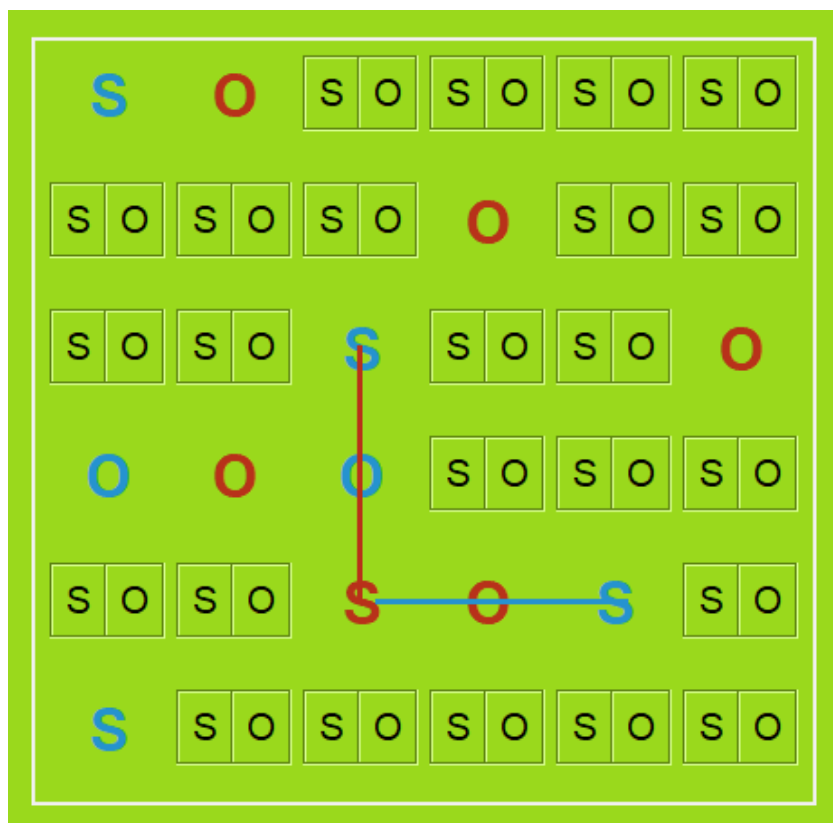


SOS

מגישים: תום נחום, עומר סלמן, אדם מונסונגו



INTRODUCTION TO ARTIFICIAL INTELLIGENCE 67842

Final Project



תוכן עניינים

3	הקדמה
3	מטרת הפרויקט
3	מהלך המשחק
4	אסטרטגיות
5	השיטה
5	המודל
6	סוכנים
6	Minimax סוכן
7	Alpha-beta pruning עם Minimax סוכן
7	Expectimax סוכן
7	יריבים
7	יריב random
8	יריב greedy
8	היוריסטיקות
8	היוריסטיקת ניקוד
8	היוריסטיקת חסימה
9	היוריסטיקת חיסור
9	היוריסטיקה משולבת
9	היוריסטיקת ניקוד מינימלי של היריב
10	תוצאות ומסקנות
10	זמני ריצה
12	ניקוד ממוצע
15	משחק נגד שחקן חמדן
16	סיכום
17	ביבליוגרפיה

הקדמה

מטרת הפרויקט

מטרתנו בפרויקט זה היא לבנות סוכן AI שינצח במשחק SOS בהסתברות גבוהה מאוד (סוכן זה יהיה סוכן חיפוש בהתאם למבנה המשחק עליו נפרט בהמשך), נשתמש באלגוריתם minimax ונשווה בין היוריסטיקות שונות לניקוד מצבים. כמו כן מכיוון שמספר המצבים במשחק זה גדול מאוד, נשתמש גם באלגוריתם Alpha-beta pruning על מנת לקצץ את עץ החיפוש שיווצר במהלך המשחק וכך להפחית את זמן הריצה של הסוכן עד לפעולה, גם פה נשווה בין היוריסטיקות מיון קודקודים שונות כדי למזער אפילו יותר את זמן הריצה של הסוכן ולבחון את הקשר בין השניים. בסופו של דבר נרצה למצוא אסטרטגיה אופטימלית של סוכן אשר גם ינצח בהסתברות גבוהה כל שחקן וגם יעשה זאת בזמן קצר לכל צעד.

מהלך המשחק

המשחק SOS הוא משחק סכום אפס סימטרי, דינמי, בעל ידיעה שלמה של שני שחקנים שמשחקים אחד נגד השני. המשחק מתנהל על גבי לוח בגודל משתנה המחולק למשבצות (לפחות בגודל 3×3). כל שחקן בתורו צריך לשים במשבצת פנויה בלוח את האות S או את האות O. כאשר שחקן בתורו מצליח להשלים את המילה SOS בשורה, עמודה או אלכסון הוא זוכה בנקודה על כל SOS שהשלים בצעד זה ובתור נוסף. המשחק מסתיים כאשר הלוח מלא והמנצח הוא השחקן עם מירב הנקודות. כמו כן, כמו במשחקים דומים (למשל איקס עיגול) גם כאן ייתכן מצב של תיקו בו לשני השחקנים סכום נקודות זהה.

אסטרטגיות

נשים לב כי במשחק זה, בניגוד לאיקס עיגול, כל שחקן יכול לשים אותיות זהות לשחקן היריב ולכן עליו להיזהר לא "להכין את הקרקע" כך שבתור הבא היריב יוכל להשלים sos בזכות אותיות שהניח, כלומר ישנן אסטרטגיות שנראות כי בסופו של דבר יובילו להרבה SOS-ים אבל הם לא בהכרח יקרו בתור שימקסם את הניקוד של השחקן שמשתמש באסטרטגיות אלו.

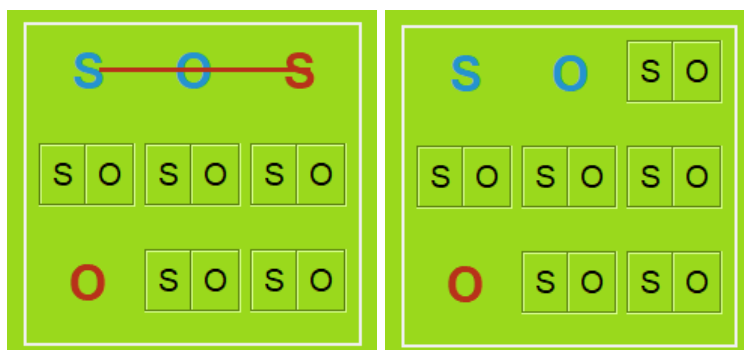
אסטרטגיה אחת מאוד הגיונית היא לעולם לא לשים במשבצת אות שתכין SOS לשחקן היריב.

נראה דוגמה:

השחקן הכחול רוצה להשלים SOS ולכן שם O ליד S, אולם לאחר שהתור התחלף דווקא השחקן האדום יכול לנצל את האותיות שהניח קודם לכן בסדר הנ"ל השחקן הכחול ולהשלים SOS וכך בעצם לזכות בניקוד "על חשבון" יריבו - השחקן הכחול, לכן נסיק כי אלו פעולות שנרצה להימנע מהן.

צעד של השחקן האדום

צעד של השחקן הכחול



מצד שני, האסטרטגיה אשר פעל לפיה השחקן האדום - בכל תור לשאוף להשלים כמה שיותר SOS-ים בצורה חמדנית נראית כמו אסטרטגיה מוצלחת וכמובן שנבחן זאת בהמשך.

נשים לב בנוסף כי במקרה של שני שחקניים אופטימליים המשחקים אחד נגד השני תוצאה של תיקו הינה אפשרית והגיונית למדי (בדומה למשחק איקס עיגול).

השיטה

על מנת לפתור את הבעיה נשתמש באלגוריתמים מתחום החיפוש שנלמדו בקורס (לדוגמא: expectimax, minimax) תוך אופטימיזציה באמצעות α - β pruning עם שימוש בהיוריסטיקות שונות (על פי סדר המעבר על הקודקודים, ניקוד של מצבים וכו'...). כיוון שהמשחק המדובר הינו משחק סכום אפס עם שני שחקנים, נובע כי עץ החיפוש הוא עץ משחק שבו כל קודקוד הוא מצב של המשחק והעלים הם מצבים שבהם המשחק הסתיים (כפי שלמדנו בקורס). כל שחקן מעריך את הניקוד של העלים ובהנחה שהשחקן רציונלי הוא ישאף למקסם את הניקוד שלו על מנת לנצח. על כן נובע כי מבנה המשחק מתאים לבעיית חיפוש ואלגוריתמי החיפוש שהצענו (אשר נלמדו בקורס) על מנת להתמודד עם הבעיה אכן מתאימים לפתרונה. את תוצאות הסוכנים שנממש נבחן באמצעות הניקוד (מספר ה-SOS אשר השלימו) וזמן הריצה עד לסיום המשחק. נשווה פרמטרים אלו בין מספר אלגוריתמי חיפוש שונים תוך שימוש בהיוריסטיקות שונות ולבסוף נסיק מהו האלגוריתם והיוריסטיקה האופטימליים.

המודל

לצורך ההשוואה בין היוריסטיקות החלטנו לכתוב בעצמנו גרסה של המשחק SOS. המשחק נכתב בפייתון 3.7, והגרפיקה באמצעות הספרייה של tkinter. גרסה זו היא מאוד ידידותית למשתמש, דינמית וגמישה, כל הפרמטרים למשחק נבחרים בצורה פשוטה בעזרת ה-GUI בלבד. נדרש רק להריץ את הקובץ sos.py ולבחור את הגדרות המשחק כרצונך, ההגדרות הן: גודל הלוח - ניתן לבחור גודל לוח באמצעות כפתור תפריט המכיל גדלים בין 3X3 ל-9X9, הלוח תמיד ריבועי, כחלק מהגדרות המשחק. מי משחק - ניתן לבחור לשחק אחד נגד השני (שני אנשים), או לבחור ששני סוכני AI ישחקו אחד נגד השני. לכל שחקן AI ניתן לבחור אסטרטגית פעולה (היוריסטיקה) בעזרת כפתור ה-dropdown.

כדי להשיג את הנתונים לגרפים שנראה בהמשך כתבנו קוד ללא גרפיקה שמריץ מספר משתנה של משחקים (למשל 10 משחקים ברצף) ומחשב עבורם ממוצע של זמן ריצה, ממוצע של ניקוד וכמו כן גם סופר כמה פעמים ניצח כל שחקן.

סוכנים

ראשית נקדים לומר שבכל אחד מהאלגוריתמים הבאים מימשנו בנוסף פעולה רנדומלית לא טריוויאלית. במהלך הניסויים גילינו שבמצב שבו אין אף פעולה טובה יותר משאר הפעולות האלגוריתם תמיד יבחר את הפעולה האחרונה וזהו מצב שלא מוביל לשום דבר ולכן נרצה להימנע ממנו.

בנוסף לכך, נשים לב כי בחירת פעולה רנדומלית עלולה להכין את הקרקע לכך שהיריב ינצח. לכן כתבנו פונקציה קצרה שעוברת על כל הפעולות החוקיות ובחרת מהן פעולה רנדומלית שאינה מייצרת מצב הכנה ל SOS של היריב (מצב בו היריב בתור הבא ישלים SOS). פעולה שכזאת נבחרת רק במצב שהיורסיטיקה הראשית לא מצאה פעולה אופטימלית כלל (כל הפעולות החזירו את אותו ניקוד).

במסמך זה אנו מתייחסים לפעולה של שחקן מקסימום ופעולה של שחקן מינימום כעומק אחד בעץ החיפוש.

בנוסף, ביצענו התאמה של האלגוריתמים השונים לחוקי המשחק SOS: כשיורדים בעומק עץ החיפוש לא בהכרח מתחלף השחקן (מכיוון שכאשר שחקן מצליח ליצור רצף של SOS הוא מקבל תור נוסף). לכן באלגוריתם המינימקס לדוגמה אם שחקן מקסימום קיבל SOS נשלח לעומק הרקורסיה שוב את שחקן המקסימום.

Minimax סוכן

אלגוריתם מינימקס הוא אלגוריתם חיפוש ששואף למזער את ההפסד המקסימלי וגם למקסם את הרווח המינימלי. נניח בלי הגבלת הכלליות שאנחנו מחפשים למקסם ערך מסויים, אלגוריתם המינימקס יחפש לנו החלטה שתניב רווח מקסימלי, אם נעמיק הוא יבחר רווח מקסימלי מתוך כל ההפסדים המקסימלים ובכך ימזער את ההפסד (תוך הסתכלות על פעולות היריב). האלגוריתם ממשיך כך ככל שיורדים בעומק החיפוש ובהתאם לעומק החיפוש נשתמש בפונקציית הערכה כדי לתת ניקוד למצב שהגענו אליו בעץ, בהתאם להיורסיטיקה שהגדרנו עבור סוכן המינימקס.

סוכן Minimax עם Alpha-beta pruning

אלגוריתם זהה לאלגוריתם מינימקס אבל עם קיצוץ של ענפים על מנת לא לעבור על כל עץ החיפוש שנוצר ובכך לקצר את זמן הריצה. תוך כדי הריצה נוכל ללמוד אם יש ענף שלא יצליח למצוא שיפור לתוצאה שאנחנו מחזיקים בה כרגע וכך נוכל לחסוך את כל החיפוש בענף זה. בנוסף לכך הוספנו לאלגוריתם זה היוריסטיקת מיון - אם נמייין את הבנים של כל קודקוד לפי היוריסטיקת מיון טובה נוכל בהסתברות גבוהה לחסוך המון ענפים לעבור עליהם ובכך למנוע הרבה כניסות רקורסיה מיותרות לעומק העץ. נשים לב כי עבור שחקן מקסימום נמייין בסדר יורד את המצבים ועבור שחקן מינימום נמייין בסדר עולה. נראה בהמשך כי התוצאות מדברות בשם עצמן.

סוכן Expectimax

אלגוריתם זה בדומה לאלגוריתם מינימקס הוא אלגוריתם חיפוש כלל החלטה ששואף למקסם את הרווחים שלו. אולם הפעם, בניגוד למינימקס, כאשר נעמיק נבחר רווח מקסימלי מתוך ערכי התוחלת של שחקן המינימום. כלומר הפעם לא נהיה פסימיים ונבחר למזער את ההפסד המקסימלי אלא נבחר למקסם את הרווח אל מול תוחלות ההפסדים.

יריבים

יריב random

היריב הרנדומלי מסתכל על כל הפעולות האפשריות בלוח ובוחר בהסתברות אחידה פעולה אקראית מביניהן.

יריב greedy

היריב החמדני הוא יריב שהמטרה שלו היא למקסם את הניקוד שלו בהינתן מצב נוכחי של לוח. היריב מסתכל רק על מצב הלוח הנוכחי ומנסה באותו הרגע למקסם את הניקוד שלו. (אפשר לחשוב על זה בתור אלגוריתם מינימקס בעומק חצי עם היוריסטיקה שרוצה למקסם את ניקוד היריב).

היוריסטיקות

היוריסטיקת ניקוד

ע"פ היוריסטיקה זו השחקן ישאף במהלכו להשלים כמה שיותר רצפי SOS ובכך להרוויח כמה שיותר נקודות, היוריסטיקה זו מתאימה לאסטרטגיה חמדנית ואגרסיבית של השחקן. היוריסטיקה זו מחזירה בעצם את הניקוד של השחקן (מספר ה SOS-ים שהוא השלים בתור זה). באלגוריתם Minimax שחקן המקסימום שואף למקסם את הניקוד שלו בעוד שהוא מניח שהיריב שואף למזער את הניקוד שלו.

היוריסטיקת חסימה

ע"פ היוריסטיקה זו השחקן ישאף לחסום כמה שיותר את היריב, כלומר יבחר משבצות לשבץ בהן S או O על מנת למנוע מהיריב להשלים SOS באותו מקום בתורו, על כן ע"פ הגדרתה היוריסטיקה זו מתאימה לאסטרטגיה הגנתית ופסיבית של השחקן. היוריסטיקה זו מחזירה ציון נמוך לכל מצב של SO_ או S_S (כמובן גם לכל מצב סימטרי שקול), על כן באלגוריתם Minimax שחקן המקסימום יתן ציון נמוך למצבים אלו. מימוש: סכמנו את המקרים בהם היריב יכול להשלים SOS והחזרנו 1 חלקי הסכום. לכן ככל שיש יותר מצבים ליריב להשלמת SOS המצב יקבל ציון נמוך יותר. אם אין מצבים כאלו כלל בלוח נחזיר 2 (וזוה מצב אופטימלי מבחינת היוריסטיקה). באופן זה קיבלנו טווח ערכים עקבי עבור הניקוד שהיינו רוצים להעניק לכל מצב.

היוריסטיקת חיסור

ע"פ היוריסטיקה זו השחקן יבחר את המהלך שבו הוא ממקסם את מספר הנקודות שלו פחות מספר הנקודות של היריב, כלומר ישנה התחשבות גם במספר הנקודות שהיריב ירוויח עקב בחירת המהלך הנכחי, זאת בניגוד להיוריסטיקת הניקוד על-פיה בחירת המהלך מתבצעת אך ורק באופן חמדני על מנת למקסם את נקודותיו תוך התעלמות מניקוד היריב.

היוריסטיקה משולבת

היוריסטיקה זו שואפת לשלב בין היוריסטיקת הניקוד להיוריסטיקת החסימה באופן הבא:
השחקן ישאף תמיד למקסם את הניקוד שלו (חמדני) אך במצבים בהם לא יוכל להשיג ניקוד גבוה יותר, יעבור לאסטרטגיית החסימה וישאף לחסום את היריב כמה שאפשר.

היוריסטיקת ניקוד מינימלי של היריב

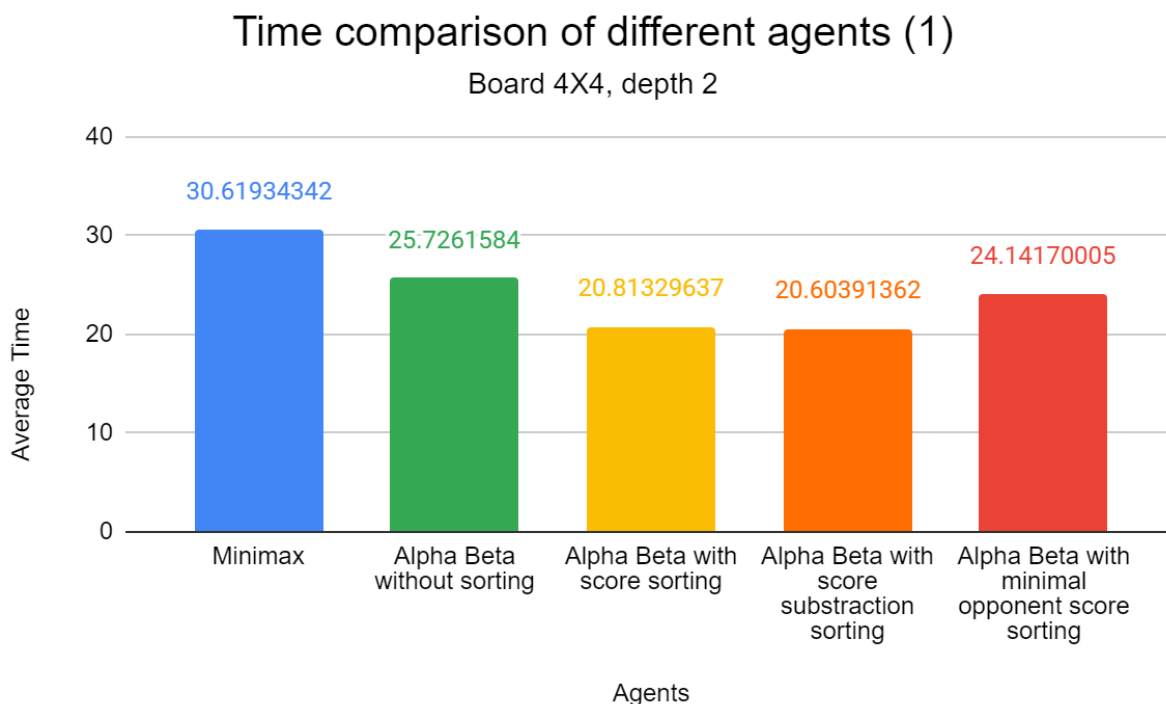
היוריסטיקה זו מנסה למזער את הניקוד של היריב. אפשר לחשוב עליה בתור פונקציית חסימה עם זמן ריצה יותר מהיר שכן גם היוריסטיקה זו תשאף לכך שהיריב לא יצליח להשלים SOS-ים אבל הצורה שבה היא עושה זאת שונה ולכן התוצאות גם כן שונות.

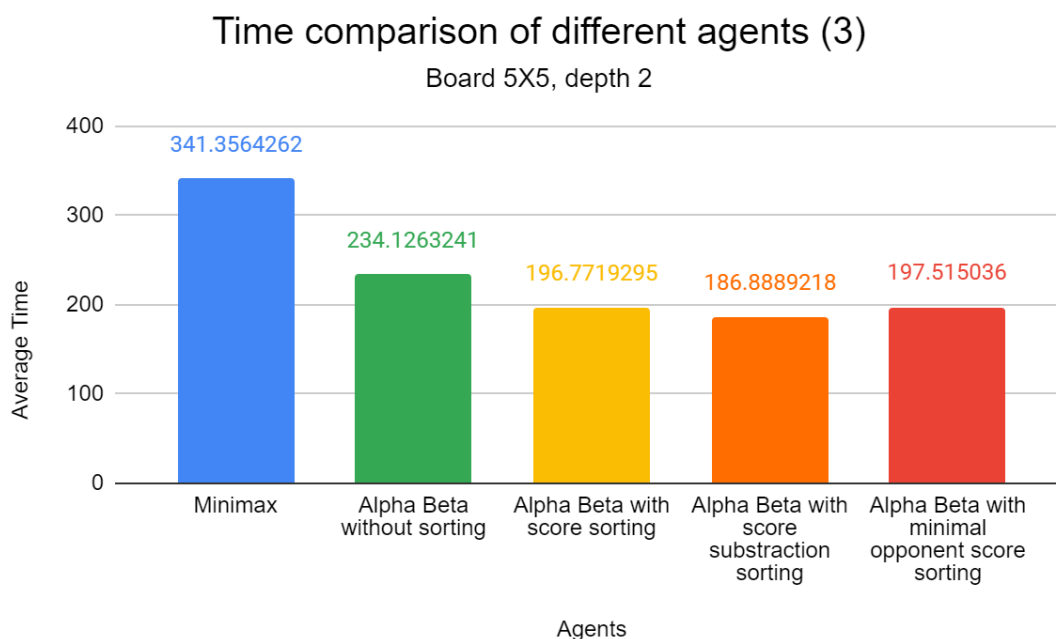
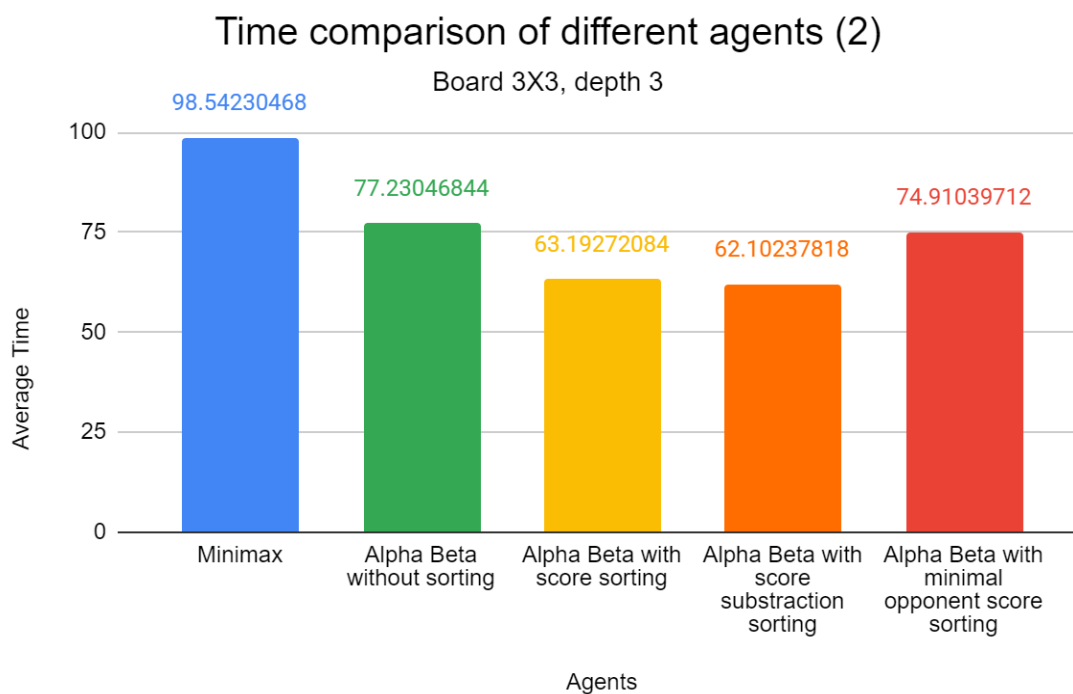
תוצאות ומסקנות

זמני ריצה

נתחיל עם השוואת זמני ריצה בין האלגוריתמים השונים. ההשוואה תתבצע עם פונקציית ניקוד זהה לכולם (היוריסטיקת הניקוד הסטנדרטית), אבל עם היוריסטיקת מיון בנים שונות באלגוריתם Alpha-beta pruning.

נשים לב כי בגרפים הבאים אנחנו לא משתמשים בהיוריסטיקת חסימה למיון הבנים מכיוון שזמן הריצה שלה גדול מידי. היוריסטיקה זו חייבת לסרוק את הלוח ולחפש מצבים "מסוכנים" ולכן אנחנו לא חושבים שזמן הריצה הגבוה הוא בעיה של המימוש הספציפי שלנו אלא בעיה שנובעת מהטבע של היוריסטיקה. נציין כי בחישובי זמן הריצה נבדוק וננתח את התוצאות עבור עומק חיפוש של 2 ומעלה כיוון שבעומק 1 המקרה פשוט מדי ואין שינויים מהותיים בזמני הריצה.

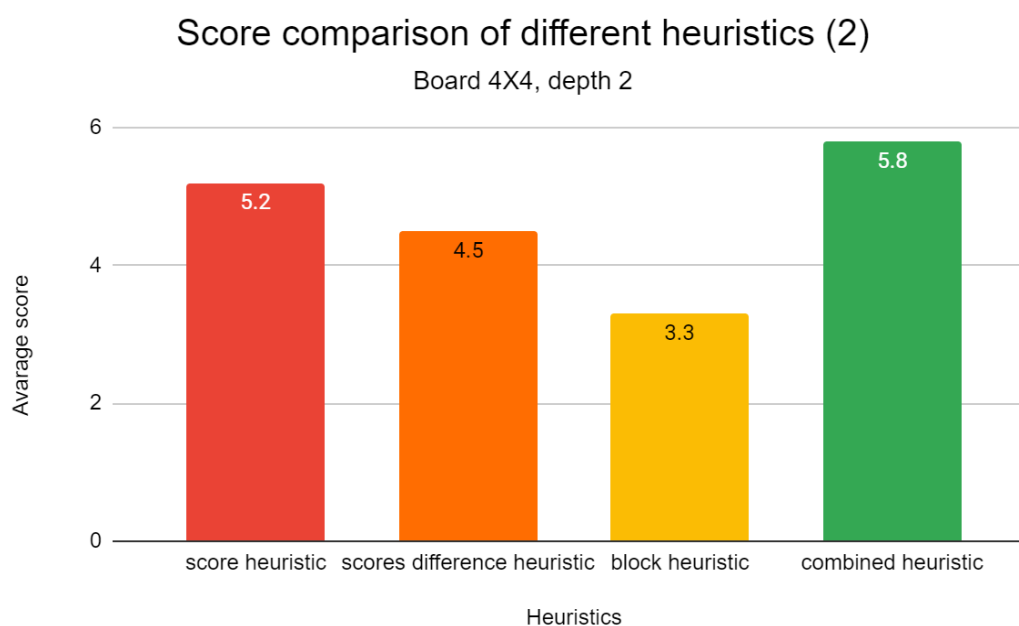
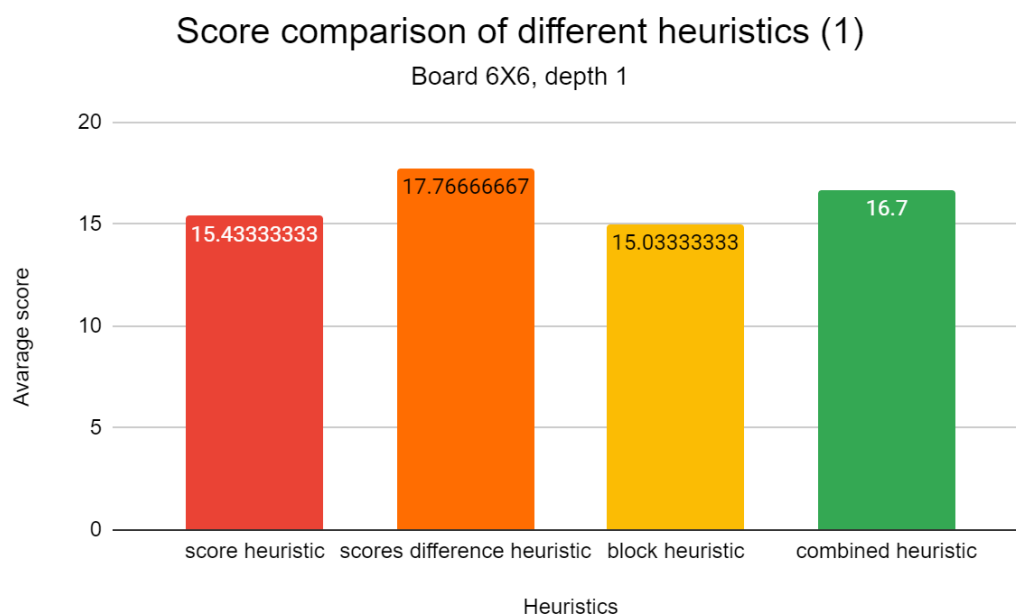




נשים לב מהתוצאות שאנחנו מצליחים לשפר משמעותית את זמני הריצה של אלגוריתם Minimax באמצעות אלגוריתם Alpha-beta pruning עם מיון בנים באמצעות היוריסטיקות הניקוד ובאמצעות היוריסטיקת החיסור. בסך הכל היוריסטיקות המיון כחלק מאלגוריתם Alpha-beta pruning שיפרו את זמן הריצה ב-33% לעומת אלגוריתם Minimax אשר לא מבצע מיון לבנים.

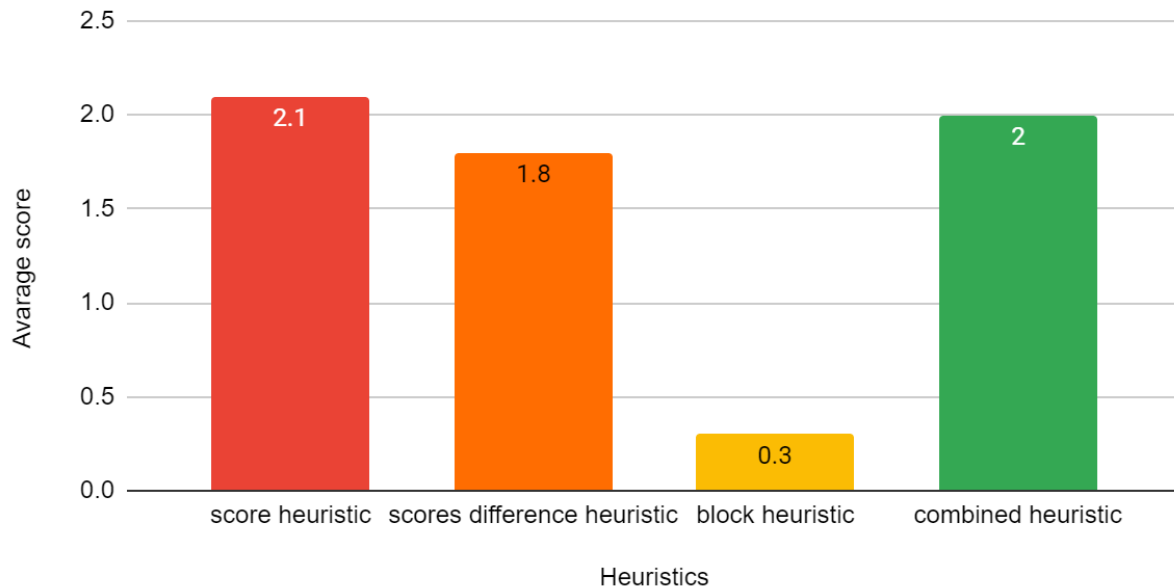
ניקוד ממוצע

נמשיך עם השוואת הניקוד הממוצע בעומקים שונים. נשים לב כי הניקוד הממוצע שונה בין הגרפים השונים בהתאם לגודל הלוח, ולא כתוצאה מעומק החיפוש. הניקוד מחושב לפי מספר ה-SOS-ים שהסוכן הצליח להשלים במהלך המשחק.



Score comparison of different heuristics (3)

Board 3X3, depth 3

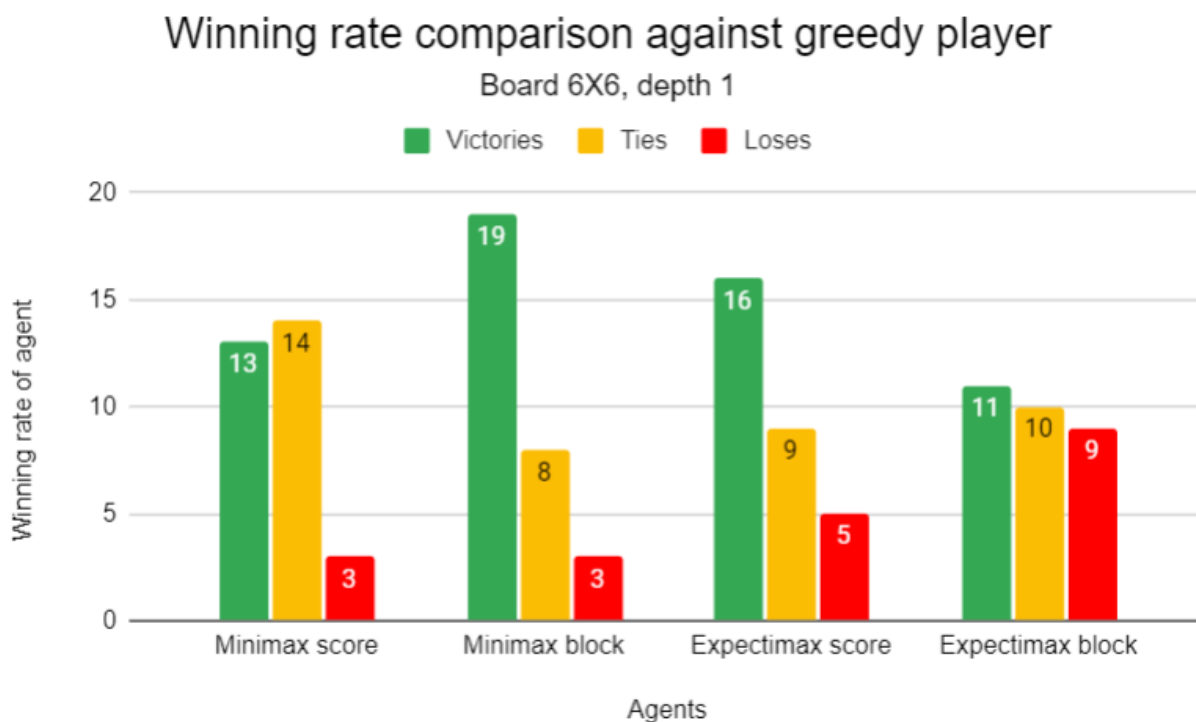


כאן נשים לב כי בעומקים שונים ישנן היורסטיקות מנצחות שונות. בעומק אחד היוריסטיקות הן פחות או יותר שוות מבחינת ביצועים. ככל שנכנסים עמוק יותר בעץ שמים לב שהיוריסטיקת החסימה לבדה מאכזבת מבחינת הניקוד. בסופו של דבר אפשר להסיק מהגרפים כי ישנן 3 היוריסטיקות שנתייחס אליהן כמוצלחות בצורה עקבית: היוריסטיקת הניקוד, היוריסטיקת החיסור והיוריסטיקה המשולבת. בהמשך נבחן שוב את היוריסטיקת החסימה נגד שחקן לא טריוויאלי.

משיקולים קומבינטוריים, ניתן להבין כי מספר גבוה של רצפי SOS (ביחס לגודל הלוח) מרמז לניצחון, לכן מספר גבוה של רצפי SOS בממוצע עבור שחקן גורר הסתברות גבוהה לניצחון. בגרפים לעיל מספר רצפי SOS גבוה ובהתאם הסוכן שלנו ניצח ברוב המוחלט של המשחקים.

משחק נגד שחקן חמדן

נקדים לומר ששני הסוכנים המוצגים מטה מנצחים שחקן רנדומלי ב-100% מהמקרים. כעת נתבונן באחוזי ההצלחה נגד שחקן חמדן, כלומר נאתגר את הסוכנים שלנו מול שחקן יותר מתוחכם. הנתונים נלקחו מתוך 30 משחקים.



מתוך הגרף אנו רואים כי שתי היוריסטיקות מצליחות לנצח גם שחקן חמדן בהסתברות גדולה מחצי. נשים לב כי נגד השחקן החמדן אנחנו רואים הרבה יותר משחקים שהסתיימו בתוצאה של תיקו והדבר עקבי עם ההנחה שלנו כי שני שחקנים אופטימליים יגיעו לתיקו בהסתברות גבוהה (השחקן החמדן יותר קרוב יותר לשחקן אופטימלי מאשר השחקן הרנדומלי). ניתן לראות כי אלגוריתם Minimax מניב תוצאות טובות יותר מאשר אלגוריתם Expectimax וגם מסקנה זו עקבית עם ההנחות שלנו בהינתן שהשחקן היריב הוא כעת קרוב יותר לשחקן אופטימלי.

סיכום

בפרויקט זה הצגנו את המשחק SOS ואסטרטגיות לניצחון בו. כתבנו סוכני AI עם היוריסטיקות שונות במטרה למצוא סוכן אופטימלי גם מבחינת איכות התוצאות וגם מבחינת זמן הריצה.

ניתוח הנתונים הוביל למסקנות הבאות:

ראינו כי סוכן ה-Alpha-beta עם היוריסטיקת מיון הוא המהיר ביותר (כאשר היוריסטיקות המיון המהירות ביותר הן הניקוד והחיסור). סוכן זה מהיר יותר ב-33% לעומת סוכן ה-Minimax. השיפור בזמן הריצה נובע מ-Alpha-beta pruning ומהיוריסטיקת מיון הבנים. היוריסטיקות ההערכה שהניבו את התוצאות המרשימות ביותר הן: היוריסטיקת הניקוד, היוריסטיקת החיסור וההיוריסטיקה המשולבת. קשה לומר איזו מבינהן הייתה הטובה ביותר שכן התוצאות היו צמודות ולא עקביות בין לוחות שונים ועומקי חיפוש שונים.

הרצנו את הסוכנים השונים נגד יריב אקראי ונגד יריב חמדני, וראינו כמצופה כי הסוכנים הניבו תוצאות פחות טובות נגד השחקן החמדני' למרות זאת עדיין הסוכנים שלנו הניבו תוצאות מרשימות וניצחו גם אותו (או הניבו תיקו) ברוב המשחקים.

ביבליוגרפיה

1. Playing the SOS Game Using Feasible Greedy Strategy
<https://journal.binus.ac.id/index.php/commit/article/view/6167/3806>
2. SOS wikipedia
[https://en.wikipedia.org/wiki/SOS_\(game\)](https://en.wikipedia.org/wiki/SOS_(game))