תכנון דינמי

29 באוקטובר 2002

כרגיל, לא קל לנסח את העיקרון באופן מילולי ומלא. להלן כמה מהמאפיינים העיקריים של אלגוריתמים המבוססים על תכנון דינמי:

1. אלו בעיות אופטימיזציה שיש בהן מושג ברור של תת-בעיה.
2. פתרון של הבעייה הכוללת משרה גם פתרונות לתת הבעיות.
3. הפתרונות המושרים מפתרון כולל אופטימלי, הם פתרונות אופטימליים של התת בעיות.
4. בונים את הפתרון האופטימלי הכולל "מן הקטן אל הגדול". כלומר פותרים באופן אופטימלי את הבעיות הקטנות ומוצאים איך לשלב את הפתרונות החלקיים יחד לפתרון בעיות גדולות יותר עד לפתרון הבעייה כולה.

הדוגמה הראשונה שבה נטפל באה מביולוגיה מולקולרית חישובית. זהו האלגוריתם של Smith-Waterman להשוואת רצפים. כידוע מקודדות מולקולות ה-DNA מידע ביולוגי חיוני. בגלל התהליך האבולוציוני שבו התפתחו רצפי ה-DNA ניתן לגלות רצפים שונים עם דמיון רב ביניהם. רצפים כאלה מופיעים הן באורגניזמים שונים והן במקומות שונים ברצף ה-DNA של אותו אורגניזם. הוא הדין גם ברצפי חלבון. על מולקולת DNA ניתן לחשוב כ"מילה" ארוכה מאוד בא"ב של ארבע אותיות. מולקולות חלבון הן "מילים" בנות כמה עשרות - כמה אלפי אותיות בא"ב של 20 אותיות (ל"אותיות" האלה קוראים חומצות אמינו). איך נשווה את מידת הדמיון של שתי מילים כאלה? תהיינה $x_1, \ldots, x_m$, $y_1, \ldots, y_n$ שתי מילים כאלה (כשמדובר בחלבונים, כ"א מה-x_i וה-y_j הם איברים בקב' Σ בת 20 איברים). מחפשים התאמה מיטבית (אופטימלית) בין איברי הסידרה הראשונה והשנייה. אנו ניתן "ציון" לכל התאמה ונחפש את זו בעלת הציון הגבוה ביותר. נביט למשל בהתאמה:

$$\begin{array}{ccccccc} x_1, & x_2, & x_5. & x_6, & x_7, & x_8, & x_9 \\ | & | & | & | & | & | & | \\ y_1, & y_2, & y_3, & y_4, & y_5, & y_{10}, & y_{11} \end{array}$$

איזה ציון ניתן לה? יש כאן שלושה סוגי תופעות:

1. התאמה של אות לאות, למשל x_7 ו-y_5.
2. דילוג בסידרה x: שימו לב למשל שהאיברים x_3, x_4 לא הותאמו לאף y_i.
3. דילוג בסידרה y, למשל y_6, y_7, y_8, y_9 לא הותאמו לשום x.

אנו מעירים גם שההתאמה מונוטונית וחד-חד-ערכית וזהו חלק מהגדרת הבעייה. כלומר אם x_i מותאם ל- y_j, ו- x_k ל- y_l ואם $k > i$ אז בהכרח $l > j$ ולהיפך.

הציון להתאמה הנ"ל בנוי מציונים לצעדים האינדיבידואליים. נניח ש- $x_i = \sigma$ ו- $y_j = \tau$ הותאמו. יש מטריצה ממשית סימטרית $A_{\Sigma \times \Sigma}$ והציון על ההתאמה הנ"ל הוא $a_{\sigma,\tau}$. על פתיחת פער של t מקומות בסידרה x אנו משלמים קנס של $t\Delta$ כש- Δ קבוע מתאים. כיו"ב על פתיחת פער ב- y. כך למשל הציון על ההתאמה הנ"ל יהיה:

$$a_{x_1,y_1} + a_{x_2,y_2} - 2\Delta + \dots$$

(הביטוי האחרון מתאים לכך שפסחנו על x_3, x_4) יש עדיין צורך בהסבר מדוע נבחרה דווקא השיטה הזו להערכת הדמיון בין רצפים, אך את זאת נשאיר לקורסים בביואינפורמטיקה. נסכם אם כן: הבעיה מוגדרת ע"י א"ב סופי, מטריצה ממשית סימטרית A כנ"ל ופרמטר Δ. קלט לבעייה ניתן ע"י שתי מילים סופיות $x_1, \dots, x_m$, $y_1, \dots, y_n$ בא"ב Σ. בהינתן שתי מילים כאלה, עלינו למצוא את ההתאמה בעלת הציון הגבוה ביותר.

נשים לב שמספר ההתאמות האפשריות בין סידרה מאורך m לסידרה מאורך n כאשר m ו- n גדולים ושווים בערך, הוא מעריכי ב- n, לכן מעבר ממצה על כל ההתאמות האפשריות אינו בא בחשבון. נדרשת כאן שיטה יעילה יותר. תיאור קומבינטורי של הבעייה יעזור לנו בפתרונה: נתבונן במהלך על נקודות השריג המישורי. המהלך יוצא מהראשית $(0,0)$ ומגיע ל-(m,n) כל צעד הוא אחד מבין האפשרויות הבאות: א. $(1,1)$ - אם הצעד הוא מ-$(i-1,j-1)$ ל-(i,j) משמע שאנו מתאימים את x_i ל- y_j. ב. $(1,0)$ - דילוג על ה-x הנוכחי. ג.$(0,1)$ - דילוג על ה-y הנוכחי. קל להתאים לכל מהלך כזה מחיר ע"פ הכללים שניסחנו. הבעייה היא למצוא מהלך במחיר מזערי.

הרעיון לפתרון: כאן בדיוק אנו משתמשים בעיקרון הבסיסי של תכנון דינמי. במ-קום לחשב רק את המחיר של ההתאמה הכוללת הטובה ביותר, אנו נחשב mn ערכי אופטימום. נגדיר: $c_{k,l}$ זהו המחיר האופטימלי של התאמה בין הרישא מאורך k של הסידרה x והרישא מאורך l של הסידרה y. המטרה היא, אם כן לחשב את המספר $c_{m,n}$ ונעשה זאת ע"י חישוב כל mn המספרים $c_{k,l}$. בדיון שלהלן נדבר לחילופין על התאמות אופטימליות ועל מסילות שריגיות אופטימליות. ננסה לחשב את $c_{k,l}$. במ-סילה אופטימלית מהראשית ל- (k,l) נביט מהו הצעד האחרון. כמובן זהו אחד משלושה הצעדים המותרים לנו: $(0,1),(1,0),(1,1)$. לכן נוכל להסיק את הביטוי הבא

$$c_{k,l} = max\{c_{k-1,l-1} + a_{x_k y_l} \ ; \ c_{k,l-1} - \Delta \ ; \ c_{k-1,l} - \Delta\}$$

יוצא שאם המספרים $c_{k-1,l-1}; c_{k-1,l}; c_{k,l-1}$ ידועים לנו, אנו מסוגלים לחשב את $c_{k,l}$. נחשב אם כן את כל המספרים $c_{i,j}$ ע"פ סדר עולה של $i+j$ עד שנמצא את $c_{m,n}$ מספר הצעדים הנדרש הוא $O(mn)$.

שרשרת של כפלי מטריצות כזכור, כפל מטריצות הוא אסוציאטיבי, כלומר $(AB)C = A(BC)$ נניח שעלינו לחשב שרשרת של כפלי מטריצות $A_1, A_2, ..., A_n$. בגלל החוק האסוציאטיבי נקבל את התוצאה הנכונה, יהיה הסדר אשר בו נבצע את הכפלים אשר יהיה. יחד עם זה, לסדרים שונים יש מחיר חישובי שונה. ברצוננו למצוא את סדר הפעולות החסכוני ביותר מבחינת מספר הפעולות. נתבונן בפעולת הכפל AB של שתי מטריצות $A_{p \times q}$ ו $B_{q \times r}$. כאשר כופלים אותן "לפי הספר" מבצעים pqr פעולות כפל במספרים ממשיים. (נזכיר שיש אלגוריתמים מתוחכמים יותר ויעילים יותר למשל

אלגוריתם הכפל של Strassen הכופל שתי מטריצות $n \times n$ ב n^{α} ב פעולות כאשר $\alpha = \log_2 7 < 3$, וגם שיטות משוכללות עוד יותר. אנו מתעלמים מכל זה כאן וחוזרים לכפל מטריצות כפי שלומדים בקורס באלגברה ליניארית). קל להשתכנע שבסדרים שונים של הכפלות נשלם מחיר שונה. איך נמצא את הסדר המיטבי? שוב, כמו בדוג-מאות אחרות של תכנון דינמי נרצה לפצל את הבעייה לתת בעיות מאותו הטיפוס. נניח שבפתרון האופטימלי, הפעולה האחרונה מתבצעת "במקום ה i". משמע שבטרם הפעולה האחרונה כבר חישבנו את $P = A_1A_2...A_i$, וכן את $Q = A_{i+1}A_{i+2}...A_n$, ועתה אנו מכפילים את PQ ומקבלים את התשובה הסופית. ברור שאת P ואת Q כדאי לנו לחשב גם כן באופן אופטימלי ("התת בעיות הרלוונטיות גם כן נפתרון באופן אופטימלי"). זה מוביל אותנו למסקנה הבאה: שוב כרגיל לא נחפש רק את האופטימום לבעייה דומה. לכל $1 \le i \le j \le n$ נגדיר את $c_{i,j}$ כמחיר המזערי לחישוב המטריצה $A_iA_{i+1}...A_j$, ונחשב את כל המספרים האלה כשאנו עולים בגודל $j-1$. אם המטריצה A_i היא מגודל $\alpha_i \times \alpha_{i+1}$ אז כמובן

$$c_{i,i+1} = \alpha_i \cdot \alpha_{i+1} \cdot \alpha_{i+2}$$

. על מנת לחשב את $c_{i,j}$ כאשר $j - i > 2$ נשאל את עצמנו מהו המקום t שבו נעשתה פעולת הכפל האחרונה בחישוב אופטימלי של $A_iA_{i+1}...A_j$. תתקבל הנוסחה הבאה:

$$c_{i,j} = min_{i<t<j}\{c_{i,t} + c_{t+1,j} + \alpha_i \cdot \alpha_{t+1} \cdot \alpha_{j+1}\}$$

מכיון ש $t-i, j-t-1 < j-i$, כל הביטויים המופיעים בנוסחה זו כבר ידועות לנו (כזכור אנו מחשבים את המספרים $c_{i,j}$ ע"פ סדר עולה של $j-1$. תוך $O(n^3)$ פעולות חישוב נמצא גם את $c_{1,n}$, שזו התשובה לשאלתנו.