

תשובות-תרגיל 2:

1. א. יצוג כל פתרון נבחר להיות מיוצג ע"י מטריצת numpy דו מימדית, כאשר כל תא ריק מיוצג ע"י 0 כנדרש (עד יצירת אוכלוסיית הפתרונות). כל פתרון מוחזק ברשימה ששמה population .

ב. פונקציית ההערכה הינה fitness(detail).
פונקציה זו מקבלת פתרון (פרט) וסופרת את הכפילויות שלו בשורה בעזרת הפונקציה rowCheck(detail, row_num) וסוכמת את סך הכפילויות בכל השורות, וכן"ל לגבי העמודות ו- colCheck(detail, col_num). וכפילויות בריבועים הקטנים ו- squareCheck(detail, row_nu, cell_num) שבדוק אותם. לבסוף פונקציית ההערכה תחזיר 243 פחות סכום הכפילויות בכל השורות, עמודות והריבועים הקטנים של 3X3. כאשר סכום הכפילויות הוא 0 זאת אומרת שהסודוקו פתור, ופונקציית ההערכה תחזיר 243 (3*81). לכן התנאי לפתרון הוא "אם פונקציית ה-fitness מחזירה את הערך 243, עצור את התוכנית והדפס את הפיתרון.

פונקציית ההערכה היא:
$$f(x) = 243 - (\sum_{i=0}^8 |Gi| + \sum_{j=0}^8 |Hj| + \sum_{k=0}^8 |Lk|)$$

כך ש-Gi הוא מס' המספרים היחודיים בשורה ה-i, Hj הוא מס' המספרים היחודיים בשורה ה-j ו-Lk הוא מס' המספרים היחודיים בריבוע ה-K.

ג. פעולת המוטציה מתבצעת בפונקציה mutation(detail, zeroPlacesList).
רשימת תאי האפס בקלט נועדה על מנת לא לעשות מוטציות בתאים הקשיחים. בפונקציה ישנה לולאה שרצה בהתאם לאחוזים שניתנו בקבוע MUTATION_RATE. במהלך כל איטרציה נבחר באקראי תא אשר ניתן לבצע עליו מוטציה, ונבחר באקראי לאיזה ערך התא ישתנה. יש 88.9% שהתא ישתנה למספר שונה ממה שהוא היה קודם, ו-11.1% שהמספר האקראי שיבחר הוא גם המספר המקורי ולכן התא לא ישתנה.

ד. פעולת השחלוף בין הפתרונות התבצעה בפונקציה crossover(detail1, detail2). הפונקציה מקבלת שני פתרונות הורים ששרדו וזוכים להתרבות לדור הבא. מוגרל באקראי מספר שורות שאנו נרצה לשחלף בין ההורים, ז"א היכן תתבצע נקודת השבירה בכרומוזום, ואז מתבצעת לולאת השחלוף. יש לשים לב כי שחלוף שורות שונות בפרטים שונים יצור בילבול ב"מספרים הקשיחים" (אלה שלא היו 0 בסודוקו הקלט), לכן בוצע שיחלוף של אותה מספר שורה. הפונקציה מחזירה צאצא כאשר בסבירות גבוהה (87.5%) הוא מוכלא ברמה זו או אחרת בין הוריו. אם מספר השורות המוכלאות הנו 0 (הסתברות של 12.5%) אז הצאצא יהיה העתק ישיר של הפרט הראשון בקלט.

ה. לאחר שהשורדים מהדור הקודם נבחרו ונמצאים ברשימת fit[], ניגש ליצור את הדור הבא. מיקום הפרט עם הציון הטוב ביותר נשמר ב-best_index והוא מועתק ומהווה 20% מהדור הבא. בשאר ה-80% מצטרפים פתרונות בזוגות כאשר הפתרון הראשון הנו הכלאה של פרט עמיד אקראי מתוך ה-fit[], אם פרט עמיד נוסף לא אקראי (ה-i בלולאה). הפתרון השני הנו צאצא שחלוף בין פרט עמיד אקראי לפרט ה-i, אך הפעם עם פעולת המוטציה שנעשית עליו. לאחר ניסוי וטעייה הוחלט כי כאשר מוטציות מתבצעות על תוצרי שחלוף לעתים אז התוצאות גבוהות יותר. תוצרי השחלוף הנם של פרטים עמידים באוכלוסיה אשר אחד מהם נבחר באקראי על מנת להרחיב ככל הניתן את הקומבינציות האפשריות בין הפרטים העמידים.

ו. מדיניות האליטיזם מוסיפה מאוד לגרף השיפור של כל דור כיוון שאנו יכולים להיות בטוחים שמוטציות או שחלופים לא יפגמו לנו בתוצאה הטובה ביותר. האליטיזם בא לידי ביטוי בכך ש-20% מהאוכלוסיה של הדור הבא הנה הפרט הכי עמיד. בנוסף לכך גם כאשר מגיעים ל-local optima או התכנסות מוקדמת ומתבצעת זריעה מחדש, עקרון האליטיזם גם נשמר על מנת לא לדרדר את האוכלוסיה לגמרי (פירוט בשאלה לגבי התכנסות מוקדמת).

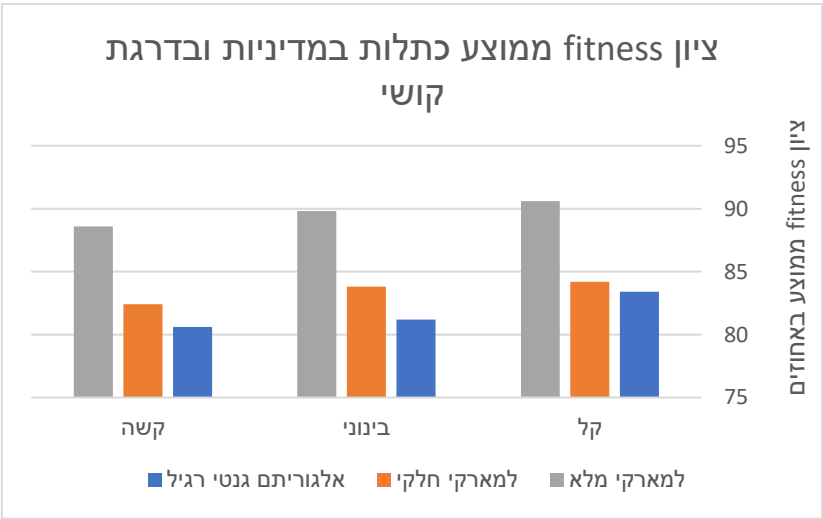
2. פונקציית האופטימיזציה הינה `optimization(detail, zeroPlacesList, type)`. פונקציה זו מקבלת את הפרט, את רשימת התאים הגמישים שהיו תאים ריקים (אפסים) כאשר התקבל הקלט הראשוני של הסודוקו, ואת `type` שהוא המצב שנבחר על ידי המשתמש מבין אלגוריתם למארקי חלקי (`type = 2`) או אלגוריתם למארקי מלא (`type = 3`).
 הפונקציה עוברת בלולאה על כל המספרים, אך מבצעת את השינויים רק על התאים הלא קשיחים. כאשר היא נתקלת בתא לא קשיח, היא מחשבת האם המספר שנמצא כרגע בתא זה מועיל לפתרון או לא. אם מספר זה חוזר על עצמו בטור, עמודה או ריבוע, אז התועלת שלו לציון היא 1-1. "ז"א הציון לכל מספר קיים יכול לנוע בין 0 לבין 3- (`cur_score`). כעת, בלולאה הכי פנימית הפונקציה תעבור על אילו מספרים כדאי לה להכניס במקום המספר הקיים. הציון מחושב ל- "`rep_score`". וכך יהיה מעבר על כל התאים וכל אפשרויות ההחלפה, כאשר לבסוף נבחר תא אחד שמביא את התועלת המקסימלית לציון והוא זה שמוחלף.
 אם הקלט בסוג האלגוריתם יהיה 2, אז הפונקציה תחזיר אך ורק את ציון ההערכה החדש כנדרש ללא השינוי של הפרט. אם הקלט יהיה 3 אז הפונקציה תחזיר את הפתרון המשופר.
3. מהטבלה המצורפת למטה אנחנו יכולים לראות שבמדיניות האלגוריתם הרגיל, ציון ה-`fitness` נמצא בטווח מספרים נמוך יחסית (הציון הגבוה ביותר הוא 83%). לעומת זאת, באלגוריתם למארקי מלא, ציון ה-`fitness` נמצא נמצא בטווח הגבוה ביותר. טווח הציונים באלגוריתם למארקי חלקי נמצא נמוך מאלגוריתם למארקי מלא וגבוה מאלגוריתם רגיל. כלומר, יש צורך ביותר דורות על מנת להגיע לפתרון באלגוריתם הגנטי הרגיל והכי פחות דורות באלגוריתם למארקי מלא. ניתן להסביר את ההבדל ע"י פונקציית האופטימיזציה.
 באלגוריתם גנטי רגיל, אין התערבות מכוונת הגורמת לשינוי גנטי וכל שינוי מספר הינו מאורע אקראי שמתבצע ע"י מוטציות רנדומליות או שחלופים רנדומלים. לעומת זאת, באלגוריתם למארקי חלקי, יש שימוש בפונקציית אופטימיזציה שעל פיה ייקבע ציון ה-`fitness`. על כן, ניתן לראות עליה בטווח הציונים הממוצע אך עם זאת הדור הבא של הפתרונות נקבע על פי הפתרונות לפני האופטימיזציה. באלגוריתם למארקי מלא, לא רק שקיימת התערבות מכוונת ע"י פונקציית אופטימיזציה, הדור הבא נקבע על פי הפתרונות לאחר האופטימיזציה. לכן, הם הקרובים ביותר לפתרון הסודוקו ובעלי ציון ה-`fitness` הגבוה ביותר.
4. ככל שרמת הקושי עולה, יש צורך ביותר דורות על מנת להגיע לפתרון הסודוקו. ההבדל נובע מכך שכל שרמת הקושי יותר קלה, יש פחות משבצות ריקות ויותר מספרים קבועים. כתוצאה מכך, היכולת של האלגוריתם להגיע לפתרון הסודוקו תוך מס' דורות מוגבל עולה. כלומר, ככל שקיימות יותר משבצות למלא בסודוקו- האלגוריתם יצטרך לרוץ יותר זמן ויותר דורות על מנת להגיע לפתרון הסודוקו.
5. ההתכנסות המוקדמת טופלה באמצעות זריעה מחדש של אוכלוסיה באופן חלקי (90% אוכלוסיה חדשה) אשר יכולה להתרחש כעבור מספר דורות מסוים (המשתנה `conch` סופר את כמות הדורות הרצופים) שאין שינוי בו בפרט המוצלח ביותר. כאשר מטפלים בבעיה זו כך, חוץ מפתרון להתכנסות המוקדמת זה גם יכול לעזור כאשר מגיעים ל-`local optima` ואין אצל הדור את החומר הגנטי הנדרש על מנת לשפר עוד את הפרטים שלו, והמוטציות לא מהוות פאקטור כי ככל שהפרטים בדור משתפרים, כך מצטמצמים סיכוייה של מוטציה אקראית (או שחלוף) לשפר את הפרט, וגדלים הסיכויים שמוטציה זו רק תהרוס. לכן אנו מזריקים חומר גנטי חדש לאוכלוסיה, כאשר 10% יהיו הפרטים הכי טובים מהדור הקודם, ושאר ה-90% מהדור יתמלא בפרטים אקראיים חדשים שיוולדו בעזרת פונקציית ה-`population_generator(sudokuMat)`.

טבלת נתונים של 5 ניסיונות הרצה:
מאפיינים: 50 דורות, 3% מוטציות, 30% שכפול אליטיסטים.

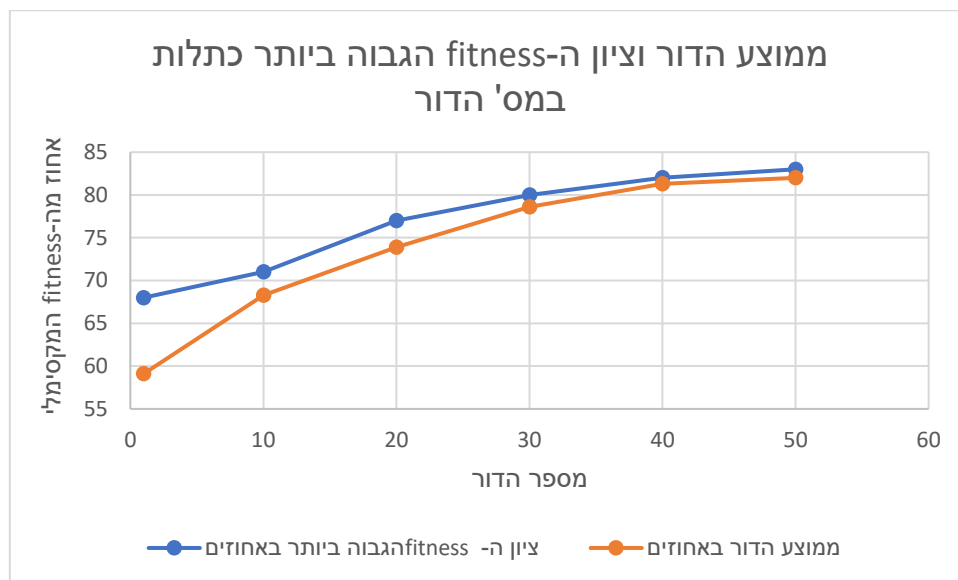
פתרון למארקי חלקי						אלגוריתם גנטי רגיל						
ממוצע דור אחרון			ציון הפרט המוצלח			ממוצע דור אחרון			ציון הפרט המוצלח			
קשה	בינוני	קל	קשה	בינוני	קל	קשה	בינוני	קל	קשה	בינוני	קל	
82.10%	78.90%	83.70%	85%	85%	86%	80.90%	79.50%	78.60%	82%	82%	82%	ניסיון 1
82.60%	81.90%	81.90%	84%	83%	83%	64.10%	80.40%	83.30%	78%	81%	84%	ניסיון 2
80.90%	81.20%	81.30%	82%	83%	83%	80.50%	81.40%	66.70%	81%	82%	83%	ניסיון 3
79.20%	79.40%	82.70%	81%	84%	85%	79.20%	77.90%	83.80%	80%	79%	85%	ניסיון 4
78%	79%	82.60%	80%	84%	84%	78.30%	81.30%	82.70%	82%	82%	83%	ניסיון 5
80.56%	80.08%	82.44%	82.40%	83.80%	84.20%	76.60%	80.10%	79.02%	80.60%	81.20%	83.40%	ממוצע

פתרון למארקי מלא						
ממוצע דור אחרון			ציון הפרט המוצלח			
קשה	בינוני	קל	קשה	בינוני	קל	
81.20%	83.90%	87.90%	88%	92%	89%	ניסיון 1
85.70%	87.70%	88.70%	87%	89%	91%	ניסיון 2
81.80%	84.70%	90.40%	89%	91%	92%	ניסיון 3
87.50%	85.40%	89.20%	88%	87%	90%	ניסיון 4
89.10%	83.50%	89.60%	91%	90%	90%	ניסיון 5
85.06%	85.04%	89.16%	88.60%	89.80%	90.60%	ממוצע

ציון fitness כתלות במדיניות ובדרגת הקושי:
מאפיינים: 50 דורות, 3% מוטציות, 30% שכפול אליטיסטים.



ממוצע הדור וציון ה-fitness הגבוה ביותר כתלות במס' הדור:
 מאפיינים: דרגת קושי קלה, 50 דורות, 3% מוטציות, 30% שכפול אליטיסטים.



ניתן לראות שקיימת עליה באחוז ה-fitness יחד עם העליה במס' הדורות. גרף זה מייצג את ההתנהגות ברמת קושי קלה ותקפה גם לשאר דרגות הקושי.