מרצה: ד"א סגל הלוי דוד אראל

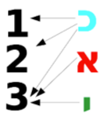
נכתב ע"י צבי מינץ – zvimints@gmail.com

הרצאה 1: חלוקת עוגה וקרקע

- חלוקה שוות שטח – לא טוב כי החלוקה לא הומוגנית (כל שטח שווה ערך אחר)
- חלוקה שוות ערך – לא טוב כי הערך הוא סובייקטיבי
- אלגוריתם "חתוך ובחר" – אלגוריתם ל2 שנותן חלוקה פרופרציונלית וללא קנאה
- חלוקה פרופרציונלית – $V_i(X_i) \geq \frac{V_i(C)}{n}$ (שמחים בחלקם)
- חלוקה ללא קנאה – $V_i(X_i) \geq V_i(X_j)\ \forall i,j$ (מסתכלים על הדשא של השכן)

<u>חלוקת פרופרציונלית אבל עם קנאה:</u>

- אלגוריתם המפחית האחרון (חותכים לפי סדר, האחרון שחותך מקבל) - נותן חלוקת פרופרציונלית (הוכחה באינדוקציה על מספר השחקנים שכל אחד מקבל לפחות $\frac{1}{n}$ מערך העוגה בעיניו) אבל $O(n^2)$
- אלגוריתם אבן הפז (כל שחקן מחלק ל$\frac{1}{2}$ ואז חותכים את העוגה בחציון, שולחים כל שחקן לחצי שלו ואז ברקורסיה) – חלוקה פרופורציונלית (הוכחה באינדוקציה על מספר השחקנים שכל אחד מקבל לפחות $\frac{1}{n}$ מערך העוגה בעיניו) ביעילות $O(n \cdot logn)$ וזה <u>חסם תחתון</u> לחלוקה פרופרציונלית אם משתמשים בשאילתות מסוג (הערכה Eval ו-סימון Mark)
- צריך למצוא דוגמא למה האלגוריתם המפחית האחרון ואבן הפז הוא עם קנאה



אחרי הקיצוץ של י הופך ל:

<u>חלוקה ללא קנאה (⇐ חלוקה פרופרציונלית)</u>

- אלגוריתם Selfridge Conway (עם השארית) (הוכחה שזה ללא קנאה ע"י שידוך בגרף)
- גילוי שחלוקה ללא קנאה ל-n שותפים (אם זה 3 אז זה סלפרידג'-קונווי ב5 שאילתות) לוקח לפחות n^2 שאילתות, וזה חסום ע"י $O(n^{n^5})$ ול-4 משתתפים זה חסום ע"י 200 שאילתות

הערה: רק אלגוריתם המפחית האחרון ואבן הפז נותן פרוסות קשירות

<u>חלוקת עוגה כמעט-ללא-קנאה עם פרוסות קשירות</u>

- <u>רקע:</u> כל חלוקה מוגדרת ע"י n מספרים חיובים שסכומם 1 באופן הבא: $l_1 + \cdots + l_n = 1$ יוצר מרחב n ממידי כאשר מרחב החלוקות הוא **סימפלקס** $n-1$ ממדי (קטע, משולש, טטראדר שזה כמו קובייה הונגרית רק בצורה של משולש ובאופן כללי סימפלקס)
  בכל נקודה בסימפלקס (יש אינסוף נקודות, אז איך בוחרים? נגיד ללא קנאה עד מילמטר אחד ואז נבנה סימפלסון במקום נקודה) ואז נחלק לסימפלקסונים קטנים שאורך כל צלע של כל אחד מהם הוא מילמטר אחד, נשאל כל שחקן "איזה חתיכה אתה הכי רוצה?" ונחפש נקודה עם וקטור $(x_1, x_2, \dots x_n)$ כך ש- $x_i \neq x_j\ \forall i,j$ זה יתן חלוקה **ללא קנאה**, אם נרצה **כמעט-ללא-קנאה** אז לא נחפש נקודה אלא **סימפלקסון** שבו אפשר לחלק כל קודקוד בסימפסקון שמקבל ערך שונה מ-x_i – זה נקרא סימפלקס n מלא עם n מספרים שונים
- אלגוריתם סימוס-סו (מוצא סימפלקס n מלא עם n מספרים שונים)
- הלמה של ספרנר Sperner Lemma : לכל n קיים סימפלקס n מלא עם n מספרים שונים
  תנאי ללמה: כל מספר על צומת בשפה הוא מספר שנמצא על קצות השפה
  זה מתקיים כי: כל שחקן מעדיף פרוסה לא ריקה
  ==ההוכחה== היא באינדוקציה על מספר המימדים שקיים סימפלקסון מלא שכזה, מוכיחים שיש מספר מעברים אי-זוגי (יש מספר א"ז של סימפלקסונים מלאים)
- לסיכום: (פרופורציונאלי, ללא קנאה, כמעט ללא קנאה, קשיר)
  (חתוך ובחר, המפחית האחרון, אבן הפז, סלפרידג'-קונוואי, סימוס סו)

הרצאה 2: חלוקה יעילה של משאבים הומוגניים

<u>חלוקת משאבים הומוגניים</u> (חומרים שחשוב כמות ולא איזה סוג מקבלים, לדוגמא עץ פלדה מניות נפט..)

- חלוקה הוגנת – כל אחד מקבל $\frac{1}{n}$ מכל משאב (לא יעיל כי אחד מסתכל על כל משאב אחרת)
- שיפור פארטו – מצב א' נקרא שיפור פארטו של מצב ב' אם הוא טוב יותר לחלק מהמשתתפים וטוב לפחות באותה מידה לכולם
- יעיל פארטו – אם לא קיים מצב אחר שהוא שיפור פארטו (תנאי הכרחי לבחירה הגיונית)
- אלגוריתם דיקטטור – משהו מקבל הכל (יעיל פארטו אבל לא הוגן)
- חתוך ובחר לא יעיל פארטו (כי משהו יכול לקבל משאב שהוא שם לו ערך 0)
- חלוקה שממקסמת סכום ערכים $\max_X \sum_{i=1}^{n} V_i(X_i)$ (האלגוריתם הוא לתת כל משאב לשחקן עם הערך הכי גבוהה) – זה יעיל פארטו אבל **לא הוגן**
- **משפט: כל חלוקה <u>הממקסמת את סכום הערכים</u> היא יעילה פארטו (הוכחה בהנחה בשלילה)**
- חלוקה הממקסמת סכום של פונקציה עולה של ערכים $max \sum_{i=1}^{n} f(V_i(X_i))$

- לכל פונ׳ קעורה (טיפוס על הר) יש נקודת מקסימום אחת ויחידה בכל תחום קמור ולכן קיים אלגו׳ מהיר למציאת נק׳ מקסימום
- אם $f(x)=\log(x)$ אז החלוקה היא לא רק יעילה פארטו אלא גם **ללא קנאה** (הוכחה)
- איך מוצאים את החלוקה? מסמנים ב-x את החלק שעמי מקבל מהאזור השמאלי לדוגמא ואז נקבל

$$\begin{aligned} \max \quad & f(81x+19)+f(80(1-x)+20) \\ s.t \quad & 0\le x\le 1 \end{aligned}$$

ואז נגזור, נשווה לאפס ונמצא את x

- $\log_a f(x)=\frac{f'(x)}{f(x)\cdot\ln(a)}$
- ספריית cvxpy
- ראינו שניתן למצוא חלוקה שהיא:

יעילה פארטו וללא קנאה – מקסום לוגים (יכול להיות לא קשיר)
קשירה וללא קנאה – סלפרידג׳-קונוואי, חתוך ובחר או סימונס סו (יכול להיות לא יעיל)
יעילה וקשירה – דיקטטורה (יש קנאה)
אבל אי אפשר את שלושתם.

הרצאה 3: חלוקה הוגנת של שכר דירה

- מטרה: דירה עם n חדרים ו-n שותפים ודמי שכירות R נרצה לחלק (חדר+מחיר) כך שלא יהיה קנאה
- אלגוריתם ל-2 שותפים: אחד מחלק את שכר-הדירה והשני בוחר חדר
- **הנחה חדרים סבירים** – בכל חלוקה של שכר דירה כל שוכר מוכן לקבל חדר כלשהו
- **הנחה דייר עניים** – כל שוכר מעדיף חדר בחינם על פני חדר בתשלום
- מודל אורדינלי – ההנחות הינם דיירים עניים, חדרים סבירים
- פתרון במודל אורדינלי:

עבור $n\ge 3$ נשתמש באלגוריתם סו כך שכל נקודה $(R\cdot x,R\cdot y,R\cdot z)$ כך ש $x+y+z=1$
וזה מקיים את הלמה של ספרנר כי לפי "הנחת דיירים העניים" כל שוכר מעדיף חדר בחינם ולכן כל מספר על צומת בשפה הוא מספר שנמצא על קצות השפה ולכן תנאי ספרנר מתקיים ולכן לפי הלמה של ספרנר, קיים סימפקסון שבו כל דייר מעדיף חדר אחר וזוהי חלוקה ללא קנאה בקירוב (שבלונה ללמה של ספרנר)

- **החסרון המרכזי במודל האורדינלי** הוא הנחת "הדיירים העניים". רוב השוכרים לא בהכרח מעדיפים חדר בחינם על חדר בתשלום, לדוגמא מרתף בחינם או יחידת הורים בשקל ואז אי אפשר להשתמש בלמה של ספרנר
- **קוואזי ליניאריות** – התועלת של דייר שמקבל חדר הינה ערך החדר פחות המחיר שלו, כלומר אם שוכר כלשהו חושב שחדר שווה $V_i(X_i)$ והוא משלם P_i אז התועלת הינה $V_i(X_i)-P_i$ (סותר בד"כ את "הדיירים העניים")
- מודל קרדינלי – ההנחות הינם חדרים סבירים וקוואזי ליניאריות
- פתרון במודל הקרדינלי:

במקום $(R\cdot x,R\cdot y,R\cdot z)$ אז להגדיר כל נקודה ל-$(\frac{1000}{x},\frac{1000}{y},\frac{1000}{z})$ ואז הלמה של ספרנר מתקיימת ⇐
חלוקה ללא קנאה, הבעיה הינה **שהחישוב עלול להתכנס מאד לאט**
אלגוריתם חישוב מהיר (סונג-ולאך): מביא השמה דיירים ללא קנאה (מקסום סכום ערכים הכוונה שזה טוב לבעל הבית)
קלט: מטריצה $n\times n$ כך שתא - $(i,j)=V_i(j \text{ חדר})$
פלט:

1. השמת דיירים לחדרים (יש $n!$ סה"כ השמות אבל אפשר לעבור רק על השמה אחת שממקסמת סכום ערכים)
2. וקטור עם n חדרים

משפט 1: בכל השמה ללא קנאה סכום הערכים של הדיירים בחדרים הוא מקסימלי (הוכחה)
משפט 2: כל וקטור מחיר ללא קנאה יישאר ללא קנאה לכל השמה ממקסמת סכום ערכים (הוכחה)
מסקנה שבשביל לבצע את שלב (1.) מספיק למצוא השמה אשר ממקסמת את סכום הערכים
כדי למצוא – רדוקציה לשידוך עם משקל מקסימלי בגרף כאשר הקב׳ זה (דייר, חדר) ויש משקל על כל קשת ויש לזה פתרון מהיר ע"י **האלגוריתם ההונגרי** ויש לזה מימוש בספרייה **networkx**
כעת מצאנו השמה, נשאר למצוא מחירים כך שסכום המחירים יהיה שווה לשכר דירה ע"י תיכנות לינארי ע"י הספרייה **scripy.linprog**

- בעיית הטרמפיסט – יכול להיות שבכל חלוקה ללא קנאה אחד הדיירים ישלם מחיר שלילי (צריך לשלם לו שיגור איתנו)

הרצאה 4: חלוקה הוגנת של חפצים בדידים (חפצים לא ניתנים לחלוקה, בד"כ אי אפשר למצוא חלוקה פרופרציונלית וללא קנאה לדוג׳ בית ל2 אנשים)

- **חלוקה ללא קנאה בקירוב** (חלוקה נקראת ללא קירוב מלבד 1 (EF1) אם לכל שני משתתפים א וב׳ אז קיים חפץ כלשהו שאם נוריד מהסל של ב׳ אז שחקן א׳ לא יקנא בו) – רמת הקנאה היא הקטנה ביותר האפשרית היות והחפצים הם בדידים
- קיים אלגוריתם כך שכשהחפצים בדידים תמיד קיימת חלוקה *EF1*
- אלגוריתם מעגלי הקנאה (עלול להחזיר תוצאה שאינה יעילה פארטו)

- אם יש m חפצים ו-n שחקנים אז זמן הריצה הוא $O(mn^3)$ (הוכחה)
- האלגוריתם מחזיר חלוקה *EF1* – באינדוקציה על כמות האטרקציות כי בכל שלב האלגוריתם החלוקה הנוכחית היא *EF1*
- **תזכורת:** כשהעוגה רציפה תמיד קיימת חלוקה *EF* ויעילה (סכום הלוגים או מכפלת הערכים $\sum log = \log(\Pi)$)
- כאשר החפצים בדידים – חלוקה הממקסמת את מכפלת הערכים היא *EF1* ויעילה פארטו אם:
  - העדפות אדיטיביות – ערך של סל הוא סכום הערכים של החפצים בסל
  - קיימת לפחות חלוקה אחת שבה כל שחקן מקבל ערך גדול מ0

  הוכחה: יעיל פארטו – הנחה בשלילה, ואז קיימת מכפלה גדולה יותר – סתירה

  EF1 – (הוכחה)
- **מסקנה: החלוקה "האידאלית" של חפצים בדידים היא חלוקה הממקסמת את מכפלת הערכים**
- מוצאים חלוקה זו ע"י פתרון *LP* של **שלמים** $\boldsymbol{x_{i,g} \in \{0,1\}}$ שזה קשה

$$\max \sum_{i=1}^{n} \log\left(\sum_{g=1}^{m} x_{i,g} \cdot V_i(g)\right)$$
$$s.t\ \forall g:\ \sum_{i=1}^{n} x_{i,g} = 1$$

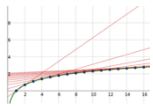
$$\max \sum_{i=1}^{n} W_i$$

הטריק לעקוף הוא (1) להוסיף משתנים $s.t\ W_i \le \log\left(\sum_{g=1}^{m} x_{i,g} \cdot V_i(g)\right)$ וצעד (2) הוא להניח שכל הערכים

$$\forall g:\ \sum_{i=1}^{n} x_{i,g} = 1$$

הם בין 1 ל-1000 ואז נחליף את האילוץ ל:

$$W_i \le \log k + [\log(k+1) - log k] + \left[\sum_{g=1}^{m} x_{i,g} \cdot V_i(g) - k\right]$$
$$\forall k \in \{1,2 \dots 1000\}$$



ואז זה נהפכת להיות בעיה לינארית כי במקום להיות מתחת לעקומה, זה כמו להיות מתחת ל1000 קווים, ופתרון לבעיה הלינארית הוא גם פתרון אופטימלי לבעיה המקורית

- **שיתוף מספר מינימלי של חפצים** (לפעמיים לא רוצים ללא-קנאה-בקירוב כי רוצים הוגן ממש)

  לדוגמא ילדים – משמרות משותפות, דירות מגורים – השכרה וחלוקת רווחים, דירות נופש – שימוש בזמנים שונים

  מודל: שני שותפים ו-m חפצים שיש עליהם מחלוקת, כל שותף מייחס ערך באחוזים לכל נושא. נרצה להשיג שלא יהיה קנאה, יעילות פארטו ונצטרך לחלק/לשתף לכל היותר חפץ אחד (אם לא נשתף יהיה קנאה)
- ניסיון ראשון: אחד מחלק השני בוחר – אין קנאה, חפץ אחד נחתך אבל **לא יעיל פארטו**
- ניסיון שני: כל חפץ נמסר למי שהכי רוצה אותו – יעיל פארטו, אף חפץ לי נחתך **אבל יש קנאה** (מקסום ערכים)
- ניסיון שלישי: מקסום מכפלת הערכים – אין קנאה, יעיל פארטו אבל לא ברור כמה **חפצים נחתכים**
- אלגוריתם המנצח המתוקן: נסדר את החפצים בסדר עולה של היחס שחקן א/שחקן ב. לאחר מכן ניתן את כל החפצים לשחקן ב', ואז נעביר את החפצים לשחקן א' לפי הסדר עד ש: (1) הסכום של שחקן א' שווה לסכום של שחקן ב' או (2) יש חפץ אחד שאם נחלוק אותו הסכום ישתווה
- האלגוריתם המנצח המתוקן מחזיר תמיד חלוקה ללא קנאה (הוכחה), מחזיר חלוקה יעילה והוגנת עם שיתוף של חפץ אחד לכל היותר (הוכחה)
- אלגוריתם לחלוקת חפצים בדידים:
  - חלוקת שכר דירה – דורש כסף
  - המנצח המתוקן – דורש שיתוף של חפץ אחד
  - מקסום מכפלת הערכים – יש קנאה (EF1)

הרצאה 5: אלגוריתמים מגלי-אמת (רוצים למקסם רווח – זה טוב למוכר)

- <u>בעיה 1</u>: מציאת מסלול זול ביותר בין 2 נק' בהינתן רשת כאשר לכל קשת יש עלות מעבר

  אם העלות של כל קשת ידוע – הבעיה קלה (Dijkstra), אבל אם העלות של כל קשת ידועה רק לבעל הקשר זו בעיה
- <u>בעיה 2</u>: בחירת פרסומות לדף רשת כאשר בהינתן m מפרסמים שונים כך שלכל מפרסם יש ערך שונה להקלה על הפרסומות שלו, ובדף יש $k < m$ מיקומים עם אחוזי הקלה שונים, יש צורך לבחור k מפרסמים ולתת מיקום לכל מפרסם כך שתוחלת סכום הערכים תיהי מקסימלית. (אם הערך של כל מפרסם ידוע לכולם – אפשר לפתור בצורה חמדנית, אם זה ידוע רק למפרסם אז זה בעיה)
- <u>בעיה 3</u>: בהינתן m מפרסמים שונים, לכל מפרסם יש פרסומת באורך שונה וגם ערך שונה להשמעת הפרסומת שלו, ובתוכנית יש זמן T קצוב אז צריך לבחור את הפרסומות שימלאו לכל היותר את התוכנית כך שסכום הערכים גדול ביותר (אם הערך ידוע לכולם אז זה בעיית התרמיל, אם זה ידוע רק למפרסם זה בעיה)
- **אלגוריתם אמיתי/מגלה אמת** – אם לכל משתתף כדאי להגיד את הערך האמיתי שלו – לא משנה מה האחרים עושים

  <u>יתרונות</u>: מונע ריגול, לקיחת סיכון ועומס על השרתים
- **מוטיבציה:** אם הערכים ידועים לכולם מחשבים מקסימום, אם לא והערכים הם מידע פרטי שידוע רק למוכר אז המציאו את המכרז
- מכרז מחיר ראשון לא טוב (יש צורך בריגול), כדי להוכיח שהוא אינו אמיתי יש צורך להביא דוגמה נגדית אחת, נניח שהערך של שחקן כלשהו הוא 10 והערך של השני הוא 5, הכרזה אמיתית תיתן לו תועלת 0 אבל ההכרזה לא אמיתית תיתן לו תועלת גדולה מ-0 לדוגמא 4 עבור ההכרזה 6
- אם התועלת שלי גדולה ביותר כשאני מכריז את הערך האמיתי בלי קשר לאחרים אז המכרז הוא **אמיתי**
- מכרז מחיר שני / ויקרי: המשתתפים כותבים את הכרזות במעטפות, המעטפות נפתחות ומסודות בסדר יורד, בעל ההכרזה הגבוהה ביותר זוכה בחפץ ומשלם את ההכרזה השנייה בגבוהה

- מכרז מחיר שני הוא **אמיתי** כאשר לשחקנים יש העדפות **קוואזי-ליניאריות** (מסתכלים על התועלת ולא על הערך)
<u>הוכחה:</u> נניח כי הערך שלי הוא v והערך המקסימלי של האחרים הוא x אזי לא משנה מה אגיד התועלת שאשיג אם אזכה היא $\max(0, v-x)$ כאשר $v-x$ אם $x>v$ ו-0 אם $v<x$ וכשאני מכריז v אני מקבל בדיוק תועלת זו
- שלב הבא – כמה פרסומות בעמוד אחד <u>(בעיה 2)</u>
- <u>מודל:</u> יש מכרז פרסון (לכל משבצת k יש הסתברות הקלקה r_k כאשר $r_1 > r_2 > \cdots$ ולכן מפרסם j יש ערך הקלקה v_j מכאן כל מפרסם מעריך את משבצת k ב-$r_k \cdot v_j$, המטרה למצוא אלגוריתם אמיתי למקסום ערכים
- אלגורתמים חמדני (כשכולם רואים את המחיר אז זה לא אמיתי)
- מכרז ויקרי-קלארק-גרובס (VCG) – מכרז אמיתי אם מידע פרטי (יעיל פארטו ואמיתי)
ההנחות למודל הוא שיש מספר סופי של תוצאות אפשריות **וקוואזי ליניאריות** (מסתכלים על התועלת)
- בהינתן קלט מטריצה (אפשריות × שחקנים) כאשר האפשריות זה העמודות, מחשבים את הסכום הערכים הגבוהה ביותר עבור כל אפשרות, שזה סכימה של השורות בכל עמודה, לאחר מכן בודקים מה ההאפשרות (עמודה) עם הערך הגבוהה ביותר.
לאחר מכן, בונים מטריצה של (אפשריות × ללא שחקן i) ומחשבים ומסמנים ב-■ את התוצאה הגבוהה ביותר בכל שורה שזה (2) – כי זה מה שהאלגוריתם היה נותן, הוא בוחר את המקסימום מבין התוצאות.
ולאחר מכן בונים מטריצה של (תשלום | ערך | תועלת × שחקנים + סה״כ) כאשר התשלום זה ההפרש בין ■ לבין הערך בעמודה (שזה 1) את סה״כ התשלום משקיעים בלוטו
- <u>הוכחה שהוא אמיתי:</u> התועלת של כל שחקן הוא (הערך של השחקן – כמה הוא משלם) וכמה הוא משלם זה (הסכום של שאר השחקנים בלעדיו – הסכום של שאר השחקנים כאשר הוא פה) שזה יוצא שווה מתמטית ל-**סכום הערכים של כל השחקנים פחות מספר שאינו תלוי בהצהרה שלו** ולכן הוא צריך למקדם את הסכום הזה שזה אומר למקסם את סכום הערכים שזה אומר למקסם את הערך האמיתי בלי קשר לערכים ולכן המכרז הוא **אמיתי**
- מכרז VCG למסלול זול ביותר (הגרף) – רדוקציה למינוסים
    הסכום הוא (המסלול הקצר ביותר ללא הצלע)–
    התשלום הוא הסכום פחות (כמה שאר השחקנים משלמים כאשר הוא משתתף)

הרצאה 6: אלגוריתמים מגלי-אמת מתקדמים (מאריסון + בעיית התרמיל)

- בעיית התרמיל
- אלגוריתם חמדני א׳: ממיינים לפי ערך גדול ביותר (לא טוב)
- אלגוריתם חמדני ב׳: ממיינים לפי $\frac{\text{ערך}}{\text{משקל}}$ גדול ביותר (לא טוב)
- אלגוריתם חמדני א׳ + ב׳: מריצים את אלגוריתם א׳ וב׳ ולוקחים את התוצאה הטובה ביותר נותן קירוב $\frac{1}{2}$ (הוכחה)
- אלגוריתם חמדני א׳ + ב׳ **לא אמיתי** (מעדיף לשקר/לבדוק מה עם האחרים ולהביא טיפה יותר בשביל להכנס)
- אלגוריתם חמדני א׳ + ב׳ **לא אמיתי** עם תשלומי VCG $\{\frac{54\$}{52k}, \frac{52\$}{51k}, \frac{49\$}{49k}\}$ והראשון זוכה ומקבל תועלת של -1 ולכן הוא מעדיף לתת מחיר נמוך יותר, וזה נופל כי האלגוריתם לא ממקסמים סכום ערכים
- **כאשר לא ניתן למקסם ערכים (קשה חישובית או סתם לא רוצים) אז נשתמש מכרז מאריסון**
- למכרז מאריסון יש **כלל בחירה** (קובע לכל משתתף אם הוא נבחר או לא) ודרוש **כלל תשלום** שאיתו המכרז יהיה אמיתי
- **כלל בחירה** - פונקציה c המקבלת קקלט את הערכים של כל המשתתפים ומחזירה וקטור בינארי (1 אם נבחרת)
- **כלל תשלום** – פונקציה p המקבלת קקלט את הערכים של כל המשתתפים ומחזירה וקטור מספרי של תשלומים
- **כלל בחירה מונוטוני** – אם לכל i ההסתברות ש-i יבחר הוא פונקציה עולה של v_i
- **כלל בחירה דטרמיניסטי** הוא מונוטוני אם עבור כל i אז כאשר הוא נבחר עם ערך x אז הוא יבחר גם עם ערך גדול מ-x
- משפט מאריסון:
    - **לכל** כלל בחירה לא מונוטוני – אין כלל תשלום אמיתי
    - לכל כלל בחירה **מונוטוני** – קיים כלל תשלום אמיתי
    - אם כל שחקן שלא נבחר – לא משלם, אז כלל התשלום הוא יחיד
- לפונקציית בחירה מונוטונית יש ערך סף – הערך שבו הפונקציה c מתחלפת מ-0 ל-1.
דוגמה: עבור הכלל ״בחר את הערך הגבוהה ביותר״ אז הערך הסף של הנבחר הוא המחיר השני
- **כדי למצוא את כלל התשלום אחד והיחיד המועמד להיות אמיתי:**
    - לכל שחקן i יש ערך סף מסויים t_i אשר נקבע ע״י c
    - אם $v_i > t_i$ אז השחקן נבחר ומשלם t_i
    - אחרת, השחקן לא נבחר ולא משלם
- אם יש מכירה רק של חפץ אחד אז מכרז מאריסון זהה למכרז מחיר שני
- **מכרז מאריסון** לא חייב למקסם סכום ערכים

הרצאה 7: אלגוריתמים למיקסום רווח (זה טוב לקונים) - מאריסון עם חפץ יחיד

- בגלל שלמקסם רווח זה תלוי בערך של השחקן נעבור למודל הסתברותי אשר ממקסם **תוחלת** רווח אשר דורש מידע סטטיסטי מקדים על הערכים של השחקנים
- נרצה למקסם רווח – מכרז ויקרי ממקסם סכום ערכים וזה לא טוב לקונה (רווח נמוך) ולכן נעבור למכרז מאריסון, רק צריך להגדיר כלל בחירה מתאים.

- דוגמה: ידוע שקונה משלם אחיד [10,30] אז כדי למצוא את תוחלת הרווח המקסימלית נגדיר כלל בחירה כלשהו אשר אפשרי לשחקן אחד והוא "בחר את הקונה אם"ם ערכו גדול מערך הסף p" ולכן התוחלת הינה:
(עבור ערך סף קטן מ-10 אז תוחלת הרווח הוא המספר ואם בין 10 ל30 אז זה ההסתברות שהוא נבחר כפול כמה הוא משלם ועבור 30 זה אפס) – באופן כללי זה נראה כך:
$$R(p)=\frac{30-p}{30-10}\cdot p\Rightarrow R'(p)=\frac{30-p}{20}+\frac{-1}{20}\cdot p\Rightarrow p=15$$
- באופן כללי, נסמן $P[v<p]=F(p)$, אצלנו זה $\frac{p-10}{20}$
- פונקציית הערך הווירטואלי - $r(v):=v-\frac{1-F(v)}{F'(v)}$
- המכרז אופטימלי הוא: **מכרז אם"ם $r(v)>0$**, בדוגמא שלנו זה $v-\frac{1-\frac{v-10}{20}}{\frac{1}{20}}$ וזה יוצא למכור אם $v>15$ (אם ההתפלגות היא בין 0 ל-1000 אז הערך הווירטואלי הוא $2v-1000$ לדוגמא)
- מקסום רווח בשיטת מאריסון – בהינתן **שוק חד-פרמטרי** (לכל משתתף יש ערך כספי יחיד להיבחרות והערך של משתתף j לקוח מהתפלגות F_j) דרוש (1) כלל בחירה לבחירת תת קבוצה של משתתפים ו-(2) כלל תשלומים שאיתו כלל הבחירה הוא אמיתי
- **תוחלת הרווח של כלל בחירה c שווה לתוחלת סכום הערכים הווירטואליים של הנבחרים**

ולכן כלל הבחירה הממקסם את תוחלת הרווח הוא: **בחר את הקבוצה שבה סכום הערכים הווירטואליים הוא הגדול ביותר אבל ש-0 $r_i(v)>0$**
- לפי שיש פחות כסף לשלם (ההסתברות $[F_{b_{low}}-\epsilon,F_{b_{high}}-\epsilon]$ = F_a) אז הערך הווירטואלי גדול יותר וכאן משלם פחות ונבחר (הוא משלם כמו ערך הסף כי זה מאירסון)

הרצאה 8: חלוקת עלויות (למצוא חלוקה הוגנת לדמי נסיעה בין נוסעים)

- שאלה א׳: **חלוקה הוגנת:** איך לחלק דמי-נסיעה בין הנוסעים?
- שאלה ב׳: **מכרז:** איך להחליט מי ישתתף בנסיעה?
- <u>מודל:</u> יש קבוצה של שחקנים N ולכל תת קבוצה S העלות של מתן שירות רק לתת קבוצה הזאת זה $c(S)$ המטרה היא לגבות מכל שחקן j תשלום $p(j)$ כך שסכום התשלומים הכולל $c(N)$ מכסה את הנסיעה וכלל התשלום הוגן
- **עלות שולית** – העלות השולחית של שחקן j ביחס לקבוצת שחקנים S היא התוספת שהוא מוסיף לעלות כשהוא מצטרף לקבוצה $c(S\cup\{j\})-c(S)$
- **עקרון ההגינות:** כלל תשלום נקרא *סימטרי* אם הוא תלוי רק בעליות השוליות: אם לשני שחקנים יש עליות שוליות זהות ביחס לכל הקבוצות אז הם צריכים לשלם אותו הדבר
- **עקרון האפס:** שחקן שעבורו כל העליות הן אפס אז הוא משלם 0
- **עיקרון הלינאריות**: אם מכפילים עליות בקבוע – כל התשלומים נכפלים באותו הקבוע, לדוגמא המרת שקלים, אם מחברים שטי טבלאות-עליות כל התשלומים מתחברים, לדוגמא חישוב דלק בנפרד ועלות אגרת כביש בנפרד
- **משפט שאפלי Shapley:** ישנו כלל תשלומים יחיד המקיים את כל התכונות (עקרון ההגינות, עקרון האפס, עקרון הלינאריות) כלל התשלומים הזה נקרא **ערך שאפלי**

אלגוריתם לחישוב ערך שאפלי (בצבע בצד עם ה37 בסכום) (בעיית NP כי צריך למצוא $n!$ סידורים)

קבוצה:	0	א	ב	ג	א,ב	א,ג	ב,ג	א,ב,ג
עלות:	0	10	15	25	20	25	30	37

סדר:	א-ב-ג	א-ג-ב	ב-א-ג	ב-ג-א	ג-א-ב	ג-ב-א	ממוצע
א:	10	10	5	7	0	7	6.5
ב:	10	12	15	15	12	5	11.5
ג:	17	15	17	15	25	25	19
סכום:	37	37	37	37	37	37	37

- הוכחת נכונות: **כיסוי מלא** ← נכון לכל סדר בנפרד ← נכון גם לממוצע על כל הסדרים, **סימטריה** ← ערך שאפלי של כל שחקן נקבע רק לפי העליות השוליות שלו, **אפס** ← העליות השוליות 0 ← הממוצע 0, **לינאריות** ← ערך שאפלי הוא פונקייה לינארית של הערכים טבלה
- חלוקת עלות של בנייה מסלול-המראה בין חברות תעופה הצריכות אורכים שונים – בודקים קווים חופפים מהמסלול הכי קטן ואז המחיר שלו לחלק בכמה משתתפים, ואז הבא בתור זה מה שחשבנו + המשך המסלול חלקי כמה משתתפים וכו׳

הרצאה 9: חלוקת עלויות (איך להחליט מי משתתף בנסיעה)

- <u>מודל:</u> לכל שחקן j יש ערך נסיעה v_j אם תת קבוצה נוסעת,
הרווחה החברתית היא הסכום הערכים של הנוסעים פחות עלות הנסיעה,
רוצים כלל החלטה שהוא (1) יעיל פארטו – ממקסם את רווחה החברתית
(2) אמיתי – מעודד כל שחקן jלגלות את v_j
- מכרז לקבל שירות בשיטת VCG: התוצאה האפשרית היא תת קבוצה מ-2^n, הערך של שחקן j הוא v_j אם הוא נוסע ו-0 אחרת, הערך של הנהג הוא מינוס עלות הנסיעה. בוחרים את התוצאות הממקסמת את סכום הערכים.
תשלום שחקן j: הסכום בלי j פחות הסכום של האחרים (כולל הנהג) כש-j נמצא
הבעיה במכרז הוא **גירעון (חוסר)** ולכן הוא **לא** מתאים לבעיה

	יעיל פארטו	מאוזן תקציבית	אמיתי
VCG	כן	לא	כן
מולין-שנקר	לא	כן	כן
תשלום = ערך	כן	כן	לא

קבוצה:	0	א	ב	ג	א,ב	א,ג	ב,ג	א,ב,ג
עלות:	0	10	15	25	20	25	30	37
תועלת מהנסיעה:		8	22	16				
ערך נוסע א:	0	8	0	0	8	8	0	8
ערך נוסע ב:	0	0	22	0	22	0	22	22
ערך נוסע ג:	0	0	0	16	0	16	16	16
ערך "נהג":	0	-10	-15	-25	-20	-25	-30	-37
סכום:	0	2-	7	9-	**10**	1-	8	9
נבחר:	FALSE	FALSE	FALSE	FALSE	**TRUE**	FALSE	FALSE	FALSE
סכום בלי א:	0	-10	7	-9	2	-9	**8**	1
סכום בלי ב:	**0**	-2	-15	-9	-12	-1	-14	-13
סכום בלי ג:	0	-2	7	-25	**10**	-17	-8	-7

	תשלום	ערך	תועלת	
א:	6	8	2	
ב:	12	22	10	
ג:	0	0	0	
נהג:		-20	**-2**	גירעון←

- אלגוריתם מכרז לקבל שירות – מולין-שנקר:
- כלל תשלום נקרא **מונוטוני** אם כשהקבוצה קטנה, אז התשלום גדל או שווה
  $if\ S \leq T \quad then\ for\ all\ j: \quad P(S,j) \geq P(T,j)$
- מכרז מולין-שנקר הוא **אמיתי** אם כלל התשלום מונוטוני (לא מתקיים אם 2 נוסעים ב-2 קצות שונים) הוא **לא יעיל פארטו**

הרצאה 9.5: תקצוב השתתפותי (שהסכום הכולל קבוע מראש, התקצוב הוא בינארי – כל פריט מתוקצב או לא)

- יש שאלון עם צק' בוקס על פרויקטים שרוצים לעשות בחברה ואז יש "תרמילי" עם פרויקט, כמה אנשים הצביעו וכמה עולה התקציב ואז ממלאים פריטים עד שנגמר התקציב (לפי Total Votes) – **לא טוב**, זה לא הוגן אם זה הייתה בעיית התרמיל כי יש פה בעיה של הגינות (51% גרים בשכונה א' והשאר בב' אז התקציב יהיה לא')
- <u>מודל:</u> יש $X = \{x_1, .., x_m\}$ פריטים, פונקציית עלות $c: X \to N$ והצבעות $V = \{V_1, \dots, V_n\}$ כך ש-$V_i \subseteq X$ כלומר כל הצבעה היא תת קבוצה של פריטים **ושיטת תקצוב** כך שבהינתן סכום כולל L יש לחשב תקציב (קבוצה של פריטים) $X' \subseteq X$ המקיים $c(X') \leq L$
- תקציב נקרא **פרופורציונלי-חזק** אם לכל קבוצת מצביעים בגודל k אם כל חברי הקבוצה מסכימים על פריטים שהעלות הכוללת שלהם לפחות $\frac{kL}{n}$ אז הסכום המוקצב לפריטים, שלפחות אחד מחברי-הקבוצה רוצה הוא לפחות $\frac{kL}{n}$ הרעיון שלכל אזרח יש "זכות" לקבוע לגבי יחידת-תקציב אחת $\frac{L}{n}$, **חזק מדי** עבור $L = 4, c(x_1) = 2.5$ ו-4 אזרחים
- תקציב נקרא **פרופורציונלי-חלש** אם לכל קבוצת מצביעים בגודל k הסכום המושקע בפריטים, שלפחות אחד מחברי הקבוצה רוצה הוא **לפחות** העלות הגדולה ביותר של קבוצת-פריטים שכל חברי הקבוצה מסכימים עליהם, ועולה לכל היותר $\frac{kL}{n}$
- תקציב פרופורציונלי תמיד קיים

סכום כולל: 30;
סכום לאזרח: 5.

אזרח	1,2,3,4	5,6
פריטים	א,ב	ג

פריט	א	ב	ג	ד
עלות	**20**	15	15	10

16 תת-קבוצות של פריטים:
- אבגד, אבג, אבד, אגד, בגד, אב, אג, אד, בג, בד, גד -- יקרים מדי.
- א – הקבוצה K בגודל 4, העלות 20, מתקצבים.
  - *נשארו אזרחים 5,6 ופריטים ב,ג,ד.*
- ב, ג, ד – יקרים מדי; סיימנו.

- אלגוריתם עזיז-לי-טלמון מוצא את התקציב הפרופורציונלי

  אתחול:
  תקציב – קבוצה ריקה
  מקופחים – כל האזרחים

  מסדרים את 2^m קבוצות הפריטים בסדר יורד של עלות
  לכל קבוצת פריטים Y מהיקרה לזולה:
  נחשב את קבוצת-האזרחים K שרוצים את כל הפרטים ב-Y,
  נניח שבקבוצה יש k אזרחים. אם עלות הפריטים ב-Y היא לכל היותר $\frac{kL}{n}$:
  הוסיף את הפריטים ב-Y לתקציב והורד את האזרחים ב-K מקב' מקופחים (אם נשאר עודף, אפשר לקנות עוד אם זה הוגן)
- אלגוריתם עזיז-לי-טלמון אף פעם לא חורב מגבולות התקציב (הוכחה), ותמיד מחזיר תקציב פרופורציונלי חלש (צ"ל שסך המימון המיועד לפריטים שחברי K רוצים הוא לפחות $c(X)$ כאשר K מסכימים על X ו-$c(X) \leq \frac{kL}{n}$) סיבוכיות $O(n \cdot 2^m)$

הרצאה 10: תיאום תרומות (האזרחים תורמים מרצונם – כל אחד מחליט כמה לתת, התקצוב הוא רציף)

- עד עכשיו דיברנו שהסכום הכולל קבוע מראש, התקצוב הוא בינארי – כל פריט מתוקצב או לא
- הסבר למה צריך אלגוריתם כדי למקסם תועלת זמן, כל 1000 שקל מאפשרים לתפעל שעה, יש 2 משתתפים עם 3000 שקל שאוהבים את (א,ב) והשני (ב,ג) ואז אם הם לא היו מתואמים התקציב היה (1500,3000,1500) וזה נותן 4 וחצי שעות במקום (0,6000,0) שזה 6 שעות – שזה יעיל פארטו (**אפשר גם להשיג יעילות וגם לתרום**)
- רוצים מאלגוריתם תיאום תרומות – **יעיל פארטו, עידוד השתתפות, אמיתי**
- **קלט לכל אחד מהאלגוריתמים:** לכל אחד יש 100 שקל וההצבעות (א+ב),(א+ג),(א+ד),(ב+ג),א
- אלגוריתם האוטיליטרי – ממקסם סכום התועלת, מעביר את הכסף לנושאים עם הכי הרבה תומכים (**פלט:** (0,0,0,500)) – יעיל פארטו, אמיתי אבל **לא** מעודד השתתפות

כש-$b < 4$ אז הכל ילך ל-א

- **אלגוריתם האוטיליטרי-על-תנאי** – ממקסם את סכום התועלות תחת האילוץ שכל אזרח תורם רק לנושאים שהוא תומך בהם, כל אזרח תורם לנושאים, מאלה שהוא תומך בהם, עם הכי הרבה תומכים אחרים (**פלט**: (0,50,50,400)) – אמיתי, מעודד השתתפות (עבור יחידים) אבל לא יעיל פארטו
- **אלגוריתם מקסום המכפלה** – מקסום מכפלת התועלת (**פלט**: (0,65,65,370)) – יעיל פארטו, מעודד השתתפות אבל לא אמיתי
- לא קיים אלגוריתם המקיים בו-זמנית את התכונות הבאות (יעיל פארטו, מגלה אמת, מעודד השתתפות)

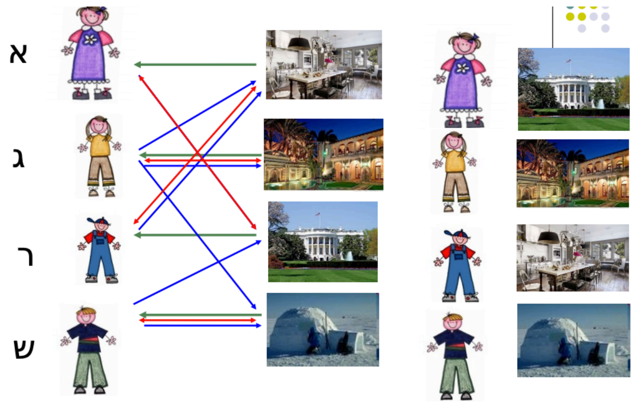
הרצאה 10.5: שידוכים (מחלקות וסטודנטים), אלגוריתם "הקבלה על תנאי")

- בעיות חד צדדיות (שיבוץ סטודנטים לחדרים בדירה שכורה) או בעיות דו צדדיות (שיבוץ סטודנטים למחלקות באוני')
- שוק דו צדדי – *שוק דו-צדדי* הוא שוק שבו צריך להתאים בין משתתפים משתי קבוצות, כאשר לכל משתתף יש העדפות בשוק דו צדדי אי אפשר לצפות **להגינות** (אם המחלקה לא רוצה – חיים קשים) אבל אפשר לצפות יעיל פארטו או **יציבות**
- שני סטודנטים c_1, c_2 ושתי מחלקות m_1, m_2 כאשר הסטודנטים מדרגים $m_1 > m_2$ והמחלקות מדרגות $c_1 > c_2$ אז השידוך $c_1 - m_2$ הוא יעיל פארטו (כי מחלקה 2 הכי מרוצה והסטודנט קיבל מחלקה הכי טובה (בשידוך השני)) אבל אם מחלקה 1 תפנה לסטודנט 1 אז השידוך יתפרק, אנשים לא ישתפו פעולה, **השוק ייפרם** (פוגעים לאט לאט...)
- **זוג מעורר**: סטודנט ומחלקה שאינם משודכים, והם מעדיפים זה את זה על פני "השידוכים" הנוכחים שלהם
- **שידוך יציב**: שידוך בלי זוגות מערערים
- **אלגוריתם "הקבלה על תנאי" (יש הנחות בסיס – בדף נוסחאות) – למציאת שידוך יציב**
- האלגוריתם **מסתיים**, **בשידוך** ו**יציב**
- מי מציע ומי מקבל? מסתבר שעדיף להיות בצד המציע (הוכחה)
- ב**מחלקות גדולות** שבהם יש כמה מקומות, כל מחלקה "מקבלת על תנאי" את הסטודנטים הטובים ביותר שפנו אליה, עד לכמות המקומות הפנויים אצלה ודוחה את כל השאר, זה עובד כשאין תלות בין סטודנטים שונים וזה לא עובד כאשר לדוגמאת זוגות סטודנטים נשואים רוצים להתקבל לאותה מחלקה (יש תלות)
- אלגוריתם "קבלה על תנאי" הוא **לא אמיתי** עבור המחלקות, אבל **אמיתי** עבור הסטודנטים כאשר הסטודנטים מציעים
- אלגוריתם "קבלה על תנאי" – כשהסטודנטים מציעים – אמיתי לסטודנטים אבל לא למחלקות, כנ"ל למחלקות.

הרצאה 11: אלגוריתמי החלפה

- החלפה של בתים, החלפת תורניות בין עובדים, החלפת חדרים בין סטודנטים במעונות
- למה לא להריץ אלגוריתם **לחלוקה הוגנת**? כי סטודנטים שכבר יש להם חדרים יחששו להפסיד ויעדיפו לא להשתתף
- אלגוריתם הוא **מעודד השתתפות** אם מצבו של כל משתתף לאחר הביצוע טוב לפחות כמו מצבו לפני הביצוע
- האם קיים אלגוריתם החלפה שהוא אמיתי, יעיל פארטו ומעודד השתתפות?
- **הערה**: אלגוריתם שהוא אמיתי ויעיל פארטו (עם כסף זה *VCG*) בלי כסף – (כל סטודנט מדרג את כל המעונות מהכי טוב להכי פחות טוב ואז עוברים על הסטודנטים בסדר כלשהו ונותנים לכל סטודנט את החדר הכי טוב ברשימה שלו) – לשים רק שלשה ראשונים זה לא אמיתי כי אז הוא ישים את הכי טוב במקום הרביעי
- **קואליציה מערערת** – קבוצת משתתפים שיכולה **לפרוש** ולבצע החלפה שהיא טובה באותו מידה לכל חברי הקבוצה וטובה יותר לחלק מחבריה (זה מכיל את המושג זוג מערער = קואלציה מעוררת בגודל 2)
- **שיבוץ יציב** – שיבוץ שבו אין קואליציה מערערת: **יציבות** ← **עידוד השתתפות**, **יעילות פארטו** (לא מעודד השתתפות אז אחד האנשים נזקו ואז היה קואליציה מערערת בגודל 1 שזה הוא עצמו, וזה גם יעיל פארטו כי אם היה שיפור פארטו אז הייתה קואלציה מערערת)
- האם קיים אלגוריתם החלפה שהוא אמיתי ומוצא שיבוץ יציב? - כן
- **אלגוריתם "מעגלי המסחר"**: אם כל יחסי העדפות הם חזקים (אין אדישות) אז (1) קיים שיבוץ יציב אחד ויחיד ו-(2) קיים אלגוריתם אמיתי המוצא אותו אשר נקרא "אלגוריתם מעגלי המסחר"

מאתחלים גרף מכוון שבו: הצמתים הם האנשים והבתים, יש קשת מכל אדם לבית שהוא הכי רוצה ומכל בית לאדם שגר בו עכשיו ואז (דף נוסחאות)

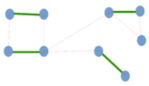


- יש מעגל בגרף אם $|E| \geq |V|$ מכוון שהגרף סופי
- אלגוריתם מעגלי המסחר **מסתיים** (כל עוד גרף לא ריק יש לפחות מעגל מכוון אחד ולכן בכל שלב הגרף קטן עד שמתרוקן)

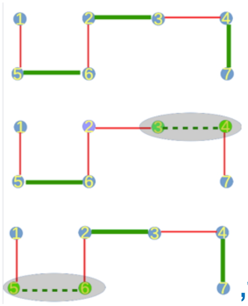
- אלגוריתם **מעודד השתתפות** (שהבית שלך יוצא מהגרף אתה מייד מקבל בית חדש, הבית החדש הוא בית שהצבעת עליו והוא טוב ביותר מבין הבתים הזמינים ולכן הוא טוב לפחות כמו הבית שלך)
- אם כל יחס העדפות חזקים (אין אדישות) אז אלגוריתם מעגלי המסחר **מוצא שידוך יציב**
- אלגוריתם מעגלי המסחר הוא **אמיתי** (**הוכחה**)
- אם כל יחס העדפות חזקים (אין אדישות) אז יש **שידוך יציב** אחד וזה חשוב כי אם יש 2 אז אפשר להתלונן אך אם יש 1 אז זה המצב

הרצאה 12: החלפת כליות

- כמעט בכל המדינות יש מחסור בכליות להשתלה, **אסור** לתרום כליות תמורת כסף **ומותר** לתרום כליה תמורת כליה
- תורת **מוכן** לתרום לחולה אבל **לא מתאים** בגלל סוג דם או סיבות אחרות
- אלגוריתם מעגלי המסחר לא פותר את הבעיה כי המעגלים ארוכים מדי, בהחלפת כליות מעדיפים מעגלים קצרים כי כל ההשתלות במעגל חייבות להתבצע **במקביל** ולכן במקום לחפש מעגלים נחפש **שידוכים** (**גדול ביותר**)



- **מסלול שיפור** – מתחיל ומסתיים בצמתים לא משודכים, ומתחלף בין קשתות בתוך ומחוץ לשידוך
האלגוריתם: כל עוד יש מסלול שיפור – הפוך אותו (הסיבה שיש זה בגלל שהוא גרף לא דו-צדדי)



- **הלמה של ברג' *(Berge's Lemma)* – שידוך הוא מקסימלי אם"ם אין מסלול שיפור (יעיל פארטו)**
- איך מוצאים מסלול שיפור? בעזרת **אלגוריתם הפרחים** $O(|V|^2|E|)$
- אפשר למצוא גם שידוך עם **משקל גדול ביותר**, נמצא בספרייה *network.max_weight_matching*
- מי השחקנים בבעיית שידוך הכליות? **הזוגות** – יכולים לכל היותר להסתיר קשתות, אבל זה לא יעזור להם
המרכזים הרפואיים – יכולים להסתיר זוגות , וכך לשדך אותם באופן פנימי
האינטרס של המרכזים הרפואיים הוא לדאוג לחולים "שלהם" – שכמה שיותר חולים שלהם יקבלו כלייה
- **אין** אלגוריתם שהוא גם **יעיל פארטו וגם אמיתי** עבור המרכזים הרפואיים
- **תמריצים של מרכזים רפואיים – קירוב $\frac{1}{2}$ (כיוון שאין אלגוריתם אמיתי המשיג את השידוך הגדול ביותר, נרצה אלגוריתם שידוך שהוא גדול ביותר בקירוב אבל אמיתי) – האלגוריתם זה המנגנון** (מחשבים שידוך הגדול ביותר מבין כל השידוכים שבהם מספר הקשתות פנימיות בכל מרכז רפואי הוא **מקסימלי**)
- כיום אפשר לבצע שישה ניתוחים בו זמנים ע"י החלפת כליות במעגל באורך 3, איך מוצאים הכי הרבה מעגלים באורך 3? בעיית *NP-Hard*

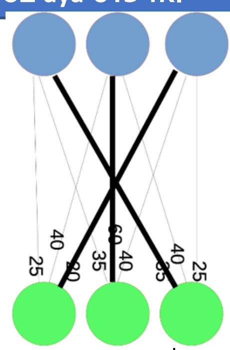
קטעי קוד python

הרצאה 2 – יעיל פארטו

```python
import math, cvxpy
from cvxpy import log

print("\n\n\nPROBLEM #1")
print("A cake with three regions has to be divided among 2 people with values:")
print("2 3 4")
print("8 7 6")

# Define x,y,z = the fraction of each region given to player 1.
x = cvxpy.Variable()
y = cvxpy.Variable()
z = cvxpy.Variable()

print("\nMaximize the sum of logs:")
prob = cvxpy.Problem(
    objective   =  cvxpy.Maximize(log(2*x + 3*y + 4*z) + log(8*(1-x)+7*(1-y)+6*(1-z))),
    constraints = [0 <= x, x <= 1, 0 <= y, y <= 1, 0 <= z, z <= 1])
prob.solve()
print("status:", prob.status)
print("optimal value", prob.value)
print("optimal product", math.exp(prob.value))
print("optimal x", x.value)
print("optimal y", y.value)
print("optimal z", z.value)
```

הרצאה 3 – שידוך עם משקל מקסימלי
ואז פלט aya בסלון וכו'

```python
import networkx as nx


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=25)
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=60)
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=25)

print("Maximum-value matching: ", nx.max_weight_matching(G))
```



הרצאה 3 מציאת מחיר לחדר

aya צריכה לא לקנא

```python
import cvxpy
from scipy.optimize import linprog
from timeit import timeit


m, h, s = cvxpy.Variable(),cvxpy.Variable(), cvxpy.Variable()
prob = cvxpy.Problem(
    cvxpy.Minimize(0),
    constraints = [m+h+s==100,
        m >= 0, h >= 0, s >= 0,
        35-s >= 40-h, 35-s >= 25-m,   # aya
        60-h >= 35-s, 60-h >= 40-m,   # batya
        20-m >= 40-h, 20-m >= 25-s,   # gila
        ]
)
print(timeit(lambda: prob.solve(), number=100))  # in seconds
```

```python
# Order the buyers from high value to low value
buyers.sort(key = lambda buyer: buyer.value, reverse = True)
# check length arr
if len(buyers) >= 2:
```

הרצאה 7 - ערך וירטואלי

```python
class Buyer:
    def __init__(self, val, name, age):
        self.value = val
        self.name = name
        self.age = age
        self.support = 100 if age<20 else (150 if age>60 else 0)
        self.virtual_value = 2*val-1000
        self.expected_profit = self.virtual_value + self.support

    def threshold_value(self, threshold_profit):
        return (threshold_profit + (1000-self.support))/2
```

sample exam

```python
class Buyer:
  def __init__(self,val,name):
    self.value = val
    self.name = name

def sellHouse(buyers:list):
  # Find the buyer with the largest value
  max_value_buyer = max(buyers, key = lambda: buyer: buyer.value)
  if max_value_buyer >= 500: # 500 is the Ereh Asaf (v)
    print(max_value_buyer.name + " gets the house")
  else:
    print("no buyer gets the house")
```