

מעבדה בבינה מלאכותית

דו"ח תרגיל בית 1 – Genetic Algorithms

תאריך הגשה: 31.3.

ד: 3194

ג: 3980

1. נוסף 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++)
    {
        // ... (code for calculating standard deviation) ...
    }
    return sqrt(sum);
}
```

חישוב סטיית תקן: נשתמש בנוסחאת סטיית התקן.

2. 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 2 מכמות ההרצות עם ההיוריסטיקה המקורית הגיעה למספר מקסימלי של איטרציות בלי להגיע למטרה. ידוע כי מוטציה מתרחשת בהסתברות נמוכה. ההיוריסטיקה שלנו נותנת יתרון ל שמקרבת אותנו למילת המטרה.

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

```

D:\Dropbox\University\Third Year\Second\AI LAB\hw1\ConsoleApplication1\Debug\ConsoleApplic...
Best: Hello!wprld! (3) Fitness Average: 20.6294 Standard deviation: 22.5255
Best: Hello!wprld! (3) Fitness Average: 18.917 Standard deviation: 22.341
Best: Hello!wprld! (3) Fitness Average: 17.4116 Standard deviation: 22.0177
Best: Hello!wprld! (3) Fitness Average: 15.7075 Standard deviation: 21.5052
Best: Hello!wprld! (2) Fitness Average: 15.7275 Standard deviation: 22.545
Best: Hello!wprld! (2) Fitness Average: 15.3862 Standard deviation: 23.2147
Best: Hello!wprld! (2) Fitness Average: 14.2397 Standard deviation: 22.4421
Best: Hello!wprld! (2) Fitness Average: 13.1392 Standard deviation: 21.6962
Best: Hello!wprld! (2) Fitness Average: 12.9595 Standard deviation: 22.3609
Best: Hello!wprld! (2) Fitness Average: 12.6504 Standard deviation: 22.216
Best: Hello!wprld! (1) Fitness Average: 12.1216 Standard deviation: 21.7126
Best: Hello!wprld! (1) Fitness Average: 11.6841 Standard deviation: 21.9942
Best: Hello!wprld! (1) Fitness Average: 11.8379 Standard deviation: 22.1123
Best: Hello!wprld! (1) Fitness Average: 11.5947 Standard deviation: 21.8143
Best: Hello!wprld! (1) Fitness Average: 11.4829 Standard deviation: 21.6445
Best: Hello!wprld! (1) Fitness Average: 11.6016 Standard deviation: 21.8772
Best: Hello!wprld! (1) Fitness Average: 12.1436 Standard deviation: 22.6603
Best: Hello!wprld! (1) Fitness Average: 12.5249 Standard deviation: 23.5605
Best: Hello!wprld! (1) Fitness Average: 11.0391 Standard deviation: 22.2968
Best: Hello!wprld! (1) Fitness Average: 11.0015 Standard deviation: 22.0964
Best: Hello!wprld! (1) Fitness Average: 11.4517 Standard deviation: 22.2276
Best: Hello!wprld! (1) Fitness Average: 11.6245 Standard deviation: 22.4329
Best: Hello!wprld! (1) Fitness Average: 11.7612 Standard deviation: 23.1317
Best: Hello!wprld! (0) Fitness Average: 10.3203 Standard deviation: 21.5438
Number OF Iteration: 47

D:\Dropbox\University\Third Year\Second\AI LAB\hw1\ConsoleApplication1\Debug\ConsoleApplic...
Best: Gdln!grslc (13) Fitness Average: 37.1006 Standard deviation: 22.5444
Best: Gdln!grslc (13) Fitness Average: 33.2568 Standard deviation: 22.1926
Best: Gdln!grslc (13) Fitness Average: 30.187 Standard deviation: 22.2463
Best: Ifkln!xprlc (10) Fitness Average: 27.4038 Standard deviation: 21.5502
Best: Ifkln!xprlc (9) Fitness Average: 25.6177 Standard deviation: 21.5825
Best: Ifkln!xprlc (8) Fitness Average: 24.6377 Standard deviation: 22.4346
Best: Ifkln!xprlc (7) Fitness Average: 22.3853 Standard deviation: 21.1659
Best: Ifkln!xprlc (6) Fitness Average: 22.0137 Standard deviation: 21.795
Best: Ifkln!xprlc (6) Fitness Average: 20.978 Standard deviation: 22.6237
Best: Ifkln!xprlc (5) Fitness Average: 19.8364 Standard deviation: 22.2897
Best: Ifkln!xprlc (5) Fitness Average: 18.4785 Standard deviation: 22.116
Best: Ifkln!xprlc (4) Fitness Average: 17.4663 Standard deviation: 22.2584
Best: Ifkln!xprlc (4) Fitness Average: 17.1318 Standard deviation: 22.3029
Best: Ifkln!xprlc (4) Fitness Average: 16.4058 Standard deviation: 22.6406
Best: Ifkln!xprlc (3) Fitness Average: 15.8584 Standard deviation: 22.6647
Best: Ifkln!xprlc (3) Fitness Average: 14.1475 Standard deviation: 21.5081
Best: Ifkln!xprlc (3) Fitness Average: 14.4868 Standard deviation: 22.4334
Best: Ifkln!xprlc (2) Fitness Average: 13.6021 Standard deviation: 21.4058
Best: Ifkln!xprlc (2) Fitness Average: 13.6416 Standard deviation: 22.1699
Best: Ifkln!xprlc (2) Fitness Average: 13.2173 Standard deviation: 21.8558
Best: Ifkln!xprlc (2) Fitness Average: 13.062 Standard deviation: 21.5498
Best: Hello!world! (1) Fitness Average: 12.8662 Standard deviation: 22.2203
Best: Hello!world! (1) Fitness Average: 12.0605 Standard deviation: 21.9337
Best: Hello!world! (0) Fitness Average: 11.7251 Standard deviation: 21.565
Number OF Iteration: 39

D:\Dropbox\University\Third Year\Second\AI LAB\hw1\ConsoleApplication1\Debug\ConsoleApplic...
Best: Hcklm xoolc! (10) Fitness Average: 30.1782 Standard deviation: 22.2185
Best: Idnln xoolc! (9) Fitness Average: 28.4023 Standard deviation: 23.4382
Best: Ifkln!voslc! (8) Fitness Average: 25.6743 Standard deviation: 22.7797
Best: Ifkln!voslc! (8) Fitness Average: 23.1455 Standard deviation: 22.0121
Best: Ifkln!voslc! (7) Fitness Average: 22.2944 Standard deviation: 22.9123
Best: Ifkln!voslc! (7) Fitness Average: 19.9756 Standard deviation: 21.4688
Best: Ifkln!voslc! (6) Fitness Average: 19.4507 Standard deviation: 22.0909
Best: Ifkln!voslc! (4) Fitness Average: 18.2109 Standard deviation: 21.6398
Best: Ifkln!voslc! (4) Fitness Average: 17.9238 Standard deviation: 22.0596
Best: Ifkln!voslc! (4) Fitness Average: 16.3555 Standard deviation: 21.4431
Best: Ifkln!voslc! (3) Fitness Average: 15.979 Standard deviation: 21.8883
Best: Ifkln!voslc! (3) Fitness Average: 15.2856 Standard deviation: 22.5148
Best: Ifkln!voslc! (3) Fitness Average: 14.3496 Standard deviation: 21.6263
Best: Ifkln!voslc! (2) Fitness Average: 15.1016 Standard deviation: 23.1397
Best: Ifkln!voslc! (2) Fitness Average: 13.0156 Standard deviation: 21.9098
Best: Ifkln!voslc! (2) Fitness Average: 13.021 Standard deviation: 22.1245
Best: Ifkln!voslc! (2) Fitness Average: 12.7407 Standard deviation: 21.9529
Best: Ifkln!voslc! (2) Fitness Average: 12.3149 Standard deviation: 21.7036
Best: Hello!world! (1) Fitness Average: 12.5542 Standard deviation: 21.9088
Best: Hello!world! (1) Fitness Average: 11.6367 Standard deviation: 21.6535
Best: Hello!world! (1) Fitness Average: 11.6948 Standard deviation: 21.2244
Best: Hello!world! (1) Fitness Average: 11.9907 Standard deviation: 21.9181
Best: Hello!world! (1) Fitness Average: 11.2178 Standard deviation: 21.8343
Best: Hello!world! (0) Fitness Average: 11.4355 Standard deviation: 22.1464
Number OF Iteration: 42
    
```

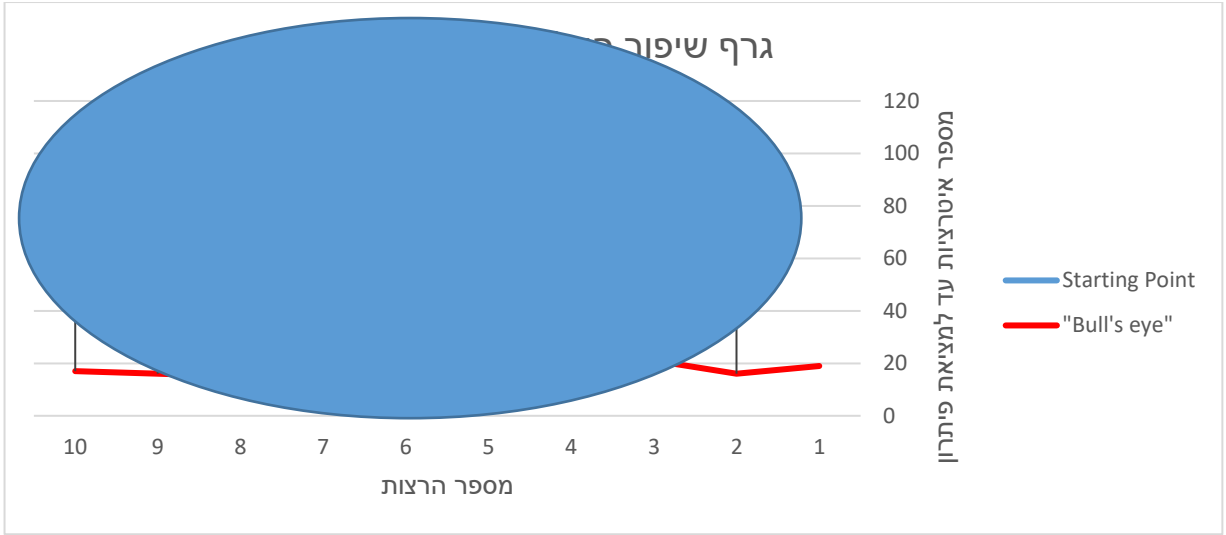
דוגמא 35 הרצות עם שינוי בהיוריסטיקה

```
D:\Dropbox\University\Third Year\Second\AI LAB\hw1\ConsoleApplication1\Debug\ConsoleApplic...
Best: tUgUo>'j1.l.q (801)Fitness Average: 1132.84 Standard deviation: 88.3895
Best: JehUo>'j1.l.q (682)Fitness Average: 1057.7 Standard deviation: 74.2181
Best: JehUo>'j1.l.q (682)Fitness Average: 996.461 Standard deviation: 72.693
Best: JehUo>'j1.l.q (682)Fitness Average: 937.016 Standard deviation: 71.6649
Best: emAo.birra! (612)Fitness Average: 879.5 Standard deviation: 72.8903
Best: 7enU\Jurmh! (596)Fitness Average: 823.969 Standard deviation: 72.2165
Best: H_hio>'erra! (544)Fitness Average: 767.653 Standard deviation: 71.1856
Best: Hxovo sirra! (471)Fitness Average: 715.133 Standard deviation: 75.4864
Best: JehUo>'wkuld! (398)Fitness Average: 661.419 Standard deviation: 79.5815
Best: 7ebvo wkuld! (344)Fitness Average: 603.951 Standard deviation: 81.3635
Best: HeZlg wkrlw! (289)Fitness Average: 547.844 Standard deviation: 80.4765
Best: Hello"worhj! (192)Fitness Average: 491.856 Standard deviation: 81.0927
Best: HenloAworld! (184)Fitness Average: 433.577 Standard deviation: 80.5341
Best: Hello"world! (62)Fitness Average: 380.239 Standard deviation: 81.1935
Best: Hello"world! (62)Fitness Average: 329.146 Standard deviation: 81.6841
Best: Hello world! (61)Fitness Average: 274.997 Standard deviation: 79.3709
Best: Hello world! (0)Fitness Average: 223.837 Standard deviation: 76.0822
Number OF Iteration: 16_

D:\Dropbox\University\Third Year\Second\AI LAB\hw1\ConsoleApplication1\Debug\ConsoleApplic...
Best: A'huy nZjy6+ (803)Fitness Average: 1135.38 Standard deviation: 86.2686
Best: ^unlo ujl5rD (723)Fitness Average: 1062.06 Standard deviation: 72.6695
Best: ^unlo uVu (696)Fitness Average: 1002.29 Standard deviation: 67.7487
Best: ^unlo uVu (696)Fitness Average: 948.947 Standard deviation: 65.8529
Best: Heklo uVu (535)Fitness Average: 895.833 Standard deviation: 68.1351
Best: Heklo uVu (535)Fitness Average: 843.769 Standard deviation: 71.929
Best: Heklo udnal (457)Fitness Average: 789.081 Standard deviation: 79.3048
Best: Hello'umt1P! (348)Fitness Average: 727.251 Standard deviation: 86.3233
Best: Heklo uorl# (248)Fitness Average: 662.622 Standard deviation: 92.4653
Best: Heklo uorl# (248)Fitness Average: 591.694 Standard deviation: 92.9403
Best: HKllo worl# (216)Fitness Average: 521.962 Standard deviation: 93.7651
Best: Hunlo worl! (205)Fitness Average: 452.91 Standard deviation: 87.5621
Best: Heklo wokld! (128)Fitness Average: 390.485 Standard deviation: 83.2019
Best: Hello worlh! (64)Fitness Average: 332.728 Standard deviation: 76.2191
Best: Hello worlc! (61)Fitness Average: 285.634 Standard deviation: 73.8646
Best: Hello worlc! (61)Fitness Average: 239.406 Standard deviation: 68.0685
Best: Hello world! (0)Fitness Average: 198.609 Standard deviation: 64.8852
Number OF Iteration: 16_

D:\Dropbox\University\Third Year\Second\AI LAB\hw1\ConsoleApplication1\Debug\ConsoleApplic...
Best: Mifl~UihYnZ1 (818)Fitness Average: 1132.96 Standard deviation: 86.0843
Best: ChoZgLnocdC (764)Fitness Average: 1062.09 Standard deviation: 71.344
Best: ChoZgLnocdC (764)Fitness Average: 1004.17 Standard deviation: 66.8592
Best: Zeotg1xjblp4 (707)Fitness Average: 949.246 Standard deviation: 66.2263
Best: Heotg1XRysd4 (669)Fitness Average: 898.348 Standard deviation: 69.3455
Best: Hegl<4stonP! (589)Fitness Average: 843.578 Standard deviation: 70.5072
Best: Ceol<4qorwJ! (542)Fitness Average: 788.107 Standard deviation: 74.4645
Best: Hegla sNrLZ+ (436)Fitness Average: 734.502 Standard deviation: 76.4382
Best: Hellr3wWp1H! (376)Fitness Average: 680.604 Standard deviation: 79.1592
Best: Hegla whsld2 (344)Fitness Average: 623.03 Standard deviation: 81.5824
Best: Hslllo w2hld! (265)Fitness Average: 564.271 Standard deviation: 84.1702
Best: Htglo wlrld! (203)Fitness Average: 501.64 Standard deviation: 86.766
Best: Hello wlrld! (140)Fitness Average: 439.517 Standard deviation: 85.7337
Best: Hello wlrld! (63)Fitness Average: 378.75 Standard deviation: 84.544
Best: Hello wlrld! (63)Fitness Average: 318.808 Standard deviation: 81.3792
Best: Hello world! (0)Fitness Average: 268.256 Standard deviation: 82.5112
Number OF Iteration: 15_
```

לאחר הרצות רבות- ניתן לראות את השינוי בהירוריסטיקה בעזרת גרף:



3. מינימום של פונקציה

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

```
struct ga_struct
{
    double x1,x2;           // the string
    double fitness;        // its fitness
};
```

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

פונקציית MathFuction(population[i]):

```
double MathFunction(ga_struct temp){
    return (20 + pow(temp.x1 , 2) + pow(temp.x2 , 2) -10*(cos(2 * PI * temp.x1) + cos(2 * PI * temp.x2) ) );
}
```

חישוב היוריסטיקה: חישוב ההיוריסטיקה הוא על פי הנוסחא שקיבלנו. נמצא זוג מספרים x1 ו-x2 שהוגרלו- נבדוק את ערך הפונקציה ונעדכן את ערך ה-fitness.

כמו כן, אנחנו מעוניינים בערך מינימלי ואנחנו יודעים ש

4. בעיית n המלכות:

ראשית נבצע שינוי בstruct:

קטע קוד:

```
struct ga_struct
{
    int QArray[NUMQ];           // the string
    unsigned int fitness;       // its fitness
};
```

NUMQ define של מספר המלכות.

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

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

קטע קוד:

פונקציית Mate:

קטע קוד: בחרנו להשתמש באלגוריתם שיטת השחלוף: O

הסבר:

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

נבחר מחצית מהאיברים מהאב (אדום)

8	2	4	3	7	5	1	0	9	6
4	1	7	6	2	8	3	9	5	0

7	2	4	8	3	5	1	9	0	6
---	---	---	---	---	---	---	---	---	---

פונקציית Mutate:

קטע קוד: בחרנו להשתמש במוטציה ...

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

Example: (1 2 **3** 4 **5** 6 7 8 9)
3rd and 5th selected randomly
new tour becomes (1 2 **5** 4 **3** 6 7 8 9)

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

....מתקיים, נבצע: fitnees++.

5.

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

קטע קוד:

.....

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

ב) נבצע אלגוריתם RWS: SELECTION = 2

נשים לב שעל מנת להפעיל את RWS

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

```
else if (SELECTION == 2) //RWS
{
    unsigned long place = rand() % RWSsum;
    i1 = RWS (population, place, (RWSmin+RWSmax) );

    buffer[i].str = population[i1].str;
}
```

הסבר:

בפונקציית mate:

חישבנו את כל ה-fitness של הדור - RWSsun.

כאשר נ

פונקציית RWS:

קטע קוד:

הסבר: ביצענו את פעולת הנירמול ובדיקה האם הסכום שיש לנו כרגע- נמצא /////

אחת הבעיות של RWS היא שב.....
הוספנו מנגנון במטרה לשפר את ביצועי 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

GRAPH

DR:

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
קטע קוד:

הסבר:

הוספנו 2 משתנים: LocationOne ו-LocationTwo: אלה מבצעים rand ובוחרים את שני האינדקסים האקראיים שביניהם 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

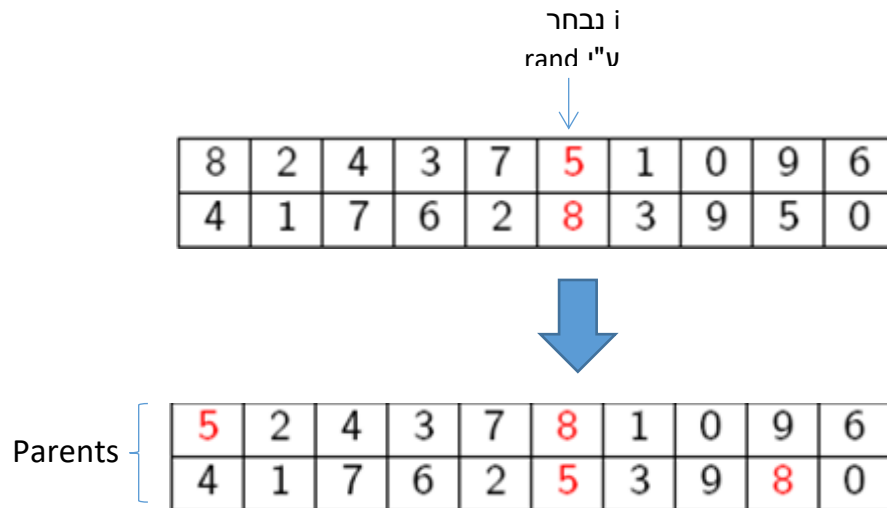
(ב) הוספת אסטרטגיית מוטציה ושיחלוף בבעית N המלכות:
שיחלוף: נשתמש בשיטת שיחלוף: PMX – Partially Matched crossover:

נוסיף את קטע הקוד לפונקציית Mutate:

קטע קוד:

```
else if ( MOTATECONFIGURATION == 2)
{
    if ( ipos < r )
    {
        int temp= ipos;
        ipos = r;
        r = temp;
    }
    for ( int z = r , k=ipos ; z < ipos ; z++,k--)
    {
        int temp = member.QArray[k];
        member.QArray[k] = member.QArray[z];
        member.QArray[z] = temp;
    }
}
```

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



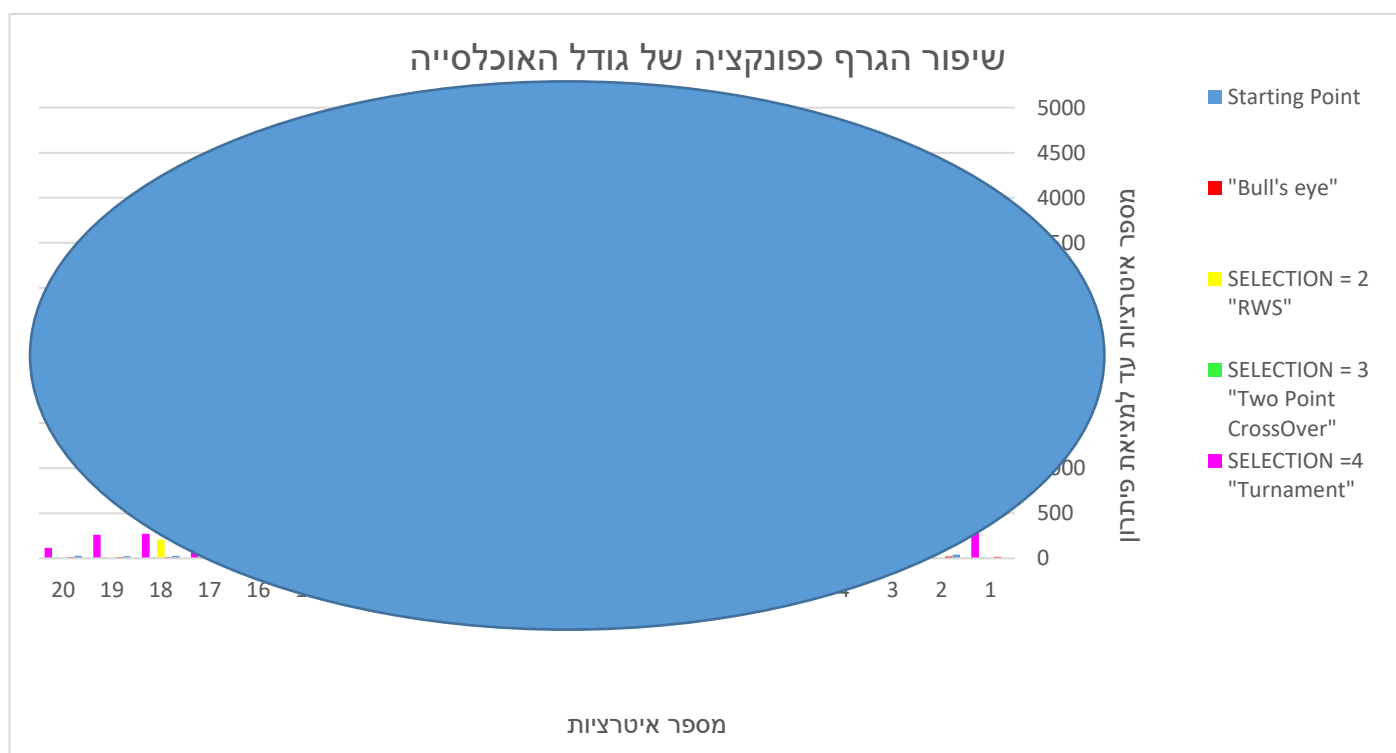
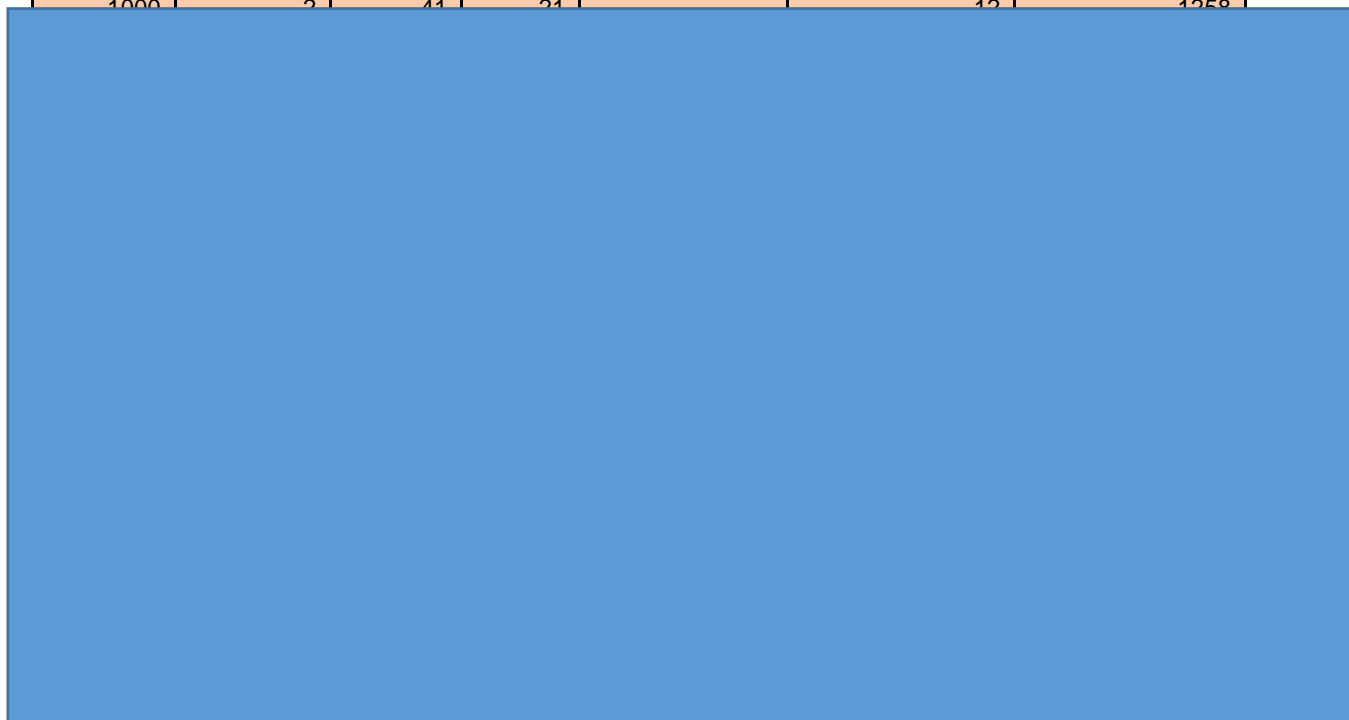
מוטציה: נשתמש בשיטת המוצטיה: מוטצית היפוך - inversion mutation:

קטע קוד:

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

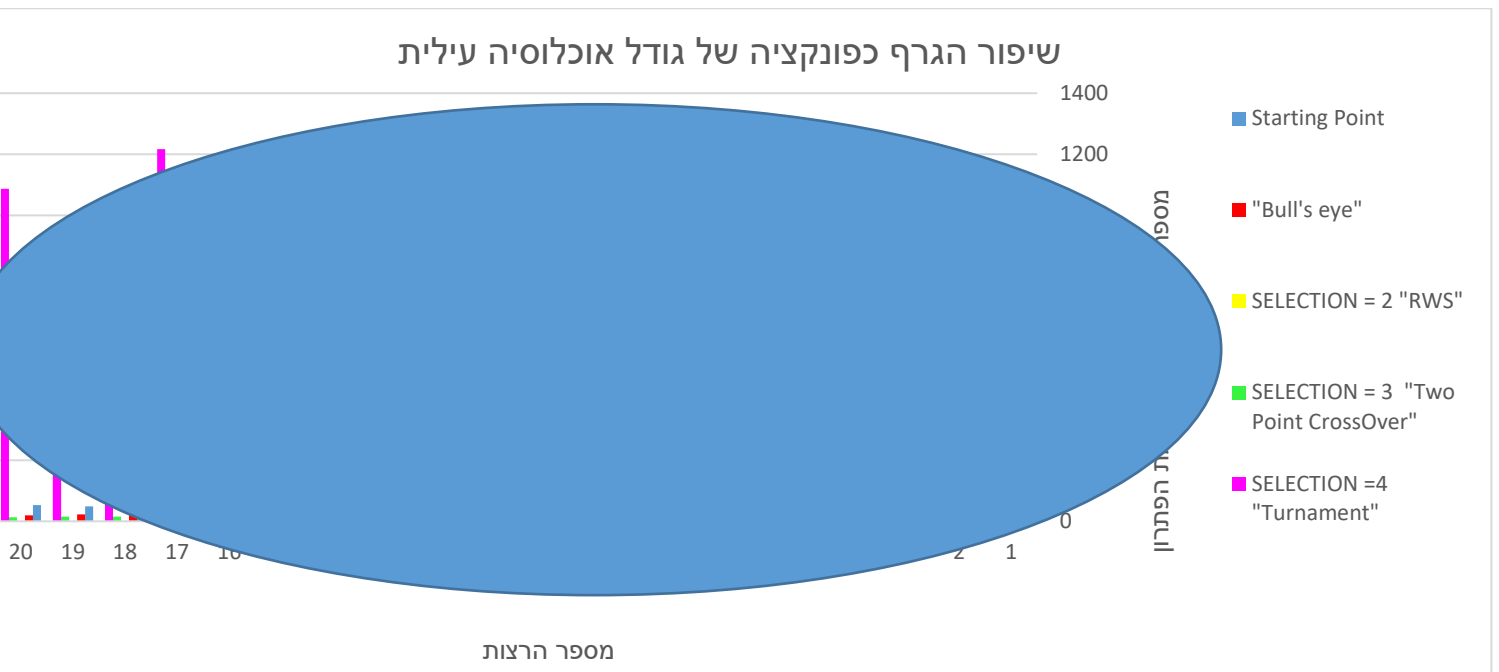
Example: (1 2 3 4 5 6 7 8 9)
(3 4 5) selected and inserted after 7
new tour becomes (1 2 6 7 5 4 3 8 9)

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



ב) הסתברות מוטציה:

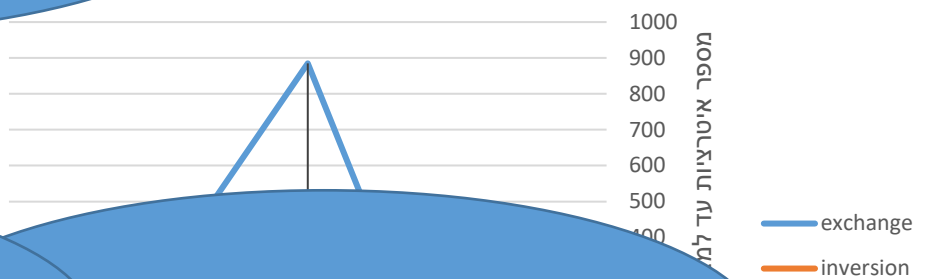
Mutation rate	iteration	Starting Point	"Bull's eye"	SELECTION = 2 "RWS"	SELECTION = 3 "Two Point CrossOver"	SELECTION =4 "Turnament"
20%	1	34	17		10	113
30%	2	35	15		11	113
40%	3	36	14		12	113
50%	4	37	13		13	113
60%	5	38	12		14	113
70%	6	39	11		15	113
80%	7	40	10		16	113
90%	8	41	9		17	113
100%	9	42	8		18	113
	10	43	7		19	113
	11	44	6		20	113
	12	45	5		21	113
	13	46	4		22	113
	14	47	3		23	113
	15	48	2		24	113
	16	49	1		25	113
	17	50			26	113
	18	51			27	113
	19	52			28	113
	20	53			29	113
	21	54			30	113
	22	55			31	113
	23	56			32	113
	24	57			33	113
	25	58			34	113
	26	59			35	113
	27	60			36	113
	28	61			37	113
	29	62			38	113
	30	63			39	113
	31	64			40	113
	32	65			41	113
	33	66			42	113
	34	67			43	113
	35	68			44	113
	36	69			45	113
	37	70			46	113
	38	71			47	113
	39	72			48	113
	40	73			49	113
	41	74			50	113
	42	75			51	113
	43	76			52	113
	44	77			53	113
	45	78			54	113
	46	79			55	113
	47	80			56	113
	48	81			57	113
	49	82			58	113
	50	83			59	113
	51	84			60	113
	52	85			61	113
	53	86			62	113
	54	87			63	113
	55	88			64	113
	56	89			65	113
	57	90			66	113
	58	91			67	113
	59	92			68	113
	60	93			69	113
	61	94			70	113
	62	95			71	113
	63	96			72	113
	64	97			73	113
	65	98			74	113
	66	99			75	113
	67	100			76	113
	68	101			77	113
	69	102			78	113
	70	103			79	113
	71	104			80	113
	72	105			81	113
	73	106			82	113
	74	107			83	113
	75	108			84	113
	76	109			85	113
	77	110			86	113
	78	111			87	113
	79	112			88	113
	80	113			89	113
	81	114			90	113
	82	115			91	113
	83	116			92	113
	84	117			93	113
	85	118			94	113
	86	119			95	113
	87	120			96	113
	88	121			97	113
	89	122			98	113
	90	123			99	113
	91	124			100	113
	92	125			101	113
	93	126			102	113
	94	127			103	113
	95	128			104	113
	96	129			105	113
	97	130			106	113
	98	131			107	113
	99	132			108	113
	100	133			109	113
	101	134			110	113
	102	135			111	113
	103	136			112	113
	104	137			113	113
	105	138			114	113
	106	139			115	113
	107	140			116	113
	108	141			117	113
	109	142			118	113
	110	143			119	113
	111	144			120	113
	112	145			121	113
	113	146			122	113
	114	147			123	113
	115	148			124	113
	116	149			125	113
	117	150			126	113
	118	151			127	113
	119	152			128	113
	120	153			129	113
	121	154			130	113
	122	155			131	113
	123	156			132	113
	124	157			133	113
	125	158			134	113
	126	159			135	113
	127	160			136	113
	128	161			137	113
	129	162			138	113
	130	163			139	113
	131	164			140	113
	132	165			141	113
	133	166			142	113
	134	167			143	113
	135	168			144	113
	136	169			145	113
	137	170			146	113
	138	171			147	113
	139	172			148	113
	140	173			149	113
	141	174			150	113
	142	175			151	113
	143	176			152	113
	144	177			153	113
	145	178			154	113
	146	179			155	113
	147	180			156	113
	148	181			157	113
	149	182			158	113
	150	183			159	113
	151	184			160	113
	152	185			161	113
	153	186			162	113
	154	187			163	113
	155	188			164	113
	156	189			165	113
	157	190			166	113
	158	191			167	113
	159	192			168	113
	160	193			169	113
	161	194			170	113
	162	195			171	113
	163	196			172	113
	164	197			173	113
	165	198			174	113
	166	199			175	113
	167	200			176	113
	168	201			177	113
	169	202			178	113
	170	203			179	113
	171	204			180	113
	172	205			181	113
	173	206			182	113
	174	207			183	113
	175	208			184	113
	176	209			185	113
	177	210			186	113
	178	211			187	113
	179	212			188	113
	180	213			189	113
	181	214			190	113
	182	215			191	113
	183	216			192	113
	184	217			193	113
	185	218			194	113
	186	219			195	113
	187	220			196	113
	188	221			197	113
	189	222			198	113
	190	223			199	113
	191	224			200	113
	192	225			201	113
	193	226			202	113
	194	227			203	113
	195	228			204	113
	196	229			205	113
	197	230			206	113
	198	231			207	113
	199	232			208	113
	200	233			209	113
	201	234			210	113
	202	235			211	113
	203	236			212	113
	204	237			213	113
	205	238			214	113
	206	239			215	113
	207	240			216	113
	208	241			217	113
	209	242			218	113
	210	243			219	113
	211	244			220	113
	212	245			221	113
	213	246			222	113
	214	247			223	113
	215	248			224	113
	216	249			225	113
	217	250			226	113
	218	251			227	113
	219	252			228	113
	220	253			229	113
	221	254			230	113
	222	255			231	113
	223	256			232	113
	224	257			233	113
	225	258			234	113
	226	259			235	113
	227	260			236	113
	228	261			237	113
	229	262			238	113
	230	263			239	113
	231	264			240	113
	232	265			241	113
	233	266			242	113
	234	267			243	113
	235	268			244	113
	236	269			245	113
	237	270			246	113
	238	271			247	113
	239	272			248	113
	240	273			249	113
	241	274			250	113
	242	275			251	113
	243	276			252	113
	244	277			253	113
	245	278			254	113
	246	279			255	113
	247	280			256	113
	248	281			257	113
	249	282			258	113
	250	283			259	113
	251	284			260	113
	252	285			261	113
	253	286			262	113
	254	287			263	113
	255	288			264	113
	256	289			265	113
	257	290			266	113
	258	291			267	113
	259	292			268	113
	2					



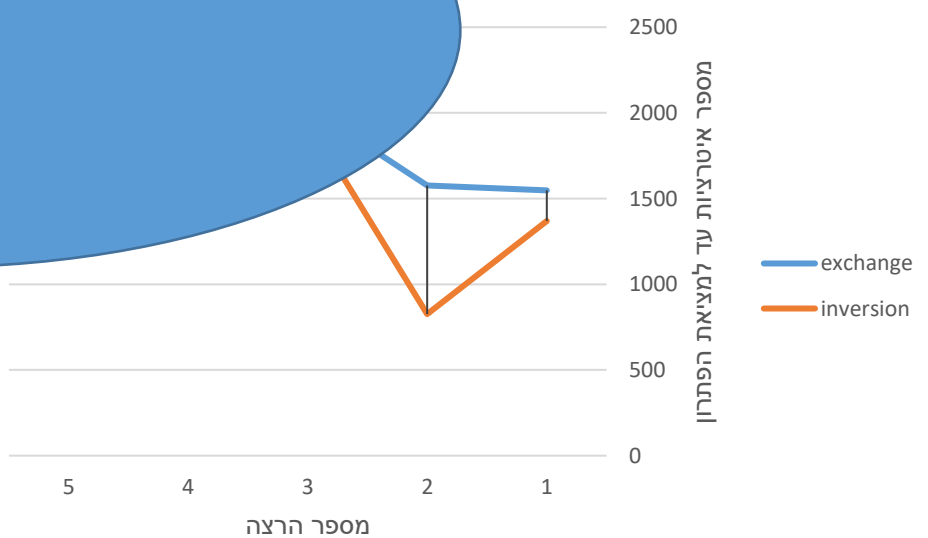
ה) אסטרטגיית שיחלוף- בעיית N המלכות.
נראה שילובים של אסטרטגיות השיחלוף השונות:

ox	exchange	inversion
1	49	49

OX שיפור אסטרטגיות השחלוף כפונקציה של מספר הרצה



pmx	exchange	inversion
1	1548	1260



מסקנה:

לאחר שביצענו את התרגיל עם מספר שיטות שחלוף, מצינו כי ישנה חשיבות רבה לשיטות אלו, וכל אחד מהם יכול להוות פתרון יעיל.

1. עבור בעיית המחרוזות הוספת היותו של האות 'א' או 'ב' לאות 'א' או 'ב' (11).
2. מסקנה

8. על מנת לפתור את הבעיה נרצה לבדוק את כל האפשרויות, נאמר כי כל איבר במערך הינו מספר שלם.

יות שלנו במטרה ליצור גן תקין (שק) וזהו הפתרון הנכון. את החישוב הדורי נעשה כעת בעזרת פונקציה אשר תחשב את הערך של כל איבר במערך ויצירת השק האופטימלי.