

מבוא לבינה מלאכותית (236501)

חורף 2014-15

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: 4.12.14
- המטלה להגשה בזוגות בלבד.
- שאלות לגבי התרגיל, נא לשלוח לעומר omergcs@cs.technion.ac.il



תיאור הבעיה

מזל טוב! התקבלתם לעבודה בחברת BetterWayz שמפתחת תוכנה לניווט כבישים. התכנה מוסיפה אפשרות חדשה של "ניווט מובטח": המשתמש מספק יעד וזמן אחרון אשר עד אליו הוא מעוניין להגיע ליעד. התכנה מחזירה את הדרך המהירה ביותר המבטיחה הגעה ליעד לפני הזמן המבוקש ברמת בטחון קבועה (פרמטר של המשתמש).

התוכנה תעבוד עם קובץ נתונים המייצג את רשת הכבישים ישראל. אנו ביצענו הורדה של מפת ישראל מאתר www.openstreetmap.org והמרה לפורמט שיהיה נוח לעבודה עם python אשר יכלול רק את המידע הרלוונטי מתוך שלל הנתונים המקוריים, זהו הקובץ db/israel.csv. בנוסף למפה מסופקים כמה מודולים בפיתוח, נא לקרוא ולהבין לעומק את הקוד המסופק.

חלק תאורטי (יבש) [35 נק']

1. השאלה הבאה מתייחסת לאלגוריתם A^* כפי שהוא מוצג בפסאודוקוד מהשיעורים. מעוניינים לשנות את אלגוריתם A^* על מנת שיימצא מסלול פתרון זול ביותר מבין אלו שמקיימים תנאי לוגי מסוים שנקרא לו *FeasibilityCodition*. לדוגמה, יתכן שנדרוש נסיעה בכבישים סלולים היטב לפחות פרופורציה מסוימת מהדרך, או שנדרוש לעבור דרך מקטע מסוים בשל הנוף המרהיב בו. סטודנט הציע להוסיף לאלגוריתם בדיקה של התנאי על המסלול לפני החזרת פתרון (החלק המודגש חדש):

```
if problem[goalp](next) and FeasibilityCodition(path(next))  
  
    then return(path(next))
```

חברו טען אח"כ שהאלגוריתם החדש כבר אינו קביל ואינו שלם תחת התנאים הרגילים לקבילות A^* (יוריסטיקה קבילה ופונקציית מחיר חיובית וחסומה מלמטה).

הצדיקו קביעה זו באמצעות דוגמה לפגיעה בשלמות.

בנוסף [10 נק']: הסבירו מדוע לא ניתן להדגים פגיעה בקבילות אך לא בשלמות.

2. הציעו שינוי אלגוריתמי כך שהשלמות והקבילות לא ייפגעו.

a. תארו מילולית את השינוי.

b. כתבו פסאודוקוד מפורט דוגמת זה מהשיעור.

c. הוכיחו את שלמות וקבילות האלגוריתם.

3. בתכנת Waze, בהינתן נקודת מוצא ונקודת יעד יש אפשרות לקבל הצעות לכמה מסלולים אלטרנטיביים. אתם מתבקשים להציע רעיון אלגוריתמי לסיפוק אפשרות שכזו. ציינו אילו שיקולים מעניינים אתכם בבחירת המסלולים ולמה הם עשויים להיות חשובים למשתמש. תארו את

השיטה מילולית או באמצעות פסאודוקוד ברמת אבסטרקציה גבוהה והסבירו מדוע השיטה שבחרתם תספק מענה טוב לשיקולים שצינתם לגבי המסלולים. דוגמה לפלט סביר היא למשל שלושת המסלולים הבאים מרמת ישי לתל אביב.



חלק רטוב א' [35 נק']

4. פיתחו את הקבצים `db/israel.csv`, `db/lights.csv` ותארו את המבנה שלהם במדויק: מה מייצגת כל שורה ומה הפרמטרים בה. לצורך כך, עיינו בקוד של השגרה `load_map_from_csv` בקובץ הקוד המסופק `ways/graph.py`.
5. 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.

בשלב זה חשוב לקרוא לעומק את כל הקוד המסופק עם התרגיל

6. כתבו בפייתון פונקציה `run_astar` המחזירה את המסלול מנקודת המוצא לייעד שיוצר האלגוריתם A^* תחת שימוש בפונקציית מחיר כלשהי ויוריסטיקה תואמת.

7. נרצה להפעיל את מתודת החיפוש הנ"ל עם פונקציית מחיר של זמן נסיעה ויוריסטיקה קבילה 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  
על הפלט להיות רשימה פשוטה של מספרי צמתים, כולל קצוות:  
30 21 44 73 55
```

8. צרו 20 בעיות חיפוש אקראיות והריצו עליהן חיפוש A^* . פלטו לקובץ `results/AStarRuns.txt`. עבור כל אחת מהבעיות את הזמן המשוערך ע"י היוריסטיקה מהמוצא לייעד וזמן הנסיעה בפועל. הציגו בדו"ח גרף ובו נקודה לכל אחת מההרצות הנ"ל. מה ניתן ללמוד מהגרף על הקשר בין המשתנים?

חשוב להיות מודעים לכך שחלקים מהמפה אינם קשירים (מסיבות לא ברורות). לכן, על אחריותכם לסנן בעיות חיפוש שאינן פתירות.

9. ישנם משתמשים המבקשים להגיע לייעדם תוך מעבר במספר קטן של צמתים מרומזרים. כתבו קוד התומך בחיפוש עם פונקציית מחיר המשקללת את השיקול החדש של הימצאות רמזור עם השיקול המקורי של זמן נסיעה. פונקציית המחיר הנ"ל צריכה להיות גנרית במובן זה שהאיזון בין השיקולים מושג באמצעות פרמטריזציה כרצונכם אשר אתם מתבקשים להסביר בדוח. כתבו גם יוריסטיקה קבילה תואמת לפונקציית המחיר החדשה והסבירו בדו"ח מדוע היא קבילה.

```
יש לאפשר בדיקה של הקוד דרך שורת הפקודה:  
$ python main.py lights 30 55
```

10. הציגו בעיה לדוגמה בה המסלולים המוחזרים עם ובלי שיקול הרמזורים נבדלים האחד מהשני. הראו את המסלולים גרפית (לשם כך ניתן להשתמש בקוד ב- `ways/draw.py`). הסבירו מדוע כל חיפוש הניב את מסלולו תוך התייחסות לערכי הפרמטרים בהם השתמשתם להגדרת החיפוש מונחה הרמזורים.

חלק רטוב ב' - ניווט מובטח [30 נק']

בחלק זה תפתרו את בעיית הניווט המובטח תוך שימוש בפונקציית מחיר "זמן נסיעה" מהחלק הקודם והיוריסטיקה המתאימה.

קלט הבעיה: נקודת מוצא, נקודת יעד, זמן נסיעה מקסימלי, רמת בטחון.

פלט הבעיה: מסלול מהיר מהמוצא ליעד אשר זמן הנסיעה בו קצר מהמקסימלי וזאת ברמת בטחון הרצויה.

אין ברשותכם מידע מדויק לגבי מהירויות המקטעים בכבישים. במקום זאת, יש ברשותכם מידע היסטורי לגבי המהירויות ב N (100) דגימות זמן (נניח שכולן מאותה השעה ואותו היום בשבוע).

אלגוריתם למציאת ניווט מובטח

1. בחרו אקראית יוניפורמית k מתוך N דגימות הזמן הנתונות.
2. עבור כל דגימת זמן מתוך k שהוגרלו, הריצו את A^* מנקודת המוצא לנקודת היעד. כך מקבלים k מסלולים פוטנציאליים לנסיעה.
3. עבור כל אחד מ k המסלולים שהתקבלו, חשב את זמן הנסיעה במסלול בכל אחת מ N דגימות הזמן. כך מקבלים לכל אחד מ k המסלולים מדגם בגודל N של זמני נסיעה במסלול.
4. השאירו בקבוצת המסלולים רק את אלו שזמן נסיעתם יהיה קצר מהזמן המקסימלי, ברמת בטחון הנדרשת עפ"י המדגם שנאסף (יוסבר בהמשך).
5. מתוך המסלולים הפוטנציאליים שנותרו החזירו את המסלול בעל זמן הנסיעה הממוצע הקצר ביותר.

בדיקת עמידה בזמן נסיעה ברמת בטחון

קלט: זמן מקסימלי (T) , ברמת בטחון (p) , מדגם זמני נסיעה (X_1, X_2, \dots, X_N) .

1. ספור כמה מהזמנים במדגם X_i קטנים מהזמן המקסימלי T :

$$\sum_{i=1}^N I(X_i < T)$$

2. חשב את פרופורציית הדגימות העומדות בהגבלת הזמן:

$$\frac{\sum_{i=1}^N I(X_i < T)}{N}$$

3. אם פרופורציה זו גדולה/שווה p – החזר $True$, אחרת $False$.

13. האם האלגוריתם למציאת ניווט מובטח שלם והאם קביל? הסבירו.

14. ממשו את האלגוריתם המתואר מעלה עבור ניווט מובטח.

על מנת לקבל דגימות מהירויות שונות של המקטעים יש לשנות את הgeneration של המפה
כמוסבר בקובץ [ways/graph.py](#).

יש לאפשר בדיקה של הקוד דרך שורת הפקודה.
עבור זמן נסיעה מקסימלי 237 דקות, וביטחון של 99.5%:
`$ python main.py assured 30 55 237 99.5`

15. הריצו את האלגוריתם לפתרון אוסף של 20 בעיות אקראיות עם ערכי פרמטרים N, k כרצונכם
ועם לפחות 2 ערכים שונים לפרמטר q . הציגו בדו"ח סיכום של התוצאות.

16. בחרו בעיה מסוימת והציגו עבודה באופן גרפי לפחות 3 מסלולים אלטרנטיביים שהיו מועמדים
לפתרון לפי אחת מדגימות הזמן.

חלק יבש : דו"ח מחקר

- זהו חלק חשוב מאוד בעבודה ויהווה גורם מרכזי בקביעת ציונכם.
- 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.

חלק רטוב : הקוד

- עליכם להגיש כל קוד שנכתב לצורך ביצוע המטלה.
- הנחיות לגבי קונבנציות בפייתון מופיעות כאן: www.python.org/dev/peps/pep-0008/ מומלץ לעבוד לפיהן אך עמידה בהן לא תפגע בציון.
- תיעוד למבנה התיקיות מופיע בתיקיה *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 לעומר במייל. שימו לב שבפייתון 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](http://db.israel.csv)
 - כווצו את התיקייה לארכיון *ZIP\RAR*.
- את הארכיון שיצרתם יש להגיש אלקטרונית באתר הקורס.

בהצלחה ... ותיהנו!