

אופטימיזציה דיסקרטית - 67855

14 בינואר 2013

מרצה: יובל רבני

איני לוקחת אחריות על מה שכתוב כאן, so tread lightly
אין המרצה או המתרגל קשורים לסיכום זה בשום דרך.
הערות יתקבלו בברכה - noga.rotman@gmail.com. אהבתם? יש עוד!
<http://bit.ly/integrali>

תוכן עניינים

3	מנהלות	0.1	
3	תכנון לינארי	1	
3	הגדרה	1.1	
4	אלגוריתם ראשון - שיטת הסימפלקס (Dantzig, 1947)	1.2	
5	דואליות	1.3	
7	מסקנות מעניינות מדואליות	1.3.1	
8	אלגוריתם האליפסואיד (1979)	1.4	
8	הגדרות	1.4.1	
9	סקירה של פעולת האלגוריתם	1.4.2	
10	הוכחת תכונת האליפסואיד	1.4.3	
12	הכח של האלגוריתם	1.4.4	
13	חיתוך מטרואידים	2	
13	הגדרות	2.1	
13	דוגמאות לבעיות שניתן לפתור בעזרת חיתוך מטרואידים	2.2	
13	זיווג מקסימום בגרף דו צדדי	2.2.1	
13	עץ פורש מכוון עם שורש r	2.2.2	
13	כיוון קשתות	2.2.3	
13	עץ פורש צבעוני	2.2.4	
14	שימוש בחיתוך מטרואידים לפתרון בעיות - דיון כללי	2.3	
15	מציאת עץ פורש מינימום מכוון	2.4	
17	הסכמה הפרימלית דואלית	3	
17	עץ שטיינר	3.1	
19	בעיית $K - MST$	3.2	
19	כופלי לגראנג'	3.2.1	
20	הבעיה $K - MST$	3.2.2	
23	שיטת העדכון הכפלי - Multiplicative Weights Update Method	3.3	
23	מוטיבציה	3.3.1	
23	דוגמא ראשונה - מחשיבה לימודית	3.3.2	
25	המשחק המוכלל	3.3.3	
26	הגדרת שיטת העדכון הכפלי	3.3.4	
26	משחקים סופיים סכום אפס בשני שחקנים	3.3.5	

0.1 מנהלות

יהיו ארבעה תרגילים (או אולי שלושה?), הציון יתבסס כולו עליהם. מי יבדוק? לא ברור. הספרים שנשתמש בהם:

- A. Schrijver. Theory of Linear and Integer Programming. Wiley, 1999
 A. Schrijver. Combinatorial Optimization: Polyhedra and Efficiency. Springer, 2003
 S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. Theory of Computing, 8:121-164, 2012

22/10/2012

יושלים בקרוב! (:

1 תכנון לינארי**1.1 הגדרה**

29/10/2012

נביט על תוכנית לינארית בהצגה סטנדרטית:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \\ & c, x \in \mathbb{R}^n \\ & A \in \mathbb{R}^{m \times n} \\ & b \in \mathbb{R}^m \end{aligned}$$

הקלט הוא הוקטור c , המטריצה A והוקטור b , ואנו מחפשים וקטור x שמזער את הביטוי מעלה. יש תחום פיזיבילי שמקיים את הדברים הללו - פאון מסויים. הראנו שתמיד יש קודקוד מסויים שהוא פתרון אופטימלי.

משפט 1.1 נניח ש- x מקיים את האילוצים של התוכנית הלינארית:

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned}$$

תהי A_x המטריצה שהעמודות שלה הן העמודות של A שמתאימות לאיברים של x ששונים מ-0. אזי x קודקוד של התחום:

$$P = \{y | Ay = b, y \geq 0\}$$

אמ"מ העמודות של A_x בת"ל.

הערה 1.2 זה אומר שכל קודקוד של x מתאים לאוסף של עמודות בת"ל מתוך A , אבל זה לא אומר שכל אוסף של עמודות בת"ל של A מתאים לקודקוד x כלשהוא.

כלומר, אם נסמן ב- x^+ את הוקטור המקבל מ- x ע"י מחיקת האיברים שהם 0 ב- x , אזי המשפט אומר כי:

$$x^+ = A_x^{-1}b$$

הוכחה: נראה את השלילה של שני הכיוונים.

בכיוון הראשון, נניח ש- x איננו קודקוד. בפרט, קיים וקטור $y \neq 0$ עבורו גם $x - y \in P$ וגם $x + y \in P$, ולכן:

$$A(x - y) = A(x + y) = 0$$

ולכן בהכרח $Ay = 0$.

מכך נקבל כי עמודות A_y (מההנחה יש לפחות עמודה אחת כזו כי $y \neq 0$) תלויות לינארית. ממה שהוכחנו בשיעור הקודם, $x - y, x + y \geq 0$ ולכן אם $x_j = 0$ אז $y_j = 0$, כלומר, העמודות של A_y הן תת קבוצה של העמודות של A_x . על כן, גם העמודות של A_x תלויות לינארית. בכיוון השני, נניח כי העמודות של A_x תלויות לינארית. בפרט, קיים $y \neq 0$ עבורו $A_x y = 0$. נרחיב את y לווקטור ב- $\mathbb{R}^n$ ע"י הוספת אפסים בקואורדינטות החסרות. נקבל $y \in \mathbb{R}^n, y \neq 0$, עבורו:

$$\begin{aligned} 1. & A_y = 0 \\ 2. & x_j = 0 \Rightarrow y_j = 0 \end{aligned}$$

לכן קיים קבוע $\lambda > 0$ עבורו $x - \lambda y, x + \lambda y \geq 0$. אזי $x - \lambda y, x + \lambda y \in P$, ועל כן x איננו קודקוד. ■
עכשיו אנחנו מוכיחים לראות אלגוריתם לפתירת בעיית התכנון הלינארי.

1.2 אלגוריתם ראשון - שיטת הסימפלס (Dantzig, 1947)

יהי $\bar{x}$ קודקוד של $P = \{x | Ax = b, x \geq 0\}$. נניח ש- $rank(A) = m$, ונסמן $B = \{j | \bar{x}_j > 0\}$. נניח בשלב ראשון ש- $|B| = m$. נסמן $A_B = A_{\bar{x}}$, וכן נסמן ב- $\bar{x}_B$ את האיברים שהאינדקסים שלהם ב- B . לסיום, נסמן: $N = \{1, \dots, n\} \setminus B$. אזי אפשר להציג את $\bar{x}$ ע"י:

$$\begin{aligned} \bar{x}_B &= A_B^{-1} b > 0 \\ \bar{x}_N &= 0 \end{aligned}$$

במקרה ש- $|B| < m$ נוסיף ל- B עמודות בלתי תלויות. ל- B קוראים בסיס. ל- $\bar{x}$ קוראים פתרון בסיסי של התוכנית¹. נכתוב מחדש את התוכנית באופן הבא:

$$\begin{aligned} \min & c_B^T x_B + c_N^T x_N \\ \text{s.t.} & A_B x_B + A_N x_N = b \\ & x_B, x_N \geq 0 \end{aligned}$$

אזי:

$$x_B = A_B^{-1} (b - A_N x_N)$$

מכאן נוכל לרשום את פונקציית המטרה רק כפונקציה של x_N :

$$\min c_B^T A_B^{-1} b - c_B^T A_B^{-1} A_N x_N + c_N^T x_N$$

נגדיר:

$$\begin{aligned} \tilde{c}^T &= c_N^T - c_B^T A_B^{-1} A_N x_N \\ a &= c_B^T A_B^{-1} b \end{aligned}$$

ונקבל את ההצגה הלא סטנדרטית הבאה:

$$\begin{aligned} \min & a + \tilde{c}^T x_N \\ & A_B x_B + A_N x_N = b \\ & x_B, x_N \geq 0 \end{aligned}$$

נדאג ש- $b \geq 0$, ו- A_B תהיה מטריצת יחידה (אם לא, נדאג שתהיה כזו בעזרת אלימינציה גאוס - נפתור מערכת משוואות).

יש לנו שתי אפשרויות:

1. האפשרות הפשוטה: אם $\tilde{c} \geq 0$, אין שום פתרון שיתן ערך יותר קטן (כי המתאימים ל- x_N הם 0 במטריצה), ולכן הקודקוד שלנו $\bar{x}$ הוא פתרון אופטימלי.

¹השם הגיוני, שכן ראינו שמתאים לו בסיס כלשהוא (אם $|B| = m$ הוא יחיד).

2. אפשר לשפר את פונקציית המטרה, כלומר קיים j עבורו $\tilde{c}_j < 0$.

במקרה השני, נרצה לקחת את הפתרון שלנו $\bar{x}$, ונרצה להגדיל את $\bar{x}_j = 0$ ($j \in N$) - אם נגדיל אותו פונקציית המטרה תרד (כי המקדם המתאים שלילי). אבל אם נגדיל לאו דווקא נקבל פיתרון פיזיבלי, לכן כאשר מגדילים יש להקטין (או לשנות בהתאם) מקדמים אחרים במערכת המשוואות כדי שהמשוואות לא תופרנה. כלומר, בתמורה צריך לשנות $\bar{x}_i$ -אים עבור $i \in B$ כדי לשמור על פיזיביליות. אפשר להניח (אם התוכנית חסומה, אחרת נוכל להמשיך ולשנות ולא נעצור) שלפחות אחד ה- $\bar{x}_i$ -אים הללו צריך לקטון. נגדיל את $\bar{x}_j$ עד שאיזה שהוא $\bar{x}_i$ כזה יתאפס. בנקודה זו נוציא את i מ- B ונכניס את j , ונקבל בסיס חדש, עבורו פונקציית המטרה ממש יותר קטנה מהערך אותו התחלנו. כעת אפשר לבצע שוב את התהליך הזה שוב ושוב עד שנגיע לאפשרות הראשונה.

הערה 1.3 יש כאן בחירה, שכן יתכן ועבור הרבה j , $\tilde{c}_j = 0$.

הערה 1.4 יתכן וניכנס כאן ללולאה אינסופית, כך יש לבדוק אם הגענו לאותו בסיס. בעיקרון קיים תנאי שניתן להוסיף כדי לוודא שלא תהיה כזו לולאה, אבל הוא מסבך מאוד את הפעולה ולכן לא משתמשים בו.

הערה 1.5 בשנות השבעים הוכיחו Klee ו-Minty שיתכן והאלגוריתם הזה יפעל בזמן אקספוננציאלי. היום לא מכירים תנאי שיגרום לאלגוריתם לנוע בזמן פולינומי. מצד שני, באופן מעשי האלגוריתם הוא מאוד יעיל (בערך לינארי במספר האילוצים ולוגריתמי במספר המשתנים), ואין לו ממש מתחרה היום (המתחרים מאוד מסובכים). גיל קלעי למשל מצא תנאי שנותן תשובה בזמן שאינו אקספוננציאלי.

נשאלת השאלה, איפה מוצאים קודקוד להתחיל איתו את הפעולה? נחזור לתוכנית המקורית:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

נוסיף משתנים חדשים $x'_1, x'_2, \dots, x'_m$, ונסתכל על התוכנית:

$$\begin{aligned} \min \quad & \sum_{i=1}^m x_i \\ \text{s.t.} \quad & Ax + Ix' = b \\ & x, x' \geq 0 \end{aligned}$$

כאן קל כבר למצוא קודקוד:

$$x = 0, x' = b$$

נריץ את תוכנית הסימפלקס על התוכנית החדשה, ונקבל קודקוד אופטימלי לתוכנית החדשה. אם $x' = 0$, אז נוכל ממנו להגיע לקודקוד של התוכנית המקורית. אם $x' > 0$, לא קיים פתרון לתוכנית המקורית.

1.3 דואליות

זהו המושג הכי חשוב שנלמד בקורס. נביט על תוכנית לינארית:

$$\begin{aligned} z = \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

נסמנה ב- P . אנחנו רוצים לייצר חסם תחתון על z בצורה מאוד מסויימת. כיצד נעשה זאת? ניקח קומבינציה של השורות של A שחסומה מלמעלה ע"י c :

$$\begin{aligned} y^T A &\leq c^T \\ \Rightarrow y^T A x &\leq c^T x \end{aligned}$$

(x כאן הוא פתרון אופטימלי לתוכנית הלינארית). אנו גם יודעים כי $y^T A x = y^T b$. נקבל:

$$y^T b \leq c^T x = z$$

לכן:

$$\begin{aligned} w &= \max y^T b \\ \text{s.t. } y^T A &\leq c^T \end{aligned}$$

כאשר $w \leq z$. קיבלנו תוכנית לינארית חדשה - נסמנה ב- D . D נקראת התוכנית הדואלית ל- P . למעשה הוכחנו כאן את משפט הדואליות החלש:

משפט 1.6 $w \leq z$.

מכאן נגיע למשפט הרבה פחות אינטואיטיבי - משפט הדואליות החזק:

משפט 1.7 $w = z$.

נוכיח אותו בהמשך.

טענה 1.8 התוכנית הדואלית של D היא P .

הוכחה: כדי להוכיח זאת, תחילה יש להביא את D לצורה יותר נוחה:

$$\begin{aligned} \max \{b^T y \mid A^T y \leq c\} &= \min \{-b_y^T \mid A^T y + Is = c, s \geq 0\} \\ y=y^+-y^- \min \{-b^T y^+ + b^T y^- \mid A^T y^+ - A^T y^- + Is = c, y^+, y^-, s \geq 0\} \end{aligned}$$

אזי, התוכנית הדואלית של D תהיה:

$$\begin{aligned} \max \{c^T (-x) \mid -A(-x) \leq b, A(-x) \leq -b, I(-x) \leq 0\} \\ = \min \{c^T x \mid Ax = b, x \geq 0\} \end{aligned}$$

■

וזו בדיוק התוכנית P איתה התחלנו!

נוכיח את משפט הדואליות החזק: **הוכחה:** בשיעור הקודם הוכחנו את הלמה של פרקש (יש לה כמה הצגות, בהתאם להצגה של התוכנית הלינארית):

למה 1.9 בדיוק אחד המקרים הבאים מתקיים:

1. קיים y עבורו $A^T y \leq c$.

2. קיים $x \geq 0$ עבורו $Ax = 0$ אבל $c^T x < 0$.

הראנו כבר כי $w \leq z$, לכן די להראות כי $z \leq w$. ישנם שני מקרים (קלים):
נניח שיש ל- P פתרון פיזיבילי. אם P לא חסומה, אזי $z = -\infty$, ולכן לפי משפט הדואליות החלש $w = -\infty$.
אחרת, יש פיתרון אופטימלי x^* ל- P . אזי:

$$\begin{aligned} Ax^* &= b \\ c^T x^* &= z \\ x^* &\geq 0 \end{aligned}$$

רוצים להראות שיש y עבורו $A^T y \leq c$, ו- $b^T y \geq z$. נניח בשלילה שאין y כזו. אזי למערכת:

$$\begin{pmatrix} A^T \\ -b^T \end{pmatrix} y \leq \begin{pmatrix} c \\ -z \end{pmatrix}$$

אין פיתרון. כלומר, מהלמה של פרקש, קיים $x \geq 0$ עבורו:

$$\begin{pmatrix} A^T \\ -b^T \end{pmatrix}^T x = 0, \begin{pmatrix} c \\ -z \end{pmatrix} x < 0$$

נסמן $x = x'$ ללא האיבר האחרון x_{n+1} . אזי:

$$x', x_{n+1} \geq 0, Ax' = x_{n+1}b$$

$$c^T x' < x_{n+1}z$$

מקרה א' - $x_{n+1} = 0$. במקרה זה:

$$Ax' = 0, c^T x' < 0$$

במקרה זה $x^* + x'$ הוא פיתרון אפשרי ל- P שערכו קטן מ- z , וזו סתירה. מקרה ב' - $x_{n+1} > 0$. במקרה זה $\frac{x'}{x_{n+1}}$ הוא פתרון שערכו:

$$c^T \left(\frac{x'}{x_{n+1}} \right) < z$$

וזה גם סתירה. לכן קיים y כזה, כנדרש. ■

1.3.1 מסקנות מעניינות מדואליות

נביט בתוכנית:

$$\begin{aligned} \min c^T x \\ \text{s.t. } Ax = b \\ x \geq 0 \end{aligned}$$

ובתוכנית הדואלית שלה:

$$\begin{aligned} \max y^T b \\ \text{s.t. } y^T A \leq c^T \end{aligned}$$

יהי x פתרון פרימלי פיזיבילי ו- y פתרון דואלי פיזיבילי. לפי דואליות חלשה, אנחנו יודעים ש- $c^T x \geq y^T b$.

הגדרה 1.10 ההפרש $c^T x - y^T b$ נקרא **פער הדואליות** של x, y .

נסמן: $s^T = c^T - y^T A$ (מציין כמה רחוק מלהיות הדוק). אזי:

$$c^T x - y^T b = c^T x - y^T Ax = (c^T - y^T A)x = s^T x$$

אם גם x וגם y פתרונות אופטימליים, אז הביטוי מעלה שווה לאפס. איך זה יקרה? $x \geq 0$ לפי התנאי, ו- $s \geq 0$ לפי סוג האילוצים בתוכנית הדואלית. אזי בכל מקום בו x אינו אפס, s חייב להיות אפס. נראה כי זהו תנאי הכרחי ומספק להיות הפתרונות אופטימליים.

משפט העודפות המשלימה (Complementary slackness):

משפט 1.11 התנאים הבאים שקולים עבור זוג פתרונות פיזיביליים x, y :

1. x, y אופטימליים.

2. $s^T x = 0$.

3. $s_j x_j = 0$ לכל j .

4. אם $s_j > 0$, אז $x_j = 0$.

המעבר בין 1 ל-2 נובע מדואליות חזקה, והשאר כבר הוכחנו. המשפט הזה נותן לנו לפעמים תובנות קומבינטוריות מעניינות.

5/11/2012

1.4 אלגוריתם האליפסואיד (1979)

זהו אלגוריתם הפותר בעיות לינאריות בזמן פולינומי. אין לאלגוריתם הזה משמעות מעשית, שכן לבעיות "אמיתיות" הוא נותן זמן ארוך הרבה יותר מאלגוריתם הסימפלקס. ישנה ספרות ענפה מאוד בנושא. בכל זאת מלמדים אותו, מדוע?

- יש אלגוריתמים פולינומיים שהם טובים לפחות כמו הסימפלקס, הבעיה היא שהם הרבה יותר מסובכים גם לתיאור וגם להבנה, וכל הסיבוך הזה מאוד טכני ולא מראה משהו יפה או מיוחד.
- לאלגוריתם הזה יש חשיבות היסטורית מאוד גדולה. מי שהוכיח זמן פולינומי עבורו הוא קצ'ן ($sp?$) במהלך המלחמה הקרה, והיכה את האמריקאים בתדהמה.
- יש לאלגוריתם הזה חשיבות תיאורית מאוד גדולה, שכן אפשר לפתור בעזרתו בזמן פולינומי בעיות שאנחנו לא יודעים לפתור אחרת (מלבד הבעיה של תכנון לינארי כפי שהצגנו אותה כאן), למשל בעיות תכנון לינארי שלא נתונות בצורה מפורשת.

תוכנית עבודה:

- נגדיר מהו אליפסואיד.
- סקירה ממעוף הציפור על האלגוריתם.
- נעבור על החישובים הנדרשים לכל ההוכחות.

1.4.1 הגדרות

הגדרה 1.12 מטריצה D סימטרית ממשית $n \times n$ היא מוגדרת חיובית אמ"מ התבנית הריבועית $x^T D x > 0$ לכל $x \in \mathbb{R}^n$, $x \neq 0$.

נזכר בכמה דברים שאנחנו יודעים על מטריצות שכאלה:
 D סימטרית ממשית ניתנת להצגה באופן RR^T , וכן R איננה סינגולרית (R מוגדרת ע"י וקטורים עצמיים של D), כלומר ל- D אין אף ערך עצמי שהוא 0. כל הע"ע של D > 0 .

$$\lambda_k(D) = \max_{x \neq 0, x \perp v_1, \dots, v_{k-1}} \frac{x^T D x}{x^T x}$$

הגדרה 1.13 אליפסואיד הוא קבוצה מהצורה:

$$E = E(D, y) = \left\{ x \in \mathbb{R}^n \mid (x - y)^T D^{-1} (x - y) \leq 1 \right\}$$

כאשר D מטריצה $n \times n$ מוגדרת חיובית, ו- $y \in \mathbb{R}^n$.

נשים לב כי אם D מטריצה מוגדרת חיובית, כך גם D^{-1} , שכן:

$$D = RR^T \Rightarrow D^{-1} = (R^{-1})^T R^{-1}$$

כמו כן, נשים לב כי אליפסואיד זוהי הכללה של כדור:

$$\begin{aligned} B &= B(r, y) = E(r^2 I, y) = \left\{ x \in \mathbb{R}^n \mid (x - y)^T \frac{1}{r^2} I (x - y) \leq 1 \right\} \\ &= \left\{ x \in \mathbb{R}^n \mid (x - y)^T (x - y) \leq r^2 \right\} \end{aligned}$$

נסמן ב- B_n את כדור היחידה סביב 0: $\mathbb{R}^n \supseteq B_n(1, \vec{0})$

1.4.2 סקירה של פעולת האלגוריתם

נסמן:

$$P_{<} = \{x \in \mathbb{R}^n \mid Ax < b, b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}\}$$

אנו נעבוד עם אילוצים שכאלו ממימד מלא (נראה איך מגיעים לזה). במקרה הזה נראה כי הפתרון נמצא על השפה של הפאון, כלומר כאן נעבוד על הפנים (בניגוד לניתוח הקודם שלנו) - העבודה היא בעצם מחוץ לפאון, וברגע שנצליח להגיע לתוך הפאון פתרנו את הבעיה.

בהנתן אוסף אילוצים כאלה, אלגוריתם האליפסואיד מוצא $x \in P_{<}$ או מראה ש- $P_{<} = \emptyset$. מה שעוד גרוע בענין הוא שאלגוריתם האליפסואיד מצליח אם הקלט A, B מקיים את התכונות הבאות:

• ידוע s עבורו $P_{<} \subseteq B(s, \vec{0})$, כלומר יש כדור שחוסם את הקלט.

• אם הקלט מקיים $P_{<} \neq \emptyset$, אזי ידועים r, y עבורו:

$$B(r, y) \leq P_{<}$$

למרבה המזל, תוכניות לינאריות תמיד מקיימות תכונות אלו, וזה מה שקצ'ן הוכיח. זהו חלק טכני למדי שאולי לא נראה.

שתי התכונות הללו נותנות לנו חסם עליון U וחסם תחתון L על הנפח של הקבוצה $P_{<}$. האלגוריתם עובד באיטרציות. באיטרציה t נתון לנו אליפסואיד:

$$E^t = E(D^T, x^t)$$

(בהתחלה $E^0 = B(s, \vec{0})$). נשים לב כי קבוצה זו בכל איטרציה תשמור על האיזומורפיזם של הכלת $P_{<}$. אנו כבר מכירים קצת את האלגוריתם הזה - הוא פועל כמו "אריה במדבר". כיצד הוא עובד? נבדוק אם $x^t \in P_{<}$ - עוברים על האילוצים אחד אחד ובודקים אותם. אם היא לא בקבוצה, למעשה מצאנו אילוץ שהופר, כלומר:

$$\begin{aligned} a_i^T x &< b_i \in P_{<} \\ a_i^T x^t &\geq b_i \end{aligned}$$

אזי, קיים על מישור כלשהוא העובר דרך המרכז, אשר המשוואה שלו היא:

$$a_i^T x = a_i^T x^t$$

ש- x יושב עליו, ש- $P_{<}$ יושב כולו בצד שלו. אזי מצאנו את ה"חצי של המדבר" שם יושב ה"אריה" שלנו. אבל נשים לב כי חצי אליפסואיד הוא אינו אליפסואיד, ואם נמשיך כך הביטוי יעשה יותר ויותר מורכב, ולאו דווקא יהיה קיים מרכז שקל לחשב אותו. על כן, כיצד נתקדם הלאה?

אזי, במקום לקחת את מחצית האליפסואיד, נחשב את האליפסואיד המינימלי בנפח החוסם את חצי האליפסואיד שמצאנו (לא יהיה חשוב לנו שהוא מינימלי), ונשתמש בו באיטרציה הבאה.

כאן נכנסת ההבחנה המהותית היחידה המסתתרת באלגוריתם הזה: חצי אלפסואיד בוודאי שיש לו נפח קטן מהאליפסואיד המקורי. אם יכולנו לעבוד עם חצי כל הזמן, מספר האיטרציות היה $\frac{\log(U)}{L}$. אבל, לא ברור כלל וכלל כי הנפח של אליפסואיד החדש קטן יותר מהנפח של האליפסואיד המקורי (זוהי תכונת האליפסואיד). על כן יש להוכיח כי נפחו קטן במידה מספיק משמעותית לעומת הנפח של האליפסואיד כולו.

מה נרוויח מכאן? לא נרוויח הרבה. נגדיר את האליפסואיד החדש להיות E^{t+1} , ונראה:

$$\text{vol}(E^{t+1}) \leq e^{-\frac{1}{2(n+1)}} \cdot \text{vol}(|E^t|)$$

נשים לב כי לפי פיתוח טיילור:

$$e^{-\frac{1}{2(n+1)}} \leq 1 - \frac{1}{2(n+1)} + \frac{1}{8(n+1)^2}$$

למה זה טוב? אם נריץ $2(n+1)$ איטרציות, מובטח לנו כי נפח האליפסואיד ירד בפקטור של e^{-1} , וזה המצב שרצינו - אחרי מספר לינארי של איטרציות הורדנו את נפח האליפסואיד בפקטור קבוע, וכל מה שנשאר הוא להפעיל $\log\left(\frac{U}{L}\right)$ פעמים, ואז מובטח שנמצא פיתרון.

טענה 1.14 אחרי $t^* = \lceil 2(n+1) \log \left(\frac{U}{L} \right) \rceil$ אם $x^{t^*} \notin P_{<}$ אז $P_{<} = \emptyset$.

איך פותרים תוכנית לינארית בעזרת אלגוריתם האליפסואיד?
נביט בתוכנית הנתונה באופן הבא:

$$\begin{aligned} z &= \max c^T x \\ \text{s.t. } Ax &\leq b \end{aligned}$$

אפשר לתרגם כל תוכנית לינארית לפורמט הזה. מלבד זאת, נרצה שהקבוצה הפיזיבילית תהיה או ריקה או ממימד מלא.

בעזרת האלגוריתם, נמצא $x_0 \in P_{<}$ אם x_0 הוא לא פיתרון אופטימלי, אפשר להוסיף אילוץ נוסף $-c^T x_0 < -c^T x$ לאוסף האילוצים שלנו. אם x_0 לא אופטימלי, $P_{<}$ החדש אינו ריק, ונוכל למצוא בתוכו נקודה, ואז נחזור על התהליך.

כל פעם שנמצא x כשזה, אנו משפרים את פונקציית המטרה של התוכנית.

טענה 1.15 אחרי לכל היותר:

$$N = 2n(n+1) \cdot \left\lceil \log \left(\frac{2s^2 \|c\|}{r\varepsilon} \right) \right\rceil$$

צעדים, נקבל $\hat{x}$ המקיים $c^T \hat{x} \geq z - \varepsilon$.

נוכיח את הלב של ההוכחה הזו - התכונה של האליפסואיד, אבל לא נוכיח אותה ממש.
עושה רושם כי אנו מקבלים קירוב "טוב כרצוננו". למה זה מאפשר לנו לפתור את התוכנית הלינארית? הרעיון הכללי הוא כזה:

קודם כל נשנה קצת את התוכנית הלינארית כך שיהיה לה פתרון אופטימלי יחיד, ומתוך זה אפשר לגזור איזשהוא פתרון לתוכנית המקורית (ע"י כפיית סדר על איברי הבסיס והוספה של "משהו" לפונקציית המטרה כך שתבודד קודקוד בודד). הרעיון: אפשר להוכיח (כנראה לא נראה - הוכחה מאוד טכנית ולא מעניינת) שקודקודים של הפאון ניתנים להצגה כך שכל קואורדינטה הוא איזשהיא מנה של מספרים "לא יותר מידי גדולים". לכן יש נקודה בודדת על השריג שקיבלנו שמועמדת להיות הפתרון. אזי כל מה שצריך הוא להגיע לאפסילון "הרבה יותר קטן" מהרזולוציה הדרושה, ואז ישנה נקודה אחת אפשרית. הרזולוציה כמובן תהיה אקספוננציאלית קטנה במספר הביטים הדרושים להצגה, אבל התלות של זמן הריצה באפסילון נותרת קטנה מספיק.

1.4.3 הוכחת תכונת האליפסואיד

הרעיון של ההוכחה הוא מאוד פשוט: נביט על החצי החיובי של כדור היחידה, ואליפסואיד ספציפי החוסם אותו. זה יספיק כי אפשר להפעיל טרנספורמציה אפינית על E^t לכדור היחידה, המשמרת יחס של נפחים, ואז מה שנוכיח לגבי הצורה הספציפית הזו יעבוד גם באופן כללי.
נסמן:

$$\begin{aligned} D &= \left(\frac{n^2}{n^2-1} \left(I - \frac{2}{n+1} e_1 e^T \right) \right) \\ E &= \left\{ x \in \mathbb{R}^n \left| \left(x - \frac{e_1}{n+1} \right)^T D^{-1} \left(x - \frac{e_1}{n+1} \right) \right. \right\} \\ y &= \begin{pmatrix} \frac{1}{n+1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \frac{1}{n+1} e_1 \end{aligned}$$

נניח $n > 2$. נרצה להוכיח:

1. D מוגדרת חיובית.

2.

$$\frac{\text{vol}(E)}{\text{vol}(B_n)} \leq e^{-\frac{1}{2(n+1)}}$$

.3

$$B_n \cap \{x \in \mathbb{R}^n | x_1 \geq 0\} \subseteq E$$

הוכחה: נוכיח תחילה את 1:

$$D = \begin{pmatrix} \frac{n^2}{n^2-1} \left(1 - \frac{2}{n+1}\right) & & & \\ & \frac{n^2}{n^2-1} & & \\ & & \ddots & \\ & & & \frac{n^2}{n^2-1} \end{pmatrix}$$

שכן מהנוסחא שהוגדרה נקבל מטריצה אלכסונית שכל ערכיה על האלכסון הם חיוביים כפול משהו חיובי, ולכן כל ע"ע חיוביים ממש, ועל כן היא אכן מוגדרת חיובית. מסקנה: E אליפסואיד. כעת נוכיח את 3 - נציב בנוסחא ונקבל:

$$\begin{aligned} \left(x - \frac{e_1}{n+1}\right)^T D^{-1} \left(x - \frac{e_1}{n+1}\right) &= \sum_{i=1}^n D_{ii}^{-1} \left(x - \frac{e_1}{n+1}\right)_i^2 \\ &= \left(\frac{n+1}{n}\right)^2 \cdot \left(x_1 - \frac{1}{n+1}\right)^2 + \frac{n^2-1}{n^2} \sum_{i=2}^n x_i^2 \\ &= \frac{n^2-1}{n^2} \sum_{i=2}^n x_i^2 + \frac{1}{n^2} - \frac{2(n+1)}{n^2} x_1 + \frac{2(n+1)}{n^2} x_1^2 \end{aligned}$$

נשים לב כי מכיוון והנקודה x היא בכדור היחידה, $\sum_{i=1}^n x_i^2 \leq 1$. על כן בסה"כ נקבל כי הביטוי בגודל המתבקש. נותר אם כך להוכיח את תכונה 2.

$$\frac{\text{vol}\left(E\left(D, \frac{1}{n+1}e_1\right)\right)}{\text{vol}(B_n)} = \sqrt{\det D}$$

מדוע? זוהי תכונה של טרנספורמציה אפינית. מכאן נקבל:

$$= \sqrt{\det \left(\frac{n^2}{n^2-1} \left(I - \frac{2}{n+1} e_1 e_1^T \right) \right)} = \frac{n}{n+1} \left(\frac{n^2}{n^2-1} \right)^{\frac{n-1}{2}}$$

נעריך את הביטוי באופן הבא:

$$\begin{aligned} 1 - \frac{1}{n+1} &\leq e^{-\frac{1}{n+1}} \\ \left(1 + \frac{1}{n^2-1}\right)^{\frac{n-1}{2}} &= \left(\left(1 + \frac{1}{n^2-1}\right)^{(n^2-1)} \right)^{\frac{n-1}{2(n^2-1)}} \leq e^{-\frac{1}{2(n+1)}} \end{aligned}$$

ולכן בסה"כ הביטוי שקיבלנו מסתכם ל:

$$\leq e^{-\frac{1}{n+1}} \cdot e^{-\frac{1}{2(n+1)}} = e^{-\frac{1}{2(n+1)}}$$

■

הערה 1.16 לא היה בהוכחה משהו מתוחכם, רק חישובים. התחכום בא מהמעבר לחצי האליפסואיד.

לסיכום, נגדיר פורמלית העתקה אפינית, כדי שנוכל להצדיק את השימוש בה:

הגדרה 1.17 תהי $A \in \mathbb{R}^{n \times n}$ מטריצה $n \times n$ לא סינגולרית, $b \in \mathbb{R}^n$. טרנספורמציה אפינית היא:

$$\begin{aligned} T_{A,b} : \mathbb{R}^n &\rightarrow \mathbb{R}^n \\ T_{A,b}(x) &= Ax + b \end{aligned}$$

נצטט מספר תכונות של טרנספורמציה שכזו:

- אם $X \subseteq X'$, אז $T_{A,b}(X) \subseteq T_{A,b}(X')$.
- אם X ממימד מלא וקמורה, אזי:

$$\text{vol}(T_{A,b}(X)) = |\det(A)| \text{vol}(X)$$

נקבל כי אם $E = E(D, y)$, $D = RR^T$ אזי:

$$E = T_{R,y}(B_n)$$

אזי כדי לבצע פעולות על האליפסואיד נוכל להעביר לכדור היחידה, ולחשב ולחזור לאליפסואיד, ולהשאר עם אותו הנפח.

1.4.4 הכח של האלגוריתם

מה נדרש כדי להפעיל את אלגוריתם האליפסואיד?

1. חסמים U ו- L על כדור חוסם וכדור חסום, ואת הכדור החוסם צריך במפורש (כדי להתחיל ממנו).
 2. בהנתן נקודה $x \in \mathbb{R}^n$, צריך אלגוריתם שבודק אם $x \in P_{<}$, ואם $x \notin P_{<}$, צריך להחזיר אילוץ לינארי שמתקיים ב- x אבל לא מתקיים באף נקודה של $P_{<}$ (בפרט, $P_{<}$ חייבת להיות קבוצה קמורה).
- כך אין צורך בכלל לדעת מהי התוכנית הלינארית - פרוצדורה שכזו נקראת אוב הפרדה (separation oracle). זוהי הסיבה המרכזית להיות הכלי הזה מאוד מאוד חזק, שכן הוא לא דורש הרבה.

2 חיתוך מטרואידים

2.1 הגדרות

12/11/2012

יושלים בהמשך...

הגדרה 2.1 מטרואיד הוא זוג (E, I) , כאשר E הוא קבוצה סופית, ו- I הוא אוסף לא ריק של תתי קבוצות של E המקיים:

1. (תורשיות) אם $A \in I$ ו- $B \subseteq A$, אזי $B \in I$.

2. (החלפה) אם $A, B \in I$, $|A| < |B|$, אזי קיים איבר $b \in B \setminus A$ כך ש- $A \cup \{b\} \in I$.

19/11/2012

נתונים שני מטרואידים:

$$M_1 = (E, I_1), M_2 = (E, I_2)$$

ומחפשים $S \in I_1 \cap I_2$, שאינה תלוי בשתייהן, עם $|S|$ מקסימלי.
לבעיה הזו יש גם גרסה ממשוקלת - לא נראה לה אלגוריתם, אך נראה אלגוריתם לבעיה ספציפית בגרסה הממושקלת.
אילו בעיות ניתן לפתור בעזרת חיתוך מטרואידים?

2.2 דוגמאות לבעיות שניתן לפתור בעזרת חיתוך מטרואידים

2.2.1 זיווג מקסימום בגרף דו צדדי

כל אחד מהצדדים מגדיר מטרואיד, קבוצת הבסיס היא קב' הקשתות.

2.2.2 עץ פורש מכוון עם שורש r

בגרף מכוון בוחרים צומת ספציפי r , ומחפשים עץ פורש מכוון מהשורש r . אפשר להוריד את כל הצלעות הנכנסות לשורש, ולכן אפשר להניח שדרגת הכניסה שלו היא 0.
המטרואיד הראשון יהיה המטרואיד הגרפי, כאשר מתעלמים מכיווני הקשתות. המטרואיד השני יהיה מטרואיד חלוקה, כשנדרוש של- r לא יכנסו קשתות, ולשאר מקסימום אחד. החיתוך של שני המטרואידים הללו יתן את התוצאה הדרושה (בתנאי שהמטרואיד השני מספיק גדול).

2.2.3 כיוון קשתות

נתון גרף לא מכוון $G = (V, E)$ ופונקציה $k : V \rightarrow \mathbb{N}$, והשאלה היא: האם ניתן לכוון את כל הקשתות ב- G , כך שדרגות הכניסה של הצמתים יהיו לכל היותר k (כלומר, עבור כל $v \in V$, דרגת הכניסה של v היא לכל היותר $k(v)$)?

נבנה את המטרואידים - נחליף כל קשת $\{u, v\}$ בזוג קשתות מכוונות (v, u) , (u, v) .
המטרואיד הראשון יחלק לפי קשתות - כל החלקים הם בגודל 2, ואפשר לקחת לכל היותר 1.
המטרואיד השני יחלק לפי צמדתם - עבור צומת v נדרוש מהחלק המתאים לו כי ילקחו לכל היותר $k(v)$ איברים.
נפתור את בעית חיתוך המטרואידים - אם הגודל הוא בדיוק מספר הקשתות בגרף המקורי אז יש תשובה, אחרת אין.

2.2.4 עץ פורש צבעוני

נתון גרף לא מכוון $G = (V, E)$, ולכל קשת יש צבע:

$$c : E \rightarrow \{1, \dots, k\}$$

רוצים למצוא עץ פורש, בו לכל קשת יש צבע אחר.
המטרואיד הראשון יהיה המטרואיד הגרפי.
המטרואיד השני יהיה החלוקה, בו מכל צבע ניתן לקחת איבר אחד.
אם $|S|$ בדיוק מספר הקשתות, אזי יש פתרון, אחר אין.

יש עוד הרבה בעיות מהסוג הזה.

2.3 שימוש בחיתוך מטרואידים לפתרון בעיות - דיון כללי

נתבונן בקבוצה $S \in I_1 \cap I_2$ (לא דווקא זו שאנו מחפשים, אלא S כלשהוא), ותהא $U \subseteq E$. אפשר לרשום:

$$S = (S \cap U) \cup (S \cap (E \setminus U))$$

מכיוון וזהו איחוד זר:

$$|S| = |S \cap U| + |S \cap (E \setminus U)| \leq \rho_{M_1}(U) + \rho_{M_2}(E \setminus U)$$

כאשר ρ_{M_i} הוא חסם כללי - קטן מהחסם בשני המטרואידים. בפרט, אפשר לקחת מקסימום בצד שמאל ומינימום מימין, ולקבל:

$$\max_{S \in I_1 \cap I_2} |S| \leq \min_{U \subseteq E} \{\rho_{M_1}(U) + \rho_{M_2}(E \setminus U)\}$$

אי השיוויון הזה הוא באופן כללי אי שיוויון של דואליות: למקסימום של דבר אחד יש חסם ע"י המינימום של ביטוי אחר. עם זאת, בדואליות אנו מצפים לשיוויון, ואכן, המשפט המרכזי שנוכיח הוא באמת השיוויון.

משפט 2.2 לכל שני מטרואידים כנ"ל מתקיים שיוויון:

$$\max_{S \in I_1 \cap I_2} |S| = \min_{U \subseteq E} \{\rho_{M_1}(U) + \rho_{M_2}(E \setminus U)\}$$

איך נוכיח?

נראה אלגוריתם שיבנה קבוצה S ובמקביל קבוצה U . עבור הקבוצות הספציפיות הללו, נראה שמתקיים השיוויון. אזי, בפרט הקבוצות שבנינו הן אופטימום - דואליים, לכן באיזהו מובן, האלגוריתם יבנה קבוצות דואליות.

הרעיון הכללי יהיה מאוד דומה למציאת זיווג מקסימום בגרף דו צדדי: מה שאנחנו ממספרים .. בחתך שמופיע לבעיית המינימום (אם נוסף מקור ובור ונהפוך את הגרף המקורי לרשת זרימה).

בגרף של זיווג מלא - U יהיה קבוצה ריקה, $E \setminus U = E$.

כדי לעבור ממטרואידים לגרפים, אנו זקוקים למושג הבא:

הגדרה 2.3 נתון מטרואיד $M = (E, I)$ ו- $S \in I$. גרף ההחלפה של המטרואיד, שיסומן ב- $G_M(S)$, הינו גרף דו צדדי עם הצמתים $S, E \setminus S$, כאשר בהנתן $x \in E \setminus S, y \in S$ יש קשת (y, x) אם $\{y\} \cup \{x\} \in I$.

הערה 2.4 כלומר, אם לאחר ההחלפה של שני הקודקודים, הקבוצה עדיין ב- I , כלומר הגדף יכול קשתות בין כל האיברים שניתן להחליף ביניהם.

כעת, שתי למות ללא הוכחה (אולי ינתנו בתור תרגיל בהמשך):

למה 2.5 תהייה $S, T \in I$ (בת"ל באיזהו מטרואיד $M = (E, I)$), עבורן $|S| = |T|$. אזי, ב- $G_M(S)$ יש זיווג מושלם בין $S \setminus T$ לבין $T \setminus S$.

הערה 2.6 בפרט זה אומר שניתן להחליף את כל איברי $S \setminus T$ באיברי $T \setminus S$, ועדיין תיוותר קבוצה בת"ל.

הטענה ההפוכה לא נכונה בהכרח. אולם:

למה 2.7 תהי $S \in I$ במטרואיד $M = (E, I)$, ותהי $T \subseteq E$ עבורה $|T| = |S|$ אם ב- $G_M(S)$ יש זיווג מושלם יחיד בין $S \setminus T$ לבין $T \setminus S$, אזי $T \in I$.

הגדרה 2.8 יהיו $M_1 = (E, I_1), M_2 = (E, I_2)$. גרף ההחלפה $G_{M_1, M_2}(S)$ הוא גרף דו צדדי מכוון, כאשר הצדדים הם S ו- $E \setminus S$, והצלעות מוגדרות באופן הבא:

1. יש קשת קשת מ- y ל- x אם $\{y\} \cup \{x\} \in I_2$.

2. יש קשת מ- x ל- y אם $\{y\} \cup \{x\} \in I_1$.

.....

26/11/2012

2.4 מציאת עץ פורש מינימום מכון

נושא זה יקשר בין מטרואידים לבין הפרק החש - הסכמה הפרימלית דואלית. נתון גרף מכון $G = (V, E)$, וישנה פונקציית מחיר על קשתות מכוונות:

$$c : A \rightarrow \mathbb{N}$$

וכן נתון צומת ספציפי $r \in V$ שהוא שורש העץ.

המטרה: למצוא עץ מכון עם שורש r שפורש את G ומחירו מינימלי.

כמובן שלא בהכרח יש עץ כזה - במקרה שאין האלגוריתם יתקע וידוע שלא קיים כזה (אפשר להניח שאין קשתות שנכנסות ל- r , אחרת פשוט נוציא אותו, כי ברור שהן לא יהיו חלק מהעץ הפורש).

כבר דיברנו על הבעיה הזו - זו בעיה מוכללת של חיתוך מטרואידים עם משקלים:

המטרואיד $M_1 = (A, I_1)$ הוא המטרואיד הגרפי של G , כאשר מתייחסים אליו כגרף לא מכון (אוסף כל הקבוצות בהן אין מעגל).

המטרואיד השני $M_2 = (A, I_2)$ הוא מטרואיד החלוקה, שבו מרשים לכל צומת $v \in V \setminus \{r\}$ לכל היותר קשת אחת נכנסת.

כל עץ פורש מכון עם שורש r הוא קבוצה בלתי תלויה גם ב- M_1 וגם ב- M_2 , שגודלה $|V| - 1$. אזי, אם קיים חיתוך יש פתרון, אחרת אין. מבין כל הקבוצות המקסימליות, אנו רוצים את זו בעלת המשקל המינימלי.

לא נראה את האלגוריתם הממושקל של חיתוך מטרואידים, אבל יש כזה.

כן נראה אלגוריתם שפותר את הבעיה הספציפית הזו - הבה נתחיל לפתח אותו:

השלב הראשון בפיתוח האלגוריתם הוא לשים לב שניתן לנסח את הפתרון לבעיה בצורה אחרת - בתור פתרון בעיית תכנון בשלמים:

נגדיר לכל קשת מכוונות $a \in A$ משתנה $x_a \in \{0, 1\}$ המציין אם הקשת מופיעה בפתרון או לא. אזי, ניתן לנסח את הפתרון כך:

$$\begin{aligned} \min \quad & \sum_{a \in A} c(a) \cdot x_a \\ \text{s.t.} \quad & \sum_{a \in A; a \text{ enters } S} x_a \geq 1 \quad \forall S \subseteq V \setminus \{r\} \\ & x_a \in \{0, 1\} \quad \forall a \in A \end{aligned}$$

מספיק לדרוש כי הוא יכול עץ - כי המינימום יבטיח שלא נקבל יותר מעץ.

יש כאן מספר גדול מאוד של אילוצים, וכן היא בשלמים - לכן הניסוח הזה לא נותן לנו בעיה הניתן לפתרון בזמן פולינומי בהכרח.

כל בעיית קומבינציה קומבינטורית ניתן לנסח באופן שכזה, ונראה זאת בהמשך.

עד כאן - לקחנו בעיה "קלה" והפכנו אותה ל"קשה" שאנחנו לא יודעים לפתור. הבעיה שעושה אותה קשה היא האילוצים מהסוג השני. לכן, נבצע רילקסציה:

$$\begin{aligned} \min \quad & \sum_{a \in A} c(a) \cdot x_a \\ \text{s.t.} \quad & \sum_{a \in A; a \text{ enters } S} x_a \geq 1 \quad \forall S \subseteq V \setminus \{r\} \\ & x_a \geq 0 \quad \forall a \in A \end{aligned}$$

אין צורך לדרוש $x_a \in [0, 1]$, כי זוהי בעיית מינימיזציה - כל אחד מ- x_a הגדולים מ-1 ניתן להקטין מבלי לפגוע באיכות הפתרון, ועל כן יתקיים מבלי שנדרוש אותו במפורש.

זוהי תוכנית לינארית - אומנם עם מספר אקספוננציאלי של אילוצים, אבל עדיין...

במקרה הזה, ניתן להתגבר על מספר האילוצים. מדוע?

נזכור את מה שאמרנו על אלגוריתם האליפסואיד - ניתן להשתמש בו תחת תנאים מאוד קלים, וכן צריך לדאוג שיהיה או הפרדה. לכן לא צריך לעבור על האילוצים באופן מפורש, רק לבדוק אם הוקטור מקיים את התנאים, ואם לא - להצביע על האילוץ המפר אותם.

לקבלת חתך המינימום המפריד בין r לבין S - מפעילים אלגוריתם זרימה, ואם הקיבול שניתן לקשת a - x_a , לפחות 1 אין שום אילוץ שמופר, אם קטן מ-1 יש הצבעה על אילוץ קונקרטי שהופר. לכן, את התוכנית הזו ניתן לפתור בעזרת אלגוריתם האליפסואיד.

אבל, לא נשתמש בו - נרצה אלגוריתם הרבה יותר יעיל.

במעבר בין שתי התוכניות - תחום האופטימיזציה גדל, לכן אם נסמן את המקורית ב- Z_{int} ואת החדשה ב- Z_{frac} , מתקיים:

$$Z_{frac} \leq Z_{int}$$

ראינו את זה במקרה של אופטימיזציה במטרואידים. לכן, אם נמצא פתרון לתוכנית השנייה, נקבל חסם תחתון לבעיה המקורית. נסתכל על התוכנית הדואלית לתוכנית השנייה:

$$\begin{aligned} W_{frac} = \max \quad & \sum_{S \subseteq V \setminus \{r\}} y_S \\ \text{s.t.} \quad & \sum_{S: a \text{ enters } S} y_S \leq c(a) \quad \forall a \in A \\ & y_S \geq 0 \quad \forall S \subseteq V \setminus \{r\} \end{aligned}$$

ממשפט הדואליות החזק, אנו יודעים כי:

$$W_{frac} = Z_{frac} (\leq Z_{int})$$

לכל פתרון פרימלי x ודואלי y , מתקיים:

$$\sum_S y_S \leq \sum_a c(a) x_a$$

אנחנו נבנה פתרון פרימלי שלם x , ופתרון דואלי כלשהוא y , ונראה שעבורם מתקיים:

$$Z_{int} \geq Z_{frac} \left(\geq \sum_S y_S = \sum_a c(a) x_a \right) \geq Z_{int}$$

לכן כל אי השוויונים חייבים להיות שוויונים, ולכן הפתרון הוא אופטימלי גם לשלמים וגם לתוכנית הלינארית. האלגוריתם יהיה פולינומי. זהו מצב די נדיר, איך נעשה זאת? נזכר בתנאי העודפיות המשלימה:

$$\begin{aligned} I) x_a > 0 & \Leftrightarrow \sum_{a \in A; a \text{ enters } S} y_S = c(a) \\ II) y_S < c(a) & \Leftrightarrow \sum_{a \in A; a \text{ enters } S} x_a = 1 \end{aligned}$$

ראינו שבמקרה זה הפתרונות אופטימליים.

די ברור שיותר קל להתחיל מהדואלי מאשר מהפרימלי - פתרון פיזיבלי פשוט הוא הפתרון שנתחיל איתו - $y_S = 0$ לכל S .

האלגוריתם ישמר מספר דברים (יבנה פתרון דואלי, ועל סמך זה תוך כדי יפתור את הפרימלי):

1. הדואלי יהיה כל הזמן פיזיבלי (בסוף גם הפתרון הפרימלי) - יהיו מעט מאוד S -אים עבורם $y_S \neq 0$, ואז זמן הריצה אכן יהיה פולינומי.

2. תנאי העודפיות המשלימה הראשון.

עבור כל a , $x_a = 0$ בהתחלה. נגדיר y_S וברגע ש- $\sum y_S = c(a)$, נהפוך את x_a המתאים לו ל-1 (לא נהפוך אותו אם זה לא הדוק).

האלגוריתם יפעל באופן הבא:

• **פאזה ראשונה:**

- בכל שלב נבחר S המקיימת את התכונה הבאה:
 S הוא רכיב קשיר בגרף המושרה ע"י הקבוצה $\{a \in A | x_a = 1\}$. כלומר, בגרף זה אפשר להגיע מכל צומת בקבוצה S לכל צומת אחרת ב- S . בנוסף, נדרוש כי $r \notin S$, ואין קשת הנכנסת ל- S .
 מהי הקבוצה? אוסף כל הקשתות שכבר בחרנו.

- נגדיל את y_S עד שאיזשהו אילוץ דואלי נעשה הדוק, כלומר $\sum y_S = c(a)$.
- נשנה את x_a להיות 1. אזי יש קשת שנכנסת ל- S , ולכן היא לא תבחר בפעם הבאה.

• פאזה שניה

- באיזשהו שלב נגמרות ה- S -אים העומדות בתנאי, ואז האלגוריתם נתקע - בשלב זה עוצרים.
- נראה כי בשלב זה יש מ- r מסלול לכל צומת בגרף, אלב זה לא בהכרח עץ. לכן בשלב זה עוברים על כל הקשתות שכבר לקחנו בסדר הפוך, ומורידים כל קשת מיותרת במובן שמשאיר את התנאי שמ- r יש מסלול לכל צומת בגרף.

הטענה כי כי בסיום הריצה, התוצאה תקיים את תנאי העודפיות השניה. מכיוון והתנאי הראשון נשמר לאורך כל הריצה, הפתרון הוא אופטימלי. כמו כן יש להוכיח כי בסיום פאזה 1 באמת אפשר להגיע מ- r לכל צומת בגרף. תנאי 1 תמיד מתקיים על סמך פעולת האלגוריתם בפאזה הראשונה. תנאי זה לא מופר בפאזה השניה, כי רק מורידים קשתות, ולכן הוא עדיין מתקיים.

טענה 2.9 בסוף הפאזה הראשונה ניתן להגיע מ- r לכל קודקוד.

הוכחה: נתבונן במצב תוך כדי הפאזה הראשונה של האלגוריתם - נתון גרף שהקשתות שלו הן $\{a \in A | x_a = 1\}$. נכוץ כל רכיב קשיר חזק לקודקוד: *איור 2* הגרף הוא א-ציקלי (כמובן שהרכיבים שלו מכילים את r). כל עוד אי אפשר להגיע מ- r לכל הצמתים, חייב להיות רכיב שכזה שלא נכנסת אליו קשת, לכן כל עוד לא נא ניתן להגיע מ- r לכל הצמתים, פאזה 1 נמשכת, ולכן בסיום פאזה זה ניתן להגיע מ- r לכל צומת אחרת. ■

טענה 2.10 בסוף הפאזה השניה מתקבל עץ פורש מכוון עם שורש r .

הוכחה: בוודאי אפשר להגיע מ- r לכל צומת - שכן מחקנו קשת רק אם היא לא פגעה בתכונה זו. נותר להראות שלכל צומת שאיננה r נכנסת בדיוק קשת אחת. נניח בשלילה שיש צומת v אליה נכנסות שתי קשתות. נניח (u_1, v) נבחרה לפני (u_2, v) . אזי, כשנגיע בפאזה השניה ל- (u_2, v) נמחק אותה, שכן אפשר להגיע ל- v דרך (u_1, v) , וזו בסתירה להנחה ששתי הקשתות נותרו בפתרון. ■

טענה 2.11 אם T היא קבוצת הקשתות בפתרון בסוף הפאזה השניה, אזי לכל S עברה $y_S > 0$, מתקיים שיש בדיוק קשת אחת ב- T הנכנסת ל- S .

הוכחה: נניח בשלילה שיש S כזו שאליה נכנסות שתי קשתות ב- T . נסמן ב- a_1, a_2 היתה פעילה באלגוריתם בשלב מסוים, כי $y_S > 0$. בשלב זה בחרנו כבר קשתות שקישרו רכיב קשיר חזק על S , ו- a_1, a_2 עדיין לא נבחרו. אם a_2 הקשת המאוחרת יותר שנכנסה ל- S , כאשר נגיע אליה בפאזה השניה נמחק אותה, כי S עדיין קשירה חזק. לכן יש בדיוק קשת אחת לתוך S , וחייבת להיות אחד כזו, אחרת לא היה אפשר להגיע מ- r לכל קודקוד ב- S . ■

גישה זו נקראת **הסכמה הפרימלית דואלית** - בנקודה זו נסיים עם מטרואידים ונעבור לדון בה:

3 הסכמה הפרימלית דואלית

נלמד את הסכמה הזו דרך מספר מקרים פרטיים, כמו זה שראינו מעלה.

3.1 עץ שטיינר

נתון גרף לא מכוון $G = (V, E)$, פונקציית מחיר על הקשתות:

$$c: E \rightarrow \mathbb{N}$$

וקבוצה $V \supset T$ של טרמינלים.

רוצים לבחור קבוצה של קשתות במחיר כולל מינימלי המקשרת את כל הטרמינלים ב- T (בפרט הקבוצה הזו תשרה תת גרף שהוא עץ) - יתכן ועוד טרמינלים יהיו מקושרים - אלו יקראו נקודות שטיינר. קל לראות שהעלים חייבים להיות טרמינלים - אחרת היינו יכולים לוותר על הקשתות (הפוך - לא).

בניגוד לבעיית הקודמת, זוהי בעיה NP -קשה. אפשר לתת לה קירובים, אפילו די טובים. השיא העולמי כרגע לגרסא זו של הבעיה הוא 1.55 (לא נראה אותו). נראה קירוב בפקטור 2 (אפשר גם לחפש עץ מינימום לטרמינלים - גם פתרון שכזה יתן קירוב 2).

נקודת ההתחלה: נאפיין את התוכנית בשלמים:

$$\begin{aligned} Z_{ST} = \text{Min} \quad & \sum_{e \in T} c(e) \cdot x_e \\ \text{s.t.} \quad & \sum_{e \in E, |e \cap S|=1} x_e \geq 1 \quad \forall S \subset V \quad S \cap T \neq \emptyset, (V \setminus S) \cap T \neq \emptyset \\ & x_e \in \{0, 1\} \quad \forall e \in E \end{aligned}$$

התוכנית הזו בדיוק מאפיינת את בעיית עץ שטיינר. כאמור, את זה אנחנו לא יודעים לפתור ולכן **מבצעים רלקסציה**. נקבל:

$$\begin{aligned} Z_{LP} = \text{Min} \quad & \sum_{e \in T} c(e) \cdot x_e \\ \text{s.t.} \quad & \sum_{e \in E, |e \cap S|=1} x_e \geq 1 \quad \forall S \subset V \quad S \cap T \neq \emptyset, (V \setminus S) \cap T \neq \emptyset \\ & x_e \geq 0 \quad \forall e \in E \end{aligned}$$

כאן $Z_{LP} \leq Z_{ST}$ - כאן לא יהיה שיוויון שכן כאמור הבעיה המקורית NP -קשה. **התוכנית הדואלית** ל- Z_{LP} תהיה:

$$\begin{aligned} W_{LP} = \text{Min} \quad & \sum_{e \in T} y_S \quad \forall S \subset V \quad S \cap T \neq \emptyset, (V \setminus S) \cap T \neq \emptyset \\ \text{s.t.} \quad & \sum_{e \in E, |e \cap S|=1} y_S \leq c(e) \quad \forall e \in E \\ & y_S \geq 0 \quad \forall S \subset V \quad S \cap T \neq \emptyset, (V \setminus S) \cap T \neq \emptyset \end{aligned}$$

מדואלית:

$$W_{LP} = Z_{LP} \leq Z_{ST}$$

פאזה 1:

נתחיל עם פתרון דואלי פיזיבילי כלשהוא בו כל המשתנים הם 0, ונתחיל עם פתרון פרימלי **לא** פיזיבילי שגם בו הכל 0, ונעלה משתנים דואליים. בניגוד למקודם, נגדיל כמה משתנים דואליים במקביל (לא אחד אחד). אילוץ מתאים לקשת יכול להיות הדוק משתי סיבות:

1. צד אחד של הקשת היה ב- S שהגדלנו, והקצה השני לא היה באף רכיב קשיר (לא מחובר לשום טרמינל, ואינו טרמינל בעצמו). במקרה זה נוסיף את הקשת לפתרון, ונגדיל את S כך כך שהקשת תוכל בה.

2. המקרה המסובך: קשת אשר שני הצדדים שלה נמצאים בשני רכיבים שונים, ומגדילים את שניהם במקביל. אזי הקשת נעשית הדוקה משני הכיוונים. במקרה הזה מוסיפים את הקשת לפתרון, ומחליפים את שני הרכיבים ברכיב אחד.

כאשר מתחילים הרכיבים הם הטרמינלים, וכאשר ניוותר עם רכיב בודד - מסיימים פאזה זו.

פאזה 2:

עוברים על הקשתות שהוספנו בסדר הפוך, ו"זורקים" קשתות עד שנקבל את הפורש את טרמינלים.

נראה x פתרון פיזיבילי לבעיית עץ שטיינר ופתרון דואלי שברי (לא בהכרח שלם) שאינם שווים, כך שהפרימלי לא יקר ביותר מפי 2 מהדואלי, ולכן לא יותר מפי 2 מהפתרון האופטימלי. יושלם בהמשך...

24/12/2012

3.2 בעיית $K - MST$

3.2.1 כופלי לגראנג'

31/12/2012

נסתכל על תוכנית לינארית:

$$\begin{aligned} \min c^T x \\ \text{s.t. } Ax \geq b \\ x \geq 0 \\ x, x \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \end{aligned}$$

זוהי לא הצגה קנונית או סטנדרטית, אבל נשתמש בה לצרכים אלה. ניתן לעשות בה מה שעושים גם בהצגות אחרות.

הרעיון עם כופלי לגראנג' באופן כללי - להפטר מאילוץ ע"י שינוי פונקציית המטרה. כלומר, נחליף את האילוץ באיזושהי תוספת לפונקציית המטרה, ונראה כי האופטימום של הפונקציה החדשה יקיים את מה שרצינו מהתוכנית המקורית.

לשם כך, נגדיר וקטור של כופלי לגראנג' $\lambda \in \mathbb{R}^m$. הרעיון: הוא יכפיל את ההפרש בין צד שמאל לצד ימין באילוץ. כלומר נרצה להסתכל על הפונקציה:

$$\min c^T x + \dots$$

כאשר נוסיף איזושהי תיקון שיהיה בעייתי רק כאשר האילוץ הזה מופר. התיקון שנרצה²:

$$\begin{aligned} \min c^T x + \lambda^T (Ax - b) = f(x, \lambda) \\ \text{s.t. } x, \lambda \geq 0 \end{aligned}$$

זאת בעיית אופטימיזציה שהיא כמעט לא מאולצת - יש אילוץ סימן, אך דבר מעבר (אם היה שווה במקום גדול שווה בפונקציה המקורית היינו מקבלים משהו שכלל אינו מאולץ). אנו עובדים עם פונקציות לינאריות שמקיימות את כל התכונות שמעניינות אותנו, ולכן אין לנו בעיה עם וקטור הכופלים. נשים לב כי במקרה זה, אם האילוץ המקורי מופר אז מקבלים תוספת חיובית ל- $f(x, \lambda)$, וכאשר אנו מחפשים מינימום זה "קנס".

המינימום שאנו מחפשים תהיה נקודת קיצון, מקומית בפרט (במקרה הזה כל נקודת קיצון מקומית תהיה גם גלובלית אך לא נדון בכך).

אפשר להסתכל על הנגזרות החלקיות - במקרה של המינימום היא תצטרך להתאפס. אילוץ הסימן אומרים - אם איזה משתנה שאנו בודקים ערכו אפס, אין זה אומר שהנגזרת שם אפס. אולם אנו יודעים את הסימן של הנגזרת - או אפס או גדולה או שווה לאפס (כי אנו מנסים להקטין הרי).

נניח (x^*, λ^*) פתרון אופטימלי. אזי:

$$\bullet \text{ אם } x_i^* > 0 \text{ אזי } \left. \frac{\partial f}{\partial x_i} \right|_{x^*, \lambda^*} = 0$$

$$\bullet \text{ אם } x_i^* = 0 \text{ אזי } \left. \frac{\partial f}{\partial x_i} \right|_{x^*, \lambda^*} \geq 0$$

$$\bullet \text{ אם } \lambda_j^* > 0 \text{ אזי } \left. \frac{\partial f}{\partial \lambda_j} \right|_{x^*, \lambda^*} = 0$$

$$\bullet \text{ אם } x_i^* = 0 \text{ אזי } \left. \frac{\partial f}{\partial \lambda_j} \right|_{x^*, \lambda^*} \geq 0$$

תנאים אלו נקראים מקדמי kkt (עקב משפט בנושא זה).

פונקציית המטרה החדשה היא לינארית ב- x , λ^T הוא וקטור שורה באורך n . הקואורדינטה ה- i שלה אם כך תהיה:

$$\bullet \text{ אם } x_i^* > 0 \text{ אזי } c_i - \lambda^T A_{\cdot i} = 0$$

$$\bullet \text{ אם } x_i^* = 0 \text{ אזי } c_i - \lambda^T A_{\cdot i} \geq 0$$

לכן בשה"כ $c + \lambda^T A \geq 0$, כלומר $\lambda^T A \geq -c^T$.

²למעשה אנו מקבלים כאן את הלגראנג'יאן! הידד!

$$\bullet \text{ אם } \lambda_j^* > 0 \text{ אזי } \left. \frac{\partial f}{\partial \lambda_j} \right|_{x^*, \lambda^*} = 0$$

$$\bullet \text{ אם } x_i^* = 0 \text{ אזי } \left. \frac{\partial f}{\partial \lambda_j} \right|_{x^*, \lambda^*} \geq 0$$

לכן הנגזרת של f לפי x_i תהיה:

$$b_j - A_j \cdot x \geq 0 \Rightarrow Ax \geq b$$

מה המסקנה עד עכשיו?

אם מוצאים את האופטימום ל- $f(x, \lambda)$, בפרט הוא יקיים את האילוץ המקורי $Ax \geq b$, ונטען כי התוספת עצמה תהיה 0.

בנקודה שבה $\lambda_j^* > 0$, ההפרש $b_j - A_j \cdot x$ שווה לאפס ולכן התוספת אפס. בנקודה בה הוא שווה לאפס, הכפל שווה לאפס, כלומר התוספת היא אפס. על כן אופטימיזציה לבעיה השניה יוצרת לנו פתרון אופטימלי לבעיה הראשונה. ננסה להבין מי הם כופלי לגראנז' בהם השתמשנו במונחים "שאנחנו כבר יודעים" - מהו λ האופטימלי? בדיוק הערך של הפתרון הדואלי אופטימלי! זו בעצם דרך אחרת להבנת מהו הדואלי - הדואלי מחשב את כופלי לגראנז' האופטימליים:

$$\begin{aligned} \max \lambda^T b \\ \text{s.t. } \lambda^T A \geq -c^T \\ \lambda \geq 0 \end{aligned}$$

למה הדואלי הוא באמת האופטימלי?

$$\begin{aligned} (\lambda^T A)_i &= \begin{cases} -c_i & x_i^* > 0 \\ m & x_i^* = 0 \end{cases} \\ \Rightarrow \lambda^* A x^* &= -c^T x^* \end{aligned}$$

טוב, יש פה בעיית סימן - אבל אמורה להתקבל כאן בדיוק התוכנית הדואלית... בשלב זה נצטרך להאמין, ולעבור לבעיה הקונקרטית בה נשתמש בה.

3.2.2 הבעיה - $K - MST$

נתון גרף לא מכוון $G = (V, E)$, משקלים על הקשתות $x: E \rightarrow \mathbb{N}$, $k \in \mathbb{N}$, שורש $r \in V$. המטרה: למצוא עץ עם שורש r הפורש לפחות k צמתים ומשקלו מינימלי. בעיית עץ פורש מינימלי היא מקרה פרטי של בעיה זו, אך בעוד הבעיה הראשית היא NP -קשה, תת הבעיה קלה מאוד לפתרון.

הסתכלנו על הבעיה בעזרת תוכנית בשלבים:

$$\begin{aligned} \min \sum_{e \in E} c(e) x_e \\ \text{s.t. } \sum_{e: |e \cap S|=1} x_e + \sum_{T: S \subseteq T} z_T \geq 1 \quad \forall S \subset V, r \notin S \\ \sum_S z_S |S| \leq |V| - k \\ x_e \in \{0, 1\} \quad \forall e \in E \\ z_S \in \{0, 1\} \quad \forall S \subseteq V \end{aligned}$$

כאשר x_e הוא משתנה אינדיקטור שיציין אם הצלע בפתרון או לא, z_S יציין אם צומת בחתך אינה בפתרון (אז המשתנה יקבל 0).

נבצע רלקסציה - שני התנאים האחרונים יוחלפו בתנאי $x, z \geq 0$, והתוכנית החדשה תהווה רלקסציה לבעיית $k - MST$.

הרלקסציה הזו נראית מאוד דומה לבעיית איסוף הפרסים - נתרם את הבעיה הזו לבעיית איסוף פרסים בעזרת כופלי לגראנג'. ניקח את האילוץ שמפריע לנו - האילוצים מהסוג השני (מפריע שכן הכיוון שלו הפוך). נוסיף כופלי לגראנג' λ (נחשב את ערכה) ונקבל:

$$\begin{aligned} \alpha_{LAG} = \min \quad & \sum_{e \in E} c(e) x_e + \lambda \left(\sum_S z_S |S| - (|V| - k) \right) \\ \text{s.t.} \quad & \sum_{e: |e \cap S|=1} x_e + \sum_{T: S \subseteq T} z_T \geq 1 \quad \forall S \subset V, r \notin S \\ & x, z \geq 0 \end{aligned}$$

נטען כי קיבלנו את בעיית איסוף הפרסים, כאשר ה"קנס" הוא בדיוק λ .
הבעיה הדואלית לבעיה הזו (בהנתן λ כלשהיא) הינה:

$$\begin{aligned} \beta_{LAG} = \max \quad & \sum_S y_S + \lambda (|V| - k) \\ \text{s.t.} \quad & \sum_{e: |e \cap S|=1} y_S \leq 1 \quad \forall e \in E \\ & \sum_{T: S \subseteq T} y_T \leq |S| \lambda \quad \forall S \subset V, r \notin S \\ & x, z \geq 0 \\ & y \geq 0 \end{aligned}$$

אזי נקבל:

$$\beta_{LAG} = \alpha_{LAG} \leq k - MST$$

כעת, הבה נחשוב מה יכול לקרות כאן.
הרלקסציה היא בדיוק רלקסציה של בעיית איסוף פרסים³, כאשר לכל הקודקודים יש בדיוק אותו הפרס. ננסה לנחש את λ . מה קורה איתו?

• אם $\lambda = 0$, הגורם שמצורף אליו כלל לא משתתף, ואז אין סיבה לקבל עץ, והפתרון האופטימלי יהיה 0 - לא יפרש שום קודקוד.

• מצד שני, אם λ מאוד מאוד גדול, אז "שווה" לנו שכל ה- z -אים יהיו אפס - ברור שיש λ מספיק גדול שזה נכון בשבילו - לקחת את כל הקשתות ולא לוותר על אף קודקוד. במקרה זה נקבל עץ (אין סיבה לקחת משהו קטן ממנו) שיפרוש את כל ה- z (כולם יהיו 0).

בקשת בין שני הערכים הללו, ככל שנגדיל את λ , מספר הקודקודים יהיה יותר גדול. הערך $\sum_S |S| z_S$ לא גדל באופן רציף, אבל בשלב מסוים מספר הקודקודים שאנו בוחרים יעבור ממשוה קטן מ- k למשוה גדול מ- k . נפעיל את האלגוריתם הפרימלי דואלי על בעיית איסוף הפרסים עם קנסות λ . האלגוריתם מוציא פתרון פרימאלי שלם T , ופתרון דואלי y . מה אנחנו יודעים על הפתרון הזה? ההבטחה שראינו לגבי בעיית איסוף הפרסים:

$$\sum_{e \in T} c(e) + 2\lambda (|V \setminus V(T)|) \leq 2 \sum_S y_S$$

אם "יש לנו מזל", בחרנו λ עבורו $|V(T)| = k$. אזי:

$$\sum_{e \in T} c(e) \leq 2 \left(\sum_S y_S - \lambda (|V| - k) \right) = 2 \cdot \beta_{LAG}$$

ולכן זה 2-קירוב לבעיית $k - MST$.

³הוכח בע"פ ולא הספקתי לעקוב.

כעת, נותר למצוא λ שיתאים לנו. נוכל לפסול ערכי λ אפשריים לפי מספר הצמתים שהפתרון יכול. כך אפשר להמשיך בחיפוש בינארי. נשים לב כי לא חייב שחיפוש זה יתכנס לפתרון בו מספר הצמתים הנפרשים הוא בדיוק k .

במקרה כללי, נמצא λ_1, λ_2 עבורם:

$$\lambda_2 - \lambda_1 \leq \frac{C_{min}}{2 \cdot |V| (2|V| + 1)}$$

כלומר הם מאוד קרובים זה לזה (אי אפשר להגיע לקירבה אינסופית, שכן זה עלול לקחת לנו אינסוף איטרציות - גודל שכזה גם מספיק קטן לצרכינו, כבר נראה זאת, וגם מספר האיטרציות הוא לוגריתמי בערך באורך הקלט חלקי הייצוג הנדרש וכו', ובכל אופן פולינומי בגודל הבעיה), והאלגוריתם נותן עצים T_1, T_2 שגודלם $k_1 > k > k_2$. נקבע μ_1, μ_2 עבורם $\mu_1 + \mu_2 = 1$, $\mu_1 k_1 + \mu_2 k_2 = k$ (מעין קומבינציה קמורה של שני העצים). נקבל:

$$\begin{aligned} \mu_1 \sum_{e \in T_1} c(e) + \mu_2 \sum_{e \in T_2} c(e) &\leq 2\mu_1 \left(\sum_S y_S^1 - \lambda_1 (|V| - k_1) \right) + 2\mu_2 \left(\sum_S y_S^2 - \lambda_2 (|V| - k_2) \right) \\ &= 2 \left(\mu_1 \sum_S y_S^1 + \mu_2 \sum_S y_S^2 \right) - 2\lambda_2 (|V| - k) + 2\mu_1 (\lambda_2 - \lambda_1) (|V| - k_1) \\ &\leq 2(k - MST) + \varepsilon(k - MST) \end{aligned}$$

כאשר הביטוי הימני והאמצעי הוא קירוב 2 לבעיית $k - MST$, והשאר הוא משהו מאוד קטן. אם $\mu_1 \geq \frac{1}{2}$, אזי:

$$\sum_{e \in T_1} c(e) \leq 2 \left(\mu_1 \sum_{e \in T_1} c(e) + \mu_2 \sum_{e \in T_2} c(e) \right) \leq 4 \cdot k - MST$$

הבעיה הגדולה נוצרת כאשר המשקל של העץ הקטן יותר היא יותר משמעותית, כלומר $\mu_1 < \frac{1}{2}$, ניקח את T_2 (כאמור הוא עץ קטן מידוי) ונוסיף צמתים:

יש לפחות $3l \geq k_1 - k_2$ צמתים שהם ב- T_1 ולא ב- T_2 . נכפיל את הקשתות של T_1 ונקבל גרף אוילר שיש בו מעגל אוילר. במעגל אוילר נבחר מסלול קל ביותר שפורש $k - k_2$ צמתים שאינם ב- T_2 . המחיר הכולל של מעגל אוילר הוא $2 \cdot \sum_{e \in T_1} c(e)$. נסתכל על קטעים רציפים במעגל: בוודאי שיש קטע שגודלו $\frac{k-k_2}{l} \cdot 2 \sum_{e \in T_1} c(e)$ מתוך סך כל הקטע. גודל זה $\geq \frac{k-k_2}{k_1-k_2} \cdot 2 \sum_{e \in T_1} c(e)$. נשים לב כי $\mu_1 = \frac{k-k_2}{k_1-k_2}$. נוסף קטע שמשקלו $\geq 2\mu_1 \sum_{e \in T_1} c(e) \geq 4 \cdot (k - MST)$. צריך עוד לחבר את הקטע הזה ל- T_2 . נחבר אותו ישירות לשורש. כאן צריך להזהר - החיבור לשורש יכול לעלות המון. **כאן נוסף פרט לאלגוריתם שהוסתר מאיתנו עד כה!** אנחנו צריכים לנחש כאן את הפתרון האופטימלי, ולזרוק את כל הצמתים המרוחקים מהשורש מגודל זה. איך נעשה את זה? חיפוש בינארי: נתחיל ממספר קטן, למשל המשקל המינימלי של צלע, ונבדוק אם קיים פתרון. אם לא - נכפיל בשניים ונמשיך הלאה. זה יקח מקסימום לוגריתמי, וכו'.

בסופו של דבר החיבור יעלה לנו עוד $2 \cdot (k - MST)$.

קיבלנו קירוב בגודל קבוע - אפשר להתקרב כרצוננו ל-2 (ואולי אפילו ממש?) בטכניקות שהן קצת יותר מתוחכמות אבל מתבססות על רעיונות דומים.

כאן אנו מסכמים את הפרק על הפרק הפרימלי דואלי בסכמה זו.

בפרק הבא והאחרון נלמד טכניקה ספציפית למימוש הסכמה הפרימלית דואלית, דרך שימושים שלו (ויש לו רבים) באופן שונה ממה שראינו עד כה - היא עוזרת ליצור פתרון פרימלי שברי.

ננסה בכל זאת להוכיח את כופלי לגראנג'. הפעם נסתכל בהצגה הסטנדרטית של תוכנית לינארית (אפשר להעביר אותה להצגות אחרות ואז הכל בסדר):

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

נוסף את כופלי לגראנג', ונקבל:

$$\begin{aligned} \min \quad & c^T x + \lambda^T (b - Ax) \\ \text{s.t.} \quad & x \geq 0 \end{aligned}$$

נטען כי פתרון אופטימלי בתוכנית זו הוא גם פתרון אופטימלי בתוכנית המקורית, והאילוץ מתקיימים. נביט בנגזרות:

$$\begin{aligned}x_i > 0 &\Rightarrow c_i - \lambda A_{.i} = 0 \\x_i = 0 &\Rightarrow c_i - \lambda A_{.i} \geq 0 \\from\ both &\Rightarrow \lambda^T A \leq c^T \\\lambda_j &\Rightarrow b_j - A_{.j} x = 0 \\\Rightarrow Ax &= b\end{aligned}$$

כלומר בפתרון אופטימלי לתוכנית השניה, האילוץ הזה מתקיים. בפרט אם זה מתקיים, אז לביטוי הזה בפונקציית המטרה של ההצגה השניה אין משמעות, ונקבל את אותו הערך.

3.3 שיטת העדכון הכפלי - Multiplicative Weights Update Method

3.3.1 מוטיבציה

יש לשיטה זו המון המון שימושים. נתחיל מדברים פשוטים, ונראה לאן נגיע. לשיטה זו יש קשר לסכמה הפרימלית דואלית ולכופלי לגראנג'. כופלי לגראנג' הם למעשה המשתנים הדואליים, ומבצעים רלקסציה שקשורה אליהם. השיטה הזו היסטורית צמחה באופן בלתי תלוי במגוון גדול של תחומים - כלכלנים השתמשו בה לראשונה כבר בשנות ה-50, לאחר מכן השתמשו בה כדי לספור זרימה, אנשי החשיבה הלימודית המציאו אותה שוב, ועוד ועוד... יחסית לאחרונה יצאו עבודות המציגות את כל השיטות הזו בדרך כללית. אנחנו נציג את הבעיה דרך דוגמאות.

3.3.2 דוגמא ראשונה - מחשיבה לימודית

נביט במשחק הבא:

יש m חזאים, שבכל יום חוזים אם ירד שלג או לא. אנחנו מחליטים כן שלג\לא שלג על סמך התחזיות (מותר להתעלם מהם לגמרי, מותר לתת יותר חשיבות לחלק ופחות לאחרים וכו'). יש יריב מרושע שמוריד שלג או לא (הוא רואה את התחזיות ואת ההחלטה שלנו). המטרה שלנו: לחזות טוב לפחות כמו החזאי הטוב ביותר. למשל, אם $m = 1$ זה מאוד קל - תמיד נאמר מה שהוא אומר. היריב יכול להכשיל אותנו בכל הימים, אבל החזאי "הכי טוב" גם יפשל בהם, אז עומדים במטרה. המטרה של היריב המרושע: למנוע מאיתנו להשיג את המטרה שלנו.

הערה 3.1 נשים לב כי האסטרטגיה של היריב להכשיל אותנו בכל פעם לא בהכרח תעבוד, שכן יתכן שהוא גם בדרך מכשיל את החזאי הכי טוב...

זוהי נראית מטרה קשה להשגה. נראה כי אפשר למצוא קירובים די טובים.

מקרה פשוט: נניח שיש חזאי אחד מושלם.

אלגוריתם פשוט: נלך לפי דעת הרוב. אם טעינו, "נזרוק" את הטועים.

בכל פעם שזורקים את הטועים, יורדים בפקטור 2. אם התחלנו עם m חזאים, ועל כל טעות זורקים לפחות $\frac{1}{2}$, אזי מספר הטעויות שלנו $\log_2 m \geq$. זהו החסם הטוב ביותר ממנו אפשר להתרחק מהחזאי האופטימלי.

מקרה שני: נניח שהחזאי הטוב ביותר טועה M פעמים.

פתרון: אם נעשה את אותו הדבר, נקבל מספר טעויות $(M + 1) \log_2 m \geq$.

במקרה השני גם החזאי הכי טוב עלול לטעות, כלומר יתכן ואנחנו זורקים חזאים טובים. מה אפשר לעשות במקום?

נסתכל על רב ממושקל של החזאים, ונעניש חזאים שטעו, אבל לא יותר מידי. ה"עונש" הזה הוא בדיוק העדכון הכפלי - נכפיל את המשקל של חזאי שטעה בקבוע קטן מאחד.

שיטת הרב הממושקל

לכל חזאי i נשמור משקל W_i . נסמן ב- W_i^t את המשקל בזמן t .

האתחול: $W_i^1 \leftarrow 1$ לכל i .

התחזית שלנו בזמן t : הרב הממושקל לפי $W_1^t, W_2^t, \dots, W_m^t$.
לכל חזאי i שטעה בזמן t , מעדכנים:

$$W_i^{t+1} \leftarrow \frac{W_i^t}{2}$$

אחרת:

$$W_i^{t+1} \leftarrow W_i^t$$

ניתוח האלגוריתם

נסמן ב- M^t את מספר השגיאות שלנו עד זמן t , M_i^t את מספר השגיאות של החזאי i עד זמן t , וכן נסמן:

$$\phi^t = \sum_{i=1}^m W_i^t, \quad \phi^1 = m$$

מה אנחנו יודעים על ϕ for all i ?

$$\forall i : \phi^{t+1} \geq W_i^{t+1} = \left(\frac{1}{2}\right)^{M_i^t}$$

כדי להשלים את הניתוח, יש להכליל את השגיאה שלנו.
נניח ששגיננו בצעד t . זה אומר שהרב הממושקל של החזאים טעה - כלומר משקל החזאים שטעו הוא לפחות $\frac{1}{2}$ מהמשקל הכולל. לכל חזאי שטעה, משקלו יורד ב-2. לכן המשקל הכולל מפסיד לכל הפחות $\frac{1}{4}$ מערכו:

$$\phi^{t+1} = \sum_{i=1}^m W_i^{t+1} \leq \frac{3}{4} \sum_{i=1}^m W_i^t = \frac{3}{4} \phi^t$$

ואם נסתכל על כל המהלך, נקבל:

$$\begin{aligned} \left(\frac{1}{2}\right)^{M_i^t} &\leq \phi^{t+1} \leq \left(\frac{3}{4}\right)^{M^t} \cdot m \\ \Rightarrow -M_i^t &\leq \left(-\log_2 \frac{4}{3}\right) \cdot M^t + \log_2 m \\ \Rightarrow M^t &\leq \frac{1}{\log_2 \left(\frac{4}{3}\right)} \cdot (M_i^t + \log_2 m) \end{aligned}$$

וזוהי תוצאה הרבה יותר טובה ממה שקיבלנו קודם.

פקטור 2 שבחרנו היה שרירותי, ואכן אפשר לשפר את התוצאה בעזרת פקטור אחר - אם נחלק ב- $1 + \varepsilon$ במקום 2:

$$M^t \leq 2(1 + \varepsilon) \cdot M_i^t + O\left(\frac{1}{\varepsilon} \cdot \log_2 m\right)$$

כלומר, בהתאם לבחירה של ε , אפשר להתקרב לפי 2 טעויות המחזאי, ועוד גורם חיבורי. התוצאה לאו דווקא תהיה הכי טובה עבור אפסילון קטן - זה תלוי במספר הטעויות שעושה החזאי הטוב ביותר.

3.3.3 המשחק המוכלל

כל חזאי בוחר ערך $v_i^t \in [-1, +1]$ (למעשה לא נעשה שימוש בערכים האלה, רק נותן לנו "מוטיביציה" לבחור את הדברים בהמשך).

אנחנו בוחרים התפלגות: $p^t = (p_1^t, p_2^t, \dots, p_m^t)$ על החזאים.
היריב בוחר וקטור מחירים: $c^t = (c_1^t, c_2^t, \dots, c_m^t)$, $c_i^t \in [-1, +1]$.
אנחנו משלמים $\sum_{i=1}^m p_i^t c_i^t$ - כאן יש לנו אפשרות גם להרוויח וגם להפסיד.

החזאי i -י שלם בדיקת הקנסות שלו - $c_i^1, c_i^2, \dots$, ולזה נרצה להשוות את עצמנו.

אסטרטגיית Hedge:

נשייך לכל חזאי משקל W_i^t , $W_i^1 \leftarrow 1$ לכל i . כמו כן:

$$\phi^t = \sum_{i=1}^m W_i^t$$

בצעד ה- i :

$$p_i^t = \frac{W_i^t}{\phi^t}$$

אחרי שנראה את c^t נעדכן:

$$W_i^{t+1} \leftarrow W_i^t \cdot e^{-\varepsilon c_i^t}$$

עבור $0 < \varepsilon \leq 1$ הוא פרמטר. אם c_i^t שלילי, החזאי ה- i "מקבל פרס" - ואכן המשקל שלו יעלה, ואם c_i^t חיובי הוא "נקנס".

משפט 3.2 לכל חזאי i :

$$\sum_{t=1}^T p^t \cdot c^t \leq \sum_{t=1}^T c_i^t + \frac{1}{\varepsilon} \ln(m) + \varepsilon T$$

הוכחה:

$$\phi^{T+1} \geq W_i^{T+1} = e^{-\varepsilon \sum_{t=1}^T c_i^t}, \quad \phi^1 = m$$

$0 < \varepsilon \leq 1$, וכן $-1 \leq c_i^t \leq +1$, ולכן $-1 \leq \varepsilon c_i^t \leq +1$.
עבור $x \in [-1, +1]$, מפיתוח טיילור:

$$e^x \leq 1 + x + x^2$$

על כן נקבל:

$$\begin{aligned} \phi^{t+1} &= \sum_{i=1}^m W_i^{t+1} = \sum_{i=1}^m W_i^t e^{-\varepsilon c_i^t} \leq \sum_{i=1}^m W_i^t (1 - \varepsilon c_i^t + \varepsilon^2 (c_i^t)^2) \\ &\leq \sum_{i=1}^m W_i^t (1 - \varepsilon c_i^t + \varepsilon^2) = \sum_{i=1}^m W_i^t (1 + \varepsilon^2) - \sum_{i=1}^m W_i^t \cdot \varepsilon c_i^t \\ &= (1 + \varepsilon^2) \phi^t - \varepsilon \sum_{i=1}^m \phi^t p_i^t c_i^t = \phi^t (1 + \varepsilon^2 - \varepsilon p^t \cdot c^t) \\ &\leq \phi^t e^{\varepsilon^2 - \varepsilon p^t \cdot c^t} \\ &\Rightarrow \forall i: \phi^{T+1} \geq e^{-\varepsilon \sum_{t=1}^T c_i^t} \\ &\Rightarrow \forall i: \phi^{T+1} \leq m \cdot e^{\varepsilon^2 T - \varepsilon \sum_{t=1}^T p^t \cdot c^t} \\ &\Rightarrow \forall i: \varepsilon \sum_{t=1}^T c_i^t \leq \ln(n) m + \varepsilon^2 T - \varepsilon \sum_{t=1}^T p^t \cdot c^t \end{aligned}$$

ולאחר העברת אגפים וכו' נקבל את המשפט.
 אם המשקלים הם בקטע $[-\rho, +\rho]$, אז אפשר להעביר לקטע $[-1, +1]$ ע"י חלוקה ב- ρ , ואז להריץ את *Hedge*, ולכפול בחזרה:

$$\sum_{t=1}^T p^t \cdot c^t \leq \sum_{t=1}^T c_i^t + \frac{\rho}{\varepsilon} \ln(n) + \varepsilon T$$

נחזור למקרה הבסיסי - אם $T \geq \frac{1}{\varepsilon^2} \cdot \ln(m)$, אזי לכל חזאי i :

$$\frac{1}{T} \sum_{t=1}^T p^t c^t \leq \frac{1}{T} \sum_{t=1}^T c_i^t + 2\varepsilon$$

כלומר צעד עולה בממוצע בממוצע 2ε ועוד העלות של ההכי טוב, במידה ומספר הצעדים מספיק גדול. לזה יש שיפור קטן:

3.3.4 הגדרת שיטת העדכון הכפלי

לא נוכיח, אך ההוכחה זהה עבור ההוכחות הקודמות.
 שוב המשקולות $c_i^t \in [-1, +1]$, אך הפעם:

$$W_i^{t+1} \leftarrow W_i^t (1 - \varepsilon c_i^t)$$

השגיאה:

$$\sum_{t=1}^T p^t \cdot c^t \leq \sum_{t=1}^T c_i^t + \frac{1}{\varepsilon} \ln(m) + \varepsilon \sum_{t=1}^T |c_i^t|$$

אפשר לדבר על הבעיה הזו גם כאשר הקטע לא סימטרי.
 ה- ρ שראינו נקרא הרוחב של הקטע, וזהו מושג חשוב בניתוחים ובשימושים של השיטה הזו.
 כאן ניסינו למזער את הקנס - אפשר להסתכל גם על הבעיה בה מנסים למקסם את ה"פרס", ואז להפוך את הסימנים ושוב להפעיל על *hedge*.

3.3.5 משחקים סופיים סכום אפס בשני שחקנים

זהו מודל מתורת המשחקים.
 במשחק כזה יש שני שחקנים - MIN ו- MAX . מטרתו של הראשון להקטין ככל האפשר את תוצאת המשחק, השני - למקסם. ישנן קבוצות סופיות של אסטרטגיות עבור שחקנים אלו S_{MIN} ו- S_{MAX} , כאשר עבור $i \in S_{MIN}$, $j \in S_{MAX}$, יש ערך $M(i, j) \in [0, 1]$ - כלומר, מהו המהלך הבא.
 בחיים רוב המשחקים הם לא משחקים סכום אפס, אבל יש כאלו - שוק המרות מטבע חוץ הוא כזה למשל (על כל אחד שמרוויח יש אחד שמפסיד), בניגוד לשוק המניות (בו כולם יכולים להרוויח).
 זהו לא מודל טבעי לתיאור משחקים, כי הוא לא מתאר משחק כמהלכים ומהלכי-נגד.
 במשחקים, צריך לדבר על אסטרטגיות. S_{MIN} , S_{MAX} הן קבוצות האסטרטגיות הטהורות. לרב אולם נדבר על אסטרטגיה מעורבת:

הגדרה 3.3 אסטרטגיה מעורבת עבור שחקן MIN היא התפלגות D על S_{MIN} , ועבור שחקן ה- MAX היא התפלגות P על S_{MAX} .

כלומר, ניתן לחשוב על אסטרטגיה מעורבת כאלגוריתם הסתברותי.

משפט המינימקס (von - Neumann):

3.4 משפט

$$\lambda^* = \min_D \max_{j \in S_{MAX}} \{\mathbb{E}_{i \sim D} M(i, j)\} = \max_P \min_{i \in S_{MIN}} \{\mathbb{E}_{j \in P} M(i, j)\}$$

כלומר, אין חשיבות לסדר הבחירה.

אנחנו בעצם מכירים את ההוכחה הזו - בעצם דואליות של תוכנית לינארית! השוויון נובע מכך שאחד הדואלי של השני.
יש משפטים דומים למשחקים כלליים.

ניתן מפת דרכים לקראת השבוע הבא:
נרצה למצוא את האסטרטגיות האופטימליות האלה. אם נרצה באופן מדויק, בעיה זו שקולה לתכנון לינארי. מה שננסה לעשות כאן הוא דבר אחר - נראה שבצורה הרבה יותר יעילה אפשר למצוא קירובים טובים לאסטרטגיות האופטימליות:
נרצה עבור פרמטר $\delta > 0$ למצוא אסטרטגיות מעורבות $\hat{D}, \hat{P}$, עבורן:

$$\lambda^* - \delta \leq \min_{i \in S_{MIN}} \mathbb{E}_{j \sim \hat{P}} M(i, j), \quad \max_{j \in S_{MAX}} \mathbb{E}_{i \sim \hat{D}} M(i, j) \leq \lambda^* + \delta$$

כך נראים רב השימושים בשיטה הזו - השיטה הזו נותנת לנו בפרט שיטות מהירות לפתור בעיות בקירוב טוב (אם כי לא בצורה אופטימלית).

מה בקשר בין זה לבין הבעיה הקודמת? נראה שיש פה קשר:
ה"חזאים" שלנו הם איברי S_{MIN} , ה"אירועים" (מה שהיריב מייצר) נלקחים מ- S_{MAX} , וה"מחיר" של חזאי $i \in S_{MIN}$ עבור "אירוע" j הוא $M(i, j)$. נשים לב כי האלגוריתם לבעיית החזאים מגדיר התפלגות על החזאים, כלומר אסטרטגיה מעורבת. הרוחב פה הוא 1.

נריץ את המשחק הזה איזשהוא מספר של צעדים: נתחיל מהתפלגות כלשהיא. בכל צעד היריב "יכה אותנו" באסטרטגיה אופטימלית (הנחה חישובית שלנו - בהנתן התפלגות מסויימת, באותו התור ניתן לחשב את האטרטגיה האופטימלית עבור ה- MAX). אנחנו "נכה" את ההתפלגות הנוכחית שלנו ב- j הכי גרוע לנו. כתוצאה מכך נחשב התפלגות חדשה, כלומר התפלגות על S_{MIN} . "נכה אותה" שוב באסטרטגיה הכי גרועה עבורה, וכך הלאה...
נריץ את האלגוריתם מספר מסויים של צעדים, ובסוף הריצה הזו תהיה לנו התפלגות מסויימת. נטען כי לאחר מספר מסויים של צעדים, ההתפלגות מתכנסת לאסטרטגיה יעילה.

מה יהיה $\hat{P}$ במשחק הזה?
בכל צעד משחקים איזשהיא אסטרטגיה "טהורה" עבור שחקן ה- MAX . כדי לחשב את $\hat{P}$, פשוט נמצע - מספר הפעמים ששיחקנו את ה- j הזה, חלקי מספר הצעדים הכולל.