**ממ"ן 12 חישוביות ביולוגית**

1. האלגוריתם שמוסבר כעת הוא אלגוריתם גנרי למציאת מתושלחים במשחק החיים של קונווי.

הקוד מחולק לשלוש מחלקות:

1. ChromosomeVisualiser – מטרתה להציג ויזואלית את פתרון מסוים. המחלקה יודעת להמיר את הייצוג הבינארי של הכרומוזום לתצוגה ויזואלית של tkinter. כך ניתן לראות את התקדמות הדורות ואת הפתרונות הכי טובים.
2. GameOfLifeSimulator – מטרתה לסמלץ ריצה של משחק החיים על כרומוזום מסוים. במחלקה זו משתמש האלגוריתם הגרני לחשב כשירות. ככל שהכרומוזום (כקונפיגורציה התחלתית של משחק החיים) מגיע ליותר דורות עד שהוא נעצר או נכנס ללופ הכשירות גבוהה יותר.
3. MetushelahFinder – מבצע האלגוריתם הגנרי אותו אתאר כעת.
4. האלגוריתם משתמש באלמנטים האבולוציוניים הבאים:
   1. ייצוג הכרומוזום: בחרתי לייצג את הקונפיגורציה של משחק החיים כרצף ביטים של GRID\_SIZE x GRID\_SIZE (כאשר GRID\_SIZE הוא גודל המטריצה של משחק החיים – כרגע מוגדר כ50). כל תא חי מיוצג ע"י 1 ותא מת ע"י 0. למשל אם תא במיקום (10,20) חי אז ברצף הבינארי הביט במיקום (10\*50+20=520) יהיה 1.
   2. תקינות הכרומוזום: בכל שלב בחיים של כרומוזום (יצירתו, הוספת מוטציות ושחלוף עם כרומוזום אחר) נרצה לוודא שכמות התאים החיים בתחילת הסימלוץ של משחק החיים לא תעלה מעל כמות מסוימת (כרגע מוגדרת 30 תאים אך ניתן לשנות ע"י MAX\_INITIAL\_CONFIGURATION\_SIZE), זאת על מנת שלא נגיע למצב שבעזרת הרבה מוטציות ושחלופים נקבל מתושלח שכמות התאים ההתחלתיים בו היא כל הרשת ובאופן כללי על מנת לעמוד בתנאי שמתושלח הוא קונפיגורציה התחלתית קטנה. תקינות זו נאכפת באופנים שונים בכל שלב באלגוריתם.
   3. מוטציה: כיאה לייצוג של רצף בינארי, מוטציה מוגדרת כשינוי אפס לאחד או אחד לאפס. עבור כל תא נבדר הסיכוי למוטציה ובנוסף נבדקת כמות התאים החיים כרגע בכרומוזום ואם הגענו למקסימום שהוגדר, לא נדליק תא נוסף. את סדר הביטים שנעשה להם מוטציה נגריל על מנת שלא ייצא מצב שכל המוטציות ייקרו רק בהתחלה.
   4. שחלוף: שוב, כיאה לייצוג של רצץ בינארי, נגדיר שחלוף כשילוב של שני הורים ע"י הגרלת נקודת החיתוך ושילוב של חלק א' של הורה א' עם חלק ב' של הורה ב' ושל חלק ב' של הורה א' עם חלק א' של הורה ב' ובכך ליצור שני צאצאים חדשים באוכלוסייה. ההורים נבחרו בשיטת הרולטה ובהסתברות מסוימת (כרגע מוגדרת 90 אחוז אך ניתן לשנות ע"י CROSSOVER\_CHANCE) ההורים משתחלפים. אחרת, הם ממשיכים איך שהם לדור הבא של האלגוריתם. בדיקות תקינות הכרומוזומים החדשים היא ספירת התאים הדולקים וביצוע עוד ועוד שחלופים עד שיתקבלו 2 כרומוזומים חדשים תקינים.
   5. אוכלוסייה: בחרתי לעבוד עם אוכלוסייה של 100 כרומוזומים, כך שבכל דור מתבצע מעבר של האליטה מהדור הקודם לדור הבא (10 אחוז עוברים – אליטיזם) ושאר התשעים אחוזים מתמלאים ע"י שחלופים או מעברים של הדור הנוכחי לדור הבא.
   6. פונקציית כשירות: פונקציית הכשירות לאלגוריתם זה הוא חישוב של משחק החיים ע"י שימוש במחלקה GameOfLifeSimulator. התנאי עצירה שבחרתי הוא חזרה של קונפיגורציה על עצמה במהלך הדורות של סימלוץ משחר החיים. ע"פ ההגדרה, עוצמת המתושלח נקבעת ע"י כמות הדורות שלוקח לו להתייצב. הכוונה בהתייצבות זו לתפישתי היא שכל קבוצה תאים נמצאת בלולאה שחוזרת על עצמה או בחוסר שינוי בין הדורות. לכן ניתן לבדוק אם קונפיגורציה חוזרת על עצמה וברגע שכן להכריע שהכרומוזום סיים ולהחזיר את מספר הדורות ככשירות הכרומוזום.

הדרך שבה בחרתי לייצג חזרתיות של קונפיגורציה במשחק החיים היא ע"י רשימת hash שמחזיקה את כל הדורות הקודמים של הכרומוזום הנוכחי וברגע שנמצאת קונפיגורציה חדשה שהhash שלה כבר קיים ברשימה החישוב עוצר.

* 1. תנאי עצירה: כיוון שאין ידע לגבי המתושלח האולטימטיבי שיכול להמשיך לנצח, קבעתי שאם האלגוריתם קבוע (לא מוצא כשירות טובה יותר) במשך 500 דורות, הוא יבין שנתקע ויעצור את ריצתו.

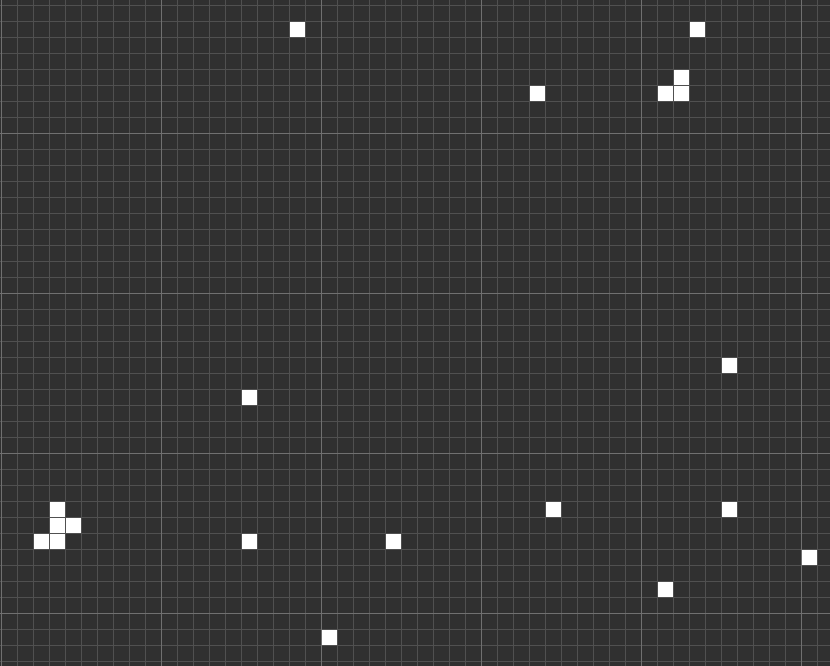
1. נציג מספר מתושלחים שנמצאו ע"י האלגוריתם, תצוגה של ריצתם מוגשת כסרטוני סימולציה של תכנת Golly בלינק הבא:

<https://drive.google.com/drive/folders/1khJgP0N8jioujAmNQ4ZGjhhSzeYD2Wof?usp=share_link>

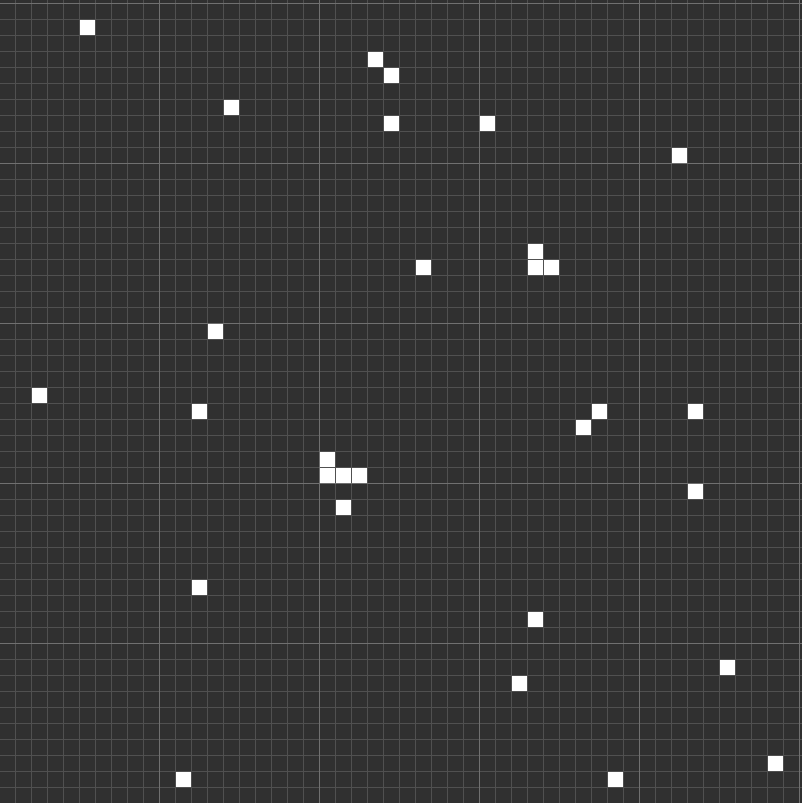
* 1. A picture containing indoor, toilet, tiled

     Description automatically generated1343 דורות

* 1. 1172 דורות



* 1. 978 דורות



1. לפנינו גרף המתאר את הכשירות הממוצעת והכשירות המקסימלית בכל דור כתלות ברצף הדורות.

גרף הכשירות המקסימלית קופץ במדרגות כיוון שלוקח לאלגוריתם זמן למצוא מועמד יותר טוב כל פעם. כלומר ניכר מהגרף שתהליך אבולוציוני הוא לא שיפור לינארי כיוון שלוקח לאלגוריתם הרבה ניסיונות רנדומליים של מוטציות ושחלופים על מנת למצוא פתרון בעל כשירות גבוהה יותר משהייתה עד כה.

עוד הבחנה מעניינת היא שגרף הכשירויות הממוצעות אמנם במגמת עלייה אך מבצע הרבה תנודות סביב גרף המגמה. ניתן להסביר זאת כיוון שבכל דור עלולים להיעלם מהאוכלוסייה פתרונות בעלי כשירות גבוהה עקב שחלופים או מוטציות ויחליפו אותם פתרונות בעלי כשירות נמוכה יותר. אך ככל שנמצאים פתרונות בעלי כשירות גבוהה יותר בסופו של דבר (כיוון שאנחנו פועלים עם שיטה הסתברותית לבחירת הורים – רולטה) ייבחרו הורים שיניבו פתרונות חדשים שיכולים להקפיץ את ממוצע הכשירויות חזרה אל מעבר לקו המגמה.

**הסבר על הקוד:**

<https://drive.google.com/file/d/1ks7X_1cIh-ojDFYd6392jEDBupD8d_1R/view?usp=share_link>

על מנת להריץ את הקוד נדרשות הספריות הבאות (ניתן להתקין בעזרת pip אם לא קיימות):

* random
* tkinter
* bitarray
* numpy
* uuid
* re