בס"ד

ביולוגיה חישובית – שידוך גנטי

דו"ח זה הינו עבור אלגוריתם גנטי למציאת שידוך אופטימלי. האלגוריתם קורא קובץ המכיל אוסף של העדפות של שידוכים, כך שהעדפה הגדולה יותר תהיה במיקום נמוך יותר בהעדפת האדם, מבצע שיפורים על ידי עקרונות אלגוריתם גנטי ולבסוף מוצא פתרון טוב.

בכללות, **אלגוריתמים גנטיים** בנויים ממספר שלבים פשוטים:

1. יצירת האוכלוסייה – מרחב הפתרונות
2. ביצוע הכלאה – אליטיזם, זיווג בין פתרונות מהדור הקודם
3. ביצוע מוטציות על חלק מהילדים החדשים
4. יצירת דור חדש בשילוב האליטיזם יחד עם הילדים החדשים

וחוזר חלילה עד התכנסות.

דו"ח זה מחולק לשמונה חלקים:

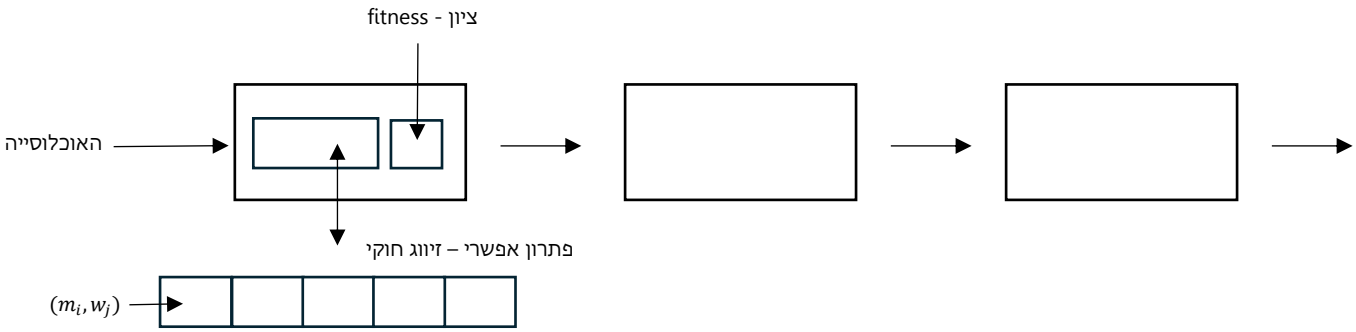
א. ייצוג הבעיה והפתרונות
ב. הגדרת הפתרון האופטימלי
ג. פונקציית הערכה (fitness)
ד. הכלאה בין פתרונות שונים (crossover)
ה. ביצוע מוטציות
ו. טיפול בהתכנסות מוקדמת ומקסימום לוקאלי
ז. האופרטורים הגנטיים באלגוריתם
ח. דוגמות הרצה

חלק א' - ייצוג הבעיה והפתרונות

כשם שהזכרנו לעיל, האלגוריתם מקבל נתונים על העדפות זיווג של 30 גברים ו- 30 נשים, כך שכל משתתף מדרג את העדפותיו לגבי בין המין השני מהגבוה לנמוך.

את הבעיה בחרנו לייצג על ידי רשימה של תבניות, כאשר גודלה של הרשימה שווה לגודל האוכלוסייה. כל תבנית מורכבת מזוג סדור של רשימה של פתרונות (זיווגים חוקיים) המיוצגים כמערך של זוגות סדורים, ולבסוף הציון (הסבר על הציון – fitness, בחלק ג' בדו"ח) שלהם.

נמחיש על ידי תרשים:



חלק ב' – הגדרת הפתרון האופטימלי

בד"כ כשאנו חושבים על פתרון אופטימלי אנו מסתכלים על הפתרון האופטימלי ה"אמיתי" – משמע הפתרון לבעיית האופטימיזציה.

לעומת זאת, בהרצת אלגוריתמים גנטיים, אנו לא בהכרח נגיע לפתרון האופטימלי הגלובלי, אלא אנו יכולים לעצור לפני כן. למשל, בקוד שלנו כאן, בחרנו לבצע עצירה כאשר אנו רואים התכנסות במשך עשרה דורות עבור הציון 90.

ההתכנסות ובדיקת הציון נעשית ביחס לפתרון הטוב ביותר באותו הדור, כלומר הזיווג החוקי בעל הציון הגבוה ביותר בדור עליו אנו מסתכלים.

חלק ג' – פונקציית הערכה (fitness)

דיברנו והזכרנו כבר מספר פעמים את המונח "ציון" עבור פתרון. על מנת לבצע אלגוריתם גנטי עלינו להגדיר מעין "יחס סדר" בין הפתרונות השונים, על ידי מדד לטיב פתרון הקיים באוכלוסייה.

נשים לב, שבצורה אינטואיטיבית פתרון טוב, יהיה פתרון שייתן לכל אחד ואחת, זיווג בצורה מיטבית על פי ההעדפה שלהם. לכן, בחרנו לבצע את הערכת הפתרונות באופן הבא:

```
def my_fitness(m):
    """
    Calculate fitness score for a given match.
    """
    global CALLS_FITNESS_FUNCTION
    score = 0
    for man, woman in m:
        man_score = (COUPLES_AMOUNT - int(MEN_FAVOR[man - 1][woman - 1])) *
                    FITNESS_VALUE
        woman_score = (COUPLES_AMOUNT - int(WOMEN_FAVOR[woman - 1][man - 1])) *
                      FITNESS_VALUE
        score += (man_score + woman_score) / 2 + 0.01
    CALLS_FITNESS_FUNCTION += 1
    # Normalized with COUPLES_AMOUNT
    return score / COUPLES_AMOUNT
```

1. נאתחל משתנה לציון הפתרון ל- 0.
2. בצע לולאה על כל הזוגות איש, אישה בזיווג:
   a. חשב עבור האיש את טיב האישה שקיבל על פי העדפותיו, כך שהאישה הטובה ביותר תקבל 99.9 וכל אחת אחרת תקבל ציון יחס ביחס למקומה -> ציון האיש.
   b. חשב באותו האופן עבור האישה -> ציון האישה.
   c. הוסף לציון הפתרון ממוצע בין ציון האיש לציון האישה (ותוסיף 0.01 כדי לעגל למאה).
3. עדכן את כמות הקריאות לפונקציה (מדד להמשך).
4. החזר את ממוצע ציוני הפתרונות.

כעת, נסביר את הרעיון מאחורי הלוגיקה.

ראשית, אנו נותנים משקל אף להעדפות האיש והן להעדפות האישה בעת חישוב טיב הפתרון, כך אנו דואגים תמיד לבדוק שהאמד שלנו אכן משקף את העדפות שני הצדדים.

בנוסף, על מנת לבצע אמידה לטיב פתרון בצורה מיטבית, אנו נותנים משקל יחסי לכל זיווג (בין לאיש ובין לאישה) על פי העדפותיו, בכך שאנו "קונסים" זיווג על פי הידרדרות מיקומו ביחס להעדפות.

סה"כ אנחנו מנרמלים את המדד כדי שיהיה בין 0.01-100, על מנת לאפשר הבנה ברורה יותר לטיב הפתרון הסופי למתבונן מהמצד.

חלק ד' – הכלאה בין פתרונות שונים (crossover)

צעד ההכלאה, הינו צעד קריטי במימוש האלגוריתם הגנטי.

<u>צעד זה מחולק אצלנו לשלושה חלקים עיקריים:</u>

1) ביצוע אליטיזם בקרב האוכלוסייה והעברתם לדור הבא
2) יצירת זיווגים מתוך המחצית העליונה של האוכלוסייה על פי טיבם
3) ביצוע מוטציות

נתבונן בקוד ולאחר מכן נסביר את ההיגיון מאחוריו:

```
def crossover(parents):
    """
    Perform crossover between parents to generate children.
    """
    global POPULATION_SIZE
    children_amount = POPULATION_SIZE * (1-SELECTION_RATE)
    children_pop = []

    while children_amount > 0:
        daddy = weighted_choice(parents)[1]
        mummy = weighted_choice(parents)[1]

        # Give another chance to chose different parents (using while could cause
          infinite loop)
        if mummy == daddy:
            mummy = weighted_choice(parents)[1]

        # Generate crossover point
        crossover_point = random.randint(1, COUPLES_AMOUNT - 1)

        child1 = daddy[:crossover_point] + mummy[crossover_point:]
        child2 = mummy[:crossover_point] + daddy[crossover_point:]

        children_pop.append((0, child1))
        children_pop.append((0, child2))

        children_amount -= 2  # We create 2 children per iteration

    children_pop = mutation(children_pop)
    children_pop = validation(children_pop)
    children_pop = do_fitness(children_pop)

    return children_pop
```

1. נגדיר את כמות הילדים שניצור על פי כמות האוכלוסייה פחות האליטיזם – הפתרונות הכי מוצלחים בדור הנוכחי שנעביר לדור הבא (גודל האליטיזם נבחר על ידי היפר-פרמטר עליו נפרט בחלק ז' בדו"ח)
2. נאתחל רשימה ריקה עבור הילדים בדור הנוכחי
3. בצע לולאה עד שיצרת את כלל הילדים:
    a. תגריל אבא מתוך האליטיזם על פי הסתברויות התלויות בטיב הפתרון.
    b. תגריל אמא באותו האופן.
    c. אם האמא שנבחרה שווה לאבא, תגריל אמא אחרת (נעשה פעם אחת בלבד).
    d. בחר נקודת חיתוך i כלשהי.
    e. צור ילד ראשון על פי האיברים עד הנקודה ה- i באב והחל מהנקודה ה- i באם.
    f. צור ילד שני ממה שנותר באב ובאם.
    g. הוספת הילד הראשון לרשימת הילדים.
    h. הוספת הילד השני לרשימת הילדים.
    i. עדכן את מספר הילדים שנוצרו.
4. בצע מוטציות על חלק מהילדים (יוסבר בחלק ה' בדו"ח).

5. בצע ולידציה על הילדים, ותקן את הפתרונות שלהם במידת הצורך.
6. עדכן את הציון עבור כלל הילדים.
7. החזר את הילדים החדשים באוכלוסייה.

כעת, נסביר את הרעיון מאחורי הקוד.

המטרה בהכלאה, היא ליצור דור חדש "מוצלח" יותר מהדור הקודם. לכן, קודם כל "נשמור בצד" – כלומר, נעביר כמה מהם, את הפתרונות הטובים (תיאור ופירוט ההיפר-פרמטרים בחלק ז' בדו"ח) כמו שהם לדור הבא. כך אנו משמרים פתרונות מוצלחים מדור לדור ולא מאבדים אותם בתהליך ההכלאה – קבוצה זו קרויה "אליטיזם".

לאחר מכן, אנו צריכים להשלים את שאר האוכלוסייה עבור הדור הבא. כשם שהסברנו לעיל, אנו רוצים ליצור דור מוצלח יותר מהדור שקדם לו, לכן, נבצע הכלאה בין המחצית העליונה של האוכלוסייה וניצור מהם פתרונות חדשים ובתקווה מוצלחים יותר מהדור שלפני.

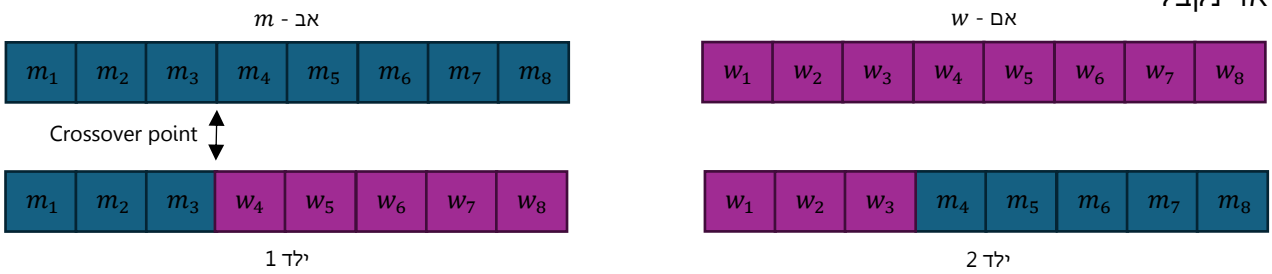
תהליך ההכלאה מתבצע כך, שהוא מתעדף בחירת "הורים" על פי טיב הציון שלהם, כך שפתרון בעל ערך גבוה יותר יקבל סיכוי גבוה יותר להיות מוגרל. לאחר מכן, מוגרלת נקודה כלשהי לאורך אורכו של הפתרון, שם מתבצע "חיתוך" של פתרון האב ופתרון האם ויצירת שני ילדים.

נציין רק, כי בתהליך בחירת האם (ההורה השני) ביצענו בדיקה חד פעמית האם האמא זהה לאבא, זאת על מנת לבצע מאמץ מסוים להימנע מהכלאה של פתרון עם עצמו ובכך לשכפלו 3 פעמים. הסיבה שלא מימשנו זאת על ידי לולאת while המחכה לקבל אם ששונה מהאב, הינה הימנעות מלולאה אינסופית וייעול זמן הריצה.

נמחיש את תהליך יצירת שני ילדים חדשים, על ידי דוגמה פשוטה:

- נניח כי הגרלנו את אב m ואם w כך ש- $m \neq w$
- נניח כי הגרלנו crossover_point = 3

אזי נקבל -



לאחר יצירת כלל הילדים, אנו מבצעים שלושה דברים.

ראשית, אנו מבצעים מוטציות על חלקם (התהליך מתואר בפירוט בחלק ה' בדו"ח), שנית, אנו מבצעים וידוא על כלל הילדים, כלומר אנו בודקים האם הפתרונות שקיבלנו אכן חוקיים. הגיוני שייוצרו פתרונות שאינם חוקיים לאחר הכלאה, כיוון שהגרלת ההורים ונקודת החיתוך נעשה בצורה רנדומית. במידה ופתרון איננו חוקי, נבצע תיקון בצורה שרירותית במקומות החוזרים על עצמם.

דוגמה לתיקון (המספרים המודגשים תוקנו):

לכלל הילדים לאחר תהליכי המוטציות והולידציה, מחושב הציון שלהם והם נשלחים יחד עם הוריהם (האליטיזם) לדור הבא.

הסיבה לשימוש במחצית מהאוכלוסייה לתהליך ההכלאה, הינה כי לעיתים ישנם פתרונות פחות מוצלחים באוכלוסייה שיש להם חלק מסוים שהוא מעולה ואפילו אופטימלי, ואם נזנח אותם לחלוטין אנו עלולים לאבד חלקים טובים סתם. בנוסף, אנו יודעים שכאשר אוכלוסייה שלמה נוצרת על ידי אב קדמון אחד (או כמה בודדים) סופה להיכחד.

סה"כ התהליך משמר את הפתרונות הטובים ביותר מהדור הנוכחי, יוצר פתרונות חדשים על בסיס חיבור והכלאה בין זוג פתרונות טובים, ועל כן אמור ליצור שיפור בדור הבא.

חלק ה' – ביצוע מוטציות

במהלך הדו"ח דיברנו על שימוש בהליך יצירת מוטציות. המוטציות הכרחיות על מנת ליצור שילובים של זיווגים שלא בהכרח היו נוצרות אילולא היינו מבצעים תהליך מלאכותי זה של יצירת המוטציות, כך יצירת המוטציות מאפשרת את הרחבת הגיוון של האוכלוסייה והגעה לפתרונות נוספים.

את תהליך המוטציות ביצענו באופן הבא:

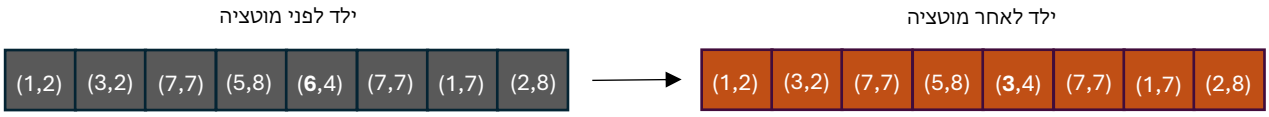
```
def mutation(children):
    """
    Perform mutation on a given set of children.
    """
    updated_kids = []
    for score, child in children:
        mutated_child = []
        for gene in child:
            if random.random() < MUTATION_RATE:
                # Mutate the gene with a random value
                mutated_gene = (gene[0], random.randint(1, COUPLES_AMOUNT))
            else:
                mutated_gene = gene
            mutated_child.append(mutated_gene)
        updated_kids.append((0, mutated_child))
    return updated_kids
```

1. נאתחל רשימה ריקה עבור כלל האוכלוסייה הניתנת בקלט הפונקציה.
2. בצע לולאה על כל סוג סדור של ניקוד ופתרון מהאוכלוסייה:
    a. נאתחל רשימה ריקה עבור כל הזוגות בפתרון.
    b. בצע לולאה על כל אחד מהזוגות ברשימה:
        i. בדוק אם המספר המוגרל קטן מהערך הסף עבור ביצוע מוטציה (הסבר על ההיפר-פרמטרים בחלק ז' בדו"ח):
            1. אם כן - הגרל שידוך אחר לבן הזוג.
            2. אחרת - אל תשנה את השידוך לזוג זה.
        ii. הכנס לרשימת הזוגות את השידוך שהתקבל.
    c. הכנס לרשימת האוכלוסייה את הזוג הסדור המורכב מהציון 0 (מתבצע חישוב ציון קונקרטי עבור כלל הילדים רק לאחר הולידיה על מנת לחסוך בקריאות לפונקציית הציון) ורשימת השידוכים החדשה.
3. החזר את האוכלוסייה החדשה שנוצרה.

כעת, נסביר את הרעיון מאחורי הלוגיקה.

אנו עוברים על כלל האוכלוסייה החדשה שיצרנו בתהליך ההכלאה, אשר עליה נבצע את תהליך הוספת המוטציה. עבור כל פרט באוכלוסייה נעבור על כל זוג במערך השידוכים שלו ובהסתברות כלשהי, נשנה את בת זוגתו של הבן בשידוך. כך אנו מוסיפים בצורה מלאכותית פתרונות נוספים למרחב הפתרונות של האוכלוסייה בעודנו שומרים את עקרונות ההכלאה עליהם דיברנו לפני.

נשים לב לדוגמה הבאה, עבור הכלאה העוברת מוטציה בחלקה (החלק המודגש):



נשים לב, שהילד בשלב זה יכול להיות פתרון לא חוקי.

חשוב לציין, כי בפונקציה זו אנו לא מגדירים לכל רשימת זיווגים את הציון שלה, מכיוון שהפתרון עובר ולידציה ורק לאחר מכן קבלת ציון בפונקציית ההכלאה, זאת על מנת לחסוך בקריאות לפונקציית הציון לפתרונות שאולי ישתנו (במידה ואינם תקינים).

חלק ו' – טיפול בהתכנסות מוקדמת או מקסימום לוקאלי

אחת הבעיות העלולות להיווצר באלגוריתם גנטי הינו התכנסות מוקדמת או מקסימום לוקאלי, כלומר, הגעה למצב בו האוכלוסייה מתקבעת לפתרון טוב (יחסי), אבל לא מצליחה ליצור ממה שיש בידה פתרון מוצלח יותר. מצב זה הגיוני בהתחשב בשימור האליטיזם אותו אנו מבצעים בכל מעבר דור, הכלאת המחצית הטובה יותר של האוכלוסייה ומיעוט השימוש במוטציות.

ניסינו לחשוב כיצד לאתר מצב עם פוטנציאל רב לבעיה זו וכיצד לפתור אותה. לשם כך הכנסנו שני מנגנוני "בטיחות" לריצת הקוד, האחד מהווה אינדיקציה לגיוון האוכלוסייה והשני לזיהוי plateau בפתרון הטוב ביותר.

- המנגנון הראשון הינו, בדיקה בכל דור האם הפרש ציוני ערכי הקיצון, הגדול ביותר והקטן ביותר, בדור קטן מ- 5. כאשר ההפרש אכן קטן, אנו יודעים שכלל ערכי האוכלוסייה קרובים מידי, דבר בעייתי במידה ונרצה להגיע לפתרונות מוצלחים יותר.
- המנגנון השני הינו, בדיקה בכל דור האם הערך המקסימלי זהה במשך עשר ריצות ברצף, דבר המסמל התכנסות והגעה למקסימום לוקאלי (ואולי גלובלי).

כאשר אחד משני הכלים הללו מזהה מצב בעייתי מעין אלו, אנו בודקים מה ערך הפתרון הכי טוב בדור הנוכחי. במידה וערך הפתרון גדול מהציון 90, אנו מסיימים את הריצה, כי אכן התכנסנו לפתרון מעולה ויתכן שאופטימלי (הרי לפי הגדרת הציון שלנו 100 מהווה מצב בו כולם מקבלים את שרצו, אך לפעמים רצונות מתנגשים).

במידה והציון הגבוה ביותר קטן או שווה ל- 90, אנו מנסים לצאת מהקיבעון הלוקאלי על ידי ביצוע סכמת המוטציה על כלל האוכלוסייה, למעט הפתרון הטוב ביותר באותו הדור (על מנת לקבל פתרון בדור הבא גדול או שווה לאחד בדור הזה), כך אנו משמרים על מוניטוניות עולה. כך אנו מבצעים "ריענון" על כלל איברי האוכלוסייה ומאפשרים בצורה מסיבית ליצור פתרונות נוספים.

חלק ז' – האופרטורים הגנטיים באלגוריתם

נסביר על שני אופרטורים גנטיים בהם השתמשנו בקוד:

1) MUTATION RATE –
ערך זה מגדיר את ההסתברות לביצוע מוטציה על זיווג מסוים בפתרון i בעת ביצוע סכמת המוטציות, לאחר ביצוע הרצות עם מגוון ערכים, עבור ערך זה בחרנו ב- 0.05.
במהלך הריצות שמנו לב, שעבור ערך גדול יתרחשו שינויים רבים וביצוע ההכלאה על ידי מחצית האוכלוסייה בעלת הציונים הגבוהים יותר יאבד מערכו, בנוסף לא נוכל לקחת ערך קטן מידי כמו 0.01, כי עבורו כמות השידוכים שישתנו תהיינה קטנה מידי ולא תהיה השפעה מרובה (אם בכלל) על האוכלוסייה, דבר שיגרור אוכלוסייה לא מגוונת, אי הגעה לפתרונות חדשים ואף ימנע יציאה ממקסימום לוקאלי בעת הצורך.

2) SELECTION RATE –

ערך זה מגדיר את ה"אליטה", אחוז הזיווגים הטובים ביותר מהאוכלוסייה שנעביר לדור הבאה, ביצענו ניסיונות רבים עם ערכים שונים עד שלבסוף בחרנו בערך 0.2.

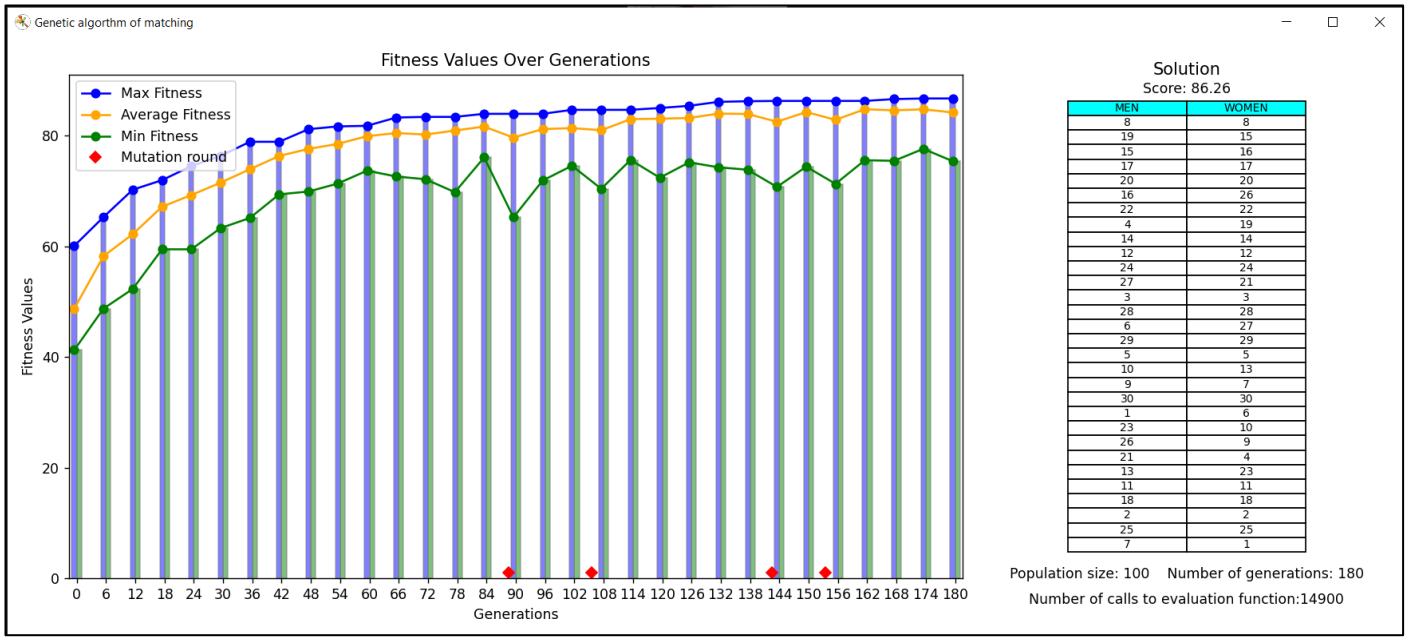
כאשר מנסים לאמוד בחירה נכונה לערך זה, רואים שעבור ערך גדול מידי נקבל אוכלוסייה חד גונית החוזרת על עצמה, זאת מפני שהאליטה עוברת לדור הבא כמות שהיא. בנוסף, עלינו לשים לב שאנו לא מקטינים ערך זה יותר מידי, אחרת נאבד פתרונות טובים בין דור אחד למשנהו.

חשוב לציין, כי ערך זה קובע בצורה ישירה את כמות האיברים בהכלאות, כיוון שאת ההכלאות אנו מבצעים כדי ליצור את שאר האוכלוסייה שנותרה בדור הבא (במקרה שלנו $1 - 0.2 = 0.8$).

בנוסף, ניסינו שלוש קומבינציות (פירוט בחלק ח' בדו"ח) ליחס אוכלוסייה וכמות דורות מקסימלית, ולבסוף בחרנו באוכלוסייה בגודל ובכמות דורות, עקב קבלת תוצאות טובות יותר (ביחס לטיב הפתרון המתקבל ומספר הקריאות לפונקציית הציון) בממוצע על פני האפשרויות האחרות.

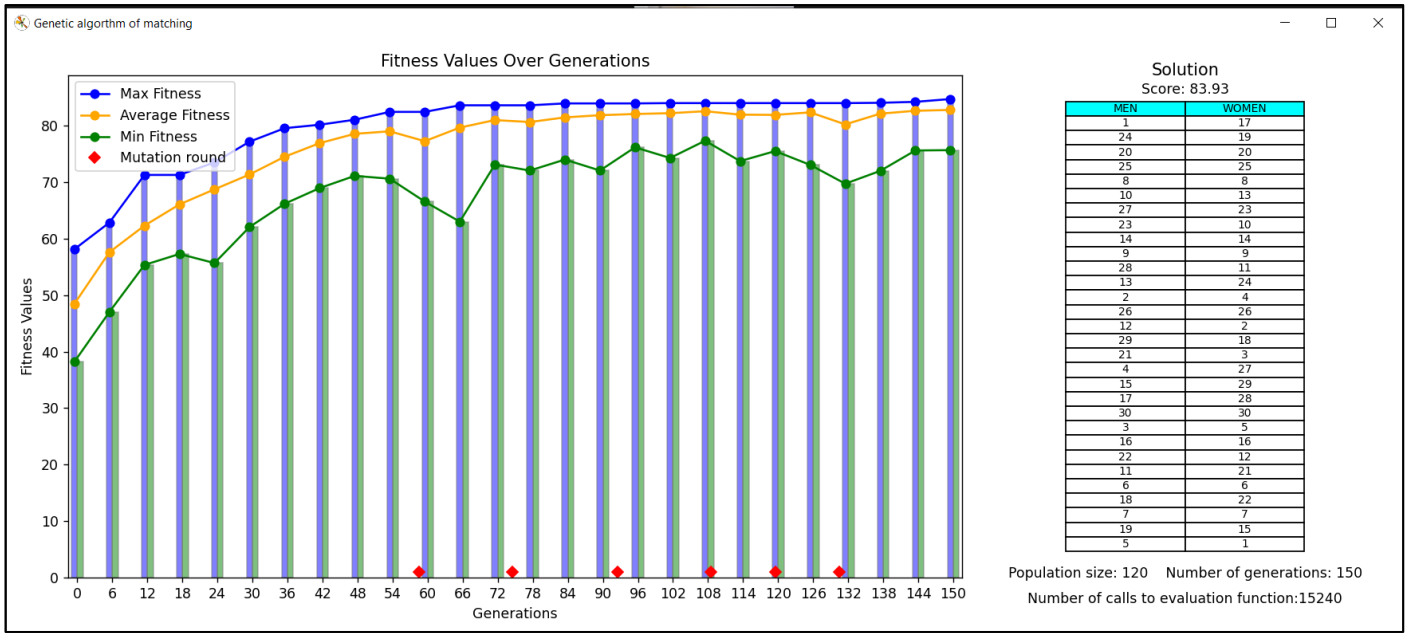
חלק ח' – דוגמות הרצה

הרצה ראשונה | מספר דורות: 180 | גודל אוכלוסייה: 100



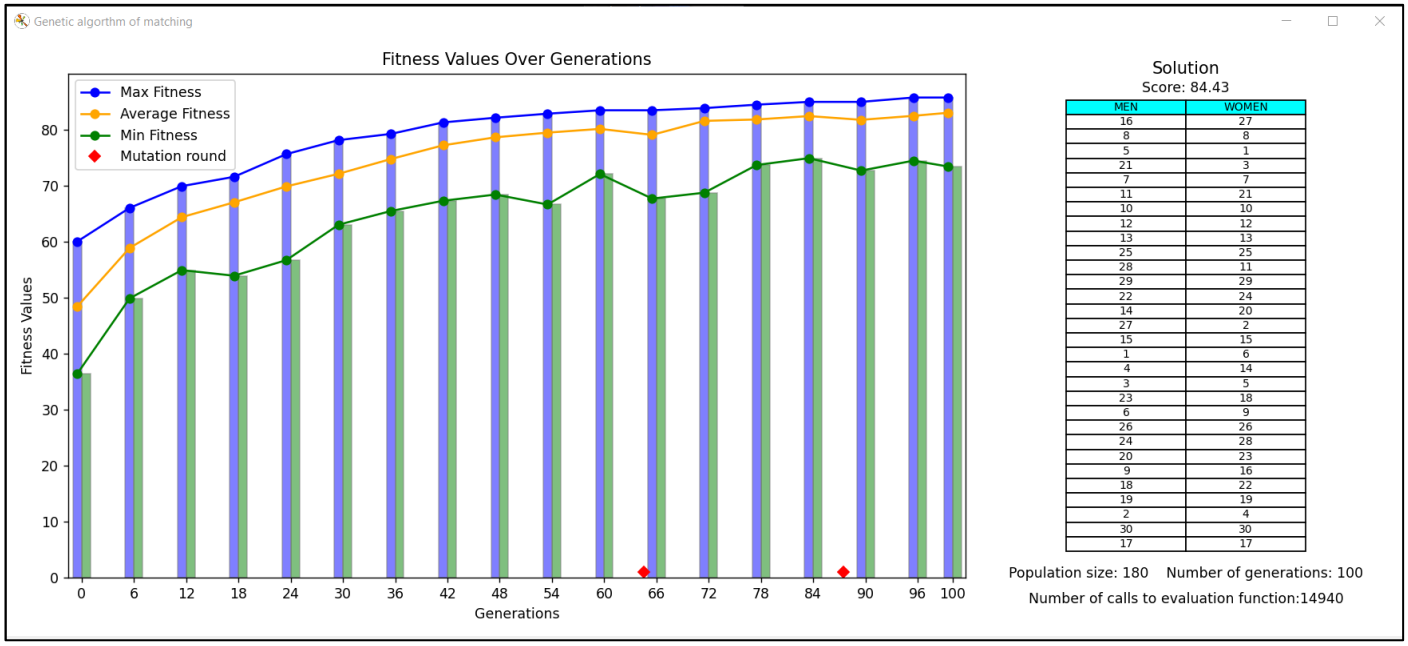
MEN	WOMEN
8	8
19	15
15	16
17	17
20	20
16	26
22	22
4	19
14	14
12	12
24	24
27	21
3	3
28	28
6	27
29	29
5	5
10	13
9	7
30	30
1	6
23	10
26	9
21	4
13	23
11	11
18	18
2	2
25	25
7	1

הרצה שנייה | מספר דורות: 120 | גודל אוכלוסייה: 150



MEN	WOMEN
1	17
24	19
20	20
25	25
8	8
10	13
27	23
23	10
14	14
9	9
28	11
13	24
2	4
26	26
12	2
29	18
21	3
4	27
15	29
17	28
30	30
3	5
16	16
22	12
11	21
6	6
18	22
7	7
19	15
5	1

הרצה שלישית | מספר דורות: 100 | גודל אוכלוסייה: 180



את היחס הטוב ביותר, בין הציון לפתרון המתקבל לבין מספר הקריאות לפונקציית, קיבלנו בהרצת האופציה הראשונה בה הזיווג שהוחזר קיבל את הציון 86.26, עם מספר הקריאות מינימלי לפונקציית חישוב הציון 14900 פעמים.

מן הראוי לציין, שכאשר האלגוריתם נתקל במקסימום מקומי (כמפורט בחלק ו'), הוא מבצע מוטציות לכלל האוכלוסייה למעט הפתרון הטוב ביותר (מסומן בנקודות אדומות) מה שגורם לעיתים להעלאת מדד הזיווג המקסימלי בדורות לאחר מכן ולשימור פונקציה מונוטונית עולה ביחס לפתרון הטוב ביותר, דבר המראה בכללותו על טיב הפתרון.

על מנת להריץ את הקוד יש להשתמש במערכת windows, לשים קובץ טקסט יחד עם דאטה עבור העדפות האוכלוסייה בשם - "GA_input.txt" בתיקייה יחד עם קובץ ההרצה "main.py", וללחוץ עליו פעמיים. יש לשים לב שקובץ ההרצה איננו מביא כפלט את הגרפים לעיל, על מנת לאפשר הרצה ללא התקנת ספריות.