Συστηματική Αναζήτηση και Ενισχυτική Μάθηση για το Επιτραπέζιο Παιχνίδι Backgammon

Στέλιος Τσιγδινός

Σχολή Ηλεκτρονικών Μηχανικών & Μηχανικών Υπολογιστών

Πολυτεχνείο Κρήτης

Εξεταστική Επιτροπή:

Αν. Καθ. Μιχαήλ Γ. Λαγουδάκης (επιβλέπων)

Καθ. Μιχαήλ Ε. Ζερβάκης

Καθ. Ευριπίδης Γ.Μ Πετράκης

Χανιά 2014

Περίληψη

Τα παιχνίδια απασχολούσαν, από τότε που υπάρχει πολιτισμός, τις διανοητικές λειτουργίες του ανθρώπου. Στα πλαίσια της Τεχνητής Νοημοσύνης, η αφηρημένη φύση των παιχνιδιών καθώς και η δυσκολία επίλυσής τους τα καθιστά ένα ενδιαφέρον πεδίο μελέτης.

Στην παρούσα διπλωματική εργασία υλοποιούμε ένα πράκτορα για το επιτραπέζιο παιχνίδι Backgammon καθώς και ένα γραφικό περιβάλλον στο οποίο μπορούν να διεξαχθούν παρτίδες του παιχνιδιού αυτού με αντίπαλο τον πράκτορα μας. Σκοπός μας, είναι η εύρεση μιας καλής στρατηγικής (policy), η οποία θα επιτρέπει στον πράκτορά μας να αυξήσει τις πιθανότητές του, με τη κατάλληλη επιλογή κινήσεων, να οδηγηθεί σε μία τερματική κατάσταση νίκης. Ο μεγάλος παράγοντας διακλάδωσης του δέντρου αναζήτησης για το παιχνίδι αυτό, που πολλές φορές μπορεί να φτάσει μέχρι και κάποιες εκατοντάδες κινήσεις, καθώς και το στοιχείο της τύχης που υπάρχει στη φύση του παιχνιδιού, λόγω του ότι χρησιμοποιούνται ζάρια για την υπόδειξη των δυνατών αποστάσεων στις κινήσεις των δύο αντιπάλων, αυξάνει σημαντικά την δυσκολία αναζήτησης και εύρεσης της βέλτιστης αυτής στρατηγικής. Η στρατηγική αυτή θα προσδιορίζει ουσιαστικά την συμπεριφορά του πράκτορα κατά την διάρκεια του παιχνιδιού.

Χρησιμοποιώντας ειδικές τεχνικές αναζήτησης, όπως αυτή του αλγόριθμου MiniMax και κάποιες παραλλαγές του όπως αυτή του αλγόριθμου AlphaBeta πετύχαμε αποδεκτές ταχύτητες αναζήτησης σε ικανοποιητικό βάθος, στο δέντρο αναζήτησης του παιχνιδιού. Η συστηματική αναζήτηση σε συνδυασμό με τη χρήση τεχνικών από το πεδίο της ενισχυτικής μάθησης (Reinforcement Learning), οδήγησαν στην εύρεση μιας στρατηγικής, η οποία επιτρέπει στον πράκτορά μας να ανταγωνιστεί αρκετά καλούς φυσικούς αλλά και τεχνητούς παίκτες στο παιχνίδι Backgammon.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

1.1 Τεχνητή Νοημοσύνη και Παιχνίδια

Ο ακριβής ορισμός της ανθρώπινης νοημοσύνης είναι μάλλον μια άλυτη ακόμα εξίσωση. Ο λόγος που η έννοια αυτή παραμένει ασαφής είναι επειδή ο άνθρωπος δεν έχει καταφέρει ακόμα να ανακαλύψει όλες τις πτυχές και τις παραμέτρους της. Σε γενικές όμως γραμμές θα μπορούσαμε να πούμε ότι είναι η ικανότητα της επεξεργασίας διάφορων πληροφοριών, σε διάφορα επίπεδα, και η εξαγωγή λογικών συμπερασμάτων και αποτελεσμάτων μετά από αυτήν τη διαδικασία.

Μέσα από την αγωνία του ανθρώπου να δημιουργήσει κάτι, το οποίο θα μπορούσε να μιμηθεί κάποιες από τις συμπεριφορές του αλλά και κάποιες από τις διανοητικές του λειτουργίες, "γεννήθηκε" η έννοια της Τεχνητής Νοημοσύνης στον κλάδο της επιστήμης των υπολογιστών. Στόχος λοιπόν της Τεχνητής Νοημοσύνης είναι η δημιουργία τέτοιων συμπεριφορών, είτε από προγράμματα-πράκτορες είτε από μηχανήματα-ρομπότ, οι οποίες τουλάχιστον θα υπονοούν κάποια βασική ευφυΐα. Η ευφυΐα αυτή μπορεί να αποτελείται από στοιχεία ικανότητας μάθησης, εξαγωγής συμπερασμάτων, επίλυσης προβλημάτων και πολλά άλλα.

Μία από τις μεγαλύτερες διανοητικές προκλήσεις του ανθρώπου, ανά τους αιώνες, ήταν και είναι τα πνευματικά παιχνίδια. Η πολυπλοκότητα και η δυσκολία που αυτή συνεπάγεται, όσον αφορά την επίλυση των απαιτούμενων προβλημάτων αλλά και την λήψη των κατάλληλων αποφάσεων κατά την διάρκεια ενός παιχνιδιού, είναι τα στοιχεία τα οποία "οικοδόμησαν" το μεγάλο αυτό ενδιαφέρον του ανθρώπου να ασχοληθεί μαζί τους.

Έχοντας υπόψη μας όλα τα παραπάνω, θα ήταν παράδοξο εάν στο "χώρο" της Τεχνητής Νοημοσύνης δεν καταλάμβανε μεγάλο μερίδιο το κομμάτι των παιχνιδιών. Η σχέση της Τεχνητής Νοημοσύνης με τα παιχνίδια είναι μια σχέση αμφίδρομη. Η

μαθηματική μοντελοποίηση των προβλημάτων που ανακύπτουν από την ανάγκη λήψης καθοριστικών αποφάσεων κατά την διάρκεια ενός παιχνιδιού, αλλά και οι σύγχρονοι αλγοριθμικοί μέθοδοι επίλυσης τέτοιων προβλημάτων είναι τα στοιχεία της Τεχνητής Νοημοσύνης, τα οποία έχουν προσφέρει σημαντικά στην πρόοδο όσον αφορά την βέλτιστη συμπεριφορά σε ένα περιβάλλον παιχνιδιού. Από την άλλη, η φύση των παιχνιδιών και η πολυπλοκότητα της επίλυσής τους, έδωσαν και συνεχίζουν να δίνουν το έναυσμα για την δημιουργία νέων "δρόμων" στην συλλογιστική της επίλυσης τέτοιων σύνθετων προβλημάτων.

1.2 Περιγραφή της εργασίας

Στα πλαίσια της παρούσας διπλωματικής εργασίας υλοποιήθηκε ένας πράκτορας για το παιχνίδι Backgammon, καθώς επίσης και ένα γραφικό περιβάλλον με το οποίο μπορούν να διεξαχθούν παρτίδες του παιχνιδιού αυτού με αντίπαλο τον πράκτορά μας. Η φύση του παιχνιδιού, ως προς την πολυπλοκότητα και την αλληλεξάρτηση των κινήσεων ανάμεσα στους δύο αντιπάλους, καθώς και το στοιχείο της τύχης που προκύπτει από τη ρίψη ζαριών, μας οδήγησε στην χρησιμοποίηση ειδικών τεχνικών αναζήτησης πάνω στο δέντρο αναζήτησης του προβλήματος. Ο πράκτοράς μας επιτυγχάνει αρκετά ικανοποιητικές ταχύτητες αναζήτησης, φτάνοντας και σε βάθος 5 στρώσεων στο δέντρο του παιχνιδιού. Με την χρήση τεχνικών ενισχυτικής μάθησης βελτιώνεται η στρατηγική του συνεχώς καθώς παίζει. Ύστερα από δεκάδες χιλιάδες εκπαιδευτικά παιχνίδια απέναντι στον εαυτό του καταλήξαμε σε μία ικανοποιητική στρατηγική με την οποία ο πράκτοράς μας είναι σε θέση να αντιμετωπίσει ακόμα και φυσικούς παίκτες κάποιας εμπειρίας στο παιχνίδι Backgammon.

Τα αποτελέσματα από το εγχείρημα της χρησιμοποίησης των συγκεκριμένων μεθόδων αναζήτησης και μάθησης, ως προς τον χρόνο απόκρισης αλλά και ως προς την απόδοση του πράκτορά μας απέναντι σε πραγματικούς παίκτες και άλλους πράκτορες, απέδειξαν την αποτελεσματικότητα τους σε πραγματικά σύνθετα προβλήματα. Τόσο η μέθοδος αναζήτησης Expectiminimax, σε συνδυασμό με το κλάδεμα Άλφα-Βήτα, όσο και η μέθοδος μάθησης Χρονικών Διαφορών (TD) που χρησιμοποιήσαμε, μπόρεσαν να ανταποκριθούν αρκετά καλά απέναντι στην πρόκληση ενός τόσο απρόβλεπτου και πολύπλοκου παιχνιδιού όπως το Backgammon. Ωστόσο, εκτός από την αξία της απόδοσης του πράκτορά μας, η εργασία αυτή αποτελεί και μία πολύ καλή "βάση" για την περαιτέρω εξέλιξη και μελέτη του

συγκεκριμένου παιχνιδιού με εναλλακτικούς αλγόριθμους τόσο στο κομμάτι της συστηματικής αναζήτησης όσο και στο κομμάτι της ενισχυτικής μάθησης.

1.3 Διάρθρωση της εργασίας

Στο δεύτερο κεφάλαιο αρχικά αναφερόμαστε σε κάποιες βασικές τεχνικές πάνω στις οποίες στηρίζονται πολλοί από τους σύγχρονους αλγόριθμους αναζήτησης. Στην συνέχεια παρουσιάζονται οι συνιστώσες που ορίζουν κάποιο παιχνίδι σαν πρόβλημα αναζήτησης καθώς επίσης και το δέντρο παιχνιδιού και πως διαμορφώνεται αυτό όταν εμπλέκεται κάποιο στοιχείο τύχης. Ακολουθεί η ανάλυση του αλγόριθμου MiniMax ο οποίος χρησιμοποιείται σε προβλήματα αναζήτησης με αντιπαλότητα, όπου οι στόχοι κάθε πράκτορα έρχονται σε σύγκρουση με τους στόχους των υπόλοιπων πρακτόρων. Παρουσιάζονται κάποιες παραλλαγές του αλγόριθμου αυτού όπως ο ExpectiMiniMax ο οποίος χρησιμοποιείται σε περιβάλλοντα στα οποία υπάρχει κάποιος απρόβλεπτος παράγοντας, καθώς επίσης και ο Άλφα-Βήτα ο οποίος χρησιμοποιώντας την τεχνική του κλαδέματος καταφέρνει να μειώσει δραστικά τον παράγοντα διακλάδωσης του δέντρου αναζήτησης. Τέλος γίνεται αναφορά στην συνάρτηση αξιολόγησης και πώς αυτή επηρεάζει την συμπεριφορά του πράκτορα μας, καθώς επίσης και στην μέθοδο χρονικών διαφορών η οποία προέρχεται από τον χώρο της ενισχυτικής μάθησης και χρησιμοποιείται για την εκπαίδευση ενός πράκτορα αναφορικά στην εύρεση εκείνης της στρατηγικής που θα του αποφέρει το μέγιστο κέρδος.

Στο τρίτο κεφάλαιο αναφέρουμε κάποια ιστορικά στοιχεία για την προέλευση του παιχνιδιού Backgammon και κάποια στοιχεία για την πολυπλοκότητά του. Παρουσιάζονται οι στόχοι της εργασίας αυτής καθώς επίσης και οι κυριότερες εργασίες που έχουν γίνει για το παιχνίδι αυτό μέχρι τώρα, όπως αυτή του Gerry Tesauro ο οποίος υλοποίησε το TD-Gammon.

Στο τέταρτο κεφάλαιο παρουσιάζουμε τον τρόπο με τον οποίο εμείς προσεγγίσαμε το πρόβλημα της σχεδίασης και εκπαίδευσης ενός πράκτορα για το παιχνίδι Backgammon. Αναφέρονται τα χαρακτηριστικά στα οποία καταλήξαμε να χρησιμοποιήσουμε για την συνάρτηση αξιολόγησης του πράκτορά μας, όπως επίσης και την μεθοδολογία που ακολουθήσαμε στην προσπάθειά μας να τον εκπαιδεύσουμε με στόχο την βελτιστοποίηση των κερδών του.

Το πέμπτο κεφάλαιο αναλύει τα σημαντικότερα προβλήματα τα οποία συναντήσαμε κατά την υλοποίηση του πράκτορά μας και πως τα αντιμετωπίσαμε. Ιδιαίτερο βάρος δίνεται στην πολυπλοκότητα των νόμιμων κινήσεων των δύο παικτών και πως διαφοροποιούνται αυτές αναλόγως με την κατάσταση του παιχνιδιού.

Στο έκτο κεφάλαιο παρουσιάζονται τα αποτελέσματα από την διαδικασία της εκπαίδευσης του πράκτορά μας. Αρχικά αναφερόμαστε στην σύγκριση κάποιων συνόλων από διαφορετικά χαρακτηριστικά για την συνάρτηση αξιολόγησης ενώ στην συνέχεια παρατίθενται τα αποτελέσματα από την σύγκριση του πράκτορά μας απέναντι σε φυσικούς παίκτες διαφόρων επιπέδων εμπειρίας αλλά και άλλους πράκτορες-προγράμματα.

Κεφάλαιο 2

Αναγκαίο γνωστικό υπόβαθρο

2.1 Αναζήτηση

2.1.1 Δέντρο αναζήτησης

Στην επιστήμη των μαθηματικών η έννοια του γράφου ορίζεται ως η οπτική απεικόνιση η οποία αποτελείται από ένα σύνολο στοιχείων, τα οποία συνήθως ονομάζονται κόμβοι, καθώς επίσης και ένα σύνολο ακμών τα οποία συνδέουν κάποια ζεύγη από αυτούς τους κόμβους. Η ακολουθία δύο ή περισσότερων συνεχόμενων τέτοιων συνδέσμων ονομάζεται μονοπάτι ή διαδρομή. Στην επιστήμη της πληροφορικής η ανάγκη της μαθηματικής μοντελοποίησης κάποιων προβλημάτων αναζήτησης, με σκοπό την επίλυσή τους, έκανε απαραίτητη την υιοθέτηση της θεωρίας των γράφων. Κάθε κόμβος λοιπόν συνήθως αντιπροσωπεύει και ένα μεμονωμένο στιγμιότυπο σε ένα τέτοιο πρόβλημα, το οποίο συνήθως ονομάζεται κατάσταση, ενώ οι ακμές αντιπροσωπεύουν τις δυνατές μεταβάσεις από μια τέτοια κατάσταση σε μία άλλη, και ονομάζονται συνήθως ενέργειες.

Τα δέντρα αναζήτησης είναι μία από τις συνηθέστερες μορφές γράφων που χρησιμοποιούνται σε προβλήματα αναζήτησης και η διαφοροποίησή τους από οποιονδήποτε άλλο γράφο έγκειται στις παρακάτω ιδιότητες :

- Οι μεταβάσεις μεταξύ των κόμβων έχουν κατεύθυνση.
- Οποιοιδήποτε δύο κόμβοι συνδέονται μεταξύ τους με ένα μοναδικό μονοπάτι.
- Ακολουθώντας οποιοδήποτε μονοπάτι του δέντρου δε μπορεί να σχηματιστεί κάποιος “κύκλος”, δηλαδή δεν μπορούμε να καταλήξουμε στον ίδιο κόμβο από τον οποίο αρχίσαμε.

- Υπάρχει ένας κόμβος ο οποίος συνδέεται μόνο με ακμές που καταλήγουν σε άλλους κόμβους.

- Υπάρχει τουλάχιστον ένας κόμβος από τον οποίο δεν ξεκινάει καμία ακμή που να καταλήγει σε άλλο κόμβο.

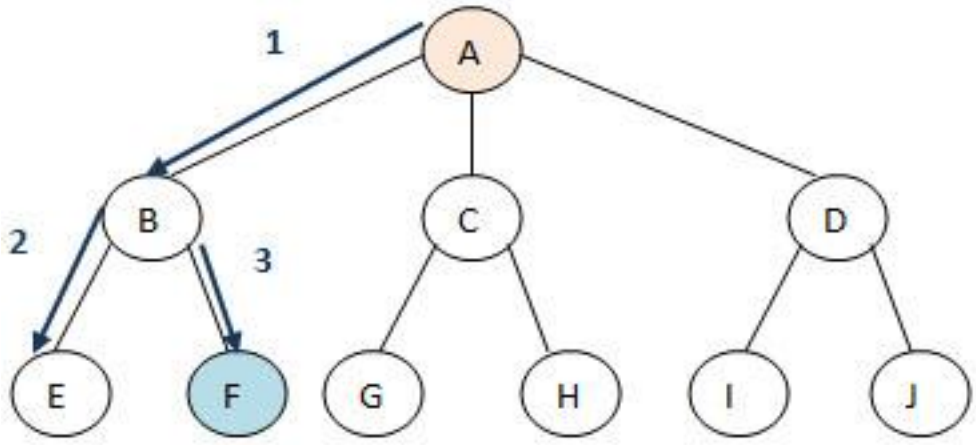
Λόγω των κατευθυνόμενων μεταβάσεων μεταξύ των κόμβων ενός δέντρου, εύκολα μπορούμε να δώσουμε τους χαρακτηρισμούς του κόμβου-απογόνου και του κόμβου-προγόνου. Σε ένα δέντρο, ο κόμβος ο οποίος δεν έχει κόμβους-προγόνους ονομάζεται ρίζα του δέντρου ή αρχική κατάσταση, ενώ οι κόμβοι οι οποίοι δεν έχουν κόμβους-απογόνους ονομάζονται κόμβοι-φύλλα του δέντρου. Σε ένα συνηθισμένο πρόβλημα αναζήτησης η αρχική κατάσταση εκκίνησης της αναζήτησης αυτής είναι ο κόμβος-ρίζα του δέντρου, ενώ ο στόχος της αναζήτησης βρίσκεται σε κάποιον ή κάποιους από τους κόμβους-φύλλα του.

2.1.2 Αναζήτηση πρώτα σε βάθος

Η αναζήτηση πρώτα σε βάθος (depth first search) [1], είναι μία μέθοδος τυφλής αναζήτησης (blind search). Οι μοναδικές πληροφορίες δηλαδή οι οποίες παρέχονται είναι αυτές που ορίζει το ίδιο το πρόβλημα. Ευρισκόμενοι σε οποιονδήποτε κόμβο του δέντρου αναζήτησης του προβλήματος, η στρατηγική της συγκεκριμένης μεθόδου δίνει προτεραιότητα για επέκταση στον πρώτο βαθύτερο ανεξερεύνητο κόμβο και δημιουργεί όλους τους κόμβους-παιδιά αυτού. Η αναζήτηση προχωράει αναδρομικά μέχρις ότου φτάσουμε σε κάποιο παιδί-φύλλο, όπου εκεί η αναζήτηση "οπισθοχωρεί" στον αμέσως ρηχότερο κόμβο που έχει ανεξερεύνητους διαδόχους.

Ένα παράδειγμα της στρατηγικής αναζήτησης πρώτα σε βάθος φαίνεται στο σχήμα 2.1. Αρχίζοντας από τον κόμβο-ρίζα "A", του δέντρου αναζήτησης, δημιουργούμε τους κόμβους-παιδιά "B", "C" και "D". Σαν επόμενος κόμβος για επέκταση επιλέγεται ο "B" από τον οποίο δημιουργούνται οι διάδοχοι κόμβοι "E" και "F". Ακολουθώντας την ίδια λογική ο κόμβος "E" είναι ο επόμενος κόμβος ο οποίος επιλέγεται για εξερεύνηση. Επειδή όμως ο συγκεκριμένος κόμβος είναι φύλλο και δεν επιδέχεται περεταίρω επέκταση, η αναζήτηση "προχωράει" προς εξερεύνηση στον επόμενο διαθέσιμο βαθύτερο κόμβο, ο οποίος είναι ο "F". Προχωρώντας η αναζήτηση, θα επιλεγεί σαν επόμενος κόμβος προς επέκταση ο κόμβος "C" και αυτό

γιατί είναι ο αμέσως ρηχότερος κόμβος με ανεξερεύνητους διαδόχους αφού ο κόμβος "B", μετά την εξερεύνηση του "F", δεν έχει άλλο διάδοχο. Η διαδικασία ακολουθείται μέχρις ότου βρεθεί ο κόμβος-στόχος.



Σχήμα 2.1 : Παράδειγμα επέκτασης κόμβων για την αναζήτηση πρώτα σε βάθος.

2.1.3 Αναζήτηση με υπαναχώρηση

Όπως είδαμε και στο παράδειγμα του Σχήματος 2.1, για την εξερεύνηση ενός κόμβου παιδιού, ακολουθώντας πάντα την στρατηγική της αναζήτησης πρώτα σε βάθος, πρέπει να επεκταθούν όλοι οι κόμβοι παιδιά από τον ίδιο κόμβο-γονέα. Επίσης, όλοι αυτοί οι προς επέκταση κόμβοι στα διάφορα σημεία του δέντρου πρέπει να διατηρούνται σε κάποια δομή, έως ότου έρθει η σειρά τους να επεκταθούν. Το γεγονός αυτό επιβαρύνει σημαντικά την μνήμη του συστήματος, και κατά συνέπεια την όλη απόδοση του πράκτορά μας ο οποίος εκτελεί την αναζήτηση αυτή.

Στο παραπάνω πρόβλημα έρχεται να δώσει λύση μία επέκταση της αναζήτησης πρώτα σε βάθος η οποία ονομάζεται αναζήτηση με υπαναχώρηση [1]. Με την μέθοδο αυτή παράγεται μόνο ένας διάδοχος κόμβος κάθε φορά στον οποίο γίνεται αναδρομική κλήση και όχι όλοι οι διάδοχοι. Εφόσον τελειώσει αναδρομικά η μερική αναζήτηση στο υποδέντρο του κόμβου που επεκτάθηκε, ο κόμβος γονέας δημιουργεί τον επόμενο κόμβο διάδοχο στον οποίο γίνεται νέα αναδρομική κλήση και με αυτόν τον τρόπο συνεχίζεται η διαδικασία της αναζήτησης.

Η παραπάνω διαδικασία προϋποθέτει φυσικά την ανάλογη υποστήριξη από το πρόγραμμα-πράκτορα. Τα οφέλη όμως από την υλοποίηση της είναι σημαντικά, ειδικά σε δέντρα αναζήτησης στα οποία ο παράγοντας διακλάδωσης είναι πολύ μεγάλος, πόσο μάλλον όταν η μνήμη είναι ένας πολύ σημαντικός παράγοντας που καθορίζουν την απόδοση του πράκτορα αυτού. Πρέπει να σημειωθεί ότι η αναζήτηση πρώτα σε βάθος με ή χωρίς υπαναχώρηση διαπερνά ολόκληρο το δέντρο, οπότε ο χρόνος που απαιτείται είναι δεδομένος και εξαρτάται από το πλήθος των κόμβων σ'όλο το δέντρο.

2.1.4 Αναζήτηση περιορισμένου βάθους

Ένα από τα μειονεκτήματα της αναζήτησης πρώτα σε βάθος είναι ότι ο πράκτοράς μας μπορεί να αφιερώσει, άσκοπα, πολύ χρόνο σε ένα υποδέντρο του συνολικού δέντρου του προβλήματος μας, ενώ μια διαφορετική επιλογή θα μας οδηγούσε σε μία λύση η οποία μάλιστα μπορεί και να βρισκόταν πολύ κοντά στη ρίζα του δέντρου. Ειδικά σε προβλήματα, των οποίων το δέντρο αναζήτησης είναι πολύ μεγάλο και ο χρόνος απόκρισης του πράκτορα είναι πολύ σημαντικός παράγοντας όσον αφορά την απόδοσή του, η μέθοδος της αναζήτησης πρώτα σε βάθος, με ή χωρίς υπαναχώρηση, γίνεται απαγορευτική.

Για την επίλυση του παραπάνω προβλήματος χρησιμοποιούμε την μέθοδο της αναζήτησης περιορισμένου βάθους (depth-limited search) [1]. Ουσιαστικά με την μέθοδο αυτή παρέχουμε στην αναζήτηση πρώτα σε βάθος ένα προκαθορισμένο όριο βάθους *l,* ένα βάθος δηλαδή το οποίο δεν μπορεί ο πράκτοράς μας να ξεπεράσει κατά την διάρκεια της αναζήτησής του. Έτσι όλοι οι κόμβοι οι οποίοι βρίσκονται στο συγκεκριμένο αυτό προκαθορισμένο βάθος, αντιμετωπίζονται από αυτόν σαν να ήταν κόμβοι-φύλλα του δέντρου. Το κέρδος, σε χρόνο, που αποκομίζουμε χρησιμοποιώντας την μέθοδο αυτή, αντισταθμίζεται με το γεγονός ότι θυσιάζουμε την πληρότητα της αναζήτησης, δεδομένου ότι η λύση που αναζητούμε μπορεί να βρίσκεται σε βάθος μεγαλύτερο από αυτό που έχουμε θέσει ως όριο. Για να επιτευχθεί η πληρότητα, μπορούμε να επαναλάβουμε την αναζήτηση με σταδιακά αυξανόμενο όριο βάθους, έως το πραγματικό βάθος του δέντρου.

2.2 Τα παιχνίδια ως πρόβλημα αναζήτησης

2.2.1 Συνιστώσες ενός παιχνιδιού

Ο κλάδος των οικονομικών ο οποίος ασχολείται με την θεωρία παιγνίων, θεωρεί ως παιχνίδι ένα πολυπρακτορικό περιβάλλον, με την προϋπόθεση ότι η επίδραση του κάθε πράκτορα είναι "σημαντική" για τους υπόλοιπους ανεξάρτητα εάν αυτοί μεταξύ τους είναι συνεργατικοί ή ανταγωνιστικοί. Στα πλαίσια του πεδίου της Τεχνητής Νοημοσύνης, ως παιχνίδια αντιμετωπίζονται τα περιβάλλοντα τα οποία είναι αιτιοκρατικά και πλήρως παρατηρήσιμα, μέσα στα οποία δύο ή και περισσότεροι πράκτορες, οι ενέργειες των οποίων εναλλάσσονται, λειτουργούν ανταγωνιστικά και οι στόχοι τους συγκρούονται.

Ένα παιχνίδι ουσιαστικά μπορεί να αντιμετωπιστεί ως πρόβλημα αναζήτησης με αντιπαλότητα οριζόμενο από τις παρακάτω συνιστώσες :

- Την αρχική κατάσταση (initial state), η οποία περιλαμβάνει τη διάταξη των πιονιών, πάνω στο ταμπλό, των παικτών κατά την εκκίνηση του παιχνιδιού, ενδεχομένως την ένδειξη κάποιου ζαριού, καθώς επίσης και το ποιος παίκτης έχει την πρώτη κίνηση.

- Μια συνάρτηση διαδόχων (successors function), η οποία έχοντας σαν δεδομένο μία κατάσταση του παιχνιδιού, επιστρέφει όλες τις καταστάσεις στις οποίες θα επέλθει το παιχνίδι, εκτελώντας όλες τις νόμιμες, σύμφωνα με τους κανόνες του παιχνιδιού, κινήσεις της κατάστασης αυτής.

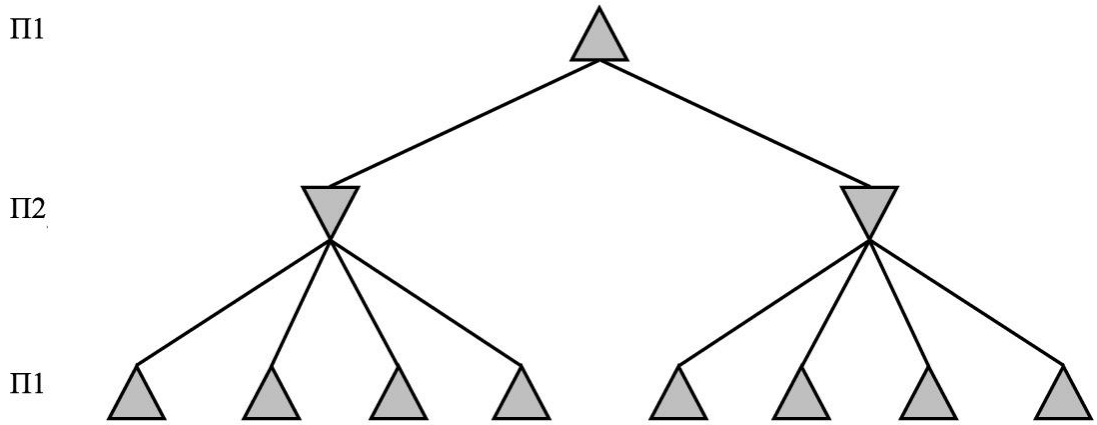
- Έναν έλεγχο τερματισμού (terminal test), ο οποίος ουσιαστικά εκτελείται μετά από κάθε νόμιμη κίνηση και προσδιορίζει το πότε τελειώνει το παιχνίδι. Οι καταστάσεις στις οποίες έχει τελειώσει το παιχνίδι ονομάζονται τερματικές καταστάσεις (terminal states).

- Μια συνάρτηση χρησιμότητας (utility function), η οποία δίνει μια αριθμητική τιμή στις τερματικές καταστάσεις. Οι τιμές αυτές ποικίλουν ανάλογα με το παιχνίδι, για τις καταστάσεις της νίκης, της ήττας και της ισοπαλίας, εάν αυτή ορίζεται από το παιχνίδι.

2.2.2 Δέντρο παιχνιδιού

Βάζοντας σαν στόχο την παρατήρηση και την αξιολόγηση των ενεργειών ενός πράκτορα μέσα σε ένα τέτοιο ανταγωνιστικό περιβάλλον, δημιουργείται η ανάγκη ορισμού του δέντρου του παιχνιδιού. Έχοντας σαν αφετηρία την αρχική κατάσταση του παιχνιδιού, και εφαρμόζοντας τη συνάρτηση διαδόχων σε αυτήν αλλά και σε όλους τους διαδόχους που θα προκύψουν στην συνέχεια, δημιουργούμε το δέντρο αναζήτησης του παιχνιδιού.

Ένα πολύ απλοϊκό παράδειγμα δέντρου παιχνιδιού φαίνεται στο Σχήμα 2.2. Ξεκινώντας από την αρχική κατάσταση ο παίκτης Π1 έχει δύο νόμιμες κινήσεις, δημιουργώντας αντίστοιχα δύο κόμβους στο δέντρο. Παίρνοντας σειρά ο παίκτης Π2, και έχοντας τέσσερις νόμιμες κινήσεις για κάθε ένα από τους κόμβους-καταστάσεις που προέκυψαν δημιουργεί τους επόμενους κόμβους στο δέντρο του παιχνιδιού. Όπως μπορούμε εύκολα να παρατηρήσουμε οι κινήσεις των παικτών εναλλάσσονται μεταξύ τους δημιουργώντας έτσι κάποια επίπεδα, κάθε ένα από τα οποία ονομάζεται στρώση (ply).



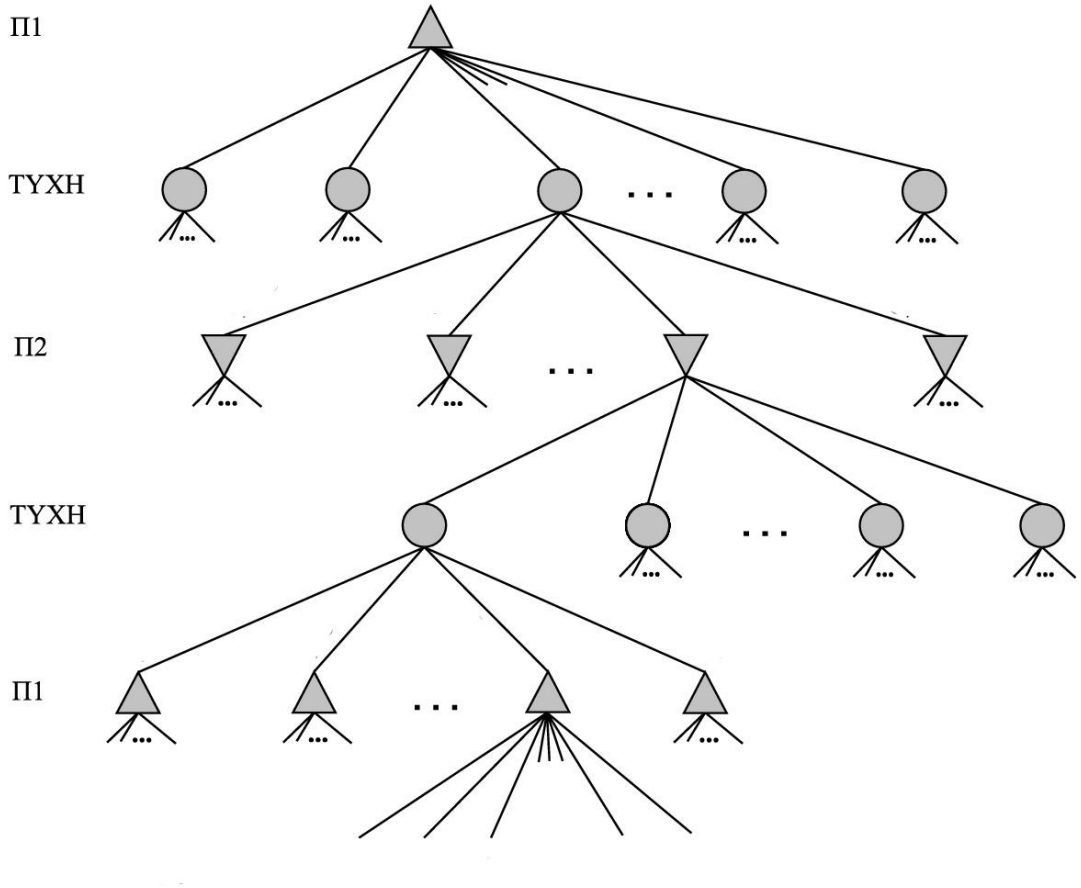
Σχήμα 2.2 Αναπαράσταση ενός δέντρου παιχνιδιού.

2.2.3 Παιχνίδια που εμπεριέχουν στοιχείο τύχης

Σε πολλά παιχνίδια ανάμεσα στην απλή εναλλαγή κινήσεων μεταξύ των αντιπάλων του παιχνιδιού, έρχεται να συμπεριληφθεί και κάποιο απρόβλεπτο συμβάν. Στα περισσότερα από αυτά τα παιχνίδια, το απρόβλεπτο συμβάν αντανακλάται με ένα

στοιχείο τύχης, όπως η ρίψη ενός η περισσότερων ζαριών, το οποίο καθορίζει με ουσιαστικό τρόπο την εξέλιξη του παιχνιδιού.

Παρατηρώντας το Σχήμα 2.3, στο οποίο απεικονίζεται ένα απλό δέντρο παιχνιδιού το οποίο εμπεριέχει στοιχείο τύχης, βλέπουμε ότι ανάμεσα στις στρώσεις των δύο παικτών υπάρχει και μία στρώση η οποία αντιπροσωπεύει το στοιχείο αυτό της τύχης. Διαφοροποιώντας λοιπόν το δέντρο το οποίο είδαμε στο Σχήμα 2.2, μετά την αρχική κατάσταση, όπου έχει σειρά ο παίκτης Π1, αναπτύσσονται οι κόμβοι από όλα τα δυνατά ενδεχόμενα τα οποία μπορούν να προκύψουν από το στοιχείο της τύχης. Σαν διάδοχοι αυτών και έχοντας πάντα σαν γνώμονα τις νόμιμες κινήσεις του κάθε κόμβου, δημιουργούνται οι κόμβοι για τον παίκτη Π2.



Σχήμα 2.3 Δέντρο παιχνιδιού που εμπεριέχει κόμβους τύχης.

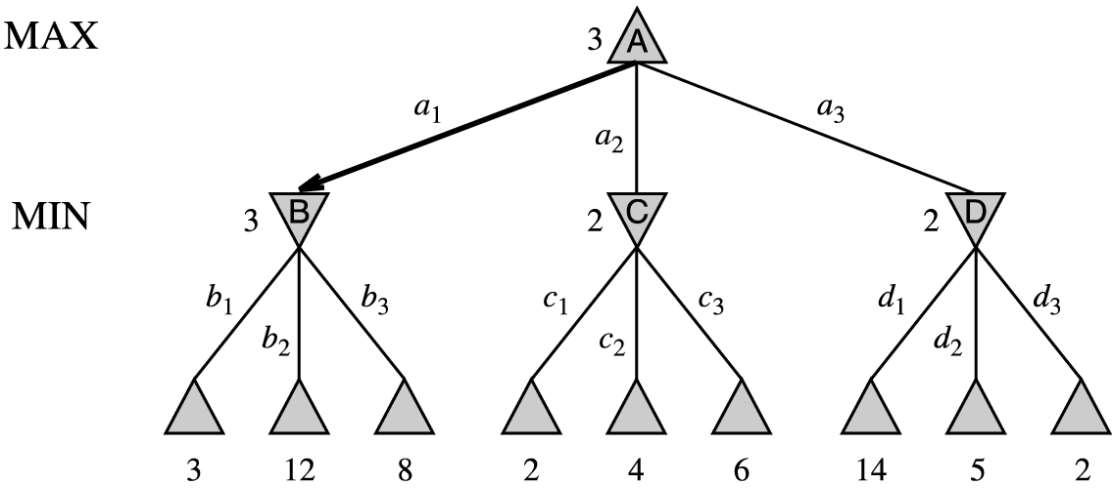
2.3 Αναζήτηση με αντιπαλότητα και MiniMax

2.3.1 Ο αλγόριθμος MiniMax

Σε ένα ανταγωνιστικό περιβάλλον, όπως είναι τα παιχνίδια, οι κινήσεις των δύο παικτών εναλλάσσονται μεταξύ τους έως ότου φτάσουμε σε έναν κόμβο φύλλο. Οι κόμβοι φύλλα ουσιαστικά σηματοδοτούν το τέλος του παιχνιδιού και καθορίζουν, με τη βοήθεια της συνάρτησης χρησιμότητας, το ποιός είναι ο νικητής και ο ηττημένος, ή το αν έχει επέλθει ισοπαλία μεταξύ τους.

Ο κάθε παίκτης έχοντας σαν στόχο την νίκη ή την μεγιστοποίηση των κερδών του, πρέπει σε κάθε του κίνηση να λαμβάνει υπ' όψη του ότι ο αντίπαλος παίκτης έχει ακριβώς τα αντίθετα συμφέροντα από αυτόν. Σε κάθε κατάσταση λοιπόν του παιχνιδιού θα πρέπει να κάνει την κατάλληλη επιλογή κινήσεων, από αυτές που έχει στη διάθεσή του, έτσι ώστε να οδηγήσει την ροή του παιχνιδιού, μέσα από σωστά μονοπάτια του δέντρου, σε κάποιο τερματικό κόμβο ο οποίος θα ικανοποιεί τον στόχο αυτό.

Το πρόβλημα λοιπόν που ανακύπτει, μέσα σε αυτό το περιβάλλον συγκρουόμενων συμφερόντων, είναι το πώς μπορεί ο κάθε παίκτης κάθε φορά να κρίνει, δεδομένης μιας κατάστασης, ποια είναι αυτή η κίνηση η οποία του δίνει μεγαλύτερες ελπίδες να κερδίσει τον αντίπαλό του. Στην επίλυση τέτοιων προβλημάτων χρησιμοποιούνται τεχνικές που βασίζονται πάνω στον αλγόριθμο MiniMax [1]. Ο αλγόριθμος αυτός στηρίζεται στη στρατηγική της αναζήτησης πρώτα σε βάθος και ουσιαστικά προσομοιώνει την επιθυμία του κάθε παίκτη, μέσα από τη "σύγκρουση" αυτή, να μεγιστοποιήσει το κέρδος του, αλλά και την επιθυμία του αντιπάλου του να την ελαχιστοποιήσει. Για τον λόγο αυτό στο δέντρο παιχνιδιού που απεικονίζεται στο Σχήμα 2.4, οι παίκτες αναγράφονται σαν max και min αντίστοιχα.



Σχήμα 2.4 Αναπαράσταση ενός δέντρου παιχνιδιού με τους παίκτες max και min.

Παρατηρώντας το Σχήμα 2.4 βλέπουμε ότι το δέντρο αποτελείται από τρεις στρώσεις, μία για την αρχική κατάσταση, όπου παίζει ο παίκτης max, μία για τον παίκτη min καθώς και η τελευταία στρώση όπου βρίσκονται και οι τερματικοί κόμβοι του παιχνιδιού. Κάτω από κάθε τερματικό κόμβο αναγράφεται και η τιμή η οποία προσδίδεται σε αυτόν από τη συνάρτηση χρησιμότητας.

Ξεκινώντας από την αρχική κατάσταση-κόμβο A, ο παίκτης max έχει τρεις διάδοχους κόμβους B,C και D οι οποίοι θα τον διαδεχθούν εάν ενεργήσει σύμφωνα με τις νόμιμες κινήσεις a1, a2 και a3 αντίστοιχα. Η τιμή η οποία θα προσδώσει ο αλγόριθμος MiniMax στον κόμβο A, θα είναι η μεγαλύτερη μεταξύ των τιμών των κόμβων-διαδόχων του. Για την εύρεση λοιπόν των τιμών αυτών στους κόμβους B,C και D πρέπει να λειτουργήσουμε ανάλογα. Στη στρώση όμως αυτή, σειρά έχει να παίξει ο παίκτης min, ο οποίος επιζητά την ελαχιστοποίηση των κερδών του αντιπάλου του max. Έτσι λοιπόν σε κάθε κόμβο της στρώσης του παίκτη min, ο αλγόριθμος MiniMax θα προσδώσει την μικρότερη τιμή μεταξύ των κόμβων-διαδόχων του καθενός εξ' αυτών, οι οποίοι είναι τερματικοί κόμβοι στο δέντρο του Σχήματος 2.4 και η τιμή τους μπορεί να οριστεί άμεσα από την συνάρτηση χρησιμότητας.

Λειτουργώντας λοιπόν με την λογική της επιλογής των μέγιστων δυνατών τιμών από μέρους των κόμβων του παίκτη max, της επιλογής των ελάχιστων δυνατών τιμών από μέρους των κόμβων του παίκτη min και ανασύροντας τις τιμές από τους κόμβους-φύλλα του δέντρου προς την ρίζα αυτού, στα πλαίσια μιας αναζήτησης πρώτα σε

βάθος με υπαναχώρηση, μπορούμε να προσδιορίσουμε την τιμή MiniMax για τον αρχικό-κόμβο του δέντρου του παιχνιδιού. Κατά συνέπεια, στους κόμβους B, C, D ορίζονται οι τιμές 3, 2 και 2 αντίστοιχα, καθώς και στην αρχική κατάσταση-κόμβο A η τιμή 3.

Παρατηρώντας ξανά την ακολουθία των "συλλογισμών" του αλγόριθμου MiniMax, βλέπουμε ότι θεωρήσαμε δεδομένη την βέλτιστη απόκριση, όσον αφορά το συμφέρον του παίκτη min. Αυτό όμως είναι και το σημείο το οποίο μας εξασφαλίζει ότι οποιεσδήποτε και αν είναι οι κινήσεις του παίκτη min, ο παίκτης max θα έχει κέρδος τουλάχιστον αυτό το οποίο επιστρέφεται σαν τιμή εφαρμόζοντας τον αλγόριθμο MiniMax.

Αναλυτικότερα ο αλγόριθμος MiniMax χωρίζεται σε δύο επιμέρους συναρτήσεις. Η μία εκτελείται στους κόμβους max του δέντρου του παιχνιδιού και η άλλη στους κόμβους min. Μία απεικόνιση του αλγόριθμου σε ψευδοκώδικα φαίνεται στο Σχήμα 2.5. Η συνάρτηση Succesors αντιπροσωπεύει την συνάρτηση διαδόχων και η συνάρτηση Utility αντιπροσωπεύει την συνάρτηση χρησιμότητας.

```
Function MiniMaxDecision (κατάσταση)
inputs: Κατάσταση, τρέχουσα κατάσταση παιχνιδιού
returns: Μια ενέργεια
  u ← MaxValue (κατάσταση)
  return Ενέργεια ∈ Successors (κατάσταση) που έχει τιμή u

Function MaxValue (κατάσταση)
inputs: Μια κατάσταση
returns: Μια τιμή χρησιμότητας
  if TerminalTest (κατάσταση) then
    return Utility (κατάσταση)
  end if
  u ← −∞
  for all s ∈ Successors (κατάσταση) do
    u ← max(u, MinValue (κατάσταση))
  end for
  return u

Function MinValue (κατάσταση)
inputs: Μια κατάσταση
returns: Μια τιμή χρησιμότητας
  if TerminalTest (κατάσταση) then
    return Utility (κατάσταση)
  end if
  u ← +∞
  for all s ∈ Successors (κατάσταση) do
    u ← min(u, MaxValue (κατάσταση))
  end for
  return u
```

Σχήμα 2.5 Ο αλγόριθμος MiniMax σε ψευδοκώδικα.

2.3.2 Κλάδεμα Άλφα-Βήτα

Ο MiniMax είναι αλγόριθμος ο οποίος μας δίνει την απάντηση της επιλογής της βέλτιστης κίνησης σε μία κατάσταση ενός δέντρου αναζήτησης με αντιπαλότητα. Ένα όμως από τα μεγαλύτερα μειονεκτήματά του είναι ότι στηρίζεται στην στρατηγική των αλγόριθμων πρώτα σε βάθος, η οποία είναι μία εξαντλητική αναζήτηση του δέντρου του παιχνιδιού ανεξαρτήτως του βάθους του. Αυτή η παρατήρηση καθιστά απαγορευτική την εφαρμογή του σε δέντρα τα οποία έχουν πολύ μεγάλο βάθος, όπως είναι τα περισσότερα δέντρα των παιχνιδιών.

Μία καλύτερη εξέταση όμως του MiniMax θα μας οδηγήσει στην παρατήρηση ότι θα μπορούσαμε να εισπράξουμε την ίδια απάντηση, όσον αφορά την βέλτιστη κίνηση, ακόμα και αν δεν εξετάσουμε ένα μέρος του δέντρου του παιχνιδιού. Θα μπορούσαμε δηλαδή να παραβλέψουμε κάποια υποδέντρα του συνολικού δέντρου του παιχνιδιού και να αποφύγουμε την άσκοπη αναζήτηση σε αυτά, γνωρίζοντας εκ των προτέρων ότι δεν μπορούν να επηρεάσουν το αποτέλεσμα της αναζήτησής μας.

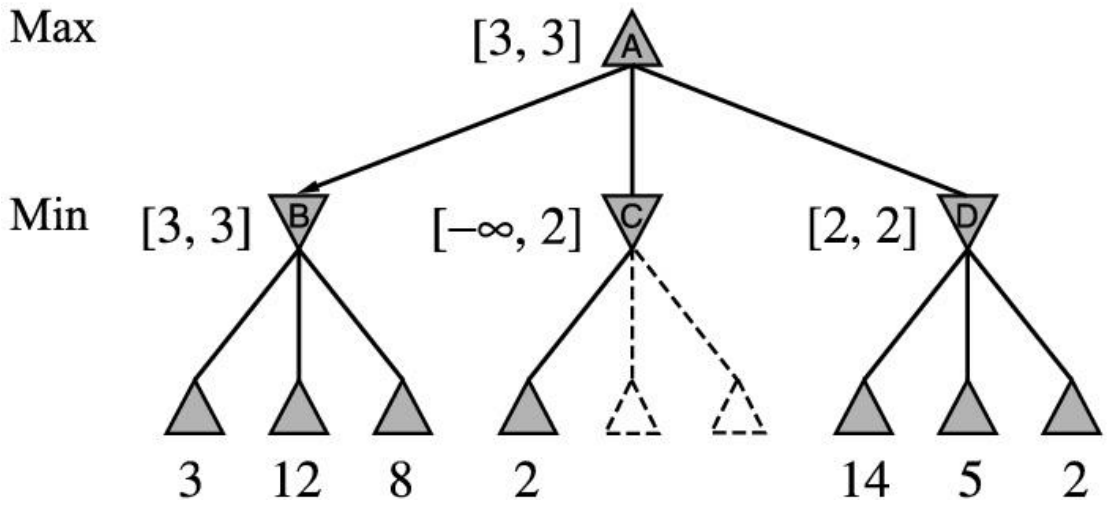
Η τεχνική η οποία υλοποιεί την παραπάνω παρατήρηση και ενσωματώνεται στον αλγόριθμο MiniMax ονομάζεται κλάδεμα Άλφα-Βήτα (alpha-beta pruning) [1]. Η γενική αρχή του κλαδέματος Άλφα-Βήτα στηρίζεται στην παραδοχή ότι αν ένας κόμβος n βρίσκεται κάπου στο δέντρο του παιχνιδιού και ο παίκτης έχει εξασφαλίσει μία καλύτερη τιμή, σε ένα κόμβο m, ο οποίος βρίσκεται σε οποιοδήποτε σημείο πάνω από τον n, τότε ο παίκτης δεν θα φτάσει ποτέ στον κόμβο n στο πραγματικό παιχνίδι. Αρχίζοντας λοιπόν την εξέταση σε ορισμένους από τους απογόνους του κόμβου n μπορούμε να καθορίσουμε το μέγιστο κέρδος που μπορούμε να αποκομίσουμε από αυτούς, και όταν αυτό βρεθεί κάτω από την τιμή την οποία έχουμε ήδη εξασφαλίσει σαν ελάχιστη τιμή κέρδους, τότε μπορούμε να κλαδέψουμε τον κόμβο n και έτσι να αποφύγουμε την περαιτέρω άσκοπη αναζήτησή μας στους υπόλοιπους απογόνους του. Το μεγάλο πλεονέκτημα του κλαδέματος Άλφα-Βήτα είναι ότι με την συγκεκριμένη τεχνική μπορούμε να κλαδέψουμε ολόκληρα υποδέντρα, τα οποία μπορεί να αποτελούν αρκετά μεγάλο μέρος του συνολικού δέντρου αναζήτησης. Είναι προφανές ότι όσο μεγαλύτερος είναι ο "όγκος" των υποδέντρων που κλαδεύονται, τόσο μεγαλύτερα θα είναι και τα κέρδη σε χρόνο της αναζήτησής μας.

Για την παρακολούθηση των ορίων, τα οποία αντικατοπτρίζουν το ελάχιστο εξασφαλισμένο κέρδος του κάθε παίκτη, όσο προχωρά η αναζήτηση για την βέλτιστη επιλογή κίνησης μέσα στο δέντρο του παιχνιδιού, η τεχνική του κλαδέματος Άλφα-Βήτα χρησιμοποιεί δύο παραμέτρους, a και b, από όπου πήρε και το όνομά της. Οι παράμετροι αυτοί ορίζονται ως εξής :

a = η τιμή της καλύτερης, μέχρι στιγμής, επιλογής για τον παίκτη max σε οποιοδήποτε σημείο της διαδρομής της αναζήτησης. Δηλαδή αυτής με την μεγαλύτερη τιμή.

b = η τιμή της καλύτερης, μέχρι στιγμής, επιλογής για τον παίκτη min σε οποιοδήποτε σημείο της διαδρομής της αναζήτησης. Δηλαδή αυτής με την μικρότερη τιμή.

Οι τιμές των a, b διατηρούνται σε τοπικές μεταβλητές, οι οποίες "περνούν" ως ορίσματα από τους κόμβους-γονείς προς τα παιδιά τους και ενημερώνονται τοπικά σε κάθε κόμβο. Ένα παράδειγμα της εφαρμογής του κλαδέματος Άλφα-Βήτα απεικονίζεται στο Σχήμα 2.6. Ο αλγόριθμος MiniMax εξετάζει σαν πρώτο διάδοχο του κόμβου Α, τον κόμβο Β. Για τον κόμβο αυτόν θα επιλεγεί και θα του καταχωρηθεί η ελάχιστη τιμή από αυτές που έχουν οι διάδοχοί του, σαν κόμβος του παίκτη min, δηλαδή η τιμή 3, όπως φαίνεται στο παράδειγμά μας. Μέχρι εδώ μπορούμε εύκολα να συμπεράνουμε ότι ο κόμβος Α, οποιαδήποτε και αν είναι η εξέλιξη της αναζήτησης, θα έχει σαν κάτω όριο στο προσδοκώμενο κέρδος του την τιμή 3, αφού σαν κόμβος του max επιλέγει την μεγαλύτερη τιμή μεταξύ αυτών που έχουν καθοριστεί για τους διαδόχους του Β, C και D. Στην συνέχεια εξετάζεται ο κόμβος C. Ο πρώτος διάδοχος του κόμβου αυτού επιστρέφει την τιμή 2 και γνωρίζοντας ότι ο κόμβος C είναι κόμβος του παίκτη min, συμπεραίνουμε ότι οποιεσδήποτε και αν είναι οι τιμές από τους υπόλοιπους διαδόχους του, το κέρδος γι' αυτόν δεν θα ξεπεράσει ποτέ την τιμή αυτή. Έτσι δεν θα επιλεγεί ποτέ σαν βέλτιστη κίνηση από τον μητρικό του κόμβο Α, επειδή το 2 είναι αυστηρά μικρότερο από την τιμή 3 που έχουμε ήδη εξασφαλίσει σαν ελάχιστο κέρδος του κόμβου Α, επομένως η εξέταση των υπόλοιπων διαδόχων του κόμβου C θεωρείται άσκοπη και δεν πραγματοποιείται ποτέ.



Σχήμα 2.6 Παράδειγμα της τεχνικής κλαδέματος Άλφα-Βήτα.

Η διαμόρφωση του αλγόριθμου MiniMax, μετά την προσθήκη της τεχνικής του κλαδέματος Άλφα-Βήτα, φαίνεται στο Σχήμα 2.7.

Function AlphaBetaSearch (κατάσταση)
inputs: Κατάσταση, τρέχουσα κατάσταση παιχνιδιού
returns: Μια ενέργεια

$u \leftarrow$ MaxValue (κατάσταση, $-\infty$, $+\infty$)
return Ενέργεια $\in$ Successors (κατάσταση) που έχει τιμή u

Function MaxValue (κατάσταση, α, β)
inputs: Μια κατάσταση
α, τιμή καλύτερης εναλλακτικής επιλογής του Max πάνω στην διαδρομή προς την κατάσταση
β,τιμή καλύτερης εναλλακτικής επιλογής του Min πάνω στην διαδρομή προς την κατάσταση
returns: Μια τιμή χρησιμότητας

if TerminalTest (κατάσταση) **then**
return Utility (κατάσταση)
end if
$u \leftarrow -\infty$
for all $s \in$ Successors (κατάσταση) **do**
$u \leftarrow \max(u,$ MinValue (κατάσταση, α, $\beta))$
if $u \geq \beta$ **then**
return u
end if
$\alpha \leftarrow \max(\alpha, u)$
end for
return u

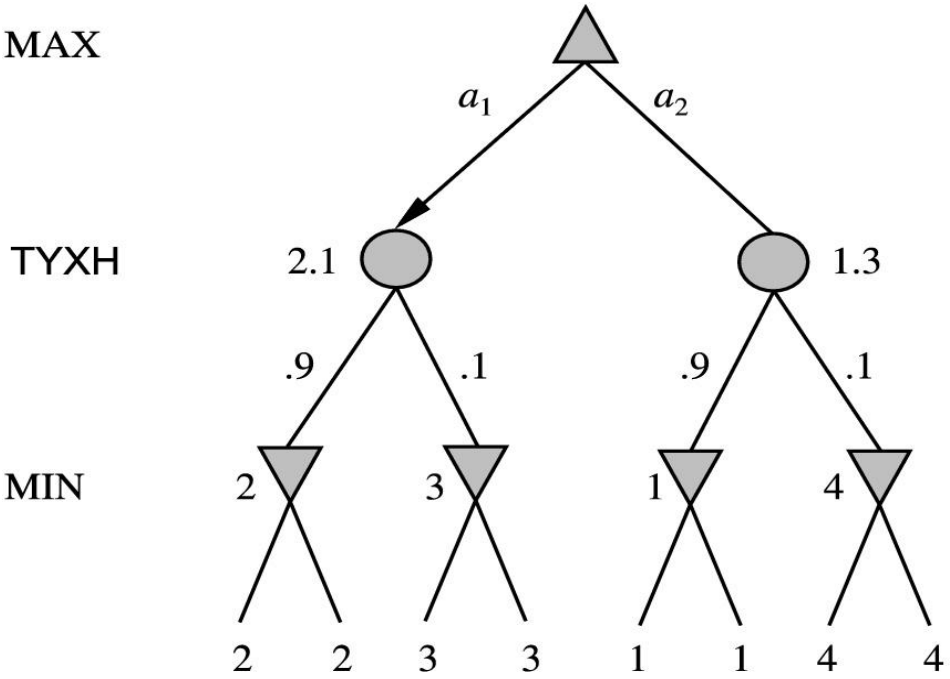
Function MinValue (κατάσταση, α, β)
inputs: Μια κατάσταση
α, τιμή καλύτερης εναλλακτικής επιλογής του Max πάνω στην διαδρομή προς την κατάσταση
β,τιμή καλύτερης εναλλακτικής επιλογής του Min πάνω στην διαδρομή προς την κατάσταση
returns: Μια τιμή χρησιμότητας

```
if TerminalTest (κατάσταση) then
  return Utility (κατάσταση)
end if
u ← +∞
for all s ∈ Successors (κατάσταση) do
  u ← min(u, MaxValue (κατάσταση, α, β))
  if u ≤ α then
    return u
  end if
  β ← min(β, u)
end for
return u
```

Σχήμα 2.7 Ο αλγόριθμος MiniMax με κλάδεμα Άλφα-Βήτα σε ψευδοκώδικα.

2.3.3 Ο αλγόριθμος ExpectiMiniMax

Επανερχόμαστε στο πρόβλημα της αναζήτησης της βέλτιστης κίνησης σε δέντρα παιχνιδιών τα οποία εμπεριέχουν το στοιχείο της τύχης. Όπως είδαμε στα δέντρα αυτά, ανάμεσα στις στρώσεις του παίκτη max και στις στρώσεις του παίκτη min παρεμβάλλεται μία ακόμα στρώση με κόμβους τύχης, οι οποίοι αντιπροσωπεύουν το απρόβλεπτο του παιχνιδιού αυτού. Όπως φαίνεται και στο Σχήμα 2.8 σε κάθε έναν από τους διαδόχους αυτών των κόμβων αναγράφεται και ένας προκαθορισμένος αριθμός. Ο αριθμός αυτός είναι η πιθανότητα να συμβεί το ενδεχόμενο το οποίο αντιπροσωπεύει ο συγκεκριμένος διάδοχος κόμβος.



Σχήμα 2.8 Υπολογισμός τιμών σε δέντρο με κόμβους τύχης με τον ExpectiMiniMax.

Σε ένα απλό δέντρο παιχνιδιού το οποίο δεν περιέχει κόμβους τύχης, η τιμή MiniMax μπορεί να καθοριστεί ακριβώς για κάθε κόμβο του παιχνιδιού, είτε είναι κόμβος min είτε είναι κόμβος max. Προφανώς σε ένα παιχνίδι το οποίο εμπεριέχει και το στοιχείο της τύχης, ο σκοπός παραμένει ο ίδιος, δηλαδή η εύρεση της βέλτιστης κίνησης. Οι νέες θέσεις όμως οι οποίες προκύπτουν σε μία τέτοια περίπτωση δεν έχουν προκαθορισμένες τιμές MiniMax. Για τους κόμβους τύχης λοιπόν μπορούμε μόνο να υπολογίσουμε την αναμενόμενη τιμή (expected value), η οποία προκύπτει λαμβάνοντας υπόψη όλα τα δυνατά ενδεχόμενα τα οποία θα μπορούσαν να συμβούν.

Η ιδιαιτερότητα αυτή των παιχνιδιών που περιέχουν κόμβους τύχης μας οδηγεί στη γενίκευση της τιμής MiniMax, που αναλύσαμε μέχρι τώρα, σε μια τιμή ExpectiMiniMax [1]. Για τους κόμβους max και min καθώς και για τους τερματικούς κόμβους, για τους οποίους καθορίζεται απευθείας κάποια τιμή από την συνάρτηση χρησιμότητας, η αντιμετώπιση είναι ακριβώς η ίδια όπως και στον αλγόριθμο MiniMax. Η διαφοροποίηση γίνεται στους κόμβους τύχης οι οποίοι αξιολογούνται, από τον αλγόριθμο ExpectiMiniMax, με τον υπολογισμό του σταθμισμένου μέσου όρου των τιμών που προκύπτουν από όλα τα δυνατά ενδεχόμενα. Στο Σχήμα 2.9 φαίνεται η συνάρτηση με την οποία γίνεται ο υπολογισμός της τιμής ExpectiMiniMax για τους κόμβους τύχης, όπου n είναι κάποιος κόμβος τύχης και m είναι ο αριθμός

από τα ενδεχόμενα τα οποία θα μπορούσαν να συμβούν. Σαν F ορίζεται η συνάρτηση η οποία επιστρέφει την τιμή MiniMax από τον αντίστοιχο διάδοχο ενός συγκεκριμένου ενδεχόμενου, ενώ P είναι η τιμή της πιθανότητας αυτό το ενδεχόμενο να συμβεί.

$$Expectimimimax(n) = F(1)*P(1) + F(2)*P(2) +\ldots+ F(m)*P(m)$$

Σχήμα 2.9 Εξίσωση υπολογισμού της τιμής ExpectiMiniMax για κόμβους τύχης.

Για την καλύτερη κατανόηση της λειτουργίας του ExpectiMiniMax ας δούμε το παράδειγμα του Σχήματος 2.8. Μετά τις νόμιμες κινήσεις, a1 και a2, του παίκτη max η αναζήτηση φτάνει σε δύο κόμβους τύχης, έναν για κάθε μία από τις κινήσεις αυτές. Σύμφωνα με τον απρόβλεπτο παράγοντα του συγκεκριμένου παραδείγματος, σε κάθε έναν από τους κόμβους αυτούς αποδίδονται δύο ενδεχόμενα, με αντίστοιχες πιθανότητες να συμβούν 0,9 και 0,1 αντίστοιχα. Αρχίζοντας από τον πρώτο κόμβο τύχης βλέπουμε ότι ακολουθούν δύο κόμβοι του παίκτη min στους οποίους, ακολουθώντας την ανάλυσή μας για τον MiniMax, τους έχουν αποδοθεί τιμές 2 και 3 αντίστοιχα. Σύμφωνα λοιπόν με την εξίσωση στο Σχήμα 2.9, για τον υπολογισμό του σταθμισμένου μέσου όρου, η τιμή ExpectiMiniMax που αποδίδεται στον πρώτο κόμβο τύχης είναι : 2x0,9 + 3x0,1 = 2,1. Ακολουθώντας την ίδια λογική και για το υπόλοιπο του δέντρου μπορούμε να αντιγράψουμε τις τιμές με αναδρομή μέχρι την ρίζα του δέντρου, όπως ακριβώς γινόταν και με τον MiniMax.

Η επιπλέον αυτή διαδικασία του υπολογισμού των τιμών για τους κόμβους τύχης ενός δέντρου παιχνιδιού, έρχεται να προσθέσει και μία τρίτη υποσυνάρτηση στις ήδη υπάρχουσες του αλγόριθμου MiniMax η οποία αφορά ακριβώς αυτούς τους κόμβους. Μία σύντομη απεικόνιση του αλγόριθμου ExpectiMiniMax απεικονίζεται στο Σχήμα 2.10 όπου n είναι ένας κόμβος τύχης. Η Utility είναι η συνάρτηση χρησιμότητας και η Successors είναι η συνάρτηση διαδόχων, η οποία στην περίπτωση των κόμβων τύχης απλά αναπαράγει την κατάσταση του n με κάθε δυνατό ενδεχόμενο του απρόβλεπτου παράγοντα του παιχνιδιού, ενώ P είναι η πιθανότητα να συμβεί καθένα από αυτά.

$$\text{Expectiminimax}(n) = \begin{cases} \text{Utility}(n) & \text{αν n είναι τερματική κατάσταση} \\ \text{Max}_{s \in \text{Successors}(n)}\ \text{Expectiminimax}(s) & \text{αν n είναι κόμβος max} \\ \text{Min}_{s \in \text{Successors}(n)}\ \text{Expectiminimax}(s) & \text{αν n είναι κόμβος min} \\ \sum_{s \in \text{Successors}(n)} P(s)*\text{Expectiminimax}(s) & \text{αν n είναι κόμβος τύχης} \end{cases}$$

Σχήμα 2.10 Ο αλγόριθμος ExpectiMiniMax.

2.4 Συνάρτηση αξιολόγησης

2.4.1 Περιορισμός χρόνου και έλεγχος αποκοπής

Έως εδώ αναλύσαμε την λειτουργία του αλγόριθμου MiniMax, ο οποίος μας επιτρέπει σε ένα ανταγωνιστικό περιβάλλον να επιλέξουμε την καλύτερη κίνηση για τον παίκτη μας. Παράλληλα με την βοήθεια της τεχνικής του κλαδέματος Άλφα-Βήτα καταφέραμε να αποφύγουμε την άσκοπη αναζήτηση σε ένα μεγάλο μέρος του δέντρου, κερδίζοντας έτσι χρόνο και βελτιώνοντας την απόδοση του πράκτορά μας.

Παρ' όλες τις βελτιώσεις όμως που μπορεί να επιδεχθεί ο αλγόριθμος MiniMax, για να φέρει το σωστό αποτέλεσμα θα χρειαστεί να ψάξει τουλάχιστον ένα μέρος από τους τερματικούς κόμβους του δέντρου του παιχνιδιού, οι οποίοι περιέχουν και την πραγματική "αλήθεια" για την εξέλιξη του και να ανασύρει τις τιμές αυτές, με τον τρόπο που περιγράψαμε παραπάνω, έως την ρίζα του. Ερχόμενοι όμως να εφαρμόσουμε τις τεχνικές αυτές αναζήτησης σε πραγματικά δέντρα παιχνιδιών, βρισκόμαστε συνήθως αντιμέτωποι με τον περιορισμό του χρόνου. Τα περισσότερα παιχνίδια απαιτούν να γίνονται οι κινήσεις μέσα σε ένα λογικό χρονικό διάστημα, ενώ από την άλλη το βάθος του δέντρου τους είναι πολύ μεγάλο, με αποτέλεσμα να είναι απαγορευτική η εφαρμογή του αλγόριθμου MiniMax.

Για να αντιμετωπίσουμε το παραπάνω πρόβλημα, εφαρμόζουμε την τεχνική αναζήτησης περιορισμένου βάθους. Ουσιαστικά δηλαδή σταματάμε την αναζήτηση του αλγόριθμου MiniMax σε ένα βάθος τέτοιο, το οποίο του επιτρέπει να φέρει κάποιο αποτέλεσμα μέσα σε ένα χρονικό διάστημα "αποδεκτό" από τους κανόνες του

παιχνιδιού. Ο έλεγχος τερματισμού (terminal test) λοιπόν, που θέσαμε σαν συνιστώσα για ένα παιχνίδι, αντικαθίσταται από τον έλεγχο αποκοπής (cutoff test), ο οποίος είναι υπεύθυνος για τον περιορισμό της αναζήτησης σε αυτό το προκαθορισμένο βάθος.

2.4.2 Συνάρτηση αξιολόγησης

Στις περισσότερες όμως περιπτώσεις στο περιορισμένο αυτό σε βάθος δέντρο, δεν υπάρχουν τερματικοί κόμβοι για να μπορεί ο MiniMax να ανασύρει τις τιμές τους. Για τον λόγο αυτό θεωρούμε τους μη τερματικούς κόμβους από την στρώση του μέγιστου επιτρεπτού βάθους της αναζήτησης σαν τερματικά φύλλα. Φτάνοντας στα τερματικά αυτά φύλλα, εφαρμόζουμε μια ευρετική συνάρτηση αξιολόγησης (evaluation function), η οποία ουσιαστικά αντικαθιστά την συνάρτηση χρησιμότητας (utility function) που εφαρμόζαμε μέχρι τώρα.

Η συνάρτηση αξιολόγησης, εφαρμοζόμενη σε μια δεδομένη θέση του δέντρου, επιστρέφει μια εκτίμηση της αναμενόμενης χρησιμότητας του παιχνιδιού, εάν θεωρούσαμε την θέση αυτή τερματικό κόμβο του. Δεδομένης της αβεβαιότητας για τη περαιτέρω εξέλιξη του, σε μία θέση του παιχνιδιού η οποία μπορεί να απέχει πολύ από τους πραγματικούς τερματικούς κόμβους του δέντρου αναζήτησης, η συνάρτηση αξιολόγησης ουσιαστικά προσπαθεί να μαντέψει το τελικό αποτέλεσμα.

Ένα από τα βασικά κριτήρια για να χαρακτηριστεί μια συνάρτηση αξιολόγησης καλή, είναι ο χρόνος με τον οποίο επιβαρύνει την συνολική διαδικασία της αναζήτησης. Η επιβάρυνση αυτή δεν θα πρέπει να αναιρεί τον λόγο της αναζήτησης περιορισμένου βάθους, δηλαδή την εξοικονόμηση χρόνου. Επίσης η διάταξη, όσον αφορά την τιμή της εκτιμώμενης χρησιμότητάς τους, οποιουδήποτε συνόλου τερματικών κόμβων ενός δέντρου, εφαρμόζοντας σε αυτούς την συνάρτηση αξιολόγησης, θα πρέπει να είναι ίδια με την διάταξη που θα είχαν αν εφαρμοζόταν σε αυτούς η πραγματική συνάρτηση χρησιμότητας. Σε διαφορετική περίπτωση ο πράκτοράς μας, φτάνοντας την αναζήτησή του σε θέσεις του δέντρου με τερματικούς κόμβους, θα επέλεγε σαν καλύτερη κίνηση κάποια η οποία δεν θα ανταποκρινόταν στην τιμή του πραγματικού κέρδους. Ένα ακόμα κριτήριο για την αναμενόμενη απόδοση της συνάρτησης αξιολόγησης είναι η συσχέτιση των μη τερματικών κόμβων, όσο γίνεται πιο στενά με τις πραγματικές πιθανότητες νίκης. Όσο πιο σωστά “στηθεί” λοιπόν μια συνάρτηση αξιολόγησης τόσο μεγαλύτερες πιθανότητες έχει ο παίκτης, ακολουθώντας τις

εκτιμώμενες τιμές των θέσεων, να οδηγηθεί σε πραγματικούς τερματικούς κόμβους που θα του αποφέρουν το επιθυμητό κέρδος.

Για την υλοποίηση μιας τέτοιας συνάρτησης αξιολόγησης χρησιμοποιούνται συνήθως συναρτήσεις, οι οποίες αντανακλούν κάποια χαρακτηριστικά (features) της κάθε κατάστασης του παιχνιδιού. Ουσιαστικά τα χαρακτηριστικά αυτά είναι κάποιες μετρικές, οι οποίες ορίζονται αυθαίρετα, ανάλογα με τις ιδιαιτερότητες του κάθε παιχνιδιού. Ο ορισμός τους θα πρέπει να γίνεται με τρόπο τέτοιο ώστε να μπορούν να αποδώσουν όσον το δυνατόν πιο ρεαλιστικά την κατάσταση ενός παιχνιδιού. Η συνάρτηση αξιολόγησης μπορεί να είναι μια οποιαδήποτε αυθαίρετα ορισμένη συνάρτηση, είτε με γραμμικό είτε με μη γραμμικό χαρακτήρα. Η πιο συνηθισμένη μορφή της όμως είναι μια σταθμισμένη γραμμική συνάρτηση των χαρακτηριστικών αυτών όπως φαίνεται στο Σχήμα 2.11.

$$Eval(s) = w_1 f_1(s) + w_2 f_2(s) + ... + w_n f_n(s) = \sum_{i=1}^{n} w_i f_i(s)$$

Σχήμα 2.11 Η εξίσωση της συνάρτησης αξιολόγησης.

Τα $f_i(s)$ είναι οι συναρτήσεις για τα χαρακτηριστικά μιας κατάστασης s, ενώ τα w_i είναι κάποια βάρη με τα οποία πολλαπλασιάζονται οι επιστρεφόμενες τιμές των συναρτήσεων αυτών για να τους προσδώσουν την σχετική σημαντικότητά τους στο συνολικό άθροισμα. Η συνάρτηση αξιολόγησης και κατά συνέπεια τα χαρακτηριστικά της f, είναι μία από τις σημαντικότερες παραμέτρους για την απόδοση ενός πράκτορα που υλοποιείται, με σκοπό την εφαρμογή τέτοιων τεχνικών αναζήτησης. Γι' αυτό το λόγο θα πρέπει να δίνεται ιδιαίτερο βάρος για την εύρεση καλών χαρακτηριστικών.

2.5 Ενισχυτική μάθηση

2.5.1 Μάθηση και συνάρτηση αξιολόγησης

Στην μέχρι τώρα ανάλυσή μας, ορίσαμε την λειτουργία της συνάρτησης αξιολόγησης, καθώς επίσης επισημάναμε και την σημαντικότητα της όσον αφορά την καλή απόδοση του πράκτορά μας. Είναι αυτή που καθορίζει σε μεγάλο βαθμό, το αν η

συμπεριφορά του μέσα σε ένα ανταγωνιστικό περιβάλλον παιχνιδιού χαρακτηρίζεται επιτυχής ή όχι. Ένα από τα μεγαλύτερα προβλήματα όμως το οποίο καλούμαστε να λύσουμε είναι το πώς θα φτιάξουμε μια τέτοια καλή συνάρτηση αξιολόγησης.

Βλέποντας πάλι την εξίσωση του Σχήματος 2.11 παρατηρούμε ότι ουσιαστικά μια τέτοια συνάρτηση μπορεί να χωριστεί σε δύο μεγάλες ομάδες συστατικών. Η μία είναι αυτή των χαρακτηριστικών (features) και η άλλη είναι αυτή που αποτελείται από τα βάρη αυτών. Στο πρόβλημα της εύρεσης καλών χαρακτηριστικών τα μοναδικά εργαλεία που έχουμε στην διάθεσή μας είναι η εμπειρία μας πάνω στο παιχνίδι και η σύγκριση μεταξύ κάποιων ομάδων από τέτοια χαρακτηριστικά, ώστε να καταλήξουμε στα πιο αποτελεσματικά. Ουσιαστικά λοιπόν εστιάζουμε το πρόβλημά μας στην εύρεση καλών ή και βέλτιστων βαρών, τα οποία καθορίζουν την αξία του κάθε χαρακτηριστικού στην συνολική επιστρεφόμενη τιμή από την συνάρτηση αξιολόγησης.

Θα ήταν μάταιο αν προσπαθούσαμε εξ' αρχής να βρούμε ποιες είναι αυτές οι τιμές για τα βάρη των χαρακτηριστικών που θεωρούνται οι καλύτερες. Αφού λοιπόν δεν μπορούμε να τις προκαθορίσουμε, στρέφουμε την προσπάθειά μας στο να τις μάθουμε, εκπαιδεύοντας τον πράκτορά μας. Έτσι λοιπόν χρησιμοποιούμε τεχνικές από τον χώρο της ενισχυτικής μάθησης (reinforcement learning) [1][2][3]. Οι τεχνικές αυτές στηρίζονται στην λογική της ανταμοιβής (reward) ή ενίσχυσης (reinforcement) του πράκτορά μας σε κάθε κίνηση που θα επιφέρει την νίκη ή την ήττα γι' αυτόν, με θετική ή αρνητική ανταμοιβή αντίστοιχα. Σαν τελικό στόχο επομένως θέτουμε την μάθηση των τιμών για τα βάρη της εξίσωσης 2.11 έτσι ώστε η συνάρτηση αξιολόγησης να προσεγγίζει όσον το δυνατόν περισσότερο την πραγματική συνάρτηση χρησιμότητας.

2.5.2 Μάθηση χρονικών διαφορών (TD)

Μια μέθοδος η οποία ακολουθεί την λογική της ενισχυτικής μάθησης είναι και η μέθοδος χρονικών διαφορών (temporal difference) [2]. Η μέθοδος αυτή ονομάζεται έτσι γιατί ουσιαστικά εκμεταλλεύεται την κάθε μετάβαση από μία κατάσταση s σε μία άλλη s', μετά από μία κίνηση του παίκτη μας κατά την διάρκεια του παιχνιδιού, με σκοπό την βελτίωσή του.

Η λογική της μεθόδου χρονικών διαφορών στηρίζεται στην γενική παραδοχή ότι η κατάσταση s' εμπεριέχει μεγαλύτερο μέρος "αλήθειας", σε σχέση με τον πρόγονό της s, βρισκόμενη σε μεγαλύτερο βάθος του δέντρου του παιχνιδιού και έτσι πιο κοντά σε τερματικούς κόμβους που είναι και αυτοί που περιέχουν την πραγματική αξία. Έχοντας σαν γνώμονα το παραπάνω, η μέθοδος αυτή προσπαθεί ουσιαστικά να εξαλείψει την διαφορά ανάμεσα στις τιμές της συνάρτησης αξιολόγησης για τις δύο διαδοχικές καταστάσεις, "συμμορφώνοντας", κατά κάποιο τρόπο, την κατάσταση-πρόγονο s σύμφωνα με την κατάσταση-απόγονο s'. Η εξίσωση που υλοποιεί την ενημέρωση των βαρών, της συνάρτησης αξιολόγησης, σύμφωνα με την μέθοδο των χρονικών διαφορών είναι αυτή που απεικονίζεται στο Σχήμα 2.13.

$$w_i \leftarrow w_i + \alpha f_i(s)(r + \quad Eval(s') - Eval(s))$$

Σχήμα 2.13 Η εξίσωση για την ενημέρωση των βαρών της συνάρτησης αξιολόγησης με την μέθοδο των χρονικών διαφορών (temporal difference).

Στην παραπάνω εξίσωση παρατηρούμε ότι η ενημέρωση αυτή εκτελείται για κάθε μεταβλητή βάρους w_i της συνάρτησης αξιολόγησης ξεχωριστά. Η $f_i(s)$ αναπαριστά την τιμή για το αντίστοιχο χαρακτηριστικό του βάρους που ενημερώνεται, ενώ οι Eval(s) και Eval(s') είναι οι τιμές που επιστρέφει η συνάρτηση αξιολόγησης για τις καταστάσεις s και s' αντίστοιχα. Ο συντελεστής α καθορίζει, όπως φαίνεται, τον αντίκτυπο που θα έχουν οι διαφορές των τιμών, μεταξύ των δύο καταστάσεων, και ουσιαστικά λειτουργεί σαν ρυθμιστής της ταχύτητας μάθησης του πράκτορά μας, ενώ οι τιμές του κυμαίνονται μεταξύ 0 και 1. Ο συντελεστής r είναι η τιμή της ανταμοιβής ή ενίσχυσης του πράκτορά μας για κάθε κίνησή του. Η τιμή του r είναι 0 σε όλες τις άλλες καταστάσεις εκτός από τις τερματικές στο παιχνίδι που μελετούμε.

Εφαρμόζοντας στην πράξη την μέθοδο των χρονικών διαφορών, ουσιαστικά έχουμε δύο επιλογές όσον αφορά την τιμή της συνάρτησης αξιολόγησης, που αφορά την κατάσταση-απόγονο, την οποία θα χρησιμοποιήσουμε στην εξίσωση της ενημέρωσης των βαρών. Η μία επιλογή είναι η τιμή που θα μας επιστραφεί σαν βέλτιστη, εφαρμόζοντας τις τεχνικές αναζήτησης που βασίζονται πάνω στον MiniMax, αδιαφορώντας έτσι για την πραγματική κίνησή του αντιπάλου. Στην περίπτωση αυτή

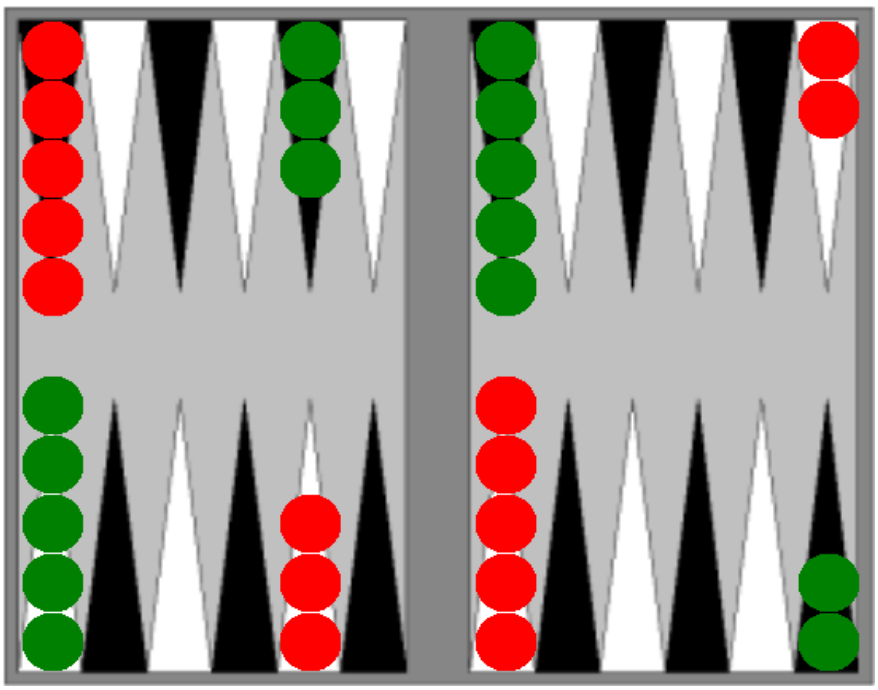
καταφέρνουμε να εξερευνήσουμε μεγαλύτερο μέρος του δέντρου και έτσι να συμπεριλάβουμε στην εκπαίδευση του πράκτορά μας, κομμάτια του δέντρου στα οποία σε άλλη περίπτωση μπορεί να μην οδηγούμασταν ποτέ. Η δεύτερη επιλογή είναι η τιμή της κατάστασης που θα προκύψει μετά την πραγματική απάντηση του αντιπάλου στην κίνησή μας. Σε αυτήν την περίπτωση ο πράκτοράς μας μαθαίνει να παίζει βέλτιστα απέναντι στον συγκεκριμένο αντίπαλο, πράγμα το οποίο θα ήταν στην πράξη επιθυμητό μόνο εάν ο αντίπαλος αυτός θεωρείται αντικειμενικά πολύ καλός.

Κεφάλαιο 3

Το Παιχνίδι Backgammon

3.1 Ιστορία και προέλευση του παιχνιδιού Backgammon

Το Backgammon [10] είναι ένα επιτραπέζιο παιχνίδι που παίζεται με δύο παίκτες. Κάθε παίκτης ξεκινάει με 15 πιόνια τα οποία στήνονται με μία συγκεκριμένη αρχική διάταξη πάνω στο ταμπλό του παιχνιδιού όπως φαίνεται στο Σχήμα 3.1. Ο παίκτης ο οποίος έχει τα κόκκινα πιόνια έχει σαν αρχική θέση την πάνω δεξιά γωνία (με τα δύο κόκκινα πιόνια) και κινείται αριστερόστροφα. Ο παίκτης με τα πράσινα πιόνια έχει σαν αρχική θέση την κάτω δεξιά γωνία (με τα δύο πράσινα πιόνια) και κινείται δεξιόστροφα.



Σχήμα 3.1 Αρχική κατάσταση στο παιχνίδι Backgammon.

Σκοπός τού παιχνιδιού είναι να συγκεντρώσει ο παίκτης όλα του τα πιόνια μέσα στις 6 πρώτες θέσεις του αντιπάλου του, και μετά να αφαιρέσει σταδιακά τα πιόνια του από το ταμπλό σύμφωνα με τις δυνατές κινήσεις που του επιτρέπουν τα ζάρια. Νικητής του παιχνιδιού είναι ο αυτός ο οποίος θα αφαιρέσει πρώτος όλα του τα πιόνια. Σε κάποιες παραλλαγές του παιχνιδιού όταν κάποιος παίκτης αφαιρέσει όλα του τα πιόνια και ο αντίπαλος δεν έχει προλάβει να αφαιρέσει κανένα από τα δικά του ακόμα, τότε ο παίκτης αυτός κερδίζει αμέσως δύο νίκες για το παιχνίδι αυτό.

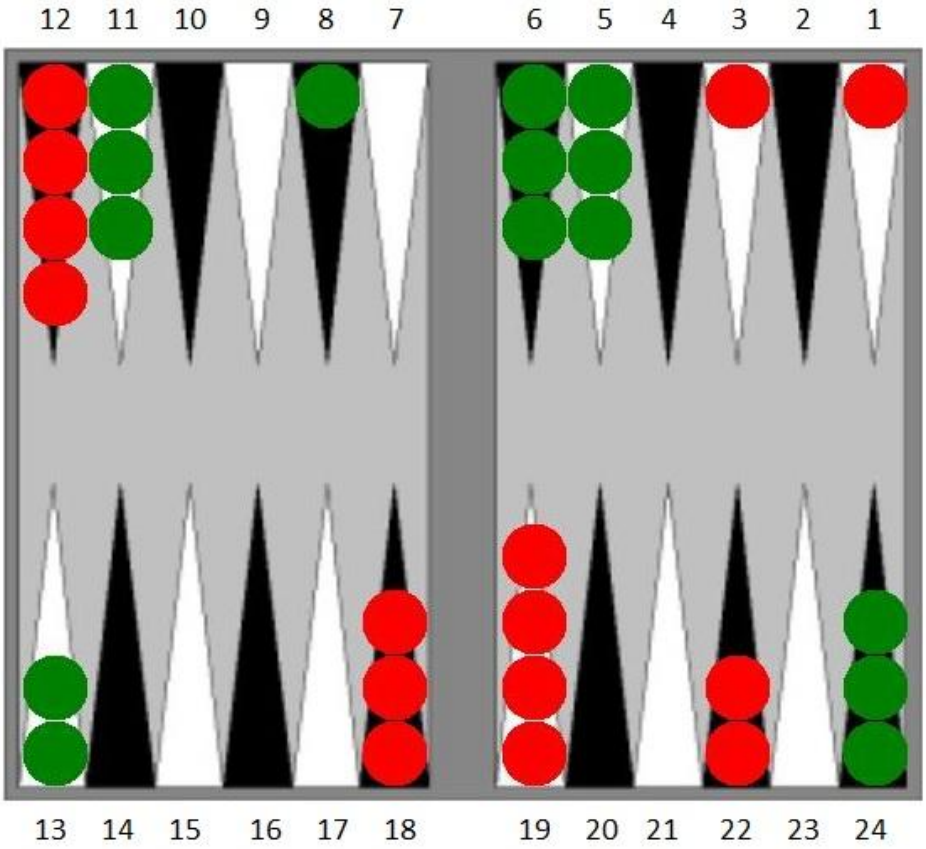
Ο όρος "Τάβλι" (Backgammon) στις Αγγλόφωνες χώρες ταυτίζεται ουσιαστικά με το παιχνίδι "πόρτες". Σε άλλες χώρες, όπως η Ελλάδα, ο όρος τάβλι συμπεριλαμβάνει πολλά παιχνίδια που παίζονται στο ίδιο ταμπλό με το παιχνίδι πόρτες, αλλά έχουν πολλές διαφορές στον τρόπο παιξίματος από αυτό. Κάποια είδη παιχνιδιών χρησιμοποιούν τρία ζάρια αντί για δύο, σε κάποια είδη οι θέσεις εκκίνησης είναι διαφορετικές, ενώ σε κάποια άλλα και οι δύο παίκτες κινούν τα πιόνια τους προς την ίδια κατεύθυνση.

Η πρώτη αναφορά στο παιχνίδι στην αρχαία Ελλάδα συναντάται με το όνομα "Πεσσοί" ενώ στη Ρωμαϊκή Αυτοκρατορία με το όνομα "Ludus Duodecim Scriptorum" ("παιχνίδι των δώδεκα γραμμών").

Ιστορικά, ο όρος "Τάβλι" προέρχεται από τους προγόνους του παιχνιδιού, καταγεγραμμένους στην Αγγλία ως "Tables" από το 14ο αιώνα μέχρι και το τέλος του 17ου, και πριν από αυτή την ονομασία συναντάται η Λατινική ονομασία "Tabula", από την οποία και προέκυψε η ονομασία "Tables". Η ονομασία όμως "Tables" είναι ένας γενικός όρος που καλύπτει ποικιλία παιχνιδιών ελαφρώς διαφορετικών μεταξύ τους. Η αντικατάσταση του όρου από το "Backgammon" το 17ο αιώνα πιθανότατα αντικατοπτρίζει την αυξανόμενη δημοτικότητα της παραλλαγής "πόρτες", με την οποία είμαστε περισσότερο οικειοποιημένοι και αναφορές της μπορούν να βρεθούν πίσω στο 13ο αιώνα στην Ισπανία με το όνομα "Todas Tablas".

Το όνομα "tavli" προέρχεται από το μεσαίωνα, όταν και το παιχνίδι είχε ήδη διαδοθεί στην Ευρώπη με ονόματα όπως "Tables", "Tavola Reale" και "Tablas Reales". Το σύγχρονο όνομα του παιχνιδιού στα Αγγλικά (Backgammon), προέρχεται από τα μέσα του 17ου αιώνα, όταν οι Σάξονες το ονόμασαν από τις λέξεις "bac"

(πίσω) και "gamen" (παιχνίδι), επειδή στο παιχνίδι αυτό όταν ένα πιόνι "χτυπήσει" ένα άλλο, αυτό επιστρέφει πίσω στην αρχή. Στο Σχήμα 3.2 φαίνεται ένα χαρακτηριστικό στιγμιότυπο σε μία παρτίδα Backgammon :



Σχήμα 3.2 Στιγμιότυπο από παρτίδα Backgammon

Παίρνοντας σαν δεδομένο την κατάσταση από το στιγμιότυπο του παραπάνω σχήματος και θεωρώντας ότι έχει σειρά να παίξει ο παίκτης που έχει στην κατοχή του τα κόκκινα πιόνια, ένας υποθετικός συνδυασμός των δύο ζαριών με αντίστοιχες τιμές 4 και 5, μεταξύ άλλων, θα έδινε τις παρακάτω χαρακτηριστικές νόμιμες κινήσεις στον παίκτη αυτό :

- Κίνηση του πιονιού από την θέση 3 χρησιμοποιώντας την τιμή 5 του ενός ζαριού και στην συνέχεια, χρησιμοποιώντας την τιμή 4 του δεύτερου ζαριού, κίνηση οποιουδήποτε πιονιού από τις θέσεις μεταξύ των 8, 12, 18 και 19. Οποιαδήποτε από τις παραπάνω κινήσεις θα οδηγούσε σε αιχμαλωσία του πιονιού του αντιπάλου που βρίσκεται στην θέση 8.
- Κίνηση ενός πιονιού από την θέση 18 χρησιμοποιώντας την τιμή 5 από το

αντίστοιχο ζάρι καταλήγοντας στην θέση 23 και κίνηση ενός πιονιού από την θέση 19 χρησιμοποιώντας την τιμή 4 από το άλλο ζάρι με κατάληξη και αυτού στην θέση 23. Το αποτέλεσμα της παραπάνω κίνησης θα ήταν η δημιουργία μιας "πόρτας" του παίκτη με τα κόκκινα πιόνια στην θέση 23 τα αντίστοιχα πιόνια της οποίας θα ήταν προστατευμένα από αιχμαλωσία.

3.2 Η πολυπλοκότητα στο παιχνίδι Backgammon

Πριν αρχίσουμε να αναλύουμε την υλοποίηση του παιχνιδιού θα πρέπει να αναφέρουμε κάποιες λεπτομέρειες για την πολυπλοκότητα του Backgammon αλλά και την αλληλεπίδραση των δύο παικτών μεταξύ τους όσον αφορά στις κινήσεις τους πάνω στο ταμπλό του παιχνιδιού.

Το Backgammon είναι ένα παιχνίδι δύο παικτών στο οποίο αυτοί παίζουν εναλλάξ, ρίχνοντας δύο ζάρια ο καθένας, μετακινώντας τα πιόνια τους σε αντίθετες κατευθύνσεις ο ένας από τον άλλο πάνω στο ταμπλό. Αυτά που καθορίζουν ουσιαστικά τις δυνατές κινήσεις που έχει ένας παίκτης στη διάθεσή του σε κάποιο συγκεκριμένο στιγμιότυπο του παιχνιδιού είναι τα εξής :

α) Η ένδειξη των δύο ζαριών.

Όσον αφορά στην ένδειξη των ζαριών, κάθε παίκτης έχει τη δυνατότητα να μετακινήσει ένα από τα διαθέσιμα πιόνια του για κάθε ζάρι ή ένα πιόνι για το άθροισμα των ζαριών που θα φέρει. Εάν τα ζάρια φέρουν τον ίδιο αριθμό, τότε αυτός έχει στην διάθεσή του τις διπλάσιες κινήσεις από ότι σε μία περίπτωση όπου τα ζάρια θα ήταν διαφορετικά.

Τα παραπάνω σενάρια, για τις κινήσεις του κάθε παίκτη σε σχέση με τα ζάρια, είναι δυνατά εφόσον και μόνο έχει στην διάθεσή του νόμιμες κινήσεις να εφαρμόσει, όπως θα εξηγηθεί παρακάτω.

β) Η θέση και η διάταξη των πιονιών του παίκτη.

Αναφορικά στη θέση και στη διάταξη των πιονιών του, ο κάθε παίκτης έχει τη δυνατότητα να ξεκινήσει μία κίνηση του από ένα σημείο του ταμπλό όπου βρίσκονται

τουλάχιστον ένα από τα δικά του πιόνια. Σε περίπτωση όπου κάποια από τα πιόνια του έχουν αιχμαλωτιστεί από τον αντίπαλο, τότε αυτός είναι υποχρεωμένος να τα επανεισαγάγει όλα μέσα στο ταμπλό προτού να έχει τη δυνατότητα να κάνει οποιαδήποτε άλλη κίνηση.

γ) Η θέση και η διάταξη των πιονιών του αντιπάλου.

Όσον αφορά στην θέση και στην διάταξη των πιονιών του αντιπάλου, κάθε παίκτης έχει την δυνατότητα να μετακινήσει κάποιο από τα διαθέσιμα πιόνια του σε μία νέα θέση εφόσον και μόνο στη θέση αυτή βρίσκεται το πολύ ένα πιόνι του αντιπάλου. Εάν ο αντίπαλος διαθέτει πάνω από ένα δικά του πιόνια στη θέση αυτή τότε ο παίκτης δεν μπορεί να μετακινήσει εκεί κάποιο δικό του και ουσιαστικά μπλοκάρεται οποιαδήποτε κίνησή του προς την θέση αυτή.

Εάν η διάταξη των πιονιών του αντιπάλου είναι τέτοια ώστε να μπλοκάρονται όλες οι κινήσεις του παίκτη, σύμφωνα με την ένδειξη των ζαριών, τότε αυτός χάνει τη σειρά του και συνεχίζει ο αντίπαλος. Εάν η διάταξη των πιονιών του αντιπάλου είναι τέτοια ώστε να αφήνει περιθώριο στον παίκτη να παίξει μόνο το ένα από τα δύο ζάρια, ή το ένα ή το άλλο, αλλά όχι και τα δύο μαζί, τότε αυτός είναι υποχρεωμένος να παίξει το μεγαλύτερο από αυτά και θα τελειώσει εκεί η κίνησή του.

Ο προγραμματισμός ενός υπολογιστή για να παίξει σε υψηλό επίπεδο τάβλι έχει βρεθεί να είναι ένα μάλλον δύσκολο εγχείρημα. Με μια απλουστευμένη λογική θα μπορούσε να σχεδιαστεί ένας πράκτορας ο οποίος να συμβουλεύεται ένα look-up table ο οποίος θα περιέχει ουσιαστικά τις αντιστοιχίες καταστάσεων-ενεργειών (state-actions). Ωστόσο, μια τέτοια προσέγγιση δεν είναι εφικτή για το παιχνίδι Backgammon, λόγω του τεράστιου αριθμού των πιθανών καταστάσεων (που εκτιμάται σε πάνω από 10^{20}). Επιπλέον, η κλασική μέθοδος της βαθιάς αναζητήσεως, η οποία λειτούργησε τόσο καλά σε παιχνίδια όπως το σκάκι, την ντάμα και το Othello, δεν είναι εφικτή λόγω του μεγάλου παράγοντα διακλάδωσης του δέντρου αναζήτησης.

Οι δυνατοί συνδυασμοί για τα ζάρια είναι 21 και ένας τέτοιος συνδυασμός μπορεί να δίνει την δυνατότητα στον παίκτη για πάνω από 20 νόμιμες κινήσεις, με αποτέλεσμα ο παράγοντας διακλάδωσης να φτάνει πολλές φορές, για κάποια συγκεκριμένη κατάσταση τις μερικές εκατοντάδες. Ο παράγοντας αυτός

διακλάδωσης στο παιχνίδι Backgammon είναι κατά πολύ μεγαλύτερος από τα παιχνίδια όπως το checkers και το σκάκι, όπου ο λόγος διακλάδωσης στα δέντρα αναζήτησής τους κυμαίνεται από 8-10 και 30-40 αντίστοιχα. Ακόμα και η χρήση υπέρ-υπολογιστών δεν θα μπορούσε να κάνει μια αξιόλογη σε βάθος αναζήτηση σε ένα τέτοιο δέντρο όπως αυτό του Backgammon.

3.3 Ο Στόχος μας

Στόχος μας είναι η δημιουργία ενός προγράμματος-πράκτορα που θα μπορεί να παίξει όσο το δυνατόν καλύτερα το παιχνίδι Backgammon. Για την επίτευξη του στόχου αυτού θα χρησιμοποιήσουμε ειδικές μεθόδους αναζήτησης καθώς και τεχνικές από το πεδίο της ενισχυτικής μάθησης. Αρχικά θα χρειαστεί να υλοποιήσουμε ένα πρόγραμμα με γραφικό περιβάλλον πάνω στο οποίο θα μπορούν να διεξαχθούν παρτίδες του παιχνιδιού αυτού και στο οποίο θα στηριχθούμε για να υλοποιήσουμε και να εκπαιδεύσουμε τον πράκτορά μας. Οι αλγόριθμοι αναζήτησης που θα χρησιμοποιηθούν βασίζονται στον αλγόριθμο MiniMax, ο οποίος ουσιαστικά υπολογίζει την τιμή της τρέχουσας κατάστασης χρησιμοποιώντας έναν απλό αναδρομικό υπολογισμό των αντίστοιχων τιμών κάθε διάδοχης κατάστασης, υλοποιώντας άμεσα τις εξισώσεις που τις ορίζουν. Θα χρησιμοποιήσουμε επίσης μια καλύτερη εκδοχή του αλγόριθμου αυτού η οποία ονομάζεται Άλφα-Βήτα και η οποία εφαρμόζοντας την τεχνική του κλαδέματος (pruning) μειώνει αποτελεσματικά τον παράγοντα διακλάδωσης του δέντρου αναζήτησης του παιχνιδιού. Για την εκπαίδευση του πράκτορά μας θα χρησιμοποιηθεί η μέθοδος TD (Temporal Difference), η οποία εκμεταλλεύεται την κάθε μετάβαση από μία κατάσταση s_1 σε μία άλλη s_2, μετά από μία κίνηση του παίκτη μας, με σκοπό την βελτίωσή του.

3.4 Προηγούμενες εργασίες για το Backgammon

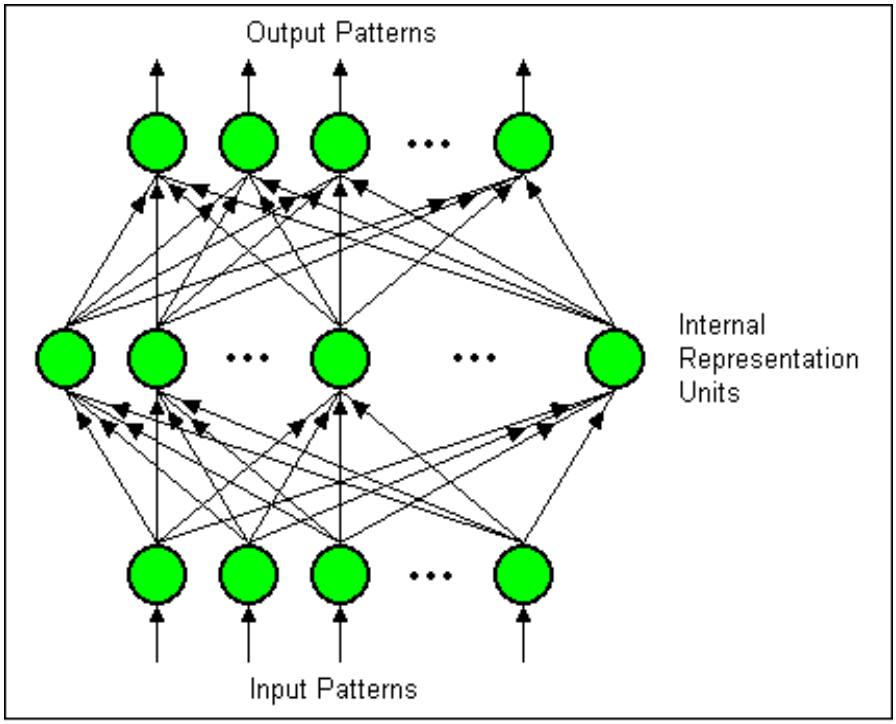
Πλήθος εργασιών έχουν πραγματοποιηθεί για το παιχνίδι Backgammon όπως πολλά είναι και τα εμπορικά προγράμματα για τα οποία οι κατασκευαστές τους υποστηρίζουν ότι μπορούν να αντιμετωπίσουν ισάξια φυσικούς παίκτες. Τα πιο γνωστά από αυτά είναι το Snowie [13], το JellyFish [12] καθώς και το ελληνικό "Παλαμήδης" [11], το οποίο μάλιστα κατέκτησε την πρώτη θέση στην Ολυμπιάδα

Προγραμμάτων για ηλεκτρονικά παιχνίδια που διοργανώνεται από τον οργανισμό ICGA για το έτος 2011.

Τα περισσότερα από αυτά τα εμπορικά προγράμματα χρησιμοποιούν μεθόδους μάθησης πάνω σε Νευρωνικά Δίκτυα (Neural Networks). Οι εφαρμογές αυτές ουσιαστικά στηρίζονται στην εργασία του Gerry Tesauro, ο οποίος το 1992 μετά από έρευνα πολλών ετών κατέληξε στην δημιουργία του TD-Gammon [4], ενός πράκτορα ο οποίος παίζοντας με αντίπαλο τον εαυτό του, "εκπαιδεύει" ένα περίπλοκο πολύ-επίπεδο νευρωνικό δίκτυο το οποίο χρησιμοποιεί σαν συνάρτηση αξιολόγησης για την επιλογή της κατάλληλης κίνησης σε μια δεδομένη κατάσταση του παιχνιδιού. Για την ανανέωση και προσαρμογή στα βάρη των κόμβων του νευρωνικού δικτύου, ο Tesauro χρησιμοποίησε τον αλγόριθμο TD(lambda). Η αλλαγή στα βάρη γίνεται μετά από κάθε κίνηση μέσα στο παιχνίδι και περιγράφεται από την παρακάτω εξίσωση :

$$w_{t+1} - w_t = \alpha(Y_{t+1} - Y_t) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w Y_k$$

Στο Σχήμα 3.3 φαίνεται η αρχιτεκτονική του νευρωνικού δικτύου που χρησιμοποίησε ο Gerry Tesauro για την δημιουργία του TD-Gammon.



Σχήμα 3.3 Το νευρωνικό δίκτυο του TD-Gammon [4].

Εφαρμόζοντας τις παραπάνω τεχνικές και μετά από κάποιες χιλιάδες παιχνίδια

“εκπαίδευσης” του νευρωνικού δικτύου, ο αλγόριθμος του Tesauro κατάφερε να φτάσει στην σύγκλιση των τιμών στα βάρη των κόμβων και τελικά στην εύρεση αποδοτικών στρατηγικών απέναντι σε φυσικούς-παίκτες στο παιχνίδι του Backgammon.

Η εργασία του Tesauro αποτελεί ακόμα και σήμερα την πρώτη αναφορά σε πολλές εργασίες πάνω στο επιστημονικό πεδίο της ενισχυτικής μάθησης. Είναι επίσης σημαντικό να αναφέρουμε ότι κάποιοι από τους καλύτερους παίκτες παγκοσμίως, αναφορικά με το παιχνίδι Backgammon, αναθεώρησαν την λογική που αντιμετώπιζαν κάποιες “κλασικές” κινήσεις στο παιχνίδι αυτό παρακολουθώντας την απόκριση από εφαρμογές που στηρίχθηκαν πάνω στην εργασία του Tesauro.

Κεφάλαιο 4

Η δική μας προσέγγιση

4.1 Κάποιες παραδοχές για το παιχνίδι

Κατά την διάρκεια της βιβλιογραφικής και διαδικτυακής μας έρευνας για την εργασία αυτή, βρεθήκαμε αντιμέτωποι με διάφορες παραλλαγές όσον αφορά στο παιχνίδι Backgammon. Η διαφοροποίηση των παραλλαγών αυτών δεν ήταν τόσο στους κανόνες του παιχνιδιού, αλλά περισσότερο στο τρόπο με τον οποίο οριζόταν η ανταμοιβή (reward) για τις τερματικές καταστάσεις, αλλά και για διάφορες ενέργειες κατά την διάρκεια του. Η λογική της ανταμοιβής είναι ένας παράγοντας ο οποίος καθορίζει σημαντικά τόσο την εκπαίδευση του πράκτορά μας όσο και την περαιτέρω συμπεριφορά του απέναντι στους αντιπάλους του.

Στην εργασία αυτή έγινε τελικά υλοποίηση της πιο κοινότυπης εκδοχής από τις παραλλαγές αυτές. Θεωρούμε ότι στα πλαίσια μιας παρτίδας Backgammon η νίκη για κάποιον παίκτη σημαίνει αυτόματα και την ήττα για τον αντίπαλό του. Δεν υπάρχει ουσιαστικά κατάσταση "μερικής νίκης" ή "μερικής ήττας" με αντίστοιχο καταμερισμό στα κέρδη. Επίσης οι κανόνες του παιχνιδιού δεν επιτρέπουν εκ των πραγμάτων την ύπαρξη καταστάσεων ισοπαλίας. Επομένως οι μοναδικές ανταμοιβές οι οποίες υιοθετήθηκαν για τις τερματικές καταστάσεις, σύμφωνα με τη λογική του μηδενικού αθροίσματος (zero sum), είναι δύο τιμές αντίθετες μεταξύ τους, που αντιστοιχούν στις περιπτώσεις της νίκης και της ήττας για τον κάθε παίκτη και συγκεκριμένα οι τιμές +1 και -1 αντίστοιχα.

Το περιβάλλον του παιχνιδιού θεωρείται πλήρως παρατηρήσιμο, επομένως ανά πάσα στιγμή αποθηκεύεται σαν κατάσταση η πλήρης εικόνα του παιχνιδιού, σε μια

κατάλληλη δομή. Η δομή αυτή περιέχει τις θέσεις των πιονιών τα οποία είναι ενεργά για τον κάθε παίκτη πάνω στο ταμπλό, τα πιόνια τα οποία είναι "φυλακισμένα" από τον αντίπαλο για κάθε πλευρά, τα πιόνια τα οποία έχουν αφαιρεθεί από κάθε παίκτη, καθώς και η τιμή των ζαριών. Ο αριθμός από τα πιόνια τα οποία έχουν τοποθετηθεί εκτός παιχνιδιού θα μπορούσε να βρεθεί αφαιρετικά, επειδή όμως η τιμή αυτή χρησιμοποιείται σε κάποιο χαρακτηριστικό της συνάρτησης αξιολόγησης, όπως θα δούμε παρακάτω, επιλέξαμε να το αποθηκεύουμε ξεχωριστά για λόγους ταχύτητας.

4.2 Μηχανισμός αναζήτησης

Το παιχνίδι Backgammon είναι ένα ανταγωνιστικό παιχνίδι στο οποίο ο στόχος του κάθε παίκτη συγκρούεται άμεσα με τους στόχους του αντιπάλου του. Για τις ανάγκες λοιπόν της αναζήτησης της βέλτιστης κίνησης μέσα σε ένα τέτοιο περιβάλλον, χρησιμοποιήσαμε την λογική του αλγόριθμου MiniMax. Σύμφωνα με την λογική αυτή οι ενέργειες των δυο παικτών, οι οποίες εναλλάσσονται μεταξύ τους, λειτουργούν με τρόπο τέτοιο ώστε να μεγιστοποιούν κάθε φορά τα κέρδη τους, λαμβάνοντας πάντα υπόψη τις αντιδράσεις του αντιπάλου. Για τον παίκτη max η μεγιστοποίηση σημαίνει και το μεγαλύτερο δυνατό αριθμητικά κέρδος ενώ για τον παίκτη min σημαίνει το μικρότερο δυνατό αριθμητικά κέρδος.

Ο παράγοντας διακλάδωσης του δέντρου αναζήτησης του παιχνιδιού για τους κόμβους max και min, δηλαδή οι νόμιμες κινήσεις που έχουν στην διάθεσή τους οι δύο παίκτες κατά την διάρκεια του παιχνιδιού, είναι πολύ μεγάλος και μπορεί να κυμαίνεται από την τιμή 0 έως και μερικές εκατοντάδες. Αυτό πρακτικά μας επηρεάζει σε σχέση με το μέγιστο βάθος στο οποίο μπορεί να φτάσει η αναζήτηση μας μέσα σε ένα λογικό χρονικό διάστημα. Συνεπώς σε καταστάσεις στις οποίες οι τιμές των ζαριών και το στήσιμο των πιονιών των δύο αντιπάλων πάνω στο ταμπλό δεν επιτρέπει μεγάλο "άνοιγμα" του δέντρου από την αρχή ενός "κύκλου" αναζήτησης του MiniMax, το βάθος αναζήτησης μπορεί να φτάσει έως και επτά στρώσεις, μέσα σε ένα ικανοποιητικό χρόνο. Απεναντίας σε καταστάσεις στις οποίες παρατηρείται μεγάλο "άνοιγμα" από την αρχή, για τον ίδιο χρόνο το βάθος αναζήτησης μπορεί να φτάσει μέχρι και τρεις στρώσεις. Λόγω της ανομοιομορφίας αυτής που παρατηρήθηκε στα διάφορα σενάρια αναζήτησης, καταλήξαμε, μετά από την παρατήρηση ενός μεγάλου αριθμού παρτίδων, να θέσουμε, στην δική μας

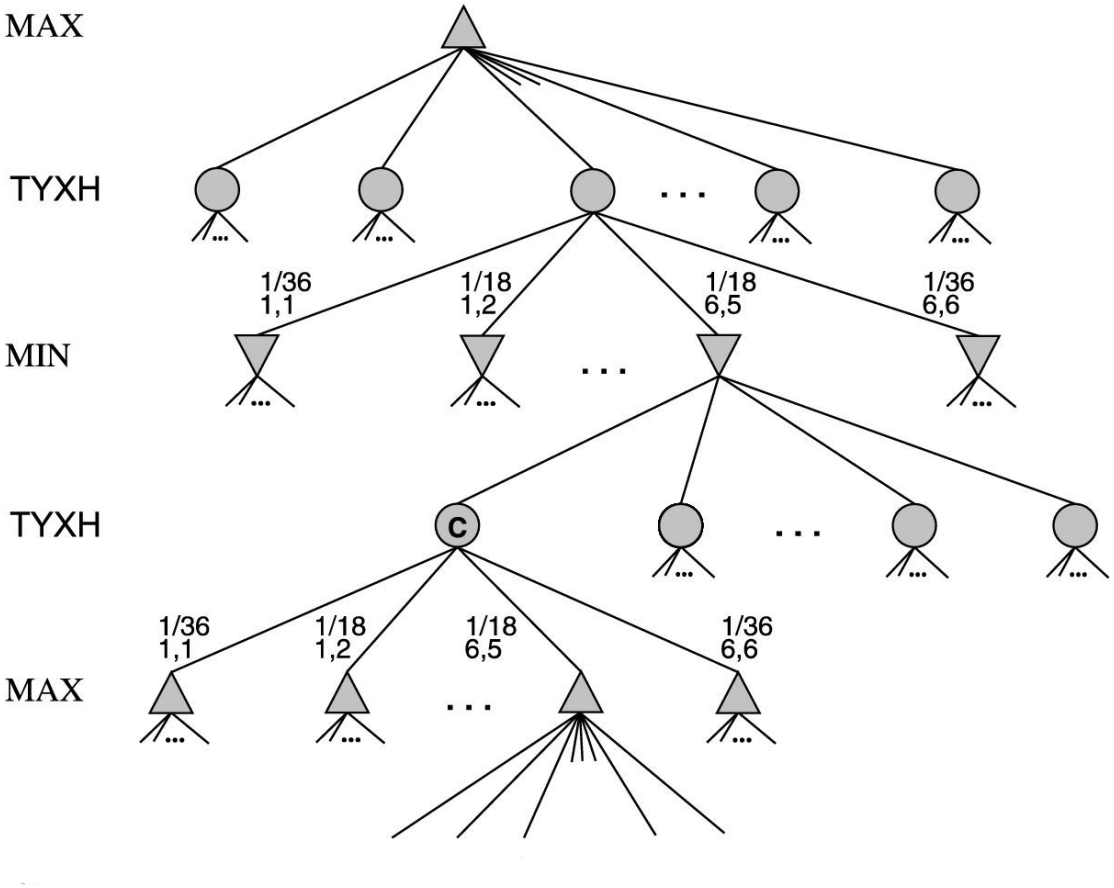
προσέγγιση, σαν μέγιστο βάθος αναζήτησης του αλγόριθμου MiniMax τις πέντε στρώσεις. Δηλαδή, πέρα από την τρέχουσα κατάσταση (ρίζα του δέντρου) εξετάζουμε μία κίνηση του παίκτη μας ($1^{η}$ στρώση) μία ζαριά ($2^{η}$ στρώση) και μία κίνηση ($3^{η}$ στρώση) του αντιπάλου και στη συνέχεια μία ζαριά ($4^{η}$ στρώση) και μία κίνηση ($5^{η}$ στρώση) του παίκτη μας.

Τα δύο ζάρια είναι αυτά τα οποία αντιπροσωπεύουν το στοιχείο της τύχης στο παιχνίδι Backgammon. Η ύπαρξη του απρόβλεπτου αυτού παράγοντα διαμορφώνει το δέντρο αναζήτησης του παιχνιδιού όπως φαίνεται στο Σχήμα 4.1. Όπως βλέπουμε στο σχήμα ανάμεσα στις στρώσεις του max και του min παρεμβάλεται μία στρώση για τους κόμβους τύχης του παιχνιδιού που αντιπροσωπεύουν τα αντίστοιχα δυνατά αποτελέσματα από την ρίψη των δύο ζαριών. Οι τιμές των δύο αυτών ζαριών θεωρούνται ισοπίθανες και κυμαίνονται από την τιμή ένα (1) έως την τιμή έξι (6), για το καθένα. Η σειρά με την οποία γίνεται η ρίψη τους δεν διαφοροποιεί τις νόμιμες κινήσεις του παίκτη που έχει σειρά να παίξει εκείνη τη στιγμή. Επομένως η τιμή που αποδίδεται, σαν πιθανότητα να συμβεί το συγκεκριμένο ενδεχόμενο, στους κόμβους τύχης του δέντρου είναι αντίστοιχα για τους συνδυασμούς των ζαριών με διαφορετική τιμή ο αριθμός 2/36, ενώ για τους συνδυασμούς με ίδια τιμή ο αριθμός 1/36.

Η ύπαρξη κόμβων τύχης στο δέντρο αναζήτησης του παιχνιδιού Backgammon μας οδήγησε στην χρησιμοποίηση της παραλλαγής ExpectiMiniMax. Με την μέθοδο αυτή ουσιαστικά προστίθεται μία ακόμα διαδικασία στην λογική που ακολουθεί ο MiniMax. Η νέα αυτή διαδικασία αποδίδει τιμές στους κόμβους τύχης του δέντρου όπως ακριβώς γίνεται και τους κόμβους max και min. Η τιμή αυτή είναι το άθροισμα των τιμών MiniMax των κόμβων-απογόνων του κάθε κόμβου τύχης πολλαπλασιαζόμενος ο κάθε ένας με την τιμή της πιθανότητας του αντίστοιχου συνδυασμού ζαριών όπως αναφέρεται παραπάνω. Έτσι η τιμή ενός κόμβου τύχης είναι η αναμενόμενη τιμή βάσει των τιμών των παιδιών του και της κατανομής πάνω στις πιθανές ζαριές.

Ο μεγάλος αριθμός καταστάσεων, λόγω της ύπαρξης του τυχαίου παράγοντα αλλά και του μεγάλου αριθμού νόμιμων κινήσεων που μπορεί να έχει στην διάθεσή του κάποιος παίκτης, κατέστησε αναγκαία τη χρήση της τεχνικής αναζήτησης με

υπαναχώρηση αλλά και της τεχνικής του κλαδέματος Άλφα-Βήτα. Με την τεχνική της υπαναχώρησης αποθηκεύουμε ουσιαστικά μόνο τις απαραίτητες καταστάσεις, κατά την αναζήτηση του δέντρου, ενώ παράλληλα μπορούμε να κάνουμε αναίρεση κάποιας κίνησης και να συνεχίσουμε από μία προηγούμενη κατάσταση, καταφέρνοντας με αυτόν τον τρόπο να έχουμε μεγάλο κέρδος όσον αφορά την χρήση μνήμης του υπολογιστή. Με την τεχνική του κλαδέματος Άλφα-Βήτα πραγματοποιήθηκε σημαντική μείωση των κόμβων που πρέπει να εξετάσει ο πράκτοράς μας, πράγμα το οποίο μειώνει σημαντικά τον χρόνο αναζήτησης και βελτιώνει την συνολική απόδοσή του.



Σχήμα 4.1 Μέρος από το δέντρο αναζήτησης για το παιχνίδι Backgammon.

Ένας από τους κανόνες του παιχνιδιού ο οποίος επηρέασε την αναζήτηση του πράκτορά μας ήταν αυτός της διπλής κίνησης όταν οι τιμές των ζαριών είναι οι ίδιες. Οι "διπλές" όπως ονομάζονται, υποχρεώνουν τον παίκτη που έχει σειρά να παίξει να κάνει δύο συνεχόμενες κινήσεις χωρίς ο αντίπαλος να κάνει καμία ενέργεια. Τα ήδη στενά περιθώρια που είχαμε όσον αφορά τον χρόνο απόκρισης και του βάθους

αναζήτησης δεν μας επέτρεψαν να ακολουθήσουμε την λογική των δύο "κανονικών" κινήσεων όπως με τις υπόλοιπες περιπτώσεις. Υποχρεωτικά έπρεπε να θυσιάσουμε το βάθος αναζήτησης, για να μπορέσουμε να κρατήσουμε τον χρόνο απόκρισης του πράκτορά μας μέσα σε λογικά πλαίσια. Ο τρόπος που προσεγγίσαμε το παραπάνω πρόβλημα ακολουθεί την λογική της μηδενικής κίνησης (null move). Με την τεχνική αυτή ουσιαστικά εκτελούμε μια ρηχή αναζήτηση μίας στρώσης στο δέντρο αναζήτησης του παιχνιδιού για την πρώτη κίνηση του παίκτη μας, δηλαδή αναζητούμαι την βέλτιστη επιλογή από όλους τους κόμβους-διαδόχους, έχοντας εκτελέσει όλες τις νόμιμες κινήσεις που προκύπτουν από την τρέχουσα κατάσταση. Μετά την εύρεση της βέλτιστης αυτής επιλογής, η κίνηση αυτή εκτελείται και η κατάσταση που προκύπτει ορίζεται σαν αρχική κατάσταση για την διαδικασία της αναζήτησης της δεύτερης κίνησης η οποία εκτελείται κανονικά όπως την έχουμε περιγράψει μέχρι τώρα.

4.3 Συνάρτηση αξιολόγησης

Η μορφή της συνάρτησης αξιολόγησης που χρησιμοποιήσαμε είναι μια σταθμισμένη γραμμική εξίσωση που αποτελείται από τα χαρακτηριστικά τα οποία επιλέξαμε τελικά πολλαπλασιαζόμενα επί κάποια βάρη. Στην προσπάθειά μας να βρούμε κάποια καλά χαρακτηριστικά για την συνάρτησή μας κάναμε πολλούς πειραματισμούς με διάφορες ομάδες από τέτοια. Βασικός μας στόχος ήταν η ομάδα αυτή των χαρακτηριστικών που θα χρησιμοποιήσουμε να δίνει, όσο το δυνατόν καλύτερα, την δυνατότητα να απεικονιστεί η πραγματική κατάσταση αλλά και η δυναμική του παιχνιδιού σε κάθε περίπτωση.

Κάποιες από αυτές τις ομάδες απορρίφθηκαν εξαρχής λόγω του ότι κατά την διάρκεια της εκπαίδευσης του πράκτορά μας δεν υπήρξε ποτέ σύγκλιση στις τιμές των βαρών. Τα αποτελέσματα της εκπαίδευσης, απέναντι πάντα σε έναν σταθερό αντίπαλο, χρησιμοποιώντας κάθε φορά και μια ομάδα από αυτά τα χαρακτηριστικά, συγκρίθηκαν μεταξύ τους σε τουρνουά των 50 παιχνιδιών. Τελικά τα αρχικά χαρακτηριστικά στα οποία καταλήξαμε ήταν αυτά τα οποία απεικονίζονται στο Σχήμα 4.2.

Μετά την εκπαίδευση και την διεξαγωγή αρκετών παρτίδων, διαπιστώσαμε ορισμένες αδυναμίες στην συμπεριφορά του πράκτορά μας, ειδικά σε καταστάσεις όπως αυτές

που είναι κοντά στις τερματικές καταστάσεις του παιχνιδιού αλλά και σε αυτές όπου κάποιο από τα πιόνια ενός παίκτη βρισκόταν σε κατάσταση αιχμαλωσίας. Η συμπεριφορά αυτή οφειλόταν στο ότι ο πράκτορας μας θα έπρεπε να λαμβάνει υπόψη του περισσότερους παράγοντες του παιχνιδιού στις ιδιαίτερες αυτές καταστάσεις. Για τον λόγο αυτό δημιουργήσαμε κάποια επιπλέον χαρακτηριστικά τα οποία είναι συνδυασμός αυτών που είχαμε επιλέξει στην αρχή έτσι ώστε να δώσουμε την δυνατότητα στην συνάρτησή μας να ανταποκριθεί καλύτερα σε αυτές τις περιπτώσεις. Τα επιπλέον αυτά χαρακτηριστικά ουσιαστικά είναι το γινόμενο της τιμής του ενός χαρακτηριστικού με την συμπληρωματική τιμή του άλλου, όπως φαίνεται παρακάτω και απεικονίζονται στο Σχήμα 4.3.

$$f\_i(s) \times (1 - f\_k(s))$$

Ο παραπάνω τρόπος συνδυασμού χαρακτηριστικών υποδηλώνει ότι μεγάλες τιμές στο νέο χαρακτηριστικό επιτυγχάνονται όταν στο ζευγάρι που συνδυάζεται υπάρχει αντι-συσχέτιση, ενώ αντίθετα η ύπαρξη συσχέτισης οδηγεί σε μικρές τιμές. Καθ' όλη την διάρκεια των πειραματισμών αυτών για την εύρεση των κατάλληλων χαρακτηριστικών γινόταν πάντοτε κανονικοποίηση των τιμών τους ώστε αυτές να κυμαίνονται στο διάστημα από το μηδέν έως το ένα.

Όνομα	Περιγραφή
F1	Ο αριθμός από τις "πόρτες" που έχει σχηματίσει ο παίκτης μας εκτός των έξι τελευταίων θέσεων του.
F2	Ο αριθμός από τις "πόρτες" που έχει σχηματίσει ο παίκτης μας εντός των έξι τελευταίων θέσεων του.
F3	Ο αριθμός από τις "πόρτες" που έχει σχηματίσει ο αντίπαλος εντός των έξι τελευταίων θέσεων του.
F4	Πόσα δικά μας πιόνια έχει αιχμαλωτίσει ο αντίπαλος.
F5	Η διαφορά της σχετικής θέσης του παίκτη μας με την αντίστοιχη σχετική θέση του αντιπάλου. (Ο ορισμός δίνεται παρακάτω στο κείμενο)
F6	Με πόσες "ζαριές" μπορεί να αιχμαλωτίσει κάποιο από τα πιόνια μας ο αντίπαλος.
F7	Τα πιόνια του παίκτη μας τα οποία είναι μέσα στις έξι πρώτες θέσεις του.
F8	Τα πιόνια του αντιπάλου τα οποία είναι μέσα στις έξι πρώτες θέσεις του.

Σχήμα 4.2 Χαρακτηριστικά μιας κατάστασης για το παιχνίδι backgammon.

Όνομα	Περιγραφή πρώτου χαρακτηριστικού	Περιγραφή δεύτερου χαρακτηριστικού
F9	Ο αριθμός από τις "πόρτες" που έχει σχηματίσει ο παίκτης μας εκτός των έξι τελευταίων θέσεων του. (F1)	Με πόσες ζαριές μπορεί να αιχμαλωτίσει κάποιο από τα πιόνια μας ο αντίπαλος. (F6)
F10	Πόσα δικά μας πιόνια έχει αιχμαλωτίσει ο αντίπαλος. (F4)	Η Διαφορά της σχετικής θέσης του παίκτη μας με την αντίστοιχη σχετική θέση του αντιπάλου. (F5)
F11	Πόσα δικά μας πιόνια έχει αιχμαλωτίσει ο αντίπαλος. (F4)	Ο αριθμός από τις "πόρτες" που έχει σχηματίσει ο αντίπαλος εντός των έξι τελευταίων θέσεων του. (F3)
F12	Τα πιόνια του αντιπάλου τα οποία είναι μέσα στις έξι πρώτες θέσεις του. (F8)	Ο αριθμός από τις "πόρτες" που έχει σχηματίσει ο παίκτης μας εντός των έξι τελευταίων θέσεων του. (F2)
F13	Τα πιόνια του παίκτη μας τα οποία είναι μέσα στις έξι πρώτες θέσεις του. (F7)	Ο αριθμός από τις "πόρτες" που έχει σχηματίσει ο αντίπαλος εντός των έξι τελευταίων θέσεων του. (F2)

Σχήμα 4.3 Συνδυαστικά χαρακτηριστικά μιας κατάστασης για το παιχνίδι Backgammon.

Όπως έχουμε προαναφέρει, οι δύο παίκτες κινούνται αντίθετα ο ένας από τον άλλον. Σαν πρώτες θέσεις για τον καθένα, σύμφωνα πάντα με την φορά που έχει μέσα στο ταμπλό του παιχνιδιού, θεωρούμε τις πιο απομακρυσμένες από την τελευταία θέση την οποία μπορεί να τοποθετήσει κάποιο πιόνι του πριν αρχίσει την διαδικασία της αφαίρεσης αυτών από το ταμπλό. Οι θέσεις αυτές είναι ταυτόχρονα και οι τελευταίες

θέσεις για τον αντίπαλό του. Με τον όρο "πόρτες", που αναφέρουμε στους πίνακες με τα χαρακτηριστικά, εννοούμε τις θέσεις στις οποίες κάθε παίκτης έχει πάνω από ένα πιόνι δικό του. Τα πιόνια τα οποία βρίσκονται σε μία θέση όπου έχει σχηματιστεί "πόρτα" είναι προστατευμένα από τον κίνδυνο αιχμαλώτισης και πάνω σε αυτές δεν μπορεί να τοποθετηθεί κάποιο από αυτά του αντιπάλου. Σαν σχετική θέση του κάθε παίκτη ορίζουμε το άθροισμα των γινομένων από τα πιόνια που έχει σε κάθε θέση του ταμπλό επί την απόσταση της θέσης αυτής από την τελευταία θέση του. Η απόσταση στα πιόνια τα οποία έχουν αιχμαλωτιστεί από τον αντίπαλο είναι κατά μία μονάδα αυξημένη από την μεγαλύτερη δυνατή πραγματική απόσταση που μπορεί να έχει κάποιο ενεργό πιόνι του. Οι "ζαριές" που υπολογίζουμε σε ένα από τα χαρακτηριστικά του Σχήματος 4.2 είναι ο αριθμός των διαφορετικών συνδυασμών από τις τιμές των δύο ζαριών, σύμφωνα με τους οποίους προκύπτει κάποια νόμιμη κίνηση για τον αντίπαλο που αν την ακολουθήσει αιχμαλωτίζει κάποιο πιόνι του παίκτη μας.

4.4 Διαδικασία μάθησης

Η μέθοδος της ενισχυτικής μάθησης στηρίζεται πάνω στην λογική της ανταμοιβής, αλλά και σε αυτήν της "δοκιμής-σφάλματος" (trial and error). Ένας πράκτορας μέσα σε ένα περιβάλλον άγνωστο γι' αυτόν, δοκιμάζει διάφορες ενέργειες σε διάφορες καταστάσεις, χωρίς να γνωρίζει τον αντίκτυπό τους σε ότι αφορά στο κέρδος του. Η τελική ανταμοιβή της νίκης ή της ήττας είναι το μοναδικό στοιχείο το οποίο μπορεί να εκμεταλλευτεί για να μπορέσει να βελτιώσει την συμπεριφορά του μέσα σε αυτό το περιβάλλον και να μεγιστοποιήσει τα κέρδη του.

Αυτό που παίζει καθοριστικό ρόλο στην εκπαίδευση ενός πράκτορα είναι να "δοκιμάσει" όσο το δυνατόν περισσότερες διαφορετικές ανταμοιβές σε διάφορες καταστάσεις. Πρέπει δηλαδή να δοκιμάσει ενέργειες, στην περίπτωση του Backgammon, οι οποίες θα του αποφέρουν και την νίκη αλλά και την ήττα. Είναι εξίσου σημαντικό εκτός από θετική ανταμοιβή να λάβει και αρνητική ανταμοιβή, για να μπορέσει, μέσα από την διαδικασία της εκπαίδευσής του να καταλήξει σε μία πιο αποδοτική γι' αυτόν στρατηγική.

Αρχικά κατασκευάσαμε διάφορα μοντέλα αντιπάλων για τον σκοπό της εκπαίδευσης. Δημιουργήσαμε γι' αυτούς συναρτήσεις αξιολόγησης με κάποια βασικά

χαρακτηριστικά και δώσαμε τέτοιες τιμές στα βάρη τους έτσι ώστε η συμπεριφορά τους να είναι πιο "επιθετική" ή πιο "αμυντική" κατά περίπτωση. Στα βάρη των χαρακτηριστικών της συνάρτηση αξιολόγησης του πράκτορά μας δόθηκαν αρχικά τυχαίες τιμές. Σε κάποια παιχνίδια που πραγματοποιήθηκαν, πριν γίνει οποιαδήποτε εκπαίδευση, απέναντι στους διάφορους αυτούς αντιπάλους ο πράκτοράς μας δεν είχε καμία νίκη. Αντιθέτως, ύστερα από κάποιο αριθμό παιχνιδιών όπου χρησιμοποιήθηκε ενισχυτική μάθηση για την εκπαίδευσή του, ο πράκτοράς μας σημείωνε πολύ περισσότερες νίκες από ότι οι αντίπαλοί του. Τα σημάδια της μάθησης ήταν εμφανή μετά από την διαδικασία αυτή, αλλά διαπιστώσαμε ότι ο πράκτοράς μας έφτανε γρήγορα σε ένα μέγιστο σε ότι αφορά στην ικανότητά του και από εκεί και πέρα σταματούσε η βελτίωσή του. Ο λόγος που συνέβαινε αυτό ήταν διότι μετά από κάποιο σημείο οι αντιδράσεις των αντιπάλων του δεν τον οδηγούσαν εύκολα σε "λάθος" ενέργειες ή σε νέες καταστάσεις μέσα από τις οποίες θα μπορούσε να μάθει και να βελτιώσει τις ικανότητές του. Η ανταμοιβή δηλαδή που εκλάμβανε ήταν τις περισσότερες φορές θετική, ενώ οι αρνητικές ανταμοιβές ήταν πολύ λιγότερες για να του επιτρέψουν να βελτιωθεί.

Για τον λόγο λοιπόν αυτό, έπρεπε να βρεθεί ένας μηχανισμός τέτοιος ώστε από την μία να εκπαιδεύεται ο πράκτοράς μας και από την άλλη να συνεχίζει να δέχεται τόσο θετικές όσο και αρνητικές ανταμοιβές που θα τον βοηθήσουν στην συνέχιση της μάθησής του. Έτσι λοιπόν ακολουθώντας την παραπάνω λογική, αρχικά σαν αντίπαλο απέναντι στον πράκτορά μας τοποθετήσαμε ένα ακριβές αντίγραφό του. Τα χαρακτηριστικά της συνάρτησης αξιολόγησης του αντιπάλου ήταν τα ίδια καθώς και οι τιμές των βαρών τους, οι οποίες όμως παρέμεναν αμετάβλητες. Η σύγκριση αρχικά μεταξύ τους δεν θα είχε νόημα αν λάβουμε βέβαια υπόψη μας και το στοιχείο της τύχης. Έπειτα όμως από την εφαρμογή ενισχυτικής μάθησης κατά την διεξαγωγή αρκετών εκατοντάδων παρτίδων, και την σύγκλιση των νέων τιμών στα βάρη των χαρακτηριστικών του παίκτη μας η βελτίωση φάνηκε από τον αυξημένο αριθμό από νίκες που κατάφερνε ο πράκτοράς μας. Η επόμενη κίνησή μας ήταν να αντικαταστήσουμε τον προηγούμενο αντίπαλο, δηλαδή την συνάρτηση αξιολόγησής του, με την βελτιωμένη έκδοση στην οποία καταλήξαμε μετά την ολοκλήρωση της εκπαίδευσης.

Την παραπάνω διαδικασία την επαναλάβαμε αρκετές φορές και διαπιστώναμε συνεχώς την βελτίωση του πράκτορά μας. Φέρνοντάς τον δηλαδή κάθε φορά αντιμέτωπο με τον βελτιωμένο εαυτό του και επιχειρώντας την εκπαίδευση απέναντι σε αυτόν, καταφέραμε, ο πράκτοράς μας, να δέχεται συνεχώς τόσο θετικές όσο και αρνητικές ανταμοιβές καλύπτοντας ένα ευρύ φάσμα καταστάσεων.

Μετά από αρκετές αντικαταστάσεις του αντιπάλου και επανεκπαίδευσης του πράκτορά μας απέναντι σε αυτόν διαπιστώσαμε ότι ο ρυθμός βελτίωσής του μειωνόταν συνεχώς έως ότου δεν φαινόταν πια κάποια σημαντική πρόοδος. Στο σημείο αυτό κρατήσαμε τις τιμές των βαρών, για τα χαρακτηριστικά της συνάρτησής μας, τις οποίες χρησιμοποιούμε πλέον οριστικά, χωρίς μεταβολές, στο παιχνίδι Backgammon.

Αυτό το "φράγμα" που προέκυψε στην βελτίωση του πράκτορά μας με την παραπάνω μεθοδολογία, οφείλεται κατά κύριο λόγο στην "δυναμική" που πηγάζει από το σύνολο των χαρακτηριστικών που χρησιμοποιούμε στην συνάρτηση αξιολόγησης, καθώς και στην συγκεκριμένη μέθοδο μάθησης που χρησιμοποιήσαμε. Γι' αυτόν τον λόγο, σε επόμενο κεφάλαιο, οι παράμετροι αυτοί προτείνονται για βελτίωση σε μια πιθανή επανεξέταση της εργασίας αυτής.

4.5 Μέθοδος ενισχυτικής μάθησης

Για τις ανάγκες της μάθησης μιας καλής στρατηγικής για τον πράκτορά μας χρησιμοποιήσαμε πρακτικές από τον χώρο της ενισχυτικής μάθησης. Συγκεκριμένα υλοποιήσαμε την μέθοδο χρονικών διαφορών (TD). Κατά την διάρκεια της εκπαίδευσης η μέθοδος αυτή χρησιμοποιεί την επιστρεφόμενη τιμή μετά από την αναζήτηση του αλγόριθμου ExpectiMiniMax και ενημερώνει τα βάρη της συνάρτησης αξιολόγησης σύμφωνα με την παρακάτω εξίσωση. Στον συντελεστή α, που όπως έχουμε προαναφέρει λειτουργεί σαν ρυθμός μάθησης, δώσαμε στην τιμή 0,01.

$$w_i \leftarrow w_i + \alpha f_i(s)(r + \quad Eval(s') - Eval(s))$$

Κεφάλαιο 5

Θέματα υλοποίησης

5.1 Γενικές πληροφορίες

Ο πράκτοράς μας υλοποιήθηκε και μπορεί να εκτελεστεί σε περιβάλλον Windows. Το προγραμματιστικό περιβάλλον στο οποίο αναπτύχθηκε είναι το Visual Studio [8] της Microsoft, ενώ η γλώσσα προγραμματισμού την οποία χρησιμοποιήσαμε είναι η C++ [7], η οποία μας έδωσε και την ευελιξία απέναντι στις απαιτήσεις σε πολυπλοκότητα και ταχύτητα που είχε η εφαρμογή αυτή. Για το γραφικό περιβάλλον χρησιμοποιήθηκαν οι κατάλληλες συναρτήσεις από την βιβλιοθήκη του .Net [9], η εγκατάσταση του οποίου φυσικά είναι και προαπαιτούμενο για να μπορεί να λειτουργήσει η εφαρμογή μας.

5.2 Δυνατότητες εφαρμογής και γραφικό περιβάλλον

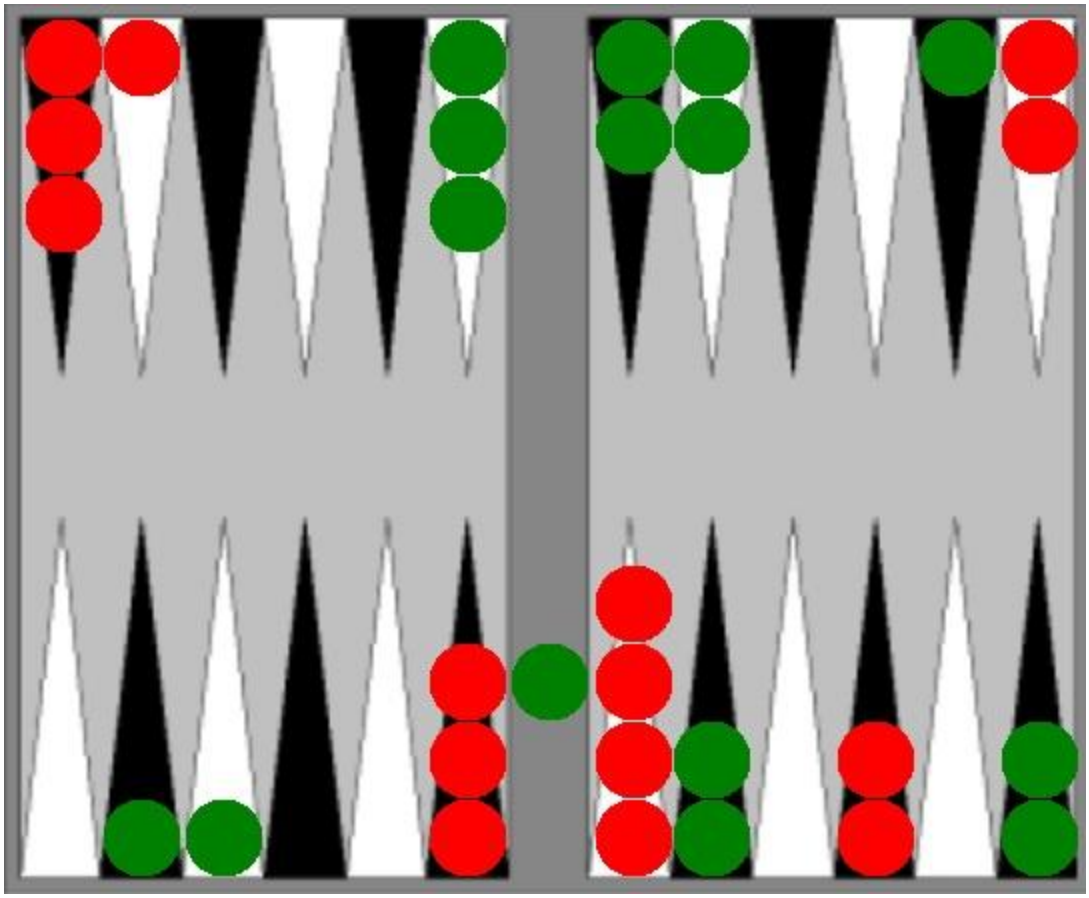
Κατά τη διάρκεια της υλοποίησης της εφαρμογής μας εκτός από τις βασικές συναρτήσεις, ο οποίες ήταν απαραίτητες για την ορθή λειτουργία του πράκτορά μας, υλοποιήθηκαν και αρκετές βοηθητικές συναρτήσεις για έλεγχο. Οι συναρτήσεις αυτές για κάποιον ο οποίος θέλει να μελετήσει σε μεγαλύτερο βάθος ή να συμβουλευτεί από την απόκριση του πράκτορά μας όσον αφορά το παιχνίδι Backgammon θα μπορούσαν να του φανούν πολύ χρήσιμες.

Το γραφικό περιβάλλον (Σχήμα 5.1), εκτός φυσικά από την αντιμετώπιση του πράκτορά μας σε μια κανονική παρτίδα Backgammon, δίνει στον χρήστη και τις παρακάτω δυνατότητες:

- Δημιουργία οποιασδήποτε κατάστασης και ορισμός της σαν αρχική για την διαδικασία της μάθησης με δυνατότητα καθορισμού αρχικών τιμών έναρξης

στα βάρη για τα χαρακτηριστικά της συνάρτησης αξιολόγησης.

- Εύρεση της καλύτερης προτεινόμενης κίνησης για οποιαδήποτε μεμονωμένη κατάσταση ορισμένη από τον χρήστη.
- Εύρεση όλων των δυνατών-νόμιμων κινήσεων για οποιαδήποτε μεμονωμένη κατάσταση ορισμένη από τον χρήστη.
- Εύρεση όλων των δυνατών συνδυασμών για τα δυο ζάρια σύμφωνα με τους οποίους προκύπτει κάποια νόμιμη κίνηση για τον αντίπαλο, που έχει σαν αποτέλεσμα την αιχμαλώτιση κάποιου από τα πιόνια του παίκτη μας.



Σχήμα 5.1 Γραφικό περιβάλλον της υλοποίησης του παιχνιδιού Backgammon.

5.3 Υλοποίηση πράκτορα

Το σύνολο της πληροφορίας που είναι απαραίτητο για την απεικόνιση οποιουδήποτε

στιγμιότυπου της σκακιέρας του παιχνιδιού την αποθηκεύουμε μέσα σε έναν μόνο πίνακα. Ουσιαστικά στον πίνακα αυτό αποθηκεύεται η θέση για κάθε πιόνι και των δύο παικτών, τα πιόνια τα οποία είναι αιχμαλωτισμένα από τον αντίπαλο για τις δύο πλευρές, τα πιόνια τα οποία ο κάθε παίκτης έχει θέσει νόμιμα εκτός παιχνιδιού αλλά και οι τιμές από τα δύο ζάρια.

Για την υλοποίηση συγκεκριμένων λειτουργιών του πράκτορα χρησιμοποιήσαμε βοηθητικούς πίνακες με την ίδια δομή όπως αυτή του βασικού πίνακα. Η διπλή κίνηση κάθε παίκτη, μία για το ένα ζάρι και μία για το δεύτερο, είναι μία από τις περιπτώσεις στις οποίες χρησιμοποιούνται αυτοί οι βοηθητικοί πίνακες για τις ενδιάμεσες καταστάσεις οι οποίες προκύπτουν. Τέτοιους πίνακες χρησιμοποιούμε και για την εύρεση της τιμής κάποιων από τα χαρακτηριστικά της συνάρτησης αξιολόγησης.

Όπως έχουμε προαναφέρει κατά την διάρκεια της αναζήτησης αποθηκεύονται μόνο οι απαραίτητες καταστάσεις και με την βοήθεια της αναδρομής ουσιαστικά διατρέχουμε το δέντρο αναζήτησης κερδίζοντας σε χρόνο αλλά και σε ταχύτητα. Η συνάρτηση η οποία είναι υπεύθυνη για την αλλαγή των καταστάσεων είναι η Backgammonsim. Τα βασικά ορίσματα της συνάρτησης αυτής είναι η εκάστοτε δομή στην οποία είναι αποθηκευμένη η κατάσταση την οποία θέλουμε να μεταβάλουμε καθώς και η νόμιμη κίνηση την οποία θέλουμε να εκτελέσουμε.

Κατά την διάρκεια της λειτουργίας του πράκτορά μας, όλες οι δομές στις οποίες αποθηκεύονται προσωρινά δεδομένα, για την εκάστοτε περίπτωση, δεν αποδεσμεύονται από την μνήμη για να δημιουργηθούν ξανά όταν αυτό γίνει απαραίτητο. Αντίθετα γίνεται δέσμευση της απαιτούμενης μνήμης από την αρχή της λειτουργίας του πράκτορά μας η οποία αρχικοποιείται κάθε φορά όταν καταστεί απαραίτητο. Με τον τρόπο αυτό κερδίζουμε τον χρόνο που θα απαιτούνταν κάθε φορά για την δέσμευση και αποδέσμευση της απαραίτητης μνήμης. Ένας από τους πίνακες στους οποίους εφαρμόζουμε την παραπάνω λογική είναι αυτός στον οποίο αποθηκεύονται όλες οι νόμιμες κινήσεις για κάποια κατάσταση του παιχνιδιού.

Για λόγους φορητότητας, οι τελικές τιμές για τα βάρη των χαρακτηριστικών της συνάρτησης αξιολόγησης, στις οποίες καταλήξαμε μετά από όλη την διαδικασία της εκπαίδευσης του πράκτορά μας αποθηκεύονται σαν σταθερές τιμές πλέον μέσα στον κώδικα της εφαρμογής. Κατά την διάρκεια της μάθησης όμως οι τιμές αυτές

αποθηκεύονταν σε ένα εξωτερικό αρχείο για λόγους καλύτερης παρακολούθησης και αξιολόγησής τους.

5.4 Συνάρτηση εύρεσης νόμιμων κινήσεων

Μία από τις σημαντικότερες συναρτήσεις, η οποία αποτελεί ένα μεγάλο μέρος του "πυρήνα" του παιχνιδιού Backgammon είναι η συνάρτηση εύρεσης των νόμιμων κινήσεων. Η συνάρτηση αυτή στην εφαρμογή μας ονομάζεται Backgammonactions και τα βασικότερα ορίσματά της είναι ο πίνακας με την κατάσταση για την οποία θέλουμε να βρούμε τις νόμιμες κινήσεις καθώς και ο πίνακας στο οποίο αποθηκεύονται οι κινήσεις αυτές για να χρησιμοποιηθούν μετά στην συνέχεια της αναζήτησης.

Ο λόγος για τον οποίο γίνεται ιδιαίτερη αναφορά στην συνάρτηση αυτή είναι γιατί οι κανόνες του παιχνιδιού αλλάζουν, ως προς τις νόμιμες κινήσεις τις οποίες έχει στη διάθεσή του ο κάθε παίκτης, ανάλογα με την τρέχουσα κατάσταση στην οποία βρίσκεται. Οι διαφοροποιήσεις των κανόνων εύρεσης των νόμιμων κινήσεων παρουσιάζονται στις παρακάτω ομάδες καταστάσεων :

- Καταστάσεις στις οποίες ένα ή περισσότερα από τα πιόνια του παίκτη είναι αιχμαλωτισμένα από τον αντίπαλο.
- Καταστάσεις στις οποίες ο παίκτης έχει τοποθετήσει όλα τα ενεργά του πιόνια μέσα στις έξι τελευταίες θέσεις του.
- Καταστάσεις στις οποίες κανένα πιόνι του παίκτη δεν είναι αιχμαλωτισμένο από τον αντίπαλο και υπάρχει τουλάχιστον ένα πιόνι δικό του εκτός των έξι τελευταίων θέσεων.

Στην πρώτη περίπτωση, όπου υπάρχουν αιχμαλωτισμένα πιόνια, ο παίκτης είναι υποχρεωμένος να τα επανατοποθετήσει μέσα στην σκακιέρα πριν κάνει οποιαδήποτε άλλη κίνηση με κάποιο από τα υπόλοιπα πιόνια του. Στις καταστάσεις αυτές είναι και η μοναδική περίπτωση στην οποία μπορεί να "επανατροφοδοτηθεί" η πρώτη θέση της σκακιέρας για τον αντίστοιχο παίκτη.

Στην δεύτερη περίπτωση, ο παίκτης έχει δικαίωμα να θέσει εκτός σκακιέρας κάποιο από τα πιόνια του, εφόσον βέβαια έχει αυτήν την δυνατότητα. Σε αυτές τις καταστάσεις οποιοδήποτε πιόνι, εφόσον δεν παραβιάζει τους κανόνες του παιχνιδιού,

θα μπορούσε να χρησιμοποιηθεί για την κίνηση όλων των συνδυασμών από τα δύο ζάρια.

Στην τρίτη περίπτωση περιλαμβάνεται το μεγαλύτερο πλήθος των καταστάσεων του παιχνιδιού και εδώ ανήκουν όλες οι υπόλοιπες καταστάσεις εκτός των δύο παραπάνω περιπτώσεων που αναφέραμε. Οι κανόνες σε αυτήν την περίπτωση είναι συγκεκριμένοι και έχουν περιγραφεί σε προηγούμενο κεφάλαιο.

Η μεγάλη διαφοροποίηση στους κανόνες του παιχνιδιού ανάμεσα σε διάφορες καταστάσεις μας οδήγησε στην τμηματική υλοποίηση της συνάρτησης εύρεσης νόμιμων κινήσεων. Έτσι λοιπόν δημιουργήσαμε τρεις ξεχωριστές συναρτήσεις, μία για κάθε περίπτωση που αναφέρεται παραπάνω.

Το πρόβλημα περιπλέκεται ακόμα περισσότερο από το γεγονός ότι οποιαδήποτε κίνηση στο παιχνίδι Backgammon, εκτός ελαχίστων περιπτώσεων, αποτελείται ουσιαστικά από δύο μικρότερες κινήσεις. Οι δευτερεύουσες αυτές κινήσεις ουσιαστικά υφίστανται λόγω της ύπαρξης των δύο ζαριών και της υποχρέωσης του κάθε παίκτη να χρησιμοποιήσει τις τιμές και των δύο για να μετακινήσει ένα η περισσότερα από τα πιόνια που έχει στην διάθεσή του. Πρακτικά αυτό μπορεί να σημαίνει ότι η εκτέλεση του ήμισυ μιας ολοκληρωμένης κίνησης, η οποία μπορεί να υπακούει στους κανόνες μίας από τις παραπάνω περιπτώσεις, μπορεί να οδηγήσει σε μία ενδιάμεση κατάσταση η οποία ανήκει σε διαφορετική κατηγορία με διαφορετικούς κανόνες.

Για την αντιμετώπιση του παραπάνω προβλήματος δημιουργήσαμε δομές οι οποίες είναι ίδιες με αυτήν την οποία περιγράψαμε παραπάνω και αποθηκεύουν όλη την πληροφορία για την σωστή απεικόνιση μιας κατάστασης του παιχνιδιού. Σε κάθε περίπτωση όπου ερχόμαστε αντιμέτωποι με το παραπάνω πρόβλημα, η ενδιάμεση κατάσταση η οποία δημιουργείται ανάμεσα στις δύο δευτερεύουσες κινήσεις, αποθηκεύεται σε μία τέτοια βοηθητική δομή η οποία "περνάει" σαν όρισμα στην κατάλληλη υπο-συνάρτηση της γενικότερης συνάρτησης εύρεσης των νόμιμων κινήσεων.

Κεφάλαιο 6

Αποτελέσματα

6.1 Διαδικασία αναζήτησης

Ο πράκτορας που δημιουργήσαμε επιτυγχάνει βάθος αναζήτησης έως και πέντε στρώσεις, ενώ ο χρόνος που απαιτείται για να επιστρέψει κάποιο αποτέλεσμα ο αλγόριθμος ExcpectiMiniMax κυμαίνεται από ελάχιστα δέκατα του δευτερολέπτου έως και είκοσι δευτερόλεπτα. Το βάθος αναζήτησης αλλά και ο χρόνος απόκρισης θεωρούνται ικανοποιητικά αν λάβουμε υπόψη μας δύο πολύ σημαντικούς παράγοντες.

Ο πρώτος είναι ο παράγοντας διακλάδωσης του δέντρου αναζήτησης του παιχνιδιού, ο οποίος μπορεί να φτάσει και πάνω από 200 για τους κόμβους max και min, ενώ για τους κόμβους τύχης είναι 21. Ο δεύτερος είναι η μεγάλη πολυπλοκότητα που αντιμετωπίσαμε, με αντίκτυπο φυσικά στην ταχύτητα, όσον αφορά στην συνάρτηση διαδόχων κόμβων αλλά και στην εύρεση των τιμών για τα χαρακτηριστικά της συνάρτησης αξιολόγησης.

6.2 Βασικά και Συνδυαστικά Χαρακτηριστικά

Όπως έχουμε προαναφέρει εκτός από τα βασικά χαρακτηριστικά τα οποία χρησιμοποιήσαμε για την δημιουργία της συνάρτησης αξιολόγησης, φτιάξαμε και ένα σετ από συνδυαστικά χαρακτηριστικά. Επομένως έχουμε δύο ομάδες χαρακτηριστικών, η πρώτη με τα βασικά χαρακτηριστικά μόνο και η δεύτερη με τα βασικά και τα συνδυαστικά χαρακτηριστικά μαζί. Για την σύγκριση των δύο αυτών ομάδων έγινε εκπαίδευση του πράκτορά μας πρώτα με την πρώτη ομάδα (βασικά χαρακτηριστικά) σε ένα αριθμό 3000 παιχνιδιών και στην συνέχεια έγινε εκπαίδευση του πράκτορά μας με την δεύτερη ομάδα (σύνολο χαρακτηριστικών) στον ίδιο αριθμό παιχνιδιών, πάντα απέναντι στον ίδιο σταθερό αντίπαλο.

Μετά την εκπαίδευση και με τις δύο ομάδες χαρακτηριστικών έγινε ένα τουρνουά από 500 παιχνίδια για να συγκρίνουμε την απόδοση τους. Τα αποτελέσματα των παιχνιδιών αυτών παρατίθενται στο Σχήμα 6.1 όπου είναι εμφανής η υπεροχή της δεύτερης ομάδας η οποία περιέχει το σύνολο των χαρακτηριστικών τα οποία δημιουργήσαμε.

	Ομάδα Α **Βασικά** **(8 χαρακτηριστικά)**	**Ομάδα Β** **Βασικά + Συνδυαστικά** **(13 χαρακτηριστικά)**
Νίκες	183	317

Σχήμα 6.1 Αποτελέσματα από τη σύγκριση των δύο ομάδων χαρακτηριστικών.

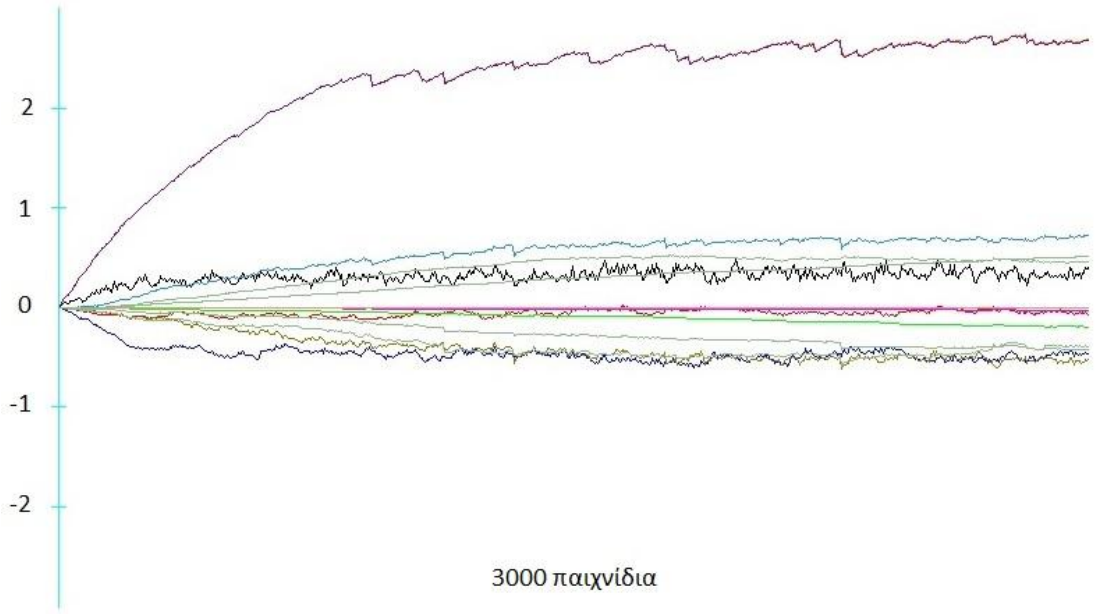
6.3 Διαδικασία εκμάθησης

Μετά την σύγκριση των δύο ομάδων από χαρακτηριστικά συνεχίσαμε την εκπαίδευση του πράκτορά μας με την δεύτερη ομάδα, αυτήν δηλαδή με το σύνολο των χαρακτηριστικών, αφού ήταν εμφανής η υπεροχή της απέναντι στην πρώτη. Όπως έχουμε προαναφέρει κατά την διαδικασία της μάθησης ως αντίπαλός του πράκτορά μας οριζόταν πάντα κάποιο αντίγραφο του εαυτού του. Κατά την έναρξη της διαδικασίας τα βάρη της συνάρτησης αξιολόγησης ήταν τα ίδια και για τους δύο παίκτες. Στην πορεία όμως της εκπαίδευσης τα βάρη του πράκτορά μας μεταβάλλονταν με την βοήθεια της μεθόδου ενισχυτικής μάθησης χρονικών διαφορών TD, σε αντίθεση με τα βάρη του αντιπάλου του τα οποία παρέμεναν σταθερά.

Μετά την διαδικασία ενός κύκλου μάθησης 3000 παιχνιδιών, γινόταν ένα τουρνουά 100 παιχνιδιών ανάμεσα στον τροποποιημένο πλέον “εαυτό” του πράκτορα μας και στον αντίπαλό του για να διαπιστώσουμε ότι υπήρξε όντως βελτίωση. Στην συνέχεια τα τροποποιημένα αυτά βάρη της συνάρτησης αξιολόγησης, οριζόταν και σαν βάρη για τον αντίπαλο και η διαδικασία άρχιζε από την αρχή, με στόχο την αύξηση της δυσκολίας του αντιπάλου και την συνεχή βελτίωση του πράκτορά μας. Μετά το τέλος κάθε κύκλου μάθησης η βελτίωση του πράκτορά μας, όσον αφορά τις νίκες τις οποίες κατάφερνε, μειωνόταν με αποτέλεσμα μετά από κάποιο σημείο σχεδόν να μηδενιστεί.

Συνολικά πραγματοποιήθηκαν 7 κύκλοι μάθησης και οι τιμές στις οποίες συνέκλιναν τα βάρη από τον τελευταίο κύκλο μάθησης είναι και αυτά τα οποία χρησιμοποιούμε πλέον οριστικά στον πράκτορά μας.

Η απεικόνιση από την σύγκλιση των τιμών στα βάρη του συνόλου των χαρακτηριστικών της δεύτερης ομάδας, κατά την διάρκεια της εκπαίδευσης του πράκτορά μας, στον έβδομο κύκλο μάθησής του, φαίνεται στο Σχήμα 6.2.



Σχήμα 6.2 Σύγκλιση των τιμών στα βάρη της Β ομάδας σε 3000 παιχνίδια κατά τον έβδομο κύκλο μάθησής του πράκτορά μας.

Στο Σχήμα 6.3 απεικονίζονται οι τιμές από όλες τις τιμές των βαρών στις οποίες καταλήξαμε μετά από την ολοκλήρωση του τελευταίου κύκλου μάθησης του πράκτορά μας.

Όνομα	**Τιμή βάρους**
W1 (Χαρακτηριστικό F1)	-0,5381233
W2 (Χαρακτηριστικό F2)	0,4312553
W3 (Χαρακτηριστικό F3)	-0,1135148
W4 (Χαρακτηριστικό F4)	-0,2818067
W5 (Χαρακτηριστικό F5)	2,449752
W6 (Χαρακτηριστικό F6)	-0,5527733

W7 (Χαρακτηριστικό F7)	0,4155801
W8 (Χαρακτηριστικό F8)	0,525891
W9 (Χαρακτηριστικό F9)	-0,0207155
W10 (Χαρακτηριστικό F10)	0,467243
W11 (Χαρακτηριστικό F11)	-0,4842022
W12 (Χαρακτηριστικό F12)	-0,5778358
W13 (Χαρακτηριστικό F13)	0,0340492

Σχήμα 6.3 Απεικόνιση των τιμών των τελικών βαρών μετά την ολοκλήρωση της εκπαίδευσης του πράκτορά μας.

6.4 Σύγκριση με φυσικούς παίκτες

Μετά την επιλογή της ομάδας των χαρακτηριστικών και την εκπαίδευση του πράκτορά μας πάνω σε αυτά, έγινε σύγκριση του απέναντι σε φυσικούς παίκτες. Η επιλογή των παικτών αυτών προσπαθήσαμε να γίνει έτσι ώστε να έχουν διαφορετική εμπειρία όσον αφορά στο παιχνίδι Backgammon και να εξετάσουμε με τον τρόπο αυτό την απόδοση του πράκτορά μας απέναντι σε διαφορετικής δυσκολίας αντιπάλους. Με κάθε τέτοιο αντίπαλο έγινε ένα τουρνουά 50 παιχνιδιών όπου τα αποτελέσματα αυτών παρατίθεται στο Σχήμα 6.4.

	Νίκες	**Ήττες**
Παίκτης1	7	43
Παίκτης2	13	37
Παίκτης3	22	28
Παίκτης4	26	24
Παίκτης5	39	11

Σχήμα 6.4 Αποτελέσματα παιχνιδιών με φυσικούς παίκτες.

Στον παραπάνω πίνακα η ταξινόμηση των παικτών γίνεται ανάλογα με την εμπειρία τους πάνω στο παιχνίδι. Στην δεύτερη στήλη αναγράφονται οι νίκες για τον παίκτη-αντίπαλο και στην τρίτη στήλη αναγράφονται οι ήττες του, δηλαδή οι νίκες τις οποίες

κατάφερε ο πράκτοράς μας. Ο Παίκτης1 είχε πολύ χαμηλή εμπειρία με το παιχνίδι γι' αυτό και η διαφορά στις νίκες είναι τόσο μεγάλη. Οι Παίκτης2, Παίκτης3 και Παίκτης4 είχαν μια σχετικά μέση εμπειρία στο παιχνίδι και όπως φαίνεται από τα αποτελέσματα του πίνακα ο πράκτοράς μας κατάφερε μια ικανοποιητική αναλογία σε νίκες αντιμετωπίζοντάς τους. Ο Παίκτης5 είναι ένας παίκτης ο οποίος είχε αρκετά μεγάλη εμπειρία πάνω στο παιχνίδι και αυτό φάνηκε και από την μεγάλη διαφορά σε νίκες που είχε απέναντι στον πράκτορά μας.

Σε κάθε περίπτωση, πριν την αξιολόγηση οποιουδήποτε αποτελέσματος, θα πρέπει να λαμβάνουμε πάντα υπόψη μας το στοιχείο της τύχης, το οποίο υπάρχει στο παιχνίδι Backgammon και είναι τα ζάρια. Είναι ένας απρόβλεπτος παράγοντας ο οποίος επηρεάζει άμεσα την εξέλιξη του παιχνιδιού όσον αφορά την στρατηγική που πρέπει να ακολουθήσει ο κάθε παίκτης και θα πρέπει πάντα να συνυπολογίζεται πριν την εξαγωγή οποιουδήποτε συμπεράσματος.

6.5 Σύγκριση με άλλους πράκτορες

Στην προσπάθειά μας να αξιολογήσουμε την απόδοση των τεχνικών που υιοθετήσαμε για την υλοποίηση του πράκτορά μας, εκτός από την σύγκριση με κάποιους φυσικούς παίκτες, έγινε σύγκριση και με κάποια προγράμματα τα οποία έχουν υλοποιηθεί για να παίζουν το παιχνίδι Backgammon. Οι πράκτορες αυτοί οι οποίοι χρησιμοποιήθηκαν για να συγκρίνουμε την απόδοση τους σε σχέση με την δική μας υλοποίηση ήταν οι Snowie [13] και Jellyfish [12]. Οι κατασκευαστές και των δύο αυτών εφαρμογών υποστηρίζουν ότι για την υλοποίηση της εφαρμογής τους στηρίχθηκαν πάνω στην εργασία TD-Gammon του Gerry Tesauro [4] την οποία αναφέραμε σε προηγούμενο κεφάλαιο.

Στην προσπάθειά μας γι' αυτήν τη σύγκριση δεν έγινε εφικτό να βρούμε κάποιο αυτόματο τρόπο για την διεξαγωγή των παρτίδων. Δεν μπορέσαμε δηλαδή να βρούμε κάποια υλοποίηση των προγραμμάτων αυτών με την οποία θα μπορούσε ο πράκτοράς μας να επικοινωνήσει χωρίς την δική μας παρέμβαση. Γι' αυτόν το λόγο η διαδικασία αυτή της σύγκρισης έγινε με χειροκίνητο τρόπο. Το παιχνίδι δηλαδή διεξαγόταν και στα δύο περιβάλλοντα με την κίνηση του κάθε πράκτορα να μεταφέρεται στον αντίπαλό του από εμάς.

Η σύγκριση με τους πράκτορες Snowie και Jellyfish έγινε σε ένα τουρνουά των 50

παρτίδων με τον κάθε έναν, τα αποτελέσματα των οποίων παρατίθεται στο Σχήμα 6.5.

	Νίκες	**Ήττες**
Snowie	31	19
Jellyfish	29	21

Σχήμα 6.5 Σύγκριση με άλλους πράκτορες.

Στην δεύτερη στήλη του παραπάνω πίνακα αναγράφονται οι νίκες που κατάφερε ο αντίστοιχος πράκτορας απέναντι στην δική μας υλοποίηση, ενώ στην τρίτη στήλη αναγράφονται οι ήττες γι' αυτόν, δηλαδή οι νίκες για τον δικό μας πράκτορα. Παρόλο που στα δύο σετ των 50 παιχνιδιών ο πράκτοράς μας κατάφερε λιγότερες νίκες απέναντι στα προγράμματα Snowie και Jellyfish, από τα παραπάνω στοιχεία φαίνεται ότι μπόρεσε να "σταθεί" αρκετά καλά απέναντι στα δύο αυτά προγράμματα λαμβάνοντας πάντα υπόψη ότι και οι δύο αυτοί αντίπαλοί του είναι εμπορικά προϊόντα, πράγμα που τα καθιστά άκρως ανταγωνιστικά.

Κεφάλαιο 7

Συμπεράσματα

7.1 Συμπεράσματα

Στα πλαίσια της εργασίας αυτής υλοποιήσαμε έναν πράκτορα για το παιχνίδι Backgammon. Ο πράκτορας ο οποίος υλοποιήσαμε επιτυγχάνει βάθος αναζήτησης πέντε στρώσεων και ο χρόνος απόκρισης του αλγόριθμου αναζήτησης είναι αρκετά ρεαλιστικός αφού κυμαίνεται από κάποια δέκατα του δευτερολέπτου μέχρι κάποια ελάχιστα δευτερόλεπτα. Οι επιδόσεις αυτές θεωρούνται ικανοποιητικές αν λάβουμε υπόψη μας την πολυπλοκότητα του παιχνιδιού Backgammon. Η πολυπλοκότητα αυτή οφείλεται κυρίως στην πολυμορφία των διάφορων καταστάσεων του παιχνιδιού, στην μεγάλη αλληλεπίδραση των κινήσεων των δύο παικτών αλλά και στην αλλαγή των κανόνων του παιχνιδιού ανάλογα με την εκάστοτε κατάσταση. Ένας πολύ σημαντικός παράγοντας ο οποίος επηρέασε επίσης την απόδοση της αναζήτησής μας είναι ο μεγάλος παράγοντας διακλάδωσης του δέντρου αναζήτησης, ο οποίος μπορεί να φτάσει και τις μερικές εκατοντάδες ανάλογα βέβαια με την αρχική κατάσταση του δέντρου σε έναν "κύκλο" αναζήτησης.

Η σύγκριση του πράκτορά μας, για την εκπαίδευση του οποίου χρησιμοποιήσαμε την μέθοδο χρονικών TD, με φυσικούς παίκτες διαφόρων επιπέδων εμπειρίας αλλά και με άλλους πράκτορες-προγράμματα όπως το Snowie και το Jellyfish, οι οποίοι έχουν υλοποιηθεί με άλλες μεθόδους, απέδειξε ότι μπορεί να σταθεί ισάξια απέναντι σε αρκετά καλούς παίκτες. Στο σημείο αυτό οφείλουμε να αναφέρουμε ότι ο πράκτοράς μας παρουσιάζει μια αδυναμία να αντιμετωπίσει παίκτες με αρκετά μεγάλη εμπειρία πάνω στο παιχνίδι Backgammon, γεγονός το οποίο οφείλεται κυρίως στην δυναμική της συνάρτησης αξιολόγησης την οποία υιοθετήσαμε αλλά και στην μέθοδο μάθησης την οποία χρησιμοποιήσαμε.

7.2 Μελλοντικές βελτιώσεις

Η συνάρτηση αξιολόγησης, όπως έχουμε επισημάνει και σε προηγούμενο κεφάλαιο, είναι μια πολύ σημαντική παράμετρος η οποία καθορίζει την συμπεριφορά ενός πράκτορα. Μία σημαντική λοιπόν βελτίωση θα μπορούσε να γίνει εμπλουτίζοντας την συνάρτηση αξιολόγησης που χρησιμοποιήσαμε στην υλοποίησή μας με περισσότερα και πιο "φιλοσοφημένα" χαρακτηριστικά.

Όσον αφορά την αναζήτηση θα μπορούσαν να χρησιμοποιηθούν και άλλες τεχνικές βελτιστοποίησης του αλγόριθμου MiniMax, εκτός από το κλάδεμα Άλφα-Βήτα. Όσον αφορά την μάθηση θα μπορούσαν να εξεταστούν εναλλακτικοί αλγόριθμοι μάθησης, όπως αυτοί των ελαχίστων τετραγώνων (LSTD) [5] και (LSPI) [6].

Τέλος, λόγω των υψηλών υπολογιστικών απαιτήσεων της εφαρμογής, θα μπορούσαν να χρησιμοποιηθούν τεχνικές παράλληλου προγραμματισμού για την πλήρη εκμετάλλευση των δυνατοτήτων των επεξεργαστών, πράγμα το οποίο θα είχε και άμεσο όφελος στην ταχύτητα απόκρισης της εφαρμογής.

Βιβλιογραφία

[1] Stuart Russel and Peter Norvig, "Artificial Intelligence: A Modern Approach". Prentice Hall, 2 ed., 2003.

[2] Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning: An Introduction". The MIT Press, 1 ed., 1998.

[3] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey". Journal of Artificial Intelligence Research vol. 4, pp. 237-285, 1996.

[4] G. Tesauro. "Temporal Difference Learning and TD-Gammon". Communications of the ACM, 38(3), March 1995.

[5] S. J. Bradtke, A. G. Barto, and P. Kaelbling, "Linear least-squares algorithms for temporal difference learning". Machine Learning, pp. 22-33, 1996.

[6] Michail G. Lagoudakis, Ronald Parr, "Least-Squares Policy Iteration".The Journal of Machine Learning Research vol. 4, pp. 1107-1149, 2003.

[7] C++ Programming Language. http://www.stroustrup.com/C++.html.

[8] Microsoft Visual Studio. http://www.visualstudio.com/.

[9] Microsoft .Net. http://www.microsoft.com/net.

[10] Backgammon, The game. http://www.bkgm.com/.

[11] N Papahristou, I Refanidis – "Improving Temporal Difference Learning Performance in Backgammon Variants". Advances in Computer Games, pp. 134-145, 2012 – Springer. http://ai.uom.gr/nikpapa/Palamedes/.

[12] Backgammon Program JellyFish. http://www.backgammoned.net/all-about-backgammon/jellyfish.html.

[13] Backgammon Program Snowie. http://www.bgsnowie.com/.