אלגוריתמים מקוונים – Online Algorithms

בבעיות שפגשנו עד עתה, היה נתון קלט ורצינו לפתור בעיות על הקלט הנתון. בבעיות מקוונות, לעומת זאת, הקלט אינו נתון מראש, אלא ניתן במהלך הריצה של האלגוריתם ובכל רגע נתון צריך לבצע חישוב או החלטה.
דוגמאות:

1. דפדוף (paging): מערכת ההפעלה צריכה להחליט אילו דפים היא תביא מהזיכרון האיטי (הדיסק) לזיכרון המהיר (המטמון, או ה cache), מבלי לדעת אילו דפים יידרשו בעתיד. באופן כללי יותר – מערכת ההפעלה צריכה להחליט על הקצאת משאבים למשימות שונות מבלי לדעת אילו משימות היא תקבל בעתיד.
2. ניתוב ברשת: החלטות איך להעביר הודעות ברשת, כאשר העומס על הרכיבים השונים של הרשת עתידים להשתנות.
3. ניווט רובוט: עליו למצוא את הדרך לצאת מהמבוך, ולקבל החלטות על סמך התמונה החלקית שיש ברשותו ברגע נתון.
4. בעיית החתונה: בחור מסוים מחפש כלה. הוא בוחן מועמדות שונות ולכל אחת יש ציון לפי מידת ההתאמה שלה להיות בחירת ליבו. צריך לקבל החלטה על בחירת בת הזוג מבלי לדעת האם בעתיד תגיע מועמדת בעלת ציון התאמה גבוה יותר.

ננסח את הבעיה בצורה פורמאלית:

נתונה קבוצות בקשות אפשריות R, ומקבלים סדרת בקשות/אירועים: לכל $1 \le i$ $\sigma_i \in R$.

בשלב ה i יש בידינו סדרה: $\sigma^{(i)} = \sigma_1\sigma_2\ldots\sigma_i$. בשלב ה 0: סדרה ריקה $= \sigma^{(0)}$. ישנה קבוצת תשובות/פעולות אפשריות A. בשלב ה i על האלגוריתם לתת תשובה/לעשות פעולה $a_i \in A$ כך שהפלט בשלב ה i הוא: $a^{(i)} = a_1a_2\ldots a_i$. התשובות ניתנות ע"י פונקציות $f_i : R^i \to A$, כלומר – בשלב ה i: $f_i(\sigma^{(i)}) = a_i$.

בכל שלב – מקבלים בקשה ושולחים תשובה כתלות בקלט שהיה עד עכשיו. נשים לב שהגדרת f מגדירה את הבעיה כמקוונת, כי התשובה תלויה בבקשות שהגיעו עד לשלב ה i ואינה יכולה להתחשב בבקשות שיגיעו (אולי) בעתיד.

נניח כעת שמדובר בבעיית מינימיזציה: מוגדר מחיר לכל סדרת בקשות ותשובות, ומחיר האלגוריתם בשלב ה i הוא $ALG(\sigma^{(i)}) = \mathrm{cost}_i(\sigma^{(i)}, a^{(i)})$. מטרתנו היא להביא למינימום את המחיר של ALG.

על מנת להעריך את ביצועי האלגוריתם ALG (או ON) נרצה להשוות אותם לאלגוריתם האופטימלי OPT. מסיבות פרקטיות אנחנו נדבר על אלגוריתם אופטימלי לא מקוון (offline). גם ל OPT יש סדרת תשובות, $b^{(i)}$, והמחיר שלו הוא $OPT(\sigma^{(i)}) = \mathrm{cost}_i(\sigma^{(i)}, b^{(i)})$. כעת אפשר להשוות בין ביצועי האלגוריתם המקוון לאלגוריתם האופטימלי על ידי הגדרת <u>**יחס תחרותיות חזקה (competitive ratio):**</u>

בהינתן אלגו' מקוון ON (מסומן גם כ ALG) יחס התחרותיות החזקה הוא α אם מתקיים $ON(\sigma) \le \alpha \cdot OPT(\sigma)$.

נעסוק בשתי בעיות מקוונות – בעיית הסקי ובעיית תזמון עומסים.

בעיית הסקי:

תיאור הבעיה: יוצאים לחופשת סקי, ולא יודעים כמה זמן נישאר עד שיימאס לנו/ניאלץ להפסיק. השאלה היא האם עדיף לשכור ציוד, תמורת תשלום יומי או לקנות ציוד בהשקעה חד פעמית.
הגדרת הבעיה:

- מחיר השכרת ציוד סקי ליום היא 1.
- מחיר קניית ציוד הוא $B > 1$.

- חופשת הסקי תארך n ימים.

נגדיר את הקלט להיות:

- $\sigma_i = 1$, ממשיכים לעשות סקי.
- $\sigma_i = 0$, נשברה הרגל ונוסעים הביתה.

סדרת הקלט תהיה: $\underbrace{111\ldots1}_{n}0$ (חופשת הסקי אורכת n לא ידוע של ימים ומסתיימת ביום ה $n+1$).

קל לראות שהפיתרון האופטימלי הוא $OPT(\sigma) = \min\{B, n\}$.

נסתכל על שני אלגוריתמים מקוונים: (1): ON_S, ו (2): ON_B:

(1):האלגוריתם הראשון - עבור $S \geq 1$ נגדיר: ON_S האלגוריתם שבו שוכרים ציוד, עד היום ה S וביום ה S קונים. עלות האלגוריתם היא:

$$ON_S(\sigma) = \begin{cases} n & S > n \\ S - 1 + B & S \leq n \end{cases}$$

(2): האלגוריתם השני:

$$ON_B(\sigma) = \begin{cases} n & B > n \\ 2B - 1 & B \leq n \end{cases}$$

<u>משפט 1</u>: האלגוריתם ON_B הוא $2 - \frac{1}{B}$ תחרותי חזק.

<u>משפט 2</u>: $2 - \frac{1}{B}$ הוא החסם התחתון על יחס התחרותיות של בעיית הסקי.

<u>הוכחת משפט 1</u>: ישנם שני מקרים, נראה שיחס התחרותיות מתקיים בשניהם.

1. $B > n \quad \Leftarrow \quad \frac{ON_B(\sigma)}{OPT(\sigma)} = \frac{n}{n} = 1$
2. $B \leq n \quad \Leftarrow \quad \frac{ON_B(\sigma)}{OPT(\sigma)} = \frac{2B-1}{B} = 2 - \frac{1}{B}$

מש"ל.

<u>הוכחת משפט 2</u>: צ"ל לכל $S \geq 1$ קיימת סדרה σ כך ש $\frac{ON_B(\sigma)}{OPT(\sigma)} \geq 2 - \frac{1}{B}$.

נבנה את הסדרה על ידי כך שנקבע לכל $S \geq 1$ ש- $n = S$, ואז:

$$\frac{ON_S(\sigma)}{OPT(\sigma)} = \frac{S-1+B}{\min\{S,B\}} = \frac{S}{\min\{S,B\}} + \frac{B-1}{\min\{S,B\}} \geq \frac{S}{S} + \frac{B-1}{B} = 2 - \frac{1}{B}$$

מש"ל.

עד כה עסקנו באלגוריתם מקוון דטרמיניסטי אך באלגוריתם מקוון רנדומי ניתן לשפר את הביצועים ולהתגבר על החסם של $2 - \frac{1}{B}$. נגדיר יחס תחרותיות עבור אלגוריתם מקוון רנדומי:

ON הוא α - תחרותי חזק אם $E[ON(\sigma)] \leq \alpha OPT(\sigma)$.

דוגמה – בעיית הסקי:

נגדיר אלגוריתם ON רנדומי: נגריל בהסתברות θ קנייה ביום הראשון, אחרת נפעיל את ON_B.

תוחלת מחיר האלגוריתם: $E[ON(\sigma)] = \theta B + (1-\theta)ON_B(\sigma)$. קיימים שני מקרים:

א. $n < B$:
$$\frac{E[ON(\sigma)]}{OPT(\sigma)} = \frac{\theta B}{n} + (1-\theta) \leq \theta B + (1-\theta)$$

ב. $n \geq B$:
$$\frac{E[ON(\sigma)]}{OPT(\sigma)} = \theta + (1-\theta)(2 - \frac{1}{B}) = -\theta + 2 - \frac{1}{B} + \frac{\theta}{B}$$

נשווה את הביטויים ב-א- וב-ב-:

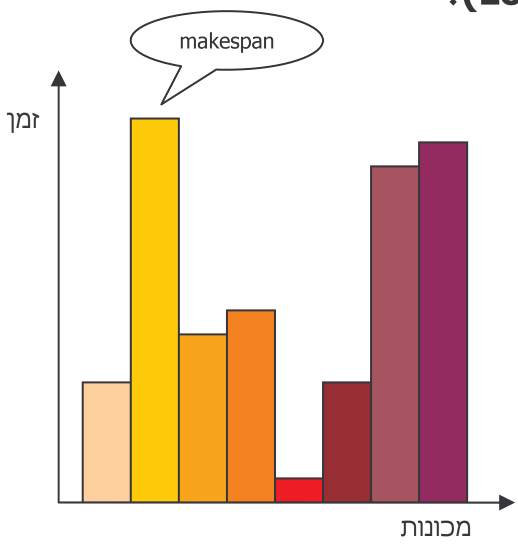
$$\theta(B-1)=(1-\theta)(\frac{B-1}{B})$$

$$B(\theta)=(1-\theta)$$

$$\theta=\frac{1}{B+1}$$

התוחלת המתקבלת היא $2-\frac{2}{B+1}$, כך ששיפרנו את החסם של האלגוריתם המקוון הדטרמיניסטי.

בעיית תזמון עומסים (Load Balancing):



נתונות m מכונות חישוב זהות, וסדרת תהליכים (Jobs בלע"ז). התהליך ה i מעמיס את המכונה שעליה ירוץ ב $a_i>0$. צריך למצוא השמה של תהליך i למכונה j. המטרה היא למזער את העומס המקסימלי על מכונה (makespan). כלומר, נרצה להביא למינימום את העומס על המכונה העמוסה ביותר.

נגדיר l_j^i העומס של המכונה j בזמן i. בסימון זה $makespan=\max_j l_j^i$ בזמן i.

<u>משפט I</u>: עבור $m=2$ יש חסם תחתון של $\frac{3}{2}$ על יחס התחרותיות החזק.

<u>משפט</u>: האלגוריתם החמדן $2-\frac{1}{m}$ תחרותי חזק.

<u>הוכחה למשפט II</u>: באינדוקציה על אורך הסדרה n.

בסיס: עבור $n=0$, $OPT=ON=0$.

הנחת האינדוקציה: עבור $n-1$, $ON(\sigma^{(n-1)})\le\left(2-\frac{1}{m}\right)OPT(\sigma^{(n-1)})$. נוכיח עבור n.

<u>מקרה א</u>: $ON(\sigma^{(n)})=ON(\sigma^{(n-1)})$, אזי הטענה נובעת ישירות מהנחת האינדוקציה כיוון ש $OPT(\sigma^{(n-1)})\le OPT(\sigma^{(n)})$.

<u>מקרה ב</u>: $ON(\sigma^{(n)})=\min_j l_j^{n-1}+a_n$ (למעשה זהו אלגוריתם חמדני).

נשים לב: $*\quad \frac{1}{m}\sum_{i=1}^{n}a_i=\frac{1}{m}\sum_j l_j^n\le OPT(\sigma^{(n)})=\max_j l_j^n$ ובנוסף OPT(σ$^{(n)}$)≥a$_n$.

באותו האופן, $\min_j l_j^{n-1}\le\frac{1}{m}\sum_j l_j^{n-1}=\frac{1}{m}\sum_{i=1}^{n-1}a_i$.

ולכן: $ON(\sigma^{(n)})=\min_j l_j^{n-1}+a_n$

$$ON(\sigma^{(n)})\le\frac{1}{m}\sum_{i=1}^{n-1}a_i+a_n=$$

$$=\frac{1}{m}\sum_{i=1}^{n-1}a_i+\frac{1}{m}a_n+(1-\frac{1}{m})a_n=\frac{1}{m}\sum_{i=1}^{n}a_i+(1-\frac{1}{m})a_n\overset{*}{\le}OPT(\sigma^{(n)})+(1-\frac{1}{m})OPT(\sigma^{(n)})=(2-\frac{1}{m})OPT(\sigma^{(n)})$$

מש"ל