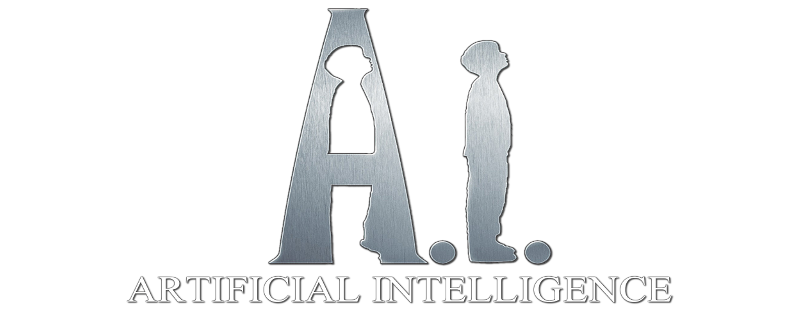
מעבדה בבינה מלאכותית, תשע"ח

פרויקט #2 – גנטיקה.

**מגישים:**

באסל סעד \*\*\*\*\*\*\*\*\*

עלי ספדי \*\*\*\*\*\*\*\*\*\*



**This Project can be found at:**

[**https://github.com/BaselSD/artificial-Intel-18/tree/master/Genetics**](https://github.com/BaselSD/artificial-Intel-18/tree/master/Genetics)

**היררכיה:**

1. **בעיית המחרוזת:**
   1. **סקירת פונקציית הפיטנס.**
   2. **ניסויים לבחירת טכניקה הכי טובה והשוואות בין שתי פונקציות הפיטנס (בול פגיעה והפונקציה המקורית).**
   3. **ניסויים – השפעת פרמטרים שונים על הביצועים של המנוע (בשימוש בפונקציית בול-פגיעה).**
2. **בעיית N-Queens:**
   1. **סקירת פונקציית הפיטנס.**
   2. **ניסויים לבחירת טכניקה הכי טובה.**
   3. **ניסויים – השפעת פרמטרים שונים על הביצועים של המנוע.**
3. **השוואת ביצועים בין המנוע הגנטי לבין מנוע ה- Minimal-Conflicts.**
4. **סקירת והשוואת ביצועי המנוע הגנטי בפתירת בעיית השק מול התוצאות הנתונות.**
5. **בעיית המחרוזות – Hello world! :**

**פונקציית בול-פגיעה:**

מימשנו את הפונקציה בצורה שתתן ערך טוב ברגע שיש "פגיעה" וערך טוב מאד ברגע שיש "בול".   
אלגוריתם:

1. עבור כל האותיות שבמחרוזת:
   1. אם האות הוא בול כמו שבמחרוזת המטרה:
      1. תגדיל את ערך "מספר הבולז" ב- 1.
      2. תסמן במערך האותיות שהאות הזאת כבר חושבה.
   2. אם האות היא פגיעה:
      1. תגדיל את ערך "מספר הפגיעות" ב- 1.
      2. תסמן במערך האותיות שהאות הזאת כבר חושבה.
   3. אחרת:
      1. תגריל מספר בין 0 ל גודל מחרוזת המטרה, תכפיל אותו בהפרש בין האות הזאות לבין האות שבמחרוזת המטרה ותשמור את התוצאה במשתנה "הניקוד".
2. תן למחרוזת ערך:
3. אם המחרוזת שונה ממחרוזת המטרה בלכל היותר 3 אותיות, תחליף לה את ערך הפיטנס למספר האותיות השונים.

כפי שרואים, הפונקציה נוטה לעשות מינימיזציה ככל שהמחרוזת קרובה למחרוזת המטרה. ולחילופין, ממקסמת את הערך יותר ככל שהמחרוזת רחוקה יותר.

**ניסויים לבחירת הטכניקה הכי טובה:**

כנדרש, הוספנו כמה דברים למנוע:

1. פונקציית בול-פגיעה.
2. שחלוף דו נקודתי, שחלוף יוניפורמי.
3. בחירת רולטה, בחירת טורניר.
4. Aging.

ביצענו כמות אדירה של ניסויים על מנת לנסות להגיע לטכניקה הכי טובה. כמו כן,

להלן הניסויים:

**השוואה בין טכניקות השחלוף (הניסוי בוצע עם אלגוריתם בחירה רנדומלית ובשימוש בפונקציית בול-פגיעה).**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Random selection – single point crossover – Bull's eye fitness | | | | | | |
| End heuristic deviation | End heuristic average | Start heuristics deviation | Start heuristic average | Generated generations | Total time  (Seconds) | Tries |
| 209 | 206 | 2360 | 2292 | 42 | 0.59 | 1 |
| 230 | 224 | 686 | 2436 | 36 | 0.5 | 2 |
| 206 | 206 | 675 | 2431 | 44 | 0.65 | 3 |
| 202 | 207 | 682 | 2431 | 43 | 0.63 | 4 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Random selection – two crossover – Bull's eye fitness | | | | | | |
| End heuristic deviation | End heuristic average | Start heuristics deviation | Start heuristic average | Generated generations | Total time  (Seconds) | Tries |
| 215 | 217 | 669 | 2396 | 33 | 0.44 | 1 |
| 230 | 215 | 657 | 2425 | 39 | 0.58 | 2 |
| 127 | 170 | 659 | 2423 | 45 | 0.67 | 3 |
| 210 | 210 | 667 | 2414 | 35 | 0.5 | 4 |
| 231 | 218 | 671 | 2429 | 35 | 0.58 | 5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Random selection – uniform crossover – Bull's eye fitness | | | | | | |
| End heuristic deviation | End heuristic average | Start heuristics deviation | Start heuristic average | Generated generations | Total time  (Seconds) | Tries |
| 223 | 220 | 664 | 2399 | 28 | 0.64 | 1 |
| 206 | 205 | 673 | 2454 | 33 | 0.77 | 2 |
| 231 | 220 | 691 | 2414 | 28 | 0.66 | 3 |
| 218 | 210 | 675 | 2415 | 30 | 0.72 | 4 |
| 226 | 227 | 681 | 2431 | 27 | 0.61 | 5 |

אפשר לראות שטכניקת ה- **two points crossover** הניבה ביצועי זמן הכי טובים אבל טכניקת ה- **uniform crossover** הניבה ביצועים יותר טובים מבחינת כמות הדורות שהמנוע יצר עד שהתכנס לפתרון. מבחינת ממוצע וסטיית תקן, הביצועים לא היו שונים במיוחד.

**השוואה בין טכניקות הבחירה בשתי פונקציות הפיטנס – (בול-פגיעה והפונקציה המקורית). כמו כן, הגיל המקסימלי שמותר להגיע אליו הוא 2 – בהמשך נתייחס לפרמטר הזה.**

1. **פונקציית בול-פגיעה:**

|  |  |  |
| --- | --- | --- |
| Roulette selection – single point crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| 480 | 6.6 | 1 |
| 390 | 5.3 | 2 |
| 332 | 4.4 | 3 |

|  |  |  |
| --- | --- | --- |
| Tournament selection – single point crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| 32 | 1.4 | 1 |
| 29 | 1.3 | 2 |
| 74 | 2.2 | 3 |

|  |  |  |
| --- | --- | --- |
| Roulette selection – two points crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| 273 | 4.2 | 1 |
| 337 | 5.7 | 2 |
| 426 | 6.0 | 3 |

|  |  |  |
| --- | --- | --- |
| Tournament selection – two points crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| 30 | 1.3 | 1 |
| 26 | 1.1 | 2 |
| 26 | 1.2 | 3 |

|  |  |  |
| --- | --- | --- |
| Roulette selection – uniform crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| 327 | 7.6 | 1 |
| 266 | 6.1 | 2 |
| 414 | 9.6 | 3 |

|  |  |  |
| --- | --- | --- |
| Tournament selection – uniform crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| **18** | **1.0** | **1** |
| **21** | **1.1** | **2** |
| **19** | **1.0** | **3** |

אפשר לראות שטכניקת הבחירה **Tournament selection** הניבה את הביצועים הכי טובים בכללי מבחינת זמן וכמות הדורות שמנוע יצר עד שהגיע לפתרון בשימוש בטכניקת השחלוף - **uniform crossover.**

ה- **Roulette selection**  הניב ביצועים טובים יחסית עם טכניקת השחלוף -  **two points crossover.**

1. **הפונקציה המקורית :**

|  |  |  |
| --- | --- | --- |
| Roulette selection – single point crossover – aging 40 | | |
| Generated generations | Total time  (Seconds) | Tries |
| 26 | 0.3 | 1 |
| 25 | 0.3 | 2 |
| 26 | 0.3 | 3 |
| Roulette selection – two points crossover – aging 40 | | |
| Generated generations | Total time  (Seconds) | Tries |
| **23** | **0.3** | **1** |
| **24** | **0.3** | **2** |
| **20** | **0.3** | **3** |

|  |  |  |
| --- | --- | --- |
| Tournament selection – single point crossover – aging 40 | | |
| Generated generations | Total time  (Seconds) | Tries |
| 21 | 0.9 | 1 |
| 23 | 1.0 | 2 |
| 19 | 0.8 | 3 |
| Tournament selection – two points crossover – aging 40 | | |
| Generated generations | Total time  (Seconds) | Tries |
| 19 | 0.9 | 1 |
| 19 | 0.8 | 2 |
| 21 | 0.9 | 3 |

|  |  |  |
| --- | --- | --- |
| Roulette selection – uniform crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| 15 | 0.3 | 1 |
| 18 | 0.4 | 2 |
| 28 | 0.6 | 3 |

|  |  |  |
| --- | --- | --- |
| Tournament selection – uniform crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| **13** | **0.7** | **1** |
| **13** | **0.7** | **2** |
| **15** | **0.8** | **3** |

פונקציית הפיטנס המסופקת הניבה ביצועים יותר טובים מפונקציית בול-פגיעה וכמו כן – אפשר לראות שאלגוריתם הבחירה RWS הניב ביצועי זמן יותר טובים מאלגוריתם הטורניר ואילו אלגוריתם הטורניר הניב ביצועים יותר טובים מבחינת כמות הדורות.

ולכן, לפי התוצאות שקיבלנו הטכניקות הכי טובות עד כאן הן:

1. **.Tournament selection – uniform crossover**
2. **Roulette selection – two points crossover.**

**השוואה בין הטכניקות עם אחוזי מוטציות שונים.**

**MUTATION RATE = 0.5:**

**הפונקציה המקורית:**

|  |  |  |
| --- | --- | --- |
| Roulette selection – two points crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| 19 | 0.18 | 1 |
| 21 | 0.2 | 2 |
| 23 | 0.2 | 3 |

|  |  |  |
| --- | --- | --- |
| Tournament selection k = 100 – uniform crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| 14 | 0.2 | 1 |
| 16 | 0.2 | 2 |
| 15 | 0.2 | 3 |

**פונקציית בול-פגיעה:**

|  |  |  |
| --- | --- | --- |
| Roulette selection – two points crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| 1526 | 15.8 | 1 |
| 414 | 3.9 | 2 |
| 1074 | 10.9 | 3 |

|  |  |  |
| --- | --- | --- |
| Tournament selection k = 100 – uniform crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| 43 | 0.8 | 1 |
| 56 | 0.9 | 2 |
| 189 | 3.3 | 3 |

**----------------------------------------------------------------------------**

**MUTATION RATE = 0.7:**

**הפונקציה המקורית:**

|  |  |  |
| --- | --- | --- |
| Roulette selection – two points crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| 20 | 0.2 | 1 |
| 16 | 0.1 | 2 |
| 17 | 0.1 | 3 |

|  |  |  |
| --- | --- | --- |
| Tournament selection k = 100 – uniform crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| 17 | 0.2 | 1 |
| 18 | 0.3 | 2 |
| 15 | 0.2 | 3 |

**פונקציית בול-פגיעה:**

|  |  |  |
| --- | --- | --- |
| Roulette selection – two points crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| 247 | 2.4 | 1 |
| 115 | 1.2 | 2 |
| 205 | 2.1 | 3 |

|  |  |  |
| --- | --- | --- |
| Tournament selection k = 100 – uniform crossover | | |
| Generated generations | Total time  (Seconds) | Tries |
| **36** | **0.6** | **1** |
| **30** | **0.5** | **2** |
| **27** | **0.49** | **3** |

כמו שרואים פה כמו כן, טכניקת הטורניר הניבה ביצועים יותר טובים. בכלל כאשר העלינו את אחוז המוטציות הביצועים השתפרו.

בנוסף לכל הניסויים שהזכרנו, ביצענו ניסויים מקיפים בשימוש בפרמטרים שנותרו ובסופו של דבר הגענו למסקנה שטכניקת הטורניר יותר טובה מטכניקת ה- RWS.

הערה

**בהמשך נבדוק את השפעת הפרמטרים על תפקוד המנוע.**

**ניסויים לבדיקת השפעת הפרמטרים השונים על ביצועי המנוע בשימוש בפונקציית בול-פגיעה.**

**POP SIZE:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tournament selection k = 100 – uniform crossover – MUTATION .25 | | | 1024 |
| Generated generations | Total time  (Seconds) | Tries |
| 272 | 4.83 | 1 |
| 2775 | 47.1 | 2 |
| 318 | 5.67 | 3 |
| Tournament selection k = 100 – uniform crossover MUTATION .25 | | | 2048 |
| Generated generations | Total time  (Seconds) | Tries |
| 28 | 0.83 | 1 |
| 55 | 1.63 | 2 |
| 472 | 13.76 | 3 |
| Tournament selection k = 100 – uniform crossover MUTATION .25 | | | 4096 |
| Generated generations | Total time  (Seconds) | Tries |
| **31** | **1.81** | **1** |
| **15** | **0.85** | **2** |
| **25** | **1.50** | **3** |

רואים שככל שגודל האוכלוסייה גדל, הביצועים נהיים יותר טובים. אבל זה תחת אחוז מוטציה של 0.25. בהמשך נראה שההפך דווקא הוא נכון כאשר אחוז המוטציה גדל.

**Mutation-Rate**

|  |  |  |  |
| --- | --- | --- | --- |
| Tournament selection k = 100 – uniform crossover – POPSIZE - 1024 | | | 0.25 |
| Generated generations | Total time  (Seconds) | Tries |
| 272 | 4.83 | 1 |
| 2775 | 47.1 | 2 |
| 318 | 5.67 | 3 |
| Tournament selection k = 100 – uniform crossover - POPSIZE - 1024 | | | 0.5 |
| Generated generations | Total time  (Seconds) | Tries |
| 40 | 0.72 | 1 |
| 41 | 0.75 | 2 |
| 33 | 0.60 | 3 |
| Tournament selection k = 100 – uniform crossover- POPSIZE - 1024 | | | 0.7 |
| Generated generations | Total time  (Seconds) | Tries |
| **16** | **0.27** | **1** |
| **31** | **0.55** | **2** |
| **32** | **0.58** | **3** |

**(את הניסוי הזה כמו כן עשינו עם גודל אוכלוסייה של 4096, אבל התוצאות היו פחות טובות ולכן הגענו למסקנה שככל שגודל האוכלוסייה גדל תחת אחוז מוטציה של 0.7, הביצועים יהיו פחות טובים.)**

אפשר לראות שככל שאחוז המוטציה גדל יותר, זמן ההתכנסות לפתרון נעשה קצר יותר וכמות הדורות המיוצרים קטנה יותר. זה נכון מהעובדה שהמוטציה תורמת לרנדומליות - מה שמגדיל את הסיכוי להגיע לפתרון בזמן יותר קצר.

**ELITISIM RATE:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tournament selection k = 100 – uniform crossover -POPSIZE – 1024 – MUTATION-RATE – 0.7 | | | 0.1 |
| Generated generations | Total time  (Seconds) | Tries |
| **16** | **0.27** | **1** |
| **31** | **0.55** | **2** |
| **32** | **0.58** | **3** |
| Tournament selection k = 100 – uniform crossover-POPSIZE – 1024 – MUTATION-RATE – 0.7 | | | 0.3 |
| Generated generations | Total time  (Seconds) | Tries |
| 29 | 0.48 | 1 |
| 63 | 1.04 | 2 |
| 27 | 0.4 | 3 |
| Tournament selection k = 100 – uniform crossover-POPSIZE – 1024 – MUTATION-RATE – 0.7 | | | 0.5 |
| Generated generations | Total time  (Seconds) | Tries |
| 41 | 0.59 | 1 |
| 53 | 0.74 | 2 |
| 38 | 0.54 | 3 |

אפשר לראות שככל שהגדלנו את כמות הגנים האליטיסטים, ככל שהביצועים ירדו וזה קורה בגלל שבמקרה זה, המון גנים לא רצויים עוברים מדור לדור וזה לא מתאים לנו.

**סיכום בעיית המחרוזת – בול-פגיעה:**

* **הטכניקה הכי טובה על פי הניסויים שביצענו:**

Tournament selection: k = 100

Uniform crossover

Population-size: 1024

Max-age: 2

Mutation rate: 0.7

**Elitism rate: 0.1**

* **פונקציית בול פגיעה היא פחות טובה מהפונקציה הנתונה וזה מכיוון שהיא פחות כללית ומצ'פרת מקרים יותר ספציפיים. לעומת זאת, הפונקציה הנתונה נותנת ערך בהתחשב בתמונה הכללית – המרחק הכולל של האותיות ממחרוזת המטרה.**
* **כאשר אחוז המוטציה קטן: ככל שגודל האוכלוסייה גדל, ככל שיש פיזור יותר רחב של האלמנטים, ככל שהסיכוי להגיע לפתרון בשלב מוקדם יותר גדל. אבל מאד חשוב להדגיש – כאשר אחוז המוטציה קטן, האלמנטים לא משתנים הרבה אבל, עצם העובדה שיש המון אלמנטים – זה דואג ליצירת גוונים וסוגים שונים של אלמנטים.**
* **כאשר אחוז המוטציה גדול: עדיף דווקא לעבוד עם כמות אוכלוסייה יותר קטנה וזה בגלל שהמוטציה מפתחת אופקים וגוונים חדשים. אם ישנה אוכלוסייה גדולה, זה עלול להוביל לזמן ריצה ארוך בגלל הכמות הגדולה של האפשריות.**
* **עדיף להקדיש מקום לא גדול לגנים האלסטיים, על מנת לתת אפשרות לגנים חדשים להתפתח ולהוביל בדורות העתידיים. כמו כן, אם מעבירים הרבה גנים מדור לדור, גנים לא רצויים עלולים לעבור באיזשהו שלב.**
* **לגבי הגיל המקסימלי של גן מסוים, תחת התנאים שהצבנו, הביצועים הכי טובים שקיבלנו היו כאשר הגיל המקסימלי היה קטן יחסית. זה קרה מהעובדה שגיל קטן מונע חזרות ומכריח את הגנים להמשיך להתחדש מדור לדור.**
* **בשימוש ב-פונקציה המקורית, מצאנו כי ה- RWS נותן ביצועי מהירות יותר טובים ואילו ה- טורניר נותן פחות שימוש בזיכרון (פחות דורות).**

1. **בעיית השק**

כדי לפתור את הבעיה הזו, בנינו מנוע גנטי חדש עם כמה שינויים.

* הגדרנו את פונקציית הפיטנס להיות ערך הרווח עבור כל גן. כמו כן, אם גן מסוים עבר את המשקל המקסימלי של השק, הפונקציה תדאג להוריד ממנו אחדים עד שהמשקל הכולל יהפוך לחוקי.
* את הייצוג של הבעיה עשינו באמצעות מחרוזת בינארית כאשר 1 במקום ה- i מסמל שהפריט ה- i נמצא בתוך השק, ו 0 מסמל שהפריט לא נמצא.
* שינינו את אלגוריתם הבחירה כך שהגן הכי טוב באוכלוסייה נבחר **תמיד** להיות אב ואילו האב השני נבחר באמצעות אלגוריתם הטורניר.
* מימשנו את המוטציה בצורה כזאת שבה אנחנו בוחרים פריט **רנדומלי שלא נמצא בשק** ומכניסים אותו. כלומר בוחרים מקום במחרוזת שמכיל אפס ומשנים ל 1.
* מכיוון שלא תמיד אפשר למצוא את הפתרון האופטימלי, ניסינו כמה וכמה תנאי עצירה ובסוף בחרנו בשניים.  
  + SC-1

אם הגענו לפתרון האופטימלי הידוע מראש, אז נעצור.  
בעצם על מנת להשתמש בתנאי זה, אנחנו חייבים לדעת את הפתרון האופטימלי מראש. **אבל בכל זאת; אם הפתרון האופטימלי לא נתון לנו, נוכל לחילופין להכניס חסם עליון יותר גדול (למשל – משקל הפריט הכי יקר כפול מספר הפריטים הכולל).**

* + SC-2

התנאי עצירה השני שמימשנו היה – אם הנפח שנשאר פנוי בשק הוא יותר קטן ממשקל הפריט הכי קל **וגם** ערך הפיטנס הנוכחי הוא המקסימלי לאורך כל הדורות עד כה, אז נעצור.  
תנאי זה בעצם הוא סוג של ניחוש מושכל, הרי אם אין עוד שטח פנוי בשק אז יש סיכוי שהרווח מאד קרוב לפתרון האופטימלי, וכמו כן התנאי השני דואג לקרב את הפתרון עוד טיפה.

* ניסינו להריץ את המנוע על הבעיות שבלינק, ואלו היו התוצאות:

Sc-1 = Stop condition – 1

Sc-2 = Stop condition – 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Optimal Solution | SC-2  Max-Profit +  Representation | SC-1  Max-Profit +  Representation | Experiment | Problem |
| **309**  **1111010000** | **1111010000**  **(309)** | **1111010000**  **(309)** | 1 | 1 |
| **1111010000**  **(309)** | **1111010000**  **(309)** | 2 |
| **1111010000**  **(309)** | **1111010000**  **(309)** | 3 |
| **1111010000**  **(309)** | **1111010000**  **(309)** | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Optimal Solution | SC-2  Max-Profit +  Representation | SC-1  Max-Profit +  Representation | Experiment | Problem |
| **51**  **01110** | **01110**  **(51)** | **01110**  **(51)** | 1 | 2 |
| **01110**  **(51)** | **01110**  **(51)** | 2 |
| **01110**  **(51)** | **01110**  **(51)** | 3 |
| **01110**  **(51)** | **01110**  **(51)** | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Optimal Solution | SC-2  Max-Profit +  Representation | SC-1  Max-Profit +  Representation | Experiment | Problem |
| **150**  **110010** | **110010**  **(150)** | **110010**  **(150)** | 1 | 3 |
| **110010**  **(150)** | **110010**  **(150)** | 2 |
| **110010**  **(150)** | **110010**  **(150)** | 3 |
| **110010**  **(150)** | **110010**  **(150)** | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Optimal Solution | SC-2  Max-Profit +  Representation | SC-1  Max-Profit +  Representation | Experiment | Problem |
| **900**  **10111011** | **10111011**  **(900)** | **10111011**  **(900)** | 1 | 5 |
| **10111011**  **(900)** | **10111011**  **(900)** | 2 |
| **10111011**  **(900)** | **10111011**  **(900)** | 3 |
| **10111011**  **(900)** | **10111011**  **(900)** | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Optimal Solution | SC-2  Max-Profit +  Representation | SC-1  Max-Profit +  Representation | Experiment | Problem |
| **107**  **1001000** | **1001000**  **(107)** | **1001000**  **(107)** | 1 | 4 |
| **1001000**  **(107)** | **1001000**  **(107)** | 2 |
| **1001000**  **(107)** | **1001000**  **(107)** | 3 |
| **1001000**  **(107)** | **1001000**  **(107)** | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Optimal Solution | SC-2  Max-Profit +  Representation | SC-1  Max-Profit +  Representation | Experiment | Problem |
| **1735**  **0101001** | **0101001**  **(1735)** | **0101001**  **(1735)** | 1 | 6 |
| **0101001**  **(1735)** | **0101001**  **(1735)** | 2 |
| **0101001**  **(1735)** | **0101001**  **(1735)** | 3 |
| **0101001**  **(1735)** | **0101001**  **(1735)** | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Optimal Solution | SC-2  Max-Profit +  Representation | SC-1  Max-Profit +  Representation | Experiment | Problem |
| **1458**  **101010111000011** | 1455  110001111000011 | **1458**  **101010111000011** | 1 | 7 |
| 1465  011100111000011 | 1449  000010001110111 | 2 |
| 1447  000100001101111 | **1458**  **101010111000011** | 3 |
| 1454  011010111000101 | **1458**  **101010111000011** | 4 |
| **1458**  **101010111000011** | 1453  000000111001111 | 5 |
| **1458**  **101010111000011** | 1455  110001111000011 | 6 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Optimal Solution | SC-2  Max-Profit +  Representation | SC-1  Max-Profit +  Representation | Experiment | Problem |
| **13549094**  **11011000110100100000111** | 110111001110000100000101  (13509694) | 110111100110100110010100 (13524340) | 1 | 8 |
| **110111000110100100000111 (13549094)** | 110000001110101110000100 (13456545) | 2 |
| 110001101110101110000001 (13415655) | 110110001110000100000110 (13509264) | 3 |
| 110100000110101100000111 (13460973) | 010100001110100110000011 (13459919) | 4 |

**סיכום:**

* בהתייחס שהמנוע מהווה אלגוריתם קירוב לבעיה, אפשר להגיד שהוא נתן ביצועים טובים. אם נסתכל בבעיה הכי קשה (האחרונה) באחד הניסויים, עם התנאי עצירה השני, האלגוריתם נתן תוצאה מדויקת ואילו בשאר הוא נתן תוצאות קרובות. כנ"ל גם לבעיה השביעית שגם כן היא קשה, האלגוריתם נתן ביצועים לא רעים.
* בששת הבעיות הראשונות האלגוריתם נתן פתרון אופטימלי **ובזמן מאד קצר.**
* מכיוון שבמקרים אלה, מספר הפריטים כמעט היה קטן יחסית, אז מספר האפשריות השונות היה קטן גם כן ולכן גודל האוכלוסייה הוא פרמטר מאד חשוב בבעיה זו. למשל אם נסתכל בבעיה השישית – יש שם 7 פריטים ובסה"כ: אפשריות למחרוזות שונות. אנחנו פתרנו את בעיה עם גודל אוכלוסייה יותר גדול מהמספר הזה ובגלל זה המנוע מצא את הפתרון בדור מוקדם.
* עוד אספקט חשוב – ההתכנסות. שמנו לב אחרי כמה ניסויים, שהגנים באוכלוסייה נעשים יותר דומים בכל דור חדש, ולכן חשבנו על עוד תנאי עצירה **שלא מימשנו** שהוא בעצם אחוז הדמיון בקרב האוכלוסייה.
* כמו כן השתמשנו בשיטת ה- Aging אבל ניסינו המון פרמטרים בשיטה זו וזה לא ממש השפיע על הפתרון.
* בשונה מבעיות קודמות, פה בבעיה זו, הקטנו את כמות האיטרציות המקסימליתוהרבה פעמים דווקא כדאי לתת למנוע לרוץ כמות מסוימת של איטרציות – מסיבה אחת פשוטה – בגלל שזה אלגוריתם קירוב ולא דטרמיניסטי.

**GENETICS VS MINIMAL CONFLICTS**

אנו רוצים להשוות בין אלגוריתם הגנטיקה לבין אלגוריתם ה MINIMAL CONFLICTS בפתירת בעיית ה N Queens .  
לצורך ההשוואה נכניס למנוע שפותר הבעיה בעזרת אלגוריתם הגנטיקה את הפרמטרים שנותנים ביצועים הכי טובים, כלומר:

גודל אוכלוסייה = 1024

הסתברות מוטציה = 0.25

פרופורציית האוכלוסייה האליטיסטית = 0.05

בנוסף:

, swap mutation, random select Cyclic Crossover

* הזמן מוגדר בשניות.

הבדיקות נעשו על גדלי לוח שונים.

**תוצאות אלגוריתם הגנטיקה:**



**תוצאות אלגוריתם ה MINIMAL CONFLICTS:**



**אפשר לראות מהתוצאות שה MINIMAL CONFLICTS הרבה יותר טוב מאלגוריתם הגנטיקה, כך שהוא נותן פתרונות לבעיה עבור גודל לוח מסוים בזמן קצר מאוד ובמקביל אלגוריתם הגנטיקה ייקח לו זמן גדול מאוד כדי למצוא פתרון - אם מצא.**

**האם ניתן להכליא בין שתי הבעיות?**

חשבנו על זה כך: אם נשתמש באלגוריתם ה- Minimal-Conflicts בשלבים השונים בתוך המנוע הגנטי זה יהיה מעניין ואף שווה בדיקה. ישנן כמה אלטרנטיבות:

1. לפני שלב השחלוף, נבצע איטרציה של ה- MC על שני הגנים.
2. בשלב המוטציה נבצע איטרציה של ה- MC על הגן.
3. נוכל גם כן לבצע מנוע דו קיומי כלומר, בעיות גדולות פותרים באמצעות ה- MC ובעיות קטנות באמצעות הגנטיקה. כמו כן, ה- MC משתמע כאלגוריתם חמדני, הוא לא מסתכל קדימה, ואין באמת ידע על המצב הכולל של הלוח בכל זמן נתון ואילו בגנטיקה יש יותר אינפורמציה שעלולה להיות בעלת תועלת במקרים מסוימים.

ולכן לסיכום, כן אפשר להכליא בין שתי הטכניקות.