισμένες κατευθύνσεις για πιθανές μελλοντικές επεκτάσεις των μεθόδων που παρουσιάστηκαν.

7.1 Συμπεράσματα

Στο πρώτο στάδιο, μελετήθηκαν οι γενετικοί αλγόριθμοι και πιο συγκεκριμένα η δυνατότητα ανάπτυξης ευφυών πρακτόρων που βασίζονται αμιγώς στη χρήση των εκπαιδευμένων χρωμοσωμάτων. Για αυτόν τον σκοπό, παρουσιάστηκε μία διαδικασία για την κωδικοποίηση ενεργειών ανάλογα με την κατάσταση του κόσμου σε ηλεκτρονικά παιχνίδια μέσω γενετικών αλγορίθμων. Μία ξεχωριστή κωδικοποίηση χρησιμοποιήθηκε για το σύνολο των ενεργειών κίνησης, επιτρέποντας την επαναχρησιμοποίηση τμήματος του γενότυπου σε διαφορετικό περιβάλλον. Επιπλέον, μία μεθοδολογία για κωδικοποίηση των καταστάσεων με N-πλειάδες εφαρμόστηκε προκειμένου να αξιοποιηθεί όσο το δυνατόν περισσότερη πληροφορία εντός των υπολογιστικών περιορισμών. Ο τελικός πράκτορας εφαρμόστηκε και αξιολογήθηκε σε δύο παιχνίδια του περιβάλλοντος του διαγωνισμού γενικευμένης τεχνητής νοημοσύνης για ηλεκτρονικά παιχνίδια, ξεπερνώντας το μοντέλο αναφοράς του ΓΑ κυλιόμενου ορίζοντα σε ποσοστό νίκης.

Απο τη διαδικασία εκπαίδευσης τονίστηκε η καθοριστική επίδραση της συνάρτησης ποιότητας στη συμπεριφορά του εκπαιδευμένου πράκτορα. Όπως ήταν αναμενόμενο, η συνάρτηση ποιότητας καθοδηγεί σε μεγάλο βαθμό την εκπαίδευση του ΓΑ και η περαιτέρω διερεύνηση του σχεδιασμού της θα μπορούσε να προσφέρει χρήσιμα συμπεράσματα. Η χρήση των N-πλειάδων φάνηκε ιδιαίτερα αποτελε-

σματική, υπογραμμίζοντας τη σημασία της έρευνας προς την κατεύθυνση των μεθόδων αναπαράστασης των καταστάσεων. Επιπλέον, όπως διαπιστώθηκε από τα πειράματα η επαναχρησιμοποίηση τμήματος της κωδικοποίησης του ΓΑ συνέβαλε στην επιτάχυνση της εκπαίδευσής του για το δεύτερο περιβάλλον, αναδεικνύοντας τη σημασία της μεταφοράς μάθησης στο συγκεκριμένο πεδίο.

Στη συνέχεια εξετάστηκε ο αλγόριθμος δενδρικής αναζήτησης Μόντε Κάρλο, ο οποίος αποτελεί την αιχμή της τεχνολογίας σε πολλά στοχαστικά περιβάλλοντα παιχνιδιών. Στο Κεφάλαιο 4 προτάθηκαν συγκεκριμένες βελτιώσεις του αλγορίθμου όσον αφορά στο στάδιο της προσομοίωσης. Συγκεκριμένα, δοκιμάστηκε η ενσωμάτωση ενός ταξινομητή ενίσχυσης κλίσης με στόχο την εισαγωγή γνώσης πεδίου κατά την αξιολόγηση των κόμβων. Επίσης, μελετήθηκε η ιδέα της πρώιμης προσομοίωσης στα αρχικά στάδια του παιχνιδιού προκειμένου να αξιοποιηθεί η πρόβλεψη σε κόμβους πιο βαθιά στο δέντρο αναζήτησης, η οποία είναι πιο ακριβής σε σχέση με την αντίστοιχη πρόβλεψη των κόμβων κοντά στη ρίζα του δέντρου. Μία άλλη τεχνική για την αποτελεσματικότερη αξιολόγηση των κόμβων περιλαμβάνει το συνδυασμό των προβλέψεων του ταξινομητή με το αποτέλεσμα προσομοιώσεων με πολιτική τυχαίων ενεργειών, καθώς και τη στοχαστική αξιολόγηση των κόμβων λαμβάνοντας υπόψη το μέγεθος του χώρου καταστάσεων.

Τα παραπάνω αξιολογήθηκαν μέσω ενός ευφυούς πράκτορα που υλοποιήθηκε στο περιβάλλον Metastone, το οποίο προσομοιώνει το ηλεκτρονικό παιχνίδι καρτών Hearthstone. Το συγκεκριμένο περιβάλλον εμπεριέχει μερική παρατηρησιμότητα και στοχαστικότητα που το καθιστούν κατάλληλο για τη δοκιμή της εξεταζόμενης μεθοδολογίας. Οι διαφορετικές παραλλαγές εξετάστηκαν μεμονωμένα αλλά και συνδυαστικά, με τον πράκτορα που ενσωματώνει ταυτόχρονα τις διαφορετικές τροποποιήσεις να είναι ο πιο αποτελεσματικός, ξεπερνώντας τόσο την απλή εκδοχή του MCTS όσο και τον αλγόριθμο αξίας κατάστασης παιχνιδιού που αποτελεί το μοντέλο αναφοράς στο συγκεκριμένο περιβάλλον. Μεμονωμένα, από τις διαφορετικές τροποποιήσεις η εισαγωγή του ταξινομητή και ο συνδυασμός της πρόβλεψής του με το αποτέλεσμα μίας τυχαίας προσομοίωσης παρουσίασαν τη μεγαλύτερη επίδραση στην απόδοση του πράκτορα. Αυτό οφείλεται κατά κύριο λόγο στη μείωση της διακύμανσης που επιτυγχάνεται με τη χρήση γνώσης πεδίου. Όσον αφορά στις διαφορετικές προσεγγίσεις εκπαίδευσης του μοντέλου, δεν παρατηρήθηκε αξιοσημείωτη υπεροχή των ταξινομητών που εκπαιδεύτηκαν σε εξειδικευμένα σύνολα δεδομένων ανά περίπτωση έναντι του γενικού ταξινομητή, ωστόσο αυτό το συμπέρασμα δε μπορεί να γενικευτεί καθώς εξαρτάται σε μεγάλο βαθμό από τα χαρακτηριστικά του περιβάλλοντος.

Ακολούθως, δόθηκε έμφαση στην ενίσχυση της φάσης επιλογής του MCTS (Κεφάλαιο 5). Αρχικά, προτάθηκε μία μεθοδολογία κλαδέματος με χρήση βαθιάς μάθησης με στόχο τη μείωση του χώρου καταστάσεων. Δύο νευρωνικά δίκτυα συνδυάστηκαν ώστε να προβλεφθεί το καταλληλότερο χρονικό σημείο (αριθμός επαναλήψεων του αλγορίθμου) και το βέλτιστο υποσύνολο ενεργειών που μπορούν να κλαδευτούν από το δέντρο αναζήτησης, ελαχιστοποιώντας την πιθανότητα να απορριφθεί η βέλτιστη ενέργεια. Για την εκπαίδευση των δικτύων, συντέθηκε ένα σύνολο δεδομένων από ένα ειδικά σχεδιασμένο μοντέλο προσομοίωσης του προβλήματος ληστή πολλαπλών χεριών. Επιπλέον, διαφορετικές κατανομές χρησιμοποιήθηκαν για την επιλογή της αναμενόμενης αξίας των ενεργειών, οδηγώντας στη δημιουργία δύο διαφορετικών συνόλων δεδομένων στα οποία εκπαιδεύτηκαν τα αντίστοιχα ζεύγη δικτύων κλαδέματος.

Για την αξιολόγηση του αλγορίθμου ένας νέος πράκτορας υλοποιήθηκε για το περιβάλλον του Metastone, στον οποίο εν συνεχεία ενσωματώθηκαν και οι τεχνικές ενίσχυσης της φάσης προσομοίωσης που αναφέρονται παραπάνω. Ο τελικός πράκτορας υπερίσχυσε στα πειράματα που διεξήχθησαν υπογραμμίζοντας τη χρησιμότητα των νευρωνικών δικτύων κλαδέματος, ιδιαίτερα στις περιπτώσεις με μεγάλο χώρο ενεργειών. Καθοριστικό ρόλο στην αποτελεσματικότητα της προτεινόμενης τεχνικής αποδείχθηκε ότι παίζει η επιλογή της κατονομής που χρησιμοποιείται για τη δημιουργία των δεδομένων

εκπαίδευσης. Από τα πειραματικά αποτελέσματα φάνηκε ότι ανάλογα με την κατανομή της αναμενόμενης αξίας των ενεργειών, τα δεδομένα εκπαίδευσης μπορεί να οδηγήσουν σε δίκτυα κλαδέματος με πολύ διαφορετική συμπεριφορά και επομένως απαιτείται διερεύνηση διαφορετικών κατανομών ανά περίπτωση.

Στο πλαίσιο της φάσης επιλογής του αλγορίθμου, προτάθηκε στη συνέχεια μία τεχνική για αποδοτικότερη χρήση των στατιστικών των κόμβων του δέντρου αναζήτησης χωρίς την εισαγωγή γνώσης πεδίου. Ο στόχος σε αυτή την περίπτωση είναι η ταχύτερη εκτίμηση της αξίας των ενεργειών συνδυάζοντας στατιστικές τιμές από διαφορετικούς κόμβους. Συγκεκριμένα, για την επιλογή του καταλληλότερου κόμβου σε κάθε περίπτωση υπολογίζεται η ομοιότητα μεταξύ των κόμβων με βάση την κατάσταση παιχνιδιού και οι τιμές των πιο κοντινών κόμβων (με την έννοια της ομοιότητας των καταστάσεων) συνδυάζονται. Δύο διαφορετικές προσεγγίσεις προτείνονται για τον καθορισμό της ομοιότητας καταστάσεων, στηριζόμενες στην ακολουθία των ενεργειών που οδηγεί στον κάθε κόμβο. Η προτεινόμενη μέθοδος δοκιμάστηκε για διαφορετικά παιχνίδια στο περιβάλλον του GVGAI πετυχαίνοντας υψηλότερα ποσοστά νίκης από αντίστοιχες τεχνικές ταχείας εκτίμησης αξίας ενέργειας στην πλειοψηφία των περιπτώσεων. Τα ποσοστά των πρακτόρων που αξιολογήθηκαν υπογραμμίζουν τη συσχέτιση της ακολουθίας των ενεργειών σε ένα μονοπάτι του δέντρου αναζήτησης με την κατάσταση του αντίστοιχου κόμβου. Συγχρόνως, αναδεικνύονται τα οφέλη από το διαμοιρασμό πληροφοριών μεταξύ κόμβων σε διαφορετικά επίπεδα του δέντρου, ιδιαίτερα στην περίπτωση που αναπαριστούν παρόμοιες καταστάσεις.

Τέλος, διερευνήθηκε το πεδίο της ενισχυτικής μάθησης και πιο συγκεκριμένα η ενσωμάτωση μίας τεχνικής γεννητικής επαύξησης δεδομένων. Η βασική ιδέα είναι η αύξηση της ποικιλομορφίας των δειγμάτων σε περιπτώσεις που συναντώνται λιγότερο συχνά κατά την αλληλεπίδραση του πράκτορα με το περιβάλλον, με στόχο την επιτάχυνση της εκπαίδευσης. Για την υλοποίηση αυτής της μεθόδου χρησιμοποιήθηκαν δύο μοντέλα διάχυσης και ένα μοντέλο αντίστροφης δυναμικής για τη δημιουργία συνθετικών δεδομένων. Τα συνθετικά δείγματα αφορούν περιπτώσεις με πολύ υψηλή ή πολύ χαμηλή ανταμοιβή, καθώς η πλειοψηφία των πραγματικών δεδομένων περιλαμβάνει δείγματα με μηδενική άμεση ανταμοιβή. Με αυτή τη μεθοδολογία τα νέα δείγματα είναι ανεξάρτητα από τα πραγματικά, σε αντίθεση με τα δεδομένα που προκύπτουν από τις παραδοσιακές τεχνικές επαύξησης, τα οποία είναι κατά βάση παραλλαγές των δειγμάτων που έχει ήδη συλλέξει ο πράκτορας.

Ο αλγόριθμος αξιολογήθηκε σε τρία διαφορετικά περιβάλλοντα παιχνιδιών, υπερτερώντας πρακτόρων που ενσωματώνουν κλασικές μεθόδους επαύξησης δεδομένων όσον αφορά στη συνολική ανταμοιβή και στην ταχύτητα εκπαίδευσης. Παρατηρήθηκε ότι σε αντίθεση με αυτές τις τεχνικές, οι οποίες παρουσιάζουν μεγάλη ευαισθησία στα ιδιαίτερα χαρακτηριστικά των καταστάσεων του κάθε περιβάλλοντος, η γεννητική επαύξηση δεδομένων οδηγεί σε πιο συνεπείς πράκτορες που δεν επηρεάζονται σε μεγάλο βαθμό από τις ιδιαιτερότητες των διαφορετικών παιχνιδιών. Κομβικής σημασίας για την αποτελεσματικότητα της προτεινόμενης μεθοδολογίας είναι το μοντέλο αντίστροφης δυναμικής. Κατά την πειραματική αξιολόγηση διαπιστώθηκε, μέσω της διερεύνησης του ορίου εμπιστοσύνης των προβλεπόμενων ενεργειών, ότι η αύξηση της ακρίβειας του μοντέλου μπορεί να συμβάλει σε σημαντική βελτίωση των συνθετικών δεδομένων εκπαίδευσης και κατ΄ επέκταση του ευφυούς πράκτορα. Επίσης, με το μοντέλο που υλοποιήθηκε στην παρούσα διατριβή, έγινε αντιληπτό ότι η αύξηση του ποσοστού συνθετικών δειγμάτων εκπαίδευσης ανά παρτίδα πάνω από ένα κατώφλι έχει αρνητικές συνέπειες. Αυτό ενδεχομένως οφείλεται στην ακρίβεια των προβλέψεων του συγκεκριμένου μοντέλου και πιθανώς η βελτίωση του να μπορεί να εξασφαλίσει και μεγαλύτερη χρησιμοποίηση των συνθετικών δεδομένων χωρίς να επηρεάζεται αρνητικά η εκπαίδευση.

7.2 Μελλοντικές Κατευθύνσεις

Ολοκληρώνοντας, παρατίθενται κάποιες σκέψεις σχετικά με τις πτυχές των προτεινόμενων αλγορίθμων που θα μπορούσαν να ερευνηθούν περαιτέρω. Αναφορικά με το πρώτο μέρος της διατριβής, ένα χαρακτηριστικό που χρήζει επιπλέον διερεύνησης είναι η επιλογή των N-πλειάδων για την αναπαράσταση του χώρου καταστάσεων. Στην παρούσα εργασία χρησιμοποιήθηκαν συγκεκριμένες τριπλέτες για αυτόν τον σκοπό, επιτυγχάνοντας τη διατήρηση του μεγαλύτερου μέρους της χρήσιμης για τον πράκτορα πληροφορίας χωρίς ιδιαίτερα υψηλό κόστος. Παρόλα αυτά, η αναπαράσταση του κόσμου του παιχνιδιού μπορεί να επιτευχθεί με πολλές διαφορετικές προσεγγίσεις N-πλειάδων και αξίζει να εξεταστούν πιθανοί συνδυασμοί που ενδεχομένως να οδηγούν σε ακόμα καλύτερη αξιοποίηση των δεδομένων εισόδου. Επίσης, τα περιβάλλοντα που εξετάστηκαν ανήκουν στην ίδια κατηγορία παιχνιδιών προκειμένου να είναι εφικτή η επαναχρησιμοποιήση τμήματος του ΓΑ από το πρώτο περιβάλλον στο δεύτερο. Αυτό είναι αναγκαίο ώστε να υπάρχει ένα κοινό υποσύνολο χρωμοσωμάτων για τα διαφορετικά περιβάλλοντα, ωστόσο θα ήταν χρήσιμη η υλοποίηση και δοκιμή του αλγορίθμου σε διαφορετικές κατηγορίες (με την προϋπόθεση σε κάθε περίπτωση ο ΓΑ να εκπαιδεύεται για τα παιχνίδια μίας συγκεκριμένης κατηγορίας) ώστε να αξιολογηθεί η δυνατότητα εφαρμογής του σε περιβάλλοντα με διαφορετικά χαρακτηριστικά.

Στο πλαίσιο της ενίσχυσης του αλγορίθμου δενδρικής αναζήτησης Μοντε Κάρλο αναδείχθηκε η επίδραση του μεγέθους του χώρου ενεργειών στην αποτελεσματικότητά του. Αυτό, όσον αφορά στη φάση αξιολόγησης, αντιμετωπίστηκε στην προτεινόμενη μεθοδολογία μέσω της εισαγωγής στοχαστικής αξιολόγησης των κόμβων. Μία προσέγγιση που δε συναντάται στη σχετική βιβλιογραφία και έχει ενδιαφέρον είναι η χρήση τεχνικών συσταδοποίησης προκειμένου να ομαδοποιηθούν παρόμοιες ενέργειες. Σε αυτή την περίπτωση θα μπορούσε να δοκιμαστεί αξιολόγηση αρχικά σε επίπεδο συστάδων και εν συνεχεία σε επίπεδο κόμβων, με στόχο την εστίαση της αναζήτησης σε μικρότερο υποσύνολο των ενεργειών. Επιπλέον, το μοντέλο αξιολόγησης των ενεργειών που χρησιμοποιήθηκε στην παρούσα έρευνα προβλέπει το τελικό αποτέλεσμα μίας παρτίδας. Μία επέκταση που ενδεχομένως να οδηγήσει σε ακριβέστερη αξιολόγηση, είναι η εκαίδευσή του κατά τέτοιο τρόπο ώστε οι προβλέψεις να αντικατοπτρίζουν εκτός από το τελικό αποτέλεσμα και το βαθμό δυσκολίας της επίτευξής του. Αυτό απαιτεί μία μέθοδο εκτίμησης της αξίας της τελικής κατάστασης του παιχνιδιού κατά τη δημιουργία του συνόλου δεδομένων εκπαίδευσης, προκειμένου να αντιμετωπιστεί η πρόβλεψη ως πρόβλημα παλινδρόμησης.

Στην περίπτωση της φάσης επιλογής του MCTS και συγκεκριμένα στην προτεινόμενη τεχνική κλαδέματος με νευρωνικά δίκτυα, διερευνήθηκαν δύο διαφορετικές κατανομές για τη δημιουργία των δεδομένων εκπαίδευσης των δικτύων. Καθώς από τα πειραματικά αποτελέσματα προέκυψαν σημαντικές διαφορές στη συμπεριφορά των δύο πρακτόρων, η εκπαίδευση των δικτύων κλαδέματος σε δεδομένα που βασίζονται σε διαφορετικές κατονομές μπορεί να βοηθήσει στην περαιτέρω κατανόηση της επίδρασης της κατανομής των πραγματικών αναμενόμενων αξιών των ενεργειών στον τελικό αλγόριθμο. Πέραν αυτού, θα μπορούσε να εξεταστεί το προσωρινό κλάδεμα ώστε να μην αποκλείονται εντελώς ενέργειες που ενδεχομένως να έχουν αξιολογηθεί εσφαλμένα έπειτα από μικρό αριθμό επισκέψεων. Όπως παρατηρείται στην πειραματική διαδικασία αυτό το σενάριο είναι αρκετά πιθανό, ειδικά στην περίπτωση των δικτύων που βασίζονται στη διτροπική κατανομή. Δεδομένου ότι η προτεινόμενη μέθοδος περιλαμβάνει επαναληπτικό κλάδεμα, η αναίρεση κλαδέματος σε διαφορετικά στάδια (που καθορίζονται από τα δίκτυα) της δενδρικής αναζήτησης θα μπορούσε να χρησιμοποιηθεί με στόχο τη μείωση του κινδύνου απόρριψης ενεργειών με υψηλή αξία. Υπό αυτό το πρίσμα, μία υποσχόμενη προσέγγιση θα ήταν η στοχαστική αναίρεση κλαδέματος, καθώς και η χρήση ενός ξεχωριστού

μοντέλου για την επιλογή των ενεργειών που μπορούν να ενσωματωθούν εκ νέου στο δέντρο αναζήτησης. Όσον αφορά στον αλγορίθμο ταχείας εκτίμησης αξίας ενέργειας με ομοιότητα καταστάσεων, καθώς ο χώρος ενεργειών των παιχνιδιών που εξετάστηκαν είναι περιορισμένος, θα ήταν χρήσιμο να αξιολογηθούν οι προτεινόμενες παραλλαγές και σε παιχνίδια με υψηλότερο παράγοντα διακλάδωσης προκειμένου να διερευνηθεί η ικανότητα κλιμάκωσης των αλγορίθμων. Επιπλέον, υβριδικές μέθοδοι που συνδυάζουν τους καταλληλότερους κόμβους υψηλότερων επιπέδων όπως προκύπτουν από διαφορετικές μεθοδολογίες, θα μπορούσαν να εξεταστούν ως προς το ενδεχόμενο βέλτιστης αξιοποίησης των πλεονεκτημάτων της κάθε τεχνικής.

Τέλος, τα πειράματα που εκτελέστηκαν για την αξιολόγηση της χρήσης γεννητικής επαύξησης δεδομένων στο πεδίο της ενισχυτικής μάθησης, τόνισαν την επίδραση του μοντέλου πρόβλεψης ενέργειας στο ρυθμό μάθησης του πράκτορα και κατά συνέπεια στη συνολική του απόδοση. Η ποιότητα των συνθετικών δειγμάτων εξαρτάται από αυτό το μοντέλο και επηρεάζει σε μεγάλο βαθμό τη συμπεριφορά του πράκτορα. Συνεπώς, ο πειραματισμός με διαφορετικές αρχιτεκτονικές του μοντέλου αντίστροφης δυναμικής με στόχο τη μεγαλύτερη ακρίβεια των προβλέψεων, μπορεί να οδηγήσει εμμέσως σε αύξηση της συνολικής απόδοσης του αλγορίθμου. Επιπροσθέτως, τα μοντέλα επαύξησης θα μπορούσαν να εκπαιδευτούν παράλληλα με το δίκτυο του πράκτορα, ώστε να διερευνηθεί και η μεταξύ τους αλληλεπίδραση καθώς ο πράκτορας αλληλεπιδρά με το περιβάλλον. Σε αυτό το πλαίσιο, καθώς ο πράκτορας συλλέγει νέα δείγματα, η περιοδική επανεκπαίδευση των δικτύων επαύξησης ενδεχομένως επίσης να βελτιώσει την ακρίβεια και την ποικιλομορφία των συνθετικών δειγμάτων.

Παράρτημα Α΄

Αρχιτεκτονικές Νευρωνικών Δικτύων

Πίνακας Α΄.1: Αρχιτεκτονική μοντέλου αντίστροφης δυναμικής

	Layer	Output Size	Input Layer	Kernel Size	Activation
	#1 Input	$84 \times 84 \times 5$	-	-	-
	#2 Conv2D	$42 \times 42 \times 128$	#1	3×3	-
	#3 BatchNorm	$42 \times 42 \times 128$	#2	-	ReLU
	#4 SepConv2D	$42 \times 42 \times 256$	#3	3×3	-
	#5 BatchNorm	$42 \times 42 \times 256$	#4	-	ReLU
	#6 SepConv2D	$42 \times 42 \times 256$	#5	3×3	-
Res1	#7 BatchNorm	$42 \times 42 \times 256$	#6	-	-
	#8 MaxPooling2D	$21 \times 21 \times 256$	#7	3×3	-
	#9 Conv2D	$21 \times 21 \times 256$	#3	1×1	-
	#10 Add	$21 \times 21 \times 256$	#8, #9	1×1	ReLU
	#11 SepConv2D	$21 \times 21 \times 512$	#10	3×3	-
	#12 BatchNorm	$21 \times 21 \times 512$	#11	-	ReLU
	#13 SepConv2D	$21 \times 21 \times 512$	#12	3×3	-
Res2	#14 BatchNorm	$21 \times 21 \times 512$	#13	-	-
	#15 MaxPooling2D	$11 \times 11 \times 512$	#14	3×3	-
	#16 Conv2D	$11 \times 11 \times 512$	#10	1×1	-
	#17 Add	$11 \times 11 \times 512$	#10, #16	1×1	ReLU
	#18 SepConv2D	$11 \times 11 \times 728$	#17	3×3	-
	#19 BatchNorm	$11 \times 11 \times 728$	#18	-	ReLU
	#20 SepConv2D	$11 \times 11 \times 728$	#19	3×3	-
Res3	#21 BatchNorm	$11 \times 11 \times 728$	#20	-	-
	#22 MaxPooling2D	$6 \times 6 \times 728$	#21	3×3	-
	#23 Conv2D	$6 \times 6 \times 728$	#22	1×1	-
	#24 Add	$6 \times 6 \times 728$	#17, #22	1×1	ReLU
	#25 SepConv2D	$6 \times 6 \times 1024$	#24	3×3	-
	#26 BatchNorm	$6 \times 6 \times 1024$	#25	-	ReLU
	#27 GlobalAvPooling2D	1024	#26	-	-
	#28 Dropout (0.5)	1024	#27	-	-
	#29 Dense	actions	#28	-	Softmax

Πίνακας Α΄.2: Αρχιτεκτονική μοντέλου πράκτορα ενισχυτικής μάθησης

Layer	**Output Size**	**Input Layer**	**Kernel/Pool Size**	**Activation**
#1 Input	$84 \times 84 \times 4$	-	-	-
#2 Conv2D	$20 \times 20 \times 32$	#1	8×8	ReLU
#3 Conv2D	$9 \times 9 \times 64$	#2	4×4	ReLU
#4 Conv2D	$7 \times 7 \times 64$	#3	3×3	ReLU
#5 Dense	512	#4	-	ReLU
#6 Dense	actions	#5	-	Softmax

Πίνακας Α΄.3: Υπερπαράμετροι μοντέλου πράκτορα ενισχυτικής μάθησης

Hyperparameter	**Value**
Learning rate	1×10^{-4}
Optimizer	Adam
Batch size	32
Discount factor (γ)	0.99
Real buffer size	100000
Synthetic buffer size	20000
$epsilon_{max}$	1.0
$epsilon_{min}$	0.1
Initial random timesteps	30000
Evaluation episodes	50
Train frequency (timesteps)	4
Update target network frequency (timesteps)	5000

Βιβλιογραφία

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu και Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

[2] Pulkit Agrawal, Joao Carreira και Jitendra Malik. Learning to see by moving. Στο *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[3] M. T. Al-Hajri και M. A. Abido. Assessment of genetic algorithm selection, crossover and mutation techniques in reactive power optimization. Στο *2009 IEEE Congress on Evolutionary Computation*, σελίδες 1005–1011, 2009.

[4] Firas Alabsi και Reyadh Naoum. Comparison of selection methods and crossover operations using steady state genetic based intrusion detection system. *Journal of Emerging Trends in Computing and Information Sciences*, 3(7):1053–1058, 2012.

[5] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3:397–422, 2003.

[6] Randall D Beer. A dynamical systems perspective on agent-environment interaction. *Artificial intelligence*, 72(1-2):173–215, 1995.

[7] M. G. Bellemare, Y. Naddaf, J. Veness και M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[8] Christopher M Bishop και Nasser M Nasrabadi. *Pattern recognition and machine learning*, τόμος 4. Springer, 2006.

[9] Tobias Blickle και Lothar Thiele. A comparison of selection schemes used in evolutionary algorithms. *Evol. Comput.*, 4(4):361–394, 1996.

[10] Blizzard Entertainment, Irvine, CA, USA. *Hearthstone*, 2014.

[11] David Bourg και Glenn Seemann. *AI for Game Developers Creating Intelligent Behavior in Games*. O'Reilly Media, 2014.

[12] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang και Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[13] T Bullen και M Katchabaw. Using genetic algorithms to evolve character behaviours in modern video games. *Proceedings of the GAMEON-NA*, 2008.

[14] Jenna Carr. An introduction to genetic algorithms. *Senior Project*, σελίδες 1–40, 2014.

[15] Tristan Cazenave. Generalized rapid action value estimation. Στο *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[16] Guillaume M JB Chaslot, Mark HM Winands, H Jaap Van Den Herik, Jos WHM Uiterwijk και Bruno Bouzy. Progressive strategies for monte-carlo tree search. *New Mathematics and Natural Computation*, 4(03):343–357, 2008.

[17] Paul Christiano, Zain Shah, Igor Mordatch, Jonas Schneider, Trevor Blackwell, Joshua Tobin, Pieter Abbeel και Wojciech Zaremba. Transfer from simulation to real world through learning deep inverse dynamics model. *arXiv preprint arXiv:1610.03518*, 2016.

[18] Jack Copeland. *Artificial intelligence: A philosophical introduction*. John Wiley & Sons, 1993.

[19] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest και Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

[20] Adrien Couëtoux, Jean Baptiste Hoock, Nataliya Sokolovska, Olivier Teytaud και Nicolas Bonnard. Continuous upper confidence trees. Στο *International Conference on Learning and Intelligent Optimization*, σελίδες 433–445. Springer, 2011.

[21] Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. Στο *International conference on computers and games*, σελίδες 72–83. Springer, 2006.

[22] Rémi Coulom. Computing "elo ratings" of move patterns in the game of go. *ICGA journal*, 30(4):198–208, 2007.

[23] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan και Quoc V Le. Autoaugment: Learning augmentation strategies from data. Στο *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, σελίδες 113–123, 2019.

[24] Claudio Comis Da Ronco και Ernesto Benini. *A Simplex-Crossover-Based Multi-Objective Evolutionary Algorithm*, σελίδες 583–598. Springer Netherlands, Dordrecht, 2014.

[25] R. S.da Silva και R. S. Parpinelli. Playing the original game boy tetris using a real coded genetic algorithm. Στο *2017 Brazilian Conference on Intelligent Systems (BRACIS)*, σελίδες 282–287, 2017.

[26] Jörg Denzinger και Michael Kordt. Evolutionary online learning of cooperative behavior with situation-action pairs. Στο *MultiAgent Systems, 2000. Proceedings. Fourth International Conference on*, σελίδες 103–110. IEEE, 2000.

[27] Richard O Duda και Peter E Hart. *Pattern classification*. John Wiley & Sons, 2006.

[28] Stan Franklin και Art Graesser. *Is It an agent, or just a program?: A taxonomy for autonomous agents*, σελίδες 21–35. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[29] Sylvain Gelly και David Silver. Monte-carlo tree search and rapid action value estimation in computer go. *Artificial Intelligence*, 175(11):1856–1875, 2011.

[30] Sylvain Gelly, Yizao Wang, Rémi Munos και Olivier Teytaud. *Modification of UCT with patterns in Monte-Carlo Go*. Διδακτορική Διατριβή , INRIA, 2006.

[31] Yoonhee Gil, Jongchan Baek, Jonghyuk Park και Soohee Han. Automatic data augmentation by upper confidence bounds for deep reinforcement learning. Στο *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, σελίδες 1199–1203. IEEE, 2021.

[32] GitHub, Inc. *MetaStone simulator*, 2016.

[33] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel και Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. Στο *International conference on machine learning*, σελίδες 1861–1870. PMLR, 2018.

[34] Danijar Hafner, Timothy Lillicrap, Jimmy Ba και Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[35] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee και James Davidson. Learning latent dynamics for planning from pixels. Στο *Proceedings of the 36th International Conference on Machine Learning,*Kamalika Chaudhuri και Ruslan Salakhutdinov, επιμελητές, τόμος 97 στο *Proceedings of Machine Learning Research*, σελίδες 2555–2565. PMLR, 2019.

[36] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi και Jimmy Ba. Mastering atari with discrete world models, 2022.

[37] Nicklas Hansen, Hao Su και Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in neural information processing systems*, 34:3680–3693, 2021.

[38] Nicklas Hansen και Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. Στο *2021 IEEE International Conference on Robotics and Automation (ICRA)*, σελίδες 13611–13617. IEEE, 2021.

[39] Simon Haykin. *Neural networks and learning machines, 3/E*. Pearson Education India, 2009.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. Deep residual learning for image recognition. Στο *Proceedings of the IEEE conference on computer vision and pattern recognition*, σελίδες 770–778, 2016.

[41] Geoffrey Hinton και Terrence J Sejnowski. *Unsupervised learning: foundations of neural computation*. MIT press, 1999.

[42] Sepp Hochreiter και Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[43] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. Στο *The collected works of Wassily Hoeffding*, σελίδες 409–426. Springer, 1994.

[44] Jonathan Ho, Ajay Jain και Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[45] W. L. Hsu και Y.p. Chen. Learning to select actions in starcraft with genetic algorithms. Στο *2016 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, σελίδες 270–277, 2016.

[46] Jing Huang, Zhiqing Liu, Benjie Lu και Feng Xiao. Pruning in uct algorithm. Στο *2010 International Conference on Technologies and Applications of Artificial Intelligence*, σελίδες 177–181. IEEE, 2010.

[47] Tao Huang, Jiachen Wang και Xiao Chen. Accelerating representation learning with view-consistent dynamics in data-efficient reinforcement learning. *arXiv preprint arXiv:2201.07016*, 2022.

[48] Brijnesh J. Jain, Hartmut Pohlheim και Joachim Wegener. On termination criteria of evolutionary algorithms. Στο *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, GECCO'01, σελίδες 768–768, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[49] Dinesh Jayaraman και Kristen Grauman. Learning image representations tied to ego-motion, 2016.

[50] Khalid Jebari και Mohammed Madiafi. Selection methods for genetic algorithms. *International Journal of Emerging Sciences*, 3(4):333–344, 2013.

[51] Michael N. Katehakis και Arthur F. Veinott. The multi-armed bandit problem: Decomposition and computation. *Mathematics of Operations Research*, 12(2):262–268, 1987.

[52] Diederik P Kingma και Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[53] Levente Kocsis και Csaba Szepesvári. Bandit based monte-carlo planning. Στο *European conference on machine learning*, σελίδες 282–293. Springer, 2006.

[54] Ilya Kostrikov, Denis Yarats και Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

[55] Oliver Kramer. *Genetic Algorithm Essentials*. Springer International Publishing, 2017.

[56] Alex Krizhevsky, Ilya Sutskever και Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[57] Anand Kumar. *Network Design using Genetic Algorithm*, σελίδα 256. LAP LAMBERT Academic Publishing, 2016.

[58] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel και Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020.

[59] Yann LeCun, Léon Bottou, Yoshua Bengio και Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[60] Felix Leibfried, Nate Kushman και Katja Hofmann. A deep learning approach for joint video frame and reward prediction in atari games. *arXiv preprint arXiv:1611.07078*, 2016.

[61] Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy Hospedales, Neil M Robertson και Yongxin Yang. Differentiable automatic data augmentation. Στο *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, σελίδες 580–595. Springer, 2020.

[62] Ian Millington και John Funge. *Artificial Intelligence for Games, Second Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2η έκδοση, 2009.

[63] Frederick Mills και Robert Stufflebeam. Introduction to intelligent agents, 2005.

[64] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA, 1998.

[65] Tom M Mitchell. Machine learning, 1997.

[66] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver και Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning, 2016.

[67] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra και Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[68] P Read Montague. Reinforcement learning: an introduction, by sutton, rs and barto, ag. *Trends in cognitive sciences*, 3(9):360, 1999.

[69] Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik και Sergey Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. Στο *2017 IEEE International Conference on Robotics and Automation (ICRA)*, σελίδες 2146–2153, 2017.

[70] Allen Newell. The knowledge level. *Artificial intelligence*, 18(1):87–127, 1982.

[71] Thanh Nguyen, Tung M Luu, Thang Vu και Chang D Yoo. Sample-efficient reinforcement learning representation learning with curiosity contrastive forward dynamics model. Στο *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, σελίδες 3471–3477. IEEE, 2021.

[72] Nils J Nilsson. *Artificial intelligence: a new synthesis*. Morgan Kaufmann, 1998.

[73] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis και Satinder Singh. Action-conditional video prediction using deep networks in atari games. *Advances in neural information processing systems*, 28, 2015.

[74] Abdessamed Ouessai, Mohammed Salem και Antonio M Mora. Improving the performance of mcts-based μrts agents through move pruning. Στο *2020 IEEE Conference on Games (CoG)*, σελίδες 708–715. IEEE, 2020.

[75] Keiran Paster, Sheila A McIlraith και Jimmy Ba. Planning from pixels using inverse dynamics models. *arXiv preprint arXiv:2012.02419*, 2020.

[76] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros και Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. Στο *Proceedings of the 34th International Conference on Machine Learning*,Doina Precup και Yee Whye Teh, επιμελητές, τόμος 70 στο *Proceedings of Machine Learning Research*, σελίδες 2778–2787. PMLR, 2017.

[77] Tom Pepels, Mark HM Winands και Marc Lanctot. Real-time monte carlo tree search in ms pac-man. *IEEE Transactions on Computational Intelligence and AI in games*, 6(3):245–257, 2014.

[78] Diego Perez-Liebana, Spyridon Samothrakis, Julian Togelius, Tom Schaul και Simon Lucas. General video game ai: Competition, challenges and opportunities. Στο *Proceedings of the AAAI Conference on Artificial Intelligence*, τόμος 30, 2016.

[79] Diego Perez-Liebana, Spyridon Samothrakis, Julian Togelius, Tom Schaul, Simon M Lucas, Adrien Couëtoux, Jerry Lee, Chong U Lim και Tommy Thompson. The 2014 general video game playing competition. *IEEE Transactions on Computational Intelligence and AI in Games*, 8(3):229–243, 2015.

[80] Diego Perez, Spyridon Samothrakis, Simon Lucas και Philipp Rohlfshagen. Rolling horizon evolution versus tree search for navigation in single-player real-time games. Στο *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, σελίδες 351–358, New York, NY, USA, 2013. ACM.

[81] Sébastien Racanière, Théophane Weber, David Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adrià Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li και others. Imagination-augmented agents for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[82] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov και Rob Fergus. Automatic data augmentation for generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 34:5402–5415, 2021.

[83] Olaf Ronneberger, Philipp Fischer και Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. Στο *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, σελίδες 234–241. Springer, 2015.

[84] Peter Norvig Russell. Artificial intelligence: a modern approach by stuart. *Russell and Peter Norvig contributing writers, Ernest Davis...[et al.]*, 2010.

[85] Stuart Russell. Learning agents for uncertain environments (extended abstract). Στο *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, σελίδες 101–103, New York, NY, USA, 1998. ACM.

[86] Stuart J. Russell και Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[87] Tim Salimans, Jonathan Ho, Xi Chen και Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *CoRR*, 1703.03864, 2017.

[88] Robert E Schapire και Yoav Freund. Foundations of machine learning. 2012.

[89] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford και Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[90] H.P. Schwefel. *Numerical Optimization of Computer Models*. Interdisciplinary systems research. John Wiley and Sons, 1981.

[91] Bob Scott. *AI Game Programming Wisdom*, σελίδες 16–20. Charles River Media, 2002.

[92] Nick Sephton, Peter I Cowling, Edward Powley και Nicholas H Slaven. Heuristic move pruning in monte carlo tree search for the strategic card game lords of war. Στο *2014 IEEE Conference on Computational Intelligence and Games*, σελίδες 1–7. IEEE, 2014.

[93] Chiara F Sironi και Mark HM Winands. Comparison of rapid action value estimation variants for general game playing. Στο *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, σελίδες 1–8. IEEE, 2016.

[94] Nitasha Soni και Tapas Dr. Kumar. A ring crossover genetic algorithm for the unit commitment problem. *International Journal of Computer Science and Information Technologies*, 5, 2014.

[95] Richard S Sutton, David McAllester, Satinder Singh και Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[96] Mandy JW Tak, Mark HM Winands και Yngvi Bjornsson. N-grams and the last-good-reply policy applied in general game playing. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(2):73–83, 2012.

[97] A. M. Turing. I.—computing machinery and intelligence. *Mind*, (236):433–460, 1950.

[98] AJ Umbarkar και PD Sheth. Crossover operators in genetic algorithms: A review. *ICTACT journal on soft computing*, 6(1), 2015.

[99] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser και Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[100] John Von Neumann και Oskar Morgenstern. Theory of games and economic behavior, 2nd rev. 1947.

[101] Geogios N. Yannakakis. Game ai revisited. Στο *Proceedings of the 9th Conference on Computing Frontiers*, CF '12, σελίδες 285–292, New York, NY, USA, 2012. ACM.

[102] Denis Yarats, Rob Fergus, Alessandro Lazaric και Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021.

[103] Zhecheng Yuan, Guozheng Ma, Yao Mu, Bo Xia, Bo Yuan, Xueqian Wang, Ping Luo και Huazhe Xu. Don't touch what matters: Task-aware lipschitz data augmentation for visual reinforcement learning. *arXiv preprint arXiv:2202.09982*, 2022.

[104] Tao Yu, Cuiling Lan, Wenjun Zeng, Mingxiao Feng, Zhizheng Zhang και Zhibo Chen. Playvirtual: Augmenting cycle-consistent virtual trajectories for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:5276–5289, 2021.

[105] Hanping Zhang και Yuhong Guo. Generalization of reinforcement learning with policy-aware adversarial data augmentation. *arXiv preprint arXiv:2106.15587*, 2021.

Συντομογραφίες - Αρκτικόλεξα - - Ακρωνύμια

ΓΑ	Γενετικός Αλγόριθμος
ΕΑ	Εξελικτικός Αλγόριθμος
ΕΠ	Ευφυής Πράκτορας
ΜΔΑ	Μαρκοβιανή Διαδικασία Αποφάσεων
ΜΚ	Μόντε Κάρλο
ΤΝΔ	Τεχνητό Νευρωνικό Δίκτυο
ΤΝ	Τεχνητή Νοημοσύνη
ΧΔ	Χρονική Διαφορά

AI	Artificial Intelligence
AMAF	All Moves As First
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CRAVE	Cancel-out Rapid Action Value Estimation
DDPM	Denoising Diffusion Probabilistic Model
DQN	Deep Q-Learning
EA	Evolutionary Algorithm
GA	Genetic Algorithm
GAMER	Genetic Algorithm with Motion Encoding Reuse
GRAVE	Generalized Rapid Action Value Estimation
GSV	Game State Value
GVGAI	General Video Game Artificial Intelligence
IA	Intelligent Agent
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
MAB	Multi-Armed Bandit
MAE	Mean Absolute Error
MC	Monte Carlo
MCTS	Monte Carlo Tree Search
MDP	Markov Decision Process
MSE	Mean Squared Error

NPC	Non-Player Character
NRAVE	N-grams based Rapid Action Value Estimation
OLEMAS	On-Line Evolution of Multi-Agent Systems
PMX	Partially Matched Crossover
PPO	Proximal Policy Optimization
RAVE	Rapid Action Value Estimation
ReLU	Rectified Linear Unit
RHGA	Rolling Horizon Genetic Algorithm
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SAC	Soft Actor-Critic
SARSA	State-Action-Reward-State-Action
SGD	Stochastic Gradient Descent
SW	Synthetic Weight
TD	Temporal Difference
UCB	Upper Confidence Bound
UCT	Upper Confidence bounds applied to Trees
VGDL	Video Game Description Language
XGBoost	eXtreme Gradient Boosting

Απόδοση Ξενόγλωσσων Όρων

Absolute	Απόλυτος
Action	Ενέργεια
Action-Value Function	Συνάρτηση Αξίας Ενέργειας
Activation	Ενεργοποίηση
Activation Function	Συνάρτηση Ενεργοποίησης
Actor	Δράστης
Age-based	Βασισμένος στην Ηλικία
Agent Function	Συνάρτηση Πράκτορα
Agent Program	Πρόγραμμα Πράκτορα
All Moves As First	Όλες οι Ενέργειες Σαν Πρώτες
Ancestor	Πρόγονος
Applied	Εφαρμοσμένος
Artificial Intelligence	Τεχνητή Νοημοσύνη
Artificial Neural Network	Τεχνητό Νευρωνικό Δίκτυο
Artificial Neuron	Τεχνητός Νευρώνας
Autonomy	Αυτονομία
Average	Μέσος Όρος
Back-propagation	Οπισθοδιάδοση
Baseline	Σημείο Αναφοράς
Batch	Παρτίδα
Bias	Πόλωση
Bimodal	Διτροπικός
Bit Flip	Αντιστροφή Ψηφίου
Bottom-Up	Από Κάτω προς τα Πάνω
Boundary	Όριο
Branching Factor	Παράγοντας Διακλάδωσης
Buffer	Προσωρινή Μνήμη
Cancel-out	Ακύρωση
Categorical	Κατηγορικός
Chain Rule	Κανόνας Αλυσίδας
Chromosome	Χρωμόσωμα
Classification	Ταξινόμηση
Competitive	Ανταγωνιστικός
Condition-Action Rule	Κανόνας Συνθήκης-Ενέργειας
Confidence Threshold	Κατώφλι Εμπιστοσύνης

Connection	Σύνδεση
Continuous	Συνεχής
Convolutional	Συνελικτικός
Cooperative	Συνεργατικός
Cost Function	Συνάρτηση Κόστους
Critic	Κριτής
Cross-Entropy	Σταυροειδής Εντροπία
Crossover	Διασταύρωση
Curiosity	Περιέργεια
Cutout	Αποκοπή
Cycle	Κυκλικός
Data Augmentation	Επαύξηση Δεδομένων
Deep Q-Learning	Βαθιά Εκμάθηση-Q
Denoising Diffusion Probabilistic Model	Πιθανοτικό Μοντέλο Διάχυσης Αποθορύφωσης
Description Language	Περιγραφική Γλώσσα
Deterministic	Ντετερμινιστικός
Dilemma	Δίλημμα
Dimensionality	Διαστατικότητα
Directed Acyclic Graph	Κατευθυνόμενος Ακυκλικός Γράφος
Discount Factor	Συντελεστής Μείωσης
Discounted Return	Μειωμένη Επιστροφή
Discrete	Διακριτός
Displacement	Μετατόπιση
Distribution	Κατανομή
Domain Dependent	Εξαρτώμενος από το Πεδίο
Domain Independent	Ανεξάρτητος Πεδίου
Domain Knowledge	Γνώση Πεδίου
Drive-Preference	Αρχική Γνώση
Dropout	Απόσυρση
Dynamic	Δυναμικός
Effector	Ενεργοποιητής
Encoding	Κωδικοποίηση
Environment	Περιβάλλον
Episode	Επεισόδιο
Episodic	Επεισοδιακό
Epoch	Εποχή
Error	Σφάλμα
Every-visit	Κάθε Επίσκεψης
Evolutionary	Εξελικτικός
Exploitation	Εκμετάλλευση
Exploration	Εξερεύνηση
Expansion	Επέκταση
Exponential	Εκθετικός
Extreme Gradient Boosting	Ακραία Ενίσχυση Κλίσης
Feature	Χαρακτηριστικό
Feedback	Ανάδραση

Feed-Forward	Πρόσθιας Τροφοδότησης
Finite State Machine	Μηχανή Πεπερασμένων Καταστάσεων
First-Order Logic	Λογική Πρώτης Τάξης
First-visit	Πρώτης Επίσκεψης
Fitness Function	Συνάρτηση Ποιότητας
Flat	Επίπεδος
Flip	Αντιστροφή
Forward Diffusion Process	Πρόσθια Διαδικασία Διάχυσης
Framework	Προγραμματιστικό Περιβάλλον
Fully Connected	Πλήρως Διασυνδεδεμένο
Fully Observable	Πλήρως Παρατηρήσιμο
Game Artificial Intelligence	Τεχνητή Νοημοσύνη σε Παιχνίδια
Game State Value	Αξία Κατάστασης Παιχνιδιού
Gene	Γονίδιο
General	Γενικευμένος
Generation	Γενιά
Genetic	Γενετικός
Genotype	Γενότυπος
Given the model	Δεδομένου Μοντέλου
Global	Καθολικός
Goal-based Agent	Πράκτορας βασισμένος σε Στόχους
Greedy	Άπληστος
Grid Search	Αναζήτηση Πλέγματος
Groundtruth	Πραγματική Τιμή
Growth Factor	Παράγονατς Αύξησης
Half-uniform	Ημιομοιόμορφος
Hard Pruning	Μόνιμο Κλάδεμα
Hidden	Κρυφό
Hyperbolic Tangent	Υπερβολική Εφαπτομένη
Imagined	Φανταστικός
Input	Είσοδος
Insertion	Εισαγωγή
Intelligent Agent	Ευφυής Πράκτορας
Internal State	Εσωτερική Κατάσταση
Inverse	Αντίστροφος
K-Nearest Neighbors	Κ-Πλησιέστεροι Γείτονες
Knowledge-based	Βασισμένος σε Γνώση
Label	Ετικέτα
Layer	Επίπεδο
Learn the model	Εκμάθησης Μοντέλου
Learning Agent	Πράκτορας Εκμάθησης
Learning Element	Στοιχείο Εκμάθησης
Learning Rate	Ρυθμός Μάθησης
Linear	Γραμμικός
Local	Τοπικός
Long Short-Term Memory	Μακράς Βραχύχρονης Μνήμης

Machine Learning	Μηχανική Μάθηση
Markov Decision Process	Μαρκοβιανή Διαδικασία Αποφάσεων
Match	Αντιστοίχιση
Mean Absolute Error	Μέσο Απόλυτο Σφάλμα
Mean Squared Error	Μέσο Τετραγωνικό Σφάλμα
Meta-learning	Μετα-μάθηση
Model-based	Βασισμένος σε Μοντέλο
Model-free	Ανεξάρτητος Μοντέλου
Monte Carlo Tree Search	Δενδρική Αναζήτηση Μόντε Κάρλο
Motion Encoding Reuse	Επαναχρησιμοποίηση της Κωδικοποίησης Κίνησης
Multi-Agent	Πολλών Πρακτόρων
Multi-Armed Bandit	Ληστής Πολλαπλών Χεριών
Multi-Point	Πολλαπλών Σημείων
Mutation	Μετάλλαξη
N-gram	Ν-γραμμο
N-group	Ν-πλειάδα
Narrow AI	Εξειδικευμένη Τεχνητή Νοημοσύνη
Navigation	Πλοήγηση
Node	Κόμβος
Non-player Character	Χαρακτήρας Υπολογιστή
Normalization	Κανονικοποίηση
Observability	Παρατηρησιμότητα
Off-line	Εκτός Λειτουργίας
Off-policy	Εκτός Πολιτικής
On-line	Πραγματικού Χρόνου
On-policy	Εντός Πολιτικής
One-Armed Bandit	Μηχανή Κουλοχέρη
Open Source	Ανοιχτού Κώδικα
Operator	Τελεστής
Optimal	Βέλτιστος
Outlier	Ακραία Τιμή
Output	Έξοδος
Padding	Παραγέμισμα
Partially	Μερικώς
Parent	Γονέας
Pattern	Πρότυπο
Penalty	Ποινή
Percept	Αντίληψη
Performance Element	Στοιχείο Επίδοσης
Performance Standard	Πρότυπο Επίδοσης
Permutation	Μετάθεση
Phenotype	Φαινότυπος
Pixel	Εικονοστοιχείο
Planning	Σχεδιασμός
Policy	Πολιτική
Policy-based	Βασισμένος στην Πολιτική

Policy Evaluation	Αξιολόγηση Πολιτικής
Policy Gradient	Κλίση Πολιτικής
Policy Improvement	Βελτίωση Πολιτικής
Policy Iteration	Επανάληψη Πολιτικής
Pooling	Συμψηφισμός
Population	Πληθυσμός
Posterior	Δεσμευμένος
Probability Network	Δίκτυο Πιθανότητας
Problem Generator	Γεννήτρια Προβλημάτων
Programming	Προγραμματισμός
Progressive Unpruning	Προοδευτική Αναίρεση Κλαδέματος
Progressive Widening	Προοδευτική Διεύρυνση
Proximal Policy Optimization	Κοντινή Βελτιστοποίηση Πολιτικής
Pruning	Κλάδεμα
Q-Learning	Εκμάθηση-Q
Rapid Action Value Estimation	Ταχεία Εκτίμηση Αξίας Ενέργειας
Rank	Βαθμός
Rational	Ορθολογιστικός
Rationality	Λογικότητα
Rectified Linear	Διορθωμένη Γραμμική
Recurrent	Αναδρομικός
Reduced	Μειωμένος
Regression	Παλινδρόμηση
Reinforcement Learning	Ενισχυτική Μάθηση
Relative	Σχετικός
Residual	Υπολειμματικός
Reverse Diffusion Process	Αντίστροφη Διαδικασία Διάχυσης
Reward	Ανταμοιβή
Reward Function	Συνάρτηση Ανταμοιβής
Ring	Δακτύλιος
Robust	Ισχυρός
Rolling Horizon	Κυλιόμενος Ορίζοντας
Rollout	Προσομοίωση
Rotation	Περιστροφή
Roulette-Wheel	Ρουλέτα
Safety Network	Δίκτυο Ασφάλειας
Sample	Δείγμα
Sarrogate	Αναπλήρωση
Scramble	Αναδιάταξη
Search Space	Χώρος Αναζήτησης
Secure	Ασφαλής
Selection	Επιλογή
Self-Attention	Αυτοπροσοχή
Sensing Capability	Αισθητήρια Δυνατότητα
Sensor	Αισθητήρας
Sequential	Ακολουθιακός

Shuffle	Ανάμειξη
Sigmoid	Σιγμοειδής
Similarity	Ομοιότητα
Simple Reflex Agent	Πράκτορας Απλής Αντανάκλασης
Single-Agent	Ενός Πράκτορα
Single-Point	Ενός Σημείου
Soft Actor-Critic	Ελαστικός Δράστης-Κριτής
Soft Pruning	Προσωρινό Κλάδεμα
Sprite	Χαρακτήρας
State	Κατάσταση
State-Value Function	Συνάρτηση Αξίας Κατάστασης
Static	Στατικός
Stochastic	Στοχαστικός
Stochastic Gradient Descent	Στοχαστική Κάθοδος Κλίσης
Strategy	Στρατηγική
Stream	Ροή
Subsymbolic	Υποσυμβολικός
Supervised Learning	Επιβλεπόμενη Μάθηση
Swap	Ανταλλαγή
Symbolic	Συμβολικός
Synthetic	Συνθετικός
Temporal Difference	Χρονική Διαφορά
Test Data	Δεδομένα Ελέγχου
Top-Down	Από Πάνω προς τα Κάτω
Tournament	Διαγωνισμός
Training Data	Δεδομένα Εκπαίδευσης
Translation	Μετατόπιση
Tree Reuse	Επαναχρησιμοποίηση Δέντρου
Trial and Error	Δοκιμή-Λάθος
Truncation	Περικοπή
Uncertain	Αβέβαιος
Uniform	Ομοιόμορφος
Universal	Καθολικός
Unknown	Άγνωστος
Unsupervised Learning	Μη Επιβλεπόμενη Μάθηση
Upper Confidence Bound	Ανώτατο Όριο Εμπιστοσύνης
Utility-based Agent	Πράκτορας βασισμένος στην Ωφελιμότητα
Value-based	Βασισμένος στην Αξία
Value Function	Συνάρτηση Αξίας
Value Iteration	Επανάληψη Αξίας
Vector	Διάνυσμα
View	Προβολή
Video Game	Ηλεκτρονικό Παιχνίδι
Weight	Βάρος

Βιογραφικό Σημείωμα του Συγγραφέα

Ο Τάσος Παπαγιάννης έλαβε το δίπλωμα του Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών από το Εθνικό Μετσόβιο Πολυτεχνείο (ΕΜΠ) το 2017. Στη συνέχεια εκπόνησε διδακτορικές σπουδές στο πεδίο της τεχνητής νοημοσύνης, στο Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών (ΗΜΜΥ) του ΕΜΠ έως τον Απρίλιο του 2024. Σε αυτό το διάστημα δημοσίευσε μία σειρά από ερευνητικά άρθρα σε επιστημονικά περιοδικά και συνέδρια σχετικά ως επί το πλείστον με τεχνικές ανάπτυξης ευφυών πρακτόρων και παρευρέθηκε σε συνέδρια με παρεμφερείς θεματικές ενότητες.

Κατά την παρουσία του στο εργαστήριο, παρακολούθησε την εξέλιξη διπλωματικών εργασιών συναφών με την ερευνητική περιοχή που μελέτησε και παρείχε επικουρικό έργο στα εργαστήρια σχετικών μαθημάτων του τομέα Τεχνολογίας Πληροφορικής και Υπολογιστών της σχολής ΗΜΜΥ ΕΜΠ. Παράλληλα συμμετείχε στα ερευνητικά προγράμματα «Παροχή Συμβουλευτικών Υπηρεσιών για το Σχεδιασμό και Ανάπτυξη Μοντέλου Πρόβλεψης Κυκλοφορίας για την Κυκλοφορία Οχημάτων» με στόχο την εφαρμογή τεχνικών μηχανικής μάθησης για πρόβλεψη της κυκλοφοριακής κίνησης και «Έξυπνες Συστάσεις Τουριστικών Δράσεων Βασισμένες σε Αποδοτική Εξόρυξη Γνώσης από Ηλεκτρονικές Πλατφόρμες» με στόχο την ανάλυση συναισθήματος και την εξαγωγή πληροφοριών από σχόλια σε τουριστικές πλατφόρμες. Τα ερευνητικά του ενδιαφέροντα περιλαμβάνουν εφαρμογές τεχνητής νοημοσύνης, τεχνικές βαθιάς μάθησης και συστήματα ευφυών πρακτόρων.

Κατάλογος Δημοσιεύσεων του Συγγραφέα

Δημοσιεύσεις σχετικές με τη διατριβή

Περιοδικά με κρίση

- Papagiannis Tasos, Georgios Alexandridis, and Andreas Stafylopatis. "Pruning Stochastic Game Trees Using Neural Networks for Reduced Action Space Approximation." Mathematics 10.9 (2022): 1509
- Papagiannis Tasos, Georgios Alexandridis, and Andreas Stafylopatis. "Boosting Deep Reinforcement Learning Agents with Generative Data Augmentation." Applied Sciences 14.1 (2023): 330

Συνέδρια με κρίση

- Papagiannis Tasos, Georgios Alexandridis, and Andreas Stafylopatis. "GAMER: A Genetic Algorithm with Motion Encoding Reuse for Action-Adventure Video Games." Applications of Evolutionary Computation: 22nd International Conference, EvoApplications 2019, Held as Part of EvoStar 2019, Leipzig, Germany, April 24–26, 2019, Proceedings 22. Springer International Publishing, 2019
- Papagiannis Tasos, Georgios Alexandridis, and Andreas Stafylopatis. "Applying gradient boosting trees and stochastic leaf evaluation to MCTS on hearthstone." 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2020
- Papagiannis Tasos, Georgios Alexandridis, and Andreas Stafylopatis. "State similarity based Rapid Action Value Estimation for general game playing MCTS agents." Proceedings of the 17th International Conference on the Foundations of Digital Games. 2022

Δημοσιεύσεις εκτός διατριβής

- Ioannou George, et al. "Visual interpretability analysis of Deep CNNs using an Adaptive Threshold method on Diabetic Retinopathy images." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021

- Papaoikonomou Antonios, et al. "Deep learning techniques for in-core perturbation identification and localization of time-series nuclear plant measurements." Annals of Nuclear Energy 178 (2022): 109373
- Papagiannis Tasos, et al. "Analyzing User Reviews in the Tourism & Cultural Domain-The Case of the City of Athens, Greece." IFIP International Conference on Artificial Intelligence Applications and Innovations. Cham: Springer Nature Switzerland, 2023