**Intro to AI HW2 – Dry:**

**לתקן גוף ראשון יחד לגוף ראשון רבים**

**סטודנטים:**

עמית ענבר - 315836569

לינוי גנטי - 208536284

1. האסטרטגיה של שחקן זה היא להימנע מבורות בעומק 2 (2 מהלכים מראש)

ולהיצמד לקירות (-1 במפה) .

יתרונות:

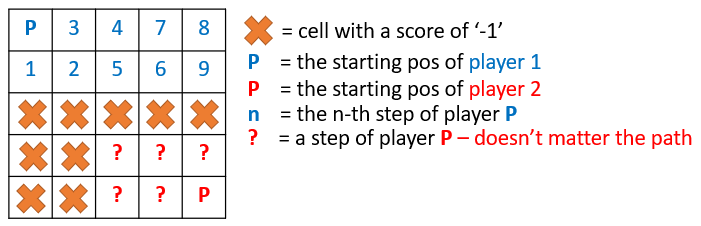
1. ממלא מפה שלא מגיעה עם בורות בצורה מיטבית אם היה משחק בה לבד.
2. נמנע מלעבור למשבצת אם אין ממנה מהלכים אפשריים (נמנע מתאים **שמסווגים** כבורות) כל עוד יש לשחקן אפשרות נוספת (ייתכן והמהלך היחיד שנותר הוא להיכנס לבור)

חסרונות:

1. "ראייה" של 2 מהלכים קדימה סה"כ והיא אינה מתחשבת במהלכים אפשריים של

ה-"אויב".

1. גם עם "ראייה" של 2 מהלכים קדימה ניתן להיכנס לבורות בעומק של יותר ממש מ-1:
2. דוגמא:



הסבר:

נבחין כי במקרה זה, לא משנה מה השחקן האדום (היריב) עושה, הכחול תמיד ינצח כיוון שהשחקן

הפשוט יימלא את כל התאים בצד שלו (כממוספר) בעוד שהאדום ייחסם לכל היותר אחרי 5 צעדים מצידו.

1. יתרונות:

* דירוג גבוה יותר של מסלולים שעוברים דרך פירות ובכך משפר את הסיכוי לקבל ניקוד גבוה.
* שימוש במידע נוסף מלבד חיפוש עיוור (היוריסטיקה מיודעת).

חסרונות:

* ייתכן ולא קיים מסלול בין המיקום הנוכחי לפרי (יש חסימה בכל מסלול אפשרי לפרי).
* יכולים להיות קירות (-1) החוסמים את כל המסלולים מהמיקום הנוכחי לפרי שאורכם הוא מרחק מנהטן מינימלי מהמיקום הנוכחי לפרי כך שיכול להיווצר מצב בו אנחנו רק "מתרחקים" במסלול הקיים מהמיקום הנוכחי לפרי. דוגמא:



* ההיוריסטיקה יכולה "להתבדות" אם קיים מסלול עם מרחק מנהטן אך הפרי נעלם לפני.
* יכול להיווצר מצב בו "נחתוך" את השטח שהשחקן שלנו יכול להגיע אליו ובכך "להיחסם" מוקדם יותר.
* אם הפרי נמצא בתוך בור אולי לא משתלם לקחת אותו.

1. **נגדיר פרמטר בו אשתמש**: יהי  קבוע שלם כלשהו כך שהוא קטן-שווה ממחצית הצלע הקטנה במפה.

* רכיב 1 – מספר תאים פנויים בסביבה הקרובה:

אגדיר מטריצה  כך שהשחקן שלנו במרכז

(תוך התעלמות מתאים שנמצאים מחוץ למפה) ואסכום בה את הניקוד של כל התאים שערכם חיובי אחרי הוספת 1 (פנויים או מכילים פרי), כי לסכום תאים פנויים שערכם 0 לא עוזר לנו, ואשמור במשתנה:



* רכיב 2 – מספר קירות/מכשולים בסביבה הקרובה:

אגדיר מטריצה זהה למטריצה שברכיב הקודם רק שהפעם אסכום את כל התאים במטריצה הממורכזת שהם בלוקים (שליליים) ואכפיל ב-(1-) כדי שהתוצאה תהיה חיובית. (אם יש 4 קירות אז אקבל 4-):



* רכיב 3 – מרחק מנהטן מינימלי מהפרי הכי קרוב:

כמו שהוגדר בשאלה 3, אחשב רכיב זה ואשמור במשתנה:



כאשר האינדיקטור מחזיר 1 רק אם:

* + הפרי לא יעלם עד שנגיע אליו.
  + השחקן היריב רחוק יותר (עפ"י מנהטן) מהפרי.
* רכיב 4 – מרחק מנהטן מינימלי מהשחקן היריב:

מרחק מנהטן מהשחקן שלנו ליריב:



כעת, אאחד להיוריסטיקה אחת:



כפי שניתן לראות, ישנו אינדיקטור שיחזיר 0 רק אם המצב s הוא בור ואז ניתן לו דירוג מינימלי של 0.

כמו כן, הוספת ה-(1+) נועדה למנוע מצב של חלוקה ב-0 ובמונה הוסף "כתיקון" להוספה במכנה.

מוטיבציה:

* **מונה (כחול/אדום):** חלוקה של הניקוד של התאים הפנויים בתאים החסומים ייתן לנו תוצאה מינימלית כאשר אנחנו נכנסים "לאיזור" עם הרבה תאים חסומים ותוצאה מקסימלית אם ניכנס לאיזור עם הרבה תאים פנויים+פירות.
* **מכנה (כתום + ירוק):** ככל שהמכנה יותר קטן כך ההיוריסטיקה יותר גבוהה.

(מכנה מינימלי = היוריסטיקה מקסימלית ולהיפך).

**כלומר**, אם המרחק מנהטן מהפרי הכי קרוב (אם ניתן להגיע אליו בזמן) מינימלי נקבל היוריסטיקה גבוהה וזה טוב מאחר ואכילת פירות מקדמת אותנו מאוד במשחק (מטרה משנית).

**בנוסף**, אם המרחק מהיריב מינימלי נקבל היוריסטיקה גבוהה (ולהיפך) – הרעיון פה הוא שבשילוב עם שאר הרכיבים, ההיוריסטיקה תביא להתנהגות יחסית אגרסיבית שתשאף לחסום את היריב בהזדמנות ראשונה.

נ.ב: (אולי נמחק את השורה) ניתן גם לשים כפל במקום חיבור אך אני מפחד מהערכים הקיצוניים שניתן להגיע אליהם. (נשאר פה בהערה מודגשת לשיקול בהמשך)

רעיונות למימוש לתחרות?:

* אולי כדאי פעם בכמה תורות לעדכן משתנה שנקבע ע"י "האם קיים מסלול bfs/astar בין השחקנים וברגע שההחזרה היא false מפסיקים לבדוק את זה ואז מתעלמים מהרכיב של המרחק מהיריב.
* אפשר להוסיף בדיקה ספציפית שמתחת למרחק מסויים מהיריב (קבוע) מתחילים באלגוריתם דטרמיניסטי לחסימה של היריב.
* במקום מטריצה שמורכבת רק מ-a אולי להוסיף גם רכיב b (צלע קטנה-גדולה)...

אולי בכלל במקום מטריצה לעבור על כל התאים עד מרחק מנהטן מסויים ולספור אותם.

* אולי במקום לתת דירוג זהה לכל איבר במטריצה, לתת דירוג יותר גבוה לתאים הקרובים יותר למרכז.

דוגמא לשינוי בסגנון של הרכיב הראשון: (ניתן גם ברכיב השני)



1. כן – בכל שלב בעץ המינימקס בו בוררים צעד של אחד האויבים נלקח המינימום מהבנים שלו (רקורסיבית עד שמגיעים לשלב בעץ בו אנחנו בוררים את התור שלנו – שם נקח מקסימום)
2. חסרונות:
   1. מקדם הסיעוף גדול בצורה משמעותית בשיטה זו ולכן גם זמן החישוב לכל צעד בודד. – לא פרקטי.
   2. מתייחסים לכל האויבים באופן שווה ו-"מפחדים" מכולם – בעצם אסטרטגיה פחדנית ואף פרנואידית שיכולה לגרום לתוצאה שרחוקה מלהיות אופטימלית.
3. טרמינולוגיה: עבור  שחקנים: , .

תהיי  פונקציית דירוג (היוריסטיקה) לשחקן עבור כל אחד מצעדיו האפשריים.

לכל שחקן  וצעד אפשרי  נחשב את הדירוג: .

ונקח את התוחלת של של כל הצעדים האפשריים ונמקם אותם בטבלת דירוג הממוינת

לפי  אותה (את הטבלה) נעדכן בכל צעד שנבצע.

אקדיש לכל האויבים שלי שכבה אחת בעץ המינימקס בכל איטרציה באופן הבא:

אמשקל את כל צמתי הבנים (מתוכם אבחר את המינימלי) כך:



(אדגים עם טבלת דירוגים ספציפית לדוגמא)

טבלה:

|  |  |  |
| --- | --- | --- |
| average score | player | rank |
| 789 | 2 | 1 |
| 456 | 4 | 2 |
| 123 | 3 | 3 |

כעת, לכל צעד אפשרי של האויב אסכום את הניקוד המשוקלל של כל האויבים:



בדוגמא שלנו, בתור הבא של (מטעמי נוחות ומינימליות אניח כי יש לכל היותר 2 פעולות אפשריות):

בהינתן:



מתקיים:



ואמקם בעץ כבן של  עבור האופרטור  ומתוך הבנים האלו אבחר את המינימלי. ***(יש המשך )***

וכך גם אעשה עבור שאר השחקנים רק שעבורם, עבור שחקן :



בעצם, הרעיון באלגוריתם שלנו הוא לדרג את האויבים שלנו ולמשקל אותם לפי טיב היסטואציה שלהם ובכך לגרום לסוכן שלנו "לפחד" מההתקדמות של הסוכנים "החזקים" יותר מאשר מההתקדמות של הסוכנים "החלשים" תוך מיקסום (מקסימום) הניקוד העצמי.

1. תשובה:
2. **כן**, גיזום הענפים ב-alpha-beta יחסוך ביקורים במצבים שפסלנו ולכן יהיה מהיר יותר מ-minimax.
3. **לא**, alpha-beta פוסל רק צעדים שיחמירו את מצב הסוכן שלנו ולכן יתקדם בדיוק כמו minimax רק מהר יותר. בנוסף אין סיבה לחשוש מגיזום של ענפים שב-minimax היו משמשים אותנו לדרך הטובה ביותר מאחר ובשאלה התבקשנו להשוות בין האלגוריתמים כאשר **שניהם** כפופים לאותה מגבלה על העומק.

1. תשובה:
2. **לא** , בגיזום הענפים ב-alpha-beta עם סידור ילדים אנו מסתמכים על היורסטיקה כדי לסדר את הילדים ולכן נחלק למקרים:

במקרה בו ריצת היורסיטקה מהירה אנו משארים כי גיזום alpha-beta עם סידור ילדים ירוץ יותר מהר מאשר בגיזום alpha-beta .אך במקרה בו ריצת היוריסטיקה מאוד ארוכה ריצת בשיטת גיזום -alpha-beta תהיה מהירה יותר מאשר בשיטת גיזום alpha-beta עם סידור ילדים.

1. **לא,** נניח שהאלגוריתם alpha-beta והאלגוריתם alpha-beta עם סידור ילדים מפתחים משמאל לימין וכי יש במהלך הפיתוח של העץ בעומק 2 שני ערכים בעלי ערך זהה. אלגוריתם alpha-beta יבחר ללכת לצד שמאל בתור מהלך אך במקרה של alpha-beta עם ילדים יש אפשרות כי אלגוריתם זה יסדר את הילדים הזהים בערכם בסדר הפוך( היורסטיקה סידרה אותם כך) ולכן למרות שאלגוריתם זה יבחר גם את הצד השמאלי בטור המהלך הבא שלו ,בצד זה ישנו מהלך שונה מהמלך שנבחר באלפא בטא ולכן האלגוריתמים לא בהכרח יבחרו באותו מהלך בהכרח.
2. תשובה:

Anytime contact הינה שיטה בה בעצם משנים את הגדרת הבעיה. במקום להחזיר את הצעד הטוב ביותר אנו רוצים להחזיר את הצעד הטוב ביותר (שמצאנו בינתיים) תוך k שניות.

העמקה הדרגתית זו שיטה המקובלת להתמודדות עם מגבלת הזמן לכן נגביל את העומק אליו האלגוריתם יכול להגיע בכל איטרציה ובכך נעמיק את החיפוש ב-"קצת" כל איטרציה עד שיגמר הזמן.

אם ב-minimax בלי הגבלת עומק אנחנו **יודעים** להסיק את הערכים של כל העלים בעץ מלכתחילה אז בווריאציית Anytime contact של minimax, בונים עץ minimax בעומק מוגדר מראש (בד"כ יהיה מאוד קטן באיטרציה הראשונה), מחשבים מסלול מיטבי ואז ובכל איטרציה מגדילים את עומק החיפוש כדי למצוא צעדים יותר טובים (כי הם יותר "מיודעים") מהצעדים שנמצאו באיטרציה הקודמת ומכיוון שלא "נתון" לנו, בכל איטרציה, המשך העץ המלא עד העלים (מצבי היעד) מחשבים היוריסטיקה כלשהי עבור הצמתים העמוקים ביותר בכל איטרציה (שכנראה אינם עלים בעצמם באיטרציות הראשונות) ולפי ערכים אלו בונים עץ minimax "זמני", שמתאים לאיטרציה הנוכחית, בעומק מוגבל ומוצאים את המסלול המיטבי עבור אותו העומק.

1. תשובה: הבעיה בהעמקה הדרגתית המוצגת בהרצאה הינה בעיית האיטרציה האחרונה.

בממוצע, האלגוריתם יופסק באמצע האיטרציה האחרונה שלו והצעד "המיטבי" שיוחזר הוא התוצאה של האיטרציה הקודמת. האיטרציה האחרונה צורכת הרבה משאבים וחבל לא להשתמש במידע שהגילינו שם לכן הפתרון שהוצא בהרצאה היה שבכל איטרציה נשמור את הערך המינימקס של כל אחד מהבנים ברמה העליונה ובכך ננצל את המידע שהשגנו למרות שחלק מהבנים שברמה העליונה מחושבים עד עומקים שונים.

1. בהתחלה אגדיר רעיונות ומושגים ובסוף אחבר את הכל:

* **רעיון 1:**

אבחנה: נראה שבתורות הראשונים אין הרבה סיכונים ובתורות האחרונים אין הרבה אפשרויות בחירה גם ככה אז אקצה פחות זמן בהתחלה ובסוף.

אז איך אגדיר מתי אני בהתחלה/אמצע/סוף?:

* + אם מימדי המפה הם  ונתון לי מספר התאים החסומים (-1): 

אז אגדיר: 

ואתאים פונקציה: 

כך שבינתיים אפשר להגדיר:



* **רעיון 2:** (מתאים בעיקרון רק למשחק שבמטלה כרגע אך ניתן לשנות אותו בהתאם למשחק)

אבחנה: אם הגענו לתור בו היריב "קרוב" לסוכן שלנו נרצה להקצות יותר זמן עבור חישוב הצעד.

איך אגדיר קרבה ואיך נשתמש בה?:

* + בהינתן פונקציית מרחק מנהטן כמוגדר בסעיפים הקודמים נסמנה: 

אז אגדיר: 

וכעת אחבר לרעיון הקודם כך שבינתיים:



נבחין כעת כי הרעיון השני יכול לגרום לנו לחרוג מהזמן הגלובלי הנתון לקראת סוף המשחק לכן אגדיר חסם שרירותי (שניתן לשנותו בהתאם לבדיקות וניסויים עם פרמטרים שונים) של 90% כאשר אם עבר 90% מהזמן הגלובלי הנתון אניח שברוב המקרים זה אומר שעברנו את המקטע הבעייתי עם היריב או שלפחות הגענו ל-phase של סוף המשחק (phase=3) אז החל מנקודה זאת אפעיל את האלגוריתם הנאיבי לטיפול בזמן רק שבמקום הפרמטר global\_time אציב את הזמן הנותר

(שאמור להיות באיזור ה-0.1\*global\_time)

ואחשב מחדש את num\_turns כך שיהיה רלוונטי.

1. אפקט האופק הינו תופעה בה אלגוריתם מוגבל משאבים בוחר צעדים "סתמיים" כדי לדחות "צרות" מעבר לאופק החיפוש כלומר הוא נמנע מצעדים מסוכנים לו ומבצע פעולה כי הוא מוגבל בעומק ולכן מחליט לא להחליטהפתרון לאפקט האופק הינו העמקה סלקטיבית -מפתחים את העץ עד העומק המוגדר באיטרציה הנוכחית ואם מקבלים עלים "לא שקטים" (נקבע עפ"י קריטריון מוגדר מראש) נעמיק עוד שכבה בחיפוש לבדוק אם הערך ההיוריסטי התייצב – אחרת מעמיקים עוד קצת ועוד קצת עד שנגיע לרגיעה (התייצבות הערך ההיוריסטי למשל) או עד עומק קבוע מסויים שמוגדר מראש להעמקות סלקטיביות של האלגוריתם.

(בתרגול סימנו עומק זה כ-k ולכן עומק תת העץ של הבן הלא יציב הוא סה"כ D+k כאשר D הוא עומק ההעמקה באיטרציה הנוכחית)

פתרון זה יכול לעזור לנו במשחק הנתון במקרים:

1. מקרה בו הסוכן שלנו יכול לפגוע בעצמו ע"י החלטה גרועה.
2. מקרה בו האויב מקדם את עצמו מאוד כתוצאה מהזנחה של נקיטת פעולות נגדו.

דוגמאות:

1. מצב בו אנחנו יכולים להכניס את עצמנו לבור (יכולים לחסום את עצמנו):

אתייחס לעומק חיפוש 2 כ-"שדה ראייה" של 2 צעדים קדימה.

כלומר, בהינתן המצב: (אדום יריב, כחול "אני")



"שדה הראייה" של הסוכן שלי הוא:



וקל לראות שהסוכן שלנו ייחסם תוך 2 צעדים בשביל לקחת את הפרי בעוד שאם היה מנסה ללכת לפרי השני כנראה שהיה לו סיכוי והיה מנצח.

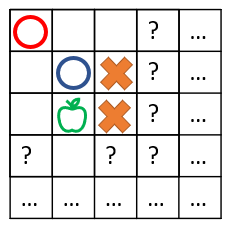
1. מצב בו האויב לוקח פרי ומעלה לעצמו משמעותית את הניקוד:

(אשתמש באותם פרמטרים ודימויים כמו בדוגמא הקודמת)

כלומר, בהינתן המצב:



"שדה הראייה" (=2 צעדים קדימה) של הסוכן שלי הוא:



כעת קל לראות שהסוכן שלנו לא ינסה לחסום את היריב אלא ייקח את הפרי הקרוב אליו בעוד שאם היה מעמיק אפילו בצעד בודד היה חוסם את היריב ולוקח את "השלל" לעצמו ובכך מנצח.

1. בעצם ערך המינימקס מהריצה הקודמת זה החסם התחתון המהודק של  והחסם העליון המהודק של לאחר ריצת האלגוריתם.

אז כעת, במקום לאתחל את  למינוס אינסוף ואת  לאינסוף (או ערך ספציפי אחר כלשהו), ניתן לאתחל אותם לערך המינימקס שנשמר ובכך לגזום "מראש" את כל(?) המהלכים שגזמנו בריצה הקודמת לפי הקריטריונים של  ו-.

במקרה הטוב ביותר כמעט כל העץ למעט "המסלול" המיטבי נגזם ואז בריצה החוזרת

1. ג
2. א. החלטתו של השחקן לא תשאיר זמן בכלל לחישוב *B-1* הצעדים הבאים, אראה ע"י חישוב פשוט באמצעות ערך משולש:



כלומר, צריך  דקות על מנת לחפש בצעד הראשון עד לעומק  ואז לא ישאר זמן לשאר *B-1* הצעדים הנותרים אז כנראה שייבחר באקראי צעד בכל אחד מהם או שהשחקן פשוט ייפסל.

ב. תשובות:

1. מס' הקשתות שעל השחקן לשמור הוא כמספר העלים פחות אחד רק שבמקרה זה אנחנו צריכים גם לשמור את הקשת של הצעד המוביל לתת-העץ הנבחר לכן:



1. *כאשר נשמור את האסרטגיה של הצעד הראשון, בעצם במקום שנצטרך לחשב תוך 0 זמן (מסעיף א' לא נשאר זמן ל-B-1 הצעדים בהמשך) אנחנו יודעים "מראש" להגיב לכל צעד שהיריב יבחר כך שבעצם באלגוריתם החדש, אם נחלק את כל הצעדים שהשחקן שלנו ייעשה לאורך המשחק למקטעים של B צעדים אז בכל מקטע כזה הצעד הראשון יהיה קבוע הצעד הראשון שנמצא בעץ ושאר B-1* הצעדים ייחושבו לפי מינימקס קלאסי.

*ובכך בעצם שיפרנו את המצב מסעיף א' שאם קודם האופציות היו או להיפסל או לבחור צעד באקראי ב-B-1 המהלכים הבאים אז עכשיו יש אסטרטגיה עבור B-1 צעדים אלו.*

1. *מתי האלגוריתם המשופר טוב יותר:*
   * *אם במקרה, בכל קבוצה בגודל B של צעדים הצעד הראשון שנבחר לפיתוח ע"י האלגוריתם הראשון הוא צעד שלא היה נבחר ע"י מינימקס סטנדרטי שבודק עד עומק D אך מעבר "לאופק" של D צעדים קדימה מתגלה שהצעד שנבחר "אקראית" ע"י האלגוריתם המשופר הוא בעצם הצעד הטוב ביותר שיכל להיבחר בעוד שצעד זה לא נבחר ע"י המינימקס הסטנדרטי.*

*מתי האלגוריתם המשופר מתנהג בצורה דומה:*

* + *כמו בתשובה הקודמת רק שבמקום הצעד הראשון כל B צעדים הוא צעד שהמינימקס הסטנדרטי לא היה בוחר, פה מדובר על מקרה בו הצעד הראשון שנבחר "אקראית" ע"י האלגוריתם המשופר מתלכד עם הבחירה של האלגוריתם הסטנדרטי.*

*מתי האלגוריתם המשופר מתנהג גרוע יותר:*

* + *האמת שברוב המקרים האקראיים, יותר סביר ששני הסיטואציות שתיארתי קודם לא מתרחשות והבחירה באקראי של הצעד הראשון כל B צעדים כנראה שתביא לתוצאה גרועה יותר מאשר אם היינו בוחרים גם את הצעד הראשון ע"י "ניחושים מושכלים" ולא רק את B-1 הצעדים הבאים.*
  + *נ.ב: לבדוק בסוף: אולי בגלל שהוא שומר אסטרטגיה ל-D צעדים קדימה אז צריך לשנות את סעיף (ב-ii) כי ההשפעה משתנה ואולי רק הצעד הראשון במשחק נבחר באקראי.*
  + *b>d or vice-versa*