נשתמש ב Pareto-Weighted-Sum-Tuning על מנת לפתור (Multiobjective, Multicriteria) Optimization Problems. נכתב על ידי HarryWang\_LOD

בהינתן אוסף של מניות,אנחנו נרצה למקסם 2 מטרות- להרוויח כמה שיותר ממנייה כלשהי וגם להפסיד ממנה כמה שפחות.  
נשתמש ב Weighted-Sum לשם כך.בקוד שלנו אנחנו יוצרים אוסף של N מניות,אשר כל אחת מכילה את הרווח המרבי ואת ההפסד המרבי.

המטרה:

המשקל של אילוץ   
 הכמות של אילוץ

*כאשר*

הקושי הוא להבין מה המשקל של כל אילוץ,כמה הוא חשוב לנו.

ניתן ל דומגאות ספציפיות של

כאשר:

למנייה מוגדר :

*- הרווח הגדול ביותר האפשרי להשגה ממנייה .*

*- ההפסד המיריבי שנספוג ממנייה*

*כלומר :*

*ניתן ל* לדרג את ה ,כלומר לתת משקל לכל ובכך נגדיר את חשיבותו לעומת אחרים.   
כאשר הדירוג נע מ0 -ה הגרוע ביותר עד ה הטוב ביותר.

***הפסדו קוד:***

*-ניצור וקטור משקלים[אלפא וקטור] ריק.  
-   
- ניצור את ה כפי שהוסבר מקודם.  
-ניצור מערך =מכיל גדלים שונים לBATCH   
-נגדיר מספר איטרציות מירבי=EPOHCES  
-ניקח מידע מהמשתמש: alpha\_vector, tolerance\_vector וכך ניצור פרופיל למשתמש  
-נגדיר alpha\_vectors\_learned מערך ריק*

*ניצור מה בגודל*

1. *-לכל J ב :*
   1. *לכל אשר קטן ממספר EPOHCES:*
      1. *ניקח את הBATCH שיצרנו ואת המידע שהמשתמש נתן וניצור alpha\_one\_vector\_learned /// זה נעשה כאילו על ידי "המשתמש" בעזרת Ranking-SVM.המחשב מדמה משתמש ומדרג כל TUPLE ,כמו שסברתי מקודם ומסדר לפי הסדר. בסוף הוא מחזיר כאילו מחזיר לנו alpha\_one\_vector\_learned [לא ברור לי מה זה.לבדוק אצל אור]אולי זה מחזיר את* 
      2. *ניצור BATCH חדש - ניקח BATCH בגודל מבין כל הזוגות שלא היו BATCH הקודם.*
      3. *נוסיף ל alpha\_vectors\_learned את ה alpha\_one\_vector\_learned הנוכחי שלמדנו*
      4. *בmean\_alpha\_vector\_learned נשים ממוצע של כל הוקטורים אשר נמצאים ב alpha\_vectors\_learned*
      5. *נשמור את השגיאה*
   2. *נעשה עוד כמה דברים*
   3. *נחזיר mean\_alpha\_vectors[-1]*

*mean\_alpha\_vectors מכיל ה .כאשר הינו הממוצע של ,בעצם המשקל הממוצע שנתנו למנייה .*

*Text, letter

Description automatically generated*