

עבודה באופטימיזציה

מימוש אלגוריתם גנטי לפתרון בעיית אופטימיזציה

תמר ינץ ת.ז. 207139940
עודד ליפמן ת.ז. 060516119

תקציר

מאז ומתמיד רצה האדם לטוס ולהשתלב בשמים – מהאחים רייט בטיסה המאוישת הראשונה בשנת 1903, דרך מטוסים ומסוקים מאוישים ועד לכלי טיס בלתי מאוישים (כט"מ"מים). בעשור האחרון אנו עדים להתפתחות משמעותית – הרחפן. הרחפנים הינם כלי טייס בלתי מאוישים בעלי יכולת להמריא ולנחות אנכים ולרחף באוויר (כמו מסוק). הרחפן הנגיש את היכולות האוויריות לכל החפץ בכך – כלי פשוט למשתמש, בעל שימושים רבים (צילום, ביצוע משלוחים ואפילו תקיפה) ובעלות נמוכה. כיום מתפתחים בעולם שימושים רבים לרחפנים – משלוחים של תרופות, משלוחי מזון, שליחויות דואר, ריסוס חקלאי ועוד. בעוד עולם התעופה המאויש הינו סדור ומאורגן בנהלים ברורים – נתיבי תעופה מוגדרים, מרשה טיסה לכל טיסה וכמות כלים "נשלטת", השימוש הגובר ברחפנים מציב אתגרים רבים באופן בו ניתן "לסדר" את השמים כך שרחפנים יבצעו את כלל המשימות להם הם נדרשים בבטיחות. דוגמאות לבעיות כאלו ניתן למצוא בצה"ל אשר מוטרד ומחפש פתרונות למניעת התנגשות בין רחפני כלי טייס מאוישים הטסים נמוך (לדוגמה מסוק חילוץ אשר טס בקרבת רחפן). דוגמה נוספת, אשר נוגעת בקשר ישיר לעבודה זו, ניתן לראות במיזם אשר מבוצע בימיי אלו ע"י משרד התחבורה, חברת נתיבי אילון ובשיתוף רשות החדשנות: מיזם הרחפנים הלאומי (ניידות עירונית במרחב האווירי). מטרת המיזם הינה לבחון את אופן ההטסה של מספר רב של רחפנים באזור אורבני, באופן בטוח. הפוטי' הגלום באלטרנטיבה של משלוחים ע"י רחפנים הוא עצום. מפעיל אחד עתיד לשלוט על כ-10 רחפנים במקביל, אשר מבצעים 10 טיסות ביום כל אחד וע"י כך לאפשר הורדה משמעותית במחירי השליחויות, זמני השליחות וזמינות ה"שליחים". בעבודה זו עשינו רידוקציה לאתגרים אשר ציינו כאשר בנינו בעיה של חברת משלוחים נתונה אשר לה 40 חנויות ו-80 לקוחות אותם היא נדרשת לשרת במרחב גיאוגרפי נתון אשר תואר ע"י מטריצה בגודל 320×320 . כל תא במטריצה הינו נקודת ציון אפשרית למיקום חנות או לקוח. ציינו מספר "חוקי משחק" לבעיה אותה אנו מעוניינים לפתור. חוקי המשחק הינם בהקשר ישיר לאתגרים במציאות ומביאים לידי ביטוי חיסכון בכסף, חסכון באנרגיה של הרשת שמפעילה את החנויות והרחפנים ומביאים לידי מקסום את שביעות הרצון של הלקוחות ע"י משלוח מהיר. בחרנו לפתור את הבעיה ע"י אלגוריתם גנטי אשר משתפר ומביא פתרון טוב יותר בכל דור (בכל איטרציה). האלגוריתם הגנטי מורכב ממספר שלבים. נדרשנו להנדס כל שלב בכדי להתאים לבעיה אותה הגדרנו. בנוסף לכך לאחר שלב הבניה הראשוני טייבנו את תוצאות האלגוריתם ע"י ביצוע שינויים והתאמות אשר כללו שינויי היפר פרמטרים, הכנסת אקראיות ובחינות מימוש שונות לשלביו השונים של האלגוריתם הגנטי. מימוש האלגוריתם בוצע ע"י שימוש בתוכנת פיתון וכלל העזרות בספריות חיצוניות. הקוד זמין ב[google colab](https://colab.research.google.com/).

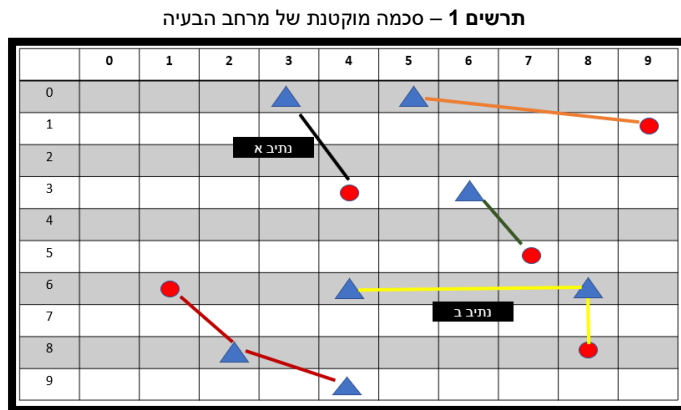
1. מבוא

בעיות אופטימיזציה הינן בעיות אשר נדרשים להתמודד עימן בתחומי חיים רבים. בעבודה זו אנו מתארים בעיה הלקוחה מהעולם האמיתי ומתעצמת בשנים האחרונות ונוגעת בשימוש ושילוב רחפנים במרחב האווירי. לאחר תיאור ואפיון הבעיה בפרק 2 ניגשנו לפתרון הבעיה. לרוב בפתרון בעיות אופטימיזציה נרצה למצוא את הפתרון המיטבי – במידה וקיים פתרון כזה. בבעיה אותה תיארונו קיים פתרון מיטבי (לבעיה מספר סופי של פתרונות ובאופן תיאורטי ניתן לעבור על כל הפתרונות האפשריים ולבחון מיהו הפתרון האופטימלי), אך כאשר כמות הפרמטרים בבעיה עולים המורכבות למציאת פתרון זה הינה ברמת סיבוכיות גבוהה ודורשת זמני חישוב והרצה גבוהים. על כן בחנו מספר אלגוריתמים לפתרון הבעיה – אלגוריתם שידוך מקסימלי, אלגוריתם KNN ואלגוריתם גנטי. בפרק הדיון, פרק 5, נפרט את השיקולים אשר גרמו לנו לבסוף להשתמש במשפחת האלגוריתמים הגנטיים אשר מביאים לאורך מימושן פתרונות טובים יותר בכל איטרציה ואיטרציה עד אשר נקבע תנאי עצירה. בפרק 3 פירטנו על האלגוריתם הגנטי הכללי ובהמשך הפרק הרחבנו על המימוש הפרטי אותו יישמנו. פרק 4 מפרט את התוצאות המרכזיות של העבודה ובפרק 5 ו-6 קיימנו דיון על האלמנטים המרכזיים וסיכמנו את העבודה.

2. פירוט בעיית האופטימיזציה וחוקי המשחק:

נתונה עיר בה יש כמות נתונה של חנויות (40) השייכת לרשת (לדוגמה Amazon) וכמות משתנה של לקוחות (80). כל לקוח מבצע הזמנה טלפונית ממרכז השירות של הרשת, לא מחנות ספציפית, ומזמין לעצמו מוצר מהרשת. את המוצר מקבל הלקוח ע"י טיסת רחפן מאחת מחנויות הרשת. לטובת מורכבות הבעיה כלל 80 ההזמנות מבוצעות באותו רגע נתון. לבעיה שהגדרנו יש מספר חוקי משחק, המתבססים על הבעיות הקיימות בשוק. חוקי המשחק נדרשים בכדי לקבוע את בעיית האופטימיזציה.

- מרחב הבעיה אשר כולל: רשת חנויות, לקוחות ורחפנים אשר מבצעים שליחויות:
- (1) מרחב העיר בה מבוצעים המשלוחים הינה מטריצה בגודל 320×320 . כל ריבוע במטריצה הינו נקודת ציון בה ניתן למקם חנות או לקוח.
 - (2) בתחילת האלגוריתם אנו ממפים באופן אקראי 40 חנויות השייכים לרשת ו-80 לקוחות במרחב העיר. כל לקוח וחנות מוצבים בנקודת ציון. כל לקוח מבצע הזמנה אחת מהרשת (הלקוח אדיש לחנות תאשר תספק לו את ההזמנה, ההזמנה היא מהרשת שכוללת 40 חנויות).
 - (3) רחפן לוקח משלוח מנקודת ציון של חנות לנקודת ציון של לקוח. רחפן אחד יכול לקחת עד 3 משלוחים, כלומר נתיב מחנות ללקוח ראשון, משם ממשיכים ללקוח השני ואז לשלישי.
 - (4) הטיסה הינה בקווים ישירים, המרחק הינו מרחק אוקלידי.
 - (5) כל לקוח מבצע הזמנה אחת בלבד. במידה ומבוצעות מאותה נקודת ציון 2 הזמנות (לקוח מבצע 2 הזמנות), נחשיב זאת כשני לקוחות שונים.
 - (6) לא קיימת מגבלת רחפנים לחנות, כלומר מחנות אחת ניתן להוציא לטובת משלוחים כל כמות רחפנים שתיבחר.



בתרשים 1 סכמה המתארת עיר בגודל 10×10 . העיגולים האדומים מסמנים חנויות, המשולשים מסמלים לקוחות. בתרשים רואים לדוגמה נתיב מחנות אשר נמצאת בנקודת ציון (8,8) ל-2 לקוחות בנתיב שעובר תחילה בנקודה (6,8) ומשם לנקודה (6,4) (מסומן בתרשים כנתיב ב). עוד נתיב הינו מחנות (3,4) אשר מבצעת משלוח ללקוח (0,3). לאחר קבלה של כלל ההזמנות – הצבת כלל הלקוחות במטריצה שמייצגת את העיר – נדרש לפתור בעיית אופטימיזציה למציאת פתרון מיטבי לביצוע המשלוחים. פתרון מיטבי הינו פתרון אשר ממקסם את התועלת הכלכלית של הרשת - מיני שימוש באנרגיה של הרחפן ומיני רחפנים ופתרון המספק את רצונו של הלקוח – זמן מזערי לקבלת המשלוח. הפונקציה אותה הגדרנו שנדרש למזער הינה:

min

$$f = \text{total distance} * \text{transition time} - \alpha * \text{number of shops price}$$

where

$$\text{number of shop price} = \sum_{i=1}^3 \text{delivery time}_i * \text{costumer}_i$$

המרכיב הראשון של פונקציה f נועד לצמצם את המרחק שהרחפן טס. נשים לב כי המרחק וזמן הטיסה הם בקשר ישיר ועל כן נתונים מענה גם לדרישת הרשת בחסכון באנרגיה וגם ללקוח אשר מקבל משלוח מהיר: המשתנה total distance מתאר את המרחק אותו טס הרחפן בעוד המשתנה transition time הינו היפר פרמטר המגדיר מהי "העלות" לכל יחידת מרחק - כל כל תא במטריצה. כפי שצוין רחפן בודד יכול לעשות משלוח לחנות אחת בלבד, ל-2 חנויות או ל-3 חנויות. המשתנה $\text{number of shops price}$ נועד בכדי למזער את הפונקציה כאשר יש שימוש ברחפן אחד למספר לקוחות רב וזאת לאור העובדה כי הרשת מעוניינת לצמצם את השימוש בכמות הרחפנים משיקולי התייעלות כלכלית. משתנה זה מורכב מ- costumer_i – כמות הלקוחות אשר מקבלים את המשלוח במקום ה- i . כאשר לדוגמה $\text{costumer}_3 = 6$, הכוונה היא ש-6 לקוחות מקבלים את המשלוח שלהם ע"י רחפן שעבר לפני שהגיע אליהם אצל 2 לקוחות שונים. את כמות הלקוחות מכפילים בהיפר פרמטר delivery time_i . ככל ש- i גדל ערך הפרמטר הנ"ל גדל ולאור זאת רחפן אשר מבצע נתיב מלא ל-3 לקוחות ממזער את הפונקציה יותר מרחפן אשר מבצע משלוח ללקוח למספר קטן יותר של לקוחות.

3. שיטה

3.1 אלגוריתם גנטי, genetic algorithm:

בפרק זה נפרט בקצרה על מימוש אלגוריתם גנטי באופן כללי, בפרק הבא נפרט את המימוש הפרטני אשר יצרנו בכדי לפתור את בעיית הרחפנים. אלגוריתמים גנטיים הינם משפחה של אלגוריתמים אשר משמשים לביצוע אופטימיזציה ע"י חיפוש אדפטיבי, תוך שימוש ברעיון המבוסס על גנטיקה של אוכלוסייה ועקרון הברירה הטבעית. האלגוריתם מורכב משלבים. בשלב ראשון קיים סט של פתרונות אשר נקרא אוכלוסייה, פתרון מוגדר ע"י כרומוזום. גודל

¹ Kumar, Manoj, et al. "Genetic algorithm: Review and application." Available at SSRN 3529843 (2010).

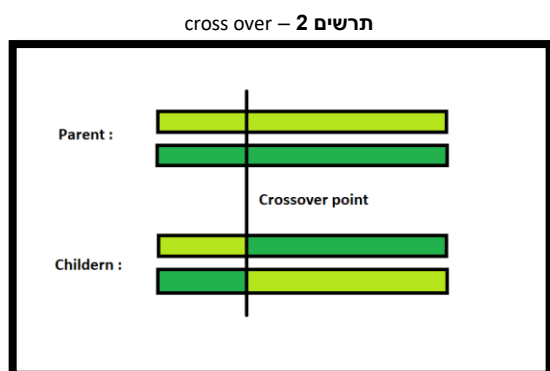
האוכלוסיה מוגדר לכל דור וניתן לשינוי. בכל דור נבדקת איכות הפתרון ע"י פונקציית fitness ואז ע"י מודל הסתברותי נבחרים הכרומוזומים אשר עוברים דור. חלק מהכרומוזומים "מזדווגים" ומייצרים צאצאים "offspring". כאשר מייצרים צאצאים מתבצעים שני תהליכים – crossover ו mutation.

מכיוון שבכל דור בוחרים מהאוכלוסיה, בסבירות גבוהה יותר, את הכרומוזומים בעלי ערך ה fitness הגבוה ביותר ומהאחרונים גם "נולדים" צאצאים חדשים קיימת סבירות גבוהה כי בכל דור נתוני האוכלוסיה ישתפרו, קרי ערכי ה fitness שלהם יהיו גבוהים יותר. התהליך של יצירת דורות חדשים נמשך עד אשר מגיעים לתוצאה הרצויה – ערך fitness מספק (נשים לב כי בתיאור הכללי של הבעיה התייחסו לכרומוזום טוב ככרומוזום עם ערך fitness גבוה. בבעיה הפרטנית שלנו שתפורט בהמשך היעד הינו מזעור ה fitness ולכן נתייחס לכרומוזום טוב ככרומוזום עם ערך fitness נמוך).

3.2 פירוט האלגוריתם עפ"י השלבים השונים:

אתחול: מאתחלים אוכלוסיה ראשונית – מספר כרומוזומים מוגדר. לרוב מדובר במאות או אלפים. זהו תהליך רנדומלי אך ייתכן כי חלק מהכרומוזומים יבנו תוך בנייה של פתרון אשר קרוב יותר לפתרון אופטימלי.

בחירה: כל כרומוזום מקבל "ציון" המוגדר ע"י fitness function. לאור הציון של הכרומוזומים השונים מבוצעת בחירה של הכרומוזומים הנבחרים. אלו מיועדים למעבר לשלב הבא או לחלופין להוות הורים לצאצאים. לרוב מבצע שימוש גם בכרומוזומים עם ציון נמוך יותר וזאת בכדי לייצר גיוון ולמנוע התכנסות לפתרון שאינו מיטבי.



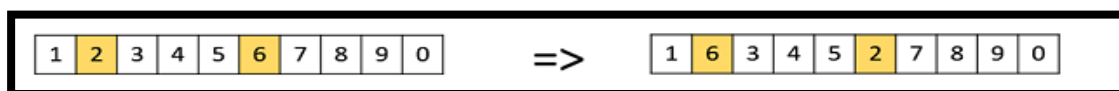
יצירת דור חדש: נבחרים זוג הורים ומהם מייצרים צאצאים ע"י 2 שיטות מרכזיות: Crossover – לרוב כרומוזום בנוי כמערך. בשלב ה crossover בוחרים מהורה אחד (אבא) חלק מהמערך ומההורה השני את שאר המערך. כמתואר באיור מספר 2.

Mutation – בשלב זה מבצעים שינוי בכרומוזום באופן רנדומלי לטובת יצירת שינוי חדש. לדוגמה – שינוי ערך של תא במערך הכרומוזום, החלפת שני ערכים וכו' (דוגמה בתרשים 3)

בסיום תהליך הבחירה ויצירת הצאצאים נוצר דור חדש והתהליך חוזר על עצמו עד אשר מתקבלת תוצאה מספקת לפתרון (כרומוזום) מיטבי.

עצירת האלגוריתם יכולה להתבצע במספר שלבים – כמות איטרציות מוגדרת, כאשר מגיעים לסף מסוים בפונקציית ה fitness או לחלופין כאשר השונות בין דור לדור נמוכה.

תרשים 3 – mutation



אילוצים² – ניתן להכניס אילוצים לאלגוריתם. אלו יבואו לרוב לידי ביטוי כפונקציות "ענישה" (penalty) בפונקציית ה fitness ובכך האילוף ייושם. יש מגוון דרכים ליישום אילוצים ביניהן:

- death penalty – מצב בו כאשר האילוף לא מתקיים ערך פונ' ה fitness הוא אינסופי ובכך הכרומוזום שאינו באילוצים לא נבחר

$$fitness(a) = f(a) + p(a) \text{ s.t } p(a) = \begin{cases} \infty & \text{if } a > \text{constrian} \\ 0 & \text{if } a \leq \text{constrian} \end{cases}$$

- static penalty – עונש קבוע, אשר אינו תלוי בדורות לכל לאילוף שאינו מתקיים

$$fitness(a) = f(a) + \sum_{i=1}^k d_i$$

K מספר האילוצים ו d_i יכול להיות מספר קבוע c או לחלופין $|h(x)|$ כאשר $h(X)$ פונקציה אשר מענישה ע"י המרחק מהאילוף.

- dynamic penalty – ענישה משתנה מבוססת מרחק

$$fitness(a, t) = f(a) + \sum_{i=1}^k s_{(i)}(t) * d_i$$

כאשר s פונ' מונוטונית עולה ככל גדל t ו- d_i הינו משתנה כפי שהוגדר בסעיף הקודם.

3.3 מימוש אלגוריתם גנטי לטיוב משלוחי הרחפנים

² Coit, David W., and Alice E. Smith. "Penalty guided genetic search for reliability design optimization." *Computers & industrial engineering* 30.4 (1996): 895-904.

את בעיית האופטימיזציה איתה אנו מתמודדים נפתור ע"י אלגוריתם גנטי. בפרק זה נפרט את המימוש הפרטני שלנו עפ"י השלבים השונים שפורטו בסעיף 3.2.

אתחול:

העיר - מטריצה בגודל 320×320 . 80 לקוחות אשר מיקומם נבחר רנדומלית בתא במטריצה אשר מייצג 2 נקודות ציון. יתכן מצב בו בתא אחד המטריצה יש מספר לקוחות = מספר הזמנות. 80 הלקוחות מיוצגים ע"י מספרים אי זוגיים מ-1 ועד 159. כל מספר מייצג לקוח. הרשת מורכבת מ-40 חנויות. החנויות מוצבות באופן רנדומלי כפי שהלקוחות מוצבים. לא יתכנו 2 חנויות באותה נקודת ציון (תא במטריצה). החנויות מיוצגות ע"י מספרים זוגיים מ-2 ועד 80.

דור 0, אוכלוסייה ראשונית: הגדרנו אוכלוסייה ראשונית המונה 1600 כרומוזומים (גודל האוכלוסייה משתנה בהמשך יצירת הדורות. יפורט בפרק התוצאות). כל כרומוזום הוגדר כמערך בגודל מספר החנויות כפול 4 - 160. המערך מחולק לרביעיות (40 רביעיות). כל רביעייה מתארת נתיב - נקודת מוצא של הרחפן (חנויות) ועד 3 נקודות למשלוח (לקוחות). במערך במקומות ה- i^*4 , $i \in [0,1,\dots,39]$, הצבנו את החנויות ובשאר המקומות הצבנו את לקוחות. נשים לב שמספר הלקוחות הינו קטן מכמות התאים שנותרה (נותרו 120 תאים בעוד יש רק 80 לקוחות). במקומות בהם לא הוצב לקוח הוצבה הספרה 0:

2	1	7	3	4	5	9	15	6	0	0	0	8	11	13	0	10	17	0	0
---	---	---	---	---	---	---	----	---	---	---	---	---	----	----	---	----	----	---	---

הסבר לדוגמא: התאים האדומים במערך מסמנים את החנויות. "שם" החנות מסומן במספרים זוגיים. התאים הכחולים מסמנים לקוחות, אשר מסומנים במספרים אי זוגיים, בתאים הנותרים הצבנו את הספרה 0. המערך מתאר רחפן שיצא מחנות "2" ועשה משלוח ללקוח "1", לאחריו ללקוח "7" ולבסוף ללקוח "3". הרחפן מחנות "10" עשה משלוח ללקוח אחד בלבד - "17" ואז סיים (זו משמעות האפסים). נשים לב כי מחנות 6 לא יצא רחפן בכלל ולכן השלישייה אחרי חנות "6" מונה אפסים בלבד. אתחול האוכלוסייה הראשונית בוצע ב-3 שיטות שונות, בכל שיטה אותחלו חלק מהכרומוזומים של האוכלוסייה הראשונית:

(1) **אתחול A.** בשיטה זו החנויות הוצבו במקום ה- i^*4 בצורה רנדומלית, כך שיתכן ואותה חנות תוצב מספר פעמיים. הלקוחות הוצבו גם הם בצורה רנדומלית בתאים המיועדים להם. בתאים הרקים (40 תאים) הוצבו אפסים.

(2) **אתחול B.** שיטה זו דומה לשיטה A בהבדל אחד בלבד - אין חזרתיות בחנויות. כל חנות הוצבה בדיוק פעם אחת בלבד בכל כרומוזום.

(3) **אתחול C.** שיטה זו היא שיטה אשר בונה את הכרומוזום באופן שהוא, בשאיפה, יותר קרוב לפתרון האופטימלי. בשיטה זו חילקנו את העיר, המטריצה, ל-4 רבעים (רבע ראשון - כל התאים במטריצה שקטנים משורה מספר 160 ועמודה מספר 160, רבע שני כל התאים במטריצה שקטנים משורה מספר 160 וגדולים שווים לעמודה מספר 160 וכך הלאה לשאר הרבעים). כל כרומוזום נבנה באופן תורני כך שהחנויות בכל רבע יבצעו משלוחים ללקוחות אשר נמצאים ברבע שלהם. באופן כזה מראש לא יהיו מרחקים גדולים מידי שרחפן ידרש לטוס.

בניית הדור הבא הינו התהליך המורכב יותר אשר נדרשו לאפיין ולממש בעבודה זו. מימוש מוצלח של שלב זה קשור באופן חד חד ערכי להצלחת המודל. בחנו 4 אפשרויות שונות למימוש בניית הדור הבא ע"י חמישה אלגוריתמים שונים. סדר האלגוריתמים מעיד על איכותם ומורכבותם (כלל שמספור האלגוריתם גבוה יותר כך מורכבותו עולה וכמו כן הוא מביא לתוצאות טובות יותר). כל אלגוריתם התחיל משני כרומוזומים, שני הורים - לשם פשטות נכנה אותם "אבא" ו"אמא" אשר מייצרים כרומוזום חדש "ילד". אלגוריתמים אלו מממשים את שלב cross and mutation ומכאן ואילך נכנה אותם "שיטת הצרחה והחלפה".

(1) הצרחה והחלפה 1 - באלגוריתם זה מימשנו את שלב ה crossover ע"י פיצול הכרומוזום של האבא והאמא לשני חצאים (בוצע ע"י חלוקה של כל הורה לחצי ראשון ממקום 0 ועד מקום 79 במערך ממנו מורכב וממקום 80 ועד הסוף). הרציונל של אלגוריתם זה הינו ליצר ילד ע"י הרכבה של שני חצאים - חצי ראשון מהאבא וחצי שני מהאמא. בכדי לבצע הרכבה זו באופן חוקי - כלומר שהילד החדש יכלול את כל החנויות של מרחב הבעיה האלגוריתם נבנה באופן הבא: החצי הראשון של האבא מועתק לחצי הראשון של הילד. באשר לחצי השני של הילד שמועתק מהחצי השני של האמא, הנ"ל מבוצע בשלבים ובאופן המונע חזרה על לקוחות: תחילה מועתקות כל החנויות מהחצי השני של האמא ישירות לחצי השני של הילד, לאחר מכן מועתקים כל הלקוחות לאחר בדיקה שהלקוחות אשר מועתקים מהחצי השני של האמא לא הופיעו כבר בילד (כלומר לא הופיעו בחצי הראשון של האבא). עד כאן יצרנו ילד ע"י תהליך crossover אבל בילד הזה ייתכן וחסרים לקוחות. לכן בשלב האחרון שזהו שלב מותאם שבנינו כשלב mutation מוצבים בילד באופן רנדומלי החנויות החסרות. חנויות אלו מוצבות לאורך כל המערך בתאים הרקים.

דוגמא על מערך בגודל 16:

אבא:

6	1	7	3	4	5	9	0	6	0	0	0	8	11	13	15
---	---	---	---	---	---	---	---	---	---	---	---	---	----	----	----

אמא:

2	1	7	0	4	13	9	0	6	5	15	0	8	11	3	0
---	---	---	---	---	----	---	---	---	---	----	---	---	----	---	---

Crossover

שלב ראשון - לקיחת חצי מערך מהאבא:

6	1	7	3	4	5	9	0								
---	---	---	---	---	---	---	---	--	--	--	--	--	--	--	--

שלב שני – השלמת חצי מערך שנותר מהאמא ללא חזרתיות על לקוחות:

6	1	7	3	4	5	9	0	6	15			8	11		
---	---	---	---	---	---	---	---	---	----	--	--	---	----	--	--

Mutation – השלמת המערך עם לקוחות חסרים ואפסים במקומות שנותרו

6	1	7	3	4	5	9	0	6	15	0	13	8	11	0	0
---	---	---	---	---	---	---	---	---	----	---	----	---	----	---	---

(2) הצרחה והחלפה 2 – שכלול של "הצרחה והחלפה 1" ע"י שילוב אקראיות בשלושה שלבים. שלב ראשון שבו שילבנו אלמנט אקראיות הינו בסיום בניית הילד. לאחר שנוצר ילד חדש (עפ"י "הצרחה והחלפה 1") הילד מורכב מ-40 מסלולים (כל מסלול הוא בעצם רחפן שיוצא מחנות ל-1/2/3 לקוחות). באלגוריתם זה ביצענו mutation נוסף ע"י הצרחה בין מסלולים. כלומר ילד מהצורה:

6	1	7	3	4	5	9	0	6	15	0	13	8	11	0	0
---	---	---	---	---	---	---	---	---	----	---	----	---	----	---	---

הפך ע"י הצרחה של המסלול הראשון והאחרון במערך (לשם ההמחשה) לילד בצורה הבאה:

8	11	0	0	4	5	9	0	6	15	0	13	6	1	7	3
---	----	---	---	---	---	---	---	---	----	---	----	---	---	---	---

בשלב הבא ביצענו הצרחה בין לקוחות (בכדי להביא לכדי שינוי במסלולים). אלמנט האקראיות האחרון הושג ע"י החלפה אקראית של חנויות בחנויות ממאגר החנויות. כלל ההחלפות וההצרחות בוצעו באופן רנדומלי וסטטיסטי לאחר טיוב האלגוריתם.

(3) הצרחה והחלפה 3 – זהו אלגוריתם המשלב בבניית הילד את פונק' הfitness בכדי לאפשר בחינה חלקית כי אנחנו בוחרים את המסלולים הטובים ביותר. אופן המימוש: שלב crossover מבוצע ע"י פירוק של האבא והאמא ביחד ל-80 נתיבי טיסה (כמו שהוזכר כל נתיב הוא 4 משבצות במערך (חנות לקוחות + אפסים. סה"כ ההורים ביחד הם מערך בגודל 320 המכיל 80 נתיבים). את הנתיבים האלו מיינו בסדר עולה עפ"י ערך הfitness שכל נתיב מקבל. לטובת הילד בחרנו את 40 הנתיבים עם הציון הנמוך ביותר (נזכיר כי אנו ממזערים את פונ' הfitness). כמו בשיטות הקודמות שלב ה crossover יכול לייצר ילדים שאינם "חוקיים", כלומר ייתכן כי אחד יש חזרתיות על מספר לקוחות ומאידך ייתכן כי מספר חנויות השומטו. על כן לאחר סיום שלב ה crossover עברנו לשלב ה mutation אשר גם מכניס אקראיות וגם מוודא כי הילד "חוקי". בשלב זה מחקנו לקוחות אשר מופעים יותר מפעם אחת ובסיום הכנסנו למערך (לילד) את כל הלקוחות החסרים.

(4) הצרחה והחלפה 4 – זהה לאלגוריתם 3 אך בשלב הסופי הכנסנו עוד אלמנט של אקראיות לחנויות (עד כה היה אלמנט אקראיות ללקוחות בלבד). בסיום מימוש "הצרחה והחלפה 3" הוספנו שורות קוד אשר לוקחות כ-12% מהחנויות (היפר פרמטר) ומחליפות אותן באופן אקראי בחנויות אחרות.

שלב הבחירה. שלב זה בוצע במספר שלבים. תחילה מיינו את כל הכרומוזומים (ילדים) בדור הקודם עפ"י ציון ה fitness. 14 הילדים עם הציון הטוב ביותר (הנמוך ביותר) הועברו באופן אוטומטי לדור הבא וזאת בכדי לשמר את הטובים ביותר בכל דור ודור. לאחר מכן נבחרו 20 ההורים עם הציון הנמוך ביותר ולהם הוספנו באופן אקראי עוד 4 הורים מכלל הכרומוזומים בדור. 24 ההורים שנבחרו הם בנק ההורים לדור הבא. מהורים אלו יצרנו את הילדים. בכל פעם נבחרו 2 הורים באופן אקראי (ייתכן ונבחרו אותם הורים פעמיים) להזדווגות ליצירת 2 ילדים. ילד אחד נוצר ע"י "הצרחה והחלפה 2" והילד השני ע"י "הצרחה והחלפה 4". הילדים שנוצרו התווספו ל-14 הכרומוזומים הטובים ביותר מהדור הקודם וביחד איתם יצרו את הדור הבא.

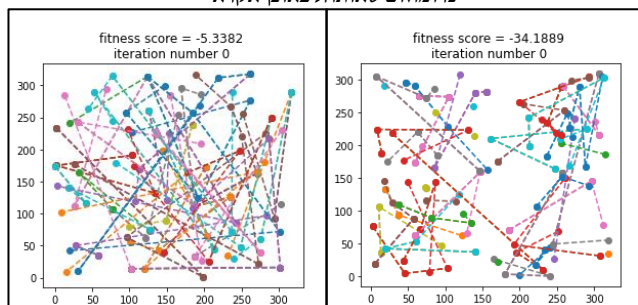
4. תוצאות

תחילה נציין כי מדד ההצלחה שלנו היה עפ"י הכרומוזום בעל הציון הנמוך ביותר בכל דור. קרי על השאלה האם השתפרנו תוך 8 איטרציות ענינו ע"י שאילת השאלה: האם הציון של הכרומוזום הטוב ביותר בדור $x+n$ נמוך מהציון הנמוך ביותר בדור x . תשובה חיובית משמעותה כי בוצע שיפור לאורך 8 איטרציות.

אתחול – ניכר כי אתחול "חכם" כפי שביצענו אשר מחלק את המטריצה לרבעים ומנסה לשמר חנויות ולקוחות באותו רבע תחת אותם מסלולים מקבל תוצאות טובות יותר משמעותית מאתחול אקראי:

בתרשים 4 ניתן לראות אתחול שבוצע בשיטה A (אתחול ללא מגבלות) בצד השמאלי ואתחול בשיטה C בצד הימני. ניתן להתרשם כי הציון שהתקבל משיטה C משמעותית טוב יותר (נמוך יותר)

תרשים 4 - מימין כרומוזום שאותחל ע"י התחשבות במרחקים, משמאל כרומוזום שאותחל באופן אקראי



משיטה A וכמו כן בשיטה C ניכרת לעין החלוקה לרבעים. בחירת היפר פרמטרים בפונקציית ה fitness הינה קריטית למימוש נכון והצלחת המודל. פונק' ה fitness אותה אנו רוצים למזער, כפי שהוצגה בפרק 2 מורכב מ-2 משתנים:

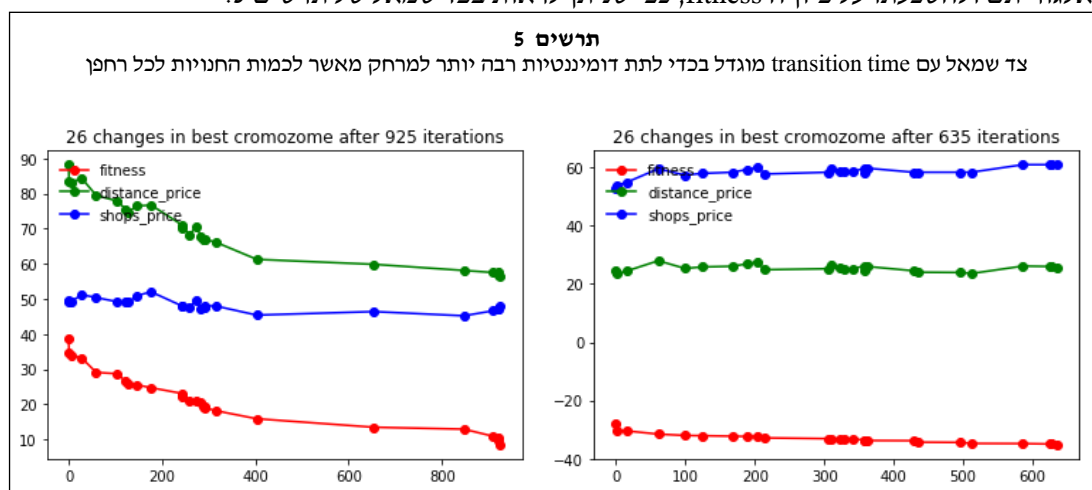
\min

$$f = \text{total distance} * \text{transition time} - \alpha * \text{number of shops price}$$

where

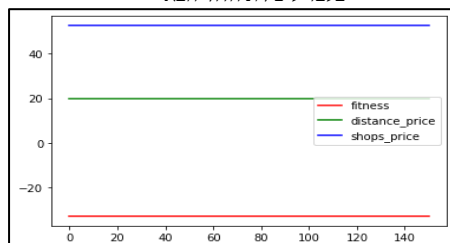
$$\text{number of shop price} = \sum_{i=1}^3 \text{delivery time}_i * \text{costumer}_i$$

בכדי להגיע לתוצאות טובות נדרש כיול מדויק של ההיפר פרמטרים α ו transition time . כיול הפרמטרים היה קריטי למימוש נכון ומציאת כרומוזום מוצלח. תרשים 5 מתאר את ציון ה fitness בקו אדום לאורך כמות איטרציות (ציר x) מוגדרות. בנוסף ניתן לראות בתרשים את הקו הירוק אשר מתאר את משתנה המרחק (אותו נדרש למזער) ואת הקו הכחול אשר מתאר את המשתנה של כמות החנויות (אותו נדרש למקסם) לכל רחפן. בתחילה קבענו היפר פרמטרים אשר גרמו לשיפור במהלך האיטרציות אשר נבע משינויים קלים במשתני המרחק וכמות החנויות אבל ניתן לראות בצד ימין של תרשים 5 כי הירידה המרכזית של הגרף האדום, ציון ה fitness, נבעה בעיקר מהגדלת משתנה החנויות (קו כחול) – כלומר יותר רחפנים ביצעו מספר משלוחים ועל כן השתפר ציון ה fitness. רצינו כי המדד המשמעותי יותר יהיה צמצום המרחק ועל כן הגדלנו את ה transition time (היפר פרמטר של המרחק) ובכך ניתן ביטוי גדול יותר למרחק באלגוריתם ולהשפעתו על ציון ה fitness, כפי שניתן לראות בצד שמאל של תרשים 5.

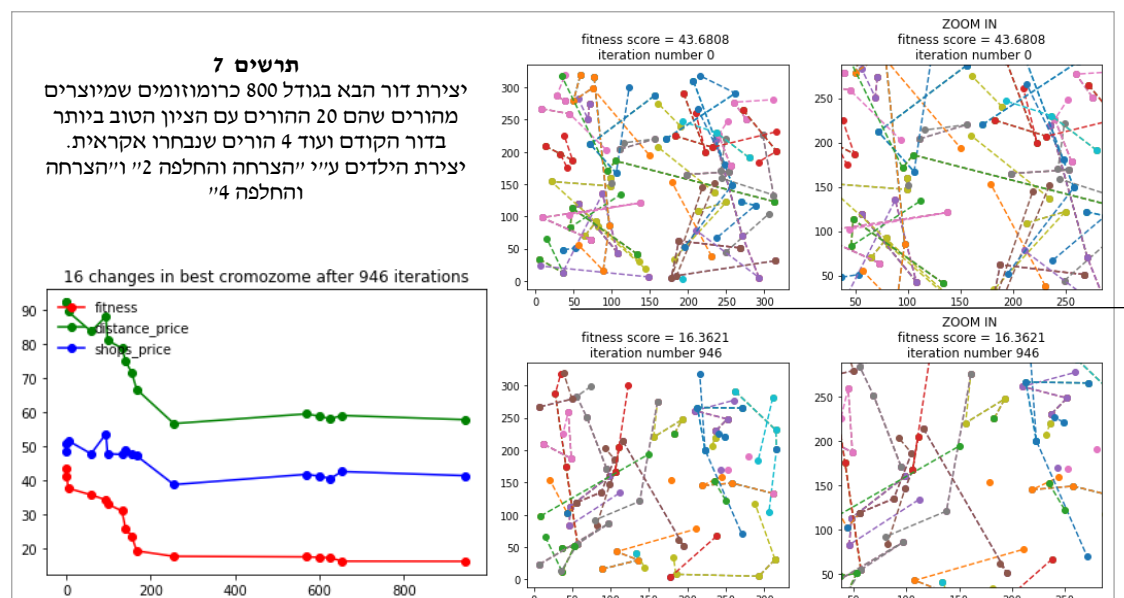


בהמשך בחנו באיזה פונקציית הצרחה והזזה נכון להשתמש ליצירת הילדים. תחילה נבחנו אפשרות להשתמש אך ורק בפונקציה אחת ליצירת הילדים. כאשר השתמשנו בפונקציות "הצרחה והזזה 1/3/4" קיבלנו כי אין שיפור לאורך מספר רב מידי של איטרציות – כפי שניתן לראות בתרשים 6 בו בוצע שימוש ב"הצרחה והזזה 4" בלבד. כאשר השתמשנו אך ורק ב"הצרחה והזזה 2" היה שיפור לאורך האיטרציות אך הוא היה לא משמעותי. הסקנו כי הבדל זה נובע ככל הנראה מהעובדה ש"הצרחה והזזה 2" משלב רנדומליות רבה משאר שיטות ההצרחה והזזה. התוצאות הטובות ביותר התקבלו ביצירת הדור הבא ע"י שילוב של 2 שיטות: "הצרחה והזזה 2" ו- "הצרחה והזזה 4". ההורים אשר יצרו את הדור הבא וגודל הדור הבא היו משתנה אשר השפיע רבות על אופן התכנסות המודל. באשר להורים בחנו אפשרויות רבות אשר משלבות הורים עם הציון הטוב ביותר בתוספת עם הורים עם ציון אקראי. גילינו כי בכדי להגיע לשיפור בזמן חישוב סביר נדרש לתת דגש להורים עם ציונים גבוהים – 20 ההורים עם הציונים הגבוהים ביותר ולהכניס אלמנט של אקראיות של כ-10%. הגדלת בנק ההורים והוספה של הורים עם ציונים נמוכים יותר (מעבר לשתואר) גרמה לביצועים פחותים במודל. גודל הדור הבא - התמקדנו במספר אפשרויות – 400, 800 ו-1600. דור בגודל 800 הביא לתוצאות טובות בשילוב זמן ריצה סביר. באופן כללי, בכל הבדיקות שעשינו, ככל שכמות האיטרציות גדלה כך קצה השינוי קטן. ניתן להבחין בכך בתרשים 5 תמונה שמאלית, בתמונה זו רואים כי מעל 200 איטרציות קצב השינוי קטן אל מול כמות האיטרציות - שינוי בכרומוזום עם הציון הגבוה ביותר בכל דור מתבצע כאחת ל 100 דורות (100 איטרציות). תופעה מעניינת נוספת היא האופן בו הכנסת הרנדומליות והרעש משפיעה. ניתן לראות בתרשים 5 בתמונה הימנית כי קיימות "תקופות שקט" בהם אין שינוי לאורך איטרציות, אבל כאשר מבוצע שינוי – כלומר נכנס מספיק רעש למודל המאפשר שיפור הכרומוזום הטוב ביותר, השינוי גורר

תרשים 6 – שימוש בהצרחה והחלפה 4 בלבד ליצירת הדור הבא



אחריו רצף שינויים ושיפורים, אחריהם שוב פעם אני חווים תקופת שקט ללא שינוי במודל. המימוש שבחרנו אכן מזער את פונק' המטרה שלנו, פונקציית ה-fitness. כתלות במשחק בפרמטרים, בפונקציות, בהיפר פרמטרים ובמשתנים נוספים כך השתנתה יעילות האלגוריתם, מהירות ההרצה ולבסוף הציון הסופי. לסיכום פרק זה נציג בתרשים 7 מימוש שלם של הרצה עם הנתונים הבאים: גודל כל דור, למעט האוכלוסייה הראשונית, נבחר להיות בגודל 800, ההורים הורכבו מ-20 הכרומוזומים עם הציון הגבוה ביותר ועוד 4 כרומוזומים שנבחרו באקראי ובנוסף לכך כמו בכל ההרצות הועברו באופן ישיר 16 הכרומוזומים הטובים ביותר לדור הבא. בהרצה זו נבחר ערך יחסית גבוה, 0.1, להיפר פרמטר שמכפיל את המרחק וזאת בכדי לתת משקל גבוה למרחק, הנ"ל מתבטא בגרף השמאלי בתרשים 7 ע"י הירידה המשמעותית במרחק וכמו כן ע"י הקורלציה הגבוהה בין המרחק לפונקציית ה-fitness. בנוסף בהרצה זו ניתן לראות אלמנטים נוספים שצינו: מאיטרציה מספר 255 ועד איטרציה 570 לא היה כל שינוי בכרומוזום הטוב ביותר אך אחת ונהיה שינוי (הוכנס רעש מספק לאלגוריתם שיצר שינוי), בוצעו 4 שינויים ברצף לאורך 73 איטרציות. עוד ניתן לראות כי עד איטרציה 255 הגורם המשפיע ביותר הינו המרחק, הירידה בגרף הירוק היא המשפיעה ביותר על הירידה בגרף האדום. מאיטרציה 255 ועד סיום ההרצה קיימת עליה איטית בכמות החנויות לכל רחפן, גרף כחול, והיא המשפיעה על הפתרון. הגרפים בצד ימין של תרשים 7 הינם "מבט על" על מרחב הבעיה הכולל תרשים גרפי של נקודות (חנויות ולקוחות) ומסלולי טיסה (קווים), כאשר כל צבע מבטא מסלול. בצד ימין של התרשים ניתן לראות בתמונה העליונה את גרף החנויות והמסלולים באיטרציה מספר 0. התמונה העליונה מורכבת מ-2 תמונות: התמונה השמאלית היא הגרף האמיתי בעוד התמונה הימנית היא תקריב. באיטרציה מספר 0 הציון של הכרומוזום הטוב ביותר הינו 43.68 ולאחר 946 איטרציות, הציון הטוב ביותר ירד ל 16.36, ב-2 התמונות התחתונות בתרשים 7, צד ימין. לסיכום פרק זה נזכיר כי מטרתנו הייתה להקטין את אורך וכמות הקווים. בהתרשמות ע"י הסתכלות בגרפים אשר מתארים את מסלולי הטיסה מ"מבט על" ניתן לראות כי מטרה זו הושגה – כמות המסלולים, אורך המסלולים וצפיפותם לא רק קטנים יותר מתמטית, כפי שמתואר בערכים המספריים של ציוני ה-fitness ובגרפים המתארים את השיפור לאורך האיטרציות, אלא ניתן לראות זאת ע"י בחינת תמונת הנתיבים באיטרציה 0 לעומת תמונת הנתיבים באיטרציה 946.



5. דיון

ניתן היה לממש את פתרון בעיית האופטימיזציה הנתונה ע"י מספר אלגוריתמים. בתחילת הדרך בחנו בנוסף לאלגוריתם הגנטי שימוש באלגוריתם K-nearest neighbours³ אשר מימושו יחסית פשוט וניתן בקלות לבצע את ההתאמות הנדרשות לבעיה שהגדרנו. בנוסף בחנו הסתכלות על הבעיה כבעיית שידוך מקסימלי מוכללת (many to many) ושימוש ברשתות זרימה למציאת זרימת מקסימום⁴, שיטה מורכבת יותר ופחות טבעית בראייתנו לפתרון הבעיה. לבסוף החלטנו לממש את האלגוריתם הגנטי מתוך העמקה באלגוריתם זה והבנה כי מימושו מתאים לבעיה שהגדרנו ויביא לכדי תוצאות טובות.

ביצענו התאמות רבות בעבודה זו בכדי ליישם את האלגוריתם הגנטי ולהתאים את הכתוב בספרות לבעיה שלנו. תחילה נדרש היה לאפיין את הכרומוזום עצמו. בעוד בספרות, במרבית המקרים, הכרומוזום באלגוריתם גנטי מתואר בייצוג בינארי, ניסינו בעבודה זו לממש ייצוג מובן ונוח לשימוש אשר מתאים לבעיה שלנו. בשלב האחרון הראנו כי בעוד נדרשת אקראיות ונדרשים מגוון פתרונות, אתחול "חכם" מקרב אותנו מראש לפתרון אופטימלי ומקצר משמעותית את זמני הריצה של האלגוריתם.

אלמנט הרנדומליות הינו האלמנט המרכזי בעבודה זו. למדנו ע"י ניסוי ותהייה רבים, בכלל השלבים השונים של האלגוריתם הגנטי, כי אנו מחויבים לשלב רנדומליות בכדי לאפשר לאלגוריתם להשתפר בזמני חישוב

³ Peterson, Leif E. "K-nearest neighbor." *Scholarpedia* 4.2 (2009): 1883.

סבירים. הכנסת רנדומליות, כמוה כהכנסת רעש למשתנים ולאוכלוסייה, חיונית בכדי לאפשר השתנות והשתפרות מדור לדור ובקשר ישיר חיונית להצלחת האלגוריתם הגנטי. שילבנו רנדומליות באתחול האוכלוסייה הראשונית, שילבנו רנדומליות בשני המימושים המרכזיים של שיטות ההצרכה והחלפה השונות ולבסוף שילבנו רנדומליות באופן יצירת הדור הבא – הן בבחירת הצאצאים מהדור הקודם לדור הבא והן בשימוש בשתי שיטות שונות של הצרכה והחלפה ליצירת הדור הבא. כאשר לא שילבנו אלמנטים מספקים של רנדומליות האלגוריתם נכשל. נוכחנו לגלות זאת כאשר השתמשנו בשיטות הצרכה והחלפה 1 ו-3 שאינן כוללות אלמנטים של רנדומליות ויתרה מכך הפתענו לגלות כי אפילו שימוש בשיטת "הצרכה והחלפה 4" (השיטה המוצלחת ביותר) ללא שילוב של שיטת הצרכה והחלפה נוספת מביא לאלגוריתם לא יעיל שאינו משתפר לאורך זמן איטרציה סביר. כיוול הפרמטרים למדידת פונקציית הfitness היה מהותי בכדי לאפשר יחס נכון בין המרחק לכמות החנויות לאורך נתיב אחד. ביציאה לדרך, בשלב הטיוטה, האלמנט היחיד אותו רצינו למזער היה אורך הנתיב של כל רחפן. בשלב המימוש גילינו כי זה לא מספיק ונדרש להוסיף משתנה שאנחנו התייחסנו אליו כאילו : number of shops. משתנה זה מאלץ את האלגוריתם להשתמש בכל רחפן למספר מירבי של לקוחות ובכך מבטא גם חסכון לרשת בהפעלת רחפנים וגם במימוש הכנסת פרמטר זה והיפר פרמטרים למשתנה זה ולמשתנה המרחק אפשרה לנו לשלוט בקצב ההתכנסות של המודל לפתרון.

6. סיכום

עבודה זו נכתבה ומומשה על בסיס השראה שקיבלנו מבעיה הלקוחה מהעולם האמיתי. הפתרון אשר קיבלנו לבסוף אינו בהכרח הפתרון המיטבי אך זהו, עבורנו כמגדירי הבעיה, פתרון מספק באיזון שבין הפתרון עצמו (התועלת הכלכלית שהוא מביא בבעיה שלנו) אל מול העלות והזמן הנדרש למציאת הפתרון. המשפט האחרון מגלם את יתרונותיו וחסרונותיו המרכזיים של אלגוריתם גנטי. בעוד כי ייתכן וקיימות לבעיית האופטימיזציה אותה אנו רוצים לפתור פתרון יחיד (או מספר פתרונות) שהוא המיטבי, קיימים מצבים בהם המורכבות למציאת הפתרון המיטבי גבוהה או לחלופין לא אפשרית, אלגוריתם גנטי מאפשר למשתמש בו שיטה לחשב פתרון "מספיק טוב" בחישוב פשוט ופרק זמן סביר. תחילת הדרך אשר התחילה בהנדסה ראשונית של המימוש הנכון של הבעיה אינה דומה למימוש הסופי. כמו האלגוריתם אשר עובר אבולוציה מדור לדור כך גם עבודתנו עברה אבולוציה בכדי להגיע לתוצאות הנדרשות. בעבודה זו נדרשנו לאפיין את מרחב הבעיה, לאפיין פונ' fitness אשר תביא פתרון נכון לבעיה, לכייל היפר פרמטרים ואחרי כל אלו להביא מספר אלמנטים רחב של מימוש רנדומליות לאורך כל האלגוריתם. הרנדומליות וכיוול נכון של היפר פרמטרים הם המרכיבים המרכזיים אשר אפשרו מימוש נכון של הפתרון לבעיה ואפשרו לאוכלוסייה להשתנות ולהשתפר בכל דור. העבודה נבנתה בשלבים. כל שלב הונדס ונבדק בנפרד ובהמשך, כאשר כל השלבים היו מוכנים, בוצעו הטיובים הנדרשים בכדי לממש את האלגוריתם השלם.

7. ביבליוגרפיה

Peterson, Leif E. "K-nearest neighbor." *Scholarpedia* 4.2 (2009): 1883.

Coit, David W., and Alice E. Smith. "Penalty guided genetic search for reliability design optimization." *Computers & industrial engineering* 30.4 (1996): 895-904.

Kumar, Manoj, et al. "Genetic algorithm: Review and application." *Available at SSRN* 3529843 (2010).