יעילות פארטו

בשיעור הזה נלמד על שיקול נוסף שהוא חשוב מאד בכלכלה - **יעילות**.

מהי יעילות כלכלית? נסביר ע"י דוגמה.

שלושה אחים רוצים ללכת יחד למסעדה ומתלבטים באיזו מסעדה לבחור. כל אח מדרג את המסעדות מהכי גרועה בעיניו (1) להכי טובה בעיניו (5):

מסעדה:	**א**	**ב**	**ג**	**ד**	**ה**
עמי:	1	2	3	4	5
תמי:	3	1	2	5	4
רמי:	3	5	5	1	1

איזו מסעדה הכי טובה? זה תלוי - מסעדה ה הכי טובה עבור עמי, מסעדה ד הכי טובה עבור תמי, ומסעדות ב,ג הכי טובות עבור רמי.

אבל יש מסעדה אחת שאפשר להגיד עליה בוודאות שהיא "לא טובה". מה היא?

תשובה: מסעדה ב! למה? כי מסעדה ג יותר טובה ממסעדה ב עבור עמי ותמי, ומבחינת רמי הן אותו הדבר. אז אין שום סיבה שהאחים יבחרו את מסעדה ב - זה יהיה **לא יעיל**.

עכשיו נגדיר פורמלית את תכונת היעילות. התכונה הוגדרה פורמלית לראשונה ע"י וילפרדו פארטו (Pareto) - כלכלן איטלקי שחי לפני כ-100 שנה. אולם הרעיון קיים כבר בתלמוד בעיקרון "זה נהנה וזה לא חסר". הנה ההגדרות:

- מצב א נקרא **שיפור פארטו** (Pareto improvement) של מצב ב, אם מצב א טוב יותר לחלק מהמשתתפים ("זה נהנה"), וטוב לפחות באותה מידה לכל השאר ("וזה לא חסר").
- מצב נקרא **יעיל פארטו** (Pareto efficient / Pareto optimal) אם לא קיים מצב אחר שהוא שיפור פארטו שלו.

עכשיו אפשר לבדוק ולראות בדוגמה למעלה, שכל המסעדות הן יעילות-פארטו חוץ ממסעדה ב.

העקרון של יעילות-פארטו לא אומר לנו מה בדיוק לבחור - הרי אין בחירה אחת שהיא "הכי טובה" עבור כולם. אבל יעילות-פארטו היא תנאי הכרחי לכך שהבחירה שלנו היא הגיונית. אם בחרנו באפשרות שהיא לא יעילה פארטו - כנראה לא התאמצנו מספיק כדי להשביע את רצונם של המשתתפים.

אגב, המושג של יעילות פארטו מסביר את ההבדל בין פוליטיקאי לבין כלכלן: פוליטיקאי אמור לשרת אינטרסים של קבוצה מסויימת גם אם זה בא על-חשבון אחרים - לשם כך הוא נבחר. התפקיד של כלכלן הוא להציע שיפורי פארטו - להציע שיפורים להתנהלות המשק, שישפרו את המצב של כל האוכלוסיה (או לפחות ישפרו את המצב של חלק מהאוכלוסיה בלי לפגוע באחרים).

יעילות פארטו בחלוקת קרקעות ועוגות

עכשיו נחזור לבעיית חלוקת העוגה (והקרקע). האם האלגוריתמים לחלוקה הוגנת מחזירים חלוקה שהיא יעילה-פארטו? לצורך הדיון נסתכל על האלגוריתם הפשוט ביותר - "חתוך ובחר" - לחלוקת עוגה בין שני אנשים. האם האלגוריתם הזה תמיד מחזיר חלוקה יעילה פארטו?

התשובה היא לא!

הנה דוגמה. נניח שב"עוגה" יש ארבעה איזורים, והשחקנים מעריכים אותם לפי הטבלה הבאה:

איזור:	**א**	**ב**	**ג**	**ד**
עמי:	2	0	0	2
תמי:	1	1	1	1

נניח שעמי חותך את העוגה בדיוק באמצע - בין פרוסה ב לפרוסה ג.

תמי בוחרת את אחת הפרוסות, נניח את הפרוסה הימנית.

הערך של עמי הוא 2 ושל תמי גם 2.

זה לא יעיל! כי אילו עמי היה חותך בין א ל-ב, או בין ג ל-ד, אז הערך שלו היה עדיין 2, אבל הערך של תמי היה עולה ל-3 - זה היה שיפור פארטו - "זה נהנה וזו לא חסרה"!

אמנם במקרים מסויימים האלגוריתם הזה כן יעיל פארטו.

משפט: אלגוריתם "חתוך ובחר" מחזיר תוצאה יעילה פארטו אם מתקיימים התנאים הבאים:

1. העוגה חד-ממדית.
2. שני השחקנים רוצים רק חתיכות קשירות.
3. לכל נקודה בעוגה יש ערך חיובי ממש.
4. החותך חותך לשני חלקים שוים ולא "מתחכם".

הוכחה:
לפי תנאים 1+2, יש רק שתי אפשרויות: או שהחותך משמאל והבוחר מימין, או הפוך.
לפי תנאי 3, בסדר שנבחר, אין שיפור פארטו.
לפי תנאי 4, גם בסדר ההפוך אין שיפור פארטו. ***

אבל מה קורה אם השחקנים לא מעוניינים דווקא בחתיכות קשירות?
לדוגמה, נניח שה"עוגה" היא לא עוגה פיסית, אלא אוסף של משאבי-מיחשוב שצריך לחלק בין כמה משתמשים בשרת מרכזי, ולכל אחד מהמשתמשים ישנן העדפות שונות לכל אחד מהמשאבים:

משאב:	**דיסק**	**זיכרון**	**מעבד**
עמי:	0	19	81
תמי:	20	0	80

במקרה הזה, אין משמעות לקשירות של הפרוסות - הדבר היחיד שמשנה הוא, איזה כמות מכל אחד מהמשאבים מקבל כל אחד מהשחקנים.

במקרה זה, אלגוריתם "חתוך ובחר" לא בהכרח מחזיר חלוקה יעילה פארטו.

האם קיים אלגוריתם שתמיד מחזיר חלוקה יעילה פארטו?

ואיך בכלל מוצאים חלוקות יעילות פארטו?

דרך אחת למצוא חלוקות יעילות-פארטו היא לפתור בעיית אופטימיזציה. לבעיות אופטימיזציה ישנם שימושים רבים בהמון תחומים, וביניהם גם בכלכלה. יש גם הרבה כלי-תוכנה שונים המאפשרים לפתור בעיות אופטימיזציה ביעילות - נראה אחד מהם בהמשך השיעור.

הנה בעיית אופטימיזציה פשוטה הקשורה לחלוקת עוגה:

- מצא את החלוקה שבה סכום הערכים של השחקנים הוא מקסימלי.

בניסוח מתימטי:

Maximize $V_1(X_1) + V_2(X_2)$
such that (X_1,X_2) is a partition

הבעיה הזאת ניתנת לפתרון בקלות במקרה של חלוקת עוגה: פשוט נותנים כל איזור מהעוגה לשחקן שעבורו הערך של האיזור הזה הוא הכי גדול. בדוגמה למעלה - כל הדיסק לתמי, כל הזיכרון לעמי, וכל המעבד לעמי. זה נותן את סכום הערכים המקסימלי - 120.

משפט: כל חלוקה הממקסמת את סכום הערכים 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. ***

אז מצאנו חלוקה יעילה - אבל היא בבירור לא הוגנת: עמי מקבל הרבה ותמי מקבלת מעט מאד. החלוקה לא פרופורציונלית ויש בה קנאה.

אנחנו צריכים למצוא חלוקה יעילה אחרת, שתהיה גם הוגנת. איך מוצאים חלוקות יעילות אחרות? - אפשר לפתור בעיית אופטימיזציה אחרת. תהי f פונקציה כלשהי שהיא מונוטונית-עולה. נתבונן בבעיית האופטימיזציה הבאה:

Maximize $f(V_1(X_1)) + f(V_2(X_2))$
such that (X_1,X_2) is a partition

הבעיה מאד דומה לבעיה הקודמת, רק שהפעם, לפני שחישבנו את סכום הערכים, הפעלנו על כל אחד מהערכים את הפונקציה f. גם הבעיה הזאת נותנת לנו חלוקה יעילה פארטו.

משפט: כל חלוקה הממקסמת פונקציה עולה כלשהי של סכום הערכים היא יעילה פארטו.
הוכחה: נתונה חלוקה א הממקסמת את הסכום הנתון.
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 f היא מונוטונית-עולה, בחלוקה ב הסכום גבוה יותר – בסתירה לכך שחלוקה א ממקסמת את הסכום. ***

איך אפשר לפתור את בעיית האופטימיזציה הנ"ל? יש כמה דרכים.
דרך אחת היא לכתוב פונקציה בכמה משתנים, לגזור אותה ולמצוא נקודת מקסימום - כמו שלמדתם בקורס חדו"א. בדוגמה למעלה, ברור שתמי תקבל את כל הדיסק ועמי יקבל את כל הזיכרון, אז מספיק משתנה אחד - x - מספר בין 0 ל-1 המייצג את אחוז זמן-המעבד שיקבל עמי.
במקרה זה, הערך של עמי הוא 81x+19 והערך של תמי הוא 80-80x+20. נניח שהפונקציה f היא שורש-ריבועי. אז בעיית האופטימיזציה היא:

Maximize sqrt(81x+19) + sqrt(80-80x+20)
such that 0 <= x <= 1

עכשיו אפשר לגזור ולמצוא נקודת מקסימום.

אבל, בקורס הזה נפתור את הבעיות האלו בצורה אחרת - נשתמש בתוכנות מתמטיות היודעות לפתור בעיות כאלו באופן אוטומטי. למשל, בתוכנת Mathematica אפשר לכתוב את הפקודה הבאה:

```
FindMaximum[{(81 x + 19)^0.5 + (80 (1 - x) + 20)^0.5, 0 <= x <= 1}, x]
```

ומקבלים מייד את התשובה:

```
{15.4601, {x -> 0.512327}}
```

משמעות התשובה: עמי צריך לקבל בערך 51.23% מזמן-המעבד, ותמי צריכה לקבל את השאר. במקרה שלנו הפתרון אכן נותן חלוקה הוגנת - הערך של כל אחד מהשחקנים הוא מעל 50%.

אבל האם זה יהיה כך בכל מקרה? לא בטוח. אנחנו יכולים, בכל בעיה בנפרד, לנסות כל מיני פונקציות f, לפתור את בעיות האופטימיזציה המתאימות, ולבדוק את החלוקה יעילה. אבל יש דרך טובה יותר. מסתבר שיש פונקציה אחת f, שאם ממקסמים את הסכום שלה, מקבלים חלוקה שהיא **תמיד** ללא קנאה - זו הפונקציה הלוגריתמית. כלומר פתרון הבעיה הבאה נותן חלוקה שהיא גם יעילה-פארטו וגם ללא קנאה:

Maximize $\quad \text{sum}_i \log(V_i(X_i))$
such that $\quad (X_1,\ldots,X_n)$ is a partition

משפט: כל חלוקה הממקסמת את סכום לוגי הערכים היא חלוקה ללא קנאה.
הוכחה: נסתכל בפרוסת עוגה אינפיניטיסימלית, Z.
התרומה שלה ל-$f(V_j(X_j))$ היא הערך של Z כפול הנגזרת של f בנקודה הנוכחית:

$$f'(V_j(X_j)) * V_j(Z)$$

לכן, אלגוריתם האופטימיזציה ייתן כל פרוסה Z לשחקן j שהמכפלה הזאת עבורו גדולה ביותר:

$$f'(V_j(X_j)) * V_j(Z) \geq f'(V_i(X_i)) * V_i(Z)$$

נסכם את המשוואה על כל הפרוסות שניתנו ל-j:

$$f'(V_j(X_j)) * V_j(X_j) \geq f'(V_i(X_i)) * V_i(X_j)$$

כאשר f היא פונקציה לוגריתמית, מקבלים:

$$(1 / V_j(X_j)) * V_j(X_j) \geq (1 / V_i(X_i)) * V_i(X_j)$$

מעבירים אגף ומקבלים, לכל שני שחקנים j,i:

$$V_i(X_i) \geq V_i(X_j)$$

וזו בדיוק ההגדרה של חלוקה ללא קנאה! ***

המשפט הזה נותן לנו אלגוריתם פשוט למציאת חלוקה "מושלמת" - גם הוגנת וגם יעילה. במקרה שלנו אפשר למצוא חלוקה כזאת ע"י פקודה אחת בשפת Mathematica:

```
FindMaximum[{Log[81 x + 19] + Log[80 (1 - x) + 20], 0 <= x <= 1}, x]
```

והתשובה היא:

```
{8.18043, {x -> 0.507716}}
```

כלומר עמי צריך לקבל 50.77% מזמן-המעבד ותמי את השאר.

חלוקה קשירה, יעילה וללא קנאה

למדנו שתמיד קיימת חלוקה שהיא **ללא-קנאה וקשירה**, וחלוקה שהיא **ללא-קנאה ויעילה**.

קל למצוא חלוקה שהיא **קשירה** וגם **יעילה** (איך?).

האם תמיד קיימת חלוקה שיש לה כל שלוש התכונות - **ללא-קנאה, קשירה, ויעילה** בו-זמנית?

התשובה היא לא. הנה דוגמה: נתונה עוגה חד-מימדית המחולקת לשבעה איזורים סמוכים. הערכים של עמי, תמי וצומי לכל איזור נתונים בטבלה למטה. כל אחד מהשחקנים רוצה חתיכה קשירה בלבד.

עמי	2	0	3	0	2	0	0
תמי	0	0	0	0	0	7	0
צומי	0	2	0	2	0	0	3

בכל חלוקה הוגנת, תמי חייבת לקבל את הפרוסה השישית (או חלק ממנה). עכשיו יש שתי אפשרויות:

- עמי מקבל פרוסה מימין לתמי, וצומי מקבלת פרוסה משמאל לתמי. אז צומי מקנאת בעמי.
- עמי וצומי שניהם מקבלים פרוסה מימין לתמי. אם צומי מקבלת 2+2 אז עמי מקנא; אם צומי מקבלת רק 2, אז החלוקה לא יעילה פארטו - אפשר להשיג שיפור פארטו ע"י העברת החלק של צומי לצד שמאל.

כאן אנחנו נתקלים בתופעה שחוזרת על עצמה הרבה בכלכלה (וגם בתחומים אחרים בחיים): יש שלוש דרישות, שאי-אפשר למלא בו-זמנית. צריך לבחור על מה לוותר: האם לוותר על קשירות? על הגינות? על יעילות? התשובה תלויה בבעיה ומשתנה בכל מקרה לגופו.

לסיום, דוגמה מהחיים לסתירה בין הגינות ליעילות. כשתיכננתי את המבחן לקורס אלגוריתמים בשנת ה'תשע"ח, רציתי לתת לסטודנטים הרבה זמן, כך שלא יהיו לחוצים. אז כתבתי בבחינה "משך הבחינה - 4 שעות". אבל מדור בחינות פסל את הבחינה בטענה שהמקסימום הוא 3 שעות. כששאלתי: למה? ענו לי: כי ביממה יש 12 שעות עבודה, ויש 3 בחינות ביום. אמרתי: מעולה, ב-12 שעות יש מקום בדיוק ל-3 בחינות של 4 שעות! אז הם הסבירו לי, שיש סטודנטים עם אישור רפואי להארכת זמן, ומגיע להם לקבל 25% יותר זמן מכל שאר הכיתה. אז אם כולם מקבלים 3 שעות, הם מקבלים 3:45 וזה מסתדר, אבל אם כולם מקבלים 4 שעות, הם יצטרכו לקבל 5 שעות וזה כבר לא מסתדר.

יש כאן סתירה בין הגינות ליעילות: ההגינות מחייבת לתת לסטודנטים עם אישור את הארכת-הזמן המגיעה להם. אבל כתוצאה מכך נוצר מצב שהוא לא יעיל פארטו (עבור הסטודנטים), שהרי היה עדיף לכולם - גם לסטודנטים עם אישור וגם לסטודנטים בלי אישור - שהבחינה תהיה 4 שעות לכולם.

במקרה זה, הנהלת האוניברסיטה העדיפה את ההגינות על-פני היעילות. מה קורה במקרים אחרים?

מאמרים להרחבה ולמטלת רשות

1. Anna Bogomolnaia , Hervé Moulin , Fedor Sandomirskiy , Elena Yanovskaya (2017): "Competitive division of a mixed manna"
2. Walter Stromquist (2007), "A Pie That Can't Be Cut Fairly"
3. Dall'Aglio, M. (2001): "The Dubins–Spanier optimization problem in fair division theory". J. Comput. Appl. Math. 130(1–2), 17–40
4. Dall'Aglio, M., Hill, T.P. (2003): "Maximin share and minimax envy in fair-division problems". J. Math. Anal. Appl.
5. D Kurokawa, AD Procaccia, N Shah (2018), "Leximin Allocations in the Real World".
6. Y Chen, JK Lai, DC Parkes, AD Procaccia (2013): "Truth, justice, and cake cutting".

סיכם: אראל סגל-הלוי.