# אלגוריתמים לחלוקה הוגנת של שכר דירה - המשך

## המודל הקרדינלי

במודל הקרדינלי, כל אחד מהשותפים מייחס **ערך מספרי** לכל חדר, המשקף את הסכום המירבי (לחודש) שהוא מוכן לשלם עבור החדר. ההנחות הן:

* החדרים הם סבירים - סכום הערכים שכל שותף מייחס לכל החדרים שווה לפחות כמו שכר-הדירה הכולל.

* הדיירים הם קוואזי-ליניאריים - אם דייר חושב שחדר מסויים שווה $v$, ושכר-הדירה על החדר הזה הוא $p$, אז התועלת של הדייר (= רמת השמחה שלו) היא $v$ פחות $p$.

במודל זה, חלוקה ללא קנאה היא חלוקה שבה כל שותף מקבל חדר עם תג-מחיר, כך שהתועלת של כל שותף מהחדר שלו גדולה לפחות כמו התועלת שלו מכל חדר אחר (בהינתן המחירים).

הנחת הקוואזי-ליניאריות בדרך-כלל סותרת את "הנחת הדיירים העניים" מהמודל האורדינלי (מדוע?). לכן, אלגוריתם סימונס-סו לא יעבוד.

## הגינות ויעילות

כהכנה לאלגוריתם, נוכיח משפט מעניין המקשר בין הגינות ליעילות בבעיית חלוקת החדרים.

**משפט**: בכל השמה ללא קנאה, סכום הערכים של הדיירים בחדרים שהם גרים בהם הוא מקסימלי.

**הוכחה**: תהי $X$ השמת-חדרים ללא קנאה. תהי $Y$ השמה אחרת כלשהי. לפי הגדרת קנאה:

$$V_i(X_i) - P(X_i) \geq V_i(Y_i) - P(Y_i)$$

נסכום על כל הדיירים, $i$ בין 1 ל-$n$:

$$\sum(V_i(X_i) - P(X_i)) \geq \sum(V_i(Y_i) - P(Y_i))$$

$$\sum V_i(X_i) - \sum P(X_i) \geq \sum V_i(Y_i) - \sum P(Y_i)$$

בשני הצדדים, סכום המחירים שווה למחיר הדירה. לכן אפשר לצמצם אותו ומקבלים:

$$\sum V_i(X_i) \geq \sum V_i(Y_i)$$

מכאן: בהשמה $X$, סכום הערכים גדול לפחות כמו בהשמה $Y$. ***

שימו לב - יש הבדל בין הבעיה הזאת לבעיית חלוקת העוגה: בחלוקת העוגה, היינו צריכים לעבוד קשה כדי למצוא חלוקה שהיא גם ללא-קנאה וגם יעילה. כאן, **כל** חלוקה ללא-קנאה היא יעילה.

המשפט הזה נותן לנו כיוון לאלגוריתם למציאת חלוקה ללא קנאה. נעבוד בשני שלבים:

- שלב א: נמצא השמה של אנשים לחדרים, הממקסמת את סכום הערכים.
- שלב ב: נקבע מחיר לכל חדר, כך שההשמה עם המחירים היא ללא קנאה.

# שלב א: מציאת השמה עם סכום ערכים מקסימלי

נניח שיש n אנשים ו-n חדרים. כמה השמות שונות של אנשים לחדרים יש? תשובה: n! (עצרת). אם n קטן אפשר לבדוק את כולן, אבל אם n גדול זה לא מעשי. אנחנו נראה איך אפשר לפתור את הבעיה הזאת ביעילות.

השם הכללי של הבעיה הזאת הוא בעיית שידוך עם משקל מקסימלי (maximum-weight matching). הקלט לבעיה הוא גרף דו-צדדי: הצמתים בגרף מחולקים לשתי קבוצות, וכל קשת מקשרת בין צומת בקבוצה א לצומת בקבוצה ב. לכל קשת בגרף יש משקל חיובי. שידוך בגרף הוא אוסף של קשתות, כך שכל צומת נוגע בקשת אחת לכל היותר (כל צומת משודך לצומת אחר יחיד, או לא משודך כלל). שידוך מושלם בגרף הוא שידוך שבו כל הצמתים משודכים - כל צומת נוגע בקשת אחת בדיוק. המשקל של שידוך כלשהו הוא סכום משקלי הקשתות המשתתפות בשידוך. המטרה היא למצוא שידוך מושלם, שהמשקל הכולל שלו הוא הגדול ביותר האפשרי.

אגב, לבעיה הזאת יש שם נוסף - בעיית ההשמה (assignment problem). יש לה הרבה מאד שימושים בתעשיה ובחקר-ביצועים - לא רק בחלוקה הוגנת. בבעיית ההשמה, יש n עובדים ו-n משימות. לכל זוג של עובד-משימה ישנה עלות מסויימת הקובעת, למשל, כמה זמן העובד צריך כדי לבצע את המשימה. צריך לתת משימה אחת לכל עובד, כך שסכום העלויות הוא מינימלי. לדוגמה: תחנת מוניות מסויימת מקבלת בו-זמנית 3 הזמנות לנסיעה. לתחנה יש 3 מוניות הנמצאות במקומות שונים. לכל זוג מונית-נוסע ישנה עלות הנובעת מהזמן שייקח למונית להגיע אל הנוסע. התחנה צריכה להתאים מונית לכל נוסע, כך שסכום זמני ההגעה יהיה מינימלי. בבעיית חלוקת החדרים, אנחנו מחפשים השמה שבה סכום הערכים הוא מקסימלי, אבל ההבדל הזה הוא טכני בלבד - אם יש לנו פתרון לבעיית ההשמה עם סכום מינימלי, אנחנו יכולים פשוט להריץ אותו עם ערכים שליליים, ונקבל פתרון עם סכום מקסימלי.

יש אלגוריתמים רבים לפתרון בעיית ההשמה. אחד המפורסמים שבהם הוא האלגוריתם ההונגרי (Hungarian Algorithm / Munkres Algorithm / Kuhn Algorithm). אבל, בשיעור הזה נלמד אלגוריתם אחר המסתמך על תיכנות ליניארי.

ראינו בשיעור הקודם, שישנם אלגוריתמים וכלי-תוכנה יעילים לפתרון בעיות אופטימיזציה עם משתנים רציפים, למשל ב Mathematica. הבעיה היא, שבעיית האופטימיזציה הנוכחית היא בדידה - אין כאן משתנים רציפים - כל חדר נמסר לאדם אחד לכל היותר.

הפתרון הוא להציג את הבעיה כבעיה רציפה, ואז להוכיח שהפתרון המתקבל הוא בדיד.

לכל קשת בגרף (בין דייר i לחדר j), נגדיר משתנה x[i,j] ערך המשתנה יהיה 1 אם הקשת נמצאת בשידוך, ו-0 אם הקשת לא נמצאת בשידוך.

כדי לוודא שהמשתנים אכן מייצגים שידוך מושלם, נגדיר אילוץ האומר שלכל צומת בגרף, סכום המשתנים של הקשתות הסמוכות לצומת הוא 1:

- For all i: $Sum_j$ x[i,j] = 1 , For all j: $Sum_i$ x[i,j] = 1

נניח שהמשקל של הקשת i,j הוא w[i,j]. אז המשקל הכולל של השידוך הוא:

$Sum_{i,j}$ w[i,j] * x[i,j]

לסיכום, הבעיה שיש לפתור היא (בעיה 1):

Maximize $Sum_{i,j}$ w[i,j] * x[i,j]

Such that For all i: $Sum_j$ x[i,j] = 1 ; For all j: $Sum_i$ x[i,j] = 1

For all i,j: 1 ≥ x[i,j] ≥ 0

For all i,j: x[i,j] in Z

כל האילוצים מתאימים למשתנים רציפים, פרט לאילוץ האחרון - האילוץ האחרון קובע שהמשתנים צריכים להיות מספרים שלמים (בקבוצה Z).

לפתור בעיות עם אילוץ של מספרים שלמים זה עלול להיות מאד קשה. למעשה, הבעיה של תיכנות ליניארי בשלמים (Integer Linear Programming) ידועה כבעיה NP-קשה - זה אומר שכנראה לא נמצא פתרון כללי לבעיה זו.

למרבה השמחה, במקרה המיוחד שלנו, אפשר לפתור את הבעיה בלי האילוץ האחרון - והפתרון שנקבל יהיה שלם!

לפני שנוכיח את זה, נסו את זה ב-Mathematica (יש מחברת מוכנה בתת-תיקיה code) - הכניסו את הבעיה עם האילוצים הרציפים בלבד, ותראו שהפתרון המתקבל הוא שלם. עכשיו נוכיח שזה תמיד נכון.

**משפט**: אם לבעיה 1 קיים פתרון מקסימלי, אז קיים גם פתרון מקסימלי שבו כל ערכי המשתנים הם שלמים.

**הוכחה**: יהי x פתרון כלשהו עם משקל מקסימלי. כזכור, x הוא מטריצה עם $n^2$ מספרים בין 0 ל-1. אם כל המספרים ב-x הם שלמים - סיימנו. אחרת, נבחר משתנה אחד שאינו מספר שלם. נניח שהמשתנה הזה הוא $x[i_1, j_2]$. כיוון שסכום המשתנים הסמוכים לצומת $j_2$ חייב להיות 1 (שהוא מספר שלם), חייב להיות משתנה נוסף שאינו מספר שלם - נניח $x[i_3,j_2]$. לפי אותם שיקולים עבור $i_3$, חייב להיות משתנה נוסף שאינו שלם - נניח $x[i_3,j_4]$. וכן הלאה. כיוון שמספר המשתנים סופי, חייב להיות מעגל של משתנים לא-שלמים. נניח שהמשתנה האחרון במעגל הוא $x[i_1,j_{2m}]$, כלומר מספר הקשתות במעגל הוא $2m$ (מספר הקשתות במעגל חייב להיות זוגי כי זה גרף דו-צדדי).

עכשיו נניח שאנחנו מורידים מספר קטן מאד e מכל המשתנים האיזוגיים במעגל, ומוסיפים את אותו e לכל המשתנים הזוגיים במעגל. מה קורה?

- האילוץ של כל צומת עדיין מתקיים: לכל צומת, הורדנו e מקשת סמוכה אחת, והוספנו e מקשת סמוכה אחרת, כך שסכום המשתנים הסמוכים נשאר זהה - 1.
- אם e מספיק קטן, אז כל המשתנים עדיין בין 0 ל-1. אפשר לבחור את e כך שלפחות משתנה אחד יהפוך למספר שלם: נבחר את ההפרש הקטן ביותר בין משתנה איזוגי לבין אפס, או את ההפרש הקטן ביותר בין משתנה זוגי לבין 1. כך משתנה אחד יהפוך לשלם, וכל השאר יישארו בין אפס לאחד.
- המשקל הכולל של הפתרון לא משתנה: הוא לא יכול לגדול - כי הנחנו ש-x הוא פתרון עם משקל מקסימלי. הוא לא יכול לקטון - כי אילו הוא היה קטן, היינו יכולים להחליף את e במינוס e ומקבלים פתרון חדש עם משקל גדול יותר - וזו שוב סתירה לכך ש x הוא פתרון עם משקל מקסימלי.

בסך-הכל, קיבלנו פתרון מקסימלי חדש, עם משתנה לא-שלם אחד פחות.

נמשיך כך להוריד מעגלים, ונקבל פתרון שבו כל המשתנים שלמים! ***

כאמור, ישנם אלגוריתמים שונים לפתרון הבעיה, כגון האלגוריתם ההונגרי. אבל הם הרבה יותר מסובכים למימוש. לדוגמה, בויקיפדיה בדף Hungarian Algorithm ישנם כמה מימושים של אלגוריתם זה. מימוש טיפוסי דורש 300 שורות קוד. לעומת זאת, פתרון בעיית אופטימיזציה ב Mathematica דורש בערך 10 שורות קוד.

## שלב ב: קביעת המחירים

מצאנו השמה הממקסמת את סכום הערכים. אז אנחנו כבר יודעים איזה דייר לשים באיזה חדר.

כדי להשלים את הפתרון, אנחנו צריכים לקבוע מחיר לכל חדר, כך שההשמה תהיה ללא קנאה.

בנוסף, אנחנו צריכים לוודא שסכום המחירים מכסה את שכר-הדירה.

איך נעשה את זה?

דרך אחת אפשרית היא לפתור עוד בעיית אופטימיזציה. הפעם הבעיה תהיה רציפה - כי המחיר הוא משתנה רציף. יהיו לנו רק n משתנים - לכל חדר i יהיה משתנה p[i] הקובע את מחיר החדר. נגדיר את האילוצים כך שהחלוקה תהיה ללא קנאה.

יש כמה בעיות שאפשר לפתור. למשל, אפשר למצוא את וקטור-המחירים עם הסכום הקטן ביותר. בתוכנית למטה, הביטוי d[i] מציין את הדייר המשוייך לחדר מספר i (לפי הפתרון של הבעיה הקודמת):

Minimize Sum$_i$ p[i]

Such that For all i, j: w[d[i], i] - p[i] ≥ w[d[i], j] - p[j]

עכשיו נשאר רק לוודא שסכום המחירים שווה למחיר הכולל של הדירה. זה קל - אם הסכום קטן יותר או גדול יותר, פשוט מחלקים את ההפרש במספר החדרים, ומוסיפים או מפחיתים לכל חדר את אותה כמות. החלוקה עדיין נשארת ללא קנאה.

## מימושים והדגמות

אלגוריתם לחלוקת שכר-דירה מומש בכמה מקומות - אתם מוזמנים לנסות:

- אתר לקבוצות רכישה http://tora.us.fm/fairness/home
- אתר לחלוקת ירושות http://tora.us.fm/fairness/home/ab.html
- אלג. גל-מש-פרוקצ'יה-זיק: http://www.spliddit.org/apps/rent

## בעיית הטרמפיסט

אחת הבעיות במודל הקרדינלי היא, שהמחירים עלולים להיות שליליים. לדוגמה, נניח שהערכים הם:

| | סלון | מרתף |
|---|---|---|
| דייר א | 150 | 0 |
| דייר ב | 140 | 10 |

ומחיר הדירה הוא 100. ההשמה חייבת למקסם את סכום הערכים - לכן חייבים לשים את דייר א בסלון ואת דייר ב במרתף.

כדי שדייר ב לא יקנא, ההפרש במחירים חייב להיות לפחות 130. לכן המחיר של הסלון חייב להיות 115 והמחיר של המרתף מינוס 15. משלמים לדייר ב כדי שיגור במרתף!

הפתרון הזה לא מתקבל על הדעת - הדיירים בוודאי לא יסכימו לשלם למישהו כדי שיגור במרתף - הם יעדיפו לוותר על המרתף...

פתרון פשוט למקרה זה הוא להפוך את כל המחירים השליליים ל-0, ולחלק את היתרה בין השחקנים האחרים. הבעיה היא, שהפתרון יהיה עם קנאה. אבל, הדיירים היחידים שיקנאו יהיו הדיירים שמקבלים חדר בחינם.

האם אפשר למצוא פתרון שהוא גם ללא-קנאה וגם בלי טרמפיסטים? התשובה היא כן - הפתרון האורדינלי. אבל לפתרון האורדינלי יש בעיה אחרת - הוא מניח את הנחת "הדיירים העניים".

שוב הגענו ל"טרילמה" (דילמה משולשת): יש לנו שלושה פתרונות, כל פתרון מקיים שתי תכונות, ואין אף פתרון שמקיים את כל שלוש התכונות (דיירים לא-עניים, אין קנאה, אין טרמפיסטים).

# מקורות


- Bernd Gärtner and Jiří Matoušek (2006). "Understanding and Using Linear Programming". Pages 31-37.
- Gal, Ya'akov (Kobi); Mash, Moshe; Procaccia, Ariel D.; Zick, Yair (2016). "Which Is the Fairest (Rent Division) of Them All?". ACM: 67–84. doi:10.1145/2940716.2940724. ISBN 9781450339360.


סיכם: אראל סגל-הלוי.