دانشکده علوم ریاضی
دانشگاه صنعتی شریف

بهینه‌سازی ترکیبیاتی

محمدهادی فروغمنداعرابی
بهار ۱۳۹۶

الگوریتم پیدا کردن تطابق بیشینه در گراف غیردوبخشی

جلسه بیست‌ودوم

نگارندگان: سینا اکبری، محمد طه طوغانی

۱ مرور

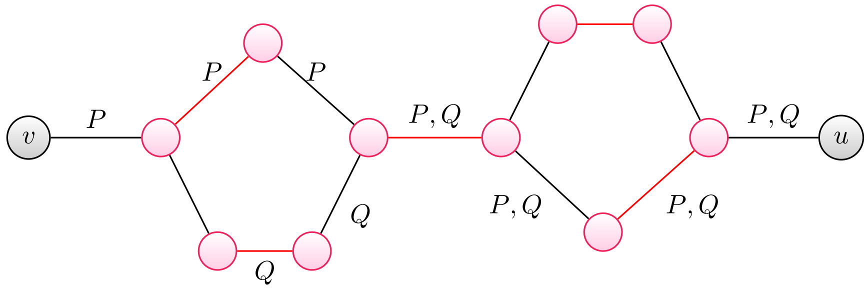
قبلا دیدیم که برای پیدا کردن تطابق بیشینه در گراف دوبخشی می‌توانیم به صورت زیر عمل کنیم:

Algorithm 1 BIPARTITE MATCHING

```
M ← ∅
while there is an M − augmenting path do
    FIND AN AUGMENTING PATH P
    M ← MΔP
```

می‌خواهیم از الگوریتمی مشابه برای پیدا کردن تطابق بیشینه در گراف غیردوبخشی استفاده کنیم. مشکلی که الگوریتم در گراف‌های غیردوبخشی به آن برمی‌خورد، وجود دورهای فرد است. به مثال زیر توجه کنید:

مثال ۱. *در گراف شکل ۱، مسیر افزایشی P از راس u به راس v وجود دارد اما اگر طبق الگوریتم قبلی در اجرای DFS از مسیر Q برویم، مسیر افزایشی پیدا نخواهد شد. بنابراین الگوریتم بالا برای گراف‌های غیردوبخشی لزوما کار نمی‌کند. (یال‌های تطابق به رنگ قرمز و یال‌های غیرتطابق به رنگ سیاه نمایش داده شده‌اند)*



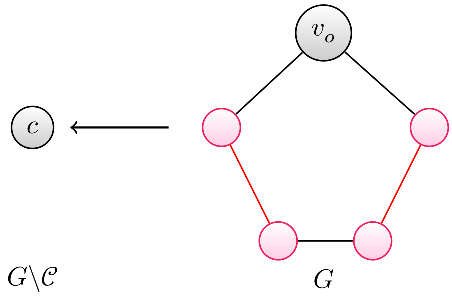
شکل ۱: مثالی از یک گراف با دورهای فرد که الگوریتم تطابق برای آن کار نمی‌کند.

۲ ایده‌ی اصلی الگوریتم

می‌خواهیم به نحوی دورهای فرد گراف را حذف کنیم تا بتوانیم از الگوریتم قبلی استفاده کنیم. اما باید دقت کنیم که با تغییر گراف، وجود یا عدم وجود مسیر افزایشی نباید تغییر کند.

تعریف ۲. *به دور فردی که دقیقا یک راس نپوشیده داشته‌باشد، غنچه می‌گوئیم. در واقع غنچه دور فردی است که یال‌های آن یکی‌درمیان از یال‌های تطابق و غیرتطابق هستند و راس v_o از آن نپوشیده‌است.*

تعریف ۳. *در طول الگوریتم، در صورت لزوم یک غنچه را از گراف حذف کرده و یک راس جایگزین آن خواهیم کرد. به این کار، منقبض کردن غنچه می‌گوئیم. (به ازای یال‌های متصل به رئوس غنچه، به راس جایگزین یال متصل می‌کنیم.)*



شکل ۲: منقبض کردن غنچه‌ی $\mathcal{C}$

توجه. *ویژگی مهم غنچه این است که هیچ یال تطابقی از بیرون به آن وصل نیست. چون همه‌ی رئوس آن با یال‌های تطابق درون خود غنچه پوشیده شده‌اند به جز راس v_o که آن هم نپوشیده است و در نتیجه به یال تطابقی وصل نیست. بنابراین، پس از منقبض کردن یک غنچه، تمام یال‌های متصل به راس جایگزین، یال‌های غیرتطابق هستند.*

قضیه ۴. *فرض کنید G یک گراف، M یک تطابق در G و $\mathcal{C}$ یک غنچه در G باشد. G مسیر M−افزایشی دارد اگر و تنها اگر $G\backslash\mathcal{C}$ مسیر M−افزایشی داشته‌باشد.*

اثبات.($\Leftarrow$) فرض کنید P یک مسیر M−افزایشی در G باشد. اگر P از $\mathcal{C}$ نگذرد، به وضوح مسیر P در $G\backslash\mathcal{C}$ هم وجود دارد چون با منقبض کردن غنچه‌ی $\mathcal{C}$، بقیه‌ی رئوس G هیچ تغییری نمی‌کنند. پس فرض کنید مسیر P از راس نپوشیده‌ی u شروع می‌شود و فرض کنید راس v اولین راسی از $\mathcal{C}$ باشد که P از آن می‌گذرد. قسمتی از مسیر P که از u شروع و به v ختم می‌شود را P' بنامید. حالا در گراف $G\backslash\mathcal{C}$، مسیر P' وجود دارد که از راس نپوشیده‌ی u شروع می‌شود و به راس c ختم می‌شود. (در مسیر P' به جای راس v، راس جایگزین غنچه را قرار می‌دهیم.) در توجه قبلی گفتیم که تمام یال‌های متصل به راس c غیرتطابق هستند. بنابراین راس c نپوشیده است و در نتیجه P' یک مسیر M−افزایشی در $G\backslash\mathcal{C}$ است.

($\Rightarrow$) فرض کنید P یک مسیر M−افزایشی در $G\backslash\mathcal{C}$ باشد. اگر P از راس c نگذرد، به وضوح مسیر P در گراف G یک مسیر M−افزایشی است. پس فرض کنید P از راس c می‌گذرد و v_o راس نپوشیده‌ی غنچه‌ی $\mathcal{C}$ است. مسیر P را در گراف G در نظر می‌گیریم. دو حالت ممکن است:

۱. مسیر P از v_o وارد $\mathcal{C}$ شود. در این صورت مسیر P در G، M—افزایشی است. (باز هم توجه کنید که v_o نپوشیده‌است.)

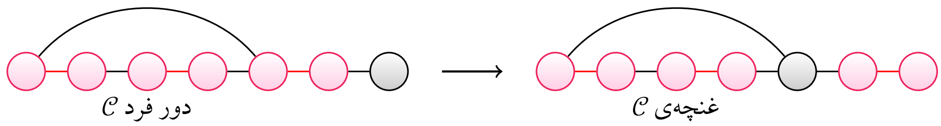
۲. مسیر P از راسی مانند u که $u \neq v_o$ وارد $\mathcal{C}$ شود. در این صورت با یک یال غیرتطابق وارد راس پوشیده‌ی u شده‌ایم. (با توجه به تعریف غنچه) حالا ادامه‌ی مسیر را روی غنچه‌ی $\mathcal{C}$ حرکت می‌کنیم (یکی در میان با یال‌های تطابق و غیرتطابق) تا به راس v_o برسیم. مسیر حاصل یک مسیر M—افزایشی در G است.

□

با توجه به قضیه‌ی بالا، اگر در اجرای الگوریتم به یک غنچه برخوردیم، می‌توانیم آن را منقبض کنیم و الگوریتم را ادامه دهیم.

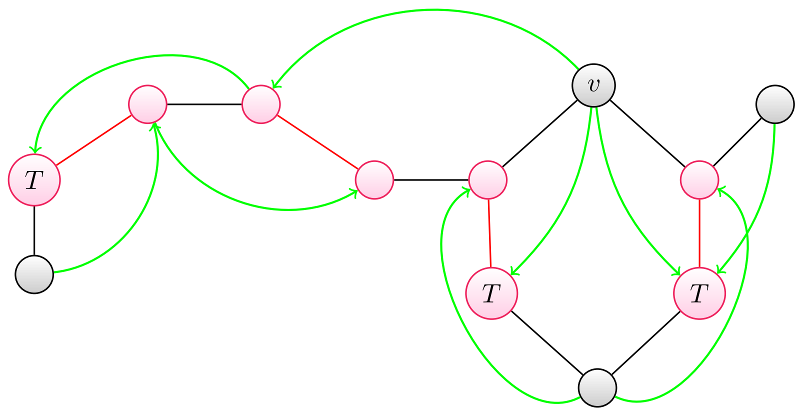
۳ پیدا کردن غنچه

فرض کنید تطابق M در گراف غیردوبخشی G را داریم و الگوریتم BFS را با یال‌های یکی در میان تطابق و غیرتطابق اجرا می‌کنیم و به راس $mark$ شده‌ای می‌رسیم که دور فرد $\mathcal{C}$ را می‌سازد. فرض کنید از راس v_o وارد این دور شده‌ایم. چون یال‌ها یکی‌درمیان تطابق و غیرتطابق هستند و به یک راس حداکثر یک یال از تطابق می‌تواند متصل باشد، هر دو یال متصل به v_o که درون دور هستند، غیرتطابق هستند. بنابراین با یال تطابق وارد دور شده‌ایم. چون راس v_o پوشیده‌شده است، دور $\mathcal{C}$ در شرایط غنچه بودن صدق نمی‌کند. بنابراین، همه‌ی یال‌های مسیر تا رسیدن به این دور را $switch$ می‌کنیم. (روی این مسیر، یال‌های غیرتطابق را به تطابق اضافه و یال‌های تطابق را از تطابق حذف می‌کنیم.) توجه کنید که با این کار، تطابق M' به دست می‌آید که سایز آن با M برابر است و حالا در تطابق جدید، $\mathcal{C}$ یک غنچه است. بنابراین، می‌توانیم $\mathcal{C}$ را منقبض کنیم و به دنبال یک مسیر $M'\backslash\mathcal{C}$—افزایشی در $G\backslash\mathcal{C}$ بگردیم.



شکل ۳: تبدیل دور فرد به غنچه

با توجه به آن‌چه در بالا گفتیم، باید روشی پیدا کنیم تا بتوانیم BFS را با یال‌های یکی درمیان تطابق و غیرتطابق اجرا کنیم. برای این کار، گراف جهت‌دار D را از روی گراف G به این طریق می‌سازیم که به ازای یک یال غیرتطابق و یک یال تطابق متوالی، یک یال جهت‌دار از ابتدای یال غیرتطابق به انتهای یال تطابق وصل می‌کنیم. برای مثال در شکل زیر، یال‌های گراف G با دو رنگ سیاه (یال غیرتطابق) و قرمز (یال تطابق) و یال‌های گراف جهت‌دار متناظر با رنگ سبز نمایش داده شده‌اند:



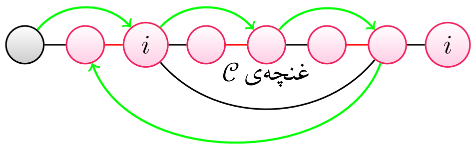
شکل ۴: ساختن گراف جهت‌دار برای اجرای BFS

حالا فرض کنید در گراف شکل ۴، از راس v شروع می‌کنیم. هدف پیدا کردن مسیری جهت‌دار در گراف D از v به یکی از راس‌هایی است که در شکل با T مشخص شده‌اند. (این رئوس در واقع راس‌هایی هستند که همسایه‌ی نپوشیده دارند.)

ادعا ۵. اگر در گراف D از راس نپوشیده‌ی v کوتاه‌ترین مسیر جهت‌دار به یکی از رئوس مجموعه‌ی T (رئوسی که حداقل یک همسایه‌ی نپوشیده دارند) پیدا کنیم، یا یک مسیر Mـافزایشی در G و یا یک غنچه در G یافته‌ایم.

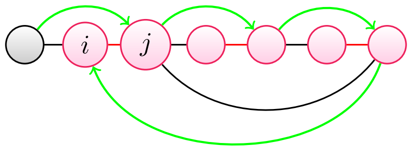
اثبات. فرض کنید P یک کوتاه‌ترین مسیر جهت‌دار از راس نپوشیده‌ی v به T در D باشد. اگر هر یال جهت‌دار را با همان دو یال متوالی در تعریف D جایگزین کنیم، مسیر P، معادل گشتی مانند P' در گراف G خواهد بود. اگر P' راس تکراری نداشته‌باشد (یک مسیر باشد)، می‌توانیم راس نپوشیده‌ی همسایه را به این مسیر اضافه کنیم و در نتیجه یک مسیر افزایشی در G یافته‌ایم. (توجه کنید که اگر این راس نپوشیده قبلا در P' موجود باشد به سادگی می‌توانیم قسمتی از مسیر تا رسیدن به این راس را به عنوان مسیر افزایشی در نظر بگیریم.) پس فرض می‌کنیم گشت P' راس تکراری دارد و اولین راسی که تکرار می‌شود، راس i است. دو حالت در نظر می‌گیریم:

حالت اول) راس i، راس میانی یال جهت‌دار باشد. در این‌صورت، با شروع از راس i، یک غنچه داریم. (غنچه‌ی $\mathcal{C}$ در شکل ۵)



شکل ۵: حالت اول

حالت دوم) راس i، راس انتهایی یال جهت‌دار باشد. با توجه به این که با یک یال تطابق به راس انتهایی یال جهت‌دار وارد می‌شویم و هر راس حداکثر به یک راس دیگر در تطابق متصل است، چون راس i تکرار شده‌است، به وضوح راس متصل به i در تطابق (راس j در شکل ۶) قبل از i در P' تکرار می‌شود. بنابراین این حالت امکان‌پذیر نیست و ادعا ثابت شد.



شکل ۶: حالت دوم

□

با توجه به ادعای بالا، برای پیدا کردن مسیر Mـافزایشی در گراف G، کافی است BFS را در گراف D اجرا کنیم و در صورت لزوم غنچه‌ی پیدا شده را ببندیم و ادامه دهیم. در بخش بعدی الگوریتم پیدا کردن تطابق بیشینه آمده‌است.

۴ الگوریتم

Algorithm 2 Maximum Matching in a Non-Bipartite Graph

```
M ← ∅
for v in the set of Vertices do
    Apply BFS on G from vertice v with alternating non-matching and matching edges
    if There is M-alternating W-W walk then
        Let P be a shortest such walk
        if P is a path then
            M ← MΔP
        else if There is a blossom C then
            G′ ← G/C
            M′ ← M/C
            Apply the algorithm recursively on the G’ with M’
```

قضیه ۶. مرتبه زمانی اجرای الگوریتم فوق از $O(|V|^{۲}|E|)$ می‌باشد.

اثبات. در ابتدا $M = \emptyset$ است و حداکثر این الگوریتم $|V|$ بار تکرار می‌شود. در هر بار اجرا، مدت زمانی که طول می‌کشد تا الگوریتم BFS اجرا شود از مرتبه $|E|$ است و از آن‌جایی که هر مرحله ممکن از به دلیل بازگشتی بودن چندین بار فراخوانی داخلی داشته باشد، تعداد این فراخوانی‌های تو در تو حداکثر $|V|$ است که در نتیجه زمان اجرای الگوریتم از مرتبه $O(|V|^۲|E|)$ خواهد بود.