

# Genetic Algorithm

## מטרות:

- א. פיתוח ויישום אלגוריתם גנטי באופן ממוחשב GA
- ב. פתרון מטה היוריסטי לבעיית תכנית אב לייצור \רכש (גודל מנה אופטימאלי לביקוש משתנה).
- ג. יישום פתרון אופטימלי באופן ממוחשב - אלגוריתם WW (לשם השוואה עם פתרון GA) – ניתן ליישום רק על בעיות "קלות" ומכאן עולה הצורך באלגוריתם אחר (GA ניתן כדוגמא בעבודה זאת)

## 1. מבוא:

- א. אחת הבעיות הנפוצות בעולם היצרני, הינה קביעת גודל מנת ייצור\רכש אופטימלי. "אופטימלי" = לצורך העניין (וכהנחת יסוד בפרויקט זה) – העלות הנמוכה ביותר  $\min(\text{Total Cost})$
- ב. בעיה זאת קשה לפתרון ככל שיש תקופות זמן רבות יותר באופק התכנון.
- ג. הקושי נובע מכמות האפשרויות הרבות שיש לבדוק – לעיתים במצבים מסוימים הכמות כל כך רבה עד שחישוב פתרון אופטימלי הופך להיות בלתי אפשרי - בנקודת הזמן הנוכחית בשימוש הטכנולוגיות הקיימות כיום.
- ד. לאור האמור התפתחו קבוצות אלגוריתמים – הנקראים היוריסטיקות. אלגוריתמים אלה מוצאים פתרון שאינו הפתרון האופטימלי (בהכרח), אך במקרים רבים הוא פתרון מספיק טוב או פתרון העדיף מלא כלום.
- ה. בנוסף קיימות קבוצות אלגוריתמים נוספים שמבוססות בעיקר על סטטיסטיקה, באופן כזה שהן מחקות עיקרון מסוים בטבע. אלו הן מטה-היוריסטיקות – והן נחשבות חדשניות ומודרניות יותר. בעבודה זאת בוצע שימוש במטה-היוריסטיקה מסוג אלגוריתם גנטי – Genetic Algorithm לצורך פתרון בעיות מהסוג שיפורט בהמשך.
- ו. בנוסף לכך, במסגרת העבודה בוצע יישום לאלגוריתם WWA(Wagner Within). האלגוריתם פותר את הבעיה הנדונה כך שהפתרון הוא אופטימלי (בניגוד מטה-היוריסטיקות שלא בהכרח מוצאת אותו). המטרה היא להשוות את הפתרונות משני האלגוריתמים. כך ניתן לקבל אינדיקציה לגבי כמות החישובים (דורות) שנדרשים לאלגוריתם הגנטי כדי להגיע לפתרון אופטימאלי או קרוב אליו.
- ז. האלגוריתמים פותחו ויושמו בשפת C.

2. הגדרת הבעיה שהפריקט עוסק בו:

- $t$  – מספר תקופות התכנון (ימים, שבועות, חודשים, רבעונים וכו..)
- $Dt$  – ביקושים\דרישות נטו (לצורך העניין תכנית אב לביצוע)
- $K$  – עלות הזמנה (ללא תלות בכמות)
- $h$  – עלות (\$) אחסון פריט אחד במלאי לתקופה אחת

t	1	2	3	4	...
Dt	200	300	450	80	...

- א. הבעיה: לקבוע כמה לייצר\לרכוש בכל תקופה, תוך התחשבות בעלויות אחסון.
- ב. הנחה מרכזית: חוסר אסור.
- 3. מבנה הפתרון:
  - 1 – המלאי במחסן שעבורו נדרש תשלום – בהתאם ל  $h$  – עלות המלאי
  - $Q_t$  – כמות לרכוש\ייצור בתקופה  $t$

$Q_t$	500	0	450	80	...
$I$	-	300	-	-	...

\*\*\* הערה: הנתונים שלעיל בדוגמא זאת ניתנו לשם המחשה, לא מהווים פתרון נכון.

- ג. חישוב עלות כוללת – (כמות ההזמנות \*  $k$ ) + (סכום המלאים במחסן \*  $h$ )
- ד. לכל "סידור" של הפתרון  $Q_t$  – קיימת עלות כוללת שנובעת ממנו
- ה. האלגוריתם שיוצג להלן "מחפש" את העלות המינימאלית מבין כל האפשרויות שעליהן האלגוריתם "עובר". באופן תיאורטי יש  $t!$  אפשרויות לבדוק. לכן כאשר אופק התכנון רחב בעייה זאת הופכת להיות בעיית  $Kn$  במושגי מחשוב, לא ניתנת לפתרון בזמן סביר בטכנולוגיה הקיימת.
- הערה - בתקופה הראשונה חייבים לרכוש(כמות שתספיק לפחות לתקופה זאת) שכן חוסר אסור. אין להגיע למועד האספקה בלי מלאי זמין.

4. הסבר על עיקרון הפעולה של התכנית הממוחשבת:

- א. בתחילת פעולתה התכנית מפעילה אלגוריתם גנטי לצורך פתרון הבעיה. האלגוריתם רץ מספר מוגדר של איטרציות (דורות) על פי הגדרה מראש, כפי שיפורט.
- ב. עם סיום האלגוריתם הגנטי מופעל אוטומטית אלגוריתם WW(wagner within) – מציאת פתרון אופטימלי לאותה הבעיה, במטרה לראות האם האלגוריתם הגנטי "הגיע" לפתרון האופטימלי, אם לא אזי מה מידת הקירבה שלו לפתרון?
- ג. **הערה** – ברור שלדוגמאות שניתן להריץ עליהן את האלגוריתם אסור להיות מסובכות במיוחד, כי אז לחישוב הפתרון האופטימלי יידרש זמן רב מדי. לכן לעיתים האלגוריתם הגנטי עשוי להראות "מיותר" (כי לכאורה יש אלגוריתם שמוצא אופטימום). אך יתרון האלגוריתם הגנטי הוא דווקא כשהבעיה מאוד מסובכת וכשלא ניתן להשתמש ב-**ww**, במצב כזה ניתן למצוא פתרון "טוב מספיק" בהתחשב בקושי הרב שבמציאת פתרון אופטימלי.
5. האלגוריתם הגנטי: Genetic Algorithm
- ד. בהמשך לניסיונות מדענים בשנות ה-50 וה-60 למציאת פתרונות של אופטימיזציה של בעיות הנדסיות ומדעיות, באמצעות חיקוי מנגנונים בטבע, התפתח שיטה המבוססת על תהליך גנטי מתורת האבולוציה.
- ה. השימוש בפעולות הבסיסיות: שחלוף, **crossover** איחוד, **recombination** מוטציה **mutation** ובחירה **selection** מרכיבים את השלבים השונים באלגוריתם הגנטי כאסטרטגיה לפתרון בעיות. קיימים שימושים רבים לאלגוריתמים גנטיים הן לפתרון בעיות תיאורטיות למשל, בעיית הסוכן הנוסע והן בנושאים פרקטיים כלכלה, הנדסה וכו.
- ו. קל לזכור את העיקרון:
- a. לכל זוג פרטים יש את היכולת לייצר צאצא ולהעביר לו את המטען הגנטי שלהם.
- b. רק המתאימים ביותר שורדים על ידי העלאת סיכויי ההתרבות שלהם.
- c. יש צאצאים עם מוטציות
- ז. בעבודה הנוכחית בוצעה בוצע ייצוג כרומוזום באופן בינארי(קידוד):
- 1101111000011110**
- ח. משמעות: כל "ביט" מייצג את התשובה לשאלה: "האם לקנות בתקופה x?" כך שמיקום הביט, בדומה למערך נתונים בשפת תכנה, הוא בעצם האינדקס של המערך וזה גם מספר התקופה פחות 1. כמובן שיש לפענח את הכרומוזום כדי לדעת כמה לקנות בכל תקופה.
- ט. למעשה כל כרומוזום מהסוג שלעיל מהווה פתרון אפשרי לבעיה שתוארה.
- י. המעבר בין הפתרון לבין ייצוג הכרומוזומים מבוצע ע"י **קידוד ופענוח**: כלומר אם צריך "לקנות" בתקופה 4 אז בביט 5 יופיע: "1" ואופן הפענוח הוא הכפלה במערך הביקושים(demands[t]) באינדקס המתאים לו. הנושא מפורט בתוך שורות הקוד עצמו שבתכנית.

יא. מבנה עקרוני של האלגוריתם:

1. צור אוכלוסייה התחלתית
2. הערך את ההתאמה של כל פרט באוכלוסייה (fitness/ranking)
3. בחר את הפרטים המתאימים ביותר לזיווג
4. צור דור חדש של פרטים על ידי זיווג ומוטציה של הדור הקודם
5. חזור על שלבים 2–4 עד סיום הסימולציה

יב. עקרון פעולת התכנית:

a. התכנית מתחילה בהגדרות הבעיה והאוכלוסייה:

i. הבעיה נשמת במבנה נתונים מסוג `struct aProblem` המכיל את

כל הפרמטרים החיוניים של הבעיה שאותה האלגוריתם מנסה  
לפתור :

```
struct problemData{ // data structure for holding a problem data
    int t; // plan scope
    float h; // store cost - one unit for one period
    float k;
    int structSize; // for allocating memory
    // order fixed price
    int D[]; // demands for each period
}
```

b. האוכלוסייה הנבחרת מיוצגת ע"י כרומוזומים בינארים כפי שיתואר  
בהמשך.

c. לכל פרט באוכלוסייה, מחושבים

6. אופן הרצת התכנית:

```
int numGenerations=10;
int rankMethod=1;
int populationSize;
float percentOfCross=0.35;
```

## א. הגדרות משתמש:

a. מספר דורות (איטרציות) לחישוב `numberOfGenerations` =  
**כברירת מחדל התכנית משתמשת ב-10 דורות**, ניתן לשנות את הערך  
לכל מספר דורות רצוי.

b. Ranking – שיטת דרוג.

ניתן להשתמש ב-Fitness (`rankMethod = 1`)

או ב-Ranking (`rankMethod = 0`)

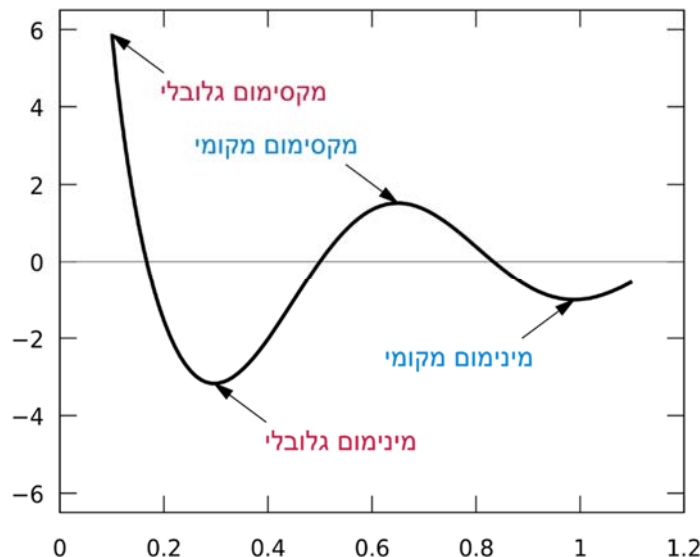
ברירת המחדל – `Ranking = 1`

הנ"ל תלוי בנתוני כל בעייה ספציפית.

\*\*\*\* אם אין הבדל משמעותי ב-fitness של כל כרומוזום יש לבחור

בשיטת RANKING. אחרת התכנית עלולה להכנס למצב של

סטגנציה (מינימום מקומי) לדוגמא:



c. גודל האוכלוסייה – מספר כרומוזומים שיהוו את האוכלוסייה של  
הפתרונות. בתכנית זאת המספר קבוע והוא נקבע שרירותית על פי  
מספר התקופות באופק התכנון. כלומר בבעיה של 12 חודשים גודל  
האוכלוסייה יהיה 12 כברירת מחדל. בבעיה של שבוע ימים (6) גודל  
האוכלוסייה יהיה 6 כרומוזומים. \* שיטה זאת יעילה כאשר מספר  
התקופות קטן כי עבור מספר תקופות גדול מספר האפשרויות הינו  
בעצרת ויתכן שגודל האוכלוסייה לא יהיה יעיל והחתיירה לפתרון טוב  
תהיה איטית מאוד. לצורכי לימוד ועיסוק בבעיות קצרות יחסית (עד  
10 תקופות) ניתן להשאיר את ברירת המחדל הקיימת. המשתנה  
`populationSize` : בתכנית

d. precentOfCross – הגדת אחוז הכרומוזומים מתוך האוכלוסייה שיעבור הכלאה. מאחר והבחירה נעשית על בסיס סטטיסטי יתכן שכרומוזום מסוים יבחר שוב באותו דור להיות הורה יותר מפעם אחת. הדבר תקין מבחינת האלגוריתם. גם בטבע זה כך.

e. בחירות המשתמש בתפריט:

### 1 – ברירת מחדל :

המערכת תריץ את האלגוריתם על בעיית ברירת המחדל של התכנית:

```
int t = 6;
float h = 0.3;
float k = 100;
int Demands[] = {250, 200, 180, 420, 350, 250};
float Qt[t];
for (i=0; i<t; i++) Qt[i]=0;
```

2- קריאת נתוני בעיה חדשה מקובץ (הזנת הנתונים בקובץ input.csv). אם המשתמש בוחר באפשרות זאת יש להזין את נתוני הבעיה שאותה רוצים לפתור.

**בקובץ Input.csv** קיימים נתוני בעיית ברירת המחדל כפי שרואים בטבלה הבאה (שניתנים לשינוי ע"י המשתמש):

t	8	(Enter periods number)		בשדה זה יש להקליד את מספר התקופות באופק התכנון				
h	1	(Enter unit storage cost)		בשדה זה יש להקליד את עלות אחסון יחידת מוצר אחת במלאי לתקופה אחת				
k	100	(Enter order cost)		בשדה זה יש להקליד את עלות ההזמנה – בהנחה שאין הגבלת כמות להזמנה				
period num	1	2	3	4	5	6	7	8
Demands	40	20	0	15	70	0	10	50

**\*\*\*\* יש לשנות אך ורק את השדות המודגשים בצהוב!**

**כל שינוי אחר במבנה קובץ הקלט עלול להוביל לשגיאת זמן ריצה!**

ניתן כמובן להוסיף ולהוריד תקופות וביקושים ללא הגבלה, כל עוד השדה t מכיל את מספר התקופות הנכון.

- \*\*\*הקובץ input.csv חייב להיות בתקיה הפעילה במידה והמשתמש בוחר: 2 – קריאה מקובץ.
- הקובץ חייב להיות בתקיה בה נמצא קובץ ההרצה EXE.
- אין לשנות את שם קובץ הנתונים)

סביבת הפיתוח של התכנית: Visual Studio code שימוש בקומפיילר gcc. התכנית הורצה ונבדקה במערכת הפעלה windows 10.

## 7. פלט לדוגמא:

א. נתוני (הבעיה מסומנים בתכלת). האלגוריתם הורץ למשך 20 דורות.

GeneticAlgoritem implementation :

looking a solution for problem :

Number of periods=6      holdCost = 0.3      invititionCost=100.0

Demands:

period 1: [250] period 2: [200] period 3: [180] period 4: [420] period 5: [350] period 6 [250] :

להלן עלות כוללת התחלתית שממנה האלגוריתם מתחיל בחיפוש אחר עלויות נמוכות יותר ויותר:

Initial Population :(current best Total cost = 565.0)

תצוגת הכרומוזמים לאחר קידוד (כל כרומוזום מייצג בצורה בינארית האם לקנות בתקופה המיוצגת ע"י אינדקס המיקום במערך הכרומוזום) :

chromosom serial:0, TotalCost:681.0, fitness:0.095, selection:0---> [1][1][0][0][1][0][  
chromosom serial:1, TotalCost:565.0, fitness:0.286, selection:0---> [1][0][1][1][0][1][  
chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][  
chromosom serial:3, TotalCost:626.0, fitness:0.143, selection:0---> [1][1][1][0][1][1][  
chromosom serial:4, TotalCost:565.0, fitness:0.238, selection:0---> [1][0][1][1][0][1][  
chromosom serial:5, TotalCost:573.0, fitness:0.190, selection:0---> [1][0][0][1][0][1][

תצוגת תהליך האבולוציה:

Generation --> 1

-----

Parents serial: 1 + 1 Were selected for crossOver Random Mask:[1][1][1][1][1][0][

chromosom serial: 1 was mutate

Parents serial: 1 + 1 Were selected for crossOver Random Mask:[1][0][0][0][0][0][

chromosom serial: 5 was mutate

chromosom serial:0, TotalCost:681.0, fitness:0.095, selection:0---> [1][1][0][0][1][0][  
chromosom serial:1, TotalCost:560.0, fitness:0.286, selection:0---> [1][0][1][1][1 [1][  
chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][  
chromosom serial:3, TotalCost:626.0, fitness:0.143, selection:0---> [1][1][1][0][1 [1][  
chromosom serial:4, TotalCost:565.0, fitness:0.238, selection:0---> [1][0][1][1][0 [1][  
chromosom serial:5, TotalCost:568.0, fitness:0.190, selection:0---> [1][0][0][1][1 [1][

best Chromosom: Total Cost =560.0 [1][0][1][1][1][1][

Generation --> 2

-----



---

Parents serial: 1 + 1 Were selected for crossOver Random Mask:[1][0][0][0][0] [0][

chromosom serial: 1 was mutate

Parents serial: 1 + 1 Were selected for crossOver Random Mask:[1][1][1][1][0] [0][

chromosom serial: 3 was mutate

chromosom serial:0, TotalCost:681.0, fitness:0.095, selection:0---> [1][1][0][0][1] [0][

chromosom serial:1, TotalCost:560.0, fitness:0.286, selection:0---> [1][0][1][1][1] [1][

chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:3, TotalCost:626.0, fitness:0.143, selection:0---> [1][1][1][0][1][1][

chromosom serial:4, TotalCost:565.0, fitness:0.238, selection:0---> [1][0][1][1][0][1][

chromosom serial:5, TotalCost:568.0, fitness:0.190, selection:0---> [1][0][0][1][1][1][

**best Chromosom: Total Cost =560.0 [1][0][1][1][1][1][**

**Generation --> 3**

-----

Parents serial: 0 + 4 Were selected for crossOver Random Mask:[1][1][1][1][1][1][

chromosom serial: 4 was mutate

Parents serial: 0 + 5 Were selected for crossOver Random Mask:[1][1][1][1][1][1][

chromosom serial: 1 was mutate

chromosom serial:0, TotalCost:681.0, fitness:0.095, selection:0---> [1][1][0][0][1][0][

chromosom serial:1, TotalCost:560.0, fitness:0.286, selection:0---> [1][0][1][1][1][1][

chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:3, TotalCost:626.0, fitness:0.143, selection:0---> [1][1][1][0][1][1][

chromosom serial:4, TotalCost:573.0, fitness:0.190, selection:0---> [1][0][0][1][0][1][

---

chromosom serial:5, TotalCost:568.0, fitness:0.238, selection:0---> [1][0][0][1][1][1][

best Chromosom: Total Cost =560.0 [1][0][1][1][1][1][

Generation --> 4

-----

Parents serial: 5 + 1 Were selected for crossOver Random Mask:[1][1][1][1][1][1][

chromosom serial: 5 was mutate

Parents serial: 1 + 4 Were selected for crossOver Random Mask:[1][1][0][0][0][0][

chromosom serial: 3 was mutate

chromosom serial:0, TotalCost:681.0, fitness:0.095, selection:0---> [1][1][0][0][1][0][

chromosom serial:1, TotalCost:573.0, fitness:0.190, selection:0---> [1][0][0][1][0][1][

chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:3, TotalCost:626.0, fitness:0.143, selection:0---> [1][1][1][0][1][1][

chromosom serial:4, TotalCost:560.0, fitness:0.238, selection:0---> [1][0][1][1][1][1][

chromosom serial:5, TotalCost:554.0, fitness:0.286, selection:0---> [1][1][0][1][1][1][

best Chromosom: Total Cost =554.0 [1][1][0][1][1][1][

Generation --> 5

-----

Parents serial: 4 + 1 Were selected for crossOver Random Mask:[1][1][1][1][1][1][

chromosom serial: 1 was mutate

Parents serial: 5 + 5 Were selected for crossOver Random Mask:[1][1][1][1][1][0][

chromosom serial: 3 was mutate

---

chromosom serial:0, TotalCost:681.0, fitness:0.143, selection:0---> [1][1][0][0][1][0][  
chromosom serial:1, TotalCost:568.0, fitness:0.190, selection:0---> [1][0][0][1][1][1][  
chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][  
chromosom serial:3, TotalCost:736.0, fitness:0.095, selection:0---> [1][1][1][0][0][1][  
chromosom serial:4, TotalCost:560.0, fitness:0.238, selection:0---> [1][0][1][1][1][1][  
chromosom serial:5, TotalCost:554.0, fitness:0.286, selection:0---> [1][1][0][1][1][1][

best Chromosom: Total Cost =554.0 [1][1][0][1][1][1][

Generation --> 6

-----

Parents serial: 0 + 5 Were selected for crossOver Random Mask:[1][1][0][0][0][0][

chromosom serial: 5 was mutate

Parents serial: 5 + 0 Were selected for crossOver Random Mask:[1][0][0][0][0][0][

chromosom serial: 5 was mutate

chromosom serial:0, TotalCost:706.0, fitness:0.143, selection:0---> [1][1][0][0][1][1][  
chromosom serial:1, TotalCost:568.0, fitness:0.238, selection:0---> [1][0][0][1][1][1][  
chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][  
chromosom serial:3, TotalCost:736.0, fitness:0.095, selection:0---> [1][1][1][0][0][1][  
chromosom serial:4, TotalCost:560.0, fitness:0.286, selection:0---> [1][0][1][1][1][1][  
chromosom serial:5, TotalCost:600.0, fitness:0.190, selection:0---> [1][1][1][1][1][1][

best Chromosom: Total Cost =554.0 [1][1][0][1][1][1][

Generation --> 7

---

Parents serial: 1 + 1 Were selected for crossOver Random Mask:[1][1][0][0][0][0][

chromosom serial: 0 was mutate

Parents serial: 4 + 1 Were selected for crossOver Random Mask:[1][1][0][0][0][0][

chromosom serial: 5 was mutate

chromosom serial:0, TotalCost:921.0, fitness:0.095, selection:0---> [1][1][0][0][0][1][

chromosom serial:1, TotalCost:560.0, fitness:0.286, selection:0---> [1][0][1][1][1][1][

chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:3, TotalCost:736.0, fitness:0.143, selection:0---> [1][1][1][0][0][1][

chromosom serial:4, TotalCost:568.0, fitness:0.238, selection:0---> [1][0][0][1][1][1][

chromosom serial:5, TotalCost:626.0, fitness:0.190, selection:0---> [1][1][1][0][1][1][

**best Chromosom: Total Cost =554.0 [1][1][0][1][1][1][**

**Generation --> 8**

---

Parents serial: 1 + 4 Were selected for crossOver Random Mask:[1][1][1][1][1][1][

chromosom serial: 5 was mutate

Parents serial: 1 + 5 Were selected for crossOver Random Mask:[1][1][1][0][0][0][

chromosom serial: 3 was mutate

chromosom serial:0, TotalCost:921.0, fitness:0.095, selection:0---> [1][1][0][0][0][1][

chromosom serial:1, TotalCost:561.0, fitness:0.286, selection:0---> [1][0][1][0][1][0][

chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:3, TotalCost:861.0, fitness:0.143, selection:0---> [1][1][1][0][0][0][

---

chromosom serial:4, TotalCost:568.0, fitness:0.238, selection:0---> [1][0][0][1][1[1][

chromosom serial:5, TotalCost:600.0, fitness:0.190, selection:0---> [1][1][1][1][1[1][

best Chromosom: Total Cost =554.0 [1][1][0][1][1[1][

Generation --> 9

-----

Parents serial: 4 + 1 Were selected for crossOver Random Mask:[1][1][1][0][0[0][

chromosom serial: 5 was mutate

Parents serial: 4 + 3 Were selected for crossOver Random Mask:[1][1][1][1][1[0][

chromosom serial: 0 was mutate

chromosom serial:0, TotalCost:1166.0, fitness:0.095, selection:0---> [1][0][0][0][0[1][

chromosom serial:1, TotalCost:560.0, fitness:0.286, selection:0---> [1][0][1][1][1[1][

chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0[0][

chromosom serial:3, TotalCost:861.0, fitness:0.143, selection:0---> [1][1][1][0][0[0][

chromosom serial:4, TotalCost:821.0, fitness:0.190, selection:0---> [1][0][0][0][1[0][

chromosom serial:5, TotalCost:605.0, fitness:0.238, selection:0---> [1][1][1][1][0[1][

best Chromosom: Total Cost =554.0 [1][1][0][1][1[1][

Generation --> 10

-----

Parents serial: 5 + 1 Were selected for crossOver Random Mask:[1][1][1][1][1[0][

chromosom serial: 5 was mutate

Parents serial: 4 + 3 Were selected for crossOver Random Mask:[1][1][1][1][1[1][

---

chromosom serial: 0 was mutate

chromosom serial:0, TotalCost:921.0, fitness:0.095, selection:0---> [1][1][0][0][0][1][  
chromosom serial:1, TotalCost:560.0, fitness:0.286, selection:0---> [1][0][1][1][1][1][  
chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][  
chromosom serial:3, TotalCost:861.0, fitness:0.143, selection:0---> [1][1][1][0][0][0][  
chromosom serial:4, TotalCost:821.0, fitness:0.190, selection:0---> [1][0][0][0][1][0][  
chromosom serial:5, TotalCost:655.0, fitness:0.238, selection:0---> [1][1][1][1][0][0][

best Chromosom: Total Cost =554.0 [1][1][0][1][1][1][

Generation --> 11

-----

Parents serial: 1 + 0 Were selected for crossOver Random Mask:[1][1][1][1][1][0][

chromosom serial: 3 was mutate

Parents serial: 0 + 0 Were selected for crossOver Random Mask:[1][1][1][1][1][1][

chromosom serial: 0 was mutate

chromosom serial:0, TotalCost:1121.0, fitness:0.095, selection:0---> [1][1][0][0][0][0][  
chromosom serial:1, TotalCost:560.0, fitness:0.286, selection:0---> [1][0][1][1][1][1][  
chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][  
chromosom serial:3, TotalCost:601.0, fitness:0.238, selection:0---> [1][1][1][0][1][0][  
chromosom serial:4, TotalCost:821.0, fitness:0.143, selection:0---> [1][0][0][0][1][0][  
chromosom serial:5, TotalCost:655.0, fitness:0.190, selection:0---> [1][1][1][1][0][0][

best Chromosom: Total Cost =554.0 [1][1][0][1][1][1][

---

## Generation --> 12

-----

Parents serial: 3 + 1 Were selected for crossOver Random Mask:[1][0][0][0][0][0][

chromosom serial: 1 was mutate

Parents serial: 1 + 1 Were selected for crossOver Random Mask:[1][1][1][1][1][0][

chromosom serial: 4 was mutate

chromosom serial:0, TotalCost:1121.0, fitness:0.095, selection:0---> [1][1][0][0][0][0][

chromosom serial:1, TotalCost:601.0, fitness:0.190, selection:0---> [1][1][1][0][1][0][

chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:3, TotalCost:560.0, fitness:0.238, selection:0---> [1][0][1][1][1][1][

chromosom serial:4, TotalCost:543.0, fitness:0.286, selection:0---> [1][0][0][1][1][0][

chromosom serial:5, TotalCost:655.0, fitness:0.143, selection:0---> [1][1][1][1][0][0][

**best Chromosom: Total Cost =543.0 [1][0][0][1][1][0][**

## Generation --> 13

-----

Parents serial: 0 + 1 Were selected for crossOver Random Mask:[1][0][0][0][0][0][

chromosom serial: 1 was mutate

Parents serial: 0 + 0 Were selected for crossOver Random Mask:[1][1][0][0][0][0][

chromosom serial: 3 was mutate

chromosom serial:0, TotalCost:601.0, fitness:0.190, selection:0---> [1][1][1][0][1][0][

chromosom serial:1, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:2, TotalCost:1441.0, fitness:0.095, selection:0---> [1][0][0][0][0][0][

---

chromosom serial:3, TotalCost:535.0, fitness:0.286, selection:0---> [1][0][1][1][1][0][

chromosom serial:4, TotalCost:543.0, fitness:0.238, selection:0---> [1][0][0][1][1][0][

chromosom serial:5, TotalCost:655.0, fitness:0.143, selection:0---> [1][1][1][1][0][0][

best Chromosom: Total Cost =535.0 [1][0][1][1][1][0][

Generation --> 14

-----

Parents serial: 5 + 4 Were selected for crossOver Random Mask:[1][1][1][1][1][0][

chromosom serial: 5 was mutate

Parents serial: 3 + 0 Were selected for crossOver Random Mask:[1][1][1][1][1][0][

chromosom serial: 4 was mutate

chromosom serial:0, TotalCost:601.0, fitness:0.238, selection:0---> [1][1][1][0][1][0][

chromosom serial:1, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:2, TotalCost:1441.0, fitness:0.095, selection:0---> [1][0][0][0][0][0][

chromosom serial:3, TotalCost:535.0, fitness:0.286, selection:0---> [1][0][1][1][1][0][

chromosom serial:4, TotalCost:821.0, fitness:0.143, selection:0---> [1][0][0][0][1][0][

chromosom serial:5, TotalCost:609.0, fitness:0.190, selection:0---> [1][1][0][1][0][0][

best Chromosom: Total Cost =535.0 [1][0][1][1][1][0][

Generation --> 15

-----

Parents serial: 3 + 3 Were selected for crossOver Random Mask:[1][1][1][1][0][0][

chromosom serial: 4 was mutate



---

Parents serial: 5 + 2 Were selected for crossOver Random Mask:[1][1][1][1][1][1][

chromosom serial: 5 was mutate

chromosom serial:0, TotalCost:601.0, fitness:0.238, selection:0---> [1][1][1][0][1][0][

chromosom serial:1, TotalCost:1441.0, fitness:0.143, selection:0---> [1][0][0][0][0][0][

chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:3, TotalCost:535.0, fitness:0.286, selection:0---> [1][0][1][1][1][0][

chromosom serial:4, TotalCost:1441.0, fitness:0.095, selection:0---> [1][0][0][0][0][0][

chromosom serial:5, TotalCost:1121.0, fitness:0.190, selection:0---> [1][1][0][0][0][0][

best Chromosom: Total Cost =535.0 [1][0][1][1][1][0][

Generation --> 16

-----

Parents serial: 0 + 3 Were selected for crossOver Random Mask:[1][0][0][0][0][0][

chromosom serial: 3 was mutate

Parents serial: 5 + 3 Were selected for crossOver Random Mask:[1][0][0][0][0][0][

chromosom serial: 5 was mutate

chromosom serial:0, TotalCost:535.0, fitness:0.286, selection:0---> [1][0][1][1][1][0][

chromosom serial:1, TotalCost:1441.0, fitness:0.143, selection:0---> [1][0][0][0][0][0][

chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:3, TotalCost:1121.0, fitness:0.190, selection:0---> [1][1][0][0][0][0][

chromosom serial:4, TotalCost:1441.0, fitness:0.095, selection:0---> [1][0][0][0][0][0][

chromosom serial:5, TotalCost:535.0, fitness:0.238, selection:0---> [1][0][1][1][1][0][

best Chromosom: Total Cost =535.0 [1][0][1][1][1][0][

---

Generation --> 17

-----

Parents serial: 0 + 4 Were selected for crossOver Random Mask:[1][1][0][0][0][0][

chromosom serial: 5 was mutate

Parents serial: 0 + 0 Were selected for crossOver Random Mask:[1][1][1][1][1][1][

chromosom serial: 5 was mutate

chromosom serial:0, TotalCost:1441.0, fitness:0.143, selection:0---> [1][0][0][0][0][0][

chromosom serial:1, TotalCost:1441.0, fitness:0.095, selection:0---> [1][0][0][0][0][0][

chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:3, TotalCost:1121.0, fitness:0.190, selection:0---> [1][1][0][0][0][0][

chromosom serial:4, TotalCost:535.0, fitness:0.286, selection:0---> [1][0][1][1][1][0][

chromosom serial:5, TotalCost:821.0, fitness:0.238, selection:0---> [1][0][0][0][1][0][

best Chromosom: Total Cost =535.0 [1][0][1][1][1][0][

Generation --> 18

-----

Parents serial: 5 + 0 Were selected for crossOver Random Mask:[1][1][1][1][0][0][

chromosom serial: 3 was mutate

Parents serial: 4 + 5 Were selected for crossOver Random Mask:[1][1][0][0][0][0][

chromosom serial: 4 was mutate

chromosom serial:0, TotalCost:821.0, fitness:0.238, selection:0---> [1][0][0][0][1][0][

chromosom serial:1, TotalCost:1441.0, fitness:0.095, selection:0---> [1][0][0][0][0][0][

---

chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:3, TotalCost:1441.0, fitness:0.143, selection:0---> [1][0][0][0][0][0][

chromosom serial:4, TotalCost:1121.0, fitness:0.190, selection:0---> [1][1][0][0][0][0][

chromosom serial:5, TotalCost:535.0, fitness:0.286, selection:0---> [1][0][1][1][1][0][

best Chromosom: Total Cost =535.0 [1][0][1][1][1][0][

Generation --> 19

-----

Parents serial: 4 + 5 Were selected for crossOver Random Mask:[1][1][1][1][1][1][

chromosom serial: 0 was mutate

Parents serial: 1 + 4 Were selected for crossOver Random Mask:[1][1][1][0][0][0][

Chromosom serial: 5 was mutate

chromosom serial:0, TotalCost:1441.0, fitness:0.095, selection:0---> [1][0][0][0][0][0][

chromosom serial:1, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:2, TotalCost:1441.0, fitness:0.190, selection:0---> [1][0][0][0][0][0][

chromosom serial:3, TotalCost:1441.0, fitness:0.143, selection:0---> [1][0][0][0][0][0][

chromosom serial:4, TotalCost:1121.0, fitness:0.238, selection:0---> [1][1][0][0][0][0][

chromosom serial:5, TotalCost:615.0, fitness:0.286, selection:0---> [1][0][1][1][0][0][

best Chromosom: Total Cost =535.0 [1][0][1][1][1][0][

Generation --> 20

-----

Parents serial: 3 + 0 Were selected for crossOver Random Mask:[1][0][0][0][0][0][

chromosom serial: 4 was mutate

Parents serial: 1 + 0 Were selected for crossOver Random Mask:[1][1][0][0][0][0][

chromosom serial: 0 was mutate

chromosom serial:0, TotalCost:1166.0, fitness:0.190, selection:0---> [1][0][0][0][0][1][

chromosom serial:1, TotalCost:1441.0, fitness:0.095, selection:0---> [1][0][0][0][0][0][

chromosom serial:2, TotalCost:1441.0, fitness:0.048, selection:0---> [1][0][0][0][0][0][

chromosom serial:3, TotalCost:1441.0, fitness:0.143, selection:0---> [1][0][0][0][0][0][

chromosom serial:4, TotalCost:609.0, fitness:0.286, selection:0---> [1][1][0][1][0][0][

chromosom serial:5, TotalCost:615.0, fitness:0.238, selection:0---> [1][0][1][1][0][0][

**best Chromosom: Total Cost =535.0 [1][0][1][1][1][0][**

## 8. תוצאות ומסקנות:

- a. מניתוח הפלט לדוגמא שלעיל ניתן לראות שהאלגוריתם התחיל מעלות 565\$ ובסוף הגיע לתוצאה 535\$ (לאחר 20 דורות)
- b. לא בכל דור קיימת התקדמות! יתרה מכך יש דורות שבהם הפתרונות (הצאצאים) אף גרועים יותר בהשוואה לנקודת ההתחלה. זה נובע כי האלגוריתם מתבסס על מנוע סטטיסטי, ה"התקרבות" לפתרון משמעותית רק ככל שמקיימים יותר ויותר איטרציות (דורות).
- c. לאחר סיום האלגוריתם הגנטי התכנית קוראת לפונקציה WWA ומחשבת פתרון אופטימלי (לפי וואגנר). במקרה זה הפונקציה חישה: TC=529\$. זה הפתרון האופטימלי, לעומת 535\$ בשיטת GA

```
+++++
Calculate optimal solution according Wagner within algoritem:
+++++
period 1 buy: 250
period 2 buy: 380
period 3 buy: 0
period 4 buy: 420
period 5 buy: 600
period 6 buy: 0
WWA optimal Total Cost TC= 529$
Soulution File: soulution.csv Created Successfully!
+++++
```

d. אם הפתרון לא מספיק טוב ניתן להגדיל את מספר הדורות ואף לשנות פרמטרים, כתלות במשאבי המחשב שיש לרשות הפתרון.

e. לעיתים ניתן לשלוט ב"קצב" ההתכנסות מדור לדור – ע"י שינוי פרמטרים כמו גודל האוכלוסייה, אופן ההכלאה, שיטת הדרוג של כל כרומוזום(פתרון) ועוד...

f. הפרמטרים השונים של האלגוריתם הגנטי משפיעים על קצב ההתכנסות לפתרון (גודל אוכלוסייה, כמות הכלאות, אופן ביצוע מוטציות ועוד..).

g. מצאתי שלא ניתן לקבוע מראש את הפרמטרים שכן האלגוריתם יכול באופן עקרוני לחפש פתרון לכל סוג של בעיה (**שיש לה פונקציית מטרה ידועה**). הדבר דורש ניסוי ותהייה. אבל בכל מקרה כשכמות הדורות עולה מטבע הדברים האלגוריתם מתקרב מאוד לפתרון האופטימלי ואף מוצא אותו!

h. אחד החסרונות המעניינים באלגוריתם GA הוא שהאלגוריתם לא "יודע" שהוא הגיע לפתרון האופטימלי. אם לצורך העניין האלגוריתם מצא באופן אקראי את הפתרון כבר בהתחלה הוא יחזיק בו כפתרון הטוב ביותר עד כה וימשיך לחפש פתרונות טובים יותר עד לסיום הריצה.

i. בסוף כל סיום הרצה התכנית יוצרת קובץ CSV ובו נתוני הפתרונות של האלגוריתם הגנטי בהשוואה לפתרון אופטימלי. הקובץ מיוצר באופן אוטומטי בתקיה הנוכחית: **solution.csv**. אם קיים כבר קובץ בשם זה במיקום הנכחי הוא "נדרס" בכל הרצה של התכנית, כך שהקובץ תמיד מכיל את נתוני הפתרון של ההרצה האחרונה.

j. לעתיד –

i. בדיקת השפעת הפרמטרים(מספר צאצאים בכל דור, שיטת ניקוד RANKING וכו') על קצב ההתכנסות לפתרון. ע"י הרצת התכנית על מגוון רב של בעיות ניתן יהיה לייצר בסיס נתונים שישמש לאופטימיזציה של האלגוריתם.

ii. פיתוח שיטה לקביעת גודל האוכלוסייה האופטימלי (מס כרומוזומים) בבעיות שלהן יש אופק תכנון גדול - 24 תקופות ומעלה.