

בינה מלאכותית – תרגיל מספר 2 (דמקה)

חלק א' – הבנת המשחק

1. המבנה של כל מצב (state) במרחב המצבים של המשחק:

self.board – מבנה הלוח הנוכחי.

self.curr_player – צבע השחקן (אדום או שחור).

self.turns_since_last_jump – מספר התורות מאז המהלך האחרון (לכל היותר יהיו 50 תורות במשחק).

2. מידע שנחוץ כדי לייצג כל אופרטור GameMove:

player_type – סוג שחקן מלך שחור (BK) \ מלך אדום (RK) \ שחקן שחור (BP) \ שחקן אדום (RP).

origin_loc – אינדקס (x,y) של השחקן בלוח.

target_loc – אינדקס (x,y) שאליו נרצה להזיז את השחקן (אם מדובר במהלך הכולל כמה קפיצות, האינדקס יהיה המיקום האחרון אליו השחקן מגיע בסוף הקפיצות).

jumped_locs – אם במהלך קיימות קפיצות אז זו רשימה של המיקומים שהשחקן עבר במהלך הקפיצות. מכאן, אם השחקן לא קפץ בכלל הרשימה תהיה ריקה או None ונדע שמדובר בתנועה רגילה (movement) ואם הרשימה לא ריקה אז נדע שמדובר בפעולת קפיצה (capture).

חלק ב' – הבנת השחקן הפשוט

3. הגישה הנאיבית בה נוקט השחקן לשם חלוקת הזמן בין כל k המהלכים היא שבכל תור הזמן שנקבע עבור מהלך ספציפי מתוך K המהלכים הוא הזמן שנותר חלקי מספר התורות הנותרים ועוד מרווח זמן של 0.05.

4. החיסרון בחלוקת הזמן באופן שווה בין k המהלכים ייתכן שקיים מהלך אחד שנדרש עבורו זמן חישוב גבוה יותר מאשר מהלך אחר שנדרש עבורו זמן חישוב קטן יותר והדבר יוביל לפספוס מהלך אפשרי טוב יותר עבור המהלך שנדרש עבורו זמן חישוב גבוה יותר.

5. הפונקציה היוריסטית שבה משתמש השחקן היא utility שהיא בעצם מחשבת הפרש בין כמות הכלים של השחקן שלי לבין השחקן היריב. הסיבה שבחרו לממש את הפונקציה היוריסטית הזו עבור חישוב המהלך הבא היא לתת עדיפות לבחירת מהלכים שמובילים להפרש גדול בכלים בין השחקן שלי לבין השחקן היריב. בכך נשיג יותר קפיצות עבור השחקן שלי וזה יניב סיכוי גבוהה לנצח.

חלק ג' – שיפור השחקן

6. היוריסטיקה שבחרנו מביאה משקל על פי סוגי החיילים הקיימים בלוח ומתחשבת במיקום שלהם. נוסחה יוריסטית:

$$H(\text{player}=b/r)=1$$

חייל אדום או שחור יקבל את המשקל 5.

$$H(\text{player}=B/R)=1.5$$

מלך אדום או שחור יקבל את המשקל 7.5.

$$H(\text{player}= B/b/R/r, \text{row_number}=\text{last_row})= 0.8$$

מלך שחור או חייל שחור הנמצא בשורה במיקום 0 יקבל את המשקל 5. הסיבה ללקיחת פרמטר זה היא שנרצה שהחיילים שלנו ישאפו להגיע לשורה האחרונה כי כאשר חיילים שלי נשארים בשורה האחרונה כך אני שומרת עליהם מוגנים וגם מונעת שהיריב יהפוך למלך.

$$H(\text{player}= B/b/R/r, 2<\text{row_number}<5 \ \&\& \ 1<\text{column_number}<6)= 0.5$$

חייל או מלך הנמצאים באמצע הלוח, כלומר בין עמודות 2 ו-5 ובין שורות 3 ו-4. נשאף שהחיילים יגיע לאמצע הלוח כי ממיקום זה יש להן דרכי פעולות רבות יותר להתקדם לצד היריב.

$$H(\text{player}= B/b/R/r, 2<\text{row_number}<5 \ \&\& \ (\text{column_number}<2 \ || \ \text{column_number}>5))= 0.1$$

חייל או מלך הנמצאים בשולי אמצע הלוח, כלומר בין שורות 3 ו-4 ובין עמודות 2 ו-5. כאשר חיילים נמצאים במיקום זה הם נמצאים במיקום מוגן ובנוסף יכולים לתקוף.

$$H(\text{player}== B/b/R/r \ \&\& \ \text{vulnerable})= -0.6$$

חייל או מלך הנמצאים במקום פגיע, כלומר נמצאים במיקום שהיריב בתור הבא יקפוץ מעלי בוודאות. נתן משקל נמוך לחייל הנמצא במקום פגיע כי לא נרצה שבעתיד נגיע למצב כזה שוב.

$$H(\text{player}== B/b/R/r \ \&\& \ \text{protected})= 0.6$$

חייל או מלך שהם מוגנים כלומר לא ייתכן שהיריב יקפוץ מעליהם בשלב הבא. נשאף שהחיילים יגיעו למצב מוגן כי כך נשמור על כמה שיותר חיילים חיים במשחק.

המשקלים והיוריסטיקה שבחרנו מבוססים על פרויקט שמצאנו בגיטהאב. ביצענו התאמות על פי יחס המשקלים של השחקן הפשוט.

<https://github.com/kevingregor/Checkers/blob/master/Final%20Project%20Report.pdf>

7. שיפרנו את הזמנים באמצעות 4 פונקציות שונות:

פונקציה 1 – מצב בלוח בו קיימת לפחות קפיצה אחת.

בפונקציה זו נחזיר את הפעולה שמניבה הכי הרבה קפיצות ולא נבצע חיפוש minmax ובכך חסכנו זמן חיפוש.

פונקציה 2 – מצב בלוח בו קיים לפחות חייל אחד יהיה מותקף, כלומר מצב שהיריב בתורו יקפוץ מעל אותו חייל ובנוסף לחייל קיימת דרך מילוט. במצב זה ניתן זמן מקסימלי שהוא 180% מהזמן הממוצע לפעולה.

פונקציה 3 – מצב בו בתור הבא היריב יקפוץ מעל חייל שלי ובמצב כזה ניתן זמן נוסף על מנת למקסם את הפעולות העתידיות כי הגענו למצב בו כמות החיילים פחתה. הזמן שניתן הוא 150% מהזמן הממוצע לפעולה.

פונקציה 4 – מצב בו קיימים לפחות שני חיילים הנמצאים במרכז הלוח. שליטה במרכז הלוח מהווה יתרון תמרון ותקיפה לכן ניתן יותר זמן במצב כזה. הזמן שניתן הוא 130% מהזמן הממוצע לפעולה.

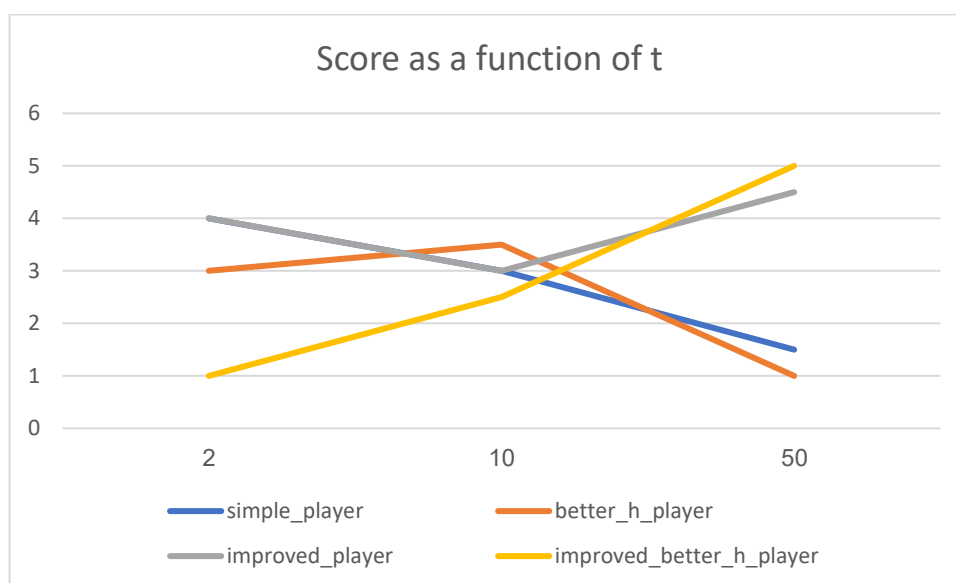
קיים תיעדוף לפי סדר הפונקציות. המוטיבציה מאחורי השיטה היא להשקיע במצבים קריטיים בהם חיילים יכולים להיות מותקפים ובמצבים בהם השחקן יכול לתקוף את היריב. נשקיע פחות זמן במצב בו טריוויאלי של תקיפה תקיפה וודאית (מתוקף חוקי המשחק).

השחקן הפשוט מחלק את זמני הריצה בין k המהלכים באופן שווה ולא נותן חשיבות גבוהה יותר למהלכים בהם נדרש זמן חישוב גדול יותר ואינו חוסך זמן במצבים בהם ניתן לצמצם את זמן החישוב. לעומת השיטה בה השתמשנו שכן מתחשבת במצבים אלו ולכן תהווה שיפור משמעותי לעומת השחקן הפשוט.

חלק ד' – ניסויים, תוצאות ומסקנות

10. א. גרף המתאר עבור כל אחת ממגבלות הזמנים את הניקוד הכולל של כל שחקן כפונקציה של מגבלת הזמן:

T=50	T=10	T=2	
1.5	3	4	simple_player (1)
1	3.5	3	better_h_player (2)
4.5	3	4	improved_player (3)
5	2.5	1	improved_better_h_player (4)



ב. השוואת ביצועי שחקנים simple_player, improved_player (עם היוריסטיקה הפשוטה), לשחקנים better_h_player, improved_better_h_player (עם שיפור היוריסטיקה שלנו) ביחס למגבלת הזמן ע"י ניתוח של מגמות השיפור והדעיכה בניקוד הכולל של כל אחד מהשחקנים:

כאשר $T=2$ השחקנים עם היוריסטיקה הפשוטה מנצחים. בשלב זה שיפור היוריסטיקה לא מועילה לשיפור השחקן הפשוט.

כאשר $T=10$ better_h_player מנצח את כולם אבל improved_better_h_player מפסיד לכולם. בזכות שיפור היוריסטיקה השחקן ניצח בשלב זה.

כאשר $T=50$ better_h_player מפסיד לכולם ו-improved_better_h_player, improved_player דומים בתוצאות הגבוהות שלהם לכן ניתן להסיק בשלב זה שהיוריסטיקה ששיפרנו לא משפרת משמעותית את השחקן אך שיפור הזמנים מקנה לשחקן סיכוי גבוה יותר לניצחון.

ב. השוואת ביצועי שחקנים `improved_player`, `improved_better_h_player` (עם שיפור הזמנים), לשחקנים `better_h_player`, `simple_player` (עם פונקציית זמן נאיבית) ביחס למגבלת הזמן ע"י ניתוח של מגמות השיפור והדעיכה בניקוד הכולל של כל אחד מהשחקנים:

כאשר $T=2$ השחקנים `improved_player`, `simple_player` מנצחים. למרות שיפור הזמנים השחקן המשופר עדיין לא מנצח את השחקן הפשוט.

כאשר $T=10$ השחקן עם היורסיטקה המשופרת מנצח את כולם. גם בשלב זה השחקנים `improved_player`, `simple_player` מגיעים לאותן תוצאות.

כאשר $T=50$ ניתן לראות בצורה מובהקת כי בזכות שיפור הזמנים השחקנים `improved_player`, `improved_better_h_player` הגיעו לתוצאות גבוהות יותר לעומת שני השחקנים שמשחקים לפי פונקציית חלוקת זמנים נאיבית. כיוון שבשלב זה ניתן 50 שניות עבור k מהלכים, לפונקציית חלוקת הזמנים שקבענו היה יותר משמעות וכיוון שהיא אכן תורמות לשחקן לבצע מהלכים בצורה מחושבת יותר (מבחינת זמנים) שני השחקנים הללו ניצחו.