* 1. : the number of ways to arrange the apartments between themselves
  2. : before every apartment we visit(k) we choose with repetitions which lab to visit(m) or not at all(1)
  3. : choosing the lab the ambulance visits after the last apartment
* Assuming there is a single path between every pair of points(labs and apartments)

1. Bonus

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | Estimated calculation time |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

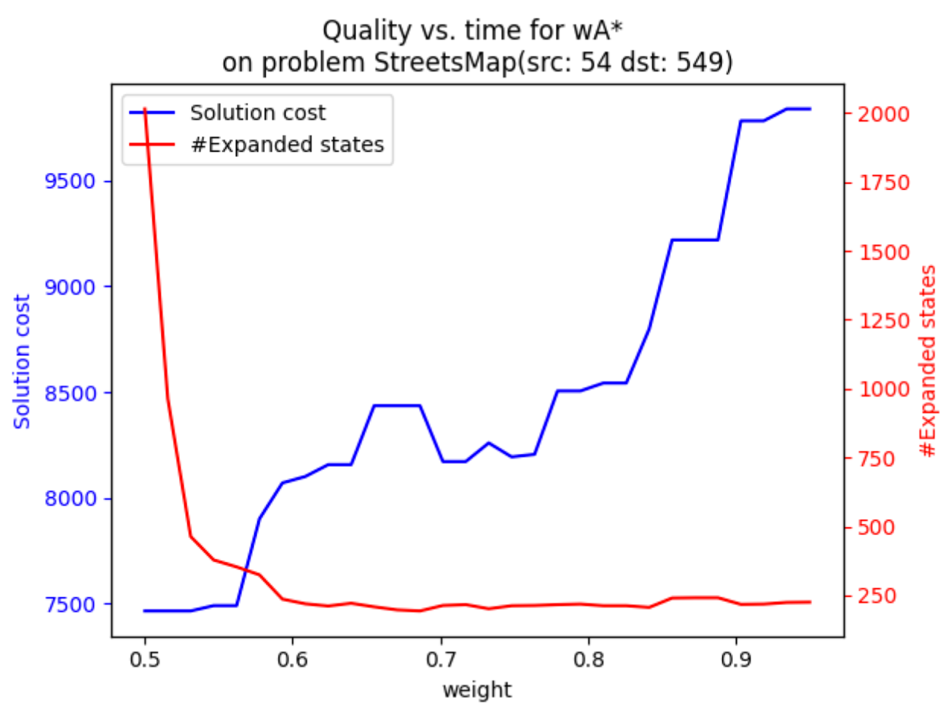
1. דרגת היציאה המינימאלית היא בקבוצת המצבים המסיימים הבאה

במילים, קבוצת המצבים הסופיים כך שבמסלול ביקרו בכל המעבדות,  
אם לא ביקרנו במעבדה מסויימת אז מהגדרת האופרטור נוכל לבקר בה, ולכן דרגת היציאה אינה 0.

דרגת יציאה מקסימלית היא והיא מתקיימת כאשר אנחנו במצב התחלתי ויש מספיק מטושים כדי לבדוק כל דירה לכן אפשר לנסוע לכל דירה() או אפשר לנסוע לכל המעבדות() לאסוף מטושים.

1. לא, נתבונן באופרטורים.  
   נניח שכן, נניח גם שקיימת קשת עם האופרטור לביקור בדירה , האופרטור מבצע ,מהגדרת האופרטור כדי לחזור ל צריך ש לא יהיה ב אך אין אופרטור שהופך את הפעולה.  
   לכן אם קיים מעגל הוא בהכרח בנוי רק מהאופרטורים , זאת אומרת שנמצאים במעבדה ולאחר כמה נסיעות(בין מעבדות) חוזרים לאותה במעבדה בלי לעבור בדירות בדרך, אך הדבר לא אפשרי מבחינת משום שלא אספנו בדיקות נוספות וגם ביקרנו במעבדה בעבר()
2. מספרי המטושים הם מספרים קבועיים לכן השדה matoshim יכול לקבל כל ערך טבעי, אז מספר המצבים הוא אין סופי.  
   לא כל המצבים ישיגים, לפחות בגלל העובדה שמספר המטושים הוא קבוע ישנם ערכים שלא יכולים להתקבל.   
   כן, למשל אם לאחר מעבר בכל המעבדות מספר המטושים (ההתחלתי + המטושים במעבדות) לא מספיקים לשום דירה, נתקע במעבדה האחרונה.
3. המסלול הקצר ביותר הוא באורך הוא במצב ההתחלה יש לאמבולנס מספיק מטושים כדי לבצע בדיקות לכל הדיירים בכל הדירות וגם יש באמבולנס מספיק מקום לכל הבדיקות, כך שמסלול יכלול מעבר בין כל הדירות ויסיים במעבדה.  
   המסלול הארוך ביותר הוא באורך , הוא מורכב מביקור בכל המעבדות ברצף() לאחר מכן נבקר בדירות ובמעבדות לסירוגין().

כדי ליצור את המסלול הארוך ביותר נרצה למקסם את הביקורים במעבדות, לשם כך בין כל שתי דירות נבקר במעבדה, סדרת הביקורים הזאת תביא לקשתות. ולפני זה נבקר בכל המעבדות .

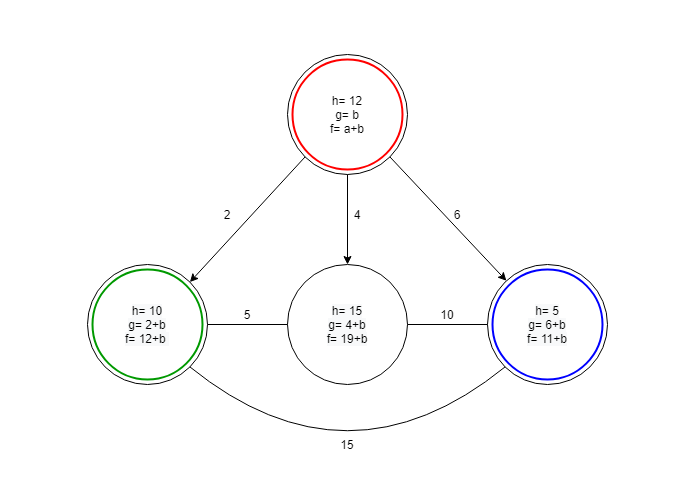
2. *  
   ניתן לראות שככול הw קטן איכות הפתרון עולה ומספר המצבים שפותחו גדל. אזורים בהם מספר המצבים שפותחו לא גדל באופן מעריכי ועם cost לא גדול מידי, כלומר החל מ~0.55 עד ~0.8.  
     
   בנוסף, ניתן לראות על פי כלל האצבע לא נכון באופן גורף, לדוגמה איכות הפתרון ב0.78 טובה יותר מ0.75 בסתירה לכלל האצבע.*
3. *שימוש ב"מרחב על" גורם להוספה של מצבים נוספים רבים לבעיה, דבר המביא לפגיעה בביצועים משום שהוא מגדיל בצורה משמעותית את עומק הפתרונות שצריך לבדוק, בנוסף הדבר היה פוגע ביכולתנו לשמור Cache של מרחקים בין נקודות עניין(מעבדות ודירות) בגלל אורכי הפתרונות.*
   1. *@dataclass(frozen=True)*
   2. *שימוש בFrozenSet מבטיח כי לא יהיו שינויים במבני הנתונים הפנימיים*
   3. *כן, כפי שראינו בהרצאה במהלך הריצה האלגוריתם A\* עשוי לפגוש מצב שפגש בעבר   
      השורה באלגוריתם היא*

* 1. *נניח שמאתחלים את קבוצת המעבדות שבוקרו עבור המצב הבא כך:   
      next\_visited\_labs = state\_to\_expand.visited\_labs*

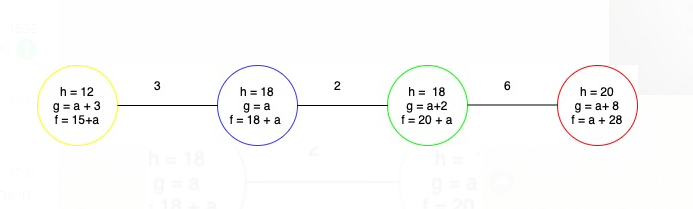
*next\_visited\_labs.add(next\_lab)*

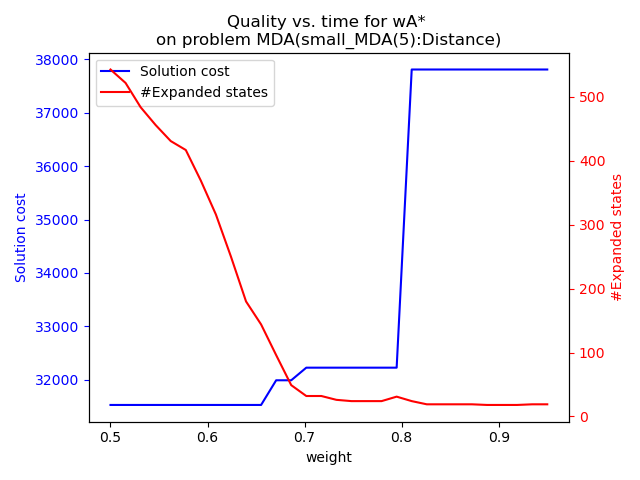
*הקוד יביא להוספת next\_lab לקבוצת המעבדות שבוקרו כבר גם עבור המצב הנוכחי, באיטרציה השניה כאשר נאתחל מעבדה נוספת שצריך לבקר בה קבוצת המעבדות שבוקרו תכיל גם את המעבדה מהאיטרציה הקודמת, דבר שיביא לריצה שגויה.*

1. *היוריסטיקה* MDAMaxAirDistHeuristic עבור אינה קבילה.
   1. *נפריך עם דוגמא נגדית, נניח שהגענו למצב הכחול(הדירה האחרונה שהיה צריך לבקר בה) וצריך לבחור לאיזה מעבדה ללכת(המצבים התחתונים).  
      Astar עם היוריסטיקה הנ"ל תבחר ללכת למצב הכחול משום ש שהוא המינימלי, אך הפתרון האופטימלי יהיה ללכת למעבדה הקרובה ביותר(הירוקה) ולסיים בה, המחיר הסופי יהיה אבל הערך היוריסטי שם גבוהה יותר ולכן הצומת לא יבחר.*

**

1. *היוריסטיקה* MDASumAirDistHeuristic עבור אינה קבילה.
   1. *נפריך עם דוגמא נגדית, נסתכל על הדירות הבאות שצריך לבקר,  
      נניח כי כעת אנחנו נמצאים במצב הכחול. במצב זה, היוריסטיקה תלך לירוק, אחר כך לצהוב ואז לאדום. כלומר ערכה יהיה - 18=2+2+3+3+2+6. אך ישנו מסלול קצר יותר והוא ללכת לכתום, אחר כך לירוק ואז לאדום. ערך מסלול זה יהיה- 14=3+3+2+6. כלומר ניתן לראות שהיוריסטיקה לא תיתן לנו אתה ערך אופטימלי כלומר היוריסטיקה אינה קבילה.*

**

1. *היוריסטיקה*MDAMSTAirDistHeuristic עבור קבילה.
   1. מאותה סיבה כמו היוריסטיקה הקודמת, הבעיה נובעת שחישוב המרחקים נעשה בין נקודות(צמתים) שלא צריך לבקר בהם כלל.
2. *באזור אנחנו רואים שיש אזור שבו הפתרון קרוב לאופטימלי ומספר המצבים שפותחו קרוב למינימאלי, כמובן שכפי שלמדנו אין תשובה חד משמעית. אם איכות הפתרון חשובה אז כדאי להגדיר לאומת זאת אם מהירות קבלת פתרון חשובה אז כדאי להגדיר באזור זה מתקבל פתרון שהוא לא רחוק מאופטימלי ומספר המצבים המפותח נמוך יחסית.*

|  |  |  |  |
| --- | --- | --- | --- |
|  | MDAMaxAirDistHeuristic | MDASumAirDistHeuristic | MDAMSTAirDistHeuristic |
|  | *X* | *X* | *X* |
|  | *קביל* | *X* | *קביל* |

1. *השינוי בפונקציית העלות הביא לבחירת מסלולים שונים בהשוואה בין סעיף זה לפלט מסעיף 19 נזהה ש הוא שונה לחלוטין,  
   בסעיף 19 קיבלנו ביחידות מרחק,  
   לאומת סעיף זה בו קיבלנו ביחידות כסף,  
   משום שנחברו מסלולים שונים ניתן גם לזהות בעמודת שהערכים מקיימים עבור סעיף 19 המרחק קטן יותר, ועבור סעיף זה המחיר נמוך יותר.*
2. היוריסטיקה MDATestsTravelTimeToNearestLabHeuristic עבור קבילה
   1. *הוכחה:*

*נכתוב את פונקציית היוריסטיקה עבור מצב* z *כלשהו*

*כאשר apps*

*נרצה להראות שבכל צעד מתקיים*

*במצב מסיים במעבדה כשאין יותר דירות לבקר מתקיים*

*אחרת נותרו דירות לביקור.*

*נסמן ב*את המרחק למעבדה הקרובה ביותר לדירה d.

*נסמן ב את המרחק שהאמבולנס נסע בפועל עד למעבדה.*

*עבור כל דירה בהכרח מתקיים .*

*ולכן*

משום שמספר הדיירים הוא קבוע לכל דירה גם יתקיים

1. *נזהה שבהשוואה להרצות הקודמות של הבעיה moderate\_MDA הערכים של המדד test-travel היו משמעותית יותר גדולות ביחס להרצה הנוכחית בהרצה הנוכחית קיבלנו 131265 לאומת ההרצות הקודמות שבהן התוצאות היו באזור ה170 אלף. בנוסף נזהה כי בהרצה הנוכחית שאר המדדים גרועים יותר משאר ההרצות משום שהיוריסטיקה לא מנסה להביא אותם לערך אופטימלי.*
2. *האלגוריתם מחזיר פתרון,  
   בשלב הראשון האלגורתים מריץ A\* ובהרצאה ראינו שהוא שלם,  
   בשלב השלישי האלגוריתם מריץ UCS על הפתרונות מהשלב הקודם(כולם פתרונות תקינים) וגם עבור UCS אנו יודעים שהוא שלם כאשר משקל הקשתות חסום(מן ה*הבהרה אנו יודעים שאכן כך)
3. *נניח בשלילה שלא, זאת אומרת שהפתרון שחזר אינו אופטימלי מבחינת הקריטריון המשולב.  
   ז"א שאחד משני דברים קראו:*
   1. *UCS החזיר פתרון שהוא לא אופטימלי מבין המסלולים שמשקלם בניגוד לכך שUCS קביל.*
   2. *A\* החזיר שאינו מינימלי, בניגוד לכך שA\* קביל.*

*קיבלנו סתירה, לכן בהכרח שהפתרון שחזר אופטימלי מבחינת הקריטריון המשולב.*

|  |  |  |  |
| --- | --- | --- | --- |
| *סעיף* | *אלגוריתם* | *dist* | *tests-travel* |
| *28* |  | *43034.794* | *176505.013* |
| *38* |  | *65686.522* | *132209.981* |
| 35 |  | *93226.428* | *131265.153* |



*לכן התשובה היא כן.*

1. *נפריך עם דוגמא נגדית*
2. *Asd*
3. *גרף המצבים של הוא כל המסלולים החוקיים בבעיית מד"א, לכן מספר המצבים גדול מאוד.  
    רץ על בעיית מד"א ולכן מרחב החיפוש קטן יותר.*
4. *מספר הפיתוחים קטן יותר, משום שמשתמשים בMDASumAirDist כדי לבחור את הצומת הבא לפיתוח(משום שהוא מיודע יותר מ MDAMSTAirDist)*

***חלק י***

1. *זכרון  
   ב לא צריך לשמור קבוצה של צמתים זמניים ולכן צריכת הזיכרון היא רק למשתנים לוקליים לפעולה הרקורסיבית.*
   1. *זמן ריצה*
   2. *משום שלא שומרים מצבים בזיכרון יש צורך לפתח את אותם המצבים שוב.*
   3. *על פי מה שלמדנו בתרגול 4, סיבוכיות זמן הריצה של לא נפגעת.  
      בין אין שינוי בממד משום ששניהם בודקים את כל המצבים בכל עומק.*
2. *ג*
   1. *במקרה הגרוע ביותר של גרף שרוך בו לכל צומת ערך הf גדל ב1, בכל איטרציה נפתח צומת אחד חדש, לכן מספר האיטרציות = עומק הגרף = מספר הצמתים בגרף.*