

Ατομική Διπλωματική Εργασία

ΠΡΟΣΕΓΓΙΣΗ ΚΥΠΡΙΑΚΟΥ ΠΡΟΒΛΗΜΑΤΟΣ
ΜΕ ΕΝΙΣΧΥΤΙΚΗ ΜΑΘΗΣΗ ΠΟΛΛΑΠΛΩΝ ΠΡΑΚΤΟΡΩΝ
ΣΕ ΠΑΙΓΝΙΑ ΓΕΝΙΚΟΥ ΑΘΡΟΙΣΜΑΤΟΣ

Αντρέας Ιωάννου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2017

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΠΡΟΣΕΓΓΙΣΗ ΚΥΠΡΙΑΚΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΜΕ ΕΝΙΣΧΥΤΙΚΗ ΜΑΘΗΣΗ
ΠΟΛΛΑΠΛΩΝ ΠΡΑΚΤΟΡΩΝ ΣΕ ΠΑΙΓΝΙΑ ΓΕΝΙΚΟΥ ΑΘΡΟΙΣΜΑΤΟΣ**

Αντρέας Ιωάννου

Επιβλέπων Καθηγητής

Δρ. Χρίστος Χριστοδούλου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του
Πανεπιστημίου Κύπρου

Μάιος 2017

Ευχαριστίες

Θα ήθελα να ευχαριστήσω ιδιαιτέρως τον Δρ. Χρίστο Χριστοδούλου, ο οποίος σαν επιβλέπων καθηγητής μου, με καθοδηγούσε συνεχώς με σιγουριά και ήταν πάντα πρόθυμος να βοηθήσει όποτε χρειαζόμουν βοήθεια η καθοδήγηση σε αυτό το έργο.

Επίσης θα ήθελα να ευχαριστήσω θερμά την οικογένεια μου, πού με όποιο τρόπο μπορούσε ήταν πάντα στο πλευρό μου και με βοήθησε να φτάσω μέχρι εδώ.

Περίληψη

Η διπλωματική αυτή παρουσιάζει την προσέγγιση του κυπριακού προβλήματος σαν ένα παίγνιο γενικού αθροίσματος, το bargaining game και με την βοήθεια ενισχυτικής μάθησης, να δούμε κατά πόσο οι 2 πλευρές μπορούν να φτάσουν σε συμβιβασμό μέσω του περιβάλλοντος του επαναλαμβανόμενου bargaining game. Διερευνήθηκαν οι καταστάσεις και οι απαιτήσεις των θέσεων των 2 πλευρών, της Ελληνοκυπριακής(ΕΚ) πλευράς και της Τουρκοκυπριακής (ΤΚ) πλευράς για το παρόν χρονικό στάδιο, και προσομοιώνεται μία επαναλαμβανόμενη διαπραγμάτευση η οποία αφορά τη γεωγραφική πτυχή του κυπριακού προβλήματος μεταξύ των 2 πλευρών σε 2 αλγόριθμους της ενισχυτικής μάθησης, τον αλγόριθμο Q-learning και το αλγόριθμο SARSA με e-greedy τακτική. Στην αντιστοίχιση έχει προσαρμοστεί το σύστημα έτσι ώστε να ασχολείται με το γεωγραφικό, καθαρά, μέρος του κυπριακού, αφήνοντας τις άλλες πτυχές σαν τιμωρίες η έμμεσους παράγοντες. Τα αποτελέσματα που παίρνουμε μας δείχνουν ότι όντως μπορεί να μοντελοποιηθεί και να προσεγγιστεί στα δεδομένα του κυπριακού προβλήματος το επαναλαμβανόμενο bargaining game καθώς παρουσιάζονται οι συμπεριφορές των 2 πλευρών πολύ ρεαλιστικά. Επίσης τα αποτελέσματα δείχνουν ότι η διαπραγμάτευση μπορεί να καταλήξει σε συμβιβασμό μετά από κάποιο αριθμό επαναλήψεων όπου τελικά θα μάθουν οι 2 πράκτορες (πλευρές) ότι ο συμβιβασμός, είναι η καλύτερη κοινή λύση. Για να φτάσουν στο συμβιβασμό θα δοκιμάσουν άλλες επιλογές, συνήθως πιο εγωιστικές και ατομικές για τον λόγο ότι δεν θέλουν να κάνουν υποχωρήσεις, όμως θα μάθουν από αυτές όλες τις προσπάθειες και επαναλήψεις ότι είναι σχεδόν αδύνατο να έχουμε ένα μόνο κερδισμένο, αντιθέτως το ενδεχόμενο να βγούν και οι 2 ηττημένοι έχει περισσότερη πιθανότητα να συμβεί. Μελετήθηκαν επίσης διάφορες παράμετροι του κάθε αλγορίθμου και συνδυασμοί, καθώς και σύγκριση μεταξύ των 2 αλγορίθμων για να δούμε αν θα έφερναν πιο ρεαλιστικά ή πιο καλά αποτελέσματα και ποιος αλγόριθμος είναι καλύτερος .

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	9
Κεφάλαιο 2	Γνωσιολογικό υπόβαθρο.....	12
	2.1 Στοιχεία Ενισχυτικής μάθησης.....	12
	2.2 Μέθοδοι Ενισχυτικής Μάθησης.....	16
	2.3 Q-Learning.....	17
	2.4 SARSA.....	18
	2.5 Θεωρία Παιγνίων.....	19
	2.6 Bargaining Game.....	22
	2.7 Επαναλαμβανόμενο Bargaining Game.....	23
	2.8 Κυπριακό Ζήτημα.....	23
	2.9 Προηγούμενη δουλειά πάνω στο θέμα.....	25
Κεφάλαιο 3	Περιγραφή συστήματος.....	27
	3.1 Εισαγωγή.....	27
	3.2 Σχεδιασμός	28
	3.3 Υλοποίηση.....	29
Κεφάλαιο 4	Αντιστοίχιση «Κυπριακού Προβλήματος» στο Bargaining game....	39
	4.1 Εισαγωγή.....	39
	4.2 Κυπριακό Πρόβλημα σήμερα	39
	4.3 Αντιστοίχιση Κυπριακού προβλήματος στο σύστημα.....	43

Κεφάλαιο 5	Αποτελέσματα και ανάλυση.....	52
5.1	Εισαγωγή.....	53
5.2	Αποτελέσματα και ανάλυση κανονικού συστήματος για παιχνίδι Bargaining Game	53
5.3	Αποτελέσματα και ανάλυση Αντιστοίχισης Κυπριακού προβλήματος σε Bargaining Game στο σύστημα.....	56
5.3.1	Πράκτορες με αλγόριθμο Q-learning.....	56
5.3.1.1	Μελέτη παραμέτρου ρυθμού μάθησης.....	56
5.3.1.2	Μελέτη παραμέτρου έκπτωσης.....	60
5.3.1.2.1	Μελέτη με διαφορετικούς παραμέτρους έκπτωσης για κάθε πράκτορα.....	62
5.3.1.3	Μελέτη παραμέτρου επαναλήψεων.....	64
5.3.1.4	Βέλτιστες παραμέτροι για αλγόριθμο Q-learning...	66
5.3.2	Πράκτορες με αλγόριθμο SARSA με e-greedy τακτική..	67
5.3.2.1	Μελέτη παραμέτρου ρυθμού μάθησης.....	68
5.3.2.2	Μελέτη παραμέτρου έκπτωσης.....	71
5.3.2.3	Μελέτη παραμέτρου επαναλήψεων.....	74
5.3.2.4	Μελέτη παραμέτρου ϵ	75
5.3.2.5	Βέλτιστες παραμέτροι για αλγόριθμο SARSA με e-greedy τακτική.....	78
5.3.3	Πειραματισμός με διαφορετικά ποσά συμβιβασμού.....	79
5.3.4	Συμπεράσματα και σύγκριση αλγορίθμων.....	83

Κεφάλαιο 6	Συμπεράσματα και μελλοντική εργασία.....	86
6.1	Σύνοψη.....	86
6.2	Συμπεράσματα	87
6.3	Σύγκριση με άλλες Εργασίες.....	88
6.4	Μελλοντική εργασία.....	89
Αναφορές		91
Παράρτημα Α: Πηγαίος Κώδικας.....		A-1
Παράρτημα Β: Παραδείγματα αρχείων εισόδου.....		B-1

Λίστα ακρωνύμων

ΕΚ	Ελληνοκυπριακή πλευρά/ Ελληνοκύπριοι
ΤΚ	Τουρκοκυπριακή πλευρά/ Τουρκοκύπριοι
ΕΔΦ	Επαναλαμβανόμενο Δίλλημα φυλακισμένου
ΕΜ	Ενισχυτική μάθηση
ΧΔ	Χρονική Διαφορά
ΔΠ	Δυναμικός προγραμματισμός

Κεφάλαιο 1

Εισαγωγή

Σήμερα, υπάρχουν πολλά πραγματικά συμβάντα τα οποία δημιουργούν καταστάσεις στις οποίες οι εμπλεκόμενοι βρίσκονται σε αντίπαλη θέση μεταξύ τους, και μέσω αυτής της εχθρικότητας και των απαιτήσεων τις κάθε πλευράς υπάρχουν θέματα που εμποδίζουν τους εμπλεκόμενους να φτάσουν σε συμβιβασμό.

Η ενισχυτική μάθηση[5 9] , ένας κλάδος της Τεχνητής Νοημοσύνης υπήρχε σαν ενεργό πεδίο εδώ και πολλά χρόνια (Sutton και Barto)(1998)[9]. Η ενισχυτική μάθηση είναι μια μέθοδος όπου ένας πράκτορας μπορεί να μάθει μέσω κάποιων αλληλεπιδράσεων με το περιβάλλον τι είναι καλό και τι κακό για αυτόν. Οι αλληλεπιδράσεις αυτές είναι ότι, για κάθε δράση του αξιολογείται θετικά με επιβράβευση, και αρνητικά με τιμωρία, με απώτερο σκοπό να μάθει μέσα από αυτές τις αξιολογήσεις ποιες δράσεις του θα του επιφέρουν την μεγαλύτερη συνολική ανταμοιβή, δηλαδή καλύτερες πιθανότητες να επιτύχει τον σκοπό του. Λόγω της φύσης της, η ενισχυτική μάθηση μπορεί εύκολα να κατανοηθεί από εμάς , καθώς και εμείς την εφαρμόζουμε καθημερινά στην ζωή μας, μαθαίνοντας από τα λάθη και τις σωστές πράξεις μας.

Στην ενισχυτική μάθηση υπάρχουν κάποιες τακτικές που μπορεί να επιλέξει ο πράκτορας, οι άπληστες , οι μερικώς άπληστες και οι μη-άπληστες. Μια τακτική χαρακτηρίζεται σαν άπληστη όταν δεν επιτρέπει στον πράκτορα να εξερευνά το περιβάλλον, δηλαδή ο πράκτορας εκμεταλλεύεται μόνο την γνώση που έχει. Αντιθέτως μια τακτική θεωρείται μη-άπληστη όταν ο πράκτορας εξερευνά όλο και περισσότερο το περιβάλλον μη λαμβάνοντας υπόψη την γνώση που έχει. Τέλος η μερικώς άπληστη τακτική είναι κάτι ενδιάμεσο στις 2 προηγούμενες, καθώς με μία πιθανότητα e ο πράκτορας εξερευνά , και με την υπόλοιπη πιθανότητα $(1-e)$ εκμεταλλεύεται την γνώση του.

Η θεωρία παιγνίων [2] είναι μια μεθοδολογία ανάλυσης αποφάσεων και συμπεριφορών μεταξύ των συμμετεχόντων με σκοπό να κερδίσει ο καθένας το μεγαλύτερο κέρδος που μπορεί. Εμάς μας βοηθά στο να δούμε πως συμπεριφέρονται και πώς αλληλοεπιδρούν οι οντότητες, οι οποίες ονομάζονται παίκτες, και πώς

αποφασίζουν ποιες δράσεις θα εκτελέσουν από κάθε πιθανή κατάσταση. Επομένως η εκμάθηση σε πραγματικές και ρεαλιστικές καταστάσεις μπορεί να προσεγγιστεί και να αντιστοιχηθεί με παίγνια .

Τα παίγνια μπορούν να κατηγοριοποιηθούν σε 2 κατηγορίες αναλόγως με την φύση και δομή τους, τα παίγνια μηδενικού αθροίσματος (zero-sum games) και τα παίγνια μη-μηδενικού αθροίσματος ή γενικού αθροίσματος (non zero-sum games ή general sum games). Στα παίγνια μηδενικού αθροίσματος το κέρδος ενός παίκτης είναι η ζημία του άλλου, επομένως δεν υπάρχει ενδεχόμενο συμβιβασμού, ενώ στα παίγνια μη-μηδενικού αθροίσματος το άθροισμα των κερδών δεν είναι 0, δηλαδή δεν υπάρχει πάντα ένας νικητής και ένας ηττημένος, καθώς υπάρχουν ενδεχόμενα όπου και οι πράκτορες θα χάσουν, ή και οι 2 θα βγουν κερδισμένοι . Εμείς θα ασχοληθούμε με παίγνια μη-μηδενικού αθροίσματος μεταξύ 2 παικτών (ή γενικού αθροίσματος) και πιο συγκεκριμένα το bargaining game.

Το bargaining game[8] είναι ένα παίγνιο γενικού αθροίσματος στο οποίο οι παίχτες έχουν ένα αρχικό ποσό το οποίο θέλουν να μοιράσουν, και να κερδίσουν όσο δυνατό το περισσότερο καθένας για τον εαυτό του, και σε κάθε γύρο κάνουν προσφορές ο ένας του άλλου με εναλλαγή μέχρι κάποιος να δεχτεί ή το ποσό να γίνει 0. Το αρχικό ποσό δηλαδή μειώνεται κάθε γύρο, με συνέπεια να υπάρχει η δυνατότητα να εξαντληθεί, και τότε κανείς από τους παίχτες να πιάσει μερίδιο του ποσού. Σε αυτήν την περίπτωση οι παίχτες είναι και οι 2 ηττημένοι. Επίσης υπάρχει το ενδεχόμενο να μοιραστούν το ποσό, όπου κάποιος θα πιάσει περισσότερο, ή τέλος να μοιραστούν το ποσό στην μέση με αποτέλεσμα να βγουν και οι 2 κερδισμένοι, πριν εξαντληθεί το ποσό.

Το κυπριακό ζήτημα, ή Κυπριακό πρόβλημα είναι ένα ζήτημα διεθνούς δικαίου που οφείλεται στην παράνομη στρατιωτική εισβολή το 1974 και κατοχή του βορείου τμήματος της Κυπριακής Δημοκρατίας από την Τουρκία. Συνέπειες αυτής της εισβολής που εξακολουθούν να υπάρχουν μέχρι και σήμερα είναι το ότι το ~37 % του Κυπριακού εδάφους (στην Βόρεια Κύπρο) το κατέχει η Τουρκία. Πιο συγκεκριμένα σε αυτό το 37 % της Κύπρου κατοικούν Τουρκοκύπριοι και στο υπόλοιπο 63% Ελληνοκύπριοι. Με αυτό το συμβάν οι 2 πλευρές , (ΕΚ ,ΤΚ) θέλουν να φτάσουν σε μία συνεργασία που θα επιφέρει οφέλη και στις 2 πλευρές. Η μεν Ελληνοκυπριακή πλευρά θέλει να πιάσει εδαφική έκταση από αυτήν που κατέχει η Τουρκοκυπριακή πλευρά , και η δε Τουρκοκυπριακή απαιτεί ποσοστά στην διακυβέρνηση του κυπριακού κράτους, τα

οποία κατέχει περισσότερα η Ελληνοκυπριακή πλευρά, για τον λόγο ότι δεν θεωρείται ανεξάρτητο κράτος το έδαφος που κατέχει.

Στην διπλωματική αυτή θα μοντελοποιηθεί το κυπριακό πρόβλημα, και πιο συγκεκριμένα η εδαφική πτυχή του, στο bargaining game [8], και μέσω κάποιων αλγορίθμων τις ενισχυτικής μάθησης, τον αλγόριθμο Q-learning και τον αλγόριθμο SARSA με e-greedy τακτική να δούμε αν οι πράκτορες μπορούν να φτάσουν σε συμβιβασμό σε πραγματικά και πιο περίπλοκα προβλήματα.

Κεφάλαιο 2

Γνωσιολογικό Υπόβαθρο

- 2.1 Στοιχεία Ενισχυτικής μάθησης
 - 2.2 Μέθοδοι Ενισχυτικής Μάθησης
 - 2.3 Q-Learning
 - 2.4 SARSA
 - 2.5 Θεωρία Παιγνίων
 - 2.6 Bargaining Game
 - 2.7 Επαναλαμβανόμενο Bargaining Game
 - 2.8 Κυπριακό Ζήτημα
 - 2.9 Προηγούμενη δουλειά πάνω στο θέμα
-

2.1 Στοιχεία Ενισχυτικής μάθησης

Ενισχυτική μάθηση[5-9] είναι η μάθηση όπου ο εκπαιδευόμενος μαθαίνει τι είναι σωστό και τι είναι λάθος μέσα από τις πράξεις του, έχοντας ένα εξωτερικό κριτή για να δίνει επιβράβευση, η τιμωρία για κάθε πράξη του με σκοπό να αυξήσει τη μέγιστη δυνατή αμοιβή για το στόχο του. Μπορεί να εκφραστεί και σαν «μάθηση μετά εμπειρίας» καθώς είναι η πιο βιολογική μορφή μάθησης που εφαρμόζεται από νεαρή ηλικία στους ανθρώπους.

Η ενισχυτική μάθηση αποτελείται από 3 συστατικά:

1. Πράκτορας

Ο εκπαιδευόμενος, η αυτός που παίρνει αποφάσεις, παραδείγματος χάρη ένα ρομπότ ή ένας παίκτης σε παιχνίδια.

2. Περιβάλλον

Τα γεγονότα έξω από το πράκτορα και τα μεταβάλλοντα δεδομένα και πόροι.

3. Δράσεις

Οι ενέργειες που αποφασίζει και εκτελεί ο πράκτορας από κάθε κατάσταση.

Όλα αυτά τα συστατικά συνεργάζονται έτσι ώστε να δημιουργηθεί ένα σύστημα όπου ο πράκτορας θα παίρνει κάποιες δράσεις από συγκεκριμένες καταστάσεις που οφείλονται στο περιβάλλον, με σκοπό να μεγιστοποιήσει τη αμοιβή για το στόχο του.



Σχήμα 2.1.1

Σχήμα μοντέλου ενισχυτικής μάθησης με συστατικά

Όπως φαίνεται και στο σχήμα 2.1.1 η λογική της ενισχυτικής μάθησης μεταξύ των συστατικών είναι ότι ο πράκτορας δεδομένου της κατάστασης που βρίσκεται ,την οποία λαμβάνει από το περιβάλλον, εκτελεί μία δράση. Αυτή η δράση στην συνέχεια επιφέρει κάποιες αλλαγές στο περιβάλλον, τις οποίες αλλαγές το περιβάλλον δίνει στον πράκτορα μαζί με ένα ενισχυτικό σήμα για να δει ο πράκτορας ποια δράση θα εκτελέσει από αυτήν την καινούργια κατάσταση, δεδομένης της τιμωρίας η επιβράβευσης που έχει πιάσει από την προηγούμενη δράση του. Το ενισχυτικό αυτό σήμα είναι η τιμωρία , ή επιβράβευση αναλόγως, της δράσεως του πράκτορα. Επίσης να σημειωθεί ότι αυτό το

σήμα δεν λέει στον πράκτορα δεν λέει ποια δράση θα ήταν καλύτερη για να εκτελέσει, αλλά αξιολογά αναλόγως με το αποτέλεσμα την δράση που έχει εκτελέσει.

Συνάρτηση αμοιβής είναι η συνάρτηση η οποία καθορίζει τί είναι καλό για την δεδομένη στιγμή, δηλαδή ποια δράση θα επιφέρει μεγαλύτερη εκτιμώμενη αμοιβή από την παρούσα κατάσταση που βρίσκεται ο πράκτορας, για παράδειγμα το να πιάσει ο παίχτης ένα πόνι του αντιπάλου σε ένα αγώνα σκακιού. Η συνάρτηση αμοιβής καθορίζεται από το περιβάλλον και το πόσο κοντά στον στόχο έχει επιφέρει η δράση τον πράκτορα

Για τον υπολογισμό της εκτίμησης ή την πρόβλεψη της αμοιβής που θα επιφέρει μια δράση από μία κατάσταση η μάθηση μπορεί να περιγραφεί σαν συσχετιζόμενη, η μη συσχετιζόμενη. Η συσχετιζόμενη μάθηση εκτός από το ενισχυτικό σήμα περιλαμβάνει και ερεθίσματα από το περιβάλλον, ενώ η μη-συσχετιζόμενη λαμβάνει μόνο το ενισχυτικό σήμα.

Ένα γνωστό χαρακτηριστικό της ενισχυτικής μάθησης είναι το δίλλημα της εξερεύνησης και της εκμετάλλευσης. Εξερεύνηση υπάρχει όταν ο πράκτορας δεν εκτελέσει κάποια από τις διαθέσιμες και ελεγμένες δράσεις από μια κατάσταση, και αντιθέτως δοκιμάσει μια καινούργια για να δει την αποτέλεσμα θα έχει και αν θα φέρει μεγαλύτερη αμοιβή. Εκμετάλλευση είναι το αντίθετο της εξερεύνησης, δηλαδή ο πράκτορας θα χρησιμοποιήσει την υπάρχουσα γνώση για μια κατάσταση και θα εκτελέσει μια δράση που έχει είδη εκτελεστεί, και για την οποία γνωρίζει τι εκτιμώμενο αποτέλεσμα θα επιφέρει . Ο τρόπος με τον οποίο ένας πράκτορας επιλέγει από την παρούσα κατάσταση μια δράση είναι καλύτερη για αυτόν, δηλαδή μια θα έχει μεγαλύτερη αμοιβή λέγεται τακτική η πολιτική.

Βάσης αυτού του διλλήματος που έχουμε αναφέρει υπάρχουν συγκεκριμένες τακτικές. Υπάρχει η άπληστη, η μερικώς- άπληστη και η μη-άπληστη τακτική. Στην άπληστη τακτική (greedy) , όπως λέει και το όνομα της, από μία κατάσταση ο πράκτορας λαμβάνει υπόψη μόνο την υπάρχουσα γνώση και επιλέγει την ενέργεια με την μεγαλύτερη εκτιμώμενη αξία για την συγκεκριμένη κατάσταση, άρα δεν εξερευνά ποτέ για νέες δράσεις που ίσως θα έφερναν καλύτερο αποτέλεσμα . Από την άλλη, η μη-άπληστη(non-greedy) τακτική , αρκετά συχνά εξερευνά καινούργιες καταστάσεις . Η τελευταία και ίσως θεωρητικά καλύτερη τακτική είναι η μερικώς άπληστη(e-greedy) , η

οποία συνδυάζει και τις 2 προηγούμενες τακτικές. Σε αυτήν την τακτική με μία μικρή πιθανότητα e ο πράκτορας εξερευνά για καινούργιες καταστάσεις, και με μια μεγαλύτερη πιθανότητα $1-e$, χρησιμοποιά την υπάρχουσα γνώση για επιλογή δράσης από την παρούσα κατάσταση.

Συνάρτηση αξίας είναι ένα μοντέλο για να υπολογίσουμε τι είναι καλό για το μέλλον, μακροπρόθεσμα. Με αυτό εννοούμε για παράδειγμα το τι τελικό αποτέλεσμα (νίκη ή ήττα) θα έχει μια συγκεκριμένη ενέργεια στην παρούσα κατάσταση σε ένα αγώνα σκακιού.

Όπως είναι προφανές και η συνάρτηση αμοιβής και η συνάρτηση αξίας πρέπει να ληφθούν υπόψη από ένα πράκτορα για να μεγιστοποιήσει το κέρδος και να έχει ένα καλύτερο αποτέλεσμα στο τέλος.

Για τον λόγο ότι δεν είναι εύκολο να υπολογίσουμε την συνάρτηση αξίας, χρησιμοποιούμε μια παράμετρο η οποία λέγεται συντελεστής έκπτωσης [10 11 12]. Η χρησιμοποίηση αυτής της παραμέτρου μας βοηθά να υπολογίσουμε πόσο θα λάβουμε υπόψη την τελική αμοιβή. Ένας συντελεστής έκπτωσης 0, θα κάνει τον πράκτορα «μυωπικό», λαμβάνοντας υπόψη μόνο τις τρέχουσες ανταμοιβές, δηλαδή για την χρονική στιγμή t , τον ενδιαφέρει μόνο να βελτιστοποιήσει την αμοιβή για το χρόνο $t+1$, ενώ ένας συντελεστής που προσεγγίζει το 1 αφορά πιο μακροπρόθεσμες υψηλές ανταμοιβές, δηλαδή ο πράκτορας βρισκόμενος στην χρονική στιγμή t θέλει να βελτιστοποιήσει την ανταμοιβή για την χρονική στιγμή $t+k$ όπου $k>1$. Ο ρυθμός έκπτωσης μπορεί να θεωρηθεί σαν το μοντέλο που λαμβάνει υπόψη τις καθυστερημένες αμοιβές και τις άμεσες. Η εξίσωση 2.1.1 χαρακτηρίζει τις μελλοντικές αμοιβές για την χρονική στιγμή t .

$$R(t) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^n \gamma^k r_{k+t+1}$$

Εξίσωση 2.1.1

Όπου $0 \leq \gamma \leq 1$ ρυθμός έκπτωσης και r_t αμοιβή για την χρονική στιγμή t και n η τελευταία χρονική στιγμή, ή το άπειρο.

Η συνάρτηση αξίας έχει 2 τύπους :

1) $V\pi(s)$: Η συνάρτηση αξίας για την κατάσταση s με σταθερή τακτική από την αρχή π .

2) $Q\pi(s,a)$: Η συνάρτηση αξίας από την κατάσταση s με δράση a και σταθερή πολιτική π .

Το ποια συνάρτηση αξίας θα χρησιμοποιήσουμε εξαρτάτε από το πιο αλγόριθμο ενισχυτικής μάθησης θα επιλέξουμε. Εμείς θα ασχοληθούμε με το Q-Learning και το SARSA με e-greedy τακτική.

2.2 Μέθοδοι ενισχυτικής Μάθησης

Η Ενισχυτική μάθηση, έχει 3 κύριους μεθόδους, τον Δυναμικό Προγραμματισμό (ΔΠ), την μέθοδο Monte Carlo και την μέθοδο Χρονικών Διαφορών (ΧΔ). Η κάθε μέθοδος έχει πλεονεκτήματα και μειονεκτήματα. Όλες οι μέθοδοι εφαρμόζουν αξιολόγηση πολιτικής, που είναι η εκτίμηση της Συνάρτησης Αξίας και βελτιστοποίηση τακτικής μέσω ενημέρωσης της Συνάρτησης Αξίας, και βελτιστοποίησης της τακτικής του πράκτορα.

Ο ΔΠ μπορεί να υπολογίσει τις βέλτιστες τακτικές, δηλαδή επιλογή ενέργειας. Η μέθοδος ΔΠ χρησιμοποιεί Συναρτήσεις Αξίας για να υπολογίσει τις βέλτιστες τακτικές. Για να το κάνει αυτό χρησιμοποιά εξισώσεις Bellman (1957) για την προσέγγιση επιθυμητών Συναρτήσεων Αξίας, μέσω 2 κυρίων μεθόδων, αξιολόγηση τακτικής και βελτίωση τακτικής. Η αξιολόγηση τακτικής είναι ο υπολογισμός της Συνάρτησης κατάστασης-αξίας ($V\pi(s)$), λαμβάνοντας υπόψη τις επόμενες εκτιμήσεις καταστάσεων, τις οποίες έχει όλες αποθηκευμένες, το οποίο όμως είναι υπολογιστικά ακριβό. Βελτίωση τακτικής είναι το πώς να βρεθεί μια τακτική καλύτερη από την προηγούμενη που θα επιφέρει μεγαλύτερο αναμενόμενο κέρδος. Η επανάληψη τακτικής συνδυάζει αξιολόγηση και βελτίωση τακτικής με την περικοπή των 2 ξεχωριστών διαδικασιών με σκοπό να γίνει υπολογιστικά φθηνότερη.

Το κύριο μειονέκτημα του ΔΠ είναι ότι για να επιτύχει βέλτιστες τακτικές χρειάζεται ένα πλήρες μοντέλο του περιβάλλοντος, το οποίο όμως δεν είναι πάντα διαθέσιμο. Επίσης όπως έχουμε αναφέρει είναι αρκετά υπολογιστικά ακριβός σε σύγκριση με τις άλλες 2 μεθόδους ενισχυτικής μάθησης.

Η μέθοδος Monte Carlo δεν απαιτεί πλήρη γνώση του περιβάλλοντος καθώς μαθαίνει μέσα από εμπειρία. Όπως και για τον ΔΠ, η μέθοδος Monte Carlo βασίζεται στην Συνάρτηση Αξίας. Υπολογίζει το μέσο όρο αποτελεσμάτων σε μία κατάσταση για να δώσει αξία στην τακτική, και το κάνει αυτό μέχρι ο μέσο όρος να συγκλίνει, δηλαδή να μην αλλάξει.

Η μέθοδος ΧΔ[10, 11, 12, 15] έχει πλεονεκτήματα των άλλων 2 μεθόδων καθώς δεν απαιτεί πλήρη γνώση περιβάλλοντος (όπως ΔΠ), άρα συμπεριφέρεται όπως η Monte Carlo, και βασίζει την εκτίμηση σε προηγούμενες μαθημένες εμπειρίες, όπως τον ΔΠ. Για την αξιολόγηση τακτικής χρησιμοποιούμε εμπειρία, όπως φαίνεται και στην εξίσωση (2.2).

$$V(S_t) = V(S_t) + a * [r_{t+1} + \gamma V(S_{t+1}) - V(S_t)] \quad (2.2)$$

Όπου $V(S_t)$ είναι η συνάρτηση αξίας για κατάσταση s την χρονική στιγμή t , a είναι μια παράμετρος η οποία ονομάζεται ρυθμός μάθησης (δηλαδή πόσο γρήγορα μαθαίνει ο πράκτορας), r_{t+1} η αμοιβή που λήφθηκε την χρονική στιγμή $t+1$, γ ο ρυθμός έκπτωσης και $V(S_{t+1})$ η συνάρτηση αξίας της κατάστασης s για την χρονική στιγμή $t+1$.

Η βάση όλων των μεθόδων των ΧΔ είναι το πώς να κατανέμουν αμοιβή σε ενέργειες οι οποίες παρήγαγαν την τελική αμοιβή (Πρόβλημα Χρονικής Απόδοσης Τιμών).

2.3 Q-learning

Ο αλγόριθμος Q-learning είναι μία μέθοδος των ΧΔ, ο οποίος είναι ένας αλγόριθμος εκτός –τακτικής (off-policy), που σημαίνει ότι χρησιμοποιά οποιαδήποτε τακτική προκειμένου να υπολογίσει την καταλληλότερη συνάρτηση δράσης-αξίας ($Q(s,a)$).

Η εξίσωση που χαρακτηρίζει τον Q-learning είναι η :

$$Q(S_t, a_t) = Q(S_t, a_t) + \alpha * (r_{t+1} + \gamma * \max [Q(S_{t+1}, a)] - Q(S_t, a_t))$$

Όπου α είναι ο ρυθμός μάθησης, γ ο ρυθμός έκπτωσης $Q(S_t, a_t)$ η συνάρτηση δράση κατάστασης για την χρονική στιγμή t , r_{t+1} είναι η ανταμοιβή της εκτέλεσης της a_t από την κατάσταση s_t , όπου s_t η κατάσταση την χρονική στιγμή t και a_t η δράση που εκτελείτε την χρονική στιγμή t . Τέλος, $\max [Q(S_{t+1}, a)]$ είναι η βέλτιστη συνάρτηση δράσης-αξίας της επόμενης κατάστασης (χρονική στιγμή $t+1$).

Δηλαδή ο αλγόριθμος Q-learning από κάθε κατάσταση, εκτελεί την δράση η οποία θα έχει το μεγαλύτερη συνάρτηση προεξοφλημένη αμοιβή για την επόμενη κατάσταση. Αυτό χαρακτηρίζεται σαν άπληστη στρατηγική.

Ο αλγόριθμος του Q-learning είναι ο εξής :

Αρχικοποίησε όλα τα $Q(s,a)$ αυθαιρέτως

Ξεκίνα από την κατάσταση s

Επέλεξε μια δράση a από την κατάσταση s

Επανάλαβε για κάθε βήμα του επεισοδίου {

Πιάσε μια δράση a , παρατήρα ανταμοιβή r και επόμενη κατάσταση s'

Επέλεξε δράση a' από την επόμενη κατάσταση s'

$$Q(s,a) = Q(s,a) + \alpha * (r + \gamma * \max [Q(s', a')] - Q(s,a))$$

$$s = s';$$

μέχρι s να είναι τελικό }

2.4 SARSA

Ο αλγόριθμος SARSA (STATE-ACTION-REWARD-STATE-ACTION) είναι μία μέθοδος των ΧΔ, ο οποίος είναι ένας αλγόριθμος εντός-τακτικής (on-policy), που σημαίνει ότι χρησιμοποιά την ίδια τακτική προκειμένου να υπολογίσει την καταλληλότερη συνάρτηση δράσης-αξίας ($Q(s,a)$).

Η εξίσωση που χαρακτηρίζει τον SARSA είναι η :

$$Q(S_t, a_t) = Q(S_t, a_t) + \alpha * (r_t + \gamma * Q(S_{t+1}, a_{t+1}) - Q(S_t, a_t))$$

Όπου α είναι ο ρυθμός μάθησης, γ ο ρυθμός έκπτωσης $Q(S_t, a_t)$ η συνάρτηση δράσης κατάστασης για την χρονική στιγμή t , r_t είναι η ανταμοιβή της εκτέλεσης της a_t από την κατάσταση s_t , όπου s_t η κατάσταση την χρονική στιγμή t και a_t η δράση που εκτελείτε την χρονική στιγμή t . Τέλος, $Q(S_{t+1}, a_{t+1})$ είναι η βέλτιστη συνάρτηση δράσης-αξίας της επόμενης κατάστασης (χρονική στιγμή $t+1$).

Ο αλγόριθμος του SARSA είναι ο εξής :

Αρχικοποίησε όλα τα $Q(s, a)$ αυθαιρέτως

Ξεκίνα από κατάσταση s

Επέλεξε μια δράση a από την κατάσταση s

Επανάλαβε για κάθε βήμα του επεισοδίου {

Πιάσε μια δράση a , παρατήρα ανταμοιβή r και επόμενη κατάσταση s'

Επέλεξε δράση a' από την επόμενη κατάσταση s'

$$Q(s, a) = Q(s, a) + \alpha * (r + \gamma * Q(s', a') - Q(s, a))$$

$$s = s'; a = a';$$

μέχρι s να είναι τελικό}

2.5 Θεωρία Παιγνίων

Για να μπορέσουμε πρώτα να καταλάβουμε τι είναι το bargaining game πρέπει να δούμε λίγο τι είναι τα παίγνια .

- Εισαγωγή

Θεωρία παιγνίων[2] είναι η επιστήμη όπου επικεντρώνεται στο πώς δύο η περισσότεροι παίκτες παίρνουν αποφάσεις σε διάφορες καταστάσεις με απώτερο σκοπό το μέγιστο κέρδος τους, εις βάρος των άλλων παικτών ή σε συνεργασία με τους άλλους. Συνήθως ο κάθε παίκτης έχει μια δική του στρατηγική στην λήψη απόφασης, και το κέρδος του δεν εξαρτάται μόνο από την επιλογή του αλλά και από τις επιλογές των άλλων παικτών. Έτσι δημιουργείται το παίγνιο όπου ο κάθε παίκτης συγκρούεται με τους άλλους οι οποίοι θεωρούνται αντίπαλοι με θεωρητικό νικητή αυτό με το περισσότερο κέρδος .

- Βασικά στοιχεία και Εφαρμογές

Ο όρος παίγνιο-παιγνίδι παραπέμπει σε ένα διαγωνισμό όπου συνήθως έχουμε ένα νικητή και ένα ηττημένο παραδείγματος χάρη στο σκάκι, τάβλι. Σήμερα η θεωρία παιγνίων βρίσκει παραπάνω χρήση στην οικονομία όπου η κάθε απόφαση με μια συγκεκριμένη κατάσταση πρέπει να επιφέρει το μέγιστο κέρδος. Βασικά το παίγνιο σε αυτούς τους τομείς μπορεί να θεωρηθεί σαν ένα μαθηματικό μοντέλο όπου με τύπους επιφέρεται η μέγιστη αμοιβή με βάση κάποιες συνθήκες.

Σε ένα παίγνιο έχουμε τους παίκτες-πράκτορες οι οποίοι είναι αυτόνομες μονάδες που λαμβάνουν απόφαση βάση της στρατηγικής που ακολουθούν και δεδομένων κάποιων συναρτήσεων αμοιβών και τιμωριών , συνήθως προσαρμόζουν την στρατηγική τους βάση αυτών των σημάτων που λαμβάνουν για κάθε απόφαση τους.

Στρατηγική είναι το σύνολο των κανόνων που ορίζει την κατάλληλη επιλογή από ένα σύνολο εφικτών επιλογών σε μια κατάσταση μέχρι το τέλος του παιγνιδιού, με απώτερο σκοπό να βελτιστοποιηθεί το κέρδος του παίκτη. Οι στρατηγικές μπορούν να είναι μικτές, δηλαδή με μία παράμετρο πιθανότητας ο παίκτης συνδυάζει διάφορες στρατηγικές, ή αμιγής όπου ο κάθε παίκτης μένει σε μία στρατηγική από την αρχή μέχρι το τέλος του παιγνιδιού.

Σε κάθε παίγνιο υπάρχουν κάποιοι κανόνες οι οποίοι ορίζουν ποιες επιλογές είναι εφικτές σε κάθε κατάσταση για τους παίκτες.

- Κατηγοριοποίηση

Τα παίγνια χωρίζονται σε κάποιες κατηγορίες αναλόγως με τους τον αριθμό παικτών που έχουν, την σχέση που έχουν μεταξύ τους οι παίκτες, το εάν μια πράξη επηρεάζει μελλοντικές αποφάσεις και κάποια από την δομή του παιγνιδιού.

Οι κατηγορίες των παιγνίων που δημιουργούνται όσο αφορά τον αριθμό των παικτών είναι αυτά που έχουν 2 παίκτες και αυτά που έχουν πιο πολλούς από 2, καθώς είναι ευνόητο ότι με ένα παίκτη δεν μπορεί να υπάρχει ούτε ανταγωνισμός, ούτε συνεργασία συνεπώς ούτε παιγνίδι! Καλό είναι να αναφέρουμε ότι συνήθως τα παίγνια 2 παικτών είναι ανταγωνιστικής φύσης. Βεβαίως όπως θα δούμε και με το δικό μας παίγνιο αυτό δεν είναι απόλυτο! Ενώ στα παίγνια με περισσότερους από 2 παίκτες υπάρχουν πιο πολλές συνεργασίες για τον λόγο ότι κάποιοι παίκτες ίσως έχουν κοινό όφελος, ή απλά κοινή στρατηγική.

Εάν η κατηγοριοποίηση των παιγνίων γίνει με βάση την σχέση που έχουν οι παίκτες μεταξύ τους μπορούν να χωριστούν σε ομαδικά και μη ομαδικά. Ομαδικά είναι τα παίγνια όπου εξαρχής κάποιοι παίκτες είναι μαζί με το ίδιο σκοπό και παίζουν εντελώς καθαρά για το καλό της ομάδας τους καθώς τα μη ομαδικά είναι αυτά όπου εάν και υπάρχει η πιθανότητα κάποιοι παίκτες να συνεργαστούν, δεν αποτελούν ομάδα, καθώς ο καθένας έχει σαν στόχο τον δικό του συμφέρον και αποτελεί αυτοτελή μονάδα.

Υπάρχει ακόμη ένας παράγοντας ο οποίος μπορεί να διαχωρίσει τα παίγνια σε κατηγορίες, και είναι αυτός που έχουμε αναφέρει πιο πριν, η δομή. Η δομή είναι ένας πιο περίπλοκος παράγοντας καθώς είναι κάπως αφηρημένος και μπορεί να συνδυάσει όλες της παραπάνω κατηγοριοποιήσεις. Έτσι έχουμε τα παιγνίδια μηδενικού αθροίσματος, όπου το κέρδος ενός παίκτη είναι απώλεια του άλλου, δηλαδή το άθροισμα των κερδών όλων των παικτών είναι 0, ή αν το θέσουμε αλλιώς ο αρχικός πόρος δεν μειώνεται ούτε αυξάνεται και πάντα έχουμε ένα νικητή και ένα ηττημένο. Σε αντίθεση υπάρχουν τα παίγνια μη μηδενικού, ή γενικού αθροίσματος όπου είναι τα παίγνια όπου ο αρχικός πόρος μεταβάλλεται κατά την διάρκεια του παιγνιδιού και επίσης στο τέλος του παιγνιδιού δεν υπάρχει απαραίτητως ένας νικητής και ένας ηττημένος καθώς υπάρχει η δυνατότητα συμβιβασμού και διαχωροποίησης του πόρου έτσι ώστε να βγαίνουν και οι 2 παίκτες κερδοφόροι ή και η δυνατότητα να ζημιωθούν και οι 2 παίκτες χωρίς να βγει κερδισμένος κάποιος από τους 2.

2.6 Bargaining Game

Το bargaining game[8], το οποίο θα ασχοληθούμε σε αυτή την διπλωματική είναι καταρχάς ένα παίγνιο γενικού αθροίσματος . Δηλαδή όπως έχουμε αναφέρει πιο πάνω δεν σημαίνει αναγκαίως ότι θα βγει ένας, η κάποιος νικητής . Πιο συγκεκριμένα σε αυτό το παιχνίδι οι παίκτες λόγω της φύσης του παιχνιδιού θα έχουν καλύτερο κέρδος αν συμβιβαστούν διότι υπάρχει κίνδυνος να χάσουν και οι 2 αν δεν συνεργαστούν.

Μια κλασική επεξήγηση αυτού του είδους παιχνιδιών είναι το εξής παράδειγμα:

Έστω έχουμε 2 παίκτες και ένα αρχικό ποσό x_{start} . Σκοπός αυτού του παιχνιδιού είναι να μοιράσουν οι παίκτες αυτό το ποσό, δηλαδή αν ο παίκτης A πιάσει z μονάδες από το αρχικό x_{start} ποσό, τότε ο παίκτης B θα πιάσει $x_{start} - z$. Η δυσκολία και τροποποίηση όμως αυτού του παιχνιδιού είναι ότι το ποσό μειώνεται σταδιακά και υπάρχει περίπτωση να φτάσει το 0, άρα κανείς παίκτης δεν θα έχει κέρδος! Έτσι η λογική αυτού του παιχνιδιού είναι ότι, κάθε γύρο οι παίκτες κάνουν προσφορές μεταξύ τους μέχρι να φτάσουν σε συμφωνία ή να λείψει το ποσό. Έστω δηλαδή ο παίκτης A κάνει μια προσφορά στον παίκτη B, όπου αυτή η προσφορά αντιπροσωπεύει το πόσες μονάδες από το ποσό προσφέρει ο A στον παίκτη B, ο παίκτης B έχει επιλογή ή να δεχτεί αυτή την προσφορά, ή να την απορρίψει και να κάνει μια δική του αντιπροσφορά στον παίκτη A. Αν ο παίκτης B αποδεχτεί, τότε παίρνει αυτές τις μονάδες ποσού που του προσφέρει ο παίκτης A και ο παίκτης A που έκανε την προσφορά παίρνει το υπόλοιπο του ποσού αυτού του γύρου. Αν ο παίκτης B δεν αποδεχτεί την προσφορά, θα κάνει αυτός μια αντιπροσφορά στον παίκτη A. Αν έχουμε 2 επαναλαμβανόμενες απορρίψεις προσφορών (ο παίκτης A στον B και ο παίκτης B στον A αντίστοιχα) τότε το ποσό μειώνεται κατά μία μονάδα, από το ποσό του προηγούμενου γύρου άρα θα έχουμε $x_{i+1} = x_i - 1$ (όπου i παρών γύρος). Αυτή η μονάδα που χάθηκε από το ποσό, τώρα δεν είναι διαθέσιμη για μοίρασμα επομένως κανείς δεν την διεκδικά πλέον. Με αυτή την λογική αν φτάσουμε στον γύρο όπου το ποσό έχει γίνει 0 (που σημαίνει ότι κανείς από τους 2 παίκτες δεν είχε αποδεχτεί προσφορά του άλλου) τότε ο κάθε παίκτης δεν παίρνει τίποτα από το αρχικό ποσό, με αποτέλεσμα να βγαίνουν και οι 2 ηττημένοι. Βεβαίως αν ένας παίκτης αποδεχτεί προσφορά που κάνει ο άλλος παίκτης πριν το ποσό μηδενιστεί και αυτή η προσφορά είναι πιο μεγάλη από το ποσό που θα πιάσει ο άλλος παίκτης τότε θεωρητικά νικά αυτός με το πιο μεγάλο ποσό.

Όμως επειδή στην φύση των ανθρώπων είναι το να θέλουν να είναι πιο κερδισμένοι από τον αντίπαλο δεν συνηθίζουν να κάνουν προσφορά που να τους επιφέρει λιγότερο κέρδος από τον αντίπαλο.

2.7 Επαναλαμβανόμενο Bargaining Game

Το επαναλαμβανόμενο bargaining game είναι το ίδιο με το παιχνίδι που ανέφερα πιο πάνω με την μόνη διαφορά ότι το παιχνίδι θα παίζεται για ένα αριθμό επαναλήψεων. Δηλαδή την πρώτη φορά που θα φτάσει το ποσό στο 0 ή μοιράσουν το ποσό, τότε θα ξαναπαιχτεί το παιχνίδι από την αρχή με το αρχικό ποσό.

Σκοπός αυτής της επανάληψης είναι οι παίκτες να μάθουν τι κέρδος τους είχε επιφέρει απόφαση σε προηγούμενα παιχνίδια για να βελτιώσουν τις αποφάσεις τους και να κερδίσουν πιο μεγάλο ποσό. Βέλτιστη θεωρητικά λύση για αυτό το παιχνίδι, για το κοινό καλό είναι οι 2 παίκτες να μοιράσουν το ποσό στην μέση το πρώτο γύρο και να πιάσουν από μισά. Ειδικά αν προχωρήσουν το ποσό θα μειώνεται και επειδή ο καθένας θέλει να πιάσει παραπάνω από τον άλλο δεν θα συμβιβαστούν και θα χάσουν και οι 2 με κέρδος 0, αφού το ποσό θα έχει λείψει.

2.8 Κυπριακό Ζήτημα

Το Κυπριακό ζήτημα ή Κυπριακό πρόβλημα και συνηθέστερα, απλώς «Κυπριακό», είναι ζήτημα διεθνούς δικαίου που οφείλεται στην παράνομη στρατιωτική εισβολή το 1974 και κατοχή του βορείου τμήματος της Κυπριακής Δημοκρατίας από την Τουρκία. Συνέπειες αυτής της εισβολής που εξακολουθούν να υπάρχουν μέχρι και σήμερα είναι το ότι το ~37 % του Κυπριακού εδάφους (στην Βόρεια Κύπρο το κατέχει η Τουρκία). Πιο συγκεκριμένα σε αυτό το 37 % κατοικούν Τουρκοκύπριοι και στο υπόλοιπο 63% Ελληνοκύπριοι. Με αυτό το συμβάν οι ΤΚ που κατοικούσαν στην Κύπρο αυξηθήκαν με αποτέλεσμα να θέλουν περισσότερες απαιτήσεις. Η εδαφική

περιοχή που κατέχουν οι TK δεν θεωρείτε ανεξάρτητο κράτος και την διακυβέρνηση του κράτους κατέχει σε περισσότερο βαθμό η ΕΚ πλευρά. Έτσι έχουμε σαν συνέπεια η ΕΚ πλευρά να απαιτεί έδαφος από την γη που κατέχει η TK πλευρά, και η TK πλευρά με την σειρά της να απαιτεί ποσοστά στην διακυβέρνηση του κράτους. Καταλήγουμε έτσι στις 2 πλευρές, (ΕΚ, TK) οι οποίες θέλουν να φτάσουν σε μια κοινή λύση, να μοιράσουν αυτό το ποσοστό του 37 % έτσι ώστε το μείρασμα αυτός, σε συνδυασμό και άλλων παραμέτρων όπως ποσοστά διακυβέρνησης, να ικανοποιά και τις 2 πλευρές. Αξίζει να αναφερθεί ότι στην πιο πρόσφατη προσπάθεια λύσης στην Γενεύη (2017) η TK πλευρά δεν δέχεται να μείνει με έδαφος μικρότερο του 29.2 % της Κύπρου καθώς η ΕΚ πλευρά διατίθεται να πιάσει έδαφος που ανέρχεται στις 8.8 «μονάδες» από τις 37 που κατέχει η TK, αφήνοντας την TK με έδαφος 28.2 % της Κύπρου. Με βάση αυτά τα δεδομένα και γεγονότα θα επιχειρήσουμε να μοντελοποιήσουμε το Κυπριακό στο Bargaining game. Οι 2 πράκτορες όπως είναι φανερό είναι η TK πλευρά και η ΕΚ πλευρά. Το αρχικό ποσό που θέλουν να μοιράσουν είναι το ποσό της κατεχόμενης Κύπρου, που ανέρχεται περίπου στο 37 % της Κύπρου το οποίο κατέχει η TK πλευρά. Η κάθε πλευρά θα κάνει με τη σειρά της μια προσφορά για κάθε γύρο. Η TK πλευρά θα κάνει προσφορά για παραχώρηση εδάφους και η ΕΚ πλευρά θα κάνει προσφορά από τα ποσοστά διακυβέρνησης, τα οποία θα φαίνονται έμμεσα στο σύστημα μέσω τιμωριών, (θα εξηγηθεί στο κεφάλαιο 4) απαιτώντας έδαφος από την TK πλευρά. Οι προσφορές και απαιτήσεις θα αντιπροσωπεύουν ποσοστό από το 37 % της Κύπρου, δηλαδή την κατεχόμενη Κύπρο και η άλλη πλευρά αναλόγως των απαιτήσεων της θα δεχτεί, ή θα απορρίψει και θα κάνει αυτή μια δική της αντιπροσφορά. Με το να απορρίπτουν τις προσφορές και να μην φτάνουν σε συμβιβασμό, δηλαδή το ποσό να μειώνεται μέχρι να γίνει 0, θα έχουμε και τις 2 πλευρές ηττημένες, διότι με το να έχει γίνει 0 το ποσό, η ΕΚ πλευρά δεν θα έχει πιάσει έδαφος από την TK πλευρά, που έχει σαν συνέπεια η TK πλευρά να μην έχει πιάσει ποσοστό από την διακυβέρνηση από την ΕΚ πλευρά και έτσι να μείνουν και οι 2 πλευρές ανικανοποίητες και στάσιμες. Σκοπός μας είναι να δούμε ότι οι πράκτορες μαθαίνουν μέσα από τις επαναλήψεις, ότι ο συμβιβασμός είναι η καλύτερη κοινή λύση, και να φτάσουν σε ένα θεωρητικό συμβιβασμό (ο οποίος θα λαμβάνει υπόψη τα δεδομένα) που η εκπλήρωση αυτού του στόχου θα σημαίνει ότι όντως μπορεί να υπάρξει μια λύση στο Κυπριακό, και ότι η ενισχυτική μάθηση μπορεί να μοντελοποιηθεί και να εφαρμοστεί σε ζητήματα και προβλήματα ρεαλιστικά που υπάρχουν στην καθημερινότητα μας.

2.9 Προηγούμενη δουλεία πάνω στο θέμα

Όσο αφορά την ενισχυτική μάθηση και παίγνια γενικού αθροίσματος και εφαρμογές πάνω σε σύγχρονα και καθημερινά προβλήματα υπάρχουν κάποιες προηγούμενες εργασίες.

Υπάρχουν αρκετές εργασίες που να μοντελοποιούν το Κυπριακό ζήτημα σαν ένα παίγνιο γενικού αθροίσματος. Όλες αφορούσαν το παίγνιο Επαναλαμβανόμενο Δίλλημα του Φυλακισμένου(ΕΔΦ)[1].

Ο Lumsden, M. (1973) [7] είχε μοντελοποιήσει το κυπριακό στο παίγνιο ΕΔΦ. Είχε αξιολογήσει το κυπριακό κάτω από 4 ενδεχόμενα [Πόλεμος, Ειρήνη, Διχοτόμηση, Ένωση] όπου είχε σαν καλύτερη λύση την ειρήνη(συμβιβασμός) και χειρότερη τον πόλεμο.

Οι Yesilada και Sozen (2002) [16] ασχολήθηκε με το αν είναι η ένταξη της Κύπρου στην Ευρωπαϊκή Ένωση καλό ή κακό όσο αφορά λύση στο πρόβλημα, με το παίγνιο ΕΔΦ.

Ο Βασιλειάδης Β. (2010) [13] ανέλυσε κατά πόσο μπορεί να υπάρξει συνεργασία μεταξύ 2 πρακτόρων οι οποίοι εκπαιδεύονται με Ενισχυτική Μάθηση σε περιβάλλον ενός παίγνιου γενικού-αθροίσματος, συγκεκριμένα το ΕΔΦ.

Πιο πρόσφατα, ο Καραολής Γ (2010)[6] είχε ξαναμοντελοποιήσει το κυπριακό στο ίδιο παίγνιο ΕΔΦ. Ο Καραολής επιχείρησε σε κάθε φάση του κυπριακού προβλήματος να συσχετίζει τις αμοιβές για κάθε ενέργεια των 2 πλευρών και έφτασε στο συμπέρασμα ότι όντως μπορεί να επιτευχθεί μία λύση, υπό κάποιες προϋποθέσεις πάντα .

Οι Βασιλειάδης Β. , Χριστοδούλου Χ. , Κλεάνθους Α. (2011)[12] ερευνούν την ενισχυτική μάθηση στο «Επαναλαμβανόμενο Δίλλημα του φυλακισμένου» με spiking και non-spiking πράκτορες για να εξερευνήσουν τις καταστάσεις που χρειάζονται για να ενισχύσουν το αποτέλεσμα συμβιβασμού .

Μια διαφορετική χρήση του παίγνιου ΕΔΦ μαζί με την ενισχυτική μάθηση επιχείρησε να κάνει η Ευσέβεια Κ. (2014) [3]. Η Κυριάκου μελέτησε κατά πόσο μπορεί να υπάρξει συμβιβασμός σε μία «κούρσα εξοπλισμού» μεταξύ Ελλάδας και Τουρκίας. Κούρσα εξοπλισμού ονομάζεται το φαινόμενο όπου 2 πράκτορες (κυρίως χώρες) έχουν έξοδα πάνω σε στρατιωτικό οπλισμό , που είναι αρκετά δαπανηρά και προσπαθούμε να μειώσουμε αυτά τα έξοδα και να συμβιβαστούν οι 2 χώρες για να τα ξοδέψουν πιο έξυπνα. Η Ευσέβεια έφτασε στο συμπέρασμα ότι μπορεί να υπάρξει συμβιβασμός έχοντας μεγάλο συντελεστή έκπτωσης και χρησιμοποιώντας την e-greedy τακτική.

Η Γεωργίου Α. (2015) [4] χρησιμοποίησε και αυτή το ΕΔΦ μαζί με την ενισχυτική μάθηση σε ένα πολύ διαφορετικό πρόβλημα, το οποίο βέβαια έχει σαν βάση την σύγκρουση μεταξύ 2 «πρακτόρων». Η Γεωργίου μοντελοποίησε το ΕΔΦ στην σύγκρουση του άνω με του κάτω μέρους του ανθρώπινου εγκεφάλου με σκοπό να συμβιβαστούν, αναπτύσσοντας έτσι την έννοια του αυτοελέγχου! Ο αλγόριθμος ενισχυτικής μάθησης που χρησιμοποιήθηκε ήταν ο SARSA (State-Action-Reward-State-Action), η οποία βασίζεται στη μέθοδο χρονικών διαφορών (Temporal Difference) όπου η επιλογή επόμενης κίνησης είχε την e-greedy τακτική.

Κεφάλαιο 3

Περιγραφή συστήματος

3.1 Εισαγωγή

3.2 Σχεδιασμός

3.3 Υλοποίηση

3.1 Εισαγωγή

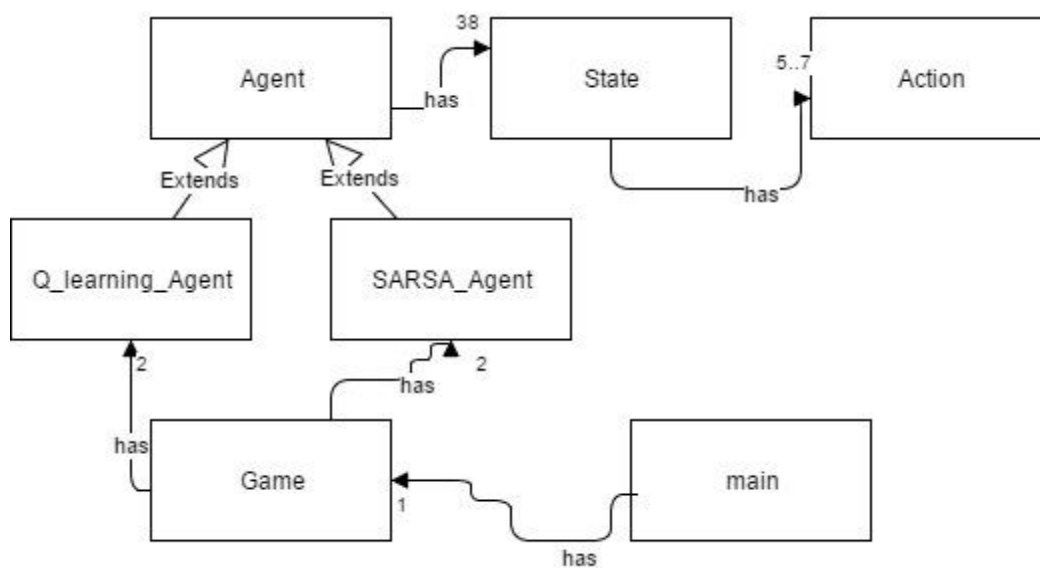
Σε αυτό το κεφάλαιο θα εξηγήσουμε την υλοποίηση και το σχεδιασμό του προγράμματος, χωρίς την αντιστοίχιση στο κυπριακό πρόβλημα.

Δηλαδή θα εξηγήσω γενικά τις κλάσεις, τις σχέσεις μεταξύ των αντικειμένων και των κλάσεων και το τι κάνει η κάθε κλάση.

Στα πλαίσια αυτής της διπλωματικής δημιούργησα ένα σύστημα στο οποίο οι πράκτορες μαθαίνουν να συμβιβάζονται μέσω ενισχυτικής μάθησης παίζοντας το Bargaining Game. Οι 2 αλγόριθμοι της ενισχυτικής μάθησης με τους οποίους μαθαίνουν οι πράκτορες είναι Q-learning και SARSA με e-greedy τακτική. Το πρόγραμμα που υλοποιήθηκε μπορεί να επεκταθεί με περισσότερους αλγορίθμους για μάθηση των παικτών πολύ εύκολα, απλά προσθέτοντας τον κώδικα για τον αλγόριθμο και δημιουργώντας πράκτορες που να ενεργούν με αυτούς τους αλγορίθμους.

3.1 Σχεδιασμός

Ο σχεδιασμός μου είναι αντικειμενοστρεφής, δηλαδή μέσα στο πρόγραμμα υπάρχουν πολλά αντικείμενα από διάφορες κλάσεις, τα οποία έχουν σχέση ή εξάρτηση από άλλα αντικείμενα. Στο παρακάτω σχήμα θα δείξω τις κλάσεις καθώς και την αλληλεπίδραση που έχουν μεταξύ τους και πόσα αντικείμενα περιλαμβάνει η κάθε 1 από άλλες κλάσεις.



Σχήμα 3.1

Στο σχήμα αυτό φαίνονται οι σχέσεις μεταξύ των κλάσεων. Βλέπουμε ότι οι κλάσεις **Q_learning_Agent** και **SARSA_Agent** κληρονομούν από την κλάση **Agent**, και πόσα αντικείμενα κάθε κλάση έχει από κάποια άλλη. Για παράδειγμα η κλάση **main** έχει 1 αντικείμενο της κλάσης **Game**.

Όπως φαίνεται και στο σχήμα 3.1 το πρόγραμμα αποτελείται από τις κλάσεις :**main**, **Agent**, **Q_learning_Agent**, **SARSA_Agent**, **State**, **Action**, **Game**. Η κύρια κλάση του προγράμματος είναι η κλάση **main**. Σε αυτήν την κλάση έχουμε 1 αντικείμενο της κλάσης **Game**, δηλαδή ένα μοντέλο παιχνιδιού. Στη κλάση **Game** υπάρχουν αντίστοιχα 2 αντικείμενα της κλάσης **SARSA_Agent** και 2 αντικείμενα της κλάσης

Q_learning_Agent. Με άλλα λόγια υπάρχουν 2 πράκτορες που χρησιμοποιούν αλγόριθμο SARSA και 2 που χρησιμοποιούν αλγόριθμο Q-learning . Ο κάθε πράκτορας , οποιουδήποτε αλγορίθμου κληρονομεί τα χαρακτηριστικά που έχει η κλάση Agent, που περιλαμβάνει όλα τα χαρακτηριστικά που έχουν κοινά οι πράκτορες ανεξαρτήτως αλγορίθμου, όπως για παράδειγμα ρυθμό μάθησης, πίνακα με καταστάσεις. Η κλάση Agent περιέχει ένα πίνακα από 37 states που το κάθε state αντιστοιχεί στη προσφορά που θα δεχτεί ο κάθε πράκτορας . Αυτή η προσφορά όπως είναι λογικό θα είναι από 0-37 που είναι το αρχικό ποσό. Τέλος κάθε state, περιλαμβάνει ένα αριθμό από actions που μπορούν να πραγματοποιηθούν αναλόγως από αυτό. Αυτά τα actions μπορούν να είναι από 5 μέχρι 7 αναλόγως από το state.

3.3 Υλοποίηση

Έχω χρησιμοποιήσει την γλώσσα προγραμματισμού java, η οποία είναι αρκετά αποδοτική, εύκολη και κυρίως αντικειμενοστρεφής. Επίσης μπορεί να τρέξει σε πολλά λειτουργικά συστήματα, όπως Windows, Linux, Unix, Macintosh.

Σε αυτό το πρόγραμμα υπάρχουν 2 πράκτορες που μαθαίνουν χρησιμοποιώντας τον αλγόριθμο Q-learning της ενισχυτικής μάθησης και 2 που μαθαίνουν χρησιμοποιώντας τον αλγόριθμο SARSA με e-greedy τακτική . Όπως ανέφερα και πιο πάνω το πρόγραμμα μπορεί να επεκταθεί εύκολα και με άλλους αλγορίθμους, καθώς θα πρέπει να προστεθεί ο αλγόριθμος και κάποια χαρακτηριστικά που διαφέρουν από τα χαρακτηριστικά που έχουν οι περισσότεροι αλγόριθμοι ενισχυτικής μάθησης, τα οποία βρίσκονται στην κλάση πατέρα Agent. Τα πιθανά ενδεχόμενα στο τέλος ενός γύρου, όπου γύρο θεωρούμε 2 συνεχόμενες απορρίψεις προσφοράς και από τους 2 παίκτες, δηλαδή αν έστω ο παίκτης a κάνει προσφορά στο παίκτη b και ο παίκτης b την απορρίψει , και κάνοντας αντιπροσφορά στον παίκτη a , ο παίκτης a απορρίψει και αυτός με την σειρά του την αντιπροσφορά από το παίκτη b, τότε θεωρούμε ότι έχει περάσει ένας γύρος και ότι το ποσό έχει μειωθεί κατά 1 μονάδα.

Πριν να εξηγήσουμε τι κάνει κάθε κλάση να δούμε ένα γενικό ψευδοκώδικα του συστήματος ο οποίος θα μας επιδείξει την λογική του προγράμματος όσο αφορά 2 πράκτορες που μαθαίνουν μέσω της ενισχυτικής μάθησης να μάθουν να συμβιβάζονται

μέσα από κάποιες επαναλήψεις παίζοντας το Bargaining game. Στον παρακάτω ψευδοκώδικα δεν περιλαμβάνεται ακόμη η αντιστοίχιση του Bargaining game με το κυπριακό . [Αυτό θα περιλαμβάνεται στο επόμενο κεφάλαιο]

1)Simvivasmoi=0, Oxi_simvivasmoi=0, Nikes_praktoraA=0, Nikes_praktoraB=0;

2)Για κάθε επανάληψη i για το παιχνίδι {

3)Αρχικοποιά τους 2 πράκτορες με τις επιθυμητές παραμέτρους και το επιθυμητό αλγόριθμο.

4)Θέσε αρχικό ποσό χ

5)Όσο $\chi > 0$ {

6)Επέλεξε ένα πράκτορα τυχαία, έστω a

7)Ο πράκτορας a κάνει προσφορά offer στο άλλο πράκτορα, έστω b

8)Κάνε update τον πίνακα με τις συναρτήσεις αξίας για δράση

9)Αν η προσφορά είναι προσφορά συμβιβασμού {

10)Αν ο πράκτορας b δεχτεί την προσφορά {

11)Simvivasmoi=simvivasmoi+1;

12) Επιβράβευση και για τους 2 πράκτορες

13)Πήγαινε στο βήμα [45] }

14)Ειδάλλως {

15)Αν ο b δεχτεί την προσφορά offer {

16) Αν $offer > x - offer$ {

17)Nikes_praktoraB=nikes_praktoraB+1;

18)Τιμωρία για πράκτορα a, επιβράβευση για πράκτορα b}

19)Ειδήλλως{

20)Nikes_praktoraA=Nikes_praktoraA+1;

21)Τιμωρία για πράκτορα b, επιβράβευση
για πράκτορα a}

22) Πήγαινε στο βήμα [45] }

23)Ειδήλλως{

24)Ο πράκτορας b κάνει προσφορά offer2 στο
άλλο πράκτορα, έστω a

25)κάνε update πίνακα με συναρτήσεις αξιών
πράκτορα b

26)Αν η προσφορά είναι προσφορά
συμβιβασμού{

27)Αν ο πράκτορας a δεχτεί την
προσφορά{

28)Simvivasmoi=simvivasmoi+1;

29)Ανταμοιβή για πράκτορες a και
b

30)Πήγαινε στο βήμα [45] }}

31)Ειδήλλως{

32)Αν ο a δεχτεί την προσφορά offer2 {

33) Αν offer2>x-offer2 {

34)Nikes_praktoraA=nikes_praktoraA+1;

35) Τιμωρία για πράκτορα
b, επιβράβευση για
πράκτορα a}

36) Ειδάλλως {

37) $Nikes_praktoraB = Nikes_praktoraB + 1;$

38) Τιμωρία για πράκτορα
a, επιβράβευση για
πράκτορα b}

39) Πήγαινε στο βήμα [45]}

40) Ειδάλλως {

41) $x = x - 1;$

42) Πήγαινε στο βήμα [45] } }

} }

}

43) $Oxi_simvivasmoi = oxi_simvivasmoi + 1;$

44) τιμωρία και για τους 2 πράκτορες

45) $i = i + 1;$

46) Πήγαινε στο βήμα 2}

Αυτό που λέει με λίγα λόγια ο ψευδοκώδικας πιο πάνω είναι να τρέξουμε το παιχνίδι όσες φορές θέλουμε, αρχικοποιώντας κάθε φορά όπως θέλουμε τους 2 πράκτορες. Επιλέγουμε τυχαία ένα πράκτορα, και τότε αυτός κάνει προσφορά. Αν κάνει προσφορά συμβιβασμού και δεχτεί ο άλλος πράκτορας τότε έχουμε συμβιβασμό και οι 2 πράκτορες πιάνουν επιβράβευση. Αν κάνει κανονική προσφορά και δεχτεί ο άλλος

πράκτορας τότε νικητής είναι αυτός με το πιο μεγάλο ποσό , με επιβράβευση για τον νικητή και τιμωρία στον πράκτορα με μικρότερο. Αν δεν δεχτεί αυτήν την κανονική προσφορά , τότε κάνει αυτός με τη σειρά του αντιστοίχως ότι έκανε ο προηγούμενος πράκτορας. Τέλος αν το ποσό εξαντληθεί, δεν έχουμε συμβιβασμό και οι 2 πράκτορες τιμωρούνται. Μετά από κάθε δράση πράκτορα γίνεται update ο πίνακας με τις εκτιμώμενες αμοιβές του πράκτορα.

Αυτός ο ψευδοκώδικας λέει ποια είναι η λογική του προγράμματος μας, χωρίς να κάνουμε την αντιστοιχία στο κυπριακό ακόμη. Δηλαδή απλά μας λέει για ένα γενικό Bargaining game , και 2 πράκτορες που μαθαίνουν με ενισχυτική μάθηση.

Τα 2 κύρια συστατικά του συστήματος είναι οι:

- 2 πράκτορες
- Προσομοίωση παιχνιδιού

Τα συστατικά αυτά θεωρούνται κύρια καθώς αυτά περιλαμβάνουν τα υπόλοιπα και επίσης είναι τα πιο σημαντικά και πιο μεγάλα .

Τώρα ας πάμε να δούμε ξεχωριστά τι γενική ιδέα πίσω από κάθε κλάση.

main: Αυτή είναι η κύρια κλάση , όπου δημιουργούμε 1 αντικείμενο της κλάσης Game, το οποίο είναι προσομοίωση ενός παιχνιδιού. Τρέχουμε αυτήν την προσομοίωση για ένα αριθμό επαναλήψεων , ο οποίος δηλώνει πόσες φορές θα παιχτεί το παιχνίδι. Αυτή όλη την διαδικασία την επαναλάβουμε για ένα ακόμη αριθμό επαναλήψεων που δεν αφορά όμως το παιχνίδι, αλλά για σκοπούς γραφικών και πιο γενικών συμπερασμάτων. Δηλαδή , στον αριθμό των επαναλήψεων που λέγεται επαναλήψεις παιχνιδιού , κάθε φορά που τελειώνει το παιχνίδι ξαναπαίζεται χωρίς όμως να αρχικοποιούνται οι καταστάσεις και οι γνώσεις κάθε πράκτορα. Σε αντίθεση με τον άλλο αριθμό ο οποίος είναι καθαρά για γραφικές και γενικά αποτελέσματα, όπου σε αυτόν τον αριθμό αρχικοποιούνται και οι καταστάσεις των πρακτόρων κάθε επανάληψη. Τέλος εκτυπώνουμε στο αρχείο results.txt για κάθε ένα από τα 4 ενδεχόμενα (ΣΥΜΒΙΒΑΣΜΟΙ,ΟΧΙ ΣΥΜΒΙΒΑΣΜΟΙ, ΝΙΚΗ_ΠΡΑΚΤΟΡΑ_1, ΝΙΚΕΣ_ΠΡΑΚΤΟΡΑ_B) πόση πιθανότητα υπάρχει να επιτευχθεί σε κάθε γύρο. [Παράδειγμα αυτού του αρχείου και επεξήγηση θα γίνει στο Παράρτημα με τον κώδικα.]

Game: Αυτή είναι καθαρά η κλάση όπου γίνεται προσομοίωση του παιχνιδιού για ένα αριθμό γύρων που παίρνει σαν παράμετρο. Εδώ δημιουργούμε 2 πράκτορες για το αλγόριθμο Q-learning και 2 για το αλγόριθμο SARSA με e-greedy τακτική και παίζεται το παιχνίδι μέχρι να ολοκληρωθούν οι γύροι και τότε θα έχουμε τον αριθμό των φορών για κάθε ενδεχόμενο.

Agent: Αυτή η κλάση περιγράφει τα χαρακτηριστικά ενός γενικού πράκτορα ανεξαρτήτως αλγορίθμου, δηλαδή τον πίνακα των καταστάσεων, το ρυθμό μάθησης, την κατάσταση που βρίσκεται και την δράση που εκτέλεσαι ο παίκτης, καθώς και την προηγούμενη κατάσταση και προηγούμενη δράση. Τέλος έχει το ποσό έκπτωσης και το μερίδιο που παίρνει στον παρών γύρο ο πράκτορας.

Q learning Agent: Αυτή η κλάση κληρονομιά όλα τα χαρακτηριστικά που αναφέραμε για την κλάση Agent, καθώς επίσης υλοποιεί τις συναρτήσεις για απόφαση που αφορά προσφορά που δέχεται και προσφορά που θα κάνει ο παίκτης με την λογική του αλγορίθμου Q-learning.

SARSA Agent : Αυτή η κλάση κληρονομιά όλα τα χαρακτηριστικά που αναφέραμε για την κλάση Agent, και επίσης έχει ένα δικό της χαρακτηριστικό, το ϵ , το οποίο αντιστοιχεί στην πιθανότητα εξερεύνησης, διότι όπως έχουμε αναφέρει έχουμε υλοποιήσει e-greedy τακτική σε αυτό τον αλγόριθμο. Το ϵ περνιέται σαν παράμετρος κατά την αρχικοποίηση των πρακτόρων από την κλάση Game. Επίσης υλοποιεί τις συναρτήσεις για απόφαση που αφορά προσφορά που δέχεται και προσφορά που θα κάνει ο παίκτης με την λογική του αλγορίθμου SARSA.

State: Αυτή η κλάση αντιστοιχεί στις καταστάσεις που μπορεί να έχει ένας πράκτορας. Έτσι φυλάγονται όλες οι δράσεις που μπορεί να εκτελέσει ο πράκτορας από την παρών κατάσταση, σε μία λίστα.

Action: Αυτή η κλάση αντιστοιχεί σε μία δράση του πράκτορα. Δηλαδή σε αυτήν την κλάση, κάθε αντικείμενο αυτής της κλάσης διατηρεί πληροφορίες για την δράση, όπως πια απόφαση περιγράφει η δράση και μία εκτίμηση για το πόσο καλή θα είναι αυτή η δράση ως προς το τελικό στόχο του πράκτορα.

Τέλος θα εξηγήσουμε μια σημαντική έννοια που αφορά την υλοποίηση. Θα εξηγήσουμε γενικά την ιδέα και τις δράσεις μίας κατάστασης.

Όπως έχουμε αναφέρει υπάρχουν 38 καταστάσεις , που αντιστοιχούν σε κάθε προσφορά που μπορεί να δεχτεί ένας πράκτορας(0-37). Έτσι στον πίνακα 3.3.1 πιο κάτω φαίνεται ο πίνακας καταστάσεων ενός πράκτορα στην αρχικοποίηση. Ο πίνακας αυτός χαρακτηρίζεται σαν Q_table για τον λόγο ότι περιλαμβάνει τις Q values τιμές για όλες τις καταστάσεις .

Καταρχάς το πώς μπορούμε να προσδιορίσουμε μία κατάσταση, είναι με το ποσό που υπάρχει διαθέσιμο και την προσφορά που έχουμε δεχτεί. Από την κάθε κατάσταση υπάρχουν κάποιες δράσεις που μπορούν να εκτελεστούν, όχι πάντα οι ίδιες από κάθε κατάσταση. Για να μπορούμε να προσδιορίσουμε μια δράση χρειαζόμαστε το αν θα δεχτεί η όχι την προσφορά ο πράκτορας, δηλαδή την απόφαση του, και την αντιπροσφορά για την απόφαση αυτή.

Είναι στην λογική ότι αν δεχτούμε μια προσφορά δεν χρειάζεται να κάνουμε αντιπροσφορά, η ότι αν το ποσό έχει λείψει πάλι δεν μπορούμε να κάνουμε αντιπροσφορά.

State		Action		
Offer	Pot	Decision (Accept/Refuse)	Counteroffer	Value Function
0	0	Accept	0	0
	>0	Refuse	<0.5	0.5
	>0	Refuse	>0.5	0.5
	>0	Refuse	=0.5	0.5
1	*	Refuse	<0.5	0.5
		Accept(<0.5)	*	0.5
		Refuse	>0.5	0.5
		Accept (>0.5)	*	0.5
		Refuse	=0.5	0.5
		Accept(0.5)	*	0.5

2	*	Refuse	<0.5	0.5
		Accept(<0.5)	*	0.5
		Refuse	>0.5	0.5
		Accept (>0.5)	*	0.5
		Refuse	=0.5	0.5
		Accept(0.5)	*	0.5
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
37	*	Refuse	<0.5	0.5
		Accept(<0.5)	*	0.5
		Refuse	>0.5	0.5
		Accept (>0.5)	*	0.5
		Refuse	=0.5	0.5
		Accept(0.5)	*	0.5

Πίνακας 3.3.1

Αυτός είναι ο πίνακας καταστάσεων του συστήματος για ένα πράκτορα στο γενικό bargaining game, κατά την αρχικοποίηση του πράκτορα και του παιχνιδιού. Το * υποδηλώνει ότι δεν μας νοιάζει το περιεχόμενο. Το <0.5 σημαίνει προσφορά μικρότερη του μισού του παρόντος ποσού, και τα >0.5, =0.5 μεγαλύτερο του μισού του παρόντος ποσού και ίσο του μισού του παρόντος ποσού αντίστοιχα.

Όπως βλέπουμε στον πίνακα 3.1.1 οι καταστάσεις είναι όσες και οι προσφορές που μπορούν να μας γίνουν. Θα επεξηγήσω γενικά τοN πίνακα πρώτα. Καταρχάς όπως είχα αναφέρει και πιο πάνω για να προσδιοριστεί μια κατάσταση χρειάζονται το ποσό που είναι διαθέσιμο και η προσφορά που δέχεται ο πράκτορας. Αυτό μας υποδηλώνει η πρώτη από τις 2 μεγάλες στήλες, η οποία αναφέρει πληροφορίες για την συγκεκριμένη κατάσταση που αντιστοιχεί σε αυτούς τους 2 παραμέτρους που προαναφέραμε. Στην δεύτερη στήλη, είναι οι δράσεις, ομαδοποιημένες για κάθε κατάσταση. Για να προσδιορίσουμε μια δράση για μία κατάσταση χρειαζόμαστε, την αντιπροσφορά και το τι απόφαση παίρνουμε για την προσφορά που μας γίνεται.

Γενικά ο πράκτορας έχει 2 είδη αποφάσεων για κάθε κατάσταση, να δεχτεί ή να απορρίψει, και 3 είδη αντιπροσφορών, να κάνει αντιπροσφορά μεγαλύτερη από το μισό του παρόντος ποσού, μικρότερη ή ίση. Έτσι δημιουργούνται 6 δράσεις από κάθε κατάσταση, με εξαίρεση την 0. Πιο ειδικά βλέπουμε ότι οι καταστάσεις χωρίζονται σε 2 είδη με βάση τις διαθέσιμες δράσεις τους. Το πρώτο είδος είναι η κατάσταση 0, δηλαδή όταν ο πράκτορας δέχεται προσφορά ίση με 0, και το δεύτερο είδος όταν ο πράκτορας δέχεται προσφορά μεγαλύτερη από 0.

Στο πρώτο είδος κατάστασης, δηλαδή 0, υπάρχουν 4 δράσεις που μπορούν να γίνουν. Αν το ποσό είναι μεγαλύτερο του 0, ο πράκτορας μπορεί να απορρίψει την προσφορά 0 και να κάνει αντιπροσφορά η μεγαλύτερη, ή μικρότερη, ή ίση με το ποσό. Αν η προσφορά είναι 0 και το ποσό 0 δεν έχει επιλογή πάρα να την δεχτεί θεωρητικά.

Για το δεύτερο είδος καταστάσεων, όλες οι προσφορές μεγαλύτερες του 0, ο πράκτορας έχει 6 δράσεις για να επιλέξει, που όμως μερικές είναι αντίθετες μεταξύ τους, άρα δεν μπορεί να επιλέξει κάθε φορά ανάμεσα σε όλες. Αν η προσφορά που του έχει γίνει είναι μικρότερη από το μισό του παρόντος ποσού μπορεί να το αποδεχτεί με το decision [Accept(<0.5)]. Αν η προσφορά είναι μεγαλύτερη από το μισό, η ίση με το μισό του ποσού μπορεί να την δεχτεί με τα decision [Accept(>0.5)] και [Accept($=0.5$)] αντίστοιχα. Αν ο πράκτορας επιλέξει να αποδεχτεί την προσφορά τότε υπονοείται ότι κάθε φορά, μία από τις 3 δράσεις που ανάφερα πιο πάνω μπορεί να επιλέξει, και αυτή η δράση εξαρτάται από την προσφορά και τη σχέση έχει με το ποσό. Αν όμως ο πράκτορας επιλέξει να απορρίψει την προσφορά έχει διαθέσιμες ακόμα 3 δράσεις. Η να την απορρίψει κάνοντας προσφορά μεγαλύτερη του μισού του ποσού, ίση ή μικρότερη του μισού του παρόντος ποσού. Έτσι καταλήγουμε στο συμπέρασμα ότι από οποιαδήποτε κατάσταση δευτέρου είδους, μπορεί να επιλέξει ανάμεσα από 4 από τις 6 διαθέσιμες δράσεις διότι οι άλλες αλληλοσυγκρούονται. Τέλος η τελευταία στήλη του πίνακα, υποδηλώνει πόσο κοντά στο να επιτύχω το καλύτερο δυνατό αποτέλεσμα είμαι. Δηλαδή όσο πιο ψηλό είναι αυτό το ποσό, τόσο πιο μεγάλη πιθανότητα υπάρχει να επιτύχω μακροπρόθεσμα καλό κέρδος, αν επιλέξω αυτήν την δράση της οποίας αντιστοιχεί το ποσό αυτό. Για αυτό το λόγο, όλα τα ποσά των δράσεων από τις καταστάσεις αρχικοποιούνται το ίδιο, με 0.5 εκτός στην κατάσταση όπου δέχεται ο παίκτης προσφορά 0 και το ποσό έχει λείψει. Σε αυτήν την περίπτωση δεν υπάρχει

πιθανότητα να κερδίσει κάτι, άρα το ποσό της δράσης είναι 0 και είναι η τερματική περίπτωση.

Κεφάλαιο 4

Αντιστοίχιση «Κυπριακού Προβλήματος» στο Bargaining game

4.1 Εισαγωγή

4.2 Κυπριακό Πρόβλημα σήμερα

4.3 Αντιστοίχιση Κυπριακού προβλήματος στο σύστημα

4.1 Εισαγωγή

Στο προηγούμενο κεφάλαιο εξηγήσαμε την γενική λογική σχεδιασμού και υλοποίησης του συστήματος μας. Σε αυτό το κεφάλαιο, αφού πρώτα δούμε την κατάσταση του προβλήματος του Κυπριακού την σήμερα, θα δούμε πως έχουμε αντιστοιχήσει το κυπριακό με το Bargaining game και την μοντελοποίηση του όσο αφορά υλοποίηση πάνω στο σύστημα μας.

4.2 Κυπριακό Πρόβλημα σήμερα

Όπως έχουμε αναφέρει και πιο πριν, το πρόβλημα το οποίο εξετάζουμε κατά πόσο μπορεί να αναχθεί στο Bargaining game, και με ενισχυτική μάθηση να δούμε αν μπορούν οι πράκτορες να φτάσουν σε συμβιβασμό, είναι το κυπριακό. Δηλαδή οι 2 πράκτορες μας θα είναι η Τουρκοκυπριακή πλευρά, στην οποία θα αναφερόμαστε σαν TK και η Ελληνοκυπριακή πλευρά, στην οποία θα αναφερόμαστε σαν ΕΚ. Οι 2 πράκτορες αυτοί έχουν ο καθένας τις δικές του απαιτήσεις και ο καθένας έχει κάτι που ο άλλος πράκτορας επιδιώκει να μοιραστεί μαζί του. Σήμερα το περίπου 37 % του εδάφους της Κύπρου το κρατά η TK πλευρά. Έτσι σκοπός της ΕΚ πλευράς είναι να

μοιραστεί αυτό το ποσοστό μαζί με την ΤΚ. Από την άλλη η ΕΚ πλευρά κατέχει την διακυβέρνηση όλου του κράτους, καθώς δεν είναι αναγνωρισμένο σαν ανεξάρτητο κράτος το έδαφος που κατέχουν οι ΤΚ. Άρα σκοπός των ΤΚ είναι να μοιραστούν αυτήν την διακυβέρνηση , ή στοιχεία της μαζί με την ΕΚ πλευρά. Δηλαδή η ΤΚ πλευρά έχει να δώσει ποσοστό γεωγραφικής έκτασης, και η ΕΚ πλευρά έχει να δώσει ποσοστό σε στοιχεία της διακυβέρνησης του κράτους. Όταν αναφερόμαστε σε διακυβέρνηση του κράτους, εννοούμε τις 3 εξουσίες. Οι 3 εξουσίες αυτές είναι η εκτελεστική, η νομοθετική και η δικαστική. Εκτελεστική εξουσία είναι ο πρόεδρος, ο αντιπρόεδρος και το υπουργικό συμβούλιο του κράτους. Νομοθετική εξουσία είναι η βουλή αντιπροσώπων του κράτους και δικαστική εξουσία είναι το Ανώτατο Δικαστικό Συμβούλιο του κράτους. Όταν αναφερόμαστε σε ποσοστό διακυβέρνησης που απαιτεί η ΤΚ πλευρά εννοούμε θέσεις σε αυτές τις ομάδες , για παράδειγμα θέσεις ατόμων τα οποία να είναι ΤΚ στο Ανώτατο Δικαστικό Συμβούλιο, στην βουλή αντιπροσώπων , στο υπουργικό συμβούλιο ή ακόμη και εκ περιτροπής προεδρία, δηλαδή κάθε χρόνο να έχουμε εναλλαγή από ΕΚ σε ΤΚ πρόεδρο η αντιπρόεδρο. Με την σειρά της η ΕΚ πλευρά απαιτεί γεωγραφικό ποσοστό, δηλαδή κομμάτια εδάφους και δικαιώματα πάνω σε αυτά που κατέχει η ΤΚ πλευρά. Όταν αναφερόμαστε στη εδαφική πτυχή του Κυπριακού, εννοούμε τον διαμοιρασμό του εδάφους του 37 % που κατέχει η ΤΚ πλευρά, με παράδοση περιοχών που μπορούν να κατοικηθούν από ΕΚ πρόσφυγες, παραδείγματος χάρη, Μόρφου, Αμμόχωστο και άλλες, καθώς και αριθμό προσφύγων που δικαιούνται να πάνε σε αυτές τις περιοχές . Επίσης όταν αναφερόμαστε στη γεωγραφική πτυχή του κυπριακού ή «εδαφικό», περιλαμβάνεται και ο διαμοιρασμός της ακτογραμμής καθώς και η θαλάσσια έκταση, που έχει δικαίωμα η κάθε πλευρά να εκμεταλλευτεί θαλάσσιους πόρους που μπορεί να υπάρχουν στην θαλάσσια αυτή έκταση, γνωστή και σαν Αποκλειστική Οικονομική Ζώνη (ΑΟΖ). Με αυτές τις απαιτήσεις και δεδομένα μία λύση συμβιβασμού περιλαμβάνει υποχωρήσεις και από τις 2 πλευρές. Εμείς θα ασχοληθούμε καθαρά με το εδαφικό παράγοντα του κυπριακού, δηλαδή με το έδαφος, στο οποίο συμπεριλαμβάνονται και ΑΟΖ και ακτογραμμές όπως έχουμε αναφέρει, που αναμένεται να χωρίσουν οι 2 πλευρές. Επειδή όμως θέλουμε να δούμε ρεαλιστική αντιστοίχιση του προβλήματος, η ΤΚ πλευρά , αν πάρουμε το γεωγραφικό τομέα δεν έχει να επιζητήσει κάτι από την ΕΚ πλευρά, καθώς κατέχει το 37 % που είναι πολύ περισσότερο απο το πλυθισμιακό ποσοστό της (18 %) και το οποίο θέλει να μοιραστεί η ΕΚ πλευρά. Έτσι θα θεωρήσουμε, όπως θα εξηγήσουμε και

στην συνέχεια ότι όταν η TK πλευρά αρχίζει να δέχεται και να κάνει κάνει διαπραγματεύσεις για γεωγραφική περιοχή, η ΕΚ πλευρά με την σειρά της αρχίζει να κάνει διαπραγματεύσεις για την διακυβέρνηση, τις οποίες θα δούμε γενικά και δεν θα τις συμπεριλάβουμε μέσα στο εδαφικό πρόβλημα, παρά μόνο σαν «εξωτερικός» παράγοντας που θα αναγκάσει την TK πλευρά να αρχίσει να δέχεται διαπραγματεύσεις. «Εξωτερικός παράγοντας» εννοούμε όσο αφορά το σύστημα μας ,το οποίο έχει σαν σκοπό να μοιράσουν την γεωγραφική περιοχή οι 2 πλευρές, όσο αφορά το κυπριακό σαν γενικό πρόβλημα δεν είναι καθόλου εξωτερικός παράγοντας όπως έχουμε αναφέρει και πιο πάνω, καθώς είναι αυτό που έχει να προσφέρει η ΕΚ πλευρά.

Με βάση αυτά που επικρατούν σήμερα και τις απαιτήσεις των 2 πλευρών, και με συνάρτηση τις πρόσφατες διαπραγματεύσεις στην Γενεύη, θα θεωρήσουμε μια δική μας λύση όσο αφορά το γεωγραφικό διαμοιρασμό του ποσού 37 % που κατέχει η TK πλευρά. Σύμφωνα με τις πρόσφατες διαπραγματεύσεις η TK πλευρά δεν δέχεται να μείνει με ποσοστό μικρότερο του 29.2 %, ενώ σε αντίθεση η ΕΚ πλευρά απαιτεί ποσοστό τάξεως του 8.8 % που θα αφήσει στην TK πλευρά ποσοστό 28.2 % . Άρα όπως βλέπουμε υπάρχει μια διαφορά της τάξεως 1 % με αυτά που συμβιβάζονται οι 2 πλευρές. Εγώ για σκοπούς υλοποίησης θα θεωρήσω σαν θεωρητική λύση συμβιβασμού μεταξύ των 2 πλευρών το ποσοστό για τους TK να είναι 27 % και για τους ΕΚ 10 %. Δηλαδή από τις 37 «μονάδες» εδάφους οι 27 θα μείνουν στους TK και οι 10 στους ΕΚ. Βεβαίως όπως εξήγησα αυτό το ποσό είναι απλά ενδεικτικό από εμένα για να υπάρχει μια τερματική περίπτωση στο σύστημα, ποσοστό όμως που είναι σε λογικά πλαίσια με τις διαπραγματεύσεις μέχρι σήμερα. Το ποσοστό αυτό είναι απλά μια παράμετρος κατά την αρχικοποίηση των πρακτόρων στο σύστημα, άρα εύκολα μπορεί κάποιος να το αλλάξει και να τρέξει το σύστημα με δικό του ποσό για συμβιβασμό.

Επίσης για το σύστημα πρέπει να δηλώσουμε ένα ποσοστό το οποίο να ικανοποιεί την ΕΚ πλευρά και ένα την TK πλευρά , έτσι ώστε να μπορούμε να βγάλουμε κερδισμένους και ηττημένους κάθε επανάληψης. Δηλαδή αν ο πράκτορας κατά τον μοίρασμα του ποσού , κατέχει μονάδες περισσότερες από αυτές που απλά ικανοποιείται θεωρείται κερδισμένος της επανάληψης, σε σύγκριση με το άλλο πράκτορα ο οποίος θα έχει πιάσει λιγότερες μονάδες από αυτές που είναι ικανοποιημένος, καθώς αφού μοιράζονται το ποσό δεν υπάρχει περίπτωση να έχουμε και τους 2 νικητές με μονάδες περισσότερες από αυτές που απλά τους ικανοποιούν. Στο σύστημα έχω δηλώσει σαν

ποσό ικανοποίησης του ΕΚ πράκτορα το 10, δηλαδή αν κατά τον διαμοιρασμό του ποσού πιάσει ποσό περισσότερο του 10 είναι θεωρητικά κερδισμένος της επανάληψης, και για τον ΤΚ το ποσό που θεωρητικά θα είναι κερδισμένος της επανάληψης ανέρχεται στο 27 αντίστοιχα.

Πριν προχωρήσουμε στην μοντελοποίηση και αντιστοίχιση να πούμε κάποια γενικά πράγματα που αφορούν την αντιστοιχία. Όπως έχουμε αναφέρει οι 2 πράκτορες του συστήματος θα είναι η ΤΚ πλευρά και η ΕΚ πλευρά. Το ποσό που θέλουν να μοιράσουν αναφέρεται στο 37 % γεωγραφικής έκτασης της Κύπρου που κατέχει Η ΤΚ πλευρά. Η μέγιστη λύση συμβιβασμού για το κοινό καλό που έχω αναφέρει πιο πάνω για το σύστημα είναι 10 μονάδες από τις 37 στην ΕΚ πλευρά και οι υπόλοιπες 27 στην ΤΚ.

Με την έννοια γύρος, αναφερόμαστε στον αριθμό του ποσού που υπάρχει διαθέσιμο ανά χρονική στιγμή. Δηλαδή όταν ξεκινήσει το παιχνίδι θα έχουμε ποσό 37 και θα παίζεται ο αρχικός , πρώτος γύρος. Όταν μειωθεί το ποσό κατά 1 μονάδα, δηλαδή μετά από 2 διαδοχικές απορρίψεις προσφορών , 1 για κάθε πράκτορα θα έχουμε το 2 γύρο και ούτω κάθεξής. Άρα όταν το ποσό έχει γίνει 0 θα βρισκόμαστε στο τελευταίο γύρο.

Με τα προαναφερθέντα δεδομένα κάθε επανάληψη μπορεί να τελειώσει σε ένα από τα 4 παρακάτω ενδεχόμενα που φαίνονται στο πίνακα 4.2 .

<u>Ενδεχόμενο</u>	<u>Ποσό ΕΚ</u> <u>(ΜΟΝΑΔΕΣ)</u>	<u>Ποσό ΤΚ</u> <u>(ΜΟΝΑΔΕΣ)</u>	<u>ΓΥΡΟΣ ΠΟΥ</u> <u>ΜΠΟΡΕΙ ΝΑ</u> <u>ΕΠΙΤΕΥΧΘΕΙ</u>
ΣΥΜΒΙΒΑΣΜΟΣ	10	27	ΜΟΝΟ ΣΤΟ ΑΡΧΙΚΟ(1 ^ο)
ΟΧΙ ΣΥΜΒΙΒΑΣΜΟΣ	0	0	ΜΟΝΟ ΣΤΟ ΤΕΛΕΥΤΑΙΟ(37 ^ο)
ΝΙΚΗ ΕΚ	>10	<27	1 ΜΕΧΡΙ 36 ΓΥΡΟ
ΝΙΚΗ ΤΚ	<10	>27	2 ΜΕΧΡΙ 36 ΓΥΡΟ

Πίνακας 4.2

Πίνακας πιθανών ενδεχομένων κατάληξης κάθε επανάληψης.

Όπως φαίνεται και στο πίνακα 4.2 τα ενδεχόμενα τερματισμού μιας επανάληψης είναι συμβιβασμός, όχι συμβιβασμός, νίκη EK, νίκη TK. Η μόνη περίπτωση συμβιβασμού που να έχουμε λύση για το κοινό καλό είναι να πάρει ο EK 10 μονάδες και ο TK 27. Αυτό προϋποθέτει ότι ο διαμοιρασμός θα επιτευχθεί μόνο στο πρώτο γύρο της επανάληψης, δηλαδή πριν το ποσό μειωθεί, διότι όπως έχουμε αναφέρει το ποσό μειώνεται κάθε γύρο κατά 1 μονάδα στο Bargaining Game. Έτσι αν δεν φτάσουν σε συμβιβασμό το πρώτο γύρο δεν υπάρχει δυνατότητα συμβιβασμού πιο μετά. Αν δεν συμβιβαστούν στο πρώτο γύρο, το ποσό μειώνεται και τότε υπάρχουν τα υπόλοιπα 3 ενδεχόμενα διαθέσιμα για τερματισμό. Αν ένας από τους 2 πράκτορες δεχτεί προσφορά από τον άλλο πριν το ποσό γίνει 0 θα έχουμε είτε Νίκη EK(αν πιάσει μονάδες περισσότερες από 10) ή νίκη TK(αν πιάσει μονάδες περισσότερες από 27). Τέλος όταν το ποσό εξαντληθεί υπάρχει το ενδεχόμενο μόνο του μη συμβιβασμού, όπου δεν υπάρχει πλέον γεωγραφική περιοχή να μοιράσουν άρα δεν φτάνουν σε συμβιβασμό και τότε έχουμε και τις 2 πλευρές ηττημένες διότι δεν πήραν κάτι που να απαιτούσαν. (Αφού δεν έχουν φτάσει σε μοίρασμα γεδάφους, τότε ούτε η EK πλευρά δίνει ποσοστά διακυβέρνησης στην TK)

4.3 Αντιστοίχιση κυπριακού στο σύστημα

Σε αυτό το υποκεφάλαιο θα εξηγήσουμε πως το σύστημα μας ξεφεύγει από την κανονική υλοποίηση ενός γενικού αθροίσματος παιχνιδιού με 2 παίκτες, τί το κάνει δηλαδή να αντιστοιχεί το Κυπριακό πρόβλημα σαν ένα bargaining game. Με λίγα λόγια θα αναφέρουμε τις αλλαγές της κανονικής υλοποίησης που αναφέρεται στο Κεφάλαιο 3 με την υλοποίηση που κάναμε για αντιστοιχία του κυπριακού στο bargaining game.

Πρώτα από όλα θα θυμηθούμε κάποια βασικά συστατικά υλοποίησης για το γενικό bargaining game. Τα κύρια συστατικά που υπάρχουν είναι η προσομοίωση και οι 2 πράκτορες μας. Η προσομοίωση θα επεξηγηθεί πιο αναλυτικά στην συνέχεια κατά την

διάρκεια αναφοράς σε ένα τρέχον παράδειγμα. Οι 2 πράκτορες που έχουμε είναι η ΤΚ πλευρά και η ΕΚ πλευρά. Συμβιβασμός όπως έχουμε ορίσει στο προηγούμενο υποκεφάλαιο είναι να πιάσει η ΤΚ πλευρά 27 μονάδες από τις 37 και τις υπόλοιπες 10 η ΕΚ πλευρά. Οποιοσδήποτε άλλος διαμοιρασμός δεν θεωρείται συμβιβασμός, και επίσης ο συμβιβασμός αυτός μπορεί να επιτευχθεί μόνο στον πρώτο γύρο της επανάληψης , δηλαδή πριν μειωθεί το ποσό, καθώς δεν υπάρχει έννοια για λύση αν το ποσό μειωθεί. Το ποσό μειώνεται έτσι ώστε οι 2 πλευρές αφού φτάσουν σε αρκετούς μη συμβιβασμούς να αρχίσουν να σκέφτονται το ενδεχόμενο να προτείνουν συμβιβασμό με την λύση που αναφέραμε πιο πάνω.

Αυτό που κάνει το σύστημα μας να αντιστοιχείται το κυπριακό σε bargaining game , είναι καταρχάς η ιδιότητα αρχικοποίησης των 2 πρακτόρων. Όπως και στα παραπάνω ρεαλιστικά προβλήματα διαπραγματεύσεων , υπάρχει ένας πράκτορας που έχει το πλεονέκτημα σε κάποιους παράγοντες, πλεονέκτημα το οποίο τον κάνει να έχει το «πάνω χέρι».

Στο σύστημα μας, το οποίο εξετάζει το ενδεχόμενο συμβιβασμού όσο αφορά το εδαφικό παράγοντα του κυπριακού, αυτό το πλεονέκτημα το έχει η ΤΚ πλευρά, καθώς όπως έχουμε είδη εξηγήσει κατέχει το 37 % , δηλαδή ολόκληρη την γεωγραφική έκταση που θέλουμε να μοιραστεί, και όσο αφορά γεωγραφικά, δεν έχει να κερδίσει κάτι με το να δώσει γη στην ΕΚ πλευρά ή έστω να διαπραγματευτεί. Αυτό φαίνεται στο ΤΚ πράκτορα από τον πίνακα καταστάσεων και τιμών του που φαίνεται στον πίνακα 4.3.3 και θα εξηγήσουμε σε λίγο.

State		Action		
Offer	Pot	Decision (Accept/Refuse)	Counter Demand	Value Function
0	0	Accept	0	0
	>0	Refuse	<0.5	0.5
	>0	Refuse	>0.5	0.5
	>0	Refuse	=0.5	0.5

1	*	Refuse	<0.5	0.5
		Accept(<0.5)	*	0.5
		Refuse	>0.5	0.5
		Accept (>0.5)	*	0.5
		Refuse	=0.5	0.5
	*	Accept(0.5)	*	0.5
		Refuse	<0.5	0.5
		Accept(<0.5)	*	0.5
		Refuse	>0.5	0.5
		Accept (>0.5)	*	0.5
	*	Refuse	=0.5	0.5
		Accept(0.5)	*	0.5
		Refuse	<0.5	0.5
		Accept(<0.5)	*	0.5
		Refuse	>0.5	0.5
	*	Accept (>0.5)	*	0.5
		Refuse	=0.5	0.5
		Accept(0.5)	*	0.5
		Refuse	<0.5	0.5
		Accept(<0.5)	*	0.5
37	*	Refuse	>0.5	0.5
		Accept (>0.5)	*	0.5
		Refuse	=0.5	0.5
		Accept(0.5)	*	0.5
		Refuse	<0.5	0.5
	*	Accept(<0.5)	*	0.5
		Refuse	>0.5	0.5
		Accept (>0.5)	*	0.5
		Refuse	=0.5	0.5
		Accept(0.5)	*	0.5

Πίνακας 4.3.1

Πίνακας αρχικοποίησης καταστάσεων-τιμών για ΕΚ πράκτορα.

State		Action			
Offer	Pot	Decision (Accept/Refuse)	Counter Demand	Counter Offer	Value Function
0	0	Accept	0	*	0

	>0	Refuse	<0.5	Θέσεις στο Ανώτατο Δικαστικό συμβούλιο, θέσεις στην βουλή και παραχώρηση βέτο σε αριθμό ΤΚ υπουργών, εκ περιτροπής προεδρία, θέσεις στο υπουργικό συμβούλιο	0.5
		Refuse	>0.5		0.5
		Refuse	=0.5		0.5
1	*	Refuse	<0.5		0.5
		Accept(<0.5)	*		0.5
		Refuse	>0.5		0.5
		Accept (>0.5)	*		0.5
		Refuse	=0.5		0.5
2	*	Accept(0.5)	*		0.5
		Refuse	<0.5		0.5
		Accept(<0.5)	*		0.5
		Refuse	>0.5		0.5
		Accept (>0.5)	*		0.5
37	*	Refuse	=0.5		0.5
		Accept(0.5)	*		0.5
		Refuse	<0.5		0.5
		Accept(<0.5)	*		0.5
		Refuse	>0.5		0.5
.	.	Accept (>0.5)	*		0.5
		Refuse	=0.5		0.5
		Accept(0.5)	*		0.5
.	.	.	.		.
.	.	.	.		.
.	.	.	.		.

Πίνακας 4.3.2

Υποθετικός Πίνακας αρχικοποίησης καταστάσεων-τιμών για ΕΚ πράκτορα
λαμβάνοντας υπόψη και τα στοιχεία διακυβέρνησης.

Πριν να συγκρίνουμε τους 2 πίνακες αρχικοποίησης των καταστάσεων των 2 πλευρών,
να δούμε την τροποποίηση του πίνακα αρχικοποίησης καταστάσεων της ΕΚ πλευράς σε

σύγκριση με τον πίνακα ενός γενικού πράκτορα , πίνακα 3.3. Η μόνη διαφορά είναι ότι καθώς ο ΕΚ πράκτορας έχει μηδενικό ποσοστό εδάφους από το 37 % που κατέχει η ΤΚ πλευρά δεν υπάρχει η έννοια της προσφοράς ποσού εδάφους. Αυτό που εννοούμε είναι ότι η ΕΚ πλευρά δεν προσφέρει γη, απλά απαιτεί. Αυτό που έχει να προσφέρει είναι στοιχεία από την διακυβέρνηση. Αυτό φαίνεται στον πίνακα 4.3.2 ο οποίος μας δίνει ένα παράδειγμα σημασίας του πίνακα 4.3.1 σε αντιστοίχιση σε όλο το Κυπριακό πρόβλημα και όχι μόνο στο γεωγραφικό τομέα.

State		Action		
Offer	Pot	Decision (Accept/Refuse)	Counteroffer	Value Function
0	0	Accept	0	0
	>0	Refuse	<0.5	0.5
	>0	Refuse	>0.5	0.5
	>0	Refuse	=0.5	0.5
	>0	Refuse	0	1
1	*	Refuse	<0.5	0.5
		Accept(<0.5)	*	0.5
		Refuse	>0.5	0.5
		Accept (>0.5)	*	0.5
		Refuse	=0.5	0.5
		Accept(0.5)	*	0.5
2	*	Refuse	0	1
		Refuse	<0.5	0.5
		Accept(<0.5)	*	0.5
		Refuse	>0.5	0.5
		Accept (>0.5)	*	0.5
		Refuse	=0.5	0.5
	*	Accept(0.5)	*	0.5
		Refuse	<0.5	0.5
		Accept(<0.5)	*	0.5
		Refuse	>0.5	0.5
		Accept (>0.5)	*	0.5
		Refuse	=0.5	0.5
	*	Accept(0.5)	*	0.5
		Refuse	<0.5	0.5
		Accept(<0.5)	*	0.5
		Refuse	>0.5	0.5
		Accept (>0.5)	*	0.5
		Refuse	=0.5	0.5

		Refuse	0	1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
37	*	Refuse	<0.5	0.5
		Accept(<0.5)	*	0.5
		Refuse	>0.5	0.5
		Accept (>0.5)	*	0.5
		Refuse	=0.5	0.5
		Accept(0.5)	*	0.5
		Refuse	0	1

Πίνακας 4.3.3

Πίνακας αρχικοποίησης καταστάσεων-τιμών για ΤΚ πράκτορα.

Η διαφορά του πίνακα 4.3.3, ο οποίος αντιπροσωπεύει την αρχικοποίηση των καταστάσεων και των συναρτήσεων αξίας για τον ΤΚ πράκτορα, με τον πίνακα 4.3.1 , ο οποίος αντιπροσωπεύει την αρχικοποίηση των καταστάσεων και των συναρτήσεων αξίας για τον ΕΚ πράκτορα είναι ότι σε κάθε κατάσταση ο ΤΚ παίκτης έχει μια δράση παραπάνω. Αυτή η δράση φαίνεται σημειωμένη στον πίνακα 4.3.3. Η λογική αυτής της πράξης είναι ότι ο πίνακας καταστάσεων του ΤΚ πράκτορα , έχει επιπλέον την επιλογή να απορρίπτει από κάθε κατάσταση οποιαδήποτε προσφορά από τον ΕΚ , χωρίς να δίνει καν αντιπροσφορά. Αυτό φαίνεται από το ότι η αξία αυτής της δράσης έχει την τιμή 1, που είναι μεγαλύτερη από όλες τις άλλες δράσεις, που έχει σαν αποτέλεσμα ο ΤΚ να επιλέγει αυτήν την δράση πάντα. Αυτό ισχύει και στην πραγματικότητα καθώς όπως αναφέραμε δεν έχει λόγο να δεχτεί προσφορά ή να δώσει ποσοστό του εδάφους που κατέχει στην ΕΚ πλευρά. Αυτό που θα τον κάνει να αρχίσει να δέχεται προσφορές και να κάνει διαπραγματεύσεις είναι το να δώσει στοιχεία από την διακυβέρνηση η ΕΚ πλευρά, που φαίνονται στον υποθετικό πίνακα καταστάσεων για την ΕΚ πλευρά για όλες τις πτυχές του Κυπριακού. Όμως επειδή εμείς θα ασχοληθούμε σε αυτήν την διπλωματική μόνο για διαμοιρασμό της γεωγραφικής περιοχής των κατεχομένων θα

υποθέσουμε ότι ξεκινώντας το παιχνίδι, η ΕΚ πλευρά αρχίζει και δίνει στοιχεία από την διακυβέρνηση στην ΤΚ πλευρά, που είναι τι δεν έχει η ΤΚ πλευρά και τι απαιτεί, και με αυτήν την πράξη αναγκάζεται να αρχίσει διαπραγματεύσεις για το έδαφος που κατέχει (37 %). Αυτή η ανάγκη για παραχώρηση διακυβέρνησης στην ΤΚ από την ΕΚ, έχει την μορφή προσωρινών τιμωριών στον ΤΚ πράκτορα, ο οποίος δεχόμενου αυτών των τιμωριών θα αρχίσει να διαπραγματεύεται την γεωγραφική έκταση που κατέχει. Οι τιμωρίες αυτές επηρεάζουν την δράση που επιλέγεται από τον ΤΚ πράκτορα στους αρχικούς γύρους της επανάληψης αυτό θα έχει σαν συνέπεια να πέσει η συνάρτηση αξίας της δράσης αυτής από 1 που ήταν στην αρχή σε < 0.5 που θα επιφέρει την δυνατότητα στο ΤΚ πράκτορα να μην απορρίπτει όλες τις προσφορές, αλλά πλέον να τις λαμβάνει υπόψη και να παραχωρεί έδαφος για προσφορά. Με λίγα λόγια, στους πρώτους γύρους του παιχνιδιού ο ΤΚ θα απορρίπτει όλες τις προτάσεις από τον ΕΚ, διότι από κάθε κατάσταση η δράση που θα έχει καλύτερο αποτέλεσμα για αυτό είναι η δράση που προσθέσαμε και είναι σημειωμένη στον πίνακα 4.3.3, με συνάρτηση αξίας 1. Στην συνέχεια θα δέχεται τιμωρία ο ΤΚ με αποτέλεσμα από κάθε κατάσταση να πέφτει η συνάρτηση αξίας της δράσης αυτής, μέχρι να γίνει μικρότερη του 0.5, που είναι η αρχική συνάρτηση αξίας για τις άλλες δράσεις, και έτσι ο ΤΚ πράκτορας θα αρχίζει να λαμβάνει υπόψη και τις άλλες δράσεις και να αρχίσει διαπραγματεύσεις, που θα τον κάνει να συμπεριφέρεται σαν τον ΕΚ πράκτορα, αφού πλέον δεν θα απορρίπτει τις προσφορές. Αυτό θα φανεί και θα εξηγηθεί ακόμη περισσότερο στο κεφάλαιο 5 που θα έχουμε γραφικές και αποτελέσματα.

Κάτι άλλο που είναι σημαντικό να αναφέρω για την υλοποίηση είναι ότι όταν ο γύρος ξεκινήσει και ο ΤΚ πράκτορας έχει διαθέσιμη την δράση να απορρίψει εκ των προτέρων την προσφορά-απαίτηση του ΕΚ πράκτορα τότε η επανάληψη τερματίζεται και θεωρείται σαν ΝΙΚΗ του ΤΚ πράκτορα διότι στο παρόν στάδιο δεν έχει λόγο να δώσει γη (πριν να αρχίσουν οι τιμωρίες- ΕΚ πλευρά να δίνει ποσοστά από διακυβέρνηση) και έτσι επειδή την κατέχει θεωρείται κερδισμένος της επανάληψης (37 μονάδες ΤΚ- 0 μονάδες ΕΚ). Όταν φτάσει στο σημείο να κάνει διαπραγματεύσεις τότε όλα τα καταληκτικά σενάρια είναι διαθέσιμα (ΣΥΜΒΙΒΑΣΜΟΣ, ΜΗ ΣΥΜΒΙΒΑΣΜΟΣ, ΝΙΚΗ ΕΚ, ΝΙΚΗ ΤΚ).

Με όλα αυτά που έχουμε αναφέρει ως δώσουμε μια γενική επεξήγηση του τροποποιημένου συστήματος στο οποίο έχουμε πλέον αντιστοιχήσει το Κυπριακό πρόβλημα στο bargaining game, πριν πάμε να δούμε τα αποτελέσματα.

Το παιχνίδι θα παιχτεί για ένα αριθμό επαναλήψεων, καθώς έχουμε επαναλαμβανόμενο bargaining game, που σημαίνει ότι οι πράκτορες έχουν γνώση από προηγούμενες επαναλήψεις και αποφάσεις τους. Αρχικοποιούμε τους 2 πράκτορες με τους πίνακες 4.3.3 για τον TK και 4.3.1 για τον EK. Στους πρώτους γύρους, λόγω αυτών των πινάκων ο TK πράκτορας σε ότι προσφορά-απαίτηση έχει ο EK θα την απορρίπτει και θα νικάει την επανάληψη διότι είδη κατέχει τις 37 μονάδες εδάφους, ενώ ο EK 0. Άρα γεωγραφικά ικανοποιείται ο TK (μεγαλύτερες από 27 μονάδες ικανοποιείτε). Στην συνέχεια ο EK παίκτης θα κάνει προσφορά ποσοστά από την διακυβέρνηση το οποίο θα εκφράζεται σαν τιμωρία στον TK πράκτορα, αφού ο EK αρχίζει να διαπραγματεύεται αυτά που απαιτεί ο TK, και ο TK απλά απορρίπτει το να δώσει έδαφος. Αυτές οι τιμωρίες θα τον αναγκάσουν να λαμβάνει υπόψη και τις άλλες δράσεις από κάθε κατάσταση, και να αρχίσει να διαπραγματεύεται και αυτός προσφέροντας με την σειρά του μονάδες εδάφους στον EK. Όταν πλέον ο TK σταματήσει να απορρίπτει χωρίς να λαμβάνει υπόψη τις άλλες δράσεις που έχει διαθέσιμες, τότε οι τιμωρίες σταματούν να του επιβαρύνονται και τότε ο TK θα λειτουργεί σαν τον EK πράκτορα. Σε αυτήν την χρονική στιγμή θα υπάρχει πλέον το ενδεχόμενο οι 2 πράκτορες να συμβιβάζονται με το ποσό 27 για TK και 10 για EK που ικανοποιά και τους 2 σύμφωνα με τις υποθέσεις της λύσης μας, ενδεχόμενο το οποίο όμως μπορεί να επιτευχθεί στον πρώτο γύρο, δηλαδή χωρίς να μειωθεί το ποσό που διαπραγματεύονται. Αν μειωθεί έστω και μια μονάδα τότε δεν υπάρχει πλέον αυτό το ενδεχόμενο για την τυχόν επανάληψη και υπάρχουν 3 άλλα ενδεχόμενα για να τερματιστεί η επανάληψη. Τα 3 αυτά ενδεχόμενα είναι, ΝΙΚΗ EK, ΝΙΚΗ TK, ΟΧΙ-ΣΥΜΒΙΒΑΣΜΟΣ. Το ενδεχόμενο όχι-συμβιβασμού πετυχαίνετε μόνο όταν το ποσό γίνει 0, που έχει σαν συνέπεια να μην μοιραστούν το έδαφος, άρα να μείνουν στάσιμοι στην διαπραγμάτευση, γεγονός το οποίο θα είναι αποτυχία και για τους 2 διότι δεν παίρνουν κάτι από τις απαιτήσεις τους, καθώς ο EK δεν θα πιάσει τίποτε ποσοστό εδάφους, Ο TK δεν θα πιάσει μερίδιο στην διακυβέρνηση από τον EK. Τα άλλα 2 ενδεχόμενα μπορούν να επιτευχθούν από το 1 γύρο μέχρι τον προτελευταίο(διότι στον τελευταίο γύρο δεν μπορούν να μοιράσουν το ποσό καθώς έχει γίνει 0). Έτσι αν κάποιος από τις 2 πλευρές δεχτεί πρόταση της άλλης σε αυτό το

διάστημα γύρων τερματίζεται η επανάληψη και μοιράζεται το παρών ποσό στους 2 πράκτορες. Αν το ποσό που έχει κερδίσει η ΕΚ πλευρά είναι μεγαλύτερο από την ικανοποίηση της (ενδεχόμενο συμβιβασμού που παίρνει 10 μονάδες) τότε έχουμε ΝΙΚΗ ΕΚ και ήττα ΤΚ καθώς δεν τους απομένει ποσό μεγαλύτερο από την ικανοποίηση τους (μεγαλύτερο από 27). Από την άλλη αν ο ΤΚ πράκτορας μείνει με μονάδες μεγαλύτερες από τις μονάδες που απλά συμβιβάζεται (27 μονάδες) έχουμε ΝΙΚΗ ΤΚ και ήττα ΕΚ διότι αυτό συνεπάγεται ότι το ποσό που θα κερδίσει ο ΕΚ πράκτορας είναι μικρότερο από τις ελάχιστες απαιτήσεις συμβιβασμού (10).

Αυτή όλη η διαδικασία παίζεται για ένα αριθμό επαναλήψεων όπου θα δούμε τα αποτελέσματα στο κεφάλαιο που ακολουθεί.

Κεφάλαιο 5

Αποτελέσματα και ανάλυση

5.1 Εισαγωγή

5.2 Αποτελέσματα και ανάλυση κανονικού συστήματος για παιχνίδι Bargaining Game

5.3 Αποτελέσματα και ανάλυση Αντιστοίχισης Κυπριακού προβλήματος σε Bargaining Game στο σύστημα

5.3.1 Πράκτορες με αλγόριθμο Q-learning

5.3.1.1 Μελέτη παραμέτρου ρυθμού μάθησης

5.3.1.2 Μελέτη παραμέτρου έκπτωσης

5.3.1.2.1 Μελέτη με διαφορετικούς παραμέτρους έκπτωσης

για κάθε πράκτορα

5.3.1.3 Μελέτη παραμέτρου επαναλήψεων

5.3.1.4 Βέλτιστες παραμέτροι για αλγόριθμο Q-learning

5.3.2 Πράκτορες με αλγόριθμο SARSA με e-greedy τακτική

5.3.2.1 Μελέτη παραμέτρου ρυθμού μάθησης

5.3.2.2 Μελέτη παραμέτρου έκπτωσης

5.3.2.3 Μελέτη παραμέτρου επαναλήψεων

5.3.2.4 Μελέτη παραμέτρου ϵ

5.3.2.5 Βέλτιστες παραμέτροι για αλγόριθμο SARSA με e-greedy τακτική

5.3.3 Πειραματισμός με διαφορετικά ποσά συμβιβασμού

5.3.4 Συμπεράσματα και σύγκριση αλγορίθμων

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο, θα δούμε πρώτα τα αποτελέσματα ενός κανονικού παιχνιδιού bargaining game, μεταξύ 2 πρακτόρων. Έπειτα θα δούμε και θα αναλύσουμε τα αποτελέσματα του συστήματος με αντιστοίχιση του κυπριακού προβλήματος στο bargaining game, όπως αναλύσαμε στο προηγούμενο κεφάλαιο, στους αλγόριθμους Q-learning και SARSA. Θα μελετήσουμε επίσης και θα σχολιάσουμε κάποιες παραμέτρους των αλγορίθμων αυτών και τέλος θα συγκρίνουμε όλα τα αποτελέσματα του συστήματος.

5.2 Αποτελέσματα και ανάλυση κανονικού συστήματος για παιχνίδι Bargaining Game

Στις γραφικές που ακολουθούν θα δούμε αποτελέσματα ενός κανονικού bargaining game μεταξύ 2 πρακτόρων.



Γραφική Παράσταση 5.2.1

Αυτή η γραφική είναι η γραφική πιθανότητας για κάθε ενδεχόμενο σε κάθε επανάληψη για το κανονικό bargaining game για μέσο όρο 100 επαναλήψεων.

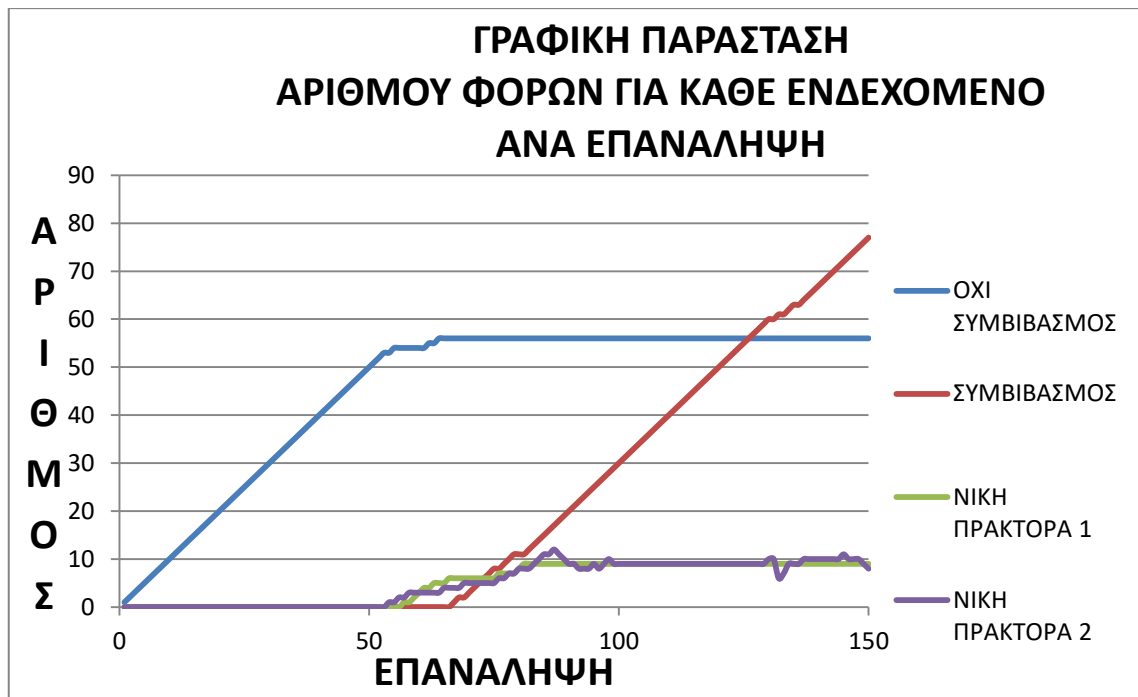
Στην γραφική παράσταση 5.2.1 τρέξαμε το σύστημα για το κανονικό bargaining game με τις παραμέτρους που φαίνονται στο πίνακα 5.2.1:

Αριθμός Επαναλήψεων	150
Ρυθμός μάθησης	0.1
Παράγοντας Έκπτωσης	0.5

Πίνακας 5.2.1

Παράμετροι για αποτελέσματα γραφικής παράστασης 5.2.1

Αυτό το οποίο θέλουμε να μας ενδιαφέρει περισσότερο από τις γραφικές του συστήματος με κανονικό bargaining game είναι ότι οι πράκτορες μαθαίνουν. Όπως φαίνεται και στην γραφική παράσταση 5.2.1 η πιθανότητα συμβιβασμού στις τελευταίες επαναλήψεις φτάνει το 1! Ενώ όπως είναι λογικό στην αρχή είναι 0 καθώς ο κάθε πράκτορας βλέπει το συμφέρον του. Να τονίσουμε επίσης ότι το ποσό συμβιβασμού στις τελευταίες επαναλήψεις είναι 50, άρα το μισό του αρχικού ποσού, που σημαίνει ότι οι πράκτορες συμβιβάζονται με τον μέγιστο τρόπο για το κοινό καλό. Επίσης βλέπουμε ότι οι πιθανότητες για νίκη του κάθε πράκτορα είναι χαμηλές, στα ίδια ποσοστά και ότι αυτές δημιουργούνται καθώς οι παίκτες μαθαίνουν ότι το ενδεχόμενο όχι-συμβιβασμού δεν τους επιφέρει καλό μακροπρόθεσμο αποτέλεσμα, και έτσι τότε αρχίζουν να πειραματίζονται κάνοντας αυτές τις νίκες ο κάθε ένας.



Γραφική Παράσταση 5.2.2

Γραφική παράσταση αριθμού φορών που συμβαίνει κάθε ενδεχόμενο σε κάθε επανάληψη για κανονικό bargaining game 2 πρακτόρων

Στην γραφική παράσταση 5.2.2 βλέπουμε την άνοδο του αριθμού μη-συμβιβασμών στους αρχικούς γύρους και έπειτα την σταθεροποίησή του, και την άνοδο του ενδεχομένου συμβιβασμού στο σημείο σταθεροποίησης του ενδεχομένου όχι-συμβιβασμού. Η γραφική 5.2.2 έχει τις ίδιες παραμέτρους με την γραφική 5.2.1 εκτός το γεγονός ότι η προσομοίωση έχει παιχτεί για 1 φορά, ενώ στην γραφική παράσταση 5.2.1 έχουμε τον μέσο όρο 100 φορών.

Σε αυτές τις 2 γραφικές φαίνεται ότι οι πράκτορες μαθαίνουν από την ενισχυτική μάθηση και συγκεκριμένα με τον αλγόριθμο Q-learning που εφαρμόζεται στο κανονικό bargaining game και μάλιστα τα ποσοστά συμβιβασμού στο τέλος είναι πολύ μεγάλα.

5.3 Αποτελέσματα και ανάλυση Αντιστοίχισης Κυπριακού προβλήματος σε Bargaining Game στο σύστημα

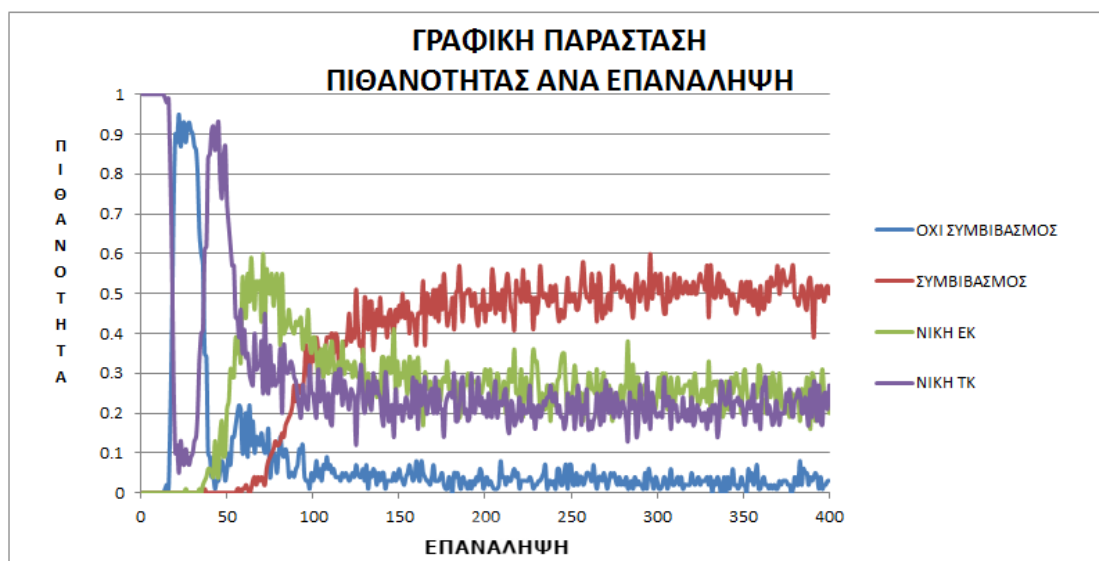
Πάμε τώρα να δούμε και να αναλύσουμε τα αποτελέσματα του συστήματος στην αντιστοίχιση του Κυπριακού Προβλήματος με το bargaining game .

5.3.1 Πράκτορες με αλγόριθμο Q-learning

Οι 2 πράκτορες(EK και TK) σε αυτό το υποκεφάλαιο επιλέγουν τις δράσεις από τις καταστάσεις βάση του αλγορίθμου Q-learning. Θα δούμε διαφορετικά αποτελέσματα και θα πειραματιστούμε με τις παραμέτρους για να βρούμε τις βέλτιστες παραμέτρους για ικανοποιητικά και ρεαλιστικά αποτελέσματα καθώς και να δούμε πώς οι παραμέτροι επηρεάζουν τα αποτελέσματα. Επίσης να αναφέρουμε ότι τα αποτελέσματα είναι μέσος όρος 100 επαναλήψεων σε όλες τις γραφικές που θα ακολουθήσουν .

5.3.1.1 Μελέτη παραμέτρου ρυθμού μάθησης

Κρατώντας τις άλλες παραμέτρους σταθερές θα μελετήσουμε τα αποτελέσματα που προκύπτουν αλλάζοντας το παράμετρο ρυθμού μάθησης.



Γραφική Παράσταση 5.3.1.1α

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.1.1α με πράκτορες που χρησιμοποιούν αλγόριθμο Q-learning με ρυθμό μάθησης 0.8

Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.8
Παράγοντας Έκπτωσης	0.6

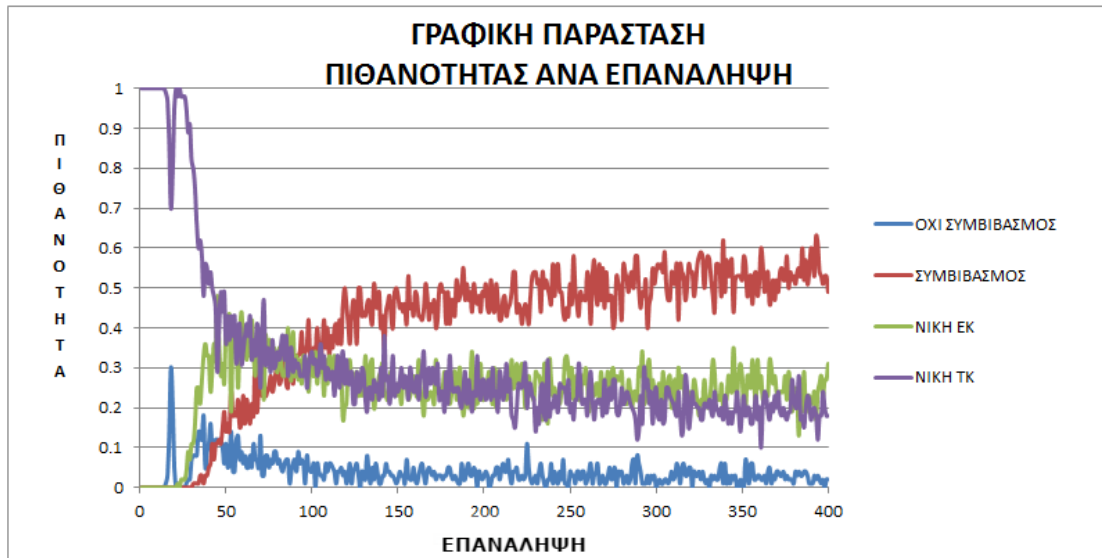
Πίνακας 5.3.1.1α

Όπως φαίνεται στην γραφική παράσταση 5.3.1.1 τα αποτελέσματα δεν είναι τα ίδια με αυτά της γραφικής παράστασης 5.2.1 για τον απλούστατο λόγο ότι εδώ έχουμε πιο ρεαλιστικά δεδομένα που αντιστοιχούν στις πραγματικές συνθήκες του κυπριακού προβλήματος.

Ας ξεκινήσουμε από τις αρχικές επαναλήψεις. Βλέπουμε ξεκάθαρα ότι στις αρχικές επαναλήψεις έχουμε πιθανότητα νίκης ΤΚ πλευράς ίση με 1, που σημαίνει ότι πάντα νικά. Αυτό είναι όντως αλήθεια και το περιμέναμε για τον λόγο ότι όπως έχουμε αναφέρει στο προηγούμενο κεφάλαιο που κάναμε αντιστοίχιση του κυπριακού στο bargaining game στο σύστημα μας, η ΤΚ ξεκινά απορρίπτοντας όλες τις προσφορές τις ΕΚ πλευράς για τον λόγο ότι δεν έχει λόγο να διαπραγματευτεί για το εδαφικό, αφού είδη την κατέχει. Στην συνέχεια βλέπουμε ότι πέφτει το ποσοστό αυτό και αυτό

εξηγείτε στο γεγονός ότι τότε αρχίζουν να λαμβάνουν τόπο οι τιμωρίες πάνω στην δράση της TK πλευράς να απορρίπτει, που όπως έχουμε αναφέρει, οι τιμωρίες αυτές αντιστοιχούν στο γεγονός ότι η EK αρχίζει και προσφέρει ποσοστά διακυβέρνησης και έτσι αναγκάζεται η TK πλευρά να αρχίζει τις διαπραγματεύσεις και να κάνει προσφορές εδάφους. Κάτι άλλο που είναι προφανές και λογικό είναι ότι στις αρχικές επαναλήψεις έχουμε το ενδεχόμενο του όχι συμβιβασμού να φτάνει πιθανότητα 0.9 και αυτό επειδή ο κάθε πράκτορας θέλει να νικήσει και δεν δέχεται προσφορές φτάνοντας στο σημείο το ποσό να μηδενίζεται και να έχουμε όχι συμβιβασμό. Αυτή η πιθανότητα και το ενδεχόμενο του μη συμβιβασμού, κάπου στην επανάληψη 30 αρχίζει και αποδυναμώνεται αφού τότε αρχίσουν οι πράκτορες να αντιλαμβάνονται και να μαθαίνουν ότι δεν τους συμφέρει να μην συμβιβάζονται, έτσι αρχίζουν και «πειραματίζονται» με το να δέχονται πλέον προσφορές και να μην φτάνουμε στο σημείο του όχι συμβιβασμού. Αυτές οι αποδοχές προσφορών, οι οποίες φτάνουν μέχρι και πιθανότητα 0.4 επιφέρουν νίκες και στους 2 πράκτορες και ελαφρώς πιο πολλές στο EK για τον λόγο ότι επειδή το ποσοστό ικανοποίησης του TK είναι 27 (πιο μεγάλο από το 10 του EK) μέχρι να μάθει ο TK πράκτορας να μην κάνει προσφορές μικρότερες του 27 έχουμε τις νίκες αυτές του EK. Όμως μέχρι και τις τελευταίες επαναλήψεις βλέπουμε ότι οι νίκες EK και TK είναι στα ίδια ποσοστά και ότι πιθανότητα έχει πέσει από 0.4 που ήταν πιο πριν σε 0.2. Τέλος βλέπουμε ότι το ενδεχόμενο συμβιβασμού αρχίζει να υπάρχει αφού οι 2 πράκτορες φτάσουν σε αρκετούς όχι συμβιβασμούς. Στην συνέχεια ανεβαίνει εκθετικά και ξεπερνά το ποσοστό νίκης του κάθε πράκτορα ξεχωριστά καθώς οι πράκτορες αρχίζουν και μαθαίνουν ότι ο συμβιβασμός είναι η καλύτερη για το κοινό καλό λύση. Αυτό το ποσοστό του συμβιβασμού φτάνει το 0.5 στις τελευταίες επαναλήψεις και εδώ βλέπουμε μια ρεαλιστικότητα και πιο δυσκολία στο να συμβιβαστούν οι 2 πλευρές σε σύγκριση με το να συμβιβαστούν 2 πράκτορες στο κανονικό bargaining game, και αυτό οφείλεται στην πολυπλοκότητα του σεναρίου και το ότι όντως σε πραγματικά δεδομένα δεν είναι τόσο εύκολοι οι συμβιβασμοί. Όμως μας ικανοποιεί το ποσοστό αυτό γιατί έχουμε ικανοποιητική αύξηση του ποσοστού συμβιβασμού από τις πρώτες επαναλήψεις που είναι 0 μέχρι τις τελευταίες που είναι 0.5.

Καταλήγουμε στο συμπέρασμα ότι η γραφική παράσταση αυτή ανταποκρίνεται στην πραγματικότητα και τα αποτελέσματα της είναι πολύ λογικά για το κυπριακό πρόβλημα.



Γραφική Παράσταση 5.3.1.1β

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.1.1β με πράκτορες που χρησιμοποιούν αλγόριθμο Q-learning με ρυθμό μάθησης 0.3

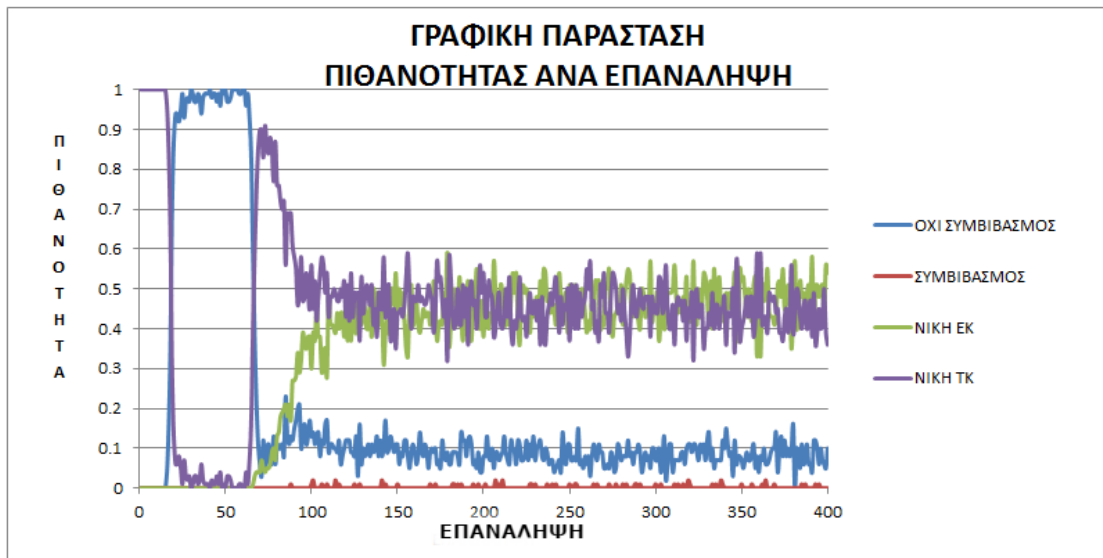
Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.3
Παράγοντας Έκπτωσης	0.6

Πίνακας 5.3.1.1β

Στην γραφική παράσταση 5.3.1.1β δεν βλέπουμε κάποια κύρια διαφορά σε σύγκριση με την γραφική παράσταση 5.3.1.1α παρά μόνο το γεγονός ότι η πιθανότητα να συμβιβαστούν οι 2 πράκτορες φτάνει σχεδόν το 0.55, ελαφρώς πιο ψηλό από την γραφική 5.3.1.1α που είχαμε πολύ μεγαλύτερο ρυθμό μάθησης. Αυτό οφείλεται στο γεγονός ότι οι 2 πράκτορες με μεγάλο ρυθμό μάθησης μαθαίνουν μηχανικά τις καλύτερες δράσεις.

5.3.1.2 Μελέτη παραμέτρου έκπτωσης

Κρατώντας τις άλλες παραμέτρους σταθερές θα μελετήσουμε τα αποτελέσματα που προκύπτουν αλλάζοντας την παράμετρο έκπτωσης.



Γραφική Παράσταση 5.3.1.2α

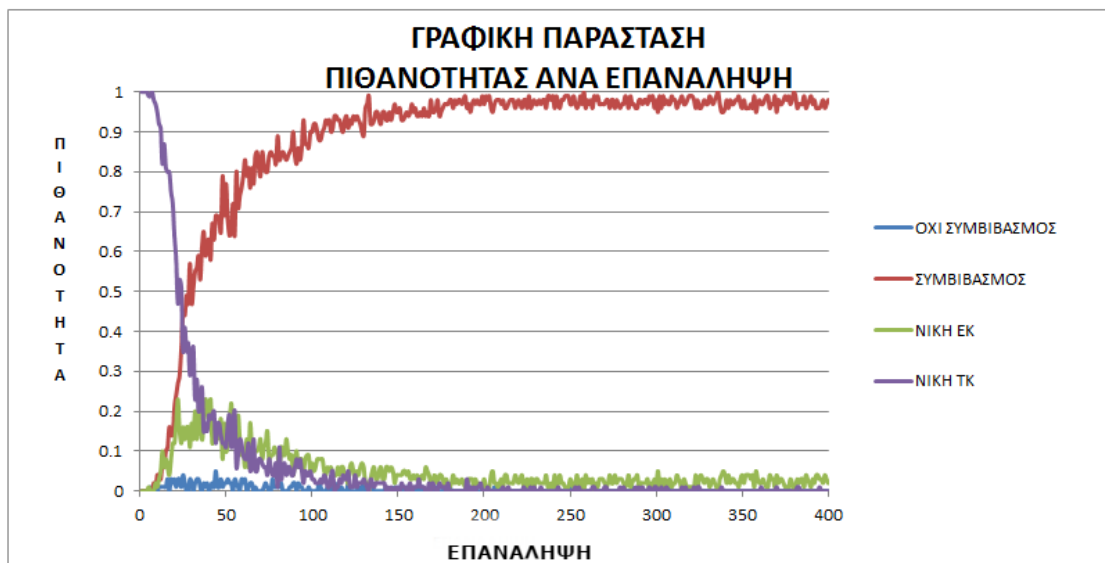
Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.1.2α με πράκτορες που χρησιμοποιούν αλγόριθμο Q-learning με παράγοντα έκπτωσης 0.9

Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.5
Παράγοντας Έκπτωσης	0.9

Πίνακας 5.3.1.2α

Όπως βλέπουμε στην γραφική παράσταση 5.3.1.2α όπου έχουμε αυξήσει το ρυθμό έκπτωσης τα αποτελέσματα είναι πολύ χαμηλά. Το ποσοστό συμβιβασμού είναι πολύ χαμηλό, γύρω στο 0.1 καθώς το ποσοστό όχι συμβιβασμού αν και δεν έχει τροποποιηθεί από τις προηγούμενες γραφικές, πάλι είναι ψηλότερο από το ποσοστό συμβιβασμού. Αυτό βέβαια έχει σαν συνέπεια την άλλη φανερά διαφορά από τις προηγούμενες διαφορές, δηλαδή ότι τα ποσοστά νίκης για το κάθε πράκτορα είναι πιο

ψηλά από όλα τα άλλα ενδεχόμενα καθώς φτάνουν τη πιθανότητα του 0.5. Αυτό ίσως οφείλετε στο γεγονός ότι αυτή η στρατηγική των πρακτόρων να υπολογίζουν την τελική αμοιβή περισσότερο από την αμοιβή της επερχόμενης πράξης να μην τους βοηθά να βρουν το καλύτερο κοινό αποτέλεσμα που είναι ο συμβιβασμός. Με λίγα λόγια οι πράκτορες βλέπουν μόνο μακρυπρόθεσμα, το οποίο δεν τους βοηθά, διότι και σε ένα αγώνα σκακιού παραδείγματος χάρη, αν ένας παίκτης θέλει να νικήσει πρέπει να κάνει και δράσεις που του επιφέρουν αμοιβή και απο την παρούσα χρονική στιγμή, όχι μόνο για το τέλος, δηλαδή να δεί πως να νικήσει πιόνια για να φτάσει στην νίκη.



Γραφική Παράσταση 5.3.1.2β

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.1.2β με πράκτορες που χρησιμοποιούν αλγόριθμο Q-learning με παράγοντα έκπτωσης 0.1

Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.5
Παράγοντας Έκπτωσης	0.1

Πίνακας 5.3.1.2β

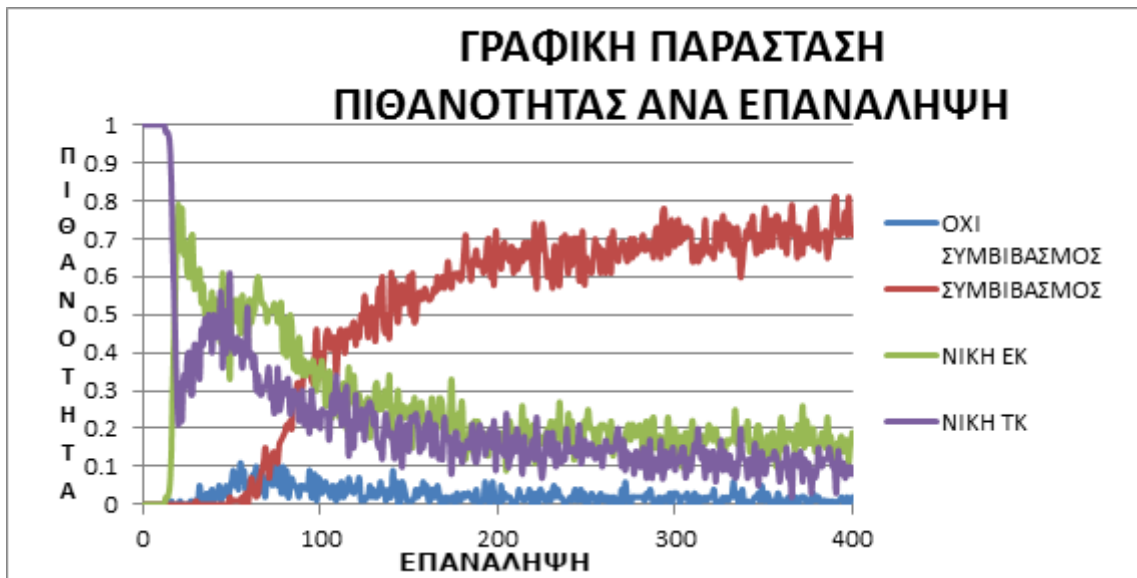
Σε αντίθεση με την προηγούμενη γραφική παράσταση 5.3.1.2α, στην γραφική παράσταση 5.3.1.2β όπου έχουμε μειώσει το παράγοντα έκπτωσης και ο πράκτορας

βλέπει μυωπικά, φαίνεται πώς αυτή η άπληστη στρατηγική του επιφέρει μεγαλύτερη πιθανότητα συμβιβασμού και από την πρώτη γραφική. Βεβαίως τα ποσοστά αυτά δεν ανταποκρίνονται πλήρως στα πραγματικά δεδομένα διότι ένας πράκτορας έχει και μακροπρόθεσμους στόχους και επίσης η πιθανότητα συμβιβασμού 1 είναι σχεδόν αδύνατη να επιτευχθεί.

5.3.1.2.1 Μελέτη με διαφορετικούς παραμέτρους

έκπτωσης για κάθε πράκτορα

Σε αυτό το υποκεφάλαιο θα μελετήσουμε τα αποτελέσματα που προκύπτουν αλλάζοντας τις παραμέτρους έκπτωσης για κάθε πράκτορα, κρατώντας τις άλλες παραμέτρους σταθερές. Δηλαδή θα δούμε κατά πόσο επηρεάζει τα αποτελέσματα αν ένας πράκτορας σκέπτεται μακροπρόθεσμα και ο άλλος μυωπικά και για τις 2 πλευρές.



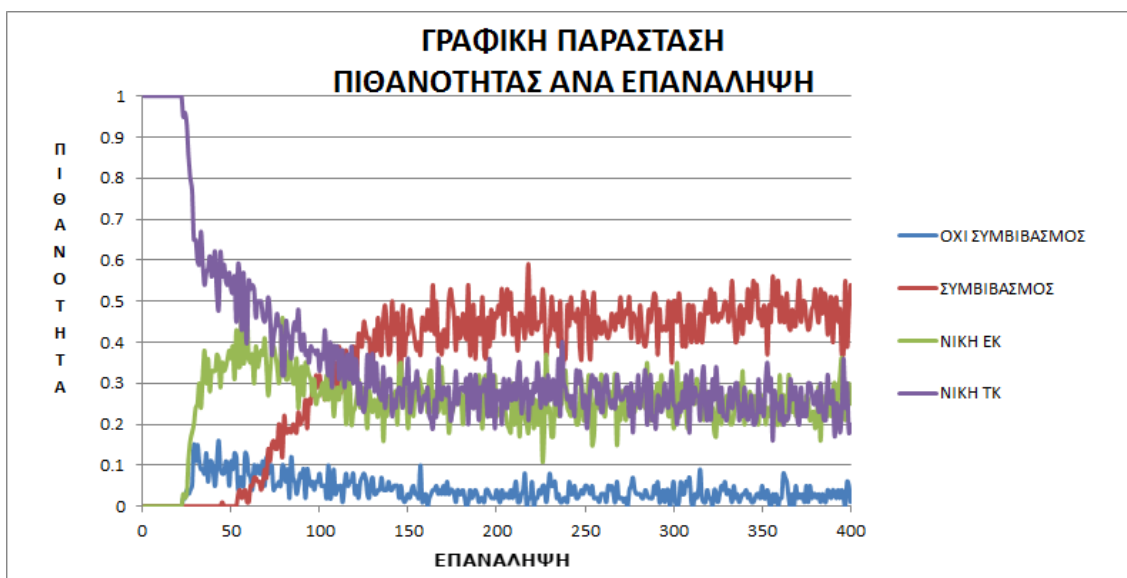
Γραφική Παράσταση 5.3.1.2.1α

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.1.2.1α με πράκτορες που χρησιμοποιούν αλγόριθμο Q-learning με παράγοντα έκπτωσης 0.2 για το ΤΚ πράκτορα και παράγοντα έκπτωσης 0.9 για τον ΕΚ

Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης (και για τους 2)	0.5
Παράγοντας Έκπτωσης ΕΚ	0.9
Παράγοντας Έκπτωσης ΤΚ	0.2

Πίνακας 5.3.1.2.1α

Αλλάζοντας το παράγοντα έκπτωσης για τους 2 πράκτορες, θέτοντας τον σε 0.9 για το ΕΚ και 0.2 για ΤΚ παρατηρούμε ότι ο ΕΚ πράκτορα έχει περισσότερη πιθανότητα να κερδίσει. Αυτό οφείλεται στο γεγονός ότι ο ΕΚ με παράγοντα μάθησης 0.9 βλέπει πιο μακροπρόθεσμα και όχι μυωπικά, δηλαδή βλέπει αμοιβές προς το τέλος κάθε επανάληψης και όχι την προσωρινή κάθε γύρο με αποτέλεσμα να έχει περισσότερες νίκες.



Γραφική Παράσταση 5.3.1.2.1β

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.1.2.1β με πράκτορες που χρησιμοποιούν αλγόριθμο Q-learning με παράγοντα έκπτωσης 0.9 για το ΤΚ πράκτορα και παράγοντα έκπτωσης 0.1 για τον ΕΚ

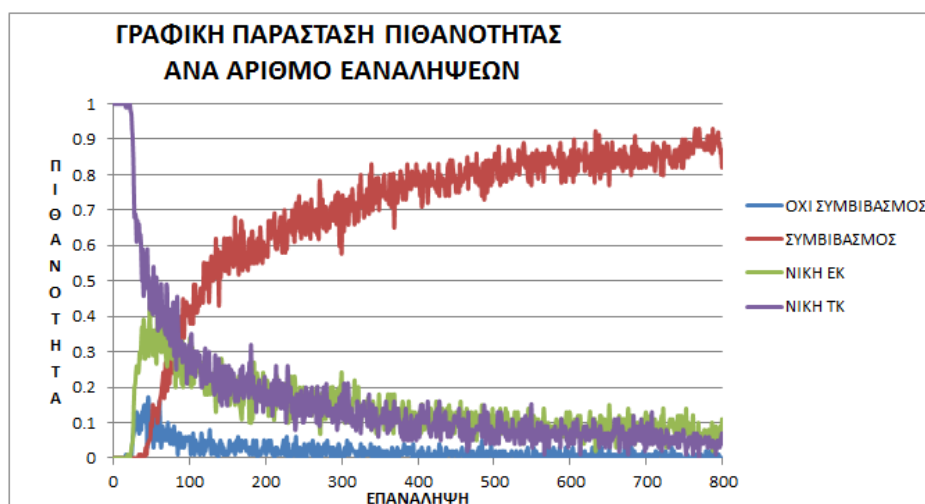
Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης (και για τους 2)	0.5
Παράγοντας Έκπτωσης ΕΚ	0.1
Παράγοντας Έκπτωσης ΤΚ	0.9

Πίνακας 5.3.1.2.1β

Αλλάζοντας το παράγοντα έκπτωσης για τους 2 πράκτορες, θέτοντας τον σε 0.9 για το ΤΚ και 0.1 για ΕΚ παρατηρούμε ότι ο ΤΚ πράκτορας έχει περισσότερη πιθανότητα να κερδίσει. Αυτό οφείλεται στο γεγονός ότι ο ΤΚ με παράγοντα μάθησης 0.9 βλέπει πιο μακροπρόθεσμα και όχι μυωπικά, δηλαδή βλέπει αμοιβές προς το τέλος κάθε επανάληψης και όχι την προσωρινή κάθε γύρο με αποτέλεσμα να έχει περισσότερες νίκες. Βέβαια ο αριθμός των νικών είναι ελαφρώς μικρότερος από τον αριθμό των νικών που θα είχε ο ΕΚ με μεγαλύτερο παράγοντα έκπτωσης, όπως βλέπουμε στην προηγούμενη γραφική 5.3.1.2.1α και εδώ παίζει ρόλο οι τιμωρίες που δέχεται ο ΤΚ στην αρχή των επαναλήψεων.

5.3.1.3 Μελέτη παραμέτρου επαναλήψεων

Κρατώντας τις άλλες παραμέτρους σταθερές θα μελετήσουμε τα αποτελέσματα που προκύπτουν αλλάζοντας το παράμετρο των επαναλήψεων.



Γραφική Παράσταση 5.3.1.3

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.1.3 με πράκτορες που χρησιμοποιούν αλγόριθμο Q-learning με αριθμό επαναλήψεων 800

Αριθμός Επαναλήψεων	800
Ρυθμός μάθησης	0.5
Παράγοντας Έκπτωσης	0.6

Πίνακας 5.3.1.3

Όπως ήταν αναμενόμενο με πιο πολλές επαναλήψεις και για τον λόγο ότι έχουμε επαναλαμβανόμενο bargaining game που σημαίνει ότι οι πράκτορες διατηρούν την γνώση από προηγούμενες επαναλήψεις, η πιθανότητα συμβιβασμού στις τελευταίες επαναλήψεις όπως βλέπουμε και στην γραφική παράσταση 5.3.1.3 έχει φτάσει το 0.9 σε σύγκριση με τη πιθανότητα 0.6 που είχαμε για 400 επαναλήψεις. Κάτι που αξίζει να αναφέρουμε για αυτήν την γραφική επίσης είναι ότι βλέπουμε πως η πιθανότητα νίκης κάποιου πράκτορα μειώνεται μέχρι 0.1 σε σύγκριση με αριθμό επαναλήψεων 400, και ότι η πιθανότητα όχι-συμβιβασμού είναι σχεδόν μηδενική. Επίσης όπως είναι προφανές για πιο λίγες επαναλήψεις, για παράδειγμα 200, όπως βλέπουμε στην ίδια γραφική η πιθανότητα του ενδεχομένου όχι συμβιβασμού είναι πιο μεγάλη για το λόγο ότι

χρειάζονται ακόμη περισσότερες επαναλήψεις για να μάθουν ότι ο συμβιβασμός είναι η καλύτερη κοινή λύση.

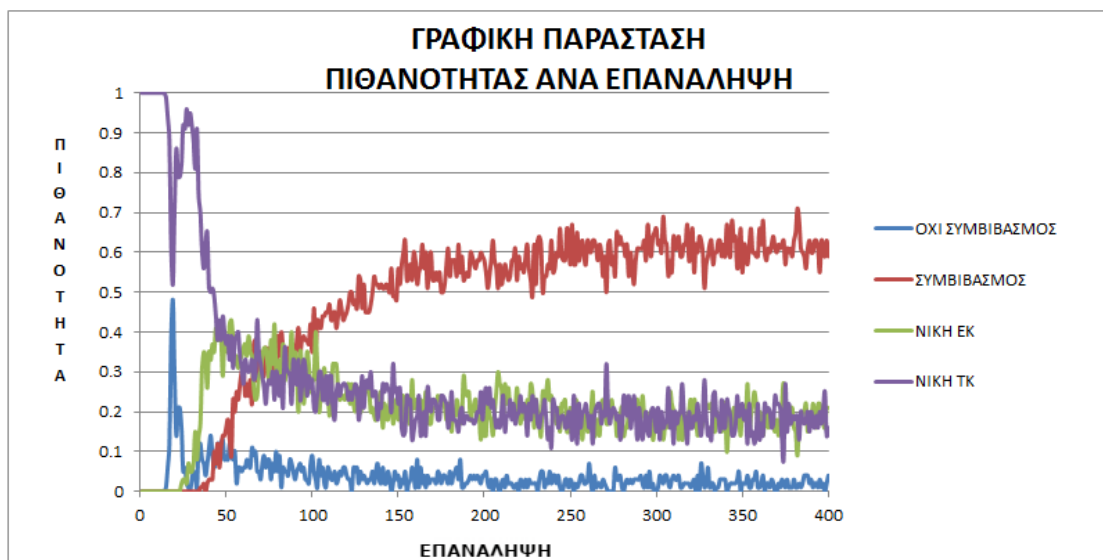
5.3.1.4 Βέλτιστες παραμέτροι για αλγόριθμο Q-learning

Σε αυτό το υποκεφάλαιο θα δούμε ποιές παραμέτροι επιφέρουν πιο ινακοποιητικά και πιο ρεαλιστικά αποτελέσματα για τον αλγόριθμο Q-learning μετα απο τους πειραματισμούς που έχουμε κάνει.

Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.5
Παράγοντας Έκπτωσης	0.6

Πίνακας 5.3.1.4

Παράμετροι για αποτελέσματα γραφικής παράστασης 5.3.1.4



Γραφική Παράσταση 5.3.1.4

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.1.4 με πράκτορες που χρησιμοποιούν αλγόριθμο Q-learning

Στη γραφική παράσταση 5.3.1.4 βλέπουμε τις παραμέτρους οι οποίες μετά απο δοκιμές επιφέρουν τα πιο ρεαλιστικά και ικανοποιητικά αποτελέσματα. Αυτές οι παραμέτροι είναι 0.5 για ρυθμό μάθησης και 0.6 για παράγοντα έκπτωσης. Όπως αρχικά βλέπουμε, οι παραμέτροι έχουν τις βέλτιστες τιμές τους κάπου στην μέση των ορίων τους (0-1) . Αυτό είναι λογικό καθώς όσο αφορά τη παράμετρο ρυθμού μάθησης, ένας πράκτορας δεν πρέπει να μαθαίνει ούτε μηχανικά ούτε εντελώς τυχαία, αλλά με κάτι ενδιάμεσο. Επίσης όσο αφορά το παράγοντα έκπτωσης ισχύει το ίδιο , διότι ένας πράκτορας πρέπει να βλέπει και μυωπικά και μακροπρόθεσμα για να έχει καλύτερα αποτελέσματα. Με αυτές τις τιμές για τις παραμέτρους παίρνουμε αποτελέσματα τα οποία είναι ικανοποιητικά (0.6 πιθανότητα συμβιβασμού) όμως και ρεαλιστικά καθώς βλέπουμε ότι η πιθανότητα συμβιβασμού δέν είναι σχεδόν 1 όπως για κάποιες παραμέτρους, επειδή είναι φυσιολογικό 2 πράκτορες να μην συμβιβάζονται πάντα. Τέλος όσο αφορά τα υπόλοιπα ενδεχόμενα βλέπουμε ότι ανταποκρίνονται στο κυπριακό, καθώς πάλι στις αρχικές επαναλήψεις έχει πιθανότητα να κερδίσει ο TK πράκτορας με 1 διότι όπως είπαμε κατέχει τις 37 μονάδες απο τις 37 που θέλουν να μοιράσουν, στην συνέχεια αρχίζει και διαπραγματεύεται όταν ο EK πράκτορας δίνει ποσοστά διακυβέρνησης τα οποία εκφράζονται σαν έμμεση τιμωρία και τότε οι 2 πράκτορες αρχίζουν να μην συμβιβάζονται επειδή ο καθένας θέλει περισσότερα για τον εαυτό του και βλέπουν ατομικά, επομένως το ποσό γίνεται 0 αφού δεν αποδέχεται καποιος προσφορά. Έπειτα αρχίζουν να πειραματίζονται κάνοντας νίκες και τέλος η πιθανότητα συμβιβασμού υπερτερεί των άλλων ενδεχομένων.

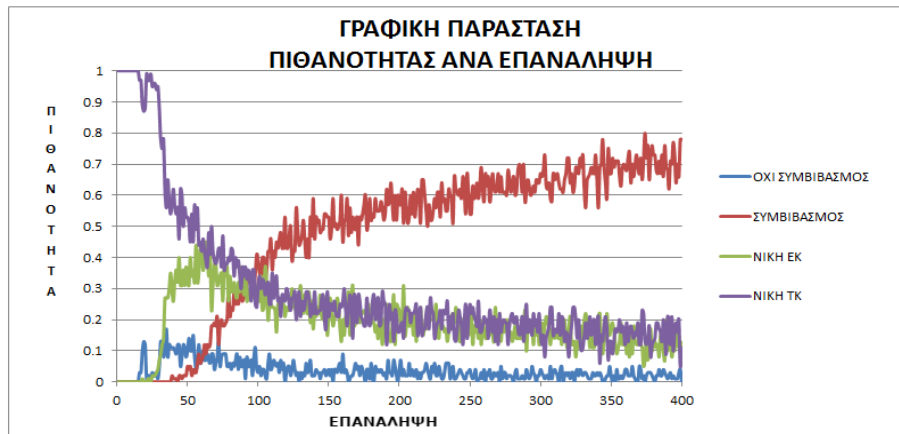
5.3.2 Πράκτορες με αλγόριθμο Sarsa με e-greedy τακτική

Οι 2 πράκτορες(EK και TK) σε αυτό το υποκεφάλαιο επιλέγουν τις δράσεις από τις καταστάσεις βάση του αλγορίθμου SARSA με τακτική e-greedy. Δηλαδή με μερικώς άπληστη τακτική, που σημαίνει ότι με πιθανότητα ϵ θα επιλέγουν τυχαία κάποια δράση απο μια κατάσταση και εμ την υπόλοιπη πιθανότητα $1-\epsilon$ θα εκμεταλεύονται την γνώση τους. Θα δούμε διαφορετικά αποτελέσματα και θα πειραματιστούμε με τις παραμέτρους για να βρούμε τις βέλτιστες παραμέτρους για ικανοποιητικά και ρεαλιστικά αποτελέσματα καθώς και πώς οι παραμέτροι επηρεάζουν τα αποτελέσματα.

Επίσης να αναφέρουμε ότι τα αποτελέσματα είναι μέσος όρος 100 επαναλήψεων σε όλες τις γραφικές που θα ακολουθήσουν .

5.3.2.1 Μελέτη παραμέτρου ρυθμού μάθησης

Κρατώντας τις άλλες παραμέτρους σταθερές θα μελετήσουμε τα αποτελέσματα που προκύπτουν αλλάζοντας το παράμετρο ρυθμού μάθησης.



Γραφική Παράσταση 5.3.2.1α

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.2.1α με πράκτορες που χρησιμοποιούν αλγόριθμο SARSA με ρυθμό μάθησης 0.2

Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.2
Παράγοντας Έκπτωσης	0.6
Πιθανότητα εξερεύνησης (e)	0.3

Πίνακας 5.3.2.1α.

Παράμετροι για αποτελέσματα γραφικής παράστασης 5.3.2.1α

Όπως φαίνεται στην γραφική παράσταση 5.3.2.1 τα αποτελέσματα δεν είναι τα ίδια με αυτά της γραφικής παράστασης 5.2.1 για τον απλούστατο λόγο ότι εδώ έχουμε πιο

ρεαλιστικά δεδομένα που αντιστοιχούν στις πραγματικές συνθήκες του κυπριακού προβλήματος.

Ας ξεκινήσουμε από τις αρχικές επαναλήψεις. Βλέπουμε ξεκάθαρα ότι στις αρχικές επαναλήψεις έχουμε πιθανότητα νίκης TK πλευράς ίση με 1, που σημαίνει πάντα νικά. Αυτό είναι όντως αλήθεια και το περιμέναμε για τον λόγο ότι όπως έχουμε αναφέρει στο προηγούμενο κεφάλαιο που κάναμε αντιστοίχιση του κυπριακού στο bargaining game στο σύστημα μας, η TK ξεκινά απορρίπτοντας όλες τις προσφορές τις EK πλευράς για τον λόγο ότι δεν έχει λόγο να διαπραγματευτεί για την γη, αφού είδη την κατέχει. Στην συνέχεια βλέπουμε ότι πέφτει το ποσοστό αυτό και αυτό εξηγείτε στο γεγονός ότι τότε αρχίζουν να λαμβάνουν τόπο οι τιμωρίες πάνω στην δράση της TK πλευράς να απορρίπτει, που όπως έχουμε αναφέρει, οι τιμωρίες αυτές αντιστοιχούν στο γεγονός ότι η EK αρχίζει και προσφέρει ποσοστά διακυβέρνησης και έτσι αναγκάζεται η TK πλευρά να αρχίζει τις διαπραγματεύσεις και να κάνει προσφορές εδάφους. Κάτι άλλο που είναι προφανές και λογικό είναι ότι στις αρχικές επαναλήψεις έχουμε το ενδεχόμενο του όχι συμβιβασμού να φτάνει πιθανότητα 0.1 και αυτό επειδή ο κάθε πράκτορας θέλει να νικήσει και δεν δέχεται προσφορές φτάνοντας στο σημείο το ποσό να μηδενίζεται και να έχουμε όχι συμβιβασμό. Αυτή η πιθανότητα και το ενδεχόμενο του μη συμβιβασμού ,κάπου στην επανάληψη 50 αρχίζει και αποδυναμώνεται αφού τότε αρχίσουν οι πράκτορες να αντιλαμβάνονται και να μαθαίνουν ότι δεν τους συμφέρει να μην συμβιβάζονται, έτσι αρχίζουν και «πειραματίζονται» με το να δέχονται πλέον προσφορές και να μην φτάνουμε στο σημείο του όχι συμβιβασμού. Αυτές οι αποδοχές προσφορών οποίες φτάνουν μέχρι και πιθανότητα 0.3 επιφέρουν νίκες και στους 2 πράκτορες και ελαφρώς πιο πολλές στο EK για τον λόγο ότι επειδή το ποσοστό ικανοποίησης του TK είναι 27 (πιο μεγάλο από το 10 του EK) μέχρι να μάθει ο TK πράκτορας να μην κάνει προσφορές μικρότερες του 27 έχουμε τις νίκες αυτές του EK. Όμως μέχρι και τις τελευταίες επαναλήψεις βλέπουμε ότι οι νίκες EK και TK είναι στα ίδια ποσοστά και ότι η πιθανότητα έχει πέσει από 0.3 που ήταν πιο πριν σε 0.12. Τέλος βλέπουμε ότι το ενδεχόμενο συμβιβασμού αρχίζει να υπάρχει αφού οι 2 πράκτορες φτάσουν σε αρκετούς όχι συμβιβασμούς. Στην συνέχεια ανεβαίνει εκθετικά και ξεπερνά το ποσοστό νίκης του κάθε πράκτορα ξεχωριστά καθώς οι πράκτορες αρχίζουν και μαθαίνουν ότι ο συμβιβασμός είναι η καλύτερη κοινή λύση. Αυτό το ποσοστό του συμβιβασμού φτάνει το 0.7 στις τελευταίες επαναλήψεις και εδώ

βλέπουμε μια ρεαλιστικότητα και πιο δυσκολία στο να συμβιβαστούν οι 2 πλευρές σε σύγκριση με το να συμβιβαστούν 2 πράκτορες στο κανονικό bargaining game, και αυτό οφείλεται στην πολυπλοκότητα του σεναρίου και το ότι όντως σε πραγματικά δεδομένα δεν είναι τόσο εύκολοι οι συμβιβασμοί. Όμως μας ικανοποιεί το ποσοστό αυτό γιατί έχουμε ικανοποιητική αύξηση του ποσοστού συμβιβασμού από τις πρώτες επαναλήψεις που είναι 0 μέχρι τις τελευταίες που είναι 0.7.

Καταλήγουμε στο συμπέρασμα ότι η γραφική παράσταση αυτή ανταποκρίνεται στην πραγματικότητα και τα αποτελέσματα της είναι πολύ λογικά και για το κυρίακό πρόβλημα.



Γραφική Παράσταση 5.3.2.1β

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.2.1β με πράκτορες που χρησιμοποιούν αλγόριθμο SARSA με ρυθμό μάθησης 0.9

Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.9
Παράγοντας Έκπτωσης	0.6
Πιθανότητα εξερεύνησης (e)	0.3

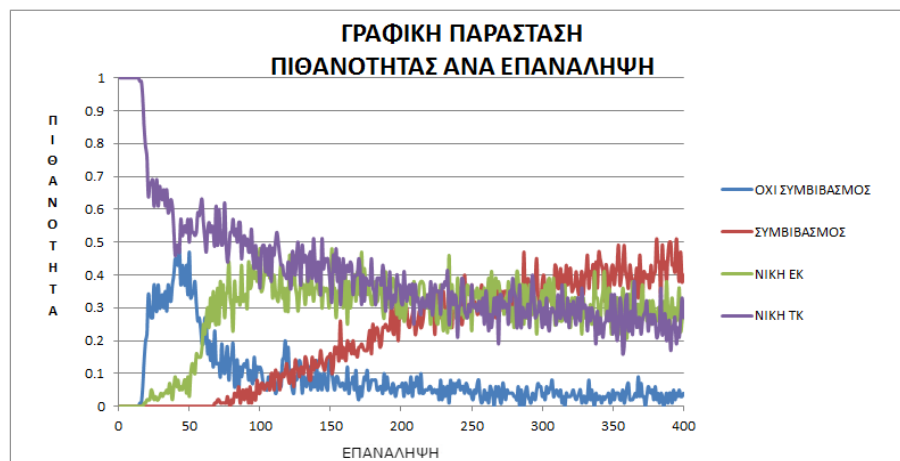
Πίνακας 5.3.2.1β.

Παράμετροι για αποτελέσματα γραφικής παράστασης 5.3.2.1β

Σε αυτήν την γραφική παράσταση όπως βλέπουμε τα αποτελέσματα είναι καλύτερα από την γραφική παράσταση 5.3.2.1α. Η πιθανότητα συμβιβασμού φτάνει σχεδόν 1 στις τελευταίες επαναλήψεις και οι νίκες των 2 πρακτόρων σχεδόν 0. Αυτό ίσως επιτυγχάνεται για το λόγο ότι ενώ αυξήσαμε το ρυθμό μάθησης και επειδή σε αυτό το αλγόριθμο προσθέσαμε πιθανότητα εξερεύνησης, τυγχάνει στις πρώτες επαναλήψεις να εξερευνήσει καινούργιες καταστάσεις οι οποίες είχαν επιφέρει καλύτερο κέρδος, και με την βοήθεια της εξίσωσης 2.2 όπου ο ρυθμός μάθησης βοηθά να ανεβεί περισσότερο η συνάρτηση αξίας η πιθανότητα συμβιβασμού ανεβαίνει ποιο πολύ. Βεβαίως τα αποτελέσματα αυτά δεν ανταποκρίνονται στην πραγματικότητα διότι πιθανότητα συμβιβασμού 1 είναι σχεδόν αδύνατο να υπάρξει, επομένως δεν θα θεωρήσουμε σαν βέλτιστες αυτές τις παραμέτρους

5.3.2.2 Μελέτη παραμέτρου έκπτωσης

Κρατώντας τις άλλες παραμέτρους σταθερές θα μελετήσουμε τα αποτελέσματα που προκύπτουν αλλάζοντας τη παράμετρο έκπτωσης.



Γραφική Παράσταση 5.3.2.2α

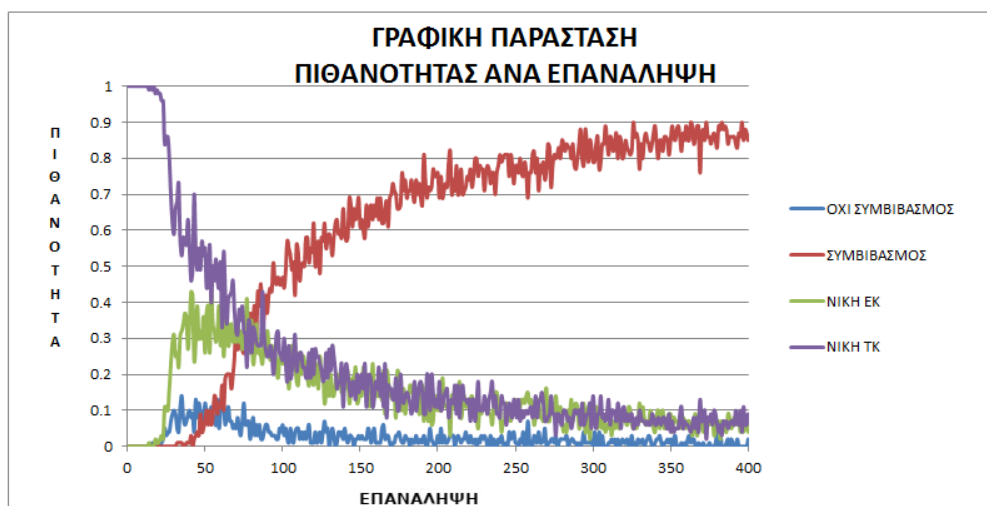
Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.2.2α με πράκτορες που χρησιμοποιούν αλγόριθμο SARSA με παράγοντα έκπτωσης 0.8

Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.5
Παράγοντας Έκπτωσης	0.8
Πιθανότητα εξερεύνησης (e)	0.3

Πίνακας 5.3.2.2α.

Παράμετροι για αποτελέσματα γραφικής παράστασης 5.3.2.3α

Όπως βλέπουμε στην γραφική παράσταση 5.3.2.2α όπου έχουμε αυξήσει το ρυθμό έκπτωσης τα αποτελέσματα είναι πολύ χαμηλά. Το ποσοστό συμβιβασμού είναι πιο χαμηλό, γύρω στο 0.5 καθώς το ποσοστό όχι συμβιβασμού είναι ελαφρώς ψηλότερο από το ποσοστό συμβιβασμού των προηγούμενων γραφικών. Αυτό βέβαια έχει σαν συνέπεια την άλλη φανερή διαφορά από τις προηγούμενες διαφορές, δηλαδή ότι τα ποσοστά νίκης για το κάθε πράκτορα είναι πιο ψηλά από όλα τα άλλα ενδεχόμενα καθώς φτάνουν τη πιθανότητα του 0.3. Αυτό ίσως οφείλετε στο γεγονός ότι αυτή η στρατηγική των πρακτόρων να υπολογίζουν την τελική αμοιβή περισσότερο από την αμοιβή της επερχόμενης πράξης να μην τους βοηθά να βρουν το καλύτερο κοινό αποτέλεσμα που είναι ο συμβιβασμός.



Γραφική Παράσταση 5.3.2.2β

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.2.2β με πράκτορες που χρησιμοποιούν αλγόριθμο SARSA με παράγοντα έκπτωσης 0.3

Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.5
Παράγοντας Έκπτωσης	0.3
Πιθανότητα εξερεύνησης (e)	0.3

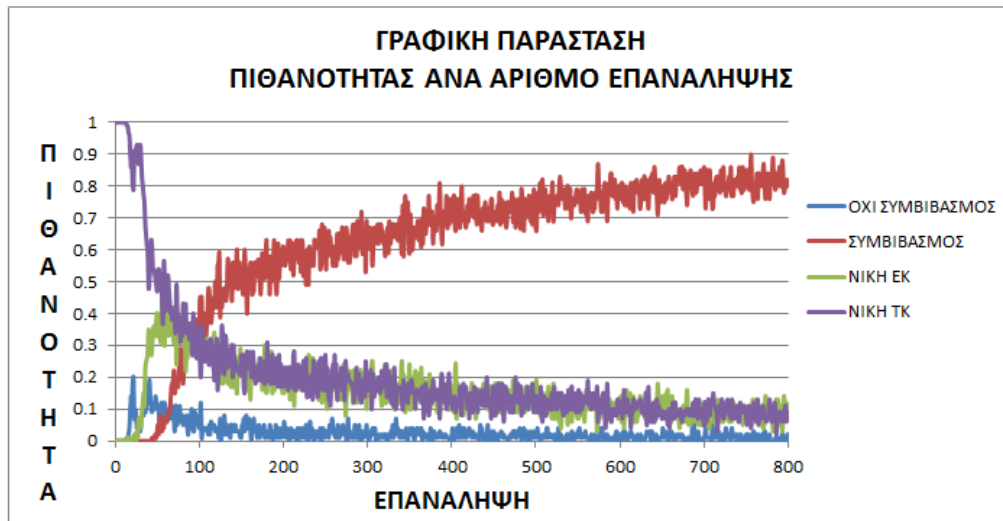
Πίνακας 5.3.2.2β.

Παράμετροι για αποτελέσματα γραφικής παράστασης 5.3.2.2β

Σε αντίθεση με την προηγούμενη γραφική παράσταση 5.3.2.2α, στην γραφική παράσταση 5.3.2.2β όπου έχουμε μειώσει το παράγοντα έκπτωσης και ο πράκτορας βλέπει μυωπικά, φαίνεται πώς αυτή η άπληστη στρατηγική του επιφέρει μεγαλύτερη πιθανότητα συμβιβασμού. Βεβαίως τα ποσοστά αυτά δεν ανταποκρίνονται πλήρως στα πραγματικά δεδομένα διότι ένας πράκτορας έχει και μακροπροθέσμους στόχους. Επίσης η πιθανότητα συμβιβασμού 1 είναι σχεδόν αδύνατη να επιτευχθεί σε πραγματικά δεδομένα. Επομένως δεν θα θεωρήσουμε αυτές τις παραμέτρους βέλτιστες επειδή δεν επιφέρουν ρεαλιστικά αποτελέσματα.

5.3.2.3 Μελέτη παραμέτρου επαναλήψεων

Κρατώντας τις άλλες παραμέτρους σταθερές θα μελετήσουμε τα αποτελέσματα που προκύπτουν τρέχοντας το σύστημα για περισσότερες επαναλήψεις.



Γραφική Παράσταση 5.3.2.3

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.2.3 με πράκτορες που χρησιμοποιούν αλγόριθμο SARSA με αριθμό επαναλήψεων 800

Αριθμός Επαναλήψεων	800
Ρυθμός μάθησης	0.5
Παράγοντας Έκπτωσης	0.6
Πιθανότητα εξερεύνησης (e)	0.3

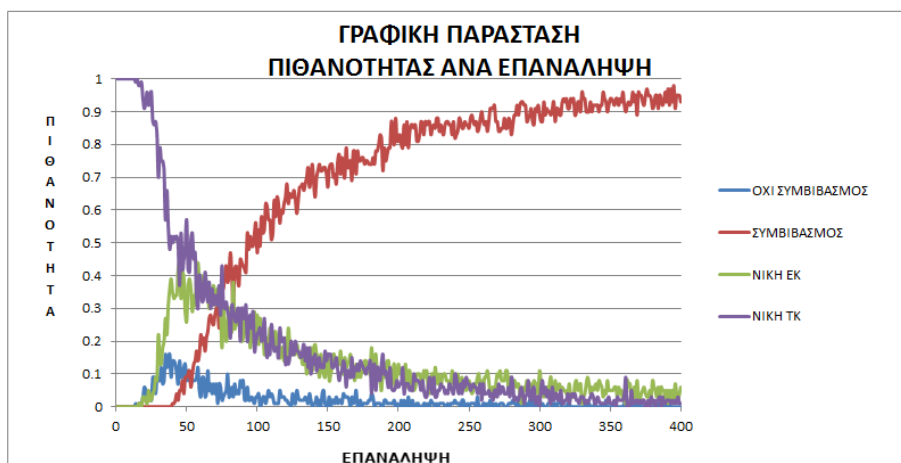
Πίνακας 5.3.2.3.

Παράμετροι για αποτελέσματα γραφικής παράστασης 5.3.2.3

Όπως ήταν αναμενόμενο με πιο πολλές επαναλήψεις και για τον λόγο ότι έχουμε επαναλαμβανόμενο bargaining game που σημαίνει ότι οι πράκτορες διατηρούν γνώση από προηγούμενες επαναλήψεις, η πιθανότητα συμβιβασμού στις τελευταίες επαναλήψεις όπως βλέπουμε και στην γραφική παράσταση 5.3.2.3 έχει φτάσει το 0.85 σε σύγκριση με τη πιθανότητα 0.7 που είχαμε για 400 επαναλήψεις. Κάτι που αξίζει να αναφέρουμε για αυτήν την γραφική επίσης είναι ότι βλέπουμε πως η πιθανότητα νίκης κάποιου πράκτορα μειώνεται μέχρι 0.1 σε σύγκριση με αριθμό επαναλήψεων 400, και ότι η πιθανότητα όχι-συμβιβασμού είναι σχεδόν μηδενική. Επίσης όπως είναι προφανές για πιο λίγες επαναλήψεις, για παράδειγμα 200, όπως βλέπουμε στην ίδια γραφική η πιθανότητα αυτή του ενδεχομένου όχι συμβιβασμού είναι πιο μεγάλη για το λόγο ότι χρειάζονται ακόμη επαναλήψεις για να μάθουν ότι ο συμβιβασμός είναι η καλύτερη κοινή λύση.

5.3.2.4 Μελέτη παραμέτρου e

Κρατώντας τις άλλες παραμέτρους σταθερές θα μελετήσουμε τον παράγοντα της πιθανότητας εξερεύνησης e .



Γραφική Παράσταση 5.3.2.4α

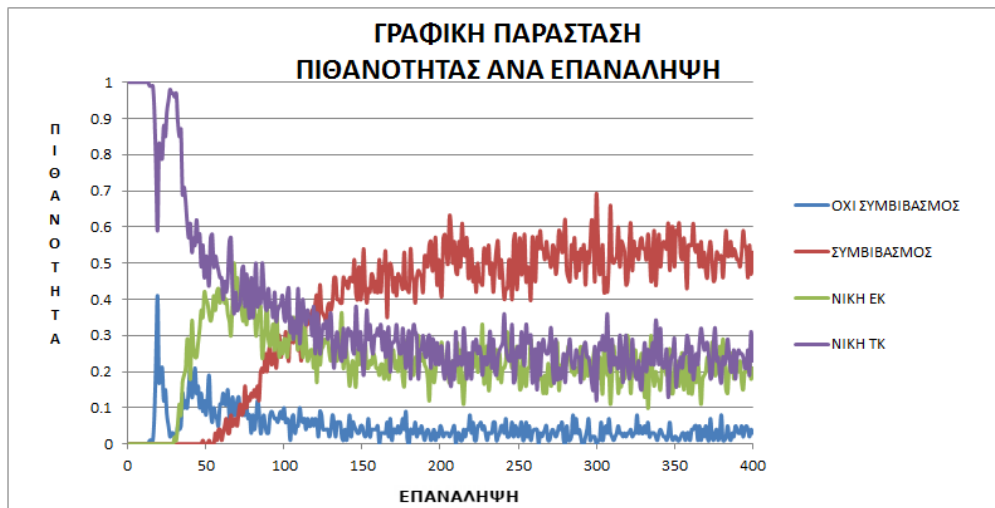
Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.2.4α με πράκτορες που χρησιμοποιούν αλγόριθμο SARSA με πιθανότητα εξερεύνησης 0.8

Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.5
Παράγοντας Έκπτωσης	0.6
Πιθανότητα εξερεύνησης (e)	0.8

Πίνακας 5.3.2.4α.

Παράμετροι για αποτελέσματα γραφικής παράστασης 5.3.2.4α

Βλέπουμε από την γραφική παράσταση 5.3.2.4α ότι με πιθανότητα εξερεύνησης μεγαλύτερη , έχουμε πιο μεγάλη πιθανότητα συμβιβασμού. Βέβαια αυτό δεν είναι πάντα απόλυτο, καθώς για πιο ρεαλιστικά και πιο σίγουρα αποτελέσματα είναι καλύτερο να μην είναι τόσο μεγάλη, έτσι ώστε να εκμεταλλευόμαστε και την γνώση. Στο σύστημα μας , αυτό πιθανώς οφείλεται στο γεγονός ότι λόγω της πιθανότητας αυτής που είναι πιο μεγάλη, στις αρχικές επαναλήψεις μπορεί οι 2 πράκτορες να μην ακολουθούν την προτεραιότητα της λογικής, δηλαδή να δρουν εγωιστικά, και έχοντας μεγαλύτερη πιθανότητα εξερεύνησης να επιλέξουν από τις αρχικές επαναλήψεις την δράση του συμβιβασμού. Επειδή όμως ο συμβιβασμός έχει καλύτερη επιβράβευση από τις άλλες δράσεις, όπως είναι λογικό, οι πράκτορες μετά από μια τυχαία επιλογή αυτής της δράσης δεν έχουν λόγο να δοκιμάσουν άλλες αφού έχει μεγαλύτερη συνάρτηση αξίας η οποία αυξάνεται και με την επιβράβευση. Η διαφορά δηλαδή είναι στο γεγονός ότι αντί οι πράκτορες να ακολουθούν την καθορισμένη λογική σειρά δράσεων (στην αρχή να είναι εγωιστές και να φτάνουν σε μη συμβιβασμούς, και έπειτα να δοκιμάζουν για συμβιβασμό) να επιλέγουν λόγο της μεγάλης πιθανότητας εξερεύνησης μια δράση η οποία θα επιλεγόταν πιο μετά.



Γραφική Παράσταση 5.3.2.4β

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.2.4β με πράκτορες που χρησιμοποιούν αλγόριθμο SARSA με πιθανότητα εξερεύνησης 0.1

Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.5
Παράγοντας Έκπτωσης	0.6
Πιθανότητα εξερεύνησης (ϵ)	0.1

Πίνακας 5.3.2.4β.

Παράμετροι για αποτελέσματα γραφικής παράστασης 5.3.2.4β.

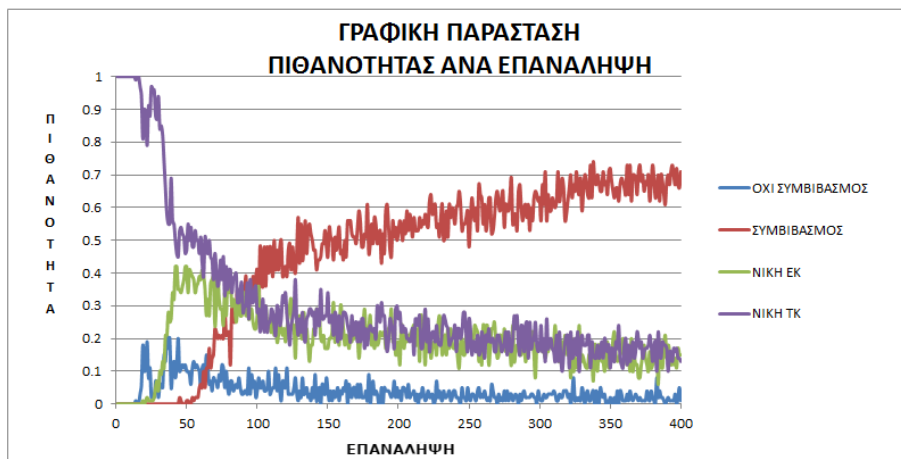
Στην γραφική παράσταση 5.3.2.4β όπου έχουμε μειώσει την πιθανότητα εξερεύνησης βλέπουμε ότι οι πράκτορες συμπεριφέρονται πιο κοντά στις αποφάσεις των πρακτόρων του αλγορίθμου Q-learning, στον οποίο δεν εφαρμόσαμε e-greedy τακτική, αλλά άπληστη.

5.3.2.5 Βέλτιστες παραμέτροι για αλγόριθμο SARSA με e-greedy τακτική

Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.5
Παράγοντας Έκπτωσης	0.6
Πιθανότητα εξερεύνησης (ϵ)	0.3

Πίνακας 5.3.2.5

Παράμετροι για αποτελέσματα γραφικής παράστασης 5.3.2.5



Γραφική Παράσταση 5.3.2.5

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.2.5 με πράκτορες που χρησιμοποιούν αλγόριθμο SARSA

Στη γραφική παράσταση 5.3.2.5 βλέπουμε τις παραμέτρους οι οποίες μετά απο δοκιμές επιφέρουν πιο ρεαλιστικά και ικανοποιητικά αποτελέσματα. Αυτές οι παραμέτροι είναι

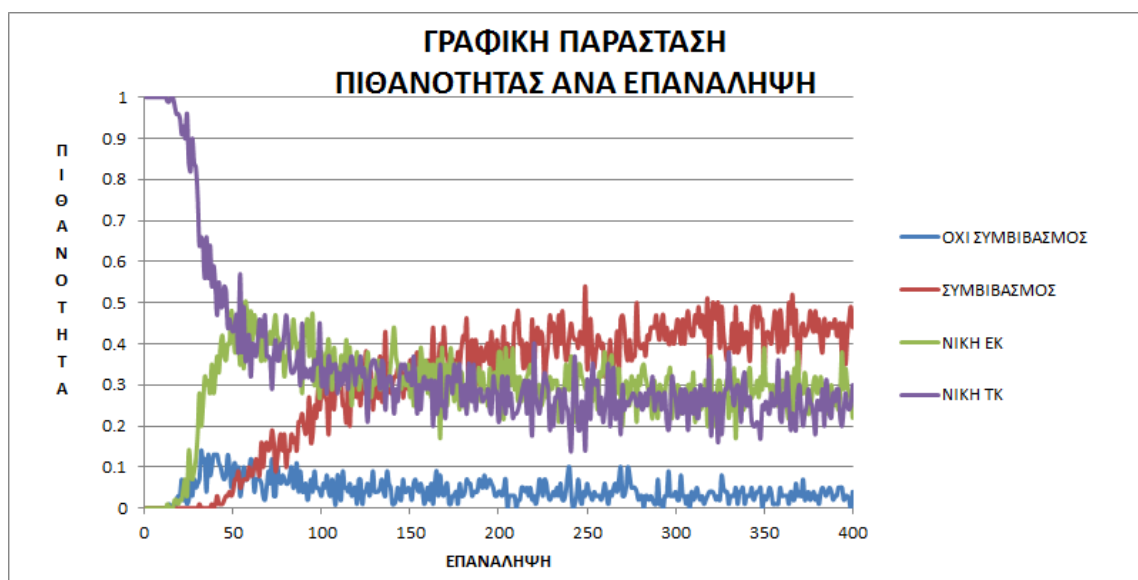
0.5 για ρυθμό μάθησης, 0.6 για παράγοντα έκπτωσης και 0.3 πιθανότητα εξερεύνησης. Όπως αρχικά βλέπουμε, οι παραμέτροι έχουν τις βέλτιστες τιμές κάπου στην μέση των ορίων τους (0-1) . Αυτό είναι λογικό καθώς όσο αφορά τη παράμετρο ρυθμού μάθησης, ένας πράκτορας πρέπει να μαθαίνει ούτε μηχανικά ούτε εντελώς τυχαία, αλλά με κάτι ενδιάμεσο. Επίσης όσο αφορά το παράγοντα έκπτωσης ισχύει το ίδιο , διότι ένας πράκτορας πρέπει να βλέπει και μυωπικά και μακροπρόθεσμα για αν έχει καλύτερα αποτελέσματα. Ακόμη όσο αφορά την πιθανότητα εξερεύνησης ένας πράκτορας πρέπει αν εκμεταλλεύεται και την γνώση, αλλά καί να εξερευνά τυχαίες καταστάσεις και καινούργιες δράσεις. Με αυτές τις τιμές για τις παραμέτρους παίρνουμε αποτελέσματα τα οποία είναι ικανοποιητικά (0.7 πιθανότητα συμβιβασμού) όμως και ρεαλιστικά καθώς βλέπουμε ότι η πιθανότητα συμβιβασμού δεν είναι σχεδόν 1 όπως για κάποιες παραμέτρους, επειδή είναι φυσιολογικό 2 πράκτορες να μην συμβιβάζονται πάντα. Τέλος όσο αφορά τα υπόλοιπα ενδεχόμενα βλέπουμε ότι ανταποκρίνονται στο κυπριακό, καθώς πάλι στις αρχικές επαναλήψεις έχει πιθανότητα να κερδίσει ο TK πράκτορας με 1 διότι όπως είπαμ εκατέχει τις 37 μονάδες απο τις 37 που θέλουν να μοιράσουν, στην συνέχεια αρχίζει και διαπραγματεύεται όταν ο EK πράκτορας δίνει ποσοστά διακυβέρνησης τα οποία εκφράζονται σαν έμμεση τιμωρία και τότε οι 2 πράκτορες αρχίζουν να μην συμβιβάζονται επειδή ο καθένας θέλει περισσότερα για τον εαυτό του και βλέπουν ατομικά, επομένως το ποσό γίνεται 0 αφού δεν αποδέχεται καποιος προσφορά. Έπειτα αρχίζουν να περιματίζονται κάνοντας νίκες και τέλος η πιθανότητα συμβιβασμού υπερτερεί των άλλων ενδεχομένων.

5.3.3 Πειραματισμός με διαφορετικά ποσά συμβιβασμού

Σε αυτό το υποκεφάλαιο θα κάνουμε ένα μικρό πειραματισμό με διαφορετικά ποσα συμβιβασμου για τις 2 πλευρές, κρατώντας τις άλλες παραμέτρους σταθερές. Δηλαδή θα ορίσουμε ένα διαφορετικό ποσο συμβιβασμού για τους πράκτορες απο αυτο που είχαμε θέσει στο Κεφάλαιο 4 (27 μονάδες TK και 10 μονάδες EK).

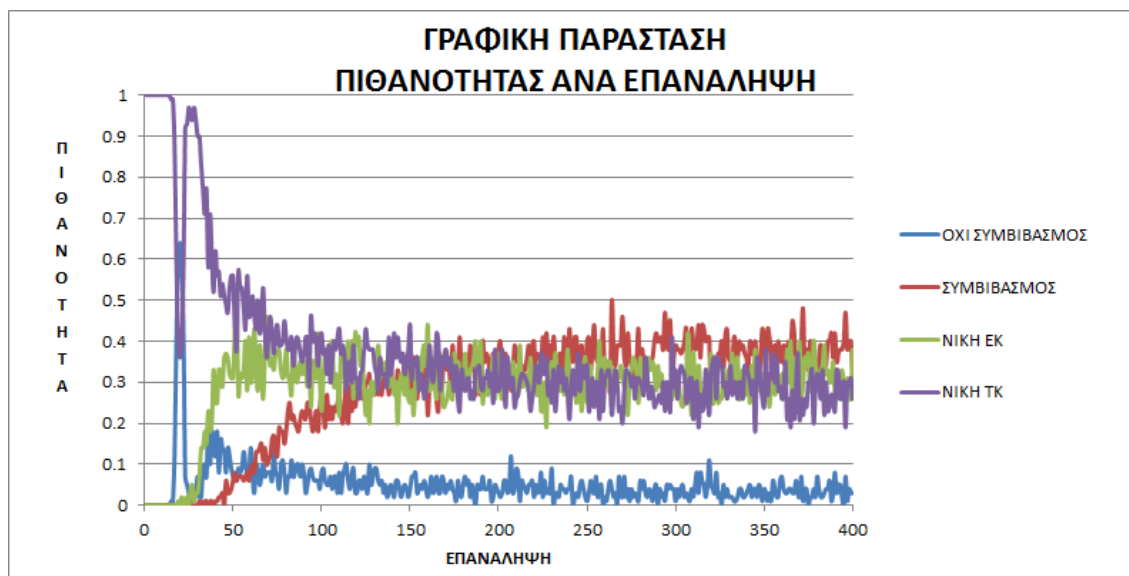
Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.5
Παράγοντας Έκπτωσης	0.6
Πιθανότητα εξερεύνησης (e) (ΓΙΑ ΑΛΓΟΡΙΘΜΟ SARSA)	0.3
ΠΟΣΟ ΣΥΜΒΙΒΑΣΜΟΥ ΓΙΑ ΕΚ	15(ΜΟΝΑΔΕΣ ΑΠΟ ΤΙΣ 37)
ΠΟΣΟ ΣΥΜΒΙΒΑΣΜΟΥ ΓΙΑ ΤΚ	22(ΜΟΝΑΔΕΣ ΑΠΟ ΤΙΣ 37)

Πίνακας 5.3.3.1



Γραφική Παράσταση 5.3.3.1α

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.3.1 με πράκτορες που χρησιμοποιούν αλγόριθμο SARSA με πιθανότητα εξερεύνησης 0.3 και ποσά συμβιβασμού 15 μονάδες για ΕΚ και 22 για ΤΚ

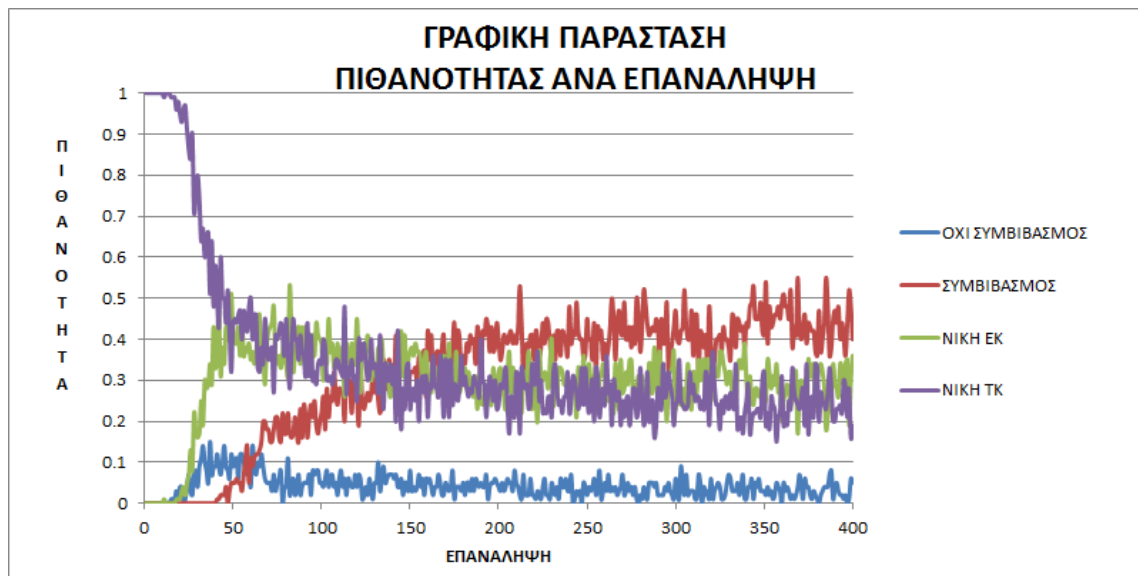


Γραφική Παράσταση 5.3.3.1β

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.3.1 με πράκτορες που χρησιμοποιούν αλγόριθμο Q-learning και ποσά συμβιβασμού 15 μονάδες για ΕΚ και 22 για ΤΚ

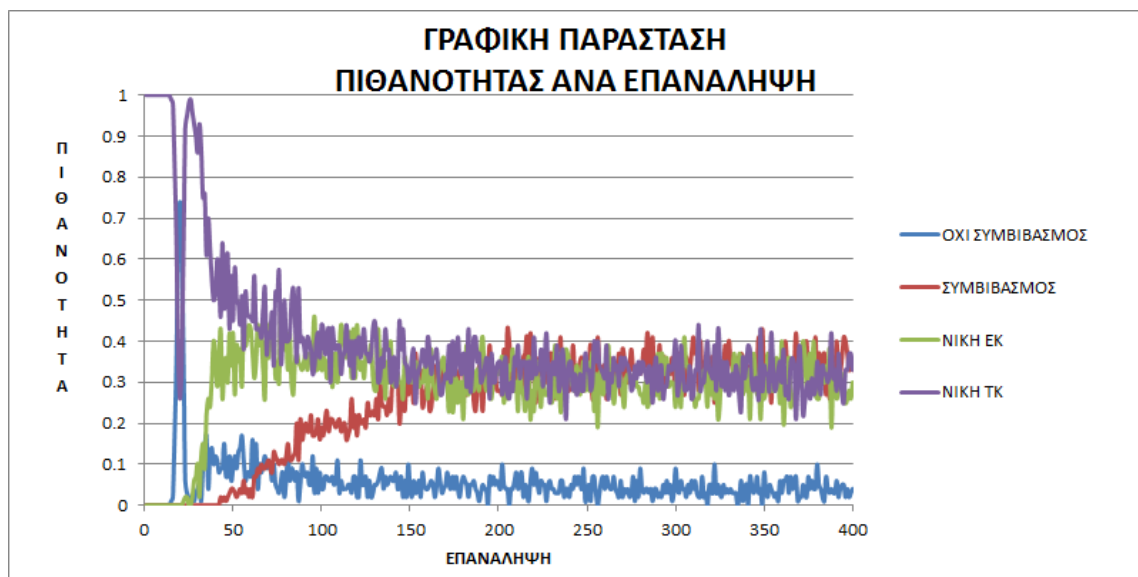
Αριθμός Επαναλήψεων	400
Ρυθμός μάθησης	0.5
Παράγοντας Έκπτωσης	0.6
Πιθανότητα εξερεύνησης (e) (ΓΙΑ ΑΛΓΟΡΙΘΜΟ SARSA)	0.3
ΠΟΣΟ ΣΥΜΒΙΒΑΣΜΟΥ ΓΙΑ ΕΚ	18
ΠΟΣΟ ΣΥΜΒΙΒΑΣΜΟΥ ΓΙΑ ΤΚ	19

Πίνακας 5.3.3.2



Γραφική Παράσταση 5.3.3.2α

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.3.2 με πράκτορες που χρησιμοποιούν αλγόριθμο SARSA με πιθανότητα εξερεύνησης 0.3 και ποσά συμβιβασμού 18 μονάδες για ΕΚ και 19 για ΤΚ



Γραφική Παράσταση 5.3.3.2β

Γραφική Παράσταση με παραμέτρους του πίνακα 5.3.3.2 με πράκτορες που χρησιμοποιούν αλγόριθμο Q-learning και ποσά συμβιβασμού 18 μονάδες για ΕΚ και 19 για ΤΚ

Τρέχοντας το σύστημα για διαφορετικά ποσά συμβιβασμού για τις πλευρές (σε προηγούμενα υποκεφάλαια 27 μονάδες για TK και 10 για EK και τώρα πιο πολλές μονάδες για EK και λιγότερες για TK) γενικά βλέπουμε ότι η πιθανότητα συμβιβασμού μειώνεται. Αυτό οφείλεται για το λόγο ότι και στις 2 περιπτώσεις που αλλάξαμε τα ποσά συμβιβασμού έχουμε αυξήσει το ποσό για τον EK και μειώσει το ποσό για το TK που αυτό σημαίνει ότι επειδή όπως έχουμε πεί ο TK έχει πλεονέκτημα όσο αφορά το εδαφικό, και ορίζουμε ποσό συμβιβασμού μικρότερο από αυτό που ικανοποιείται πιο δύσκολα θα φτάσουν σε συμβιβασμό οι 2 πράκτορες. Γιαυτό και η πιθανότητα συμβιβασμού είναι μειωμένη. Κάτι άλλο που μπορούμε να προσέξουμε είναι ότι αυτή η πιθανότητα μειώνεται ακόμη περισσότερο αν μειώσουμε το ποσό συμβιβασμού για τον TK και αυξήσουμε το ποσό για τον EK (γραφική 5.3.3.2α και 5.3.3.2.β) όπου τα ποσά είναι 18 μονάδες για EK και 19 για TK σε σύγκριση με τις γραφικές (5.3.3.1α και 5.3.3.1β) όπου τα ποσά ήταν 22 μονάδες για TK και 15 για EK. Τέλος αυτό που περιμέναμε επίσης είναι ότι ο αλγόριθμος SARSA με e-greedy τακτική έχει ελαφρώς καλύτερα αποτελέσματα από τον αλγόριθμο Q-learning και σε αυτό το πείραμα, και βεβαίως αυτό οφείλεται στο γεγονός όπου έχουμε εξηγήσει και στις προηγούμενες γραφικές ότι δεν είναι τόσο άπληστος σε σχέση με τον Q-learning και αυτό επειδή έχει και πιθανότητα εξερεύνησης για κάθε κατάσταση.

5.3.4 Συμπεράσματα και σύγκριση αλγορίθμων

Δοκιμάζοντας όλες αυτές τις προσομοιώσεις και πειραματίζοντας με όλους τους παραμέτρους πήραμε κάποια γενικά συμπεράσματα και συγκρίσεις. Το πρώτο συμπέρασμα και πιο σημαντικό είναι ότι από τις 2 γραφικές παραστάσεις 5.3.1.4 και 5.3.2.5 των 2 αλγορίθμων όπου είναι οι βέλτιστες παραμέτροι ώστε να ανταποκρίνονται πλήρως στο να βγουν απολύτως ρεαλιστικά, λογικά και ικανοποιητικά αποτελέσματα, όντως η αντιστοίχιση του κυπριακού προβλήματος στο bargaining game μας έχει επιφέρει αποτελέσματα που ανταποκρίνονται στις λογικές και ρεαλιστικές συνθήκες που επικρατούν στις 2 πλευρές σήμερα. Όλα τα δεδομένα και ανισότητες φαίνονται στις γραφικές και επεξηγούν με το παραπάνω τις δράσεις και τα ενδεχόμενα που καταλήγουν οι 2 πλευρές στις προσομοιώσεις .

Ένα άλλο συμπέρασμα που βγάλαμε από τους 2 αλγορίθμους, είναι ότι ο αλγόριθμος SARSA στον οποίο είχαμε προσθέσει και πιθανότητα εξερεύνησης , που τον κάνει να διαφέρει ακόμη περισσότερο από την απληστία του Q-learning , έχει καλύτερα

αποτελέσματα, με πιο μεγάλες πιθανότητες συμβιβασμού στις τελευταίες επαναλήψεις. Αυτό εξηγείτε όπως έχουμε αναφέρει στο γεγονός ότι δεν είναι πάντα μυωπικός και άπληστος, αλλά εξερευνά καινούργιες καταστάσεις και δράσεις με κάποια πιθανότητα που τον κάνει να βρει ότι ο συμβιβασμός είναι καλύτερος σαν για το κοινό καλό λύση πιο γρήγορα και έτσι οι πράκτορες συμβιβάζονται με περισσότερη πιθανότητα.

Τέλος ας αναφερθούμε στις παραμέτρους που μελετήσαμε. Καταρχάς έχουμε δει ότι ο πειραματισμός για τις παραμέτρους κάθε αλγόριθμου διαφορετικά έχει επιφέρει σχεδόν τα ίδια αποτελέσματα για κάθε αλγόριθμο, με εξαίρεση το ρυθμό μάθησης. Δηλαδή, όσο αφορά ρυθμό μάθησης φτάσαμε στο συμπέρασμα ότι αν είναι πολύ μεγάλος δεν έχει τόσο καλά αποτελέσματα και αν είναι πολύ μικρός δεν επηρεάζει το αλγόριθμο Q-learning, ενώ στον αλγόριθμο SARSA, αν είναι μικρός δεν έχει διαφορά, όμως αν είναι αρκετά μεγάλος επιφέρει καλύτερα αποτελέσματα, και αυτό για τον λόγο ότι εκτός να μαθαίνει μηχανικά υπάρχει η πιθανότητα εξερεύνησης η οποία τον κάνει να μην είναι τόσο άπληστος και να μαθαίνει απολύτως μηχανικά. Όσο αφορά τις τροποποιήσεις για τον παράγοντα έκπτωσης οι 2 αλγόριθμοι συμπεριφέρονται το ίδιο. Και στους 2 όταν μειωθεί αυτός ο παράγοντας έχει καλύτερα αποτελέσματα τα οποία δεν είναι ρεαλιστικά όμως, ενώ αν αυξηθεί έχει χειρότερα. Για την πιθανότητα εξερεύνησης καταλήξαμε ότι όσο πιο μικρή είναι τόσο πιο «όμοια» συμπεριφέρονται οι 2 αλγόριθμοι. Τέλος όσο αφορά τον αριθμό επαναλήψεων, εξακριβώσαμε ότι όντως όσο πιο πολλές οι επαναλήψεις τόσο πιο ψηλή πιθανότητα συμβιβασμού, και χαμηλότερη για τα άλλα ενδεχόμενα, και όσο πιο λίγες επαναλήψεις μικρή πιθανότητα συμβιβασμού με περισσότερες πιθανότητες για τα άλλα ενδεχόμενα.

Όσο αφορά τους πειραματισμούς με διαφορετικά ποσά συμβιβασμού (υποκεφάλαιο 5.3.3) είδαμε ότι η πιθανότητα συμβιβασμού μειώνεται αν αυξήσουμε το ποσό συμβιβασμού για την ΕΚ πλευρά και μειώσουμε το ποσό συμβιβασμού για ΤΚ πλευρά, και αυτό οφείλεται στο γεγονός ότι η ΤΚ πλευρά έχει πελονέκτημα όσο αφορά την εδαφική πτυχή που την κάνει να φτάνει σε συμβιβασμό πιο δύσκολα, αφού μειώνουμε το ποσο συμβιβασμού σε σχέση με το ποσο ικανοποίησης της.

Ακόμη, όσο αφορά τις βέλτιστες παραμέτρους για κάθε αλγόριθμο έχουμε καταλήξει στο γεγονός ότι πρέπει αν είναι κάπου ενδιάμεσα στα όρια τους, δηλαδή ούτε πολύ ψηλά, ούτε πολύ χαμηλά καθώς ένας πράκτορας πρέπει να βλέπει και μυωπικά και

μακροπρόθεσμα ός αφορά παράγοντα έκπτωσης, αλλά ούτε να μαθαίνει μηχανικά, αλλά ούτε και να επιλέγει τυχαία όσο αφορά ρυθμό μάθησης.

Κεφάλαιο 6

Συμπεράσματα και μελλοντική εργασία

6.1 Σύνοψη

6.2 Συμπεράσματα

6.3 Σύγκριση με άλλες Εργασίες

6.4 Μελλοντική εργασία

6.1 Σύνοψη

Η μάθηση και γενικά το machine learning είναι κάποιες από τις σημαντικότερες πτυχές της Τεχνητής Νοημοσύνης σήμερα. Η ενισχυτική μάθηση, η οποία είναι η πιο κοντινή στο τρόπο λειτουργίας των ανθρώπων, λόγω της βιολογικής τάσης μας να μαθαίνουμε από τα αποτελέσματα των πράξεων μας γίνεται εφαρμογή πλέον σε πολλούς τομείς, κάνοντας δυνατό το ενδεχόμενο για πράκτορες να συμπεριφέρονται με την λογική που συμπεριφερόμαστε εμείς οι άνθρωποι.

Μέσω αυτής της διπλωματικής, όπως έχουμε δείξει στο κεφάλαιο 5.1 ένας πράκτορας χρησιμοποιώντας ενισχυτική μάθηση μπορεί εύκολα να προσαρμοστεί στις απαιτήσεις του περιβάλλοντος και να μάθει τι του επιφέρει καλές και τι κακές συνέπειες. Αυτό το μοντέλο, μπορεί να χρησιμοποιηθεί σε πολλές ρεαλιστικές καταστάσεις, και εδώ έρχεται η παραπάνω χρησιμότητα του, να το χρησιμοποιήσουμε για να αναπαραστήσουμε ρεαλιστικά γεγονότα και μέσω αυτού του μοντέλου να δούμε συμπεριφορές των πρακτόρων για πραγματικά προβλήματα.

Αυτό είχε σαν σκοπό αυτή η διπλωματική εργασία. Να χρησιμοποιηθεί αυτό το μοντέλο της ενισχυτικής μάθησης μέσω ενός παιχνιδιού γενικού αθροίσματος, το

bargaining game για να αντιστοιχήσουμε το κυπριακό πρόβλημα με τα δεδομένα που υπάρχουν σήμερα και να δούμε αν 2 πράκτορες, η ΕΚ και η ΤΚ πλευρά μπορούν να φτάσουν σε μία λύση, μια κοινή συνεργασία όσο αφορά το γεωγραφικό διαμοιρασμό της κατεχόμενης Κύπρου, η οποία θα έχει οφέλη και για τις 2 πλευρές, και να μάθουν ότι αυτός ο συμβιβασμός είναι η καλύτερη λύση για το κοινό καλό.

6.2 Συμπεράσματα

Σκοπός της διπλωματικής αυτής όπως είχαμε αναφέρει και στην αρχή ήταν να δούμε αν μπορούμε να προσεγγίσουμε το κυπριακό πρόβλημα, και πιο ειδικά τη γεωγραφική πτυχή του με ενισχυτική μάθηση και να το αντιστοιχήσουμε στο παιχνίδι γενικού αθροίσματος bargaining game.

Τρέχοντας το σύστημα μας και παίρνοντας κάποια αποτελέσματα φτάνουμε στο συμπέρασμα ότι όντως μπορεί να γίνει αυτή η αντιστοίχιση και προσέγγιση, και μάλιστα απολύτως ρεαλιστικά. Επίσης καταλήξαμε ότι μέσω κάποιων προϋποθέσεων και υποχωρήσεων μπορούν να φτάσουν σε συμβιβασμό οι 2 πλευρές.

Αυτό φαίνεται πιο συγκεκριμένα στο κεφάλαιο 5.3 και στα αποτελέσματα των γραφικών παραστάσεων. Από τις γραφικές παραστάσεις φαίνεται καθαρά ότι η πιθανότητα που συμβαίνει κάθε ενδεχόμενο (ΣΥΜΒΙΒΑΣΜΟΣ, ΟΧΙ-ΣΥΜΒΙΒΑΣΜΟΣ, ΝΙΚΗ ΕΚ, ΝΙΚΗ ΤΚ) ανά επανάληψη ανταποκρίνεται στα πραγματικά δεδομένα του προβλήματος. Στην αρχή των επαναλήψεων οι ΤΚ πλευρά δεν δέχεται να διαπραγματευτεί για εδαφική έκταση, για τον λόγο ότι είδη την κατέχει και δεν έχει κάποιο όφελος να την μοιράσει, επομένως έχει νίκες επειδή κατέχει όλο το γεωγραφικό ποσοστό που θέλουμε να μοιραστούν οι 2 πράκτορες. Στην συνέχεια λόγω του ότι η ΕΚ πλευρά προσφέρει ποσοστά διακυβέρνησης στην ΤΚ, η ΤΚ πρέπει να αρχίσει και αυτή να προσφέρει έδαφος και να διαπραγματεύεται την περιοχή που κατέχει με αποτέλεσμα οι 2 πράκτορες να φτάσουν σε όχι συμβιβασμούς στις περισσότερες φορές για τον λόγο ότι και οι 2 θέλουν παραπάνω. Στην συνέχεια βλέπουμε τις 2 πλευρές να έχουν και οι 2 κάποιο αριθμό νικών πειραματίζοντας έτσι για να βρουν την βέλτιστη λύση, και τέλος στις τελευταίες επαναλήψεις οι πράκτορες φτάνουν σε συμβιβασμό και πιο συγκεκριμένα με πιθανότητα μεγαλύτερη του 0.6.

Όλη αυτή η σειρά των ενδεχομένων καθώς και η πιθανότητα να συμβούν σε κάθε επανάληψη είναι απολύτως λογική και μπορούμε να την κατανοήσουμε μέσα από τα πραγματικά δεδομένα που υπάρχουν. Έτσι οι 2 πράκτορες συμπεριφέρονται όπως θα συμπεριφέρονται και οι πλευρές σήμερα με συμπέρασμα ότι η αντιστοίχιση του κυπριακού προβλήματος, όσο αφορά γεωγραφική έκταση στο bargaining game να μας επιβεβαιώνει ότι έχει γίνει σωστά και ότι μπορεί όντως να επιτευχθεί. Τέλος μπορούμε να φτάσουμε και στο συμπέρασμα ότι μέσω κάποιων προϋποθέσεων η λύση, η οποία είναι ο συμβιβασμός μπορεί να επιτευχθεί και στις 2 πλευρές στην πραγματικότητα μέσω του συστήματος, αφού ανταποκρίνεται σε ρεαλιστικές συνθήκες.

Τέλος ένα σημαντικό συμπέρασμα το οποίο δεν φαίνεται από τις γραφικές , αλλά από το σύστημα, είναι ότι οι 2 πλευρές χρειάζονται μια εξωτερική δύναμη, η εξωτερικό πράκτορα να τους βοηθήσει να συμβιβαστούν. Δηλαδή χρειάζονται ένα παράγοντα που να πιέσει την TK πλευρά να αρχίσει να διαπραγματεύεται. Αυτός ο παράγοντας στο σύστημα είναι οι έμμεσες τιμωρίες που δέχεται η TK πλευρά και την αναγκάζουν να αρχίσει να διαπραγματεύεται. Αυτό ισχύει και στην καθημερινότητα, δηλαδή στο πραγματικό κυπριακό πρόβλημα και σε κάθε ρεαλιστικό πρόβλημα που θέλουμε συμβιβασμό, για τον λόγο ότι αν αφήσουμε τις 2 πλευρές απο μόνες τους δύσκολα θα καταφέρουν να φτάσουν σε συμβιβασμό, αν δεν τις πιέσει κάποιος, η να υπάρχει ένας εξωτερικός παράγοντας που να επιβλέπει και να προωθεί τον συμβιβασμό. Στην περίπτωση των πραγματικών δεδομένων για το Κυπριακό, αυτός ο εξωτερικός παράγοντας που επηρεάζει την TK πλευρά είναι η Αμερική, και σαν πιο ουδέτερος παράγοντας τα Ηνωμένα Έθνη, επομένως βλέπουμε πάλι ότι και σε αυτό το συμπέρασμα το σύστημα μας ανταποκρίνεται στα πραγματικά δεδομένα του προβλήματος , άρα η αντιστοίχιση είναι ορθή και ρεαλιστική.

6.3 Σύγκριση με άλλες εργασίες

Σκοπός αυτής της διπλωματικής ήταν η εφαρμογή της ενισχυτικής μάθησης σε πραγματικά προβλήματα. Πιο συγκεκριμένα , να μοντελοποιηθεί το κυπριακό πρόβλημα στο παιχνίδι γενικού αθροίσματος , bargaining game χρησιμοποιώντας ενισχυτική μάθηση.

Το τελευταίο καιρό αυξάνονται οι δοκιμές εφαρμογής της ενισχυτικής μάθησης σε πραγματικά προβλήματα, όπως κούρσες εξοπλισμών, αποφυγή πολέμων και πολλά άλλα περιστατικά διαμαχών που είναι παρόμοια με το κυπριακό πρόβλημα. Επίσης αυτή η εφαρμογή επιτυγχάνεται με την μοντελοποίηση των προβλημάτων σαν παίγνια γενικού αθροίσματος. Στις περισσότερες εργασίες που έχουν προηγηθεί, έχει γίνει αυτή η αντιστοίχιση στο prisoner dilemma [1], ένα άλλο παιχνίδι γενικού αθροίσματος.

Όσο αφορά το κυπριακό πρόβλημα η δική μου δουλειά είναι από τις πρώτες, ίσως και η πρώτη που γίνονται μοντελοποιώντας το κυπριακό στο bargaining game και όχι στο prisoner dilemma. Η εργασία του Καραολή[6] μπορεί να θεωρηθεί ως η πιο κοντινή στην δική μου, η οποία είχε αντιστοίχιση του κυπριακού στο prisoner dilemma.

6.4 Μελλοντικές εργασίες

Το γεγονός ότι η εργασία μου είναι από τις πρώτες όσο αφορά μοντελοποίηση του κυπριακού στο bargaining game και όχι στο prisoners dilemma μπορεί να φέρει πολλές ευκαιρίες για μελλοντική εργασία πάνω σε αυτό το θέμα . Επίσης δεδομένου ότι η μοντελοποίηση μου αφορά την γεωγραφική έκταση για το πρόβλημα του κυπριακού, η επεκτάσεις που μπορούν να γίνουν είναι πάρα πολλές. Για αρχή θα μπορούμε να μοντελοποιηθεί ολόκληρο το πρόβλημα σαν bargaining game, συμπεριλαμβάνοντας δηλαδή και άλλες πτυχές του κυπριακού, όπως για παράδειγμα και τα ποσοστά διακυβέρνησης, τα ποσοστά προσφύγων, δικαιώματα και πολλά άλλα θέματα που οι 2 πλευρές αντιμετωπίζουν.

Μια ακόμη διαφοροποίηση που θα μπορούσε να γίνει είναι να γίνει η μοντελοποίηση που έχω κάνει, να γίνει για κάθε φάση του κυπριακού, από την αρχή μέχρι σήμερα, για μερικές η και όλες τις πτυχές του και να παρατηρηθούν περισσότερα γεγονότα.

Επίσης μια διαφοροποίηση της δικής μου εργασίας που θα μπορούσε να γίνει, είναι αντί να μειώνεται το ποσό κάθε γύρο, οι πράκτορες να δέχονται περισσότερη τιμωρία κάθε γύρο, που λογικά θα τους κάνει να συγκλείσουν στον συμβιβασμό πάλι.

Τέλος , για όλα τα παραπάνω που θα μπορούσαν να γίνουν, ακόμη και για την δική μου εργασία, θα μπορούσε κάποιος να δοκιμάσει και άλλους αλγόριθμους ενισχυτικής

μάθησης με συνάρτηση και άλλων τακτικών που ίσως θα φέρουν καλύτερα αποτελέσματα.

ΑΝΑΦΟΡΕΣ

1 Davis, W. (1997). Iterated Prisoner's Dilemma Online Game and Simulation

<http://www.iterated-prisoners-dilemma.net/>

(Accessed 29/12/2016).

2 Dresher, M. (1961). The Mathematics of Games of Strategy: Theory and Applications Prentice-Hall, Englewood Cliffs, NJ

3 Ευσέβεια Κυριάκου (2014). Μοντελοποίηση κούρσας εξοπλισμού όπλων με το Δίλημα του Φυλακισμένου και εκπαίδευση των πρακτόρων με ενισχυτική μάθηση.

4 Γεωργίου Άννα (2015). Μελέτη σχέσης αυτοελέγχου και συνειδητότητας. Μοντελοποίηση άνω και κάτω μέρους εγκεφάλου σαν πράκτορες που παίζουν το παίγνιο γενικού αθροίσματος, Δίλημα του Φυλακισμένου.

5 Kaelbling, L.P. –Littman, M.L. – Moore AW. (1996). Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research. 4: 237–285.

6 Καραολής, Γ. (2010). Προσέγγιση του κυπριακού προβλήματος με ενισχυτική μάθηση πολλαπλών πρακτόρων σε παίγνια γενικού αθροίσματος. Διπλωματική Εργασία, Τμήμα Πληροφορικής, Πανεπιστήμιο Κύπρου.

7 Lumsden, M. (1973). The Cyprus Conflict as a Prisoner's Dilemma Game, Journal of Conflict Resolution 17 (1), 7-32.

8 Rubinstein, A. (1982). Perfect equilibrium in a bargaining model. Econometrica, 50(1), 99-109.

9 Sutton, R.S. – Barto, A.G. (1998). Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA.

10 Sutton, R.S. (1988). Learning to predict by the methods of temporal differences. Machine Learning. 3 (1): 9–44.

11 Sutton, R.S. (1984). Temporal Credit Assignment in Reinforcement Learning (PhD thesis). University of Massachusetts, Amherst, MA.

12 Tesauro G. (March 1995). Temporal Difference Learning and TD-Gammon, Communications of the ACM, 38(3), 58-68.

13 Vassiliades, V. and Christodoulou, C. (2010). Multiagent Reinforcement Learning in the Iterated Prisoner's Dilemma: Fast Cooperation through Evolved Payoffs. Proceedings of 198 the Int Joint Conf on Neural Networks (part of the World Congress on Computational Intelligence), July 2010, Barcelona, Spain, pp. 2828-2835.

14 Vassilis Vassiliades, Aristodemos Cleanthous, Chris Christodoulou(2011).Multiagent Reinforcement Learning: Spiking and Nonspiking Agents in the Iterated Prisoner's Dilemma", IEEE Transactions on Neural Networks vol.22, pp.639-653, 2011

15 Watkins, C.J.C.H., (1989), Learning from Delayed Rewards. Ph.D. Thesis, Cambridge University.

16 Yesilada, B.A.- Sozen, A(2002). Negotiating a resolution to the Cyprus problem: Is potential European Union membership a blessing or a curse? J. Int. Negot., vol. 7, no. 2, pp. 261–285, 2002.

ΠΑΡΑΡΤΗΜΑ Α : ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ

ΚΛΑΣΗ AGENT

```
//this class is the super class for agents
public class Agent {
    double win;//the amount that the agents wins in an iteration
    int prevstate;//previous state
    int state;//state
    int satisfied;//the amount of being satisfied
    State Q[];// the states table
    int init_area;//initial area
    int adv;//option if the agent has advantage
    int action;//action that agents do
    int prevaction;//previous action
    //in this function the agent make offer
    int make_offer(double area,String type){
        return 0;}
    //in this action the agent will decide for offer
    int decide_for_offer(Agent a,Agent b,int offer,double area,int iterations,int
count){
        return 0;}

}
```

ΚΛΑΣΗ ACTION

```
//this class represents the actions that agents do
public class Action {
    String decision; //if the action is REFUSE OR ACCEPT
    String counteroffer; //the counteroffer style (<,>=)
    double value_function; //the estimated value of the action
    //constructor
    public Action(String decision,String counteroffer,double value_function){
        this.decision=decision;
        this.counteroffer=counteroffer;
        this.value_function=value_function;
    }
}
```

ΚΛΑΣΗ STATE

```
import java.util.LinkedList;
//this class represent the states of the agents
public class State {
    LinkedList<Action> actions; //list of actions of the state
    //constructor
    public State(){
        this.actions=new LinkedList<Action>();
    }
}
```

KΛΑΣΗ main

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.LinkedList;
import java.util.Scanner;
public class main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner s = new Scanner(System.in);

        int iterations=0;
        int simvivasmoi=0;
        int oxi_simvivasmoi=0;
        int NIKES_TK=0;
        int NIKES_EK=0;
        double learning_rate_EK=0;
        double learning_rate_TK=0;
        double discount_factor_EK=0;
        double discount_factor_TK=0;
        double epsilon=0;
        boolean a=false,b=false,c=false,d=false,e=false,
        f=false,g=false;
        String algorithm="";
        LinkedList<String> tmp=new LinkedList<String>();
        //list of results for each iteration

        //reading form input.txt file
```

```

        try {
            BufferedReader br = new BufferedReader(new
FileReader("input.txt"));

            String line = br.readLine();

            while (line != null) {

                String tmps[]=line.split(" ");
                if(tmps[0].compareTo("Iterations:")==0){
                    iterations=Integer.parseInt(tmps[1]);a=true;}
                if(tmps[0].compareTo("Algorithm:")==0){
                    b=true;algorithm=tmps[1];}
                if(tmps[0].compareTo("learning_rate_EK:")==0){

learning_rate_EK=Double.parseDouble(tmps[1]);c=true;}
                if(tmps[0].compareTo("learning_rate_TK:")==0){

learning_rate_TK=Double.parseDouble(tmps[1]);d=true;}
                if(tmps[0].compareTo("discount_factor_EK:")==0){

e=true;discount_factor_EK=Double.parseDouble(tmps[1]);}
                if(tmps[0].compareTo("discount_factor_TK:")==0){

f=true;discount_factor_TK=Double.parseDouble(tmps[1]);}
                if(tmps[0].compareTo("epsilon:")==0){
                    g=true;epsilon=Double.parseDouble(tmps[1]);}

                line = br.readLine();
            }
            br.close();
        } catch (Exception e2) {e2.printStackTrace();
        }

```



```

//checking if input file is missing any parameter
if(algorithm.compareTo("Q_learning")==0){
    if(a&&b&&c&&d&&f&&e){
    else System.out.println("Wrong parameters in
input file,Please check the example");}
else if(algorithm.compareTo("SARSA")==0){
    if(a&&b&&c&&d&&f&&g&&e){
    else System.out.println("Wrong parameters
in input file,Please check the example");
}
else System.out.println("Wrong parameters
in input file,Please check the example");

//running the simulation with parameters of file
Game game=new Game();
tmp=game.play(iterations,algorithm,learning_rate_EK,
learning_rate_TK,discount_factor_EK,discount_factor_TK,epsilon);
for(int i=0;i<tmp.size();++i){
    if(tmp.get(i).compareTo("EK")==0)NIKES_EK++;
    else if(tmp.get(i).compareTo("TK")==0)NIKES_TK++;
    else if(tmp.get(i).compareTo("NS")==0)oxi_simvivasmoi++;
    else if(tmp.get(i).compareTo("S")==0)simvivasmoi++;

}

//printing results in file results.txt
try{PrintWriter w=new PrintWriter("results.txt","UTF-8");
w.print("RESULTS:\n\nSIMVIVASMOI: "+simvivasmoi+
" NIKES_EK: "+NIKES_EK+" NIKES_TK: "+NIKES_TK+
" OXI_SIMVIVASMOI: "+oxi_simvivasmoi+"\n\n");
for(int l=0;l<tmp.size();++l)

```

```

        {w.print("ITERATION "+(l+1)+"\t");
        if(tmp.get(l).compareTo("NS")==0)w.println("OXI SIMVIVASMOS");
        else
        if(tmp.get(l).compareTo("S")==0)w.println("SIMVIVASMOS");
        else if(tmp.get(l).compareTo("TK")==0)w.println("NIKH TK");
        else if(tmp.get(l).compareTo("EK")==0)w.println("NIKH EK");

        }

        w.close();
    }catch(IOException ex){}

    }
}

```

KΛΑΣΗ Game

```

import java.io.IOException;
import java.io.PrintWriter;
import java.util.*;
//simulation class of game
public class Game {

    double init_area;//initial area
    double area=37;// currentarea
    //constructor

```

```

public Game(){
    area=37;
    init_area=area;}

//function tha runs the simulation and returns
//    results of iterations in list
LinkedList<String> play(int iterations,
String algorithm,double learning_rate_EK,
double learning_rate_TK,double discount_factor_EK,
double discount_factor_TK,double epsilon){
    LinkedList<String> r=new LinkedList<String>();
    Agent EK;
    Agent TK;
    //initialization of q learning agents
    if(algorithm.compareTo("Q_learning")==0){
        EK=new Q_learning_Agent((int)area,
            (int)(27),0,learning_rate_EK,
                discount_factor_EK);
        TK=new Q_learning_Agent((int)area,
            (int)(73),1,learning_rate_TK,
                discount_factor_TK);}

    //initialization of sarsa agents
    else if(algorithm.compareTo("SARSA")==0){
        EK=new SARSA_Agent((int)area,
            (int)(27),0,epsilon,learning_rate_EK,
                discount_factor_EK);
        TK=new SARSA_Agent((int)area,
            (int)(73),1,epsilon,learning_rate_TK,
                discount_factor_TK );}

    //default initilization if error in file
    else {
        System.out.println("WRONG ALGORITHM GIVEN,
        please check example");
        EK=new Q_learning_Agent((int)area,

```

```

(int)(27),0,learning_rate_EK,
        discount_factor_EK);
TK=new Q_learning_Agent((int)area,
(int)(73),1,learning_rate_TK,
        discount_factor_TK);
System.exit(0);}

//running simulation for iterations number

for(int i=0;i<iterations;++i){

    int turn;
    //random parameter for choosing the agent that
    //makes first offer with same possibilities
    if(Math.random()>0.5)turn=1;
    else turn=0;
    area=37;
    int offer;
    boolean simvivasmos=false;
    if(turn==1){//EK makes first offer
        while(area>0){
            String off;
            if(area==37)off=makeoffer1(EK);
            else off=makeoffer2(EK);
            offer=EK.make_offer(area,off);
            //TK decides for offer
            int action=TK.decide_for_offer(TK,EK,
            offer,area,iterations,i);
            TK.action=action;
            //case TK accepts
            if(TK.Q[TK.state].actions.get(action).
            decision.compareTo("Accept")==0){

```

```

        if((TK.Q[TK.state].actions.get(action).
        counteroffer.compareTo("<")==0)||

(TK.Q[TK.state].actions.get(action).

counteroffer.compareTo(">")==0)){
    //case accepts greater
    //or less offer of satisfying amount
    EK.win=area-offer;
    TK.win=offer;
    if(TK.win<27&&EK.win>10){//EK wins
        TK.Q[TK.state].actions.
        get(TK.action).value_function-
=0.15;

        EK.Q[EK.state].actions.

        get(EK.action).value_function+=0.05;

        r.add("EK");

    }
    else if(TK.win>27&&EK.win<10){
        //TK wins

        TK.Q[TK.state].actions.

        get(TK.action).value_function+=0.25;

        EK.Q[EK.state].actions.
        get(EK.action).value_function-
=0.05;

        r.add("TK");

```

```

    }
    else { //OXI SIMVIVASMOS
        r.add("NS");
        EK.Q[EK.prevstate].actions.
        get(EK.prevaction).
        value_function-=0.10;
        TK.Q[TK.prevstate].actions.
        get(TK.prevaction).
        value_function-=0.10;}

        break;
    }else if(TK.Q[TK.state].actions.
    get(action).counteroffer.
    compareTo("0")==0){

        break;

    }
    else {simvivasmos=true; //SIMVIVASMOS

    EK.win=area-offer;
    TK.win=offer;
    TK.Q[TK.state].actions.get(TK.action).
    value_function+=0.2;
    EK.Q[EK.state].actions.get(EK.action).
    value_function+=0.2;

    r.add("S");

    }
    break; }
else if(TK.Q[TK.state].actions.
get(TK.action).
counteroffer.compareTo("*")==0){

```

```

//
//case TK wins because the punishment
//      didn't started (STARTING iterations)
r.add("TK");

TK.Q[TK.state].actions.get(TK.action).
value_function-=1;

break;
}
else{
//area --
if(area>0)
{String dec;
if(offer==0){if(action==1)
dec("<0.5");
else if(action==2)
dec(">0.5");
else if(action==3)
dec("");
else dec="*";}
else {if(action==0)
dec("<0.5");
else if(action==2)dec(">0.5");
else if(action==4)dec("");
else dec="*";}
//now TK makes offer
int offer2=TK.make_offer(area,dec);
int decision2=EK.decide_for_offer(EK,TK,
offer2,area,iterations,i);
EK.action=decision2;
if(EK.Q[EK.state].actions.get(decision2).
decision.compareTo("Accept")==0){

```

```

//if EK accepts

if(EK.Q[EK.state].actions.
get(decision2).
counteroffer.compareTo("<")==0||
    EK.Q[EK.state].actions.
    get(decision2).

counteroffer.compareTo(">")==0)
{

    EK.win=offer2;
    TK.win=area-offer2;

if(TK.win<27&&EK.win>10){//EK wins

    EK.Q[EK.state].actions.
    get(EK.action).
    value_function+=0.05;

    r.add("EK");

    TK.Q[TK.state].actions.
    get(TK.action).
    value_function-=0.15;
}
else
if(TK.win>27&&EK.win<10){

    //TK wins
    EK.Q[EK.state].actions.
    get(EK.action).
    value_function-=0.05;
    TK.Q[TK.state].actions.

```



```

get(TK.action).
value_function+=0.25;

r.add("TK");
}
else { //OXI SIMVVIVASMOS
r.add("NS");

EK.Q[EK.prevstate].actions.

get(EK.prevaction).
value_function-=0.10;

TK.Q[TK.prevstate].actions.

get(TK.prevaction).
value_function-=0.10;}

break;}
else if(EK.Q[EK.state].actions.
get(action).
counteroffer.compareTo("0")==0){

break;}
else {simvivasmos=true;
//SIMVVIVASMOS

TK.Q[TK.state].actions.get(TK.action).
value_function+=0.2;
EK.Q[EK.state].actions.get(EK.action).
value_function+=0.2;
r.add("S");

break;
}

```

```

        }
        else
            area--;
    }
    else break;
}
}}
else { //TK makes offer first
    while(area>0){
        String off;
        if(area==37)off=makeoffer1(TK);
        else off=makeoffer2(TK);
        offer=TK.make_offer(area,off);

        int action=EK.decide_for_offer(EK,TK,
            offer,area,iterations,i);
        EK.action=action;
        if(EK.Q[EK.state].actions.get(action).
            decision.compareTo("Accept")==0){
            //EK Accepts

            if((EK.Q[EK.state].actions.get(action).
                counteroffer.compareTo("<")==0)||

(EK.Q[EK.state].actions.get(action).

counteroffer.compareTo(">")==0)){
                TK.win=area-offer;
                EK.win=offer;
                if(TK.win<27&&EK.win>10){ //EK wins
                    TK.Q[TK.state].actions.

```

```

    =0.15;

    get(TK.action).value_function-

    EK.Q[EK.state].actions.

    get(EK.action).value_function+=0.05;

    r.add("EK");
}
else if(TK.win>27&&EK.win<10){
    //TK wins

    TK.Q[TK.state].actions.

    get(TK.action).value_function+=0.25;

    EK.Q[EK.state].actions.
    get(EK.action).value_function-

    =0.05;

    r.add("TK");

}
else {//OXI SIMVIVASMOS
    r.add("NS");

    EK.Q[EK.prevstate].actions.
    get(EK.prevaction).
    value_function-=0.20;
    TK.Q[TK.prevstate].actions.
    get(TK.prevaction).
    value_function-=0.20;}

    break;
}else if(EK.Q[EK.state].actions.
get(action).

```

```

counteroffer.compareTo("0")==0){
    break;
}
else {
    //SIMVIVASMOS
    simvivasmos=true;

    TK.Q[TK.state].actions.get(TK.action).
    value_function+=0.2;
    EK.Q[EK.state].actions.get(EK.action).
    value_function+=0.2;

    r.add("S");

}
break; }
else{
    //area--;
    if(area>0/*17*/)
    {String dec;
    if(offer==0){
        if(action==1)
            dec="<0.5";
        else if(action==2)
            dec=">0.5";
        else if (action==3)
            dec="=";
        else dec="*";}
    else {
        if(action==0)
            dec="<0.5";
        else if(action==2)
            dec=">0.5";

```

```

        else if(action==4)
            dec="=";
        else dec="*";}

//EK turn to make offer
int offer2=EK.make_offer(area,dec);
int decision2=TK.decide_for_offer(
TK,EK,offer2,area,iterations,i);
TK.action=decision2;
if(TK.Q[TK.state].actions.get(decision2).
decision.compareTo("Accept")==0){
    //TK accepts
    if(TK.Q[TK.state].actions.
get(decision2).
counteroffer.compareTo("<")==0||
        TK.Q[TK.state].actions.
get(decision2).

counteroffer.compareTo(">")==0)
    {

        TK.win=offer2;
        EK.win=area-offer2;

if(TK.win<27&&EK.win>10){//EK wins

        EK.Q[EK.state].actions.
get(EK.action).
value_function+=0.05;

        r.add("EK");

        TK.Q[TK.state].actions.
get(TK.action).

```

```

value_function-=0.15;
}
else
if(TK.win>27&&EK.win<10){

//TK wins
EK.Q[EK.state].actions.
get(EK.action).
value_function-=0.05;
TK.Q[TK.state].actions.
get(TK.action).
value_function+=0.25;

r.add("TK");

}
else { //OXI SIMVIVASMOS
r.add("NS");

EK.Q[EK.prevstate].actions.

get(EK.prevaction).
value_function-=0.20;

TK.Q[TK.prevstate].actions.

get(TK.prevaction).
value_function-=0.20;}

break;}
else if(TK.Q[TK.state].
actions.get(action).
counteroffer.compareTo("0")==0){
break;}
else { //SIMVIVASMOS

```

```

simvivasmos=true;

TK.Q[TK.state].actions.
get(TK.action).
value_function+=0.05;
EK.Q[EK.state].actions.
get(EK.action).
value_function+=0.05;

r.add("S");

break;
}

else if(TK.Q[TK.state].actions.
get(TK.action).
counteroffer.compareTo("*")==0){
    //CASE TK wins because punishments
// didn't start yet(STARTING ITERATIONS)

    r.add("TK");

    TK.Q[TK.state].actions.get(TK.action).
    value_function-=1;

    break;
}
else
    area--;
}
else break;
}
}

```

```
if(area<=0){//AREA 0 => OXI SIMVIVASMOS
```

```

    r.add("NS");
    EK.Q[EK.prevstate].actions.
    get(EK.prevaction).          value_function-=0.20;
    TK.Q[TK.prevstate].actions.
    get(TK.prevaction).          value_function-=0.20;
}

}

```

```
return r;}
```

```
//function that an agents make offer based
```

```
//on the best value of its states
```

```
//table in the first round
```

```
String makeoffer1(Agent a){
    double maxq=a.Q[0].actions.get(0).
    value_function;
    String ans="";
    for(int i=0;i<a.Q.length;++i)
        for(int j=0;j<a.Q[i].actions.size();++j){
            if(j==0||j==2||j==4||j==6)
                if(a.Q[i].actions.get(j).
                value_function>maxq){
                    maxq=a.Q[i].actions.get(j).
                    value_function;
                    ans=a.Q[i].actions.get(j).decision;
                    a.prevstate=a.state;
                    a.prevaction=a.action;
                    a.action=j;a.state=i;}
        }
}

```



```

    return ans;
}
//function that the agent makes
// offer in any other round
String makeoffer2(Agent a){
    if(a.state!=0){
        if(a.Q[a.state].actions.get(0).value_function
        >=a.Q[a.state].actions.get(2).value_function&&
        a.Q[a.state].actions.get(0).value_function
        >=a.Q[a.state].actions.get(4).
        value_function){
            a.prevstate=a.state;a.prevaction=a.action;
            a.action=0;return "<0.5";}
        else if(a.Q[a.state].actions.get(2).
        value_function>=a.Q[a.state].actions.get(0).
        value_function&&a.Q[a.state].actions.get(2).
        value_function>=a.Q[a.state].actions.get(4).
        value_function){a.prevstate=a.state;
            a.prevaction=a.action;
            a.action=1;return ">0.5";}
        else if(area<init_area){
            if(a.Q[a.state].actions.get(0).
            value_function>=a.Q[a.state].actions.get(2).
            value_function){
                a.prevstate=a.state;a.prevaction=a.action;
                a.action=0;return "<0.5";}
            else {a.prevstate=a.state;
                a.prevaction=a.action;
                a.action=1;return ">0.5";}}
        else {a.prevstate=a.state;
            a.prevaction=a.action;
            a.action=2; return "=";}
    }
}

```

```

    }
    else {
        if(a.Q[a.state].actions.get(1).value_function
        >=a.Q[a.state].actions.get(2).value_function&&
            a.Q[a.state].actions.get(1).value_function
            >=a.Q[a.state].actions.get(3).
            value_function){
            a.prevstate=a.state;a.prevaction=a.action;
            a.action=1;return "<0.5";}
        else if(a.Q[a.state].actions.get(2).
        value_function>=a.Q[a.state].actions.get(1).
        value_function&&a.Q[a.state].actions.get(2).
        value_function>=a.Q[a.state].actions.get(3).
        value_function){
            a.prevstate=a.state;a.prevaction=a.action;
            a.action=2;return ">0.5";}
        else if(area<init_area){
            if(a.Q[a.state].actions.get(1).
            value_function>=a.Q[a.state].actions.get(2).
            value_function){
                a.prevstate=a.state;a.prevaction=a.action;
                a.action=1;return "<0.5";}
            else {a.prevstate=a.state;
            a.prevaction=a.action;
            a.action=2;return ">0.5";}
        }
        else {a.prevstate=a.state;
        a.prevaction=a.action;
        a.action=3; return "=";}
    }
}

```

```
}
```

ΚΛΑΣΗ Q_learning_Agent

//this class represents q learning agent

```
public class Q_learning_Agent extends Agent{
```

```
    double discount;//discount factor of agent
```

```
    double learning_rate;//learning rate of agent
```

```
    //constructor
```

```
    public Q_learning_Agent(int area,int satisfied,  
int adv,double learning_rate,double discount_factor){
```

```
        this.satisfied=satisfied;
```

```
        this.learning_rate=learning_rate;
```

```
        this.discount=discount_factor;
```

```
        Q=new State[area];
```

```
        state=0;action=0;
```

```
        prevstate=0;
```

```
        prevaction=0;
```

```
        this.adv=adv;
```

```
        //initialization depending if the agent has advantage or not
```

```
        if(adv==0){
```

```
            for(int i=0;i<area;i++){
```

```
                Q[i]=new State();
```

```
                init_area=area;
```

```
                if(i!=0){
```

```
                    Q[i].actions.add(new
```

```
                    Action("Refuse","<0.5",0.5));
```

```
                    Q[i].actions.add(new Action("Accept",>",0.5));
```

```
                    Q[i].actions.add(new
```

```
                    Action("Refuse",>0.5",0.5));
```

```

        Q[i].actions.add(new Action("Accept", "<", 0.5));
        Q[i].actions.add(new Action("Refuse", "=", 0.5));
        Q[i].actions.add(new Action("Accept", "=", 0.5));
    }
    else{
        Q[i].actions.add(new Action("Accept", "0", 0));
        Q[i].actions.add(new
Action("Refuse", "<0.5", 0.5));

        Q[i].actions.add(new
Action("Refuse", ">0.5", 0.5));

        Q[i].actions.add(new Action("Refuse", "=", 0.5));
    }
}

else {
    for(int i=0;i<area;i++){
        Q[i]=new State();
        init_area=area;
        if(i!=0){
            Q[i].actions.add(new
Action("Refuse", "<0.5", 0.5));

            Q[i].actions.add(new Action("Accept", ">", 0.5));

            Q[i].actions.add(new

Action("Refuse", ">0.5", 0.5));

            Q[i].actions.add(new Action("Accept", "<", 0.5));

            Q[i].actions.add(new Action("Refuse", "=", 0.5));

            Q[i].actions.add(new Action("Accept", "=", 0.5));
            Q[i].actions.add(new Action("Refuse", "*", 1));

```

```

    }
    else{
        Q[i].actions.add(new
Action("Accept","0",0));

        Q[i].actions.add(new Action("Refuse","<0.5",0.5));
        Q[i].actions.add(new Action("Refuse",>0.5",0.5));
        Q[i].actions.add(new Action("Refuse","=",0.5));
        Q[i].actions.add(new Action("Refuse","*",1));
    }
}
}
}

//in this function the agent sees what is the max value of the next state
double max_q(int offer,double area,State[] table){
    if(offer==0){
        if(area==0)
            return table[0].actions.get(0).value_function;
        else {
            double max=0;
            for(int i=1;i<table[0].actions.size();++i)
                if(table[0].actions.get(i).value_function>max)
                    if(area<init_area&&i==3){}
                    else

max=table[0].actions.get(i).value_function;
            return max;}
        }
    else {if(offer==1)//if offer is -1 means that
// is a code that the other agent did cooperation offer
        offer=(int)(Math.round((satisfied*area/100)));
        double max=0;

```

```

        for(int i=0;i<table[offer].actions.size();++i){
            if(table[offer].actions.get(i).value_function>max)
                if(area<init_area&&(i==4||i==5)){
                    else max=table[offer].actions.get(i).value_function;
                }
        }
        return max;

    }}

//in this function the agent decides what will be
// the action of the current state

int decide_for_offer(Agent a,Agent b,int offer,
double area,int iterations,int count){
    boolean flag=false;
    boolean flag2=false;
    if(offer==0||(offer==-1&&area==1)){offer=0;
    prevstate=state;
    prevaction=action;
    state=0;
    if(area==0){this.action=0;return 0;}
    else{
        double max_next_q=0;
        int maxaction=1;
        double max=a.Q[0].actions.get(1).value_function;;
        for(int i=1;i<a.Q[0].actions.size();++i)
            if(area<init_area&&(i==3)){
                else
                {double t=a.Q[0].actions.get(i).value_function;
                if(t>max){
                    maxaction=i;
                    max=t;
                }
            }

        String tmp1=a.Q[0].actions.get(maxaction).counteroffer;

```

```

int tmp2=make_offer(area,tmp1);

int tmp3=decide_for_offer2(b,area,tmp2);
max_next_q=max_q(tmp3,area,a.Q);
    if(b.Q[b.state].actions.get(b.action).
        counteroffer.compareTo(""+')==0){
        }
else{
    //updating table
    if(maxaction==1)
    {if(adv==1&&area>33)
        a.Q[0].actions.get(maxaction).value_function
        +=learning_rate*(0.44+discount*max_next_q-a.Q[0].
        actions.get(maxaction).value_function);
    else a.Q[0].actions.get(maxaction).value_function+=
        learning_rate*(0.3+discount*max_next_q-a.Q[0].
        actions.get(maxaction).value_function);}
    else if(maxaction==2)
        a.Q[0].actions.get(maxaction).value_function
        +=learning_rate*(0.35+discount*max_next_q-a.Q[0].
        actions.get(maxaction).value_function);
    else if(maxaction==3)
        a.Q[0].actions.get(maxaction).value_function+=
        learning_rate*(0.55+discount*max_next_q-a.Q[0].
        actions.get(maxaction).value_function);
    else {a.Q[0].actions.get(maxaction).value_function+=
        learning_rate*(0.5+discount*max_next_q-a.Q[0].
        actions.get(maxaction).value_function);}
    }        this.action=maxaction;
return maxaction;
}
}
else {

```

```

        boolean bigger=false;
        boolean smaller=false;
        if(offer==1||(offer==0.5*area)){
            flag=true;
            offer=(int)(Math.round((satisfied*area/100)));
        }
        else if(offer<(0.5*area))smaller=true;
        else bigger=true;

        prevaction=action;
        prevstate=state;
        state=offer;
        int maxaction=0;
        double max=0;
        double max_next_q=0;
        for(int i=0;i<a.Q[offer].actions.size();++i)
        {
            if(flag&&(i==1||i==3||i==4)){

            }else if((i==4||i==5)&&(bigger||smaller)&&
            (area<((double)init_area))){}
            else if(bigger&&(i==3||i==5)){}
            else if(smaller&&(i==1||i==5)){}
            else {
                double r=Math.random();
                if(true){

                }

            }

        }

        if(a.Q[offer].actions.get(i).value_function>max)

        {max=a.Q[offer].actions.get(i).value_function;
            maxaction=i;
        }
    }
}

```



```

    }
}

```

```

String tmp1=a.Q[offer].actions.get(maxaction).counteroffer;
int tmp2=make_offer(area,tmp1);
int tmp3=decide_for_offer2(b,area,tmp2);
max_next_q=max_q(tmp3,area,a.Q);

```

//updating

```

if(maxaction==0)// refuse counter offer <
{if(adv==1&&area>33)
    a.Q[offer].actions.get(maxaction).value_function+=
learning_rate*(0.4+discount*max_next_q-a.Q[offer].actions.
get(maxaction).value_function);
else
    a.Q[offer].actions.get(maxaction).
value_function+=learning_rate*(0.3+discount*
max_next_q-a.Q[offer].actions.get(maxaction).value_function);

a.Q[offer].actions.get(maxaction).value_function-=
(count/2*0.005);
}
else if(maxaction==1)//accept counteroffer > satisfying
{a.Q[offer].actions.get(maxaction).value_function
+=learning_rate*(0.55+discount*max_next_q-a.Q[offer].
actions.get(maxaction).value_function);
a.Q[offer].actions.get(maxaction).value_function+=0.005;;
}
else if(maxaction==2){//refuse counteroffer >
    a.Q[offer].actions.get(maxaction).value_function
+=learning_rate*(0.35+discount*max_next_q-a.Q[offer].
actions.get(maxaction).value_function);
}
}

```

```

a.Q[offer].actions.get(maxaction).value_function
-=(count/2*0.005);
}
else if(maxaction==3){//accept counter offer <satisfying
    a.Q[offer].actions.get(maxaction).value_function
    +=learning_rate*(0.35+discount*max_next_q-a.Q[offer].
    actions.get(maxaction).value_function);
a.Q[offer].actions.get(maxaction).value_function+=0.005;
}
else if(maxaction==4){//refuse make counteroffer = satisfying of
other player

    a.Q[offer].actions.get(maxaction).value_function+=
    learning_rate*(0.35+discount*max_next_q-a.Q[offer].
    actions.get(maxaction).value_function);
a.Q[offer].actions.get(maxaction).value_function-=
(count/2*0.005);
}
else if(maxaction==5){//accept satisfying offer
    a.Q[offer].actions.get(maxaction).value_function
    +=learning_rate*(0.35+discount*max_next_q-a.Q[offer].
    actions.get(maxaction).value_function);
    a.Q[offer].actions.get(maxaction).value_function+=0.05;
}else {
    a.Q[offer].actions.get(maxaction).value_function
    +=learning_rate*(0.7+discount*max_next_q-a.Q[offer].
    actions.get(maxaction).value_function);
}
this.action=maxaction;

return maxaction;

}

}

//the other agent decide for our possible action
int decide_for_offer2(Agent a,double area,int offer){
    if(offer==0){

```

```

        if(area==0)return
make_offer(area,a.Q[0].actions.get(0).counteroffer);
        else {int maxaction=0;
double max=0;
for(int i=1;i<=3;++i)
{

        if(a.Q[0].actions.get(i).value_function>max)
            if(area<init_area&&(i==3)){
                else
                {max=a.Q[0].actions.get(i).value_function;
maxaction=i;
                }
            }
        return
make_offer(area,a.Q[0].actions.get(maxaction).counteroffer);}
    }
    else {
        if(offer==1)offer=(int)(area-Math.round((satisfied*area/100)));
        int maxaction=0;
        double max=0;
        for(int i=0;i<a.Q[offer].actions.size();++i)
        {
            if(a.Q[offer].actions.get(i).value_function>max)
                if(area<init_area&&(i==4||i==5)){
                    else{max=a.Q[offer].actions.get(i).value_function;
maxaction=i;
                    }
                }
            }
        return
make_offer(area,a.Q[offer].actions.get(maxaction).counteroffer);
    }

```

```

    }
    //in this function the agent make offer
    int make_offer(double area,String type){

        if(type.compareTo(">0.5")==0){//code for counteroffer less
        // than satisfying amount of other agent
            double r = (Math.random() * (area - area*0.51))+area*0.51 ;
            return (int)r;

        }
        else if(type.compareTo("<0.5")==0){//offer greater
        //than satisfying amount of other agent
            double r = (Math.random() * (area*0.5 - 0)) ;
            return (int)r;

        }
        else if(type.compareTo("=")==0){//offer equal
        // with the satisfying amount of other agent
            return -1;

        }
        else return 0;

    }
}

```

KΛΑΣΗ SARSA_Agent

```

//this class represents the SARSA agent
public class SARSA_Agent extends Agent{

    double discount;//discount factor of agent
    double learning_rate;//learning rate of agent

```

```

double e;//epsilon possibility for explore
//constructor
public SARSA_Agent(int area,int satisfied,
int adv,double e,double learning_rate,double discount_factor){
    this.discount=discount_factor;
    this.learning_rate=learning_rate;
    this.satisfied=satisfied;
    this.e=e;
    Q=new State[area];
    state=0;action=0;
    prevstate=0;
    prevaction=0;
    this.adv=adv;
    //initialization of agent states depending on its advantage
    if(adv==0){
        for(int i=0;i<area;i++){
            Q[i]=new State();
            init_area=area;
            if(i!=0){
                Q[i].actions.add(new
Action("Refuse","<0.5",0.5));
                Q[i].actions.add(new Action("Accept",">",0.5));
                Q[i].actions.add(new
Action("Refuse",">0.5",0.5));
                Q[i].actions.add(new Action("Accept","<",0.5));
                Q[i].actions.add(new Action("Refuse","=",0.5));
                Q[i].actions.add(new Action("Accept","=",0.5));
            }
        }
    }
}

```

```

        else{
            Q[i].actions.add(new
Action("Accept","0",0));

            Q[i].actions.add(new Action("Refuse","<0.5",0.5));
            Q[i].actions.add(new Action("Refuse",">0.5",0.5));
            Q[i].actions.add(new Action("Refuse","=",0.5));
        }
    }

    //advantage agent
    else {
        for(int i=0;i<area;i++){
            Q[i]=new State();
            init_area=area;
            if(i!=0){
                Q[i].actions.add(new
Action("Refuse","<0.5",0.5));

                Q[i].actions.add(new Action("Accept",">",0.5));

                Q[i].actions.add(new
Action("Refuse",">0.5",0.5));

                Q[i].actions.add(new Action("Accept","<",0.5));

                Q[i].actions.add(new Action("Refuse","=",0.5));

                Q[i].actions.add(new Action("Accept","=",0.5));
                Q[i].actions.add(new Action("Refuse","*",1));
            }
        }
        else{
            Q[i].actions.add(new
Action("Accept","0",0));

```

```

        Q[i].actions.add(new Action("Refuse", "<0.5", 0.5));
        Q[i].actions.add(new Action("Refuse", ">0.5", 0.5));
        Q[i].actions.add(new Action("Refuse", "=", 0.5));
        Q[i].actions.add(new Action("Refuse", "*", 1));
    }
}

}

//this function return the maximum value of next state
double max_q(int offer, double area, State[] table){
    if(offer==0){
        if(area==0) return table[0].actions.get(0).value_function;
        else {

            double max=0;
            for(int i=1; i<table[0].actions.size(); ++i)
                if(table[0].actions.get(i).value_function>max)
                    if(area<init_area&& i==3){}
                    else

max=table[0].actions.get(i).value_function;
            return max;}
    }
    //if offer is greater than 0
    else {if(offer==-1) offer=(int)(Math.round((satisfied*area/100)));
    double max=0;

    for(int i=0; i<table[offer].actions.size(); ++i){
        if(table[offer].actions.get(i).value_function>max)

```

```

        if(area<init_area&&(i==4||i==5)){
            else max=table[offer].actions.get(i).value_function;
        }
        return max;

    }}

//in this function the agent decide what action
//it will take from current state
int decide_for_offer(Agent a,Agent b,
int offer,double area,int iterations,int count){
    boolean flag=false;
    boolean flag2=false;
    //if offer is zero
    if(offer==0||(offer==-1&&area==1)){offer=0;
    prevstate=state;
    prevaction=action;
    state=0;
    if(area==0){this.action=0;return 0;}//if area is zero
    else{
        //if area is greater than zero

        double max_next_q=0;
        int maxaction=1;
        double max=a.Q[0].actions.get(1).value_function;;
        for(int i=1;i<a.Q[0].actions.size();++i)
            if(area<init_area&&(i==3)){
                else
                {double t=a.Q[0].actions.get(i).value_function;
                if(t>max){
                    maxaction=i;
                    max=t;
                }
            }

String tmp1=a.Q[0].actions.get(maxaction).counteroffer;

```



```

int tmp2=make_offer(area,tmp1);

int tmp3=decide_for_offer2(b,area,tmp2);
max_next_q=max_q(tmp3,area,a.Q);

double r=Math.random();
int r2;
if(tmp3==1)
    tmp3=(int)(Math.round((satisfied*area/100)));
r2=(int)(Math.random()*(a.Q[tmp3].actions.size()-1));
if(r<e)
    max_next_q=a.Q[tmp3].actions.get(r2).value_function;
if(b.Q[b.state].actions.get(b.action).
counteroffer.compareTo("")==0){
}
else{
    if(maxaction==1)
        {if(adv==1)
            a.Q[0].actions.get(maxaction).value_function
            +=learning_rate*(0.5+discount*max_next_q-a.Q[0].
            actions.get(maxaction).value_function);
        else a.Q[0].actions.get(maxaction).value_function
            +=learning_rate*(0.3+discount*max_next_q-a.Q[0].
            actions.get(maxaction).value_function);}
    else if(maxaction==2)
        a.Q[0].actions.get(maxaction).value_function
        +=learning_rate*(0.35+discount*max_next_q-a.Q[0].
        actions.get(maxaction).value_function);
    else if(maxaction==3)
        a.Q[0].actions.get(maxaction).value_function
        +=learning_rate*(0.55+discount*max_next_q-a.Q[0].
        actions.get(maxaction).value_function);
    else {a.Q[0].actions.get(maxaction).value_function

```

```

        +=learning_rate*(0.5+discount*max_next_q-a.Q[0].
        actions.get(maxaction).value_function);}
    }          this.action=maxaction;
    return maxaction;

}
}

//if offer is greater than zero
else {
    boolean bigger=false;
    boolean smaller=false;
    if(offer==1||(offer==0.5*area)){//checking if offer is demanded
        flag=true;
        offer=(int)(Math.round((satisfied*area/100)));
    }
    else if(offer<(0.5*area))smaller=true;
    else bigger=true;

    prevaction=action;
    prevstate=state;
    state=offer;
    int maxaction=0;
    double max=0;
    double max_next_q=0;
    for(int i=0;i<a.Q[offer].actions.size();++i)
    {
        if(flag&&(i==1||i==3||i==4)){

        }else if((i==4||i==5)&&(bigger||smaller)&&
        (area<((double)init_area))){}
        else if(bigger&&(i==3||i==5)){}
        else if(smaller&&(i==1||i==5)){}
        else {

```

```

        double r=Math.random();
        if(true){

if(a.Q[offer].actions.get(i).value_function>max)

{max=a.Q[offer].actions.get(i).value_function;
                    maxaction=i;
                    }
                }

        }
    }

String tmp1=a.Q[offer].actions.get(maxaction).counteroffer;
int tmp2=make_offer(area,tmp1);
int tmp3=decide_for_offer2(b,area,tmp2);
max_next_q=max_q(tmp3,area,a.Q);

double r=Math.random();
int r2;
if(tmp3==1)
    tmp3=(int)(Math.round((satisfied*area/100)));
r2=(int)(Math.random()*(a.Q[tmp3].actions.size()-1));
if(r<e)
    max_next_q=a.Q[tmp3].actions.get(r2).value_function;

if(maxaction==0)//refuse and make offer < satisfying of other
agent

{if(adv==1)
    a.Q[offer].actions.get(maxaction).value_function
+=learning_rate*(0.5+discount*max_next_q-a.Q[offer].
actions.get(maxaction).value_function);
else    a.Q[offer].actions.get(maxaction).value_function

```

```

+=learning_rate*(0.3+discount*max_next_q-a.Q[offer].
actions.get(maxaction).value_function);

a.Q[offer].actions.get(maxaction).value_function
-=(count/2*0.005);
}
else if(maxaction==1){//accept the > of satisfying offer
    a.Q[offer].actions.get(maxaction).value_function
    +=learning_rate*(0.55+discount*max_next_q-a.Q[offer].
    actions.get(maxaction).value_function);

a.Q[offer].actions.get(maxaction).value_function+=0.005;;
}
else if(maxaction==2){//refuse and make offer <
//satisfying of other agent
    a.Q[offer].actions.get(maxaction).value_function
    +=learning_rate*(0.35+discount*max_next_q-a.Q[offer].
    actions.get(maxaction).value_function);
    a.Q[offer].actions.get(maxaction).value_function
    -=(count/2*0.005);
}
else if(maxaction==3){//accept the <
//of the satisfying amount offer
    a.Q[offer].actions.get(maxaction).value_function
    +=learning_rate*(0.35+discount*max_next_q-a.Q[offer].
    actions.get(maxaction).value_function);
    a.Q[offer].actions.get(maxaction).value_function+=0.005;
}
else if(maxaction==4){//make satisfying offer
    a.Q[offer].actions.get(maxaction).value_function
    +=learning_rate*(0.35+discount*max_next_q-a.Q[offer].
    actions.get(maxaction).value_function);
    a.Q[offer].actions.get(maxaction).value_function

```

```

        -= (count/2*0.005);
    }
    else if(maxaction==5){//accept the satisfying offer
        a.Q[offer].actions.get(maxaction).value_function
        +=learning_rate*(0.35+discount*max_next_q-a.Q[offer].
        actions.get(maxaction).value_function);
        a.Q[offer].actions.get(maxaction).value_function+=0.05;
    }else {
        a.Q[offer].actions.get(maxaction).value_function
        +=learning_rate*(0.7+discount*max_next_q-a.Q[offer].
        actions.get(maxaction).value_function);
    }
    this.action=maxaction;
    return maxaction;

}

}

//in this function we see what the other agent
//would have decide for our counter offer in order
// to see what action it will take next
int decide_for_offer2(Agent a,double area,int offer){
    if(offer==0){
        if(area==0)
            return
make_offer(area,a.Q[0].actions.get(0).counteroffer);
        else {int maxaction=0;
        double max=0;
        for(int i=1;i<=3;++i)
        {
            if(a.Q[0].actions.get(i).value_function>max)
                if(area<init_area&&(i==3)){
                    else

```

```

        {max=a.Q[0].actions.get(i).value_function;
        maxaction=i;
        }
    }
    return
make_offer(area,a.Q[0].actions.get(maxaction).counteroffer);}
}
else {
    if(offer==-1)
        offer=(int)(area-Math.round((satisfied*area/100)));
    int maxaction=0;
    double max=0;
    for(int i=0;i<a.Q[offer].actions.size();++i)
    {

        if(a.Q[offer].actions.get(i).value_function>max)
            if(area<init_area&&(i==4||i==5)){
                else{

max=a.Q[offer].actions.get(i).value_function;
                maxaction=i;

                }
            }
        return make_offer(area,a.Q[offer].actions.
get(maxaction).counteroffer);
    }
}
//in this function the agent decides what offer to make
int make_offer(double area,String type){

    if(type.compareTo(">0.5")==0){//code for offer
    //greater than satisfying amount of other agent
        double r = (Math.random() * (area - area*0.51))+area*0.51 ;

```

```

        return (int)r;

    }

    else if(type.compareTo("<0.5")==0){//code for offer
//      less than satisfying amount of other agent

        double r = (Math.random() * (area*0.5 - 0)) ;

        return (int)r;

    }

    else if(type.compareTo("=")==0){//code of satisfying
//      offer for the other agent

        return -1;

    }

    else return 0;

}

```

ΠΑΡΑΡΤΗΜΑ Β: ΠΑΡΑΔΕΙΓΜΑΤΑ ΑΡΧΕΙΩΝ ΕΙΣΟΔΟΥ

Τα πύό κάτω παραραδείγματα αναφέρονται για το αρχείο input.txt απο το οποίο το πρόγραμμα διαβάζει τις παραμέτρους της προσομοίωσης

ΠΑΡΑΔΕΙΓΜΑ 1 (Q_learning)

Simulation options

Iterations: 900

Algorithm: Q_learning

learning_rate_EK: 0.5

learning_rate_TK: 0.5

discount_factor_EK: 0.6

discount_factor_TK: 0.6

epsilon: 0.3

ΠΑΡΑΔΕΙΓΜΑ 2(SARSA)

Simulation options

Iterations: 900

Algorithm: SARSA

learning_rate_EK: 0.5

learning_rate_TK: 0.5

discount_factor_EK: 0.6

discount_factor_TK: 0.6

epsilon: 0.3