**תקציר**

- מאמר זה מציג מספר גישות עבור לפתור משחק בזמן אמת המורכב של ספינה כי חייב לבקר מספר נקודות ציון מפוזרות סביב מבוך דו מימדי מלא מכשולים. המשחק, סוכן נסיעות פיזית בעיה (PTSP), אשר הוצגו בשתי תחרויות כנס IEEE במהלך 2012, מספק איזון טוב בין לטווח ארוך תכנון (מציאת רצף אופטימלי של נקודות ציון לבקר), ותכנון לטווח קצר (נהיגה בספינה במבוך). הדף הזה מתמקד באלגוריתם שזכה בשני תחרויות PTSP: הוא מנצל את הפיסיקה של המשחק כדי לחשב את הסדר האופטימלי של נקודות ציון, והיא מעסיקה מונטה קרלו חיפוש עץ (MCTS) כדי לנהוג את הספינה. האלגוריתם משתמש חזרות של פעולות (מאקרו פעולות) כדי לצמצם את שטח החיפוש ניווט. וריאציות של אלגוריתם זה מוצגות ומנתח, על מנת להבין את כוחו של כל אחד שלה ומבחינת מה עושה גישה כזו הבקר הטוב ביותר שנמצא עד כה ל- PTSP.

**הקדמה**

בינה מלאכותית (AI) עבור משחקי וידאו בזמן אמת יש להיות תחום מחקר חשוב בשנים האחרונות, ובתוך בתחום יש מגוון רחב של אתגרים מעניינים- אמות מידה. המשחקים שנחקרו משתנים מאוד במורכבות לפעמים יש להתמודד עם כמה בעיות כדי להתקדם בתוך תקציבי זמן החלטה קטנים. במאמר זה, המוקד הוא על נסיעה פיזית בעיית מכירות (PTSP), וידאו של שחקן יחיד בזמן אמת שבו השחקן צריך להשלים את הפאזל לספק פעולות בקצב של כל אחד 40ms. משחק זה מוצג בשני תחרויות IEEE בינלאומיות במהלך 2012 [1], קבלת ערכים רבים שניסו לפתור את הבעיה המעסיקה מגוון של טכניקות שונות. מאמר זה מרוכז על היישום של מונטה קרלו חיפוש עץ (MCTS) [2] ל- PTSP, באמצעות מתכננים שונים ואת מנגנוני כדי לצמצם את שטח החיפוש כדי לשפר ביצועים. MCTS מבוסס על שני מושגים עיקריים: הראשון האחד הוא האיזון בין חקירה וניצול שטח החיפוש, באמצעות מנגנון בחירת עץ מבוסס על אמון העליון Bound עבור עצים (UCT) [3]. ה השני הוא השימוש במונטה קרלו (MC) סימולציות לחשב את ההטבות המשוערות ליישום פעולות שונות מדינות מסוימות. הבקר הסביר כאן הוא המנצח של שתי מהדורות של התחרות. מטרת המאמר היא לספק תובנה רכיבים שונים של הבקר, מנתח את עוצמותיו, חולשותיו וחשיבותו של כל אחת מהן אחד המרכיבים שלה. MCTS הוא משחק עץ משחק החיפוש שהראה ביצועים יוצאי דופן שבו כמה טכניקות אחרות, כגון כמו minimax או עומק החיפוש הראשון, נכשלו. דוגמה ברורה של זה הוא המשחק של Go, שבו MCTS הוא האלגוריתם הראשון מסוגל לספק רמה מקצועית של משחק בכמה גירסאות של המשחק [4]. מאז, MCTS הוחל על רחב מגוון של משחקים ויישומים שונים. בראון ואחרים. [5] בהווה, בסקר מלא, תיאור האלגוריתם, וריאציות ויישומים. בתחום של משחקים בזמן אמת, MCTS הוחל לבעיות שונות. לדוגמה, Samothrakis et al. [6] לעומת שונות וריאנטים MCTS במשחק של טרון, כולל heuristics שהניע את השחקן מלהופיע תנועות אובדניות. כמו כן, דן Teuling [7], לאחרונה נייר, מציע שחקן UCT המטפל מהלכים בו זמנית ומנבא תוצאות מבלי להשלים סימולציה מלאה. תחום נוסף שבו הוחל ב- MCTS בהצלחה היא גברת Pac-Man. במשחק הזה, השחקן שולט גב 'Pac-Man שמטרתה לאכול את כל הכדורים המפוזרים במבוך תוך הימנעות להיות שנתפסו על ידי רוחות. Ikehata et al. [8] להציע א גרסה של MCTS שמזהה מהלכים מסוכנים ואזורים במפה שבה צפוי להיתפס.

שחקן יחיד בזמן אמת משחקים, כגון PTSP, יש גם שימשו כאמות מידה עבור טכניקות MCTS. לדוגמה, Zhongjie et al. [9] שימוש ב- MCTS למשחק הידוע של טטריס. המחברים כללו מנגנון חדש ב- MCTS כי ענפים גזומים של העץ כדי להסיר את הפעולות האלה היו ליצור חורים במערך של בלוקים. השינוי הגדילו את הזמן הדרוש כדי להחליט על מהלך, אבל הפעולות נלקח תוצאות טובות יותר מאשר גרסה ללא שינוי של האלגוריתם. PTSP עצמו יש גם פנה באמצעות MCTS טכניקות. גרסה קודמת של הבעיה, שבה מפות לא היו מכשולים ולא גבולות, שימש פרז Et al. [10] כדי להצביע על חשיבות הידע ההוריסטי במדיניות חיפוש העץ ובסימולציות של מונטה קרלו. לבסוף, האלגוריתם המתואר במאמר זה זכה ל- PTSP התחרות פעמיים בשנת 2012, תוך שימוש MCTS. הראשון גרסה של הבקר המנצח מוסבר ב [11]. אחד מאת ההיבטים העיקריים של בקר זה הוא השימוש של מאקרו פעולות. כמו PTSP הוא שחקן בודד משחק דטרמיניסטי, זה הוא נוח מאוד לקבוצה פעולות יחיד לתוך מוגדר מראש רצפים כדי לצמצם את גודל שטח החיפוש.

לדוגמה, פעולה מאקרו תהיה חזרה על פעולה אחת a עבור שלבי זמן T. הפשטות אלה יכולים לעזור להפחית את גורם הסתעפות סימולציות MC של האלגוריתם יכול להסתכל קדימה קדימה עם דיוק מושלם. הרעיון של macroactions (או פעולות המורחבת זמנית) יש היסטוריה ארוכה בינה מלאכותית (ראה [12]) ושהשתמש בה באופן נרחב בתחומים שבהם גודל של שטח המדינה היה עושה את עלות החיפוש אוסרני עם חישובית הנוכחי משאבים (למשל, [13])

באלה ופרן [14] יישמו את UCT לאסטרטגיית Real Time משחק וורגוס. המשחק הזה מבוסס על תכנון תקיפה טקטי, שבו יחידות מרובות במקביל ביצוע פעולות ב את המשחק ואת פעולות שונות, להורג בו זמנית, יכול לקחת משך משתנה. המחברים discretize זה רציף המדינה בהפשטה כי מפשט את המאפיינים של פריסת המשחק הנוכחית: מעברים בין מדינות מנוהלים על ידי פעולות המשפיעות על קבוצת יחידות, להקטין את הגודל של חיפוש עץ ומאפשר סימולציות יעיל יותר. מרחבי פעולה מופשטים שימשו, למשל, על ידי Childs et al. [15] (ולאחרונה חזרה על ידי ואן אייק et אל. [16]), אשר החיל קבוצות משולבות של פעולות בעת השימושUCT במשחק עץ מלאכותי P-Game. עץ P-Game הוא עץ minax שבו המנצח נקבע על ידי ספירה מנגנון על המצב הסופי של הלוח. במחקרם, מחברים קבוצה מספר פעולות דומות בצעד אחד כדי לצמצם את גורם הסתעפות, קבלת תוצאות מבטיחות. פאולי ואחרים. [17] השתמשו בטכניקה דומה בעת יישוםMCTS למשחק הקלפים של דאו די ג'ו, שם מהלך אחד הבחירה מחולקת לפעמים לשניים נפרדים ורציפים החלטות כדי להפחית גורם הסתעפות על חשבון העץ עוֹמֶק.

שאר מאמר זה מאורגן כדלקמן: חלק II מתאר את המשחק PTSP ואת סעיף III מסביר הכללי ארכיטקטורת בקר ה- PTSP. סעיף IV מציג ולאחר מכן את שיטות החיפוש עץ המועסקים במחקר זה, ו סעיפים V, VI ו- VII מתארים את המרכיבים השונים הפרמטרים של הבקר: פעולות מאקרו, פונקציות ערך ו TSP פותר, בהתאמה. הגדרת הניסוי היא המתואר בסעיף VIII והתוצאות והניתוח הם המובא בסעיף IX. לבסוף, את המסקנות ואת העתיד העבודה מוצגת בסעיף X.

**בעיה טכנית (חלק 2)**

המשחק

בעיית סוכן נסיעות פיזית (PTSP) הוא הסתגלות של סוכן המכירות הנוסע (TSP) כדי להמיר זה לתוך שחקן יחיד בזמן אמת משחק. ה- TSP הוא באר ידוע אופטימיזציה מצב קומבינטורית שבו סדרה של ערים (או צמתים בגרף) ועלויות הנסיעה בין הם ידועים. איש מכירות חייב לבקר בכל הערים האלה בדיוק פעם אחת ולחזור לעיר המוצא בעקבות הנתיב בעלות מינימלית. ב PTSP, השחקן (כלומר מוכר) קובע חללית כי חייב לבקר סדרה של נקודות ציון מפוזרים מסביב מבוך מהר ככל האפשר. PTSP הוא בזמן אמת משחק, רומז כי פעולה חייב להיות מסופק כל 40 אלפיות השנייה. הפעולות הזמינות מסוכמות באיור 1. פעולות אלה מורכבות משני תשומות שונות המיושמות בו זמנית: תאוצה והיגוי. האצה יכולה לקחת שני ערכים אפשריים (לסירוגין), בעוד היגוי יכול לפנות את הספינה שמאלה, ימינה או לשמור את זה ישר. זה מוביל לסך של שש פעולות שונות. במקרה שהבקרה נכשלת להחזיר פעולה לאחר מגבלת הזמן, פעולה NOP (ID: 0) הוא מיושם, אשר מבצעת שום הגה ולא האצה. (תמונה 1 עמוד 2) מצב הספינה נשמר משלב משחק אחד הבא, שלושה ווקטורים שונים משונים לאחר החלת פעולה אחת. הכיוון של הספינה משתנה כפי שמוצג ב משוואה 1, בהתחשב בכיוון של הספינה בשלב האחרון dt ואת זווית זווית α (זווית קבועה שיכולה להיות חיובית או שלילי, בהתאם לתחושה של קלט ההיגוי, או 0 אם הספינה הוא אמר ללכת ישר). משוואה 2 מציינת כיצד ווקטור המהירות משתנה, בהתחשב במהירות הקודמת Vt, האוריינטציה החדשה, קבוע האצה (K) ו- a מקדם הפסד החיכוך (L). במקרה זה, קלט האצה קובע את הערך של Tt: להיות 1 אם הפעולה מרמזת דחף או 0 אחרת. לבסוף, משוואה 3 מעדכנת את המיקום של על ידי הוספת וקטור המהירות למיקומה הקודם שלב המשחק (pt). מודל הפיזיקה הזה שומר על האינרציה של הספינה בין מחזורי המשחק, מה שהופך את המשימה של ניווט הספינה יותר מאתגר. (נוסחאות עמוד 2).

מכשולי המבוך אינם פוגעים באונייה, אלא להכות אותם מייצרת התנגשות אלסטית, אשר משנה את מהירות הספינה (הן בכיוון והן בסדר גודל). An דוגמה של מפה המופץ עם המשחק מוצג ב איור 2. בנוסף החזרת פעולה כל 40 אלפיות השנייה, הבקר שתוכנן עבור PTSP חייב לכבד שני אחרים מגבלות זמן: זמן אתחול ומגבלה להגיע לשלב הבא נקודת ציון במשחק. שני הזמנים תלויים במספר נקודות ציון, אשר בניסויים המתוארים זה נייר הם 30, 40 ו 50. זמן האתחול, זה יכול להיות המשמש את הבקר לתכנן את המסלול לעקוב לפני המשחק מתחיל, מוגדר 1 שנייה לכל 10 נקודות ציון. הזמן להגיע לנקודת ציון הבאה תלוי בצפיפות של נקודות ציון במבוך: מפות עם 30 נקודות ציון לאפשר 700 צעדים, 40 נקודות ציון לרמוז 550 המשחק קרציות היתר 50 את השימוש של 400 מדרגות משחק. מונים אלה נקבעים על זה להגביל בכל פעם נקודת ציון הוא ביקר, והם ירדו ב 1 בכל מחזור. (תמונה 2 עמוד 3)

אם הערך מגיע 0, המשחק נגמר. שם הוא גם מונה שני שסופר את המספר הכולל של T שלבים של המשחק. הציון הסופי של המשחק מוגדר על ידי tuple (W, T) כאשר W הוא מספר נקודות הביקור ו - T הוא סך הכל בשימוש. בין שני משחקים, זה מקבל יותר נקודות ציון נחשב הטוב ביותר. במקרה של תיקו, אחת שסיימה את המשחק במדרגות זמן פחות מנצחת. בכל שלב משחק, הבקר מסופק עם עותק של מצב המשחק הנוכחי ואת נקודת הזמן בשל המהלך. מצב המשחק מכיל מידע על הספינה עצמה (מיקום, מהירות, כיוון), המשחק (הזמן נותר, מספר נקודות ציון), המפה (מיקום המכשולים, מציאת שאילתות) ואת נקודות ציון (עמדות והאם הם ביקרו או לא). עם זאת, התכונה החשובה ביותר של מצב משחק זה הוא מספק מודל קדימה לרוץ סימולציות. במילים אחרות, ניתן לבדוק מה מצבי העתיד של המשחק יהיה אם רצף מסוים של נעשו מהלכים. PTSP הוא דטרמיניסטי ו singleplayer המשחק ואת הסימולציות שבוצעו הם לחלוטין מְדוּיָק. זה עושה חיפוש עץ גישות סימולציה במיוחד מתאים במיוחד עבור סוג זה של בעיה.

. PTSP vs TSP

גישה סבירה לפתרון PTSP היא לחלק את הבעיה לתוך שני המרכיבים שלה: סדר נקודות ציון ל לעקוב ולנווט את הספינה לכל אחד מהם. למרות זאת, שתי הבעיות הללו הן תלויות זו בזו, כהתעלמות מודל הפיזיקה של המשחק עלול להוביל לאופטימלי סדר נקודות ציון. בפרט, אם את הסדר של נקודות ציון מתקבל רק את המרחקים בין נקודות הציון (מחושב, למשל, באמצעות אלגוריתם A \*) התוצאה הסופית עשויה להיות לא אופטימלית. ראוי לציין את זה המטרה הסופית היא למזער את הזמן שנדרש כדי לבקר את כל נקודות ציון, ולא את המרחק. לפיכך, נתיבים שלא נראה אופטימלי בגלל אורך שלהם יכול לספק טוב יותר פתרונות אם הספינה מסוגלת להסיע אותם במהירות. באופן כללי, זה הוצע [1] כי מסלולים למזער את השינויים של הכיוון ואת נקודות ציון לכסות ישר (או כמעט ישר) שורות טובים יותר עבור PTSP. איור 3 (מתוך [1]) מציג תמונה ברורה דוגמה למצבים אלה ב- PTSP. (תמונה 3 עמוד 4).

תחרות PTSP

ה- PTSP הוצג בשתי תחרויות בשנת 2012, שהתקיימו ב שני כנסים בינלאומיים: הקונגרס העולמי IEEE על אינטליגנציה חישובית (WCCI) ועידת IEEE על אינטליגנציה חישובית ומשחקים (CIG). בשתי התחרויות, המשתתפים יכלו ליצור בקרי ב- Java על ידי הורדת ערכת Starter המכילה כמה בקרי מדגם ו 10 מפות שונות. שרת, נגיש בכתובת www.ptsp-game.net, רץ ברציפות, קבלת הגשות וביצוע הבקרות בהן 10 מפות וקבוצות אחרות של מפות, שנוצרו כדי לספק מבחן מנגנון עבור המשתתפים במכונה שבה הסופי ההערכה תבוצע. לאחר המועד האחרון של התחרות, בקרי שיחק סדרה של 20 מפות חדשות שמעולם לא נראו או שיחקו על ידי המשתתפים לפני. כל בקר בוצע 5 פעמים על כל מפה. התוצאה הסופית היתה הממוצע של את שלושת המשחקים הטובים ביותר, שמציעות נקודות ציון וזמן נלקח. השחקנים, מדורגים לפי מספר נקודות ציון ביקר ואת הזמן נלקח, הוענקו בעקבות פורמולה אחד בסגנון (מתואר בסעיף IX-C). המנצח של התחרות היא השחקן ששלח את הבקר השיגו את רוב הנקודות בכל המפות. בתחרות WCCI [18], רק מפות עם 10 נקודות ציון נשקלו. עם זאת, במהדורת CIG של התחרות, המפות השתנו כדי להכיל 30, 40 ו -50 נקודות ציון. במחקר זה, המפות המשמשות את הניסויים הם אלה המועסקים לדירוג הסופי של CIG תַחֲרוּת. שיטת הזכייה של שתי תחרויות בוט היתה הכניסה על ידי פאולי ואח ', שתיארו את בקר ה- WCCI שלהם [11]. בקר זה, המבוסס על MCTS, פעולות מאקרו ו physicbased TSP פותר, הוא הבסיס עבור ניסויים המתואר הדף הזה.

**בקר ה- PTSP (חלק 3)**

הסעיפים הבאים מתארים את השונות המרכיבים של בקר PTSP, כגון פותרי ה- TSP, שיטות חיפוש ו discretization מאקרו פעולה. איור 4 מראה את הארכיטקטורה הכללית של הבקר, שהוא משותף לכל התצורות שנבדקו במחקר זה. (תמונה 4 עמוד 3).

במהלך שלב האתחול, מתכנן כביש TSP (או TSP Solver) מקבל את הסדר של נקודות ציון לעקוב במהלך המשחק. לאחר מכן, המשחק מבקש פעולה של PTSP הבקר (המוזכר גם כאן כנהג) בכל משחק צעד, לספק מידע על מצב המשחק, עד המשחק הסתיים. בכל אחד מהמחזורים הללו, הבקר מדמה פעולות במשחק, מחליטה מה יהיה הבא פעולה לביצוע. כפי שיוסבר בהמשך סעיף V, מאקרו פעולות משמשים כדי discretize את שטח החיפוש. בזמן הבקר הוא ביצוע פעולה מאקרו נתון בזמן t, סימולציות מבוצעות על מנת לברר מי הוא הטוב ביותר מאקרו פעולה לבצע בזמן t + T (כאשר T הוא אורך של כל מאקרו פעולות).

**שיטות חיפוש עבור בקר ה- PTSP (חלק 4)**

שלוש שיטות חיפוש שונות הועסקו בכך מחקר כדי להתמודד עם PTSP: עומק חיפוש ראשון, מונטה קרלו סימולציות מונטה קרלו עץ חיפוש. כל השיטות האלה לחלוק את אותו זמן אילוץ: החלטה החלטה חייבת להיעשות בתוך זמן נתון (40ms). תכונה זו קובעת את הדרך האלגוריתמים מתוכננים, כהתאמה לתקציב זמן זה עושה את זה בלתי אפשרי להגיע לסוף המשחק במרחב העצום ברוב המקרים. בנוסף, PTSP הוא משחק פתוח: זה לא אפשר למצוא מדינה במרחב החיפוש שבו כל נקודות ציון הם ביקרו רק על ידי החלת פעולה אקראית באופן אחיד בְּחִירָה. ואכן, אפילו להגיע לנקודת הביקורת הבאה היא לבקר לא ניתן להשיג בדרך כלל באמצעות סימולציות אקראיות אחידות. זה ולכן חיוני להוסיף פונקציה ערך היוריסטי מסוגל של הערכת איכותן של המדינות השונות המצוינות בסעיף (לתיאור אלה ראה סעיף VI). הסעיף הזה מפרט את נהלי החיפוש המועסקים במחקר זה.

חיפוש עומק ראשון

עומק חיפוש ראשון (DFS) הוא גרף מסורתי או עץ אלגוריתם החיפוש. הוא מחפש באופן ממצה בכל צומת של העץ, מרחיב באופן איטי את הילד הבא מחפש עמוק עד הצומת לא להרחבה הוא הגיע. החיפוש מנווט את העץ כדי למצוא את הצומת הבלתי צפוי הבא וממשיך תהליך זה, עד כל הצמתים מתחת לשורש יש ביקרתי. החל מהשורש, המייצג את המצב הנוכחי של המשחק מוחלים פעולות המובילות למצבי משחק עתידיים, תוך שימוש במודל הקדמי הנזכר בסעיף II. העתיד הזה קובע, נגזר לאחר יישום פעולה, נלקחים כילדים של צומת האב. לבסוף, הפעולה שהובילה את הציון הטוב ביותר להשיג נלקח כמו מהלך לשחק במשחק. על מנת להגשים את מגבלות התקציב בזמן, מקסימום עומק עבור החיפוש צריך להיות החליט. במקרה זה, זה נקבע באופן אמפירי כי כאשר עומק 4 כבר להגיע, DFS לא יכול להרחיב עוד יותר מבלי לחרוג - גבול הזמן המוטל. לכן, כאשר הצומת בעומק זה הוא להגיע, הערכה מבוצעת על מנת להקצות ציון על פי ממצאי המדינה בסעיף VI. בשל הגבלה זו עומק, מספר קובע כי מוערכים עבור כל החלטה היא 4^6 (6 בחזקת 4) = 1296.

סימולציות מונטה קרלו ו UCB1

מונטה קרלו (MC) שיטות מבוססות על סימולציות אקראיות כי מדגם את שטח החיפוש אחיד או מודרך על ידי Heuristics. במקרה של PTSP, סימולציות מבוצעות מן המצב הנוכחי של העץ (או שורש), בחירת רצף של פעולות באופן אחיד באקראי עד עומק מקסימלי כבר הגיע. מצב המשחק מוערך לאחר מכן. ערכים שונים עבור עומק הסימולציה נוסו את הניסויים המתוארים במאמר זה. עומק סימולציה ואת התקציב 40ms זמן קובע את המגבלה של איך סימולציות רבות ניתן לבצע. אם פעולות נבחרות באופן אחיד באקראי, זה צפוי כי כל פעולה נבחרה אחת השישית של הזמן, בייצור חיפוש מבוזר באופן שווה. גישה זו, לעומת זאת, לא ליהנות מהעובדה כי אחד או יותר פעולות יכול להיות טוב יותר אחרים, כפי שהוצע על ידי ציונים שהושגו על ידי הערכות בסוף הסימולציות. כדי להתגבר על בעיה זו, מבחר חמוץ bandit המדיניות מועסקת. רב חמוש השודדים הוא בעיה מתורת ההסתברות שבה כל אחד מכונות המזל מרובים מייצרת תגמולים רצופים rt: r1, r2,. . . (לזמן T = 1, 2,. . . ) שמקורו בחלוקה הסתברותית לא ידועה. המטרה היא למזער את החרטה, כלומר את ההפסדים לא לבחור את הזרוע האופטימלית. פוליטיקאים חמושים בחר פעולות בבעיה זו על ידי איזון החיפושים של פעולות זמינות (משיכת הזרועות) וניצול אלה המספקים תמורה טובה יותר (אופטימיזציה מול חוסר ודאות).

Auer et al. [19] הציע את בטחון אמון העליון (UCB1) מדיניות בחירת שודד: המטרה היא למצוא פעולה שממקסמת את הערך נתון על ידי משוואת UCB1. כאן, Q (s, a) מייצג את ממוצע אמפירי של בחירת פעולה ב. זה מחושב כמו הפרס המצטבר של בחירת פעולה מחולק על ידי כמה פעמים פעולה זו נבחרה. Q (s, a) הוא המונח ניצול, ואילו המונח השני (משוקלל על ידי C קבוע) הוא מונח החיפושים. שים לב, אם האיזון משקל C = 0, UCB1 עוקב בתאוותנות פעולה שיש לו סיפק את התוצאה הממוצעת הטובה ביותר עד כה. תקופת החיפושים מתייחס למספר הפעמים שבהן כל פעולה נבחרה מן המדינה נתון נתון (N (ים, א)) ואת כמות הבחירות נלקח ממצב הנוכחי (N (s)). כאשר פעולה א נבחרה, הערכים של N (s, a) ו- N (s) עלייה. ההשפעה של זה היא כי המונח חיפוש אחר פעולות שונות מאשר א גידול, המאפשר חקירה מגוונת יותר של שונות פעולות זמינות באיטרציות עתידיות של האלגוריתם. שווי יתרות C בין שני מונחים אלה. אם Rewards Q (s, a) מנורמל בטווח [0, 1], נפוץ הערך המשוער עבור C במשחקים בודדים הוא √ 2. זה גם חשוב לשים לב לכך, כאשר N (s, a) 0 = עבור כל פעולה, יש לבחור בפעולה זו. במילים אחרות, זה לא אפשרי כדי ליישם את UCB1 אם לא נבחרו כל הפעולות ממדינה לפחות פעם אחת. כאשר זמן התקציב נגמר, הפעולה לבצע ב המשחק הוא זה שנבחר יותר פעמים על ידי UCB1. (נוסחה עמוד 5)

עץ חיפוש מונטה קרלו

מונטה קרלו עץ חיפוש (MCTS) היא טכניקה לחיפוש עץ זה יוצר עץ על ידי ביצוע סימולציות מונטה קרלו. ה הגרסה הידועה ביותר של MCTS הוא אמון העליון Bound עבור עצים (UCT), המוצעים על ידי Kocsis ו Szepesvari [3], [20]. ' UCT משתמש UCB1 (כמתואר בסעיף IV-B) לבנות פוטנציאל עץ אסימטרי כי הוא גדל לעבר יותר חלקים מבטיחים של מרחב החיפוש. בכל פעם ש- UCT בוחרת פעולה שמובילה למדינה שאינה מיוצגת ב עץ, צומת חדשה מתווסף לו סימולציה מונטה קרלו הוא התחיל משם. אלגוריתם MCTS ניתן לראות תהליך של ארבעה שלבים זה חוזר על עצמו כמה חזרות. במהלך הצעד הראשון (בחירה), מדיניות בחירת העץ (למשל UCB1 עבור UCT) בוחר את הפעולה הבאה שתבצע. הפעולה מובילה צומת בעֵץ. אם הצומת כבר קיים בעץ ולאחר מכן את הבחירה המדיניות מוחלת שוב (וכך הלאה רקורסיבית); אם לא, זה הוסיף כמו הצומת עלה חדש (צעד הרחבה). סימולציה MC הוא יזם מצומת זה עד עומק שנקבע מראש או סוף המשחק. המדינה הגיעה בסוף הסימולציה מוערכת (שלב סימולציה). סימולציה זו MC הוא מונע על ידי מדיניות ברירת מחדל, כי יכול להיות באופן אקראי באופן אחיד (כפי שהוא במחקר זה) או בהנחיית האוריסטי.

השלב האחרון באיטרציה הוא Backpropagation. במהלך זה שלב, את הפרס המתקבל על ידי פונקציית הערכה הוא backpropagated לאורך הצמתים ביקר עד השורש. כל ה הצמתים חצו את הנתונים הסטטיסטיים הפנימיים שלהם: N (s), N (s, a) ו- Q (s, a), איסוף ידע עבור שלב בחירת העץ באיטרציה הבאה של האלגוריתם. כמו במקרים הקודמים, את עומק סימולציה MC חייב להיות מוגבל, אם לא, זה לא יהיה אפשרי להגיע סוף המשחק בתוך התקציב הזמן מותר. אלה ערכי עומק מוגדרים בהגדרת הניסוי, אבל זה חשוב לשים לב כי עומק הסימולציה מתאים מספר הפעולות שנלקחו משורש העץ, ולא מהצומת המורחבת החדשה בכל איטרציה. זה גמור על מנת לספק את אותו מבט קדימה מילדים של השורש למרות צמיחת העץ האסימטריה.

**פעולות MACRO ב PTSP (חלק 5)**

האלגוריתמים המוצגים במאמר זה מתייחסים ל- PTSP כשניים (אבל קשורים) בעיות משנה: ניווט הספינה לתכנן את סדר נקודות הציון כדי להיות אחריו. הסעיף הזה מתאר את הגישה המשמש לנהוג, או לנווט, את הספינה יחד התוואי שנבחר על ידי השיטות המתוארות בסעיף VII, דהיינו איך ללכת בשבילים מנקודת ציון אחת לאחרת. (בשורה עם ספרות משחקים, "היגוי" כאן כולל האצה כמו גם סיבוב של הספינה.) קלט לספינה נשלחת פעם אחת כל 40ms, כדי לנווט את הספינה סביב מכשולים לעבר נקודות ציון. בעיית ההיגוי עבור PTSP יכולה להיחשב כ בעיית החלטה רציפה על עץ. זה יכול גם להיחשב כבעיה בגרף מכוון, כלומר לאפשר נתיבים מרובים לאותו הצומת ואולי מחזורי, אבל בשביל הפשטות גרף הוא הניח להיות עץ. הצמתים של העץ הם מדינות ואת הקצוות הם פעולות. פעולה מאקרו - פעולה מוגדרת כ רצף הפעולות M = ha1,. . . , אני. ביצוע מקרואקציה מתאים לנגן את רצף הפעולות הכלול בתוך מאקרו פעולה. עץ החלטה של ​​macroactions ניתן לבנות: את הצומת קבוצה של עץ זה הוא משנה של הצומת של עץ ההחלטה המקורי.

ב PTSP, את אוסף של פעולות משפטיות ממדינה (כלומר קבוצה של הקצוות מתוך הצומת המקביל) זהה עבור כל המדינות. אם זה לא היה המקרה, יהיה צורך יותר טיפול בהגדרת פעולות מאקרו: אם יש ליישם את המאקרו – פעולה ב s0 המדינה, ו si היא המדינה שהושגה בסופו של דבר על ידי החלת כל הפעולות, אז ai + 1 חייב להיות פעולה משפטית מ si עבור PTSP, מטרת פעולות מאקרו היא להפחית את הגודל של הבעיה ולהגדיל את היכולת של חיפוש עץ שיטות לביצוע תכנון קדימה. זה יכול להיות מושגת על ידי הפחתת הפירוט של נתיבים אפשריים ומניעת הספינה מ ביצוע קטן (לפעמים חסר משמעות) התאמות כדי מהירות וכיוון. מאקרו פעולות בשימוש במאמר זה הם ללא ספק הפשוטה ביותר האפשרית, המורכבת של ביצוע אחד של שש פעולות זמינות (ראה איור 1), עבור מספר קבוע של צעדים זמן T. פעולות מאקרו מורכבות יותר נבדקו, כגון סיבוב אחד מזוויות מסוימות בעת דחיפה או לא דחף. בעיה אחת שהתעוררה היתה כי שיטות החיפוש לא לבצע גם כאשר פעולות שונות לקח אורכים שונים של זמן לביצוע: מאז ההערכות בשימוש (סעיף VI) הם פונקציות מרומזות של המרחק, שיטת החיפוש אופטימיזציה אלה פונקציות ערך נוטה פשוט לטובת macroactions עוד קצרים יותר.

לאחר כל עומק של העץ המתאים באותו רגע בזמן שיש את חיפוש רול החוצה לעומק קבוע פירושו במקום החיפוש השיטות יממשו את אורך הנתיב (על ידי הגדלת המרחק נסע בפרק זמן קבוע). על המפה של 50 נקודות ציון, המשחק תמיד לוקח פחות מ 20000 פעמים זמן (זמן רב יותר מרמז כי הזמן הגבול בין נקודות ציון עלה לפחות פעם אחת). לכן עץ ההחלטה יש עד 6 20000 צמתים. המשחק לוקח פחות מ 20000 T מאקרו פעולות, ולכן גודל של מאקרו פעולה עץ מוקף מעל 6 20000 T, המייצג מאות של צמצום בסדר גודל של גודל הבעיה כדי להיפתר (כאשר T ≥ 2). כדי להמחיש זאת, תן לנו לעשות הערכה שמרנית של 2000 צעדים הזמן הממוצע אורך המשחק, ולהגדיר T = 15. עץ המשחק מכיל 6 2000 ≈ 101556 קובע, ואילו עץ מאקרו פעולה מכיל 6 2000 15 ≈ 10103 מדינות. גודלו של עץ המאקרו בפעולה זו דוגמא דומה לגודל עץ המשחק למכלול מורכב לוח המשחק: להשוואה, מספר מדינות 9 × 9 עבור הוא חסום מעל 81! ≈ 10120. עץ המאקרו-פעולה של סדר 101453 פעמים קטן יותר מאשר את המשחק עץ מלא

הפרמטר T שולט במסחר בין הפירוט של נתיבים אפשריים ושל פוטנציאל התכנון קדימה חיפוש עץ. מאז צעד סיבוב אחד מתאים סיבוב מתוך 3 מעלות, הגדרה של T = 30 מגבילה את הספינה רק עושה 90◦ מסתובב. (שים לב כי הספינה עשויה להיות מהירות ראשונית, והוא עלול לדחוף תוך כדי סיבוב, ולכן ההגבלה ל 90 מעלות מסתובב אינו מגביל את הנתיב ל -90 מעלות זוויות). בעת שימוש זה הגדרה, אלגוריתמים החיפוש ימצא נתיבים כי יש להקפיץ את הקירות או לעקוב אחר מסלולים מפותלים כדי לעלות עם נקודות ציון. בחירה של T = 10 מקבילה ל 30 פונה, המאפשר שליטה עדינה יותר על הספינה ונתיבים חלקים יותר. ההבדל מתוארת באיור 5 שבו השביל עם 90◦ פונה הוא יותר משונה (ולוקח זמן רב יותר) מאשר עם זה 30◦ מסתובב. מצד שני, כאשר T = 10 להגיע לעומק D בעץ החיפוש מתאים לנקודה בזמן מוקדם יותר עם T = 30. זה משפיע על יכולתו של בקר ההיגוי לתכנן מראש. (תמונה 5 עמוד 6)

ראוי להזכיר כי גישה שונה יכולה להיות גם אפשרית: במקום לבצע את כל הפעולות יחיד מן הפעולה המקרו הטובה ביותר שנמצאה, אפשר היה ליישם רק את ראשון. מאקרו פעולות מועסקים במהלך סימולציות MC, אבל כל המסלולים האפשריים עבור הנהג כמו macroaction הבא הטוב ביותר מחושבת כל 40ms. גישה זו, עם זאת, מאפשר פחות זמן להחליט על הצעד הבא, כמשחק אחד בלבד צעד ניתן להשתמש כדי לבחור את הפעולה הבאה לקחת.

**פונקציות VALUE (חלק 6)**

מדינת המטרה ב- PTSP (מצב שבו כל נקודות הציון נאספו) יכול להיות כמה אלפי צעדים זמן, או כמה מאות פעולות מאקרו, אל העתיד. מציאת כזה מדינה אך ורק על ידי חיפוש עץ הוא בלתי נסבל. במקום החיפוש מבוצעת לעומק קבוע, תוך שימוש בהערכה היוריסטית הפונקציה למצב nonterminal וכתוצאה מכך חיפוש עץ כדי לייעל את הערך ההוריסטי הזה. פונקציות הערך מבוססות סביב "הנתיב הקצר ביותר" המרחק אל נקודת ציון היעד, לוקח מכשולים בחשבון. סעיף VI-A מתאר את מבנה הנתונים המחושבים שאנו משתמשים בהם כדי לחשב את המרחקים האלה, ואת סעיפים VI-B ו- VI-C לתת שני הפונקציות ערך נבדק במאמר זה.

מחשוב מרחק מפות

לתכנון המסלול, יש צורך לאמוד את הנסיעה הזמן בין כל זוגות של נקודות ציון, לוקח מכשולים הִתחַשְׁבוּת. עבור פונקציית הערכה בשימוש על ידי נהיגה אלגוריתמים, יש צורך למצוא את המרחק בין ספינה ואת נקודת ציון הבאה, שוב חשבונאות עבור מכשולים. המרחקים בין כל נקודת ציון זה לזה ללא מכשול נקודה על המפה מחושבים מראש. זֶה ניתן לעשות זאת במהירות וביעילות באמצעות סריקה למילוי סריקה אלגוריתם: מפות עבור PTSP מיוצגים כ 2 מימדי מערכים, כאשר כל תא הוא או מכשול או שטח פתוח. זה ייצוג דמוי מפת סיביות הוא נוח במיוחד אלגוריתמים מגרפיקה ממוחשבת. מפות המרחק מחושבות באמצעות אלגוריתם מילוי סריקה שונה מילוי [21]; לִרְאוֹת אלגוריתם 1. המערך הדו מימדי המתקבל נקרא מפת מרחק. לאחר חישוב מפות המרחק האלה, מציאת המרחק בין נקודת ציון לנקודה אחרת (נקודת ציון או ספינה) מיקום) ניתן להסתכל במהירות בזמן O (1). האלגוריתם ממקם כל כניסה במערך + ∞, מלבד הערך המתאים עמדת נקודת ציון i אשר מאותחל ל 0. האלגוריתם שומר על תור תאים שממנו לסרוק, החל עמדת נקודת ציון. מכל תא, האלגוריתם סורק שמאלה וימינה. עבור כל תא שנסרק (x, y), מרחק dt טנטטיבי (x, y) הוא כפי שחושב (נוסחה עמוד 6)

שם(x ', y') טווחי מעל השכן אורתוגונליים ואלכסון תאים של (x, y). התא נחשב למילוי אם dt (x, y) < Di [X, y], כלומר, אם המרחק המהוסס קטן מהמרחק מאוחסן כעת ב- Di . במהלך הסריקה, תאים מילוי הם מעודכן על ידי הגדרת Di [X, y] = dt (x, y); הסריקה מסתיימת כאשר נתקלים בתא בלתי ניתנת למימוש. בזמן הסריקה, אלגוריתם בודק את התאים מיד מעל ומתחת השורה הנוכחית: עם המעבר מאזור של תאים בלתי ניתנים למימוש לאזור של תאים מילוי, התא מילוי הראשון הוא enqueued. דוגמה לכך מוצגת באיור 6. (אלגוריתם+2 תמונות עמוד 7)

תאי מכשול נחשבים תמיד בלתי ניתנים למימוש. כל מכשול הוא מוקף גם עם אזור בלתי מסודר + שהרדיוס שלו שווה לרדיוס האונייה. זה יעיל כלומר, את מפת המרחק, ואת האלגוריתמים שמשתמשים זה, להתעלם כל רווחים או מסדרונות כי הם צרים מדי להתאים את הספינה. A + מעוצב ולא אזור עגול משמש אך ורק עבור מהירות חישובית: למטרות הסרת המסדרונות הצרים, הן יעילות באותה מידה. אם תא הוא מילוי תלוי המרחק הטנטטיבי, אשר בתורו תלוי התוכן של מפת המרחק סביב התא, תא שכבר מילא בעבר נחשב unillable יכול להיות מילוי שוב מאוחר יותר. זֶה תואמת נתיב חדש שנמצא לתא, קצר מ הנתיב הקצר ביותר הידוע בעבר. זאת בניגוד ל קלאסי מילוי אלגוריתם [21], שם מלא בעבר פיקסל הוא לא מילא שוב. איור 7 מציג דוגמה למפת מרחק. ציין זאת שכן המרחקים מחושבים על בסיס אורתוגונלית ואלכסון תאים סמוכים, קווי המתאר הם מתומן ולא עָגוֹל. קווי המתאר המעגליים יכולים להיות קרובים יותר על ידי הגדלת השכונה השכנות, אבל זה יישום הוא עסקה טובה בין מהירות ודיוק. המפות מרחק לא צריך להיות מדויק לחלוטין, רק מדויק מספיק כדי לשמש כמדריך לתכנון המסלול הימנעות מכשולים.

מיופייק מעריך (??)

בכל שלב נתון במהלך המשחק, המתכנן התוואי (סעיף VII) מזהה את נקודת היעד הנוכחית. בואו sr מציינים את השבר של המרחק שנסע לכיוון היעד נקודת ציון, scaled כך sr = 0 כאשר הספינה נמצאת נקודת ציון שנאספה בעבר (או בתחילת מיקום אם לא נקודות ציון טרם נאספו) ו sr = 1 כאשר הספינה הגיע לנקודת היעד. כאן "המרחק" הוא הקצר ביותר מרחוק, הרים את מבטו במפת המרחק של היעד (סעיף VI-A). ההערכה הקצרה של מצב המדינה מחושבת כדלקמן: (נוסחה עמוד 8 למעלה) כלומר, ערך המדינה הוא יחסי למרחק עד לנקודת היעד של היעד, עם כל הדואר שלאחר קובע שיש אותו ערך αw. מיטוב המרחק בדרך זו, במקום נוקשה בעקבות ירידה ירידה נתיבים המוצעים על ידי שלב התכנון המסלול, מעודד הבקר לקחת קיצורי דרך לנקודת ההתייחסות הבאה היתרון של הפיזיקה של המשחק. חשוב כי αw> 1, כלומר מדינות שלאחר קולקציה יש ערך גבוה יותר מאשר כל מדינות מראש אוסף, כמו זה תמריצים החיפוש לקראת אוסף נקודות ציון. בלי זה, ההערכה קרובה מאוד למיקומים בשכונה של נקודת ציון, אשר לעתים קרובות התוצאות הספינה מחכה ליד או להסתחרר סביב זה (בדרך כלל מתקרב עם כל סיבוב) אבל לא ממש עובר דרכו.

Stepping Evaluator

המעריך הקצר הוא קצר-ראייה בכך שלא להבחין מדינות לאחר איסוף של נקודת ציון היעד. אַחֵר פונקציה ערך מוגדר לתיקון זה, אשר גורם את הספינה כדי לאסוף את נקודת היעד הנוכחית בסופו של דבר עמדה חיובית לאיסוף נקודת ציון אחרי זה. ההערכה של הספינה להיות במדינה מסוימת היא מחושב כדלקמן: (נוסחה 7 עמוד 8 באמצע).

שבו SW הוא מספר נקודות ציון שנאספו עד כה לאורך התוואי המומלץ על ידי תכנון התוואי שלב (סעיף VII), s'r הוא היחס בין המרחק נסע מנקודת המוצא האחרונה לנקודת היעד הבאה ב על פי מפת המרחק (סעיף VI-A) ו- s היא מהירות הספינה, כדי לעודד את אלגוריתם ההיגוי כדי לשמור על המומנטום. ערכי α הם משקולות שצריכות כדי להיות מכוון. שים לב כי s'r מוגדר באופן דומה ל- sr עבור קוצר הראייה מעריך, עם הבדל אחד מכריע: sr רואה את היעד דרך השורש, בעוד s'r רואה את היעד נקודת ציון ב. זוהי ההבחנה בין קוצר הראייה ו דריכה מעריכים: פעם אחת את נקודת היעד הוא שנאסף, המעריך הקצר רואה את כל המדינות טובות באותה מידה ואילו מעריך דריכה מתחיל אופטימיזציה של המרחק אל נקודת הגישה הבאה. בחירה αr <αw אומר שיש הוא צעד (קפיצה רציפה) בפרס הקשור איסוף נקודת הגישה הבאה (ראה איור 8). ההערכה במפורש אינה מתגמלת את הסוכן עבור איסוף נקודות ציון מוקדם (מתוך סדר המסלול). אחרת הנהג MCTS יש נטייה לעשות מעקפים בתאווה לאסוף נקודות ציון, אשר מתברר בדרך כלל להיות מזיק בטווח הרחוק. מאז כמות הזמן נלקח לא מיוצג ישירות ההערכה, הערכה שונה משמש במדינות סופיות כאשר כל נקודות הציון נאספות: (נוסחה 8 עמוד 8 למטה).

איפה W הוא מספר נקודות ציון, Tout הוא פסק זמן בין איסוף נקודות ציון, t ≤ ToutW הוא המספר הכולל של צעדים זמן נלקח כדי לאסוף את כל נקודות ציון, ו αt הוא פרמטר לכוונן. זה מבטיח כי מדינות מסוף יש הערכה של לפחות αww, שהוא גבוה יותר מזה של כל מצבים שאינם סופניים, ומצבים סופניים אלו בפחות זמן יש הערכה גבוהה יותר מאשר מדינות סופיות אשר לקח יותר זמן. מעריך הדריכה פעל היטב בתחרות ה- PTSP, אבל יש כמה פגמים. בעיקר, אם הספינה חייבת להיכנס את הכיוון הנגדי לאחר איסוף נקודת ציון, בתחילה לאחר איסוף אותו פונקצית ערך יהיה ירידה כמו הספינה נוסע רחוק יותר מנקודת המטרה החדשה, ואנו נעשה זאת יש sr <0. זה לפעמים גורם אלגוריתם נהיגה לא איסוף נקודת ציון, כמדינות שבהן הוא נאסף להיראות גרוע יותר מאשר מדינות שבו הוא לא. מצב זה יכול ניתן לראות בתרשים 8 כמו לטבול בפרס מיד לאחר מכן את ספייק עבור איסוף נקודת ציון; אם עומק הטבילה חרג מעל גובה ספייק, הסבירות היא כי הבקר נכשל לאסוף את נקודת ציון. לפתור את זה בעיה דורש איזון זהיר של הפרמטרים αw ו Αr, כדי להבטיח איסוף נקודת ציון תמיד התוצאות ברשת- עליית ערך. (תמונה 8 עמוד 8).

**TSP solvers (חלק 7)**

סעיף זה מתאר את השיטות שנבדקו לתכנון סדר שבו נקודות ציון הם ביקרו. ההזמנה היא מכרעת ביצועים, שכן הזמנה טובה תגרום מהיר וקצר מסלולים הזמנה רע יכול להיות בלתי אפשרי לחצות ( הבקר ייגמר הזמן בין נקודות ציון). נוסף, טכניקות החיפוש המשמשים ההיגוי הספינה לא מתכננים רחוק מספיק קדימה כדי למצוא הזמנות דרך טובה. בכל המקרים, המרחק בין נקודות ציון מתייחס למרחק המבול נלקח מתוך מפות המרחק מראש (סעיף VI-A) ו לא המרחק האוקלידי.

הנקודה הקרובה ביותר עכשיו

"המתכנן" הפשוט ביותר שנבדק אינו מתכנן מראש. זה תמיד מורה לאלגוריתם ההיגוי לאסוף את נקודת הגישה הקרובה ביותר. נקודת היעד של היעד עשויה להשתנות בנתיב שבין שתי נקודות ציון, כאשר אחד הופך קרוב יותר מאשר אחר. זה יכול להוביל להתנהגות בלתי החלטית בבקר, נקודת היעד עשויה להשתנות לעתים קרובות ואין נקודות ציון הם שנאספו. כל שאר שיטות התכנון נמנעות מבעיה זו מחליט מראש הזמנה מראש.

נקודת הגישה הקרובה ביותר

מתכנן זה בוחר נקודת ציון הזמנת בחמדנות, על ידי בחירה הנקודה הקרובה ביותר בכל צעד. במילים אחרות, ith הדרך היא הדרך הקרובה ביותר (לא להתרחש מוקדם יותר במסלול) אל (1 - i) נקודת ציון. אלה לא תמיד ניתן לחצות את ההזמנות, שכן הזמנת חמדן עלול לגרום נקודות ציון סמוכות כי הם רחוקים מאחד לשני.

TSP Planner

כל המפות בשימוש בניסויים יש 30, 40 או 50 נקודות ציון. פתרונות ל -51 צומת מינימום נתיב המילטון בעיות ניתן למצוא במהירות באמצעות מרובת חמדנים (22) ו 3-opt שיפור מקומי [23], [24] ואת אלה קרובים מספיק כדי אופטימלית למטרות שלנו. קטע מרובה [22] הוא חמדן חמדני לבנייה שביל המילטוניאני. האלגוריתם מוצא את קבוצת הקצוות את הנתיב, הוספת iteratively את הקצה עם משקל מינימלי זה אינו מבטל את הנתיב עד כה (כלומר אינו יוצר מחזור או לגרום לצומת בעל תואר גדול מ -2). 3-opt [23] היא טכניקת חיפוש מקומית עבור זיקוק נתיב כזה כפי שנבנה על ידי שבר מרובים: מהלך 3-opt מורכב של הסרת שלושה קצוות מהשביל והוספת שלושה קצוות כדי להשיג נתיב חדש; האופרטור 3-opt חוזר על עצמו 3-opt המהלכים כי להפחית את העלות של הנתיב, עד לא כזה מהלכים אפשריים. עבור כל משולשת הקצוות של השביל, שניים מהלכים 3-opt הם אפשריים, כפי שמוצג באיור 9. מתכנן זה משתמש בשבר מרובים 3-opt, עם קצה משקולות מחושב על ידי מפת המרחק, כדי לחשב את המסלול. (תמונה 9 עמוד 9)

Physics Enabled TSP Planner

המתכנן TSP מניח כי הזמן שנדרש כדי לחצות א המסלול הוא פרופורציונלי לאורכו. כאמור בסעיף II-B, זה לא נכון ב PTSP: מסלול ישר ארוך ללא חד סיבובים יכולים לעתים קרובות לחצות מהר יותר מאשר קצר יותר יותר מתפתל המסלול. כדי לזהות פניות חדות, כדאי להעריך את הזוויות שבו הספינה תיכנס ותשאיר כל נקודת ציון על מסלול מסוים. זה לא פשוט את הזווית של קו ישר נמשך בין נקודת ציון אחת לאחרת, כמו קו זה עשוי להיות חסום. מפות המרחק ניתן לחצות למצוא נתיב מ u ל V. תן dv להיות מפה המרחק עבור נקודת ציון נ ​​'ואז נתיב הוא p0, p1,. . . , Pk, כאשר p0 = u, pk = v, pi + 1 הוא השכן של pi אשר dv [pi + 1] הוא מינימלי. מרחק זה נתיבי נתיב מפות, אם כי "קצרים" ביחס ל מדד מסוים, הם אומדן אומלל של הנתיבים שנלקחו על ידי בקר ההיגוי של MCTS. נתיב הנתיב לעבר u כלפי V, מסומן - → uv, הוא קירוב של הכיוון שבו הספינה עוזבת כאשר נסיעה לכיוון V, או נכנס u בעת נסיעה מ v המתקבל על ידי מעקב אחר נתיב חציית נתיב המרחק מ – u עד v במקרה הראשון שבו הקו בין u לבין נקודה נוכחית הוא נחסם. - → uv נלקח להיות וקטור יחידה לכיוון pi -u. זה מתואר באיור 10. זה תהליך של הליכה לאורך השביל עד קו הראייה עם נקודת המוצא אבודה, ולא למשל. דורך לאורך קבוע המרחק, מבטיח כי הנתיב כיוונים לא סובלים אותו הטיה כלפי תנועה אלכסונית אורתוגונלית כמו השבילים עצמם ובכך מקרוב יותר את הכיוון של הספינה (בהנחה שאלגוריתם ההיגוי אינו בוחר נתיב שונה לחלוטין למסלול חציית מפת המרחק, אשר אינה מובטחת). תהליך זה דומה למחרוזת טכניקה [25] המשמשת לעתים קרובות ב pathfinding עבור משחקי וידיאו.

"הישיר" היוריסטי מוצג גם, אשר מבוסס על העיקרון כי נתיבים בין שתי נקודות ציון כי יש להעדיף פחות מכשולים על נתיבים אשר יש מכשולים רבים פונה חזק. נמדדת ישירות על ידי חישוב היחס בין המרחק האוקלידי לבין המרחק המרחק בין שני נקודות ציון. אם יש כמה המכשולים יחס זה יהיה קטן (נוטה לכיוון 1), עם זאת אם הנתיב ארוך ומורכב, יחס זה יהיה גדול. לָכֵן נתיב נחשב עקיף יותר ככל שהוא חורג ממנו קו ישר בחלל פתוח 3-opt בדרך כלל מבקש למזער את הסכום של משקולות קצה בדרך. כדי להתחשב בשיקולים הפיזיים המתואר לעיל, המטרה היא למזער את פונקציית העלות, שילוב מונחים המענישים פניות חדות בנקודות ציון נתיבים עקיפים בין נקודות ציון בנוסף לסכום של משקולות. מקטע מרובה מספק רק ראשוני לנחש להיות מעודן על ידי 3-opt, כל כך מעט יהיה להשיג על ידי שינוי קטע מרובים בצורה דומה. עבור קודקודים u ו- v:

1. תן d (u, v) להיות המרחק הנתיב הקצר בין u ו- v, מחושב באמצעות מפת המרחק;

2) תן e (u, v) להיות המרחק האוקלידית בין u ו v;

3) תן - → uv להיות בכיוון הנתיב לעבר u כלפי v.

אז העלות של נתיב v0, v1,. . . , Vn, בהנחה כי הספינה

הוא בתחילה מול לכיוון u0, הוא (נוסחה עמוד 10 למעלה).

עבור קבועים βw ו βp. המונח הראשון הוא סכום הקצה משקולות. המונח שונה על ידי βp מודד את ההישג של את הנתיב מנקודת ציון אחת לאחרת כשיעור של מרחק הנתיב והמרחק האוקלידי. המונח שונה על ידי β מודד את החדות של הפניות הספינה צריכה לבצע בעת עזיבת נקודת המוצא ובנסיעה דרך כל נקודת ציון. אם עובר דרך נקודת ציון נתונה אינו מחייב שינוי כיוון, נכנסות ו ווקטורים יוצאים לכיוונים מנוגדים, ולכן הנקודה שלהם המוצר הוא -1 (כלומר העלות שלילית). אם עובר דרך דרכון דורש 180 מעלות סיבוב, מוצר הנקודה הוא 1 (כלומר העלות חיובית). Β ו βw הם פרמטרים להיות מכוון. 3-opt משתמש זה האוריסטי בדרך הטבעית: כאשר שוקלים אם לבצע מהלך 3 opt, הערך של c (v0, ... vn) נחשב במקום אורך הנתיב הכולל. שים לב שאם Β = = βw = 0, המתכנן הזה זהה למתכנן ה- TSP ללא פיזיקה.

**הגדרת הניסוי (חלק 8)**

כל הניסויים המתוארים במאמר זה בוצעו במפות 20 המשמשות להפעלת ההערכות הסופיות של תחרות ה- PTSP של IEEE CIG 2012, בכל מפה שיחק 10 פעמים. מפות אלה מכילות 30, 40 ו 50 נקודות ציון, מה שהופך את הבעיה הרבה יותר מאתגרת מפות של תחרות WCCI (רק 10 נקודות ציון). גַם, זה אוסף של מפות מאפשר השוואה פשוטה עם את שאר הערכים לתחרות. בנוסף, המכונה המשמש להפעלת הניסויים הוא אותו שרת שממנו התחרות הופעלה (שרת Intel Core i5 ייעודי, 2.90GHz 6MB, ו 4GB של זיכרון), אשר מספק אמין השוואות.

הניסויים המוצגים במאמר זה מנתחים את ההשפעה של כל אחד מהפרמטרים הבאים:

• שיטת חיפוש: DFS, MC או MCTS. הפרמטר C עבור משוואה 4 נקבע ל- 1, ערך שנקבע באופן אמפירי.

• עומק סימולציות (D): מספר פעולות מאקרו בשימוש, מוגדר 120, 24, 12, 8, 6 ו - 4.

• אורך פעולה מאקרו (T): כל עומק סימולציה מוקצה לאחד מהערכים הבאים עבור T: 1, 5, 10, 15, 20 ו -30, בהתאמה. שים לב כי כל זוג נבדק (Di, Ti) מייצרת מבט קדימה בעתיד של די × Ti = 120 צעדים (או פעולות בודדות). הדבר יוצר השוואה הוגנת בין אלגוריתמים המשתמשים בערכים שונים עבור שני פרמטרים אלה.

• פונקציה מעריך: מיופי מעריך לעומת דריכה Evaluator.

• TSP Solver: נקודת ציון הקרובה עכשיו, המתכנן נקודת הדרך הקרובה ביותר, מתכנן TSP ו פיזיקה מופעל TSP Planner

הניסויים חולקו לשתי קבוצות של בדיקות. קבוצה ראשונית של ניסויים היה לרוץ על מנת למצוא נהג טוב עבור PTSP, המשלבת רק את ארבעת הפרמטרים הראשונים שתוארו לעיל. בדיקות אלה נועדו לזהות את הנהגים המסוגלים להגיע לרוב נקודות הציון של כל מפה, תוך שימוש בפותר ה- TSP הפשוט ביותר שהוכן עבור מחקר זה (נקודת הגישה הקרובה ביותר, ראה סעיף VII-A). זה גם מאפשר לנו להתעלם כמה ערכי פרמטר כי לבצע גרוע יותר, על מנת למקד את אצווה השני של בדיקות על ערכי הפרמטר המבטיחים ביותר, שבו שלושת הנותרים TSP פותחים. השוואת שתי הוצאות להורג עם כמה ריצות אינה טריוויאלית כפי שהיא נראית. האפשרות הפשוטה ביותר היא לחשב את הממוצע של נקודות ציון ביקר ואת הזמן שנדרש לעשות זאת. עם זאת, זה עלול להוביל כמה מצבים לא הוגנים. דוגמא תהיה שני נהגים, A ו- B, אשר להשיג כמויות שונות של נקודות ציון בממוצע (WA = 30, wb = 29) עם הזמנים שלהם בהתאמה (ta = 2500, tb = 1800). לאחר הגדרת המשחק של פותר אשר הוא טוב יותר (כמתואר בסעיף II), A צריך להיחשב טוב יותר מאשר B בגלל w> wb. עם זאת, ניתן לטעון כי B הוא טוב יותר כפי שהוא הרבה יותר מהר ואת ההבדל בין נקודות ציון הוא לא גדול מדי. לכן, שני צעדים שונים נלקחים כדי להשוות את התוצאות:

• יעילות: מספר נקודות ציון ביקר, בממוצע. ערך זה הוא להיות מוגדל.

• יעילות: היחס t / w, המייצג ממוצע של הזמן הדרוש כדי לבקר בכל נקודת ציון. הבעיה עם יחס זה היא כי היא לא בקנה מידה טוב כאשר כמות נקודות ציון הוא קטן מאוד, או אפילו 0. לכן, רק התאמות שבו כל נקודות ציון כבר ביקרו נחשבים למדוד זה. ככל שערך זה קטן יותר, כך יינתן הפתרון.

**תוצאות וניתוח (חלק 9)**

סעיף זה מציג את התוצאות של כל הניסויים שבוצעו במחקר זה. תוצאות אלה זמינות גם במסמך 1 ובאתר התחרות PTSP. המסמך מכיל קישורים לכל אחת מהריצות, המפרטות מידע סטטיסטי והתוצאה של כל התאמה. בנוסף, הדף המכיל את פרטי ההפעלה מכיל קישורים המאפשרים לצפות בכל ההתאמות ביישומון Java.

אורך מאקרו פעולה ומעריך פונקציה

קבוצה הראשונה של הניסויים מנתחת את ההשפעות של שימוש באורכים שונים מאקרו פעולה פונקציות ערך. במקרה זה, כל הבקרים להעסיק את נקודת הגישה הקרובה עכשיו TSP פותר (כמתואר בסעיף VII-A). המטרה היא להיות מסוגל לזהות את התכונות הטובות ביותר עבור הנהג.

איורים 11 ו -12 מראים את מספר נקודות הציון המתבצעות באמצעות הפונקציות של מיופיוס וסטייפ, בהתאמה, למדידת יעילותם של הפותחים. שתי התמונות כוללות קו אופקי המציג את המספר המרבי של נקודות ציון אפשריות (39), שמקורן בחלוקת נקודות ציון במפות שנבדקו (7 מפות עם 30 נקודות ציון, 8 מפות המכילות 40 ו -5 מפות עם 50). (תמונות 11, 12 עמוד 11).

שני איורים אלו מראות כי הגישה מיופיק מבצעת מעט יותר גרוע מאשר פונקציה מעריך דריכה. עבור שתי הפונקציות, מספר נקודות הציון שביקרו גדל כאשר אורך פעולות המקרו מגיע לערך גבוה (10 עד 20). טבלה I מראה כי מספר נקודות הציון ביקר ירידה קלה כאשר אורך מאקרו פעולה גדל ל 30. תופעה זו מתרחשת בשלוש שיטות החיפוש המוצגות במאמר זה, דבר המצביע על כך צמצום הבעיה המושגת על ידי הכללת פעולות מאקרו מוביל לביצועים טובים יותר. באופן כללי, DFS מבצעת יותר גרוע מהאלגוריתמים האחרים, ואפילו באותם מקרים שבהם אורך הפעולות המקרו קטנות (1 ו -5), מתקבלות תוצאות טובות יותר עם MC ו- MCTS. כדאי גם לציין שאף אחד מהאלגוריתמים אינו מסוגל להגיע ליעילות אופטימלית (39 נקודות ציון). במילים אחרות, כל השיטות לא מצליחות לתפוס את כל נקודות הציון בלפחות אחת מ 200 המשחקים ששימשו, מה שמראה את המורכבות של המשחק.

היעילות של הבקרים מחושבת כממוצע של יחס היעילות (זמן / נקודות ציון) באותם משחקים שבהם כל נקודות הציון הם ביקרו. איורים 13 ו -14 מראים תובנות עבור מעריכי הפונקציה מיופיק וסטייפ בהתאמה. כפי שהוסבר בסעיף VIII, יחס זה יכול להיות מטעה אם מספר נקודות ציון ביקר רחוק מכמות אותם במפות. מסיבה זו, אורך מאקרו פעולה של 1 אינו נכלל בתמונות אלה.

היחס המתקבל (ככל שהקטן, טוב יותר) מראה שוב ששיטות MC ו- MCTS משיגות תוצאות טובות יותר מאשר DFS, ותוצאות אלו טובות יותר כאשר אורך המאקרו גבוה (10 עד 20). עם זאת, כפי שמוצג בטבלה I, העלאת הערך של T עד 30 מספקת תוצאות גרועות יותר. בהשוואה בין מעריכי הפונקציה Myop ו- Stepping, יעילות טובה יותר מתקבלת גם עם מעריך ההפסדה, ובמקרה זה ההבדל משמעותי יותר (כ -30 נקודות).

טבלה I מציגה את התוצאות המספריות של הבדיקות, פיגורציות שנבדקו, כולל השגיאה הסטנדרטית של הממוצעים שהתקבלו. נקודה אחת של עניין היא היחס המתקבל כאשר אורך של מאקרו פעולה הוא 1: ארבעה מתוך שש תצורות אינם מסוגלים לספק תוצאות סבירות. רק MC ו- MCTS, באמצעות הפונקציה מעריך דריכה, הם יכולים לבקר ב לפחות פעם אחת את כל נקודות ציון מפוזרות סביב מפות נבדק. תוצאה מעניינת נוספת היא כי MCTS משיגה את הביצועים הטובים ביותר עבור כמעט כל שילובים של אורך מאקרו פעולה ואת הפונקציה ערך. MC רק מקבל תוצאות טובות יותר לשניים מקרים: מספר נקודות ביקורים, באמצעות פונקציית דריכה מעריך, עבור מאקרו פעולה באורך של 15 ו – 20.

במקרים רבים, התוצאות המתקבלות על ידי כל אלגוריתם דומים מאוד זה לזה. מסיבה זו, מבחן Mann-Whitney Wilcoxon (MW-test) בוצע על מנת לאשר אם קיים הבדל סטטיסטי באותם מקרים שבהם התוצאות דומות. לשם החלל, לא כל תוצאות MW-test כלולות במאמר זה, אך המעניינות ביותר מוסברות כאן: המבחנים מראים כי קיים הבדל סטטיסטי בין MCTS לבין שתי השיטות האחרות עבור 15 ו -20 מאקרו פעולה (T = 15, p ערכי של 0.109 ו 0.001 כנגד DFS ו- MC בהתאמה. 5 × 10-6 ו 1.5 × 10-6 עבור T = 20). במילים אחרות, זה מראה כי היחס התחתון המוצג באיור 14 מגיב על ביצועים טובים יותר של MCTS על MC ו DFS במקרים אלה. באשר ליעילות, שיטות אלה מספקות תוצאות זהות בתרחישים שהוזכרו.

TSP Solvers

לאחר הניתוח שבוצע בחלק הקודם, חלק מהערכים עבור הפרמטרים שנבדקו מושלכים על מנת להתמקד באלו המספקים תוצאות טובות יותר. המטרה בשלב זה היא להעריך כיצד פותר TSP שונים להשפיע על בקרי נבדק. במקרה זה, הפרמטרים מוגדרים באופן הבא:

• שיטת חיפוש: DFS, MC או MCTS.

• עומק סימולציות (D): 12, 8 ו - 6.

• אורך מאקרו פעולה (T): 10, 15 ו -20.

• מעריך פונקציה: רק דריכה מעריך

טבלה 2 מציגה את התוצאות המתקבלות עם כל פותרי ה- TSP פרמטרים אלה.

ניתן לבצע מספר תצפיות על הביצועים המוצגים בבדיקות אלו. ראשית, הכולל הטוב ביותר TSP פותר הוא הפיזיקה מופעל המבול למלא Planner, קבלת ביצועים טובים יותר מבחינת נקודות הביקור ביקר יחס הביקורים. תוצאה זו מאשרת את ההשערה המצויה בסעיף II-B: ביצועים טובים יותר מתקבלים כאשר הפיזיקה של המשחק נלקחים בחשבון בעת תכנון סדר נקודות הביקורת לבקר. עבור מתכנן ה- TSP הטוב ביותר, MCTS מתנהג טוב יותר הן MC והן DFS בשני הצעדים שננקטו. לגבי אורך המקרואקציה, ערך של 15 מספק את התוצאה הטובה ביותר ביחס של נקודות ציון, כאשר 20 נראה כי הבחירה הטובה ביותר עבור היעילות של הבקר. למעשה, הבקר כי השיג את המקום הראשון בתחרות PTSP היה התצורה באמצעות MCTS, פיזיקה מופעל Flood Fill Planner ו- T = 15.

שוב, מבחנים סטטיסטיים חושבו בין הערכים המעניינים ביותר. לדוגמה, Flood Fill Planner ופיסיקה מופעלת Flood Fill Planner הושוו עבור MCTS עם ערכי T של 15 ו -20. MW-Test מספק ערכים של 0.015 ו -0.0252, בהתאמה, מראה ביטחון גבוה במקרה של T = 15 ו רק אינדיקציה הוגנת לכך שההתפלגויות שונות עבור T = 20. תוצאה מעניינת נוספת היא שהפתרון הקרוב ביותר לפתרון נקודת המוצא נראה כי הוא מניב תוצאות טובות יותר מאשר גרסת המתכנן של אותו אלגוריתם. זה יכול להיות מפתיע כי התוואי המתוכנן של נקודות ציון מתנהג גרוע יותר מאשר גרסה כי רק רודף את נקודת הגישה הקרובה ביותר בכל עת, אבל ההסבר הוא שוב בפיזיקה של המשחק. הנקודה הקרובה ביותר עכשיו TSP פותר רק לוקח בחשבון את המרחקים בין נקודות ציון, אבל מתעלם האינרציה ואת המהירות של הספינה בעת שנסע ברחבי המבוך. אינרציה זו יכולה לקחת את הספינה קרוב יותר לנקודת ציון אחרת (אשר לא יכול להיות הבא הבא לבקר בגרסה המתוכננת) אבל ברור, זה יכול להיות מהיר יותר לבקר את זה במקום לשנות את הקורס כדי לעקוב אחר התוכנית. פתרון טוב יותר מזה שסופק על ידי נקודת הדרך הקרובה עכשיו עכשיו פותר מתקבל עם הפיזיקה מופעלת המבול למילוי Planner (80.10 לעומת 81.78, MW-test p ערך של 3.07 × 10-5), תוך שימוש ב- T = 15 אשר נראה הכי טוב הכולל ערך עבור אורך מאקרו פעולה.

ניתוח לפי מפה

השוואה מעניינת בין הטכניקות המוצגות במאמר זה היא המפה על ידי השוואת מחירים. זה אפשרי כי שיטות מסוימות להתנהג טוב יותר במפות שונות, תלוי איך המכשולים מופצים סביב המבוך. מערכת ניקוד המדגישה את ההבדלים בין המפות היא זו שמשמשת בתחרות ה- PTSP (ראה פרק II-C). זו תוכנית הדירוג פרסים נקודות עבור הדירוגים על כל אחת מהמפות שבו הבקרים מוערכים. הזוכה הסופי הוא הבקר שמשיג יותר נקודות בסך הכל. עבור כל מפה, נקודות מוענקות כמו באליפות פורמולה אחת: 25 נקודות עבור הבקר הטוב ביותר, 18 נקודות עבור השני הטוב ביותר, 15 עבור השלישי 12, 10, 8, 6, 4, 2 ו 1 עבור הרובוטים מ 4 עד 10, בהתאמה. התפלגות הנקודות בתכנית זו מעניקה את הבקר הטוב ביותר על כל מפה. טבלה III מציגה את הנקודות שהושגו בכל מפה, בהתאם לתוכנית הדירוג של PTSP. הבקרים המוצגים הם עשרת הרובוטים הטובים ביותר המדורגים בדרך זו. מעניין לראות כי הבקר הטוב ביותר הוא עדיין אחד שהוגש לתחרות PTSP כי מדורגת הראשון. עם זאת, הוא אינו מסוגל להשיג את התוצאות הטובות ביותר בכל המפות: במפה 18, הוא מקבל 0 נקודות (שים לב שזה לא אומר שזה לא לבקר בכל נקודת ציון.זה רק גרוע יותר הבקר העשירי במפה זו בפרט ). התוצאה הטובה ביותר במפה זו מתקבלת על ידי אותו אלגוריתם, אך משתמשת ב- T = 20 כאורך פעולה מאקרו, דבר המצביע על כך שבמפות מסוימות פרמטר זה חיוני לקבלת ביצועים טובים. תוכנית PTSP תחרות מעדיף בקרי אלה לבקר נקודות ציון יותר, עם הזמן המושקע relegated כמו מפסק תיקו כאשר הביקורים זהים. זו הסיבה בדירוגים אלה אלגוריתם המשתמש בנקודת הדרך הקרובה ביותר עכשיו פותר מוסמך השני (כפי שניתן לראות בטבלה 2, את היעילות של פותר זה TSP הוא גבוה). (טבלאות עמוד 13)

איור 15 מציג את יחסי נקודת הציון למפה, המאפשר השוואה בין היעילות של הבקרים השונים. לשם הבהירות, תמונה זו מציגה רק את ארבעת הבקרים הטובים ביותר על פי הדירוג של סעיף זה. כפי שניתן לראות, הגישה הטובה ביותר בדרך כלל צריך פחות זמן צעדים לבקר נקודות ציון, ואילו בקר השני לעתים קרובות משתמשת זמן רב יותר עבור זה. כמה יוצאים מן הכלל הם מפות 4 ו -6, כאשר הבקר באמצעות נקודת הדרך הקרובה עכשיו מספק פתרון טוב יותר. זה קשור מאוד לתוצאות המוצגות עבור מפות אלה בטבלה III, שם הבקר הטוב ביותר אינו משיג את הביצועים הטובים ביותר.

לבסוף, מפות 18 ו 19 חסרים כמה ברים עבור כמה בקרי. אלה הם המקרים שבהם הרובוטים לא יכלו לבקר בכל נקודות הציון בכל אחד מהתאמות המופיעות במפות הללו, ולכן חישוב היחס אינו תיאורי ביותר. במקרים אלה, הבקרות השיגו 0 נקודות במפות, כפי שמוצג בטבלה III.

ביצועים בתחרות PTSP

כפי שצוין קודם לכן, התצורה שסיפקה את התוצאות הטובות ביותר במחקר זה (MCTS, T = 15, דריכה המעריך פונקציה ו הפיזיקה מופעל המבול מילוי Planner) היה המנצח של התחרות PIGP CIG. זה יהיה מעניין לראות אם כל תצורות אחרות נבדק כאן היה זוכה בתחרות בכל מקרה, תוך שימוש בתוצאות שהושגו במחקר זה. ההנמקה מאחורי בדיקה זו היא כדלקמן: בתחרויות רבות, הבקרים ואת heuristics הם tweaked עד שהם מקבלים את הביצועים הטובים ביותר האפשריים. אפשר לשאול, אם כן, אם הזוכה בתחרות זו (או אחרת) מספקת שיטה מעולה, או רק היוריסטי טוב מאוד שיכול לעבוד עם טכניקות שונות של AI. המסמך המצוין בתחילת סעיף IX מכיל קישורים לתוצאות הפוטנציאליות שהיו מתקבלות אם כל בקר פוטנציאלי הוגש במקום זה שנשלח בפועל. השורות הבאות מסכמות את הנתונים הסטטיסטיים שנלקחו מההגשות הפוטנציאליות הבאות:

• אף אחד מהבקרים שנבדקו לא היה זוכה בתחרות באמצעות מעריך פונקציה מיופיק.

• מתוך התצורות שנבדקו, כל הבקרים שהשתמשו בפיסיקה של Flood Fill Planner, עדיין היו זוכים בתחרות.

• מתוך אותם בקרים ששימשו את Flood Fill Planner (ללא פיסיקה), כל אחד מלבד אחד היה מסיים את התחרות במקום הראשון. רק MC עם T = 10 לא ינצח, במקום השני במקום.

• רק בקרי ה- MCTS המשתמשים בפותר ה- TSP הקרוב ביותר למפתחים של נקודת הגישה היו מסיימים את המיקום הראשון בכל המקרים שנבדקו.

• עבור נקודת המבט הקרובה ביותר, עכשיו רק 14 מתוך 36 (38%) עדיין ינצחו.

• 38 מבין 63 המבחנים השונים שבוצעו (60%) היו זוכים בתחרות.

כמובן, צעדים אלה הם ממחישים, והם עשויים להיות שונים אם המתחרים האחרים הגישו בקרי שונות, אבל המבחן עדיין מאפשר ציור של כמה מסקנות. לדוגמה, מספרים אלה מצביעים על כך ש- MCTS, עם פונקציית ההערכה המתאימה, הוא האלגוריתם הטוב ביותר שהוגש עד כה לתחרות ה- PTSP. הערכים של T כדי להשליך הם 1, 5 ו -30, אבל כל הערכים האחרים (10, 15 ו -20) לייצר תוצאות טובות מספיק כדי לנצח את התחרות.

**מסקנות ועבודות עתידיות (חלק 10)**

מאמר זה מציג מחקר של האלגוריתם שזכה בתחרות PEEP IEEE CIG 2012, כמה וריאציות שלו, וכיצד הערכים השונים של המרכיבים של האלגוריתם משפיעים על הביצועים הכוללים של הבקר. ביצועים אלה נמדדים במונחים של יעילות (או כמה נקודות ביקורת הם ביקרו) ואת היעילות (את הצעדים הממוצע הזמן הנדרש כדי לבקר נקודות ציון) של בקרי. אחת המסקנות הראשונות שמציע מחקר זה היא כי השימוש בפעולות מאקרו ב- PTSP משפר את איכות הפתרונות בצורה עצומה. זה מעניין במיוחד בהתחשב בפשטות לחזור על T- פעמים פעמים של מאקרו פעולות בשימוש. אורך הפעולה של מאקרו שסיפק ביצועים טובים הוא ערכים מ -10 עד 20, קבלת התוצאות הטובות ביותר עם T = 15.

כדאי גם להזכיר שהתכנון של פונקצית ערך מתאימה, במיוחד עבור משחקים כמו ה- PTSP שבו המגבלות בזמן אמת, המונעות בדרך כלל מהאלגוריתם להגיע למצב משחק קצה, הוא חיוני כדי להשיג תוצאות טובות. במקרה הספציפי הזה, מעריך מצב מיופייק פשוט מתנהג בצורה גרועה בהשוואה למעריך דריכה מתוחכם יותר. היבט נוסף המודגש במאמר זה הוא התלות בין סדר נקודות הציון לבין סגנון הנהיגה של הבוט. במילים אחרות, איך הפיזיקה של המשחק להשפיע על רצף אופטימלי של נקודות ציון לעקוב. זה הוכח בתוצאות שהתקבלו כי פותר TSP רואה את הפיזיקה של המשחק משיגה תוצאות טובות יותר באופן משמעותי מאשר אחרים.

לבסוף, MCTS הוכח להתנהג טוב יותר מאשר שני אלגוריתמים אחרים בהשוואה במאמר זה, MC ו DFS. חשיבות מיוחדת היא העובדה כי כל אחד MCTS וריאציות שנבדקו במחקר זה, באמצעות הפונקציה דפוסי מעריך ואת המתאים מאקרו פעולה אורכים, היה זוכה בתחרות PTSP CIG. מחקר זה ניתן להרחיב בכמה דרכים. לדוגמה, יהיה מעניין לחקור כיצד צורת המפות משפיעה על הביצועים של האלגוריתמים, ובאיזה תנאים מסוימים נראה שאורכי מאקרו-פעולה מסוימים פועלים טוב יותר. בנוסף, כפי שנחשף במאמר זה, את העיצוב של מאקרו פעולות היא פשוט חזרה פשוטה של ​​פעולות. זה יהיה מעניין לפתח macroactions מתוחכמים יותר להשוות את הביצועים שלהם. אפשרות נוספת היא להמשיך ולשפר את האלגוריתמים המוצגים: על ידי בחינת התוצאות והמשחקים, ברור כי הביצועים אינם אופטימליים. שילוב MCTS עם טכניקות אחרות, כגון אלגוריתמים אבולוציוניים או TD-learning, יכול לשפר את התוצאות המוצגות במחקר זה. לבסוף, עוד מבחן מעניין יהיה לצמצם את זמן התקציב עוד ולבדוק איך זה משפיע על איכות הפתרונות שהושגו. ניסויים ראשוניים מראים כי לפחות MCTS עדיין מייצרת פתרונות איכותיים כאשר התקציב מצטמצם ל 10ms. עם זאת, יש צורך בבדיקות נוספות כדי להבין כיצד מגבלה זו משפיעה על שיטות החיפוש המוצגות כאן, כמו גם מציאת זמן התקציב המינימלי המאפשר טכניקות אלה כדי לשמור על תוצאות טובות בתחום זה ודומיהם, שבו ההחלטות צריך להילקח בתוך כמה אלפיות השנייה .