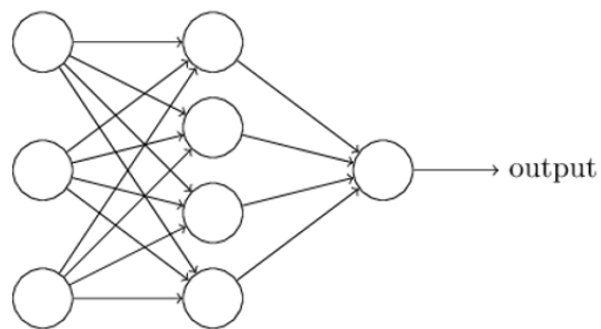


Multi-Object Optimization Course

## Artificial Neural Network Optimization



### Authors:

Stav Jacob

Yarden Turgeman

Alon Mizrahi

Alon Laron

## תוכן עניינים

1	מבוא.....	3
2	הגדרת הבעיה.....	3
3	שיטת הפתרון.....	5
4	האלגוריתמים שנבחרו.....	6
4.1	הסבר לבחירת האלגוריתמים.....	6
4.2	NSGA II.....	6
4.3	MOEA/D.....	8
5	הדגמת הרצה עבור דור בודד.....	9
6	הצגת התוצאות וניתוח סטטיסטי.....	10
6.1	הצגת התוצאות.....	10
6.2	ניתוח סטטיסטי.....	13
6.3	הצגת פתרונות אופניים.....	15
6.4	בעיה תלת ממדית (בונוס).....	16
7	סיכום ומסקנות.....	17
8	מקורות.....	17

## 1 מבוא

רשת עצבית מלאכותית (ANN – Artificial Neural Network), רשת נוירונים או רשת קשרית הוא מודל מתמטי חישובי, שפותח בהשראת תהליכים מוחיים או קוגניטיביים המתרחשים ברשת עצבית טבעית ומשמש במסגרת למידת מכונה. רשת מסוג זה מכילה בדרך כלל מספר רב של יחידות מידע (קלט ופלט) המקושרות זו לזו, קשרים שלעיתים קרובות עוברים דרך יחידות מידע "חבויות" (Hidden Layer). צורת הקישור בין היחידות, המכילה מידע על חוזק הקשר, מדמה את אופן חיבור הנוירונים במוח. השימוש ברשתות עצביות מלאכותיות נפוץ בעיקר במדעים קוגניטיביים, ובמערכות תוכנה שונות - בהן: מערכות רבות של אינטליגנציה מלאכותית המבצעות משימות מגוונות - זיהוי תווים, זיהוי פנים, זיהוי כתב יד, חיזוי שוק ההון, מערכת זיהוי דיבור, זיהוי תמונה, ניתוח טקסט ועוד.

## 2 הגדרת הבעיה

משימת הקצה אותה אנו מבצעים על ידי שימוש ברשת עצבית (רשת נוירונים כפי שהוסברה קודם לכן, ועוד יורחב עליה בהמשך) היא משימת קלסיפיקציה - כלומר משימת סיווג. באופן ספציפי משימת הסיווג שתבוצע במסגרת פרויקט זה כוללת סיווג בין 5 מצבים שונים של כף יד אנושית (1-5 class) זאת כפי שמתואר באיור הבא.



איור 1: 5 מצבי היד השונים

חיזוי זה נעשה מבלי להתבסס על מידע וויזואלי, חשמלי, עצבי או כל סוג מידע אחר הנוגע לכף היד עצמה אלא מבוסס על סט רחב של חיישני כוחות הממוקמים באמה. מערכת החיישנים אשר נעשה בה שימוש הינה מערכת מקורית, אשר תוכננה ונבנתה לשם ביצוע מחקר מקיף בנושא קלסיפיקציה של מצבי כף היד (רוב המערכת מודפסת במדספת תלת מימד).



איור 2: מערכת החיישנים עצמה והמערכת בתצורתה "המולבשת" על הנסיין.

כעת נגדיר באופן מפורט את פרמטרי הבעיה:

בעבור הרשת העצבית האותות המתקבלים ממערכת החישה שהוצגה קודם לכן יוגדרו מעתה כאותות INPUT. חמשת מצבי כף היד יוגדרו כ- OUTPUT של הרשת העצבית. את הרשת נעביר שלבי אימון על מנת שתוכל "ללמוד" את אופן החיזוי הנכון כאשר שלבים אלו יבוצעו בעזרת סט מדידות אחד. בנוסף לכך, הרשת תעבור מספר הרצות של בחינת אופן הדיוק שלה, כלומר בכמה אחוזים הרשת דייקה בחיזוי שלה, זאת יבוצע על ידי סט נתונים אחר.

פרמטרי התכנן - שלושת פרמטרי התכנן עליהם נשלט בעת ביצוע האופטימיזציה הם מספר השכבות החבויות ברשת, גודלן של השכבות וכן גודל השכבה הראשונה.

מרחב המטרות - מרחב המטרות שלנו מורכב משתי מטרות שונות. המטרה הראשונה היא דיוק מרבי של הרשת. לפיכך, זוהי בעיית מקסימום שכן אנו מעוניינים לקבל דיוק גבוהה ככל האפשר (בשאיפה אופטימלית

$$\text{ל- } 100\% \text{ דיוק) של חיזוי מצבי היד השונים. } accuracy = \frac{\sum \blacksquare \text{correct classifications}}{\sum \blacksquare \text{total classifications}} \times 100\%$$

חשוב לציין כי דיוק הרשת נבדק על סט בדיקה שלא בוצע עליו אימון.

המטרה השנייה היא גודל הרשת, כאשר נבצע הערכה של גודל הרשת בעזרת מדידת מספר הפרמטרים המצויים ברשת שלנו. גודל הרשת משפיע על מגוון רחב של פרמטרים כדוגמת זמן החישוב, מאמץ מיחשובי ועוד. גודל הרשת, שכאמור ישוער על ידי מספר הפרמטרים, הוא פונקציה של שכבות הנוירונים וכן של גודל כל אחת מהשכבות:

$$size = func(number\ layers, layer\ size)$$

לפיכך, זוהי בעיית מינימום שכן אנו שואפים לבצע חישוב מדויק ככל האפשר, אך בלי לבזבז כח מחשוב גדול מדי ולאורך זמן ארוך מדי. כך שלבסוף מתקבלת עבורנו בעיית מקסימום - מינימום.

### 3 שיטת הפתרון

בחלק זה נתאר את הפונקציה שלנו ואופן ביצוע החישובים. היות ובמקרה זה לא מדובר בפונקציה אלגברית שניתנת לתיאור מתמטי מידי, אלא ברשת נוירונים המבצעת אימון וחיזוי (לטובת ביצוע קלסיפיקציה, כפי שתואר קודם) של מצבי כף יד, נציג קוד Python אשר מבצע הלכה למעשה את המשימה. כלל הקוד יצורף כנספח בסוף הדוח.

בראשית הקוד נייבא את כלל החבילות הדרושות ל-Python לשם ביצוע המטלות הדרושות. השלב הבא הינו ייבוא של המידע. מידע זה נאסף קודם לכן על ידי מערכת החיישנים אשר מונחת על האמה ותוארה גם כן בפרק קודם. לאחר מכן מתבצע עיבוד וקידוד של המידע למשתנים המתאימים ביותר על מנת לאפשר עבודה נוחה בשלבים הבאים. לאחר מכן נגדיר את הרשת העצבית עצמה, אשר משמשת כפונקציית המטרה, כאשר בתוך פונקציה זו מתבצע אימון ובחינה של הרשת. הפונקציה מקבלת את משתני התכן ומחזירה את המטרות:

```
def obj_func(n_first_layer, n_hidden, s_hidden):
    set_seed(5)
    # define model
    model = Sequential()
    model.add(Dense(n_first_layer, input_dim=28, activation='relu', input_shape=(28,)))
    for i in range(n_hidden):
        model.add(Dense(s_hidden, activation='relu'))
    model.add(Dense(5, activation='softmax'))
    # model.summary()
    trainableParams = np.sum([np.prod(v.get_shape()) for v in model.trainable_weights])
    nonTrainableParams = np.sum([np.prod(v.get_shape()) for v in model.non_trainable_weights])
    totalParams = trainableParams + nonTrainableParams
    # compile the model
    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
    # fit the model
    model.fit(X_train, y_train, epochs=50, batch_size=20, verbose=0)
    # evaluate the model
    loss, acc = model.evaluate(X_test, y_test, verbose=0)
    return(totalParams, round(acc*100, 2))
```

בקטע המוצג לעיל ניתן לראות את אופן הגדרת הרשת (בהתאם למספר השכבות, גודל השכבות, וגודלה של השכבה הראשונה), ניתן לראות שפונקציית האקטיבציה הנבחרת עבור מרבית השכבות הינה "relu" ואילו עבור השכבה האחרונה מוגדרת פונקציית האקטיבציה "softmax" אשר דרושה בשכבה האחרונה היות וישנו צורך לספק קלסיפיקציה בין 5 מצבים סופיים.

לאחר מכן ניתן לראות את הגדרת הפרמטרים המאומנים ואלו שאינם מאומנים, כאשר לאחר שלב זה מתבצע שלב האימון של הרשת בהתאם למידע המסופק, ניתן לראות שמספר ה epochs (כלומר מחזורי האימון) הוא קבוע בשלב זה על ערך של 50. בנוסף הפרמטר batch\_size המתאר את כמות דוגמאות האימון שבהן יעשה שימוש בכל איטרציה גם הוא מקובע על ערך של 20. לבסוף ניתן לראות שמבוצע חישוב של הדיוק המתקבל בסוף שלב האימון וזהו גם הפלט של פונקציה זו.

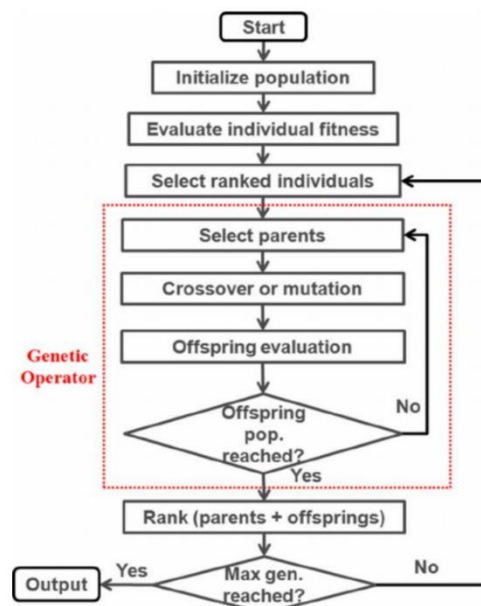
## 4 האלגוריתמים שנבחרו

### 4.1 הסבר לבחירת האלגוריתמים

NSGA II אלגוריתם זה נבחר בשל הפופולריות שלו ויעילותו הידועה בפתרון בעיות אופטימיזציה בעלות שתי מטרות. בנוסף, על פי מאמר [1] אלגוריתם זה יודע להתמודד עם בעיות בדיקה מורכבות, מסוגל למצוא פיזור טוב יותר של פתרונות והתכנסות טובה יותר לחזית האמיתית של פארטו-אופטימלית. MOEAD אלגוריתם אשר נפוץ גם כן בפתרון בעיות עם יותר ממשתנה מטרות אחד. בנוסף, על פי מאמר [2], נמצא כי אלגוריתם זה בעל ביצועים טובים יותר או דומים ל-NSGA II וברצוננו לאמת זאת עבור הבעיה שלנו. נציין כי האלגוריתם שנבחר לנו על ידי המרצה לא קיים במאגר pymo, לכן זה היה שיקול נוסף בבחירת האלגוריתם.

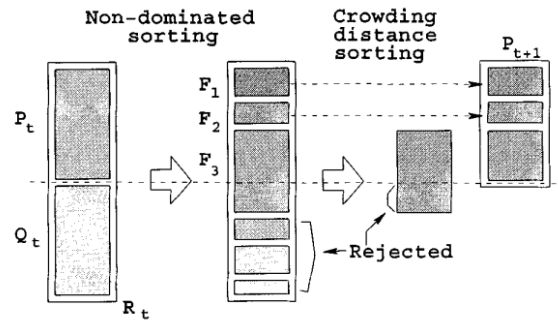
### 4.2 NSGA II

האלגוריתם NSGA II הינו אחד האלגוריתמים הנפוצים ביותר ממשפחת אלגוריתמים אבולוציוניים רב-אובייקטיביים. אלגוריתם זה מציג גישה של מיון מהיר עבור הפתרונות הבלתי נשלטים. כמו כן, קיים אופרטור גנטי היוצר מאגר זיווג על ידי שילוב אוכלוסיות ההורים והצאצאים ובחירת הפתרונות הטובים ביותר.



איור 3: תרשים זרימה של אלגוריתם NSGA II

כפי שמופיע באיור 1, תחילה יש לבצע אתחול של אוכלוסיית ההורים והצאצאים. אוכלוסייה זו נבחרת בצורה אקראית ועבור הקבוצה המאוחדת מחשבים את משתני האופטימיזציה בכדי לבצע מיון של הבלתי נשלטים. בשלב הבא מבצעים דירוג לפי שיטת המחסנית (Tournament Selection-TS), כלומר, ראשית נכנס סט הפתרונות עם הדירוג הגבוה ביותר  $1F$ . במידה וקבוצה זו קטנה מ- $N$  (גודל האוכלוסייה) נכניס את הסט הבא בדירוג  $2F$ , כפי שמופיע באיור 2, בדרך זו האליטיזם מובטח.



#### איור 4: תהליך ה-NSGA II

לאחר מכן, האוכלוסייה עוברת תהליך אבולוציוני של Crossover&Mutation ליצירת אוכלוסייה חדשה בגודל  $N$ . הפתרונות שיבחרו עבור האוכלוסייה החדשה יקבעו לפי שיטת Crowding Distance (CD). בשיטה זו הוא לפלג באופן אחיד כמה שיותר פתרונות ולתת עדיפות לפתרונות רחוקים יותר שנותנים יותר מידע. ככל שערך מדד זה גדול יותר כך הוא טוב יותר. פתרון  $i$  בשיטת CD יוביל על פתרון  $j$  בשיטת TS אם אחד מהתנאים הבאים מתקיים:

1. אם לפתרון  $i$  יש דירוג גבוה יותר.
  2. אם לשני הפתרונות יש את אותו הדירוג, אך לפתרון  $i$  יש CD גבוה יותר מאשר לפתרון  $j$ .
- במידה והתהליך אינו נפגש בתנאי עצירה, תהליך זה חוזר על עצמו עד שיתקבלו פתרונות מספקים שיפיקו את החזית האופטימלית.

### MOEA/D 4.3

האלגוריתם MOEA/D הינו אלגוריתם אבולוציוני רב-אובייקטי המבוסס על דקומפוזיציה (פירוק). האלגוריתם הוצע לראשונה על ידי Zhang and Li בשנת 2007 ומאז צבר תאוצה ופותחו תתי אלגוריתמים נוספים על בסיסו. האלגוריתם מפרק בעיית אופטימיזציה רב-אובייקטית למספר תת-בעיות אופטימיזציה סקלריות ובו-זמנית מייצל אותן. כל תת-בעיית עוברת אופטימיזציה על ידי שימוש במידע ממספר בעיות המשנה השכנות, מה שגורם ל-MOEA/D להיות בעל מורכבות חישובית נמוכה יותר בכל דור מאשר MOGLS ו-NSGA II. יתרון נוסף הוא שניתן לשלוט על כמות תת הבעיות שהאלגוריתם פותר, כלומר ניתן לשלוט ברזולוציה של החזית או לחלופין להתמקד באזור מסוים בחזית.

The algorithm works as follows:

**Input:**

- MOP (1);
- a stopping criterion;
- $N$ : the number of the subproblems considered in MOEA/D;
- a uniform spread of  $N$  weight vectors:  $\lambda^1, \dots, \lambda^N$ ;
- $T$ : the number of the weight vectors in the neighborhood of each weight vector.

**Output:** EP.

**Step 1) Initialization:**

**Step 1.1)** Set  $EP = \emptyset$ .

**Step 1.2)** Compute the Euclidean distances between any two weight vectors and then work out the  $T$  closest weight vectors to each weight vector. For each  $i = 1, \dots, N$ , set  $B(i) = \{i_1, \dots, i_T\}$ , where  $\lambda^{i_1}, \dots, \lambda^{i_T}$  are the  $T$  closest weight vectors to  $\lambda^i$ .

**Step 1.3)** Generate an initial population  $x^1, \dots, x^N$  randomly or by a problem-specific method. Set  $FV^i = F(x^i)$ .

**Step 1.4)** Initialize  $z = (z_1, \dots, z_m)^T$  by a problem-specific method.

**Step 2) Update:**

For  $i = 1, \dots, N$ , do

**Step 2.1) Reproduction:** Randomly select two indexes  $k, l$  from  $B(i)$ , and then generate a new solution  $y$  from  $x^k$  and  $x^l$  by using genetic operators.

**Step 2.2) Improvement:** Apply a problem-specific repair/improvement heuristic on  $y$  to produce  $y'$ .

**Step 2.3) Update of  $z$ :** For each  $j = 1, \dots, m$ , if  $z_j < f_j(y')$ , then set  $z_j = f_j(y')$ .

**Step 2.4) Update of Neighboring Solutions:** For each index  $j \in B(i)$ , if  $g^{te}(y'|\lambda^j, z) \leq g^{te}(x^j|\lambda^j, z)$ , then set  $x^j = y'$  and  $FV^j = F(y')$ .

**Step 2.5) Update of EP:**

Remove from EP all the vectors dominated by  $F(y')$ .

Add  $F(y')$  to EP if no vectors in EP dominate  $F(y')$ .

**Step 3) Stopping Criteria:** If stopping criteria is satisfied, then stop and output EP. Otherwise, go to **Step 2**.

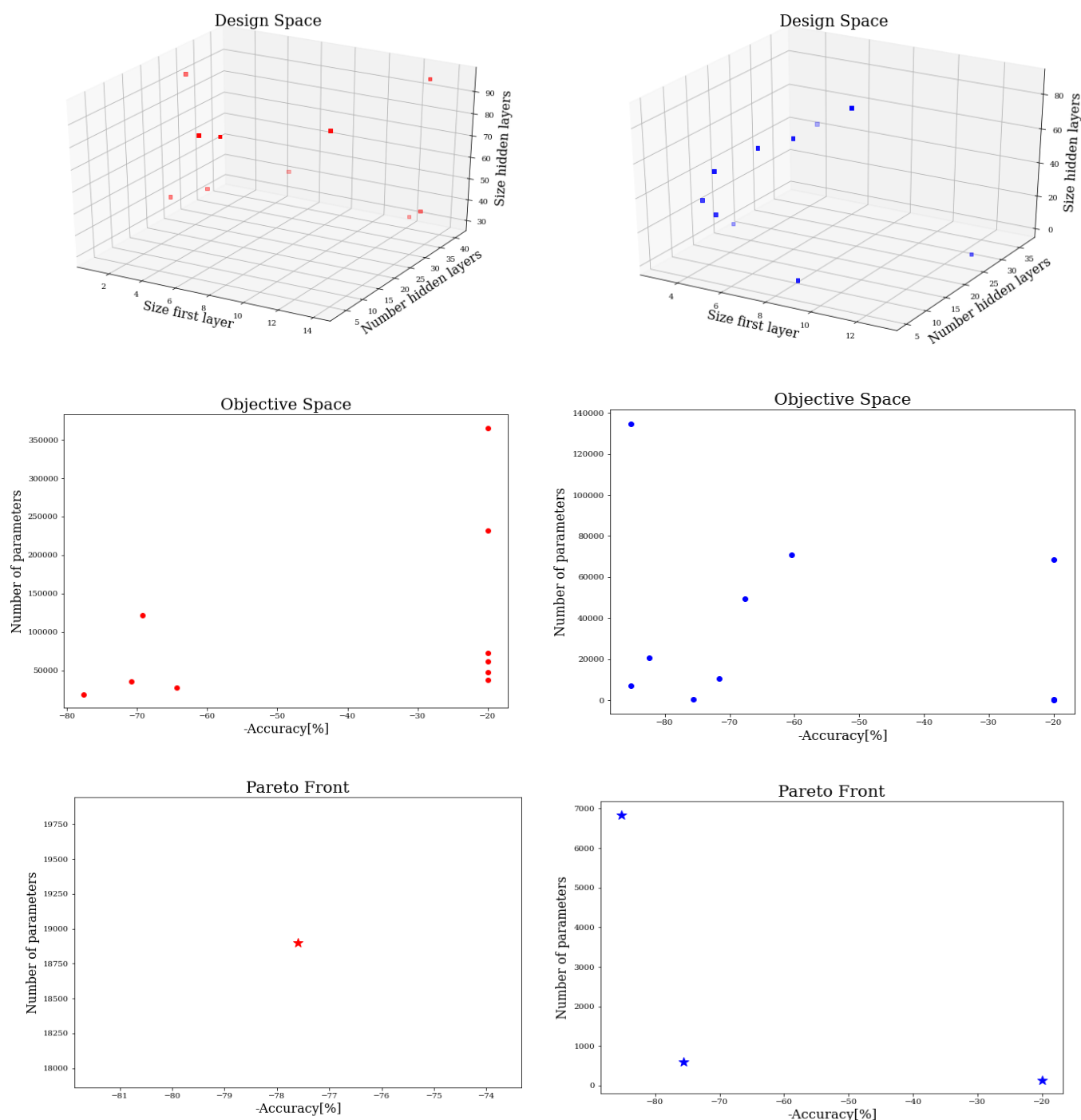
#### איור 5 : תיאור השלבים ב-MOEA/D

באלגוריתם זה תחילה נדרש לאתחל קבוצה  $EP = \emptyset$  – קבוצה חיצונית אליה יוכנסו אוכלוסיית הפתרונות הבלתי נשלטים בכל דור. בסיום התהליך נקבל פלט של קבוצה זו ממנה נדרש לבחור את הפתרון האידיאלי עבורנו. לאחר מכן, האלגוריתם מחשב את המרחק בין כל שני וקטורי משקל ואז מחשב את וקטורי המשקל ה- $T$  (כמות השכנים עבור תת קבוצה  $i$ ) הקרובים ביותר לכל וקטור משקל. בשלב הבא מיוצרת אוכלוסייה ראשונית בגודל  $N$   $\{x^1, \dots, x^N\}$  אשר מייצגת את תת בעיות האופטימיזציה הסקלריות. תהליך זה מבוצע באופן רנדומלי. לאחר מכן מתבצעת הערכה לפונקציות האופטימיזציה  $\{FV^1, \dots, FV^N\}$  כאשר  $FV^i = F(x^i)$ . בשלב הבא מבצעים אתחול ל- $z = \{z_1, \dots, z_m\}$  כאשר  $z_i$  מייצג את הערך הטוב ביותר (הנקודה האידיאלית) שנמצא עד כה עבור אובייקט  $f_i$ . בשלב הבא נכנס למשחק החלק האבולוציוני. עבור תת קבוצה  $i$  בוחרים באופן רנדומלי שני אינדקסים  $k$  ו- $l$  מהקבוצה  $B(i)$  ואז מייצרים פתרון חדש  $y$  מתתי הקבוצות  $x^k$  ו- $x^l$  על ידי שימוש באופרטור הגנטי. לאחר מכן משפרים את היוריסטיקה על  $y$  ליצירת  $y'$  ואז בודקים האם נקודת הרפרנס שלנו  $z_j$  קטנה מערך  $f_j(y')$ . אם כן, מאתחלים את ערך נקודת הרפרנס לערך החדש. העדכון מתבצע גם על פתרונות השכנים, והפתרונות הבלתי נשלטים נכנסים לקבוצת ה-EP. בשלב זה במידה וקיימים בקבוצת ה-EP פתרונות שנשלטים על ידי  $F(y')$  הם מוסרים מהקבוצה. התהליך חוזר חלילה עד שפוגש בקריטריון העצירה ומוציא פלט של קבוצת ה-EP.



## 5 הדגמת הרצה עבור דור בודד

להלן דוגמת הרצה עבור דור בודד המכיל 10 פתרונות עבור האלגוריתמים הנבחרים:



איור 6: דוגמת הרצה עבור דור בודד - NSGAII מימין בכחול, MOEAD משמאל באדום

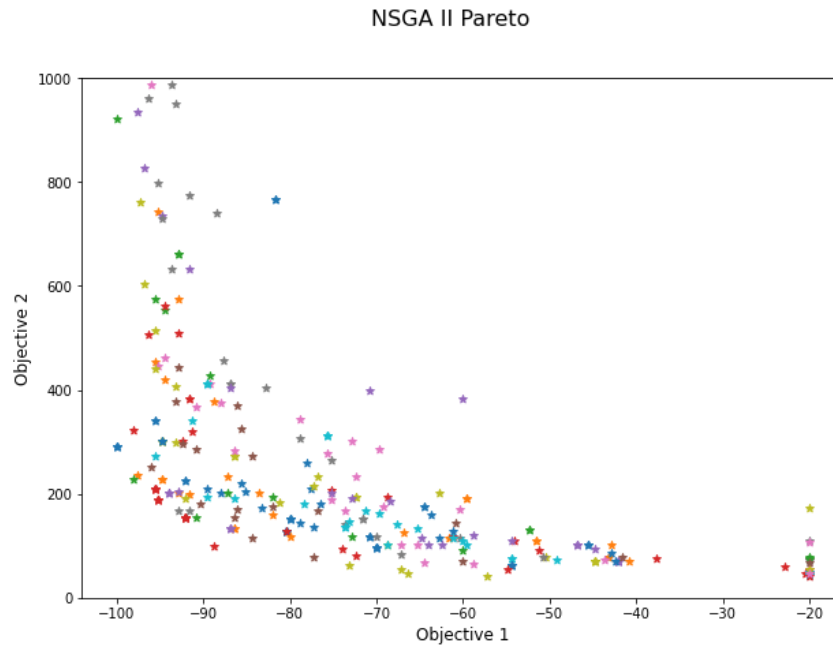
מההשוואה עבור דור בודד ניתן לראות כי אלגוריתם NSGA II הפיק מספר גדול יותר של פתרונות בלתי נשלטים לעומת אלגוריתם MOEAD. בנוסף לכך, הפתרונות בעלי פיזור טוב כך שקיימים שני פתרונות המקיימים מינימום בקצוות בכל אחד ממשתני האופטימיזציה ופתרון נוסף המקיים מינימום בשני המשתנים יחד ומציג תוצאה טובה מאוד מהצופה, בהתחשב בכך שמדובר בדור בודד.

## 6 הצגת התוצאות וניתוח סטטיסטי

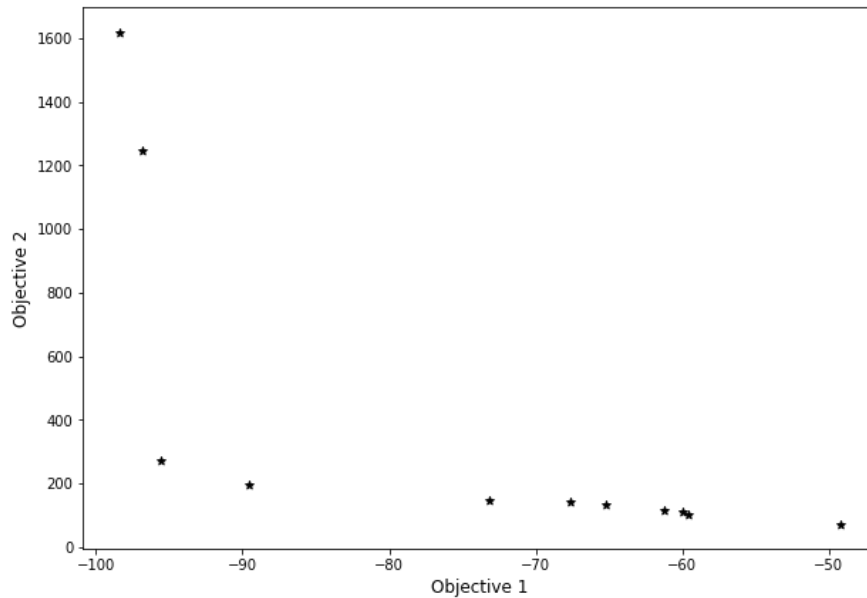
### 6.1 הצגת התוצאות

בחלק זה מוצגות תוצאות עבור 30 הרצות לכל אלגוריתם, כל הרצה היא בעלת 10 דורות עם אוכלוסייה התחלתית של 30 פרטים.

תוצאות עבור 30 הרצות באלגוריתם NSGA II :

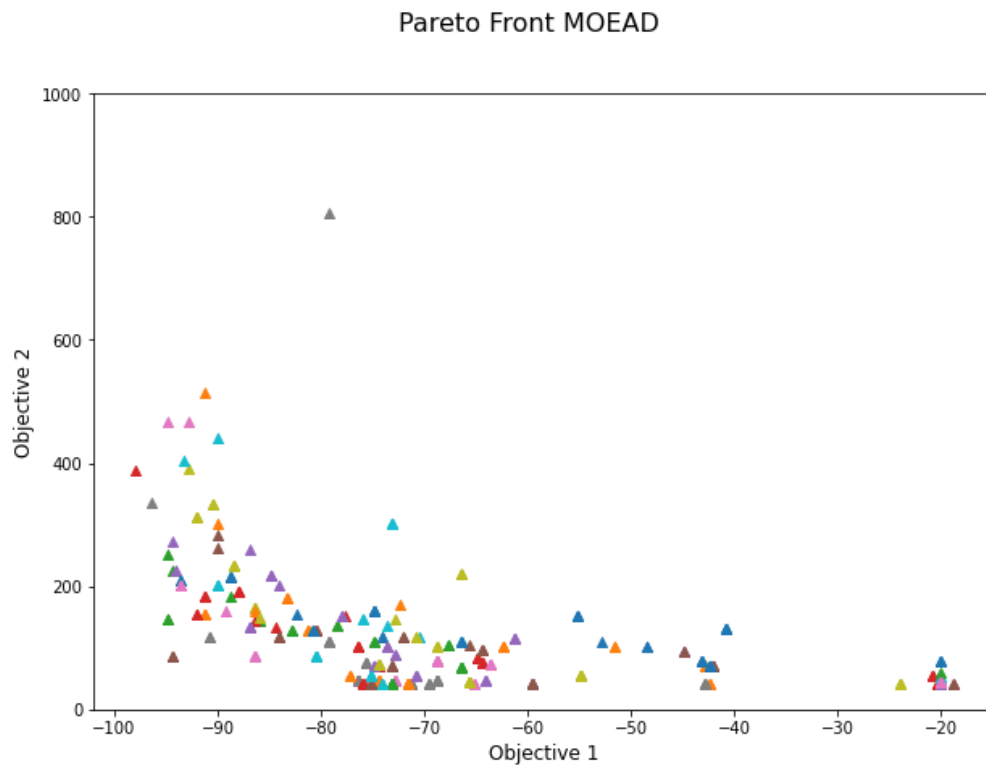


איור 7 : חזית האיחוד עבור 30 הרצות NSGA II  
NSGA II Pareto Front

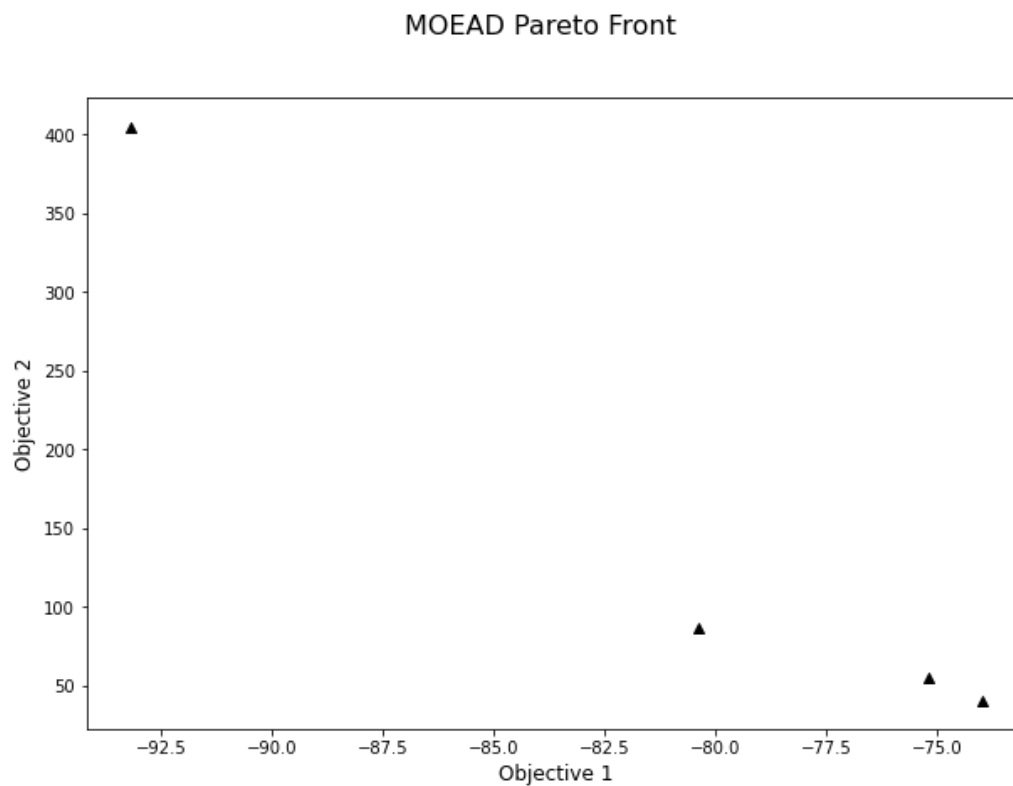


איור 8 : חזית פרטו עבור 30 הרצות NSGA II

תוצאות עבור 30 הרצות באלגוריתם MOEAD :



איור 9 : חזית האיחוד עבור 30 הרצות MOEAD

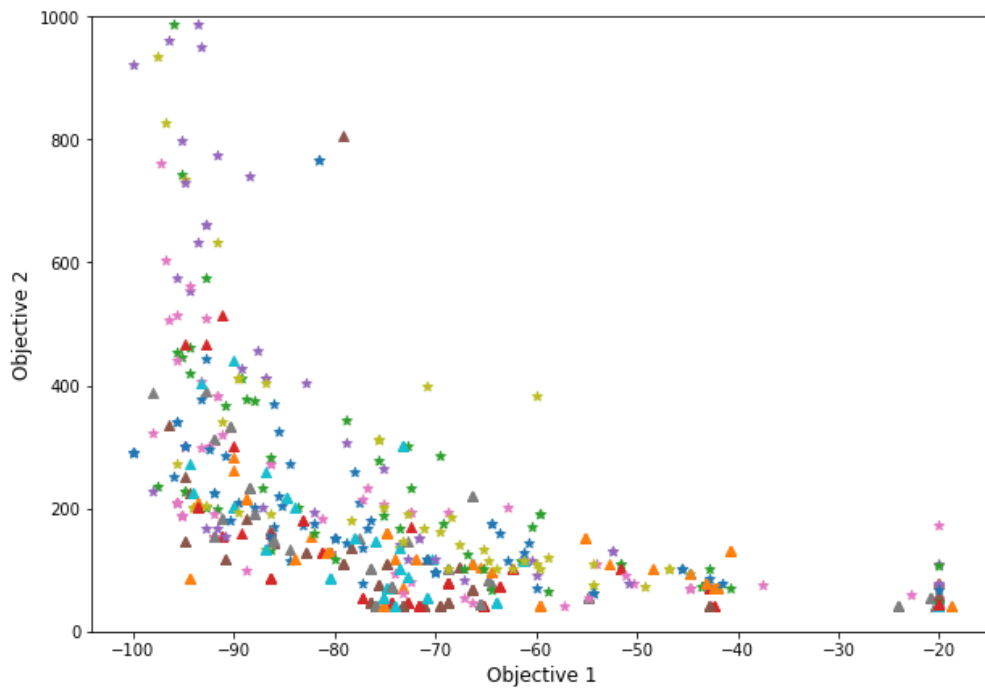


איור 10 : חזית פרטו עבור 30 הרצות MOEAD

ניתן לראות כי באיור 8 יש 11 פתרונות בחזית בעוד שבאיור 10 יש 4 פתרונות בחזית.

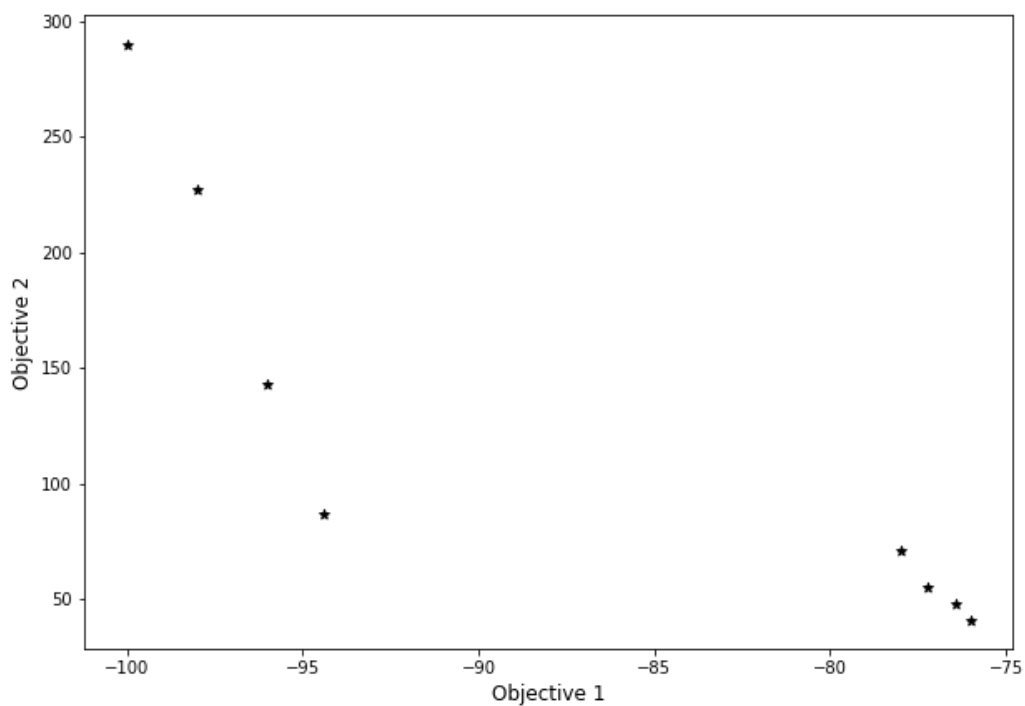
חזית פרטו המאוחדת עבור כל 60 ההרצות עם 2 האלגוריתמים :

All Pareto



איור 11 : חזית האיחוד עבור 62 הרצות כוכב-NSGA II, משולש - MOEAD

United Pareto Front



איור 12 : חזית פרטו עבור 60 הרצות

באיור 11 ניתן לראות את האיחוד החזיתות של 2 האלגוריתמים שנבדקו עבור 30 דורות בנוסף ל-2 הרצות ארוכות של 30 דורות, כאשר לכל אלגוריתם סימון שונה שניתן יהיה להבדיל ביניהם, ניתן לראות כי מתקבל פיזור דומה בין 2 האלגוריתמים שנבדקו.

איור 12 היא חזית הרפרנס איתה נעשה שימוש לחישוב המדדים בהמשך. כאשר בוחנים את החזיתות שהתקבלו באיור 8 - 10 ניתן להבחין כי מתקבלות יותר נקודות בחזית עבור האלגוריתם NSGA II לעומת MOEAD.

## 6.2 ניתוח סטטיסטי

עבור הניתוח הסטטיסטי נעשה שימוש במדדים לבחינת טיב התוצאות והחזיתות שהתקבלו, בכדי לבצע את הניתוח יש צורך בחזית רפרנס, במקרה שלנו חזית הרפרנס היא החזית שהתקבלה כתוצאה מאיחוד 2 הרצות ארוכות של 30 דורות לכל הרצה. המדדים שנעשה בהם שימוש :

### • IGD – Inverted Generational Distance

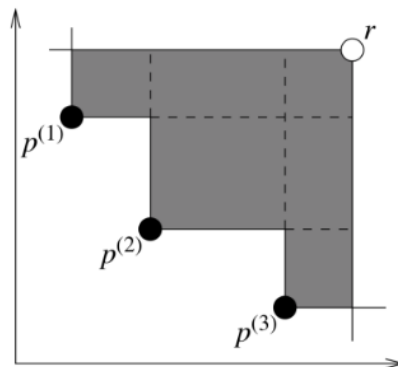
מדד זה מחשב מרחק אוקלידי בין כל נקודה  $Z$  לנקודה הקרובה ביותר  $A$ , כאשר  $Z$  היא נקודה על חזית הרפרנס  $A$  היא נקודה אותה אנו רוצים לבחון על החזית שהתקבלה. מדד זה הוא מדד מובנה בספריית Pymoo והנוסחה לחישוב מדד זה היא :

$$IGD(A) = \frac{1}{|Z|} \left( \sum_{i=1}^{|Z|} \hat{d}_i^p \right)^{1/p}$$

כלל שערכו של מדד זה קטן יותר כך יש התאמה טובה יותר לחזית פרטו ולכן נרצה ערכים מינימליים של מדד זה.

### • HV – HyperVolume

מדד זה מחשב את השטח בין נקודת ייחוס ( $r$ ) שאנו קובעים לפתרונות המתקבלים ( $p$ ), ראה איור 13, מדד זה הוא מדד מובנה בספריית Pymoo גם כן.



איור 13 : תיאור חישוב מדד HV

עבור מדד זה נרצה לקבל שטח כמה שיותר גדול, כלומר ככל שהשטח גדול יותר הפתרונות הנבדקים ישלטו על הפתרונות האחרים ולכן נרצה ערכים מקסימליים במדד זה.

### • Spread

מדד זה מחשב את הפיזור בין הפתרונות בחזית ונותן אינדיקציה על הפיזור שלהם. מדד זה אינו פונקציית מובנית בספריית Pymoo ולכן כתבנו פונקציה שמקבלת 2 נקודות ומחשבת את המרחק ביניהן ומבצעת מיצוע על המרחקים. במדד זה נרצה ערכים גדולים מכיוון שאלו מצביעים על פיזור בין הנקודות על החזית.

התוצאות שהתקבלו :

טבלה 1 - מדדים עבור 30 הרצות

NSGA II					MOEAD				
index	IGD	HV	SPREAD	TIME [min]	index	IGD	HV	SPREAD	TIME [min]
0	25.2	68894.4	89.7	41.8	0	399.4	0.0	673.0	49.5
1	27.9	68035.2	75.8	38.4	1	37.3	63632.4	53.5	30.6
2	1980.7	61281.6	3919.8	49.2	2	26.8	67196.8	65.8	29.7
3	23.3	67408.8	52.4	38.4	3	32.2	68598.0	69.2	31.0
4	33.1	65670.8	61.7	47.5	4	33.1	65280.4	61.1	33.8
5	313.3	59876.8	676.7	46.0	5	33.9	61592.4	55.4	32.5
6	472.2	61060.4	914.2	36.9	6	14.0	66410.0	55.9	21.6
7	1322.3	56896.4	2258.3	31.6	7	9.7	70281.6	57.9	21.9
8	167.1	58065.2	276.6	30.5	8	20.8	55474.4	46.7	26.7
9	842.0	57913.2	890.4	29.4	9	36.4	61692.4	61.5	20.3
10	1925.4	51786.8	3277.4	34.1	10	39.2	65041.6	59.9	22.6
11	8064.0	58815.2	16027.5	37.2	11	2395.4	63735.2	4848.5	25.4
12	222.9	62432.0	486.7	31.3	12	8.3	69555.2	43.2	25.3
13	170.4	62127.2	389.0	34.7	13	14.4	64465.6	42.1	22.1
14	1207.6	53032.0	2266.3	34.8	14	11.8	65838.8	98.8	25.5
15	31.8	66853.2	87.0	25.4	15	1611.0	64559.6	3251.8	21.5
16	13515.8	53826.0	23786.6	34.6	16	22.6	66206.8	73.8	24.0
17	3477.8	60369.2	4652.9	29.9	17	26.4	54586.0	76.2	29.9
18	27.6	63909.6	122.0	26.2	18	32.0	64237.2	69.6	18.1
19	58.5	60958.4	163.2	25.6	19	216.6	46578.0	486.7	20.2
20	24.0	66868.4	96.1	20.3	20	35.4	57749.2	38.0	19.6
21	460.9	64375.2	918.9	36.4	21	21.4	64643.2	77.2	24.9
22	582.9	63118.8	1172.4	34.0	22	16133.2	0.0	32440.1	48.0
23	17925.6	66990.4	35899.3	32.0	23	7.9	63412.0	52.8	20.6
24	4064.5	63499.2	8031.8	29.4	24	70.9	59906.4	183.9	20.9
25	129.7	65266.0	347.1	30.9	25	0.6	70466.4	17.1	23.0
26	43.4	63000.4	159.5	32.3	26	41.2	40417.2	26.2	36.7
27	1062.0	65630.4	2009.7	31.6	27	20.6	55873.2	36.0	31.8
28	15498.1	68143.6	31077.1	32.1	28	13.1	64614.8	77.4	30.2
29	233.7	65705.6	473.8	30.3	29	13.0	65283.6	54.5	20.5
average	474.5	62506.9	882.7	34.1	average	45.9	59964.2	100.5	26.4
median	196.7	63059.6	431.4	33.1	median	26.4	64465.6	59.9	24.9
std	589.2	4411.4	1058.7	6.9	std	79.5	13572.4	141.0	6.7

בטבלה 1 ניתן לראות את הניתוח הסטטיסטי שהתקבל, כאשר פתרונות שחרגו ביותר מ-5 סטיות תקן לפחות באחד המדדים נחשב כ- "outlier" והוצא מהחישוב.

טבלה 2 : השוואת מדדים סטטיסטיים בין האלגוריתמים

	IGD [%]	HV [%]	SPREAD [%]	TIME [%]
average	90.3	4.1	88.6	22.4
median	86.6	-2.2	86.1	24.8
std	86.5	-207.7	86.7	2.3

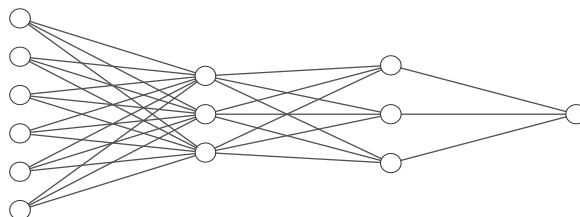
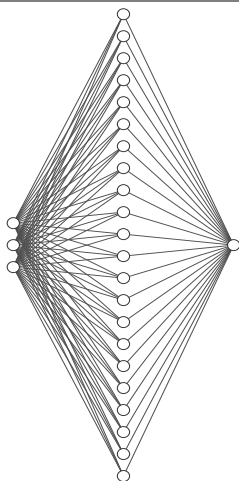
בטבלה 2 יש השוואה בין המדדים הסטטיסטיים עבור שני האלגוריתמים, כאשר הערכים המוצגים בטבלה הם השינוי מהמדדים שהתקבלו באלגוריתם NSGA II למדדים שהתקבלו ב-MOEAD, כאשר אחוז חיובי מעיד על שיפור בממד לטובת MOEAD ואחוז שלילי מעיד על שיפור בממד לטובת NSGA II. וניתן לראות בבירור שיפור בכל המדדים למעט HV לטובת MOEAD, כלומר במדדי האיכות בהם השתמשנו להעריך את הפתרונות רוב הפתרונות שהתקבלו מאלגוריתם MOEAD טובים יותר. יש לציין כי יש לשקול את מדדים אלו מכיוון שכמות הפתרונות אשר התקבלו בחזית פרטו בין השני האלגוריתמים היא גדולה כפי שצויין קודם. עבור הזמן ניתן לראות כי משתנה זה אחיד באופן יחסי בין שני האלגוריתמים ועם סטיית תקן קטנה.

### 6.3 הצגת פתרונות אופניים

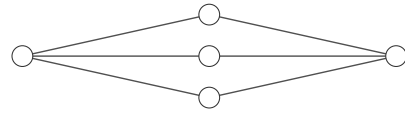
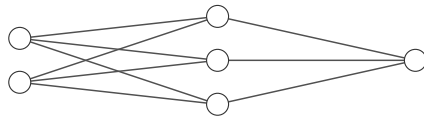
בחלק זה נציג פתרונות אופניים מחזית פרטו המאוחדת, נזכיר כי מטרת האופטימיזציה הייתה למצוא את הארכיטקטורה המינימלית של הרשת עם ביצועי הדיוק הגבוהים ביותר, כאשר הדיוק הוא על מקרי ה-test ולא ה-train, כך שמדד זה מציג באופן נאמן את הדיוק של הרשת.

טבלה 3 : תוצאות הפתרונות האופניים

Solution number	Accuracy	Neural network size
1	100%	290
2	98%	227
3	94.4%	89
4	77.2%	55



איור 14 : פתרונות אופניים, משמאל פתרון 1, מימין פתרון 2



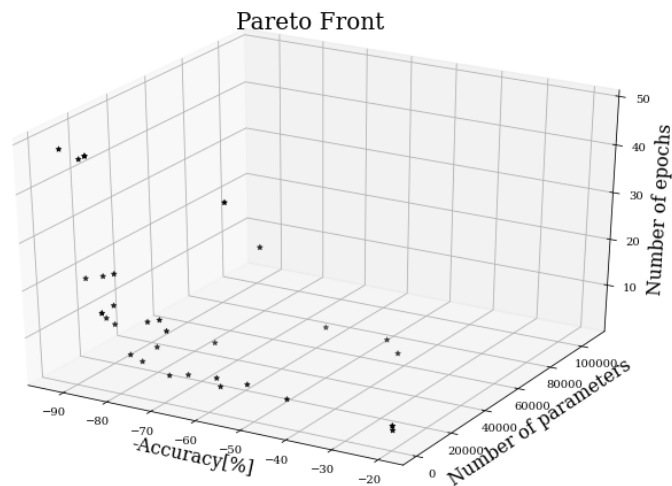
איור 15 : פתרונות אופייניים, משמאל פתרון 3, מימין פתרון 4

ניתן לראות באיור 14 ו-15 חלק מהפתרונות האופייניים של תהליך האופטימיזציה, כפי שניתן היה לצפות ככל שגודל הרשת גדל כך יגדל גם הדיוק וכוח החישוב הנדרש.

#### 6.4 בעיה תלת ממדית (בונוס)

בחלק זה הוספנו פרמטר תכן אשר משמש גם כמטרה, על מנת להפוך את הבעיה לתלת מימדית. פרמטר זה הינו מספר epochs, כלומר מספר מחזורי האימון שהרשת מבצעת, כאשר השאיפה היא לצמצם את מספר מחזורי האימון בכדי לצרוך פחות כוח חישובי באימון הרשת וגם להקטין את זמן אימון הרשת.

לאופטימיזציה זו עשינו שימוש באלגוריתם NSGA II כפי שנלמד בכיתה, עם גודל אוכלוסייה של 60 פריטים ולאורך של 20 דורות.



איור 16 : חזית פרטו תלת מימד

באיור 16 ניתן לראות את חזית פרטו עם 3 מטרות, כאשר ציר  $Z$  הוא הציר שהתווסף לאופטימיזציה. נשים לב כי יש פריסה של פתרונות לאורך ציר זה ויש לו משמעות, כמו כן, פריסת הפתרונות לאורך מטרה זו הוא יותר משמעותי מפריסת הפתרונות לאורך הציר שמייצג את המטרה השנייה באופטימיזציה (מספר הפרמטרים ברשת), וניתן לומר כי עבור אופטימיזציה זו מטרת המיניזציה של Number of Epochs היא בעלת משקל גדול יותר מהמטרה של Number of Parameters.

ניתן לראות בנספח – הקוד המצורף, עוד פלוטים עבור הבעיה התלת מימדית.



## 7 סיכום ומסקנות

במסגרת פרויקט זה בוצעה אופטימיזציה לרשת נוירונים שמטרתה חיזוי וקלסיפיקציה בין חמישה מצבים שונים. ייעדי האופטימיזציה, כלומר מרחב המטרות, כללו דיוק מקסימלי לצד גודל רשת מינימאלי. על מנת לבצע את האופטימיזציה, הוגדרו שלושה פרמטרי תכן הכוללים את מספר השכבות החבויות ברשת, גודלן של השכבות וכן גודל השכבה הראשונה. האלגוריתמים שנבחרו לביצוע האופטימיזציה הם NSGA II ו-MOEA/D.

במסגרת ביצוע הפרויקט עלו מספר מסקנות חשובות:

- ניתן לקבוע שבעבור שני האלגוריתמים התקבלו תוצאות טובות, ובפרט התקבלו חזיתות פרטו איכותיות המציגות פתרונות מדויקים (Accuracy) ובגדלי רשת קטנים יחסית.
- מהסתכלות על קצב התקדמות הפתרונות במרחב המטרות ניתן לומר כי הן ב-2D והן ב-3D מגיעים להתכנסויות מהירות לחזית פרטו אופטימלית.
- האלגוריתם NSGA II הפיק מספר גדול יותר של פתרונות בלתי נשלטים בהשוואה ל-MOEA/D.
- מבחינת המדדים הסטטיסטיים ניתן ללמוד שאלגוריתם MOEA/D הציג ביצועים טובים יותר.
- למרות ההמדדים הסטטיסטיים שנתו לטובת MOEA/D, ניתן לראות כי יש חשיבות רבה להסתכלות ויזואלית על החזית המתקבלת.
- בסעיף הבנוס ניתן לראות שמספר ה epochs הוא פרמטר משמעותי שיכול לסייע באופטימיזציה של רשתות.
- ניתן לראות שכמה מהפתרונות האופייניים שונים זה מזה משמעותית, לכן נדרשת מערכת קבלת החלטות נוספת על מנת לבצע בחירת פתרון ספציפי.

בשל הביצועים האיכותיים שהוצגו בדוח זה ובשל העובדה שהמאמץ המחשובי הדרוש לשימוש באלגוריתמים אלו אינו גדול, שימוש בהם (או באלגוריתמים שונים) בעת עבודה עם רשתות עצביות עשויה לסייע ביעול מבנה הרשת ושיפור הביצועים שלה. בשל כך, ובשל העובדה שיעדי הפרויקט הפרטניים הושגו אנו סבורים שמטרתו הכוללת הושגה.

## 8 מקורות

1. Matplotlib - Python library for data visualization. (n.d.). Retrieved from <https://matplotlib.org/>
2. NumPy - Mathematical Python library. (n.d.). Retrieved from <https://numpy.org/>
3. PYMOO - Python library for Multi Objective Optimization. (n.d.). Retrieved from <https://pymoo.org/index.html>
4. Scikit-learn - python library for ML&AI Programing. (n.d.). Retrieved from <https://scikit-learn.org/stable/>
5. Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," in *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, Dec. 2007, doi: 10.1109/TEVC.2007.892759.