به نام خدا

تمرین سری دوم
طراحی الگوریتم‌های پیشرفته

نعیم اصفهانی
۸۴۲۰۱۰۰۳

26.2-4

$$c_f(u, v) + c_f(v, u)$$

با توجه به تعریف داریم:

$$= c(u, v) - f(u, v) + c(v, u) - f(v, u)$$

با استفاده از خاصیت تقارنی ($f(u, v) = -f(v, u)$):

$$= c(u, v) + c(v, u)$$

26.3-4

بر روی کاردینالیتی L استقرا می‌زنیم:

اگر $|L| = 1$ باشد که این امر بدیهی است. فرض می‌کنیم شرط برای گراف‌هایی با مجموعه‌ی ورودی دارای کاردینالیتی کوچکتر از n برقرار باشد. گراف G با کاردینالیتی مجموعه‌ی ورودی $n+1$ را درنظر می‌گیریم. دو حالت امکان دارد: (۱)یا به ازای همه‌ی $A \subset L$ شرط $|A| < |N(A)|$ برقرار است و داریم $|A| + 1 \leq |N(A)|$ و (۲)یا به ازای یکی از این زیر مجموعه‌ها داریم $|A| = |N(A)|$ بدون از دست دادن کلیت کوچکترین A با این خاصیت را در نظر می‌گیریم. باتوجه به تساوی کاردینالیتی مجموعه‌ی چپ و راست کافی است به ازای $A \subset L$ اثبات شود چون به هر حال $|A| \leq |N(A)|$ به ازای $A = L$ برقرار می‌شود (در صورت قضیه $A \subseteq L$ بود).

حالت اول:

یک $x \in A$ و یک $y \in N(A)$ انتخاب می‌کنیم. (با توجه به فرض $N(A)$ حتما یک عضو دارد). حال این دو راس و یال‌های متصل به آن‌ها را از گراف حذف کرده. همان طور که گفته شد به ازای همه‌ی $A \subset L$ داریم $|A| + 1 \leq |N(A)|$ بنابراین در گراف جدید حذف یک راس (y) از R از $N(A)$ ها حداکثر یک عضو کم می‌کند:

$$|N'(A)| + 1 \geq |N(A)| \geq |A| + 1 \Rightarrow |A| \leq |N'(A)|$$

(توجه داشته باشیم که $N'(A) = N(A) \vee N'(A) \cup \{y\} = N(A)$) پس برای گراف جدید که $|L| = n$ است شرط مذکور برقرار است؛ جواب (تطابق کامل) ما عملا جواب مساله جدید است که دو عضو x, y به همراه یال متصل کننده‌ی آن‌ها به آن اضافه شده‌اند.

حالت دوم:

به توجه به فرض داریم $A \subset L \wedge |N(A)| = |A|$. دو گراف دوبخشی به صورت زیر می‌سازیم که کاردینالیتی مجموعه‌ی چپ و راستشان برابر است:

$$G^* : L^* = A, R^* = N(A)$$

$$G^{**} : L^{**} = L - A, R^{**} = R - N(A)$$

با استفاده از فرض استقرا نشان می‌دهیم G^*, G^{**} تطابق کامل دارند و با اجتماع تطابق آن‌ها تطابق نهایی را می‌سازیم.

توجه داریم که چون A زیرمجموعه‌ی سره‌ی L بود $\left|L^*\right|,\left|L^{**}\right| \le n$.
برای هر $A^* \subset L^*$ محدودیت $\left|N(A^*)\right|$ در G^* معادل همان محدودیت $\left|N(A)\right|$ در G است. برای G^{**} هم اگر یک $A^{**} \subset L^{**}$ پیدا شود که داشته باشیم $\left|N(A^{**})\right| < \left|A^{**}\right|$، در گراف G خواهیم داشت $\left|N(A \cup A^{**})\right| \le \left|A \cup A^{**}\right| - 1$ چون $N(A \cup A^{**}) = N(A) \cup N(A^{**})$، چون گراف G شرط اساسی را ارضاء می‌کند این امر غیر ممکن است پس گراف G^{**} هم این شرط را ارضاء می‌کند.

26.4-2

در هر عمل *Relabel(u)* همه‌ی یال‌های $(u,v) \in E_f$ را می‌آزمایم. تعداد عمل *Relabel* در هر راس برابر $2|V|-1$ است. بنابراین برای هر دو یال $(u,v),(v,u)$، $4|V|-2$ بار این عمل انجام می‌شود($2|V|-1$ برای u و $2|V|-1$ بار دیگر برای v). تعداد یال‌های رفت و برگشت در E_f برابر $|E|$ است. پس کل زمان مصرفی برای اعمال *Relabel* حداکثر $(4|V|-2)|E| = O(VE)$ است.

26.4-4

صورت مشروحی از این الگوریتم در مقاله‌ی ضمیمه وجود دارد.

26.5-2

باید ثابت کنیم این صف ترتیب توپولوژیک را حفظ می‌کند و شرط انتها هم همان شرط است و در این صورت تمام تحلیل‌هایی که در مورد قبلی انجام دادیم برقرار است.

شرط اتمام:

در این حالت وقتی به انتها می‌رسیم که صف خالی باشد و این در صورتی است که در هیچ‌کدام از عناصر دیگر صف که دیده شده‌اند شاره‌ی اضافه‌ای انتقال داده نشده باشد که این همان شرط قبلی چون در آن حالت هم در همین صورت به انتهای صف می‌رسیدیم.

وضعیت صف:

در مورد وضعیت صف هم با حذف عنصر سر صف عملا کاری می‌کنیم که عناصری که شاره‌ی اضافی ندارند اصولا آزموده نشوند و این همانند شرط الگوریتم اصلی است که در صورت نداشتن شاره‌ی اضافه راس مربوطه چک نمی‌شود.

عملا صف مربوطه به گونه‌ای است که یا یک راس در آن وجود دارد که این عناصر قسمت سمت راست عنصری از آرایه‌ی اصلی هستند که چک می‌شوند و عناصری هم که در صف نیستند شاره‌ی اضافی ندارند و عناصر سمت چپ آن را تشکیل می‌دهند. بنابراین این الگوریتم شکل دیگری از همان الگوریتم است.

26-2

a.

گراف ارائه شده در راهنمایی را تشکیل می‌دهیم و ظرفیت همه‌ی یال‌ها را برابر ۱ قرار می‌دهیم. با این عمل کاری می‌کنیم که یک راس((x_i, y_i)) حداکثر یک واحد شاره‌ی ورودی به آن و حداکثر یک واحد شاره‌ی خروجی از آن داشته باشیم که این عملا باعث می‌شود در صورت گذر شاره از این راس در گراف اصلی متناظر تنها یک یال ورودی و خروجی داشته باشیم و مسیر ساخته شود نه زیر گراف.

روش ۱:

با استفاده از برهان خلف نشان می‌دهیم که الگوریتم بیشترین شاره جواب کوچک‌ترین پوشش مسیری را می‌دهد:
فرض می‌کنیم پوشش جواب بیشترین شاره قابل کاهش باشد. بنابراین در اثر این کاهش یال (V_i, V_j) اضافه شده که دو مسیر از مسیرهای جواب شاره را می‌توانسته بهم وصل کند، راس مقصد(V_j) این یال هیچ ورودی‌ای نداشته است و راس مبدا(V_i) این یال هیچ خروجی‌ای نداشته است تا مسیر بودن حفظ شود. اگر چنین ساختاری وجود داشته باشد، در شبکه‌ی شاره یال‌های $(x_0, x_i), (x_i, y_j), (y_j, y_0)$ وجود داشته‌اند پس در گراف متمم شاره یک مسیر افزایشی وجود داشته و شاره‌ی ما شاره‌ی بیشینه نبوده است.

روش ۲:

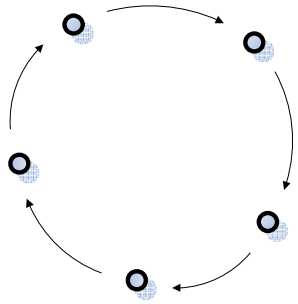
لم: در اثر کاهش پوشش مسیری سراسری حداقل تعداد یال‌ها یک عدد اضافه می‌شود.
اثبات: در عمل کاهش پوشش یا تنها یک یال اضافه می‌شود و دو مسیر مجزا را به هم وصل می‌کند، یا n یال حذف شده و $n+1$ یال اضافه می‌شود. قسمت اول بدیهی است. در مورد قسمت دوم باید گفت که حذف کردن n یال باعث می‌شود تعداد مسیرها n تا بیشتر شوند (با فرض نداشتن دور) و برای کاهش باید تعداد مسیرها $n+1$ تا کم شوند. برای این کار حداقل تعداد $n+1$ یال لازم است. پس برای کاهش تعداد مسیرها باید حداقل یک یال اضافه کرد.

اثبات تئوری:
به همان صورت روش ۱ فرض خلف می‌کنیم: که تعداد مسیرهای کمتری از تعدادی که شاره‌ی بیشینه به ما می‌دهد وجود داشته باشد پس حداقل باید به تعداد یال‌ها یک یال اضافه شود. یال‌هایی که از آن‌ها جریان

می‌گذرد بر روی برش حداقل قرار دارند (در گراف متمم ظرفیت آن‌ها صفر می‌شود) اگر تعداد این یال‌ها بتوانند اضافه شوند ظرفیت برش کمینه قابل افزایش بوده پس شاره‌ی ارائه شده شاره‌ی بیشینه نبوده است.

b.

خیر کار نمی‌کند. گراف زیر را در نظر بگیرید:



جواب الگوریتم در قبال این گراف خود گراف است. زیرا می‌تواند تمام یال‌ها را اضافه کند و در شبکه‌ی شاره جلوی برقرای دور گرفته نمی‌شود. در نتیجه جواب ارائه شده یک دور است و نه یک مسیر. برای حل این مشکل باید دورها را پیدا کرد و از هرکدام یک یال حذف کرد. با توجه به اعمال قانون یک ورودی و خروجی در شبکه، دورها حتما ساده هستند و می‌شود با حذف یک یال مساله را حل کرد.

26-3

a.

اگر یک آزمایش در سمت T یک برش و وسیله‌ی مربوطه در سمت S برش واقع باشد پس یال متصل کننده‌ی آن‌ها حتما روی برش قرار می‌گیرد. از آنجا که ظرفیت این یال نامحدود است، ظرفیت برش هم نامحدود می‌شود. پس اگر یک آزمایش در قسمت T قرار بگیرد، همه‌ی وسایل مورد نیاز آن هم در T قرار می‌گیرند.

b.

ظرفیت برش کمینه برابر مجموع درآمد آزمایش‌های انتخاب نشده و ابزارهای برده شده است. برای بدست آوردن سود باید مجموع درآمد آزمایش‌های انتخاب شده را از هزینه‌ی ابزارهایشان کم کرد که طبق رابطه‌ی زیر کافی است مجموع درآمد همه‌ی آزمایش‌ها را منهای ظرفیت برش کمینه کرد:

$$C_{\min cut} = \sum_{i \in notselectedE} p_i + \sum_{j \in selectdI} c_j$$
$$\sum_{i \in E} p_i = \sum_{i \in notselectedE} p_i + \sum_{i \in selectedE} p_i$$
$$\sum_{i \in E} p_i - C_{\min cut} = \sum_{i \in selectedE} p_i - \sum_{j \in selectdI} c_j$$

می‌گوییم آزمایش‌های داخل مجموعه‌ی T از برش کمینه آزمایش‌هایی هستند که باید انجام گیرند. برای اثبات این موضوع باید دو چیز اثبات شوند:

۱. انتخاب آزمایش‌های مذکور سود را منفی نمی‌کند.
۲. سود ارائه شده توسط آزمایش‌های داخل T قابل افزایش نیست.

مورد اول:

فرض می‌کنیم که سود در صورت انتخاب آزمایش‌های داخل مجموعه‌ی T برای برش کمینه منفی باشد. برش جدیدی را مطرح می‌کنیم: یال‌های اتصال دهنده‌ی آزمایش‌های انتخاب شده به t را به جای یال‌های ابزارهایش در بر گیرد در بر گیرد (علاوه بر آزمایش‌های قبلی که در بر گرفته و داخل T نبوده‌اند). تفاوت برش جدید با برش قبلی(برش کمینه) در این است که یال‌های مربوط به تمام ابزارهای آزمایش‌های انتخاب شده از برش برداشته شده و یال‌های مربوط به آزمایش‌های آن‌ها اضافه شده. طبق فرض منفی بودن سود ظرفیت یال‌های آزمایش‌ها از ابزارهای آن‌ها کم‌تر است بنابراین ظرفیت برش جدید کم‌تر از برش قبلی است. این خلاف فرض است چون برش قبلی کمینه بود.

مورد دوم:

فرض می‌کنیم که سود قابل افزایش باشد. بنابراین آزمایش‌ها و ابزارهای متفاوت با آنچه ما انتخاب کرده بودیم را در بر می‌گرفته است. برش جدید را به این ترتیب تعریف می‌کنیم: این آزمایش‌ها و ابزارهایشان را درون مجموعه‌ی T قرار می‌دهیم، می‌دانیم که سود حاصل از این کار بیش‌تر از حالت قبل است و عبارت $\sum_{i \in E} p_i - C_{cut}$ بیش‌تر از حالت قبل است. قسمت اول این عبارت که ثابت است پس برای بیشتر شدن عبارت باید قسمت دوم که از یک مقدار ثابت کم می‌شود کم‌تر شود. به عبارتی برش ارائه شده ظرفیت کمتری دارد و این محال است چون برش قبلی برش کمینه بود.

بنابراین آزمایش‌های درون T از برش کمینه آزمایش‌هایی هستند که باید انتخاب شوند.

c.

تعداد راس‌های گراف ساخته شده برابر $n+m+2$ است و تعداد یال‌ها هم برابر $n+m+r$ است. بهترین الگوریتم برای بدست آوردن بیشترین شاره الگوریتم *LIFT-TO-FRONT* $O(V^3)$ است. با این الگوریتم که برای ما هزینه‌ی $O((n+m+2)^3)$ دارد بیشترین شاره را پیدا کرده و به تبع آن برش کمینه را با $O(E)$ پیدا می‌کنیم. پس هزینه‌ی ما می‌شود: $O((n+m+r)+(n+m+2)^3) = O((n+m+2)^3)$