דוח ביולוגיה חישובית תרגיל 2

אייל קזולה 209133693 – מדעי המחשב

דניאל טמיר 323127498 – מדעי המחשב

אנחנו מודעים לדרישת הנוכחות בקורס כפי שפורטו במכתבים ובשיעור הראשון ולכך שמי שלא עומד בדרישה זו לא יוכל לעבור את הקורס

GITHUB REPOSITORY:

<https://github.com/Dani3lTamir/Computational-Biology-2>

על מנת להריץ את הקוד יש להוריד ולחלץ את הריפוזיטורי ולאחר מכן:

* לפתוח את קובץ ההרצה magic\_square.exe
* יופיע CLI אינטרקטיבי בו תוכלו להזין את גודל הריבוע ((n
* לאחר מכן תתבקשו להזין את שיטת האבולוציה הרצויה (0- הרגילה, 1- הלמארקית, 2- הדארווינית)
* יופיעו עדכונים על מהלך הריצה ולבסוף GUI בו מידע נוסף על מה שהתקבל
* תוכלו להתחיל ריצה נוספת ע"י הזנת 'y' או לסיים את התוכנית ע"י 'n'

**תיעוד**

* איך יוצגו הפתרונות - כל פתרון הוא מערך בגודל n^2 שמכיל פרמוטציה של כל המספרים מ1 עד n^2 , כאשר אנחנו רוצים להתייחס אליהם כמטריצה אנחנו עושים reshape למטריצה בגודל nXn
* מהי פונקצית ההערכה .- עבור n שלא מתחלק ב4 פונקציית ההערכה מחשבת את הסטייה מ M קבוע הקסם של כל שורה, עמודה ואלכסון וסוכמת את סך הסטיות ועל מנת שסטייה 0 תהיה המקסימום מחשבת את הגודל כלומר כשהריבוע הוא ריבוע קסם פונקציית הערכה מחזירה 1.

עבור n שמתחלק ב 4 פונקציית הערכה מחשבת רכיב חדש שמטרתו הגעה לריבוע קסם מושלם (רכיב זה מוכפל ב0.5 על מנת שלא ייפגע בפונקצייה הכללית) על מנת לחשב רכיב זה מחשבים בנוסף את הסטייה מהגדרת ריבוע קסם מושלם (הסטייה של סכום כל ריבוע בגודל 2X2 מ M) ומוסיפים זאת לתוצאת פונקציית הערכה כלומר כאן ההערכה יכולה להיות גדולה מ1

* איך ביצעתם את פעולת ה over-cross בין פתרונות. – בוחרים 2 הורים על ידי שיטת tournament selection כלומר כל פעם מגרילים קבוע של הורים אפשריים ובוחרים את ההורה עם פונקציית ההערכה הטובה ביותר, כעת מגרילים רצף אינדקסים אותו לוקחים מאחד ההורים ואת כל שאר הערכים לוקחים מההורה השני כאשר מדלגים על כל הערכים שכבר נמצאים, עקב העובדה ששני ההורים פתרונות תקינים ומכילים את כל המספרים הילד גם יהיה פתרון תקין. בנוסף יש אליטיזם כלומר הפתרונות הטובים ביותר נשארים לדור הבא גם. שיטת הטורניר גם עוזרת לבריחה מאופטימום מקומי ומניעת התכנסות מוקדמת בכך שהיא נותנת הזדמנות לפתרונות שמוצלחים ביחס לנישה מצומצמת יותר.
* כיצד מומשו מוטציות. – ישנו פרמטר שמגדיר את קצב המוטציה הבסיסי ולאורך הריצות אנחנו מגדילים או מקטינים אותו בהתאם לכמה האוכלוסיה דומה אחד לשני, בהסתברות ששוה לסיכוי זה כל ילד עובר מוטציה כאשר ישנם 3 סוגי מוטציות אפשריות שיכולות לקרות והסיכויים של כולן שווים – החלפה ערבול והפיכה  
  החלפה – מחליפים 2 ערכים רנדומלים במיקום

ערבול – בוחרים מקטע רציף בפתרון ומערבבים את הערכים בתוכו באופן אקראי

הפיכה- בוחרים מקטע רציף והופכים את הסדר של הערכים בו

* האם/איך התייחסתם לבעיית ההתכנסות המוקדמת. – אנו מחשבים בכל דור את שונות האוכלוסיה ואם השונות קטנה מפרמטר שמוגדר בהתחלה אנו מכניסים פתרונות אקראיים חדשים (מהגרים) במקום הפתרונות הגרועים ביותר.
* איך החלטתם מתי לעצור את הריצה – אם n לא מתחלק ב 4 עוצרים את הריצה כאשר מוצאים ריבוע קסם או שמספר הדורות המוגדר עבר  
  אם n לא מתחלק ב 4 עוצרים אם נמצא ריבוע מושלם או שמספר הדורות עבר, בכל מקרה אנחנו שומרים את הריבוע הטוב ביותר ומציגים אותו בסוף. לאחר ניסויים רבים הגענו למסקנה כי לעיתים מאוד מאוד נדירות פתרון תקף נמצא לאחר 500 דורות של אי הצלחה ולכן זה מספר הדורות המקסימלי שקבענו.
* הסבר על אבולוציות שונות – באלגוריתמים הדארויני והלמארקי ישנה אופטימזיה לוקלית בכל דור, היא רצה n איטרציות, בכל איטרציה היא בודקת מספר קבוע של החלפות אקראיות ומחשבת את הfitness עבור כל אחת ושומרת את הטובה ביותר, אם לא נמצאה החלפה משפרת אנו בכל זאת בוחרים החלפה גרועה יותר על מנת לאפשר לאלגוריתם לברוח מאופטימום מקומי.

**הסבר על הגרפים והתוצאות**

בסוף כל ריצה אנו מדפיסים לCLI באיזו אבולוציה השתמשנו, מהו קבוע הקסם שמתאים ל N ומה הפתרון הטוב ביותר שנמצא שמודפס כמטריצה ומה סכום כל שורה עמודה ואלכסון שלו. ובנוסף האם נמצא ריבוע קסם או לא, כאשר N מתחלק ב4 נדפיס גם האם נמצא ריבוע קסם מושלם ואת סכום תתי הריבועים.

במהלך הריצה כל 25 דורות אנו מדפיסים את הפתרון עם הFITNESS הגבוה ביותר שנמצא באוכלוסייה, הFITNESS הממוצע, השונות, וההסתברות למוטציה (שמשתנה אדפטיבית לפי השונות).

בנוסף אנחנו מציגים למסך 4 גרפים:

פונקציית הערכה לאורך הדורות כאשר מוצג גם הממוצע וגם הציון הגבוה ביותר

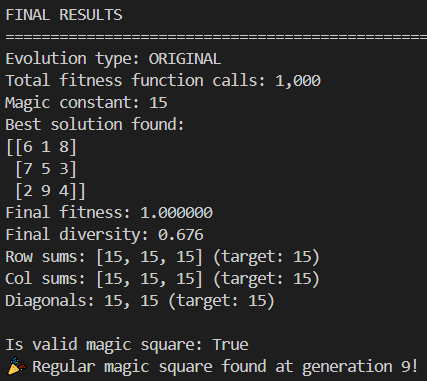
השונות של האוכלוסיה בכל דור

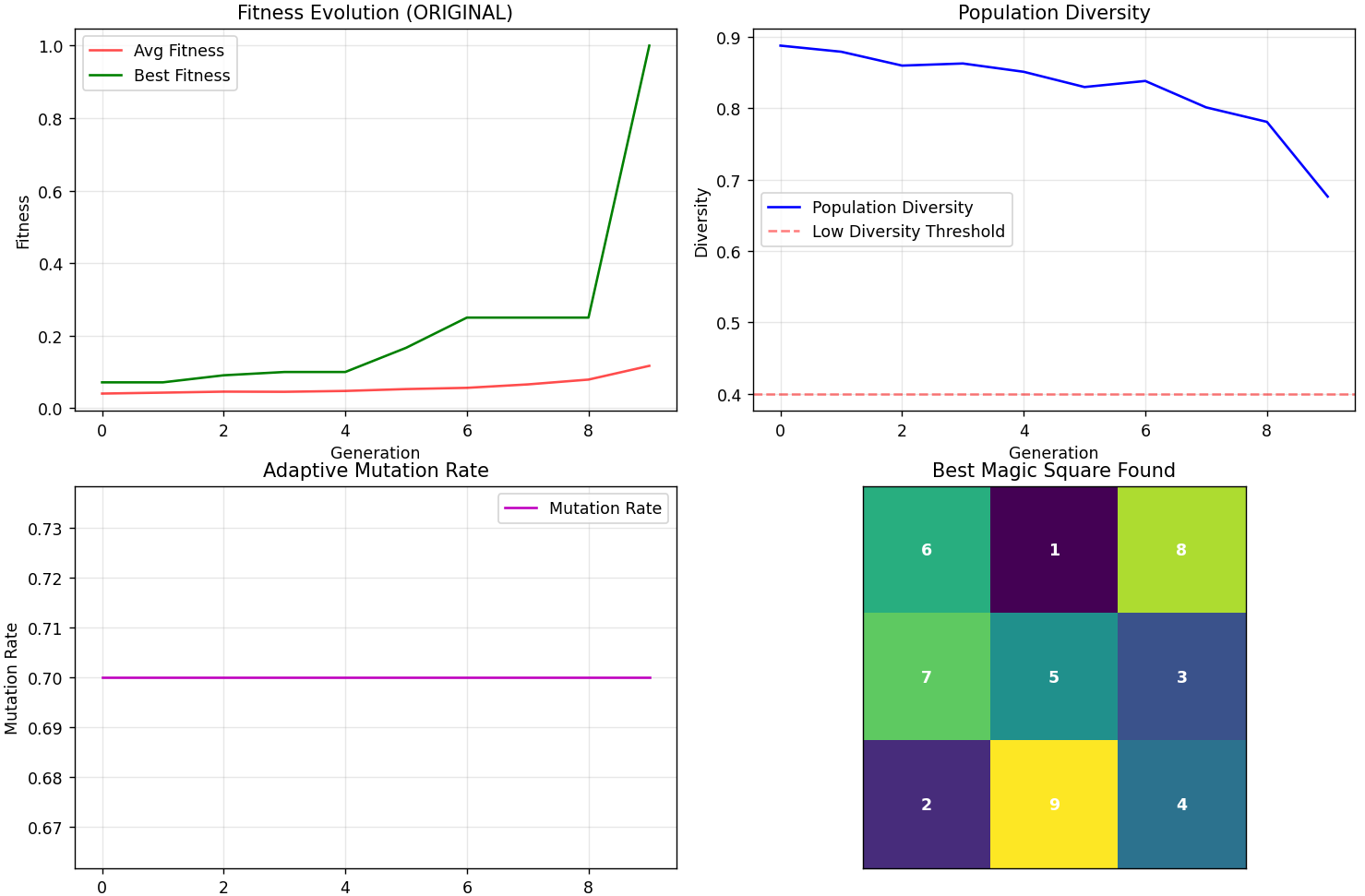
הסיכוי למוטציה בכל דור (ביחס הפוך לשונות)

והריבוע הטוב ביותר שנמצא

**ניתוח התוצאות על N שונים ואבולוציות שונות**

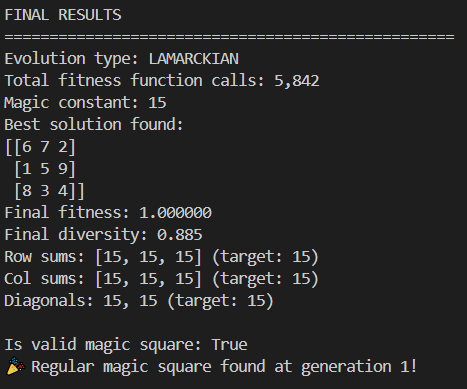
אנו מוצאים ריבוע קסם באופן עקבי ל3X3 גם עם אבולוציה רגילה:

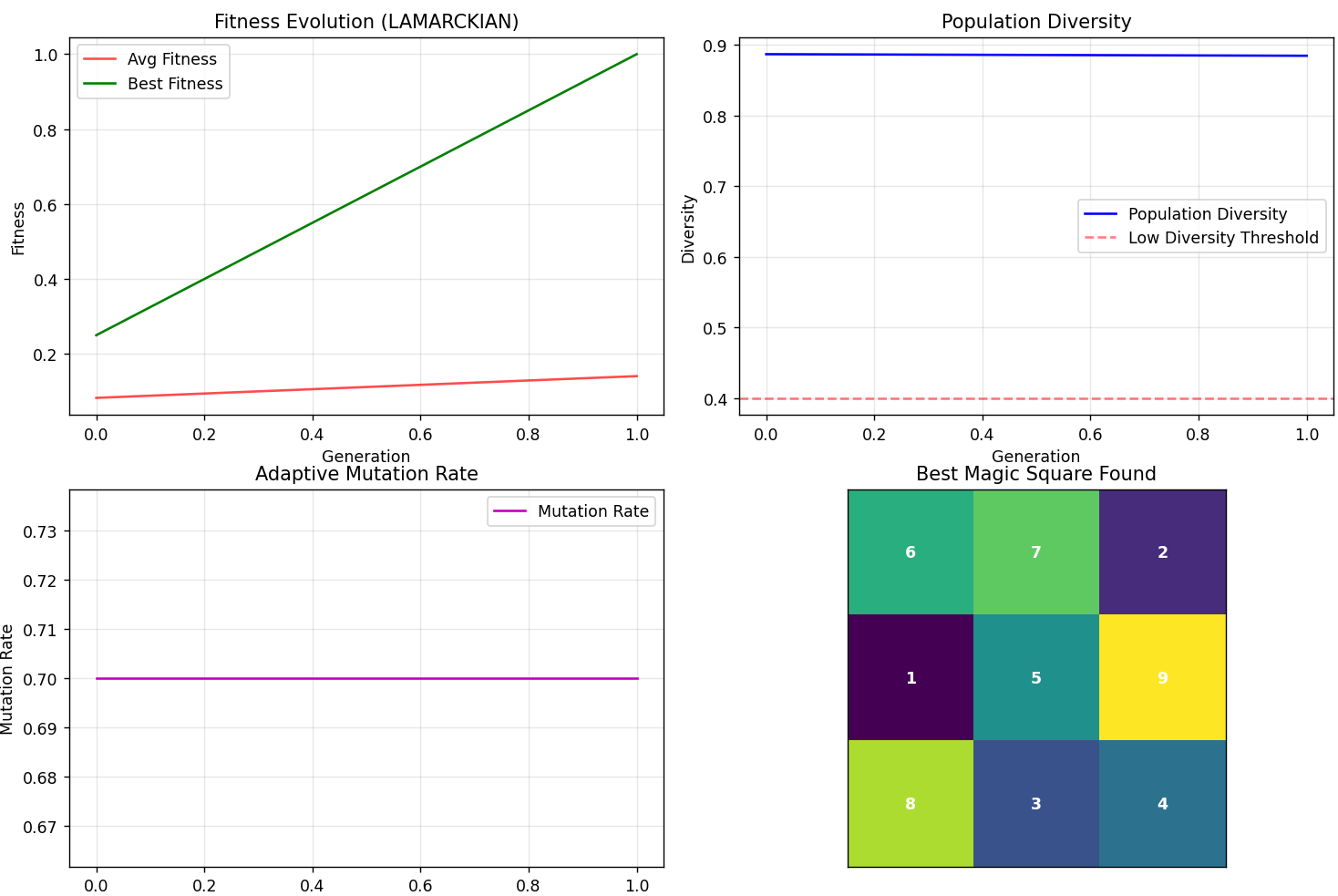


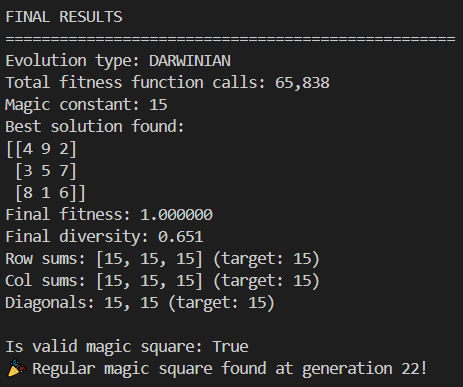


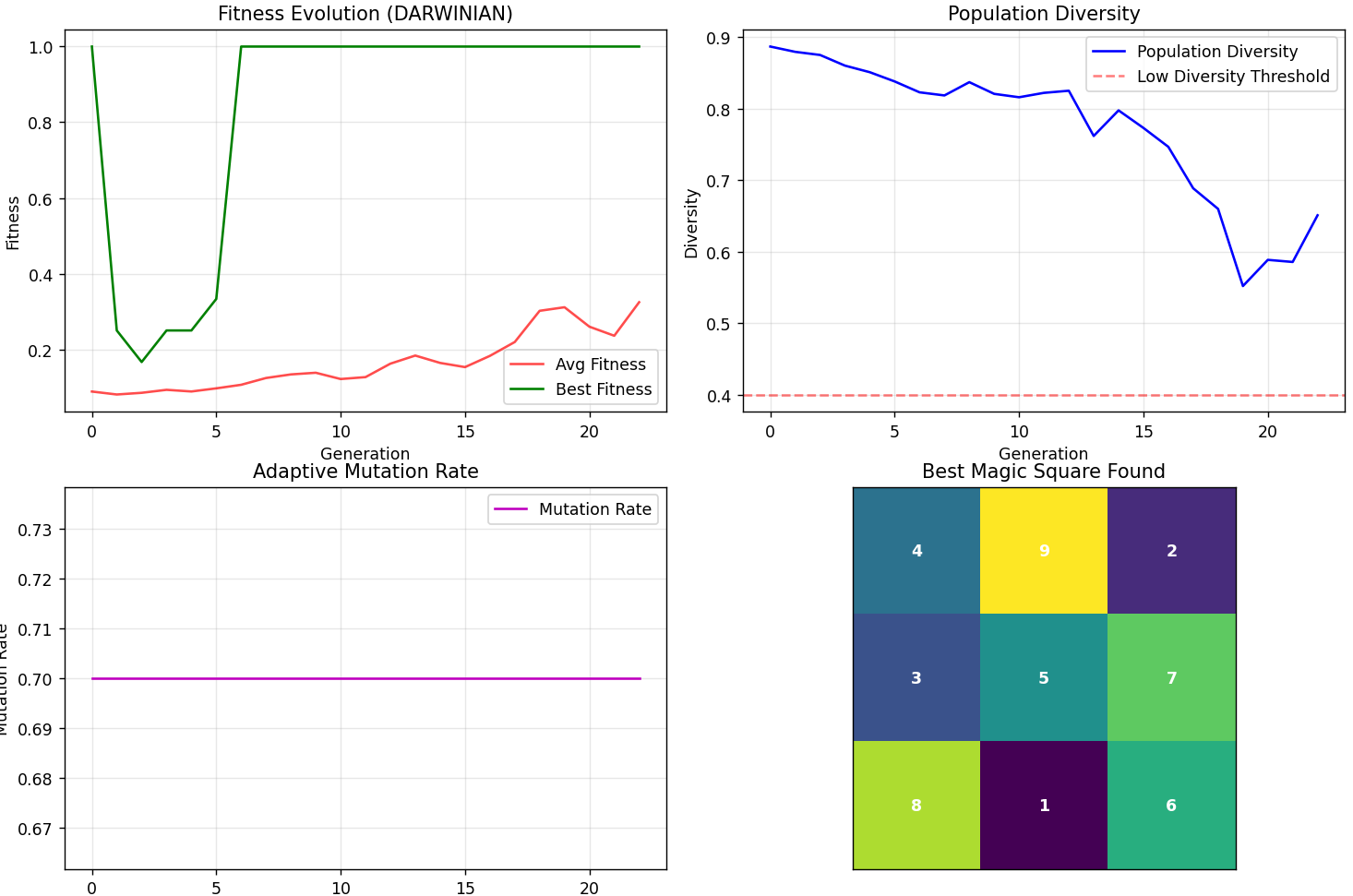
**ועם למארק ממש מהר:**

בדרך כלל בדור או שניים





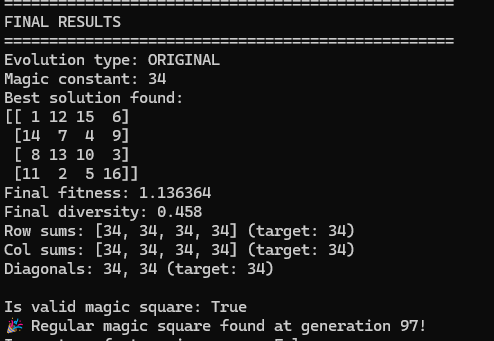
דרוואניאן משמעותית פחות טוב:  


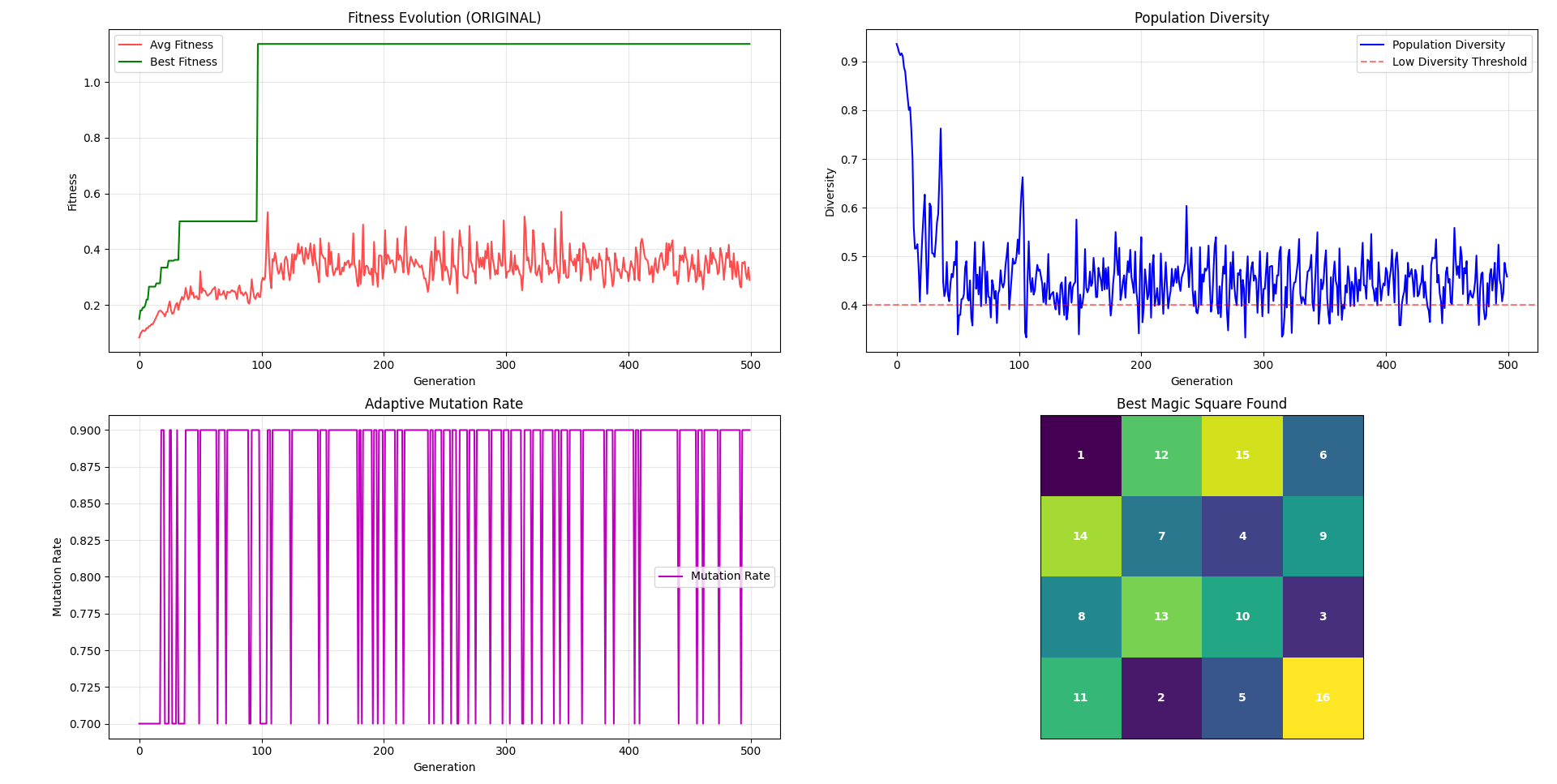


כפי שניתן לראות לדרוויני לוקח הרבה יותר זמן וכמות השוואות של פונק' ההערכה, אנו מסיקים שזה קורה בגלל שהיקף הפתרונות במימד זה הוא קטן מידי בשביל שהפתרונות שפוטנציאל השיפור שלהם יתרמו למציאת החיפוש מהר יותר מפתרונות שהערכה הנוכחית שלהם היא גבוהה ולכן זה רק מעכב את מציאת הפתרון.

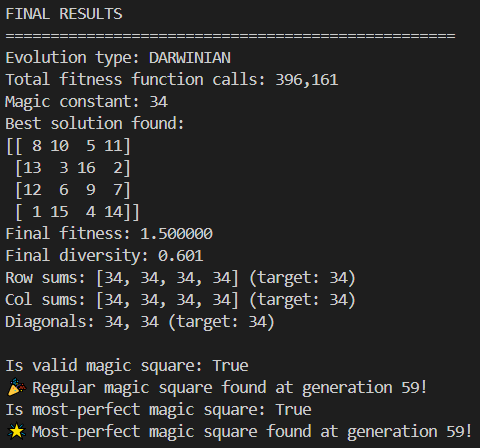
**עבור ריבועים בגודל 4X4**

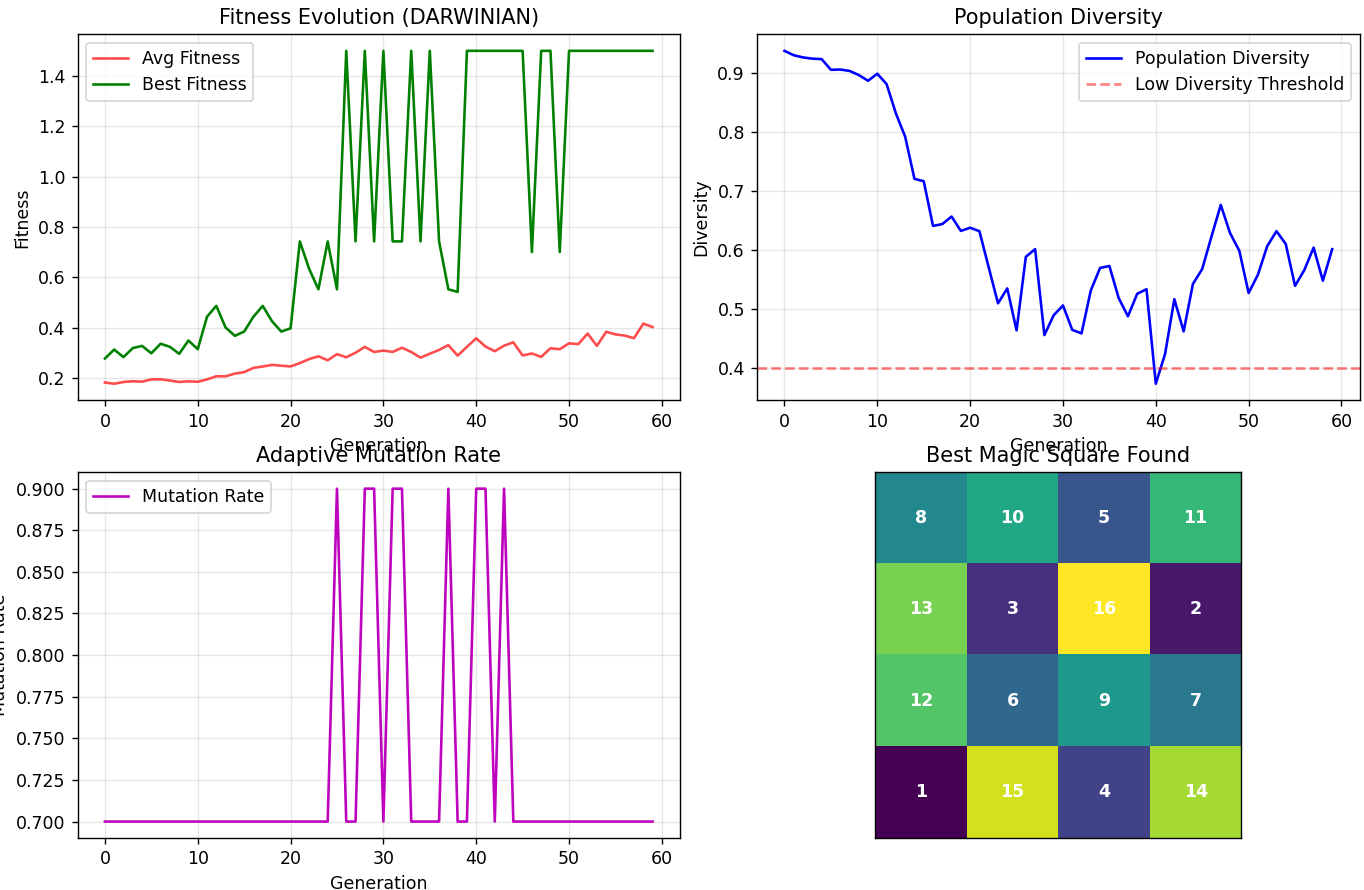
האלגוריתם מוצא לעיתים נדירות ריבוע קסם גם כשמשתמשים באבולוציה רגילה אך משמעותית לאט יותר (בהמשך הדוח נראה תוצאות עם למארק בקשר לריבוע קסם מושלם)



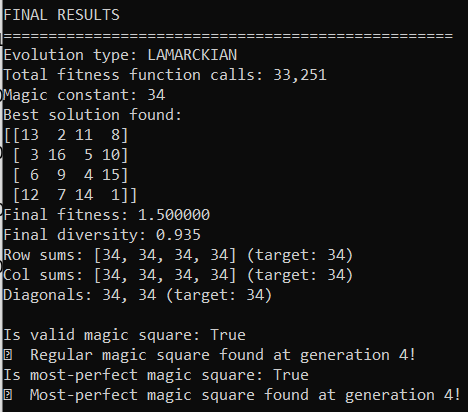


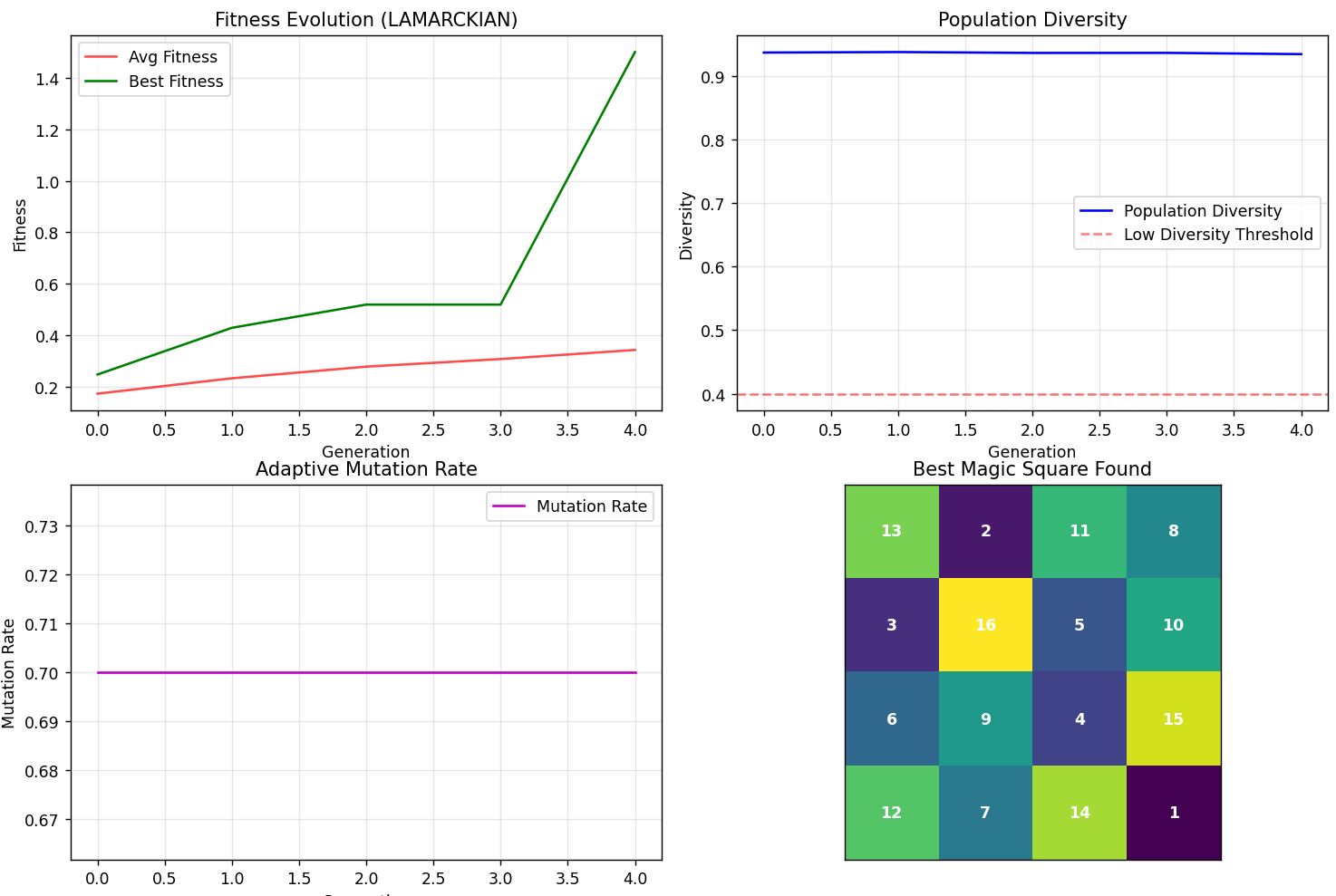
ב 4 נראה שדארווין מתחיל להתגבר על הקלאסי אמנם מספר הקריאות לפונקציית ההערכה גדולה אקספוננציאלית, אך הוא מוצא הרבה יותר פעמים ריבוע קסם וגם ריבוע קסם מושלם באופן קונסיסטנטי:



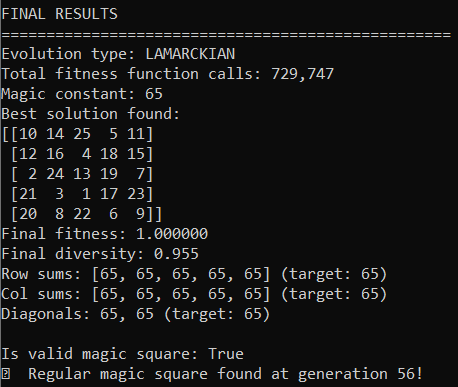


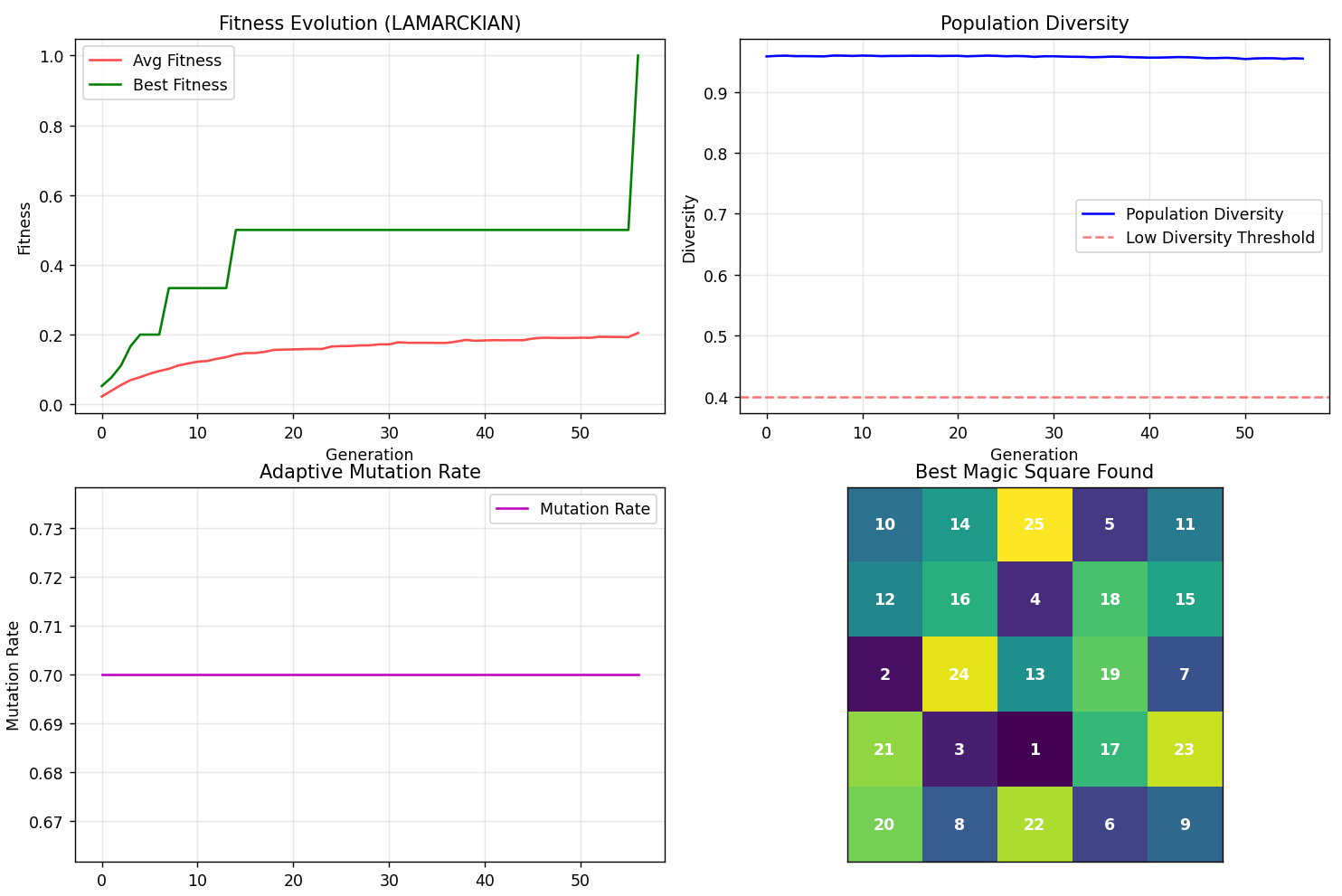
למארק ב4 מוצא בקלות ריבוע קסם + ריבוע קסם מושלם במספר מועט מאוד של דורות הודות לאופטימזציה:





כאשר מנסים למצוא 5\*5 באבולוציה רגילה/דרוויניסטית בדרך כלל לא מוצאים, כשהשתמשנו בלמארק הצלחנו למצוא בצורה יותר עקבית (בערך 1 מתוך 3 נסיונות) הודות לאופטימיזציה הלוקלית.





מצאנו שבלמארק אליטזים גדול עוזר לנו משמעותית כיוון שהשיפור הלוקאלי משפר עוד ועוד את הפרטים שנשמרו בכל שלב, ולכן גם השונות של האוכלוסיה לא משתנה מהותית.

לסיכום ניתן לראות שב3 על 3 האלגוריתם הקלאסי הוא האופטימלי לשימוש כאשר משקללים את היעילות וזמן הריצה ודארווין הגרוע ביותר, ב4 ו-5 ניתן לראות כי ההסתברות למציאת פתרון של דארווין טובה בהרבה משל הקלאסי אך למארק נותן ביצועים הרבה יותר טובים בהם הן מבחינת נכונות הפתרון והן מבחינת זמן ריצה.

טבלאה מסכמת:

|  |  |  |  |
| --- | --- | --- | --- |
| למארקי | דארוויני | קלאסי |  |
| 3: תמיד  4:תמיד  5: שליש מהפעמים | 3: תמיד  4: לעיתים קרובות  5: לעיתים נדירות | 3: רוב הפעמים  4: לעיתים רחוקות  5: לא | הצליח למצוא פתרון |
| תמיד | לעיתים קרובות | לעיתים רחוקות | פתרון מושלם? עבור 4 |
| 3: ~6,000  4: ~30,000  5: ~700,000 | 3: ~60,000  4: ~600,000  5: ~6.5 מיליון | 3: 1000  4: ~70,000  5: ~70,000 | מספר קריאות ממוצע לפונקציית ההערכה |
| 3: 1  4: 5  5: 70 | 3: 32  4: 80  5: 500 | 3: 9  4: 500  5: 500 | מספר דורות ממוצע  (500 זה החסם העליון) |