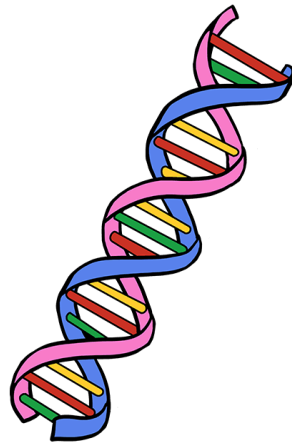


אלגוריתמים גנטיים

מגישים: טל אריאל זיו, יעל דהרי



תוכן עניינים

3	מבוא
4	פירוט האלגוריתם הגנטי
11	הרצות שונות ותוצאותיהן
17	מסקנות
18	סיכום

מבוא

נעמיק ברעיון של אלגוריתמים גנטיים, ונרצה לפתור באמצעות קונספט זה בעיית שידוכים מקסימלית. בעיית השידוכים מתארת מצב בו ישנם 30 גברים ו-30 נשים, כאשר כל משתתף מדרג את העדפותיו לגבי בני המין השני. מטרתנו היא למצוא התאמה (matching) אופטימלית בין הגברים לנשים שתמקסם את שביעות הרצון הכוללת של כל המשתתפים.

האלגוריתם מתחיל את הריצה עם m פתרונות רנדומליים. לאחר מכן, האלגוריתם הגנטי מתחיל לפעול ומבצע פעולות של הכלאה, מוטציות, העדפה ואופרטורים נוספים עליהם נפרט בהמשך המטלה. לבסוף, לאחר n דורות, אנו עוצרים את האלגוריתם ובוחנים את הפתרונות. סך הכל, מכפלת המשתתפים $m * n$ יוצא 18,000.

את איכות השידוך של המשתתפים אנו קובעים לפי פונקציית $fitness$ אשר מתחשבת בהעדפת הגבר והאישה באופן שווה. ככלל שהזוגות יותר מרוצים, כך ערך ה $fitness$ של הפתרון יהיה גבוה יותר. אנו רוצים למצוא פתרון מקסימלי לפונקציית ה $fitness$, כלומר הזיווג שיביא לערך מקסימום (מוחלט) של הפונקציה.

בהתאם, בכל דור שמרנו נתונים שיעזרו לנו בהמשך לנתח את טיב האלגוריתם, ביניהם: ערך התאמה מקסימלי, ערך התאמה מינימלי וערך ממוצע. מנתונים אלה יצרנו גרפים, אשר מראים באופן ויזואלי את התקדמות הפתרון שלנו.

נתאר בנוסף כיצד נמנענו מהתכנסות מוקדמת, וכיצד ניתן לראות זאת בתוצאות השונות.

בעמוד הבא נפרט על אופן ההורדה וההרצה של הקוד, וכן כמה **פרטים קריטיים** בנוגע לכך.

פירוט האלגוריתם הגנטי

בחלק זה נפרט פרטים על האלגוריתם הגנטי שלנו, כיצד ייצגנו את הפתרונות, איך הכלאנו בין פתרונות שונים, כיצד ביצענו מוטציות ואיך נמנענו מהתכנסות בשלב מוקדם מדי.

אופן ייצוג הפתרונות

בחרנו לייצג את הפתרונות כפרמוטציה של המספרים 1-30, כך שהמספר ה- i במיקום ה- j מייצג את השידוך של האישה ה- i לגבר ה- j . כלומר, כל פתרון הוא רשימה של 30 מספרים, כאשר כל מספר מייצג אישה ומספר התא שבו המספר מופיע מייצג גבר.

לדוגמה:

עבור המערך הבא - [10, 15, 3, 22, 27, 8 ...], גבר מספר 1 שודך עם אישה מספר 10, גבר מספר 2 שודך עם אישה מספר 15, גבר מספר 3 שודך עם אישה מספר 3, וכן הלאה.

ייצוג זה מאפשר לנו להזיז את הנשים בלבד בכל איטרציה, מה שחוסך זמן ומקום ומבטיח פתרון חוקי ללא צורך בבדיקות תקינות מורכבות.

בנוסף, ייצוג זה מונע את הצורך בתחזוקה של שתי פרמוטציות נפרדות (אחת לגברים ואחת לנשים) או ייצוג בעזרת המספרים 1-60, מה שמפשט את תהליך החישוב והתחזוקה. בהמשך, כאשר נגיע לתהליך ה- *crossover* – נסביר כיצד אנו שומרים על חוקיות הפתרון על אף ששיטה זו לא תמיד מניבה פתרון חוקי בתצוגה הנ"ל.

פונקציית ההערכה

פונקציית ההערכה שלנו מתמקדת בחישוב שביעות הרצון של כל אחד מבני הזוג. עבור כל שידוך, כל בן זוג מקבל ציון בהתאם למידת העדיפות שלו: בן הזוג בעדיפות הראשונה מקבל ציון 30, הבא בתור 29, וכן הלאה עד האחרון שמקבל ציון 1.

נחשב את הערך הזה הן לגבר והן לאישה ונשמור את הממוצע ביניהם. נמשיך בתהליך זה לכל הזוגות ברשימה ונחשב את ממוצע הציונים עבור כל הזוגות. הערך המתקבל יהיה פונקציית ההערכה שלנו.

```
def calculate_fitness(chromosome, men_preferences, women_preferences):
    total_satisfaction = 0
    n = len(chromosome) # Number of pairs

    for man in range(n):
        woman = chromosome[man]
        # Men's satisfaction
        men_preference_score = (n - men_preferences[man].index(woman))
```

```

# Women's satisfaction
women_preference_score = (n - women_preferences[woman].index(man))

# Average satisfaction for this couple
couple_satisfaction = (men_preference_score +
women_preference_score) / 2.0
total_satisfaction += couple_satisfaction

# Overall average satisfaction
average_satisfaction = total_satisfaction / n
return average_satisfaction

```

יתרונות פונקציית ההערכה

פונקציית ההערכה שבחרנו משקפת את סך שביעות הרצון של כל הנשים והגברים בצורה אחידה ומונעת מקרים שבהם ציון חריג של זוג אחד ישפיע בצורה לא פרופורציונלית על הערך הכללי. בכך, אנו מבטיחים שהפתרון יהיה מאוזן ומייצג את שביעות הרצון הכוללת בצורה הוגנת ויעילה.

הכלאה בין פתרונות – Crossover

בתהליך ההכלאה בין פתרונות (crossover), אנו יוצרים פתרון חדש (ילד) משני פתרונות קיימים (הורים). בתהליך זה, הפתרון שקיבל את הציון הגבוה ביותר נשאר ללא שינוי וממשיך לאוכלוסייה החדשה כדי להבטיח שנשמר את הפתרון הטוב ביותר שזוהה עד כה.

לאחר מכן, נבחר שני הורים בעזרת בחירת תחרות (tournament selection). בשיטה זו, נגריל k פתרונות רנדומליים מתוך האוכלוסייה, ונבחר את הפתרון בעל הציון הגבוה ביותר מביניהם להיות אחד ההורים. נחזור על התהליך לבחירת ההורה השני.

תהליך יצירת הילד

לאחר שבחרנו את שני ההורים, תהליך יצירת הילד מתבצע באופן הבא:

- בחירת נקודות התחלה וסוף: נגריל שני אינדקסים מ-1 עד 30 שייקבעו את נקודות ההתחלה והסוף עבור הקטע שנעתיק מההורה הראשון.
- העתקת קטע מההורה הראשון: נעתיק לילד את הערכים מההורה הראשון מהאינדקס ההתחלתי עד האינדקס הסופי.

- השלמת הערכים מההורה השני: נעבור על הילד ובכל מקום פנוי (כלומר, מיקום שבו לא הועתק ערך מההורה הראשון), נכניס ערך שנמצא בהורה השני ושאינו מופיע כבר בילד, לפי סדר הופעתם בהורה השני.

דוגמה לתהליך ה-crossover

נניח שיש לנו את שני ההורים הבאים:

הורה 1: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ..., 30]

הורה 2: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]

נניח שהגרלנו את נקודות ההתחלה והסוף להיות 5 ו-10.

העתקת קטע מההורה הראשון

הילד מקבל את הקטע מההורה הראשון בין האינדקסים 5 עד 10:

[None, None, None, None, 5, 6, 7, 8, 9, 10, None, ..., None]

השלמת הערכים מההורה השני

נתחיל מהתחלה של הילד, ונמלא את הערכים מההורה השני שלא נמצאים כבר בילד.

הילד המתקבל יהיה:

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

שמירה על חוקיות הפתרון

אופן בניית הילד עוזר לשלב מידע גנטי משני ההורים תוך שמירה על חוקיות הפתרון. זאת משום שכל ערך מופיע פעם אחת בלבד, והסדר שבו מופיעים הערכים נשמר באופן יחסי. זהו עקרון קריטי בבעיות שידוך שבהן כל מספר חייב להופיע פעם אחת בלבד ויש חשיבות לסדר שלהם.

סיכום

תהליך ההכלאה שתיארנו מאפשר שילוב אפקטיבי של מידע גנטי משני ההורים ליצירת פתרון חדש תוך שמירה על חוקיות הפתרון. בעזרת בחירת תחרות, אנו מבטיחים שההורים שנבחרים לתהליך הם בעלי סיכוי גבוה להיות פתרונות טובים, ובכך משפרים את איכות האוכלוסייה על פני הדורות.

מוטציות - Mutation

בתהליך האבולוציוני של הגנטיקה האלגוריתמית, מוטציות ממלאות תפקיד קריטי בשמירה על הגיוון הגנטי באוכלוסייה ומניעת התכנסות מוקדמת לפתרונות מקומיים. תהליך המוטציה שלנו מתבצע בצורה מבוקרת על מנת להבטיח שכל פתרון נשאר חוקי ולשפר את יכולת החיפוש שלנו במרחב הפתרונות.

תהליך המוטציה

המוטציות קורות באופן הבא:

1. הגרלת מספר רנדומלי: מגרילים מספר בין 0 ל-1 בהתפלגות אחידה. אם המספר המוגרל גדול מ-mutation rate, מבצעים מוטציה בפתרון הנוכחי.
2. בחירת אינדקסים להחלפה: דוגמים שני אינדקסים רנדומליים מתוך הפתרון בהתפלגות אחידה, ומחליפים בין הערכים שנמצאים במיקומים האלו בפתרון.

שמירה על חוקיות הפתרון

המוטציה שומרת על חוקיות הפתרון מכיוון שהיא מחליפה בין שני ערכים קיימים בפרמוטציה, ולכן תמיד נשארת בגבולות הפרמוטציה המקורית. כלומר, כל מספר בין 1 ל-30 יופיע בדיוק פעם אחת בכל פתרון.

שליטה בכמות המוטציות

המשתנה mutation_rate מאפשר לנו לשלוט בכמות המוטציות שמתרחשות בכל איטרציה. הגדרת mutation_rate: בתחילת האלגוריתם נבחר ערך התחלתי ל-mutation_rate. ערך זה ישפיע על הסיכוי של כל פתרון לעבור מוטציה בכל דור.

התאמת mutation_rate במהלך הריצה: במהלך ריצת האלגוריתם, ניתן לשנות את ערך mutation_rate בהתאם לצורך. לדוגמה, אם נזהה שהתכנסות האוכלוסייה מהירה מדי, נוכל להגדיל את mutation_rate כדי להוסיף גיוון ולמנוע התכנסות לפתרון מקומי. מנגד, אם המוטציות הרבות מונעות התכנסות לפתרון יציב, נוכל להקטין את mutation_rate כדי לאפשר לאלגוריתם להתכנס לפתרון טוב.

התאמת שיעור המוטציה והכנסת פתרונות רנדומליים חדשים

כדי לשמור על גיוון האוכלוסייה ולמנוע התכנסות מוקדמת לפתרונות לא מיטביים, הוספנו אלגוריתם נוסף שמבצע את הפעולות הבאות:

1. השוואת הפתרונות הטובים והגרועים ביותר: בכל דור, נבדוק את ההבדל בין הפתרון הטוב ביותר לפתרון הגרוע ביותר באוכלוסייה. אם ההבדל בין הפתרונות קטן מסף מסוים, זה עשוי להצביע על כך שהאוכלוסייה מתכנסת מהר מדי.

2. הגדלת שיעור המוטציה: אם ההבדל בין הפתרונות קטן מהסף שנקבע, נגדיל את שיעור המוטציה כדי להוסיף גיוון באוכלוסייה.

3. הוספת פתרונות רנדומליים חדשים: במידה וההבדל בין הפתרונות קטן מהסף, נכניס גם פתרונות רנדומליים חדשים לאוכלוסייה. כמות הפתרונות הרנדומליים החדשים תהיה 5% מגודל האוכלוסייה הנוכחית.

סיכום

המוטציות בתהליך הגנטי מספקות דרך לשמור על גיוון גנטי באוכלוסייה ומונעות התכנסות מוקדמת לפתרונות לא מיטביים. באמצעות החלפת שני ערכים בפרמוטציה, אנו שומרים על חוקיות הפתרון. המשתנה `mutation_rate` מאפשר שליטה מדויקת על כמות המוטציות שמתרחשות, ומאפשר התאמה דינמית של האלגוריתם בהתאם למצב האוכלוסייה. בנוסף, הוספת פתרונות רנדומליים חדשים ושינוי שיעור המוטציה בזמן הנכון מבטיחים שהאלגוריתם ישמור על איזון בין חיפוש רחב למיטוב מעמיק, וישפר את הסיכוי למצוא את הפתרון האופטימלי.

בעיית ההתכנסות המוקדמת

בעיית ההתכנסות המוקדמת היא בעיה מוכרת וחשובה באלגוריתמים גנטיים. מדובר במצב שבו האוכלוסייה מתכנסת לפתרון מקומי מיטבי אך אינו הפתרון האופטימלי הגלובלי. בעיה זו עלולה למנוע מהאלגוריתם למצוא את הפתרון הטוב ביותר ולהשאירו במצב לא אופטימלי.

למניעת הבעיה הזו, מימשנו מספר אסטרטגיות:

- שימור שונות (Diversity Maintenance)

שימור השונות באוכלוסייה הוא קריטי למניעת התכנסות מוקדמת. במהלך תהליך יצירת הילדים ובחירת הפתרונות, אנו מבטיחים שהפתרונות הדומים לא יוצגו יותר מדי פעמים. זאת על ידי שימוש בתהליך בחירה מיוחד שנקרא בחירה תחרותית (Tournament Selection), אשר מונע השתלטות של פרטים דומים על האוכלוסייה.

- בחירה תחרותית (Tournament Selection)

כאמור, בחירה תחרותית היא טכניקת בחירה המבטיחה גיוון באוכלוסייה. התהליך פועל כך: אנו מגרילים קבוצה קטנה של k פתרונות רנדומליים מהאוכלוסייה. מתוך הקבוצה הזו, נבחר את הפתרון בעל הציון הגבוה ביותר. היות ו- k קטן ביחס לגודל האוכלוסייה (לדוגמה - 5 מתוך 100), התהליך מגדיל את הסיכוי לשימור גיוון גנטי ומקטין את הסיכוי שהפרטים הטובים ביותר ישתלטו על מרחב הפתרונות מוקדם מדי.

- רמת מוטציה משתנה (Adaptive Mutation Rate)

רמת המוטציה המשתנה היא טכניקה שמטרתה להציג מידע גנטי חדש לאוכלוסייה לאורך זמן: בתחילת תהליך הריצה של האלגוריתם, נבחר ערך התחלתי נמוך ל-`mutation_rate` כדי לאפשר לאוכלוסייה הראשונית להתקדם לקראת פתרונות טובים.

ככל שהאלגוריתם מתקדם ומספר האיטרציות עולה, נעלה בהדרגה את רמת המוטציה. כמו כן, אם ההפרש בין הפתרון הטוב ביותר לפתרון הגרוע ביותר באוכלוסייה קטן מסף מסוים, נעלה את רמת המוטציה. הדבר נעשה כדי להכניס גיוון גנטי חדש ולמנוע מהאוכלוסייה להתכנס מוקדם מדי לפתרונות מקומיים. העלאת רמת המוטציה מונעת מהאלגוריתם להיתקע במקסימום מקומי ומאפשרת חיפוש רחב יותר במרחב הפתרונות.

- הוספת פתרונות רנדומליים חדשים (Random Repopulation)

במידה וההפרש בין הפתרון הטוב ביותר לפתרון הגרוע ביותר באוכלוסייה קטן מסף מסוים, נוסף 5% של פתרונות רנדומליים חדשים לאוכלוסייה. הפתרונות החדשים "ירעננו" את האוכלוסייה ויעזרו להימנע מהתכנסות מוקדמת כי הכנסנו גיוון לאוכלוסייה.

שילוב האסטרטגיות

האלגוריתם שלנו משלב את שלוש האסטרטגיות הנ"ל במטרה להבטיח הימנעות מהתכנסות מוקדמת והישארות במקסימום מקומי. כל אחת מהאסטרטגיות פועלת בדרכה להבטיח גיוון באוכלוסייה:

- שימור השונות מבטיח שהאוכלוסייה תישאר מגוונת.
- בחירה תחרותית מצמצמת את הסיכוי להשתלטות פתרונות דומים.
- רמת מוטציה משתנה מכניסה מידע גנטי חדש באופן מתמשך.
- הגרלת פתרונות חדשים מבטיחה גיוון באוכלוסייה.

על ידי שילוב אסטרטגיות אלו, אנו מבטיחים שהאלגוריתם ישמור על איזון בין חיפוש רחב ומעמיק במרחב הפתרונות, ויסתגל לשינויים באוכלוסייה בצורה אופטימלית.

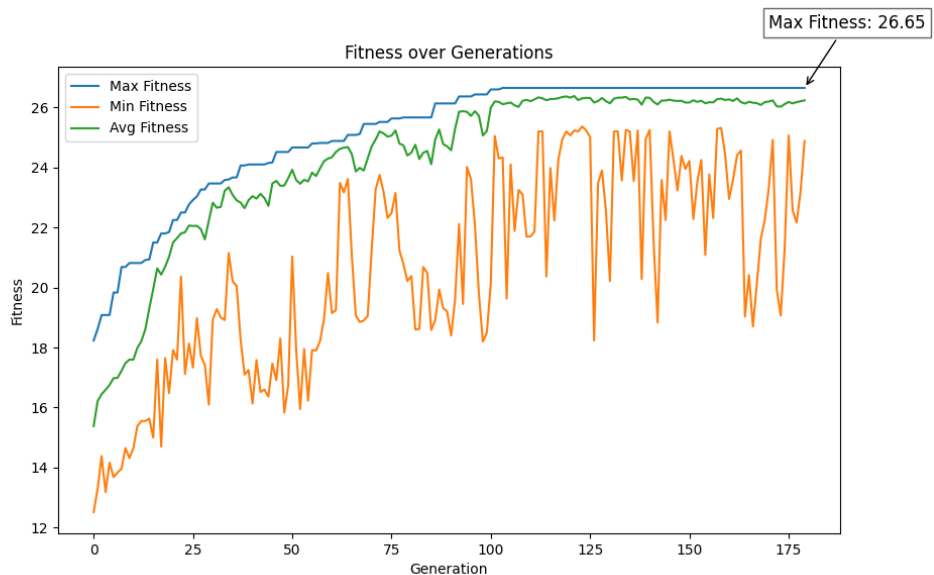
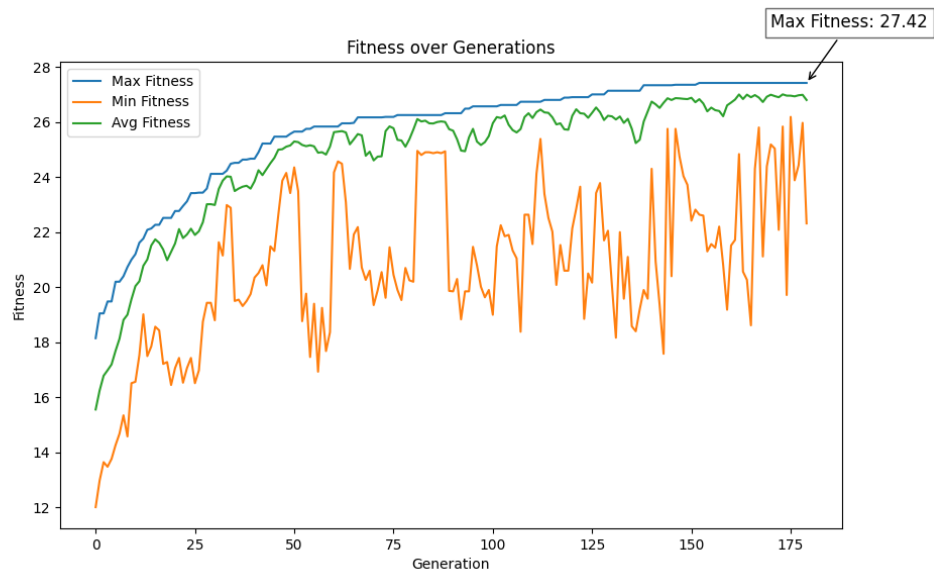
סיכום

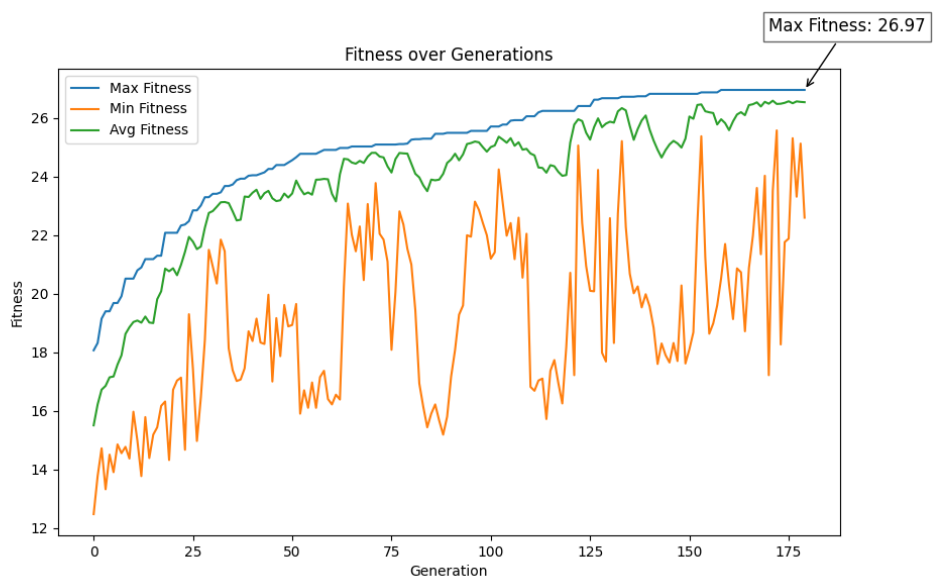
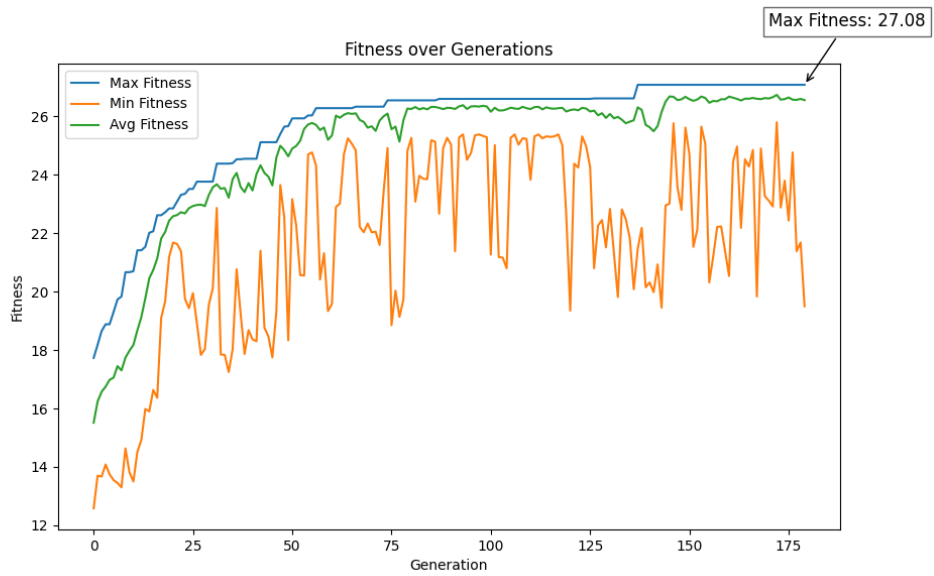
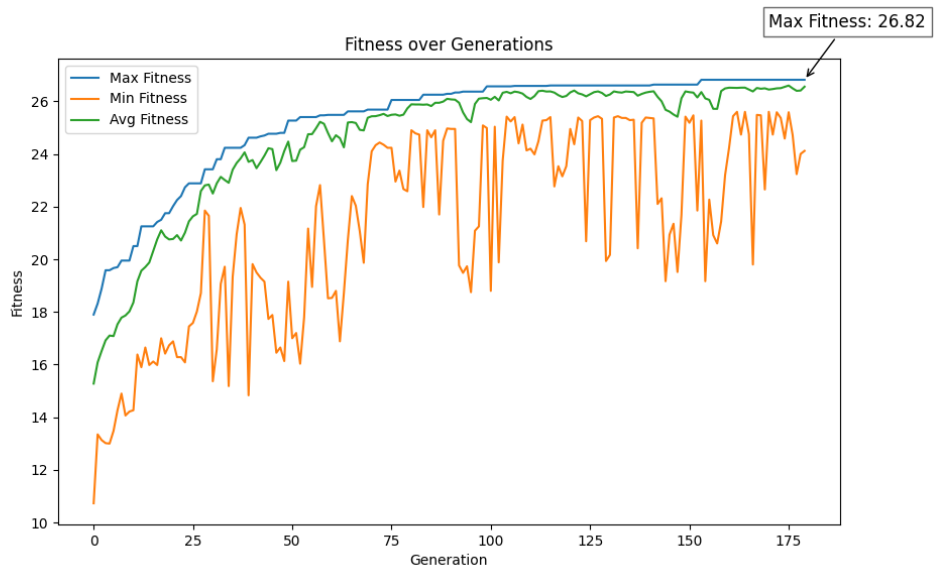
האלגוריתם שלנו משתמש בשימור שונות, בחירה תחרותית, התאמת שיעור המוטציה והגרלת פתרונות חדשים כדי להבטיח שהאוכלוסייה תישאר מגוונת ושנמנע מהתכנסות מוקדמת לפתרונות מקומיים. על ידי שילוב האסטרטגיות הללו, האלגוריתם שלנו שומר על איזון בין חיפוש רחב ומעמיק במרחב הפתרונות, ומשפר את הסיכוי למצוא את הפתרון האופטימלי.

הרצות שונות ותוצאותיהן

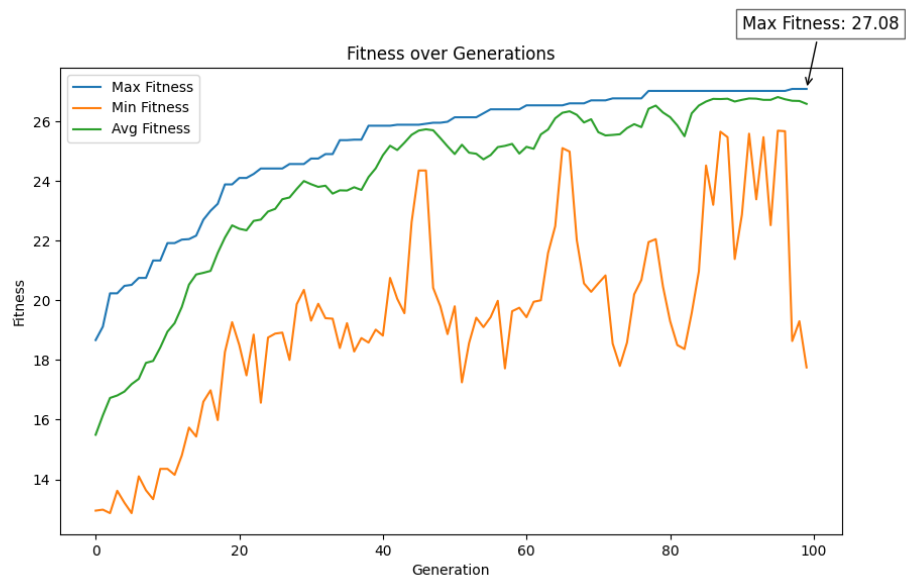
בחלק זה נציג גרפים של הרצות שונות של האלגוריתם הגנטי. בכל פעם נריץ עם ערך $population$ ו $generations$ שונה, כך שהכפל ביניהם יצא: 18,000.

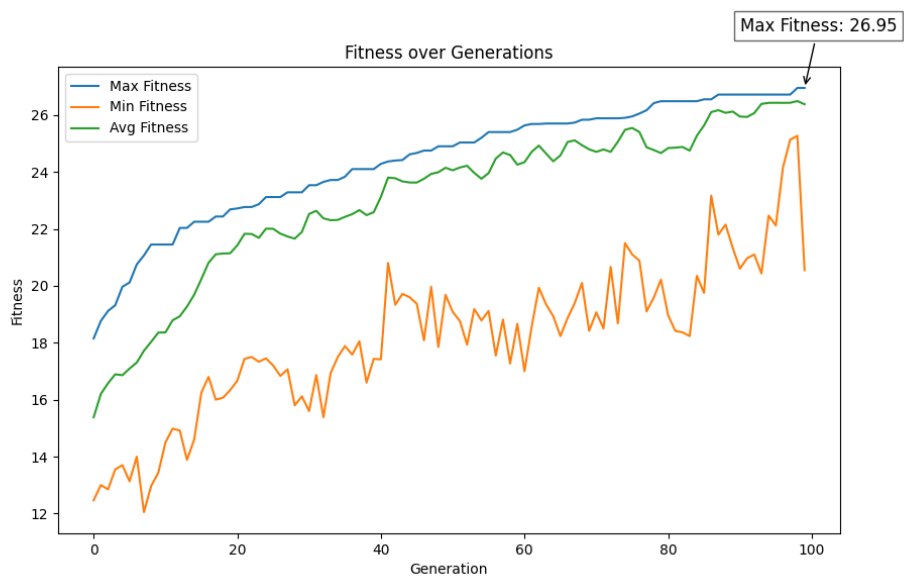
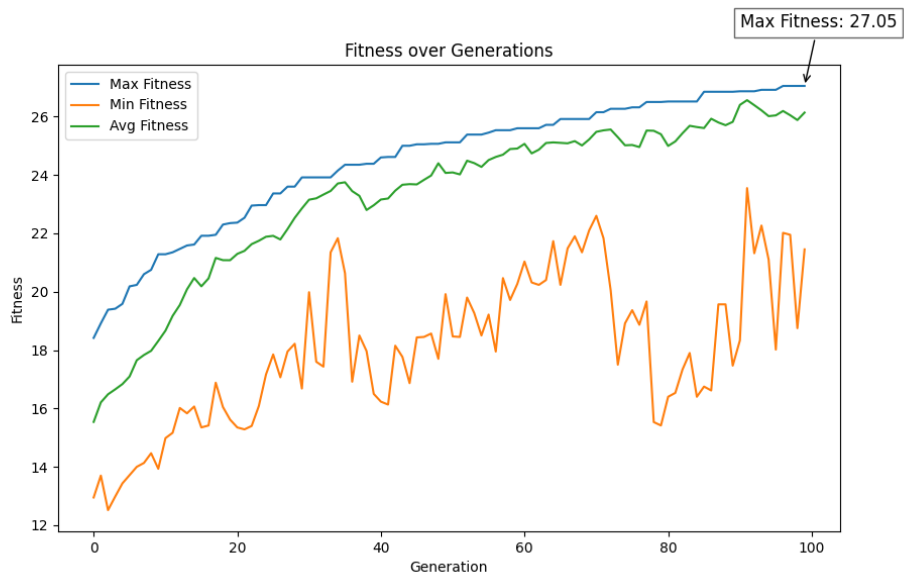
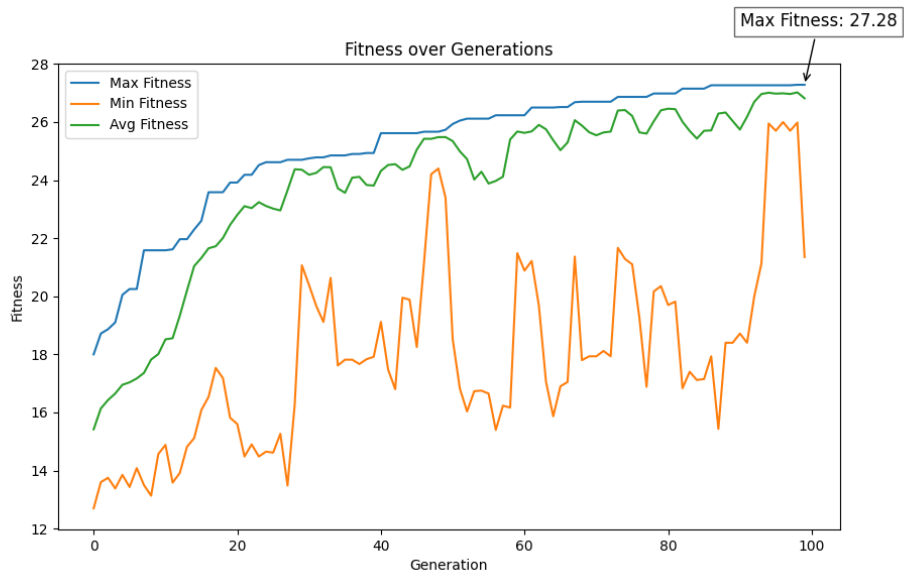
- ניסיון ראשון: $population = 100$, $generations = 180$



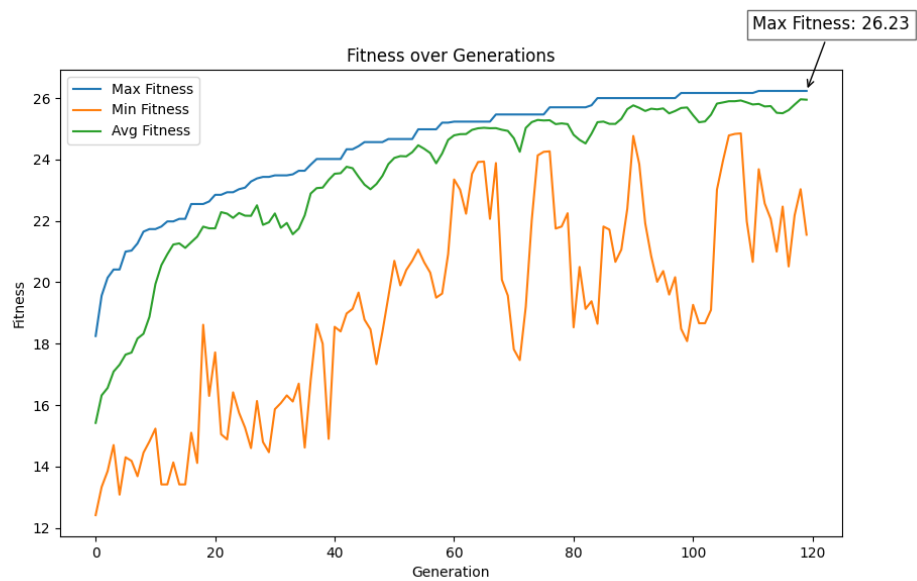
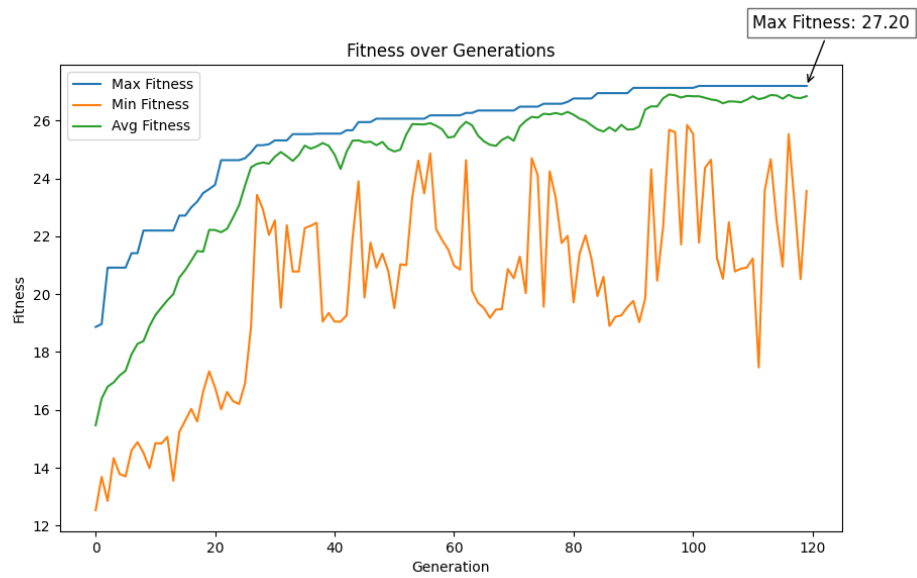


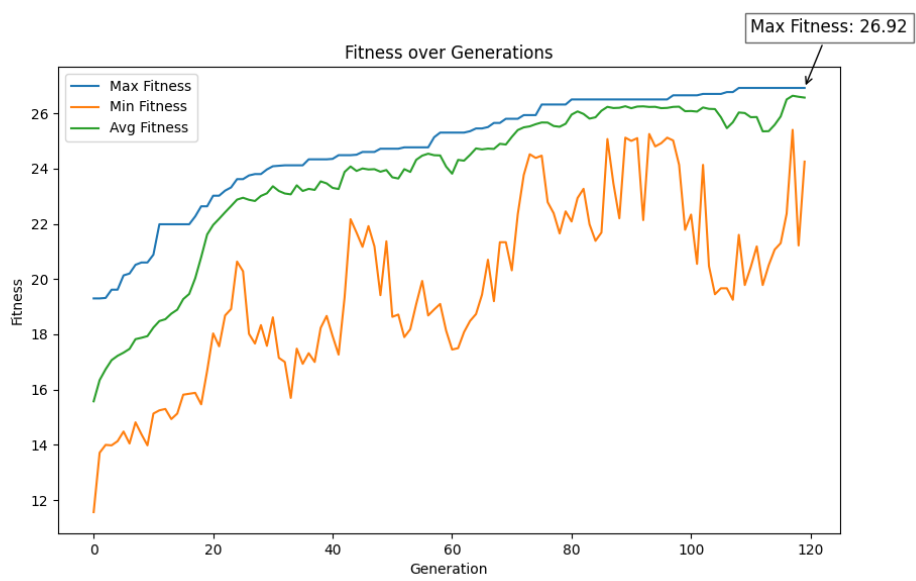
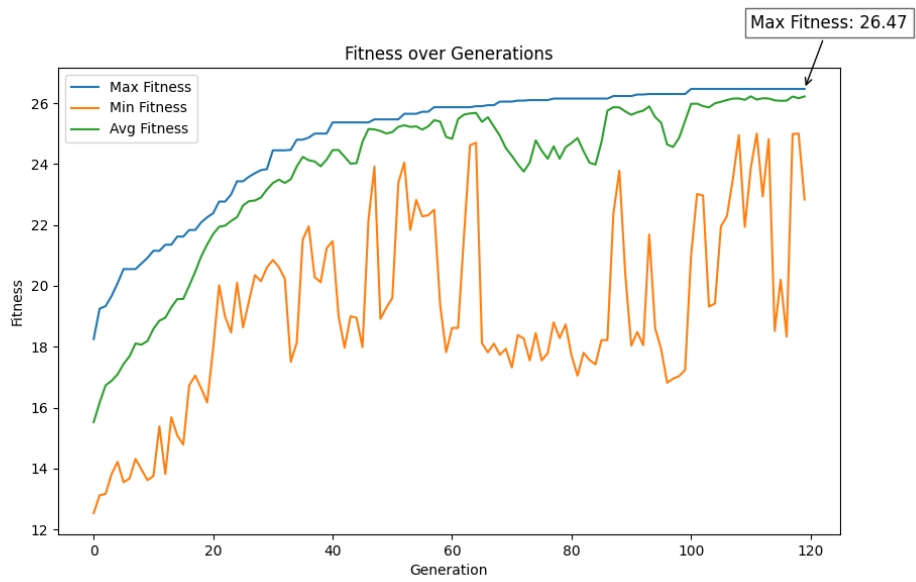
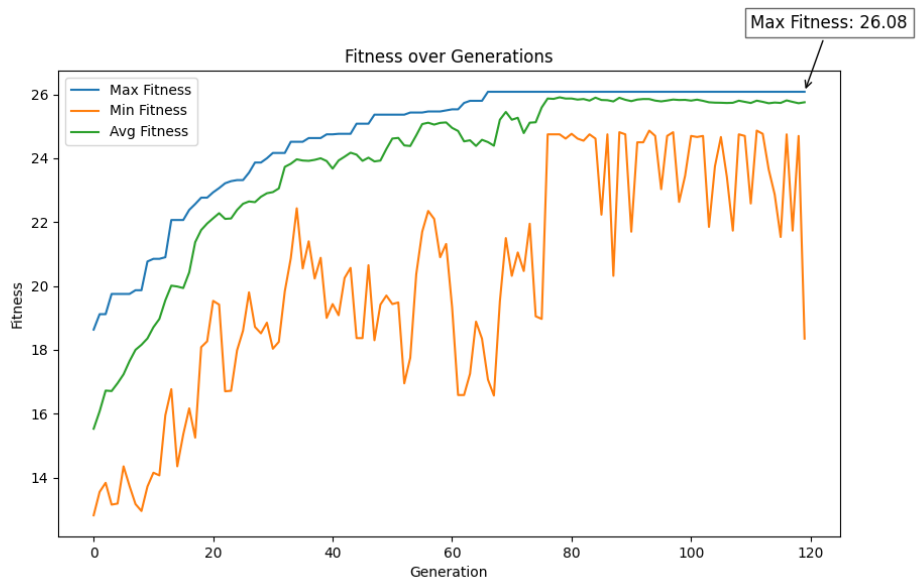
- ניסיון שני: $population = 180, generations = 100$





- ניסיון שלישי: $population = 150$, $generations = 120$





מסקנות

לאחר התבוננות בגרפים השונים, ניתן לראות בבירור כי הצירוף $population = 180$, $generations = 100$ הניב את התוצאות הטובות ביותר.

התוצאות שערכים אלו הניבו היו עקביות יותר משאר התוצאות בצירופים השונים, וגם היו גבוהות יותר מערכים אלו.

הממוצעים השונים ואחוזי ההצלחה:

- ניסיון ראשון – ממוצע 26.984, אחוזי הצלחה 89.9%.
- ניסיון שני – ממוצע 27.116, אחוזי הצלחה 90%.
- ניסיון שלישי – ממוצע 26.58, אחוזי הצלחה 88.6%.

כפי שטענו, אחוזי ההצלחה והממוצע היו גבוהים יותר בניסיון השני. כאשר האוכלוסייה בה השתמשנו הייתה גדולה, קיבלנו תוצאות טובות יותר למרות מספר הדורות הנמוך.

מה שחשוב לראות זה שכל הניסויים הניבו אחוזי הצלחה גבוהים מאוד. בהתחשב בעובדה שאנו לא יודעים מהו הפתרון האופטימלי והאם קיים בכלל פתרון חוקי שהוא "מושלם" (כלומר שכולם מזווגים לבחירה הראשונה שלהם), פתרון עם אחוזי הצלחה של 88-90% הוא מוצלח במיוחד.

סיכום

בפרויקט זה התמקדנו בפתרון בעיית שידוכים מקסימלית באמצעות אלגוריתמים גנטיים. בעיית השידוכים כללה 30 גברים ו-30 נשים, כאשר כל משתתף דירג את העדפותיו לגבי בני המין השני. מטרתנו הייתה למצוא התאמה אופטימלית שתמקסם את שביעות הרצון הכוללת של כל המשתתפים.

התחלנו את האלגוריתם עם m פתרונות רנדומליים והפעלנו את האלגוריתם הגנטי במשך n דורות, כאשר המכפלה של n ו- m הינה 18,000. כל פתרון ייצג פרמוטציה של המספרים 1-30, שם כל מספר ייצג אישה והמיקום ברשימה ייצג את הגבר ששודך לה.

פונקציית ההערכה חישה את שביעות הרצון של כל זוג לפי דירוגי ההעדפות של הגברים והנשים, כאשר הציונים נעו בין 1 ל-30. הערך המתקבל היה ממוצע הציונים של כל הזוגות.

הכלאה בין פתרונות (crossover) התבצעה על ידי בחירת שני הורים בעזרת בחירת תחרות, יצירת ילד על ידי העתקת קטע מהורה אחד והשלמת הערכים מההורה השני. תהליך זה שמר על חוקיות הפתרון בכך שכל ערך הופיע פעם אחת בלבד.

בגרפים שנוצרו ניתן לראות את התקדמות הפתרונות לאורך הדורות, תוך שמירה על נתונים כמו ערך התאמה מקסימלי, מינימלי וממוצע. הצלחנו להימנע מהתכנסות מוקדמת בעזרת בחירת תחרות ושימוש במוטציות להוספת גיוון לאוכלוסייה.

הניסויים הראו שהצירוף $population=180, generations=100$ הניב את התוצאות הטובות ביותר עם אחוזי הצלחה גבוהים במיוחד, כאשר כל הניסויים הניבו אחוזי הצלחה של 88-90%. זה מראה שהשיטה הצליחה למצוא פתרונות מוצלחים מאוד, בהתחשב בעובדה שאיננו יודעים אם קיים פתרון מושלם שבו כל המשתתפים מזווגים לבחירה הראשונה שלהם.

לסיכום, הצלחנו להראות כי באמצעות אלגוריתם גנטי ניתן למצוא קירוב ברמת דיוק גבוהה מאוד לבעיית השידוכים עם העדפות. בעזרת מימוש טכניקות שונות ושילוב של אופרטורים מגוונים האלגוריתם נמנע מהתכנסות מוקדמת ושמר על גיוון באוכלוסייה.