סיבוכיות

ⓒ ארזים

6 ביוני 2017

1 קושי לעומת אקראיות

נציג שתי טענות יחסית אינטואיטיביות.

טענה 1.1 האקראיות עוזרת. הכוונה היא למשל שיש בעיות מתוך BPP שאינן בתוך P.

טענה 1.2 יש פונקציות קשות לחישוב. הכוונה היא למשל שבמחלקה E יש שפות שדורשות מעגלים בגודל $2^{\Omega(n)}$.

שתי הטענות האלה נראות נכונות, אבל אנחנו לא יודעים להוכיח אותן. לעומת זאת, אנחנו כן יודעים להוכיח תוצאה על הקשר ביניהן:

משפט 1.3 אם במחלקה E יש שפה הדורשת מעגלים בגודל $2^{\Omega(n)}$, אזי $\mathrm{BPP} = \mathrm{P}$.

הרעיון משתמשים בפונקציה הקשה כדי לייצר כמות פולינומית של מחרוזות שנראות אקראיות לאלגוריתם BPP.

2 הוכחות אינטראקטיביות

אפשר לחשוב על NP בתור מחלקת השפות שלהן יש פרוטוקול בין מוכיח למוודא, כך שהמוכיח שולח הוכחה פולינומיאלית למוודא, שבזמן פולינומיא בודק נכונות.

שאלה האם יש יותר כח כאשר מרשים יותר סיבובי תקשורת (כמות פולינומיאלית של סיבובים)?

תשובה לא - אם V דטרמיניסטי אפשר לדחוס כמות פולינומיאלית של סיבובים לסיבוב יחיד.

שאלה מה אם נרשה למוודא V להיות אלגוריתם BPP?

משפט 2.1 מחלקת השפות להן יש הוכחה אינטראקטיבית עם מוודא BPP היא PSPACE.

דוגמא נדבר על בעיית Graph Isomorphism. נתונים שני גרפים G_1, G_2 על n קודקודים. נאמר כי $G_1 \cong G_2$ אם יש העתקה בין הקודקודים $\pi : V(G_1) \to V(G_2)$ חד־חד־ערכית ועל בין $E(G_1), E(G_2)$.אחרת נאמר כי $G_1 \ncong G_2$. ברור שבעיית Graph Isomorphism (GISO) שייכת למחלקה NP. לא ידוע לגבי Graph Non-Isomorphism (GNISO). אנחנו נראה פרוטוקול אינטראקטיבי עבורה.

פרוטוקול בכל סיבוב, V יבחר $b \in \{1,2\}$ באקראי, ופרמוטציה $\pi : [n] \to [n]$ באקראי, וישלח את $\pi(G_b)$. P יחזיר ביט b'. V יקבל אם $b = b'$.

2.1 הוכחות באפס ידיעה

דוגמא נתבונן בשפה 3COL של גרפים שהם שלוש־צביעים. אנחנו יודעים שהיא NP שלמה.

פרוטוקול נראה איך אפשר להוכיח שגרף הוא שלוש־צביע בלי להציג את הצביעה. P יצבע את קודקודי הגרף ויכסה אותם. V יציע על קשת באקראי ויבקשת לחשוף את צבעי הקודקודים בקצוות הקשת. V מקבל אם הצבעים הם כחול/אדום/ירוק ושני הקודקודים בצבעים שונים. אם יש צביעה חוקיתת אפשר לגרום לו לקבל בהסתברות 1. אם אין, כל צביעה היא לא חוקית ותתגלה בהסתברות לפחות $\frac{1}{|E|} \geq \frac{1}{n^2}$. אם נרצה לחזור על הפרוטוקול כמה פעמים, P יבצע פרמוטציה לשמות הצבעים ויצבע את הגרף בהתאם.

3 אופטימיזציה, קירובים ובעיות פער

3.1 בעיות אופטימיזציה

נתון לנו יחס $R \subseteq \{0,1\}^* \times \{0,1\}^*$, כאשר חושבים על זוג (x,y) בתור קלט (x) ופתרון מוצע (y), כאשר $|y| \leq |x|^c$ וניתן לחשב ביעילות את R. כמו כן, לכל y עבורו $(x,y) \in R$ יש ערך, כלומר

$$\mathrm{Val} : Y_x \to \mathbb{R}_+$$

כאשר $Y_x = \{y \mid (x,y) \in R\}$. נסמן

$$\mathrm{opt}(x) = \max_{y \in Y_x} \mathrm{Val}(y)$$

הגדרה 3.1 נאמר כי אלגוריתם A מוציא α־קירוב אם בהינתן קלט x, A מוציא y עבורו

$$\frac{1}{\alpha}\mathrm{opt}(x) \leq \mathrm{val}(y) \leq \mathrm{opt}(x)$$

הערה 3.2 באופן סימטרי אפשר להגדיר בעיית מינימליזציה.

דוגמא בעיית $\mathrm{Vertex-Cover}$ היא למצוא קבוצת קודקודים קטנה ביותר בגרף שנוגעת בכל הקשתות. בעיה זו NP קשה. האלגוריתם החמדן מוסיף לקבוצה שלנו תמיד את הקודקוד בעל הדרגה המקסימלית שלא בחרנו.

טענה 3.3 האלגוריתם החמדן מוציא פתרון גדול פי $\log n$ מהפתרון המינימלי.

הוכחה: ניקח $m = 2^k$ ונבנה גרף עם m שכבות. בשכבה i יהיו $\frac{m}{2^i}$ קודקודים, וכל קודקוד ברמה i יחובר אל 2^i קודקודים ברמה 0 (שונים). הכי כדאי לקחת את כל הקודקודים ברמה 0, אבל האלגוריתם החמדן ייקח את הקודקודים במורד הרמות (יאסוף את כל הקודקודים). גודל הגרף הוא בערך $m \log m$, ולכן המנה היא אכן $\log m$. ■

טענה 3.4 יש אלגוריתם 2־קירוב עבור $\mathrm{Vertex}-\mathrm{Cover}$.

הוכחה: נתאר את האלגוריתם. נתחיל עם $S=\emptyset$. כל עוד יש קשת לא מכוסה, נוסיף את שני הקודקודים שלה לתוך S, ונמחק את הצלעות שנוגעות בהם. נראה שפתרון זה יהיה גדול לכל היותר פי 2 מהפתרון האופטימלי.

הקשתות שנבחרו באלגוריתם מהוות זיווג (קשתות שזרות בקודקודים). בתוך $\mathrm{Vertex}-\mathrm{Cover}$ חייב להופיע לפחות קודקוד אחד מכל קשת בזיווג, לכן הפתרון האופטימלי הוא לפחות $\frac{1}{2}$ משלנו.

נתאר אלגוריתם נוסף. לכל קודקוד יהיה משתנה x_v, שמקבל ערכים $x_v\in\{0,1\}$. לכל $e=(v,u)$ נדרוש $x_v+x_u\geq 1$, ונרצה $\mathrm{opt}=\min\sum_v x_v$. זו בעיה של תכנות לינארי בשלמים. נפתור אותה לא בשלמים, כלומר $x_v\in[0,1]$. נקבל ערך אופטימאלי חדש opt^*, שהוא לכל היותר גדול כמו opt. אנחנו נפתור את התכנות הלינארי הזה ונגדיר

$$S=\left\{v\mid x_v\geq\frac{1}{2}\right\}$$

נשים לב כי S הוא אכן פתרון ־ לכל קשת $e=(v,u)$ מתקיים

$$x_u+x_v\geq 1$$

ולכן אחד מהם הוא לפחות $\frac{1}{2}$. כמו כן,

$$|S|=\sum_{v\in S}1\leq 2\sum_{v\in S}x_v\leq 2\sum_{v\in V}x_v=2\mathrm{opt}^*\leq 2\mathrm{opt}$$

■

בעיה פתוחה: האם יש אלגוריתם יעיל המוציא $2-\varepsilon$ קירוב של הבעיה ($\varepsilon>0$).

בעיה נתבונן בבעיה $\mathrm{Set}-\mathrm{Cover}$. הקלט הוא עולם בגודל n, נאמר $[n]$. נתונות תת קבוצות $A_1,\ldots,A_m$ של העולם, כאשר $m\leq n^c$, כך שמתקיים

$$\bigcup_{i=1}^{m}A_i=[n]$$

ננסה למצוא מספר מינימלי של קבוצות שאיחודן הוא $[n]$. נשים לב שזה מכליל את הבעיה הקודמת ־ שם יכולנו לקחת את העולם להיות E, כשלכל $v\in V$ יש קבוצה $A_v=\{e\mid v\in e\}$. פתרון של זה יהיה פתרון של $\mathrm{Vertex}-\mathrm{Cover}$. ראשית נתאר אלגוריתם חמדן לבעיה שלנו:

אלגוריתם בכל שלב נוסיף לניסוי את הקבוצה שמכסה הכי הרבה איברים לא מכוסים.

טענה 3.5 הפתרון שנקבל הוא לכל היותר $\log n$ גדול יותר מהאופטימלי.

הוכחה: נניח שהפתרון האופטימלי הוא בגודל k. נשים לב שבכל שלב של האלגוריתם החמדן יש קבוצה שמכסה לפחות $\frac{1}{k}$מהקודקודים שנותרו. לכן בשלב 0 יש n קודקודים לכסות. אחרי שלב 1 יש לכל היותר $n\left(1-\frac{1}{k}\right)$. אחרי שלב 2 יש לכל היותר $n\left(1-\frac{1}{k}\right)^2$. באופן כללי, בשלב t , נותרו לכל היותר $n\left(1-\frac{1}{k}\right)^t$. אם $t=k\ln n+1$, אז לא יישאר אף קודקוד לא מכוסה. לכן הפתרון שלנו הוא בגודל לכל היותר $k\ln n+1$. ■

3.2 בעיות פער

עבור בעיית אופטימיזציה, נגדיר בעיית הכרע המתאימה. נניח לרגע שמדובר בבעיית מקסימום (מינימום דורש טיפול דומה). בהינתן בעיית אופטימיזציה וקבועים $\beta < \alpha$ נרצה להכריע את השאלה הבאה:

שאלה נתון קלט x, ומובטח שמתקיים אחד משני המצבים הבאים:

1. יש $y \in Y_x$ עם $\mathrm{Val}(y) \geq \alpha$.
2. לכל $y \in Y_x$ מתקיים $\mathrm{Val}(y) < \beta$.

נרצה להכריע באיזה מהמצבים אנחנו נמצאים.

טענה 3.6 אם יש אלגוריתם $\frac{\alpha}{\beta}$ קירוב אז ניתן לפתור את בעיית ההכרעה בזמן פולינומי.

הוכחה: נניח כי x במצב 1 (זה לא משנה, אם צודקים בוודאות במצב זה אז עונים הפוך לכאלה במצב ב'). נריץ את אלגוריתם הקירוב, ונקבל אם הערך שחזר גדול מאשר β. ברור שנקבל רק אם x מראש מקיים את מצב 1. כמו כן, מההטחה על האלגוריתם, הערך שנקבל הוא לפחות

$$\frac{1}{\frac{\alpha}{\beta}}\mathrm{opt}(x) \geq \beta$$

שהרי הנחנו כי $\mathrm{opt}(x) \geq \alpha$. ■

3.2.1 רדוקציות משמרות פער

בעיות פער מסומנות

$$\mathrm{Gap}{-}R[\alpha, \beta]$$

כאשר R הוא היחס והפרמטרים הם α, β.

הגדרה 3.7 בהינתן שתי בעיות $\mathrm{Gap} - R_1[\alpha_1, \beta_1]$, $\mathrm{Gap} - R_2[\alpha_2, \beta_2]$, נאמר שרדוקציה f ביניהן היא משמרת פער אם כאשר x הוא כזה שמקיים

$$\mathrm{Val}_{R_1}(x) \geq \alpha_1$$

אזי

$$\mathrm{Val}_{R_2}(f(x)) \geq \alpha_2$$

ואם

$$\mathrm{Val}_{R_1}(x) < \beta_1$$

אזי

$$\mathrm{Val}_{R_2}(f(x)) < \beta_2$$