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 נלמד עכשיו אלגוריתם הפותר את בעיית ההשמה (assignment problem) בזמן סביר.

לבעיית ההשמה יש הרבה מאד שימושים בתעשיה ובחקר-ביצועים - לא רק בחלוקה הוגנת. הניסוח המקורי של הבעיה הוא: 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 קטן ביותר.

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. אנחנו נלמד אלגוריתם אחד שנקרא האלגוריתם ההונגרי (Hungarian Algorithm / Munkres Algorithm / Kuhn Algorithm).

הקלט לאלגוריתם הוא מטריצה ריבועית בגודל n על n. כל שורה מייצגת עובד, כל עמודה מייצגת משימה, וכל משבצת מייצגת את העלות של הזוג עובד-משימה.

האלגוריתם מסתמך על שתי הלמות הבאות:

למה1: אם מוסיפים את אותו מספר לכל המשבצות באותו טור/שורה, ההשמה המינימלית לא משתנה.

הוכחה: כל השמה כוללת משבצת אחת מכל טור ומשבצת אחת מכל שורה. לכן, אם מוסיפים x לכל המשבצות באותו טור או באותה שורה, הערך של כל ההשמות גדל ב-x. לכן ההשמה שהיתה מינימלית קודם נשארת מינימלית גם עכשיו. ***

למה2: אם כל המספרים במטריצה הם לפחות 0, ומצאנו השמה שבה כל המשבצות הן 0, אז זו השמה עם עלות מינימלית.

הוכחה: אם כל המספרים הם לפחות 0, לא יכולה להיות השמה עם עלות נמוכה יותר מ-0. ***

משפט נוסף שנצטרך הוא משפט כללי בתורת הגרפים שנקרא **משפט קניג**. המשפט מקשר בין שני מושגים בתורת הגרפים: שידוך (matching) וכיסוי (covering). למדתם על זה בקורס קודם, אבל כדי לרענן את הזיכרון:

- שידוך הוא אוסף של קשתות, כך שכל צומת נוגע בקשת אחת לכל היותר (כך שכל צומת יכול להיות משודך לצומת אחד אחר, או להישאר לא משודך).
- כיסוי הוא אוסף של צמתים, כך שכל קשת נוגעת בצומת אחד לפחות.

אם קיים שידוך בגודל k, אז כל כיסוי חייב להיות בגודל k לפחות - אחרת לא נצליח לכסות את כל הקשתות בשידוך.

מכאן: השידוך הגדול ביותר בגרף, שווה או קטן מהכיסוי הקטן ביותר באותו גרף.

משפט קניג: בגרף דו-צדדי, מספר הקשתות בשידוך הגדול ביותר שווה למספר הצמתים בשידוך הקטן ביותר.

עכשיו נחזור למטריצת העלויות. כדי למצוא השמה מינימלית, אנחנו נחסר מספרים מטורים ושורות, ננסה ליצור כמה שיותר אפסים, ואז ננסה ליצור השמה עם עלות 0.

- **צעד 1**. עבור כל שורה/טור, מצא את המספר המינימלי בשורה/טור, והפחת אותו מכל המשבצות באותה שורה/טור. עכשיו, בכל שורה/טור יש לפחות אפס אחד.
- **צעד 2**. צור את גרף האפסים - גרף דו-צדדי שבו, בצד אחד נמצאות השורות, בצד שני הטורים, ויש קשת בין שורה לטור אם"ם יש 0 במשבצת המתאימה.
- **צעד 3**. מצא שידוך גדול ביותר בגרף האפסים. אם גודל השידוך שנמצא הוא n - מצאנו השמת-אפסים, וזו ההשמה המינימלית.
- **צעד 4**. אם השידוך הגדול ביותר קטן מ-n, אז מצא כיסוי קטן ביותר בגרף האפסים. לפי משפט קניג, גודל הכיסוי שווה לגודל השידוך שמצאנו בצעד 3, כלומר הוא קטן מ-n. במטריצה, "כיסוי" משמעו פשוט העברת קוים על שורות ועל טורים, כך שכל האפסים מכוסים; לפי משפט קניג, מספר הקוים קטן מ-n.
- **צעד 5**. עכשיו כל האפסים מכוסים, אבל מספר הקוים קטן מ-n ולכן יש משבצות לא מכוסות. מצא את הערך הקטן ביותר שאינו מכוסה, נניח x (שהוא מספר חיובי). הפחת x מכל שורה לא-מכוסה, והוסף x לכל טור מכוסה. שימו לב - כל המספרים במטריצה נשארים לפחות 0. למה? כי לפני השינוי, כל המספרים הלא-מכוסים היו לפחות x, כך שגם אחרי שהורדנו x הם נשארו לפחות 0. לגבי המספרים המכוסים - אם הם מכוסים ע"י שורה, לא הפחתנו מהם שום דבר, ואם הם מכוסים ע"י טור, הוספנו להם x.
- חזור לצעד 3.

מידע נוסף על האלגוריתם ההונגרי ניתן למצוא כאן:

- http://www.hungarianalgorithm.com - הדגמה
- http://www.math.harvard.edu/archive/20_spring_05/handouts/assignment_overheads.pdf - שקפים

סיכם: אראל סגל-הלוי.