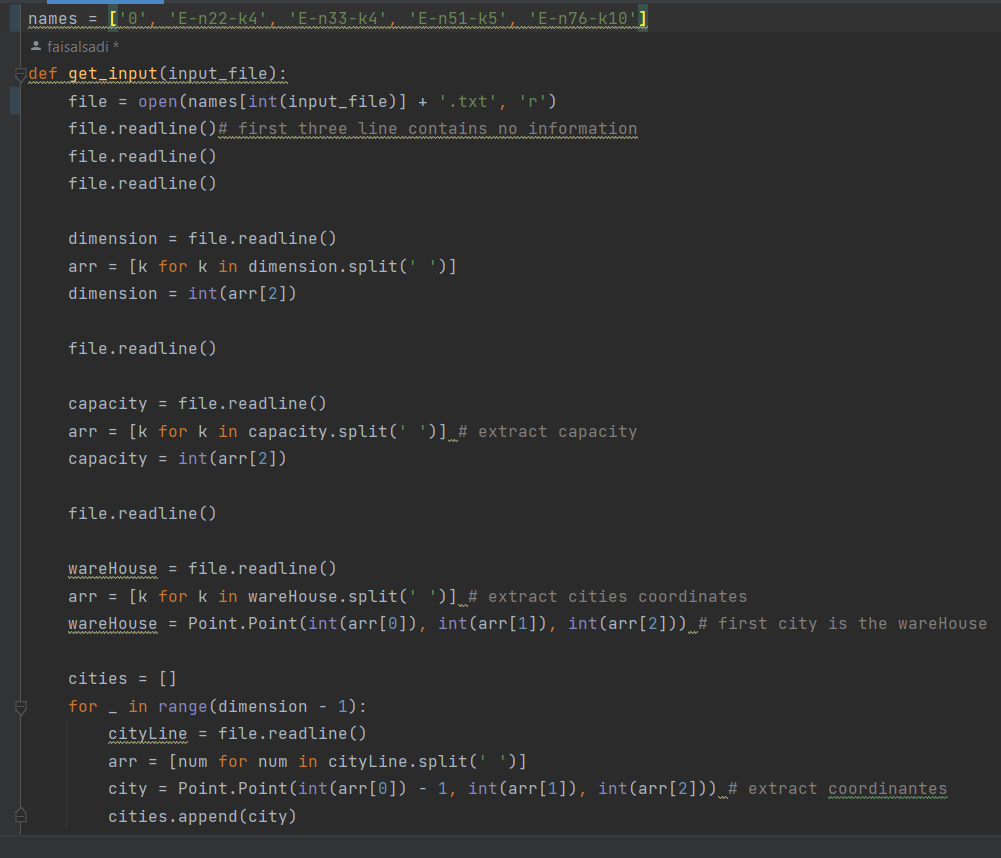
**AI LAB 3**

פייסל סעדיה 208336321

אחמד גבארין 314722307

שאלה 1:

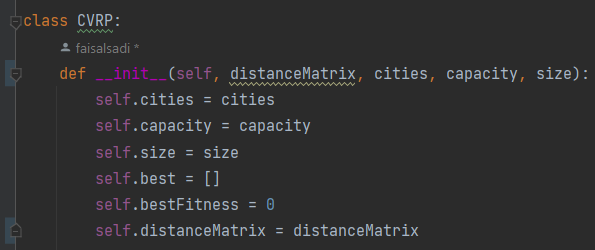
אנחנו חילצנו את הפרמטרים שאנחנו צריכים מקבצי ה txt הנמצאים באתר ע"י קריאת השורות ואז לחלץ מכל שורה את הפרמטרים הרלוונטים לפי הפורמט של כל קבצי ה txt שיש באתר , קודם כל המימד אחר כך ה capacity ולאחר מכן הקורדינאטות וכן הלאה , הקוד שבו אנו מחלצים את הפרמטרים הוא כך :



שאלה 2:

לצורך תיאום הבעיה לאלגוריתמים השונים בנינו אובייקט שמתאר את הבעיה קראנו לו CVRP אובייקט זה מכיל את כל המידע הנדרש : מקומות הערים , מטריצת המרחקים ביו כל 2 ערים , capacity , demand ועוד.....

הינה הגדרת ה Class :

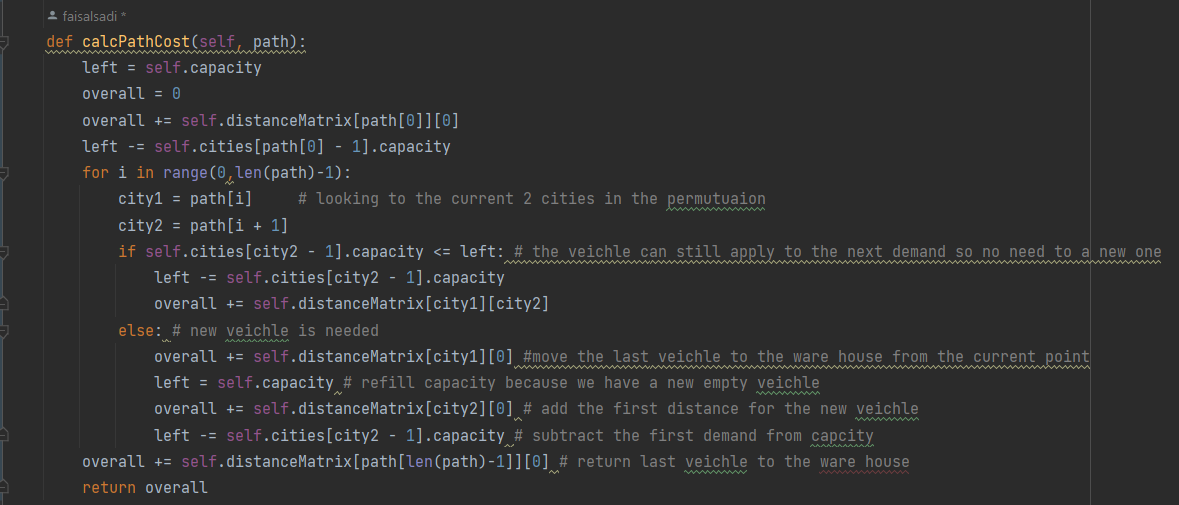


לאובייקט זה הוספנו גם מתודות שמחשבות את העלות של פתרון מסויים שייצגנו אותו בתור מערך ומחלצים ממנו את הסכום ומספר המכוניות הנדרשות כדי להשלים את המסלול

למשל עבור הבעיה בתרגיל המערך הזה [1,2,3,4] אז חישוב העלות עבור מערך זה יחזיר לנו 80.6 ו 2 רכבים , כאשר רוצים להדפיס את הפתרון אז עוברים על המערך משמאל לימין ומחסירים את ה demand מ capacity עד שנגיע למצב שבו אנחנו לא יכולים לספק את העיר הנוכחי ואז מוסיפים רכב חדש שממשיך את המסלול מהנקודה האחרונה .

אנחנו הסתכלנו בכל פעם על השכנים של פתרון מסויים ( פרמוטציה ) ובוחרים את השכן שמקטין את אורך המסלול

מימוש :



לאורך כל האיטרציות אנו שומרים את הפתרון הטוב ביותר ואז בסוף מחלצים את המסלולים באותה שיטה שבא אנחנו חישבנו את העלות של כל המסלולים בפרמוטציה מסויימת.

שאלה 3:

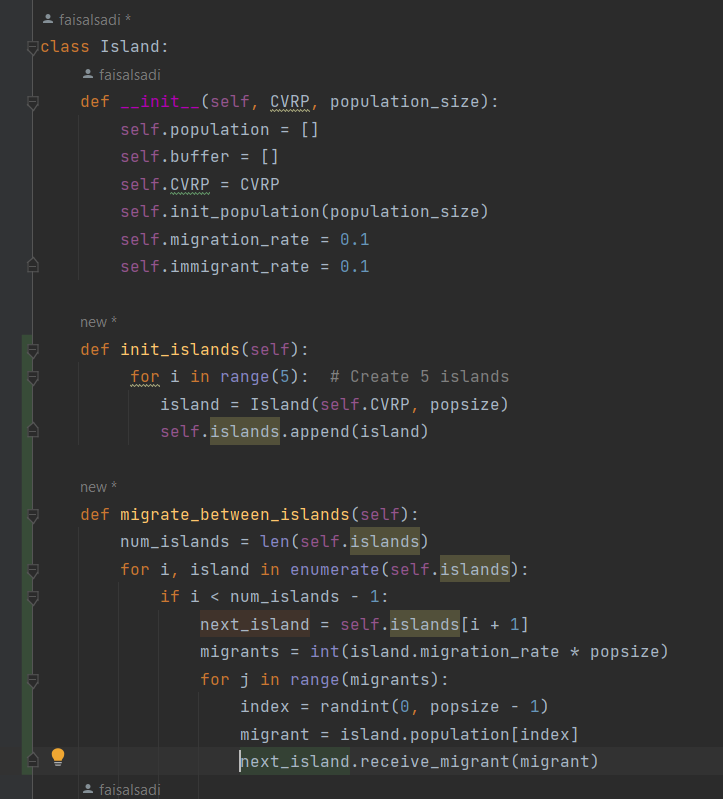
כמו שנאמר בסעיף הקודם אנו מסתכלים על סדרת ערים מסוימת ( פרמוטציה של ערים כמו שהיה לנו בבעיית ה N מלכות ) מייצגים אותה בתור מערך ואז מפעילים את הטכניקות שלמדנו במעבדות קודמות כמו שחלוף ומוטציות ועוד כדי להסתכל על השכנים של פתרון נוכחי.

מבצעים את החיפוש עד שמחזירים את הפתרון עם הכי פחות עלות , נעזרים במטריצת המרחקים שאנו בונים באתחול הבעיה ובכל פעם עבור משאית מסויימת אנחנו מתקדמים לעיר הכי קרובה לפי מטריצת המרחקים TSP.

שאלה 4 :

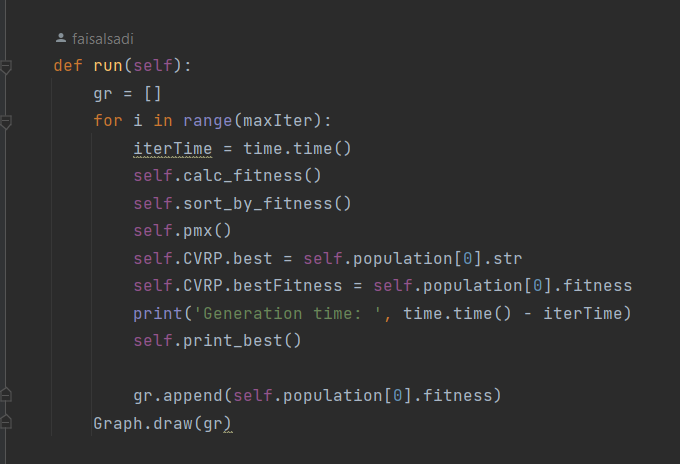
מימוש האלגוריתמים :

GA with island Model :

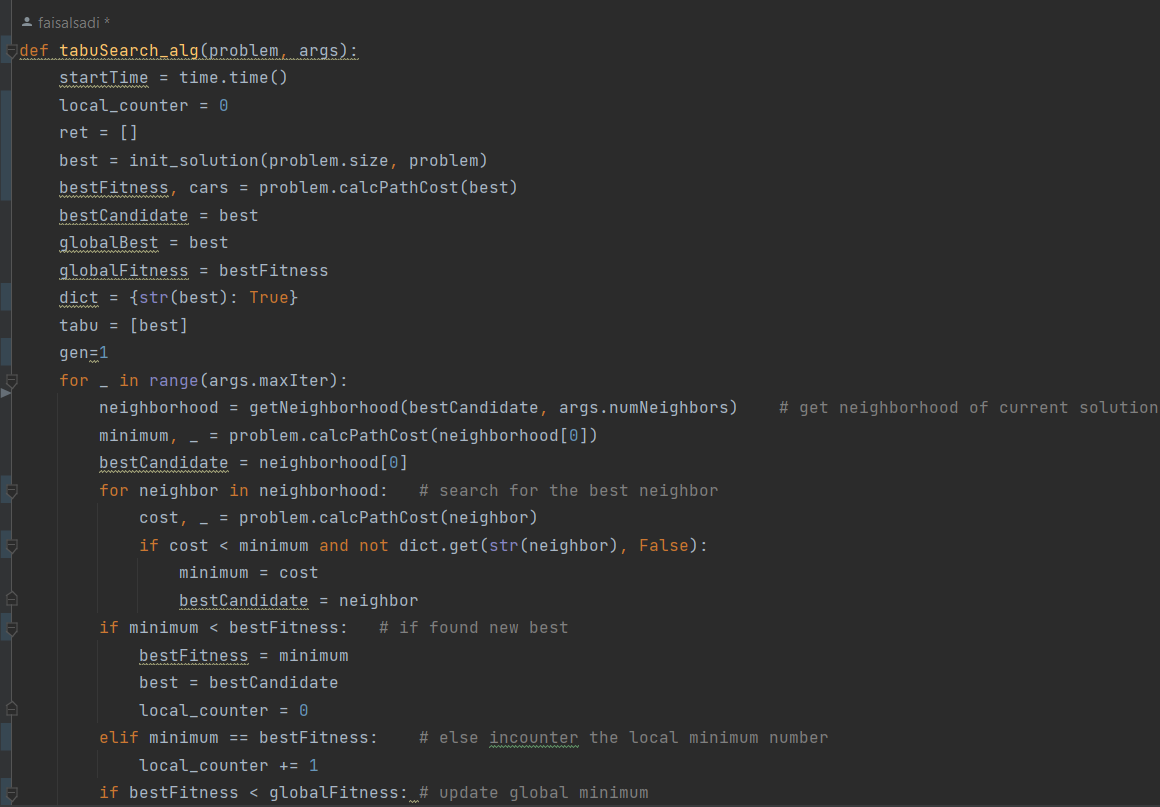


תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי



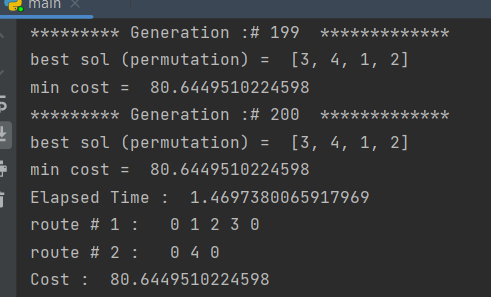
Tabu Search :



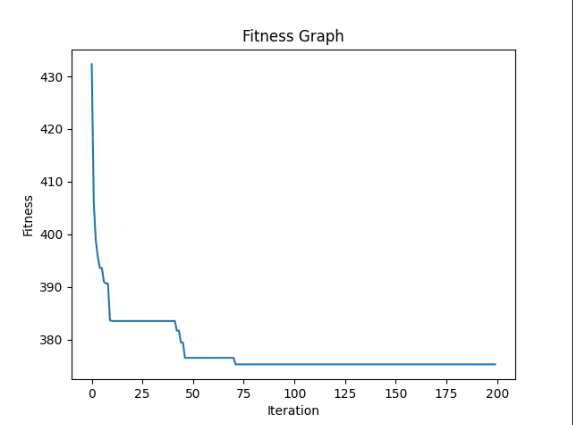
תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

דוגמת הרצה : עבור הבעיה בתרגיל



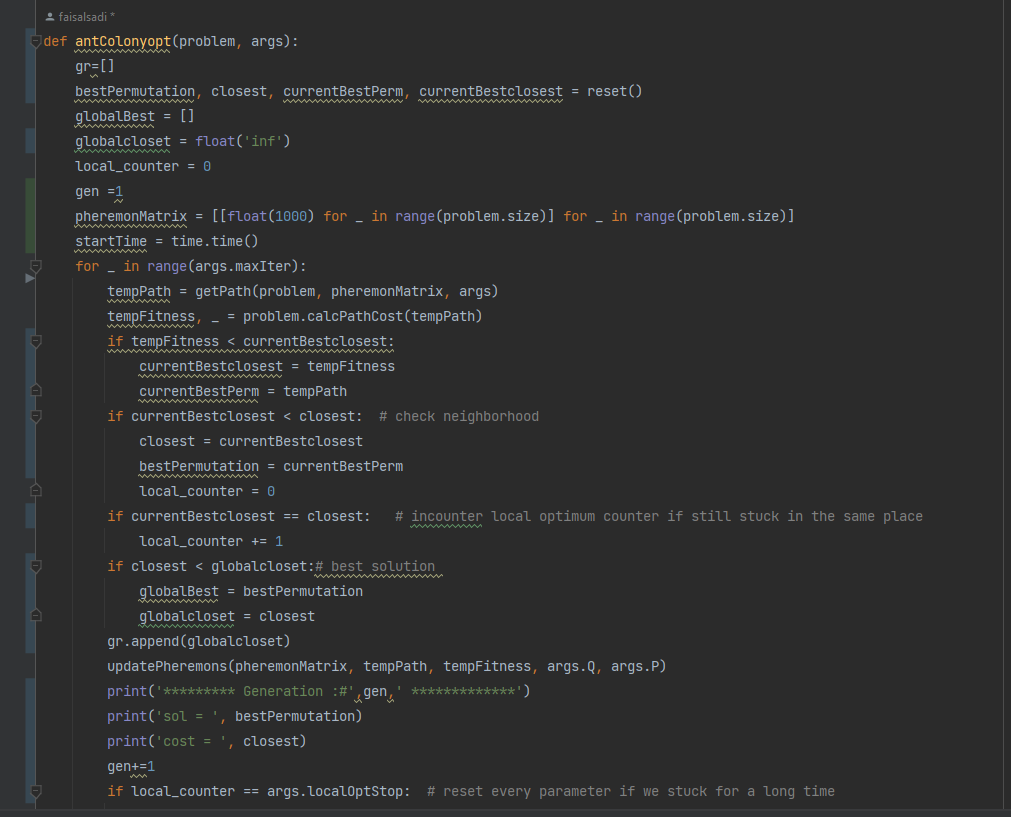
* גרף שיפור הפיטנס לבעיה הראשונה באתר [E-n22-k4](https://www.dca.fee.unicamp.br/projects/infobiosys/vrp/E-n22-k4.txt) :



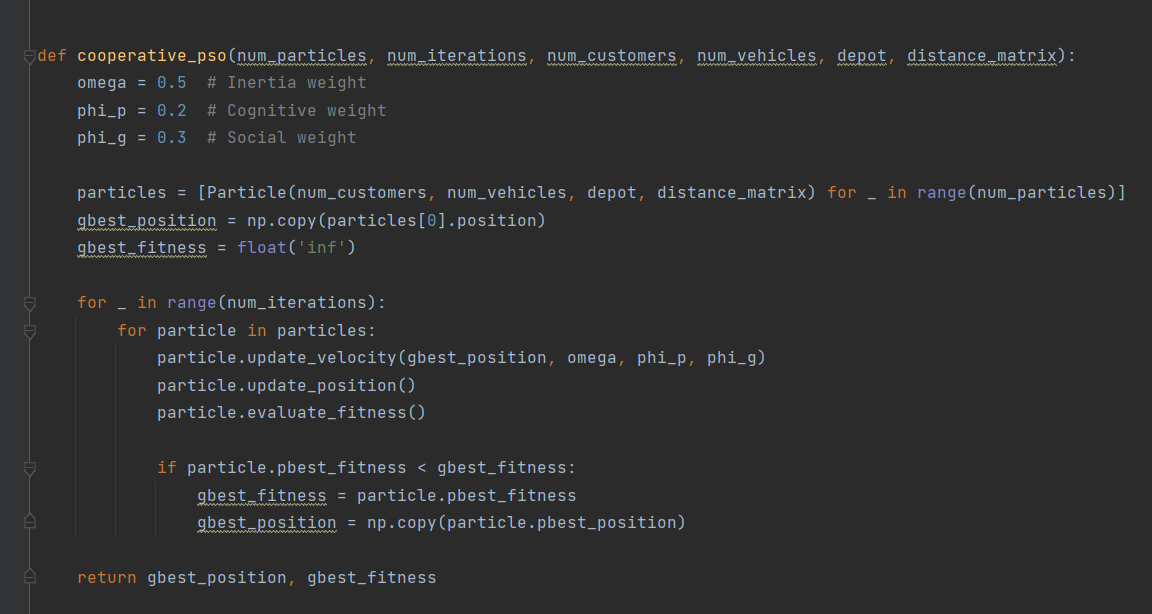
תמונה שמכילה טקסט, גופן, צילום מסך, מספר

התיאור נוצר באופן אוטומטי

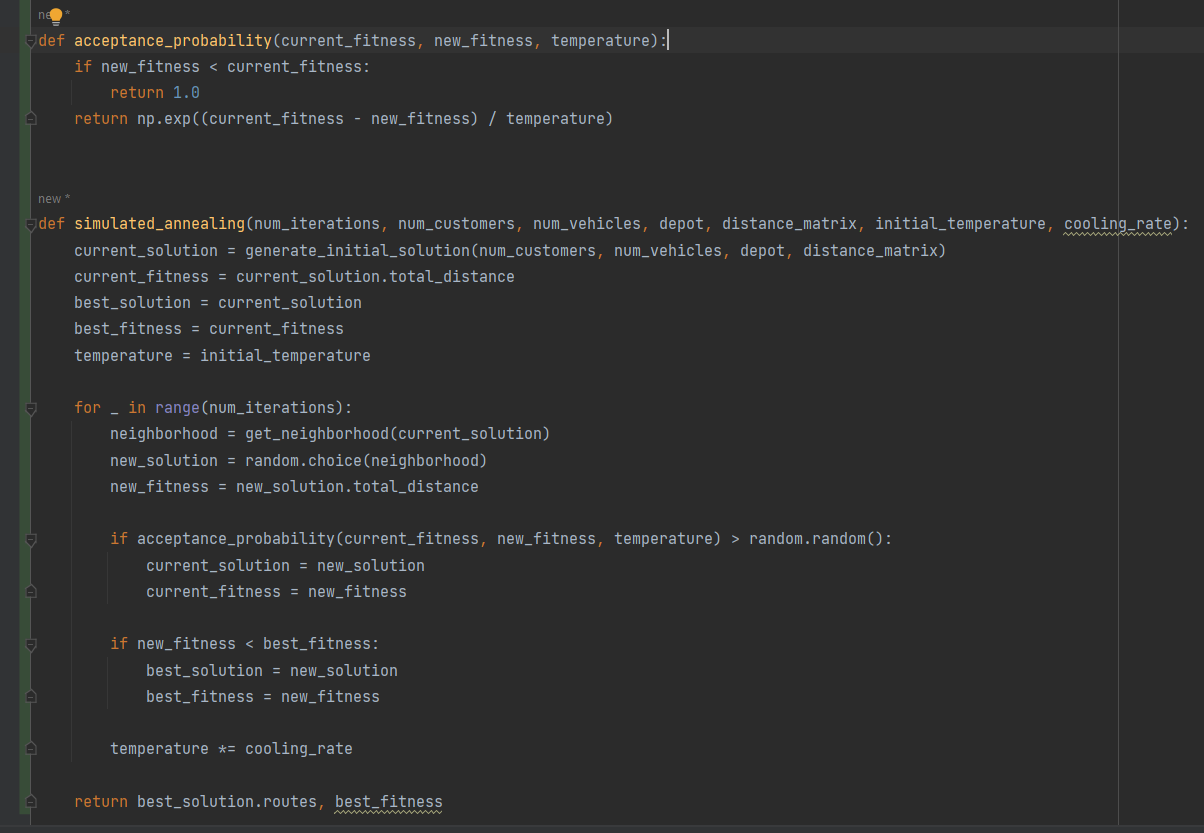
Ant Colony :



Cooperative Pso:



Simulated Annealing :



שאלה 5:

כיוון שאנו מסתכלים על השכנים של כל פתרון שמקטינים את אורך הסמלול הכולל אזי אחרי מספר של איטרציות אנחנו יכולים לשאוף לאורך האופטימלי הנדרש , כיוון שההיוריסטיקה משפרת את אורך המסלול באופן איטרטיבי וזה מתאים לאופי של בעיית ה CVRP .

לגבי יעילות ואיכות פתרון שמנו לב ש Tabu Search הניב את התוצאות הכי טובות מבחינת זמן ומבחינת אופטימליות והסיבות לכך הן:

1. גיוון הפתרון exploration : ברגע שנתקלים במינימום לוקלי אז אפשר לחלץ את התוכנית מהמצב הזה

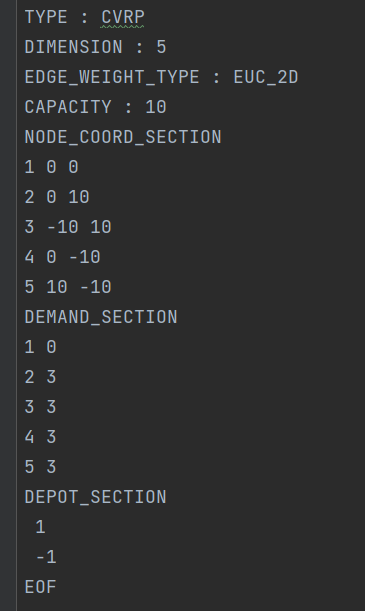
2. חיפוש מבוסס על זכרון : אנחנו שומרים מצבים טובים כך שלא צריך לחזור לנקודת ה 0 בכל פעם שנתקעים

3. שומר על איזון בין exploration ל exploitation

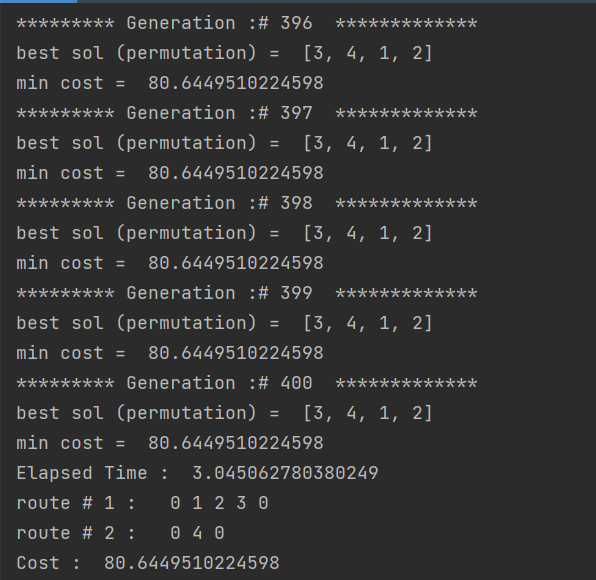
כמובן שהבדלים משמעותיים ניתן לראות עבור הבעיות הגדולות יותר כמו הדוגמא 3 או 4 ששם המימדים הם יותר גדולים

שאלה 6:

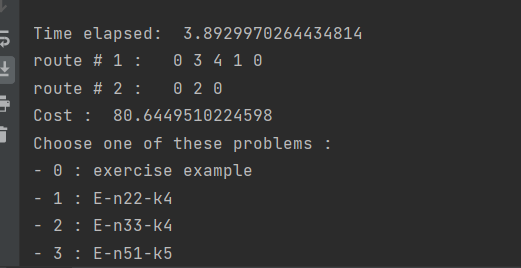
לצורך זה הוספנו קובץ txt חדש בפורמט של הדוגמאות באתר שמתאר את הדוגמא בקובץ המשימה :



תוצאה אחרי הרצת GA :



**הערה :** ספיציפית בדוגמא שיש בקובץ המשימה יש יותר מפתרון אופטימלי אחד ולא רק זה שסופק לנו בקובץ המשימה , למשל הפתרון הבא הוא גם אופטימלי :



שאלה 7:

גרף שמתאר את שיפור הפיטנס בבעיה 1:

תמונה שמכילה טקסט, צילום מסך, תרשים, קו

התיאור נוצר באופן אוטומטי