מגישים:

אפרת כהן | 207783150

ליאל צרפתי | 312350622

**Analyzing Tie-Breaking Strategies for the A\* Algorithm**

**Lab Report**

תוכן עניינים

[הקדמה 2](#_Toc122942469)

[מטרת הניסוי 2](#_Toc122942470)

[תיאור מהלך הניסוי 3](#_Toc122942471)

[תוצאות הניסוי 4](#_Toc122942472)

[בעיית הפאזל: 5](#_Toc122942473)

[בעיית המבוך 8](#_Toc122942474)

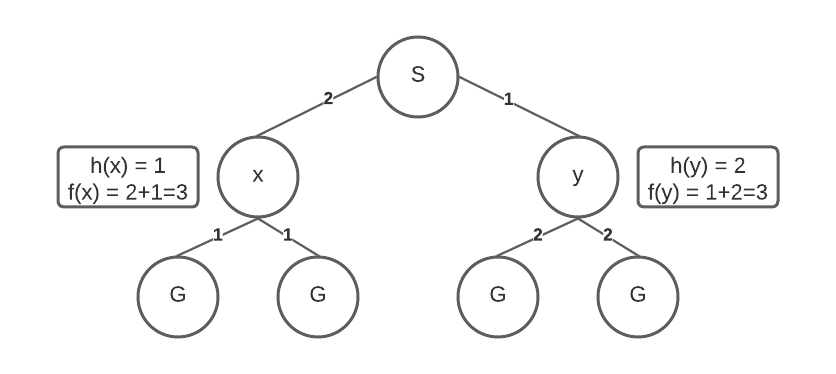
[מסקנות 9](#_Toc122942475)

[נספחים 9](#_Toc122942476)

# הקדמה

**אולי להוסיף הסבר על הבעיות**

אלגוריתם A\* הוא אלגוריתם חיפוש היוריסטי שמשתמש בתור עדיפויות (open list) כדי להרחיב מצבים חדשים (שהוכנסו לתור ע"י ההרחבה של המצב הקודם אליהם ("האבא")), על מנת להגיע ממצב התחלתי S למצב סופי G. תור העדיפויות מתנהל לפי פונקציה f המחושבת עבור כל קודקוד n כסכום של g(n) שהוא עלות הנתיב ממצב S ל-n, ו-h(n) שהוא העלות ההיוריסטית למצב היעד G. אלגוריתם A\* יחזיר פתרון אופטימלי עם עלות מינימלית C\* (במידה וקיים פתרון) במקרה שבו היוריסטיקה h היא אדמיסבילית, קרי הפונקציה לא תעריך את העלות למצב היעד מעבר לעלות האמיתית.

במהלך החיפוש ייתכן שב-open list יהיו כמה מצבים עם אותו ערך של f. במקרה כזה צריך לשבור את השוויון ולבחור רק קודקוד אחד כדי להרחיב. הדעה הרווחת בספרות היא שיש לבצע את שבירת השוויון לטובת מצבים עם ערכי h קטנים יותר. דוגמה להשפעה שיכולה להיות לשבירת שוויון ניתן לראות בדוגמה הבאה:

בעת הרחבת מצב S יכנסו לתור המצבים x ו-y עם ערכי f שווים (ל-3). כעת כדי לבחור איזה מצב להרחיב צריך לשבור את השוויון. אם נבחר לשבור את השוויון לפי ערכי h אזי מצב x יורחב ונקבל פתרון אופטימלי. מנגד, הרחבה של מצב y (באם לא נשתמש בערכי h לשבירת השוויון) תביא לפתרון לא אופטימלי.

כלל האצבע של אסטרטגיית שבירת השוויון לפי המינימום של ערך h היא יעילה במקרים רבים, אך יש לבחון האם היא תמיד תשיג את התוצאות הטובות ביותר בהשוואה לשיטות אחרות.

# מטרת הניסוי

*בניסוי זה אנחנו באים לבחון שיטות שונות לשבירת שוויון. נרצה לבחון את האסטרטגיות על בעיות חיפוש שונות, ולמדוד ביצועים של כל שיטה, לפי כמה פרמטרים.*

*ההשערה היא שנקבל כי השיטה הקיימת לשבירת השוויון באמצעות ערכי h תביא לביצועים הטובים ביותר, אבל ייתכן כי נמצא שיטות אחרות היעילות גם כן במצבים מסוימים.*

*בנוסף, נרצה לבדוק האם יש צורך בהגדרה נוספת לשבירת שוויון ברמה 2. כלומר, במידה והשוויון לא ישבר גם לאחר הפעלת האסטרטגיה הראשונה (לאחר השוואת ערכי f), יש צורך בשבירת שוויון נוספת. נרצה למדוד את שכיחות מקרים אלו, ולהראות שגם לאסטרטגיה של שבירת שוויון ברמה 2 יש חשיבות.*

*ניסיון זה מתבצע בהשראת מאמרים העוסקים בתחום:*

1. Asai, Masataro, and Alex Fukunaga. "[Tie-breaking strategies for cost-optimal best first search](https://www.jair.org/index.php/jair/article/view/11039)." *Journal of Artificial Intelligence Research* 58 (2017): 67-121.
2. Corneil, Derek G., et al. "[A tie-break model for graph search](https://www.sciencedirect.com/science/article/pii/S0166218X15002978)." *Discrete Applied Mathematics* 199 (2016): 89-100.
3. Corrêa, Augusto B., André Grahl Pereira, and Marcus Ritt. "[Analyzing Tie-Breaking Strategies for the A\* Algorithm](https://icaps18.icaps-conference.org/fileadmin/alg/conferences/icaps18/workshops/workshop02/docs/HSDIP2018.pdf#page=12)." *IJCAI*. 2018.
4. Asai, Masataro, and Alex Fukunaga. "[Tiebreaking strategies for A\* search: How to explore the final frontier](https://ojs.aaai.org/index.php/AAAI/article/view/10071)." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. No. 1. 2016.

# תיאור מהלך הניסוי

**תיאור מהלך הניסוי: איזה קוד כתבתם, איך התבצעו הניסויים (כמה הרצות, איזה מחשב, איזו בעיה פתרתם וכו')**

*הגדרנו מספר אסטרטגיות לשבירת שוויון: להוסיף נימוק והגיון לכל שיטה!*

1. *LIFO – last in first out – שבירת השוויון תהיה לפי המצב שנכנס באחרונה ל-open list.*
2. *FIFO – first in first out – נרחיב את המצב שנכנס לתור ראשון מבין המצבים עם אותו ערך f.*
3. *H-value – שבירת שוויון לפי ערך h. את השיטה הזאת בחנו בכמה פרמוטציות שונות לשבירת שוויון ברמה 2:*
   1. *H-LIFO – במידה ויש שוויון גם בערכי h, יבחר המצב שנכנס אחרון לתור מבין המצבים עם השוויון.*
   2. *H-FIFO - במידה ויש שוויון גם בערכי h, יבחר המצב שנכנס ראשון לתור מבין המצבים עם השוויון.*
   3. *H-G – ערכי g, כלומר המצב הקרוב ביותר למצב ההתחלתי יורחב בעת שוויון בערכי h בנוסף לשוויון בערכי f.*

# תוצאות הניסוי

*נציג את זמני הריצה עבור כל בעיה ושיטת שבירת שוויון:*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Run Time (sec) | | | | | | | | | | |
| Tie-Breaking method | **DEPTH-F** | **DEPTH-L** | **FIFO** | **G-H** | **H-FIFO** | **H-G** | **H-LIFO** | **H-RAND** | **LIFO** | **RANDOM** |
| Domain (heuristic function) |
| Puzzle (Manhattan distance) |
| 4x4.a | 424.652 | 781.779 | 743.383 | 782.330 | 67.894 | **65.804** | 73.939 | 70.567 | 363.289 | 680.101 |
| 4x4.b | 2702.945 | 9142.035 | 8838.821 | 9591.882 | **326.651** | 327.292 | 331.975 | 327.643 | 863.001 | 7908.499 |
| 4x4.c | 1.362 | 4.8227 | 5.0883 | 5.3943 | 0.2192 | 0.2271 | 0.2117 | **0.2018** | 3.7217 | 3.9112 |
| 4x4.d | 0.071 | 0.189 | 0.216 | 0.251 | 0.020 | 0.020 | 0.018 | **0.017** | 0.129 | 0.182 |
| 4x4.e | 14.480 | 29.810 | 28.737 | 30.636 | 2.531 | 2.364 | 2.413 | **2.354** | 18.048 | 25.327 |
| Maze (Euclidean distance) |  |  |  |  |  |  |  |  |  |  |
| 25X25 | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 | **0.003** | 0.004 |
| 125X125 | 1.513 | 1.529 | 1.498 | 1.434 | 1.442 | 1.534 | 1.425 | 1.506 | 1.448 | **1.362** |
| 225X225 | 15.677 | 15.001 | **14.562** | 16.283 | 16.417 | 15.792 | 15.652 | 15.208 | 16.275 | 15.939 |
| 325X325 | 82.635 | 78.092 | 68.226 | 81.666 | 74.285 | **66.909** | 68.727 | 72.221 | 69.119 | 83.222 |
| Maze (Manhattan distance) |  |  |  |  |  |  |  |  |  |  |
| 25X25 | 0.004 | 0.006 | 0.006 | 0.005 | **0.003** | 0.004 | 0.004 | 0.005 | 0.005 | 0.005 |
| 125X125 | 1.672 | 1.161 | 0.990 | 0.990 | 1.383 | 0.969 | 1.076 | **0.824** | 1.131 | 1.482 |
| 225X225 | 11.414 | 11.091 | 11.351 | 11.434 | **7.940** | 12.697 | 14.313 | 8.850 | 9.970 | 11.555 |
| 325X325 | 61.151 | 68.556 | 47.897 | 53.047 | 49.419 | 48.773 | 39.222 | **37.122** | 41.581 | 48.477 |

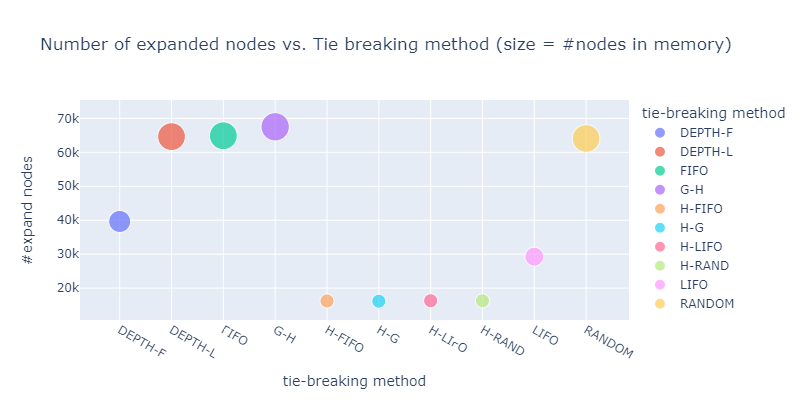
*בנוסף, נתסכל על הזיכרון שהיה בשימוש (מספר המצבים שנשמרו בזיכרון עד למציאת הפתרון) עבור כל בעיה:*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nodes in Memory | | | | | | | | | | |
| Tie-Breaking method | **DEPTH-F** | **DEPTH-L** | **FIFO** | **G-H** | **H-FIFO** | **H-G** | **H-LIFO** | **H-RAND** | **LIFO** | **RANDOM** |
| Domain (heuristic function) |
| Puzzle (Manhattan distance) |
| 4X4.a | 109013.7 | 149487.3 | 150841 | 150892 | 43133 | **42643** | 43952 | 43753.67 | 100250 | 150339 |
| 4X4.b | 250204.7 | 424748.3 | 424005 | 447197 | 103099 | **103036** | 103052 | 103069 | 147807 | 419138.3 |
| 4X4.c | 5724.667 | 11637 | 12406 | 12688 | 2290 | 2314 | **2270** | 2302 | 10812 | 11191.67 |
| 4X4.d | 1194.333 | 2186.667 | 2476 | 2585 | 501 | 484 | **481** | 495.3333 | 1649 | 2307.667 |
| 4X4.e | 20170 | 29807.33 | 30056 | 30205 | 8060 | **7991** | 8034 | 8015.667 | 22126 | 29842.67 |
| Maze (Euclidean distance) |  |  |  |  |  |  |  |  |  |  |
| 25X25 | 269.2 | 278.1 | 282.4 | **254.2** | 280.1 | 263.4 | 264 | 258.7 | 256.2 | 265.8 |
| 125X125 | 6794.5 | 6897.4 | 6859.2 | 6631.7 | 6733.2 | 6870.9 | 6677.8 | 6820.8 | 6709 | **6531.5** |
| 225X225 | 22123.5 | 21943.6 | **21550.2** | 22743.3 | 22819.3 | 22370.8 | 22320.4 | 22030.7 | 22576.4 | 22537 |
| 325X325 | 46561.7 | 46832.3 | 45525 | 47209.4 | 47481.1 | **44382** | 45854.3 | 45085.2 | 46069.8 | 47005.7 |
| Maze (Manhattan distance) |  |  |  |  |  |  |  |  |  |  |
| 25X25 | 181.3 | 220.8 | 218.3 | 220.5 | **157.7** | 174.6 | 163.1 | 217 | 178.1 | 217.8 |
| 125X125 | 5798.2 | 4565.2 | 4155.4 | 4091.7 | 4815.8 | 4049.3 | 4412.3 | **3684.6** | 4410.6 | 5331.8 |
| 225X225 | 14769 | 14674.8 | 15002.7 | 15205.6 | **11817.2** | 15472.3 | 17381.3 | 12538.9 | 13802.7 | 15206.4 |
| 325X325 | 35469.8 | 37559.8 | 31076.1 | 33146.9 | 31431 | 31720.1 | 27212.5 | **26911.8** | 29229.8 | 30939.4 |

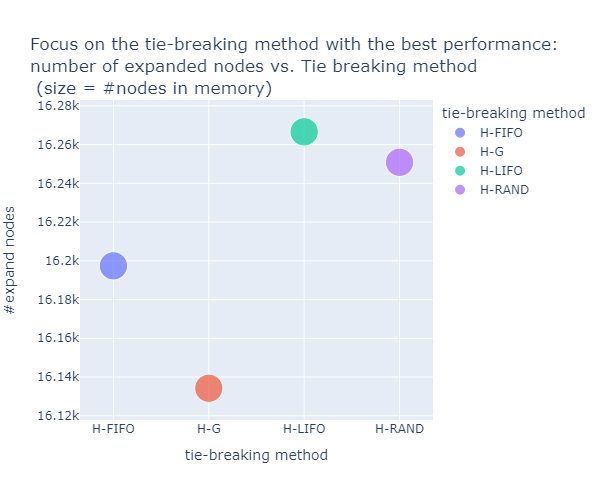
ניתן לראות כי אסטרטגיות לשבירת שוויון שהתבססו על ערכי h הניבו ברוב המקרים ביצועים טובים יותר, הן מבחינת זמן הריצה והן מבחינת סיבוכיות הזיכרון. נשים לב כי מקבלים מגמה אחרת, הנראית קצת אקראית, עבור בעיית המבוך כאשר משתמשים במרחק אוקלידי כפונקציה היוריסטית.

## בעיית הפאזל:

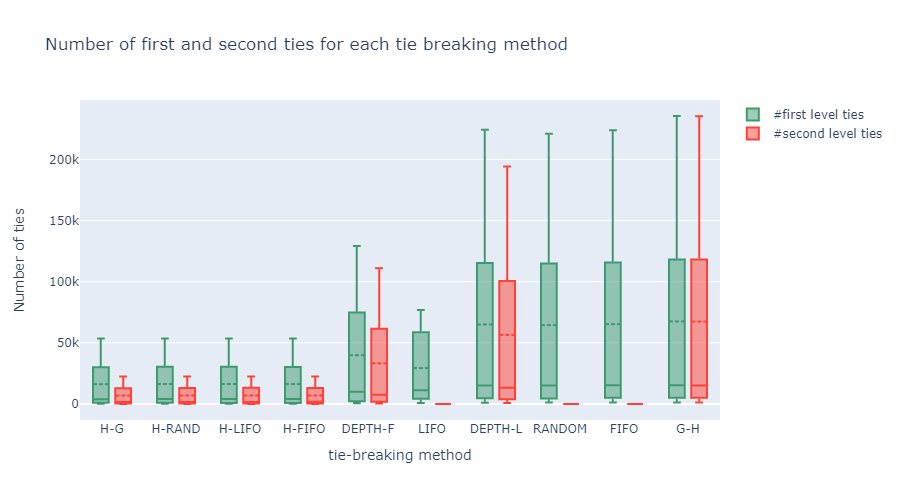
נסתכל בתרשים להלן, המציג עבור כל שיטה לשבירת שוויון את מספר המצבים שהורחבו (ציר y) ואת כמות המצבים שנשמרו בזיכרון (גודל הנקודה) בממוצע עבור כל הניסויים של הפאזל.



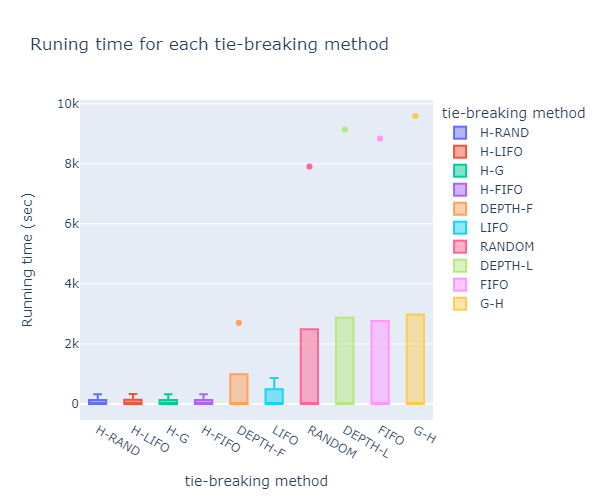
שוב ניתן לראות כי שיטות שהתבססו על ערכי הפונקציה ההיוריסטית לשבירת השוויון בשלב הראשון, השיגו ביצועים טובים יותר: מספר המצבים שהורחבו עבור שיטות אלו היו סביב ה-16.2K, לעומת 67.5K עבור השיטה הגרועה ביותר (G-H). נתמקד על השיטות האלו (H-FIFO, H-G, H-LIFO, H-RAND):



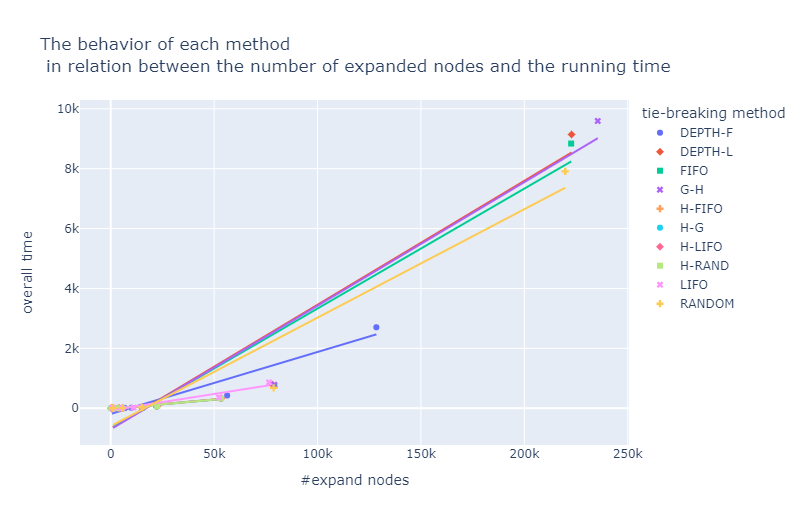
נראה כי על אף ששיטות אלו השיגו תוצאות דומות וטובות, עדיין קיים שוני במספר המצבים שפותחו בין האסטרטגיות השונות. השיטה לשבירת שוויון ברמה 2 השפיעה וחלחלה על המשך הפתרון ובכך על מספר המצבים שהורחבו. ההפרשים במספרים הם נמוכים, ולכן קשה להסיק מכך ראיה חד משמעית.

כדי להסיק על שכיחות התופעה שמצריכה שבירת שוויון, בדקנו כמה פעמים התרחש שוויון בין ערכי f של המצבים בעלי הערך הנמוך ביותר בתור ה-open list. בנוסף, עבור האסטרטגיות שטיפלו גם בשבירת שוויון ברמה 2, הסתכלנו על מספר המופעים של שוויון כזה:

ניתן לראות הבדלים משמעותיים בין השיטות: בעוד השיטות השוברות שוויון לפי ערך h הביאו לכ-16.2K שוויונות, השיטות האחרות הביאו למספר גבוה של שוויונות: 30k-67K בממוצע. גם מספר השוויונות ברמה 2 גדל בהתאם לשוויונות ברמה 1 ושיטות עם ביצועים פחות טובים הביאו לצורך להכריע את השוויון השני בתדירות גבוהה יותר.

נסתכל על זמן הריצה הממוצע, העליון והתחתון עבור כל שיטה:

כפי שראינו בטבלה 1, הפערים הם גדולים בין השיטות: האסטרטגיות הטובות הביאו לזמן ריצה ממוצע של 2.3K שניות, לעומת 30K שניות עבור השיטות האחרות.

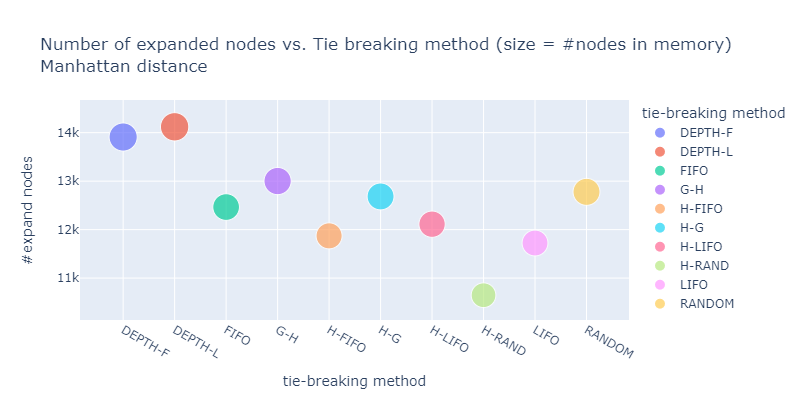
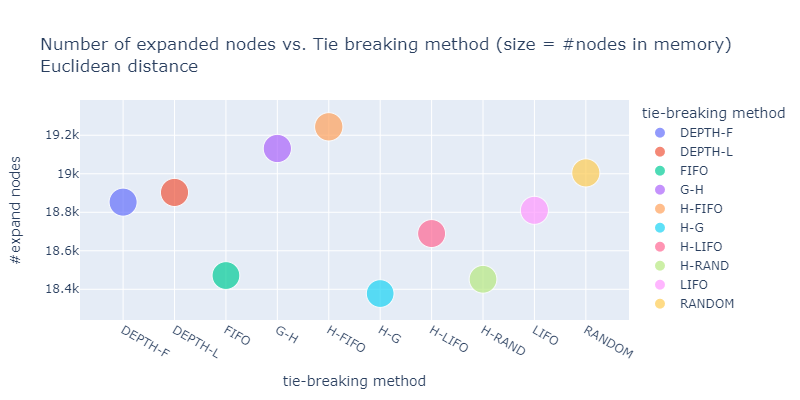
לבסוף, ניתן להסתכל על התנהגות כל שיטה: כיצד פיתוח מספר רב יותר של מצבים הביא לזמן ריצה גדול יותר:

אסטרטגיות טובות הביעו לזמן ריצה יחסית קבוע עבור מספר שונה של מצבים שהורחבו (שיפוע שואף ל-0). השיפוע נהיה לינארי ככל שהשיטה פחות טובה, וזמן הריצה גדל ככל שמספר המצבים שמורחבים גדל.

## בעיית המבוך

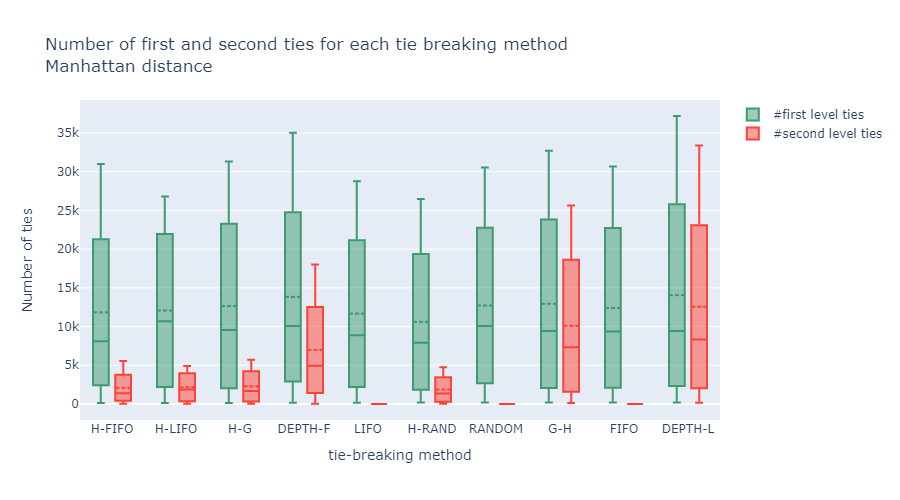
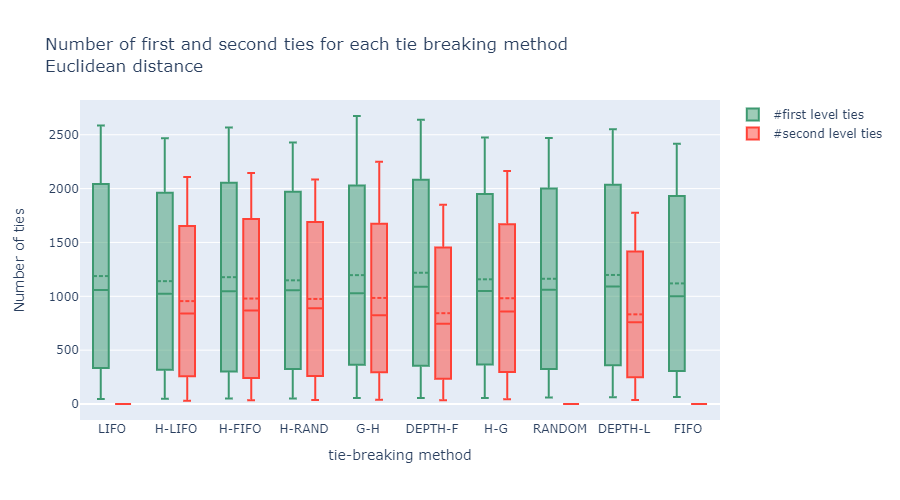
בבעיה זו רצינו לבחון גם את ההבדל בין היוריסטיקה שנחשבת יחסית טובה לבעיה זו (מרחק מנהטן) להיוריסטיקה פחות טובה (מרחק אוקלידי).

Cנסתכל על מספר המצבים שהורחבו בכל שיטה (ציר y) ומספר המצבים שנשמרו בזיכרון במהלך החיפוש אחר הפתרון (גודל הנקודה), בהשוואה בין שני ההיוריסטיקות:



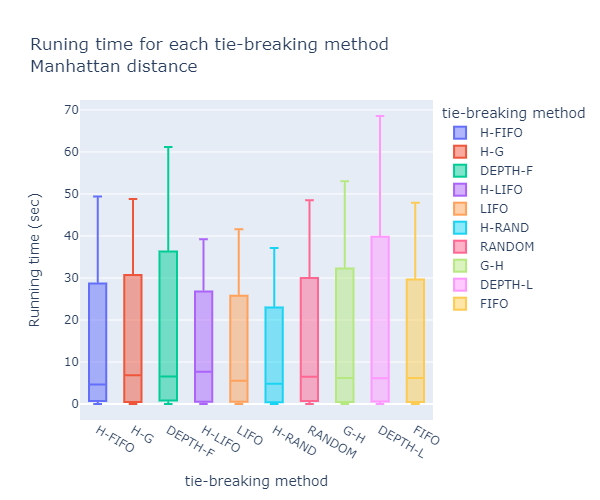
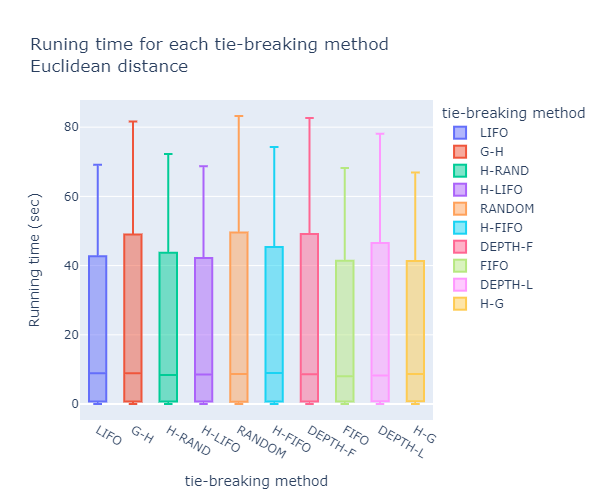
ניתן לראות שסדרי הגודל של ציר ה-y שונים בין שני המקרים, ומרחק מנהטן מביא לפיתוח מספר קטן יותר של מצבים. להוסיף

נשווה גם בין מספר השוויונות (ברמה 1 ו-2) שיש בכל שיטה, בהשוואה בין שני ההיוריסטיקות:



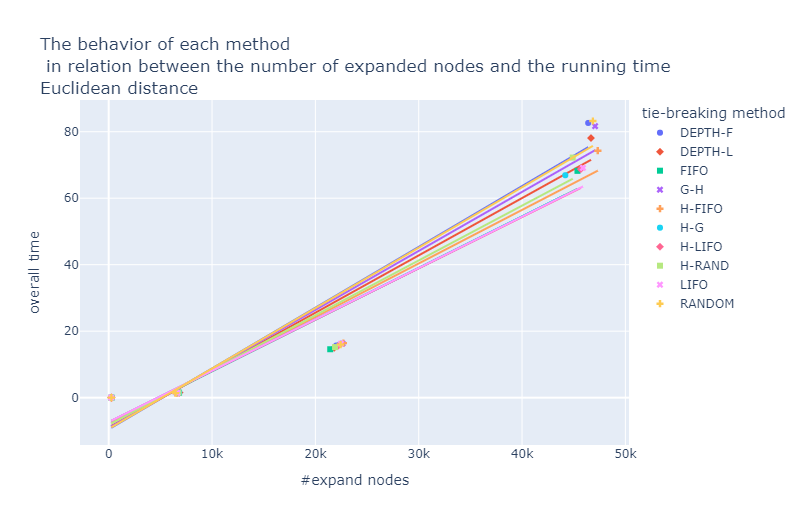
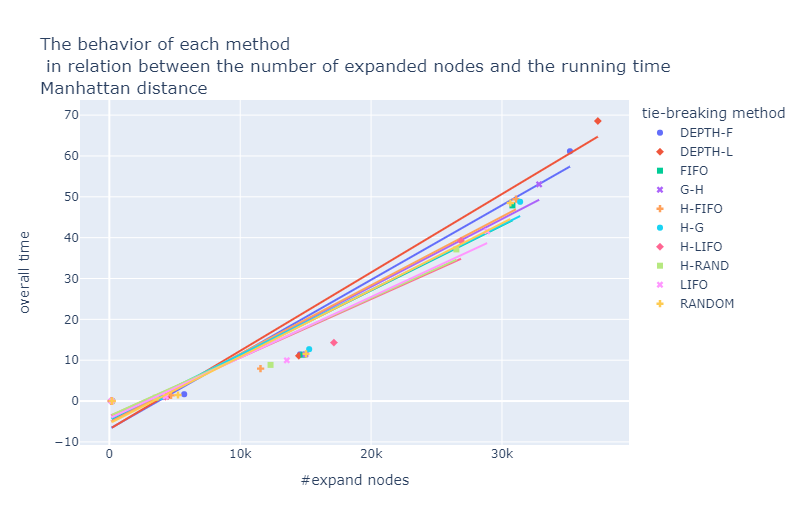
להסביר איך יכול להיות שיש מספרים כל כך שונים בציר הY

זמן הריצה היה ארוך יותר עבור שיטה לשבירת שוויון כאשר השתמשנו במרחק אוקלידי כפונקציה היוריסטית, לעומת מרחק מנהטן.



ההבדלים בזמני ריצה בין אסטרטגיות שהתבססו על ערכי h לשבירת שוויון לעומת שיטות אחרות היו גדולים יותר עבור מרחק מנהטן, לעומת מרחק אוקלידי ששם קשה לראות מגמה ברורה לטובת שיטות אלו.

גם מבחינת השיפוע של היחס בין זמן הריצה לכמות המצבים שפותחו, נראה כי הפונקציה ההיוריסטית של מרחק מנהטן הביא לשיפוע מתון יותר לעומת מרחק אוקלידי:

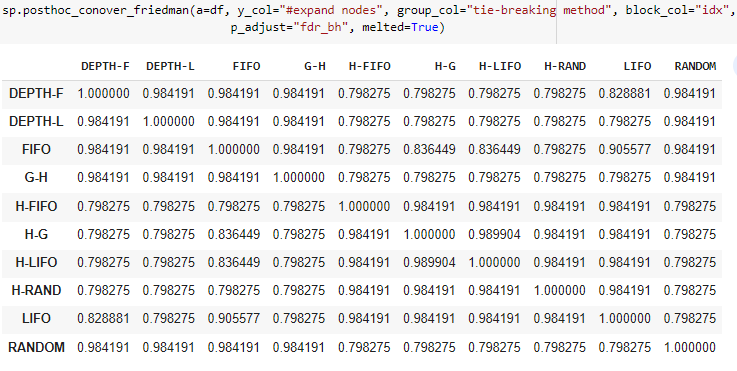


# מסקנות

**מסקנות הניסוי: האם ההשערה אוששה, מדוע? האם היו תוצאות מפתיעות וכו...**

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטיביצענו מבחן פרידמן על מספר המצבים שפותחו לפי כל שיטה, בכל ניסוי. כל ניסוי מוגדר לפי סוג הבעיה, גודל הבעיה (פאזל שונה או מבוך בגודל שונה) ומספר האיטרציה.

קיבלנו כי יש הבדל מובהק סטטיסטי בין האסטרטגיות השונות במספר המצבים שכל שיטה מפתחת (p.v=0.000039). לכן עברנו לשלב הבא (post-hoc test) לבדוק את היחסים בין השיטות השונות:

לא קיבלנו הבדלים מובהקים, אבל ניתן לראות שיש הבדלים גדולים יותר (ערך נמוך יותר) בין שיטות המתבססות על ערכי h לעומת כאלו שלא, והבדלים קטנים יותר (ערכים גדולים בטבלה) בין שיטות המתבססות על ערכי h בשלב הראשון.

# נספחים

קישור לקוד - <https://github.com/lielserf/Search-in-Artificial-Intelligence---project.git>

ניתוח הנתונים וגרפים - <https://colab.research.google.com/drive/1SowUvwFVvb4_UQpcLaKDuioPTMKYC-eU?usp=sharing>

טבלאות הנתונים - <https://drive.google.com/drive/folders/1NRaUfp85dymESLUCc5N_F9NY0Hfc1DzE?usp=share_link>