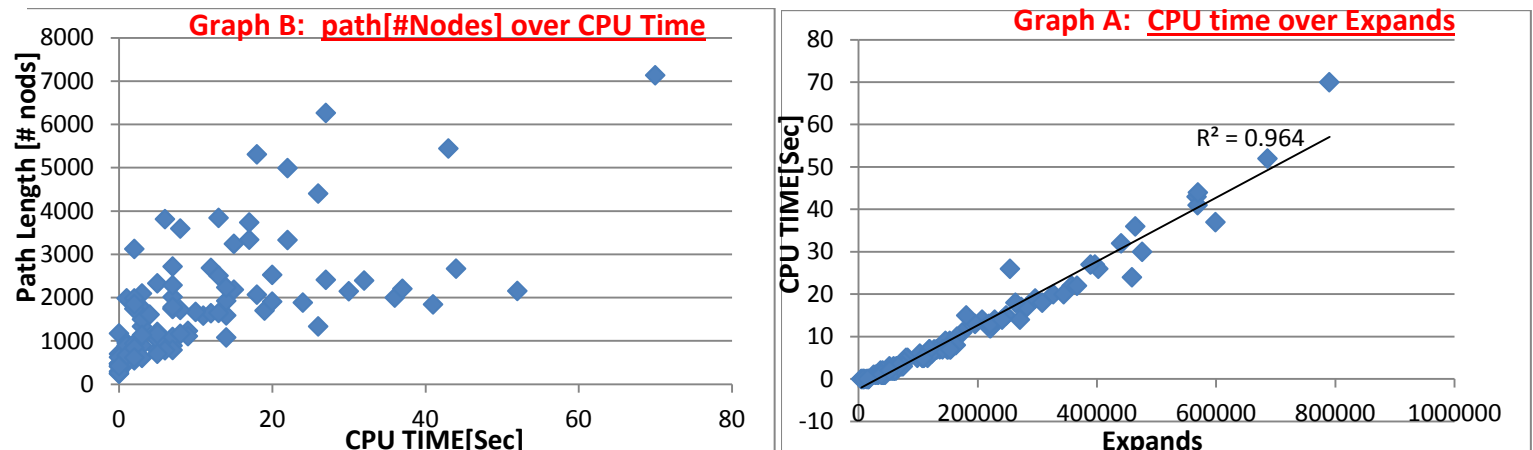


## הקדמה

1. גרף המצבים בבעיית הניווט מכיל צמתים (Junctions) כמפגשי רחובות וקשתות המייצגות את הדרכים ביניהם. בתרגום לבעיית החיפוש, לכל צומת יש ID שהוא מזהה הצומת ולכל קשת יש משקל שהוא עלות לנסיעה בדרך זו. חישוב משקלה של הקשת תלוי בגורמים שונים התלויים לפי מטרת הבעיה.  
לדוגמא, עבור מציאת הדרך הקצרה ביותר, משקל הקשת יזדקק רק למרחק.  
עבור מציאת הדרך המהירה ביותר, משקל הקשת יחושב לפי המרחק והמהירות הממוצעת בקשת.  
ואילו עבור הדרך החסכונית ביותר בדלק נזדקק למרחק, למהירות ולסוג הרכב של המשתמש.  
אנו רוצים למזער את כמות המידע שיש בצומת חיפוש (מעבר למידע שיש בצומת בגרף) מהתובנה שאנו רק מחפשים במרחב המצבים את הדרך הקצרה ביותר (לפי המטריקה המבוקשת), כלומר החיפוש הוא חיפוש במפה קיימת – כלומר המידע של הצמתים כבר נמצא בזכרון (להבדיל מבעיות כמו שח מט שמרחב המצבים לא קיים מראש). אז בשביל להקטין את משאבי הזכרון אנו נקטין את התקורה של המידע שנשמר בצומת חיפוש מעבר למידע שנשמר בצומת בגרף המצבים.  
לכאורה, מרחב המצבים הוא קטן (פחות ממיליון מצבים), לכן הבעיה אמורה להיות קלה יחסית. כמו כן, מקדם הסיעוף אינו גדול מאוד. אך, לפי הסטטיסטיקות הכבישים ברובם הם קצרים. בנוסף, היוריסטיקות שלנו הן קבילות ולכן כוחן חלש יחסית. לפיכך, החיפוש לא יהיה כל כך מהיר.  
לפיכך, אני נשמור סט מינימאלי של פרמטרים על העץ עצמו, ונשתמש ב-ID של הצומת ובפונקציות ב-CountryMap כדי לחשב ולשלוף את הנתונים הנדרשים לחישוב משקלה של קשת, או לחישוב אלמנטים בהיוריסטיקה.
2. להלן פונקציות המחיר השונות עבור הקשתות בגרף (links):
  - עבור Shortest Route, המחיר הוא מרחק הדרך (distance).
  - עבור Quickest Route המחיר הוא הזמן הנדרש לעבור קטע דרך כלשהו
  - עבור Fuel Saving Route המחיר הוא נצילות הדלק של הרכב בהתחשב בסוג הרכב ומהירות הנסיעה בדרך הנוכחית (ביחידות של ליטרים ליחידת מרחק) כפול מרחק הנסיעה.
3. היוריסטיקות קבילות
  - a. מרחק אווירי לצומת מטרה. ידוע כי המרחק האווירי הוא המרחק הקצר ביותר בין שתי נקודות כלשהן. לכן בהכרח עבור המסלול האופטימאלי  $P = \langle n_1, \dots, n_k \rangle$  המרחק האווירי  $h(n_1, n_k)$  קטן או שווה מאורך המסלול P כאשר עוברים בקשתות המסלול. לכן היוריסטיקה קבילה.
  - b. מרחק אווירי לצומת מטרה מחולק ב-120 קמ"ש (120 קמ"ש היא המהירות המקסימלית בגרף). ברור כי כל מסלול אופטימאלי עוברים מרחק גדול או שווה למרחק האווירי בין הצומת הנוכחי לסיום. כמו כן בכל קטע דרך, המהירות הינה קטנה או שווה ל-120. לכן, זמן הנסיעה במסלול האופטימאלי מהצומת הנוכחי לסיום בהכרח קטן או שווה להיוריסטיקה. מכאן שהיא קבילה.
  - c. מרחק אווירי לצומת מטרה כפול הניצולת המינימלית (ביחידות של ליטרים ליחידת מרחק) של סוג הרכב הנתון. בכל קטע דרך נצילות הדלק גדולה או שווה מהנצילות האופטימאלית עבור סוג רכב כלשהו בדרך זו – יש מינימום גלובלי לנצילות כפונקציה של המהירות. מכיוון שהמרחק האווירי בהכרח קטן או שווה מאורך הדרך מהצומת הנוכחי למטרה, הרי שכמות הדלק במסלול האופטימאלי מהצומת הנוכחי למטרה בהכרח גדולה או שווה לכמות הדלק המחושבת ע"י היוריסטיקה. מכאן שהיא קבילה (אינטואיטיבית, היא אופטימית כי מניחה שהמסלול הוא הקצר ביותר האפשרי – מרחק אווירי, והיא מניחה שאפשר לנסוע במהירות האידיאלית שתאפשר צריכת דלק מינימלית).

4. תוצאות ההרצה על 100 מופעים נמצאים בטבלה results\_11012013\_4.xlsx המצורפת. אנו בחרנו את פרופיל רכב 1 להיות פג'ו. פרופיל רכב 2 להיות פורד פוקוס ואת המשקלים להיות 0.3,0.3,0.4. נתבונן במספר גרפים מייצגים (על מנת לחסוך במקום, במידה ובמספר מטריקות התקבלו גרפים דומים אנו נציג רק את הגרף עבור מטריקה אחת – המטריקה של דרך קצרה ביותר במטרים).

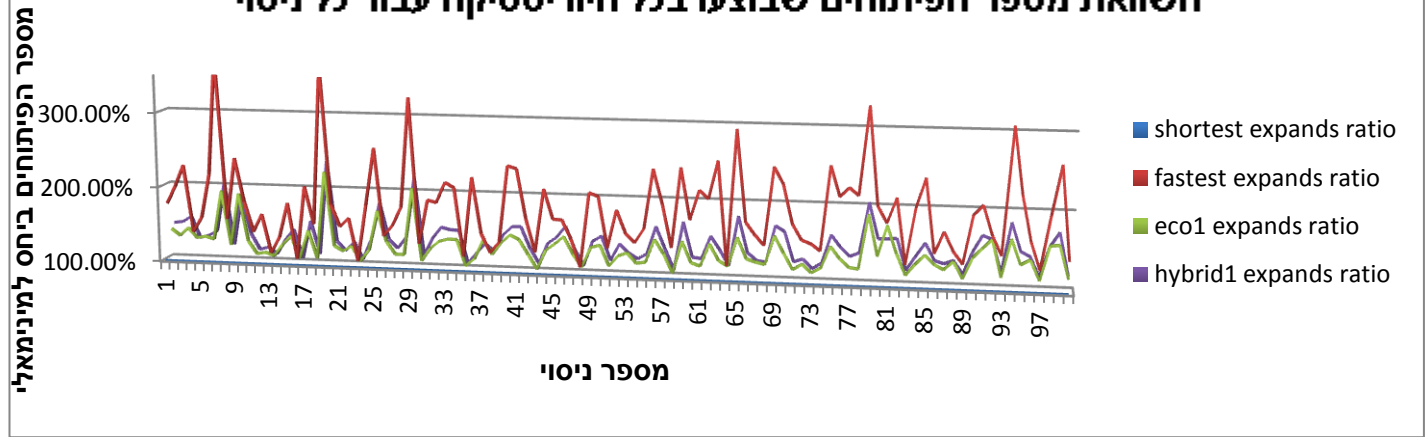
א. גרף **A** הבא מציג את זמן החישוב בשניות כתלות במספר הקריאות לexpand של האלגוריתם (זה Scatter plot X,Y). אנו מצפים כמובן ליחס ישר וזה אכן מה שהתקבל (קו המגמה ממחיש זאת). התוצאות בגרף הן עבור המטריקה של דרך קצרה ביותר במטרים. במטריקות האחרות התקבלו גם כן תוצאות המעידות על יחס ישר. זו כמובן תוצאה צפויה מכיוון שהגיוני שזמן החיפוש יהיה תלוי במספר הפיתוחים שנעשו מכיוון שמקדם לא מאוד משתנה במהלך החיפוש (סביר שבתצפיות המעטות שרחוקות מהקו היו סטיות ממקדם הסיוע הממוצע בגרף).



ב. בגרף **B** לעיל, ניתן לראות כי לא קיים קשר חזק בין מספר הצמתים במסלול שהוחזר לזמן החישוב של המסלול בשניות. המשמעות של כך היא שאורך המסלול שהוחזר לא מעיד באופן מדויק על כמות ה"פיתוחים המיותרים" שנעשו. במידה והיוריסטיקה הייתה מושלמת היינו מצפים שהקשר הזה יהיה חזק. במקרה כזה, אנו הולכים **עבור על מופע של הבעיה**, "ישירות למטרה" ולכן היה מתקבל יחס ישר. קשר חלש, מעיד על אי אחידות של ההיוריסטיקה שלנו. זה מעיד שיש מופעים שהיא כנראה יותר "חלשה עבורם". יש בכך הגיון, ההיוריסטיקות שלנו קבילות לכן הן צפויות להיות די חלשות באופן כללי. למשל היוריסטיקה של מסלול קצר במטרים, עלולה לפעמים לבחור במהלך הפיתוח דרכים שמקרבות אווירית למטרה, אבל ייתכן שאחר כך לא ניתן בכלל להגיע כי אין קשירות ליעד הסופי בדרך שפותחה עד כה. יחד עם זאת, לא תמיד דבר כזה יקרה ולכן אנו רואים קשר חלש.

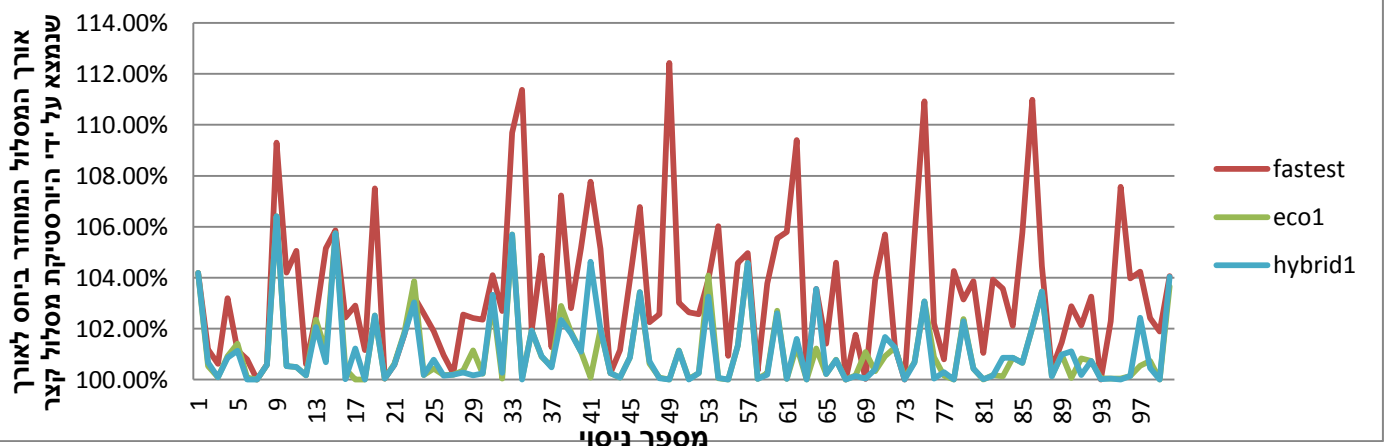
ג. הגרף הבא, מציג את היחס של מספר הצמתים שפותחו בכל מטריקה למספר הצמתים שפיתחה המטריקה שפיתחה הכי פחות צמתים למופע זה של הבעיה. אם עבור כל זוג כל המטריקות היו מפתחים פחות או יותר אותו מספר של צמתים, היינו מצפים לקבל 4 עקומות שנראות בקירוב כמו פונקציה קבועה מסביב ל100 אחוז. זה לא המצב. ניתן לראות באופן מובהק כי תמיד ההיוריסטיקה של מסלול קצר ביותר במטרים, מפתחת הכי מעט צמתים. האחרות, גרועות ממנה בסדר גודל של יותר מעשרות אחוזים! ההיוריסטיקה של מסלול מהיר ביותר, היא הכי גרועה מהבחינה הזו. אנו לומדים מכאן שההיוריסטיקה של מסלול קצר ביותר היא הטובה מבין כל האחרות, מכיוון שהיא מפתחת פחות צמתים מכולם. זה הגיוני כי כנראה שמרחק אווירי הוא מדד טוב יחסית ל"כמה אתה קרוב לנקודת המטרה". מצד שני, ההיוריסטיקות האחרות מושפעות ממהירות הנסיעה בכביש, אשר משתנה בין הדרכים השונות, ומכיוון שההיוריסטיקות האחרות הן למעשה "מרחק אווירי בתחפושת" (מרחק אווירי כפול פקטור קבוע כלשהו), הן אינן מוצלחות במיוחד – אין להן דרך לחזות מראש למשל שעדיף לסוע בין רעננה לתל אביב דרך כביש 2, גם אם זה אומר לסוע בדרכים פחות מהירות או אפילו להתרחק מהיעד בשביל להשתלב בכביש 2. (זאת מכיוון שכביש 4 הוא ממש קרוב לרעננה). לגבי ההיוריסטיקה ההיברידית, לאור בחירת המשקולות סביר שהיא תהיה איפשהו בין ההיוריסטיקות האחרות וזה אכן המצב. ההיוריסטיקה של חסכון בדלק מושפעת אמנם ממהירות הנסיעה בדרך שמפותחת אבל גם בסוג הרכב עצמו וצריכת הדלק שלו. בקוד שניתן לנו, ניתן לראות שנצילות הדלק של הרכב לא משתנה בהרבה (באחוזים) בין המהירויות השונות ממילא. לפיכך, צפוי שההיוריסטיקה של חסכון בדלק לא תהיה גרועה כמו זו של מסלול מהיר בזמן כי מהירויות הכבישים משתנות מאוד במפה.

## השוואת מספר הפיתוחים שבוצעו בכל היוריסטיקה עבור כל ניסוי

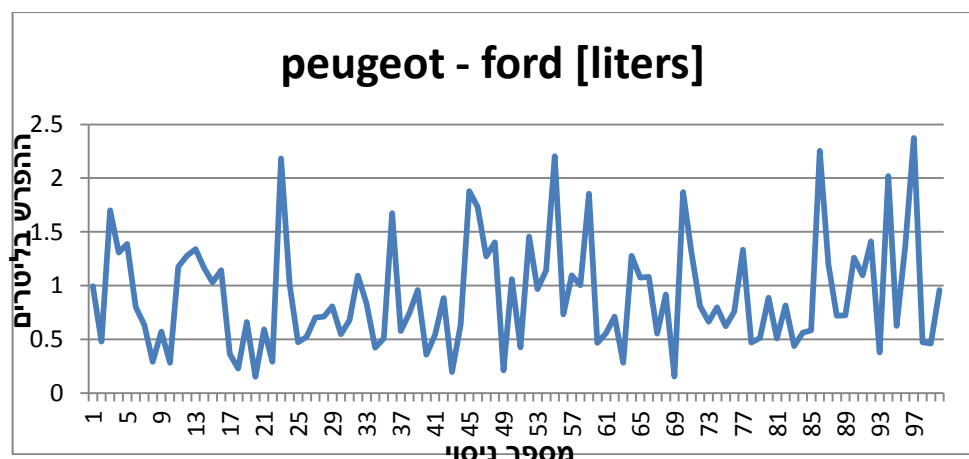


ד. בגרף הבא ניתן ללמוד בכמה יש להאריך את הדרך במטרים על מנת לקבל את המסלול האופטימלי עבור כל אחת מההיוריסטיקות שאינן מסלול קצר ביותר במטרים (פרופיל רכב 1 בלבד, השני הושמט מחוסר עניין). הגרף מראה שלא צריך להוסיף הרבה לדרך במטרים בשביל לקבל מסלול אופטימלי עבור היוריסטיקה אחרת. זה הגיוני. למשל כי למרות השונות הגבוהה במהירויות, אין מסלול שניתן לסווע בו במהירות מאוד מאוד גבוהה בממוצע ושהוא יהיה משמעותית ארוך יותר מהמסלול האחר (זה סביר כי הדרכים הן ברובן קצרות זאת אומרת שבשביל להגיע לכביש מהיר לא צריך לסווע דרך ארוכה בכיוון ההפוך ליעד). רואים שהמסלול המהיר ביותר הוא זה שיחסית ליתר ההיוריסטיקות יש ל"שלם הרבה בהארכת הדרך" בשביל להגיע למסלול כזה. זה הגיוני מסיבה שכבר הזכרנו – המסלול החוסך בדלק לא רגיש כמעט לשינויים במהירויות כמו המסלול שחוסך בזמן. הרי, אחד הפקטורים במסלול שחוסך בדלק הוא ממילא הק"מ וכלל שמארכים את הדרך אם לא משנים את המהירות בהרבה ו/או צריכת הדלק לא משתנה אז זה לא כדאי כי העלות תהיה גבוהה יותר.

## השוואת אורך המסלול המוחזר של כל היוריסטיקה ביחס לאורך המוחזר על ידי היוריסטיקת המסלול הקצר ביותר

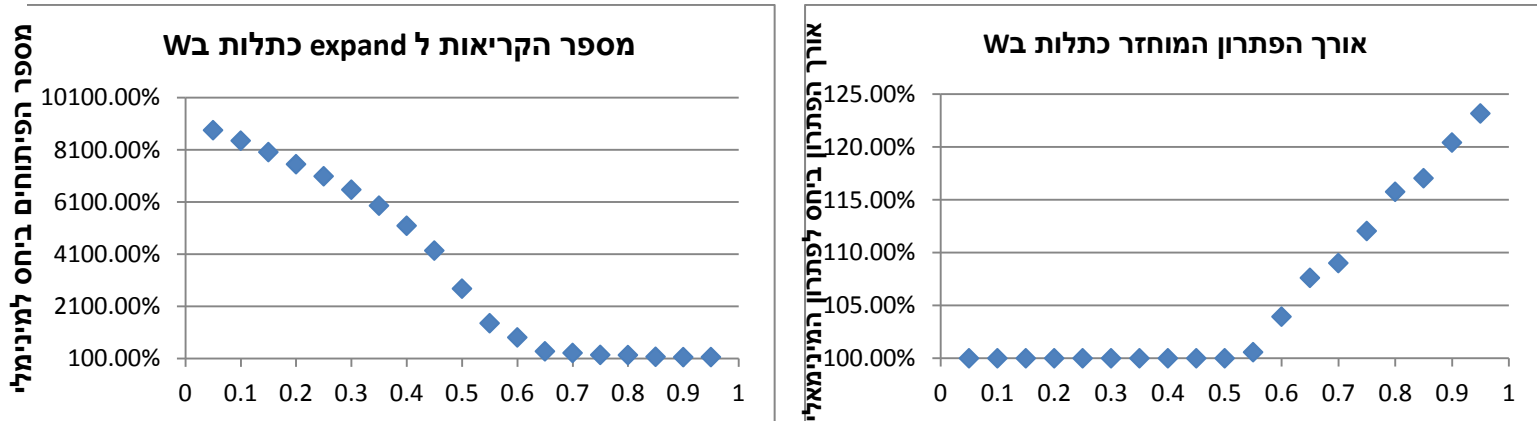


ה. בגרף הבא, רואים שפג'ו הוא רכב פחות יעיל מבחינת צריכת דלק מפורד פוקוס. הגרף מראה את ההפרש בין מחירי המסלול האופטימלי מבחינת צריכת דלק לכל מופע של הבעיה עבור שני סוגי הרכבים. מכיוון שההפרש הוא תמיד אי שלילי (והוא יותר מחצי ליטר כמעט תמיד) אז אפשר לחסוך יותר בדלק אם נוסעים בפורד. זה מתיישב כמובן עם הנתונים בקוד שמראים שהנצילות הטובה יותר היא של פורד. הסיבה ל"ספייקים" בגרף היא ככל הנראה העובדה שהנצילות הדלק של שני הרכבים לא שונה בכל מהירות. כלומר, יש מהירויות שבהן הן שווים. אז, ברור שלפעמים ההבדל לא יהיה גדול יחד עם זאת – זה תלוי במהירויות האפשריות בלינקים הספציפיים הרלוונטים למסלול המסוים – כלומר זה משתנה בין בעייה לבעייה ולכן ה"קפיצות" בגרף.



5. תוצאות ההרצה על 40 מופעים של הבעיה עבור 19 ערכים של  $W$  בין 0 ל 1 נמצאים בטבלה results\_11012013\_5.xlsx. פרופיל הרכב כאן הוא פג'ו. המשקלים עבור ההיברידי הם 0.3,0.3,0.4. נתבונן בשני גרפים מייצגים עבור המטריקה של דרך קצרה ביותר במטרים בלבד (אין הבדלים מהותיים בסעיף זה בין המטריקות השונות). בסעיף זה בחנו:

- כיצד שינוי הפרמטר  $W$  משפיע על אורך הפתרון שאלגוריתם  $A^*$  הממושקל מחזיר (מחיר המסלול המוחזר על ידי  $A^*$  (weighted)). – כלומר בחנו את אופטימליות הפתרון המוחזר כתלות ב  $W$ . זה מה שמוצג בגרף הימני למטה.
- כיצד שינוי הפרמטר  $W$  משפיע על כמות הזמן שלוקח לאלגוריתם  $A^*$  הממושקל להחזיר פתרון. במילים אחרות, כיצד קביעה של  $W$  משפיעה על אורך החיפוש (למדנו כבר כי אורך זמן החיפוש הוא יחס ישר של מספר הקריאות (expand)).



### הסבר לגבי משמעות הגרפים והצורה שבו יוצרו:

הגרפים הנ"ל יוצרו באופן דומה. נסביר כיצד יצרנו את הגרף הימני למעלה. בדקנו, עבור כל בעיה, מה ההערך המינימלי מבחינת אורך הפתרון שהוחזר מבין כל ערכי  $W$ . לאחר מכן מצאנו את היחס בין הערך המינימלי ליתר הערכים שהוחזרו עבור כל הערכים של  $W$ , זה מראה פי כמה יותר גרוע מהמינימום האורך שהוחזר עבור כל ערך של  $W$  לבעיה מסוימת. קיבלנו, 19 מספרים שמייצגים באופן יחסי פי כמה על ערך של  $w$  גרוע מהמינימום עבור בעיה מסוימת. לאחר מכן, מצענו את המספרים האלו על כל 40 הבעיות. סה"כ קיבלנו הערכה לפי כמה יותר גרוע אורך הפתרון המוחזר כתלות ב  $W$ . המשמעות של 100% היא שזה המינימום – כלומר הכי טוב.

### ניתוח המשמעות של התוצאות הנ"ל:

המשמעות של הגרף הימני למעלה היא שכל שמגדילים את  $W$  (כלומר מסתמכים יותר על ההיוריסטיקה, יותר דומה ל greedy best first), אז הפתרון יותר גרוע כלומר פחות אופטימלי. כזכור,  $W=1$  זה חיפוש greedy best first, שאיננו בהכרח אופטימלי (למשל אם ההיוריסטיקה לא מושלמת, כמו במקרה שלנו). יחד עם זאת greedy best first, הוא כן מהיר יחסית כי מתקדם מהר לפתרון בלי לבדוק אלטרנטיבות, כי מתבסס רק על הפונקציה ההיוריסטית, כלומר כל פעם מתקרב קצת וממשיך הלאה בצורה חמדנית.  $W=0$  זה חיפוש uniform cost (שהוא אופטימלי מבחינת הפתרון שמוחזר) אבל יכול להיות מאוד איטי כי הוא לא מסתמך על היוריסטיקה כלל (אז החיפוש שלו "לא החלטי" כלומר הוא מפתח במקביל את כל המסלולים לפי המחיר ששולם עד החזית).

לפיכך הערכים שקיבלנו בקצוות הגיוניים. כמו כן, הערך בגרפים עבור 0.5 הגיוני כי הוא מבחינת אורך פתרון הוא מינימלי (זה  $A^*$  רגיל ולכן אופטימלי עם היוריסטיקה קבילה). מבחינת זמן החיפוש הוא גם הגיוני כי ערכים קטנים יותר מ 0.5 יהיו אופטימליים (כמו 0.5 אבל איטיים יותר כפי שלמדנו בהרצאה). לגבי הערכים בין 0.5 ל 1 כפי שלמדנו בהרצאה, הם אמורים להאיץ את זמן החיפוש אבל פוגעים באופטימליות הפתרון. לפיכך, כל התוצאות שאנו רואים בגרף מאוד הגיוניות ומתיישבות עם החומר התאורטי שלמדנו בהרצאה.

### חלק ב תוצאות וניתוחים

1. האלגוריתם המוצא על ידנו מנסה למזער את מספר הצעדים הנלקחים בעקבות עדכון קשת כלשהי בגרף, בעת ריצת AStar.

האלגוריתם מפריד בין שני סוגי עדכונים של קשת:

- א. ירידה במחיר קשת כלשהי – האלגוריתם יחזיר את הצומת העליונה של הקשת חזרה ל-Open, וימשיך בריצה רגילה של AStar.
- ב. עליה במחיר קשת – במקרה זה נריץ DFS על תת העץ הנפרס מהצומת העליונה של הקשת עד לחזית החיפוש הנוכחית (כל צמתי Open), שיעדכן את מחירי המסלולים בעקבות השינוי. בהגעה לכל צומת ב-DFS נשאל האם יש לשנות את האב המצביע על הצומת, כיוון שיתכן שכעת יותר כדאי להגיע לצומת חיפוש כלשהו ממסלול אחר דווקא. לשם כך, נחזיק מיפוי בין כל מצב (state) לבין כל צמתי החיפוש (Node) הידועים עבורו

(בין אם הם Open או Closed). עבור כל צומת חדש ש-DFS מגיע אליו, נשאל מיהו צומת החיפוש (תזכורת: צומת חיפוש = מצב + צומת החיפוש האב + מחיר המסלול מהשורש עד אליו) בעל המשקל הכדאי ביותר. בפעולה זו נזכור לא לשקול את צומת החיפוש שמסלולו נפסל בעקבות העדכון. לאחר שה-DFS יסיים, נמשיך בריצה רגילה של ה-AStar.

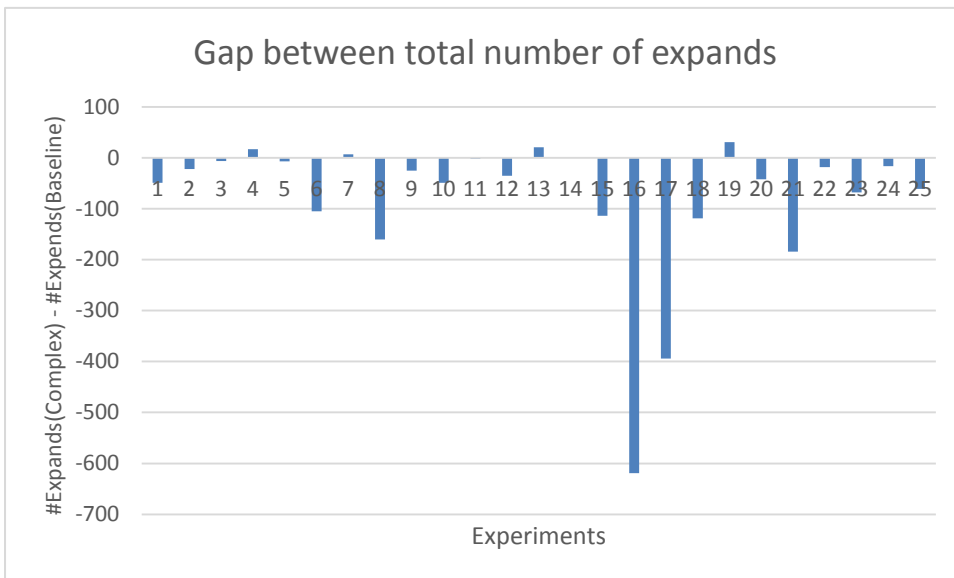
## 2. הוכחת נכונות (קבילות)

נפריד למקרים לפי סוג עדכון הקשת:

- נשים לב שירידה במחיר קשת אינה יכולה לשנות מסלול בין צומת ב-Closed לבין צומת ב-Open, שכן המסלול גם לפני השינוי היה הקל ביותר, וכעת הוא נעשה קל אפילו יותר. מכאן שמרגע המעבר של צמתי הקשת ל-Open מובטח שצמתים אלו יהיו ראשונים בתור ה-Open (כיוון שתת המסלול מהשורש אליהם, קל יותר מכל מסלול מהשורש לחזית הקודמת) ולכן AStar יעדכן את עצמו במהלך ריצתו.
- עבור מקרים בהם משקל הקשת עלה התמונה יותר מסובכת, כיוון שכעת כל תת העץ הנפרס מהצמתים שעודכנו עד החזית כבר לא מעודכן, ומסלולים אל צמתי החיפוש בו יכולים להשתנות. לשם כך נריץ את ה-DFS עד החזית. נשים לב שה-DFS יגיע אל כל צומת שמושפע בצורה זו או אחרת מהשינוי במשקל הקשת ויעדכן מסלולים במקרה הצורך. העדכון יסתיים ברגע שהגענו אל החזית הנוכחית של AStar ועדכנו את מחירי המסלולים בה (למעשה החזרנו תור Open חדש). מרגע זה AStar יכול להמשיך כרגיל, והנכונות נובעת מכך שכל צומת ב-Open מייצג מסלול חיפוש זול ביותר מהשורש אליו.

## 3. ניתוח ביצועי האלגוריתם:

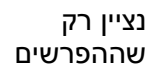
- השווינו את ביצועי האלגוריתם כנגד אלגוריתם ה-Baseline המתחיל את ריצתו מהתחלה בכל פעם שמתבצע עדכון. ביצענו שתי השוואות של האלגוריתמים, כאשר בכל אחת נבחנו 25 זוגות מסלולים זהים. כיוון שאין דרך אמיתית להגדיל את אותם קשתות עדכון בכל ריצה של שתי האלגוריתמים (כיוון שהם תלויות בצמתים שפותחו בכל ריצה) אנו משווים את הטרנד של כל הריצות מתוך תקווה שאותו מספר ממוצע של עדכונים פותח בשני האלגוריתמים.
- ההרצה הראשונה איפשרה מספר כלשהו של עדכונים מפונקציית GetSpeedUpdates. פה איבחנו יתרון קל ל-Baseline במספר פיתוחי הצמתים. אנו מייחסים זאת לעובדה שעדכון המכיל מספר צמתים גורם להתחלה חדשה אחת של Baseline, בעוד האלגוריתם שלנו מפתח תת עץ שלם עבור כל עדכון (עבור עדכון חיובי יפותח התת עץ במסגרת ה-DFS, ואילו עבור עדכון שלילי הוא יעודכן במסגרת המשך ריצת ה-AStar).



גרף 1 הפרש פיתוחי הצמתים בין Baseline ו-Complex, מספר קשתות בעדכון

גרף 1 הפרש פיתוחי הצמתים בין Baseline ו-Complex מתאר את ההפרש במספר פיתוחי הצמתים בין Baseline לבין האלגוריתם שלנו Complex. הפרש שלילי מצביע על יתרון ל-Baseline במספר פיתוחי הצמתים (פיתח פחות צמתים). כפי שניתן לראות הממוצע מצביע על יתרון ל-Baseline.

כפי שניתן לראות מגרף 2 הפרש פיתוחי הצמתים בין Baseline ו-Complex, קשת אחת לעדכון כאן כבר ישנו יתרון קל ל-Complex, שכן כל עדכון מפתח מחדש רק תת-עץ בגרף ולא את כל העץ מהתחלה.



הנ"ל הם לרוב זעומים יחסית למספר הצמתים הכולל שפותח בכל אלגוריתם, ולרוב מסתכמים בכ--0.05%. 0.01 מכלל הצמתים שמפותחים. הסיבה לכך היא ההגבלה על מספר הקשתות שמתעדכנות (10) הינה קטנה במיוחד ביחס לאורך המסלול הממוצע. כנראה שעבור מספר הגרלות רב יותר הטרנד יהיה יותר ברור לטובת Complex.