

דו"ח הגשה – ביולוגיה חישובית מטלה 2 :

מגישים :

אלעד ויצנבליט, ת"ז – 315944611

אלינוי עמר, ת"ז – 318532132

הרצת התוכנית :

- בתחילת הרצת התוכנית יש לבחור את האלגוריתם אותו רוצים להריץ ע"י (יש להכניס רק מספר) :
- 1 – האלגוריתם הרגיל.
 - 2- האלגוריתם עם האבולוציה הדארווינית.
 - 3- האלגוריתם עם האבולוציה הלאמארקית.

בסוף ריצת התוכנית, מודפס למסך כמות הקריאות לפונקציית ה-fitness.

אנחנו עובדים עם אוכלוסייה בגודל 100 שאותחלה באופן רנדומלי, שכל פתרון באוכלוסייה הוא מילון עם 26 מפתחות (המפתח הוא האות שאותה מפענחים והערך הוא האות שאליה נתרגם).

לצורך חישוב הפונקציית fitness השתמשנו בשלושת הקבצים שקיבלנו במסגרת התרגיל, כאשר בקובץ המילון השתמשנו על מנת לחשב את כמות המילים בטקסט שנמצאות גם במילון, ובקבצי השכיחויות השתמשנו על מנת להשוות את השכיחויות הנתונות לנו בפענוח (ככל שהמרחק בין השכיחויות שלנו לשכיחויות הנתונות בקבצים גבוה יותר, כך הפיטנס המתקבל הוא נמוך יותר).

האלגוריתם :

לאחר שיצרנו את האוכלוסייה באופן רנדומלי, אנחנו נעבור עליה לכל היותר כ-150 דורות (MAX_GENERATION).

כתחילה נחשב את הציון fitness של כל איבר באוכלוסייה, ולאחר מכן אנחנו לוקחים את 20% הפתרונות הכי טובים בהתאם לציון פיטנס שקיבלו (ELITE_SELECTION) ומעתיקים אותם לדור הבא.

לאחר מכן, אנחנו מבצעים crossover מתוך כלל האוכלוסייה של הדור הנוכחי עד שאנחנו מגיעים שוב לגודל אוכלוסייה של 100. את ה-crossover אנחנו מבצעים על ידי הגרלה של מיקום רנדומלי במילון ואז לוקחים חלק אחד מההורה הראשון ואת החלק השני מההורה השני. אחרי ביצוע ה-crossover אנחנו מבצעים בדיקה לראות שאין לנו מפתחות עם אותם ערכים (כלומר אין כפילויות של תרגום). במקרה, ויש כפילויות אנחנו מחליפים את אחד מהכפילויות להיות אות חסרה.

את ההורים אנחנו בוחרים באופן הבא :

אנחנו בוחרים את ההורים בשיטה של טורניר, כאשר אנחנו מגרילים 5 מילונים מתוך האוכלוסייה ואז אנחנו לוקחים מתוכם את המילון עם ציון ה-fitness הגבוה ביותר להיות ההורה הראשון. וחוזרים על התהליך בשנית כדי לבחור את ההורה השני.

אחרי ה-crossover אנחנו עוברים על כל האוכלוסייה ומבצעים מוטציה לכל פתרון בהסתברות של MUTATION_RATE. את המוטציה אנחנו מבצעים על ידי החלפת הערכים של 2 מפתחות רנדומליים במילון.

במקרה שנבחר אלגוריתם דארווני או אלגוריתם למארקי, בשלב הזה מתבצע הריצה של האופטימיזציה הלוקלית ועדכון ציוני הפיטנס בהתאם להגדרת כל אלגוריתם.

לאחר מכן, אנחנו בוחרים את הציון הטוב ביותר הדור החדש, ובודקים אם היה שינוי מהציון הטוב ביותר בדור הקודם.

אם היו מספר דורות ללא שינוי (NO_IMPROVEMENT_THRESHOLD) אז אנחנו מניחים שהאלגוריתם התכנס, ועוצרים את הריצה.

באלגוריתם טיפלו בהתכנסות מוקדמת, ע"י בדיקה של הציון פיטנס שהתקבל מהאלגוריתם הגנטי.

במקרה וקיבלנו ציון הנמוך מרף מסויים (FITNESS_THRESHOLD) אנחנו מבצעים ריצה חדשה של האלגוריתם, כך עד שנקבל ציון פיטנס הגבוה מהרף או עד 5 ריצות סה"כ מקסימום.

בסופו של דבר, אנחנו בוחרים את הפתרון עם ציון הפיטנס הגבוה ביותר ומפענחים איתו את הקובץ.

מילון משתנים גלובליים (פרמטרים):

POPULATION_SIZE - גודל האוכלוסייה

MAX_GENERATIONS – כמות דורות מקסימלי

MUTATION_RATE - הסתברות למוטציה

REPLACEMENT_IN_BLOCK - הכמות עבור כל אחד משלושת הבלוקים שדיברנו עליהם באלגוריתם אותה אנו רוצים לשמר לדור הבא

NO_IMPROVEMENT_THRESHOLD - כמה דורות להמשיך ללא שינוי

ELITE_SELECTION - אחוז הפתרונות הכי טובים איתם נרצה להמשיך לדור הבא

OPTIMIZATION_SWAPS_COUNT - כמות ההחלפות שברצוננו לבצע בלמארק ודארווין

DO_DARWIN - האם לבצע Darwin

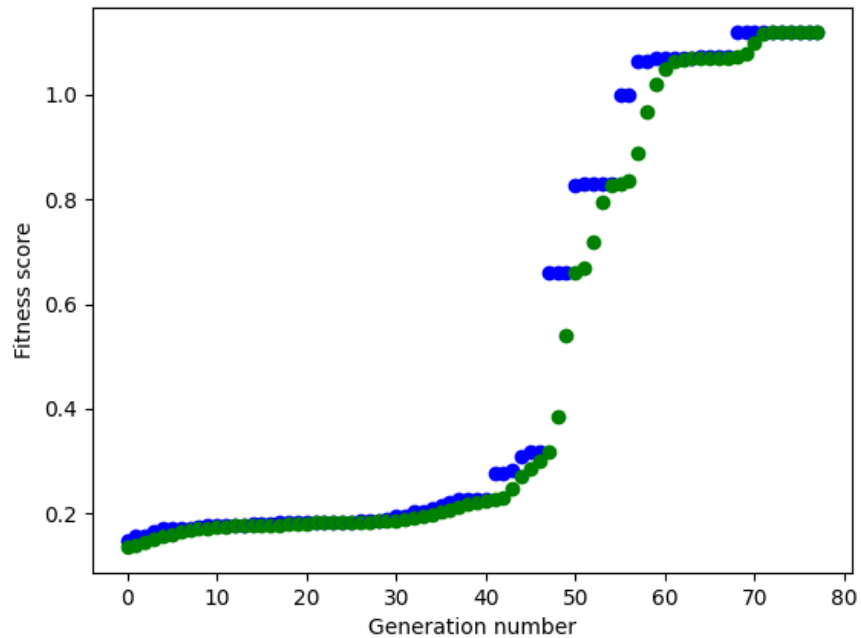
DO_LAMARCK - האם לבצע Lamarck

FITNESS_THRESHOLD - מתחת לזה יחשב כמקסימום מקומי.

הצגת התנהגות האלגוריתמים:

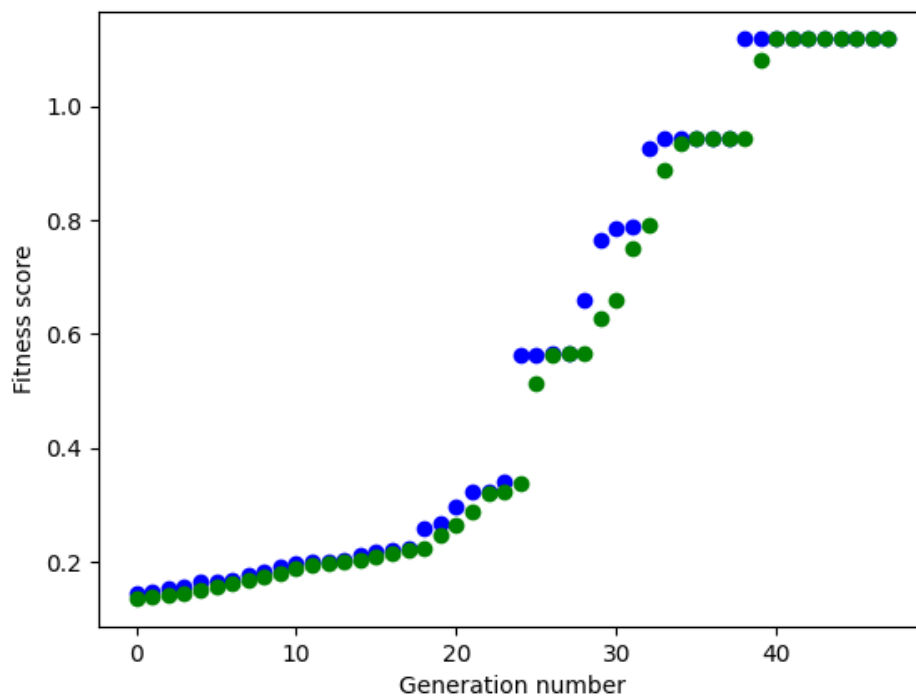
כחול – ציון הפיטנס הגבוה ביותר. ירוק – ציון הפיטנס הממוצע.

האלגוריתם הרגיל:



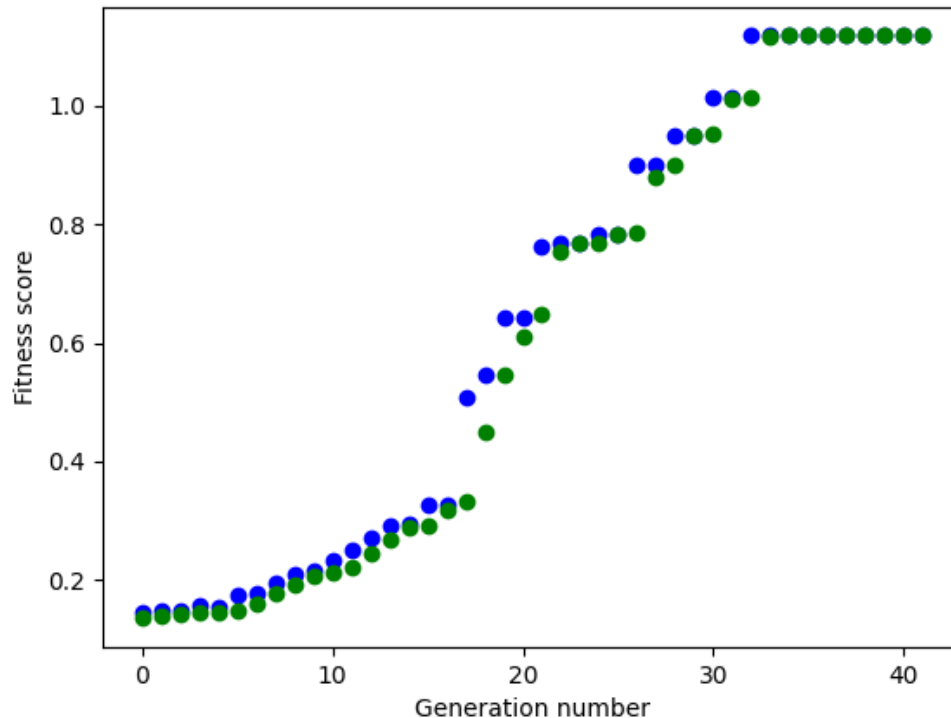
Number of calls to fitness function is: 9700

האלגוריתם הדארוויני:



Number of calls to fitness function is: 24100

האלגוריתם הלמארקי :



Number of calls to fitness function is: 11500

סיכום גרפים :

הרצנו על הטקסט שקיבלנו במטלה, ובכל הריצות של האלגוריתם קיבלנו צופן מפוענח ב-100%. כלומר, כל המפתחות סווגו לערכים הנכונים.

כעת, בגרפים ניתן לראות שהגרף הרגיל התכנס לאחר הכי הרבה זמן (כ-80 דורות), האלגוריתם הדארוויני התכנס לאחר כ-50 דורות, והאלגוריתם הלמארקי היה הכי מהיר להגיע להתכנסות (כ-40 דורות).

בשלושת הגרפים ניתן לראות שההתחלה של שיפור ציון פיטנס מאוד איטית, אך בשלב כלשהו, האוכלוסייה כולה משתפרת ויש קפיצה משמעותית בציון והאלגוריתם מצליח להגיע להתכנסות בציון טוב, תוך זמן קצר יחסית (בגרפים נראה בממוצע כ-15 דורות).

משחק עם פרמטרים:

כל פעם קיבענו את כל הפרמטרים ושיחקנו עם השאר.

ערכי ברירת המחדל:

20% = ELITE_SELECTION ; 100=גודל אוכלוסייה; 1=MUTATION_RATE

- MUTATION_RATE

ערך	מספר דור אחרון
1.0	generation 69: best fitness = 1.1176122804416047
0.8	generation 72: best fitness = 0.9106205455651285
0.4	generation 64: best fitness = 0.18734869873696378
0.2	generation 78: best fitness = 1.0353120448375193

גודל אוכלוסייה -

ערך	מספר דור אחרון
100	generation 69: best fitness = 1.1176122804416047
150	generation 83: best fitness = 1.1176122804416047
50	generation 69: best fitness = 1.1126248624101152
15	generation 38: best fitness = 0.15738351938255263

– ELITE_SELECTION

ערך	מספר דור אחרון
20%	generation 69: best fitness = 1.1176122804416047
5%	generation 68: best fitness = 1.1176122804416047
50%	generation 97: best fitness = 1.1176122804416047
10%	generation 70: best fitness = 1.1176122804416047