درس طراحی الگوریتم
دانشکده ریاضی. دانشگاه صنعتی خواجه نصیرالدین طوسی. پاییز ۱۴۰۱
کتاب مرجع: طراحی الگوریتم. جان کلاینبرگ و اوا تاردش

۱ زمان اجرای الگوریتم فورد فولکرسون

سلسله مشاهدات زیر را داریم.

- از طرز کار الگوریتم فورد فولکرسون نتیجه میگیریم که اگر ظرفیت یالها اعداد صحیح باشد آنگاه جریان بیشینه ای که الگوریتم بدست می آورد یک صحیح خواهد بود. مقدار جریان روی همه یالها هم اعداد صحیح خواهند بود. نتیجه میگیرم همواره یک جریان بیشینه $f: E \to \mathbb{Z}^+$ برای شبکه ای که ظرفیت یالها اعداد صحیح باشند، وجود دارد.

- همانطور که دیدیم در هر مرحله از الگوریتم به اندازه گلوگاه مسیر الحاقی که در گراف باقیمانده پیدا میشود به جریان اضافه میشود. چون مقدار گلوگاه b یک عدد صحیح مثبت و بیشتر از صفر است، پس در هر مرحله جریان به اندازه حداقل 1 واحد افزایش می یابد. لذا اگر $v(f_{\max})$ مقدار جریان بیشینه از s به t باشد، آنگاه تعداد مراحل الگوریتم حداکثر $v(f_{\max})$ خواهد بود.

- در هر مرحله چه کارهایی انجام میشود؟ با استفاده از f، جریان کنونی، گراف باقیمانده G_f ساخته میشود. اگر تعداد یالهای گراف اصلی G عدد m باشد، تعداد یالهای گراف باقیمانده حداکثر $2m$ خواهد بود (یک یال پیشرو و یک یال پسرو برای هر یال). گراف باقیمانده را میتوانیم در زمان $O(2m+n)$ بسازیم.

- در گراف باقیمانده G_f یک مسیر از s به t پیدا میشود. برای این کار، میتوانیم از الگوریتمهای BFS یا DFS استفاده کنیم. زمان اجرای این گام از الگوریتم $O(2m+n)$ خواهد بود.

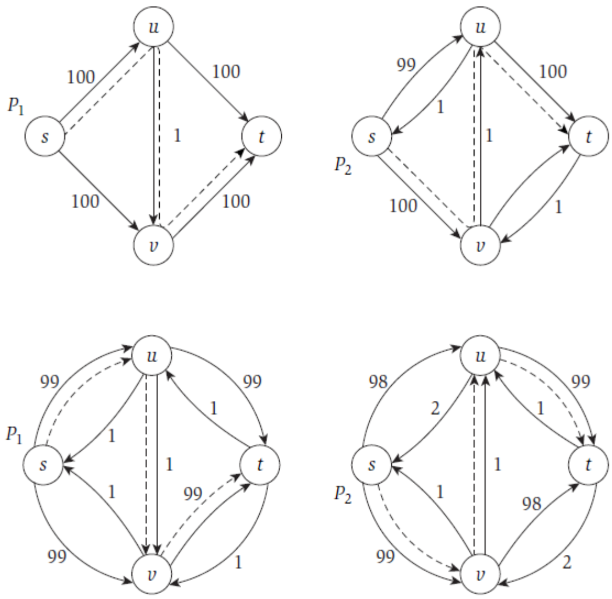
- سپس گلوگاه مسیر پیدا میشود و b محاسبه میشود. این گام هم در زمان $O(n)$ قابل انجام است. چون طول مسیر حداکثر $n-1$ یال است.

- بعد از محاسبه گلوگاه مسیر، جریان کنونی f بروزرسانی میشود. این هم در زمان $O(m)$ قابل انجام است.

- نتیجه میگیرم در مجموع هر مرحله در زمان $O(m+n)$ قابل پیاده سازی است.

- لذا در کل زمان اجرای الگوریتم $O(v(f_{\max})(m+n))$ خواهد بود.

- اگر چه تعداد مراحل میتواند خیلی کمتر از $v(f_{max})$ باشد، ممکن است تعداد مراحل الگوریتم واقعا $v(f_{max})$ بشود. به مثال زیر توجه کنید.



در این مثال مقدار جریان بیشینه از s به t برابر با 200 است. الگوریتم میتواند با انتخاب مسیرهای مناسب کار را در دو مرحله تمام کند. یک بار 100 واحد جریان از مسیر بالا بفرستد و بار دیگر 100 واحد از مسیر پایین به t بفرستد.

اما الگوریتم اگر مسیر الحاقی که با خط چین مشخص شده را انتخاب کند، هر بار تنها یک واحد به مقدار جریان افزوده میشود. به گراف باقیمانده توجه کنید. الگوریتم با انتخابهای بد، هر بار یکی از دو مسیر $P1$ و $P2$ را انتخاب میکند و در نتیجه هر مرحله فقط یک واحد به جریان اضافه میشود. در نهایت تعداد مراحل 200 خواهد شد!

- زمان اجرای $O(v(f_{max})(m+n)$ یک زمان اجرای چند جمله ای نیست. مانند مسئله کوله پشتی، اینجا هم زمان اجرا نسبت چند جمله ای با مقدار یکی از پارامترهای ورودی دارد. در مسئله کوله پشتی هم زمان اجرای الگوریتم پیشنهادی نسبت چند جمله ای با مقدار W که یکی از پارامترهای ورودی داشت. اینجا هم زمان اجرا با مقدار جریان $v(f_{max})$ نسبت چند جمله ای دارد. اگر $v(f_{max})$ یک عدد خیلی بزرگ، مثلا 2^n باشد (وقتی n تعداد رئوس گراف است.) زمان اجرای الگوریتم نمایی خواهد شد.

- با انجام تغییراتی در الگوریتم فورد فولکرسون میتوان زمان اجرا را چندجمله ای کرد. در این ایده، انتخاب مسیر الحاقی از s به t در گراف باقیمانده هوشمندانه تر خواهد شد. در واقع مسیری را انتخاب میکنیم که بیشترین گلوگاه را داشته باشد (این مسئله را چطور حل میکنید؟).

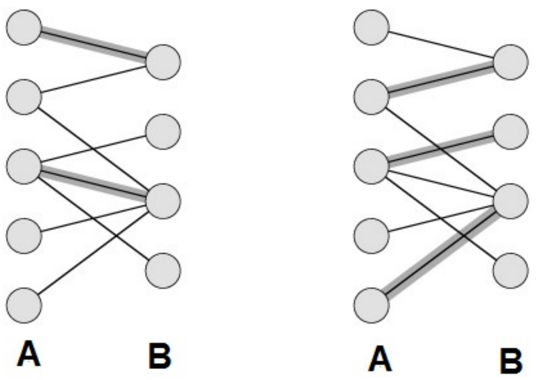
  با اعمال این تغییر، اثبات میشود که الگوریتم پس از طی حداکثر $O(m \log v(f_{max}))$ مرحله خاتمه خواهد یافت. اینجا زمان اجرای الگوریتم با پارامتر m تعداد یالها و لگاریتم $v(f_{max})$ نسبت چند جمله ای دارد و لذا زمان اجرا چند جمله ای خواهد بود.

۲ نتایج و کاربردهای مسئله جریان بیشینه

مسئله جریان بیشینه و قضیه تساوی برش کمینه و جریان بیشینه کاربردهای وسیع و فراوانی در بهینه سازی ترکیبیاتی و تئوری گرافها دارد. در این قسمت به ذکر چند کاربرد و نتیجه میپردازیم.

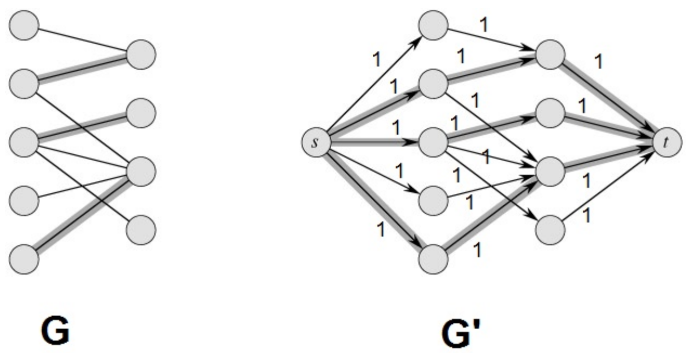
۱.۲ تطابق بیشینه در گرافهای دو بخشی

رئوس گراف دوبخشی G را میتوان به دو بخش A و B افراز کرد بطوریکه همه یالهای گراف یک سرشان در A و سر دیگر در B داشته باشند. فرض کنید گراف دوبخشی $G = (A \cup B, E)$ داده شده است. اینجا دو بخش A و B مشخص شده اند. میخواهیم، یک زیرمجموعه از یالها $M \subseteq E$ را انتخاب کنیم بطوریکه هیچ دو یالی در M راس مشترک نداشته باشند. به این زیر مجموعه از یالها یک تطابق matching گفته میشود.



در شکل بالا سمت چپ یک گراف دوبخشی دیده میشود. یک تطابق، شامل دو یال را میبینید. در سمت راست برای همان گراف یک تطابق شامل سه یال نشان داده شده است.

در مسئله تطابق بیشینه، دنبال یک تطابق با بیشترین تعداد یال میگردیم. برای حل این مسئله در گرافهای دوبخشی میتوانیم از مسئله جریان بیشینه استفاده کنیم. در واقع با داشتن گراف دوبخشی $G = (A \cup B, E)$ یک شبکه G' با رئوس مبدا s و t میسازیم. بطوریکه از حل مسئله جریان بیشینه در G' یک راه حل برای مسئله تطابق بیشینه برای گراف G بدست بیاید.



همانطور که در شکل بالا نشان داده شده است، ساخت شبکه G' کار آسانی است. اولا جهت همه یالها از A به سمت B خواهد بود. دو راس s و t اضافه شده اند. از راس مبدا s به همه رئوس A یک یال قرار میدهیم. همینطور از رئوس B به راس مقصد t یک یال قرار میدهیم. ظرفیت همه یالها 1 قرار میدهیم.

از آنجا که ظرفیتها عدد صحیح هستند، جریان بیشینه نیز در شبکه G' اعداد صحیح خواهد بود. جریان روی هر یال عدد صفر یا 1 خواهد بود. یالهای بین A و B با جریان 1 همان یالهای تطابق بیشینه هستند.

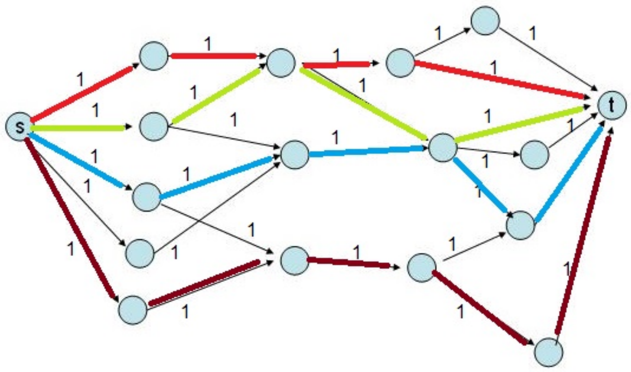
۲.۲ مسیرهای یالی مجزا بین s و t

گراف جهت دار $G = (V, E)$ داده شده است. دو راس s و t در G مشخص شده است.

دنبال مجموعه ای از مسیرهای ساده $P_1, \ldots, P_t$ از s به t میگردیم بطوریکه هیچ دو P_i و P_j یال مشترکی نداشته باشند. در واقع به دنبال بیشترین تعداد مسیر ساده از s به t میگردیم که اشتراک یالی نداشته باشند.

قبل از اینکه این مسئله را تبدیل به یک مسئله جریان بیشینه کنیم، یک مشاهده ساده انجام میدهیم. اگر s یال ورودی داشته باشد میتوانیم آن را حذف کنیم. همینطور اگر t یال خروجی داشته باشد میتوانیم آن را حذف کنیم. این یالها نمیتوانند سهمی در مسیر s به t داشته باشند. لذا حذف کردن آنها مشکلی ایجاد نمیکند.

حال با این تغییر کوچک، آماده هستیم که مسئله را تبدیل به یک مسئله جریان بیشینه کنیم. این کار مشابه مسئله تطابق بیشینه در گراف دوبخشی خیلی سرراست و آسان است. کافی است که به همه یالها ظرفیت 1 بدهیم. همانند مشاهداتی که برای مسئله تطابق بیشینه داشتیم، چون ظرفیتها عدد صحیح هستند، در واقع همه عدد 1 هستند، جریان بیشینه از s به t بصورت مجموعه ای از مسیرهای مجزا (بدون یال مشترک) از s به t در خواهد آمد.



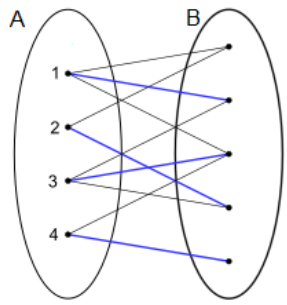
در گراف بالا چهار مسیر مجزا از s به t بدست آمده است. به عبارت دیگر، از s به t میتوان به اندازه 4 واحد جریان فرستاد. هر واحد از جریان از طریق یکی از مسیرها به انتقال پیدا میکند.

۳.۲ قضیه ازدواج هال

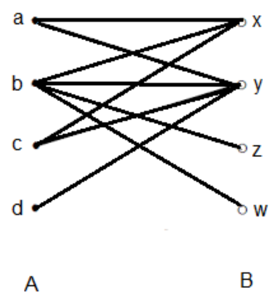
به مسئله تطابق بیشینه در گرافهای دوبخشی برمیگردیم. فرض کنید گراف دوبخشی $G = (A \cup B, E)$ داده شده است. پرسش ما باز هم در مورد تطابق در این گراف است. ممکن است کسی سوال کند، آیا تطابقی در گراف دوبخشی G وجود دارد که همه رئوس سمت A را پوشش دهد؟ به عبارت دیگر، تطابقی وجود دارد که در آن هر راس A با یکی از رئوس B یک زوج را تشکیل دهند. من اینجا مخصوصا از کلمه زوج استفاده کردم چون گاها برای معرفی این مسئله را از عبارت مسئله ازدواج استفاده میشود. یک بیان دیگر این مسئله میتواند بدین بصورت باشد. تعدادی شخص (مجموعه A) میخواهند برای خودشان همسر انتخاب کنند. هر شخص در مجموعه A برای خودش افرادی را در مجموعه B مشخص کرده و مایل است که با یکی از آنها ازدواج کند. آیا میتوانیم برای هر شخص در مجموعه A همسری پیدا کنیم، بطوریکه هر کس با یکی از افراد مورد علاقه اش ازدواج کند؟

یک بیان دیگر از همین مسئله با استفاده از نظریه مجموعه هاست. یک مجموعه متناهی X داریم. برای سادگی فرض کنید $X = \{1, 2, \ldots, m\}$. علاوه بر این یک سری مجموعه $S_1, \ldots, S_n$ داریم بطوریکه هر $S_i \subseteq X$. پرسشی که مطرح است این است که آیا میتوان از هر مجموعه S_i یک عضو را انتخاب کرد بطوریکه در مجموع هیچ عضوی از X دو بار یا بیشتر انتخاب نشده باشد؟

اینها همه بیانهای مختلف از یک مسئله واحد هستند.



خوب اجازه بدهید از همان بیان گراف دوبخشی و تطابق استفاده کنیم. یک شرط لازم برای اینکه تطابقی وجود داشته باشد که همه رئوس A را پوشش دهد (یک تطابق که به اندازه $|A|$) این است که هر زیرمجموعه S از A که شما در نظر بگیرید تعداد همسایه های S در طرف B باید حداقل به اندازه $|S|$ باشد. فرض کنید $N(S)$ مجموعه همسایه های رئوس S باشد. به عبارت دیگر باید داشته باشیم $|N(S)| \geq |S|$.



شکل بالا نمونه ای را نشان میدهد که تطابقی به اندازه $|A| = 4$ وجود ندارد. چون مجموعه $S = \{a, c, d\}$ همسایه هایش $\{x, y\}$ هستند. لذا اینجا $N(\{a, c, d\}) = 2 < 3$.

شرطی که از آن صحبت کردیم، باید برای هر زیرمجموعه S از برقرار باشد. به عبارت دیگر، باید داشته باشیم

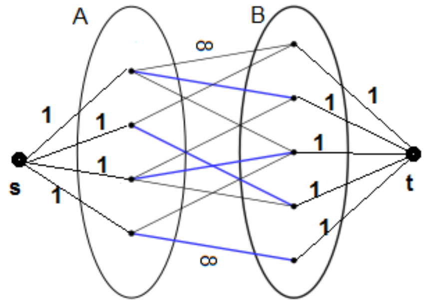
$$\forall S \subseteq A, \quad |N(S)| \geq |S|$$

طبیعتا این شرط برای وجود تطابقی که همه A را پوشش دهد لازم است. اما آیا کافی نیز هست؟ جواب این سوال مثبت است. قضیه هال که توسط ریاضیدانی به نام فیلیپ هال (و البته خیلی افراد دیگر) اثبات شده است همین را میگوید.

قضیه هال: در گراف دوبخشی $G = (A \cup B, E)$ اگر برای هر زیرمجموعه $S \subseteq A$ داشته باشیم $|N(S)| \geq |S|$ آنگاه یک تطابق به اندازه $|A|$ در گراف G وجود دارد.

اثبات: برای اثبات، گراف دوبخشی G را مانند موارد قبل تبدیل به یک شبکه جریان $s - t$ میکنیم و نشان میدهیم اگر شرایط قضیه هال برقرار باشد، آنگاه یک جریان بیشینه با اندازه $|A|$ از s به t وجود دارد که معادل با وجود یک تطابق با $|A|$ یال است. تنها یک تفاوت در تبدیل ما اینجا وجود دارد. ظرفیت یالهایی که A را به B وصل میکند به جای 1 بینهایت ∞ قرار میدهیم. این تغییر هیچ تاثیری در افزایش یا کاهش جریان از s به t ندارد ولی همانطور که خواهیم دید خیلی به ساده شدن اثبات کمک میکند.

شکل زیر یک نمونه از این تبدیل را نشان می دهد. دقت کنید که ظرفیت همه یالهای بین A و B حالا بینهایت است.



چند مشاهده که در اثبات به ما کمک میکند.

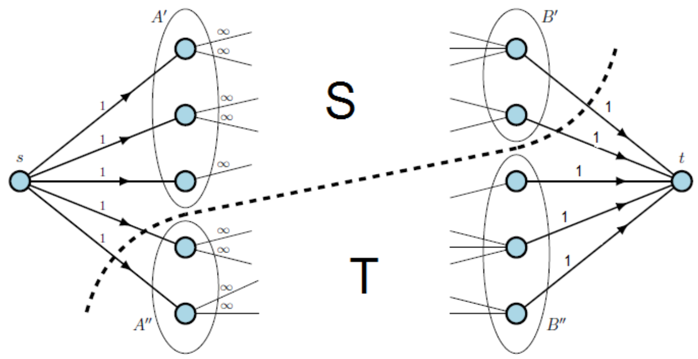
- فرض کنید که f_{max} جریان بیشینه از s به t باشد. کافی است نشان دهیم اگر شرط قضیه هال برقرار باشد، داریم
$$v(f_{max}) = |A|$$
این معادل این است که بگوییم تطابقی به اندازه $|A|$ در گراف وجود دارد.

- واضح است که $v(f_{max}) \leq |A|$ چون مقدار جریان از s به t نمی تواند از مجموع ظرفیت یالهای خروجی از s بیشتر باشد.

- بنا به یک قضیه اساسی، ظرفیت برش کمینه $s-t$ برابر با مقدار جریان بیشینه از s به t است. فرض کنید (S,T) برش کمینه در گراف ما باشد و $c(S,T)$ ظرفیت این برش باشد. داریم
$$c(S,T) = v(f_{max}) \leq |A|$$

- بنا به مشاهده قبلی، ظرفیت برش کمینه عددی متناهی است. در نتیجه برش کمینه (S,T) هیچ یالی بین A و B که ظرفیت بینهایت دارند را قطع نمیکند.

- فرض کنید $S = \{s\} \cup A' \cup B'$ و $T = \{t\} \cup A'' \cup B''$. دقت کنید اینجا $A' \subseteq A, B' \subseteq B$ و به همین ترتیب $A'' \subseteq A, B'' \subseteq B$. وضعیت در شکل زیر نشان داده شده است.



چون برش (S,T) هیچ یالی بین A و B را قطع نمیکند، ظرفیت برش برابر است با

$$c(S,T) = |A''| + |B'|$$

از طرف دیگر داریم

$$N(A') \subseteq B'$$

پس

$$c(S,T) = |A''| + |B'| \geq |A''| + |N(A')|$$

چون شرط قضیه هال برقرار است باید داشته باشیم $|N(A')| \geq |A'|$. لذا

$$c(S,T) = |A''| + |B'| \geq |A''| + |N(A')| \geq |A''| + |A'| = |A|$$

پس ظرفیت برش کمینه از $|A|$ کمتر نیست. چون $v(f_{max}) = c(S,T)$ لذا $v(f_{max}) \geq |A|$ و این یعنی $v(f_{max}) = |A|$ چون قبلا مشاهده کردیم که $v(f_{max}) \leq |A|$.