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 T_k.
במקרה הפשוט ביותר, בו $k = 1$, יש רק סידור אחד אפשרי (איסוף אחד, הורדה אחת) ולכן $T_1 = 1$.
עבור מקרה כללי בו ישנן k הובלות, ישנן גם k אפשרויות לאיסוף הראשון. לאחר שבחרנו את האיסוף הראשון, אנחנו מקבלים את אותה הבעיה עבור $k - 1$ כאשר בנוסף, צריך לשבץ את ההורדה של האיסוף הראשון (שלא משפיעה על שאר $k - 1$ ההובלות, בהנחת קיבולת מטען אינסופית למשאית).
שיבוץ ההורדה של המטען שנאסף ראשון יכולה לפיכך להתבצע בכל אחד מ- $2k - 1$ המקומות שנשארו (המקום הראשון תפוס ע"י האיסוף של אותו מטען).
לכן נקבל: $T_k = k\,(2k - 1)\,T_{k-1}$.
לפי אינדוקציה, נקבל ש:

$$T_k = k! \cdot \prod_{i=1}^{k} (2i - 1)$$

2.

k	# possible paths	$\log_2(\#\ possible\ paths)$	Calculation time
5	113400	16.791	$< 1_{sec}$
8	$8.173 \cdot 10^{10}$	36.25	76.117_{sec}
9	$1.25 \cdot 10^{13}$	43.508	3.235_{hours}
10	$2.376 \cdot 10^{15}$	51.077	0.07_{years}
11	$5.488 \cdot 10^{17}$	58.929	16.208_{years}
15	$8.095 \cdot 10^{27}$	92.709	$239k_{\{million\ years\}}$

3. 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 d_i 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.

4. 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 d_i) יגרור את הכנסת d_i 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.

5. 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 $k-1$ 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- 3^{k-1} 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: $1+2k\cdot 3^{k-1}$

6. הבורות לא ייתכנו. כאשר $TrunkCapacity$ מאפשר לבצע $pick$ 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 $drop$ עבור כל אחת מההובלות שהעמסנו. לכן הבורות היחידים הם מצבים בהם לא ניתן להעמיס וגם לא ניתן להוריד בו זמנית, שזהו מצבי המטרה.

7. 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.

8. הגרף הוא עץ מכוון וחסר מעגלים (ולולאות עצמיות). בנוסף גרף המצבים בעל עומק מקסימלי של 2k (אשר שייך ל-k מצבים בלבד – כשכל המטענים הורדו מהמשאית, והמיקום הוא באחת מנקודות ההורדה). בנוסף, לאורך ההתקדמות בגרף המצבים, ניתן לומר שדרגת היציאה היא מונוטונית לא עולה.

9. פונקציית העוקב $Succ: S \to \mathcal{P}(S)$ תוגדר כך:

$$
\begin{aligned}
Succ(s) = \Big\{ & (d_i.pick, s.Loaded \cup \{d_i\}, s.Dropped) \mid i \\
& \in \{1,2,\dots,k\} \cap (d_i \in D) \cap (d_i \notin s.Loaded \cup s.Dropped) \\
& \cap \left(d_i.pkgs \le TrunckCapacity - \sum_{d \in s.Loaded} d.pkgs \right) \Big\} \\
& \cup \{(d_i.drop, s.Loaded \setminus \{d_i\}, s.Dropped \cup \{d_i\}) \mid i \\
& \in \{1,2,\dots,k\} \cap (d_i \in D) \cap (d_i \in s.Loaded)\}
\end{aligned}
$$

10. 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]

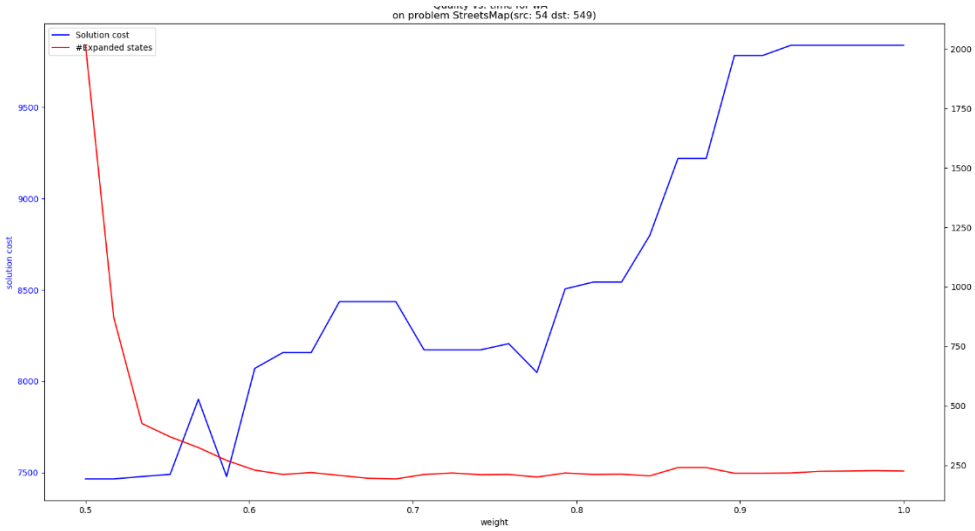
StreetsMap(src: 54 dst: 549) A* (h=0, w=0.500) time: 0.70 #dev: 17354 |space|: 17514 total_g_cost: 7465.52560 |path|: 137 path: [54 ==> 55 ==> … ==> 548 ==> 549]

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: $\frac{17354-2015}{17354} = 0.876$, כלומר, חסכנו 87.6%.

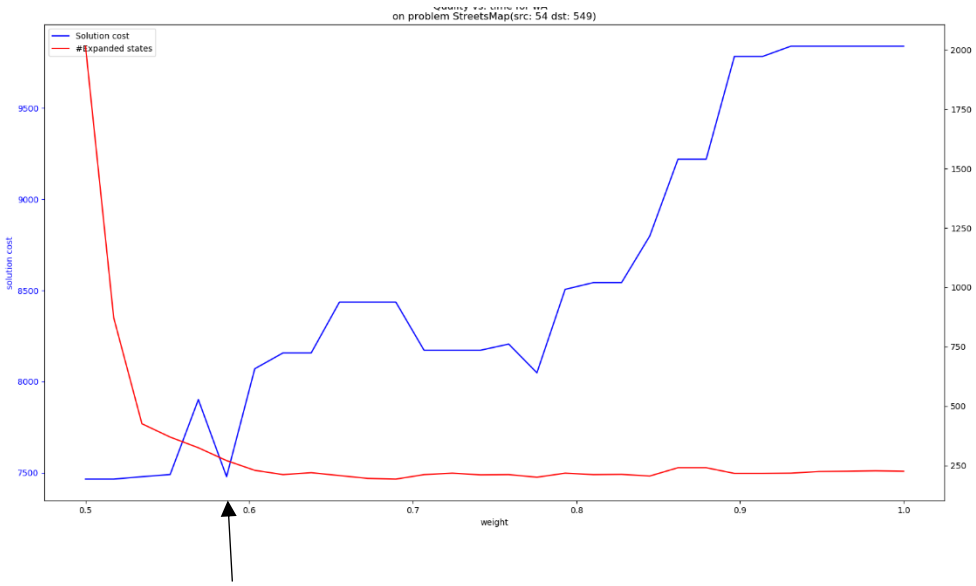
14. להלן גרף המתאר את עלות הפתרון (לפי מרחק) שמצא אלגוריתם A^* הממושקל ואת מספר הצמתים שפותחו בדרך לפתרון, כתלות במשקל עצמו.



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.



19. ההיוריסטיקה TruckDeliveriesMaxAirDistHeuristic קבילה, מפני שכפי שנראה מיד, לעולם לא תיתן הערכה יקרה יותר מהמחיר האמיתי. בהמשך השאלה נתייחס למחיר האמתי ולערך ההיוריסטיקה, כאשר המחיר האמתי הוא המרחק המינימלי שתעבור המשאית עד לסיום משימתה.

במידה והמשאית הגיעה לנקודה האחרונה (ההיוריסטיקה בודקת רק צומת אחד, המתאים מקום הנוכחי), המחיר עד ליעד (המחיר האמתי) יהיה שווה לערך שתיתן ההיוריסטיקה – אפס.

במידה ונשארו שתי נקודות בלבד (כולל המקום הנוכחי), המחיר האמתי יהיה לכל הפחות המרחק האווירי ביניהן. הערך שתיתן ההיוריסטיקה יהיה גם הוא המרחק האווירי ביניהן, כיוון שהן שתי הנקודות (צמתים) היחידות שניתנות לה לבדיקה.

במידה ונשארו למשאית יותר משתי צמתים ההיוריסטיקה תיתן ערך נמוך יותר מאשר המחיר האמתי (אלא אם כל הצמתים נמצאים בדיוק על קו ישר אחד, והמסלול בהם יהיה לכיוון אחד בלבד, ואז הערכים יהיו שווים). נניח שהערך שנתנה ההיוריסטיקה הוא המרחק האווירי בין צומת א' לצומת ב'. שני צמתים אלו הם צמתים שהמשאית חייבת לעבור בהם גם כך, כלומר, במסלול שלה מהצומת הנוכחי ועד לסיום יופיעו שני הצמתים, כאשר נאמר שצומת א' הוא זה שיופיע קודם. המשאית תצטרך לעבור מהצומת הנוכחי לצומת א', מצומת א' (אולי דרך צמתים אחרים) לצומת ב', ומצומת ב' לצומת הסיום.

כיוון שהמרחק הקצר ביותר בין שני צמתים יהיה המרחק האווירי (אי-שוויון המשולש), המרחק שתעשה המשאית בין הצמתים יהיה לכל הפחות הערך שתחזיר ההיוריסטיקה, כאשר יש להוסיף לכך גם המרחק שתעבור עד לצומת א', ואת המרחק מצומת ב' לצומת הסיום (שני מרחקים שיכולים להיות אפס במידה ויש התלכדות).

22. ההיוריסטיקה TruckDeliveriesSumAirDistHeuristic אינה קבילה.

נשתמש בדוגמה הנגדית הבאה:

כאשר הכוכבים מסמנים מיקומים של צמתים במפה, והמיקום הנוכחי מסומן ע"י הכוכב הכתום. הדוגמא בנויה כך שבחיפוש אחר הצומת הקרוב ביותר נלך כל פעם שמאלה, עד שנגיע לכוכב השמאלי ביותר, ואז נחזור את כל הדרך עד לכוכב הימני הקיצוני. סכום המרחקים בדרך זו יהיה הערך שתיתן ההיוריסטיקה.

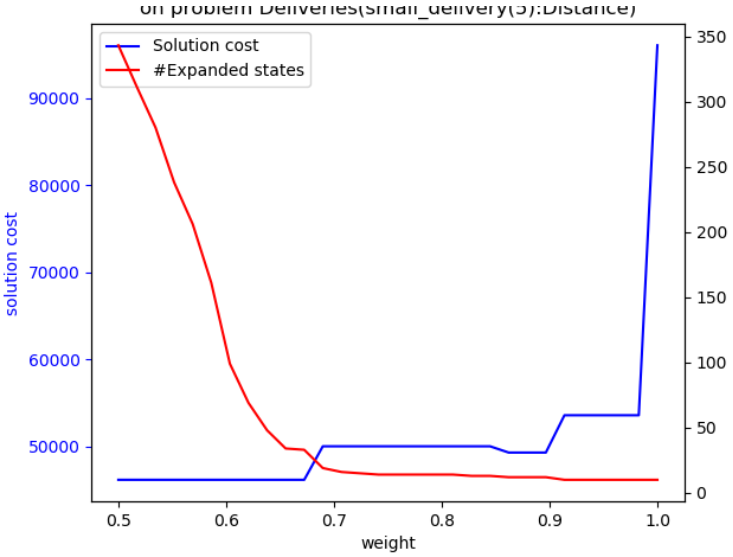
לעומת זאת מסלול שיתחיל ימינה (אל הכוכב הימני הקיצוני) ומשם ימשיך רק שמאלה קצר יותר (מהערך שתיתן ההיוריסטיקה). על כן, קיימת דוגמא נגדית בה ההיוריסטיקה נותנת מחיר גבוה יותר מאשר המחיר האמתי.

על כן, ההיוריסטיקה אינה קבילה.

25. ההיוריסטיקה TruckDeliveriesMSTAirDistHeuristic הינה היוריסטיקה קבילה.

המסלול הקצר ביותר, אשר הוא גם המחיר האמתי, יהיה קשיר, יכיל את כל הצמתים (שנשארו למשאית) ויהיה ללא מעגלים (בכל צומת מבקרים פעם אחת) – על כן הוא עץ פורש. לכן בוודאי שמשקלן הכולל של הקשתות (המרחק המינימלי שהמשאית יכולה לעבור) בו יהיה גדול או שווה למשקל הכולל של הקשתות בעץ פורש מינימלי.

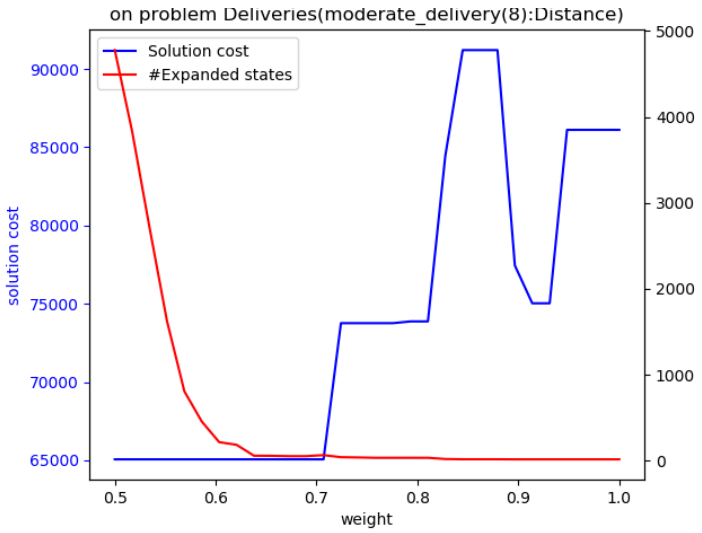
26. הגרף הראשון מתאר את מגמת מחיר הפתרון ומספר הפיתוחים (כתלות במשקל של האלגוריתם) עבור ההיוריסטיקה TruckDeliveriesMSTAirDistHeuristic.



ניתן לראות שהמגמות שתוארו בכלל האצבע בהרצאה, ובנוסף גם בתשובה לסעיף 14 מתקיימות גם כאן, כמו גם אי-היותו נכון באופן מוחלט. ניתן עוד לראות שההבדל הגדול במחיר הפתרון קרה כאשר נתנו את כל המשקל להיוריסטיקה (ללא התייחסות ל-g), קפיצה של עשרות אחוזים, כאשר לאורך כל הציר עד אותו שינוי, העלייה במחיר הפתרון היא כ-20%.

אם היינו צריכים לבחור משקל, היינו בוחרים בערך סביב 0.65. במשקל כזה איכות הפתרון היא עדיין אופטימלית (או קרובה מאוד לאופטימלית) ומספר המצבים שפותחו גם הוא קרוב לערכו האופטימלי (מבין כלל המשקלים).

הגרף השני מתאר את מגמת מחיר הפתרון ומספר הפיתוחים (כתלות במשקל של האלגוריתם) עבור ההיוריסטיקה TruckDeliveriesSumAirDistHeuristic.



בניגוד לגרף הקודם, מחיר הפתרון חווה ירידות גדולות (של עשרות אחוזים) עם הגדלה של המשקל (עבור משקלים בין 0.85 ל-0.9) כאשר הערך המקסימלי של מחיר פתרון מתקבל דווקא סביב 0.85. מספר המצבים המפותחים דווקא מציג מגמת ירידה לכל אורך הגרף. יש לזכור, שההיוריסטיקה בה השתמשנו כאן אינה קבילה. אף על פי כן, עבור משקלים נמוכים אנו מקבלים פתרון שהוא הפתרון האופטימלי (כזה שמצאנו גם עם היוריסטיקות קבילות).

אם היינו צריכים לבחור במשקל מסוים, היינו בוחרים במשקל סביב 0.63 שם המשקל שניתן להיוריסטיקה יחסית נמוך, מספר המצבים המפותחים נמוך משמעותית מערכו סביב חצי (וקרוב לערכו האופטימלי מבין כלל המשקלים) ומחיר הפתרון הוא הטוב ביותר.

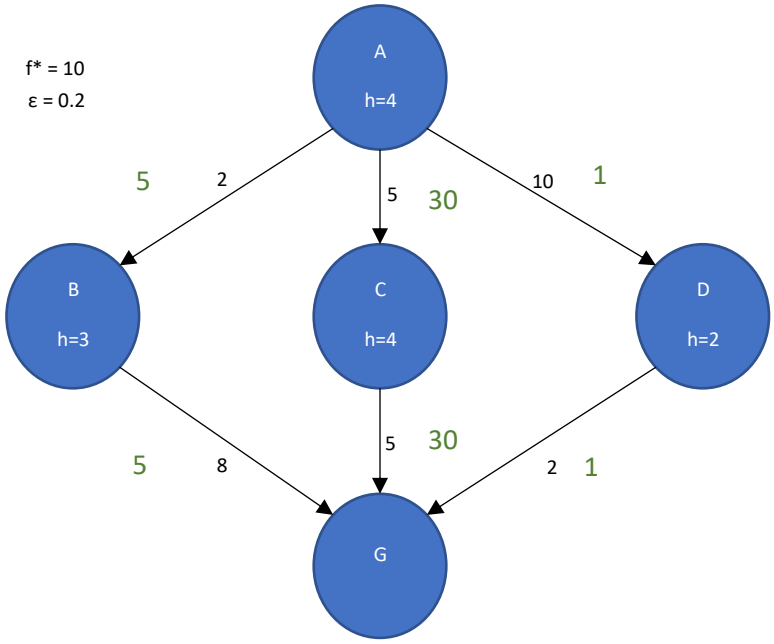
29. התוצאות שמתקבלות משתי הריצות הרלוונטיות:

Deliveries(small_delivery(5):Time) A* (h=TruckDeliveriesMSTAirDist, w=0.500) time: 52.96 #dev: 425 |space|: 562 total_g_cost: 30.18082 total_cost: DeliveryCost(dist= 46771.762 meter, time= 30.181 minutes, money= 151.961 nis)

Deliveries(small_delivery(5):Money) A* (h=TruckDeliveriesMSTAirDist, w=0.500) time: 81.44 #dev: 518 |space|: 591 total_g_cost: 104.23259 total_cost: DeliveryCost(dist= 46763.990 meter, time= 35.248 minutes, money= 104.233 nis)

ניתן להבחין שהזמן יוצא חמש דקות פחות אם מנסים למזער את הזמן, בהשוואה לתוצאה של ריצה בה מנסים למזער את עלות הכספית. ולהיפך, בריצה שבה מנסים למזער את עלות הכספית, עלות כספית של הפיתרון המתקבל היא שני שליש מתוצאה שמתקבלת אם מנסים למזער את הזמן.

30. הטענה של התלמיד אינה נכונה בהכרח, נציג דוגמה נגדית.



הגרף מציג שלושה מסלולים אפשריים ממצב התחלתי למצב מטרה, כאשר ההיוריסטיקה והמשקלים הם לפי כסף. המסלול הימני הוא היקר ביותר, פי 1.2 מהשניים האחרים. אולם, הזמן שלוקח לעבור אותו (מצוין בירוק במפה) יהיה המינימלי מבין השלושה, נבחר זמן של 2 דקות, דקה לכל קשת. הזמן שלוקח במסלול השמאלי יהיה עשר דקות, חמש דקות לכל קשת. הזמן שלוקח במסלול האמצעי הוא שעה, חצי שעה לכל קשת.

המסלול הימני הוא המהיר ביותר, ונמצא (תחת אפסילון ששווה ל-0.2, כפי שמצוין בגרף) בתוך קבוצת הפתרונות הכמעט אופטימליים מבחינת מחיר. לכן, הוא הפתרון שצריך להיבחר. נראה דרך מעבר על הגרף, שהפתרון שהציע עומר לא ייתן אותו, ובכך נפריך את טענתו.

נשים לב שההיוריסטיקה לפי הכסף בה השתמשנו היא קבילה (אף פעם לא פסימית).

בהתחלה יהיה הצומת A ב-open. נפתח את כל בניו ונכניס אותם ל-open. B יהיה עם f=5, C עם f=9 ו-D עם f=12. ה-f המינימלי שייך ל-B (שווה לחמש), ויהיה היחיד שייכנס ל-focal (שמוגבל על ידי $f = 5 * 1.2 = 6$), ולכן גם היחיד שיפותח.

בפיתוחו נעביר את B ל-close, ונוסיף את G ל-open עם f=10. במצב הנוכחי, ב-open יהיו C עם f=9, G עם f=10 וגם D עם f=12. ה-f המינימלי הוא 9 (של C) ולכן ההגבלה על focal תהיה $f = 9 * 1.2 = 10.8$, מה שיכניס גם את C וגם את G ל-focal. מבין שניהם הפתרון המהיר יותר הוא של G (עשר דקות) לעומת חצי שעה של C.

לכן G ייבחר להיות הצומת הבא לפיתוח, הוא צומת מטרה, ולכן יינתן כפתרון – כשהמסלול הוא המסלול השמאלי.

הפתרון שניתן לפי עומר נתן פתרון שאינו נכון (הפתרון הנכון הוא הימני) ולכן אינו נכון.

31. נפתור את הבעיה בשלבים. נרצה למצוא את כל הפתרונות שמצמצמים את הכסף, כך שעלותם לא גדולה מ-$(1+\varepsilon)f^*$. לאחר שנמצא את כולם, נחשב את המהיר ביניהם.

נדרש לעשות מספר שינויים:

ראשית, ברגע שנגיע לפתרון האופטימלי (הצומת הבא לפיתוח הוא, לראשונה, צומת מטרה) נעדכן שדה גלובלי *limit* להיות בעל ערך של $(1+\varepsilon)f^*$. נמשיך לפתח צמתים עד אשר הצומת הבא ב-*open* יהיה בעל f גדול מ- *limit*. עד שנמצא פתרון נאתחל את *limit* להיות אינסוף. במצב זה נוכל להגיד בוודאות שכל פתרון ששייך לקבוצה (לפי כסף) נבדק.

בנוסף נרצה לשמור פתרונות גם אם הם לא אופטימליים, במקרה של הצטלבות מסלולים. נאפשר יותר מהורה אחד לצומת. כאשר נגיע להצטלבות, נפעל כמו באלגוריתם הרגיל, רק שנוסיף להורים של הצומת את המסלול השני, כאשר נשמור גם את ההפרש במחיר וההפרש בזמנים מהפתרון הכי טוב עד אותה הצומת (הצטלבות).

כל פתרון שנמצא (מצב מטרה) נשמור גם בצד, באוסף פתרונות.

כאשר נסיים נעבור על כל באוסף הפתרונות מהסוף להתחלה ונמצא את כל המסלולים (דרך ההצטלבויות שנשמרו) בצורה רקורסיבית. בכל פעם כשנגיע לצומת שהייתה בו התלכדות של כמה מסלולים – הצטלבות, נבחר מסלולים מהמיטבי מבחינת הזמן להכי ארוך בזמן. בהתלכדות הזאת נוסיף לסכום מסוים את ההפרש ששמרנו בעת ההצטלבות (מופיע בפסקה מעל) – ואם נעבור את limit נפסיק לבדוק את המסלול הזה, נחזור מעלה ברקורסיה להצטלבות, ונבחר במסלול הכי מהיר הבא. אם ברקורסיה מצאנו פתרון שעונה על הדרישה של הכסף, אין טעם לבדוק פתרונות מהצטלבויות אחרות – כי הגישה שלנו הייתה לבחור כל פעם את הכי מהיר. באותו מעבר נמדוד גם את הזמן – ונשמור את המסלול המהיר ביותר שנמצא במשתנה גלובלי.

כך נוכל לעבור על כל הפתרונות בקבוצה, למצוא זמן עבור כל אחד – ולמצוא את המיטבי.

אולם יש לציין, שבכך פגענו באלגוריתם מבחינת זיכרון, מפני שנשמרים מסלולים חלופיים בהצטלבויות. בבעיה מלאה בהצטלבויות נשמור הרבה מסלולים חלופיים – שיביא לשימוש רב בזיכרון. מעבר לזה, גם ישנה פגיעה בזמן החישוב כי נצטרך לעבור על כל אותם מסלולים אפשריים כדי למצוא את הכי מהיר.

בנוסף, זמן הריצה גדל, כי אנחנו נחשב פתרונות עד הגבול limit, ולא עד שנמצא פתרון ראשון.

33. קיבלנו את התוצאה הבאה:

Solve the truck deliveries problem (moderate input, distance objective, using A*eps, use non-acceptable heuristic as focal heuristic).

Deliveries(moderate_delivery(8):Distance) A* (h=TruckDeliveriesMSTAirDist, w=0.500)
time: 90.47 #dev: 6376 |space|: 9045 total_g_cost: 65062.81195 total_cost:
DeliveryCost(dist= 65062.812 meter, time= 42.946 minutes, money= 210.589 nis)

Deliveries(moderate_delivery(8):Distance) A*eps (h=TruckDeliveriesMSTAirDist, w=0.500)
time: 49.74 #dev: 6339 |space|: 9025 total_g_cost: 65062.81195 total_cost:
DeliveryCost(dist= 65062.812 meter, time= 42.946 minutes, money= 210.589 nis)

כלומר חסכנו מעט בפיתוחים – פחות 37, מבלי לפגוע בפתרון.

ציפינו לחסוך במספר הפיתוחים כי השתמשנו ביוריסטיקה נוספת בבחירה מ-focal, שהיא ההיוריסטיקה Sum, מיודעת יותר מאשר לבחור רנדומלית בין האפסילון-קרובים עבור יוריסטיקה Mst. כאשר מתבצע שימוש בהיוריסטיקה מיודעת יותר, אנו מצפים להגיע לפתרון מהר יותר, ולפיכך תוך מספר פיתוחים קטן יותר.

35. התוצאות שקבלנו עבור Anytime A* הן:

Solve the truck deliveries problem (moderate input, only distance objective, Anytime-A*, .MSTAirDist heuristics)

Deliveries(moderate_delivery(8):Distance) Anytime-A* (h=TruckDeliveriesMSTAirDist, w=0.688) time: 37.60 #dev: 335 |space|: 200 total_g_cost: 65459.89155 total_cost: DeliveryCost(dist= 65459.892 meter, time= 43.581 minutes, money= 212.405 nis)

Solve the truck deliveries problem (big input, only distance objective, Anytime-A*, SumAirDist .& MSTAirDist heuristics)

Deliveries(big_delivery(15):Distance) Anytime-A* (h=TruckDeliveriesSumAirDist, w=0.844) time: 167.31 #dev: 2030 |space|: 1861 total_g_cost: 155572.84122 total_cost: DeliveryCost(dist= 155572.841 meter, time= 103.868 minutes, money= 508.621 nis) |path|: 31

Deliveries(big_delivery(15):Distance) Anytime-A* (h=TruckDeliveriesMSTAirDist, w=0.875) time: 42.43 #dev: 2401 |space|: 1869 total_g_cost: 147869.82115 total_cost: DeliveryCost(dist= 147869.821 meter, time= 98.608 minutes, money= 483.292 nis) |path|: 31

אז מה בעצם קיבלנו?

עבור היוריסטיקה MST, ו-moderate input, ניתן להשוות את התוצאות לתוצאות שקבלנו עבור A* בסעיף 33:

Deliveries(moderate_delivery(8):Distance) A* (h=TruckDeliveriesMSTAirDist, w=0.500) time: 90.47 #dev: 6376 |space|: 9045 total_g_cost: 65062.81195 total_cost: DeliveryCost(dist= 65062.812 meter, time= 42.946 minutes, money= 210.589 nis)

Deliveries(moderate_delivery(8):Distance) Anytime-A* (h=TruckDeliveriesMSTAirDist, w=0.688) time: 37.60 #dev: 335 |space|: 200 total_g_cost: 65459.89155 total_cost: DeliveryCost(dist= 65459.892 meter, time= 43.581 minutes, money= 212.405 nis)

נשים לב כי Anytime A* לקח פחות ממחצית זמן הריצה של A*, כאשר מספר הפיתוחים וגודל המקום קטנים בערך פי 20 מאשר של A* (כצפוי, כי הגבלנו את מספר הפיתוחים המקסימלי), כאשר הפיתרון שמצא Anytime A* הוא לא פחות טוב מהותית בהשוואה לזה של A* - ב-2 ש"ח יותר יקר (כאחוז), חצי דקה של זמן יותר (מעט מעל אחוז), 500 מטר ארוך יותר שזה זניח מבחינת סדרי גודל (פחות מאחוז). המסקנה מכך היא ש-Anytime A* צפוי להיות הרבה יותר מהיר מ-A* לבצע משמעותית פחות פיתוחים, כאשר יש מגבלה על מס' הפיתוחים, והחשוב - לא לסטות מהותית מפיתרון האופטימלי.

בנוסף, עבור ריצות שונות של Anytime A* על big input, בהשוואה עבור היוריסטיקות שונות כאשר אחת מהן קבילה (MST) ואחת לא (SUM), עבור היוריסטיקה קבילה מתקבל פיתרון יותר טוב בכל המדדים (אך לא שונה מהותית מזה של הירוסטיקה לא קבילה) אבל בזמן פי 4 יותר קצר. זה מדגים כמה חשוב שהיוריסטיקה תהיה קבילה.

לגבי השוואה לחלק א': עבור ריצות של Anytime A* על big input, נשים לב שמספר ההובלות הוא 15, ובמקרה הגרוע, עבור היוריסטיקה לא קבילה, הצלחנו למצוא פיתרון מספיק טוב תוך פחות מ-3 דקות (ועם היוריסטיקה קבילה – פחות מ-40 שניות!), שזהו תוצאה מדהימה בהשוואה לסעיף 2 שבו הדגמנו שעבור k=15 ומחשב שיכול לבדוק 2^{30} פתרונות בשניה, פיתרון הבעיה ב-brute-force היה לוקח לנו $239k(mln\ years)$!