

## פרויקט סיום קורס "מבוא לאופטימיזציה"- פתרון בעית "מציאת צוות עובדים אופטימלי"- דו"ח עבודה-

חלק ראשון- שקלול תכונות אופי ומקצועיות (איכות) תוך הגעה למינימום עלות:

**הבהרה שרצינו להדגיש:** הרעיון, כלל האילוצים, תתי המטרות ומטרת העל ודרכי ההתמודדות איתה, החשיבה על הבעיה והחשוב ביותר- כלל הקוד והלמידה וההתמודדות עם החבילה הנדרשת לכך הם עבודה אישית שלנו, שום קוד בפ'יצר זה לא הועתק מהאינטרנט ונעשתה פה עבודה תאורטית ומעשית רבה (מאוד!) שעוד תתפתח ותגדל במהלך שידורג הפ'יצר והבעיה בפרויקט הגמר.

### הצגת הבעיה והמוטיבציה לאפליקציה :

בכל פרויקט\צוות עבודה דרושים תחומי ידע שונים אשר הוצאתם לפועל תביא לידי תוצר סופי (אחד או רבים) לשמו הצוות נוצר. לכל ראש צוות עולה השאלה-

"כיצד מוגדר העובד האידיאלי? ובעקבות שאלה זו עולות שאלות נוספות כמו:

"האם מדובר בעובד הדורש סכום מינימלי עבור עיסוקו?", "האם אופיו הוא החשוב או בכלל המקצועיות שלו בתחום?",

"אולי בעצם כלל הפרמטרים האלו חשובים?"...

וגם כאשר יש לראש הצוות את התשובה לכמה כל אחד מפרמטרים אלו חשוב לו במועסק החדש, עולה שאלה חדשה של איך לשקלל כלל פרמטרים אלו עבור כל מציאת הצוות, וכיצד צוות זה יעבוד במשותף ויצור שיתופי פעולה טובים ביותר.

כלומר, לאחר שאילת קושיות אלו, ראש פרויקט\צוות שם לנגד עיניו דרישות ורצונות במציאת צוות בעל מכלול תכונות אופטימלי- (בתכונות אילו נכללות תכונות המוזדוות את אופיו של העובד ותכונות המוזדוות את רמת מקצועיותו) ואף שם לנגד עיניו אילוצים אישיים כמו תקציב ושעות נדרשות לכל משרה, ואף המלצות או דרישות ממשלתיות כמו העסקת נשים ובעלי מוגבלויות. אך כמובן שכל ראש פרויקט\צוות נותן חשיבות שונה לכל אחד מתחומים אלו, ואף משקל כל תכונה נדרשת בעובד האידיאלי משתנה בהתאם לאופי הפרויקט.

תהליך בניית צוות העבודה "מבזבז" שעות רבות של ראיונות עבודה רבים ובדיקת פרמטרים שונים בעזרת צוות משאבי האנוש. גם לאחר כל תהליך זה, לעיתים ראש הצוות לא מגיע לשקלול האופטימלי הנדרש עבורו.

יתרה מכך, ישנם ראשי צוות המנהלים פרויקטים קטנים או קצרים, כך שתהליך בניית הצוות האופטימלי לפרויקט הנוכחי יכול להיות מייגע ואף לא כלכלי וחורג מגבולות הזמן הנדרשים.

הפ'יצר הראשון שנציג פותר את כלל בעיות אלו!

### הצגת הפיתרון בעזרת תיאור כללי של האפליקציה:

הפ'יצר הראשון הוא כלי עזר לבניית צוות עבודה. בעזרת נתונים בסיסיים הנאספים בשלב הראיונות הראשוני אנו מגיעות לצוות אופטימלי ביותר אשר עומד במטרות-

#### 1. מיזעור עלות

2. מיזעור דירוגי עובדים מבחינת מקצועיות (דירוג נמדד בטווח 1 עד 10 כך ש1 הוא הדירוג הטוב ביותר). ע"י כך תמוקסם מקצועיות הצוות. (סך תכונות אלו ישוקללו עבור כל עובד בהתאם למשקל הניתן ע"י ראש הצוות לכל תכונה)

3. מיזעור דירוגי עובדים מבחינת תכונות אופי (דירוג נמדד בטווח 1 עד 10 כך ש1 הוא הדירוג הטוב ביותר). ע"י כך תמוקסם איכות הצוות. (סך תכונות אלו ישוקללו עבור כל עובד בהתאם למשקל הניתן ע"י ראש הצוות לכל תכונה).

הערה: בתכונות נכללים אלמנטים חשובים העשויים להשפיע על הקשר בין העובדים וכתוצאה מכך על שיתופי הפעולה העשויים להווצר ביניהם- דבר המתקשר לפ'יצר השני שנציג

בין היתר בפרויקט זה אנו נחקור כיצד מתן חשיבות שונה לכל אחד משלושת מטרות אלו משפיע על הקבוצה האופטימלית.

כלומר, כחלק מפרויקט זה אנו עוסקות בפתירת בעיה מסוג **multi objective optimization**

כפי שצינו לעיל- על מנת לפתור בעיה מסוג זה -כלל המטרות יתבטאו במטרת על אחת- עליה נרצה למצוא את הערך המינימלי, כך שכל אחת מבין שלושת המטרות תקבל משקל שונה בהתאם לצרכי ובחירת המשתמש.

## קלט לאפליקציה

כלל הקלטים ינתנו ע"י משתמש האפליקציה בתחילת השימוש, חלק ע"י הקלדת הנתון הנדרש וחלק ע"י צירוף קובץ כפי שנתאר כעת- הקלט נאסף משני גורמים

\*ראש הפרויקט (מידע על הפרויקט עצמו)

\*משאבי אנוש (מידע על כל מועמד)

הערה: המשתנים המסומנים בבחול יבואו לידי ביטוי בקוד המצורף בשמות אלו או מקוצרים

קלטים שניתנים ע"י המשתמש (ראש הפרויקט):

1. קובץ משרות שיש לאייש (ניתן בקלט לתוכנה)

כל רשומה היא משרה עם האלמנטים הבאים (לפי סדר, מופרדים בפסיק):

- מספר מזהה של המשרה.
- שעות יומיות נדרשות למשרה  $H_j$
- ותק נדרש למשרה  $S_j$

המידע המוסק מכך הוא המשתנים:

- ❖ numJobs : מספר המשרות שיש לאייש.
- ❖ jobs\_arr : מערך שמסדר את כלל הנתונים הללו בצורה מסודרת.

2. gExpenses : סך התקציב המקסימלי המוקצה ליום בפרויקט.

3. extraHPayment : אחוז תוספת תשלום עבור שעה נוספת (1.25 לדוגמא שווה ערך ל-125% מהשכר הרגיל לשעה- כמקובל בחוק)

4. חשיבות לכל תכונת אופי של עובד, ככל שתכונה חשובה יותר הערך שלה גדול יותר (חשיבות ניתנת בין 1 ל-10)

- team : יכולת עבודה בצוות.
- selfLearn : יכולת עבודה עצמית.
- loyal : אמינות.
- Passion : תשוקה למקצוע
- persistence : התמדה
- adaptability : יכולת הסתגלות

הערה: התכונות וסידורן קבוע

המידע המוסק מכך הוא המשתנה:

- ❖ Personal\_weight - וקטור משקולות של התכונות אופי

5. חשיבות לכל תכונת מקצועיות של עובד, ככל שתכונה חשובה יותר הערך שלה גדול יותר (חשיבות ניתנת בין 1 ל-10)

- anilitic : אנליטיות
- creative : יצירתיות
- Planning : יכולת תכנון
- Communnation : יכולת תקשורתית
- Problemsolve : יכולת לפתירת בעיות

הערות: התכונות וסידורן קבוע

המידע המוסק מכך הוא המשתנה:

- ❖ Pro-weight - וקטור משקולות של התכונות מקצועיות

חשיבות נוספת שנותן המשתמש באותו הנושא היא -

Seniority\_w - חשיבות למספר שנות הותק של העובד (נכנס רק בממוצע משוקלל)

6. המשקל הניתן לכל אחד לשלוש האלמנטים במטרת העל, כך שסכום המשקלים הוא 10. כלומר-

- $costF1 + costF1 + costF1 = 10$  (float costFi הוא מטיפוס float) לדוגמא-  
התחשבות ברצון להגעה למינימום תשלום עבור צוות העובדים-2 ( $costF1$ )
- התחשבות ברצון ליצירת צוות שסך דירוגי האופי שלו טובים ביותר (מתוך רצון לבדוק כמה דירוג זה משפיע על שיתופי פעולה פורים ונשנים)-3 ( $costF2$ )
- התחשבות ביצירת צוות שסך דירוגי המקצועיות שלו טובים ביותר-5 ( $costF3$ )

קלטים שמולאו ע"י משאבי אנוש/צוות מראיינים ראשוני:

קובץ מועמדים, בכל רשומה:

הערה: נשתמש באינדקסים j, i: כאשר i בטווח [1,numInterview] וכאשר j בטווח [1,numJobs].

1. שם עובד \ מספר מזהה עבור העובד
2. שם המשרה \ מספר מזהה עבור המשרה אליה הוא התראיין
3. ותק למשרה
4. דירוג עבור תכונות האופי של המראיין. הדירוג ניתן בטווח 1-10, כאשר 1 הוא הדירוג הכי טוב. הדירוג ניתן לפי הסדר הבא ויכנס למשתנים המתאימים עבור כל עובד, כאשר כל דירוג מופרד בפסיק-

- o team : יכולת עבודה בצוות.
- o selfLearn : יכולת עבודה עצמית.
- o loyal : אמינות.
- o Passion: תשוקה למקצוע
- o persistence : התמדה
- o adaptability : יכולת הסתגלות

הערה: התכונות וסידורן קבוע

המידע המוסק מכך הוא המשתנה:

- ❖ Personal\_w\_i - וקטור משקולות של התכונות אופי עבור כל עובד, כך שכל תכונה קיבלה את המשקל אשר בחר לה הבוס
5. דירוג עבור תכונות המקצועיות של המראיין. הדירוג ניתן בטווח 1-10, כאשר 1 הוא הדירוג הכי טוב. הדירוג ניתן לפי הסדר הבא ויכנס למשתנים המתאימים עבור כל עובד, כאשר כל דירוג מופרד בפסיק-

- o anilitic : אנליטיות
- o creative : יצירתיות
- o Planning: יכולת תכנון
- o Communtation: יכולת תקשורתית
- o Problemsolve: יכולת לפתירת בעיות

הערה: התכונות וסידורן קבוע

המידע המוסק מכך הוא המשתנה:

- ❖ Professional\_w\_i - וקטור משקולות של התכונות מקצועיות עבור כל עובד, כך שכל תכונה קיבלה את המשקל אשר בחר לה הבוס
- מכלל 11 הדירוגים הללו מתקבל סכום דירוג התכונות של עובד- סכום זה נע בין 11 ל110 (מכיוון שכל דירוג תכונה נע בין 1 ל10 ויש 11 תכונות סה"כ)-

Sum\_attributes\_i - נע בין 11 ל110 עבור כל עובד

6. התשלום לשעה שביקש העובד:  $R_i$ .
7. שעות משרדיות (רגילות) ליום בפרויקט שעובד יכול לעבוד:  $H_i$ .  
(הבהרה: יוכנס למשתנה זה מינימום בין השעות הנדרשות למשרה אליה התמייין לבין הערך שנתן, כחלק מעיבוד מקדים שמונע מצב בו העובד יעבוד במשרה יותר ממה שנדרש, מתוך הנחה שכבר בראיון הראשוני הוא ידע מה השעות הנדרשות למשרה שבחר)
8. מין העובד - m/f  
מוטיבציה: כנשים צעירות הנכנסות בקרוב ל"עולם של הגדולים" (ויש יגידו לעולם של הגברים) חשוב לנו לעלות למודעות את חשיבות העסקת נשים בכל פרויקט/צוות עבודה, לכן באמצעות פרמטר זה נקבע אילוץ להעסקת נשים.
9. האם העובד בעל צרכים מיוחדים/בעל מוגבלות פיזית-handicapped/ not\_handicapped  
מוטיבציה: כחלק מחוקי העסקה- על כל חברה יש חובה להעסיק עובדים בעלי צרכים מיוחדים על מנת לשלבם בקהילה, היה חשוב לנו לתת לערך חשוב זה מקום ואכן לעמוד בדרישות ואילוץ נוספים פחות "טרוויאליים" ואף לעודד מקומות עבודה להעסיקם בעזרת תגמול כספי ומימון חלקי
10. האם העובד ורסטילי או לא  
מוטיבציה: אדם ורסטילי הוא אדם בעל כישורים מגוונים המסוגל להתאים את עצמו למצבים שונים. אדם זה לצורך העניין יוכל לתת עזרה לעובד מתחום שונה בעת לחץ, להחליף עובד בעת חופשה, ליצור שיתופי פעולה עם עובדים שונים מתוך הבנה בתחומיהם, ואף לבצע את תפקידם בעת מחלה, חל"ת או פיטורים, דבר שמאוד נחוץ, בטח בתקופת הקורונה בה אנו נמצאים, עובדים ורסטילים יחסכו לנו משאבים כמו כסף או זמן בעת הצורך ולכן נאלץ (בהמשך) שלפחות חמישית מהם יהיו בעלי תכונה זו

המידע המוסק מקריאת קובץ זה הוא המשתנים:

- ❖ numInterview: מספר האנשים בקובץ המועמדים (מספר האנשים שרואיינו)
- ❖ candidates\_arr: מערך של רשימות המכיל את כלל המידע על המראיינים
- ❖ threshold: מספר סף שיקבע המשתמש הנע בין 11 ל110 (טווח זה הוא הטווח של סכום ערכי הדירוגים של תכונות העובד, כך שדירוג טוב הוא קטן ביותר) אם סכום ערכי הדירוגים של תכונות העובד יהיה גדול ממספר זה- העובד לא יתקבל

מילוני עזר התלויים בdata בלבד (ממולאים ידנית וייחשבו כקבועים):

1.  $isCandidate_{ij}$  : מילון של מילונים בו כל תא מכיל ערך בינארי המבטא האם המועמד ה-i הגיש מועמדות למשרה ה-j.

2.  $S_{ij}$  : מילון של מילונים בו כל תא מכיל את הותק של העובד ה-i במשרה j (יקבל ערך 1- אם לא הגיש מועמדות)

3.  $isPro_{ij}$  : מילון של מילונים בו כל תא מכיל ערך בינארי המבטא האם העובד ה-i עומד בותק הנדרש למשרה ה-j.

$$IsPro_{i,j} = \begin{cases} 1, & S_{i,j} - S_j \geq 0 \\ 0, & S_{i,j} - S_j < 0 \end{cases}$$

כלומר יקבל את הערך 1 אם ורק אם ההפרש בין שנות הותק של שיש לו פחות שנות הותק הנדרשות גדול שווה 0.

4.  $Payment_{ij}$  - מילון של מילונים בו כל תא מכיל מה השכר הכולל שישולם לעובד i עבור משרה j אם אכן יעסוק בה (אם לא הגיש מועמדות למשרה הערך כמובן יהיה 0)

5.  $Gender_i$  - מילון בו בכל תא יש את מין העובד (1 אם אישה, 0 אם גבר)

6.  $Handicapped_i$  - מילון בו בכל תא מכיל אם העובד בעל צרכים מיוחדים או לא (1-כן, 0-לא)

7.  $handicapped\_scholarship$  - משתנה שיקבל ערך 1 אם המועמד אינו בעל צרכים מיוחדים, אחרת- יקבל את האחוז היחסי שצריך לשלם עבור שעה של אדם מוגבל, משתנה זה מקובע כעת לערך 0.85, כלומר ישנו תקצוב של 15% לשכר שדרש העובד בעל הצרכים המיוחדים, אחר שיקלול ערכים רלוונטים של החוק בישראל)

8.  $Versatile_i$  - מילון בו בכל תא מכיל אם העובד בעל תכונת הורסטליות או לא (1-כן, 0-לא)

משתנים חשובים נוספים-

1.  $Prob$  - אובייקט של בעית האופטימיזציה, מאותחל להיות  $LpMinimize$  (מתוך היותנו רוצות למצוא מינימום למטרת העל)

2.  $Job\_ind$  - רשימת אינדקסים עבור המשחות (אינדקס יכול להיות מילולי או מזהה משרה)

3.  $Worker\_inds$  - רשימת אינדקסים עבור העובדים (אינדקס יכול להיות מילולי או מזהה עובד)

4.  $Matruxe\_inds$  - רשימת האינדקסים עבור מטריצה בגודל  $numJobs * numInterview$ , מטריצה זו נבנתה ע"י מכפלה של האינדקסים של שתי המטריצות הקודמות ( $Job\_inds$  ו- $Worker\_inds$ )

5.  $F\_percent$  - המספר המינימלי של נשים שיש להעסיק במידה ולפחות רבע מהעובדים חייבות להיות ממין נקבה

6.  $V\_percent$  - המספר המינימלי של עובדים וירסטילים שיש לקבל במידה ולפחות חמישית מהעובדים חייבים להיות בעלי תכונה זו

7.  $Costf1$  - התחשבות ברצון להגעה למינימום תשלום עבור צוות העובדים.

8.  $Costf2$  - התחשבות ברצון ליצירת צוות שסך דירוגי האופי שלו טובים ביותר (מתוך רצון לבדוק כמה דירוג זה משפיע על שיתופי פעולה פורים ונשנים).

9.  $Costf3$  - התחשבות ביצירת צוות שסך דירוגי המקצועיות שלו טובים ביותר.

משתני החלטה:

$isJob_{ij}$  : האם העובד ה-i יועסק במשרה j

משתני עזר התלויים במשתני החלטה:

$AllPay_i$  : תשלום יומי כולל עבור עובד i, משתנה זה מקבל את הערך  $\sum_{j=1}^{numJobs} isJob_{ij} * payment_{ij}$

#### אילוצים:

1. ניסוח פורמאלי-  $\forall i, j: \text{IsCandidate}_{i,j} \geq \text{isJob}_{i,j}$   
 הסבר מילולי-עובד שלא הגיש מועמדות למשרה אינו יכול להתקבל למשרה
2. ניסוח פורמאלי-  $\forall i, j: \text{IsPro}_{i,j} \geq \text{isJob}_{i,j}$   
 הסבר מילולי- עובד שאינו עומד בוותק הנדרש לתפקיד אינו יכול להתקבל למשרה
3. ניסוח פורמאלי-  $\sum_{i=1}^{\text{numInterview}} \text{AllPay}_i \leq \text{gExpenses}$   
 הסבר מילולי-סכום התשלום היומי עבור העובדים אינו חורג מהמקסימום הקיים
4. ניסוח פורמאלי-  $\forall i: \sum_{j=1}^{\text{numJobs}} \text{isJob}_{i,j} \leq 1$   
 הסבר מילולי-מראיין אינו יכול להתקבל ליותר ממשרה אחת
5. ניסוח פורמאלי-  $\forall j: \sum_{i=1}^{\text{numInterview}} \text{isJob}_{i,j} = 1$   
 הסבר מילולי- כל משרה בפרויקט חייבת להיות מאויישת
6. ניסוח פורמאלי-  $\sum_{i=1}^{\text{numInterview}} \sum_{j=1}^{\text{numJobs}} \text{isJob}_{i,j} = \text{numJobs}$   
 הסבר מילולי- כלל האנשים שהתקבלו שווה למספר המשרות הנדרש לפרויקט
7. ניסוח פורמאלי-  $\sum_{i=1}^{\text{numInterview}} \sum_{j=1}^{\text{numJobs}} \text{isJob}_{i,j} * \text{gender}_i \geq \text{f\_percent}$   
 הסבר מילולי- אחוז הנשים בחברה עומד בדרישת המשתמש
8. ניסוח פורמאלי-  $\sum_{i=1}^{\text{numInterview}} \sum_{j=1}^{\text{numJobs}} \text{isJob}_{i,j} * \text{versatile}_i \geq \text{v\_percent}$   
 הסבר מילולי- אחוז העובדים הורסטילים בחברה עומד בדרישת המשתמש
9. ניסוח פורמאלי-  $\forall i: \sum_{j=1}^{\text{numJobs}} \text{isJob}_{i,j} * \text{sum\_attribute}_i \leq \text{threshold}$   
 הסבר מילולי- סך דירוגי תכונות העובד (אופי ומקצועיות) אינו חורג מהסף שהגדיר המשתמש

#### מטרות:

1. מינימום תשלום יומי – נרצה למצוא את הערך המינימלי עבור
- $$F1 = \sum_{i=1}^{\text{numInterview}} \text{AllPay}_i$$
2. מינימום דירוגי תכונות אופי (דירוג נמוך ביותר הוא דירוג טוב ביותר, כמו בתחרות)– נרצה למצוא את הערך המינימלי עבור -

$$F2 = \sum_{i=1}^{\text{numInterview}} \sum_{j=1}^{\text{numJobs}} \text{personal\_w}_i * \text{isJob}_{i,j}$$

- 3 מינימום דירוגי מיקצועיות –(דירוג נמוך ביותר הוא דירוג טוב ביותר, כמו בתחרות) נרצה למצוא את הערך המינימלי עבור -

$$F3 = \sum_{i=1}^{\text{numInterview}} \sum_{j=1}^{\text{numJobs}} \text{professional\_w}_i * \text{isJob}_{i,j}$$

לאחר חקר הנושא, דרך ההתמודדות שבה בחרנו עם בעיה מסוג –

multi objective optimization היא מתן משקל לכל אחת מהמטרות וכך מתקבלת מטרת על משוקללת אחת-

$$\text{Costf1} * \text{f1} + \text{Costf2} * \text{f2} + \text{Costf3} * \text{f3}$$

## דוגמת הרצה לנוחות המשתמש-

דוגמת הרצה זו מתבססת על הקבצים המצורפים בקוד המתבססים על 130 מועמדים ו501 משרות:

```
C:\Users\konopatz\AppData\Local\Programs\Python\Python38\python.exe C:/Users/konopatz/PycharmProje
Welcome to Optimal Team Builder System.
Please follow the instructions.
please enter a path for a jobs file, with the following structure:
1. The file must be a text file.
2. format:
JOB_NAME.REQUIRED_HOURS.REQUIRED_SENIORITY
3. Must end with enter.
4. For example: CEO.500.10
Enter now:
Jobs.txt
File not found. Please try again.
Enter now:
Jobs50.txt
Now, please enter the following information:
Please enter the general expenses for day.
Please enter an integer number.
If you will enter a non-integer, the system will request you to try again.
100000
Please enter the amount of extra hours payment.
For example: for 25% more for each extra hour, please enter 1.25.
1.25
Now please enter the 'cost' for each personal and professional attribute of the candidates.
The more the number is bigger, the attribute is more important for you.
The numbers must be at range of 1-10 and integers. If not, the system will request you to try again
Personal attributes:
team player: 4
self learning: 2
loyalty: 6
passionate: 4
persistence: 5
adaptability: 2
Professional attributes:
analytic: 4
creative: 1
planning: 8
communication: 7
problem solving: 6
Now select the importance of the seniority of the candidates.
8
Now please choose the cost of each objective.
The more the number is bigger, the objective is more important for you.
The target is to minimize the objective function, and to minimize each category.
The sum of the costs should be 10. If not, the system will request you to try again.
minimal payment cost: 3
minimal personal cost: 3.5
minimal professional cost: 3.5
please enter a path for a candidates file, with the following struc
1. The file must be a text file.
2. format:
NAME.JOB_NAME.SENIORITY_AT_THIS_JOB.{LIST_OF_PERSONAL_ATTRIBUTES}.{LIST_OF_PROFESSIONAL_ATTRIBUTES
3. Must end with enter.
4. For example: Bob.CEO.20.{1,2,4,1,3,1}.{6,1,3,3,4}.150.800.m.not_handicapped.v
Note: for the attributes - the more the number is bigger, it means that the candidate is less fit.
Enter file name:
candidates.txt
File not found. Please try again.
Enter file name:
candidates50.txt
Please enter also a number between 11-110, an integer.
The number will represent the sum of acceptance threshold of the attributes.
For example: if these are the attributes: {1,5,2,1,7,1}.{3,2,6,4,6} you must enter a number bigger
If you will enter 110, you can accept each worker, and if you enter 11 you can't accept anyone.
enter threshold: 90
```

```
candidates50.txt
Please enter also a number between 11-110, an integer.
The number will represent the sum of acceptance threshold of the attributes.
For example: if these are the attributes: {1,5,2,1,7,1}. {3,2,6,4,6} you must enter a number bigger
If you will enter 110, you can accept each worker, and if you enter 11 you can't accept anyone.
enter threshold: 90
C:\Users\konopatz\AppData\Local\Programs\Python\Python38\lib\site-packages\pulp\pulp.py:1199: User
warnings.warn("Spaces are not permitted in the name. Converted to '_'")
Welcome to the CBC MILP Solver
Version: 2.9.0
Build Date: Feb 12 2015

command line - C:\Users\konopatz\AppData\Local\Programs\Python\Python38\lib\site-packages\pulp\api
At line 2 NAME          MODEL
At line 3 ROWS
At line 11381 COLUMNS
Result - Optimal solution found

Objective value:           106975.250000000
Enumerated nodes:           0
Total iterations:           0
Time (CPU seconds):         0.09
Time (Wallclock seconds):   0.09

Option for printingOptions changed from normal to all
Total time (CPU seconds):     0.21   (Wallclock seconds):     0.21

This is the solution:
Optimal solution found!
This is the minimal cost of the objective that meets all the constraints:
Total Costs = 106975.25000000001
Total payment: 33172.25
The following workers were selected:
1GEORGE: CEO
5DAVID: Vice_President
7SHONE: Senior_Account_Manager
8MIKE: Executive_Assistant
10MIKEL: Senior_Business_Development
11VERONICA: Business_Development
13TOM: Enterprise_Account_Manager
15RONA: Senior_Associate
16TOMMY: Senior_Data_Engineer
19DYLAN: Data_Engineer
21MATTHIEW: Data_Scientist
23JONATHAN: Junior_Data_Scientist
25SHARON: Business_Intelligence_Manager
30NOA: SR_Manager
32STUART: SR_Data_Scientist
33LIBBY: Business_Analyst
36BILLY: NLP_Data_Scientist
38ASHTON: Student_Data_Engineer
39JENNY: Technical_Program_Manager
43MARRY: Optimization_Data_Scientist
44JHON: HR_Assistant
45BOB: Recruiter
47MIKI: HR_Manager
49MONTY: Recruiting_Manager
52JINNI: Marketing_Manager
53DANI: Marketing_Coordinator
54DANIEL: Senior_Solutions_Architect
55DANIELLA: Solutions_Architect
58MARK: Application_Engineer
63MIKA: Android_Engineer
65EYLON: SDE
```

```
67PHIL: SDET
68CLEO: QAE
71EMA: Student_QA
72ANNA: Junior_QA
74YONI: Senior_QA
77JULY: Development_Manager
79MAYA: Technical_Trainer
80RAYEN: Instructional_Designer
82REBEKA: Sales_Trainer
83DIN: IT_Trainer
86MIKI: Hardware_Development_Engineer
88VICTOR: Junior_Hardware_Development
89VIKI: Hardware_Validation_Engineer
96BILLY: Product_Design_Manager
97BONNI: Product_Design_Engineer
101LIA: Hardware_Platform_Engineer
79MAYA: Technical_Trainer
80RAYEN: Instructional_Designer
82REBEKA: Sales_Trainer
83DIN: IT_Trainer
86MIKI: Hardware_Development_Engineer
88VICTOR: Junior_Hardware_Development
89VIKI: Hardware_Validation_Engineer
96BILLY: Product_Design_Manager
97BONNI: Product_Design_Engineer
101LIA: Hardware_Platform_Engineer
104PHILIP: Machine_Learning_Scientist
107TOMAS: Manager_Research_Science
108RAYEN: Data_Sciences_Leader

Process finished with exit code 0
```

דוגמת הרצה זו מתבססת על הקבצים בעלי 50 משרות ו111 עובדים, כוללת השמת קלטים לא נכונים ומראה את ההתמודדות עם מקרה קצה



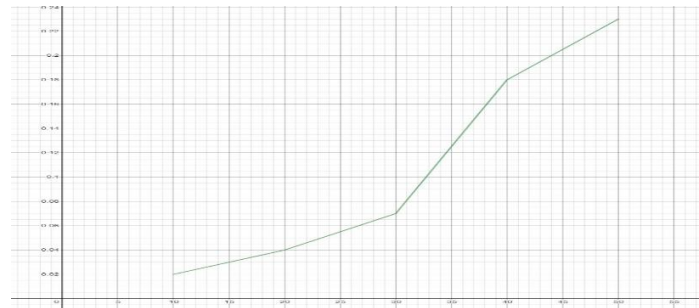
### מסקנות חשובות בעזרת סדרי גודל שונים ובחינת פרמטרים שונים הנוגעות לפיתרון האופטימלי ולזמן הריצה עבורו-

את המסקנות נציג באמצעות גרפים היכולים לסייע למשתמש לבחור פרמטרים שונים לבעיה כמו כמות העובדים שהוא מעוניין לראיין בהתחשבות באילוצין האישיים כמו תקציב זמן וכן נסייע לו לבחור משקל עבור כל אחת ממטרותיו בהצגת הצבות שונות והצגת costn בהתאמה-

גרף עבור זמני ריצה-

ציר ה-x – מספר המשרות שיש לאייש (נשים לב שעבור 10 משרות ריאיינו 20 אנשים, עבור 20 משרות ריאיינו 43 אנשים, עבור 30 משרות ריאיינו 64 אנשים, עבור 40 משרות ריאיינו 83 אנשים, עבור 50 משרות ריאיינו 111 אנשים- הקצבים מצורפים)

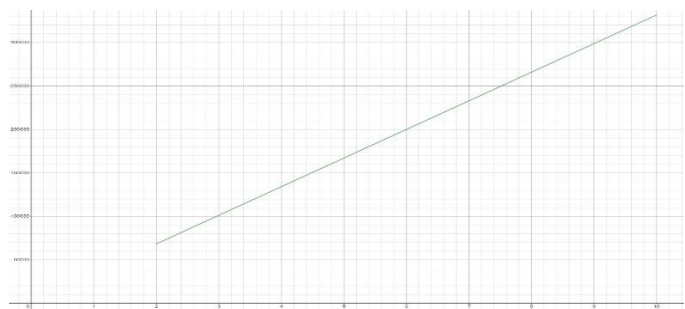
ציר ה-y – זמן הריצה



גרף עבור השפעת cost1 על costn הכללי-

ציר ה-x – חשיבות cost1 כך שcost2 ו-cost3 מתחלקים שווה בשארית 10

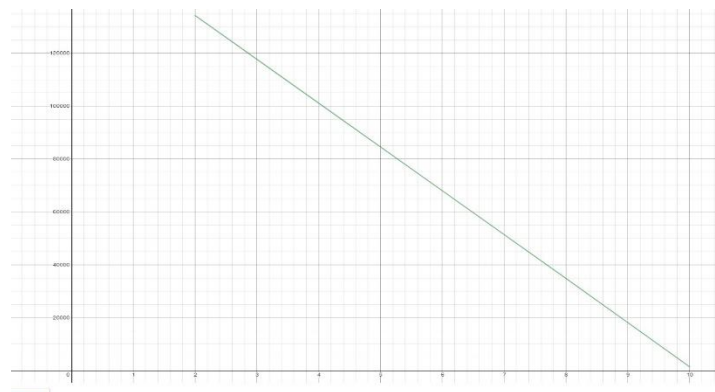
ציר ה-y – costn הכללי



גרף עבור השפעת cost2 על costn הכללי-

ציר ה-x – חשיבות cost2 כך שcost1 ו-cost3 מתחלקים שווה בשארית 10

ציר ה-y – costn הכללי



נשים לך שמהגרפים עולים התובנות שהעלות והתכונות מתנהלים הפוך- ככל שנותנים לעלות חשיבות גדולה יותר כך costn הכללי גדל

ובכל שנותנים לתכונות אופי/מקצועיות חשיבות גדולה יותר כך cost הכולל קטן עבור הערכים בקבצים שהצגנו.

### חלק שני- יצירת :

בפ' יצר הראשון הצגנו כחלק מהבעיה פרמטרים שונים כמו תכונות אופי ומקצועיות ואף ורסטיליות העובד העשויים להשפיע על יעילות צוות העבודה ככלל ועל שיתופי הפעולה בניהם בפרט. כעת נציג כעת פ' יצר קטן נוסף שיפותח בהמשך כחלק מפרויקט הגמר המתבסס על הנחה נוספת הנוגעת לשיתופי פעולה.

#### הצגת הבעיה-

כחלק מפרויקט הגמר, אנו משותפות במעט במחקר העוסק בין היתר בקשר בין גודל הקבוצה לבין שיתופי הפעולה ככלל וכמה שיתופי פעולה אלו נשנים או מובילים לתוצאות חזקות בפרט, ישנה הנחה (שנעסוק בה רבות בהמשך הפרויקט) הטוענת שבצוות קטן יכולים להיוולד שיתופי פעולה פורים יותר- לשם כך נציג אלגוריתם נוסף "set-cover" בהקבלה לבעיה למציאת צוות עובדים אופטימלי.

כידוע הבעיה היא: נתונה קבוצה סופית  $S$  של קבוצות סופיות, שאיחודן הוא הקבוצה  $U$ , יש למצוא תת-קבוצה של  $S$  בגודל מינימלי, שאיחוד הקבוצות בה הוא  $U$ , כלומר יש למצוא את הכיסוי הקטן ביותר לקבוצה  $U$  שהוא תת-כיסוי של  $S$ .

בבעיה שלנו  $U$  היא רשימת תחומי המקצועיות/התעסוקה שיש במשרות שיש לאייש לפרויקט. כל אדם הוא קבוצה של תחומי מקצועיות בהם הוא יכול להיות שותף ולתרום לפרויקט ובעצם כל אדם "מכסה" תחומי מקצועיות בצוות, כלומר כל אדם הוא קבוצה סופית וכלל האנשים המרואיינים הם הקבוצה  $S$ , נרצה למצוא את הכיסוי הקטן ביותר (ובעצם את מספר האנשים המצומצם ביותר לפרויקט מתוך הנחה שבקבוצה זו שיתופי הפעולה יהיו פורים ביותר) לקבוצה  $U$  (תחומי התעסוקה) שהוא תת-כיסוי של  $S$  (קבוצת המרואיינים).

מזהה העובד בקבצים הוא מספר, ומזהה המשרה הוא אות/שם

#### דוגמאת הרצה-

```
setCover x
C:\Users\konopatz\AppData\Local\Programs\Python\Python38\python.exe C:/Users/konopatz/PycharmProje
Enter the name of a jobs files:jobsFile.txt
Enter the name of a candidates files:candidatesFile.txt
This is the cover:
The following workers selected:
Worker: 10, Jobs: {'j', 'v', 'n', 'f', 't'}
Worker: 12, Jobs: {'s', 'w', 'c', 'd'}
Worker: 1, Jobs: {'a', 'u', 'y'}
Worker: 6, Jobs: {'q', 'l', 'p'}
Worker: 8, Jobs: {'h', 'r', 'i'}
Worker: 11, Jobs: {'m', 'j', 'a', 'k'}
Worker: 14, Jobs: {'z', 't', 'o'}
Worker: 16, Jobs: {'g', 'r', 'e', 't'}
Worker: 2, Jobs: {'b', 'a', 'r', 't'}
Worker: 3, Jobs: {'x'}
```