מכרזים אלגוריתמיים

היום נראה איך אלגוריתמים שלמדתם בקורסים קודמים משתלבים עם המכרזים שלמדנו בקורס זה.

מכרז המסלול הזול ביותר

נתונה רשת-תקשורת. כל קשת מייצגת חיבור ברשת. אנחנו רוצים להעביר חבילת-תקשורת מסויימת מנקודה מסויימת ("מקור") לנקודה אחרת ("יעד").

לכל קשת ישנה עלות מסויימת להעברת חבילה דרכה (למשל, לפי הזמן שלוקח לחבילה לעבור בקשת, העומס על הקשת וכו'). אנחנו רוצים להעביר את החבילה במסלול **הזול ביותר**.

אם עלות כל קשת ידועה לכולם - אפשר לפתור את הבעיה ע"י אלגוריתם שאתם כבר מכירים - אלגוריתם דייקסטרה מוצא מסלול קל ביותר בגרף ממושקל.

אבל מה אם כל קשת היא מחשב ששייך לאדם מסויים, ורק הבעלים יודע מה העלות שלה? אנחנו יכולים לשאול כל אחד מה העלות של העברת חבילה דרכו, אבל למה שיגיד את האמת? אולי כדאי לו להגיד שהעלות הרבה יותר גבוהה, כדי שנחליט לשלוח את החבילה דרך מקום אחר? הדבר רלבנטי במיוחד בימינו, ברשתות כגון ביטקוין-לייטנינג (bitcoin-lightning), שבהם צריך להחליט מה המסלול הזול ביותר להעברת תשלום בין שני צרכנים.

בכל מצב שבו הקלט לאלגוריתם מסויים הוא מידע פרטי של אנשים עם אינטרסים אישיים (שאינם בהכרח תואמים לאינטרסים שלנו), אנחנו צריכים להשתמש ב**מכרז** (או **מנגנון**). ורצוי שהמכרז יהיה **אמיתי**, כדי שלאנשים אכן יהיה כדאי לחשוף את המידע האמיתי שלהם.

בשיעור הקודם למדנו מכרז אמיתי כללי - מכרז ויקרי-קלארק-גרובס (VCG). אנחנו יכולים להשתמש בו במקרה שלנו (ראו דוגמה במצגת).

- אוסף התוצאות האפשריות הוא אוסף כל המסלולים מהמקור ליעד.
- השלב הראשון במכרז וק"ג הוא חישוב התוצאה עם סכום-הערכים הגדול ביותר. במקרה זה אנחנו מחפשים את סכום העלויות הקטן ביותר, אבל ההבדל הוא טכני בלבד - אפשר להציג כל עלות כערך עם סימן מינוס. לכן, אנחנו יכולים לבצע את השלב הראשון במכרז בעזרת אלגוריתם דייקסטרה.
- השלב השני הוא חישוב התשלומים. כדי לחשב את התשלום של שחקן מסויים (במקרה זה, של קשת מסויימת), אנחנו צריכים לחשב מה היתה התוצאה אילו השחקן הזה לא היה משתתף. אנחנו צריכים למחוק (באופן זמני) את הקשת מהגרף, ולמצוא את המסלול הזול ביותר בגרף שהתקבל, כלומר להריץ שוב את אלגוריתם דייקסטרה. אחרי שמצאנו את המסלול הזה, אנחנו צריכים לחשב את סכום הערכים של כל שאר הקשתות, לחסר את סכום הערכים של כל שאר הקשתות בלי המחיקה, וההפרש הוא התשלום שתצטרך הקשת לשלם.

שימו לב - כדי להריץ את מנגנון וק"ג צריך להריץ את אלגוריתם דייקסטרה E+1 פעמים, כאשר E הוא מספר הקשתות בגרף: פעם אחת כדי לחשב את התוצאה הסופית, ועוד E פעמים כדי לחשב את התשלום של כל קשת וקשת.

מכרז התרמיל

מכניסים אתכם לחדר מלא חפצים, נותנים לכם תרמיל שיכול להכיל עד 100 ק"ג, ואומרים לכם "כל מה שתצליחו להכניס לתרמיל – שלכם". לכל חפץ יש משקל אחר וערך אחר. איך תבחרו חפצים שהמשקל שלהם לכל היותר 100, וסכום-ערכיהם גדול ביותר?

אם הערך של כל חפץ ידוע לכולם - יש לנו בעיה אלגוריתמית. אמנם, זו בעיה קשה יותר מבעיית המסלול הזול ביותר, אבל היא פתירה. למשל, אם מספר החפצים הוא קטן, אפשר לבדוק את כל תת-הקבוצות של חפצים, ולבחור את תת-הקבוצה עם הערך הגבוה ביותר מבין תת-הקבוצות שסכום המשקלים שלהן הוא לכל היותר 100.

אבל מה אם הערך של כל חפץ ידוע רק לחפץ? הנה דוגמה מעשית למצב זה: אתם מנהלים ערוץ רדיו ומקציבים 100 שניות לפרסומות. יש כמה מפרסמים המתחרים על המקום הזה. לכל מפרסם יש פרסומת באורך אחר (בשניות), וכל מפרסם מייחס ערך אחר לפרסומת שלו. אתם רוצים לשדר את הפרסומות עם סכום הערכים הגבוה ביותר, בתנאי שהן נכנסות ב-100 השניות המוקצבות. אבל אתם לא יודעים את הערכים. מה תעשו?

כאמור, בכל מצב שבו הקלט לאלגוריתם הוא מידע פרטי של אנשים עם אינטרסים אישיים, אנחנו צריכים להשתמש במכרז. אפשר להשתמש במכרז וק"ג כמו שראינו בסעיף הקודם. בשלב הראשון של המכרז, אנחנו צריכים למצוא את התוצאה שממקסמת את סכום הערכים. אם מספר החפצים (= המפרסמים) הוא קטן, זה עובד מצויין. אבל מה אם מספר החפצים הוא גדול? בעיית התרמיל ידועה כבעיה NP-שלמה - וזה אומר שכנראה לא קיים אלגוריתם הפותר בעיות גדולות בזמן סביר. איך נוכל להריץ את מכרז וק"ג, אם אנחנו לא יכולים לחשב את הצעד הראשון שלו?!

בעולם האלגוריתמים, כשנתקלים בבעיה שהיא NP-קשה, השלב הבא הוא לחפש עבורה אלגוריתמי-קירוב. אם זה קשה למצוא את הצירוף עם סכום הערכים המקסימלי - ננסה למצוא צירוף עם סכום-ערכים שהוא "די קרוב" למקסימלי. איך מוצאים אלגוריתמי קירוב? אפשרות אחת, שעובדת בהרבה מקרים, היא לנסות אלגוריתם חמדני (greedy algorithm). נראה כמה אלגוריתמים כאלה עבור בעיית התרמיל.

אלגוריתם חמדני א:

- סדר את החפצים בסדר יורד של הערך.
- בחר חפצים לפי הסדר עד שהתרמיל מתמלא.

האלגוריתם הזה עובד יפה כשהחפצים בערך באותו משקל, אבל כשיש הבדלים גדולים במשקלים - הוא עלול להיות מאד גרוע.
למשל, נניח שגודל התרמיל הוא 100, יש 51 חפצים, והערכים והמשקלים שלהם לפי הסדר הם:
$100/100k, $20/2k, $20/2k, $20/2k ...
רק החפץ הראשון נכנס לתרמיל - הערך הוא $100. אילו היה מדלג על החפץ הראשון, היה יכול להכניס 50 חפצים שהערך של כל אחד הוא $20, כך שהערך היה גדול פי 10.
מעבר לכך, גם אם נשתמש באלגוריתם הזה כשלב הראשון במכרז דמוי-וק"ג, התוצאה תהיה מאד גרועה: השחקן הראשון יזכה, אבל הוא יצטרך לשלם $1000 (סכום הערכים של השחקנים כשהוא לא משתתף), שזה פי 10 יותר מהערך שלו. זה כמובן לא מכרז אמיתי - השחקן יעדיף לא להשתתף כלל.

אלגוריתם חמדני ב:

- סדר את החפצים בסדר יורד של היחס ערך/משקל.
- בחר חפצים לפי הסדר עד שהתרמיל מתמלא.

האלגוריתם הזה עובד יפה על הדוגמה הקודמת, אבל עובד גרוע על הדוגמה הבאה:
$20/2k, $100/100k.
רק החפץ הראשון נכנס לתרמיל - הערך הוא $20. אילו היה מדלג על החפץ הראשון, הערך היה גדול פי 5.
גם כמכרז, התוצאות לא טובות: השחקן הראשון זוכה, והוא משלם $100 (סכום הערכים של האחרים כשהוא לא משתתף) – יותר מהערך שלו.

ראינו שני אלגוריתמים, כל אחד מהם נכשל על קלט מסויים. מתברר ששניהם ביחד נותנים תוצאה לא רעה בכלל.

אלגוריתם א+ב: הפעל את שני האלגוריתמים החמדניים, ובחר את התוצאה עם הסכום הגבוה.
משפט: אלגוריתם א+ב נותן קירוב 1/2 לבעיית התרמיל. כלומר, הוא מוצא תת-קבוצה שסכום ערכיה הוא לפחות 1/2 מהסכום הגבוה ביותר האפשרי.
הוכחה:

- נניח שאלגוריתם ב נתקע אחרי k חפצים. אילו היה יכול להוסיף גם את החפץ ה-k+1 – הסכום היה לפחות כמו המקסימום - אולי אפילו יותר (כי האלגוריתם ממקסם את הערך ליחידת משקל).
- אלגוריתם א בוחר לפחות חפץ אחד עם הערך הגבוה ביותר. כלומר, הסכום של אלגוריתם א הוא לפחות כמו ערך החפץ ה-k+1.
- מכאן, הסכום של אלגוריתמים א+ב הוא לפחות כמו המקסימום.
- מכאן, לפי עקרון שובך היונים, הסכום של אחד משני האלגוריתמים הוא לפחות כמו המקסימום / 2! ***

עשינו צעד גדול קדימה - מצאנו אלגוריתם שזמן-הריצה שלו הוא ליניארי, והוא מוצא קירוב טוב לבעיית התרמיל.
האם אפשר להשתמש בו גם כשלב הראשון במכרז וק"ג?

למרבה הצער, התשובה היא לא. הנה דוגמה:

$54/52k, $52/51k, $49/49k.

במקרה זה, גם אלגוריתם א וגם אלגוריתם ב בוחרים את החפץ הראשון, ולכן גם אלגוריתם א+ב בוחר אותו. הערך הוא $54. כמה הוא צריך לשלם? - אילו הוא לא היה, אז שני האלגוריתמים היו בוחרים את שני החפצים האחרים, וסכום-הערכים היה $101. זה אמנם קירוב-1/2 - $54 זה קצת יותר מ-1/2 מהערך האופטימלי. אבל זה לא מכרז אמיתי, כי הראשון יצטרך לשלם $101, שזה יותר מהערך שלו!

אנחנו רואים כאן חיסרון גדול של מנגנון וק"ג - הוא חייב למצוא את סכום-הערכים המקסימלי, וחייב למצוא אותו במדוייק - לא בקירוב. אם מציאת המקסימום היא בעיה חישובית קשה - אנחנו לא יכולים להשתמש ב-VCG. צריך מנגנון אחר.

מכרז מיירסון

מכרז מיירסון (על-שם ממציאו Myerson, חתן פרס נובל לכלכלה) הוא מכרז אמיתי, המיועד למצבים שבהם לא יכולים או לא רוצים למקסם את סכום-הערכים.

מכרז מיירסון מקבל כקלט כלל-בחירה - כלל הקובע, לגבי כל שחקן, אם הוא "נבחר" או לא.
כלל-הבחירה הזה יכול להיות "בחר את השחקן עם הערך הגבוה ביותר", או "בחר את קבוצת-השחקנים עם סכום-הערכים הגבוה ביותר", ואז מכרז מיירסון זהה למכרז וק"ג. אבל אפשר להשתמש גם בכלל-הבחירה אחר, למשל "הרץ אלגוריתם-קירוב חמדני לבעיית התרמיל ובחר את השחקנים שנבחרו ע"י האלגוריתם".

לכל שחקן i ישנו ערך כספי v_i ל"היבחרות". אם שחקן i נבחר אז הערך שלו הוא v_i, ואם השחקן לא נבחר אז הערך שלו הוא 0. אם נשתמש רק בכלל-הבחירה - המכרז לא יהיה אמיתי, כי כל שחקן שהערך שלו חיובי, ירצה לשנות את הדיווח שלו כדי לוודא שהוא ייבחר. כדי שהמכרז יהיה אמיתי, אנחנו צריכים להוסיף לו כלל-תשלום - כלל הקובע, לכל שחקן, כמה כסף הוא צריך לשלם. אם שחקן i נבחר ומשלם p_i, אז התועלת שלו היא ההפרש - v_i-p_i. אנחנו מחפשים כלל-תשלום שיגרום לכך שהמכרז יהיה אמיתי.

נסביר שוב במילים אחרות. **מכרז** הוא זוג של פונקציות - פונקציית-בחירה ופונקציית-תשלום. כל פונקציה מקבלת כקלט את הערכים שמדווחים השחקנים. פונקציית-הבחירה מחזירה כפלט, עבור כל שחקן, "1" אם הוא נבחר ו"0" אם הוא לא נבחר. פונקציית-התשלום מחזירה כפלט, עבור כל שחקן, את כמות הכסף שהוא צריך לשלם.

במכרז-מיירסון, כלל-הבחירה נתון וקבוע מראש, והמטרה שלנו היא למצוא כלל-תשלום תואם, כך שהמכרז המתקבל יהיה אמיתי. השאלה המרכזית היא: האם לכל כלל-בחירה קיים כלל-תשלום תואם? התשובה היא לא, אבל ישנם הרבה כללי-בחירה שעבורם קיים כלל-תשלום.

הגדרה: כלל-בחירה נקרא מונוטוני אם, עבור כל שחקן i, הכלל הוא פונקציה מונוטונית-עולה של v_i.
בקורס זה אנחנו עוסקים רק בכללי-בחירה בינאריים - כל שחקן נבחר או לא נבחר (אפשר גם להכליל את הרעיון לכללי-בחירה רציפים, שבהם כל שחקן יכול להיבחר בהסתברות מסויימת בין 0 ל-1, אבל במסגרת זו לא נעשה זאת). כלל-בחירה בינארי הוא מונוטוני אם עבור כל שחקן i, אם הוא נבחר כשהערך שלו x, אז הוא נבחר גם כשהערך שלו הוא כל מספר הגדול מ-x.

דוגמאות לכללים מונוטוניים:

- בחר את 3 הערכים הגדולים ביותר.
- בחר את הערך הגדול ביותר, בתנאי שהוא מעל 10.
- בחר בעזרת אלגוריתם חמדני א / ב / א+ב.

דוגמאות לכללים לא מונוטוניים:

- בחר את הערך השני מלמעלה.
- בחר את הערך הגדול ביותר, אם הוא מתחת ל-7.

משפט מיירסון: מונוטוניות היא תנאי הכרחי ומספיק לאמיתיות. כלומר:
(א) לכל כלל-בחירה לא-מונוטוני - אין כלל-תשלום אמיתי.
(ב) לכל כלל-בחירה מונוטוני - קיים כלל-תשלום אמיתי, והוא יחיד.

בהמשך נוכיח את משפט מיירסון, ותוך כדי זה גם נגלה מהו כלל-התשלום היחיד שהופך את המכרז לאמיתי.

הוכחת משפט מיירסון: בהוכחה נשתמש בסימונים הבאים:

- כלל-הבחירה – c - פונקציה המקבלת כקלט את הערכים של כל המשתתפים, ומחזירה וקטור בינארי (1 = "ברכותיי, נבחרת!", 0 = "לא נבחרת!"). c הוא נתון וקבוע.
- כלל התשלום – p - פונקציה המקבלת כקלט את הערכים של כל המשתתפים, ומחזירה וקטור מספרי של תשלומים. את p אנחנו מחפשים.

התועלת של משתתף עם ערך v שאומר x היא:

$$v*c(x) - p(x)$$

במכרז אמיתי, חייב להתקיים:

$$v*c(v) - p(v) \geq v*c(x) - p(x)$$

התועלת של משתתף עם ערך x שאומר v היא:

$$x*c(v) - p(v)$$

במכרז אמיתי חייב להתקיים:

$$x*c(x) - p(x) \geq x*c(v) - p(v)$$

מחברים את המשוואות ומקבלים:

$$\mathbf{v*[c(v)-c(x)] \quad \geq \quad p(v)-p(x) \quad \geq \quad x*[c(v) - c(x)]}$$

נבדוק את אי-השיוויון הזה בשני מצבים.

מצב א: $c(v)=c(x)$. במקרה זה, המשוואה המודגשת נראית כך:

$$\mathbf{0 \quad \geq \quad p(v)-p(x) \quad \geq \quad 0}$$

מכאן נובע ש: $p(v)=p(x)$. המשמעות היא, שהתשלום של שחקן שנבחר - אינו תלוי בערך שלו, וגם התשלום של שחקן שאינו נבחר - אינו תלוי בערך שלו. זה די מובן - אילו הייתי יכול להיבחר עם שני

תשלומים שונים, הייתי תמיד אומר ערך שיבטיח לי להיבחר עם התשלום הנמוך יותר, ואז המכרז לא היה אמיתי.

מצב ב: $c(v)>c(x)$, כלומר $c(v)=1$ וגם $c(x)=0$ (המצב $c(v)<c(x)$ הוא סימטרי).
במקרה זה, המשוואה המודגשת נראית כך:

$$\mathbf{v \geq p(v)-p(x) \geq x}$$

מכאן נובע ש $v>x$, כלומר - הפונקציה c חייבת להיות מונוטונית.

מכאן אפשר גם לחשב את כלל התשלומים. נניח ש-v ו-x הם מאד מאד קרובים - כמעט זהים; הם ממש ליד ה"סף" שבו הפונקציה c מתחלפת מ-0 ל-1. x קצת מתחת לסף, ו-v קצת מעל הסף. אז, שני צדדי המשוואה כמעט שוים, ולכן גם ההפרש $p(v)-p(x)$ כמעט שווה להם. מכאן, שההפרש $p(v)-p(x)$ חייב להיות שווה לערך הסף - הערך שבו הפונקציה c מתחלפת מ-0 ל-1.

שימו לב: ערך-הסף יכול להיות זהה לכל השחקנים, אבל יכול להיות גם שונה משחקן לשחקן. דוגמאות:

- אם הכלל הוא "בחר את כל הערכים הגדולים מ-10", אז ערך-הסף לכל השחקנים הוא 10.
- אם הכלל הוא "בחר את 3 הערכים הגבוהים ביותר", אז ערך-הסף של כל שחקן שנבחר הוא הערך של השחקן הרביעי.
- אם הכלל הוא "הרץ אלגוריתם חמדני ב" - אז ערך-הסף של כל שחקן יהיה תלוי במשקל שלו (נראה בהמשך).

הוכחנו שיש רק כלל-תשלום יחיד שהוא מועמד להיות אמיתי. המכרז עם כלל-התשלום הזה נראה כך:

- כלל-הבחירה הנתון קובע, עבור כל שחקן i, ערך-סף מסויים t_i.
- אם $v_i > t_i$, אז השחקן נבחר ומשלם t_i.
- אחרת, השחקן לא נבחר ולא משלם.

נשאר להוכיח, שכלל-התשלום הזה הוא אמיתי. הנה ההוכחה:

- אם נבחרת, ותכריז מעל t_i - כלום לא ישתנה, אתה עדיין תיבחר ותשלם את אותו ערך.
- אם לא נבחרת, ותכריז מתחת ל t_i - כלום לא ישתנה, אתה עדיין לא תיבחר ולא תשלם.
- אם נבחרת, ותכריז מתחת ל-t_i - אתה לא תיבחר והתועלת שלך תהיה 0, בעוד שקודם לכן התועלת שלך היתה חיובית (כי $v_i > t_i$).
- אם לא נבחרת, ותכריז מעל t_i - אתה תיבחר ותשלם t_i, והתועלת שלך תהיה שלילית (כי $v_i < t_i$).

***

מכרז מיירסון למילוי תרמיל

עכשיו נחזור לבעיית התרמיל ונפתור אותה בעזרת מכרז מיירסון.
אנחנו מניחים שהמשקל של כל משתתף ידוע; כל משתתף צריך להגיד רק את הערך שלו.
נבדוק כמה כללי-בחירה אפשריים.

אלגוריתם חמדני א: סדר את החפצים בסדר יורד של הערך, ובחר חפצים לפי הסדר עד שהתרמיל מתמלא. האלגוריתם הזה הוא לא כל כך יעיל, אבל הוא מונוטוני! שחקן שמגדיל את הערך שלו, רק מגדיל את סיכוייו להיבחר. לכן אפשר להשתמש במכרז מיירסון. לדוגמה, אם יש 51 חפצים עם הערכים הבאים:

\$100/100k, \$20/2k, \$20/2k, \$20/2k ...

אז רק הראשון נבחר, והוא משלם את ערך הסף שלו, שהוא \$20 (כי אם היה אומר פחות מ-\$20, הכלל לא היה בוחר אותו).

אלגוריתם חמדני ב: סדר את החפצים בסדר יורד של ערך/משקל, ובחר חפצים לפי הסדר עד שהתרמיל מתמלא. גם האלגוריתם הזה הוא מונוטוני - שחקן שמגדיל את הערך שלו (כשהמשקל נשאר קבוע), רק מגדיל את סיכוייו להיבחר. נשתמש במכרז מיירסון. לדוגמה, אם יש 3 חפצים עם הערכים הבאים:
$20/2k, $5/1k, $100/100k.

שני הראשונים נבחרים, והפעם לכל אחד יש ערך-סף שונה:

- ערך-הסף של הראשון הוא $2 - כי אילו היה אומר פחות מ-$2, יחס ערך/משקל שלו היה קטן משל החפץ השלישי, והוא לא היה נבחר.
- ערך-הסף של השני הוא $1 - מאותה סיבה.

לכן הראשון ישלם $2 והשני ישלם $1.

אלגוריתם א+ב: הפעל את שני האלגוריתמים החמדניים. בחר את התוצאה עם הסכום הגבוה.
גם האלגוריתם הזה מונוטוני. הנה כמה דוגמאות לתוצאה של מכרז מיירסון:
$54/52k, $52/51k, $49/49k

הראשון זוכה ומשלם: (52k/51k) * $52

$100/100k, $20/2k, $20/2k

הראשון זוכה ומשלם $40.

$100/100k, $60/2k, $60/2k

שני האחרונים זוכים ומשלמים: ?

סיכום: ויקרי-קלארק-גרובס לעומת מיירסון

ראינו שתי דרכים לבנות מכרזים אמיתיים - לכל דרך יש יתרונות וחסרונות:

- היתרון של וק"ג - שהוא עובד גם כשהשחקנים הם רב-פרמטריים - יש להם הרבה ערכים שונים (למשל בבעיית בחירת המסעדות - לכל שחקן יש ערך לכל מסעדה). החסרון של וק"ג - שכלל-הבחירה היחיד שהוא יודע לעבוד איתו זה כלל "מיקסום סכום הערכים".
- היתרון של מיירסון - שהוא עובד עם כל כלל-בחירה מונוטוני; החיסרון שלו - שהוא יודע לעבוד רק עם שחקנים חד-פרמטריים - לכל שחקן יש רק מספר אחד המציין את הערך שלו ל"היבחרות". הוא לא יכול לפתור בעיות עם שחקנים רב-פרמטריים.

האם אפשר לשלב את היתרונות של שני המכרזים? האם אפשר לתכנן מכרז שעובד עם כלל-בחירה מגוון (לא מיקסום סכום הערכים), וגם עם שחקנים רב-פרמטריים? זו אחת השאלות הפתוחות החשובות ביותר בתחום.

מקורות

- ...

מאמרים להרחבה ולמטלת רשות

1. N Nisan, A Ronen (2001): "Algorithmic mechanism design"

2. D. Mishra et al (2014): "Multidimensional mechanism design in single peaked type spaces"

3. T. Roughgarden and I Talgam-Cohen (2019): "Approximately optimal mechanism design"

4. Chakrabarty and Swamy (2014): "Welfare maximization and truthfulness in mechanism design with ordinal preferences"

סיכם: אראל סגל-הלוי.