

מבוא לבינה מלאכותית (236501)

חורף 2015-16

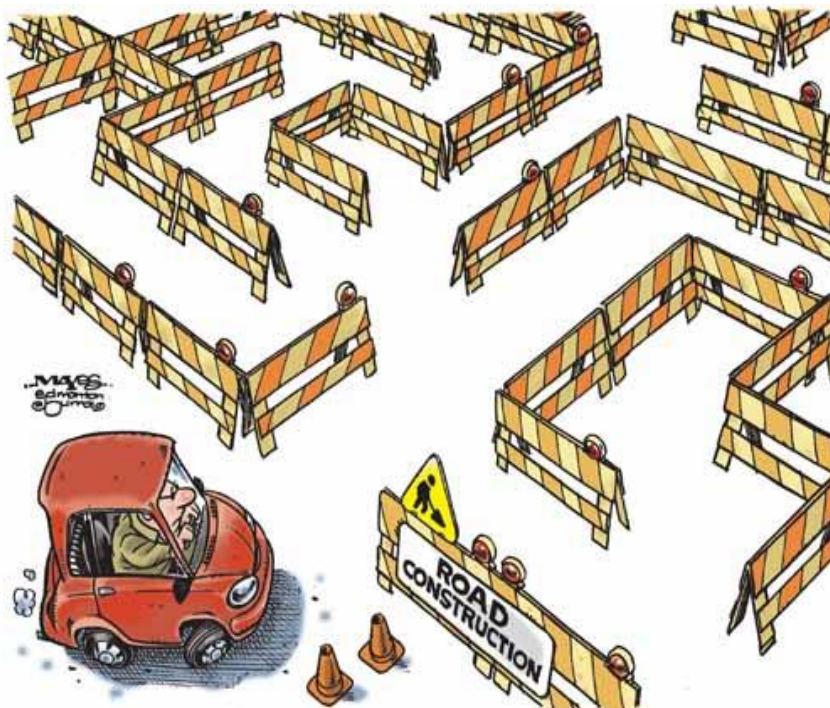
תרגיל בית 1: ניווט כבישים

מטרות המשימה

- התנסות בייצוג בעיות אקטואליות ומציאותיות כמרחבי מצבים.
- התנסות באלגוריתמי חיפוש יוריסטיים.
- תכנון ווריאציה אלגוריתמית בהתאם לבעיה.

הערות:

- תאריך הגשה: 4/12/2015
- המטלה להגשה **בזוגות בלבד!**
- שאלות לגבי התרגיל, נא לשלוח לאמיר בלבד: amirc@cs.technion.ac.il
- בקשות מוצדקות לדחייה בתרגיל יש לשלוח למיטל: maytal@tx.technion.ac.il



הקדמה

בכיתה למדנו לחשב מסלול אופטימלי בגרף ע"י שימוש בפונקציית מחיר המקבלת כקלט שני צמתים עוקבים s_1, s_2 ומחזירה את העלות של מעבר מצומת s_1 ל s_2 . בתרגיל זה נרצה לאפשר פונקציית מחיר דינמית, שיכולה להחזיר ערכים שונים כתלות בפרמטר נוסף t המסמל את השעה שבה מתבצע החיפוש. בצורה זו אנו מאפשרים לאלגוריתם להבדיל בין זמני נסיעה שונים במקטעים בשעות שונות. למשל – נסיעה מצומת A ל B יכולה לקחת 10 דקות בשעה 13:00, ו-15 דקות בשעה 9:00.

תיאור הבעיה

חיסרון אחד של תוכנות הניווט הפופולאריות היום (כדוגמת waze או google maps) הוא חוסר היכולת להתמודד עם עומס משתנה בכביש. כלומר, ברגע חישוב המסלול, האפליקציה ממשקלת את זמן הנסיעה בכל קטע דרך לפי נתונים שהיא מקבלת בזמן אמת. תכונה זו עלולה לגרום לאי דיוקים, מכיוון שעד שנגיע לקטע דרך מסוים המהירות הממוצעת בו עלולה להשתנות. לדוגמא, רכב שיצא לדרך בדיוק לפני שעת העומס, יראה זמן נסיעה קצר יותר מהזמן שייקח לו לעבור את הדרך בפועל, שכן עומס הולך להיבנות בכבישים ולהאט את המהירות הממוצעת בהם. השיפור שנעבוד עליו בתרגיל זה ייתן מענה חלקי לבעיה זו ע"י קירוב זמן הנסיעה בכל קטע לזמנים שנמדדו באותו זמן בימים קודמים.

תוכנת Waze מוצאת את המסלול בעל זמן הנסיעה הצפוי הקצר ביותר, כאשר זמן הנסיעה בכל קטע משוערך על ידי המהירות הממוצעת העכשווית באותו קטע.

תוכנת BetterWaze אותה תכתבו תנסה לבצע שיערוך מדויק יותר של המהירות הממוצעת בכל קטע על ידי שילוב של המהירות הממוצעת הנוכחית והמהירות הממוצעת ההיסטורית השמורה מימים קודמים.

התוכנה תעבוד עם קובץ נתונים המייצג את רשת הכבישים של ישראל. אנו ביצענו הורדה של מפת ישראל מאתר www.openstreetmap.org והמרה לפורמט שיהיה נוח לעבודה עם python ושיכלול רק את המידע הרלוונטי מתוך שלל הנתונים המקוריים, זהו הקובץ `israel.csv`.

חלק 1 – מבוא והקדמה

1. פיתחו את הקבצים `db/israel.csv` ותארו את המבנה שלהם במדויק: מה מייצגת כל שורה ומה פרמטרים בה. לצורך כך, עיינו בקוד של השגרה `load_map_from_csv` בקובץ הקוד המסופק `ways/graph.py`.
2. מלאו את תוכן שגרת העזר `map_statistics` המחשבת פרמטרים המאפיינים את המפה הנתונה.

יש לאפשר להריץ את הקוד עבור הסעיף דרך שורת הפקודה

```
$ python stats.py
```

הפלט צריך להיות תוצאת הדפסה פשוטה של המילון אותו מחזירה הפונקציה.

חלק 2 - A* בסיסי

בחלק זה נממש את A* כפי שנלמד בכיתה, בתוספת פרמטר המסמל את זמן הרצת החיפוש. לשם הפשטות, לכל אורך התרגיל נניח שזמן נמדד בשלמים מ-1 עד $1440 = 24 \times 60$. לפני מענה על השאלות בחלק הזה מומלץ לעבור על כל הקוד המצורף לתרגיל.

3. כתבו בפייתון פונקציה `run_astar`, המקבלת חמישה פרמטרים - צומת התחלה, צומת סיום, פונקציית מחיר, פונקציה היוריסטית ואת השעה בה מופעל החיפוש (נסמן ב- t_0), ומחזירה את המסלול המהיר ביותר מנקודת המוצא ליעד בעזרת אלגוריתם A*.
4. נרצה להפעיל את מתודת החיפוש הנ"ל עם פונקציית מחיר של זמן נסיעה ויוריסטיקה קבילה מתאימה. על פונקציית המחיר לחשב את זמן הנסיעה בכל קטע דרך בהסתמך על המהירות בקטע בזמן t_0 . כלומר, החתימה של הפונקציה תהיה $cost(s_1, s_2, t_0): s_1, s_2 \in S, t_0 \in [1, 1440]$. עליכם לממש את פונקציית המחיר ויוריסטיקה מתאימה. הסבירו בדו"ח את מבנה הקלט והפלט של שתי הפונקציות הנ"ל בקוד ובנוסף מדוע היוריסטיקה אכן קבילה.

```
הקובץ main.py מרכז את הממשק אל שורת הפקודה; יש לכתוב בו מעט ככל הניתן.  
יש לאפשר בדיקה של הקוד עבור שני הסעיפים הקודמים.  
למשל, אם נקודת המוצא היא 30 ונקודת היעד 55:  
  
$ python main.py simple 30 55 700  
על הפלט להיות רשימה פשוטה של מספרי צמתים, כולל קצוות:  
30 21 44 73 55
```

5. צרו 20 בעיות חיפוש אקראיות והריצו עליהן חיפוש A*. עבור כל אחת מהבעיות, פלטו לקובץ `results/AStarRuns.txt` את הזמן המשוער ע"י היוריסטיקה מהמוצא ליעד ואת זמן הנסיעה בפועל. הציגו בדו"ח גרף ובו נקודה לכל אחת מההרצות הנ"ל, המציג את הקשר שבין המשתנים. מה ניתן ללמוד מהתוצאות על קשר זה? בהסבר יש להתייחס לגרף עצמו.
- הערות חשובות:

- לכל אורך התרגיל נניח כי זמן הרצת החיפוש וזמן היציאה לדרך זהים. כלומר, אנו מניחים שהמשתמש מבקש מסלול נסיעה מתוך כוונה לצאת לדרך מיידית, ואנו מזניחים את זמן חישוב המסלול.
- חלקים מהמפה אינם קשירים, לכן על אחריותכם לסנן בעיות חיפוש שאינן פתירות.

חלק 3 – שיפור ל- A^*

בחלק זה נממש שיפור ל- A^* כמו שהוגדר בהקדמה ובתיאור הבעיה.

6. בהנחה שהרצתם את אלגוריתם A^* בדיוק לפני שהעומס בכביש התחיל, האם בהכרח המסלול המתקבל יהיה אופטימלי? הסבירו את קביעתכם.

נגדיר את הפונקציות הבאות:

a. $T_r(s_1, s_2, t)$ – מקבלת את הזמן כרגע (t) ושני צמתים s_1 ו- s_2 , ומחזירה את הזמן

שייקח לנסוע מצומת s_1 ל- s_2 לפי העומס בכביש המחבר ביניהם בזמן הנוכחי t .

b. $T_h(s_1, s_2, t)$ – מחזירה את הזמן המשוערך לנסוע מצומת s_1 ל- s_2 לפי שקלול

היסטוריית המהירויות בכביש המחבר ביניהם בזמן t בימים האחרונים.

7. כדי להתמודד עם הבעיה מסעיף 6, הוצע לשנות את פונקציית המחיר כך שתכלול גם את הזמן שבו יוצאים מ- s_1 ל- s_2 . כלומר, החתימה החדשה של הפונקציה תהיה

$cost(s_1, s_2, t_0, current_time)$: $s_1, s_2 \in S, t_0, current_time \in [1, 1440]$ כאשר המשתנה

t_0 מציין את זמן הרצת החיפוש ו- $current_time$ מציין את הזמן בו יוצאים מהצומת s_1 . הוסיפו

מימוש חדש ל- A^* , $run_astar_with_time$ (אותה חתימה כמו מקודם), המכיל תמיכה

בפונקציית המחיר החדשה.

8. הוגדרה פונקציית המחיר הבאה –

$$cost(s_1, s_2, t, t_0) = \frac{1}{|Focus|} \sum_{(s_3, s_4) \in Focus} \frac{T_r(s_3, s_4, t_0)}{T_h(s_3, s_4, t_0)} \cdot T_h(s_1, s_2, t)$$

$$s_1, s_2, s_3, s_4 \in S, t, t_0 \in [1, 1440]$$

הסבר על הפרמטרים:

• s_1 - צומת המוצא.

• s_2 - צומת עוקב של s_1 .

• t - הזמן בו מתעתדים לצאת מצומת s_1 ל- s_2 .

• t_0 - הזמן בו מריצים את האלגוריתם.

כאשר הקבוצה $Focus$ מכילה אוסף קשתות בקרבת צומת המקור (s_1). לחתימת הפונקציה יש

להסתכל בנספח.

a. הסבירו במילים את הרציונל מאחורי פונקציית המחיר.

- b. תנו דוגמא בליווי הסבר מניח את הדעת למקרה שבו פונקציית המחיר החדשה נותנת פתרון אופטימלי ושפונקציית המחיר הישנה (כלומר שאינה תלויה בפרמטר הזמן) נותנת פתרון שאינו אופטימלי. (כלומר שהמסלול שיוחזר בפועל ע"י פונקציית המחיר המקורית לא יהיה אופטימאלי תחת ההנחה שהעומס משתנה כפונקציה של הזמן).
- c. תנו דוגמא בליווי הסבר מניח את הדעת למקרה בו המסלול שיוחזר ע"י פונקציית המחיר החדשה אינו אופטימאלי. כלומר בפועל קיים מסלול מהיר יותר מהמסלול המוחזר ע"י A^* עם פונקציית המחיר החדשה.
- d. ממשו את פונקציית המחיר שהוגדרה בתחילת השאלה.
- e. כתבו היוריסטיקה לא טריוויאלית (כלומר לא היוריסטיקת האפס), כך שאלגוריתם A^* המשתמש בה יהיה קביל תחת פונקציית המחיר שהוגדרה. הוכיחו את בחירתכם.

יש לאפשר להריץ את הקוד עבור שני הסעיפים הקודמים דרך שורת הפקודה (הפרמטרים לפי הסדר הם – צומת התחלה, צומת יעד, וזמן יציאה לדרך)

```
$ python main.py improved 30 55 700
```

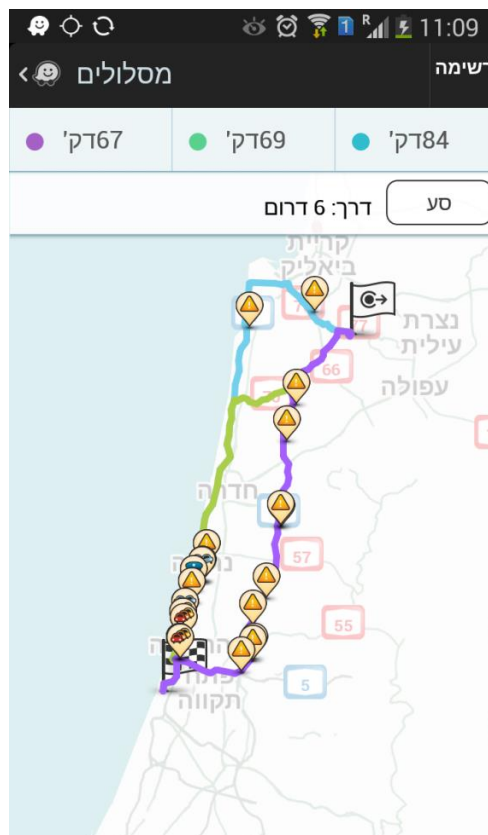
על הפלט להיות רשימה פשוטה של מספרי צמתים, כולל קצוות:

```
30 21 44 73 55
```

9. חזרו על סעיף 5 עם פונקציות המחיר והיוריסטיקה החדשות.

10. **שאלת רשות:** בתכנת Waze, בהינתן נקודת מוצא ונקודת יעד יש אפשרות לקבל הצעות לכמה מסלולים אלטרנטיביים.

- a. חשבו על מצבים שבהם המשתמש ירצה לקבל הצעות למספר מסלולים שונים. ציינו מהם השיקולים בבחירת המסלולים ולמה הם עשויים להיות חשובים למשתמש.
- b. אתם מתבקשים להציע רעיון אלגוריתמי לסיפוק אפשרות שכזו. תארו את השיטה מילולית או באמצעות פסאודוקוד ברמת אבסטרקציה גבוהה והסבירו מדוע השיטה שבחרתם תספק מענה טוב לשיקולים שציננתם לגבי המסלולים. דוגמה לפלט סביר היא למשל שלושת המסלולים הבאים מרמת ישי לתל אביב. **בנוסף יינתן לרעיונות יצירתיים במיוחד.**



הנחיות הגשה

עליכם להגיש דרך אתר הקורס תיקייה מכונצת לארכיון *zip/rar*. על התיקייה להכיל את הקבצים הבאים:

חלק יבש : דו"ח מחקר

- זהו החלק אשר יקבע את ציונכם, הקדישו לו את הזמן הראוי לו.
- אין צורך להדפיס את הדו"ח. הגישו אלקטרונית בלבד כקובץ *PDF* הממוקם בתיקיית העל של ארכיון *RAR/ZIP* שאתם מגישים.
- הקפידו לכתוב את האימיילים שלכם גם בראש הדו"ח לצורך קבלת המשוב.
- נמקו היטב את כל תשובותיכם, הסבירו את המוטיבציה ליורסטיקות, הסבירו תוצאות מעניינות שרואים בגרפים או בכל מידע אחר שאספתם ונראה לכם רלוונטי.

חלק רטוב : הקוד

- עליכם להגיש כל קוד שנכתב לצורך ביצוע המטלה.
- הנחיות לגבי קונבנציות בפיתון מופיעות כאן: www.python.org/dev/peps/pep-0008/
- מומלץ לעבוד לפיהן אך אי עמידה בהן לא תפגע בציון.
- תיעוד למבנה התיקיות מופיע בתיקייה *docs*.
- הקוד שסופק לכם יכול להיות מוכל בפרויקט המוגש, מלבד קובץ המפה בשל גודלו.
- במידה ואתם חושבים שיש צורך **חיוני** בשינוי כלשהו בקבצי הקוד המסופקים - אתם מוזמנים לפנות לאמיר במייל. שימו לב שבפיתון ניתן לשנות אובייקטים רבים באופן דינמי.

טרם הגשה:

- עדכנו את הקובץ *docs/submissions.txt*. עם הפרטים שלכם.
- שימו לב שכל הפנייה למיקום קובץ/תיקייה כלשהי בקוד תהיה רלטיבית ולא אבסולוטית (*relative path*), כך שהקוד יעבוד כפי שהוא על כל מחשב בכל מיקום שנבחר לתיקיית הפרויקט. הקפידו לבדוק זאת לפני ההגשה!
- אם אתם משתמשים בחבילות חיצוניות מלבד *matplotlib*, *numpy* ו *scipy*, שלא ניתן להתקין באמצעות הפקודה *pip install*, יש לשים את החבילה בתיקיית הפרויקט, על מנת שיהיה אפשר להריץ זאת על כל מחשב. אם החבילה תופסת המון מקום, יש להתייעץ עם הסגל לפני כן.
- שימוש בכל קוד חיצוני מצריך הצהרה מפורשת על המקור שלו. במקרה של שימוש בספריה מסודרת, יש להוסיף את שמה לקובץ *docs/dependencies.txt*.
- הסירו את קובץ המפה מהתיקייה db/israel.csv.

בהצלחה!

Appendix – functions documentation

All of the following functions are defined under road class.

Roads.link_speed_history(link,time)

link – a link in the graph.

time – the time that we want to sample

Return - driving speed on this link at time “time”, according to the “history” we collected

Roads.link_speed_realtime(link,time)

link – a link in the graph.

time – the time that we want to sample

Return - the speed of the link in “real time”

Roads.return_focus(node)

node – an index of the node we want to generate focus from

return - some small number of links around the nod.

Note: the input is an index, not a junction. The number of links returned may vary

Note: Time values should be saved as float, as doing int(time) might result in time staying 0 all the time (since the time to traverse a link is usually less than a minute. don't worry though, there are A LOT of links)