دانشکده علوم ریاضی
دانشگاه صنعتی شریف

کاربرد برنامه‌ریزی ریاضی در تولید الگوریتم‌های تقریبی

محمدهادی فروغمنداعرابی

پاییز ۱۳۹۶

شار صحیح چند منظوره و SDP

جلسه‌ی یازدهم

نگارنده: میلاد برزگر

۱ گرد کردن تصادفی برنامه‌ریزی خطی

در این بخش مساله‌ای در مورد شار صحیح چند منظوره مطرح می‌کنیم و با گرد کردن تصادفی برنامه‌ریزی خطی آرام الگوریتمی تقریبی برای حل آن تولید می‌کنیم.

۱.۱ شار صحیح چند منظوره[۱]

در مساله‌ی شار صحیح چند منظوره گراف $G=(V,E)$ و k جفت $s_i,t_i \in V$ از رئوس داده شده است. هدف انتخاب یک مسیر بین s_i و t_i به ازای هر $i \in [k]$ است طوری که بیشینه تعداد مسیرهایی که شامل هر یال هستند کمینه شود. به بیان دیگر اگر هر s_i مبدا و t_i مقصد متناظرش باشد باشد، می‌خواهیم از هر مبدا به مقصد متناظرش شار (صحیح) واحد بفرستیم طوری که بیشینه شار عبوری از یال‌ها کمینه شود. مجموعه‌ی مسیرهای بین s_i و t_i را با $\mathcal{P}_i$ نشان دهید. فرض کنید به ازای هر $P \in \mathcal{P}_i$ متغیر $x_P \in \{۰,۱\}$ نشان دهنده‌ی انتخاب مسیر P باشد. در این صورت انتخاب دقیقاً یک مسیر بین s_i و t_i به معنی $\sum_{P\in\mathcal{P}_i} x_P = ۱$ است. همچنین تعداد مسیرهای شامل یال e برابر است با $\sum_{e\in P} x_p$. فرض کنید W حداکثر شار عبوری از یال‌ها باشد. در این صورت به ازای هر $e \in E$ داریم $\sum_{e\in P} x_p \le W$ و W کوچک‌ترین عدد با این خاصیت است. در نتیجه برنامه‌ریزی صحیح زیر مساله‌ی بالا را به طور دقیق مدل می‌کند

$$
\begin{aligned}
\text{minimize} \quad & W \\
\text{subject to} \quad & \sum_{e\in P} x_P \le W && ,\forall e \in E \\
& \sum_{P\in\mathcal{P}_i} x_P = ۱ && ,\forall i \in [k] \\
& x_P \in \{۰,۱\} && ,\forall P \in \bigcup_i \mathcal{P}_i .
\end{aligned}
$$

[۱] Integer Multicommodity Flow

برنامه‌ریزی خطی آرام شده به صورت زیر است

$$\begin{aligned}
\text{minimize} \quad & W \\
\text{subject to} \quad & \sum_{e\in P} x_P \le W \qquad , \forall e \in E \\
& \sum_{P\in\mathcal{P}_i} x_P = ۱ \qquad , \forall i \in [k] \\
& x_P \ge ۰ \qquad\qquad , \forall P \in \bigcup_i \mathcal{P}_i.
\end{aligned}$$

الگوریتم ۱. الف) $\mathrm{LP} \to (x^*, W^*)$ را حل کن؛

ب) به طور مستقل به ازای هر i عضوی از $\mathcal{P}_i$ با توزیع x^*_P انتخاب کن ($\mathbb{P}(P \text{ انتخاب}) = x^*_P$) (توجه کنید که به دلیل شرط دوم x^*_P یک توزیع احتمال است).

قضیه ۱. فرض کنید $X_۱, \dots, X_n$ متغیرهای برنولی مستقل باشند و $X = X_۱ + \dots + X_n$. در این صورت به ازای هر U $\mathbb{E}[X] \le U$ و $۰ < \delta \le ۱$ داریم

$$\mathbb{P}(X \ge (۱+\delta)U) < e^{-U\delta^۲/۳}.$$

برای اثبات و توضیحات بیش‌تر در مورد قضیه‌ی فوق به کتاب مراجعه شود.

تحلیل الگوریتم ۱. قرار دهید $y_e = \sum_{e\in P} x_P$. در نتیجه $W = \max_{e\in E} y_e$. با توجه به این که $\mathbb{E}[x_P] = x^*_P$ داریم $\mathbb{E}[y_e] = W^*$. فرض کنید $X^i_e \in \{۰,۱\}$ نشان دهنده‌ی این باشد که مسیر بین s_i و t_i شامل e هست یا نه. $X^۱_e, \dots, X^k_e$ مستقل‌اند و $y_e = \sum_{i\in[k]} X^i_e$. در نتیجه طبق قضیه‌ی ۱ به ازای هر $۰ < \delta \le ۱$ داریم

$$\forall e \in E: \ \mathbb{P}(y_e > (۱+\delta)W^*) < e^{-W^*\delta^۲/۳}.$$

در نتیجه

$$\begin{aligned}
\mathbb{P}(\exists e \in E : y_e > (۱+\delta)W^*) &\le \sum_{e\in E} \mathbb{P}(y_e > (۱+\delta)W^*) \\
&< n^۲ e^{-W^*\delta^۲/۳} \\
&= e^{۲\ln n - W^*\delta^۲/۳}.
\end{aligned}$$

در نتیجه

$$\begin{aligned}
\mathbb{P}(W \le (۱+\delta)W^*) &= \mathbb{P}(\max_{e\in E} y_e \le (۱+\delta)W^*) \\
&= \mathbb{P}(\exists e \in E : y_e > (۱+\delta)W^*) \\
&\ge ۱ - e^{۲\ln n - W^*\delta^۲/۳}.
\end{aligned}$$

اگر $W^* \ge ۹\ln n$، با قرار دادن $\delta = ۳\sqrt{\ln n / W^*}$ در رابطه‌ی فوق نتیجه می‌شود الگوریتم ۱ با احتمال زیاد ۲-تقریب است (شرط $W^* \ge ۹\ln n$ برای این لازم بود که δ باید در بازه‌ی $(۰,۱]$ باشد). اگر $۱ \le W^* \le ۹\ln n$، با تکرار محاسبات فوق با $\ln n$ به جای W^* و $\delta = ۱$ نتیجه می‌شود الگوریتم ۱ با احتمال زیاد $O(\ln n)$-تقریب است.

۲ برنامه‌ریزی نیمه معین[۲]

تعریف ۱ (مثبت نیمه معین). ماتریس متقارن $X \in \mathbb{R}^{n\times n}$ را مثبت نیمه معین می‌گوییم و با $X \succcurlyeq ۰$ نشان می‌دهیم هرگاه یکی از شرایط معادل زیر برقرار باشد:

۱. به ازای هر بردار $v \in \mathbb{R}^n$ داشته باشیم $v^T X v \ge ۰$؛

۲. ماتریس $V \in \mathbb{R}^{m\times n}$ با $m \le n$ وجود داشته باشد به طوری که $X = V^T V$؛

۳. بردارهای $w_۱, \dots, w_n \in \mathbb{R}^n$ و اعداد نامنفی $\lambda_۱, \dots, \lambda_n$ وجود داشته باشد به طوری که $X = \lambda_۱ w_۱ w_۱^T + \dots + \lambda_n w_n w_n^T$؛

[۲] Semi-Definite Programming (SDP)

۴. همه‌ی مقادیر ویژه‌ی X نامنفی باشند.

تعریف ۲ (SDP، برنامه‌ریزی نیمه معین). فرض کنید $X = [x_{i,j}]$ ماتریس متغیرها باشد. برنامه‌ریزی نیمه معین به صورت زیر است

$$\begin{aligned} \max/\min \quad & \sum_{i,j} c_{i,j} x_{i,j} \\ \text{subject to} \quad & \sum_{i,j} a_{i,j,k} x_{i,j} = b_k \quad , \forall k \\ & X \succcurlyeq 0. \end{aligned}$$

قرار دهید $C = [c_{i,j}]$ و به ازای هر k، $A_k = [a_{i,j,k}]$. در این صورت SDP را می‌توان به صورت زیر نوشت

$$\begin{aligned} \max/\min \quad & \mathrm{tr}(C^T X) \\ \text{subject to} \quad & \mathrm{tr}(A_k^T X) = b_k \quad , \forall k \\ & X \succcurlyeq 0. \end{aligned}$$

تعریف ۳ (VP، برنامه‌ریزی برداری). برنامه‌ریزی برداری به صورت زیر است

$$\begin{aligned} \max/\min \quad & \sum_{i,j} c_{i,j} \langle v_i, v_j \rangle \\ \text{subject to} \quad & \sum_{i,j} a_{i,j,k} \langle v_i, v_j \rangle = b_k \quad , \forall k \\ & v_i \in \mathbb{R}^n \qquad\qquad , i = 1, \dots, n. \end{aligned}$$

نشان می‌دهیم برنامه‌ریزی نیمه معین و برنامه‌ریزی برداری معادل‌اند. فرض کنید v^* جواب VP باشد. قرار دهید $x_{i,j} = \langle v_i^*, v_j^* \rangle$. در این صورت داریم $X = V^T V$، که در آن V ماتریسی است که ستون iام آن v_i^* است. در نتیجه X مثبت نیمه معین است. واضح است که X بقیه‌ی شرایط SDP را نیز برآورده می‌کند. در نتیجه X جوابی برای SDP است. برعکس، فرض کنید X^* جواب SDP باشد. چون $X^* \succcurlyeq 0$، طبق خاصیت ۲ در تعریف ۱ ماتریس $V \in \mathbb{R}^{m \times n}$ با $m \leq n$ وجود دارد که $X^* = V^T V$. با اضافه کردن صفر به انتهای V آن را به یک ماتریس مربعی تبدیل کنید و ستون‌های ماتریس حاصل را به عنوان v_i ها در نظر بگیرید. به راحتی می‌توان دید که v جوابی برای VP است.

تذکر ۱. تحت شرایطی، به ازای هر $\varepsilon > 0$ الگوریتمی وجود دارد که در زمان $O($چندجمله‌ای $+ \frac{1}{\varepsilon})$ جوابی با تقریب جمعی ε (یعنی جوابی در فاصله‌ی حداکثر ε از جواب بهینه) برای SDP بدست می‌آورد.

۱.۲ مساله‌ی برش بیشینه[۳]

گراف وزن‌دار $G = (V, E)$ را در نظر بگیرید که در آن یال e وزن w_e دارد. می‌خواهیم رئوس G را طوری به دو مجموعه افراز کنیم که مجموع وزن یال‌هایی که دو سرشان در دو مجموعه‌ی مختلف است بیشینه شود. فرض کنید $U \subset V$ و قرار دهید

$$y_i = \begin{cases} 1 & , i \in U \\ -1 & , i \in U^c \end{cases}$$

در این صورت $y_i y_j = -1$ اگر و تنها اگر i و j در مجموعه‌های مختلف باشند و $y_i y_j = 1$ اگر و تنها اگر i و j در یک مجموعه باشند. در نتیجه مجموع وزن یال‌هایی که دوسرشان در یک مجموعه نیست برابر است با $\frac{1}{2} \sum_{e=\{i,j\}} w_e (1 - y_i y_j)$. در نتیجه برنامه‌ریزی صحیح درجه دو زیر جواب دقیق این مساله را پیدا می‌کند.

$$\begin{aligned} \text{maximize} \quad & \frac{1}{2} \sum_{e=\{i,j\}} w_e (1 - y_i y_j) \\ \text{subject to} \quad & y_i \in \{-1, 1\} \quad , \forall i \in V. \end{aligned} \tag{1}$$

برنامه‌ریزی درجه دو فوق قابل حل در زمان چندجمله‌ای نیست.

[۳] MAX CUT problem

فرض کنید $n = |V|$. برنامه‌ریزی برداری زیر را در نظر بگیرید

$$\begin{aligned} \text{maximize} \quad & \frac{1}{2} \sum_{e=\{i,j\}} w_e(1 - \langle v_i, v_j \rangle) \\ \text{subject to} \quad & \langle v_i, v_i \rangle = 1 \quad , \forall i \in V \\ & v_i \in \mathbb{R}^n \quad , \forall i \in V. \end{aligned} \tag{2}$$

برنامه‌ریزی (۲) آرام شده‌ی برنامه‌ریزی (۲) زیرا فرض کنید y جوابی شدنی برای برنامه‌ریزی (۱) باشد. قرار دهید $v_i = (y_i, 0, \ldots, 0)$. در این صورت داریم $\langle v_i, v_i \rangle = y_i^2 = 1$. در نتیجه v جوابی شدنی برای برنامه‌ریزی (۲) است. در نتیجه (۲) آرام شده‌ی (۱) است. به ویژه داریم $Z_{VP} \geq \text{OPT}$، که در آن Z_{VP} جواب برنامه‌ریزی برداری است.

می‌توان در زمان چندجمله‌ای جوابی تقریبی برای برنامه‌ریزی برداری فوق بدست آورد. در جلسه‌ی بعد با گرد کردن تصادفی برنامه‌ریزی برداری فوق الگوریتمی تقریبی برای مساله‌ی برش بیشینه تولید می‌کنیم.