

## תיאום תורמים

האלגוריתמים של תקצוב השתתפותי, שהצגנו בחלק הראשון, נותנים לאזרחים בחירה מסויימת בשאלה על מה להוציא את כספי המיסים, אבל עדיין אין להם בחירה האם וכמה מיסים לשלם.

בחלק השני נדבר על אלגוריתמים מתקדמים יותר, שמטרתם לעודד את האזרחים לשלם מיסים מרצונם החופשי. החזון הוא, שבעתיד לא יהיה בכלל צורך בכפייה; בינתיים זה נראה חזון רחוק, אבל אנחנו כאנשי מדעי-המחשב חושבים קדימה וכבר מכינים את התשתית.

כדי להסביר למה בכלל צריך אלגוריתם, ואיך אלגוריתם יכול לעזור כאן, נתחיל בדוגמה. נניח שבעיר מסויימת יש שלושה מבנים – מגרש כדורסל, מועדון שחמט, וספריה ציבורית. המבנים כבר נבנו (הם תוקצבו, נניח, ע"י האלגוריתם לתקצוב השתתפותי שלמדנו קודם, או בשיטה אחרת כלשהי). אבל עכשיו צריך לתפעל אותם (לנקות, לשלם לשומרים וכד').

נניח שהתחזוקה של כל אחד מהמבנים האלה, לשעה יומית, עולה 1000 ש"ח לחודש. בימינו הכסף הזה מגיע מכספי מיסים הנגבים בכפייה, אבל לפי חזון החירות, בעתיד הוא יגיע מתרומות התנדבותיות של האזרחים. האזרחים ירצו לתרום אם הם יידעו שהתרומה שלהם תביא להם תועלת – בזכות התרומה שלהם, המבנים הללו יפעלו ויספקו שירותים להם ולבני-משפחותיהם.

לכל אזרח יש כמות שונה של כסף שהוא רוצה לתרום. לצורך הדוגמה, נניח שיש שני אזרחים – עמי ותמי, וכל אחד מהם מוכן לתרום 3000 ש"ח בחודש. אבל יש להם העדפות שונות: עמי מעדיף בילויים תחרותיים (כדורסל או שחמט), ותמי מעדיפה בילויים במקומות סגורים (שחמט או ספריה).

אם כל אחד מהם יתרום לבד, בלי תיאום, מסתבר שכל אחד יתרום כמות שווה לכל מקום שהוא רוצה לתמוך בו, והתוצאה תהיה: כדורסל = 1.5 שעות ביום, שחמט = 3 שעות ביום, ספריה = 1.5 שעות ביום. כתוצאה מכך, לכל אחד מהאזרחים יהיו 4.5 שעות בילוי ביום (1.5+3). זה כמובן טוב – טוב יותר ממה שכל אחד מהם יכל להשיג לבד עם 3000 ש"ח – אבל זה לא אופטימלי. יש אפשרות טובה יותר. האם יש לכם רעיון מהי?

- תשובה: אם גם עמי וגם תמי יתרמו למועדון השחמט, אז יהיה אפשר להפעיל אותו 6 שעות ביום, ואז התועלת של שניהם תהיה 6 – כלומר יש כאן שיפור פארטו חזק לעומת המצב שבו כל אחד תורם לבד. התהליך הזה נקרא **תיאום תורמים – donor coordination**. זהו תהליך שבו אנחנו אומרים לאנשים לאן כדאי להם לתרום, כדי להשיג שיפור-פארטו לגבי המצב שבו כל אחד תורם באופן עצמאי.

הערה: תיאום התורמים תלוי באופן משמעותי בפונקציות התועלת של האזרחים. בדוגמה למעלה הנחנו שפונקציית התועלת היא אדיטיבית – התועלת שכל אזרח מפיק מהתקציב שווה (עד כדי כפל בקבוע) לסכום התרומות לנושאים שהוא תומך בהם. אז כשאין תיאום, התועלת של כל אזרח היא 1500+3000, וכשיש תיאום, התועלת היא 0+6000.

היה אפשר להניח הנחות אחרות. למשל, נניח שהתועלת שכל אזרח מפיק מהתקציב שווה למינימום התרומות לנושאים שהוא תומך בהם. במקרה זה, כאשר אין תיאום, התועלת של כל אזרח שווה ל 1500 (מינימום בין 1500 ל-3000). כשיש תיאום, התיאום כאן יתבצע בצורה שונה: התיאום ימליץ לעמי לתרום 2000 לכדורסל ו-1000 לשחמט, ולתמי לתרום 1000 לשחמט ו-2000 לספריה. במצב זה התועלת של כל

אזרח תהיה 2000, כי כל אחד משני הנושאים שהוא תומך בהם מקבל תקציב של 2000. גם כאן, התיאום מביא לשיפור פארטו חזק לעומת מצב של חוסר-תיאום.

בהרצאה זו נתמקד במקרה שבו פונקציות התועלת הן אדיטיביות. ההרחבה לפונקציות תועלת אחרות היא נושא למחקר עתידי.

## הגדרת הבעיה

נתונים  $n$  אזרחים. לכל אזרח  $i$  יש סכום-כסף מסויים שהוא מוכן לתרום, נניח  $C_i$ . נסמן ב- $C$  את סכום כל התרומות.

יש  $m$  נושאים. לכל אזרח  $i$  יש תת-קבוצה  $A_i$  של נושאים שהוא תומך בהם.

המטרה היא לחשב וקטור  $D$  המגדיר את סכום התרומה לכל נושא, כך שסכום כל ההעברות הוא  $C$ .

התועלת של כל אזרח  $i$  מהוקטור  $D$  שווה לסכום כל התרומות שניתנו לנושאים מתוך  $A_i$ . לדוגמה, אם עמי תומך בשחמט וכדורסל, והתקציב נותן 1000 לשחמט, 2000 לכדורסל ו-4000 לספריה, אז התועלת שלו 3000.

אלגוריתם תיאום תורמים הוא אלגוריתם המקבל כקלט את הסכומים  $C_i$  והקבוצות  $A_i$ , ומוציא כפלט את הוקטור  $D$ .

## תכונות רצויות

יש כמה תכונות שהיינו רוצים שיהיו לאלגוריתם תיאום תורמים – את כולן כבר ראינו בהקשרים אחרים, וזו הזדמנות טובה לחזור על התכונות הללו ולראות איך הן מתבטאות בהקשר נוסף.

א. **יעילות פארטו**: לא קיים וקטור  $D'$  הנותן תועלת גדולה או שווה לכולם, ותועלת גדולה ממש למישהו.

ב. **עידוד השתתפות** (individual rationality): התועלת שכל אזרח מפיק מהשתתפות באלגוריתם גדולה לפחות כמו התועלת שהיתה לו אילו היה תורם באופן עצמאי.

ג. **גילוי אמת** (truthfulness): לכל אזרח  $i$  כדאי לגלות את הקבוצה  $A_i$  האמיתית, לא משנה מה עושים האזרחים האחרים.

## אלגוריתמים

נראה שלושה אלגוריתמים עם תכונות שונות. נדגים את כל האלגוריתמים על הדוגמה הבאה:

יש 4 נושאים (א, ב, ג, ד) ו-5 אזרחים, כל אחד תורם 100, והם תומכים בנושאים הבאים: **אב, אג, אד, בג, א**.

1. הדרך המקובלת להשיג יעילות פארטו, שכבר ראינו בעבר, היא למקסם פונקציה כלשהי. **האלגוריתם**

**האוטיליטרי** (utilitarian) מחשב וקטור הממקסם את סכום התועלות - הוקטור  $D$  שעבורו סכום התועלת של כל האזרחים הוא הגדול ביותר. איך מוצאים את  $D$  הזה? אחרי מעט מחשבה, קל להבין שכדי למקסם את סכום התועלות, צריך להשקיע את כל התקציב  $C$  רק בנושאים שמספר התומכים בהם הוא הגדול ביותר. במקרה הקיצוני, אם יש נושא אחד שתומכים בו, נניח, 1000 איש, ובשאר הנושאים תומכים

999 איש או פחות, אז האלגוריתם האוטיליטרי ייתן את כל התקציב לנושא זה, כי כל שקל שמשקיעים בנושא זה תורם 1000 לסכום התועלות, וכל שקל שמשקיעים בנושא אחר תורם פחות מ-1000. בדוגמה שלנו, כל הכסף יושקע בנושא **א** בלבד, ווקטור התקציב יהיה **(0, 0, 0, 500)**.

האלגוריתם האוטיליטרי הוא כמובן יעיל פארטו, כמו כל אלגוריתם הממקסם סכום של פונקציה עולה של התועלת. האלגוריתם הוא גם אמיתי. רעיון ההוכחה: צריך להוכיח, שאזרח  $i$  המשנה את הקבוצה  $A_i$  שלו, לא מגדיל את התועלת שלו. כלומר לא מגדיל את התקציב המועבר לנושאים בקבוצה  $A_i$  המקורית שלו. נבדוק שני מקרים: (א) האזרח מסתיר נושא מסויים הנמצא ב- $A_i$ . כתוצאה מכך, מספר התומכים של נושא זה קטן ב-1, ומספר התומכים של שאר הנושאים ב- $A_i$  לא משתנה. האלגוריתם מעביר תקציב רק לנושאים עם מספר תומכים מקסימלי, ולכן לנושאים ב- $A_i$  יש פחות סיכוי לקבל תקציב. (ב) האזרח מוסיף נושא מסויים שאינו נמצא ב- $A_i$  האמיתי. כתוצאה מכך, מספר התומכים של אותו נושא (שהאזרח לא באמת תומך בו) גדל ב-1. לכן, לנושאים שהאזרח באמת תומך בהם יש פחות סיכוי לקבל תקציב.

הבעיה הגדולה באלגוריתם האוטיליטרי היא שהוא לא מעודד-השתתפות: הוא עלול לקחת את כל התרומה של תורם מסויים, ולהשקיע אותה בנושא שהתורם הזה בכלל לא תומך בו, כך שהתועלת של התורם תהיה אפס (בדוגמה שלנו, התועלת של האזרח הרביעי, התומך בנושאים ב,ג, היא אפס).

2. **האלגוריתם האוטיליטרי-על-תנאי** (conditioned utilitarian) מחשב וקטור שבו סכום התועלות הוא הגדול ביותר תחת האילוץ שכל אזרח תורם רק לנושאים שהוא תומך בהם. איך מוצאים אותו? - אחרי מעט מחשבה, מבינים שכדי למקסם את סכום התועלות צריך להגיד לכל אזרח  $i$  להשקיע את כל התקציב שלו  $C_i$  בנושאים שמספר התומכים בהם הוא הגדול ביותר, מבין הנושאים שהוא תומך בהם. לדוגמה, אם תמי תומכת בשחמט וספריה, אבל יש עוד 10 אזרחים התומכים בשחמט ועוד 20 אנשים התומכים בספריה, אז האלגוריתם האוטיליטרי-על-תנאי יגיד לתמי להשקיע את כל התקציב שלה בספריה. בדוגמה שלנו, האלגוריתם ייתן את כל כספם של האזרחים התומכים ב-א לנושא א, ואת הכסף של האזרח הרביעי יחלק בין ב לבין ג, ווקטור התקציב הוא **(0, 50, 50, 400)**.

דוגמה נוספת: נניח שהאזרחים תומכים בנושאים הבאים: **בד, אג, אד, בג, א**. לנושא א יש שלושה תומכים, ולכן האלגוריתם אומר לכל אזרח שתומך בו לתת לו את כל כספו. לנושאים ב, ג, ד יש שני תומכים, ולכן האלגוריתם אומר לשאר האזרחים לחלק את התרומה שלהם ביניהם. וקטור התקציב הוא **(50, 50, 100, 300)**.

האלגוריתם האוטיליטרי-על-תנאי מעודד השתתפות: כיוון שכל אזרח משקיע את כל התרומה שלו בנושאים שהוא תומך בהם, הוא לא מפסיד שום דבר מהשתתפות באלגוריתם.

האלגוריתם הוא גם אמיתי - ההוכחה דומה לאלגוריתם האוטיליטרי.

הבעיה היא, שהאלגוריתם לא תמיד יעיל-פארטו. בדוגמה השניה, וקטור התועלות הוא: [150, 350, 350, 300]. אפשר להשיג שיפור-פארטו ע"י העברת 50 מ-ג ל-א והעברת 50 מ-ד ל-ב. מתקבל וקטור-תקציב **(0, 0, 150, 350)** עם וקטור תועלות: [150, 350, 350, 350] - האזרח החמישי מרויח ואף אחד לא מפסיד.

3. **אלגוריתם מיקסום המכפלה** מחשב וקטור  $D$  שבו מכפלת התועלות היא הגדולה ביותר. כדי לחשב את הוקטור  $D$  צריך לפתור בעיה של תיכנות ליניארי (כבר ראינו בשיעורים הקודמים שאפשר לפתור בעיות כאלו בקלות בפייתון, למשל ע"י הספרייה `cvxpy` או `scipy.linprog`. ראו דוגמה בתיקיית הקוד). האלגוריתם כמובן יעיל פארטו.

בדוגמה הראשונה **אב, אג, אד, בג, א** מתקבל בקירוב (0, 65, 65, 370).

בדוגמה השנייה **בד, אג, אד, בג, א** מתקבל בקירוב (0, 0, 200, 300).

בניגוד לאלגוריתם האוטיליטרי, כאן לא צריך להגיד בפירוש את התנאי "תחת האילוץ שכל אזרח תורם לנושאים שהוא תומך בהם" – התנאי הזה מתקיים אוטומטית! זה נובע מהמשפט הבא:

משפט: יהי  $D$  וקטור חלוקת-התקציב הממקסם את מכפלת הערכים. לכל נושא  $x$ , נסמן ב- $D(x)$  את התקציב המועבר ל- $x$ . לכל אזרח  $i$ , נסמן ב- $u_i(D)$  את התועלת הכוללת שלו מהוקטור  $D$ . אז לכל נושא  $x$  מתקיים:

$$\sum_i (C_i / u_i(D)) = 1$$

כאשר הסכום הוא על כל האזרחים התומכים בנושא  $x$ .

הוכחה: במאמר.

מהמשפט נובע, שכדי לממש את כלל מיקסום המכפלה, מספיק להגיד לכל אזרח  $i$  לחלק את הכסף שלו  $C_i$  באופן הבא (אחרי שחישבנו את  $D$  ואת התועלות):

- לכל נושא בקבוצה  $A_i$ , לתרום  $D(x) * (C_i / u_i(D))$
- לכל נושא מחוץ לקבוצה  $A_i$ , לתרום 0.

המשפט מבטיח, שסכום התרומות לכל נושא  $x$  הוא בדיוק  $D(x)$ . כלל-החלוקה מבטיח שהאלגוריתם מעודד השתתפות – כל אזרח תורם את כל כספו רק לנושאים שהוא תומך בהם. למעשה, ניתן לממש את האלגוריתם בכלל בלי לגבות כסף מהאזרחים: האלגוריתם רק ממליץ לכל אזרח איך "להשקיע" את כספי התרומות שלו.

למשל, בדוגמה הראשונה, האלגוריתם ממליץ (ראו בקוד):

- לאזרח הראשון – להשקיע בנושא א 85 ובנושא ב 15;
- לאזרח השני – להשקיע בנושא א 85 ובנושא ג 15;
- לאזרח השלישי והחמישי – להשקיע בנושא א את כל ה-100;
- לאזרח הרביעי – להשקיע בנושא ב 50 ובנושא ג 50.

הבעיה באלגוריתם מיקסום-המכפלה, שהוא אינו מגלה-אמת. נניח שהקלט האמיתי הוא כמו בדוגמה הראשונה, ועכשיו האזרח הראשון מחליט במקום להגיד **אב** להגיד **בד**, כמו בדוגמה השנייה. אז אפשר לראות שהאזרח הראשון מרויח – כשהוא אומר אמת (אב) התועלת שלו היא  $435 = 370 + 65$ , וכשהוא אומר לא-אמת (בד) התועלת האמיתית שלו (סכום הכסף המושקע בנושאים שהוא באמת תומך בהם – א+ב) היא  $500 = 300 + 200$ !

## טרילמה

האם אפשר להשיג בו-זמנית את כל שלוש התכונות – יעילות-פארטו, עידוד-השתתפות, וגילוי-אמת? התשובה היא לא! ההוכחה לכך נמצאה ע"י תוכנת מחשב להוכחות אוטומטיות (ראו במאמר). אבל מתוך ניתוח הפלט של התוכנה, אפשר להוכיח משפט קצת יותר חלש:

משפט: לא קיים אלגוריתם המקיים בו-זמנית את התכונות הבאות:

- יעיל-פארטו;
- מגלה-אמת;
- מעודד-השתתפות (או אפילו מבטיח תועלת גדולה מאפס לכל משתתף);
- אנונימי - לא מושפע משינוי הסדר בין האזרחים (כל האלגוריתמים שראינו למעלה הם אנונימיים – הם מתייחסים רק לקבוצות הנושאים הרצויים ולא לסדר שבו מכניסים את הקלט לאלגוריתם. דוגמה לאלגוריתם לא אנונימי שכבר ראינו: "חתוך ובוחר" – התוצאה מושפעת משינוי סדר המשתתפים. דוגמה לאלגוריתם אנונימי לחלוקת עוגה: אבן-פז).
- נייטרלי - לא מושפע משינוי הסדר בין הנושאים (כל האלגוריתמים שראינו למעלה הם נייטרליים – הם מתייחסים רק לקבוצת התומכים בכל נושא, ולא לסדר בין הנושאים).

הוכחה: נניח בשלילה שקיים אלגוריתם כזה, ונבדוק מה אמור להיות הפלט שלו על הקלט של הדוגמה מלמעלה: **אב, אג, אד, בג, א.**

בקלט המקורי, הנושאים **בג**, הם סימטריים לחלוטין: כל אחד מהם מופיע פעם אחת עם נושא **א**, ועוד פעם אחת הם מופיעים יחד. כיוון שהאלגוריתם הוא נייטרלי ואנונימי, הוא חייב לתת לשניהם אותו סכום. והסכום הזה חייב להיות חיובי, כי אזרח הרביעי תומך רק בנושאים **אלה**, והוא חייב לקבל תועלת חיובית.

עכשיו נניח שהאזרח הראשון, במקום להגיד **אב**, אומר **בד**. אז הקלט הופך להיות: **בד, אג, אד, בג, א.** מה יחזיר האלגוריתם שלנו על הקלט החדש? בקלט החדש, הנושאים **גד**, הם סימטריים (שניהם מופיעים פעם אחת עם **ב** ופעם אחת עם **א**), ולכן האלגוריתם חייב לתת להם אותו סכום. עכשיו יש שני מקרים:

- מקרה ראשון: האלגוריתם נותן לנושאים **גד**, ד סכום אפס. כלומר הוא מחלק את כל הכסף בין נושאים **א,ב**. במקרה זה, האזרח הראשון מרויח מאי-גילוי-אמת! כי בקלט המקורי (האמיתי) חלק מהכסף הלך לנושא **ג** שהוא לא תומך בו, ובקלט החדש (הלא-אמיתי) כל הכסף הולך רק לנושאים **א,ב** שהוא תומך בהם.
- מקרה שני: האלגוריתם נותן לנושאים **גד**, ד סכום חיובי ממש (וזהה). במקרה זה, התוצאה אינה יעילה פארטו: אפשר להעביר את כל הסכום מנושא **ג** לנושא **א**, ומנושא **ד** לנושא **ב**. זה שיפור פארטו: לארבעת האזרחים הראשונים זה לא משנה (כי כל אחד מהם תומך בנושא אחד מבין **א,ב**, ובנושא אחד מבין **ג,ד**), אבל האזרח החמישי מרויח.

הגענו לסתירה: או שהאלגוריתם אינו אמיתי, או שאינו יעיל פארטו.

שוב הגענו למצב המוכר של טרילמה – שלוש תכונות רצויות, שאפשר להשיג רק שתיים מתוכן.

## מקורות

- *FLORIAN BRANDL, FELIX BRANDT, DOMINIK PETERS, CHRISTIAN STRICKER, WARUT SUKSOMPONG (2019), "Donor Coordination", <http://brandlf.com/docs/donate.pdf>.*

סיכום: אראל סגל-הלוי.