

מבוא לבינה מלאכותית – תרגיל 1

מגשים:

גילי קרני 305200420

אפיק פריזברג 302575287

פרק ראשון

חלק א

1.

k	$\frac{(2k)!}{2^k}$
1	1
2	1
3	6
4	90
5	2520
6	113400
7	7484400
8	681080400
9	81729648000
10	125046361

חלק ב

2. מרחב הסיעוף הוא מספר הצמתים הישיגים מהצומת שיש בהם שינוי במצב הרשימות, כלומר, יש שם תחנת עליה או ירידה. לכן, המספר המינימלי יהיה 1, כאשר יש רק אפשרות אחת לשינוי מצב ההזמנות הנוכחי. המספר המקסימלי הוא k , לא יתכן כי יותר מ- k הזמנות יעברו מרשימה כלשהי לאחרת ולכל הזמנה יש רק רשימה אחת לכל היותר שהיא יכולה לעבור אליה ולכן לא ייתכנו יותר מ- k .

3. לא יתכנו מעגלים, מהגדרת האופרטור, בכל מעבר במצב ישתנו לפחות שתיים מהרשימות, נשים לב, שהזמנה יכולה לעבור רק מ- W ל- B , או מ- B ל- F ולכן לא ניתן לחזור לאותו מצב פעמיים.

4. מצבי המטרה הם מצבים בהם סיימנו לשרת את כל ההזמנות. כלומר, כל ההזמנות עברו לרשימת ההזמנות הגמורות. הניסוח הפורמלי:

$$G = \{(v, \emptyset, \emptyset, Ord) | v \in V, \exists i, ti = v\}$$

5. אם אין שתי הזמנות שמסתיימות באותו הצומת, מספר מצבי המטרה הוא כמספר צמתי הסיום של ההזמנות. כלומר, מספר צמתי המטרה הוא כמספר ההזמנות - k . (ניתן להגיד שיש $|V|$ מצבי מטרה אבל לא כולן ישיגים מהגדרת האופרטור)

6. לפי הגדרת התרגיל, הגרף קשיר, לכן, מכל מצב שאינו מצב מטרה תמיד אפשר להעלות מישהו על האוטובוס או להוריד מישהו מהאוטובוס ובכך נעבור למצב אחר. כלומר, תמיד אפשר להתקדם ולכן אין בורות ישיגים.

$$7. Succ((v1, W1, B1, F1)) = \{(v2, W2, B2, F2) | W2 = W1 / \{i | i \in W1 \wedge si = v2\} B2 = B1 / \{i | i \in B1 \wedge ti = v2\} \cup \{i | i \in W1 \wedge si = v2\}, F2 = F1 \cup \{i | i \in B1 \wedge ti = v2\}, v2 \in V\}$$

8. העומק המינימלי שנוכל למצוא הוא אם בכל מצב שנעבור מלבד הראשון והאחרון יעלה מישהו על האוטובוס וירד מישהו מהאוטובוס, במצב הראשון, רק יעלה ובאחרון רק ירד. במצב כזה, העומק הוא k+1. העומק המקסימלי שנוכל למצוא הוא אם בכל מצב עולה רק הזמנה אחת לאוטובוס או יורדת הזמנה אחת מהאוטובוס. כל הזמנה עולה פעם אחת ויורדת פעם אחת ולכן העומק המקסימלי הוא 2k.

9. הקוד טוען את נתוני הבעיה מקובץ CSV, לאחר מכן לוקחים הזמנה אחת ומדפיסים את פרטי הבעיה ואת החסם התחתון שהוא המרחק האווירי בין 2 הנקודות בהזמנה.

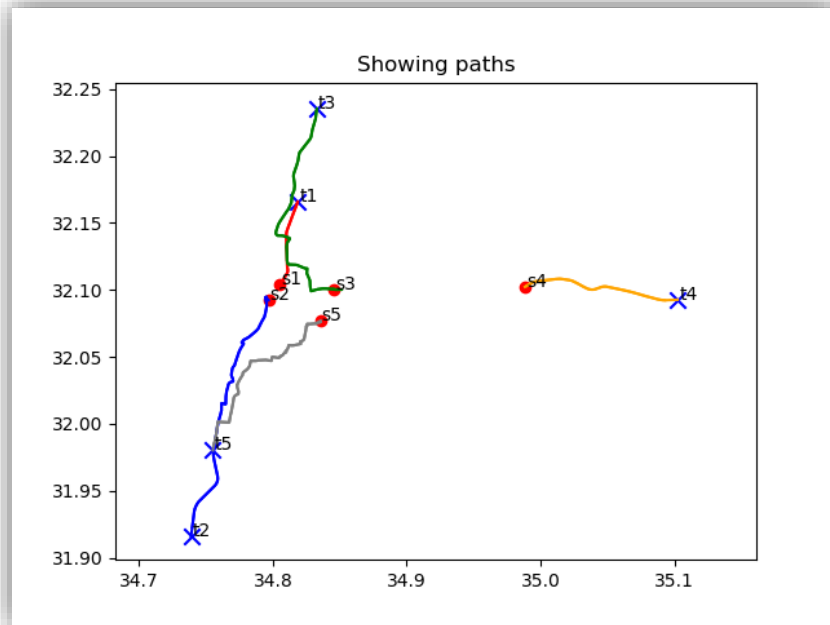
```
One of the orders is from junction #23695 at (32.1022894, 34.9882995) to #33320 at (32.0926573, 35.1022635)
A lower bound on the distance we need to drive for this order is: 10.78km
Path length: 161.11km
```

לאחר מכן מכניסים מסלול לדוגמא, מציירים גרף של ההזמנות ושל המסלול לדוגמא ואז מדפיסים את מרחק המסלול.
10. רשימת המצבים שהתקבל

```
Junction idx: #851288
waiting for bus: [(23695, 33320)]
orders on bus: [(32056, 834603), (851288, 533396)]
finished orders: [(47521, 606430), (466524, 29249)]
Is goal? False
```

11. התוצאה שהתקבלה:

```
Shortest distance for order from #32056 to #834603: 7.15km
Shortest distance for order from #47521 to #606430: 22.34km
Shortest distance for order from #466524 to #29249: 20.30km
Shortest distance for order from #23695 to #33320: 11.28km
Shortest distance for order from #851288 to #533396: 16.28km
Total distance: 77.35km
```



12. לא ניתן לקבוע אם סכום המסלולים הינו חסם עליון או תחתון למחיר מינימלי לבעיית האוטובוס על אותן ההזמנות. נראה שתי דוגמאות נגדיות:
דוגמה נגדית בה סכום המסלולים הוא ארוך יותר מהמחיר האופטימלי:



נניח שבגרף זה המחיר של מעבר בין שני צמתים הוא 1. סכום המסלולים הוא 5, למרות שמחיר המסלול האופטימלי הוא 4.
דוגמה נגדית בה סכום המסלולים הוא קצר יותר מהמחיר האופטימלי:



נניח שבגרף זה המחיר של מעבר בין שני צמתים הוא 1. סכום המסלולים הוא 1, למרות שמחיר המסלול האופטימלי הוא 2.
13. הטבלה שהתקבלה היא:

Input file	Solution cost (total distance in KM)
TLV_5	135.06 km
SDEROT_50	539.59 km
HAIFA_100	565.68 km
BEER_SHEVA_100	1329.21 km

14. בסעיף 6 הראנו שאין בורות בגרף (מלבד מצבי מטרה) ולכן בהכרח נגיע למצב מטרה כלשהו. כיוון שהאלגוריתם חמדן, נפתח רק מסלול יחיד. בסעיף 8 הראנו שעומק הגרף d חסום

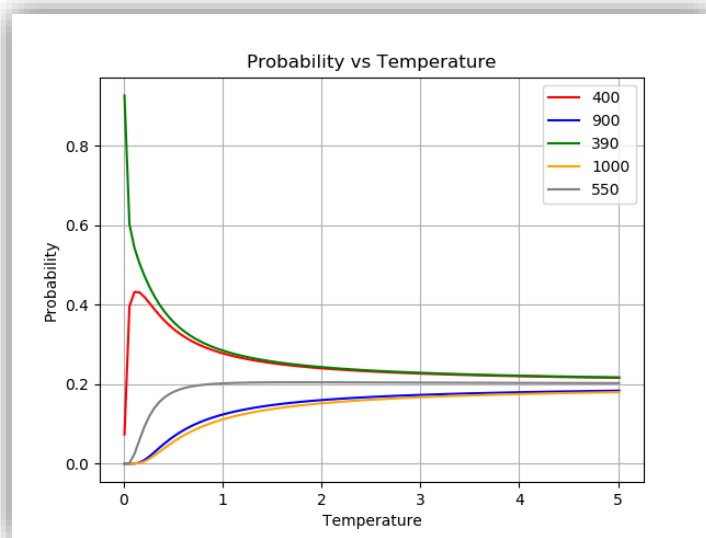
$k + 1 \leq d \leq k$ ולכן מספר המצבים שיפותחו חסום כמו אורך המסלול. החסם התחתון הוא $k+1$ והחסם העליון הוא $2k$.

15. לפי החסמים שהראנו סיבוכיות המקום היא $O(k)$.

16. עבור $pnt_i \notin best\ N\ points$ הערך לא משתנה. עבור $pnt_i \in best\ N\ points$ מתקיים:

$$\begin{aligned} \forall pnt_i \in pending: \Pr(pnt_i) &= \frac{(x_i)^{-\frac{1}{T}}}{\sum_{pnt_h \in best\ N\ points} (x_h)^{-\frac{1}{T}}} = \frac{\frac{1}{\alpha^{-\frac{1}{T}}}}{\frac{1}{\alpha^{-\frac{1}{T}}} \sum_{pnt_h \in best\ N\ points} (x_h)^{-\frac{1}{T}}} \\ &= \frac{\left(\frac{x_i}{\alpha}\right)^{-\frac{1}{T}}}{\sum_{pnt_h \in best\ N\ points} \left(\frac{x_h}{\alpha}\right)^{-\frac{1}{T}}} \end{aligned}$$

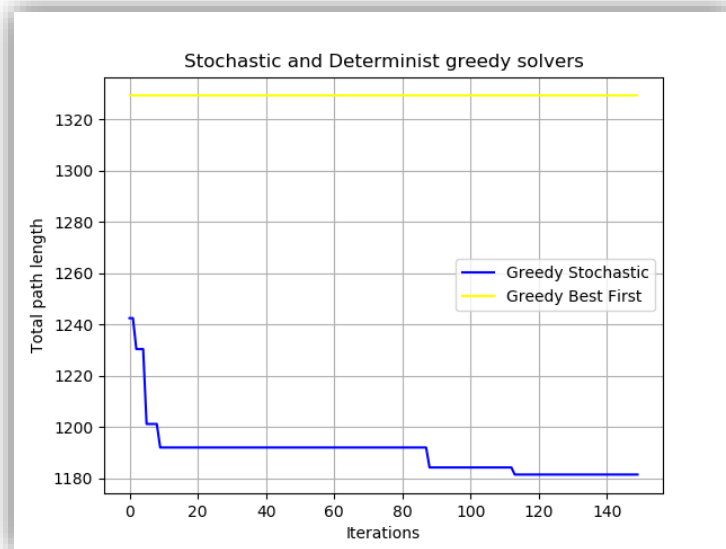
17. הגרף שהתקבל הוא:



18. עבור $T \rightarrow 0$ משקל ההבדל בין ערכי x גדל ולכן בוודאות (בהסתברות 1) נבחר את ערך x הקטן ביותר.

19. עבור $T \rightarrow \infty$ משקל ההבדל בין ערכי x קטן - המעריך של החזקה מתאפס - ולכן נקבל שההסתברות עבור כל אחד מהערכים היא $\frac{1}{N}$ ובמקרה שלנו $\frac{1}{5}$.

20. הגרף שהתקבל הוא:



21. התוצאות שהתקבלו הן:

```
Average of x_i is 1.301009e+03
Standart deviation of x_i is 5.888196e+01
p-value is 3.067575e-08
```

לפי מה שלמדנו בתרגול הראשון, נדחה את השערת האפס אם $p\text{-value} < 0.01$. במקרה שלנו התקבל $p\text{-value}$ קטן בהרבה ולכן השארת האפס באמת נדחית. ניתן לראות בגרף שהתוצאה של האלגוריתם הדטרמיניסטי גדולה תמיד מהתוצאה של האלגוריתם הסטוכסטי, התוחלת צריכה להיות בטווח הערכים של האלגוריתם ולכן הגיוני שקיבלנו שההסתברות שהתוחלת תהיה גבוהה יותר קטנה מאוד. השערת האפס באמת לא מתאימה למה שמוצג בגרף.

22. מספר המסלולים מהמצב ההתחלתי למצב מטרה כלשהו הוא $10! \cdot \frac{1}{32}$. נרצה למצוא את מספר הסידורים האפשריים של נקודות הסיום וההתחלה כך שכל נקודות ההתחלה הן לפני נקודות הסיום המתאימות להן. מספר הסידורים של הנקודות בלי הגבלה זו הוא $10!$. נבחין כי עבור כל נקודה מתקיים כי מספר הסידורים של הנקודות כך שנקודת ההתחלה היא לפני נקודת הסיום זהה למספר הסידורים של הנקודות כך שנקודת ההתחלה היא אחרי נקודת הסיום (הסידורים האסורים) ולכן שווה ל- $\frac{1}{2} 10!$. מספר האפשרויות לסדר בין כל אחד מהזוגות הוא 2^5 . ולכן מסימטריה נקבל שמספר המסלולים הוא $10! \cdot \frac{1}{32}$. כי כל אחד מסידורים אלו מהווה מסלול אפשרי.

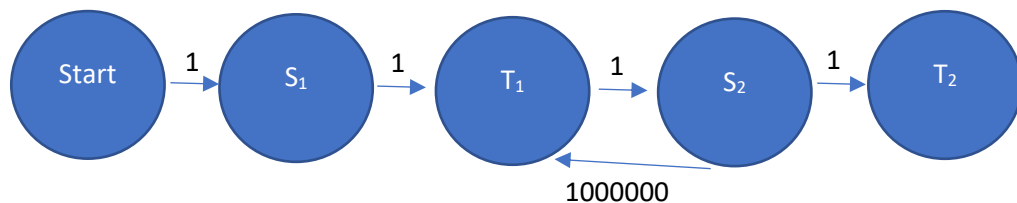
23. המרחקים שהתקבלו הם:

```
Greedy result: 135.06km
Stochastic (200 repetitions): 127.291km
A* (null heuristic): g(G)=127.29km, h(I)=0.00km, developed: 1640 states
```

24. החישובים עבור היוריסטיקות:

(נשתמש בהגדרה מהתרגולים עבור יוריסטיקות מיוזעות $h_1(s) = h_2(s)$)

- a. היוריסטיקה קבילה: היוריסטיקה צריכה לייצג את המרחק שייקח ל-s להגיע למצב מטרה כלשהו. כדי שיוכל להגיע למצב כזה הוא חייב להגיע מכל s_i לכל t_i , מרחק זה הוא לכל הפחות המרחק האווירי ביניהם ולכן המרחק של s ממצב מטרה כלשהו הוא לפחות המרחק המקסימלי.
- b. היוריסטיקה קבילה: נהיה חייבים לעבור בכל נקודות ההתחלה והסיום של ההזמנות ברשימת ההזמנות הממתינות בדרך למצב מטרה כלשהו, כלומר, כל מסלול למצב מטרה מכיל את כל הנקודות האלה ולכן קיים תת מסלול בין כל שתי נקודות כאלה על מסלול למצב המטרה, לכן נהיה חייבים לעבור לפחות את המרחק האווירי בין כל שתי נקודות כאלה ובפרט את המרחק המקסימלי. היוריסטיקה הזו מיוזעת מ-a כי יש לה יותר אפשרויות למסלולים ולכן לא ייתכן כי אורך המסלול המקסימלי קצר יותר.
- c. היוריסטיקה לא קבילה: המרחק האווירי בין הצמתים ברשימה B יכול להעיד גם על מרחקים שכבר עברנו ולכן יכול להיות שהמרחק שנשאר לעבור הוא קטן יותר. לדוגמה, אם אנחנו במצב האחרון ונשארה רק הזמנה אחת ב-B, יתכן כי המרחק המינימלי לצומת הסיום של הזמנה זו קטן מהמרחק האווירי בין צומת ההתחלה וצומת הסיום של הזמנה זו. כלומר, היוריסטיקה הזו לא אופטימית ולכן לא קבילה.
- d. היוריסטיקה קבילה: כדי להגיע לצומת מטרה כלשהו, נהיה חייבים להגיע מהצומת הנוכחי לכל נקודות ההתחלה ברשימת ההזמנות הממתינות ולכן בפרט נהיה חייבים לעבור את המרחק האווירי לכל אחת מנקודות ההתחלה ובפרט לעבור את המרחק המקסימלי מבין מרחקים אוויריים אלו.
- e. היוריסטיקה לא קבילה: נראה דוגמה נגדית (נחשב יוריסטיקה עבור מצב ההתחלה): Start $\rightarrow s_1 \rightarrow s_2 \rightarrow t_1, t_2$. נקבל ש- $h_a = 2, h_d = 2$, קיבלנו ש- $h_e = 4 > 3$.
- f. היוריסטיקה קבילה: כדי להגיע למצב מטרה כלשהו נהיה חייבים להגיע מכל נקודות ההתחלה לכל נקודות הסיום ברשימת ההזמנות הממתינות ולכן נהיה חייבים לעבור לפחות את המרחק בין כל אחת מהן (לא סכום, אלא כל אחד מהמרחקים) ובפרט את המרחק הארוך ביותר. היוריסטיקה מיוזעת מ-a כיוון שהמרחק האווירי מהווה חסם תחתון למרחק האמיתי.
- g. היוריסטיקה לא קבילה: נראה דוגמה נגדית לחישוב היוריסטיקה ממצב ההתחלה:



קיבלנו ש- $h_g(\text{Start}) = 1000000$ למרות שהמרחק האמיתי הוא 4 ולכן היוריסטיקה לא קבילה.

25. פיתחנו את יוריסטיקה d והתוצאה שקיבלנו היא:

```
A* (Custom heuristic): g(G)=127.29km, h(I)=20.41km, developed: 1604 states
```

רואים שהתוצאה שיקבלנו עבור $g(G)$ זהה לתוצאה שהתקבלה ביוריסטיקת האפס, זה הגיוני האלגוריתם קביל ולכן תמיד מוצא את הפתרון האופטימלי. ניתן לראות שמספר הצמתים

שפותחו קטן ממספר הצמתים שפותחו עבור יוריסטיקת האפס, גם זה הגיוני כיוון שיוריסטיקה d שולטת על יוריסטיקת האפס.
26. התוצאה שקיבלנו עבור יוריסטיקת MST:

A* (MST heuristic): $g(G)=127.29\text{km}$, $h(I)=81.34\text{km}$, developed: 475 states

רואים שהתוצאה שיקבלנו עבור $g(G)$ זהה לתוצאה שהתקבלה ביוריסטיקת האפס וב- d , זה הגיוני האלגוריתם קביל ולכן תמיד מוצא את הפתרון האופטימלי. כיוון שיוריסטיקה זו מוצאת את המרחק המינימלי לביקור בכל הצמתים היא מיועדת מיוריסטיקה d ולכן הגיוני שמספר הצמתים שפותחו אפילו קטן יותר.

פרק שני

(נשתמש בהגדרה מהתרגולים עבור יוריסטיקות מיועדות $h_1(s) \geq h_2(s)$)
א. עבור כל s $\text{Applicable}_h(s) = \text{True}$ היוריסטיקה קבילה, כלומר, אופטימית. עבור כל $\text{Applicable}_h(s) = \text{False}$ מקבלים שהיוריסטיקה היא 0. כיוון שלכל s המחיר גדול מ- δ חיובי כלשהו בהכרח המחיר גדול מהיוריסטיקה ולכן זו יוריסטיקה אופטימית ולכן קבילה.
ב. נגדיר יוריסטיקה חדשה:

```
def h1(s):
    if (Applicable_h(s)):
        return h(s)
    else:
        return min (h0(state), state in sons of s)
```

אם h מוגדרת על s נחזיר את $h(s)$, אחרת, נחזיר את הערך המינימלי המתקבל מהפעלת h_1 על הבנים של s בעץ המצבים.
לכל s מתקיים $h_1(s) \geq h_0(s)$ ולכן היוריסטיקה הזו מיועדת יותר.
ג. נגדיר יוריסטיקה חדשה:

```
def h2(s) :
    if (Applicable_h(s)):
        return h(s)
    else:
        return  $\delta$ 
```

לכל s מתקיים $h_2(s) \geq h_0(s)$ ולכן היוריסטיקה הזו מיועדת יותר.
ד. הטענה נכונה. נתאר אלגוריתם שמקיים טענה זו. אם נשתמש ב- h' נדע כבר בצומת ההתחלה מהו המחיר שאנחנו צריכים לשלם על המסלול.
נשתמש באלגוריתם דמוי A^* . במצב ההתחלה מחשבים את היוריסטיקה של מצב ההתחלה, זהו המחיר של מצב המטרה האופטימלי, נסמן מחיר זה ב- P . נמשיך בחיפוש כמו A^* שמשתמש ב- $h_0(h', s)$. אם מגיעים למצב שאינו מצב מטרה ומחיר ההגעה אליו גדול או שווה ל- P , לא נמשיך את החיפוש בתת העץ של המצב הזה בעץ החיפוש. אנחנו יכולים להפסיק את החיפוש כי בהכרח כבר לא נמצא מחיר אופטימלי בתת העץ הזה.
האלגוריתם המוצג הוא קביל כי הוא בהכרח ימצא פתרון אופטימלי, כי הוא בוודאות אפשר למצוא את הפתרון האופטימלי עם המחיר P .

מספר המצבים שיפותחו באלגוריתם המוצג חסום מלמעלה ע"י מספר המצבים שיפותחו ע"י A^* שמשמש ב- $h_0(h', s)$. זאת מכיוון של- A^* אין חסם על עומק עץ החיפוש בחיפושים בתתי עץ שאין בהם פתרון ובאלגוריתם המוצג יש.

האלגוריתם שאנחנו מציעים דומה לאלגוריתם שהוצג בהצעה בו מגבילים את עומק החיפוש עם מספר כלשהו, פה נגביל לפי יוריסטיקת מצב ההתחלה.