

מעבדה ברשתות

רקע:

סרטון ובו רקע על תוכנת נט-לוגו [התקנת נט-לוגו על המחשב](#)
J14 הינו ארגון של מתנדבים שהוקם ב-2011 בימי המחאה החברתית, מטרת הארגון הייתה להעביר את מסרי המחאה לציבור, ולא להסתמך על העיתונות שברובה נשלטה ע"י טייקונים שהתנגדו למחאה.
במעבדה זאת תשתמשו ברשת J14 שהורדה מה-Facebook. באמצעות רשת זאת עליכם לארגן הפגנה בהשתתפות כל ארגוני המחאה הישראלים. ההפגנה תעשה באמצעות שליחת אימייל שיופץ בין כל הארגונים החברים ברשת.
הרשת נשמרה בקובץ מסוג .gdf.
מבנה קובץ .gdf:
הקובץ מתחלק לשני חלקים:

1. תאור הצמתים שמכיל כמינימום את זיהוי הצומת, והתוית.
2. חיבורי הצמתים שמכיל את כל זוגות הצמתים שמחוברים ביניהם.

בין שני חלקי הקובץ מופיע השורה:

```
edgedef>node1 VARCHAR,node2 VARCHAR
```

מטרות המעבדה:

- קורס מזורז (crash course) של Pandas.
- קורס מזורז (crash course) של Netlogo.

סביבת עבודה:

- Jupiter, (Ipython notebook)
- לינוקס
- Netlogo

קריאה מומלצת:

- tutorial של נטלוגו בכתובת: <https://ccl.northwestern.edu/netlogo/docs/tutorial1.html>
- קריאת נספח על [ipython notebook \(Jupyter\)](#).
- מצ"ב לינק להסבר על [extension](#) של nw, שבעזרתו ניתן לנתח רשתות. ([NetLogo 6.0 User Manual](#))
([Nw Extension](#))

שלב א' :

מכיוון שההפגנה נערכת בישראל, יש ליצור רשת חדשה שהיא תת-רשת של J14 ובא מופיעים רק הארגונים שפועלים בארץ.

- צפו בשני הסרטונים המצורפים hadgama1.mp4, hadgama2.mp4. צרו תקייה בשם netlogowork, הורידו אליה את הקובץ IPYTHON פתרון טיסות" (hadgama.ipynb) ופתחו אותו ב-jupyter. (ראו בנספחים).
- פתחו את הקובץ "קובץ הדגמה של טיסות" (tisot.gdf) ושימו לב למבנה הקובץ.
- קראו ובצעו את ההדגמה.
- ע"ס הדגמה זאת הכינו את הקובץ j14.gdf, לטעינה ל-netlogo, כך שיופיעו בו רק הארגונים החברתיים הפועלים בארץ והקשרים ביניהם.
- יש לחבר בקוד (python) את שני הקבצים שנוצרו בסעיף הקודם כך שיופיעו בקובץ הסופי גם הארגונים וגם הקשרים ביניהם. יש להשאיר את הכותרות של קובץ הגdf המקורי.

שלב ב':

כעת נעבוד על הקובץ שיצרנו בשלב הקודם ונפתור בעזרתו את המשימה (ארגון ההפגנה) על ידי שימוש בתוכנת NetLogo.

סרטון עזר נוסף לשרותכם [טעינת הקובץ לתוכנת נט-לוגו](#).

מטלה א' - השלמת הקוד למדד centrality יחיד:

הורד ממודל את הקובץ gephi.nlogo ופתח אותו בעזרת התוכנה (פתיחת NETLOGO ע"י כניסה ל HOME → NET (LOGO06 → enter

הכנס ללשונית code והשלם את הקוד לפי ההוראות עד לסיום הפרוצדורה go.

מומלץ לקרוא את הנספח צורת העבודה שבסיום תדריך זה.

הסבר על הsetup:

את קובץ gdfn או פותחים באמצעות פקודה של ההרחבה nw. יש להקפיד שהagentset והlinks יוגדרו בהתאם לשמות שנתנו להם בתחילת הקובץ (orgs וlikes).

אנו מגדירים את המשתנים email? (האם הארגון קבל אימייל) ו winner? (האם הארגון הינו הארגון שאליו נפיץ את האימייל הראשון) עם סימן שאלה בסופם מכיוון שהם משתנים בוליאנים וככה נהוג להגדיר משתנים בוליאנים בשפה הזאת.

בתוך הsetup אנו מחפשים את הorg עם מדד הbetweenness centrality הגבוה ביותר אשר ממנו נתחיל להפיץ את האימייל. כיוון שהוא המקור, תכונת ה- winner? שלו תהיה true. Max-n-of לא מדבר בהכרח על יחיד מקסימלי אלא יכול להתייחס לכמה ארגונים, בהתאם לבחירתנו, עם הערכים הגבוהים ביותר. למשל, אם במקום הערך 1 שרשמנו נרשום 2 אז יוצגו שני ארגונים עם מדד betweenness centrality הגבוה ביותר. הערך שנרשום זהו מספר הארגונים שיקבלו את האימייל הראשוני מאיתנו והם יפיצו אותו הלאה.

הסבר על הlayout:

בגלל שיש לנו המון ארגונים אנו חייבים להרחיב את שטח התצוגה למקסימום (יעשה בהמשך) ולעשות כך שכל ארגון יהיה במרחק המקסימלי האפשרי מארגון סמוך כך שכל הארגונים יוצגו פחות או יותר בפיזור שווה ע"פ שטח מקסימלי על המסך.

הסבר על הgo:

הפרוצדורה תרוץ בלולאה אין סופית לכן אנו חייבים תנאי עצירה. תנאי העצירה שלנו הינו 10 יחידות זמן (ticks). היינו מעדיפים לקבוע תנאי עצירה לפיו הפרוצדורה תיעצר רק לאחר שכל הארגונים יקבלו את האימייל, אבל כיוון שתנאי זה לא יתקיים במקרה הנוכחי (לא כל הארגונים יקבלו את האימייל) נשתמש בתנאי עצירה לפי יחידות זמן, שכן זהו תנאי שתמיד יעבוד.

[[email? = true] ask orgs with :

אנו פונים לארגונים שקבלו את האימייל (כלומר שתכונת ה email? שלהם היא true)

[ask link-neighbors :

אנו פונים לארגונים שקשורים אליהם בעזרת הקשרים שהגדרנו בהתחלה

[(0.2 * count (link-neighbors)) >= (count (link-neighbors with [email? = true])) if count :

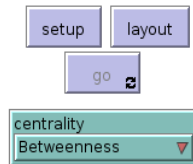
אם 20% מהארגונים שקשורים לארגון מסויים קיבלו את האימייל, הארגון יקבל את האימייל גם ויתחיל להפיצו גם כן.

יש לבדוק את נכונות הקוד אנא העזרו בנספח צורת העבודה בסוף התדריך.

מי שעדיין מתקשה יכול להעזר בנספח פקודות שימושיות שנמצא בסוף התדריך.

מטלה ב' - הגדרת ממשק המשתמש:

נכנס ללשונית Interface וניצור ממשק משתמש שנראה כלהלן:

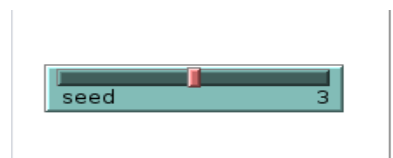


רמז: העזרו ב- [tutorial](#).

כאשר כפתור setup מבצע את פעולת setup, כפתור layout מבצע את פעולת layout וכפתור go מבצע את פעולת go. שימו לב שבכפתור go מסומנת האפשרות של forever ואפשרות disable until ticks start. כך שהמשתמש לא יוכל ללחוץ על go, לפני הלחיצה על setup. על מנת לראות בבירור את כל הארגונים, אנא הגדילו את המסך כך שיהיו 60 patches בכיוון x ו-30 patches בכיוון y. בצעו זאת ע"י לחיצה במקש הימני של העכבר על משטח העבודה ובחירת Edit. centrality מייצג את המדדים המרכזיים של הרשת. כרגע נשתמש רק במדד ה-Betweenness, ובהמשך נשתמש במדדים נוספים. בעזרת מדדים אלה נמצא לאילו ארגונים כדי לנו להפיץ את האימיילים הראשוניים. לרשותכם סרטון עזר [הגדרת ממשק משתמש](#).

משימות להגשה:

1. זהו את הארגון שקיבל את האימייל הראשון.
רמז: בעברית זה נשמע כך: שאל את הארגונים שיש להם winner להראות את Label שלהם.
2. מצאו כמה ארגונים קיבלו את האימייל לאחר עשרה ticks.
כיוון שלא כל הארגונים קיבלו את האימייל, עלינו לשלוח אימייל ראשוני ליותר מארגון אחד. לשם כך נוסיף סליידר seed.
המשמעות של seed הינה מספר הארגונים ההתחלתיים שיקבלו את האימייל הראשוני. נגדיר אותו מ 1 עד 5.



4. שנו את הקוד כך שמספר הארגונים שיקבלו את האימייל ההתחלתי יהיה בהתאם לערך של הסליידר seed.
5. שנו את תנאי העצירה כך שבמקום בעשרה טיקים הישום יעצור לאחר שכל הארגונים קיבלו את האימייל.
רמז: השתמשו ב-reporter, שהוא all?
6. קבעו את seed ל-5 ומצאו תוך כמה טיקים כל הארגונים מקבלים את האימייל.
לרשותכם סרטון עזר [מענה לשאלות](#)

מטלה ג':

במטלה זאת אנו נכין את הקוד כך שנוכל למצוא את הארגונים אליהם כדאי לנו להפיץ את האימייל כך שיופץ לכמה שיותר ארגונים במצבים שונים.

שנו את chooser כך שיכיל את המדדים הבאים: closeness, eigenvector, page-rank, betweenness, ו-1. בעזרת מדדים אלה נמצא לאילו ארגונים כדי לנו להפיץ את האימיילים הראשוניים.

1. שנו את הקוד להלן:

1. סמנו כהערה את שורה מס' 12 "Betweenness" let centrality כך שה- centrality תקבע ע"י ה-chooser.
2. שנו בחזרה את תנאי העצירה כך שהישום יעצר לאחר עשרה טיקים.
3. הרחיבו את ה-setup כך שיכיל את כל מדדי centrality שהוגדרו ב-chooser. רמז: כאשר משתמשים בתוסף של הרחבת nw, לכל המדדים פרט ל- page-rank מופיעה הסיומת - centrality. לדוגמא: nw:eigenvector-centrality.
4. נתון בקוד ה-reporter, בשם zokhe, יש להסביר בשורות הריקות מה כל שורה מבצעת.
5. האם הארגונים הזוכים (ZOKHE) משתנים בהתאם לבחירת מדדי ה- CENTRALITY ?

מטלה ד':

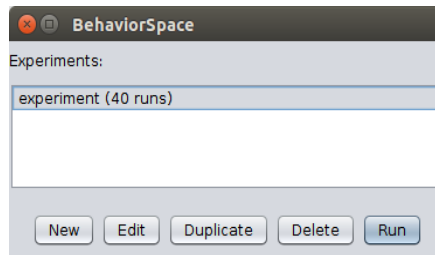
הרצת ניסוי מסודר ב-behavior space למציאת הפרמטרים שיוגדרו בהמשך.

1. ב-netlogo לחץ על tools ובחר את BehaviorSpace. תצרו ניסוי חדש באמצעות להיצה על new ומלאו את הפרמטרים הריקים בחלון שנפתח בהתאם לתמונה המצורפת להלן:

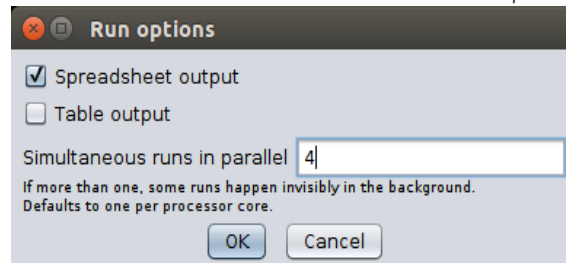
בגיליון התוצאות יש להציג את ה-reporters הבאים:

- א. מספר הטיקים
 - ב. את מספר הארגונים שקבלו את האימייל
 - ג. את הזוכים, הארגונים שקבלו ראשונים את האימייל.
- בחלון Measure runs using these reporters יש למלא את ה-reporters שעונים על הדרישות. לחצו OK.

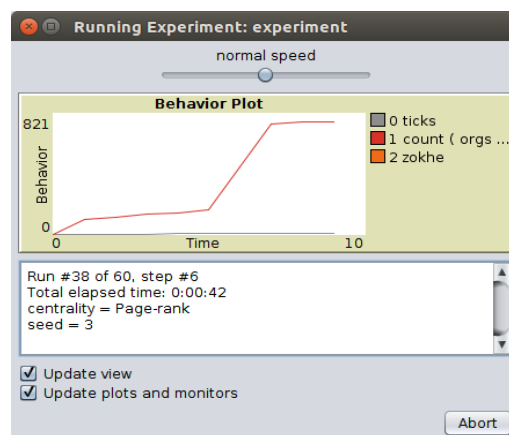
2. בחלון שנפתח סמנו את הניסוי שהגדרתם ולחצו Run.



בחלון החדש שנפתח קבעו 4 חזרות במקביל:



לחצו OK ושמרו את קובץ התוצאות בתקייה של NetLogoWork. הניסוי יתחיל לרוץ כשאתם רואים את החלון להלן:



3. במידה ומותר לשלוח את האמייל הראשוני רק לשלושה ארגונים:

- לפי איזה מדד centrality כדאי לנו לבחור את הארגונים הללו, ע"מ שהאימייל יגיע לכל הארגונים בצורה המהירה ביותר?
 - לאילו ארגונים כדאי לשלוח אותו על מנת שהאימייל יגיע לכל הארגונים?
4. מיהו הארגון שיפיץ את האימייל בעצמו לכמה שיותר ארגונים בכמה שפחות זמן ובאיזה מדד centrality מדובר?
5. בהנחה שאסור לעבור את ה-SEED, (מקס' 5) ועלינו לארגן הפגנה אליה יגיעו כל הארגונים תוך 4 ימים (4 steps).

למי כדאי לנו לשלוח את האימיילים ובאיזה מדד centrality מדובר?
סרט הדרכה נוסף [BehaviorSpace](#)

בדוח המעבדה יש להגיש (בנוסף לסעיפים הרגילים)

1. קובץ JDF מתוקן
2. קוד NELOGO שלם (לאחר מילוי החלקים החסרים)
3. צילום מסך של ממשק הנט-לוגו

4. קובץ CSV - תוצר של מטלה ד'
5. תשובות לשאלות שנשאלו במטלה ב' ובמטלה ד'.

נספחים:

צורת העבודה:

- בתפריט Tools->Preferences סמן את Show line numbers.
- אם ברצונך לבדוק את נכונות הקוד שלך במהלך מטלה א' שבשלב ב':
2. אם ברצונך לקבל הסבר על מילה שמורה ב-netlogo כתוב אותה ולחץ על F1.
 3. מחק את הקו של השורה שברצונך למלא.
 4. מלא את השורה בקוד המבוקש.
 5. לחץ על כפתור ה-Check ובדוק אם טעית.
 6. לאחר מילוי כל הפרוצדורה setup אם ברצונך להריצה גש ל-Interface ובשורת הפקודה כתוב setup.
 7. מלא את כל שאר הפרוצדורות ע"פ ההוראות והרץ אותן באמצעות שורת הפקודה.
 8. אם ברצונך לבדוק את פרוצדורת go, הרץ repeat 10 [go] בשורת הפקודה.

פקודות שימושיות:

פקודה	דוגמה	הסבר
1 Clear-all או ca		מנקה את כל הפעילויות הקודמות.
2 Set	Set color brown	השמת ערך למאפיין. בדוגמה השמת המאפיין צבע לחום.
3 Reset-ticks		מאפס את "מונה הזמן" לאפס, מאפס את הגרפים.
4 Breed-own[bool1? bool2?]	Workers-own[male? kavua?]	הוספת שני מאפיינים בוליאנים לסוכנים, בדוגמה הסוכנים הינם עובדים. העובדים ישמרו כזכרים אם למאפיין male נשים את הערך 1, כנ"ל לקביעות.
5 Nw:load-gdf path breeds links	nw:load-gdf "c:/mylogo/" airports flights	טעינת קובץ GDF בדוגמה הצמתים הינם שדות תעופה והקשרים הינם הטיסות ביניהם.
6 Let	let win []	בדוגמה יצירת משתנה מקומי בשם win והשמת רשימה ריקה לתוך משתנה זה.
7 lput	set win lput label win	בדוגמה השמת הערך של label בתור האיבר האחרון ברשימה win.
8 who		מאפיין של אובייקט ה-turtle שומר את מספר הזיהוי שלו. בד"כ מספר הזיהוי תואם את סדר יצירת האובייקטים.
9 to-report	to-report zokhe[] report win	יצירת reporter זוהי פונקציה שמחזירה ערך. בדוגמה הפונקציה מחזירה רשימה בשם win
10 If _condition_ [stop]	If tickes=10 [stop]	תנאי להפסקת פעולה אחרת

בהצלחה!