Ημερομηνία: 03/05/2021

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ: ΗΜΜΥ

Τεχνητή Νοημοσύνη

ΠΛΗ311

2η Προγραμματιστική Άσκηση

Ομάδα Εργασίας LAB31146722 :

Περάκης Εμμανουήλ ΑΜ: 2017030099

Γεωργακάς Ιωάννης-Ιάσων ΑΜ: 2017030021

Εισαγωγή και Περιγραφή του Προβλήματος:

Στην παρούσα προγραμματιστική εργασία σχεδιάστηκε και υλοποιήθηκε ένας πράκτορας ο οποίος δύναται να παίζει μία παραλλαγή του κλασικού παιχνιδιού σκάκι, το οποίο ονομάζεται TUC-CHESS. Η «νοημοσυνη» του πράκτορα που του επιτρέπει να βρίσκει μία ικανοποιητική στρατηγική, στα πλαίσια των κανόνων του παιχνιδιού, έγκειται σε δύο αλγορίθμους αναζήτησης υπό αντιπαλότητα (adversarial search), τον Minimax και τον Monte Carlo Tree Search (MCTS). Επιπλέον, παρέχεται η δυνατότητα χρήσης της τεχνικής α-β pruning στον αλγόριθμο Minimax.

Υλοποίηση και Ανάλυση Αποτελεσμάτων:

Οι αλγόριθμοι αναζήτησης υπό αντιπαλότητα υλοποιήθηκαν ως μέρος του ήδη υπάρχοντος κώδικα του Πελάτη (Client) που παρέχεται εκ μέρους του διδάσκοντα του μαθήματος, και είναι γραμμένος σε γλώσσα Java. Πιο συγκεκριμένα η κλάση World, ένα στιγμιότυπο της οποίας διαχειρίζεται ο Client, προσομοιώνει την κατάσταση στην οποία βρίσκεται η σκακιέρα ανά πάσα χρονική στιγμή και είναι υπεύθυνη για την σωστή διεκπεραίωση της εκάστοτε κίνησης, η οποία λαμβάνεται από το server, από οποιονδήποτε παίκτη. Οι κινήσεις ενός παίκτη προέρχονται από την επανειλημμένη εκτέλεση ενός από τους αλγορίθμους αναζήτησης υπο αντιπαλότητα. Η λογική αλλά και η δομή των αλγορίθμων Minimax και MCTS αναλύεται σε βάθος στα μέρη Α και Β αντίστοιχα.

Μέρος Α

Αλγόριθμος Minimax:

Ο αλγόριθμος Minimax είναι ένας αλγόριθμος αναζήτησης υπό αντιπαλότητα ο οποίος δημιουργεί αναδρομικά όλες τις πιθανές εκβάσεις του παιχνιδιού έως ότου φτάσει σε μία τερματική κατάσταση ή στο προκαθορισμένο βάθος που δίνεται σαν όρισμα. Κάθε κόμβος του δέντρου αναπαριστά μία πιθανή κατάσταση της σκακιέρας, και κάθε ακμή είναι μία κίνηση η οποία οδηγεί σε μία νέα κατάσταση.Ο αλγόριθμος επιστρέφει μία κίνηση η οποία μεγιστοποιεί το κέρδος του παίκτη το οποίο υπολογίζεται αναδρομικά από την συνάρτηση αξιολόγησης (evaluateMove) στις τερματικές καταστάσεις υποθέτοντας ότι ο αντίπαλος παίκτης παίζει για να ελαχιστοποιήσει το κέρδος του.Ο Minimax αλγόριθμος εκτελεί μία πλήρη αναζήτηση κατά βάθος στο δέντρο του παιχνιδιού.Θεωρώντας ότι το μέγιστο βάθος του δέντρου είναι ίσο με m και ότι υπάρχουν b διαθέσιμες κινήσεις σε κάθε κόμβο, η χρονική πολυπλοκότητα του αλγορίθμου είναι $O(b^m)$, ενώ η χωρική πολυπλοκότητα είναι $O(bm)$.

Αλγόριθμος α-β pruning :

Ο αλγόριθμος a-b pruning χρησιμοποιείται ως μέθοδος αποκοπής (cut off) σε ένα δέντρο Minimax ώστε να μην απαιτείται η εξερεύνηση ολόκληρου του δέντρου για την εύρεση της καλύτερης κίνησης το οποίο θα απαιτούσε περισσότερες κινήσεις και κατά συνέπεια περισσότερο χρόνο. Αντιθέτως αποκλείονται ορισμένα μονοπάτια να εξερευνηθούν που είναι γνωστό ότι δεν μπορούν να μας επιστρέψουν κάποια καλύτερη κίνηση από την ήδη προϋπάρχουσα. Εφαρμόζοντας τον αλγόριθμο α-β pruning στο ίδιο δέντρο που εφαρμόζεται και ο Minimax επιλέγεται η ίδια κίνηση.

Λεπτομέρειες υλοποίησης Minimax και α-β pruning:

Στην παρούσα υλοποίηση του Minimax αλγορίθμου, αρχικά βρίσκει όλες τις πιθανές κινήσεις που μπορούν να πραγματοποιηθούν από τον εκάστοτε παίκτη, στη συγκεκριμένη κατάσταση σκακιέρας, και εν συνεχεία εκτελεί κάθε μία από τις κινήσεις με χρήση της συνάρτησης makeMove. Η συνάρτηση makeMove κάνει την κίνηση στην σκακιέρα και ανανεώνει τους πόντους των δύο παικτών καθώς και τον αριθμό των πιονιών τους. Ύστερα καλείται αναδρομικά ο αλγόριθμος για τον αντίπαλο παίκτη και αυξάνεται το βάθος. Κατά την υποχώρηση από τις τερματικές καταστάσεις, ο αλγόριθμος αναιρεί όλες τις κινήσεις που έγιναν μέχρι αυτό το βάθος και υπολογίζει την τιμή του μονοπατιού. Τέλος, επιλέγεται η κίνηση που είναι άμεσος απόγονος του παίκτη και προσφέρει το υψηλότερο κέρδος.

Στην περίπτωση του α-β pruning χρησιμοποιούνται οι βοηθητικές μεταβλητές ‘alpha’ και ‘beta’ οι οποίες ανανεώνονται κατά την αναζήτηση και μόλις η κατάσταση της σκακιέρας του εκάστοτε κόμβου είναι χειρότερη από τις προηγούμενες τιμές των alpha και beta αντίστοιχα κλαδεύεται το ανάλογο κλαδί του κόμβου.

Singular Extensions :

Οι μοναδικές επεκτάσεις (singular extensions) αποτελούν προσέγγιση της ανθρώπινης σκέψης με στόχο να βρεθεί μία κίνηση ανώτερη από όλες τις υπόλοιπες. Συγκεκριμένα υλοποιήθηκαν δύο μοναδικές επεκτάσεις, η πρώτη ελέγχει αν μπορεί ο αντίπαλος βασιλιάς να σκοτωθεί από ένα δικό μας πιόνι και επιστρέφει τη κίνηση αυτή αντί οποιασδήποτε άλλης κίνησης. Η δεύτερη επέκταση η οποία πραγματοποιήθηκε προσεγγίζει την ανθρώπινη σκέψη να αμυνθεί ο βασιλιας αν απειλείται από κάποιο πιόνι του αντιπάλου. Αυτο υλοποιήθηκε στην συνάρτηση defendKing η οποία ελέγχει τα γειτονικά τετραγωνάκια του βασιλιά αν

υπάρχει κάποιο πιόνι το οποίο τον απειλεί και τα τετραγωνάκια σε απόσταση το πολύ τριών τετραγώνων αν βρίσκεται κάποιος αντίπαλος πύργος ο οποίος να αποτελεί κίνδυνο.

Move Ordering :

Η σειρά εξέτασης των κινήσεων είναι άρρηκτα συνδεδεμένη με την αποτελεσματικότητα του αλγορίθμου a-b pruning καθώς μπορούν να εξεταστούν νωρίτερα οι καλύτερες από τις κινήσεις που είναι διαθέσιμες από την εκάστοτε κατάσταση. Συγκεκριμένα η σειρά εξερεύνησης των κινήσεων πραγματοποιείται εξετάζοντας πρώτα αν το πιόνι του αντιπάλου που απειλείται είναι βασιλιάς (K), ύστερα αν το πιόνι είναι πύργος (R) και τέλος αν είναι στρατιωτάκι (P) δηλαδή δίνεται προτεραιότητα στην εξέταση των κινήσεων που επιφέρουν την αφαίρεση ενός πιονιού του αντιπάλου. Σε κάθε μία από τις πιθανές κινήσεις ενός παίκτη ανατίθεται μία προτεραιότητα και ύστερα πραγματοποιείται μία ταξινόμηση με βάση τις προτεραιότητες αυτές. Η υψηλότερη δυνατή προτεραιότητα στην παρούσα υλοποίηση είναι το μηδέν (0).

Μέρος Β

Αλγόριθμος Monte Carlo Tree Search(MCTS):

Ο αλγόριθμος MCTS πραγματοποιείται σε 4 στάδια ,την επιλογή του κόμβου-φύλλου για επέκταση,την επέκταση του κόμβου που επιλέχθηκε,την προσομοίωση ενός παιχνιδιού με τυχαίες κινήσεις από τους δύο παίκτες από την κατάσταση της σκακιέρας του κόμβου και τέλος την υπαναχώρηση στη ρίζα του δέντρου ανανεώνοντας τις κατάλληλες τιμές των κόμβων.Αρχικά η επιλογή του κόμβου-φύλλου πραγματοποιείται με βάση τη μεγαλύτερη τιμή UCT η οποία δίνεται από το παρακάτω τύπο :

$$UCT = V + 2C\sqrt{\frac{ln(N)}{n}}$$

όπου V : το μέσο reward του παιδιού

C : μία σταθερά ίση με $\frac{1}{\sqrt{2}}$

N : ο αριθμός των φορών που ο τρέχων κόμβος έχει δεχτεί επίσκεψη

n : ο αριθμός των φορών που ο κόμβος-παιδί *i* έχει δεχτεί επίσκεψη

Στη συνέχεια ο κόμβος-φύλλο που επιλέχθηκε επεκτείνεται βρίσκοντας όλες τις πιθανές καταστάσεις της σκακιέρας με βάση τις επιτρεπόμενες κινήσεις και προσθέτοντας τα ως παιδιά του κόμβου. Ύστερα λαμβάνει χώρα η προσομοίωση ενός παιχνιδιού από τυχαίες

κινήσεις και από τους δύο παίκτες έως ότου τελειώσει το παιχνίδι ή μέχρι ένα συγκεκριμένο βάθος επιστρέφοντας μία αξία για τη κατάσταση που βρίσκεται το παιχνίδι. Τελος πραγματοποιείται η υπαναχώρηση στη ρίζα του δέντρου ανανεώνοντας τις τιμες των rewards(V) στους κατάλληλους κόμβους . Η διαδικασία επαναλαμβάνεται για μία πεπερασμένη χρονική διάρκεια των 6 δευτερολέπτων και επιλέγεται η καλύτερη κίνηση από τη ρίζα προς τα παιδιά της με βάση τη τιμή UCT των παιδιών.

Μέρος Γ

Στο μέρος αυτό συγκρίνεται η απόδοση των δύο αλγορίθμων αναζήτησης υπό αντιπαλότητα, με χρήση της παρακάτω συνάρτησης αξιολόγησης.

Evaluation function :

Η συνάρτηση αξιολόγησης που χρησιμοποιήθηκε αρχικά είναι η εξής:

$$|my\ points\ +\ my\ pieces|\ -\ |opponent's\ points\ +\ opponent's\ points|$$

Δοκιμάζοντας διαφορετικές παραλλαγες της συνάρτησης αξιολόγησης παρατηρήθηκε σημαντική βελτίωση στην απόδοση των αλγορίθμων χρησιμοποιώντας την παρακάτω συνάρτηση αξιολόγησης :

$$|my\ points\ +\ my\ value\ of\ pieces|\ -\ |opponent's\ points\ +\ opponent's\ value\ of\ pieces|$$

όπου το value of pieces υπολογίζεται ως έξη :

$$value\ of\ pieces\ =\ w_1P\ +\ w_2R\ +\ w_3K$$

όπου το w_1 είναι ίσο με 1

το w_2 είναι ίσο με 3

το w_3 είναι ίσο με 8

το P είναι ο αριθμός των στρατιωτών

το R είναι ο αριθμός των πύργων

το K είναι ο αριθμός των βασιλιάδων

Η παραπάνω συνάρτηση αξιολόγησης επιλέχθηκε καθώς λαμβάνει υπόψη και τη μέγιστη δυνατή απώλεια των καλύτερων πιονιών του αντιπάλου και την διατήρηση των δικών του δυνατοτερων πιονιών του συνδυάζοντας το επιθετικό κομμάτι του παιχνιδιού μαζι με τη αμυντική προσέγγιση που απαιτείται.

Μετά από έναν ικανοποιητικό αριθμό παρτίδων ανάμεσα σε έναν πράκτορα που βασίζεται στον Minimax αλγόριθμο και έναν πράκτορα που βασίζεται στον MCTS αλγόριθμο παρατηρείται το γεγονός ότι ο πρώτος αναδεικνύεται στην πλειονότητα των περιπτώσεων νικητής. Αυτό ήταν αναμενόμενο αφού το συγκεκριμένο παιχνίδι είναι αιτιοκρατικό και με τέλεια πληροφόρηση, πράγμα που ευνοεί τον αλγόριθμο Minimax καθώς οι κινήσεις που επιλέγει δεν είναι τυχαίες. Εν αντιθέσει, ο αλγόριθμος MCTS κατά την προσομοίωση ακολουθεί τυχαίες κινήσεις που μπορεί να μην επιφέρουν κάποιο καλό αποτέλεσμα στο χρονικό περιθώριο που δίνεται. Αξίζει να σημειωθεί ότι, στη θεωρία, όσο αυξάνεται ο αριθμός των προσομοιώσεων στον MCTS αυξάνεται και η πιθανότητα να βρεθεί κάποια καλύτερη κίνηση.

Επιπροσθέτως εξετάζεται ο μέσος παράγοντας διακλάδωσης (branch factor) των δύο αλγορίθμων. Ο παράγοντας αυτός επηρεάζεται από τις μεθόδους cut-off, move ordering και τα singular extensions που χρησιμοποιούνται στον Minimax αλλά και από τον αριθμό των προσομοιώσεων στην περίπτωση του MCTS. Ένα καλό move ordering για παράδειγμα μπορεί να οδηγήσει σε γραμμική αναζήτηση του δέντρου πράγμα που μειώνει δραματικά το χρόνο εύρεσης μιας κίνησης. Παρακάτω φαίνεται ένας Πίνακας με ενδεικτικά branch factors για τους δύο αλγορίθμους.

game	Minimax with α-β pruning	MCTS
1	8.75	11.59
2	11.94	10.38
2	9.63	8.7
3	10.16	7.22
4	13.4	6.36
5	9.1	8.11
6	11.7	8.23
7	11.7	8.23
8	8.41	12.22

Παράρτημα

Κανόνες Παιχνιδιού TUC-CHESS:

- Οι παίκτες παίζουν εναλλάξ και ο λευκός παίζει πρώτος.
- Ο λευκός παίκτης μετακινεί τους λευκούς πεσσούς και ο μαύρος τους μαύρους.
- Τα πιόνια μετακινούνται πάντα προς τη μεριά του αντιπάλου, είτε ένα βήμα μπροστά αν δεν υπάρχει κάποιος πεσσός σε εκείνη τη θέση είτε ένα βήμα διαγωνίως (αριστερά ή δεξιά) αν υπάρχει κάποιος πεσσός του αντιπάλου σε εκείνη τη θέση.
- Οι πύργοι μετακινούνται μέχρι και κατά τρεις θέσεις προς οποιαδήποτε κάθετη και οριζόντια κατεύθυνση (μη διαγώνια). Αν στις ενδιάμεσες θέσεις κάποιας κατεύθυνσης υπάρχει πεσσός τότε δεν επιτρέπεται η μετακίνηση.
- Οι βασιλιάδες μετακινούνται προς οποιαδήποτε κάθετη και οριζόντια κατεύθυνση κατά μία θέση.
- Τα πιόνια που φτάνουν στην τελευταία γραμμή (γραμμή 0 για τα λευκά και γραμμή 6 για τα μαύρα) απομακρύνονται από τη σκακιέρα, κερδίζοντας 1 πόντο.
- Όλοι οι πεσσοί που αιχμαλωτιζονται απομακρύνονται από τη σκακιέρα διαπαντός.
- Η αιχμαλώτιση ενός πιονιού αυξάνει τη βαθμολογία του παίκτη κατά 1 πόντο.
- Η αιχμαλώτιση ενός πύργου αυξάνει τη βαθμολογία του παίκτη κατά 3 πόντους.
- Η αιχμαλώτιση ενός βασιλιά αυξάνει τη βαθμολογία του παίκτη κατά 8 πόντους.
- Ένα bonus δίνει, στον παίκτη που μετακινείται στη θέση που αυτό βρίσκεται, 1 πόντο με πιθανότητα 0.9 και 0 πόντους με πιθανότητα 0.1.
- Μετά την κίνηση ενός παίκτη ένα νέο bonus εμφανίζεται σε κάποια από τις θέσεις στις οποίες δεν υπάρχει πεσσός ή bonus, με πιθανότητα 0.1.
- Για οποιαδήποτε κίνηση ABCD ισχύει ότι A, C $\in$ {0, 1, 2, 3, 4, 5, 6} (7 γραμμές) και B, D $\in$ {0, 1, 2, 3, 4} (5 στήλες).
- Κάθε παρτίδα λήγει είτε όταν ένας από τους δύο βασιλιάδες αιχμαλωτιστεί ή όταν στο παιχνίδι δεν έχει μείνει κανένας άλλος πεσσός εκτός των δύο βασιλιάδων ή όταν ξεπεραστεί το χρονικό όριο των 14 λεπτών.
- Για την αποστολή της κίνησή σας, όταν είναι η σειρά σας να παίξετε, μην ξεπερνάτε τα 6 δευτερόλεπτα.

Πρωτόκολλο Επικοινωνίας :

Η επικοινωνία μεταξύ του client και του server επιτυγχάνεται με χρήση του πρωτοκόλλου UDP. Ο server δέχεται μηνύματα στη θύρα(Port) 9876 και η διεύθυνση IP του είναι η διεύθυνση του local host. Τα πακέτα δεδομένων που στέλνει και δέχεται ο server μπορούν να έχουν μέγεθος μέχρι 200 bytes.

Documentation:

[1] Artificial Intelligence A Modern Approach Third Edition, Stuart Russell, Peter Norvig

[2]//https://www.baeldung.com/java-monte-carlo-tree-search?fbclid=IwAR2ikonffNmUoDyr44z6mTdWoDyBLXZeTP-R0t6DVruWLWXjzV1NQDgpr3M

[3]https://pastebin.com/VSehqDM3

[4]https://pastebin.com/rZg1Mz9G

[5]COMP311notes_1.pdf

[6]https://medium.com/@quasimik/monte-carlo-tree-search-applied-to-letterpress-34f41c86e238