دانشکده‌ی علوم ریاضی

آنالیز الگوریتم‌ها ۸ اردیبهشت ۹۲

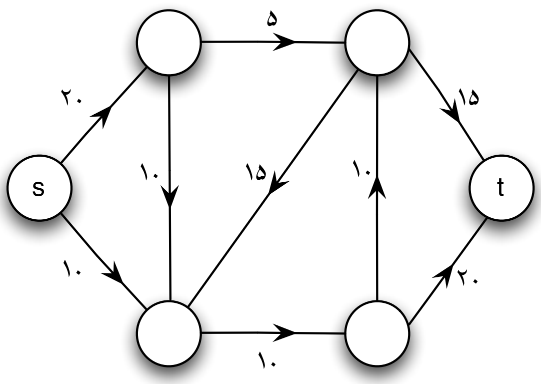
جلسه‌ی ۱۶: جریان‌دهی بیشینه در گراف

مدرّس: دکتر شهرام خزائی *نگارنده: علی کوچک زاده*

۱ مقدمه

به یک گراف جهت‌دار با یال‌های وزن دار به طوری که یک رأس مبدأ و یک رأس مقصد داشته باشد، شبکه‌ی جریان[1] می‌گوییم. وزن هر یال، نشان دهنده‌ی ظرفیت جریان عبوری از آن یال می‌باشد و جهت هر یال نشان دهنده‌ی جهت قابل عبور جریان است. در مسأله‌ی جریان بیشینه[2] می‌خواهیم حداکثر جریانی را پیدا کنیم که می‌توان از مبدا به مقصد منتقل کرد، به طوری که از هر یال حداکثر به اندازه‌ی ظرفیت آن و در جهت مجاز جریان عبور کند.

مثال ۱ *به عنوان مثال در گراف زیر می‌خواهیم بیشترین جریان از رأس s (مبدأ) به رأس t (مقصد) برسد به طوری که از هر یال حداکثر به اندازه‌ی ظرفیتش جریان عبور کند. (ظرفیت هر یال در کنار آن نوشته شده است.)*



شکل ۱: یک شبکه‌ی جریان

۲ تعاریف

تعریف ۱ *جریان‌دهی برای گراف وزن‌دار $G = (V, E)$ با ظرفیت C_e برای یال $e \in E$ به معنی پیدا کردن تابع $f : E \rightarrow \mathbb{R} \geq 0$ است به طوری که:*

۱. *شرط محدودیت جریان برای هر یال برقرار باشد؛ یعنی:* $\forall e \in E : 0 < f_e \leq C_e$.

۲. *شرط بقای جریان باید در هر رأس برقرار باشد؛ یعنی:* $\forall v \in V - \{s, t\} : \sum_u f_{uv} = \sum_w f_{vw}$

به عبارت دیگر، جمع جریان‌های ورودی به یک رأس باید با جمع جریان‌های خروجی آن برابر باشد. (s و t به ترتیب رأس‌های مبدا و مقصد هستند.)

---
[1] network flow
[2] maximum Flow

تعریف ۲ (اندازه‌ی جریان) *اندازه‌ی یک جریان‌دهی برابر با میزان جریان خارج شده از رأس شروع s است:*

$$|f| = \sum_{w} f_{sw}$$

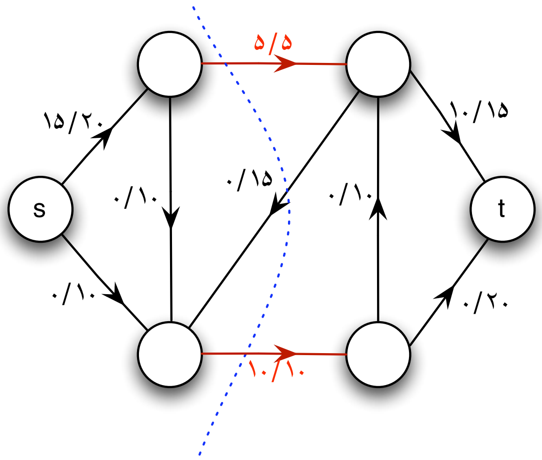
به سادگی می‌توان نشان داد که اندازه جریان خارج شده از رأس شروع با اندازه جریان وارد شده به رأس مقصد برابر است.

لم ۱ *اندازه جریان خارج شده از رأس شروع با اندازه جریان وارد شده به رأس مقصد برابر است. یعنی:* $|f| = \sum_w f_{wt}$.

تعریف ۳ (برش برای یک شبکه‌ی جریان) *افرازی از رأس‌های شبکه‌ی جریان به دو مجموعه‌ی S و T به طوری‌که $s \in S, t \in T$ را یک برش گوییم و با $\langle S,T \rangle$ نمایش می‌دهیم. ظرفیت برش برابر است با مجموع ظرفیت یال‌های عبوری از S به T و با $|\langle S,T \rangle|$ نشان داده می‌شود. به عبارت دیگر:*

$$|\langle S,T \rangle| = \sum_{vw \in E, v \in S, w \in T} C_{vw}$$

مثال ۲ *جریان‌دهی بیشینه برای گراف مثال ۱ به صورت زیر خواهد بود:*



شکل ۲: یک جریان‌دهی بیشینه

از برش رسم شده‌ی فوق به هیچ نحوی نمی‌توان جریان بیشتر از ۱۵ عبور داد (مجموع ظرفیت یال‌های عبوری از برش مبدا به مقصد). به عبارت دیگر ظرفیت این برش ۱۵ است. همچنین برش با ظرفیت بیشتری در این شبکه نمی‌توان پیدا کرد. لذا بیشترین اندازه جریان‌دهی ۱۵ می‌باشد.

تعریف ۴ (مسأله برش کمینه[۳]) *مسأله برش کمینه پیدا کردن برش با ظرفیت کمینه است.*

قضیه ۲ *بیشترین اندازه‌ی جریان‌دهی برابر است با ظرفیت برش کمینه.*

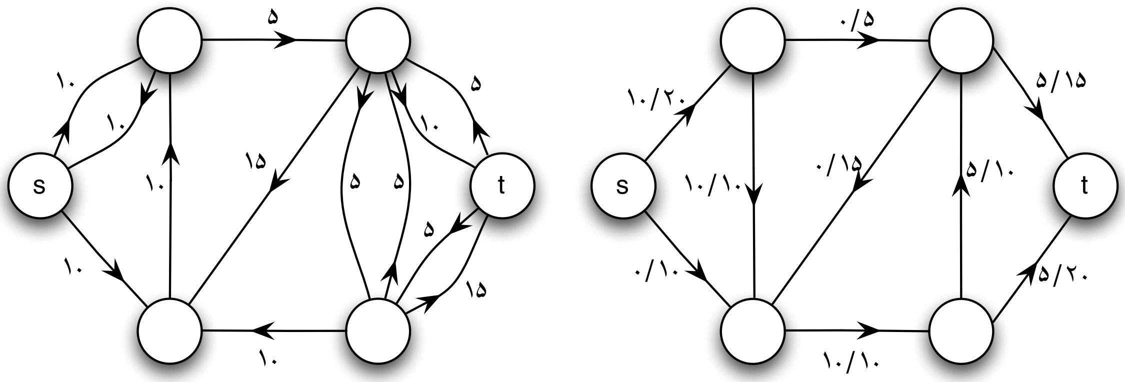
تعریف ۵ (ظرفیت پسماند[۴]) *برای یک جریان‌دهی f، ظرفیت پسماند برای هر یال را به این صورت تعریف می‌کنیم.*

$$C^f_{uv} = \begin{cases} C_{uv} - f_{uv} & uv \in E \\ f_{uv} & vu \in E \end{cases}$$

تعریف ۶ (گراف پسماند) *گراف جهت‌داری که ظرفیت یال‌های آن ظرفیت‌های پسماند باشد. در این گراف یال‌های با ظرفیت پسماند صفر را حذف می‌کنیم.*

[۳] min-cut
[۴] residual capacity

مثال ۳ *یک شبکه جریان با جریان‌دهی f و گراف پسماند متناظر با آن:*



(ب) گراف پسماند آن   (آ) جریان‌دهی f

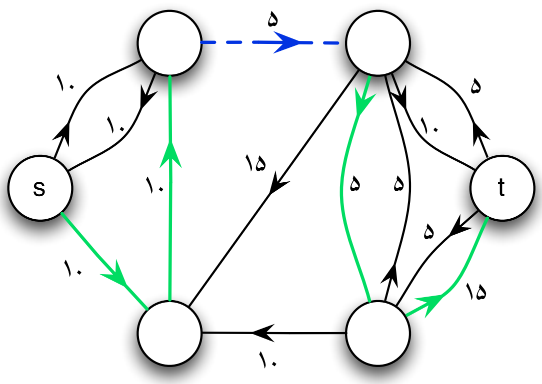
شکل ۳: گراف پسماند

تعریف ۷ *هر مسیر از s به t در گراف پسماند را یک مسیر افزاینده[5] می‌نامیم.*

تعریف ۸ *برای هر مسیر افزاینده P در گراف پسماند، گلوگاه[6] را برابر با مینیمم ظرفیت پسماند در مسیر P تعریف می‌کنیم و با B نشان می‌دهیم، یعنی:*

$$B = \min_{uv \in P} \{C^f_{uv}\}$$

مثال ۴ *در گراف پسماند مثال قبل یک مسیر افزاینده مسیر مشخص شده با رنگ سبز در شکل زیر است. همچنین گلوگاه این مسیر با خط‌چین آبی رنگ نشان داده شده است.*



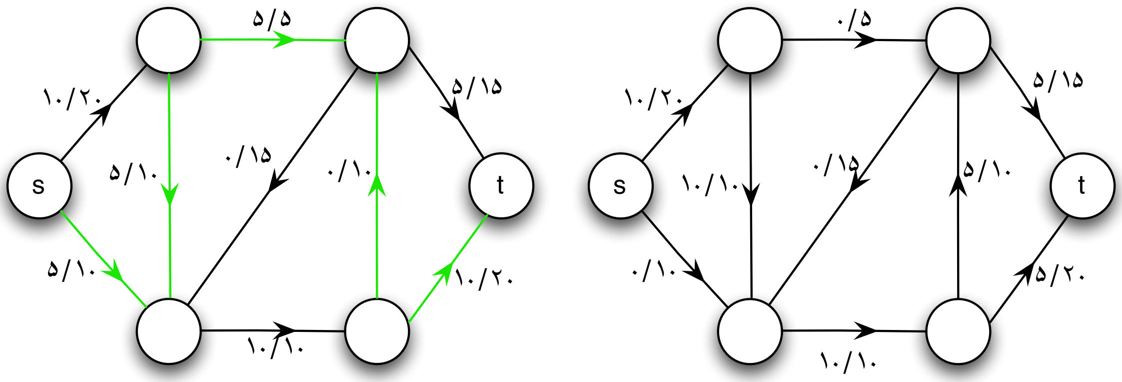
شکل ۴: یک مسیر افزاینده در گراف پسماند

تعریف ۹ *جریان افزوده متناظر با مسیر افزاینده P با گلوگاه B را به این صورت تعریف می‌کنیم:*

$$f'_{uv} = \begin{cases} f_{uv} + B & uv \in P \\ f_{uv} - B & vu \in P \\ f_{uv} & oth. \end{cases}$$

[5] augmenting path
[6] bottleneck

مثال ۵ *در صورتی که مسیر افزاینده‌ی مثال قبل را در نظر بگیریم، جریان افزوده متناظر با جریان‌دهی f در مثال ۳ به صورت زیر خواهد بود. (در اینجا $B = ۵$)*



(ب) جریان افزوده با $B = ۵$ (آ) جریان‌دهی f

شکل ۵: جریان افزوده

لم ۳ *جریان افزوده یک جریان‌دهی مجاز نسبت به ظرفیت اولیه‌ی است.*
برهان. *باید ثابت کنیم که*

۱. *مجموع جریانهای ورودی و خروجی در هر رأس برابرند.*

۲. $۰ \leq f'_{uv} \leq C_e$

گزاره‌ی اول به سادگی از روی تعریف جریان افزوده ثابت می‌شود. اگر رأسی روی یک مسیر افزاینده نباشد، مقدار جریان وردی و خروجی آن تغییر نمی‌کند. اگر رأسی روی یک مسیر افزاینده با گلوگاه B باشد، به تعداد B واحد جریان به آن وارد شده و به همان مقدار نیز خارج شده است. برای نشان دادن گزاره‌ی دوم، اگر $uv \in P$ باشد با توجه به تعریف جریان افزوده واضح است که جریان عبوری از هر یال بزرگتر مساوی صفر است:

$$uv \in P \Rightarrow f'_{uv} = f_{uv} + B \geq ۰$$

همچنین:

$$f'_{uv} = f_{uv} + B \leq f_{uv} + C^f_{uv} = f_{uv} + C_{uv} - f_{uv} = C_{uv}$$

در نتیجه $۰ \leq f'_{uv} \leq C_{uv}$. به طور مشابه نیز می‌توان نشان داد که $uv \notin P \Rightarrow ۰ \leq f'_{uv} \leq C_{uv}$. ■

۳ الگوریتم

حال با توجه به قضایا و تعریف‌های داده شده الگوریتم پیدا کردن جریان بیشینه را بیان می‌کنیم. کافی است که هر بار یک مسیر افزاینده از s به t در گراف پسماند پیدا کنیم، سپس B را بدست آورده و جریان افزوده را به گراف نسبت دهیم.
الگوریتم فورد-فالکرسون:

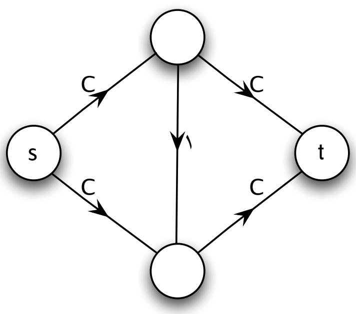
Algorithm 1 Algorithm: Ford-Fulkerson

function MaximumFlow(Graph G, Capacity C)
  [*assumes* $C = \{C_e \geq 0 : e \in E\}$]
  Start with zero flow
  while Until there is no $s \to t$ path **do**
    Augment the flow along some augmenting path.

لم ۴ *اگر f' یک جریان افزوده متناظر با جریان f و مسیر افزاینده‌ی P باشد، $|f| < |f'|$.*
برهان. *با توجه به اینکه هر بار به اندازه‌ی B به بعضی از یال‌ها اضافه شده می‌توان نتیجه گرفت که اندازه‌ی جریان‌دهی کل گراف افزایش می‌یابد.* ■

قضیه ۵ *اگر مقادیر ظرفیت‌ها صحیح نامنفی باشند، الگوریتم نهایتاً متوقف می‌شود و مقدار جریان نهایی، یک جریان بیشینه است.*

برهان. واضح است که اگر ظرفیت‌ها اعداد صحیح نامنفی باشند، مقادیر جریان در هر مرحله از الگوریتم هم یک عدد صحیح نامنفی خواهند بود. بالاخص اگر مسیر افزاینده‌ای در گراف پسماند وجود داشته باشد، مقدار گلوگاه آن یک عدد صحیح مثبت است. بنابراین، هر مسیر افزاینده دلخواه حداقل ۱ واحد به اندازه‌ی جریان‌دهی اضافه می‌کند. لذا پس از تعداد متناهی بار تکرار، الگوریتم متوقف می‌شود. اثبات بیشینه بودن اندازه‌ی جریان‌دهی (که خود نیز یک عدد صحیح نامنفی است) پس از توقف الگوریتم به خواننده واگذار می‌شود. ■
اگر جریان‌دهی بیشینه را با f^* نمایش دهیم، حداکثر با $|f^*|$ بار تکرار به جریان‌دهی بیشینه می‌رسیم. لذا پیچیدگی این الگوریتم از $\mathcal{O}(m|f^*|)$ است که در آن m تعداد یال‌های گراف می‌باشد. دقت کنید که پیچیدگی این الگوریتم نسبت به اندازه ورودی نمایی است و نه چندجمله‌ای چون اندازه‌ی $|f^*|$ را با حداکثر $\log|f^*|$ بیت می‌توان نمایش داد. مثال‌هایی وجود دارند که زمان اجرا $\Theta(|f^*|)$ خواهد بود. مثلا در گراف زیر که C یک عدد صحیح است، اگر هر بار مسیر با طول سه از s به t به عنوان مسیر افزاینده انتخاب شود، زمان اجرا $\Theta(C)$ است.



شکل ۶: مثال با زمان اجرای زیاد

نکته ۱ *اگر مقادیر ظرفیت‌ها اعدد حقیقی باشند، نه تنها ممکن است الگوریتم هیچ وقت پایان نیابد بلکه ممکن است به جریان بیشینه همگرا نشود. البته این مسأله در عمل مشکلی ایجاد نمی‌کند زیرا کامپیوتر ها اعداد حقیقی را با دقت متناهی ذخیره می‌کنند.*

برای کاهش پیچیدگی الگوریتم دو ایده مطرح شده است که پیچیدگی الگوریتم به چندجمله ای کاهش می‌یابد:

۱. یکی اینکه هر بار مسیر افزاینده‌ای را انتخاب کنیم که تعداد یال‌های کمتری داشته باشد. در این حالت به پیچیدگی $\mathcal{O}(nm^۲)$ می‌رسیم [Dinits, Edmonst-Karp, 1972].

۲. ایده‌ی دیگر این است که هر بار مسیر افزاینده‌ای را انتخاب کنیم که بیشترین مقدار گلوگاه را داشته باشد. در این صورت الگورتیم در زمان $\mathcal{O}(m^۲ \log m \log|f^*|)$ قابل پیاده سازی است [Edmonst-Karp, 1972].

۴ کاهش به مسأله برنامه‌ریزی خطی

این مسأله به صورت یک مسأله‌ی بهینه‌سازی نیز قابل بیان است. متغیرهای مسأله، f_e ها هستند و $۲m$ نا معادله به صورت $۰ \le f_e$ و $f_e \le C_e$ داریم. همچنین شرط بقای جریان نیز در هر رأس باید برقرار باشد، که در نهایت به $n - ۲$ معادله‌ی خطی می‌رسیم. هدف ماکسیمم کردن یک ترکیب خطی از f_e هاست. در حالت کلی به این مسأله برنامه‌ریزی خطی[۷] می‌گویند که به صورت زیر می‌توان آن را نوشت:

$$\boldsymbol{x} = (x_۱, x_۲, \ldots, x_n) \in \mathbb{R}^n$$

[۷] linear programming

$$\max \boldsymbol{c}^T\boldsymbol{x} \;\; \text{s.t.} \;\; \boldsymbol{x} \geq ۰, \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$$

می‌توان نشان داد که هر مسأله‌ی کمینه‌سازی یا بیشینه‌سازی یک تابع هدف خطی را با تعدادی محدودیت که می‌توانند شامل معادلات، نامعادلات، محدودیت روی متغیرها یا عدم محدودیت روی متغیرها باشد، به یک مسأله‌ی برنامه‌ریزی خطی به فرم کلی فوق قابل کاهش است. مثلا اگر به جای max بخواهیم تابع هدف را min کنیم کافی است تا همه ضرایب تابع هدف را منفی کنیم. همچنین اگر روی یکی از متغیرها (مثلا x_i) شرط $x_i > ۰$ نداشته باشیم. می‌توانیم به جای متغیر x_i متغیرهای کمکی $x'_i \geq ۰$ و $x''_i \geq ۰$ را تعریف کنیم به طوری که $x_i = x'_i - x''_i$. الگوریتم‌های متعددی برای حل این مسائل مطرح شده است. یکی از قدیمی‌ترین روش‌های حل این مسأله روش سادک[۸] است که هر چند با سرعت بالایی بسیاری از نمونه‌های عملی را حل می‌کند اما دارای پیچیدگی چندجمله‌ای نیست.
اولین الگوریتم چندجمله‌ای (اما غیر کارا) در بحبوحه دوران جنگ سرد در سال ۱۹۷۹ توسط لئونید خاچیان[۹]، ریاضیدان اهل شوروی سابق، ابداع شد. چند سال بعد یک دانشجوی دکترای دانشگاه برکلی به الگوریتم چندجمله‌ای متفاوتی (که در عمل نیز کاراست) دست یافت که به روش نقطه‌ی میانی [۱۰] معروف است.
با این وجود یافتن جواب بهینه صحیح یک نمونه مسأله برنامه‌ریزی خطی، یعنی،

$$\boldsymbol{x} = (x_۱, x_۲, \ldots, x_n) \in \mathbb{Z}^n$$
$$\max \boldsymbol{c}^T\boldsymbol{x} \;\; \text{s.t.} \;\; \boldsymbol{x} \geq ۰, \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$$

مسأله آسانی نیست. این مسأله که به مسأله‌ی برنامه‌ریزی خطی[۱۱] معروف است یک مسأله‌ی NP-hard است و انتظار نمی‌رود که راه حل در زمان چندجمله‌ای برای آن پیدا شود. یافتن چنین الگوریتمی مسأله باز "$P = NP?$" را حل می‌کند که در جلسات آینده درباره آن صحبت خواهیم کرد.
دقت کنید که مسأله‌ی جریان بیشینه با ظرفیت صحیح را می‌توان به یک مسأله‌ی برنامه‌ریزی خطی صحیح کاهش داد. با اینکه مسأله دوم در حالت کلی در زمان چندجمله‌ای قابل حل نیست اما راه حل‌های چندجمله‌ای موجود برای حل مسأله برنامه‌ریزی خطی، نمونه‌های کاهش یافته مسأله اول را در زمان چندجمله‌ای حل می‌کنند.

[۸] simplex
[۹] Leonid Khachiyan
[۱۰] interior point
[۱۱] integer linear programming