دانشکده ریاضی و علوم کامپیوتر

پایان نامه برای دریافت درجه کارشناسی ارشد در رشته ریاضی کاربردی
گرایش تحقیق در عملیات

برش کمینه در گراف

استاد راهنما

دکتر محمود امین طوسی

استاد مشاور

دکتر مهدی زعفرانیه

پژوهشگر:

فاطمه سادات حسینی

شهریور ۱۳۹۳

سوگند نامه دانش آموختگان دانشگاه حکیم سبزواری

به نام خداوند جان و خرد کزین برتر اندیشه بر نگذرد

اینک که به خواست آفریدگار پاک، کوشش خویش و بهره گیری از دانش استادان و سرمایه های مادی و معنوی این مرز و بوم، توشه ای از دانش و خرد گردآورده ام، در پیشگاه خداوند بزرگ سوگند یاد می کنم که در به کارگیری دانش خویش، همواره بر راه راست ودرست گام بردارم. خداوند بزرگ، شما شاهدان، دانشجویان و دیگر حاضران را به عنوان داورانی امین گواه می گیرم که ازهمه دانش و توان خود برای گسترش مرزهای دانش بهره گیرم و از هیچ کوششی برای تبدیل جهان به جایی بهتر برای زیستن، دریغ نورزم. پیمان می بندم که همواره کرامت انسانی را در نظر داشته باشم و همنوعان خود را در هر زمان و مکان تا سر حد امکان یاری دهم. سوگند می خورم که در به کارگیری دانش خویش به کاری که با راه و رسم انسانی، آیین پرهیزگاری، شرافت و اصول اخلاقی برخاسته از ادیان بزرگ الهی، به ویژه دین مبین اسلام، مباینت دارد دست نیازم. همچنین در سایه اصول جهان شمول انسانی و اسلامی، پیمان می بندم از هیچ کوششی برای آبادانی و سرافرازی میهن و هم میهنانم فروگذاری نکنم و خداوند بزرگ را به یاری طلبم تا همواره در پیشگاه او و در برابر وجدان بیدار خویش و ملت سرافراز، بر این پیمان تا ابد استوار بمانم.

نام و نام خانوادگی: فاطمه سادات حسینی

تاریخ و امضا:

تأییدیه ی صحت و اصالت نتایج

باسمه تعالی

اینجانب فاطمه سادات حسینی به شماره دانشجویی ۹۱۱۳۱۳۳۰۱۱ دانشجوی رشته ریاضی کاربردی مقطع تحصیلی کارشناسی ارشد تأیید می نمایم که کلیه ی نتایج این پایان نامه حاصل کار اینجانب و بدون هرگونه دخل و تصرف است و موارد نسخه برداری شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده ام. درصورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انضباطی ...) با اینجانب رفتار خواهد شد و حق هرگونه اعتراض درخصوص احقاق حقوق مکتسب و تشخیص و تعیین تخلف و مجازات را از خویش سلب می نمایم. در ضمن، مسؤولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذی صلاح (اعم از اداری و قضایی) به عهده ی اینجانب خواهد بود و دانشگاه هیچ گونه مسؤولیتی در این خصوص نخواهد داشت.

نام و نام خانوادگی: فاطمه سادات حسینی

تاریخ و امضا:

مجوز بهره برداری از پایان نامه

بهره برداری از این پایان نامه در چهارچوب مقررات کتابخانه و با توجه به محدودیتی که توسط استاد راهنما به شرح زیر تعیین می شود، بلامانع است:

☐ بهره برداری از این پایان نامه برای همگان بلامانع است.

☐ بهره برداری از این پایان نامه با اخذ مجوز از استاد راهنما، بلامانع است.

☐ بهره برداری از این پایان نامه تا تاریخ ..... ممنوع است.

استاد راهنما: دکتر محمود امین طوسی

تاریخ:

امضا:

تقدیم به:

همسر و فرزندانم

و

پدر و مادرم

قدردانی

سپاس خداوندگار حکیم را که با لطف بی کران خود، آدمی را زیور عقل آراست.

در آغاز وظیفه خود می دانم از زحمات بی دریغ استاد راهنمای خود، جناب آقای دکتر محمود امین طوسی، صمیمانه تشکر و قدردانی کنم که قطعاً بدون راهنمایی های ارزنده ایشان، این مجموعه به انجام نمی رسید.

از جناب آقای دکتر مهدی زعفرانیه که زحمت مطالعه و مشاوره این رساله را تقبل فرمودند و در آماده سازی این رساله، به نحو احسن اینجانب را مورد راهنمایی قرار دادند، کمال امتنان را دارم.

همچنین لازم می دانم از پدید آورندگان بسته زی پرشین، مخصوصاً جناب آقای وفا خلیقی، که این پایان نامه با استفاده از این بسته، آماده شده است و همه دوستانمان در گروه پارسی لاتک کمال قدردانی را داشته باشم.

در پایان، بوسه می زنم بر دستان خداوندگاران مهر و مهربانی، پدر و مادر عزیزم و بعد از خدا، ستایش می کنم وجود مقدس شان را و تشکر می کنم از خانواده عزیزم به پاس عاطفه سرشار و گرمای امیدبخش وجودشان، که بهترین پشتیبان من بودند.

فاطمه سادات حسینی

شهریور ۱۳۹۳

فرم چکیده ی پایان نامه ی دوره ی تحصیلات تکمیلی

مدیریت تحصیلات تکمیلی

دانشگاه حکیم سبزواری

نام خانوادگی دانشجو:حسینی	نام:فاطمه سادات	ش. دانشجویی:۹۱۱۳۱۳۳۰۱۱
استاد راهنما: دکتر محمود امین طوسی		
استاد مشاور: دکتر مهدی زعفرانیه		
دانشکده ریاضی و علوم کامپیوتر	رشته: ریاضی کاربردی	گرایش:تحقیق در عملیات
مقطع: کارشناسی ارشد	تاریخ دفاع:شهریور ۱۳۹۳	تعداد صفحات: ۹۱
عنوان پایان نامه: برش کمینه در گراف		
کلید واژه ها:برش کمینه ،جریان بیشینه،الگوریتم استور واگنر، الگوریتم کارگر،الگوریتم های فرا ابتکاری		

چکیده: در نظریه گراف منظور از برش، تقسیم رئوس گراف به دو زیرمجموعه ناتهی جدا از هم S و (V/S) می باشد.یالهای برش به یال هایی گویند که بین این دو زیر مجموعه آن باشند. در مسئله برش کمینه هدف یافتن این دو زیر مجموعه به نحوی است که ظرفیت یالهای برش کمینه شود.

با توجه به برابری مقدار جریان بیشینه با برش کمینه در گراف ، روشهای اصلی مواجهه با این مسئله به دو دسته روشهای مبتنی بر جریان در گراف و سایر روشها تقسیم می شوند. در این پایان نامه به بررسی روشهایی از هر دو دسته می پردازیم.

روشهای به دست آوردن جریان بیشینه در گراف های عمومی به دو دسته کلی الگوریتم های مسیر افزایشی و الگوریتم های ارسال پیش جریان تقسیم می شوند. در فصل دو به بررسی روشهای مبتنی بر جریان می پردازیم که تعمیمی از اولین الگوریتم مسیر افزایشی ارائه شده توسط فورد و فولکرسون می باشند و تفاوت اصلی آنها در نحوه انتخاب مسیر افزایشی است.

از جمله روشهای غیر مبتنی بر جریان بررسی شده در این پایان نامه الگوریتم استور واگنر، الگوریتم کارگر و روشهای فرا ابتکاری جستجوی ممنوعه و شبیه سازی تبریدی می باشند.

فهرست مطالب

فهرست تصاویر

فهرست جداول

فهرست الگوریتم ها

پیش گفتار

مسئله برش کمینه در گراف از جمله مسائل مشهوری است که تحقیقات متعددی را در علوم مختلف و من جمله در ریاضیات و کامپیوتر به خود معطوف نموده است. به صورت خلاصه هدف در مسئله برش کمینه در گراف، افراز مجموعه رئوس گراف به نحوی است که مجموع وزن لبه های بین دو مجموعه (لبه های برش) کمینه گردد.

این پایان نامه شامل ۴ فصل است:

در فصل اول، تعاریف مقدماتی و موردنیاز مسئله، روشهای اصلی مواجهه با مسئله بیان خواهند شد. روشهای اصلی مواجهه با این مسئله به دو دسته روشهای مبتنی بر جریان در گراف و سایر روشها تقسیم می شوند.

در فصل ۲ روشهای مبتنی بر جریان به تفصیل بیان خواهند شد.

فصل ۳ به ذکر روشهای غیر مبتنی بر جریان تخصیص یافته است. در این فصل دو شیوه پیشنهادی در این پایان نامه نیز بیان خواهند شد.

در فصل چهارم پیشنهادات این پایان نامه و نتایج پیاده سازی برخی روشهای مذکور در فصول پیشین ، خواهد آمد.

فصل ۱

مقدمه

در نظریه گراف منظور از برش، تقسیم رئوس گراف به دو زیرمجموعه ناتهی جدا از هم S و V/S می باشد. به یال هایی که بین دو مجموعه ی برش باشند، یالهای برش گویند. در مسئله برش کمینه هدف یافتن این دو زیر مجموعه به نحوی است که ظرفیت یالهای برش کمینه شود. مدل ریاضی برش کمینه به صورت زیر است:

$$\min \sum_{i\in S, j\in (V/S)} c_{ij}$$
$$st; \quad S, (V/S) \neq \emptyset$$

که c_{ij} ظرفیت یال (i,j) است. مسئله $st-$ برش کمینه حالت خاصی از مسئله برش کمینه است که عبارت است از مینیمم کردن ظرفیت یالهای برش به طوری که $s \in S\ ,\ t \in (V/S)$ ومدل ریاضی آن به صورت زیر است :

$$\min \sum_{i\in s, j\in (V/S)} c_{ij}$$
$$st; \quad s \in S, t \in (V/S)$$

در ادامه در ابتدا برخی اصطلاحات و مفاهیمی که در ادامه نوشتار موردنیاز هستند بیان و تعریف خواهند شد و سپس مرور مختصری بر برخی روشهای مواجهه با این مسئله خواهیم داشت.
برای تعاریف وقضایای این فصل تا پایان قضیه ۱-۲-۶ از مرجع [۲] و پس از آن از مرجع [۱] و [۳] استفاده شده است.

۱-۱ گراف

تعریف ۱-۱-۱. گراف G، شامل یک مجموعه ناتهی متناهی $V(G)$ و یک مجموعه $E(G)$ (که می تواند تهی باشد) از زیر مجموعه های دو عضوی $V(G)$ است که $V(G)$، **مجموعه رئوس گراف** و $E(G)$ **مجموعه یالهای**[۱] **گراف** نامیده می شود.

تعریف ۱-۱-۲. [۲] **گراف جهت دار** D، شامل یک مجموعه ناتهی متناهی $V(D)$ و یک مجموعه $E(D)$ (که می تواند تهی باشد) زیر مجموعه ای از زوج های مرتب $V(D)$ است که $V(G)$، **مجموعه رئوس گراف** و $E(D)$ **مجموعه کمان های**[۳] **گراف** نامیده میشود.

تعریف ۱-۱-۳. گراف جهتدار $D = (V, E)$ را که دارای شرایط زیر است یک **شبکه جریان**[۴] می نامیم و با N نشان میدهیم:

- هر یال $(u, v) \in E$ دارای یک میزان ظرفیت غیر منفی یعنی $c(uv) \geq ۰$ است. ظرفیت زوج های (u, v) که در E وجود ندارند را صفر در نظر می گیریم.
- دو راس مجزای s با نام مبدا و t با نام مقصد در این گراف وجود دارد هر راس دیگر گراف در مسیری از s به t قرار می گیرد. درجه ورودی s ودرجه خروجی t صفراست ویا کمانهای ورودی به s و کمانهای خروجی در t زائدند و در مسائل جریان شبکه اثر نمی گذارد.

تعریف ۱-۱-۴. تابع صحیح وغیر منفی C۶ که روی $E(D)$ تعریف می شود و مقادیرصحیح غیر منفی و ∞ را می گیرد **تابع ظرفیت**[۵] نامیده میشود.که ∞ ظرفیت نامحدود را نشان میدهد.اگر ظرفیتها گویا بودند ،آنها را در عدد صحیح مناسب (کمترین ضریب مشترک) ضرب می کنیم و اعداد گنگ را بااعداد گویا تقریب می زنیم.

تعریف ۱-۱-۵. برای هر $x \in V(D)$،

$N^+(x) = \{y \in V(D) | (x, y) \in E(D)\}$ را **همسایگی خروجی**[۶]

و $N^-(x) = \{y \in V(D) | (y, x) \in E(D)\}$ را **همسایگی ورودی**[۷] می نامیم .

تعریف ۱-۱-۶. برای هر رأس دلخوه x

$A_۱ = \sum_{y \in N^+(x)} f(x, y) - \sum_{y \in N^-(x)} f(y, x)$ را **برایند جریان خروجی** و

$A_۲ = \sum_{y \in N^-(x)} f(y, x) - \sum_{y \in N^+(x)} f(x, y)$ را **برایند جریان ورودی** می نامیم.

---

[۱] edge
[۲] directed graph
[۳] arc
[۴] network flow
[۵] capacity
[۶] in neighborhood
[۷] out neighborhood

تعریف ۱-۱-۷. تابع صحیح وغیر منفی f روی $E(D)$ که در شرایط زیر صدق می کند را **جریان** می نامیم .

$$\circ \leqslant f(a) \leqslant c(a) \qquad \forall a \in E(D) \qquad (۱-۱)$$

$$\sum_{y \in N^+(x)} f(x,y) = \sum_{y \in N^-(x)} f(y,x) \qquad \forall x \in V(D) - \{s,t\} \qquad (۱-۲)$$

خاصیت اول به این معنا است که جریان در یک کمان نمی تواند از ظرفیتش تجاوز کند و به آن **محدودیت ظرفیت**[۸] می گویند و خاصیت دوم یعنی اگر x رأسی باشد که مبدأ و مقصد نیست، $A_۱$ و $A_۲$ هردو صفرند و به آن **معادله بقا**[۹] می گویند.

جریان در شبکه N به صورت زیر نیز تعریف می شود:

$$f(N) = \sum_{y \in N^+(s)} f(s,y) - \sum_{y \in N^-(s)} f(y,s)$$

که همان برایند جریان خروجی از s است.

می توانیم فرض کنیم D نامتقارن است یعنی اگر (x,y) کمانی از D باشد ،آنگاه (y,x) کمانی از D نیست.این فرض معقول است زیرا اگر $(x,y),(y,x)$ هردو کمانهای D باشند می توان یکی از آنها مثلاً (y,x) را با مسیر y,z,x جابجا کرد که $c(y,z) = c(z,x) = c(y,x)$ و z رأس جدید است. اگر D' گراف جهتدار بدست آمده از D به روش بالا و N' شبکه حاصل باشد. آنگاه تناظر یک به یک بین جریانهای N و N' برقرار است.بنابراین می توان فرض کرد گراف جهتدار زمینه نا متقارن است.

اگر X و Y زیر مجموعه های نا تهی از $V(D)$ باشند(ممکن است عضو مشترک داشته باشند)داریم:

$$(X,Y) = \{(x,y) | x \in X, y \in Y\} \subseteq E(D)$$
$$f(X,Y) = \sum_{(x,y) \in (X,Y)} f(x,y)$$
$$c(X,Y) = \sum_{(x,y) \in (X,Y)} c(x,y)$$

و اگر $(X,Y) = \emptyset$ آنگاه $f(X,Y) = c(X,Y) = \circ$.

تعریف ۱-۱-۸. مجموعه ای از کمانها به صورت $(P,\bar{P})$ در D که P مجموعه رأسهایی شامل s است که شامل t نیست و $\bar{P} = V(D) - P$ را یک **برش**[۱۰] در شبکه N می نامیم .

$c(P,\bar{P})$ ظرفیت برش و $f(P,\bar{P})$ جریان از مجموعه P به مجموعه $\bar{P}$ است.

[۸] capacity constrain
[۹] conservation equation
[۱۰] cut

قضیه ۱-۱-۹. برای هر برش $(P, \bar{P})$ از شبکه N رابطه ی زیر همواره برقرار است.

$$\begin{aligned} f(N) &= \sum_{(x,y)\in(P,\bar{P})} f(x,y) - \sum_{(x,y)\in(\bar{P},P)} f(x,y) \\ &= f(P,\bar{P}) - f(\bar{P},P) \end{aligned}$$

اثبات: با توجه به تعریف جریان و معادله بقا داریم

$$\begin{aligned} f(N) &= \sum\nolimits_{y\in N^+(s)} f(s,y) - \sum\nolimits_{y\in N^-(s)} f(y,s) \\ \circ &= \sum\nolimits_{y\in N^+(x)} f(x,y) - \sum\nolimits_{y\in N^-(x)} f(y,x) \qquad\qquad \forall x \in P - \{s\} \end{aligned}$$

با جمع این دو رابطه، رابطه زیر بدست می آید:

$$f(N) = \sum_{x\in P} (\sum\nolimits_{y\in N^+(x)} f(x,y) - \sum\nolimits_{y\in N^-(x)} f(y,x))$$

$$= \sum_{x\in P} \sum\nolimits_{y\in N^+(x)} f(x,y) - \sum_{x\in P} \sum\nolimits_{y\in N^-(x)} f(y,x)$$

از طرفی

$$\begin{aligned} \sum_{x\in P} \sum\nolimits_{y\in N^+(x)} f(x,y) &= \sum_{(x,y)\in(P,P)} f(x,y) + \sum_{(x,y)\in(P,\bar{P})} f(x,y) \\ \sum_{x\in P} \sum\nolimits_{y\in N-(x)} f(x,y) &= \sum_{(x,y)\in(P,P)} f(y,x) + \sum_{(x,y)\in(P,\bar{P})} f(y,x) \end{aligned}$$

که $\sum_{(x,y)\in(P,P)} f(x,y)$ و $\sum_{(x,y)\in(P,P)} f(y,x)$ هردو برابر با مجموع جریان در تمام کمانهایی که دو انتهایشان در P است لذا با هم برابرند و داریم:

$$f(N) = \sum_{(x,y)\in(P,\bar{P})} f(x,y) - \sum_{(x,y)\in(P,\bar{P})} f(y,x)$$

قضیه ۱-۱-۱۰. جریان در هر شبکه حداکثر به بزرگی ظرفیت برش کمینه است.

$$f(N) \leqslant \min\{c(P,\bar{P})\}$$

اثبات: طبق قضیه ۱-۱-۹ وتعریف ۱-۱-۷ برای هر برش $(P,\bar{P})$ از شبکه N داریم:

$$\begin{aligned} f(N) &= \sum_{(x,y)\in(P,\bar{P})} f(x,y) - \sum_{(x,y)\in(P,\bar{P})} f(y,x) \\ &\leqslant \sum_{(x,y)\in(P,\bar{P})} f(x,y) = f(P,\bar{P}) \leqslant c(P,\bar{P}) \end{aligned}$$

چون این رابطه برای هر برش دلخواه برقرار است، داریم:

$$f(N) \leqslant \min\{c(P,\bar{P})\}$$

قضیه ۱-۱-۱۱. جریان در شبکه N با جریان ورودی به راس مقصد، برابر است.

$$f(N) = \sum\nolimits_{y\in N-(t)} f(y,t) - \sum\nolimits_{y\in N^+(t)} f(t,y)$$

اثبات: فرض کنید $P = V(D) - \{t\}, \bar{P} = \{t\}$ بنابراین،
اگر $y = t$ و $x \in N^-(x)$ ، آنگاه $(x,y) \in (P,\bar{P})$ و اگر $y = t$ و $x \in N^+(x)$ ، آنگاه $(x,y) \in (\bar{P},P)$
و بنا به قضیه ۱-۱-۹ داریم:

$$f(N) = \sum_{(x,y)\in(P,\bar{P})} f(x,y) - \sum_{(x,y)\in(P,\bar{P})} f(y,x) = \sum\nolimits_{y\in N^-(t)} f(y,t) - \sum\nolimits_{y\in N^+(t)} f(t,y)$$

۱-۲ جریان ماکزیمم

اگر برای هر جریان f و هر برش $(x,\bar{x})$ از N داشته باشیم:

$$c(x,\bar{x}) \leqslant c(P,\bar{P})$$
$$f^*(N) \geq f(N)$$

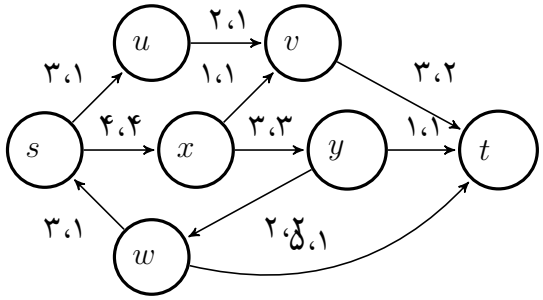
آنگاه f^* جریان ماکزیمم و $(P,\bar{P})$ برش کمینه است.

توجه داریم که همواره $(s, V(D) - \{s\})$ یک برش و $f(a) = 0 \quad \forall a \in E(D)$ یک تابع جریان برایN است بنابراین N دارای کمترین برش وبیشترین جریان است.

تعریف ۱-۲-۱. یک دنباله متناهی و متناوب از رأسها و کمانها که با رأس u_0 آغاز وبه رأس u_n پایان می پذیرد

و رأس تکراری ندارد را یک **شبه مسیر**[11] $u_0 - u_n$ می نامیم.

مثال ۱-۲-۲. شبکه زیر را در نظر بگیرید



جریان در شاخه (s,u) برابر با ۱ وظرفیت در این شاخه برابربا ۳ می باشد.

جریان اولیه درشبکه برابر $f(N) = ۵ - ۱ = ۴$ می باشد.

$Q_۱ : s, (w,s), w, (w,t), t$ یک شبه مسیر است.

تعریف ۱-۲-۳. شبه مسیر Q در D، $f-$ **اشباع نشده**[12] است اگر برای هر $۱ \leqslant i \leqslant n$ یکی از شرایط زیر برقرار باشد:

$$a_i = (u_{i-۱}, u_i) \qquad f(a_i) < c(a_i) \qquad (۳-۱)$$

$$a_i = (u_i, u_{i-۱}) \qquad f(a_i) > ۰ \qquad (۴-۱)$$

اگر Q یک $st-$ شبه مسیر $f-$اشباع نشده باشد ، آنگاهQ یک شبه مسیر $f-$افزایشی[13] است.

قضیه ۱-۲-۴. جریان f در شبکه N بیشترین جریان است اگر و تنها اگر هیچ شبه مسیر $f-$ افزایشی در گراف زمینه آن D نباشد.

اثبات: فرض کنید s و t مبدا و مقصد N ، D گراف زمینه آن و $Q : s = u_۰, a_۱, u_۱, ..., u_{n-۱}, a_n, u_n = t$ یک شبه مسیر $f-$ افزایشی باشد . برای هر $۱ \leqslant i \leqslant n$ معادله ۱-۳ یا ۱-۴ برقرار است. اگر معادله ۱-۳ برقرار باشد، آنگاه قرار می دهیم؛ $\Delta_i = c(a_i) - f(a_i)$ و اگر معادله ۱-۴ برقرار باشد، آنگاه قرار می دهیم؛ $\Delta_i = f(a_i)$ و Δ را به صورت زیر تعریف می کنیم:

$$\Delta = \min\{\Delta_i | ۱ \leqslant i \leqslant n\}$$

و تابع f^* را روی کمانهای $E(D)$ به صورت زیر تعریف میکنیم:

---

[11] semi path
[12] unsaturated
[13] augmenting

$f^*(a) = \begin{cases} f^*(a) = f(a) + \Delta & a \in E(D), a = (u_{i-1}, u_i) \\ f^*(a) = f(a) - \Delta & a \in E(D), a = (u_i, u_{i-1}) \\ f^*(a) = f(a) & a \notin E(D) \end{cases}$ کافی است نشان دهیم f^* یک جریان برای N است که $f^*(N) \geqslant f(N)$.

اگر حالت ۱ برقرار باشد چون f جریان است $0 \leqslant f(a) \leqslant c(a)$ و لذا $0 \leqslant f^*(a) \leqslant c(a)$

اگر حالت ۲ برقرار باشد چون f جریان است $0 \leqslant f(a) \leqslant c(a)$ و نیز $f^*(a) = f(a) + \Delta$ بنابراین $0 \leqslant f^*(a)$ از طرفی طبق آنچه در بالا گفته شد اگر حالت ۲ برای a برقرار باشد، آنگاه طبق تعریف Δ داریم:

$\Delta \leqslant c(a) - f(a)$ بنابراین $f^*(a) = f(a) + \Delta \leqslant f(a) + c(a) - f(a) = c(a)$

اگر حالت ۳ برقرار باشد چون f جریان است $0 \leqslant f(a) \leqslant c(a)$ و نیز $f^*(a) = f(a) - \Delta$ و

$f(a) - \Delta \leqslant f(a) \leqslant c(a)$ از طرفی طبق آنچه در بالا گفته شد اگر حالت ۳ برای a برقرار باشد طبق تعریف Δ داریم $\Delta \leqslant f(a)$ بنابراین $0 \leqslant f^*(a)$.

پس در هر حالت شرط اول جریان بودن (محدودیت ظرفیت) برقرار است. حال نشان می دهیم f^* در شرط معادله بقا صدق می کند:

$$\sum_{y \in N^+(x)} f(x,y) = \sum_{y \in N^-(x)} f(y,x) \quad \forall x \in V(D) - \{u_0, u_n\}$$

اگر x راسی باشد که در Q نیست (با هیچ u_i برابر نباشد) آنگاه

$$\forall y \in N^+(x) \quad f(x,y) = f^*(x,y)$$
$$\forall y \in N^-(x) \quad f(y,x) = f^*(y,x)$$

وچون f جریان است $\sum_{y \in N^+(x)} f(x,y) = \sum_{y \in N^-(x)} f(y,x)$

بنابراین در این حالت $\sum_{y \in N^+(x)} f^*(x,y) = \sum_{y \in N^-(x)} f^*(y,x)$

اگر $x \in Q$

باشد، آنگاه یک i وجود دارد که $1 \leqslant i \leqslant n-1$ و $x = u_i$.

بنابراین چهار حالت داریم:

۱. $a_i = (u_i, u_{i-1})$ $a_{i+1} = (u_i, u_{i+1})$

۲. $a_i = (u_{i-1}, u_i)$ $a_{i+1} = (u_{i+1}, u_i)$

۳. $a_i = (u_{i-۱}, u_i)\ \ a_{i+۱} = (u_i, u_{i+۱})$

۴. $a_i = (u_i, u_{i-۱})\ \ a_{i+۱} = (u_{i+۱}, u_i)$

در حالت اول f^* برای راس u_i فقط روی دو یال $a_{i+۱}$ و a_i تغییر می کند (چون هر راس تنها دو بار در شبه مسیر ظاهر می شود).

که در این حالت هر دو یال خروجی u_i اند بنابراین:

$$\sum_{y \in N^+(x)} f^*(x,y) = \sum_{y \in S} f(x,y) + f^*(a_i) + f^*(a_i + ۱)$$
$$\sum_{y \in N^-(x)} f^*(y,x) = \sum_{y \in N^-(x)} f(y,x)$$

که $S = N^+(x) - \{u_{i-۱}, u_{i+۱}\}$ و بنا به تعریف f^* : $f^*(a_i) = f(a_i) + \Delta$ و $f^*(a_{i+۱}) = f(a_{i+۱}) - \Delta$

بنابراین $\sum_{y \in N^+(x)} f^*(y,x) = \sum_{y \in N^+(x)} f(y,x)$

وچون f جریان است، $\sum_{y \in N^+(x)} f(x,y) = \sum_{y \in N^-(x)} f(y,x)$. درنتیجه

$$\sum_{y \in N^+(x)} f^*(x,y) = \sum_{y \in N^-(x)} f^*(y,x)$$

در حالت دوم هر دو یال ورودی u_i اند بنابراین

$$\sum_{y \in N^-(x)} f^*(y,x) = \sum_{y \in S} f(x,y) + f^*(a_i) + f^*(a_i + ۱)$$
$$\sum_{y \in N^+(x)} f^*(x,y) = \sum_{y \in N^+(x)} f(y,x)$$

که $S = N^-(x) - \{u_{i-۱}, u_{i+۱}\}$ و بنا به تعریف f^* : $f^*(a_i) = f(a_i) - \Delta$ و $f^*(a_{i+۱}) = f(a_{i+۱}) + \Delta$

بنابراین $\sum_{y \in N^-(x)} f^*(y,x) = \sum_{y \in N^-(x)} f(y,x)$

وچون f جریان است، $\sum_{y \in N^+(x)} f(x,y) = \sum_{y \in N^-(x)} f(y,x)$. درنتیجه

$$\sum_{y \in N^+(x)} f^*(x,y) = \sum_{y \in N^-(x)} f^*(y,x)$$

در حالت سوم

$$\sum_{y\in N^-(x)} f^*(y,x) = \sum_{y\in S_1} f(y,x) + f^*(a_i)$$
$$\sum_{y\in N^+(x)} f^*(y,x) = \sum_{y\in S_2} f(y,x) + f^*(a_i+1)$$

که $S_1 = N^-(x) - \{u_{i-1}\}$ و $S_2 = N^+(x) - \{u_{i+1}\}$. و بنا به تعریف f^* :
بنابراین $f^*(a_i) = f(a_i) + \Delta$ و $f^*(a_{i+1}) = f(a_{i+1}) + \Delta$

$$\sum_{y\in N^-(x)} f^*(y,x) = \sum_{y\in N^-(x)} f(y,x) + \Delta$$
$$\sum_{y\in N^+(x)} f^*(y,x) = \sum_{y\in N^+(x)} f(y,x) + \Delta$$

وچون f جریان است، $\sum_{y\in N^+(x)} f(x,y) = \sum_{y\in N^-(x)} f(y,x)$ درنتیجه

$\sum_{y\in N^+(x)} f^*(x,y) = \sum_{y\in N^-(x)} f^*(y,x).$

و در حالت چهارم

$$\sum_{y\in N^-(x)} f^*(y,x) = \sum_{y\in S_1} f(y,x) + f^*(a_i+1)$$
$$\sum_{y\in N^+(x)} f^*(y,x) = \sum_{y\in S_2} f(y,x) + f^*(a_i)$$

که $S_1 = N^-(x) - \{u_{i+1}\}$ و $S_2 = N^+(x) - \{u_{i-1}\}$ و بنا به تعریف f^*:
بنابراین $f^*(a_i) = f(a_i) - \Delta$ و $f^*(a_{i+1}) = f(a_{i+1}) - \Delta$

$$\sum_{y\in N^-(x)} f^*(y,x) = \sum_{y\in N^-(x)} f(y,x) - \Delta$$
$$\sum_{y\in N^+(x)} f^*(y,x) = \sum_{y\in N^+(x)} f(y,x) - \Delta$$

وچونfجریان است، $\sum_{y\in N^+(x)} f(x,y) = \sum_{y\in N^-(x)} f(y,x)$. درنتیجه

$\sum_{y\in N^+(x)} f^*(x,y) = \sum_{y\in N^-(x)} f^*(y,x)$

بنابراین f^* یک جریان در N است. حال نشان می دهیم $f^*(N) \geq f(N)$.

اگر $a_1 = (s,u)$ آنگاه

$$\begin{aligned} f^*(s,u) &= f(s,u) + \Delta \\ f^*(s,y) &= f(s,y) && \forall y \in N^+(s) - \{u\} \\ f^*(y,s) &= f(y,s) && \forall y \in N^-(s) \end{aligned}$$

بنابراین

$$\begin{aligned} f^*(N) &= \sum_{y \in N^+(s)} f^*(s,y) - \sum_{y \in N^-(s)} f^*(y,s) \\ &= f^*(s,u) + \sum_{y \in N^+(s) - \{u\}} f^*(s,y) - \sum_{y \in N^-(s)} f^*(y,s) \\ &= f(s,u) + \Delta + \sum_{y \in N^+(s) - \{u\}} f(s,y) - \sum_{y \in N^-(s)} f(y,s) \\ &= \Delta + \sum_{y \in N^+(s)} f(s,y) - \sum_{y \in N^-(s)} f(y,s) \\ &= f(N) + \Delta \end{aligned}$$

و اگر $a_1 = (u,s)$ آنگاه

$$\begin{aligned} f^*(u,s) &= f(u,s) - \Delta \\ f^*(y,s) &= f(y,s) && \forall y \in N^-(s) - \{u\} \\ f^*(s,y) &= f(s,y) && \forall y \in N^+(s) \end{aligned}$$

بنابراین

$$
\begin{aligned}
f^*(N) &= \sum_{y\in N^+(s)} f^*(s,y) - \sum_{y\in N^-(s)} f^*(y,s) \\
&= \sum_{y\in N^+(s)} f^*(s,y) - \sum_{y\in N^-(s)-\{u\}} f^*(y,s) - f^*(u,s) \\
&= \sum_{y\in N^+(s)} f(s,y) - \sum_{y\in N^-(s)-\{u\}} f(y,s) - (f(u,s)-\Delta) \\
&= \sum_{y\in N^+(s)} f(s,y) - \sum_{y\in N^-(s)} f(y,s) + \Delta \\
&= f(N)+\Delta
\end{aligned}
$$

یعنی در هر حالت $f^*(N) \geqslant f(N)$. در نتیجه f بیشترین جریان نیست.

بر عکس، فرض کنید هیچ شبه مسیر $f-$افزایشی در D وجود نداشته باشد، ابتدا نشان می دهیم که در این حالت یک برش $(P,\bar{P})$ در N وجود دارد به گونه ای که

$$
\begin{aligned}
&\forall a \in (P,\bar{P}) && f(a) = c(a) \\
&\forall a \in (\bar{P},P) && f(a) = 0
\end{aligned}
$$

فرض کنید P نمایانگر مجموعه تمام راسهای x در D باشد که برای آن یک $sx-$ شبه مسیر $f-$ اشباع نشده وجود داشته باشد، در این صورت $t \notin P$, $s \in P$ درنتیجه $(P,\bar{P})$ یک برش در N است. فرض کنید $(y,w) \in (P,\bar{P})$ در این صورت چون $y \in P$، یک $sy-$ شبه مسیر $f-$ اشباع نشده Q در D وجود دارد. بنابراین $f(y,w) = c(y,w)$؛ در غیر این صورت ، شبه مسیر $Q, (y,w), w$: Q^* یک $sw-$ شبه مسیر $f-$ اشباع نشده است که با فرض $w \notin P$ در تناقض است. به طور مشابه اگر $(y,w) \in (\bar{P},P)$، آنگاه $f(y,w) = 0$. بنابراین با توجه به قضیه ۱-۱-۹

$$
f(N) = f(P,\bar{P}) - f(\bar{P},P) = f(P,\bar{P}) - 0 = c(P,\bar{P})
$$

اگر f^* جریان ماکزیمم و $(X,\bar{X})$ برش کمینه باشد.آنگاه با توجه به قضیه ۱-۱-۱۰، $f^*(N) \leqslant c(X,\bar{X})$. بنابراین

$$
f(N) \leqslant f^*(N) \leqslant c(X,\bar{X}) \leqslant c(P,\bar{P}) = f(N)
$$

که ایجاب می کند $f(N) = f^*(N)$، یعنی f بیشترین جریان است.

قضیه ۱-۲-۵. بیشترین جریان-کمترین برش[۱۴] در هر شبکه بیشترین جریان با ظرفیت کمترین برش برابر است .

اثبات:فرض کنید f بیشترین جریان در شبکه N وD گراف جهت دار زمینه آن باشند ، در این صورت بنا به قضیه ۱-۲-۴ هیچ شبه مسیر $f-$افزایشی درD وجود ندارد . برش$(P, \bar{P})$ را مانند قضیه ۱-۲-۴ در نظر می گیریم. بنابراین

$$f(P, \bar{P}) = c(P, \bar{P}) \quad , \quad f(\bar{P}, P) = ۰$$
$$f(N) = f(P, \bar{P}) - f(\bar{P}, P) = c(P, \bar{P}) - ۰ = c(P, \bar{P})$$

اگر $c(X, \bar{X})$ کمترین برش باشد ، آنگاه بنابر قضیه ۱-۱-۱۰ داریم $f(N) \leqslant c(X, \bar{X})$ بنابراین

$$f(N) \leqslant c(X, \bar{X}) \leqslant c(P, \bar{P}) = f(N)$$
$$\Rightarrow f(N) = c(X, \bar{X})$$

فرض کنیدD گراف جهتدار زمینه در شبکهN باشد D' را به صورت زیر تعریف می کنیم:

$$V(D') = V(D)$$
$$E(D') = \{(x, y) | (x, y) \in E(D), f(x, y) < c(x, y) \text{ یا } (y, x) \in E(D), f(x, y) > ۰\}$$

بنا به این تعریف اگر $(x, y) \in E(D)$ سه حالت داریم:

- اگر $c(x, y) > f(x, y) > ۰$، آنگاه $(x, y) \in E(D')$ و $(y, x) \in E(D')$.
- اگر $c(x, y) = f(x, y) > ۰$، آنگاه $(x, y) \notin E(D')$ و $(y, x) \in E(D')$.
- اگر $c(x, y) > f(x, y) = ۰$، آنگاه $(x, y) \in E(D')$ و $(y, x) \notin E(D')$.

قضیه ۱-۲-۶. فرض کنیدD گراف جهتدار زمینه در شبکهN باشدD' به صورت بالا تعریف شده باشد و نیزf جریان وcتابع ظرفیت باشد. در این صورت گراف D' شامل یک $st-$ مسیر جهتدار است اگر و تنها اگرD

[۱۴] max-Flow- min-Cut

شامل یک شبه مسیر $f-$ افزایشی باشد. بعلاوه طول کوتاهترین $s-t$ مسیر در D' همان طول کوتاهترین شبه مسیر $f-$ افزایشی درD است.

اثبات:فرض کنیم D' شامل $s-t$ مسیر $Q' : u_0, u_1, ..., u_{n-1}, u_n$ باشد. و به ازای هر $0 \leqslant i \leqslant n-1$، (u_{i-1}, u_i) یا (u_i, u_{i-1}) کمانی ازD است. در این صورت $Q : u_0, a_1, u_1, ..., u_{n-1}, a_n, u_n$ یک شبه مسیردرD است.کافی است نشان دهیم Q یک شبه مسیر $f-$ اشباع نشده است. اگر $a_i = (u_{i-1}, u_i)$ آنگاه a_i یک کمان از D' هست و $f(a_i) \leqslant c(a_i)$ و اگر $a_i = (u_i, u_{i-1})$ آنگاه $f(a_i) \geqslant 0$. پس Q یک $st-$ شبه مسیر $f-$ اشباع نشده و بنابراین یک شبه مسیر $f-$ افزایشی است. وبعلاوه Q و Q' طول یکسانی دارند.

برعکس:فرض کنیم D شامل یک شبه مسیر$f-$افزایشی $Q : u_0, a_1, u_1, ..., u_{n-1}, a_n, u_n$ باشد. دراین صورت برای $0 \leqslant i \leqslant n-1$، $a_i = (u_{i-1}, u_i)$ و $f(a_i) \leqslant c(a_i)$ یا $a_i = (u_i, u_{i-1})$ و $f(a_i) \geqslant 0$.

اگر حالت اول برقرار باشد، آنگاه $(u_{i-1}, u_i) \in E(D')$ ودرحالت دوم $(u_i, u_{i-1}) \in E(D')$.

بنابراین $Q' : u_0, u_1, ..., u_{n-1}, u_n$ یک $st-$ مسیر در D' است و Q و Q' طول یکسانی دارند.

تعریف ۱-۲-۷. فرض کنید G یک گراف جهتدار با مجموعه کمانهای E باشد. **مجموعه پیکان**[۱۵]**ها** یک مجموعه ی $E * \{\pm 1\}$ است.به ازای هر کمان $e \in E$ دو "پیکان" جهتدار خلاف جهت هم، یکی در همان جهت e و دیگری در جهت مخالف تعریف می شود. پیکان هم جهت با e را با $\langle e, 1 \rangle$ (گاهی به اختصار با e) وپیکان دیگر را با $\langle e, -1 \rangle$ نمایش می دهیم.

تعریف ۱-۲-۸. تابع $rev(.)$ را که هر پیکان را به پیکان متناظرش در جهت مخالف انتقال می دهد. **تابع معکوس** می نامیم.

$$rev(\langle e, i \rangle) = \langle e, -i \rangle$$

تعریف ۱-۲-۹. سر و دم پیکان d به گونه ای است که جهت پیکان از دم به سر میباشد.پیکان d را میتوان به صورت (i, j) نشان داد که در آن $i = tail(d)$ **دم پیکان** و $j = head(d)$ **سر پیکان** است.

تعریف ۱-۲-۱۰. یک **گشت**[۱۶] **ناتهی** از x به y دنباله ای ناتهی از پیکان های $d_1, d_2, ..., d_k$ است که $x = tail(d_1)$ و $y = head(d_k)$ و به ازای $i = 2, ..., k$ داشته باشیم $tail(d_{i-1}) = head(d_i)$.

یک گشت شامل یک رأس است اگر آن رأس، سر یا دُم یکی از پیکان های تشکیل دهنده آن گشت باشد. رأس آغازین گشت، همان دُم d_1 و راس پایانی همان سر d_k است.رئوس آغازین و پایانی گشتWرا به ترتیب با $start(W)$ و $end(W)$ نشان می دهیم.

---

[۱۵] dart
[۱۶] walk

تعریف ۱-۲-۱۱. گشتی که تنها شامل یک راسvاست یک **گشت تهی** است و $start(W) = end(W) = v$.

تعریف ۱-۲-۱۲. گشتی که در آن هیچ پیکانی بیش از یک بار ظاهر نشود را یک **مسیر**[۱۷] می نامیم .اگر علاوه بر این داشته باشیم ،آنگاه آن را یک **دور**[۱۸] گوییم.

تعریف ۱-۲-۱۳. اگر هر کدام از پیکان های یک مسیر دارای جهت یکسان با کمان متناظرشان باشند، آنگاه آن مسیر را **مسیر جهتدار** می نامیم .

تعریف ۱-۲-۱۴. یک زیردنباله ی متوالی از دنباله ی پیکانها ی گشت W را یک **زیرگشت** از W گوییم.به ازای یک گشت W شامل رئوس u و v، $W[u, v]$ نشان دهنده ی یک زیرگشت از u به v در W است.
و یک زیردنباله ی متوالی از دنباله پیکانهای مسیر W را یک **زیرمسیر**از W گوییم.

تعریف ۱-۲-۱۵. یک زیرمسیر P از W را یک **پیشوند**[۱۹] از W گوییم اگر داشته باشیم $start(W) = start(P)$.
یک زیرمسیر PازW را یک **پسوند**[۲۰]ازWگوییم اگر داشته باشیم $end(W) = end(P)$.

$W[., v]$ اختصاری برای $W[start(W), v]$ و $W[u, .]$ اختصاری برای $W[u, end(W)]$ است.
$W[u, v)$ گشت حاصل از حذف آخرین پیکان $W[u, v]$، $W(u, v]$ و $W(u, v)$ نیز به همین ترتیب تعریف می شوند.

تعریف ۱-۲-۱۶. معکوس گشت $W = d_1, d_2, ..., d_k$ که با $rev(W)$ نشان داده می شود گشت $rev(d_k), rev(d_{k-1}), ..., rev(d_1)$ است.

تعریف ۱-۲-۱۷. R را **زیرمسیر مشترك بیشین** دو مسیر P و Q گوییم هرگاهP و Q زیرمسیر مشترك دیگری شامل R نداشته باشند.

تعریف ۱-۲-۱۸. گراف H را یک **زیرگراف** از گراف G گوییم اگر مجموعه ی رئوس H زیرمجموعه ای از مجموعه رئوس G باشد و همچنین هر کمان (u, v) از H کمانی از G باشد.

تعریف ۱-۲-۱۹. یک گراف را **همبند**[۲۱] گوییم اگر به ازای هر جفت رأس u و v شامل یک مسیر بدون جهت بین u و v باشد.

---
[۱۷] path
[۱۸] cycle
[۱۹] prefix
[۲۰] suffix
[۲۱] connected

تعریف ۱-۲-۲۰. یک گراف همبند را **درخت** گوییم .اگر هیچ دوری نداشته باشد.

تعریف ۱-۲-۲۱. در یک گراف همبند G راس v را **رأس برشی** گوییم هرگاه گراف حاصل از حذف v و یالهای مجاورش همبند نباشد. همچنین یک رأس را **غیر برشی** گوییم اگر یک رأس برشی نباشد.

لم ۱-۲-۲۲. هر گراف همبند حداقل یک رأس غیر برشی دارد.
اثبات:اگر گراف دارای دور باشد هر رأس آن دور یک رأس غیر برشی است و اگر دور نداشته باشد پس یک درخت است و هر برگ آن یک رأس غیر برشی است.

تعریف ۱-۲-۲۳. زیرگرافی از گراف G که شامل همه رئوسG باشد و یک درخت باشد را یک **درخت پوشا**[۲۲] **ی** گراف G گوییم.

فرض کنید T یک درخت پوشا از گراف G باشد. چون T همبند است و هیچ دوری نیز ندارد، بین هر دو رأس u و v از G در T یک مسیر یکتا وجود دارد. آن مسیر را با $T[u,v]$ نمایش می دهیم.همچنین اگر رأسی از T را به عنوان ریشه در نظر بگیریم، $T[u]$ مسیر u به ریشه را نشان میدهد.فرض کنید $\{i,j\}$ یک یال از T باشد،به صورتی که $T[i]$ شامل j باشد.در این صورت، پیکانی متناظر با این یال که دارای جهت از i به j است را پیکان پدر[۲۳] رأس i در T تعریف می کنیم.

تعریف ۱-۲-۲۴. زیر گراف H از گراف G را یک **مؤلفه**[۲۴] از G گوییم .هرگاه H همبند باشد و زیرگراف همبندی از G به غیر از H وجود نداشته باشد که H زیرگراف آن باشد.

تعریف ۱-۲-۲۵. یک برش از یک گراف همبند را یک **برش ساده**[۲۵] گوییم اگر گراف حاصل از حذف کمان های متناظر با آن برش دقیقاً دو مؤلفه داشته باشد.

تعریف ۱-۲-۲۶. در یک برش $[P,\bar{P}]$ یک کمان (i,j) را یک **کمان پیش رو** [۲۶] گوییم اگر داشته باشیم : $i \in P, j \in \bar{P}$.

تعریف ۱-۲-۲۷. یک مجموعه را **مجموعه چندگانه** [۲۷] گوییم ، اگر اعضای آن بتوانند بیش از یک بار ظاهر شوند.

---

[۲۲] spanning tree
[۲۳] parent dart
[۲۴] component
[۲۵] simple
[۲۶] forward arc
[۲۷] multi-set

۱-۳ فضای برداری کمانها

فضای کمانی گراف $G(V,E)$ فضای برداری $R^E = \{f|f : E \to R\}$ است.

بردار δ در فضای کمانی[۲۸] ، به هر کمان $e \in E$ یک عدد حقیقی $\delta[e]$ نسبت می دهد.به ازای یک پیکان $\langle e, i\rangle$ $(i = \pm ۱)$ داریم:

$$\delta[\langle e, i\rangle] = i\delta[e]$$

یعنی اگر d پیکانی در همان جهت کمان e متناظرش باشد، آنگاه $\delta[d] = \delta[e]$ و اگر dدر جهت مخالف باشد، آنگاه $\delta[d] = -\delta[e]$ بردار δ_e در فضای کمانی به صورت زیر تعریف می شود.

$$\delta_e[e'] = \begin{cases} ۱ & e = e' \\ \\ ۰ & \text{در غیر این صورت} \end{cases} \qquad (۵-۱)$$

به ازای یک مجموعه ی چند گانهP از پیکانها تعریف می کنیم:

$$\delta_P = \sum_{d \in P} \delta_d$$

تعریف ۱-۳-۱. فضای دوری[۲۹] گرا ف G زیرفضایی از فضای کمانی است که به صورت زیر تولید می شود،

$$\varsigma = \{\delta_C| \text{ یک دور از گراف } G \text{ است } C\}$$

بردارهای فضای دوری را **گردش** [۳۰] می نامیم.

۱-۴ گراف های مسطح

یک رسم از یک گراف روی یک صفحه، رئوس گراف را به نقاط مجزا و هر یال آن را به یک خم ساده تصویر می کند به طوری که دو انتهای خم متناظر به هر یال، به رئوس انتهایی آن یال متناظر شده باشد.

تعریف ۱-۴-۱. در یک رسم، نقطه ای که دو خم متناظر به یالها یکدیگر را قطع کنند و آن نقطه متناظر به رأس انتهایی مشترك بین آن دو یال نباشد را **تقاطع** [۳۱] می نامیم .

[۲۸] arc space
[۲۹] cycle space
[۳۰] circulation
[۳۱] crossing

تعریف ۱-۴-۲. یک رسم بدون تقاطع روی یک صفحه را **مسطح**[۳۲] گویند.

تعریف ۱-۴-۳. هر رسم بدون تقاطع روی یک صفحه، آن صفحه را به ناحیه هایی تقسیم می کند که هر کدام از آن نواحی یک **وجه** از گراف نامیده می شوند.

تعریف ۱-۴-۴. در یک رسم مسطح، وجه نامتناهی را **وجه بی نهایت**[۳۳] و دیگر وجه ها را **وجه های داخلی**[۳۴] گویند. وجه بی نهایت را با f_∞ نشان می دهیم.

تعریف ۱-۴-۵. گرافی که دارای یک رسم مسطح باشد را **گراف مسطح** می نامند.

تعریف ۱-۴-۶. مجموعه ی همه کمانهایی که وجه fرا در برمی گیرند ، **مرز**[۳۵] **وجه** f نامیده می شود. مرز یک وجه را می توان توسط یک دور نشان داد. دقت کنید که هر کمان گراف، حداکثر به مرز دو وجه تعلق دارد.
مرز وجه f را با ∂f نشان می دهیم.
به طور قرار دادی جهت مثبت آن را جهت پاد ساعتگرد در نظر می گیریم.

تعریف ۱-۴-۷. مرز وجه بی نهایت گراف مسطح G را **مرز گراف مسطح** Gمی نامیم وآن را با ∂G نشان می دهیم.

طبق قرارداد، به ازای یک وجه f غیر از وجه بی نهایت ، ∂f دارای جهت پادساعتگرد و∂f_∞ دارای جهت ساعتگرد است.

تعریف ۱-۴-۸. ، وجه های f و $\bar{f}$ را **مجاور**[۳۶] گوییم ، اگرمرزها یشان شامل حداقل یک کمان مشترك باشد.
توجه کنید که اگر دو وجه تنها در یک رأس مشترك باشند، آنگاه آنها مجاورنیستند.

تعریف ۱-۴-۹. دو رأس s,t از یک گراف مسطح Gرا در نظر بگیرید. G را $st-$**مسطح** گوییم اگر رئوس s و t هر دو بر مرز وجه بی نهایت واقع شده باشند.

گزاره ۱-۴-۱۰. (فرمول اویلر) اگر یک گراف مسطح همبند دارای n راس و m کمان و f وجه باشد، آنگاه $f = m - n + ۲$.

اثبات به مرجع [۱] مراجعه کنید.

گزاره ۱-۴-۱۱. در یک گراف ساده و مسطح با n راس و m کمان داریم: $m < ۳n$.

اثبات به مرجع [۱] مراجعه کنید.

---

[۳۲] planar
[۳۳] infinite face
[۳۴] internal face
[۳۵] boundary
[۳۶] adjacent

۱-۴-۱ دوگان گراف های مسطح

دوگان [۳۷] گراف G را با G^* نمایش می دهیم و به صورت زیر تعریف می کنیم:

ابتدا یک رأس f^* درون هر وجه f از G قرار می دهیم. هر کمان G یک کمان متناظر در G^* دارد که آن را دوگان آن کمان می نامیم. به ازای هر کمان (i,j) از G متعلق به مرز دو وجه، مثلاً $f_۱, f_۲$ گراف G^* دارای کمان $(f_۱^*, f_۲^*)$ است. جهت این کمان دوگان به گونه ای است که اگر در محل کمان (i,j) ناظری را در نظر بگیریم که دید آن رو به جهت این کمان باشد، آنگاه دُم کمان دوگان در وجه سمت چپ و سر کمان دوگان در وجه سمت راست خواهد بود.

توجه کنید که اگر کمان (i,j) از G متعلق به مرز تنها یک وجه، مثلاً $f_۱$ باشد، آنگاه در گراف دوگان G^* یک حلقه ی $(f_۱^*, f_۱^*)$ خواهیم داشت. همچنین اگر دو وجه در بیش از یک کمان مجاور هم باشند، آنگاه گراف دوگان دارای کمانهای موازی خواهد بود. تعداد رئوس در گراف دوگان برابر تعداد وجه ها در گراف اولیه است و برعکس، تعداد وجه ها در گراف دوگان برابر تعداد رئوس در گراف اولیه است. هر دو گراف اولیه و دوگان تعداد کمان یکسانی دارند. علاوه بر این، دوگان گراف دوگان، همان گراف اولیه است.

یک دور در گراف دوگان، معرف یک برش در گراف اولیه است و بر عکس.

فرض کنید G یک گراف مسطح وهمبند باشد، دراین صورت مجموعه ی بردارهای $\{\delta_{\partial f} | f \neq f_\infty\}$ یک پایه برای فضای دوری G است و در نتیجه هر بردار $\eta \in \varsigma$ را می توان به عنوان یک ترکیب خطی از این بردارهای پایه نوشت .

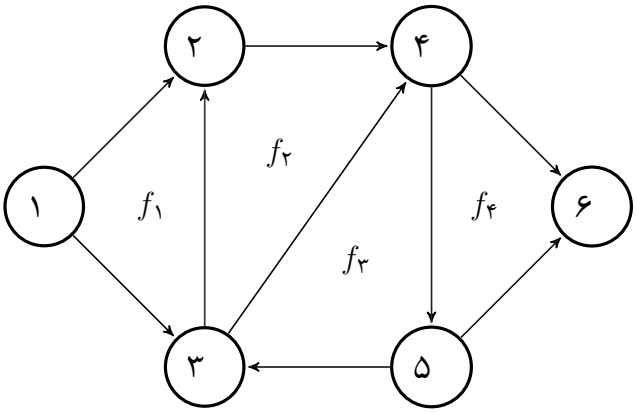
$$\eta = \sum_{f \neq f_\infty} \phi[f]\delta_{\partial f}$$

ϕ را یک انتساب پتانسیل [۳۸] و ∂f را پتانسیل وجه f می گوییم. طبق قرارداد داریم: $\phi[f_\infty] = ۰$.

مثال ۱-۴-۱۲. در شکل زیر گردش ساعت گرد متناظر دور $C_۱ = (۲ - ۴ - ۵ - ۳ - ۲)$ را می توان به صورت ترکیب خطی زیر نوشت:

$$\delta_{C_۱} = (۰)\delta_{\partial f_۱} + (-۱)\delta_{\partial f_۲} + (-۱)\delta_{\partial f_۳} + (۰)\delta_{\partial f_۴}$$

[۳۷] dual
[۳۸] potential



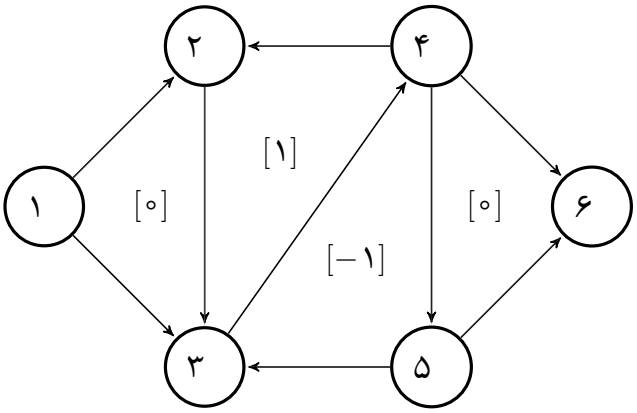
تعریف ۱-۴-۱۳. دور ساده ی C از G، یک پیکان، رأس یا وجه را اکیداً در برمی گیرد، اگر آن پیکان، رأس یا وجه با در نظر گرفتن f_∞ به عنوان وجه خارجی، داخل C واقع شده باشد.
فرض کنید C یک دور باشد (نه لزوماً ساده) و ϕ پتانسیل متناظر با گردش δ_C باشد. در این صورت:

- دور C یک وجه f را در بر می گیرد، اگر $\phi[f] \neq ۰$
- دور C یک پیکان d را اکیداً در بر می گیرد، اگر C وجه های سمت چپ و راست d (به ترتیب $tail_{G^*}(d)$ و $head_{G^*}(d)$) را در بر بگیرد.
- دور C یک پیکان d را در بر میگیرد اگر C، d را اکیداً در بر بگیرد و $d \in C$ یا $rev(d) \in C$.
- دور C یک راس v را اکیداً در بر می گیرد، اگر C تمام پیکانهای مجاور v را اکیداً در بر بگیرد.
- دور C یک راس v را در بر می گیرد، اگر C تمام پیکانهای مجاور v را در بر بگیرد.

تعریف ۱-۴-۱۴. یک گردش را پادساعت گرد [۳۹] گوییم اگر پتانسیل هر وجه آن نامنفی باشد و ساعت گرد [۴۰] گوییم اگر پتانسیل هر وجه آن نامثبت باشد. یک دور جهت دارC، از پیکان ساعت گرد است اگر δ_C ساعت گرد باشد.

مثال ۱-۴-۱۵. در شکل زیر دور ۳ − ۲ − ۱ نه ساعت گرد است و نه پادساعت گرد پتانسیل آن صفر است. دور ۴ − ۳ − ۲ یک دور پادساعتگرد با پتانسیل ۱و دور ۵ − ۴ − ۳ یک دور ساعتگرد با پتانسیل ۱ −است.

[۳۹] counterclockwise
[۴۰] clockwise



تعریف ۱-۴-۱۶. یک جریان x را یک **چپ ترین جریان** گوییم اگر گراف باقیمانده نسبت به جریان x هیچ دور باقیمانده ی ساعت گردی نداشته باشد.

لم ۱-۴-۱۷. فرض کنید x یک چپ ترین جریان و P چپ ترین مسیر افزایشی باشد. در این صورت جریان x' که از اشباع P حاصل می شود، یک چپ ترین جریان است.

اثبات: (فرض خلف) فرض کنید که x' یک چپ ترین جریان نباشد. در این صورت طبق تعریف ۱-۴-۱۶ باید یک دور باقیمانده ی ساعت گرد C برحسب x' وجود داشته باشد. بدون این که خللی به کلیت مسئله وارد شود فرض کنید که C ساده باشد. از آنجا که x چپ ترین بوده، C پیش از افزایش، باقیمانده نبوده و بنابراین P باید یک پیکان d مشترك با $rev(C)$ داشته باشد. فرض کنید $P' = P[.,tail(d)]oC[tail(d);tail(d)]oP[tail(d),.]$ که C دور ساعت گرد با شروع از $tail(d)$ است و P' یک گشت است و $\delta_{P'} - \delta_P = \delta_P + \delta_C - \delta_P = \delta_C$ یک گردش ساعت گرد است، یعنی P' در سمت چپ P است. بنابراین P چپ ترین مسیر افزایشی نبوده است.

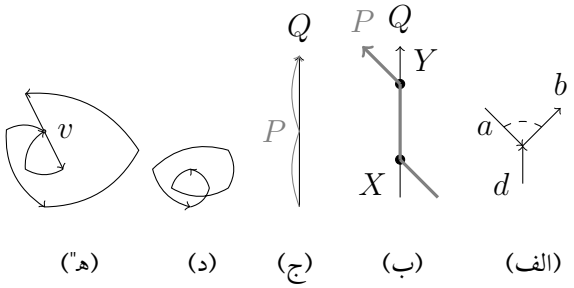
تعریف ۱-۴-۱۸. اگر a و b و d پیکانهایی باشند که $head(a) = tail(b) = head(d)$ آنگاه گوییم d در $head(a)$ وارد aob می شود. اگر علاوه بر این d در دنباله پیکان ها به ترتیب پادساعت گرد بین b و $rev(a)$ ظاهر نشود، آنگاه گوییم d در $head(a)$ از سمت راست وارد aob می شود.(شکل ۱-۱(الف)) ورود از سمت چپ، خروج از سمت راست و مانند آن نیز به همین صورت تعریف می شود.

تعریف ۱-۴-۱۹. مسیر P از مسیر Q عبور می کند، هرگاه زیرمسیر مشترك بیشین R از P و Q وجود داشته باشد، به گونه ای که P در $start(R)$ از سمت راست وارد Q و در $end(R)$ از سمت چپ از Q خارج شود، یا P در $start(R)$ از سمت چپ وارد Q و در $end(R)$ از سمت راست از Q خارج شود.(شکل ۱-۱(ب))

تعریف ۱-۴-۲۰. مسیرهایی که از یکدیگر عبور نمی کنند را **نامتقاطع** می نامیم .(شکل ۱-۱(ج))

تعریف ۱-۴-۲۱. یک مسیر یا دور را **ناخودمتقاطع** [۴۱] می نامیم ، هرگاه به ازای هر جفت PوQ از زیرمسیرهای آن، P از Q عبور نکند.(شکل ۱-۱(ه"))

[۴۱] non-self-crossing



شکل ۱-۱: ورود و خروج و تقاطع مسیرها [۱]

به ازای هر وجه f مرز f یک دور ناخودمتقاطع است.

شکل ۱-۱ را در نظر بگیرید.در شکل ۱-۱(الف) پیکان d از سمت راست وارد مسیر aob می شود.در شکل ۱-۱(ب) مسیر P از مسیر Q عبور می کند،P در نقطه x از سمت راست واردQ می شود و در نقطه y از سمت چپ از Q خارج می شود.در شکل ۱-۱(ج) مسیر های P و Q نامتقاطع اند.شکل ۱-۱(د) یک دور خودمتقاطع را نشان می دهد و شکل ۱-۱(ه") یک دور ناخودمتقاطع ولی غیرساده است زیرا رأس v دو بار ظاهر می شود.

لم ۱-۴-۲۲. (ترکیب) فرض کنید C یک دور ناخودمتقاطع باشد و A یک مسیر ناخودمتقاطع با نقاط پایانی واقع بر C باشد که هیچ بخشی از Aتوسط Cدر بر گرفته نشده باشد. در این صورت

$$AoC[end(A), start(A)]$$

یک دور ناخودمتقاطع است.

اثبات : چون هیچ بخشی از A توسط Cدر بر گرفته نشده AازCعبور نمی کند و بنابراین دور $AoC[end(A), start(A)]$ ناخودمتقاطع است.

تعریف ۱-۴-۲۳. اگر $\delta_A - \delta_B$ یک گردش ساعت گرد باشد، آنگاه گشت A از x به y در سمت چپ گشت B از x به y است. به همین ترتیب A در سمت راست B است اگر $\delta_A - \delta_B$ یک گردش پاد ساعت گرد باشد.

روابط چپ بودن و راست بودن ترتیب جزئی هستند؛ یعنی دارای خاصیت تعدی، بازتابی و پادتقارن می باشند.

تعریف ۱-۴-۲۴. یک گشت A از x به y **چپ ترین گشت** از x به y است.اگر به ازای هر گشت B از x به y ، A سمت چپ B باشد.

یک چپ ترین گشت لزوماً همیشه وجود ندارد؛ فرض کنید $Q = PoC$ چپ ترین گشت باشد که Q یک مسیر از x به y و C یک دور ساعت گرد باشد. در این صورت $R = PoC$ یک گشت است که سمت چپ P است و RoC سمت چپ R و الی آخر. ولی لم زیر به ما اجازه می دهد که هنگام بررسی گراف های بدون دور ساعت گرد، تنها مسیرهای ساده را بررسی کنیم.

لم ۱-۴-۲۵. فرض کنید G یک گراف بدون دور ساعت گرد باشد. اگر P یک چپ ترین گشت باشد، آنگاه P یک مسیر ساده است.

اثبات: (فرض خلف) فرض کنید که P یک مسیر ساده نباشد و فرض کنید x رأسی باشد که حداقل دو بار در P ظاهر می شود. x_1 اولین بار و x_2 آخرین باری باشد که x در P ظاهر می شود. در این صورت $C = P[x_1, x_2]$ یک دور ساده است. بنا به فرض این دور پادساعت گرد است. $P' = P[., x_1]oP[x_2, .]$ مسیری است که از حذف C از P به دست می آید. گردشِ $\delta_P - \delta'_P = \delta_C$ پادساعت گرد است، پس P کاملاً در سمت راست P' است که این با چپ ترین گشت بودن P تناقض دارد.

لم ۱-۴-۲۶. هر زیرمسیر از یک چپ ترین مسیر، یک چپ ترین مسیر است.

اثبات. فرض کنید P یک چپ ترین مسیر باشد و Q یک زیر مسیر x به y از P باشد. فرض کنید یک مسیر x به y دیگر $Q' \neq Q$ وجود داشته باشد که سمت چپ Q باشد و $P' = P[., x]oQ'oP[y, .]$. در این صورت گردشِ

$$\delta'_P - \delta_P = \delta_{P[.,x]oQ'oP[y,.]} - \delta_{P[.,x]oQoP[y,.]} = \delta'_Q - \delta_Q$$

ساعت گرد است، زیرا Q' سمت چپ Q است. بنابراین $P' \neq P$ سمت چپ P است که این یک تناقض است.

قضیه ۱-۴-۲۷. (قضیه عدم تقاطع) اگر P و Q مسیر های x به y ناخودمتقاطع مجزایی باشند که از یکدیگر عبور نمی کنند، آنگاه P یا سمت راست و یا سمت چپ Q است.

اثبات به مرجع [۳] و [۸] مراجعه شود.

۱-۴-۲ جستجوی اول-راست

یک جستجوی اول-راست [۴۲] در یک گراف مسطح بدون جهت، یک جستجوی اول-عمق است که در آن در هر گام،همه ی گزینه های ممکن برای پیش روی از رأس کنونی، در ترتیب چرخشی حول یک رأس، به ترتیبِ از راست به چپ در نظر گرفته می شوند. به بیان دقیقتر به این صورت عمل می شود :

فرض کنید در یک گام از جستجو، رأس v رأس کنونی از مسیر جستجو است و (w, v) یال کنونی (یال جلویی)

[۴۲] right-first search

است. اگر یال دیگری مجاورv وجود داشته باشد که هنوز در آن مرحله در نظر گرفته نشده است، از میان چنین یالهایی یال بعدی (طبق ترتیب پادساعت گرد)پس از (w,v) در لیست همسایگی مرتب v برای پیش روی انتخاب می شود. جستجوی اول-راست در یک گراف جهتدار نیز در اصل کاملاً مشابه است، با این تفاوت که در اینجا تنها کمانهایی که از رأس کنونی خارج می شوند، گزینه های ممکن برای پیش روی هستند.
جستجوی اول-راست را می توان به صورت پس رو نیز انجام داد. یعنی کمانهایی که به رأس کنونی وارد می شوند، گزینه های ممکن برای پیش روی هستند.

قضیه ۱-۴-۲۸. (دوگانگی دور، برش) در یک گراف مسطح همبند، یک مجموعه از کمانها تشکیل یک دور جهتدار و ساده در گراف اولیه می دهند اگر و تنها اگر کمان های متناظرشان در گراف دوگان تشکیل یک برش جهتدار و ساده در گراف دوگان را بدهند.
اثبات:فرض کنیدC یک دور جهتدار و ساده در گرافG باشد. در این صورت، تصویرC صفحه را به دو مولفه همبند تقسیم می کند. یعنی وجه هایG به دومولفه همبند تقسیم می شوند. از آنجا که هر دو وجه موجود در یک مولفه ی یکسان را می توان با یک کمان در صفحه به هم وصل کرد، بدون این که این کمان، پیکان های دورC را قطع کند. بنابراین C^* دوگان پیکانهای C یک برش ساده برای G^* است.
برعکس : فرض کنید C^* یک برش ساده در G^* باشد. در این صورت،C^* یک افراز از رئوس G^* (یا به طور معادل، یک افراز از وجه های G) به دو مؤلفه ی همبند است. پیکان هایی ازG که روی مرز این دو مؤلفه قرار دارند ،دقیقاً همان پیکان های C هستند. بنابراین C یک دور در Gاست.

یکی دیگر از ویژگی های مهم گراف دوگان، مربوط به درخت پوشای آن است. فرض کنید $\{T,\bar{T}\}$ یک افراز از E باشد، در این صورت T یک درخت پوشا برای G است اگر وتنها اگر $\bar{T}^*$ (یعنی دوگان یال هایی که درT نیستند.)یک درخت پوشا برای G^* باشد.

قضیه ۱-۴-۲۹. (درخت های پوشای جفت) به ازای یک درخت پوشای T از گراف ساده، مسطح و همبند Gمجموعه ی کمان های متناظر با کمان ها یی که درT نیستند، تشکیل یک درخت پوشا برای گراف دوگان G^* را می دهند.
اثبات:فرض کنیم T مجموعه ی یال های یک درخت پوشا برای گراف Gباشد.مجموعه ی دوگان یال هایی که درT نیستند را $\bar{T}^*$ می نامیم.ابتدا نشان می دهیم چونT دور ندارد، $\bar{T}^*$ همبند است .فرض کنید همبند نباشد. مؤلفه های همبند آن را در نظر بگیرید. گرافHرا با توجه به این مؤلفه ها ی همبند از روی G^* به این صورت می سازیم: در گرافHبه جای هر مولفه یک راس قرار می دهیم و اگر در G^* بین رأسی از مؤلفه ی متناظر با i و

رأسی از مؤلفه ی متناظر با j یالی وجود داشته باشد، دو راس i و j در H را با یک یال به هم وصل می کنیم.
واضح است کهHنیز همبند خواهد بود، در نتیجه طبق لم ۱-۲-۲۲ حداقل یک رأس غیر برشی دارد. فرض کنید v یک رأس غیر برشی از H باشد. مجموعه ی رئوس مولفه متناظر با راس v در گراف G^* را با P نشان می دهیم.
با توجه به غیر برشی بودن رأس v برش $[P, \bar{P}]$ یک برش ساده در G^* است.طبق قضیه ی ۱-۴-۲۸ متناظر با چنین برشی یک دور ساده یCدر Gوجود دارد.هر کمان از برش $[P, \bar{P}]$ کمانی از $\bar{T}^*$ نیست، پسC یک دور درTاست و این با درخت بودن$\boldsymbol{T}$ در تناقض است، پس $\bar{T}^*$ همبند است.
فرض کنید $\bar{T}^*$ شامل یک دور مانند$\boldsymbol{C}$ باشد.چون Tدرخت پوشا است و یالهای$\boldsymbol{T}$ و $\bar{T}^*$ یکدیگر را قطع نمی کنند. دورCرئوسی ازGکه در داخل آن واقع می شوند را از رئوسی که بیرون آن قرار دارند جدا می کند و این با همبند بودنTتناقض دارد.پس $\bar{T}^*$ دور ندارد و بنابراین یک درخت پوشا برای G^* است.

فصل ۲

روشهای حل مسئله برش کمینه مبتنی بر جریان

تاریخچه ی مسئله ی برش کمینه [۸] ، [۱] به شدت به گراف های مسطح گره خورده است. در بحبوحه ی جنگ سرد، کشور ایالات متحده امریکا تلاش قابل ملاحظه ای را صرف تجزیه وتحلیل شبکه ی ریلی شوروی سابق کرد: "موفقیت در بازدارندگی تا حدود زیادی بستگی به اقدامات بازدارنده در قابلیت جابجایی افراد و مهمات دشمن دارد".هریس و راس [۱] [۹] شبکه ی ریلی شوروی را به صورت یک گراف مسطح مدلسازی کردند؛ آنها برای این منظور، دوگان گراف مسطح تشکیل شده از مرزهای بخشهای کلیدی را با یال های بیان کننده ی ظرفیت انتقال بین این بخش ها در نظر گرفتند. آنها سپس مسئله ی تعیین بهترین راه برای ایجاد اختلال در شبکه ی ریلی شوروی را مطالعه کردند، یعنی کمترین تعداد خطوط آهنی که باید قطع می شدند تا جلوی جابجایی مهمات و افراد بین مکانهای حساس تاکتیکی گرفته شود را به دست آوردند: آنهایک برش کمینه در گراف یافتند. فورد و فولکرسون [۲] [۱۰] این تحقیقات را ادامه دادند که منجربه مقاله ی برجسته ای شد که ضمن **اثبات** (قضیه ی "جریان بیشینه - برش کمینه" ۱-۲-۵) الگوریتم مسیر افزایشی را نیز بیان می کرد. برخی از تحقیقات مرتبط با این موضوع در نیروی هوایی آمریکا تا سال ۱۹۹۹ محرمانه بودند.

الگوریتم فورد و فولکرسون در روند **اثبات** قضیه ۱-۲-۴ بیان شده است. ادموندز و کارپ [۳] [۲] نشان دادند که در صورتی که در هر تکرار، کوتاهترین مسیر افزایشی انتخاب شود، حداکثر nm تکرار وجود دارد و اجرای هر تکرار از مرتبه $O(m)$ است.آنها الگوریتمی از مرتبه $O(nm^2)$ ارائه دادند.که m تعداد کمانها و n تعداد راسها است.

همجنین فورد و فولکرسون در [۱۱] یک نسخه ی ویژه از الگوریتم مسیر افزایشی برای پیدا کردن جریان بیشینه در یک گراف $st-$مسطح ارائه کردند. این الگوریتم در گرافی که مبدأ آن در سمت چپ و مقصد آن در سمت راست

[۱] Harris and Ross
[۲] Ford and Fulkerson
[۳] Edmonds and Karp

رسم شده، به صورت تکراری، بالاترین مسیر [۴] افزایشی را افزایش می دهد.این الگوریتم دارای این ویژگی است که جریان روی کمان هیچ گاه کاهش نمی یابد.چون هر افزایش، حداقل یک کمان را ناباقیمانده می کند ،الگوریتم حداکثر به m افزایش نیاز دارد که m تعداد کمانها است.ایتای و شیلاچ [۵] در [۱۲] نشان دادند که هر تکرار الگوریتم بالاترین مسیر را با استفاده از یک صف اولویت از پیکان های باقیمانده می توان در زمان $O(logn)$ پیاده سازی کرد که n تعداد راسها است.در نتیجه الگوریتم در زمان $O(nlogn)$ قابل اجرا است.(با توجه به ۱-۴-۱۱ یک گراف مسطح ساده با n رأس، حداکثر $3n$ کمان دارد.) در سال ۲۰۰۸ ، بورَدایل و کلین [۶] [۳] اولین الگوریتم درست از مرتبه ی $O(nlogn)$ را برای یافتن جریان بیشینه در یک گراف مسطح جهت دار ارائه کردند،که s و t لزوما در یک وجه نیستند . این الگوریتم که "چپ ترین مسیر [۷] " نام دارد،سریع ترین و کامل ترین الگوریتم شناخته شده برای مسئله ی جریان در یک گراف مسطح جهت دار است.

بنا برقضیه ۱-۲-۵ با به دست آوردن جریان بیشینه برای یک گراف مقدار برش کمینه نیز به دست می آید. روشهای به دست آوردن جریان بیشینه در گرافهای عمومی (مسطح یا نا مسطح) به دو دسته کلی تقسیم می شوند،

- الگوریتم های مسیر افزایشی
- الگوریتم های ارسال پیش جریان

روشهای بیان شده در این فصل ازنوع الگوریتم های مسیر افزایشی می باشند. همان گونه که در بالا گفته شد، اولین الگوریتم مسیر افزایشی توسط فورد و فولکرسون ارائه شد و همان گونه که در قضیه ۱-۲-۴ بیان شد، این الگوریتم به صورت تکراری عمل می کند:یک مسیر P از مبدأ به مقصد پیدا کرده و جریان را در امتداد این مسیر ارسال می کند. به عبارت دیگر، به ازای هر کمان در P میزان جریان به اندازه Δ که این اندازه نمی تواند از ظرفیت باقیمانده ی هر کمان P بیشتر باشد، افزایش می دهد.

در الگوریتم های ارسال پیش جریان به جای ارسال جریان در امتداد مسیرها، جریان روی کمان های خاصی ارسال می شود.این الگوریتم در طی زمان اجرای خود یک جریان شدنی را حفظ نمی کند؛ عمل افزایش را روی کمان ها انجام می دهد تا جریان را به شدنی بودن نزدیک کند. برای جزئیات بیشتر پیرامون این دسته از الگوریتم ها به فصل ۷ از [۱۳] مراجعه کنید.نمونه ای از این الگوریتم ها الگوریتم [۱۴] است.

در این فصل ابتدا نشان می دهیم اگر قانونی برای انتخاب مسیر نداشته باشیم ممکن است تکرار الگوریتم های مسیر افزایشی به ∞ میل کند. پس از آن یک الگوریتم مسیر افزایشی منسوب به ادموندز- کارپ [۲] وسپس الگوریتم چپ ترین مسیر منسوب به بورَدایل- کلین [۳] را با یک مثال شرح می دهیم. تفاوت اصلی این الگوریتم ها در نحوه انتخاب شبه مسیر افزایشی است که باعث شده مرتبه زمانی آنها متفاوت باشد.

---

[۴] uppermost path
[۵] Itai and Shiloach
[۶] Borradaile and Klein
[۷] leftmost path

الگوریتم ۲-۱ الگوریتم فورد و فولکرسون [۲]

ورودی: f (جریان اولیه که می تواند صفر باشد)، c (تابع ظرفیت)،s,t (گره های مبدا ومقصد) و D(گراف زمینه جهتدار)

خروجی: ماکزیمم جریان ، برش کمینه و مجموعه های $S,(V/S)$

۱: گراف جهتدار D' را به صورت زیر بسازید

$$V(D') = V(D)$$
$$E(D') = \{(x,y)|(x,y) \in E(D), f(x,y) < c(x,y) \text{ یا } (y,x) \in E(D), f(x,y) > ۰\}$$

۲: تعیین کنید که آیا D' شامل$st-$مسیر است یا نه. اگر D' شامل $st-$ مسیر نیست، آنگاه به مرحله ۵ بروید در غیر این صورت $Q' = u_۰, u_۱, ..., u_{n-۱}, u_n$ یک $st-$ مسیردر D' است و ادامه دهید.

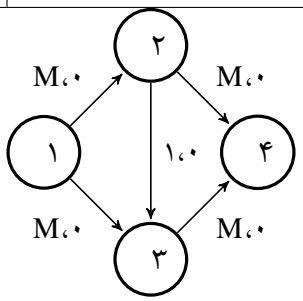
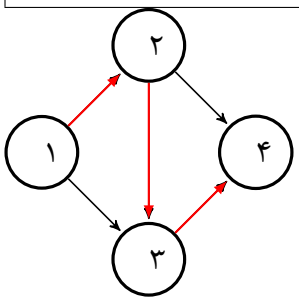
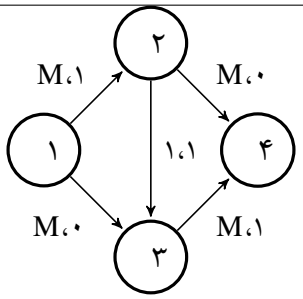
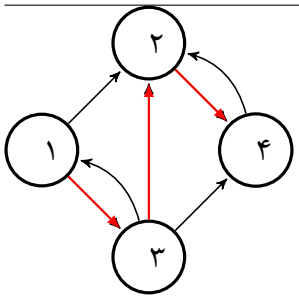
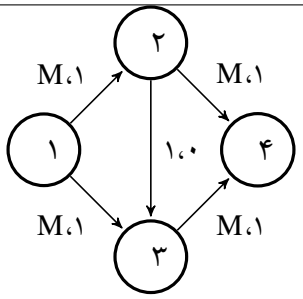
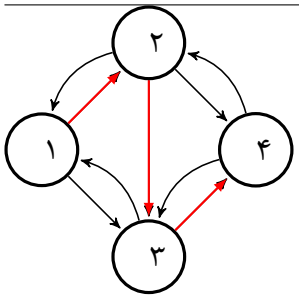
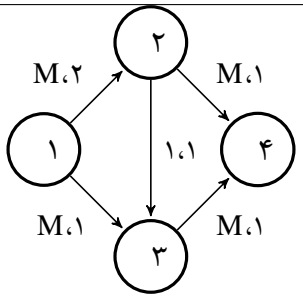
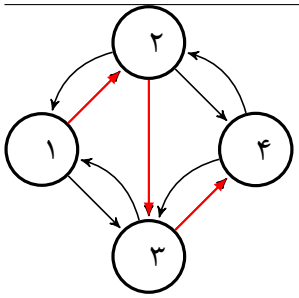
۳: فرض کنیم $Q = u_۰, a_۱, u_۱, ..., u_{n-۱}, a_n, u_n$ که a_iکمانD است.
اگر $a_i = (u_{i-۱}, u_i)$و $f(a_i) \leqslant c(a_i)$، آنگاه قرار دهید $\Delta_i = c(a_i) - f(a_i)$ واگر $f(a_i) \geq ۰$ و $a_i = (u_{i-۱}, u_i)$، آنگاه قرار دهید $\Delta_i = f(a_i)$ و نیز قرار دهید $\Delta = \min\{\Delta_i | ۱ \leqslant i \leqslant n\}$. اگر حالت اول برقرار است ،آنگاه $f(a_i) = f(a_i) + \Delta$ ودر حالت دوم $f(a_i) = f(a_i) - \Delta$.

۴: به مرحله۱ بروید.

۵: Dبرای تمام کمانهای $f(a)$ را به عنوان خروجی تعیین کنید. فرض کنید P مجموعه ای از رأسهای x از D' باشد که یک $s-x$ مسیر در D' وجود دارد. در این صورت (P,P') یک کمترین برش است.

همان گونه که در جدول ۲-۱ نشان داده شده است. شبه مسیر انتخابی در مراحل زوج برابر با $۴-۲-۳-۱$ و شبه مسیر انتخابی در مراحل فرد برابر با $۴-۳-۲-۱$ بوده و M۲ تکرار لازم است تا این مسیر ها اشباع شوند و با میل کردن M به ∞ تعداد تکرار ها نیز به ∞ میل می کند.

جدول ۲-۱: الگوریتم مسیر افزایشی

	گراف D	گراف D' و $st-$ مسیر
تکرار ۱		
تکرار ۲		
تکرار ۳		
تکرار ۴		

۲-۱ الگوریتم ادموندز-کارپ

الگوریتم ادموندز -کارپ یک الگوریتم مسیر افزایشی است که در هر مرحله جریان را در امتداد یک کوتاه ترین مسیر از مبدأ به مقصد ارسال می کند.

در الگوریتم ادموندز -کارپ منظور از کوتاهترین شبه مسیر ،شبه مسیر با کمترین تعداد یال است.

الگوریتم ۲-۲ الگوریتم ادموندز- کارپ [۲]

ورودی: f (جریان اولیه که می تواند صفر باشد)، c (تابع ظرفیت)،s,t (گره های مبدا ومقصد) و D(گراف زمینه جهتدار)

خروجی: ماکزیمم جریان ، برش کمینه و مجموعه های $S,(V/S)$

۱: گراف جهتدار D' را به صورت زیر بسازید

$$V(D') = V(D)$$
$$E(D') = \{(x,y)|(x,y) \in E(D), f(x,y) < c(x,y)\ or\ (y,x) \in E(D), f(x,y) > ۰\}$$

۲: الگوریتم مور [۸] را برای D' به کار ببرید تا تعیین کنید که آیا D' شامل کوتاهترین $st-$ مسیر است یا نه. اگر D' شامل $st-$ مسیر نیست، به مرحله ۵ بروید در غیر این صورت $Q' = u_۰, u_۱, ..., u_{n-۱}, u_n$ یک کوتاهترین$st-$مسیردرD' است و ادامه دهید.

۳: فرض کنیم $Q = u_۰, a_۱, u_۱, ..., u_{n-۱}, a_n, u_n$ که a_i کمان D است.
اگر $a_i = (u_{i-۱}, u_i)$و $f(a_i) \leqslant c(a_i)$، قرار دهید $\Delta_i = c(a_i) - f(a_i)$ واگر $f(a_i) \geq ۰$ و $a_i = (u_{i-۱}, u_i)$، قرار دهید $\Delta_i = f(a_i)$ و نیز قرار دهید $\Delta = \min\{\Delta_i | ۱ \leqslant i \leqslant n\}$. اگر حالت اول برقرار است، آنگاه $f(a_i) = f(a_i) + \Delta$ ودر حالت دوم $f(a_i) = f(a_i) - \Delta$.

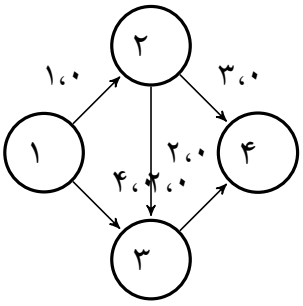
۴: به مرحله۱ بروید.

۵: Dبرای تمام کمانهای $f(a)$ را به عنوان خروجی تعیین کنید. فرض کنید P مجموعه ای از رأسهای x از D' باشد که برچسبهای متناهی را در مرحله۲ بعد از به کار بردن الگوریتم مور برایD' دریافت می کند. در این صورت (P, P') یک کمترین برش است.

۲-۱-۱ الگوریتم جستجوی اول – سطحی مور

این الگوریتم کمترین فاصله بین دو راس دلخواه از گراف را به دست می آورد. فرض کنیم می خواهیم فاصله بین راسهای s و t را در یک گراف (یا گراف جهتدار) تعیین کنیم. در ابتداs برچسب ۰ دارد که فاصله s از خودش را نشان می دهد و سایر راسها برچسب ∞ دارند.سپس به تمام راسهای مجاور s (یا قابل دسترسی از s) برچسب ۱ می دهیم.به تمام رئوس دارای برچسب ∞ که با رئوس دارای برچسب ۱، مجاورند (یا از طریق آنها قابل دسترسی اند)، برچسب ۲ می دهیم. به همین روش ادامه می دهیم تاt یک برچسب متناهی بگیرد، یا در یک مرحله تمام راسهای مجاور (یا قابل دسترس) رئوس برچسب گذاری شده قبلا ، برچسب متناهی گرفته باشند.در پایان الگوریتم هر راس w با برچسب متناهی ، برچسب $d(u,w)$ دارد که $d(s,w) \leqslant d(s,t)$.

الگوریتم مور کوتاهترین مسیر را بدون در نظر گرفتن فاصله راسها (وزن یالها) به دست می آورد. یعنی مسیر با کمترین تعداد یال را مشخص می کند.



شکل ۲-۱: شبکه

الگوریتم ۲-۳ الگوریتم جستجوی اول – سطحی مور [۲]

ورودی: گراف زمینه(جهتدار) D ، راس مبدا s و راس مقصد t

خروجی: کوتاه ترین مسیر بینs و t

۱: برچسب s را برابر ۰ و برچسب تمام رئوس به جز s را ∞ قرار دهید
$L(s) = ۰ \ , \ L(v) = \infty \ , \ v \neq \{s\}$
که L نشاندهنده فاصله تا مبدأ است .متغیر اندیسگذار i رامقدار صفر دهید. صف Q ، شامل تمام رأسهایی از D است که فاصله آنها تا مبدأ مشخص شده ، در ابتدا تنها شامل s است. ومجموعه P' در ابتدا مقدار اولیه $V(D) - \{s\}$ را دارد.

۲: اگر $Q \neq \emptyset$ راس x را از Q حذف کنید، در غیر این صورت توقف کنید، چون هیچ $-st$ مسیری وجود ندارد.

۳: برای هر راس y مجاور x(یا قابل دسترس برای x)که $L(y) = \infty$ است ، قرار دهید: $PARENT(y) = x$ و $L(y) = L(x) + ۱$ و y را به صف Q اضافه کنید. ($PARENT(y)$ پدر راس y در درخت جستجوی اول – سطحی است.)

۴: اگر $L(y) = \infty$ ،به مرحله ۲ برگردید؛ در غیر این صورت به مرحله ۵ بروید.

۵: قرار دهید $k = L(v), u_k = v, u_۰ = s$

۶: اگر $k \neq ۰$ آنگاه $u_{k-۱} = PARENT(k)$ در غیر این صورت به مرحله ۸ بروید.

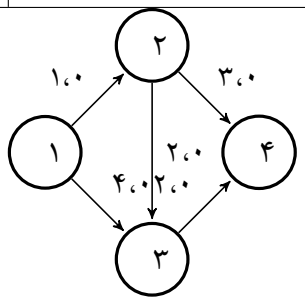
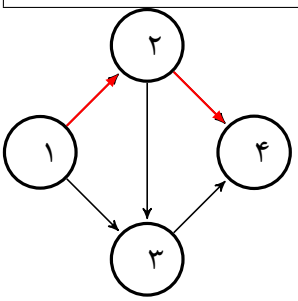
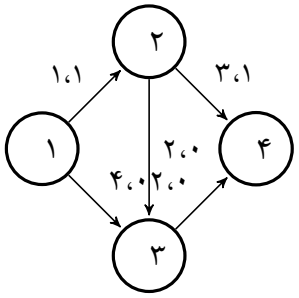
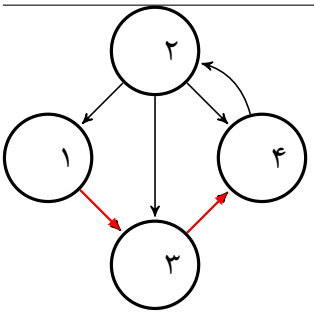
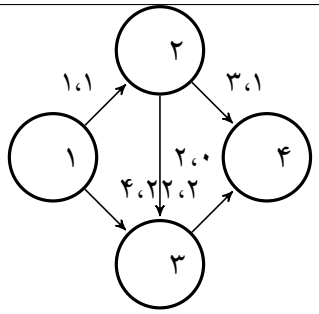
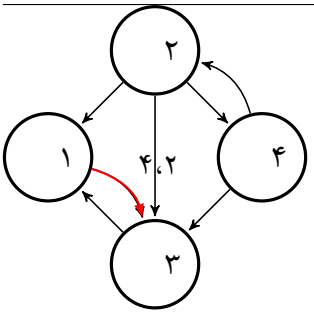
۷: قرار دهید $k = k - ۱$ و به مرحله ۶ بروید.

۸: خروجی $u_۰, u_۱, ..., u_k$ کوتاهترین $-st$ مسیر است.

مثال ۲-۱-۱. شبکه ۲-۱ را در نظر بگیرید ماکزیمم جریان را برای این شبکه به روش ادموندز -کارپ به دست می آوریم. تکرارهای این روش در جدول ۲-۲ آمده است. در این جدول گراف (D, D') نشان داده شده است. یالهای قرمز نشان دهنده کوتاهترین مسیر به دست آمده از الگوریتم مور است که تکرارهای آن در جدولهای ۲-۳، ۲-۴ و ۲-۵ آمده است. در جدول ۲-۳ کوتاه ترین مسیر برابر است با ۱, ۲, ۴ با طول ۲ که معادل با شبه مسیر ۱, ۲, ۴ در D است. قرار می دهیم

$$u_۰ = ۱ \quad a_۱ = (۱, ۲) \quad u_۱ = ۲ \quad a_۲ = (۲, ۴) \quad u_۲ = ۴$$

جدول ۲-۲: تکرار های الگوریتم ادموندز- کارپ

	گراف D	گراف D' وکوتاهترین $st-$ مسیر
تکرار ۱		
تکرار ۲		
تکرار ۳		

باتوجه به توضیحات بالا داریم:

$$\Delta_۱ = c(a_۱) - f(a_۱) = ۱ - ۰ = ۱ \quad \Delta_۲ = c(a_۲) - f(a_۲) = ۳ - ۰ = ۳ \quad \Rightarrow \quad \Delta = ۱$$

بنابراین تغییرات زیر را درD داریم:

$$f(a_۱) = f(a_۱) + \Delta \quad f(a_۲) = f(a_۲) + \Delta$$

تکرار ۲ الگوریتم ادموندز -کارپ مشابها انجام می شود و در تکرار ۳ هیچ شبه مسیر f افزایشی درDپیدا نشده، به پایان الگوریتم می رسیم و مجموعه های برش با توجه به جدول ۲-۵ مشخص می شوند. در این جدول تنها۱ و

جدول ۲-۳: الگوریتم مور برای تکرار ۱

صف Q	۴	۳	۲	۱
۱	$(\infty,-)$	$(\infty,-)$	$(\infty,-)$	۰
۲,۳	$(\infty,-)$	$(۱,v۱)$	$(۱,v۱)$	۰
۴	$(۱,v۱)$	$(۱,v۱)$	$(۱,v۱)$	۰

جدول ۲-۴: الگوریتم مور برای تکرار ۲

صف Q	۴	۳	۲	۱
۱	$(\infty,-)$	$(\infty,-)$	$(\infty,-)$	۰
۳	$(\infty,-)$	$(۱,v۱)$	$(\infty,-)$	۰
۴	$(۲,v۳)$	$(۱,v۱)$	$(\infty,-)$	۰

۳برچسب متناهی دارند؛ بنابراین

$$P = \{۱,۳\} \quad \bar{P} = \{۲,۴\}$$

اگر الگوریتم ادموندز کارپ را برای مثال جدول ۲-۱ به کار ببریم باز هم مانند مثال جدول ۲-۲ در تکرار۳ به پایان می رسد.در هر تکرار ارسال جریان برابر با M است.

۲-۲ الگوریتم چپ ترین مسیر

این الگوریتم در سال ۲۰۰۸، توسط بورَدایل و کلین برای یافتن جریان بیشینه در یک گراف مسطح جهت دار ارائه شد. پس از یک مرحله ی پیش پردازش که شامل یافتن فاصله کوتاه ترین مسیر از یک مبدأ در گراف دوگان است، الگوریتم آن ها شامل اشباع پیاپی "چپ ترین مسیر"باقیمانده[۹] از s به t است.

الگوریتم چپ ترین مسیر، تعمیم مستقیمی از الگوریتم بالاترین مسیر فورد- فولکرسون است. در این تعمیم، به جای بالاترین مسیر، چپ ترین جریان یافته می شود. در شروع، الگوریتم با یک چپ ترین جریان با میزان صفر شروع می کند. گام های این الگوریتم به صورت زیر است:

۱. یک وجه مجاور t را به عنوان f_∞ در نظر بگیر.

[۹] remainder

جدول ۲-۵: الگوریتم مور برای تکرار ۳

صف Q	۴	۳	۲	۱
۱	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	۰
۳	$(\infty, -)$	$(۱, v۱)$	$(\infty, -)$	۰
$\emptyset$	$(\infty, -)$	$(۱, v۱)$	$(\infty, -)$	۰

۲. دورهای ساعت گرد را اشباع کن.(الگوریتم چپ ترین گردش)

۳. تا زمانی که یک مسیر افزایشی s به t وجود دارد، چپ ترین این مسیرها را اشباع کن.(الگوریتم چپ ترین جریان)

۲-۲-۱ حذف دورهای ساعت گرد (الگوریتم چپ ترین گردش)

دومین مرحله از الگوریتم چپ ترین مسیر، توسط الگوریتم زیر از کولر [۱۰] و همکارانش [۱۵] پیاده سازی می شود. این الگوریتم با استفاده از فاصله های کوتاه ترین مسیر از یک مبدأ واحد و در نظر گرفتن ظرفیت ها به عنوان فاصله در گراف مسط ح دوگان، یک انتساب پتانسیل به وجه ها را محاسبه می کند و سپس با توجه به آن، یک گردش (جریان با میزان صفر) η در گراف اولیه می یابد به گونه ای که گراف باقیمانده نسبت به آن هیچ دور ساعت گردی نداشته باشد. روال کار به صورت زیر است.

فرض کنید f_∞ وجه بینهایت در گرافGباشد. ابتدا یک تابع پتانسیل Φ که به هر وجه f از گراف اصلی G فاصله از f_∞ به f در دوگان را نسبت می دهد، تعریف می کنیم. فرض کنید d کمانی از G باشد که وجه f_l در سمت چپ و وجه f_r در سمت راست آن است. اگر داشته باشیم $\Phi[f_r] > \Phi[f_l]$ آنگاه جریان روی کمان d را قرار می دهیم $\eta_d = \Phi[f_r] - \Phi[f_l]$ و در غیر این صورت قرار می دهیم $\eta_d = ۰$ و $\eta_{rev(d)} = \Phi[f_l] - \Phi[f_r]$.
در هر صورت داریم: $\eta_d - \eta_{rev(d)} = \Phi[f_r] - \Phi[f_l]$.

[۱۰] Khuller

الگوریتم ۲-۴ الگوریتم چپ ترین گردش [۳] ، [۱]

ورودی: گراف G و وزن یالها U و وجه f_∞

خروجی: η

۱: وزن یالهای گراف اصلی را به عنوان طول یال های متناظر آنها در گراف دوگان G^* قرار دهید.

۲: برای هر وجه f، $\Phi[f] = dist_{G^*}(f_\infty, f)$ که در آن $dist_{G^*}(f_\infty, f)$ طول کوتاه ترین مسیر از وجه بینهایت (f_∞) به وجه f است.

۳: $\eta_d = \Phi[head_{G^*}(d)] - \Phi[tail_{G^*}(d)]$.

۴: η را برگردانید.

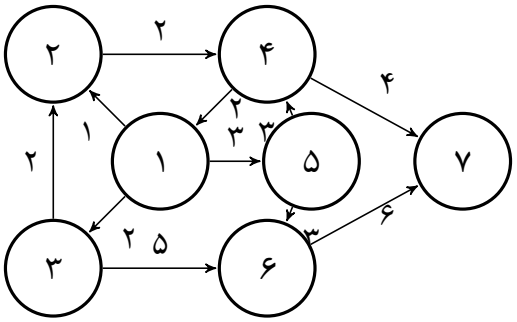
تابع η در محدودیت های ظرفیت کمان و محدودیت های توازن جریان صدق می کند. زیرا اولا برای هر کمان d در G کمان d^* در دوگان، f_l را به f_r وصل می کند. بنابراین مقدار $\Phi[f_l] - \Phi[f_r]$ حداکثر برابر طول d^* که همان ظرفیتdاست خواهد بود، یعنی جریان روی هیچ کمانی از ظرفیت آن کمان فراتر نخواهد رفت. ثانیا دوگان کمانهای خروجی از یک رأس v تشکیل یک دور می دهند.(به ازای هر کمان ورودی به v یک پیکان در جهت عکس و خروجی از v داریم.)از طرفی برای هر دور در G^* حاصل جمع $\eta_d - \eta_{rev(d)} = \Phi[f_r] - \Phi[f_l]$ به ازای کمان های دور برابر صفر است.بنابراین η یک تابع جریان شدنی است.

دور ساعت گرد C در G وجهی مانند f را در بر می گیرد. کوتاه ترین مسیر از f_∞ به f در گراف دوگان باید شامل یک کمان d^* باشد که دوگان یک کمان d از دور C است. از آنجا که d^* در درخت کوتاه ترین مسیرها قرار دارد، مقدار η_d برابراست با طول d^* که همان ظرفیت d است. بنابراین ظرفیت باقیمانده d نسبت به η برابر $u_d - (\eta_d - \eta_{rev(d)}) = ۰$ است و این یعنی d یک کمان باقیمانده نسبت به η نیست و بنابراین C یک دور باقیمانده نسبت به η نیست.

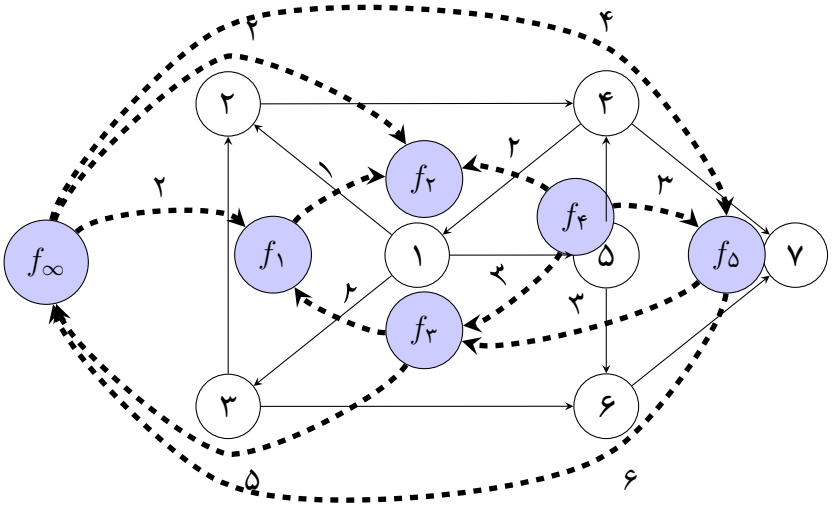
بخش زمانبر الگوریتم چپ ترین گردش مربوط به محاسبه ی کوتاه ترین مسیرها از رأس f_∞^* در گراف دوگان می باشد. در صورتی که برای این منظور از الگوریتم هنزینگر [11] و همکارانش [۱۲] استفاده شود، این زمان از مرتبه ی $O(n)$ خواهد بود، پس الگوریتم قابل پیاده سازی در زمان $O(n)$ است.

مثال ۲-۲-۱. شکل ۲-۲ یک مثال برای الگوریتم چپ ترین جریان است. در ۲-۲(الف) گراف اصلی رسم شده و ظرفیت یالها مشخص شده است. در ۲-۲(ب) گراف دوگان و طول کمان ها نشان داده شده است و با توجه به آن در جدول ۲-۶ کوتاه ترین مسیرها از f_∞ و همچنین پتانسیل هر وجه به دست آمده است. در ۲-۲(ج) گردش یافته شده توسط الگوریتم و در ۲-۲(د) گراف باقیمانده نشان داده شده است.

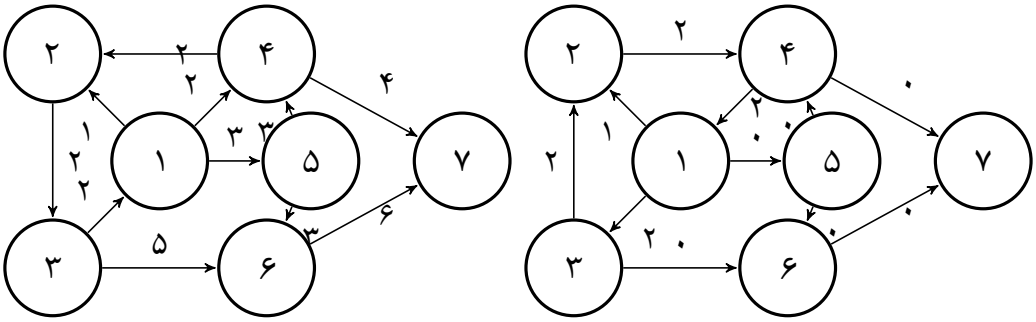
[11] Henzinger



(الف) ظرفیت یالهای گراف اصلی



(ب) طول یالهای دوگان



(ج) گردش یافته شده توسط الگوریتم

(د) گراف باقی مانده

شکل ۲-۲: چپ ترین گردش

جدول ۲-۶: جدول کوتاه ترین مسیرها برای مثال الگوریتم چپ ترین گردش

وجه	کوتاه ترین مسیر از f_∞	طول مسیر	پتانسیل
$f_۱$	$(f_\infty, f_۱)$	۲	۲
$f_۲$	$(f_\infty, f_۲)$	۲	۲
$f_۳$	$rev((f_۳, f_\infty))$	۰	۰
$f_۳$	$rev((f_۳, f_\infty)) - rev((f_۴, f_۳))$	۰	۰
$f_۵$	$rev((f_۵, f_\infty))$	۰	۰
f_∞	$-$	۰	۰

الگوریتم ۲-۵ خلاصه الگوریتم چپ ترین جریان [۳] ، [۱]

ورودی: گراف باقیمانده $G_\circ$ و وزن یالها $U_\circ$ و وجه f_∞ و رئوس s و t

خروجی: جریان f

۱: $f := \circ$

۲: تا زمانی که $st-$ مسیر اشباع نشده نسبت به f و $U_\circ$ وجود دارد مرحله ۳ را تکرار کنید.

۳: چپ ترین $st-$ مسیر اشباع نشده را اشباع کنید و f را به روز کنید.

۴: جریان f را برگردانید.

الگوریتم چپ ترین جریان

اکنون سومین مرحله از الگوریتم چپ ترین مسیر یعنی "چپ ترین جریان" را بررسی می کنیم.این الگوریتم با شروع از گراف باقیمانده $G_\circ$ خروجی دومین مرحله ، به صورت تکراری چپ ترین مسیر افزایشی از s به t را اشباع می کند.الگوریتم چپ ترین جریان، یک جریان بیشینه ی fدر $G_\circ$ می یابد. با اضافه کردن η به fیک جریان بیشینه در گراف اصلی به دست می آید. جریان اولیه به گونه ای انتخاب می شود که گراف باقیمانده اش هیچ دور ساعت گردی نداشته باشد (طبق تعریف ۱-۴-۱۶ ، یک چپ ترین جریان باشد) و در هر تکرار، جریان از اشباع یک چپ ترین مسیر به دست می آید. بنابراین با توجه به لم ۱-۴-۱۷ در طی مراحل اجرای الگوریتم چپ ترین جریان، $G_\circ$ هیچ دور باقیمانده ساعت گردی نسبت به جریان f ندارد.

نتیجه ۲-۲-۲. در طی مراحل اجرای الگوریتم چپ ترین جریان، هیچ دوری از پیکان ها که همگی جریان مثبت داشته باشند وجود ندارد.

اثبات: (فرض خلف) فرض کنید که C یک دور از پیکان هایی باشد که همگی جریان مثبت دارند.بنا به آنچه گفته شد C ساعت گرد نیست. اگر C پادساعت گرد باشد، آنگاه $rev(C)$ باقیمانده است که این با گفته بالا در تناقض است.

الگوریتم ۲-۶ پیاده سازی الگوریتم چپ ترین جریان [۳] ، [۱]

ورودی: گراف باقیمانده G_0 و وزن یالها U_0 و وجه f_∞ و رئوس tوs

خروجی: جریان f

۱: $f := 0$

۲: درخت اولیه T را با روش جستجوی اول راست به دست آورید. در این صورت کمانهای T دارای جهت به سمت t اند.

۳: فرض کنید Gگراف به دست آمده از G_0 باشد که با حذف تمام رئوسی به دست می آید که در T نیستند.

۴: درخت اولیه T^* از تمام یالهای G^* تشکیل شده است که دوگان آنها در T نیست.

۵: تکرار کنید

۶: اگر $-st$ مسیر اشباع نشده وجود دارد، آنگاه f را به روز کنید.

۷: فرض کنید $d = (u, v)$ نزدیکترین یال اشباع نشده نسبت به t در $T[s]$ باشد.

۸: فرض کنید $d^* = (f_1, f_2)$ دوگان یال d باشد.(f_1 وجه سمت چپ d و f_2 وجه سمت راست d باشد.)

۹: اگر در T^* از f_1 به f_2 مسیری وجود دارد، f را برگردانید.

۱۰: فرض کنید $e^* = (f_2, f_3)$ یالی از T^* است.

۱۱: فرض کنید $e = (i, j)$ یال اولیه متناظر با e^* باشد.(i راس سمت راست e^* و j راس سمت چپ آن باشد.)

۱۲: در درخت T^* یال e^* راحذف و $rev(d^*)$ رااضافه کنید.

۱۳: در درخت T یال d را حذف و e را اضافه کنید.

۱۴: پایان تکرار.

قضیه ۲-۲-۳. (از کار افتادگی) فرض کنید در یک مرحله جریان روی کمان a اضافه شود و در مرحله ای دیگر جریان روی $rev(a)$ اضافه شود. در این صورت جریان نمی تواند بار دیگر روی کمان a اضافه شود .

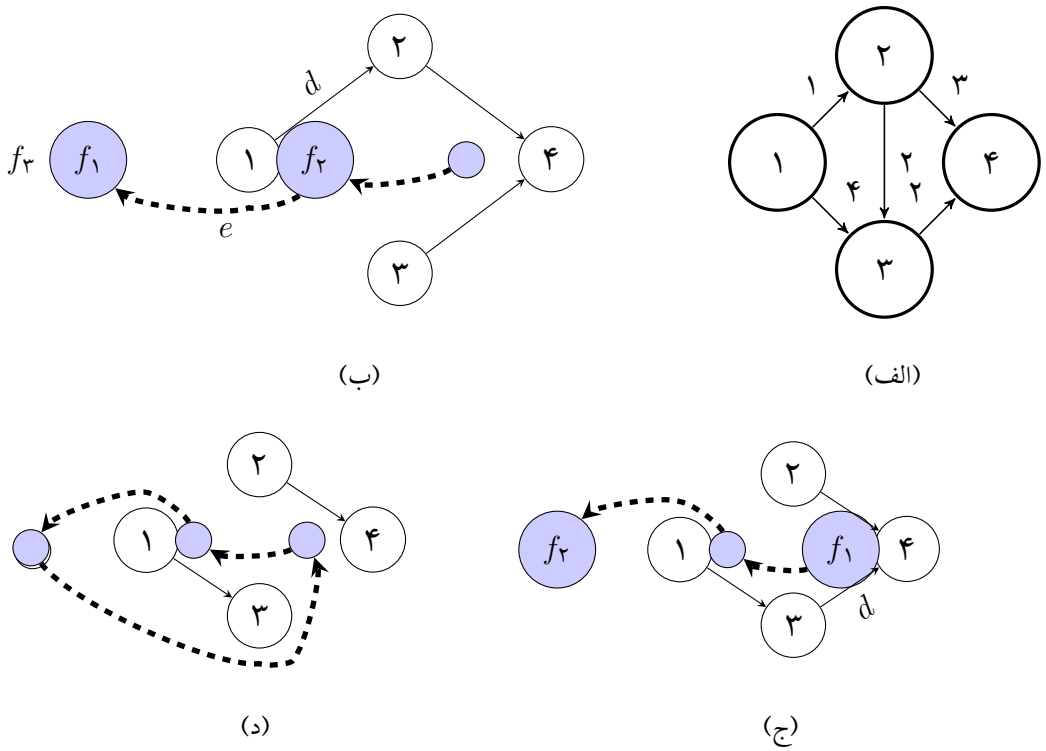
اثبات : به مرجع [۸] مراجعه شود.

برای رسیدن به زمان $O(nlogn)$ برای هر تکرار، یک پیاده سازی از الگوریتم چپ ترین جریان توسط بوردایل و کلین ارائه شده که در آن در بعضی از تکرارها در واقع هیچ جریانی ارسال نمی شود، ولی آنها نشان دادند که با استفاده از قضیه ۲-۲-۳ می توان تعداد تکرارها را به سه برابر تعداد کمان ها محدود کرد.

الگوریتم ۲-۶ یک درخت پوشای T که راس آن در مقصد است و درخت T^* در گراف دوگان که ریشه آن در وجه بینهایت است را نگهداری می کند. این الگوریتم ابتدا بررسی می کند که $-st$ مسیر اشباع نشده وجود دارد یا نه. در صورت وجود مسیر اشباع شده، جریان f به روز رسانی می شود. سپس پیکان $d = (u, v)$ نزدیکترین یال اشباع نشده نسبت بهt در $T[s]$ انتخاب و $d^* = (f_1, f_2)$ دوگان این یال در نظر گرفته می شود. سپس اگر در T^* از f_1 به f_2 مسیری وجود داشته باشد، الگوریتم به پایان می رسد. در غیر این صورت در درخت T^* یال $e^* = (f_2, f_3)$ حذف و $rev(d^*)$ اضافه می شود، در درختT یال e ، یال اولیه متناظر با e^* اضافه و یال d

حذف می شود و الگوریتم به ابتدای حلقه باز می گردد.

مثال ۲-۲-۴. شکل ۲-۳ مراحل الگوریتم چپ ترین جریان را نشان می دهد. ۲-۳(الف) گراف اولیه (گراف خروجی از مثال الگوریتم چپ ترین گردش) است. ۲-۳(ب) درخت پوشای گراف اولیه است که از جستجوی اول-راست پس رو به دست آمده و همچنین دوگان آن (با کمان های خط چین) نمایش داده شده است. در هر مرحله، کمانd, eو رئوس $f_۱, f_۲$ و $f_۳$ مشخص شده اند. توجه کنید که در ۲-۳(ج) مسیری از $f_۱$ به $f_۲$ وجود دارد .بنابراین الگوریتم پایان می یابد و مطابق ۲-۳(د) با حذف d از T و اضافه کردن $rev(d^*)$ به T^* دور مینیمم در گراف دوگان و برش کمینه در گراف اولیه به دست می آید .



شکل ۲-۳: مثال الگوریتم چپ ترین جریان

گزاره ۲-۲-۵. در تمام مراحل الگوریتم، Tیک درخت پوشا برایGو T^* دوگان یالهایی که در T نیستند، یک درخت پوشا برای G^* است.

اثبات : در آغاز کار، جستجوی اول-راست پس رو در گام ۲ یک درخت پوشا یTبرایG ایجاد می کند.و در گام (۴) طبق قضیه (درخت های پوشای جفت) ۱-۴-۲۹ یک درخت پوشای T^* برای G^* ایجاد میشود. حذف e^* از T^* در گام ۱۲ T^* را به دو مؤلفه ی همبند تقسیم می کند که یکی شامل نوادگان f_2 (راسهایی که از آنها به f_2 در T^* مسیر وجود دارد) و دیگری شامل سایر پیکان ها است.هنگام اجرای گام ۱۲ شرط گام ۹ برقرار نیست، پس f_1 یک نوه از f_2 در T^* نیست. پس با اضافه کردن $rev(d^*) = (f_1, f_2)$ دو مؤلفه ی همبند به یکدیگر ملحق می شوند. پس T^* در پایان گام ۱۲ یک درخت پوشا برای G^* است. بنابراین طبق قضیه درخت های پوشای جفت، ۱-۴-۲۹ بعد از گام ۱۳، Tیک درخت از G است.

توجه کنید که گام ۱۴ می تواند منجر به جابجایی پدر و فرزند در بعضی از یال ها شود. (تمام راسها باید به سمت tجهت دهی شده باشند).

گزاره ۲-۲-۶. فرض کنید e^* یالی از T^* و d پیکانی از G باشد که دوگان پیکانی از e^* است که جهت آن در T^* به سمت رأس f_∞ نیست.(از رأس f_∞ دور می شود). آنگاه d ناباقیمانده است.

اثبات: به مرجع [۱] مراجعه شود.

تعریف ۲-۲-۷. یک پیکان d را یک پیکان **غیر درختی**" می نامیم.اگر یال متناظر آن در T نباشد.

لم ۲-۲-۸. هیچ دور ساده ی ساعت گردی که پیکان های غیر درختی آن همگی باقیمانده باشند وجود ندارد.

اثبات: (فرض خلف) فرض کنید که C چنین دوری باشد و S مجموعه ی پیکان های غیر درختی و باقیمانده در دور C باشد. در این صورت طبق گزاره ۲-۲-۵ به ازای هر پیکان $d \in S$ درخت T^* شامل یال متناظر d است. از آنجا که هر پیکان S باقیمانده است، لذا طبق گزاره ۲-۲-۶ دوگان پیکان های S در T^* به سمت ریشه یعنی f_∞ اند. C ساعت گرد است پس به ازای هر پیکان $d \in C$ ، $head(d^*)$ توسط C در بر گرفته می شود. چون T یک درخت وC یک دور است، C شامل حداقل یک پیکان غیر درختی است. مسیر جهت دار $T^*[head(d^*)]$ به صورت کامل توسط C در بر گرفته شده است، که این یعنی $f_\infty = end(T^*[head(d^*)])$ را نیز در بر می گیرد، که یک تناقض است.

در دو نتیجه بعدی نشان داده شده است که الگوریتم ۲-۶ همان الگوریتم ۲-۵ را پیاده سازی می کند. یعنی همواره چپ ترین جریان را اشباع می کند. نتیجه ۲-۲-۹ نشان می دهد که هر افزایش، در امتداد چپ ترین مسیر افزایشی از s به t است و نتیجه ۲-۲-۱۰ نشان می دهد که این الگوریتم تا زمانی که هیچ مسیر افزایشی از s به t وجود نداشته باشد، پایان نمی یابد.

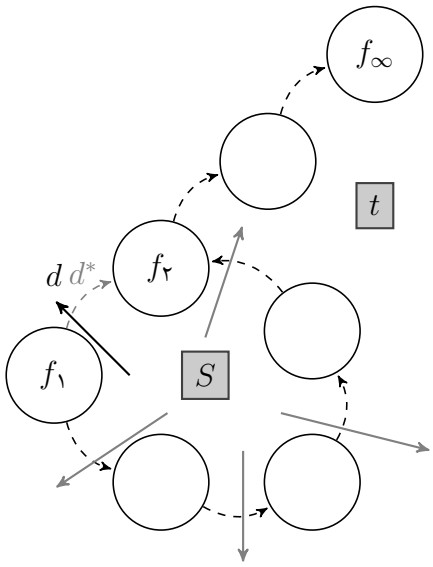
نتیجه ۲-۲-۹. به ازای هر رأس v هیچ مسیر افزایشی وجود ندارد که در سمت چپ $T[v]$ واقع شده باشد.

اثبات: (فرض خلف) فرض کنید که یک مسیر افزایشی وجود دارد که در سمت چپ $T[v]$ است. در این صورت، چپ ترین مسیر افزایشی از v به t را P می نامیم که باید در سمت چپ $T[v]$ باشد. فرض کنید Q زیرمسیری از P باشد به گونه ای که نقاط پایانی Q بر $T[v]$ واقع باشند ولی Q و $rev(Q)$ هر دو نسبت به $T[v]$ یال-مجزا باشند (هیچ یال مشترکی با $T[v]$ نداشته باشند).

چون P چپ ترین مسیر است ، طبق لم ۱-۴-۲۶ هر زیرمسیر از آن نیز یک چپ ترین مسیر است.بنابراین Q در سمت چپ $T[start(Q), end(Q)]$ است، پس $Qorev(T[start(Q), end(Q)])$ یک دور ساده و ساعت گرد است که پیکان های غیر درختی آن یعنی پیکان های Q باقیمانده هستند که با لم ۲-۲-۸ در تناقض است.

نتیجه ۲-۲-۱۰. جریان f خروجی از الگوریتم چپ ترین جریان، بیشینه است.

اثبات. شکل ۲-۴ را ملاحظه کنید.



شکل ۲-۴: پایان الگوریتم چپ ترین جریان

هنگامی که الگوریتم ۲-۶ پایان می یابد، $f_۱$ در T^* نوه $f_۲$ است. فرض کنید C دور ساده ی $d^*oT^*[f_۲, f_۱]$ در دوگان باشد، در این صورت در گراف اولیه G پیکان های C تشکیل یک برش جهت دار $[S, \bar{S}]$ می دهند. هر پیکان C به جز d^* یک پیکان غیر درختی است، پس مسیر از $v = head_G(d)$ به t در T از دوگان هیچ پیکان یا معکوس هیچ پیکان C استفاده نمی کند. با توجه به این که در گراف اولیه، رأس مقصد t در مرز وجه بی نهایت است، t توسط دور C در بر گرفته نشده است و بنابراین مجموعه ی رئوس S شامل رأس t نیست. به همین ترتیب، مسیر از s به $u = tailG(d)$ در T از دوگان هیچ پیکان یا معکوس هیچ پیکان C استفاده نمی کند.از آنجا که d ، دور C را قطع می کند، مجموعه ی رئوس S شامل رأس s است، لذا برش $[S, \bar{S}]$ یک $-st$

برش است. از طرفی تمام پیکان های تشکیل دهنده ی این $st-$ برش ناباقیمانده هستند زیرا d در آخرین تکرار از الگوریتم اشباع شده و سایر پیکان ها نیز طبق قضیه ی ۲-۲-۶ ناباقیمانده هستند، پس هیچ مسیر افزایشی از s به t وجود ندارد.بنابراین طبق قضیه ی ۱-۲-۴ جریان بیشینه است.

فرض کنید d یک پیکان باشد، در این صورت با توجه به ۲-۶ موارد زیر برقرارند:

۱. اگر d در لحظه i باقیمانده باشد و در لحظهj ناباقیمانده باشد. $(i < j)$ یک افزایش شامل d در لحظه ای بین i و j وجود داشته است.

۲. اگر d در لحظه i نا باقیمانده باشد و در لحظه j باقیمانده باشد. $(i < j)$ یک افزایش شامل $rev(d)$ در لحظه ای بین i و j وجود داشته است.

۳. هنگامی که e در T درج می شود، e باقیمانده است.

۴. هنگامی که d از T حذف می شود. d ناباقیمانده است.

لم ۲-۲-۱۱. یک پیکان $\langle a, ۱\rangle$ که aکمانی از $G_۰$ است، حداکثر یک بار حذف می شود.
اثبات در مرجع *[۸]*.

لم ۲-۲-۱۲. یک پیکان $\langle a, -۱\rangle$ که aکمانی از $G_۰$ است، حداکثر دو بار حذف می شود.
اثبات در مرجع *[۸]*.

قضیه ۲-۲-۱۳. حداکثر $۳m$ تکرار گام محوری در ۲-۶ (پیاده سازی الگوریتم چپ ترین جریان) وجود دارد.
اثبات: بنا به لم ۲-۲-۱۱ پیکان متناظر با یک کمان $G_۰$ حداکثر یک بار حذف می شود و بنا به لم ۲-۲-۱۲ پیکان متناظر با معکوس یک کمان $G_۰$ حداکثر دوبار حذف می شود.از طرفی در هر تکرار گام محوری در پیاده سازی الگوریتم چپ ترین جریان، یک پیکان حذف می شود، پس تعداد تکرارها حداکثر سه برابر تعداد کمان ها خواهد بود.

۲-۳ جمع بندی

در این فصل دو الگوریتم ادموندز -کارپ و چپ ترین مسیر را بیان کردیم که این الگوریتم ها از نوع الگوریتم های مسیر افزایشی می باشند. الگوریتم ادموندز -کارپ در هر مرحله با اشباع کوتاه ترین شبه مسیر افزایشی جریان بیشینه را می یابد. بنا به قضیه ۱-۲-۵ این مقدار با برش کمینه برابر است. الگوریتم چپ ترین مسیر در هر مرحله

چپ ترین شبه مسیر افزایشی را اشباع می کند. طبق قضیه ۲-۲-۱۰ خروجی این روش جریان بیشینه و بنابه قضیه ۱-۲-۵ برابر با برش کمینه است.

مرتبه زمانی الگوریتم چپ ترین مسیر کمتر از مرتبه زمانی الگوریتم ادموندز -کارپ است ولی برای گراف های نامسطح کاربرد ندارد.

در فصل ۴ الگوریتم ادموندز- کارپ را با چند روش از الگوریتم های غیر مبتنی بر جریان که در فصل ۳ ذکر شده اند ،مقایسه می کنیم.

فصل ۳

روشهای غیر مبتنی بر جریان در برش کمینه

در این فصل روشهایی بررسی می شوند که بر جریان مبتنی نیستند . یکی از این روشها بررسی تمام حالات است که بسیار زمان بر است.در یک گراف جهت دار دارای

$$\sum_{i=۱}^{|V|}(\binom{|V|-۱}{۰}+\binom{|V|-۱}{۱}/۲+\binom{|V|-۱}{۲}/۳+..+\binom{|V|-۱}{|V|-۲}/|V|-۱)$$

حالت مختلف است و در گراف های بدون جهت این تعداد نصف می شود. اولین الگوریتم دقیق برای به دست آوردن برش کمینه توسط ناگاموچی و ایباراکی [۱] در سال ۱۹۹۲ ارائه شد [۱۶]. الگوریتم استور- واگنر [۲] در سال ۱۹۹۴ برای گراف های بدون جهت ارائه شد که در مرجع [۴] آمده است و دارای مرتبه زمانی $O(|V|^۲||E|| + |V|log|V|)$ می باشد.این الگوریتم تعمیمی از الگوریتم مرجع [۱۶] است که مرتبه زمانی آن کمی بیشتر است ولی پیچیدگی کمتری نسبت به الگوریتم قبلی دارد .در این پایان نامه به معرفی الگوریتم استور واگنر می پردازیم. روش سیمپلکس [۳] در مرجع [۵] یک روش کلی برای حل مسائل برنامه ریزی خطی با حداکثر $\binom{m}{n}$ تکرار است. که n تعداد محدودیت ها و m تعداد متغیرها می باشد. در این پایان نامه مسئله برش کمینه را به صورت یک مسئله برنامه ریزی خطی مدلسازی می کنیم، سپس این روش را با یک مثال شرح می دهیم. در این صورت $|E| + ۱$ محدودیت و $|E| + |V|$ متغیر داریم و حداکثر در $\binom{|E|+|V|}{۱+|E|}$ تکرار به جواب بهینه می رسیم . چون مسئله جریان بیشینه دوگان مسئله برش کمینه است و با مدل سازی مسئله جریان بیشینه به صورت یک مسئله برنامه ریزی خطی می توان جواب بهینه مسئله برش کمینه را به دست آورد.و نیز با توجه به قیمت های سایه، راسها و یالهای برش را به دست آورد .

---

[۱] Nagamochi and Ibaraki
[۲] Stoer - Wagner
[۳] simplex

روشهای ذکر شده در بالا جزء روشهای دقیق بهینه سازی می باشند. الگوریتم کارگر [۴]، یک الگوریتم تصادفی برای محاسبه ی برش کمینه از یک گراف همبند است. این الگوریتم توسط دیوید کارگر اختراع و در سال ۱۹۹۳ برای نخستین بار چاپ شد. ایده ی الگوریتم مبنی بر مفهوم تلفیق یال در گراف بدون جهت است. به بیان غیر رسمی، تلفیق یک یال رأس های ابتدا و انتهای آن یال را در هم ادغام می کند و تعداد کل رأس های گراف را یکی کاهش می دهد. تمام یال های دیگر که به ابتدا و یا به انتهای یال متصل هستند، به رأس ادغامی «دوباره ضمیمه می شوند» و در عمل یک گراف چندگانه تولید می شود. الگوریتم اصلی کارگر یکی پس از دیگری یال هایی را که به صورت تصادفی انتخاب شده اند با هم تلفیق می کند تا فقط دو رأس باقی بماند؛ آن دو رأس نمایانگر یک برش در گراف اولیه هستند. با تکرار این الگوریتم تا زمانی که دو راس باقی بماند، با احتمال موفقیت زیادی می توان یک برش کمینه پیدا کرد. این الگوریتم برای گراف های جهت دار خطای بیشتری دارد که در فصل بعد به آن اشاره شده است.

برای گرافهایی با تعداد راسهای بالا الگوریتمها ی ذکر شده زمانبر است. به همین دلیل در این پایان نامه الگوریتم های فرا ابتکاری شبیه سازی تبریدی [۵] و جستجوی ممنوعه [۶] بر روی مسئله برش کمینه بررسی می شوند .

۳-۱ بررسی تمام برشها ی گراف

تعداد تمام $st-$ برشهای کمینه برابر است با:

$$\binom{|V|-۲}{۰}+\binom{|V|-۲}{۱}+\binom{|V|-۲}{۲}+..+\binom{|V|-۲}{|V|-۲}$$

تعداد کل برشها برابر است با:

$$T=\sum_{i=۱}^{|V|}(\binom{|V|-۱}{۰}+\binom{|V|-۱}{۱}/۲+\binom{|V|-۱}{۲}/۳+..+\binom{|V|-۱}{|V|-۲}/(|V|-۱)$$

اگر گراف D بدون جهت یا متقارن باشد تعداد کل برشها برابر است با: $T/۲$.

در گراف با تعداد راس بالا بررسی تمام حالات زمان بسیار زیادی لازم دارد.

۳-۲ استور واگنر

این روش برش کمینه را با الگوریتم بازگشتی به دست می آورد [۴]. فرض کنید G گراف بدون جهت با مجموعه رئوس V و مجموعه یالهای E باشد که هر لبه e دارای وزن نا منفی $W(e)$ است. اگر s و t دو راس از G باشند و $G/\{s,t\}$ گراف به دست آمده از ادغام s و t باشد برش کمینه G برابر است با کوچکترین $st-$ برش کمینه G یا برش کمینه $G/\{s,t\}$.

[۴] Karger's algorithm
[۵] Simulated Annealing
[۶] Tabu Search

الگوریتم ۳-۱ الگوریتم استور واگنر [۴]

ورودی: گراف بدون جهت G و W وزن لبه های G

خروجی: برش کمینه و مجموعه رئوس برش $(S, V-S)$

۱: تا زمانی که $|V| > ۱$ ادامه دهید

۲: مرحله ی برش کمینه را به دست آورید.

۳: اگر برش فازی از برش کمینه جاری کمتر است ، برش فازی را به عنوان برش مینیم جاری در نظر بگیرید.

الگوریتم ۳-۲ مرحله ی برش کمینه

ورودی: گراف بدون جهت G و W وزن لبه های G

خروجی: $st-$ برش کمینه

۱: راس دلخواه a را انتخاب وقرار دهید $A = a$

۲: اگر $A \neq V$ ادامه دهید

۳: راسی راکه بیشترین اتصال به A را دارد به A اضافه کنید .

۴: . در هر مرحله وزن برش (وزن یالهای منقبض شده) و راسهای اضافه شده اخیر را ذخیره کنید.

در $G/\{s,t\}$ ، گراف به دست آمده از ادغام s و t رئوس فقط s و t یکی می شوند. ولی در مورد یالها، یال متصل کننده s و t حذف شده و بقیه یالهای متصل به s یا t به این گره جدید متصل می شوند. در این الگوریتم ابتدا راس دلخواه a را انتخاب وقرار می دهیم $A = a$ واگر $A \neq V$ ادامه می دهیم . راسی که بیشترین اتصال بهAدارد را انتخاب وبه A اضافه می کنیم. یالهای بین این راس و مجموعه A را منقبض می کنیم . وزن این یالها با وزن برش در این مرحله برابر است. این مرحله را مرحله ی برش کمینه می نامیم. این مرحله را تا زمانی که $|V| > ۱$ تکرار می کنیم. ودر هر مرحله وزن برش را ذخیره می کنیم. مینیمم وزن این برش ها با برش کمینه برابر است. در هر مرحله مجموعه A مجموعه S وبقیه رئوس مجموعه $V - S$ اند. راس z اضافه شده به A در هر مرحله به صورت زیر تعیین می شود:

$$W(A, z) = \max\{W(A, y) | y \notin A\}$$

وزن برش در هر مرحله برابر است با $W(A, y)$ که همان مجموع وزنهای تمام یالهای متصل شده از مجموعه A به y است . در هر بار تکرار مرحله ی برش کمینه a به طور دلخواه انتخاب می شود.

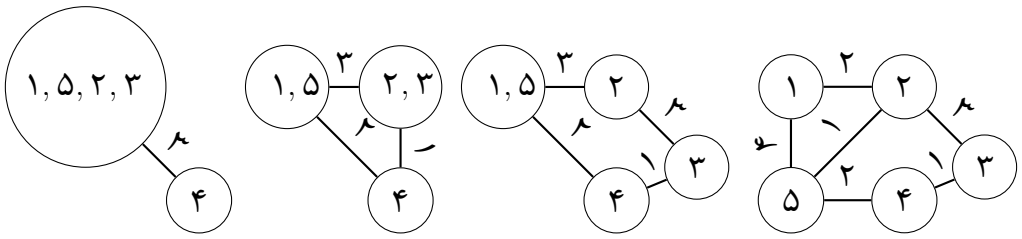
۳-۲-۱ اثبات درستی الگوریتم

قضیه ۳-۲-۱. هر مرحله از الگوریتم یک $st-$ برش ازگراف فعلی است که s و t دو راس اضافه شده در مرحله قبل ۱ اند.

اثبات:فرض کنید C یک $st-$ برش دلخواه ازگراف فعلی باشد که سنگین ترین برش مرحله قبل است. فرض کنید v راس فعال و a راسی باشد که $v \neq a$. v و رئوس اضافه شده قبل از v دو بخش مختلف C اند.فرض کنید $W(C)$ وزن برش و A_v مجموعه تمام رئوس اضافه شده قبل از v (به جز v) باشد و C_v برش $A_v \cup \{v\}$ به دست آمده از C باشدبرای هر راس فعال v نشان می دهیم که: $W(A_v, v) \leqslant W(C_v)$.

اثبات را با استقرا روی رئوس فعال انجام می دهیم. برای اولین راس فعال نامساوی بالا تبدیل به مساوی می شود. فرض کنید نامساوی برای تمام رئوس فعال که قبل از v فعال شده اند برقرار باشد و v راس فعال شده بعدی باشد که اضافه می شود. دراین صورت داریم: $W(A_u, u) = W(A_v, u) + W(A_u/A_v, u) := \alpha$ اکنون چون v به عنوان راسی که بیشترین اتصال به A_v را دارد انتخاب شده است. $W(A_v, u) \leqslant W(A_v, v)$. بنا به فرض استقرا $W(A_v, v) \leqslant W(C_v)$. تمام یالهای بین A_u/A_v و u بخشهای مختلفCرا متصل می کنند. بنابراین آنها در $W(C_u)$ شرکت دارند و در $W(C_v)$ شرکت ندارند. بنابراین

$$W(C_u) = W(A_u/A_v, u) + W(C_v) \geqslant \alpha.$$



(الف) $w = ۶, S = \{۱\}$ (ب) $w = ۴, S = \{۳\}$ (ج) $w = ۵, S = \{۱,۵\}$ (د) $w = ۳, S = \{۱,۵,۲,۳\}$

شکل ۳-۱: مثال الگوریتم استور واگنر

مثال ۳-۲-۲.

الگوریتم ۳-۳ الگوریتم سیمپلکس [۵]

ورودی: مدل ریاضی مساله

خروجی: مقدار بهینه تابع هدف و مقدار بهینه برای متغیرها

۱: در قدم اول باید مساله را به صورت استاندارد تبدیل کنیم.

$$\max CX$$
$$st \quad AX \leqslant b \qquad X \geqslant 0$$

۲: برای پر کردن ماتریس سیمپلکس باید یک جواب موجه اساسی تعیین نمود.

۳: **انتخاب متغیر ورودی**

$x_k = \min\{z_j - c_j; z_j - c_j \leqslant 0\}$. ستون زیر متغیر kام ستون لولا است.

۴: **انتخاب متغیر خروجی**

$x_r = \frac{b_r}{y_{rk}} = \min \frac{b_i}{y_{ik}}, y_{ik} > 0$

ردیف متغیر خروجی را سطر لولا و محل تلاقی سطر و ستون لولا را عنصر لولا می نامیم.

۵: ضرایب سطر لولای جدید را با استفاده ازفرمول زیر محاسبه می کنیم: $\frac{\text{سطر لولای قدیم}}{\text{عدد لولا}} =$ سطر لولا ی جدید

۶: ضرایب ردیف های دیگر را براساس فرمول زیرمحاسبه میکنیم:

(ضرایب مربوط به سطر لولای جدید *ضرایب مربوط در ستون لولا) - ضرایب سطر قدیم = ضرایب سطرجدید

۷: اگرهمه مقادیر سطر صفر برای مساله بیشینه سازی غیر منفی باشند، جواب بهینه حاصل شده است در غیراین صورت به مرحله ۳ برگردید

۳-۳ سیمپلکس

روش سیمپلکس [۵] یک روش کلی برای حل مسائل برنامه ریزی خطی است. دراین روش، ابتدا مدل وارد یک جدول می شود و سپس یک سری مراحل ریاضی ابتدایی بر روی جدول اجرا می گردد. مراحل روش سیمپلکس به نحو اثربخشی بیان گر فرآیند روش ماتریسی می باشد که حرکت از یک گوشه به گوشه ای با تابع هدف بهتر (حداقل نه بدتر) حرکت می کند. ابتدا برای پر کردن ماتریس سیمپلکس باید یک جواب موجه اساسی تعیین نمود.

انتخاب متغیر ورودی

در یک مساله بیشینه سازی باید متغیری با $z_j - c_j$ منفی را انتخاب کنیم و بهتر است که منفی ترین متغیر را انتخاب کنیم.ستون زیر این متغیر را "ستون لولا" می نامیم.

$$x_k = \min\{z_j - c_j; z_j - c_j \leqslant 0\}$$

انتخاب متغیر خروجی

باتوجه به متغیر ورودی متغیرخروجی ، متغیری است که دارای حداقل حاصل تقسیم مقادیر سمت راست بر عناصر مثبت ستون لولا باشد.ردیف متغیر خروجی را سطر لولا و محل تلاقی سطر و ستون لولا را عنصر لولا می نامیم.

$$x_r = \frac{b_r}{y_{rk}} = \min \frac{b_i}{y_{ik}}, y_{ik} > ۰$$

جدول جدید سیمپلکس

$$\text{سطر لولا ی جدید} = \frac{\text{سطر لولای قدیم}}{\text{عدد لولا}}$$

(ضرایب مربوط به سطر لولای جدید *ضرایب مربوط در ستون لولا) - ضرایب سطر قدیم = ضرایب سطرجدید

شرط بهینگی

شرط بهینگی را با کنترل عناصر سطرصفر بررسی کنید. اگرهمه مقادیر سطر صفر برای مساله بیشینه سازی غیر منفی باشند، جواب بهینه حاصل شده است در غیراین صورت به مرحله ۳ برگردید.

روش سیمپلکس برای حل مساله حداقل سازی

یک مساله با تابع هدف حداقل سازی را نیز می توان ابتدا به یک مساله با تابع هدف بیشینه سازی تبدیل کرد، سپس مراحل بیان شده را انجام داد. همچنین یک مساله حداقل سازی را میتوان به صورت مستقیم نیز دقیقا مانند یک مساله بیشینه سازی حل نمود تنها در گام ۳ به منظور انتخاب متغیر ورودی باید متغیری با مقادیر مثبت در سطر صفر انتخاب شود و هنگام تست بهینگی (گام ۷) جواب بهینه هنگامی بدست آمده که کل ضرایب سطر صفر نامثبت باشند.

گزاره ۳-۳-۱. هرگاه ضریب یک متغیر غیراساسی در سطر صفر تابلوی بهینه سیمپلکس مساوی صفر باشد. آن مساله دارای جواب بهینه چندگانه است.

گزاره ۳-۳-۲. هرگاه در تابلوی آخر سیمپلکس، امکان انتخاب متغیر ورودی وجود داشته باشد اما متغیر خروجی به دلیل مثبت نبودن ضرایب ستون لولا قابل تعریف نباشد، مدل برنامه ریزی خطی دارای جواب نامتناهی می باشد.

گزاره ۳-۳-۳. هرگاه در یک تابلو بیش ار یک متغیر شرایط خروج از پایه را داشته باشند، آنگاه در جدول

جدول ۳-۱: جدول سیمپلکس

متغیرها	X_B	X_N	سمت راست
z (سطر صفر)	۰	$z_j - c_j$	$c_B B^{-1} b$
X_B	I	Y	$B^{-1} b$

بعدحداقل یکی از متغیرهای اساسی برابر با صفر می شوند. در این صورت تابلوی بدست آمده نشان دهنده یک راس تبهگن می باشد. اگر تابلوی بدست آمده بهینه باشد، آنگاه آن گوشه، گوشه بهینه تبهگن است و اگر تابلوی حاصل غیر بهینه باشد، آنگاه گوشه متناظر یا از نوع تبهگن موقت است، که تابلوی بعدی سیمپلکس دیگر در سمت راست صفر نخواهد داشت و یا اینکه تابلوی بعد باز هم تبهگن می شود.

مدل ریاضی مسئله جریان بیشینه از گره s به گره t برای گراف با n راس به صورت زیر است:

$$\max f \tag{۱-۳}$$

s.t.

$$\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = \begin{cases} f & i = s \text{ اگر} \\ 0 & i \neq s, t \text{ اگر} \\ -f & i = t \text{ اگر} \end{cases}$$

$$0 \leqslant x_{ij} \leqslant c_{ij} \quad i, j = 1, ..., n$$

که در آن مجموع ها و نامعادله ها روی یالها تعریف شده اند و به فرمول یال-گره مسئله جریان بیشینه معروف است. چون ماتریس محدودیت ها یک ماتریس وقوع یال-گره است. باید توجه داشت که f یک متغیر است و با نشان دادن ماتریس وقوع یال-گره با A مدل ریاضی مسئله جریان بیشینهبه صورت زیر است:

$$\max f \tag{۲-۳}$$

s.t.

$$(e_t - e_s) f + AX = 0$$

$$0 \leqslant X \leqslant C$$

دوگان مسئله جریان بیشینه به صورت زیر است:

$$\min \sum_{i=۱}^{n} \sum_{j=۱}^{n} c_{ij} h_{ij} \qquad (۳-۳)$$

s.t.

$$w_t - w_s = ۱$$

$$w_i - w_j + h_{ij} \geqslant ۰ \quad i, j = ۱, ..., n$$

$$h_{ij} \geqslant ۰ \quad i, j = ۱, ..., n$$

اگر $(X, \bar{X})$ برش دلخواهی باشد و فرض کنیم:

$$w_i = \begin{cases} ۰ & \text{اگر } i \in P \text{ باشد} \\ ۱ & \text{اگر } i \in \bar{P} \text{ باشد} \end{cases}$$

$$h_{ij} = \begin{cases} ۱ & \text{اگر } (i, j) \in (P, \bar{P}) \text{ باشد} \\ ۰ & \text{در غیر این صورت} \end{cases}$$

آنگاه این انتخاب ویژه از w و h یک جواب شدنی برای مسئله دوگان فراهم می آورد که تابع هدف دوگانش با ظرفیت برش مساوی است.

مثال ۳-۳-۴. شبکه ۲-۳(الف) را در نظر بگیرید.مدل ریاضی مسئله جریان بیشینه و دوگان آن برای این شبکه

به صورت زیر است:

$$
\begin{array}{lll}
P)\ \max\ f & & \max\ f' - f'' - MR_۱ - MR_۲ - MR_۳ - MR_۴ \quad (۴\text{-}۳) \\
\text{s.t.} & & \text{s.t.} \\
x_{۱۲} + x_{۱۳} - f = ۰ & & x_{۱۲} + x_{۱۳} - f' + f'' + R_۱ = ۰ \\
-x_{۱۲} + x_{۲۳} + x_{۲۴} = ۰ & & -x_{۱۲} + x_{۲۳} + x_{۲۴} + R_۲ = ۰ \\
-x_{۱۳} - x_{۲۳} + x_{۳۴} = ۰ & & -x_{۱۳} - x_{۲۳} + x_{۳۴} + R_۳ = ۰ \\
-x_{۲۴} - x_{۳۴} + f = ۰ & & -x_{۲۴} - x_{۳۴} + f' - f'' + R_۴ = ۰ \\
۰ \leqslant x_{۱۲} \leqslant ۱ & & x_{۱۲} + s_۱ = ۱ \\
۰ \leqslant x_{۱۳} \leqslant ۴ & \Rightarrow & x_{۱۳} + s_۲ = ۴ \\
۰ \leqslant x_{۲۳} \leqslant ۲ & & x_{۲۳} + s_۳ = ۲ \\
۰ \leqslant x_{۲۴} \leqslant ۳ & & x_{۲۴} + s_۴ = ۳ \\
۰ \leqslant x_{۳۴} \leqslant ۲ & & x_{۳۴} + s_۵ = ۲ \\
 & & x_{ij}, R_i, s_i, f', f'' \geqslant ۰, M \gg ۰
\end{array}
$$

$$
\begin{aligned}
D)\quad \min\ & h_{۱۲} + ۴h_{۱۳} + ۲h_{۲۳} \\
& + ۳h_{۲۴} + ۲h_{۳۴} \\
\text{s.t.}\quad & \\
& w_t - w_s = ۱ \\
& w_۱ - w_۲ + h_{۱۲} = ۰ \\
& w_۱ - w_۳ + h_{۱۳} = ۰ \\
& w_۲ - w_۳ + h_{۲۳} = ۰ \\
& w_۲ - w_۴ + h_{۲۴} = ۰ \\
& w_۳ - w_۴ + h_{۳۴} = ۰ \\
& h_{۱۲}, \geqslant ۰ \\
& h_{۱۳} \geqslant ۰ \\
& h_{۲۳} \geqslant ۰ \\
& h_{۲۴} \geqslant ۰ \\
& h_{۳۴} \geqslant ۰
\end{aligned}
\qquad
\begin{aligned}
& \min\ h_{۱۲} + ۴h_{۱۳} + ۲h_{۲۳} \\
& + ۳h_{۲۴} + ۲h_{۳۴} - MR \\
& \text{s.t.} \\
& w_t - w_s + R = ۱ \\
& w'_۱ - w''_۱ - w'_۲ + w''_۲ + h_{۱۲} = ۰ \\
& w'_۱ - w''_۱ - w'_۳ + w''_۳ + h_{۱۳} = ۰ \\
\Rightarrow\ & w'_۲ - w''_۲ - w'_۳ + w''_۳ + h_{۲۳} = ۰ \\
& w'_۲ - w''_۴ - w'_۲ + w''_۴ + h_{۲۴} = ۰ \\
& w'_۳ - w''_۴ - w'_۳ + w''_۴ + h_{۳۴} = ۰ \\
& h_{ij}, w'_i, w''_i, R \geqslant ۰ \\
& M \gg ۰
\end{aligned}
\qquad (۵-۳)
$$

جدول ۳-۲ جدول اولیه و بهینه برای مسئله P وجدول ۳-۳ جدول اولیه و بهینه برای مسئله D را نشان می دهد. عناصر زیر متغییرهای مصنوعی و مازاد در سطر هدف در جدول بهینه هر یک از مسئله ها متناظر با متغییر های مسئله دوگان است. از جدول ۳-۲ داریم:

$$Z^* = f^* = ۳,\ x^*_{۱۲} = ۱,\ x^*_{۱۳} = ۲,\ x^*_{۲۳} = ۰,\ x^*_{۲۴} = ۱,\ x^*_{۳۴} = ۲$$

$$h^*_{۱۲} = h^*_{۳۴} = ۱,\ w^*_۲ = w^*_۴ = ۱\ \Rightarrow P = \{۱, ۳\}, \bar{P} = \{۲, ۴\},\ (P, \bar{P}) = \{x_{۱۲}, x_{۳۴}\}$$

و از جدول ۳-۳ داریم:

$$Z^* = ۳,\ h^*_{۱۲} = h^*_{۳۴} = ۱,\ h^*_{۱۳} = h^*_{۲۳} = h^*_{۲۴} = ۰,\ w^*_۲ = w^*_۴ = ۰,\ w^*_۱ = w^*_۳ = -۱$$

$$x^*_{۱۳} = ۲,\ x^*_{۲۳} = ۰,\ x^*_{۲۴} = ۱,\ x^*_{۳۴} = ۲\ \Rightarrow P = \{۱, ۳\}, \bar{P} = \{۲, ۴\},\ (P, \bar{P}) = \{x_{۱۲}, x_{۳۴}\}$$

جدول ۳-۲: جدول سیمپلکس برای مسئله P

	$x_{۱۲}$	$x_{۱۳}$	$x_{۲۳}$	$x_{۲۴}$	$x_{۳۴}$	f'	f''	$R_۱$	$R_۲$	$R_۳$	$R_۴$	$s_۱$	$s_۲$	$s_۳$	$s_۴$	$s_۵$	RSB
Z	۰	۰	۰	۰	۰	-۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
$R_۱$	۱	۱	۰	۰	۰	-۱	۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰
$R_۲$	-۱	۰	۱	۱	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰
$R_۳$	۰	-۱	-۱	۰	۱	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰
$R_۴$	۰	۰	۰	-۱	-۱	۱	-۱	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰
$s_۱$	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۱
$s_۲$	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۴
$s_۳$	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۲
$s_۴$	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۳
$s_۵$	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۲
	:	:									:					:	:
Z	۰	۰	۱	۰	۰	۰	۰	۰	۱	۰	۱	۱	۰	۰	۰	۱	۳
$R_۱$	۰	۰	۰	۰	۰	۰	۰	۱	۱	۱	۱	۰	۰	۰	۰	۰	۰
$x_{۲۴}$	۰	۰	۱	۱	۰	۰	۰	۰	۱	۰	۰	۱	۰	۰	۰	۰	۱
$x_{۳۴}$	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۲
f	۰	۰	-۱	۰	۰	۱	-۱	۰	۱	۰	۱	۱	۰	۰	۰	۱	۳
$x_{۱۲}$	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۱
$s_۲$	۰	-۱	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۱	۰	۰	-۱	۲
$s_۳$	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۲
$s_۴$	۰	۰	-۱	۰	۰	۰	۰	۰	-۱	۰	-۱	۰	۰	۰	۱	۰	۲
$x_{۱۳}$	۱	۱	۰	۰	۰	۰	۰	۰	۰	-۱	۰	۰	۰	۰	۰	۱	۲

جدول ۳-۳: جدول سیمپلکس برای مسئله D

	w'_1	w''_1	$\cdots$	h_{12}	h_{13}	h_{23}	h_{24}	h_{34}	R	s_1	s_2	s_3	s_4	s_5	RSB
Z	$5-M$	$M-5$	$\cdots$	۰	۰	۰	۰	۰	۰	-1	-4	-2	-3	-2	M
R	-1	۱	$\cdots$	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۱
h_{12}	۱	-1	$\cdots$	۱	۰	۰	۰	۰	۰	-1	۰	۰	۰	۰	۰
h_{13}	۱	-1	$\cdots$	۰	۱	۰	۰	۰	۰	۰	-1	۰	۰	۰	۰
h_{23}	۰	۰	$\cdots$	۰	۰	۱	۰	۰	۰	۰	۰	-1	۰	۰	۰
h_{24}	۰	۰	$\cdots$	۰	۰	۰	۱	۰	۰	۰	۰	۰	-1	۰	۰
h_{34}	۰	۰	$\cdots$	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	-1	۰
:	:						:							:	:
Z	۰	۰	$\cdots$	۰	-2	-2	-2	۰	$5-M$	-1	-2	۰	-1	-2	۳
w''_1	-1	۱	$\cdots$	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۱
h_{12}	۰	۰	$\cdots$	۱	۰	۰	۱	۰	۰	-1	۰	۰	-1	۰	۱
w''_3	۰	۰	$\cdots$	۰	۱	۰	۰	۰	۰	۰	-1	۰	۰	۰	۱
w'_4	۰	۰	$\cdots$	۰	۱	۰	۱	۰	۰	۰	۰	۰	-1	۰	۰
s_3	۰	۰	$\cdots$	۰	۱	-1	۱	۰	۰	۰	-1	۱	-1	۰	۱
h_{34}	۰	۰	$\cdots$	۰	۱	۰	۰	۱	۰	۰	-1	۰	۰	-1	۱

۳-۴ روش کارگر

این الگوریتم توسط دیوید کارگر [۶] اختراع و در سال ۱۹۹۳ برای نخستین بار چاپ شد. مانند روش استور واگنر در هر مرحله یک یال را منقبض می کند، ولی با این تفاوت که یال را به صورت تصادفی انتخاب می کند و الگوریتم را تا زمانی ادامه می دهد که دو راس باقی بماند. برش آخر را به عنوان برش کمینه در نظر می گیرد. (وزن برشها را در هر مرحله ذخیره نمی کند) برای اینکه احتمال رسیدن به جواب این روش را به دست آوریم نیاز به تعاریف و قضایای زیر داریم . فرض کنیم گرافGبدون وزن است

تعریف ۳-۴-۱ (احتمال شرطی X به شرط Y). را به صورت زیر تعریف میکنیم:

$$P(X=x|Y=y)=\frac{P((X=x)\cap(Y=y))}{P(Y=y)}$$

بنابراین داریم $P((X=x)\cap(Y=y))=P(X=x|Y=y)*P(Y=y)$
واگر X,Y مستقل باشند ، آنگاه $P((X=x)\cap(Y=y))=P(X=x|Y=y)*P(Y=y)$.
اگر n $\eta_1,\eta_2,...,\eta_n$ مجموعه باشند که لزوما مستقل نیستند، آنگاه

$$P(\bigcap_{i=1}^{n}\eta_i)=P(\eta_1)*P(\eta_2|\eta_1)*P(\eta_3|\eta_1\cap\eta_2)*...*P(\eta_n|\eta_1\cap\eta_2\cap...\cap\eta_{n-1})$$

قضیه ۳-۴-۲. اندازه برشG/xy حداقل به بزرگی برش درGاست و هر برشG/xy متناظربا یک برش در Gاست.

قضیه ۳-۴-۳. اگر $e_1,e_2,...,e_n$ یک مجموعه از یالهای G باشند که هیچ یک از آنها در برش کمینه نیستند. آنگاه $G'=G/e_1,e_2,...,e_n$ تنها یک یال چند گانه دارد که این یال چند گانه مطابق با برش کمینه است.

قضیه ۳-۴-۴. اگراندازه برش کمینه گرافGبرابر باK باشد آنگاه $|E(G)|\geqslant\frac{Kn}{2}$ کهn تعداد راسهاست. **اثبات** :درجه راسها حداقلk است زیرا اگر راسی مانند xبا درجه کمتراز kوجود داشته باشدآنگاه برش کمینه گرافGبرابر است با $[v/x,x]$ که اندازه آن کمتر ازkاست و این خلاف فرض است . بنابراین درجه راسها حداقلkاست و بنا براین مجموع درجات راسها حداقلnkاست .ازطرفی تعداد یالها برابر است با نصف مجموع درجات راسها و بنابراین $|E(G)|\geqslant\frac{Kn}{2}$.

قضیه ۳-۴-۵. اگر یال را برای انقباض تصادفی انتخاب کنیم با احتمال $\frac{2}{n}$ این یال در برش کمینه است.

الگوریتم ۳-۴ الگوریتم کارگر [۶]

ورودی: گراف بدون وزنG

خروجی: برش کمینه و مجموعه رئوس برش $(S, V-S)$

۱: قراردهید $G_۰ = G$و $i = ۰$

۲: تا زمانی که G_i بیش از دو راس دارد تکرار کنید.

۳: یک یال تصادفیe_iاز گراف G_iانتخاب کنید و قرار دهید $G_{i+۱} = G_i/e_i, i = i + ۱$

۴: فرض کنید $(S, V/S)$ برش مطابق با $G_{|V|-۲}$ در گراف اصلی باشد$(S, V/S)$ را برگردانید.

اثبات:گراف حداقل $\frac{Kn}{۲}$ یال دارد ودقیقا k یال در برش کمینه است.احتمال بودن هریال در برش کمینه کمتر مساوی $\frac{K}{\frac{Kn}{۲}}$ است.

این الگوریتم همیشه یک برش را برمیگرداند که کوچکتر از برش کمینه نیست.

قضیه ۳-۴-۶. خروجی این روش با احتمال $\binom{n}{۲}^{-۱} = \frac{۲}{n(n-۱)}$ برش کمینه است.کهn تعداد راسهایGاست.

اثبات: در مرحله اول احتمال اینکه یال تصادفی در برش کمینه نباشد،بزرگتر مساوی $\frac{۲}{n} - ۱$ است. فرض کنید رویداد η_i اتفاق می افتد، اگر e_i یال انتخاب شده از G_i یالی از برش کمینه نباشد و فرض کنید n_i تعداد رئوس گراف G_i باشد. اگر تمام رویدادهای $\eta_۰, \eta_۱, ..., \eta_{n-۳}$ رخ دهند، آنگاه تمام یالها ی انتخابی بیرون از برش کمینه اند و خروجی الگوریتم برش کمینه است.

در مرحله i ام احتمال اینکه تمام لبه ها ی انتخاب شده برای انقباض خارج از برش کمینه باشند برابر است با :

$$P(\eta_i|\eta_۰ \cap \eta_۱ \cap ... \cap \eta_{i-۱}) \geqslant ۱ - \frac{۲}{n_i} = \frac{۲}{n-i}$$

از طرفی داریم:

$$P(\bigcap_{i=۰}^{n-۳} \eta_i) = P(\eta_۰) * P(\eta_۱|\eta_۲) * P(\eta_۲|\eta_۱ \cap \eta_۰) * ... * P(\eta_{n-۳}|\eta_۰ \cap \eta_۱ \cap ... \cap \eta_{n-۴})$$

وبنابراین احتمال اینکه خروجی الگوریتم برش کمینه باشد برابر است با :

$$P(\bigcap_{i=۰}^{n-۳} \eta_i) \geqslant \prod_{i=۰}^{n-۳} (۱ - \frac{۲}{n-i}) = \frac{۲}{n(n-۱)}$$

بنابراین احتمال همگرایی الگوریتم حداقل برابر است با: $\frac{۲}{n(n-۱)}$.

لم ۳-۴-۷. اگر $۰ \leqslant x \leqslant ۱$ آنگاه $۱ - x \leqslant e^{-۲}$.

اثبات:بسط تیلور e^x برابر است با:

$$۱ + x + \frac{x^۲}{۲!} + \frac{x^۳}{۳!} + ... + \frac{x^n}{n!} + ...$$

بنابراین

$$e^{-x} = ۱ - x + \frac{x^۲}{۲!} - \frac{x^۳}{۳!} + ... + \frac{x^{۲n}}{(۲n)!} - \frac{x^{۲n+۱}}{(۲n+۱)!}...$$

از طرفی چون $۰ \leqslant x \leqslant ۱$

$$x^{۲n} \geqslant x^{۲n+۱} \rightarrow \frac{x^{۲n}}{(۲n)!} \geqslant \frac{x^{۲n+۱}}{(۲n)!} \geqslant \frac{x^{۲n+۱}}{(۲n+۱)!}$$

بنابراین $۱ - x \leqslant e^{-x}$.

قضیه ۳-۴-۸. اگر الگوریتم $n(n-۱)$ بار تکرار شود ، آنگاه احتمال شکست حداکثر ۰.۱۴ است *[۱۷]*.

اثبات:بنا به قضیه ۳-۴-۶ درهر بار اجرای الگوریتم احتمال شکست حداکثر $۱ - \frac{۲}{n(n-۱)}$ است. و بنا برلم ۳-۴-۷

$$۱ - \frac{۲}{n(n-۱)} \leqslant e^{-\frac{۲}{n(n-۱)}} \rightarrow (۱ - \frac{۲}{n(n-۱)})^{(n(n-۱))} \leqslant e^{-۲} = ۰/۱۴$$

بنابراین اگر الگوریتم را $n(n-۱)$ بار تکرار کنیم ، آنگاه احتمال شکست حداکثر ۰.۱۴ است.

۳-۵ سایر روش ها

مسئله برش کمینه را می توان به عنوان یک مسئله ی خوشه بندی نیز درنظر گرفت و از روشهای خوشه بندی نیز برای حل آن استفاده نمود. کلاسترینگ یا خوشه بندی از جمله الگوریتم های قطعه بندی به حساب می آید. الگوریتم خوشه بندی اطلاعاتی را که ویژگی های نزدیک به هم و مشابه دارند را در دسته های جداگانه که به آن خوشه گفته می شود قرار می دهد.هدف اصلی در خوشه بندی تقسیم بندی اشیاء به گونه ای است که بیشترین شباهت در یک گروه و بیشترین تفاوت با اشیاء گروه های دیگر را دارا باشد. بعنوان تعریف ساده تر می توان گفت که اشیاء در خوشه مخصوص خود دارای بیشترین شباهت و در برابر اشیای متعلق به خوشه های دیگر دارای بیشترین تفاوت هستند.

خوشه بندی داده ها را از هم جدا می کند و هر خوشه داده های مخصوص خود را دارد و از تداخل داده در خوشه جلوگیری می شود. البته خوشه بندی فازی جدا از مسئله فوق می باشد و اجازه می دهد که یک شیء متعلق به چند گروه وابسته باشد.

به جز الگوریتم های مشهور مورد استفاده در این حوزه مانند الگوریتم K-means الگوریتم های فرابتکاری نیز برای حل این مسائل بکار گرفته شده اند.

الگوریتم های فراابتکاری، یکی از انواع الگوریتم های بهینه سازی تقریبی هستند که دارای راهکارهای خروج از بهینه محلی می باشند و قابل کاربرد در طیف گسترده ای از مسائل هستند.

در ادامه الگوریتم K-means و دو روش فراابتکاری «شبیه سازی تبریدی» و «جستجوی ممنوعه (تابو)» که در این پایان نامه برای حل مسئله برش کمینه بکار گرفته شده اند بیان خواهند شد.

۳-۵-۱ الگوریتم K-means

الگوریتم K-means یکی از روش های خوشه بندی ساده و سریع است [۷]. این الگوریتم دارای یک پارامتر به نام K است که تعداد خوشه هایی که باید بدست آید را مشخص می کند. الگوریتم K-means پایه به صورت زیر است:

k داده را به عنوان مرکز خوشه انتخاب می کنیم. سپس فواصل بقیه داده ها با مرکز خوشه ها را تعیین می کنیم و داده هایی که به مرکز هر خوشه نزدیکترند را در آن خوشه قرار می دهیم. میانگین هر خوشه را به عنوان مرکز جدید خوشه انتخاب می کنیم . این مراحل را تا زمانی ادامه می دهیم که خوشه ها بدون تغییر باقی بمانند.

به طور معمول، مرکز خوشه های اولیه به صورت تصادفی از میان نمونه های اولیه گزینش می شوند. به همین دلیل، مرکز خوشه های اولیه در دو خوشه بندی مستقل K-means می توانند متفاوت باشند. این موضوع موجب می شود که خوشه های بجا مانده از دو اجرای مختلف K-means با هم متفاوت باشند. بنابراین همواره به بهینه سراسری نمی رسد، ممکن است به بهینه محلی برسد. در الگوریتم K-means می توان از معیار های فاصله ی گوناگون بهره گرفت و خوبی یا بدی بکارگیری آن معیار بستگی به نوع داده هایی دارد که باید خوشه بندی شوند.

اگر یک ماتریس $m * n$ را به عنوان ورودی به K-means بدهیم K-means این ماتریس را m داده با n خصوصیت در نظر می گیرد و اگر K را دو درنظر بگیریم داده ها را به دو دسته تقسیم می کند، با این شرط که اعضایی در یک دسته اندکه خصوصیات آنها به یکدیگر نزدیکتر باشد یعنی سطر متناظر با آنها در ماتریس ورودی شباهت بیشتری به یکدیگر داشته باشد.

اگر m دانشجو n درس را انتخاب واحد کرده باشند، ماتریس A_{mn} را به این صورت در نظر می گیریم ، $a_{ij} = ۱$ است، اگر دانشجوی i درس j را انتخاب کرده باشد و $a_{ij} = ۰$ است، اگر دانشجوی i درس j را انتخاب نکرده

الگوریتم ۳-۵ الگوریتم$K-means$ [۷]

ورودی: خصوصیاتnداده وK تعداد دسته ها

خروجی: k دسته که داده های هر دسته از نظر شاهت به هم نزدیک و از دسته های دیگر دورند.

۱: K داده را به عنوان مرکز خوشه انتخاب می کنیم .

۲: مرحله سوم تا پنجم را تا رسیدن به عدم تغییر در خوشه ها تکرار می کنیم .

۳: فواصل بقیه داده ها با مرکز خوشه ها را تعیین می کنیم.

۴: داده هایی که به مرکز هر خوشه نزدیکترند در آن خوشه قرار می گیرند

۵: میانگین هر خوشه را به عنوان مرکز جدید خوشه در نشر می گیریم.

جدول ۳-۴: انتخاب واحد ۸ دانشجو

شماره دانشجو/کد درس	A	B	C	D	E	F	G	H
۱	۱	۱	۱	۰	۱	۰	۰	۰
۲	۰	۰	۰	۰	۱	۱	۱	۱
۳	۱	۱	۰	۱	۰	۰	۱	۱
۴	۰	۱	۱	۰	۰	۰	۰	۰
۵	۱	۱	۰	۰	۱	۰	۱	۱
۶	۰	۱	۱	۱	۱	۱	۱	۰
۷	۰	۱	۰	۰	۰	۱	۱	۱
۸	۱	۰	۱	۱	۰	۱	۰	۰

باشد. حال اگر ماتریس P_{mn} که میزان شباهت دانشجویان به یکدیگر را نشان بدهد، آن را به این صورت در نظر می گیریم ، در ابتدا تمام مولفه های ماتریس صفر است . اگر دانشجویi, jهردو درس k را انتخاب کرده باشند یا هردو این درس را انتخاب نکرده باشند، $1/m$ به مقدار $p(i, j)$ اضافه می شود. اگر بخواهیم بااستفاده از الگوریتم K-means در$MATLAB$ دانشجویان را براساس شباهت درسهای انتخاب شده توسط آنها خوشه بندی کنیم، باید از ماتریسAبه عنوان ورودی استفاده کنیم .چون K-means خود شباهت ها را محاسبه می کند و بعد خوشه بندی را انجام می دهد. اگر از ماتریس P به عنوان ورودی استفاده کنیم ، جوابهای درستی به ما نمی دهد.چون در این صورت K-means شباهت ها را خصوصیت در نظر می گیرد و دانشجویانی را در یک دسته قرار می دهد که ازنظر شباهت با دانشجوان دیگر شبیه باشند. مثلا دانشجویi و jدر یک دسته اند اگر هردو به یک اندازه به دانشجویk شبیه باشند.

مثال ۳-۵-۱. باتوجه به جدول ۳-۴ ماتریسAو Pبه صورت زیر است

$$A = \begin{pmatrix} ۱ & ۱ & ۱ & ۰ & ۱ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۰ & ۱ & ۱ & ۱ & ۱ \\ ۱ & ۱ & ۰ & ۱ & ۰ & ۰ & ۱ & ۱ \\ ۰ & ۱ & ۱ & ۰ & ۰ & ۰ & ۰ & ۰ \\ ۱ & ۱ & ۰ & ۰ & ۱ & ۰ & ۱ & ۱ \\ ۰ & ۱ & ۱ & ۱ & ۱ & ۱ & ۱ & ۰ \\ ۰ & ۱ & ۰ & ۰ & ۰ & ۰ & ۱ & ۱ \\ ۱ & ۰ & ۱ & ۱ & ۰ & ۱ & ۰ & ۰ \end{pmatrix}$$

$$P = \begin{pmatrix} ۰ & ۰/۲۵ & ۰/۳۷۵ & ۰/۷۵ & ۰/۶۲۵ & ۰/۵ & ۰/۳۷۵ & ۰/۵ \\ ۰/۲۵ & ۰ & ۰/۳۷۵ & ۰/۲۵ & ۰/۶۲۵ & ۰/۵ & ۰/۶۲۵ & ۰/۲۵ \\ ۰/۳۷۵ & ۰/۳۷۵۵ & ۰ & ۰/۳۷۵ & ۰/۷۵ & ۰/۳۷۵ & ۰/۷۵ & ۰/۳۷۵ \\ ۰/۷۵ & ۰/۲۵ & ۰/۳۷۵ & ۰ & ۰/۳۷۵ & ۰/۵ & ۰/۶۲۵ & ۰/۵ \\ ۰/۶۲۵ & ۰/۶۲۵ & ۰/۷۵ & ۰/۳۷۵ & ۰ & ۰/۳۷۵ & ۰/۷۵ & ۰/۱۲۵ \\ ۰/۵ & ۰/۵ & ۰/۳۷۵ & ۰/۵ & ۰/۳۷۵ & ۰ & ۰/۳۷۵ & ۰/۵ \\ ۰/۳۷۵ & ۰/۶۲۵ & ۰/۷۵ & ۰/۶۲۵ & ۰/۷۵ & ۰/۳۷۵ & ۰ & ۰/۱۲۵ \\ ۰/۵ & ۰/۲۵ & ۰/۳۷۵ & ۰/۵ & ۰۱۲۵ & ۰/۵ & ۰/۱۲۵ & ۰ \end{pmatrix}$$

اگرماتریسAرا به عنوان ورودی به K-means بدهیم برای k=۲ دو دسته ی خروجی به صورت زیر است :

$c_۱$ ۲ ۱ ۲ ۲ ۲ ۱ ۲ ۱

$c_۲$ ۲ ۱ ۱ ۲ ۱ ۲ ۱ ۲

اگرماتریسP را به عنوان ورودی به$K-means$بدهیم برای k=۲ دو دسته ی خروجی به صورت زیر است:

$c_۳$ ۲ ۲ ۲ ۲ ۱ ۱ ۱ ۱

واگربرش کمینه را برای ماتریسPبه دست آوریم دو دسته ی خروجی به صورت زیر است:

$c_۴$ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۰

اگر مجموع ظرفیت یالهای بین دو مجموعه را برای $c_۱, c_۲, c_۳, c_۴$ روی P به دست آوریم آنگاه داریم:

$c_۱$ ۵/۸۷۵

$c_۲$ ۵/۷۵

$c_۳$ ۸/۲۵

$c_۴$ ۲/۳۷۵

اگر بخواهیم برش کمینه را برای ماتریس P به دست آوریم ، استفاده از روش K-means خوب نیست .البته اگر ماتریس خصوصیات را داشته باشیم و K-means را روی آن اجرا کنیم، دسته های مشخص شده به دسته های برش کمینه نزدیکترند.

اگر بخواهیم دانشجویان را به دو گروه تقسیم کنیم که دانشجویان هر دسته دروس مشترک بیشتری داشته باشند ودروس مشترک بین دو گروه مینیمم شود.استفاده از روش K-means بهتر از روش برش کمینه است .چون در برش کمینه فقط شباهت دروس بین دو دسته را مینیمم میکندو شباهت دروس مشترک هر دسته را در نظر نمیگیرد.مثلا دانشجو ۸و۶ در ۴ درس از ۸ درس یعنی در نیمی از دروس شبیه اند ولی وقتی برش کمینه را برای P به کار میبریم این دو دانشجو در یک دسته نیستند.درحالی که دانشجوی ۶و۳ با ۰.۳۷۵ شباهت در یک دسته اند.

۳-۵-۲ شبیه سازی تبریدی

شبیه سازی تبریدی یک الگوریتم مبتنی بر جستجوی محلی [۷] می باشد که قادر است خود را از دام بهینه محلی رها کند. راحتی به کار گیری، همگرایی و استفاده از حرکات خاص جهت دوری از قرارگیری در دام بهینه محلی از جمله خصوصیاتی است که باعث جلب توجه محققان به آن شده است. ایده اصلی شبیه سازی تبرید بر گرفته از فرآیند آنیل کردن یا سرد سازی فلزات است که اولین بار توسط متروپولیس [۸] و همکارانش در سال ۱۹۵۳ ارائه شد [۱۸] و بعدها توسط کیرک پاتریک [۹] و همکارانش (۱۹۸۳) جهت حل مسایل بهینه سازی ترکیباتی [۱۰] به کار برده شد. [۱۹]. تشابهی که بین مسأله ترکیبی بهینه سازی و یک جسم فیزیکی وجود دارد بر پایه دو مطلب زیر است:

۱. جوابهای ممکن مسأله بهینه سازی ترکیبی با وضعیتهای جسم متناظر است.

---

[۷] local search
[۸] Metropolis
[۹] Kirkpatrick
[۱۰] combinatorial optimization

۲. مقدار تابع هدف به ازای یک جواب ممکن، با مقدار انرژی یک جسم، متناظر است.

همانگونه که میدانیم، وقتی یک جسم را حرارت میدهیم کم کم منبسط می شود و در اثر انبساط و گرما از نظر فیزیکی در سطح انرژی بالایی قرار میگیرد که در این حالت در واقع اتم ها به نظر خیلی بی نظم می رسند، ولی وقتی جسم را به آرامی سرد میکنیم، آرایش اتمها به ظاهر منظم تر میشود و اصطلاحاً گفته می شود که سیستم از نظر فیزیکی در سطح انرژی پایینی قرار دارد. الگوریتم شبیه سازی تبرید که درخیلی از مواقع الگوریتم انجماد تدریجی نیز نامیده می شود، از پدیده انجماد فیزیکی فوق الگو گرفته است [۲۰].

روش کار الگوریتم شبیه سازی تبریدی

فرض کنید فضای جواب Aیک مجموعه متناهی از تمام جواب ها و fمقدار تابع هدف، برای هر یک از عناصر باشد. هدف پیدا کردن یک جواب یا حالتی مانند $i \in A$ است به طوری که $f(i)$ روی A کمینه باشد.
از آن جایی که تبرید شبیه سازی شده، یک الگوریتم مبتنی بر جستجوی محلی است ، لذا ابتدا الگوریتم جستجوی محلی را بررسی می کنیم. الگوریتم با یک جواب اولیه شروع می کند و از بین همسایگی های این جواب، آن را که جواب حاضر را بهبود دهد، انتخاب می کند و این همسایگی جدید به جای جواب اولیه قرار می گیرد. این فرآیند ادامه می یابد تا اینکه دیگر هیچ یک از همسایگی های جواب جاری نتواند مقدار تابع هدف را بهبود بخشد. در این حالت جواب جاری یک بهینه محلی است. یکی از راههایی که می توان برای بهبود این روش استفاده کرد این است که الگوریتم را چندین بار با جوابهای اولیه مختلف اجرا کنیم و در نهایت بهینه محلی را انتخاب کنیم.
شبیه سازی تبرید به جای استفاده ازاین استراتژی، سعی می کند در یک بهینه محلی قرار نگیرد و بنابراین برای این کار، گاهی اوقات جواب هایی که مقدار تابع هدف را افزایش داده می پذیرد. قبول یا رد این جواب به دنباله ای از اعداد تصادفی وابسته به یک کنترل احتمالی بستگی دارد. به علاوه اجتناب از جواب بهینه محلی، بستگی به برنامه آنیلینگ (گرم کردن و سپس به تدریج سرد کردن) ، انتخاب دمای اولیه، تعداد تکرارهای لازم در هر دما و مقدار کاهش دما در هر مرحله دارد که درادامه به آنها اشاره خواهد شد. الگوریتم شبیه سازی تبرید در بسیاری از مسائل بهینه سازی ترکیبی کاربرد دارد.
برای یک جواب یا پیکره بندی مشخص یک سری محدود از حرکت یا تغییرات ابتدایی در نظر گرفته می شود. اگر یک تغییر شکل، منجر به کاهش تابع هدف یا انرژی گردد، آن تغییر شکل پذیرفته می شود و در صورتی که این تغییر شکل منجر به افزایش تابع هدف یا انرژی به اندازهE گردد ، تغییر شکل با احتمال $e^{(\frac{-\Delta E}{T})}$ پذیرفته می شود. که $P(\Delta E, T) = e^{(\frac{-\Delta E}{T})}$ تابع پذیرش نام دارد.این پذیرش از طریق تولید یک عدد تصادفی با توزیع یکنواخت در فاصله بین ۰و۱ و مقایسه آن با تابع تعریف شده انجام می گردد. در صورتی که مقدار عدد به دست آمده کوچکتر از تابع تعریف شده باشد، تغییر شکل پذیرفته می شود. هر جواب از روی جواب پذیرفته شده قبلی

خود تولید می گردد و با تکرار تولید جواب ها و پذیرش آن ها با استفاده از قاعده متروپلیس، یک توالی از جواب ها بوجود می آید که زنجیره مارکوف را بوجود آورده اند. پایان یافتن این زنجیره با طول محدود و کافی را میتوان به رسیدن یک سیستم فیزیکی به تعادل ترمودینامیکی در یک دمای مشخص تشبیه کرد.

انتخاب پارامترها

برای به کار بردن شبیه سازی تبرید جهت یک مساله خاص تعداد زیادی تصمیم باید گرفته شود .این تصمیم ها را می توان به دو گروه تقسیم بندی کرد، گروه اول تصمیم های کلی که مرتبط با پارامترهای خود الگوریتم هستند که شامل عواملی از قبیل دمای اولیه، قاعده کاهش دما و قاعده توقف می باشد. گروه دوم تصمیم های مربوط به یک مساله خاص شامل انتخاب فضای جواب موجه، ساختار همسایگی و شکل تابع هزینه می باشد. هر دو گروه تصمیم ها باید با دقت گرفته شوند و ثابت شده که موثر بر سرعت الگوریتم و کیفیت جواب های بدست آمده هستند.

دمای اولیه

تعیین دمای اولیه T_0تاثیر زیادی در انتخاب کردن و یا انتخاب نکردن جابجایی ها دارد و تاکنون روش دقیقی برای آن بوجود نیامده است. اگر T_0(دمای اولیه) خیلی بزرگ باشد آن گاه $e^{(\frac{-\Delta E}{T})}$ نزدیک ۱ می شود و در نتیجه هر جابجایی که تابع هدف را بهبود ندهد، نیز پذیرفته می شود و به نوعی الگوریتم شبیه سازی تبرید مانند الگوریتم جستجوی تصادفی عمل می کند [۲۱]. حال اگر T_0 مقدار کوچکی داشته باشد، آنگاه انتخاب یا پذیرش جوابهای خیلی بد کاهش می یابد. در واقع پایین بودن بیش از حد دمای اولیه، باعث توقف در جوابهای محلی می شود، زیرا در ابتدای الگوریتم، اجازه جستجو را از خود می گیریم و به عبارت دیگر با این کار احتمال پذیرش جواب با سطح انرژی بالا کمتر می شود، آنگاه سرعت الگوریتم، بسیار کاهش می یابد. بنابراین برای جلوگیری از این مشکلات، دمای اولیه T_0را نسبتا بزرگ انتخاب می کنیم و بعد از هر تکرار، آن را کاهش می دهیم. همچنین پایین بودن بیش از حد دمای نهایی، باعث اتلاف وقت در انتهای الگوریتم میشود. در واقع در انتهای الگوریتم، از یک دمای خاص کمتر، دیگر جوابهای جدید پیشرفت چندانی در پایین آوردن سطح انرژی نمی کنند [۲۲].

تابع تبرید

به طور کلی بین کیفیت نتایج و سرعت انجماد رابطه قوی وجود دارد مقدار دما همواره مقدار مثبت بوده و زمانی که تعداد تکرار ها به سمت بی نهایت میل کند، مقدار دما نیز به سمت صفر میل می کند. در این جا دو روش کاهش درجه حرارت که به طور وسیعی در الگوریتم هایSAبه کار می روند، ذکر می شود [۲۳] :

- **کاهش بسیار آرام** :این روش اولین بار توسط لاندی و میس ارائه شد. طبق این روش دما خیلی آهسته و طبق فرمول

$$T_{i+۱} = \frac{T_۰}{۱ + \beta T_i}$$

کاهش می یابد کهβیک مقدار کوچک است.

- **قاعده هندسی** : این روش به صورت

$$T_{i+۱} = \alpha T_i$$

معرفی می گردد و در آن $۰ < \alpha < ۱$ ضریب ثابتی می باشد. تجربه نشان داده است که مقادیر نسبتا بزرگ α بهترین نتیجه را دارد و اکثر موفقیت های گزارش شده از مقادیر بین۰/۸و۰/۹۹استفاده کرده اند [۲۴].

قاعده توقف

در الگوریتم شبیه سازی تبرید، معمولاً از یکی از معیار های زیر برای شرط توقف استفاده می شود .

- رسیدن به درجه حرارت نهایی است.
- اجرای یک تعداد تکرار معین که در آ ن ها هیچ بهبودی در بهترین جواب یافت شده ، دیده نشود.

پیاده سازی شبیه سازی تبریدی برای برش کمینه

تاکنون الگوریتم شبیه سازی تبریدی که از جمله الگوریتم های فراابتکاری حل مسائل بهینه سازی است برای حل مسئله برش کمینه بکار گرفته نشده است. در مقاله [۲۵] چگونگی بکارگیری این شیوه ی حل مسئله برای مواجهه با مسئله ی برش کمینه بیان و مزایای آن در قالب اجرای مثالهای مختلف در فصل ۴ بیان شده است.
بخش اصلی هر روش مبتنی بر شبیه سازی تبریدی، نحوه نمایش یک جواب ممکن و تعریف همسایگی یک نقطه در فضای جواب است. در شیوه ی پیشنهادی هر نقطه در فضای جواب با یک آرایه با تعداد عناصر برابر با تعداد رئوس گراف نمایش داده می شود که اگر عنصر iام برابر یک باشد، به معنی تعلق به دسته ی اول و اگر برابر با صفر باشد به معنی تعلق به گروه دوم خواهد بود. جواب اولیه به صورت تصادفی ایجاد می شود. برای ایجاد نقطه همسایه، روشهای مختلفی مورد بررسی قرار گرفت که روشی که بهترین نتایج را داشت به شرح زیر است:
برای یک جواب مفروض، یک گروه به صورت تصادفی انتخاب می شود (مثلا گروه اول)، سپس رأسی که مجموع

وزن لبه های بین آن و سایر اعضای این گروه کمتر از مجموع وزن لبه های بین این رأس و رئوس گروه دوم باشد، به گروه دوم منتقل می شود. عمل انتقال رأس به شرطی انجام می شود که هیچ یک از دو گروه تهی نشوند.

۳-۵-۳ جستجوی ممنوعه

الگوریتم جستجوی ممنوعه ، (TS) یک الگوریتم بهینه سازی فراابتکاری است که برای اولین بار در سال ۱۹۸۶ توسط گلوور [۱۱]معرفی شد [۲۶]. در سال ۱۹۹۷، اولین کتابی که کاملاً به جستجوی ممنوعه اختصاص داشت توسط گلوور و لاگونا منتشر شد [۲۷]. واژۀ تابو از تُنگان [۱۲] زبان مردم جزایر پلینزی [۱۳]در اقیانوس آرام گرفته شده است. این واژه به معنای شیء مقدسی است که به دلیل قداست نباید آن را لمس کرد [۲۸]. بر اساس واژه نامه ی وبستر [۱۴]، امروزه این واژه در معنای «ممنوعیت ایجاد شده به دلیل فرهنگ اجتماعی برای ایجاد اقدام حفاظتی» یا «ممنوعیت چیزی که دارای ریسک است»، به کار می رود [۲۹]. معنای اخیر واژۀ تابو، با تکنیک جستجوی ممنوعه کاملاً سازگار است. ریسکی که در الگوریتم جستجوی ممنوعه از آن اجتناب می شود، خطر مسیرهای نامناسب است [۲۸].

ساختار کلی جستجوی ممنوعه

الگوریتم جستجوی ممنوعه ابتدا از یک جواب اولیه شروع به حرکت می کند. سپس الگوریتم بهترین جواب همسایه را از میان همسایه های جواب فعلی انتخاب می کند. در صورتی که این جواب در لیست ممنوعه [۱۵] قرار نداشته باشد، الگوریتم به جواب همسایه حرکت می کند؛ در غیراین صورت الگوریتم معیاری به نام معیار تنفس [۱۶]را چک خواهد کرد. بر اساس معیار تنفس اگر جواب همسایه از بهترین جواب یافت شده تا کنون بهتر باشد، حتی اگر آن جواب در لیست ممنوعه باشد، الگوریتم به آن حرکت خواهد کرد.

پس از حرکت الگوریتم به جواب همسایه، فهرست ممنوعه بروزرسانی می شود؛ به این معنا که حرکت قبل که بوسیله ی آن به جواب همسایه حرکت کردیم، در لیست ممنوعه قرار داده می شود تا از بازگشت مجدد الگوریتم به آن جواب و ایجاد دور جلوگیری شود. در واقع لیست ممنوعه ابزاری در الگوریتم جستجوی ممنوعه است که توسط آن از قرار گرفتن الگوریتم در بهینه ی محلی جلوگیری می شود. پس از قرار دادن حرکت قبلی درلیست ممنوعه، تعدادی از حرکت هایی که قبلاً در لیست ممنوعه قرار گرفته بودند از فهرست خارج می شوند. مدت زمانی که حرکت ها در لیست ممنوعه قرار می گیرند ،توسط یک پارامتر که اعتبار تابو [۱۷] نام دارد، تعیین می شود. پس از انجام هر تکرار، از زمان ممنوعیت هر یک از اعضای لیست ممنوع کاسته می شود. هر چقدر زمان

[۱۱] Glover
[۱۲] Tongan
[۱۳] Polynesia
[۱۴] Webester
[۱۵] tabu list
[۱۶] aspiration criteria
[۱۷] tabu tenure

ممنوعیت اعضای این لیست بیشتر باشد ، فرآیند جستجو محدودتر شده و تنوع نقاط تحت بررسی کمتر می گردد. با توجه به تاثیر حرکت های ممنوع در انحراف مسیر جستجو، یک معیار ایده آل بر مبنای بهترین پاسخ یافته شده تعریف می گردد. حرکت از جواب فعلی به جواب همسایه تا جایی ادامه می یابد که شرط خاتمه دیده شود. شرط های خاتمه متفاوتی می توان برای الگوریتم در نظر گرفت. به طور مثال محدودیت تعداد حرکت به جواب همسایه می تواند یک شرط خاتمه باشد. معیارهای توقف معمولاً شامل یک یا چند مورد از موارد زیر می باشد:

۱. بهترین توالی به دست آمده از الگوریتم به اندازه کافی به حد پایین مقدار بهینه مسئله نزدیک باشد.

۲. تعداد مراحل(تکرارهای) بدون ایجاد بهبود در توابع هدف الگوریتم از مقدار معینی بیشتر شود.

۳. زمان حل الگوریتم و یا کل تعداد مراحل آن از مقدار مشخصی بیشتر شود.

دو رکن اساسی جستجوی ممنوع، فضای جستجو و ساختار همسایگی است. فضایی از همه جواب های ممکن که در طی جستجو می تواند در نظر گرفته شود، فضای جستجو نامیده می شود. و مجموعه جواب هایی که با یک انتقال یا حرکت از جوابsبدست می آید، همسایگی نامیده می شود [۳۰].

استراتژی های پیشرفته ی جستجوی ممنوعه

ساختار کلی جستجوی ممنوعه اغلب جوابگوی مسائل بزرگ نیست. بنابراین به منظور افزایش قدرت الگوریتم از استراتژی های زیر که معروف به استراتژی های پیشرفته جستجوی ممنوعه هستند استفاده می شود [۳۱]:

استراتژی فهرست کاندید [۱۸]:در یکTSعادی، برای حرکت از یک جواب فعلی به یک جواب همسایه، باید مقدار تابع هدف برای هر عنصر از همسایه ها ارزیابی شود. این کار می تواند از لحاظ محاسباتی بسیار هزینه بر باشد.اگر به جای آن که تمامی همسایه ها بررسی شود، تنها یک زیرمجموعه ی تصادفی از همسایه ها در نظر گرفته شود، که در نتیجه هزینه ی محاسباتی به طور قابل ملاحظه ای کاهش می یابد. انتخاب زیرمجموعه ای از جوابهای همسایه به صورت تصادفی، می تواند به عنوان یک مکانیزم ضد چرخه عمل کند؛ این کار اجازه
می دهد که از فهرست ممنوعه ی کوچکتری نسبت به کل همسایگی، استفاده شود. البته باید در نظر داشت که این کار یک عیب مهم دارد و آن احتمال از دست دادن جوابهای خوب است، بنابراین احتمال هایی را نیز می توان به کار برد تا معیارهای ممنوعه فعال شود.

استراتژی تقویت [۱۹] : استراتژی تقویت به معنای یافتن حرکت های خوب و افزایش انجام آن حرکت ها در الگوریتم است. تقویت، در بسیاری از پیاده سازی هایTSاستفاده می شود، اما همیشه ضروری نیست، زیرا حالت های

[۱۸] candidate list strategy
[۱۹] intensification strategy

بسیاری وجود دارد که در آنها جستجوی معمولی کفایت می کند.

استراتژی تنوع بخشی [۲۰]:روش های مبتنی بر جستجوی محلی، آن قدر محلی هستند که زمان زیادی و یا تمامی زمان خود را در بخش محدودی از فضای جستجو صرف می کنند. نتیجه ای که از این واقعیت می توان گرفت، این است که هر چند جواب های خوبی به وسیله ی این روشها به دست می آید، اما ممکن است جستجو از اکتشاف مناطق بهتر باز بماند و بنابراین به جواب هایی برسد که از جواب بهینه، بسیار دور هستند. تنوع بخشی، یک مکانیزم الگوریتمیک است که برای حل این مشکل تلاش می کند. برای انجام این کار، تنوع بخشی، جستجو را مجبور می کند به سوی مناطقی که تا کنون کشف نشده، حرکت کند.

مجوز دادن به جوابهای نشدنی: در حالت هایی که محدودیت های مسئله بسیار محدود کننده باشند و از جستجوی موثر فضای جواب جلوگیری کنند از این استراتژی استفاده می شود. طی این استراتژی محدودیت های مسئله آزاد شده و بجای آنها یک مقدار جریمه به تابع هدف اضافه می شود.

معیار تنفس: بخش مهمی از انعطاف پذیری در الگوریتم تابو بوسیله معیارتنفس تعریف می شود. ممنوع بودن یک جواب امری مطلق نیست و در صورتی که شرایط خاصی اتفاق افتد، می تواند نادیده گرفته شود یعنی جواب ممنوعی که از تمام جواب هایی که تاکنون یافت شده بهتر است، می تواند موجه در نظر گرفته شود .

ساختار حافظه

یکی از نکات مهم در موردTSاستفاده از حافظه کوتاه مدت وبلند مدت است. ساختار حافظه کوتاه مدت در رویکرد $TabuSearch$ به طور معمول به بخشی از حافظه اطلاق می شود که یا برای پیاده سازی استراتژی تقویت به کار برده می شود و یا برای انجامLSمورد نیاز است. از آن جمله می توان حافظه اختصاص یافته به لیست ممنوع $(TabuList)$ را به عنوان حافظه ای که برای عدم تکرار به کار برده می شود، در این دسته قرار داد. حافظه بلند مدت، درTSبا هدف جهت دادن به روند جستجو پیاده سازی می شود. به عبارت بهتر، معمولا این حافظه پیاده سازی استراتژی تنوع است. تذکر این نکته لازم است که در TSبرای افزایش پراکندگی جستجو و هدایت آن، معمولا راه حل ها از یک مساله به مساله دیگر تفاوت می کند.

پیاده سازی جستجوی ممنوعه برای برش کمینه

تاکنون الگوریتم جستجوی ممنوعه که از جمله الگوریتم های فراابتکاری حل مسائل بهینه سازی است برای حل مسئله برش کمینه بکار گرفته نشده است. در مقاله [۳۲] چگونگی بکارگیری این شیوه ی حل مسئله برای مواجهه با مسئله ی برش کمینه بیان و مزایای آن در قالب اجرای مثالهای مختلف در فصل ۴ بیان شده است.

[۲۰] diversification strategy

جواب اولیه آرایه به اندازه تعداد رئوس با درایه های صفر ویک است که درایه هایی که مقدار یک دارند، متناظر با مجموعه اول اند .یال با کمترین هزینه را انتخاب می کنیم و راسها اگر به مبدا یال نزدیکتر باشند .در مجموعه اول و در غیر این صورت در مجموعه دوم قرار می گیرند. جواب بهینه را برابر با این آرایه قرار می دهیم.
در هر مرحله یک عدد تصادفی بین صفر ویک انتخاب می شود، اگر کمتر از نیم بود، آنگاه تغییر روی یکی از اعضای مجموعه اول انجام می شود بااین شرط که این عضو در لیست ممنوعه نباشد و اتصال این عضو به مجموعه اول کمتر از اتصال به مجموعه دوم باشدبیت متناظر با این عضو از یک به صفر تغییر میکند واگر عدد تصادفی بیشتر از نیم بود، آنگاه تغییر روی یکی از اعضای مجموعه دوم انجام می شود . و این عضو تا $|V|/۲$ مرحله در لیست ممنوعه قرار می گیرد . اگر نقطه بدست آمده باعث بهبود مقدار تابع هدف شود، آنگاه آن را جایگزین جواب بهینه می کنیم.

۳-۶ جمع بندی

در این فصل الگوریتم های غیر مبتنی بر جریان را برای مسئله برش مینیمم بیان کردیم.بررسی تمام حالات، روش سیمپلکس و الگوریتم استور واگنر مانند روشهای فصل ۲ جواب دقیق و الگوریتم کارگر ، شبیه سازی تبریدی و جستجوی ممنوعه جواب های تقریبی را به دست می آورند.
در فصل ۴ روش سیمپلکس را با الگوریتم ادموندز کارپ که در فصل ۲ ذکر شد،مقایسه می کنیم. نتیجه در شکل ۴-۱ آمده است.الگوریتم ادموندز کارپ زمان اجرای بهتری دارد.
سپس الگوریتم کارگر و SA ونیز الگوریتم کارگر و TS را روی گراف های وزندار با هم مقایسه می کنیم.نتایج نشان می دهد ، الگوریتم کارگر روی گراف های وزندار کارایی خوبی ندارد، زمان اجرای SA و TS تقریبا ثابت است درحالی که زمان اجرای الگوریتم کارگر برای گرافهایی با تعداد راس بالا زیاد است ونیز درصد خطای SA و TS بسیار پایین است ،بنابراین آنها را از نظر زمان اجرا با الگوریتم ادموندز کارپ مقایسه کرده ایم.

فصل ۴

پیشنهادات

در این فصل روشهایی که در این پایان نامه به عنوان شیوه های جدید حل مسئله برش کمینه پیشنهاد شده اند به همراه نتایج پیاده سازی برخی روشهای موجود بیان خواهد شد. قابل ذکر است که دو روش پیشنهادی مبتنی بر شبیه سازی تبریدی و جستجوی ممنوعه در قالب مقالات [۲۵] و [۳۲] ارائه شده است.

۴-۱ سیستم پیاده سازی شده

برنامه های نوشته شده در MATLAB پیاده سازی شده اند. برای تمام الگوریتم ها نیاز به یک گراف با اندازه برش کمینه از قبل دانسته بود که چنین پایگاه داده ای وجود ندارد. لذا در این پایان نامه با یک برنامه گرافهایی تصادفی که اندازه برش کمینه آنها از قبل مشخص باشد و بتوان دقت الگوریتم ها را سنجید تولید می شود. برنامه های نوشته شده در پیوست الف آمده است. این گرافها را به صورت زیر تولید کردیم :

مجموعه ی رئوس با تعداد زوج n را در نظر می گیریم. رئوس را به دو دسته ۱ تا $\frac{n}{2}$ و $\frac{n}{2}$ تا n تقسیم می کنیم. بین رئوس هر یک از این مجموعه ها اتصالات قوی ، حداقل با وزن ۱ و بین دو مجموعه اتصالات ضعیفی داریم به گونه ای که مجموع وزن تمام یال های بین این دو مجموعه از مجموع اتصالات هر راس کمتر باشد. در نتیجه همیشه برش کمینه در این گرافها با تقسیم بندی بالا برابر است و مقدار برش کمینه برابر با مجموع وزن تمام یال های بین این دو مجموعه می باشد.

۲-۴ پیاده سازی روش کارگر

همان گونه در فصل قبل گفته شد ،روش کارگر برای گرافهای بدون وزن بیان شده است. به دست آوردن درخت پوشای مینیمم در یک گراف بدون وزن با به دست آوردن هر درخت پوشای تصادفی برابر است.درخت پوشای گراف با n راس $n-۱$ یال دارد.اگر از این درخت پوشای تصادفی یال آخر حذف شود ، گراف به دو مولفه تقسیم می شود. در الگوریتم کارگر نیز $n-۲$ یال به صورت تصادفی برای انقباض انتخاب می شوند . بنابراین اگر در یک گراف بدون وزن درخت پوشای مینیمم را به دست آوریم و یال آخر را حذف کنیم ، مولفه های به دست آمده با مولفه های حاصل از الگوریتم کارگر یکسان می باشد.

برای پیاده سازی الگوریتم کارگر روی گراف وزن دار G نمی توان از تولید درخت پوشای مینیمم روی همین گراف استفاده کرد. زیرا در تولید درخت پوشای مینیمم یالها به ترتیب صعودی انتخاب می شوند ودر الگوریتم کارگر باید این یالها تصادفی انتخاب شوند. بنابراین برای پیاده سازی الگوریتم کارگر روی گراف وزن دار G ، ابتدا گراف مجاورت G' را از روی G می سازیم. سپس درخت پوشا ی مینیمم را روی این گراف به دست می آوریم.

$$G'(i,j)=\begin{cases}۱ & G(i,j)\neq ۰\\ ۰ & \text{در غیر این صورت}\end{cases}$$

۱-۲-۴ درآمدی بر خطای الگوریتم کارگر در گرافهای وزن دار

بنا بر قضایای ۳-۴-۶ و ۳-۴-۸ اگر الگوریتم کارگر را برای گراف های بدون وزن به کار ببریم ، آنگاه خروجی الگوریتم با احتمال $\frac{۲}{n(n-۱)}$ برش کمینه است و با $n(n-۱)$ بار تکرار الگوریتم، احتمال شکست حداکثر ۰.۱۴ است.نشان می دهیم اگر این الگوریتم را برای گرافهای وزن دار جهتدار به کار ببریم، آنگاه خروجی با احتمال $\frac{۲}{n!(n-۱)}$ برش کمینه است. و با $n!(n-۱)$ بار تکرار الگوریتم، احتمال شکست حداکثر ۰.۱۴ است.برای گراف های وزن دار بدون جهت خروجی با احتمال $\frac{۲^{n-۱}}{n!(n-۱)}$ برش کمینه است. و با $\frac{n!(n-۱)}{۲^{n-۱}}$ بار تکرار الگوریتم، احتمال شکست حداکثر ۰.۱۴ است. یعنی برای گراف های وزن دار احتمال رسیدن به برش کمینه کمتر است واگر بخواهیم به خطای حداکثر ۰.۱۴ برسد تعداد تکرار های بیشتری لازم دارد.بنابراین الگوریتم کارگر برای گراف های وزن دار مناسب نیست.

لم ۴-۲-۱. اگر $[P,\bar{P}]$ برش مینیمم باشند، آنگاه P و $\bar{P}$ مجموعه های همبند هستند.

اثبات:(فرض خلف)اگر P همبند نباشد، آنگاه حداقل دو مولفه ی همبند $P_۱$ و $P_۲$ دارد.نشان می دهیم $[P_۱,\bar{P}\bigcup P_۲]$

برشی است که ظرفیت آن کمتر از برش $[P, \bar{P}]$ است و این با فرض تناقض دارد.

$$C(P_۱, \bar{P} \bigcup P_۲) = C(P_۱, \bar{P}) + C(P_۱, P_۲) = C(P_۱, \bar{P})$$
$$C(P, \bar{P}) = C(P_۱ \bigcup P_۲, \bar{P}) = C(P_۱, \bar{P}) + C(P_۲, \bar{P})$$
$$\Rightarrow C(P_۱, \bar{P} \bigcup P_۲) \leqslant C(P, \bar{P})$$

لم ۴-۲-۲. اگر تعداد کل یالهای گراف G را با m ، تعداد رئوس Gرا با n و تعداد کل یال های برش را با m_{cut} نشان دهیم،آنگاه $m \geqslant m_{cut} + \{n - ۲\}$

اثبات:تعداد رئوس مجموعه های P و $\bar{P}$ را با n_P و $n_{\bar{P}}$ وتعداد یالهای آنها را با m_P و $m_{\bar{P}}$ نشان می دهیم. بنا به لم ۴-۲-۱، P و $\bar{P}$ مجموعه های همبند هستند. بنابراین

$$m_P \geqslant n_P - ۱ \quad , \quad m_{\bar{P}} \geqslant n_{\bar{P}} - ۱$$
$$\Rightarrow m = m_{cut} + m_P + m_{\bar{P}} \geqslant m_{cut} + n_P + n_{\bar{P}} - ۲$$
$$\Rightarrow m \geqslant m_{cut} + n - ۲$$

قضیه ۴-۲-۳. در یک گراف وزن دار جهت دار ، اگر یک یال را تصادفی انتخاب کنیم ، آنگاه با احتمال بزرگتر مساوی $\frac{n-۲}{n(n-۱)}$ این یال در برش کمینه نیست.

اثبات:احتمال نبودن هریال در برش کمینه به صورت زیر محاسبه می شود:

$$P(step) = ۱ - \frac{m_{cut}}{m} = \frac{m - m_{cut}}{m}$$
$$\overset{\text{بنا به لم ۴-۲-۲}}{\Rightarrow} P(step) \geqslant \frac{n - ۲}{m}$$

گراف حداکثر $n(n - ۱)$ یال دارد .بنابراین $P(step) \geqslant \frac{n-۲}{n(n-۱)}$

نتیجه ۴-۲-۴. در یک گراف وزن دار بدون جهت اگر یک یال را تصادفی انتخاب کنیم ، آنگاه با احتمال بزرگتر مساوی $\frac{۲(n-۲)}{n(n-۱)}$ این یال در برش کمینه نیست.

اثبات: تعداد یال های گراف حداکثر $\frac{n(n-۱)}{۲}$ است.بنابراین $P(step) \geqslant \frac{n-۲}{m} \geqslant \frac{۲(n-۲)}{n(n-۱)}$

قضیه ۴-۲-۵. خروجی این روش در گراف وزن دار جهت دار با احتمال $\frac{۲}{n!(n-۱)}$ برش کمینه است.

اثبات: بنا به قضیه ۴-۲-۳ در مرحله اول احتمال اینکه یال تصادفی در برش کمینه نباشد،بزرگتر مساوی $\frac{(n-۲)}{n(n-۱)}$

است. فرض کنید رویداد η_iاتفاق می افتد اگر e_i یال انتخاب شده ازG_iیالی از برش کمینه نباشد . و فرض کنید n_i تعداد رئوس گراف G_i باشد. اگر تمام رویدادهای $\eta_۰, \eta_۱, ..., \eta_{n-۳}$ رخ دهند ،تمام یالها ی انتخابی بیرون از برش کمینه اندوخروجی الگوریتم برش کمینه است. در مرحلهiام احتمال اینکه تمام لبه ها ی انتخاب شده برای انقباض خارج از برش کمینه باشند برابر است با :

$$P(\eta_i|\eta_۰ \cap \eta_۱ \cap ... \cap \eta_{i-۱}) \geqslant \frac{(n_i - ۲)}{n_i(n_i - ۱)} = \frac{(n - i - ۲)}{n - i(n - i - ۱)}$$

از طرفی داریم:

$$P(\bigcap_{i=۰}^{n-۳} \eta_i) = P(\eta_۰) * P(\eta_۱|\eta_۲) * P(\eta_۲|\eta_۱ \cap \eta_۰) * ... * P(\eta_{n-۳}|\eta_۰ \cap \eta_۱ \cap ... \cap \eta_{n-۴})$$

وبنابراین احتمال اینکه خروجی الگوریتم برش کمینه باشد برابر است با :

$$P(\bigcap_{i=۰}^{n-۳} \eta_i) \geqslant \prod_{i=۰}^{n-۳} \frac{(n - i - ۲)}{n - i(n - i - ۱)} = \frac{۲}{n!(n - ۱)}$$

بنابراین احتمال همگرایی الگوریتم حداقل برابر است با: $\frac{۲}{n!(n-۱)}$.

نتیجه ۴-۲-۶. در یک گراف وزن دار بدون جهت خروجی این روش با احتمال $\frac{۲^{n-۱}}{n!(n-۱)}$ برش کمینه است.

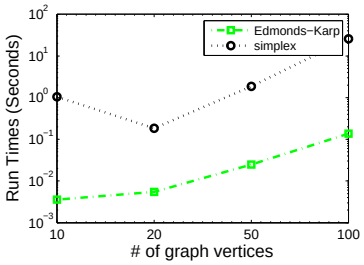
اثبات: بنا به نتیجه ۴-۲-۴ در مرحله اول احتمال اینکه یال تصادفی در برش کمینه نباشد،بزرگتر مساوی $\frac{۲(n-۲)}{n(n-۱)}$ است.بنابراین احتمال اینکه خروجی الگوریتم برش کمینه باشد برابر است با :

$$P(\bigcap_{i=۰}^{n-۳} \eta_i) \geqslant \prod_{i=۰}^{n-۳} \frac{۲(n - i - ۲)}{n - i(n - i - ۱)} = \frac{۲^{(n-۱)}}{n!(n - ۱)}$$

قضیه ۴-۲-۷. برای گراف وزن دار جهت دار اگر الگوریتم $n!(n-۱)$ بار تکرار شود ،احتمال شکست حداکثر ۰.۱۴ است.

اثبات: بنا به قضیه ۴-۲-۵ درهر بار اجرای الگوریتم احتمال شکست حداکثر $۱ - \frac{۲}{n!(n-۱)}$ است. و بنا برلم ۴-۳-۷

$$۱ - \frac{۲}{n!(n - ۱)} \leqslant e^{-\frac{۲}{n!(n-۱)}} \rightarrow (۱ - \frac{۲}{n!(n - ۱)})^{(n!(n-۱))} \leqslant e^{-۲} = ۰/۱۴$$



شکل ۴-۱: مقایسه الگوریتم ادموندز- کارپ و سیمپلکس

بنابراین اگر الگوریتم را $n!(n-1)$ بار تکرار کنیم، آنگاه احتمال شکست حداکثر ۰.۱۴ است.

نتیجه ۴-۲-۸. در یک گراف وزن دار بدون جهت خروجی اگر الگوریتم $\frac{n!(n-1)}{2^{n-1}}$ بار تکرار شود، آنگاه احتمال شکست حداکثر ۰.۱۴ است.

اثبات: بنا به نتیجه ۴-۲-۶ درهر بار اجرای الگوریتم احتمال شکست حداکثر $1 - \frac{2^{n-1}}{n!(n-1)}$ است.و بنا برلم ۳-۴-۷

$$1 - \frac{2^{n-1}}{n!(n-1)} \leqslant e^{-\frac{2^{n-1}}{n!(n-1)}} \rightarrow (1 - \frac{2^{n-1}}{n!(n-1)})^{\frac{n!(n-1)}{2^{n-1}}} \leqslant e^{-2} = 0/14$$

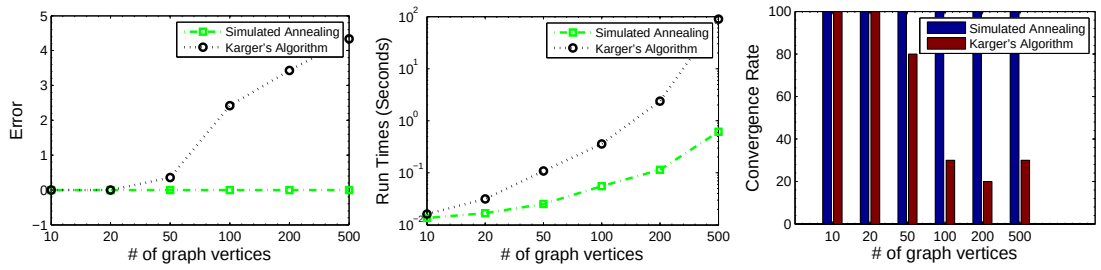
بنابراین اگر الگوریتم را $\frac{n!(n-1)}{2^{n-1}}$ بار تکرار کنیم، آنگاه احتمال شکست حداکثر ۰.۱۴ است.

۴-۳ مقایسه الگوریتم ادموندز کارپ و سیمپلکس

هر دو روش ادموندز -کارپ و سیمپلکس جواب دقیق را برای مسئله برش کمینه به دست می آورند. در این بخش از نظر زمان اجرا این دو روش را روی گراف هایی با ۱۰، ۲۰، ۵۰ و۱۰۰ راس، مقایسه می کنیم . همان گونه که در شکل ۴-۱ دیده می شود،
الگوریتم ادموندز- کارپ زمان اجرای کمتری دارد.

۴-۴ مقایسه الگوریتم کارگر و شبیه سازی تبریدی

در فصل ۳ روش کارگر و چگونگی بکارگیری الگوریتم شبیه سازی تبریدی برای مواجهه با مسئله ی برش کمینه بیان شد . و در بخش ۴-۲-۱ نشان داده شد که الگوریتم کارگر برای گراف های وزن دار مناسب نیست.در ادامه نتایج مقایسه این دو روش بر روی گراف های تصادفی تولید شده که در بالا به آن اشاره شد ، را نشان می دهیم.



(الف) نرخ همگرایی (ب) زمان اجرا (ج) میانگین خطا

شکل ۴-۲: مقایسه نتایج روش پیشنهادی (Simulated Annealing) با الگوریتم کارگِر

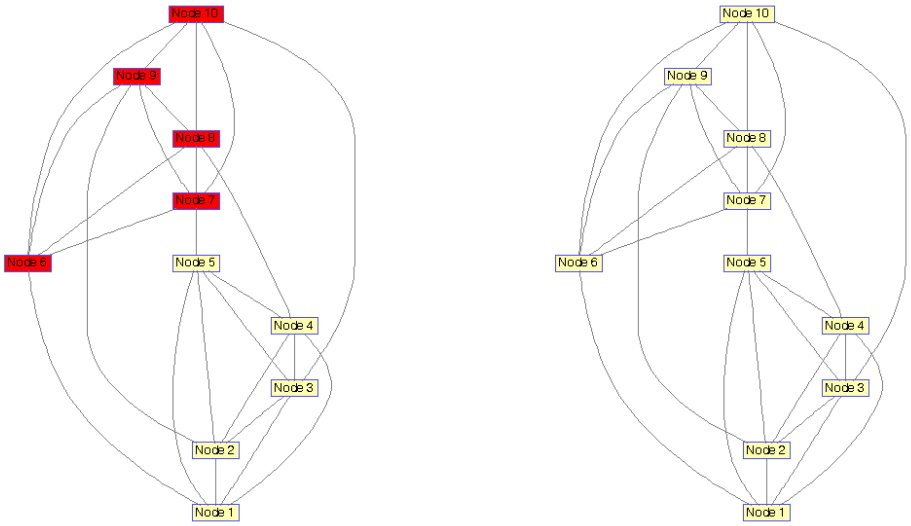
۴-۴-۱ نتایج پیاده سازی

روش کارگِر و شیوه ی پیشنهادی در نرم افزار MATLAB پیاده سازی شده و نتایج اجرا از جنبه های مختلف مورد بررسی قرار گرفته است. هر دو روش روی گراف هایی با تعداد ۱۰، ۲۰، ۵۰، ۱۰۰، ۲۰۰ و ۵۰۰ رأس و با جواب از قبل مشخص اجرا شدند. به لحاظ ماهیت تصادفی هر دو روش، هر روش روی هر مسئله ۲۰ بار اجرا شد و بهترین جواب، میانگین زمان اجرا و فرکانس همگرایی محاسبه گردید. نتایج آزمایشات در شکل ۴-۲ آمده است. منظور از فرکانس همگرایی، درصد اجراهایی است که اختلاف جواب بدست آمده و جواب اصلی مسئله از یک درصد جواب اصلی کمتر بوده است.

همان گونه که مشاهده می شود، در تمام آزمایشات روش شبیه سازی تبریدی به جواب رسیده است و نرخ همگرایی آن ۱۰۰ درصد بوده است. زمان اجرای شبیه سازی تبریدی با افزایش اندازه مسئله به مقدار بسیار کمی افزایش پیدا کرده است، درحالیکه زمان اجرای الگوریتم کارگِر با افزایش اندازه مسئله به صورت نمایی زیاد شده است به صورتی که برای ۵۰۰ رأس، زمان هر اجرای آن تقریبا ۱۰۰ ثانیه شده است که بیش از ۱۰۰ برابر زمان اجرای روش شبیه سازی تبریدی است. در شکل ۴-۳ گراف تصادفی تولید شده و خروجی الگوریتم SA برای گراف با ۱۰ راس نشان داده شده است. راسهای متعلق به دسته P با رنگ قرمز و راسهای متعلق به دسته $\bar{P}$ با رنگ زرد مشخص شده اند.

۴-۵ مقایسه الگوریتم کارگر و جستجوی ممنوعه

در فصل ۳ روش کارگر و چگونگی بکارگیری الگوریتم جستجوی ممنوعه برای مواجهه با مسئله ی برش کمینه بیان شد . و در بخش ۴-۲-۱ نشان داده شد که الگوریتم کارگر برای گراف های وزن دار مناسب نیست. در ادامه نتایج مقایسه این دو روش بر روی گراف های تصادفی تولید شده که در بالا به آن اشاره شد، را نشان می دهیم.

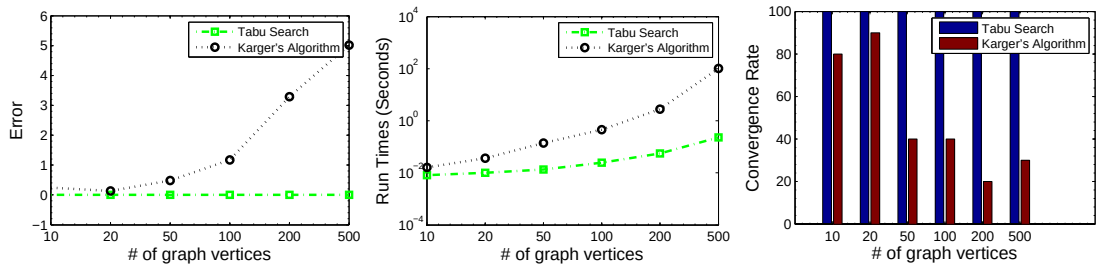


(الف) گراف تولید شده (ب) خروجی الگوریتم SA

شکل ۴-۳: خروجی روش SA

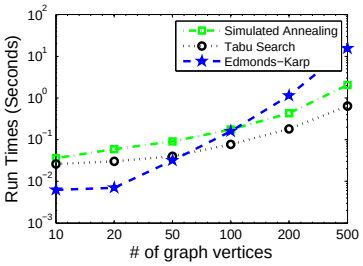
۴-۵-۱ نتایج پیاده سازی

روش کارگِر و شیوه ی پیشنهادی در نرم افزار MATLAB پیاده سازی شده و نتایج اجرا از جنبه های مختلف مورد بررسی قرار گرفته است. هر دو روش روی گراف هایی با تعداد ۱۰، ۲۰، ۵۰، ۱۰۰، ۲۰۰ و ۵۰۰ رأس و با جواب از قبل مشخص اجرا شدند. به لحاظ ماهیت تصادفی هر دو روش، هر روش روی هر مسئله ۲۰ بار اجرا شد و بهترین جواب، میانگین زمان اجرا و فرکانس همگرایی محاسبه گردید. نتایج آزمایشات در شکل ۴-۴ آمده است. منظور از فرکانس همگرایی، درصد اجراهایی است که اختلاف جواب بدست آمده و جواب اصلی مسئله از یک درصد جواب اصلی کمتر بوده است. همان گونه که مشاهده می شود، در تمام آزمایشات روش جستجوی ممنوعه به جواب رسیده است و نرخ همگرایی آن ۱۰۰ درصد بوده است. زمان اجرای جستجوی ممنوعه با افزایش اندازه مسئله به مقدار بسیار کمی افزایش پیدا کرده است، درحالیکه زمان اجرای الگوریتم کارگِر با افزایش اندازه مسئله به صورت نمایی زیاد شده است به صورتی که برای ۵۰۰ رأس، زمان هر اجرای آن تقریبا ۱۰۰ ثانیه شده است که بیش از ۱۰۰ برابر زمان اجرای روش جستجوی ممنوعه است.



(الف) نرخ همگرایی (ب) زمان اجرا (ج) میانگین خطا

شکل ۴-۴: مقایسه نتایج روش پیشنهادی (Tabu Search) با الگوریتم کارگِر



شکل ۴-۵: مقایسه الگوریتم ادموندز کارپ و شبیه سازی تبریدی و جستجوی ممنوعه

۴-۶ مقایسه الگوریتم ادموندز-کارپ و شبیه سازی تبریدی و جستجوی ممنوعه

در این بخش الگوریتم های شبیه سازی تبریدی و جستجوی ممنوعه را با الگوریتم ادموندز -کارپ از نظر زمان اجرا روی گراف هایی با ۱۰، ۲۰، ۵۰، ۱۰۰، ۲۰۰ و ۵۰۰ راس، مقایسه کرده ایم. طبق شکل های ۴-۴ و ۴-۲ درصد خطای SA و TS بسیار پایین است و بنا به شکل ۴-۵ زمان اجرای SA و TS برای گراف های با تعداد راس بالا کمتر از الگوریتم ادموندز -کارپ است.بنابراین برای گراف های با تعداد راسهای بالا استفاده از روشهای پیشنهادی SA و TS مناسب تر است.

مراجع

[۱] جوکار، احسان. جریان بیشینه در گرافهای مسطح. پایان"نامه کارشناسی"ارشد، دانشگاه یزد، ۱۳۹۱.

[۲] چارتراند، گری و اولرمن، آرترود. نظریه الگوریتمی و کاربردی گرافها. ترجمه"ی هاشمی، مهدی تشکری. دانشگاه صنعتی امیرکبیر، ویرایش اول، ۱۳۸۳.

[3] Borradaile, G and Klein, Ph. An O(N Log N) Algorithm for Maximum St-flow in a Directed Planar Graph. *Journal of ACM*, 56(2):1–30, April 2009.

[4] Stoer, M and Wagner, F. A simple min-cut algorithm. *Journal of ACM*, 44(4):585–591, July 1997.

[5] Bazaraa, M.S., Jarvis, J.J., and Sherali, H.D. *Linear Programming and Network Flows*. Wiley, 2011.

[6] Karger, D. and Stein, C. A new approach to the minimum cut problem. *J. ACM*, 43(4):601–640, July 1996.

[7] Alpaydin, E. *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2004.

[8] Schrijver, A. On the history of the transportation and maximum flow problems. *Mathematical Programming*, 91(3):437–445, 2002.

[9] Harris, T.E. and Ross, F.S. *Fundamentals of a Method for Evaluating Rail Net Capacities*. Rand Corporation, 1955.

[10] Ford, L.R.and Fulkerson, D.R. Maximum flow through a network. *Mathematics*, 8:399–411, 1956.

[11] Ford, L.R.and Fulkerson, D.R. *Flows in Networks*. Princeton University Press, Princeton, 1962.

[12] Henzinger, Monika Rauch, Klein, Philip N., Rao, Satish, and Subramanian, Sairam. Faster shortest-path algorithms for planar graphs. *J. Comput. Syst. Sci.*, 55(1):3–23, 1997.

[13] Ahuja, Ravindra K., Magnanti, Thomas L., and Orlin, James B. *Network Flows, theory, algorithms, and applications*. Prentice-Hall, New Jersey, 1993.

[14] Goldberg, A V and Tarjan, R E. A new approach to the maximum flow problem. in *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, pp. 136–146, New York, NY, USA, 1986. Journal of ACM.

[15] Khuller, Samir, Naor, Joseph, and Klein, Philip N. The Lattice Structure of Flow in Planar Graphs. *SIAM J. Discrete Math.*, 6(3):477–490, 1993.

[16] Nagamochi, H and Ibaraki, T. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM J. Discret. Math.*, 5(1):54–66, February 1992.

[17] Har-Peled, Sariel. Minimum cut in a graph. University Lecture (Randomized Algorithms), 2002. http://sarielhp.org/teach/2002/a/notes/min_cut.pdf.

[18] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and E.Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[19] Cerny, V. Thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm. *J. Optim. Theory Appl.*, 45:41–51, 1985.

[20] Fielding, M. Simulated annealing with an optimal fixed temperature. *siamo*, 11(2):289–307, 2000.

[21] Rayward-Smith, V.J., Osman, I.H., Reeves, C.R., and Smith, G.D. *Modern heuristic search methods*. Wiley, 1996.

[22] Dowsland, K.A. *Simulated Annealing*. Advanced Topics in Computer Science Series. McGraw-Hill Book Company, 1995.

[23] .Eglese, R .W. Simulated annealing: A tool for operational research. *Operational Research*, pp. 271–281, 1990.

[24] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. Optimization by simulated annealing. *SCIENCE*, 220(4598):671–680, 1983.

[۲۵] حسینی، فاطمه سادات و امین طوسی، محمود. برش کمینه ی گراف با شبیه سازی تبریدی. در هفتمین کنفرانس بین المللی انجمن ایرانی تحقیق در عملیات، صفحات ۳۷۴-۳۷۵، سمنان، ۱۳۹۳.

[26] Glover, F. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.*, 13(5):533–549, May 1986.

[27] Glover, F and Laguna, M. *Tabu Search*. Kluwer Academic, Dordrecht, 1997.

[28] F., Glover, Glover, F., and Laguna, M. Tabu search. in Pardalos, P.M. and Resende, M.G.C., eds. , *Handbook of Applied Optimization*, pp. 178–183. Oxford University Press, 2002.

[29] Merriam-Webster, Inc. *Merriam-Webster's collegiate dictionary*. Merriam-Webster, 1997.

[۳۰] یقینی، م و اخوان کاظم زاده، م. الگوریتمهای بهینه سازی فراابتکاری. جهاد دانشگاهی امیرکبیر، ویرایش اول، ۱۳۹۰.

[31] Gendreau, M. An Introduction to Tabu Search. in Glover, F. and Kochenberger, G., eds. , *Handbook of Metaheuristics*, chap. 2, pp. 37–54. Kluwer Academic Publishers, 2003.

[۳۲] حسینی، فاطمه سادات و امین طوسی، محمود. برش کمینه ی گراف باجستجوی ممنوعه. در هفتمین کنفرانس بین المللی انجمن ایرانی تحقیق در عملیات، صفحات ۴۱۱-۴۱۲، سمنان، ۱۳۹۳.

پیوست الف

برنامه های نوشته شده

در این قسمت بخش اصلی برنامه های نوشته شده برای پایان نامه ذکر شده اند.

برنامه الف-۱ کد تولید گراف تصادفی با مشخص بودن لبه ها و هزینه ی برش کمینه را نشان می دهد.

برنامه الف-۱: کد تولید گراف تصادفی با مشخص بودن لبه ها و هزینه ی برش کمینه

```
% Generate Random mincut problem
function [nC,nV,D,G,C,cutCost] = rand_mincut_problem(N,
   nCutEdges)
if N<1
    disp('N should be an even positive number');
    return;
end
% N Should be even
if mod(N,2)~=0
    N = N+1;
end
if nargin<2
    nCutEdges = N/2;
end
nC = 2;

minEW = 1; % Minimum edge's weight for strong connections
cs = N/2; % Clique Size
% Clique No. 1, Strong Connections
G1 = (rand(cs)+minEW);
tG1 = tril(G1,-1);
G1 = tG1 + tG1';
% Clique No. 2
G2 = (rand(cs)+minEW);
tG2 = tril(G2,-1);
```

```
G2 = tG2 + tG2';

G = zeros(N);
G(1:cs,1:cs) = G1;
G(cs+1:N,cs+1:N) = G2;
% Connect 2 clique, Weak Connections

D = tril(G,-1);
nV = size(G,1);

c1minDegree = sum(G1);
c2minDegree = sum(G2);
minDegreeCost = min([c1minDegree c2minDegree]);

maxDagreeBetweenClusters = ceil(nCutEdges / cs);
newVals = rand(1,nCutEdges);
newVals = newVals / sum(newVals);
newVals = newVals.*minDegreeCost;
minCutCost = sum(newVals);

newVals = sort(newVals);
i = 1;
while newVals(end)<minEW && i<nCutEdges
    minVal = newVals(i);
    elementsToDecrease = i:ceil(.9*nCutEdges);
    newVals(elementsToDecrease) = newVals(
       elementsToDecrease)-minVal;
    incVal = numel(elementsToDecrease)*minVal;
    elementsToIncrease = elementsToDecrease(end)+1:
       nCutEdges;
    newVals(elementsToIncrease) = newVals(
       elementsToIncrease)+incVal/numel(elementsToIncrease)
       ;
    i = i+1;
end

if newVals(end) < minEW
    disp('Edges Rewighting unsuccessful...');
    disp('MinSpannig Tree will produce good results.')
end

c1 = 1:cs;
c2 = cs+1:N;

idxInC1 = repmat(randperm(cs),1,maxDagreeBetweenClusters);
idxInC1(nCutEdges+1:end) = [];
```

```
۶۷
۶۸ idxInC2 = repmat(randperm(cs)+cs,1,maxDagreeBetweenClusters
      );
۶۹ idxInC2(nCutEdges+1:end) = [];
۷۰
۷۱ % New graph
۷۲ G = D + D';
۷۳
۷۴ for k=1:nCutEdges
۷۵     v1 = idxInC1(k); v2 = idxInC2(k);
۷۶     G(v1,v2) = newVals(k);
۷۷     G(v2,v1) = newVals(k);
۷۸ end
۷۹ D = tril(G);
۸۰ cutCost = 0;
۸۱ for i=1:numel(c1)
۸۲     for j=1:numel(c2)
۸۳         % v1 is < v2
۸۴         v1 = c1(i); v2 = c2(j);
۸۵         cutCost = cutCost + G(v1,v2);
۸۶     end
۸۷ end
۸۸ % Cluster indices
۸۹ C = ones(1,nV);
۹۰ C(c2) = 2;
```

برنامه الف-۲ کد الگوریتم ادموندز کارپ (الگوریتم ۲-۲)را نشان می دهد.

برنامه الف-۲: کد الگوریتم ادموندز کارپ

```
۱ %Edmonds_Karp algorithm for max flow  ,min cut of graph;
۲ %function [f,S,objectivevalue]=Edmonds_Karp(G,f,s,t)
۳ function objectivevalue=Edmonds_Karp(G,f,s,t)
۴ if nargin<2
۵   f=zeros(size(G));
۶   s=1;
۷ t=size(G,1);
۸ end
۹ nV=size(G,1);
۱۰ nE=length(find(G));
۱۱ complexity=nV*nE*nE/2;
۱۲ for i=1:complexity
۱۳ G1=remaindergraph(G,f);
۱۴ [distance,path,S1] = mBFS(G1,s,t);
۱۵ if  distance == inf
۱۶     S=zeros(1,nV);
۱۷     S(S1)=1;
```

```
    S2=(find(~S));
    objectivevalue=sum(sum(G(S1,S2)));
    break
else
 f=access_flow(f,G,G1,path);
end
end
```

برنامه الف-۳ کد الگوریتم کارگر (الگوریتم ۳-۴) را نشان می دهد.

برنامه الف-۳: الگوریتم کارگر

```
function kargerCosts=karger(G,n)
UG=sparse(G);
[ST,pred] = graphminspantree(UG);
%h = view(biograph(ST,[],'ShowWeights','on','ShowArrows','
   off'));
 [i,j,s]=find(ST);
  m=randi(n-1,1,1);
  ST(i(m),j(m))=0;
% ST(i(n-1),j(n-1))=0;
%ST=ful(ST);
[S,C] = graphconncomp(ST,'Weak',true,'Directed',false);
kargerCosts=cost_cut_objfun(G,C);
% G1=zerose(n);
% for k=1:n-3
%  [i,j,s]=find(G);
%  h=randi(length(i),1,1);
%  t=graphisdag(G1);
% %  if t==1
% %  G1(i(h),j(h))=1;
% %  G(i(h),j(h))=0;
% %  end
% end
```

برنامه الف-۴ کد الگوریتم شبیه سازی تبریدی را نشان می دهد.

برنامه الف-۴: الگوریتم شبیه سازی تبریدی

```
function minCutCost = min_cut_sa(G)
n = size(G,1);
for i=1:n
    G(i,i)=inf;
end
[i,s]=min(G);
[j,l]=min(i);
```

```
s0(s(l))=1;
s0(l)=0;
a=s(l);
b=l;
for i=3:n
     if sum(G(a,i))+sum(G(i,a))>sum(G(b,i))+sum(G(i,b))
        s0(i)=1;
       % a=[a i];
     elseif sum(G(a,i))+sum(G(i,a))<=sum(G(b,i))+sum(G(i,b)
        )
        s0(i)=0;
       % b=[b i];
     end
end
maxIter = 200;
T = 100;
fVals =cost_cut_objfun(G,s0);
sbest=s0;
cbest=cost_cut_objfun(G,s0);
 for i=1:maxIter
    s1 =   randNb(s0,G);
    deltaE = cost_cut_objfun(G,s1) - cost_cut_objfun(G,s0);
    if deltaE < 0
        sbest=s1;
        cbest=cost_cut_objfun(G,s1);
        s0 = s1;
   elseif exp(-deltaE/T)>rand
        s0 = s1;
    end
   T = 0.95 * T;
    fVals =[fVals cost_cut_objfun(G,s0)];
 end
minCutCost = cost_cut_objfun(G,sbest);
  cut_show(sbest,G);
```

برنامه الف-۵ کد الگوریتم جستجوی ممنوعه را نشان می دهد.

برنامه الف-۵: الگوریتم جستجوی ممنوعه

```
function minCut_Cost = min_cut_tabu(G)
N=size(G,1);
for i=1:N
    G(i,i)=inf;
end
[i,m]=min(G);
[j,l]=min(i);
s(m(l))=1;
```

```
s(l)=0;
a=m(l);
b=l;
for i=3:N
     if sum(G(a,i))+sum(G(i,a))>sum(G(b,i))+sum(G(i,b))
        s(i)=1;
     elseif sum(G(a,i))+sum(G(i,a))<=sum(G(b,i))+sum(G(i,b)
        )
        s(i)=0;
     end
end
 for i=1:N
     G(i,i)=0;
 end
e = cost_cut_objfun(G,s);
sbest = s; ebest = e;
maxIter = 100;
fVals = e;
tabuList = zeros(1,N);
tabuTenure = N/5;
for i=1:maxIter
    candidateList = [];
    a=find(s==1);
    b=find(s~=1);
      l=rand(1);
      sNeighborhood=a(1);
     if l<.5
         for i=1:size(a,2)
            if sum(G(a,a(i)))<=sum(G(b,a(i)))
               sNeighborhood=a(i);
               break;
            end
         end
     elseif l>=.5
       for i=1:size(b,2)
           if sum(G(b,b(i)))<=sum(G(a,b(i)))
              sNeighborhood=b(i);
              break;
           end
       end
    end
    for sCandidate = sNeighborhood
         i1 = sCandidate(1);
        if ~tabuList(i1)
            candidateList = [candidateList ; sCandidate'];
        end
```

```
    end
    [snew,enew,idxNew] = LocateBestCandidate(s,
       sNeighborhood,G);
    if isempty(candidateList) || enew<ebest
        candidateList = sNeighborhood;
    else
         [snew,enew,idxNew] = LocateBestCandidate(s,
            candidateList,G);
    end
        s = snew;
        fVals = [fVals, enew];
        sCandidate = candidateList(idxNew,:);
        i1 = sCandidate(1);
        tabuList(i1) = tabuTenure;
        cond = tabuList>0 & tabuList<=tabuTenure;
        tabuList(cond) = tabuList(cond)-1;
        if(enew < ebest)
            sbest = snew;
            ebest = enew;
        end
end
minCut_Cost=ebest;
end

function [sbest,ebest,idxBest] = LocateBestCandidate(s,
   candidateList,G)
for i=1:size(candidateList,1);
    [snew,enew]  = neighbor(s,candidateList,i,G);
    if i==1
        sbest = snew; ebest = enew; idxBest = i;
    else
        if enew < ebest
            sbest = snew; ebest = enew; idxBest = i;
        end
    end
end
end
function [snew,enew] = neighbor(s,candidateList,i,G)
sCandidate = candidateList(i,:);
snew = s;
i1 = sCandidate(1);
   snew(i1) =(~s(i1));
   if all(snew==1)
    r=randi(size(s,1),1,1);
    snew(r)=0;
elseif  all(snew==0)
```

```
    r=randi(size(s,1),1,1);
    snew(r)=1;
end
   enew = cost_cut_objfun(G,snew);
end
```

برنامه الف-۶ نحوه استفاده از جریان بیشینه را نشان می دهد.

برنامه الف-۶: کد نحوه استفاده از جریان بیشینه

```
    [nC,nV,D,DG,C,cutCost] = rand_mincut_problem(nV);
    cost=cutCost
    G=(DG~=0);
    Truth_MinCut(k)=cost;
    n=nV;
    ST = sparse(D');
    stbG = biograph(ST,[],'ShowWeights','on');
    h = view(stbG);
    [M,F,K] = graphmaxflow(ST,1,nV)
```

واژه نامه فارسی به انگلیسی

ادموندز کارپ . . . . . Edmonds -Karp
اشباع نشده . . . . . unsaturated
افزایشی . . . . . augmenting
الگوریتم استور واگنر . . . . . Stoer and Wagner algorithm
الگوریتم کارگر . . . . . Karger‘s algorithm
بالاترین مسیر . . . . . uppermost path
برش کمینه . . . . . minimum Cut
بورَدایل و کلین . . . . . Borradaile - Klein
بهینه سازی ترکیباتی . . . . . combinatorial optimization
پتانسیل . . . . . potential
جریان بیشینه . . . . . maximum flow
جستجوی اول-راست . . . . . right-first search
جستجوی محلی . . . . . local search
جستجوی ممنوعه . . . . . tabu search
چپ ترین مسیر . . . . . leftmost path
دوگان . . . . . dual
سیمپلکس . . . . . simplex
شبیه سازی تبریدی . . . . . simulated annealing
فرا ابتکاری . . . . . metaheuristic
فضای دوری . . . . . cycle space
فورد و فولکرسون . . . . . Ford and Fulkerson
گردش . . . . . circulation
مسطح . . . . . planar
مور . . . . . Moor

واژه نامه انگلیسی به فارسی

افزایشی . . . . . augmenting
بورَدایل و کلین . . . . . Borradaile and Klein
گردش . . . . . circulation
بهینه سازی ترکیباتی . . . . . combinatorial optimization
فضای دوری . . . . . cycle space
دوگان . . . . . dual
ادموندز و کارپ . . . . . Edmonds and Karp
فورد و فولکرسون . . . . . Ford and Fulkerson
الگوریتم کارگر . . . . . Karger‘s algorithm
چپ ترین مسیر . . . . . leftmost path
جستجوی محلی . . . . . local search
جریان بیشینه . . . . . maximum flow
فرا ابتکاری . . . . . metaheuristic
برش کمینه . . . . . minimum Cut
مور . . . . . Moor
مسطح . . . . . planar
پتانسیل . . . . . potential
جستجوی اول-راست . . . . . right-first search
سیمپلکس . . . . . simplex
شبیه سازی تبریدی . . . . . simulated annealing
جستجوی ممنوعه . . . . . tabu search
اشباع نشده . . . . . unsaturated
بالاترین مسیر . . . . . uppermost path

Hakim Sabzevari University

An Outline of MSc. Thesis

Surname:Hoseini	Name:Fateme sadat	Student No.:9113133011
Supervisor: Dr. Mahmood Amintoosi		
Advisor: Dr. Mahdi Zaferanieh		
Faculty of Mathematics and Computer Science	Applied Mathematics	Operational Research

Title of thesis: Graph Minimum Cut

Keywords: minimum Cut , maximum flow , Stoer and Wagner algorithm ,Karger‘s algorithm , metaheuristic algorithm.

Abstract: In graph theory, cutting means to divide graph vertices into two disjoint nonempty subsets s and $(v - s)$. cutting edgesof a cutting are those edges which are in it‘s cutting set. The aim of the minimum cut problem is to find these two subsets to minimize the capacity of the cutting edges.

Since the amount of maximum flow is equal to minimum cutting in the graph, there are two methods to face this problem: flow based graph methods and other methods. In this thesis both these two methods are discussed.

In general graphs, methods to obtain maximum flow are divided into augmenting path algorithm and preflow push algorithm. In chapter 2 flow based methods which are generalization of the first algorithm presented by Ford and Fulkerson,are discussed.The major difference is in choosing augmenting path.

Methods which are not based on flow including Stoer and Wagner algorithm,Karger‘s algorithm, tabu search and simulated annealing are also discussed in this thesis.

Hakim Sabzevari University
Faculty of Mathematics and Computer Science

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Science in Applied Mathematics

Graph Minimum Cut

Supervisor:

Dr. Mahmood Amintoosi

Advisor:

Dr. Mahdi Zaferanieh

By:

Fateme sadat Hoseini

september 2014