דו"ח מסכם

תרגיל בית 1

יונתן בתן

302279138

Yonibettan@gmail.com

עמרי פרוינד

301695490

omrifro@gamil.com

חלק א'

סעיף 1:

מספר הפרמוטציות עבור $k = 1,2,...10$ ו- $l = 5$

K	ללא אילוץ דלק $k!$	עם אילוץ דלק $(k!)l^{k-1}$
1	1	1
2	2	10
3	6	150
4	24	3000
5	120	75000
6	720	2250000
7	5040	78750000
8	40320	3150000000
9	362880	141750000000
10	3628800	7087500000000

חלק ב'

אין תרגילים

חלק ג'

סעיף 2:

בגבול העליון יש לנו k לקוחות הממתינים למשלוח ו- l תחנות דלק אפשריות ולכן מקדם הסיעוף המקסימלי הינו $k + l$. בגבול התחתון כל ההזמנות סופקו וכבר ביקרנו בכל תחנות הדלק ולכן מקדם הסיעוף הינן 0.

סה"כ נקבל $max = k + l \ , \ min = 0$.

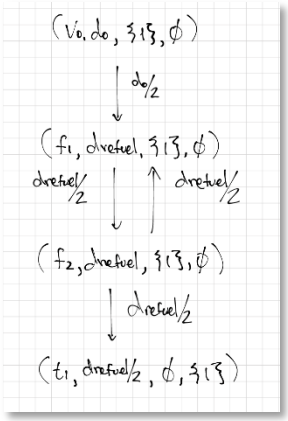
סעיף 3:

יתכנו מעגלים.

כיוון שהגעה לתחנת דלק אינה משנה את רשימות המשלוחים ומצב הדלק במצבים אלו הינו d_{refuel} כל עוד המרחק בין 2 תחנות דלק קטן מ- d_{refuel} יהיה קיים אופרטור מאחת לשנייה ונקבל מעגל.

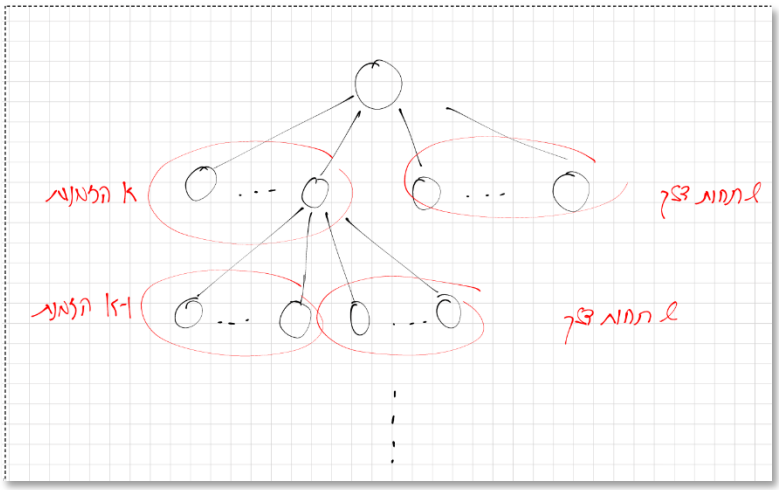
לדוגמא עבור $Ord = \{t1\}$, $GasStations = \{f1, f2\}$ וקבוצת המצבים

$S_d = \{(v_0, d_0, \{1\}, \emptyset), (f_1, d_{refuel}, \{1\}, \emptyset), (f_2, d_{refuel}, \{1\}, \emptyset), (t_1, d_{refuel}, \emptyset, \{1\})\}$ כאשר $dist(u, v)$ מסומן על הקשתות לכל u, v צמתים של מצבים מסוימים נקבל את המעגל הבא:



סעיף 4:

- מהמצב הראשון יש לנו $k + l$ יורשים אפשריים
- מהמצב השני יש לנו $(k - 1) + l$ יורשים אפשריים
- אנו נעצור כאשר לכל צומת יש רק l עלים
- סה"כ יש לנו $\frac{(k+l)!}{l!}$ מצבים



סעיף 5:

יתכנו בורות.

אם החלטנו ללכת לספק משלוח, שאינו המשלוח האחרון, במקום ללכת לתדלק והגענו למצב שאין לנו מספיק דלק כדי להגיע למשלוח הבא או לתחנת דלק קרובה נתקע בבור.

סעיף 6:

$$Succ_t\big((v_1,d_1,T_1,F_1)\big)=\left\{(t_i,d,T,F)\middle|\begin{matrix} t_i\in Ord \\ d_1-dist(v_1,t_i)\geq 0 \\ d=d_1-dist(v_1,t_i) \\ T=T_1\backslash\{i\} \\ F=F_1\cup\{i\}\end{matrix}\right\}$$

$$Succ_f\big((v_1,d_1,T_1,F_1)\big)=\left\{(f,d,T,F)\middle|\begin{matrix} f\in GasStations \\ d_1-dist(v_1,f)\geq 0 \\ d=d_{refuel} \\ T=T_1 \\ F=F_1\end{matrix}\right\}$$

$$Succ\big((v_1,d_1,T_1,F_1)\big)=Succ_t\big((v_1,d_1,T_1,F_1)\big)\cup Succ_f\big((v_1,d_1,T_1,F_1)\big)$$

סעיף 7:

במקרה המינימלי יהיה לנו דיי דלק בכדי לספק את כל המשלוחים ללא עצירה בתחנת דלק ולכן עבור הוספת צומת ההתחלה נקבל עומק מינימלי של k.

חלק ד'

סעיף 8:

- ניתן לראות את הפתרון לבעיית המפה עבור אלגוריתם חיפוש UniformCost עבור משקלי קשתות 1 ולכן האלגוריתם קביל ואנו מקבלים פתרון אופטימלי.

```
(ai_hw1) ybettan hw1 (devel) $ python main.py
load_map_from_csv: 1.87sec

Solve the map problem.
Map(src: 54 dst: 549)                UniformCost                    time:   0.01   #dev: 212     total_cost:    12.00
000   |path|: 13    path: [   54,    55,    56,    57,    58,    59,    60, 28893, 14580, 14590, 14591, 14592, 14593]
```

- עבור אותה בעיה כעת משקלי הקשתות אינם זהים ומייצגים את המרחק האווירי בין צמתים שונים במפה לכן מן הסתם שנקבל פתרון שונה שהרי המפה שונה.

```
(ai_hw1) ybettan hw1 (devel) $ python main.py
load_map_from_csv: 1.60sec

Solve the map problem.
Map(src: 54 dst: 549)                UniformCost                    time:   0.64   #dev: 17355   total_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, 24
841, 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, 8220
9, 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]
```

חלק ה'

סעיף 9:

בקוד

סעיף 10:

כפי שניתן לראות אכן קיבלנו את אותה תוצאה עבור uniform_cost ו- a_star עם יוריסטיקה $h = 0$.

```
(ai_hw1) ybettan hw1 (devel) $ python main.py
load_map_from_csv: 1.66sec

Solve the map problem.
Map(src: 54 dst: 549)                UniformCost                time:   0.59   #dev: 17355   total_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, 24
841, 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, 8220
9, 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]
Map(src: 54 dst: 549)                A* (h=0, w=0.500)          time:   0.71   #dev: 17355   total_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, 24
841, 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, 8220
9, 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]
```

סעיף 11:

כמתוכנן כיוון שהיוריסטיקה קבילה קיבלנו פתרון קביל (זהה ל- uniform cost) אך יעיל יותר מפתרון עם יוריסטיקה "טיפשה" שאינה מוסיפה מידע לאלגוריתם מעבר למחיר הצומת הנוכחי.

```
Map(src: 54 dst: 549)                A* (h=AirDist, w=0.500)       time:   0.11   #dev: 2016    total_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, 24
841, 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, 8220
9, 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]
```

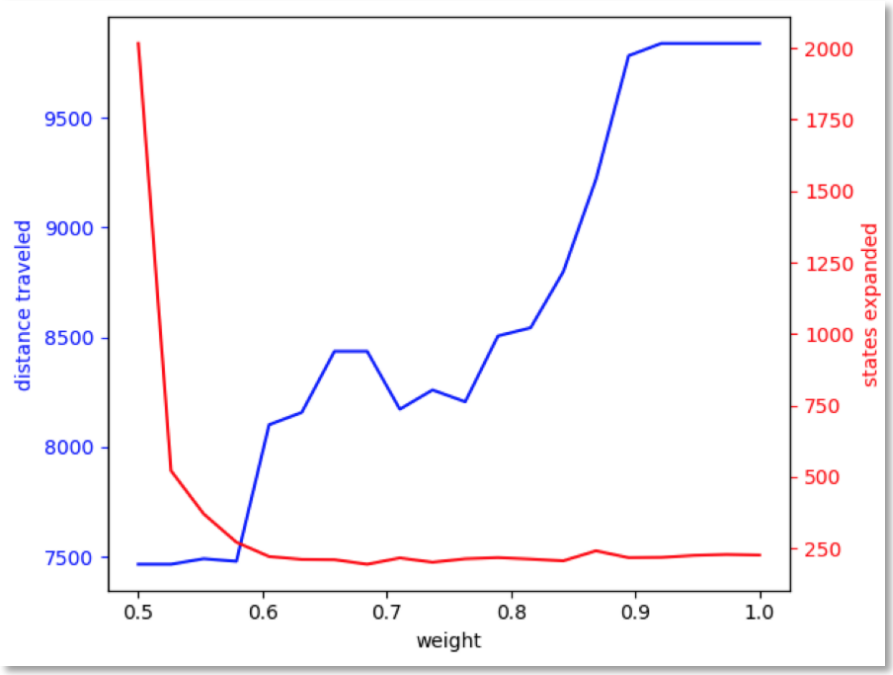
סעיף 12:

כפי שלמדנו ככל שליורסטיקה משקל כבד יותר האלגוריתם רץ כמה שיותר מהר למטרה על חשבון אופטימליות הפתרון וככל שמשקל היוריסטיקה נמוך יותר כך אנחנו משפרים את אופטימליות הפתרון על חשבון זמן הריצה ולכן הגרף שקיבלנו מסתדר עם התיאוריה.

עקומת המרחק אינה מונוטונית לחלוטין שכן היוריסטיקה אינה מושלמת וניתן לתכנן גרף באופן כזה שלא יהיה פגיעה במחיר הפתרון ולכן כפי שניתן לראות המגמה מראה פגיעה באופטימליות הפתרון עבור הגדלת משקל היוריסטיקה אך לא באופן "חלק"

נקודה מעניינת היא שהקטנה המשקל הניתן ליורסטיקה בקנה מידה קטנה פוגע בצורה משמעותית באופטימליות הפתרון.

במקרה הנ"ל $weight \approx 0.58$ היה נותן פתרון אופטימלי כמעט באותה מידה אך בעלות זולה הרבה יותר.



חלק ו'

סעיף 13:

בקוד

סעיף 14:

היוריסטיקה $h(s) = \max_{i \in T} AirDist(s.v, t_i)$ הינה קבילה.

במידה ויש הזמנה יחידה j אז נקבל $h(s) = AirDist(s.v, t_j) \leq AirDist(s.v, t_j)$ אחרת יש יותר מהזמנה יחידה שעדיין לא סופקה אז בהכרח סכום המרחקים האוויריים שלהם בהכרח גדול מהמרחק האווירי המקסימלי מבין כל אחת בנפרד.

סעיף 15:

בקוד

סעיף 16:

קיבלנו מסלול חוקי באורך 11 צמתים העובר ב- 2 תחנות דלק

```
Solve the relaxed deliveries problem.
RelaxedDeliveries(big_delivery)       A* (h=MaxAirDist, w=0.500)    time:   4.50   #dev: 3908    total_cost: 40844.21165   |path|: 11    path: [33919, 18409, 77726, 26690, 31221, 63050, 84034, 60664, 70557
, 94941, 31008]   gas-stations: [31221, 70557]
```

סעיף 17:

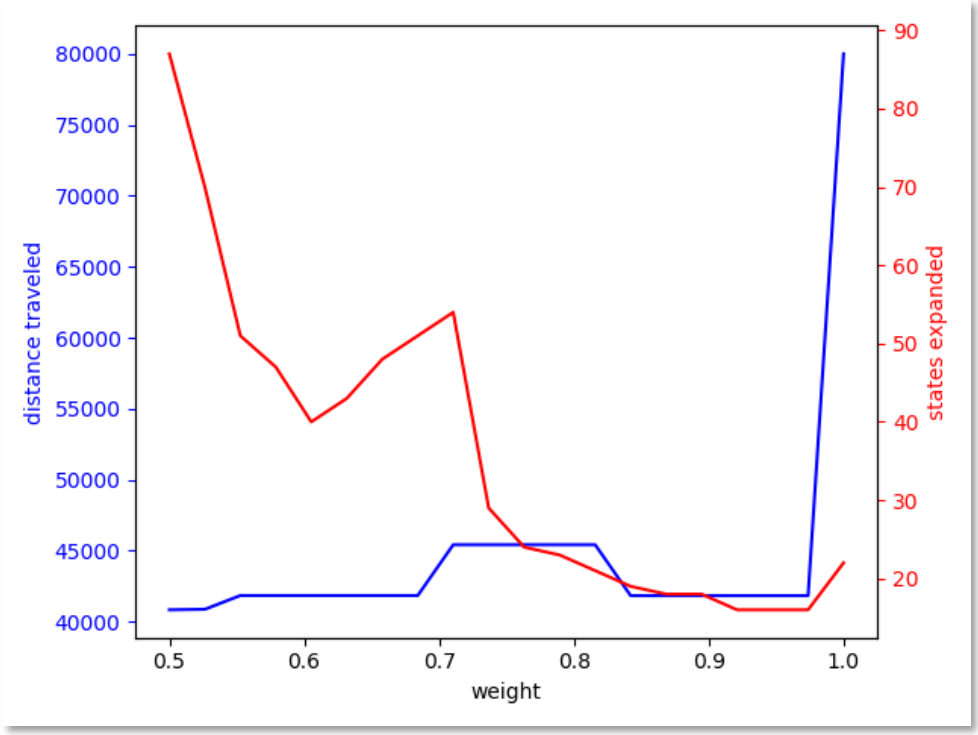
כפי שמוסבר בתרגיל למרות שהיוריסטיקה אינה מאוד מיודעת היא עדיין חוסכת לנו המון פיתוחים, וכמובן עדיין מחזירה את הפתרון האופטימלי כיוון שיוריסטיקה זו קבילה.

```
Solve the relaxed deliveries problem.
RelaxedDeliveries(big_delivery)       A* (h=MaxAirDist, w=0.500)    time:   5.01   #dev: 3908    total_cost: 40844.21165   |path|: 11    path: [33919, 18409, 77726, 26690, 31221, 63050, 84034, 60664, 70557
, 94941, 31008]   gas-stations: [31221, 70557]
RelaxedDeliveries(big_delivery)       A* (h=MSTAirDist, w=0.500)    time:   1.50   #dev: 87      total_cost: 40844.21165   |path|: 11    path: [33919, 18409, 77726, 26690, 31221, 63050, 84034, 60664, 70557
, 94941, 31008]   gas-stations: [31221, 70557]
```

סעיף 18:

אמנם הבעיה אותה אנחנו פותרים הינה שונה מבעיית המפה אך עדיין נוכל לקבל תחושה כללית אינדיקטיבית. ניתן לראות שאופטימליות הפתרון יחסית חסין מפני הגדלת המשקל של היוריסטיקה (עד גבול מסוים) שכן יוריסטיקה זו מיודעת יותר מהיוריסטיקה של מרחק אווירי (הכוונה היא למרחק אווירי ישיר למרחק מטרה ולא דרך סכום המרחקים האווירים של צמתי הביניים).

בנוסף ככל שנגדיל את המשקל הניתן לערך היוריסטי (עד גבול מסוים) לעומת מחיר הצומת כך אנחנו בעצם "רצים" מהר יותר לכיוון המטרה עם פגיעה מינימלית באופטימליות הפתרון כפי שהוסבר קודם לכן.



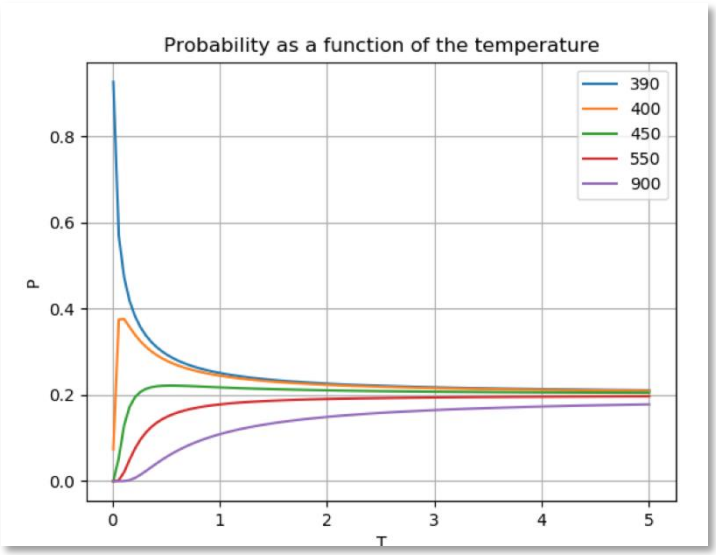
חלק ז'

סעיף 19:

$$\forall x_i \in x^t, \Pr_{normalized}(x) = \frac{\left(\frac{x_i}{\alpha}\right)^{-\frac{1}{T}}}{\sum_{pnt_h \in best\ N\ points}\left(\frac{x_h}{\alpha}\right)^{-\frac{1}{T}}} = \frac{\cancel{\left(\frac{1}{\alpha}\right)^{\frac{1}{T}}}(x_i)^{-\frac{1}{T}}}{\cancel{\left(\frac{1}{\alpha}\right)^{\frac{1}{T}}}\sum_{pnt_h \in best\ N\ points}(x_h)^{-\frac{1}{T}}} = \Pr_{not-normalized}(x)$$

סעיף 20:

להלן הגרף שהתקבל, הסבר יינתן בסעיפים הבאים.



סעיף 21:

כאשר הטמפרטורה אפסית אנחנו בעצם לא מאפשרים "רעש" כלומר לא נותנים לאלגוריתם הרבה חופש בחירה לבחור בצומת עם ערך יוריסטי שאינו המינימלי כלומר אינו מעניק "סקרנות" לאלגוריתם.

ניתן לראות שאכן האלגוריתם יבחר את הצומת הבא לפיתוח כאל הצומת עם היוריסטיקה הנמוכה ביותר בהסתברות של כמעט 1 וכל צומת אחר בהסתברות כמעט 0, במילים אחרות יתנהג בצורה דומה לאלגוריתם דטרמיניסטי.

סעיף 22:

עבור טמפרטורה מקסימלית האלגוריתם בוחר צומת כמעט באקראיות מוחלטת מבין N הצמתים בעלי ערך יוריסטי נמוך ביותר כלומר במקרה זה לאלגוריתם "מקסימום סטוכסטיות".

כלומר פרמטר זה בעצם מאפשר לנו לבצע tuning על מידת "ההרפתקנות" של האלגוריתם אל מול הדטרמיניסטיות שלו.

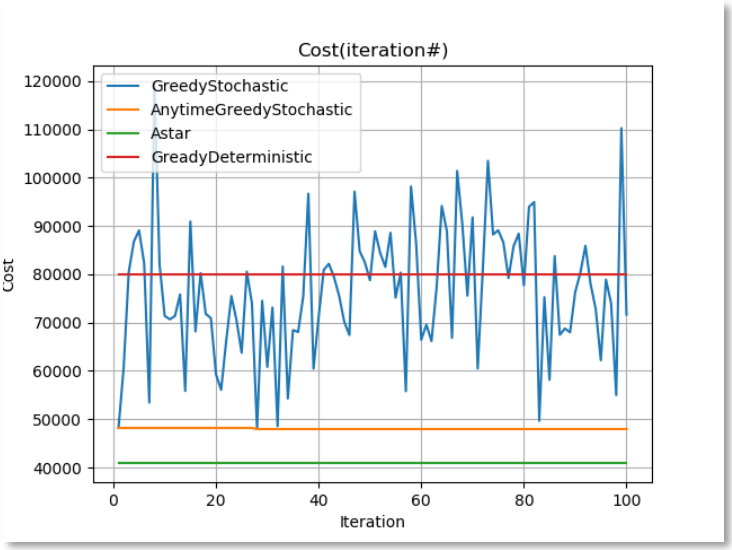
סעיף 23:

בקוד

סעיף 24:

להלן התוצאות שקיבלנו, כיוון שבמקרה הריצה של האלגוריתם הסטוכסטי נתנה פתרון מוצלח כבר באיטראציה הראשונה קשה להבחין בגרף המונוטוני יורד של any time greedy stochastic ולכן החלטנו לצרף את הגרף שהתקבלת מהמחשב שלי (שאר הגרפים הוצאו מהמחשב של השותף שלי).

כפי שציינו במייל אשר שלחנו לסגל מבנה הנתונים heapdict אשר מימשתם לצורך מימוש OPEN אינו דטרמניסטי עבור צמתים בעלי ערך מיון זהה ולכן הגרפים של ה- greedy deterministic אינם זהים ב-2 המחשבים אך בכל זאת ניתן הסבר לגרף השני כיוון שהוא ברור יותר לצורף הסבר ההתנהגות.

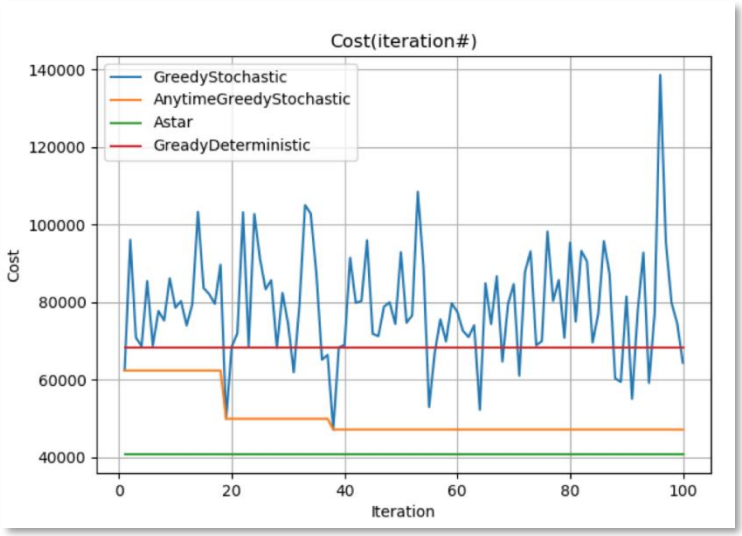


להלן הגרף שהתקבל מהמחשב שלי, נתחיל בברור מאליו A* כידוע נותן פתרון אופטימלי עבור היוריסטיקה קבילה ולכן כצפוי לא התקבל פתרון טוב ממנו.

האלגוריתם gready deterministic אינו נותן פתרון קביל וזה גם בהתאם לציפיות שגם אלגוריתם חמדן אינו בהכרח מספק פתרון אופטימלי.

האלגוריתם greedy stochastic מאפשר בעצם יותר "הרפתקנות" מאשר הגרסה הדטרמיניסטית שלו ודבר זה גורם לעיטים למציאת פתרון טוב יותר ולעיטים פתרון טוב פחות אשר עולה בקנה אחד עם התיאוריה.

לבסוף האלגוריתם any time greedy stochastic "דוגם" את הפתרון הטוב ביותר שהתקבל עד כה באמצעות greedy stochastic ולכן מסתדר לנו שהתקבל גרף מונוטוני יורד בצורת "מדרגה"



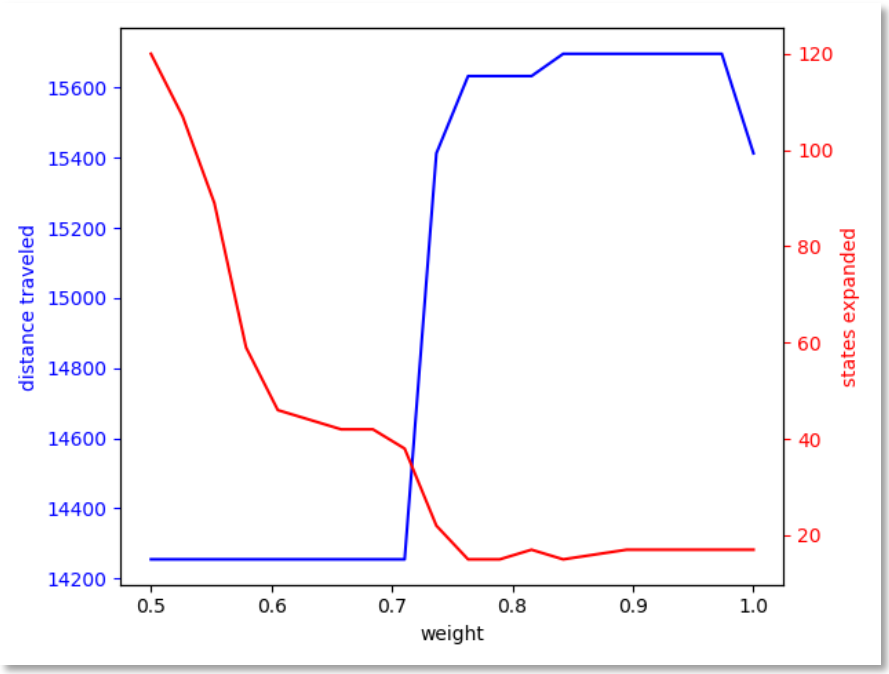
חלק ח'

סעיף 25

בקוד

סעיף 26

ההסבר שקול לסעיף 12, עבור משקל כבד יותר להיוריסטיקה, האלגוריתם רץ מהר יותר על חשבון אופטימליות הפתרון וככל שמשקל ההיוריסטיקה נמוך יותר כך אנחנו משפרים את אופטימליות הפתרון על חשבון זמן הריצה. הגרף שהתקבל תואם איכותית לסעיף 12 וכמובן מתיישב עם התיאוריה.
ניתן לראות שעד משקל של כ-0.71 מספר המצבים שפותחו יורד (שיפור בזמן הריצה) אך ללא פגיעה באופטימליות הפתרון (המרחק הכולל) עבור הבעיה הנ"ל. החל מנקודה זו הגדלת משקל ההיוריסטיקה פוגע משמעותית באופטימאליות הפתרון.



סעיף 27

לכל מצב s במרחב המצבים של StrictDeliveryProblem נבחר את h(s) להיות מחיר הצומת הסופי בהפעלת מימוש RelaxedDelivaryProblem עם היוריסטיקת MSTAirDist על אותה הבעיה. אם אין פתרון (למשל אין מספיק דלק) נחזיר h = ∞.
בהפעלת StrictDeliveryProblem נבצע את השינויים הבאים: המצב ההתחלתי הוא s, כמות הדלק ההתחלתית היא הדלק הנותר ב-s ורשימת ההזמנות היא הרשימה שנותרה ב-s.

היוריסטיקה זו קבילה משום שראינו ש-RelaxedDelivaryProblem עם היוריסטיקת MSTAirDist היא קבילה – כלומר, הפעלתה מצומת מסוים יחזיר לנו מסלול אופטימלי למצב המטרה.

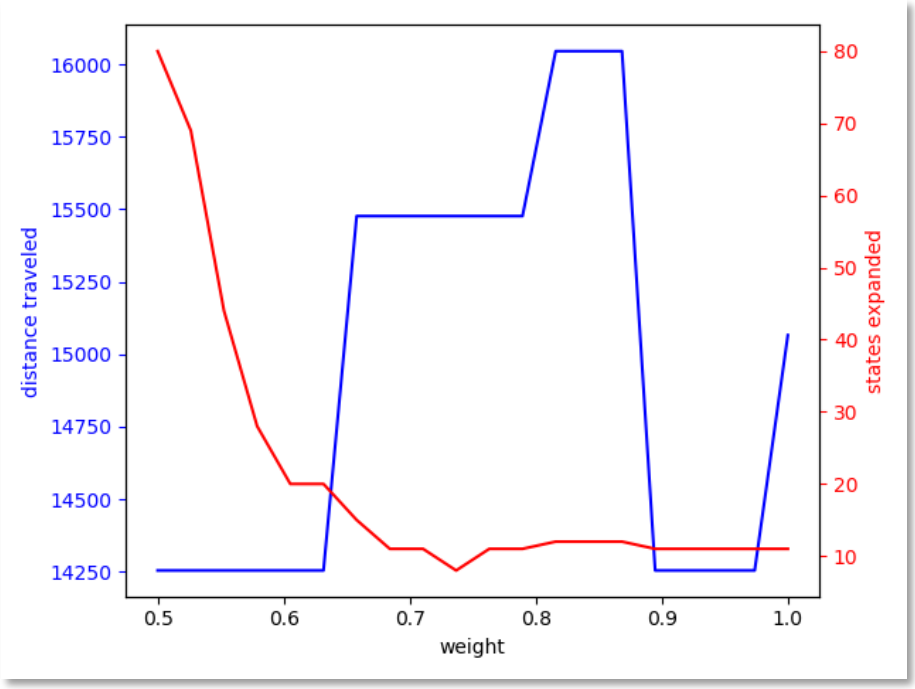
בעצם מובטח לנו שההערכה היוריסטית שלנו תחזיר עלות מדויקת כלומר קיבלנו את היוריסטיקה המושלמת h^* ומכיוון שלכל s מתקיים $0 \leq h(s) = h^*(s) \leq h^*(s)$ אזי קיבלנו יוריסטיקה קבילה, שאגב היא המיודעת ביותר אשר עדיין קבילה.

סעיף 28

בהשוואה לתוצאות מסעיף 26, ראשית נשים לב כי עבור משקל w=0.5 (מימוש אלגו' A* המקורי) שני האלגוריתמים מקבלים את אותו המחיר האופטימלי. תוצאה זו מחזקת את הטענה שלנו מסעיף 27 על פיה ההיוריסטיקה קבילה שכן אם היינו מקבלים מסלול ארוך יותר זה היה מראה שההיוריסטיקה לא קבילה. שנית, ניתן לראות שעם ההיוריסטיקה הנוכחית פיתחנו רק $80 \sim$ צמתים לעומת $120 \sim$ עם ההיוריסטיקה של סעיף 26 שכן זוהי היוריסטיקה מושלמת אשר מיודעת יותר מהיוריסטיקה של סעיף 26 ולכן "עושה פחות טעויות" מיוריסטיקה פחות מיודעת דבר הגורם לפיתוח מספר קטן של צמתים (פחות תיקונים של מסלולים אם בכלל).

לא היה משקל w עבורו מספר הפיתוחים בסעיף 26 היה נמוך ממספר הפיתוחים בסעיף 28

זמן הריצה השתפר. כידוע זמן הריצה מושפע ישירות ממספר הצמתים שפותחו. בסעיף 28, לכל משקל מספר הפיתוחים קטן משמעותית ולכן זמן הריצה השתפר כיוון שהיוריסטיקה מיודעת יותר.



שאלה תיאורטית

א.

נתון כי h קבילה, מכאן שעבור כל מצב s בו h מוגדרת $h_0(s) = h(s)$ ולכן $h_0(s) \le h^*(s)$, עבור מצבים s בהם h אינה מוגדרת $h_0(s) = 0 \le h^*$ לכן $0 \le h_0(s) \le h^*$ לכל s ולכן קבילה.

ב.

נגדיר את היורסטיקה הבאה:

$$h(s)=\begin{cases} h(s) & Applicable_h(s)\ is\ True \\ h(s.father)-operator\_cost(s.father,s) & s\ is\ not\ root \wedge Applicable_h(s.father)\ is\ True \\ 0 & else \end{cases}$$

ולמעשה נריץ עם יוריסטיקה זו A* ונקבל את התוצאה הרצויה.

ג.

לכל צומת חדש המתגלה במרחב החיפוש יש אב שגילה אותו, נסמן אב זה ב- $s.father$ ונשתמש בדיוק באותה יוריסטיקה מהסעיף הקודם.
במידה ונגיע לצומת שכבר נמצא ב- OPEN או ב- CLOSE בעלות זולה יותר נעדכן את הערך היורסטי בהתאם לאב החדש.

ד.

קיים אלגוריתם כנ"ל.
נגדיר את היוריסטיקה הבאה:

$$h(s)=\begin{cases} h'(s) & s\ is\ root \\ h(s.father)-operator\_cost(s.father,s) & else \end{cases}$$

נריץ A* בעזרת היוריסטיקה h.

נוכיח:

1) היוריסטיקה קבילה ולכן האלגוריתם קביל
2) h מיודעת יותר מאשר $h_0(h',s)$
3) האלגוריתם יעיל לפחות כמו A* המשתמש ב- $h_0(h',s)$

.1

לכל צומת s נוכיח באינדוקציה על מספר הקשתות מהשורש ל- s.

בסיס:

עבור 0 קשתות מתקיים $s = root$ ולכן $h(root) = h'(root) = h^*(root) \leq h^*(root)$

צעד:

נניח ש- $h(t) \leq h^*(t)$ עבור צומת הנמצא במרחק k קשתות ונוכיח ש- $h(s) \leq h^*(s)$ עבור צומת הנמצא במרחק $k+1$ קשתות.

$$h(s) = h(s.father) - operatorCost(s.fahter, s) \leq^* h^*(s.father) - operatorCost(s.father, s) = h^*(s)$$

כאשר המעבר * נובע מהנחת האינדוקציה.

.2

עבור $s = root$ מתקיים $h(s) = h_0(h', s) \geq h_0(h', s)$

עבור $s \neq root$ נקבל $h(s) = h(s.father) - operatorCost(s.father, s) > 0 = h_0(h', s)$

ולכן h הינה מיודעת יותר מ- $h_0(h', s)$ לכל s

.3

לפי משפט מההרצאה אלגוריתם הרץ עם יוריסטיקה מיודעת יותר יעיל לפחות כמו אלגוריתם הרץ עם יוריסטיקה פחות מיודעת ולכן A* עם h יעיל לפחות כמו A* הרץ עם $h_0(h', s)$