תרגיל בית 1

ולדיסלב בוגון - 320907645

איילה אביטל - 203242003

חלק א

.1

$(k!)\cdot(m+1)^k\cdot m$

$k!$ - סידור של דירות. $(m+1)^k$ - לפני כל דירה בחר מעבדה או לא. m - בחר מעבדה סופית.

.2

$\left(\sum_{i=0}^{m}\frac{m!}{(m-i)!}\right)^k\cdot k!\cdot m$

$k!$- סידור של דירות. m -בחירת מעבדה הסופית. $\left(\sum_{i=0}^{m}\frac{m!}{(m-i)!}\right)^k$ - לפני ביקור בכל דירה בחר קבוצה של מעבדות לעבור בהן ברצף.

.3

טבלה:

k	m	#possiblePaths	Estimated calculation time	
7	2	22.04E+06	18.5	[secs]
7	3	2.48E+08	3.85	[mins]
8	3	79.27E+08	2.26	[hours]
8	4	6.30E+10	19.6	[hours]
9	3	28.54E+10	3.7	[days]
10	3	11.42E+12	5.3	[months]
11	3	5.02E+14	21.1	[years]
12	3	2.41E+16	1.1	[thousands years]
12	4	46.78E+16	22.4	[thousands years]
13	4	30.41E+18	1.5	million years]

חלק ג

.4

$d_{max}=k+m$

אם במצב התחלתי יש מספיק מטושים כדי לעבור בכל דירה וגם קיבולת האמבולנס מספיק גדולה ,מכיוון שעוד לא עברנו באף מעבדה נוכל להפעיל $k+m$ אופרטורים שזה מספר כל האופרטורים האפשריים בבעיה זאת.

$d_{min}=0$

נניח כי לכל $i\in[k]$ מתקיים $AmbulanceTestCapacity<d_i.roommates$ לכן לא נוכל להפעיל את אופרטור o_{d_i}. ונניח כי עברנו על כל המעבדות, יהי מצב s בו אנו נמצאים, לא נוכל להפעיל את אופרטור o_{l_j} כך ש $j\in[m]$, כלומר לא נוכל להפעיל אף אופרטור על מצב s.

.5

לא ייתכנו מעגלים. נניח בשלילה כי קיים מעגל אזי קיים מצב $s\in S_{MDA}$ בו ניתן לעבור פעמיים עם לפחות מצב אחד באמצע. אם מצב זה הוא של דירה אז קיבלנו סתירה להגדרת האופרטורים בכך שלא עוברים בדירה יותר מפעם אחת. אם המצב הוא מעבדה $j\in[m]$ נניח כי המצב לפני שחוזרים אל j הוא גם של מעבדה, כלומר המקררים ריקים ולפי ההנחה כבר ביקרנו ב-j -זו סתירה להגדרת אופרטור o_l.

אחרת נניח כי מצב לפני שחוזרים אל j הוא של דירה i , מכיוון שחוזרים בדיוק לאותו מצב s השדה $s.Transfered$ נשאר אותו

דבר וזה סתירה להגדרת אופרטור o_{l_j} לפיו המצב הבא חייב להכיל את הבדיקות שיש בתוך המקרר של האמבולנס ובמקרה שלנו זה בדיקות של דירה i שעברנו בה לפני שחזרנו אל מעבדה j.

.6

יש אינסוף מצבים כי אחד המרכיבים של כל מצב זה מספר המטושים הזמינים באמבולנס שיכול לקבל כל ערך מקבוצת המספרים הטבעיים שהיא אינסופית.

ייתכנו מצבים לא ישיגים לדוגמא כל מצב בו מספר המטושים גדול יותר בסכום של מטושים בכל המעבדות ומספר המטושים ההתחלתי באמבולנס. סכום הנ"ל הוא חסם עליון על מספר המטושים שניתן להגיע אליו ממצב ההתחלתי.

.7

כן, בור הוא מצב בו דרגת היציאה שלו 0, בסעיף 4 הראנו איך זה יכול להתאפשר.

.8

נסמן את הטווח האורכים האפשריים ב$[a,b]$ אזי $a=2$ מכיוון ש$Apartments \neq \emptyset$ אזי בבעיה קטנה ביותר $|Apartments| = 1$ לכן חייבים קשת ממצב התחלתי אל מצב בו אנו אוספים בדיקות של הדירה היחידה, ואז עוד קשת אל מצב בו מוסרים בדיקות של הדירה אל המעבדה,

זה יהיה מצב מטרה כי אין עוד בדיקות לבצע ואנו נמצאים במעבדה. אורך 1 לא ייתכן כי חייבים לבקר לפחות בדירה אחת ולפחות במעבדה אחת.

$b = m+k+k$, זה ייתכן אם לפני ביקור בדירה הראשונה נעבור קודם בכל המעבדות כדי לאסוף מטושים

ואז אחרי כל ביקור בדירה נבקר במעבדה כדי לפרוק מקררים. ערך גבוה יותר לא ייתכן כי אז נעבור פעמיים בדירה או במעבדה בתנאי לא חוקי.

$[2,\ m+2k]$

.9

נגדיר את: $Succ_{MDA} : S \to P(S)$. בהינתן $s \in S_{MDA}$ אזי:

$Succ_{MDA}(s) = Succ_{Ap}(s) \cup Succ_{Lab}(s)$ כאשר:

$$Succ_{Ap}(s) = \left\{ \left(\begin{array}{c} d_i.loc, s,\ Taken \cup \{d_i\} \\ s.Transfered,\ s.Matoshim - d_i.roommates \\ s.VisitedLabs \end{array} \right) \middle| \begin{array}{c} i \in [k]\ , d_i \notin s.Taken \cup s.Transferred \\ d_i.roommates \leq s.Matoshim \\ d_i.roommates \leq AmbulanceTestCapacity - \sum\limits_{d \in s.Taken} d.roommates \end{array} \right\}$$

$$Succ_{Lab}(s) = \left\{ \left(\begin{array}{c} l_j.loc, s,\ \emptyset, \\ s.Transfered \cup s.Taken,\ s.Matoshim + l_j.Matoshim \cdot I_{[l_j \notin s.VisitedLabs]} \\ s.VisitedLabs \cup \{l_j\} \end{array} \right) \middle| \begin{array}{c} j \in [m], \\ (s.Taken \neq \emptyset \vee l_j \notin s.VisitedLabs) \end{array} \right\}$$

חלק ה

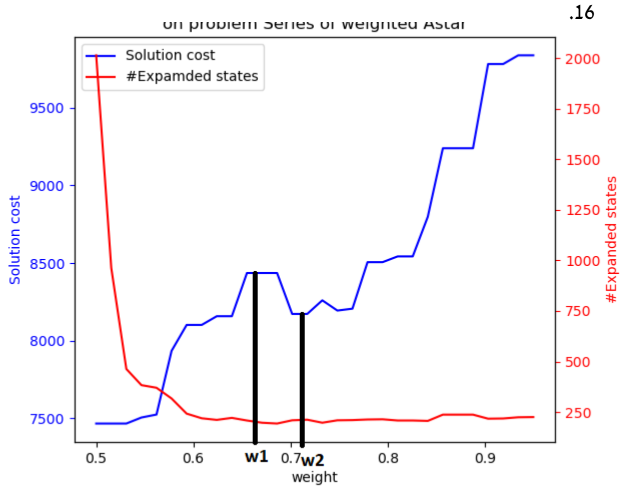
.14

```
StreetsMap(src: 54 dst: 549)          A* (h=0, w=0.500)           time:   1.15   #dev: 17354
StreetsMap(src: 54 dst: 549)          A* (h=AirDist, w=0.500)     time:   0.20   #dev: 2015
```

$\nu = \frac{17354-2015}{17354} = \frac{15339}{17354} = 0.8838884407$

חסכנו הרבה!

.16



—גרף מתאר יחס בין משקל w לבין מחיר של פתרון הבעיה ומספר מצבים מפותחים בריצת אלגוריתם $Weighted\ Astar$ למציאת פתרון בעיית $small\ toy\ problem$. איזור כדאי הוא $0.55 \leq w \leq 0.6$ כי בו מצליחים למזער את מחיר הפתרון ואת משאבי חיפוש הפתרון.

—התנהגות שראוים מתאימה לכלל אצבע שלמדנו, עבור $w=1$ אלגורימים מתחשב רק בפונקציה יוריסטית, וכל פעם בוחר מצב הבא לפיתוח כך שיהיה הכי קרוב למטרה. ולכן מצפים שיפתח פחות מצבי חיפוש ובגלל שלא בודק הרבה אפשרויות כנראה יהיה יקר יותר. לעומת זאת עבור $w=0.5$ מתחשב באופן אחיד בין פונקציה יוריסטית לבין מחיר של הפתרון, התחשבות לאורך הפתרון גורמת לו לפתח יותר מצבים כדי לנסות למצוא פתרונות בעלי מחיר יותר נמוך. ככל ש w יגדל אל 1 אז מחיר יגדל ומשאבי ריצה יוקטנו.

ניתן לראות בעולם האמיתי כלל אצבע לא תמיד פועל, לדוגמא נקודות $w_1 < w_2$ אבל מחיר של פתרון עבור w_1 גדול יותר ממחיר של פתרון עבור w_2, מצבים מפותחים נשארו בסדר גודל דומה.

חלק ו

.19

החיסרונות בגישה זו הם הגדלת מספר המצבים במרחב החיפוש שנצטרך לתחזק ברשימות $Close$ ו־$Open$, מקדם ההסתעפות יגדל דבר שיפגע בצריכת הזיכרון וזמן הריצה של האלגוריתם. בנוסף לא נוכל לנצל בצורה מיטבית את ה־ $cachedMapDist$, בפתרון הנוכחי אנו שומרים מרחק בין כל שני מיקומים שעברנו בהם ובכך חוסכים בחישובים נוספים של אותו מרחק. אם נשלב בין שתי הבעיות יהיה לכך פחות משמעות כי האלגוריתם כל פעם יפתח את הצומת הישיגה הקרובה מהצומת הנוכחית.

.20

$:(i)$

```
@dataclass(frozen=True)
class MDAState(GraphProblemState):
```

(ii): זה לא מספיק כי זה רק מבטיח שאי אפשר יהיה לשנות את השדות של איובייקט מטיפוס המחלקה. חלק משדות המחלקה הם set שהם $mutable$ וניתן לשנותם אחרי אתחול.

לכן נרצה שדשות האלו יהיו מטיפוס $frozenset$ שהוא $immutable$.

(iii): ייתכן שזה יקרה כי אנחנו מחפשים את המצב ברשימות $Close$ ו־$Open$ וייתכן הוא כבר נמצא שם, שורות מההרצאה:

$old - node \leftarrow findstate(OPEN, s)$

$old - node \leftarrow findstate(CLOSE, s)$

(iv): אנחנו רוצים ש־ $MDAState$ יהיה $hashable$ כדי לשמור אובייקטים מטיפוס זה ב־ set או כמפתח ל־ $dictionary$. בנוסף זה הכרחי לנכונות האלגוריתם שבטעות אף מצב לא ישתנה בזמן ריצה שלא לצורך ולכן זה מגן עלינו.

בהינתן מצב $prev$ בפונקציה $expand\_state\_with\_costs$ אנו יוצרים מצב $succ$. אם נבצע $succ.tests\_transferred = prev.tests\_transferred$ אזי מצב $succ$ ומצב $prev$ משתפים את אותו אובייקט מטיפוס set, כל מצב וצאצא שלו משתפים אותו set וכאשר אחד מהם עושה שינוי ל־ set אז set של המצב השני גם משתנה זה שגורם לאלגוריתם להיות שגוי.

23. טענה נכונה, נוכיח:

תהי קבוצה של צמתים $juncs = \{v_1, v_2, ...v_k\}$ בהם יש לעבור כולל צומת נוכחי שנסמנו v_{curr}.

נסמן זוג צמתים בעלי מרחק אווירי מקסימלי ב־v_i, v_j. נסמן מרחק אווירי בינהם $D_{Air}(v_i, v_j)$ אזי $D_{Air}(v_i, v_j) \geq 0$. כלומר $h = D_{Aur}(v_i, v_j) \geq 0$.

נניח כי קיים פתרון. נסמן ב־P מסלול אופטימלי מצומת v_{curr} אל צומת מטרה לפי $cost^{dist}_{MDA}$ אזי $h^* = |P|$.

נראה ש־$h \leq h^*$: v_i, v_j בהכרח נמצאים ב־P, נניח בה"כ כי v_i בא לפני v_j ב־ P, נסמן את תת־מסלול שלו מ־v_i אל v_j ב־P'.

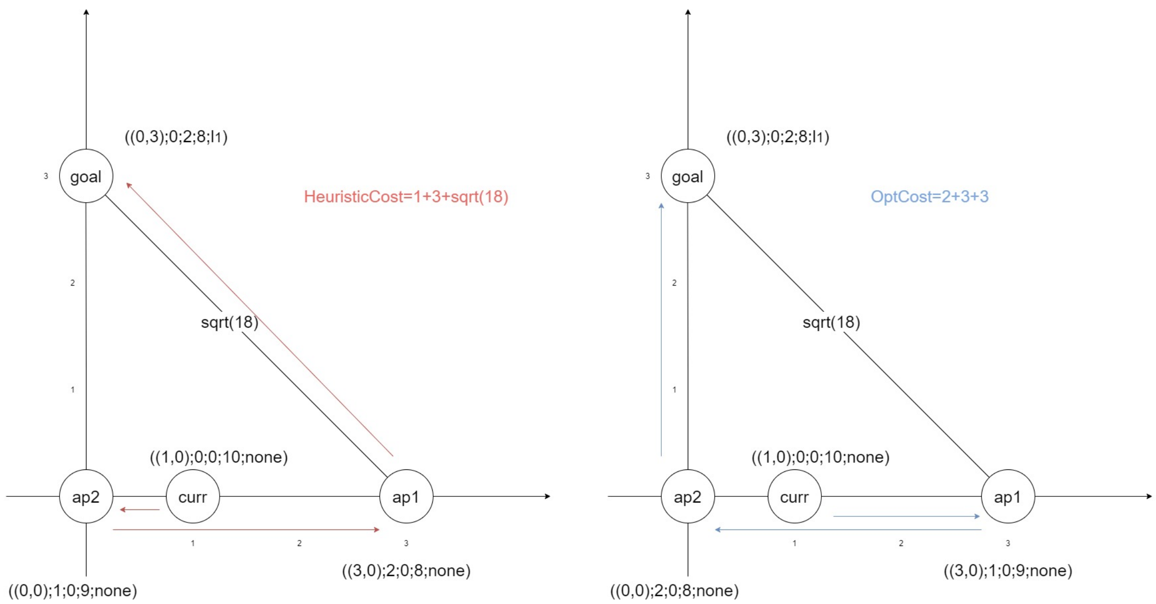
לפי משפט אי־שוויון המשולש שמתקיים במרחב אוקלידי R^2 נקבל: $|P| \geq |P'| = |path(v_i, v_j)| \geq D_{Air}(v_i, v_j)$.

אם פתרון לא קיים אזי $h^* = |P| = \infty$. בסה"כ נקבל: $h = D_{Air}(v_i, v_j) \leq |P'| \leq |P| \leq h^*$ כלומר $0 \leq h \leq h^*$ ולכן h קבילה.

26.הפרכה:

בהינתן מרחב של 4 צמתים: $curr, ap_1, ap_2, goal$ נראה שמסלול אופטימלי מ $curr$ אל $goal$ קצר יותר מאשר ערך יוריסטי של $MDASumAirDist$ עבור אותה בעיה:

נניח כי כל קשת בין שני צמתים היא כביש שהוא קו ישר בין הצמתים כלומר אורכו שווה למרחק האווירי.



עבור מצב $curr$ מתקיים כי $h^*(curr) = dist(curr, ap_1) + dist(ap_1, ap_2) + dist(ap_2, goal) = 2 + 3 + 3 = 8$ כאשר $h(curr) = air(curr, ap_1) + air(ap_1, ap_2) + air(ap_2, goal) = 1 + 3 + \sqrt{19} = 4 + \sqrt{18}$ מתקיים כי $h(curr) > h^*(curr)$ ולכן יוריסטיקה הנ"ל אינה קבילה.

29. נוכיח כי היוריסטיקה $MDAMSTAirDistHeuristic$ הינה קבילה:

בהינתן צומת נוכחי v_{curr} וקבוצת צמתים $S = \{v_1, v_2, ...v_k\}$ בהם יש עוד לעבור, נגדיר $V = I \cup \{v_{curr}\}$. נבנה G כך ש־ $G = (V, E)$, כך שלכל $u, v \in V$ שונים $(u, v) \in E$ ו־$w(u, v) = AirDist(u, v) \geq 0$. את העפ"מ של G נסמן ב־ $T = (V, E_T)$ אזי: $h = w(T) \geq 0$. נניח כי קיים פתרון לבעיה, ויהא P מסלול אופטימלי לפי $cost^{dist}_{MDA}$ מ־v_{curr} אל צומת מטרה. את המסלול אפשר להראות בצורה הבאה: $P' = u_1 u_2 ... u_k$ כך שכל u_i מסמן צומת של דירה והחסרנו מצב מטרה שהוא מעבדה, בהכרח $V \subseteq P'$ כי P פתרון.

לכל $1 \leq i \leq k-1$ ב־P' מתקיים שקיימת קשת $e = (u_i, u_{i+1}) \in E$ כך ש־ $w(e) \leq dist_P(u_i, u_{u+1})$, זה נובע מאי־שוויון המשולש. נסמן את קבוצת קשתות הנ"ל ב־E'.

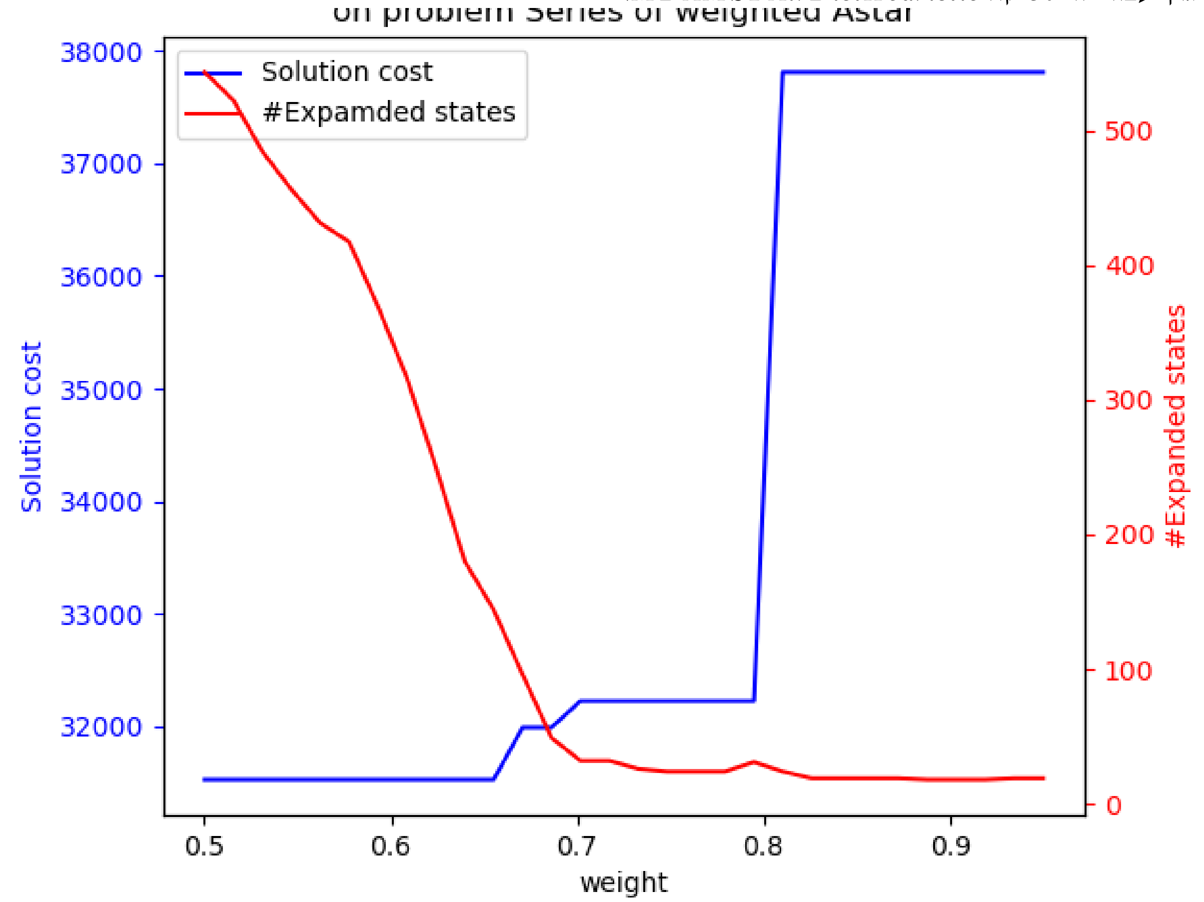
לכן $h^* = |P| \geq |P'|$. מתקיים כי $G' = (V, E') \subseteq G$ הינו גרף קשיר ו־ $|E'| = k-1$ ולכן G' הינו עץ פורש של T, ולכן $w(T) \leq w(G')$. אזי:

$h = w(T) \leq w(G') = \sum_{i=1}^{k-1} w(u_i, u_{i+1}) \leq \sum_{i=1}^{k-1} dist_P(u_i, u_{i+1}) = |P'| \leq h*$.

אם פתרון לא קיים אז $h^* = |P| = \infty$ ולכן גם במקרה זה $h \leq h^*$ כי מרחקים אוויריים סופיים. לבסוף $0 \leq h \leq h^*$.

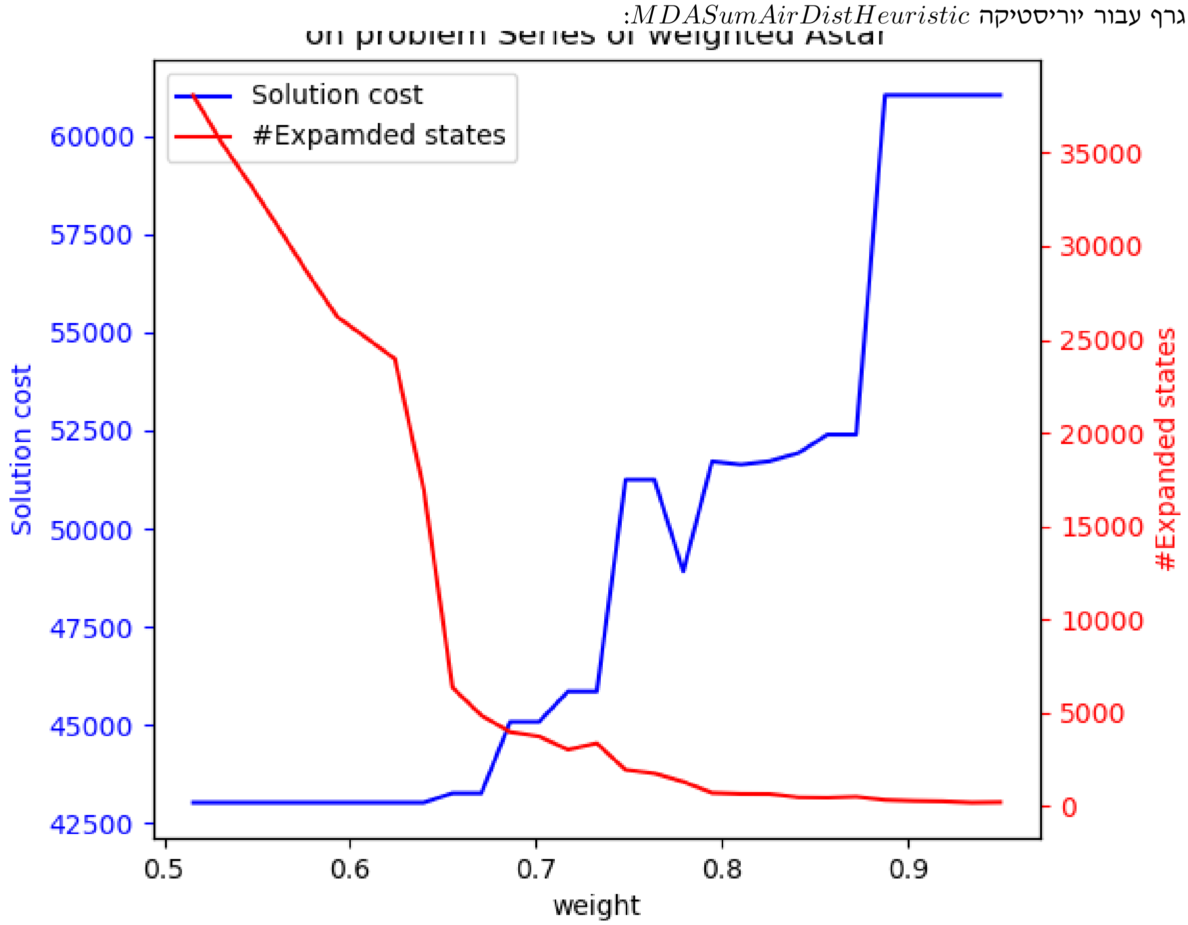
30.

גרף עבור יוריסטיקה $MDAMSTAirDistHeuristic$:



האזורים בהם גם $solution\_cost$ וגם $expande\_nodes$ ביחד קטנים הם האזורים העדיפים, כלומר עבור $0.66 \leq weight \leq 0.8$.

גרף עבור יוריסטיקה $MDASumAirDistHeuristic$:



האזור הכדאי הינו $0.68 \leq weights \leq 0.74$ כי גם מחיר של הפתרון וגם מספר צמתים מפותחים יחסית נמוכים.

חלק ז

31.הנחות:

־בכל דירה יש לפחות דייר אחד.

־$gasPrice$ יכול להיות קטן ככל האפשר כדי ליצור תוצאה סופית של $cost_{MDA}^{monetary}$ קטנה מספיק כך שהיוריסטיקות כבר לא יהיו אופטימיות.

	$MDAMaxAirDistHeuristic$	$MDASumAirDistHeuristic$	$MDAMSTAirDistHeuristic$
$cost_{MDA}^{test\_travel}$	לא	לא	לא
$cost_{MDA}^{monetary}$	לא	לא	לא

32.פלט עבור בעיה $small$ ו $moderate$ עם $cost_{MDA}^{dist}$ בהתאמה:

```
total_cost: MDACost(dist=  31528.659m, money=     49.717NIS, tests-travel=  52112.429m)
```

```
total_cost: MDACost(dist=  43034.794m, money=     95.847NIS, tests-travel= 176505.013m)
```

פלט עבור בעיות $small$ ו $moderate$ עם $cost_{MDA}^{money}$ בהתאמה:

```
total_g_cost:    42.04962   total_cost: MDACost(dist=  31923.809m, money=     42.050NIS, tests-travel=  53317.118m)
total_g_cost:    77.20101   total_cost: MDACost(dist=  54951.037m, money=     77.201NIS, tests-travel= 172922.318m)
```

ניתן לראות כי $monetary\_cost$ קטן בפלטים של ריצות עם $optimization\_cost = monetary$ מאשר בפלטים של ריצות עם $optimization\_cost = distance$.

33.הוכחה:

אם קיים פתרון אז נסמן את מסלול אופטימלי לפי $cost_{MDA}^{travel}$ מצומת v_{curr} אל צומת המטרה ב־P. תהי קבוצה של דירות בהן יש לעבור: $S = \{v_1, v_2 ... v_k\}$

וקבוצת מעבדות $L = \{u_i, u_2 ... u_m\}$ אזי: לכל דירה v_i קיימת מעבדה u_j אליה הועברו טסטים מדירה זו, בנוסף נסמן ב־ v_j^* את המעבדה הקרובה ביותר לדירה v_i לפי מרחק אווירי.

נסמן גם בn_i את מספר הדיירים של הדירה. אזי: $h = \sum_{i=1}^{k} airdist(v_i, u_j^*) \leq \sum_{i=1}^{k} airdist(v_i, u_j) \underset{triangle}{\leq} \sum_{i=1}^{k} dist_P(v_i, u_j) \leq \sum_{i=1}^{k} n_i \cdot dist_P(v_i, u_j) = cost_{MDA}^{travel} = h^*$

אם פתרון לא קיים אזי $h = \infty$ וכי מרחק אווירי הינו ערך סופי, ולכן $h \leq h^*$. בנוסף לזה מרקח אווירי הינו ערך אי שלילי ולכן: $0 \leq h \leq h^*$ כלומר פונקציה יוריסטית קבילה.

.35

פלט עבור בעיה $moderate$ עם $cost_{MDA}^{travel}$ ניתן להשוות עם פלט עבור אותה בעיה עם $cost_{MDA}^{dist}$ או $cost_{MDA}^{money}$ שנמצא בסעיף 32 כי אכן קיבלנו תוצאה קטנה יותר עבור $cost_{MDA}^{test\_travel}$:

```
total_cost: MDACost(dist=  93226.428m, money=    127.199NIS, tests-travel= 131265.153m)
```

.36

הוכחה: אם קיים פתרון במרחב המקורי S_{MDA} עם פונקציית עלות $cost_{MDA}^{dist}$ מצומת s_0 אל צומת מטרה $s_{t'}$ עבור $t' \in \mathbb{N}$ נסמן מסלול זה ב־ $P = s_0 \overset{o_0}{\rightarrow} s_1 \overset{o_1}{\rightarrow} ... s_{t'-1} \overset{o_{t'-1}}{\rightarrow} s_{t'}$. יהי $\epsilon \geq 0$ וC_{dist}^* אורך מסלול שנפלט מהרצת A^*על S_{MDA}. נתון שיוריסטיקה קבילה ולכן C_{dist}^* הינו אורך מסלול אופטימלי לפי $cost_{MDA}^{dist}$.

נניח בשלילה כי אלגוריתם A_1 לא יחזיר פתרון, ונסתכל על סדרת המסלולים הבאה:

$< s_0 >, < s_0 \overset{o_0}{\rightarrow} s_1 >, ..., < s_0 \overset{o_0}{\rightarrow} s_1 \overset{o_1}{\rightarrow} ... s_{t'-1} \overset{o_{t'-1}}{\rightarrow} s_{t'} >$ כל מסלול בסדרה מקיים $p \in S^P$ לפי הגדרה, ולכל $0 \leq k < t$ מתקיים $o_k \in O^P$ בנוסף $s_0 \in < I >$ ו־ $s_{t'} \in G$ ולכן $< s_0 \overset{o_0}{\rightarrow} s_1 \overset{o_1}{\rightarrow} ... s_{t'-1} \overset{o_{t'-1}}{\rightarrow} s_{t'} > \in G^P$ ראינו כי אלגוריתם UCS שלם אם פונק' המשקל חסומה מלמטה ומתקיים $1 \cdot m_0 \leq cost_{MDA}^{test\ travel}(s_t, o) = \left[\sum_{d \in s.Taken} d.roommates \right] \cdot cost_{MDA}^{dist}(s_t, o)$ כאשר m_0 הינו המרחק הקצר ביותר בין שני צמתים שונים (זה מוגדר היטב כי יש מספר קומבינציות סופי). כלומר שלב ii יחזיר פתרון בסתירה להנחה.

.37

הוכחה: נסתכל על שלב ii: יהי $\epsilon \geq 0$ וC_{dist}^*אורך מסלול שנפלט מהרצת A^*על S_{MDA}. נתון שיוריסטיקה קבילה ולכן C_{dist}^*הינו אורך מסלול אופטימלי לפי $cost_{MDA}^{dist}$.

נוכיח באינדוקציה כי בכל איטרציה אנו מוסיפים ל־$open$ מסלול $p = s_0 \overset{o_0}{\rightarrow} s_1 \overset{o_1}{\rightarrow} ... s_{t-1} \overset{o_{t-1}}{\rightarrow} s_t$ שעבורו מתקיים $p \in DistEpsOptimalPaths$ ו$cost(p) = cost_{MDA}^{travel}(s_{t-1}, o_{t-1})$:

בסיס: איטרצית אפס:אנו מוסיפים ל$open$ את $p = s_0$ כלומר $cost_{MDA}^{dist}(p) = 0 < (1+\epsilon)\cdot C_{dist}^{*}$ ולכן $p \in DistEpsOptimalPaths$.

צעד: נניח נכונות עבור כל איטרציה $i \le k$ ונוכיח עבור איטרציה $i = k+1$: נחלק למקרים: -נניח כי עבור מסלול p שהוצא מ-$open$ לא נוצר אף מסלול עוקב אזי

הטענה מתקיימת באופן ריק. מהנחה ידוע כי $open$ לא ריק ואלגוריתם ממשיך. -נניח עבור מסלול $p = s_0 \xrightarrow{o_0} s_1 \xrightarrow{o_1} \dots s_0 \xrightarrow{o_{t-1}} s_t$

שהוצא מ$open$ ועבור אופרטור

$o \in O_{MDA}$ כך ש: $p' \in Domain(o^p)$ נוצר מסלול עוקב $p' = o^p(p)$ אזי אם

$cost_{MDA}^{dist}(s_t, o) + \sum_{i=0}^{t-1} cost_{MDA}^{dist}(s_i, o_i) \le (1+\epsilon)\cdot C_{dist}^{*}$

אזי $cost(p, o^p) = cost_{MDA}^{test\_travel}(s_t, o)$ סופי ולכן מכניסים מסלול p' ל$open$. אחרת $cost(p, o^p) = \infty$ ומסלול לא נכנס ל$open$.

מהטענה הנ"ל ומקבילות של UCS

נובע כי אלגוריתם יחזיר מסלול אופטימלי לפי $cost_{MDA}^{test\_travel}$ מבין כל המסלולים שאורך שלהם לא עולה על $(1+\epsilon)\cdot C_{dist}^{*}$.

כלומר מוחזר פתרון אופטימלי לפי קריטריון המשולב.

.38

רואים כי באמת נשמר איזון בין המדדים. $test\_travel$ לא גדל משמעותית אבל $dist$ כן קטן משמעותית.

```
total_cost: MDACost(dist=  65686.522m, money=     99.486NIS, tests-travel= 132209.981m)
```

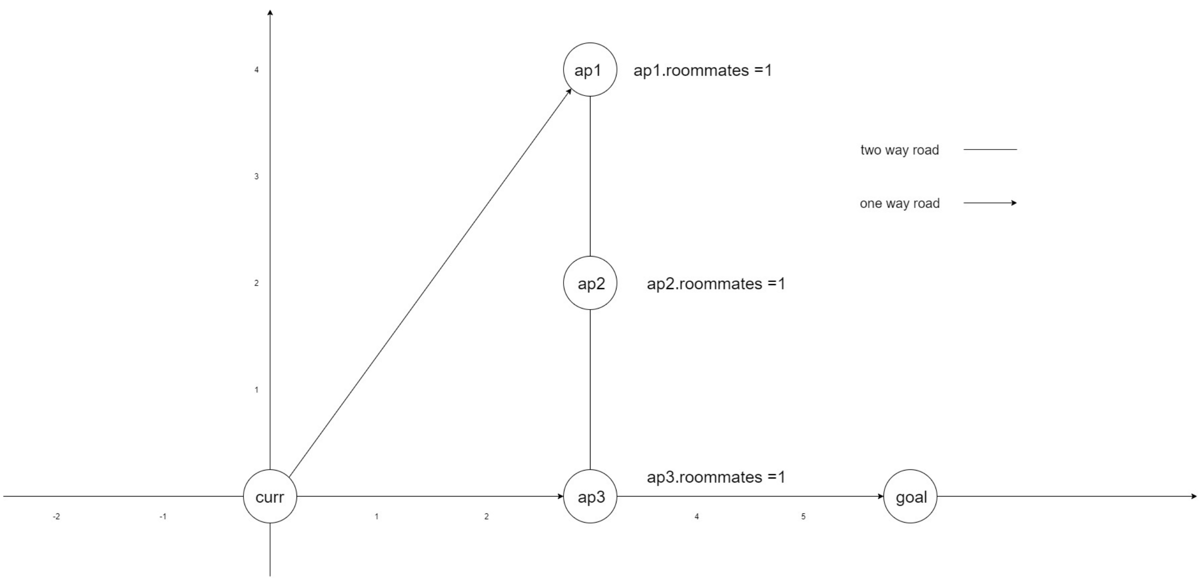
$\frac{distCost}{C^*} - 1 = 0.5263584583037706 < eps = 0.6$.

.39 הפרכה:

נראה שקיים פתרון במרחב מ- $curr$ אל $goal$ אבל A_2 לא מחזיר פתרון עבור אותה בעיה:

נניח כי כל קשת בין שני צמתים היא כביש שהוא קו ישר בין הצמתים כלומר אורכו שווה למרחק האווירי.

מתקיים כי $C_{dist}^{*} = 12$ נבחר $\epsilon = 0.01$ כך ש- $(1+\epsilon)C_{dist}^{*} = 12.12$ שלבי ריצת האלגוריתם :



טבלת מעקב:

מסלול עד לצומת הנוסף	g	$close$	$open$	צומת נשלף מ- $open$ בתחילת הריצה	שלב ריצה באלגוריתם UCS
	0		$[curr]$		אתחול $open$ ומצב התחלתי
		$[]$			אתחול $close$
				$curr$	
$curr$		$[curr]$			
$curr \to ap1$	0	$[curr]$	$[ap1]$		יצירת צומת ל- $ap1$ והכנסתה ל- $open$ $((3,4),1,0,9,\emptyset)$
$curr \to ap3$	0	$[curr]$	$[ap1, ap3]$		יצירת צומת ל- $ap3$ והכנסתה ל- $open$ $((3,0),1,0,9,\emptyset)$
$curr \to ap2$	0	$[curr]$	$[ap1, ap3, ap2]$		יצירת צומת ל- $ap2$ והכנסתה ל- $open$ $((3,2),1,0,9,\emptyset)$
				$[ap3]$	
$curr \to ap1'ap3 \to$	4	$[curr, ap3]$	$[ap1, ap2, ap1']$		יצירת צומת ל- $ap1'$ והכנסתה ל- $open$ $((3,4),2,0,8,\emptyset)$
$curr \to ap2'ap3 \to$	2	$[curr, ap3]$	$[ap1, ap2, ap1', ap2']$		יצירת צומת ל- $ap2'$ והכנסתה ל- $open$ $((3,2),2,0,8,\emptyset)$
$curr \to goalap3 \to$	3	$[curr, ap3]$	$[ap1, ap2, ap1', ap2', goal]$		יצירת צומת ל- $goal$ והכנסתה ל- $open$ $((6,0),0,1,9,l_1)$
				$[ap1]$	
		$[curr, ap3, ap1]$	$[ap2, ap1', ap2', goal]$		מוצא את $ap2'$ ולא צריך לעדכן מחיר

$curr \rightarrow ap3' ap1 \rightarrow$	4	$[curr, ap3, ap1]$	$[ap2, ap1', ap2', goal, ap3']$		יצירת צומת ל- $ap3'$ והכנסתה ל- $open$ $((3,0),2,0,8,\emptyset)$
		$[curr, ap3, ap1]$	$[ap2, ap1', ap2', goal, ap3']$		מוצא את $goal$ ולא צריך לעדכן מחיר
				$[ap2]$	
		$[curr, ap3, ap1, ap2]$	$[ap1', ap2', goal, ap3']$		מוצא את $ap1'$ ומעדכן את המחיר שלו ל- 2
		$[curr, ap3, ap1, ap2]$	$[ap1', ap2', goal, ap3']$		מוצא את $ap3'$ ומעדכן את המחיר שלו ל- 2
		$[curr, ap3, ap1, ap2]$	$[ap1', ap2', goal, ap3']$		מוצא את $goal$ ולא צריך לעדכן מחיר
				$[ap1']$	
$curr \rightarrow ap1' \rightarrow ap3 \rightarrow ap2''$	8	$[curr, ap3, ap1, ap2, ap1']$	$[ap2', goal, ap3', ap2'']$		יצירת צומת ל- $ap2''$ והכנסתה ל- $open$ $((3,2),3,0,7,\emptyset)$
		$[curr, ap3, ap1, ap2, ap1']$	$[ap2', goal, ap3', ap2'']$		יצירת צומת ל- $goal'$ ולא מכניסה בגלל מרחק $((6,0),0,2,8,l_1)$
				$[ap3']$	
		$[curr, ap3, ap1, ap2, ap1', ap3']$	$[ap2', goal, ap2'']$		מוצא את $ap2''$ ולא צריך לעדכן מחיר
$curr \rightarrow ap3' \rightarrow ap1 \rightarrow goal'$	10	$[curr, ap3, ap1, ap2, ap1', ap3']$	$[ap2', goal, ap2'', goal']$		יצירת צומת ל- $goal'$ והכנסתה ל- $open$ $((6,0),0,2,8,l_1)$
				$[ap2']$	
$curr \rightarrow ap2' \rightarrow ap3 \rightarrow ap1''$	6	$[curr, ap3, ap1, ap2, ap1', ap3' , ap2']$	$[goal, ap2'', goal', ap1'']$		יצירת צומת ל- $ap1''$ והכנסתה ל- $open$ $((3,4),3,0,7,\emptyset)$
				$[goal]$	
		$[curr, ap3, ap1, ap2, ap1' , ap3', ap2', goal]$	$[ap2'', goal', ap1'']$		אין צמתים לפיתוח צומת בור

				$[ap1'']$	
		$[curr, ap3, ap1, ap2, ap1', ap3'$ $, ap2', goal, ap1'']$	$[ap2'', goal']$		יצירת צומת ל- $goal''$ ולא מכניסה בגלל מרחק $((6,0),0,3,7,l_1)$
				$[ap2'']$	
		$[curr, ap3, ap1, ap2, ap1', ap3'$ $, ap2', goal, ap1'', ap2'']$	$[goal']$		יצירת צומת ל- $goal''$ ולא מכניסה בגלל מרחק $((6,0),0,3,7,l_1)$
				$[goal']$	
		$[curr, ap3, ap1, ap2, ap1', ap3'$ $, ap2', goal, ap1'', ap2'', goal']$	$[]$		אין צמתים לפיתוח צומת בור

$open$ ריקה ולכן האלגוריתם הסתיים ללא פתרון.

.40

הוכחה: מההנחה של השאלה נובע כי A_2מחזיר פתרון אזי:

בשלב i A^* רץ עם פונקציה $cost^{dist}_{MDA}$ ועם יוריסטיקה קבילה ולכן C^*_{dist}המוחזר הינו אורך מסלול אופטימלי של הבעיה.

נניח בשלילה כי הפתרון שמובטח מההנחה אינו אופטימלי לפי הקירטריון משולב.

בשלב UCS ii רץ על פונקציה $cost^{test\_travel}_{MDA}$ ומקבילות של UCS נובע כי הוא ימצא את הפתרון האופטימלי לפי פונקציה הנ"ל.

מההנחה בשלילה נובע כי הפתרון המוחזר p אינו שייך לקבוצה $DistOptimalPaths$ ולכן $cost^{dist}_{MDA}(p) > (1+\epsilon)\cdot C^*_{dist}$. אחרת הוא היה מינימלי לפי $cost^{travel}_{MDA}$ ושייך ל$DistOptimalPaths$ ולכן

אופטימלי לפי קריטריון המשולב. עבור מסלול p קיים תת מסלול $p' = s_0 \to s_1 \to ...s_{t-1} \to s_t$ עבורו בפעם ראשונה מתקיים ש$cost^{dist}_{MDA}(p') > (1+\epsilon)\cdot C^*_{dist}$, כלומר כשנוצר צומת s_t

העלות שלו היית גדולה מ$(1+\epsilon)\cdot C^*_{dist}$ ולכן לפי אופן פעולת האלגוריתם לא היינו מוסיפים אותו ל$open$ אבל מעובדה ש $s_t \in p$ שהוא מסלול פתרון מתקיים שכן הוספנו את s_tאל $open$ וזו סתירה.

.41

מרחב חיפוש של $A1$ הוא הרבה יותר גדול מזה של $A2$ כי בשביל $A2$ כל צומת מכיל מצב בודד וכל צומת מ$A1$ הינו מסלול ממצב התחלתי אל מצב כלשהו.

קומבינטורית מקבוצה של מצבים בגודל מסוים ניתן להרכיב קבוצה של מסלולים בגודל הרבה יותר משמעותי. ולכן זה משפיע באופן ישיר על זמן הריצה.

.44

```
MDA(small_MDA(5):Distance)          A* (h=MDA-MST-AirDist, w=0.500)   time:   9.65   #dev: 543
MDA(small_MDA(5):Distance)          A*eps (h=MDA-MST-AirDist, w=0.500)   time:   1.14   #dev: 492
```

ניתן לראות כי חסכנו $\frac{543-492}{543} * 100 \simeq 9.4\%$ פיתוחי צמתים. מכיוון שצומת הבא לפיתוח נבחר באוסף צמתים יותר גדול וצומת הבא

לפיתוח נבחר לפי ערך של יוריסטיקה לא קבילה

יוריסטיקה זאת נותנת לחלק מהצמתים ערך שהוא גדול יותר מהמרחק האמיתי לכן כך אנחנו מתקרבים יותר לצומת מטרה.

חלק י

א) זיכרון:

בכל איטרציה שלIDA^* מתבצע חיפוש לעומק ולכן דרישת זיכרון הינה תמיד לינארית בעומק הפתרון אם הוא קיים.

כאשרA^*רגיל מחזיק כל צומת שפותח באחד הרשימות - $open, closed$ ולכן דורש הרבה יותר זיכרון.

ב)

i: צמתים מפותחים.

ii: ייתכן שאלגוריתם יבצע הרבה איטרציות של חיפוש לעומק וכל פעם הוא יוצר כל צמתים מחדש כי הוא לא שומר אותם באף מקום.

iii: זה מתנהג שונה מהשוואה ל$ID-DFS$ כי שם מאיטרציה לאיטרציה ההגבלת עומק של ריצה בודדת גדלה במספר קבוע -1.

במקרה של $IDASTAR$ ייתכן שערכי פונקציה f לכל הצמתים מקבלים ערכים אחידים ומאיטרציה לאיטרציה אנחנו מגדילים את $flimit$ במספר מאוד קטן ולכן עושים הרבה איטרציות בכללי.

ג)

i: חסם עליון על מספר איטרציות: $\lceil (C_S^* - Q_k(h(I))) \cdot k \rceil$. נשים לב כי מאיטרציה לאיטרציה של אלגוריתם A_1אנחנו מגדילים את $Flimit$ ב-$\frac{1}{k}$ או בערך שיותר גדול ממנו. $Flimit$ ההתחלתי הינו $Q_k(h(I))$, וכאשר באיטרציה כלשהי $Flimit$ בפעם הראשונה נהיה גדול-שווה מC_S^* אזי באיטרציה זו הריצה של A_1 מסתיימת. ולכן מספר איטרציות הגדול ביותר הוא:

$$\lceil \frac{C_S^* - Q_k(h(I))}{\frac{1}{k}} \rceil = \lceil (C_S^* - Q_k(h(I))) \cdot k \rceil$$

ii: נסתכל על האיטרציה האחרונה של A_1, בה בפעם הראשונה מתקיים $Flimit \geq C_S^*$ ולכן $prevFlimit < C_S^*$ בנוסף $Flimit \in A_k$. מכיוון שיוריסטיקה קבילה $origFlimit = C_S^*$.

מתקיים: $Flimit = max\{prevFlimit + \frac{1}{k}, Q_k(C_S^*)\} \geq C_S^*$ כלומר $Flimit = min\{a | a \in A_k \wedge a \geq C_S^*\} = \lceil \frac{C_S^*}{1/k} \rceil \cdot \frac{1}{k}$ אזי נוכל למצוא פתרון במחיר הנ"ל ולכן:

$$\varepsilon(A_1, S) = \lceil \frac{C_S^*}{1/k} \rceil \cdot \frac{1}{k} - C_S^*$$