**AI LAB 1-2**

פייסל סעדיה 208336321

אחמד גבארין 314722307

1) הוספנו קובץ parents selection שבו מימשנו את הפונקציות .

a)



b)

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

C)

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

2) הוספנו שדה חדש ל- gene.py שזה age .

כך שב-elitism נותנים פחות סיכויים לגינומים שמזדקנים.

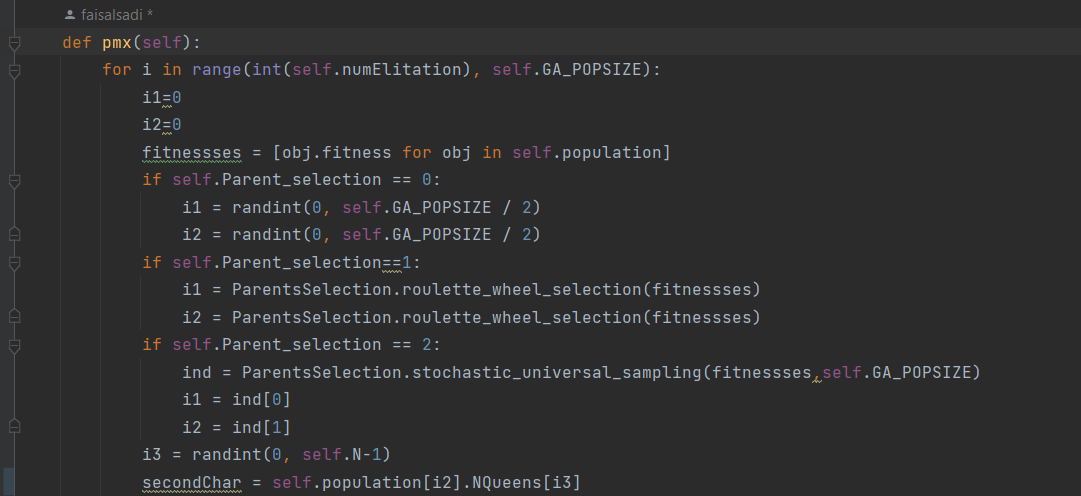
3)

a) ייצוג :מחרוזת באורך N תייצג לוח באורך N\*N כך שבעזרת האינדקס והמספר שנמצא בתוך האינדקס נקבל את הקורדינאטה של המלכה ה i למשל :

[1,2,3,4,5,6,7] המלכה הראשונה נמצאת ב (1,1) השנייה ב (2,2) וכן הלאה ..

b)

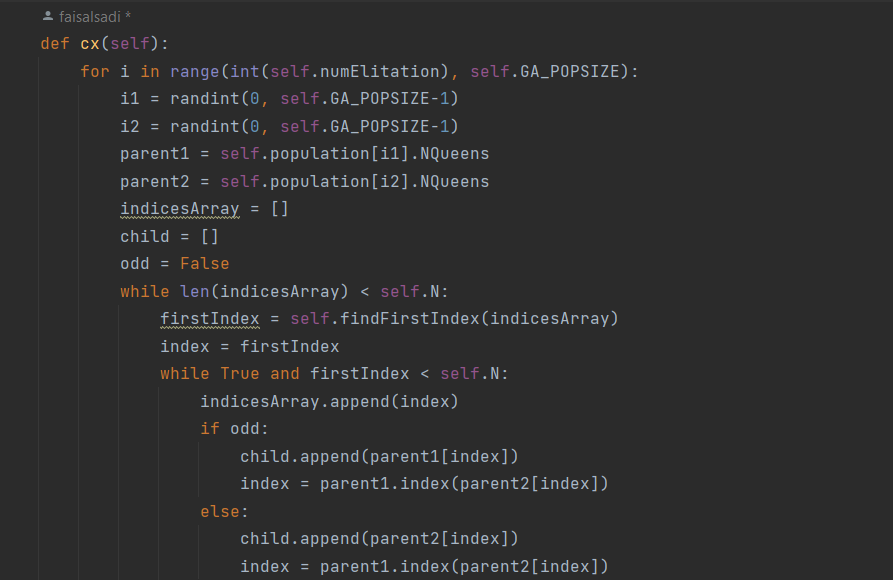
Pmx : מגרילים אינדיקס ואז מחליפים את האותיות בהורים



תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

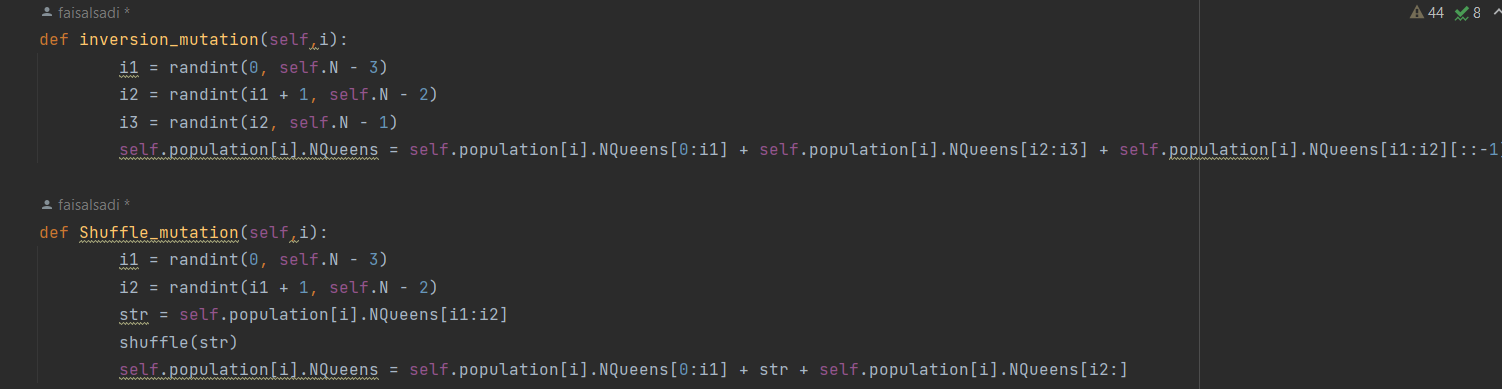
CX :



תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

C )



D )

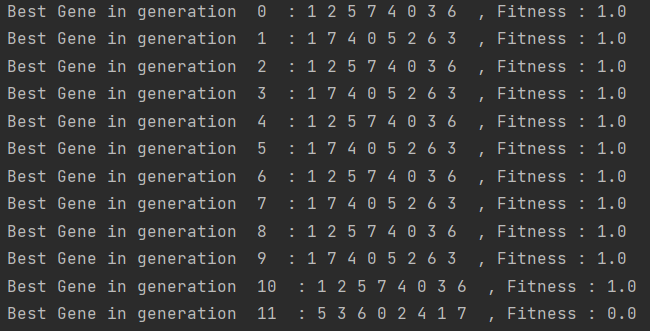
מימשנו פונקציית פיטנס שמסתמכת על מספר ההתנגשיות בפרמוטציה מסויימת כך שאם יש הרבה התנגשויות אז יש מקבלים קנס יותר גבוה , עוברים על מערך המלכות ומחשבים את מספר ההתנגשויות ( מחלקים ב 2 בגלל הסימטריה " מלכה I מאיימת על J זה אותו דבר כמו שמלכה J תאיים על I " )



תמונה שמכילה טקסט

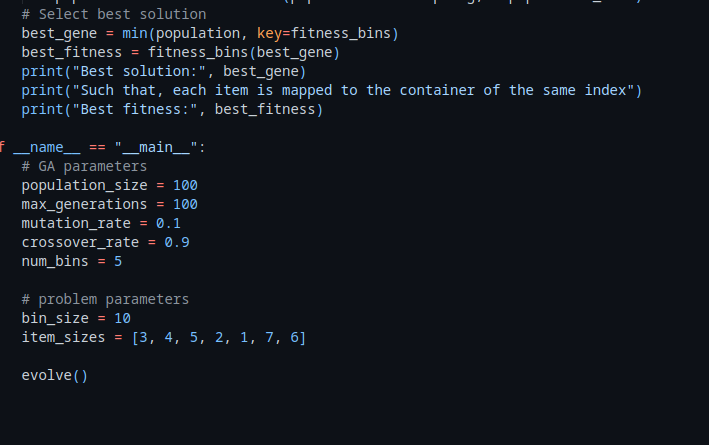
התיאור נוצר באופן אוטומטי

ודוגמה להרצה עבור שיטת שחלוף PMX ומוטציית ערבול ובשיטת בחירת הורים Naïve Parent Selection

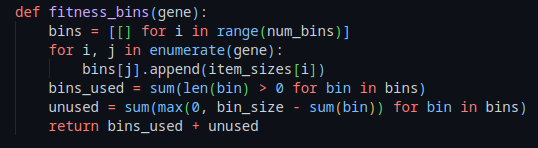


4) הוספנו קובץ BinPackaging.py שמטפל בבעית ה- Bin Packaging.





1. פונקצית הפיטנס שמימשנו :

היא לוקחת גינום כקלט אשר מיוצר על ידי הפונקציה :

כך שכול גינום הוא מהפורמט הבא:

למשל: [4, 3, 2, 2, 0, 3, 0]

החפץ באינדקס 0 הולך למכולת בעלת אינדקס 0

החפץ באינדקס 1 הולך למכולת בעלת אינדקס 3

החפץ באינדקס 3 הולך למכולת בעלת אינדקס 2

החפץ באינדקס 4 הולך למכולת בעלת אינדקס 2

וכו

הפונקציה אחר כך מסדרת אותם לתוך bins

**הערך של הפיטנס:** אנחנו מעוניינים בגינומים שמהווים פתרון כך שבו השתמשנו בפחות מכולות כך שכל אחת מהם תכיל כמה שיותר חפצים, לכן רוצים למקסם את הערך של bins\_used + unused.

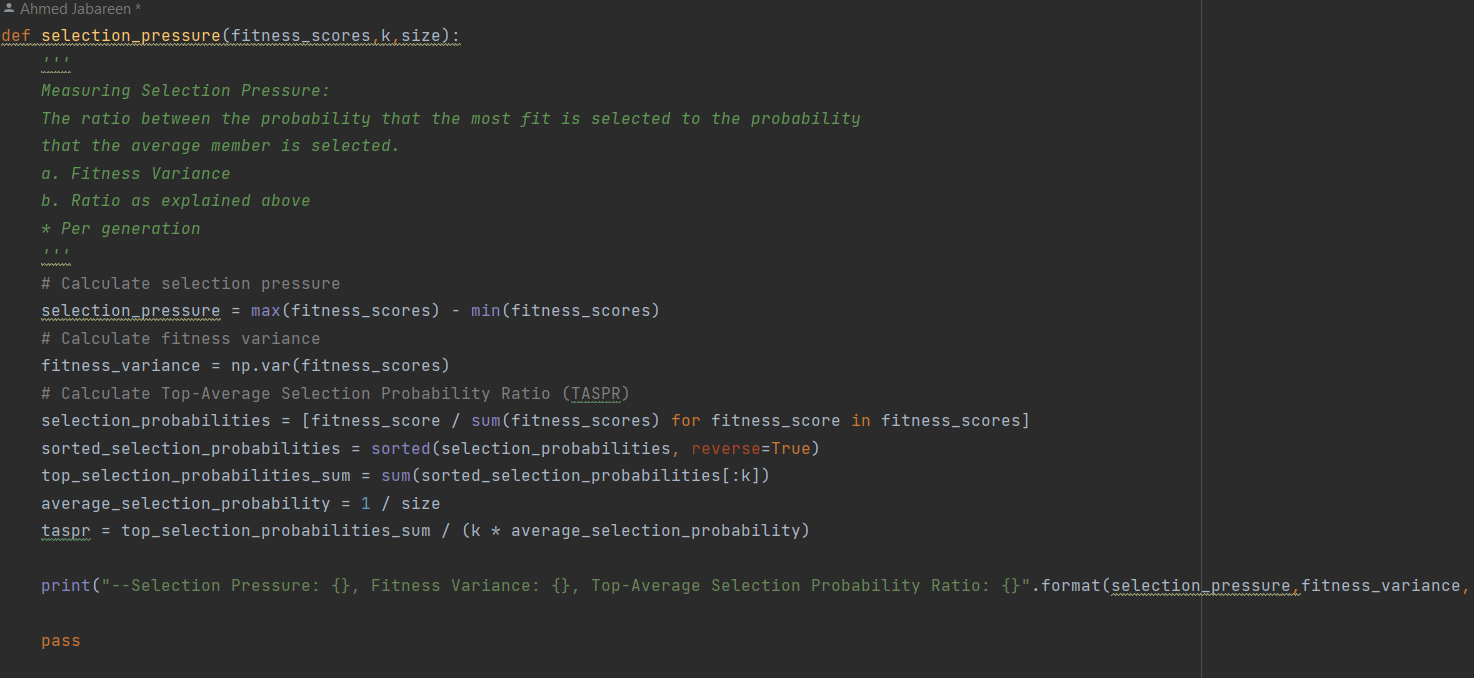
1. האלגוריתם החמדני רץ באופן מהיר יותר כתוצאה מכך שהוא לא עושה חישובים מסובכים ואינו משתמש בהרבה משאבים.

האלגוריתם הגינטי לוקח לו קצת יותר זמן (בסביבות 150-200 מל שניות) אבל נותן תוצאות יותר קרובות לפתרון האופטמלי.

5.

אנו מציגים 3 מדדים : טווח ערכי הפיטנס , שונות הפיטנס ו Top-Average Selection Probability Ratio

החישוב נעשה בפונקציה הזאת שמשתמשים בה בכל הבעיות הקודמות



6.

אנו מציגים 2 מדדים : המרחק של הגנים ומספר האללים השונים

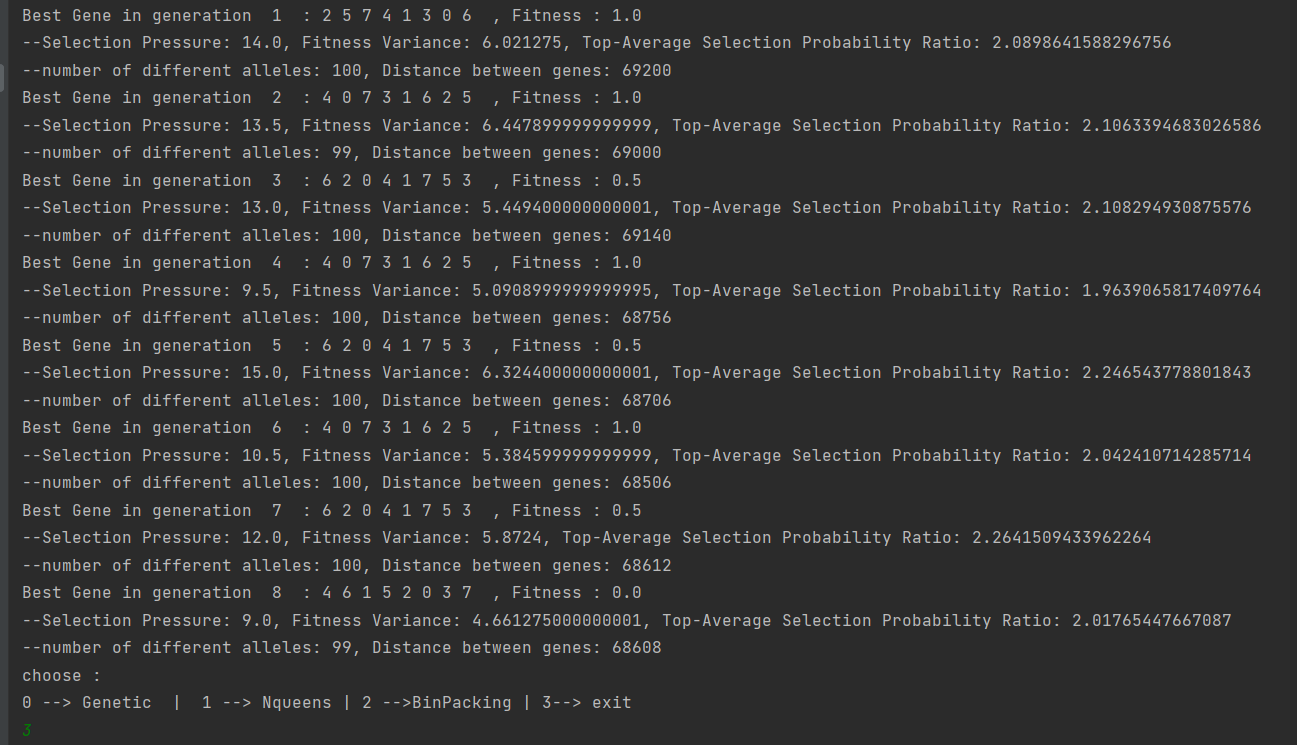
החישוב נעשה בפונקציה הזאת שמשתמשים בה בכל הבעיות הקודמות

פונקציית המרחק הייתה hamming distance שבה אנחנו מחזירים את מספר האינדיקסים שבהם שני גינים אינם תואמים

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

הרצה והצגת הדיווח בכל דור עבור הסעיפים 5 ו 6 ( עבור בעיית המלכות ):



7.

a)

גודל האוכלוסיה: עבור שתי הבעיות היינו מקבלים פחות דורות אבל זמן ריצה יותר גדול עם הגדלת גודל האוכלוסיה

b)

הסתברות למוטציה: בבעית N המלכות אם מגדילים הסתברות למוטציה אז מספר הדורות וזמן הריצה שניהם משתפרים תמיד אבל binpacking כן השתפר ברוב המקרים אבל לפעמים זמן הריצה היה יותר גרוע בפחות משנייה אבל לרוב השתפר

c)

אסטרטגיית הבחירה בשני המקרים הטורניר זה הכי מהר וטוב שהיה

d)

אסטרטגיית השרידות:לפי AGING יותר טוב בשני המקרים

e)

ל N המלכות CX היה יותר טוב ול binpacking שיטת ה Uniform הניבה תוצאות יותר טובות.

8)

עבור הבעיות הפרמטרים הכי טובים היו

Pop size=200

Elite rate=0.1

Mutation rate=0.3

) a

בדקנו עבור כל פרמטר את ההשפעה שלו על הגיוון והלחץ ושאפנו לכמה שיותר איזון בין הגיוון והלחץ וזה מה שהוביל אותנו לפרמטרזציה המיטבית

b)

שמירה על איזון בין גיוון ולחץ : כדי להגיע לתוצאות הכי טובות הפרמטרים שאנחנו מעבירים צריכים לשמור על האיזון , כלומר שלא יהיה גיוון גבוה מידי ואז אנחנו נהרוס מועמדים טובים לדור הבא ובאותו אופן שלא יהיה לחץ בחירה גבוה מאוד ואז נתקעים באופטימום לוקאלי למשך זמן רב.