236501 מבוא לבינה מלאכותית

# תרגיל בית 1 – חלק יבש

## מגישים: מקסים ליפטרוב 327154464, עומרי רוזן 313223521

1. נגדיר את מספר הסידורים החוקיים אשר עונים על האילוצים המקלים, עבור k הובלות להיות .

במקרה הפשוט ביותר, בו , יש רק סידור אחד אפשרי (איסוף אחד, הורדה אחת) ולכן .

עבור מקרה כללי בו ישנן k הובלות, ישנן גם k אפשרויות לאיסוף הראשון. לאחר שבחרנו את האיסוף הראשון, אנחנו מקבלים את אותה הבעיה עבור כאשר בנוסף, צריך לשבץ את ההורדה של האיסוף הראשון (שלא משפיעה על שאר ההובלות, בהנחת קיבולת מטען אינסופית למשאית).

שיבוץ ההורדה של המטען שנאסף ראשון יכולה לפיכך להתבצע בכל אחד מ- המקומות שנשארו (המקום הראשון תפוס ע"י האיסוף של אותו מטען).

לכן נקבל: .

לפי אינדוקציה, נקבל ש:

|  |  |  |  |
| --- | --- | --- | --- |
| Calculation time |  | # possible paths |  |
|  | 16.791 | 113400 | 5 |
|  | 36.25 |  | 8 |
|  | 43.508 |  | 9 |
|  | 51.077 |  | 10 |
|  | 58.929 |  | 11 |
|  | 92.709 |  | 15 |

1. דרגת היציאה נקבעת לפי מספר האופרטורים שאפשר להפעיל על מצב מסוים (שמתאים לצומת במרחב החיפוש). בבעיה המתוארת, האופרטורים שנוכל להפעיל הם העמסה של מטען והורדה של מטען, כאשר ישנם סה"כ k מטענים. לא ייתכן, שבמצב כלשהו נוכל גם להעמיס את המטען וגם להוריד אותו, מכיוון שההורדה אפשרית רק כאשר ההעמסה קרתה כבר בפעולה קודמת למצב הנוכחי. לכן, לגבי כל אחד מהמטענים נוכל לכל היותר לבצע פעולה אחת (העמסה או הורדה) מכל מצב. מכיוון שמדובר ב-k מטענים, אלו לכל היותר k פעולות. יש לזכור שלאחר שנוריד מטען מהמשאית לא נוכל לבצע עליו יותר אף פעולה (אין העמסה חוזרת).

לכן, בכל מצב עד ההורדה הראשונה של מטען מהמשאית נוכל לבצע k פעולות, ודרגת היציאה של מצבים אלו תהיה k, היא גם דרגת היציאה המקסימלית במרחב החיפוש.

דרגת היציאה קטנה עם מספר ההורדות שבוצעו, כאשר במצב בו כל המטענים הורדו מהמשאית, דרגת היציאה תהיה מינימלית ושווה לאפס.

1. ממצב כלשהו ניתן לבצע פעולת העמסה או הורדה בלבד. אם ביצענו פעולת הורדה של מטען, ההובלה שמתאימה למטען תעבור מ-loaded ל-dropped. מכיוון שאין העמסה חוזרת, ההובלה תישאר מכאן והלאה בכל המצבים ב-dropped, בניגוד למצב הנוכחי בו היא ב-loaded. אם ביצענו פעולת העמסה, המטען שהועמס (ששייך להובלה ( יגרור את הכנסת ל-loaded. הובלה שנכנסה ל-loaded יכולה לעבור משם רק ל-dropped, ובכל מקרה מכאן והלאה תופיע ב-loaded או ב-dropped, בשונה מהמצב הנוכחי (בו לא הופיע באף אחד מהם). לפיכך, אין אפשרות לחזור למצב קודם, כלומר, לא ייתכנו מעגלים במרחב המצבים.
2. מלבד המצב ההתחלתי, יש למשאית 2k מקומות להיות בהם, כאשר כל אחד מקושר להעמסה של מטען אחד או להורדה של מטען אחד (תחת הנחת היסוד שאין מקום שמשותף לשתי העמסות, שתי הורדות או העמסה והורדה). בכל מקום שמקושר להעמסה של מטען, ההובלה ששייכת לו תופיע תחת loaded במצב זה, כי המשאית הגיעה לשם בשביל להעמיס אותו. בכל מקום שמקושר להורדה של מטען, ההובלה ששייכת לו תופיע תחת dropped במצב זה, כי המשאית הגיעה לשם בשביל להוריד אותו. לגבי שאר המטענים אין הגבלה, ולכל אחד מהם שלוש אפשרויות: להיות ב-loaded, ב-dropped או לא להופיע בכלל (לא העמסנו את המטען עדיין). בגלל אי התלות ביניהם מדובר ב- אפשרויות (עבור מקום מתוך 2k המקומות שאינם המצב ההתחלתי). כדי לקבל את מספר המצבים הכולל, נכפול גודל זה במספר המקומות האפשריים ונוסיף את המצב ההתחלתי. לכן, סה"כ:
3. הבורות לא ייתכנו. כאשר מאפשר לבצע עבור אחת מהובלות שהוזמנו ועוד לא בוצעו, קיים מצב שניתן לעבור אליו – העמסה עבור אותה ההובלה. אחרת המשאית מלאה, לכן לא ניתן להעמיס אך ניתן לבצע עבור כל אחת מההובלות שהעמסנו. לכן הבורות היחידים הם מצבים בהם לא ניתן להעמיס וגם לא ניתן להוריד בו זמנית, שזהו מצבי המטרה.
4. אורך מסלול מחושב לפי מספר הקשתות. בכל מסלול ממצב התחלתי למצב מטרה נבצע k העמסות ו-k הורדות. כל קשת (שמבטאת הפעלה של אופרטור) מתאימה להעמסה יחידה או הורדה יחידה. לכן סה"כ יהיו בכל מסלול למצב הסופי 2k קשתות.
5. הגרף הוא עץ מכוון וחסר מעגלים (ולולאות עצמיות). בנוסף גרף המצבים בעל עומק מקסימלי של 2k (אשר שייך למצב אחד בלבד – כשכל המטענים הורדו מהמשאית). בנוסף, לאורך ההתקדמות בגרף המצבים, ניתן לומר שדרגת היציאה היא מונוטונית לא עולה.
6. פונקציית העוקב תוגדר כך:
7. f. לאחר הרצת התוכנית נקבל את הפלט הבא:

Solve the map problem.

StreetsMap(src: 54 dst: 549) UniformCost time: 0.76 #dev: 17354 |space|: 17514 total\_g\_cost: 7465.52560 |path|: 137 path: [ 54 ==> 55 ==> 56 ==> 57 ==> 58 ==> 59 ==> 60 ==> 28893 ==> 14580 ==> 14590 ==> 14591 ==> 14592 ==> 14593 ==> 81892 ==> 25814 ==> 81 ==> 26236 ==> 26234 ==> 1188 ==> 33068 ==> 33069 ==> 33070 ==> 15474 ==> 33071 ==> 5020 ==> 21699 ==> 33072 ==> 33073 ==> 33074 ==> 16203 ==> 9847 ==> 9848 ==> 9849 ==> 9850 ==> 9851 ==> 335 ==> 9852 ==> 82906 ==> 82907 ==> 82908 ==> 82909 ==> 95454 ==> 96539 ==> 72369 ==> 94627 ==> 38553 ==> 72367 ==> 29007 ==> 94632 ==> 96540 ==> 9269 ==> 82890 ==> 29049 ==> 29026 ==> 82682 ==> 71897 ==> 83380 ==> 96541 ==> 82904 ==> 96542 ==> 96543 ==> 96544 ==> 96545 ==> 96546 ==> 96547 ==> 82911 ==> 82928 ==> 24841 ==> 24842 ==> 24843 ==> 5215 ==> 24844 ==> 9274 ==> 24845 ==> 24846 ==> 24847 ==> 24848 ==> 24849 ==> 24850 ==> 24851 ==> 24852 ==> 24853 ==> 24854 ==> 24855 ==> 24856 ==> 24857 ==> 24858 ==> 24859 ==> 24860 ==> 24861 ==> 24862 ==> 24863 ==> 24864 ==> 24865 ==> 24866 ==> 82208 ==> 82209 ==> 82210 ==> 21518 ==> 21431 ==> 21432 ==> 21433 ==> 21434 ==> 21435 ==> 21436 ==> 21437 ==> 21438 ==> 21439 ==> 21440 ==> 21441 ==> 21442 ==> 21443 ==> 21444 ==> 21445 ==> 21446 ==> 21447 ==> 21448 ==> 21449 ==> 21450 ==> 21451 ==> 621 ==> 21452 ==> 21453 ==> 21454 ==> 21495 ==> 21496 ==> 539 ==> 540 ==> 541 ==> 542 ==> 543 ==> 544 ==> 545 ==> 546 ==> 547 ==> 548 ==> 549]

10. g. השורה שמופיע מטה היא זו שקובעת שהאובייקטים מטיפוס mapState יהיו בלתי ניתנים לשינוי:

@dataclass(frozen=True)

שורה זו מופיעה מעל להגדרת המחלקה.

אין סיבה שבגינה נרצה שישתנו פרטיו של מצב מסוים (כי אז הוא יתאר מצב אחר).

כל שינוי של השדה junction.index (למשל באג בו שמים בשדה זה את אותו ערך כל הזמן) עלול לגרום לאלגוריתם לא לעבוד. בבדיקתנו, כשהגענו למצב חדש, נרצה לראות אם נתקלנו במצב זה כבר קודם. שינוי של מצבים, עלול להוביל אותנו למסקנה שגויה בשאלה זו.

13. לאחר הרצת התוכנית נקבל את הפלט הבא:

Solve the map problem.

StreetsMap(src: 54 dst: 549) UniformCost time: 0.68 #dev: 17354 |space|: 17514 total\_g\_cost: 7465.52560 |path|: 137 path: [ 54 ==> 55 ==> … ==> 548 ==> 549]

StreetsMap(src: 54 dst: 549) A\* (h=0, w=0.500) time: 0.70 #dev: 17354 |space|: 17514 total\_g\_cost: 7465.52560 |path|: 137 path: [ 54 ==> 55 ==> … ==> 548 ==> 549]

StreetsMap(src: 54 dst: 549) A\* (h=AirDist, w=0.500) time: 0.17 #dev: 2015 |space|: 2229 total\_g\_cost: 7465.52560 |path|: 137 path: [ 54 ==> 55 ==> … ==> 548 ==> 549]

כאשר המסלול שנמצא זהה עבור כולם, וזהה לזה שמופיע בשאלה 10.

מספר פיתוחי המצבים היחסי שנחסך (לעומת הריצה הקודמת) הינו:  *, כלומר, חסכנו 87.6%.*

*14. להלן גרף המתאר את עלות הפתרון (לפי מרחק) שמצא אלגוריתם A\* הממושקל ואת מספר הצמתים שפותחו בדרך לפתרון, כתלות במשקל עצמו.*

תמונה שמכילה מפה, טקסט

התיאור נוצר באופן אוטומטי

ראשית ניתן לראות שהמגמות של עלות הפתרון ושל מספר הצמתים שפותחו, עם עליית המשקל, הן הפוכות. המגמה של מחיר הפתרון (לפי מרחק) היא לעלות עם עליית המשקל מחצי לאחד, בעוד מגמת מספר הצמתים המפותחים היא ירידה, בהתאם לכלל האצבע הכללי שנאמר בהרצאה.

בנוסף ניתן לראות ששני הגרפים אינם מונוטוניים, בהתאם לאמירה שכלל האצבע אינו נכון באופן גורף לכל זוג משקלים.

אנחנו היינו בוחרים ב-w אשר החץ מצביע עליו בשרטוט מטה. במצב זה עלות הפתרון קרובה מאוד לזו האופטימלית, וגם מספר הצמתים שפותחו קרוב מאוד למספר האופטימלי שמושג עבור משקלים גבוהים יותר.

תמונה שמכילה מפה, טקסט

התיאור נוצר באופן אוטומטי