

ביולוגיה חישובית – מטלה 2 – אלגוריתמים גנטיים – דו"ח תרגיל

שלומי בן שושן, ת"ז 311408264

תקציר

דו"ח זה מתאר את עיקרי פרטי המימוש של אלגוריתם גנטי שנועד לפתור לוחות משחק Futoshiki, ומתעד ניסויים נבחרים שנעשו באמצעות במטרה ללמוד על ההשפעה של אופטימיזציות שונות על ביצועי האלגוריתם.

סעיף א' – עיקרי פרטי המימוש

האלגוריתם הגנטי מתחיל מהצבה אקראית של מספרים במטריצה שיהוו הדור הראשון, ובאמצעות הפעולות הגנטיות רפליקציה, מוטציה ושחלוף, מייצר את הדורות הבאים שהולכים ומתקרבים אל הפתרון החוקי.

להלן עיקרי המימוש:

1. ייצוג הפתרונות

בהינתן לוח קלט מסדר $N \times N$ בו נתונים מראש k מספרים, כל פתרון באוכלוסיית הפתרונות הפוטנציאליים עבור הקלט ייוצג ע"י וקטור בממד $N \times N - k$ של מספרים טבעיים בין 1 ל- N . למשל, עבור קלט מסדר 3×3 עם שני מספרים נתונים (באדום), הייצוג יהיה:

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 2 | 3 | 1 |
| 3 | 1 | 2 |

 \Rightarrow

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 3 | 2 | 3 | 1 | 1 | 2 |
|---|---|---|---|---|---|---|

2. פונקציית הערכה

יהי מרחב פתרונות $S \subseteq \mathbb{N}^{N \times N - k}$, אזי פונקציית הערכה $fitness: S \rightarrow \mathbb{N}$ תוגדר ע"י מספר האילוצים שפתרון $s \in S$ מספק בלוח הקלט. לכל איבר בוקטור פתרון יש בהכרח שני אילוצים לספק – אילוץ שורה ואילוץ עמודה, ובנוסף ישנם אילוצי יחס. לפיכך, ניתן לחשב את מספר האילוצים שמקיים פתרון חוקי בתור $2 \cdot N \cdot N + r$ כאשר r הוא מספר אילוצי היחס בלוח. נשים לב שכאשר נתונים מראש k מספרים, הפונקציה חסומה מלמטה ע"י $2k$. מספר הקריאות הוא מדד לטיב האלגוריתם.

3. פונקציית שחלוף

בהינתן שני פתרונות הורים s_1 ו- s_2 , פונקציית השחלוף מגרילה אינדקס $i \in [0..N \times N - k]$ וקובעת את הפתרון החדש להיות $s_{new} = s_1[0..i - 1] \cdot s_2[i..N \times N - k]$. בכל דור, פונקציה זו נקראת כמספר הצאצאים בדור הבא. בחירת ההורים נעשית ע"י Bias Selection.

4. פונקציית מוטציה

בהינתן פתרון s , פונקציית המוטציה מגרילה מספר $x \in \{1, 2, 3\}$. עבור $x = 1$ הפונקציה בוחרת שני אינדקסים אקראיים i ו- j ומחליפה בין $s[i]$ ו- $s[j]$. עבור $x = 2$ הפונקציה מגרילה i ומחליפה בין $s[i]$ ו- $s[i - 1]$. עבור $x = 3$ הפונקציה מגרילה i ומספר $k \in [1..N]$ וקובעת $s[i] = k$. פונקציה זו נקראת בכל דור לאחר שחלוף עבור כל פתרון שאינו מוגדר כ-elite.

5. התמודדות עם ההתכנסות המוקדמת

התכנסות מוקדמת מזוהה כאשר ערך ה- $fitness$ של הפתרון הטוב ביותר וערך ה- $fitness$ של הפתרון הגרוע ביותר בדור זהים. לאחר הזיהוי, האלגוריתם מטפל בהתכנסות ע"י שמירת הפתרון הטוב ביותר והתחלה מחדש של האלגוריתם ע"י החלפת כל פתרונות האוכלוסייה בפתרונות אקראיים חדשים.

6. הרכב הדור הבא

הדור הבא נוצר על-פי המדיניות הבאה:

- שחלוף: כ-80% מהפתרונות מתקבלים ע"י cross-over מבוסס biased selection.
- שכפול: כ-20% מהפתרונות מתקבלים ע"י רפליקציה מבוססת biased selection.
- אליטיזם: הפתרון הטוב ביותר מועתק לדור הבא כמו שהוא.
- מוטציה: כל פתרון חדש שאינו אליטיסטי עובר מוטציה כמתואר לעיל.

מימוש bias selection נעשה ע"י יצירת מערך בו האינדקס של הפתרון ה- i שערך ה- $fitness$ שלו הוא f מופיע f פעמים. למשל, עבור פתרון $fitness(s_i) = 50$, האינדקס i יופיע במערך 50 פעמים. לאחר מכן, בחירה מוטה של פתרון נעשית ע"י בחירת אינדקס אקראי מהמערך ושליפת הפתרון מרשימת הפתרונות לפי האינדקס.

7. קביעת מספר הדורות להרצה

כדי לקבוע את מספר הדורות להרצה, האלגוריתם הורץ מספר פעמים (ללא אופטימיזציות) על לוחות מסדר 5×5 ברמת קושי easy, ולרוב נמצא פתרון חוקי לאחר כ-3,000-4,000 דורות. בהינתן שרמת הקושי עשויה לעלות, וכן גם הסדר של לוחות הקלט, יידרשו יותר דורות למציאת פתרון חוקי. עם זאת, שימוש באופטימיזציות עשוי להפחית את מספר הדורות הנדרש, ושאלת המחקר מתייחסת להשוואה בין אופטימיזציות שונות. לכן, מספר הדורות הוגבל כברירת מחל ל-5,000 דורות (המשתמש באפליקציה רשאי לשנות זאת). חשוב לציין שמספר הדורות מתייחס לכלל הניסיונות בניסוי. כלומר, בהינתן שהאלגוריתם הגיע בניסיון t להתכנסות מוקדמת בדור g , הדור הראשון בניסיון $t + 1$ שלאחר הפעלה מחדש יהיה $g + 1$. בנוסף, יש לציין שהאלגוריתם עוצר כאשר נמצא פתרון שמספק את כל האילוצים.

סעיף ב' – ניסויים ומסקנות

בסעיף זה תוצג השוואה בין שלוש אסטרטגיות אופטימיזציה:

1. ללא אופטימיזציה – בחינת האלגוריתם הגנטי לעיל.
2. אופטימיזציה דארווינית – כל פתרון עובר לכל היותר N אופטימיזציות, אך הדור הבא נוצר על-פי הפתרונות המקוריים בדור הנוכחי.
3. אופטימיזציה למארקית – כל פתרון עובר לכל היותר N אופטימיזציות, והדור הבא נוצר על-פי הפתרונות המשופרים בדור הנוכחי.

מימוש האופטימיזציה

בהינתן פתרון לשיפור, אלגוריתם האופטימיזציה מחשב לכל תא בלוח את מספר האילוצים שלא מסתפקים לאחר השמת הערך לפי הפתרון. האלגוריתם מגריל מספר $lim \in [1..N]$ ובוחר לשפר את lim הערכים בתאים שמתנגשים עם הכי הרבה אילוצים. הסיבה להגרלה היא הימנעות משיפור קיצוני שעלול לגרום לאחידות בין פתרונות ולהתכנסות מוקדמת. השיפור של תא נבחר נעשה ע"י בדיקת ערכים מותרים עבור אותו תא (לפי אילוצי שורה, עמודה ויחס) ובחירה של ערך אקראי מבין הערכים המותרים. אם לא ניתן לשפר את הפתרון, האלגוריתם יחזיר את אותו פתרון (ולא יגריל אחר במקומו) כי הוא עשוי ליצור פתרון טוב יותר בשחלוף עם פתרון אחר.

השוואת אסטרטגיות

שאלת מחקר: האם היחס בין ביצועי האלגוריתמים השונים משתנה ברמות הקושי ובגדלים השונים?

שיטה: ביצועי האלגוריתמים מוערכים לפי מספר הקריאות לפונקציית ההערכה. מספר הקריאות שמבצע האלגוריתם הרגיל יסומן ב- V (מלשון vanilla), מספר הקריאות שמבצע האלגוריתם הדארוויני יסומן ב- D ומספר הקריאות שמבצע האלגוריתם הלמארקי יסומן ב- L . לאחר כל ניסוי, חושבו היחסים $\frac{D}{V}$ ו- $\frac{L}{V}$ ולבסוף נבחנו ההבדלים.

ניסוי מס' 1: בחינת ביצועי האלגוריתמים בלוח מסדר 5×5 ברמת קושי easy.

| No Optimization | | Darwin Optimization | | Lamarck Optimization | |
|-----------------|-----------|---------------------|-----------|----------------------|-----------|
| Item | Value(s) | Item | Value(s) | Item | Value(s) |
| Solution: | 3 5 4 1 2 | Solution: | 3 5 4 1 2 | Solution: | 3 5 4 1 2 |
| | 2 1 5 4 3 | | 2 1 5 4 3 | | 2 1 5 4 3 |
| | 1 2 3 5 4 | | 1 2 3 5 4 | | 1 2 3 5 4 |
| | 4 3 1 2 5 | | 4 3 1 2 5 | | 4 3 1 2 5 |
| | 5 4 2 3 1 | | 5 4 2 3 1 | | 5 4 2 3 1 |
| Correctness: | True | Correctness: | True | Correctness: | True |
| Fitness: | 58 | Fitness: | 58 | Fitness: | 58 |
| Runtime: | 0:01:30 | Runtime: | 0:01:12 | Runtime: | 0:00:23 |
| Generations: | 3951 | Generations: | 3198 | Generations: | 546 |
| Fitness Calls: | 705254 | Fitness Calls: | 571255 | Fitness Calls: | 117271 |
| Mutate Calls: | 388674 | Mutate Calls: | 314825 | Mutate Calls: | 53460 |
| X-Over Calls: | 314080 | X-Over Calls: | 254404 | X-Over Calls: | 43200 |
| Restarts: | 24 | Restarts: | 19 | Restarts: | 5 |

אבחנות: שלושת האלגוריתמים פותרים את הלוח בתוך 5,000 דורות, כאשר ביצועי הלמארקי הם הטובים ביותר.

$$\frac{D}{V} = \frac{571255}{705254} = 0.810, \quad \frac{L}{V} = \frac{117270}{705254} = 0.166$$

ניסוי מס' 2: בחינת ביצועי האלגוריתמים בלוח מסדר 5×5 ברמת קושי tricky.

| No Optimization | | Darwin Optimization | | Lamarck Optimization | |
|-----------------|-----------|---------------------|-----------|----------------------|-----------|
| Item | Value(s) | Item | Value(s) | Item | Value(s) |
| Solution: | 2 3 4 1 5 | Solution: | 5 3 4 2 1 | Solution: | 5 3 4 2 1 |
| | 1 5 2 4 3 | | 4 5 2 1 3 | | 1 5 3 4 2 |
| | 5 2 1 3 4 | | 3 2 1 5 4 | | 4 2 1 3 5 |
| | 4 1 3 5 2 | | 2 1 3 4 5 | | 3 1 2 5 4 |
| | 3 4 5 2 1 | | 1 4 5 3 2 | | 2 4 5 1 3 |
| Correctness: | False | Correctness: | False | Correctness: | False |
| Fitness: | 60 | Fitness: | 60 | Fitness: | 60 |
| Runtime: | 0:02:20 | Runtime: | 0:04:05 | Runtime: | 0:04:00 |
| Generations: | 5000 | Generations: | 5000 | Generations: | 5000 |
| Fitness Calls: | 893046 | Fitness Calls: | 1178874 | Fitness Calls: | 1029009 |
| Mutate Calls: | 492426 | Mutate Calls: | 492426 | Mutate Calls: | 491832 |
| X-Over Calls: | 397920 | X-Over Calls: | 397920 | X-Over Calls: | 397440 |
| Restarts: | 26 | Restarts: | 26 | Restarts: | 32 |

אבחנות: שלושת האלגוריתמים כשלו. הציון המקסימלי בניסוי הוא $2 \cdot 5 \cdot 5 + 11 = 61$, כך שהפתרונות שנמצאו כללו אי-התאמה אחת או שתיים. האלגוריתם המהיר ביותר היה ללא אופטימיזציה. האופטימיזציה הלמארקית גרמה ליותר התכנסויות מוקדמות ביחס לאופטימיזציה הדארווינית. ניתן לנמק זאת ע"י כך שהאלגוריתם הלמארקי יוצר את הדור הבא לפי האוכלוסייה המשופרת, מה שעלול ליצור פתרונות דומים יותר (בכיוון הפתרון החוקי) וכתוצאה מכך לגרור התכנסות מוקדמת.

$$\frac{D}{V} = \frac{1178874}{893046} = 1.320, \quad \frac{L}{V} = \frac{1029009}{893046} = 1.152$$

ניסוי מס' 3: בחינת ביצועי האלגוריתמים בלוח מסדר 6×6 ברמת קושי easy.

| No Optimization | | Darwin Optimization | | Lamarck Optimization | |
|-----------------|-------------|---------------------|-------------|----------------------|-------------|
| Item | Value(s) | Item | Value(s) | Item | Value(s) |
| Solution: | 5 2 6 3 4 1 | Solution: | 5 6 4 2 3 1 | Solution: | 5 6 1 2 4 3 |
| | 2 1 5 6 3 4 | | 1 4 5 6 2 3 | | 3 1 5 4 2 6 |
| | 3 6 4 2 1 5 | | 3 5 2 4 1 6 | | 1 2 4 6 3 5 |
| | 6 5 1 4 2 3 | | 6 2 1 3 4 5 | | 6 5 2 3 1 4 |
| | 1 4 3 5 6 2 | | 2 1 3 5 6 4 | | 2 4 3 5 6 1 |
| | 4 3 2 1 5 6 | | 4 3 6 1 5 2 | | 4 3 6 1 5 2 |
| Correctness: | False | Correctness: | False | Correctness: | False |
| Fitness: | 81 | Fitness: | 80 | Fitness: | 81 |
| Runtime: | 0:02:00 | Runtime: | 0:04:12 | Runtime: | 0:04:08 |
| Generations: | 5000 | Generations: | 5000 | Generations: | 5000 |
| Fitness Calls: | 893836 | Fitness Calls: | 1223563 | Fitness Calls: | 1064711 |
| Mutate Calls: | 493416 | Mutate Calls: | 493317 | Mutate Calls: | 491337 |
| X-Over Calls: | 398720 | X-Over Calls: | 398640 | X-Over Calls: | 397040 |
| Restarts: | 16 | Restarts: | 17 | Restarts: | 37 |

אבחנות: שלושת האלגוריתמים כשלו. בניסוי זה הציון המקסימלי הוא 82, כלומר האלגוריתם ללא האופטימיזציה והאופטימיזציה הלמארקית הגיעו לאי-התאמה אחת והאופטימיזציה הדארווינית הגיע לשתי אי-התאמות.

$$\frac{D}{V} = \frac{1227140}{893836} = 1.373, \quad \frac{L}{V} = \frac{1065396}{893836} = 1.191$$

ניסוי מס' 4: בחינת ביצועי האלגוריתמים בלוח מסדר 6×6 ברמת קושי tricky.

| No Optimization | | Darwin Optimization | | Lamarck Optimization | |
|-----------------|-------------|---------------------|-------------|----------------------|-------------|
| Item | Value(s) | Item | Value(s) | Item | Value(s) |
| Solution: | 4 2 3 1 5 6 | Solution: | 4 1 2 3 5 6 | Solution: | 4 3 2 1 6 5 |
| | 3 1 5 2 6 4 | | 5 6 3 4 1 2 | | 6 2 3 5 1 4 |
| | 5 3 1 6 4 2 | | 3 2 5 6 4 1 | | 1 6 5 2 4 3 |
| | 6 5 2 4 3 1 | | 6 3 1 5 2 4 | | 5 1 6 4 3 2 |
| | 1 4 6 3 2 5 | | 2 4 6 1 3 5 | | 2 4 1 3 5 6 |
| | 2 6 4 5 1 3 | | 1 5 4 2 6 3 | | 3 5 4 6 2 1 |
| Correctness: | False | Correctness: | False | Correctness: | False |
| Fitness: | 88 | Fitness: | 87 | Fitness: | 88 |
| Runtime: | 0:03:56 | Runtime: | 0:06:50 | Runtime: | 0:06:39 |
| Generations: | 5000 | Generations: | 5000 | Generations: | 5000 |
| Fitness Calls: | 893836 | Fitness Calls: | 1227140 | Fitness Calls: | 1065396 |
| Mutate Calls: | 493416 | Mutate Calls: | 493317 | Mutate Calls: | 491832 |
| X-Over Calls: | 398720 | X-Over Calls: | 398640 | X-Over Calls: | 397440 |
| Restarts: | 16 | Restarts: | 17 | Restarts: | 32 |

אבחנות: גם בניסוי זה מספר האילוצים שלא סופקו ע"י הפתרונות שממצאו הוא בין 1 ל-2.

$$\frac{D}{V} = \frac{1227140}{893836} = 1.373, \quad \frac{L}{V} = \frac{1065396}{893836} = 1.191$$

ניסוי מס' 5: בחינת ביצועי האלגוריתמים בלוח מסדר 7×7 ברמת קושי easy.

| No Optimization | | Darwin Optimization | | Lamarck Optimization | |
|-----------------|---|---------------------|---|----------------------|---|
| Item | Value(s) | Item | Value(s) | Item | Value(s) |
| Solution: | 6 3 2 7 4 1 5 1 5 3 2 6 7 4 2 7 6 5 3 4 1 5 2 1 4 7 6 3 7 6 4 3 1 5 2 3 4 7 1 5 2 6 4 1 5 6 2 3 7 | Solution: | 6 7 3 1 4 2 5 5 1 7 6 3 4 2 3 4 6 2 7 5 1 4 5 1 7 2 6 3 7 3 2 4 5 1 6 2 6 5 3 1 7 4 1 2 4 5 6 3 7 | Solution: | 5 7 3 1 4 2 6 4 2 1 6 7 3 5 3 6 7 5 2 1 4 2 4 5 7 1 6 3 7 3 2 4 6 5 1 6 1 4 3 5 7 2 1 5 6 2 3 4 7 |
| Correctness: | False | Correctness: | False | Correctness: | False |
| Fitness: | 111 | Fitness: | 114 | Fitness: | 116 |
| Runtime: | 0:05:00 | Runtime: | 0:10:03 | Runtime: | 0:09:25 |
| Generations: | 5000 | Generations: | 5000 | Generations: | 5000 |
| Fitness Calls: | 894389 | Fitness Calls: | 1202349 | Fitness Calls: | 1070620 |
| Mutate Calls: | 494109 | Mutate Calls: | 494010 | Mutate Calls: | 492624 |
| X-Over Calls: | 399280 | X-Over Calls: | 399200 | X-Over Calls: | 398080 |
| Restarts: | 9 | Restarts: | 10 | Restarts: | 24 |

אבחנות: תוצאות ניסוי זה גרועות מבקודמים, ומכך ניתן להסיק כי הגדלת סדר לוח הקלט מקשה על האלגוריתמים יותר מהעלאה ברמת הקושי. ניתן לנמק זאת ע"י כך שהגדלת הסדר מגדילה את מרחב החיפוש באופן משמעותי יותר מהעלאה ברמת הקושי. בנוסף, שלושת האלגוריתמים חוו פחות התכנסויות מוקדמות.

$$\frac{D}{V} = \frac{1202349}{894389} = 1.344, \quad \frac{L}{V} = \frac{1070620}{894389} = 1.197$$

ניסוי מס' 6: בחינת ביצועי האלגוריתמים בלוח מסדר 7×7 ברמת קושי tricky.

| No Optimization | | Darwin Optimization | | Lamarck Optimization | |
|-----------------|---|---------------------|---|----------------------|---|
| Item | Value(s) | Item | Value(s) | Item | Value(s) |
| Solution: | 3 6 1 5 7 4 2 5 1 7 4 2 6 3 7 2 4 1 6 3 5 6 4 2 3 5 1 7 4 5 3 2 1 7 6 1 7 5 6 3 2 4 2 3 6 7 4 5 1 | Solution: | 3 6 4 7 1 2 5 5 1 6 4 2 7 3 2 7 3 6 5 4 1 6 5 2 3 7 1 4 7 4 1 5 6 3 2 1 3 5 2 4 6 7 4 2 7 1 3 5 6 | Solution: | 4 2 3 7 1 5 6 7 1 4 3 2 6 5 5 6 1 2 3 7 4 6 7 2 5 4 1 3 2 4 7 6 5 3 1 1 3 5 4 6 2 7 3 5 6 1 7 4 2 |
| Correctness: | False | Correctness: | False | Correctness: | False |
| Fitness: | 112 | Fitness: | 113 | Fitness: | 113 |
| Runtime: | 0:05:25 | Runtime: | 0:09:36 | Runtime: | 0:09:18 |
| Generations: | 5000 | Generations: | 5000 | Generations: | 5000 |
| Fitness Calls: | 894310 | Fitness Calls: | 1247347 | Fitness Calls: | 1082273 |
| Mutate Calls: | 494010 | Mutate Calls: | 493911 | Mutate Calls: | 492525 |
| X-Over Calls: | 399200 | X-Over Calls: | 399120 | X-Over Calls: | 398000 |
| Restarts: | 10 | Restarts: | 11 | Restarts: | 25 |

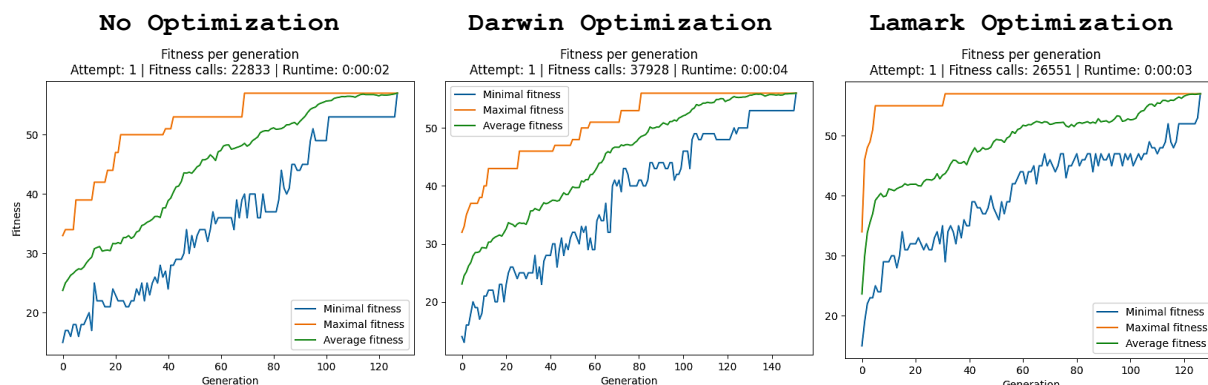
אבחנות: תוצאות הניסוי דומות לתוצאת הניסוי הקודם, דבר המחזק את ההבנה כי הגדלת סדר לוח הקלט מקשה על האלגוריתמים יותר מהעלאה רמת הקושי. עוד ניתן לראות שמספר קריאות ה-fitness שנעשו ע"י האלגוריתם הלמארקי קטן מבדארוני, למרות שהלמארקי הגיע ל-14 התכנסויות מוקדמות יותר מהדארוני. הטיפול בהתכנסות מוקדמת במימוש זה כולל יצירת אוכלוסיית פתרונות חדשה, ויצירה של כל פתרון גוררת קריאה לפונקציית fitness, ולכן לכאורה הגעה ליותר התכנסויות מוקדמות אמורה לגרור יותר קריאות fitness. ניתן להסביר זאת ע"י כך שאלגוריתם האופטימיזציה מייצר פתרון חדש משופר רק אם ניתן לשפר את הפתרון שהוא מקבל. כלומר, במקרים רבים באלגוריתם הלמארקי האופטימיזציה לא הצליחה, ולכן לא יצרה פתרון חדש, ולכן לא קרא שוב ל-fitness.

$$\frac{D}{V} = \frac{1247347}{894310} = 1.395, \quad \frac{L}{V} = \frac{1082273}{894310} = 1.210$$

ניסוי מס' 7: בחינת ציון ה-fitness של הפתרון הטוב ביותר וציון ה-fitness הממוצע כפונקציה של דור. בניסוי זה הקלט הוא לוח מסדר 5×5 ברמת קושי easy, שלושת האלגוריתמים הורצו עד להתכנסות המוקדמת הראשונה. ברגע זה כל אלגוריתם הפיק גרף שמלמד כיצד ציון הפתרון הטוב ביותר והציון הממוצע באוכלוסייה (וציון הפתרון הגרוע ביותר) משתנים לאורך הדורות. יש לציין שמיד לאחר ההתכנסות המוקדמת, כל אלגוריתם מגריל פתרונות

חדשים ומתחיל מהתחלה, ואז התהליך המתואר בתרשימים מטה חוזר על עצמו עד להתכנסות המוקדמת הבאה או עד למציאת הפתרון החוקי (המוקדם מביניהם).

תוצאות הניסוי:



אבחנות:

- כתוצאה מהאליטיזם, ציון הפתרון הטוב ביותר (בכתום) לא יכול לרדת.
- ה-bias selection גורמת לכלל הפתרונות להשתפר מדור לדור עד להתכנסות לפתרון הטוב ביותר.
- האלגוריתם הלמארקי משפר במהרה את רוב האוכלוסייה ואז אין שינוי משמעותי בציון הפתרון הטוב ביותר עד להתכנסות.
- אין הבדל משמעותי בין האלגוריתם הדארוויני לאלגוריתם הרגיל ביחס בין הפתרון הטוב ביותר לממוצע.

מסקנות

- כל האלגוריתמים משפרים את אוכלוסיית הפתרונות אודות ל-bias selection, ומוצאים פתרון מקורב.
- כאשר שלושת האלגוריתמים מורצים על מספר זהה של דורות, האלגוריתם הרגיל הוא המהיר ביותר והחסכוני ביותר בקריאות fitness.
- אם האלגוריתם הלמארקי לא מוצא פתרון חוקי יחסית מהר, הוא מגיע ליותר התכנסויות מוקדמות מהאחרים.
- מספר ההתכנסויות המוקדמות אליהן מגיע האלגוריתם הדארוויני דומה למספר של האלגוריתם הרגיל.
- כתוצאה מהאליטיזם, ציון הפתרון הטוב ביותר לא יכול לרדת במהלך הדורות.
- האלגוריתם הלמארקי משפר במהרה את רוב האוכלוסייה ואז לא חל שיפור בציון הטוב ביותר עד להתכנסות.

טבלת יחסים:

| ניסוי מס' | 1 | 2 | 3 | 4 | 5 | 6 |
|-----------|-------|--------|-------|--------|-------|--------|
| סדר | 5 × 5 | 5 × 5 | 6 × 6 | 6 × 6 | 7 × 7 | 7 × 7 |
| רמת קושי | easy | tricky | easy | tricky | easy | tricky |
| D/V | 0.810 | 1.320 | 1.373 | 1.373 | 1.344 | 1.395 |
| L/V | 0.166 | 1.152 | 1.191 | 1.191 | 1.197 | 1.210 |
| D/L | 4.880 | 1.146 | 1.153 | 1.153 | 1.123 | 1.153 |

במימוש זה האלגוריתמים עוצרים במידה ומצאו פתרון חוקי, כלומר מציאת פתרון חוקי גוררת ריצה של פחות דורות, וכתוצאה מכך יצירה של פחות פתרונות ופחות קריאות fitness. לכן, ניסוי מס' 1 לא רלוונטי למענה על שאלת המחקר.

סיכום

היחס בין ביצועי האלגוריתמים – הרגיל, הדארוויני והלמארקי – משתנה בהתאם לגדלי לוחות הקלט באופן משמעותי יותר מאשר בהתאם לרמות הקושי. ניתן לזהות בטבלה לעיל הבדלים קטנים ביחסים בין ביצועי האלגוריתמים במעברים מ-easy ל-tricky תחת אותו סדר (מבלי להתייחס לניסוי 1). לעומת זאת, בהסתכלות על מעברים בין סדרי גודל של אותה רמה (למשל ברמה tricky בין שלושת הסדרים), ניתן לזהות הבדלים מעט יותר משמעותיים. כך או כך, מה שמשפיע יותר על יחסי הביצועים במימוש זה הוא מציאת פתרון חוקי, כמו שניתן לראות בהבדל בין ניסוי 1 ל-2 מבחינת רמת קושי או בין ניסוי 1 ל-3 או 4 מבחינת סדר.