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.

הוכחת קיום

[שימו לב: ההוכחה שונה מההוכחה שניתנה בהרצאה. היא אמורה להיות ברורה יותר]

לפני שנתחיל לפתח אלגוריתם חדש, צריך קודם להוכיח שתמיד קיימת חלוקה ללא קנאה, גם במודל הקוואזי-ליניארי. במקום הנחת הדיירים העניים, נשתמש בתכונה אחרת שתמיד קיימת במודל הקוואזי-ליניארי: קיים מספר כלשהו T, כך שאף דייר לא רוצה לקחת חדר שמחירו T או יותר.

מדוע זה נכון? כי הבעיה סופית, ולכן קיים חסם עליון על הערכים שהדיירים מייחסים לחדרים. נניח שהחסם הזה הוא T; אז כל חדר שמחירו T נותן תועלת שלילית – ואף דייר לא רוצה לקבל תועלת שלילית.

עכשיו, לצורך הוכחת הקיום נשתמש באותה שיטה של סימונס וסו – שיטת המשולשונים – רק שהפעם נייחס לכל נקודה במשולש ערך אחר: לנקודה (x1,...,xn) נייחס וקטור מחירים (p1,...,pn), כאשר לכל i:

pi = T – (Tn-R)*xi

[כזכור, n הוא מספר הדיירים ו-R הוא שכר-הדירה הכולל].

עבור כל נקודה בסימפלקס, סכום המחירים הוא בדיוק R – כי הסכום של כל ה -xi הוא בדיוק 1. לכל i שעבורו xi=0, המחיר pi=T. כיוון שאף דייר לא רוצה חדר שמחירו T, מתקיים התנאי של ספרנר: כל דייר בוחר, בכל קודקוד, חדר i שעבורו xi > 0. לכן קיים משולשון המייצג חלוקה ללא-קנאה-בקירוב. אם מחלקים למשולשונים יותר ויותר קטנים, מקבלים סדרה אינסופית המתכנסת לנקודה המייצגת חלוקה ללא קנאה.

שימו לב: הפעם יכולים להיות גם מחירים שליליים. מבחינת האלגוריתם זה לא מהווה בעיה: אפשר להציג לכל דייר וקטור הכולל גם מחירים שליליים, ולשאול אותו איזה חדר הוא מעדיף. בהחלט ייתכן שדיירים קוואזי-ליניאריים יעדיפו חדר עם מחיר שלילי על-פני חדר עם מחיר חיובי. לדוגמה: מציעים לכם לגור

במרתף טחוב תמורת מינוס 5 ש"ח לחודש (כלומר לקבל 5 ש"ח לחודש על כך שאתם מסכימים לגור שם), או לגור בחדר-שינה מרווח תמורת פלוס 100 ש"ח לחודש. אנשים רבים יעדיפו את האפשרות השניה.

לסיכום, אנחנו יודעים שקיימת חלוקה ללא קנאה, אבל עדיין אין לנו אלגוריתם **יעיל** המוצא חלוקה כזאת; אלגוריתם הסימפלקסונים מוצא חלוקה ללא קנאה רק בקירוב. בהמשך נציג אלגוריתם המוצא חלוקה ללא קנאה במדוייק.

הגינות ויעילות

האלגוריתם שנציג מסתמך על Sung and Vlach, 2004.

כהכנה לאלגוריתם, נוכיח שני משפטים המקשרים בין הגינות ליעילות בבעיית חלוקת החדרים.

משפט 1: 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.

משפט 2: כל וקטור-מחיר ללא קנאה, יישאר ללא-קנאה לכל השמה ממקסמת-סכום-ערכים.

הוכחה: תהי X,P השמת-חדרים ללא קנאה. לפי המשפט הקודם, X ממקסמת סכום ערכים. תהי Y השמה אחרת הממקסמת סכום ערכים:

$$\sum V_i(X_i) = \text{MaxSum} = \sum V_i(Y_i)$$

השיוויון נשאר גם כאשר מפחיתים מכל ערך את מחיר החדר המתאים:

$$\sum[V_i(X_i)-P(X_i)] = \text{MaxSum} - \text{TotalRent} = \sum[V_i(Y_i)-P(Y_i)]$$

נתון ש-X ללא קנאה. לכן לפי הגדרת קנאה, לכל i:

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

נסכום על כל הדיירים, i בין 1 ל-n:

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

אבל כיוון שמתקיים שיוויון בין הסכומים, חייב להתקיים שיוויון בכל איבר, לכל i :

$$V_i(X_i) - P(X_i) = V_i(Y_i) - P(Y_i)$$

לכן גם Y,P ללא קנאה.

המשפטים האלה נותנים 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.

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, למשל האלגוריתם ההונגרי (Hungarian Algorithm / Munkres Algorithm / Kuhn Algorithm). יש גם אלגוריתמים יעילים יותר. אלגוריתמים לבעיית ההשמה נלמדים בקורס על אלגוריתמים בתורת הגרפים. לצורך הקורס הנוכחי, אנחנו מתייחסים לאלגוריתמים האלה כ"קופסה שחורה". כדי למצוא השמה הממקסמת את סכום הערכים, פשוט נבנה גרף דו-צדדי עם שבו המשקלים הם הערכים, ונפעיל את אחד האלגוריתמים. לדוגמה, עבור בעיה עם שלושה אנשים ושלושה חדרים, אפשר להפעיל את הקוד הבא בשפת פייתון (המשתמש בספריה networkx); כאשר במקום המספרים 111, 222, 333 נכניס את הערכים האמיתיים של האנשים לחדרים (ראו דוגמה בתיקיית הקוד).

```python
import networkx as nx

G=nx.Graph()   # Construct an empty graph:

G.add_edge('person 1','room 1' ,weight=111)
G.add_edge('person 1','room 2' ,weight=222)
G.add_edge('person 1','room 3' ,weight=333)

G.add_edge('person 2','room 1' ,weight=444)
...
...
...
G.add_edge('person 3','room 3' ,weight=999)

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

הפונקציה nx.max_weight_matching היא פונקציה המוצאת שידוך עם משקל גדול ביותר.

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 בעיית תיכנות ליניארי. כזכור, בהרצאה הקודמת למדנו לפתור בעיית תיכנות קמור בהקשר של חלוקת משאבים: לכל משאב ולכל אדם, היה משתנה המגדיר איזה חלק מהמשאב נמסר לאותו אדם, היו לנו אילוצים הקובעים שכל משתנה חייב להיות בין 0 ל-1, וסכום המשתנים המתאימים למשאב מסויים שווה 1. תחת האילוצים האלה, ניסינו למצוא את סכום הלוגריתמים של הערכים.

הפעם יש לנו בעיה של תיכנות ליניארי – שהיא מקרה פרטי של תיכנות קמור. יש לנו n משתנים - לכל חדר i יש משתנה p[i] הקובע את מחיר החדר. האילוצים הם שהחלוקה צריכה ללא קנאה, ושסכום המחירים צריך להיות שווה למחיר הדירה. במקרה זה לא צריך לבצע מקסימיזציה/מינימיזציה כי אנחנו מעוניינים רק באילוצים. בתוכנית למטה, הביטוי d[i] מציין את הדייר המשוייך לחדר מספר i (לפי הפתרון של הבעיה הקודמת - בעיית השידוך עם משקל מקסימלי):

Minimize 1

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

Sum_i p[i] = TotalRent

ניתן לפתור את הבעיה הזאת בעזרת cvxpy כמו שלמדנו בהרצאה הקודמת; ראו בתיקיית הקוד.

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.