دانشکده علوم ریاضی
دانشگاه صنعتی شریف

کاربرد برنامه‌ریزی ریاضی در تولید الگوریتم‌های تقریبی

محمدهادی فروغمنداعرابی

پاییز ۱۳۹۶

گرد کردن تصادفی برنامه‌ریزی نیمه معین

جلسه‌ی دوازدهم

نگارنده: میلاد برزگر

۱ مروری بر مباحث گذشته

در جلسه‌ی پیش دیدیم برنامه‌ریزی نیمه معین به شکل

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

و برنامه‌ریزی برداری به شکل

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

است. همچنین دیدیم این دو برنامه‌ریزی معادل‌اند.

مساله‌ی برش بیشینه معرفی شد که در آن گراف وزن‌دار $G = (V, E)$ داده شده و هدف پیدا کردن بیشینه وزن برش‌ها است. دیدیم برنامه‌ریزی درجه دو صحیح

زیر این مساله را به طور دقیق مدل می‌کند

$$\begin{aligned} \text{maximize} \quad & \frac{۱}{۲} \sum_{e=\{i,j\}} w_e(۱ - y_i y_j) \\ \text{subject to} \quad & y_i \in \{-۱, ۱\} \quad , \forall i \in V. \end{aligned} \tag{۱}$$

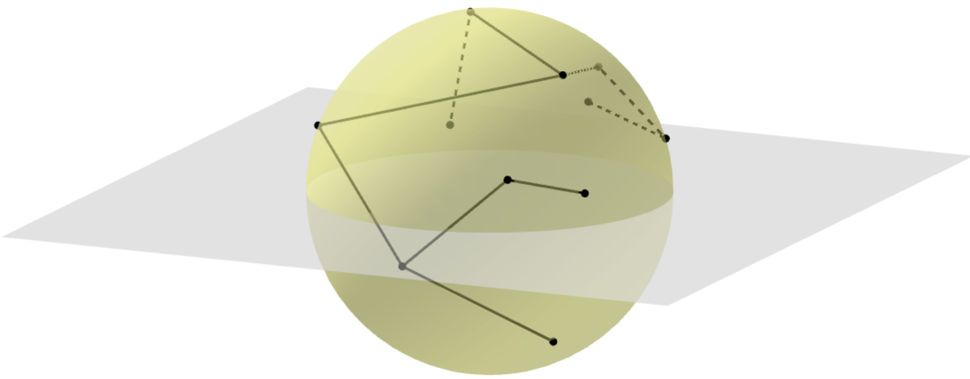
همچنین برنامه‌ریزی برداری آرام شده‌ی زیر برای مساله‌ی فوق ارائه شد

$$\begin{aligned} \text{maximize} \quad & \frac{۱}{۲} \sum_{e=\{i,j\}} w_e(۱ - \langle v_i, v_j \rangle) \\ \text{subject to} \quad & \langle v_i, v_i \rangle = ۱ \quad , \forall i \in V \\ & v_i \in \mathbb{R}^n \quad , \forall i \in V, \end{aligned} \tag{۲}$$

که در آن $n = |V|$.

۲ مساله‌ی برش بیشینه

برنامه‌ریزی (۱) در واقع راس‌های گراف G را طوری روی نقاط $(۱, ۰)$ و $(-۱, ۰)$ از $\mathbb{R}^۲$ قرار می‌دهد که مجموع وزن‌دار طول یال‌ها بیشینه شود. به طور مشابه برنامه‌ریزی (۲) راس‌های گراف را طوری روی سطح کره‌ی واحد n-بعدی پخش می‌کند (چون $\langle v_i, v_i \rangle = ۱$) که مجموع وزن‌دار طول یال‌ها بیشینه شود.



برنامه‌ریزی برداری قابل حل (تقریبی) در زمان چندجمله‌ای است. در ادامه الگوریتمی برای گرد کردن تصادفی جواب برنامه‌ریزی (۲) برای بدست آوردن جوابی تقریبی برای مساله‌ی برش بیشینه ارائه می‌کنیم.

الگوریتم ۱. الف) $v^* \leftarrow$ برنامه‌ریزی برداری را حل کن؛

ب) $r \leftarrow$ یک بردار تصادفی از کره‌ی واحد n-بعدی انتخاب کن؛

ج) $U \leftarrow \{i : \langle v_i, r \rangle > ۰\}$،

$U^c \leftarrow \{i : \langle v_i, r \rangle \leq ۰\}$.

ابرصفحه‌ی عمود بر r یک ابرصفحه‌ی تصادفی است. این الگوریتم در واقع کره را با یک ابرصفحه‌ی تصادفی می‌برد و راس‌های متناظر با بردارهای واقع در دو نیم‌کره‌ی حاصل را به عنوان برش مطلوب معرفی می‌کند.

تحلیل الگوریتم ۱. فرض کنید X_e نشان دهنده‌ی برشی بودن یا نبودن یال e باشد (در واقع $X_e = ۱$ اگر و تنها اگر دو سر e در دو طرف ابرصفحه‌ی برنده باشد).

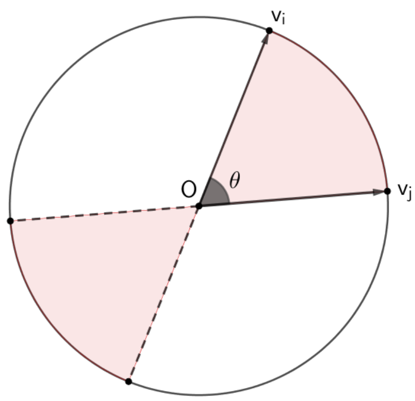
$$\mathbb{E}[\text{سود ما}] = \mathbb{E}\left[\sum_{e \in E} w_e X_e\right] = \sum_{e \in E} w_e \mathbb{E}[X_e] = \sum_{e \in E} w_e \mathbb{P}(X_e = ۱). \tag{۳}$$

در لم بعد $\mathbb{P}(X_e = ۱)$ را محاسبه می‌کنیم.

لم ۱. احتمال این که $e = \{i, j\}$ یال برشی باشد برابر است با $\frac{1}{\pi}\cos^{-1}\langle v_i, v_j\rangle$.

اثبات. ابرصفحه‌ی عمود بر r صفحه‌ی شامل v_i و v_j را به احتمال ۱ در یک خط قطع می‌کند (زیرا این اتفاق فقط در صورتی رخ نمی‌دهد که r بر صفحه‌ی شامل v_i و v_j عمود باشد و احتمال این پیشامد صفر است). آن را l می‌نامیم. این خط توزیع یکنواخت دارد. برشی بودن یال $e = \{i, j\}$ معادل است با این که v_i و v_j در دو طرف صفحه‌ی عمود بر r باشند و این پیشامد معادل است با این که l در ناحیه‌ی رنگی از شکل قرار بگیرد. با توجه به یکنواخت بودن توزیع l، احتمال این پیشامد برابر است با $\frac{2\theta}{2\pi}$. در نتیجه با توجه به این که طول v_i و v_j یک است، داریم

$$\mathbb{P}(X_e = 1) = \mathbb{P}(\text{یال } e \text{ برشی باشد}) = \frac{\theta}{\pi} = \frac{1}{\pi}\cos^{-1}\langle v_i, v_j\rangle.$$



□

حال به تحلیل الگوریتم ادامه می‌دهیم. با جایگذاری در رابطه‌ی (۳) بدست می‌آید

$$\mathbb{E}[\text{سود ما}] = \frac{1}{\pi}\sum_{e\in E} w_e \cos^{-1}\langle v_i, v_j\rangle \geq \left(\min_{x\in[-1,1]} \frac{\frac{1}{\pi}\cos^{-1}(x)}{\frac{1}{2}(1-x)}\right)\frac{1}{2}\sum_{e\in E} w_e\big(1-\langle v_i, v_j\rangle\big) \geq 0/878\, Z_{VP} \geq 0/878\,\mathrm{OPT}.$$

در بالا از این حقیقت استفاده شد که به ازای هر $x \in [-1, 1]$ داریم $\frac{1}{\pi}\cos^{-1}(x) \geq 0/878\,\frac{1}{2}(1-x)$. برای اثبات به کتاب مراجعه شود.

در نتیجه الگوریتم ۱ یک الگوریتم ۰/۸۷۸-تقریب برای مساله‌ی برش بیشینه است.

تذکر ۱. تصادفی انتخاب کردن y_i ها یک الگوریتم ۰/۵-تقریب است. زیرا داریم $\mathbb{E}[y_i y_j] = 0$ و درنتیجه

$$\mathbb{E}[\text{سود}] = \frac{1}{2}\sum_{e\in E} w_e\big(1-\mathbb{E}[y_i y_j]\big) = \frac{1}{2}\sum_{e\in E} w_e \geq \frac{1}{2}\mathrm{OPT}.$$

تذکر ۲. اگر $\mathrm{P} \neq \mathrm{NP}$، الگوریتم بهتر از ۰/۹۴۱۰۰-تقریب برای مساله‌ی برش بیشینه وجود ندارد. همچنین اگر UGC برقرار باشد الگوریتم بهتر از الگوریتم فوق (الگوریتمی که ضریب تقریب بهتری داشته باشد) برای مساله‌ی برش بیشینه وجود ندارد.