236501 - מבוא לבינה מלאכותית

תרגיל בית 2: *Heuristic Search*

שאלות

לצורך תרגיל זה השתמשנו בגרסאות העדכניות של כל הקבצים. לצורך כך יצרנו מחדש את קובץ המרכזים, את המרחבים האבסטרקטיים ואת $dataSet$.

חלק ב' - מימוש אלגוריתם A*

1.

נגדיר $h_{1,B}(s)$ המרחק האווירי, המחושב ע"י $compute\ distance$ (שבקובץ $ways/tools.py$, מומר ליחידות מטרים, כך שנוכל להשוות לפונקציית המחיר שלנו) בין A, הצומת שמספרה הוא s, לבין צומת היעד - B.
היוריסטיקה קבילה מכיוון שמרחק אווירי הוא מטריקה, ולכן מקיים אי שליליות, ואת אי שוויון המשולש. כלומר לכל $s \in S$ מתקיים $h_{1,B}(s) \geq 0$.
וכן $h(s) \leq h^*(s)$ (כאשר $h^*(s)$ הוא המחיר האופטימלי - כלומר סכום אורכי הכבישים של המסלול האופטימלי, ולכן מאי שוויון המשלוש מתקיים $h(s) \leq h^*(s)$).

2.

היוריסטיקה שהגדרנו קבילה. לפי משפט שראינו בהרצאה, A^* המשתמש בה קביל גם הוא. כלומר המסלול שאנו מקבלים יהיה הזול ביותר מבין המסלולים.

3.

מימוש הפונקציה a_star.

חלק ג' - בניית אלגוריתם החיפוש $(A*)^3$

4.

מימוש הפונקציה a_star_exp3.

5.

אלגוריתם A^* משתמש ביוריסטיקה לשם הערכת מרחק (בגרף החיפוש. כלומר עלות הדרך אל המטרה) אל המטרה. כלומר היוריסטיקה תלויה במטרות האפשריות. כאשר מטרה אחת ידועה, זה פשוט. כאשר המטרה לא ידועה, זה כבר לא טריוויאלי. למשל במקרה שלנו, כאשר אנו מחפשים מרכז קרוב ביותר (בעל עלות נמוכה ביותר) הדבר היחיד שאנו יודעים על המטרות האפשריות הוא שהן מרכזים. אפשר לחשוב על יוריסטיקה שלכל צומת בגרף מחשבת את המרחק המינימלי לכל המרכזים האפשריים. זה כמובן כבר לא טריוויאלי, וגם לא יעיל. אפשר כמובן היה להשתמש ביוריסטיקת מרחק אווירי למטרה B, ולהחזיר את המרכז הראשון שאנו מגיעים אליו. צריך לבחון פתרון זה ולראות האם הוא אכן טוב יותר, מכיוון שייתכן ולא נעבור במרכז, או שנמצא מרכז זה בשלב מאוחר, כשכבר פיתחנו הרבה צמתים, כלומר בזבזנו זמן רב, וכן מזערנו את משקל המסלול $junc_1 - junc_2$ שהוא בעצם מטרת כל האלגוריתם הזה.

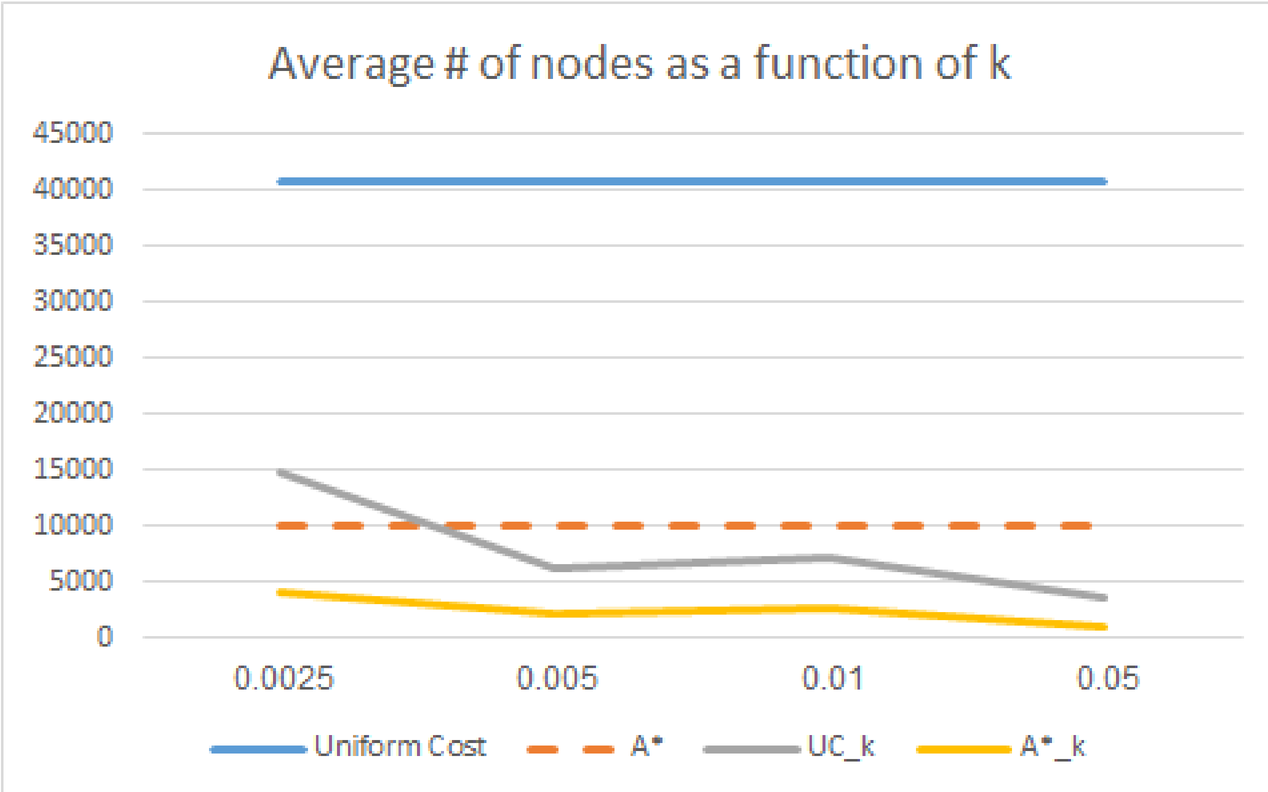
חלק ד' - בחינת האלגוריתמים A^* ו-$(A^*)^3$

6.

ביצוע הניסוי. נמצא ב-$db/experiment2.csv$ וניתוח הניסוי נמצא ב-$experiment2\_analysis.xlsx$.

.7

.א



.ב

k	Uniform Cost	A*	UC_k	A*_k
0.0025	40940.15	10149.35	14843.4	4036.1
0.005	40940.15	10149.35	6333.15	2309.95
0.01	40940.15	10149.35	7294.55	2601.2
0.05	40940.15	10149.35	3737.75	1101.1

.ג

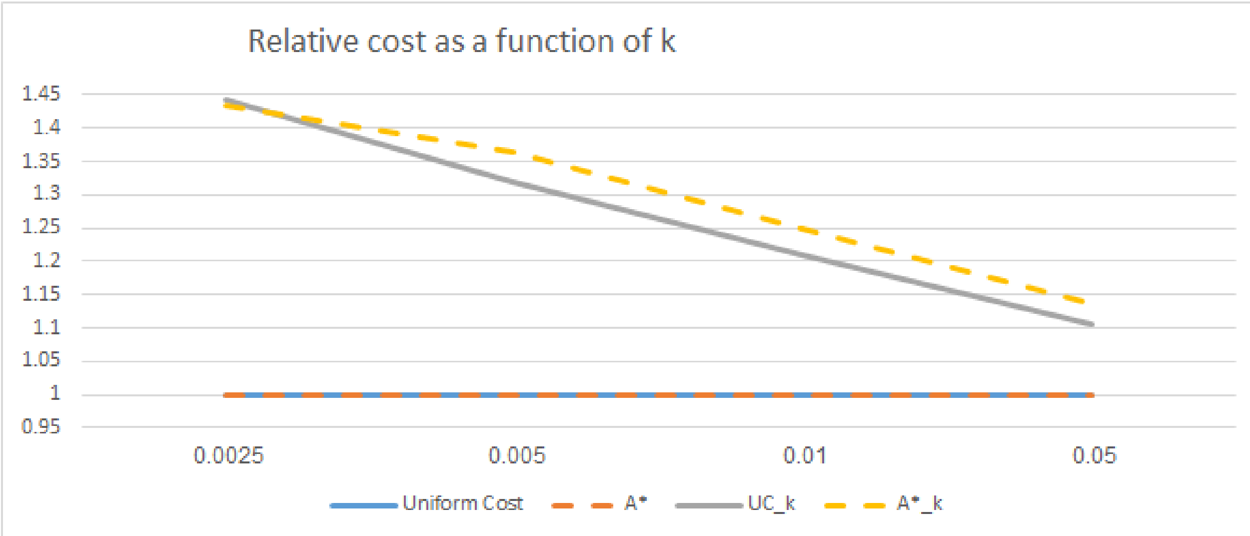
כצפוי, שימוש באלגוריתמים מיודעים שיפר את יעילות אלגוריתמי החיפוש. אנו משתמשים כאן במידע נוסף על הבעיה (מרחקים אוויריים), ומקבלים בתמורה מציאה מהירה יותר של המסלולים.

.ד

מגמות השיפור והדעיכה ביעילות של $(A^*)^3$ ו-UC^3 דומות. יש ירידה גדולה בין k_0 ל-k_1. אח"כ עליה קטנה בין k_1 ל-k_2. ובסוף ירידה גדולה בין k_2 ל-k_3. המגמות דומות מכיוון שהאלגוריתמים מאוד דומים. השינוי היחידי (חוץ משימוש ב-UCS לעומת A^* כמובן) הוא מציאת $junc_1$, כאשר ב-UC^3 אנו מוצאים אותו ע"י UCS, כצומת הקרובה ביותר (מבחינת מרחק נסיעה בכבישים), וב-$(A^*)^3$ אנו מוצאים אותו ע"י מרחק אווירי. הבדל זה די זניח, מכיוון שעיקר הרווח שלנו הוא החסכון בכך שאנו זוכרים את המסלול בין $junc_1$ ל-$junc_2$ (כפי שהסברנו בתרגיל הקודם). ההסבר למגמה (בשני האלגוריתמים) הוא שככל שהמרחב האבסטרקטי גדול יותר, אמנם נדרשנו ליותר חישובים בהתחלה, אך אנו מקבלים מרחב צפוף יותר, אשר נותן לנו מסלולים טובים יותר מהבחינה שמשקל המסלול $junc_1 - junc_2$ גדל, כלומר אנו חוסכים יותר. מצד שני, ככל שהמרחב האבסטרקטי גדל, החיפוש בו ארוך יותר, כלומר אנו מפסידים מעט בשלב מציאת המסלול $junc_1 - junc_2$. זה מה שמסביר את הקפיצה הקטנה בין 0.005 לבין 0.01. כפי שניתן לראות השפעה זו די זניחה עבור ערכי k שבדקנו. בנוסף, ככל שהמרחב גדול יותר, מספר השכנים גדול יותר, אז אולי גם פחות צריך לחפש בו. יש כאן איזון עדין שצריך לבדוק לעומק.

.8

.א



.ב

k	Uniform Cost	A*	UC_k	A*_k
0.0025	1	1	1.442777437	1.435082614
0.005	1	1	1.319825121	1.363145497
0.01	1	1	1.20942836	1.248605005
0.05	1	1	1.105811215	1.136301968

.ג

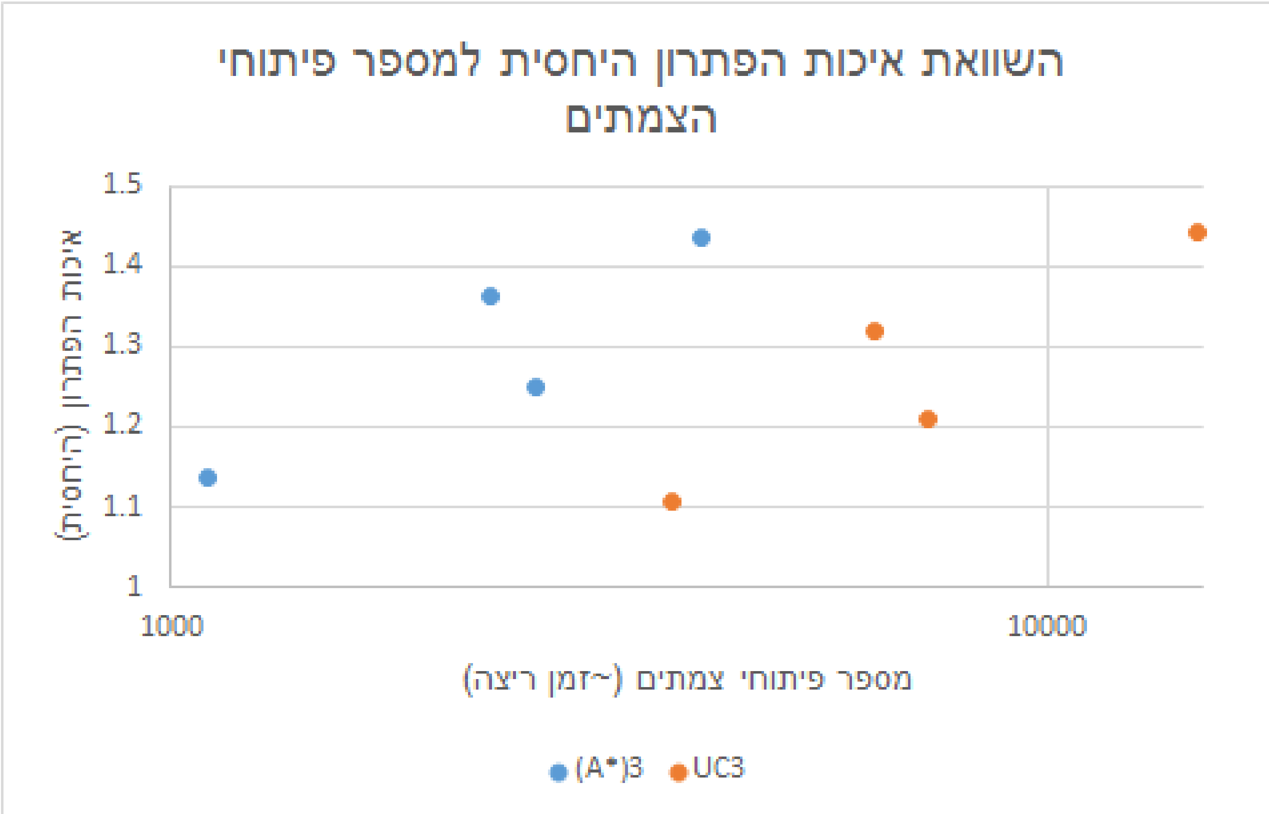
תחילה, מכיוון ש־A^* שלנו משתמש ביוריסטיקה קבילה, הוא קביל. כלומר עלות הפתרון זהה לעלות הפתרון של UCS (שכפי שראינו הוא גם קביל). בקשר לשני האלגוריתמים הנותרים ־ איכות הפתרון מאוד דומה בין שניהם (עבור כל ערך k כמובן), כצפוי, מכיוון ש־A^* עם יוריסטיקה קבילה ו־UCS מחזירים מסלול אופטימלי מבחינת עלות. השוני נובע מההבדל באלגוריתמים (מה שהזכרנו קודם עם שיטת מציאת $junc_1$). מבחינה זו אלגוריתם UC_k^3 טוב יותר (עלות הפתרון נמוכה יותר בממוצע), מכיוון שאת $junc_1$ אנחנו מוצאים בו ע"י UCS, כלומר קיבלנו את המרכז הכי קרוב מבחינת נסיעה. כלומר מתמזערת כאן העלות. לעומת זאת ב־$(A^*)_k^3$ את $junc_1$ אנו מוצאים ע"י קרבה מבחינת מרחק אווירי. כפי שרואים אין הרבה הבדל בעלות, מכיוון שבהרבה מקרים מרכז קרוב מבחינת מרחק אווירי, הוא גם קרוב מבחינת נסיעה. אך יש מקרים, כמו למשל בריצת $(A^*)_{0.05}^3$ על הזוג $30435, 89826$, ש־$junc_1$ שאנו מקבלים ממש מפריע לנו. במקרה זה הקטע בין 30435 ל־$junc_1$ הוא באורך 4000מטרים, לעומת זאת הכביש בין 30435 לבין $junc_1$ בריצת $UC_{0.05}^3$ הוא קטן בהרבה ־ 1600 מטרים.

.ד

מגמות השיפור והדעיכה ביעילות של $(A^*)^3$ ו־UC^3 דומות. יש ירידה גדולה בין כל זוג ערכי k. המגמות דומות מכיוון שהאלגוריתמים מאוד דומים. השינוי היחידי (חוץ משימוש ב־UCS לעומת A^* כמובן) הוא מציאת $junc_1$, כאשר ב־UC^3 אנו מוצאים אותו ע"י UCS, כצומת הקרובה ביותר (מבחינת מרחק נסיעה בכבישים), וב־$(A^*)^3$ אנו מוצאים אותו ע"י מרחק אווירי. הבדל זה די זניח, מכיוון שעיקר הרווח שלנו הוא החסכון בכך שאנו זוכרים את המסלול בין $junc_1$ ל־$junc_2$ (כפי שהסברנו בתרגיל הקודם). ההסבר למגמה (בשני האלגוריתמים) הוא שככל שהמרחב האבסטרקטי גדול יותר, עולה הסיכוי למצוא בממוצע מרכזים קרובים לצמתי המקור\יעד שלנו ולכן טובים יותר. המסלולים בין המרכזים הם אופטימלים, וככל שמשקלם במסלול גדול יותר, כך מורגשת יותר תרומתם. עבור צומת יעד למשל שהמרכז הקרוב אליה לא עובר באחד המסלולים האופטימלים, ואף אולי רחוק מאחד כזה, יגרום לכך שנסטה בהרבה ממסלול אופטימלי. ככל שמשקל המסלול $junc_1, junc_2$ גדול יותר, אנו מקבלים מסלול טוב יותר (משקל ־ כלומר חלקו במסלול הסופי).

9.

א.



ב.

$(A*)3\ or\ UC3$	איכות הפתרון (היחסית)	מספר פיתוחי צמתים (~זמן ריצה)
(A*)3	1.435082614	4036.1
(A*)3	1.363145497	2309.95
(A*)3	1.248605005	2601.2
(A*)3	1.136301968	1101.1
UC3	1.442777437	14843.4
UC3	1.319825121	6333.15
UC3	1.20942836	7294.55
UC3	1.105811215	3737.75

ג.

עבור כל אלגוריתם, ככל שזמן ריצת האלגוריתם קטן יותר, כך איכות הפתרון טובה (קרובה יותר ל-UCS רגיל). זה נובע ממספר המרכזים, וצפיפותם. כלומר הקשר הישיר כאן הוא ל-k, כמו בשאלות הקודמות. ככל ש-k גדול יותר (מבין הערכים שבחרנו, ומבחינת המגמה) איכות הפתרון וגם זמן הריצה טובים יותר.

חלק ה' - חיפוש עם פונקציית מחיר חדשה

.10

לכל $s_1, s_2 \in S$ כך שיש כביש $link\left(s_1, s_2\right)$ ביניהם, נגדיר (נגדיר את הפונקציה כמקבלת לינק ולא 2 צמתים. כפי שנאמר ב־FAQ שמותר):

$$expected\_time\left(link\left(s_1,s_2\right)\right) \triangleq \frac{\frac{link(s_1,s_2).distance[m]}{1000}}{current\_average\_speed\left(link\left(s_1,s_2\right)\right)\left[\frac{km}{h}\right]} = \frac{1}{1000}\cdot\frac{link\left(s_1,s_2\right).distance[km]}{current\_average\_speed\left(link\left(s_1,s_2\right)\right)\left[\frac{km}{h}\right]}$$

$$= \frac{1}{1000}\cdot\frac{link\left(s_1,s_2\right).distance}{current\_average\_speed\left(link\left(s_1,s_2\right)\right)}\left[h\right]$$

מכיוון ש־$distance$ ביחידות מטרים, נצטרך להמיר ל־ק"מ (חילוק ב־1000). כלומר קיבלנו את $expected\_time$ ביחידות של שעה, כנדרש.

.11

מימוש $expected\_time$ ב־$cost.py$.

.12

נגדיר $\left[h\right]$ $h_{cur,B}\left(s\right) \triangleq \frac{compute\_distance(s,B)[km]}{MAX\_LEGAL\_SPEED[\frac{km}{h}]} = \frac{compute\_distance(s,B)}{MAX\_LEGAL\_SPEED}$.

כמובן שהארגומנטים ש־$compute\_distance$ מקבלת שונים (מקבלת $lon\backslash lat$ משני הצמתים), אך הכוונה בכל מקרה היא למרחק האווירי בין s, B.
המהירות המקסימלית המותרת היא 110 (מופיע בקבצי התרגיל).
נסביר מדוע זו יוריסטיקה קבילה. תחילה מרחק הוא מטריקה לכן אי־שלילי. מהירות מקסימלית היא 110 - כלומר חיובית.
כמו כן מתקיים $h_{cur,B}\left(s\right) \le h^*\left(s\right)$ לכל s מכיוון שכל מסלול בין s, B יהיה לכל הפחות מאורך המרחק האווירי בין s, B (מעבר (1)). ראינו זאת בהוכחת הקבילות של היוריסטיקה הקודמת), וכן המהירות המקסימלית לאורך כל הקטע תהיה לכל היותר 110 (מעבר (2)).
לכן עבור כל מסלול P בין s, B המורכב מ־$links$ (סתם לשם הנוחות. אפשר לכתוב בייצוג של צמתים), מתקיים:

$$h_{cur,B}\left(s\right) = \frac{compute\_distance\left(s,B\right)\left[km\right]}{MAX\_LEGAL\_SPEED\left[\frac{km}{h}\right]} \overset{(1)}{\le} \frac{\sum\limits_{link\ in\ P} link.distance\left[km\right]}{MAX\_LEGAL\_SPEED\left[\frac{km}{h}\right]} = \sum_{link\ in\ P}\frac{link.distance\left[km\right]}{MAX\_LEGAL\_SPEED\left[\frac{km}{h}\right]}$$

$$\overset{(2)}{\le} \sum_{link\ in\ P}\frac{link.distance\left[km\right]}{current\_average\_speed\left(link\right)\left[\frac{km}{h}\right]} = \sum_{link\ in\ P}\frac{1}{1000}\cdot\frac{link.distance\left[m\right]}{current\_average\_speed\left(link\right)\left[\frac{km}{h}\right]}$$

$$= \sum_{link\ in\ P} expected\_time\left(link\right) = \sum_{link\ in\ P} cost\left(link\right) = cost\left(P\right)$$

זה נכון עבור כל מסלול P בין s, B, ולכן $h_{cur,B}\left(s\right) \le \underset{\substack{P\ is\ a\ path\\ between\ s,B}}{MIN}\left(cost\left(P\right)\right) = h^*\left(s\right)$.

בסה"כ עבור כל $s \in S$ מתקיים $0 \le h_{cur,B}\left(s\right) \le h^*\left(s\right)$ כנדרש.

.13

מימוש $\_heuristic\_minimal\_time$ בקובץ $algorithms.py$.

.14

מימוש uc_time ו־a_star_time ב־$main2b.py$.

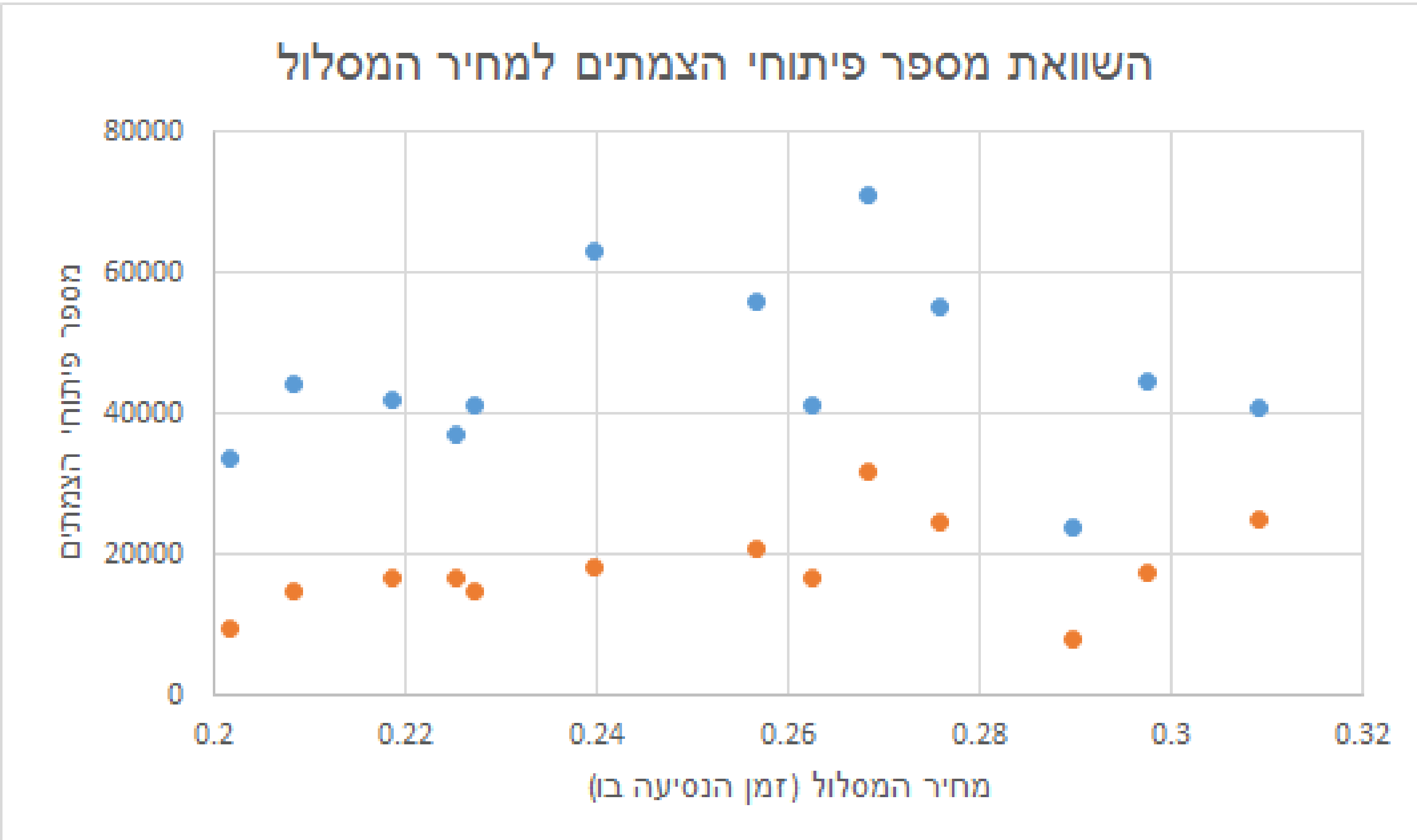
חלק ו' - השוואת האלגוריתמים

.15

ביצוע הניסוי. נמצא ב־$db/experiment3.csv$ וניתוח הניסוי נמצא ב־$experiment3\_analysis.xlsx$.

.16

א.



ב.

מחיר מסלול	מספר פיתוחי צמתים (UC3)	מספר פיתוחי צמתים (A*)3	מחיר מסלול	מספר פיתוחי צמתים (UC3)	מספר פיתוחי צמתים (A*)3
0.225349269	37130	16587	0.23971972	63020	18265
0.142921893	17780	5026	0.275737734	55205	24469
0.262388291	41282	16585	0.170345638	33960	7675
0.218645153	41968	16820	0.178278797	16410	4097
0.227155542	41040	14843	0.171715171	34339	8455
0.153116956	23825	8206	0.289566508	23680	8155
0.256527208	56056	20775	0.297394001	44490	17284
0.268314864	71052	31928	0.309035877	40939	24901
0.20169752	33620	9429	0.208194196	44090	14703
0.174292723	26936	7690	0.179416245	31879	12933

כמובן שב־$3(A^*)$ ו־$UC3$ הכוונה היא ל־A^*_{time} ו־UC_{time} בהתאמה, אך הייתה טעות בדף התרגיל, אז זרמתי אתכם D :.

ג.

כצפוי, עלות הפתרונות המתקבלת זהה בין שני האלגוריתמים (UCS_{time} תקף, וכן גם A^*_{time} מכיוון שמשתמש ביוריסטיקה תקפה), אך A^*_{time} מנצח את UC_{time} במבחן מספר פיתוחי הצמתים (~זמן הריצה).