

پیوست آ

روش‌های بهینه‌سازی دقیق

۱.آ معرفی

مسائل برنامه‌ریزی و زمان‌بندی به‌طور ذاتی ساده‌اند و به‌صورت برنامه‌ریزی خطی که به‌آسانی قابل حل در الگوریتم‌های موجود کارا هستند قابل فرمول‌بندی می‌باشند. این الگوریتم‌های کارا معمولاً به‌عنوان الگوریتم‌های چندجمله‌ای زمان معرفی می‌شوند. این مسئله می‌تواند توسط یک الگوریتم کارا و زمان چندجمله‌ای حل شود، که مثال‌های خیلی بزرگ از آن مسئله با صدها یا هزاران فعالیت می‌توانند در یک مدت زمان نسبتاً کوتاه در یک کامپیوتر حل شود.

از این رو، بسیاری از مسائل برنامه‌ریزی وجود دارند که ذاتاً بسیار سخت می‌باشند. این مسائل به‌عنوان مسائل $NP - hard$ معرفی می‌شوند. این مسائل معمولاً مسایل ترکیبی هستند که نمی‌توانند با مسائل خطی حل شوند و هیچ قانون یا الگوریتم ساده‌ای که جواب‌های بهینه در مدت زمان محدود کامپیوتر به‌دست آورد وجود ندارد. اغلب ممکن است که برنامه عدد صحیح یا برنامه ریزی گسسته مناسبی برای چنین مسائلی وجود داشته باشد، اما حل اینگونه مسائل ممکن است به یک زمان بسیار زیاد کامپیوتری نیاز داشته باشد.

دسته‌های متنوع و مفیدی از روش‌ها برای به‌دست آوردن جواب‌های بهینه برای اینچنین مسائل $NP - hard$ وجود دارد. یکی از این دسته‌ها، روش‌های برنامه‌ریزی

پویا را معرفی می‌کند. برنامه‌ریزی پویا یکی از گسترده‌ترین تکنیک‌های مورد استفاده برای پرداختن به مسائل بهینه‌سازی ترکیبی است. برنامه‌ریزی پویا روشی است که برپایه‌ی رویکرد تقسیم و موفقیت می‌باشد. برنامه‌ریزی پویا برای مسائلی که قابل حل در زمان چندجمله‌ای هستند به خوبی مسائل $NP - hard$ می‌توانند به کار برده شوند. اگر یک مسئله برنامه‌ریزی بتواند توسط یک برنامه‌ریزی عدد صحیح فرموله شود، آنگاه از تکنیک‌های مختلف دیگری نیز قابل اجرا است. شناخته شده‌ترین روش‌ها برای حل برنامه‌ریزی عدد صحیح، عبارت‌اند از:

۱. روش‌های شاخه و کران

۲. روش‌های چندسطحی برش

۳. روش‌های ترکیبی

اولین دسته از روش‌ها، شاخه و کران است که یکی از محبوب‌ترین دسته‌ها از تکنیک‌های مورد استفاده برای برنامه‌ریزی عدد صحیح است. شاخه کردن یک تقسیم‌بندی از فضای جواب ارائه می‌دهد و هر بخش از فضای جواب به طور مجزا در نظر گرفته می‌شود. محدود کردن (کرانی کردن) توسعه کران‌ها را برای بخش‌هایی از فضای جواب معرفی می‌کند (با فرض مینیمم کردن هدف). اگر کران پایینی یافت شود که در یک بخش از فضای جواب بزرگ‌تر از جواب صحیحی باشد که قبلاً در بخش دیگری از فضای جواب به دست آمده است، آنگاه بخش مربوط به کران بزرگتر می‌تواند نادیده گرفته شود.

دومین دسته از روش‌ها، روش‌های برنامه‌ریزی برش و تمرکز بر آزادسازی برنامه‌ریزی خطی در برنامه‌ریزی عدد صحیح است. این روش‌ها محدودیت‌های خطی اضافی را تولید می‌کنند، یعنی برنامه‌ریزی برش برای متغیرهایی که بایستی صحیح شوند اعمال می‌شود. تعداد نامعادلات محدودیت‌های اضافی مجموعه‌ی شدنی بیشتر از نامعادلات مجموعه‌های اصلی خطی بدون برش در جواب‌های صحیح است. حل کردن آزادسازی LP از IP با محدودیت‌های اضافی یک جواب جدید را به دست می‌دهد که ممکن است صحیح باشد. اگر جواب صحیح باشد، روش متوقف می‌شود و جواب به دست آمده به عنوان جواب نهایی برای مسئله اصلی IP است. اگر متغیرها

عدد صحیح نباشند معادلات بیشتری ایجاد می‌شود. روش‌های ترکیبی معمولاً ترکیب ایده‌هایی از روش‌های متنوع و مختلف است. به عنوان مثال روش برنامه‌ریزی صفحه برش در سال‌های اخیر در میان استفاده کنندگان از این روش در ترکیب با شاخه و کران محبوب‌اند. وقتی روش شاخه و کران در تکنیک‌های برنامه‌ریزی برش استفاده می‌شود، آن را به عنوان شاخه و برش معرفی می‌کنند.

۲.آ برنامه‌ریزی پویا

برنامه‌ریزی پویا اساساً یک برنامه شمارشی کامل است که از طریق یک روش تقسیم و موفقیت، برای به حداقل رساندن مقدار مورد محاسبه، تلاش می‌کند. روش حداقل کردن یک سری از زیرمسئله‌ها تا یافتن یک جواب برای مسئله اصلی است. جواب بهینه را برای هر زیرمسئله و سهم آن در تابع هدف را تعیین می‌کند. در هر تکرار جواب بهینه را برای یک زیرمسئله فعلی با استفاده از تمام اطلاعاتی که قبلاً در جواب‌های هم‌ی زیرمسئله‌های قبلی به دست آمده است پیدا می‌کند. برنامه‌ریزی پویا با سه نوع معادله مشخص می‌شود. یعنی

۱. شرایط اولیه

۲. رابطه بازگشتی و

۳. مقدار تابع بهینه

در برنامه‌ریزی پویا برای انتخاب یک گزینه، می‌توان از برنامه‌ریزی پویای پیشرو و برنامه‌ریزی پویای پسرو استفاده نمود. مثال زیر یک برنامه‌ریزی پویای پیشرو را نشان می‌دهد.

مثال ب. ۲.۱ (فرمول برنامه‌ریزی پویای پیشرو). یک ماشین و n فعالیت را در نظر بگیرید. اگر فعالیت j در زمان C_j کامل شود و یک هزینه $h_j(C_j)$ در برداشته باشد.

مسئله مورد بررسی این است که کارها به گونه‌ای ترتیب داده شوند که هدف

$$\sum_{j=1}^n h_j(C_j)$$

حداقل شود. این مسئله، یک مسئله بسیار مهم در نظریه برنامه‌ریزی است که شامل بسیاری از توابع هدف مهم در موارد خاص است، می‌باشد. به عنوان مثال، هدف یک تعمیمی از هدف است، که از این رو یک مسئله $NP - hard$ می‌باشد. فرض کنید j یک زیر مجموعه از n فعالیت باشد و فرض کنید که همه‌ی فعالیت‌هایی که در j هستند قبل از هر یک از فعالیت‌هایی که در مجموعه j نیستند انجام شوند. فرض کنید

$$V(J) = \sum_{j \in J} h_j(C_j)$$

مجموعه‌ای از فعالیت‌های j باشند که در ابتدا انجام شوند. فرمول برنامه‌ریزی پویا در مسئله به شرایط اولیه، مقدار تابع هدف و رابطه بازگشتی بستگی دارد.
شرایط اولیه:

$$V(j) = h_j(p_j) \quad j = 1, 2, \dots, n$$

رابطه بازگشتی:

$$V(J) = \min_{j \in J} \left(V(J - \{j\}) + h_j\left(\sum_{k \in J} p_k\right) \right)$$

مقدار تابع بهینه:

$$V(\{1, \dots, n\})$$

ورای این روش برنامه‌ریزی پویا نسبتاً آسان است. در هر عمل از توالی بهینه برای یک زیرمجموعه از فعالیت‌ها (یک زیر مجموعه j که شامل L فعالیت است) تعیین می‌شود و فرض می‌شود این زیرمجموعه در ابتدا اعمال شود. این روند برای هر زیر مجموعه با اندازه L انجام می‌شود. که $n!/(l!(n-l)!)$ زیرمجموعه می‌باشد. برای هر

زیرمجموعه، سهم L برنامه‌ریزی شده‌ی فعالیت‌ها برای تابع هدف محاسبه می‌شود. از طریق رابطه بازگشتی به همه‌ی زیرمجموعه‌هایی که شامل $L + 1$ فعالیت است، گسترش داده می‌شود. هریک از $L + 1$ فعالیت به عنوان یک کاندیدا که در ابتدا باید انجام شود، در نظر گرفته می‌شود. وقتی که با استفاده از رابطه بازگشتی توالی واقعی L فعالیت از زیرمجموعه‌های کوچکتر مورد توجه قرار نمی‌گیرد، تنها بخشی از فعالیت‌های L در هدف شناخته می‌شود. پس از اینکه ارزش $V(\{1, 2, \dots, n\})$ مشخص شد، توالی بهینه از طریق یک روش پسرو و ساده به دست می‌آید.

$V(J)$ باید برای همه‌ی زیرمجموعه‌هایی که شامل L فعالیت می‌باشد مشخص شود. که به تعداد $n!/(l!(n-l)!)$ زیرمجموعه است. همچنین مجموع اعداد تخمینی که بایستی انجام شود نیز

$$\sum_{l=1}^n \frac{n!}{l!(n-l)!} = O(2^n)$$

می‌باشد.

مثال ب. ۲.۲ (کاربرد برنامه‌ریزی پویای پیشرو). مسئله‌ی مثال قبل را با فعالیت‌های زیر در نظر بگیرید.

فعالیت‌ها	۱	۲	۳
p_j	۴	۳	۶
$h_j(C_j)$	$C_1 + C_2$	$3 + C_2^2$	$8C_3$

همچنین $V(\{1\}) = 20$ و $V(\{2\}) = 30$ و $V(\{3\}) = 48$ هستند. دومین اثر متقابل روش همه مجموعه‌های شامل ۲ فعالیت را در نظر بگیرید. به کار بردن رابطه بازگشتی حاصل می‌دهد:

$$\begin{aligned} V(\{1, 2\}) &= \min \left(V(\{1\}) + h_2(P_1 + P_2), V(\{2\}) + h_1(P_2 + P_1) \right) \\ &= \min(20 + 346, 30 + 56) = 86 \end{aligned}$$

۳.آ روش‌های بهینه‌سازی برای برنامه‌ریزی عدد صحیح

۴.آ مثال‌هایی از کاربردهای شاخه و کران