

בינה מלאכותית – תרגיל 1

(1)

כל רשומה היא צומת שמכילה –

מזהה הצומת, קו רוחב, קו אורך ורשימה של מחלפים (כביש יציאה, מרחק, וסוג הכביש)

(8)

```
37 def g(node):  
38     return node.path_cost
```

```
25 def step_cost(self, s, a):  
26     return (a.distance / 1000) / SPEED_RANGES[a.highway_type][1]
```

פונקציית המחיר –

קלט: קודקוד

פלט: המחיר עד אליו

פונקציית המחיר מוגדרת להיות זמן הנסיעה בין הצמתים, חישוב של המרחק עם המהירות המקסימלית של הכביש, תחזיר לנו את זמן הנסיעה.

```
30 def heuristic(node, problem):  
31     junction_start = problem.G[node.state]  
32     junction_end = problem.G[problem.goal]  
33     return compute_distance(junction_start.lat, junction_start.lon, junction_end.lat, junction_end.lon) / 110
```

פונקציית יוריסטיקה –

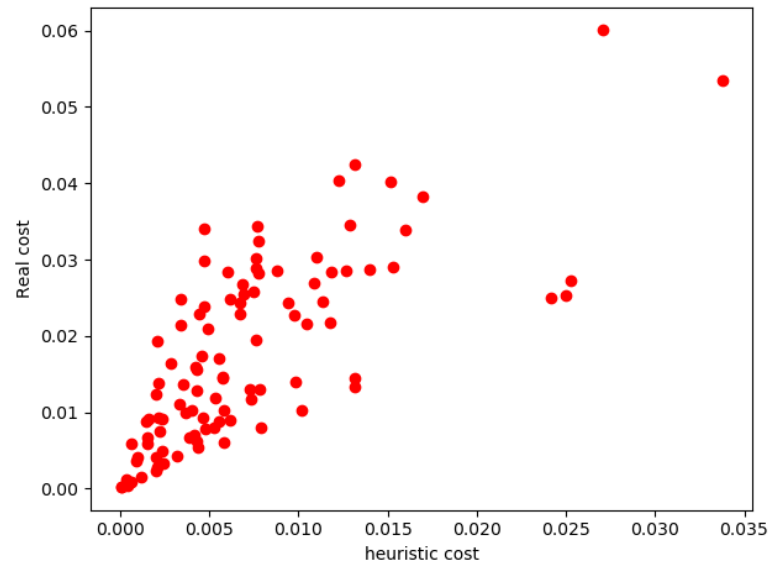
קלט: קודקוד והבעיה (שמכילה את נקודת הסיום)

פלט: חישוב מרחק אווירי עד לנקודת הסיום חלקי 110 שזה המהירות המקסימלית בכבישים

הפונקציה קבילה מכיוון שהיא עומדת ב3 התנאים:

- היא לא שלילית – אי אפשר לקבל מרחק שלילי והמהירות היא חיובית ולכן בהכרח התוצאה תהיה שלילית
- $H(goal) = 0$ המרחק בין נקודת הסיום לעצמה היא 0 ולכן $0 = 0/110$
- היא בהכרח נותנת שיערוך קטן או שווה לעלות האמיתית – כי היא מחשבת מרחק אווירי ולפי מהירות הכביש הכי מהיר (ולא ייתכן דרך יותר מהירה מזאת).

(9)



ניתן ללמוד מהגרף (בעיקר מקנה המידה של הצירים) כי הפונקציה היוריסטית היא די "תמימה" ומחזירה זמן נסיעה קצר יותר מזמן הנסיעה האמיתי מכיוון שהיא לוקחת מרחק אווירי ומהירות גבוהה, דבר שמציאות לא קורה מכיוון שאין כביש ישיר ליעד ובחלק מהכבישים יש הגבלת מהירות.

(10)

לא בהכרח, מכיוון שהאלגוריתם שלנו מתייחס למרחק עם מהירות נסיעה אופטימלית (הכי מהר שאפשר באותו הכביש) ולכן המסלול עם המרחק הקצר בשיקלול עם מהירות הנסיעה בו יבחר אך אם יש עומס על הכביש אזי יכול להיווצר מקרה שהכביש שאומנם הכי קצר ומהיר ליעד **עמוס** ולכן מהירות הנסיעה בו **תפחת**, וכביש עוקף (שאומנם יותר ארוך) יהיה יותר מהיר מכיוון שמהירות הנסיעה בו ללא פקק תהיה מהירה מהנסיעה בכביש הפקוק.

(11)

התשובות 8,9,10 זהות בדיוק אותו דבר מכיוון ששני האלגוריתמים הם שלמים ולכן שניהם מוצאים את אותן הפתרונות (עד לדיוק של מסלול שמחירו שווה). לשניהם אותה פונקציה יוריסטית ולכן סעיף 9 זהה.

(12)

אלגוריתם	UCS	*A	IDA*(5 דוגמאות בלבד)
זמן ריצה ממוצע	0.012332	0.003764	1.38515
סטיית תקן	0.02478	0.005699	2.481738

על פי הטבלה ניתן לראות כי A* הינו האלגוריתם שרץ בזמן הקצר ביותר, וזאת מכיוון שהסתכלות על כמות הפיתוחים של הקודקודים הוא האלגוריתם שפיתח הכי פחות. מכיוון שהוא מפתח כל פעם רק צומת אחת (בדומה לUCS) אך בנוסף הוא חכם יותר מUCS מכיוון שהוא לא מתחשב רק במסלול עד הצומת אלא גם משתמש ביוריסטיקה על מנת לחזות את המרחק ליעד ולוקח זאת בחשבון.

ניתן לשים לב באלגוריתם IDA* שסטיית התקן גדולה – זה נובע גם מכך שהמדגם קטן (רק 5 דוגמאות) אך גם מכיוון שכאשר אנו בודקים מסלול שמצריך מספר רק של פיתוחים הרקורסיה נהיית יקרה מאוד ולכן דוגמאות "קשות" ייקחו הרבה מאוד זמן בהשוואה לבעיות "קלות". בנוסף לוקח הוא הכי איטי מכיוון שהו מפתח צמתים מספר רב של פעמים על מנת לחסוך בפיתוח מיותר של צמתים חדשים.

(14)

בתור האלגוריתם הייתי מאפשר לי להוריד "כבישים" לא רצויים ובכך לחפש מסלול בלי הכבישים הללו. יכול להתבצע פשוט ע"י מחיקת הכבישים מ"המפה" שאני טוען.

בתור חיפוש נאיבי עבור מסלולים אלטרנטיביים יכולתי פשוט למחוק כל פעם כביש אחר שהשתמשתי בו בדרך ובכך הייתי מוצא מסלולים שונים טובים גם הם, לאחר מכן הייתי משווה ומוצא את הטובים מבין המסלולים הללו שמצאתי.

על מנת לשדרג את האלגוריתם ובדומה לWaze הייתי מחזיק מידע על כבישים "מועדים" זאת אומרת כבישים שאני יודע שהם פקוקים בימים ככה וככה בשבוע ובשעות מסוימות, וכאשר הייתי מחפש מסלולים הייתי מריץ חיפוש ללא הכבישים הללו ובכך מוצא דרכים שאין בהם פקקים. (אופציה זאת היא עבור עומסי תנועה כבדים)

אם למוצר שלי יש הרבה מאוד מידע הייתי יכול לשקלל את זמן הפקק בזמן הנסיעה הרגיל בו ובכך ליצור פונקציית משקל חכמה (על פי תזמוני נסיעות לפי ימים, שעות וכו') ובכך לא בהכרח הייתי נמנע מכבישים עם פקקים אם האופציה האחרת שלי היא כביש שהוא ממש ארוך (דבר שכן יכל לקרות ברעיון הקודם שהצעתי).

בנוסף כמו בכל אפליקציית ניוטים הייתי מאפשר למשתמש לבחור בין מסלול קצר, לבין מסלול מהיר שקלולים אלו יכולים לבוא לידי ביטוי בהוצאות הדלק והעדפות הנהג לגבי אופי הנסיעה. לפי כל בחירה כזאת הייתי משנה את פונקציית המחר לפי היחס (אם פונקציית המשקל בנויה ממרחק הנסיעה + זמן הנסיעה המשוער על פי הנתונים הקיימים, על פי בחירת המשתמש הייתי כופל כל אחד מהנתונים במספר המתאים).