

## מבוא לבינה מלאכותית – תרגיל בית 2 – פאקמן:

### א. הכרות עם הקוד והמשחק:

1. שחקן בסיסי ReflexPlayer בוחר בכל צומת החלטה את הפעולה הבאה לביצוע ע"י בחינת המצבים הבאים אליהם כל פעולה אפשרית מובילה, ובחירת הפעולה המובילה למצב בעל הניקוד הגבוה ביותר. פונ' היוריסטיקה היא, אם כן, הניקוד המקס' של המצב המבין סט המצבים הבאים

$$Heuristic\{State\} = \max_{score} (state.next\_states.score)$$

### ב. בניית סוכן משופר:

1. רצינו שהיוריסטיקה שלנו תהא מורכבת מקומביניציה לינארית של המאפיינים הבאים (הפרמטרים ניתנים לכוונון):

- ניצחון/הפסד
- מרחק מינימלי לאוכל
- ניקוד
- כמות האוכל במגרש
- כמות הקפסולות במגרש
- האם קפסולה
- כמות הרוחות הרעות במגרש

2. החישוב שלנו עבור היוריסטיקה מובא להלן:

$$h(s) = \begin{cases} -1000 & \text{lost} \\ score + \frac{1}{minD2food} - \frac{1}{dis2badghost} + \frac{1}{dis2goodghost} + \frac{1}{numofcapsules} & \text{else} \end{cases}$$

- יש לשים לב למקדמים השליליים עבור חלק מהפרמטרים. הם קיימים על מנת לאפשר

לוגיקה הפוכה; המצב נבחר כמקסימלי, לכן נרצה לתת ערך גבוה ככל האפשר עבור:

1. מרחק מינימלי לאוכל – נבחר להתקדם לכיוון האוכל הקרוב ביותר על פני כיוון האוכל הרחוק.

2. במקרה בו המשבצת הסמוכה אלינו מכילה קפסולה, נבחר להתקדם אליה על פני משבצת בה אין קפסולה.

3. נבחר להתקדם למשבצת בה מס' הרוחות הפעילות נמוך יותר, כלומר למשבצת בה ביצענו אכילה של רוח רעה, על פני משבצת בה לא ביצענו זאת.

4. נרצה להתרחק מרוחות רעות ולכן ניתן משקל שלילי לפעולות שיביאו אותנו קרוב לרוחות רעות.

- בפועל, בגלל אילוצים של זמן ריצה וזמן חישוב השתמשנו ביוריסטיקה המתחשבת רק בשתי הנקודות הראשונות. כלומר, הסוכן שלנו לא מודאג מרוחות כלל, אלא רק שואף לניקוד הגבוה ביותר.

### ג. בניית סוכן Min-Max

1. על ידי שימוש בעץ אסטרטגיה, אנו למעשה מניחים כי:
  - קבלת החלטות במשחק מתקבלת בתורות. פעם תור הסוכן ובפעם תור כל אחת מהרוחות. כלומר, אנו מניחים שרוח לא תקבל החלטה פעמיים בלי שהסוכן קיבל החלטה פעם אחת באמצע. הנחה זו לא בהכרח נכונה שכן קבלת ההחלטות מתקבלת אצל הסוכנים השונים במקביל.
  - השחקן היריב גם משחק בצורה אופטימלית.
3. דרך נוספת לחישוב ערך minimax היא להתייחס לכל הרוחות יחד כסוכן אחד. בדרך זו נשארת שכבת min אחת בעץ האסטרטגיה.
  - חסרון שיטה זו בא לידי ביטוי במקדם הסיעוף. בעץ בעלת שכבת min אחת לכל רוח, מרחב המצבים השכנים יכלול את המצבים הבאים לאחר פעולת הרוח הספציפית. בעץ בעל שכבת min אחת בלבד, מרחב המצבים השכנים יכלול את המצבים הבאים לאחר פעולה של כל אחת מהרוחות ולכן זהו מרחב מצבים גדול בצורה ניכרת, כלומר מקדם הסיעוף B גדול במצב זה משמעותית.
  - לפי משפט מהתרגול, סיבוכיות הזיכרון היא  $\theta(B \cdot D)$ . אפילו שעומק העץ הצטמצם, כיוון שמקדם הסיעוף גדל כל כך, חיסרון השיטה היא בכמות הזיכרון הגדולה הנדרשת ליישומה לעומת השיטה של עץ בעל כמה שכבות min.
  - יתרון שיטה זו בא לידי ביטוי בעומק העץ. לפי משפט מהתרגול סיבוכיות הזמן אקספוננציאלית בעומק -  $\theta(B^D)$ . אפילו שמקדם הסיעוף גדל, עצם העובדה שהצלחנו להקטין את עומק העץ תביא לשיפור בזמן ריצת התוכנית לעומת שיטה המשתמשת בעץ אסטרטגיה בעל כמה שכבות min.

### ד. בניית סוכן Alpha-Beta

1. מבנה העץ החדש ישפיע על אלגוריתם alpha-beta באותו אופן שבו השפיע על אלגוריתם minimax. באלגוריתם זה יש שימוש בשני חסמים,  $\alpha$  (חסם תחתון מקסימלי) ו- $\beta$  (חסם עליון מינימלי) שיאפשרו צמצום של הצמתים שנפתחו בעץ. נפתח רק צמתים שיקיימו בכל זמן נתון  $\alpha < \beta$ 

$$minimax\ value < \beta$$
- גיוזם  $\alpha$ : פיתוח צומת min. אם החסם של  $\alpha$  גבוה מהערך המתקבל באחד השכנים נבצע גזימה של הענף כיוון שמובטח לנו שערכו של צומת ה-max העליונה יהיה בתחום  $\alpha < \beta$

$\beta < \text{minimax value}$  ולכן אין צורך לפתח ענף המביא לערכים שלא בטווח. אמנם בעץ זה מעל צומת  $\min$  אין בהכרח צומת  $\max$  כמו בעץ אסטרטגיה של שני משתתפים, אך לפי הגדרת העומק אנו יודעים כי בסוף כל שכבות ה- $\min$  של הרוחות, תמצא שכבת  $\max$  של פקמן. לכן גם בעץ זה אין טעם לפעפע מעלה מסלולים שבמילא לא יתחשבו בהם בשכבת ה- $\max$  הבאה.

גיזום  $\beta$ : פיתוח צומת  $\max$ . אם החסם של  $\beta$  נמוך יותר מהערך המתקבל באחד השכנים, נבצע גזימה של הענף. אין צורך לפתח ענף שבודאות לא יבחר בשכבת ה- $\min$  הבאה. מאותם שיקולים כמו בגיזום  $\alpha$ , למרות שמבנה העץ שונה, מובטח לנו שאחרי כל שכבת  $\max$ , תהיה שכבת  $\min$  ולכן נכונות הגיזום בפרט והאלגוריתם בכלל, נשמרת.

3. סוכן ה-  $\alpha$ - $\beta$  יתנהג שונה מסוכן  $\text{minimax}$  בשני האופנים:

- זמן הריצה יתקצר – זמן חישוב הצעד הבא יתקצר שכן עבור אלגוריתם זה אין טעם לחשב את ערך  $\text{minimax}$  עבור כל המסלולים, אלא מסלולים שיועדים מראש שלא יבחרו לא נלקחים בחשבון. מספר החישובים קטן לעומת אלגוריתם  $\text{minimax}$  וכתוצאה מכך גם זמן הריצה.
- ערך ה- $\text{minimax}$  הסופי יישאר זהה לזה של סוכן ה- $\text{minimax}$ , אך תתכן בחירת מהלכים שונה וזאת בעקבות מצבים של שוויון בין ערכי  $\text{minimax}$ . עבור אלגוריתם זה אם יש שוויון וערך  $\text{minimax}$  של צומת שווה ל- $\alpha$  או ל- $\beta$ , הענף נגזם והצומת בוודאות לא תיבחר לפעפע הלאה, זאת בניגוד לאלגוריתם  $\text{minimax}$  שיכל היה לבחור אחרת.

## ה. בניית סוכן Expectimax לרוח רנדומלית

כעת יש בידנו מידע נוסף, תנועת הרוחות נבחרת על ידי התפלגות אחידה על כל האפשרויות. Minimax ו- $\alpha$ - $\beta$  שניהם מניחים כי היריב יעשה את הצעד הכי טוב בשבילו וכאן אנחנו יודעים שלא כך המצב. אין סיבה לתת לצומת את ה- $\text{value}$  הכי נמוך, אלא ניתן להעריך אותה כתוחלת הערכים האפשריים.

- נצפה שבממוצע מספר הפעמים ש- סוכן Expectimax יפסיד יהיה גדול יותר ממספר הפעמים שסוכני  $\text{minimax}$  או  $\alpha$ - $\beta$  יפסידו.
- נצפה שבמצב שבו פקמן לכוד בין רוחות נראה התנהגות שונה בין אלגוריתם זה לאלגוריתמים הקודמים. עבור האלגוריתמים הקודמים, ההנחה היא תמיד שהרוחות יעשו את הצעד הכי גרוע לפקמן, פקמן יחשוב שאין לו מצב שלא מוביל להפסד ולכן יפסיד בוודאות. מה שקורה בפועל הוא שיש סבירות מסוימת שהרוחות ילכו לכיוון שלא מוביל

לפקמן, כלומר לא הכי גרוע בשבילו. סוכן Expectimax ידע להתחשב באפשרות זו ולכן חלק מהפעמים יחלץ מהמלכוד וינצח.

## 1. בניית סוכן Expectimax לרוח לא רנדומלית

1. הרוח הכיוונית מקיימת מדיניות סטוכסטית של תנועה בהסתברות גבוהה בכיוון המקיים מרחק הקצר ביותר מפאקמן אם היא במצב פעיל, ותנועה בכיוון המרחק הארוך ביותר מפאקמן בהסתברות גבוהה אם היא במצב מבוהל. ספציפית ההסתברות נבנית כך שאם קיימים מספר כיוונים בעלי מרחק הקצר ביותר/ארוך ביותר, ההסתברות עליהם אחידה, וכן אחידה עברו המצבים המשלימים להם. למשל אם הרוח במצב פעיל,  $1 - 3$  כיוונים מקיימים מרחק מנהטן מינימלי לפאקמן, נניח בה"כ

$$P_x(x) = \begin{cases} \frac{4}{15} : up \\ \frac{4}{15} : down \\ \frac{4}{15} : right \\ \frac{3}{15} : left \end{cases} \quad \text{למעלה, ימינה, למטה. אז ההתפלגות תהא:}$$

נשים לב גם שככל שיש פחות כיוונים אפטימליים במידה שווה, ההסתברות עליהם גדלה ואילו ההסתברות על כל אחד מהמצבים המשלימים קטנה.

2. בקוד.

3. ההבדל הוא באופן חישוב התוחלת. בעוד randomExpectimax הניח התפלגות אחידה של הבחירות של הרוח, וביצע חישוב של התוחלת באופן ה"ל, directionalExpectimax מבצע משקול שונה של הבחירות של הרוח לכיוון התקדמות, לפי ההסתברות הידועה לנו שהבחירה אכן תילקח.

4. רעיונות לשיפור הרוח:

- ניתן להציע פונ' התפלגות טובה יותר עבור הבחירות של הרוחות, כך שתתקיים הבדלה בין מצבים לא אופטימליים יותר ופחות. לדוג', נציע בחירה של הכיוונים האופטימליים בהסתברות אחידה כפי שמתקיים עכשיו, ופילוג גאוסי על היתר. זה יבטיח בחירה בכיוון הכי פחות אופטימלי בהסתברות נמוכה יותר מאשר הכיוון הלא אופטימלי הבא אחריו.
- בניה ועדכון של מודל הבחירה של פאקמן בהתאם לצעדים שהוא לוקח. תחת ההנחה שמודל הבחירה של פאקמן לא ידוע לרוח (הנחה חזקה שכן אנחנו הנחנו בdirectional שמודל הבחירה של הרוח כן ידעו לנו), נוכל לאמן רשת נוירונים שמטרתה לשערך את פונ'

הevaluation של פאקמן, ואז תמיד לבחור פעולה אופטימלית מולו. נבצע את הצעדים הבאים:

1. נאתחל את כל הבחירות של פאקמן להסתברות שווה
2. נמדוד פרמטרים שונים של המשחק (מיקום פאקמן, מיקום רוחות, כמות אוכל, כמות רוחות פעילות וכו')
3. נבצע feed forward של הפרמטרים הנמדדים דרך רשת נוירונים המבצעת classification, נקבל פילוג על הכיוונים, ובבחר במקס'.
4. נבצע צעד ונמדוד את הבחירה שפאקמן עשה.
5. נעדכן את פרמטרי הרשת על פי פונ' מחיר הלוקחת בחשבון את הבחירה של פאקמן ואת השערוך שביצענו.
6. לאחר מספר איטרציות כאלו, במידה והרשת התכנסה נגיע לשערוך אופטימלי של הבחירה של פאקמן בכל צעד.

## ז. ניסוח השערות במשחק פאקמן

1. נגדיר משתנה אקראי ברנולי עבור משחק בודד של הסוכן שלנו מול המחשב:

$$X_{game} = \begin{cases} 1 & \text{if Agent won (with probability } p) \\ 0 & \text{otherwise (with probability } 1-p) \end{cases}$$

2. נניח השערת האפס כי הסוכן לא טוב מהמחשב, בממוצע. כלומר, ההסתברות לנצח זהה

להסתברות להפסיד למחשב:  $p = 0.5$

3. נגדיר את ההשערה הנגדית כהנחה כי קיימת הסתברות גבוהה יותר לניצחון הסוכן על המחשב

$$\begin{aligned} H_0 : p = 0.5 \\ H_1 : p > 0.5 \end{aligned} \quad \text{אם כן: (מבחן חד צדדי)}$$

5. נגדיר  $t$  סטטיסטי של הנתונים עבור המבחן:

$$t = \frac{\hat{p} - p_0}{se(\hat{p})} \quad : \quad \hat{p} = \frac{1}{n} \sum_{k=1}^n X_k = \overline{X_n}, \quad p_0 = 0.5, \\ se(\hat{p}) = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

6. נגדיר את חוק המבחן: נדחה את השערת האפס אם  $t > c$  עבור  $c$  כלשהו.

7. נשים לב שעבור הסטטיסטי t הנ"ל מתקיים:

$$E[X_i] = p$$

under CLT theorem, we denote that the sum of bernoulli RV's

$$\text{is Normal-Distributed: } \sum_{k=1}^n X_k \sim N(np, np(1-p))$$

$\hat{p}$  is Normal-Distributed also:

$$\hat{p} = \frac{1}{n} \sum X_i \stackrel{N \sim N(np, np(1-p))}{=} \frac{1}{n} \cdot N \sim N\left(p, \frac{p(1-p)}{n}\right)$$

לכן עבור המשערך של p מתקיים:

ואז עבור הסטטיסטי t כולו, נשים לב שמתקיים נרמול למשתנה נורמלי ממורכז:

$$\hat{p} \xrightarrow[\text{Distribution}]{\rightarrow p} t = \frac{\sqrt{n}(\hat{p} - p_0)}{\sqrt{\hat{p}(1-\hat{p})}} \sim N(0,1)$$

תחת חוק המספרים הגדולים המקיים:

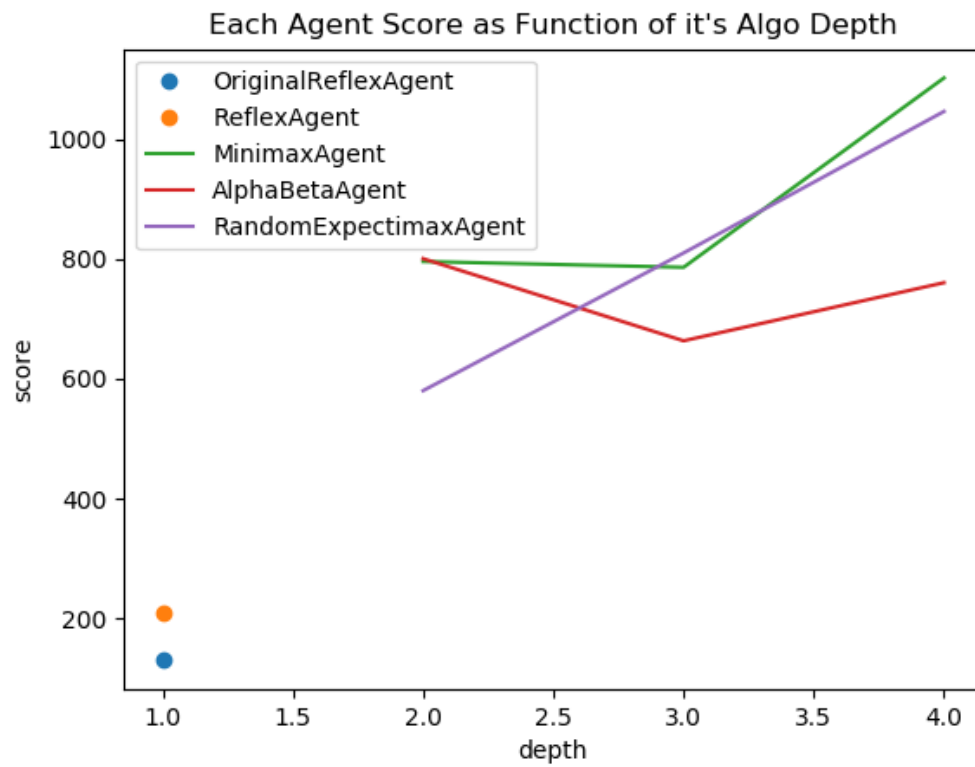
8. לסיכום, נדחה את השערת האפס בהסתברות  $P(t > c) = 1 - \Phi(c) = p\text{-value}$ , ונקבע את

הקבוע לפי רמת המובהקות הקריטית הדרושה.

## ח. ניסויים תוצאות ומסקנות

1. מצורף קובץ experiments.csv.

2.



name	1	2	3	4
OriginalReflexAgent	130.557			
ReflexAgent	210.128			
MinimaxAgent		795.513	785.827	1101.516
AlphaBetaAgent		800.244	663.301	760.214
RandomExpectimaxAgent		580.229	809.757	1045.728

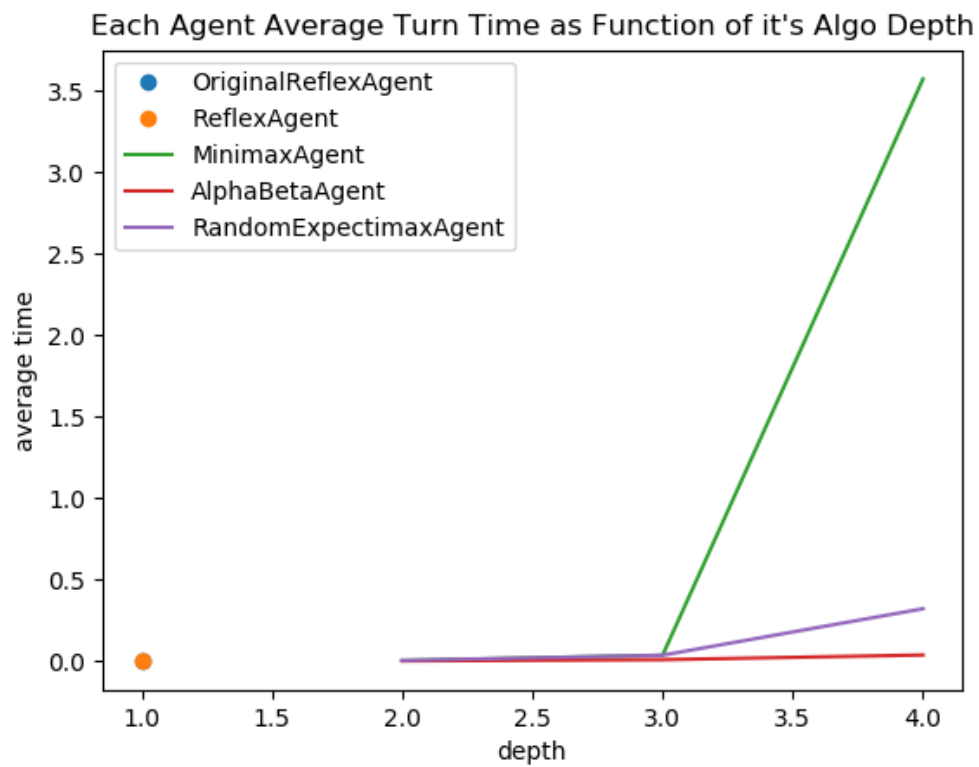
3. ראשית יש לשים לב כי האלגוריתמים הרפלקסיביים (לא מנסים לתכנן קדימה), השיגו ביצועים

נחותים באופן משמעותי מכל היתר.

שנית, נשים לב כי אלגוריתם ה-  $\text{Expectimax}$  הינו בעל תוצאות נחותות בממוצע ברוב העומקים לעומת שאר האלגוריתמים המתכננים. זה גם הגיוני משום ששאר האלגוריתמים מניחים את המצב הגרוע ביותר מבחינתם (הבחירה הטובה ביותר של היריב), ולכן נוקטים פעולה אופטימלית תמיד. סוכן  $\text{Expectimax}$  לקוח בחשבון את התגובה הממוצעת של היריב, ולכן לפעמים מבצע פעולות המביאות אותו להפסד שהיה יכול להימנע.

בניגוד לציפיותינו, אמנם סוכן  $\text{AlphaBetaAgent}$  השיג תוצאות זהות לסוכן  $\text{MinimaxAgent}$  בעומק 2, אך בעומקים אחרים ביצעו פחות טובים. ניתן להסביר זאת בהשפעה של היוריסטיקה איתה בחרנו לרוץ. כיוון שהיוריסטיקה שלנו אינה מתחשבת ברוחות שבלוח ופיתוח בעומקים 3 ו-4 מתבצעים לפי שיקולים של זמינות אוכל, ייתכן שהסוכן קיצץ ענפים בעץ האסטרטגיה שהיו מובילים לנתיב בריחה מהרוחות. הדבר מוביל לכמות הפסדים גדולה יותר מבחינת  $\text{AlphaBetaAgent}$  ולכן לממוצע ניקוד נמוך יותר.





name	1	2	3	4
OriginalReflexAgent	0.000137			
ReflexAgent	0.000356			
MinimaxAgent		0.005074	0.037451	3.574321
AlphaBetaAgent		0.002975	0.010291	0.039204
RandomExpectimaxAgent		0.005644	0.037842	0.323193

5. התוצאות תואמות היטב את ציפיותינו. קל להבחין בשיפור המשמעותי ששיגי סוכן AlphaBeta

על פני שאר הסוכנים המתכננים. חשוב להדגיש גם כי ככל שהעומק גדל, כן היתרון של האלגוריתם גדל. נק' חשובה היא העומק 2 – ניתן להבחין בכך שלכל האלגוריתם אותו זמן בקירוב, וזה עקב אופן הפעולה של אלגוריתם AlphaBeta אשר צריך לפתח עוד שלב למטה על מנת לבצע גיזום (מסתכל על 'הנכד' של השכן), ולכן עבור עומק 2 לא מבצע גיזום.

6.

RandomExpectimaxAgent	4	trickyClassic	1858	7.48E-01	RandomGhost
RandomExpectimaxAgent	4	trickyClassic	1678.8	6.01E-01	DirectionalGhost
DirectionalExpectimaxAgent	4	trickyClassic	1316.8	5.35E-01	RandomGhost
DirectionalExpectimaxAgent	4	trickyClassic	1315.8	1.13E+00	DirectionalGhost

כפי שניתן לראות, עבור סוכן המניח התפלגות אחידה ורוח שאכן מקיימת התפלגות אחידה, הניקוד גבוה יותר מאשר עבור רוח המקיימת התפלגות לא אחידה. עבור סוכן המניח התפלגות מכוונת התקבל ניקוד זהה עבור שני סוגי הרוחות. אין לי במה לתלות את זה מלבד בעובדה שמדובר על מדגם קטן מידי של 5 הרצות בלבד. כן חשוב לציין בהקשר הזה שזמן חישוב התור הממוצע ארוך הרבה יותר עבור הרוח הכיוונית וזאת עקב החישוב העודף המשמעותית הדרוש לשם חישוב ההתפלגות.

7. טבלת סיכום – minimaxClassic בעומק 4:

OriginalReflexAgent	1	minimaxClassic	225.57	8.48E-05
ReflexAgent	1	minimaxClassic	82.86	1.56E-04
MinimaxAgent	2	minimaxClassic	515.14	1.71E-03
MinimaxAgent	3	minimaxClassic	367	4.31E-03
MinimaxAgent	4	minimaxClassic	224.29	1.15E-02
AlphaBetaAgent	2	minimaxClassic	83.14	1.68E-03
AlphaBetaAgent	3	minimaxClassic	366.86	3.22E-03
AlphaBetaAgent	4	minimaxClassic	510.57	5.04E-03
RandomExpectimaxAgent	2	minimaxClassic	371.86	1.94E-03
RandomExpectimaxAgent	3	minimaxClassic	511.71	5.02E-03
RandomExpectimaxAgent	4	minimaxClassic	514.29	1.38E-02

הלוח minimaxClassic מאופיין במרחב תמרון צר, מיעוט אוכל וריבוי רוחות יחסי לשטח המצומצם. אלמנט זה מוביל למשחקיות קשה, ומעניק יתרון יחסי לאלגוריתמים שמתכננים היטב את צעדיהם הבאים שכן ניתן להיכנס למצב שאין ממנו מוצא בקלות. ניתן להבחין בקלות כי אלגוריתם minimax וכן אלגוריתם expectimax השיגו תוצאות

עודפות על האלגוריתם הרפלקסיבי. אלגוריתם alphabeta אמור להיות דומה בביצועיו לאלגו' minimax, אך לא ניתן להבחין בכך בקלות עקב גודל מדגם קטן מידי של משחקים (כך אנו משערים).

ניתן גם להבחין בעדיפות של אלגוריתם expectimax על פני minimax הרגיל, וזאת עקב היכולת שלו לקבל החלטות תת אופטימליות, כלומר, גם במצבים בהם קיימים 2 יריבים, ושילוב הפעולה האופטימלית שלהם מביא להפסד ודאי במשחק, אלגוריתם minimax למעשה יתאבד (יקח את אחד הפעולות בהתאם לתגובות הנגד האופטימליות שיובילו להפסד), ואילו אלגו' expectimax יקבל החלטות בהתאם לתוחלת הפעולות של היריבים, כלומר יצא מהמצב הנחות בהסתברות כלשהי.

8. טבלת סיכום – trappedClassic בעומק 4:

Agent	Score	Time
MinimaxAgent	384.29	9.64E-04
AlphaBetaAgent	-501	5.75E-04
RandomExpectimaxAgent	-58.86	8.59E-04

הלוח הזה מאופיין בהיותו קטן, יש רק מסלול אחד לנוע בו ובכך שפקמן לכוד בין שתי רוחות, ימנית ושמאלית.

- סוכן AlphaBetaAgent מפסיד תמיד בלוח זה כיוון שמחשב 4 צעדים קדימה לכיוון האוכל, תמיד מניח שהרוחות יסגרו עליו בדרך וגוזם ענפים אחרים שיכולים להוביל עבורו לדרכי מילוט מהרוחות. הוא תמיד יבחר בענף שבו הרוח הימנית סוגרת עליו.
- סוכן RandomExpectimaxAgent, מצליח להימלט מהרוח הימנית כיוון שלוקח בחשבון את האפשרות שהרוח השמאלית לא תאכל אותו. לכן מצליח לנוע במסלול היחיד שקיים, ניצחון או הפסד במשחק זה תלוי בתנועת הרוח השמאלית.
- סוכן MinimaxAgent, מניח תמיד את האופציה הגרוע ביותר עבורו במרחק של 4 צעדים קדימה, היינו מצפים שהתוצאות עבור סוכן זה יהיו פחות טובות מהתוצאות עבור סוכן RandomExpectimaxAgent, כיוון ש-MinimaxAgent מניח את האופציה הגרועה ביותר ולכן בהרבה מקרים יעדיף לסיים את המשחק כי יחשוב שבלואי הכי הרוח תאכל אותו. מניתוח התוצאות הגענו למסקנה שאמנם MinimaxAgent רואה את האופציה הגרוע ביותר עבורו, אך

הוא בוחר "להתאבד" על ידי בחירה ברוח השמאלית ולא הימנית. אם הרוח השמאלית תלך לכיוון הנגדי ממנו הוא מספיק לאסוף נקודות ואולי גם לנצח.

שני הסוכנים RandomExpectimaxAgent ו-MinimaxAgent אפקטיביים נעים באותו מסלול. ניצחון או הפסד במשחק נתון תלוי באקראיות תנועת הרוחות, ולכן התוצאות בטבלה אינן מספיקות על מנת לשקף מגמה.

## 9. סעיף סיכום:

### יוריסטיקה:

היוריסטיקה איתה בחרנו לבצע את ההרצות מבוססת בעיקרה על ניקוד ועל המרחק של שחקן פקמן מאוכל. פקמן יעדיף לנוע לכיוון שיש בו אוכל מאשר כזה שאין בו, כל עוד התנועה לא גורמת לו להפסיד.

יוריסטיקה זו עדיפה על היוריסטיקה הפשוטה המתחשבת רק בניקוד במקרים בהם פקמן נמצא באזור שבו אין אוכל, אך עדיין יש אוכל על הלוח. במקרים אלו אין העדפה מבחינת ניקוד לצעד הבא, אלא אם כן רוח תתקרב לאזור ותבריא את פקמן. לעומת זאת לפי היוריסטיקה שלנו, פקמן יעדיף עצמאית ללכת לאזורים בהם יש אוכל. כך יחסך זמן והורדת ניקוד.

### מגבלת עומק:

מגבלת העומק השפיעה על הסוכנים השונים בצורה שונה.

#### • **סוכן RandomExpectimaxAgent** - עבור סוכן זה ככל שהעומק גדול יותר, כך גם הניקוד

שהוא מצליח להרוויח גדל (גרף לינארי).

#### • **סוכן AlphaBetaAgent** - עבורו ניתן לראות רגרסיה בביצועים עבור מגבלת עומק 3

לעומת מגבלת עומק 2. עבור מגבלת עומק 2 הגזימה לא רלוונטית ולכן הסוכן מתנהג בצורה דומה ל-MinimaxAgent. עבור מגבלת עומק 3, הסוכן מסתכל קדימה אך לא מספיק קדימה ולכן יבחר צעדים שיקדמו אותו נקודתית (לכיוון אוכל וניקוד גבוה יותר), ויגזום ענפים שאולי לא מקדמים אותו נקודתית אך הם הדרך בה צריך לבחור כדי לנצח. גזימת היתר הזו היא זו שמובילה לירידה בביצועים. לעומת זאת עבור עומק 4 ניתן לראות שיפור, אמנם עדיין יש גזימת יתר אך הסוכן מודע יותר והגזום מודע יותר. נצפה שככל שהגבלת העומק תגדל, כך ישתפרו ביצועי הסוכן מבחינת ניקוד.

#### • **סוכן MinimaxAgent** - באופן דומה לסוכן AlphaBetaAgent, גם פה ניתן לראות

רגרסיה עבור מגבלת עומק 3 ושיפור משמעותי עבור מגבלת עומק 4. הסיבות דומות - מגבלת עומק 3 גורמת לשחקן לקבל החלטות לא נכונות בטווח הרחוק ולכן הביצועים יורדים. בעומק 4 קפיצה בביצועים בעקבות קבלת החלטות נכונה יותר.

רואים הבדל משמעותי בביצועים בין שחקן AlphaBetaAgent לבין MinimaxAgent לטובת הסוכן השני. ההבדל נעוץ בגיזום מסלולים שמאפשרים נתיב מילוט מהרוחות, אך לא מקדמים את השחקן מבחינת ניקוד ו\או אוכל מבחינת סוכן AlphaBetaAgent. נצפה לכך שבהינתן יוריסטיקה המתחשבת בבריחה מרוחות רעות הביצועים יהיו דומים.

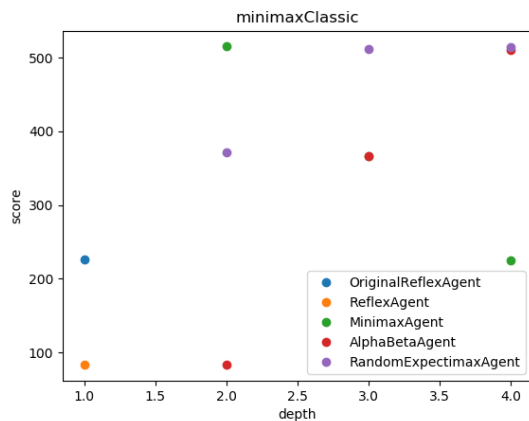
#### זמן ממוצע לתור:

ניתן לראות tradeoff בין זמן ממוצע לתור לבין ביצועי הסוכן כתלות בעומק. ראשית, ככל שהעומק גדול יותר, כך הסוכן צריך לבצע יותר חישובים בתור ולכן זמן ביצוע תור גדל. שנית ניתן לראות הבדלים גם בין הסוכנים, כאשר לסוכן MinimaxAgent זמן חישוב גדול ביותר. זאת כיוון שבכל תור הוא מחשב את כל האופציות העומדות לראשותו ומבניהן בוחר. לסוכן AlphaBetaAgent זמן ממוצע לתור קטן ביותר כיוון שבעקבות הגיזום, מרחב האפשרויות שלו קטן ולכן החישוב שהוא מבצע פחות מסובך ודורש פחות זמן חישוב. נצפה שמצב של מגבלת זמן במקום מגבלת עומק היה הופך את תמונת הביצועים והיינו מקבלים שהסוכן בעל הביצועים הטובים ביותר היה AlphaBetaAgent ולא MinimaxAgent.

#### הלוחות השונים:

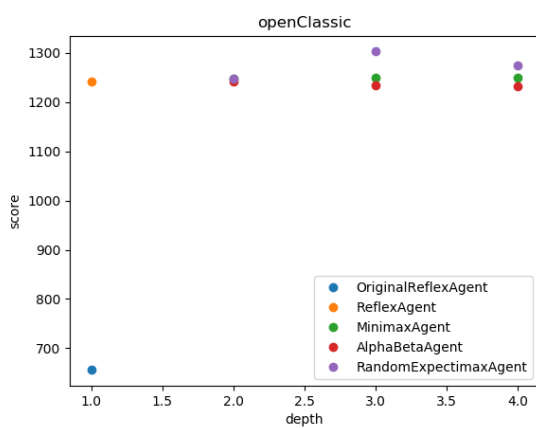
להלן טבלה המסכמת את תוצאת ההרצה על הלוחות השונים. חשוב להדגיש כי תוצאות אלו הן על סמך מדגם הרצה אחד, כאשר כל סוכן רץ על כל לוח 7 פעמים בלבד. בלוחות בהם אין סוכן חד משמעי טוב ביותר, ייתכן כי בהינתן מדגם הרצה אחר יהיו תוצאות אחרות או בהינתן מדגם הרצה גדול יותר, יהיו תוצאות כן חד משמעיות. אלמנט האקראיות הוא רכיב חשוב בניתוח התוצאות. עוד נדגיש כי שני השחקנים OriginalReflexAgent ו- ReflexAgent אשר לא מעמיקים בבדיקה שלהם הציגו בכל המשחקים את התוצאות הגרועות ביותר ועל כן לא נרחיב על ההשוואה ביניהם לבין שאר הסוכנים.

<p>capsuleClassic</p> <table><thead><tr><th>depth</th><th>OriginalReflexAgent</th><th>ReflexAgent</th><th>MinimaxAgent</th><th>AlphaBetaAgent</th><th>RandomExpectimaxAgent</th></tr></thead><tbody><tr><td>1.0</td><td>-250</td><td>-100</td><td></td><td></td><td></td></tr><tr><td>2.0</td><td></td><td></td><td>-400</td><td>180</td><td>-350</td></tr><tr><td>3.0</td><td></td><td></td><td>-220</td><td>350</td><td>-350</td></tr><tr><td>4.0</td><td></td><td></td><td>-250</td><td>-20</td><td>-200</td></tr></tbody></table>	depth	OriginalReflexAgent	ReflexAgent	MinimaxAgent	AlphaBetaAgent	RandomExpectimaxAgent	1.0	-250	-100				2.0			-400	180	-350	3.0			-220	350	-350	4.0			-250	-20	-200	<p>CapsuleClassic:</p> <p>ניתן לראות שבלוח זה יש עדיפות לסוכן AlphaBetaAgent והוא הסוכן שמגיע לתוצאות הטובות ביותר בכל העומקים.</p>
depth	OriginalReflexAgent	ReflexAgent	MinimaxAgent	AlphaBetaAgent	RandomExpectimaxAgent																										
1.0	-250	-100																													
2.0			-400	180	-350																										
3.0			-220	350	-350																										
4.0			-250	-20	-200																										
<p>contestClassic</p> <table><thead><tr><th>depth</th><th>OriginalReflexAgent</th><th>ReflexAgent</th><th>MinimaxAgent</th><th>AlphaBetaAgent</th><th>RandomExpectimaxAgent</th></tr></thead><tbody><tr><td>1.0</td><td>500</td><td>400</td><td></td><td></td><td></td></tr><tr><td>2.0</td><td></td><td></td><td>1450</td><td>1350</td><td>1600</td></tr><tr><td>3.0</td><td></td><td></td><td>1300</td><td>1480</td><td>1800</td></tr><tr><td>4.0</td><td></td><td></td><td>1700</td><td>1500</td><td>1520</td></tr></tbody></table>	depth	OriginalReflexAgent	ReflexAgent	MinimaxAgent	AlphaBetaAgent	RandomExpectimaxAgent	1.0	500	400				2.0			1450	1350	1600	3.0			1300	1480	1800	4.0			1700	1500	1520	<p>contestClassic:</p> <p>בלוח זה אין העדפה ברורה לסוכן ספציפי, או לעומק ספציפי. עבור סוכני AlphaBetaAgent ו-MinimaxAgent התוצאות הטובות מתקבלות בעומק 4, אך לא כך הדבר עבור RandomExpectimaxAgent.</p>
depth	OriginalReflexAgent	ReflexAgent	MinimaxAgent	AlphaBetaAgent	RandomExpectimaxAgent																										
1.0	500	400																													
2.0			1450	1350	1600																										
3.0			1300	1480	1800																										
4.0			1700	1500	1520																										
<p>mediumClassic</p> <table><thead><tr><th>depth</th><th>OriginalReflexAgent</th><th>ReflexAgent</th><th>MinimaxAgent</th><th>AlphaBetaAgent</th><th>RandomExpectimaxAgent</th></tr></thead><tbody><tr><td>1.0</td><td>-50</td><td>400</td><td></td><td></td><td></td></tr><tr><td>2.0</td><td></td><td></td><td>1600</td><td>1550</td><td>1520</td></tr><tr><td>3.0</td><td></td><td></td><td>1350</td><td>1780</td><td>1720</td></tr><tr><td>4.0</td><td></td><td></td><td>1780</td><td>1520</td><td>1800</td></tr></tbody></table>	depth	OriginalReflexAgent	ReflexAgent	MinimaxAgent	AlphaBetaAgent	RandomExpectimaxAgent	1.0	-50	400				2.0			1600	1550	1520	3.0			1350	1780	1720	4.0			1780	1520	1800	<p>mediumClassic:</p> <p>בלוח זה אין העדפה ברורה לסוכן ספציפי, או לעומק ספציפי. עבור סוכני RandomExpectimaxAgent ו-MinimaxAgent התוצאות הטובות מתקבלות בעומק 4, אך לא כך הדבר עבור AlphaBetaAgent.</p>
depth	OriginalReflexAgent	ReflexAgent	MinimaxAgent	AlphaBetaAgent	RandomExpectimaxAgent																										
1.0	-50	400																													
2.0			1600	1550	1520																										
3.0			1350	1780	1720																										
4.0			1780	1520	1800																										



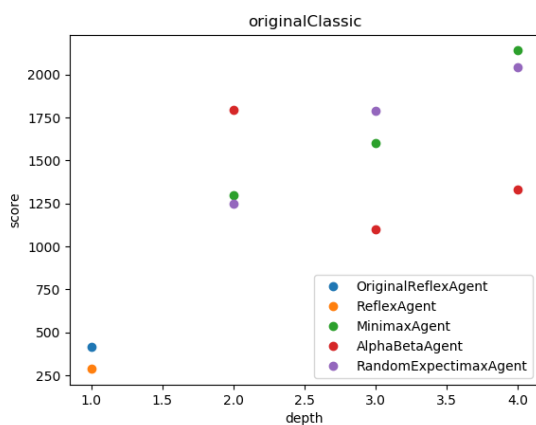
#### minimaxClassic:

בלוח זה אמורה להיות העדפה ברורה לסוכן MinimaxAgent ואכן רואים זאת בעומקים 2 ו-3. בעומק 4 קיבלנו תוצאה לא צפויה, ב-7 הרצות אחרות עם סוכן זה בעומק 4 יצאו תוצאות המהוות שיפור משמעותי לעומת המדגם על בסיסו תוצאות הגרף.



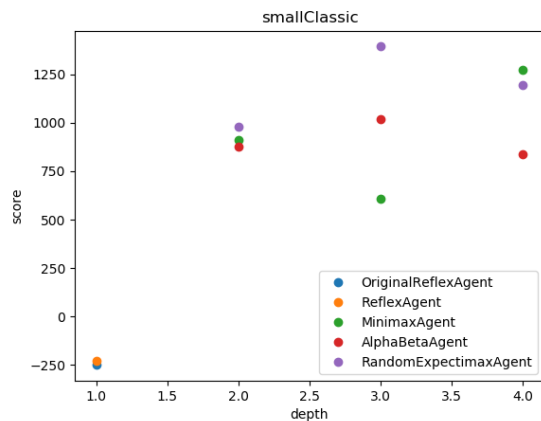
#### openClassic:

בלוח זה התוצאות יצאו דומות. כיוון שאין בו מכשולים, הסיכוי להצליח לחמוק מהרוחות גדול. ניתן לראות יתרון קטן לסוכן RandomExpectimaxAgent על פני האחרים.



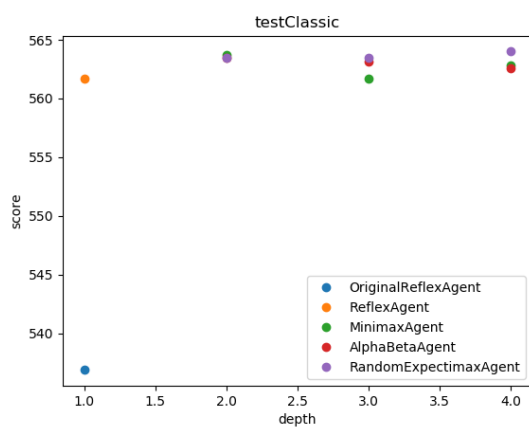
#### originalClassic:

עבור הסוכנים RandomExpectimaxAgent ו-MinimaxAgent קיבלנו שבכל שהעומק גדל כך הביצועים משתפרים. MinimaxAgent הוא הסוכן המוביל ללוח זה.



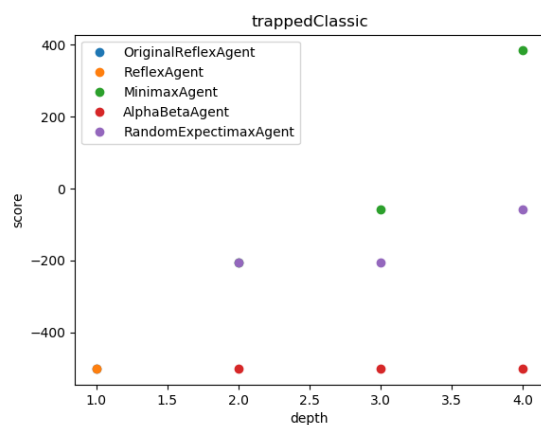
### smallClassic

בממוצע הסוכן הטוב בלוח זה הוא RandomExpectimaxAgent, אך רק בגלל שמתקיימת המגמה עליה דיברנו קודם, בה בעומק 3 סוכן MinimaxAgent מקבל החלטות שגויות בטווח הרחוק יותר. ניתן לראות שבעומק 4 הוא שוב הסוכן המוביל.



### testClassic:

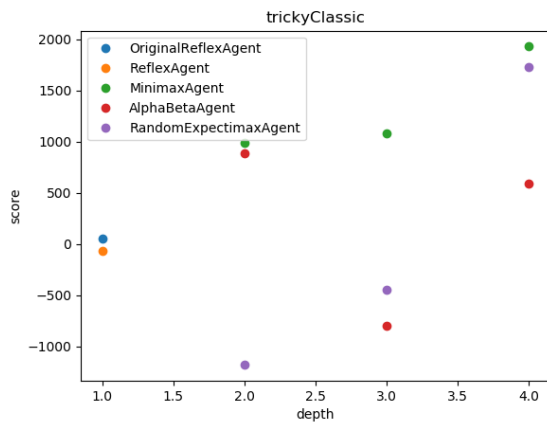
גם כאן תוצאות דומות לכל הסוכנים. זהו לוח קטן ללא מכשולים ועם רוח אחת, לכן הסוכנים כולם יצליחו להימלט ממנה. ההפרש בניקוד נובע מהפרשי זמנים היכול להשתנות ממדגם אחד לשני ולכן בלוח זה אין סוכן עדיף.



### trappedClassic:

כמו שהורחב בסעיף 8, בלוח זה סוכן AlphaBetaAgent, מפסיד תמיד כיוון שגוזם את דרך המילוט שלו. שני הסוכנים האחרים פועלים באותה הדרך ולכן הפרש התוצאות ביניהם הוא עניין מדגמי. נצפה שבלוח גדול יותר, לסוכן RandomExpectimaxAgent תהיה עדיפות.





### trickyClassic:

בלוח זה יתרון בולט לסוכן

MinimaxAgent. אנו מעריכים

שבהינתן יוריסטיקה שמתחשבת

בבריחה מרוחות, היינו מקבלים

תוצאות דומות גם עבור סוכן

AlphaBetaAgent, כי לא היה

מתרחש גיזום יתר.

## ט. תחרות בקורס

השחקן שהוגש לתחרות משתמש ביוריסטיקה אחרת מזה שהשתמשו הסוכנים שהרצנו בשאר התרגיל. כיוון שבשאר היו הרצות רבות אשר הכבידו בכוח החישוב שהן דרשו על המחשבים האישיים, השתמשנו ביוריסטיקה אשר לא מתחשבת ברוחות ובבריחה מהם. היוריסטיקה של השחקן המתחרה היא היוריסטיקה המלאה המפורטת בסעיף ב'.

בתחרות הגבלה של זמן ריצה, מבין האלגוריתמים שפיתחנו alpha-beta הוא המהיר ביותר ולכן הוא האלגוריתם הנבחר. שאפנו לתת לו לפתח עד עומק 4, אך הזמנים היו גבוליים ביחס לדרישות ולכן השחקן אותו אנו מגישים הוא שחקן המתנהל לפי היוריסטיקה הנ"ל (כאמור מפורטת בסעיף ב') ומפתח עד עומק 3. לפי ההבהרה לפיה מספר הרוחות המקסימלי בתחרות הוא 2, שחקן זה עומד בדרישות.