שאלות כלליות על המשחק:

עבור ו נגדיר פונקציית עוקב באופן הבא:

2.

1. s הוא מצב ניצחון עבור שחקן i אם"ם :

2. s הוא מצב תיקו אם"ם :

3.

נפריד למקרים:

עבור השורש , מקדם הסיעוף יהיה 4 משום שממצב התחלתי יהיה אפשר להתקדם לכל היותר ל4 כיוונים אפשריים.

עבור שאר הצמתים – מקדם הסיעוף יהיה 3 משום שממב ביניים יהיה אפשר להתקדם לכל היותר ל 3 כיוונים אפשריים (לא ניתן לחזור על מיקום שממנו הגעת)

שאלות על SimplePlayer:

4.

אסטרטגיית השחקן היא לבחור בצעד בו מספר הצעדים העתידיים הוא המינימלי בתנאי שהצעד שנבחר לא גורם להפסד השחקן. המשמעות היא שהאסטרטגיה של השחקן היא בכל צעד לנסות להשאר כמה שיותר קרוב לאזור בו השחקן היה בצעדים הקודמים ובעצם לא להשאיר אזורים פתוחים שהוא יכל להגיע אליהם.

יתרונות:

* האסטרטגיה הכי טובה למקסום וניצול השטח הקרוב בו נמצא השחקן. כל אסטרטגיה אחרת הייתה מתרחקת מהאזור בו נמצא השחקן ובכך משאירה משבצות פתוחות (ערך 0) שלא היה בהם ביקור, כאשר הסיכוי לנצל אותם בעתיד קטן ככל שהמשחק מתקדם עקב הצעדים של השחקן והיריב.
* צריכת משאבים נמוכה – אסטרטגיה שזמן החישוב שלה הוא מיידי. בנוסף היא אינה צורכת זיכרון נוסף

חסרונות:

* אינה מתייחסת כלל למיקום היריב – היריב יכול לחשב אסטרטגיה שתחסום את השחקן כי הוא פשוט לא "רואה אותו" (עד למצבים בהם שני השחקנים נמצאים במרחק של 2 צעדים ופחות).
* הגנתית מדי – כיוון שהשחקן ממקסם את השטח הקרוב אליו, ולא מתרחק מהאזור שלו יותר מדי, מביא לחוסר שליטה בשטחים של הלוח. היריב יכול לחשוב על אסטרטגיה כזו שסוגרת לשחקן כמה שיותר שטחים ובעצם מביאה אותו למקסם ולצעוד בשטחים קטנים יותר.
* אחד המפתחות במשחק הוא לא רק למקסם שטחים קרובים, אלא באותו זמן שליטה על כמה שיותר אזורים בלוח שיאפשרו לשחקן כמה שיותר צעדים בעתיד וכמה שפחות צעדים אפשריים ליריב.
* אסטרטגיה חמדנית – במהלך חישוב הצעד הבא השחקן מתחשב רק בצעדים הקרובים אליו ולא מסתכל על הלוח במלואו, דבר שעלול לשחקן לבחור צד שאולי טוב לו בעתיד הקרוב אך גרוע בעתיד הרחוק.

5.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

כאשר השחקן הוא הכחול והיריב אדום.

במקרה הזה האסטרטגיה של השחקן תהיה אופטימלית. החולשה של האסטרטגיה נבעה בעיקר עקב חוסר התחשבות בצעדים של היריב. במצב המתואר אין ליריב כל השפעה על השחקן וגם לשחקן על היריב לכן הדרך היחידה לניצחון היא לעבור על כמה שיותר משבצות ולקוות שליריב יגמרו הצעדים מהר יותר. לכן האסטרטגיה הטובה ביותר כאן היא למקסם את השטח בו נמצא השחקן.

שאלות על הגדרת היוריסטיקה:

6.

החיסרון ביוריסטיקה הזו שהיא אינה מתייחסת למיקום של היריב ולמספר הצעדים האפשריים שלו.

דוגמה למצב בו היוריסטיקה הזו הייתה מביאה להפסד של השחקן הכחול:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

במצב זה לפי היוריסטיקה הנתונה יבחר לזוז ימינה ובכך יפסיד אם היריב יזוז למעלה ולמעלה (הכחול יתקע באותו אזור)

7. הרכיבים של היורסיטיקה יהיו:

א. מספר משבצות פנויות של הסוכן/ של היריב בהינתן צעד של הסוכן.

ב. "זמן משחק" שיוצג על ידי היחס בין מספר המשבצות הפנויות במצב מסוים (ההפרש בין מספר המשבצות הפנויות במצב התחלתי לבין מספר המשבצות התפוסות) לבין מספר המשבצות הפנויות במצב ההתחלתי

ג. סה"כ המשבצות הישיגות מהמיקום של השחקן

למעשה המשחק שלנו יהיה בהתחלה התקפתי (כאשר כמות המשבצות הפנויות הוא גבוה) .והמטרה תהיה לנסות ולהתקרב ליריב (להקטין את המרחק מנטן) ולנסות למנוע ממנו לבצע צעדים (לתת משקל גבוה יותר על חסימת היריב על ידי ספרית מספר המשבצות הפנויות שלו) , ובכל שהמשחק ממשיך אז מספר המשבצות הפנויות על הלוח מצמטמצם אז להפוך להיות יותר מגננתי ,כלומר לנסות לברוח מהיריב (להגדיל את המרחק מנטן) וגם לתת יותר משקל למספר המשבצות הפנויות של הסוכן משל היריב .

שאלות על Minimax בווריאצית AnytimeContract

8.

בווריאצית Anytime Contract של האלגוריתם Minimax מוחזר הפיתרון הכי טוב שניתן להחזיר תוך זמן קבוע שנקבע מראש לאלגוריתם. הרעיון מאחורי ווריאציה זו הוא שלרוב סוכנים מוגבלים במשאבים (בעיקר זמן) לפני שעליהם לפעול, וגם עקב מקדם סיעוף גבוה מדי במקרים רבים לא ניתן לחשב תחת המגבלות הללו באמת את המסלול האופטימלי לניצחון, לכן מנסים למצוא צעד מספיק טוב.

בMinimax שיטה מקובלת להתמודדות עם זמן היא העמקה הדרגתית. בכל איטרציה מחשבים עץ עם הגבלת עומק הולכת וגדלה בין איטרציה לאיטרציה. כאשר לאחר כל צעד נשמר הצעד הטוב ביותר שנבחר מהאיטרציה הקודמת. כאשר הזמן נגמר מוחזר הפיתרון הכי טוב שחושב עד כה.

הבעיה הנוגעת להעמקה הדרגתית המוצגת בהרצאה היא בעיית האיטרציה האחרונה.כאשר בכל איטרציה מגדילים את הגבלת העומק ב1 זמן החישוב של האיטרציה עבור עומק d גדל פי b מזמן החישוב האיטרציה עם הגבלת עומק של d-1 כאשר b הוא מקדם הסיעוף.כלומר זמני החישוב בין איטרציה לאיטרציה גדלים אקספוננציאלית ככל שמעמיקים בעיקר עבור מקדם סיעוף גבוה. מכיוון שאנו תמיד לא נספיק לחשב את האיטרציה האחרונה אנו נבזבז זמן חישוב יקר מאד שילך לפח.

9.

הפיתרון שמוצע בהרצאה הוא לשמור את ערך המינימקס של כל אחד מהבנים ברמה העליונה. באופן הזה אם נניח שבבמוצע האלגוריתם מפסיק באמצע האיטרציה האחרונה, אזי בממוצע נספיק לחשב עבור חצי מהבנים לעומק d וחצי מהבנים יחושבו לעומק d-1.

10. נבצע את הפישוט האלגברי הבא ,

יהי מספר העלים בעומק ברמה ה נניח כעת כי מקדם הסיעוף הוא 3 (נתעלם מ4 שקיים במצב ההתחלתי) , ונגדיר את להיות הזמן של האיטרציה האחרונה ואת להיות הזמן של האיטרציה הבאה לצורך הפיתוח עד העומק . נגדיר את להיות כמות העלים בעומק ה באיטרציה האחרונה ואת להיות מספר העלים באיטרציה הבאה (משום שמקדם הסיעוף הינו 3).

נגדיר את להיות הזמן שלוקח לפתח את העלים באיטרציה האחרונה ונגדיר את להיות זמן הפיתוח עד לעומק כולל באיטרציה הבאה

ולכן נקבל :

כך שמתקיים כי - הינו מהווה הזמן לפיתוח של עלה

ומכאן אם נחסום את זה נקבל סהכ-

ולכן החסם הינו

11. בוורציאת anytime contract , הסוכן alpha-beta צפוי להיות הרבה יותר טוב משום שהסוכן למעשה יחסוך בפיתוח של צמתים מיותרים (שאין צורך לפתח אותם) ולכן הוא יוכל להגיע לעומק רב יותר -לפי מה שראינו בהרצאה עומק שהוא כמעט פי 2 וכך לקבל פתרון הרבה יותר טוב.

12. לא ,התשובה לא תהייה זהה משום שבשני המקרים אנחנו חוסמים את העומק של של העץ וכך למעשה ההיתרון בלהגיע לעומק גדול יותר כבר לא קיים. ולכן למעשה גם המימנקס הבסיסי וגם alpha-beta יביואו את אותו הפתרון אף על פי השימוש בגיזום

*13.* היוריסטיקה תהיה מוגדרת מ :

* מספר צעדים אפשרים שיריב יכול לבצע בהינתן צעד מסוים, מספר הצעדים האפשריים שסוכן יכול לבצע בהינתן צעד מסוים.
* מהלך המשחק - כמות המשבצות שנותרו מתוך כלל המשבצות הפנויות שקיימות בלוח
* מרחק אווירי בין הסוכן ליריב

*15.*

*ניהול הזמן של הפונקציה make\_move נעשה באופן דומה לדוגמה שנתונה בסעיף 10.*

1. *אתחלנו*
2. *חישבנו את הצעד הכי טוב בעומק ושמרנו בצד.*
3. *שמרנו בצד את הזמן שלקח לנו לחישוב האחרון.*
4. *חישבנו את החסם העליון עבור זמן הריצה עבור עומק בדומה לסעיף 10.*
5. *אם הזמן הנותר קטן מהחסם העליון שחושב בשלב 4 בצענו וחזרנו לשלב 2. אחרת החזרנו את הצעד שנשמר עד כה.*