ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Μηχανισμοί ενισχυτικής μάθησης και εξελικτικής υπολογιστικής για αυτόνομους πράκτορες

Διδακτορική Διατριβή

που εκπονήθηκε ως μερική εκπλήρωση των απαιτήσεων
για την απονομή του τίτλου του Διδάκτορα Μηχανικού

του
Κυριάκου Χατζηδημητρίου του Χριστόδουλου και της Μαρίας
MSc in Computer Science
Colorado State University
Δίπλωμα Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Επιβλέπων
Περικλής Α. Μήτκας
Καθηγητής

Συμβουλευτική Επιτροπή
Μιχαήλ-Γεράσιμος Στρίντζης — *Ομότιμος Καθηγητής*
Αναστάσιος Ν. Ντελόπουλος — *Αναπληρωτής Καθηγητής*

Υποστηρίχθηκε δημόσια στη Θεσσαλονίκη, Νοέμβριος 2012

Εξεταστική Επιτροπή

.....	.....	.....
Περικλής Α. Μήτκας	Μιχαήλ-Γεράσιμος Στρίντζης	Αναστάσιος Ν. Ντελόπουλος
Καθηγητής	*Ομότιμος Καθηγητής*	*Αναπληρωτής Καθηγητής*
.....	.....	.....
Ιωάννης Θεοχάρης	Βασίλειος Πετρίδης	Ανδρέας Λ. Συμεωνίδης
Καθηγητής	*Καθηγητής*	*Λέκτορας*

.....
Γρηγόριος Τσουμάκας
Λέκτορας

στη Βίκυ

Περίληψη

Απώτερος στόχος της τεχνητής νοημοσύνης είναι η δημιουργία πλήρως αυτόνομων συστημάτων, τα οποία θα μαθαίνουν, θα συλλογίζονται, θα εξελίσσονται και θα λειτουργούν στον πραγματικό κόσμο. Τα συστήματα αυτά, συχνά αναφέρονται με τον όρο αυτόνομοι πράκτορες. Μία από τις πλέον κατάλληλες προσεγγίσεις για τη δημιουργία αυτόνομων πρακτόρων είναι αυτή της ενισχυτικής μάθησης. Οι αλγόριθμοι ενισχυτικής μάθησης είναι μία κλάση τεχνικών με σκοπό την εύρεση μίας πολιτικής, δηλαδή της αντιστοίχησης των ενεργειών ενός πράκτορα με τις καταστάσεις του, χωρίς παραδείγματα βέλτιστης συμπεριφοράς, παρά μόνο θετικές ή αρνητικές ανταμοιβές για τις ενέργειές του, ανάλογες του στόχου που θέλει να επιτύχει. Η βέλτιστη πολιτική θα πρέπει να μεγιστοποιεί την επιβράβευση του αυτόνομου πράκτορα σε βάθος χρόνου. Ένα από τα κύρια συστατικά ενός αλγορίθμου ενισχυτικής μάθησης είναι η συνάρτηση αξίας, η οποία συσχετίζει καταστάσεις ή ζεύγη καταστάσεων-ενεργειών με μία τιμή, που καθορίζει την μακροπρόθεσμη αξία τους για τον πράκτορα. Για μικρά προβλήματα μπορεί να πάρει τη μορφή ενός απλού πίνακα. Πρόθεση της παρούσας διατριβής είναι η δημιουργία πολιτικών για αυτόνομους πράκτορες σε πραγματικά και σύνθετα προβλήματα, με εξαιρετικά μεγάλο αριθμό καταστάσεων. Σε τέτοιου είδους εφαρμογές, κρίνεται συνήθως αναγκαία η παρουσία μίας συνάρτησης σε παραμετρική μορφή, η οποία θα προσπαθεί να προσεγγίσει τη συνάρτηση αξίας και να γενικεύσει από τα ζεύγη καταστάσεων-ενεργειών που έχει συναντήσει στο παρελθόν, ώστε να βοηθήσει τον πράκτορα να λάβει τις σωστές αποφάσεις και για καταστάσεις που δεν έχει αντιμετωπίσει προηγουμένως.

Στόχος της διατριβής είναι η αυτόνομη προσαρμογή συναρτήσεων προσέγγισης με τη χρήση τεχνικών ενισχυτικής μάθησης και εξελικτικής υπολογιστικής. Η προσαρμογή τους πραγματοποιείται ανάλογα με το πρόβλημα, χωρίς να απαιτείται πρότερη λήψη αποφάσεων ως προς το σχεδιασμό τους. Η βασική μέθοδος που αναπτύχθηκε, η NEAR (NeuroEvolution of Augmented Reservoirs), χρησιμοποιεί τρία βασικά συστατικά: α) τα δίκτυα ηχωικών καταστάσεων (ΔΗΚ), ως υπολογιστικό μοντέλο για τις συναρτήσεις προσέγγισης, τα οποία είναι νευρωνικά δίκτυα με αναδράσεις και γραμμικό τρόπο εκμάθησης βαρών, έτσι ώστε να μπορούν να μοντελοποιήσουν και πολιτικές σε μη-γραμμικά περιβάλλοντα, με μη-Μαρκοβιανά σήματα κατάστασης, δηλαδή σε περιβάλλοντα όπου είναι απαραίτητη η ύπαρξη μνήμης, β) τη NEAT (NeuroEvolution of Augmented Topologies), ως μέθοδος μετα-αναζήτησης τοπολογιών και βαρών νευρωνικών δικτύων, προσαρμοσμένη στις ανάγκες των ΔΗΚ, για την εξέλιξη των τοπολογιών των ΔΗΚ και γ) το συνδυασμό εξέλιξης και μάθησης, με απώτερο στόχο την εξέλιξη τοπολογίων ΔΗΚ με αλγορίθμους φυσικής επιλογής, στα οποία η μάθηση είναι πιο αποδοτική. Η εξέλιξη αναζητά στο συνολικό διάστημα των παραμέτρων και αποτελεί τη μακροσκοπική προσέγγιση στο πρόβλημα, ενώ η μάθηση κάνει τοπική βελτιστοποίηση και στοχεύει στη μικροσκοπική βελτίωση του μοντέλου. Πέρα από τη NEAR, αναπτύχθηκε και η επέκτασή του ως προς τεχνικές μεταφοράς μάθησης. Η διαδικασία μεταφοράς μάθησης μεταφέρει τη γνώση που αποκτιέται σε ένα πρόβλημα,

το πηγαίο πρόβλημα (source task), σε ένα άλλο, παρόμοιο, το πρόβλημα στόχου (target task). Σκοπός είναι η βελτίωση της επίδοσης και της ταχύτητας μάθησης του πράκτορα στο τελικό πρόβλημα. Στη μεθοδολογία μεταφοράς μάθησης που αναπτύχθηκε στα πλαίσια της διατριβής, μεταφέρονται τοπολογίες δικτύων που βρέθηκαν στο πηγαίο πρόβλημα ως εμπειρία προς χρήση στο πρόβλημα στόχο.

Η μέθοδος NEAR αξιολογήθηκε σε δέκα (10) διαφορετικές παραλλαγές προβλημάτων ενισχυτικής μάθησης, σε πρόβλεψη τριών (3) προβλημάτων χρονοσειρών δυναμικών συστημάτων και μίας (1) χρονοσειράς ενεργειακού ενδιαφέροντος σε λειτουργία επιβλεπόμενης μάθησης. Από τη σύγκρισή της με ανταγωνιστικούς αλγορίθμους προκύπτει η επικράτηση της NEAR στα περισσότερα από τα παραπάνω προβλήματα. Στη συνέχεια, ΔΗΚ, υπό το πρίσμα της ενισχυτικής μάθησης, χρησιμοποιήθηκαν ως στοιχεία στρατηγικής σε έναν πράκτορα εμπορίου για τη διαχείριση της εφοδιαστικής αλυσίδας, ως στοιχεία μηχανισμού πλειοδοσίας πράκτορα εμπορίου για τη διαχείριση διαδικτυακής διαφημιστικής καμπάνιας, καθώς και ως μοντέλα μικτής στρατηγικής σε πράκτορα για το παιχνίδι του Πόκερ.

Doctoral Dissertation

Title

Reinforcement learning and evolutionary computing mechanisms for autonomous agents

Abstract

The ultimate goal of artificial intelligence is the creation of fully autonomous systems, which will be able to learn, reason, evolve and function in the real world. Such systems are usually referred to as autonomous agents. One of the most appropriate paradigms for creating autonomous agents is that of reinforcement learning. In reinforcement learning problems the goal is to find a policy, a mapping of states to actions, without examples of correct behavior, but only with positive or negative rewards based on the goal the agent is trying to achieve. The optimal policy maximizes the long-term reward of the agent. One of the main ingredients of a reinforcement learning system is the value function, a function that estimates the long-term expected reward for every state or state-action pair. For small-scale problems it can take the form of an array. For larger problems though, the function needs to be represented by a function approximator in parametric form. The reason is the generalization capabilities of the approximator, which will help the agent take correct actions for states that has not encountered before.

The goal of the dissertation is the autonomous adaptation of function approximators with the use of reinforcement learning and evolutionary computing. The algorithm will adapt the parameters of the function approximator to the problem at hand with little or no human input. The name of the method developed is NEAR (NeuroEvolution of Augmented Reservoirs) and uses three basic ideas: a) Echo state networks (ESN), as function approximators, a model of reservoir computing with recursive nature and capabilities of linear learning rules and modeling non-linear environments and non-Markovian state signals, b) NEAT (NeuroEvolution of Augmented Topologies) methodology as a meta-search algorithm, adapted to evolve ESNs and c) evolution coupled with learning with the goal of evolving ESNs that are better able to learn. Evolution performs global-search in the space of parameters, while learning performs local-search for the final tweaking of parameters towards the global optimum. Moreover, a transfer learning procedure was tested in order to transfer knowledge in the form of reservoirs, developed in a source task to a similar target task, with the goal to improve the performance and speed of learning in the target task.

The NEAR method was tested on ten (10) different reinforcement learning testbeds and four (4) time-series prediction problems in a supervised learning mode. NEAR was compared against state-of-the-art algorithms and was found superior in

most of the testbeds. In addition, ESNs and NEAR were tested in three more demanding problems: in the bidding mechanisms of trading agents for supply chain management and advertisement auctions and as a mixed strategy model in the game of Poker.

Kyriakos C. Chatzidimitriou
Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
November 2012

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντά μου, Καθηγητή κ. Περικλή Μήτκα για την εμπιστοσύνη που μου έδειξε τόσο κατά την ανάθεση του θέματος όσο και κατά τη διάρκεια διεκπεραίωσης της διατριβής, καθώς και για τις καίριες υποδείξεις του σε ερευνητικό και επαγγελματικό επίπεδο.

Επίσης, θα ήθελα να ευχαριστήσω τον φίλο και Λέκτορα Ανδρέα Συμεωνίδη, για τις συμβουλές του και για τις εποικοδομητικές συναντήσεις στα πλαίσια της συνεργασίας μας σε ερευνητικά έργα και ομάδες.

Ακόμη, θα ήθελα να ευχαριστήσω τους φίλους και συναδέλφους στην ομάδα ISSEL, Κώστα, Φανή, Φώτη, τους παλαιότερους, Αλέξανδρο, Χρήστο και Σωτήρη, τους νεώτερους, Αντώνη και Μιχάλη, καθώς και τα παιδιά στην ομάδα MUG για την ώσμωση ιδεών και συναισθημάτων.

Επιπλέον, θα ήθελα να ευχαριστήσω, τους Λάμπρο Σταυρογιάννη και Γρηγόρη Αθανασιάδη, για τη συνεργασία μας στην εφαρμογή του πράκτορα διαχείρισης διαφημιστικής καμπάνιας, τον Ιωάννη Σαράφη, για τη συνεργασία μας στην εφαρμογή του παίκτη Πόκερ και τον Ιωάννη Παρτάλα, για την συνεργασία μας στη μεταφορά μάθησης.

Ειδικό ευχαριστώ στους γονείς μου που πάντα είναι εκεί και με στηρίζουν σε όλες τις επιλογές μου.

Τέλος, ιδιαίτερες ευχαριστίες στη γυναίκα μου, Βίκυ, για την αμέριστη συμπαράστασή της και την υπομονή που υπέδειξε στην αναβολή πολλών πραγμάτων για "... μετά το διδακτορικό". Μαζί της τα πράγματα φαίνονται ευκολότερα.

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Κατάλογος Αλγορίθμων

Κεφάλαιο 1

Εισαγωγή

Ο σημαντικότερος στόχος της τεχνητής νοημοσύνης είναι η δημιουργία πλήρως αυτόνομων και λειτουργικών οντοτήτων, είτε σε φυσική μορφή (π.χ. ρομπότ, αυτοκινούμενα οχήματα κτλ.), είτε σε άυλη μορφή (π.χ. λογισμικό), που θα συνυπάρχουν στον πραγματικό κόσμο με τους ανθρώπους και με ομόλογούς τους [Sto07]. Μία από τις βασικότερες μεταφορές για την αναπαράσταση τέτοιου είδους συστημάτων είναι η έννοια του πράκτορα. Ως *πράκτορα* (agent) ορίζουμε ένα σύστημα, που βρίσκεται σε ένα περιβάλλον, το αντιλαμβάνεται και ενεργεί σε αυτό συνεχώς, ακολουθώντας την ατζέντα του η οποία θα καθορίσει τις μελλοντικές του ενέργειες [FG97]. Σύμφωνα με τον παραπάνω ορισμό, πράκτορας μπορεί να θεωρηθεί από ένα απλό σύστημα όπως ένας θερμοστάτης, μέχρι ένα εξαιρετικά περίπλοκο, όπως ο άνθρωπος.

Ένας πλήρως αυτόνομος πράκτορας θα πρέπει να μπορεί να μαθαίνει, να συλλογίζεται και να εξελίσσεται. Ο πράκτορας θα πρέπει να μαθαίνει ώστε να αποδίδει καλύτερα με το χρόνο και να βελτιώνει τις ενέργειές του χωρίς πρότερη γνώση του προβλήματος που καλείται να επιλύσει. Η διαδικασία αυτή ονομάζεται γενίκευση (generalization). Επίσης, θα πρέπει να μπορεί να συλλογίζεται επαγωγικά, δηλαδή να αντιστοιχίζει την εμπειρία του σε μελλοντικές ενέργειές του. Μια τέτοια διαδικασία είναι π.χ. ο περιπτωσιακός συλλογισμός (case-based reasoning). Τέλος, ένας αυτόνομος πράκτορας θα πρέπει να εξελίσσεται έτσι ώστε να μπορεί να ενεργεί, να μαθαίνει και να συλλογίζεται πιο αποδοτικά από “γενιά” σε “γενιά”.

Για την ανάπτυξη ενός τέτοιου πράκτορα πρέπει να συγκεραστούν ερευνητικά αποτελέσματα από διάφορες περιοχές της τεχνητής νοημοσύνης, ώστε να προκύψουν πρακτικές εφαρμογές. Στη συνέχεια, μέσω ενός βρόγχου θετικής ανάδρασης, η πράξη θα πρέπει να τροφοδοτεί τη θεωρία με νέα προβλήματα. Η κατασκευή αυτόνομων πρακτόρων που επιλύουν συγκεκριμένα, πραγματικά προβλήματα θα οδηγήσει στην εμφάνιση άγνωστων ιδιοτήτων και συμπεριφορών των χρησιμοποιούμενων αλγορίθμων, που με τη σειρά τους θα θέτουν στη θεωρητική μοντελοποίηση νέα προβλήματα. Αυτή η προσέγγιση ακολουθήθηκε στην παρούσα διατριβή: συνδυάστηκαν περιοχές της μηχανικής μάθησης και της εξελικτικής υπολογιστικής για τη δημιουργία κατάλληλων εργαλείων, τα οποία, στη συνέχεια, χρη-

σιμοποιούνται σε εφαρμογές αυτόνομων πρακτόρων λογισμικού.

Το παρόν κεφάλαιο ξεκινάει με την περιγραφή του προβλήματος, συνεχίζει με τους στόχους της διατριβής, αγγίζοντας επιδερμικά τη μεθοδολογία που ακολουθήθηκε, και καταλήγει σε μία σύνοψη της συμβολής και των αποτελεσμάτων που παρείχθησαν.

1.1 Περιγραφή του προβλήματος

Υποστηρίζεται [Sto07] ότι μία από τις καταλληλότερες προσεγγίσεις στο πρόβλημα της δημιουργίας πλήρως αυτόνομων οντοτήτων είναι αυτή της ενισχυτικής μάθησης (reinforcement learning) [SB98]. Παράλληλα, μελέτες δείχνουν ότι οι εγκέφαλοι των θηλαστικών χρησιμοποιούν μηχανισμούς μάθησης που παρουσιάζουν συμπεριφορές παρόμοιες με αυτές που παράγονται με την ενισχυτική μάθηση [FDM08]. Ένας αυτόνομος πράκτορας, υπό το πρίσμα της ενισχυτικής μάθησης, προσανατολίζεται στο να μεγιστοποιεί το συνολικό όφελος που αποκομίζει με το χρόνο, βάσει της εμπειροτεχνικής μεθόδου (trial-and-error method), προσαρμόζοντας συνεχώς την πολιτική του, δηλαδή την αντιστοίχιση της κατάστασής του, στις ενέργειες που θα εκτελέσει. Η προσαρμογή της πολιτικής βασίζεται στην αλληλεπίδραση με το περιβάλλον, χωρίς όμως ο πράκτορας να βλέπει παραδείγματα σωστής συμπεριφοράς, όπως γίνεται στην επιβλεπόμενη μάθηση (supervised learning), αλλά λαμβάνοντας απλά, άμεση, θετική ή αρνητική ανταμοιβή, ανάλογα με την κατάσταση στην οποία βρίσκεται τη δεδομένη χρονική στιγμή. Στόχος της προσαρμογής είναι να επιλέξει ενέργειες που θα μεγιστοποιούν το μακροπρόθεσμο προσδοκώμενο όφελος, όπως αυτό έχει ορισθεί (goal-directed agent), βάσει των άμεσων ανταμοιβών. Συνεπώς, η ενισχυτική μάθηση είναι μια υπολογιστική μεθοδολογία βέλτιστου ελέγχου για τη λήψη μίας αλληλουχίας αποφάσεων που θα βοηθήσουν στην επίτευξη ενός στόχου σε βάθος χρόνου. Οποιοσδήποτε αλγόριθμος λύνει τέτοιου είδους προβλήματα θεωρείται αλγόριθμος ενισχυτικής μάθησης.

Ένας από τους βασικούς τρόπους μοντελοποίησης της πολιτικής ενός πράκτορα και ταυτόχρονα ένα από τα κύρια συστατικά ενός συστήματος ενισχυτικής μάθησης είναι η *συνάρτηση αξίας* (value function), η οποία συσχετίζει καταστάσεις ή ζεύγη καταστάσεων-ενεργειών με την μακροπρόθεσμη αξία τους για τον πράκτορα. Στόχος των αλγορίθμων ενισχυτικής μάθησης είναι η εκτίμηση μιας τέτοιας συνάρτησης. Για προβλήματα μικρών διαστάσεων ως προς τον αριθμό καταστάσεων και ενεργειών, η πολιτική μπορεί να μοντελοποιηθεί με μορφή πίνακα αναζήτησης, όπου για κάθε κατάσταση αποθηκεύεται και η αντίστοιχη ενέργεια. Ωστόσο, πρόθεση της διδακτορικής διατριβής είναι η εφαρμογή των παραγόμενων αλγορίθμων σε πραγματικά και σύνθετα προβλήματα, με εξαιρετικά μεγάλο αριθμό ζευγών καταστάσεων και ενεργειών, σε συνεχή πεδία ορισμού, όπου θα ήταν δύσκολη η αποθήκευση, η ακριβής εκτίμηση και η ανανέωση των τιμών ενός τέτοιου πίνακα. Για πολύπλοκα προβλήματα, όπως αυτά που συναντάμε στον πραγματικό κόσμο, με πολυδιάστατους και συνεχείς χώρους καταστάσεων και ενεργειών, προκύπτει η ανάγκη ανάπτυξης *προσεγγιστικών συναρτήσεων* (function approximators) που θα

αναπαριστούν την πολιτική του πράκτορα. Οι συναρτήσεις αυτές φέρουν και την ιδιότητα της *γενίκευσης* (generalization), ώστε να προσεγγίζουν την προσδοκώμενη αξία ζευγών καταστάσεων-ενεργειών, τα οποία ο πράκτορας δεν έχει συναντήσει στο παρελθόν. Συνεπώς, κρίνεται αναγκαία η ύπαρξη μίας συνάρτησης σε παραμετρική μορφή (γραμμική ή μη), η οποία θα προσπαθεί να προσεγγίσει τη συνάρτηση αξίας και θα προσπαθεί να γενικεύσει τα παραδείγματα που είδε στο παρελθόν, με στόχο να βοηθήσει τον πράκτορα να λάβει τις σωστές αποφάσεις ακόμα και για καταστάσεις που δεν έχει επισκεφτεί ποτέ. Χωρίς την ύπαρξη μιας τέτοιας συνάρτησης αλλά και κάποιου μηχανισμού μάθησης, οι μηχανικοί ευφυών συστημάτων θα έπρεπε να προδιαγράψουν εκ των προτέρων όλες τις πιθανές περιπτώσεις τις οποίες θα μπορούσε να αντιμετωπίσει ένα έξυπνο σύστημα καθώς και τις αντίστοιχες ενέργειες που θα έπρεπε να εκτελέσει. Ειδικότερα, στην περίπτωση σχεδίασης ευφυών πρακτόρων για τον πραγματικό κόσμο, πολλές πολιτικές είναι δύσκολο όχι μόνο να καταγραφούν αλλά ακόμα και να προβλεφθούν. Η περιοχή της ενισχυτικής μάθησης συνήθως δανείζεται μηχανισμούς ορισμού των συναρτήσεων αυτών από την περιοχή της επιβλεπόμενης μάθησης. Χαρακτηριστικά παραδείγματα είναι τα νευρωνικά δίκτυα (neural networks), τα δένδρα αποφάσεων (decision trees), οι ακτινικές συναρτήσεων βάσης (radial basis functions) αλλά και οι αρθρωτοί ελεγκτές μοντελοποίησης παρεγκεφαλίδας (cerebellar model articulator controllers or CMAC) [SB98].

Κατά τη διαδικασία σχεδίασης των συναρτήσεων προσέγγισης απαιτούνται κατάλληλες επιλογές, τόσο στην αρχιτεκτονική τους, όσο και στις παραμέτρους τους, προκειμένου να έχουν υψηλές επιδόσεις. Οι απαιτήσεις του προβλήματος μετατοπίζονται στη σωστή επιλογή της μορφής των παραμέτρων και στην εύρεση των βέλτιστων τιμών τους, ώστε να επιτυγχάνεται η καλύτερη δυνατή σύγκλιση στη βέλτιστη συνάρτηση αξίας για το συγκεκριμένο πρόβλημα. Η εφαρμογή τους σε πράκτορες λογισμικού, είναι μια επίπονη διαδικασία που εκτελείται από ειδικούς στο πεδίο εφαρμογής. Το πρόβλημα συνίσταται στο γεγονός ότι ο άνθρωπος καλείται να σχεδιάζει το σύστημα για κάθε πρόβλημα ξεχωριστά, καθώς μέχρι στιγμής είναι λίγες οι πραγματικές εφαρμογές που υλοποιούνται με πράκτορες λογισμικού. Υποστηρίζεται ότι τροχοπέδη στην προσπάθεια αυτή είναι η ανυπαρξία γενικευμένων εργαλείων ανάπτυξης πρακτόρων λογισμικού, σε αντίθεση με π.χ. εργαλεία ανάλυσης δεδομένων [Sto07]. Τέτοιου είδους εργαλεία θα ελευθέρωναν το χρήστη από σχεδιαστικές αποφάσεις που σχετίζονται με τους μηχανισμούς απόφασης των πρακτόρων και των αλγορίθμων μάθησης, καθώς οι τελευταίοι θα αντιμετωπίζονταν ως “μαύρα κουτιά”, δίνοντας ταυτόχρονα τη δυνατότητα αυτόματης βελτιστοποίησής τους στο εκάστοτε πρόβλημα. Υπό αυτό το πρίσμα και έχοντας υπόψιν ότι βασικός σκοπός της επιστήμης των υπολογιστών είναι η αυτοματοποίηση των εργασιών και η γενίκευση των λύσεων σε προβλήματα, αναπτύχθηκε η περιοχή των προσαρμοζόμενων προσεγγιστικών συναρτήσεων [Sto07]. Σκοπός της περιοχής είναι η δημιουργία τεχνικών όπου τόσο η αρχιτεκτονική όσο και οι παράμετροι των προσεγγιστικών συναρτήσεων θα προσαρμόζονται ανάλογα με το πρόβλημα προς επίλυση με μικρή ή καθόλου βοήθεια από τον ανθρώπινο παράγοντα. Η προσαρμογή θα βασίζεται στη συνέργεια της μάθησης (το τοπικό κομμάτι

της βελτιστοποίησης) και της εξέλιξης (ολική, στοχαστική αναζήτηση) με σκοπό να ανακαλύψουμε συναρτήσεις προσέγγισης που θα μαθαίνουν καλύτερα [WS06], είτε μέσω μάθησης χρονικών διαφορών (temporal difference learning), είτε μέσω αναζήτησης πολιτικών (policy search).

1.2 Στόχοι και μεθοδολογία της διατριβής

Ο βασικός στόχος της διδακτορικής διατριβής είναι η δημιουργία προσαρμοζόμενων προσεγγιστικών συναρτήσεων (adaptive function approximators) με τη χρήση τεχνικών ενισχυτικής μάθησης και εξελικτικής υπολογιστικής. Οι προσεγγιστικές συναρτήσεις θα προσαρμόζονται σύμφωνα με το εκάστοτε πρόβλημα, χωρίς να απαιτείται πρότερη λήψη αποφάσεων ως προς το σχεδιασμό τους. Θεωρητικά, η προσαρμογή προσεγγιστικών συναρτήσεων είναι ένας τρόπος προσθήκης της έννοιας της μακροβιότητας στους πράκτορες, όπου η συνάρτηση προσαρμόζεται ανάλογα με το εκάστοτε πρόβλημα που αντιμετωπίζει ο πράκτορας. Με άξονα τον παραπάνω στόχο, παρουσιάζεται μία μεθοδολογία που συνθέτει με καινοτόμο τρόπο, αυτοτελή ερευνητικά αποτελέσματα διαφόρων περιοχών της μηχανικής μάθησης και της υπολογιστικής νοημοσύνης. Η μεθοδολογία προσαρμοζόμενων προσεγγιστικών συναρτήσεων που αναπτύχθηκε υπό το ακρωνύμιο NEAR (neuroevolution of augmented reservoirs) αποτελείται από τρία συστατικά: α) τη συνάρτηση προσέγγισης, β) τη σύζευξη μάθησης και εξέλιξης και γ) τη μέθοδο νευροεξέλιξης.

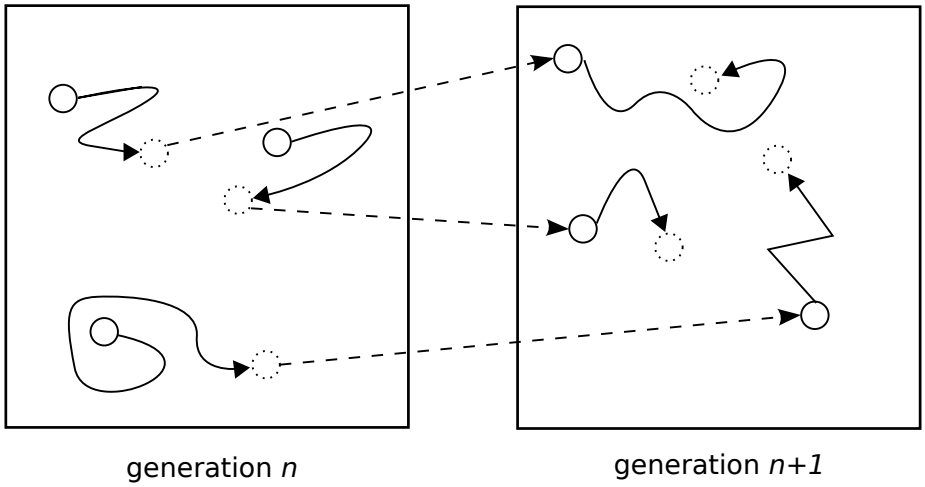
Σχετικά με το πρώτο συστατικό της μεθοδολογίας, η συνάρτηση προσέγγισης που επιλέξαμε ήταν τα δίκτυα ηχωικών καταστάσεων (echo state networks), από την περιοχή της υπολογιστικής ταμιευτήρων νευρώνων (reservoir computing). Δεδομένου του προφίλ τους ως νευρωνικά δίκτυα με αναδράσεις (recurrent neural networks), τα δίκτυα ηχωικών καταστάσεων [Jae01, Jae02] αποτελούν μια ταιριαστή λύση στα προβλήματα ενισχυτικής μάθησης που αντιμετωπίζει ένας πράκτορας. Τα δίκτυα ανάδρασης μπορούν α) να προσεγγίσουν μη γραμμικές (non-linear) συναρτήσεις, μία απαίτηση απαραίτητη όταν έχουμε να αντιμετωπίσουμε προβλήματα στον πραγματικό κόσμο, και β) να μοντελοποιήσουν, εάν υπάρχουν, μη-Μαρκοβιανά σήματα καταστάσεων (non-Markovian state signals)[1]. Παράλληλα, το δυναμικό μη-γραμμικό σύστημα που αναπτύσσεται στον ταμιευτήρα του δικτύου μπορεί να λειτουργήσει ως είσοδος σε πολλαπλές συναρτήσεις προσέγγισης στο επίπεδο εξόδου του δικτύου (multiple read-out functions), μια χρήσιμη ιδιότητα για τους πράκτορες, όταν στο σύστημα με το οποίο αλληλεπιδρούν, υπάρχει η ανάγκη δημιουργίας πολλαπλών πολιτικών για ταυτόχρονες αλλά συσχετισμένες εργασίες. Επιπρόσθετα, δυσκολίες που προκύπτουν εξαιτίας διακλαδώσεων (bifurcations), μαθησιακές ή δυσκολίες μνήμης μακράς διάρκειας καθώς και υλοποίησης, συχνά αναφερόμενες ως μειονεκτήματα των δικτύων ανάδρασης,

[1] Στην παρούσα διατριβή η έννοια του μη-Μαρκοβιανού σήματος κατάστασης σημαίνει ότι στο σήμα εισόδου του πράκτορα δεν υπάρχει η πληροφορία για να μπορέσει ο πράκτορας να προβλέψει την επόμενη κατάσταση δεδομένης της ενέργειάς του, αλλά χρειάζεται και την πληροφορία των σημάτων εισόδου παρελθόντων βημάτων [SB98].

αντικαθίστανται από τα πλεονεκτήματα των δικτύων ηχωικών καταστάσεων, όπως ακρίβεια στη μοντελοποίηση, χωρητικότητα, βιολογική αληθοφάνεια, επεκτασιμότητα και λακωνικότητα [LJ09]. Αν και τα δίκτυα ηχωικών καταστάσεων μπορούν να χειριστούν μη γραμμικά περιβάλλοντα, η μάθηση μπορεί να πραγματοποιηθεί, ακόμα και κατά τη διάρκεια λειτουργίας του πράκτορα, χρησιμοποιώντας γραμμικούς κανόνες μάθησης και μόνο στο επίπεδο εξόδου του δικτύου. Αντίθετα, η δημιουργία δικτύων ηχωικών καταστάσεων με τυχαίο τρόπο, όπως αρχικά είχε προταθεί, πολλές φορές είναι αναποτελεσματική. Διαφαίνεται λοιπόν η ανάγκη για συγκεκριμένες τοπολογίες ταμιευτηρίων, προσαρμοζόμενες στο τρέχον πρόβλημα, που εικάζεται ότι θα φέρουν καλύτερα αποτελέσματα [Pro05, LJ09]. Υπό το πρίσμα αυτό, τα δίκτυα ηχωικών καταστάσεων, φαντάζουν πανίσχυρα και προβάλλουν ως ιδανικοί υποψήφιοι για διττή προσαρμογή, τόσο μέσω μάθησης (στο επίπεδο εξόδου του δικτύου), όσο και μέσω εξέλιξης (στο επίπεδο της τοπολογίας και βαρών του ταμιευτηρίου του δικτύου).

Το βασικό επιχείρημα για την επιλογή της σύνθεσης μάθησης και εξέλιξης για την προσαρμογή προσεγγιστικών συναρτήσεων είναι η συμπληρωματικότητα των δύο μεθόδων. Η δομή παίζει κρίσιμο ρόλο στη μαθησιακή συμπεριφορά, είτε βιολογικών, είτε τεχνητών νευρικών συστημάτων [IS08]. Επομένως, μπορούμε να εφαρμόσουμε αλγορίθμους εξελικτικής υπολογιστικής για τη δομική προσαρμογή των δικτύων νευρώνων, ειδικά εφόσον απουσιάζουν "κλασικές" μέθοδοι στον τομέα αυτό. Επιπλέον, με τις μεθόδους εξελικτικής υπολογιστικής είναι ευκολότερο να βελτιστοποιηθούν μετα-παράμετροι, όπως για παράδειγμα, ρυθμοί μάθησης, πυκνότητες δικτυωμάτων κτλ., ή ακόμα και συναρτήσεις προσέγγισης που δεν είναι διαφορίσιμες [Mii11]. Βέβαια, θα μπορούσε κάποιος να παρουσιάσει αυτή την άμεση αναζήτηση μέσω εξέλιξης στο χώρο των πολιτικών, ως αναζήτηση "βελόνας στα άχυρα" [HN87]. Τα δίκτυα ηχωικών καταστάσεων φαντάζουν ιδανικά για προσαρμογή, τόσο μέσω εξέλιξης, όσο και μέσω μάθησης. Τα πλεονεκτήματα της εξέλιξης στοιχειοθετούνται στη στοχαστικότητα της εξελικτικής αναζήτησης, μια έμφυτη ιδιότητα αυτού του τύπου δικτύων, αφού η προτεινόμενη χρήση τους είναι μέσω τυχαία δημιουργημένων ταμιευτηρίων. Ακόμα, οι εξελικτικές μέθοδοι μπορούν να χειριστούν τη βελτιστοποίηση των μετα-παραμέτρων των δικτύων, όπως η πυκνότητα (density) ή η φασματική ακτίνα (spectral radius) του ταμιευτηρίου. Από την άλλη, υπέρ της μάθησης συνηγορούν η διαφορισιμότητα στο επίπεδο εξόδου των δικτύων καταστάσεων ηχούς, γεγονός που επιτρέπει τη χρήση "κλασικών" γραμμικών μεθόδων βελτιστοποίησης των βαρών, προκειμένου να πραγματοποιηθούν οι κατάλληλες αλλαγές προς τα τοπικά ελάχιστα. Με άλλα λόγια, η εξέλιξη αναζητεί δομές που επιτρέπουν α) στη μάθηση να συγκλίνει γρηγορότερα σε τοπικά ελάχιστα και β) στις δύο μαζί να συγκλίνουν ταχύτερα στο ολικό ελάχιστο. Η ιδέα αυτή παρουσιάζεται στο Σχήμα 1.1.

Τέλος, το τρίτο συστατικό της μεθοδολογίας NEAR είναι η νευροεξέλιξη αυξανόμενων τοπολογιών (neuro-evolution of augmented topologies or NEAT), μια μέθοδος αναφοράς νευροεξέλιξης, ικανή να εξελίσσει νευρωνικά δίκτυα σε επίπεδο βαρών και τοπολογίας (topology and weight evolution of artificial neural networks or TWEANN). Η μέθοδος NEAT [Sta04, SM02] χρησιμοποιήθηκε ως ένας αλ-



Σχήμα 1.1: Συνδυασμός εξέλιξης και μάθησης. Οι κύκλοι με τις συμπαγείς γραμμές αναπαριστούν τον πληθυσμό των δικτύων της μεθόδου νευροεξέλιξης, απλωμένα στο χώρο αναζήτησης. Μέσω της μάθησης, χρησιμοποιώντας για παράδειγμα τη μέθοδο κατάβασης πλαγιάς (gradient descent), τα δίκτυα προσαρμόζονται προς περιοχές μικρότερου σφάλματος (διακεκομμένοι κύκλοι). Μέσω της εξέλιξης (Λαμαρκιανής εξέλιξης στη συγκεκριμένη περίπτωση), μετακινούνται σε μια άλλη περιοχή στο χώρο αναζήτησης, όπου, πιθανώς, η γειτονιά θα περιέχει χαμηλότερα τοπικά ελάχιστα τα οποία θα ανακαλύψει η διαδικασία της μάθησης.

γόριθμος μετα-αναζήτησης για να εξελίξει δίκτυα ηχωικών καταστάσεων, ώστε να μάθουν να λύνουν το δεδομένο πρόβλημα αποδοτικότερα. Συγκεκριμένα, έγινε μετασχηματισμός των βασικών ιδεών στις οποίες βασίζεται η μεθοδολογία NEAT στις ιδιαιτερότητες των δικτύων ηχωικών καταστάσεων. Οι ιδέες αυτές είναι: α) χρήση διασταύρωσης (crossover) με τη βοήθεια ιστορικών προσημειώσεων, β) δημιουργία ειδών (species) για την προστασία καινοτομιών και γ) διαδοχική αύξηση της πολυπλοκότητας της προσεγγιστικής συνάρτησης, ξεκινώντας μινιμαλιστικά και διογκώνοντας τις τοπολογίες για την εύρεση των πιο σύντομων αναπαραστάσεων δικτύων. Η επιλογή της NEAT ως μηχανισμού μετα-αναζήτησης προέκυψε καθώς η εισαγωγή της έλυσε αναγνωρισμένα προβλήματα της νευροεξέλιξης, όπως η πρόωρη σύγκλιση και οι ανταγωνιστικές συμβάσεις. Επομένως, η προσαρμογή της NEAT στα δίκτυα ηχωικών καταστάσεων, τα οποία με τη σειρά τους δύναται να προσαρμοστούν με αλγορίθμους μάθησης γραμμικών συστημάτων και να υπολογίσουν χρονικά χαρακτηριστικά (temporal features) δυναμικών συστημάτων, δημιουργούν έναν ισχυρό συνδυασμό.

Συνολικά μπορεί κανείς να τοποθετήσει τη μέθοδο NEAR στην περιοχή της γενετικά βασισμένης μηχανικής μάθησης. Πιο συγκεκριμένα, πρόκειται για μία μετα-μαθησιακή μέθοδο που βελτιστοποιεί έναν βασικό μαθησιακό μοντέλο, όπως θεωρούμε τους ταμιευτήρες νευρώνων.

Πέραν του βασικού στόχου και της μεθοδολογίας επίτευξης αυτού, στη διατρι-

βή έχουν δεικτοδοτηθεί και επιμέρους στόχοι. Πιο συγκεκριμένα:

Α Μελέτη δυνατότητας μετατροπής τεχνικών μεταφοράς μάθησης στους ταμιευτήρες νευρώνων και πιο συγκεκριμένα στη μεθοδολογία NEAR. Στη μεταφορά μάθησης, σκοπός είναι να μεταφέρουμε τη γνώση που αποκτάται σε ένα πρόβλημα, το οποίο ονομάζουμε πηγαία εργασία ή πρόβλημα (source task) σε ένα άλλο, παρόμοιο, το οποίο ονομάζουμε εργασία ή πρόβλημα στόχο (target task). Σκοπός είναι να βελτιώσουμε την επίδοση του πράκτορα και την ταχύτητα μάθησης στο τελικό πρόβλημα. Ουσιαστικά μεταφέρονται τοπολογίες και βάρη δικτύων που ανακαλύφθηκαν στην πηγαία εργασία ως γνώση προς χρήση στην εργασία στόχο. Η επίτευξη ενός τέτοιου στόχου θα βοηθούσε τόσο στην αυτονομία, όσο και στη μακροβιότητα ενός πράκτορα (life long learning).

Β Επαλήθευση της μεθοδολογίας που σχεδιάστηκε. Ως κύριο πεδίο εφαρμογής ορίζονται προβλήματα στα οποία χρειάζεται να υπάρχει μνήμη ή να λαμβάνονται διαδοχικές αποφάσεις, όπως π.χ. συμβαίνει στην ενισχυτική μάθηση και στην πρόβλεψη χρονοσειρών με αυτο-παλινδρομική μοντελοποίηση (auto-regressive). Τόσο η ενισχυτική μάθηση όσο και η επιβλεπόμενη μάθηση είναι απαραίτητα εργαλεία σε έναν πράκτορα, με το πρώτο να βοηθά στη λήψη αποφάσεων και το δεύτερο στη μοντελοποίηση του περιβάλλοντος. Για το λόγο αυτό είναι απαραίτητος ο ενδελεχής έλεγχος της μεθοδολογίας ως προς τα δύο αυτά σενάρια, με συγκρίσεις με αλγορίθμους αναφοράς και στατιστική αξιολόγηση. Στην παρούσα διατριβή η μεθοδολογία NEAR εφαρμόστηκε στα παρακάτω προβλήματα:

Β.1 Πρόβλεψη χρονοσειρών:

- Μη γραμμικών δυναμικών συστημάτων
- Ενεργειακού ενδιαφέροντος

Β.2 Κλασσικά προβλήματα ελέγχου υπό το πρίσμα της ενισχυτικής μάθησης

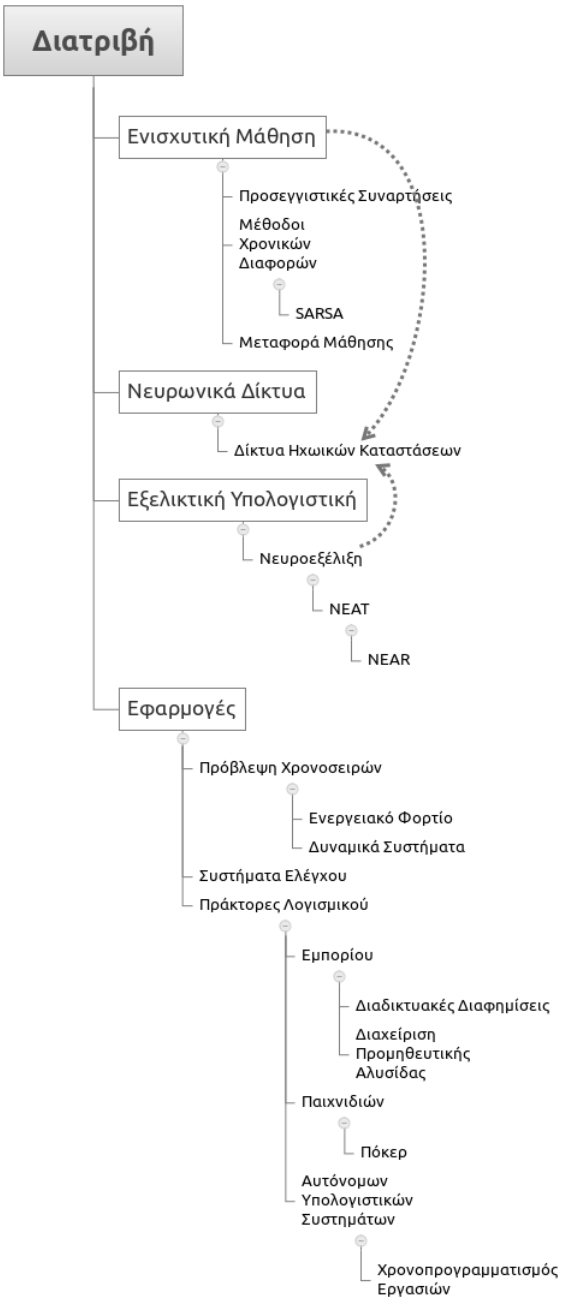
Β.3 Τμήματα αυτόνομων πρακτόρων λογισμικού:

- Πράκτορας εμπορίου για τη διαχείριση της εφοδιαστικής αλυσίδας
- Πράκτορας εμπορίου για τη διαχείριση διαδικτυακής διαφημιστικής καμπάνιας
- Πράκτορας για την παραλλαγή του παιχνιδιού πόκερ με την ονομασία heads-up limit Texas hold'em.

Στο Σχήμα 1.2 απεικονίζονται οι περιοχές και οι συσχετίσεις τους, στις οποίες αναφέρεται η παρούσα διατριβή.

1.3 Συμβολή της διατριβής

Το σημαντικότερο αποτέλεσμα της διατριβής είναι η μέθοδος NEAR, όπως αυτή σχεδιάστηκε, υλοποιήθηκε και ελέγχθηκε. Η μέθοδος προσπαθεί να κάνει τις λι-



Σχήμα 1.2: Οι περιοχές και οι συσχετίσεις της διατριβής.

γότερες δυνατές παραδοχές για το πρόβλημα που καλείται να αντιμετωπίσει. Σχεδιάστηκε βάσει απαιτήσεων που ορίστηκαν από την αρχή, όπως η ικανότητα μοντελοποίησης μη-γραμμικών και μη-Μαρκοβιανών συστημάτων και καταστάσεων, η αυτονομία προσαρμογής για την εύρεση της βέλτιστης τοπολογίας ανεξαρτήτως του προβλήματος, και η ύπαρξη "κλασικών" και θεωρητικά θεμελιωμένων μεθόδων μάθησης που θα προσαρμόζουν τα βάρη των συνάψεων των νευρώνων εκ των προτέρων ή και κατά τη διάρκεια λειτουργίας του πράκτορα. Ο έλεγχος της μεθοδολογίας έγινε με την εφαρμογή της NEAR σε πολλά διαφορετικά προβλήματα, τόσο ενισχυτικής, όσο και επιβλεπόμενης μάθησης και με τη σύγκρισή της με ανταγωνιστικούς αλγορίθμους, με αποτέλεσμα σε μεγάλο ποσοστό εξ αυτών να αποδειχθεί καλύτερη. Επιπλέον, η μέθοδος εφαρμόστηκε και σε προβλήματα μεγαλύτερης κλίμακας ως μέλος τμημάτων πρακτόρων λογισμικού, επιδεικνύοντας ικανοποιητικά αποτελέσματα. Τέλος, η μεταφορά μάθησης εφαρμόστηκε με επιτυχία καθώς προσάρμοσε τη συμπεριφορά του πράκτορα γρηγορότερα και αποδοτικότερα.

Παράλληλα, στα πλαίσια της διατριβής δημιουργήθηκε και η βιβλιοθήκη ανοικτού λογισμικού *fiterr*. Η βιβλιοθήκη αυτή επεκτείνει τη βιβλιοθήκη *rl-glue* [TW09], ώστε να υλοποιεί και πράκτορες που βασίζονται στη νευροεξέλιξη. Στη βιβλιοθήκη περιλαμβάνονται διάφορα προβλήματα ενισχυτικής μάθησης και χρονοσειρών, καθώς και διάφοροι αλγόριθμοι και μοντέλα προσεγγιστικών συναρτήσεων όπως τα δίκτυα ηχωικών καταστάσεων και οι μέθοδοι NEAT και NEAR.

Τμήματα της παρούσας διατριβής έχουν δημοσιευθεί σε διεθνή περιοδικά ή έχουν ανακοινωθεί σε συνέδρια, βοηθώντας έτσι στην επικύρωση της μεθοδολογίας και των αποτελεσμάτων μέσω του συστήματος κριτών. Πιο συγκεκριμένα:

1. τρεις δημοσιεύσεις σε περιοδικά με σύστημα κριτών: [CM12, CS09, CSKM08]
2. επτά δημοσιεύσεις σε πρακτικά διεθνών συνεδρίων με σύστημα κριτών: [CSM12, CPMV11, CCSM11, CSSM11, TCM11, CM10, CSM08, KCSM06]
3. δύο ανακοινώσεις σε διεθνή συνέδρια χωρίς πρακτικά με σύστημα κριτών: [CPM10, DCSM08]

Παράλληλα, οι πράκτορες λογισμικού για τις διαδικτυακές διαφημίσεις και τη διαχείριση της εφοδιαστικής αλυσίδας λαμβάνουν μέρος κάθε χρόνο στο διεθνή διαγωνισμό για πράκτορες εμπορίου (trading agent competition or TAC). Το 2012 ο πράκτορας Mertacor κατέλαβε την πρώτη θέση στον διαγωνισμό TAC Ad Auctions Game 2012, ενώ παλαιότερα είχε λάβει την 3η θέση στους διαγωνισμούς TAC Ad Auctions 2010 και TAC Supply Chain Management 2005. Στην παρούσα διατριβή παρουσιάζονται βελτιωμένες εκδόσεις των πρακτόρων στις οποίες συμπεριλαμβάνεται και η μέθοδος NEAR.

1.4 Διάρθρωση της Διατριβής

Η διατριβή αποτελείται από 10 κεφάλαια, μαζί με το παρόν εισαγωγικό. Εννοιολογικά χωρίζεται σε δύο μέρη: στο θεωρητικό μέρος (κεφάλαια 2 έως 5), όπου

παρουσιάζονται οι αλγόριθμοι και η αξιολόγησή τους, και στο μέρος των εφαρμογών (κεφάλαια 6 έως 9), όπου η μεθοδολογία NEAR εφαρμόζεται σε διάφορα προβλήματα. Αναλυτικότερα:

Το Κεφάλαιο 2 περιλαμβάνει μια περίληψη των περιοχών της μηχανικής μάθησης και της υπολογιστικής ευφυΐας τις οποίες πραγματεύεται η παρούσα διατριβή, δίνοντας έμφαση σε συγκεκριμένα αντικείμενα προκειμένου να δημιουργηθεί το κατάλληλο υπόβαθρο στον αναγνώστη για την ευκολότερη ανάγνωση του κειμένου. Παράλληλα, παραθέτει τη σχετική βιβλιογραφία που εμπίπτει στο πλαίσιο της μεθοδολογίας NEAR, καθώς και τον τρόπο που αυτή διαφοροποιείται από τις υπόλοιπες προσεγγίσεις.

Στο Κεφάλαιο 3 παρουσιάζεται η μέθοδος NEAR και δίνονται όλες οι πληροφορίες για την υλοποίησή της με τη μορφή εξισώσεων, σχημάτων και αλγορίθμων. Επίσης, γίνεται αναλυτική καταγραφή των ομοιοτήτων και των διαφορών με τη βασική μέθοδο αναφοράς NEAT.

Στο Κεφάλαιο 4 η μεθοδολογία αξιολογείται σε μία ευρεία γκάμα προβλημάτων ενισχυτικής μάθησης και συγκρίνεται με αλγορίθμους αναφοράς. Η αξιολόγηση περιλαμβάνει ελέγχους στατιστικής σημαντικότητας καθώς και μελέτες αποκοπής τμημάτων του αλγορίθμου προς επικύρωση των σχεδιαστικών επιλογών.

Το Κεφάλαιο 5 ασχολείται με μια επέκταση της μεθόδου NEAR στην περιοχή της μεταφοράς μάθησης. Πιο συγκεκριμένα, ερευνά τρόπους ώστε τα δίκτυα που αναπτύσσονται σε κάποιο πρόβλημα να μεταφέρονται σε άλλο παραπλήσιο πρόβλημα ώστε να επιτυγχάνεται καλύτερη και ταχύτερη εύρεση λύσεων.

Από το Κεφάλαιο 6 ξεκινάει το δεύτερο μέρος της διατριβής με την εφαρμογή της μεθόδου NEAR σε περισσότερο απαιτητικά προβλήματα. Στο κεφάλαιο αυτό ο αλγόριθμος προσαρμόζει την τοπολογία και τα βάρη των δικτύων του πληθυσμού του, προκειμένου να βελτιώσει τις προβλέψεις του σε προβλήματα χρονοσειρών. Πιο συγκεκριμένα μελετούνται οι χρονοσειρές Mackey-Glass, Lorentz attractor, Multiple-Superimposed Oscillator, αλλά και η πρόβλεψη του συνολικού φορτίου ενέργειας με πραγματικά δεδομένα από τον Ελληνικό χώρο.

Στο Κεφάλαιο 7 η μεθοδολογία NEAR προσαρμόζει δίκτυα ηχωικών καταστάσεων για την πρόβλεψη των τιμών κλεισίματος δημοπρασιών προμηθειών ηλεκτρονικών υπολογιστών και αποτελεί μέρος ενός συστήματος βελτιστοποίησης πολλαπλών συσχετιζόμενων προσφορών πραγματικού χρόνου.

Το Κεφάλαιο 8 παρουσιάζει τον πράκτορα λογισμικού Mertacor ο οποίος έχει το ρόλο πλειοδότη σε δημοπρασίες διαδικτυακών διαφημίσεων. Πιο συγκεκριμένα, η μέθοδος NEAR χρησιμοποιείται για την ανάπτυξη δικτύων προκειμένου ο πράκτορας να βελτιστοποιήσει τις προσφορές του πράκτορα και να εξασφαλίσει διαφημιστικό χώρο στα οργανικά αποτελέσματα μίας μηχανής αναζήτησης.

Το δεύτερο μέρος της διατριβής κλείνει με το Κεφάλαιο 9, όπου δίκτυα ταμιευτήρων νευρώνων προσαρμόζονται μέσω διαδικασιών μάθησης από δεδομένα, προκειμένου να αναπτύξουν μικτές στρατηγικές (mixed strategies) για μια συγκεκριμένη παραλλαγή του πόκερ, το heads-up limit Texas hold'em.

Τέλος, στο Κεφάλαιο 10, μετά από μια σύνοψη της έρευνας που διατελέσθηκε στα πλαίσια της διδακτορικής διατριβής, προβάλλονται τα συμπεράσματα που

προέκυψαν, ενώ παράλληλα δίνονται οι πολλαπλές μελλοντικές επεκτάσεις που ανοίγει η παρούσα διατριβή και οι οποίες αποτελούν πεδία ενδιαφέρουσας έρευνας στο αντικείμενο.

Η αναφορά καταλήγει με το Παράρτημα A, στο οποίο αναφέρονται οι τιμές των παραμέτρων που χρησιμοποιήθηκαν για τη μεθοδολογία NEAR προκειμένου κάθε ενδιαφερόμενος να μπορεί να επαναλάβει την πειραματική διαδικασία. Στο τέλος της διατριβής παρατίθεται η Βιβλιογραφία.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο και βιβλιογραφική επισκόπηση

Το παρόν κεφάλαιο παρέχει το βασικό θεωρητικό υπόβαθρο των περιοχών πάνω στις οποίες αναπτύχθηκε ο υπό εξέταση μηχανισμός. Στην πρώτη ενότητα παρουσιάζονται οι βασικές αρχές της *ενισχυτικής μάθησης* (reinforcement learning), μιας κατηγορίας προβλημάτων μάθησης όπου ο πράκτορας μαθαίνει να λαμβάνει αποφάσεις βάσει των ανταμοιβών που προκύπτουν από τις ενέργειες που εκτελεί. Στη συνέχεια, παρουσιάζονται οι βασικές έννοιες στην περιοχή της *εξελικτικής υπολογιστικής* (evolutionary computing), δηλαδή αλγορίθμων στοχαστικής έρευνας βασισμένων σε πληθυσμούς (stochastic search population based algorithms) με ιδέες που εντοπίζονται στις αρχές της φυσικής επιλογής και εξέλιξης. Στην παρούσα διατριβή μεθοδολογίες από αυτές τις δύο ευρύτερες ερευνητικές περιοχές συνδυάζονται με το υπολογιστικό μοντέλο των *υπολογιστικών ταμιευτηρίων* (reservoir computing) και ειδικότερα με τα *Δίκτυα Ηχωικών Καταστάσεων (ΔΗΚ)*, μέσω της *νευροεξέλιξης* (neuroevolution) και του υβριδικού συνδυασμού μάθησης και εξέλιξης, με αποτέλεσμα τη μεθοδολογία *NeuroEvolution of Augmented Reservoirs (NEAR)*, που θα παρουσιαστεί στο επόμενο κεφάλαιο. Στις τελευταίες ενότητες του παρόντος κεφαλαίου παρουσιάζεται η μεθοδολογία *NeuroEvolution of Augmented Topologies (NEAT)*, που αποτέλεσε και τη βασική πηγή έμπνευσης της NEAR, καθώς και η παρουσίαση της σχετικής βιβλιογραφίας σχετικά με τη βελτιστοποίηση των ΔΗΚ.

2.1 Ενισχυτική μάθηση

Μία από τις καταλληλότερες προσεγγίσεις στο πρόβλημα της δημιουργίας πλήρως αυτόνομων οντοτήτων είναι η ενισχυτική μάθηση [SB98, KLM96]. Ο αναγνώστης μπορεί να ανατρέξει στις δύο αυτές αναφορές για ορισμούς εννοιών που χρησιμοποιούνται στην παρούσα ενότητα. Η ενισχυτική μάθηση είναι μια κλάση προβλημάτων με στόχο την εύρεση μίας βέλτιστης *πολιτικής* (policy), που αντιστοιχεί την κατάσταση του πράκτορα σε ενέργειες που μπορεί να εκτελέσει στη δεδομέ-

νη κατάσταση. Κατά την ενισχυτική μάθηση, η πολιτική καταστρώνεται από τον πράκτορα ύστερα από άμεση αλληλεπίδραση με το περιβάλλον, χωρίς όμως ο ίδιος να "βλέπει" παραδείγματα σωστής συμπεριφοράς (όπως στη μάθηση με επίβλεψη - supervised learning), αλλά μόνο με θετική ή αρνητική ανταμοιβή, ανάλογη με το στόχο που έχει τεθεί στην αρχή (goal-directed agent). Η βέλτιστη πολιτική του θα πρέπει να μεγιστοποιεί την επιβράβευσή του σε βάθος χρόνου, ακόμα και αν η επίτευξη του στόχου δεν είναι άμεση. Με άλλα λόγια, η ενισχυτική μάθηση είναι μια υπολογιστική μεθοδολογία βέλτιστου ελέγχου για αλληλουχία αποφάσεων με στόχο την μακροπρόθεσμη επίτευξη ενός στόχου. Οποιοσδήποτε αλγόριθμος λύνει τέτοιου είδους προβλήματα, μπορεί να θεωρείται αλγόριθμος ενισχυτικής μάθησης.

Σε κάθε χρονική στιγμή t, ο πράκτορας δέχεται ως είσοδο μια αναπαράσταση της κατάστασης του περιβάλλοντος, $s_t \in S$, όπου S το σύνολο όλων των πιθανών καταστάσεων που μπορεί να αντιμετωπίσει ο πράκτορας, και επιλέγει μια ενέργεια $a_t \in A(s_t)$, όπου $A(s_t)$ το σύνολο των ενεργειών που μπορεί να εκτελέσει ο πράκτορας όταν βρίσκεται στην κατάσταση s_t. Την επόμενη χρονική στιγμή ο πράκτορας λαμβάνει την άμεση ανταμοιβή r_{t+1} και μεταβαίνει στην κατάσταση s_{t+1}, ως αποτέλεσμα της δυναμικής του περιβάλλοντος και της ενέργειας που επέλεξε, για να αποφασίσει εκ νέου για την a_{s+1}. Η αλληλουχία: *Κατάσταση (State)* $\Rightarrow$ *Ενέργεια (Action)* $\Rightarrow$ *Ανταμοιβή (Reward)* $\Rightarrow$ *Κατάσταση (State)* $\Rightarrow$ *Ενέργεια (Action)*, έδωσε και το όνομά της στον αλγόριθμο *SARSA*, έναν αλγόριθμο μάθησης χρονικών διαφορών (temporal difference learning algorithm), ο οποίος χρησιμοποιείται και στην παρούσα διατριβή.

Ένα από τα κύρια συστατικά ενός κλασικού συστήματος ενισχυτικής μάθησης είναι η *συνάρτηση αξίας* Q (*Q-value function*), η οποία συσχετίζει ζεύγη καταστάσεων - ενεργειών $Q(s_t, a_t)$, δίνοντάς τους μια τιμή που καθορίζει την μακροπρόθεσμη αξία τους για τον πράκτορα. Στόχος των αλγορίθμων ενισχυτικής μάθησης είναι η εκτίμηση μιας τέτοιας συνάρτησης για την εύρεση της βέλτιστης ενέργειας στην εκάστοτε κατάσταση. Για περιορισμένης έκτασης προβλήματα, η συνάρτηση αξίας Q μπορεί να πάρει τη μορφή ενός απλού πίνακα. Ωστόσο, πρόθεση της διδακτορικής διατριβής είναι η εφαρμογή της παραγόμενης μεθοδολογίας σε πραγματικά και σύνθετα προβλήματα, με εξαιρετικά μεγάλο αριθμό καταστάσεων, όπου θα ήταν δύσκολη η ακριβής εκτίμηση και η ανανέωση των τιμών του πίνακα. Επομένως, κρίνεται αναγκαία η παρουσία μιας συνάρτησης σε παραμετρική μορφή (γραμμική ή μη), η οποία θα προσπαθεί να προσεγγίσει τη συνάρτηση αξίας (function approximator) γενικεύοντας τα παραδείγματα που προέκυψαν στο παρελθόν, ώστε να βοηθήσει τον πράκτορα να λάβει τις σωστές αποφάσεις ακόμα και για καταστάσεις που δεν έχει επισκεφτεί προηγουμένως. Την περίπτωση της συνάρτησης προσέγγισης, οι απαιτήσεις του προβλήματος μετατοπίζονται στην σωστή επιλογή της μορφής των παραμέτρων και στην εύρεση των βέλτιστων τιμών τους, ώστε να επιτυγχάνεται η καλύτερη δυνατή σύγκλιση στη βέλτιστη συνάρτηση αξίας για το συγκεκριμένο πρόβλημα.

2.1.1 Εξερεύνηση και εκμετάλλευση

Στόχος της συνάρτησης αξίας Q είναι η εκτίμηση του προσδοκώμενου οφέλους μιας ενέργειας σε μια κατάσταση. Για να επιτευχθεί αυτό, οι αλγόριθμοι ενισχυτικής μάθησης προσπαθούν να προσεγγίσουν τη συνάρτηση αξίας Q χρησιμοποιώντας τις άμεσες ανταμοιβές μέσω της μεθόδου των χρονικών διαφορών. Για τον ικανοποιητικό υπολογισμό της συνάρτησης Q χρειάζονται αρκετές επαναλήψεις. Κάθε χρονική στιγμή t ο πράκτορας επιλέγει την ενέργεια a_t που μεγιστοποιεί την τιμή της συνάρτησης Q, $a_t = \arg\max_a Q(s_t, a)$. Όταν συμβαίνει αυτό, ο πράκτορας εκμεταλλεύεται (exploit) τη γνώση που έχει αποκτήσει από την αλληλεπίδραση με το περιβάλλον, επιλέγοντας με άπληστο (greedy) τρόπο την ενέργεια που μεγιστοποιεί την προσδοκώμενη ανταμοιβή του (expected reward), όπως αυτή έχει εκτιμηθεί από τη συνάρτηση Q. Στα πρώτα στάδια της εκπαίδευσης, η συνάρτηση Q προτιμά τις ενέργειες οι οποίες μεγιστοποιούν το άμεσο κέρδος από το περιβάλλον, με αποτέλεσμα να μην προτιμώνται ενέργειες οι οποίες μελλοντικά θα οδηγούσαν σε καλύτερα αποτελέσματα για τον πράκτορα. Κατά συνέπεια ο πράκτορας πρέπει να εξερευνά (explore) και εναλλακτικές δράσεις, μη βέλτιστες. Το δίλημμα αυτό είναι γνωστό στη βιβλιογραφία και ως *exploration vs. exploitation trade off*. Τεχνικές που αντιμετωπίζουν το παραπάνω δίλημμα είναι η ϵ-greedy στρατηγική, όπου με πιθανότητα ϵ εξερευνάται μια τυχαία δράση, διαφορετικά επιλέγεται η άπληστη ενέργεια, και η *softmax* στρατηγική, η οποία επιλέγει τυχαία μια ενέργεια με πιθανότητα ανάλογη της αξίας Q:

$$P[a = a_t] = \frac{e^{Q(s_t,a_t)/\tau}}{\sum_{i=1}^{n} e^{Q(s_t,a_i)/\tau}} \tag{2.1}$$

όπου $n = |A(s_t)|$ και $\tau > 0$ μια παράμετρος που ονομάζεται θερμοκρασία (temperature). Όταν $\tau \gg 0$ η επιλογή των δράσεων είναι ισοπίθανη, ενώ όταν $\tau \to 0$ η επιλογή εκφυλίζεται σε άπληστη.

2.1.2 Αλγόριθμος SARSA

Ο αλγόριθμος SARSA, στην απλή του μορφή, όταν η συνάρτηση αξιών Q αναπαρίσταται με πίνακα, λαμβάνει την μορφή του Αλγορίθμου 2.1. Στην υποενότητα 2.3.1 παρουσιάζεται η μορφή του αλγορίθμου SARSA όταν χρησιμοποιούμε τα ΔΗΚ ως συνάρτηση προσέγγισης της συνάρτησης Q.

2.2 Εξελικτική υπολογιστική

Η εξελικτική υπολογιστική [ES03], παρακλάδι της *υπολογιστικής νοημοσύνης* (computational intelligence), είναι μια περιοχή που περιλαμβάνει υπολογιστικές διαδικασίες στοχαστικής εξεύρεσης λύσεων ακολουθώντας το παράδειγμα της εξέλιξης των ειδών στο φυσικό περιβάλλον. Κύρια συστατικά της είναι η εξέλιξη ενός πληθυσμού λύσεων, τα λεγόμενα χρωμοσώματα (chromosomes), με τεχνικές όπως

Αλγόριθμος 2.1 Ο αλγόριθμος SARSA με αναπαράσταση πίνακα. Η μεταβλητή γ είναι ο ρυθμός έκπτωσης (discount rate) με $0 < \gamma \leq 1$ και καθορίζει την τρέχουσα αξία των μελλοντικών αμοιβών.

SARSA

initialize $Q(s, a)$
for all episodes **do**
  $s \leftarrow$ initial episode state
  $a \leftarrow \arg\max_a Q(s, a)$ using ϵ-greedy
  for all steps **do**
    observe reward r
    observe next state s'
    $a' \leftarrow \arg\max_a Q(s', a)$ using ϵ-greedy
    $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$
    $s \leftarrow s'$
    $a \leftarrow a'$
  end for
end for

αυτές που παρατηρούνται στη φύση, για παράδειγμα φυσική επιλογή με βάση την ικανότητα ενός οργανισμού, διασταύρωση χρωμοσωμάτων και μετάλλαξη γονιδίων μεταξύ άλλων. Κυριότερες μεθοδολογίες που έχουν αναπτυχθεί στην περιοχή της εξελικτικής υπολογιστικής είναι οι γενετικοί αλγόριθμοι, ο γενετικός προγραμματισμός και οι εξελικτικές στρατηγικές. Οι μεθοδολογίες αυτές αναπτύχθηκαν με σκοπό να χειρίζονται ικανοποιητικά περιοχές αναζήτησης με πολλαπλά τοπικά ελάχιστα, καθώς είναι δυσκολότερο να παγιδευτούν σε σχέση με τους αλγορίθμους κατάβασης πλαγιάς (gradient descent), αλλά και για την ικανότητά τους να δίνουν λύσεις σε προβλήματα όπου δεν υπάρχει άμεση και ακριβής συνάρτηση σκοπού (objective function) [Yao99]. Οι μεθοδολογίες εξελικτικής υπολογιστικής, μπορούν εξαρχής να χειρίζονται προβλήματα βελτιστοποίησης μεγάλου εύρους, πολύπλοκα, μη διαφορίσιμα, μη συνεχή, πολυτροπικά (multimodal) και θορυβώδη [Yao99, ES03, IS08].

Τα βασικότερα στοιχεία ενός αλγορίθμου εξελικτικής υπολογιστικής είναι τα ακόλουθα:

- **Πληθυσμός (Population)**: Οι αλγόριθμοι εξελικτικής υπολογιστικής χρησιμοποιούν έναν πληθυσμό στον οποίο κάθε στοιχείο του είναι και μία πιθανή λύση στο υπό εξέταση πρόβλημα. Τα στοιχεία του πληθυσμού ονομάζονται συνήθως χρωμοσώματα, αλλά μπορεί να αναφέρονται και ως οργανισμοί (organisms) ή γονιδιώματα (genomes).

- **Γενιές (Generations)**: Ο εξελικτικός αλγόριθμος εξελίσσει τον πληθυσμό για έναν προκαθορισμένο αριθμό γενεών.

- **Επιλογή (selection)**: Κατά την επιλογή προκρίνονται οι ικανότεροι οργανι-

σμοί προκειμένου να διασταυρωθούν και να προκύψουν απόγονοι που θα συνδυάζουν τα κατάλληλα στοιχεία από τους δύο γονείς και κατ' επέκταση να αποτελέσουν βελτιωμένη λύση για το πρόβλημα. Υπάρχουν διάφορες στρατηγικές επιλογής:

  - **Επιλογή Ρουλέτας (Roullette wheel)**: Κάθε χρωμόσωμα επιλέγεται ως γονέας με πιθανότητα ανάλογη της ικανότητας του κάθε χρωμοσώματος, όπως αυτή δίνεται από τη συνάρτηση αξιολόγησης του εκάστοτε προβλήματος σύμφωνα με τον τύπο: $P(k) = \frac{fitness(k)}{\sum_{i=1}^{n} fitness(i)}$.
  - **Επιλογή Κατάταξης (Ranking)**: Στην περίπτωση αυτή τα χρωμοσώματα ταξινομούνται με κλειδί την τιμή της συνάρτησης αξιολόγησης και στη συνέχεια επιλέγεται ένα ποσοστό αυτών.
  - **Επιλογή διαγωνισμού (Tournament)**: Στην περίπτωση που θέλουμε να επιλέξουμε N χρωμοσώματα ως γονείς για την αναπαραγωγή της επόμενης γενιάς, τότε δημιουργούμε N τυχαία τουρνουά. Στα τουρνουά λαμβάνουν μέρος όλα τα χρωμοσώματα του πληθυσμού και πραγματοποιούνται αγώνες μεταξύ δύο χρωμοσωμάτων όπου κάθε ένα μπορεί να κερδίσει με πιθανότητα p ανάλογη της τιμή της συνάρτησης αξιολόγησης. Οι αγώνες συνεχίζονται έως ότου παραμείνει ένα χρωμόσωμα το οποίο επιλέγεται ως γονέας.
  - **Ελιτισμός (Elitism)**: Ένα ποσοστό των ικανότερων χρωμοσωμάτων της τρέχουσας γενιάς μεταπηδά αυτούσιο στην επόμενη γενιά.

- **Αναπαραγωγή (Reproduction)**: Αφού έχει επιλεγεί το σύνολο των γονέων, συνεχίζουμε στο στάδιο της αναπαραγωγής για τη δημιουργία των απογόνων. Η αναπαραγωγή πραγματοποιείται κυρίως μέσω δύο γενετικών τελεστών, της διασταύρωσης και της μετάλλαξης:

  - **Διασταύρωση (Crossover)**: Στην περίπτωση της διασταύρωσης επιλέγονται τυχαία δύο γονείς και στη συνέχεια με πολλούς διαφορετικούς τρόπους δημιουργείται ο απόγονος επιλέγοντας γενετικό υλικό και από τους δύο γονείς. Στόχος είναι το γενετικό υλικό που συμβάλλει στη λύση και από τους δύο γονείς να ενωθεί ώστε να υπάρξει καλύτερη λύση στο πρόβλημα.
  - **Μετάλλαξη (Mutation)**: Η διαδικασία της μετάλλαξης αλλάζει με τυχαίο τρόπο κάποιο σημείο του γενετικού υλικού του απογόνου. Πραγματοποιείται συνήθως μετά τη διασταύρωση. Δημιουργεί μικρές παραλλαγές στη λύση που δίνεται μετά τη διασταύρωση και κύριος σκοπός της είναι να βελτιώσει την έρευνα στο χώρο αναζήτησης λύσεων, αποκολλώντας τον αλγόριθμο από τοπικά, μη-βέλτιστα, σημεία.

Η βασική δομή ενός εξελικτικού αλγορίθμου εμφανίζεται στον Αλγόριθμο 2.2.

Αλγόριθμος 2.2 Βασικός εξελικτικός αλγόριθμος.

EVOLVE

```
1: Initialize population
2: for all Generations do
3:    for all Individuals do
4:       Evaluate individual
5:    end for
6:    Selection
7:    Reproduction
8:    Update population
9: end for
```

2.2.1 Νευροεξέλιξη

Η νευροεξέλιξη είναι περιοχή της εξελικτικής υπολογιστικής που εστιάζει στη βελτιστοποίηση των νευρωνικών δικτύων. Πιο συγκεκριμένα, ένας αλγόριθμος νευροεξέλιξης μπορεί να βελτιστοποιεί τα βάρη του δικτύου, την τοπολογία του και τις μετα-παραμέτρους μεταξύ άλλων. Ένας αλγόριθμος νευροεξέλιξης επιλέγεται αντί ενός αλγορίθμου μάθησης διότι: α) μπορεί να συνεξελίξει πολλά χαρακτηριστικά ενός νευρωνικού δικτύου, β) έχει πιο ευέλικτες συναρτήσεις αξιολόγησης από ότι οι συναρτήσεις σφάλματος ή ενέργειας και γ) μπορεί να συνδυαστεί και με αλγορίθμους μάθησης [FDM08]. Τυπικό παράδειγμα όπου η νευροεξέλιξη προτιμάται έναντι κλασικών μαθησιακών μεθόδων είναι τα προβλήματα ενισχυτικής μάθησης, όπου στόχος είναι η πρόβλεψη του προσδοκώμενου κέρδους μιας ενέργειας δεδομένης μιας κατάστασης, όταν είναι δυνατή η παρατήρηση μόνο της άμεσης ανταμοιβής. Σε αυτό το παράδειγμα, τοπικοί αλγόριθμοι κατάβασης πλαγιάς θα συναντούσαν πολλές δυσκολίες έναντι του ευρύτερου φάσματος μιας συνάρτησης αξιολόγησης, καθώς ένας μεγάλος αριθμός παραμέτρων θα πρέπει να προσαρμόζεται ταυτόχρονα [Yao99, Mii11]. Επισκοπήσεις της περιοχής μπορούν να βρεθούν στα [Yao99, FDM08, Sta04].

2.2.2 Μάθηση και εξέλιξη

Εκτός από τα πλεονεκτήματα της νευροεξέλιξης απέναντι σε κλασικά παραδείγματα μάθησης, η σύμπλευση νευροεξέλιξης και μάθησης (είτε επιβλεπόμενης, είτε ενισχυτικής) μπορεί να οδηγήσει σε μια πληρέστερη προσαρμογή, καθώς η έρευνα πραγματοποιείται τόσο σε συνολικό (global, population-based stochastic search) όσο και σε τοπικό επίπεδο (learning through gradient descent, hill-climbing or other local search algorithm). Αυτός ο συνασπισμός αναφέρεται ως μιμητική υπολογιστική (memetic computing) [ES03] ή υβριδική εκπαίδευση (hybrid training) [Yao99].

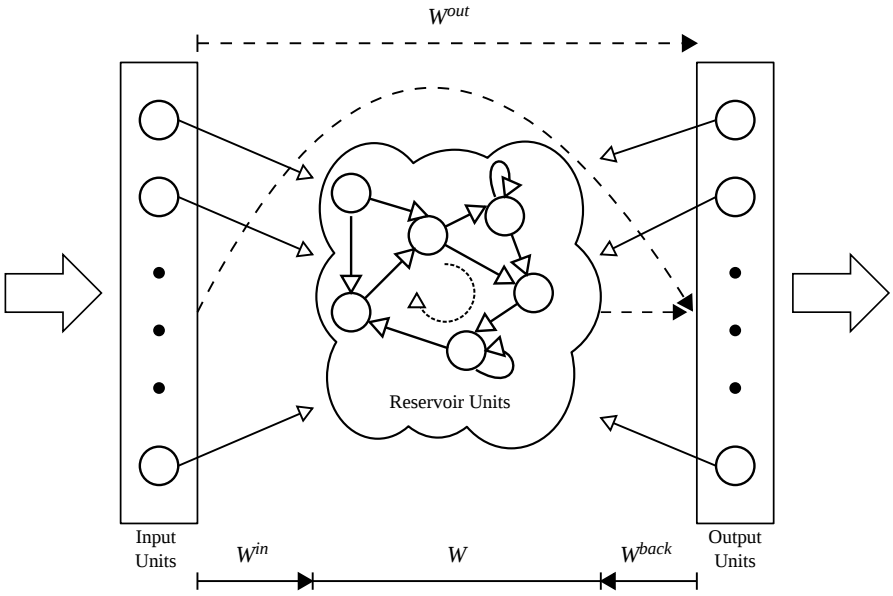
Με τη χρήση του παραπάνω συνδυασμού, γίνεται προσπάθεια να εξελίσσονται δίκτυα τα οποία θα μπορούν να μαθαίνουν καλύτερα, κάνοντας το πεδίο της ανα-

ζήτησης πιο ομαλό και ανακαλύπτοντας δίκτυα που δε θα ήταν εύκολο να αναγνωριστούν διαφορετικά [HN87]. Επίσης, παράλληλα με την αλληλεπίδραση [AL91], θα ήταν επιθυμητή η προσθήκη του φαινομένου Baldwin (*Baldwin effect*) [Bal96] στην εξελικτική διαδικασία. Σύμφωνα με το φαινόμενο αυτό, επιθυμητά χαρακτηριστικά που αναπτύσσονται μέσω της μάθησης, μεταφέρονται και στην εξελικτική διαδικασία. Γενικότερα λοιπόν, οι υβριδικοί αλγόριθμοι τείνουν να δίνουν καλύτερα αποτελέσματα σε σχέση με άλλους για έναν μεγάλο αριθμό προβλημάτων [Yao99], ενώ σύμφωνα με το θεώρημα του *μη δωρεάν γεύματος* (no-free-lunch theorem) [WM97], οι καλύτεροι αλγόριθμοι εκπαίδευσης είναι πάντα εξαρτημένοι από το πρόβλημα. Έτσι, στόχος δεν είναι η δημιουργία ενός αλγορίθμου αιχμής για έναν αριθμό συγκεκριμένων προβλημάτων, αλλά ένας εύρωστος αλγόριθμος που είναι ικανός να αντιμετωπίσει μια μεγάλη γκάμα προβλημάτων με αποδοτικό τρόπο. Στην παρούσα διατριβή γίνεται έλεγχος των πλεονεκτημάτων μιας τέτοιας συνέργειας και διερευνάται η διαφορά, όσον αφορά τη μεθοδολογία NEAR, ανάμεσα στις απόψεις που ποικίλουν, σχετικά με τους υβριδικούς αλγορίθμους, παράλληλα με τη διαμάχη για το ποιο είδος εξέλιξης είναι καταλληλότερο, η Λαμαρκιανή (Lamarckian) ή η Δαρβινική (Darwinian) εξέλιξη.

2.3 Ταμιευτήρια υπολογισμών

Η ιδέα που ακολουθούν τα ταμιευτήρια υπολογισμών (reservoir computing) και τα δίκτυα ηχωικών καταστάσεων (echo state networks) ειδικότερα, είναι η εξής: τυχαίως δημιουργημένα ταμιευτήρια νευρώνων, δηλαδή νευρωνικά δίκτυα με αναδράσεις (recurrent neural networks), τηρώντας ορισμένες αλγεβρικές συνθήκες, μπορούν να οδηγηθούν από ένα σήμα εισόδου, για να προκαλέσουν ένα σετ από πλούσιες δυναμικές στο ταμιευτήριο, με τη μορφή μη-γραμμικών σημάτων ανταπόκρισης (μη γραμμικός μετασχηματισμός της πληροφορίας σε έναν νέο πολυδιάστατο χώρο). Τα σήματα αυτά, μαζί με τα σήματα εισόδου μπορούν να συνδυαστούν για την ανάπτυξη ενός γραμμικού συνδυασμού χαρακτηριστικών, με την ονομασία **συνάρτηση ανάγνωσης** (read-out function), $y = \mathbf{w}^T \cdot \phi(\mathbf{x})$, η οποία θα μπορούσε να αποτελέσει την πρόβλεψη του επιθυμητού σήματος εξόδου, δεδομένου ότι τα βάρη w έχουν εκπαιδευθεί ανάλογα. Οι αναδράσεις, που εμφανίζονται στο ταμιευτήριο ως κύκλοι στην τοπολογία, δίνουν τη δυνατότητα στο ΔΗΚ να διατηρεί μια δυναμική μνήμη και να επεξεργάζεται χρονικά την πληροφορία. Ως συναρτήσεις ανάγνωσης μπορούν να χρησιμοποιηθούν και άλλες αναπαραστάσεις, όπως για παράδειγμα τα εμπροσθοτροφοδοτούμενα νευρωνικά δίκτυα (feedforward neural networks), ωστόσο η γραμμική συνάρτηση είναι αρκετή για να πετύχει εξαιρετικά αποτελέσματα σε πρακτικές εφαρμογές.

Η βασική μορφή ενός ΔΗΚ εμφανίζεται στο Σχήμα 2.1. Το ταμιευτήριο αποτελείται από ένα στρώμα K μονάδων εισόδου, συνδεδεμένο με τις N μονάδες του ταμιευτηρίου μέσω ενός $N \times K$ πίνακα συνδέσεων με βάρη, W^{in}. Ο πίνακας γειτνίασης της δεξαμενής, W, είναι ένας πίνακας $N \times N$. Προαιρετικά, ένας πίνακας όπισθεν προβολής (back-projection) W^{back} με διαστάσεις $N \times L$, όπου L ο αριθ-



Σχήμα 2.1: Τυπικό Δίκτυο Ηχωικών Καταστάσεων. Με διακεκομμένες γραμμές σημειώνονται οι συνάψεις που μπορούν να εκπαιδευτούν, ενώ ως βέλη με συνεχόμενη γραμμή παρουσιάζονται οι συνάψεις που είναι σταθερές κατά τη διάρκεια της εκμάθησης.

μός των μονάδων εξόδου, επαναφέρει τα σήματα εξόδου πίσω στη δεξαμενή [1]. Τα βάρη των συνδέσεων των μονάδων εξόδου (γραμμικά χαρακτηριστικά) και των μονάδων του ταμιευτηρίου (μη γραμμικά χαρακτηριστικά) με την έξοδο, συναθροίζονται στον $L \times (K+N)$ πίνακα W^{out}. Στη συγκεκριμένη διατριβή, οι μονάδες της δεξαμενής συμπεριλαμβάνουν την $f(x) = tanh(x)$ ως συνάρτηση ενεργοποίησης (μεταφοράς), ενώ οι μονάδες εξόδου χρησιμοποιούν τις σιγμοειδείς συναρτήσεις $g_1(x) = tanh(x)$ και $g_2(x) = \frac{1}{1+e^{-x}}$ καθώς και την ταυτότητα, $g_3(x) = x$, ανάλογα με το τρέχον πρόβλημα. Πιο συγκεκριμένα, η ταυτότητα είναι χρήσιμη όταν η έκταση της εξόδου δεν είναι γνωστή, ενώ οι σιγμοειδείς συναρτήσεις είναι πιο κατάλληλες για φραγμένη έξοδο ή για την εκμάθηση πιθανοτήτων, όπως για παράδειγμα για μια μικτή στρατηγική μέσω ενισχυτικής μάθησης.

Καλές πρακτικές για τη δημιουργία ΔΗΚ, δηλαδή διαδικασίες για την παραγωγή των τυχαίων πινάκων γειτνίασης W^{in}, W και W^{back}, μπορούν να βρεθούν στα [Jae02, LJ09]. Περιληπτικά αυτές είναι:

1. Ο πίνακας W πρέπει να είναι αραιός.
2. Η μέση τιμή των βαρών πρέπει να είναι γύρω στο μηδέν.
3. Το N πρέπει να είναι αρκετά μεγάλο για να παράγονται πολλά χαρακτηριστικά, δίνοντας καλή συμπεριφορά στις προβλέψεις.

[1] Οι όροι δεξαμενή και ταμιευτήριο χρησιμοποιούνται ισοδύναμα

4. Η φασματική ακτίνα ρ του W πρέπει να είναι μικρότερη του 1 ώστε πρακτικά (και όχι θεωρητικά) να βεβαιωθεί ότι το δίκτυο θα λειτουργεί ως ΔΗΚ.

5. Ένας αδύναμος, ομοιόμορφος, λευκός θόρυβος μπορεί να προστεθεί στα χαρακτηριστικά για λόγους ευστάθειας.

Στη συγκεκριμένη εργασία θεωρούμε διακριτά χρονικά μοντέλα και ΔΗΚ χωρίς όπισθεν προβολή. Στο πρώτο βήμα, κλιμακώνεται (scale) και μετακινείται (shift) το σήμα εισόδου, $\mathbf{u} \in \mathbb{R}^K$, ανάλογα με το εάν θέλουμε το δίκτυο να λειτουργεί στο γραμμικό ή μη-γραμμικό κομμάτι των σιγμοειδών συναρτήσεων. Το χαρακτηριστικό διάνυσμα του ταμιευτηρίου, $\mathbf{x} \in \mathbb{R}^N$, δίνεται από την Εξίσωση 2.2:

$$\mathbf{x}(t+1) = \mathbf{f}(W^{in}\mathbf{u}(t+1) + W\mathbf{x}(t) + \mathbf{v}(t+1)) \tag{2.2}$$

όπου **f** είναι η ανά στοιχείο εφαρμογή της συνάρτησης ενεργοποίησης του ταμιευτηρίου και **v** είναι το ομοιόμορφο διάνυσμα λευκού θορύβου. Η έξοδος, $\mathbf{y} \in \mathbb{R}^L$, δίνεται από την Εξίσωση 2.3:

$$\mathbf{y}(t+1) = \mathbf{g}(W^{out}[\mathbf{u}(t+1)|\mathbf{x}(t+1)]) \tag{2.3}$$

όπου **g** είναι η ανά στοιχείο εφαρμογή της συνάρτησης ενεργοποίησης εξόδου, ενώ ο τελεστής $|$ υποδηλώνει τη σύνδεση των διανυσμάτων. Για προβλήματα επιβλεπόμενης μάθησης, το πρόβλημα μπορεί να συνταχθεί ως πρόβλημα γραμμικής παλινδρόμησης και τα βάρη της εξόδου μπορούν να καθοριστούν από ένα πλήθος μεθόδων αριθμητικής ανάλυσης [LJ09].

Όπως συζητήθηκε στο Σχήμα 1.1, ο συγκερασμός εξελικτικής υπολογιστικής και μάθησης γίνεται για να καθοριστούν μία καλή αρχική τοπολογία και τα βάρη της μέσα από την καθολική αναζήτηση, ώστε στη συνέχεια να βελτιωθούν μέσω της τοπικής έρευνας και να βρεθούν ευκολότερα τα τοπικά ελάχιστα. Εφόσον η καθολική αναζήτηση αναγνωρίσει μια λεκάνη έλξης όπου βρίσκεται το καθολικό βέλτιστο, η τοπική αναζήτηση θα μπορέσει να βρει αυτό το καθολικό βέλτιστο [Yao99]. Η παρουσία πολλαπλών τοπικών ελαχίστων έχει αποδειχθεί ότι ισχύει στα ΔΗΚ [JLPS07], οπότε και η ανάπτυξη μιας πληθυσμιακής εξελικτικής αναζήτησης ταιριάζει απόλυτα. Επιπλέον, ο έμφυτος διαχωρισμός στα ΔΗΚ μεταξύ της δεξαμενής και της συνάρτησης ανάγνωσης συμβαδίζει με την υιοθέτηση τόσο αρχών εξέλιξης όσο και μάθησης. Μπορούν να εφαρμοστούν εξελικτικές τεχνικές στη δεξαμενή με στόχο να αναγνωριστούν δεξαμενές που βρίσκονται σε λεκάνες έλξης, μία από τις οποίες περιέχει το καθολικό βέλτιστο. Στη συνέχεια με την προσαρμογή της συνάρτησης ανάγνωσης μέσω τεχνικών μάθησης το καθολικό βέλτιστο μπορεί να αναγνωριστεί ευκολότερα.

2.3.1 Δίκτυα ηχωικών καταστάσεων και ενισχυτική μάθηση

Για προβλήματα ενισχυτικής μάθησης με K συνεχείς μεταβλητές καταστάσεων και L διακριτές ενέργειες, μπορούμε να χρησιμοποιούμε ένα ΔΗΚ για να μοντελοποιήσουμε τη συνάρτηση αξίας Q. Κάθε έξοδος του δικτύου l, αντιστοιχίζεται με

μία ενέργεια $a_l \in A = \{a_l, l = 1 \dots L\}$, όπου η έξοδος του δικτύου y_l αντικατοπτρίζει την μακροπρόθεσμη εκπτωτική αξία $Q(\mathbf{s}, a_l)$ της εκτέλεσης της ενέργειας a_l, όταν ο πράκτορας βρίσκεται στην κατάσταση $\mathbf{s}$. Για παράδειγμα, ακολουθώντας την Εξίσωση 2.3 και δεδομένου ότι $g(x) = g_3(x) = x$, οι αξίες Q υπολογίζονται από το ΔΗΚ με βάση την Εξίσωση 2.4:

$$y_l = Q(\mathbf{s}, a_l) = \sum_{i=1}^{K} w_{li}^{out} s_i + \sum_{i=K+1}^{K+N} w_{li}^{out} x_{i-K}, l = 1, \dots, L. \tag{2.4}$$

Οι ενέργειες επιλέγονται κάτω από την ϵ-greedy πολιτική [SB98], δηλαδή τυχαία με πιθανότητα ϵ και σύμφωνα με την Εξίσωση 2.5 με πιθανότητα $1 - \epsilon$:

$$l = \underset{l \in \{1 \dots L\}}{\arg\max}\, Q(\mathbf{s}, a_l) \tag{2.5}$$

Τα βάρη προσαρμόζονται χρησιμοποιώντας τη μέθοδο χρονικών διαφορών και τον αλγόριθμο SARSA σε συνδυασμό με τον αλγόριθμο κατάβασης πλαγιάς (gradient descent) [SB98, SGL06]. Στην απλούστερη περίπτωση και δίχως ίχνη επιλεξιμότητας (eligibility traces), δηλαδή SARSA(0), οι εξισώσεις ανανέωσης (Εξίσωση 2.6 και Εξίσωση 2.7) παίρνουν τη μορφή:

$$\delta = r + \gamma Q(\mathbf{s}, a') - Q(\mathbf{s}, a_l) \tag{2.6}$$

$$\mathbf{w}_l^{out} = \mathbf{w}_l^{out} + \alpha \delta \nabla_{\mathbf{w}_l^{out}} Q(\mathbf{s}, a_l) \tag{2.7}$$

όπου a' η επόμενη προς επιλογή ενέργεια, α ο ρυθμός μάθησης (learning rate) και το διάνυσμα $\mathbf{w}_l^{out}$ η l^{th} σειρά του πίνακα $\mathbf{W}^{out}$. Ο υπολογισμός της κλίσης (gradient) στην Εξίσωση 2.7 εξαρτάται από την συνάρτηση εξόδου $g(x)$. Για παράδειγμα, για τις κοινές συναρτήσεις εξόδου που χρησιμοποιούνται στην ενισχυτική μάθηση, σιγμοειδή $g_2(x)$ και ταυτότητα $g_3(x)$, η κλίση υπολογίζεται σύμφωνα με τις Εξισώσεις 2.8 και 2.9 αντίστοιχα:

$$\nabla_{\mathbf{w}_l^{out}} Q(\mathbf{s}, a_l) = Q(\mathbf{s}, a_l)(1 - Q(\mathbf{s}, a_l))F \tag{2.8}$$

$$\nabla_{\mathbf{w}_l^{out}} Q(\mathbf{s}, a_l) = F \tag{2.9}$$

με $F = [\mathbf{s}|\mathbf{x}]$. Ο πλήρης αλγόριθμος χρονικών διαφορών SARSA(λ) με ίχνη επιλεξιμότητας για ΔΗΚ, προσαρμοσμένος στις ανάγκες της διατριβής από το [SB98], παρουσιάζεται στον Αλγόριθμο 2.3.

Εκτός από τον κανόνα μάθησης κατάβασης πλαγιάς, υπάρχουν και άλλες μέθοδοι με τις οποίες πραγματοποιείται προσαρμογή των βαρών σε σημεία καλύτερης πρόβλεψης των τιμών Q. Μια τέτοια μέθοδος είναι η μέθοδος χρονικών διαφορών ελαχίστων τετραγώνων (least squares temporal difference learning - LSTD) [BB96, Boy02]. Στη μέθοδο LSTD, απαιτείται η ενημέρωση δύο πινάκων, **A** και **b**, χρησιμοποιώντας τις Εξισώσεις 2.10 και 2.11:

$$\mathbf{A}_l = \mathbf{A}_l + \mathbf{e}_l^{out}(F - F')^T \tag{2.10}$$

$$\mathbf{b}_l = \mathbf{b}_l + \mathbf{e}_l^{out} r \tag{2.11}$$

Αλγόριθμος 2.3 SARSA(λ) με γραμμική κατάβαση πλαγιάς για ΔΗΚ και ϵ-greedy πολιτική επιλογής ενεργειών.

LINEAR-GRADIENT-DESCENT-SARSA(λ)

for all episodes **do**
  $\mathbf{e}^{out} \leftarrow \mathbf{0}$
  $s, a_l \leftarrow$ initial state and action of the episode {Equation 2.5}
  $F \leftarrow [\mathbf{u}(s)|\mathbf{x}]$
  for all steps of episode **do**
    $\mathbf{e}^{out} \leftarrow \gamma\lambda\mathbf{e}^{out}$
    $g_l \leftarrow \nabla_{\mathbf{w}_l^{out}} Q(s,a)$ {Equation 2.8 or 2.9}
    $\mathbf{e}_l^{out} \leftarrow \mathbf{e}_l^{out} + g_l$ {Accumulating traces}
    $\mathbf{e}_l^{out} \leftarrow g_l$ {Replacing traces}
    Take action a_l, observe reward, r, and next state, s'
    $\delta \leftarrow r - Q(s, a_l)$
    if s' is terminal **then**
      $\mathbf{W}^{out} \leftarrow \mathbf{W}^{out} + \alpha\delta\mathbf{e}^{out}$
      Go to next episode;
    end if
    if ϵ-greedy holds **then**
      $a \leftarrow$ a random action $\in A$
    else
      $a \leftarrow \arg\max_{b \in A} Q(s', b)$
    end if
    $Q_a \leftarrow Q(s', a)$
    $\delta \leftarrow \delta + \gamma Q_a$
    $\mathbf{W}^{out} \leftarrow \mathbf{W}^{out} + \alpha\delta\mathbf{e}^{out}$
  end for
end for

μέσα στον Αλγόριθμο 2.3 και κάνοντας τις ενημερώσεις $\forall l \in \{1 \dots L\}$ σε κάθε επεισόδιο. Τα αναθεωρημένα βάρη υπολογίζονται κατ' απαίτηση μέσω της Εξίσωσης 2.12:

$$\mathbf{w}_l^{out} = \mathbf{A}_l^{-1}\mathbf{b}_l \tag{2.12}$$

Τα ΔΗΚ έχουν χρησιμοποιηθεί στο παρελθόν ως προσεγγιστικές συναρτήσεις σε εργασίες ενισχυτικής μάθησης [SGL06] στη βασική τους, μη βελτιστοποιημένη μορφή. Στην ίδια εργασία, οι συγγραφείς δείχνουν ότι τα ΔΗΚ χρησιμοποιώντας τον αλγόριθμο SARSA(λ) παρουσιάζουν την ίδια συγκλίνουσα συμπεριφορά με μια γραμμική προσεγγιστική συνάρτηση που χρησιμοποιεί SARSA(λ), όμως με την έξτρα δυνατότητα να ισχύει η συμπεριφορά αυτή και για διαδικασίες απόφασης Markov k-τάξης (k-th order Markov decision processes). Η ικανότητα αυτή κάνει τα ΔΗΚ θεωρητικά επιθυμητά ως συναρτήσεις προσέγγισης για προβλήματα ενισχυτικής μάθησης.

Ολοκληρώνοντας την παρουσίαση των ΔΗΚ σε συνδυασμό με την ενισχυτική μάθηση, πρέπει να αναφερθεί ότι και οι αλγόριθμοι αναζήτησης πολιτικών (policy search) μπορούν να εφαρμοστούν ώστε να βρεθούν τα κατάλληλα βάρη του πίνακα W^{out}, για παράδειγμα μέσω εξελικτικής υπολογιστικής. Οι μέθοδοι αναζήτησης πολιτικών συνήθως εφαρμόζονται ευκολότερα στην πράξη, ενώ ταυτόχρονα μπορούν να μεταβάλλουν και μετα-παραμέτρους ή να εφαρμοστούν σε μη διαφορίσιμες συναρτήσεις προσέγγισης. Ομοιόμορφες ή Γκαουσιανές μεταλλάξεις μπορούν να χρησιμοποιηθούν για την μεταβολή των παραμέτρων, αλλά και πιο περίπλοκοι αλγόριθμοι όπως ο Covariance Matrix Adaptation-Evolutionary Strategy (CMA-ES) [HO01]. Ο CMA-ES έχει επιτυχημένα εφαρμοστεί σε μεθόδους νευροεξέλιξης [Ige03, JBS08a, HMI09], αλλά και σε μη-προσαρμοζόμενες τοπολογίες δικτύων, όπως θα δούμε στη ενότητα 2.5. Μια σύγκριση μεταξύ μεθόδων χρονικών διαφορών και αναζήτησης πολιτικών [WTS09] κατέληξε στο γεγονός ότι και οι δύο τεχνικές έχουν μειονεκτήματα σε διαφορετικές συνθήκες (θόρυβος στα αισθητήρια, θόρυβος στη συνάρτηση αξιολόγησης). Στην περίπτωσή μας, στόχος παραμένει η δημιουργία ενός εύρωστου αλγορίθμου, που θα προσαρμόζει τη συνάρτηση προσέγγισης στο υπό εξέταση πρόβλημα και θα έχει τουλάχιστον συγκρίσιμες επιδόσεις με αλγορίθμους αιχμής, σε πολλά και διαφορετικού είδους περιβάλλοντα (μη-γραμμικά, μη-Μαρκοβιανά), με ταυτόχρονη εκμετάλλευση των πλεονεκτημάτων τόσο της μάθησης, όσο και της εξέλιξης.

Πιο λεπτομερής ανάλυση και παραδείγματα ΔΗΚ περιέχονται στα [Jae01, Jae02], ενώ για μια επισκόπηση των ταμιευτήρων υπολογισμών ο αναγνώστης μπορεί να αναφερθεί στο [LJ09].

2.4 NEAT

Παρά το γεγονός ότι έχουν αναπτυχθεί επιτυχημένες μεθοδολογίες νευροεξέλιξης, δεν παύουν να υπάρχουν και ορισμένα μειονεκτήματα. Τα πιο γνωστά είναι οι ανταγωνιστικές συμβάσεις (competing conventions) [Rad93] και η πρόωρη σύγκλιση (premature convergence) [FDM08]. Για την πρώτη περίπτωση, δεδομένου ότι η γνώση που εμπεριέχεται σε ένα νευρωνικό δίκτυο βρίσκεται διανεμημένη στα βάρη του, η διασταύρωση ενός μέρους ενός νευρωνικού δικτύου με ένα άλλο μέρος ενός άλλου νευρωνικού δικτύου πιθανόν να καταστρέψει και τα δύο νευρωνικά δίκτυα σε ορισμένες περιπτώσεις [Yao99]. Εξαιτίας λοιπόν του προβλήματος των ανταγωνιστικών συμβάσεων, ο τελεστής της διασταύρωσης δε χρησιμοποιείται συνήθως σε μεθόδους νευροεξέλιξης, αν και έχει διαπιστωθεί ότι είναι ένας χρήσιμος τελεστής για τις επιδόσεις της εξέλιξης σε ορισμένα προβλήματα [Yao99]. Στο πρόβλημα της πρόωρης σύγκλισης, δύσκολα τοπία αξιών της συνάρτησης αξιολόγησης με πολλά τοπικά βέλτιστα, μπορούν να προκαλέσουν ταχεία μείωση της ποικιλίας του πληθυσμού και έτσι να καταστήσουν την αναζήτηση αναποτελεσματική [FDM08]. Έχοντας υπόψη τα παραπάνω, η μέθοδος NeuroEvolution of Augmented Topologies (NEAT) χρησιμοποιήθηκε ως μετα-μέθοδος αναζήτησης της πλατφόρμας νευροεξέλιξης ΔΗΚ, NeuroEvolution of Augmented Reservoirs

(NEAR) και η οποία θα παρουσιαστεί στο επόμενο κεφάλαιο. Οι λόγοι είναι ότι η μεθοδολογία NEAT προτείνει λύσεις που βοηθούν στην εκτόνωση των προαναφερθέντων δυσκολιών και έτσι επιλέχθηκε ως μια πολύ καλή μέθοδος προκειμένου να προσαρμοστεί και να χρησιμοποιηθεί στην αναζήτηση ικανών ΔΗΚ.

Η NEAT [SM02, Sta04] είναι μία μέθοδος TWEANN βασισμένη σε τέσσερις πυλώνες και αποτελεί μέθοδο αναφοράς στην περιοχή της νευροεξέλιξης. Αρχικά, ο γονότυπος του νευρωνικού δικτύου είναι κωδικοποιημένος ως γραμμικό γονιδίωμα (linear genome), με αποτέλεσμα να είναι λιγότερο απαιτητικός σε μνήμη σε σχέση με τους αλγορίθμους που χρησιμοποιούν τους πλήρεις πίνακες γειτνίασης (adjacency matrices). Δεύτερον, χρησιμοποιώντας την έννοια της *ιστορικής σηματοδότησης (historical markings)*, νεόδμητες συνδέσεις σημειώνονται με έναν μοναδικό για το δίκτυο αριθμό καινοτομίας (innovation number). Η NEAT κατά τη διάρκεια της διασταύρωσης ευθυγραμμίζει τα χρωμοσώματα των γονέων ταιριάζοντας τους αριθμούς καινοτομίας των συνδέσεων και εκτελεί διασταυρώσεις μόνο στις συνδέσεις όπου υπάρχει ταίριασμα. Με τον τρόπο αυτό εκτελούνται μόνο διασταυρώσεις που έχουν νόημα. Η ιστορική σηματοδότηση επανέφερε με τον τρόπο αυτό την ωφέλιμη εφαρμογή του τελεστή της διασταύρωσης. Τρίτη αρχή της NEAT είναι η προστασία της καινοτομίας μέσω της *ομαδοποίησης* των οργανισμών του πληθυσμού σε είδη (speciation). Με τον τρόπο αυτό ο κάθε οργανισμός μάχεται για την επιβίωσή του μόνο μέσα στη ομάδα του και, έτσι, του δίνεται ο χρόνος να αναπτυχθεί όσο το δυνατόν περισσότερο. Η ομαδοποίηση είναι ένας τρόπος να αντιμετωπιστεί η πρόωρη σύγκλιση. Τέλος, η τέταρτη αρχή είναι η αρχικοποίηση του πληθυσμού με μινιμαλιστικά δίκτυα. Η NEAT ξεκινάει έχοντας στον αρχικό πληθυσμό του δίκτυα χωρίς κανέναν κρυμμένο νευρώνα, για να ξεκινήσει με το μικρότερο δυνατό διάστημα αναζήτησης και ταυτόχρονα κάθε πολυπλοκότητα που εισάγεται στο δίκτυο να αντικατοπτρίζεται και από αντίστοιχη βελτίωση στην συνάρτηση αξιολόγησης. Η NEAT περιπλέκει τα δίκτυα μέσω δομικών μεταλλάξεων, εισάγοντας κόμβους (νευρώνες) και συνδέσεις. Επιπρόσθετη προσαρμογή του δικτύου γίνεται από τη μετάλλαξη των βαρών των συνδέσεων, είτε προσθέτοντας στις συνδέσεις κάποιο θόρυβο (perturbation), είτε αλλάζοντας εντελώς τα βάρη (restart). Στην παρούσα διατριβή θεωρήθηκε ότι οι τέσσερις αυτές αρχές μπορούν να αποτελέσουν τα συστατικά ενός αλγορίθμου μετα-νευροεξέλιξης, οι οποίες θα προσαρμόζονται κάθε φορά στο επιλεγμένο νευρωνικό μοντέλο και στη συγκεκριμένη περίπτωση στα ΔΗΚ. Ο αλγόριθμος 2.4 παρουσιάζει τη βασική διαδικασία που χρησιμοποιείται από πολλούς αλγορίθμους νευροεξέλιξης και από τη NEAT.

Μετά την αρχικοποίηση, ξεκινά η εξελικτική διαδικασία. Κάθε ένας από τους οργανισμούς στον πληθυσμό περνά από τη διαδικασία αξιολόγησης. Η επίδοση του οργανισμού στο συγκεκριμένο πρόβλημα, καταγράφεται ως ικανότητα στη NEAT. Αφού ελεγχθεί όλος ο πληθυσμός, εκτελείται το βήμα της εξέλιξης (Αλγόριθμος 2.5). Αρχικά τα στάσιμα είδη, τα είδη των οποίων η καταλληλότητα δε βελτιώθηκε στο πρόσφατο παρελθόν, απομακρύνονται. Ο πληθυσμός ομαδοποιείται σε παλιά και νέα είδη με βάση ένα κριτήριο ομοιότητας από τον αντιπρόσωπο του είδους. Είδη που δεν έχουν προσελκύσει κάποιο μέλος κατά τη διάρκεια της ομαδοποίησης απομακρύνονται επίσης. Η καταλληλότητα υπολογίζεται τόσο για τα

Αλγόριθμος 2.4 Βασική διαδικασία νευροεξέλιξης.

NEUROEVOLUTION

```
1: π ← αρχικοποίηση-πληθυσμού
2: for all γενιές do
3:    for all γονιδιώματα do
4:       δίκτυο ← φαινότυπος(γονιδίωμα)
5:       καταλληλότητα ← αξιολόγηση(δίκτυο)
6:       γονιδίωμα ← καταλληλότητα
7:    end for
8:    εξέλιξη(π)
9: end for
```

γονιδιώματος όσο και για τα είδη και ανάλογα προσδίδεται ένας σχετικός αριθμός οργανισμών σε κάθε είδος για αναπαραγωγή. Κατά τη διάρκεια της αναπαραγωγής σε κάθε είδος, ο πρωταθλητής κρατείται χωρίς καμία τροποποίηση. Οι υπόλοιπες θέσεις καλύπτονται εφαρμόζοντας μετάλλαξη ή διασταύρωση ή και τα δύο, σε γονείς που περνούν ένα προκαθορισμένο κατώφλι επιβίωσης, S_{thresh}. Αναπαραγωγή μεταξύ οργανισμών που ανήκουν σε διαφορετικά είδη επιτρέπεται με πιθανότητα p_i (inter-species mating). Περισσότερες λεπτομέρειες ο αναγνώστης μπορεί να βρει στα [Sta04, SM02].

2.5 Βελτιστοποίηση ταμιευτηρίων νευρώνων

Η τυχαία δημιουργία ΔΗΚ δίνει ικανοποιητικά αποτελέσματα σε αρκετά προβλήματα, όμως υπάρχουν πολλές αναφορές στη βιβλιογραφία οι οποίες παρουσιάζουν καλύτερα αποτελέσματα σε περιπτώσεις που πραγματοποιείται κάποιου είδους βελτιστοποίηση στα βάρη και την τοπολογία του ταμιευτηρίου.

Μια απλή, πρώτη μέθοδος βελτιστοποίησης της τοπολογίας της δεξαμενής W είναι η χρήση της τοπικής αναζήτησης επόμενης ανάβασης (next ascent local search) [BT05]. Πειραματικά αποτελέσματα έδειξαν ότι το σφάλμα πρόβλεψης μπορεί να μειωθεί σημαντικά χρησιμοποιώντας αυτή τη μέθοδο, έναντι της τυχαίας αναζήτησης. Μια παρόμοια προσέγγιση προτάθηκε στο [BP05], όπου αντί της αναζήτησης επόμενης ανάβασης, χρησιμοποιήθηκε γενετικός αλγόριθμος. Κάθε μονάδα του πληθυσμού του ήταν ένα δυαδικό διάνυσμα, όπου το 1 σήμαινε την ύπαρξη μίας σύνδεσης στη δεξαμενή W, ενώ το 0 την απουσία της. Ο γενετικός αλγόριθμος χρησιμοποιούσε τον μηχανισμό της ρουλέτας για την επιλογή των μονάδων και την ομοιόμορφη διασταύρωση (uniform crossover). Στη συγκεκριμένη περίπτωση παρατηρήθηκε αυξημένη προβλεπτική ικανότητα σε σχέση με τα μη-βελτιστοποιημένα δίκτυα. Στο [IvdZBP04] οι συγγραφείς διαφοροποιούνται από την υλοποίηση μίας μεθόδου εξέλιξης τοπολογίας και βαρών (Topology and Weight Evolution of Artificial Neural Networks - TWEANN) και καταφεύγουν στην εξέλιξη των παραμέτρων N, ρ και της D, της συνδεσμικής πυκνότητας του W, προ-

Αλγόριθμος 2.5 Ο εξελικτικός μετα-αλγόριθμος της μεθόδου NEAT, χωρίς συγκεκριμένες λεπτομέρειες υλοποίησης.

NEAT

```
remove-stagnated-species
cluster-population
remove-empty-species
calculate-adjusted-gene-fitness
calculate-species-fitness
calculate-offsprings-per-species
remove-non-assigned-species
for all species do
  keep-champion-gene
  while offspring-remain do
    x ← select-parent(S_thres)
    if mutate then
      offspring ← mutate(x)
    else
      y ← select-parent(S_thres)
      if random < p_i then
        y ← interspecies-selection(S_thres)
      end if
      if mate then
        offspring ← xover(x,y)
      else
        offspring ← mutate(xover(x,y))
      end if
    end if
    add-offspring(offspring)
  end while
end for
```

κειμένου να βρεθεί ένα ικανοποιητικό σετ τιμών. Στη συνέχεια, βελτιώνουν το ισχυρότερο ΔΗΚ που παράχθηκε εφαρμόζοντας μια δεύτερη εξελικτική τεχνική απευθείας στο γονιδίωμα βαρών, όπως προκύπτει από τη διανυσματοποίηση των πινάκων βαρών ενός ΔΗΚ. Στο [XLP05] χρησιμοποιείται γενετικός αλγόριθμος για την εύρεση των βαρών της γραμμικής συνάρτησης εξόδου ενός ΔΗΚ, ο οποίος χρησιμοποιεί επιλογή με ρουλέτα, συνεξέλιξη και μετάλλαξη, ενώ η συνάρτηση καταλληλότητας τροποποιείται για να ομαλοποιεί τα σήματα εξόδου.

Σε δύο πιο πρόσφατες μεθόδους [JBS08a, JBS08b] χρησιμοποιήθηκε ο αλγόριθμος CMA-ES για τη βελτιστοποίηση διαφόρων σετ παραμέτρων των ΔΗΚ, όπως: (i) βελτιστοποίηση των βαρών της συνάρτησης εξόδου, (ii) βελτιστοποίηση των βαρών της συνάρτησης εξόδου και της φασματικής ακτίνας ρ της δεξαμενής και (iii) βελτιστοποίηση των σιγμοειδών κλίσεων των συναρτήσεων f σε προ-

βλήματα τόσο επιβλεπόμενης όσο και ενισχυτικής μάθησης. Σε παρόμοιο κλίμα, εξελικτικές στρατηγικές έχουν χρησιμοποιηθεί για την εξέλιξη των βαρών εξόδου (πίνακας W^{out}) ΔΗΚ και σε προβλήματα μη επιβλεπόμενης μάθησης [DBS08]. Οι εξελικτικές στρατηγικές χρησιμοποιούνται και σε άλλα προβλήματα νευροεξέλιξης, όπως για παράδειγμα ο αλγόριθμος CMA-ES έχει χρησιμοποιηθεί για την εξέλιξη των βαρών τόσο εμπροσθοτροφοδοτούμενων νευρωνικών δικτύων όσο και νευρωνικών δικτύων με ανάδραση για προβλήματα ενισχυτικής μάθησης παρουσιάζοντας ταχείς χρόνους σύγκλισης σε επιτυχημένες πολιτικές [Ige03, HMI09].

Άλλες επιτυχημένες μεθοδολογίες που αναπτύχθηκαν στην περιοχή της εξέλιξης των δικτύων με ανάδραση περιλαμβάνουν τον αλγόριθμο Evolino [SWGG07] και τον αλγόριθμο Cooperative Synapse Neuroevolution (CoSyNe) [GSM08]. Ο αλγόριθμος Evolino είναι ένας συνδυασμός του αλγορίθμου Enforced SubPopulations (ESP) [GM97] για την προσαρμογή των βαρών νευρωνικών δικτύων και των δικτύων Long Short Term Memory (LSTM) [HS97], που αποτελούνται από μονούς νευρώνες ικανούς να διατηρούν μνήμη για μεγάλα χρονικά διαστήματα. Στον Evolino, αφού προσδιοριστεί ο αριθμός των μονάδων LSTM, τα εσωτερικά τους βάρη εξελίσσονται σύμφωνα με τον αλγόριθμο ESP. Τα βάρη της συνάρτησης εξόδου από τις μονάδες LSTM (συνάρτηση ανάγνωσης) βρίσκονται με τη βοήθεια γραμμικής παλινδρόμησης σε προβλήματα επιβλεπόμενης μάθησης. Από την άλλη, ο αλγόριθμος CoSyNe χρησιμοποιεί συνεργατική συνεξέλιξη για την προσαρμογή των βαρών προσχεδιασμένων τοπολογιών αναδραστικών νευρωνικών δικτύων και έχει πετύχει αξιόπιστα και ικανά αποτελέσματα στο πρόβλημα ενισχυτικής μάθησης ισορρόπησης ενός στύλου (pole balancing).

Σε όλες τις παραπάνω περιπτώσεις το μέγεθος του δικτύου είναι προκαθορισμένο. Αντίθετα, ένας από τους στόχους της μεθοδολογίας που αναπτύχθηκε στην παρούσα διατριβή ήταν να υπάρχει η δυνατότητα ατέρμονης εξέλιξης με αποτέλεσμα να δημιουργούνται δίκτυα τα οποία θα προσαρμόζονται στο εκάστοτε πρόβλημα αυτόνομα. Με τον τρόπο αυτό η χωρητικότητα του δικτύου από πλευράς νευρώνων θα πρέπει να καθορίζεται από την πολυπλοκότητα του εκάστοτε προβλήματος. Εξαιτίας των παραπάνω προδιαγραφών επιλέξαμε τη NEAT ως μέθοδο αναζήτησης. Τέλος, ένα ακόμα στοιχείο που διαχωρίζει την παρούσα έρευνα από τις προαναφερόμενες είναι οι συγκρίσεις που πραγματοποιήθηκαν με μια πλειάδα ανταγωνιστικών πρακτικών, τόσο σε προβλήματα πρόβλεψης χρονοσειρών (επιβλεπόμενη μάθηση), όσο και σε προβλήματα ενισχυτικής μάθησης, για να παραχθούν πιο ολοκληρωμένα συμπεράσματα. Αρκετοί από τους προαναφερθέντες αλγορίθμους βελτιστοποίησης δεν έχουν τόσο εκτεταμένες συγκρίσεις τόσο σε επίπεδο ανταγωνιστικών αλγορίθμων όσο και σε επίπεδο προβλημάτων.

Εκτός από τη βασική μέθοδο NEAT, οι εργασίες [WS06] και [RI09] είναι πιο σχετικές με τη μέθοδο NEAR. Στο [WS06] η NEAT χρησιμοποιείται προκειμένου να δημιουργήσει νευρωνικά δίκτυα χωρίς συνδέσεις ανάδρασης, δίνοντάς έτσι το πλεονέκτημα της εφαρμογής μάθησης χρονικών διαφορών μέσω του αλγορίθμου error back-propagation, ο οποίος και προσαρμόζει τα βάρη προς τη λύση. Από την άλλη μεριά στο [RI09], οι συγγραφείς χρησιμοποιούν τον δικό τους εξελικτικό αλγόριθμο TWEANN προκειμένου να αναπτύξουν ΔΗΚ με ανταγωνιστικά αποτε-

λέσματα σε προβλήματα πρόβλεψης χρονοσειρών.

Η μέθοδος NEAR διαφοροποιείται σε αρκετά σημεία σε σχέση με τις προαναφερθείσες προσεγγίσεις. Αρχικά, όλες οι παράμετροι (η τοπολογία της δεξαμενής, τα βάρη, ο αριθμός των νευρώνων της δεξαμενής N, η φασματική ακτίνα ρ και η συνδεσμική πυκνότητα D) βελτιστοποιούνται με τη χρήση νευροεξέλιξης και μάθησης με χρονικές διαφορές. Η πρακτική αυτή διαφοροποιεί τη NEAR από όλες τις μεθοδολογίες που βελτιστοποιούν μόνο τις μακροσκοπικές παραμέτρους της δεξαμενής N, ρ και D. Σχετικά με το [WS06], εκτός από τη βασική διαφορά ότι στην παρούσα περίπτωση αναπτύσσονται ΔΗΚ έναντι ad-hoc νευρωνικών δικτύων, μία ακόμη διαφοροποίηση είναι ότι τα ΔΗΚ διατηρούν μνήμη και έτσι είναι δυνατόν να δώσουν λύσεις σε προβλήματα με μη-Μαρκοβιανά σήματα κατάστασης. Επιπλέον τα δίκτυα εκπαιδεύονται χρησιμοποιώντας απλά γραμμική παλινδρόμηση αντί για back-propagation, προσδίδοντας έτσι τόσο θεωρητικά όσο και αλγοριθμικά οφέλη. Τέλος, η παρούσα μεθοδολογία διαφοροποιείται από αυτή στο [RI09] στο ότι: α) χρησιμοποιεί μια δοκιμασμένη πρακτική στη νευροεξέλιξη ως μέθοδο μετα-αναζήτησης, β) χρησιμοποιεί διαφορετικούς τελεστές και τεχνικές (για παράδειγμα δεν υπάρχει crossover τελεστής και δε γίνεται ομαδοποίηση σε είδη) και γ) το σημείο εστίασης είναι τα προβλήματα ενισχυτικής μάθησης και λιγότερο η πρόβλεψη χρονοσειρών. Με βάση τα παραπάνω παρατηρείται ότι μια συνέργεια όπως αυτή που αναπτύχθηκε στα πλαίσια της διατριβής δεν έχει διερευνηθεί στο παρελθόν.

Κλείνοντας την ενότητα της σχετικής βιβλιογραφίας και το παρόν κεφάλαιο, πρέπει να αναφερθούν και οι τεχνικές που έχουν αναπτυχθεί από την περιοχή της ενισχυτικής μάθησης στον τομέα της κατασκευής χαρακτηριστικών (feature construction) και των προσαρμοζόμενων συναρτήσεων προσέγγισης (adaptive function approximation). Πιο συγκεκριμένα, η αποδόμηση Fourier [KOT11] και το σφάλμα Bellman [PPWLL07] χρησιμοποιούνται για την επέκταση συναρτήσεων βάσης (basis expansion), παράγοντας γραμμικούς συνδυασμούς χαρακτηριστικών. Τα βάρη των γραμμικών αυτών συνδυασμών βρίσκονται μέσω του αλγορίθμου SARSA(λ) για την πρώτη μέθοδο και μέσω σταθμισμένης τοπικής παλινδρόμησης (locally weighted regression) ή μέσω Least Squares Policy Iteration (LSPI) [LP03] για τη δεύτερη. Και στις δύο περιπτώσεις, ο αριθμός των συναρτήσεων βάσης προκαθορίζεται, σε αντίθεση με τη μέθοδο NEAR, όπου καινούργια χαρακτηριστικά προστίθενται αυτόνομα ανάλογα με το πρόβλημα. Περισσότερο σχετικές με τη μεθοδολογία NEAR είναι οι εργασίες [GDR+11] και [GP08]. Στην πρώτη, οι καταστάσεις διακριτοποιούνται και διαμορφώνονται δυαδικά χαρακτηριστικά. Η μέθοδος προσπαθεί να συνδυάσει αυτά τα δυαδικά χαρακτηριστικά προκειμένου να κατασκευάσει πιο αποδοτικά χαρακτηριστικά. Η σταδιακή προσθήκη εξαρτημένων χαρακτηριστικών δείχνει ότι ενθαρρύνει την πρόωρη γενίκευση και επιταχύνει τη μάθηση, ένα αποτέλεσμα που δυναμώνει και την τέταρτη αρχή της NEAT, αυτή της σταδιακής περιπλοκής. Στην περίπτωση της NEAR δεν υπάρχει στάδιο διακριτοποίησης, αφού οι μεταβλητές κατάστασης τροφοδοτούν απευθείας το ΔΗΚ. Στη δεύτερη περίπτωση τα Cascade-correlation networks [GP08] χρησιμοποιούνται για την αυτόματη παραγωγή χαρακτηριστικών ώστε αυτά να χρησιμοποιηθούν μέσω

του αλγορίθμου LSPI για να δημιουργηθούν οι πολιτικές. Αυτή η προσέγγιση είναι παρόμοια με αυτή της NEAR με τη διαφορά ότι χρησιμοποιούμε το συνδυασμό ΔΗΚ και NEAT για την παραγωγή των χαρακτηριστικών. Βασική διαφορά είναι ότι τα cascade-correlation networks είναι εμπροσθοτροφοδοτούμενα δίκτυα και έτσι δεν μπορούν να παραχθούν χαρακτηριστικά που θα παρέχουν χρονική πληροφορία. Το ίδιο ισχύει και για όλους τους αλγορίθμους αυτής της παραγράφου οι οποίοι για να εξάγουν χαρακτηριστικά που θα περιέχουν και χρονικά επεξεργασμένη πληροφορία θα πρέπει αυτά να προσδιοριστούν ρητά εκ των προτέρων.

2.6 Σύνοψη

Στο κεφάλαιο αυτό παρουσιάστηκε μία εισαγωγή στις περιοχές της ενισχυτικής μάθησης, της εξελικτικής υπολογιστικής, της νευροεξέλιξης, του συνδυασμού μάθησης και εξέλιξης, των υπολογιστικών ταμιευτηρίων και ειδικότερα των δικτύων ηχωικών καταστάσεων και της μεθόδου NEAT. Αναλυτικότερη περιγραφή έγινε για τα ΔΗΚ και τη NEAT, καθώς αποτελούν βασικά συστατικά της μεθοδολογίας NEAR. Τέλος, έγινε αναφορά σε μεθόδους βελτιστοποίησης ταμιευτηρίων νευρώνων και κατασκευής χαρακτηριστικών σε προβλήματα ενισχυτικής μάθησης, καθώς και στις διαφοροποιήσεις της NEAR από τις μεθόδους αυτές.

Κεφάλαιο 3

Η μέθοδος NEAR

Στο κεφάλαιο αυτό παρουσιάζεται η μέθοδος *NeuroEvolution of Augmented Reservoirs* (NEAR). Η NEAR χρησιμοποιεί τη NEAT ως έναν αλγόριθμο μεταναζήτησης και τον προσαρμόζει στις ιδιαιτερότητες των ΔΗΚ. Με αυτόν τον τρόπο η NEAR καταφέρνει να προσαρμόζει τα ΔΗΚ στο εκάστοτε πρόβλημα μέσω εξέλιξης και μάθησης. Τα βασικά βήματα της μεθόδου παρουσιάζονται στο Σχήμα 3.1 και αναλύονται λεπτομερώς στο παρόν κεφάλαιο. Συνοπτικά, για την αναπαράσταση του γονιδιώματος της μεθόδου επιλέχθηκε η απευθείας κωδικοποίηση (Ενότητα 3.1). Στη συνέχεια τα βήματα της μεθόδου έχουν ως εξής:

1. Αρχικοποιείται ο πληθυσμός (Ενότητα 3.2)

2. Ξεκινάει η επαναληπτική διαδικασία της εξέλιξης του πληθυσμού (Ενότητα 3.3) σύμφωνα και με τον Αλγόριθμο 2.5.

    i. Κάθε γονιδίωμα του πληθυσμού αξιολογείται. Στο σημείο αυτό μπορεί να υπάρχει, είτε πριν, είτε κατά τη διάρκεια της αξιολόγησης, κάποια διαδικασία μάθησης για τη βελτιστοποίηση των βαρών των ΔΗΚ. (Ενότητα 3.4)

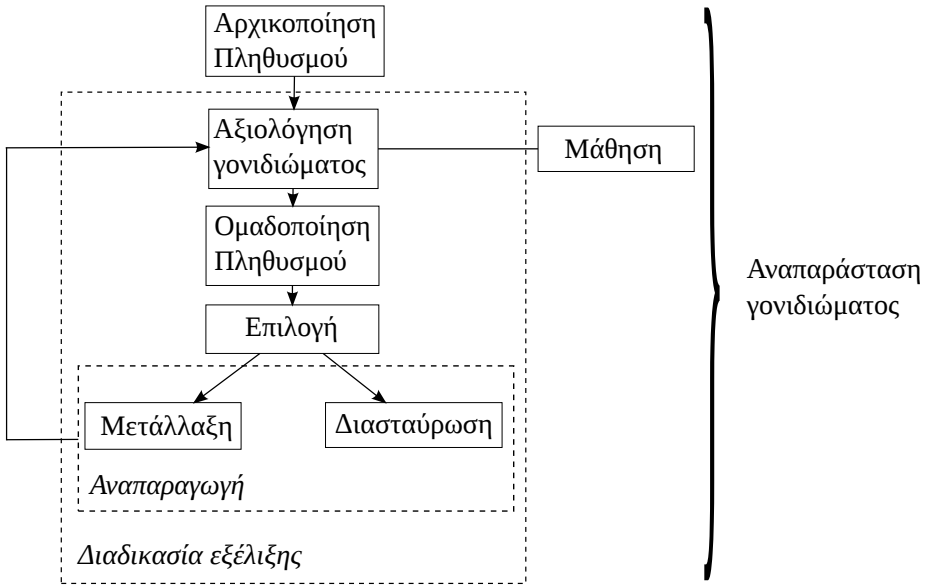
    ii. Ο πληθυσμός ομαδοποιείται σε είδη (Ενότητα 3.3.1)

    iii. Πραγματοποιείται ο καθορισμός των απογόνων ανά είδος και η επιλογή των γονιδιωμάτων για τον καθορισμό της επόμενης γενιάς. (Ενότητα 3.3.2)

    iv. Δημιουργούνται οι απόγονοι βάσει των διαδικασιών της μετάλλαξης (Ενότητα 3.3.3) και της διασταύρωσης (Ενότητα 3.3.4).

    v. Επιστροφή στο βήμα i για την αξιολόγηση της νέας γενιάς

Οι τυπικές τιμές των παραμέτρων που παρουσιάζονται στο παρόν κεφάλαιο είναι καταγεγραμμένες στο Παράρτημα A.



Σχήμα 3.1: Η μέθοδος NEAR.

3.1 Αναπαράσταση γονιδιώματος

Κάθε γονιδίωμα (genome) G στη μεθοδολογία αποτελείται από τα ακόλουθα γονίδια (genes):

- $W^{in} \in \mathbb{R}^{NK}$, τα βάρη εισόδου των συνδέσεων μεταξύ των κόμβων εισόδου και των κόμβων του ταμιευτηρίου,
- $W \in \mathbb{R}^{N^2}$, ο πίνακας γειτνίασης του ταμιευτηρίου,
- $W^{out} \in \mathbb{R}^{N(K+L)}$, τα βάρη εξόδου των συνδέσεων μεταξύ των κόμβων εισόδου και ταμιευτηρίου με τους κόμβους εξόδου, δηλαδή τα βάρη της συνάρτησης ανάγνωσης,
- $D \in \mathbb{R}$, η πυκνότητα του ταμιευτηρίου, και
- $\rho \in \mathbb{R}$, η φασματική ακτίνα με βάση την οποία κανονικοποιούνται τα βάρη του ταμιευτηρίου.

Όπως φαίνεται έμμεσα από την παραπάνω λίστα, χρησιμοποιήθηκε απευθείας κωδικοποίηση, ώστε να υπάρχει ένα-προς-ένα αντιστοίχηση με τις παραμέτρους του δικτύου και τη γενετική κωδικοποίηση κάθε γονιδιώματος. Επίσης, πρέπει να σημειωθεί ότι το γονιδίωμα περιέχει τον πίνακα W προτού αυτός κανονικοποιηθεί με βάση τη μέγιστη ιδιοτιμή, στη φασματική ακτίνα ρ. Η διαδικασία αυτή επιβάλλεται όταν ο γονότυπος εκφραστεί στο φαινότυπο, δηλαδή ως ΔΗΚ. Η απευθείας

κωδικοποίηση επιλέχθηκε διότι είναι εύκολη στην υλοποίηση [Yao99], ενώ ταυτόχρονα δίνεται η δυνατότητα επεξεργασίας και μη-αραιών ΔΗΚ, εφόσον τα προβλήματα υπό εξέταση δεν έχουν παράλογες απαιτήσεις σε μνήμη ($O(N^2)$, αφού $N >> K, L$ όπου N ο αριθμός νευρώνων του ταμιευτηρίου). Ένα γραμμικό γονιδίωμα, όπως αυτό της NEAT, θα μπορούσε να εξεταστεί εάν χρησιμοποιούνταν αναπαράσταση αραιού πίνακα για τον πίνακα γειτνίασης W [1]. Επίσης, θα διερευνηθεί σε ποιο βαθμό πυκνά ΔΗΚ είναι ικανά να οδηγήσουν σε δίκτυα υψηλών επιδόσεων, όπως έχει αναφερθεί σε άλλη εργασία [RI09]. Για αυτούς τους δύο λόγους επιλέχθηκε η απευθείας κωδικοποίηση με απαιτήσεις μνήμης $O(N^2)$.

3.2 Αρχικοποίηση πληθυσμού

Τα ταμιευτήρια του αρχικού πληθυσμού περιέχουν τουλάχιστον ένα νευρώνα, $N = 1$, έτσι ώστε να έχουν τη δυνατότητα επίλυσης του προβλήματος με τη μικρότερη μη-γραμμικότητα, του τελεστή XOR. Τα βάρη στους πίνακες W^{in} και W^{out} αρχικοποιούνται τυχαία στο διάστημα $[-1, 1]$. Η πυκνότητα D αρχικοποιείται τυχαία στο $[D_{min}, D_{max}] \in (0, 1]$ και εκφράζει το ποσοστό των ενεργών, μη-μηδενικών συνδέσεων στο ταμιευτήριο. Η φασματική ακτίνα ρ επιλέγεται τυχαία στο $[\rho_{min}, \rho_{max}] \in (0, 1)$. Οι συνδέσεις στον πίνακα W επιλέγονται στο διάστημα $[-1, 1]$ βάσει του ποσοστού D. Ένας καταχωρητής κρατά το τρέχον άθροισμα $r = \sum_i^N \sum_j^N w_{ij}$ των βαρών στο ταμιευτήριο έτσι ώστε να διατηρείται η μέση τιμή των τιμών των βαρών περίπου στο 0 κατά τη διάρκεια των μεταλλάξεων. Θεωρούμε ότι όλες οι συνδέσεις υπάρχουν, αλλά μόνο οι μη μηδενικές είναι ενεργοποιημένες.

3.3 Διαδικασία εξέλιξης

Μετά τη φάση της αρχικοποίησης, κάθε γονιδίωμα στον πληθυσμό μετατρέπεται σε ΔΗΚ και περνά στη φάση της αξιολόγησης (Αλγόριθμος 2.4). Το δίκτυο αξιολογείται για έναν αριθμό επεισοδίων και προσαρμόζεται είτε με μάθηση χρονικών διαφορών είτε με κάποιον αλγόριθμο αναζήτησης πολιτικής (Κεφάλαιο 2). Η επίδοση κάθε οργανισμού καταγράφεται ως ικανότητα και με την ολοκλήρωση όλων των αξιολογήσεων, εφαρμόζεται το εξελικτικό βήμα (Αλγόριθμος 2.5).

Ακολουθώντας το [WS06], υλοποιήθηκαν τόσο η Λαμαρκιανή όσο και η Δαρβίνια φιλοσοφία της εξέλιξης. Στην πρώτη περίπτωση, το προσαρμοζόμενο μέρος W^{out} μεταφέρεται από γενιά σε γενιά μετά την εφαρμογή των εξελικτικών τελεστών, ενώ στη δεύτερη περίπτωση τα βάρη επαναρχικοποιούνται σε κάθε γενιά. Αν και η Λαμαρκιανή εξέλιξη διαφαίνεται ως η καλύτερη μέθοδος, θα μπορούσε κανείς να υποστηρίξει ότι η μάθηση απλά βοηθάει την εξέλιξη να επιλέξει το καλύτερο γονιδίωμα για αναπαραγωγή το οποίο ίσως να μαθαίνει καλύτερα και γρηγορότερα, δίχως ωστόσο τα βάρη να μεταφέρονται στην επόμενη γενιά. Αυτό είναι μια περίπτωση του φαινομένου Baldwin.

[1] Στην περίπτωση αυτή θα έπρεπε να χρησιμοποιηθεί ο αλγόριθμος power iteration για την εύρεση της μέγιστης ιδιοτιμής σε αραιούς πίνακες κατά τη μετάβαση από το γονότυπο στο φαινότυπο

3.3.1 Ομαδοποίηση σε είδη

Για την ομαδοποίηση σε είδη των γονιδιωμάτων και εφόσον ο πίνακας W μπορεί να είναι αραιός, δε χρειάζεται να χρησιμοποιήσουμε δομικές ομοιότητες μεταξύ των οργανισμών, όπως η NEAT (η ομοιομορφία σε εκείνη την περίπτωση έχει σχέση με τα ταιριασμένα, αταίριαστα και επιπλέον γονίδια), οπότε προχωράμε στην υλοποίηση ενός μέτρου σύγκρισης βασισμένου σε μακροσκοπικά χαρακτηριστικά του ταμιευτηρίου (Εξίσωση 3.1):

$$\delta(x,r) = \frac{c_\alpha|\rho_x - \rho_r|}{\rho_r} + \frac{c_\beta|D_X - D_r|}{D_r} + \frac{c_\gamma|N_x - N_r|}{N_r} \tag{3.1}$$

όπου το r σημειώνει τον αντιπρόσωπο του είδους, ενώ οι παράγοντες c_α, c_β και c_γ είναι προκαθορισμένες παράμετροι. Εαν $\delta < T$, όπου T ένα προκαθορισμένο κατώφλι, τότε το γονιδίωμα προστίθεται στο είδος.

3.3.2 Επιλογή

Η διαδικασία της επιλογής δε διαφέρει από αυτή της NEAT. Για λόγους πληρότητας της παρουσίασης της μεθόδου NEAR, αναφέρεται και στο σημείο αυτό. Για τον υπολογισμό της αξιολόγησης κάθε γονιδιώματος χρησιμοποιείται η προσαρμοσμένη αξιολόγηση (adjusted fitness):

$$f'_i = \frac{f_i}{\sum_{j=1}^{n} sh(\delta(i,j))}$$

όπου $sh(\delta(i,j)) = 1$ εάν $\delta < T$ και $sh(\delta(i,j)) = 0$ εάν $\delta > T$.

Σε κάθε είδος, k ανατίθεται ένας αριθμός απογόνων που υπολογίζεται από την Εξίσωση 3.2:

$$n_k = \frac{\bar{F}_k}{F_{tot}}|P| \tag{3.2}$$

όπου $\bar{F}_k$ η μέση αξιολόγηση του είδους k, $|P|$ το πλήθος του πληθυσμού και $F_{tot} = \sum_k \bar{F}_k$.

Τα γονιδιώματα πρωταθλητές κάθε είδους, δηλαδή τα γονιδιώματα με τις καλύτερες επιδόσεις στο είδος τους, μεταφέρονται στην επόμενη γενιά ως έχουν (ελιτισμός). Στη συνέχεια, από τα υπόλοιπα γονιδιώματα που ξεπερνούν το κατώφλι επιλογής (S_{thres}), επιλέγονται τα γονιδιώματα γονείς. Με τις διαδικασίες της μετάλλαξης και της διασταύρωσης δημιουργείται το γονιδίωμα απόγονος και η διαδικασία αυτή συνεχίζει για κάθε είδος, έως ότου συμπληρωθεί ο αριθμός απογόνων που του ανατέθηκε.

3.3.3 Μετάλλαξη

Ο τελεστής μετάλλαξης σχετίζεται τόσο με την τοπολογία (προσθήκη κόμβου ή σύνδεσης), όσο και με τα βάρη (επανεκκίνηση ή διαταραχή των τιμών των βαρών). Πιο συγκεκριμένα, η NEAR προβλέπει τους παρακάτω τελεστές οι οποίοι

εκτελούνται σειριακά και εφαρμόζονται ανάλογα με την πιθανότητα εφαρμογής τους.

`mutate-D` και `mutate-`ρ Σύμφωνα με τις πιθανότητες p_D και p_ρ, η συνδεσμική πυκνότητα D και η φασματική ακτίνα ρ διαταράσσονται μέσα στα όρια τους (ποσοστό μεταβολής 5%) είτε επανεκκινούνται με μια μικρή πιθανότητα (0.05).

`mutate-weights` Δεδομένης μιας πιθανότητας p_w, κάθε βάρος στους πίνακες W^{in} και W μεταλλάσσεται, είτε διαταράσσοντας την τιμή του:

$$w' = w - sgn(r) \cdot p \cdot w_{pow}$$

είτε γίνεται επανεκκίνησή του με ίση πιθανότητα:

$$w' = -sgn(r) \cdot p \cdot w_{pow}$$

όπου w_{pow} είναι η δύναμη της μετάλλαξης. Για τα βάρη στο W, το r είναι η τιμή στον αθροιστικό καταχωρητή και p τυχαίος αριθμός στο $[0, 1]$, ενώ για τα βάρη στον πίνακα W^{in}, το $r = -1$ και p ένας τυχαίος αριθμός στο $[-1, 1]$. Στη Λαμαρκιανή εξέλιξη τα βάρη εξόδου επίσης μεταλλάσσονται όπως τα βάρη του W^{in}. Διευκρινίζεται στο σημείο αυτό ότι μόνο ενεργοποιημένες συνδέσεις του W μεταλλάσσονται.

`add-node` Με πιθανότητα p_n, ένας κόμβος προστίθεται στο ταμιευτήριο και όλες οι συνδέσεις του, προς και από τον κόμβο αυτό, αρχικά απενεργοποιούνται θέτοντας το βάρος τους 0. Οι πίνακες γειτνίασης αλλάζουν ως εξής:

$$W^{in'} \in \mathbb{R}^{(N+1)K},$$
$$W' \in \mathbb{R}^{(N+1)^2}$$

και

$$W^{out'} \in \mathbb{R}^{L(K+N+1)}$$

με

$$w_{ij}^{in'} = (2 \cdot p - 1) \cdot w_{pow}, i > N$$

και

$$w'_{ij} = 0, i, j > N,$$
$$w_{ij}^{out'} = 0, j > N.$$

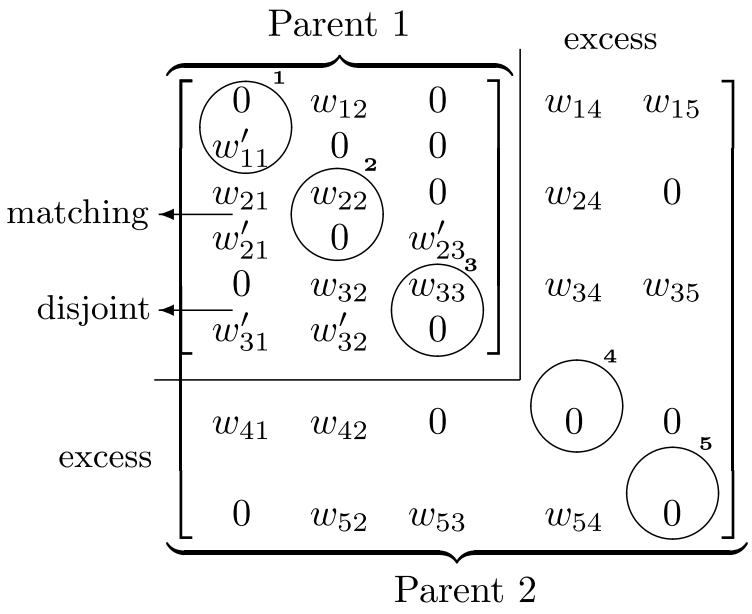
Για το ταμιευτήριο, τα βάρη των συνδέσεων προς και από το νέο κόμβο βρίσκονται σε όλη τη σειρά $N + 1$ και σε όλη τη στήλη $N + 1$ του W.

`add-connection` Με πιθανότητα p_c, μια σύνδεση ενεργοποιείται και ένα βάρος αρχικοποιείται στο ταμιευτήριο σε σχέση με τον αθροιστικό καταχωρητή που διατηρείται από το γονιδίωμα:

$$w'_{ij} = w_{ij} - sgn(r) \cdot p \cdot w_{pow}$$

3.3.4 Διασταύρωση

Η διασταύρωση είναι ένα βασικό χαρακτηριστικό τόσο της NEAT όσο και της NEAR. Αν και υπάρχουν κάποια επιχειρήματα που εναντιώνονται στη χρήση της στη νευροεξέλιξη, λόγω της παρουσίασης των ανταγωνιστικών αναπαραστάσεων, οι [HN10] υπολόγισαν ότι το πρόβλημα δεν είναι τόσο σημαντικό όσο αρχικά θεωρήθηκε. Επιπλέον στο [LPDF09] παρουσιάζονται εκτενή αποτελέσματα πειραμάτων όπου εμφανίζεται αύξηση της ευρωστίας των λύσεων που προσφέρονται όταν χρησιμοποιείται η διασταύρωση. Ακολουθώντας την ιστορική σηματοδότηση, κάθε φορά που προστίθεται ένας νέος κόμβος, του ανατίθεται ένας αριθμός καινοτομίας ίσος με τη θέση του στη διαγώνιο του πίνακα W, προσδίδοντάς του το σχετικό χρονικό διάστημα κατά το οποίο προστέθηκε ο νευρώνας στην ιστορία του συγκεκριμένου γονιδιώματος. Στη συνέχεια, κατά τη διάρκεια της διαδικασίας της διασταύρωσης, οι πίνακες W και W^{in} (καθώς και ο πίνακας W^{out} εφόσον ακολουθείται η Λαμαρκιανή εξέλιξη), ευθυγραμμίζονται με βάση τους αριθμούς καινοτομίας. Ένα παράδειγμα για τον πίνακα W δίνεται στο Σχήμα 3.2.



Σχήμα 3.2: Η ευθυγράμμιση δύο ταμιευτηρίων διαφορετικών μεγεθών. Οι κύκλοι αναπαριστούν εικονικά τους κόμβους του δικτύου, δεικτοδοτημένους από τους αριθμούς καινοτομίας στην πάνω δεξιά μεριά κάθε κύκλου.

Αρχικά θα πρέπει να καθοριστεί το μέγεθος του ταμιευτηρίου του απογόνου. Υπάρχουν δύο εκδοχές: (i) να συνεχιστεί η περίπλεξη του δικτύου και να επιλεγεί το μεγαλύτερο από τα δύο και (ii) να επιλεγεί το μέγεθος του καταλληλότερου γονέα. Υλοποιήθηκαν και οι δύο εκδοχές και κάθε μία επιλέγεται με τυχαίο τρόπο κάθε φορά.

Με βάση το μέγεθος του ταμιευτηρίου, τα επιπλέον βάρη (excess) είτε υιοθετούνται, είτε απορρίπτονται. Για τις συνδέσεις που ταυτίζονται (matching) καθώς και για τις αταίριαστες συνδέσεις (disjoint), ισχύουν οι κανόνες της Εξίσωσης 3.3:

$$w'_{ij} = \begin{cases} \frac{w^1_{ij}+w^2_{ij}}{2} & \text{if } w^1_{ij}, w^2_{ij} \neq 0 \text{ and } p < 0.5 \\ w^1_{ij} & \text{if } w^1_{ij}, w^2_{ij} \neq 0 \text{ and } 0.5 \leq p < 0.75 \\ w^2_{ij} & \text{if } w^1_{ij}, w^2_{ij} \neq 0 \text{ and } 0.75 \leq p < 1.0 \\ w^1_{ij} & \text{if } w^2_{ij} = 0 \text{ and } w^1_{ij} \neq 0 \\ w^2_{ij} & \text{if } w^1_{ij} = 0 \text{ and } w^2_{ij} \neq 0 \end{cases} \tag{3.3}$$

όπου p είναι ένας τυχαίος αριθμός ανάμεσα στο 0 και στο 1, ενώ ο άνω δείκτης σηματοδοτεί το γονέα από τον οποίο προήλθε το βάρος.

Όσον αφορά στα βάρη εισόδου και εξόδου, τα περιττά βάρη εξαρτώνται από την τελική τιμή που λαμβάνει το N, ενώ τα υπόλοιπα, είτε αθροίζονται, είτε επιλέγονται τυχαία από τον ένα γονέα ισοπίθανα. Τα D και ρ κληρονομούνται από τον επιλεγμένο γονέα.

Κάθε απόγονος επαληθεύεται πριν δοθεί προς αξιολόγηση, έτσι ώστε το D να ταυτίζεται και με την πραγματική τιμή πυκνότητας του πίνακα γειτνίασης $\tilde{D}$, έπειτα από την εφαρμογή της μετάλλαξης και της διασταύρωσης. Από τη διαφορά $D - \tilde{D}$ υπολογίζεται μια πιθανότητα σύμφωνα με την οποία αποφασίζεται αν θα προστεθούν ή θα αφαιρεθούν συνδέσεις, έτσι ώστε $D \approx \tilde{D}$.

Προσαρμοζόμενη διασταύρωση

Όταν εφαρμόζεται ο τελεστής της διασταύρωσης στη μεθοδολογία NEAT επιλέγεται ο καταλληλότερος εκ των δύο οργανισμών και αναλόγως κρατούνται ή απομακρύνονται οι περιττοί νευρώνες. Κατά την ανάπτυξη της μεθοδολογίας NEAR, παρατηρήθηκε ότι διατηρώντας πάντα τον αριθμό των κόμβων που κατέχει το καταλληλότερο γονιδίωμα, ο αλγόριθμος μπορεί να οδηγηθεί σε στασιμότητα. Για να ξεπεραστεί το πρόβλημα αυτό, επιλέγεται ισοπίθανα μία από τις δύο στρατηγικές διασταύρωσης, δηλαδή γίνεται η επιλογή του καταλληλότερου ή του μεγαλύτερου γονέα ως μήτρα για τον απόγονο.

Η αποδοτικότητα ενός αλγορίθμου εξελικτικής υπολογιστικής, και στη συγκεκριμένη περίπτωση της NEAR, μπορεί να βελτιωθεί προσαρμόζοντας τις παραμέτρους της στρατηγικής του αλγορίθμου [WS06, RI09]. Με τον τρόπο αυτό, για να βελτιστοποιηθεί η στρατηγική επιλογής μίας εκ των δύο επιλογών, προσαρμόζουμε την πιθανότητα επιλογής, μία στρατηγική που ακολουθήθηκε και από τον [RI09], όπου ο τελεστής μετάλλαξης αλλάζει άμεσα με βάση την επίδοση του συγκεκριμένου τελεστή στο συγκεκριμένο πρόβλημα.

Ακολουθώντας τη σημειογραφία στο [RI09], υπάρχει ένα σετ από πιθανούς τελεστές διασταύρωσης $\Omega = \{fittest, largest\}$ με $p_1^{(k)}, p_2^{(k)} = 1 - p_1^{(k)}$ που είναι οι πιθανότητες επιλογής, είτε του τελεστή $fittest$, είτε του $largest$ στη γενιά k. Τα $O_i\ (k)$ με $i = 1, 2$ σημειώνουν τη δημιουργία ενός απογόνου με την εφαρμογή του συγκεκριμένου τελεστή. Η μέση ποιότητα κάθε τελεστή δίνεται από την εξίσωση:

$$q_i^{(k)} = \frac{1}{|O_i\ (k)|} \sum_{g \in O_i\ (k)} max[0, \Phi(g) - max[\Phi(par_1(g)), \Phi(par_2(g))]] \quad (3.4)$$

όπου το $par(g)$ σημειώνει τον γονέα του οργανισμού g, ενώ Φ είναι η συνάρτηση καταλληλότητας. Οι πιθανότητες των τελεστών μπορούν να βρεθούν κάνοντας τους παρακάτω υπολογισμούς:

$$s_i^{(k+1)} = \begin{cases} \frac{C_\Omega \cdot q_i^{(k)}}{q_{all}^{(k)}} + (1 - C_\Omega) \cdot s_i^{(k)}, & \text{if } q_{all}^{(k)} > 0 \\ \frac{C_\Omega}{|\Omega|} + (1 - C_\Omega) \cdot s_i^{(k)}, & \text{else} \end{cases} \quad (3.5)$$

ενώ στη συνέχεια:

$$p_i^{(k+1)} = p_{min} + \frac{(1 - |\Omega| \cdot p_{min}) s_i^{(k+1)}}{\sum_{i' \in \Omega} s_{i'}^{(k+1)}} \quad (3.6)$$

με $q_{all}^{(k)} = \sum_{i' \in \Omega} q_{i'}^{(k)}$, $C_\Omega \in (0, 1]$ μια παράμετρο η οποία ελέγχει τη μείωση και p_{min} ένα όριο στην πιθανότητα. Επιλέχθηκε $C_\Omega = 0.5$ και $p_{min} = 0.1$, ενώ $s_1^{(0)} = s_2^{(0)} = 1$.

Όπως προαναφέρθηκε, μια προσέγγιση θα μπορούσε να είναι η επιλογή μιας εκ των δύο στρατηγικών ισοπίθανα.

3.4 Μάθηση

Στόχος της μάθησης είναι να προσαρμόσει τα βάρη της συνάρτησης ανάγνωσης έτσι ώστε η πρόβλεψη που γίνεται στο επίπεδο των νευρώνων εξόδου να έχει το μικρότερο δυνατό σφάλμα σε σχέση με την επιθυμητή τιμή, είτε αυτή είναι πρόβλεψη μιας χρονοσειράς, είτε είναι πρόβλεψη του εκτιμώμενου κέρδους σε ένα πρόβλημα ενισχυτικής μάθησης.

Για τη μεταβολή των βαρών μπορούν να χρησιμοποιηθούν επαναληπτικές μέθοδοι (iterative), αλλά και μέθοδοι που επεξεργάζονται δέσμες δεδομένων (batch). Οι πρώτες μπορούν να χρησιμοποιούν και το δίκτυο καθώς μαθαίνουν, ενώ οι δεύτερες θα πρέπει να συλλέγουν δεδομένα προκειμένου να αρχίσει η εκμάθηση. Για παράδειγμα, στην πρώτη κατηγορία ανήκουν η μέθοδος κατάβασης πλαγιάς (gradient descent) και η αναδρομική μέθοδος ελαχίστων τετραγώνων (recursive least squares) [FB98], ενώ στη δεύτερη κατηγορία ανήκει η μέθοδος ελαχίστων τετραγώνων (least squares). Οι παραπάνω μέθοδοι μπορούν να χρησιμοποιηθούν τόσο σε προβλήματα ενισχυτικής μάθησης όσο και σε προβλήματα επιβλεπόμενης μάθησης. Ένα παράδειγμα παρουσιάζεται στον Αλγόριθμο 2.1, όπου χρησιμοποιείται η μέθοδος κατάβασης πλαγιάς για τη μεταβολή των βαρών.

3.5 Σύνοψη και σύγκριση NEAT και NEAR

Στο παρόν κεφάλαιο παρουσιάστηκε η μέθοδος NEAR και διατυπώθηκε η επιχειρηματολογία για τις αποφάσεις σχεδίασης. Καταγράφηκαν με λεπτομέρεια οι τελεστές μετάλλαξης και διασταύρωσης, η διαδικασία ομαδοποίησης, διάφοροι αλγόριθμοι μάθησης που χρησιμοποιήθηκαν, η αρχικοποίηση του πληθυσμού καθώς και η μετάβαση από γονότυπο σε φαινότυπο.

Ο Πίνακας 3.1 παρουσιάζει τις διαφοροποιήσεις μεταξύ NEAR και NEAT. Οι βασικές τους διαφορές συνοψίζονται στις εξής:

1. Η NEAR μπορεί να συνδυάσει με την εξέλιξη οποιοδήποτε τρόπο μάθησης. Αυτή η δυνατότητα προσθέτει μεγάλη ευελιξία στη NEAR και την καθιστά εξαιρετικό εργαλείο μελέτης της συνέργειας των δύο προσεγγίσεων.

2. Τα γραμμικά χαρακτηριστικά που υπάρχουν στη NEAR και γενικότερα στα ΔΗΚ, έχοντας τις εισόδους σε απευθείας σύνδεση με την έξοδο, δεν έχουν προβλεφθεί στη NEAT

3. Η NEAR εξελίσσει αποκλειστικά ΔΗΚ μεταφέροντας τις ιδιότητες, τη θεωρία, τα πλεονεκτήματα και τα μειονεκτήματά τους, ενώ η NEAT εξελίσσει αδόμητα δίκτυα.

4. Η διασταύρωση στη NEAR γίνεται σε επίπεδο κόμβων και όχι συνδέσεων καθώς λαμβάνοντας υπόψη την πιθανή ύπαρξη αραιών πινάκων σε ένα ΔΗΚ, η πιθανότητα κοινών συνδέσεων θα μειωνόταν δραματικά, πιθανότατα εκφυλίζοντας και την έννοια της ιστορικής δεικτοδότησης.

5. Για τον ίδιο λόγο, η ομαδοποίηση σε είδη γίνεται σε επίπεδο μακρο-παραμέτρων, όπως ο αριθμός των νευρώνων του ταμιευτηρίου, η πυκνότητα του ταμιευτηρίου και η φασματική του ακτίνα.

Πίνακας 3.1: Ομοιότητες και διαφορές μεταξύ NEAT και NEAR.

Ιδιότητα	**NEAT**	**NEAR**
TWEANN	$\surd$	$\surd$
Μη γραμμικότητα	$\surd$	$\surd$
Αναδράσεις	$\surd$	$\surd$
Ομαδοποίηση σε είδη	$\surd$ (Με βάση μίκρο-παραμέτρους)	$\surd$ (Με βάση μάκρο-παραμέτρους)
Διασταύρωση	$\surd$ (Στις συνδέσεις) (Υιοθεσία του καταλληλότερου)	$\surd$ (Στους κόμβους) (Επιλογή μεγαλύτερου ή καταλληλότερου)
Αένναο	$\surd$	$\surd$
Περίπλεξη	$\surd$ (Κόμβοι)	$\surd$ (Κόμβοι)
Αρχικοί Κρυμμένοι Νευρώνες	0	1
Μάθηση	$\surd$ (Hebbian)	$\surd$ (Πλήθος)
Δομή	Ad-hoc	ΔΗΚ
Γραμμικά Χαρακτηριστικά	X	$\surd$

Κεφάλαιο 4

Αξιολόγηση επιδόσεων και συμπεριφοράς

Για τη βασική αξιολόγηση των επιδόσεων της NEAR, επιλέχθηκαν τα εξής τρία προβλήματα ενισχυτικής μάθησης: α) διαφορετικά σενάρια του προβλήματος ανίσχυρου αυτοκινήτου πλαγιάς, β) διαφορετικές εκδοχές εξισορρόπησης ράβδου, και γ) χρονοπρογραμματισμός εργασιών σε έναν εξυπηρετητή από την περιοχή των αυτόνομων υπολογιστικών συστημάτων (autonomic computing).

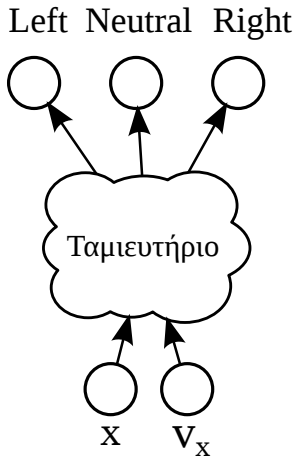
Για τη σειρά των πειραμάτων αυτών, αξιολογήθηκαν οι ακόλουθες μέθοδοι:

1. NEAT,
2. NEAR με μάθηση χρονικών διαφορών με κατάβαση πλαγιάς και Λαμαρκιανή εξέλιξη (NEAR+TD-L),
3. NEAR με μάθηση χρονικών διαφορών με κατάβαση πλαγιάς και Δαρβίνια εξέλιξη (NEAR+TD-D),
4. NEAR με αναζήτηση πολιτικής μέσω μετάλλαξης των βαρών (NEAR+PS) και
5. πολλαπλά ΔΗΚ δημιουργημένα με τυχαίο τρόπο και μάθηση χρονικών διαφορών μέσω κατάβασης πλαγιάς (ESN-TD).

Επίσης, πραγματοποιήθηκαν συγκρίσεις μεταξύ της NEAR και αποτελεσμάτων καταχωρημένων στη σχετική βιβλιογραφία [WS06, GSM06].

4.1 Αυτοκίνητο πλαγιάς

Για το πρόβλημα του αυτοκινήτου πλαγιάς (mountain car), χρησιμοποιήθηκε η έκδοση που περιγράφεται στο [SS96]. Στο κλασικό πρόβλημα 2 διαστάσεων, ένα αυτοκίνητο χαμηλής ιπποδύναμης πρέπει να ανέβει ένα λόφο. Η κατάσταση

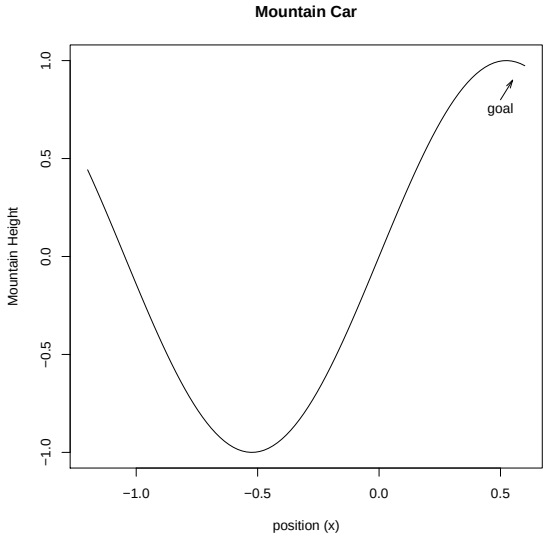


Σχήμα 4.1: Σχηματική αναπαράσταση ενός ΔΗΚ για το πρόβλημα 2-D mountain car.

του περιβάλλοντος περιγράφεται από 2 συνεχείς μεταβλητές, την οριζόντια θέση $x \in [-1.2, 0.6]$, και την ταχύτητα, $v_x \in [-0.07, 0.07]$. Οι ενέργειες είναι {*Αριστερά (Left - L), Νεκρά (Neutral - Nt) και Δεξιά (Right - R)*}, οι οποίες μεταβάλλουν την ταχύτητα κατά $-0.001, 0$ και 0.001 αντίστοιχα. Σε κάθε χρονικό βήμα η ποσότητα $-0.0025 * \cos 3x$ προστίθεται στην ταχύτητα, προκειμένου να αναπαρασταθεί η βαρύτητα. Μία σχηματική αναπαράσταση ενός ΔΗΚ για το πρόβλημα αυτό υπάρχει στο Σχήμα 4.1.

Στη συγκεκριμένη περίπτωση που εξετάστηκε, κάθε επεισόδιο ξεκινά με το όχημα να βρίσκεται σε τυχαία θέση στην πλαγιά, έχοντας τυχαία ταχύτητα και τελειώνει όταν το x γίνει μεγαλύτερο από 0.5. Σε κάθε χρονικό βήμα, ο πράκτορας που επιλύει το πρόβλημα πρέπει να επιλέξει ανάμεσα στις τρεις ενέργειες και λαμβάνει ανταμοιβή -1. Με άλλα λόγια, όσο περισσότερο παραμένει το αυτοκίνητο στην πλαγιά, τόσο η συνολική ανταμοιβή θα μειώνεται. Στόχος είναι η μετακίνηση του αυτοκινήτου στην κατάσταση-στόχο το γρηγορότερο δυνατό. Η βασική δυσκολία σε αυτό το περιβάλλον είναι ότι η δύναμη της βαρύτητας είναι πολύ μεγαλύτερη της ιπποδύναμης της μηχανής του αυτοκινήτου που δύναται αυτό να καταβάλλει. Εξαιτίας αυτού, ο πράκτορας θα πρέπει να εκτελέσει ταλάντωση μέσα στην πλαγιά ώστε να εκμεταλλευτεί προς όφελός του το βαρυτικό πεδίο και να φτάσει στην κατάσταση στόχο (Σχήμα 4.2).

Στην τρισδιάστατη εκδοχή (Σχήμα 4.3) επεκτείνεται το δισδιάστατο πρόβλημα με την προσθήκη μίας ακόμα χωρικής διάστασης [TKS08]. Το διάνυσμα καταστάσεων περιέχει τέσσερις μεταβλητές, τις χωρικές συντεταγμένες x και $y \in [-1.2, 0.6]$ και τις συνιστώσες της ταχύτητας v_x και $v_y \in [-0.07, 0.07]$. Οι επιτρεπτές ενέργειες για τον πράκτορα είναι {*Νεκρά (Neutral - Nt), Δυτικά (West - W), Ανατολικά (East - E), Νότια (North - N), Βόρεια (South - S)*} (Σχήμα 4.4). Κάθε ενέργεια *Νεκρά* δεν έχει καμία επίπτωση στην ταχύτητα του αυτοκινήτου. Αντίθετα, οι ενέργειες *Δυτικά* και *Ανατολικά* προσθέτουν -0.001 και 0.001 στην v_x

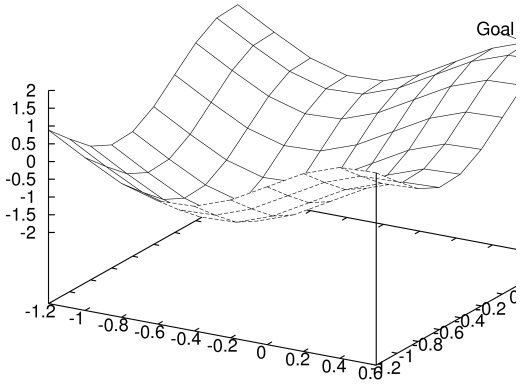


Σχήμα 4.2: Σχηματική αναπαράσταση του προβλήματος 2-D mountain car. Η γραμμή αντιπροσωπεύει την κοιλάδα μέσα στην οποία το όχημα πρέπει να κινηθεί μπρος-πίσω ώστε να φτάσει στην κορυφή.

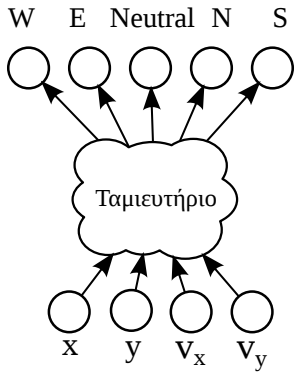
αντίστοιχα, ενώ οι ενέργειες *Νότια* και *Βόρεια* προσθέτουν στην v_y -0.001 και 0.001 αντίστοιχα. Επιπλέον, για την προσομοίωση του φαινομένου της βαρύτητας, σε κάθε χρονικό βήμα, οι ποσότητες $-0.0025 * \cos 3x$ και $-0.0025 * \cos 3y$ προστίθενται στις v_x και v_y αντίστοιχα. Όπως και στη δισδιάστατη περίπτωση, κάθε επεισόδιο ξεκινά από τυχαία θέση με τυχαία ταχύτητα. Η κατάσταση-στόχος επιτυγχάνεται όταν $x \geq 0.5$ και $y \geq 0.5$. Σε κάθε χρονικό βήμα, ο πράκτορας λαμβάνει ανταμοιβή -1. Όσο ταχύτερη είναι η έξοδος, τόσο μεγαλύτερη θα είναι και η καταλληλότητα της λύσης.

Εκτός από τις δύο εκδόσεις που προαναφέρθηκαν και οι οποίες μπορούν να θεωρηθούν ως οι Μαρκοβιανές εκδοχές (M), εξετάστηκαν ακόμα δύο εκδόσεις του περιβάλλοντος του αυτοκινήτου πλαγιάς, η μη-Μαρκοβιανή (NM) δισδιάστατη και τρισδιάστατη έκδοση. Στις εκδόσεις αυτές η ταχύτητα αποκρύπτεται από το διάνυσμα καταστάσεων που δίνεται στον πράκτορα και παραμένει μόνο η θέση του στο χώρο. Σε όλες τις περιπτώσεις κάθε επεισόδιο έχει διάρκεια $2,500$ βήματα.

Οι μέθοδοι NEAR και NEAT εξελίχθηκαν για 100 γενιές. Ο πληθυσμός τους αποτελούνταν από 100 γονιδιώματα. Κάθε γονίδιωμα αξιολογήθηκε για 100 επεισόδια με τυχαίες επανεκκινήσεις, όπως προαναφέρθηκε. Σε κάθε γενιά το γονιδίωμα πρωταθλητής επαναξιολογήθηκε για $1,000$ επεισόδια χωρίς την ενεργοποίηση των λειτουργιών της μάθησης και της εξερεύνησης για σκοπούς επαλήθευσης της γνώσης που αποκόμισε. Με βάση αυτή τη μετρική, ο πρωταθλητής των πρωταθλητών επαναξιολογήθηκε για ακόμα $1,000$ επεισόδια, ώστε να δοθεί η τελική γενικευμένη επίδοση του γονιδιώματος και του αλγορίθμου γενικότερα. Οι παράμετροι



Σχήμα 4.3: Η τοπογραφία του τρισδιάστατου προβλήματος mountain car.



Σχήμα 4.4: Σχηματική αναπαράσταση ενός ΔΗΚ για το πρόβλημα 3-D mountain car.

μάθησης για το πρόβλημα ορίστηκαν ως $\gamma = 1$, $\lambda = 0$, $\alpha = 10^{-5}$ και $\epsilon = 0.01$ στις περιπτώσεις όπου εφαρμόστηκε η μάθηση. Συνοπτικά τα παραπάνω βήματα παρουσιάζονται στον Αλγόριθμο 4.1.

Αλγόριθμος 4.1 Πειραματική αξιολόγηση συμπεριφοράς μεθόδων νευροεξέλιξης.

TEST

Αρχικοποίηση πληθυσμού με 100 γονιδιώματα
for g $\in 1:100$ γενιές **do**
  for i $\in 1:100$ γονιδιώματα **do**
    for e $\in 1:100$ επεισόδια **do**
      Τυχαία επανεκκίνηση επεισοδίου
      Αξιολόγηση γονιδιώματος με μάθηση
      fitness[i] ← fitness[i] + fitness[i,e]
    end for
    fitness[i] ← fitness[i] / 100
  end for
  generation-champion ← arg max(fitness)
  CP[g] ← Αξιολόγηση generation-champion για 1000 επεισόδια χωρίς μάθηση και τυχαία επανεκκίνηση
  Εξέλιξη πληθυσμού
end for
champion-of-champions ← arg max(championFitness)
GP ← Αξιολόγηση champion-of-champions για 1000 επεισόδια χωρίς μάθηση και τυχαία επανεκκίνηση

Οι Πίνακες 4.1, 4.2, 4.3 και 4.4 παρουσιάζουν τους μέσους όρους και τις τυπικές αποκλίσεις των επιδόσεων του πρωταθλητή των πρωταθλητών (CP) μαζί με την τελική επίδοση γενίκευσης (GP) για τις τέσσερις διαφορετικές εκδοχές του αυτοκινήτου πλαγιάς. Τα αποτελέσματα έχουν υπολογιστεί ως μέσοι όροι από 30 εκτελέσεις των υπό εξέταση αλγορίθμων. Για την περίπτωση των ESN-TD, αρχικοποιήθηκαν 100 δίκτυα χρησιμοποιώντας τις μέσες τιμές των τριών χαρακτηριστικών, N, D και ρ, όπως αυτά παρατίθενται στον Πίνακα 4.5, ενώ η μάθηση ήταν ενεργή για $1,000$ επεισόδια.

Από τους παραπάνω πίνακες είναι φανερό ότι η μέθοδος NEAR έχει δυνατότητες εύρεσης αποδοτικών δικτύων, είτε η κατάσταση των σημάτων περιλαμβάνει πληροφορίες ταχύτητας είτε όχι. Η NEAR+PS και στη συνέχεια η NEAR+TD-L δημιούργησαν τα δίκτυα με τις υψηλότερες επιδόσεις στην κλασική αλλά και στην περισσότερο μελετημένη περίπτωση 2D-MC-M, όπου πέτυχαν ίση συμπεριφορά γενίκευσης με επιδόσεις ισάξιες με αλγορίθμους αναφοράς όπως η NEAT+Q και ο SARSA με κωδικοποίηση πλακιδίων (-48) όπως αναφέρεται στο [WS06]. Από την άλλη μεριά, προκύπτει ότι τα ΔΗΚ, αν και έχουν τον ίδιο αριθμό νευρώνων, δεν μπορούν να καταστούν αρκετά ανταγωνιστικά χωρίς κάποιου είδους βελτιστοποίηση στη δεξαμενή νευρώνων. Για την ακρίβεια, η μέθοδος ESN+TD απέτυχε σε ορισμένες περιπτώσεις να βρει μια ικανοποιητική λύση στο πρόβλημα 3D-MC και

Πίνακας 4.1: Μέσος όρος (μ) και τυπική απόκλιση (σ) των επιδόσεων του πρωταθλητή των πρωταθλητών για το δισδιάστατο Μαρκοβιανό αυτοκίνητο πλαγιάς.

Algorithm	2D-MC-M			
	CP-μ	CP-σ	GP-μ	GP-σ
NEAT	-49.5	(0.5)	-51.9	(2.2)
NEAR+TD-L	-48.9	(0.4)	**-51.3**	(1.4)
NEAR+TD-D	-49.7	(0.8)	-51.5	(1.18)
NEAR+PS	**-48.7**	(0.4)	-51.4	(1.8)
ESN+TD	-59.6	(11.5)	-59.2	(10.6)

Πίνακας 4.2: Μέσος όρος (μ) και τυπική απόκλιση (σ) των επιδόσεων του πρωταθλητή των πρωταθλητών για το δισδιάστατο μη-Μαρκοβιανό αυτοκίνητο πλαγιάς.

Algorithm	2D-MC-NM			
	CP-μ	CP-σ	GP-μ	GP-σ
NEAT	-154.1	(63.7)	-173.0	(75.6)
NEAR+TD-L	**-58.8**	(2.6)	**-62.4**	(6.8)
NEAR+TD-D	-60.0	(2.9)	-64.3	(5.6)
NEAR+PS	-62.0	(2.2)	-67.9	(6.3)
ESN+TD	-77.8	(8.7)	-77.7	(8.5)

για το λόγο αυτό προέκυψαν και οι ιδιαίτερα υψηλές τιμές στις τυπικές αποκλίσεις της μεθόδου.

Ο Πίνακας 4.5 περιέχει στατιστικά υπολογισμένα για τα μακρο-χαρακτηριστικά δεξαμενών ΔΗΚ με τις υψηλότερες επιδόσεις, όπως βρέθηκαν μέσω της μεθόδου NEAR+PS. Τα στατιστικά φανερώνουν ότι όταν ο πράκτορας δε λαμβάνει πλέον όλες τις δυνατές πληροφορίες της καταστασής του (για παράδειγμα σε σχέση με τα προβλήματα 2D-MC-M και 2D-MC-NM), απαιτούνται περισσότερα χαρακτηριστικά δεξαμενής προκειμένου να καταστεί ο πράκτορας αποδοτικός (χαρακτηριστικό N). Η πυκνότητα και η φασματική ακτίνα είναι περίπου στο μεσαίο μέρος των πεδίων ορισμού, δηλαδή στο $[0.01, 0.99]$ για το D και $[0.55, 0.99]$ για το ρ.

Στη συνέχεια, η καλύτερη μέθοδος NEAR+PS για τη συγκεκριμένη σειρά πειραμάτων εξετάσθηκε απέναντι στον αλγόριθμο με την επέκταση συναρτήσεων βάσης Fourier [KOT11] στο πρόβλημα 2D-MC-M σε δύο διαφορετικές παραμετροποιήσεις. Στην πρώτη περίπτωση κάθε επεισόδιο ξεκινούσε με τυχαίο τρόπο, όπως προηγουμένως. Η επέκταση με αποδόμηση Fourier, αφού αφέθηκε να μάθει για $1,000$ επεισόδια, είχε μέση επίδοση σε 30 εκτελέσεις 56.4 με 2.01 τυπική απόκλιση. Στη δεύτερη παραμετροποίηση, η αρχική θέση του οχήματος ήταν στο μέσο της κοιλάδας με μηδενική ταχύτητα. Η επέκταση με αποδόμηση Fourier βρίσκει καλές λύσεις μετά από 4 επεισόδια. Βέβαια η NEAR+PS ανακαλύπτει καλές λύσεις κατά την εξέταση της πρώτης κιόλας γενιάς. Μετά από $1,000$ επεισόδια με την μάθηση ενεργοποιημένη, η μεθοδολογία βασισμένη στο μετασχηματισμό Fourier,

Πίνακας 4.3: Μέσος όρος (μ) και τυπική απόκλιση (σ) των επιδόσεων του πρωταθλητή των πρωταθλητών για το τρισδιάστατο Μαρκοβιανό αυτοκίνητο πλαγιάς.

Algorithm	3D-MC-M			
	CP-μ	CP-σ	GP-μ	GP-σ
NEAT	-113.8	(10.0)	-124.2	(15.4)
NEAR+TD-L	-93.0	(2.2)	-95.1	(3.8)
NEAR+TD-D	-116.5	(7.8)	-121.3	(8.7)
NEAR+PS	**-90.6**	(3.3)	**-93.7**	(3.9)
ESN+TD	-371.0	(466.8)	-375.3	(473.7)

Πίνακας 4.4: Μέσος όρος (μ) και τυπική απόκλιση (σ) των επιδόσεων του πρωταθλητή των πρωταθλητών για το τρισδιάστατο μη-Μαρκοβιανό αυτοκίνητο πλαγιάς.

Algorithm	3D-MC-NM			
	CP-μ	CP-σ	GP-μ	GP-σ
NEAT	-333	(39)	-368	(52)
NEAR+TD-L	-179	(20)	-180	(24)
NEAR+TD-D	-191	(16)	-196	(16)
NEAR+PS	**-172**	(25)	**-178**	(23)
ESN+TD	-568	(274)	-573	(273)

φτάνει στο στόχο ύστερα από 114 βήματα κατά μέσο όρο σε 30 εκτελέσεις, ενώ η NEAR+PS βρίσκει ένα μονοπάτι διαφυγής με 123 βήματα στην πρώτη γενιά και με 104 βήματα, εφόσον ο αλγόριθμος αφεθεί να βελτιστοποιήσει επιπλέον τα δίκτυα για 50 γενιές.

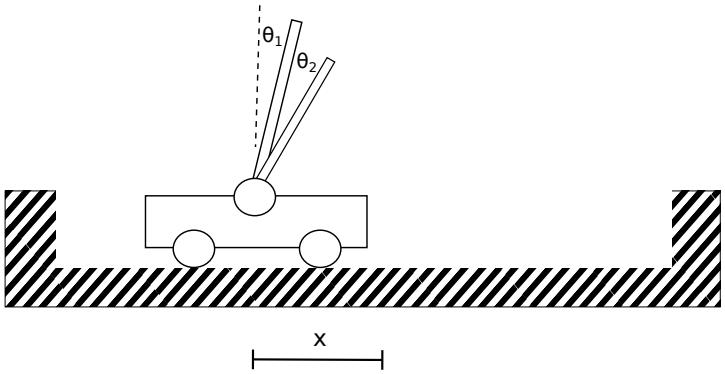
4.2 Ισορρόπηση ράβδου

Το κλασικό πρόβλημα ισορρόπησης ράβδου αφορά έναν ελεγκτή που προσπαθεί να ισορροπήσει μία ράβδο τοποθετημένη πάνω σε ένα βαγονάκι. Εφαρμόζοντας δύναμη στο βαγονάκι, αυτό μπορεί να κινηθεί μπρος-πίσω, πάνω σε μία πεπερασμένη ράγα. Για την αξιολόγηση χρησιμοποιήθηκαν 5 διαφορετικές εκδόσεις του προβλήματος που βασίζονται στο [Wie91]. Οι εκδόσεις αυτές έχουν χρησιμοποιηθεί και στη βιβλιογραφία για την εξέταση αλγορίθμων ενισχυτικής μάθησης [GSM06, GSM08] και είναι οι εξής:

1. μία ράβδος και πλήρες διάνυσμα καταστάσεων (single pole balancing markov - SPB-M),
2. μία ράβδος και μη πλήρες διάνυσμα καταστάσεων (single pole balancing non makrov - SPB-NM),
3. δύο ράβδους και πλήρες διάνυσμα καταστάσεων (double pole balancing markov - DPB-M),

Πίνακας 4.5: Μέσος όρος και τυπική απόκλιση (σε παρένθεση) για τις ιδιότητες της δεξαμενής των ικανότερων ΔΗΚ που προέκυψαν από τη μέθοδο NEAR+PS.

Parameter	2D-MC-M	2D-MC-NM	3D-MC-M	3D-MC-NM
N	4.50 (2.78)	6.13 (4.66)	3.80 (1.71)	4.23 (2.48)
D	0.42 (0.34)	0.44 (0.32)	0.54 (0.34)	0.43 (0.31)
ρ	0.73 (0.15)	0.80 (0.14)	0.81 (0.14)	0.78 (0.14)



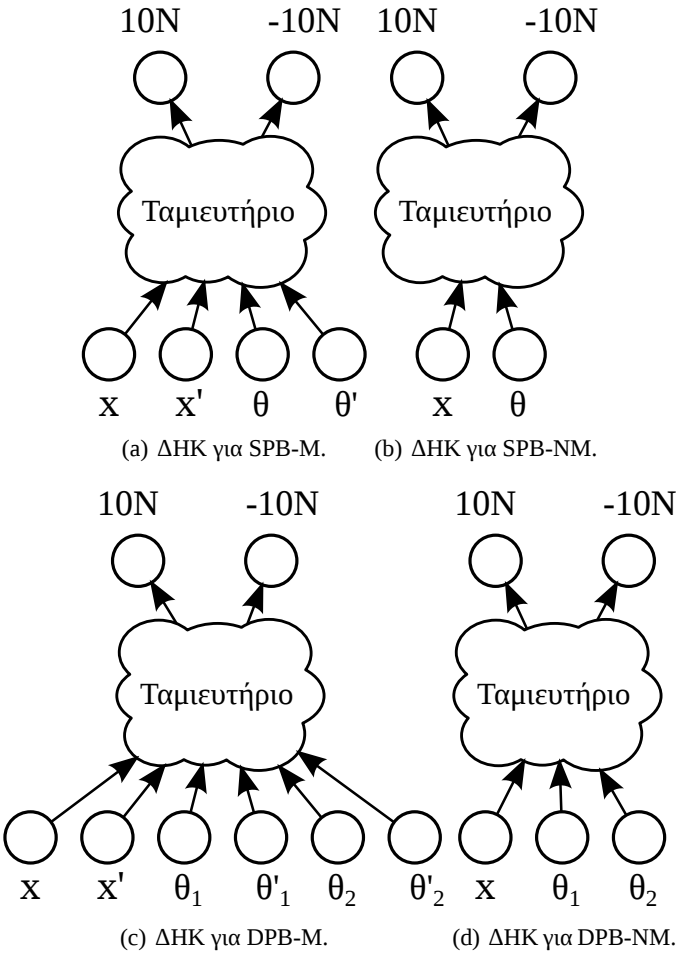
Σχήμα 4.5: Το πρόβλημα ισορρόπησης δυο ράβδων.

4. δύο ράβδους και μη πλήρες διάνυσμα καταστάσεων (double pole balancing non markov - DPB-NM) και

5. δύο ραβδους, μη πλήρες διάνυσμα καταστάσεων και αποσβετική συνάρτηση καταλληλότητας (dumping fitness), η οποία αποτρέπει τους ελεγκτές να βρίσκουν πολιτικές στήριξης κουνώντας τη ράβδο πέρα δώθε (double pole balancing non markov dumping fitness or DPB-NM-D). Η συνάρτηση χρησιμοποιήθηκε αρχικά στο [GWP96] και στη συνέχεια σε πολλές άλλες εργασίες [Sta04, GSM06, GSM08, JBS08a].

Σε κάθε χρονικό βήμα ο πράκτορας πρέπει να επιλέξει ανάμεσα σε δύο ενέργειες, εφαρμογή δύναμης $F_x = -10N$ ή $F_x = 10N$ στο βαγονάκι. Οι μεταβλητές κατάστασης είναι η θέση x, η ταχύτητα του καροτσιού $\dot{x}$, η γωνία θ_1 και η γωνιακή ταχύτητα $\dot{\theta}_1$ της μίας ή και των δύο ράβδων ανάλογα την περίπτωση (Σχήμα 4.5). Όλες οι ταχύτητες αποκρύπτονται από τον πράκτορα όταν υπάρχει μη πλήρες διάνυσμα καταστάσεων. Όσο περισσότερα βήματα ισορροπεί η ράβδος, τόσο μεγαλύτερη και η καταλληλότητα. Το πρόβλημα θεωρούμε ότι έχει λυθεί εάν ο πράκτορας ισορροπεί τη μία ή τις δύο ράβδους για $100,000$ βήματα. Το πρόβλημα αποτυγχάνει εάν η γωνία υπερβεί τις 12° ή τις 36° μοίρες για τη μία ή τις δύο ράβδους αντίστοιχα. Αναπαραστάσεις ΔΗΚ για τα προβλήματα ισορρόπησης ράβδων παρουσιάζονται στο Σχήμα 4.6.

Για το περιβάλλον της ισορρόπησης ράβδου επιλέξαμε την παραμετροποίηση στο [GSM06]. Στην ίδια αναφορά γίνεται μια εκτενής αξιολόγηση πολλών αλγορίθ-



(a) ΔΗΚ για SPB-M. (b) ΔΗΚ για SPB-NM.

(c) ΔΗΚ για DPB-M. (d) ΔΗΚ για DPB-NM.

Σχήμα 4.6: Αναπαράσταστη ΔΗΚ για τα προβλήματα ισορρόπησης ράβδου.

μων και στις πέντε διαφορετικές περιπτώσεις που προαναφέρθηκαν και οι οποίες αποτυπώνονται στο παρόν κείμενο για λόγους σύγκρισης. Δεδομένου ότι οι υψηλότερα αξιολογημένες μέθοδοι δεν είχαν ενεργοποιημένη τη μάθηση, εξετάσαμε τη μέθοδο NEAR χωρίς το μαθησιακό κομμάτι. Η μετρική σύγκρισης ήταν ο αριθμός των δικτύων που αξιολογήθηκαν πριν αναγνωρισθεί μία λύση. Εάν υπήρχε ενεργοποιημένη η μάθηση, κάθε δίκτυο θα χρειαζόταν να ελεγχθεί περισσότερες από μία φορές και έτσι τα αποτελέσματα δε θα ήταν συγκρίσιμα. Οι αξιολογήσεις αποτελούν μέσο όρο 50 εκτελέσεων. Η NEAR κατάφερε να βρει λύση και στις 50 εκτελέσεις σε όλες τις περιπτώσεις. Τα αποτελέσματα συγκεντρώνονται στον Πίνακα 4.6.

Πίνακας 4.6: Μέση τιμή (μ) και τυπική απόκλιση (σ) των αξιολογήσεων που χρειάστηκαν σε 50 εκτελέσεις για τα προβλήματα ισορρόπησης ράβδου Τα αποτελέσματα με αστερίσκο είναι από το [GSM06].

Πρόβλημα	SPB-M	SPB-NM	DPB-M	DPB-NM	DPB-NM-D
μ	**25**	**1948**	**35**	**420**	**680**
σ	25	1300	29	254	303
. Solutions Found	50	50	50	50	50
NEAT-μ^*	743	1523	3600	6929	24543
Best-μ^*	98	127	895	1249	3416
NEAR Ranking	1	6	1	1	1

Σε τέσσερις από τις πέντε εκδόσεις, η NEAR παρουσίασε την καλύτερη απόδοση. Το πρόβλημα SPB-NM παρουσίασε για τη NEAR τη μεγαλύτερη δυσκολία. Αυτό φαίνεται από τα αποτελέσματα του πίνακα 4.7 όπου παρατηρείται ότι στο συγκεκριμένο πρόβλημα χρειάστηκε μεγαλύτερος αριθμός νευρώνων για να βρεθεί μια λύση κατά μέσο όρο. Μια εξήγηση θα μπορούσε να είναι ότι μόνο δύο μεταβλητές κατάστασης οδηγούν το ΔΗΚ, με αποτέλεσμα να χρειάζονται περισσότερα μη γραμμικά χαρακτηριστικά για την ανάπτυξη της τελικής πολιτικής. Μια άλλη παρατήρηση είναι ότι στο DPB-NM πρόβλημα, η φασματική ακτίνα ρ είναι υψηλότερη από άλλες περιπτώσεις, συμπεραίνοντας έτσι ότι τα σήματα κατάστασης χρειάζεται να παραμείνουν για μεγαλύτερη περίοδο στο ταμιευτήριο, ώστε να ολοκληρωθούν οι κατάλληλοι χρονικοί υπολογισμοί.

Πίνακας 4.7: Στατιστικά των χαρακτηριστικών της δεξαμενής για τα δίκτυα με τις υψηλότερες επιδόσεις στο περιβάλλον ισορρόπησης ράβδου.

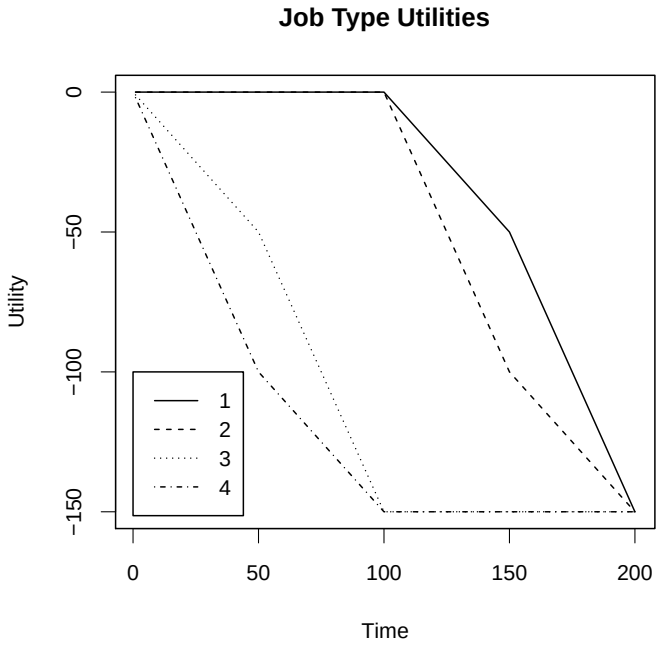
Παράμετρος	SPB	SPB-NM	DPB	DPB-NM	DPB-NM-D
N	1.00 (0.00)	4.08 (1.57)	1.00 (0.00)	2.22 (0.95)	2.18 (1.08)
D	0.46 (0.26)	0.47 (0.30)	0.55 (0.27)	0.52 (0.26)	0.65 (0.27)
ρ	0.77 (0.12)	0.77 (0.13)	0.72 (0.10)	0.80 (0.14)	0.76 (0.14)

4.3 Χρονοπρογραμματισμός διαδικασιών σε εξυπηρετητή

Ο χρονοπρογραμματισμός διαδικασιών σε εξυπηρετητή (server job scheduling) [WS06] είναι ένα πρόβλημα από την περιοχή της αυτόνομης υπολογιστικής (autonomic computing) [KC03]. Τέσσερις τύποι διεργασίων περιμένουν στην ουρά ενός εξυπηρετητή προκειμένου να εκτελεστούν, ενώ νέες έρχονται. Κάθε τύπος διεργασίας έχει μία συνάρτηση ωφέλειας, η οποία μεταβάλλεται με το χρόνο και καθορίζει την ανυπομονησία του χρήστη (Σχήμα 4.7). Ορισμένοι χρήστες θέλουν γρήγορη απόκριση από τον εξυπηρετητή ενώ άλλοι μπορούν και περιμένουν. Στόχος του χρονοπρογραμματιστή είναι να επιλέξει μια διεργασία (ενέργεια) από εκείνες που περιμένουν στην ουρά (κατάσταση), λαμβάνοντας ως ανταμοιβή την τιμή της συνάρτησης ωφέλειας της επιλεγμένης εργασίας στο χρονικό διάστημα από τη στιγμή που μετέβηκε στην ουρά (άμεση ανταμοιβή - immediate reward). Η επίδοση υπολογίζεται με βάση το άθροισμα των ανταμοιβών (μακροπρόθεσμο κέρδος - long term reward) όταν η ουρά αδειάσει. Κάθε επεισόδιο ξεκινά με την παρουσία 100 διεργασιών στην ουρά, ενώ σε κάθε βήμα μια νέα διεργασία εισέρχεται στην ουρά για τα πρώτα 100 βήματα και μια άλλη εκτελείται και διαγράφεται από την ουρά. Κάθε επεισόδιο διαρκεί 200 βήματα. Η κωδικοποίηση των καταστάσεων και των ενεργειών ακολουθεί αυτή του [WS06]. Το διάνυσμα κατάστασης αποτελείται από 16 μεταβλητές που εμφανίζουν τον αριθμό των διεργασιών που βρίσκονται στην ουρά για κάθε τύπο διεργασίας και κάθε τμήμα της συνάρτησης ωφέλειας. Επίσης, υπάρχουν 16 ενέργειες, όπου κάθε μία αναπαριστά την επιλογή μίας διεργασίας (τυχαίας) που κωδικοποιείται από μία από τις 16 καταστάσεις. Ένα τυπικό ΔΗΚ για το παρόν πρόβλημα παρουσιάζεται στο Σχήμα 4.8, όπου s_{ij} οι διαδικασίες τύπου i που βρίσκονται στη χρονική κατάσταση $50 \cdot (j-1) \leq t < 50 \cdot j$ και a_{ij} η επιλογή μίας διαδικασίας τύπου i που βρίσκεται στην χρονική κατάσταση $50 \cdot (j-1) \leq t < 50 \cdot j$.

Όπως και στην περίπτωση του οχήματος πλαγιάς, ακολουθήθηκε η παραμετροποίηση στο [WS06]. Αρχικοποιήθηκε ένας πληθυσμός 100 γονιδιωμάτων που εξελίχθηκε για 100 γενιές και κάθε οργανισμός είχε τη δυνατότητα να μάθει για 100 τυχαίως αρχικοποιημένα επεισόδια. Για τη διαδικασία της μάθησης χρησιμοποιήθηκαν οι παράμετροι μάθησης, όπως και στο αυτοκίνητο πλαγιάς. Οι επιδόσεις των αλγορίθμων, και συγκεκριμένα του πρωταθλητή των πρωταθλητών, αναφέρονται σε 1000 νέα, τυχαία αρχικοποιημένα επεισόδια. Τα αποτελέσματα που παρουσιάζονται στον Πίνακα 4.8 είναι οι μέσοι όροι από 30 εκτελέσεις.

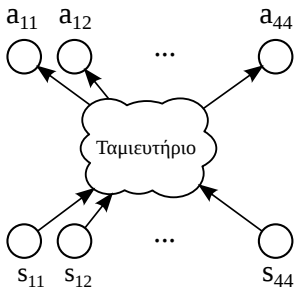
Παρατηρούμε και σε αυτή την περίπτωση ότι η NEAR+PS επιβλήθηκε απέναντι στους άλλους αλγορίθμους. Τα χαρακτηριστικά των ΔΗΚ με τις υψηλότερες επιδόσεις ήταν κατά μέσο όρο στις 30 εκτελέσεις (τυπική απόκλιση σε παρένθεση): $D = 0.37(0.29)$, $\rho = 0.80(0.14)$ και $N = 3.29(1.98)$. Μία ακόμα παρατήρηση είναι ότι η Λαμαρκιανή εξέλιξη για μία ακόμα φορά ξεπέρασε τις επιδόσεις της Δαρβίνιας, όπως και στο αυτοκίνητο πλαγιάς, ακολουθώντας επίσης τα συμπεράσματα του [WS06].



Σχήμα 4.7: Οι συναρτήσεις ωφέλειας για κάθε έναν από τους τέσσερις τύπους διεργασιών καθώς ο χρόνος αυξάνεται.

4.4 Τμηματική αξιολόγηση

Επόμενος στόχος της πειραματικής ανάλυσης και αξιολόγησης της NEAR ήταν η στοχευμένη αξιολόγηση τμημάτων του αλγορίθμου, όπως η διασταύρωση και η ομαδοποίηση σε είδη, καθώς και ο έλεγχος εάν τα συγκεκριμένα τμήματα του αλγορίθμου ευνοούν ή όχι την επίδοση της NEAR. Μελέτες αποκόλλησης (ablation studies) έγιναν επίσης και για τη μέθοδο NEAT στο [SM02] με θετικά αποτελέσματα τόσο για την ομαδοποίηση όσο και για τη διασταύρωση με ιστορική δεικτοδότηση, οπότε προέκυψε η ίδια ανάγκη προκειμένου να διερευνηθεί εάν οι αλλαγμένες μοντελοποιήσεις της διασταύρωσης και της ομαδοποίησης στη NEAR παραμένουν θετικές. Η ανάλυση πραγματοποιήθηκε στα πιο δύσκολα περιβάλλοντα, χρησιμοποιώντας την καλύτερη εναλλακτική NEAR+PS. Στον πίνακα 4.9 περιγράφονται τα αποτελέσματα της πλήρους μεθόδου NEAR+PS, καθώς και αυτά με την απουσία της ομαδοποίησης και την απουσία της διασταύρωσης. Σύμφωνα με τα αποτελέσματα και ακολουθώντας και τα συμπεράσματα του [SM02], προκύπτει ότι και τα δύο αυτά τμήματα της NEAR οδηγούν τον αλγόριθμο στη δημιουργία καλύτερων δικτύων μέσω της νευροεξέλιξης. Τα αποτελέσματα επιβεβαιώνουν ότι τόσο η διασταύρωση όσο και η ομαδοποίηση είναι σημαντικοί παράγοντες για την επίδοση της NEAR.



Σχήμα 4.8: ΔΗΚ για το πρόβλημα χρονοπρογραμματισμού εργασιών.

Πίνακας 4.8: Μέσος όρος (μ) και τυπική απόκλιση (σ) των επιδόσεων του πρωταθλητή των πρωταθλητών για το πρόβλημα του χρονοπρογραμματισμού διεργασιών σε εξυπηρετητή.

Αλγόριθμος	SJS			
	CP-μ	CP-σ	GP-μ	GP-σ
NEAT	$-5,675.18$	(206.01)	$-5,726.88$	(213.75)
NEAR+TD-L	$-7,856.34$	(231.46)	$-7,856.46$	(240.00)
NEAR+TD-D	$-9,220.41$	(608.20)	$-9,219.92$	(590.12)
NEAR+PS	**-5,154.69**	(23.28)	**-5,189.92**	(25.15)
ESN+TD	$-12,160.58$	(107.47)	$-12,164.19$	(112.23)

4.5 Σύνοψη και συζήτηση

Στο κεφάλαιο αυτό έγινε σύγκριση και αξιολόγηση διαφόρων μεθοδολογιών σε προβλήματα ενισχυτικής μάθησης. Διαπιστώθηκε ότι η μέθοδος NEAR είναι αποτελεσματική στην ανακάλυψη ΔΗΚ υψηλών επιδόσεων μέσω της μάθησης και της εξέλιξης.

Με βάση τις οδηγίες στο [Dem06] εφαρμόστηκε το τεστ Friedman [Fri37, Fri40] προκειμένου να εξεταστεί η μηδενική υπόθεση (τόσο η NEAR όσο και η NEAT έχουν κατά μέσο όρο την ίδια επίδοση). Η μέση κατάταξη στα 10 προβλήματα για τις NEAR και NEAT είναι 1.1 και 1.9 αντίστοιχα, δίνοντας τιμή για το χ_F^2, 6.4, ικανή για να καταρριφθεί η μηδενική υπόθεση με $\alpha = 0.025$. Επίσης, η εφαρμογή του τεστ Friedman με περισσότερους αλγόριθμους στα προβλήματα οχήματος πλαγιάς και χρονοπρογραμματισμού του εξυπηρετητή και πάλι απέρριψε τη μηδενική υπόθεση για την κατά μέσο όρο ίδια επίδοση των πέντε αλγορίθμων με $\chi_F^2 = 15.31$. Η επικράτηση της NEAR έναντι της NEAT μπορεί να αποδοθεί σε δύο βασικά αίτια, την απουσία αυστηρά γραμμικών χαρακτηριστικών που φαίνεται ότι στο παρελθόν έχουν βοηθήσει άλλες μεθόδους προσεγγιστικών συναρτήσεων [FP08], καθώς και τη δημιουργία ad-hoc δικτύων αντί για πιο θεωρητικά και πειραματικά τεκμηριωμένες προσεγγίσεις όπως αυτές των ΔΗΚ. Από την άλλη, οι εξελικτικοί μηχανισμοί της μεθοδολογίας NEAT, βοήθησαν σημαντικά τη NEAR και θα μπορούσαν γενικότερα να ενισχύσουν και άλλους αλγορίθμους νευροεξέλι-

Πίνακας 4.9: Μελέτη αποκόλλησης για τη μέθοδο NEAR. Η επίδοση αναφέρεται στη μετρική του εκάστοτε προβλήματος και αποτελεί μέσο όρο 30 εκτελέσεων για το αυτοκίνητο πλαγιάς και 50 εκτελέσεων για την ισορρόπηση ράβδου.

Μέθοδος	3DMC-NM	DPB-NM
NEAR+PS	-178.27	419.98
Χωρίς διασταύρωση	-233.04	538.54
Χωρίς ομαδοποίηση	-190.05	626.60

ξης.

Ακόμα, η παραπάνω σειρά πειραμάτων μας επιβεβαίωσε τα συμπεράσματα που μπορεί να βρει κάποιος σε παρόμοιες ερευνητικές εργασίες:

- Η Λαμαρκιανή εξέλιξη υπερισχύει έναντι της Δαρβινικής εξέλιξης, τουλάχιστον στα προβλήματα και στις διευθετήσεις που εξετάστηκαν παραπάνω [WS06].
- Οι ταμιευτήρες που αναπαρίστανται από πίνακες γειτνίασης δεν είναι αναγκαίο να είναι αραιοί, πολύ δε περισσότερο όταν υφίστανται και μια διαδικασία βελτιστοποίησης [RI09].
- Η αναζήτηση πολιτικών επικρατεί έναντι της μεθόδου χρονικών διαφορών όταν δεν υπάρχει θόρυβος στη συνάρτηση καταλληλότητας [WTS09].
- Η διασταύρωση μπορεί να δώσει ώθηση σε μεθόδους νευροεξέλιξης όταν πραγματοποιείται με τον κατάλληλο τρόπο [Sta04, HN10], για παράδειγμα μέσω ιστορικής δεικτοδότησης. Στο ίδιο συμπέρασμα θα μπορούσε κανείς να καταλήξει και για την ομαδοποίηση.

Οι τιμές των D και ρ των καλύτερων δικτύων κυμαίνονται στη μέση των πεδίων ορισμού των χαρακτηριστικών αυτών, σε αντίθεση με τον αριθμό νευρώνων N που είναι γενικά μικρός. Τιμές πυκνότητας κοντά στο 45% και φασματικής ακτίνας κοντά στο 0.75 μπορούν να θεωρηθούν καλές αρχικές προσεγγίσεις. Αν και τα ευρήματα αυτά αντιτίθενται στις καλές πρακτικές δημιουργίας ΔΗΚ (Κεφάλαιο 2), αυτό πιθανότατα αντισταθμίζεται από την εξέλιξη της τοπολογίας και των βαρών και τη σμίλευση της δεξαμενής ανάλογα με τις τρέχουσες τιμές D, ρ και N.

Τέλος, οι τυπικές αποκλίσεις της επίδοσης των κλασικών, τυχαία δημιουργημένων ΔΗΚ, ήταν μεγάλες λόγω της μεγάλης διαφοράς στις επιδόσεις μεταξύ των ΔΗΚ, ανάλογα με την αρχική τους τυχαία κατασκευή. Ωστόσο, η NEAR παρατηρείται ότι ασκεί πίεση για τη δημιουργία ΔΗΚ, τα οποία λύνουν το πρόβλημα αποδοτικά με μικρές αποκλίσεις στη συμπεριφορά τους. Έτσι, τα νευροεξελικτικά τμήματα της NEAT σε συνδυασμό με τη δομή των ΔΗΚ προκύπτει ότι αποτελούν εφικτές λύσεις στη δημιουργία μοντέλων αποφάσεων σε εργασίες όπου χρειάζονται ακόμα και χρονικοί υπολογισμοί.

Κεφάλαιο 5

Μεταφορά μάθησης

Παρά τα πλεονεκτήματα της ενισχυτικής μάθησης, συχνά είναι απαραίτητο να δαπανηθεί σημαντικός χρόνος για την εκμάθηση, ειδικά σε σύνθετα προβλήματα. Για την επιτάχυνση της διαδικασίας, η λύση που προτείνεται είναι η *μεταφορά μάθησης* (transfer learning) κατά την οποία, η γνώση που αποκτήθηκε σε ένα προηγούμενο πρόβλημα μάθησης, το *πηγαίο πρόβλημα* ή *πηγαία εργασία*, χρησιμοποιείται ώστε να ενισχυθεί η μαθησιακή διαδικασία σε ένα νέο, συνήθως πιο σύνθετο πρόβλημα, το *πρόβλημα στόχος* ή *εργασία στόχος*. Τα προβλήματα μπορεί να ανήκουν στον ίδιο χώρο, για παράδειγμα δύο λαβύρινθοι με διαφορετική δομή ή σε διαφορετικές περιοχές όπως, για παράδειγμα, τα παιχνίδια ντάμα και σκάκι. Όσο μεγαλύτερη η ομοιότητα ανάμεσα σε δύο προβλήματα, τόσο ευκολότερη είναι η μεταφορά μάθησης ανάμεσά τους. Η μεταφορά μάθησης παίζει σημαντικό ρόλο στη δημιουργία πλήρως αυτόνομων πρακτόρων, αφού πιστεύεται ότι είναι ένα βασικό συστατικό για πράκτορες με μηχανισμούς δια βίου μάθησης που διατηρούνται στο χρόνο [TS09].

Στο παρόν κεφάλαιο εξετάζεται μια επέκταση της μεθόδου NEAT με δυνατότητες μεταφοράς μάθησης και, πιο συγκεκριμένα, μελετάται η δυνατότητα μεταφοράς τοπολογιών ταμιευτηρίων, δηλαδή χαρακτηριστικών ταμιευτηρίων (reservoir features). Τοπολογίες προσαρμοσμένες στο πηγαίο πρόβλημα ενισχυτικής μάθησης μπορούν να χρησιμοποιηθούν ως πρότυπα για τον αρχικό πληθυσμό της νευροεξέλιξης στο πρόβλημα στόχος ενισχυτικής μάθησης. Εκτός από τις δοκιμές με τη μεταφορά των δεξαμενών μέσω αντιστοιχίας μεταξύ των μεταβλητών κατάστασης και των ενεργειών ανάμεσα στο πηγαίο και το πρόβλημα στόχος, αξιολογήθηκε και μία μέθοδος η οποία δε χρειάζεται καμία αντιστοίχηση, και η οποία λαμβάνει υπόψιν της τη φύση των ΔΗΚ ως τυχαία και πλήρως συνδεδεμένα δίκτυα ανάμεσα στο ταμιευτήριο και τους λοιπούς πίνακες βαρών. Η διαδικασία αυτή εφαρμόζεται σε μικροσκοπικό επίπεδο, δηλαδή στο επίπεδο της τοπολογίας και των βαρών του ταμιευτηρίου όπως θα δούμε στο παρόν κεφάλαιο.

5.1 Σχετική βιβλιογραφία

Η ενότητα αυτή παρουσιάζει επιγραμματικά την απολύτως σχετική βιβλιογραφία στον τομέα της ενισχυτικής μάθησης και των νευρωνικών δικτύων και τη συγκρίνει με την παρούσα προσέγγιση. Για μια αναλυτικότερη ανασκόπηση της περιοχής της μεταφοράς μάθησης σε προβλήματα ενισχυτικής μάθησης ο αναγνώστης μπορεί να αναφερθεί στο [TS09].

Η μεταφορά μάθησης με νευρωνικά δίκτυα έχει μελετηθεί στο [TWS07] και στο [BM08]. Πιο συγκεκριμένα στο [TWS07] προτείνεται η μέθοδος μεταφοράς μάθησης μέσω δια-προβληματικών αντιστοιχίσεων σε εργασίες αναζήτησης πολιτικών ενισχυτικής μάθησης (Transfer via Inter-Task Mappings for Policy Search Reinforcement Learning - TVITM-PS), όπου τα βάρη στο πρόβλημα στόχος αρχικοποιούνται χρησιμοποιώντας βάρη που έχουν διαμορφωθεί στο πηγαίο πρόβλημα, χρησιμοποιώντας συναρτήσεις αντιστοίχησης. Πρόκειται λοιπόν για μια μέθοδο μεταφοράς μάθησης για αλγορίθμους αναζήτησης πολιτικών, όπου οι πολιτικές αναπαρίστανται από νευρωνικά δίκτυα. Η εσωτερική τοπολογία των πηγαίων δικτύων μαζί με τα βάρη αντιγράφεται στο πρόβλημα στόχος χρησιμοποιώντας μια αντιστοίχηση. Η μέθοδος αυτή αξιολογήθηκε για τα δίκτυα ΔΗΚ και ειδικότερα για τη μέθοδο NEAR. Η αξιολόγηση αποκτά περισσότερο ενδιαφέρον αν λάβει κανείς υπόψιν του ότι τα ΔΗΚ έχουν εκ φύσεως πιο πολλές συνδέσεις από ότι τα δίκτυα που εξάγονται από τη μέθοδο NEAT. Παράλληλα, όπως προαναφέρθηκε, εξετάστηκε και μια μέθοδος που προσπαθεί να εκμεταλλευτεί το πλεονέκτημα της τυχαίας και πλήρους συνδεσμολογίας των ΔΗΚ από και προς το ταμιευτήριο, ώστε να μη χρειάζεται η παρουσία των συναρτήσεων αντιστοίχησης.

Όσον αφορά στο [BM08], οι Bahceci και Miikkulainen εισήγαγαν μια μέθοδο μεταφοράς ευριστικών προτύπων για παιχνίδια, όπου ο πληθυσμός εξελιγμένων δικτύων στο πηγαίο πρόβλημα (που αντιπροσωπεύουν τα πρότυπα) εισάγεται ως αρχικός πληθυσμός στο πρόβλημα στόχος. Στην περίπτωσή μας δε θα ήταν εφαρμόσιμη μία τέτοια μέθοδος καθώς υπάρχουν διαφορετικά διανύσματα μεταβλητών κατάστασης και ενεργειών σε κάθε πρόβλημα που καθιστούν την περίπτωση πιο δύσκολη.

5.2 Μεταφορά τοπολογιών ταμιευτήρων

Η ιδέα πίσω από αυτή την επέκταση είναι ότι ορισμένα χαρακτηριστικά από τα ΔΗΚ υψηλών επιδόσεων στο πηγαίο πρόβλημα μπορούν να χρησιμοποιηθούν ως πρότυπα δικτύων για τον αρχικό πληθυσμό στο πρόβλημα στόχο. Αυτά τα χαρακτηριστικά των ΔΗΚ θα πρέπει να έχουν κάποια ιδιαίτερα γνωρίσματα που θα διαχωρίζουν ένα ΔΗΚ από ένα άλλο. Στην περίπτωσή μας τα χαρακτηριστικά που επιλέχθηκαν είναι:

- το ταμιευτήριο, που καθορίζεται από τον πίνακα W, και μαζί του ο αριθμός N των νευρώνων στο ταμιευτήριο, ο γράφος της τοπολογίας και τα βάρη της συνδεσμολογίας.

- η πυκνότητα D του ταμιευτηρίου, και
- η φασματική ακτίνα ρ.

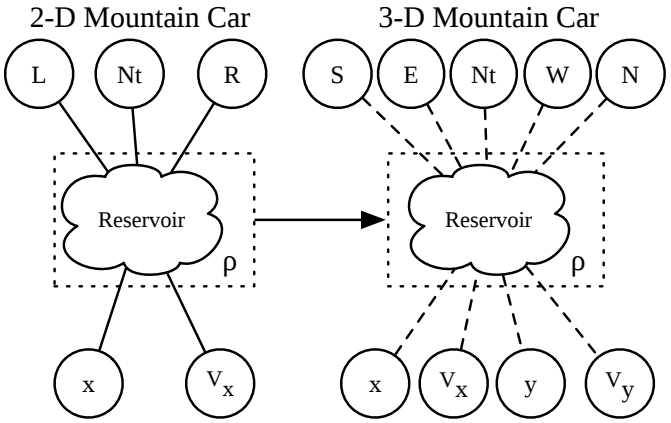
Τελικός στόχος είναι να απαλειφθεί εντελώς η διαδικασία ορισμού της συνάρτησης αντιστοίχησης για τις μεταβλητές κατάστασης και τις ενέργειες ανάμεσα στα προβλήματα, μεταφέροντας τον πίνακα γειτνίασης και τα χαρακτηριστικά του και αφήνοντας τους πίνακες W^{in} και W^{out} να προσαρμοστούν μέσω της NEAR. Κατά τον κλασικό, τυχαίο τρόπο δημιουργίας των ΔΗΚ, αλλά και στη μέθοδο NEAR, τόσο ο πίνακας W^{in} όσο και ο πίνακας W^{out} αρχικοποιούνται τυχαία αφήνοντας στη συνέχεια είτε την εξέλιξη είτε τη μάθηση να προσαρμόσουν τα βάρη στις κατάλληλες τιμές. Με τον τρόπο αυτό η μέθοδος αγνοεί οποιαδήποτε αντιστοίχιση ανάμεσα στα δύο προβλήματα.

Παρακάτω παρουσιάζονται οι μέθοδοι μεταφοράς που εξετάστηκαν. Μία μέθοδος δεν περιέχει διαδικασία αντιστοίχησης, ενώ οι άλλες δύο περιέχουν. Οι τονισμένες μεταβλητές αφορούν το πρόβλημα στόχο.

1. **Reservoir-Transfer**: Η μέθοδος αυτή δε λαμβάνει υπόψιν κάποια αντιστοίχιση, αλλά μεταφέρει αυτούσιο τον πίνακα γειτνίασης $W' = W$ και την φασματική ακτίνα $\rho' = \rho$ και αρχικοποιεί τυχαία τους πίνακες W^{out} και W^{in} (Σχήμα 5.1).

2. **Mapping**: Με τη μέθοδο αυτή είναι αναγκαία η ύπαρξη μίας αντιστοίχισης που να συνδέει τις μεταβλητές κατάστασης και τις ενέργειες του πηγαίου και του προβλήματος στόχου όπως ακριβώς παρουσιάζεται στο [TWS07] για τη μέθοδο NEAT. Αρχικά ο πίνακας W μεταφέρεται ως έχει. Στη συνέχεια η αντιστοίχιση δείχνει ποια βάρη από τους πίνακες W^{in} και W^{out} από το επιλεγμένο ΔΗΚ του πηγαίου προβλήματος θα αντιγραφούν και σε ποιες θέσεις στους πίνακες W'^{in} και W'^{out} των αρχικών δικτύων του προβλήματος στόχου. Αυτού του είδους οι αντιστοιχίσεις προδιαγράφονται από έναν εξειδικευμένο χρήστη (Σχήμα 5.2). Αυτή η μέθοδος επιλέχθηκε προκειμένου να συγκριθεί η αγνωστικιστική προσέγγιση με μία προσέγγιση αναφοράς, όπως αυτή των δια-προβληματικών αντιστοιχίσεων στα νευρωνικά δίκτυα.

3. **Mapping+Doubling**: Σε αυτή τη μέθοδο χρησιμοποιούνται οι αντιστοιχίσεις για τους πίνακες W^{in} και W^{out}, όμως παράλληλα αυξάνονται οι νευρώνες του ταμιευτηρίου ώστε να ληφθεί υπόψιν και η αυξημένη πολυπλοκότητα του προβλήματος. Η αύξηση είναι άμεσα συνδεδεμένη με τον αριθμό των καταστάσεων και ενεργειών ανάμεσα στα δύο προβλήματα. Χρησιμοποιήθηκε ο διπλασιασμός, εξαιτίας του διπλασιασμού του αριθμού των μεταβλητών κατάστασης και των ενεργειών στα πεδία εφαρμογής. Με αυτόν τον τρόπο δημιουργήθηκε ένα νέος πίνακας γειτνίασης W', όπου ο πίνακας W εμφανίζεται πάνω αριστερά ($1 \leq i' \leq N$, $1 \leq j' \leq N$) και κάτω δεξιά ($N+1 \leq i' \leq N'$, $N+1 \leq j' \leq N'$), με τα υπόλοιπα κελιά του πίνακα να είναι 0 (Σχήμα 5.3). Η μέθοδος αυτή υλοποιήθηκε προκειμένου να μελετηθεί εάν σε προβλήματα με περισσότερες διαστάσεις, οι υπολογιστικές μονάδες

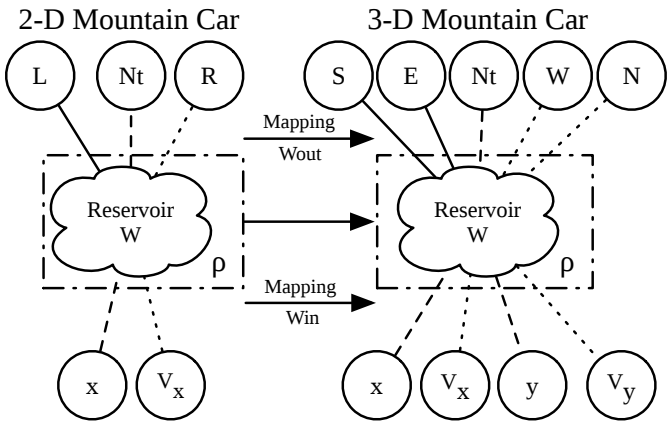


Σχήμα 5.1: Παράδειγμα μεταφοράς ταμιευτήρα για το πρόβλημα του αυτοκινήτου πλαγιάς κατά την αγνωστικιστική περίπτωση. Στην περίπτωση αυτή μόνο το ταμιευτήριο μεταφέρεται και έμμεσα τα N και D, ενώ η φασματική ακτίνα ρ τίθεται άμεσα στο αρχικό γονιδίωμα. Οι πίνακες W^{in} και W^{out} αρχικοποιούνται τυχαία και προσαρμόζονται μέσω της διαδικασίας NEAR.

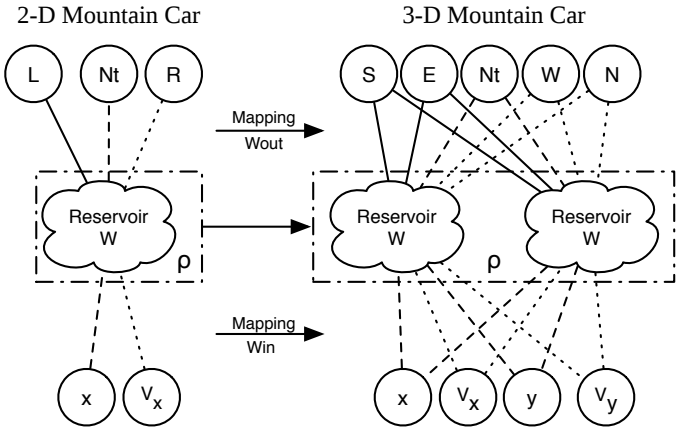
του δικτύου θα πρέπει επίσης να αυξηθούν, δηλαδή εξετάστηκε πειραματικά εάν θα μπορούσαν οι δύο ταμιευτήρες να χειριστούν τον ίδιο αριθμό εισόδων και εξόδων όπως στην πηγαία εργασία, και να αφήσουν τη νευροεξέλιξη να δημιουργήσει διασυνδέσεις μεταξύ τους.

Συνοψίζοντας, παρατηρεί κανείς ότι σε όλες τις παραπάνω περιπτώσεις οι βασικές ιδιότητες ενός ΔΗΚ, N, D και ρ, μεταφέρονται είτε άμεσα είτε έμμεσα.

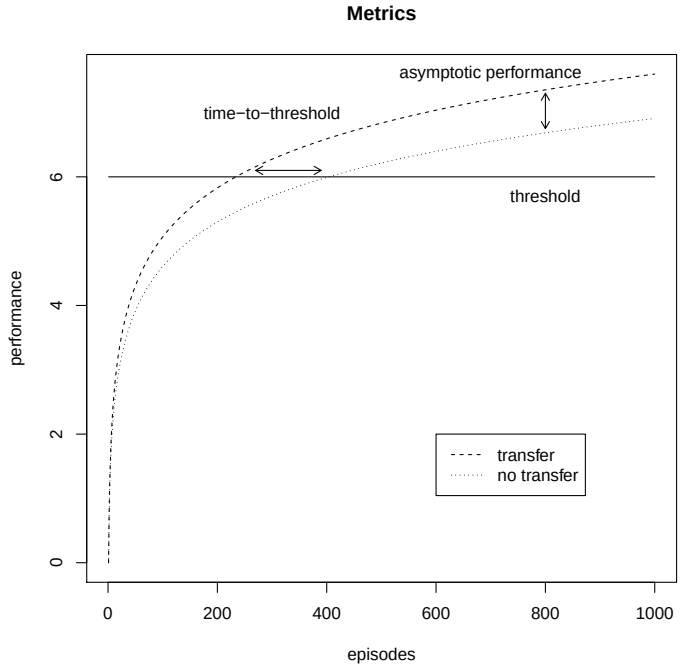
Η μελέτη που πραγματοποιήθηκε αφορά εργασίες πηγής-στόχου που διαφέρουν η μία από την άλλη όσον αφορά την πολυπλοκότητα των προβλημάτων σε επίπεδο ενεργειών και μεταβλητών κατάστασης, αλλά ανήκουν στην ίδια περιοχή (inter-task problems). Για παράδειγμα, στη μία περίπτωση ο πράκτορας μεταβαίνει από ένα πρόβλημα δύο διαστάσεων στο ίδιο πρόβλημα τριών διαστάσεων, αυξάνοντας ταυτόχρονα τους αισθητήρες (μεταβλητές κατάστασης), αλλά και τις ενέργειές του. Οι βασικοί στόχοι που προσπαθεί να επιτύχει η μεταφορά ταμιευτηρίων στη μελέτη αυτή είναι: (α) αυξημένη ασυμπτωτική συμπεριφορά (asymptotic performance) του πράκτορα που προήλθε από τη μεταφοράς μάθησης και, (β) βελτίωση των χρόνων προσαρμογής σε προκαθορισμένα όρια επίδοσης, μια μετρική που είναι γνωστή και ως χρόνος για το κατώφλι (time-to-threshold). Οι μετρικές αυτές εμφανίζονται καλύτερα στο Σχήμα 5.4.



Σχήμα 5.2: Παράδειγμα μεταφοράς ταμιευτήρα για το πρόβλημα του αυτοκινήτου πλαγιάς με δια-εργασιακή αντιστοιχία. Εκτός από τη μεταφορά του ταμιευτήρα, τα βάρη της πηγαίας εργασίας των πινάκων W^{in} και W^{out} μεταφέρονται στην εργασία στόχο μέσω δια-εργασιακής αντιστοιχίας όπως περιγράφεται στο [TWS07].



Σχήμα 5.3: Παράδειγμα μεταφοράς ταμιευτήρα για το πρόβλημα του αυτοκινήτου πλαγιάς με διπλασιασμό ταμιευτηρίου και δια-εργασιακή αντιστοιχία. Το αρχικό ταμιευτήριο μεταφέρεται και διπλασιάζεται, ενώ γίνεται ταυτόχρονα και η δια-εργασιακή αντιστοίχηση.



Σχήμα 5.4: Οι μετρικές μεταφοράς μάθησης που χρησιμοποιήθηκαν για την αξιολόγηση των μεθόδων μεταφοράς ταμιευτηρίων. Ο οριζόντιος άξονας είναι ο αριθμός των επεισοδίων στα οποία λαμβάνει χώρα η μάθηση και ο κάθετος παρουσιάζει μία εικονική μέτρηση της επίδοσης του πράκτορα.

5.3 Πειραματική αξιολόγηση

Η μεθοδολογίες αξιολογήθηκαν εμπειρικά σε δύο προβλήματα ενισχυτικής μάθησης: α) στο αυτοκίνητο πλαγιάς και β) στο χρονοπρογραμματισμό διεργασιών σε εξυπηρετητή, όπως αυτά παρουσιάστηκαν στο Κεφάλαιο 4. Τα δύο αυτά προβλήματα έχουν και στο παρελθόν χρησιμοποιηθεί για έρευνα στο χώρο της μεταφοράς μάθησης. Για το αυτοκίνητο πλαγιάς η πηγαία εργασία είναι το δισδιάστατο πρόβλημα, ενώ το τρισδιάστατο είναι η εργασία στόχος [TJS08]. Για τον χρονοπρογραμματισμό διεργασιών επιλέχθηκαν μόνο οι διεργασίες τύπου 1 και 3 για το πηγαίο πρόβλημα και οι τέσσερις τύποι διεργασιών για την εργασία στόχο [TWS07]. Για το πρόβλημα του αυτοκινήτου πλαγιάς η αντιστοιχία μεταξύ των δύο εργασιών των μεταβλητών κατάστασης και των ενεργειών παρουσιάζεται στον Πίνακα 5.1. Για το πρόβλημα του χρονοπρογραμματισμού οι είσοδοι και οι έξοδοι που αφορούν τον τύπο διεργασίας 1 αντιστοιχήθηκαν και για τον τύπο διεργασίας 2 και, ομοίως, για τον τύπο 3 αντιστοιχήθηκαν με τον τύπο 4.

Κάθε ένα από τα πειράματα επαναλήφθηκε 10 φορές. Ο πληθυσμός αρχικοποιήθηκε με 100 δίκτυα και εξελίχθηκε 50 γενιές. Σε κάθε γενιά, κάθε οργανισμός

Πίνακας 5.1: Πίνακας αντιστοίχισης μεταβλητών κατάστασης και ενεργειών για το πρόβλημα του αυτοκινήτου πλαγιάς.

Κατάσταση 3Δ	Κατάσταση 2Δ	Ενέργεια 3Δ	Ενέργεια 2Δ
x	x	Νεκρά	Νεκρά
y	x	Δυτικά	Αριστερά
v_x	v_x	Ανατολικά	Δεξιά
v_y	v_x	Νότια	Αριστερά
		Βόρεια	Δεξιά

εκπαιδεύτηκε για 100 τυχαίως αρχικοποιημένα επεισόδια. Σε κάθε γενιά ο πρωταθλητής αξιολογήθηκε για επιπλέον $1,000$ επεισόδια με τυχαίες επανεκκινήσεις και η επίδοσή του καταγράφηκε. Με βάση αυτή την καταγεγραμμένη επίδοση επιλέχθηκε το δίκτυο από το οποίο πραγματοποιήθηκε η μεταφορά σύμφωνα με όσα παρουσιάστηκαν στην Ενότητα 5.2.

Ο Πίνακας 5.2 παρουσιάζει τα αποτελέσματα από τις τέσσερις μεθόδους για όλα τα πεδία εφαρμογής. Τα αποτελέσματα δείχνουν την ασυμπτωτική συμπεριφορά του πράκτορα, δηλαδή το συνολικό κέρδος που αποκόμισε με το χρόνο. Η πρώτη παρατήρηση είναι ότι όσο η δυσκολία του προβλήματος αυξάνει, αυξάνουν και τα περιθώρια κέρδους από τη μεταφορά μάθησης.

Πίνακας 5.2: Μέσοι όροι (μ) και τυπικές αποκλίσεις (σ) της επίδοσης γενίκευσης για όλα τα πεδία εφαρμογής και τους αλγορίθμους μεταφοράς μάθησης. Το πρόβλημα 3D-MC είναι το τρισδιάστατο αυτοκίνητο πλαγιάς με Μαρκοβιανό (M) και μη-Μαρκοβιανό (NM) διάνυσμα κατάστασης, ενώ το πρόβλημα SJS είναι αυτό του χρονοπρογραμματισμού εργασιών.

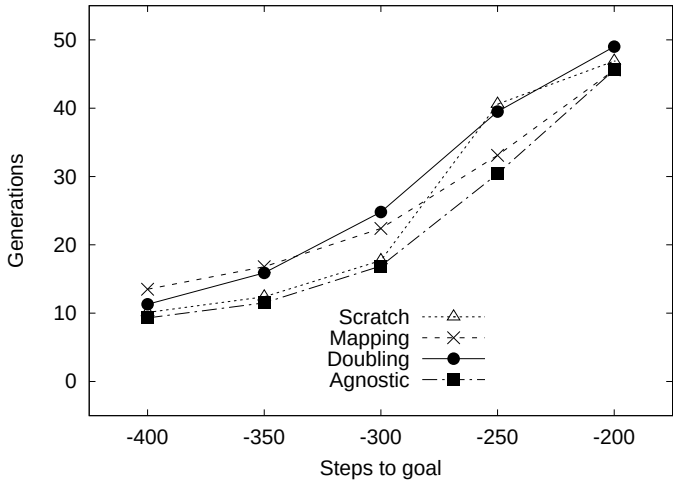
Πρόβλημα	μ	σ
3D-MC-M	-100.04	3.51
3D-MC-M+Reservoir-Transfer	-96.47	4.48
3D-MC-M+Mapping	-97.67	4.40
3D-MC-M+Mapping+Doubling	-97.95	4.32
3D-MC-NM	-293.52	47.86
3D-MC-NM+Reservoir-Transfer	-288.52	62.31
3D-MC-NM+Mapping	-261.92	59.30
3D-MC-NM+Mapping+Doubling	-279.71	33.67
SJS	-5243.08	82.20
SJS+Reservoir-Transfer	-5187.02	31.24
SJS+Mapping	-5176.42	16.74
SJS+Mapping+Doubling	-5187.17	30.46

Το περιβάλλον χρονοπρογραμματισμού μπορεί να θεωρηθεί δυσκολότερο πεδίο εφαρμογής από αυτό του αυτοκινήτου πλαγιάς, τουλάχιστον σε ότι αφορά τον αριθμό των μεταβλητών κατάστασης και των ενεργειών που πρέπει να διαχειριστεί

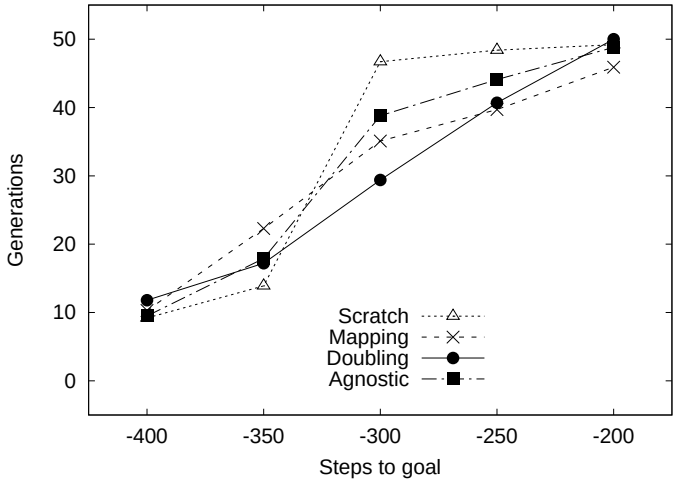
ο πράκτορας. Στο συγκεκριμένο πεδίο εφαρμογής, όσον αφορά στη μέση ασυμπτωτική συμπεριφορά, όλες οι μέθοδοι μεταφοράς ξεπέρασαν την εξέλιξη χωρίς μεταφορά μάθησης. Οι διαφορές στην απόδοση μεταξύ των μεθόδων μεταφοράς και χωρίς μεταφορά είναι στατιστικά σημαντικές με βάση το t-test στο 95%. Επιπλέον, μικρότερη μεταβολή στην ασυμπτωτική συμπεριφορά των αλγορίθμων παρατηρήθηκε όταν υπήρχε μεταφορά. Αυτό αποκτά σημαντική αξία, όταν αναφερόμαστε σε αυτόνομους πράκτορες που πρέπει να χτίσουν τέτοιους μηχανισμούς χωρίς την ανθρώπινη επίβλεψη.

Η μεταφορά μάθησης μέσω δια-εργασιακής αντιστοίχισης δεν κατάφερε να επιβληθεί στατιστικά έναντι της αγνωστικιστικής μεθόδου. Αυτό μπορεί να υποδεικνύει ότι το ταμιευτήριο μαζί με τις ιδιότητές του μπορεί να παρακρατήσει σημαντική γνώση του πεδίου εφαρμογής. Αυτή η γνώση μπορεί να είναι συγκεκριμένοι υπο-γράφοι (sub-graphs), οι οποίοι με συγκεκριμένες τιμές στα βάρη είναι δυνατόν να κάνουν χρονικούς υπολογισμούς και να παράγουν αποδοτικές πολιτικές ταχύτερα από ότι αρχίζοντας από το μηδέν.

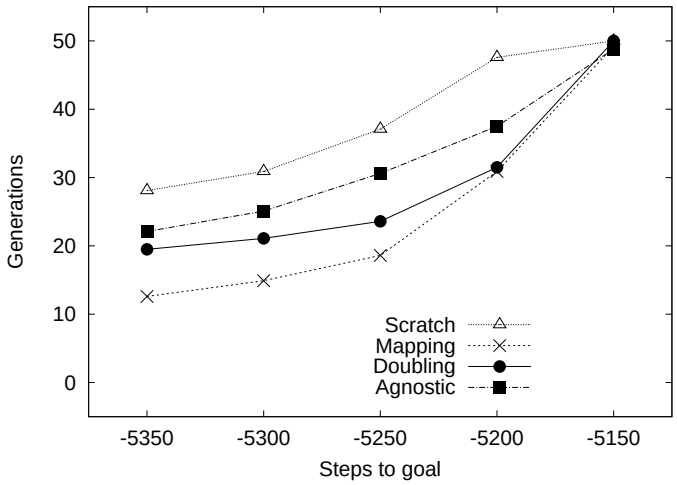
Οι πρακτικές μεταφοράς ταμιευτηρίων επιβλήθηκαν και στις κούρσες κατωφλιών επίδοσης έναντι της βασικής μεθόδου. Αυτό φαίνεται στα Σχήματα 5.5, 5.6 και 5.7. Η μεταφορά μάθησης κέρδισε 7 φορές, ακολουθούμενη από την αντιστοίχηση με 6 φορές και το διπλασιασμό του ταμιευτηρίου με 1 φορά. Επίσης, αξίζει να παρατηρήσουμε ότι όσον αφορά την Μαρκοβιανή παραλλαγή του αυτοκινήτου πλαγιάς, η αγνωστικιστική μέθοδος φαίνεται να είναι ο καλύτερος αλγόριθμος, καθώς διατηρεί πάντα μια μικρή διαφορά έναντι των υπολοίπων.



Σχήμα 5.5: Ο αριθμός των γενεών που χρειάστηκαν για τον υπερκερασμό των προκαθορισμένων κατωφλιών στο Μαρκοβιανό πρόβλημα του αυτοκινήτου πλαγιάς.



Σχήμα 5.6: Ο αριθμός των γενεών που χρειάστηκαν για τον υπερκερασμό των προκαθορισμένων κατωφλιών στο μη Μαρκοβιανό πρόβλημα του αυτοκινήτου πλαγιάς.



Σχήμα 5.7: Ο αριθμός των γενεών που χρειάστηκαν για τον υπερκερασμό των προκαθορισμένων κατωφλιών στο πρόβλημα χρονοπρογραμματισμού διεργασιών.

5.4 Σύνοψη

Στο κεφάλαιο αυτό παρουσιάστηκαν μέθοδοι μεταφοράς των ταμιευτηρίων ικανών ΔΗΚ από μια πηγαία διεργασία σε μια διεργασία στόχο με σκόπο την επιτάχυνση και βελτίωση της διαδικασίας προσαρμογής στην εργασία στόχο. Η βασική συμβολή είναι η παρουσίαση μιας αγνωστικιστικής μεθόδου που έχει παραπλήσια επίδοση με μια μέθοδο αναφοράς, αυτής των αντιστοιχίσεων διεργασιών (inter-

task mapping) και συμβάλλει στην αυτοματοποίηση της διαδικασίας μεταφοράς μάθησης.

Οι μέθοδοι εξετάστηκαν σε τρία διαφορετικά προβλήματα σε δύο πεδία εφαρμογής. Τα αποτελέσματα υποδεικνύουν ότι καθώς η δυσκολία των εργασιών που καλείται να φέρει εις πέρας ο πράκτορας αυξάνει, σε όρους πολυπλοκότητας, αλλά και σε καταστάσεις και ενέργειες, τα κέρδη από τη μεταφορά γνώσης αυξάνουν. Μπορεί να υποθέσει λοιπόν κανείς ότι, καθώς η δυσκολία του προβλήματος αυξάνει, ο ρόλος των μεθόδων αρχικοποίησης του πληθυσμού τω μεθόδων νευροεξέλιξης θα γίνεται ολοένα και πιο σημαντικός. Η μέθοδος αντιστοίχισης επιβλήθηκε έναντι των υπολοίπων, καθώς είναι αρκετά δυσκολότερο μία μέθοδος που αγνοεί τη γνώση των ειδικών να είναι ανώτερη. Ωστόσο, η μεταφορά του ταμιευτηρίου φαίνεται ως μια εφικτή λύση ιδιαίτερα όταν η πολυπλοκότητα του προβλήματος είναι πολύ μεγάλη, και οι αντιστοιχίσεις γίνονται δυσδιάκριτες. Η μεθοδολογία NEAR είναι ούτως ή άλλως ικανή να βρίσκει πολύ ικανοποιητικές λύσεις με ταχύτητα στα παραπάνω προβλήματα, γεγονός που καθιστά κάθε στατιστικά σημαντική βελτίωση στην επίδοσή της ένα υποσχόμενο αποτέλεσμα για ακόμα περισσότερα κέρδη, όσο η πολυπλοκότητα του προβλήματος αυξάνεται.

Κεφάλαιο 6

Πρόβλεψη χρονοσειρών

Παρά το γεγονός ότι η συγκεκριμένη διατριβή εστιάζει σε προβλήματα ενισχυτικής μάθησης, η αξιολόγηση της μεθόδου NEAR έγινε και σε προβλήματα πρόβλεψης χρονοσειρών, ένα σχήμα επιβλεπόμενης μάθησης, ώστε να είναι περισσότερο πλήρης. Για την αξιολόγηση χρησιμοποιήθηκαν τέσσερις χρονοσειρές:

1. η χρονοσειρά Mackey-Glass,
2. ο ταλαντωτής πολλαπλών ημίτονων (multiple superimposed oscillator),
3. η χρονοσειρά Lorentz, και
4. η χρονοσειρά ζήτησης ενεργειακού φορτίου από την Ελληνική αγορά του χρηματιστηρίου ηλεκτρικής ενέργειας.

6.1 Δυναμικά συστήματα

Για τις τρεις πρώτες περιπτώσεις χρονοσειρών χρησιμοποιήθηκε ακριβώς η πειραματική διάταξη και οι συμβολισμοί που αναφέρονται στο [RI09], έτσι ώστε τα αποτελέσματα να είναι συγκρίσιμα. Σε κάθε πρόβλημα κατασκευάστηκαν τέσσερις ακολουθίες:

1. η ακολουθία καθαρισμού $\mathbf{U}^{wash}$ μήκους W, που χρησιμεύει ως είσοδος για τα 100 πρώτα βήματα, έτσι ώστε να ξεπλύνει αρχικά μεταβατικά φαινόμενα (initial transient dynamics) του δικτύου. Η αρχική κατάσταση του δικτύου είναι μηδέν.

2-3. οι ακολουθίες εισόδου και στόχου, $\mathbf{U}^{train}$ και $\mathbf{D}^{train}$, μήκους T, που χρησιμοποιούνται μετά την ακολουθία ξεπλύματος έτσι ώστε να εκπαιδευτούν τα βάρη εξόδου του ΔΗΚ με τη μέθοδο των ελαχίστων τετραγώνων. Το μήκος εξαρτάται από το εκάστοτε πρόβλημα. Στόχος είναι η πρόβλεψη της χρονοσειράς ένα βήμα μετά. Με βάση αυτό, $d^{train}(t) = u^{train}(t+1)$.

4. η ακολουθία επαλήθευσης $\mathbf{D}^{valid}$ μήκους V. Το μήκος V ορίζεται με βάση το πρόβλημα.

Με βάση τα παραπάνω, εάν η ακολουθία συμβολίζεται με **U**, οι τέσσερις ακολουθίες δίνονται από τις Εξισώσεις 6.1 έως 6.4:

$$\mathbf{U}^{wash} = \{u(0) \dots u(W-1)\} \tag{6.1}$$

$$\mathbf{U}^{train} = \{u(W) \dots u(W+T-1)\} \tag{6.2}$$

$$\mathbf{D}^{train} = \{u(W+1) \dots u(W+T)\} \tag{6.3}$$

$$\mathbf{D}^{valid} = \{u(W+T) \dots u(W+T+V-1)\} \tag{6.4}$$

Ο υπολογισμός της καταλληλότητας κάθε οργανισμού με βάση το σφάλμα που προκύπτει από τη χρονοσειρά επαλήθευσης θα καθιστούσε την αναφορά του σφάλματος της μεθόδου μικρότερη από το πραγματικό σφάλμα γενίκευσης. Για το λόγο αυτό η καταλληλότητα υπολογίζεται σε J μικρότερα ζευγάρια ακολουθιών ελέγχου $\mathbf{U}^{wash}$, $\mathbf{U}^{train}$ και $\mathbf{D}^{train}$. Κάθε ένα έχει μια ακολουθία ξεπλύματος μήκους W και μια ακολουθία πρόβλεψης μήκους F. Ο αριθμός των ακολουθιών J υπολογίζεται από την Εξίσωση 6.5:

$$J = \lfloor \frac{(W+T)}{(W+F)} \rfloor - 1 \tag{6.5}$$

Με το πέρας της ακολουθίας ξεπλύματος, το δίκτυο οδηγείται αναδρομικά χρησιμοποιώντας σε κάθε βήμα την έξοδο ως είσοδο για το επόμενο βήμα και κατ' επέκταση για την επόμενη πρόβλεψη. Η καταλληλότητα κάθε οργανισμού δίνεται από την Εξίσωση 6.6:

$$\Phi(g) = \frac{1}{NRMSE_g^{test}} \tag{6.6}$$

όπου $NRMSE^{test}$ είναι η κανονικοποιημένη τετραγωνική ρίζα του μέσου τετραγωνισμένου σφάλματος (normalized root mean square error) για τις ακολουθίες ελέγχου (test) και δίνεται από την Εξίσωση 6.7:

$$NRMSE^{test} = \frac{1}{J} \sum_{j=0}^{J-1} NRMSE \tag{6.7}$$

Αν υποθέσουμε, όπως στην τρέχουσα περίπτωση, $NRMSE$ με μία έξοδο, ο τύπος υπολογισμού δίνεται από την εξίσωση 6.8:

$$NRMSE = \sqrt{\sum_{i=0}^{F-1} \frac{(o(i) - d(i))^2}{var(\mathbf{U}^{train})}} \tag{6.8}$$

όπου $var(\mathbf{X})$ η διακύμανση της ακολουθίας **X**, $o()$ η έξοδος του ΔΗΚ και $d()$ η επιθυμητή τιμή.

Για την αξιολόγηση της συμπεριφοράς των δικτύων που προέκυψαν μετά την εξελικτική διαδικασία, για παράδειγμα για το δίκτυο πρωταθλητή, χρησιμοποιείται

η ακολουθία επαλήθευσης, $\mathbf{D}^{valid}$. Οι $\mathbf{U}^{wash}$ και $\mathbf{U}^{train}$ γίνονται ακολουθίες ξεπλύματος και στη συνέχεια το δίκτυο μπαίνει σε αναδρομική λειτουργία προσπαθώντας να προβλέψει τη χρονοσειρά επαλήθευσης. Το $NRMSE^{valid}$ υπολογίζεται από την Εξίσωση 6.8.

Οι τρεις χρονοσειρές δυναμικών συστημάτων που χρησιμοποιήθηκαν ήταν οι ακόλουθες:

1. Η χρονοσειρά Mackey-Glass, που περιγράφει την αναγέννηση λευκοκυττάρων (leukocytes) για αρρώστους με λευχαιμία (Σχήμα 6.1(a)). Χρησιμοποιήθηκε το μοντέλο που δίνεται από την Εξίσωση 6.9 με $\alpha = 0.2$, $b = 0.1$, $c = 10$ και $\tau = 17$:

$$\dot{y}(t) = \frac{\alpha y(t-\tau)}{1 + (y(t) - \tau)^c} - by(t) \tag{6.9}$$

2. Ο ταλαντωτής πολλαπλών ημίτονων (Σχήμα 6.1(b)), ο οποίος δημιουργήθηκε προσθέτοντας οκτώ ημιτονοειδής συναρτήσεις με συχνότητες που δεν είναι ακέραια πολλαπλάσια η μία της άλλης, όπως περιγράφεται από την Εξίσωση 6.10 με $\lambda_1 = 0.2$, $\lambda_2 = 0.311$, $\lambda_3 = 0.42$, $\lambda_4 = 0.51$, $\lambda_5 = 0.63$, $\lambda_6 = 0.74$, $\lambda_7 = 0.85$ και $\lambda_8 = 0.97$:

$$y(k) = \sum_{i=1}^{8} sin(\lambda_i k) \tag{6.10}$$

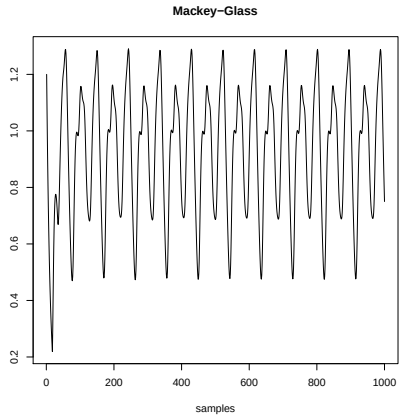
3. Ο έλκτης Lorentz (Lorentz attractor) (Σχήμα 6.1(c)), όπου το σύστημα αποτελείται από τρεις συζευγμένες μη γραμμικές διαφορικές εξισώσεις, τις Εξισώσεις 6.11, 6.12 και 6.13 με $\alpha = 10$, $b = 28$ και $c = 8/3$, ενώ η χρονοσειρά προκύπτει από τη συντεταγμένη x του συστήματος.

$$\dot{x} = \alpha(y - x) \tag{6.11}$$
$$\dot{y} = x(b - z) - y \tag{6.12}$$
$$\dot{z} = xy - cz \tag{6.13}$$

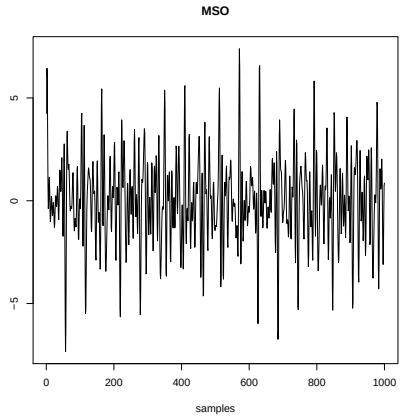
Ο Πίνακας 6.1 συνοψίζει τα μήκη όλων των ακολουθιών στα τρία προβλήματα με βάση την παραπάνω πλατφόρμα αξιολόγησης.

Πίνακας 6.1: Τα μήκη των ακολουθιών που θα χρησιμοποιηθούν για πειραματισμό.

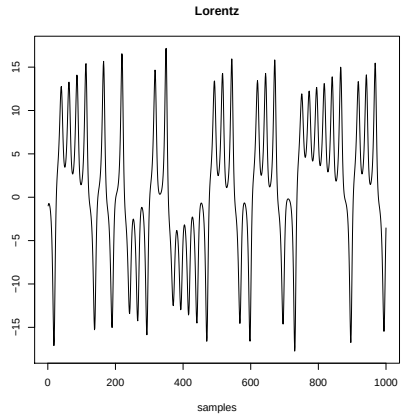
Μήκος	Mackey-Glass	MSO	Lorentz
W	100	100	100
T	2,900	1,000	2,900
V	84	1,000	20
F	50	1,000	20



(a) η χρονοσειρά Mackey-Glass



(b) η χρονοσειρά multiple superimposed oscillator



(c) η χρονοσειρά Lorentz

Σχήμα 6.1: Οι χρονοσειρές των τριών πεδίων εφαρμογής με δειγματοληψία σε $1,000$ σημεία.

Εφόσον επιλέχθηκε το ίδιο πειραματικό μοντέλο με αυτό που αναφέρεται στο [RI09], τα αποτελέσματα της νευροεξέλιξης αυτής της μεθόδου μπορούν να συγκριθούν με τη μέθοδο NEAR. Για πιο δίκαιη αξιολόγηση, αυξήσαμε τον αρχικό αριθμό των νευρώνων στη δεξαμενή σε 25, καθώς η μέθοδος στο [RI09] ξεκινά με 30 νευρώνες αλλά έχει τη δυνατότητα να διαγράψει κόμβους. Χρησιμοποιείται ένας πληθυσμός από 50 γονιδιώματα και ο αριθμός των εξελικτικών βημάτων διαρκεί 100 γενιές. Τα αποτελέσματα προκύπτουν από τον μέσο όρο 30 εκτελέσεων του αλγορίθμου. Επιπλέον, εξετάστηκε η επίδοση τυχαία δημιουργημένων ΔΗΚ, όπου δημιουργήθηκε ένα πλήθος από ΔΗΚ προκειμένου να βρεθεί ένα με καλές επιδόσεις. Δύο τέτοιες αναζητήσεις εφαρμόστηκαν, μία με 50 νευρώνες ταμιευτηρίου (ESN-50) και μία με 100 (ESN-100). Για κάθε εκτέλεση της αναζήτησης αξιολογήθηκαν 100 ΔΗΚ. Το καλύτερο ΔΗΚ επιλέγεται με βάση το σφάλμα στην ακολουθία τεστ, ενώ στο τέλος, ως σφάλμα γενίκευσης παρουσιάζεται το σφάλμα για την ακολουθία επαλήθευσης. Στο σημείο αυτό θα πρέπει να παρατηρήσουμε ότι οι όροι έλεγχος (test) και επαλήθευση (validation) χρησιμοποιούνται ανάποδα από την κλασική βιβλιογραφία της μηχανικής μάθησης, ωστόσο ακολουθούμε τη μοντελοποίηση στο [RI09].

6.1.1 Mackey-Glass

Για το πρόβλημα Mackey-Glass, τα αποτελέσματα για τις μετρικές $NRMSE^{test}$ και $NRMSE^{val}$ συνοψίζονται στον Πίνακα 6.2. Η μέθοδος NEAR αναζητά μία βέλτιστη τοπολογία, ενώ ο αλγόριθμος ελαχίστων τετραγώνων (least-squares or LS) προσαρμόζει τα βάρη εξόδου στις βέλτιστες τιμές με βάση το σετ δεδομένων εκπαίδευσης. Η μέθοδος NEAR+LS υπερισχύει της μεθόδου ESN-100, παρόλο που τα ΔΗΚ που ανακαλύπτει η NEAR έχουν περίπου 35 νευρώνες ταμιευτηρίου. Από την άλλη, η NEAR+MUT, χρησιμοποιώντας μεταλλάξεις για την εύρεση των κατάλληλων βαρών, δεν καταφέρνει να μάθει το μοντέλο δημιουργίας της χρονοσειράς Mackey-Glass.

Πίνακας 6.2: Τα σφάλματα $NRMSE^{test}$ και $NRMSE^{val}$ για τη χρονοσειρά Mackey-Glass.

Αλγόριθμος	$NRMSE^{test}$	$NRMSE^{val}$
NEAR+LS	$5.39 \cdot 10^{-3}$	$\mathbf{2.85 \cdot 10^{-3}}$
NEAR+MUT	$8.09 \cdot 10^{-1}$	$9.57 \cdot 10^{-1}$
[RI09]	$3.62 \cdot 10^{-2}$	$6.48 \cdot 10^{-2}$
ESN-50	$1.02 \cdot 10^{-2}$	$1.04 \cdot 10^{-2}$
ESN-100	$1.56 \cdot 10^{-3}$	$3.84 \cdot 10^{-3}$

6.1.2 Multiple-Sinewave Oscillator

Ο Πίνακας 6.3 παρουσιάζει τα σφάλματα για τη χρονοσειρά MSO. Η μέθοδος NEAR δεν καταφέρνει αυτή τη φορά να προσαρμόσει τα βάρη των ΔΗΚ έτσι

ώστε να μάθουν τη δυναμική της χρονοσειράς. Αυτό είναι ένα γνωστό μειονέκτημα των ΔΗΚ και της υποστήριξης που παρέχουν όταν υπάρχουν επικαλυπτόμενες γεννήτριες σημάτων, εξαιτίας της ισχυρής σύζευξης μεταξύ των νευρώνων του ταμιευτηρίου [LJ09]. Περαιτέρω εξέταση θα πρέπει να γίνει σε αυτήν την περίπτωση, η οποία θα συζητηθεί στο Κεφάλαιο 10.

Πίνακας 6.3: Τα σφάλματα $NRMSE^{test}$ και $NRMSE^{val}$ για τη χρονοσειρά MSO.

Αλγόριθμος	$NRMSE^{test}$	$NRMSE^{val}$
NEAR+LS	$9.33 \cdot 10^{-1}$	$9.46 \cdot 10^{-1}$
NEAR+MUT	$9.98 \cdot 10^{-1}$	$9.87 \cdot 10^{-1}$
[RI09]	$3.55 \cdot 10^{-3}$	$\mathbf{8.95 \cdot 10^{-3}}$
ESN-50	$9.82 \cdot 10^{-1}$	$9.89 \cdot 10^{-1}$
ESN-100	$9.75 \cdot 10^{-1}$	$9.84 \cdot 10^{-1}$

6.1.3 Lorentz Attractor

Τα αποτελέσματα για το πρόβλημα πρόβλεψης της χρονοσειράς του έλκτη Lorentz εμφανίζονται στον Πίνακα 6.4. Η NEAR+LS υπερισχύει έναντι οποιασδήποτε άλλης μεθόδου από αυτές που εξετάστηκαν αν συγκρίνουμε το σφάλμα επαλήθευσης. Επιπλέον, ο συγκεκριμένος αλγόριθμος δείχνει και τις ικανότητες γενίκευσης που διαθέτει καθώς το σφάλμα επαλήθευσης είναι χαμηλότερο από το $NRMSE^{test}$.

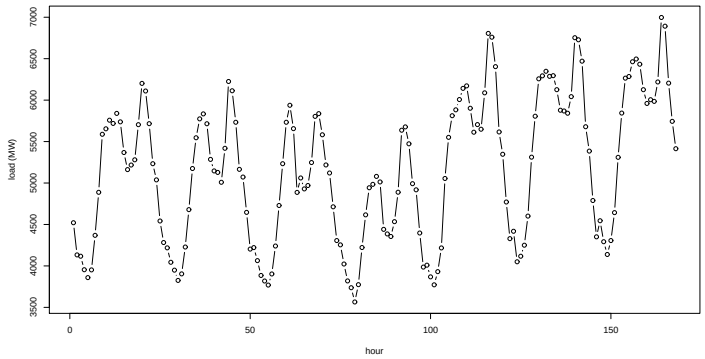
Πίνακας 6.4: Τα σφάλματα $NRMSE^{test}$ και $NRMSE^{val}$ για τη χρονοσειρά Lorentz Attractor.

Αλγόριθμος	$NRMSE^{test}$	$NRMSE^{val}$
NEAR+LS	$4.44 \cdot 10^{-2}$	$\mathbf{7.93 \cdot 10^{-3}}$
NEAR+MUT	$8.80 \cdot 10^{-1}$	$4.44 \cdot 10^{-1}$
[RI09]	$2.16 \cdot 10^{-2}$	$1.20 \cdot 10^{-2}$
ESN-50	$2.63 \cdot 10^{-1}$	$4.04 \cdot 10^{-2}$
ESN-100	$5.66 \cdot 10^{-2}$	$4.01 \cdot 10^{-2}$

6.2 Χρονοσειρά ενεργειακού φορτίου

Για την επίδειξη των δυνατοτήτων της μεθόδου NEAR, χρησιμοποιήσαμε ακόμη ένα πρόβλημα πρόβλεψης χρονοσειρών με βάση ιστορικά δεδομένα από το Ελληνικό χρηματιστήριο ενέργειας (http://www.desmie.gr/). Στο Σχήμα 6.2 παρουσιάζεται η χρονοσειρά φορτίου για περίοδο μίας εβδομάδας.

Τα δεδομένα που συλλέχθηκαν αφορούν την περίοδο από την 1η Μαΐου 2003 έως τις 30 Σεπτεμβρίου 2010. Το πρόβλημα που καλείται να επιλύσει η μέθοδος



Σχήμα 6.2: Χρονοσειρά ηλεκτρικού φορτίου για την περίοδο μιας εβδομάδας.

NEAR είναι η ημερήσια πρόβλεψη του ηλεκτρικού φορτίου για περίοδο ενός μήνα. Τα δεδομένα παρέχονται ανά ώρα και λαμβάνονται καθημερινά σε 24-ώρα πακέτα από την ώρα $00:00$ έως την ώρα $23:00$ τη μέρα d, αφού γίνει η εκκαθάριση στην αγορά. Στόχος είναι η πρόβλεψη του φορτίου την επόμενη μέρα $d+1$, γνωρίζοντας μόνο δεδομένα από τη μέρα d.

Αξιολογήθηκαν προσεγγίσεις με ταμιευτήρες νευρώνων σε σχέση με αλγορίθμους παλινδρόμησης που υπάρχουν στην πλατφόρμα WEKA [HFH+09], όπως η γραμμική παλινδρόμηση και τα δένδρα παλινδρόμησης M5'. Δημιουργήθηκαν ξεχωριστά μοντέλα όπου κάθε ένα θα προβλέπει την τιμή την ώρα i με βάση όλες τις ωριαίες τιμές της προηγούμενης ημέρας:

$$\hat{x}_i^d = f(x_0^{d-1}, x_1^{d-1}, \ldots, x_{23}^{d-1}), i = 0 \ldots 23$$

όπου f η συνάρτηση προσέγγισης που παράγει κάθε αλγόριθμος.

Η αξιολόγηση έγινε ως προς το μέσο απόλυτο ποσοστό σφάλματος (Mean Absolute Percentage Error - MAPE) και το μέγιστο σφάλμα (MAXIMAL) που δίνονται από τις παρακάτω εξισώσεις:

$$MAPE = 100 \cdot \frac{\sum_{i=1}^{N} \sum_{j=1}^{H} \frac{|L_{Rij} - L_{Pij}|}{L_{Rij}}}{N \cdot H} \tag{6.14}$$

και

$$MAXIMAL = max(|L_{Rij} - L_{Pij}|) \tag{6.15}$$

όπου L_{Rij} είναι η πραγματική τιμή τη μέρα i και την ώρα j και L_{Pij} η πρόβλεψη.

Για τις προσεγγίσεις με ΔΗΚ (απλά ΔΗΚ και NEAR), χρησιμοποιήσαμε δείγματα από ένα χρόνο (365 δείγματα) ως ακολουθία ξεπλύματος, έτσι ώστε να αρχικοποιηθεί το δίκτυο και να μειωθούν τα αρχικά φαινόμενα, και δύο χρόνια (730 δείγματα) ως δεδομένα εκπαίδευσης. Τα MAPE και MAXIMAL υπολογίζονται καθημερινά. Τα σφάλματα αναφέρονται σε περίοδο ενός μηνός, του Σεπτεμβρίου

2010 (30 δείγματα). Στην περίπτωση του NEAR και προκειμένου να μην εξελιχθούν τα ΔΗΚ βελτιστοποιώντας τα ως προς τα δείγματα του Σεπτεμβρίου, χρησιμοποιήθηκε ως ακολουθία ελέγχου ο Οκτώβριος του 2010. Η συνάρτηση καταλληλότητας ορίστηκε ως $1/MAPE$. Ο πληθυσμός περιείχε 50 οργανισμούς που εξελίχθηκαν για 100 γενιές. Χρησιμοποιήθηκε ο ίδιος αρχικός αριθμός νευρώνων όπως στην περίπτωση των κλασικών ΔΗΚ.

Κάθε ΔΗΚ έχει 25 εισόδους, μία για κάθε τιμή φορτίου για κάθε ώρα της προηγούμενης ημέρας και μία επιπλέον είσοδο πόλωσης (bias), καθώς και 24 εξόδους, μία για κάθε τιμή φορτίου για κάθε ώρα της επόμενης ημέρας. Οι παράμετροι που χρησιμοποιήθηκαν για τα ΔΗΚ φαίνονται στον Πίνακα 6.5. Για τα μοντέλα παλινδρόμησης (γραμμικής και M5') δημιουργήθηκαν 24 σετ δεδομένων και από αυτά προέκυψαν 24 μοντέλα πρόβλεψης, ένα για κάθε ώρα της ημέρας.

Πίνακας 6.5: Οι παράμετροι αρχικοποίησης των ΔΗΚ και μέσα σε παρένθεση αυτές που βρέθηκαν από τη NEAR.

ESN Parameter	**Load: Init. (Found)**
Φασματική Ακτίνα	0.85 (0.79)
Πυκνότητα	25% (11%)
Συνάρτηση ενεργοποίησης ταμιευτηρίου (f)	$tanh$
Συνάρτηση ενεργοποίησης εξόδου (g)	$identity$
Μέγεθος ταμιευτηρίου	200 (232)
Επίπεδο θορύβου	10^{-7}

Ο Πίνακας 6.6 παρουσιάζει τις μέσες τιμές (AVG) και τις τυπικές αποκλίσεις (STD) των σφαλμάτων των τιμών του Σεπτεμβρίου του 2010, για όλους τους αλγορίθμους. Το σφάλμα MAPE είναι χαμηλό και υποδεικνύει μία αρκετά ακριβή πρόβλεψη της τιμής φορτίου. Οι αλγόριθμοι που χρησιμοποιούν ΔΗΚ υπερισχύουν έναντι των άλλων, κυρίως λόγω της δυνατότητάς τους να διατηρούν μνήμη και να υπολογίζουν χρονικά χαρακτηριστικά, σημαντικά για την εργασία της πρόβλεψης χρονοσειρών. Η μεθοδολογία NEAR καταφέρνει να βελτιστοποιήσει τους ταμιευτήρες και τις ιδιότητές τους, οδηγώντας τα σφάλματα σε χαμηλά επίπεδα με δυνατές ικανότητες γενίκευσης.

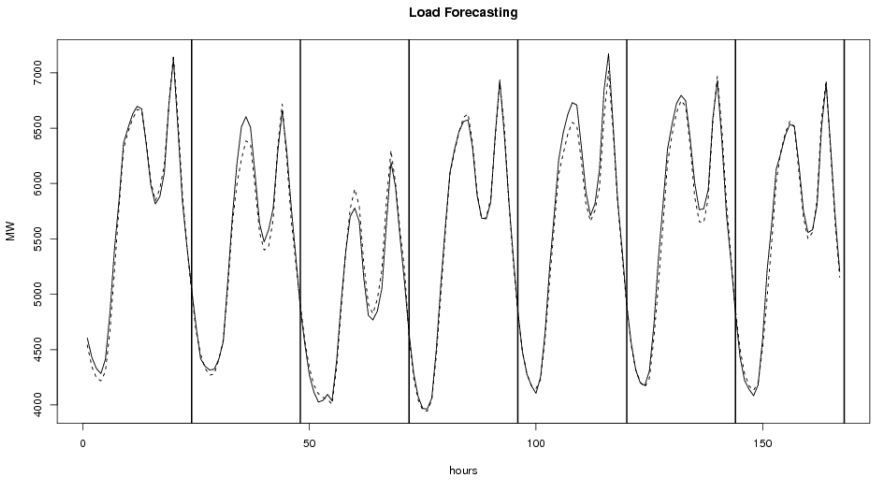
Στο Σχήμα 6.3 παρουσιάζεται η πρόβλεψη και η πραγματική τιμή για το ηλεκτρικό φορτίο την εβδομάδα μεταξύ 24 και 30 Σεπτεμβρίου του 2010 χρησιμοποιώντας τη μεθοδολογία NEAR. Είναι ενδεικτική η ικανότητα των ΔΗΚ να προβλέπουν χρονοσειρές με κάποιου είδους περιοδικότητα.

6.3 Σύνοψη

Στο κεφάλαιο αυτό παρουσιάστηκαν οι δυνατότητες της μεθόδου NEAR στην πρόβλεψη χρονοσειρών, ενός προβλήματος επιβλεπόμενης μάθησης. Η αξιολόγη-

Πίνακας 6.6: Οι μέσοι όροι και οι τυπικές αποκλίσεις για τα σφάλματα MAPE και MAXIMAL στην περίοδο 30 ημερών υπό εξέταση για όλους τους αλγορίθμους που εξετάστηκαν για την πρόβλεψη φορτίου.

Method	**MAPE AVG (%) 30 Days**	**MAXIMAL (MW) 30 Days**
ESN	2.14	316.13
NEAR	**1.91**	**280.14**
Lin. Regression	3.08	436.32
M5'	3.03	570.97
Method	**MAPE STD (%) 30 Days**	**MAXIMAL STD (MW) 30 Days**
ESN	1.20	155.16
NEAR	0.87	114.66
Lin. Regression	1.85	163.57
M5'	1.16	178.49



Σχήμα 6.3: Πρόβλεψη ηλεκτρικού φορτίου για την εβδομάδα 24 με 30 Σεπτεμβρίου του 2010. Με συμπαγή γραμμή είναι το πραγματικό φορτίο, ενώ με διακεκκομένη γραμμή είναι πρόβλεψη του φορτίου.

ση έγινε σε τέσσερα προβλήματα χρονοσειρών, απέναντι σε τυχαία δημιουργημένα ΔΗΚ αλλά και αλγορίθμους αναφοράς. Ένα από τα συμπεράσματα είναι ότι η βελτιστοποίηση των ΔΗΚ μέσω της NEAR έχει απτά αποτελέσματα στα σφάλματα πρόβλεψης. Ένα άλλο συμπέρασμα είναι ότι η συνεργασία μάθησης και εξέλιξης λειτούργησε πολύ καλύτερα από ότι στα προβλήματα ενισχυτικής μάθησης που εξετάστηκαν στο Κεφάλαιο 4, όπου η μέθοδος NEAR+PS υπερκέρασε τη μέθοδο NEAR+TD σε όλες σχεδόν τις περιπτώσεις. Ο βασικός λόγος είναι ότι με τη

μάθηση χρονικών διαφορών ο αλγόριθμος προσπαθεί να υπολογίσει την πραγματική τιμή Q, δηλαδή την πρόβλεψη του μελλοντικού προσδοκώμενου κέρδους, ενώ αντίθετα με τις μεθόδους αναζήτησης πολιτικής, ο αλγόριθμος μένει ικανοποιημένος εφόσον επιλεγεί η σωστή κίνηση, ανεξαρτήτως της τιμής Q. Κάτι τέτοιο δεν είναι ικανοποιητικό στα προβλήματα πρόβλεψης χρονοσειρών, όπου χρειαζόμαστε ακρίβεια στην τιμή της εξόδου, προκειμένου να έχουμε σωστή πρόβλεψη και χαμηλό σφάλμα. Πρόκειται λοιπόν για μία επιλογή που καλείται να κάνει ο χρήστης, ανάλογα με τις απαιτήσεις και τη φύση του εκάστοτε προβλήματος.

Κεφάλαιο 7

Εφαρμογή 1: Βελτιστοποίηση πολιτικής πωλήσεων σε δυναμικό περιβάλλον εφοδιαστικής αλυσίδας

Μια τυπική εφοδιαστική αλυσίδα αποτελείται από διάφορες οντότητες, όπως τους προμηθευτές, τους κατασκευαστές, τους διανεμητές και τους πελάτες, ενώ η λειτουργικότητά της ορίζεται σε διάφορα επίπεδα (στρατηγικό, τακτικό, επιχειρησιακό) με παράλληλη και συνεχή δράση και αλληλεπίδραση. Όλες οι οντότητες αλληλεπιδρούν για να βελτιστοποιήσουν τη ροή των αγαθών από την παραγωγή στην κατανάλωση σε αντιστοιχία με την αντίθετη ροή της πληροφορίας από τη ζήτηση στην προσφορά. Στόχος της περιοχής της διαχείρισης της προμηθευτικής αλυσίδας είναι η μείωση του κόστους, η αύξηση του κέρδους και η μεγιστοποίηση του επιπέδου των υπηρεσιών και της ικανοποίησης του πελάτη [SLKSL03].

Το περιβάλλον στο οποίο γίνεται η διαχείριση της εφοδιαστικής αλυσίδας είναι ιδιαίτερα απαιτητικό, καθώς πρόκειται για ένα περιβάλλον (συνεργατικό και ανταγωνιστικό μαζί) μη πλήρως παρατηρήσιμο, δυναμικό, στοχαστικό και εξαιρετικά πολύπλοκο. Αλλαγές σε ένα σημείο του συστήματος μπορεί να επιφέρουν αλλαγές με τη μορφή χιονοστιβάδας, όπως συμβαίνει στο φαινόμενο του μαστιγίου (bullwhip effect). Σε αυτή την περίπτωση μια μικρή αύξηση στη ζήτηση, μεγαλώνει καθώς μετακινούμαστε προς τα ανώτερα επίπεδα της αλυσίδας, με αποτέλεσμα να χρειάζονται μεγαλύτερα στοκ ασφαλείας, που συνεπάγονται μεγαλύτερα αποθηκευτικά κόστη. Επίσης, η είσοδος πιο δυναμικών εμπορικών σχέσεων μεταξύ εταίρων, η παγκοσμιοποίηση, η ευκολία συναλλαγών μέσω διαδικτύου, κάνουν τη χρήση ημιαυτόνομων και προσαρμοζόμενων πολιτικών ιδιαίτερα ελκυστική [Elm04]. Σε ένα τέτοιο περιβάλλον, οι τεχνολογίες πρακτόρων λογισμικού συνδυασμένες με τεχνικές τεχνητής νοημοσύνης και μηχανικής μάθησης, μπορούν να διαδραματίσουν πρωτεύοντα ρόλο σε διάφορες όψεις της εφοδιαστικής αλυσίδας, όπως στη διαχείριση πελατών και προμηθευτών, στην τιμολόγηση, στις προβλέψεις με

μόνο κριτήριο τη διασφάλιση της ασφάλειας και της αξιοπιστίας αυτών των λύσεων [ZZC05, SCAM07].

Για τη μελέτη τέτοιου είδους πρακτικών, αναπτύχθηκε ο διεθνής διαγωνισμός Trading Agent Competition (TAC) και συγκεκριμένα το παιχνίδι Supply Chain Management (SCM), το οποίο ευνοεί την ανάπτυξη πλήθους στρατηγικών και μεθοδολογιών από τους αυτόνομους πράκτορες που συμμετέχουν σε αυτό. Στα πλαίσια αυτού του διαγωνισμού αναπτύχθηκε ο πράκτορας Mertacor [CSKM08]. Στο κεφάλαιο αυτό παρουσιάζεται μία μεθοδολογία βελτίωσης της πολιτικής πωλήσεων του πράκτορα, μοντελοποιώντας το πρόβλημα ως πρόβλημα ενισχυτικής μάθησης και εφαρμόζοντας τη μεθοδολογία NEAR. Στόχος είναι η εξαγωγή της πολιτικής υπό τη μορφή ΔΗΚ με προσαρμοσμένη την αρχιτεκτονική και τις παραμέτρους στο υπό εξέταση πρόβλημα. Η αξιολόγηση αυτής της προσέγγισης γίνεται με δεδομένα του διαγωνισμού και με σύγκρισή της με μεθόδους μηχανικής μάθησης που χρησιμοποιούσε ο πράκτορας στο παρελθόν, αλλά και με άλλες μεθόδους όπως ευριστικές πολιτικές ή βελτιστοποίησης λήψης αποφάσεων με σμήνος σωματιδίων (particle swarm optimization).

7.1 Η πλατφόρμα προσομοίωσης TAC SCM

Στο σενάριο του διαγωνισμού TAC SCM [AS05, CAS+06] κάθε πράκτορας που συμμετέχει αντιπροσωπεύει έναν κατασκευαστή προσωπικών υπολογιστών, ο οποίος προμηθεύεται εξαρτήματα υπολογιστών, συναρμολογεί τις μονάδες στο εργοστάσιό του και τις πουλάει σε πελάτες. Η δυναμικότητα του κάθε εργοστασίου είναι περιορισμένη και ανέρχεται σε $2,000$ εικονικούς κύκλους ανά ημέρα. Έξι (6) τέτοιοι πράκτορες ανταγωνίζονται στην πώληση δεκαέξι (16) διαφορετικών τύπων προσωπικών υπολογιστών σε πελάτες. Κάθε τύπος έχει διαφορετική διαμόρφωση. Με τη σειρά της κάθε διαμόρφωση απαιτεί την ύπαρξη τεσσάρων βασικών εξαρτημάτων: 1) κεντρικής μονάδας υπολογισμών, 2) μητρικής πλακέτας 3) μνήμης και 4) σκληρού δίσκου. Συνολικά, τα διαφορετικά εξαρτήματα ανέρχονται σε δέκα (10), τα οποία ο πράκτορας προμηθεύεται στέλνοντας προσφορές σε προμηθευτές. Οι προμηθευτές έχουν και αυτοί περιορισμένη παραγωγή εξαρτημάτων και καθώς είναι και αυτοί οντότητες με στόχο τη μεγιστοποίηση του κέρδους, η διαθεσιμότητα σε εξαρτήματα δεν είναι δεδομένη την κάθε χρονική στιγμή. Οι εργασίες που καλούνται να φέρουν εις πέρας οι πράκτορες καθημερινά είναι να διαπραγματευτούν συμβόλαια με τους προμηθευτές, να μειοδοτήσουν σε προσφορές που ζητούν οι πελάτες, να διαχειριστούν τη γραμμή παραγωγής τους και να αποστείλουν στους πελάτες τις παραγγελίες. Κάθε μέρα οι πελάτες ζητούν προσφορές (Request-for-Quote - RFQ) για συγκεκριμένο αριθμό και τύπο υπολογιστών και οι πράκτορες μειοδοτούν ανάλογα με τη δυνατότητά τους να εκπληρώσουν την παραγγελία στο χρόνο που ορίζει ο πελάτης. Η τιμή μειοδοσίας δεν μπορεί να ξεπερνά μία μέγιστη τιμή (στην προσομοίωση επιλέγεται τυχαία μεταξύ 75% και 125% της βασικής τιμής του προϊόντος) που θέτει ο πελάτης (reserve price). Την επόμενη ημέρα ο πελάτης παραγγέλνει από τον πράκτορα που έδωσε τη χαμηλότερη τιμή. Ο πρά-

κτορας, με τη σειρά του, πρέπει να στείλει το προϊόν έγκαιρα, ενώ σε διαφορετική περίπτωση πληρώνει ένα πρόστιμο για τις επόμενες πέντε ημέρες καθυστέρησης και στη συνέχεια η παραγγελία ακυρώνεται. Νικητής είναι ο πράκτορας με το μεγαλύτερο τραπεζικό υπόλοιπο στο τέλος του παιχνιδιού. Το παιχνίδι διαρκεί 220 εικονικές ημέρες και κάθε ημέρα διαρκεί 15 δευτερόλεπτα πραγματικού χρόνου. Με τον τρόπο αυτό προσομοιώνονται 220 ημέρες σε 55 λεπτά της ώρας.

Ένας πράκτορας στον διαγωνισμό TAC SCM θα πρέπει: *"Να πουλάει σε όσο το δυνατόν υψηλότερη τιμή και να αγοράζει σε όσο το δυνατόν χαμηλότερη, διατηρώντας στο μέγιστο τη ρυθμαπόδοση τόσο στο εργοστάσιο όσο και στην αποθήκη και αποφεύγοντας τις αστοχίες στις παραδόσεις των υπολογιστών "* [CSM08]. Με την έννοια ρυθμαπόδοση (throughput) εννοείται υψηλός βαθμός χρησιμοποίησης του εργοστασίου αλλά και διατήρηση σταθερού ρυθμού προώθησης των προσωπικών υπολογιστών στην αγορά σε πλήρη αντιστοιχία με τον ρυθμό παραγωγής τους. Το παρόν κεφάλαιο εστιάζει στο πρόβλημα των πωλήσεων και της ρυθμαπόδοσης του πράκτορα για τη βελτίωση της απόδοσής του. Εξετάζεται διεξοδικά η βελτιστοποίηση της καθημερινής εργασίας του πράκτορα ώστε αυτός να επιλέγει τις προσφορές στις οποίες θα απαντήσει και την τιμή που θα προσφέρει, δεδομένου του μικρού χρόνου απόφασης (near real-time) και των πεπερασμένων πόρων που διαθέτει.

7.2 Μηχανισμός πλειοδοσίας

Ο μηχανισμός που αναπτύχθηκε βοηθάει τον πράκτορα να επιλέξει τα αιτήματα των πελατών στα οποία θα απαντήσει και να τιμολογήσει τις προσφορές που θα δώσει. Εκτός από τη μέγιστη τιμή του αιτήματος, ο πράκτορας δεσμεύεται και από την προσφορά που υπάρχει σε εξαρτήματα, αλλά και την παραγωγική δυνατότητα του εργοστασίου. Η παρούσα μεθοδολογία εστιάζει στη βελτιστοποίηση της πολιτικής πώλησης και παραγωγής του πράκτορα. Για το λόγο αυτό έχουν γίνει δύο βασικές παραδοχές: α) ο πράκτορας έχει ένα απεριόριστο στοκ εξαρτημάτων, αποσυνδέοντας έτσι από το πρόβλημα το κομμάτι των προμηθειών και β) όλοι οι αλγόριθμοι μειοδοτούν στα τρέχοντα ημερήσια αιτήματα των πελατών, χωρίς να κρατούν ελεύθερους πόρους για μελλοντικά, ίσως πιο προσοδοφόρα αιτήματα. Όσον αφορά την πρώτη παραδοχή, εάν ο πράκτορας βρίσκονταν σε κανονική διαγωνιστική λειτουργία θα έπρεπε να εξασφαλίζει τα απαιτούμενα εξαρτήματα στην ώρα τους, ενώ σχετικά με τη δεύτερη παραδοχή, σε διαγωνιστικό περιβάλλον, μπορούν να παράγονται εικονικά μελλοντικά αιτήματα [PS09] και ο αλγόριθμος να μειοδοτεί τόσο στα πραγματικά όσο και στα εικονικά αιτήματα, κρατώντας εμμέσως πόρους μέσω των εικονικών αιτημάτων.

Για τη δημιουργία του χαρτοφυλακίου αιτημάτων (RFQs) ο πράκτορας εφαρμόζει μια πρακτική μεγιστοποίησης της προσδοκώμενης ωφέλειας [PS09, CSKM08], κυρίως λόγω της αβεβαιότητας που προκαλεί η ανταγωνιστικότητα. Εφόσον δε λαμβάνονται υπόψη οι στρατηγικές προμηθειών, η μείωση του κόστους αγοράς αγνοείται και η ωφέλεια υπολογίζεται με βάση τον τζίρο (revenue) αντί του κέρ-

δους, χωρίς ωστόσο να επωφελείται ο ένας μηχανισμός έναντι του άλλου. Τα αιτήματα ταξινομούνται με φθίνουσα σειρά ανάλογα με το εκτιμώμενο κέρδος, κανονικοποιημένα βάσει των κύκλων κατασκευής του προϊόντος [BGG+04]:

$$\textit{Utility} = \frac{P(\textit{offer} = \textit{accepted}|\text{offer price}) \cdot \text{offer price}}{Cycles(Product)}. \tag{7.1}$$

Στον μηχανισμό που αναπτύχθηκε ακολουθήθηκε η προσέγγιση των *μερικών παραγγελιών*, όπου ο πράκτορας μπορεί να κερδίσει μέρος της παραγγελίας ανάλογο με την πιθανότητα αποδοχής της προσφοράς στη δεδομένη τιμή [PS09]:

$$\textit{PartialCycles} = P(\textit{offer} = \textit{accepted}|\text{offer price}) \cdot \textit{Quantity} \cdot \textit{Cycles}(\textit{Product}). \tag{7.2}$$

Οι πράκτορες δεν μπορούν να κερδίσουν μερικές παραγγελίες, ωστόσο η ιδέα είναι ότι η συνολική ποσότητα των κερδισμένων κύκλων θα προσεγγίζεται από το άθροισμα των μερικώς κερδισμένων κύκλων.

Οι επιλεγμένες αναζητήσεις προσφορών (Selected RFQs) στις οποίες ο πράκτορας θα κάνει προσφορές είναι το σύνολο:

$$\text{Selected RFQs} = \{x \in \textit{RFQBundle}| \sum PartialCycles(x) \leq 2000; \text{ and } Utility(x) > Utility(y), y \notin SelectedRFQs\}. \tag{7.3}$$

Ο Αλγόριθμος 7.1 συνοψίζει την παραπάνω μεθοδολογία. Οι τιμές προσφοράς καθορίζονται από τον πράκτορα και μπορούν να προκύψουν με διάφορους τρόπους, όπως με πρόβλεψη από παλιά δεδομένα ή μέσω κάποιας διαδικασίας βελτιστοποίησης.

Αλγόριθμος 7.1 Μεγιστοποίηση κέρδους δεδομένων των τιμών προσφοράς.

PROFIT-MAXIMIZATION

1: For each RFQ select a price to bid on offer price
2: For each RFQ bid calculate the probability of acceptance $Pr(accepted|bid)$
3: For each RFQ calculate its estimated utility
4: Sort RFQ Bundle based on utility
5: $CyclesRemaining \leftarrow 2000$
6: **while** $CyclesRemaining > 0$ **do**
7: $RFQ \leftarrow$ RFQBundle.next()
8: $CyclesRemaining- = RFQ.quantity \cdot RFQ.prob \cdot RFQ.cycles$
9: OfferBundle.add(RFQ)
10: **end while**
11: Send offers

7.2.1 Εκτίμηση πιθανότητας αποδοχής

Για το πρόβλημα της εκτίμησης της πιθανότητας αποδοχής μίας προσφοράς δεδομένης της τιμής μειοδοσίας, $Pr(accepted|\text{offer price})$, υπάρχουν αρκετές προσεγγίσεις. Στα [BGG+04, PS05] χρησιμοποιείται η μέθοδος της παλινδρόμησης για

τη συσχέτιση της τιμής με την πιθανότητα, στο [PS09] χρησιμοποιούνται φίλτρα σωματιδίων, ενώ είναι δυνατή και η χρήση ευριστικών μεθόδων για το πρόβλημα [SSF06] ή η εφαρμογή σιγμοειδών καμπυλών [CS09]. Το πρόβλημα μπορεί να λυθεί και με πιο περίπλοκα σχήματα όπως είναι ο συνδυασμός δικτύων ακτινικών συναρτήσεων βάσης (radial basis function networks), τα οικονομικά καθεστώτα και οι μηχανισμοί συσχέτισης που προσαρμόζονται δυναμικά [HKvD+09].

Χρησιμοποιήθηκε η λογιστική παλινδρόμηση ως αλγόριθμος εκμάθησης σιγμοειδών καμπυλών των οποίων η παραμετρική μορφή επιτρέπει την προσαρμογή ανάλογα με τις ιδιότητες των αιτημάτων προσφορών και των καταστάσεων της αγοράς, ως χαρακτηριστικά. Η πιθανότητα επιλογής δίνεται από την εξίσωση:

$$f(z) = \frac{1}{1 + exp^{-z}}, \tag{7.4}$$

με

$$z = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n, \tag{7.5}$$

όπου x_i συμβολίζει το κάθε χαρακτηριστικό και w_i το αντίστοιχο βάρος. Τα βέλτιστα βάρη μπορούν να βρεθούν μέσα από ρουτίνες βελτιστοποίησης. Εάν χρησιμοποιηθεί η μέθοδος κατάβασης πλαγιάς, το μοντέλο της λογιστικής παλινδρόμησης έχει το πλεονέκτημα του χειρισμού μεγάλου αριθμού δεδομένων και της δυναμικής προσαρμογής μέσω επαναληπτικής διαδικασίας ανανέωσης βαρών.

Εννέα χαρακτηριστικά χρησιμοποιήθηκαν για την παραμετροποίηση της λογιστικής καμπύλης. Τα χαρακτηριστικά μπορεί να είναι είτε ιδιότητες της αίτησης προσφοράς, είτε να αφορούν την κατάσταση της αγοράς. Τα χαρακτηριστικά με βάση τις ιδιότητες της αίτησης προσφοράς είναι:

1. τρέχουσα ημέρα *current date* (CD),
2. ημέρα προθεσμίας *due date* (DD),
3. μέγιστη τιμή *reserve price* (RP),
4. ποσότητα *quantity* (QNT) και
5. τιμή βάσης προϊόντος *product base price* (BP).

Τα χαρακτηριστικά με βάση τα στοιχεία της αγοράς είναι:

6. 5-ήμερος μέσος όρος μέγιστης τιμής πώλησης του προϊόντος *5-day average maximum selling product price* (AMAX),
7. 5-ήμερος μέσος όρος μικρότερης τιμής πώλησης του προϊόντος της αίτησης *5-day average minimum selling product price* (AMIN) και
8. συνολική ποσότητα ζητούμενων προϊόντων από τους πελάτες την τρέχουσα μέρα *total quantity of customer requested products* (TQNT).

Τέλος, υπάρχει και ένα τελευταίο χαρακτηριστικό:

9. η τιμή της προσφοράς *offer price* (OP), που υπολογίζεται στο βήμα 1 του Αλγορίθμου 7.1 και αναλύεται διεξοδικά στην επόμενη ενότητα.

Η μεταβλητή στόχος που θέλουμε να προβλέψουμε, είναι εάν η προσφορά στην τιμή που ορίζεται από το χαρακτηριστικό OP, έγινε δεκτή (1) ή όχι (0). Τα χαρακτηριστικά κανονικοποιήθηκαν βάσει της Εξίσωσης 7.6. Οι μέγιστες και ελάχιστες τιμές προέκυψαν για ορισμένα χαρακτηριστικά από τις προδιαγραφές του διαγωνισμού, ενώ για άλλα από το σετ δεδομένων της εκπαίδευσης, όπου λόγω του τεράστιου όγκου δεδομένων είναι δυνατή μια καλή εκτίμησή τους.

$$x' = \frac{x - \bar{x}}{range(x)} = \frac{x - \bar{x}}{max(x) - min(x)} \tag{7.6}$$

Ο τελικός υπολογισμός της πιθανότητας δίνεται από την Εξίσωση 7.7:

$$Pr(accepted|\text{offer price}) = \begin{cases} f & \text{if offer price} <= \text{reserve price,} \\ 0 & \text{if offer price} > \text{reserve price.} \end{cases} \tag{7.7}$$

όπου reserve price η μέγιστη τιμή που θέτει ο πελάτης στο RFQ.

7.2.2 Εκτίμηση τιμής προσφοράς

Το δεύτερο τμήμα του υπό ανάπτυξη μηχανισμού είναι υπεύθυνο για την εκτίμηση της καλύτερης τιμής προσφοράς. Αναφέρεται στο βήμα 1 του αλγορίθμου `PROFIT-MAXIMIZATION` (Αλγόριθμος 7.1). Το πρόβλημα μπορεί να προσεγγιστεί είτε εφαρμόζοντας στοχαστική βελτιστοποίηση [BGNT04] είτε άπληστη αναζήτηση [PS05]. Και στις δύο περιπτώσεις η κεντρική ιδέα είναι ότι δεδομένης μίας εκτίμησης της πιθανότητας αποδοχής, αναζητούνται τιμές που θα βελτιστοποιήσουν το πακέτο προσφορών, σύμφωνα με τους περιορισμούς. Η πρώτη προσέγγιση απαιτεί εξειδικευμένο λογισμικό και μεγάλη υπολογιστική ισχύ για την παραγωγή λύσεων σε σχεδόν πραγματικό χρόνο, οπότε ακολουθήθηκε η δεύτερη προσέγγιση.

Πρόβλεψη Μειοδοτούσας Τιμής

Μια πρώτη προσέγγιση είναι να εφαρμοστούν τεχνικές μηχανικής μάθησης σε ιστορικά δεδομένα για την πρόβλεψη της μειοδοτούσας τιμής. Στη συγκεκριμένη περίπτωση χρησιμοποιήθηκαν δένδρα ταξινόμησης και παλινδρόμησης (classification and regression trees or CART) [BFSO84]) ή δένδρα M5' [WW97]. Χρησιμοποιώντας ως μεταβλητές πρόβλεψης τα χαρακτηριστικά 1 έως 8 σύμφωνα με την Ενότητα 7.2.1, στόχος είναι η πρόβλεψη της τιμής που θα μετατρέψει μια προσφορά σε μία παραγγελία.

Στο παρελθόν έχουν χρησιμοποιηθεί παρόμοιες μέθοδοι όπως M5' [CSKM08] και k-Nearest Neighbors [KMJ+09]. Το βασικό τους μειονέκτημα είναι ότι τα μοντέλα δημιουργούνται από ιστορικά δεδομένα με πιθανόν διαφορετικούς πράκτορες ή σετ πρακτόρων σε σχέση με αυτούς που θα συναντήσει ο πράκτορας. Αν και

τα χαρακτηριστικά που σχετίζονται με την κατάσταση στην αγορά περιέχουν αρκετή πληροφορία για τη βοήθεια της γενίκευσης, η μη-στασιμότητα (non-stationarity) της διαδικασίας μπορεί να οδηγήσει σε μειωμένη ικανότητα πρόβλεψης. Για το λόγο αυτό οι προβλέψεις πολλές φορές βελτιώνονται και κατά τη λειτουργία του πράκτορα [KMJ+09, HKvD+09].

Βελτιστοποίηση με σμήνος σωματιδίων

Μία ακόμη μέθοδος που εφαρμόστηκε στο πρόβλημα είναι η βελτιστοποίηση με σμήνος σωματιδίων (particle swarm optimization - PSO) [KE95]. Με τη μέθοδο PSO κάθε σωματίδιο αντιπροσωπεύει μία λύση στο πρόβλημα, δηλαδή είναι ένα διάνυσμα τιμών για κάθε προσφορά. Η αρχικοποίηση των διανυσμάτων γίνεται τυχαία στο διάστημα $[0.9p, 1.1p]$, όπου p είναι η πρόβλεψη που προήλθε από ένα εκπαιδευμένο δένδρο με τον αλγόριθμο M5'. Εκτός του διανύσματος τιμών $\tilde{x}$, κάθε σωματίδιο περιέχει και ένα διάνυσμα ταχυτήτων $\tilde{v}$, καθώς και μια τιμή επίδοσης f. Τα σωματίδια διαθέτουν και μνήμη για τη διατήρηση της καλύτερης επίδοσης, P_{bst}, που έχουν επιτύχει από την έναρξη του αλγορίθμου. Επίσης, κάθε σωματίδιο έχει και γνώση της επίδοσης, G_{bst}, του καταλληλότερου σωματιδίου στη γειτονιά του, όπως αυτή ορίζεται από κάποια μετρική απόστασης (π.χ. Ευκλείδεια) μεταξύ των διανυσμάτων που περιέχουν τις λύσεις. Στη συγκεκριμένη περίπτωση γειτονιά ορίστηκε όλος ο πληθυσμός.

Σε κάθε επανάληψη τα σωματίδια ανανεώνουν τη "θέση" τους $\tilde{x}$, βάσει της "ταχύτητας" που διαθέτουν, ενώ οι ταχύτητες ανανεώνονται ανάλογα με τα διανύσματα P_{bst} και G_{bst}. Σύμφωνα με την πολιτική αυτή, τα σωματίδια οδηγούνται προς τις περιοχές των καταλληλότερων σωματιδίων και ταυτόχρονα προς την καταλληλότερη περιοχή που έχουν επισκεφθεί. Οι εξισώσεις μεταβολής της κατάστασης των σωματιδίων περιγράφονται από τις εξισώσεις:

$$\tilde{v}_t = \chi(\tilde{v}_{t-1} + r_1\phi_1(P_{bst} - \tilde{x}_{t-1}) + r_2\phi_2(G_{bst} - \tilde{x}_{t-1})) \tag{7.8}$$

$$\tilde{x}_t = \tilde{x}_{t-1} + \tilde{v}_t \tag{7.9}$$

όπου χ είναι ο συντελεστής αδράνειας, ϕ_1 και ϕ_2 είναι οι σταθερές επιτάχυνσης και τα r_1, r_2 είναι τυχαία επιλεγμένες σταθερές στο διάστημα $U(0, 1)$. Ο συντελεστής αδράνειας ορίζει πόσο θα αντιδρά το σωματίδιο στις μεταβολές της ταχύτητας, ενώ οι σταθερές επιτάχυνσης ορίζουν τον παράγοντα με βάση τον οποίο τα σωματίδια πηγαίνουν προς τις θέσεις P_{bst} και G_{bst}. Οι τυχαίες μεταβλητές χρησιμοποιούνται για να εισάγουν στοχαστικότητα στη μετάβαση, η οποία χρησιμεύει στην απώθηση των σωματιδίων από το σημείο ισορροπίας μεταξύ των σημείων P_{bst} και G_{bst}, τα οποία δεν είναι κατ' ανάγκη βέλτιστα [KE95]. Τεχνικές σμήνους σωματιδίων έχουν ένα πλεονέκτημα έναντι των γενετικών αλγορίθμων λόγω της επιδεξιότητάς τους με πραγματικούς αριθμούς.

Μειοδοσία με έλεγχο

Μια εναλλακτική προσέγγιση στο πρόβλημα είναι η μοντελοποίησή του ως πρόβλημα ελέγχου. Σκοπός είναι η μεταβολή των τιμών προσφοράς, ώστε μια μετρική να συγκρατείται κοντά στη βέλτιστη τιμή. Για παράδειγμα, μια τέτοια μετρική θα μπορούσε να είναι η διατήρηση της χρησιμοποίησης της παραγωγικής ικανότητας του εργοστασίου στο 100%, χωρίς αργοπορημένες ή ακυρωμένες παραγγελίες. Εάν οι προτεινόμενες τιμές είναι υψηλές, τότε το εργοστάσιο θα υπολειτουργεί ως συνέπεια της απώλειας παραγγελιών. Αντίθετα, εάν οι τιμές είναι χαμηλές, το εργοστάσιο θα λειτουργεί πλήρως, αλλά μπορεί να δημιουργούνται καθυστερήσεις με συνέπεια αργοπορημένες και ακυρωμένες παραγγελίες με ό,τι αυτό συνεπάγεται για τα κέρδη του πράκτορα. Προηγούμενες προσεγγίσεις με έλεγχο στο περιβάλλον TAC SCM αφορούσαν διανεμημένο έλεγχο με ανάδραση, ασαφή λογική και ευριστικό έλεγχο [KPS+04, HRLJ06, SSF06].

Στην παρούσα προσέγγιση, χρησιμοποιήθηκε ένας ευριστικός μηχανισμός του `PhantAgent` [SSF06], νικητή του διαγωνισμού το 2007, και τέθηκε ως πρόβλημα ενισχυτικής μάθησης. Ο στόχος ήταν η εύρεση μιας πολιτικής η οποία θα βελτιστοποιεί το κομμάτι των πωλήσεων και των κατασκευών του πράκτορα μέσω προσαρμοζόμενων συναρτήσεων προσέγγισης. Ο ευριστικός ελεγκτής του πράκτορα `PhantAgent` παρουσιάζεται στον Αλγόριθμο 7.2. Ανάλογα με τον αριθμό των παραγγελιών που θα δεχθεί ο πράκτορας, τη χρήση του εργοστασίου και τις παραγγελίες που αναμένουν προς παρασκευή και παράδοση, ο πράκτορας `PhantAgent` μεταβάλλει έναν παράγοντα f που προσαρμόζει τη μέγιστη τιμή πώλησης στην αγορά τις τρεις τελευταίες μέρες για το συγκεκριμένο προϊόν στη ζήτηση της προσφοράς. Πιο συγκεκριμένα, ο αλγόριθμος παρακολουθεί διάφορες μεταβλητές:

- τον παράγοντα f με αρχική τιμή 1,
- τη μεταβλητή won, που αποθηκεύει τον συνολικό αριθμό των κύκλων που απαιτούνται για την παρασκευή όλων τον παραγγελιών που κερδήθηκαν την τελευταία μέρα,
- τη μεταβλητή cap, η οποία είναι μία σταθερά της χωρητικότητας του εργοστασίου για μία μέρα,
- τη μεταβλητή $price$, που αποθηκεύει την τελική τιμή της προσφοράς,
- τη μεταβλητή $high$, που υποδεικνύει την υψηλότερη τιμή πώλησης του συγκεκριμένου προϊόντος τις τρεις τελευταίες μέρες και
- τη μεταβλητή $pqueue$, που περιέχει τον συνολικό αριθμό κύκλων που χρειάζονται για να παρασκευαστούν παλαιότερα κερδισμένες παραγγελίες, οι οποίες ακόμα δεν έχουν παραδοθεί.

Θεωρήθηκε ότι ο αλγόριθμος περιορίζεται από τη μη βελτιστοποιημένη προσαρμογή του παράγοντα f και έτσι περισσότερα οφέλη μπορούν να επιτευχθούν.

Αλγόριθμος 7.2 Ο ευριστικός αλγόριθμος ελέγχου του PhantAgent.

OFFER-PRICE-CONTROL

1: $f = f + \frac{(won - cap)}{(cap * 5)}$
2: **if** $pqueue > (2 * cap)$ **then**
3: $\quad f \leftarrow f + 0.01$
4: **end if**
5: **if** $pqueue < cap$ **then**
6: $\quad f \leftarrow f - 0.005$
7: **end if**
8: **if** $f < 0.9$ **then**
9: $\quad f \leftarrow 0.9$
10: **end if**
11: **if** $f > 1.05$ **then**
12: $\quad f \leftarrow 1.05$
13: **end if**
14: $price \leftarrow high \cdot f$

Ο Αλγόριθμος 7.2, ελαφρώς προσαρμοσμένος, μπορεί να τεθεί και ως πρόβλημα ελέγχου ενισχυτικής μάθησης. Η Μαρκοβιανή διαδικασία απόφασης που καλείται να επιλύσει ο αλγόριθμος ενισχυτικής μάθησης είναι η εξής:

$$S = \{\textit{won cycles}/capacity, \textit{queued cycles}/capacity, \textit{total quantity}/3500\} \tag{7.10}$$

$$A = \{0.9, 0.91, 0.92, ..., 1.04, 1.05\}, |A| = 16 \tag{7.11}$$

$$r = -|\frac{won - capacity}{capacity}| - |\frac{queue - capacity}{capacity}| \tag{7.12}$$

Η συνάρτηση ανταμοιβής σχεδιάστηκε έτσι ώστε να τιμωρεί στρατηγικές που υπερβαίνουν το όριο των 2000 κύκλων στις κερδισμένες παραγγελίες κάθε μέρα και τους 2000 κύκλους παραγγελιών που είναι στην ουρά προς κατασκευή στο εργοστάσιο. Η μεταβλητή *total quantity* κανονικοποιήθηκε μεταξύ 0 και 1 χρησιμοποιώντας το ανώτερο όριο των 3500 υπολογιστών προς ζήτηση σε μία μέρα από τους πελάτες. Το άνω όριο εκτιμήθηκε από τις προδιαγραφές του παιχνιδιού και από ιστορικά δεδομένα.

7.3 Πειράματα και αποτελέσματα

Τα πειράματα διεξήχθησαν χρησιμοποιώντας ιστορικά δεδομένα από τους τελικούς του διαγωνισμού TAC SCM 2011. Τα παιχνίδια των τελικών ήταν δεκαέξι (16) και των ημιτελικών είκοσι (20). Όλα τα μοντέλα (λογιστική παλινδρόμηση, CART, M5' και NEAR) εκπαιδεύτηκαν στα δεδομένα των ημιτελικών και εξετάστηκαν στα δεδομένα των τελικών. Τα μοντέλα επιβλεπόμενης μάθησης εκπαι-

Πίνακας 7.1: Τα βάρη της προσαρμοσμένης λογιστικής καμπύλης. Το βασικότερο χαρακτηριστικό είναι η τιμή της προσφοράς (OP) η οποία μαζί με το δεύτερο σε τάξη χαρακτηριστικό, τη μέση μέγιστη τιμή πώλησης (AMAX) έχουν αρνητική επίδραση στην πιθανότητα επιτυχούς προσφοράς όσο υψηλότερη είναι η τιμή τους. Τρίτη πιο σημαντική επιρροή ασκεί η οριακή τιμή (RP). Η σημαντικότητα ορίζεται από τη συνεισφορά τους στο τελικό άθροισμα και είναι συγκρίσιμη λόγω της κανονικοποίησης που έχει γίνει πριν την εκπαίδευση.

CD	BP	DD	QNT	AMAX	AMIN	TQNT	RP	OP
-0.19	-0.67	-1.26	0.80	17.96	1.61	0.52	2.39	-22.94

δεύτηκαν με το 40% των συνολικών δεδομένων, δηλαδή με $952, 953$ (συγκαταλέγονται τόσο οι προσφορές που έγιναν αποδεκτές, όσο και αυτές που δεν έγιναν) περιπτώσεις για τη λογιστική παλινδρόμηση και $312, 891$ για τα δενδρικά μοντέλα παλινδρόμησης (περιλαμβάνει μόνο τις προσφορές που έγιναν αποδεκτές). Για την υλοποίηση των CART και M5' χρησιμοποιήθηκαν το πακέτο της πλατφόρμας R `rpart` [R D11, TpbBR11] και η βιβλιοθήκη WEKA [HFH+09] αντίστοιχα. Για την εξέταση και τη σύγκριση των αλγορίθμων πραγματοποιήθηκαν προσομοιώσεις, χρησιμοποιώντας αναζητήσεις προσφοράς από πελάτες στις οποίες είχαμε το τελικό αποτέλεσμα, δηλαδή έγινε τουλάχιστον μια προσφορά.

Αρχικά εξετάστηκε η λογιστική παλινδρόμηση. Τα βάρη των χαρακτηριστικών παρουσιάζονται στον Πίνακα 7.1. Το Σχήμα 7.1 παρουσιάζει την προσαρμοσμένη λογιστική καμπύλη για μια συγκεκριμένη αίτηση προσφοράς σε μια ευρεία διακύμανση της τιμής προσφοράς στο διάστημα $[0, 1.25 \cdot basePrice]$, για διαφορετικές τιμές του ορίου αποδεκτής τιμής (Σχήμα 7.1(a)) και της ημέρας παράδοσης (Σχήμα 7.1(b)). Μπορεί εύκολα κάποιος να διαπιστώσει την επίδρασή τους στην καμπύλη. Μια ποσοτική μετρική της απόδοσης της καμπύλης λογιστικής παλινδρόμησης είναι η πρόβλεψη εάν μία παραγγελία θα γίνει αποδεκτή ($Pr \geq 0.5$) ή όχι ($Pr < 0.5$). Έαν πάρουμε τις τιμές που προβλέπει το μοντέλο CART σε ένα σετ δεδομένων εξέτασης και έχουμε την προσαρμοσμένη λογιστική για την πρόβλεψη της πιθανότητας αποδοχής, το αποτέλεσμα είναι 72.6% ακρίβεια στην ταξινόμηση (classification accuracy).

Για τη σύγκριση μεταξύ των στρατηγικών πώλησης ανάμεσα στις διαφορετικές τεχνικές βελτιστοποίησης που δοκιμάσαμε χρησιμοποιήθηκε η μετρική του προσαρμοσμένου συνολικού τζίρου (adjusted total revenue - ATR).

$$\text{ATR} = \text{total revenue} - max(\overline{\text{daily cycles}} - 2000, 0) \cdot \overline{\text{lost cycle cost}}. \quad (7.13)$$

Η Εξίσωση 7.13 προσαρμόζει το συνολικό τζίρο (total revenue) ώστε να συμπεριλαμβάνει και κόστη που προκύπτουν από την υπέρβαση του ορίου των ημερήσιων κύκλων εργασίας του εργοστασίου ($\overline{\text{daily cycles}}$) των παραγγελιών που κερδήθηκαν και βρίσκονται στην ουρά πάνω από το όριο χωρητικότητας των $2,000$ κύκλων. Για να ληφθούν υπόψιν οι έξτρα κύκλοι και να γίνουν τα αποτελέσματα συγκρίσιμα, πρέπει να υπολογιστεί το κόστος επιβάρυνσης κάθε έξτρα κύκλου πλέον των

$2,000$ κατά μέσο όρο ($\overline{\text{lost cycle cost}}$) για τον πράκτορα. Με τον τρόπο αυτό, το κόστος θα αφαιρεθεί από το συνολικό τζίρο.

Σύμφωνα με τις προδιαγραφές του παιχνιδιού, κάθε αίτημα προσφοράς έχει κατά μέσο όρο ποσότητα 10 υπολογιστών με μέσο όρο 5.5 κύκλους εργοστασίου για κάθε έναν υπολογιστή, δηλαδή σύνολο 55 κύκλους. Η μέση τιμή των 10 υπολογιστών είναι 10 φορές η μέση βασική τιμή ($2,000$ χρηματικές μονάδες), οπότε και υπολογίζεται στις $20,000$. Οι κυρώσεις για τις καθυστερημένες παραγγελίες ανέρχονται κατά μέσο όρο στο 10% της τιμής βάσης, οπότε το κόστος που αφαιρούμε από τον τζίρο ανέρχεται στις $2,0000 \cdot 1.1 = 22,000$ για 55 κύκλους ή σε 400 ανά κύκλο. Εάν αυτό συμβαίνει για 216 ημέρες παιχνιδιού, τότε το ποσό προς αφαίρεση για κάθε έξτρα κύκλο είναι κατά μέσο όρο $86,400$. Για παράδειγμα αν έχουμε έξτρα 100 κύκλους κατά μέσο όρο, τότε από το συνολικό τζίρο θα αφαιρεθούν $86,400 \cdot 100 = 8,640,000$ χρηματικές μονάδες.

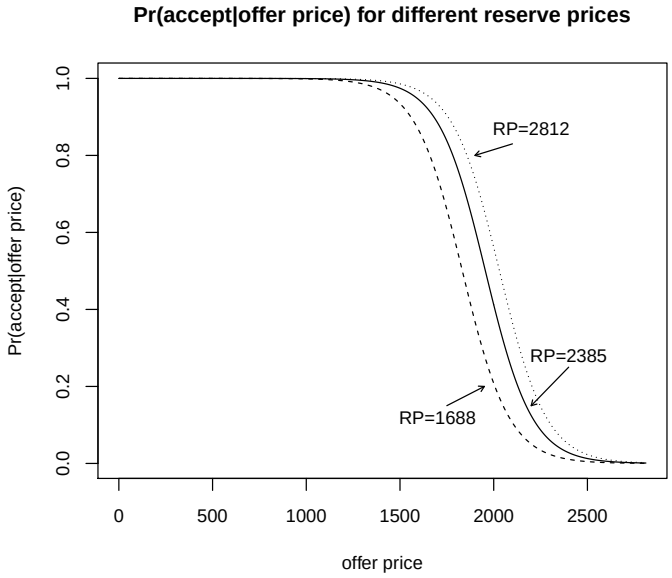
Η μέθοδος NEAR εκτελέστηκε με έναν πληθυσμό από 100 δίκτυα, για 100 γενιές. Η επίδοση του δικτύου πρωταθλητή σε όρους μέσης συνολικής ανταμοιβής φαίνεται στο Σχήμα 7.2 και είναι ενδεικτικό της εξελικτικής πίεσης που ασκεί ο αλγόριθμος στην εξεύρεση ικανών δικτύων για την αναπαράσταση της πολιτικής του πράκτορα.

Για το σμήνος σωματιδίων χρησιμοποιήθηκαν 100 σωματίδια, ενώ πραγματοποιήθηκαν 100 επαναλήψεις. Σε έναν μέσο υπολογιστή χρειάζεται περίπου 1 δευτερόλεπτο για να εκτελεστεί ο αλγόριθμος. Οπότε, αυτή η διαμόρφωση επιτρέπει να υπάρχει χρόνος για τη δημιουργία εικονικών αιτήσεων προσφορών ώστε να γίνει η βελτιστοποίηση συνολικά για 10 ημέρες, όταν ο πράκτορας διαγωνίζεται. Οι παράμετροι χ και $\phi_{1,2}$ τέθηκαν ίσες με 0.1 και 0.5, αντίστοιχα.

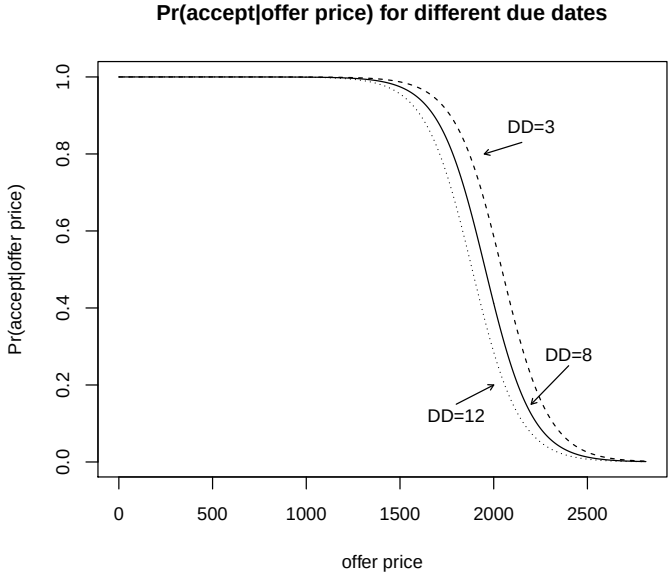
Ο Πίνακας 7.2 και ο Πίνακας 7.3 συνοψίζουν τον προσαρμοσμένο συνολικό τζίρο (ATR) και τον μέσο όρο κύκλων εργοστασίου (average daily cycles - ADC) που χρησιμοποιήθηκαν από τον πράκτορα προκειμένου να κατασκευαστούν κερδισμένες παραγγελίες σε κάθε ένα από τα παιχνίδια που προσομοιώθηκαν από τα δεδομένα των τελικών του 2011.

Βάσει των οδηγιών στο [Dem06] εφαρμόστηκε το στατιστικό τεστ Wilcoxon signed-ranks test [Wil45] για τη στατιστική σημαντικότητα ανά δύο, ανάμεσα στα αποτελέσματα της μεθόδου ενισχυτικής μάθησης και NEAR και των άλλων μεθόδων.Υπολογίστηκαν το στατιστικό z για το οποίο ισχύει: εάν $z < -1.96$ μπορούμε να απορρίψουμε τη μηδενική υπόθεση ότι οι δύο ανταγωνιστές έχουν κατά μέσο όρο την ίδια συμπεριφορά για $\alpha = 0.05$. Οι τιμές του z ήταν $z = -2.32$, $z = -2.53$, $z = -2.63$ και $z = -3.51$ κατά τη σύγκριση της NEAR με τους CART, M5', PSO και τον αλγόριθμο ευριστικού ελέγχου, δίνοντας στατιστική σημαντικότητα στην υπεροχή τόσο της μοντελοποίησης με ενισχυτική μάθηση όσο και στην εφαρμογή της NEAR. Επίσης, χρησιμοποιήθηκε και το τεστ Friedman για να αντιμετωπίσουμε τη μηδενική υπόθεση ότι όλοι οι αλγόριθμοι παρουσιάζουν κατά μέσο όρο την ίδια απόδοση, που απορρίφθηκε με τιμή χ_F^2, 19.95.

Εκτός από τις μεθόδους ευριστικού ελέγχου και ενισχυτικής μάθησης, οι υπόλοιπες μέθοδοι συνεχώς υποτιμούν τις προσφορές τους με αποτέλεσμα να κερδίζουν περισσότερες παραγγελίες και να έχουν λειτουργία στο εργοστάσιό τους



(a) Διαφορετικές τιμές μέγιστων ορίων προσφορών (reserve prices or RP).



(b) Διαφορετικές μέρες παράδοσης (due dates or DD).

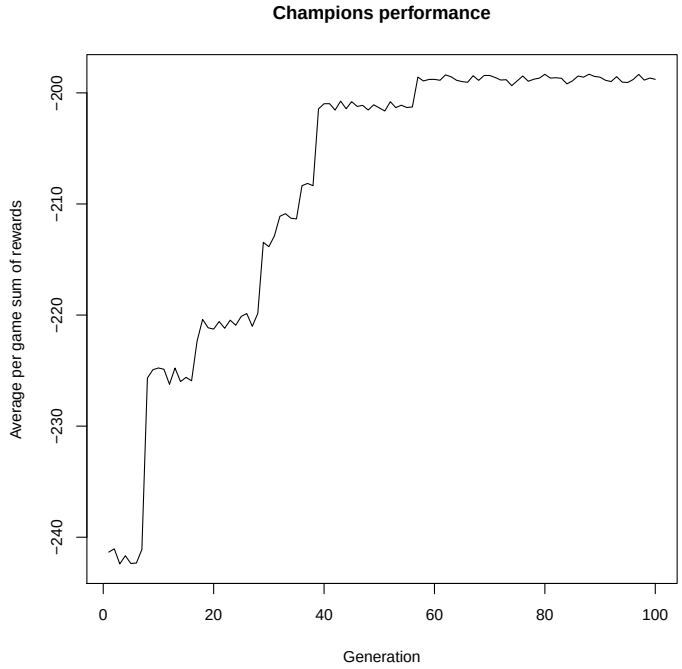
Σχήμα 7.1: Καμπύλες πιθανοτήτων αποδοχής δεδομένων προσφορών. Τα δύο σχήματα αναπαριστούν την πιθανότητα αποδοχής μίας προσφοράς (άξονας y) δεδομένης της τιμής προσφοράς (άξονας χ) για διαφορετικές τιμές χαρακτηριστικών που επηρεάζουν την καμπύλη, όπως της οριακής τιμής και των ημερών παράδοσης. Η αναπαράσταση της πιθανότητας αποδοχής ως μια λογιστική (σιγμοειδή) καμπύλη είναι κοντά στην πραγματικότητα, αφού οι τιμές κοντά στην οριακή τιμή δεν έχουν πολλές πιθανότητες αποδοχής σε ένα ανταγωνιστικό περιβάλλον, ενώ οι πολύ χαμηλές τιμές είναι πολύ πιο σίγουρο ότι θα γίνουν αποδεκτές. Η απότομη πλαγιά βρίσκεται στο σημείο κατάστασης της αγοράς.

Πίνακας 7.2: Προσαρμοσμένος συνολικός τζίρος (ATR) για όλα τα παιχνίδια των τελικών. Το ATR είναι σε εκατομμύρια.

Game ID	CART	M5'	Heuristic	PSO	RL
4584	119.9	121.1	124.9	120.7	**129.2**
4585	**126.2**	123.9	121.5	123.5	125.6
4586	123.0	122.4	121.8	121.4	**125.7**
4587	**132.0**	125.6	125.4	125.4	128.9
4588	123.4	126.1	125.0	124.8	**129.4**
4589	123.6	123.4	124.0	122.9	**130.0**
4590	108.2	108.9	104.5	108.0	**109.6**
4591	**135.0**	132.1	124.5	132.9	128.3
4360	137.6	132.0	141.4	135.3	**145.1**
4361	120.8	119.4	119.4	120.3	**123.6**
4362	112.7	111.0	112.9	110.9	**116.6**
4363	97.8	110.3	109.5	108.4	**111.8**
4364	116.0	**119.1**	107.8	117.6	113.1
4365	120.4	120.2	124.8	120.2	**127.1**
4366	118.9	118.3	118.9	119.2	**123.3**
4367	97.1	102.9	105.9	101.3	**108.7**
Average	119.5	119.8	119.5	119.6	**123.5**

Πίνακας 7.3: Μέσος όρος ημερήσιων κύκλων (ADC) για όλα τα παιχνίδια των τελικών.

Game ID	CART	M5'	Heuristic	PSO	RL
4584	2338	2256	1882	2287	1984
4585	2174	2157	1909	2181	1971
4586	2259	2130	1915	2134	1994
4587	2321	2294	1930	2322	1995
4588	2324	2252	1908	2302	1992
4589	2289	2251	1906	2258	1996
4590	2239	2089	1827	2141	1910
4591	2084	2003	1831	2059	1906
4360	2234	2206	1921	2181	1990
4361	2184	2116	1862	2113	1944
4362	2290	2088	1905	2130	1979
4363	2349	2054	1920	2107	1980
4364	2137	2100	1799	2143	1907
4365	2218	2120	1918	2125	1983
4366	2244	2049	1915	2098	1998
4367	2300	2102	1891	2155	1973
Average	2249	2141	1889	2171	1968



Σχήμα 7.2: Το μέσο άθροισμα ανταμοιβών ανά παιχνίδι (επεισόδιο) στις 216 ημέρες της προσομοίωσης από το δίκτυο πρωταθλητή σε κάθε γενιά. Η εξελικτική πίεση οδηγεί την αναζήτηση ολοένα και καλύτερων πολιτικών μέσω περισσότερο αποδοτικών δικτύων.

που ξεπερνά τους $2,000$ κύκλους ημερησίως κατά μέσο όρο. Αυτή η συμπεριφορά στην αρχή φαντάζει περίεργη καθώς στα τελικά του διαγωνισμού οι πράκτορες είναι περισσότερο ανταγωνιστικοί, ενώ από την άλλη στα πειράματα οι μέθοδοι, παρόλο που έχουν εκπαιδευτεί σε λιγότερο ανταγωνιστικά παιχνίδια, κερδίζουν πολλές προσφορές. Ωστόσο αυτό οφείλεται στη συγκεκριμένη διαμόρφωση των πρακτόρων της συγκεκριμένης χρονιάς που οδήγησαν σε ένα λιγότερο ανταγωνιστικό περιβάλλον στους τελικούς, κάτι που γίνεται εμφανές από την απόσταση των αποτελεσμάτων μεταξύ της 1ης και της 6ης θέσης, η οποία ενώ για τις χρονιές 2009 και 2010 είναι περίπου $12M$ και $13M$ εκατομμύρια αντίστοιχα, τη συγκεκριμένη χρονιά του 2011 εκτοξεύεται στα $29M$ [1].

Οι μέθοδοι ελέγχου, και ειδικά η ενισχυτική μάθηση, δουλεύουν με κύριο μοχλό την ανάδραση από το περιβάλλον δίνοντας καλύτερες δυνατότητες ευρωστίας και γενίκευσης. Θα μπορούσε να πει κάποιος ότι η μέθοδος PSO θα ήταν αποδοτική, αφού εξετάζει πολλές λύσεις και διαλέγει την καλύτερη από αυτές κάθε στιγμή. Το γεγονός ότι τα πειραματικά στοιχεία δεν έδειξαν κάτι τέτοιο ίσως να οφείλεται

[1] Τα αποτελέσματα βρίσκονται στην ιστοσελίδα http://tac.cs.umn.edu/. Η τελευταία πρόσβαση στην ιστοσελίδα έγινε στις 6 Σεπτεμβρίου 2012.

στην ανάγκη ύπαρξης καλύτερης συνάρτησης πιθανότητας αποδοχής στην οποία η μέθοδος PSO βασίζεται σε μεγάλο βαθμό.

7.4 Σύνοψη

Στο κεφάλαιο αυτό παρουσιάστηκε η σχεδίαση ενός μηχανισμού στα πλαίσια του TAC SCM. Ο μηχανισμός αποτελείται από τρία τμήματα: το τμήμα της επιλογής της προσφοράς, το τμήμα εκτίμησης της συνάρτησης πιθανότητας αποδοχής μίας προσφοράς και το τμήμα της διαδικασίας επιλογής των αιτημάτων που θα μειοδοτήσει ο πράκτορας. Με το διαχωρισμό αυτό, οποιοδήποτε κομμάτι βελτιωθεί, θα βοηθήσει και τον πράκτορα συνολικά. Για την υλοποίησή του χρησιμοποιήθηκε η λογιστική παλινδρόμηση για τη συνάρτηση αποδοχής, ένας άπληστος αλγόριθμος μεγιστοποίησης της ωφέλειας, ενώ για το κομμάτι του υπολογισμού της τιμολόγησης της προσφοράς του πράκτορα, δοκιμάστηκαν πέντε διαφορετικοί μηχανισμοί. Η μέθοδος προσαρμογής συναρτήσεων προσέγγισης NEAR μέσω της μεθόδου αναζήτησης πολιτικών για τον έλεγχο της διαδικασίας προσφορών παρείχε πολύ καλή συμπεριφορά γενίκευσης σε άγνωστες καταστάσεις και επιλέχθηκε ως η καλύτερη μέθοδος ανάμεσα στις υπόλοιπες.

Κεφάλαιο 8

Εφαρμογή 2: Μηχανισμός πλειοδοσίας για διαδικτυακές διαφημίσεις

Η ραγδαία εξέλιξη του διαδικτύου άλλαξε ριζικά το πεδίο του επιχειρείν. Ένα από τα πιο διαδεδομένα παραδείγματα αυτής της αλλαγής είναι και η διαδικτυακή διαφήμιση (on-line advertising), είτε σε ιστότοπους, είτε σε μηχανές αναζήτησης (search engines). Στην επιχορηγούμενη διαφήμιση (sponsored search), διαφημιστικές καταχωρήσεις παρουσιάζονται μαζί με τα πραγματικά αποτελέσματα (organic results) της μηχανής αναζήτησης. Για το πρώτο εξάμηνο του 2011, η επιχορηγούμενη αναζήτηση αποτέλεσε την υψηλότερη πηγή εσόδων για τη διαδικτυακή διαφήμιση με τζίρο περίπου 7.3 δισεκατομμύρια δολάρια μόνο στις Ηνωμένες Πολιτείες Αμερικής [Pri11].

Στην επιχορηγούμενη αναζήτηση, ο χρήστης της μηχανής αναζήτησης εισάγει ένα ερώτημα στη μηχανή, η οποία θεωρείται ο εκδότης (publisher) των διαφημίσεων. Στο παρασκήνιο τρέχει μια δημοπρασία ανάμεσα στους ενδιαφερόμενους για το συγκεκριμένο ερώτημα διαφημιστές (advertisers). Οι διαφημιστές έχουν πλειοδοτήσει για τις συγκεκριμένες λέξεις κλειδιά με ένα ποσό. Στη σελίδα των αποτελεσμάτων της μηχανής αναζήτησης υπάρχει ένας αριθμός θέσεων (slots) για την τοποθέτηση των διαφημίσεων. Οι πρώτες θέσεις, αυτές που βρίσκονται υψηλότερα στη σελίδα, είναι περισσότερο επιθυμητές καθώς γενικά δίνουν μεγαλύτερη ρυθμαπόδοση σε χτυπήματα (click-through-rate - CTR). Στην τρέχουσα μορφή της η δημοπρασία είναι Generalized Second Price (GSP) auction [JM08]. Σύμφωνα με το πρωτόκολλο της δημοπρασίας, οι πλειοδοσίες ταξινομούνται κατά φθίνουσα σειρά (συχνά πολλαπλασιασμένες με έναν συντελεστή ποιότητας του κάθε διαφημιζόμενου). Ο νικητής της κάθε θέσης πληρώνει τη μικρότερη τιμή με την οποία θα κέρδιζε τη θέση, μια τιμή ελαφρώς υψηλότερη από την προσφορά του δεύτερου πλειοδότη και ανεξάρτητη της προσφοράς του. Τόσο η πλειοδοσία όσο και το τελικό ποσό που πληρώνει ο διαφημιζόμενος στον εκδότη είναι ανά κλικ (cost-per-click or CPC), ενώ υπάρχουν και άλλες προσεγγίσεις όπως, για παράδειγμα, το κόστος

ανά χίλιες εμφανίσεις της διαφήμισης (cost-per-mille or CPM).

Στο κεφάλαιο αυτό θα παρουσιάσουμε τον πράκτορα *Mertacor* που διαγωνίζεται στο διεθνή διαγωνισμό TAC και ο οποίος το 2012 κατέλαβε την 1η θέση στο διαγωνισμό για τη διαδικτυακή διαφήμιση (Ad Auctions). Σε μακροσκοπικό επίπεδο, η στρατηγική του Mertacor μπορεί να διαχωριστεί σε τρεις φάσεις:

1. στην εκτίμηση της αξίας-ανά-κλικ (value-per-click or VPC) για κάθε ερώτημα,

2. στην επιλογή ενός ποσοστού του VPC για την προσφορά που θα κάνει σε κάθε δημοπρασία του και

3. στον υπολογισμό του προϋπολογισμού που θα διαθέσει ο πράκτορας σε κάθε δημοπρασία/ερώτημα.

Η πλειοδοσία με ποσοστό επί του VPC είναι παρόμοια με τη στρατηγική του πράκτορα QuakTAC agent [Vor11] το 2009. Οι διαφοροποιήσεις με τον πράκτορα Mertacor του 2012 είναι:

I. η προσθήκη ενός μοντέλου ΔΗΚ για την ακριβέστερη επιλογή του ποσοστού της αξίας ανά κλικ για τη δημοπρασία,

II. η ύπαρξη ενός φίλτρου σωματιδίων για την καλύτερη πρόβλεψη της κατάστασης του πληθυσμού των χρηστών και

III. ο υπολογισμός του βέλτιστου προϋπολογισμού ανά διαφήμιση με προσομοίωση Monte-Carlo.

8.1 Περιβάλλον προσομοίωσης TAC AA

Η επιχορηγούμενη αναζήτηση είναι ένας ανοικτός, υψηλής πολυπλοκότητας μηχανισμός, ο οποίος δεν καταλήγει σε μία λύση που να υπερτερεί έναντι των άλλων (non-dominant-strategy-solvable) και επομένως θεωρείται περιοχή ενεργούς έρευνας. Για τη διερεύνηση των συμπεριφορών τόσο των μηχανισμών όσο και των στρατηγικών είναι απαραίτητη η ύπαρξη ρεαλιστικής πλατφόρμας προσομοίωσης πρακτόρων λογισμικού [FM08]. Η πλατφόρμα Ad Auctions (AA) του διαγωνισμού TAC είναι ένα τέτοιο σύστημα. Ο αναγνώστης μπορεί να βρει τις λεπτομερείς προδιαγραφές του διαγωνισμού στο [JCC+10]. Για την εξοικείωση με την πλατφόρμα προσομοίωσης, δίνονται παρακάτω οι βασικές πληροφορίες σχετικά με τις οντότητες και τις μεταξύ τους αλληλεπιδράσεις.

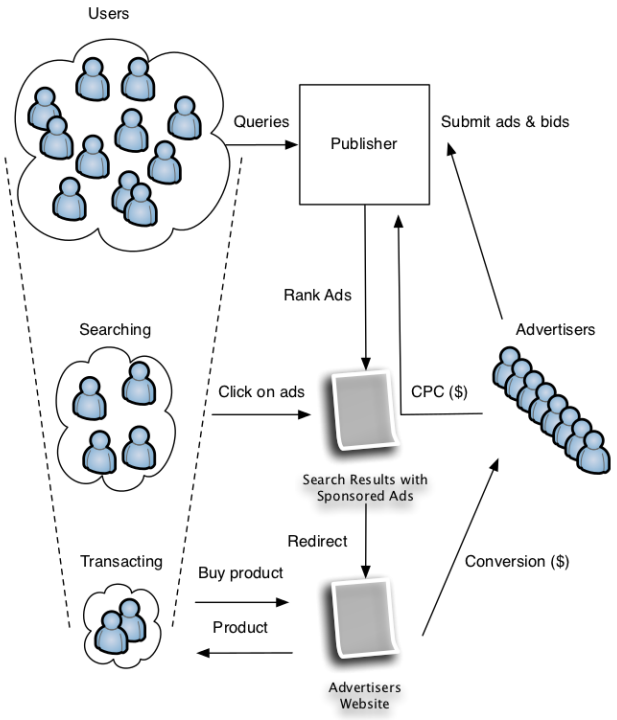
Σε ένα τουρνουά TAC AA υπάρχουν τρεις βασικές οντότητες: α) ο εκδότης *publisher*, β) ένας πληθυσμός από $90,000$ χρήστες *users* και γ) οκτώ διαφημιστές *advertisers* που αποτελούν και τους διαγωνιζόμενους αυτόνομους πράκτορες λογισμικού και αντιπροσωπεύουν τον διαφημιζόμενο. Οι διαφημιστές ανταγωνίζονται για μια θέση διαφήμισης στις σελίδες των αποτελεσμάτων της αναζήτησης. Κάθε

μια σελίδα περιέχει αποτελέσματα αναζήτησης για ένα από δεκαέξι (16) διαφορετικά ερωτήματα, δηλαδή σετ από λέξεις κλειδιά. Προκειμένου να προωθήσουν τα προϊόντα τους, οι πράκτορες συμμετέχουν σε δημοπρασίες, στέλνοντας προσφορές και μια διαφήμιση στον εκδότη, ξεχωριστά για κάθε ερώτημα που ενδιαφέρονται. Οι διαφημίσεις ταξινομούνται σε κάθε σελίδα με βάση μία γενικευμένη μέθοδο που λαμβάνει υπόψιν το συνδυασμό ταξινόμησης σύμφωνα με την προσφορά και σύμφωνα με το τζίρο της διαφήμισης. Κάθε μέρα, οι χρήστες ανάλογα με τις επιθυμίες τους και την κατάστασή τους: α) παραμένουν ανενεργοί, β) κάνουν κλικ σε διαφημίσεις, ή γ) προβαίνουν πιθανώς σε αγορές, δηλαδή μετατροπές των κλικ σε αγορές (conversions) από τους ιστοτόπους των διαφημιζομένων. Τα εμπορευόμενα προϊόντα είναι συνδυασμοί τριών εταιριών και τριών ειδών από το χώρο της οικιακής διασκέδασης. Ο μικρός αριθμός προϊόντων επιτρέπει την έρευνα να εστιάσει σε ένα μικρό σύνολο από προκαθορισμένες λέξεις κλειδιά, χωρίς να ασχοληθεί με το πρόβλημα για παράδειγμα της επιλογής των λέξεων κλειδιών στις οποίες θα συμμετέχει. Οι τρεις κατασκευαστές (ονομαστικά Lionner, PG και Flat) και οι τρεις τύποι συσκευών (TV, Audio και DVD) συνδυάζονται σε εννέα προϊόντα. Η προσομοίωση διαρκεί 60 εικονικές μέρες και κάθε μέρα αντιστοιχεί σε 10 δευτερόλεπτα πραγματικού χρόνου. Μια σχηματική απεικόνιση της πλατφόρμας φαίνεται στο Σχήμα 8.1.

8.1.1 Διαφημιστές

Κάθε διαφημιστής αντιπροσωπεύει έναν πωλητή λιανικής προϊόντων οικιακής διασκέδασης και μπορεί να προμηθεύσει τον εκάστοτε χρήστη με ένα από τα εννέα προϊόντα που υπάρχουν στην πλατφόρμα. Κατά την έναρξη της προσομοίωσης, οι διαφημιστές παίρνουν δύο ειδικεύσεις, μία προς την πώληση ενός συγκεκριμένου προϊόντος και μία ως προς έναν κατασκευαστή. Η κάθε ειδίκευση αυξάνει στο ρυθμό των αγορών των χρηστών μετά από ένα κλικ (conversion rate) ή αυξάνει τα κέρδη ανά μονάδα πώλησης αντίστοιχα. Επιπλέον στους συμμετέχοντες αποδίδεται και μια πενθήμερη μέγιστη χωρητικότητα αποθήκης από τις $C^{cap} \in \{C^{LOW}, C^{MED}, C^{HIGH}\}$, έτσι ώστε, αθροιστικά μέσα σε πέντε ημέρες, αγορές πάνω από το όριο αυτό να γίνονται διαρκώς δυσκολότερο να πραγματοποιηθούν. Κάθε ημέρα d ο πράκτορας διαφημιστής πρέπει:

- Να αποστείλει τις προσφορές για τα ερωτήματα της ημέρας $d + 1$.
- Να επιλέξει μια διαφήμιση για κάθε ερώτημα που θα υποβάλει προσφορά για την ημέρα $d + 1$. Η διαφήμιση μπορεί να είναι είτε γενική (π.χ. να αναφέρεται σε ένα γενικό κατάστημα ηλεκτρικών ειδών) ή στοχευμένη (π.χ. να αναφέρει συγκεκριμένα κάποιο συνδυασμό κατασκευαστή και τύπου προϊόντος). Μια στοχευμένη διαφήμιση, η οποία συμπίπτει με τις επιθυμίες του χρήστη, αυξάνει την πιθανότητα να κάνει ο χρήστης κλικ σε αυτή.
- Να καθορίσει όρια στον προϋπολογισμό για κάθε ερώτημα, αλλά και συνολικά για όλα τα ερωτήματα για την ημέρα $d + 1$.



Σχήμα 8.1: Οι οντότητες που συμμετέχουν στο παιχνίδι TAC AA και οι ενέργειές τους. Η πλειοψηφία των χρηστών κάνουν ερωτήματα, λιγότεροι κάνουν κλικ σε διαφημίσεις και ακόμα λιγότεροι προχωρούν σε συναλλαγές.

Τέλος, τη μέρα d ο διαφημιστής λαμβάνει αναφορές για την αγορά και την καμπάνια του για τη μέρα $d-1$.

8.1.2 Εκδότες

Όπως αναφέρεται παραπάνω, ο εκδότης τρέχει μια GSP δημοπρασία για να καθορίσει την κατάταξη των προσφορών και να καθοριστεί το CPC. Η διαδικασία τοποθέτησης των διαφημίσεων παίρνει υπόψιν της και κάποια προκαθορισμένα κατώφλια. Υπάρχει ένα προκαθορισμένο κατώφλι κάτω από το οποίο μια διαφήμιση δεν εμφανίζεται και ένα δεύτερο πάνω από το οποίο μια διαφήμιση προκρίνεται σε μια διακεκριμένη θέση (promoted). Αν ξεπεραστεί ο προϋπολογισμός ενός πράκτορα που είχε μια διαφήμιση υπό εμφάνιση σε μια σελίδα, η κατάταξη επανυπολογίζεται.

8.1.3 Χρήστες

Κάθε χρήστης επιθυμεί ένα μοναδικό προϊόν. Ο χρήστης μπορεί να βρίσκεται σε διαφορετικές καταστάσεις όσον αφορά τη συμπεριφορά του στον τομέα της

αναζήτησης και της κατανάλωσης:

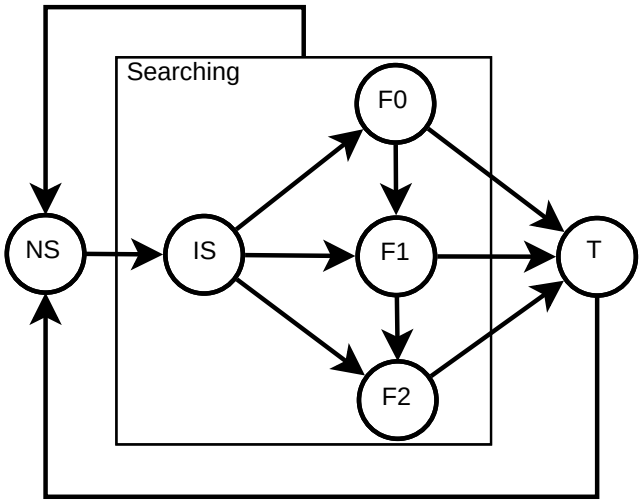
- non-searching (NS), κατάσταση στην οποία ούτε ψάχνει και προφανώς ούτε θέλει να αγοράσει.
- informational searching (IS), κατάσταση στην οποία ψάχνει ανάλογα με την επιθυμία του αλλά δεν αγοράζει
- focus levels 0, 1 και 2 (F1, F2, F3), καταστάσεις στις οποίες ο πράκτορας ψάχνει με διάφορα επίπεδα εστίασης και έχει και αγοραστική βούληση και
- transacted (T), κατάσταση στην οποία ο χρήστης μεταβαίνει μόλις ολοκληρώσει μία αγορά.

Η επιλογή προϊόντων γίνεται με ομοιόμορφο τρόπο. Οι χρήστες υποβάλλουν ερωτήματα τριών τύπων, ανάλογα με το επίπεδο εστίασης, σε ένα σύνολο δεκαέξι (16) διαφορετικών ερωτημάτων, τα οποία είναι: ένα (1) επιπέδου εστίασης τύπου $F0$ όπου ούτε ο κατασκευαστής, ούτε ο τύπος του προϊόντος αναφέρονται, έξι (6) $F1$ ερωτήματα όπου αναφέρεται είτε ο κατασκευαστής είτε ο τύπος του προϊόντος και εννέα (9) $F2$ ερωτήματα, όπου αναφέρεται ο πλήρης καθορισμός ενός προϊόντος (κατασκευαστής και τύπος). Οι μεταβάσεις των χρηστών από τη μια κατάσταση στην άλλη μοντελοποιούνται με αλυσίδα Markov (Σχήμα 8.2). Οι χρήστες που δεν αναζητούν (NS) ή που έχουν ήδη πραγματοποιήσει συναλλαγή (T) δεν πραγματοποιούν αναζητήσεις. Οι χρήστες IS διαλέγουν τυχαία ένα από τα τρία είδη ερωτημάτων (F0,F1,F2), ενώ οι χρήστες με επίπεδο εστίασης επιλέγουν το ερώτημα ανάλογα με το επίπεδο εστίασης στο οποίο ανήκουν. Τόσο οι IS όσο και οι F χρήστες μπορούν να κάνουν κλικ σε μια διαφήμιση, ωστόσο μόνο οι F χρήστες μπορούν να πραγματοποιήσουν αγορές και να μετακινηθούν στην κατάσταση T. Το μοντέλο επιλογής των διαφημίσεων από τους χρήστες είναι βασισμένο στο επεκταμένο μοντέλο καταρράκτη [DGKM08]. Μετά το κλικ σε μια διαφήμιση, προκειμένου να πραγματοποιηθεί η αγορά, η πιθανότητα μετατροπής εξαρτάται από την κατάσταση του χρήστη, την ειδίκευση του διαφημιστή και το υπολειπόμενο στοκ στην αποθήκη.

8.2 Ο πράκτορας Mertacor

Ο πράκτορας Mertacor βασίζεται σε δύο τύπους αλγορίθμων, εκτίμησης και βελτιστοποίησης. Οι αλγόριθμοι εκτίμησης προσπαθούν να προβλέψουν διάφορα χαρακτηριστικά του περιβάλλοντος βάσει των εισόδων που έρχονται στον πράκτορα με τη μορφή ημερησίων αναφορών. Τα χαρακτηριστικά αυτά μπορεί να αφορούν τον ίδιο τον πράκτορα, τους πράκτορες ανταγωνιστές ή την αγορά. Οι αλγόριθμοι βελτιστοποίησης, χρησιμοποιώντας τα εκτιμώμενα χαρακτηριστικά βελτιστοποιούν την πολιτική αποφάσεων του πράκτορα με σκοπό το μεγαλύτερο δυνατό κέρδος για τον ίδιο.

Ποιο συγκεκριμένα ο πράκτορας Mertacor αποτελείται από τα παρακάτω αλγοριθμικά τμήματα:



Σχήμα 8.2: Οι ομάδες χρηστών και οι κατευθύνσεις μετάβασης από τη μία κατάσταση στην άλλη για την επόμενη μέρα. Οι πιθανότητες μετάβασης από F σε T εξαρτώνται από τις ικανότητες των πρακτόρων που συμμετέχουν, ενώ οι υπόλοιπες έχουν προδιαγραφεί από το διαγωνισμό.

A1. Εκτίμηση κατάστασης του πληθυσμού των χρηστών με φίλτρο σωματιδίων (Εκτίμηση)

A2. Κανόνας πρόβλεψης του στοκ που υπολείπεται (Εκτίμηση)

A3. Εκτίμηση αξίας ανά κλικ για κάθε δημοπρασία (Εκτίμηση)

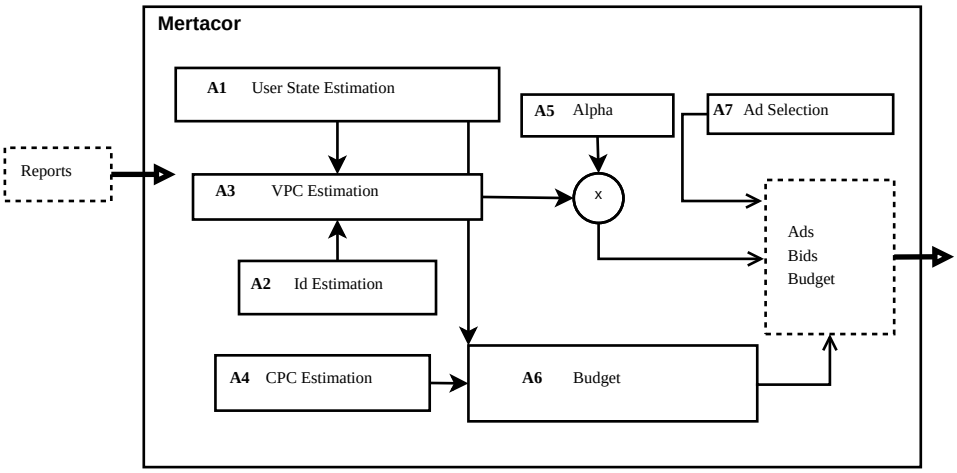
A4. Εκτίμηση CPC με βάση το bid (Εκτίμηση)

A5. Υπολογισμός α (Βελτιστοποίηση)

A6. Υπολογισμός προϋπολογισμού (Βελτιστοποίηση)

A7. Κανόνας επιλογής διαφήμισης (Βελτιστοποίηση)

Η βασική στρατηγική του πράκτορα Mertacor είναι η πλειοδοσία σε ποσοστό επί της αξίας-ανά-κλικ (value-per-click or VPC), η οποία είναι η αξία που έχει για τον πράκτορα κάθε κλικ του χρήστη. Παράδειγμα αξίας είναι το κέρδος-ανά-κλικ. Οπότε, από την αξία-ανά-κλικ ένα ποσοστό πάει στον εκδότη και ένα παραμένει στον πράκτορα. Σκοπός μας είναι η εύρεση του βέλτιστου αυτού ποσοστού. Παρόμοια στρατηγική είχε αναπτύξει και ο πράκτορας QuakTAC στο τουρνουά του 2009 [Vor11] κατακτώντας την τέταρτη θέση. Πρόκειται για μία από τις λίγες στρατηγικές που βασίζονται σε παιχνιδοθεωρητική ανάλυση, είναι ακέραια, απλή και κομψή ταυτόχρονα. Θεωρείται ότι η σχέση μεταξύ της προσφοράς και της αξίας-ανά-κλικ (v) του πράκτορα είναι γραμμική καθώς ο πράκτορας πλειοδοτεί ένα ποσοστό (shading) α της αξίας. Έτσι για κάθε ερώτημα q, η προσφορά για τη μέρα



Σχήμα 8.3: Αρχιτεκτονική του πράκτορα Mertacor.

$d+1$ είναι:

$$bid^q_{d+1} = \alpha \cdot \upsilon^q_{d+1} \tag{8.1}$$

8.2.1 Εκτίμηση κατάστασης χρηστών

Η εκτίμηση της κατάστασης του πληθυσμού των χρηστών χρησιμοποιείται τόσο για τον υπολογισμό της αξίας-ανά-κλικ όσο και για τη βελτιστοποίηση του προϋπολογισμού. Ο αλγόριθμος εκτίμησης είναι ένα φίλτρο σωματιδίων [AMGC02]. Για την ακρίβεια χρησιμοποιείται ένα φίλτρο σωματιδίων για κάθε ομάδα χρηστών που έχουν την ίδια επιλογή σε προϊόν, για παράδειγμα $\{$Lioneer,DVD$\}$. Η κάθε ομάδα χρηστών αποτελείται από $10,000$ χρήστες και τα εννέα διαφορετικά προϊόντα δίνουν τον συνολικό αριθμό των χρηστών που ανέρχεται στις $90,000$.

Κάθε φίλτρο διαθέτει $1,000$ σωματίδια και κάθε σωματίδιο περιέχει την κατανομή των $10,000$ χρηστών στις καταστάσεις τους $\{$NS, IS, F0, F1, F2, T$\}$. Το βάρος του κάθε σωματιδίου περιγράφει τη σχετική πιθανότητα η κατανομή των χρηστών να είναι η πραγματική κατανομή και μεταβάλλεται κάθε φορά που εισέρχεται μία νέα παρατήρηση στο σύστημα. Κάθε μέρα νέα σωματίδια δημιουργούνται βάσει των προηγουμένων. Επιλέγεται ένα σωματίδιο τυχαία, βάσει του βάρους του, και ένα νέο δημιουργείται από τις πιθανότητες μετάβασης των χρηστών, όπως αυτές είναι προκαθορισμένες από τις προδιαγραφές του διαγωνισμού (Σχήμα 8.2). Οι μοναδικές πιθανότητες που δε δίνονται είναι αυτές της μετάβασης στην κατάσταση T και οι οποίες υπολογίζονται από ιστορικά δεδομένα συναλλαγών (Σχήμα 8.2). Τα βάρη των σωματιδίων ανανεώνονται σύμφωνα με τις παρατηρήσεις του πράκτορα για την τρέχουσα κατάσταση του πληθυσμού. Η μοναδική χρήσιμη παρατήρηση που δίνεται από τις αναφορές του συστήματος προς τον πράκτορα είναι οι συνολικές παρουσιάσεις της διαφήμισής του σε ένα ερώτημα τύπου F2 και αυτό γιατί μπορούμε να υπολογίσουμε το σφάλμα μεταξύ της κατανομής του σωματιδίου και της παρατήρησης με μεγαλύτερη ακρίβεια. Έστω ότι x_{IS} οι χρήστες IS

και x_{F2} οι χρήστες F2 που δείχνει η κατανομή του σωματιδίου και i οι εμφανίσεις (impressions) της διαφήμισης για το ερώτημα επιπέδου F2. Τότε η διαφορά τους θα είναι:

$$diff = i - (x_{IS} + x_{F2}) = i - \frac{1}{3}x_{F2} - x_{F2} \tag{8.2}$$

Χρησιμοποιώντας μια Γκαουσιανή συνάρτηση με παραμέτρους που προέκυψαν ύστερα από πειραματισμό με τα δεδομένα, επιλέχθηκε η παρακάτω συνάρτηση επανυπολογισμού του βάρους:

$$f(d) = 1000 \cdot exp(\frac{-d * d}{1000}) \tag{8.3}$$

Η παραπάνω συνάρτηση δίνει πολύ μικρές τιμές για σωματίδια που απέχουν της παρατήρησης και πολύ μεγάλη για σωματίδια που συμπίπτουν με αυτή. Ο υπολογισμός της κατανομής των χρηστών προκύπτει από το ζυγισμένο μέσο όρο όλων των σωματιδίων. Ένα παράδειγμα εκτίμησης της κατάστασης όλων των χρηστών $F2$ μία μέρα μετά έχοντας παρατηρήσεις μία μέρα πριν την εκτίμηση παρουσιάζεται στο Σχήμα 8.4.

8.2.2 Εκτίμηση αξίας ανά κλικ

Η εκτίμηση του VPC για κάθε σετ λέξεων κλειδιών μπορεί να εκφραστεί ως το γινόμενο της πιθανότητας να γίνει μετατροπή ενός κλικ σε αγορά και της προσδοκώμενης αξίας (κέρδους, τζίρου κτλ) από μια τέτοια μετατροπή. Για κάθε ερώτημα q, ο πράκτορας υπολογίζει το VPC ως εξής:

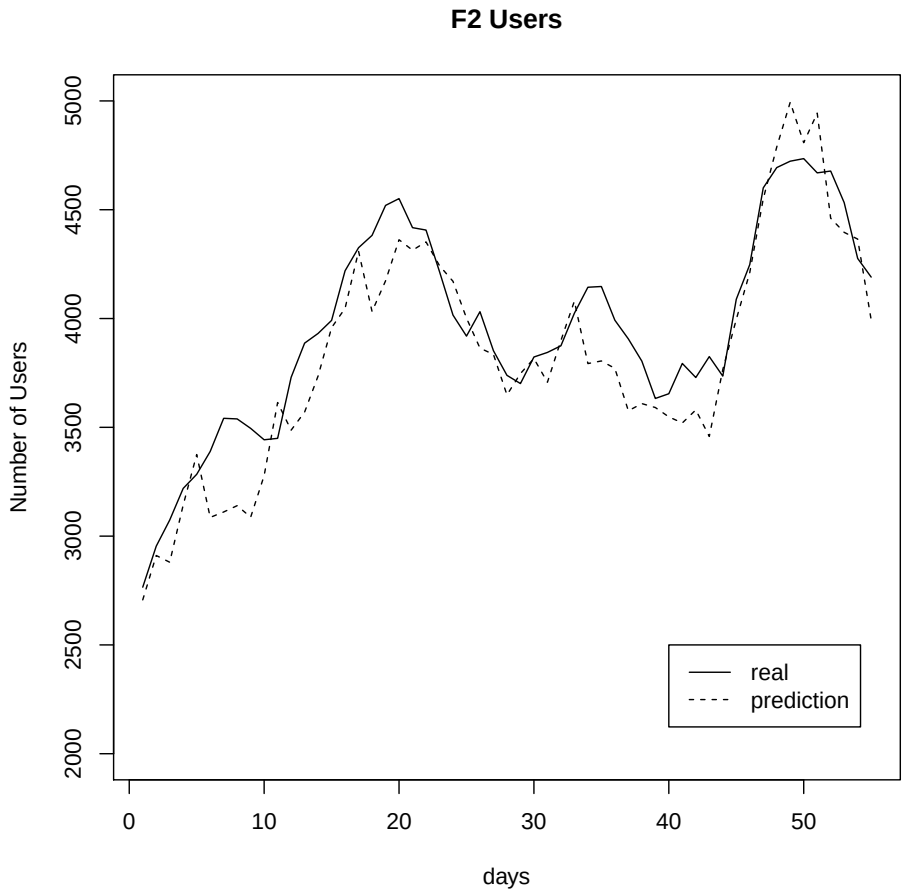
$$\widehat{v}^q = \widehat{Pr}^q\{conversion|click\} \cdot E[revenue^q|conversion] \tag{8.4}$$

Η προσδοκώμενη αξία (expected revenue) για κάθε ερώτημα q, δεδομένης μιας μετατροπής ($E[revenue^q|conversion]$), εξαρτάται αποκλειστικά από την ειδικότητα του διαφημιστή ως προς τον κατασκευαστή (Manufacturer Specialty - MS) και μπορεί να υπολογιστεί χωρίς να χρειάζεται άλλη πληροφορία όπως παρακάτω:

$$E[revenue^q|conversion] = \begin{cases} (USP \cdot (3 + MSB))/3 & \text{MS not defined in } q \\ USP \cdot (1 + MSB) & \text{MS matched in } q \\ USP & \text{MS not matched in } q \end{cases} \tag{8.5}$$

όπου USP είναι το κέρδος ανά μονάδα (Unit Sales Profit or USP) το οποίο είναι \$10 στο TAC 2012 και MSB το μπόνους ειδικότητας (manufacturer specialty bonus or MSB) το οποίο είναι 0.4 στο TAC 2010. Οι όροι κέρδος και τζίρος εναλλάσσονται καθώς πέραν του κόστους ανά κλικ δεν υπάρχει κάποιο άλλο κόστος για το διαφημιζόμενο στο διαγωνισμό.

Για τον υπολογισμό της πιθανότητας να υπάρξει μετατροπή δεδομένου ενός κλικ, χρειαζόμαστε δύο πρόσθετους εκτιμητές: α) το ποσοστό των χρηστών που



Σχήμα 8.4: Πρόβλεψη πληθυσμού χρηστών τύπου $F2$ για τη μέρα $d+1$ με φίλτρο σωματιδίων.

έχουν στόχο (focused users) και β) προηγούμενες και μελλοντικές μετατροπές, καθώς αυτές επηρεάζουν την πιθανότητα μετατροπής λόγω της ύπαρξης ορίου στο στοκ της αποθήκης:

$$\widehat{Pr}^{q}_{d+1}\{conversion|click\} = \widehat{focusedPercentage}^{q} \cdot Pr^{q}\{conversion|focused\}(\hat{I}_{d+1}) \tag{8.6}$$

Για κάθε ερώτημα q μπορούμε από το φίλτρο σωματιδίων να υπολογίσουμε το ποσοστό των στοχευμένων χρηστών έναντι αυτών που ψάχνουν για πληροφορίες. Για παράδειγμα για το ερώτημα $q = Lioneer, null$ το ποσοστό είναι:

$$\widehat{focusedPercentage}^{\{Lioneer,null\}} = \frac{F}{F+IS} \tag{8.7}$$

με

$$F = \frac{F1^{\{Lioneer,Audio\}} + F1^{\{Lioneer,TV\}} + F1^{\{Lioneer,DVD\}}}{3} \tag{8.8}$$

και

$$IS = \frac{IS^{\{Lioneer,Audio\}}/3 + IS^{\{Lioneer,TV\}}/3 + IS^{\{Lioneer,DVD\}}/3}{3} \tag{8.9}$$

όπου τα $F1^q$ και IS^q δίνονται από το φίλτρο σωματιδίων για τρεις, στη συγκεκριμένη περίπτωση, ομάδες πληθυσμού.

Όσον αφορά στον υπολογισμό της πιθανότητας μετατροπής (δεύτερος παράγοντας της Εξίσωσης 8.6), ο υπολογισμός της γίνεται με βάση τις προδιαγραφές του παιχνιδιού αρκεί να υπολογιστεί η μεταβλητή I_d. Στο [Jor10] τονίζεται ότι η επίδραση της περιορισμένης διανομής (distribution contraint effect), που αντικατοπτρίζεται στη μεταβλητή I_d, είναι ο δεύτερος σημαντικότερος παράγοντας στην επίδοση των πρακτόρων μετά την κατασκευαστική ειδίκευση. Με βάση τις προδιαγραφές για τον υπολογισμό του I_d χρειάζονται και πρέπει να υπολογιστούν οι μετατροπές-συναλλαγές, c τις μέρες d και $d+1$:

$$I_{d+1} = g(c_{d-3} + c_{d-2} + c_{d-1} + \hat{c}_d + \hat{c}_{d+1} - C^{cap}) \tag{8.10}$$

όπου g μια συνάρτηση που περιγράφεται στις προδιαγραφές του διαγωνισμού. Για τον υπολογισμό των $\hat{c_d}$ και $\hat{c_{d+1}}$ χρησιμοποιούμε τους παρακάτω απλούς μέσους όρους:

$$\hat{c_d} = \frac{c_{d-1} + c_{d-2} + c_{d-3}}{3} \hat{c_{d+1}} = \frac{\hat{c_d}}{2} \tag{8.11}$$

Για το $\hat{c_{d+1}}$ διαιρούμε δια δύο καθώς η τιμή του αφορά μελλοντικές αγορές, που θα ξεκινήσουν από το μηδέν, για να φτάσουν ιδανικά στο $\hat{c_{d+1}}$. Για το λόγο αυτό επιστρέφουμε και τη μισή εκτίμηση.

8.2.3 Μηχανισμός πλειοδοσίας

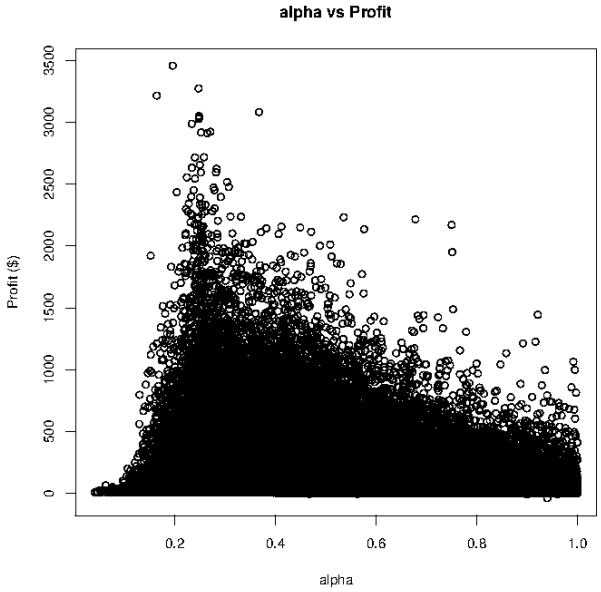
Με δεδομένο ότι οι δημοπρασίες τύπου GSP δεν είναι συμβιβασμένες με κίνητρα [Lah06], περιμένουμε $\alpha < 1$. Απομένει λοιπόν να υπολογίσουμε το ποσό αυτό α του bid. Ακολουθώντας τη μεθοδολογία στο [Vor11], περιορίζουμε το πεδίο της έρευνας σε απλά συμμετρικά σημεία ισορροπίας (simple symmetric equilibria) και σε διακριτές τιμές από 0 έως 1 με βήμα 0.1. Αυτό σημαίνει ότι όλοι, εκτός από έναν, οι διαφημιστές θα ακολουθούν την ίδια στρατηγική ($bid = \alpha v$) ενώ ένας παίκτης θα πλειοδοτεί με ποσοστό α_{single} που θα αλλάζει ανά παιχνίδι. Οι τιμές του α_{single} καθορίζονται από μια επαναληπτική μέθοδο καλύτερης απάντησης (iterative best response method), η οποία ξεκινάει από μια αληθή ομογενή κατανομή στρατηγικών προφίλ (truthful homogeneous strategy profile) με $\alpha = 1$, $\alpha_{single} = 1$. Οι τιμές αυτές είναι λογικές για την αρχή καθώς η GSP είναι μια γενίκευση της δημοπρασίας Vickrey (Vickrey auction), όπου η αληθοφανής στρατηγική για έναν πράκτορα είναι και η βέλτιση [Kri02]. Στη συνέχεια η μέθοδος αναζητά την καλύτερη απόκλιση α_{single} από το προφίλ αυτό και χρησιμοποιείται ως νέο ομογενές προφίλ. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να καταλήξει σε Bayes-Nash σημείο ισορροπίας. Για τη χρονιά 2010 και 2011 χρησιμοποιήθηκε η τιμή του $\alpha_{single} = 0.3$, η οποία βρέθηκε σε τρεις επαναλήψεις (1, 0.4 και 0.3). Για το 2012 τα πειράματα επαναλήφθηκαν εκ νέου όταν σύμφωνα με ανάλυση των δεδομένων του διαγωνισμού το 0.335 εμφανίζονταν ως η τιμή με τα περισσότερα κέρδη. Οπότε, η διαδικασία επαναλήφθηκε στο διάστημα $[0.25, 0.35]$ με βήμα 0.01, όπου και βρέθηκε ότι η βέλτιση τιμή είναι η 0.33 (Πίνακας 8.1). Στο Σχήμα 8.5 φαίνεται η σχέση μεταξύ α και κέρδους. Τα περισσότερα και μεγαλύτερα κέρδη αποκομίζονται στο διάστημα $[0.2, 0.4]$. Δηλαδή σε ένα διάστημα που αντιπροσωπεύει το 20% του συνολικού, παρουσιάζεται το 46% των συνολικών κερδών των πρακτόρων.

Πίνακας 8.1: Mertacor ($\alpha_{signle} = 0.33$) vs. Mertacors ($\alpha = 0.3$)

Κατάταξη	Πράκτορας	Μέσο σκορ ($)	Παιχνίδια
1	Mertacor	46,664	12
2	Mertacor4	44,133	12
3	Mertacor5	44,042	12
4	Mertacor3	43,545	12
5	Mertacor2	43,184	12
6	Mertacor6	42,730	12
7	Mertacor1	42,145	12
8	Mertacor7	41,002	12

8.2.4 Βελτιστοποίηση προϋπολογισμού

Για τη βελτιστοποίηση του τρόπου διάθεσης του στοκ της αποθήκης κάθε ημέρα, δημιουργήθηκε ένας αλγόριθμος προσομοίωσης τύπου Monte Carlo, ο οποίος



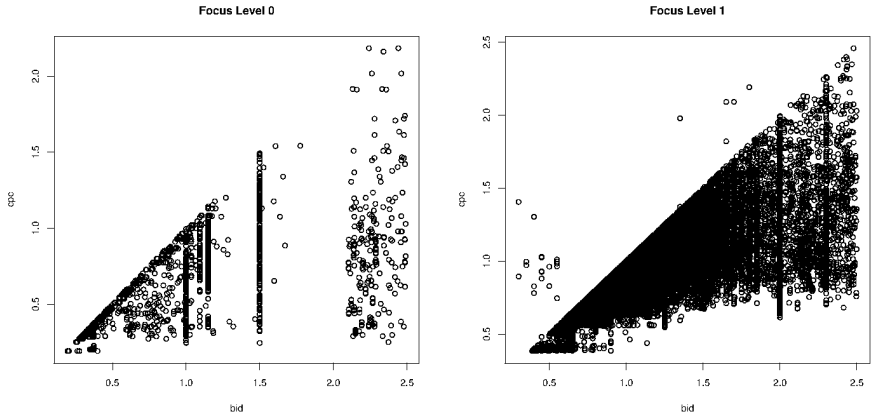
Σχήμα 8.5: Διάγραμμα συσχετισμού μεταξύ της τιμής α και του κέρδους για κάθε πλειοδοσία, σύμφωνα με τα δεδομένα του διαγωνισμού του 2012.

καταρτίζει τη συγκεκριμένη μέρα έναν προϋπολογισμό για κάθε σετ λέξεων κλειδιών, ώστε να δοθεί μεγαλύτερο κομμάτι των εξόδων σε διαφημίσεις που αναμένεται να οδηγήσουν σε υψηλότερα κέρδη (Αλγόριθμος 8.1). Στην αντίθετη περίπτωση, αν ο προϋπολογισμός ήταν ανοικτός, σταδιακά μέσα στην ημέρα, η πιθανότητα μετατροπής ενός κλικ σε αγορά θα μειώνονταν, εξαιτίας των αγορών που θα εξαντλούσαν το στοκ και το κέρδος θα ήταν μικρότερο, αφού ένας χρήστης θα έκανε κλικ αλλά θα είχε μικρότερες πιθανότητες να αγοράσει. Ο αλγόριθμος του προϋπολογισμού δίνει σε κάθε σετ από κλειδιά έναν προϋπολογισμό ο οποίος είναι μυωπικός, δηλαδή ψευδο-βέλτιστος για τη συγκεκριμένη ημέρα. Ο προϋπολογισμός πλησιάζει στον πραγματικά βέλτιστο, ανάλογα με την ακρίβεια των εκτιμητών που χρησιμοποιεί η προσομοίωση.

Αναγκαίος εκτιμητής για την προσομοίωση είναι ο υπολογισμός του CPC δεδομένου ενός bid. Το CPC, όπως προαναφέρθηκε, είναι το κόστος ανά κλικ το οποίο προκύπτει σύμφωνα με τους κανόνες της δημοπρασίας και τα bids των χρηστών. Υπολογίστηκε από ιστορικά δεδομένα παλαιότερων δημοπρασιών σύμφωνα με μια απλή γραμμική μοντελοποίηση όπως δίνεται από την Εξίσωση 8.12. Ο λόγος πίσω από την επιλογή της συνάρτηση $y = ax$ φαίνεται στα Σχήματα 8.6(a), 8.6(b)και 8.6(c), όπου η χρήση μιας γραμμικής συνάρτησης η οποία θα περνά από την αρχή των αξόνων φαίνεται να είναι μια απλή και εύρωστη λύση. Χρησιμοποιήθηκε, λοιπόν, μια γραμμική συνάρτηση για κάθε διαφορετικό επίπεδο εστίασης χρήστη (F0,F1,F2), όπου τα a_1, a_2 και a_3 προέκυψαν από τη μέθοδο ελαχίστων τετραγώνων. Οι τιμές που υπολογίστηκαν είναι $0.5188, 0.7477$ και 0.8082 αντί-

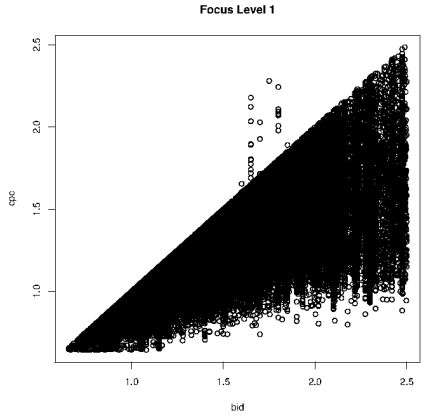
στοιχα.

$$CPC = f_i(bid) = a_i \cdot bid \tag{8.12}$$



(a) Επίπεδο εστίασης F0

(b) Επίπεδο εστίασης F1



(c) Επίπεδο εστίασης F2

Σχήμα 8.6: Αντικατοπτρισμός δεδομένων διαγωνισμού από προσφορές (bids) σε CPC.

8.2.5 Επιλογή διαφήμισης

Η εργασία της επιλογής διαφήμισης είναι αρκετά ξεκάθαρη τόσο για ερωτήματα τύπου $F0$ (χωρίς λέξεις κλειδιά), όσο και για ερωτήματα τύπου $F2$ (που περιέχουν και τη μάρκα και τον τύπο του προϊόντος). Για την πρώτη περίπτωση, υπάρχει

Αλγόριθμος 8.1 Αλγόριθμος υπολογισμού προϋπολογισμού με τεχνική Monte Carlo.

BUDGET-MONTE-CARLO

Require: bids

1: {Έστω ότι ο μέγιστος αριθμός κλικ σε μία διαφήμιση είναι 3000}
2: **for all** Runs of MC **do**
3: conversions = 0
4: runningProfit = 0
5: bestProfit = 0
6: **for** i = 1 to 3000 **do**
7: sum = 0
8: **for all** Queries **do**
9: {Η συνάρτηση f δίνει τη πιθανότητα μετατροπής δεδομένων των μετατροπών που έχουν γίνει στην προσομοίωση}
10: $conversionProbability = f(conversions)$
11: {Η συνάρτηση g μετατρέπει το bid σε cpc}
12: $cpc[q] = g(bids[q])$
13: {ppc: profit per click}
14: $ppc[q] = vpc - cpc$
15: **if** $bid < minimumbidforslot$ **then**
16: $profit[q] = 0$
17: **else**
18: $profit[q] = ppc$
19: **end if**
20: $sum+ = profit$
21: **end for**
22: **for all** Queries **do**
23: $profit[q] = profit[q]/sum$
24: **end for**
25: {Select via roulette wheel selection the query to assign budget, q'}
26: $q' = roullette(profit[q])$
27: $budget[q']+ = cpc[q']$
28: $runningProfit+ = ppc[q']$
29: $conversions++$
30: **if** $runningProfit > bestProfit$ **then**
31: $bestProfit = runningProfit$
32: $finalBudget = budget$
33: **end if**
34: **end for**
35: $MCBudget+ = finalBudget$
36: **end for**
37: $MCBudget/ = runs$
38: **return** $MCBudget$

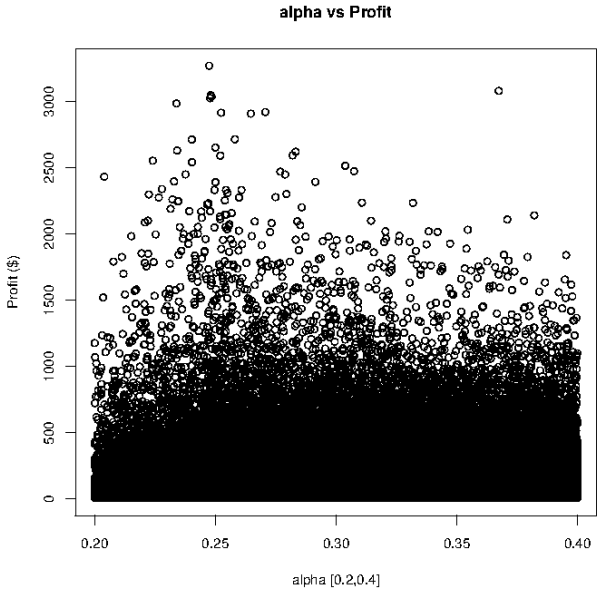
πιθανότητα ίση με $\frac{1}{9}$ να πετύχουμε τη σωστή επιλογή για το χρήστη με μια στοχευμένη διαφήμιση, οπότε μια γενική διαφήμιση είναι η καλύτερη επιλογή. Από την άλλη με ένα ερώτημα $F2$ ο χρήστης εμφανίζει την προτίμησή του με αληθοφάνεια οπότε ταιριάζει μια στοχευμένη διαφήμιση. Για τα ερωτήματα τύπου $F1$, που περιέχουν είτε τη μάρκα είτε τον τύπο του προϊόντος, επιλέγονται και εδώ στοχευμένες διαφημίσεις όποτε εμφανίζεται μια μάρκα ή ένας τύπος προϊόντος στον οποίο έχουμε ειδίκευση, στοχεύοντας στο ότι το αυξημένο κέρδος καθώς και η αυξημένη πιθανότητα μετατροπής θα ευνοήσει τον πράκτορα. Η συγκεκριμένη στρατηγική αποδείχθηκε αποτελεσματική και για τον πράκτορα QuakTAC [Vor11].

8.3 Επέκταση με NEAR

Όπως περιγράφηκε στην Ενότητα 8.2.4, η τιμή του α είναι μεν τεκμηριωμένη παιχνιδοθεωρητικά σε ένα διάστημα μεταξύ $[0.2, 0.4]$, ωστόσο η ακριβής τιμή εξαρτάται και από το σετ των πρακτόρων που συμμετέχουν στο παιχνίδι [Vor11]. Στο Σχήμα 8.7 φαίνονται οι διάφορες τιμές του α στο επιλεγμένο διάστημα. Στο σχήμα κάποιος μπορεί να παρατηρήσει παρόμοια κέρδη για διάφορες τιμές του α. Στόχος, λοιπόν, είναι η επιλογή της βέλτιστης τιμής του α ανάλογα με την κατάσταση του παιχνιδιού. Ένα καλό χαρακτηριστικό της κατάστασης του παιχνιδιού στην οποία βρίσκεται ο πράκτορας κάθε χρονική στιγμή είναι η τιμή του VPC, αφού αυτό εκφράζει την κατάσταση των χρηστών, την κατάσταση της αποθήκης και το προσδοκώμενο κέρδος από μια αγορά. Το παραπάνω πρόβλημα μπορεί να τεθεί ως ένα συσχετιστικό πρόβλημα n-κουλοχέρηδων (associative n-armed bandit problem) [SB98], όπου για κάθε μέρα και δημοπρασία θα επιλέγονταν η ενέργεια που θα έδινε το μεγαλύτερο άμεσο κέρδος. Το μοντέλο που θα διατηρεί τις συσχετίσεις θα είναι ένα ΔΗΚ το οποίο θα εξελιχθεί με τη βοήθεια της μεθόδου NEAR. Κάθε ΔΗΚ του πληθυσμού θα εκπαιδεύεται στο σετ δεδομένων από παιχνίδια που συμμετέχουν αντίπαλοι πράκτορες και θα προσπαθεί να επιλέξει το καλύτερο α από ένα σετ διακριτών τιμών, με γνώμονα τη μέγιστη ωφέλεια για τον πράκτορα. Το ΔΗΚ πέρα από τα μη γραμμικά χαρακτηριστικά που θα αναπτύσσει, θα παρουσιάζει δυνατότητες υπολογισμού και χρονικών χαρακτηριστικών όπου μέσα σε αυτά θα συγκαταλέγεται και η μεταβολή του VPC από μέρα σε μέρα.

Η χρησιμότητα επιλογής του α βάσει του οφέλους που παρουσιάζει για τον πράκτορα φαίνεται στο εξής παράδειγμα: Έστω λόγω χαμηλών τιμών VPC (π.χ μικρό στοκ, λίγοι F2 χρήστες), ο πράκτορας κρατά χαμηλό το α για να γλιτώσει την εμφάνιση της διαφήμισης και επομένως μερικά κλικ που δύσκολα θα προέβαιναν σε μετατροπές. Εν συνεχεία, όταν αυξάνονταν το στοκ της αποθήκης ή οι χρήστες τύπου F2, θα μπορούσε να δώσει μια διαφορετική τιμή του α, πιθανώς υψηλότερη, και να αποκομίσει μεγαλύτερο κέρδος από μια καλύτερη κατάταξη. Σε κάθε περίπτωση τα δεδομένα θα οδηγήσουν στην καλύτερη επιλογή της τιμής του α.

Η μέθοδος NEAR χρησιμοποιήθηκε για την ανάπτυξη ενός ΔΗΚ για τον τύπο χρηστών εστίασης επιπέδου 2 (F2), δηλαδή όταν ο πράκτορας καλύπτει τόσο την ειδίκευση στον κατασκευαστή όσο και στον τύπο προϊόντος. Αυτή άλλωστε είναι



Σχήμα 8.7: Διάγραμμα συσχετισμού μεταξύ της τιμής α και του κέρδους για το διάστημα $[0.2, 0.4]$.

και η κατηγορία από την οποία ο πράκτορας αποκομίζει τα μεγαλύτερα κέρδη. Το ΔΗΚ έχει μία είσοδο, το VPC, και 21 εξόδους, όσες οι τιμές του α στο διάστημα $[0.2, 0.4]$ με βήμα 0.01. Η ανταμοιβή αποτελείται από το κέρδος του πράκτορα $r = revenue - CPC \cdot \#clicks$. Η εκπαίδευση των βαρών γίνεται βάσει του αλγορίθμου κατάβασης πλαγιάς.

8.4 Πειράματα και αποτελέσματα

Η ενότητα αυτή παρουσιάζει τόσο τα αποτελέσματα του διαγωνισμού TAC AA 2012, όσο και ορισμένα στοχευμένα πειράματα που έγιναν για την αξιολόγηση της επέκτασης με ΔΗΚ μέσω της μεθόδου NEAR.

8.4.1 Αποτελέσματα διαγωνισμού TAC AA 2012

Προκριματικά

Στόχος των προκριματικών είναι η επιβεβαίωση της καλής λειτουργίας των πρακτόρων προκειμένου να συνεχίσουν στην επόμενη φάση. Η διάρκεια των προκριματικών ήταν μια μέρα και πραγματοποιήθηκαν 64 παιχνίδια (Πίνακας 8.2). Ο πράκτορας Mertacor δεν είχε διαφοροποιήσεις από τις εκδόσεις του πράκτορα που διαγωνίστηκαν το 2010 και το 2011.

Πίνακας 8.2: Αποτελέσματα των προκριματικών του τουρνουά TAC AA 2012.

Κατάταξη	Πράκτορας	Μέσο Σκορ ($)
1	Tau	55,145
2	Schelmazl	53,589
3	epflagent	49,816
4	Mertacor	48,861
5	Time	45,455
6	fsr	40,592
7	WayneAd	32,944
8	CrocodileAgent	32,076
9	TUCTAC	31,753
10	THA	30,276

Ημιτελικά

Στα ημιτελικά ο πράκτορας κατέκτησε την τρίτη θέση. Οι οκτώ πρώτοι των ημιτελικών προκρίνονται στα τελικά. Συνολικά πραγματοποιήθηκαν 32 παιχνίδια (Πίνακας 8.3). Όσον αφορά στον πράκτορα Mertacor, στα ημιτελικά προστέθηκαν τα φίλτρα σωματιδίων.

Πίνακας 8.3: Αποτελέσματα των ημιτελικών του τουρνουά TAC AA 2012.

Κατάταξη	Πράκτορας	Μέσο Σκορ ($)
1	Tau	54,796
2	Schelmazl	53,890
3	Mertacor	50,614
4	epflagent	49,661
5	Time	47,673
6	TUCTAC	44,065
7	fsr	42,338
8	CrocodileAgent	41,323
9	THA	33,181
10	WayneAd	30,692

Τελικά

Οι οκτώ καλύτεροι πράκτορες συναγωνίστηκαν στα τελικά, όπου νικητής αναδείχθηκε ο πράκτορας Mertacor ύστερα από 64 παιχνίδια. Στον πράκτορα Mertacor προστέθηκε η προσομοίωση του προϋπολογισμού, καθώς και η εκ νέου παιχνιδοθεωρητική αναζήτηση για το α το οποίο τέθηκε ίσο με 0.33 από 0.3

Πίνακας 8.4: Αποτελέσματα τελικών του τουρνουά TAC AA 2012

Κατάταξη	Πράκτορας	Μέσο Σκορ ($)
1	Mertacor	52,960
2	Schelmazl	52,701
3	tau	51,913
4	epflagent	47,240
5	Time	46,996
6	TUCTAC	45,594
7	CrocodileAgent	41,990
8	fsr	40,916

8.4.2 Στοχευμένα πειράματα

Για την αξιολόγηση της αποτελεσματικότητας της επέκτασης του πράκτορα με τη βοήθεια της μεθόδου NEAR, αλλά και γενικότερα της αντιμετώπισης του προβλήματος με μοντελοποίηση ενισχυτικής μάθησης δημιουργήθηκαν δύο τουρνουά. Στο πρώτο τουρνουά συμμετείχαν δύο πράκτορες τύπου Mertacor ($\alpha = 0.33$ και $\alpha = 0.3$), καθώς και έξι πράκτορες που επιλέχθηκαν από το καταθετήριο πρακτόρων [1]. Το τουρνουά διήρκεσε 60 παιχνίδια και στόχος ήταν η συλλογή δεδομένων από ένα σύνολο πρακτόρων.

Στη συνέχεια η μέθοδος NEAR χρησιμοποιήθηκε για την ανάπτυξη των ΔΗΚ που θα εκπαιδεύονταν στα δεδομένα του τουρνουά. Το μέγεθος του πληθυσμού ήταν 50 οργανισμοί και ο πληθυσμός εξελίχθηκε για 50 γενιές.

Στο δεύτερο τουρνουά χρησιμοποιήθηκε μία έκδοση του πράκτορα Mertacor ($\alpha = 0.33$), οι έξι αντίπαλοι και ο πράκτορας Mertacor+NEAR. Στο σημείο αυτό πρέπει να αναφερθεί ότι και στα δύο τουρνουά, για την ισότιμη παρουσία τους στο διαγωνισμό όλοι οι πράκτορες είχαν την ίδια, μέση χωρητικότητα αποθήκης (C^{MED}).

Τα αποτελέσματα των τουρνουά φαίνονται στους Πίνακες 8.5 και 8.6. και φανερώνουν ότι ο πράκτορας με την επέκταση NEAR έχει ένα μικρό προβάδισμα έναντι των αντιπάλων του. Παρά το γεγονός ότι οι διαφορές δεν είναι πολύ μεγάλες και δεν είναι στατιστικά σημαντικές, η ίδια συμπεριφορά παρατηρείται και στα τουρνουά, όπου οι διαφορές δεν είναι στατιστικά σημαντικές συνήθως.

8.5 Σύνοψη

Στο κεφάλαιο αυτό περιγράψαμε τα βασικά συστατικά του πράκτορα *Mertacor*, νικητή του διαγωνισμού TAC AA 2012. Είδαμε πως μία εμπειρική παιχνιδοθεωρητική προσέγγιση μπορεί να δώσει εύρωστα και κερδοφόρα αποτελέσματα χρησιμοποιώντας απλή αφαιρετική μοντελοποίηση. Όσον αφορά τη μέθοδο NEAR, είδαμε

[1] http://www.sics.se/tac/showagents.php

Πίνακας 8.5: Τουρνουά Α: Μέσο σκορ στα 60 παιχνίδια και ίσο περιορισμό χωρητικότητας $C = C^{MED}$.

Κατάταξη	Πράκτορας	Μέσο σκορ ($)
1	MertacorA($\alpha = 0.33$)	57,161
2	MertacorB($\alpha = 0.3$)	56,567
3	TacTex	55,739
4	Schlemazl	55,315
5	QuacTac	54,282
6	epflagent	41,301
7	Merlion	38,001
8	Wayne	36,206

Πίνακας 8.6: Τουρνουά Β: Μέσο σκορ στα 60 παιχνίδια και ίσο περιορισμό χωρητικότητας $C = C^{MED}$.

Κατάταξη	Πράκτορας	Μέσο σκορ ($)
1	Mertacor+NEAR	56,805
2	MertacorA($\alpha = 0.33$)	56,754
3	TacTex	55,219
4	Schlemazl	54,889
5	QuacTac	53,205
6	epflagent	40,376
7	Merlion	37,328
8	Wayne	37,160

πως μπορεί να χρησιμοποιηθεί σε προβλήματα ενισχυτικής μάθησης για επιλογή μίας ενέργειας με στόχο το μεγαλύτερο άμεσο κέρδος με θετικά αποτελέσματα.

Κεφάλαιο 9

Εφαρμογή 3: Παίκτης πόκερ

Το παιχνίδι του πόκερ αποτελεί ένα ιδανικό πρόβλημα αναφοράς για τη δημιουργία μηχανισμών λήψεων αποφάσεων στον πραγματικό κόσμο, καθώς είναι ένα παιχνίδι μη πλήρους πληροφόρησης (imperfect/partial information), στοχαστικό, με πολλούς πράκτορες (multi-player/multi-agent), στο οποίο οι πράκτορες πρέπει να κάνουν διαχείριση ρίσκου (risk management) και να αναγνωρίσουν τακτικές παιχνιδιού και εξαπατήσεις του αντιπάλου [BPSS98].

Η παραλλαγή με την οποία θα ασχοληθούμε είναι το *limit heads-up Texas Hold'em*, που είναι και η πιο απλή και χρησιμοποιείται από τους περισσότερους ερευνητές της περιοχής λόγω της μικρότερης δυνατής πολυπλοκότητας από τις υπόλοιπες παραλλαγές του είδους. Στην πραγματικότητα, η παραλλαγή no-limit με περισσότερους από δύο παίκτες είναι η πιο δημοφιλής. Ο όρος *heads-up* αντικατοπτρίζει το γεγονός ότι αναμετρώνται δύο παίκτες, ενώ ο όρος *limit* αφορά στον περιορισμό στα ποσά του πονταρίσματος, μειώνοντας σημαντικά και τον αριθμό επιτρεπτών ενεργειών των πρακτόρων.

Στόχος της παρούσας εφαρμογής είναι η ανάπτυξη μιας μικτής στρατηγικής, η οποία θα επιλέγει με τυχαίο τρόπο μια ενέργεια από ένα σετ δεδομένων ενεργειών με βάση μια πιθανοτική κατανομή [OR94]. Η εύρεση της βέλτιστης, παιχνιδοθεωρητικής, μικτής στρατηγικής δεν είναι εύκολη υπόθεση, καθώς η συγκεκριμένη απλή παραλλαγή έχει μέγεθος $O(10^{18})$. Κατά συνέπεια, δεν είναι δυνατή (τουλάχιστον όχι ακόμα), η εύρεση μιας βέλτιστης, θεωρητικά, στρατηγικής. Στόχος των ερευνητών είναι η δημιουργία αφαιρετικών μοντέλων που θα μειώσουν το μέγεθος του παιχνιδιού, δίχως ωστόσο να χάνονται χαρακτηριστικά του. Μια από τις πρώτες επιτυχημένες προσεγγίσεις στη συγκεκριμένη παραλλαγή του πόκερ είναι ο πράκτορας PsOpti του Πανεπιστημίου της Alberta [BBD+03]. Ο παίκτης PsOpti παίζει με βέλτιστη μικτή στρατηγική σε ένα μειωμένο όμως χώρο της τάξης $O(10^7)$. Γενικότερα, στόχος όλων αυτών των προσεγγίσεων είναι η εύρεση μίας βέλτιστης στρατηγικής, στο μειωμένο, όπως προαναφέρθηκε χώρο του παιχνιδιού, η οποία θα μεγιστοποιεί το κέρδος και θα ελαχιστοποιεί τη ζημιά απέναντι σε οποιαδήποτε στρατηγική του αντιπάλου. Με άλλα λόγια παίζουν με βάση την ισορροπία κατά Nash (Nash equilibrium). Εξαιτίας των προσεγγίσεων που γίνονται, οι στρατηγικές

αυτές ονομάζονται και ψευδο-βέλτιστες (pseudo-optimal). Επιπλέον, επειδή δε μοντελοποιούν τη στρατηγική του αντιπάλου, δεν εγγυώνται το μέγιστο κέρδος, απλά ότι σε άπειρο πλήθος παιχνιδιών δε θα χάσουν. Η στρατηγική που προκύπτει είναι τόσο καλή όσο και οι προσεγγίσεις που γίνονται. Κατά συνέπεια, μία αδυναμία που μπορεί να προκύψει να μπορεί να την εκμεταλλεύεται ο αντίπαλος μονίμως, αφού τέτοιου είδους στρατηγικές είναι στατικές. Επίσης, όπως προαναφέραμε, οι ψευδο-βέλτιστες στρατηγικές δεν εκμεταλλεύεται αντίστοιχα πιθανές αδυναμίες του αντιπάλου. Περισσότερο βελτιωμένες εκδόσεις του PsOpti, όπως το Polaris, έχουν καταφέρει να νικήσουν τον άνθρωπο στη συγκεκριμένη παραλλαγή του παιχνιδιού [Joh07].

Η μεθοδολογία που ακολουθείται στη συγκεκριμένη εφαρμογή είναι η κατασκευή ενός χαρακτηριστικού διανύσματος, το οποίο μέσω αφαίρεσης περιέχει τα βασικά χαρακτηριστικά του παιχνιδιού και αποτελεί την είσοδο σε ένα ΔΗΚ. Το χαρακτηριστικό διάνυσμα έχει σκοπό την απλοποίηση της πολυπλοκότητας του προβλήματος. Το ΔΗΚ αποτελεί το μετασχηματισμό του χαρακτηριστικού διανύσματος σε άλλα μη γραμμικά, χρονικά χαρακτηριστικά των οποίων ο γραμμικός συνδυασμός δίνει τις αξίες των τριών δυνατών ενεργειών που έχει ο πράκτορας στην παραλλαγή που εξετάζεται, δηλαδή $check/call$, $bet/raise$ και $fold$. Οι αξίες μετασχηματίζονται σε πιθανότητες και ορίζουν τη μικτή στρατηγική σε κάθε γύρο της παρτίδας. Τα βάρη του γραμμικού συνδυασμού βρίσκονται με αλγόριθμο μάθησης από δεδομένα παιχνιδιών. Πρέπει να σημειωθεί ότι αυτή η εφαρμογή αποτελεί μελέτη ικανότητας τόσο των χαρακτηριστικών, όσο και των ΔΗΚ να διαχειριστούν την πολυπλοκότητα του παιχνιδιού και να δημιουργήσουν εκ νέου μη γραμμικά, χρονικά χαρακτηριστικά. Εξαιτίας του μεγάλου αριθμού νευρώνων που απαιτούνται και της χρονοβόρας διαδικασίας μάθησης δεν χρησιμοποιήθηκε η μέθοδος NEAR για τη βελτιστοποίηση των τοπολογιών και βαρών. Αντίθετα, χρησιμοποιήθηκε η κλασική τυχαία δημιουργία των ΔΗΚ.

9.1 Η παραλλαγή limit heads-up Texas hold' em poker

Το παιχνίδι Texas hold'em χρησιμοποιεί την κλασική τράπουλα των πενήντα δύο φύλλων, η οποία περιέχει τέσσερα (4) χρώματα με δεκατρία (13) φύλλα το κάθε ένα. Τα χρώματα, και οι συμβολισμοί τους, είναι: τα σπαθιά [♣] (clubs), τα καρό [♢] (diamonds), τα μπαστούνια [♠] (spades) και οι κούπες [♡] (hearts). Κάθε χρώμα περιέχει τα παρακάτω φύλλα: άσο [A] (ace), παπά [K] (king), ντάμα [Q] (queen), βαλέ [J] (jack), δέκα [T] (ten), εννέα [9] (nine), οκτώ [8] (eight), επτά [7] (seven), έξι [6] (six), πέντε [5] (five), τέσσερα [4] (four), τρία [3] (three) και δύο [2] (two).

Στην παρούσα εφαρμογή και για τη συγκεκριμένη παραλλαγή, χρησιμοποιούνται οι προδιαγραφές του ετήσιου διεθνούς διαγωνισμού *Annual Computer Poker Competition*. Στο τραπέζι του πόκερ συμμετέχουν δύο παίκτες (heads-up). Εναλλάξ, σε κάθε παρτίδα οι παίκτες αποκτούν το ρόλο του *dealer*. Ο dealer "μιλάει" τελευταίος σε κάθε γύρο της παρτίδας, εκτός από τον πρώτο γύρο όπου "μιλάει"

πρώτος. Στην αρχή της παρτίδας μοιράζονται από δύο κρυφά φύλλα στους παίκτες (hole cards), οι οποίοι με τη σειρά τους τοποθετούν τα *blinds*, (υποχρεωτικά πονταρίσματα). Ο dealer πληρώνει το small blind, ποσό αντίστοιχο με το μισό του ελάχιστου πονταρίσματος, ενώ ο άλλος παίκτης πληρώνει το big blind, ποσό ίσο με το ελάχιστο ποντάρισμα. Οι προδιαγραφές που χρησιμοποιήθηκαν ορίζουν το big blind ίσο με αξία 10 και το small blind ίσο με αξία 5. Ο παίκτης/dealer που"μιλάει" πρώτος, έχει τη δυνατότητα τριών ενεργειών: (α) να κάνει call, ισοφαρίζοντας το πόσο που υπολείπεται του αντιπάλου (να βάλει ένα ποσό αξίας 5), (β) να κάνει raise, δηλαδή να ισοφαρίσει το ποσό που υπολείπεται και να αυξήσει το ποσό του τραπεζιού (pot) με το ελάχιστο ποσό πονταρίσματος (δηλαδή συνολική αξία 15) και (γ) να κάνει fold, χάνοντας την παρτίδα. Αν ο dealer κάνει call, ο άλλος παίκτης μπορεί να κάνει είτε check, δηλαδή να αφήσει τα πονταρίσματα ισοφαρισμένα και να περάσει η παρτίδα στον επόμενο γύρο, είτε να κάνει raise και να αυξήσει το πόσο του pot, αφήνοντας και πάλι τον dealer να "μιλήσει". Μόλις τα πονταρίσματα παραμείνουν ισοφαρισμένα ολοκληρώνεται ο γύρος ο οποίος ονομάζεται γύρος *pre-flop*. Στη συνέχεια η παρτίδα περνάει στο γύρο *flop*, όπου τρία κοινά φύλλα ανοίγουν στο τραπέζι (community cards) και ακολουθεί νέος γύρος πονταρίσματος. Έπειτα, το παιχνίδι συνεχίζει με το γύρο *turn*, όπου ανοίγει ακόμα ένα κοινό φύλλο και ακολουθεί ακόμα ένας γύρος πονταρίσματος και τέλος στο γύρο *river* ανοίγει το τελικό πέμπτο κοινό φύλλο και ένας ακόμα γύρος πονταρίσματος. Αν μετά το γύρο πονταρίσματος του river οι παίκτες παραμείνουν στο παιχνίδι τότε περνάμε στο showdown και ο παίκτης με την καλύτερη πεντάδα φύλλων που μπορεί να δημιουργηθεί από τα δύο κρυφά του χαρτιά και τα πέντε κοινά κερδίζει το pot, το συνολικό ποσό που έχει πονταριστεί σε όλους τους γύρους. Στο pre-flop και στο flop τα πονταρίσματα είναι όσο το big blind (10), ενώ στους γύρους turn και river τα πονταρίσματα είναι δύο φορές το big blind (20). Τέλος, στο pre-flop μπορεί να υπάρχει ένα bet, ένα raise και ένα re-raise, ενώ στους υπόλοιπους γύρους μπορεί να υπάρχει μέγιστο ένα bet, ένα raise και δύο re-raise στη συνέχεια. Ο Πίνακας 9.1 δείχνει την κατάταξη των φύλλων με παραδείγματα και τη συχνότητα εμφάνισης σε κάθε περίπτωση. Κάθε παίκτης θεωρούμε ότι έχει απεριόριστο αριθμό από μάρκες.

9.2 Ο πράκτορας TiltNet

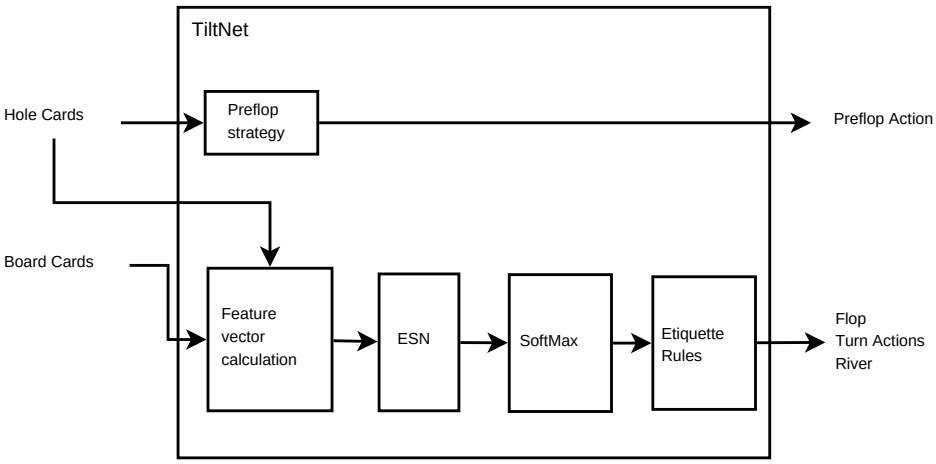
Ο πράκτορας πόκερ που αναπτύχθηκε έχει την ονομασία TiltNet και αποτελείται από πέντε τμήματα. Τα τμήματα καθώς και η ροή της πληροφορίας φαίνονται στο Σχήμα 9.1:

1. Μια βέλτιστη στρατηγική για το γύρο pre-flop, η οποία είναι ενσωματωμένη στον πράκτορα (Pre-flop strategy)

2. Τον υπολογισμό του χαρακτηριστικού διανύσματος της κατάστασης του πράκτορα (Feature vector calculation)

Πίνακας 9.1: Η κατάταξη των φύλλων, παραδείγματα και συχνότητα εμφάνισης σε φθίνουσα σειρά κατάταξης.

Τύπος Φύλλου	Παράδειγμα	Συχνότητα Εμφάνισης
Φλος Ρουαγιάλ (Royal flush)	A♠ K♠ Q♠ J♠ T♠	0.0032%
Κέντα χρώμα (Straight flush)	T♠ 9♠ 8♠ 7♠ 6♠	0.0279%
Καρέ (Four of a kind)	Q♡ Q♣ Q♠ Q♢ K♢	0.168%
Φουλ (Full house)	T♡ T♠ 2♣ 2♠ 2♢	2.60%
Χρώμα (Flush)	K♢ 9♢ 5♢ 4♢ 2♢	3.03%
Κέντα (Straight)	5♣ 4♢ 3♢ 2♡ A♠	4.62%
Τρία όμοια (Three of a kind)	8♢ 8♣ 8♠ 3♠ K♢	4.83%
Δύο ζευγάρια (Two pair)	T♢ T♠ 7♡ 7♠ 2♣	23.5%
Ζευγάρι (One pair)	A♠ A♢ 8♠ 7♠ 6♠	43.8%



Σχήμα 9.1: Τμηματική απεικόνιση του πράκτορα TiltNet.

3. Το ΔΗΚ που δέχεται ως είσοδο το χαρακτηριστικό διάνυσμα και δίνει ως έξοδο την αξία των τριών ενεργειών (ESN)

4. Το μετασχηματισμό των αξιών σε πιθανότητες μέσω του $softmax$ (Εξίσωση 2.1) και την επιλογή της ενέργειας με επιλογή ρουλέτας (SoftMax)

5. Τους κανόνες σωστής συμπεριφοράς, όπως αποδίδονται στον Αλγόριθμο 9.1 (Etiquette rules)

Στη συνέχεια της ενότητας παρουσιάζονται αναλυτικότερα τα τμήματα του πράκτορα.

9.2.1 Στρατηγική pre-flop

Για το γύρο του pre-flop, χρησιμοποιήσαμε μία έτοιμη στρατηγική που δίνεται με τη μορφή τριών πινάκων [She99]. Η συγκεκριμένη στρατηγική είναι βέλτιστη

Αλγόριθμος 9.1 Κανόνες σωστής συμπεριφοράς.

GOD-PRACTICE

Require: playerAction

```
1: if playerAction == FOLD and potIsBalanced then
2:     return CALL
3: else if playerAction == RAISE and not canRaise then
4:     return CALL
5: else
6:     return playerAction
7: end if
```

εφόσον δεν υπάρχουν άλλοι γύροι πονταρίσματος. Ανάλογα με τα κρυφά φύλλα του πράκτορα επιλέγεται μία από τις παρακάτω ενέργειες:

- F: Fold
- C: Call
- R1: Raise
- R2: Raise και ξανά Raise εάν ο αντίπαλος κάνει Raise επίσης
- R3: Raise, Reraise και ξανά Reraise εάν ο αντίπαλος κάνει Raise επίσης
- CR1: Call-Raise
- CR2: Call-Raise και Reraise εάν ο αντίπαλος κάνει Raise.

Για παράδειγμα για το φύλλο A♢ A♠ η σωστή ενέργεια είναι R3 σε οποιαδήποτε θέση και αν είμαστε, ενώ για το φύλλο K♢ 4♢ όταν ο παίκτης που το έχει είναι στο small blind η σωστή ενέργεια είναι R1. Η συγκεκριμένη pre-flop στρατηγική χρησιμοποιήθηκε και σε μία παραλλαγή του πράκτορα PsOpti με ικανοποιητικά αποτελέσματα [BBD+03].

9.2.2 Διάνυσμα χαρακτηριστικών

Το διάνυσμα χαρακτηριστικών είναι μια προσπάθεια να εκφραστεί η κατάσταση του παιχνιδιού μέσω περιορισμένου αριθμού τιμών. Για παράδειγμα, σε πρωτογενή μορφή δύο κέντες που αρχίζουν από 6, διαφορετικού χρώματος, θα έπρεπε να αναπαρίστανται διαφορετικά ως καταστάσεις τις οποίες παρατηρεί ο πράκτορας. Από την άλλη όμως η αξία τους είναι ίδια. Για να μειωθεί η διάσταση του διανύσματος εισόδου χρησιμοποιήθηκαν χαρακτηριστικά τα οποία αντικατοπτρίζουν με τις τιμές τους τα διακριτά φύλλα που παρατηρεί ο πράκτορας στο παιχνίδι και μπορούν να αντικαταστήσουν αυτές τις παρατηρήσεις. Άλλα χαρακτηριστικά έρχονται σε πρωτογενή μορφή, άλλα πρέπει να υπολογιστούν. Στον Πίνακα 9.2 αναφέρονται τα χαρακτηριστικά με τον τύπο δεδομένων τους. Στόχος είναι από

Πίνακας 9.2: Το χαρακτηριστικό διάνυσμα καταστάσεων που χρησιμοποιήθηκε.

ΑΑ	Χαρακτηριστικό	Τύπος
1	Hand Strength	numeric
2	Effective Hand Strength	numeric
3	Current round preflop	binary
4	Current round flop	binary
5	Current round turn	binary
6	Current round river	binary
7	Current pot	numeric
8	Pot odds	numeric
9	Dealer button	binary

τα χαρακτηριστικά αυτά, μέσω των ΔΗΚ, να εξαχθούν νέα μη γραμμικά, χρονικά χαρακτηριστικά που θα αντικατοπτρίζουν καλύτερα την πολυπλοκότητα του παιχνιδιού. Επίσης, μέσω της εκμάθησης των βαρών του γραμμικού συνδυασμού των χαρακτηριστικών του ΔΗΚ και της ιδιότητας της γενίκευσης, ο πράκτορας θα μπορεί θεωρητικά να είναι εφοδιασμένος με την κατάλληλη μικτή στρατηγική και για παρτίδες στις οποίες δεν έχει δει στο σετ εκμάθησης. Στη συνέχεια αναλύονται τα χαρακτηριστικά που επιλέχθηκαν.

Δύναμη φύλλου (Hand Strength)

Η δύναμη φύλλου (hand strength - HS) είναι η πιθανότητα να έχουμε καλύτερο φύλλο στο συγκεκριμένο γύρο από όλα τα τυχαία φύλλα που μπορεί να έχει ο αντίπαλος. Μπορεί να υπολογιστεί σε όλους τους γύρους της παρτίδας επακριβώς, εκτός από τον pre-flop. Για το γύρο του pre-flop χρησιμοποιήθηκε η φόρμουλα Chen [Kir00], ένας εμπειρικός κανόνας για τον υπολογισμό της δύναμης του φύλλου στο συγκεκριμένο γύρο. Ο Αλγόριθμος 9.2 παρουσιάζει τον υπολογισμό της δύναμης φύλλου, ενώ ο Αλγόριθμος 9.3 παρουσιάζει τον εμπειρικό κανόνα Chen. Το μειονέκτημα της χρήσης της δύναμης φύλλου είναι ότι δε λαμβάνει υπόψιν τα κοινά φύλλα που θα ανοίξουν στους επόμενους γύρους και επομένως δε λαμβάνεται υπόψιν η δυνατότητα βελτίωσης ή χειροτέρευσης του φύλλου. Στο σημείο αυτό θα πρέπει να αναφερθεί ότι η φόρμουλα Chen είναι ένας εύκολος τρόπος ο παίκτης να θυμάται τον πίνακα των Malmuth και Skalnsky ο οποίος χωρίζει τα ζεύγη κρυφών φύλλων σε 169 διαφορετικής δύναμης φύλλα [SM99].

Για παράδειγμα, αν το φύλλο είναι K♣ J♢ και τα κοινά φύλλα του flop ανοίξουν J♠ 8♢ 5♡ τότε υπάρχουν 1024 συνδυασμοί φύλλων του αντιπάλου που το φύλλο αυτό κερδίζει, 6 που φέρνει ισοπαλία και 51 συνδυασμοί που χάνει. Η δύναμη του φύλλου είναι 0.95 ή 95%, δηλαδή 95 στις 100 φορές το φύλλο αυτό κερδίζει. Συνεχίζοντας το παράδειγμα, αν τα κρυφά φύλλα είναι K♣ και J♢ τότε η φόρμουλα Chen δίνει ευριστική δύναμη φύλλου 7. Η φόρμουλα προσδίδει 8 πόντους για το φύλλο K♣ και αφαιρεί έναν για το κενό φύλλο που υπάρχει. Αυτό μεταφράζε-

ται σε κανονικοποιημένη δύναμη φύλλου 0.325 αφού σύμφωνα με τη φόρμουλα οι μέγιστοι πόντοι είναι 20 (A♠ A♡) και οι ελάχιστοι -1.5 (π.χ. 7♣ 2♢).

Αλγόριθμος 9.2 Αλγόριθμος υπολογισμού της δύναμης φύλλου.

HAND-STRENGTH

Require: holeCards, boardCards

```
ahead = 0
tied = 0
behind = 0
rank = rankHand(holeCards, boardCards)
for each case of opponent hole cards do
    oppRank = rankHand(oppHoleCards, boardCards)
    if rank > oppRank then
        ahead++
    else if rank == oppRank then
        tied++
    else
        behind++
    end if
end for
handStrength = (ahead + tied/2) / (ahead + tied + behind)
return handStrength
```

Ενεργός δύναμη φύλλου (Effective Hand Strength)

Η έλλειψη διορατικότητας της δύναμης φύλλου έκανε απαραίτητη την εισαγωγή ενός ακόμα χαρακτηριστικού, της ενεργού δύναμης φύλλου (effective hand strength or EHS). Η ενεργός δύναμη του φύλλου συνδυάζει τόσο τη δύναμη του φύλλου, όσο και τη δυνατότητα του φύλλου. Η δυνατότητα του φύλλου (hand potential) μπορεί να είναι είτε θετική (positive potential or PP) είτε αρνητική (negative potential or NP). Το χαρακτηριστικό PP είναι η πιθανότητα το φύλλο μας να βελτιωθεί έναντι όλων των τυχαίων συνδυασμών των φύλλων του αντιπάλου, ενώ το χαρακτηριστικό NP είναι η πιθανότητα το φύλλο μας, που προηγείται (ahead), να ηττηθεί έναντι όλων των τυχαίων συνδυασμών των φύλλων του αντιπάλου (behind). Το PP υπολογίζεται σύμφωνα με την Εξίσωση 9.1, το NP σύμφωνα με την Εξίσωση 9.2 και το EHS σύμφωνα με την εξίσωση 9.3. Οι υπολογισμοί της δυνατότητας του φύλλου παρουσιάζονται στον Αλγόριθμο 9.4.

$$PP = \frac{N_{behind,ahead} + N_{behind,tied}/2 + N_{tied,ahead}/2}{N_{behind} + N_{tied}/2} \tag{9.1}$$

$$NP = \frac{N_{ahead,behind} + N_{ahead,tied}/2 + N_{tied,behind}/2}{N_{front} + N_{tied}/2} \tag{9.2}$$

Αλγόριθμος 9.3 Η φόρμουλα του Chen.

CHEN-FORMULA

Require: holeCards

```
strength = 0
rank1 = rank of 1st hole card
rank2 = rank of 2nd hole card
suit1 = suit of 1st hole card
suit2 = suit of 2nd hole card
maxRank = max(rank1, rank2)
if maxRank == A then
    strength = strength + 10
else if maxRank == K then
    strength = strength + 8
else if maxRank == Q then
    strength = strength + 7
else if maxRank == J then
    strength = strength + 6
else
    strength = strength + (maxRank+2)/2
end if
if rank1 == rank2 then
    minimum = 5
    if rank1 == 5 then
        minimum = 6
    end if
    strength = max(2 * strength, minimum)
end if
if suit1 == suit2 then
    strength = strength + 2
end if
gap = maxRank - min(rank1, rank2) -1
if gap == 0 then
    strength++
else if gap == 1 then
    strength–
else if gap == 2 then
    strength = strength - 2
else if gap == 3 then
    strength = strength - 4
else
    strength = strength - 5
end if
if gap == 1 and maxRank < Q then
    strength++
end if
return strength
```

Πίνακας 9.3: Παράδειγμα χαρακτηριστικού διανύσματος καταστάσεων.

ΑΑ	Χαρακτηριστικό	Τιμή
1	Hand Strength	0.578
2	Effective Hand Strength	0.445
3	Current round preflop	0
4	Current round flop	0
5	Current round turn	1
6	Current round river	0
7	Current pot	0.456
8	Pot odds	0.45
9	Dealer button	1

$$EHS = HS + (1 - HS) \cdot PP \tag{9.3}$$

Απόδοση ποτ (Pot Odds)

Η απόδοση pot υπολογίζεται σύμφωνα με την Εξίσωση 9.4.

$$\text{απόδοση pot} = \frac{\text{κόστος call}}{\text{ποσό pot} + \text{κόστος call}} \tag{9.4}$$

Η απόδοση pot βοηθάει τον παίκτη να αποφασίσει εάν τον συμφέρει να κάνει call ή όχι. Για παράδειγμα, αν η πιθανότητα να κερδίσει το παιχνίδι είναι μεγαλύτερη από την απόδοση του pot τότε δικαιολογείται να γίνει call.

Λοιπά χαρακτηριστικά

Τα υπόλοιπα χαρακτηριστικά δε χρειάζονται περαιτέρω υπολογισμούς και μπορούν να εξαχθούν κατευθείαν από τις παρατηρήσεις του πράκτορα. Σε αυτά συμπεριλαμβάνονται, ο γύρος του παιχνιδιού (4 είσοδοι - Πίνακας 9.2), το ποσό του pot καθώς και το αν ο πράκτορας έχει το ρόλο του dealer ή όχι.

9.2.3 Εκπαίδευση ΔΗΚ και επιλογή ενέργειας

Κάθε ΔΗΚ δέχεται ως είσοδο το χαρακτηριστικό διάνυσμα (για παράδειγμα ένα χαρακτηριστικό διάνυσμα παρουσιάζεται στον Πίνακα 9.3) και έχει δύο εξόδους που προβλέπουν την αξία των ενεργειών check/call και bet/raise αντίστοιχα. Σκοπός της διαδικασίας εκμάθησης είναι η πρόβλεψη της αξίας που έχει για τον πράκτορα κάθε ενέργεια δεδομένου του χαρακτηριστικού διανύσματος. Το χαρακτηριστικό διάνυσμα και η αξία ήταν κανονικοποιημένα στο διάστημα $[0, 1]$. Η αξία της ενέργειας fold είναι η κανονικοποιημένη ζημιά του να κάνει ο πρακτορας fold τη δεδομένη χρονική στιγμή.

Αλγόριθμος 9.4 Αλγόριθμος υπολογισμού της δυνατότητας φύλλου.

```
POTENTIAL
Require: holeCards, boardCards
 1: rankHand = rank(holeCards,boardCards)
 2: for all Possible opponent hole cards do
 3:    oppRankHand = rank(oppHoleCards,boardCards)
 4:    if rankHand > oppRankHand then
 5:       index = ahead
 6:    else if ourRank == oppRankHand then
 7:       index = tied
 8:    else
 9:       index = behind
10:    end if
11:    HPTotal[index]++
12:    for all cards in turn do
13:       for all cards in river do
14:          boardCards += turn + river
15:          rankHand = rank(holeCards,boardCards)
16:          oppRankHand = rank(oppHoleCards,boardCards)
17:          if rankHand > oppRankHand then
18:             HP[index][ahead]++
19:          else if ourRank == oppRankHand then
20:             HP[index][tied]++
21:          else
22:             HP[index][behind]++
23:          end if
24:       end for
25:    end for
26: end for
27: {PP: positive potential}
28: PP = (HP[behind][ahead] + HP[behind][tied]/2 + HP[tied][ahead]/2) /
29: (HPTotal[behind] + HPTotal[tied]/2)
30: {NP: negative potential}
31: NP = (HP[ahead][behind] + HP[tied][behind]/2 + HP[ahead][tied]/2) /
32: (HPTotal[ahead] + HPTotal[tied]/2)
33: return PP,NP
```

Για την εκπαίδευση του ΔΗΚ χρησιμοποιήθηκαν τα παιχνίδια του Annual Computer Poker Competition 2011. Πιο συγκεκριμένα, επιλέχθηκαν οι παρτίδες (hands) οι οποίες έφτασαν σε showdown, δηλαδή αποκαλύφθηκαν τα χαρτιά των δύο αντιπάλων, το pot έφτασε στην τελική του τιμή, την οποία προσπαθούμε να προβλέψουμε, ενώ κανένας παίκτης δεν έκανε fold. Για κάθε παρτίδα δημιουργήθηκαν δύο σειρές δεδομένων, μία για κάθε παίκτη. Για κάθε ενέργεια του παίκτη υπολογίστηκε το χαρακτηριστικό διάνυσμα, καταχωρήθηκε η ενέργειά του, καθώς και η τελική τιμή που κέρδισε ή έχασε ο πράκτορας.

Εξαιτίας του όγκου των δεδομένων, αλλά και του μεγάλου αριθμού νευρώνων που χρησιμοποιήθηκαν για το ταμιευτήριο, ήταν αδύνατη η χρήση μεθόδων γραμμικής άλγεβρας που θα υπολόγιζαν απευθείας τα βάρη του ΔΗΚ. Επομένως έπρεπε να χρησιμοποιηθεί κάποιος επαναληπτικός αλγόριθμος, όπως για παράδειγμα η κατάβαση πλαγιάς.

Μετά την πρόβλεψη των αξιών χρησιμοποιήθηκε η μεθοδολογία $softmax$ για τον υπολογισμό των πιθανοτήτων (Εξίσωση 2.1). Από τις τρεις πιθανότητες, η ενέργεια που επιλέγεται προκύπτει με επιλογή ρουλέτας.

9.3 Πειράματα και αποτελέσματα

Πραγματοποιήθηκαν δύο σειρές πειραμάτων. Στο πρώτο πείραμα τα ΔΗΚ που χρησιμοποιήθηκαν είχαν 100 νευρώνες ταμιευτηρίου (TiltNet-100), ενώ στο δεύτερο 200 (TiltNet-200). Η φασματική ακτίνα ορίστηκε 0.85 και η πυκνότητα των ταμιευτηρίων στο 15% και στα δύο πειράματα. Για το πρώτο πείραμα, το σύνολο των βαρών ήταν $N \times K + 0.15 \times N \times N + L \times (N + K) = 900 + 1500 + 218 = 2,618$, από τα οποία τα 218 εκπαιδεύονται, ενώ για το δεύτερο $1800 + 6000 + 418 = 7,218$, από τα οποία τα 418 εκπαιδεύονται. Συνολικά συγκεντρώθηκαν $800,000$ σετ δεδομένων όπου κάθε ΔΗΚ εκπαιδεύτηκε στο συνολικό σετ δεδομένων 30 φορές επιλέγοντας με τυχαίο τρόπο τη σειρά εμφάνισης των σετ. Ο ρυθμός εκπαίδευσης (learning rate) α τέθηκε ίσος με 10^{-5}. Συνολικά δημιουργήθηκαν 50 ΔΗΚ σε κάθε πείραμα, τα οποία, αφού εκπαιδεύτηκαν, αξιολογήθηκαν απέναντι σε πέντε πράκτορες αξιολόγησης που παρουσιάζονται παρακάτω:

1. **Always Call** Ο παίκτης αυτός κάνει call ή check ανάλογα με την περίπτωση.

2. **Always Raise** Ο παίκτης αυτός κάνει συνέχεια bet ή raise μέχρι να εξαντληθεί το όριο που ορίζουν οι κανόνες του παιχνιδιού.

3. **Call or Raise** Με πιθανότητες 50-50 ο παίκτης αυτός επιλέγει ή check/call ή bet/raise.

4. **Heuristic** Παίζει με έναν πολύ απλό ευριστικό κανόνα που δίνεται στον Αλγόριθμο 9.5 [dLPC99].

5. **Random** Παίζει τυχαία μία από τις επιτρεπόμενες κινήσεις. Είναι ο πιο αδύναμος από τους παίκτες αξιολόγησης.

Πίνακας 9.4: Επιδόσεις του TiltNet-100 απέναντι στους 5 παίκτες αξιολόγησης.

Αντίπαλος	μ (sb/h)	Νίκες	Ήττες
Always Call	0.988	48	2
Always Raise	0.192	27	23
Call or Raise	0.574	38	12
Random	2.054	50	0
Heuristic	0.200	36	14

Και οι πέντε πράκτορες αξιολόγησης εφαρμόζουν τους κανόνες καλής συμπεριφοράς.

Αλγόριθμος 9.5 Παίκτης ευριστικού κανόνα όπου η μικτή στρατηγική ορίζεται ανάλογα με την τρέχουσα ενεργή δύναμη φύλλου του παίκτη.

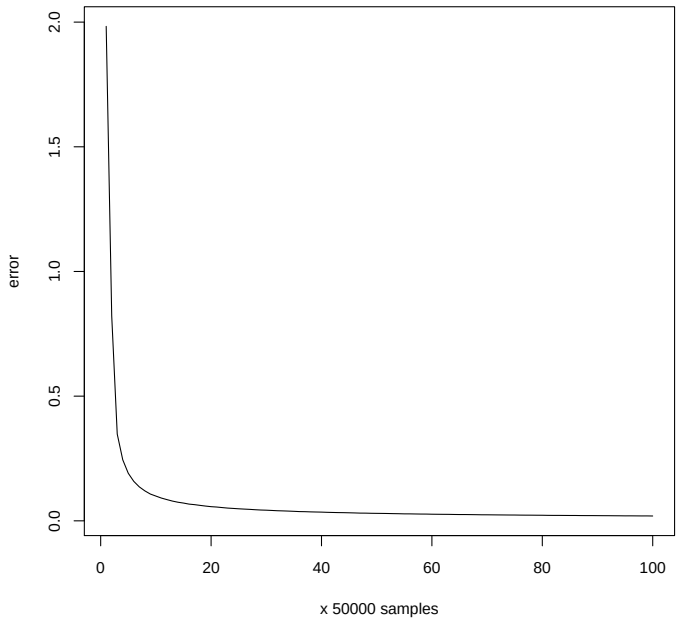
HEURISTIC-PLAYER

1: **if** $EHS > 0.75$ **then**
2: mixed $\leftarrow \{0.01, 0.24, 0.75\}$
3: **else if** $EHS \geq 0.5$ **then**
4: mixed $\leftarrow \{0.19, 0.8, 0.01\}$
5: **else**
6: mixed $\leftarrow \{0.5, 0.49, 0.01\}$
7: **end if**
8: **return** mixed

Στο Σχήμα 9.2 παρουσιάζεται η καμπύλη του σφάλματος πρόβλεψης της αξίας μίας ενέργειας ενός από τα πενήντα ΔΗΚ.

Συνολικά εντοπίστηκαν 16 ΔΗΚ που κατάφεραν να νικήσουν και τους 5 αντιπάλους αξιολόγησης στο πρώτο πείραμα και 20 στο δεύτερο. Οι Πίνακες 9.4 και 9.5 δίνουν τους μέσους όρους απόδοσης (μ) sb/h (small blinds per hand) για τα 50 ΔΗΚ κάθε πειράματος καθώς και τον αριθμό των θετικών και αρνητικών επιδόσεων απέναντι στον εκάστοτε αντίπαλο. Από κάθε πείραμα επιλέχθηκε το καλύτερο ΔΗΚ αυτό που κατάφερε να νικήσει και τους πέντε αντιπάλους και είχε μεγαλύτερο μέσο όρο απόδοσης. Αυτά επιλέχθηκαν να παίξουν και απέναντι στις εκδόσεις των προγραμμάτων `Sparbot` [BBD+03] και `PokiBot` [Dav02] όπως έχουν υλοποιηθεί στην πλατφόρμα `Poker Academy`. Τα αποτελέσματα συνοψίζονται στον Πίνακα 9.6.

Τα κυριότερα συμπεράσματα είναι ότι η παραπάνω μεθοδολογία δημιουργίας μικτών στρατηγικών μέσω ΔΗΚ και μάθησης μπορεί να παράξει δίκτυα τα οποία αποδεικνύονται καλύτερα από στατικούς και ευριστικούς αντιπάλους. Ωστόσο, παρόλο που η διαδικασία εκπαίδευσης των δικτύων ήταν πανομοιότυπη και για τα 50 ΔΗΚ κάθε πειράματος, δεν κατάφερε να επιφέρει τα ίδια αποτελέσματα και τις 50 φορές, κάτι που οδηγεί στο συμπέρασμα ότι η δημιουργία τοπολογιών ΔΗΚ με τυχαίο τρόπο δεν είναι πάντα η καλύτερη λύση. Η παρατήρηση αυτή ενισχύει την



Σχήμα 9.2: Καμπύλη σφάλματος εκπαίδευσης ΔΗΚ για την εφαρμογή του πόκερ.

Πίνακας 9.5: Επιδόσεις του TiltNet-200 απέναντι στους 5 παίκτες αξιολόγησης.

Αντίπαλος	μ (sb/h)	Νίκες	Ήττες
Always Call	0.696	50	0
Always Raise	0.595	35	15
Call or Raise	0.576	38	12
Random	2.223	50	0
Heuristic	-0.036	28	22

Πίνακας 9.6: Οι επιδόσεις των καλύτερων δικτύων TiltNet-100 και TiltNet-200 απέναντι σε 7 παίκτες αξιολόγησης.

Αντίπαλος	TiltNet-100	TiltNet-200
Always Call	**1.298**	1.021
Always Raise	2.582	**4.405**
Call or Raise	1.632	**1.970**
Random	**3.411**	3.208
Heuristic	**0.142**	0.098
PokiBot	-1.100	**-0.960**
Sparbot	-0.570	**-0.380**

παρουσία της NEAR γενικότερα, χωρίς να ακυρώνει ταυτόχρονα τα ΔΗΚ, αφού βρέθηκαν και αποδοτικές τοπολογίες. Φυσικά για τον υπερκερασμό των `Sparbot` και `PokiBot` πέρα της βελτίωσης της μικτής στρατηγικής μέσω καλύτερων ΔΗΚ και εκπαίδευσης με χρήση περισσότερων δεδομένων, είναι αναγκαία και η ύπαρξη αλγορίθμων μοντελοποίησης του αντιπάλου (opponent modeling).

9.4 Σύνοψη

Στο παρόν κεφάλαιο παρουσιάστηκε η σχεδίαση ενός αυτόνομου πράκτορα για το παιχνίδι του πόκερ. Ο πράκτορας συνδυάζει διαφορετικές τεχνικές προκειμένου να αναπτύξει τη συνολική του στρατηγική. Πιο συγκεκριμένα, χρησιμοποιήθηκαν: α) η βέλτιστη στρατηγική για τον γύρο pre-flop β) η δημιουργία ενός χαρακτηριστικού διανύσματος για την περιγραφή της κατάστασης της παρτίδας με μικρό αριθμό μεταβλητών και γ) ένα ΔΗΚ το οποίο εκπαιδεύτηκε από δεδομένα προηγούμενων παιχνιδιών και μοντελοποίηση μια μικτή στρατηγική για τους υπόλοιπους γύρους, μετασχηματίζοντας έτσι τον τεράστιο αριθμό διαφορετικών καταστάσεων του πόκερ σε μια μικτή στρατηγική. Ουσιαστικά η παρούσα εφαρμογή αποτελεί μια μελέτη εφαρμογής των ΔΗΚ στο παιχνίδι του πόκερ με θετικά αποτελέσματα.

Κεφάλαιο 10

Συμπεράσματα και Μελλοντικές Επεκτάσεις

Στο τελευταίο κεφάλαιο της διατριβής επιχειρείται μια ανακεφαλαίωση του περιεχομένου της και μια σύνοψη των συμπερασμάτων που εξήχθησαν. Τέλος, παρατίθεται μια σειρά μελλοντικών επεκτάσεων όσον αφορά στους μηχανισμούς που αναπτύχθηκαν όσο και στις εφαρμογές που παρουσιάστηκαν.

10.1 Ανακεφαλαίωση

Στόχος της διατριβής ήταν η ανάπτυξη ενός μηχανισμού προσαρμογής συναρτήσεων προσέγγισης που χρειάζονται για την αναπαράσταση μοντέλων και πολιτικών σε προβλήματα εποπτευόμενης μάθησης και ενισχυτικής μάθησης, αντίστοιχα. Οι συναρτήσεις προσέγγισης προσφέρουν την ιδιότητα της γενίκευσης σε δεδομένα και εμπειρίες τις οποίες δεν έχει συναντήσει ο πράκτορας κατά τη διάρκεια της εκπαίδευσης. Για την υλοποίηση αυτού του μηχανισμού χρησιμοποιήθηκαν συνδυαστικά διάφορες τεχνικές και αλγόριθμοι. Ως συνάρτηση προσαρμογής επιλέχθηκαν τα δίκτυα ηχωικών καταστάσεων, ενώ για το μέρος της προσαρμογής χρησιμοποιήθηκαν τόσο αλγόριθμοι μάθησης (βελτιστοποίησης των βαρών ενός δικτύου ανάλογα με τα δεδομένα και τη συνάρτηση κέρδους), όσο και αλγόριθμοι εξελικτικής υπολογιστικής. Τα ΔΗΚ επιλέχθηκαν για την ικανότητά που έχουν να μοντελοποιούν τόσο μη γραμμικά περιβάλλοντα όσο και μη Μαρκοβιανά σήματα, καθώς και να μαθαίνουν με γραμμικό τρόπο, καθιστώντας τα έτσι αρκετά χρήσιμα για τη επίλυση πολλών προβλημάτων επιβλεπόμενης και ενισχυτικής μάθησης. Όσον αφορά στους αλγόριθμους ενισχυτικής μάθησης και νευροεξέλιξης, η μέθοδος NEAT χρησιμοποιήθηκε ως αλγόριθμος μετα-αναζήτησης και προσαρμόστηκε στις ανάγκες των ΔΗΚ, με την ονομασία NeuroEvolution of Augmented Reservoirs ή NEAR.

Για τη διασφάλιση της ποιοτικής και ποσοτικής απόδοσης τόσο των ΔΗΚ όσο και της NEAR σε προβλήματα ενισχυτικής μάθησης, οι μηχανισμοί που αναπτύχθηκαν εφαρμόστηκαν σε μια πλειάδα από τεστ αξιολόγησης και η απόδοσή τους

συγκρίθηκε με παρόμοιους αλγορίθμους. Για τη βελτίωση της απόδοσης των ΔΗΚ και συγκεκριμένα της NEAR, διερευνήθηκε επίσης κατά πόσο η μεταφορά μάθησης θα βοηθούσε τόσο στην ασυμπτωτική συμπεριφορά του αλγορίθμου όσο και στη βελτίωση της ταχύτητας σύγκλισης, όταν υπήρχε μεταφορά ταμιευτηρίων από μια πηγαία εργασία στην εργασία στόχο.

Εκτός από τα τεστ αξιολόγησης, η μέθοδος NEAR εφαρμόστηκε και σε διάφορες εφαρμογές αυτόνομων πρακτόρων λογισμικού αλλά και σε προβλήματα πρόβλεψης χρονοσειρών. Πιο συγκεκριμένα, ο αλγόριθμος προέβλεψε τις επόμενες χρονοσειρές σε τρία προβλήματα δυναμικών συστημάτων, στη σειρά Mackey-Glass, στη σειρά Multiple Superimposed Oscillator (MSO) και στη σειρά Lorentz, καθώς και σε ένα κλασικό πρόβλημα πρόβλεψης ενεργειακού φορτίου. Στη συνέχεια, χρησιμοποιήθηκε ως βασικό μέρος των πολιτικών αυτόνομων πρακτόρων λογισμικού. Στην πρώτη εφαρμογή χρησιμοποιήθηκε ως μοντέλο πολιτικής πωλήσεων και ελέγχου της ρυθμαπόδοσης ενός εργοστασίου κατασκευής υπολογιστών. Στη δεύτερη εφαρμογή χρησιμοποιήθηκε ως μοντέλο πολιτικής της στρατηγικής πλειοδοσίας σε δημοπρασίες διαδικτυακών διαφημίσεων σε έναν πράκτορα διαχείρισης της διαφημιστικής καμπάνιας ενός διαδικτυακού καταστήματος. Τέλος, στην τρίτη εφαρμογή, χρησιμοποιήθηκαν τα ΔΗΚ και αλγόριθμοι ενισχυτικής μάθησης για την εύρεση μικτών στρατηγικών σε μια παραλλαγή του παιχνιδιού πόκερ και πιο συγκεκριμένα του limit heads-up Texas Hold'em poker.

10.2 Συμπεράσματα

Τα αποτελέσματα των πειραμάτων στα τεστ αξιολόγησης ενισχυτικής μάθησης αλλά και στα προβλήματα χρονοσειρών υποδεικνύουν ότι η μέθοδος NEAR είναι ένας μηχανισμός ισοδύναμος, αν όχι καλύτερος, από αντίστοιχους αλγορίθμους αιχμής και αναφοράς με βάση τα εκάστοτε κριτήρια επίδοσης. Επίσης, η δομή με αναδράσεις που υποστηρίζουν τα ΔΗΚ μπορεί να χειριστεί τόσο μη γραμμικά όσο και μη-Μαρκοβιανά σήματα κατάστασης αρκετά ικανοποιητικά, ενώ παράλληλα μπορεί να μαθαίνει με κλασικές μεθόδους βελτιστοποίησης, όπως η μέθοδος ελαχίστων τετραγώνων και η κατάβαση πλαγιάς.

Όσον αφορά στη μέθοδο NEAR, τα συμπεράσματα που μπορούν να εξαχθούν είναι αρκετά και σημαντικά. Αρχικά, η διασταύρωση κατά τη διάρκεια της νευροεξέλιξης, αν γίνεται με δικαιολογημένο τρόπο, μπορεί να οδηγήσει σε πιο εύρωστες και αποδοτικές λύσεις, σε σχέση με τη νευροεξέλιξη που βασίζεται μόνο στην ύπαρξη μεταλλάξεων. Ακόμη, οι μέθοδοι αναζήτησης πολιτικών υπερισχύουν έναντι των μεθόδων μάθησης χρονικών διαφορών, όταν μία προσέγγιση στην έξοδο είναι αρκετή για να προσδώσει τη σωστή πολιτική στον πράκτορα. Ωστόσο, όταν ακριβές λύσεις είναι αναγκαίες, η νευροεξέλιξη πρέπει να συζευχθεί και με τεχνικές μάθησης. Η μοναδική περίπτωση στην οποία η μεθοδολογία NEAR δεν κατάφερε να αντεπεξέλθει ικανοποιητικά, είναι στο πρόβλημα MSO. Το γεγονός αυτό, ωστόσο, μπορεί να αποδοθεί στην αδυναμία των ΔΗΚ να προβλέψουν επακριβώς πολλούς επικαλυπτώμενους ταλαντωτές. Για αυτή, λοιπόν, την

τάξη των προβλημάτων, η μέθοδος NEAR πρέπει να διαμορφωθεί ώστε να εξελίσσει δίκτυα με χαλαρά συνδεδεμένες συστάδες σφιχτά συνδεδεμένων κόμβων (loosely-coupled clusters of closely-coupled nodes). Από την άλλη, αποδείχθηκε πειραματικά σε πολλά προβλήματα ότι τα ΔΗΚ είναι καλές συναρτήσεις προσέγγισης για προβλήματα ενισχυτικής μάθησης, αν η τοπολογία τους βελτιστοποιηθεί μέσω κάποιας μεθόδου όπως είναι οι μέθοδοι νευροεξέλιξης. Θα πρέπει επίσης να αναφερθεί ότι στην περίπτωση ιδιαιτέρως μεγάλων ΔΗΚ, η εκπαίδευση και η εξέλιξή τους απαιτεί αυξημένες δυνατότητες υπολογιστικής ισχύος και μνήμης, όπως στο παράδειγμα στην εφαρμογή του παίκτη Πόκερ.

Στο κομμάτι των εφαρμογών, τα ΔΗΚ γενικότερα και η NEAR ειδικότερα, προσέφεραν λύσεις στο πρόβλημα της γενίκευσης καθώς μετά την εκπαίδευσή τους χρησιμοποιώντας ιστορικά δεδομένα, κατάφεραν να βελτιώσουν τη συμπεριφορά των πρακτόρων σε επεισόδια και δεδομένα τα οποία δεν είχαν προηγουμένως αντιμετωπίσει. Τέλος, η επέκταση της NEAR με δυνατότητες ενισχυτικής μάθησης κατάφερε να βελτιώσει τόσο την ασυμπτωτική του συμπεριφορά, όσο και την ταχύτητα εύρεσης καλών λύσεων.

10.3 Μελλοντικές Επεκτάσεις

Συνολικά το έργο που παρουσιάστηκε στη διατριβή είναι μια βασική προσέγγιση σε μια εύρωστη περιοχή έρευνας και ανάπτυξης εφαρμογών. Προφανώς παραμένουν πτυχές της μεθοδολογίας που επιδέχονται περαιτέρω ανάλυσης.

10.3.1 Βελτιώσεις στη μέθοδο NEAR

Πιο συγκεκριμένα, στόχος είναι η διερεύνηση της βελτίωσης της NEAR στο μέλλον προς τρεις κατευθύνσεις:

1. Το κομμάτι της νευροεξέλιξης της NEAR μπορεί να αξιολογηθεί σε μεγαλύτερο βάθος εφαρμόζοντας τον αλγόριθμο σε ολοένα δυσκολότερα προβλήματα [Pro05] και αξιολογώντας την αποτελεσματικότητα και τις δυνατότητες κλιμάκωσής του.

2. Το μοντέλο των ΔΗΚ μπορεί να επεκταθεί με ιδιότητες όπως οι νευρώνες αθροιστικής διαρροής (leaky integrator neurons), η συνδεσμολογία ανάδρασης (feedback connectivity), η εγγενής πλαστικότητα (intrinsic plasticity), οι πολλαπλές συναρτήσεις εξόδου (mutliple readout functions), ιεραρχικές ή τμηματικές αρχιτεκτονικές ΔΗΚ που θα εξελίσσονται μέσω παραλλαγών της NEAR [Jae02, LJ09].

3. Περισσότερη μελέτη θα απαιτηθεί για να αποκαλυφθούν οι μέθοδοι που συμπεριφέρονται καλύτερα στην προσαρμογή των βαρών εξόδου. Στην παρούσα διατριβή μελετήθηκαν η μέθοδος κατάβασης πλαγιάς, η μέθοδος των ελαχίστων τετραγώνων και των αναδρομικών ελαχίστων τετραγώνων (recursive least squares - RLS-TD) [XgHH02] και η μέθοδος της μετάλλαξης μέσω

διαταράξεων των βαρών. Σε άλλες περιπτώσεις θα μπορούσαν να δοκιμαστούν η απλή ανάβαση λόφου (simple hill-climbing) και η Γκαουσιανή μετάλλαξη, αλλά και πιο πολύπλοκοι αλγόριθμοι όπως ο PEGASUS [NJ00], ο LSPI [LP03], ο iLSTD [GBS06], ο GQ(λ) [MS10], ο CMA-ES [HO01].

Σε άλλες επεκτάσεις θα μπορούσε να μελετηθεί η συμπεριφορά της NEAR σε προβλήματα **συνεξέλιξης (*co-evolution*)** [RB97], όπου οι πολιτικές θα συνεξελίσσονται μέσω της NEAR σε δύο πληθυσμούς, τον ξενιστή (host) και τον παρασιτικό (parasite), με εφαρμογές, για παράδειγμα, σε παιχνίδια [PB98] ή σε άλλα ανταγωνιστικά πολυπρακτορικά περιβάλλοντα [SM04]. Επίσης, επιπλέον έρευνα απαιτείται για την αναγνώριση της συμπεριφοράς της NEAR σε **μη στάσιμα περιβάλλοντα (non-stationary)** και για τη μελέτη της ταχύτητας προσαρμογής όταν οι συνθήκες του προβλήματος αλλάζουν. Πιθανόν στην περίπτωση αυτή, θα μπορούσαν να εξελιχθούν δίκτυα μέσω Δαρβινικής εξέλιξης ώστε στη συνέχεια η μάθηση να μεταβάλλει τα βάρη, εν δράσει, στις παρούσες συνθήκες. Τέλος, η ιδέα του **συμβολικού λογισμού και της μάθησης σε νευρωνικά συστήματα (neurosymbolic learning and reasoning)** θα μπορούσε να διερευνηθεί σε σχέση με τα ΔΗΚ και τη μέθοδο NEAR.

10.3.2 Μεταφορά Μάθησης

Σχετικά με τη **μεταφορά μάθησης**, η μελέτη θα πρέπει να επεκταθεί σε ακόμη δυσκολότερα προβλήματα, όπως για παράδειγμα στο πρόβλημα keepaway [TWS07], γνωστό και ως κορόϊδο, στο οποίο πρέπει να βρεθεί μια πολιτική της ομάδας που κατέχει την μπάλα με σκοπό να τη διατηρήσει περισσότερο στην κατοχή της απέναντι σε μια άλλη ομάδα που προσπαθεί να την αρπάξει. Η μεταφορά μάθησης πραγματοποιείται πηγαίνοντας από 2-εναντίον-3 σε 3-εναντίον-4, 4-εναντίον-5 κ.ο.κ. Με τον τρόπο αυτό θα μπορέσει να διαπιστωθεί η εικασία ότι η μεταφορά μάθησης γίνεται χρησιμότερη καθώς δυσκολεύει το πρόβλημα. Σε ακόμα δυσκολότερη περίπτωση η μεταφορά μάθησης γίνεται μεταξύ διαφορετικών περιοχών, όπως η χρησιμοποίηση της περιοχής κονταροχτυπήματος ιπποτών (knight joust) ως πηγαία εργασία και του keepaway ως εργασία στόχου [TS07]. Τέλος, ακόμα ένα σημείο που θα άξιζε περαιτέρω έρευνα, είναι η ευαισθησία της μεθόδου μεταφοράς σε συνθήκες παρουσίας θορύβου τόσο στην πηγή όσο και στο στόχο.

10.3.3 Η μέθοδος NEAR και εφαρμογές αυτόνομων πρακτόρων

Οι βελτιώσεις στη μέθοδο NEAR θα βελτιώσουν και την επίδοσή του στην εφαρμογή στρατηγικής πωλήσεων στην εφοδιαστική αλυσίδα για το διαγωνισμό **TAC SCM**. Επίσης, η βελτίωση στο κομμάτι της λογιστικής παλινδρόμησης, για παράδειγμα, με την εισαγωγή ποινών στα βάρη κατά τη διάρκεια της διαδικασίας βελτιστοποίησης ή με την εισαγωγή νέων χαρακτηριστικών, θα μπορούσε να βελτιώσει το σφάλμα γενίκευσης της εκτίμησης. Από πλευράς εφαρμογής, ο αυτόνομος πράκτορας αναμένεται να βελτιωθεί συνολικά με τη βελτιστοποίηση και στο

κομμάτι των προμηθειών που είναι αλληλένδετο τόσο με το κομμάτι των πωλήσεων όσο και με την συνολική επίδοση του πράκτορα. Αν θεωρηθεί ότι το τμήμα των πωλήσεων έχει λυθεί από όλους τους ανταγωνιστές με παρόμοιες τεχνικές μηχανικής μάθησης, τότε το μεγαλύτερο περιθώριο κέρδους θα προκύψει από την εύρεση συμφέρουσων προσφορών από τη μεριά των προμηθευτών και την έγκαιρη παράδοσή τους ώστε να μην υπάρξουν αργοπορίες και αντίστοιχα ποινές.

Όπως και στην εφαρμογή της NEAR στο περιβάλλον TAC SCM, έτσι και στο περιβάλλον **TAC AA**, μελλοντικές βελτιώσεις στη μέθοδο NEAR αλλά και στη μοντελοποίηση του προβλήματος θα βελτιώσουν σημαντικά την επίδοση του πράκτορα. Η τμηματοποιημένη αρχιτεκτονική του πράκτορα, όπως παρουσιάστηκε στο αντίστοιχο κεφάλαιο, βοηθάει στην επιμέρους αντιμετώπιση κάθε προβλήματος με αποτέλεσμα όσο καλύτερη είναι η λύση που προκύπτει, τόσο καλύτερη να είναι και η απόδοση του πράκτορα. Έτσι, με τη βελτίωση των εκτιμητών (φίλτρο σωματιδίων για την κατάσταση του πληθυσμού των χρηστών, αντιστοίχιση πλειοδοσίας σε κόστος ανά κλικ, πρόβλεψη μελλοντικών αγορών από χρήστες) και του αλγορίθμου υπολογισμού του προϋπολογισμού, μπορεί να θεωρηθεί ότι συνολικά θα επωφεληθεί και ο πράκτορας.

Για την εφαρμογή του **Πόκερ**, θα μπορούσαν να χρησιμοποιηθούν τεχνικές συνεξέλιξης μέσω της NEAR, ώστε να βοηθήσουν περαιτέρω στη διαδικασία της μάθησης των ΔΗΚ, με την προϋπόθεση αντίστοιχων υπολογιστικών υποδομών και αλγορίθμων τύπου map-reduce για την κλιμάκωση της μάθησης και της αξιολόγησης των δικτύων. Επίσης, θα πρέπει να μελετηθούν και επεκτάσεις σε πιο δύσκολες παραλλαγές πόκερ όπως σε τραπέζια περισσότερων παικτών αλλά και με πονταρίσματα χωρίς όριο. Για την ταχύτερη εφαρμογή της NEAR θα πρέπει να μελετηθεί η ανάπτυξη του αλγορίθμου σε παράλληλη υλοποίηση. Επίσης, σημαντικό είναι και το δίλημμα μεταξύ πόλωσης και διασποράς (bias-variance trade-off) κατά τη διάρκεια της εκπαίδευσης ώστε να επιτυγχάνεται η καλύτερη δυνατή γενίκευση, εξαιτίας της μεγάλης πολυπλοκότητας του παιχνιδιού του πόκερ. Τέλος, για την ανταγωνιστικότητα του πράκτορα απαιτείται και ένα τμήμα μοντελοποίησης του αντιπάλου ώστε ο πράκτορας να εκμεταλλεύεται τα τρωτά σημεία του αντιπάλου.

Τελειώνοντας, μία ακόμη περιοχή εφαρμογής της μεθόδου NEAR που μελετάται είναι αυτή των παιχνιδιών στρατηγικής πραγματικού χρόνου (real time strategy or RTS). Στα πλαίσια αυτής της εφαρμογής έχει γίνει η μελέτη ενός RTS παιχνιδιού με μοντελοποίηση ως πρόβλημα ενισχυτικής μάθησης και εφαρμογή μανθάνοντων συστημάτων ταξινόμησης (learning classifier systems) με ικανοποιητικά αποτελέσματα [TCM11].

Παραρτήματα

Παράρτημα Α

Παραμετροποίηση της μεθόδου NEAR

Στο παράρτημα αυτό, παρουσιάζονται οι τιμές των παραμέτρων για τη μέθοδο NEAR και τα προβλήματα των Κεφαλαίων 4 και 6, έτσι ώστε ο ενδιαφερόμενος αναγνώστης να μπορεί να αναπαράγει τα πειράματα. Οι βασικές τιμές των παραμέτρων παρουσιάζονται στον Πίνακα Α.1, ενώ στον Πίνακα Α.2 περιέχονται οι τιμές των παραμέτρων που διαφοροποιήθηκαν στα συγκεκριμένα προβλήματα. Για τη μέθοδο NEAT τόσο για το όχημα πλαγιάς όσο και για τον χρονοπρογραμματισμό διαδικασιών χρησιμοποιήθηκαν οι παράμετροι που αναφέρονται στο [WS06], εκτός από την πιθανότητα προσθήκης μιας αναδρομικής σύνδεσης η οποία τέθηκε σε 0.2. Είσοδος πόλωσης χρησιμοποιήθηκε για τα προβλήματα χρονοσειρών και όχι για τα προβλήματα ενισχυτικής μάθησης. Τα δείγματα των χρονοσειρών $x(i)$ κανονικοποιήθηκαν με την αφαίρεση του μέσου όρου και τη διαίρεση με το διάστημα μεταξύ μέγιστης και ελάχιστης τιμής $(x(i) - mean)/(max - min)$ όπως αυτή καταγράφεται στα δεδομένα εκπαίδευσης. Οι είσοδοι στα προβλήματα ενισχυτικής μάθησης κανονικοποιήθηκαν μεταξύ -1 και 1.

Πίνακας Α.1: Βασικές τιμές παραμέτρων για τα προβλήματα ενισχυτικής μάθησης και χρονοσειρών.

Παράμετρος	Τιμή	Παράμετρος	Τιμή	Παράμετρος	Τιμή
$noise$	10^{-7}	p_ρ	0.05	p_w	0.8
c_α	0.5	p_D	0.05	p_n	0.05
c_β	1.0	p_{mate}	0.33	p_c	0.1
c_γ	1.0	p_{mutate}	0.33	T	1.0
ρ_{max}	0.99	D_{max}	1.0	w_{pow}	0.25
ρ_{min}	0.55	D_{min}	0.05	S_{thres}	0.3
p_i	0.001	Max Stag	5	$N^{initial}$	1

Οι υλοποιήσεις των χρονοσειρών Mackey-Glass, MSO και Lorentz attractor, καθώς και του προβλήματος χρονοπρογραμματισμού αναπτύχθηκαν από την αρχή.

Πίνακας Α.2: Τιμές παραμέτρων που διαφοροποιήθηκαν ανάμεσα στα προβλήματα.

Παράμετρος	SPB-NM	DPB-NM	DPB-NM-D	MG	MSO	Lorentz
w_{pow}	2.5	2.5	2.5	1	1	1
p_c	0.05	0.05	0.05	0.2	0.2	0.2
p_n	0.1	0.1	0.1	0.3	0.3	0.3
$N^{initial}$	1	1	1	25	25	25

Για τα προβλήματα οχήματος πλαγιάς και της ισορρόπησης μίας ράβδου, καθώς και για τον αλγόριθμο επέκτασης με συναρτήσεις βάσεων Fourier, χρησιμοποιήθηκαν υλοποιήσεις από τη βιβλιοθήκη rl-library [1] προσαρμοσμένες στις απαιτήσεις τις διατριβής. Τέλος, για τη βασική λειτουργικότητα του προβλήματος ισορρόπησης δύο ράβδων, χρησιμοποιήθηκε η βιβλιοθήκη ESP-Java[2] η οποία προσαρμόστηκε για τις ανάγκες των διεπαφών της βιβλιοθήκης rl-glue.

[1] http://library.rl-community.org/
[2] http://nn.cs.utexas.edu/?ESP-Java

Βιβλιογραφία

[AL91] David H. Ackley and Michael L. Littman. Interactions between learning and evolution. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II, SFI Studies in the Sciences of Complexity*, volume X, pages 487–509, Reading, MA, 1991. Addison-Wesley.

[AMGC02] Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for on-line non-linear / non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.

[AS05] Raghu Arunachalam and Norman M. Sadeh. The supply chain trading agent competition. *Electronic Commerce Research and Applications*, 4(1):66–84, 2005.

[Bal96] James M. Baldwin. A new factor in evolution. *American Naturalist*, 30:441–451, 1896.

[BB96] Steven J. Bratke and Andrew G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1-3):33–57, 1996.

[BBD+03] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron. Approximating game-theoretic optimal strategies for full-scale poker. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 661–668, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[BFSO84] Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. *Classification and Regression Trees*. Chapman and Hall, 1984.

[BGG+04] Michael Benisch, Amy Greenwald, Ioanna Grypari, Roger Lederman, Victor Naroditskiy, and Michael Tschantz. Botticelli: A supply chain management agent designed to optimize under uncertainty. *ACM Transactions on Computational Logic*, 4(3):29–37, 2004.

[BGNT04] Michael Benisch, Amy Greenwald, Victor Naroditskiy, and Michael Carl Tschantz. A stochastic programming approach to scheduling in TAC SCM. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, EC '04, pages 152–159, New York, NY, USA, 2004. ACM.

[BM08] Erkin Bahceci and Risto Miikkulainen. Transfer of evolved pattern-based heuristics in games. In *IEEE Symposium on Computational Intelligence and Games*, 2008.

[Boy02] Justin A. Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49:233–246, 2002.

[BP05] Štefan Babinec and Jiři Pospíchal. Two approaches to optimize echo state neural networks. In *Proceedings of Mendel 2005, 11th International Conference on Soft Computing*, pages 39–44, 2005.

[BPSS98] Darse Billings, Denis Papp, Jonathan Schaeffer, and Duane Szafron. Poker as a testbed for machine intelligence research. In *Twelfth Canadian Conference on Artificial Intelligence*, June 1998.

[BT05] Keith Bush and Batsukh Tsendjav. Improving the richness of echo state features using next ascent local search. In *Proceedings of the Artificial Neural Networks in Engineering Conference*, pages 227–232, St. Louis, MO, 2005.

[CAS+06] John Collins, Raghu Arunachalam, Norman Sadeh, Joakim Eriksson, Niclas Finne, and Sverker Janson. The supply chain management game for the 2007 trading agent competition. Technical Report CMU-ISRI-07-100, Carnegie Mellon University, December 2006.

[CCSM11] Kyriakos C. Chatzidimitriou, Antonios C. Chrysopoulos, Andreas L. Symeonidis, and Pericles A. Mitkas. Enhancing agent intelligence through evolving reservoir networks for prediction in power stock markets. In *Agent and Data Mining Interaction 2011 Workshop held in conjuction with the conference on Autonomous Agents and Multi-Agent Systems (AAMAS) 2011*, 2011.

[CM10] Kyriakos C. Chatzidimitriou and Pericles A. Mitkas. A neat way for evolving echo state networks. In *European Conference on Artificial Intelligence*. IOS Press, August 2010.

[CM12] Kyriakos C. Chatzidimitriou and Pericles A. Mitkas. Adaptive reservoir computing through evolution and learning. *Neurocomputing*, accepted for publication, 2012.

[CPM10] Kyriakos C. Chatzidimitriou, Fotis Psomopoulos, and Pericles A. Mitkas. Grid-enabled parameter initialization for high performance machine learning tasks. In *5th EGEE User Forum*, April 2010.

[CPMV11] Kyriakos C. Chatzidimitriou, Ioannis Partalas, Pericles A. Mitkas, and Ioannis Vlahavas. Transferring evolved reservoir features in reinforcement learning tasks. In *European Workshop on Reinforcement Learning*, 2011.

[CS09] Kyriakos C. Chatzidimitriou and Andreas L. Symeonidis. Data-mining-enhanced agents in dynamic supply-chain-management environments. *Intelligent Systems*, 24(3):54–63, 2009. Special issue on Agents and Data Mining.

[CSKM08] Kyriakos C. Chatzidimitriou, Andreas L. Symeonidis, Ioannis Kontogounis, and Pericles A. Mitkas. Agent Mertacor: A robust design for dealing with uncertainty and variation in SCM environments. *Expert Systems with Applications*, 35(3):591–603, October 2008. Cited by: Collins2008.

[CSM08] Kyriakos C. Chatzidimitriou, Andreas L. Symeonidis, and Pericles A. Mitkas. Data mining-driven analysis and decomposition in agent supply chain management networks. In *IEEE/WIC/ACM Workshop on Agents and Data Mining Interaction*, Sydney, Australia, 9-12 December 2008.

[CSM12] Kyriakos C. Chatzidimitriou, Andreas L. Symeonidis, and Pericles A. Mitkas. Policy search through adaptive function approximation for bidding in TAC SCM. In *Trading Agent Design and Analysis (TADA) 2012 Workshop held in conjuction with the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS) 2012*, Lecture Notes in Business Information Processing. Springer, 2012.

[CSSM11] Kyriakos C. Chatzidimitriou, Lampros C. Stavrogiannis, Andreas L. Symeonidis, and Pericles A. Mitkas. An adaptive proportional value-per-click agent for bidding in ad auctions. In *Trading Agent Design and Analysis (TADA) 2011 Workshop held in conjuction with the International Joint Conference on Artificial Intelligence (IJCAI) 2011*, 2011.

[Dav02] Aaron Davidson. Opponent modeling in poker: Learning and acting in a hostile and uncertain environment. Master's thesis, University of Alberta, 2002.

[DBS08] Alexandre Devert, Nicolas Bredeche, and Marc Schoenauer. Unsupervised learning of echo state netowrks: a case study in artificial embryogeny. In *Proceedings of the 8th International*

Conference on Artificial Evolution, volume 4926 of *LNCS*, pages 278–290. Springer, 2008.

[DCSM08] Christos Dimou, Kyriakos C. Chatzidimitriou, Andreas L. Symeonidis, and Pericles A. Mitkas. Creating and reusing metric graphs for evaluating agent performance in the supply chain management domain. In *First Workshop on Knowledge Reuse (KREUSE'2008) hosted at the 10th International Conference on Software Reuse*, Beijing (China), May 25-29 2008.

[Dem06] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[DGKM08] Aparna Das, Ioannis Goitis, Anna R. Karlin, and Claire Mathieu. On the effects of competing advertisements in keyword auctions. Working Paper, 2008.

[dLPC99] Maria de Lourdes Pẽna Castillo. Probabilities and simulation in poker. Master's thesis, University of Alberta, 1999.

[Elm04] W. Elmaghraby. Auctions and pricing in e-marketplaces. In D. Simchi-Levi, S.D. Wu, and Z.J. Shen, editors, *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*, pages 213–246. Kluwer, 2004.

[ES03] Agoston E. Eiben and James E. Smith. *Introduction to Evolutionary Computation*. Natural Computing Series. Springer-Verlag, 2003.

[FB98] B. Farhang-Boroujeny. *Adaptive filters: Theory and applications*. Wiley, 1998.

[FDM08] Dario Floreano, Peter Dürr, and Claudio Mattiussi. Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1:47–62, 2008.

[FG97] Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, pages 21–35, London, UK, 1997. Springer-Verlag.

[FM08] Jon Feldman and S. Muthukrishnan. Algorithmic methods for sponsored search advertising. In Zhen Liu and Cathy H. Xia, editors, *Performance Modeling and Engineering*, 2008.

[FP08] Jerome H. Friedman and Bogdan E. Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.

[Fri37] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.

[Fri40] Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.

[GBS06] Albortz Geramifard, Michael Bowling, and Richard S. Sutton. Incremental least-squares temporal difference learning. In *Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference*, pages 356–361, 2006.

[GDR+11] Alborz Geramifard, Finale Doshi, Joshua Redding, Nicholas Roy, and Jonathan P. How. Online discovery of feature dependencies. In *Proceedings of the 28th International Conference on Machine Learning*, pages 881–888, 2011.

[GM97] Faustino Gomez and Risto Miikkulainen. Incremental evolution of complex general behavior. *Adaptive Behavior*, 5(3-4):317–342, 1997.

[GP08] Sertan Girgin and Philippe Preux. Basis function construction in reinforcement learning using cascade-correlation learning architecture. In *Seventh International Conference on Machine Learning and Applications*, pages 75–82, 2008.

[GSM06] Faustino Gomez, Jürgen Schmidhuber, and Risto Miikkulainen. Efficient non-linear control through neuroevolution. In *Proceedings of the European Conference on Machine Learning (ECML 2006)*, volume 4212/2006 of *Lecture Notes in Computer Science*, pages 654–662. Springer Berlin / Heidelberg, 2006.

[GSM08] Faustino Gomez, Jürgen Schmidhuber, and Risto Miikkulainen. Accelerated neural evolution through cooperatively coevolved synapses. *Journal of Machine Learning Research*, 9:937–965, 2008.

[GWP96] Frederic Gruau, Darrell Whitley, and Larry Pyeatt. A comparison between cellular encoding and direct encoding for genetic neural networks. In J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 81–89, Cambridge, MA, 1996. MIT Press.

[HFH+09] M Hall, E Frank, G Holmes, B Pfahringer, P Reutemann, and I H Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[HKvD+09] Alexander Hogenboom, Wolfang Ketter, Jan van Dalen, Uzay Kaymak, John Collins, and Alok Gupta. Product pricing in tac scm using adaptive real-time probability of acceptance estimations based on economic regimes. In *Workshop: Trading Agent Design and Analysis (TADA) at Twenty-First International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 15–24, July 2009.

[HMI09] Verena Heidrich-Meisner and Christian Igel. Neuroevolution strategies for episodic reinforcement learning. *Journal of Algorithms*, 64(4):152–168, October 2009.

[HN87] Geoffrey E. Hinton and Steven J. Nowlan. How learning can guide evolution. *Complex Systems*, 1:495–502, 1987.

[HN10] Stefan Haflidason and Richard Neville. Quantifying the severity of the permutation problem in neuroevolution. In F. Peper, H. Umeo, N. Matsui, and T. Isokawa, editors, *Natural Computing*, volume 2 of *Proceedings in Information and Communications Technology*, pages 149–156. Springer Japan, 2010.

[HO01] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[HRLJ06] Minghua He, Alex Rogers, Xudong Luo, and Nicholas R. Jennings. Designing a successful trading agent for supply chain management. In *Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-06)*, 2006.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(9):1735–1780, 1997.

[Ige03] Christian Igel. Neuroevolution for reinforcement learning using evolution strategies. In *Proceedings of the Congress on Evolutionary Computation (CEC 2003)*, volume 4, pages 2588–2595. IEEE Press, 2003.

[IS08] Christian Igel and Bernhard Sendhoff. Genesis of organic computing systems: Coupling evolution and learning. In R. P. Würtz, editor, *Organic Computing. Understanding Complex Systems*, chapter 7. Springer-Verlag, Berlin, Heidelberg, 2008.

[IvdZBP04] Kazuo Ishii, Tijn van der Zant, Vlatko Bečanović, and Paul Plöger. Identification of motion with echo state network. In *Proceedings of the OCEANS 2004 MTS/IEEE - TECHNO-OCEAN 2004 Conference*, volume 3, pages 1205–1210, 2004.

[Jae01] Herbert Jaeger. The ‘‘echo state’’ approach to analysing and training recurrent neural networks - with an erratum note. Technical Report GMD Report 148, German National Research Center for Information Technology, 2001.

[Jae02] Herbert Jaeger. Tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the ‘‘echo state network’’ approach. Technical Report GMD Report 159, German National Research Center for Information Technology, 2002.

[JBS08a] Fei Jiang, Hugues Berry, and Marc Schoenauer. Supervised and evolutionary learning of echo state networks. In *Proceedings of 10th International Conference on Parallel Problem Solving from Nature, PPSN 2008*, volume 5199 of *LNCS*, pages 215–224. Springer-Verlag, 2008.

[JBS08b] Fei Jiang, Hugues Berry, and Marc Schoenauer. Unsupervised learning of echo state networks: Balancing the double pole. In *Proceedings of the 10th Genetic and Evolutionary Computation Conference*, pages 869–870. ACM, 2008.

[JCC+10] Patrick R. Jordan, Ben Cassell, Lee F. Callender, Akshat Kaul, and Michael P. Wellman. The ad auctions game for the 2010 trading agent competition. Technical report, University of Michigan, Ann Arbor, MI 48109-2121 USA, 2010.

[JLPS07] Herbert Jaeger, Mantas Lukoševičius, Dan Popovici, and Udo Siewert. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Networks*, 20(3):335–352, 2007.

[JM08] Bernard J. Jansen and Tracy Mullen. Sponsored search: an overview of the concept, history, and technology. *International Journal of Electronic Business*, 6(2):114–131, 2008.

[Joh07] Michael B. Johanson. Robust strategies and counter-strategies: Building a champion level computer poker player. Master’s thesis, University of Alberta, 2007.

[Jor10] Patrick R. Jordan. *Practical Strategic Reasoning with Applications in Market Games*. PhD thesis, University of Michigan, Ann Arbor, MI, 2010.

[KC03] J. O. Kephart and D. M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, January 2003.

[KCSM06] Ioannis Kontogounnis, Kyriakos C. Chatzidimitriou, Andreas L. Symeonidis, and Pericles A. Mitkas. A robust agent design for

dynamic scm environments. In *4th Hellenic Conference on Artificial Intelligence (SETN'06)*, volume 3955 of *Lecture Note in Artificial Intelligence*, pages 127–136, Heraklion, Crete, Greece, May 18–20 2006. Springer-Verlag.

[KE95] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the International Conference on Neural Networks*, pages 1942–1948, 1995.

[Kir00] Lou Kirieger. *Hold'em Excellence: From Beginner to Winner*. ConJelCo, 2nd edition edition, 2000.

[KLM96] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[KMJ+09] Christopher Kiekintveld, Jason Miller, Patrick R. Jordan, Lee F. Callender, and Michael P. Wellman. Forecasting market prices in a supply chain game. *Electronic Commerce Research and Applications*, 8:63–77, 2009.

[KOT11] George Konidaris, Sarah Osentoski, and Philip Thomas. Value function approximation in reinforcement learning using the fourier basis. In *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence*, pages 380–385, 2011.

[KPS+04] Christopher Kiekintveld, Michael P.Wellman, Satinder Singh, Joshua Estelle, Yevgeniy Vorobeychik, Vishal Soni, and Matthew Rudary. Distributed feedback control for decision making on supply chains. In *Fourteenth International Conference on Automated Planning and Scheduling*, 2004.

[Kri02] Vijay Krishna. *Auction Theory*. Academic Press, 2002.

[Lah06] Sébastien Lahaie. An analysis of alternative slot auction designs for sponsored search. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pages 218–227, New York, NY, USA, 2006. ACM.

[LJ09] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3:127–149, 2009.

[LP03] Michail G. Lagoudakis and Ronald Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003.

[LPDF09] Adi Livnat, Christos Papadimitriou, Jonathan Dushoff, and Marcus W. Feldman. A mixability theory of the role of sex in evolution. In *Proceedings of the National Academy of Sciences of the United States of America*, 2009.

[Mii11] Risto Miikkulainen. Neuroevolution. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*. Springer, 2011.

[MS10] Hamid Reza Maei and Richard S. Sutton. GQ(λ): A general gradient algorithm for temporal-di erence prediction learning with eligibility traces. In *Proceedings of the Third Conference on Artificial General Intelligence*, Lugano, Switzerland, 2010.

[NJ00] Andrew Y. Ng and Michael Jordan. Pegasus: A policy search method for large mdps and pomdps. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, page 406–415, Stanford, California, 2000.

[OR94] Martin J. Osborne and Ariel Rubinstein. *A course in game theory*. MIT Press, 1994. My library, Paperback.

[PB98] Jordan B. Pollack and Alan D. Blair. Co-evolution in the successful learning of backgammon strategy. *Machine Learning*, 32(3):225–240, September 1998.

[PPWLL07] Ronald Parr, Christopher Painter-Wakefield, Lihong Li, and Michael Littman. Analyzing feature generation for value-function approximation. In *Proceedings of the 24th International Conference on Machine Learning*, pages 737–744, 2007.

[Pri11] PricewaterhouseCoopers. Iab internet advertising revenue report: 2011 first six months results. Technical report, Interactive Advertising Bureau, September 2011.

[Pro05] Danil Prokhorov. Echo state networks: Appeal and challenges. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Montreal, Canada, August 2005.

[PS05] David Pardoe and Peter Stone. Bidding for customer orders in TAC SCM. In P. Faratin and J.A. Rodriguez-Aguilar, editors, *Agent Mediated Electronic Commerce VI: Theories for and Engineering of Distributed Mechanisms and Systems (AMEC 2004)*, volume 3435 of *Lecture Notes in Artificial Intelligence*, pages 143–157. Springer Verlag, Berlin, 2005.

[PS09] David Pardoe and Peter Stone. An autonomous agent for supply chain management. In Gedas Adomavicius and Alok Gupta, editors, *Handbooks in Information Systems Series: Business Computing*, volume 3, pages 141–72. Emerald Group, 2009.

[R D11] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[Rad93] Nicholas J. Radcliffe. Genetic set recombination and its application to neural network topology optimization. *Neural computing and applications*, 1(1):67–90, 1993.

[RB97] Christopher D. Rosin and Richard K. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5:1–29, March 1997.

[RI09] Benjamin Roeschies and Christian Igel. Structure optimization of reservoir networks. *Logic Journal of IGPL*, 2009.

[SB98] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[SCAM07] Andreas L. Symeonidis, Kyriakos C. Chatzidimitriou, Ioannis N. Athanasiadis, and Pericles A. Mitkas. Data mining for agent reasoning: A synergy for training intelligent agents. *Engineering Applications of Artificial Intelligence*, 20(8):1097–1111, December 2007.

[SGL06] István Szita, Viktor Gyenes, and András Lőrincz. Reinforcement learning with echo state networks. In *Artificial Neural Networks – ICANN 2006*, volume 4131/2006 of *Lecture Notes in Computer Science*, pages 830–839. Springer Berlin / Heidelberg, 2006.

[She99] Alex Shelby. Optimal heads-up preflop holdem. Online, 1999.

[SLKSL03] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi. *Designing and Managing the Supply Chain*. McGraw-Hill, 2003.

[SM99] D. Sklansky and M. Malmuth. *Hold 'em Poker for Advanced Players*. Two Plus Two Publications, 1999.

[SM02] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[SM04] Kenneth O. Stanley and Risto Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100, 2004.

[SS96] Satinder P. Singh and Richard S. Sutton. Reinforcement learning with replacing eligibility traces. *Machine Learning*, 22(1-3):123–158, 1996.

[SSF06] Mihai Stan, Bogdan Stan, and Adina Magda Florea. A dynamic strategy agent for supply chain management. In *Proceedings of the Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 227–232, 2006.

[Sta04] Kenneth O. Stanley. *Efficient Evolution of Neural Networks*. PhD thesis, University of Texas at Austin, 2004.

[Sto07] Peter Stone. Learning and multiagent reasoning for autonomous agents. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 13–30, January 2007.

[SWGG07] Jürgen Schmidhuber, Daan Wierstra, Matteo Gagliolo, and Faustino J. Gomez. Training recurrent networks by evolino. *Neural Computation*, 19(3):757–779, 2007.

[TCM11] Michalis Tsapanos, Kyriakos C. Chatzidimitriou, and Pericles A. Mitkas. A zeroth-level classifier system for real time strategy games. In *2011 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, volume 2, pages 244–247, 2011. 21the 142 submissions were accepted as short papers. This paper has been accepted as a short paper for oral presentation.

[TJS08] Matthew E. Taylor, Nicholas K. Jong, and Peter Stone. Transferring instances for model-based reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases*, volume 5212, pages 488–505, September 2008.

[TKS08] Matthew E. Taylor, Gregory Kuhlmann, and Peter Stone. Autonomous transfer for reinforcement learning. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 283–290, 2008.

[TpbBR11] Terry M Therneau and Beth Atkinson. R port by Brian Ripley. *rpart: Recursive Partitioning*, 2011. R package version 3.1-50.

[TS07] Matthew E. Taylor and Peter Stone. Cross-domain transfer for reinforcement learning. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, June 2007.

[TS09] Matthew Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10:1633–1685, 2009.

[TW09] Brian Tanner and Adam White. Rl-glue: Language-independent software for reinforcement-learning experiments. *Journal of Machine Learning Research*, 10:2133–2136, 2009.

[TWS07] Matthew E. Taylor, Shimon Whiteson, and Peter Stone. Transfer via inter-task mappings in policy search reinforcement learning. In *6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, 2007.

[Vor11] Yevgeniy Vorobeychik. A Game Theoretic Bidding Agent for the Ad Auction Game. In *Third International Conference on Agents and Artificial Intelligence (ICAART 2011)*, January 2011.

[Wie91] Alexis P. Wieland. Evolving neural network controllers for unstable systems. In *Proceedings of the International Joint Conference on Neural Networks*, pages 667–673, Seattle, WA, 1991. IEEE.

[Wil45] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[WM97] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1:67–82, April 1997.

[WS06] Shimon Whiteson and Peter Stone. Evolutionary function approximation for reinforcement learning. *Journal of Machine Learning Research*, 7:877–917, May 2006.

[WTS09] Shimon Whiteson, Matthew E. Taylor, and Peter Stone. Critical factors in the empirical performance of temporal difference and evolutionary methods for reinforcement learning. *Journal of Autonomous Agents and Multi-Agent Systems*, 21(1):1–27, 2009.

[WW97] Yong Wang and Ian H. Witten. Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*, pages 128–137, 1997.

[XgHH02] Xin Xu, Han gen He, and Dewen Hu. Efficient reinforcement learning using recursive least-squares methods. *Journal of Artificial Intelligence Research*, 16:259–292, 2002.

[XLP05] Dongming Xu, Jing Lan, and Jose C. Principe. Direct adaptive control: An echo state network and genetic algorithm approach. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1483–1486, Montreal, Canada, July-August 2005.

[Yao99] Xin Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.

[ZZC05] Chengqi Zhang, Zili Zhang, and Longbing Cao. Agents and data mining: Mutual enhancement by integration. In *International Workshop Autonomous Intelligent Systems: Agents and Data Mining (AIS-ADM 05)*, LNCS 3505, pages 50–61. Springer, 2005.