תרגיל בית 2 - מבוא לבינה מלאכותית 236501

בני נזימוב 314862129
ליעד ארם 315695783

.1

האסטרטגיה של ה-$SimplePlayer$ היא תמיד ללכת למשבצת שממנה קיימים הכי פחות מהלכים חוקיים (אבל קיים לפחות מהלך אחד חוקי).

יתרונות:

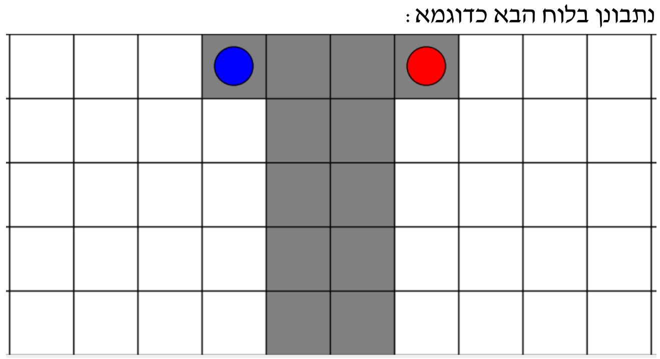
- לפי חוקי המשחק, השחקן האחרון שלא יכול לזוז מקבל קנס. לכן, האינטרס הוא לשחק כמה שיותר מהלכים עד שנגמר המשחק (בציפיה שהאויב יפסל לפנינו). האסטרטגיה של ה-$SimplePlayer$ מכסה את שטח הלוח בצורה היעילה ביותר ובדרך זו הוא מאריכה את זמן המשחק שלו.
- במצבים מסוימים, האסטרטגיה יכולה לגרום ל-$SimplePlayer$ להיצמד לשחקן היריב ובכך **לחסום** אותו ביעילות ולגרום לו לקבל את הקנס.

חסרונות:

- באסטרטגיה זו ה-$SimplePlayer$ מתעלם לחלוטין מקיומם של פירות.
- במצבים מסוימים, האסטרטגיה יכולה לגרום ל-$SimplePlayer$ להיצמד לשחקן היריב ובכך **להיחסם** ולקבל את הקנס.

.2

נתבונן בלוח הבא כדוגמא:



כפי שניתן לראות, מכיוון שבלוח זה אין פירות, העובדה שהאסטרטגיה של ה-$SimplePlayer$ מתעלמת מפירות לא תפגע בו.
בנוסף, מכיוון שקיים קיר המפריד בין השחקן הראשון לשני, לא יתכן שה-$SimplePlayer$ ייחסם על ידי האויב.
בעצם, האסטרטגיה האופטימלית לשחקן במצב זה יהיה למלא את כל השטח בצד שלו של הלוח בצורה היעילה ביותר כך שישחק כמה שיותר מהלכים, בציפיה שהשחקן השני יקבל את הקנס.
וזו היא בדיוק האסטרטגיה בה נוקט ה-$SimplePlayer$.

.3

יתרונות:

- ההיוריסטיקה תתן ערך יוריסטי גבוה למצבים בהם השחקן קרוב לפירות ובכך תעודד אותו לצבור ניקוד מפירות.

חסרונות:

- אם קיבלנו לוח ללא פירות, או שכל הפירות על הלוח כבר נעלמו, ההיורסטיקה הופכת ללא רלוונטית.
- במצבים מסוימים היורסיטקה עלולה לתת ערך יוריסטי גבוה יותר למצב בו השחקן מגיע לנקודה שקרובה לפרי, מאשר למצב שבו השחקן הגיע לנקודה כלשהי כך שבדרך לנקודה זו אכל פרי.
- מרחק מנהטן לא מתאר את המרחק האמיתי לפרי. ייתכן כי הפרי בכלל לא נגיש מהנקודה בה אנו נמצאים.

4.

ארבעת המרכיבים של היוריסטיקה:
נסמן: $\delta_{score} = player\_score - enemy\_score$

$$h_F(s) = \frac{1}{\min_{fruit}\{md(s, fruit)\}}$$

$$h_S(s) = \frac{1}{num\_available\_steps}$$

$$h_E(s) = \frac{1}{md(s, enemy)}$$

$$h_\delta(s) = \begin{cases} 1 - \frac{0.24 \cdot penalty}{|\delta_{score}|} & \delta_{score} \geq \frac{penalty}{2} \\ 0.5 + \frac{0.04 \cdot |\delta_{score}|}{penalty} & 0 < \delta_{score} < \frac{penalty}{2} \\ 0.5 & \delta_{score} = 0 \\ 0.5 - \frac{0.04 \cdot |\delta_{score}|}{penalty} & -\frac{penalty}{2} < \delta_{score} < 0 \\ \frac{0.24 \cdot penalty}{|\delta_{score}|} & \delta_{score} \leq -\frac{penalty}{2} \end{cases}$$

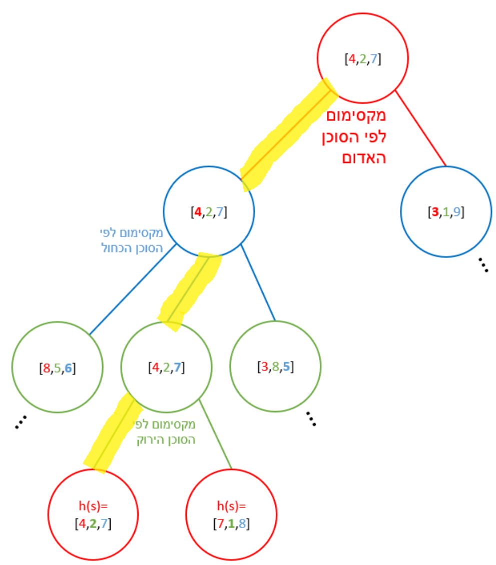
לבסוף היוריסטיקה תהיה $h(s) = w_1 \cdot h_F(s) + w_2 \cdot h_S(s) + w_3 \cdot h_E(s) + w_4 \cdot h_\delta(s)$ כך ש- $w_1 + w_2 + w_3 + w_4 = 1$.
הסבר לכל המרכיבים:

- המרכיב הראשון $h_F(s)$ זהה ליוריסטיקה בסעיף 3.
- המרכיב השני $h_S(s)$ נותן ערך יוריסטי למצבים לפי מספר המהלכים החוקיים, כך שככל שיש לשחקן פחות מהלכים חוקיים הערך גבוה יותר. ההגיון ליוריסטיקה הזו מבוסס על היתרונות באסטרטגיית ה-$SimplePlayer$.
- המרכיב השלישי $h_E(s)$ נותן ערך יוריסטי למצב כפונקציה של מרחק מנהטן מהיריב. מצבים בהם נהיה קרובים ליריב יקבלו ערך יוריסטי גבוה, כדי לעודד את השחקן לנסות לחסום את היריב.
- המרכיב הרביעי $h_\delta(s)$ נותן ערך יוריסטי גבוה (בין 0.5 ל-1) למצבים בהם אנו מובילים על היריב בניקוד, ערך 0.5 עבור תיקו, וערך יוריסטי נמוך (בין 0 ל-0.5) למצבים בהם היריב מוביל עלינו בניקוד.
  ניתן לראות בהגדרת h_δ שככל שההפרש בניקוד גבוה יותר, אם אנחנו מובילים נקבל ערך יוריסטי גבוה יותר (שואף ל-1) ואם היריב מוביל נקבל ערך יוריסטי נמוך יותר (שואף ל-0).
  ההגיון מאחורי יוריסטיקה זו היא לעודד את השחקן להגיע למצבים בהם הפרש הניקוד הוא הגבוה ביותר לטובתו.

נשים לב כי כל המרכיבים מחזירים ערכים בין 0 ל-1, וגם הפונקציה היוריסטית מחזירה בסופו של דבר ערך בין 0 ל-1. הסיבה לכך היא כדי שנהיה עקביים עם סדרי הגודל של המרכיבים בפונקציה היוריסטית. (למשל, אם לא היינו מנרמלים, אז הערך של h_δ היה תיאורטית לא חסום ואילו הערך של h_S היה בין 0 ל-3 ולא היה הגיוני לחבר ביניהם).
אנו צופים שיוריסטיקה זו תשפר את ביצועיו של ה-$SimplePlayer$ כיוון שיש בה מרכיב שמבטא במלואו את האסטרטגיה של ה-$SimplePlayer$, ובנוסף מרכיבים נוספים שמשפיעים על מהלך המשחק ולכן אמורים לעודד את הסוכן לשחק טוב יותר.

.5

אסטרטגיית ה-$Minimax$ מתאימה גם עבור משחק של יותר משני שחקנים כל עוד מניחים שכל אחד מהשחקנים האחרים לוקח את הצעד הגרוע ביותר עבורנו וכך שומרים על נכונותו של משפט ההבטחה של המינימקס.
א. החסרון המשמעותי ביותר הוא שהנחה זו לא משקפת את המציאות בצורה נאמנה, שכן השחקנים האחרים משחקים גם נגד עצמם וככל הנראה לא יקחו כל פעם את הצעד הגרוע ביותר עבורנו. לכן נקבל כי התוצאה שבה אנו משחקים היא שמרנית מידיי.
חסרון נוסף הוא שעץ ה-$minimax$ ייגדל כתלות במספר השחקנים ולכן בזמן נתון נקבל פחות מידע. כלומר, אם במשחק של שני שחקנים, היינו מפתחים לעומק $2d$ ומקבלים שפיתחנו d מהלכים שלנו ו-d מהלכים של היריב, במשחק של n שחקנים, נצטרך לפתח לעומק nd כדי שנפתח לכל שחקן d מהלכים קדימה. לכן, בהגבלת זמן מסויימת, נפתח פחות מהלכים קדימה לכל שחקן, מה שיפגע ברמת הדיוק שלנו בבחירת המהלך הטוב ביותר עבורנו.
ב. אסטרטגיה חלופית תהיה להניח שכל אחד מהשחקנים האחרים יבחר את הצעד הטוב ביותר עבור עצמו (ולא את הצעד הגרוע ביותר עבורנו). הנחה זו משקפת את המציאות בצורה טובה שכן במשחק רב משתתפים כל שחקן מנסה להביא למקסימום את הסיכוי שלו לנצחון כנגד השחקנים האחרים.
רעיון למימוש:
בהגיענו לצומת בו נפעיל את הפונקציה היוריסטית, במקום לחשב את הערך היוריסטי רק לפי עצמנו, נחשב מערך של ערכים יוריסטיים בגודל n (כאשר n הוא מספר השחקנים). בתא ה-i יהיה הערך היוריסטי של צומת זה ביחס לשחקן ה-i. כעת, נפעפע את המערך מעלה בעץ ה-$minimax$ ובכל צומת, אם תורו של השחקן ה-i לשחק, אז נבחר מבין בניו את המערך שמביא למקסימום את התא ה-i. אנו בעצם מניחים כי כל אחד מהשחקנים משחק על פי אסטרטגיית ה-$minimax$ ומנסה למקסם את התוצאה שלו לפי הפונקציה היוריסטית.
שרטוט להמחשה:



.6

א. כן. לכל עומק חיפוש זהה, זמן הריצה של סוכן ה-$Alphabeta$ יהיה לכל היותר כמו זמן הריצה של סוכן ה-$Minimax$. שוויון יתקיים במצב בו לא ניתן לגזום ענפים כלל, למשל כאשר הילדים מסודרים בצורה הגרועה ביותר. לכן, סוכן ה-

$Alphabeta$ במקרה זה יתנהג בדיוק כמו סוכן ה-$Minimax$ והזמנים יהיו שווים. לרוב זה לא יקרה ונוכל לגזום לפחות ענף 1 ובכך זמן הריצה של סוכן ה-$Alphabeta$ יהיה קטן יותר.
ב. לא. השיפור שקיים באלגוריתם ה-$Alphabeta$ ביחס ל-$Minimax$ לא משפיע בשום צורה על נכונות האלגוריתם ומשפט ההבטחה של $minimax$. הרעיון ב-$Alphabeta$ הוא למנוע חישובים מיותרים (שבין כה וכה לא יילקחו בחשבון בבחירת הצעד הטוב ביותר). לכן בהנחה ושני הסוכנים מפתחים לאותו עומק בהכרח יוחזר אותו ערך $minimax$.

.7

א. לא. זמן הריצה של סוכן ה-$Alphabeta$ עם סידור ילדים יהיה לכל היותר כמו זמן הריצה של סוכן ה-$Alphabeta$ ללא סידור ילדים. שוויון יתקיים במצב בו כל הגיזומים שנבצע בסוכן עם סידור הילדים יתבצעו גם אצל הסוכן ללא סידור הילדים. לכן, הסוכן עם סידור הילדים במקרה זה יתנהג בדיוק כמו הסוכן ללא סידור הילדים וזמני הריצה יהיו שווים. לרוב זה לא יקרה ונוכל לגזום יותר ענפים עם סידור ילדים ובכך זמן הריצה של הסוכן עם סידור הילדים יהיה קטן יותר.
ב. לא. תלוי בעומק החיפוש. כפי שהסברנו בסעיף א', זמן הריצה של סוכן עם סידור ילדים לרוב יהיה קטן יותר. לכן, יתכן כי בזמן ריצה נתון סוכן $Alphabeta$ עם סידור ילדים יצליח לפתח את עץ החיפוש לעומק גדול יותר ממה שיצליח לפתח סוכן ה-$Alphabeta$ ללא סידור הילדים, ובכך יחליט לבחור בצעד שונה מסוכן זה.
אם שני הסוכנים מפתחים את עצי החיפוש שלהם לעומק זהה, הם יבחרו בדיוק את אותו מהלך מסיבה דומה להסבר בסעיף 6ב.

.8

ווריאציית ה-$anytime\ contract$ היא ווריאציה בה אנו נדרשים לתת את הפתרון הטוב ביותר בזמן נתון וידוע מראש. העמקה הדרגתית בהקשר זה היא אחת השיטות להתמודד עם הווריאציה הנ"ל, על ידי כך שאנו קוראים לאלגוריתם ה-$minimax$ עם הגבלת עומק הולכת וגדלה עד שנגמר הזמן. בסיום כל איטרציה שומרים את הצעד הנבחר וכשנגמר הזמן מחזירים את הצעד האחרון.

.9

הבעיה בהעמקה ההדרגתית המוצעת נובעת מהאיטרציה האחרונה. לרוב האלגוריתם יופסק באמצע האיטרציה האחרונה, ואנו נחזיר את הפתרון של איטרציה אחת קודם לכן. מכיוון שעץ החיפוש גדל באופן אקספוננציאלי, באיטרציה האחרונה אנו מפתחים כמות צמתים גדולה יותר ממה שפיתחנו בכל האיטרציות קודם לכן יחד. כלומר, רוב משאבי החיפוש מושקעים בפיתוח העץ באיטרציה האחרונה. ומפני שאנו מחזירים את התוצאה של איטרציה אחת קודם לכן, הזמן שהושקע באיטרציה האחרונה בוזבז לשווא. כלומר, קיבלנו כי רוב משאבי החיפוש של האלגוריתם מבוזבזים.
הפתרון המוצע בהרצאה הוא לשמור בכל איטרציה את ערך המינימקס של כל אחד מהבנים ברמה העליונה. נניח כי כעת אנו באיטרציה האחרונה ומפתחים לעומק $d + 1$. איטרציה זו הפסקה באמצע והספקנו לחשב את ערך ה-$minimax$ רק עבור k צמתים.
כעת נסתכל על שאר הצמתים בעומק $d+1$ שאותם לא הספקנו לפתח. צמתים אלה הם בנים של צמתים בעומק d שעבורם שמור לנו ערך ה-$minimax$ מהאיטרציה הקודמת.
לכן, במקום לאבד את כל המידע שגילינו עד כה, נפעפע את ערכי המינימקס של k הצמתים החדשים שגילינו, יחד עם הצמתים ששמרנו מהאיטרציה הקודמת (האבות של הצמתים שלא הספקנו לפתח). כך למעשה, עבור צמתים מסויימים הערך ייקבע מהעומק של האיטרציה האחרונה ועבור הצמתים האחרים הערך ייקבע מהעומק של איטרציה אחת לפניכן. נבחר בצעד שמביא למקסימום, בדומה לאלגוריתם ה-$minimax$ הרגיל.
כך אנו משתמשים במידע שגילינו באיטרציה האחרונה והזמן לא בוזבז לשווא.

.10

א. נדגים את האלגוריתם הבא:
נסמן ב-f_d את התור בו נעלמים הפירות. נסמן ב-g את הזמן הגלובלי של כל שחקן.
יהי תור t קלט לאלגוריתם.

1. חשב חסם עליון למספר התורות $n_{turns} \leftarrow \left\lceil \frac{number\_of\_available\_slots}{2} \right\rceil$.

2. חשב את הנקודות A, B, C הבאות:

- $C = (n_{turns}, 0)$
- $B = \left(f_d, \frac{g}{n_{turns}-f_d}\right)$
- $A = \left(0, \frac{g}{f_d} - \frac{g}{n_{turns}-f_d}\right)$

3. חשב את משוואת הישר AB כ-$y = m_1 \cdot x + n_1$ ואת משוואת הישר BC כ-$y = m_2 \cdot x + n_2$.
4. אם $t \in [1, f_d]$:
   (א) אם $n_{turns} < 3f_d$ בצע $time\_frame \leftarrow m_1 \cdot t + n_1$.
   (ב) אחרת, בצע את $time\_frame \leftarrow -m_1 \cdot t + n_1$.
5. אם $t \in [f_d + 1, n_{turns}]$:
   (א) בצע $time\_frame \leftarrow m_2 \cdot t + n_2$.
6. החזר את $time\_frame$, כאשר $time\_frame$ הוא מסגרת הזמן לתור t.

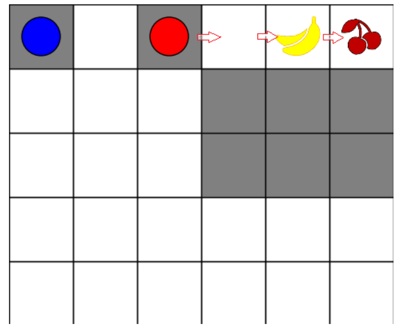
נסביר את האינטואיציה מאחורי האלגוריתם בנקודות:

- לתורות הראשונים אנו מקצים זמן הולך ועולה בקצב לינארי, שכן לתורות ה"ממש" ראשונים אין עדיין השפעה ניכרת על המשחק (אך ככל שמתקדמים ההשפעה גדלה) ולכן אנו נותנים להם כמות זמן בינונית.
- חצי מסך כל הזמן שלנו מוקצה לכל התורות עד שהפירות נעלמים, והחצי השני לשאר התורות. ההגיון לכך הוא שהתורות עד שהפירות נעלמים הם התורות הקריטיים ביותר במשחק, שכן בתורות אלו יש לנו את הסיכוי הגבוה ביותר לצבור פער משמעותי בניקוד ובכך לנצח את היריב.
- הקצאת הזמן פר תור הולכת וגדלה עד שמגיעים לתור בו נעלמים הפירות (f_d), לתור זה אנו מקצים את הזמן הארוך ביותר. ההגיון לכך הוא שלאחר שנעלמו הפירות אופי המשחק משתנה ולכן אנו צריכים זמן רב ככל הניתן על מנת לפתח את העץ לעומק הגדול ביותר ולבחור באסטרטגיה מתאימה להמשך המשחק.
- מהנקודה בה נעלמו הפירות וביצענו את המהלך הארוך ביותר, אנו מקצים לשאר התורות זמן הולך ויורד בקצב לינארי. הסיבה לכך היא שככל שאנו מתקדמים לסוף המשחק, כך עץ החיפוש המלא הולך וקטן כי מספר המצבים האפשרי הכולל הולך וקטן. לכן, ככל שהמשחק קרוב לסוף כך יותר סביר שנצליח לפתח את כל העץ בזמן שהולך וקטן.

.11

אפקט האופק היא תופעה בה אלגוריתם מוגבל משאבים בוחר צעדים לא טובים כדי לדחות תוצאה לא רצויה, אך בלתי נמנעת שנמצאת מעבר לאופק החיפוש.
הפתרון המוצע בהרצאה לבעיה זו הוא העמקה סלקטיבית. בקריטריונים מסויימים, התלויים במשחק עצמו, נמשיך לפתח גם מעבר לעומק שהקצינו. קריטריונים אלו יסווגו מצבים כ-"לא שקטים", למשל: תנודות בערך היוריסטי. במקרים אלו נמשיך לפתח עד ל"רגיעה" של הקריטריונים שהוגדרו.
עץ הפיתוח בהעמקה סלקטיבית ייראה לא אחיד, הוא יהיה עץ מלא עם תתי עצים נוספים לחלק מהעלים.
נציג שתי דוגמאות לדרך בה נוכל להשתמש בשיטה הזאת כדי להועיל לסוכן שלנו:

1. להגדיר שכל עוד הסוכן ממשיך לקחת פירות, ויש שינוי בערכי הניקוד של השחקנים, ממשיכים לפתח.
   מצב בלוח בו כדאי להשתמש בפתרון זה:



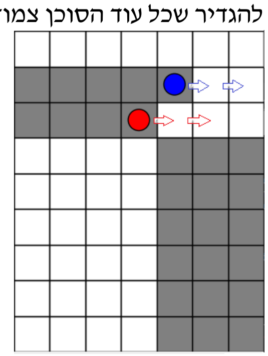
נניח כי הערך של כל פרי בלוח הוא 50, הקנס שניתן לשחקן שלא יכול לזוז הוא 300, ומצב ניקוד השחקנים הוא 0 לכל אחד.

כמו כן השחקן מפתח את עץ החיפוש עד לעומק 4 (שני מהלכים לכל שחקן).

כפי שניתן לראות בשרטוט, לסוכן האדום קיימות שלוש אפשרויות, ללכת ימינה, שמאלה או למטה. ניתן לראות כי אם הסוכן יילך ימינה הוא יפסיד, מכיוון שגם אם ייקח את שני הפירות, עדיין בשקלול עם הקנס הוא יגיע ל--200 נקודות. אך במהלך הפיתוח הוא לא מצליח לראות את ההפסד (כי לא מפתח לעומק מספיק גדול, ולכן ההפסד נמצא מעבר לאופק החיפוש), והוא יעדיף ללכת ימינה כדי לקחת את הפרי הראשון, שאותו הוא כן רואה בעומק החיפוש, ולצבור ניקוד.

אם היינו משתמשים בפתרון שהצענו לעיל, היינו ממשיכים לפתח כי אנו ממשיכים לקחת פירות גם בצעד הבא וערכי הניקוד ממשיכים להשתנות, ולכן המצב "אינו שקט" על פי מה שהגדרנו, וכך היינו מגלים את ההפסד שעתיד לבוא תור אחד לאחר מכן ולא בוחרים בצעד ימינה.

2. להגדיר שכל עוד הסוכן צמוד לאויב (שניהם במשבצות שכנות על הלוח), ממשיכים לפתח.



נניח שאף שחקן לא אכל פרי, ומצב ניקוד השחקנים הוא 0 לכל אחד.

בנוסף, נניח כי הסוכן האדום משחק לפי היוריסטיקה $h_E(s)$ שהגדרנו בסעיף 4, המקבלת ערך מקסימלי כאשר הסוכנים צמודים.

כמו כן השחקן מפתח את עץ החיפוש עד לעומק 4 (שני מהלכים לכל שחקן).

כפי שניתן לראות בשרטוט, לסוכן האדום קיימות שתי אפשרויות, ללכת ימינה או למטה. ניתן לראות כי אם הסוכן יילך ימינה הוא יפסיד, מכיוון שהכחול יראה בעץ החיפוש שלו את המצב בו הוא חוסם את האדום והאדום מפסיד, ולכן בשני התורות הבאים הכחול ילך ימינה. אך במהלך הפיתוח הסוכן האדום לא מצליח לראות את ההפסד (כי לא מפתח לעומק מספיק גדול, ולכן ההפסד נמצא מעבר לאופק החיפוש), והוא יעדיף ללכת ימינה כדי להיצמד לאויב, כי זה מצב יוריסטי עם ערך גבוה יותר.

אם היינו משתמשים בפתרון שהצענו לעיל, היינו ממשיכים לפתח כי אנו צמודים לאויב, ולכן המצב "אינו שקט" על פי מה שהגדרנו, וכך היינו מגלים את ההפסד שעתיד לבוא תור אחד לאחר מכן ולא בוחרים בצעד ימינה.

12.

נניח כי ערך ה-$minimax$ שחזר מהאלגוריתם הוא V. ונניח כי מהצומת ממנו אנו מפתחים את עץ ה-$minimax$ יש B מהלכים חוקיים (כלומר, B בנים כך שכל בן מתאר את הצעד הראשון באסטרטגיה כלשהי).

ניתן לשפר את יעילות ההרצה החוזרת באמצעות שינויים פשוטים באלגוריתם כך :

1. נריץ שוב את אלגוריתם ה-$\alpha-\beta$ עם החסמים $\alpha=\beta=V$.

2. אם סיימנו לפתח את תת העץ של אחד מ-B המהלכים, וקיבלנו את הערך V, נפסיק את האלגוריתם ונבצע את המהלך הזה.

3. אם פיתחנו את כל תתי העצים של $B-1$ המהלכים ולא קיבלנו באף אחד מהם את הערך V, נבצע את הצעד האחרון (ללא חישוב של תת העץ שלו).

נסביר את הנכונות:
מכיוון שיש בידינו את ערך ה-$minimax$ שאנו מצפים לקבל, בוודאות קיימת אסטרטגיה שתחזיר לנו את הערך V. כמו כן אנו יודעים כי זה הערך הטוב ביותר שניתן לקבל.
לכן, נוכל לשלוח את האלגוריתם עם החסמים $\alpha=\beta=V$ וכל ערך ששונה מ-V (קטן מ-α או גדול מ-β), נגזום אותו, כי הוא בהכרח לא ייבחר לאסטרטגיה הסופית.
בנוסף, מכיוון שקיימת אסטרטגיה כזאת, קיים המהלך הראשון באסטרטגיה. לכן, אם קיבלנו לאחר פיתוח של תת העץ של מהלך כלשהו $minimax\ value=V$ נוכל לבחור בצעד זה. נשים לב כי זה לא בהכרח אותו הצעד שעבורו קיבלנו את הערך V בריצה המקורית, כי ייתכן שקיבלנו את הערך V ממהלך אחר, אך מכיוון שלשני המהלכים אותו ערך $minimax$ הם שקולים, וניתן לבחור בצעד זה.
כמו כן, אם פיתחנו $B-1$ מהלכים מתוך B המהלכים האפשריים ובאף אחד לא חזר הערך V, נדע כי הערך V בהכרח הגיע מהמהלך האחרון ולכן ניתן לבחור במהלך זה בלי לפתח את תת העץ שלו.
נסביר את התהגות האלגורתם במקרים השונים:

- במקרה הטוב ביותר האלגוריתם יקבל את הערך V מפיתוח תת העץ של המהלך הראשון מבין B המהלכים החוקיים, עם גיזום מקסימלי של הענפים. נשים לב כי האלגוריתם חייב לפתח לפחות תת עץ אחד כדי לקבל ערך $minimax$ כלשהו. לכן המקרה הטוב ביותר הוא לפתח רק תת עץ אחד ועם גיזום מקסימלי. בהנחה ומקדם הסיעוף הוא B והפתרון נמצא בעומק d, האלגוריתם יפתח במקרה זה $B^{\frac{d-1}{2}}$ עלים. מפתחים רק תת עץ אחד, לכן העומק היחסי הוא $d-1$.

- במקרה הגרוע ביותר הערך V מתקבל מהמהלך האחרון, ולא יהיה גיזום לכל אורך הדרך. במקרה זה האלגוריתם יפתח את תת העץ של כל $B-1$ המהלכים הראשונים וללא גיזום כלל. נשים לב כי האלגוריתם מפתח לכל היותר $B-1$ תתי עצים, ולכן לפתח את כולם ללא גיזום זה בהכרח המקרה הגרוע ביותר. בהנחה ומקדם הסיעוף הוא B והפתרון נמצא בעומק d, האלגוריתם יפתח במקרה זה $(B-1)B^{d-1}$ עלים. הסיבה לכך היא כי מפתחים $B-1$ תתי עצים, שבכל אחד הפיתוח הוא לעומק יחסי של $d-1$ וללא גיזום.

- במקרה הכללי ערך ה-$minimax$ יתקבל באחד המהלכים המרכזיים. נשים לב שהחסמים שאנו נותנים הם הדוקים, לכן ברוב המקרים נקבל גיזום ובכל פיתוח של תת עץ אנו מצפים למספר מועט של פיתוחים. לכן המקרה הכללי כנראה יהיה קרוב יותר למקרה האופטימלי.

.13

אכן ניתן לבצע גיזום לאלגוריתם זה, על פי הכללים הבאים:

- בצמתי מקסימום ומינימום נוכל לגזום לפי אותם כללים שלפיהם גוזמים באלגוריתם $Alphabeta$ רגיל.

- בצמתים הסתברותיים, נוכל לבצע גזימה דומה, אך בצורה יותר "שמרנית".
בהגיענו לצומת הסתברותי כלשהו, נסמן ב-n את מספר בניו. מתוך n הבנים של הצומת, נסמן ב-E_i את התוחלת של i הבנים הראשונים, ובנוסף נסמן ב-p_j את ההסתברות למאורע שמייצג הבן ה-j. נפריד לשני מקרים:
אם הצומת ההסתברותי הוא בן של צומת מקסימום נגזום כאשר מתקיים:

$$E_i+5\cdot\sum_{j=i+1}^{n}p_j\leq\alpha$$

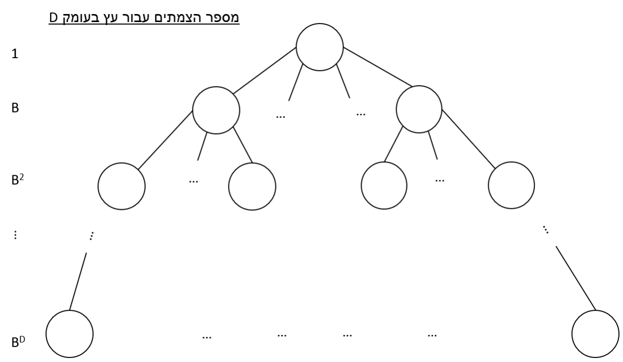
אם הצומת ההסתברותי הוא בן של צומת מינימום נגזום כאשר מתקיים:

$$E_i + (-5) \cdot \sum_{j=i+1}^{n} p_j \geq \beta$$

נסביר את הגזימה עבור צומת v_1 הסתברותי שהוא בן לצומת מקסימום. ידוע לנו כי קיים חסם α. כלומר, יש לנו את האפשרות לקבל את הערך α מצומת v_2 כלשהו. נשים לב שאם תנאי הגזימה שהצגנו מתקיים, אנו מניחים כי כל בן של v_1 שעדיין לא פותח מקבל את הערך היוריסטי הגבוה ביותר האפשרי ($h(s)=5$), ולמרות זאת אנו מקבלים ערך שקטן מ-α, שכידוע מתקבל בצומת v_2. לכן אין סיבה שנבחר ב-v_1 על פני v_2 וניתן לגזום את v_1.
נקבל הסבר סימטרי לחלוטין עבור צומת הסתברותי שהוא בן לצומת מינימום.

14.

א.



ניתן לראות על פי השרטוט כי מספר הקריאות לפונקציה היוריסטית כאשר מפתחים לעומק D הוא B^D (ככמות העלים בתת העץ). לפי הנתון, הזמן הנדרש לחיפוש בעומק D הוא $\frac{M}{B}$, כלומר הזמן הנדרש ל-B^D קריאות לפונקציה היוריסטית הוא $\frac{M}{B}$ (נתון כי שאר הפעולות זניחות).
לכן, עבור פיתוח לעומק $D+1$ נדרוש B^{D+1} קריאות לפונקציה היוריסטית, והזמן שנצטרך לשם כך הוא:

$$time(B^{D+1}) = time(B \cdot B^D) = B \cdot time(B^D) = B \cdot \frac{M}{B} = M$$

כלומר, כל זמן הריצה של השחקן עבור B המהלכים שלו, ינוצלו על חישוב המהלך הראשון, ולא יישאר זמן כלל עבור הפיתוח של $B-1$ המהלכים הבאים. לכן בהמשך יצטרך בכל מהלך לבחור שרירותית צעד חוקי כלשהו.
ב.

1. אנו מפתחים לעומק $D+1$. נזכור שבעץ האסטרטגיה אנו שומרים קשת אחת ביציאה מצומת מקסימום (אשר מתארת פעולה של השחקן) ו-B קשתות ביציאה מצומת מינימום המתארות פעולות של היריב.
לכן, נוכל למעשה להפריד את העץ לשני תתי עצים, הראשון שיחזיק רק קשתות של השחקן, והשני רק קשתות של היריב. העץ הראשון יהיה בגובה $\left\lceil \frac{D+1}{2} \right\rceil$ (כי השחקן תמיד עושה את המהלך הראשון בפיתוח של העץ), והעץ השני יהיה בגובה $\left\lfloor \frac{D+1}{2} \right\rfloor$.
סך הכל נקבל כי עבור השחקן אנו שומרים:

$$1 + B + B^2 + \ldots + B^{\left\lceil \frac{D+1}{2} \right\rceil} = \frac{\boldsymbol{B^{\left\lceil \frac{D+1}{2} \right\rceil} - 1}}{\boldsymbol{B-1}}$$

ועבור היריב אנו שומרים:

$$B+B^2+B^3+...+B^{\left\lfloor\frac{D+1}{2}\right\rfloor}=\frac{\boldsymbol{B\cdot\left(B^{\left\lfloor\frac{D+1}{2}\right\rfloor}-1\right)}}{\boldsymbol{B-1}}$$

2. בסעיף א' הוא השקיע את כל זמנו בחישוב המהלך הראשון, ולכן ב-$B-1$ המהלכים הבאים יהיה עליו לבחור שרירותית צעד חוקי כלשהי. במקרה ב' השחקן שומר את עץ האסטרטגיה שחישב במהלך הראשון, כלומר את תוכנית הפעולה המותנית, ולכן יוכל בכל מהלך לאחר מכן לבחור את הצעד ששמור בתוכנית בהתאם לאיך שיפעל היריב, וכך להבטיח לעצמו שלאחר B מהלכים יגיע לפחות לערך התוכנית המותנית כפי שחישב במהלך הראשון (לפי משפט ההבטחה של $minimax$). כלומר, במקום להיות חסר ידע לחלוטין ולבחור שרירותית הוא הולך לפי אסטרטגיה שמבטיחה לו תוצאה מסויימת.
נשים לב כי למרות שיפור המצב ייתכן שהשחקן עדיין יצטרך להתחיל לבחור שרירותית בנקודה מסוימת. שכן, עץ אסטרטגיה שמפותח לעומק $D+1$ מכיל למעשה $\left\lceil\frac{D+1}{2}\right\rceil$ מהלכים קדימה של השחקן, מפני שכל רמה בעץ מכילה את כל האפשרויות לפעולה אחת פוטנציאלית (של השחקן או של היריב). לכן אם $B<\left\lceil\frac{D+1}{2}\right\rceil$ נקבל כי לאחר $\left\lceil\frac{D+1}{2}\right\rceil$ מהלכים, הוא יגיע למהלך האחרון ששמור לו בתוכנית הפעולה המותנית ומכאן יצטרך לבחור שרירותית למשך $B-\left\lceil\frac{D+1}{2}\right\rceil$ המהלכים הבאים.

3. מקרה בו סוכן זה יהיה טוב יותר מסוכן $minimax$ רגיל הוא כאשר החיפוש לעומק נוסף חושף אופציה שלא הייתה מתגלה בפיתוח לעומק D. למשל, ייתכן כי הפיתוח לעומק $D+1$ חושף אופציה לניצחון, שלא הייתה מתגלה בפיתוח לעומק D, במצב זה ברגע שהסוכן גילה מסלול לניצחון, בגלל משפט ההבטחה של ה-$minimax$, ניתן ללכת במסלול זה ובוואות לנצח. אם הסוכן היה מפתח לעומק D בלבד (כמו סוכן $minimax$ רגיל), יכול להיות שהיה בוחר במהלך אחר שלא בהכרח מוביל לניצחון, כי ערכו היוריסטי היה גבוה יותר, ומהלך זה היה מונע ממנו את הניצחון בהמשך.
מקרה בו סוכן זה יהיה פחות טוב מסוכן $minimax$ רגיל הוא כאשר החיפוש לעומק $D+1$ מונע ממנו לגלות מצבים טובים יותר בהמשך. למשל, ייתכן כי אם היה מפתח לעומק D בלבד (כמו סוכן $minimax$ רגיל), היה מגלה שני מהלכים לאחר מכן כי קיימת אסטרטגיה לניצחון. כלומר, הפיתוח לעומק D השאיר בפניו את האפשרות לפתח לעומק D בכל מהלך נוסף מבין B המהלכים הבאים, ובפרט שני מהלכים לאחר מכן. לכן ייתכן כי בפיתוח זה הוא גילה אסטרטגיה לניצחון בעומק D (כלומר עומק $D+2$ ביחס לחיפוש הראשוני). אך מכיוון שפיתח לעומק $D+1$ בתור הראשון מבין B המהלכים, לא גילה את הניצחון הנ"ל, כי עומק החיפוש לא הספיק, ולא נשאר לו זמן לפיתוחים בהמשך. לכן לא גילה את הניצחון ונשאר עם האסטרטגיה הפחות טובה.

15.

עבור שחקן המינימקס ניסינו להשתמש ביוריסטיקה שהגדרנו בסעיף 4. לאחר הרצות רבות על לוחות שונים וכיוונון של המשקלים של כל רכיב, הגענו למסקנה שהרכיב $h_F(s)$ (כפי שהוגדר בסעיף 4) אינו מביא את התוצאות הרצויות. לכן החלטנו להחליפו ברכיב הבא:

$$h_I(s)=1-\frac{1}{md(s,initial\_position)}$$

כלומר, $h_I(s)$ נותן ערך יוריסטי גבוה יותר למצבים בהם אנו רחוקים יותר מהמיקום בו התחלנו את המשחק, וערך יוריסטי נמוך יותר למצבים בהם אנו קרובים יותר למיקום ההתחלתי. ההגיון מאחורי יוריסטיקה זו הוא למנוע מצב בו היריב יכול לחסום אותנו מכך שנשארנו באזור קטן סביב נקודת ההתחלה.
שאר שלושת הרכיבים זהים לחלוטין כפי שהוגדרו בסעיף 4: $h_\delta(s),h_E(s),h_S(s)$.

לאחר הרצות רבות עם היוריסטיקה המבוססת על ארבעת הרכיבים הנ"ל גילינו את המשקלים שהביאו לתוצאות הכי טובות וסך הכל קיבלנו:

$$h(s) = 0.4 \cdot h_\delta(s) + 0.3 \cdot h_E(s) + 0.2 \cdot h_I(s) + 0.1 \cdot h_S(s)$$

.16

נסביר את אופן פעולתו של שחקן התחרות שלנו:
שחקן התחרות הוא סוכן המחפש לפי אלגוריתם $AlphaBeta$, עם מגבלת זמן גלובלית.
כדי להתמודד עם מגבלת הזמן הגלובלית שהוקצבה למשחק השתמשנו באלגוריתם חלוקת הזמן שתואר בסעיף 10 .
נבחין כי בהסתמך על הדיון בסעיף 2, כאשר לא קיים מסלול בלוח בין שני הסוכנים, האסטרטגיה האופטימלית היא לשחק כ-$Simple\ Player$. כלומר, לכסות כמה שיותר שטח בלוח.
לכן, בכל תור של שחקן התחרות שלנו לפני פיתוח עץ המינימקס אנו מבצעים BFS על לוח המשחק על מנת לבדוק האם קיים מסלול בלוח בינינו ליריב.
אם לא קיים מסלול אנו משחקים באופן דומה לאופן פעולתו של ה-$Simple\ Player$. הפונקציה היוריסטית בה אנו משתמשים במקרה זה היא $h_S(s)$ כפי שהוגדרה בסעיף 4.
נרמלנו את כל הערכים האפשריים שיכולים להתקבל עבור עלה בעץ המינימקס להיות בין 0 ל-1 כאשר 0 מייצג הפסד ו-1 מציין נצחון ושניהם מתקבלים ממצבי מטרה בלבד. את הפונקציה היוריסטית נרמלנו כך שלכל s יתקיים $0 < h_S(s) < 1$.
בצורה כזו ערכו של מצב המייצג נצחון יהיה גבוה מהערך המקסימלי שמחזירה $h_S(s)$, וערכו של מצב המייצג הפסד יהיה נמוך מהערך המינימלי.
לכן אופן פעולה זה יותר מתוחכם מה-$Simple\ Player$ אשר בכלל לא מפתח עץ חיפוש ורואה רק מהלך אחד קדימה.
אם קיים מסלול בינינו לבין היריב, אנו משחקים לפי היוריסטיקה הבאה:

$$h(s) = 0.8 \cdot h_\delta(s) + 0.2 \cdot h_E(s)$$

כאשר $h_\delta(s)$ ו-$h_E(s)$ הן כפי שהוגדרו בסעיף 4.
ביצענו מספר רב של משחקים והגענו למסקנות הבאות:

- עומק חיפוש יותר חשוב מיוריסטיקה מושלמת ולכן עדיף יוריסטיקה קלה לחישוב כדי שנצליח לפתח לעומק גדול יותר בזמן קצוב.
- מספר רב של אלמנטים ביוריסטיקה "מבלבל" את הסוכן ועלול לפגוע בביצועיו. לכן החלטנו ללכת על יוריסטיקה המכילה רק שני רכיבים שבאופן אמפירי התגלו כחשובים ביותר.
- הרכיב היוריסטי של הפרש ניקוד $h_\delta(s)$ חשוב יותר מהרכיב של מרחק מהאויב $h_E(s)$ מפני שבסופו של דבר השחקן המפסיד הוא השחקן בעל הניקוד הנמוך ביותר ולא בהכרח זה שנחסם.

המסקנות הללו הובילו אותנו לבחור ביוריסטיקה המתוארת.
עבור המקרה בו לא קיים מסלול בין שני הסוכנים אך עדיין יש פירות על הלוח, אנו משחקים רק לפי הרכיב $h_\delta(s)$ שכן במקרה כזה אין חשיבות למרחק מהיריב.
בכל המקרים אנו מבצעים סידור ילדים בעץ המינימקס על פי ערכי היוריסטיקה הרלוונטית לאותו מקרה, על מנת למקסמם את מספר הגיזומים וכך להצליח להגיע לעומקים גדולים יותר בעץ בזמן קצוב.

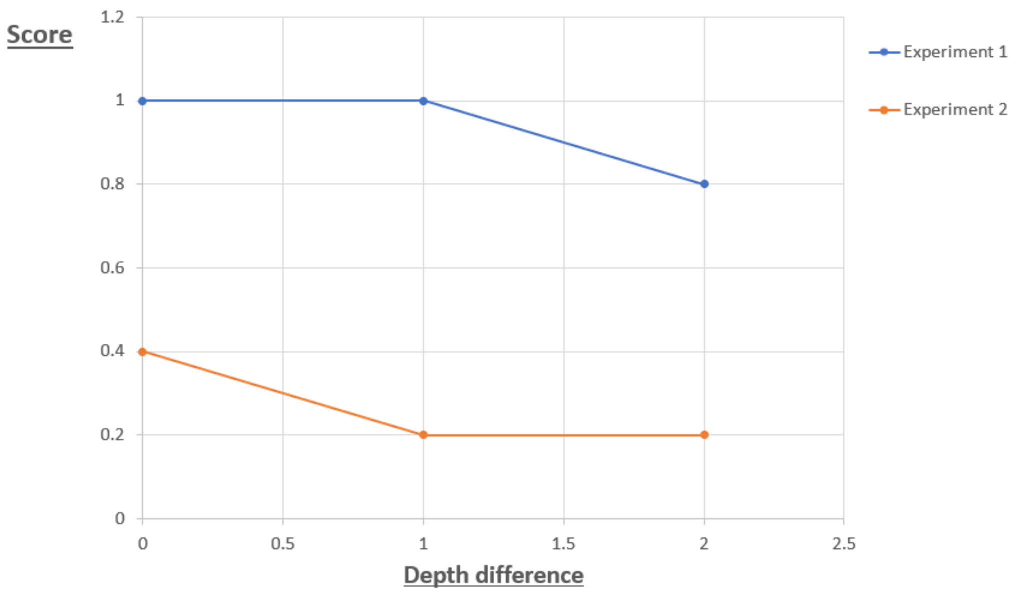
.17

עבור המקרה של הגבלת זמן גלובלי, הדרך בה ניהלנו את זמן ריצת הפונקציה $make\_move$ היא לפי האלגוריתם שתיארנו בסעיף 10, כך שמסגרת הזמן לכל תור היא הערך שמקבל $time\_frame$.
עבור המקרה של הגבלת זמן לתור, הפונקציה $make\_move$ רצה למשך הגבלת הזמן של שניתנה.
בשני המקרים, במהלך הזמן הנתון ביצענו העמקה הדרגתית כפי שלמדנו בהרצאה. אנו מריצים את אגלוריתם ה-$minimax$ עם הגבלת עומק הולכת וגדלה כאשר בכל איטרציה אנו שומרים את הצעד שהוביל לערך ה-$minimax$ הגבוה ביותר. כאשר נשאר לנו 0.01 שניות מסוף מסגרת הזמן שהוקצתה לתור זה, אנו מפסיקים את החיפוש ומחזירים את הצעד האחרון ששמרנו. הסיבה לחלון הזמן של 0.01 שניות היא כדי לאפשר לפונקציה להחזיר את הערך ולצאת בהצלחה לפני סוף מסגרת הזמן.

.18

הרצנו מספר רב של משחקים בין שני הסוכנים וכפי שציפינו סוכן ה-$Alphabeta$ ניצח את סוכן ה-$Minimax$ ברוב המוחלט של המשחקים (כ-88% מהמשחקים). תוצאה זאת מתאימה לציפיותנו מכיוון שסוכן ה-$Alphabeta$ מבצע גיזום ענפים ולכן מספיק לפתח את עץ החיפוש שלו לעומקים גדולים יותר מאשר סוכן ה-$Minimax$. ולכן יש לו תמונה נאמנה יותר של המציאות, ואופציה לבחור אסטרטגיה טובה יותר. בנוסף, המקרים בהם ניצח סוכן ה-$Minimax$ נבעו מכך שבתחילת המשחק קיבל משמעותית יותר פירות קרובים למיקומו מאשר שקיבל סוכן ה-$Alphabeta$.

.19



תחילה נסביר את תוצאות הניסוי הראשון בהתבסס על גרף 1 (כחול). ניתן לראות שסוכן ה-$HeavyAB$ ניצח את סוכן ה-$LightAB$ בכל המשחקים בשני החלקים הראשונים של הניסוי וברוב המשחקים (4 מתוך 5) בחלק השלישי של הניסוי. את הנצחונות ניתן להסביר בכך שהיוריסטיקה של סוכן ה-$HeavyAB$ מתוחכמת יותר משל סוכן ה-$LightAB$ ומכילה יותר פרמטרים של המשחק ולכן נותנת לו יתרון משמעותי בקביעת האסטרטגיה. את הירידה הקלה בחלק השלישי של הניסוי ניתן להסביר בכך שהגדלנו את עומק החיפוש של סוכן ה$LightAB$ ובכך נתנו לו לראות יותר מהלכים קדימה ולכן קיבל יתרון מסוים, אך לא משמעותי מספיק כדי להתגבר על היוריסטיקה של סוכן ה-$HeavyAB$.
כעת נסביר את תוצאות הניסוי השני בהתבסס על גרף 2 (כתום). ניתן לראות שסוכן ה-$HeavyAB$ הפסיד ברוב המשחקים לסוכן ה-$LightAB$ כאשר בחלק הראשון של הניסוי הוא הפסיד במעט יותר מחצי מהמשחקים (3 מתוך 5) ואילו בחלק השני והשלישי הפסיד באפילו יותר משחקים (4 מתוך 5). את ההפסדים ניתן להסביר בכך שעומק החיפוש שנתנו לסוכן ה-$HeavyAB$ קטן מדי מכדי שערך יוריסטי ישקף בצורה נאמנה את המציאות. לכן בחלק הראשון שני הסוכנים מתפקדים כמעט בצורה זהה ואכן תוצאות המשחקים התחלקו כמעט חצי חצי בין שניהם. לעומת זאת, בשני החלקים הנותרים של הניסוי נתנו יתרון לסוכן ה-$LightAB$ מפני שפיתח לעומק גדול יותר, ואכן ניצח ביותר משחקים.
לסיכום, תוצאות שני הניסויים משקפים את ציפיותנו וניתן להסיק מהם שככל שעומק החיפוש יותר גדול כך גדלה החשיבות של יוריסטיקה מוצלחת.