**2. בעיית N-Queens**

**2.1 פונקציית הפיטנס:**

בחירת פונקציית ה FITNESS.

פונקציית ה fitness שבחרנו היא מספר האיומים שבלוח כך שאם הגענו ל 0 איומים בלוח אז מצאנו פתרון לבעיה.

חישוב מספר האיומים נעשה על ידי חישוב מס המלכות המאיימות אחת על שניה בכל עמודה ושורה בנוסף למספר המלכות המאיימות אחת על השנייה באלכסון.

ייצוג הלוח נעשה בעזרת מערך חד ממדי ולכן חישוב האיומים נעשה בפונקציה הבאה:

int calc\_fitness\_for\_sequence(vector<int> &sequence)

{

int clashes = 0;

int rowColumnClashes = abs(int(sequence.size() - uniqueLength(sequence)));

clashes += rowColumnClashes;

int dx, dy, i, j;

// calculate diagonal clashes

for (i = 0; i < BOARD\_SIZE; i++){

for (j = 0; j < BOARD\_SIZE; j++){

if (i != j){

dx = abs(i - j);

dy = abs(sequence[i] - sequence[j]);

if (dx == dy)

clashes += 1;

}

}

}

return clashes;

**}**

**הערה\*: על מנת לחשב את מספר האיומים שהם לא באלכסון פשוט מחשבים את כמות המספרים השונים ונחסיר את התוצאה מכמות האיברים. בהתחלה אפשרנו איומים באותה שורה\עמודה אבל בהמשך הפכנו את המימוש לתמורות ולכן במימוש החדש ישנם רק איומים אלכסוניים, אבל בכל זאת השארנו את החישוב (שורה 4 בקוד למעלה).**

**2.2 ניסויים לבחירת טכניקה הכי טובה:**

המטרה שלנו פה היא לבחור קומבנציה הכי טובה עבור {אסטרטגית מוטציה, אסטרטגית שיחלוף, אסטרטגית בחירת הורים} כך ש:  
אסטרטגית מוטציה היא אחת מ : swap mutation , scramble mutation  
אסטרטגית שיחלוף היא אחת מ : ordered crossover, cyclic crossover  
אסטרטגית בחירת הורים היא אחת מ tournament select , random select

כדי לבחור קומבנציה הכי טובה חילקנו הניסויים ל 2 שלבים.  
כך שעבור גדלי לוח 8 ו 16 עשינו ניסויים עבור כל הקומבנציות האפשריים של {אסטרטגית מוטציה, אסטרטגית שיחלוף, אסטרטגית בחירת הורים} ואז לקחנו 2 קמבנציות הכי טובות לפי שני ממדים שהם ממוצע זמן וממוצע מספר איטרציות(אם הגענו למספר אטרציות 16384 ולא מצאנו פתרון אז עוצרים וכותבים FAIL) והמשכנו איתן עבור הניסויים בגודל לוח של 32 ו 64.  
בסוף בחרנו בקומבנציה שנתנה ביצועים הכי טובים בין שתי הקומבנציות.

אלו היו התוצאות:

X =crossover  
mutate = mutation strategy  
select = parents selection strategy  
  
OX = Ordered Crossover  
CX = Cyclic Crossover  
  
swap = swap mutation  
scramble = scramble mutation  
  
rand = random select  
tournament = tournament select

* הזמן מוגדר בשניות.





**ולכן אפשר לראות לפי הטבלאות שהקומבנציות הכי טובות הן:   
{ CX + swap + rand} ו { CX + swap + tournament }**

**ולכן בחרנו להמשיך איתן בבדיקות הבאות.**

**והתוצאות היו:**





**ולכן לפי התוצאות אפשר לראות שהקומבינציה {CX, swap, rand} היא הכי טובה ולכן נמשיך איתה לחלק 2.**

**2.3 בדיקת השפעת פרמטרים שונים על הביצועים:**

עכשיו אנו רוצים לבדוק מה הם הפרמטרים הכי טובים שנבחר.  
בהתחלה בדקנו מהו גודל אוכלסיה טוב ביותר ואז המשכנו איתו לבדוק מהי הסתברות מוטציה הכי טובה ובסוף בדקנו מהו פרופורצית האוכלוסיה האליטיסטית הכי טוב.  
וזה מה שיצא לנו:

**גודל אוכלסיה:**





**הסתברות מוטציה:**

  
  




**פרופורציית האוכלוסייה האליטיסטית:**

  
  




**לסיכום:**

**ולכן לפי הטבלאות ולפי הגרפים אפשר לראות שככל שגודל האוכלוסייה גדל או פרופורציית האוכלוסייה האליטיסטית גדלה אז הביצועים יהיו פחות טובים (עבור הנתונים שבחרנו), לגבי הסתברות המוטציה יצא לנו שעבור הסתברות מוטציה 0.25 זה נתן ביצועים הכי טובים אם זה היה נמוך אז זה היה לוקח יותר זמן כי זה לא מתקדם בצורה גדולה ואם היתה הסתברות גדולה זה היה מתחלף תמיד ונתן ביצועים פחות טובים מזה שנתן 0.25.  
ולכן:**

**גודל אוכלסיה = 1024**

**הסתברות מוטציה = 0.25**

**פרופורציית האוכלוסייה האליטיסטית = 0.05**