دانشکده علوم ریاضی
دانشگاه صنعتی شریف

کاربرد برنامه‌ریزی ریاضی در تولید الگوریتم‌های تقریبی

محمدهادی فروغمنداعرابی
پاییز ۱۳۹۶

حالتی مقید از مکان‌یابی تسهیلات، مشابه خوشه‌بندی

جلسه ۱۹ام

نگارنده: امیرکسری جلالدوست

۱ مروری بر مباحث گذشته

در جلسه‌ي قبل توانستیم روشی پیدا کنیم که مکان‌یابی تسهیلات بدون ظرفیت را با روش اولیه‌ـ دوگان، با ضریب تقریب ۳ حل کند. روش به این ترتیب بود که متغیرهایی برای یال‌ها و رئوس داشتیم و آنها را مرتبا زیاد میکردیم و با رجوع به دوگان درصورت رسیدن به تساوی رسیدن قیود، آنها را از حالت تساوی به حالت اکید میبردیم.

۲ مساله‌ی k میانه

ساله مشابه همان مکان‌یابی تسهیلات است با این تفاوت که قیمت تسهیلات اهمیتی ندارد و در تابع هدف ظاهر نمی‌شود ولی می‌توانیم حداکثر k تسهیلات تاسیس کنیم. صورت برنامه‌زیری صحیح برای آن به شکل زیر است:

$$\begin{aligned}
minimize \sum_{i=1}^{n} x_{ij}c_{ij} \\
\forall j \to \sum_{i} x_{ij} = 1 \\
\sum_{i} y_i \le k \\
\forall i, j \to x_{ij} \le y_i \\
\forall i, j \to y_i, x_{ij} \in \mathbb{N} \cup \{0\}
\end{aligned}$$

می‌توانیم این برنامه‌ریزی را با وانهش به برنامه‌ریزی خطی تبدیل کنیم و بعد با آوردن قید دوم به تابع هدف یک نسخه‌ی «لاگرانژی» از آن مینویسیم:

$$\begin{aligned}
P: \\
minimize \sum_{i=1}^{n} x_{ij}c_{ij} + \lambda(\sum y_i - k) \\
\forall j \to \sum_{i} x_{ij} = 1 \\
\forall i, j \to x_{ij} \le y_i \\
\forall i, j \to 0 \le y_i, x_{ij} \in \mathbb{R}
\end{aligned}$$

حال می‌توان دید که این صورت برنامه‌ریزی خطی بسیار مشابه مساله‌ی مکان‌یابی تسهیلات بدون ظرفیت با قیمت ثابت λ برای تمام تسهیلات است. تفاوت کوچکی که وجود دارد این است که از تابع هدفش به میزان $k\lambda$ کم شده است. برای این صورت‌بندی می‌توانیم دوگانی بنویسیم:

$$\begin{aligned}
D: \\
maximize \sum_{j} v_j - k\lambda \\
\forall j \to \sum_{i} w_{ij} \le \lambda \\
\forall i, j \to v_j \le w_{ij} + c_i j \\
\forall i, j \to 0 \le v_i, w_{ij} \in \mathbb{R}
\end{aligned}$$

این دو را با روش اولیه دوگان حل می‌کنیم و جواب صحیح $(y^*_\lambda, x^*_\lambda)$ و را بدست آوریم. حال به y^*_λ نگاه می‌کنیم و S_λ را مجموعه تسهیلات خریداری شده با این روش قرار میدهیم.
از راه‌حل اولیه دوگان توانسته‌ایم $(v^*_\lambda, w^*_\lambda)$ را نیز بدست بیاوریم.
برای مجموعه اندیس S متناظر با تسهیلات، تعریف می کنیم:

$$c(S) := \sum_{j \in D} min(c_{ij})$$

به این تعبیر که اگر مجموعه‌ی S از تسهیلات را اختیار کنیم و هر سرویس‌گیرنده به نزدیکترین تسهیلات برود، هزینه چقدر خواهد شد.

با توجه به راه‌حل اولیه دوگان داریم:

$$c(S) + \sum_{i \in S} f_i \leq ۳ \sum_{j \in D} v_j$$

که با توجه به یکی از تمرینات فصل میتوان آن را اندکی قوی تر کرد طوری که:

$$c(S) + ۳ \sum_{i \in S} f_i \leq ۳ \sum_{j \in D} v_j$$

از طرفی در مساله‌ي ما تمامی f_i ها برابر λ هستند و به این ترتیب داریم:

$$c(S) \leq ۳(\sum_{j \in D} v_j - \lambda|S|) = ۳.OPT$$

به این ترتیب، اگر $|S_\lambda| = k$ باشد می‌توانیم همان مجموعه‌ی S_λ را به عنوان جواب مساله با ضریب تقریب ۳ معرفی کنیم. در غیر این صورت باید به دنبال راه‌حل دیگری باشیم.

می‌توان مشاهده کرد که اگر λ را خیلی خیلی کوچک قرار دهیم، جواب LP حاوی تعداد زیادی تسهیلات خواهد بود و اگر λ را خیلی خیلی بزرگ، مثلا به اندازه‌ی جمع تمام c_{ij} ها قرار دهیم، جوابی که از صورت لاگرانژی بدست می‌آوریم هیچ تسهیلاتی نخواهد داشت. حالت مطلوب برای ما آن است که خروجی حالت لاگرانژی دقیقا k تسهیلات اتخاذ کند، چراکه اگر تعداد کمتری اتخاذ کند، نامساوی‌هایمان برقرار نخواهد ماند و اگر بیشتر اتخاذ کند برای مساله‌ی ما بی استفاده است.

ایده این است که به جستجوی دودویی روی مقادیر λ بپردازیم. اما فضا پیوسته است و قادر نیستیم تا ابد ادامه دهیم! پس قرار میگذاریم که تا جایی پیش برویم که $\lambda_۲ - \lambda_۱ \leq \mu$ شود. اگر در این حین به جوابی با اندازه‌ی مطلوب رسیدیم آن را گزارش میکنیم، اگر نه سعی میکنیم μ را طوری انتخاب کنیم که تقریب خوبی بدست دهد.

صرفا اشاره می‌کنیم که زمان اجرای الگوریتم با ثابت فرض کردن μ چندجمله‌ای خواهد بود.(چرا؟)

حال دو مجموعه‌ی $S_۱$ و $S_۲$ متناظر با $\lambda_۱$ و $\lambda_۲$ در اختیار داریم. سعی داریم آن دو را ترکیب کنیم. یک ترکیب محدب خیالی مسازیم:

$$\alpha_۱ = \frac{k - |S_۲|}{|S_۱| - |S_۲|}$$

$$\alpha_۲ = \frac{|S_۱| - k}{|S_۱| - |S_۲|}$$

$$S = \alpha_۱ S_۱ + \alpha_۲ S_۲$$

اما همه می‌دانند که نمی‌توان دو مجموعه را به این شکل ترکیب کرد. اما می‌توان جواب LP ها را با هم ترکیب محدب کرد.

$$(v, w) = \alpha_۱(v^۱, w^۱) + \alpha_۲(v^۲, w^۲)$$

به این ترتیب این جواب جدید یک جواب شدنی برای $\lambda_۲$ است.(چرا؟)

حال نشان میدهیم که اگر با همین ضرایب هزینه‌ها را ترکیب کنیم باز هم تقریب خوبی است:

$$\begin{aligned} c(S_۱) &\leq ۳(\sum_{j \in D} v_j^۱ - \lambda_۱|S_۱|) \\ &= ۳(\sum_{j \in D} v_j^۱ - (\lambda_۱ + \lambda_۲ - \lambda_۲)|S_۱|) \\ &= ۳(\sum_{j \in D} v_j^۱ - \lambda_۲|S_۱|) + (\lambda_۲ - \lambda_۱)|S_۱| \\ &= ۳(\sum_{j \in D} v_j^۱ - \lambda_۲|S_۱|) + \mu|S_۱| \end{aligned}$$

حال می‌توان مشاهده کرد که اگر $\mu = \frac{\epsilon c_{min}}{|F|}$ باشد که ϵ خطای تقریبمان باشد، عبارت بالا کمتر یا مساوی $۳(\sum_{j \in D} v_j^۱ - \lambda_۲|S_۱|) + \epsilon OPT$ خواهد بود.

حال برای ترکیب محدب هزینه‌ها داریم:

$$\alpha_1 c(S_1) + \alpha_2 c(S_2) \leq 3\alpha_1(\sum_{j\in D} v_j^1 - \lambda_2|S_1|) + \alpha_1 \epsilon OPT$$
$$+3\alpha_2(\sum_{j\in D} v_j^2 - \lambda_2|S_2|)$$

که با توجه به تعریف α_1 و α_2 و (v, w) می‌توان نتیجه گرفت:

$$\alpha_1 c(S_1) + \alpha_2 c(S_2) \leq 3\alpha_1(\sum_{j\in D} v_j - \lambda_2 k) + \alpha_1 \epsilon OPT$$
$$\leq (3+\epsilon)OPT$$

اکنون الگوریتم ما با دو حالت روبرو است:

$$\frac{1}{2} \leq \alpha_2$$
$$\frac{1}{2} > \alpha_2$$

که در حالت اول اگر S_2 را به عنوان جواب برگردانیم پاسخ قابل قبول با هزینه‌ی $2(3+\epsilon)$ است که تقریبا ۶ـتقریب است.
در حالت دوم اوضاع کمی پیچیده‌تر است: تعریف می‌کنیم:

$$c(j, S) = min_{i\in S} c_{ij}$$

که با این تعریف داریم:

$$c(S) = \sum_{j\in D} c(j, S)$$

می‌دانیم $|S_2| < |S_1|$ پس سعی میکنیم با توجه به S_2 تعداد k تا عضو از S_1 انتخاب کنیم. به ازای هر اندیس مثل i در S_2 اندیسی مثل h در S_1 انتخاب می‌کنیم که c_{hi} برای مجموعه اندیس‌های S_1 کمینه مقدار را بگیرد. اگر تعدادی کمتر از $|S_2|$ عضو انتخاب شد تعدادی عضو الکی از S_1 به آنها اضافه میکنیم و بعد هر $k - |S_2|$ عضو باقیمانده را به تصادف و به احتمال برابر از اعضای باقیمانده‌ی S_1 برمیداریم. نشان میدهیم امید هزینه با این حرکت کمتر یا مساوی $2(3_\epsilon)OPT$ است.

فرض کنید $1 \in S_1$ میان اعضای S_1 و $2 \in S_2$ میان اعضای S_2 نزدیکترین تسهیلات به سرویس‌گیرنده‌ی j هستند. همچنین فرض کنید نزدیکترین تسهیلات به ۲ اندیس $i \in S_1$ باشد.

$$P[1 \in S] = \frac{k - |S_2|}{|S_1| - |S_2|} = \alpha_1$$
$$1 - \alpha_1 = \alpha_2$$
$$E[c(j, S)] = \alpha_1 c(j, S_1) + \alpha_2 c_{ij}$$
$$c_{ij} \leq c_{2j} + c_{2i}$$
$$= c(j, S_2) + c_{2i}$$
$$\leq c(j, S_2) + c_{12}$$
$$\leq c(j, S_2) + (c(j, S_2) + c(j, S_1))$$
$$= 2c(j, S_2) + c(j, S_1)$$
$$\Rightarrow E[c(j, S)] = \alpha_1 c(j, S_1) + \alpha_2(2c(j, S_2) + c(j, S_1))$$
$$= c(j, S_1) + 2\alpha_2 c(j, S_2)$$

تمامی نامساوی‌های بالا از نامساوی مثلثی برای فواصل حاصل میشوند.
چون در این شرایط $\alpha_1 > \frac{۱}{۲} \rightarrow \alpha_۲ \leq \frac{۱}{۲}$ است. عبارت بالا حداکثر می‌تواند $۲(\alpha_۱ c(j, S_۱) + \alpha_۲ c(j, S_۲)) \leq ۲(۳ + \epsilon)OPT$ باشد و نهایتا یک تقریبا ۶ـ تقریب حاصل می‌شود.

می‌توان این الگوریتم را غیرتصادفی کرد که این نکته در یکی از تمرین‌ها آمده است. همچنین در فصل ۹ خواهیم دید که یک $۲ + \epsilon$ تقریب برای مکان‌یابی تسهیلات بدون ظرفیت وجود دارد که اتفاقا قضایای مورد نیاز ما را هم ارضا می‌کند. با استفاده از آن می‌توانیم به یک تقریبا ۴ـ تقریب برسیم.