4.1.2020

פרויקט תכנות בנושא תזמון

קורס תפ"י 2

מגישות:

נוי כשר 314963810 מאי דניאל 313177073

קבוצה 52

חלק א' – חלק תיאורטי

נתון:

סט של n משימות $\mathrm{J} = \{\mathrm{J}_1, \dots, \mathrm{J}_\mathrm{n}\}$ שיש לבצע על מכונה בודדת. לכל משימה J_j (j=1,…,n) נתון:

$\mathrm{t}_\mathrm{j}, \mathrm{d}_\mathrm{j}, \mathrm{e}_\mathrm{j}$, זמן ביצוע, זמן היעד ועלות העברה למיקור חוץ בהתאמה.

נניח:

1. $\mathrm{t}_\mathrm{j} \le \mathrm{d}_\mathrm{j}$ עבור כל משימה.
2. כל הפרמטרים בקלט שלמים ואי שליליים.
3. ללא הגבלת הכלליות, כלל המשימות ממוספרות לפי EDD.

מטרת החברה:

למזער את עלות העברת המשימות למיקור החוץ תחת האילוץ שכל המשימות שנתזמן על המכונה הבודדת יסתיימו לכל המאוחר בתאריך היעד שלהן.

פתרון לבעיה מכיל שתי קבוצות של החלטות:

1. חלוקת קבוצת המשימות לשתי תתי קבוצות זרות ומשלימות: A – קבוצת המשימות שנבצע בארגון, $\bar{\mathrm{A}}$ – קבוצת המשימות שנוציא למיקור חוץ.
2. תזמון המשימות בקבוצה A.

<u>סעיף א':</u>

צ"ל: יש לתזמן את המשימות בקבוצה A על המכונה הבודדת לפי EDD.

הוכחה:

מהגדרת הבעיה אנו יודעים שקבוצה A הינה קבוצת המשימות שהוחלט לבצע על המכונה הבודדת בתוך הארגון. מכך נובע שקיים סידור עבורו $c_j \le d_j$ לכל j בקבוצה A, כלומר אין פיגורים $T_{max} = 0$.

בכיתה הוכחנו כי סידור EDD ממזער את האיחור המקסימלי L_{max} ועשינו זאת ע"י החלפה זוגית.

נוכיח כי סידור EDD ממזער את הפיגור המקסימלי T_{max}:

נזכיר: $T_{max} = max\{0, L_j\}$

<u>מקרה 1:</u> אם $L_{max} \le 0$ אז $T_{max} = 0$.

<u>מקרה 2:</u> אם $L_{max} \ge 0$ אז $T_{max} = L_{max}$. אך מקרה זה לא ייתכן משום שהוא סותר את הגדרת קבוצה A כקבוצה שכלל המשימות בה מסתיימות לפני תאריך היעד שלהן.

סעיף ב':

נדרש: יש להציג אלגוריתם פולינומיאלי, יעיל ככל האפשר, העונה על השאלה: האם קיים פתרון אפשרי בו עלות מיקור החוץ הינה אפס ? יש להציג את סיבוכיות החישוב.

האלגוריתם הפולינומיאלי:

הראינו בסעיף א' שסידור EDD ממזער את T_{max}.

מההנחה המשימות בחלק התיאורטי מסודרת כבר EDD.

1. עבור כל משימה חשב את זמן הסיום שלה, c_j ← $O(n)$
2. עבור כל משימה חשב את L_j ← $O(n)$
3. עבור כל משימה חשב את T_j ← $O(n)$
4. מצא את T_{max} ← $O(n)$
5. אם $T_{max} = 0$ החזר "כן", אחרת החזר "לא". $O(1)$

סיבוכיות חישוב: $O(n)$

סעיף ג':

הוכח: הבעיה שייכת ל- NP-Hard.

הוכחה:

<u>הבנת בעיית הזימון והמרה לגרסת החלטה:</u>

עבור כל משימה, אם $c_j \leq d_j$ נשבצה בקבוצה A. אחרת, נשבצה בקבוצה $\bar{A}$.

האם קיים סידור כך ש: $\sum_{j\in\bar{A}} e_j \leq k$ וגם $c_j \leq d_j$ עבור כל משימה המוכלת ב-A.

<u>נבחר את בעיית העזר:</u>

הבעיה המתאימה תהיה בעיית החלוקה.

תזכורת – בעיית החלוקה: נתונים n איברים: $A = \{a_1, \dots, a_n\}$, כך שמתקיים $\sum a_j = B$. האם קיימת חלוקה לשתי תתי קבוצות: A ו- $\bar{A}$ כך ש: $\sum_{j\in A} a_j = \sum_{j\in\bar{A}} a_j = \frac{B}{2}$ ובנוסף $A\cap\bar{A} = \phi,\ A\cup\bar{A} = A$.

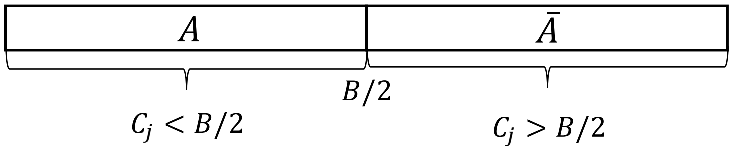
<u>נבנה מקרה פרטי שיעזור לנו להפחית את הבעיה הידועה כקשה לבעיה שלנו:</u>

נגדיר: $\sum t_j = B$

נגדיר n משימות באופן הבא: $e_j = t_j$, $\forall j\ d_j = \frac{B}{2}$ ובנוסף $k = \frac{B}{2}$. כלומר, בעיית הזימון הינה $\sum_{j\in\bar{A}} e_j \leq \frac{B}{2}$ וגם $\sum_{j\in A} t_j \leq \frac{B}{2}$.

נראה ש"כן" לבעיית החלוקה ← "כן" לבעיה שלנו:

אם קיימת חלוקה אזי מתקיים $\sum_{j\in A} t_j = \sum_{j\in \bar{A}} t_j = \frac{B}{2}$.



כל המשימות בקבוצה $\bar{A}$ יצאו למיקור חוץ, וערך המדד יהיה: $\sum_{j\in \bar{A}} e_j = \sum_{j\in \bar{A}} t_j = \frac{B}{2}$.

לכן התשובה לבעיה שלנו היא "כן".

נראה ש"לא" לבעיית החלוקה ← "לא" לבעיה שלנו:

אם לא קימת חלוקה, אזי: $\sum_{j\in A} t_j = \frac{B}{2} - \varepsilon$ וגם $\sum_{j\in \bar{A}} t_j = \frac{B}{2} + \varepsilon$ (המקרה ההפוך לא ייתכן משום ש: $\sum_{j\in A} t_j \le \frac{B}{2}$). ערך המדד יהיה: $\sum_{j\in \bar{A}} e_j = \sum_{j\in \bar{A}} t_j = \frac{B}{2} + \varepsilon$.

לכן התשובה לבעיה שלנו היא "לא".

סיכום:

הראינו שיש פתרון לבעיה שלנו אם ורק אם יש פתרון לחלוקה.

הראינו שמקרה פרטי של הבעיה הוא בעיה קשה, משום שעל מנת לפתור אותו נהיה חייבים לפתור את בעיית החלוקה הידועה כקשה, ולכן הבעיה כולה היא קשה.

הראינו שבעיית ההחלטה היא בעיה קשה, ועל כן בעיית האופטימיזציה השקולה היא בעיה קשה.

סעיף ד':

נציג את הבעיה כבעיית תכנות ליניארי:

נגדיר:

$$X = \begin{cases} 1, & \text{by the machine} \\ 0, & \text{outsourcing} \end{cases}$$

$$\min Z = \sum_{j=1}^{n} e_j (1 - X_J)$$

$$\text{s.t} \sum_{i=1}^{j} t_j X_J \le d_j \qquad , j = 1, \dots, n$$

$$X_J \in \{0,1\}$$

סעיף ה':

שאלה: מדוע ערך הפתרון האופטימאלי לבעיה המוחלשת מייצר חסם תחתון לערך הפתרון של הבעיה המקורית ? איזו בעיה קשה יותר ?

תשובה:

הבעיה המקורית הינה בעיית ניסוח לינארי בשלמים – ILP. אם נוריד את אילוץ השלמים נקבל את הבעיה המוחלשת, LP, בה ניתן לקבל ערך פתרון אופטימלי שאינו שלם. לכן:

$OPT = ILP \geq LP = LB$

מההכלה, קיבלנו שהבעיה המוחלשת הינה חסם תחתון לבעיה המקורית. כלומר, כל פתרון של ILP הוא פתרון של LP.

סיבוכיות:

בעיית תכנות לינארי בשלמים הינה בעיית NP-HARD, כלומר בעיה שזמן הריצה שלה הינו אקספוננציאלי, על כן היא קשה יותר מהבעיה המוחלשת עבורה ניתן למצוא פתרון אופטימלי בזמן פולינומיאלי, זאת בעקבות מרחב הפתרונות הרציף שלה.

סעיף ו':

נדרש: יש להציג פסואדו-קוד של אלגוריתם פסואדו-פולינומיאלי לפתרון הבעיה. יש להציג את סיבוכיות החישוב. יש להסביר מדוע האלגוריתם נקרא כך.

האלגוריתם הפסואדו-פולינומיאלי:

נגדיר מטריצה $T = t(i,z)$ עבור $i = 0,\dots,n$ ו- $z = 0,\dots,Q$ כאשר $Q = min\{\sum t_j,\ d_{max}\}$.

בתא $t(i,z)$ נציין את ערך הפתרון המינימאלי לתת-בעיה שמכילה את המשימות $(J_1,\dots J_i)$ מבין כל הפתרונות האפשריים בהם המשך הכולל של המשימות בקבוצת המשימות המקדימות E הינו z.

מההנחה המשימות בחלק התיאורטי מסודרת כבר EDD.

1. הגדר: $t(0,0) = 0$; $t(i,z) = \infty$ עבור $z = 1,\dots,Q$; $t(i,z) = \infty$ עבור $i = 1,\dots,n$ וגם $z = 0,\dots,Q$.
2. עבור $i = 1,\dots,n$:
3. עבור $z = 0,\dots,Q$:
4. אם $t(i-1,z) < \infty$:
5. אם $t(i-1,z) + e_j < t(i,z)$ אז עדכן $t(i,z) = t(i-1,z) + e_j$.
6. אם $z + t_i \leq d_i$ & $t(i-1,z) < t(i,z+t_i)$ אז עדכן $t(i,z+t_i) = t(i-1,z)$
7. סוף לולאת i
8. סוף לולאת z
9. החזר $F *= min\{t(n,z)\}$ לכל הערכים האפשריים של z.

מדוע נקרא כך: זמן הריצה שלו פולינומיאלי לגודל הקלט ולערכי הקלט.

סיבוכיות החישוב: $O(n*Q)$

סעיף ז':

הוכח/הפרך: האם האלגוריתם של מר מקורב הינו אלגוריתם קירוב ? אם כן, ספק את ערך הקירוב ההדוק ביותר.

הוכחה:

נדרש להוכיח ש - $\frac{APPROX(A)}{OPT} \leq \frac{UB(A)}{OPT} \leq \frac{UB(A)}{LB} \leq \delta$.

נחשב את החסם העליון:

בתחילה, האלגוריתם בוחר את המשימה בעלת עלות המיקור הגבוהה ביותר ב- J. בשלב זה, קבוצה A הינה קבוצה ריקה, ועל כן ניתן להוסיף את משימה זו לקבוצה A.

לא ניתן לדעת דבר על שיבוץ המשימות הנוספות לקבוצה A, ועל כן במקרה הגרוע:

$$UB(A) = \sum e_j - e_{max}$$

נחשב את החסם התחתון:

- כאשר ניתן לבצע את כל המשימות על המכונה הבודדת $LB_1 = 0$
- כאשר $\sum t_j \geq d_{max}$ בוודאות חלק מהמשימות ישלחו למיקור חוץ על מנת למנוע פיגורים, במקרה הטוב המשימה הזולה ביותר. $LB_2 = e_{min}$

➔ $LB = \max(LB_1, LB_2) = e_{min}$

➔ $\frac{APPROX(A)}{OPT} \leq \frac{UB(A)}{LB} \leq \frac{\sum e_j - e_{max}}{e_{min}} \leq \delta$ לכן האלגוריתם של מר מקורב הינו אלגוריתם קירוב.

סעיף ח':

נדרש: יש להציג היוריסטיקה פשוטה לפתרון הבעיה. סיבוכיות זמן החישוב תהיה לכל היותר $O(n^2)$.

פסואדו-קוד להיוריסטיקה:

נגדיר: $\frac{tj}{dj}$ יהיה היחס בין זמן העיבוד לזמן היעד. ככל שתוצאת חישוב זו תהיה קטנה יותר, כך נדע שזמן היעד של הג'וב גדול מזמן העיבוד.

כמו כן, נגדיר $\frac{ej}{tj/dj}$. הרציונל לחישוב זה הוא שככל שעלות מיקור החוץ גדולה יותר, והיחס בין זמן העיבוד לזמן היעד קטן יותר, תוצאת חישוב זה תגדל ואנו נדע שעלינו לשקול להכניס ג'וב זה לעיבוד במפעל ולא במיקור חוץ (שכן הג'וב ככל הנראה יקר ויש הפרש גדול דיה בין זמן היעד לזמן העיבוד).

שלב 1: בחר את הגוב עם יחס $\frac{ej}{tj/dj}$ הגדול ביותר. בחן האם ניתן להוסיפו לקבוצה A כך שכל המשימות בקבוצה זו יסתיימו לפני תאריך היעד כאשר הן מסודרות בסדר EDD .

אם כן, צרף אותו לA . אחרת, צרף אותו לA גג. מחק גוב זה מ J.

שלב 2: אם J קבוצה ריקה, עצור. אחרת, חזור לצעד 1.

סיבוכיות זמן החישוב:

סיבוכיות החישוב של אלגוריתם זה יהיה $O(n^2)$ –

תחילה נחשב את הדרוש עבור כל אחד מהג'ובים $O(1)$

בכל איטרציה נעבור על כל הג'ובים ונחפש את הג'וב עם היחס המוזכר מעלה הגדול ביותר $O(n)$

לאחר שמצאנו, ננסה להכניס את ג'וב זה לA לפי סדר EDD - $O(n)$.

בדיקת תקינות ההכנסה תהיה $O(1)$, וכן גם העברת הג'וב אל A גג.