ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
Τμήμα Ηλεκτρολόγων & Ηλεκτρονικών Μηχανικών
www.eee.uniwa.gr
Θηβών 250, Αθήνα-Αιγάλεω 12244
Τηλ. +30 210 538-1225, Fax. +30 210 538-1226

UNIVERSITY of WEST ATTICA
FACULTY OF ENGINEERING
Department of Electrical & Electronics Engineering
www.eee.uniwa.gr
250, Thivon Str., Athens, GR-12244, Greece
Tel: +30 210 538-1225, Fax: +30 210 538-1226

Πρόγραμμα Μεταπτυχιακών Σπουδών
Ηλεκτρικές & Ηλεκτρονικές Επιστήμες μέσω Έρευνας

Master of Science By Research in
Electrical & Electronics Engineering

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη αλγορίθμου Τεχνητής Ευφυίας αυτόματης εκμάθησης της στρατηγικής παίκτη στο επιτραπέζιο παιχνίδι Dominion

Μεταπτυχιακός Φοιτητής : Γεώργιος Αγγελόπουλος, ΑΜ 0022
Επιβλέπων : Δημήτριος Μετάφας, Επίκουρος Καθηγητής

ΑΙΓΑΛΕΩ, ΦΕΒΡΟΥΑΡΙΟΣ 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
Τμήμα Ηλεκτρολόγων & Ηλεκτρονικών Μηχανικών
www.eee.uniwa.gr
Θηβών 250, Αθήνα-Αιγάλεω 12244
Τηλ. +30 210 538-1225, Fax. +30 210 538-1226

UNIVERSITY of WEST ATTICA
FACULTY OF ENGINEERING
Department of Electrical & Electronics Engineering
www.eee.uniwa.gr
250, Thivon Str., Athens, GR-12244, Greece
Tel: +30 210 538-1225, Fax: +30 210 538-1226

Πρόγραμμα Μεταπτυχιακών Σπουδών
Ηλεκτρικές & Ηλεκτρονικές Επιστήμες μέσω Έρευνας

Master of Science By Research in
Electrical & Electronics Engineering

MSc Thesis

Development of an Artificial Intelligence algorithm for Auto-learning the player Strategy for the Dominion Board Game

Student: Angelopoulos George, Registration Number 0022
MSc Thesis Supervisor: Dimitrios Metafas, Assistant Professor

ATHENS-EGALEO, FEBRUARY 2021

Η Μεταπτυχιακή Διπλωματική Εργασία έγινε αποδεκτή, εξετάστηκε και βαθμολογήθηκε από την εξής τριμελή εξεταστική επιτροπή:

Επιβλέπων/ουσα	Μέλος	Μέλος
Dimitrios Metafas Digitally signed by Dimitrios Metafas Date: 2022.02.08 14:28:48 +02'00'	 Digitally signed by Ioannis Famelis Date: 2022.02.09 07:08:23 +02'00'	Ilias Zois Digitally signed by Ilias Zois Date: 2022.02.09 12:01:36 +02'00'
Μετάφας Δημήτριος	Φαμέλης Ιωάννης	Ζώης Ηλίας
Επίκουρος Καθηγητής	Καθηγητής	Επίκουρος Καθηγητής

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Αγγελόπουλος Γεώργιος του Βασιλείου, με αριθμό μητρώου MSCRES-0022 φοιτητής του Προγράμματος Μεταπτυχιακών Σπουδών «Ηλεκτρικές και Ηλεκτρονικές Επιστήμες μέσω Έρευνας» του Τμήματος Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Είμαι συγγραφέας αυτής της μεταπτυχιακής διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Τέλος, βεβαιώνω ότι η εργασία αυτή δεν έχει κατατεθεί στο πλαίσιο των απαιτήσεων για τη λήψη άλλου τίτλου σπουδών ή επαγγελματικής πιστοποίησης πλην του παρόντος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών

Αγγελόπουλος Γεώργιος

ΠΕΡΙΛΗΨΗ

Κατά τα προηγούμενα χρόνια, αρκετά επιτραπέζια παιχνίδια έχουν χρησιμοποιηθεί σαν χώρος ανάπτυξης και δοκιμής, διαφόρων τεχνικών τεχνητής νοημοσύνης. Τα επιτραπέζια παιχνίδια είναι ιδανικά για αυτό το ρόλο, καθώς προσφέρουν ένα περιβάλλον με αυστηρά καθορισμένους κανόνες, που δεν επιδέχονται εξαιρέσεις, και τα αποτελέσματα δεν παρουσιάζουν σφάλματα ή «θορύβους». Είναι ένας ιδεατός κόσμος, στον οποίο μπορούν να δοκιμαστούν θεωρίες και τεχνικές, και να εκτιμηθεί η αποτελεσματικότητα τους, πριν την επέκταση τους στο «χάος» του πραγματικού κόσμου.

Στόχος της παρούσης εργασίας ήταν η δημιουργία αλγόριθμου τεχνητής νοημοσύνης, βασισμένου στη μέθοδο της εξαναγκασμένης μάθησης, και πιο συγκεκριμένα στην τεχνική Q Learning, ικανού να αναπτύξει στρατηγική με προοπτικές νίκης, για ένα επιτραπέζιο παιχνίδι. Το ερώτημα που προσπαθούμε να απαντήσουμε, είναι το κατά πόσο η συγκεκριμένη τεχνική, είναι ικανή να ανταποκριθεί με ικανοποιητικό τρόπο, σε ένα πολύπλοκο περιβάλλον, και να εκπαιδεύσει έναν πράκτορα, ώστε να παίρνει την καλύτερη δυνατή απόφαση, όταν ο αριθμός των επιλογών είναι μεγάλος. Στην πορεία της εργασίας ανέκυψε και ένα νέο ερώτημα, κατά πόσο είναι δυνατό επιφέροντας κάποιες αλλαγές στην μέθοδο επιλογής ενεργειών του πράκτορα, να επιταχύνουμε την εκπαίδευση, χωρίς να μειώσουμε την αποτελεσματικότητά του.

Επιλέξαμε το επιτραπέζιο παιχνίδι καρτών Dominion (Κυρίαρχος) για τις δοκιμές μας, καθώς έχει αρκετά απλούς κανόνες, αλλά ο αριθμός των διαφορετικών καρτών που χρησιμοποιούνται, δημιουργεί ένα μεγάλο φάσμα διαφορετικών επιλογών, και καθιστά το στόχο της εκπαίδευσης του πράκτορα αρκετά προκλητικό. Επίσης κατά το παρελθόν, άλλες τεχνικές εξαναγκασμένης μάθησης, όπως τα νευρωνικά δίκτυα και τα Monte Carlo Trees, έχουν δοκιμαστεί πάνω στο συγκεκριμένο παιχνίδι, οπότε μπορούν να εξαχθούν χρήσιμα συμπεράσματα, από τα αποτελέσματα της κάθε τεχνικής.

ΛΕΞΕΙΣ – ΚΛΕΙΔΙΑ: Επιτραπέζια παιχνίδια, Εξαναγκασμένη μάθηση, Μέθοδος επιλογής ενεργειών, Τεχνητή νοημοσύνη, Dominion, Q Learning

ABSTRACT

In recent years, several board games have been used as a test field for the development of various artificial intelligence techniques. Board games are ideal for this role, as they offer an environment with strict rules, no exceptions, and the results are error-free and without “noise”. It is an imaginary world, in which theories and techniques can be tested, and their effectiveness evaluated, before extending to the "chaos" of the real world.

The aim of this paper was to create an artificial intelligence algorithm for a board game, based on the method of reinforcement learning, and more specifically on the Q Learning technique, capable of developing a strategy with the prospect of winning. The question we are trying to answer is whether this particular technique is capable of responding satisfactorily, in a complex environment, and training an agent which makes the best possible decisions when the number of options is quite large. In the course of the work, a new question arose, whether it is possible, by making some changes in the action selection method of the agent, to accelerate the training, without reducing its effectiveness.

We chose the Dominion card game for our tests, as it has quite simple rules, but the number of different cards used, creates a huge range of different options, and makes the goal of agent training quite challenging. Also in the past, other reinforcement learning techniques, such as neural networks and Monte Carlo Trees, have been tested on this game, so useful conclusions can be drawn from the results of each technique.

KEYWORDS: Action selection method, Artificial intelligence, Board games, Dominion, Reinforcement learning, Q learning

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου, κύριο Δημήτρη Μετάφα, για την εμπιστοσύνη που μου έδειξε, την άψογη συνεργασία του, αλλά και για την καθοδήγησή του, χωρίς την οποία δεν θα ήμουν σε θέση να ολοκληρώσω την παρούσα εργασία. Πολλά ευχαριστώ χρωστάω και στον κύριο Ποτηράκη, ο οποίος με μύησε στα μυστικά της συγγραφής εργασιών, και τον κύριο Αλεξανδρίδη για την εισαγωγή στις έννοιες της τεχνητής νοημοσύνης. Δεν θα ήθελα να ξεχάσω και τους κυρίους Ζώη και Φαμέλη και την κυρία Ραγκούση, οι οποίοι με τις εύστοχες παρατηρήσεις τους και τις συμβουλές τους, υπήρξαν μεγάλη βοήθεια.

Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου για την υποστήριξη και την υπομονή που έδειξε όλο αυτό το διάστημα.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΙΣΑΓΩΓΗ:
Αντικείμενο και διάρθρωση της εργασίας

Ο όρος τεχνητή νοημοσύνη (Artificial Intelligence, AI), εισήχθη το 1956 από τον John McCarthy [1]. Ένας από τους πρώτους ορισμούς που δόθηκαν για την τεχνητή νοημοσύνη, από τους Barr και Feigenbaum [2], είναι:

«*Τεχνητή νοημοσύνη είναι ο τομέας που ασχολείται με τη σχεδίαση ευφυών υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζονται με τη νοημοσύνη στην ανθρώπινη συμπεριφορά».*

Ο Alan M. Turing [3] εμπνεύστηκε μια δοκιμασία, η οποία μάλιστα πήρε το όνομα του (The Turing test), η οποία έχει σαν σκοπό να ελέγξει αν μια μηχανή μπορεί να χαρακτηριστεί ευφυής. Η δοκιμασία αποτελείται από μια σειρά ερωτήσεων, που υποβάλλονται ταυτόχρονα σε μια μηχανή και έναν άνθρωπο. Αν στο τέλος της δοκιμασίας, δεν μπορούμε να ξεχωρίσουμε ποιες απαντήσεις δόθηκαν από τη μηχανή και ποιες από τον άνθρωπο, τότε η μηχανή επιτυγχάνει στη δοκιμασία. Η δημιουργία μιας τέτοιας μηχανής είναι ο στόχος της «ισχυρής τεχνητής νοημοσύνης», αλλά αυτός ο στόχος είναι ακόμα πολύ μακρινός.

Η «ασθενής τεχνητή νοημοσύνη» όμως δεν έχει σαν στόχο την δημιουργία μιας μηχανής που να έχει «συνείδηση» του εαυτού της, αλλά να λύνει συγκεκριμένα προβλήματα με έξυπνο τρόπο. Η προσέγγιση της υπολογιστικής νοημοσύνης μιμείται μηχανισμούς και διεργασίες έμβιων όντων, όπως η διαδικασία μάθησης του ανθρώπινου εγκεφάλου, η εξέλιξη των ειδών, η νοημοσύνη σμήνους και άλλα, προκειμένου να επιλύσει προβλήματα που δεν θα μπορούσαν να επιλυθούν με συμβατικές μεθόδους. Βρίσκει εφαρμογές σε διάφορους τομείς, όπως η αναγνώριση προτύπων, επεξεργασία σημάτων και δεδομένων, εντοπισμός και διάγνωση βλαβών και σφαλμάτων, μοντελοποίηση συστημάτων, εύρεση βέλτιστων λύσεων και άλλα. Τα τελευταία χρόνια οι «έξυπνες» μηχανές έχουν μπει στην καθημερινότητα μας. Έξυπνες ηλεκτρικές και ηλεκτρονικές οικιακές συσκευές, οχήματα που λαμβάνουν μόνα τους αποφάσεις, αυτόνομα ρομπότ, συστήματα αναγνώρισης, συστήματα πρόβλεψης είναι μερικά παραδείγματα, στα οποία η τεχνητή νοημοσύνη έχει προχωρήσει από το θεωρητικό στάδιο, στην πρακτική εφαρμογή. Μία δημοφιλής μέθοδος δημιουργίας τέτοιων έξυπνων μηχανών είναι αυτή της εξαναγκασμένης μάθησης (reinforcement learning). Τις βάσεις της μεθόδου έθεσε ο R. Bellman [4] με τον δυναμικό προγραμματισμό, αλλά η μέθοδος δανείστηκε και από τον τομέα της ψυχολογίας την έννοια της δοκιμής και αποτυχίας (trial and error).

Ένας τομέας όπου αποτέλεσε πεδίο έρευνας και εφαρμογής των τεχνικών της εξαναγκασμένης μάθησης είναι τα παιχνίδια και δη τα επιτραπέζια παιχνίδια. Πολλοί ερευνητές ανέπτυξαν αλγόριθμους εκμάθησης παιχνιδιών, χρησιμοποιώντας διάφορες τεχνικές, αλλά αναπτύσσοντας και καινούριες. Κάποιες αξιοσημείωτες προσπάθειες ήταν αυτές των G. Tesauro [5] στο backgammon, J. Baxter, A. Tridgell και L. Weaver [6] στο σκάκι, R. Ekker [7] στο παιχνίδι go, J. Schaeffer [8] στη ντάμα, D. K. Olson [9] στην τρίλιζα, M. Pfeiffer [10] στο παιχνίδι Settlers of Catan, και R. K. Winder [11] στο παιχνίδι

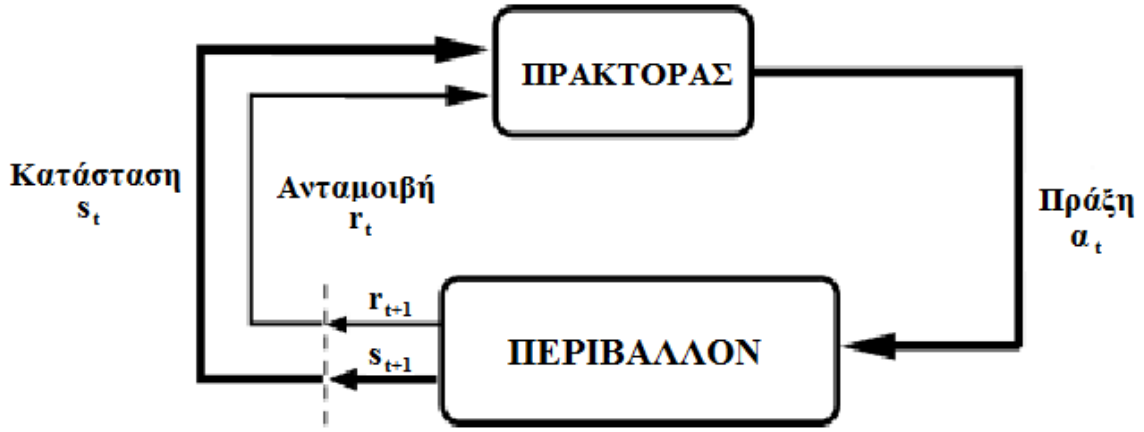
Dominion. Τα αποτελέσματα κάποιων ήταν εντυπωσιακά, όπως στο backgammon, όπου ο αλγόριθμος που αναπτύχθηκε ανέπτυξε στρατηγικές που υιοθετήθηκαν ακόμα και από παίκτες παγκοσμίου επιπέδου [5]. Κάποιες φορές τα αποτελέσματα ήταν ενθαρρυντικά, αφήνοντας περιθώριο όμως για περαιτέρω έρευνα, όπως στην περίπτωση του Dominion.

Ο Winder [11] χρησιμοποίησε την μέθοδο της εξαναγκασμένης μάθησης και πιο συγκεκριμένα τρεις διαφορετικές προσεγγίσεις , τον αλγόριθμο TD(λ), την μέθοδο hill climbing και γενετικούς αλγόριθμους. Οι παραπάνω μέθοδοι δοκιμάστηκαν εναντίον τριών αντιπάλων, προγραμματισμένων με ντετερμινιστικές στρατηγικές, χαμηλής , μεσαίας και υψηλής δυναμικής αντίστοιχα. Καθώς τα αποτελέσματα δεν ήταν ικανοποιητικά, μια τέταρτη μέθοδος που βασιζόταν στους γενετικούς αλγόριθμους, αλλά χρησιμοποιούσε, δύο νευρωνικά δίκτυα, δοκιμάστηκε, και κατάφερε να υπερκεράσει και τον αντίπαλο υψηλής στρατηγικής, κερδίζοντας το 60% των παιχνιδιών εναντίον. Παρόλη την επιτυχία, η μέθοδος αυτή δεν δοκιμάστηκε ποτέ εναντίον ανθρώπων, που είναι και το τελικό ζητούμενο.

Στην παρούσα εργασία θα εκπαιδεύσουμε έναν «πράκτορα» με μεθόδους εξαναγκασμένης μάθησης, και πιο συγκεκριμένα με την τεχνική Q Learning, στο παιχνίδι Dominion. Η γλώσσα προγραμματισμού που επιλέξαμε είναι η Java, κυρίως λόγω ταχύτητας.

ΚΕΦΑΛΑΙΟ 1:
Εξαναγκασμένη μάθηση

Η μέθοδος της εξαναγκασμένης μάθησης έχει ως στόχο την εκπαίδευση ενός «πράκτορα», μέσω της αλληλεπίδρασης με το περιβάλλον. Η αλληλεπίδραση αυτή διαιρείται σε χρονικά βήματα t. Σε κάθε χρονικό βήμα, η κατάσταση του περιβάλλοντος s_t δίνεται στον πράκτορα, αυτός επιλέγει μια ενέργεια a_t ανάλογα με την κατάσταση s_t, και το περιβάλλον βρίσκεται πλέον σε μια νέα κατάσταση s_{t+1}. Μετά από κάθε ενέργεια ο πράκτορας λαμβάνει από το περιβάλλον μια αριθμητική ανταμοιβή r_{t+1} καθώς και την νέα κατάσταση του περιβάλλοντος s_{t+1} (εικόνα 1.1).



Εικόνα 1.1 Η αλληλεπίδραση του πράκτορα με το περιβάλλον

Στόχος της εκπαίδευσης είναι ο πράκτορας να μάθει την κατάλληλη τακτική (π) επιλογής ενεργειών ώστε να μεγιστοποιήσει την συνολική αριθμητική ανταμοιβή R. Αυτό επιτυγχάνεται μέσω της συνάρτησης $Q(s_t, a_t)$, η οποία είναι ίση με την αναμενόμενη συνολική ανταμοιβή R_t, αν ο πράκτορας κατά το χρονικό βήμα t επιλέξει την ενέργεια a_t όταν το περιβάλλον βρίσκεται στην κατάσταση s_t. Όπως είναι προφανές αν γνωρίζαμε εκ των προτέρων τις τιμές της συνάρτησης $Q(s_t, a_t)$, για κάθε τιμή των s_t, a_t, τότε η βέλτιστη τακτική (π^*) θα ήταν η «greedy» (άπληστη), δηλαδή αυτή που θα επέλεγε την ενέργεια a_t για την οποία η $Q(s_t, a_t)$, θα ήταν μέγιστη. Το πρόβλημα της εξαναγκασμένης μάθησης έγκειται λοιπόν στην εύρεση μιας διαδικασίας προσέγγισης των τιμών της $Q(s_t, a_t)$.

Μπορούμε να διαχωρίσουμε τα προβλήματα εξαναγκασμένης μάθησης σε δύο κατηγορίες, αυτά στα οποία η διαδικασία συνεχίζεται επ' άπειρον και αυτά στα οποία η διαδικασία χωρίζεται σε αυτοτελή επεισόδια, όπως στην περίπτωση που μας ενδιαφέρει. Σε τέτοια προβλήματα υπάρχουν καταστάσεις του περιβάλλοντος για τις οποίες το επεισόδιο τερματίζεται. Συμβολίζουμε με t τα βήματα που κάνει ο πράκτορας κατά τη διαδικασία και με T το τελικό βήμα που οδηγεί σε τερματική κατάσταση, τότε s_1 είναι η

αρχική κατάσταση του περιβάλλοντος, s_t η κατάσταση μετά την ενέργεια a_{t-1} και r_t η αντίστοιχη ανταμοιβή. Τότε η συνολική αναμενόμενη ανταμοιβή κατά το βήμα t είναι:

$$R_t = r_{t+1} + r_{\tau+2} + \ldots + r_T \ (1)$$

Μια άλλη προσέγγιση είναι να κάνουμε «έκπτωση» στις μελλοντικές ανταμοιβές κατά έναν παράγοντα γ με $0 \leq \gamma \leq 1$. Τότε θα έχουμε:

$$R_t = r_{t+1} + \gamma \cdot r_{t+2} + \gamma^2 \cdot r_{t+3} \ldots = \sum_{k=0}^{T-t-1} \gamma^k \cdot r_{k+t+1} \ (2)$$

Αν γ=0 τότε ο πράκτορας είναι «μυωπικός», δηλαδή λαμβάνει υπόψη του μόνο την αμέσως επόμενη ανταμοιβή. Η συνάρτηση αξίας των ενεργειών ορίζεται τότε ως εξής:

$$Q(s,a) = E\{R_t | s_t = s, a_t = a\} = E\left\{ \sum_{k=0}^{T-t-1} \gamma^k \cdot r_{k+t+1} | s_t = s, a_t = a \right\} \ (3)$$

Το σύμβολο Ε, δηλώνει την αναμενόμενη (expected) ανταμοιβή. Σε κάποια προβλήματα είναι χρήσιμο να ορίσουμε και την συνάρτηση αξίας των καταστάσεων $V(s)$.

$$V(s) = E\{R_t | s_t = s\} = E\left\{ \sum_{k=0}^{T-t-1} \gamma^k \cdot r_{k+t+1} | s_t = s \right\} \ (4)$$

1.1 Ιδιότητα Markov

Ο Andrey Andreyevich Markov ήταν ένας Ρώσος μαθηματικός, μέρος της δουλειάς του οποίου ήταν αφιερωμένο στη στατιστική και τις πιθανότητες [12], και έδωσε το όνομα του σε μια σημαντική ιδιότητα των καταστάσεων του περιβάλλοντος. Μια κατάσταση του περιβάλλοντος έχει την ιδιότητα Markov, αν περιέχει όλες τις σημαντικές πληροφορίες που θα οδηγήσουν τον πράκτορα να λάβει μια απόφαση, αλλά όχι ολόκληρο το ιστορικό, για το πώς το περιβάλλον έφτασε σε αυτήν την κατάσταση. Για παράδειγμα, η θέση των κομματιών σε μια σκακιέρα δίνει όλες τις απαραίτητες πληροφορίες που χρειάζονται για να αποφασιστεί η επόμενη κίνηση, αλλά δεν περιέχει πληροφορίες για τις προηγούμενες κινήσεις που οδήγησαν σε αυτή την θέση.

Με πιο μαθηματικό τρόπο ο ορισμός του αν ένα σύστημα έχει την ιδιότητα Markov είναι ο εξής: έστω ότι κατά το χρονικό βήμα t, το περιβάλλον βρίσκεται στην κατάσταση s_t και ο πράκτορας επιλέγει την ενέργεια a_t. Αν η πιθανότητα το περιβάλλον να βρεθεί σε κατάσταση $s_{t+1} = s'$ και ο πράκτορας να λάβει ανταμοιβή $r_{t+1} = r$ είναι συνάρτηση μόνο της κατάστασης s_t και της ενέργειας a_t, τότε το σύστημα έχει την ιδιότητα Markov, αν δηλαδή η πιθανότητα (5) είναι ίση με την πιθανότητα (6)

$$P(s_{t+1} = s', r_{t+1} = r \mid s_t, a_t) \ (5)$$

$$P(s_{t+1} = s', r_{t+1} = r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, \ldots \ldots, r_1, a_0, s_0) \ (6)$$

Ένα πρόβλημα εξαναγκασμένης μάθησης που έχει την ιδιότητα Markov, ονομάζεται διαδικασία απόφασης Markov ή MDP (Markov Decision Process). Αν επιπλέον ο χώρος των ενεργειών και των καταστάσεων είναι πεπερασμένος, τότε ονομάζεται πεπερασμένο MDP. Η πιθανότητα μετάβασης από μια κατάσταση s σε μια άλλη κατάσταση s'μέσω μιας ενέργειας a, φαίνεται παρακάτω (7).

$$P_{ss'}^{a} = P(s_{t+1} = s' | s_t = s\,, a_t = a)\ (7)$$

Η αμέσως επόμενη ανταμοιβή κατά την μετάβαση από την s στην s'μέσω της a, φαίνεται στην (8).

$$R_{ss'}^{a} = E(r_{t+1}\,|s_t = s\,, a_t = a\,, s_{t+1} = s'\,)\ (8)$$

Αν $\pi(s,a)$ η πιθανότητα να επιλεγεί η ενέργεια α στην κατάσταση s, τότε, όπως αποδείχθηκε από τον Bellman [4], μπορούμε να μετασχηματίσουμε τη σχέση υπολογισμού της συνάρτησης $V(s)$ ως εξής:

$$\begin{aligned}
V(s) &= E\{R_t | s_t = s\} \\
&= E\left\{\sum_{k=0}^{T-t-1} \gamma^k \cdot r_{k+t+1}\,|s_t = s\right\} \\
&= E\left\{r_{t+1} + \gamma \cdot \sum_{k=0}^{T-t-2} \gamma^k \cdot r_{k+t+2}\,|s_t = s\right\} \\
&= \sum_{\alpha} \pi(s,a) \cdot \sum_{s'} P_{ss'}^{a} \cdot \left[R_{ss'}^{a} + \gamma \cdot E\left\{\sum_{k=0}^{T-t-2} \gamma^k \cdot r_{k+t+2}\,|s_{t+1} = s'\right\}\right] \\
&= \sum_{\alpha} \pi(s,a) \cdot \sum_{s'} P_{ss'}^{a} \cdot \left[R_{ss'}^{a} + \gamma \cdot V(s')\right]\ (9)
\end{aligned}$$

Και αντίστοιχα για την $Q(s,a)$:

$$\begin{aligned}
Q(s,a) &= E\{R_t | s_t = s, a_t = a\} \\
&= E\left\{\sum_{k=0}^{T-t-1} \gamma^k \cdot r_{k+t+1}\,|s_t = s, a_t = a\right\} \\
&= E\left\{r_{t+1} + \gamma \cdot \sum_{k=0}^{T-t-2} \gamma^k \cdot r_{k+t+2}\,|s_t = s, a_t = a\right\} \\
&= \sum_{s'} P_{ss'}^{a} \cdot \left[R_{ss'}^{a} + \gamma \cdot E\left\{\sum_{k=0}^{T-t-2} \gamma^k \cdot r_{k+t+2}\,|s_{t+1} = s'\right\}\right] \\
&= \sum_{s'} P_{ss'}^{a} \cdot \left[R_{ss'}^{a} + \gamma \cdot V(s')\right]\ (10)
\end{aligned}$$

1.2 Συνάρτηση βελτιστοποίησης Bellman

Η λύση ενός προβλήματος εξαναγκασμένης μάθησης έγκειται στο να καθορίσουμε την βέλτιστη τακτική π^* επιλογής ενεργειών, ώστε να μεγιστοποιείται η ανταμοιβή. Για αυτή τη βέλτιστη τακτική πρέπει η αναμενόμενη ανταμοιβή για κάθε κατάσταση να είναι μεγαλύτερη ή ίση, από την αναμενόμενη ανταμοιβή οποιασδήποτε άλλης τακτικής π για την αντίστοιχη κατάσταση. Με άλλα λόγια πρέπει:

$$V^{\pi^*}(s) \geq V^{\pi}(s)\ \ \forall s\ \ (11)$$

Μπορεί να υπάρχουν περισσότερες από μία βέλτιστες τακτικές, αλλά αποδεικνύεται [4] ότι όλες έχουν την ίδια συνάρτηση αξίας καταστάσεων και συνάρτηση αξίας ενεργειών τις οποίες συμβολίζουμε με $V^*(s)$ και $Q^*(s,a)$. Άρα θα είναι:

$$V^*(s) = \max_{\pi} V^{\pi}(s)\ \forall s\ \ (12) \text{ και } Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a)\ \forall s\ και\ \forall \alpha\ \ (13)$$

Παρακάτω αποδεικνύεται η σχέση βελτιστοποίησης Bellman

$$\begin{aligned} V^*(s) &= \max_a Q^{\pi^*}(s,a) \\ &= \max_a E_{\pi^*}\{R_{t+1} \mid s_t = s, a_t = a\} \\ &= \max_a E_{\pi^*}\left\{\sum_{k=0}^{T-t-1} \gamma^k \cdot r_{k+t+1} \,|s_t = s, a_t = a\right\} \\ &= \max_a E_{\pi^*}\left\{r_{t+1} + \gamma \cdot \sum_{k=0}^{T-t-2} \gamma^k \cdot r_{k+t+2} \,|s_t = s, a_t = a\right\} \\ &= \max_a E\{r_{t+1} + \gamma \cdot V^*(s_{t+1}) | s_t = s, a_t = a\} \\ &= \max_a \sum_{s\prime} P^a_{ss\prime} \cdot \left[R^a_{ss\prime} + \gamma \cdot V^*(s')\right]\ (12) \end{aligned}$$

Η αντίστοιχη σχέση βελτιστοποίησης για την συνάρτηση αξίας ενεργειών είναι:

$$\begin{aligned} Q^*(s,a) &= E\left\{r_{t+1} + \gamma \cdot \max_{a'} Q(s_{t+1}, a') \,|s_t = s, a_t = a\right\} \\ &= \sum_{s\prime} P^a_{ss\prime} \cdot \left[R^a_{ss\prime} + \gamma \cdot \max_{a'} Q(s', a')\right]\ (13) \end{aligned}$$

Για πεπερασμένα MDP με N καταστάσεις, οι εξισώσεις (12) και (13) αποτελούν συστήματα εξισώσεων NxN, τα οποία έχουν μοναδικές λύσεις. Αν καταφέρουμε να βρούμε αυτές τις λύσεις τότε η βέλτιστη τακτική π^* είναι αυτή που είναι greedy ως προς την $V^*(s)$, δηλαδή αυτή που επιλέγει την ενέργεια που οδηγεί στην κατάσταση με την μεγαλύτερη τιμή $V^*(s)$.

1.3 Δυναμικός προγραμματισμός

Για προβλήματα όπου ο χώρος των καταστάσεων είναι μεγάλος, η λύση του συστήματος των εξισώσεων (12) και (13) είναι πρακτικά αδύνατη. Έτσι έχουν αναπτυχθεί διάφοροι μέθοδοι για την προσέγγιση των τιμών της $V^*(s)$. Μία από αυτές είναι η μέθοδος του δυναμικού προγραμματισμού DP (dynamic programming). Για να είναι εφαρμόσιμη η μέθοδος πρέπει να έχουμε απόλυτη γνώση του περιβάλλοντος. Με άλλα λόγια να γνωρίζουμε όλες τις πιθανότητες μετάβασης $P^a_{ss\prime}$ και τις αναμενόμενες ανταμοιβές $R^a_{ss\prime}$. Αρχικά επιλέγουμε μια αυθαίρετη τακτική π και δίνουμε αυθαίρετες τιμές στην $V(s)$. Στη συνέχεια ανανεώνουμε τις τιμές της $V(s)$ βάση της σχέσης (14).

$$V_{k+1} = \sum_{\alpha} \pi(s,a) \cdot \sum_{s'} P^a_{ss\prime} \cdot \left[R^a_{ss\prime} + \gamma \cdot V_k(s')\right]\ (14)$$

Αποδεικνύεται [13] ότι όταν το $k \to \infty$ τότε $V_{k+1} \to V^*$. Στην ενέργεια εφαρμόζουμε τον παρακάτω αλγόριθμο:

Αλγόριθμος DP

$V(s) = 0$ *για κάθε* s

$\Delta = \xi$

Για όσο $\Delta < \theta$ (*όπου* θ, ξ *αυθαίρετες τιμές με* $\theta < \xi$) *επανέλαβε*:

Για καθε s

$$\upsilon \leftarrow V(s)$$

$$V(s) \leftarrow \sum_{\alpha} \pi(s,a) \cdot \sum_{s'} P^{a}_{ss'} \cdot [R^{a}_{ss'} + \gamma \cdot V(s')]$$

$$\Delta \leftarrow \max(\Delta, |\upsilon - V(s)|)$$

Έπειτα επιλέγουμε ως νέα τακτική την π', η οποία είναι greedy ως προς την συνάρτηση αξιών καταστάσεων που υπολογίσαμε και επαναλαμβάνουμε τον ίδιο αλγόριθμο για την π' και ούτω καθεξής. Η διαδικασία μπορεί να τερματιστεί όταν δεν θα υπάρχει αλλαγή σε έναν κύκλο για την τακτική π.

Η μέθοδος DP αν και εγγυημένα συγκλίνει προς την βέλτιστη τακτική π^*, δεν είναι εύκολα εφαρμόσιμη. Αυτό συμβαίνει γιατί σε περίπλοκα προβλήματα ο υπολογισμός των ποσοτήτων P^{a}_{ss}, και R^{a}_{ss}, που είναι απαραίτητες, είναι ιδιαιτέρως δύσκολος, και ο υπολογιστικός χρόνος που απαιτείται υπερβολικά μεγάλος.

1.4 Monte Carlo Trees

Η μέθοδος Monte Carlo Trees (MCT) διαφέρει σημαντικά σε σχέση με τον δυναμικό προγραμματισμό σε ένα βασικό σημείο. Ενώ στον δυναμικό προγραμματισμό προσπαθούμε να προσεγγίσουμε τις τιμές της $V^*(s)$ θεωρητικά, η MCT βασίζεται στη δοκιμή και αποτυχία (trial and error). Ξεκινάμε πάλι με μια αυθαίρετη τακτική π και δίνουμε μηδενικές τιμές στην $V(s)$ για όλες τις καταστάσεις s. Δημιουργούμε ένα εικονικό επεισόδιο της διαδικασίας και καταγράφουμε τις ανταμοιβές για κάθε κατάσταση και αποδίδουμε τιμές στην $V(s)$ για κάθε κατάσταση s που παρατηρήθηκε στο επεισόδιο ως εξής:

$$V(s_t) = R_t = r_{t+1} + r_{\tau+2} + \ldots + r_T \quad (15)$$

Επαναλαμβάνουμε και για κάθε νέα εμφάνιση κάθε κατάστασης s, ανανεώνουμε την τιμή της $V(s)$, ως τον μέσο όρο των ανταμοιβών όλων των εμφανίσεων. Αυτό που πρέπει να προσέξουμε έτσι ώστε η μέθοδος να έχει σωστά αποτελέσματα, είναι ότι η τακτική π πρέπει να είναι τέτοια, ώστε να υπάρχει μη μηδενική πιθανότητα να επιλεγούν ενέργειες που να οδηγούν σε όλες τις πιθανές καταστάσεις s. Σε καμία περίπτωση δηλαδή η τακτική δεν πρέπει να είναι greedy. Μια τεχνική που αντιμετωπίζει αυτό το πρόβλημα είναι αυτή των exploring starts (εξερευνητικών αρχών). Η επιλογή της τακτικής π είναι τέτοια ώστε να δίνει μη μηδενικές πιθανότητες επιλογής για όλες τις ενέργειες, αλλά σε ένα όριο να συγκλίνει στην greedy τακτική.

On policy Monte Carlo Trees

Η μέθοδος on policy MCT χρησιμοποιεί τακτικές οι οποίες είναι ε-soft. Αυτές επιλέγουν την ενέργεια με την μεγαλύτερη αξία, με πιθανότητα 1-ε, όπου ε<1, και μία τυχαία ενέργεια με πιθανότητα ε. Αν A(s) το πλήθος των ενεργειών στην κατάσταση s, τότε κάθε ενέργεια θα επιλεγεί με πιθανότητα $\frac{\varepsilon}{A(s)}$, εκτός της ενέργειας με την μέγιστη αξία που επιλέγεται με πιθανότητα $1-\varepsilon+\frac{\varepsilon}{A(s)}$. Ο αλγόριθμός φαίνεται παρακάτω:

Αλγόριθμος on policy MCT

$Q(s,a) = 0$ *για κάθε* s, a
Δημιούργησε άδειες λίστες $R(s,a)$ *για κάθε* s, a
π *μία* $\varepsilon - soft$ *τακτική*
επανέλαβε:
    Δημιούργησε ένα επεισόδιο με την τακτική π
    Για καθε s, a *που εμφανίζονται στο επεισόδιο*
        $R \leftarrow$ *ανταμοιβή που ακολουθεί την πρώτη εμφάνιση των* s, a
        πρόσθεσε το στοιχείο R *στη λίστα* $R(s,a)$
        $Q(s,a) \leftarrow$ *ο μέσος όρος των στοιχείων της* $R(s,a)$
    Για καθε s *που εμφανίζεται στο επεισόδιο*
        $\alpha^* \leftarrow argmax_a Q(s,a)$
        Για όλα τα α *της κατάστασης* s

$$\pi(s,a) \leftarrow \begin{cases} 1-\varepsilon+\frac{\varepsilon}{A(s)} & \alpha\nu\ \alpha=\alpha^* \\ \frac{\varepsilon}{A(s)} & \alpha\nu\ \alpha\neq\alpha^* \end{cases}$$

Off policy Monte Carlo Trees

Σε αυτή τη μέθοδο χρησιμοποιούμε μια τυχαία τακτική π' (η οποία δεν πρέπει να είναι greedy) για να υπολογίσουμε τη συνάρτηση αξιών της greedy τακτικής π, αρκεί κάθε κατάσταση που εμφανίζεται με την π' να εμφανίζεται και με την π. Αποδεικνύεται [13] ότι η συνάρτηση αξιών καταστάσεων της π είναι:

$$V(s) = \frac{\sum_{i=1}^{n} \prod_{k}^{T} \frac{\pi(s_k, a_k)}{\pi'(s_k, a_k)} \cdot R_i(s)}{\sum_{i=1}^{n} \prod_{k}^{T} \frac{\pi(s_k, a_k)}{\pi'(s_k, a_k)}} \quad (16)$$

Όπου n οι φορές που κατά την εκτέλεση ενός επεισοδίου ακολουθώντας την τακτική π' συναντήσαμε το ζευγάρι s, α στο χρονικό βήμα k. Ο αλγόριθμος ακολουθεί παρακάτω.

Αλγόριθμος off policy MCT

$Q(s,a) \leftarrow$ *αυθαίρετες τιμές για κάθε* s, a

$N(s,a) \leftarrow 0$

$D(s,a) \leftarrow 0$

π *μία αυθαίρετη ντετερμινιστική τακτική*

π' *μία* $\varepsilon - soft$ *τακτική*

επανέλαβε:

Δημιούργησε ένα επεισόδιο με την τακτική π:

$s_0, a_0, r_1, s_1, a_1, r_2, \dots \dots \dots \dots \dots, s_{T-1}, a_{T-1}, r_T, s_T$

Έστω α_τ *η τελευταία πράξη στο επεισόδιο που δεν επιλέγεται από την* π

Για κάθε s, a *στο επεισόδιο μετά το βήμα* τ:

$t \leftarrow$ *η πρώτη εμφάνιση των* s, a

$$w \leftarrow \prod_{k=t+1}^{T-1} \frac{1}{\pi'(s_k, a_k)}$$

$N(s,a) \leftarrow N(s,a) + w \cdot R_t$

$D(s,a) \leftarrow D(s,a) + w$

$Q(s,a) \leftarrow \frac{N(s,a)}{D(s,a)}$

Για κάθε s

$\pi(s) \leftarrow argmax_a Q(s,a)$

1.5 Temporal Difference

Μια άλλη δημοφιλής μέθοδος είναι αυτή η Temporal Difference TD (προσωρινή διαφορά). Και αυτή όπως τα MCT στηρίζεται στη δοκιμή και αποτυχία, με μια σημαντική διαφορά. Αντί να περιμένουμε ως το τέλος του επεισοδίου για να επικαιροποιήσουμε τις τιμές της συνάρτησης αξιών, το κάνουμε μετά από κάθε χρονικό βήμα t ως εξής:

$$V(s_t) \leftarrow V(s_t) + a \cdot [r_{t+1} + \gamma \cdot V(s_{t+1}) - V(s_t)] \ (17)$$

Η επικαιροποίηση της V(s) στηρίζεται εν μέρει στην εμπειρία (r_{t+1}) όπως στα MCT και εν μέρει σε εκτίμηση όπως στον DP.

S.A.R.S.A. on policy TD

Για την εύρεση της βέλτιστης τακτικής είναι προτιμότερο να εκτιμήσουμε την Q(s,α), και να επιλέξουμε μια ε-greedy τακτική ως προς την συνάρτηση αυτή.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + a \cdot [r_{t+1} + \gamma \cdot Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \ (18)$$

Ο αλγόριθμος φαίνεται παρακάτω.

Αλγόριθμος S.A.R.S.A. on policy TD

$Q(s,a) \leftarrow$ *αυθαίρετες τιμές για κάθε* s, a

επανέλαβε για κάθε επεισόδιο:

Για την αρχική s *δίαλεξε μια πράξη* a βάσει της τακτικής *που προκύπτει από την τακτική* $Q(s,a)$ $(\varepsilon - greedy)$

επανέλαβε για κάθε βήμα του επεισοδίου:

κάνε μια πράξη a *και παρατήρησε τα* r *και* s'

διάλεξε από την s'*μια πράξη* a' βάσει της τακτικής *που προκύπτει από την τακτική* $Q(s,a)$ $(\varepsilon - greedy)$

$Q(s,a) \leftarrow Q(s,a) + a \cdot [r_{t+1} + \gamma \cdot Q(s',a') - Q(s,a)]$

$s \leftarrow s'$

$a \leftarrow a'$

έως ότου η s *να είναι τερματική*

Q-learning. off policy TD

Σε αυτή την παραλλαγή της TD αντί να επικαιροποιούμε την Q(s,a) βάσει της επόμενης ενέργειας, το κάνουμε βάσει της καλύτερης ενέργειας από τις πιθανές της s'.

Αλγόριθμος Q-learning. off policy TD

$Q(s,a) \leftarrow$ *αυθαίρετες τιμές για κάθε* s, a

επανέλαβε για κάθε επεισόδιο:

επανέλαβε για κάθε βήμα του επεισοδίου:

δίαλεξε μια πράξη a *από την* s βάσει της τακτικής *που προκύπτει από την τακτική* $Q(s,a)$ $(\varepsilon - greedy)$

κάνε την πράξη a *και παρατήρησε τα* r *και* s'

$Q(s,a) \leftarrow Q(s,a) + a \cdot \left[r_{t+1} + \gamma \cdot \max_{a'} Q(s',a') - Q(s,a)\right]$

$s \leftarrow s'$

έως ότου η s *να είναι τερματική*

ΚΕΦΑΛΑΙΟ 2:
Το επιτραπέζιο παιχνίδι Dominion

Το επιτραπέζιο παιχνίδι Dominion παίζεται με κάρτες. Υπάρχουν τρεις κατηγορίες καρτών, οι κάρτες θησαυρών, οι κάρτες νίκης και οι κάρτες βασιλείου. Κάθε κάρτα έχει μία τιμή στην οποία μπορείς να την αγοράσεις (αναγράφεται κάτω αριστερά). Οι κάρτες θησαυρών χρησιμοποιούνται προκειμένου να αγοραστούν νέες κάρτες. Υπάρχουν τριών ειδών κάρτες θησαυρών, τα χάλκινα (60 κάρτες), τα οποία έχουν τιμή μηδέν και αξία ένα νόμισμα, τα ασημένια (40 κάρτες), με τιμή τρία και αξία δύο νομίσματα και τα χρυσά (30 κάρτες), με τιμή έξι και αξία τρία νομίσματα.

Εικόνα 2.1 Κάρτες θησαυρών

Οι κάρτες νίκης, όπως προδίδει και το όνομά τους, δίνουν πόντους νίκης. Και από αυτές υπάρχουν τρία είδη, τα κτήματα (24 κάρτες), με τιμή δύο, που δίνουν ένα πόντο νίκης το καθένα, τα δουκάτα (12 κάρτες), με τιμή πέντε, που δίνουν τρεις πόντους νίκης το καθένα και οι επαρχίες (12 κάρτες), με τιμή οκτώ, που δίνουν έξι πόντους νίκης το καθένα. Τέλος υπάρχουν 25 διαφορετικά είδη καρτών βασιλείου (10 κάρτες από το κάθε είδος). Κάθε είδος τέτοιων καρτών έχει διαφορετική τιμή, αλλά και διαφορετικές ιδιότητες. Σκοπός του παιχνιδιού είναι να συγκεντρώσεις όσο το δυνατόν περισσότερους πόντους νίκης.

Εικόνα 2.2 Κάρτες νίκης

Κάθε παίκτης ξεκινά το παιχνίδι με μία τράπουλα δέκα καρτών, 7 χάλκινα και 3 κτήματα. Τα υπόλοιπα χάλκινα τοποθετούνται σε μία στοίβα, όπως επίσης τα ασημένια και τα χρυσά. Φτιάχνονται επίσης τρεις στοίβες με τις κάρτες νίκης, που περιέχουν 12 κάρτες από το κάθε είδος. Τέλος επιλέγονται 10 από τα 25 διαφορετικά είδη καρτών βασιλείου και κάθε είδος τοποθετείται σε στοίβες των δέκα καρτών. Κάθε παίκτης τραβά πέντε κάρτες από την τράπουλά του, και το παιχνίδι ξεκινά. Ο γύρος κάθε παίκτης αποτελείται από τρεις φάσεις, τη φάση ενεργειών, τη φάση αγορών και τη φάση εκκαθάρισης.

Εικόνα 2.3 Κάρτες βασιλείου

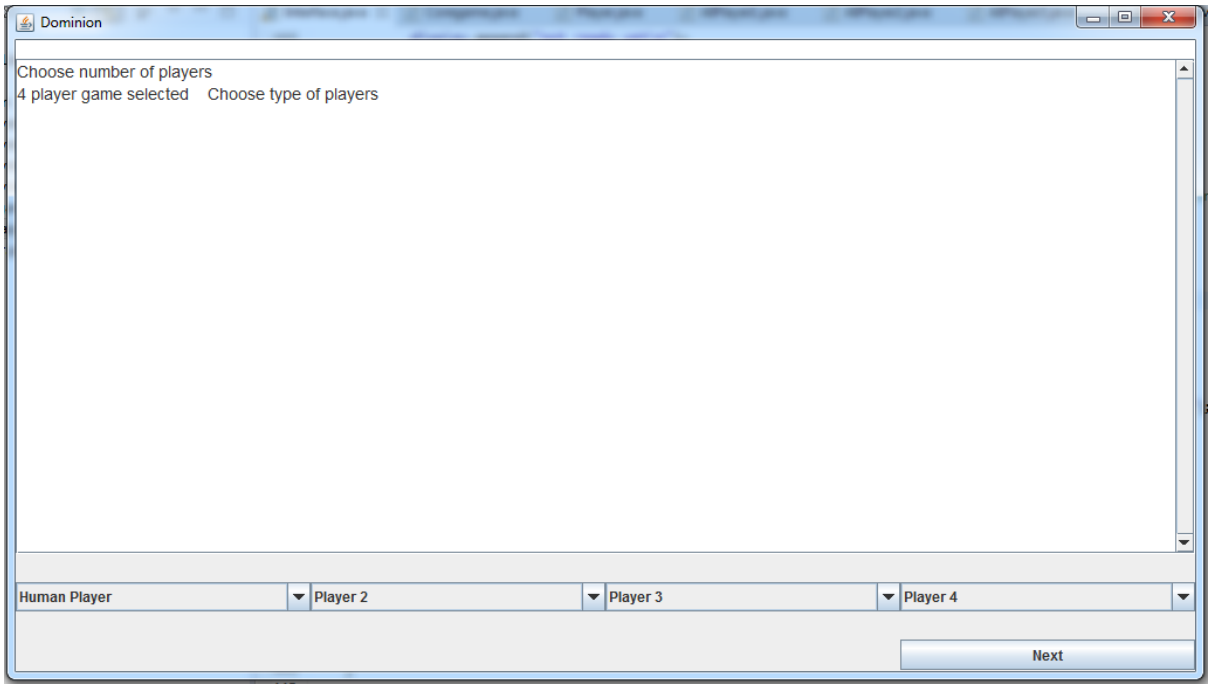
Κατά τη φάση ενεργειών μπορούν να παιχτούν κάρτες βασιλείου που έχει ο παίκτης στο χέρι του. Το παίξιμο κάθε κάρτας βασιλείου αποτελεί μία ενέργεια. Σε κάθε γύρο, ο παίκτης μπορεί να εκτελέσει μόνο μία ενέργεια. Παίζοντας μια κάρτα βασιλείου ο παίκτης αποκομίζει διαφορετικά οφέλη, ανάλογα με το είδος της κάρτας. Κάποια από αυτά είναι να τραβήξει επιπλέον κάρτες από την τράπουλα του, να εκτελέσει επιπλέον ενέργειες, να κερδίσει νομίσματα και άλλα. Κάθε κάρτα βασιλείου που παίζεται μεταφέρεται στα σκάρτα.

Στη φάση των αγορών ο παίκτης μπορεί να αγοράσει κάρτες από αυτές που είναι διαθέσιμες στις διάφορες στοίβες. Πληρώνει την αξία της κάρτας που αγοράζει χρησιμοποιώντας νομίσματα που κέρδισε παίζοντας κάρτες βασιλείου ή παίζοντας κάρτες θησαυρών. Σε κάθε γύρο μπορεί να αγοράσει μόνο μία κάρτα, εκτός και αν κατά τη φάση ενεργειών, κάποια κάρτα βασιλείου του έδωσε επιπλέον αγορές. Οι κάρτες που αγοράζονται και οι θησαυροί που χρησιμοποιήθηκαν μεταφέρονται στα σκάρτα.

Κατά τη φάση της εκκαθάρισης ο παίκτης πετά τις υπόλοιπες κάρτες που δεν χρησιμοποίησε στα σκάρτα και τραβά πέντε νέες κάρτες από την τράπουλα του. Αν δεν υπάρχουν αρκετές, ανακατεύει τα σκάρτα, τα τοποθετεί στη θέση της τράπουλας και τράβα όσες κάρτες, έως ότου να έχει πέντε στα χέρια του. Στη συνέχεια ξεκινά ο γύρος του επόμενου παίκτη και ούτω καθεξής. Το παιχνίδι συνεχίζεται έως ότου εξαντληθεί η στοίβα με τις κάρτες των επαρχιών ή εναλλακτικά εξαντληθούν τρεις οποιαδήποτε στοίβες καρτών. Ο παίκτης που έχει τότε τους περισσότερους πόντους νίκης αναδεικνύεται νικητής.

ΚΕΦΑΛΑΙΟ 3:
JDominion

Ο στόχος της παρούσης εργασίας είναι η ανάπτυξη ενός πράκτορα τεχνητής νοημοσύνης και η εκπαίδευση του στις στρατηγικές του επιτραπέζιου παιχνιδιού Dominion, με τεχνικές εξαναγκασμένης μάθησης. Στα πλαίσια αυτού του στόχου, τα πρώτα βήματα που ακολουθήθηκαν, πέραν βεβαίως της μελέτης της θεωρίας της εξαναγκασμένης μάθησης, ήταν η εξοικείωση με τους κανόνες του παιχνιδιού, αλλά και με τη γλώσσα προγραμματισμού Java. Αναπτύχθηκε λοιπόν αλγόριθμος προσομοίωσης του παιχνιδιού σε εικονικό περιβάλλον (εικόνα 3.1), ο JDominion. Κατά την πρώτη φάση υλοποίησης, ενσωματώθηκαν στον αλγόριθμο, μόνο οι κάρτες βασιλείου «Σιδεράς» (Smithy), η οποία επιτρέπει στον παίκτη να τραβήξει τρεις κάρτες, και το «Παρεκκλήσι» (Chapel), που δίνει το δικαίωμα στον παίκτη να πετάξει εκτός τράπουλας έως τρεις κάρτες από το χέρι του (όχι όμως το «Παρεκκλήσι»). Όλες οι κάρτες που ενσωματώθηκαν σε αυτή τη φάση φαίνονται στον πίνακα 3.1.1.



Εικόνα 3.1 Προσομοίωση του παιχνιδιού Dominion

3.1 AI αντίπαλοι

Ο αλγόριθμος αυτός επιτρέπει την επιλογή αριθμού παικτών (δύο, τρεις ή τέσσερις), τύπου παικτών, καθώς και αριθμό παιχνιδιών. Ο τύπος των παικτών μπορεί να είναι άνθρωπος (Human player), ο υπό εκπαίδευση πράκτορας (RL player), ή διάφοροι τύποι AI παικτών με εναλλακτικές στρατηγικές, που θα χρησιμοποιηθούν ως αντίπαλοι κατά την εκπαίδευση. Αν στο παιχνίδι συμμετέχει άνθρωπος, τότε ο αριθμός των παιχνιδιών ορίζεται αυτόματα να είναι ένα. Οι τύποι των AI παικτών είναι Random, Money, Greedy, Blacksmith και Chapel, και κάθε ένας από αυτούς έχει διαφορετική στρατηγική. Ο Random επιλέγει σε κάθε γύρω να αγοράσει μία τυχαία κάρτα, ενώ δεν χρησιμοποιεί κάρτες βασιλείου ακόμα και να έχει κάποια στο χέρι του. Ο Money αγοράζει μόνο

ασημένια και χρυσά, έως ότου βρεθούν στην τράπουλά του δύο χρυσά. Στην συνέχεια βάσει των νομισμάτων που έχει στο χέρι του, επιλέγει την πιο ακριβή κάρτα μεταξύ των «κτήμα», «ασημένιο», «δουκάτο», «χρυσό» ή «επαρχία». Ο Greedy επιλέγει να αγοράσει την πιο ακριβή κάρτα μεταξύ των «κτήμα», «ασημένιο», «δουκάτο», «χρυσό», «επαρχία» ή «σιδεράς». Αν στο χέρι του στην αρχή του γύρου έχει «σιδερά», τον χρησιμοποιεί. Ο Blacksmith αρχικά αγοράζει την πιο ακριβή κάρτα που μπορεί εκ των «ασημένιο», «σιδεράς», «χρυσό» και «επαρχία». Αν έχει στο χέρι του σιδερά, τον χρησιμοποιεί. Όταν ο αριθμός των καρτών στη στοίβα των επαρχιών γίνει μικρότερος του τέσσερα (δηλαδή όταν το παιχνίδι πλησιάζει προς το τέλος), αγοράζει μόνο κάρτες νίκης (όχι όμως κατάρες). Τέλος ο Chapel στους πρώτους δύο γύρους αγοράζει ένα «ασημένιο» και ένα «παρεκκλήσι». Στη συνέχεια και έως τον δέκατο πέμπτο γύρο «ασημένια», «χρυσά» ή «επαρχίες», ενώ μετά και έως το τέλος του παιχνιδιού επιλέγει ως αγορά «δουκάτα», «χρυσά» ή «επαρχίες». Αν στο χέρι του βρεθεί το παρεκκλήσι που αγόρασε το χρησιμοποιεί για να ξεφορτωθεί όσα «κτήματα» έχει στο χέρι του, αλλά και τόσα χάλκινα ώστε να μην επηρεαστεί η αγορά που θα κάνει στη συνέχεια. Η στρατηγική των Money και Chapel είναι όμοια με αυτή των AI παικτών που χρησιμοποίησε ο Winder [11], με μικρές παραλλαγές, ώστε να αυξηθεί η ανταγωνιστικότητά τους.

Πίνακας 3.1.1 Κάρτες που χρησιμοποιήθηκαν κατά την 1η φάση υλοποίησης

Όνομα κάρτας	Είδος κάρτας	Τιμή	Ιδιότητα
Χάλκινο	Θησαυρός	0	+1 νόμισμα
Ασημένιο	Θησαυρός	3	+2 νομίσματα
Χρυσό	Θησαυρός	6	+3 νομίσματα
Κτήμα	Κάρτα νίκης	2	+1 πόντος νίκης
Δουκάτο	Κάρτα νίκης	5	+2 πόντοι νίκης
Επαρχία	Κάρτα νίκης	8	+3 πόντοι νίκης
Κατάρα	Κάρτα νίκης	0	-1 πόντος νίκης
Παρεκκλήσι	Κάρτα Βασιλείου	2	Πέταξε έως 4 κάρτες από το χέρι στα σκουπίδια
Σιδεράς	Κάρτα Βασιλείου	4	+3 κάρτες

3.2 Εκπαίδευση, έλεγχος και αποτελέσματα

Αν στο παιχνίδι συμμετέχει ο RL player (χωρίς να υπάρχει άνθρωπος), τότε μπορούμε να επιλέξουμε αν θα εκπαιδεύσουμε (train) ή θα ελέγξουμε (test) τον πράκτορα. Το αποτέλεσμα της εκπαίδευσης, δηλαδή οι νέες τιμές της συνάρτησης Q(s,α), για κάθε κατάσταση s και κάθε ενέργεια α, αποθηκεύεται σε ένα αρχείο κειμένου (states.txt). Δημιουργείται επίσης και ένα δεύτερο αρχείο, στο οποίο καταγράφεται ο αριθμός των περασμάτων από κάθε κατάσταση (statesPasses.txt), δηλαδή πόσες φορές κατά τη διάρκεια της εκπαίδευσης ο πράκτορας βρέθηκε σε κάθε κατάσταση. Μετά από 100.000 παιχνίδια εκπαίδευσης, τα αρχεία αυτά αποθηκεύονται με διαφορετικό όνομα που δηλώνει τον αριθμό των παιχνιδιών εκπαίδευσης που έχουν προηγηθεί (π.χ. states300000 και statesPasses300000). Τελειώνοντας την εκπαίδευση το JDominion δημιουργεί και ένα αρχείο (games.txt), όπου καταγράφει τον συνολικό αριθμό παιχνιδιών εκπαίδευσης.

Ξεκινώντας μια νέα περίοδο εκπαίδευσης διαβάζει αυτό το αρχείο και προσθέτει σε αυτό το νέο αριθμό παιχνιδιών που επιλέχθηκαν.

Κάθε φορά που ολοκληρώνονται 10.000 παιχνίδια, η εκπαίδευση διακόπτεται και διενεργούνται 10.000 παιχνίδια ελέγχου, εναντίον των ίδιων αντιπάλων. Στα παιχνίδια αυτά ο πράκτορας χρησιμοποιεί greedy τακτική, βάσει των τιμών της συνάρτησης Q(s,α), όπως έχουν διαμορφωθεί από την εκπαίδευση έως εκείνη τη στιγμή. Σε αυτά τα παιχνίδια καταγράφονται διάφορα στατιστικά στοιχεία, όπως το ποσοστό νικών κάθε παίκτη, ο μέσος όρων των γύρων, ο μέσος όρος των πόντων που συγκεντρώθηκαν, ο μέσος όρων των καρτών από κάθε είδος που είχε ο κάθε παίκτης στην τράπουλά του στο τέλος κάθε παιχνιδιού και άλλα. Αντίστοιχα στοιχεία καταγράφονται και κατά τη διάρκεια των παιχνιδιών εκπαίδευσης. Τα στοιχεία αυτά αποθηκεύονται σε ένα αρχείο excel (stats.xlsx), το οποίο εμπλουτίζεται μετά το τέλος κάθε περιόδου εκπαίδευσης. Το JDominion επίσης καταγράφει ποιο ήταν το μεγαλύτερο ποσοστό νικών που πέτυχε ο πράκτορας κατά τα παιχνίδια ελέγχου και αποθηκεύει αυτό το ποσοστό (bestStatesPercentage.txt), τον αριθμό των παιχνιδιών εκπαίδευσης στο οποίο επιτεύχθηκε (bestStatesNumberOfGames.txt) και τις τιμές της Q(s,α) τότε (bestStates.txt).

Αν αντί για παιχνίδια εκπαίδευσης, επιλέξουμε έλεγχο, το JDominion μας δίνει τρεις επιλογές, manual, auto, ή all. Αν επιλέξουμε manual πρέπει να εισαγάγουμε τον αριθμό των παιχνιδιών εκπαίδευσης του πράκτορα π.χ. 300.000. Τότε ο πράκτορας θα χρησιμοποιεί greedy τακτική, βάσει των τιμών της Q(s,α), όπως αυτές είχαν διαμορφωθεί μετά από 300.000 παιχνίδια εκπαίδευσης. Αν επιλέξουμε auto, τότε θα χρησιμοποιήσει τις τελευταίες αποθηκευμένες τιμές της Q(s,α). Τέλος η επιλογή all κάνει έλεγχο σε όλες τις αποθηκευμένες τιμές της Q(s,α).

3.3 Δυναμικότητα των AI

Προσομοιώθηκαν στη συνέχεια 100.000 παιχνίδια δύο παικτών μεταξύ των AI στρατηγικών, με όλους τους πιθανούς συνδυασμούς, ώστε να προσδιοριστεί ο πιο δύσκολος αντίπαλος. Τα αποτελέσματα φαίνονται στον πίνακα 3.3.1, όπου βλέπουμε το ποσοστό νικών κάθε παίκτη, τον μέσο αριθμό πόντων που σκόραρε, καθώς και τον μέσο αριθμό γύρων που διήρκησαν τα παιχνίδια. Όπως φαίνεται η στρατηγική Random, δεν παρουσιάζει κάποιο ενδιαφέρον, όπως αναμενόταν, καθώς δεν καταφέρνει να κερδίσει ούτε ένα παιχνίδι, απέναντι σε οποιονδήποτε αντίπαλο. Καλύτερη αναδεικνύεται η στρατηγική Chapel, η οποία κερδίζει περισσότερα από τα μισά παιχνίδια, με όλους τους αντιπάλους, ενώ δεύτερη καλύτερη είναι η Money, η οποία τους κερδίζει όλους εκτός της Chapel. Θα πρέπει να σημειώσουμε εδώ, ότι υπάρχουν μικρές διαφορές στα αποτελέσματα με τον Winders [11], ο οποίος διεξήγαγε αντίστοιχα πειράματα στην εργασία του, οι οποίες πιθανό να οφείλονται σε μικρές διαφορές στον προγραμματισμό της εφαρμογής.

Στη συνέχεια διεξάχθηκαν αντίστοιχα πειράματα, μεταξύ τεσσάρων παικτών, οι οποίοι ανά δύο είχαν την ίδια στρατηγική. Ο στόχος ήταν να διαπιστωθεί αν οι στρατηγικές διατηρήσαν την ίδια δυναμική και σε παιχνίδια με τέσσερις παίκτες. Τα αποτελέσματα φαίνονται στον πίνακα 3.3.2, όπου στα ποσοστά νίκης έχει γίνει άθροιση των ποσοστών των παικτών με την ίδια στρατηγική. Εδώ φαίνεται ότι η κατάταξη των

Πίνακας 3.3.1 Αποτελέσματα προσομοίωσης 100.000 παιχνιδιών μεταξύ AI παικτών (ένας εναντίον ενός)

Παίκτης 1	Παίκτης 2	Νίκες παίκτη 1 (%)	Νίκες παίκτη 2 (%)	Μέσος αριθμός πόντων παίκτη 1	Μέσος αριθμός πόντων παίκτη 2	Μέσος αριθμός γύρων
Random	Money	*0*	*100*	*14.17*	*85.74*	*37.24*
Random	Chapel	*0*	*100*	*12.98*	*78.72*	*33.43*
Random	Greedy	*0*	*100*	*16.05*	*91.03*	*44.69*
Random	Blacksmith	*0*	*100*	*14.37*	*85.05*	*37.00*
Money	Chapel	*42.19*	*57.81*	*40.04*	*42.03*	*20.53*
Money	Greedy	*62.58*	*37.41*	*52.90*	*45.52*	*25.21*
Money	Blacksmith	*51.97*	*48.02*	*46.16*	*43.16*	*22.72*
Chapel	Greedy	*66.68*	*33.31*	*49.99*	*41.56*	*23.29*
Chapel	Blacksmith	*55.54*	*44.45*	*43.94*	*39.18*	*21.19*
Greedy	Blacksmith	*38.87*	*61.12*	*46.23*	*51.68*	*24.72*

στρατηγικών βάσει δυναμικότητας είναι με φθίνουσα σειρά Blacksmith, Greedy, Chapel και Money, με τελευταία βέβαια την Random. Η αύξηση του αριθμού των παικτών οδήγησε, όπως φαίνεται με σύγκριση ανάμεσα στους πίνακες 3.3.1 και 3.3.2, σε μείωση του μέσου αριθμού γύρων (καθώς οι κάρτες τελειώνουν πιο γρήγορα). Αυτό είχε ως αποτέλεσμα να αλλάξουν οι ισορροπίες ανάμεσα στις στρατηγικές, καθώς οι Money και Chapel προσπαθούν να χτίσουν την τράπουλα τους, και χρειάζονται μεγαλύτερο αριθμό γύρων, ενώ οι Blacksmith και Greedy προσπαθούν να τελειώσουν το παιχνίδι πιο γρήγορα, αγοράζοντας «Επαρχίες».

Πίνακας 3.3.2 Αποτελέσματα προσομοίωσης 100.000 παιχνιδιών μεταξύ AI παικτών (δύο εναντίον δύο)

Παίκτες 1&2	Παίκτες 3&4	Νίκες παικτών 1&2 (%)	Νίκες παικτών 3&4 (%)	Μέσος αριθμός πόντων παικτών 1&2	Μέσος αριθμός πόντων παικτών 3&4	Μέσος αριθμός γύρων
Random	Money	*0*	*100*	9.48	44.03	21.97
Random	Chapel	*0*	*100*	8.61	37.72	19.12
Random	Greedy	*0*	*100*	10.15	47.32	25.29
Random	Blacksmith	*0*	*100*	9.40	43.83	20.93
Money	Chapel	*46.12*	*53.87*	20.77	21.14	13.99
Money	Greedy	*40.81*	*59.18*	27.22	27.21	16.20
Money	Blacksmith	*31.28*	*68.71*	22.97	24.40	14.72
Chapel	Greedy	*38.25*	*61.74*	24.05	24.71	14.86
Chapel	Blacksmith	*30.74*	*69.25*	20.46	22.05	13.72
Greedy	Blacksmith	*39.87*	*60.21*	25.91	28.18	14.92

Επιλέξαμε λοιπόν να εκπαιδεύσουμε τον πράκτορα σε παιχνίδια τεσσάρων παικτών γιατί μικρότερος αριθμός γύρων συνεπάγεται και σημαντική μείωση του χώρου των πιθανών καταστάσεων.

ΚΕΦΑΛΑΙΟ 4:
Εξερεύνηση του χώρου των καταστάσεων

Η μέθοδος που επιλέξαμε να δοκιμάσουμε είναι αυτή του Q Learning. Επιλέχθηκε γιατί σε σύγκριση με τη S.A.R.S.A. και τα Monte Carlo Trees για την πραγματοποίηση της χρειάζεται η αποθήκευση μικρότερου αριθμού δεδομένων. Ο ρυθμός μάθησης α επιλέχθηκε να είναι 0,2 , ενώ η έκπτωση γ στις μελλοντικές ανταμοιβές 0,95. Πρέπει να σημειώσουμε εδώ ότι έχουμε κάνει μια μικρή αλλαγή, στον χρόνο ανανέωσης των τιμών της Q(s,α). Αντί ο αλγόριθμος να κοιτά στο μέλλον, και να προσδιορίζει από τις πιθανές επόμενες καταστάσεις, ποια έχει τη μέγιστη τιμή, ώστε να κάνει την ανανέωση βάσει αυτής, περιμένει ένα γύρο και κάνει την ανανέωση βάσει των πραγματικών πιθανών αγορών που μπορεί να κάνει. Η αλλαγή αυτή έγινε γιατί οι πιθανές αγορές που μπορεί να κάνει ο πράκτορας κάθε γύρο, έχουν να κάνουν με τις κάρτες που τράβηξε, και το πόσα νομίσματα κατάφερε να συγκεντρώσει. Αν η ανανέωση γινόταν με τον κλασικό τρόπο, θα υπήρχε ο κίνδυνος ο πράκτορας να κάνει ανανέωση βάσει μιας υψηλής τιμής, η οποία όμως θα παρουσιαζόταν σπάνια στην πραγματική ροή του παιχνιδιού.

4.1 ε-greedy vs Forced exploration

Ονομάσαμε την πρώτη προσπάθεια Q Learning 0 ε-greedy. Οι μεταβλητές που χρησιμοποιήθηκαν για το «χτίσιμο» του χώρου των καταστάσεων φαίνονται στον πίνακα 4.1.1. Η στήλη άνω φράγμα δηλώνει την μεγαλύτερη τιμή που μπορεί να πάρει η μεταβλητή. Αν κατά τη διάρκεια του παιχνιδιού η μεταβλητή πάρει μεγαλύτερη τιμή, τότε ο αλγόριθμος τη θέτει ίση με το άνω φράγμα. Πιο αναλυτικά ο αριθμός των γύρων επιλέχθηκε να είναι από 0 έως 38, με άνω φράγμα το 39 βάσει του μέγιστου αριθμού γύρων που παρατηρήθηκε στα παιχνίδια των AI (44). Η μεταβλητή «γύροι για το τέλος» είναι μια εκτίμηση του πόσες φορές είναι πιθανό να παίξει ακόμα ο πράκτορας, μέχρι να τελειώσει το παιχνίδι, βάσει των καρτών που έχουν αγοραστεί από όλους τους παίκτες. Τέλος οι μεταβλητές smithies, ασημένια και χρυσά στην τράπουλα έχουν εκτιμηθεί βάσει των τιμών που παρατηρηθήκαν στα παιχνίδια των AI, και πιο συγκεκριμένα στα παιχνίδια μεταξύ Greedy και Blacksmith. Το μέγεθος του χώρου των καταστάσεων που προέκυψε βάσει των πιο πάνω επιλογών είναι 288.000. Σαν ενέργεια τέλος επιλέχθηκε η κάρτα που αγοράζει ο πράκτορας, έτσι με 9 πιθανές αγορές έχουμε αύξηση του συνολι-

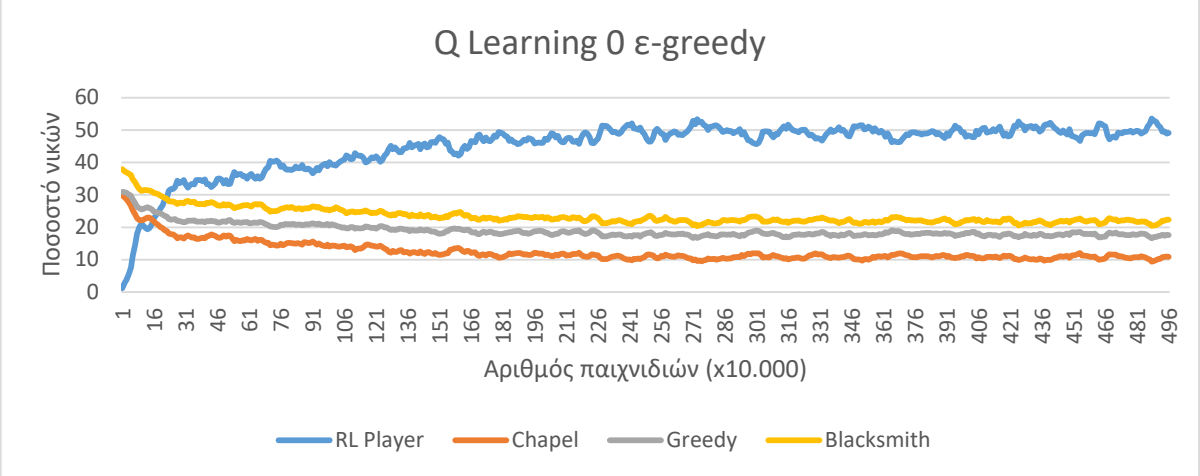
Πίνακας 4.1.1 Χώρος Καταστάσεων για το Q Learning 0

Μεταβλητή	Πιθανές τιμές	Άνω Φράγμα	Αριθμός καταστάσεων
Αριθμός γύρων	*0-39*	*39*	*40*
Γύροι για το τέλος	*0-9*	*9*	*10*
Smithies στην τράπουλα	*0-5*	*5*	*6*
Ασημένια στην τράπουλα	*0-9*	*9*	*10*
Χρυσά στην τράπουλα	*0-11*	*11*	*12*
Πιθανές αγορές			*9*
Συνολικός αριθμός καταστάσεων			*2.592.000*

κού αριθμού των καταστάσεων στις 2.592.000. Πολλές βεβαία από αυτές τις καταστάσεις είναι αδύνατο να παρατηρηθούν στην πορεία του παιχνιδιού, π.χ. δεν είναι δυνατόν στον πρώτο γύρο ο RL να έχει περισσότερα από μηδέν χρυσά στην τράπουλα του. Κάθε κατάσταση λοιπόν χαρακτηρίζεται από έξι αριθμούς, την πιθανή αγορά (α), αριθμό χρυσών στην τράπουλα (g), αριθμό ασημένιων στην τράπουλα (s), αριθμό smithies στην τράπουλα (b), γύροι για το τέλος (e), αριθμός γύρων (t). Ο αύξων αριθμός μιας κατάστασης (Q), κυμαίνεται από 0 έως 2.591.999 και υπολογίζεται από τη σχέση (19).

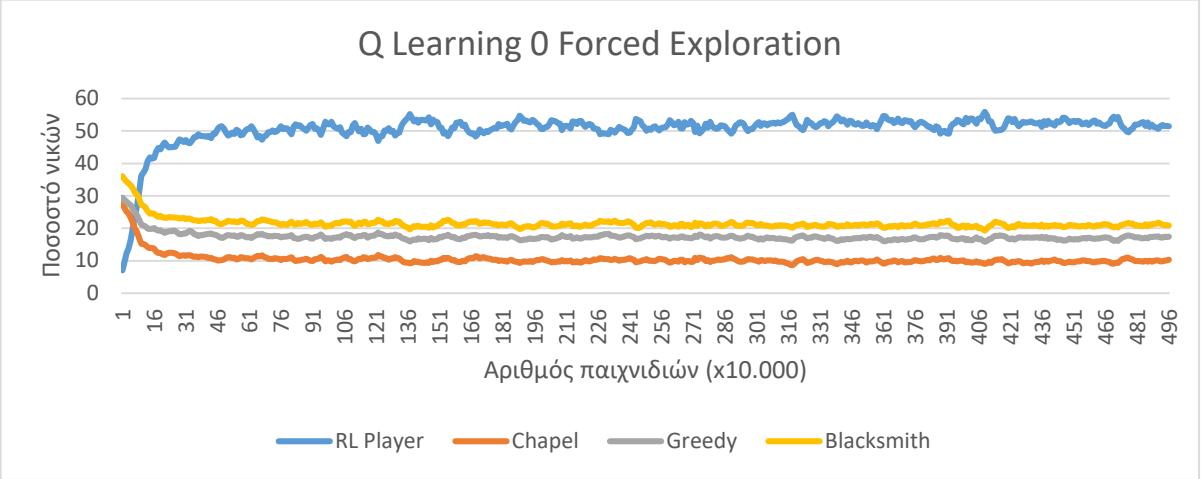
$$Q = a + g \cdot 9 + s \cdot 108 + b \cdot 1080 + e \cdot 6480 + t \cdot 64800 \quad (19)$$

Ο πράκτορας RL εκπαιδεύτηκε με αυτές τις προδιαγραφές για 5.000.000 παιχνίδια. Στην εικόνα 4.1 φαίνεται το ποσοστό των νικών που είχε ο RL στα παιχνίδια ελέγχου (για καλύτερη απεικόνιση στην εικόνα 4.1, αλλά και σε αυτές που ακολουθούν, βλέπουμε τον μέσο όρο 5 παιχνιδιών), σε σχέση με τους AI αντιπάλους, κατά την πορεία της εκπαίδευσης. Το καλύτερο ποσοστό νικών ήταν 56,61%, και επιτεύχθηκε μετά από 4.660.000 παιχνίδια. Υπάρχει όμως μια σταθεροποίηση της μέσης απόδοσης του RL, περίπου στο 50% μετά από περίπου 2.000.000 παιχνίδια.



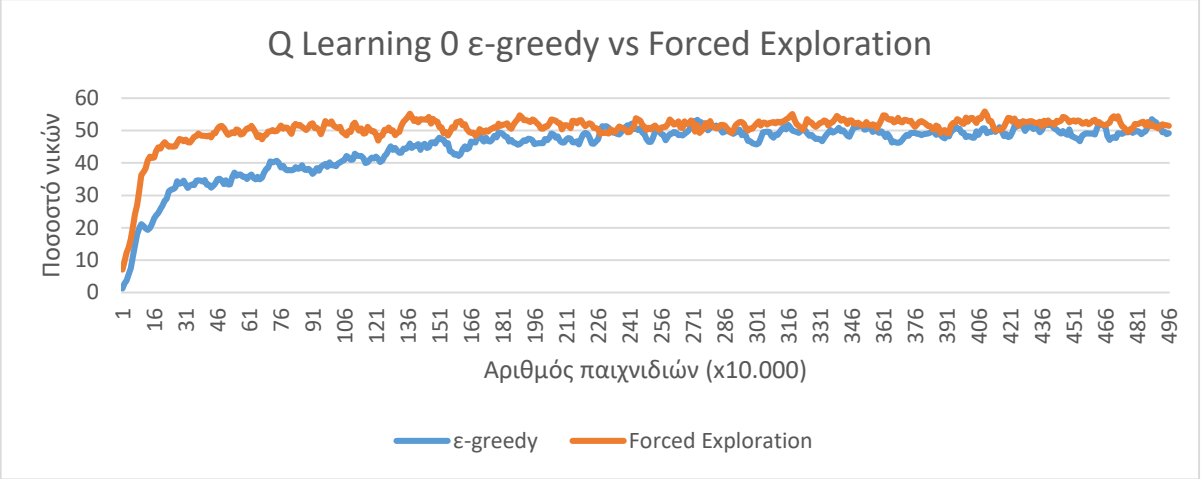
Εικόνα 4.1 Αποτελέσματα παιχνιδιών ελέγχου Q Learning 0 ε-greedy

Ο αριθμός των καταστάσεων που εξερευνήθηκαν κατά την εκπαίδευση ήταν 128.704, ενώ σε λιγότερες από τις μισές από αυτές (55.592) βρέθηκε ο RL πάνω από 10 φορές, στη διάρκεια 5.000.000 παιχνιδιών. Θέλοντας λοιπόν να εξερευνήσουμε περισσότερες καταστάσεις αλλάξαμε τον τρόπο με τον οποίο ο πράκτορας επιλέγει την ενέργεια που θα ακολουθήσει. Στην εκδοχή Q Learning 0 ε-greedy ο πράκτορας κατά την εκπαίδευση χρησιμοποιούσε την μέθοδο επιλογής ενεργειών ε-greedy με ε=0,2. Επέλεγε δηλαδή την greedy ενέργεια με πιθανότητα 0,8 και είχε πιθανότητα 0,2 να επιλέξει μία τυχαία ενέργεια. Στη νέα μέθοδο, ο πράκτορας επιλέγει βάσει των «περασμάτων» που έχει κάθε πιθανή κατάσταση. Πιο συγκεκριμένα αν κάποια από τις πιθανές ενέργειες, οδηγεί σε κατάσταση που έχει λιγότερα από 10 «περάσματα» τότε επιλέγει αυτήν. Αν δύο ή περισσότερες ενέργειες, οδηγούν σε καταστάσεις με λιγότερα από 10 περάσματα, επιλέγει μία στην τύχη. Αν τέλος όλες οι ενέργειες οδηγούν σε καταστάσεις με περισσότερα από 10 περάσματα, επιλέγει την ενέργεια με την μέθοδο ε-greedy. Ονομάσαμε αυτή τη νέα μέθοδο επιλογής καταστάσεων εξαναγκασμένη εξερεύνηση (forced exploration). Τα αποτελέσματα μετά



Εικόνα 4.2 Αποτελέσματα παιχνιδιών ελέγχου Q Learning 0 Forced Exploration

από 5.000.000 παιχνίδια εκπαίδευσης, φαίνονται στην εικόνα 4.2. Το ποσοστό των νικών της νέας μεθόδου αυξάνεται πολύ πιο γρήγορα σε σχέση με την μέθοδο ε-greedy, όπως φαίνεται στην εικόνα 4.3, αλλά έχει και ελαφρώς καλύτερα τελικά αποτελέσματα. Καλύτερο ποσοστό νικών 59,83% που το πετυχαίνουμε στα 2.710.000 παιχνίδια, και μέσος όρος νικών περίπου στο 50%, που επιτυγχάνεται αρκετά νωρίτερα, περίπου στα 500.000 παιχνίδια. Ο τελικός αριθμός των καταστάσεων που εξερευνήθηκαν είναι 200.706 με τις 92.907 να έχουν πάνω από δέκα περάσματα.



Εικόνα 4.3 Σύγκριση μεταξύ των μεθόδων ε-greedy και Forced Exploration

Για να έχουμε μία καλύτερη εικόνα του πόσο πιο γρήγορη είναι η νέα μέθοδος έναντι της ε-greedy θα θέλαμε να ξέρουμε πότε η κάθε μέθοδος ολοκληρώνει την εκπαίδευση του πράκτορα, δηλαδή από ποιο σημείο και έπειτα η εκπαίδευση δεν παρουσιάζει σημαντική πρόοδο. Ορίσαμε έτσι σαν «πρώτο καλό αποτέλεσμα», το πρώτο ποσοστό νικών στα παιχνίδια ελέγχου, το οποίο είναι μεγαλύτερο από το μέσο όρο των ποσοστών νικών των επόμενων παιχνιδιών. Για την μέθοδο ε-greedy λοιπόν το πρώτο καλό αποτέλεσμα 48,51% και το πετυχαίνουμε μετά από 1.150.000 παιχνίδια εκπαίδευσης, ενώ για την εξαναγκασμένη εξερεύνηση είναι 52,31%, μετά από 320.000 παιχνίδια. Όπως φαίνεται στον πίνακα 4.1.2 η μέθοδος της εξαναγκασμένης μάθησης υπερέχει σε όλα τα σημεία, έναντι τη ε-greedy, καθώς είναι πιο γρήγορη αλλά έχει και λίγο καλύτερα αποτελέσματα.

Πίνακας 4.1.2 Συγκριτικά αποτελέσματα ε-greedy και Forced Exploration για το Q Learning 0

Μέθοδος	Καλύτερο αποτέλεσμα		Πρώτο καλό αποτέλεσμα		Αριθμός εξερευνημένων καταστάσεων	Καταστάσεις με περισσότερα από 10 περάσματα
	Ποσοστό (%)	Αριθμός παιχνιδιών	Ποσοστό (%)	Αριθμός παιχνιδιών		
e-greedy	*56,61*	*4.660.000*	*48,51*	*1.150.000*	*128.704*	*55.592*
Forced exploration	*59,83*	*2.710.000*	*52,31*	*320.00*	*200.706*	*92.907*

4.2 Μείωση μεταβλητών

Ο αριθμός 2.592.000 για τον χώρο των καταστάσεων είναι υπερβολικά μεγάλος, ιδίως αν σκεφτούμε ότι στα επόμενα βήματα θα προσθέσουμε και άλλες κάρτες βασιλείου, οπότε θα αυξηθεί ακόμα περισσότερο. Για να μειώσουμε τις καταστάσεις δοκιμάσαμε δύο τρόπους, να αφαιρέσουμε εντελώς κάποιες μεταβλητές ώστε να ελεγχθεί η ποιότητα τους, και να μειώσουμε το άνω φράγμα σε αυτές που απέμειναν, έχοντας έτσι λιγότερες πιθανές τιμές και άρα λιγότερες καταστάσεις. Αφαιρώντας μία μεταβλητή κάθε φορά και δοκιμάζοντας τα αποτελέσματα ήταν απογοητευτικά. Εκτός από μία. Τους γύρους για το τέλος. Αφαιρώντας αυτή τη μεταβλητή τα αποτελέσματα δεν άλλαξαν αισθητά. Ονομάσαμε τη νέα αυτή εκδοχή Q Learning 1. Προσθέσαμε επίσης μία νέα μεταβλητή, το πόσα χάλκινα (coppers) έχει ο πράκτορας στην τράπουλα του. Ο συνολικός αριθμός καταστάσεων μειώθηκε στις 34.020 όπως φαίνεται στον πίνακα 4.2.1.

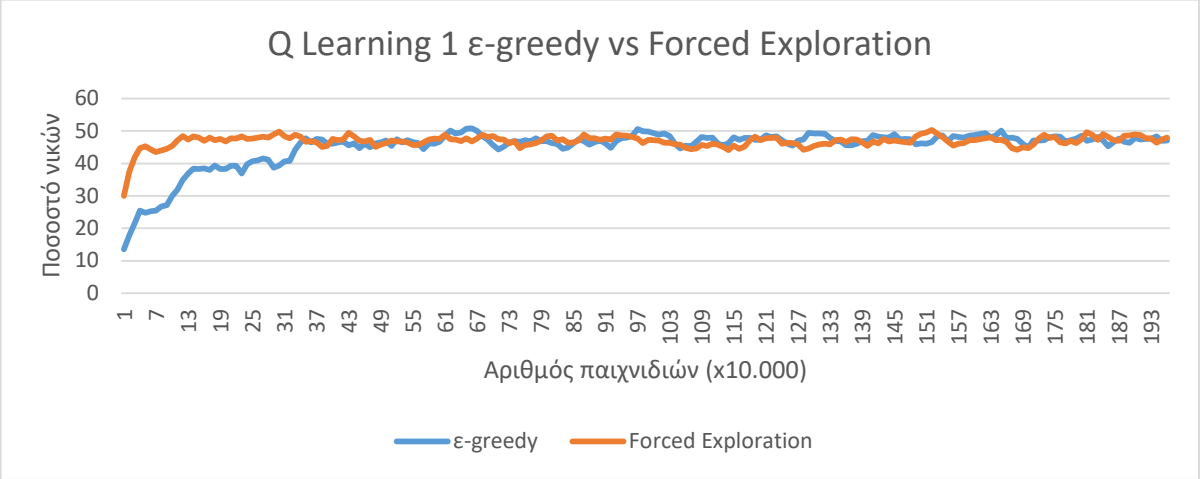
Πίνακας 4.2.1 Χώρος Καταστάσεων για το Q Learning 1

Μεταβλητή	Πιθανές τιμές	Άνω Φράγμα	Αριθμός καταστάσεων
Αριθμός γύρων	*0-34*	*34*	*35*
Smithies στην τράπουλα	*0-2*	*2*	*3*
Χάλκινα στην τράπουλα	*7-9*	*9*	*3*
Ασημένια στην τράπουλα	*0-3*	*3*	*4*
Χρυσά στην τράπουλα	*0-2*	*2*	*3*
Πιθανές αγορές			*9*
Συνολικός αριθμός καταστάσεων			*34.020*

Πάλι δοκιμάσαμε τις δύο διαφορετικές μεθόδους (ε-greedy και Forced Exploration), τα αποτελέσματα των οποίων φαίνονται στην εικόνα 4.4. Καθώς ο αριθμός των καταστάσεων μειώθηκε, ο πράκτορας πλέον επιτυγχάνει τα επιθυμητά αποτελέσματα νωρίτερα, έτσι μειώσαμε τον αριθμό των παιχνιδιών εκπαίδευσης στα 2.000.000. Και εδώ τα συμπεράσματα είναι όμοια με αυτά του Q Learning 0, όπως φαίνεται στον πίνακα 4.2.1.

Πίνακας 4.2.1 Συγκριτικά αποτελέσματα ε-greedy και Forced Exploration για το Q Learning 1

Μέθοδος	Καλύτερο αποτέλεσμα		Πρώτο καλό αποτέλεσμα		Αριθμός εξερευνημένων καταστάσεων	Καταστάσεις με περισσότερα από 10 περάσματα
	Ποσοστό (%)	Αριθμός παιχνιδιών	Ποσοστό (%)	Αριθμός παιχνιδιών		
e-greedy	*54,33*	*560.000*	*49,16*	*280.000*	*13.254*	*9.168*
Forced exploration	*53,18*	*950.000*	*49,26*	*70.00*	*16.184*	*13.608*



Εικόνα 4.4 Σύγκριση μεταξύ των ε-greedy και Forced Exploration για το Q Learning 1

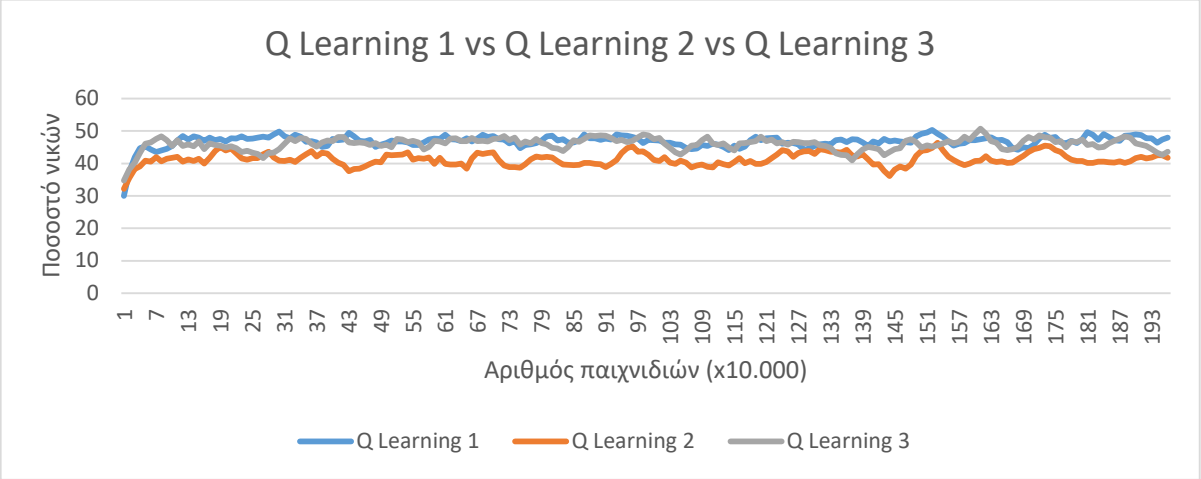
4.3 Μείωση άνω φράγματος

Η προσπάθεια για περαιτέρω μείωση του αριθμού καταστάσεων συνεχίστηκε με διερεύνηση των τιμών του άνω φράγματος για τις εναπομείναντες μεταβλητές. Στην εκδοχή Q Learning 1 επιλέχθηκαν αυθαίρετες τιμές για το άνω φράγμα κάθε μεταβλητής. Η αυθαίρετη επιλογή όμως των άνω φραγμάτων, φαίνεται πως έχει επηρεάσει τα τελικά μας αποτελέσματα. Έτσι χρησιμοποιώντας ως βάση τις αυθαίρετες τιμές του Q Learning 1, κάναμε διερεύνηση στις τιμές του άνω φράγματος κάθε μεταβλητής ξεχωριστά, ώστε να εκτιμήσουμε τις βέλτιστες τιμές. Ξεκινήσαμε με την νέα μεταβλητή, τον αριθμό των χάλκινων στην τράπουλα, ώστε να επιβεβαιώσουμε και την ορθότητα της εισαγωγής της. Στην εκδοχή Q Learning 2 δεν χρησιμοποιείται η μεταβλητή χάλκινα στην τράπουλα, ενώ στην Q Learning 3 έχουμε μειώσει το άνω φράγμα από 9 σε 8. Οι διαφορές έχουν επισημανθεί στον πίνακα 4.3.1. Οι εικόνες που παρατίθενται παρακάτω αφορούν μόνο την εκπαίδευση του πράκτορα με την μέθοδο της εξαναγκασμένης εξερεύνησης. Αντίστοιχα συμπεράσματα όμως, μπορούν να εξαχθούν και αν η εκπαίδευση γινόταν με την ε-greedy μέθοδο. Οι αντίστοιχες εικόνες μπορούν να βρεθούν στο παράρτημα 1.

Πίνακας 4.3.1 Σύγκριση Χώρου Καταστάσεων για τα Q Learning 1, Q Learning 2 και Q Learning 3

Εκδοχή	Q Learning 1		Q Learning 2		Q Learning 3	
Μεταβλητή	Άνω Φράγμα	Αριθμός καταστάσεων	Άνω Φράγμα	Αριθμός καταστάσεων	Άνω Φράγμα	Αριθμός καταστάσεων
Αριθμός γύρων	*34*	*35*	*34*	*35*	*34*	*35*
Smithies στην τράπουλα	*2*	*3*	*2*	*3*	*2*	*3*
Χάλκινα στην τράπουλα	*9*	*3*	*-*	*-*	*8*	*2*
Ασημένια στην τράπουλα	*3*	*4*	*3*	*4*	*3*	*4*
Χρυσά στην τράπουλα	*2*	*3*	*2*	*3*	*2*	*3*
Πιθανές αγορές		*9*		*9*		*9*
Συνολικός αριθμός καταστάσεων		*34.020*		*11.340*		*22.680*

Μετά από 2.000.000 παιχνίδια εκπαίδευσης για την κάθε εκδοχή, τα αποτελέσματα των οποίων φαίνονται στην εικόνα 4.5. Το συμπέρασμα είναι ότι η βέλτιστη τιμή για το άνω φράγμα στα χάλκινα είναι 8 και η καλύτερη εκδοχή η Q Learning 3, καθώς έχουμε μείωση του αριθμού των καταστάσεων, αλλά εξίσου καλά αποτελέσματα με την Q Learning 1.



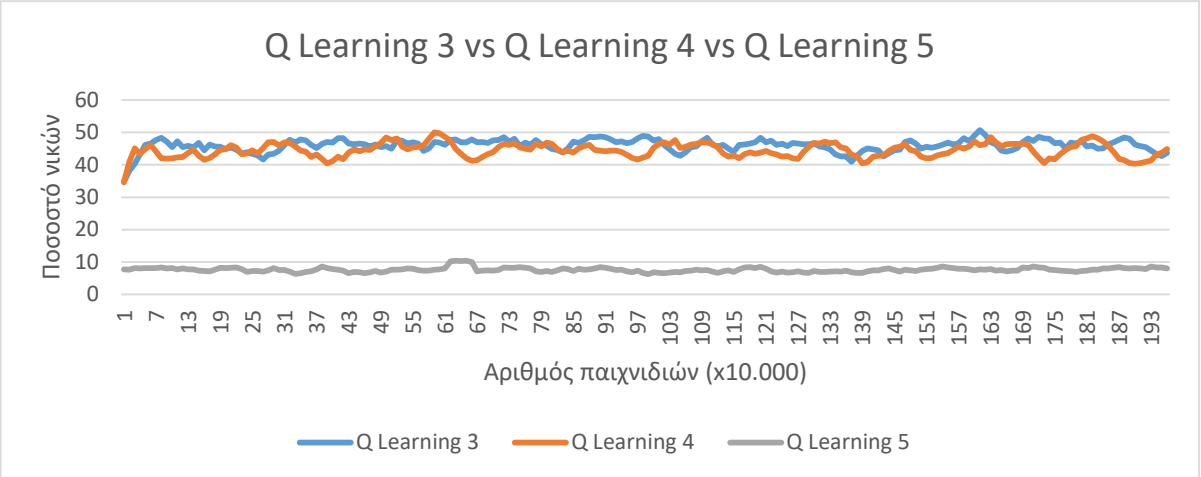
Εικόνα 4.5 Σύγκριση μεταξύ των Q Learning 3, Q Learning 4.1 και Q Learning 4.2

Στη συνέχεια ελέγχουμε την μεταβλητή Smithies στην τράπουλα, χρησιμοποιώντας ως βάση την εκδοχή Q Learning 3. Στην εκδοχή Q Learning 4 μειώνουμε το άνω φράγμα από 2 σε 1, και στην Q Learning 4.4, καταργούμε την μεταβλητή. Οι διαφορές φαίνονται στον πίνακα 4.3.2.

Πίνακας 4.3.2 Σύγκριση Χώρου Καταστάσεων για τα Q Learning 3, Q Learning 4 και Q Learning 5

Εκδοχή	Q Learning 3		Q Learning 4		Q Learning 5	
Μεταβλητή	Άνω Φράγμα	Αριθμός καταστάσεων	Άνω Φράγμα	Αριθμός καταστάσεων	Άνω Φράγμα	Αριθμός καταστάσεων
Αριθμός γύρων	*34*	*35*	*34*	*35*	*34*	*35*
Smithies στην τράπουλα	*2*	*3*	*1*	*2*	-	-
Χάλκινα στην τράπουλα	*8*	*2*	*8*	*2*	*8*	*2*
Ασημένια στην τράπουλα	*3*	*4*	*3*	*4*	*3*	*4*
Χρυσά στην τράπουλα	*2*	*3*	*2*	*3*	*2*	*3*
Πιθανές αγορές		*9*		*9*		*9*
Συνολικός αριθμός καταστάσεων		*22.680*		*15.120*		*7.560*

Όπως φαίνεται από τα αποτελέσματα της εκπαίδευσης στην εικόνα 4.6, η μείωση του άνω φράγματος σε 1 δεν είχε σημαντικές επιπτώσεις στα αποτελέσματα, άρα αποφασίσαμε να κρατήσουμε ως βάση την εκδοχή Q Learning 4.



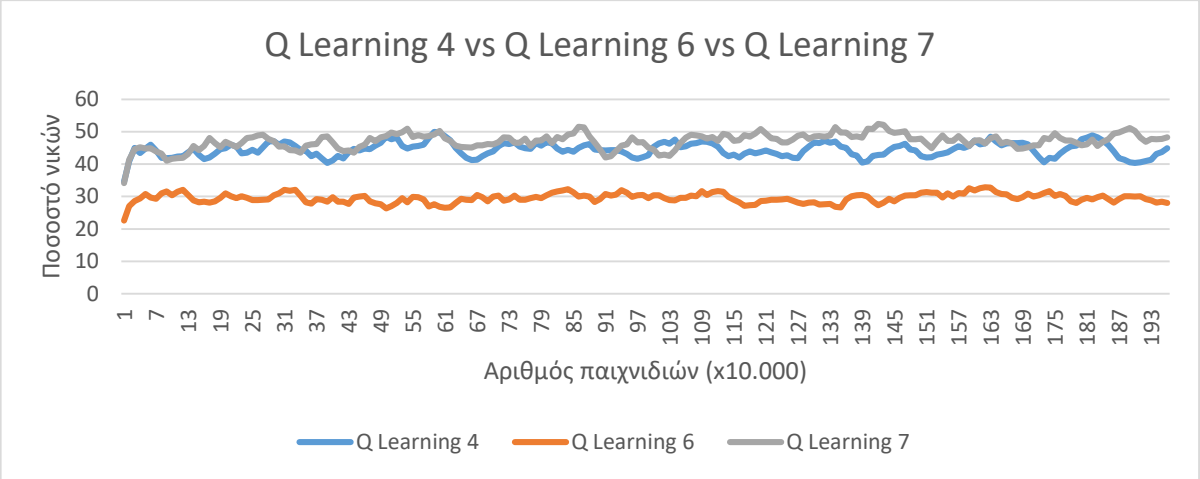
Εικόνα 4.6 Σύγκριση μεταξύ των Q Learning 3, Q Learning 4 και Q Learning 5

Για τον έλεγχο της μεταβλητής χρυσά στην τράπουλα μειώσαμε το άνω φράγμα από 2 σε 1 στην εκδοχή Q Learning 6, και το αυξήσαμε σε 3 στην Q Learning 7, όπως φαίνεται στον πίνακα 4.3.3.

Πίνακας 4.3.3 Σύγκριση Χώρου Καταστάσεων για τα Q Learning 4, Q Learning 6 και Q Learning 7

Εκδοχή	Q Learning 4		Q Learning 6		Q Learning 7	
Μεταβλητή	Άνω Φράγμα	Αριθμός καταστάσεων	Άνω Φράγμα	Αριθμός καταστάσεων	Άνω Φράγμα	Αριθμός καταστάσεων
Αριθμός γύρων	34	35	34	35	34	35
Smithies στην τράπουλα	1	2	1	2	1	2
Χάλκινα στην τράπουλα	8	2	8	2	8	2
Ασημένια στην τράπουλα	3	4	3	4	3	4
Χρυσά στην τράπουλα	2	3	1	2	3	4
Πιθανές αγορές		9		9		9
Συνολικός αριθμός καταστάσεων		15.120		10.080		20.160

Όπως βλέπουμε στην εικόνα 4.7, η αύξηση του άνω φράγματος σε 3, δεν οδήγησε σε θεαματική βελτίωση των αποτελεσμάτων. Άρα πάλι κρατάμε ως βάση την Q Learning 4.



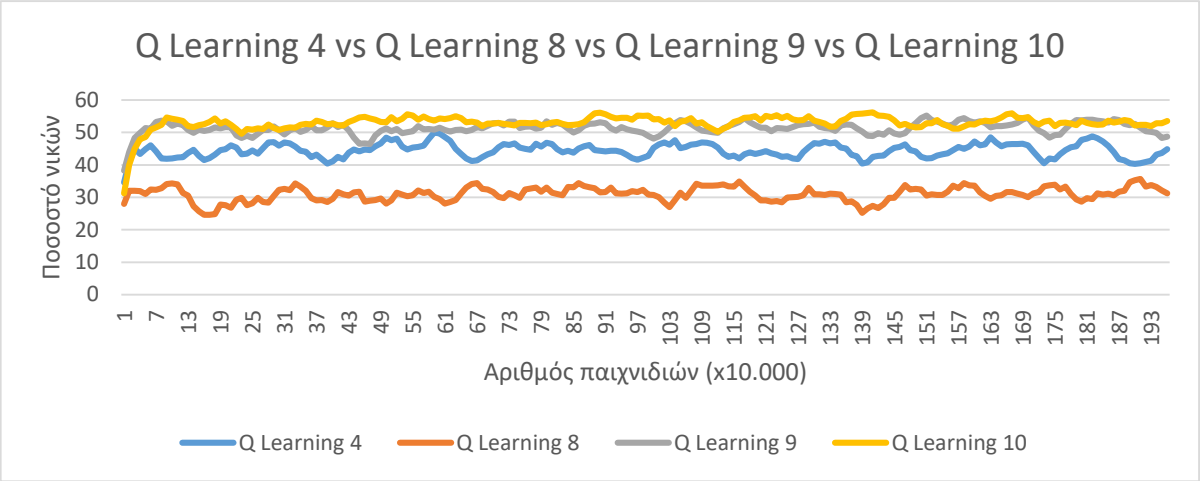
Εικόνα 4.7 Σύγκριση μεταξύ των Q Learning 4, Q Learning 6 και Q Learning 7

Για την μεταβλητή ασημένια στην τράπουλα, δοκιμάζουμε σαν τιμές για το άνω φράγμα 2, 4 και 5 , στις εκδοχές Q Learning 8, Q Learning 9 και Q Learning 10 αντίστοιχα. Οι αντίστοιχοι χώροι καταστάσεων φαίνονται στον πίνακα 4.3.4.

Πίνακας 4.3.4 Σύγκριση Χώρου Καταστάσεων για τα Q Learning 4, Q Learning 8, Q Learning 9 και Q Learning 10

Εκδοχή	Q Learning 4		Q Learning 8		Q Learning 9		Q Learning 10	
Μεταβλητή	Άνω Φράγμα	Αριθμός καταστάσεων	Άνω Φράγμα	Αριθμός καταστάσεων	Άνω Φράγμα	Αριθμός καταστάσεων	Άνω Φράγμα	Αριθμός καταστάσεων
Αριθμός γύρων	34	35	34	35	34	35	34	35
Smithies στην τράπουλα	1	2	1	2	1	2	1	2
Χάλκινα στην τράπουλα	8	2	8	2	8	2	8	2
Ασημένια στην τράπουλα	3	4	2	3	4	5	5	6
Χρυσά στην τράπουλα	2	3	2	3	2	3	2	3
Πιθανές αγορές		9		9		9		9
Συνολικός αριθμός καταστάσεων		15.120		11.340		18.900		22.680

Τα αποτελέσματα της εκπαίδευσης (εικόνα 4.8) δείχνουν ότι έχουμε σαφή βελτίωση των αποτελεσμάτων με την αύξηση του άνω φράγματος. Επιλέγουμε λοιπόν ως νέα βάση την Q Learning 9.

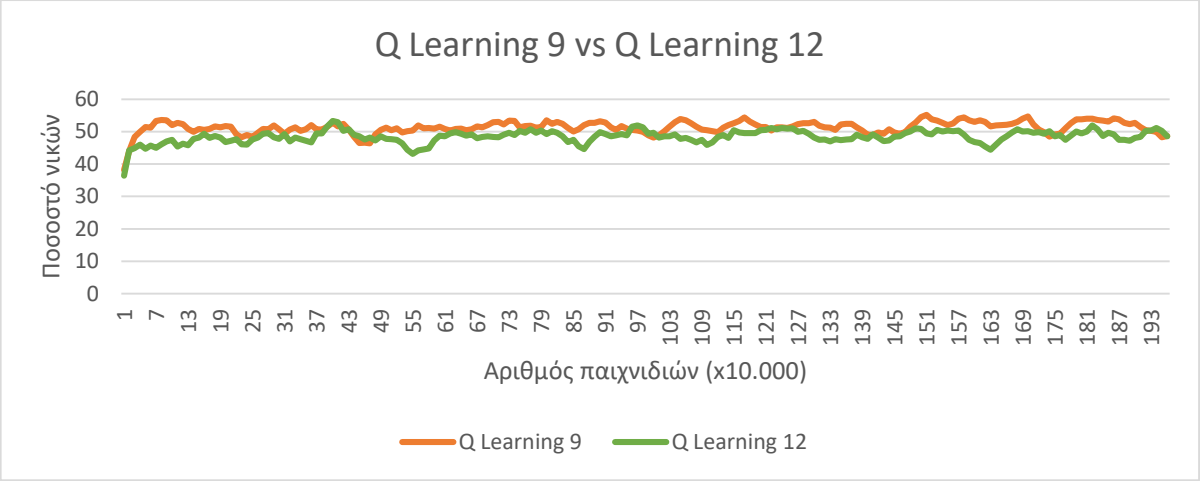


Εικόνα 4.8 Σύγκριση μεταξύ των Q Learning 4, 8, 9 και 10

Τέλος μετά από αρκετές προσπάθειες, προσδιορίσαμε ότι η βέλτιστη τιμή για το άνω φράγμα για τον αριθμό των γύρων είναι 12 (εκδοχή Q Learning 12), που οδηγεί σε συνολικό αριθμό καταστάσεων 7.020. Τα αποτελέσματα της εκπαίδευσης φαίνονται στην εικόνα 4.9. Στον πίνακα 4.3.5 βλέπουμε τον τελικό χώρο των καταστάσεων.

Πίνακας 4.3.5 Χώρος Καταστάσεων για το Q Learning 12

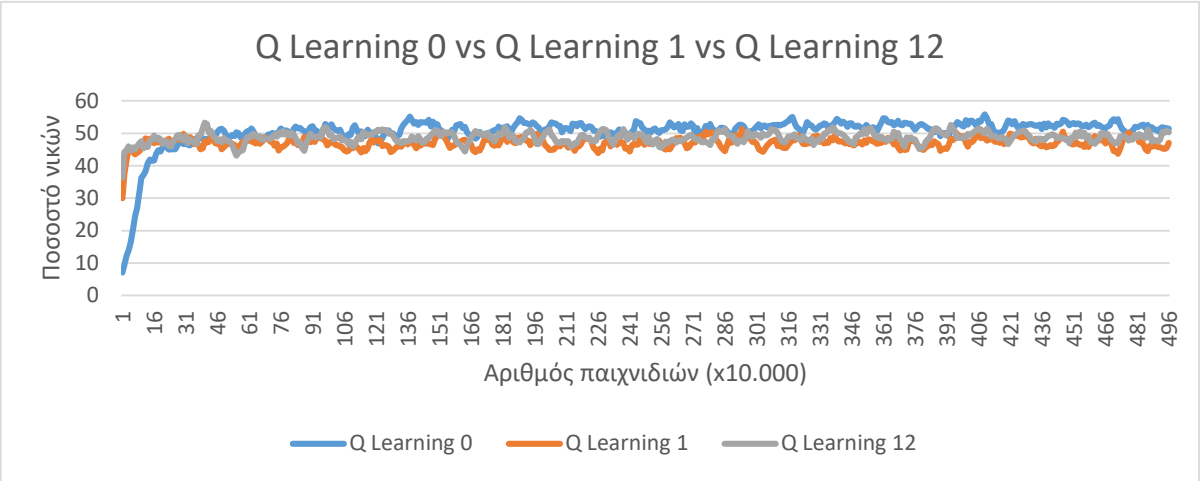
Μεταβλητή	Πιθανές τιμές	Άνω Φράγμα	Αριθμός καταστάσεων
Αριθμός γύρων	*0-12*	*12*	*13*
Χάλκινα στην τράπουλα	*7-8*	*8*	*2*
Smithies στην τράπουλα	*0-1*	*1*	*2*
Ασημένια στην τράπουλα	*0-4*	*4*	*5*
Χρυσά στην τράπουλα	*0-2*	*2*	*3*
Πιθανές αγορές			*9*
Συνολικός αριθμός καταστάσεων			*7.020*



Εικόνα 4.9 Σύγκριση μεταξύ των Q Learning 9, και Q Learning 12

Για μια συνολική εικόνα της πορείας έως τώρα, παραθέτουμε την εικόνα 4.10, στην οποία φαίνονται τα αποτελέσματα των εκδοχών Q Learning 0, 1 και 12. Για να έχουμε

καλύτερη εικόνα έχουμε επεκτείνει την εκπαίδευση στις 2 τελευταίες στα 5.000.000 παιχνίδια.



Εικόνα 4.10 Σύγκριση μεταξύ των Q Learning 0, 1 και 12

Παρόλο που η Q Learning 0 υπερέχει κατά λίγο της 12 στα αποτελέσματα, η μείωση του αριθμού των καταστάσεων από 2.592.000 σε 7.020 είναι πιο σημαντική, ειδικά αν λάβουμε υπόψιν ότι όταν προσθέσουμε στο παιχνίδι και τις υπόλοιπες κάρτες βασιλείου, περιμένουμε ο αριθμός των καταστάσεων να αυξηθεί τουλάχιστον κατά έναν παράγοντα 2^{10}.

4.4 Εκπαίδευση εναντίον διαφορετικών αντιπάλων

Μία τελευταία ερώτηση που προσπαθήσαμε να απαντήσουμε σε αυτή τη φάση των πειραμάτων ήταν το κατά πόσο η επιλογή των αντιπάλων εναντίον των οποίων εκπαιδεύτηκε ο πράκτορας, επηρεάζει την τελική απόδοση του. Χρησιμοποιώντας λοιπόν την εκδοχή Q Learning 4.13 εκπαιδεύσαμε 6 διαφορετικούς πράκτορες και στην συνέχεια ελέγξαμε την απόδοση τους, εναντίον των αντιπάλων στους οποίους εκπαιδεύτηκαν οι άλλοι πράκτορες. Τα αποτελέσματα φαίνονται στον πίνακα 4.4.1. Όπως βλέπουμε με εξαίρεση τον πράκτορα που εκπαιδεύτηκε εναντίον Random αντιπάλων, ο οποίος υστερεί σημαντικά με τους υπόλοιπους, δεν υπάρχουν μεγάλες διαφορές ανάμεσα στους άλλους πράκτορες. Πιθανόν η διαφορά του πρώτου να οφείλεται στο ότι κατά την εκπαίδευση τα παιχνίδια διαρκούσαν περισσότερο, και η στρατηγική που ανέπτυξε να μην έχει καλά αποτελέσματα εναντίον πιο γρήγορων αντιπάλων.

Πίνακας 4.4.1 Ποσοστά νίκης εναντίον διαφορετικών αντιπάλων

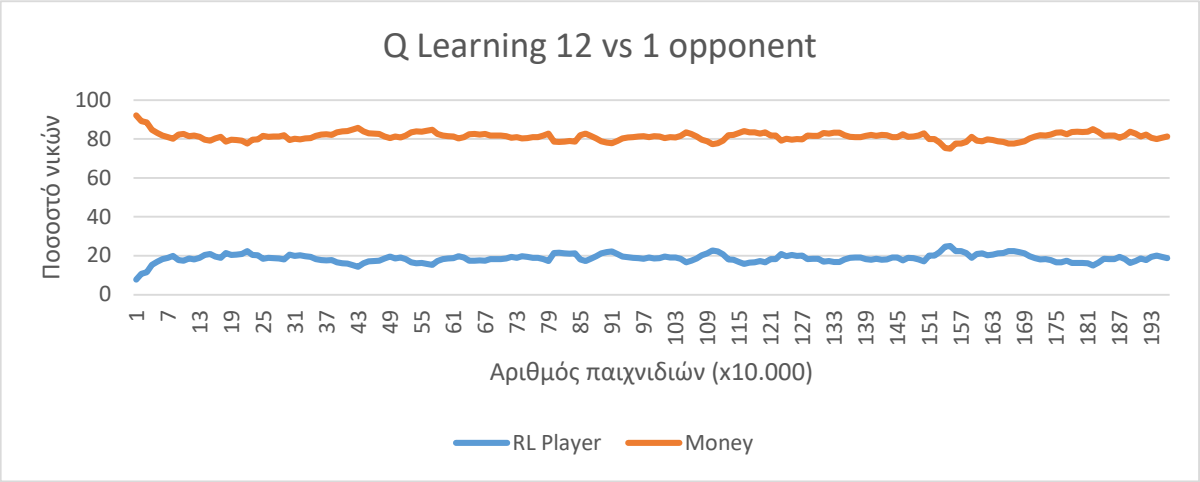
Αντίπαλοι κατά την εκπαίδευση	**Μέσος αριθμός γύρων κατά την εκπαίδευση**	**Αντίπαλοι κατά τον έλεγχο**					
		Random Random Random	Money Money Money	Greedy Greedy Greedy	Blacksmith Blacksmith Blacksmith	Chapel Chapel Chapel	Greedy Blacksmith Chapel
Random, Random, Random	*26.1835*	*100*	*68,36*	*29,99*	*35,10*	*78,24*	*48,81*
Money, Money, Money	*15.1399*	*100*	*77,86*	*38,64*	*43,84*	*85,38*	*57,89*
Greedy, Greedy, Greedy	*17.3816*	*100*	*77,99*	*39,27*	*43,40*	*84,58*	*57,97*
Blacksmith, Blacksmith, Blacksmith	*13.785*	*99,99*	*77,46*	*37,70*	*44,76*	*85,63*	*58,28*
Chapel, Chapel, Chapel	*13.54538*	*100*	*75,18*	*33,31*	*42,66*	*85,10*	*55,17*
Greedy, Blacksmith, Chapel	*15.2097*	*99,99*	*76,41*	*35,42*	*43,61*	*85,75*	*56,51*

ΚΕΦΑΛΑΙΟ 5:
Επέκταση στον πλήρη χώρο καταστάσεων

Το επόμενο βήμα ήταν να ελέγξουμε την απόδοση του πράκτορα εναντίον ενός αντιπάλου, αλλά και να δώσουμε στον πράκτορα την επιλογή αν αγοράσει και άλλες κάρτες βασιλείου, έως ότου φτάσουμε στις δέκα, όπως είναι και οι κανόνες του παιχνιδιού. Στα επόμενα πειράματα χρησιμοποιήσαμε σαν αντίπαλο κατά την εκπαίδευση τον Money, ο οποίος αγοράζει μόνο κάρτες θησαυρών, έτσι ώστε τα στατιστικά που θα εξάγουμε για τις κάρτες βασιλείου που μαθαίνει να αγοράζει ο πράκτορας, να μην επηρεάζονται από τον αριθμό των καρτών βασιλείου που αγοράζουν οι AI αντίπαλοι.

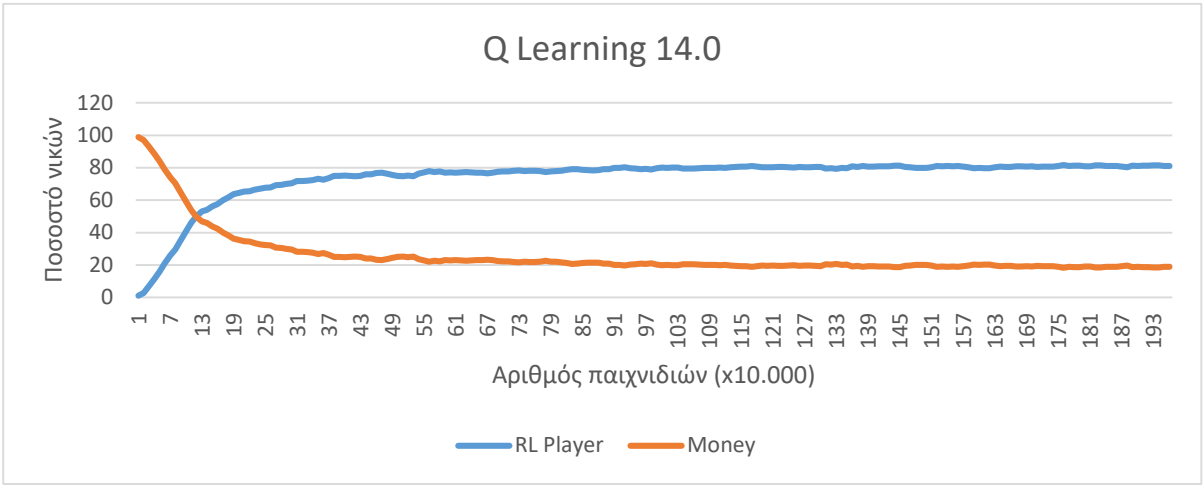
5.1 Βέλτιστος χώρος καταστάσεων για δύο παίκτες.

Δοκιμάσαμε να εκπαιδεύσουμε τον πράκτορα εναντίον ενός αντιπάλου, διατηρώντας τις ίδιες παραμέτρους για τον χώρο των καταστάσεων (Q Learning 12), και τα αποτελέσματα, όπως φαίνεται στην εικόνα 5.1, ήταν απογοητευτικά.



Εικόνα 5.1 Q Learning 12 εναντίον ενός αντιπάλου

Δημιουργήσαμε έτσι την έκδοση Q Learning 14.0, χρησιμοποιώντας τυχαίες τιμές για τα άνω φράγματα των μεταβλητών, όπως φαίνεται στον πίνακα 5.1.1. Τα αποτελέσματα της εκπαίδευσης φαίνονται στην εικόνα 5.2.



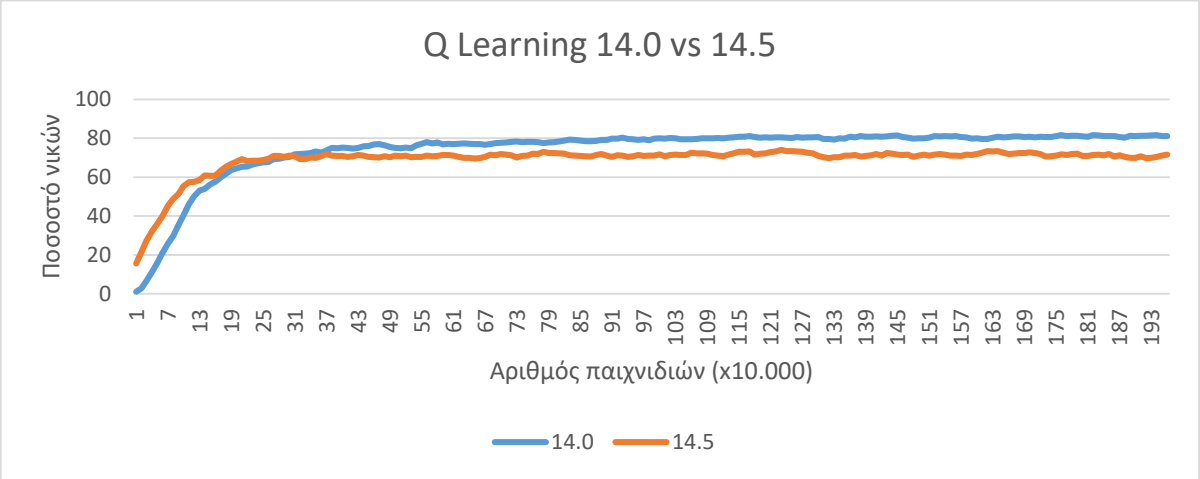
Εικόνα 5.2 Q Learning 14.0

Στη συνέχεια δημιουργήσαμε τις εκδόσεις 14.1 έως και 14.5, αλλάζοντας στην κάθε μία το άνω φράγμα μιας μεταβλητής. Σαν άνω φράγμα χρησιμοποιήσαμε τον μέσο όρο της μεταβλητής που προέκυψε από τα παιχνίδια ελέγχου της προηγούμενης έκδοσης, στρογγυλοποιημένο προς τα πάνω. Έτσι στην έκδοση 14.5 ο συνολικός αριθμός καταστάσεων έπεσε στις 37.800. Στον πίνακα 5.1.1 φαίνεται ο χώρος των καταστάσεων για τις εκδόσεις 14.0 και 14.5.

Πίνακας 5.1.1 Χώρος Καταστάσεων για τις Q Learning 14.0 και 14.5

	Q learning 14.0			Q learning 14.5		
Μεταβλητή	**Πιθανές τιμές**	**Άνω Φράγμα**	**Αριθμός καταστάσεων**	**Πιθανές τιμές**	**Άνω Φράγμα**	**Αριθμός καταστάσεων**
Αριθμός γύρων	*0-49*	*49*	*50*	*0-19*	*19*	*20*
Χάλκινα στην τράπουλα	*7-11*	*11*	*5*	*7-8*	*8*	*2*
Smithies στην τράπουλα	*0-4*	*4*	*5*	*0-2*	*2*	*3*
Ασημένια στην τράπουλα	*0-9*	*9*	*10*	*0-6*	*6*	*7*
Χρυσά στην τράπουλα	*0-9*	*9*	*10*	*0-4*	*4*	*5*
Πιθανές αγορές			*9*			*9*
Συνολικός αριθμός καταστάσεων			*1.125.000*			*37.800*

Αν και η έκδοση 14.5 έχει ελαφρώς χειρότερα αποτελέσματα, όπως βλέπουμε στην εικόνα 5.3, η μείωση του αριθμού των καταστάσεων είναι πιο σημαντική για το επόμενο βήμα, το οποίο είναι η αύξηση των καρτών βασιλείου σε δέκα, όπου περιμένουμε να δούμε τεράστια αύξηση του χώρου των καταστάσεων.



Εικόνα 5.3 Σύγκριση μεταξύ των Q Learning 14.0 vs 14.5

5.2 Νέες κάρτες βασιλείου

Σύμφωνα με τους κανόνες του παιχνιδιού, οι παίκτες έχουν διαθέσιμες 10 είδη καρτών βασιλείου, από τις οποίες μπορούν να επιλέξουν ποια θα αγοράσουν. Προσθέσαμε λοιπόν 9 ακόμα κάρτες βασιλείου έτσι ώστε ο συνολικός αριθμός να φτάσει τις 10. Οι κάρτες αυτές φαίνονται στον πίνακα 5.2.1, μαζί με μια σύντομη περιγραφή τους και το κόστος αγοράς τους. Οι νέες κάρτες προστέθηκαν μία τη φορά, με τη σειρά που τις βλέπουμε στον πίνακα 5.2.1, στις εκδόσεις Q Learning 15 έως και 23. Για κάθε έκδοση έγινε εκπαίδευση του πράκτορα για δύο και για τέσσερις παίκτες, με την μέθοδο ε-greedy, αλλά και με την Forced Exploration. Σε κάθε έκδοση το άνω φράγμα για κάθε κάρτα βασιλείου ορίστηκε να είναι ίσο με τον μέσο όρο των καρτών αυτού του είδους, που αγόρασε ο πράκτορας κατά τα παιχνίδια ελέγχου της προηγούμενης έκδοσης,

στρογγυλοποιημένο προς τα πάνω. Για οικονομία τα αποτελέσματα των ενδιάμεσων εκδόσεων παρατίθενται στο παράρτημα 2.

Πίνακας 5.2.1 Κάρτες Βασιλείου

Κάρτα	Αγγλική ονομασία	Ιδιότητα	Κόστος
Σιδεράς	*Smithy*	*+3 κάρτες*	*4*
Παρεκκλήσι	*Chapel*	*Ξεσκαρτάρισε έως 4 κάρτες*	*2*
Αγορά	*Bazaar*	*+1 κάρτα, +2 ενέργειες, +1 χρυσό*	*5*
Τυχοδιώκτης	*Adventurer*	*Τράβα κάρτες μέχρι να τραβήξεις 2 κάρτες θησαυρών*	*6*
Συνωμότης	*Conspirator*	*+2 χρυσά, αν έπαιξες πάνω από 2 κάρτες αυτό το γύρο +1 κάρτα, +1 ενέργεια*	*4*
Γιορτή	*Festival*	*+2 ενέργειες, +1 αγορά, +2 χρυσά*	*5*
Τοκογλύφος	*Moneylender*	*Ξεσκαρτάρισε ένα χάλκινο, +3 χρυσά*	*4*
Χωριό	*Village*	*+1 κάρτα, +2 ενέργειες*	*3*
Ξυλοκόπος	*Woodcutter*	*+1 αγορά, +2 χρυσά*	*3*
Χωριό εργατών	*Workers village*	*+1 κάρτα, +2 ενέργειες, +1 αγορά*	*4*

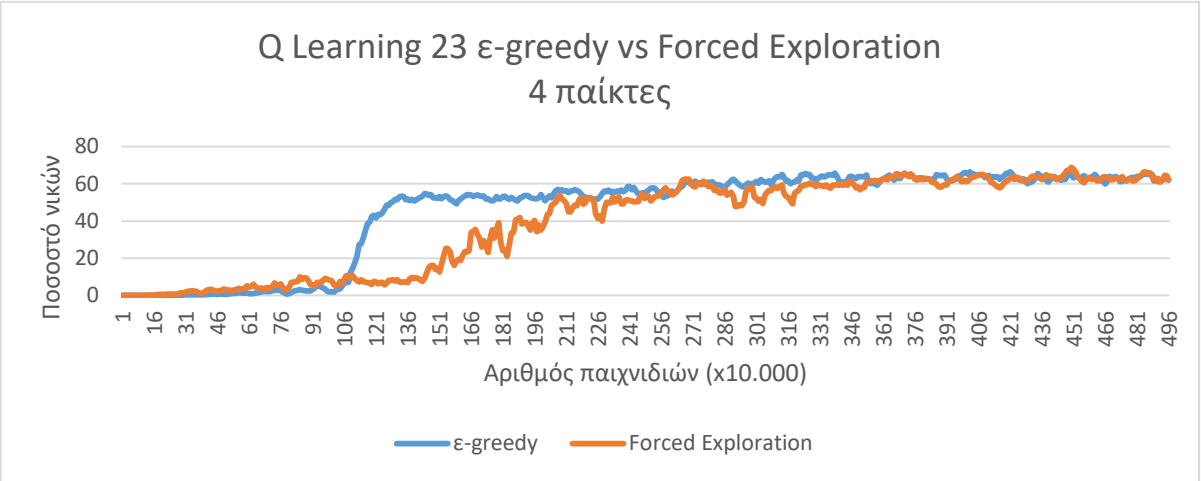
Καταλήξαμε έτσι στην έκδοση Q Learning 23 να έχουμε 7.188.480 καταστάσεις για παιχνίδια 4 παικτών και 38.707.200 καταστάσεις για παιχνίδια 4 παικτών. Αναλυτικά ο χώρος των καταστάσεων φαίνεται στον πίνακα 5.2.2.

Πίνακας 5.2.2 Χώρος Καταστάσεων για την Q Learning 23

	Για 4 παίκτες			Για 2 παίκτες		
Μεταβλητή	Πιθανές τιμές	Άνω Φράγμα	Αριθμός καταστάσεων	Πιθανές τιμές	Άνω Φράγμα	Αριθμός καταστάσεων
Αριθμός γύρων	*0-12*	*12*	*13*	*0-19*	*19*	*20*
Χάλκινα στην τράπουλα	*7-8*	*8*	*2*	*7-8*	*8*	*2*
Σιδεράδες στην τράπουλα	*0-1*	*1*	*2*	*0-2*	*2*	*3*
Ασημένια στην τράπουλα	*0-4*	*4*	*5*	*0-6*	*6*	*7*
Χρυσά στην τράπουλα	*0-2*	*2*	*3*	*0-4*	*4*	*5*
Παρεκκλήσια στην τράπουλα	*0-1*	*1*	*2*	*0-1*	*1*	*2*
Αγορές στην τράπουλα	*0-1*	*1*	*2*	*0-1*	*1*	*2*
Τυχοδιώκτες στην τράπουλα	*0-1*	*1*	*2*	*0-1*	*1*	*2*
Συνωμότες στην τράπουλα	*0-1*	*1*	*2*	*0-1*	*1*	*2*
Γιορτές στην τράπουλα	*0-1*	*1*	*2*	*0-1*	*1*	*2*
Τοκογλύφοι στην τράπουλα	*0-1*	*1*	*2*	*0-1*	*1*	*2*
Χωριά στην τράπουλα	*0-1*	*1*	*2*	*0-1*	*1*	*2*
Ξυλοκόποι στην τράπουλα	*0-1*	*1*	*2*	*0-1*	*1*	*2*
Χωριά εργατών στην τράπουλα	*0-1*	*1*	*2*	*0-1*	*1*	*2*
Πιθανές αγορές			*18*			*18*
Συνολικός αριθμός καταστάσεων			*7.188.480*			*38.707.200*

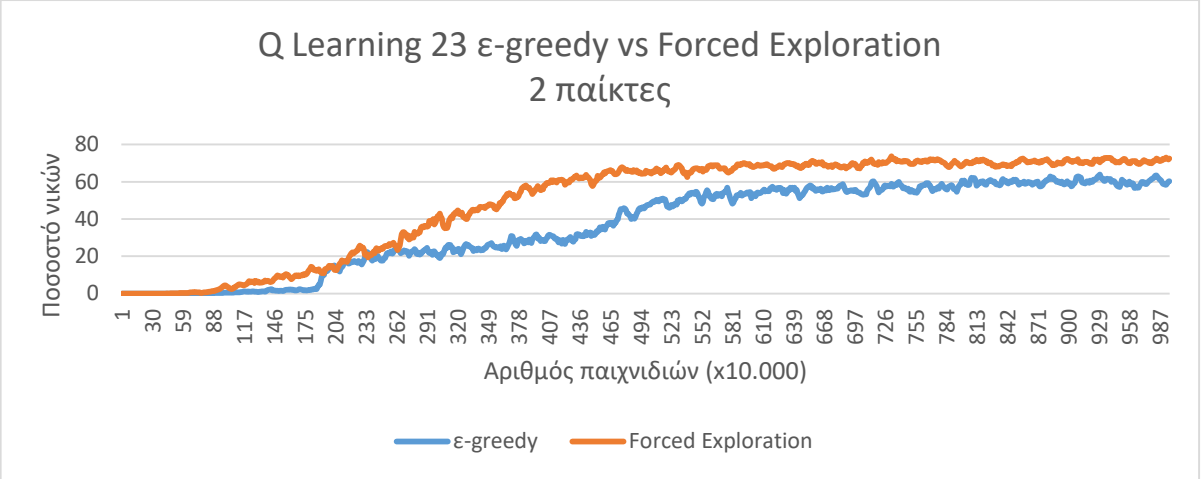
Εκπαιδεύσαμε τον πράκτορα σε παιχνίδια 2 αλλά και 4 παικτών, χρησιμοποιώντας την ε-greedy μέθοδο, και στην συνέχεια την Forced Exploration. Καθώς ο αριθμός των καταστάσεων αυξήθηκε, εκπαιδεύσαμε τον πράκτορα για 5.000.000 παιχνίδια στην περίπτωση των 4 παικτών, και για 10.000.000 παιχνίδια στους 2 παίκτες. Τα αποτελέσματα της εκπαίδευσης με 4 παίκτες φαίνονται στην εικόνα 5.4. Εδώ βλέπουμε μία ανατροπή, καθώς η ε-greedy εμφανίζει καλύτερα αποτελέσματα πιο γρήγορα. Το αποτέλεσμα αυτό δεν είναι τυχαίο καθώς η τάση αυτή είχε αρχίσει να διαφαίνεται ήδη από την έκδοση Q Learning 19, στην οποία προσθέσαμε την κάρτα Γιορτή. Επίσης τα πειράματα επαναλήφθηκαν δίνοντας διαφορετικές τυχαίες αρχικές τιμές Q(s,α) για κάθε κατάσταση, αλλά η εικόνα ήταν παρόμοια. Παρόμοια τάση φαίνεται να υπάρχει και στην εκπαίδευση του πράκτορα σε παιχνίδια για δύο παίκτες, όπως βλέπουμε στην εικόνα 5.5,

καθώς η ε-greedy αυξάνει ξαφνικά τα ποσοστά νικών μετά από περίπου 1.750.000 παιχνίδια εκπαίδευσης, αλλά εδώ η Forced exploration καταφέρνει να επικρατήσει.



Εικόνα 5.4 Q Learning 23 ε-greedy και Forced Exploration για 4 παίκτες

Μία πιθανή εξήγηση για αυτό το φαινόμενο, ίσως να βρίσκεται στη φιλοσοφία των δύο μεθόδων. Η ε-greedy επιλέγει την κατάσταση με την μεγαλύτερη τιμή Q(s,α) με πιθανότητα 80%, ή μία τυχαία κατάσταση με πιθανότητα 20%. Εξερευνά λοιπόν νέες καταστάσεις με τυχαίο τρόπο, αλλά όταν βρει μια κατάσταση που αντιστοιχεί σε καλή στρατηγική, συνεχίζει να την επιλέγει με μεγάλη πιθανότητα, και εξερευνά καταστάσεις που οδηγούν ή προκύπτουν από αυτήν πιο συχνά. Αυτό έχει ως αποτέλεσμα, μετά από αυτό το σημείο, να καταλήγει στη βέλτιστη στρατηγική αρκετά γρήγορα. Εξ ου και η ξαφνική αύξηση στο ποσοστό νικών, μετά από περίπου 1.000.000 παιχνίδια εκπαίδευσης. Αντίθετα η Forced Exploration είναι πιο μεθοδική και δίνει μεγαλύτερο βάρος στην εξερεύνηση. Σε μικρό αριθμό καταστάσεων βέβαια η μεθοδική αυτή εξερεύνηση τελειώνει αρκετά γρήγορα, και επανέρχεται στην ε-greedy δρέποντας τα οφέλη των ήδη εξερευνημένων καταστάσεων. Αυτό έχει ως συνέπεια να είναι πιο γρήγορη σε μικρό αριθμό καταστάσεων, αλλά όταν αυτός αυξάνεται χάνει πολύ χρόνο εξερευνώντας όλες τις δυνατότητες, και έτσι το ποσοστό νικών αυξάνεται πιο ομαλά από την ε-greedy μεν, πιο αργά δε. Όταν όμως ο αριθμός των καταστάσεων αυξάνεται υπερβολικά, η ε-greedy δεν καταφέρνει να κάνει παρά μόνο μικρά άλματα προόδου, οπότε η μεθοδικότητα της Forced Exploration επικρατεί.



Εικόνα 5.5 Q Learning 23 ε-greedy και Forced Exploration για 2 παίκτες

5.3 Ανάλυση των αποτελεσμάτων της εκπαίδευσης

Στον πίνακα 5.3.1 φαίνονται τα στατιστικά της εκπαίδευσης του πράκτορα στην τελική έκδοση (Q Learning 23) για δύο και για τέσσερις παίκτες, και για τις μεθόδους ε-greedy και Forced Exploration. Και εδώ επιβεβαιώνονται τα προηγούμενα μας συμπεράσματα, για τις μεθόδους ε-greedy και Forced exploration.

Πίνακας 5.3.1 Στατιστικά εκπαίδευσης για την Q Learning 23

	Για 4 παίκτες		Για 2 παίκτες	
Μεταβλητή	**ε-greedy**	**Forced Exploration**	**ε-greedy**	**Forced Exploration**
Καλύτερο ποσοστό νικών κατά τον έλεγχο	*71,98*	*70,78*	*67,47*	*75,61*
Αριθμός παιχνιδιών εκπαίδευσης για το καλύτερο αποτέλεσμα	*3.030.000*	*4.880.000*	*8.410.000*	*9.850.000*
Μέσος αριθμός γύρων	*16,34*	*16,14*	*28,74*	*28,26*
Εξερευνημένες καταστάσεις	*1.774.188*	*2.677.854*	*9.518.678*	*14.111.810*
Συνολικός αριθμός καταστάσεων	*7.188.480*	*7.188.480*	*38.707.200*	*38.707.200*

Στον πίνακα 5.3.2, βλέπουμε τη μέση τιμή του αριθμού των καρτών που αγόρασε ο πράκτορας, κατά τα παιχνίδια ελέγχου που πέτυχε το καλύτερο αποτέλεσμα, για κάθε είδος κάρτας. Όπως βλέπουμε η στρατηγική που αναπτύσσει ο πράκτορας είναι παρόμοια, ανεξαρτήτως της μεθόδου που χρησιμοποιήσαμε. Αποφεύγει να αγοράζει χάλκινα και προτιμά τα ασημένια και τα χρυσά, ενώ προτιμά τις κάρτες βασιλείου που του επιτρέπουν να τραβήξει επιπλέον κάρτες, με μακράν πρώτη το σιδερά και δεύτερη το χωριό. Η συχνότητα με την οποία εμφανίζονται οι άλλες κάρτες στην τράπουλα του είναι σχεδόν μηδαμινή.

Πίνακας 5.3.2 Μέσες τιμές αριθμού καρτών

	Για 4 παίκτες		Για 2 παίκτες	
Κάρτα	**ε-greedy**	**Forced Exploration**	**ε-greedy**	**Forced Exploration**
Χάλκινο	*7,45*	*7,46*	*7,57*	*7,58*
Ασημένιο	*3,75*	*3,53*	*5,95*	*4,62*
Χρυσό	*1,53*	*1,60*	*2,29*	*3,13*
Σιδεράς	*1,00*	*1,04*	*1,76*	*1,69*
Παρεκκλήσι	*0,01*	*0,03*	*0,02*	*0,08*
Αγορά	*0,01*	*0,00*	*0,02*	*0,18*
Τυχοδιώκτης	*0,00*	*0,02*	*0,00*	*0,03*
Συνωμότης	*0,02*	*0,10*	*0,14*	*0,09*
Γιορτή	*0,00*	*0,00*	*0,00*	*0,02*
Τοκογλύφος	*0,00*	*0,03*	*0,01*	*0,06*
Χωριό	*0,07*	*0,19*	*0,13*	*0,16*
Ξυλοκόπος	*0,00*	*0,01*	*0,04*	*0,06*
Χωριό εργατών	*0,00*	*0,02*	*0,00*	*0,02*

ΚΕΦΑΛΑΙΟ 6:
Συμπεράσματα

Όπως φαίνεται από τα παραπάνω. ο βασικός στόχος της εργασίας ήταν επιτυχής. Καταφέραμε να αναπτύξουμε αλγόριθμο εξαναγκασμένης μάθησης, ο οποίος εκπαίδευσε ικανό να κερδίζει τους AI αντιπάλους. Μάλιστα όπως δείχνουν τα αποτελέσματα των δοκιμών, η στρατηγική που ανέπτυσσε ο πράκτορας ήταν παρόμοια σε όλες τις διαφορετικές εκδοχές της εκπαίδευσης που του κάναμε, και δεν επηρεάστηκε από τις τυχαίες αρχικές τιμές της Q(s,α). Το μεγάλο ποσοστό νικών εξάλλου, στα παιχνίδια δοκιμών, μας κάνει να πιστεύουμε ότι αυτή είναι και η βέλτιστη στρατηγική, δεδομένων των καρτών βασιλείου που χρησιμοποιήθηκαν. Άρα επιβεβαιώσαμε ότι η μέθοδος Q Learning είναι ικανή να ανταποκριθεί σε πολύπλοκα προβλήματα, που δημιουργούν μεγάλους χώρους καταστάσεων.

Το πρόβλημα που αναδείχθηκε όμως, από τα παραπάνω πειράματα, δεν ήταν η αποτελεσματικότητα της μεθόδου, αλλά η δυσκολία της εφαρμογής της λόγω πρακτικών προβλημάτων. Ο αλγόριθμος που αναπτύξαμε είναι αρκετά γρήγορος, καθώς 1.000.000 παιχνίδια εκπαίδευσης διαρκούσαν περίπου δέκα λεπτά σε έναν υπολογιστή με εξαπύρηνο επεξεργαστή, συχνότητας 3,3GHz, και μνήμης 16 Gbyte, αλλά η αποθήκευση των τιμών της Q(s,α) και ακόμα χειρότερα η ανάγνωση τους κατά την εκκίνηση της εκπαίδευσης ήταν ιδιαιτέρως χρονοβόρα. Μάλιστα σε δοκιμές που κάναμε με μεγαλύτερους χώρους καταστάσεων, η εφαρμογή κατέρρεε, λόγω ανεπαρκούς μνήμης. Επίσης όπως φάνηκε από τα πειράματα μας, αύξηση του χώρου των καταστάσεων συνεπάγεται και αύξηση του αριθμού των παιχνιδιών εκπαίδευσης, έως ότου πετύχουμε καλά αποτελέσματα. Άρα λόγω πρακτικών δυσκολιών εναλλακτικές προσεγγίσεις ίσως να είναι προτιμότερες, όπως αυτή του Winder (11), ο οποίος χρησιμοποίησε νευρωνικά δίκτυα για την εκπαίδευση (Deep Q Learning). Η μέθοδος αυτή παρόλο που δεν εμφανίζει τόσο καλά αποτελέσματα όσο η Q Learning (ίσως λόγω του μικρού αριθμού παιχνιδιών εκπαίδευσης, 686.000 έναντι 10.000.000 της Q Learning), έχει το πλεονέκτημα ότι μπορεί να επεκταθεί εύκολα και σε προβλήματα με μεγαλύτερους χώρους καταστάσεων.

Τέλος, και η μέθοδος επιλογής ενεργειών που δοκιμάσαμε παρουσιάζει ενδιαφέρον, καθώς μπορεί να εφαρμοστεί και στην Deep Q Learning. Παρόλο που όπως φάνηκε δεν υπερέχει της ε-greedy σε όλες τις περιπτώσεις, τα αποτελέσματα είναι ενθαρρυντικά, και θεωρούμε ότι χρήζει περαιτέρω έρευνας.

ΒΙΒΛΙΟΓΡΑΦΙΑ - ΠΗΓΕΣ

[1] C. E. Shannon and J. McCarthy, 1956, "Automata Studies" (AM-34) (Annals of Mathematics Studies), Princeton University Press.

[2] Feigenbaum, Edward A and Barr, Avron and Cohen, Paul R, 1981, "The handbook of artificial intelligence", Addison-Wesley.

[3] A.M. Turing, 1950, "Computing machinery and intelligence", Springer.

[4] Bellman, R. E. (1957). Dynamic Programming. Princeton University Press, Princeton, NJ.

[5] G. Tesauro, "TD-Gammon, a Self-Teaching Backgammon Program, Achieves Master-Level Play" Neural Computation, vol. 6, no. 2, pp. 215-219, Mar. 1994, 10.1162/neco.1994.6.2.215

[6] Jonathan Baxter, Andrew Tridgell, and Lex Weaver (1997) KnightCap: A chess program that learns by combining TD(λ) with minimax search. Technical Report, Learning Systems Group, Australian National University

[7] R. Ekker, E.C.D. van der Werf, and L.R.B. Schomaker (2004) Dedicated TD-Learning for Stronger Gameplay: applications to Go. Benelearn'04: Proceedings of the Thirteenth Belgian-Dutch Conference on Machine Learning

[8] Jonathan Schaeffer, Markian Hlynka and Vili Jussila (2001) Temporal Difference Learning Applied to a High-Performance Game-Playing Program. Proceedings of the 2001 International Joint Conference on Artificial Intelligence (IJCAI-2001), 529-534.

[9] Daniel Kenneth Olson (1993) Learning to Play Games from Experience: An Application of Artificial Neural Networks and Temporal Difference Learning. M.S. thesis, Pacific Lutheran University, Washington.

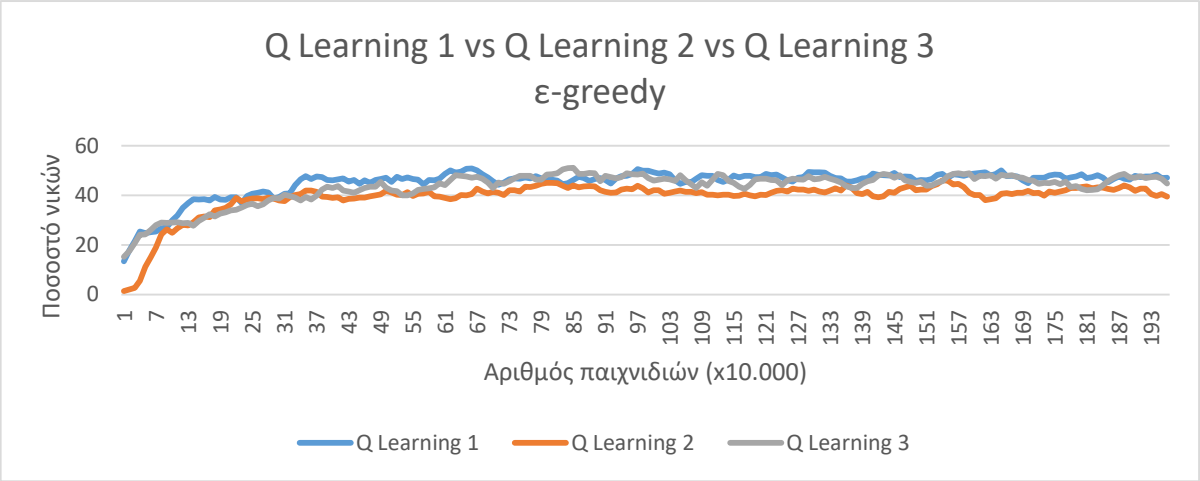
[10] M. Pfeiffer, "Reinforcement Learning of Strategies for Settlers of Catan," in International Conference on Computer Games: Artificial Intelligence, Design and Education, Wolverhampton, UK, 2004, pp. 384-388.

[11] Winder, Ransom K. "Methods for approximating value functions for the Dominion card game." Evolutionary Intelligence 6.4 (2014): 195-204

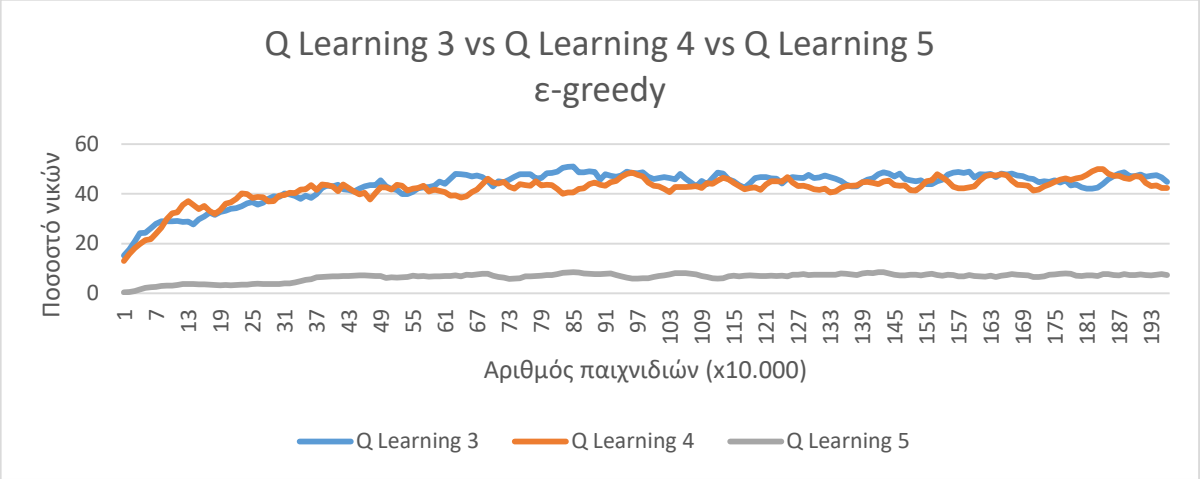
[12] A.A. Markov "The Correspondence between A.A. Markov and A.A. Chuprov on the theory of Probability and Mathematical statistics"

[13] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

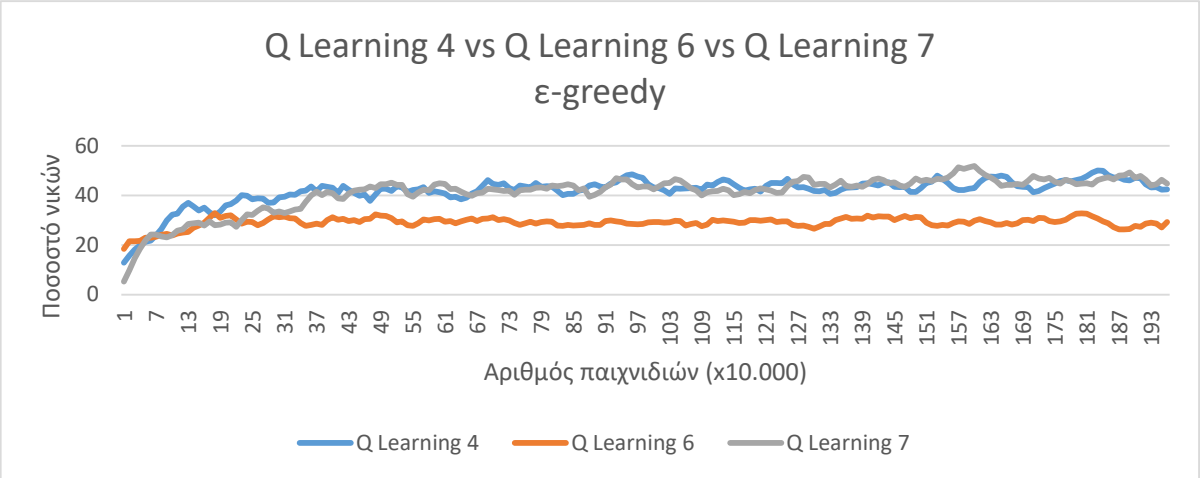
ΠΑΡΑΡΤΗΜΑ 1



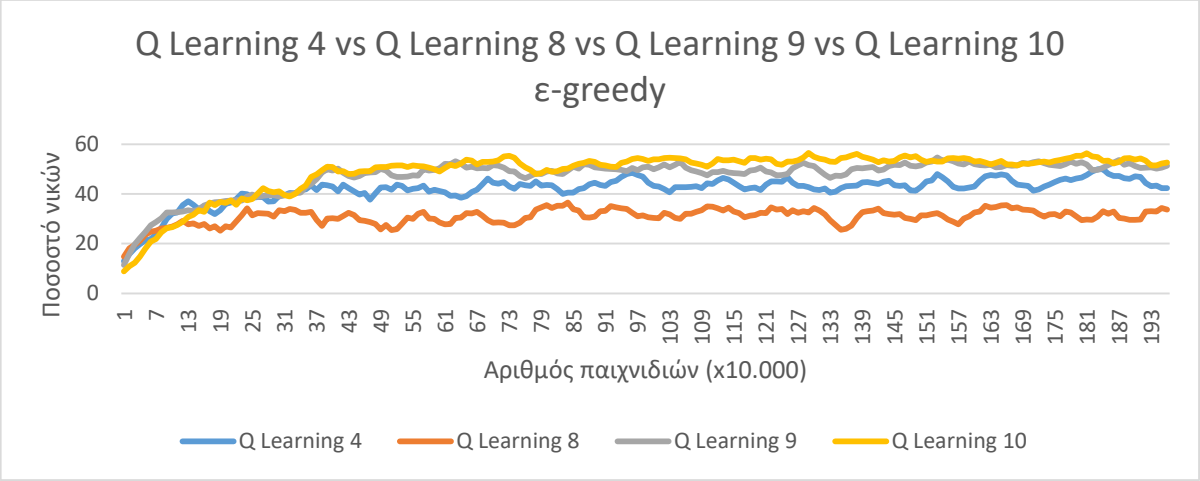
Εικόνα Π1.1 Σύγκριση μεταξύ των Q Learning 1, 2, 3 για την μέθοδο ε-greedy



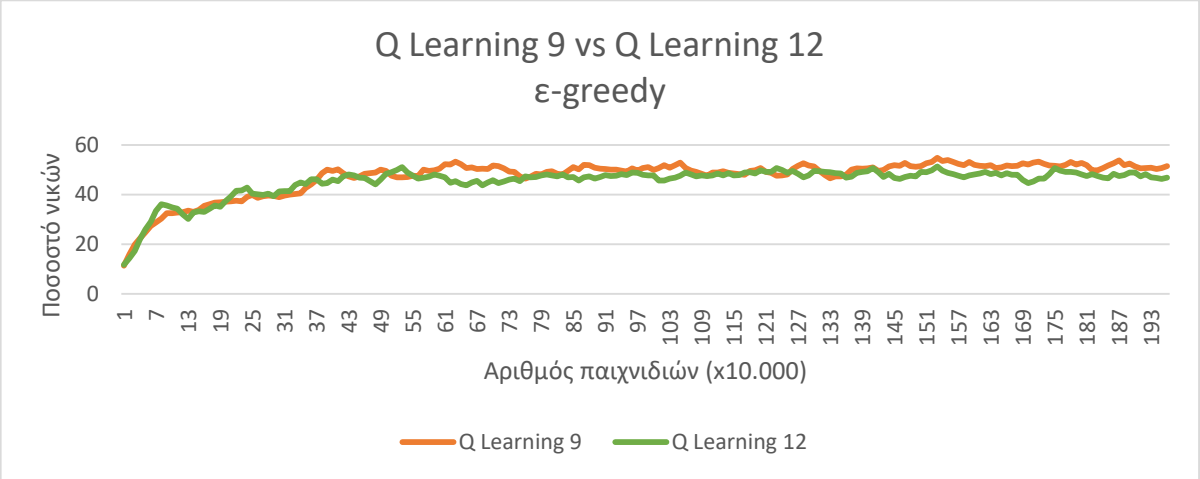
Εικόνα Π1.2 Σύγκριση μεταξύ των Q Learning 3, 4, 5 για την μέθοδο ε-greedy



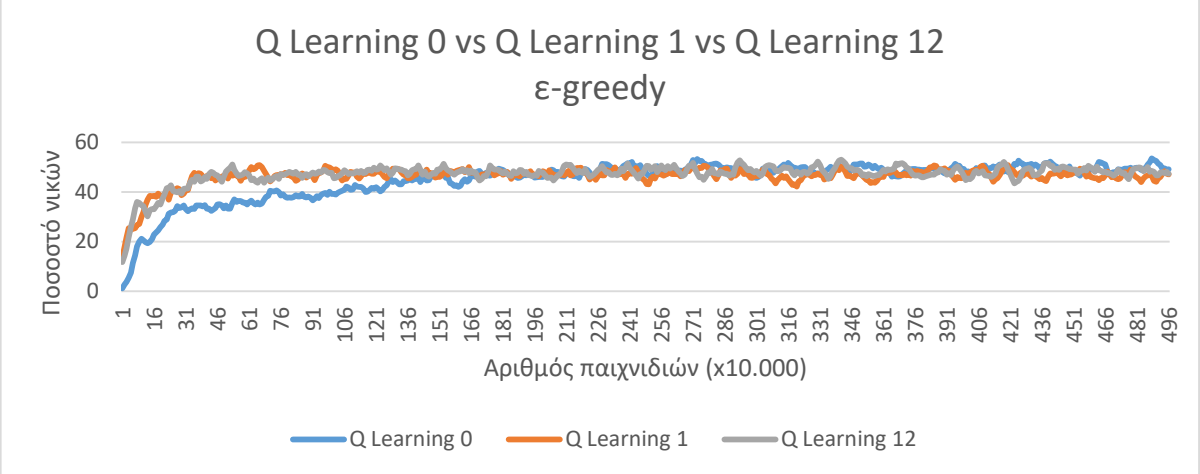
Εικόνα Π1.3 Σύγκριση μεταξύ των Q Learning 4, 6, 7 για την μέθοδο ε-greedy



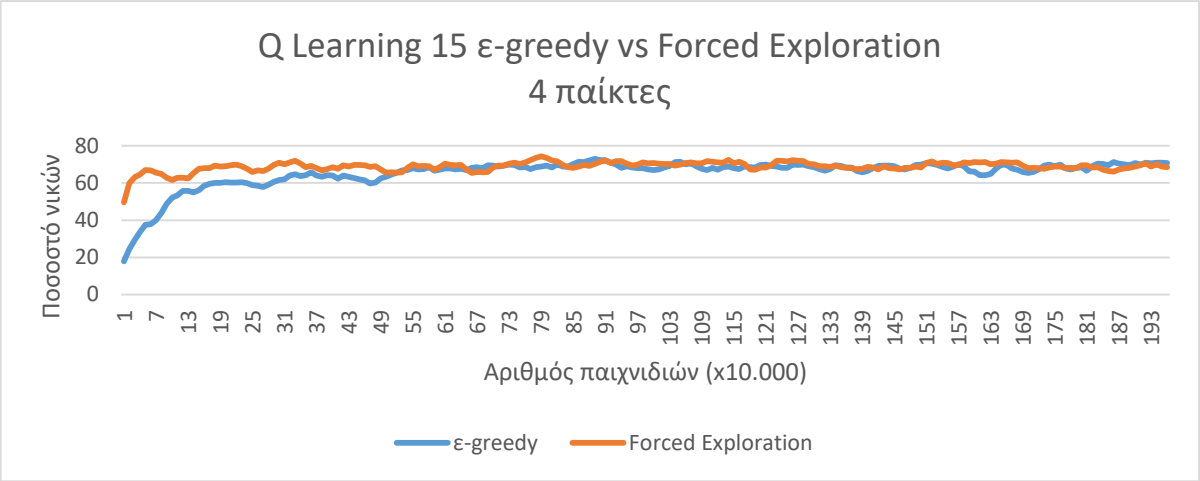
Εικόνα Π1.4 Σύγκριση μεταξύ των Q Learning 4, 8, 9, 10 για την μέθοδο ε-greedy



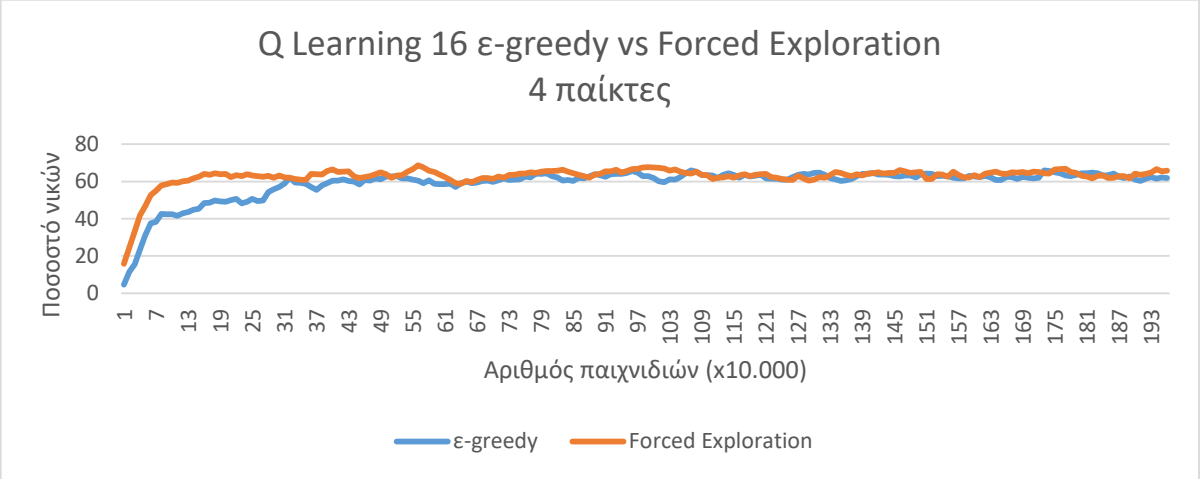
Εικόνα Π1.5 Σύγκριση μεταξύ των Q Learning 9, 12 για την μέθοδο ε-greedy



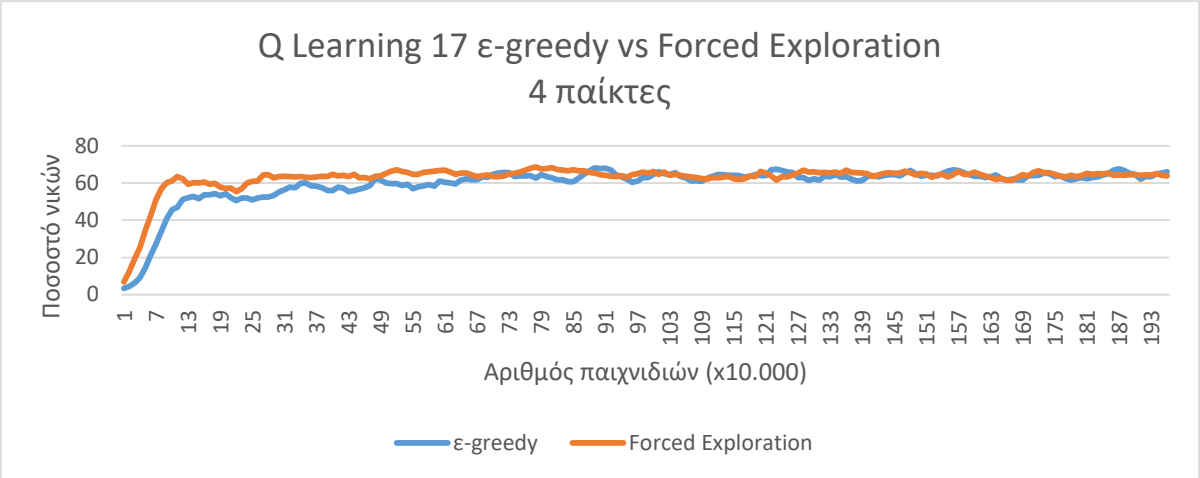
Εικόνα Π1.6 Σύγκριση μεταξύ των Q Learning 0, 1, 12 για την μέθοδο ε-greedy



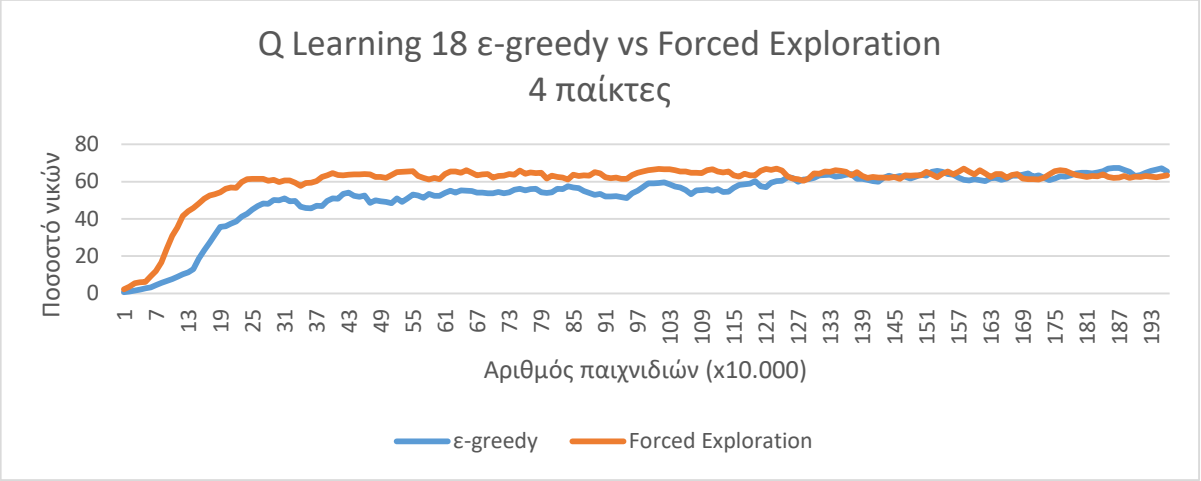
Εικόνα Π2.1 Κάρτα «Παρεκκλήσι» για 4 παίκτες



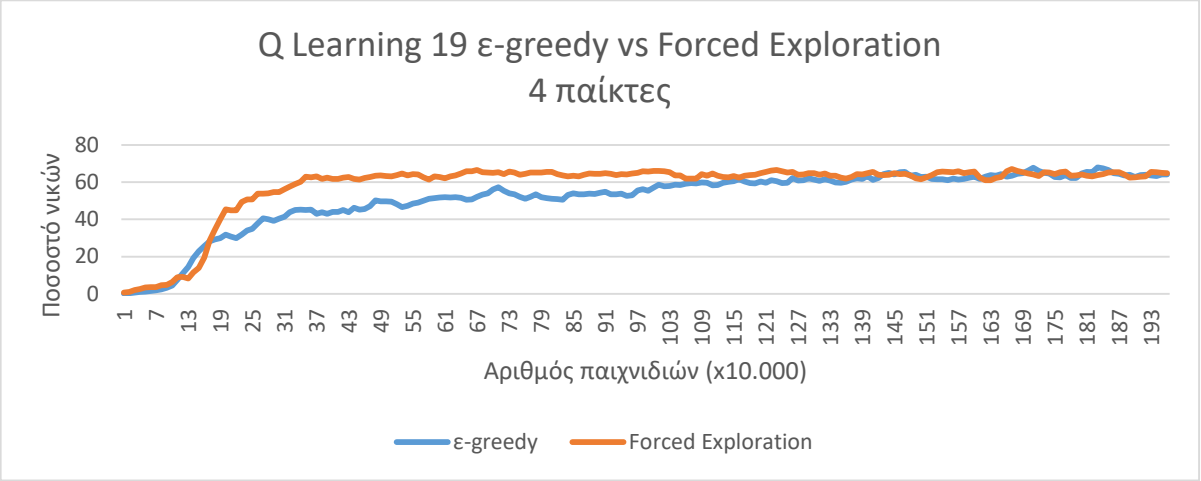
Εικόνα Π2.2 Κάρτα «Αγορά» για 4 παίκτες



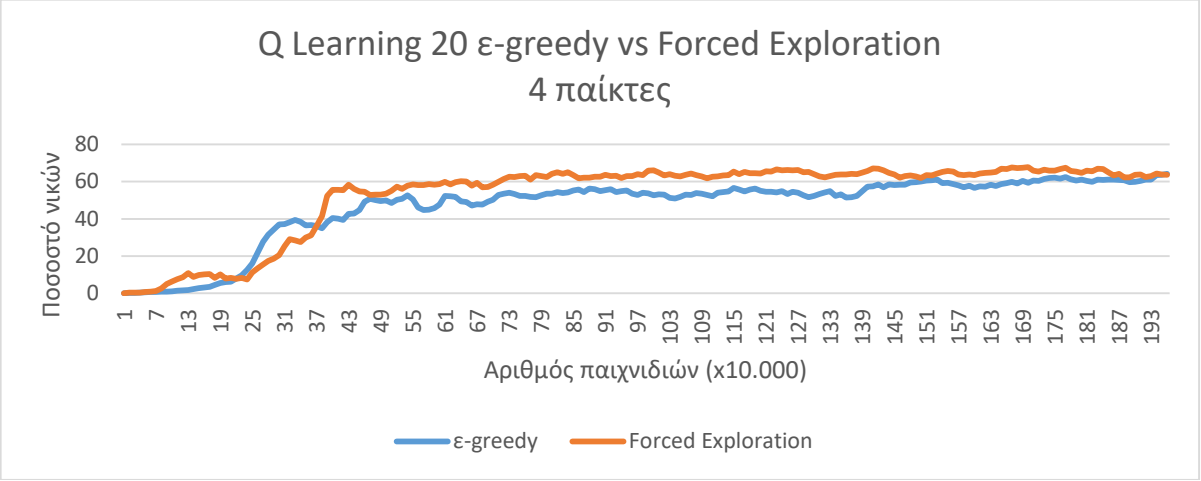
Εικόνα Π2.3 Κάρτα «Τυχοδιώκτης» για 4 παίκτες



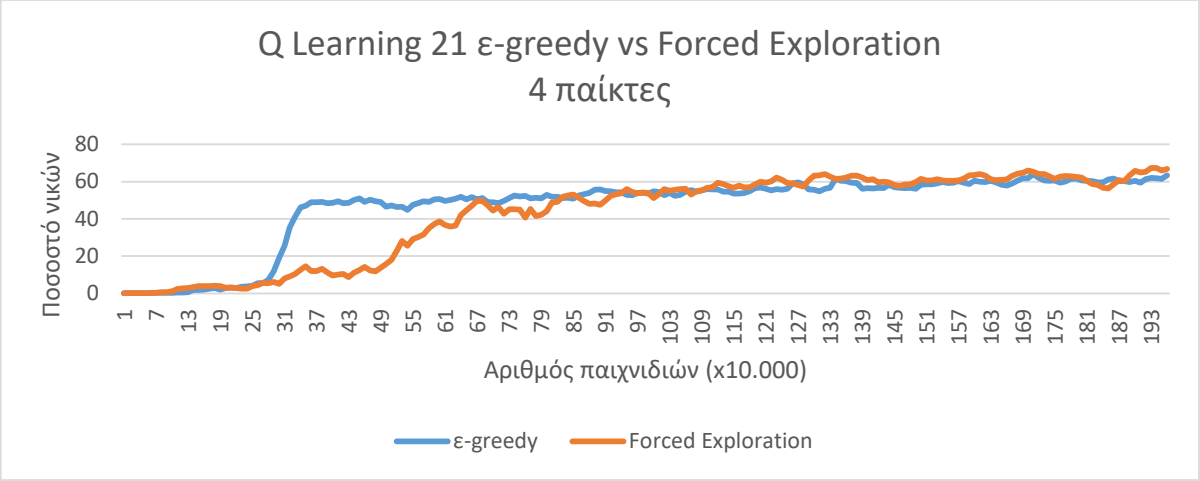
Εικόνα Π2.4 Κάρτα «Συνωμότης» για 4 παίκτες



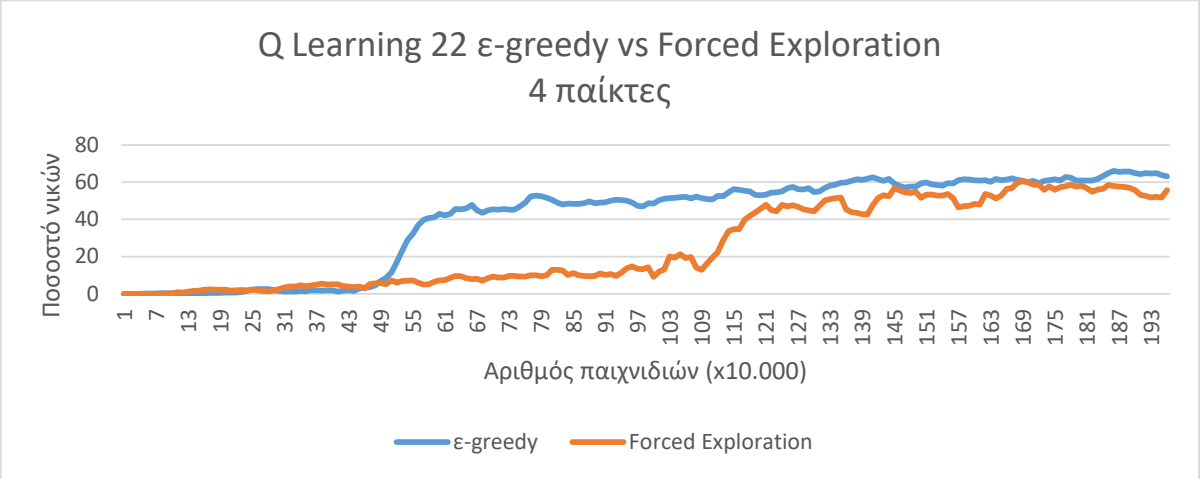
Εικόνα Π2.5 Κάρτα «Γιορτή» για 4 παίκτες



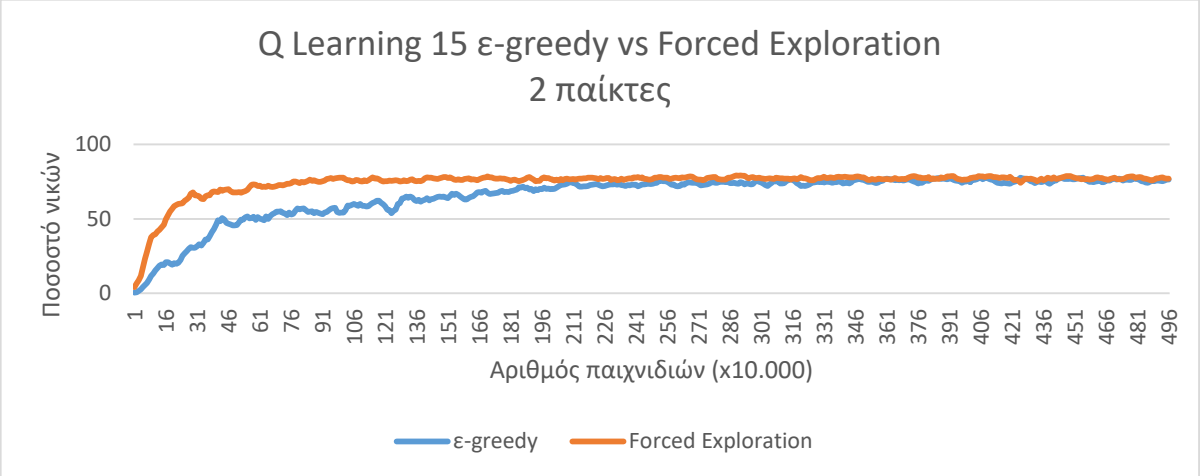
Εικόνα Π2.6 Κάρτα «Τοκογλύφος» για 4 παίκτες



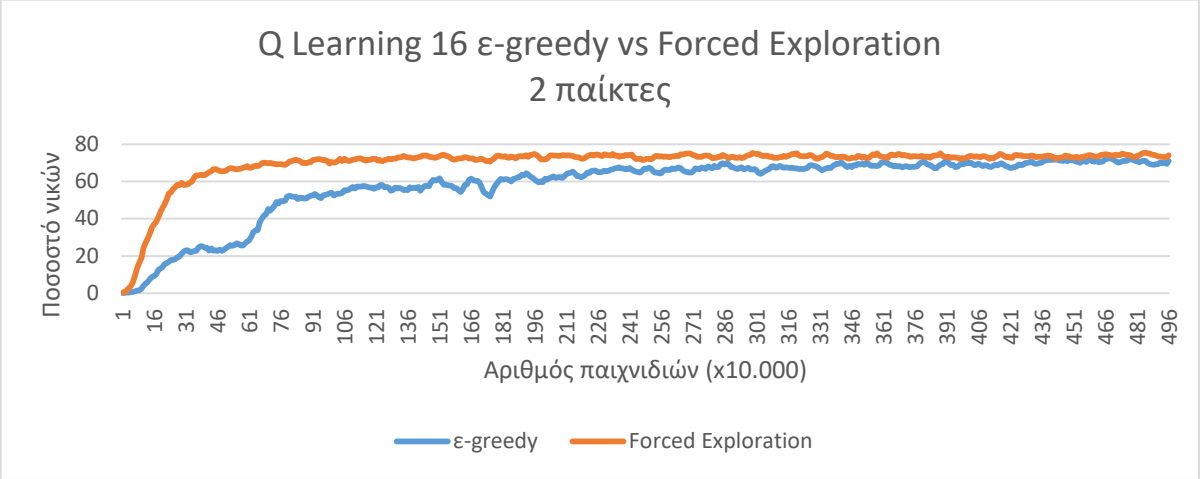
Εικόνα Π2.7 Κάρτα «Χωριό» για 4 παίκτες



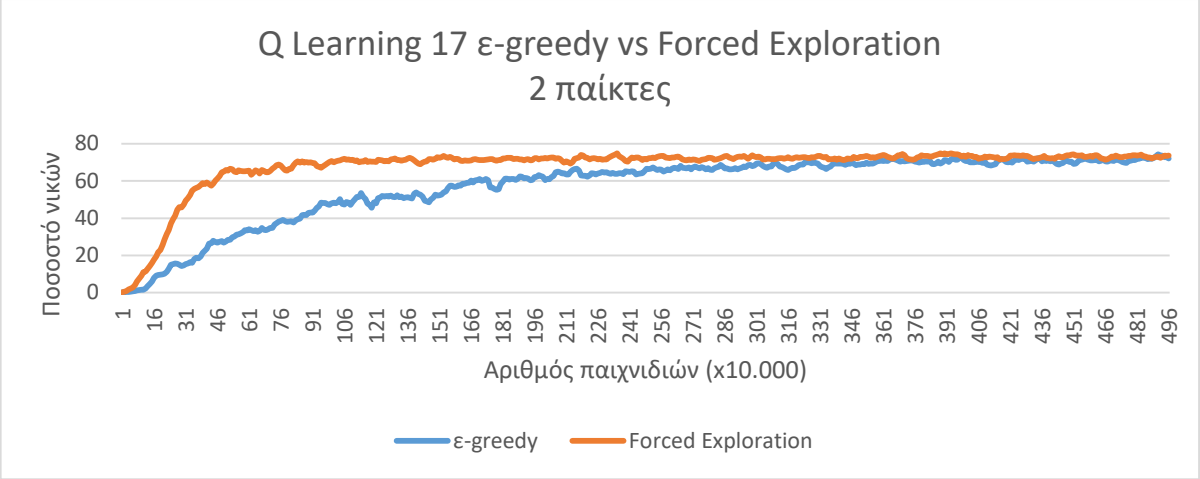
Εικόνα Π2.8 Κάρτα «Ξυλοκόπος» για 4 παίκτες



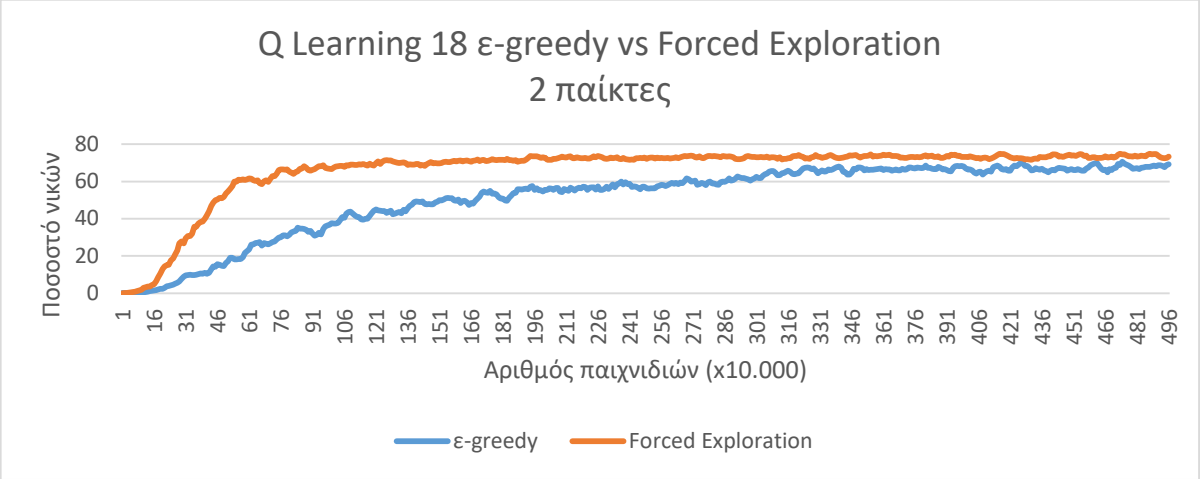
Εικόνα Π2.9 Κάρτα «Παρεκκλήσι» για 2 παίκτες



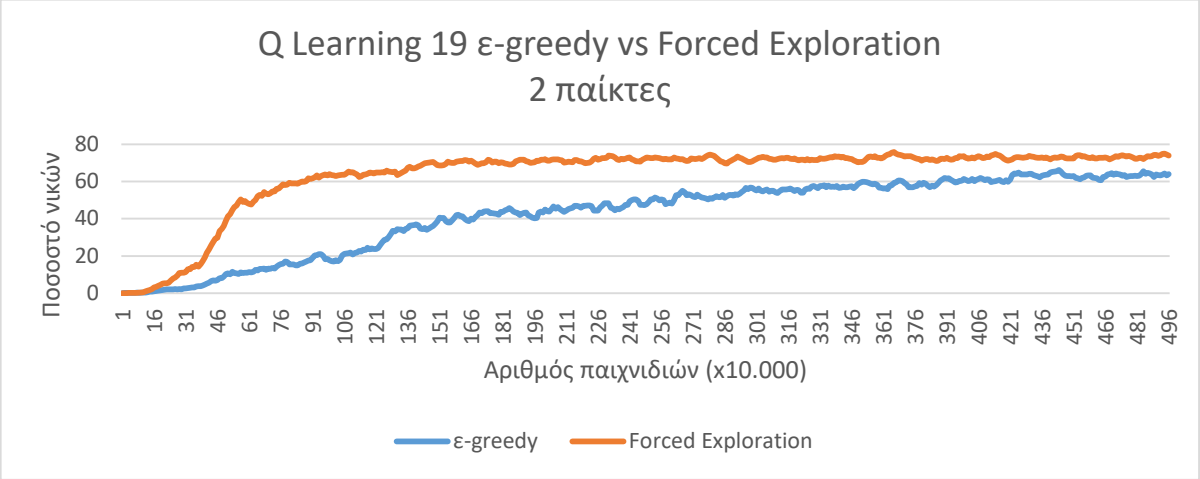
Εικόνα Π2.10 Κάρτα «Αγορά» για 2 παίκτες



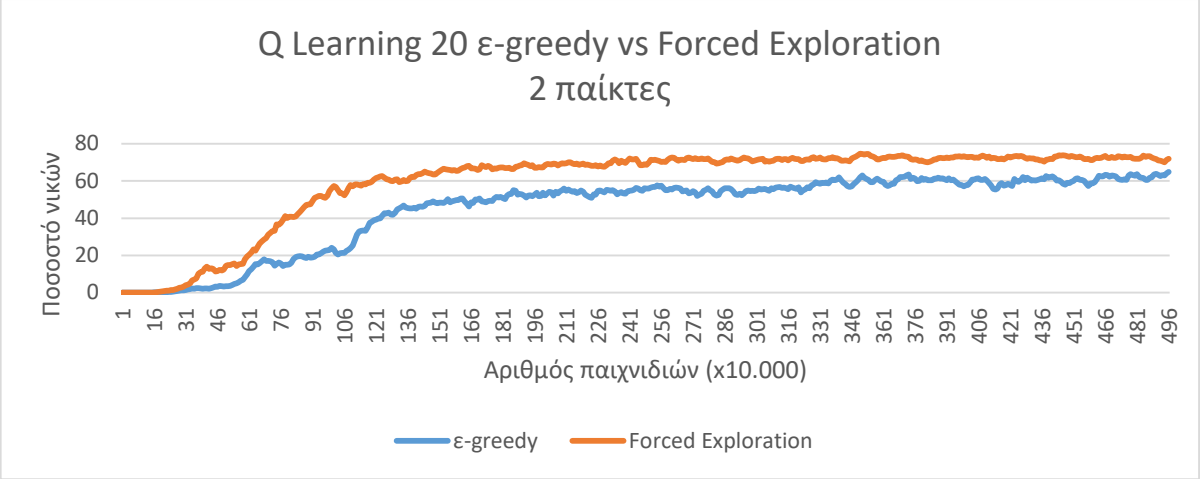
Εικόνα Π2.11 Κάρτα «Τυχοδιώκτης» για 2 παίκτες



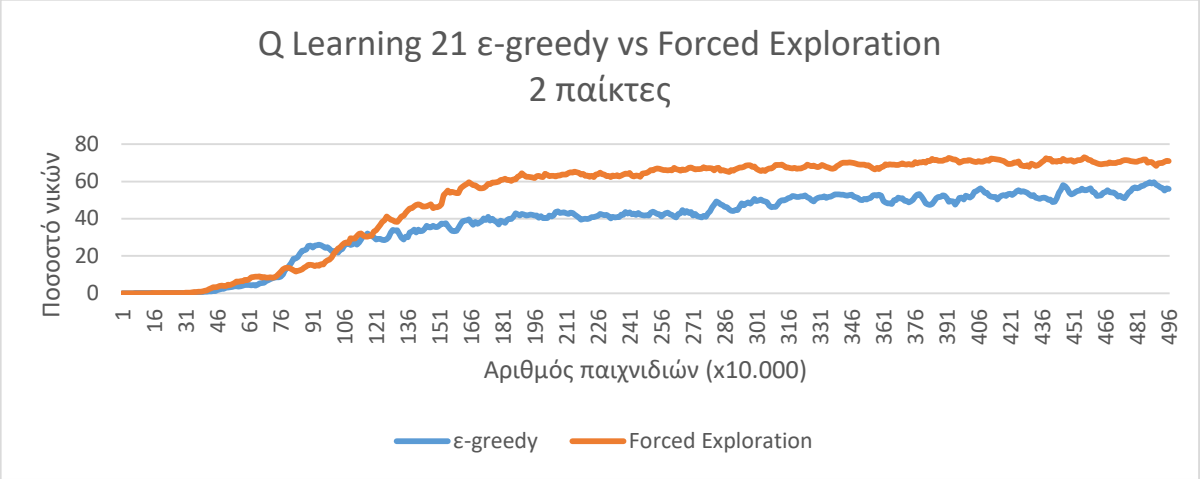
Εικόνα Π2.12 Κάρτα «Συνωμότης» για 2 παίκτες



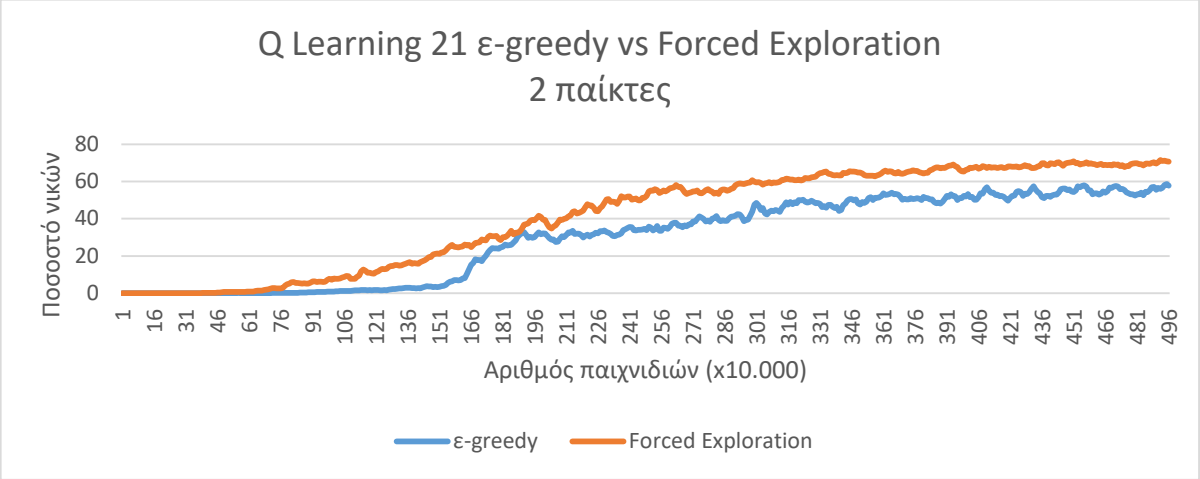
Εικόνα Π2.13 Κάρτα «Γιορτή» για 2 παίκτες



Εικόνα Π2.14 Κάρτα «Τοκογλύφος» για 2 παίκτες



Εικόνα Π2.15 Κάρτα «Χωριό» για 2 παίκτες



Εικόνα Π2.16 Κάρτα «Ξυλοκόπος» για 2 παίκτες