**מעבדה בבינה מלאכותית**

**דו"ח תרגיל בית 1 –** **Genetic Algorithms**

**תאריך הגשה:** 31.3.

**ד:** 3194

**ג:** 3980

1. נוסיף פונקציות לחישוב ממוצע ה-fitness וסטיית התקן.

//1. Calc the fitness average - it returns to print best

double calcAvgFitness(ga\_vector &population)

{

    double avg = 0 ;

    for ( int i =0 ; i< population.size() ; i++)

    {

        avg +=population[i].fitness;

    }

    return avg/GA\_POPSIZE;

}

חישוב ממוצע: סכימת ה-fitness וחלוקה ב-GA\_POPSIZE

double calcStandardDeviation(ga\_vector &population, double avg)

{

    double sum = 0;

    for (int i=0; i < population.size(); i++)

    {

    }

    return sqrt(sum);

}

חישוב סטיית תקן: נשתמש בנוסחאת סטיית התקן.

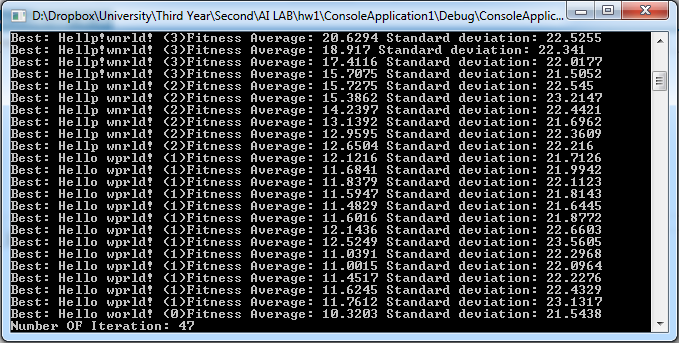
1. הוספת היוריסטיקה לשיפור:

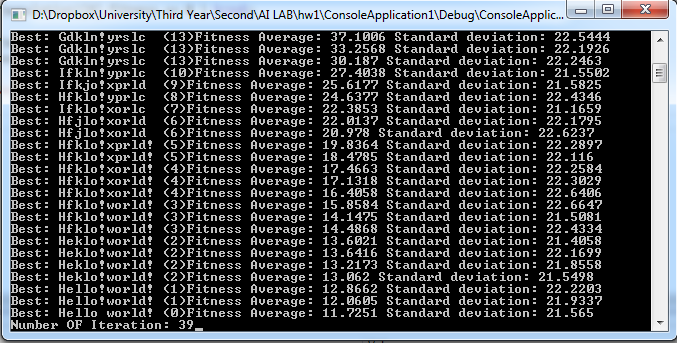
השוואה עם ההיוריסטיקה המקורית : בהיוריסטיקה המקורית לא נתנו שום יתרון למילה

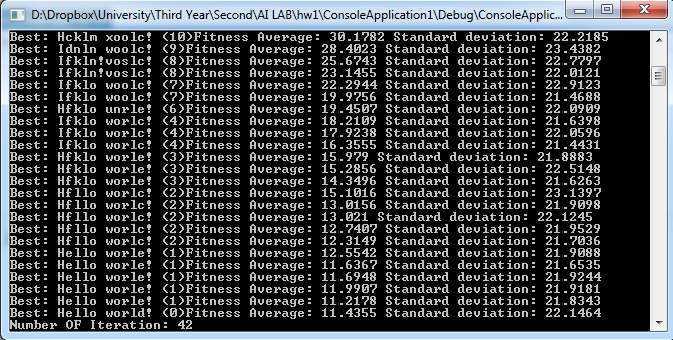
שמנו לב לשיפור משמעותי בזמן הריצה וכמות האיטרציות לאחר הוספת ההיוריסטיקה , בממוצע עשירית מכמות ההרצות עם ההיוריסטיקה המקורית הגיעה למספר מקסימלי של איטרציות בלי להגיע למטרה.

ידוע כי מוטציה מתרחשת בהסתברות נמוכה . ההיוריסטיקה שלנו נותנת יתרון למוטציה שמקרבת אותנו למילת המטרה.

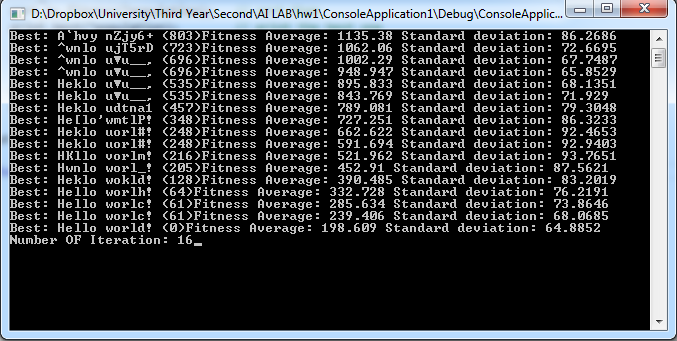
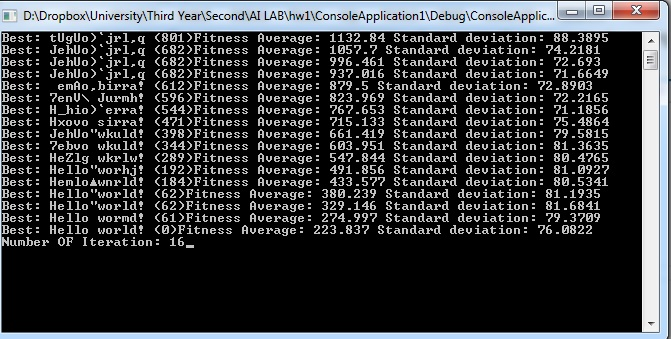
**להלן דוגמאות הרצה -** 3 הרצות מקוריות ו – 3 הרצות משופרות

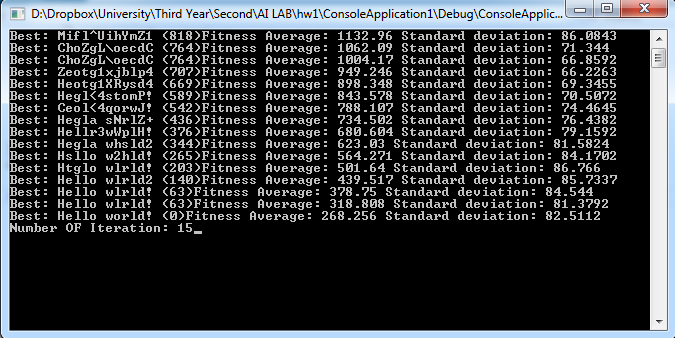






דוגמא ל3 הרצות עם שינוי בהיוריסטיקה

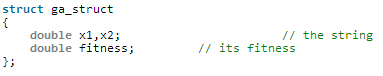




**לאחר הרצות רבות- ניתן לראות את השינוי בהיוריסטיקה בעזרת גרף:**

1. מינימום של פונקציה

ראשית, נבצע שינוי ב-struct על מנת לייצג את שני המשתנים: x1,x2



נגריל x1,x2 אקראים ואת חישוב ה-fitness נבצע ע"י פונקציית MathFunction (population[i]).

פונקציית MathFuction(population[i]):



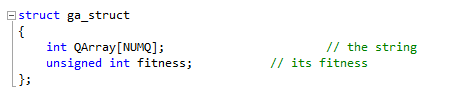
**חישוב היוריסטיקה:** חישוב ההיוריסטיקה הוא על פי הנוסחא שקיבלנו. נמצא זוג מספרים x1 ו-x2 שהוגרלו- נבדוק את ערך הפוקציה ונעדכן את ערך הfitness.

כמו כן, אנחנו מעוניינים בערך מינימלי ואנחנו יודעים ש

4. **בעיית n המלכות:**

ראשית נבצע שינוי בstruct:

קטע קוד:



NUMQ: define של מספר המלכות.

על מנת לממש את בעיית n המלכות- עלינו לבצע שינויים בפונקציות mate ו-mutate:

ראשית, נוסיף פונקציה המאתחלת QArray להיות פרמוטציה של המספרים 1 עד NUMQ.

קטע קוד:

**פונקציית Mate:**

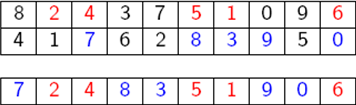
קטע קוד: בחרנו להשתמש באלגוריתם שיטת השחלוף:**OX – Ordered crossover**

הסבר:

נבחר מחצית מהאלמנטים מההורה הראשון (נאמר, האב) ונמקם אותם ע"פ האינדקסים שלהם אצל האב בגן הילד.

לאחר מכן, נעבור על הגן של האם, וכל מספר שלא נלקח ע"י האב והועבר לבן, נשים במקום הראשון הפנוי אצל הילד.

לדוגמא:



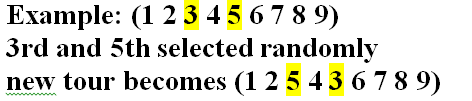
נבחר מחצית מהאיברים מהאב **(אדום)** ונעתיק לילד. את השאר **(כחול)** ניקח מהאדם ונעתיק ע"פ הסדר לבן.

**פונקציית Mutate:**

קטע קוד: בחרנו להשתמש במוטצית החלפה: **exchange**

הסבר: בחרנו באקראי 2 מספרים בוקטור המספרים- והחלפנו את ערכם זה בזה.

לדוגמא:



**פונקציה היוריסטית:** נשים לב כי i מסמן את השורה בלוח המשחק ו-QArray[i] מסמל את העמודה. בכך

נשתמש בפונקציית ההיורסיטית הבאה:

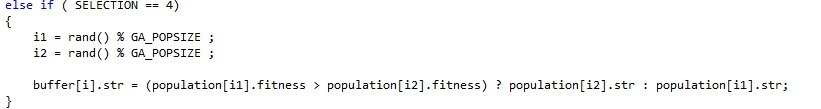
|row-col| = |QArray[row] – Qarray]col]|

אם מתקיים, נבצע: fitnees++.

5.

א) נבצע אסטרטגיית selection: "טורניר". SELECTION =4

קטע קוד:

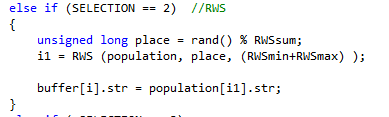


הסבר: נבחר אקראית 2 גנים מתוך מאגר הגנים הנתון- נבחר את הטוב מבינהם, ונחזיר אותו.

ב) נבצע אלגוריתם RWS: SELECTION = 2

נשים לב שעל מנת להפעיל את RWS

קטע קוד: הקוד ממומש בפנוקציית mate: עם פונקציות עזר.



הסבר:

בפונקציית mate:

חישבנו את min ו max של fitness ואת סכום כל ה-fitness של הדור - RWSsun.

כאשר נ

פונקציית RWS:

קטע קוד:

הסבר: ביצענו את פעולת הנירמול ובדיקה האם הסכום שיש לנו כרגע- נמצא בטווח התקין ע"פ האלגוריתם של RWS.

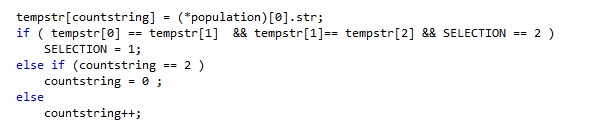
אם כן, נחזיר את המיקום הנוכחי.

נשים לב כי ה

אחת הבעיות של RWS היא שבסוף הריצה ה-fitness הוא זהה וכך המחרוזת ולעיתים רחוקות באמת עוצר.

הוספנו מנגנון במטרה לשפר את ביצועיי האלגוריתם:

קוד:



הסבר: ברגע שזיהינו 3 מחרוזות רצופות זהות- החלפנו את המנגנון של SELECTION למנגנון "בול פגיעה" – שמנו לב לשיפור חלק מהפעמים. השיפור נראה לעין כשהוא מתקיים.

**ניתן לראות שינוי בזמן ריצת האלגוריתם לאחר הוספת האסטרטגיה בעזרת גרף:**

* הערה: כאשר ההרצה הסתיימה ללא הצלחה- שמנו ערך 0

6.

א) הוספת Two Point Crossover ל"בול פגיעה": SELECTION =3.

**Two Point Crossover:** נוסיף את אסטרטגיית השיחלוף הבאה לבעיית "בול פגיעה".

את השינויים נבצע בפונקציית Mate.

קטע קוד:

הסבר:

הוספנו 2 משתנים: LocationOne ו-LocationTwo: אלה מבצעים rand ובוחרים את שני האינדקסים האקראיים שבינהם נבצע את אסטרטגיית השיחלוף.

נבצע החלפה של כל האיברים בטווח בין

**ניתן לראות שינוי בזמן ריצת האלגוריתם לאחר הוספת האסטרטגיה בעזרת גרף:**

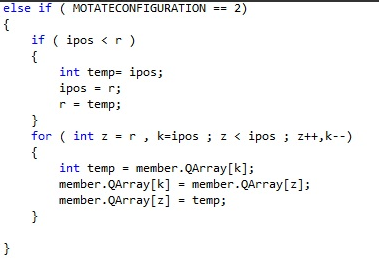
* הערה: כאשר ההרצה הסתיימה ללא הצלחה- שמנו ערך 0

ב) הוספת אסטרטגיית מוטציה ושיחלוף בבעית N המלכות:

**שיחלוף:** נשתמש בשיטת שיחלוף:**PMX – Partially Matched crossover :**

נוסיף את קטע הקוד לפונקציית Mutate:

קטע קוד:



הסבר: נשתמש באלגוריתם שניתן לנו. נבחר 2 גנים שייצגו את האב לדוגמא:



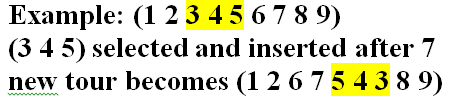
i נבחר ע"י rand



Parents

**מוטציה:** נשתמש בשיטת המוצטיה: **מוטצית היפוך - inversion mutation:**

קטע קוד:

הסבר: נשמש באלגוריתם של מוטצית היפוך: נגריל 2 מספרים ipos, r. אם לדוגמא:

7.

א) גודל האולוסיה:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Population size |  | iteration | Starting Point | "Bull's eye" | SELECTION = 2 "RWS" | SELECTION = 3 "Two Point CrossOver" | SELECTION =4 "Turnament" |
| 1000 |  | 1 |  | 16 |  | 10 | 913 |
| 1000 |  | 2 | 41 | 21 |  | 12 | 1258 |
| 1000 |  | 3 | 52 | 17 |  | 12 |  |
| 1000 |  | 4 | 1367 | 17 |  | 12 | 943 |
| 1000 |  | 5 | 739 | 17 | 592 | 12 | 1034 |
| 1500 |  | 1 | 345 | 16 | 1016 | 10 | 875 |
| 1500 |  | 2 | 34 | 18 | 3967 | 11 | 1148 |
| 1500 |  | 3 | 452 | 16 |  | 10 | 956 |
| 1500 |  | 4 | 36 | 13 |  |  |  |
| 1500 |  | 5 | 46 | 16 |  | 11 | 927 |
| 2500 |  | 1 |  | 15 |  | 10 | 179 |
| 2500 |  | 2 | 32 | 15 |  | 11 |  |
| 2500 |  | 3 | 51 | 13 |  | 10 | 526 |
| 2500 |  | 4 | 29 | 15 |  | 9 | 143 |
| 2500 |  | 5 | 27 | 15 |  | 11 |  |
| 3000 |  | 1 | 40 | 16 | 50 | 11 | 330 |
| 3000 |  | 2 | 29 | 15 | 295 | \ | 129 |
| 3000 |  | 3 | 29 | 15 | 207 | 11 | 274 |
| 3000 |  | 4 | 25 | 15 |  | 9 | 263 |
| 3000 |  | 5 | 29 | 14 |  | 9 | 116 |

ב) הסתברות מוטציה:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mutation rate | iteration | Starting Point | "Bull's eye" | SELECTION = 2 "RWS" | SELECTION = 3 "Two Point CrossOver" | SELECTION =4 "Turnament" |
| 20% | 1 | 34 | 17 |  | 10 | 113 |
| 20% | 2 | 35 | 15 |  | 11 |  |
| % | 3 | 97 | 14 | 538 | 10 | 387 |
| 20% | 4 | 30 | 16 | 242 | 10 | 268 |
| 20% | 5 | 42 | 16 |  | 10 | 146 |
| 30% | 1 | 35 | 16 |  | 12 | 111 |
| 30% | 2 | 34 | 15 |  | 11 | 123 |
| 30% | 3 | 38 | 15 | 364 | 10 | 82 |
| 30% | 4 | 51 | 16 | 580 | 10 | 128 |
| 30% | 5 | 35 | 17 | 407 | 11 | 178 |
| 40% | 1 | 35 | 16 |  | 11 | 144 |
| 40% | 2 | 30 | 16 |  | 11 | 75 |
| 40% | 3 |  | 16 |  | 11 | 75 |
| 40% | 4 | 35 | 18 |  | 10 | 83 |
| 40% | 5 | 42 | 16 | 724 | 11 | 77 |
| 50% | 1 | 44 | 15 |  | 12 | 119 |
| 50% | 2 | 52 | 16 |  | 12 | 84 |
| 50% | 3 | 43 | 17 |  | 11 | 78 |
| 50% | 4 | 33 | 19 | 548 | 12 | 103 |
| 50% | 5 | 30 | 18 | 706 | 10 | 84 |

ג) פרופורציית האוכלסייה האיליסטית

ה) אסטרטגיית שיחלוף- בעיית N המלכות.

נראה שילובים של אסטרטגיות השיחלוף השונות:

|  |  |  |
| --- | --- | --- |
| ox | exchange | inversion |
| 1 | 49 | 49 |
| 2 | 60 | 77 |
| 3 | 885 | 11 |
| 4 | 401 | 140 |
| 5 | 261 | 73 |

|  |  |  |
| --- | --- | --- |
| pmx | exchange | inversion |
| 1 | 1548 | 1369 |
| 2 | 1576 | 826 |
| 3 | 2035 | 1960 |
| 4 | 2097 | 1740 |
| 5 | 1834 | 1863 |

מסקנה :

לאחר שביצענו את התרגיל עם מספר שיטות שחלוף , מוטציה , בחירה עם מספר פרמטרים משתנים כמו גודל אוכלוסיה , סיכוי למטוציה , אוכלוסיה עילית וכו' .

1. עבור בעיית המחרוזות הוספת היוריסטיקה של bulls eye הוספת ה Two point crossover שיפרה עוד יותר את האופטימליות ( גרף עמוד 11).
2. מסקנה

8. על מנת לפתור את הבעיה נרצה לבצע רזולוציה על בעיית האלגוריתם \ בשק)

נאמר כי כל איבר במערך הינו מספר פריטים כל ש:

GENE[i] = num of items from

יות שלנו במטרה ליצור גן תקין (שק תקין לגנב- עם פריטים שעומדים בתנאי שלנו)

את החישוב הדורי נעשה כעת בעזרת ממוצע המשקלים של כל השקים בדור. נעצור כאשר השינוי יקטן עד כדי קבוע שהוחלט מראש כשינוי המינימלי לעצירה ויצירת השק האופטימלי.