**בינה תרגיל בית 2:**

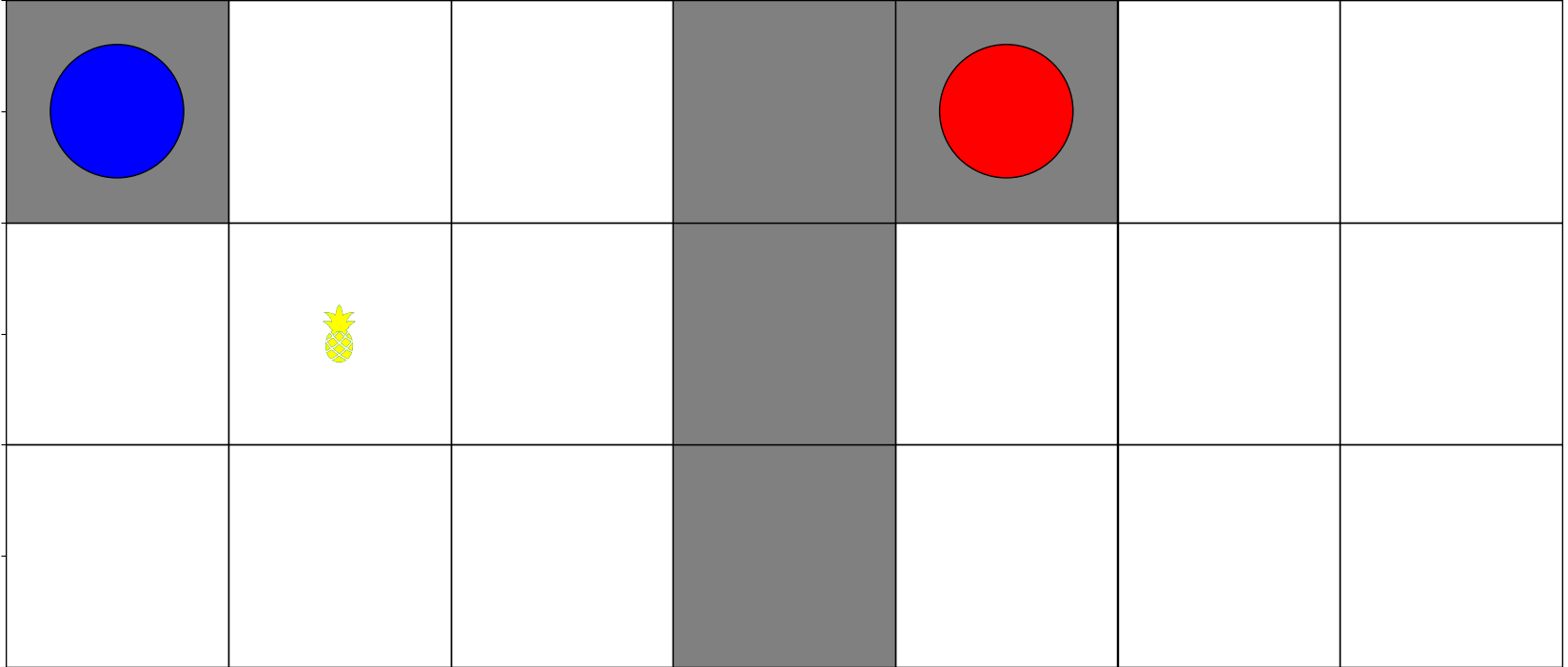
**חלק א**:

האסטרטגיה של השחקן היא להגיע למשבצות מהן יש פחות אפשרויות להתקדם, אך יש לפחות אפשרות אחת.

יתרון של שיטה זאת היא שעל ידי היצמדות לקירות, השחקן בעצם חוסם לעצמו פחות שטח, ובכך ממקסם את כמות התורות שיוכל לשחק לפני שייתקע ויקבל את העונש של סיום המשחק.

חסרון של שיטה זאת שהוא לא מתייחס לפירות, שעלולות להיות שוות יותר נקודות מאשר העונש על סיום המשחק, לכן שחקן אשר ייקח יותר פירות במחיר של קבלת העונש על סיום המשחק עלול לנצח.

נסתכל על הלוח הבא:



כאשר השחקן האדום הוא השחקן ה . במצב כזה, השחקן האדום לא יכול להשיג פירות, ולכן האסטרטגיה הכי טובה שלו היא לדחות כמה שיותר את סיום המשחק כדי לא לקבל את העונש. נשים לב שבלוח זה השחקן הכחול לא יכול להפריע לשחקן האדום, כי הם מופרדים, ולכן האסטרגיה של השחקן האדום, שהיא להיצמד כמה שיותר לקירות על מנת לא "לבזבז שטח", היא האסטרטגיה האופטימלית במצב זה.

יתרון של היוריסטיקה היא שהיא נותנת ציון יותר טוב למצבים שקרובים יותר להשגת פרי, שזה מאוד הגיוני כי קרבה לפרי צפויה להוביל בעתיד לקבלת נקודות, מה שמקרב לניצחון.

חיסרון של היוריסטיקה היא שהיא לא לוקחת בחשבון משבצות בהן השחקן לא יכול לעבור, מה שעלול לגרום ליוריסטיקה להתייחס לפרי שהשחקן אינו יכול להגיע אליו כפרי קרוב, כאשר יש פרי אחר שכן ניתן להגיע אליו.

נגדיר יוריסטיקה חדשה בצורה הבאה:

* 1. כפי ראינו בסעיף הקודם, תשתמש ביוריסטיקה המוגדרת על ידי 1 חלקי מרחק מנהטן המינימלי לפרי בלוח.
  2. יחס הפוך לסכום המרחקים מהפירות, יוריסטיקה זו תוודא שהשחקן ינסה להתקרב לכל הפירות ככל הניתן.
  3. בדומה למה שהוגדר בתרגיל הרטוב:
     1. אם לשחקן אין לאן לנוע במצב – 1-
     2. אם לשחקן יש 4 כיווני תזוזה – 0
     3. אם לשחקן יש 3 כיווני תזוזה – 1
     4. אם לשחקן יש 2 כיווני תזוזה – 2
     5. אם לשחקן יש 1 כיווני תזוזה – 3
  4. הפרש הניקוד בין 2 השחקנים.

\*ניתן להוסיף משקולות לכל הרכיבים

לדעתנו יוריסטיקה זו עדיפה על היוריסטיקה של SimplePlayer מכיוון שהיא מכילה אותה, והוספנו לה עוד רכיבים שימושיים. למשל SimplePlayer איננה מתייחסת לפירות כלל, וקיימים מקרים בהם עדיף לקבל מיוריסטיקה זה עונש בשביל להתקרב לפרי.

1. אסטרטגיית Minimax לא מתאימה עבור משחק של יותר משני שחקים.

א. החסרונות של שיטה זו הם שהאלגוריתם מניח שמתקיים משחק סכום 0, כך ששאר השחקנים תמיד מנסים להרע לו גם אם זה על חשבונם. לכן קיימים מקרים בהם Minimax תניח ששחקן יריב כלשהו יעדיף לפגוע בו מאשר בשחקן היריב השני, על אף שזו תהיה אסטרטגיה טובה יותר לפעולה. לכן Minimax לא יחזה בצורה מיטבית את מהלכי השחקנים היריבים.

ב. אסטרטגיה חלופית מתאימה תהיה לשנות את Minimax, בצורה כזאת שבחירת הצעד הבא תהיה הצעד בו השחקן יוכל לבצע את הפגיעה הגדולה ביותר באחד השחקנים. עבור שחקן i בכל צומת בעץ השחקן יבחר בפעולה אשר שתמקסם את:

אסטרטגיה זו הגיונית, משום שכאשר תור השחקן ה-i לשחק, מטרתו היא לצבור פער כמה שיותר גדול משאר השחקנים, ולאו דווקא מהשחקן המשתמש באסטרטגיה (בה"כ שחקן 1).

לכן ייתכן שיש מצב בו לשחקן יריב (בה"כ שחקן 2) עדיף לפגוע בשחקן יריב אחר (בה"כ שחקן 3) ולא בשחקן 1, כי זה עדיף לו (למשל שחקן 3 מוביל ושחקן 2 מקום שני כרגע), לכן ההנחה ששחקן יריב תמיד ינסה לפגוע בשחקן 1 אינה נכונה. בשימוש באסטרטגיה הזו, שחקן מוביל תמיד ינסה להגדיל את הפער כמה שיותר, ושחקן שאינו מוביל ינסה לצמצם פער כמה שיותר, ולכן זו אסטרטגיה יותר מתאימה למשחק עם יותר מ-2 שחקנים.

6.

א) כפי שראינו בהרצאה, זמן הריצה של alpha-beta לפחות יותר טוב מ-minimax, כי מחפשים על אותו עץ חיפוש רק ש"גוזמים" תתי עצים מסויימים ולא מחפשים בהם.

ב)לפי משפט ההבטחה שראינו בהרצאה, ערך ה-minimax של alpha-beta הוא אופטימלי, ולכן שווה לערך ה minimax של אלגוריתם Minimax, שגם הוא אופטימלי.

7.

א) כפי שראינו בהרצאה, alpha-beta עם סידור ילדים משפר את זמן הריצה, כי למעשה ה"ילד" עם ערך המינימקס הטוב ביותר תמיד יפותח ראשון.

ב)גם עבור alpha-beta עם סידור ילדים, ערך ה-minimax יישאר זהה, מכיוון שסידור הילדים משפיעה רק על סדר פיתוח הילדים, ולא על הילד שיבחר בסופו של דבר, לכן אין שינוי בערך ה-minimax.

8. Anytime contract היא וריאציה של Minimax המוגבלת בזמן ריצה, ומטרתה לבצע Minimax עם עומק כמה שיותר עמוק בזמן הניתן. לכן מבצעים העמקה הדרגתית: מריצים כל פעם Minimax עם הגבלת עומק המאותחלת ל-1, וכל עוד נשאר זמן, מעלים את העומק ב-1 ומנסים להריץ עם הגבלת העומק החדשה. בסוף מחזירים את התוצאה של ההרצה עם העומק הכי גדול אליו הצלחנו להגיע במגבלת הזמן הנתונה.

9. הבעיה של העמקה הדרגתית המוזכרת ההרצאה היא בעיית האיטרציה האחרונה- מפני שזמן הריצה גדל אקספוננציאלית, זמן הריצה של האיטרציה האחרונה עלול להיות גדול יותר משמעותית מסך כל הזמן שלקח עד איטרציה זו, ומפני שלא תסיים, כל הזמן שייקח לריצתה למעשה יבוזבז.

הפתרון המוצע לבעיית האיטרציה האחרונה היא לשמור את ערכי ה המעודכנים של הצמתים בדרגה הלפני אחרונה שכן הספיקו להיות מפותחים, ולהשוות אותם לערכים שחושבו באיטרציה אחת לפני, ולקחת מביניהם את הכי טוב. בצורה זאת גם אם האיטרציה אחרונה לא הסתיימה לגמרי, עדיין הדברים שהיא כן הספיקה לחשב נלקחים בחשבון.

10. במשחק הנתון, התורות לפני היעלמות הפירות מהלוח הם התורות המשמעותיים ביותר, ולכן אסטרטגיה טובה עשויה להיות לחלק זמן משמעותי לתורות אלה, למשל מה- . נשים לב שזוהי הנחה הגיונית, כי גם חישובי ה של כל מצב לאחר היעלמות הפירות צפויים להיות מהירים יותר, ולכן התורות לאחר היעלמות הפירות יכולים להגיע לעומק יותר גבוה בזמן הרבה יותר קצר ( זוהי גם תופעה שראינו בניסויים).

נסמן את מספר התורות עד היעלמות התורות ב , ואז נקצה ל- התורות הראשונים .

לאחר היעלמות הפירות, נוכל להשתמש בנוסחה הנאיבית שהוצעה בסעיף הקודם, אך עם שיפור:

נסמן את להיות הזמן שנשאר. במקרה הכי גרוע, כל תור ייקח . לכן בתחילת כל תור , נגדיר:

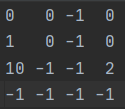
, ונקצה לתור הנוכחי .

בצורה זאת, במקרה הכי גרוע, כל תור יקבל חלוקה שווה של הזמן שנשאר, אבל במקרה שתור מסויים ייקח פחות זמן מהמגבלה שלו, זה ישאיר לתורות הבאים יותר זמן.

11. בעיית האופק נובעת מכך שעומק החיפוש מוגבל, מה שמונע משחקן לראות שהצעד שהוא יעשה הוא למעשה צעד לא נכון, כי צעד זה יוביל למצב עם ערך יוריסטי נמוך רק בעומק גבוה יותר מהגבלת העומק.

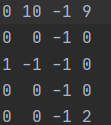
הפתרון המוצע בהרצאה הוא לבדוק מקרים שבהם יש שינוי חד בערך היוריסטיקה, ובמקרים אלו להמשיך לחפש לעומק גדול יותר מההגבלה לעומק. במשחק שלנו, מקרה זה יכול לבוא לידי ביטוי למשל אם השחקן מזהה שהוא יכול לקחת פרי השיא העומק שלו, אך צעד אחד אחרי הוא לא יכול לזוז ויקבל את העונש. שימוש בפתרון המוצע יאפשר לשחקן לבדוק לעומק יותר גבוה לאחר לקיחת הפרי, ואז הוא יראה שהוא גם יקבל את העונש מיד לאחר לקחת הפרי, ולכן יבחר לא לעשות את הצעד הזה.

דוגמא למצב כזה במשחק: נסתכל על הלוח הנ"ל:



נניח ששחקן 1 מוגבל לחשוב עד עומק 1. בעומק 1 נראה כי הצעד למטה הוא הצעד הטוב עבורו, כי זה יגרום לו לקחת פרי ולניקוד שלו לעלות. אך למעשה, צעד זה יפסיד לו את המשחק כי מיד לאחר מכן הוא יהיה תקוע ושחקן 2 יכול לזוז 2 תורות, ולכן ינצח. לעומת זאת, אם השחקן יחשוב עד עומק 3, הוא יראה כי למעשה כל צעד אחר מנצח לו את המשחק כי אז יהיה לו 3 תורות סך הכל שהוא יכול לזוז כשחקן 2 יכול לזוז רק 2 צעדים, ולכן שחקן 1 ינצח.

דוגמא נוספת: נסתכל על הלוח הנ"ל:



נניח ששחקן 1 חושב לעומק 3. מבחינתו, במהלך הראשון גם לזוז למעלה וגם למטה יוביל אותו לאותו ניקוד. לכן יכול לבחור באקראיות למשל את המהלך למטה. אך אם ייבחר לזוז למטה הוא יפסיד, כי לאחר 4 תורות לכל שחקן, לשחקן 2 יהיה 9 נקודות יותר. לעומת זאת, אם שחקן 1 יזוז למעלה, הוא ינצח, כי יסיים עם נקודה יותר.

12. לאחר הרצה זו ניתן להריץ את אלגוריתם אלפא-בטא עם אתחול אלפא ובטא לערך המינימקס שהתקבל קודם. מכיוון שאנו יודעים שקיים עלה שזהו ערכו ניתן להשתמש במידע זה על מנת לגרום את העץ בצורה יותר אגרסיבית מאשר היינו מקבלים אילו היינו משתמשים בערכים הדיפולטים (אינסוף ומינוס אינסוף). במקרה הטוב ביותר, יהיו מעט מאוד עלים בעלי ערך זה ונמצא אותם במהירות, ובמקרה הרע ביותר כל העלים יהיו בעלי אותו ערך ואז אנחנו נבצע מינימקס ללא גיזום. במקרה הכללי יתבצע גיזום כלשהו, ככל הנראה יותר מאשר היה מתבצע אם היינו משתמשים באתחול הרגיל.

13. עבור צמתים שאינם הסתברותיים, תנאי הגזימה יהיו זהים לאלגוריתם אלפא בטא הרגיל. עבור צומת הסתברותי, נניח שפיתחנו כבר את הצמתים עם ערכיים יוריסטיים כשכל אחד נבחר בהסתברות . במקרה כזה, הערך הכי גבוה שהתוחלת של הצומת ההסתברותי יכול להיות:

*וזאת בהנחה שכל שאר הצמתים יקבלו את הערך הכי גבוה. הערך הכי נמוך שנוכל לקבל יהיה:*

*לכן, נוכל לגזום לפי הערכים האלה (אם גבוהים או נמוכים יותר מ- או ). זוהי למעשה גישה שמרנית- מניחים בכל מצב ששאר החישוב שנותר לשחקן ההסתברותי ייתן את התוצאה הכי גבוהה או הכי נמוכה האפשרית.*

14.

א) לפי הנתון, פיתוח כל הצמתים עד עומק לוקח זמן. לכל צומת בעומק יש ילדים, מפני שמקדם הסיעוף הוא . לכן, לכל מסלול שנבדוק בעומק , הוא ייקח פי יותר זמן מאשר אם היינו בודקים אותו עד עומק . לכן סה"כ הזמן שייקח התור הראשון הוא , ולכן לא יישאר זמן עבור שאר התורות.

ב.

1) במקרה כזה נצטרך לשמור את כל האפשרויות של היריב, כי לא ניתן להניח את האסטרטגיה שלו. לכל מהלך של היריב, השחקן היצטרך לשמור את "התגובה" שלו, כלומר צעד אחד.

מספר התורות שהיריב יקבל ב תורות הוא . בתור הראשון יהיה לו אפשרויות, בתור הבא וכך הלאה. לכן סה"כ מספר המהלכים האפשריים של היריב הוא .

לכל מהלך של היריב נצטרך לשמור מהלך עבור השחקן עצמו, לכן כמות המהלכים שנצטרך לזכור עבור השחקן זהה. (עד כדי פלוס 1 לתור הראשון או הורדת השכבה האחרונה אם זוגי).

2) הוספה של זיכרון את האסטרטגיה משפרת את המצב לאחר סעיף א', כי בסעיף א' הוא יכול לדעת צעד אחד בלבד לאחר החישוב, ולא יישאר לו זמן לחישוב צעדים נוספים. לעומת זאת, שחקן ב' לאחר החישוב כבר יוכל לדעת את הצעד שלו ל- המהלכים הבאים.

3) אם יווצר "אפקט העומק" בעומק שאפשר לשים לב אליו רק בעומק , שחקן החושב רק לעומק עלול לבצע טעות בלתי הפיכה, לעומת זאת שחקן שחושב לעומק יוכל לדעת על הבעיה ולפעול בהתאם.

לעומת זאת, אם מקור הבעיה הוא בעומק בין ל- השחקן של סעיף ב' לא יראה אותה במהלך החישוב, ולכן יפעל צעדים כשהוא בכלל לא מודע לבעיה. לעומתו, שחקן minimax רגיל אמנם לא יראה את הבעיה בהתחלה, אך לאחר מספר צעדים קטן מ-, יוכל לשים לב לבעיה ולנסות "למזער נזקים".

15. הגדרנו את היוריסטיקה בעזרת ארבעת הרכיבים הבאים:

כאשר היוריסטיקה המלאה מוגדרת בצורה הבאה:

כאשר הפירות זמינים על הלוח, הגדרנו את היוריסטיקה כך:

וכאשר אין פירות זמינים על הלוח היוריסטיקה מוגדרת כך:

16. הגשנו את שחקן התחרות עם אותה יוריסטיקה שהגדרנו בסעיף 15. מכיוון שהשקענו זמן רב בטיוב פונקציה זו וראינו שהיא משיגה תוצאות טובות מאוד, לא ראינו סיבה לשנות אותה לחלוטין לחלק התחרות. החלק העיקרי בו טייבנו את אלגוריתם התחרות היה מימוש יעיל של "בעיית האיטרציה האחרונה", בו בניגוד לגישה שניסינו בשאר החלקים, הערכנו את זמן הריצה שמוערך של האיטרציה עם עומק d לפי האיטרציות הקודמות לפני ביצוע החיפש. בצורה זו חסכנו בזמן הריצה הכללי זמן רב שהיה מושקע בצורה לא אופטימלית.

17. במקרה של זמן מוגבל לתור – לקחנו מרחב ביטחון ליציאה מהפונקציה search בזמן. בכל כניסה רקורסיבית ל- search, בדקנו שזמן הריצה של הפונקציה הכולל לא עבר את . אם עבר, נזרקת חריגה, שגורמת ליציאה מהפונקציה search ונתפסת בפונקציה make\_move. במקרה זה נשלח לתור הבא את הצעד הטוב ביותר שנמצא בעומק האחרון שהצליח להסתיים.

במקרה של הזמן הגלובלי – ההסבר דומה להסבר שניתן בסעיף 10, אך חלוקת הזמן לאחר היעלמות הפירות קצת שונה. בחרנו בסיבה זו כי היא יותר פשוטה למימוש והגיעה לתוצאות טובות יותר. תחילה נתנו לשחקן להמשיך לשחק כרגיל עד שנשאר פחות משנייה אחת. מהרגע שנשאר שנייה אחת, הקצינו לכל תור 0.04 שניות. ראינו מניסויים אמפיריים כי זה מספיק בממוצע לחשוב לעומק 10, שנראה לנו מספיק. באסטרטגיה זו ממשיכים עד שנשאר 0.04 שניות, כלומר לא מספיק לתור אחד, ואז מקצים 0.002 שניות לכל תור, שזה הספיק לחשוב עד בערך עומק 4. בלחץ של זמן, היה לנו חשוב לעשות מהלכים כמה שיותר מהירים על מנת לא לקבל את העונש, לכן מספיק תורים מהירים.

18. התוצאות שקיבלנו לאחר הרצות רבות של ניסויים במפות שונות, ומגוון רחב של תרחישים (כולל השפעת השחקן המתחיל, והשפעת הגבלת הזמן), הם ששחקן האלפא-בטא מנצח ברוב מוחלט של המשחקים. פער זה נובע מכך שבמימוש שלנו יש השפעה ניכרת ליכולת השחקן בעומק החיפוש אותו הוא מבצע. ראינו שעומק החיפוש של שחקן האלפא-בטא גדול בכ-40% יותר משחקן המינימקס ולכן היתה לו עדיפות ברורה עליו. פער זה כפי שצפינו נובע מכך שבאלפא-בטא ישנו תוספת גיזום ענפים, ולכן האלגוריתם מייעל זמן זה שנחסך בהעמקת החיפוש.

19.

**ניתוח התוצאות**:

אנו יכולים לראות בתוצאות בשני הניסיים שנקודת הפתיחה בהם לשני השחקנים אותה דרגת העמקה, יש יתרון ברור לשחקן עם היוריסטיקה הכבדה. בנוסף בשני הניסויים אנו רואים כי ככל שנעלה את דרגת העמקה של השחקן עם היוריסטיקה הפשוטה הפערים מצטמצמים.

ההבדלים בין 2 הניסויים הם שבניסוי בו דרגת העמקה השחקן בעל היוריסטיקה המורכבת היא 3, הוא תמיד מצליח יותר מהשחקן השני והפערים רק קטנים. לעומת זאת בניסוי השני עם דרגת העמקה 2, אנו רואים שחל מהפך!

אנו משערים שהבדלים אלו מייצגים את היתרונות של חיפוש עמוק יותר לעומת יוריסטיקה טובה יותר, לדעתנו ניסוי זה מראה שנעדיף לבצע חיפוש עמוק יותר לעומת שחקן בעל יוריסטיקה טובה יותר. ייתכן כי ניתן להגדיר שחקן עם יוריסטיקה טובה משלנו, אך אפשרות סבירה היא שבהינתן הפרש גבוה מספיק בדרגת העמקה בין החיפושים של שני השחקנים, הפערים יצטמצמו. מכאן נסיק שיש ערך רב במימוש יעיל ונכון של האלגוריתם בכדי שיעמוד במגבלת הזמנים יפתח לעומק גבוה ככל הניתן.