دانشکده علوم ریاضی
دانشگاه صنعتی شریف

تحقیق در عملیات ۱

محمدهادی فروغمنداعرابی
پاییز ۱۳۹۹

الگوریتم تقریبی برش بیشینه

جلسه ۱۲

نگارنده: نگارنده شایان طاهری جم

۱ مروری بر مباحث گذشته

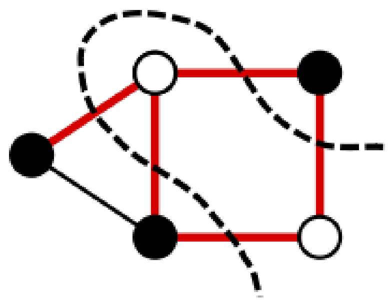
چند جلسه قبل با الگوریتم سیمپلکس آشنا شدیم که به کمک آن می‌توانستیم برنامه‌ریزی خطی را حل کنیم.
جلسه‌ی گذشته روش بیضی گون مطرح شد که با آن هم می‌توانیم برنامه‌ریزی خطی را حل کنیم (در واقع یک جواب شدنی به ما می‌دهد).

مرور روش بیضی‌گون:
در هر مرحله یک بیضی‌گون نگه می‌داشت که تمام جواب‌های شدنی برنامه‌ریزی خطی درون آن باشد. در هر مرحله چک می‌کرد که آیا مرکز بیضی‌گون شدنی است یا خیر؟ اگر شدنی بود که تمام و اگر نبود قیدی را می‌گرفت که مرکز در آن صدق نمی‌کرد و با کمک آن بیضی‌گون را برش می‌زد و بیضی‌گون کوچک‌تری ایجاد می‌کرد و همینطور بیضی‌گون را کوچک و کوچکتر می‌کرد.

یکی از مزیت‌های روش بیضی‌گون نسبت به سیمپلکس این است که بیضی‌گون واقعاً در زمان چندجمله‌ای برنامه‌ریزی خطی را حل می‌کند ولی سیمپلکس در عمل بهتر کار می‌کند.
روش بیضی‌گون می‌توانست کاری به برنامه‌ریزی خطی نداشته باشد و تابعی داشته باشد به طوری که در هر مرحله چک کند آیا مرکز شدنی است یا خیر و اگر شدنی نبود یک قید نقض شده را خروجی دهد.
این قابلیت باعث می‌شود که با روش بیضی‌گون بتوانیم برنامه‌ریزی‌های خطی با نامتناهی قید را هم حل کنیم به شرطی که آن تابع مورد نظر وجود داشته باشد.

۲ برش بیشینه

در این مسئله ورودی یک گراف است و هدف پیدا کردن مجموعه‌ای از رئوس مانند S است که $|E[S,\bar{S}]|$ بیشینه شود. این مسئله می‌تواند وزن دار هم باشد به طوری که هر یال وزنی داشته باشد و هدف پیدا کردن مجموعه‌ی S است به طوری که مجموع وزن یال‌های $E[S,\bar{S}]$ بیشینه شود.
مشابه این مسئله مسئله‌ی برش کمینه را نیز داریم که به همین صورت است با این تفاوت که هدف پیدا کردن مجموعه‌ی S است به طوری که مجموع وزن یال‌های $E[S,\bar{S}]$ کمینه باشد.
مسئله‌ی برش کمینه الگوریتم چندجمله‌ای دارد ولی مسئله‌ی برش بیشینه NP-Hard است. پس سعی می‌کنیم برای این مسئله الگوریتم تقریبی ارائه دهیم.



هدف الگوریتمی است که امید ریاضی خوبی داشته باشد.

۱.۲ الگوریتم تقریبی ۱ (چیدمان تصادفی)

در این الگوریتم هر رأس را به احتمال $\frac{۱}{۲}$ در مجموعه‌ی S قرار می‌دهیم.
ادعا می‌کنیم که امید ریاضی جوابی که با این الگوریتم حساب می‌کنیم حداقل $\frac{۱}{۲}$ جواب بهینه است.
در واقع ادعای بزرگتری می‌کنیم و می‌گوییم امید ریاضی جواب این الگوریتم حداقل $\frac{۱}{۲}$ مجموع وزن تمام یال‌های گراف است.
امید ریاضی جواب الگوریتم را حساب می‌کنیم:

$$E[w(S,\bar{S})] = E[\sum_{\{u,v\}\in E(S,\bar{S})} w(\{u,v\})]$$

در اینجا نمی‌توانیم امید ریاضی را داخل سیگما ببریم پس یک متغیر تصادفی تعریف می‌کنیم به نام X_{uv} که نشان می‌دهد یال uv در برش ما حضور دارد یا خیر. در واقع اگر یال uv در برش باشد مقدار آن ۱ و در غیر اینصورت مقدار آن ۰ خواهد بود.
حال داریم:

$$E[w(S,\bar{S})] = E[\sum_{\{u,v\}\in E(S,\bar{S})} w(\{u,v\})] = E[\sum_{uv\in E} X_{uv} w(\{u,v\})] = \sum_{uv\in E} E[X_{uv}] w(\{u,v\}) = \frac{۱}{۲}\sum_{uv\in E} w(\{u,v\})$$

که $\sum_{uv\in E} w(\{u,v\})$ برابر مجموع وزن یال‌ها است.

۲.۲ الگوریتم تقریبی ۲

می‌خواهیم به کمک برنامه ریزی خطی این مسئله را حل کنیم.
به ازای هر رأس متغیر x_i را در نظر می‌گیریم به طوری که اگر $x_i \in S$ مقدار آن برابر ۱ و در غیر اینصورت مقدار آن برابر ۱− باشد.
حال مقدار $\frac{۱}{۲}|x_i - x_j|$ به ازای هر یال اگر آن یال در برش باشد برابر ۱ می‌شود و در غیر اینصورت برابر ۰ خواهد شد.
از طرفی می‌دانیم:

$$|x_i - x_j| = (x_i - x_j)^۲ = \frac{۱}{۲}x_i^۲ + \frac{۱}{۲}x_j^۲ - x_i x_j = \frac{۱}{۲}(۱ - x_i x_j)$$

پس تابع هدف ما برابر می‌شود با:

$$\sum_{ij\in E}\frac{1}{2}(1-x_ix_j)$$

برنامه‌ریزی ما به این صورت خواهد بود:

$$\begin{aligned} \text{بیشینه کن} \quad & \sum_{ij\in E}\frac{1}{2}(1-x_ix_j) \\ \text{که} \quad & x_i\in\{1,-1\} \end{aligned}$$

که این برنامه ریزی نه خطی است و نه متغیرهای آن صحیح هستند (چون مقدار ۰ را نمی‌توانند بگیرند). حال متغیرهای جدید a_{ij} را تعریف می‌کنیم به طوری که $a_{ij}=x_i.x_j$
حال برنامه ریزی ما به این شکل در می‌آید:

$$\begin{aligned} \text{بیشینه کن} \quad & \sum_{ij\in E}\frac{1}{2}(1-a_{ij}) \\ \text{که} \quad & a_{ij}=x_i.x_j \\ & x_ix_i=1 \end{aligned}$$

یعنی ماتریسی مانند A وجود داشته باشد به طوری که درایه‌ی (i,j) آن برابر a_{ij} باشد و همینطور این درایه برابر $x_i.x_j$ باشد.
در واقع باید بردار x وجود داشته باشد به طوری که $A=x.x^T$
در واقع بردار x به صورت $(x_1,x_2,\dots,x_n)$ است.

$$x.x^T=\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}\times\begin{bmatrix}x_1 & x_2 & \dots & x_n\end{bmatrix}=\begin{bmatrix}x_1.x_1 & \dots & x_1.x_n\\ \vdots & \ddots & \\ x_n.x_1 & & x_n.x_n\end{bmatrix}=A \tag{1}$$

حال به جای بردار x از ماتریس z استفاده می‌کنیم. یعنی به جای این که بخواهیم ماتریس A را به صورت $x.x^T$ بنویسیم که x بردار است آن را به صورت $z.z^T$ می‌نویسیم که $z\in R^{n\times n}$.

در واقع این کار نوعی آرام‌سازی است چرا که اگر بتوان A را به صورت $x.x^T$ نوشت آنگاه قرار می‌دهیم $z=\begin{bmatrix}x_1 & 0 & \dots & 0\\ x_2 & 0 & \dots & 0\\ \vdots & \vdots & \ddots & \\ x_n & 0 & \dots & 0\end{bmatrix}$ و به راحتی می‌توان نشان داد $z.z^T=A$

حال می‌خواهیم کمی وارد جبرخطی بشویم.

ماتریس A را می‌توان به صورت $z.z^T$ نوشت اگر و تنها اگر A مثبت نیمه معین باشد.

تعریف: به ماتریس متقارن $A\in R^{n\times n}$ مثبت نیمه معین می‌گوییم اگر به ازای هر $y\in R^n$ داشته باشیم $y^TAy\geq 0$.

قضیه: اگر بتوان ماتریس A را به صورت B^TB نوشت آنگاه A مثبت نیمه معین است.
اثبات: فرض کنید $y\in R^n$ می‌خواهیم نشان دهیم $y^TAy\geq 0$

$$y^TAy=y^TB^TBy=(||By||_2)^2\geq 0$$

قضیه: اگر A ماتریسی مثبت نیمه معین باشد آنگاه $B\in R^{n\times n}$ وجود دارد به طوری که $A=B^T.B$
قصد نداریم این قضیه را کامل اثبات کنیم ولی اشاره‌ای به اثبات آن می‌کنیم:

هر ماتریس مثبت نیمه معین را می‌توان به صورت $Q^{-1}DQ$ نوشت به طوری که Q ماتریسی متعامد باشد($Q^{-1} = Q^T$) و ماتریس D قطری باشد به طوری که تمام درایه‌های آن مثبت باشد و در نتیجه می‌توان D را به صورت $(D^{\frac{1}{2}})^2$ نوشت به طوری که درایه‌های $D^{\frac{1}{2}}$ برابر جذر درایه‌ی متناظر آن در D است.

در نتیجه می‌توان نوشت:

$$A = Q^{-1}DQ = Q^{-1}D^{\frac{1}{2}}D^{\frac{1}{2}}Q = Q^TD^{\frac{1}{2}}D^{\frac{1}{2}}Q = (D^{\frac{1}{2}}Q)^TD^{\frac{1}{2}}Q$$

که اگر قرار دهیم $B = D^{\frac{1}{2}}Q$ نتیجه می‌شود که $A = B^TB$ و قضیه اثبات می‌شود.

نتیجه :

$$A \text{ مثبت نیمه معین باشد } \iff A = B^TB$$

الگوریتمی وجود دارد که اگر A مثبت نیمه معین باشد تجزیه‌ی A را به صورت B^TB به ما بدهد و اگر مثبت نیمه معین نباشد $y \in R^n$ را خروجی دهد به طوری که $y^TAy < 0$

اشاره‌ای به این الگوریتم می‌کنیم:

۳ الگوریتم چولسکی

این الگوریتم ماتریس A را به عنوان ورودی می‌گیرد و آن را مرحله به مرحله به ماتریس قطری نزدیک می‌کند.

در مرحله‌ی i ام ماتریس $A^{(i)}$ را به ماتریس $A^{(i+1)}$ تبدیل می‌کند که این ماتریس‌ها به این صورت هستند:

$$A^{(i)} = \begin{pmatrix} I_{i-1} & 0 & 0 \\ 0 & a_{i,i} & b_i^* \\ 0 & b_i & B^i \end{pmatrix}$$

$$L_i := \begin{pmatrix} I_{i-1} & 0 & 0 \\ 0 & \sqrt{a_{i,i}} & 0 \\ 0 & \frac{1}{\sqrt{a_{i,i}}}b_i & I_{n-i} \end{pmatrix}$$

حال ماتریس $A^{(i+1)}$ ماتریسی است که در تساوی $A^{(i)} = L_iA^{(i+1)}L_i^*$ صدق می‌کند و به صورت زیر است.

$$A^{(i+1)} = \begin{pmatrix} I_{i-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & B^i - \frac{1}{a_{i,i}}b_ib_i^* \end{pmatrix}$$

و در نهایت قرار می‌دهیم $B^T := L_1L_2 \dots L_n$ و درنتیجه $A = B^TB$

طرف دیگر این الگوریتم (اگر A مثبت نیمه معین نباشد $y \in R^n$ را خروجی دهد به طوری که $y^TAy < 0$) و اثبات درستی آن خیلی مربوط به درس ما نمی‌شود پس آن‌ها را بدون اثبات قبول می‌کنیم و الگوریتم تقریبی دوم را ادامه می‌دهیم.

۴ ادامه‌ی الگوریتم تقریبی ۲

برنامه‌ریزی ما به این اینصورت شده بود:

$$\begin{aligned} \text{بیشینه کن} \quad & \sum_{ij \in E} \frac{1}{2}(1 - a_{ij}) \\ \text{که} \quad & a_{ij} = x_i.x_j \\ & x_ix_i = 1 \end{aligned}$$

که اکنون پس از ریلکس کردن آن به این برنامه‌ریزی می‌رسیم:

$$\begin{aligned} \text{بیشینه کن} \quad & \sum_{ij\in E} \frac{1}{2}(1-a_{ij}) \\ \text{که} \quad & A \text{ مثبت نیمه معین باشد} \\ & a_{i,i} = 1 \end{aligned}$$

که این برنامه‌ریزی به این صورت است:

$$\begin{aligned} \text{بیشینه کن} \quad & \sum_{ij\in E} \frac{1}{2}(1-a_{ij}) \\ \text{که} \quad \forall y \in R^n: \quad & y^T A y \geq 0 \\ & a_{i,i} = 1 \end{aligned}$$

که این برنامه‌ریزی خطی‌ای با نامتناهی قید است.
حال می‌دانیم اگر الگوریتمی داشته باشیم که چک کند آیا نقطه‌ای شدنی است یا خیر و اگر خیر یک قید که نقطه در آن صدق نمی‌کند را خروجی دهد می‌توانیم برنامه‌ریزی خطی با نامتناهی قید را نیز به کمک الگوریتم بیضی‌گون حل کنیم و اکنون برای این برنامه‌ریزی خطی به کمک الگوریتم چولسکی توانسنیم آن الگوریتمی که لازم بود را تولید کنیم پس می‌توانیم این برنامه‌ریزی را حل کنیم.

حال که برنامه‌ریزی را حل کردیم به ازای هر رأس یک بردار z_i داریم ولی در برنامه ریزی اصلی به ازای هر رأس یک x_i داشتیم که مقدار آن ۱ یا ۱− بود. حال باید به طریق خوبی x_i ها را مشخص کنیم.
ابتدا مقداری به z_i ها دقت می‌کنیم. شرط $z_i.z_i = 1$ نشان می‌دهد که $|z_i| = 1$ پس تمامی بردارها روی کره واحد قرار دارند به طوری که مقدار $\sum_{ij\in E} z_i z_j w(i,j)$ کمینه است یعنی بردارهای متناظر با رئوسی که وزن یال‌های بینشان زیاد است از هم دور شده اند. حال باید یک صفحه از کره رد کنیم به طوری که بردارها را به دو قسمت تقسیم کند و رئوس متناظر با بردارهای یک سمت را در S و بقیه را در $\bar{S}$ قرار دهیم.
چگونه این صفحه را انتخاب کنیم؟
جواب: به صورت تصادفی صفحه‌ای دلخواه انتخاب می‌کنیم به طور شهودی چون z_i ها به طور خوبی انتخاب شده‌اند و هر رأس از رئوسی که به آن‌ها یال دارد دور است پس این صفحه‌ی دلخواه احتمالاً این رئوس را از هم جدا کند.

۵ تحلیل الگوریتم

متغیر تصادفی Y_{ij} را تعریف می‌کنیم به طوری که مقدار آن ۱ است اگر یال ij در برش ما باشد و در غیر این صورت مقدار آن ۰ است.
امتیاز ما برابر می‌شود با $\sum_{ij\in E} Y_{ij}$ و امید ریاضی امتیاز ما برابر می‌شود با $E[\sum_{ij\in E} Y_{ij}]$ از خطی بودن امید ریاضی استفاده می‌کنیم:

$$E[\sum_{ij\in E} Y_{ij}] = \sum_{ij\in E} E[Y_{ij}]$$

و چون مقدار Y_{ij} صفر و یکی است امید ریاضی آن با احتمال آن برابر است پس:

$$\sum_{ij\in E} E[Y_{ij}] = \sum_{ij\in E} \mathbb{P}r(Y_{ij} = 1)$$

چه موقع Y_{ij} برابر با ۱ می‌شود؟ اگر صفحه‌ی ما از بین z_i و z_j بگذرد و باید احتمال آن را حساب کنیم.
برای این کار شکل را رو صفحه‌ی گذرنده از z_i و z_j تصویر می‌کنیم و شکل به صورت یک دایره با دو بردار z_i و z_j می‌شود که خطی از آن گذشته و باید احتمال اینکه این خط دو بردار z_i و z_j را جدا کند را حساب کنیم که اگر زاویه بین دو بردار α باشد این احتمال برابر با $\frac{\alpha}{\pi}$.

$$\mathbb{P}r[Y_{ij} = 1] = \frac{\alpha_{ij}}{\pi} = \frac{\arccos(z_i^T z_j)}{\pi}$$

حال اگر نمودار تابع $\frac{\frac{\arccos(x)}{\pi}}{\frac{1-x}{2}}$ را رسم کنیم مشاهده می‌کنیم که مقدار آن در بازه‌ی $[-1, 1]$ همیشه بیشتر از ۰٫۸۷۸ است پس می‌توان گفت:

$$\mathbb{P}r[Y_{ij} = 1] = \frac{\alpha_{ij}}{\pi} = \frac{\arccos(z_i^T z_j)}{\pi} \geq 0.878 \times \frac{1 - z_i^T z_j}{2}$$

پس

$$E[\sum_{ij\in E} Y_{ij}] \geq ۰/۸۷۸ \times \sum_{ij\in E} \frac{۱ - z_i^T z_j}{۲} \geq ۰/۸۷۸ z^*$$

که در اینجا z^* جواب بهینه‌ی برنامه‌ریزی آرام‌شده بود.

مراجع

[JB07] Matoušek Jiri and Gärtner Bernd. *Understanding and using linear programming.* Springer, 2007.