**מבוא לבינה מלאכותית– גיליון 1**

יפתח אדלשטיין:316089952

אליאור גיגי : 318297496

**חלק יבש**

1. יתרונות היוריסטיקה: היוריסטיקה שואפת לייצר מצבים בהם יהיו מספר אפשרויות לשחקן ליצור mill ללא תלות במהלך היריב.

חסרונות היוריסטיקה: היוריסטיקה תמשיך לייצר מצבים כאלו, על חשבון מהלכים הגנתיים אחרים כמו חסימת שחקני היריב, וכן הריגת חיילים של האויב.

1. Pieces number – ההפרש בין מספר החיילים של השחקן פחות מספר החיילים של היריב על המגרש. נסמנו b1

Almost mills – ההפרש בין מספר הקונפיגורציות של שלשות עם 2 חיילים + תא ריק עבור השחקן, פחות אלו של היריב. נסמנו b2

Closed mills – ההפרש בין מספר השלשות של חיילים של השחקן, פחות אלו של היריב. נסמנו b3

Double morris – ההפרש בין מספר הdouble morris של השחקן, למספר הdouble morris של היריב. Double morris- הימצאות חיילים מאותה קבוצה בחמישה תאים אשר להם תא משותף, לדוגמה: 0,1,2,4,7 או .0,1,2,9,17 נסמנו b4

Blocked pieces- ההפרש בין מספר החיילים החסומים של היריב, למספר החיילים החסומים של השחקן. נסמנו b5

את כלל הערכים האלה, ננרמל כמובן כלפי הערך המקסימלי האפשרי עבור היוריסטיקה.

ולבסוף, ניקח קומבינציה לינארית מנורמלת של כלל הגורמים:

1. א. היתרון בהוספת גיזום אלפא ביתא הוא צמצום מספר המצבים שיש להעריך בעץ החיפוש, ובכך מפחית משמעותית את זמן החיפוש ומאפשר להעמיק בעץ. כל זאת תוך שמירה על ערך המינימקס, מסלול המינימקס ואסטרטגיית המינימקס. יתרון זה מושג ע"י העברת שני פרמטרים . כאשר:

אלפא – הערך המינימלי המובטח לשחקן MAX על המסלול מהשורש.

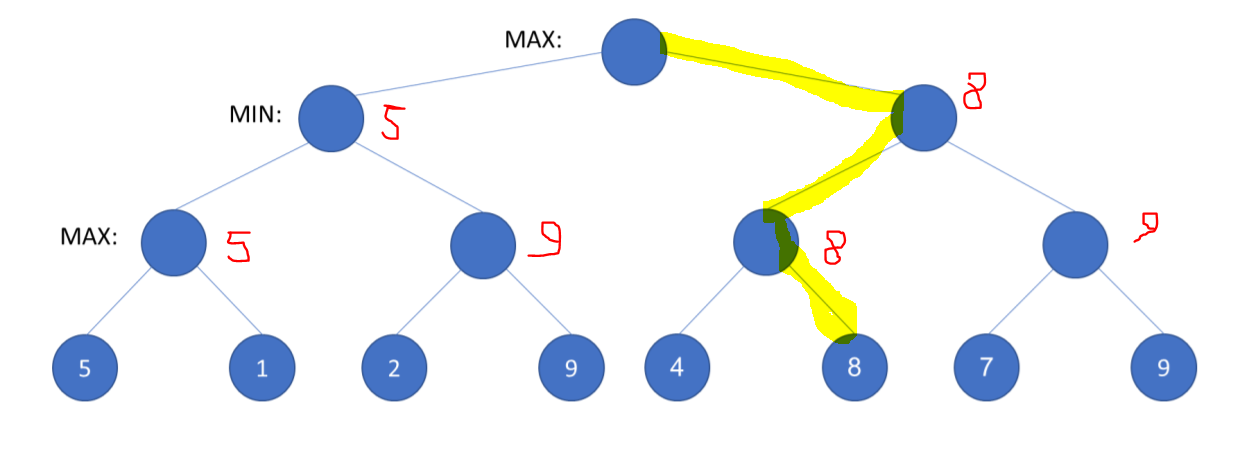
בתא- הערך המקסימלי המובטח לשחקן MIN כלשהו על המסלול מהשורש.

בעזרת שני הפרמטרים נבצע גיזום אלפא וגיזום בתא.

ב. על מנת לקבל גיזום מקסימלי, נמיין בנים של צמתי max בסדר יורד, ובנים של צמתי min בסדר עולה.

ג. אף עלה לא ייגזם על ידי אלפא ביתא.

נסמן את המסלול האופטימלי:



1. לא, ניקח לדוגמה את היוריטיקה שהוצגה בסעיף 1. מהחסרונות שכתבנו ברור שהיא לא תביא לאסטרטגיה אופטימלית, היריב יוכל במהירות ליצור רצפים של חיילים ולסיים את המשחק.
2. א. נגדיר את פונקציות המטרה הבאות, כפי שנראה בגרף הבא:

**Diagram

Description automatically generated**

בכחול מתוארים צמתי מינימום, באדום מקסימום.

בעץ הזה, ניתן לראות כי למרות שקל לראות שהמשחק אינו משחק סכום אפס, האלגוריתם עדיין יחזיר ערכי מטרה קוהרנטיים- שחקן המינימום ישאף לקחת צעדים שיפגעו בשחקן השני, וההיפך.

ב. נתאר את פונקציית המצבים הבאה:

במצב זה, שני השחקנים מנצחים ומפסידים יחדיו, המשחק אינו משחק סכום אפס יותר, ועל כן, מינימקס כבר לא יעבוד, מאחר וכל מהות האלגוריתם היא לצמצם את הרווח של היריב, וכך להגדיל את הרווח שלך. כאשר שני הצדדים מרוויחים מצעד כלשהו, האלגוריתם יכשל(כמו דילמת האסיר). ניתן עץ מתאים:

Diagram

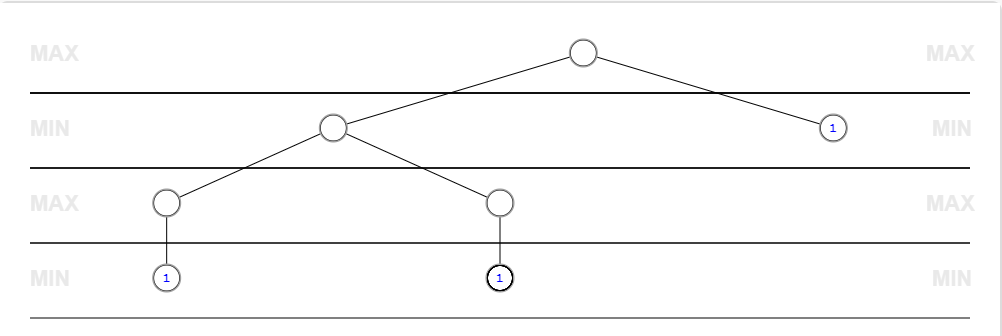
Description automatically generated

כפי שניתן לראות, מינימקס בחר את הערך 1, למרות, שבהינתן הידע המוקדם על הרווח, המסלול האופטימלי הוא דווקא ההבא:

Diagram

Description automatically generated

1. א. נציג דוגמה למצב כזה:



כאשר ערך 1 מסמן מצב סופי (ניצחון של המחשב).   
המחשב העמיק בעץ, וראה כי בכל מקרה הוא ינצח בתור הבא, ובבחירת המסלול האופטימלי בחר במסלול השמאלי.

ב. נציע את השינוי הבא- בתחילת ריצת alphabeta, נעבור ראשית על עומק 1 (צעד אחד של המחשב), אם אחד המצבים הוא מצב סופי המוביל לנצחון, נבחר בו. אחרת, נמשיך בריצת האלגוריתם כרגיל.

1. א.

עבור צומת B:

עבור צומת C:

עבור צומת D:

ב.  
MAX יבחר בפעולה a3.

ג. בהינתן כך, שהערכים של העץ expectimax אינם חסומים, אנו **איננו** מסוגלים לבצע גזימה בדומה לאלגוריתם מינימקס. לא ניתן לבצע גיזום זה, ואת זה נראה בדוגמה הבאה:

Diagram

Description automatically generated

בדוגמה הראשונה קטמנו לפי ערך האלפא שנוצר בענף השמאלי, בו נבחר ערך אלפא = 1, ועל כן גזמנו כי קיבלנו ערך קטן יותר שהיה נגזם, לעומת הדוגמה השנייה בה לא גזמנו וקיבלנו ערך אמיתי. השגיאה ברעיון הגזימה נובע מכך, שהערכים בענפים אינם חסומים, ותוחלת בהיותה תוחלת, קשורה גם לערך העלה, וגם להסתברות הענף. אין דרך לגזום אם איננו יודעים את טווח הערכים אותם אנו מחשבים. בהינתן וידענו, יכולנו לתאר אלגוריתם גזימה מתאים.

1. א. נשנה את ההגדרה של האלגוריתם:
2. A picture containing text

   Description automatically generated

אנו נשנה שורה זאת, כך שבמקום, התנאי יהיה:

1. A picture containing graphical user interface

   Description automatically generated

אנו נשנה שורה זאת, כך שבמקום התנאי השני יהיה:

כפי שניתן לראות, שינוי התנאי (1) יניב בצמתי המקסימום, ערכי מקסימום אשר גדולים בעד אפסילון מאשר הערכים ללא השינוי,

ושינוי תנאי (2) יניב בצמתי המינימום ערכים שקטנים בעד כדי אפסילון.

ב. ניתן דוגמה: כאשר נריץ alpha-beta סטנדרטי:

Diagram

Description automatically generated

Diagram

Description automatically generatedלעומת האלפא בטא המתואר, עם אפסילון = 0.2:

נתאר כיצד זה קרה:

כאשר עברנו על הענף השמאלי, ועברנו על כולו, קיבלנו אלפא=2.

בענף הימני, אנו נבדוק על העלה, עם הערך 2.09, בדיקה, האם 2.09<2, ונקבל כי לא, ולכן, לא נבצע קטימה של הענף. לאחר בדיקה זו, נקבל את הערך 2.03 לבסוף.

כאשר נשנה את האלגוריתם, נקבל כי במקום, בעלה בעל הערך 2.09, נקבל כי 2.09<2.2, פסוק אמת, ולכן נקטום את הענף, ונוציא החוצה במקום 2.03, 2.09.

א.

נוכל לחפש לעומק של . הסבר- לא נצטרך לפתח את צמתי MIN, כלומר במקום לבצע קריאה לget successors, נקבל את הצעד שהיריב יבצע בזמן אפסי. מכאן שנוכל להגיע לעומק הגדול פי 2 מאשר ללא הפרוצדורה.

ב.

ערך המינימקס עם שימוש בפרוצדורה יהיה **גדול או שווה** לערך המינימקס ללא שימוש בפרוצדורה כאשר שניהם מוגבלים לעומק d.

הסבר- אלגוריתם המינימקס מניח כי היריב יבצע את הצעד האופטימלי עבורו (כלומר את הצעד שימזער את ההפסד עבורו). אולם, עם השימוש בפרוצדורה וקבלת הצעד של היריב, היריב בהכרח מבצע צעד שהוא או טוב כמו הצעד אותו המינימקס העריך שיקח, או צעד שפחות טוב ליריב. מאחר והמשחק הוא משחק סכום אפס, צעד זה בהכרח יותר טוב לשחקן שלנו, ולכן, הצעד שלנו יהיה **לפחות** טוב כמו צעד המינימקס, ובהרבה מצבים, טוב יותר.

1. א.  
   הסטודנט אינו צודק. נסביר מדוע הסטודנט עבר רק על חלק קטן ממרחב והחיפוש ולכן אינו יכול להסיק ש5.8 הוא האופטימום הגלובלי ולא לוקלי.  
   בSAHC, בכל שלב מפתחים את כל השכנים, ומתקדמים לשכן שהכי משפר, אם יש כזה. נניח כי כמות השכנים - N- של כל מצב, בממוצע, הינה קטנה בסדרי גודל מ, כלומר . הסטודנט ביצע 1000 הרצות, אשר לכל אחת בממוצע לקח 5 צעדים ולכן עבר בממוצע על . זהו גודל קטן מאוד יחסית לעומת מרחב החיפוש.

ב.

עלינו להכניס ממד "אקראיות" והיחלצות ממישור ואופטימום מקומי. לשם כך נבחר באלגוריתם beam stochastic hill climbing , אשר מבצע חיפוש אלומה של k מצבים, ובוחר באופן רנדומלי את k המצבים לאלומה הבאה בצורה סטוכסטית עם הסתברות פרופרציונלית לגודל שיפור שלהם.

ג.

בהסתברות גבוהה החיפוש בSAHC יתחיל במצב בעל ערך 1, אשר הערך בשכנים זהה. החיפוש ייעצר אחרי שלב אחד כי אין שכן משפר.

אולם, בהסתברות גבוהה ש beam stochastic hill climbingאכן ימצא פתרון, מאחר ובשלב מסויים האלומה ״תכוון״ לכיוון הערכים המוצלחים.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 |
| 4 | 3 | 3 | 1 | 1 | 1 | 1 | 1 |

**חלק רטוב**

**חלק ה**

1. היוריסטיקה עבור MINIMAX, כפי שאמרנו בחלק היבש:

* Pieces number – ההפרש בין מספר החיילים של השחקן פחות מספר החיילים של היריב על המגרש.
* Almost mills – ההפרש בין מספר הקונפיגורציות של שלשות עם 2 חיילים + תא ריק עבור השחקן, פחות אלו של היריב.
* Closed mills – ההפרש בין מספר השלשות של חיילים של השחקן, פחות אלו של היריב.
* Double morris – ההפרש בין מספר הdouble morris של השחקן, למספר הdouble morris של היריב. Double morris- הימצאות חיילים מאותה קבוצה בחמישה תאים אשר להם תא משותף, לדוגמה: 0,1,2,4,7 או 0,1,2,9,17
* Blocked pieces- ההפרש בין מספר החיילים החסומים של היריב, למספר החיילים החסומים של השחקן.

לכן:

ופונקציה ההערכה היוריסטית היא :

1. שחקן התחרות שלנו מבוסס כאמור על שחקן אלפא-בתא בעל מגבלת זמן גלובלית לכל המהלכים יחדיו. את אופן חלוקת הזמנים נתאר בסעיף הבא. לא הגדרנו פונקציה יוריסטית חדשה עבורו, הפונקציה היוריסטית המשמשת את minimax, וalphabeta טובה גם עבורו.
2. עבור זמן מוגבל לתור:

עבור כל איטרצית עומק חיפוש, נמדוד את משך הזמן. נאפשר חיפוש בעומק הבא, אם הזמן הנותר עבור התור (time\_remaining) גדול מזמן החיפוש הקודם כפול מקדם הסיעוף.

כלומר נאפשר חיפוש בעומק אם מתקיים:

כאשר הוא זמן החיפוש בעומק

עבור זמן מוגבל גלובלי:

עבור כל ריצת make\_move (כלומר עבור כל תור) נקצה מגבלת זמן. מגבלה זו תאותחל, עבור כל תור ב , כאשר game\_time הוא הזמן הנותר למשחק. ההגיון בכך הוא לתת לצעדים הראשונים יותר זמן, על מנת לבנות תשתית טובה יותר להמשך המשחק. מרגע שהגדרנו זאת, התור יתנהל בדומה ל"זמן מוגבל לתור". אולם, זמן שלא נוצל במלואו עבור תור, יוכל לשמש אותנו עבור התורות הבאים. בנוסף, הגבלנו את העומק המקסימלי אליו ניתן להגיע ל7, מאחר ואחרי העומק הזה היחס זמן-תועלת נעשה משמעותית פחות איכותי.

**חלק ו**

1. קיבלנו כי alpha beta ינצח במשחקים. הדבר תואם את הציפיות שכן alpha beta, עבור מגבלת זמן זהה לminimax יכול להגיע עמוק יותר בעץ עקב הגיזום.
2. עבור השלב הראשון של הניסוי – בו HeavyABPlayer מחפש בעומק 3:

עבור השלב הראשון של הניסוי – בו HeavyABPlayer מחפש בעומק 2:

הסבר לתוצאות הניסוי:

כאשר עומק החיפוש של HeavyABPlayer היה 3, הוא ניצח את LightABPlayer עבור עומקי חיפוש של 3,4,5. כלומר, איכות היוריסטיקה של heavy הייתה עדיפה על עומקי חיפוש של שחקן עם יוריסטיקה נחותה.

לעונת זאת, כאשר עומק החיפוש של HeavyABPlayer היה 2, הוא נכנס למצב לולאה המונעת ממנו להפסיד לשחקן LightABPlayer. הסיבה לכך היא שעומק חיפוש 2 אינו מספיק כדי להוציא את האינפורמציה הנדרשת ולהגיע לניצחון, ובעומק 2, היוריסטיקה לא מצליחה לתת לנו יתרון משמעותי לאף כיוון של צעד שניתן לקחת, מלבד ללכת הלוך-וחזור.