دانشکده علوم ریاضی
دانشگاه صنعتی شریف

تحقیق در عملیات ۱

محمدهادی فروغمنداعرابی
پاییز ۱۳۹۹

الگوریتم تقریبی برش بیشینه

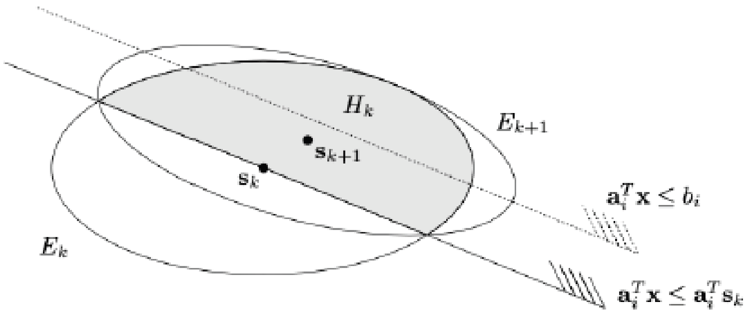
جلسه دوازدهم

نگارنده: جواد فرخ‌نژاد

۱ مروری بر مباحث گذشته

در جلسات گذشته الگوریتم سیمپلکس[۱] را برای حل مسائل برنامه‌ریزی خطی مطرح کردیم و در جلسه‌ی گذشته نیز به معرفی الگوریتم بیضی‌گون[۲] پرداختیم. در این روش ابتدا یک بیضی‌گون (به طور خاص در مرحله‌ی اول، یک کره) به مرکز مبدأ و شعاع R (که جزو ورودی‌های الگوریتم است) انتخاب می‌کنیم که فضای شدنی مسئله به تمامی در آن قرار داشته باشد. سپس در هر مرحله بررسی می‌کنیم که آیا مرکز بیضی‌گون در فضای شدنی مسئله قرار دارد یا خیر.
اگر قرار داشته باشد، توانسته‌ایم یک نقطه‌ی شدنی بیابیم و الگوریتم پایان می‌یابد.
اگر قرار نداشته باشد، آنگاه حداقل یکی از قیدها را مثل $a_i^\intercal x \leq b_i$ نقض می‌کند. پس اگر مرکز بیضی‌گون را در مرحله‌ی kام s_k بنامیم خواهیم داشت $a_i^\intercal s_k > b_i$. حال ابرصفحه‌ی $a_i^\intercal x = a_i^\intercal s_k$ را درنظر می‌گیریم که بیضی‌گون را نصف می‌کند. فضای شدنی مسئله به تمامی در یکی از دو نصفه‌ی بیضی‌گون قرار خواهد گرفت. در این صورت یک بیضی‌گون جدید و با حجم کوچکتر از قبلی می‌سازیم که همچنان فضای شدنی مسئله به تمامی در آن قرار داشته باشد و الگوریتم بالا را تکرار می‌کنیم.
با ادامه‌ی این روند در هر مرحله حجم بیضی‌گون حداقل با یک ضریب مثبت مثل δ کوچک می‌شود و نهایتاً که حجم آن از ϵ (که جزو ورودی‌های الگوریتم است) کمتر شد، مرکز بیضی‌گون را به عنوان یک نقطه‌ی شدنی معرفی می‌کنیم.

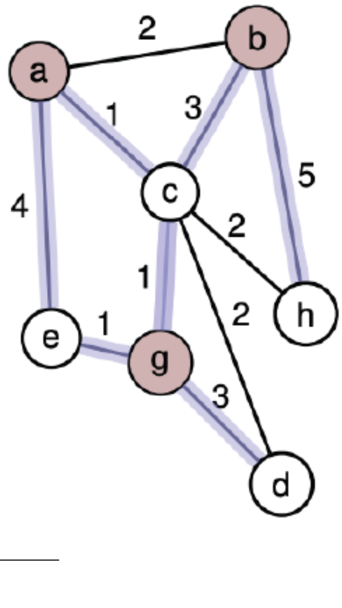
---
[1] Simplex
[2] Ellipsoid



فایده‌ای که الگوریتم بیضی‌گون نسبت به سیمپلکس دارد این است که زمان اجرای آن چندجمله‌ای است که در ادامه از این خاصیت استفاده می‌کنیم.
نکته‌ی دیگری که در الگوریتم بیضی‌گون وجود دارد این است که این الگوریتم نیازی ندارد به مسئله‌ی برنامه‌ریزی خطی دسترسی داشته باشد. درواقع می‌توانیم یک تابع داشته باشیم که به نوعی آن را مسئول برنامه‌ریزی خطی بنامیم و این تابع یک نقطه از فضا می‌گیرد و تعیین می‌کند که این نقطه در فضای شدنی مسئله قرار دارد یا خیر و در صورت قرار نداشتن، یک ابرصفحه که متناظر با یکی از قیدهای مسئله است را برمی‌گرداند که نقطه‌ی داده شده، آن قید را نقض می‌کند.
یعنی الگوریتم بیضی‌گون فقط نیاز به چنین تابعی دارد و لزومی ندارد خودش به قیدهای مسئله دسترسی داشته باشد. مثلاً ممکن است تعداد قیدهای مسئله نامتناهی باشد اما بتوانیم به گونه‌ای تشخیص دهیم که نقطه‌ی داده شده در فضای شدنی قرار دارد یا اینکه کدام قید را نقض می‌کند. در حالتی که تعداد قیدهای مسئله نامتناهی است ممکن است فضای شدنی مسئله دیگر یک چندوجهی محدب نشود.
در ادامه مثالی از این تکنیک خواهیم دید.

۲ برش بیشینه

فرض کنید گراف $G = (V, E)$ داده شده است. هدف یافتن یک برش[۳] برای این گراف است که بیشترین تعداد یال را داشته باشد. یعنی هدف یافتن زیر مجموعه‌ای از رأس‌های گراف مثل S^* است که برای هر $S \subseteq V$ داشته باشم $E[S, V \setminus S] \le E[S^*, V \setminus S^*]$.
می‌توان یال‌ها را وزن‌دار نیز فرض کرد و هدف یافتن برش با وزن بیشینه باشد. یعنی $S^* \subseteq V$ بیابیم که برای هر $S \subseteq V$ داشته باشیم $w(S) = w[S, V \setminus S] \le w[S^*, V \setminus S^*] = w(S^*)$.
مثلاً در گراف زیر داریم $S^* = \{a, b, g\}$ که یال‌های برش نیز رنگ شده‌اند.



[3] cut

به طور مشابه می‌توان مسئله‌ی برش کمینه را نیز مطرح کرد. برای مسئله‌ی برش کمینه الگوریتم چندجمله‌ای وجود دارد اما مسئله‌ی برش بیشینه یک مسئله‌ی ان‌پی‌ـ سخت[4] است. قصد داریم دو الگوریتم تقریبی برای این مسئله ارائه کنیم.

۱.۲ الگوریتم تقریبی اول

الگوریتم اول به این صورت است که هر رأس از گراف را مستقل از بقیه، با احتمال $\frac{1}{2}$ در مجموعه‌ی S قرار می‌دهیم. در نهایت، $(S, V\setminus S)$ را به عنوان برش معرفی می‌کنیم. اثبات می‌کنیم امید ریاضی وزن برشِ پیدا شده از نصف وزن برش بیشینه کمتر نیست.
برای هر یال $uv \in E$، یک متغیر تصادفی شاخص χ_{uv} تعریف کنید که برابر با ۱ است اگر و تنها اگر یال uv در برش تصادفی انتخاب شده وجود داشته باشد. یعنی از رأس‌های u و v دقیقاً یکی در S آمده باشد. امید ریاضی وزن برش انتخابی برابر است با:

$$\begin{aligned}
\mathrm{E}\left[w\left(S, V\setminus S\right)\right] &= \mathrm{E}\left[\sum_{uv\in E[S,V\setminus S]} w\left(uv\right)\right] = \mathrm{E}\left[\sum_{uv\in E}\chi_{uv} w\left(uv\right)\right] = \sum_{uv\in E}\mathrm{E}\left[\chi_{uv}\right] w\left(uv\right)\\
&= \sum_{uv\in E}\Pr\left[\chi_{uv}=1\right]\times w\left(uv\right) = \sum_{uv\in E}\Pr\left[\{u\in S, v\in V\setminus S\}\cup\{u\in V\setminus S, v\in S\}\right]\times w\left(uv\right)\\
&= \sum_{uv\in E}\left(\frac{1}{4}+\frac{1}{4}\right)\times w\left(uv\right) = \sum_{uv\in E}\frac{1}{2}w\left(uv\right) \geq \frac{1}{2}w(S^*)\\
\Rightarrow \mathrm{E}\left[w\left(S, V\setminus S\right)\right] &\geq \frac{1}{2}w(S^*)
\end{aligned}$$

دقت کنید که نامساوی آخر از اینجا نتیجه شد که برش بیشینه، حداکثر تمام یال‌ها را می‌تواند داشته باشد. پس وزن برش بیشینه از جمع وزن کل یال‌ها بیشتر نیست.
پس الگوریتمی با ضریب تقریب $\frac{1}{2}$ یافتیم. البته تعریف ضریب تقریب در منابع مختلف متفاوت است و اینجا منظور از ضریب تقریب $\frac{1}{2}$، این است که امید ریاضی وزن برشی که به طور تصادفی انتخاب می‌شود از نصف وزن برش بیشینه کمتر نیست. تعریف دیگری که برای الگوریتم با ضریب تقریب α وجود دارد این است که جوابی که پیدا می‌شود حتماً باید از α برابر جواب بهینه بدتر نباشد (نه اینکه امید ریاضی آن از α برابر جواب بهینه بدتر نباشد).
همچنین استدلال بالا نشان می‌دهد که وزن برش بیشینه، حتماً از نصف جمع وزن کل یال‌ها بیشتر است.

۲.۲ پیشنیازها برای الگوریتم دوم

برای معرفی الگوریتم تقریبی دوم برای مسئله‌ی برش بیشینه ابتدا مطالبی را از جبرخطی بیان می‌کنیم که در الگوریتم از آن‌ها استفاده می‌شود.

تعریف ۱ [*ماتریس مثبتِ نیمه‌معین*[5]] *به ماتریس متقارن* $A \in \mathbb{R}^{n\times n}$ *مثبت نیمه‌معین می‌گویند اگر و تنها اگر برای هر بردار* $y \in \mathbb{R}^n$ *داشته باشیم* $y^\intercal A y \geq 0$.

می‌توانیم تعریف قبل را تعمیم دهیم و شرط متقارن بودن A را حذف کنیم، اما در بیشتر کاربردها، مثبت نیمه‌معین بودن برای ماتریس‌های نامتقارن مطرح نمی‌شود. درواقع برای هر ماتریس A که حتی متقارن هم نباشد می‌توان ماتریس A' را این طور تعریف کرد که برای هر $1 \leq i, j \leq n$ داشته باشیم $A'_{ij} = A'_{ji} = \frac{A_{ij}+A_{ji}}{2}$. در اینصورت به سادگی می‌توان بررسی کرد که برای هر بردار $x \in \mathbb{R}^n$ داریم $x^\intercal A' x = x^\intercal A x$ و همچنین A' متقارن است. یعنی در اکثر موارد از لحاظ کاربردی کافی است مثبت نیمه‌معین را، برای ماتریس‌های متقارن مورد بحث قرار دهیم.
ماتریس‌های مثبت نیمه‌معین خواص جالبی دارند. از لحاظ هندسی این ماتریس‌ها بردارهای فضا را با زاویه‌ای کمترمساویِ $\frac{\pi}{2}$ تغییر جهت می‌دهند. یعنی اگر A مثبت نیمه‌معین باشد آنگاه برای هر بردار ناصفر $x \in \mathbb{R}^n$ که $Ax \neq 0$ باشد، حتماً زاویه‌ی بین بردارهای x و Ax کمترمساویِ $\frac{\pi}{2}$ خواهد بود. درواقع اگر زاویه‌ی بین x و Ax برابر با α باشد، این حکم به سادگی از تعریف نتیجه می‌شود زیرا

$$\langle x, Ax\rangle = x^\intercal A x \geq 0 \Rightarrow \|x\|\|Ax\|\cos(\alpha) \geq 0 \Rightarrow \cos(\alpha) \geq 0 \Rightarrow 0 \leq \alpha \leq \frac{\pi}{2}$$

قضیه ۱. *ماتریس* $A \in \mathbb{R}^{n\times n}$ *مثبت نیمه‌معین است اگر و تنها اگر ماتریس* $B \in \mathbb{R}^{n\times n}$ *موجود باشد به طوری که* $A = B^\intercal B$.

اثبات. اثبات یک طرف قضیه ساده است. درواقع با فرض اینکه $A = B^\intercal B$، برای هر بردار $y \in \mathbb{R}^n$ خواهیم داشت:

$$y^\intercal A y = y^\intercal B^\intercal B y = (By)^\intercal (By) = \|By\|^2 \geq 0 \Rightarrow y^\intercal A y \geq 0$$

[4] NP-Hard
[5] positive semi-definite

پس A در تعریف صدق می‌کند پس مثبت نیمه‌معین است. برای اثبات طرف دیگر قضیه فرض کنید A مثبت نیمه‌معین باشد، پس A متقارن است. طبق قضیه‌ای از جبرخطی می‌دانیم ماتریس‌های متقارن روی میدان اعداد حقیقی، قطری‌شدنی[6] هستند. درواقع به طور متعامد نیز قطری می‌شوند. یعنی ماتریس متعامد Q ($Q^{-1} = Q^{\intercal}$) و ماتریس قطری D وجود دارند که $A = Q^{-1}DQ$. همچنین اعداد روی قطر D نامنفی‌اند که این خاصیت از مثبت نیمه‌معین بودن A نتیجه می‌شود. زیرا اگر λ یک مقدار ویژه‌ی A متناظر با بردار ویژه‌ی v باشد، خواهیم داشت:

$$\left.\begin{array}{r} v^{\intercal}Av = v^{\intercal}\lambda v = \lambda v^{\intercal}v = \lambda\|v\|^{2} \\ v \in \mathbb{R}^{n} \Rightarrow v^{\intercal}Av \geq 0 \end{array}\right\} \Rightarrow \lambda\|v\|^{2} \geq 0 \Rightarrow \lambda \geq 0$$

دقت کنید که چون v بردار ویژه است پس ناصفر است پس $\|v\|^{2}$ نیز ناصفر است و نتیجه‌گیری آخر معتبر است.
پس اعداد روی قطر D نامنفی‌اند. ماتریس $D^{\frac{1}{2}}$ را ماتریس قطری درنظر بگیرید که درایه‌ی iiامِ آن برابر است با جذر درایه‌ی iiامِ D. حال تعریف کنید $B = D^{\frac{1}{2}}Q$. در این صورت به سادگی خواهیم داشت:

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

پس قضیه به طور کلی اثبات می‌شود. □

در کنار مثبت نیمه‌معین بودن، مثبت معین بودن نیز مطرح می‌شود که همان مثبت نیمه‌معین بودن است با این شرط اضافه که ماتریس A وارون‌پذیر نیز باشد.

تعریف ۲ [*ماتریس مثبت معین*[7]] *به ماتریس متقارن* $A \in \mathbb{R}^{n\times n}$ *مثبت معین می‌گویند اگر و تنها اگر برای هر بردار ناصفر* $y \in \mathbb{R}^{n}$ *داشته باشیم* $y^{\intercal}Ay > 0$.

می‌توان به سادگی بررسی کرد قضیه‌ی ۱ برای ماتریس‌های مثبت معین هم برقرار است فقط با این تفاوت که ماتریس B الزاماً وارون‌پذیر نیز باید باشد. درواقع مقادیر ویژه‌ی یک ماتریس مثبت نیمه‌معین ممکن است صفر باشند اما مقادیر ویژه‌ی ماتریس‌های مثبت معین الزاماً مثبت‌اند و نمی‌توانند صفر باشند.

الگوریتمی وجود دارد که برای یک ماتریس دلخواه مثل A، اگر مثبت معین باشد، ماتریس B را طوری می‌یابد که $A = B^{\intercal}B$ و اگر مثبت معین نباشد، بردار y را طوری می‌یابد که $y^{\intercal}Ay < 0$. این الگوریتم، تجزیه‌ی چولسکی[8] نام دارد که در اینجا فقط یک سمت آن را بیان می‌کنیم.
فرض کنید A مثبت معین باشد، می‌خواهیم ماتریس B را طوری بیابیم که $A = B^{\intercal}B$.
قرار دهید $A^{(1)} = A$، هدف این است که ماتریس A را مرحله به مرحله تغییر دهیم و در نهایت به ماتریس همانی برسیم. در مرحله‌ی iام می‌خواهیم یک ماتریس $A^{(i+1)}$ بیابیم که به فرم $\begin{bmatrix} I_i & 0 \\ 0 & * \end{bmatrix}$ باشد. یعنی سطرها و ستون‌های اول تا iامِ آن دقیقاً برابر با سطرها و ستون‌های اول تا iامِ ماتریس همانی باشد.
فرض کنید در مرحله‌ی iام ماتریس $A^{(i)} = \begin{bmatrix} I_{i-1} & 0 & 0 \\ 0 & a_{ii} & b_i^{\intercal} \\ 0 & b_i & B^{(i)} \end{bmatrix}$ را داریم.
تعریف کنید:

$$L_i = \begin{bmatrix} I_{i-1} & 0 & 0 \\ 0 & \sqrt{a_{ii}} & 0 \\ 0 & \frac{1}{\sqrt{a_{ii}}}b_i & I_{n-i} \end{bmatrix}$$

و قرار دهید $A^{(i+1)} = L_i^{-1}A^{(i)}\left(L_i^{\intercal}\right)^{-1}$.
می‌توان به سادگی بررسی کرد که ماتریس $A^{(i+1)}$ به صورت زیر خواهد بود:

$$A^{(i+1)} = \begin{bmatrix} I_{i-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & B^{(i)} - \frac{1}{a_{ii}}b_ib_i^{\intercal} \end{bmatrix}$$

یعنی ماتریس جدید تا سطر و ستون $(i+1)$ام مشابه با همانی است. اگر همین روند را ادامه دهیم با توجه به $A^{(i)} = L_iA^{(i+1)}L_i^{\intercal}$ در نهایت خواهیم داشت $A = L_1L_2\dots L_nL_n^{\intercal}L_{n-1}^{\intercal}\dots L_1^{\intercal}$. پس کافی است قرار دهیم $B = L_n^{\intercal}L_{n-1}^{\intercal}\dots L_1^{\intercal}$ و به این صورت تجزیه‌ی موردنظر بدست می‌آید. تنها مشکلی که ممکن است وجود داشته باشد این است که مقدارهای a_{ii} صفر یا منفی شوند و در این صورت ماتریس L_i روی

[6] diagonalizable
[7] positive definite
[8] Cholesky decomposition

میدان اعداد حقیقی تعریف نشده خواهد بود. اما چون A را مثبت معین فرض کردیم، می‌توانید به عنوان تمرین این را اثبات کنید که مقدارهای a_{ii} در هیچ مرحله‌ای صفر یا منفی نخواهند بود و همواره مثبت‌اند (راهنمایی: ابتدا اثبات کنید که ماتریس A مثبت معین است اگر و تنها اگر برای هر $1 \leq i \leq n$، دترمینان زیرماتریس مربعی $i \times i$ که از برخورد سطرها و ستون‌های اول تا iاِم A حاصل می‌شود، اکیداً مثبت باشد).
همچنین به سادگی می‌توان دید که زمان اجرای این الگوریتم چندجمله‌ای است که بعداً از آن استفاده می‌کنیم.

الگوریتم تجزیه‌ی چولسکی برای ماتریس‌های مثبت نیمه‌معین نیز وجود دارد (ممکن است a_{ii} در یکی از مراحل صفر شود و الگوریتم قبلی جواب نمی‌دهد). اما آن را بیان نمی‌کنیم و فقط فرض می‌کنیم که یک الگوریتم با زمان اجرای چندجمله‌ای وجود دارد که با گرفتن یک ماتریس A، اگر A مثبت نیمه‌معین باشد یک ماتریس $B \in \mathbb{R}^{n\times n}$ برمی‌گرداند که $A = B^{\intercal}B$ و اگر A مثبت نیمه‌معین نباشد یک بردار $y \in \mathbb{R}^n$ برمی‌گرداند که $y^{\intercal}Ay < 0$.

۳.۲ الگوریتم تقریبی دوم

اکنون قصد داریم از ابزار برنامه‌ریزی خطی و به طور خاص از الگوریتم بیضی‌گون استفاده کنیم و یک الگوریتم تقریبی با ضریب تقریب ۰/۸۷۸ برای مسئله‌ی برش بیشینه ارائه دهیم.

ابتدا مسئله را با برنامه‌ریزی صحیح مدل می‌کنیم. برای هر رأس $v \in V$ یک متغیر x_v درنظر می‌گیریم به این صورت که فقط می‌تواند مقدارهای ۱ و ۱− بگیرد. سپس در جواب بهینه، مجموعه‌ی رأس‌های v که $x_v = 1$ در یک دسته از برش، و بقیه‌ی رأس‌ها در سمت دیگر قرار می‌گیرند. تابع هدف نیز باید به گونه‌ای باشد که هرگاه برای یک یال $uv \in E$ داشته باشیم $x_u \neq x_v$ (یعنی یکی از x_u و x_v برابر با ۱ و دیگری برابر با ۱− باشد) حتماً وزن یال uv در تابع هدف جمع زده شود. اگر هم $x_u = x_v$، نباید وزن یال uv در تابع هدف به حساب بیاید.
برای این منظور می‌توانیم تابع هدف را این طور تعریف کنیم:

$$\sum_{uv\in E} \frac{|x_u - x_v|}{2} w(uv)$$

که به وضوح در حالتی که $x_u = x_v$، الزاماً ضریب $w(uv)$ در تابع هدف صفر خواهد بود و اگر هم $x_u \neq x_v$، الزاماً ضریب $w(uv)$ یک خواهد بود. یعنی این تابع هدف دقیقاً جمع وزن یال‌های برش را می‌دهد.
اما در مسائلی که بیشینه‌سازی هستند، معمولاً نمی‌خواهیم قدرمطلق ظاهر شود و در این مدل‌سازی نیز برای اینکه قدرمطلق را حذف کنیم تا تابع هدف، یک تابع خطی برحسب متغیرها شود، مجبوریم متغیرهای جدیدی تعریف کنیم. اما در اینجا تعریف متغیر جدید، ممکن است از لحاظ ترکیبیاتی برای مسئله، متناظر با عنصر مشخصی نباشد. پس این مدل را رها می‌کنیم و سعی می‌کنیم مدل دیگری ارائه کنیم که درنهایت شهود ترکیبیاتی خوبی داشته باشد. برای این منظور تابع هدف را به صورت زیر درنظر می‌گیریم:

$$\sum_{uv\in E} \frac{1 - x_u x_v}{2} w(uv)$$

به سادگی می‌توان بررسی کرد این تابع، معادل با همان صورتی است که مدنظر ما بود. پس تا اینجای کار به طور کلی توانستیم مسئله‌ی برش بیشینه را با یک برنامه‌ریزی ریاضی به صورت زیر مدل کنیم:

$$\begin{aligned} \text{بیشینه کن} \quad & \sum_{uv\in E} \frac{1 - x_u x_v}{2} w(uv) \\ \text{که} \quad & x_u \in \{1, -1\} \quad \forall u \in V \end{aligned} \tag{1}$$

این مدلِ ریاضی که پیدا کردیم هنوز با برنامه‌ریزی خطی خیلی فاصله دارد. متغیرهای آن فقط می‌توانند مقدارهای ۱ و ۱− اخذ کنند و تابع هدف نیز برحسب متغیرها خطی نیست. البته با توجه به اینکه مسئله‌ی برش بیشینه، ان‌پی‌ــسخت است، انتظار نداشتیم که بتوانیم دقیقاً یک برنامه‌ریزی خطی متناظر با مسئله بیابیم.
از این به بعد برای سادگی فرض کنید $n = |V|$ و رأس‌های گراف را با $1, 2, \ldots, n$ نشان می‌دهیم.
ابتدا برای این که مسئله‌ی خطی نبودن تابع هدف را حل کنیم n^2تا متغیر جدید تعریف می‌کنیم، به این صورت که برای هر $1 \leq i, j \leq n$ متغیر a_{ij} را برابر با $x_i x_j$ درنظر می‌گیریم. اگر این متغیرها را در یک ماتریس $n \times n$ قرار دهیم، همچنین x_iها را در یک ماتریس ستونی $n \times 1$ بگذاریم، خواهیم داشت $A = xx^{\intercal}$. زیرا درایه‌ی ijام از ماتریس A دقیقاً برابر است با $a_{ij} = x_i x_j$.
همچنین چون متغیرهای x_i فقط مقدارهای ۱ و ۱− می‌گیرند، اعداد روی قطر اصلی ماتریس A برابر با ۱ خواهند بود. پس کلاً می‌توانیم مسئله را

به صورت زیر نیز بیان کنیم:

$$\text{بیشینه کن} \quad \sum_{ij\in E} \frac{۱-a_{ij}}{۲} w(ij)$$
$$\text{که} \quad \text{وجود دارد بردار } x \text{ که } A = xx^{\intercal}$$
$$\text{اعداد روی قطر اصلی } A \text{ همگی برابر ۱ اند.}$$

اکنون مسئله را به نوعی آرام‌سازی می‌کنیم. به این صورت که به جای بردار x فرض می‌کنیم ماتریس $n \times n$ مثل Z وجود دارد که $A = ZZ^{\intercal}$. این کار به این دلیل نوعی آرام‌سازی است که اگر ماتریس Z را به صورت زیر درنظر بگیریم:

$$Z = \begin{bmatrix} x_۱ & ۰ & \cdots & ۰ \\ x_۲ & ۰ & \cdots & ۰ \\ \vdots & \vdots & \ddots & \vdots \\ x_n & ۰ & \cdots & ۰ \end{bmatrix}$$

آنگاه $ZZ^{\intercal}$ دقیقاً برابر می‌شود با $xx^{\intercal}$. یعنی اگر به جای بردار ستونی x یک ماتریس $n \times n$ درنظر بگیریم فضای شدنی مسئله احتمالاً بزرگتر می‌شود.
درواقع سطرهای ماتریس Z تشکیل n بردار مثل $z_۱, z_۲, \ldots, z_n$ می‌دهند به طوری که ضرب داخلی بردار z_i در z_j یعنی $\langle z_i, z_j \rangle$ همان درایه‌ی ijام از ماتریس A را تشکیل می‌دهد. همچنین اینکه درایه‌های روی قطر اصلی A همگی برابر با ۱ اند، معادل با این است که بردارهای z_i روی کره‌ی واحد قرار داشته باشند. زیرا درایه‌ی iiام از ماتریس A برابر است با $۱ = \|z_i\|^۲ = \langle z_i, z_i \rangle$ پس نرم اقلیدسی همه‌ی z_iها باید برابر با ۱ باشد یعنی باید روی کره‌ی واحد قرار داشته باشند.
پس می‌توان مسئله‌ی آرام‌سازی شده را به این صورت بیان کرد:

$$\text{بیشینه کن} \quad \sum_{ij\in E} \frac{۱-a_{ij}}{۲} w(ij)$$
$$\text{که} \quad \text{وجود دارد ماتریس } Z \in \mathbb{R}^{n\times n} \text{ که } A = ZZ^{\intercal}$$
$$\text{اعداد روی قطر اصلی } A \text{ همگی برابر ۱ اند.}$$

حال بنابر قضیه‌ی ۱ چنین ماتریس Zای وجود دارد که $A = ZZ^{\intercal}$ اگر و تنها اگر ماتریس A مثبت نیمه‌معین باشد، یعنی برای هر بردار $y \in \mathbb{R}^n$ داشته باشم $y^{\intercal}Ay \geq ۰$. بنابراین می‌توان مسئله را با استفاده از بی‌نهایت قید به این صورت نوشت:

$$\begin{aligned} \text{بیشینه کن} \quad & \sum_{ij\in E} \frac{۱-a_{ij}}{۲} w(ij) \\ \text{که} \quad & y^{\intercal}Ay \geq ۰ \quad \forall y \in \mathbb{R}^n \\ & a_{ii} = ۱ \qquad \forall ۱ \leq i \leq n \end{aligned} \tag{۲}$$

مدل بالا یک برنامه‌ریزی خطی است. درواقع هر کدام از قیدهایی که به فرم $y^{\intercal}Ay \geq ۰$ اند برحسب متغیرهای مسئله که همان درایه‌های ماتریس A هستند، خطی‌اند. یعنی برای هر بردار ثابت y قید متناظر با این بردار برابر است با $\sum_{\substack{۱\leq i\leq n \\ ۱\leq j\leq n}} (y_i y_j)\, a_{ij} \geq ۰$ که برحسب متغیرهای a_{ij} خطی است.

همچنین تابع هدف این مسئله نیز برحسب متغیرهای a_{ij} خطی است.
حال از روش بیضی‌گون استفاده می‌کنیم. مسئول برنامه‌ریزی خطی در اینجا از الگوریتم چولسکی استفاده می‌کند. این مسئول، در هر مرحله یک ماتریس A (که به عنوان نقطه‌ای از فضای $\mathbb{R}^{n\times n}$ درنظر می‌گیریم) را دریافت می‌کند و ابتدا بررسی می‌کند که آیا تمام درایه‌های روی قطر اصلیِ آن برابر با یک هستند یا خیر. سپس بررسی می‌کند که آیا متقارن هست یا خیر و درنهایت با الگوریتم چولسکی تعیین می‌کند که ماتریس A مثبت نیمه‌معین هست یا خیر و اگر نبود یک بردار y برمی‌گرداند که $y^{\intercal}Ay < ۰$ که دقیقاً یکی از قیدهایی است که نقطه‌ی A، آن را نقض می‌کند. دقت کنید که این مسئول، کار خود را در زمان چندجمله‌ای انجام می‌دهد.
پس الگوریتم بیضی‌گون با استفاده از مسئول برنامه‌ریزی خطی که معرفی کردیم اجرا می‌شود و به این صورت کلاً در زمان چندجمله‌ای یک جواب شدنی برای این مسئله پیدا می‌شود. یعنی در نهایت وقتی یک نقطه‌ی شدنی مثل A پیدا شد، الگوریتم تجزیه‌ی چولسکی ماتریس Z را طوری می‌یابد که $A = ZZ^{\intercal}$.

نکته‌ی قابل توجه دیگری که وجود دارد این است که نقطه‌ی A که یافتیم فقط یک نقطه‌ی شدنی برای مسئله بود نه یک نقطه‌ی بهینه. برای حل این مشکل می‌توانیم هر دفعه یک قید به صورت $\sum_{ij\in E}\frac{1-a_{ij}}{2}w(ij)\geq K$ به مجموعه‌ی قیدهای مسئله اضافه کنیم و به دنبال یک نقطه‌ی شدنی بگردیم. اگر حداقل یک نقطه‌ی شدنی پیدا کردیم یعنی جواب بهینه بیشترمساویِ K است. اگر هم هیچ جواب شدنی برای مسئله پیدا نشد یعنی جواب بهینه از K کمتر است. حال می‌توان با استفاده از نوعی جست‌وجوی دودویی روی عدد K، به دلخواه به جواب بهینه نزدیک شد و ماتریس A را به گونه‌ای پیدا کرد که برای مسئله، شدنی باشد و همچنین مقدار تابع هدف به ازای آن به دلخواه به جواب بهینه نزدیک باشد. این تقریب زدن‌ها را در همین حد شهودی درنظر می‌گیریم و به بررسی دقیق آن نمی‌پردازیم و از این به بعد فرض می‌کنیم ماتریس A دقیقاً یک جواب بهینه برای مسئله‌ی ۲ است که $A=ZZ^{T}$.

حال درنهایت اگر سطرهای ماتریس Z پیداشده را $z_1, z_2, \dots, z_n$ بنامیم، این بردارها روی کره‌ی واحد قرار دارند و مقدار $\sum_{ij\in E}\frac{1-\langle z_i,z_j\rangle}{2}w(ij)$ بیشینه است که معادل است با اینکه مقدار $\sum_{ij\in E}\frac{w(ij)}{2}\langle z_i,z_j\rangle$ کمینه باشد.

می‌توان به طور شهودی تصور کرد که برای هر $ij\in E$، یک فنر بین دو انتهای بردارهای z_i و z_j وصل شده است که ضریب سختی فنر به $w(ij)$ بستگی دارد و این فنر باعث می‌شود زاویه‌ی بین دو بردار z_i و z_j زیاد شود تا مقدار تابع هدف کمتر شود. در نهایت نیز بردارهای z_iای که متناظر با جواب بهینه‌اند درواقع نقطه‌ی تعادل برای مجموعه‌ی فنرها و بردارهای روی کره است.

تا اینجا تعدادی بردار پیدا کردیم که به نوعی یک جواب بهینه برای مسئله‌ی ۲ اند. اما هدف ما پیدا کردن یک برش بیشینه برای گراف بود. دقت کنید که هر بردار z_i در مسئله‌ی جدید به نوعی متناظر بود با متغیر x_i در مسئله‌ی ۱ که برای برش بیشینه داشتیم. اکنون باید با استفاده از z_iها مقدارهای x_i را تعیین کنیم، یعنی یک برش ارائه دهیم.

برای این منظور یک ابرصفحه به طور تصادفی و یکنواخت[9] از مبدأ فضای $\mathbb{R}^n$ می‌گذرانیم. این ابرصفحه، کره‌ی واحد را به دو بخش تقسیم می‌کند. سپس بردارهای z_i را برحسب اینکه در کدام بخش از کره قرار می‌گیرند به دو دسته تقسیم می‌کنیم و متناظر با آن نیز متغیرهای x_i به دو دسته تقسیم می‌شوند. یعنی با این روش به یک برش برای گراف می‌رسیم.

مثلاً فرض کنید یک گراف دوبخشی داریم. در این صورت بردارهای متناظر با تمام رأس‌های یک بخش در یک نقطه از کره جمع می‌شوند و بردارهای متناظر با رأس‌های بخش دیگر دقیقاً در روبروی قطری نقطه‌ی قبلی در کره قرار خواهند گرفت. حال اگر یک ابرصفحه‌ی دلخواه، کره را نصف کند با احتمال نزدیک به یک، این دو نقطه‌ی روبروی قطری از کره در دوسمت مختلف از ابرصفحه قرار خواهند گرفت و الگوریتم بیان شده دقیقاً برش بیشینه را انتخاب می‌کند.

پس درکل ابتدا یک برنامه‌ریزی ریاضی برای مسئله‌ی برش بیشینه بیان کردیم. سپس آن را به یک مسئله‌ی برنامه‌ریزی خطی گسترش دادیم که به نوعی آرام‌سازی شده‌ی برنامه‌ریزی ریاضی بود (یعنی فضای شدنی مسئله احتمالاً بزرگ‌تر شده). در این صورت جواب بهینه‌ی مسئله‌ی برنامه‌ریزی خطی از جواب بهینه‌ی برنامه‌ریزی ریاضی که همان برش بیشینه است، بدتر نیست. سپس یک جواب بهینه برای برنامه‌ریزی خطی یافتیم و متناظر با آن، n بردار روی کره‌ی واحد درنظر گرفتیم که هر بردار متناظر با یک رأس از گراف اولیه است. درنهایت یک ابرصفحه‌ی تصادفی و یکنواخت از مبدأ فضای $\mathbb{R}^n$ گذراندیم که کره‌ی واحد و درنتیجه بردارهای یافته شده را به دوبخش تقسیم می‌کند. سپس این دو بخش را به عنوان یک برش در گراف بیان می‌کنیم. دقت کنید که تمام این مراحل در زمان چندجمله‌ای قابل انجام‌اند.

اکنون به تحلیل الگوریتم می‌پردازیم و اثبات می‌کنیم امید ریاضی وزن برش تصادفی پیدا شده از ۰٫۸۷۸ برابرِ وزن برش بیشینه کمتر نیست. در این صورت یک الگوریتم تقریبی با زمان اجرای چندجمله‌ای و با ضریب تقریب ۰٫۸۷۸ ارائه کرده‌ایم.

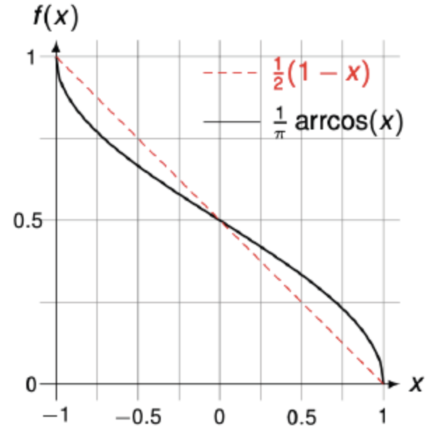
برای این منظور برای هر $ij\in E$ یک متغیر تصادفی شاخص y_{ij} تعریف می‌کنیم که برابر با یک است اگر و تنها اگر یال ij در برش انتخابی آمده باشد. ابتدا احتمال این را می‌یابیم که بردارهای z_i و z_j در دو سمتِ مختلف از ابرصفحه‌ی تصادفی قرار داشته باشند. درواقع احتمال اینکه $y_{ij}=1$ باشد را می‌یابیم. صفحه‌ای که از دو بردار z_i و z_j می‌گذرد را درنظر می‌گیریم همچنین زاویه‌ی بین دو بردار z_i و z_j را α_{ij} می‌نامیم. محل برخورد صفحه‌ی گذرنده از z_i و z_j و کره‌ی واحد، یک دایره می‌شود. ابرصفحه‌ی تصادفیِ نصف‌کننده‌ی کره، این دایره را نیز دقیقاً نصف می‌کند و درواقع می‌توان دید که برای محاسبه‌ی احتمال $y_{ij}=1$، کافی است فرض کنیم در این دایره یک قطر به طور تصادفی و یکنواخت انتخاب شده است و احتمال این را بیابیم که دو بردار در دو سمت مختلف از این قطر در دایره قرار داشته باشند. این احتمال نیز برابر است با $\frac{\alpha_{ij}}{\pi}$ که با استفاده از ضرب داخلی برابر است با $\frac{1}{\pi}\arccos(\langle z_i,z_j\rangle)$. پس درکل می‌توانیم امید ریاضی تابع هدف یعنی امید ریاضی متغیر تصادفیِ $\sum_{ij\in E}y_{ij}w_{ij}$ را بیابیم. داریم:

$$\mathrm{E}\left[\sum_{ij\in E}y_{ij}w_{ij}\right]=\sum_{ij\in E}\mathrm{E}\left[y_{ij}\right]\times w_{ij}=\sum_{ij\in E}\Pr\left[y_{ij}=1\right]\times w_{ij}=\sum_{ij\in E}\frac{1}{\pi}\arccos(\langle z_i,z_j\rangle)w_{ij}$$

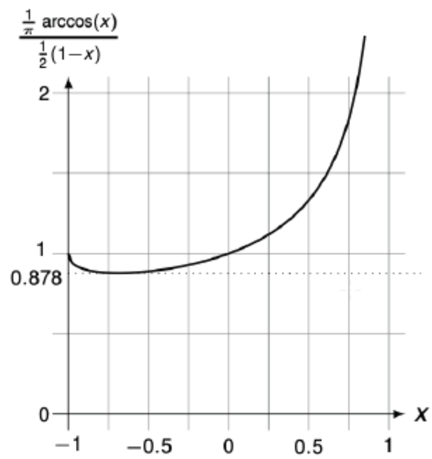
حال می‌خواهیم هرکدام از $\frac{1}{\pi}\arccos(\langle z_i,z_j\rangle)$ها را با استفاده از ضریبی از $1-\langle z_i,z_j\rangle$ تقریب بزنیم که در تابع هدف مسئله‌ی ۲ ظاهر شده

[9] Uniform

است. برای این منظور با توجه به شکل می‌بینیم که نمودارهای توابع $\frac{1}{\pi}\arccos(x)$ و $\frac{۱}{۲}(۱-x)$ بسیار نزدیک به یکدیگرند.



پس اگر نسبت این دو تابع را درنظر بگیریم و یک تخمین از کمینه‌ی نسبت آن‌ها داشته باشیم، تخمین موردنظرمان بدست می‌آید. در شکل زیر نمودار تابع $\frac{\frac{۱}{\pi}\arccos(x)}{\frac{۱}{۲}(۱-x)}$ رسم شده است که کمینه‌ی آن از عدد ۰٫۸۷۸ بیشتر است.



البته برای اثبات دقیق، باید با استفاده از بسط تیلور این توابع، مطلب بالا را نشان داد اما به همین حد بسنده می‌کنیم و اثبات دقیق را به خواننده واگذار می‌کنیم.

درنهایت خواهیم داشت:

$$\begin{aligned}
\mathrm{E}\left[\sum_{ij\in E} y_{ij}w_{ij}\right] &= \sum_{ij\in E}\frac{۱}{\pi}\arccos(\langle z_i,z_j\rangle)w_{ij}\\
&\geq \sum_{ij\in E} ۰٫۸۷۸\frac{۱-\langle z_i,z_j\rangle}{۲} = ۰٫۸۷۸\times(\text{جواب بهینه‌ی مسئله‌ی ۲})\\
&\geq ۰٫۸۷۸\times(\text{جواب بهینه‌ی مسئله‌ی ۱}) = ۰٫۸۷۸\times(\text{وزن برش بیشینه در گراف})\\
\Rightarrow \mathrm{E}\left[\sum_{ij\in E} y_{ij}w_{ij}\right] &\geq ۰٫۸۷۸\times(\text{وزن برش بیشینه در گراف})
\end{aligned}$$

بنابراین کلاً اثبات شد که الگوریتم بیان شده یک الگوریتم با ضریب تقریب ۰٫۸۷۸ است.

به طور کلی به این روش، تکنیک گردکردن[۱۰] می‌گویند که با استفاده از جواب مسئله‌ی آرام‌سازی شده بتوانیم جواب تقریبی از جواب مسئله اصلی بیابیم.

۳ ارجاع و منابع

ویدئوی جلسه‌ی دوازدهم [دانلود]

[10] rounding