

מטלה 3 אלגוריתמים כלכליים

ת"ז : 208957084

מגיש : רוי ווסקר

<https://github.com/roywasker/EX3>

שאלה 1

הסבר על הקוד-

בשלב הראשון נחשב כמה משתתפים יש לנו וכמה משאבים יש לנו. נצהיר על 2 רשימות חדשות אחת לכול המשתתפים (כמות המשאבים * כמות המשתתפים) ו סכום התועלות לכול שחקן.

```
# Declare the variables and utility
num_of_players = len(valuations)
num_of_resources = len(valuations[0])
variables = []
utility_for_player = []
```

לאחר מכן נעבור בלולאה על כל השחקנים וכול המשאבים ונחשב להם את סכום התועלות שלהם ונשמור את זה ברשימת התועלות של כול השחקנים.

```
# Calculation of the utility for all player
for i in range(num_of_players):
    utility = 0
    for j in range(num_of_resources):
        variables.append(cvxpy.Variable(num_of_players)) # fra
        utility += variables[j][i] * valuations[i][j] # Calcul
    utility_for_player.append(utility) # insert utility for p
```

לאחר מכן נצהיר על משתנה שיכיל את התועלת המינימלית , ונבנה רשימה של כול האילוצים שיש לנו שהם : הערך של המשתנים יהיה בין 0 ל 1 כולל שכול תועלת עבור משתתף תהיה גדולה או שווה לתועלת המינימלית ושגם סכום הערכים של המשתנים שמיצגים אותו משאב יהיה שווה ל 1 לא נרצה שיהיה מעל או מתחת לזה.

```
min_utility = cvxpy.Variable()

# list all the constraints for the maximize function
fixed_constraints = \
    [variables[i][j] >= 0 for i in range(num_of_resources) for j in range(num_of_players)] + \
    [variables[i][j] <= 1 for i in range(num_of_resources) for j in range(num_of_players)] + \
    [utility_for_player[i] >= min_utility for i in range(num_of_players)] + \
    [sum(variables[i]) == 1 for i in range(num_of_resources)]
```

נגדיר את הבעיה שאנחנו רוצים למקסם את התועלת המינימלית ונעביר את רשימת האילוצים ונפתור את הבעיה.

```
prob = cvxpy.Problem(cvxpy.Maximize(min_utility), constraints=fixed_constraints)
prob.solve(solver=cvxpy.ECOS)
```

בשלב האחרון נדפיס את התוצאות שקיבלנו בפורמט הרצוי.

```
for i in range(num_of_players):
    print(f"player {i} receives ", end=" ")
    for j in range(num_of_resources):
        if j == 0:
            print(f"{abs(round(variables[j][i].value * 100, 2))}% of resource {j}", end="")
        else:
            print(f" and {abs(round(variables[j][i].value * 100, 2))}% of resource {j}", end="")
    print()
```