اگر

ازای

به‌ازای

تاوقتی

جلسهlecture

جلسه اولمروری سریع بر برنامه‌ریزی خطیحامد واسعی

با توجه به این که عنوان درس کاربردهای برنامه‌ریزی خطی[۱] در الگوریتم‌های تقریبی است، لازم است در ابتدا مروری سریع بر مبحث برنامه‌ریزی خطی(LP) و مهم‌ترین قضایای آن که در این درس مورد استفاده قرار می‌گیرند، داشته باشیم.

۱ صورت‌بندی مسئله برنامه‌ریزی خطی

در برنامه ریزی خطی ما به دنبال پیدا کردن بردار x هستیم که گویا و نامنفی باشد و ضمن ارضا کردن تعدادی قید(یا شرط) خطی بر حسب آن، تابعی خطی از آن را کمینه کند. به صورت دقیق‌تر با داشتن بردارهای $c \in \mathbb{Q}^n$ و $b \in \mathbb{Q}^m$ و ماتریس $A = (a_{ij}) \in \mathbb{Q}^{m\times n}$، یک *جواب بهینه*[۲] برای مسأله‌ی برنامه‌ریزی خطی زیر یک بردار nـ‌تایی x است که تابع هدف ۷ را با ارضای شرایط ۲ و ۳ کمینه می‌کند.

$$\text{minimize} \quad \sum_{j=1}^{n} c_j x_j \tag{۱}$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, \dots, m \tag{۲}$$

$$x_j \geq 0, \qquad j = 1, \dots, n \tag{۳}$$

هر x که در شرایط ۲ و ۳ صدق کند، یک *جواب شدنی*[۳] نامیده می‌شود.[۴] هر مسأله LP که یک جواب شدنی داشته باشد، *شدنی* و در غیر این صورت *ناشدنی*[۵] نامیده می‌شود. *حل کردن* یک LP به معنای پیدا کردن یک جواب بهینه برای آن است. برای حل مسائل LP الگوریتم‌های کارایی وجود دارد که همه روزه برای حل LPهایی با هزاران متغیر و قید استفاده می‌شوند.

لازم به ذکر است که صورت ارائه شده از مسئله‌ی LP صورت کانونی[۶] این مسئله است و از عمومیت لازم برخوردار است. برای مثال با تغییر علامت تابع هدف می‌توان مسائل بیشینه‌سازی را به کمینه‌سازی تبدیل کرد. همچنین هر قید تساوی مثل $\Sigma_{j=1}^{n} a_{ij} = b_i$ را می‌توان به صورت قرار دادن دو قید $\Sigma_{j=1}^{n} a_{ij} \geq b_i$ و $-\Sigma_{j=1}^{n} a_{ij} \geq -b_i$ به صورت همزمان بیان کرد. علاوه بر این اگر بخواهیم متغیر x_j بتواند مقادیر منفی اتخاذ کند، می‌توان با تعریف دو متغیر مثبت x_j^+ و x_j^- و جایگزینی $x_j^+ - x_j^-$ به جای x_j در تابع هدف و تمامی قیود به این منظور در صورت کانونی دست پیدا کنیم.

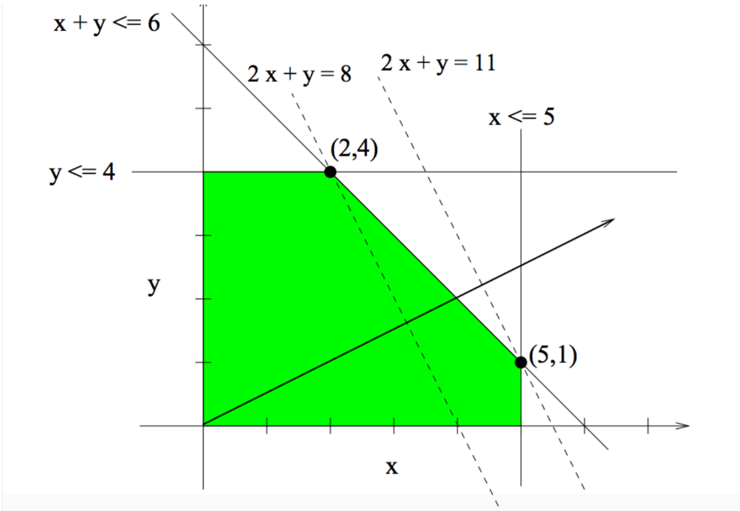
مثال ۱. *با در نظر گرفتن x و y به عنوان متغیر می‌خواهیم تابع هدف $2x + y$ را مقید به قیدهای زیر بیشینه کنیم:*

$$\begin{aligned} x + y &\leq 6 \\ x &\leq 5 \\ y &\leq 4 \\ x, y &\geq 0 \end{aligned}$$

همان‌طور که دیده می‌شود این بیان در صورت کانونی نیست. می‌توان با ضرب کردن تابع هدف و سه قید اول در -1 این مسئله را در صورت کانونی بیان کرد. اگر تابع هدف و قیود را در صفحه‌ی ۲ـ بعدی رسم کنیم، خواهیم دید که مجموعه‌ی جواب‌های شدنی این مسئله قسمت رنگی شکل ۱ خواهد بود و جواب بیشینه‌ی آن نقطه‌ی $(x, y) = (5, 1)$ خواهد بود.

[۱] linear programming
[۲] optimal solution
[۳] feasible solution
[۴] دلیل محدود کردن به اعداد گویا آن است که بتوان مسأله را به عنوان ورودی به کامپیوتر داد. می‌توان نشان داد که اگر مسأله‌ی داده شده یک جواب بهینه داشته باشد، یک جواب بهینه که همه‌ی مؤلفه‌های آن گویا باشد نیز وجود دارد! این موضوع از نظر تحلیل پیچیدگی الگوریتم‌های حل LP اهمیت دارد.
[۵] infeasible
[۶] canonical form

شکل ۱: منبع شکل: اسلایدهای مربوط به درس نظریه‌ی الگوریتمی بازی‌های دکتر کوشا اعتصامی



یک مسئله‌ی LP ممکن است جواب بهینه نداشته نباشد. این موضوع در دو صورت ممکن است رخ دهد. اول آن‌که مسئله جوابی شدنی نداشته باشد. دوم آن‌که تابع هدف روی مجموعه‌ی جواب‌های شدنی از پایین(در صورت کانونی) کراندار نباشد.[۷]

اگر یک قید $a_{ij}x_j \geq b$ به صورت تساوی ارضا شده باشد، یعنی برای $\bar{x}$ داشته باشیم $a_{ij}\bar{x}_j = b$ می‌گوییم این قید در $\bar{x}$ *فعال*[۸] است. یک x_B را *جواب پایه‌ای*[۹] گوییم اگر در آن mتا از قیود فعال باشند(با فرض مستقل خطی بودن قیود). البته این جواب پایه‌ای ممکن است ناشدنی باشد چون بقیه‌ی قیود ممکن است برقرار نباشند. قضیه‌ی اساسی آن است که هرگاه یک جواب بهینه(یا شدنی) برای مسئله وجود داشته باشد، حتما یک جواب پایه‌ای بهینه(یا شدنی) نیز وجود خواهد داشت.

اگر در یک مسئله‌ی LP ما بر روی تعدادی از متغیرها قید صحیح بودن یا قیدهایی مثل $x_j \in \{0,1\}$ را اضافه کنیم، مسئله به یک مسئله‌ی *برنامه‌ریزی خطی صحیح*[۱۰] یا به اختصار *برنامه‌ریزی صحیح*[۱۱] (IP) تبدیل می‌شود. می‌دانیم که برنامه‌ریزی صحیح از نظر پیچیدگی یک مسئله‌ی NP-Complete است. با این حال برنامه‌ریزی صحیح یک روش مهم برای مدل کردن بسیاری از مسئله‌های بهینه‌سازی ترکیبیاتی است و چنان‌که در این درس خواهیم دید بسیار مورد استفاده قرار خواهند گرفت. روند کلی که در این درس مورد استفاده قرار می‌گیرد آن است که ابتدا صورت مسئله معمولا به صورت یک مسئله‌ی IP صورت بندی می‌شود. سپس با انجام *آرام‌سازی‌هایی*[۱۲] به یک مسئله‌ی LP می‌رسیم که می‌توانیم آن را حل کنیم. سپس با گردکردن[۱۳] یک جواب برای مسئله‌ی اصلی تولید می‌کنیم و نشان می‌دهیم که این جواب تولید شده چه نسبتی با جواب بهینه دارد.

۲ دوگان

مفهوم *دوگانی*[۱۴] در برنامه‌ریزی خطی بسیار جالب و کاربردی است. ابتدا آن را در قالب یک مثال توضیح می‌دهیم و سپس به تعریف دقیق آن می‌پردازیم. مسئله‌ی LP زیر را در نظر بگیرید:

$$\begin{aligned} \text{minimize} \quad & 6x_1 + 4x_2 + 2x_3 \\ \text{subject to} \quad & 4x_1 + 2x_2 + x_3 \geq 5, \\ & x_1 + x_2 \geq 3, \\ & x_2 + x_3 \geq 4, \\ & x_i \geq 0, \quad i = 1,2,3. \end{aligned}$$

[۷] دقت داشته باشید که اگر در قیود می توانستیم از < یا > نیز استفاده کنیم ممکن بود حالت دیگری نیز رخ دهد یعنی در عین تهی نبودن مجموعه‌ی جواب‌های شدنی و کراندار بودن آن، باز هم جواب بهینه موجود نباشد.

[۸] active or binding
[۹] basic solution
[۱۰] integer linear programming
[۱۱] integer programming
[۱۲] relaxation
[۱۳] rounding
[۱۴] duality

مشاهده می‌شود که چون همه‌ی x_jها نامنفی هستند پس $6x_1 + 4x_2 + 2x_3 \geq 4x_1 + 2x_2 + x_3$. با توجه به قید اول میدانیم که $4x_1+2x_2+x_3 \geq 5$ پس تابع هدف همیشه بزرگتر یا مساوی ۵ خواهد بود و ۵ یک کران پایین برای جواب بهینه است. با ترکیب قیود می‌توانیم به کران پایین‌های بهتری دست پیدا کنیم. برای مثال می‌دانیم $6x_1+4x_2+2x_3 \geq (4x_1+2x_2+x_3)+2\times(x_1+x_2) \geq 5+2\times 3 = 11$ که از ترکیب قیدهای اول و دوم به دست آمده‌است. برای به دست آوردن یک کران پایین بهتر می‌توانیم هر سه قید را به کار گیریم که در این صورت خواهیم داشت : $6x_1 + 4x_2 + 2x_3 \geq (4x_1 + 2x_2 + x_3) + (x_1 + x_2) + (x_2 + x_3) \geq 5 + 3 + 4 = 12$

در حقیقت ما می‌توانیم یک مسئله‌ی برنامه‌ریزی خطی تعریف کنیم تا ضرایب قیدهای مختلف را برای محاسبه‌ی بهترین کران پایین برای تابع هدف محاسبه کنیم. فرض کنید که ضریب قید اول تا سوم به ترتیب y_1، y_2 و y_3 باشد که همه نامنفی هستند. در این صورت کران پایین به دست آمده برای تابع هدف $5y_1 + 3y_2 + 4y_3$ خواهد بود. ما باید اطمینان حاصل کنیم که

$$6x_1 + 4x_2 + 2x_3 \geq y_1(4x_1 + 2x_2 + x_3) + y_2(x_1 + x_2) + y_3(x_2 + x_3)$$

برای این کار باید تضمین کنیم که ضریب x_jها در سمت چپ نامساوی بزرگتر از ضریب همان متغیر در سمت راست است لذا باید داشته باشیم $4y_1 + y_2 \leq 6$، $2y_1 + y_2 + y_3 \leq 4$ و $y_1 + y_3 \leq 2$. هدف آن است که کران پایین حاصل از این قیود را بیشینه کنیم که می‌توان آن را به صورت مسئله‌ی برنامه‌ریزی زیر نوشت:

$$\begin{aligned}
\text{maximize} \quad & 5y_1 + 3y_2 + 4y_3 \\
\text{subject to} \quad & 4y_1 + y_2 \leq 6, \\
& 2y_1 + y_2 + y_3 \leq 4, \\
& y_1 + y_3 \leq 2, \\
& y_1 + y_3 \geq 0, \quad i = 1, 2, 3.
\end{aligned}$$

این مسئله‌ی بیشینه‌سازی که خود یک LPاست، *دوگان*[۱۵] مسئله‌ی *اولیه*[۱۶] است. به راحتی دیده می‌شود که هر جواب شدنی برای دوگان یک کران پایین برای جواب بهینه‌ی مسئله‌ی اولیه است.

برای هر مسئله‌ی LP در صورت کانونی با تعریف‌های ۷، ۲ و ۳ دوگان آن را می‌توان به صورت زیر تعریف کرد:

$$\text{maximize} \quad \sum_{i=1}^{m} b_i y_i \tag{۴}$$

$$\text{subject to} \quad \sum_{i=1}^{m} a_{ij} y_i \leq c_j, \quad j = 1, \dots, n, \tag{۵}$$

$$y_i \geq 0, \qquad i = 1, \dots, m. \tag{۶}$$

تمرین ۱. *فرض کنیم شما یک آدم بسیار دقیق نسبت به رژیم غذای هستید که در عین حال می‌خواهید هزینه‌ی زندگی را تا حد ممکن پایین نگه دارید. شما می‌خواهید که در هر روز مقدار کافی از ویتامین‌های $v_1, \dots, v_m$ در سبد غذایی شما باشد. غذاهای $f_1, \dots, f_n$ موجود هستند و برای هر i, j میزان ویتامین v_i موجود در هر واحد از غذای f_j را با داریه‌ی a_{ij} از ماتریس A نمایش می‌دهیم. علاوه بر این هزینه‌ی تهیه‌ی هر واحد از غذای f_j را با c_j نمایش می‌دهیم. با فرض این که بتوانیم هر مقدار گویای دلخواه از هر غذا تهیه کنیم، مسئله‌ی برنامه‌ریزی خطی متناظر با کمینه کردن هزینه در عین تأمین نیاز تمام ویتامین‌ها را به صورت یک مسئله‌ی برنامه ریزی خطی بنویسید. علاوه بر این دوگان این مسئله را نیز بنویسید و با در نظر گرفتن واحدهای مناسب برای هر کدام از متغیرها، ثوابت و ضرایب سعی کنید دوگان را تعبیر کنید.*

۳ قضایای مربوط به دوگان

در این بخش تعدادی از مهم‌ترین قضایای مربوط به برنامه‌ریزی خطی و رابطه‌ی بین مسئله‌ی اولیه و دوگان آن را بیان می‌کنیم که در طول درس بارها و بارها از آن‌ها استفاده خواهیم کرد. در اولین قضیه به صورت دقیق بیان می‌کنیم که جواب‌های مسئله‌ی دوگان کران‌های پایینی برای مسئله‌ی اولیه هستند.

قضیه ۲ (دوگانی ضعیف[۱۷]). *اگر x یک جواب شدنی برای LP اولیه و y یک جواب شدنی برای دوگان آن باشد، در این صورت خواهیم داشت: $\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i$*

اثبات. چون x و y جواب‌های شدنی هستند پس با توجه به قیدهای مسئله‌ی اولیه و دوگان می‌توان نوشت:

[۱۵] dual
[۱۶] primal
[۱۷] weak duality

$$
\begin{aligned}
\sum_{i=1}^{m} b_i y_i &\leq \sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j) y_i \\
&= \sum_{j=1}^{n} x_j \sum_{i=1}^{m} a_{ij} y_i \\
&\leq \sum_{j=1}^{n} x_j c_j
\end{aligned}
$$

□

دقت کنید که قضیه‌ی دوگانی ضعیف ما را به شاهدی برای بهینه بودن جواب مسلح می‌کند، به این معنا که اگر ما یک x^* و یک y^* ارائه کنیم که به ترتیب یک جواب شدنی برای مسئله‌ی اولیه و دوگان آن باشند و در عین حال $c^T.x^* = b^T.y^*$ طبق قضیه‌ی دوگانی ضعیف می‌دانیم که هر دوی x^* و y^* جواب‌های بهینه‌ای برای مسائل متناظر هستند.
یک قضیه‌ی بسیار جالب و حیرت‌انگیز که کاربردهای بسیاری دارد و البته اثبات آن چندان ساده نیست، بیان می‌کند که هر گاه مسئله‌ی اولیه و دوگان آن هر دو جوابی شدنی داشته باشند، در این صورت مقدار تابع هدف بهینه برای هر دو مسئله یکسان است.

قضیه ۳ (دوگانی قوی[۱۸]). *در مورد یک مسئله‌ی LP و دوگان آن همیشه یکی از چهار حالت زیر برقرار است:*

۱. *هر دو مسئله جوابی شدنی دارند و برای هر جواب بهینه مثل x^* برای مسئله‌ی اولیه و y^* برای دوگان آن داریم:* $\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^*$

۲. *مسئله‌ی اولیه جواب شدنی ندارد و دوگان آن شدنی است، اما تابع هدف روی مجموعه‌ی جواب‌های شدنی کراندار نیست.*

۳. *مسئله‌ی دوگان جواب شدنی ندارد و مسئله‌ی اولیه شدنی است، اما تابع هدف روی مجموعه‌ی جواب‌های شدنی کراندار نیست.*

۴. *هر دو مسئله شدنی نیستند.*

یک روش برای اثبات این قضیه رجوع به اثبات‌های درستی الگوریتم سیمپلکس [۱۹] است که البته جزئیات فراوانی دارد و خارج از حوصله‌ی این متن است.
در ادامه به یک نتیجه‌ی ساده و در عین حال کاربردی این قضیه می‌پردازیم. پیش از آن باید *شرایط مکمل لنگی*[۲۰] را تعریف کنیم. اگر $\bar{x}$ و $\bar{y}$ به ترتیب دو جواب شدنی برای مسئله‌ی اولیه و دوگان باشند، می‌گوییم $\bar{x}$ و $\bar{y}$ شرایط مکمل لنگی را دارا هستند اگر برای هر j که $\bar{x}_j > 0$ داشته باشیم $\sum_{i=1}^{m} a_{ij}\bar{y}_i = c_j$ و نیز برای هر i که $\bar{y}_i > 0$ داشته باشیم $\sum_{j=1}^{n} a_{ij}\bar{x}_j = b_i$، به بیان دیگر هر گاه $\bar{x}_j > 0$ شرط متناظر آن در دوگان به صورت تساوی ارضا شده باشد و هر گاه $\bar{y}_i > 0$ شرط متناظر آن در مسئله‌ی اولیه به صورت تساوی ارضا شده باشد.

قضیه ۴ (مکمل لنگی). *اگر $\bar{x}$ و $\bar{y}$ به ترتیب دو جواب شدنی برای مسئله‌ی اولیه و دوگان آن باشد، در این صورت $\bar{x}$ و $\bar{y}$ شرایط مکمل لنگی را دارا هستند اگر و تنها اگر هر دو جواب بهینه برای مسئله‌ی متناظر خود باشند.*

اثبات. اگر $\bar{x}$ و $\bar{y}$ جواب‌های بهینه باشند، در این صورت طبق قضیه‌ی دوگانی قوی،

$$
\begin{aligned}
\sum_{i=1}^{m} b_i \bar{y}_i &= \sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} \bar{x}_j) \bar{y}_i \\
&= \sum_{j=1}^{n} \bar{x}_j \sum_{i=1}^{m} a_{ij} \bar{y}_i \\
&= \sum_{j=1}^{n} \bar{x}_j c_j
\end{aligned}
$$

پس می‌توان نوشت:

$$
\sum_{j=1}^{n} (c_j - \sum_{i=1}^{m} a_{ij}\bar{y}_i)\bar{x}_j = 0
$$

[۱۸] strong duality
[۱۹] simplex
[۲۰] complementary slackness conditions

چون می‌دانیم برای هر j بین ۱ تا m هم $0 \geq \bar{x}_j$ و هم $c_j \geq \sum_{i=1}^m a_{ij}\bar{y}_i$ پس برای هر j داریم: $0 \geq (c_j - \sum_{i=1}^m a_{ij}\bar{y}_i)\bar{x}_j$ پس در مجموع می‌توان نوشت:

$$(c_j - \sum_{i=1}^m a_{ij}\bar{y}_i)\bar{x}_j = 0 \quad j = 1, \ldots, m.$$

در نتیجه برای هر j داریم:

$$\bar{x}_j > 0 \to c_j = \sum_{i=1}^m a_{ij}\bar{y}_i$$

به همین ترتیب می‌توان ثابت کرد برای هر i بین ۱ تا m داریم:

$$\bar{y}_i > 0 \to b_i = \sum_{j=1}^n a_{ij}\bar{x}_j$$

به عکس اگر $\bar{x}$ و $\bar{y}$ در شرایط مکمل لنگی صدق کنند می‌توانیم بنویسیم:

$$\bar{x}_j > 0 \to c_j = \sum_{i=1}^m a_{ij}\bar{y}_i$$
$$\bar{y}_i > 0 \to b_i = \sum_{j=1}^n a_{ij}\bar{x}_j$$

در نتیجه با استفده از استدلال بالا در جهت معکوس می‌توان نوشت:

$$\begin{aligned}\sum_{i=1}^m b_i\bar{y}_i &= \sum_{i=1}^m (\sum_{j=1}^n a_{ij}\bar{x}_j)\bar{y}_i \\ &= \sum_{j=1}^n \bar{x}_j \sum_{i=1}^m a_{ij}\bar{y}_i \\ &= \sum_{j=1}^n \bar{x}_j c_j\end{aligned}$$

چون طبق قضیه‌ی دوگانی ضعیف می‌دانیم که $\sum_{i=1}^m b_i\bar{y}_i \leq \sum_{j=1}^n \bar{x}_j c_j$ و در مسئله‌ی اول به دنبال کمینه کردن و در دوگان به دنبال بیشنه ساختن هستیم پس $\bar{x}$ و $\bar{y}$ به ناچار جواب‌های بهینه برای مسائل متناظر به خود هستند. □

۴ حل LP

سیمپلکس قدیمی‌ترین، معروفترین و پرکاربردترین الگوریتم برای حل مسائل LP است که در صورت وجود جواب بهینه، یک جواب پایه‌ای بهینه برای مسئله را پیدا می‌کند. اگر چه زمان اجرای آن(با تحلیل بدترین حالت) به صورت نمایی است، اما در عمل و در حل مسائل روزمره بسیار سریع عمل می‌کند و با حرکت بین جواب‌های پایه در جهت بهبود تابع هدف به جواب بهینه می‌رسد. با این وجود در برخی موارد در طول درس ما نیاز به حل LPهایی داریم که تعداد شرطها تابعی توانی از طول ورودی مسئله است که در این حالت‌ها دیگر نمی‌توانیم از سیمپلکس استفاده کنیم. در بعضی از این گونه موارد به کمک الگوریتم *الیپسوید*[۲۱] می‌توان این مسائل را حل کرد. در ادامه به برخی نکات حائز اهمیت در این باره اشاره می‌کنیم.

ابتدا لازم است که یک *اوراکل جداکننده*[۲۲] را تعریف کنیم. یک اوراکل جداکننده یک x به عنوان پیشنهادی برای جوابی شدنی از مسئله‌ی LP می‌گیرد و بررسی می‌کند که آیا x واقعا یک جواب شدنی است یا خیر و اگر جواب شدنی نبود یکی از قیودی که x آن را نقض می‌کند را به عنوان خروجی باز می‌گرداند. دقت کنید که چون ورودی تنها x است پس اگر یک اوراکل جداکننده داشته باشیم که زمان خطی دارد، این خطی بودن تنها بر حسب n است و نه m.

حال اگر فرض کنیم که تعداد بیت‌های لازم برای نمایش هر کدام از قیدها کران بالای ثابتی مستقل از m و n مثل ϕ دارد، در این صورت الگوریتم الیپسوید با در دست داشتن یک اوراکل جداکننده که زمان خطی داشته باشد، می‌تواند در زمانی خطی برحسب n و ϕ جوابی بهینه برای مسئله پیدا کند که در صورت نیاز می‌توان یک جواب بهینه‌ی پایه‌ای را نیز از روی آن به دست آورد. البته واقعیت آن است که در عمل معمولا از الگوریتم الیپسوید استفاده نمی‌شود، اما به هر حال برای انجام محاسبات در فصل‌های بعد می‌توانیم از فرض وجود آن استفاده کنیم.

[۲۱] ellipsoid
[۲۲] separation oracle

۵ مسئله‌ی پوشش مجموعه‌ای

حال که کلیات برنامه‌ریزی خطی را مرور کردیم، وقت آن است که محتوای اصلی درس را شروع کنیم. در انتهای این جلسه فقط به صورت بندی یک مسئله در قالب برنامه‌ریزی خطی می‌پردازیم و در جلسه‌ی بعد خواهیم دید که این صورت‌بندی چگونه به پیدا کردن یک الگوریتم تقریبی برای مسئله‌ی اصلی کمک خواهد کرد. در *مسئله‌ی پوشش مجموعه‌ای*[۲۳] به ما مجموعه‌ی $E = \{e_1, \dots, e_n\}$ و زیرمجموعه‌هایی از آن به نام‌های $S_1, S_2, \dots, S_m$ و وزن‌هایی متناظر این زیرمجموعه‌ها به نام‌های $w_1, w_2, \dots, w_m$ داده شده‌اند. هدف آن است که $I \subseteq \{1, \dots, m\}$ را بیابیم به گونه‌ای که داشته باشیم $\bigcup_{j\in I} S_j = E$ و مقدار $\sum_{j\in I} w_j$ کمینه شود. مسئله‌ی پوشش مجموعه‌ای شامل مصداق‌های مختلفی است که تحت عنوان‌های دیگری صورت‌بندی شده‌اند. مثلا یک حالت خاص آن، *مسئله‌ی پوشش رأسی*[۲۴] است. در این مسئله با داشتن گراف وزن دار $G = (V, E)$ به دنبال $C \subseteq E$ هستیم به گونه ای که تمام رأس‌های گراف پوشیده شوند و مجموع وزن یال‌های درون Cکمینه شود.

حال می‌خواهیم مسئله‌ی پوشش مجموعه‌ای را در قالب یک LP صورت بندی کنیم. ابتدا برای هر $j \in [m]$ که در آن $[m] = \{1, \dots, m\}$ یک متغیر x_j تعریف می‌کنیم که بیان می‌کند S_j انتخاب شده است یا خیر، یعنی $x_j = 1$ اگر $j \in I$ و $x_j = 0$ اگر $j \notin I$. با این تعریف تابع هدف ما به راحتی قابل بیان است یعنی ما به دنبال کمینه کردن تابع $\sum_{j=1}^{m} w_j x_j$ هستیم. در مرحله‌ی بعد باید قیدها را تعریف کنیم. ابتدا دقت داریم که چون x_jها باید صحیح باشند پس عملا ما با پیدا کردن قیود مناسب برای پوشش دادن مجموعه‌ی E به یک IP دست پیدا خواهیم کرد. شرط لازم و کافی برای آن‌که عنصر e_i در E پوشیده شده باشند را می‌توان به صورت قید زیر نوشت:

$$\sum_{j:e_i\in S_j} x_j \geq 1$$

چرا که حتما e_i در حداقل یکی از S_jها هست. همان‌طور که پیش از این گفته شد، ما در حالت عمومی قادر به حل مسائل IP به صورت کارا نیستیم لذا باید با آزادسازی به یک مسئله‌ی LP برسیم و امید داشته باشیم که جواب مسئله‌ی جدید خیلی از مسئله‌ی اصلی دور نباشد. مثلا در اینجا می‌توانیم به جای شرط $x \in \{0, 1\}$ شرط $0 \leq x_j \leq 1$ را اضافه کنیم. البته دقت داشته باشید که در این مثال اگر در یک جواب شدنی خاص اگر $x_k > 1$ باشد مثلا $x_k = 1 + \epsilon$ در اینصورت چون همه‌ی x_jها و w_jها مثبت هستند، با کم کردن مقدار ϵ از x_j همه‌ی قیود ارضا شده باقی می‌مانند و در عین حال تابع هدف به اندازه‌ی $w_k\epsilon$ کاهش می‌یابد پس هیچ جواب اپتیممی نیست که در آن این اتفاق بیافتد. با توجه به این موضوع می‌توان LP زیر را برای این مسئله نوشت:

$$\text{minimize} \quad \sum_{j=1}^{m} w_j x_j \tag{۷}$$

$$\text{subject to} \quad \sum_{j:e_i\in S_j} x_j \geq 1, \quad i = 1, \dots, n \tag{۸}$$

$$x_j \geq 0, \qquad j = 1, \dots, m \tag{۹}$$

اگر مقدار جواب مسئله‌ی اصلی را OPT ، جواب IP به دست آمده را Z^*_{IP} و جواب LP حاصله را Z^*_{LP} بنامیم، توجه داریم که در اینجا چون مجموعه‌ی جواب‌های شدنی LP شامل مجموعه‌ی جواب‌های شدنی IP است و مسئله‌ی IP دقیقا همان مسئله‌ی اصلی است پس $Z^*_{\text{LP}} \leq z^*_{\text{IP}} = \text{OPT}$. در جلسه‌ی بعد با استفاده از جواب این LP به ارائه‌ی الگوریتم‌های تقریبی برای مسئله‌ی پوشش مجموعه‌ای خواهیم پرداخت.

۶ مراجع

۱. کتاب اصلی درس

۲. جزوه‌های[۲۵] شماره‌ی ۵ و ۷ مربوط به درس «نظریه‌ی الگوریتمی بازی‌ها»ی دکتر کوشا اعتصامی در دانشگاه ادینبرو[۲۶].

[۲۳] set cover problem
[۲۴] vertex cover problem
[۲۵] lecture note

[۲۶] یا ادینبورگ