الگوریتم پیشرفته

۲۴ آبان

دکتر حامد فهیمی

دانشگاه فردوسی مشهد

دانشکده علوم ریاضی

تعریف مسئله الگوریتمی و الگو

یک مسئله الگوریتمی یک سوال کلی است با پارامترهایی برای ورودی و خروجی و شرایطی که یک راه‌حل یا پاسخ رضایت‌بخش را ارائه می‌دهند.

یک **الگو (instance)** مساله‌ای است که پارامترهای ورودی آن مشخص شده باشند.

مسئله فروشنده دوره‌گرد (TSP)

ورودی: یک گراف G

خروجی: کدام دور $(v_1, \ldots, v_n)$ مقدار

$$\sum_{i=1}^{n-1} d(v_i, v_{i+1}) + d(v_n, v_1)$$

را کمینه می‌کند؟

> هر گراف وزن‌دار یک الگو از TSP را تعریف می‌کنند.

کاهش مسائل به یکدیگر

کاهش‌ها الگوریتم‌هایی هستند که یک مسئله را به مسئله دیگری تبدیل می‌کنند.

ایده کلیدی کاهش به صورت زیر است:

دو مسئله الگوریتمی A و B را در نظر می‌گیریم. فرض کنید که الگوریتم کاهش زیر برای حل مسئله A به شما داده شده باشد:

A(G)

ورودی G را از مسئله A به یک الگوی Y از مسئله B تبدیل کنید.

سپس زیرروال الگوریتم B را برای حل الگوی Y فراخوانی کنید.
پاسخ به‌دست‌آمده برای مسئله B را به عنوان پاسخ A(G) برگردانید.

این الگوریتم به درستی مسئله A را حل خواهد کرد مشروط بر اینکه کاهش به B همیشه صحت پاسخ را حفظ کند؛ به عبارت دیگر مشروط بر اینکه ترجمه دارای این ویژگی باشد که برای هر الگوی G،

$$A(G) = B(Y)$$

اساساً این کاهش نشان می‌دهد که مسئله B آسان‌تر از مسئله A نیست. بنابراین اگر A سخت باشد، B نیز بایستی به همان اندازه یا حداقل سخت باشد.

$$A \leq_p B$$

کاهش‌ها بین مسائل ترجمه می‌شوند. شکل خروجی مورد انتظار از مسئله‌ای به مسئله دیگر ممکن است متفاوت باشد. مثلاً مسئله TSP، یک قابلیت از ردهس را به عنوان پاسخ برمی‌گرداند، در حالی که نوعی مسائل دیگر ممکن است مثلاً رشته‌های عدد صحیح را به عنوان پاسخ برگردانند.

ملاحظات زمانی کاهش

فرض کنید که کاهش الگوی G را در زمان O(P(n)) به Y تبدیل کند. در این صورت:

- اگر زیرروال B در بازه زمانی O(P'(n)) اجرا شود، این با ترجمه مسئله و سپس اجرای زیرروال B برای حل آن، الگوریتمی از زمان

$$O(P(n) + P'(n))$$

برای مسئله A در بر خواهد داشت.
- اگر بدانیم که Ω(P'(n)) یک کران پایین برای محاسبه A است، به این معنا که قطعاً الگوریتمی با کارایی بهتر برای حل آن وجود ندارد، آنگاه

$$\Omega(P'(n) - P(n))$$

یک کران پایین برای محاسبه B است. چرا؟

چالش در کاهش

یک چالش اساسی در کاهش، پیدا کردن مسئله‌ای است که می‌توان مسئله مدنظر را به آن کاهش داد. علاوه بر این، اگر می‌خواهیم تلاش‌های ما ارزش عملی داشته باشند، باید الگوریتم مبتنی بر کاهش کارآمدتر از حل مستقیم مسئله اصلی باشد.

مسائل تصمیم‌گیری (Decision Problems)

تعریف: به دسته‌ای از مسائل که پاسخ آن‌ها تنها به صورت بله/خیر (یا درست/نادرست) است، **مسائل تصمیم‌گیری** گفته می‌شود.

اهمیت: اکثر مسائل پیچیده «بهینه‌سازی» را می‌توان به مسائل «تصمیم‌گیری» تبدیل کرد.

مثال: مسئله فروشنده دوره‌گرد (TSP):

- **حالت بهینه‌سازی:** پیدا کردن کوتاه‌ترین مسیر ممکن.
- **حالت تصمیم‌گیری:** آیا در گراف $G = (V, E)$، دوری (Cycle) با هزینه حداکثر k وجود دارد؟

نحوه رسیدن به جواب بهینه: با پرسیدن متوالی سوالات تصمیم‌گیری برای مقادیر مختلف k و دریافت پاسخ‌های No و Yes، می‌توان به مقدار بهینه دست یافت.

ترجمه TSP به مسئله تصمیم‌گیری

مسئله کمینه‌سازی TSP را می‌توان به شکل زیر در قالب یک مسئله تصمیم‌گیری ترجمه کرد:

ورودی: یک گراف وزن‌دار G و عدد صحیح k.
خروجی: آیا دوری با هزینه حداکثر k وجود دارد؟

این نسخه تصمیم‌گیری از مسئله TSP، خودِ مسئله بهینه‌سازی را نیز در بر می‌گیرد. (چرا؟)

با توجه به این مطلب، از این به بعد خود را به مسائل تصمیم‌گیری محدود خواهیم کرد.

مثال ۱: مسئله نزدیک‌ترین جفت

مسئله نزدیک‌ترین جفت ناظر به پیدا کردن دو عدد با کمترین تفاضل در میان مجموعه‌ای از اعداد است. مثلاً در

$$\{10, 4, 8, 3, 12\}$$

نزدیک‌ترین جفت (3, 4) است.

ابتدا این مسئله را به صورت زیر تبدیل به یک مسئله تصمیم‌گیری می‌کنیم:

ورودی: مجموعه‌ای S از اعداد و آستانه‌ی k.
خروجی: آیا جفت (S_i, S_j)ای وجود دارد که $|S_i - S_j| \le k$؟

راهکار:
برای حل مسئله نزدیک‌ترین جفت، کافی است که اعداد را مرتب‌سازی کنیم. اکنون به‌دست آوردن پاسخ مسئله اصلی معادل است با چک کردن تمامی $|S_{i+1} - S_i|$ و پیدا کردن مینیمم آن‌ها.

مثال ۲: کاهش مسئله مینیمم به ماکسیمم

اگر الگوریتمی برای یافتن بیشینه (Maximum) یک تابع در اختیار داشته باشیم، می‌توانیم با استفاده از رابطه

$$\min f(x) = -\max[-f(x)]$$

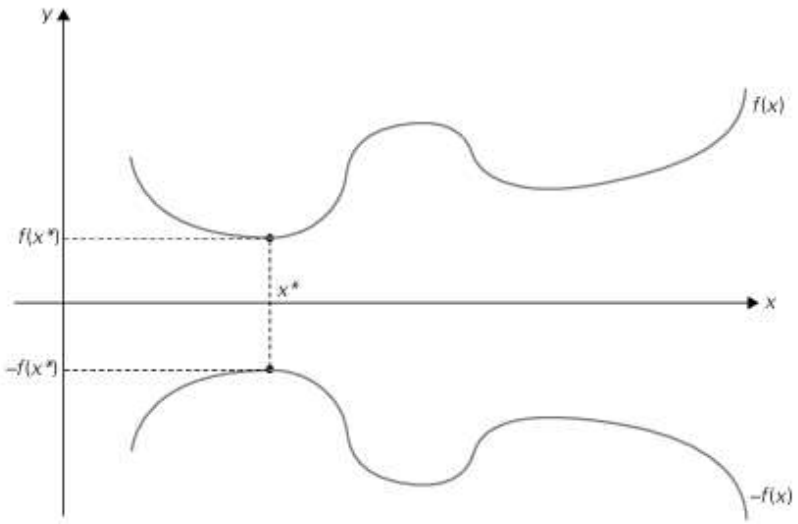
مسئله کمینه‌سازی (Minimization) را حل کنیم.

این بدان معناست که برای یافتن کمینه یک تابع، کافی است قرینه آن تابع را بیشینه‌سازی کرده و در نهایت علامت نتیجه را تغییر دهیم. این اصل عمومی برای توابع با هر دامنه و هر تعداد متغیر، حتی در حضور محدودیت‌ها (Constraints)، صادق است.

علاوه بر این، روش‌های استاندارد حساب دیفرانسیل برای یافتن نقاط اکسترمم نیز نوعی «کاهش مسئله» محسوب می‌شوند. در این فرایند، مسئله بهینه‌سازی به مسئله حل معادله

$$f'(x) = 0$$

تبدیل می‌شود تا نقاط بحرانی به‌دست آیند. اگرچه در متون درسی مسائل به گونه‌ای انتخاب می‌شوند که این معادلات به سادگی حل شوند، اما باید توجه داشت که در دنیای واقعی، حل این معادلات همیشه ساده نیست و این «کاهش»، تنها بخشی از مسیر رسیدن به پاسخ نهایی است.



مثال ۳: گراف و ماتریس مجاورت

در این بخش، یک گراف با ۴ رأس {a, b, c, d} و ماتریس مجاورت آن بررسی می‌شود.

۱. شکل گراف

گراف دارای یال‌های زیر است:

- (a, b)
- (a, c)
- (a, d)
- (c, d)

۲. ماتریس مجاورت (A)

	a	b	c	d
a	0	1	1	1
b	1	0	0	0
c	1	0	0	1
d	1	0	1	0

۳. توان دوم ماتریس (A^2)

ماتریس A^2 تعداد مسیرهای به طول ۲ بین رئوس را نشان می‌دهد:

$$A^2 = \begin{pmatrix} \mathbf{3} & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}$$

نکته: عدد ۳ (که در تصویر دور آن خط کشیده شده) نشان‌دهنده تعداد کل مسیرهای به طول ۲ است که از رأس a شروع شده و به خود a ختم می‌شوند.

لیست مسیرهای به طول ۲ از a به a:

- a → b → a
- a → c → a
- a → d → a