אלגוריתמים כלכליים סיכום

הגדרה אלגוריתמים כלכליים – אלגוריתמים ***שהקלט*** שלהם הוא משאבים והעדפות של בני-אדם, ***והפלט*** הוא חלוקה של משאבים בהתאם להעדפות.

תכונות רצויות של החלוקה: ***הוגנות***, ***יעילות***

נלמד אלגוריתמים לחלוקה הוגנת ויעילה של:

1 ***משאבים רציפים*** (קרקעות, סחורות, משאבי מיחשוב)

2 ***חפצים בדידים*** (מקומות בקורסים, חדרים, תיקים בממשלה)

3 ***תקציב*** (של מדינות, עיריות, תרומות לארגונים)

הגדרות-

פרופורציונלי- $For\ all\ i,\ j\colon\ Vi\ (Xi\ ) \geq Vi\ (C)\ /\ n$

קנאה- (זכויות שוות) כל שחקן המשחק לפי הכללים מקבל חתיכה טובה לפחות כמו שתי האחרות.

$For\ all\ i,\ j\colon\ Vi\ (Xi\ ) \geq Vi\ (Xj)$

קנאה- (זכויות שונות)- רמת הקנאה המוצדקת בין שני משתתפים i, j עם זכויות w_j, w_i היא:

$V_i(X_j)\backslash w_j - V_i(X_i)\backslash w_i$

WEF- חלוקה ללא קנאה מוצדקת = רמת הקנאה המוצדקת היא 0)לכל היותר(.

הוגנות- ללא קנאה

שיפור פארטו - נקרא שיפור, אם הוא טוב יותר לחלק מהמשתתפים, וטוב לפחות באותה מידה לכל השאר.

יעילות פארטו - אין שיפור פראטו, תנאי הכרחי לבחירה שהיא "נכונה" מנקודת-מבט כלכלית.

שופיר פארטו בחלוקה עם כסף - חלוקה א היא שיפור פארטו של חלוקה ב אם: • התועלת של כל השחקנים בחלוקה א גדולה לפחות כמו בחלוקה ב; •סכום הכסף שמשלמים השחקנים בחלוקה א גדול לפחות כמו בחלוקה ב (- המנהל לא הפסיד) •התועלת של חלק מהשחקנים בחלוקה א גדולה יותר מבחלוקה ב (- מישהו הרויח). **יעילות פארטו = יעילות אוטיליטרית**

יעילה-אוטיליטרית - חלוקה הממקסמת את סכום הערכים של השחקנים $\max\limits_{x} \sum\limits_{j=1}^{n} V_j(X_j)$

אגליטרית - חלוקה הממקסמת את הערך הקטן ביותר $\max\limits_{x} \min\limits_{i} \sum\limits_{j=1}^{n} V_j(X_j)$

עקביות- אלגוריתם לחלוקת-מושבים נקרא עקבי אם עבור כל תת-קבוצה X של מפלגות, שקיבלו ביחד n מושבים בחלוקה הכללית – אם נשתמש באותו אלגוריתם כדי לחלק את n המושבים בין המפלגות בקבוצה X בלבד, נקבל אותה חלוקה בדיוק כמו בחלוקה הכללית.

EF1 - חלוקה נקראת "ללא קנאה מלבד 1" אם לכל שני משתתפים א,ב, קיים חפץ כלשהו, שאם נוריד מהסל של ב, אז שחקן א לא יקנא בו.

אלגוריתם מדוייק- איכות הפתרון **תמיד מיטבי** זמן ריצה **גרוע.**

אלגוריתם קרוב - איכות הפתרון **לא מיטבי** זמן ריצה **פולינומיאלי.**

חלוקה מסודרת - יחסי-הערכים של החפצים בסל של שחקן ב קטנים או שווים ליחסי-הערכים של החפצים בסל של שחקן א.

שיפור פארטו חלש- חלוקה ב היא שיפור פארטו חלש של חלוקה א, אם הערך שמקבל כל שחקן בחלוקה ב גדול לפחות כמו הערך שהוא מקבל בחלוקה א.

ערך- מספר המתאר עד כמה השחקן רוצה סל מסויים של חפצים.

תועלת- מספר המתאר עד כמה השחקן רוצה סל מסויים הכולל חפצים וכסף.

פונקציית תועלת קוואזיליניארית- התועלת של סל כלשהו, הכולל קבוצה X של חפצים וסכום-כסף p היא: v(X) + p כאשר v היא פונקציית ערך כלשהי. תועלת = ערך + כסף = ערך - תשלום

מחיר גבוה מדי- מחיר גבוה מדי הוא מחיר כלשהו T, כך שאם המחיר של חדר כלשהו ≤T, והמחיר של חדר אחר כלשהו ≥0, אז אף שחקן לא בוחר בחדר עם מחיר ≤T

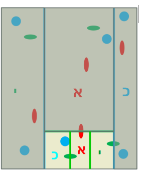
מגלה-אמת- האסטרטגיה הטובה ביותר של כל משתתף היא להגיד את הערך האמיתי שלו, לא משנה מה עושים האחרים.
מגלה-אמת תקצוב באלגו' השתתפותי נקרא - אם לכל אזרח i, כאשר הבחירות של שאר האזרחים קבועות, התועלת של i גדולה ביותר כאשר ה-Ui,j אמיתיים.

הוגנות לקבוצות- לכל קבוצה בגודל k, הסכום הכולל המועבר לנושאים שאחד מחברי-הקבוצה תומך בהם לפחות הוא k*C/n
תקציב פריק - תקציב (d1 ,…,dm) נקרא פריק אם קיימים סכומים d i,j לכל אזרח i ולכל נושא j כך ש: לכל נושא $\sum_{i=1}^{m} d_{i,j} = d_j$, לכל אזרח $\sum_{j=1}^{m} d_{i,j} = C\backslash n$ וגם $d_{i,j} > 0 \text{ only if } U_{ij} > 0$.
תקציב הוגן לקבוצות-כאשר האזרחים מחולקים לקבוצות וכל קבוצה j נותנת 100% מהתקציב לסעיף j, האלגוריתם מחלק את התקציב בין הסעיפים ביחס ישר לגדלי הקבוצות.

נושאים משאבים רציף:
בעיית חלוקת קרקעות לאנשים / חלוקת עוגה ל-n אנשים - העדפות שונות, זכויות שוות.
חתוך ובחר: אלגו': אם אחד חתוך לחצי בעיניו והאם השני החליט מה הוא מעדיף
תכונות: פרופורציונלי , ללא קנאה. חסרון: רק ל2 אנשים, לא יעילי פרטו

המפחית האחרון: אלגו': ● עמי מסמן 1/n בעיניו. ● אם תמי חושבת שזה יותר מדי - היא מפחיתה ל-1/n. וכן רמי וכו'.● האחרון שהפחית מקבל את החלק שסימן. ● ●ממשיכים ברקורסיה. תכונות: פרופורציונלי (באינדוקציה). חסרון: לא יעיל $O(n^2)$

אלגוריתם אבן-פז : אלגו': • כל שחקן מחלק לשני חלקים בשווה 1/2 בעיניו. •חותכים את העוגה בחציון של הקוים. • שולחים כל שחקן לחצי שמכיל את הקו שלו. •מחלקים כל חצי ברקורסיה. (אם יש n זוגי, חותכים את העוגה כך שבצד אחד יהיו $(n+1)\backslash 2$ קוים ובצד שני $(n-1)\backslash 2$ קוים.)
תכונות:פרופורציונלי (באינדוקציה), יעיל $O(n\log(n))$ (הכי יעיל שאפשר). חסרון: קנאה.



Selfridge Conway: אלגו':• כ חותך 3 חתיכות שוות בעיניו. • אם א, י מעדיפים חתיכות שונות – סיימנו. אחרת - • י מקצץ את החתיכה הטובה ביותר ומשווה לשניה בעיניו. א, י, כ בוחרים חתיכה. י חייב לבחור את זו שקיצץ, אם לא נבחרה קודם. • קיבלנו חלוקה ללא קנאה, אבל עם שארית.•[א או י בחרו את החתיכה המקוצצת; במקרה זה [א. י• שלא בחר את החתיכה המקוצצת, מחלק את השארית לשלוש חתיכות שוות בעיניו. א כ י בוחרים חתיכה.
תכונות: ללא קנאה (שידוך מושלם), חסרון: רק ל3 אנשים.

- חלוקת משאבים הומוגניים, לדוגמא בזל נפט וכו' מובן שמדובר בבעיה יותר קלה אבל פה נתמקד חלוקה יעילה פראטו (רציף)

דיקטטורה - אדם אחד לוקח הכל תכונות: יעיל פראטו חסרון: קנאה
חלוקה אוטיליטרית- אלגו': הגדר משתנה z המייצג את הערך הגדול ביותר. פתור את בעיית האופטימיזציה הבאה: maximize z;
תכונות: יעיל פראטו חסרון: קנאה
משפט: כל חלוקה יעילה-אוטיליטרית (ממקסמת סכום ערכים) היא יעילה פארטו.
הוכחה: נתונה חלוקה א הממקסמת סכום ערכים. נניח בשלילה שהחלוקה לא יעילה פארטו. ● אז קיימת חלוקה ב שהיא שיפור-פארטו שלה. ● בחלוקה ב, לכל השחקנים יש ערך לפחות כמו ● בחלוקה א, ולחלק מהשחקנים יש ערך גבוה יותר. לכן בחלוקה ב סכום הערכים גבוה יותר – בסתירה ● לכך שחלוקה א

ממקסמת את סכום הערכים. ***

__אוטיליטרי- code__

```
print("80 19 1")
print("79 1 20")
xw, xo, xs = cvxpy.Variable(3)# XS מהפלדה,XO מהנפט,XW שברים כמה עמי מקבל מהעצים
utility_ami  = xw*80 + xo*19 + xs*1
utility_tami = (1-xw)*79 + (1-xo)*1 + (1-xs)*20
print("\nUtilitarian division - maximum sum of utilities:")
prob = cvxpy.Problem(
    cvxpy.Maximize(utility_ami + utility_tami),
    constraints = [0 <= xw,xw <= 1,0 <= xo,xo <= 1,0 <= xs,xs <= 1])
prob.solve()
print("status:", prob.status)
print("optimal value: ", prob.value)
print("Fractions given to Ami: ", xw.value, xo.value, xs.value)
```

חלוקה אגליטרית- __אלגו':__ הגדר משתנה z המייצג את הערך הקטן ביותר. פתרו את בעיית האופטימיזציה הבאה maximize z; subject to Vi(Xi) ≥ z for all i in 1,…,n

__חסרון__: לא יעיל פרטו אם $\exists i \rightarrow Xi = 0$ קיים שחקן נתן ערך 0 לאחד המשאבים.

לקסימין-אגליטרית- חלוקה הממקסמת את וקטור הערכים המסודר מהקטן לגדול, לפי סדר מילוני. ממקסמת את הערך הקטן ביותר; בכפוף לזה, את הערך השני הכי קטן; בכפוף לזה, את הערך השלישי הכי קטן; וכו'.

__אלגו':__ ● מצא חלוקה שבה הערך המינימלי גדול ביותר (חלוקה אגליטרית). סמן ערך זה באות 1z. ● מבין כל החלוקות שבהן הערך המינימלי הוא 1z, מצא חלוקה שבה סכום שני הערכים הקטנים גדול ביותר. סמן ב: 2z + 1z ● מבין כל החלוקות עם ערך מינימלי 1z, וסכום שני ערכים מינימליים 2z + 1z, מצא חלוקה שבה סכום שלושת הערכים הקטנים גדול ביותר. ● המשך באותו אופן n פעמים.

__תכונות__: יעיל פרטו

__יעיל פרטו הוכחה:__ ● נתונה חלוקה לקסימין-אגליטרית א. נניח בשלילה שקיים לה שיפור-פארטו - חלוקה ב. ● בחלוקה ב, לכל השחקנים יש ערך לפחות כמו ב-א, ולחלק מהשחקנים יש ערך גדול יותר. ● לכן וקטור-הערכים המסודר בחלוקה ב גדול יותר, בסדר מילוני, מבחלוקה א – סתירה להנחה שחלוקה א היא לקסימין-אגליטרית. ***

__דוגמת הרצה:__ סיבוב 1: מקסימום ערך קטן ביותר = 3. סיבוב 2: מקסימום סכום שני ערכים קטנים ביותר = 3+4 =7. סיבוב 3: מקסימום סכום שלושה ערכים קטנים ביותר = 3+4+5 = 12. סיבוב 4: מקסימום סכום ארבעה ערכים קטנים ביותר 3+4+5+5 = 17.

	עצים	נפט	פלדה
א:	4	0	0
ב:	0	3	0
ג:	5	5	10
ד:	5	5	10

__אגילטרי לקסימין- code__

```python
num_of_players = 4
xw = cvxpy.Variable(num_of_players)  # fractions of wood
xo = cvxpy.Variable(num_of_players)  # fractions of oil
xs = cvxpy.Variable(num_of_players)  # fractions of steel
A=0,B=1,C=2,D=3
utilities = [
    xw[A]*4,
    xo[B]*3,
    xw[C]*5+xo[C]*5+xs[C]*10,
    xw[D]*5+xo[D]*5+xs[D]*10]
fixed_constraints = \
    [0<=t for t in xw] + [t<=1 for t in xw] + \
    [0<=t for t in xo] + [t<=1 for t in xo] + \
    [0<=t for t in xs] + [t<=1 for t in xs] + \
    [sum(xw)==1, sum(xo)==1, sum(xs)==1]

print("\nITERATION 1: Egalitarian division")
min_utility = cvxpy.Variable()    # smallest utility of a single agent
prob = cvxpy.Problem(
    cvxpy.Maximize(min_utility),
    constraints = fixed_constraints + [min_utility <= u for u in utilities])
prob.solve()
print("optimal value: ", prob.value)
print("Utilities: ", [u.value for u in utilities])
print(f"  Wood: {xw.value.round(2)},\n  oil: {xo.value.round(2)},\n  steel: {xs.value.round(2)}")
# min_utility = 3

print("\nITERATION 2: Max the smallest sum of two:")
min_utility_of_two = cvxpy.Variable()
prob = cvxpy.Problem(
    cvxpy.Maximize(min_utility_of_two),
    constraints = fixed_constraints +
        [min_utility.value <= u for u in utilities] +
        [min_utility_of_two  <= u+v for u,v in combinations(utilities,2)]
    )
prob.solve()
print("optimal value: ", prob.value)
print("Utilities: ", [u.value for u in utilities])
print(f"  Wood: {xw.value.round(2)},\n  oil: {xo.value.round(2)},\n  steel: {xs.value.round(2)}")
# min_utility_of_two = 7

print("\nITERATION 3: Max the smallest sum of three")#וממשיכים האותה דרך ככמות השחקנים
```

משפט: אם הערכים של השחקנים מנורמלים, כך שכל השחקנים מייחסים את אותו ערך לעוגה כולה, אז כל חלוקה אגליטרית (לקסימין או לא) היא פרופורציונלית.
הוכחה: ● קיימת חלוקה פרופורציונלית, למשל חלוקה שבה כל שחקן מקבל 1 חלקי n מכל משאב. ●יהי V ערך העוגה כולה (בעיני כולם). בחלוקה פרופ.. הערך הקטן ביותר הוא לפחות V חלקי n. ●לכן, בחלוקה הממקסמת את הערך הקטן ביותר, הערך הקטן ביותר הוא לפחות V חלקי n. ●לכן, חלוקה זו גם היא פרופורציונלית. ***
משפט: כל חלוקה מסודרת בין שני שחקנים אא"מ יעילה־פארטו.
משפט: לפעמים אין חלוקה אגליטרית וללא-קנאה
משפט: כל חלוקה הממקסמת סכום של פונקציה עולה כלשהי של הערכים, היא יעילה פארטו.
הוכחה: ● נתונה חלוקה א הממקסמת סכום זה. ● נניח בשלילה שהחלוקה לא יעילה פארטו. ●אז קיימת חלוקה ב שהיא שיפור-פארטו שלה. ● בחלוקה ב, לכל השחקנים יש ערך לפחות כמו בחלוקה א, ולחלק מהשחקנים יש ערך גבוה יותר. ● כיוון שהפונקציה עולה בחלוקה ב הסכום גבוה יותר – סתירה לכך שחלוקה א ממקסמת את הסכום.
הכללה: נמצא חלוקה הממקסמת את הסכום של פונקציה עולה של הערכים: $\max_x f(V_j(X_j))$. אם הפונקציה f היא לוגריתמית $f(V) = \log V$ אז החלוקה לא רק יעילה אלא גם ללא קנאה!

יעילות נאש- מצב יעיל-נאש הוא מצב הממקסם את סכום הלוגריתמים של הערכים $(f = \log)$.
תכונות: ללא קנאה, יעילה פראטו.

---—

נושאים *משאבים בדידים*:
פתרונות מקובלים: 1. קירוב (חלוקת מושבים בכנסת) 2. שיתוף (חלוקת תיקים בממשלה) 3. מיטוב (השמת עבודות בתעשיה) 4. כסף (חלוקת חדרים ושכר-דירה) 5.הגרלה ("מחיר למשתכן").

m חפצים **זהים**, n זכויות **שוות-**
חלוקה הוגנת בקירוב- *אלגו':* כל שחקן מקבל m\n מעוגל למטה או למעלה.

m חפצים **זהים**, n זכויות **שונות -**
בעית חלוקה מושבים (כנסת), איך לחלק את m המושבים בכנסת בין n המפלגות, באופן יחסי למספר קולותיהן?
בעיה בקירוב: אי אפשר לעגל כי יכול לצאת מצב של מושב אחד או יותר שלא נבחר.
המלטון- *אלגו'-* ● נותנים לכל מפלגה את מספר-המושבים המדוייק שלה מעוגל כלפי מטה. ●מחלקים את המושבים העודפים לפי סדר יורד של השארית.
דוגמא- m=100 n =100
א: 69.4 ב: 30.35 ג: 20.25
א: 69 ב: 30 ג: 20
א: ,70 ב: ,30 ג: 20 החלוקה הסופות השארית הגדולה של א
תכונות: בעיה בקירוב, הוגנת חסרון-לא עקבית

ג'פרסון- אלגו'- ● אתחול: כל מפלגה מקבלת 0 ● כל עוד יש מושבים: ● מחשבים, לכל מפלגה $\frac{\text{מספר קולות}}{\text{מספר מושבים שהמפלגה קבלה עד כו} + 1}$ ● נותנים את המושב הבא למפלגה שהמנה שלה גדולה ביותר.
תכונות: בעיה בקירוב, עקבי חסרון-קנאה
דוגמא: 5 מושבים, 500 בוחרים. א: 40, ב: 135, ג:325.
חלוקה: 0,0,0 מנה: 325,135,40.
חלוקה: 1,0,0 מנה: 162.5,135,40.

חלוקה: 2,0,0 מנה: 108.33,135,40.
חלוקה: 2,1,0 מנה: 108.33,67.5,40.
חלוקה: 3,1,0

הכליל ג'פרסון (שיטות מחלק)- <u>אלגו'-</u> • אתחול: כל מפלגה מקבלת 0 • כל עוד יש מושבים: •מחשבים, לכל מפלגה $\frac{\text{מספר קולות}}{F(\text{מספר מושבים שהמפלגה קבלה עד כו})}$ כאשר •נותנים את המושב הבא למפלגה שהמנה שלה גדולה ביותר.
<u>תכונות</u>: בעיה בקירוב.
עם מחלק בשיטת $F(s) = s + y$
•שיטת ג'פרסון $F(s) = s + 1$ •וובסטר שיטת $F(s) = s + 0.5$ •אדאמס שיטת $F(s) = s$
•אם y<0.5, יש הטיה לטובת המפלגה הקטנה. •אם y>0.5, יש הטיה לטובת המפלגה הגדולה. •אם y=0.5, אין הטיה לאף צד.
<u>דוגמא:</u>

3 מושבים, 300 בוחרים. א: 210, ב: 50, ג: 40

- שיטת אדאמס: 1 1 1
- שיטת וובסטר: 2 1 0
- שיטת ג'פרסון: 3 0 0

<u>מסכנה</u>- שיטת וובסטר ללא כל הטיה לטובת מפלגות גדולות או קטנות.
<u>*משפט*</u>- שיטת וובסטר היא השיטה היחידה לחלוקת מושבים, שהיא גם עקבית וגם הוגנת

חיפוש במרחב המצבים- <u>אלגו':</u> •נתחיל מחלוקה ריקה; •ניצור את כל n המצבים הנובעים ממצב קיים + חלוקת חפץ אחד; •נמחקן מצבים מיותרים (גיזום – pruning); •מתוך כל המצבים הסופיים (= m חפצים חולקו), נבחר מצב עם הערך המינימלי הגדול ביותר.
*גיזום: א. נמחק מצבים זהים. ב. נמחק כל מצב, שהחסם האופטימי שלו אינו טוב יותר מהחסם הפסימי הטוב ביותר שמצאנו.
*חסם פסימי = התוצאה המיטבית לא תהיה **גרועה** יותר. דוגמה: חלק את החפצים שנשארו באקראי.
*חסם אופטימי = התוצאה המיטבית לא תהיה **טובה** יותר. דוגמה: תן כל החפצים שנשארו לכולם.
<u>תכונות</u>: אלגו' מדויק.
<u>דוגמה:</u>

	חפץ א	חפץ ב	חפץ ג
שחקן 1	11	11	55
שחקן 2	22	22	33
שחקן 3	33	44	0

מצב התחלתי: (0,0,0; 0)
נתינת חפץ א: (11,0,0; 1), (0,22,0; 1), (0,0,33; 1).
נתינת חפץ ב: (22,0,0; 2), (11,22,0; 2), (11,0,44; 2)
(11,22,0; 2), (0,44,0; 2), (0,22,44; 2)
(11,0,33; 2), (0,22,33; 2), (0,0,77; 2).
נתינת חפץ ג: 27 מצבים. באופן כללי: n^m.

המצב (0,0,0; 0):
- חסם פסימי: **11** (א:1, ג:2, ב:3).
- חסם אופטימי: **77** (1:א+ב+ג, 2:א+ב+ג, 3:א+ב+ג).

המצב (22,0,0; 2):
- חסם אופטימי: **0** (נותנים את חפץ ג לכולם).
- אפשר לגזום את המצב הזה!

בעיית שיבוץ העבודות - צריך לבצע m עבודות-חישוב באורכים שונים. יש n מחשבים זהים. צריך לשבץ עבודות למחשבים כך שזמן הסיום של העבודה האחרונה יהיה קצר ביותר.<u>תכונות</u>: אלגו' קירוב.

<u>m חפצים **שונים**, n זכויות **שוות**-</u>

אלגוריתם הֶסֶבב- <u>אלגו'-</u> ●מסדרים את השחקנים בסדר שרירותי כלשהו. ●כל שחקן לוקח, מבין החפצים שנשארו, את החפץ שהוא הכי רוצה. ●אם נשארו חפצים – חוזרים לשלב הקודם.

<u>תכונות</u>: בעיה בקירוב.

<u>משפט-</u> אלגוריתם הסבב מחזיר חלוקה 1EF.

<u>הוכחה-</u> ● נניח בה"כ ששחקן א מופיע בסבב לפני שחקן ב. ●א לא מקנא כלל: על כל חפץ ש-ב בחר, א בחר לפניו.●עכשיו נניח שמורידים מהסל של א את החפץ הראשון שבחר. על כל חפץ שנשאר בסל של א, ב בחר לפניו. לכן החלוקה 1EF גם עבור שחקן ב. ***

אלגוריתם ֶסֶבב משוקלל-: <u>אלגו'-</u> ●אתחול כל שחקן מקבל 0. ●כל עוד יש חפצים: ●מחשבים לכל שחקן $\frac{\text{הזכות שלו}}{F(\text{מספר חפצים שקבלה עד כה})}$. ●השחקן, שהמנה שלו גדולה ביותר, בוחר, מבין החפצים שנשארו, את החפץ שהוא הכי רוצה. <u>תכונות</u>: בעיה בקירוב. <u>חסרון-</u> קנאה

<u>משפט-</u> אלגוריתם הסבב המשוקלל עם פונקציית-מחלק $F(s) = s + y$ מחזיר חלוקה שבה לכל שני משתתפים i, j עם זכויות wj, wi, רמת הקנאה המשוקללת היא לכל היותר:

$$y \cdot V_i(g)\backslash w_i + (1 - y) \cdot V_i(g)\backslash w_j$$

- **דוגמה**: לשחקן i זכות 1, לשחקן j זכות 2. רמת הקנאה המוצדקת של שחקן i בשחקן j היא לכל היותר:
- y*Vi(g)/1 + (1-y)*Vi(g)/2 =(y+(1-y)/2)* Vi(g)
  - y=0: Vi(g)/2 - פחות קנאה – טוב יותר לשחקן
  - y=1: Vi(g) - יותר קנאה – טוב יותר לשחקן השני
  - y=0.5: 3*Vi(g)/4 - ממוצע

<u>משפט-</u> שהמשאבים בדידים והערכות מנורמלות,**לא** כל חלוקה אגליטרית היא פרופורציונלית אבל חלוקה אגליטרית היא "הכי קרובה לפרופורציונלית" בהתחשב בחפצים הבדידים.

<u>משפט-</u> שהמשאבים **בדידים**, מציאת חלוקה אגליטרית היא בעיה NP-קשה. רידוקציה מ (Partition): "נתונים m מספרים חיוביים שסכומם 2S. האם ניתן לחלקם לשתי קבוצות שסכומן S?".

שיבוץ העבודות וחלוקה אגליטרית בעיית שיבוץ m עבודות על n מחשבים שקולה לבעיית חלוקה אגליטרית של m מטלות (=חפצים עם ערך שלילי) בין n אנשים עם הערכות זהות.

דוגמה: 4 אנשים, 9 מטלות, ערכים: -4, -4, -5, -5, -6, -6, -7, -7 חלוקה א: -5-6, -5-6, -4-7, -4-7, -4-4-7 ● ערך מינימלי: -15 חלוקה ב: -6-6, -5-7, -5-7, -4-4-4 ● ערך מינימלי: -12. חלוקה אגליטרית. חלוקה א היא קירוב 5/4 לחלוקה האגליטרית.

--—

שיבוץ רשימה- <u>אלגו':</u>
●לכל עבודה j בין 1 ל-m:
●תן את j למחשב עם זמן-סיום נוכחי קטן ביותר.
●לכל מטלה j בין 1 ל-m:
●תן את j לשחקן, שהעלות (=מינוס הערך) הנוכחית שלו קטנה ביותר (= קרובה ביותר לאפס)
* **לא** מסדרים את הערכים של העבודות בסדר יורד **בשונה** מהאלגוריתם החמדני.
<u>דוגמה-</u> 4 אנשים, 9 מטלות עם עלויות: 7 ,7 ,6 ,6 ,5 ,5 ,4 ,4 ,4.
שחקן א 4,5,7. שחקן ב 4,6. שחקן ג 4,6. שחקן ד 5,7 ● עלות מקסימלית: 16 (ערך מינימלי: מינוס 16).
<u>משפט-</u> אלגוריתם הרשימה לחלוקת מטלות מוצא חלוקה שבה העלות המקסימלית קטנה מפי 2 מהעלות המקסימלית המיטבית.

האלגוריתם החמדני- ●סדר את העבודות בסדר יורד של זמן הריצה;●הפעל "תיזמון רשימה" על הרשימה המסודרת●סדר את המטלות בסדר יורד של העלות שלהן;●חלק את המטלות בעזרת "אלגוריתם הרשימה".
<u>דוגמה-</u> 4 אנשים, 9 מטלות עם עלויות: 7 ,7 ,6 ,6 ,5 ,5 ,4 ,4 ,4.
שחקן א 7,4,4. שחקן ב 7,4. שחקן ג 6,5. שחקן ד 6,5 | עלות מקסימלית: 15 (ערך מינימלי: מינוס 15).
<u>משפט-</u> האלגוריתם החמדני מוצא חלוקת מטלות עם עלות מקסימלית קטנה מפי 4/3 מהעלות המיטבית.
<u>לסיכום-</u> אפשר להשתמש באותו אלגוריתם לחלוקת חפצים לשחקנים עם הערכות זהות: הערך המינימלי > פי 3/4 מהערך האגליטרי.לשחקנים עם הערכות שונות, הבעיה הרבה יותר קשה.

אלגוריתם מדויק + אלגוריתם קירוב <u>אלגו'-</u> ●משתמשים באלגוריתם מדוייק – חיפוש במרחב המצבים ●מחשבים חסמים פסימיים בעזרת אלגוריתם קירוב – כגון האלגוריתם החמדני שלמדנו.●אם נגמר הזמן, והחיפוש במרחב המצבים עדיין לא מצא חלוקה מיטבית – מחזירים את החלוקה הכי טובה שהחיפוש מצא עד כה. ●החסם הפסימי מבטיח, שהחלוקה הזאת טובה לפחות כמו החלוקה של אלגוריתם הקירוב

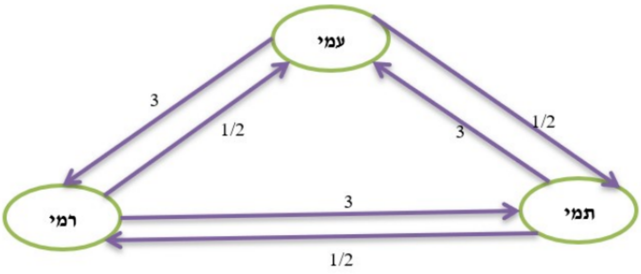
מיקסום מכפלת הערכים -החלוקה ה"אידיאלית" של חפצים בדידים היא חלוקה הממקסמת את מכפלת הערכים. <u>אלוג-</u> חיפוש במרחב המצבים, עם כללי גיזום כמו שלמדנו, רק שבמקום לבדוק את התועלת הקטנה ביותר בכל מצב, בודקים את מכפלת התועלות בכל מצב.
<u>תכונות</u>: יעילות פארטו,1EF

"חותכים" חפץ בדיד
איך "חותכים" חפץ בדיד - החפץ שצריך "לחתוך" נשאר בבעלות משותפת (כמו משמורת על ילדים).
<u>המטרה-</u> למצוא חלוקה הוגנת ויעילה עם **הכי מעט** שיתופים שאפשר.

אלגוריתם "המנצח המתוקן"- <u>אלגו'-</u> ●סדר חפצים בסדר עולה של יחס הערכים:
ערך-עבור-שחקן-א\ערך-עבור-שחקן-ב. ●אתחול: תן את כל החפצים לשחקן א. ●העבר חפצים לשחקן ב לפי הסדר, עד ש:(1)סכום הערכים של א שווה לסכום של ב, או -(2) יש חפץ אחד שאם "נחתוך" אותו הסכום ישתווה.
<u>תכונות</u>: חלוקה מסודרת (כלומר יעילה פראטו),פרופורציונלית, וללא קנאה, שיתוף של חפץ **אחד לכל היותר.**
<u>חסרון-</u> מתאים רק ל2 אנשים
<u>דוגמא-</u> דונלד ואיוונה יש במטלה

בעיית 3 שחקנים ומעלה-
<u>משפט-</u> לכל n שחקנים, קיימת חלוקה יעילה- פארטו ופרופורציונלית עם n-1 שיתופים כל היותר.

גרף ההחלפות - גרף־ההחלפות של חלוקה נתונה הוא גרף מכוון שלם, עם ●n צמתים – צומת לכל שחקן.
●קשת מכוונת בין כל שני שחקנים j,i. ●משקל הקשת $j \to i$ = היחס (ערך של i \ ערך של j) הקטן ביותר של חפץ הנמצא בסל של שחקן i.
<u>דוגמא-</u> עמי-אוהל, תמי-דירה, רמי-מחסן.



	אוהל	דירה	מחסן
עמי:	**3**	1	6
תמי:	6	**3**	1
רמי:	1	6	**3**

נחשב את משקלי הקשתות בדוגמה למעלה. כדי לחשב את משקל הקשת מעמי לתמי, יש לחשב את יחסי־הערכים של החפצים שנמצאים בסל של עמי. בחלוקה הנתונה, לעמי יש רק אוהל. יחס־הערכים של האוהל הוא 3/6. לכן משקל הקשת מעמי לתמי הוא 1/2. גם משקל הקשת מעמי לרמי נקבע לפי יחס־הערכים של האוהל, שהוא במקרה זה 3/1=..3 משקל הקשת מרמי לתמי נקבע לפי יחס הערכים של החפץ היחיד בסל של רמי, שהוא המחסן. יחס זה הוא 3/1=3.
לכל מעגל מכוון בגרף, ניתן לחשב את מכפלת המשקלים על הקשתות במעגל. לדוגמה: מכפלת המשקלים של המעגל (עמי ← תמי ← רמי ← עמי) היא ⅛;
איך לפתור- כדי שלא להעמיס בסימונים, נראה את ההוכחה עבור מעגל באורך 3 הכולל את השחקנים: $a \to c \to b$ הקורא לא יתקשה להכליל את ההוכחה למעגל בכל אורך שהוא. נסמן את ההערכות של השחקנים במעגל ב: va, vb, vc, ואת החפצים שלפיהם נקבעו משקלי הקשתות ב:$x\ (\text{שחקן } a)\ y\ (\text{שחקן } b),\ z\ (\text{שחקן } c)$ לשחק. נסמן את מכפלת המשקלים במעגל באות P.
לפי ההנחה P<1., נסמן ב־Q את השורש השלישי של P, ונשים לב שגם Q<1. כעת, נגדיר החלפת חפצים בין השחקנים, באופן הבא:● שחקן a נותן לשחקן b חלק קטן כלשהו, שנסמן ב־εx, של חפץ x. $\varepsilon_y = \varepsilon_x \cdot q \cdot \frac{Vb(x)}{Vb(y)}$ ● שחקן b נותן לשחקן c חלק εy של חפץ y, המוגדר כך: $\varepsilon_z = \varepsilon_y \cdot q \cdot \frac{Vc(y)}{Vc(z)}$ ● שחקן c נותן לשחקן a חלק εz של חפץ z, המוגדר כך $\varepsilon_x = \varepsilon_z \cdot q \cdot \frac{Va(z)}{Va(x)}$
הגודל של εx יכול להיות כל מספר חיובי. נקבע אותו כך שההעברה תהיה אפשרית, כלומר: εx יהיה קטן מהכמות של x המוחזקת בידי שחקן a, εy יהיה קטן מהכמות של y המוחזקת בידי שחקן b, ו־ εzיהיה קטן מהכמות של z המוחזקת בידי שחקן c.

<u>משפט-</u> חלוקה היא יעילה־פארטו אם־ורק־אם בגרף־ההחלפות שלה, בכל מעגל מכוון, מכפלת־המשקלים גדולה או שווה 1.
חלוקה יעילה- מחפשים מעגל עם מכפלת-משקלים < 1? ● הופכים כל משקל ללוגריתם שלו; ●מחפשים מעגל עם סכום-משקלים שלילי (למשל, בעזרת אלגוריתם בלמן-פורד).

חלוקה הוגנת ויעילה ללא שיתופים - <u>אלגו-</u> ●מבצעים חיפוש במרחב המצבים; ●גוזמים מצבים המתאימים לחלוקות לא יעילות.

גרף־הצריכה- של חלוקה נתונה הוא גרף דו־צדדי לא־מכוון וללא משקלים, שבו: ●הקודקודים בצד אחד הם n השחקנים; ●הקודקודים בצד השני הם m החפצים; ●יש צלע בין שחקן i לבין חפץ j, אם ורק אם ● שחקן i מקבל חלק חיובי של חפץ j.

<u>משפט-</u> קיים אלגוריתם עם זמן-ריצה פולינומיאלי המוצא, לכל חלוקה נתונה, שיפור־פארטו־חלש עם גרף צריכה ללא מעגלים (← לכל היותר n+m-1 צלעות ← לכל היותר n-1 שיתופים).
<u>משפט-</u> קיים אלגוריתם עם זמן-ריצה פולינומיאלי המוצא, לכל חלוקה נתונה, שיפור־פארטו־חלש עם לכל היותר n-1 שיתופים.
חלוקה הוגנת ויעילה עם n-1 שיתופים- <u>אלגו'-</u> ●נמצא חלוקה פרופורציונלית ויעילה-פארטו (למשל: לקסימין-אגליטרית עם הערכות מנורמלות). ●נמצא שיפור-פארטו-חלש עם גרף-צריכה ללא מעגלים.
<u>תכונות</u>: יעילה-פארטו,פרופורציונלית,

חלוקה הוגנת ויעילה עם n-1 שיתופים – זכויות שונות
אם לשחקנים יש זכויות שונות, ניתן להשתמש באותו אלגוריתם, אבל להתחיל מחלוקה שהיא יעילה-פארטו ופרופורציונלית בהתחשב בזכויות השונות. <u>אלגו-</u> ●נמצא חלוקה פרופורציונלית ויעילה-פארטו (למשל: לקסימין-אגליטרית עם הערכות מנורמלות בהתאם לזכויות השונות) ●נמצא שיפור-פארטו-חלש עם גרף-צריכה ללא מעגלים.
פתרון לבעיית הרכבת הממשלה - ניתן להקים ממשלה עם n מפלגות, ולחלק את התיקים בהוגנות מדוייקת, בהתאם לגדלים השונים של המפלגות, כך שיהיו לכל היותר n-1 תיקים עם רוטציה.

——--------דף 2---------------------------------------

חלוקה הוגנת של חפצים בדידים וכסף (אין חובה לכמות החפצים שמקבל כל שחקן)
גוד או אגוד- <u>אלגו'-</u> ●שחקן א מציע מחיר כלשהו p. ●שחקן ב מחליט האם לקנות או לא: אם כן – ב משלם p/2 ל-א ומקבל את החפץ. אם לא – א משלם p/2 ל-ב ומקבל את החפץ.
<u>תכונות</u>: ללא קנאה,יעילה-פארטו <u>חסרון-</u> מתאים רק ל2 אנשים
<u>משפט-</u> "גוד או אגוד" מאפשר לכל שחקן קוואזיליניארי להשיג חלוקה פרופורציונלית.
<u>הוכחה-</u> שחקן א יכול להציע va=p. התועלת שלו: ●אם ב קונה: $p\backslash 2 = Va\backslash 2$ ●אם ב לא קונה: $Va - p\backslash 2 = Va\backslash 2$ ●שחקן ב יכול לקנות אם $p < Va$. ●קנה א: $Vb - p\backslash 2 > Vb\backslash 2$ ●אם לא קנה: $p\backslash 2 \geq Vb\backslash 2$ בשני המקרים, החלוקה פרופורציונלית. ***

<u>משפט-</u> כשכל השחקנים הם קוואזיליניאריים, חלוקה היא יעילה-פארטו **אם ורק אם** היא ממקסמת את סכום הערכים.

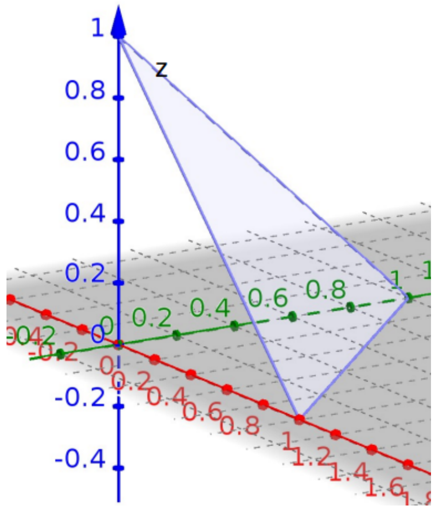
אלגוריתם המכרז השווה- <u>אלגו'-</u> ●כל שחקן רושם את ערך לכל חפץ. ●האלגוריתם מוכר כל חפץ לשחקן עם הערך הגבוה ביותר, בתמורה לערך שרשם. ●האלגוריתם מחלק את הכסף, שהתקבל מכל השחקנים, שווה בשווה.
<u>תכונות</u>: הרבה חפצים והרבה שחקנים,יעילה־פארטו,ללא-קנאה
<u>תועלת-</u> ●התועלת של כל שחקן i מהסל שלו היא: Vi(Xi) – Vi(Xi) + S/n = S/n ●התועלת של כל שחקן i מהסל של j היא: Vi(Xj) – Vj(Xj) + S/n.
<u>משפט-</u> אלגוריתם המכרז השווה מחזיר חלוקה יעילה־פארטו.
<u>הוכחה-</u> כל חפץ נמסר לשחקן המייחס לו ערך גבוה ביותר. לכן החלוקה ממקסמת סכום ערכים. לפי משפט קודם, החלוקה יעילה-פארטו. ***

חלוקה הוגנת של חפצים בדידים וכסף- כל שחקן מקבל לפחות חפץ אחד (דוג' בעיית חלוקת חדרים ושכר-דירה)

בעיית חלוקת שכר דירה נתונים- קלט- ●דירה עם n חדרים ודמי-שכירות נתונים R, ●קבוצה של n שותפים השוכרים את הדירה. הפלט ●השמה-לכל שחקן i מתאימים חדר אחד Xi. ●תמחור-לכל חדר j מתאימים מחיר j(p).

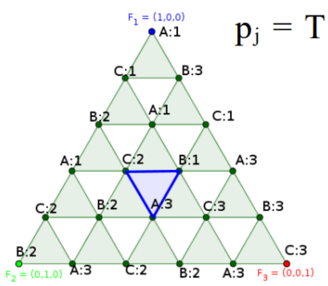
<u>האתגר:</u> להחליט מי יגור איפה, וכמה ישלם, כך שלא תהיה קנאה.
<u>ללא קנאה:</u> אף שותף לא מעדיף את החבילה (חדר+מחיר) של שותף אחר.
<u>הערה:</u> אם השחקנים קוואזיליניאריים, אז קיים מחיר גבוה מדי – למשל הערך הגבוה ביותר ששחקן כלשהו מייחס לחדר כלשהו.
<u>משפט:</u> אם קיים מחיר גבוה מדי, אז יש השמה +תימחור ללא קנאה.

סימפלקס התימחורים-
<u>סימפלקס</u> **=** אוסף הנקודות במרחב, שסכום הקואורדינטות שלהן שווה 1.
<u>לדוגמה:</u> סימפלקס במרחב 3- ממדי הוא משולש;

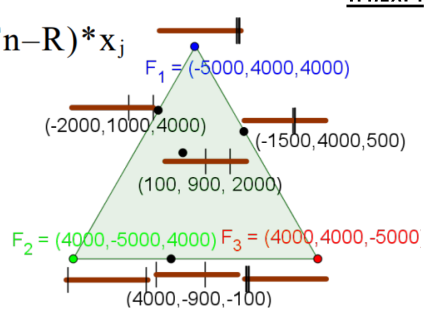


<u>משפט-</u> אם קיים מחיר גבוה מדי, אז יש השמה +תימחור ללא קנאה. (הוכחה עם סיפלקס)

סימפלקס התימחורים- הנחה: כל הדיירים הם קוואזיליניאריים. <u>אלגו'-</u> הקלט: מטריצה n x n המתארת את ערכי החדרים לכל אחד מהדיירים,הפלט: השמה X, תימחור p.אין קנאה: לכל שני שחקנים ≥ Vi(Xi) – p(Xi) Vi(Xj) – p(Xj. ●נחלק את סימפלקס התימחורים לסימפלקסונים; ●ניתן כל קודקוד לשחקן; נשאל אותו איזה חדר הוא מעדיף בתימחור המתאים לקודקוד.● נימצא סימפלקסון מגוון.
<u>תכונות:</u> תימחור ללא קנאה
<u>דוגמה:</u>



$$p_j = T - (Tn - R) * x_j$$

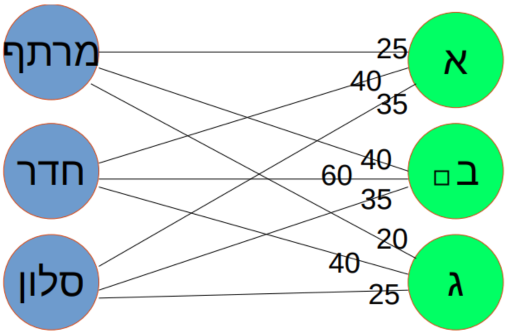


<u>משפט</u>- בכל השמה ללא קנאה, סכום הערכים של הדיירים בחדרים שהם גרים בהם הוא מקסימלי. (לפי הגדרת קנאה לדיירים קוואזיליניאריים, חדרים-השמת ללא קנאה X, השמה אחרת כלשהי Y, לכל i מקיים: $V_i(X_i) - P(X_i) \ge V_i(Y_i) - P(Y_i)$)

<u>משפט</u>- כל תימחור ללא קנאה יישאר ללא-קנאה לכל השמה ממקסמת-סכום-ערכים.

חלוקת-חדרים ללא קנאה- אלגו': ●מצא חלוקה כלשהי X הממקסמת סכום-ערכים. מציאת השמה הממקסמת את סכום הערכים = מציאת שידוך עם משקל מקסימום בגרף דו-צדדי למשל:"האלגוריתם ההונגרי" (יש מימוש בפייתון בספריה networkx). ● מצא תמחור p שאיתו החלוקה X ללא קנאה. ● <u>תכונות</u>: ללא קנאה

<u>דוגמא</u>: פלט: א←סלון ב←חדר, ג←מרתף



	מרתף	חדר	סלון
דייר א	25	40	**35**
דייר ב	40	**60**	35
דייר ג	**20**	40	25

<u>בעיה קביעת המחירים</u>: סכום המחירים יהיה שווה לשכר-הדירה:

<u>פתרון</u>: בעיית תיכנות ליניארי $For\ all\ i,\ j:\ w[d[i],\ i]\ -\ p[i]\ \ge w[d[i],\ j]\ -\ p[j]$ (אפשר לפתור בעזרת cvxpy)

<u>code-</u>

```
print("\n\nThere are three tennants and three rooms.")
# Construct an empty graph:
G=nx.Graph()
# Add edges with weights:
G.add_edge('aya','martef' ,weight=20)
G.add_edge('aya','heder',weight=40)
G.add_edge('aya','salon' ,weight=35)

G.add_edge('batya','martef' ,weight=40)
G.add_edge('batya','heder',weight=30)
G.add_edge('batya','salon' ,weight=35)

G.add_edge('gila','martef' ,weight=20)
G.add_edge('gila','heder',weight=40)
G.add_edge('gila','salon' ,weight=40)
print("Maximum-value matching: ", nx.max_weight_matching(G))
```

בעיית הטרמפיסט

<u>משפט</u>- ייתכן שבכל חלוקה ללא קנאה, אחד הדיירים ישלם מחיר שלילי (צריך לשלם לו שיסכים לגור בדירה...)

<u>הנחת הדיירים העניים</u>- כל דייר מעדיף חדר בחינם על-פני חדר בתשלום.

<u>משפט סו</u>- אם מתקיימת הנחת הדיירים העניים, אז קיימת חלוקת חדרים ללא קנאה, שבה כל דייר משלם מחיר חיובי (אין "טרמפיסטים").

אלגוריתמים מגלי-אמת

<u>בעיה בחירת פרסומות לדף רשת</u>: ●נתונים m מפרסמים שונים. לכל מפרסם יש ערך שונה להקלקה על הפרסומת שלו. ●בדף יש k מיקומים, m<k. לכל מיקום יש אחוזי-הקלקה שונים. ●צריך לבחור k מפרסמים ולתת מיקום לכל מפרסם, כך שתוחלת סכום הערכים תהיה גדולה ביותר.

<u>תכונות:</u> הערך של כל מפרסם ידוע רק למפרסם

מכרז ויקרי- <u>אלגו'-</u> ●המשתתפים כותבים הכרזות במעטפות;● המעטפות נפתחות ומסודרות בסדר יורד; ● בעל ההכרזה הגבוהה ביותר זוכה בחפץ; ● הזוכה משלם את ההכרזה השניה בגובהה.

תכונות: מגלה-אמת (כשלשחקנים יש העדפות קוואזי-ליניאריות). יעיל פארטו (ממקסם את סכום הערכים)

<u>משפט:</u> מכרז מחיר ראשון אינו מגלה-אמת.

ויקרי – קלארק – גרובס (VCG)- הנחות:● יש מספר סופי של תוצאות אפשריות. ● לכל משתתף יש ערך כספי לכל תוצאה. ● התועלת = ערך התוצאה פחות התשלום הקוואזי-ליניארית.

<u>אלגו':</u> ●בחר את התוצאה עם סכום-הערכים הגבוה ביותר. ●עבור כל שחקן: ●חשב את סכום הערכים של שאר השחקנים. ●חשב את סכום הערכים של שאר השחקנים אילו השחקן הנוכחי לא היה משתתף. ●גבה מהשחקן את ההפרש בין שני הסכומים.

תכונות: מגלה-אמת

<u>דוגמא1:</u>

r_1 = 0.1, r_2 = 0.05,
v_1 = 10, v_2 = 9, v_3=6.

המחיר למפרסם 1:

- 9*0.1 + 6*0.05 – סכום האחרים בלעדיו
- **-** 9*0.05 – סכום האחרים כשהוא נמצא
- = **7.5** * 0.1

המחיר למפרסם 2:

- 10*0.1 + 6*0.05 – סכום האחרים בלעדיו
- **-** 10*0.1 – סכום האחרים כשהוא נמצא
- = **6 *** 0.05

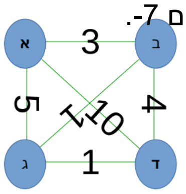
<u>דוגמא2:</u>

	א זוכה	ב זוכה	ג זוכה	אין זוכה
ערך משתתף א:	7	0	0	0
ערך משתתף ב:	0	8	0	0
ערך משתתף ג:	0	0	4	0
סכום:	7	**8**	4	
נבחר:	FALSE	**TRUE**	FALSE	
סכום בלי א:	0	**8**	4	
סכום בלי ב:	**7**	0	4	
סכום בלי ג:	7	**8**	0	
	תשלום	**ערך**	**תועלת**	
א:	0	0	0	
ב:	7	8	1	
ג:	0	0	0	
סה"כ:	7	8	1	

<u>דוגמא3:</u>

בעיית מציאת מסלול זול ביותר- נתונה רשת. לכל קשת יש עלות-מעבר. צריך להעביר חבילה בין שתי נקודות ברשת ד → א, במסלול עם עלות כוללת נמוכה ביותר.

צריך לפתור 1+6 בעיות מסלול-זול-ביותר.●כשכולם נמצאים: המסלול אבגד, הסכום 5- ● בלי אב: המסלול אגד, הסכום 6-. תשלום 4- ● בלי בג: המסלול אגד, הסכום 6-. תשלום 2- ● בלי גד: המסלול אבד, הסכום 7-. תשלום 3- ● בלי אג/אד/בד: אין שינוי, הסכום 5- תשלום 0. ● תשלום כולל 9-.



---------------------------------------—

תקצוב משתף

המטרה: תקצוב משתף הוגן.

סוגי אלגוריתמים לחלוקת תקציב:

1. **בדיד** – לכל פריט יש עלות; כל פריט ממומן במלואו או בכלל לא. * מתאים למימון פרויקטי בניה.
2. **רציף** – כל פריט יכול להשתמש בכל סכום שנותנים לו. * מתאים למימון ארגונים ועמותות.

תקצוב משתף בדיד

ייצוג הוגן לקבוצות אחידות- נתונים: ● גודל הוועדה = k. ● מספר האזרחים = n.

המיכסה- k/n = מספר האזרחים שמגיע להם להחליט לגבי מושב אחד.

קבוצה L-אחידה- קבוצת אזרחים בגודל לפחות L מיכסות, הבוחרים ביחד קבוצה זהה של L מועמדים. כמו הצבעה למפלגה. (תנאי קשה לקיום)

ייצוג הוגן לקבוצות אחידות - לכל קבוצה L-אחידה, הוועדה כוללת לפחות L מועמדים שחברי הקבוצה תומכים בהם.

קבוצה L-מגובשת - קבוצת אזרחים בגודל לפחות L מיכסות, התומכים בקבוצה כלשהי של L מועמדים.

ייצוג הוגן חזק - לכל קבוצה L-מגובשת, הוועדה כוללת לפחות L מועמדים שכל חברי הקבוצה תומכים בהם (ייצוג הוגן חזק לא תמיד אפשרי).

דוגמה שאין יצוג הוגן חזק - ● k=3, n=12, n/k=4 ● ההצבעות: אב, ב, בג, ג, ג, גד, ד, ד, דא, א, א. ● יש ארבע קבוצות -1מגובשות. (אב,ב,בג; בג,ג,ג,גד; גד,ד,ד,דא; דא,א,א,אב). ● ייצוג הוגן חזק מחייב לבחור ב,ג,ד,א – אבל k=3.

ייצוג הוגן מורחב - לכל קבוצה L-מגובשת, הוועדה כוללת לפחות L מועמדים שאחד מחברי הקבוצה תומך בהם. • לפחות אחד מחברי הקבוצה לא יצטרף לפרישה. (ייצוג הוגן מורחב תמיד אפשרי)

ייצוג הוגן יחסי - לכל קבוצה L-מגובשת, הוועדה כוללת לפחות L מועמדים, שכל אחד מהם נתמך ע"י חבר כלשהו מהקבוצה.

ייצוג הוגן חזק ← ייצוג הוגן מורחב ← ייצוג הוגן יחסי ← ייצוג הוגן לקבוצות אחידות.

שיטת התרמיל- ●מסדרים את הנושאים בסדר יורד של מספר הקולות שקיבלו. ●מכניסים נושאים לתקציב, עד שהעלות מגיעה לסכום הכולל בקופה.

חסרון: חוסר ייצוג הוגן לקבוצות

שיטת החלקים השוים- (בדיד) אלגו'- ●תן לכל אזרח "תקציב" וירטואלי בגודל k/n ●קבע את העלות של כל מועמד ל 1; ●אם יש לפחות מועמד אחד, שתומכיו יכולים לשלם את העלות שלו - בחר מועמד כזה, שהעלות לכל תומך שלו תהיה נמוכה ביותר, וחזור ל 1. ●אחרת, סיים את האלגוריתם; אם חסרים חברים בוועדה, הוסף חברים באופן שרירותי.

תכונות: ייצוג הוגן מורחב,הוגן. חסרון: שיטת החלקים השווים לא מונוטונית בגודל הוועדה

דוגמא: n=100 k=5. מתוכם 51 בוחרים א,ב,ג,ד,ה; 49 בוחרים צ,ק,ר,ש,ת.

- התקציב ההתחלתי לאזרח = 5/100. (לשם נוחות נכפיל ב100: תקציב התחלתי לאזרח=5, והעלות של מועמד=100.
- סיבוב 1: למועמדים א,ב,ג,ד,ה, דרוש 100/51~1.96 לכל תומך; למועמדים צ,ק,ר,ש,ת, רק 100/49~2.04 לכל תומך. לכן נבחר מועמד כלשהו מבין א,ב,ג,ד,ה, למשל מועמד א.
- כל אחד מ51- התומכים של מועמד א משלם 1.96, ונשאר עם 3.04.
- סיבוב 2: החישוב דומה, נבחר מועמד נוסף מבין ב,ג,ד,ה, למשל מועמד ב.
- כל אחד מ51- התומכים של מועמד א משלם 1.96, ונשאר עם 1.08.
- סיבוב 3: התקציב הכולל של תומכי ג,ד,ה הוא רק 55.08, ולכן אף אחד מהם לא נבחר.
- נבחר אחד מהמועמדים צ,ק,ר,ש,ת, נניח מועמד צ
- כל אחד מ49- התומכים של מועמד צ משלם 2.04 ונשאר עם 2.96.

- סיבוב 4: נבחר אחד מהמועמדים ק,ר,ש,ת, נניח מועמד ק.
- כל אחד מ49- התומכים של מועמד ק משלם 2.04 ונשאר עם 0.92.
- סיבוב :5 אף מועמד לא יכול להשיג מימון.
- מוסיפים עוד מועמד שרירותית כדי להשלים ל-5, ומסיימים את האלגוריתם.

מתקיים ייצוג הוגן חזק: יש שתי קבוצות 2-מגובשות, וכל אחת מהן קיבלה 2 מועמדים.

<u>משפט:</u> שיטת החלקים השוים בוחרת וועדה המקיימת ייצוג הוגן לקבוצות אחידות.
<u>הוכחה:</u> • התקציב ההתחלתי של כל אזרח הוא n/k. •לכן התקציב ההתחלתי של כל קבוצה L-אחידה הוא L. • כיוון שהקבוצה אחידה, חברי הקבוצה מממנים רק מועמדים שכל הקבוצה תומכת בהם. • התקציב שלהם מספיק כדי לממן לפחות L מועמדים.

שיטת פראגמן- (בדיד) <u>אלגו':</u> • תן לכל אזרח "תקציב" וירטואלי התחלתי 0. • קבע את העלות של כל מועמד ל 1. • הוסף לכל אזרח תקציב בקצב קבוע, עד שיש מועמד אחד, שהתומכים שלו יכולים לממן אותו. • ברגע שיש מועמד כזה, בחר אותו, והורד את היתרה של כל התומכים שלו לאפס. • אם נבחרו כבר k מועמדים - סיים את האלגוריתם. אחרת – חזור לצעד 1.
<u>תכונות:</u> לאפשר לוועדה לגדול באופן דינאמי,מונוטוניות,ייצוג הוגן יחסי <u>חסרון:</u> לא מבטיחה ייצוג הוגן מורחב. לא הוגן

<u>דוגמא:</u> n=100 k=5. מתוכם 51 בוחרים א,ב,ג,ד,ה; 49 בוחרים צ,ק,ר,ש,ת.
- התקציב ההתחלתי לאזרח = 0. לשם נוחות, העלות של מועמד = 100.
- נותנים לכולם כסף וירטואלי בהדרגה, עד שלכולם יש 1.96.
- 51 האזרחים הראשונים יכולים לממן מועמד, כי 51.*1.96=100 נניח שהם מממנים את א.
- המצב הנוכחי הוא: 51 אזרחים עם יתרה 0, 49 אזרחים עם יתרה 1.96.
- ממשיכים לתת כסף וירטואלי בהדרגה, עד שמוסיפים עוד 0.08.
- המצב הנוכחי הוא: 51 אזרחים עם יתרה 0.08, 49 אזרחים עם יתרה 2.04.
- באותה שיטה נבחר את צ,ב,ק,ג ....
- המצב הנוכחי הוא: 51 אזרחים עם יתרה 0; 49 אזרחים עם יתרה 1.8. האלגוריתם מסתיים.

בעיית בחירת ועדה לחלוקת תקציב
שיטת החלקים השוים לחלוקת תקציב- (בדיד) (הכלילה של החלקים השווים ושיטת פראגמן) <u>אלגו'-</u> • במקום המועמדים, יהיו הפריטים האפשריים בתקציב; • במקום עלות של 1 לכל מועמד, תהיה העלות האמיתית של כל פריט בשקלים • בשיטת החלקים השום, התקציב הוירטואלי ההתחלתי של כל אזרח יהיה B\n,כאשר B = התקציב הכולל.
<u>תכונות:</u> ייצוג הוגן מורחב
<u>ייצוג הוגן בחלוקת תקציב:</u>
<u>קבוצה T-מגובשת :</u>קבוצת אזרחים התומכת בכל הפריטים בקבוצה T, ומספר חברי הקבוצה הוא לפחות $cost(T) \cdot n\backslash B$.
<u>ייצוג הוגן מורחב בחלוקת תקציב:</u> לכל תת-קבוצה T של פריטים, ולכל קבוצה T-מגובשת, יש לפחות חבר אחד בקבוצה שתומך בלפחות |T| פריטים שנכנסו לתקציב.

תקצוב משתף רציף
השאלה לא איזה נושאים לתקצב אלא כמה לתת לכל נושא
<u>הגדרות-</u> כסף בקופה: C. נושאים m, אזרחים n.
התועלת של אזרח i לנושא j היא: Ui,j. **(רוצה או לא רוצה 0 \ 1).**
וקטור d המייצג תקציב: (d1 ,…,dm) כאשר d1 + … + dm=C.

התועלת של אזרח i מהתקציב d היא: $U_i(d) = \sum_{j=1}^{m} U_{i,j} \cdot d_j$.

תקציב אנארכי- אלגו': נותן לכל אזרח את חלקו בתקציב C\n, ואומר לו לחלק את הכסף כרצונו בין כל הנושאים שהוא תומך בהם.
תכונות: הוגן-לקבוצות, מגלה-אמת חסרון- לא-יעיל פרטו
משפט: כל תקציב אנארכי הוא הוגן-לקבוצות.
הוכחה- לכל קבוצה בגודל k, סכום הכסף הניתן לחברי-הקבוצה הוא k*C\n, וכל הסכום הזה מפוזר על נושאים שלפחות אחד מחברי-הקבוצה תומך בהם. ***

תקציב אוטיליטרי- תקציב הממקסם את סכום התועלות של האזרחים אלגו': ● תן את כל התקציב לנושאים שיש ● להם הכי הרבה תומכים.
תכונות: יעיל פראטו, מגלה-אמת חסרון- לא-הוגן אפילו ליחידים, דוגמא . אזרחים 1,2 בעד א, אזרח 3 בעד ב.

משפט: תקציב ָפ ִר יק אא"מ הוגן לקבוצות.
הוכחה: כיוון אחד ←נניח שהתקציב d הוא פריק. לכל קבוצה בגודל k, סכום הכסף הניתן לחברי-הקבוצה על-ידי הפירוק של d הוא k*C\n. לפי הגדרת הפירוק, כל הסכום הזה מפוזר רק על נושאים שלפחות אחד מחברי-הקבוצה תומך בהם. לכן התקציב הוגן לקבוצות. ***

נאש מיקסום המכפלה - אלגו'- קציב נאש הוא תקציב הממקסם את מכפלת התועלות של האזרחים:
$max[d]\ product[i]\ u_i(d)$ שקולה ל- $max[d]\ sum[i]\ log(u_i(d))$
תכונות: פריק, יעיל פראטו (מיקסום המכפלה), חסרון- אינו מגלה-אמת

code -

```
# There are 4 issues, denoted by a, b, c, d.
# The same letters denote the budget allocated to them.
allocations = cvxpy.Variable(4)
a, b, c, d = allocations

# There are 5 citizens. Their preferences are: ab, ac, ad, bc, a. The
total budget is 500 - 100 for each citizen
donations = [100, 100, 100, 100, 100]
utilities = [a+b, a+c, a+d, b+c, a]

sum_of_logs = cvxpy.sum([cvxpy.log(u) for u in utilities])
positivity_constraints = [v >= 0 for v in allocations]
sum_constraint = [cvxpy.sum(allocations)==sum(donations)]

problem = cvxpy.Problem(
    cvxpy.Maximize(sum_of_logs),
    constraints = positivity_constraints+sum_constraint)
problem.solve()
```

משפט- אלגוריתם נאש אינו מגלה-אמת.

הוכחה- נניח שיש ארבעה נושאים (א,ב,ג,ד,) חמישה אזרחים, והסכום הכולל 500. קלט: אב,אג,אד,בג,א; פלט (370, 65, 65 0). התועלת של אזרח "א+ב" היא 65+370=435. קלט: בד,אג,אד,בג,א; פלט (300, 200, 0, 0). התועלת של אזרח "א+ב" היא 200+300=500 -. שווה לו להגיד "ב+ד"! ***

אוטיליטרי-על-תנאי - ממקסם את סכום התועלות תחת האילוץ שהתקציב פריק. אלגו': ● כל אזרח תורם לנושאים, מאלה שהוא תומך בהם, עם הכי הרבה תומכים אחרים.
בדוגמא אב,אג,אד,בג,א: (400, 50, 50, 0). ● בדוגמא בד,אג,אד,בג,א: (300, 100, 50, 50)
תכונות: פריק, מגלה אמת (מיקסום המכפלה), חסרון- לא יעיל פרטו (אבל לא קיים שיפור-פארטו שהוא פריק)

טרילמה משפט- לא קיים אלגוריתם שהוא: יעיל-פארטו, וגם מגלה-אמת, וגם הוגן (ליחידים או לקבוצות).

מיזוג הצעות תקציב -
הגדרות- כסף בקופה: C. נושאים m, אזרחים n.
התועלת של אזרח i לנושא j היא: Ui,j.**(לא בינארי)**.
לכל אזרח i יש תקציב אידיאלי: $p_{i,1} + \ .... \ + p_{i,m} = C$
וקטור d המייצג תקציב: (d1 ,…,dm) כאשר d1 + … + dm=C.
התועלת של אזרח i מהתקציב d היא: $U_i(d) = \sum_{j=1}^{m} \left| d_j - p_{i,j} \right|$
הסבר: נניח שצריך להחליט רק על תקציב החינוך.כל אזרח i אומר מספר pi.
● אלגו' א: רוב. חסר משמעות; אולי לכל מספר יש תומך 1.
● אלגו' ב: ממוצע. הוגן לקבוצות, לא מגלה אמת, אפילו כשיש רק 2 אזרחים.
● אלגו' ג: קבוע שרירותי. לא יעיל פארטו.
● אלגו' ד: דיקטטור. לא אנונימי – מפלה בין אזרחים שונים אבל יעיל פרטו והוגן לקבוצות.

אלגוריתם החציון - ● סדר את ההצבעות בסדר עולה: p1 ≤ p2 ≤ … ≤ pn. ● בחר את הצבעה מספר n\2 (עגל למעלה). תכונות: יעיל פרטו, אנונימי,מגלה-אמת חסרון-לא יעיל לקבוצות (אם יש יותר מנושא אחד לתקציב)
*כדי להכללי את האלגו' ליותר מנושא אחד שיהיה יעיל לקבוצות אפשר להנריץ את אלגוריתם החציון על כל סעיף בנפרד ולנרמל ע"י הכפלה בשבר אבל אז האלגו' לא יהיה מגלה-אמת.
מסקנה- אלגוריתם החציון יכול לשמש לבחירת ערך בנושאים רבים נוספים שהם חד-ממדיים: ● כמה ימים בשנה צריך להיות שעון קיץ? ● מה צריך להיות מספר השרים בממשלה? ● ועוד…

אלגוריתם החציון המוכלל- אלגו'- ● בחר מראש קבוצה של הצבעות קבועות f1,…,fk. ● הוסף אותן לקבוצת הצבעות האזרחים p1,…,pn. ● הפעל את אלגוריתם החציון המקורי על קבוצת k+n ההצבעות (הקבועות ושל האזרחים).

משפט- כשיש שני סעיפי תקציב, אלגוריתם החציון המוכלל עם הצבעות קבועות מפוזרות באופן אחיד בין 0 ל-C הוא הוגן לקבוצות.
הוכחה- נניח ש-k אנשים תומכים רק בסעיף א (נותנים C), ו n-k תומכים רק בסעיף ב (נותנים 0). החציון המוכלל יהיה בהצבעה הקבועה מס' k, שערכה הוא בדיוק C*k\n.***

פונקציה F-
- בחר מראש קבוצה של פונקציות: $f_1(t),\ \dots,\ f_{n-1}(t);\ t\ in\ [0, 1]$
- כל הפונקציות רציפות ועולות, ומקיימות:$f_i(1)\ =\ C\ , f_i(0)\ =\ 0$
- לכל t בין 0 ל1,- אפשר לחשב לכל נושא, חציון מוכלל עם הצבעות קבועות $f_1(t),\ \dots,\ f_{n-1}(t)$
- עבור t=0, החציון = המינימום; הסכום ≤ C.
- עבור t=1, החציון = המקסימום; הסכום ≥ C.
- לפי משפט ערך הביניים, קיים *t שעבורו סכום הסעיפים = C
- התקציב = חציון מוכלל עם $f_1(t\ *),\ \dots,\ f_{n-1}(t\ *)$

חציון מוכלל עם פונקציות לינארית -

משפט: לכל n-1 פונקציות רציפות עולות, אלגוריתם החציון המוכלל מגלה-אמת.

נגדיר 1n- פונקציות ליניאריות: $for\ i\ =\ 1,\ \dots,\ n\ -\ 1.\ \ f_i(t)\ =\ C\ \cdot\ min(1, i\ \cdot\ t)$

<u>תכונות:</u> תקציב הוגן לקבוצות

<u>משפט-</u> אלגוריתם החציון המוכלל עם פונקציות ליניאריות אינו תמיד יעיל פארטו.

<u>הוכחה-</u> נניח שיש 9 נושאים, 3 אזרחים, C=30:
- <u>אזרח א:</u> 6, 6, 6, 0, 0; ,6, 0, 0.
- <u>אזרח ב:</u> 0, 0, 6, 6, 6, 6; ,0, 6, 0.
- <u>אזרח ג:</u> 6, 6, 0, 0, 6, 6; ,0, 0, 6.

עבור t=1/15, הצבעות קבועות 2 ,4, מתקבל: 4, 4, 4, 4, 4, 4; ,2, 2, 2; סכום=30, הפרש=24 . יש שיפור פארטו: ,5, 5, 5, 5, 5, 5; 0, 0, 0; סכום=30, הפרש=20.

<u>משפט-</u> **לא קיים** אלגוריתם מגלה-אמת, הוגן לקבוצות, ויעיל-פארטו.