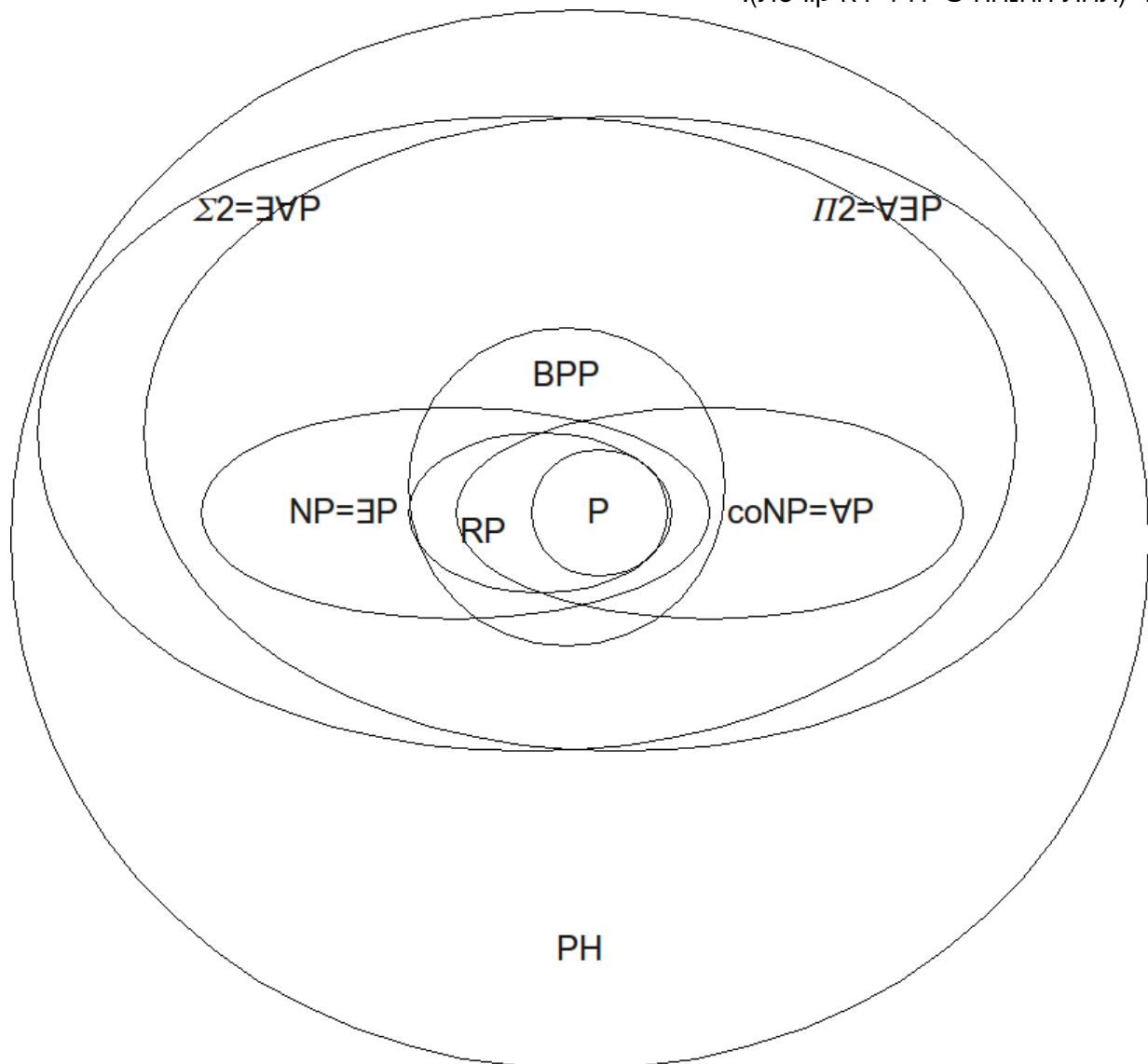


שיעור 9 - 11.05.2010

ראינו כי $P \subseteq RP \subseteq BPP \subseteq PSPACE$, $P \subseteq RP \subseteq NP \subseteq \Sigma_2 \subseteq \Sigma_3 \subseteq \dots \subseteq PH \subseteq PSPACE$
 משפט: $BPP \subseteq \Sigma_2$.
 ציור (תחת ההנחה ש-PH לא קורסת):



הערה: ממילא $BPP = coBPP \subseteq co\Sigma_2 = \Pi_2$.
 הוכחה:

תהי $L \in BPP$. בפרט, יש מ"ט הסתברותית $M(x, y)$, $|y| = m = \text{Poly}(|x|)$, ומתקיים:

$$1. \Pr_y[M(x, y) = 1] \geq \frac{1}{2} \Leftrightarrow x \in L$$

$$2. \Pr_y[M(x, y) = 1] \leq \frac{1}{4m} \Leftrightarrow x \notin L$$

שקול לכך ש- $\exists y_1, \dots, y_{2m} \in \{0, 1\}^m. \forall z \in \{0, 1\}^m. (M(x, y_1 \oplus z) = 1) \vee \dots \vee (M(x, y_{2m} \oplus z) = 1)$
 Σ_2 . הפסוק הראשון הוא פסוק $\exists i \in [1, 2m]. M(x, y_i \oplus z) = 1$.

נסמן: $G = \{y \in \{0, 1\}^m \mid M(x, y) = 1\}$.

איזה z יקיימו $M(x, y_1 \oplus z) = 1$? כל ה- z ים כך ש- $y_1 \oplus z \in G$, כלומר $z \in y_1 \oplus G$, כאשר

$$y_1 \oplus G = \{y_1 \oplus g \mid g \in G\}$$

נשים לב ש- $\forall y. |y \oplus G| = |G|$.

אם $x \notin L$:

אז לכל בחירה $y_1, \dots, y_{2m} \in \{0, 1\}^m$,

$z \in (y_1 \oplus G) \cup \dots \cup (y_{2m} \oplus G) \Leftrightarrow (M(x, y_1 \oplus z) = 1) \vee \dots \vee (M(x, y_{2m} \oplus z) = 1)$ מקיים z ולכן, מספר ה- z ימים שמספקים את הפסוק הוא לכל היותר $|y_1 \oplus G| + \dots + |y_{2m} \oplus G|$.

$|y_1 \oplus G| + \dots + |y_{2m} \oplus G| = 2m \cdot |G| \stackrel{|G| \leq \frac{2^m}{4m}}{\leq} 2m \cdot \frac{2^m}{4m} \leq \frac{2^m}{2}$ ולכן בפרט יש z שעבורו הפסוק

לא נכון, ולכן כל הפסוק False.

זה נכון כי גודל מרחב ה- z ימים הוא 2^m , וכמות ה- z ימים ה"טובים" היא לכל היותר חצי מהמרחב. לכן, יש z ימים בחצי השני.

אם $x \in L$:

לכל $z \in \{0, 1\}^m$,

$$\Pr_{\substack{y_1, \dots, y_{2m} \\ \text{בחרות בלתי תלויות}}} \underbrace{\left[z \in (y_1 \oplus G) \vee \dots \vee z \in (y_{2m} \oplus G) \right]}_{\text{מאורעות בלתי תלויים}} =$$

$$= 1 - \Pr_{y_1, \dots, y_{2m}} \left[z \notin (y_1 \oplus G) \wedge \dots \wedge z \notin (y_{2m} \oplus G) \right] = 1 - \prod_{i=1}^{2m} \Pr_{y_i} [z \notin y_i \oplus G] \geq 1 - 2^{-2m}$$

$$\Pr_y [z \notin y \oplus G] = \Pr_y \left[\underbrace{z \oplus y}_{\substack{\text{איבר אקראי} \\ \text{ב- } \{0, 1\}^m}} \notin G \right] = \frac{2^m - |G|}{2^m} \leq \frac{1}{2} \text{ ולכן } |G| \geq \frac{2^m}{2} \text{ אם } x \in L, \text{ אזי}$$

ראינו שלכל $z \in \{0, 1\}^m$, $\Pr_{y_1, \dots, y_{2m}} [z \text{ לא מכוסה}] \leq 2^{-2m}$.

$$\Pr_{y_1, \dots, y_{2m}} [z \text{ איזושהי } z \text{ לא מכוסה}] = \Pr_z \left[\bigvee_z \text{ לא מכוסה} \right] \leq$$

$$\leq \sum_z \Pr [z \text{ לא מכוסה}] \leq 2^m \cdot 2^{-2m} = 2^{-m} < 1$$

בפרט, יש $y_1, \dots, y_{2m} \in \{0, 1\}^m$ שמכסים את כל ה- z ימים, והפסוק True.

אלגוריתמי קירוב

בעיית MAX-CUT

קלט: גרף $G = (V, E)$.

פלט: הגודל של החתך הכי גדול בגרף, ומהו.

בעיה זו היא NP-קשה. לכן, נרצה לבצע לה קירוב.

נמצא חתך שגודלו "כמעט" כמו החתך הכי גדול בגרף.

בתרגיל: תמיד יש חתך שגודלו $\frac{|E|}{2}$, וגם ראינו איך למצוא אותו.

הערה: אם פותרים בעיה NP-קשה אחת עם פקטור קירוב 2, אז (כמו עכשיו) זה לא אומר כלום

על קירוב של בעיות קשות אחרות.

נחזור לבעיית MAX-CUT בהמשך.

דוגמה 1: Vertex Cover - VC

קלט: $G = (V, E)$.

פלט: קבוצה $A \subseteq V$ שמכסה את כל הקשתות וגודלה מינימלי.

בעיה זו היא NP-קשה, ולכן נרצה לקרב אותה.

אלגוריתם קירוב ל-VC

הצעה: כל פעם ניקח את הקודקוד עם הדרגה הכי גבוהה, ונזרוק את כל הקשתות שנוגעות בו (זה

אלגוריתם חמדני). כל פעם עושים את הצעד ה"לוקלי" הכי טוב.

האלגוריתם:

1. נמצא זיווג מקסימלי בגרף, $e_1, \dots, e_t$.

הערה: זיווג מקסימלי (*maximal*) - זיווג שלא ניתן להרחבה. לעומת זיווג אופטימלי

(*maximum matching*) שהוא הזיווג הגדול ביותר הקיים בין כל הזיווגים.

2. נוציא כפלט את כל הקודקודים $v_1, \dots, v_{2t}$ שמופיעים בקשתות $e_1, \dots, e_t$.

נכונות:

זהו באמת כיסוי קודקודי, כי לכל $e \in E$, אם e לא נוגעת ב- $v_1, \dots, v_{2t}$, אז אפשר להוסיף אותה לזיווג, בסתירה למקסימליות. לכן, זהו כיסוי קודקודי.
הכיסוי "כמעט" מינימלי: כל כיסוי מכסה את כל הצלעות, ולכן בפרט הוא מכסה את $e_1, \dots, e_t$. ולכן, חייב לקחת קודקוד אחד (לפחות) מכל קשת, והקודקודים האלה זרים, ולכן צריך לפחות t קודקודים. קיבלנו כיסוי בגודל $2t$, כשכל כיסוי גודלו לפחות t , דהיינו, קירוב בפקטור 2.

בעיית אופטימיזציה מוגדרת על ידי: לכל קלט x , מוגדר אוסף פתרונות אפשריים $W(x)$ (למשל VC, קליקה בגודל לפחות k , וכו'), ולכל פתרון מוגדרת עלות $C(x, s)$, $x \in [0, 1]^n$, $s \in W(x)$.

דוגמאות:

1. x - גרף, $W(x)$ - אוסף החתכים, $C(x, s)$ - מספר הקשתות בחתך. מחפשים C מקסימלי.
2. x - גרף, $W(x)$ - אוסף כיסויים קודקודיים, $C(x, s)$ - מספר הקודקודים בכיסוי. מחפשים C מינימלי.

לבעיית מקסימום, יש פתרון אופטימלי בעלות $OPT(x)$: נגיד שיש לנו אלגוריתם c -קירוב לבעיה. אם לכל קלט x , נמצא פתרון $s(x)$ כך ש- $Cost(x, s(x)) \leq OPT(x)$.
 $\frac{OPT(x)}{c} \leq Cost(x, s(x)) \leq OPT(x)$.
לביית מינימום, $OPT(x) \leq Cost(x, s(x)) \leq c \cdot OPT(x)$.
ראינו: אלגוריתם פולינומי דטרמיניסטי שפותר את בעיית 2-קירוב ל-VC.

דוגמה 2: SC - Set Cover

קלט: עולם $W = \{w_1, \dots, w_m\}$ ואוסף קבוצות $S_1, \dots, S_n \subseteq W$.

פלט: $i_1, \dots, i_k \in [n]$ כך ש- $W = S_{i_1} \cup \dots \cup S_{i_k}$.

הערה:

בעיית ה-SC היא יותר כללית מבעיית VC, כי בבעיית VC רוצים לכסות את כל הקשתות עם קודקודים, כמופע SC כאשר $W = E = \{e_1, \dots, e_n\}$, ולכל קודקוד v ,

$$S_v = \left\{ \begin{array}{l} \text{כל הקשתות} \\ \text{שנוגעות ב-} v \end{array} \right\}$$

אלגוריתם (חמדני):

בכל שלב, נבחר את הקבוצה S_i שמכסה הכי הרבה איברים חדשים.

הערה:

במקרה הפרטי של VC, כל פעם נבחר את הקודקוד עם הדרגה הכי גבוהה, ונמחק את כל הקשתות שקשורות אליו.

נכונות:

האלגוריתם מחזיר פתרון: בסוף הריצה, כל העולם אכן מכוסה.

כמה רע זה יכול להיות? נניח שנתון לנו קלט $S_1, \dots, S_n$, שאפשר לפתור בעלות OPT . באיזה עליות האלגוריתם החמדן יפתור אותו?

בהתחלה, יש m איברים לכסות. אפשר לכסות אותם עם $S_{i_1}, \dots, S_{i_{OPT}}$, ולכן יש קבוצה

$$|S_{i_j}| \geq \frac{m}{OPT} \quad (\text{לפי שובר היונים}). \text{ לכן, בפרט, הקבוצה הכי גדולה מבין } S_1, \dots, S_n \text{ גודלה}$$

$$\text{לפחות } \frac{m}{OPT}, \text{ ולכן יישאר לנו לכסות לכל היותר } \left(1 - \frac{1}{OPT}\right) \cdot m \text{ איברים.}$$

נסמן: r_k - כמה איברים נותרו לאלגוריתם החמדן אחרי k צעדים. מתחילים עם $r_0 = m$.

באופן כללי, אם אחרי k צעדים, נותרו לנו r_k איברים, ניתן לכסות אותם על ידי OPT קבוצות, ולכן הקבוצה שנבחר (הכי גדולה שאפשר) תכיל לפחות $\frac{r_k}{OPT}$ איברים חדשים.

$$\left\{ \begin{array}{l} \underbrace{r_{k+1}}_{\substack{\text{כמה נשאר אחרי} \\ \text{הצעד הנוסף}}} \leq \left(1 - \frac{1}{OPT}\right) \cdot r_k \\ r_0 = m \end{array} \right. \text{ , לכן ,}$$

מטרה: למצוא t כך ש- $r_t < 1$.

$$r_t \leq \left(1 - \frac{1}{OPT}\right)^t \cdot r_0 = \left(1 - \frac{1}{OPT}\right)^t \cdot m \stackrel{1-x \leq e^{-x}}{\leq} e^{-\frac{t}{OPT}} \cdot m$$

לכן, אם ניקח $t = OPT \cdot \ln m$, אז $r_t < e^{-\ln m} \cdot m = 1$.

האלגוריתם החמדם המשתמש בכלל היותר $OPT \cdot \ln m$ קבוצות, משיג יחס קרוב $\ln m$.

שאלות:

1. האם האלגוריתם החמדם משיג אולי יחס קירוב יותר טוב?
2. האם אולי יש אלגוריתם קירוב פולינומי אחר שמשיג יחס קירוב יותר טוב?
3. מה יחס הקירוב של האלגוריתם החמדם עבור VC?

נחזור לבעיית MAX-CUT:

קלט: גרף $G = (V, E)$.

פלט: **גודל** חתך מקסימלי בגרף.

נרצה למצוא חתך בגרף שגודלו "כמעט" מקסימלי.

אלגוריתם קירוב:

$$\text{הבעיה שקולה לתוכנית } \max \frac{1}{2} \cdot \sum_{(i,j) \in E} (1 - y_i y_j) \text{ , כאשר } \forall i \in [n] . y_i \in \{\pm 1\} .$$

הסבר: אם y_1, y_2 באותו צד (כלומר שניהם 1 או שניהם -1), אז מכפלתם היא 1, ולכן

$$\text{נקבל תרומה של } 1 - 1 = 0 \text{ לסכום. אם הם בצדדים שונים, נקבל תרומה של } \frac{1}{2} \cdot (1 + 1) = 1$$

לסכום.

$Linear Programming$ - בעיה ב- P .

כשהבעיה היא לא לינארית - יש מקרים שידוע שהם ב- P , ויש מקרים שידוע שהם קשים.

$Integer Programming$ - NP-קשה.

נכונות:

$$\text{אם יש חתך } A \subseteq V \text{ שגודלו } k \text{ , אזי נבחר } y_i = \begin{cases} 1 & i \in A \\ -1 & i \notin A \end{cases} \text{ , ואז } e \in E \text{ תורמת לסכום } 1$$

אם היא בחתך, או 0 אחרת. התוכנית תחזיר לפחות k .

אם יש פתרון שגודלו k , נסמן $A = \{i | y_i = 1\}$, $\bar{A} = \{i | y_i = -1\}$. אזי A חתך בגודל k .

במקום לפתור את הבעיה בשלמים, נפתור את הבעיה בעולם אחר.

$$\max \frac{1}{2} \cdot \sum_{(i,j) \in E} \left(1 - \underbrace{v_i \cdot v_j}_{\text{מכפלה פנימית}}\right) \text{ , כאשר } v_i \in \mathbb{R}^n \text{ , } \|v_i\| = 1 \text{ לכל } i .$$

נכונות:

הערך שהתוכנית מחזירה הוא לפחות OPT , כי ראינו שיש פתרון ב- $-1, 1$ שפותר את

הבעיה. אפשר להרחיב אותו באמצעות $v_i = (y_i, 0, \dots, 0)$.

את התוכנית הזאת אפשר לפתור ב- P בזמן פולינומי דטרמיניסטי.

איך נחליץ מהפתרון $v_1, \dots, v_n$ חתך?

נבחר על מישור אקראי $w \in \mathbb{R}^n$, $\|w\| = 1$.

כל הקודקודים ש- $v_i \cdot w > 0$ יהיו בצד אחד.

כל הקודקודים ש- $v_i \cdot w < 0$ יהיו בצד השני.