יעילות כלכלית

"אַל תִּמְנַע טוֹב מִבְּעָלָיו, בִּהְיוֹת לְאֵל יָדְךָ לַעֲשׂוֹת" (משלי ג כז)

בשיעור זו נלמד שיקול נוסף בחלוקת משאבים, שאנו רוצים להשיג במקביל להגינות, והוא יעילות במובן הכלכלי (להבדיל מיעילות חישובית). נסביר את מושג היעילות הכלכלית על-ידי דוגמה. נניח שעמי ותמי רוצים לחלוק ביניהם קרקע. עמי מייחס ערך 99 לחצי המזרחי של הקרקע ו-1 לחלקה המערבי; תמי מייחסת ערך 99 לחלק המערבי של הקרקע ו-1 לחלקה המזרחי. לשניהם פונקציות-ערך אדיטיביות. נניח שאנחנו נותנים לעמי את החצי הצפוני של הקרקע ולתמי את החצי הדרומי.

האם החלוקה היא פרופורציונלית? - כן: כל אחד מהשחקנים קיבל ערך של 50 – בדיוק ½ מערך הקרקע כולה. אבל, ברור לנו שהחלוקה הזאת לא מספיק טובה - יש בה בזבוז. אם ניתן לעמי את החלק המזרחי ולתמי את החלק המערבי, הערך של שניהם יהיה גבוה יותר; לכן החלוקה הראשונה אינה יעילה. ביחידה זו נראה כמה הגדרות למושג היעילות, ונלמד אלגוריתמים המוצאים חלוקה שהיא גם יעילה וגם הוגנת.

אנחנו נגדיר את מושגי היעילות באופן כללי, אבל נתמקד בבעיית-חלוקה של משאבים הומוגניים (homogeneous resources). ישנם כמה הבדלים בין בעיה זו לבין בעיית חלוקת העוגה:

- מספיק להחליט כמה מכל משאב מקבל כל שחקן; אין זה משנה איזה חלק מהמשאב הוא מקבל.
- אין חשיבות לתכונת הקשירות.

בנוסף, אנו מניחים ש:

- המשאבים רציפים, כלומר, ניתן לתת לכל שחקן אחוז כלשהו מכל משאב.
- לכל שחקן ישנה פונקציית-ערך רציפה: תוספת מעט משאב משפיעה רק במעט על הערך.
- פונקציות הערך הן אדיטיביות ומונוטוניות-עולות.

יעילות פארטו

תכונת היעילות היסודית הוגדרה פורמלית לראשונה ע"י וילפרדו פארטו (Vilfredo Pareto) - כלכלן איטלקי שחי לפני כ-100 שנה, ולכן היא נקראת על שמו "יעילות פארטו". אולם, הרעיון שמאחריה נזכר כבר בתלמוד (בבא קמא כ ב), בכלל "זה נהנה וזה לא חסר".

הגדרה.

- 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* או *Pareto optimal*) אם לא קיים מצב אחר שהוא שיפור פארטו שלו.

בדוגמה למעלה, החלוקה הנותנת את החצי הצפוני של הקרקע לעמי ואת החלק הדרומי לתמי אינה יעילה-פארטו, כיוון שהחלוקה הנותנת את החצי המזרחי לעמי ואת החלק המערבי לתמי היא שיפור-פארטו שלה.

שימו לב, ההגדרות מתייחסות לא רק לחלוקת-משאבים, אלא ל"מצב" מופשט כלשהו. חלוקת-משאבים היא מקרה פרטי של מצב. כדוגמה למצב מסוג אחר, נניח שעמי ותמי רוצים ללכת למסעדה, וצריכים להחליט לאיזו מסעדה ללכת. יש שלוש אפשריות: עמי מעדיף את א על ב ואת ב על ג; תמי מעדיפה את ב על א ואת א על ג. אנחנו לא יודעים במה הם יבחרו בסוף, אבל ברור לנו שמצב אחד אינו יעיל פארטו, והוא ללכת למסעדה ג. זאת, כיוון שבחירה בכל אחת מהמסעדות האחרות היא שיפור-פארטו של ג.

העקרון של יעילות-פארטו אינו 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 שאינה יעילה פארטו - כנראה לא התאמצנו מספיק כדי להשביע את רצונם של המשתתפים. מצד שני, אם הצלחנו למצוא שיפור פארטו, עשינו עבודה טובה במיוחד – הצלחנו לשפר את מצבם של חלק מהמשתתפים בלי לעורר התנגדות מצד האחרים.

יעילות פארטו בחלוקת קרקעות ועוגות

עכשיו נחזור לאלגוריתמי חלוקת העוגה (והקרקע) שלמדנו בשיעור הקודם. האם האלגוריתמים האלו מחזירים חלוקה שהיא יעילה-פארטו? לצורך הדיון נסתכל על אלגוריתם "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-פארטו, 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".

כדוגמה נוספת, נתבונן בבעייה של חלוקת משאבים הומוגניים, למשל, חלוקת סחורות וחומרי-גלם בין שני שותפים במפעל תעשייה. נניח שלכל אחד מהמשתמשים ישנן העדפות שונות לכל אחד מהמשאבים:

משאב:	**ברזל**	**דלק**	**עצים**
עמי:	1	19	80
תמי:	29	1	70

שימו לב, במקרה הזה אין חשיבות לכך שהפרוסות תהיינה קשירות. הדבר היחיד שמשנה הוא, איזה כמות מכל אחד מהמשאבים מקבל כל אחד מהשחקנים. אלגוריתם "חתוך ובחר" מחזיר חלוקה שאינה יעילה-פארטו, גם כשעמי חותך ותמי בוחרת, וגם כשתמי חותכת ועמי בוחר (בדקו ותראו).

האם קיים אלגוריתם המחזיר תמיד חלוקה יעילה-פארטו?

- התשובה היא כן, יש הרבה אלגוריתמים כאלה. האלגוריתם הפשוט ביותר שעושה זאת נקרא אלגוריתם הדיקטטורה (dictatorship). בדוגמה שלמעלה, האלגוריתם בוחר את אחד מהשחקנים באופן שרירותי (כגון: לפי גיל, או באקראי), ונותן לו את כל המשאבים.

שאלה. הוכיחו, שאם הדיקטטור מייחס ערך חיובי לכל המשאבים, אז אלגוריתם הדיקטטורה מחזיר חלוקה יעילה-פארטו.

תשובה. אם הדיקטטור מייחס ערך חיובי לכל המשאבים, אז כל חלוקה אחרת מורידה את הערך שבידי הדיקטטור, ולכן כל חלוקה אחרת אינה שיפור-פארטו. ***

אם ישנם משאבים שהדיקטטור מייחס להם ערך אפס, אז אלגוריתם הדיקטטורה – כפי שתואר למעלה – אינו מחזיר חלוקה יעילה-פארטו, אבל אפשר בקלות לתקן אותו: נותנים לדיקטטור את כל המשאבים שהוא מייחס להם ערך חיובי, ממנים דיקטטור חדש, ונותנים לו לבחור – מבין המשאבים שנשארו – את כל המשאבים שהוא מייחס להם ערך חיובי. ממשיכים כך עד שכל המשאבים נלקחו. אלגוריתם זה נקרא דיקטטורה סדרתית (באנגלית: serial dictatorship).

אלגוריתם הדיקטטורה הסדרתית הוא האלגוריתם הכי לא-הוגן שאפשר לדמיין: בדוגמה למעלה, הוא נותן את כל המשאבים לשחקן אחד, ולא נותן כלום לשחקן השני. האם קיים אלגוריתם המחזיר חלוקה יעילה-פארטו שהיא גם הוגנת?

לפני שנענה לשאלה, נתעמק יותר במושג היעילות, ונראה כמה מושגים נוספים של יעילות, שהם חזקים יותר מיעילות-פארטו.

יעילות אוטיליטרית

אוטיליטריות (*utilitarianism*) היא גישה שפותחה ע"י הפילוסופים ירמיהו בנתהאם (Jeremy Bentham) וג'ון סטיוארט מיל (John Stuart Mill). לפי גישה זו, הרווחה החברתית (באנגלית: social welfare) היא סכום הערכים של כל המשתתפים, ולכן המצב הטוב ביותר לחברה הוא המצב הממקסם את סכום הערכים.

הגדרה. מצב א נקרא אוטיליטרי (*utilitarian*) אם סכום הערכים של כל השחקנים במצב א גדול לפחות כמו סכום הערכים של כל השחקנים בכל מצב אחר.

כמו בסעיף הקודם, גם כאן אנחנו מגדירים את היעילות האוטיליטרית עבור כל "מצב" מופשט כלשהו; חלוקת-משאבים היא מקרה פרטי של מצב.

בדוגמת חלוקת הסחורות, החלוקה האוטיליטרית נותנת את כל הברזל לתמי, את כל הדלק לעמי, וגם את כל העצים לעמי. בחלוקה זו, סכום הערכים הוא 128. כדי לראות שזהו אכן הסכום הגדול ביותר האפשרי, נבדוק מה יקרה אם נעביר משאב כלשהו (או חלק ממנו) מעמי לתמי: הערך של עמי יקטן, והערך של תמי יגדל בשיעור מועט יותר, ולכן הסכום יקטן. באותו אופן, אם נעביר משאב כלשהו מתמי לעמי, אז הערך של תמי יקטן, והערך של עמי יגדל בשיעור מועט יותר, ולכן הסכום יקטן.

הכלל האוטיליטרי קובע, שכדאי להקטין במעט את הערך של אדם אחד, על-מנת להגדיל בהרבה את הערך של אדם אחר, כיוון שהדבר יביא להגדלת סכום הערכים.

הכלל האוטיליטרי הגיוני רק כאשר ישנה דרך אובייקטיבית למדוד את הערך של כל אחד מהמשתתפים, כך שכל הערכים נמדדים באותן יחידות. הנה כמה דוגמאות:

- השחקנים הם חברות מסחריות, המשאבים הם חומרי-גלם, והערך של כל שחקן הוא הרווח הכספי שהוא יכול להפיק מאותם חומרי-גלם, בשקלים. חלוקה אוטיליטרית ממקסמת את הרווח הכולל (כתוצאה מזה, היא גם ממקסמת את תקבולי המס של המדינה).
- השחקנים הם חולים במחלה מסויימת, המשאבים הם תרופות, והערך של כל שחקן הוא ההסתברות שיחלים כתוצאה מקבלת התרופה. חלוקה אוטיליטרית ממקסמת את סכום הסתברויות ההחלמה, שהוא התוחלת של מספר המחלימים.

בנתהאם ומיל טענו שאפשר להחיל את העקרון האוטיליטרי גם במצבים שבהם ה"ערך" של אדם נקבע על-פי רמת ההנאה או הכאב הסובייקטיביים שהוא מרגיש בכל מצב. לשם כך הם פיתחו שיטה בשם "חשבון הדוני" (hedonic calculus), למדידת עוצמת ההנאה או הכאב שמרגיש כל אדם בכל מצב; שיטה זו היא מעבר לנושא הקורס הנוכחי.

יעילות אוטיליטרית ויעילות פארטו

ראינו שני סוגי יעילות – של פארטו ושל האוטיליטריסטים. המשפט הבא מראה, שיעילות אוטיליטרית היא תנאי חזק יותר מיעילות פארטו. המשפט מנוסח עבור מצב כלשהו; הוא נכון בפרט עבור חלוקת משאבים.

משפט. כל מצב אוטיליטרי הוא יעיל-פארטו.

הוכחה. נתון מצב א שהוא אוטיליטרי – כלומר ממקסם את סכום הערכים. נניח בשלילה שהמצב לא יעיל פארטו. אז קיים מצב ב שהוא שיפור-פארטו של א. במצב ב, הערך של כל השחקנים גדול לפחות כמו במצב א, ולשחקן אחד לפחות יש ערך גדול יותר. לכן, במצב ב סכום הערכים גדול יותר – בסתירה לכך שמצב א הוא אוטיליטרי. ***

חישוב חלוקה אוטיליטרית

איך מוצאים חלוקה אוטיליטרית? כאשר פונקציות-הערך הן אדיטיביות, ויש מספר סופי של משאבים (או לחלופין: עוגה עם מספר סופי של איזורים), ישנו אלגוריתם פשוט המוצא חלוקה אוטיליטרית:

תן כל אחד מהמשאבים לשחקן שעבורו הערך של האיזור הזה הוא הכי גדול.

כאשר פונקציות-הערך אינן אדיטיביות, האלגוריתם שהוצג למעלה לא תמיד מוצא חלוקה אוטיליטרית. לדוגמה, נניח שיש שתי מכוניות. עמי מייחס לכל אחת מהן ערך 100, אבל מתייחס אליהן כמוצרים חליפיים, ולכן מייחס לשתיהן יחד ערך 120. תמי מייחסת לכל מכונית ערך 90, וגם היא מתייחסת אליהן כמוצרים חליפיים, ולכן מייחסת לשתיהן יחד ערך 110. האלגוריתם שתואר למעלה נותן את שתי המכוניות לעמי, אבל הפתרון האוטיליטרי כאן הוא לתת מכונית אחת לכל אחד – סכום הערכים במקרה זה הוא 190.

כדי למצוא חלוקה אוטיליטרית במקרה הכללי, אנחנו צריכים לפתור בעיית אופטימיזציה. בעיית האופטימיזציה המתאימה לחלוקה אוטיליטרית של משאב כלשהו C בין n שחקנים היא:

Maximize $v_1(X_1) + \ldots + v_n(X_n)$

such that $(X_1,\ldots,X_n)$ is a partition of C

בקורס הנוכחי לא נלמד אלגוריתמים כלליים לפתרון בעיות אופטימיזציה; נושא זה נלמד בהרחבה בקורסים אחרים. כיום ניתן למצוא, ברוב שפות-התיכנות המובילות, ספריות הפותרות בעיות אופטימיזציה כלליות. אנחנו נראה דוגמה לספריה אחת כזאת בשפת פייתון – הספריה cvxpy. ספריה זו פותרת בעיות של אופטימיזציה קמורה (convex optimization), כלומר, מציאת מינימום של פונקציה קמורה בתחום קמור. משפט ידוע בתורת האופטימיזציה קובע, שלכל פונקציה קמורה ישנה רק נקודות מינימום אחת בכל תחום קמור (חישבו למשל על הפונקציה $y=x^2$ בקטע [1,1-]), וישנם אלגוריתמים יעילים למציאת נקודה זו. אותו הדבר נכון לגבי מציאת מקסימום של פונקציה קעורה בתחום קמור; זאת, כיוון שכל פונקציה קעורה היא פונקציה קמורה עם סימן מינוס (חישבו למשל על הפונקציה $y=-x^2$ בקטע [1,1-]).

בספריה cvxpy, מציאת החלוקה האוטיליטרית בדוגמת חלוקת הסחורות תתבצע כך:

```
import cvxpy


x, y, z = cvxpy.Variable(3)
```

```
utility_ami = x*80 + y*19 + z*1
utility_tami = (1-x)*70 + (1-y)*1 + (1-z)*29


prob = cvxpy.Problem(
    cvxpy.Maximize(utility_ami + utility_tami),
    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("Fractions given to Ami: ", x.value, y.value, z.value)
print("Utility of Ami", utility_ami.value)
print("Utility of Tami", utility_tami.value)
```

מגדירים שלושה משתנים x, y, z המציינים איזה חלק מהמשאבים (עצים, דלק, ברזל בהתאמה) ניתן לעמי. החלק שניתן לתמי מכל משאב הוא $1-x, 1-y, 1-z$ בהתאמה. מחשבים את התועלת של עמי ותמי במשתני-עזר. מגדירים בעיית מקסימיזציה של סכום התועלות. מגדירים את האילוצים, והם, שערכי כל המשתנים הם בין 0 ל-1. שימו לב שהפונקציה שאנחנו ממקסמים היא פונקציה ליניארית, ולכן פונקציה קעורה. התחום המוגדר על-ידי האילוצים הוא קוביה תלת-ממדית, שהיא תחום קמור. לכן מדובר בבעיית אופטימיזציה קמורה, שניתן לפתור בקלות בעזרת cvxpy.

הקוד למעלה מחשב את התועלת של עמי ותמי בהנחה שהן אדיטיביות, אבל באותו אופן יכולנו להחליף את החישוב בכל פונקציה קעורה אחרת.

מה עושים כשיש יותר משני שחקנים? באופן כללי, בבעיית חלוקה עם n שחקנים ו-m משאבים רציפים, כל חלוקה אפשרית מוגדרת ע"י מטריצה בעלת n שורות ו-m עמודות: המספר בשורה i ובעמודה j מייצג את החלק של משאב j שנמסר לשחקן i. האילוצים הם:

- כל מספר במטריצה צריך להיות בין 0 ל-1; אילוץ זה מבטא את העובדה, שאי-אפשר לתת לשחקן מסויים כמות שלילית ממשאב מסויים, או לתת לו יותר מהמשאב כולו.
- סכום המספרים בכל עמודה שווה ל-1; אילוץ זה מבטא את העובדה, שכל משאב צריך להיות מחולק כולו בין השחקנים.

פרטים נוספים על ספריית *cvxpy* והאפשרויות השונות שלה ניתן למצוא בתיעוד הספריה ברשת. בסעיף זה הבאנו את `cvxpy` רק כדי להראות שזה קל ופשוט להגדיר ולפתור בעיית אופטימיזציה קמורה, ולכן אפשר להתייחס לבעיות כאלו כאל אבני-בניין יסודיות בפיתוח אלגוריתמים.

יעילות אוטיליטרית והגינות

האלגוריתם האוטיליטרי נראה, במבט ראשון, הוגן יותר מאלגוריתם הדיקטטורה: הוא מתייחס לכל השחקנים באופן שיוויוני, ולא נותן לאף אחד מהם מעמד מועדף. עם זאת, האלגוריתם לא תמיד מחזיר חלוקה הוגנת. בדוגמה של חלוקת-המשאבים שראינו למעלה, הערך של תמי הוא רק 29, בעוד שערך העוגה כולה בעיניה הוא 100. החלוקה אינה פרופורציונלית, ובנוסף יש בה קנאה. בסעיף הבא נראה עקרון אחר של יעילות, המשלב בתוכו גם שיקולי הגינות.

יעילות אגליטרית

אגליטריות (*egalitarianism*) היא גישה שפותחה ע"י הפילוסוף ג'ון רולס (John Rawls). לפי גישה זו, יש לדאוג קודם-כל לחוליות החלשות ביותר בחברה:

הגדרה. מצב א נקרא אגליטרי (*egalitarian*) אם הערך המינימלי של שחקן במצב א גדול לפחות כמו הערך המינימלי של שחקן בכל מצב אחר.

בדוגמת משאבי המיחשוב, בחלוקה אגליטרית, תמי מקבלת את כל הברזל (עם ערך 29), עמי מקבל את כל הדלק (עם ערך 19), והם מתחלקים בעצים, כך שעמי מקבל 0.5333 ותמי מקבלת 0.4667. הערך של עמי הוא:

$$19 + 80*0.5333 = 61.667$$

והערך של תמי הוא

$$29 + 70*(1-0.5333) = 61.667$$

כך שהערך המינימלי הוא 61.667. ניתן לוודא שלא קיימת חלוקה עם ערך מינימלי גדול יותר: אם נעביר משאב כלשהו מתמי לעמי או הפוך, לאחד מהם יהיה ערך קטן יותר (בהמשך הסעיף נראה אלגוריתם לחישוב חלוקה אגליטרית).

בדוגמה למעלה, הערך של שני השחקנים שווה – בהתאם למשמעות המושג "אגליטרי" (= שיוויוני). זה נכון באופן כללי.

משפט. אם פונקציית-הערך של כל שחקן היא רציפה ומונוטונית-עולה-ממש, אז בכל חלוקה אגליטרית, כל השחקנים מקבלים את אותו ערך בדיוק.

הוכחה. נניח בשלילה שבחלוקה אגליטרית כלשהי, הערך המינימלי הוא t, אבל לאחד השחקנים (נניח לעמי) יש ערך גדול ממש מ-t. כיוון שפונקציית-הערך רציפה, ניתן לקחת כמות קטנה ביותר של משאבים מעמי, כך שהערך שלו עדיין יהיה גדול ממש מ-t. את המשאבים שלקחנו מעמי, ניתן לחלק באופן שווה בין n-1 השחקנים האחרים. כיוון שפונקציית-הערך של כל אחד מהם מונוטונית-עולה-ממש, הערך של כולם גדול יותר. בפרט, הערך של כל השחקנים לאחר ההעברה גדול ממש מ-t. זה סותר את ההנחה שהערך המינימלי בחלוקה האגליטרית הוא t. ***

אם פונקציות-הערך של השחקנים אינן מונוטוניות-עולות-ממש, אז המשפט אינו נכון. דוגמה פשוטה:

משאב:	ברזל	דלק
עמי:	100	0
תמי:	0	50

כאן הערך האגליטרי הגדול ביותר האפשרי הוא 50, כי אי אפשר לתת לתמי ערך גבוה יותר. אבל בחלוקה אגליטרית, עמי יכול לקבל עד 100.

יעילות אגליטרית ויעילות פארטו

חלוקה אגליטרית, כפי שהגדרנו אותה למעלה, אינה בהכרח יעילה פארטו. כך ניתן לראות מהדוגמה בסוף הסעיף הקודם: חלוקה אגליטרית יכולה לתת לתמי את הדלק וחצי מהברזל, ולעמי את כל שאר הברזל; בחלוקה זו לכל אחד מהשחקנים יש ערך 50, וזה הערך הגבוה ביותר שאפשר לתת לשניהם. אבל יש לה שיפור פארטו - פשוט לתת את כל הברזל לעמי.

הבעיה בכלל האגליטרי היא, שהוא מתייחס רק לערך הקטן ביותר, ומתעלם משאר הערכים. ניתן לתקן אותו באופן הבא:

הגדרה. מצב א נקרא לקסימין-אגליטרי (*leximin-egalitarian*) אם הוא ממקסם את הערך הקטן ביותר; בכפוף לזה, הוא ממקסם את הערך השני הכי קטן; בכפוף לזה, הוא ממקסם את הערך השלישי הכי קטן; וכן הלאה.

כדי לחשב חלוקה לקסימין-אגליטרית, יש לחשב קודם-כל את הערך המינימלי הגדול ביותר האפשרי; בדוגמת שני המשאבים, ערך זה הוא 50. עכשיו יש לחשב, מבין כל החלוקות שבהן הערך המינימלי הוא 50, באיזו חלוקה הערך השני-מלמטה הוא הכי גדול; בדוגמת שני המשאבים, ערך זה הוא 100. לכן חלוקה לקסימין-אגליטרית נותנת את כל הברזל לעמי ואת כל הדלק לעמי.

דרך אחרת להגדיר את עקרון הלקסימין היא: עבור כל מצב אפשרי, מסדרים את וקטור הערכים של השחקנים בסדר עולה – מהקטן לגדול. משווים בין הוקטורים המסודרים בסדר מילוני (=לקסיקוגרפי, מכאן השם לקסימין). בוחרים את הוקטור הגדול ביותר בסדר זה. לדוגמה, אם וקטור הערכים במצב א הוא (2, 5, 4) ובמצב ב (9, 2, 3), אז הוקטורים המסודרים הם (2, 4, 5) ו (2, 3, 9); הוקטור הראשון גדול יותר בסדר מילוני מהוקטור השני, ולכן מצב א נבחר.

הערה. בספרות, חלוקה לקסימין-אגליטרית נקראת לפעמים רק "לקסימין" ולפעמים רק "אגליטרית".

בדוגמת שני המשאבים, החלוקה הלקסימין-אגליטרית היא יעילה פארטו. הדבר נכון גם באופן כללי:

משפט. כל מצב לקסימין-אגליטרי הוא יעיל-פארטו.

הוכחה. נניח שמצב א הוא לקסימין-אגליטרי. נניח בשלילה שהוא לא יעיל פארטו. אז קיים מצב ב שהוא שיפור-פארטו של א. במצב ב, הערך של כל השחקנים גדול לפחות כמו במצב א, ולשחקן אחד לפחות יש ערך גדול יותר. לכן, וקטור הערכים המסודר במצב ב זהה לוקטור הערכים המסודר במצב א, פרט למספר אחד שהוא גדול יותר. לכן הוקטור של מצב ב גדול יותר בסדר מילוני מהוקטור של מצב א – בסתירה להנחה שמצב א הוא לקסימין-אגליטרי. ***

חישוב חלוקה אגליטרית

על-פי הגדרה, בעיית אופטימיזציה המוצאת חלוקה אגליטרית היא:

Maximize $\min(v_1(X_1), \dots, v_n(X_n))$

such that $(X_1,\dots,X_n)$ is a partition of C

אולם, כלים רבים לפתרון בעיות אופטימיזציה אינם מכירים את הפונקציה min. לכן יש לתאר את הבעיה בעזרת פונקציות פשוטות יותר. דרך אחת לעשות זאת היא להוסיף משתנה נוסף z, המייצג את הערך המינימלי; להוסיף אילוצים הקובעים ש- z הוא אכן הערך המינימלי, כלומר, ש- z קטן מכל n ערכי השחקנים; ואז למקסם את z:

Maximize z

such that $(X_1,\dots,X_n)$ is a partition of C;

$z \le v_1(X_1), \dots, z \le v_n(X_n)$

כדי לחשב חלוקה לקסימין-אגליטרית, דרוש אלגוריתם מורכב יותר. האלגוריתם המתואר למטה פותח ע"י סטפן וילסון ב-1998. נתאר אותו קודם על דוגמה פשוטה, ואז נציג אותו באופן כללי. הדוגמה היא חלוקה של שלושה משאבים בין ארבעה שחקנים עם הערכים הבאים:

	ברזל	דלק	עצים
א:	4	0	0
ב:	0	3	0
ג:	5	5	10
ד:	5	5	10

ראשית, נמצא את הערך המינימלי הגדול ביותר, על-ידי פתרון בעיית האופטימיזציה האגליטרית שהוצגה למעלה. בדוגמה, הערך המקסימלי של z בבעיית האופטימיזציה האגליטרית הוא 3. מכאן ניתן להסיק, שבחלוקה לקסימין-אגליטרית, יהיה שחקן כלשהו שהערך שלו יהיה 3, והערך של כל שאר השחקנים יהיה לפחות 3.

כעת, נבדוק אם יש שחקן שאפשר להגדיל את הערך שלו, אבל בלי לפגוע בערך המינימלי. לשם כך נחשב, עבור כל שחקן, את הערך המקסימלי שהוא יכול לקבל, תחת האילוץ שהערך של כל השחקנים הוא לפחות 3. לשם כך נפתור את הבעיה הבאה, עבור כל שחקן j:

Maximize $v_j(X_j)$

such that (X_1,X_2,X_3,X_4) is a partition of C;

$3 \le v_1(X_1),\ 3 \le v_2(X_2),\ 3 \le v_3(X_3),\ 3 \le v_4(X_4)$

עבור כל שחקן j, ישנם שני מקרים:

- מקרה א: הערך המקסימלי שהתקבל בבעיית המקסימיזציה של שחקן j הוא 3. נובע מכך, שבכל חלוקה לקסימין-אגליטרית, הערך של שחקן j יהיה בדיוק 3. אנחנו קוראים לשחקן כזה "שחקן רווי", כי הערך שלו הגיע לנקודת-רווייה.
- מקרה ב: הערך המקסימלי שהתקבל בבעיית המקסימיזציה של שחקן j גדול מ-3. במקרה זה, השחקן נשאר "שחקן חופשי".

בדוגמה, הערכים המקסימליים המתקבלים עבור שחקנים א, ב, ג, ד הם 4, 3, 8.25, 8.25 בהתאמה. לכן שחקן ב הופך לשחקן רווי, ושאר השחקנים נשארים חופשיים.

כעת, נְקַבֵּע את הערך של השחקנים הרוויים בנקודת-הרוויה שלהם, ונמצא חלוקה אגליטרית עבור שאר השחקנים, על-ידי פתרון בעיית האופטימיזציה האגליטרית. בדוגמה, הבעייה שנפתור תיראה כך:

Maximize z

such that (X_1,X_2,X_3,X_4) is a partition of C;

$v_2(X_2) = 3;$

$z \le v_1(X_1),\ z \le v_3(X_3),\ z \le v_4(X_4)$

בדוגמה, הערך המקסימלי של z בבעיית האופטימיזציה הזו הוא 4. מכאן ניתן להסיק, שבחלוקה לקסימין-אגליטרית, שחקן ב יקבל ערך 3, ויהיה שחקן כלשהו שיקבל ערך 4, וכל השאר השחקנים יקבלו לפחות 4. נמשיך באותו אופן, ונבדוק אם יש שחקן חופשי שאפשר להגדיל את ערכו, בלי לפגוע בערכים הנמוכים יותר. נפתור את הבעיה הבאה, עבור כל שחקן חופשי j:

Maximize $v_j(X_j)$

such that (X_1,X_2,X_3,X_4) is a partition of C;

$v_2(X_2) = 3;$

$4 \le v_1(X_1),\ 4 \le v_3(X_3),\ 4 \le v_4(X_4).$

הפעם, הערכים המקסימליים עבור שחקנים א, ג, ד הם 4, 6, 6 בהתאמה. לכן שחקן א הופך לשחקן רווי, ושאר השחקנים נשארים חופשיים.

באיטרציה השלישית, אנחנו מקבעים את הערך של השחקנים הרוויים א,ב, ופותרים את הבעיה:

Maximize z

such that (X_1,X_2,X_3,X_4) is a partition of C;

$v_2(X_2) = 3,\ v_1(X_1) = 4;$

$z \le v_3(X_3),\ z \le v_4(X_4).$

הערך המקסימלי של z הוא 5. נמשיך ונפתור את הבעיה הבאה, עבור כל שחקן חופשי j:

$$\begin{array}{ll} \text{Maximize} & v_j(X_j) \\ \text{such that} & (X_1,X_2,X_3,X_4) \text{ is a partition of } C; \\ & v_2(X_2) = 3,\ v_1(X_1) = 4; \\ & 5 \le v_3(X_3),\ 5 \le v_4(X_4). \end{array}$$

הפעם, הערכים המקסימליים עבור שחקנים ג, ד הם 5, 5 בהתאמה. לכן שני השחקנים הופכים להיות רוויים, והאלגוריתם מסתיים. החלוקה שנמצאה היא: הברזל לשחקן א, הדלק לשחקן ב, והעצים מתחלק שווה בשווה בין ג לבין ד.

פסאודו-קוד של האלגוריתם הכללי מתואר למטה.

1. **אתחול**: S := קבוצה ריקה (קבוצת השחקנים הרוויים).

2. פתור את בעיית האופטימיזציה הבאה:

$$\begin{array}{ll} \text{Maximize} & z \\ \text{such that} & (X_1,\dots,X_n) \text{ is a partition of } C; \\ & v_i(X_i) = \text{saturated\_value}[i] \quad \text{for all } i \text{ in } S; \\ & z \le v_j(X_j) \quad \text{for all } j \text{ not in } S. \end{array}$$

יהי z_{max} הערך המקסימלי שהתקבל בבעיה זו.

3. לכל שחקן חופשי j, פתור את בעיית האופטימיזציה הבאה:

$$\begin{array}{ll} \text{Maximize} & v_j(X_j) \\ \text{such that} & (X_1,\dots,X_n) \text{ is a partition of } C; \\ & v_i(X_i) = \text{saturated\_value}[i] \quad \text{for all } i \text{ in } S; \\ & z_{max} \le v_j(X_j) \quad \text{for all } j \text{ not in } S. \end{array}$$

אם הערך המתקבל שווה ל- z_{max}, אז j הוא שחקן רווי: הוסף אותו ל-S ושמור את ערך-הרוויה שלו:

- $\text{saturated\_value}[j]=z_{max}$.

4. אם כל השחקנים רוויים – סיים והחזר את החלוקה $(X_1,\dots,X_n)$. אחרת – חזור לשורה 2.

האלגוריתם שהוצג למעלה עובד כאשר מרחב החלוקות האפשריות הוא מרחב קמור. במרחב קמור, אם לוקחים וקטורים של ערכי-המשתנים המייצגים חלוקות אפשריות, ומחשבים ממוצע חשבוני שלהם, אז גם הוקטור המתקבל מייצג חלוקה אפשרית. כפי שלמדנו בפרק על יעילות אוטיליטרית, בבעיית חלוקה עם m משאבים רציפים, כל חלוקה אפשרית מוגדרת ע"י מטריצה בעלת n שורות ו-m עמודות, שבה כל

המספרים הם בין 0 ל-1, וסכום המספרים בכל עמודה הוא 1. קל להוכיח שמרחב המטריצות האלו הוא מרחב קמור. לכן המשפט הבא נכון במקרה זה.

משפט. כאשר מרחב החלוקות האפשריות הוא מרחב קמור, ולכל שחקן יש פונקציית-ערך אדיטיבית, האלגוריתם שהוצג למעלה מסתיים תוך n איטרציות לכל היותר.

הוכחה. מספיק להוכיח, שבכל איטרציה, לפחות אחד מהשחקנים החופשיים הופך להיות רווי. נניח בשלילה, שבאיטרציה מסויימת, אף שחקן חופשי לא נעשה רווי. נסמן ב-z_{max} את הערך של בעיית המקסימיזציה הראשונה באיטרציה זו, וב-f את מספר השחקנים החופשיים.

ישנם f שחקנים חופשיים, ואף אחד מהם לא נעשה רווי. המשמעות היא, שקיימות f חלוקות אפשריות, שבכל אחת מהן, כל השחקנים הרוויים מקבלים את ערך-הרוויה שלהם, ואחד מ-f השחקנים החופשיים מקבל ערך גדול ממש מ-z_{max}. ניקח את כל f החלוקות האלו, ונחשב את הממוצע החשבוני שלהן. כיוון שמרחב החלוקות האפשריות הוא קמור, החלוקה הממוצעת היא חלוקה אפשרית. מתכונת האדיטיביות נובע, שערכי השחקנים בחלוקה הממוצעת הם הממוצעים של ערכי השחקנים ב-f החלוקות. מכאן:

- בחלוקה הממוצעת, כל השחקנים הרוויים מקבלים את ערך-הרוויה שלהם;
- בחלוקה הממוצעת, כל השחקנים החופשיים מקבלים ערך גדול ממש מ- z_{max}.

אבל משתי הנקודות הללו נובע, שהערך של בעיית המקסימיזציה הראשונה באיטרציה זו היה צריך להיות גדול ממש מ- z_{max} – סתירה. ***

משפט. האלגוריתם שהוצג למעלה מחזיר חלוקה לקסימין-אגליטרית.

הוכחה. לכל איטרציה k, נסמן ב-z_k את הערך של בעיית המקסימיזציה הראשונה (הערך של z_{max}) באיטרציה זו, וב-S_k את קבוצת השחקנים הנעשים רוויים באיטרציה זו. בחלוקה שמחזיר האלגוריתם, כל השחקנים בקבוצה S_k מקבלים סל עם ערך z_k. שימו לב שהערך z_k עולה עם k, כיוון שבכל איטרציה יש פחות אילוצים על הערך של z בבעיית האופטימיזציה הראשונה (פחות שחקנים חופשיים).

נסמן ב-X את החלוקה המוחזרת מהאלגוריתם, נניח שהיא לא לקסימין-אגליטרית, ונסמן ב-Y חלוקה אחרת שהיא לקסימין-עדיפה על X. נסמן ב-k את האינדקס הקטן ביותר כך ששחקן מקבוצה S_k מקבל ערך שונה בשתי החלוקות. נסמן שחקן זה ב-j. כיוון שהחלוקה Y לקסימין-עדיפה על X, שחקן j בהכרח מקבל ערך גזול יותר בחלוקה Y מבחלוקה X, כלומר ערך גדול מ-z_k. מה אנחנו יודעים על ערכי השחקנים האחרים בחלוקה Y?

- כל השחקנים בקבוצות $S_1, \ldots, S_{k-1}$ מקבלים את אותו ערך בדיוק בשתי החלוקות – לפי ההנחה על k.
- כל השחקנים בקבוצות S_k, S_{k+1},..., מקבלים ערך לפחות z_k – כי החלוקה Y לקסימין-עדיפה על X.

נתבונן בבעיית האופטימיזציה השניה באיטרציה k:

```
Maximize     v_j(X_j)
such that    (X_1,…,X_n) is a partition of C;
             v_i(X_i) = saturated_value[i]  for all  i in S;
             z_k ≤ v_j(X_j)  for all j not in S.
```

נשים לב שבאיטרציה זו, הקבוצה S היא איחוד הקבוצות $S_1, ..., S_{k-1}$. לכן, החלוקה Y היא פתרון אפשרי של בעיית האופטימיזציה. מכאן, שהערך המקסימלי של בעיית האופטימיזציה הוא לפחות הערך של שחקן j בחלוקה Y, שהוא גדול מ-z_k. מכאן, ששחקן j אמור להישאר חופשי באיטרציה k – בסתירה להנחה ש-j נמצא בקבוצה S_k. ***

יעילות אגליטרית והגינות

האם חלוקה אגליטרית היא גם הוגנת? תלוי לאיזה הגינות מתכוונים: לגבי חלוקה פרופורציונלית התשובה היא "כן" (בתנאים סבירים), אבל לגבי חלוקה ללא-קנאה התשובה היא "לא".

משפט. נתונה בעיית חלוקה כלשהי שבה פונקציות-הערך מנורמלות, כלומר קיים קבוע V כך ש:

$v_i(C) = V$ for all i in 1,…,n.

אם קיימת חלוקה פרופורציונלית כלשהי, אז כל חלוקה אגליטרית היא פרופורציונלית.

הוכחה. בחלוקה פרופורציונלית, הערך של כל שחקן הוא לפחות V/n. לכן הערך המינימלי הוא לפחות V/n. חלוקה אגליטרית ממקסמת את הערך המינימלי, ולכן הערך המינימלי בחלוקה אגליטרית הוא לפחות V/n. כל חלוקה כזאת היא פרופורציונלית. ***

בפרט, בבעיית חלוקת עוגה עם פונקציות-ערך רציפות אדיטיביות ומונוטוניות, קיימת חלוקה פרופורציונלית (ראו ביחידה הקודמת), ולכן כל חלוקה אגליטרית היא גם פרופורציונלית.

אולם, חלוקה אגליטרית לא תמיד מקיימת את התנאי החזק יותר של חלוקה ללא-קנאה. הדוגמה – במטלה.

האם תמיד קיימת חלוקה שהיא גם יעילה-פארטו וגם ללא-קנאה? התשובה היא כן! אבל לא נספיק ללמוד על כך השנה.