**AI – HW1**

חלק א'

|  |  |
| --- | --- |
| **פרמוטציות** | **K** |
| 1 | 1 |
| 6 | 2 |
| 90 | 3 |
| 2520 | 4 |
| 113400 | 5 |
| 7484400 | 6 |
| 681080400 | 7 |
| 81729648000 | 8 |
| 1.25046E+13 | 9 |
| 2.37588E+15 | 10 |

1. הטבלה:

חלק ג'

1. מקדם הסיעוף המינימלי הינו 0 כאשר המצבים שעבורם מקדם הסיעוף הינו 0 הם מצבי מטרה שעבורם מתקיים כאשר v הינה צומת על המפה, זאת משום שכאשר אין אופרטור חוקי המאפשר מעבר למצב אחר.

מקדם הסיעוף המקסימלי הינו k והוא ייתכן כאשר כל הזמנה הינה מצומת אחר במפה וכן קיים מסלול ישיר מצומת ההתחלה לכל אחד מצמתים אלו – כלומר, האוטובוס יכול לבחור כל אחד מההזמנות לאיסוף ראשון.

1. לא יכולים להיות מעגלים במרחב החיפוש משום שלפי האופרטורים החוקיים קבוצת ההזמנות הממתינות יורדת מונוטונית וכן קבוצת ההזמנות שבוצעו עולה מונוטונית ולפי הגדרת האופרטורים לפחות אחת מהן משתנה בעקבות הפעלת אופרטור ולכן אין מצב שבו יתקיים עבור שני מצבים כך וגם וכן נמצאת אחרי על אותו מסלול.
2. G הם קבוצת המצבים כך שלא נותרו על האוטובוס הזמנות וכן כל ההזמנות ברשימה המקורית בוצעו. הגדרה פורמלית:
3. לפי הגדרת האופרטורים והנתון בשאלה לפיו אין שתי הזמנות שמתחילות ונגמרות במיקומים זהים נסיק כי בכל מעבר בין מצבים מתקיים אחד מהמקרים הבאים: או שאדם אחד עולה על האוטובוס, או שאדם אחד יורד מהאוטובוס או שאדם אחד עולה ואדם אחד יורד. לכן, ב-G יש מצב עבור כל הזמנה כך שאדם זה הינו האחרון שיורד מהאוטובוס (לא יכולים להיות שני אנשים שונים היורדים באותה תחנה) => הביטוי המתאים הינו:
4. לא יתכנו בורות שאינם מצבי מטרה משום שנניח שאנו במצב S כלשהו שאינו מצב מטרה לכן בהכרח קיים מצב S’ שונה מ-S כך שבמצב זה מורידים או מעלים לפחות נוסע אחד וכן קיים מסלול במפה בין ו- (ההנחה לגבי המסלולים במפה ניתנת ב-FAQ ).
5. פונקציית העוקב:
6. הערך המקסימלי הינו מצב זה יקרה כאשר בכל תחנה או נאסוף נוסע או נוריד נוסע כלומר מספר האנשים על האוטובוס בכל רגע נתון הוא לכל היותר 1, כלומר מספר הנוסעים על האוטובוס יהיה : 0, 1, 0, 1 וכו'.

הערך המינימלי הוא k+1 במקרה זה בכל תחנה נאסוף נוסע ונוריד נוסע ולכן מלבד במצב ההתחלה ובמצב הסופי על האוטובוס יש בדיוק נוסע 1 ולכן נבצע k מעברים בכדי לאסוף k נוסעים וכן נעשה מעבר נוסף בכדי להוריד את הנוסע האחרון.

חלק ד'

1. *הקוד לוקח הזמנה באופן ראנדומלי ומדפיס את צומת המוצא וצומת היעד של ההזמנה וכן את הקורדינטות של שני הצמתים. בנוסף, הקוד מדפיס את המרחק האווירי בין צמתים אלו.*

*הפלט:*

*load\_map\_from\_csv: 21.79sec*

*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: -1.00km*

1. *הפלט שהתקבל:*

*Junction idx: #851288*

*waiting for bus: [(23695, 33320)]*

*orders on bus: [(851288, 533396), (32056, 834603)]*

*finished orders: [(47521, 606430), (466524, 29249)]*

*Is goal? False*

*חלק ה'*

1. *הערך שהתקבל הינו :77.35 ק"מ.*

**

1. *התשובה היא C.  
   לא ניתן לקבוע באופן כללי משום שהמספר המתקבל הינו סכום כל המסלולים על כל ההזמנות בנפרד, בסכום זה אין התחשבות במסלולים חופפים (לכן זה אינו חסם תחתון) וכן אין התחשבות במרחק בין הזמנה להזמנה (למשל המרחק בין S4 ל-S3 בגרף לא מחושב – ולכן זה אינו חסם עליון).*

*חלק ו'*



|  |  |
| --- | --- |
| Solution cost (total distance in KM) | Input file |
| *135.06km* | TLV\_5 |
| *539.59km* | SDEROT\_50 |
| *565.68km* | HAIFA\_100 |
| *1329.21km* | BEER\_SHEVA\_100 |

1. *חסם תחתון : 2 צמתים אם כל כל ההזמנות יוצאות מV ומגיעות ל U .*

*מ* I *יפותח רק המצב X = וממנו יפותח רק המצב Y=*

*חסם עליון :*

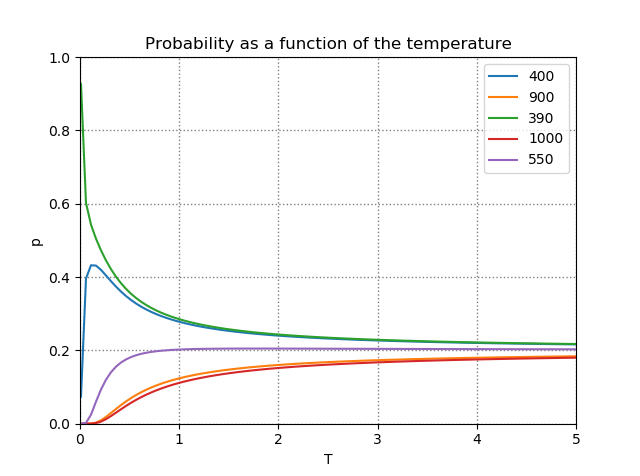
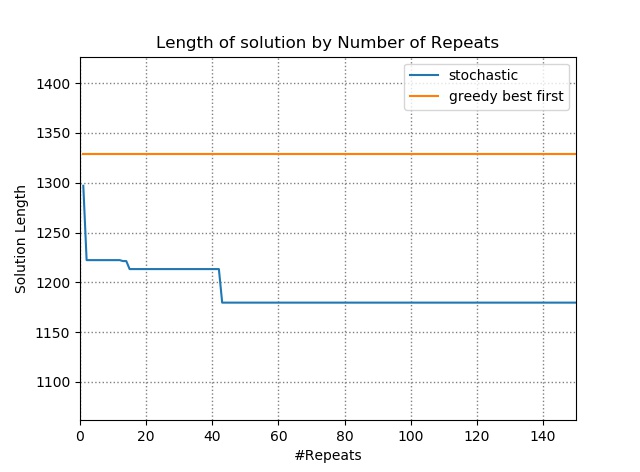
*חלק ז'*

1. *נוכיח ששינוי הסקאלה אינו משנה את התפלגות הווקטור, כלומר נראה שעבור ווקטור נתון מתקיים:*

*תהי נפריד למקרים:*

1. =>
2. =>

נשים לב כי לעולם לא יהיה 0 כי פונקציות ה-Score שלנו מחשבת מרחק אווירי ולעולם לא נקבל מרחק אווירי 0 עבור שני מצבים עוקבים – כי אז הם היו אותו מצב.

1. הגרך המתקבל:  
   
2. על פי התבוננות בגרף נראה כי כאשר נקבל כי ההסתברות של הנקודה הטובה ביותר לפי פונקציית העלות (נסמנה ) תקיים . זאת משום שכפי שצויין בחלק ז' במסמך ככל ש-T קטן יותר נבחר אפשרויות "בטוחות יותר" ובמקרה שלנו הנקודה הקרובה ביותר אווירית הינה האפשרות הבטוחה ביותר שהיא גם הקרובה ביותר בפועל.
3. על פי התבוננות בגרף נראה כי כאשר נקבל כי ההסתברות של כל אחת מ-N הנקודות הטובות יותר לפי פונקציית העלות תקיים . זאת משום שכפי שצויין בחלק ז' במסמך ככל ש-T יגדל יותר ההסתברות של כל אחת מהנקודות האפשרויות תראה לנו שווה ולכן לכל נקודה ההסתברות תשאף ל-
4. **
   * ממוצע מדגמי: 1190.38756667

סטיית תקן מדגמית:

P-Value:

* + על פי מה שנלמד בתרגול נדחה את השערת ה-0 משום שקיבלנו .
  + נתבונן בנתונים שקיבלנו עבור הנקודה הראשונה, נשים לב שתוצאת האלגוריתם החמדני הינה 1329.21 וכי ממוצע הרצות האלגוריתם הסטוכאסטי הינה כ-1190.  
    בנוסף נשים לב כי סטיית התקן קטנה משמעותית מהמרחק בין ממוצע ההרצות הסטוכאסטי לבין תוצאת האלגוריתם החמדני ולכן נסיק שרוב הרצות האלגוריתם הסטוכאסטי יפיקו תוצאה נמוכה מהתוצאה של האלגוריתם החמדן (ואף לא יתקרבו אליה משום שסטיית התקן קטנה מהמרחק בין הערכים) ולכן השערת האפס שגויה.

1. *בdataset שלנו יש 5 הזמנות עם מוצאים ויעדים שונים כלומר 10 נקודות שונות על המפה וכולן שונות מנקודת ההתחלה של האוטובוס. מספר המסלולים עד למצב מקבל כלשהו שקול למספר הפרמוטציות החוקיות על איסוף והורדת ההזמנות כולן. בשלב ראשון ניתן לאסוף כל אחד מ5 ההזמנות. לאחר מכן ניתן לאסוף כל אחד מ4 ההזמנות האחרון או להוריד את ההזמנה שעל האוטובוס. וכן הלאה . נחשב בעזרת רקורסיה. נגדיר:*

*numberOfRouts(n,k) =*

*כאשר n הוא מספר ההזמנות שמחכות וk מספר ההזמנות שנמצאות על האוטובוס. נוסחא זו נכונה רק עבור מצבים בהם אין חפיפה בין נקודות העלאה והורדה בהזמנות.*

*אז: numberOfRouts (5,0) =113400*

1. *תוצאות האלג' הדטרמיניסטי: 135.06km.*

*תוצאת האלג' הסטוכאסטי: 132.682km.*

*תוצאת A\*: 127.29km ופותחו: 1640 מצבים.*

1. **נשתמש בהגדרת היוריסטיקה מיודעת מההרצאה.**
2. **קבילה**: הפונקציה מחזירה את המרחק האווירי הגדול ביותר שיש לעבור עבור הזמנה שעדיין לא עלתה על האוטובוס (מרחק אווירי מקסימלי מנק' האיסוף של ההזמנה עד נק' ההורדה של אותה הזמנה), נשים לב שלא יכול להיות שנבצע בפועל מרחק קטן יותר מהמרחק האווירי הנ"ל עד למצב המטרה משום שעלינו לאסוף את ההזמנה ולאחר מכן להגיע אל יעדה וכפי שצוין בתרגול לא יכול להיות כביש קצר מהמרחק האווירי. בנוסף, במידה ויש יותר מהזמנה אחת יתכן ונצטרך לעבור מרחק יותר גדול מהמרחק שהפונקציה חישבה. לכן, פונקציה זו תמיד תעריך בצורה אופטימית את המרחק מהמצב הנוכחי למצב המטרה.
3. **קבילה**: הפונקציה מחזירה את המרחק האווירי הגדול ביותר שיש לעבור עבור בין נקודת איסוף ונקודת הורדה מבין הזמנות שעוד לא אספנו (מרחק אווירי מקסימלי מנק' איסוף של הזמנה כלשהי עד נק' הורדה של הזמנה כלשהי), באופן דומה לפונקציה a נשים לב שלא יכול להיות שנבצע בפועל מרחק קטן יותר מהמרחק האווירי הנ"ל עד למצב המטרה משום שעלינו להגיע אל שתי הנק' הנ"ל, למרות שכעט הסדר אינו נאכף עדיין עלינו לעשות את הדרך בין שתי הנק' וכפי שצוין בתרגול לא יכול להיות כביש קצר מהמרחק האווירי. לכן, פונקציה זו תמיד תעריך בצורה אופטימית את המרחק מהמצב הנוכחי למצב המטרה.
4. **אינה קבילה**: הפונקציה מחזירה את המרחק האווירי הגדול ביותר שיש לעבור עבור בין נקודת איסוף ונקודת הורדה של הזמנה אשר כבר אספנו. נראה דוגמה:

יהיו שתי הזמנות כאשר המרחק האווירי עבור ההזמנה הראשונה הינו 100 והמרחק האווירי עבור ההזמנה השנייה הינו 1 ונניח כי שתי ההזמנות יורדות באותה נק'. נניח שראשית אספנו את ההזמנה הראשונה וכרגע אנו בנק' בה אוספים את ההזמנה השנייה => על המצב הנוכחי שלנו הינו 100 למרות שמצב היעד במרחק אווירי של 1 => אינה קבילה. **מיודעות:** ההגדרה של היוריסטיקה מיודעת רלוונטי רק להיוריסטיקות קבילות.

1. **קבילה**: הפונקציה מחזירה את המרחק האווירי הגדול ביותר שיש לעבור מהנק' הנוכחית עד נק' האיסוף של הזמנה שעדיין לא עלתה על האוטובוס, נשים לב שלא יכול להיות שנבצע בפועל מרחק קטן יותר מהמרחק האווירי הנ"ל עד למצב המטרה משום שעלינו להגיע לנק' האיסוף של ההזמנה ולאחר מכן להגיע אל יעדה וכפי שצוין בתרגול לא יכול להיות כביש קצר מהמרחק האווירי. לכן, פונקציה זו תמיד תעריך בצורה אופטימית את המרחק מהמצב הנוכחי למצב המטרה.
2. **קבילה**: הפונקציה מחזירה את הסכום של המרחק האווירי הגדול ביותר שיש לעבור מהנק' הנוכחית עד נק' האיסוף של הזמנה שעדיין לא עלתה על האוטובוס והמרחק האווירי הגדול ביותר שיש לעבור עבור הזמנה שעדיין לא עלתה על האוטובוס, נשים לב שלא יכול להיות שנבצע בפועל מרחק קטן יותר מהמרחק האווירי הנ"ל עד למצב המטרה משום שעלינו להגיע לנק' האיסוף של שתי ההזמנות הנ"ל וכן לנק' היעד של שתי ההזמנות וכפי שצוין בתרגול לא יכול להיות כביש קצר מהמרחק האווירי. לכן, פונקציה זו תמיד תעריך בצורה אופטימית את המרחק מהמצב הנוכחי למצב המטרה.
3. **קבילה**: הפונקציה מחזירה את המרחק האמיתי הגדול ביותר שיש לעבור עבור הזמנה שעדיין לא עלתה על האוטובוס (מרחק אמיתי מקסימלי מנק' האיסוף של ההזמנה עד נק' ההורדה של אותה הזמנה), נשים לב שלא יכול להיות שנבצע בפועל מרחק קטן יותר מהמרחק האמיתי הנ"ל עד למצב המטרה משום שעלינו לאסוף את ההזמנה ולאחר מכן להגיע אל יעדה. לכן, פונקציה זו תמיד תעריך בצורה אופטימית את המרחק מהמצב הנוכחי למצב המטרה.
4. **אינה קבילה**: הפונקציה מחזירה את המרחק האמיתי הגדול ביותר שיש לעבור עבור בין נקודת איסוף ונקודת הורדה מבין הזמנות שעוד לא אספנו, נשים לב כי אנו לא מוכרחים לבצע את סדר איסוף והורדת ההזמנות לפי הסדר אותו חישבה הפונקציה כלומר: נניח כי ישנן שתי הזמנות שעוד לא אספנו וקיימים הכבישים הבאים:

נקבל במצב זה: משום שהמרחק המקסימלי הינו מ- ל- למרות שנוכל לבצע את המסלול הבא באורך קצר יותר : .

**מיודעות:**

בגלל שאנו משתמשים בהגדרה מההרצאה וכן כל הפונקציות הקבילות לעיל הן פונקציות על הקבוצה W נקבל שעבור כל מצב s כך ש- נקבל כלומר עבור כל מצב בו כבר אספנו את כל ההזמנות אך יש יותר מהזמנה אחת על האוטובוס נקבל כי פונקציית ההיוריסטיקה מחזירה את הערך 0 כך שלכל שתי פונקציות לעיל קיים מצב שאינו מצב מטרה וכן מתקיים .

1. הפלט שהתקבל:

A\* (Custom heuristic): g(G)=127.29km, h(I)=20.37km, developed: 1672 states

1. הפלט שהתקבל:

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

**פרק שני**

1. יהי מצב s, נפרק למקרים:
   * => , מהנתון h קבילה => מתקבל כי .
   * => => מתקבל כי .

בשני המקרים נקבל כי => קבילה לפי הגדרה.

1. בסעיף זה נניח כי מצבי המטרה הינם עלים (אך לא בהכרח עלה הוא מצב מטרה, מותר להניח זאת כי ניתן להגדיר את succ כך שאם מצב הינו מצב מטרה פונקציית succ תחזיר קבוצה ריקה) וכן כי פונקציית המחיר חסומה מלמטה ע"י 1. בנוסף נשתמש בהגדרת המיודעות מהתרגול בסעיף זה.

נגדיר את היוריסטיקה הבאה:

כאשר height הינה פונקציה המחזירה את המרחק מהמצב הנוכחי לעלה הקרוב ביותר בתת העץ שלו.

פסודו קוד:

H1(state,h0):

Return max(state.get\_height() , h0(state))

Get\_height():

min <- int.max\_val

For (son in self.sons)

If(son.get\_height() < min)

Min= son.get\_height()

Return min

נוכיח כי הפתרון קביל:

נתון כי מרחב המצבים הינו עץ => מצבי המטרה בעלים => פונקציית הגובה תחזיר 0 עבור מצבי מטרה ולכן תחזיר 0 עבור מצבי מטרה תמיד (גם פונקציית הגובה וגם היוריסטיקה מסעיף א' יחזירו 0 למצבי מטרה).

בנוסף נראה כי מתקיים לכל מצב שאינו מצב מטרה:

נשים לב שנתון כי קבילה ולכן אם היא המקסימלית עבור מצב s אז קבילה. אם מקסימלית נקבל את מספר המצבים המינימלי בנינו לבין העלה הקרוב ביותר, לפי ההנחה כי העלות חסומה מלמטה על ידי 1 נקבל שבמקרה הטוב ביותר הינו המרחק מ-s לעלה הקרוב ביותר וכן נקבל כי קבילה אף היא.

נוכיח כי הפתרון מיודע יותר מהסעיף הקודם:

כאשר המעבר נובע מכך שעבור כל מצב נקבל ערך השווה ל- או גדול ממנו.