**דוח 2 ביולוגיה חישובית – אלגוריתמים גנטיים**  
נתנאל לנדסמן, גל לוי

**חלק א**

**סעיף 1**

באלגוריתם גנטי זה, כל פתרון פוטנציאלי לבעיית הפענוח הוא תמורה של האלפבית האנגלי (מסומן ב-array alpha\_set). כל פתרון מיוצג כמבנה אינדיבידואלי עם שני שדות:

1. Sequence: זהו מערך של 26 תווים המייצגים תמורה ספציפית של האלפבית האנגלי. כל אינדקס של המערך ממפה לאות באלפבית הסטנדרטי בסדר אלפביתי. לדוגמה, אם התו באינדקס ה-0 של הרצף הוא 'm', זה אומר ש-'a' (האות ה-0 באלפבית) ממופה ל-'m' עבור הsequence המסוים הזה. מעתה והלאה לצורך הנוחות וההבנה בתיאור דוח זה נקרא לו מפתח פיענוח או decipher key.
2. Fitness: זהו מדד לאופן שבו הרצף מפענח את הטקסט המוצפן. הfitness מחושב באמצעות פונקציית כשירות, המעניקה ציונים גבוהים יותר לרצפים המייצרים טקסט אנגלי קוהרנטי והגיוני יותר

אז בעצם, כל מרחב הפתרונות הוא קבוצת כל התמורות האפשריות של האלפבית האנגלי, וכל פרט מייצג נקודה אחת בתוך המרחב הזה.

האלגוריתם הגנטי מייצר פתרונות חדשים (individuals חדשים) על ידי ביצוע פעולות כמו crossover וmutation באוכלוסיית הפתרונות הקיימת, ומעריך אותם באמצעות פונקציית fitness. בכל דור היא בוחרת את הפתרונות הטובים ביותר על סמך ציוני הfitness שלהם, והפתרונות הנבחרים הללו מרכיבים את האוכלוסייה לדור הבא.

בסוף כל הדורות, הרצף של הפרט הטוב ביותר (בעל הכושר הגבוה ביותר) נחשב לפתרון הסופי. לאחר מכן זה נשמר בקובץ טקסט שבו כל שורה מכילה מיפוי מתו אחד באלפבית הסטנדרטי לתו ברצף. הרצף השמור יכול לשמש כ decipher key לפענוח הטקסט המוצפן.

**סעיף 2**

פונקציית הfitness של אלגוריתם גנטי זה היא פונקציית ניקוד מותאמת אישית המיועדת לבעיית הפענוח. הפונקציה נועדה לדרג עד כמה טוב פתרון פוטנציאלי (מפתח פענוח המיוצג כרצף של תווים) בפענוח טקסט נתון. הנה תיאור מפורט יותר:

פונקציית הכושר fitness(code) לוקחת כארגומנט קוד, שהוא רצף המייצג decipher key. לאחר מכן, הפונקציה משתמשת במפתח הפענוח הזה כדי לתרגם טקסט מוצפן לטקסט רגיל פוטנציאלי.

לאחר מכן הפונקציה מחשבת ציון עבור טקסט רגיל זה על סמך מספר קריטריונים:

1. ציון שכיחות תווים: עבור כל תו בטקסט הפשוט, הפונקציה מוסיפה את השכיחות של אותו תו בשפה האנגלית (לפי טבלת שכיחויות מוגדרת מראש G.FREQ) לניקוד. זה מבוסס על הרעיון שאותיות נפוצות יותר באנגלית (כמו 'e' או 'a') צריכות להופיע בתדירות גבוהה יותר מאשר פחות נפוצות (כמו 'z' או 'q'). אם fc הוא השכיחות של התו c באנגלית ו-nc הוא מספר הפעמים ש-c מופיע בטקסט הפשוט, אזי ציון שכיחות התווים ניתן על ידי:
2. ציון מילים: הפונקציה מפצלת את הטקסט הפשוט למילים ובודקת כל מילה מול מילון מוגדר מראש G.DICTIONARY. אם המילה נמצאת במילון, היא מוסיפה 1 לניקוד. זה מעודד את האלגוריתם למצוא decipher keys שמייצרים מילים באנגלית אמיתיות. אם wi היא המילה ה-i בטקסט הפשוט ו-D(w) היא פונקציה שמחזירה 1 אם w נמצא במילון ו-0 אחרת, אז המילה ציון ניתנת על ידי:
3. ציון שכיחות זוג תווים: הפונקציה מחשיבה כל זוג תווים עוקב בטקסט הפשוט. עבור כל זוג, הוא מוסיף את השכיחות של אותו זוג באנגלית (לפי טבלת תדרים מוגדרת מראש G.FREQ2) כפול 10 לציון. זה מעודד את האלגוריתם למצוא decipher keys המייצרים זוגות משותפים של אותיות. אם fp הוא השכיחות של זוג התווים p באנגלית ו-np הוא מספר הפעמים ש-p מופיע בטקסט הפשוט, אז ציון השכיחות של זוג התווים ניתן על ידי:

ההכפלה ב10 מתבצעת על מנת לתקנן את גודל ערכי ההסתברות המופיעים בG.FREQ2 שכאמור קטנים מאוד.

הציון הכולל של decipher key הוא הסכום של ציון תדירות התווים, ציון המילה וציון התדירות של זוג התווים. ציון זה מוחזר לאחר מכן כהתאמה של מפתח הפענוח. ככל שהניקוד גבוה יותר, כך מפתח הפענוח נחשב טוב יותר בפענוח הטקסט.

זוהי פונקציית הערכה היוריסטית שתוכננה במיוחד עבור בעיית הפענוח הזו, והיא מבוססת על הנחות מסוימות לגבי טקסט באנגלית. אם הטקסט שמפוענח בפועל אינו תואם להנחות אלו (לדוגמה, אם הוא לא באנגלית), ייתכן שפונקציית כושר זו אינה מתאימה.

**סעיף 3**

ביישום זה, פעולת הcrossover (function crossover(parent\_a, parent\_b)) מתבצעת באמצעות אסטרטגיית הcrossover של שתי נקודות, שהיא טכניקה נפוצה בשימוש באלגוריתמים גנטיים. שיטה זו מייצרת שני פתרונות צאצאים חדשים משני פתרונות הורים על ידי החלפת קטע מהרצפים בין שני ההורים.

1. בחר שתי הורים עבור הcrossover: שני מיקומים (pos1 ו-pos2) נבחרים באקראי בתוך רצף פתרונות האב. שתי נקודות אלו מגדירות קטע או רצף בתוך פתרונות האב שיוחלפו.
2. החלפת תתי-רצפים: רצף המשנה בין pos1 ל-pos2 (כולל) מ-parent\_a מועתק לאותם עמדות ב-offspring\_b ולהיפך.
3. תיקון צאצאים: כתוצאה מההחלפה, אותיות מסוימות עשויות להופיע פעמיים בצאצא בעוד שאחרות לא יופיעו כלל. נוצר מיפוי כדי לזהות את הכפילויות הללו, ולאחר מכן הפונקציה map\_values ​​משמשת להחלפת התווים המשוכפלים בתווים החסרים, ומבטיחה שכל צאצא יישאר תמורה חוקית של האלפבית.

במונחים מתמטיים, נרצה לסמן את רצף האב A בתור ואת רצף האב B בתור לאחר בחירת שתי נקודות הצלבה pos1 ו-pos2, נוצרים רצפי הצאצאים O\_A ו-O\_B באופן הבא:

ניתן לראות כי נוצרו צאצאים שהם תערובות של רצפי האב. עם זאת, רצפי צאצאים אלה עשויים להכיל תווים כפולים ולהחמיץ כמה דמויות אחרות. כדי לתקן זאת, נוצר מיפוי לכל צאצא שממפה את הדמויות הכפולות לאלו החסרות. מיפוי זה משמש לאחר מכן כדי להחליף את הכפילויות בתווים החסרים ברצפי הצאצאים, ומבטיח שכל צאצא הוא תמורה חוקית של האלפבית.

**סעיף 4**

שימוש במוטציות למעשה יבצע שינויים והוספות החסרות ברצף הsequence כך שתתקבל שונות באוכלוסיה שתוביל לExpressivity גבוה יותר. עם זאת, רצפי צאצאים אלה עשויים להכיל תווים כפולים ולהחמיץ כמה תווים אחרים. כדי לתקן זאת, נוצר מיפוי לכל צאצא שממפה את הדמויות הכפולות לאלו החסרות. מיפוי זה משמש לאחר מכן כדי להחליף את הכפילויות בתווים החסרים ברצפי הצאצאים, ומבטיח שכל צאצא הוא תמורה חוקית של האלפבית.

1. יצירת מופעי מוטציה: עבור כל אלמנט ברצף, נוצר מספר אקראי בין 0 ל-1. אם מספר זה קטן משיעור המוטציה mu, האלמנט מסומן לעבור מוטציה.
2. בצע מוטציה: עבור כל אלמנט מסומן, הוא מוחלף באלמנט אחר במיקום אקראי בתוך הרצף.

בכתיב מתמטי נתאר את הרצף האב בתור . לאחר ההליך מוטציוני שביצענו ברצף כך שהתקבל באופן הבא: עבור כל אלמנט Si בS מספר רנדומי כלשהו ri ג'ונרט. לאחר מכן ביצענו: ההחלפה בוצעה עם אלמנט אחר כלשהו Sj. לאחר שכל האלמנטים ב-S עברו תהליך זה, הרצף המתקבל הוא הרצף המוטציה.

**סעיף 5**

התכנסות מוקדמת, המתרחשת כאשר האוכלוסייה באלגוריתם גנטי מתכנסת לפתרון לא אופטימלי לפני שהייתה לה הזדמנות לחקור אזורים אחרים במרחב הפתרונות, היא בעיה שכיחה. ישנם מספר היבטים של האלגוריתם הגנטי המיושם שעוזרים להפחית את הסיכון הזה:

1. פונקציית fitness מגוונת: פונקציית ה fitnessנועדה לתת ציונים רבים ושונים, מה שמעודד מטבעו גיוון באוכלוסיה. היא משתמשת בשכיחות תווים, שכיחות זוגות ובודק אם המילים שנוצרו נמצאות במילון. סוג זה של פונקציית fitness נוטה לייצר מגוון רחב יותר של ציונים מכיוון שמרחב הפתרונות הוא עצום ומורכב (26 גורמים לתמורות האלפבית). מגוון זה של ציונים מקדם גיוון בפתרונות, מה שמפחית את הסיכוי שהאלגוריתם יתקע באופטימיות מקומית.
2. פעולת mutation: אופרטור המוטציה באלגוריתם הגנטי עוזר לשמור על מגוון גנטי בתוך האוכלוסייה על ידי החדרת שינויים אקראיים קטנים בצאצאים. בקוד זה, פעולת הmutation מחליפה באופן אקראי שני תווים ברצף אם מספר אקראי שנוצר נמוך משיעור המוטציה שהוגדר מראש (mu). האקראיות הזו מבטיחה שהחדרת חומר גנטי חדש, מספקת אמצעי לחקור אזורים חדשים במרחב הפתרונות, מה שמסייע בהימנעות מהתכנסות מוקדמת.
3. Rolette wheel selection: אסטרטגיית בחירה זו מעניקה לכל הפרטים באוכלוסיה הזדמנות להיבחר לדור הבא על סמך הכשירות שלהם. לאנשים עם כשירות גבוה יותר יש סיכוי גבוה יותר, אבל גם לבעלי כשירות נמוכה הסיכוי לא אפסי. זה שומר על הגיוון באוכלוסיה על ידי הבטחה שלא רק האנשים המתאימים ביותר ייבחרו לcrossover.
4. סלקציה: בכל דור, לאחר פעולות הcrossover והnutation, האוכלוסייה ממוינת לפי הכשירות, והפרטים המובילים ב-npop נבחרים ליצירת הדור הבא. משמעות הדבר היא שאם יימצא פרט בעל ערך כשירות טוב, הוא לא יאבד בדורות הבאים, מה שמאפשר לפתרונות טובים להימשך בזמן שחקרו חדשים.
5. crossover בשתי נקודות: אסטרטגיה זו יכולה להגדיל את מגוון האוכלוסיות על ידי שילוב של חלקים שונים של שני פתרונות הורים, וגורמת ליצירה צאצאים שונים באופן משמעותי מהוריהם.

התכנסות מוקדמת היא בעיה מורכבת שיכולה להיות תלויה בגורמים רבים, כולל אופי מרחב הפתרונות, הפרמטרים של האלגוריתם הגנטי (כגון גודל האוכלוסייה ושיעור המוטציות) וסיכוי אקראי. לגישה חזקה יותר למניעת התכנסות מוקדמת, ניתן להשתמש בטכניקות מתוחכמות יותר, כגון שימוש במדד גיוון כדי להתאים את קצב המוטציות באופן דינמי, או שילוב שיטות צדדיות לשמירה על תת-אוכלוסיות עם פתרונות שונים. בדוגמא הספציפית שלנו השיטות בהן נקטנו עזרו לנו ולכן לא השתמשנו בשיטות נוספות

**סעיף 6**

אנחנו בחרנו לעצור את הריצה לאחר מספר קבוע של איטרציות. שימוש במספר קבוע של איטרציות, הוא דרך פשוטה לשלוט בזמן הריצה של אלגוריתם גנטי. גישה זו מספקת מספר יתרונות מרכזיים:

1. זמן ריצה צפוי: אלגוריתמים גנטיים יכולים להיות יקרים מבחינה חישובית, במיוחד עם גדלי אוכלוסיה גדולים או פונקציות כושר מורכבות. על ידי הגדרת מספר קבוע של דורות, אנו יכולים להעריך את זמן הריצה הכולל של האלגוריתם על סמך הזמן שלוקח להשלים דור בודד. זה מקל על ניהול משאבים ותזמון חישובים.
2. הימנעות מoverfiting: כמו כל אלגוריתם אופטימיזציה אחר, אלגוריתמים גנטיים רגישים לoverfiting, המתרחש כאשר האלגוריתם מכוון יתר על המידה לנתונים הספציפיים עליהם הוא מאומן, ומפחית את יכולתו להכליל. על ידי הגדרת מספר קבוע של דורות, תוכל לסייע במניעת הoverfiting של האלגוריתם לנתוני האימון.
3. פשטות: מספר דורות קבוע שומר על האלגוריתם פשוט. קריטריוני עצירה אחרים יכולים להוסיף מורכבות לאלגוריתם, כגון הצורך לבדוק אם ערך הכשירות הטוב ביותר לא השתפר במשך מספר מסוים של דורות, או אם מגוון האוכלוסייה נמצא מתחת לסף מסוים.

ישנם קריטריונים רבים אחרים לעצירה שניתן לשקול, כגון זמן ריצה מרבי מותר, ערך כשירות סופית לעצירה, גיוון אוכלוסיות מינימלי או חוסר שיפור משמעותי לאורך מספר מסוים של דורות. אלה יכולים לספק גמישות רבה יותר ועלולים לחסוך משאבי חישוב, אבל הם גם מוסיפים מורכבות לאלגוריתם ויכולים להיות קשים יותר לכוונון.

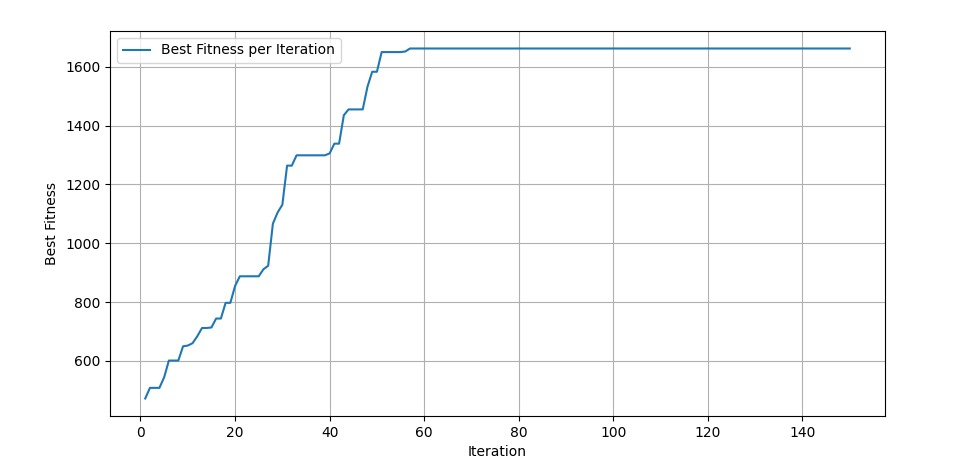
שימוש במספר קבוע של איטרציות אכן יכול להיות מועיל במיוחד כאשר יש לנו מרחב השערות גדול, כפי שקורה כאן. "מרחב היפותזה" בהקשר שלנו מתייחס למכלול של כל הפתרונות האפשריים - כל התמורות הפוטנציאליות של האלפבית. בהינתן 26 התווים באלפבית, יש 26! תמורות ייחודיות. זהו מרחב חיפוש גדול מאוד..

במרחב כה עצום של השערות, תהליך התפתחות הפתרונות באמצעות אלגוריתמים גנטיים אינו דטרמיניסטי, כלומר אין ערובה למציאת הפתרון הטוב ביותר במספר קבוע של איטרציות. אבל היתרון של אלגוריתמים גנטיים הוא היכולת שלהם להתקדם בהתמדה לעבר פתרונות טובים יותר לאורך זמן. על ידי שימוש במספר קבוע של איטרציות. כל איטרציה, או דור, מאפשרים לאלגוריתם לחקור אזור חדש במרחב ההשערה, ולחדד את הפתרונות תוך כדי.

הפעולות הגנטיות של הcrossover והmutation בשימוש ב-GA שלנו מאפשרות חיפוש רחב ואקראי של המרחב. הcrossover מחליפה חלקים בין פתרונות כדי לחקור שילובים חדשים, והmutation מציגה שינויים קטנים כדי לקדם גיוון ולמנוע מהאלגוריתם לקפוא באופטימיות מקומית. בהקשר זה, קביעת מספר קבוע של איטרציות מאפשרת ל-GA לחפש בהרחבה במרחב ההשערה, ולקדם התפתחות הדרגתית לעבר פתרונות אופטימליים או כמעט אופטימליים.

לפיכך, על ידי בחירה במספר קבוע סביר של איטרציות, אנו מייצרים איזון בין הגורמים הללו, ובמקביל מבטיחים שהאלגוריתם מקבל הזדמנות מספקת לגלות מפתח צופן יעיל בתוך מרחב כה עצום של השערות.

**גרפים:**

****גרף זה מציג את התקדמות האלגוריתם הגנטי על ידי הצגת ערך הכשירות הגבוה ביותר שהושג בכל איטרציה. בציר ה-x, ניתן לראות את מספרי האיטרציות. שמתחילים מ-1 ועולים ל-maxit. ציר ה-y מייצג את ערך הכשירות הטוב ביותר שהושג בכל איטרציה בהתאמה. הקו המשורטט בגרף מתאים לערך הכשירות הגבוה ביותר שנמצא בכל איטרציה, ומראה כיצד הכשירות של הפתרון הטוב ביותר משתנה לאורך זמן ככל שהאלגוריתם הגנטי מתקדם.

**חלק ב**

**המודל הדארוויניסטי**

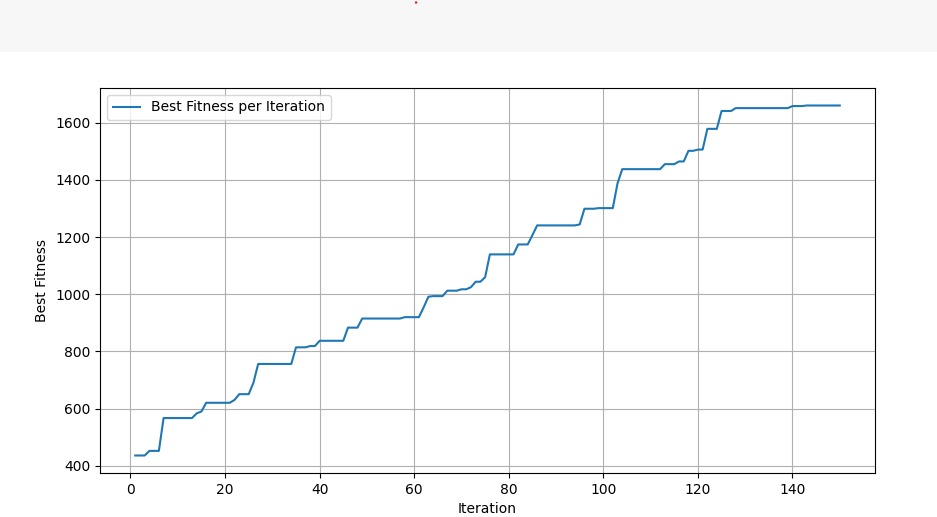
בגישה הדרוויניסטית לאלגוריתמים גנטיים מתקיים עקרון בו המתאימים ביותר שורדים. עיקרון זה מגולם על ידי היפרפרמטר beta, המוגדר לערך גבוה – 100.

ערך beta גבוה זה מעניק יתרון גדול לפרטים הכי חזקים באוכלוסייה, שכן יש סיכוי גבוה יותר שייבחרו ליצור צאצאים חדשים ויזכו לשרידות בתור הבא. לעומת זאת, האינדיבידואלים הפחות מתאימים נמצאים בנחיתות ניכרת ויש סיכוי נמוך בהרבה להיבחר לצורך יצירת צאצאים. תהליך הבחירה הזה דומה מאוד לתרחיש הישרדות הכשירים ביותר שחזה דרווין, שבו רק הפרטים החזקים ביותר, אלו המתאימים ביותר לסביבתם, שורדים ומתרבים.

למרות שגישה זו נראית יעילה אינטואיטיבית בקידום הדור של אנשים בכושר הולך וגובר, יש לה חיסרון חשוב: היא יכולה להפחית באופן משמעותי את המגוון הגנטי של האוכלוסייה. מכיוון שהפרטים הפחות מתאימים אינם מתחרים באופן עקבי, מגוון הפתרונות הגנטיים השונים הקיימים באוכלוסיה יכול להיות מצומצם יותר עם הזמן. זה דומה לתופעה הנצפית בטבע המכונה סחיפה גנטית, שבה אללים מסוימים (או פתרונות, במקרה של האלגוריתם שלנו) יכולים להתקבע באוכלוסייה בעוד שאחרים נעלמים לחלוטין.

צמצום הגיוון בדרך זו מגביל למעשה את יכולת ה'חקירה' של האלגוריתם, כלומר את יכולתו לחפש מגוון רחב של פתרונות אפשריים כדי למצוא את האופטימום הגלובלי. במקום זאת, האלגוריתם הופך ל"חמדן" יותר, מתמקד בפתרונות המבטיחים ביותר שנמצאו עד כה. התמקדות חזקה זו על פני חקירה של מרחב החיפוש יכולה להוביל את האלגוריתם להתכנס לאט לעבר האופטימום הגלובלי ואף להסתכן בללכוד באופטימיות מקומית.

זה קורה מכיוון שהאופטימום הגלובלי (הפתרון הטוב ביותר האפשרי) אולי לא בהכרח קרוב לפתרונות המבטיחים ביותר שנמצאו באוכלוסיה הראשונית. הוא עשוי להיות ממוקם באזור אחר של מרחב הפתרונות לחלוטין. אם האלגוריתם ממוקד מדי בניצול הפתרונות שנמצאו עד כה ואינו חוקר מגוון רחב מספיק של פתרונות, ייתכן שהוא לעולם לא ימצא את האופטימום הגלובלי.



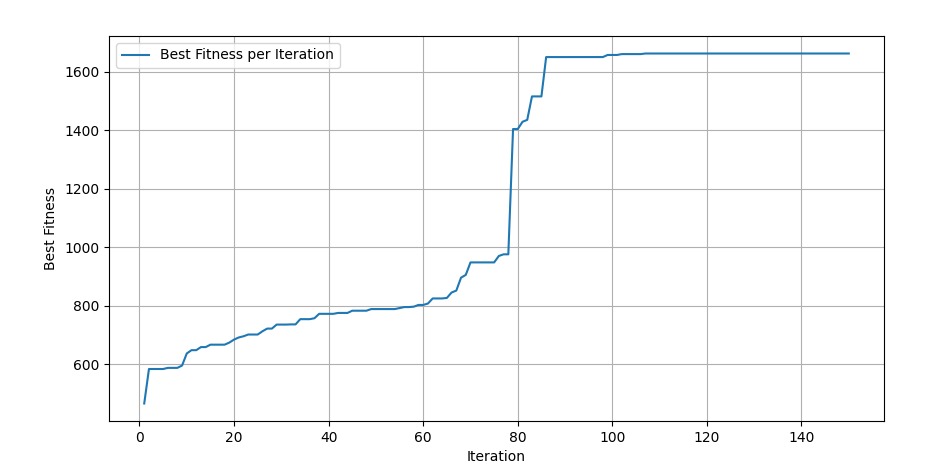
**המודל הלמארקי**

המודל של אלגוריתמים גנטיים למארקי ניגש לתהליך האופטימיזציה באופן שונה מאשר מקבילו הדרוויניסטי. בעוד שהמודל הדרוויניסטי מדגיש את הישרדות החזקים ביותר, המודל של למארקי מספק לכל הפרטים הזדמנות שווה להעביר את הגנים שלהם, ללא קשר לכושרם. עיקרון זה מגולם על ידי הגדרת היפרפרמטר beta ל-0 במודל הלמארקי.

כאשר beta מוגדר ל-0, לחץ הבחירה מנוטרל למעשה. המשמעות היא שגם לאנשים הפחות בכושר יש סיכוי לתרום את הגנים שלהם לדור הבא. גישה זו של שוויון הזדמנויות מובילה לחקר גדול יותר של מרחב הפתרונות. במילים אחרות, האלגוריתם מסוגל לדגום מגוון רחב יותר של פתרונות אפשריים, כולל אלו שבהתחלה עשויים להיראות פחות מבטיחים.

זה דומה לעקרון הירושה של למרק, שבו אורגניזם יכול לרכוש או לאבד תכונות לאורך חייו, ולאחר מכן להעביר את התכונות הללו לצאצאיו. באופן דומה, באלגוריתם הגנטי של למארק, לפתרונות פחות מתאימים יש הזדמנות להיבחר, ​​אולי לעבור mutation או crossover, ואולי להתפתח לפתרונות אופטימליים יותר לאורך זמן.

ערך beta נמוך מביא לעלייה מהירה בערך הכושר הכולל, שכן כל אדם, לא רק המתאימים ביותר, תורמים לחקר מרחב הפתרונות. האלגוריתם סורק בעצם את כל האוכלוסייה בכל איטרציה, ומניב כיסוי רחב של מרחב הפתרונות.

עם זאת, בעוד שעיקרון שוויון הזדמנויות זה מעודד חקר של מרחב ההיפותזות, הוא עשוי לבוא במחיר של חוסר התאמה. ייתכן שהאלגוריתם לא מעודד את הדחף התחרותי לחדד ולייעל את הפתרונות הטובים ביותר שנמצאו עד כה, משום שפתרונות פחות מתאימים מקבלים הזדמנויות כמעט שוות להתרבות. הדבר עלול לגרום לכך שהאלגוריתם יחמיץ אופטימיזציה נוספת של פתרונות בהתאמה גבוהה.

לסיכום, כאשר אנו משווים בין שלושת הגישות ניתן לראות כי ההרצה בה beta=1 היא ההרצה שמספקת עבורנו את התוצאות הטובות ביותר ואת זמן ההתכנסות הנמוך ביותר עבור האלגוריתם. באופן זה יש העדפה לטובת סלקציה של האינדיבידואלים הכשירים ביותר לטובת העמדת צאצאים בתור הבא אך עם זה ישנו סיכוי עבור שאר הפריטים באוכלוסייה להביחר ולהעמיד צאצאים באופן בן מתקיים שימור של הExpressivity וגם של הפרטים הטובים באוכלוסיה בצורה מספקת שמובילה לזמן ההתכנסות הקצר ביותר.