ביולוגיה חישובית תרגיל 2

יואב אליאב
312498207

האלגוריתם

האלגוריתם הגנטי שבניתי מורכב מכמה שלבים, להלן השלבים ופירוט המימוש שלהם:

1) אוכלוסיה התחלתית: ראשית בניתי סט פתרונות התחלתי שיוצר בצורה רנדומית, כל פיתרון היה בעצם זוג של פרמוטציות של האלפבית, הייצוג של פיתרון נעשה בעזרת מילון שהתאים לכל אות מוצפנת (key) את האות שהיא מייצגת (value). ניתן לראות זאת כבעצם 2 רשימות Values וKeys שיש בניהן התאמה. דוגמא להמחשת פיתרון:

Solution A:

Keys (צופן) a b c d e f.....

Values(אות אמיתית) b c d e a.....

2) פונקציית הערכה: לאחר בניית האוכלוסיה ההתחלתית, נעשה הערכה בעזרת פונקצייה זו, כדי לדעת את טיב הפיתרון הפונקציה פועלת ב2 מישורים ומשקללת אותם לצורך הציון הסופי:

- התאמת מילים: בדיקה של כמה מילים כל פיתרון מצליח לפתור מתוך סה״כ המילים בטקסט המוצפן. האלגוריתם עובר על כל מילה בטקסט המפוענח של הפיתרון, בודק האם המילה קיימת בקובץ dict.txt כך שלבסוף מתקבל ציון על סמך כמות המילים המתאימות.

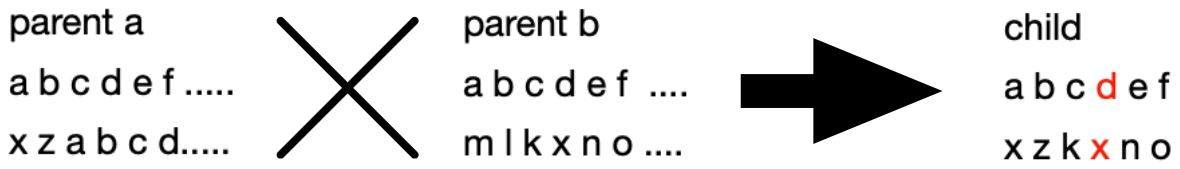
- התפלגות אותיות: בדיקה של כמה קרובה התפלגות האותיות בטקסט להתפלגות שלהן במציאות. ככל שההתפלגות דומה יותר הציון יהיה טוב יותר.

לכל אחת מהגישות יש משקולת w ערך המשקולת הינו היפר פרמטר, שניתן לשחק איתו, ההיגיון הוא שהתפלגות האותיות אמור להיות פחות משמעותי מהתאמת מילים לכן יהיה לו משקל קטן יותר בציון.

3) זיווג: לאחר שנתנו ציון לכל פיתרון, יש צורך לבחור את ההורים ולזווג אותם, פעולת הזיווג נעשתה בשני שלבים:

א) בחירת ההורים - על מנת לבחור את ההורים בחרתי סט רנדומי של פתרונות מתוך כלל הפתרונות באוכלוסיה, לאחר בניית הסט, בחרתי את הפיתרון עם הציון הכי גבוה, והוא היה הורה של הדור הבא. הדבר שומר על רנדומיות אך מגביר את הסיכויים של פיתרון איכותי להיבחר. נשים לב שפונקציית הזיווג מופעלת כל פעם כדי למצוא הורה, ולכן לדוגמא כאשר אנחנו רוצים ליצור 100 צאצאים אנחנו נצטרך להפעיל את פונקציית הזיווג כ200 פעם שכן יש צורך למצוא 100 זוגות של הורים.

ב) יצירת צאצא - כדי ליצור צאצא בחרתי בצורה רנדומית נקודת חתך n, כך שn הזוגות הראשונים של ההורה הראשון יועברו לצאצא, וכל השאר יועברו מההורה השני, נקודה רגישה בגישה זאת היא שייתכן ויווצרו לנו פתרונות לא חוקיים, שכן ייתכן ואחד המפתחות של ההורה השני מותאם לאות שכבר ישנו מפתח מהורה א׳ שמותאם אליה. לדוגמא נסתכל על המצב הבא כאשר n = 2:



כפי שניתן לראות במצב הזה 2 האיברים הראשונים מגיעים מהורה א׳, ושאר הגנים מהורה ב׳, אך עבור המפתח d יש לנו את הערך x שגם המפתח a מופנה אליו. כדי לתקן את הצאצא כאשר ראיתי שמפתח מסויים הערך שלו כבר קיים במילון, הכנסתי את המפתח עם הערך '|' מתוך הבנה שאנחנו מתאימים 26 אותיות ל26 אותיות, לכן לאחר סיום היצירה של הצאצא, עם יש לו מפתחות שהערך שלהם הוא '|' אזי אני יודע בוודאות שישנן אותיות שלא מופיעות ברשימת הvalues, במצב כזה הכנסתי את האותיות האלו לכל מפתח שהערך שלו הוא '|' וככה שימרתי את החוקיות של הפיתרון.

4) מוטציה - פעולת המוטציה נעשתה גם היא בהסתברות מסויימת, הפעולה הינה פשוט להחליף בין זוג אותיות (מפתחות) והערכים שלהן, הבחירה של הזוג אותיות שיוחלפו הינה רנדומית.

5) **התפתחות האוכלוסיה (Evolve)** - השלב הזה בעצם היווה איחוד של שאר השלבים, אנחנו יוצרים אוכלוסיה חדשה בעזרת הזיווג, ונותנים להם הערכה חדשה אם יש צורך ובהתאם לקלט של המשתמש, אנחנו נפעיל את האופטימיזציה הדארוויניסטית או הלמארקית.

<u>בעיית ההתכנסות</u>

אחד הבעיות של האלגוריתם הינה בעיית ההתכנסות, אחת הסיבות לכך היא בעצם שהשונות "הגנטית" באוכלוסיה יורדת קרי הדורות נהיים דומים יותר ויותר ולכן יכולת ההשתפרות שלהם פוחתת בצורה משמעותית. לצורך טיפול בבעיה השתמשתי בשילוב של 2 שיטות:

- אליטיזם: הרעיון הוא לשמור את הפיתרון הטוב ביותר בכל הדורות, הדבר הזה גם מבטיח לנו שאכן הפיתרון הסופי יהיה הפיתרון הטוב ביותר שמצאנו, אך גם נרצה להימנע ממצב שהפיתרון הטוב ביותר לא נבחר או יבלע במהלך הדורות, וגנים פחות טובים ישתלטו על האוכלוסיה וכך נגיע למצב של אופטימום לוקאלי.
- גודל דינאמי: כפי שנאמר אחת הסיבות להתכנסות מוקדמת היא שונות נמוכה באוכלוסיה, לכן כאשר אין לנו שיפור במהלך הדורות והאלגוריתם נתקע על אופטימום לוקאלי, אנחנו נגדיל את האוכלוסיה (נכפיל את גודלה) ההגדלה של האוכלוסיה אמורה להגדיל את השונות ולאפשר לנו לצאת מהאופטימום הלוקאלי, אם אכן יצאנו מהאופטימום, נחזיר את גודל האוכלוסיה להיות הגודל הראשוני.

<u>ביצועים</u>

ההשוואות נעשו בין שלושת האלגוריתמים הבאים:

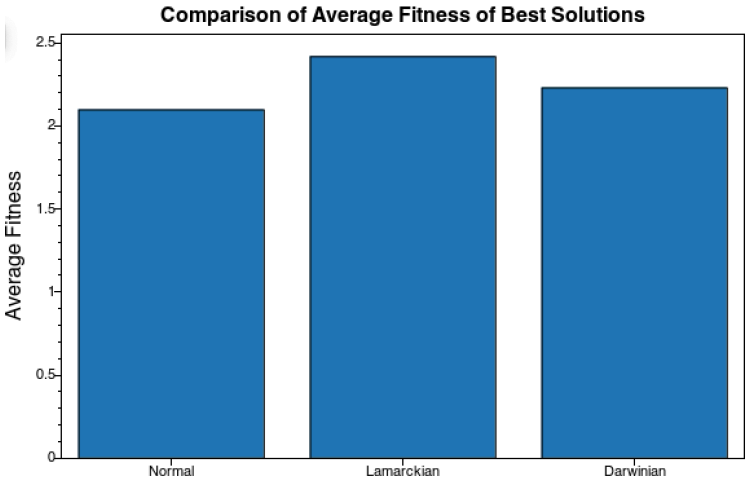
- אלגוריתם גנטי: הגרסא הקלאסית שתוארה.

- אלגוריתן גנטי דארווניסטי: בגרסה זו בנוסף לאלגוריתם הגנטי, בסיום יצירת אוכלוסיה חדשה, על כל פרט באוכלוסיה החדשה מתבצע אופטימציה לוקאלית, קרי עושים n החלפות בין זוגות ובודקים אם בוצע שיפור, אם ישנו שיפור מעדכנים את הציון של המועמד אך לא משנים אותו.

- אלגוריתם גנטי לאמרקי: גרסה זו פועלת בדומה לאלגוריתם הדארווינסטי רק שכעת אם ישנו שיפור אנחנו נעדכן את הפרט.

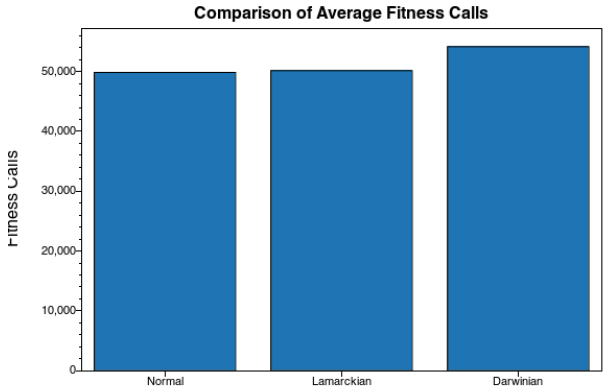
טיב הפיתרון

הבדיקה הראשונה שנבדקה הינה איכות הפיתרון, מהתוצאות נראה שבאופן כללי השיטות הצליחו להגיע לאחוזי הצלחה גבוהים וברוב הפעמים הצליחו לפצח את הצופן, אך שימוש בגרסה הלאמרקית שיפר בצורה יחסית גבוהה את טיב הפיתרון, הסיבה לכך לכל הנראה היא שביצוע ה"שדרוג הגנטי" של האוכלוסיה בצורה מכוונת מבטיח צאצאים טובים יותר.



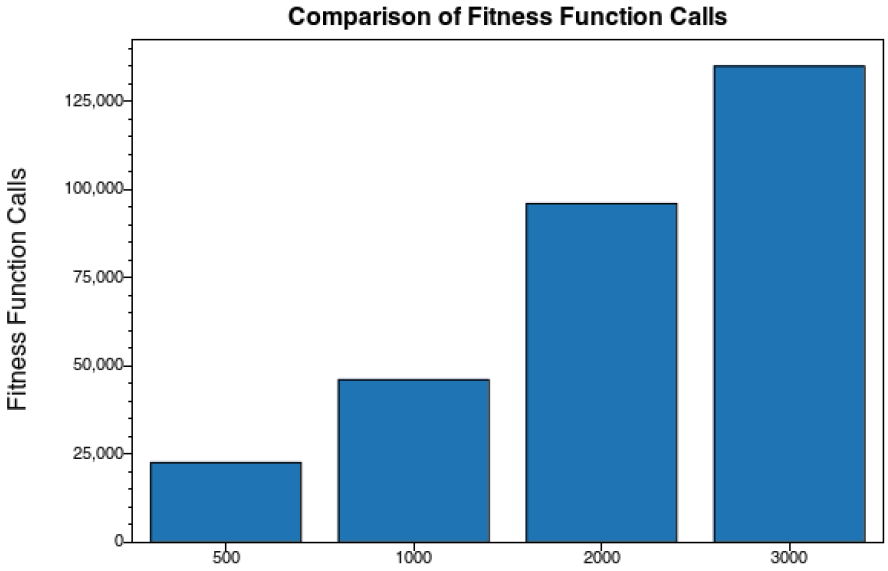
ממוצע שימוש בפונקציית הערכה

הבדיקה השנייה שנעשתה היו מספר הקריאות לפונקציית ההערכה, מהתוצאות נראה שהגרסה הדארווינסטית גוררת יותר קריאות לפונקציות הערכה.



היפר פרמטרים - גודל האוכלוסיה

הפרמטר שבחרתי לשים בו דגש הינו גודל האוכלוסיה, הפרמטר הזה חשוב כי מצד אחד אוכלוסיה גדולה תיתן לנו בהסתברות גדולה יותר שונות גדולה יותר ותמנע מאיתנו להגיע לאופטימום לוקאלי, מנגד בחירת של אוכלוסיה גדולה יגרום להכבדת האלגוריתם בצורה משמעותית, להלן תוצאת הניתוח של מהירות האלגוריתם כאשר האוכלוסיות בגדלים שונים:



כפי שניתן לראות הפערים מאוד משמעותיים, וכבר עבור אוכלוסיה גדולה מ1000 האלגוריתם נהיה מאוד כבד, מנגד כפי שניתן לראות בגרף למטה, הביצועים עבור אוכלוסיה גדולה היו מאוד טובים, לכן עבור ההשוואות מצאתי שאוכלוסייה התחלתית של 1000 יתן את השקלול תמורות הטוב ביותר, שגם יצליח לתת תוצאות טובות אך מצד שני לא יכביד על הזמן ריצה בצורה קיצונית.