دانشکده علوم ریاضی
دانشگاه صنعتی شریف

تحقیق در عملیات ۱

محمدهادی فروغمنداعرابی
پاییز ۱۳۹۹

برنامه‌ریزی صحیح: تطابق وزن‌دار کامل در گراف دوبخشی

جلسه‌ی چهارم

نگارنده: مهراد زمانی

در این جلسه با نوع خاصی از مسائل برنامه‌ریزی به نام مسائل «برنامه‌ریزی صحیح»[1] آشنا می‌شویم.[2] در این گونه از مسائل برنامه‌ریزی، به دنبال بهترین جوابی هستیم که در آن مقدار تمامی متغیرها صحیح باشد.

ابتدا به بررسی مثالی می‌پردازیم که فرمول‌بندی آن به شیوه‌ای که تا پیش از این به کار می‌بردیم لزوما جواب مناسبی به ما نمی‌دهد و این، انگیزه‌ای می‌شود برای افزودن نوع جدیدی از قیود به صورت مسئله (که قیود مربوط به صحیح بودن مقادیر متغیرها هستند) و ابداع گونه‌ای جدید از مسائل برنامه‌ریزی که آن‌ها را برنامه‌ریزی صحیح می‌نامیم. سپس نشان می‌دهیم که این گونه از مسائل دشوارتر از مسائل پیشین هستند و برای این که بتوانیم در زمان معقولی آن‌ها را حل کنیم باید ابتدا برایشان یک جواب بهینه که مقادیر متغیرها در آن لزوما صحیح نیست را پیدا کنیم و سپس سعی کنیم که با استفاده از آن، یک جواب صحیح و بهینه برای مسئله پیدا کنیم. در انتها این شیوه‌ی جدید را برای حل یک مسئله‌ی کلاسیک به کار می‌بریم.

۱ مروری بر مباحث پیشین

در جلسات پیش [۳، ۴] با گونه‌ای از مسائل به نام مسائل «برنامه‌ریزی خطی»[3] آشنا شدیم که فرم کلی آن‌ها به صورت زیر بود:

$$\begin{array}{ll} c^T x & \text{کمینه کن/بیشینه کن} \\ Ax \leq b & \text{که} \end{array}$$

[1] integer programming

[2] البته تنها به بررسی حالت خطی آن‌ها می‌پردازیم.

[3] linear programming

که در آن x بردار متغیرهاست که n درایه دارد و c هم یک بردار است. هم‌چنین A یک ماتریس $m \times n$ و b یک بردار است. تمامی درایه‌ها حقیقی هستند. به $c^T x$ «تابع هدف»[۴] گفته می‌شود و $Ax \leq b$ بدان معناست که بردار Ax به صورت درایه‌به‌درایه کوچک‌تر مساوی بردار b باشد. به عبارت دیگر، m نامساوی (قید)[۵] داریم که باید برقرار باشند. دلیل ذکر صفت «خطی» در نام این گونه از مسائل آن است که همان‌طور که مشخص است، تابع هدف و تمامی قیود بر حسب متغیرها خطی هستند.

می‌دانیم که مسائل برنامه‌ریزی خطی در زمان چندجمله‌ای بر حسب تعداد متغیرها قابل حل هستند. بسیاری از مسائل مشهور در ریاضیات گسسته قابل تبدیل به مسائل برنامه‌ریزی خطی هستند و به طور کلی این دسته از مسائل از اهمیت بسیاری برخوردار هستند.

در جلسه‌ی سوم [۴] با چند نمونه از مسائل که در دنیای بیرون کاربرد دارند آشنا شدیم و آن‌ها را به مسائل برنامه‌ریزی خطی تبدیل کردیم و در این حین با چند تکنیک برای تبدیل قیود غیرخطی به قیود خطی آشنا شدیم.

۲ مثال: برش رول کاغذ

در این بخش یک مثال را بررسی می‌کنیم که با مثال‌هایی که در جلسات پیش بررسی کردیم تفاوت خاص و مهمی دارد.

مثال ۱. *یک کارخانه‌ی تولید کاغذ رول‌هایی به طول سه متر تولید می‌کند (عرضشان برای ما اهمیتی ندارد). سفارش‌های مشتری‌ها به این صورت است که تقاضای تعدادی ورقه‌ی کاغذ می‌کنند که ممکن است طول یکسانی نداشته باشند. یکی از سفارشات به این صورت است:*

۱) ۹۷ *ورقه به طول* ۱۳۵ *سانتی‌متر*
۲) ۶۱۰ *ورقه به طول* ۱۰۸ *سانتی‌متر*
۳) ۳۹۵ *ورقه به طول* ۹۳ *سانتی‌متر*
۴) ۲۱۱ *ورقه به طول* ۴۲ *سانتی‌متر*

کارخانه برای صرفه‌جویی در هزینه‌ها تمایل دارد که کم‌ترین تعداد رول سه‌متری ممکن را برای تأمین ورقه‌های مورد نیاز این مشتری برش دهد. این کمینه‌ی تعداد رول‌ها را بیابید (به همراه روشی برای برش دادن آن‌ها که ورقه‌های مورد نیاز را تأمین کند).

حل. نکته‌ی کلیدی در تبدیل این مسئله به یک مسئله‌ی برنامه‌ریزی خطی آن است که به این توجه کنیم که کارخانه برای تأمین ورقه‌های مورد نیاز، رول‌ها را به چه حالت‌هایی می‌تواند برش دهد. در واقع به ازای هر حالت از برش یک رول و تبدیل آن به تعدادی ورقه با طول‌های مناسب، می‌توانیم یک متغیر در نظر بگیریم که تعداد رول‌هایی که به آن روش خاص برش داده می‌شوند را نشان بدهد. تعداد این حالات و در نتیجه تعداد متغیرها بسیار زیاد است. برای این که این مسئله در عمل در سریع‌ترین زمان ممکن حل شود، تمایل داریم که کم‌ترین تعداد متغیر را در مسئله‌ی برنامه‌ریزی خطیمان داشته باشیم. یک مشاهده که در کم کردن تعداد متغیرها می‌تواند به ما کمک کند این است که در نظر گرفتن حالاتی از برش دادن که در آن‌ها با برش دادن کاغذی که دور ریخته می‌شود نمی‌توان ورقه‌ای با طول مناسب تولید کرد (یعنی حالاتی که طول کاغذی که دور ریخته می‌شود کم‌تر از ۴۲ سانتی‌متر است) کفایت می‌کند؛ زیرا اگر جواب بهینه‌ای که به دست می‌آید به گونه‌ای باشد که در آن تعدادی از رول‌ها با روشی به غیر از روش‌های مذکور برش داده شوند، می‌توان با انجام برش‌هایی اضافی، نوع برش آن رول‌ها را به یکی از انواع مذکور تبدیل کرد. بدین ترتیب جواب بهینه‌ای به دست می‌آید که در آن تنها از شیوه‌های برش مذکور استفاده شده باشد. بنا بر این تنها کافی‌ست که این دوازده حالت را در نظر بگیریم و

[4] objective function
[5] constraint

برای هر کدامشان یک متغیر داشته باشیم:

$$\begin{aligned}
&P_۱ : ۲ \times ۱۳۵\\
&P_۲ : ۱۳۵ + ۱۰۸ + ۴۲\\
&P_۳ : ۱۳۵ + ۹۳ + ۴۲\\
&P_۴ : ۱۳۵ + ۳ \times ۴۲\\
&P_۵ : ۲ \times ۱۰۸ + ۲ \times ۴۲\\
&P_۶ : ۱۰۸ + ۲ \times ۹۳\\
&P_۷ : ۱۰۸ + ۹۳ + ۲ \times ۴۲\\
&P_۸ : ۱۰۸ + ۴ \times ۴۲\\
&P_۹ : ۳ \times ۹۳\\
&P_{۱۰} : ۲ \times ۹۳ + ۲ \times ۴۲\\
&P_{۱۱} : ۹۳ + ۴ \times ۴۲\\
&P_{۱۲} : ۷ \times ۴۲
\end{aligned}$$

متغیرهای مربوط به این دوازده حالت را به ترتیب $x_۱$ تا $x_{۱۲}$ می‌نامیم. تابع هدف تعداد کل رول‌هایی‌ست که برش داده می‌شوند که برابر است با:

$$\sum_{i=۱}^{۱۲} x_i$$

که کارخانه قصد دارد آن را کمینه کند. حال باید قیود را مشخص کنیم. واضح است که تعداد رول‌هایی که به یک روش خاص برش داده می‌شوند نمی‌تواند منفی باشد؛ بنا بر این به ازای هر $۱ \le i \le ۱۲$ قید $x_i \ge ۰$ را داریم. هم‌چنین کارخانه وظیفه دارد که از هر نوع ورقه به تعداد مناسب تولید کند. به طور مثال کارخانه باید حداقل ۳۹۵ ورقه به طول ۹۳ سانتی‌متر تولید کند. اگر رول‌ها را بر اساس نوع برشی که می‌خورند به دوازده دسته افراز کنیم و به ازای هر $۱ \le i \le ۱۲$ تعداد ورقه‌های ۹۳ سانتی‌متری برش iاُم را k_i بنامیم، تعداد کل ورقه‌های ۹۳ سانتی‌متری تولید شده توسط دسته‌ی iاُم برابر با $k_i x_i$ خواهد بود. بنا بر این باید داشته باشیم:

$$\sum_{i=۱}^{۱۲} k_i x_i \ge ۳۹۵$$

که با توجه به داده‌ها به این صورت درمی‌آید:

$$x_۳ + ۲x_۶ + x_۷ + ۳x_۹ + ۲x_{۱۰} + x_{۱۱} \ge ۳۹۵$$

قیود مربوط به انواع دیگر ورقه نیز به همین شیوه به دست می‌آیند.

سؤال. *آیا در چهار قید مربوط به ورقه‌های مختلف می‌توان «$\ge$» را با «$=$» جای‌گزین کرد؟ چرا؟*

بدین ترتیب فرمول‌بندی مسئله‌ی برنامه‌ریزی خطی‌مان کامل می‌شود؛ اما در این‌جا به یک مشکل اساسی برمی‌خوریم! با حل این مسئله‌ی برنامه‌ریزی خطی، مقدار $x_۱$ در جواب بهینه برابر با ۵/۴۸ می‌شود. به وضوح نمی‌توان تعداد ناصحیحی از رول‌ها را به شیوه‌ای خاص برش داد. در این مسئله، یک جواب زمانی به دردمان می‌خورد که مقادیر $x_۱$ تا $x_{۱۲}$ در آن صحیح باشد. بنا بر این باید فرمول‌بندی مسئله را تغییر داده و قیود $x_i \in \mathbb{Z}$ ($۱ \le i \le ۱۲$) را به آن اضافه کنیم. □

مسائل برنامه‌ریزی متعددی وجود دارند که همانند مثال بالا، در آن‌ها به دنبال جوابی هستیم که مقادیر متغیرها همگی صحیح باشد. در ادامه، این نوع از مسائل را دقیق‌تر بررسی می‌کنیم.

۳ مسائل برنامه‌ریزی صحیح

در این بخش، ابتدا تعریفی دقیق از مسائل برنامه‌ریزی صحیح[۶] ارائه داده و سپس نشان می‌دهیم که این گونه از مسائل دشوارتر از مسائل برنامه‌ریزی خطی هستند.

۱.۳ تعریف

تعریف ۲. *مسائل برنامه‌ریزی صحیح، گونه‌ای خاص از مسائل برنامه‌ریزی هستند که فرم کلی آن‌ها به این صورت است:*

$$\begin{aligned} \text{کمینه کن/بیشینه کن} \quad & c^T x \\ \text{که} \quad & Ax \le b \\ & x_i \in \mathbb{Z} \quad (۱ \le i \le n) \end{aligned}$$

در واقع فرمول‌بندی این مسائل مشابه با فرمول‌بندی مسائل برنامه‌ریزی خطی‌ست، با این تفاوت که در آن‌ها n قید دیگر داریم که نشان می‌دهند که مقادیر متغیرها باید صحیح باشد.

۲.۳ کلاس پیچیدگی مسائل برنامه‌ریزی صحیح

می‌دانیم که مسائل برنامه‌ریزی خطی در زمان چندجمله‌ای بر حسب تعداد متغیرها قابل حل هستند و در کلاس P قرار می‌گیرند. مسائل برنامه‌ریزی صحیح چه‌طور؟ قضیه‌ی زیر به ما نشان می‌دهد که این مسائل در کلاس NP-Hard قرار می‌گیرند.

قضیه ۳. *مسائل برنامه‌ریزی صحیح در کلاس NP-Hard قرار می‌گیرند.*

اثبات. برای این که نشان دهیم که یک مسئله در کلاس NP-Hard قرار دارد، باید نشان دهیم که تمامی مسائل NP به آن مسئله تحویل چندجمله‌ای می‌شوند. این کار معمولا به این صورت انجام می‌شود که یک مسئله‌ی NP-Complete را به مسئله‌ی مذکور تحویل چندجمله‌ای می‌کنیم. در این‌جا ما هم از همین شیوه استفاده می‌کنیم و مسئله‌ی CNF-SAT که یک مسئله‌ی NP-Complete است را به یک مسئله‌ی برنامه‌ریزی صحیح تحویل چندجمله‌ای می‌کنیم. برای فهم راحت‌تر الگوریتم تحویل، عمل‌کرد آن را روی یک نمونه از مسئله‌ی CNF-SAT توصیف می‌کنیم. به مثال زیر توجه کنید:

$$(x_۱ \vee \neg x_۲) \wedge (\neg x_۱ \vee x_۲ \vee x_۳) \wedge \neg x_۱$$

برای ارضا کردن عبارت بالا باید برای هر یک از سه متغیر $x_۱$، $x_۲$ و $x_۳$ یکی از دو مقدار صفر و یک را انتخاب کنیم به طوری که حاصل کل عبارت برابر با یک شود. برای تبدیل این نمونه به یک مسئله‌ی برنامه‌ریزی صحیح مجموعه‌ی متغیرها را همان مجموعه‌ی $\{x_۱, x_۲, x_۳\}$ در نظر می‌گیریم. مقدار هر متغیر باید برابر با صفر یا یک باشد؛ بنا بر این به ازای هر $۱ \le i \le ۳$ سه قید زیر باید برقرار باشند:

$$\begin{aligned} x_i &\in \mathbb{Z} \\ x_i &\ge ۰ \\ x_i &\le ۱ \end{aligned}$$

برای این که حاصل کل عبارت برابر با یک شود، حاصل هر یک از این سه عبارت باید برابر با یک شود:

$$\begin{aligned} x_۱ \vee \neg x_۲ \\ \neg x_۱ \vee x_۲ \vee x_۳ \\ \neg x_۱ \end{aligned}$$

به ازای هر $۱ \le i \le ۳$ عبارت $\neg x_i$ را با $۱ - x_i$ معادل‌سازی می‌کنیم. برای این که حاصل هر کدام از سه عبارت بالا برابر با یک شود، باید مقدار حداقل یکی از جملاتش برابر با یک شود. این معادل با این است که در برنامه‌ریزی صحیحمان حاصل‌جمع عبارت‌های متناظر با این جملات حداقل

[۶] در این بخش و بخش‌های بعدی، عبارت «برنامه‌ریزی صحیح» را به جای «برنامه‌ریزی صحیح خطی» به کار می‌بریم. در اصل، مسائل برنامه‌ریزی صحیح لزوما خطی نیستند اما ما در این‌جا تنها حالت خطی را مد نظر قرار می‌دهیم.

برابر با یک باشد. پس سه قید زیر باید برقرار باشند:

$$x_1 + (1 - x_2) \geq 1$$
$$(1 - x_1) + x_2 + x_3 \geq 1$$
$$(1 - x_1) \geq 1$$

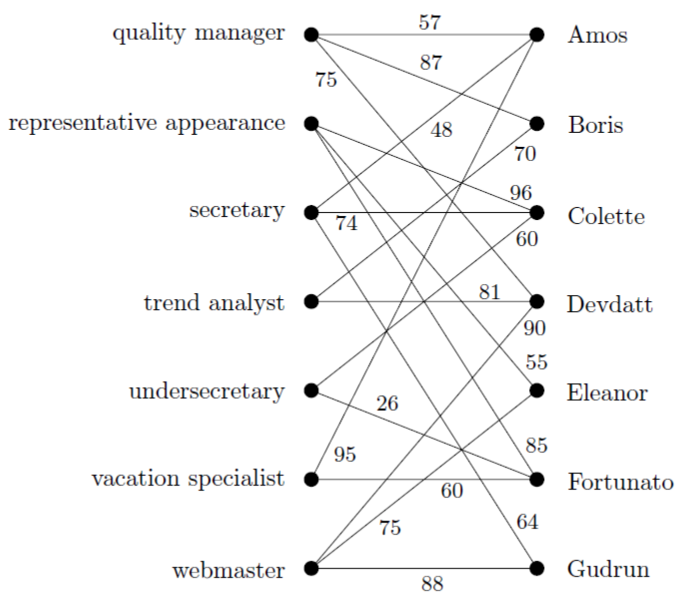
تابع هدف باید چه باشد؟ آن را باید بیشینه کرد یا کمینه؟ اهمیتی ندارد؛ زیرا در مسئله‌ی CNF-SAT صرفا به دنبال یک جواب می‌گردیم که عبارت داده‌شده را ارضا کند و شرط دیگری وجود ندارد. بنا بر این در برنامه‌ریزی صحیح می‌توانیم هر تابعی که بر حسب متغیرها خطی‌ست را به عنوان تابع هدف در نظر بگیریم و به دل‌خواه انتخاب کنیم که کمینه یا بیشینه شود.

بدین ترتیب هر نمونه‌ی دل‌خواه از مسئله‌ی CNF-SAT در زمان چندجمله‌ای به یک نمونه از مسئله‌ی برنامه‌ریزی صحیح تبدیل می‌شود، به طوری که آن نمونه از مسئله‌ی برنامه‌ریزی صحیح دارای حداقل یک جواب باشد اگر و تنها اگر عبارت داده‌شده در مسئله‌ی CNF-SAT ارضاپذیر باشد. □

۴ مثال: تطابق وزن‌دار کامل در گراف دوبخشی

در بخش قبل اثبات کردیم که مسائل برنامه‌ریزی صحیح در کلاس NP-Hard قرار می‌گیرند. بنا بر این نمی‌توان آن‌ها را در زمان چندجمله‌ای بر حسب تعداد متغیرها حل کرد (یا حداقل با دانش کنونی نمی‌توان این کار را انجام داد). آیا این بدان معناست که برنامه‌ریزی صحیح در عمل کاربردی ندارد؟ در این بخش و بخش بعدی با بررسی یک مثال نشان می‌دهیم که پاسخ این پرسش منفی‌ست.

مثال ۴. *یک شرکت می‌خواهد هفت نفر را برای هفت سِمَت مختلف استخدام کند. هر یک این هفت نفر در برخی از این هفت زمینه دارای تجربه هستند. با بررسی‌های انجام‌شده، مشخص شده‌است که انتصاب هر کدام از این هفت نفر به هر کدام از زمینه‌هایی که در آن دارای تجربه هستند چه مقدار به شرکت سود می‌رساند. گراف دوبخشی زیر نتایج به‌دست‌آمده را نشان می‌دهد:*

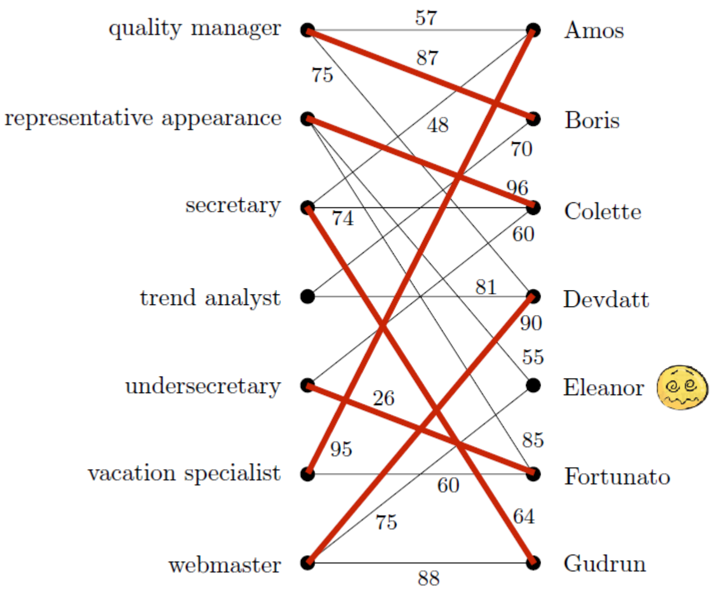


بخش سمت چپ هفت سِمَت و بخش سمت راست هفت نامزد را نشان می‌دهد. بین یک نامزد و یک سِمَت یال وجود دارد اگر و تنها اگر آن نامزد در آن زمینه دارای تجربه باشد. عدد روی هر یال نشان‌دهنده‌ی میزان سود ناشی از انتصاب یک فرد به یک سِمَت است.

هدف شرکت آن است که برای هر سِمَت یک نفر را استخدام کند به طوری که هیچ‌کس بیش‌تر از یک سِمَت را در اختیار نداشته باشد و مجموع سودهای ناشی از انتصاب‌ها بیشینه شود. با در نظر گرفتن این شرایط، یک حالت بهینه برای انتصاب‌ها به دست آورید.

حل. با کمی دقت مشخص می‌شود که باید یک تطابق کامل با وزن بیشینه در گراف بالا پیدا کنیم. این مسئله را چه‌طور باید حل کرد؟

نخستین ایده‌ای که به ذهن می‌رسد استفاده از الگوریتم‌های حریصانه است؛ یعنی با شروع از نخستین فرد (در مثال ما، Amos)، هر بار پرسودترین سِمَتی که فرد مورد نظر در آن تجربه دارد و هنوز به شخص دیگری داده نشده‌است را به او بدهیم. متأسفانه این روش لزوما جواب بهینه را به ما نمی‌دهد. در واقع اگر همین الگوریتم را روی گراف بالا اجرا کنیم به نتیجه‌ی زیر می‌رسیم:



یال‌های قرمز، یال‌هایی هستند که توسط الگوریتم انتخاب می‌شوند. همان‌طور که مشاهده می‌شود، هنگامی که الگوریتم به Eleanor می‌رسد، نمی‌تواند او را استخدام کند؛ زیرا پیش از این برای تمامی سِمَت‌هایی که Eleanor در آن‌ها تجربه دارد فردی را انتخاب کرده‌است.

حال سعی می‌کنیم که با استفاده از برنامه‌ریزی صحیح مسئله را حل کنیم. چگونه مسئله‌ی اصلی را به یک مسئله‌ی برنامه‌ریزی صحیح تبدیل کنیم؟

یک ایده این است که به ازای هر نامزد یک متغیر در نظر بگیریم که مقدار آن نشان دهد که آن فرد به کدام سِمَت منصوب شده‌است؛ اما با این نوع مدل‌سازی، بیان کردن شروط مسئله‌ی اصلی به صورت یک یا چند قید (نامساوی) بسیار پیچیده خواهد شد. پس از این روش استفاده نمی‌کنیم.

ایده‌ی دیگر آن است که به ازای هر یال یک متغیر در نظر بگیریم که نشان دهد که آن یال در تطابق آمده‌است یا خیر. بنا بر این مقدار هر متغیر باید برابر با صفر یا یک باشد. بدین ترتیب اگر متغیر مربوط به هر یال مانند e را x_e بنامیم، سه قید زیر باید برقرار باشند:

$$x_e \in \mathbb{Z}$$

$$x_e \geq ۰$$

$$x_e \leq ۱$$

برای این که یال‌های انتخاب‌شده یک تطابق کامل را تشکیل دهند، به هر رأس مانند v باید دقیقا یک یال انتخاب‌شده متصل باشد. این یعنی مجموع مقدارهای متغیرهای مربوط به یال‌های متصل به v باید دقیقا برابر با یک شود. پس اگر مجموعه‌ی یال‌های متصل به v را با E_v نشان دهیم، قیدهای زیر باید برقرار باشند:

$$\sum_{e \in E_v} x_e \geq ۱$$

$$\sum_{e \in E_v} x_e \leq ۱$$

هدف این است که مجموع وزن یال‌های انتخاب‌شده را بیشینه کنیم. بنا بر این اگر مجموعه‌ی یال‌های گراف را E و وزن هر یال مانند e را w_e بنامیم، تابع هدف برابر با

$$\sum_{e \in E} w_e x_e$$

خواهد شد که همان‌طور که بیان شد باید مقدار آن را بیشینه کنیم. بدین ترتیب مسئله‌ی اصلی را به فرم یک مسئله‌ی برنامه‌ریزی صحیح درآوردیم.

سؤال. *در صورتی که شرط کامل بودن تطابق را از مسئله‌ی اصلی حذف کنیم، چه تغییراتی باید در قیود مسئله‌ی برنامه‌ریزی صحیح‌مان اعمال شود؟*

□

راه‌حلی که برای مسئله‌ی اخیر ارائه شد همواره جواب بهینه را به ما می‌دهد؛ اما هنوز تا رسیدن به یک الگوریتم درست که در زمان معقولی قابل اجرا باشد فاصله داریم.

۵ آرام‌سازی

در این بخش یک تکنیک بسیار مهم در حل مسائل برنامه‌ریزی صحیح در زمان معقول (چندجمله‌ای) را معرفی می‌کنیم که «آرام‌سازی»[7] نام دارد. سپس با استفاده از این تکنیک و یک الگوریتم کمکی، مثال ۴ را در زمان چندجمله‌ای حل می‌کنیم تا قدرت آن در حل مسائل را ببینیم.

۱.۵ آرام‌سازی یعنی چه؟

به زبان ساده، آرام‌سازی یک مسئله‌ی برنامه‌ریزی صحیح یعنی حذف قیود مربوط به صحیح بودن مقدار متغیرها و تبدیل کردن مسئله‌ی مذکور به یک مسئله‌ی برنامه‌ریزی خطی. دلیل انتخاب این نام آن است که حذف قیود مذکور موجب افزایش آزادی عمل ما و به اصطلاحْ «آرام شدن» شروط مسئله‌ی برنامه‌ریزی می‌شود.

۲.۵ کاربرد آرام‌سازی در حل مسائل

در نگاه نخست به نظر می‌رسد که آرام‌سازی یک مسئله‌ی برنامه‌ریزی صحیح هیچ کمکی به ما نخواهد کرد؛ زیرا با حذف قیود مربوط به صحیح بودن مقدار متغیرها دیگر تضمینی وجود ندارد که در جواب بهینه مقدار همه‌ی متغیرها صحیح باشد. نکته در این‌جاست که درست است که آرام‌سازی به تنهایی دردی را دوا نمی‌کند، اما در برخی موارد می‌توان از روی جواب بهینه‌ی ناصحیح در زمان معقولی یک جواب صحیح به دست آورد که بهینه (یا «تقریبا بهینه»)[8] باشد.

همان‌طور که اشاره شد، هنگامی که یک مسئله‌ی برنامه‌ریزی صحیح را آرام‌سازی می‌کنیم آزادی عمل بیش‌تری پیدا می‌کنیم. بنا بر این می‌توان گفت که جواب بهینه‌ی مسئله‌ی برنامه‌ریزی خطی به‌دست‌آمده «بهتر» از جواب بهینه‌ی مسئله‌ی اصلی‌ست؛ به این معنا که اگر مقدار تابع هدف برای جواب بهینه‌ی برنامه‌ریزی خطی و برنامه‌ریزی صحیح را به ترتیب y_{LP} و y_{IP} بنامیم، در صورتی که هدفمان بیشینه کردن تابع هدف باشد خواهیم داشت $y_{LP} \geq y_{IP}$ و در صورتی که هدفمان کمینه کردن آن باشد خواهیم داشت $y_{LP} \leq y_{IP}$. این گزاره‌ی ساده کاربرد زیادی در طراحی الگوریتم‌ها دارد و در جلسه‌ی آینده از آن برای طراحی یک الگوریتم تقریبی بهره می‌گیریم.

۳.۵ یافتن تطابق وزن‌دار کامل با وزن بیشینه در گراف دوبخشی در زمان چندجمله‌ای

با استفاده از تکنیک آرام‌سازی در حل مثال ۴، مسئله‌ی برنامه‌ریزی صحیحمان به یک مسئله‌ی برنامه‌ریزی خطی تبدیل می‌شود. می‌دانیم که این مسئله در زمان چندجمله‌ای قابل حل است. با حل آن یک جواب بهینه به دست می‌آید که مقدار متغیرها در آن لزوما صحیح نیست. اگر بتوانیم در زمان چندجمله‌ای این جواب ناصحیح بهینه را به یک جواب صحیح بهینه تبدیل کنیم، یک الگوریتم چندجمله‌ای برای حل مسئله‌ی اصلی خواهیم داشت. قضیه‌ی زیر به ما نشان می‌دهد که خوش‌بختانه در این‌جا چنین کاری شدنی‌ست.

قضیه ۵. *در مثال ۴ اگر مسئله‌ی برنامه‌ریزی خطی ساخته‌شده از روی مسئله‌ی برنامه‌ریزی صحیح دارای حداقل یک جواب شدنی باشد، مسئله‌ی برنامه‌ریزی صحیح نیز دارای یک جواب شدنی‌ست که مقدار تابع هدف در آن برابر با مقدار تابع هدف برای جواب بهینه‌ی مسئله‌ی برنامه‌ریزی خطی‌ست (و در نتیجه جوابی بهینه است).*

اثبات. یک جواب بهینه‌ی مسئله برنامه‌ریزی خطی را در نظر بگیرید (با توجه به فرض چنین جوابی حتما وجود دارد). حال زیرگرافی را در نظر بگیرید که تمامی رئوس گراف در آن حضور دارند و یک یال در آن ظاهر شده‌است اگر و تنها اگر مقدار متغیرش مثبت باشد. آن را H می‌نامیم. با توجه به قیود می‌دانیم که تمامی مؤلفه‌های همبندی در H شامل حداقل دو رأس می‌شوند. اگر یک مؤلفه‌ی همبندی در H به شکل درخت باشد، حداقل دو برگ دارد و طبق قیود، مقدار متغیر یال متصل به هر برگ باید برابر با یک باشد. بنا بر این می‌توان گفت که همسایه‌ی هر برگ نیز یک برگ است. پس آن مؤلفه حتما به صورت یک تک‌یال است که مقدار متغیرش برابر با یک است. هم‌چنین اگر مقدار متغیر یک یال برابر با یک باشد، مؤلفه‌ای که شامل آن یال می‌شود حتما به صورت یک تک‌یال است. اگر تمامی مؤلفه‌ها به صورت تک‌یال باشند یعنی H یک تطابق کامل است و مقدار تمامی متغیرها صحیح است. پس در این حالت حکم برقرار خواهد بود. حال نشان می‌دهیم که هر حالت دیگری از H را با تغییر دادن مقدار متغیرها و بدون نقض قیود و تغییر دادن مقدار تابع هدف می‌توان به این حالت مطلوب تبدیل کرد. اگر H به صورت یک تطابق کامل نباشد، با توجه به مطالب اخیر می‌توان گفت که حداقل یک یال در آن وجود دارد که مقدار متغیرش صحیح نباشد. حال با روش زیر به تعداد یال‌هایی که مقدار متغیرشان صحیح است حداقل یکی اضافه می‌کنیم:

با توجه به این که H به صورت یک تطابق کامل نیست و طبق نتایجی که گرفتیم، حداقل یک مؤلفه‌ی همبندی در آن وجود دارد که به شکل درخت نیست و در نتیجه حداقل یک دور در H وجود دارد. یکی از دورهای H را به دل‌خواه در نظر می‌گیریم. می‌دانیم که مقدار متغیرهای تمام

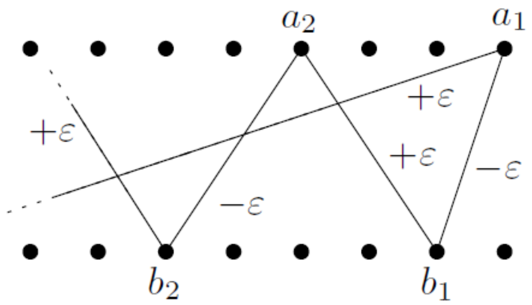
[7] relaxation

[8] در جلسه‌ی آینده مثالی را خواهیم دید که در آن با استفاده از تکنیک آرام‌سازی یک الگوریتم تقریبی را طراحی می‌کنیم.

یال‌های آن ناصحیح است. هم‌چنین با توجه به دوبخشی بودن گراف، طول این دور حتما زوج است. بدون کم شدن از کلیت مسئله فرض می‌کنیم که طولش برابر با $۲k$ باشد. حال با شروع از یک یال دل‌خواه، یال‌های دور را به ترتیب با اعداد ۱ تا $۲k$ شماره‌گذاری می‌کنیم. مجموعه‌ی یال‌هایی که شماره‌ی زوج و شماره‌ی فرد دارند را به ترتیب با $E_۰$ و $E_۱$ و مجموع وزن یال‌هایی که شماره‌ی زوج و شماره‌ی فرد دارند را به ترتیب با $W_۰$ و $W_۱$ نشان می‌دهیم. بدون کم شدن از کلیت مسئله فرض می‌کنیم که $W_۰ \geq W_۱$. به ازای هر یال مانند e مقدار متغیرش را x_e^* می‌نامیم. حال تعریف می‌کنیم:

$$\epsilon := \min(\{۱ - x_e^* \,|\, e \in E_۰\} \cup \{x_e^* \,|\, e \in E_۱\})$$

اکنون مطابق شکل صفحه‌ی بعد، مقدار متغیرهای یال‌های $E_۰$ را به اندازه‌ی ϵ افزایش و مقدار متغیرهای یال‌های $E_۱$ را به اندازه‌ی ϵ کاهش می‌دهیم.



بدین ترتیب با توجه به تعریف ϵ، مقدار متغیر حداقل یکی از یال‌ها صحیح خواهد شد و در عین حال هیچ‌یک از قیود نقض نمی‌شود و مقدار تابع هدف به اندازه‌ی $\epsilon(W_۰ - W_۱)$ افزایش می‌یابد. از طرفی با توجه به این که جواب پیشین بهینه بود، مقدار تابع هدف در آن بیشینه بود. پس داریم $\epsilon(W_۰ - W_۱) = ۰$ که یعنی مقدار تابع هدف تغییری نمی‌کند.

در حالت کلی اگر تعداد رئوس هر بخش را n در نظر بگیریم، پس از حداکثر n بار تکرار الگوریتم بالا، H به یک تطابق کامل تبدیل می‌شود و مقدار تابع هدف تغییری نخواهد کرد. بنا بر این یک جواب صحیح و بهینه خواهیم داشت. □

با توجه به این که پیدا کردن یک دور در H و اصلاح مقادیر متغیرهای یال‌های آن در زمان چندجمله‌ای قابل انجام است و تعداد دفعات تکرار این پروسه نیز بر حسب اندازه‌ی گراف چندجمله‌ای‌ست، جواب بهینه‌ی مسئله‌ی برنامه‌ریزی خطی در زمان چندجمله‌ای به یک جواب بهینه برای مسئله‌ی اصلی تبدیل می‌شود. بدین ترتیب کل مسئله در زمان چندجمله‌ای حل می‌شود.

مراجع

[1] Jiří Matoušek and Bernd Gärtner. Understanding and Using Linear Programming. *Springer-Verlag Berlin Heidelberg,* 1st edition, 2007.

[۲] اسلایدهای جلسه‌ی چهارم.

[۳] جزوه‌ی جلسه‌ی دوم.

[۴] جزوه‌ی جلسه‌ی سوم.