دانشکده علوم ریاضی
دانشگاه صنعتی شریف

کاربرد برنامه‌ریزی ریاضی در تولید الگوریتم‌های تقریبی

محمدهادی فروغمنداعرابی

پاییز ۱۳۹۶

مقدمه‌ای بر الگوریتم‌های تقریبی

جلسه دوم

نگارنده: حامد واسعی

در این جلسه ابتدا به چند تعریف اولیه در مورد الگوریتم‌های تقریبی می‌پردازیم سپس برخی از محوری‌ترین و مهم‌ترین ایده‌هایی را که در طول درس با آن‌ها سر و کار داریم را در حل یک مسئله‌ی ساده معرفی می‌کنیم.

۱ مروری بر مباحث گذشته

در جلسه‌ی قبل ما به مرور برنامه‌ریزی خطی پرداختیم. علاوه بر آن مسئله‌ی پوشش مجموعه‌ای را نیز هم در قالب یک برنامه‌ریزی صحیح مدل کردیم. پس از آن با آرام‌سازی به یک مسئله‌ی LP دست پیدا کردیم. در این جلسه تمرکز ما بر روی این مسئله خواهد بود و لذا برنامه‌ریزی خطی مربوطه را در زیر تکرار می‌کنیم:

$$\begin{aligned}
\text{minimize} \quad & \sum_{j=1}^{m} w_j x_j && (1)\\
\text{subject to} \quad & \sum_{j: e_i \in S_j} x_j \geq 1, \quad i = 1, \dots, n && (2)\\
& x_j \geq 0, \qquad j = 1, \dots, m && (3)
\end{aligned}$$

خواننده می‌تواند برای دانستن جزئیات و تعابیر مربوطه به جزوه‌ی جلسه‌ی اول مراجعه کند.

۲ تعاریف ابتدایی

مسائل بسیاری در *بهینه‌سازی ترکیبیاتی*[۱] وجود دارند که می‌دانیم NP‌ـسخت[۲] هستند. بنابراین اگر $\mathrm{P} \neq \mathrm{NP}$ باشد راه حل کارایی برای حل این مسائل وجو نخواهد داشت. در مواجهه با این مشکل یکی از روش‌ها آن است که به جای آنکه دنبال جواب بهینه برای مسئله باشیم، جوابی بیابیم که تا اندازه‌ای به جواب بهینه نزدیک باشد که البته این نزدیکی را با مقدار تابع هدف می‌سنجیم. این روش منجر به پیدایش الگوریتم‌های تقریبی برای حل مسائل بهینه‌سازی می‌شود.

تعریف ۱. *الگوریتم $\mathcal{A}$ یک الگوریتم αـتقریب*[۳] *برای یک مسئله‌ی بهینه‌سازی است اگر زمان چند جمله‌ای داشته باشد و برای هر نمونه*[۴] *از آن مسئله مثل η، مقدار تابع هدف در جواب تولید شده توسط الگوریتم یعنی $\mathcal{A}(\eta)$ در مقایسه با OPT که مقدار بهینه‌ی تابع هدف است، خاصیت زیر را داشته باشد:*

$$\begin{cases} \mathrm{OPT} \leq \mathcal{A}(\eta) \leq \alpha\mathrm{OPT}, & \textit{if } \alpha > ۱; \\ \alpha\mathrm{OPT} \leq \mathcal{A}(\eta) \leq \mathrm{OPT}, & \textit{if } \alpha \leq ۱. \end{cases}$$

α را *نسبت تقریب*[۵] *ضریب تقریب*[۶] گویند. طبق تعریف واضح است که اگر $\alpha > ۱$ باشد صورت مسئله کمینه‌سازی و یافتن جوابی است که حداکثر با نسبت α بزرگتر از جواب بهینه باشد و اگر $\alpha < ۱$ باشد صورت مسئله بیشینه‌سازی و یافتن جوابی است که حداکثر با نسبت α کوچکتر از جواب بهینه باشد.

مسائل مختلف از نظر تقریب پذیری با یکدیگر متفاوت هستند، برای مثال در مورد برخی مسائل فکر می‌کنیم که از حد مشخصی بهتر نمی‌توان با هیچ الگوریتمی آن‌ها را تقریب زد و در مقابل برخی مسائل را می‌شود با هر ضریب تقریبی، تقریب زد.

تعریف ۲. *مسئله‌ی $\mathcal{P}$ یک $PTAS$*[۷] *دارد اگر خانواده‌ای از الگوریتم‌ها مثل $\{\mathcal{A}_\epsilon\}$ داشته باشیم به طوری که برای هر $\epsilon > ۰$، $\mathcal{A}_\epsilon$ یک $(۱+\epsilon)$ـتقریب (اگر $\mathcal{P}$ کمینه‌سازی باشد) یا $(۱-\epsilon)$ـتقریب (اگر $\mathcal{P}$ بیشینه‌سازی باشد) برای $\mathcal{P}$ باشد.*

در جلسات آتی مسائلی را خواهیم دید که برای آن‌ها PTAS وجود دارد و نیز مسائلی که ثابت می‌شود هیچ PTASی برای آن‌ها وجود ندارد مگر آن‌که $\mathrm{P} = \mathrm{NP}$ باشد.

۳ گردکردن قطعی

در این بخش با استفاده از *گردکردن قطعی*[۸] یک الگوریتم تقریبی برای مسئله‌ی پوشش مجموعه ای ارائه می‌کنیم. فرض کنید ما مسئله‌ی برنامه‌ریزی خطی که از آرام‌سازی مسئله‌ی پوشش مجموعه‌ای به دست آمده است را حل کردیم و جواب x^* به دست آمده است. حال باید با استفاده از آن یه یک جواب برای پوشش مجموعه‌ها برسیم. برای این کار با داشتن x^* مجموعه‌ی S_j را انتخاب می‌کنیم اگر و تنها اگر $x_j^* \geq ۱/f$ که f در آن بیشینه‌ی تعداد دفعاتی است که یک عنصر در زیرمجموعه‌های $S_۱$ تا S_m ظاهر می‌شود، به بیان دقیق‌تر اگر تعریف کنیم $f_i = |\{j : e_i \in S_j\}|$ برای هر $۱ \leq i \leq n$ در این صورت $f = \max_i f_i$. مجموعه‌ی اندیس زیرمجموعه‌های انتخاب شده را با $I \subseteq [m]$ نمایش می‌دهیم. برای هر زیر مجموعه‌ی S_j تعریف می‌کنیم $\hat{x}_j = ۱$ اگر $x_j^* \geq ۱/f$ و $\hat{x}_j = ۰$ اگر $x_j^* < ۱/f$.

لم ۳. *زیرمجموعه‌های متناظر با اعضای I یک پوشش مجموعه‌ای برای E هستند، به عبارت دیگر:*

$$\bigcup_{j \in I} S_j = E$$

اثبات. عنصر دلخواه e_i را در نظر بگیرید. چون x^* یک جواب شدنی بهینه برای LP اولیه است پس $\sum_{j:e_i\in S_j} x_j^* \geq ۱$. با توجه به تعریف می‌دانیم تعداد جملات این مجموع f_iاست و $f_i \leq f$ در نتیجه حداقل یکی از x_j^*ها برای $j \in j : e_j \in S_j$ بزرگتر یا مساوی $۱/f$ است. اگر

[۱] combinatorial optimization also known as discrete optimization
[۲] NP-hard
[۳] α-approximation
[۴] instance
[۵] approximation ratio
[۶] approximation factor
[۷] polynomial-time approximation scheme
[۸] determinisitic rounding

اندیس متناظر آن k باشد می‌دانیم که $\hat{x}_k = ۱$ و $k \in I$ و این یعنی آن که S_k انتخاب شده است و چون $k \in j : e_j \in S_j$ پس e_i پوشیده شده است. در نتیجه همه‌ی عناصر E پوشیده می‌شوند.

□

تا اینجای کار ثابت کردیم که با استفاده از I می‌توان E را پوشاند اما ممکن است جواب حاصله وزن بسیار بیشتری از جواب بهینه داشته باشد. در ادامه بیان می‌کنیم که وزن جواب به دست آمده در مقایسه با جواب بهینه چه نسبتی دارد.

قضیه ۴. *الگوریتم گرد کردن قطعی که در بالا بیان شد یک الگوریتم fـ تقریب برای مسئله‌ی پوشش مجموعه‌ای است.*

اثبات. ابتدا باید بگوییم که الگوریتم ما زمان خطی دارد. برای این منظور دقت می‌کنیم که حل کردن LP، محاسبه‌ی f_iها، f و $\hat{x}_j$ها همگی در زمان خطی امکان پذیر است و در نتیجه تشکیل I در زمان خطی میسر است. با توجه به لم بالا، تنها کافی است ثابت کنیم:

$$\sum_{j\in I} w_j \le f.\mathrm{OPT}$$

که OPT وزن پوشش مجموعه‌ای بهینه است. طبق تعریف می‌دانیم برای هر $j \in I$ داریم $fx_j^* \ge ۱$. پس با توجه به نامنفی بودن تمام x_jها، w_jها و f می‌توان نوشت:

$$\begin{aligned}
\sum_{j\in I} w_j &\le \sum_{j=۱}^{m} w_j(fx_j^*) \\
&= f\sum_{j=۱}^{m} w_j x_j^* \\
&= f.Z^*_{\mathrm{LP}} \\
&= f.\mathrm{OPT}
\end{aligned}$$

□

طبق این قضیه در مسئله‌ی پوشش رأسی، الگوریتم گردکردن قطعی معرفی شده یک الگوریتم ۲ـ تقریب است چرا که هر یال دقیقا توسط دو رأس پوشیده شده‌است پس برای هر i می‌دانیم $f_i = ۲$ و در نتیجه $f = ۲$.

دقت کنید که پس از اجرای الگوریتم و به دست آوردن I می‌توانیم نسبت $\alpha = \sum_{j\in i} / Z^*_{\mathrm{LP}}$ را به راحتی محاسبه کنیم و بر این اساس می‌دانیم که $\sum_{j\in I} \le \alpha Z^*_{\mathrm{LP}} \le \alpha \mathrm{OPT}$. با توجه به اثبات بالا می‌دانیم $\alpha \le f$ و ممکن است به چون محاسبه‌ی α با داشتن I و جواب x^* ساده است، ممکن است گارانتی بهتری برای تقریب خود در هر مورد خاص داشته باشیم.

۴ گرد کردن جواب دوگان

خیلی از اوقات توجه به دوگان مسئله‌ی LP به دست آمده در به دست آوردن الگوریتم تقریبی کارگشا خواهد بود. فرض کنید هر عنصر e_i به اندازه‌ی $y_i \ge ۰$ هزینه در بر دارد تا به وسیله‌ی یک پوشش مجموعه‌ای پوشیده شود. طبیعی است که بعضی از عناصر بتوانند توسط زیرمجموعه‌هایی با وزن کم پوشیده شوند و برخی از اعضا نیاز به زیرمجموعه‌های با وزن بالا برای پوشیده شدن داشته باشند. می‌خواهیم به گونه‌ای به عناصر هزینه تحمیل کنیم که عناصر نوع اول کمتر و عناصر نوع دوم هزینه‌ی بیشتری بپردازند. معقول است که مجموع هزینه‌ی اعضای یک زیرمجموعه از وزن آن بیشتر نباشد، زیرا ما با پرداخت هزینه‌ی وزن آن زیرمجموعه، تمامی اعضای آن را پوشانده‌ایم! پس متناظر با هر زیرمجموعه‌ی S_j یک قید به صورت زیر تعریف می‌کنیم:

$$\sum_{i:e_i\in S_j} y_i \le w_j$$

بیشترین مقدار هزینه‌ای که می‌توان از مجموع عناصر دریافت کرد به صورت برنامه‌ریزی خطی زیر بیان می‌شود:

$$\text{maximize} \quad \sum_{i=1}^{n} y_i \tag{4}$$

$$\text{subject to} \quad \sum_{i:e_i\in S_j} y_i \le w_j, \quad j=1,\dots,m \tag{5}$$

$$y_i \ge 0, \qquad i=1,\dots,n \tag{6}$$

تمرین ۱. *با توجه به تعریف دوگان که در جلسه‌ی اول آورده شده است تحقیق کنید که LP بالا صورت‌بندی دوگان مسئله‌ی اولیه‌ی آرام‌سازی شده‌ی پوشش مجموعه‌ای است.*

با توجه به قضیه‌ی دوگانی ضعیف می‌دانیم برای هر جواب شدنی دوگان مثل y داریم $\sum_{i=1}^{n} y_i \le z^*_{\text{LP}} \le OPT$. حال با کمک دوگان الگوریتم تقریبی دیگری برای مسئله‌ی پوشش مجموعه‌ای ارائه می‌دهیم. فرض کنید y^* یک جواب بهینه برای دوگان باشد. حال تمام زیرمجموعه‌هایی که قیود متناظر آن‌ها در دوگان فعال است را انتخاب می‌کنیم. به بیان دیگر مجموعه‌ی اندیس زیرمجموعه‌های انتخاب شده I' است که شامل jهایی است که $\sum_{i:e_i\in S_j} y^*_i = w_j$. با کمک لم زیر ثابت می‌کنیم که این الگوریتم هم یک fـ‌تقریب است.

لم ۵. *زیرمجموعه‌های متناظر با اعضای I' یک پوشش مجموعه‌ای برای E هستند، به عبارت دیگر:*

$$\bigcup_{j\in I'} S_j = E$$

اثبات. فرض کنید e_k وجود دارد که پوشیده نشده است. در این صورت برای هر زیرمجموعه‌ی S_j که شامل e_k است، داریم:

$$\sum_{i:e_i\in S_j} y^*_i \le w_j \tag{7}$$

ϵ را به صورت $\epsilon = \min_{j:e_k\in S_j}(w_j - \sum_{i:e_i\in S_j} y^*_i)$ تعریف می‌کنیم. با توجه به ۷ می‌دانیم $\epsilon > 0$. حال y' را به این صورت تعریف می‌کنیم که $y'_k = y^*_k + \epsilon$ و در بقیه‌ی مؤلفه‌ها y^* و y' با یکدیگر برابرند. در این صورت y' نیز یک جواب شدنی برای دوگان است چرا که برای هر قید متناظر با هر S_j که $e_k \in S_j$ باشد داریم:

$$\sum_{i:e_i\in S_j} y'_i = \epsilon + \sum_{i:e_i\in S_j} y^*_i \le w_j$$

و برای هر S_j که $e_k \notin S_j$ باشد داریم:

$$\sum_{i:e_i\in S_j} y'_i = \sum_{i:e_i\in S_j} y^*_i \le w_j$$

و نیز همه‌ی مؤلفه‌های y' مثبت هستند پس همه‌ی قیود دوگان ارضا می‌شوند پس y' یک جواب شدنی است. از سوی دیگر داریم $\sum_{i=1}^{n} y'_i = \epsilon + \sum_{i=1}^{n} y^*_i$ که این موضوع با بهینه بودن y^* تناقض دارد، در نتیجه هیچ عنصر پوشیده نشده‌ای وجود ندارد. □

قضیه ۶. *الگوریتم گردکردن دوگان که در بالا تعریف شد یک fـ‌تقریب برای مسئله‌ی پوشش مجموعه‌ای است.*

اثبات. با توجه به تعریف I' و آنچه از قضیه‌ی دوگانی ضعیف نتیجه گرفتیم، داریم:

$$\begin{aligned}
\sum_{j\in I'} w_j &= \sum_{j\in I'}\sum_{i:e_i\in S_j} y^*_i \\
&= \sum_{i=1}^{n} |\{j\in I' : e_i\in S_j\}| y^*_i \\
&\le \sum_{i=1}^{n} f_i y^*_i \\
&\le f\sum_{i=1}^{n} y^*_i \\
&\le f\text{OPT}
\end{aligned}$$

□

با توجه به این که x^* و y^* جواب‌های بهینه‌ی مسئله‌ی اولیه و دوگان آن هستند پس در شرایط ملکمل لنگی صدق می‌کنند. پس با توجه به قضیه‌ی مکمل لنگی برای هر j که $x_j^* > 0$ باشد، قید متناظر زیرمجموعه‌ی S_j فعال است در نتیجه $j \in I'$. دقت داریم که برای هر j که $x_j^* \geq 1/f$ باشد حتما $x_j^* \geq 0$ پس:

$$j \in I \rightarrow j \in I''$$

در نتیجه $I \subseteq I$ که به این معنی است که وزن پوشش مجموعه‌ای متناظر با الگوریتم دوم همیشه بزرگتر یا مساوی پوشش مجموعه‌ای به دست آمده از الگوریتم بخش قبل است.

۵ روش اولیه‌ـدوگان

یک ضعف هر دو الگوریتمی که در بخش‌های قبل مطرح شدند، آن است که نیاز به حل یک برنامه‌ریزی خطی دارند. اگرچه برنامه‌ریزی خطی به سرعت قابل حل است اما عموما الگوریتم‌های خاص منظوره بسیار سریع‌تر هستند. ایده‌ی اصلی این بخش آن است که به جای حل یک برنامه‌ریزی خطی، یک جواب شدنی مناسب پیدا کند و گردکردن را بر روی آن انجام دهد. پیدا کردن این جواب شدنی بسیار ساده‌تر از حل یک LP است و به الگوریتم‌های بسیار سریع‌تری منجر می‌شود.

در بخش قبل ما از بهینه بودن y^* استفاده‌ی چندانی نکردیم. ویژگی‌های مورد استفاده یکی این بود که $\sum_{i=1}^{n} y_i \leq \mathrm{OPT}$ که این ویژگی برای همه‌ی جواب‌های شدنی برقرار است. ویژگی دیگر این بود که وقتی S_jهای متناظر با قیدهای فعال را در نظر می‌گرفتیم یک پوشش مجموعه‌ای حاصل می‌شد. این دو واقعیت در کنار یکدیگر منجر به اثبات fـتقریب بودن الگوریتم بخش پیش شد. پس اگر یک الگوریتم جواب شدنی برای دوگان به دست دهد که اندیس قیدهای فعال آن تشکیل یک پوشش مجموعه‌ای بدهد، این الگوریتم یک fـتقریب خواهد بود.

حال فرض کنید ما یک جواب شدنی y داریم. تعریف می‌کنیم $T = \{j : \sum_{i:e_i \in S_j} y_i = w_j\}$ که T بیانگر اندیس قیدهایی است که فعال هستند. اگر T یک پوشش مجموعه‌ای باشد که به جواب مورد نظر رسیده‌ایم. در غیر اینصورت یک عنصر مثل e_i وجود خواهد داشت که پوشیده نشده است. طبق اثبات لم ۵ می‌شود y_i را به اندازه‌ی $\epsilon > 0$ افزایش داد به گونه‌ای که جواب شدنی باقی بماند و علاوه بر آن حداقل یک قید متناظر با یک مجموعه‌ی شامل e_i فعال شود. به صورت دقیق‌تر $\epsilon = \min_{j:e_i \in S_j}(w_j - \sum_{k:e_k \in S_j} y_k)$ و $j = \mathrm{argmin}_{j:e_i \in S_j}(w_j - \sum_{k:e_k \in S_j} y_k)$ و در نتیجه قید متناظر با S_j فعال خواهد شد. به این صورت با اضافه شدن j به T، e_i به مجموعه‌ی عناصر پوشیده شده اضافه می‌شود.

با این مقدمه الگوریتم سوم را به این صورت اجرا می‌کنیم که ابتدا y را بردار 0 در نظر می‌گیریم. دقت دارید که به این تقدیر y یک جواب شدنی است. در عین حال هیچ کدام از عناصر نیز به این صورت پوشیده نخواهند بود و در نتیجه $I'' = \emptyset$ که I'' مجموعه‌ی اندیس قیدهای فعال در دوگان با جواب y است. حال تا زمانی که عضو پوشیده نشده‌ای وجود دارد، طبق آنچه گفته شد مؤلفه‌ی متناظر آن را در y بزرگتر می‌کنیم و زیرمجموعه‌ی متناظر با قید فعال شده را به I'' می‌افزاییم. این الگوریتم حتما خاتمه می‌یابد چرا که تعداد عناصر محدود است و هر بار حداقل یکی از عناصر پوشیده می‌شود.

قضیه ۷. *الگوریتم اولیه‌ـدوگان که در بالا تعریف شد یک fـتقریب برای مسئله‌ی پوشش مجموعه‌ای است.*

اثبات. مشابه اثبات قضیه‌ی ۶. □

۶ مراجع

۱. کتاب اصلی درس

۲. صفحه‌ی الگوریتم‌های تقریبی در ویکی‌پدیا