دانشکده علوم ریاضی
دانشگاه صنعتی شریف

کاربرد برنامه‌ریزی ریاضی در تولید الگوریتم‌های تقریبی

محمدهادی فروغمنداعرابی
پاییز ۱۳۹۶

مساله چند برش، برش متوازن

جلسه بیست ویکم

نگارنده: فائزه فاطمی نژاد

در جلسه‌ي قبل با استفاده از برنامه‌ریزی خطی برای مسأله برش چند طرفه کمینه یک الگوریتم تقریبی به دست آوردیم. در این جلسه دو مسأله برش دیگر را بررسی می‌کنیم.

۱ مساله‌ی چند برش

گراف وزن‌دار G و k زوج رأس (s_i, t_i) داده شده‌است. وزن هر یال $c_e \geq 0$ است و می‌خواهیم مجموعه F از یال‌های آن را حذف کنیم به طوری که برای هر i از ۱ تا k بین s_i و t_i مسیری وجود نداشته باشد و وزن F کمینه باشد.

توجه داشته باشید که مساله برش چند طرفه را می‌توان به این مساله تبدیل کرد به این صورت که هر دوتایی از s_iها را در یک زوج مرتب قرار می‌دهیم. همچنین این مسأله دوگانی برای مسأله جنگل اشتاینر است. برنامه ریزی صحیح مسأله به شکل زیر است:

$$\begin{aligned} & min \sum_e x_e c_e & \\ & \sum_{e \in P} x_e \geq 1 & \forall P \in \mathcal{P}_i, 1 \leq i \leq k, \\ & x_e \in \{0, 1\} & \forall e \in E \end{aligned}$$

که $\mathcal{P}_i$ مجموعه تمام مسیرهای بین s_i و t_i در گراف است. می‌توانیم این برنامه‌ریزی را با وانهش به برنامه‌ریزی خطی تبدیل کنیم:

$$\begin{aligned} & min \sum_e x_e c_e \\ & \sum_{e\in P} x_e \geq ۱ && \forall P \in \mathcal{P}_i, ۱ \leq i \leq k, \\ & x_e \geq ۰ && \forall e \in E \end{aligned}$$

چون تعداد شرط‌ها نمایی است از روش ellipsoid استفاده می‌کنیم. به یک اوراکل جداکننده نیاز داریم که با فرض اینکه طول هر یال x_e باشد، در زمان چند جمله‌ای کوتاهترین فاصله بین هر (s_i, t_i) را محاسبه کند. در این صورت اگر کوتاهترین فاصله بین دو رأس s_i و t_i کمتر از ۱ باشد یک شرط نقض شده داریم و در صورتی که برای تمام iها این فاصله بیشتر یا مساوی ۱ باشد یک جواب شدنی برای برنامه‌ریزی خطی داریم.

تعبیر دیگری از این مسأله این است که یال‌ها را لوله‌های آکاردئونی با سطح مقطع c_e و طول x_e در نظر بگیریم و بخواهیم مجموع حجم لوله‌ها کمینه شود به طوری که طول هر مسیر بین s_i و t_i حداقل برابر ۱ باشد. در این صورت $d(u, v)$ را کوتاهترین فاصله بین دو رأس u و v تعریف می‌کنیم.

برای گردکردن جواب برنامه‌ریزی خطی از الگوریتم زیر استفاده می‌کنیم:

الگوریتم:

الف) $LP \rightarrow X^*$ را حل کن

ب) برای هر i از ۱ تا k که هنوز بین s_i وt_i مسیر وجود دارد:

یک گوی به طول r دور s_i در نظر بگیر

یال‌هایی که این گوی را قطع می‌کنند در F قرار بده

رأس‌های درون این گوی را از گراف حذف کن

تحلیل الگوریتم:

برای این که یک الگوریتم تقریبی داشته باشیم دوست داریم مجموع حجم یال‌های حذف شده در هر مرحله کمتر یا مساوی ضریبی از حجم لوله های درون آن گوی باشد. بنابراین به دنبال rی هستیم که بتوانیم در مورد آن بگوییم:

$$c(\delta(B(s_i, r))) \leq qV(s_i, r)$$

که $B(s_i, r)$ مجموعه تمام رأس‌هایی است که در گوی s_i می افتند یعنی هنوز از گراف حذف نشده اند و فاصله آن‌ها از s_i کمتر یا مساوی r است. q ضریبی است که به دنبالش هستیم و $V(s_i, r)$ جمع حجم لوله‌های درون گوی s_i است. به عبارتی:

$$V(s_i, r) = \sum_{e:u,v \in B(s_i,r)} c_e x_e + \sum_{e:v\in B(s_i,r), u \notin B(s_i,r)} (r - d(s_i, v))c_e$$

که $(r - d(s_i, v))c_e$ حجم قسمتی از یال e است که درون گوی قرار دارد.

طبق شرط‌های برنامه ریزی خطی باید r کمتر مساوی ۱ باشد چون $d(s_i, t_i) \geq ۱$ است و می‌خواهیم t_i در گوی s_i نیفتد. همچنین برای اینکه هنگام حذف رأس های گوی یک s_i مطمئن باشیم که s_j و t_j دیگری با هم حذف نمی شوند و مسیر بینشان در جواب باقی بماند، r را کمتر از $\frac{۱}{۲}$ در نظر می گیریم.

اگر چنین rی وجود داشته باشد، الگوریتم qـ تقریب است.

در این جا برای سادگی $V(s_i, r)$ را $V(r)$ و هزینه یال‌های قطع کننده گوی را $c(r)$ می‌نامیم. مقدار $V(r)$ در $r = ۰$ صفر و در $r = \frac{۱}{۲}$ حداکثر V^* است. این تابع غیرنزولی است و نرخ رشد آن در لحظه صفر $c(\delta(B(s_i, r)))$ است تا زمانی که به انتهای یک یال برسیم. هرگاه یالی به $\delta(B(s_i, r))$ اضافه یا از آن کم شود نرخ رشد تغییر می کند. پس تابع $V(r)$ مشتق‌پذیر نیست. از طرفی اگر دو سر یک یال همزمان به گوی وارد شوند حجم یال در یک لحظه به $V(r)$ اضافه می‌شود و تابع پرش دارد. اما در آن نقاطی که پیوسته است داریم: $\frac{dV(r)}{dr} = c(r)$.

حال برای یک r تصادفی $\mathbb{E}[\frac{c(r)}{V(r)}]$ را حساب می‌کنیم:

$$\begin{aligned}\mathbb{E}[\frac{c(r)}{V(r)}] &= \frac{1}{\frac{1}{2}}\sum_{\text{قطعات}}\int \frac{c(x)}{V(x)}dx\\ &= 2.\sum_{i=0}^{l-1}\int_{r_i}^{r_{i+1}-\epsilon}\frac{c(x)}{V(x)}dx\\ &= 2.\sum_{i=0}^{l-1}\int_{r_i}^{r_{i+1}-\epsilon}\frac{\frac{dV(x)}{d(x)}}{V(x)}dx\\ &= 2.\sum_{i=0}^{l-1}[lnV(r)]_{r_i}^{r_{i+1}-\epsilon}\\ &= 2.\sum_{i=0}^{l-1}[lnV(r_{i+1}-\epsilon)-lnV(r_i)]\end{aligned}$$

زمانی که ϵ به صفر میل می‌کند داریم:

$$\begin{aligned}\mathbb{E}[\frac{c(r)}{V(r)}] &\leq 2.\sum_{i=0}^{l-1}[lnV(r_{i+1})-lnV(r_i)]\\ &= 2.(lnV(r_l)-lnV(r_0))\\ &= 2.(lnV(\frac{1}{2})-lnV(0))\end{aligned}$$

$V(\frac{1}{2})=V^*$است اما $V(0)=0$. پس همه چیز بیهوده شد! برای اینکه درست شود $V(s_i,r)$ را به اندازه $\frac{V^*}{k}$ اضافه می‌کنیم:

$$V(s_i,r)=\frac{V^*}{k}+\sum_{e:u,v\in B(s_i,r)}c_ex_e+\sum_{e:v\in B(s_i,r),u\notin B(s_i,r)}(r-d(s_i,v))c_e$$

بنابراین:

$$\begin{aligned}\mathbb{E}[\frac{c(r)}{V(r)}] &= 2.(lnV(\frac{1}{2})-lnV(0))\\ &= 2.(lnV^*-ln\frac{V^*}{k}) = 2(ln(k+1))\end{aligned}$$

حال چون با تعریف جدید داریم $\sum_{i=1}^{k}V_i\leq V^*$ و می‌دانستیم که $V(s_i,r)-\frac{V^*}{k}\leq V_i$ می‌توانیم بگوییم:

$$\begin{aligned}\sum_{e\in F}c_e=\sum_{i=1}^{k}\sum_{e\in F_i}c_e &\leq (2ln(k+1))\sum_{i=1}^{k}(V_i+\frac{V^*}{k})\\ &\leq (4ln(k+1))V^*\leq (4(ln(k+1))OPT\end{aligned}$$

پس این الگوریتم $4ln(k+1)$ـ تقریب است. اگر UGC [1] صحیح باشد، برای این مسأله الگوریتم تقریبی با ضریب عدد ثابت وجود ندارد.

نکته‌ای در مورد به دست آوردن r: زمان‌هایی که هیچ یال جدیدی به گوی وارد نمی‌شود یا یالی از گوی خارج نمی‌شود، میزان $c(r)$ ثابت است و $V(r)$ زیاد می‌شود بنابراین درست قبل از اینکه یالی به گوی اضافه شود $\frac{c(r)}{V(r)}$ کمینه است پس تعداد محدودی انتخاب برای r داریم و فقط همان نقاط را بررسی می‌کنیم و کمینه آن‌ها را در نظر می‌گیریم.

[1] Unique Game Conjecture

۲ مسأله برش متوازن

گراف G را داریم و می‌خواهیم برش کمینه‌ای از آن پیدا کنیم به طوری که به دو دسته با تعداد رأس‌های تقریبا برابر تبدیل شود.
تعریف: گراف را bـ متوازن می‌گوییم هرگاه:

$$\lfloor bn \rfloor \leq |S| \leq \lceil (۱-b)n \rceil$$

برای مثال :
$\frac{۱}{۲}$ ـ متوازن: $\lfloor \frac{n}{۲} \rfloor \leq |S| \leq \lceil \frac{n}{۲} \rceil$
$\frac{۱}{۳}$ ـ متوازن: $\lfloor \frac{n}{۳} \rfloor \leq |S| \leq \lceil \frac{۲n}{۳} \rceil$
اگر $OPT(\frac{۱}{۲})$ جواب بهینه‌ی $|S|$ $\frac{۱}{۲}$ ـ متوازن و $OPT(\frac{۱}{۳})$ جواب بهینه‌ی $|S|$ $\frac{۱}{۳}$ ـ متوازن باشد خواهیم داشت: $OPT(\frac{۱}{۳}) \leq OPT(\frac{۱}{۲})$ دوست داریم $\frac{۱}{۲}$ ـ متوازن را حل کنیم اما چون نمی‌توانیم مساله را برای $\frac{۱}{۳}$ ـ متوازن حل می‌کنیم. هدف این است که یک S پیدا کنیم که $\frac{۱}{۳}$ ـ متوازن باشد و بگوییم از ضریبی از $OPT(\frac{۱}{۲})$ بیشتر نیست.
برنامه ریزی خطی را برای $OPT(\frac{۱}{۲})$ می‌نویسیم:

$$\begin{aligned}
& min \sum_{e} c_e x_e \\
& d_{uv} \leq \sum_{e \in P} x_e && \forall u, v, P \\
& (\frac{۲}{۳} - \frac{۱}{۲})n \leq \sum_{v \in S} duv && \forall u \in S, \forall S : |S| \geq \lceil \frac{۲n}{۳} \rceil + ۱ \\
& d_{uv}, x_e \geq ۰
\end{aligned}$$

که P مسیر بین u و v است.
قضیه: زمانی که S $\frac{۱}{۲}$ ـ متوازن است می‌توان x و d را پیدا کرد به طوری که یک جواب شدنی برای برنامه ریزی خطی بالا و مقدار تابع هدف همان وزن برش باشد. در این صورت این برنامه‌ریزی خطی یک وانهش از مسأله برش $\frac{۱}{۲}$ ـ متوازن است.
اثبات: فرض کنید یک برش $\frac{۱}{۲}$ ـ متوازن داریم که رأس‌های گراف را به دو مجموعه‌ی A و B افراز کرده‌اند و اندازه هر کدام از این دو مجموعه $\frac{n}{۲}$ است. x_e تمام یال‌های بین A و B و d_{uv} هر دو رأسی که در دو دسته متفاوت قرار دارند را برابر ۱ و بقیه x_e و d_{uv}ها را صفر می‌گذاریم. حال به هر ترتیبی که یک مجموعه S با اندازه حداقل $\lceil \frac{۲n}{۳} \rceil + ۱$ انتخاب کنیم از هر دسته حداقل $\frac{n}{۶}$ رأس عضو S هستند بنابراین برای هر رأس v عضو S حداقل $\frac{n}{۶}$ رأس در دسته مقابلش قرار دارد که عضو S هستند. بنابراین $\frac{n}{۶} \leq \sum_{u \in S} d_{uv}$. همچنین هر دو رأس با $d_{uv} \geq ۰$ که در نظر بگیریم، اگر بین آن‌ها مسیری وجود داشته باشد حداقل یک یال این مسیر یالی است که یک سرش در A و سر دیگرش در B است و x_e آن برابر ۱ است. پس یک جواب شدنی برای برنامه ریزی خطی به دست آورده‌ایم.
ایده الگوریتم:
الف) مجموعه S را تهی در نظر بگیر
ب)$LP \rightarrow d, x$ را حل کن
پ) تا زمانی که $|S| < \lfloor \frac{n}{۳} \rfloor$:
رأس خوبی در نظر بگیر و یک گوی دور آن بزن و تمام رأس های درون آن گوی را به S اضافه کن

برای اینکه $|S|$ شرایط مسأله را داشته باشد، در هر مرحله دو رأس در نظر می گیریم که به اندازه کافی دور باشند و گوی هایشان به هم نرسد. سپس آن گوی که تعداد کمتری رأس دارد را به S اضافه می‌کنیم.