

חלק א'

השחקן הפשוט מתחיל: 1. הפשוט ניצח 2. הפשוט ניצח 3. תיקו בין הרנדומלי לפשוט

השחקן הרנדומלי מתחיל: 1. הרנדומלי ניצח 2. הפשוט ניצח 3. הפשוט ניצח

ניתן לראות שהתוצאה נוטה לכיוון השחקן הפשוט שיש לו אסטרטגיה יותר מתוחכמת, למרות שמספר ההרצות נמוך וקשה לקבוע באופן חד משמעי, אך זה הגיוני.

חלק ב'

1. נגדיר היוריסטיקה:

- נשים לב כי ב-simple player הפונקציה utility נותנת ניקוד ללוח לפי כמות השחקנים שיש למבצע המהלך לאחר המהלך ביחס למהלכים אחרים. נשתמש בחלק זה של ההיוריסטיקה גם בהיוריסטיקה שלנו, אבל נשדרג אותו כך שיהיה ניקוד גבוה ממש (99 נקודות) עבור שחקן שיושב בפנינה כלשהי בלוח החדש וניקוד שלילי (-24) עבור שחקן שיושב בשכנות לפנינה כלשהי בלוח החדש לעומת כל שחקן אחר שיש לו ניקוד 1.
- נציין שלא הוספנו את הניקוד המיוחד הנ"ל בשלבים מאוחרים של המשחק (לאחר ש85% מהלוח כוסה) כיון שבשלב זה פרמטר אחר מעניין אותנו והוא כמות החורים בלוח שגודלם זוגי לאחר המהלך שלנו (נסביר את המוטיבציה בסעיף הבא), את זה בדקנו ע"י backtracking בכל פעם שראינו ריבוע ריק, אבל אין כפילויות כיוון שמילאנו את החורים האלה בכוכביות לאחר שספרנו אותם ולבסוף מילאנו אותם בכלום שוב במעבר יחיד. על כל חור זוגי נתנו ללוח בפונקציה utility 5 נקודות.
- פרמטר נוסף ואחרון הוא כמה מהלכים נשארו ליריב לעשות לאחר המהלך שלנו, נתנו ניקוד שלילי של נקודה עבור כל מהלך שיש ליריב לבצע בלוח.
2. המוטיבציה להגדרה שלנו להיוריסטיקה היא שיפור האסטרטגיות לאחר ניתוח המשחק. המוטיבציות לשלושת הפרמטרים אותם לקחנו בחשבון הן:
- א. כמות המהלכים שיש ליריב לאחר המהלך שלנו- ככל שכמות זו יותר קטנה זה יותר טוב לנו, כיוון שזה נותן לו יותר כיווני התקדמות.
- ב. מיקום המהלך האחרון שלנו- נתנו עדיפות גדולה למהלך שמניח בפנינה, ועדיפות שלילית למהלך שמניח בנקודה שהיא שכנה של הפנינה. הסיבה לכך היא שכאשר הונח כלי משחק בפנינה, הוא לעולם לא יוכל להיצבע חזרה לצבע של היריב והוא יוכל לגרום להיפוך של הרבה כלי משחק של היריב כיוון ששולט על אלכסון הריבוע ושתי פאות ממנו. הסיבה שנתנו עדיפות שלילית לכל שכן של פנינה, היא מפני שאם נבצע מהלך לשם נגביר את הסיכוי לכך שהיריב ישתלט על הפנינה, שאנחנו רוצים להשתלט עליה. בנוסף בשלב מתקדם של המשחק ביטלנו את העדיפות לפינות ולשכניהם מכיוון שהיא פחות חשובה מכך שנסגור את החורים אחרונים בשלב זה (מפורט בסעיף ג').
- ג. כמות ה"חורים" בלוח שהם בגודל זוגי - בשלב מתקדם במשחק (85 אחוז מהלוח מלא) חורים אלו יתנו ניקוד חיובי ללוח. הסיבה לכך היא מכיוון שאנחנו רוצים להיות האחרונים שסוגרים את החורים בלוח כיוון שהשחקן שסוגר את החור ברוב הסיכויים יצבע לצבעו כמות יפה של כלי משחק של היריב וגם לא ייצבע חזרה לצבע היריב, הוא "אומר את המילה האחרונה".

3. ממומש

4. בכל הרצה (לא משנה מי מתחיל) better מנצח.

חלק ד'

אנחנו מצפים שיהיה הבדל בין ביצועי שחקן min-max לשחקן alpha-beta בתנאים שיש עצים גדולים לנתח (שלבים לא אחרונים של המשחק) כיוון שאנחנו נגזום ונחסוך זמן וכך נוכל להעמיק יותר. כשאין הרבה לאן להעמיק אין הבדל ביניהם.

חלק ה'

1. מה- alpha beta player אנחנו מצפים לביצועים הכי טובים מבין כל השחקנים כיוון שהוא מנתח את עץ המשחק (מה שלא נעשה ב-better player) וגם מעמיק יותר מ-min_max כיוון שהוא גוזם ענפים לא רלוונטיים.
2. א. פונקציית selective deepening – בעזרת פונקציה כזאת נוכל למשל להחליט אם אנחנו ממשיכים להעמיק בעץ לפי ערך השונות בין התוצאות של העמקות קודמות.
 ב. פונקציית time for step – בעזרת פונקצייה כזאת נוכל לחלק את הזמן בין המהלכים, כוללות במצב בהתחלת כל סבב של K מהלכים, למשל אם יש הרבה מהלכים להתקדמות במשחק אנחנו נעדיף להקדיש יותר זמן בהתחלת הסבב.

חלק ו'

תארו אפשרויות ואסטרטגיות נוסף על מה ששימשתם בחלקים הקודמים, שהתוכנה משתמשת כדי לנצח את המשחק:

- שימוש בסטטיסטיקות על משחקים קודמים מתוך מאגר משחקים.
- שימוש בפונקציית selective deepening, שמחליטה על פי תוצאות מהעמקות קודמות האם שווה להעמיק עוד.
- היוריסטיקות ממושקלות לפי החשיבות שלהן בזמנים שונים במשחק

1. הספר נותן לנו אפשרות לזהות מצבים שהיו במשחקים קודמים וגרמו לאנשים לנצח בשביל ללמוד מהם מה לעשות בצעד הבא.
2. 5 הפתיחות הראשונות במילון הן:

13493 +d3-c5+f6-f5+e6-e3+c3-f3+c4-b4

13493 +d3-c5+f6-e3+c3-f5+e6-f3+c4-b4

13493 +d3-c5+e6-f5+f6-e3+c3-f3+c4-b4

7432 +d3-c5+f6-f5+e6-e3+d6-f7+g6-e7

7432 +d3-c5+e6-f5+f6-e3+d6-f7+g6-e7

3. בנינו מילון כך שלוח של n-1 צעדים ממופה למהלך העוקב הכי נפוץ שלו ב-openbook, שאליו הגענו מכיוון שעברנו על כל הפתיחות בסדר ההפוך (תדירות עולה).

4. החסרונות הם:

א. האסטרטגיה של openbook משתנה לאחר 10 מהלכים, ויכול להיות שלא תתאים לאסטרטגיה העוקבת לה.

ב. לא לוקחים בחשבון האם הפתיחות בספר בסופו של דבר הובילו לניצחון או לא.

ג. לא לוקחים בחשבון את כל האפשרויות הסימטריות של כל רשומת פתיחה.

5. מכיוון שהמשחק הוא סימטרי ניתן להסתכל על ה-openbooks הסימטריים ל-openbook הנתון. ה-openbooks הסימטריים הם:

א. עבור הלוח שנמצא 180 מעלות ימינה מהלוח הנתון

ב. עבור הלוח שהוא transpose של הלוח הנתון

ג. עבור הלוח שנמצא 180 מעלות ימינה מהלוח שהוא transpose של הלוח הנתון

ה design יראה כך:

כמו ה-design המתואר בסעיף 3, אבל בנוסף לכל לוח נבנה את שלושת הלוחות המתאימים

לו מסעיפים א'-ג' ואם הוא מופיע במילון נחזיר את המהלך כאשר מופעלת עליו את

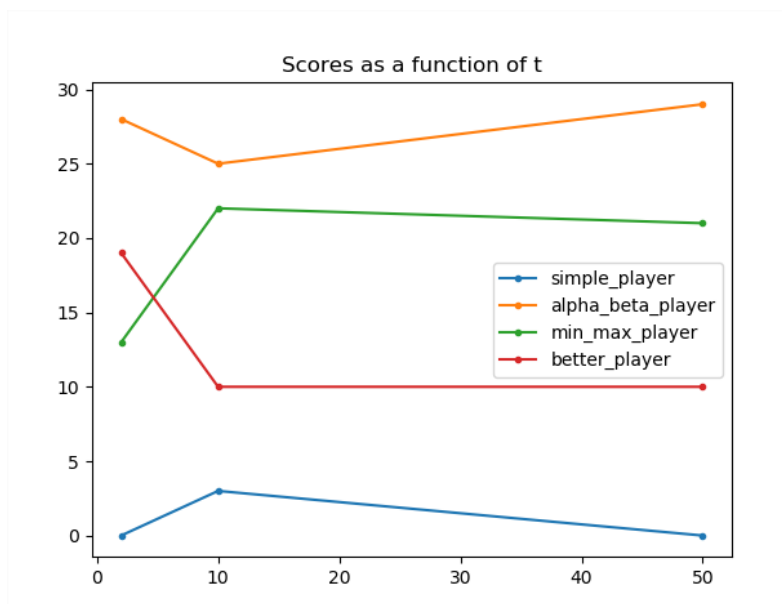
הפונקציה שהופעלה על הלוח מתוך א'-ג'. לדוגמא: אם קיבלנו לוח כלשהו A והלוח AT

מסובב ב-180 מעלות נמצא במילון, ניקח את ה-value של המהלך הבא מהמילון אבל נעשה

לו סיבוב 180 מעלות ו-transpose.

בנוסף, ניתן להסתכל בכל פתיחה רק על המהלכים של הצד המנצח כמהלכים טובים להכניס למילון.

חלק ז'



t = 2	t = 10	t = 50	player_name
0	3	0	simple_player
28	25	29	alpha_beta_player
13	22	21	min_max_player
19	10	10	better_player

ניתוח התוצאות:

נציין כי בשחקנים Min_max, alpha_beta השתמשנו בהיריסטיקה של better.

נשים לב כי בצורה די נחרצת (פרט לחריגה אחת ב-t=2) הדירוג באיכויות השחקנים הוא:

1. Alpha_beta
2. Min_max
3. Better
4. Simple

וזוה אכן הדירוג המתבקש.

Alpha_beta – ניתן לראות כי באופן יחסי לכולם הוא מנצח הכי הרבה אבל

Min_max – ניתן לראות שמינמקס עולה משמעותית בביצועיו מ-2 ל-10 שניות אבל אחר כך הוא די מתון, אפשר להסביר זאת בכך שכמות הצמתים בעץ עולה באופן אקספוננציאלי, ואם נגיד בין 2 ל-10 שניות הצלחנו לעלות מעומק 3 לעומק 6 אז מ-10 ל-50 נעלה אולי מעומק 6 ל-7. (הזמן גדל לינארית ובהתאם כמות הצמתים גדלה אקספוננציאלית).

Better – ניתן לראות שהוא יורד בביצועיו באופן סימטרי למינמקס מכיוון שמינמקס גובר עליו כשהוא מספיק לחפש יותר לעומק ולכן מנצח אותו (ולוקח לו את הנקודות). Better מנצח את מינמקס ב-2 שניות מכיוון שמינמקס לא מספיק להעמיק מספיק ו-better משתמש ב-openingBook שנותן לו יתרון על מינמקס.

Simple – ניתן לראות שהוא עולה בביצועיו כאשר יש לו 10 שניות מכיוון שהמשחקים של α_beta ומינמקס הם מקריים (אמנם בהסתברות נמוכה) מכיוון שעוד תהליכים רצים במחשב ולכן מדידת הזמן לא קבועה וכתוצאה מכך יש מעט מאוד מקרים אבל simple מנצח את α_beta פעם אחת ואת \min_max פעמיים. בסופו של דבר simple מפסיד לכולם באופן יחסי וזה אכן זהה למה שאנחנו מצפים.