**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 Futoshiki, מתעד ניסויים נבחרים שנעשו באמצעות האלגוריתם ומפרט מספר משמעויות ומסקנות.

**סעיף א' – עיקרי פרטי המימוש**

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

להלן עיקרי המימוש:

1. **ייצוג הפתרונות**

בהינתן לוח קלט מסדר בו נתונים מראש מספרים*, כל פתרון באוכלוסיית הפתרונות הפוטנציאליים עבור הקלט ייוצג ע"י וקטור בממד של מספרים טבעיים בין ל-. למשל, עבור קלט מסדר עם שני מספרים נתונים מראש (באדום), ניתן לייצג:*

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

1. **פונקציית הערכה**

יהי מרחב פתרונות , אזי פונקציית הערכה תוגדר ע"י מספר האילוצים שפתרון 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 כאשר הוא מספר אילוצי היחס בלוח. נשים לב שכאשר נתונים מראש מספרים, הפונקציה חסומה מלמטה ע"י *.* מספר הקריאות הוא מדד לטיב האלגוריתם.

1. **פונקציית שחלוף**

בהינתן שני פתרונות ו-, פונקציית השחלוף מגרילה אינדקס *וקובעת את הפתרון החדש להיות . בכל דור, פונקציה זו נקראת כמספר הצאצאים בדור הבא. בחירת ההורים נעשית ע"י* Bias Selection.

1. **פונקציית מוטציה**

בהינתן פתרון , פונקציית המוטציה מגרילה מספר . עבור הפונקציה בוחרת שני אינדקסים אקראיים ו- *ומחליפה בין ו-. עבור הפונקציה מגרילה ומחליפה בין ו-. במידה ו- הפונקציה מגרילה ומספר וקובעת . פונקציה זו נקראת בכל דור לאחר שחלוף עבור כל פתרון שאינו מוגדר כ-*elite*.*

1. **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.

1. **הרכב הדור הבא**

הדור הבא נוצר על-פי המדיניות הבאה:

* שחלוף: כ- מהפתרונות מתקבלים ע"י cross-over מבוסס biased selection.
* שכפול: כ- 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- שערך ה-fitness הוא מופיע פעמים. למשל, עבור פתרון , *האינדקס יופיע במערך פעמים. לאחר מכן, בחירה מוטה של פתרון תיעשה ע"י בחירת אינדקס אקראי מהמערך ושליפת הפתרון לפי האינדקס.*

1. **קביעת מספר הדורות להרצה**

כדי לקבוע את מספר הדורות להרצה, האלגוריתם הורץ מספר פעמים ללא האופטימיזציות המתוארות *בסעיף הבא על לוחות מסדר ברמות קושי* easy ו-tricky. ברמת easy ניתן למצוא פתרון חוקי לאחר כ-4,000 דורות, וברמת tricky ניתן למצוא פתרון חוקי לאחר כ-70,000 דורות. מצד אחד ייתכנו לוחות קלט ברמות קושי גבוהות יותר או מסדר גבוה יותר, אך מצד שני מדובר באלגוריתם ללא אופטימיזציה. לכן, מספר הדורות להרצה כברירת מחדל נקבע להיות 70,000. במידה ומעלים את רמת הקושי או את סדר הלוח, מומלץ להוסיף דורות להרצה. במידה ומשתמשים אופטימיזציות, ניתן להפחית במספר הדורות. חשוב לציין שמספר הדורות מתייחס לכלל הניסיונות בניסוי. כלומר, בהינתן שהאלגוריתם הגיע בניסיון להתכנסות מוקדמת בדור , הדור הראשון בניסיון  *שלאחר הפעלה מחדש יהיה .* בנוסף, יש לציין שהאלגוריתם עוצר כאשר נמצא פתרון שמספק את כל האילוצים.

**סעיף ב' – ניסויים ומסקנות**

בסעיף זה יוצג מימוש של אופטימיזציה מקומית שניתן להחיל על פתרון, ולאחר מכן תוצג השוואה בין שלוש אסטרטגיות אופטימיזציה:

1. ללא אופטימיזציה – בחינת האלגוריתם הגנטי לעיל.
2. אופטימיזציה דארווינית – כל פתרון עובר לכל היותר אופטימיזציות, אך הדור הבא נוצר על-פי הפתרונות המקוריים בדור הנוכחי.
3. אופטימיזציה למארקית – כל פתרון עובר לכל היותר 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 ובוחר לשפר את  *הערכים בתאים שמתנגשים עם הכי הרבה אילוצים. הסיבה להגרלה היא הימנעות משיפור קיצוני מידי 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?

ההשוואה תתבסס על השאלות האם הפתרון שנבחר ע"י האלגוריתם הוא חוקי, מה ציון ה-fitness שלו, מה זמן הריצה, כמה דורות האלגוריתם הריץ, וכמה קריאות ל-fitness ביצע. כמו כן, בכל ניסוי יוצג גרף שמתאר כיצד הציון של הפתרון הטוב ביותר והציון הממוצע באוכלוסייה משתנים במשך הדורות.

ניסוי מס' 1: לוח מסדר ברמת קושי **easy** ואלגוריתם **ללא אופטימיזציה**.

1. פתרון נבחר:
2. נכונות הפתרון:
3. ציון fitness:
4. זמן ריצה:
5. דורות:
6. קריאות fitness:
7. ציוני Fitness כפונקציה של דור:

קובץ ReadMe (פרמטרים, קבצים) ושימוש באפליקציה.

סיכום.

תיעוד הקוד.