طراحی الگوریتم
دانشکده ریاضی. دانشگاه صنعتی خواجه نصیرالدین طوسی. پاییز ۱۴۰۳
کتاب مرجع: طراحی الگوریتم. جان کلاینبرگ و اوا تاردش

۱ زمان اجرای الگوریتم فورد فولکرسون

سلسله مشاهدات زیر را داریم.

۱. از منطق الگوریتم فورد فولکرسون نتیجه می‌گیریم 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 می‌گیریم همواره یک جریان بیشینه صحیح $f : E \to \mathbb{Z}^+$ برای شبکه ای که ظرفیت یالها اعداد صحیح باشند، وجود دارد.

۲. همانطور که دیدیم در هر مرحله از الگوریتم به اندازه گلوگاه مسیر افزایشی که در گراف باقیمانده پیدا می‌شود به مقدار 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 $value(f_{\max})$ مقدار جریان بیشینه از s به t باشد، آنگاه تعداد مراحل الگوریتم حداکثر $value(f_{\max})$ خواهد بود.

۳. در هر مرحله چه کارهایی انجام می‌شود؟

(آ) ابتدا از روی جریان کنونی $f : E \to \mathbb{Z}$ 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.

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

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

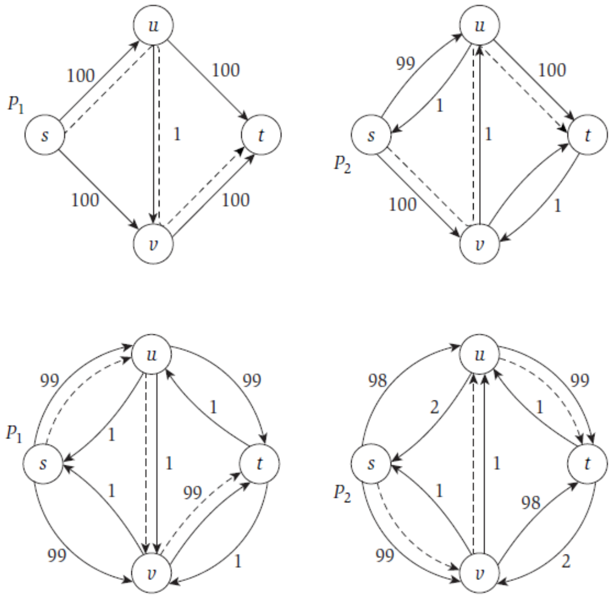
(د) بعد از محاسبه گلوگاه مسیر، جریان کنونی f طبق روالی که گفته شد، بروزرسانی می‌شود. این هم در زمان $O(m)$ قابل انجام است.

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

۵. از بحثهای بالا نتیجه می‌شود که زمان اجرای الگوریتم حداکثر $O(value(f_{max})(m+n))$ است (دقت کنید این برای حالتی صادق است که ظرفیت یالها عدد صحیح است.) اگر فرض بگیریم C ظرفیت خروجی راس s باشد (به عبارت دیگر C مجموع ظرفیت یالهای خروجی از s است)، چون $value(f_{max}) \leq C$ یک کران بالای ساده برای زمان اجرا $O(mnC)$ خواهد بود [۱]

[۱] متاسفانه زمان اجرای $O(C(m+n))$ یک زمان اجرای چند جمله ای نیست. مانند مسئله کوله پشتی، زمان اجرا نسبت چند جمله ای با اندازه ورودی ندارد. دقت کنید اندازه ورودی با تعداد یالها m و تعداد رئوس n مشخص می‌شود در حالیکه زمان اجرای الگوریتم نسبت مستقیم با پارامتر C دارد که می‌تواند عدد خیلی بزرگی باشد. مثلا اگر C در حد 2^n باشد (وقتی n تعداد رئوس گراف است.) زمان اجرای الگوریتم نمایی خواهد شد.

۶. اگر چه تعداد مراحل می‌تواند خیلی کمتر از C باشد، ممکن است تعداد مراحل الگوریتم واقعا C 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 افزایشی 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!

۷. مثالهایی وجود دارد که اگر ظرفیت یالها اعداد گنگ باشند، الگوریتم فورد فولکرسون می‌تواند بینهایت مرحله داشته باشد و حتی به جواب بهینه همگرا نشود.

۲ تسریع الگوریتم فورد - فولکرسون

با انجام تغییراتی در الگوریتم فورد فولکرسون می‌توانیم وابستگی زمان اجرا به پارامتر C را کاهش دهیم. یک ایده کلی این است که مسیر افزایشی از s به t در گراف باقیمانده را هوشمندانه تر انتخاب کنیم. مثلا مسیری را انتخاب کنیم که بیشترین گلوگاه را داشته باشد (در نتیجه در یک مرحله بیشترین مقدار را به جریان اضافه شود.) یا مثلا مسیری را انتخاب کنیم که کمترین تعداد یال را داشته باشد. خوشبختانه هر دوی این ایده‌ها باعث تسریع الگوریتم فورد فولکرسون خواهند شد. در ادامه ایده اول را بررسی می‌کنیم.

۱.۲ انتخاب مسیر افزایشی : مسیر با بیشترین گلوگاه

فرض کنید G_f گراف باقیمانده نسبت به جریان f باشد. حال فرض کنید همه یالهایی که ظرفیتی کمتر از مقدار آستانه Δ دارند را از گراف G_f حذف کنیم. گراف حاصل را G_f^Δ می‌نامیم. بدیهی است که هر مسیر از s به t در گراف G_f^Δ گلوگاهی به اندازه حداقل Δ خواهد داشت.

بر اساس ایده گلوگاه، استراتژی زیر برای انتخاب مسیر افزایشی پیشنهاد شده است:

۱. $C \leftarrow$ مجموع ظرفیت یالهای خروجی از s

۲. $\Delta \leftarrow 2^{\lfloor \log C \rfloor}$. به عبارت دیگر، مقدار اولیه Δ بزرگترین توان ۲ کمتر یا مساوی با C است.

۳. $f \leftarrow 0$

۴. تا زمانیکه $\Delta \geq 1$ گامهای زیر را تکرار کن:

(آ) گراف باقیمانده G_f^Δ را بساز.

(ب) تا زمانیکه یک مسیر از s به t در گراف G_f^Δ وجود دارد، یک مسیر افزایشی در این گراف پیدا کن و همانند الگوریتم فورد فولکرسون جریان کنونی f را بروزرسانی کن. این کار را تکرار کن تا زمانیکه مسیری از s به t در گراف G_f^Δ پیدا نشود.

(ج) $\Delta \leftarrow \frac{\Delta}{2}$

۵. جریان f را به عنوان جواب گزارش کن.

درستی الگوریتم به هر اجرای گام ۴.(ب) یک فاز از الگوریتم می‌گوییم. در اولین فاز، $\Delta = 2^{\lfloor \log C \rfloor}$ و سپس در فازهای بعدی مقدار Δ تقسیم بر ۲ می‌شود تا اینکه در آخرین فاز مقدار Δ به ۱ می‌رسد. در انتهای این فاز، دیگر مسیری از s به t پیدا نمی‌شود. روشن است چون $\Delta = 1$ وضعیت مانند پایان الگوریتم فورد فولکرسون است و لذا می‌توانیم ادعا کنیم که جواب نهایی الگوریتم بالا یک جریان بیشینه است.

تعداد فازهای الگوریتم چون Δ از $2^{\lfloor \log \rfloor}$ شروع می‌شود و هر بار تقسیم بر ۲ می‌شود پس تعداد فازهای الگوریتم $\lfloor \log C \rfloor$ است.

تعداد تکرارها در یک فاز ادعا می‌کنیم که در یک فاز خاص، تعداد دفعاتی که جریان افزایش می‌یابد حداکثر $2m$ است. برای این منظور نیاز به آوردن مقداری استدلال داریم.

لم ۱: فرض کنید f تابع جریان در پایان یک فاز Δ باشد. آنگاه داریم:

$$value(f_{max}) - value(f) \leq m\Delta$$

اثبات: گراف باقیمانده G_f^Δ را در نظر بگیرید. در پایان این فاز مسیری از s به t پیدا نشده. تعریف می‌کنیم A مجموعه همه رئوسی باشد که از s به آنها مسیری در G_f^Δ وجود داشته باشد. همچنین B را بقیه رئوس گراف قرار می‌دهیم. روشن است که $t \in B$ و $s \in A$. حال به برش (A, B) را در گراف اصلی توجه کنید. برای هر یال $e = (u, v)$ که $u \in A$ و $v \in B$ باشد، باید داشته باشیم $f(e) > c(e) - \Delta$. چون اگر $f(e) \leq c(c) - \Delta$ آنگاه باید یک یال پیشرو از u به سمت v در گراف باقیمانده G_f^Δ وجود داشته باشد که این با $v \in B$ در تناقض است. از طرف دیگر، برای هر یال $e = (v, u)$ که $v \in B$ و $u \in A$ باید داشته

باشیم $f(e) < \Delta$. چون اگر $f(e) \geq \Delta$ باید یال پسرو از سمت u به سمت v وجود داشته باشد و این باز هم با $v \in B$ در تناقض خواهد بود.
داریم

$$
\begin{aligned}
value(f) &= f_{out}(A) - f_{in}(A) && (۱)\\
&= \sum_{e \text{ of out } A} f(e) - \sum_{e \text{ into } A} f(e) && (۲)\\
&\geq \sum_{e \text{ of out } A} (c(e) - \Delta) - \sum_{e \text{ into}} \Delta && (۳)\\
&\geq \sum_{e \text{ of out } A} c(e) - \sum_{e \text{ of out } A} \Delta - \sum_{e \text{ into}} \Delta && (۴)\\
&\geq cap(A, B) - m\Delta && (۵)\\
&\geq value(f_{max}) - m\Delta && (۶)
\end{aligned}
$$

گزاره لم از (۶) نتیجه می‌شود. □

لم ۲: تعداد افزایش جریان در هر فاز الگوریتم حداکثر $2m$ است.

اثبات : فرض کنید در یک فاز Δ هستیم. هر بار که یک مسیر افزایشی پیدا می‌کنیم جریان به اندازه حداقل Δ افزایش می‌یابد. از این مشاهده و لم قبلی استفاده می‌کنیم.

فرض کنید در اولین فاز الگوریتم هستیم. اینجا $\frac{C}{2} \leq \Delta \leq C$. اگر در این فاز تعداد افزایش ها از $2m$ بیشتر باشد. یعنی ما جریان را به اندازه حداقل $(2m+1)\Delta$ افزایش داده‌ایم. این امکانپذیر نیست چون $\Delta \geq \frac{C}{2}$ و لذا مقدار افزایش جریان در این فاز حداقل $(2m+1)\frac{C}{2} > C$ خواهد بود و این با $value(f_{max}) \leq C$ متناقض است.

حال فرض کنید در انتهای یک فاز Δ هستیم و f جریان کنونی است. بنا به لم قبلی داریم:

$$value(f_{max}) - value(f) \leq m\Delta \qquad (۷)$$

در فاز بعدی مقدار آستانه $\Delta/2$ خواهد بود. ادعا می‌کنیم در فاز بعدی تعداد افزایش جریان حداکثر $2m$ است. اگر f' جریان در انتهای انتهای فاز $\Delta/2$ باشد داریم (باز هم بنا به لم قبلی)

$$value(f_{max}) - value(f') \leq m\Delta/2. \qquad (۸)$$

اگر تعداد افزایش جریان در این فاز بیشتر از $2m$ باشد پس

$$value(f') - value(f) \geq (2m+1)\Delta/2 = m\Delta/2 + \Delta/2. \qquad (۹)$$

با کسر کردن (۸) از (۷) بدست می‌آید:

$$value(f') - value(f) \leq m\Delta/2. \qquad (۱۰)$$

این با (۹) در تناقض است. پس گزاره لم درست است. $\square$

از آنجا که هر افزایش جریان در زمان $O(m+n)$ قابل انجام است، زمان اجرای الگوریتم برابر است با :

$$O(m+n) \times \text{حداکثر دفعات افزایش جریان در یک فاز} \times \text{تعداد فازها}$$

که این حداکثر $O(\log C \times 2m \times (m+n))$ است.