فصلنامه پژوهش در علوم، مهندسی و فناوری
ISSN: 2476-5082
دوره ۲، شماره ۱، بهار ۱۳۹۵
www.uctjournals.com
صفحات ۲۵-۳۸

دوگان جریان با مینیمم هزینه در شبکه های مسطح

لیلا کلاهدوزی[1]، دکترحسن صالحی فتح آبادی[2] و دکتر مجید زهره بندیان[3]

۱کارشناسی ارشد/ ریاضی کاربردی - تحقیق در عملیات/دانشگاه آزاد اسلامی، واحد کرج،گروه ریاضی، کرج، ایران
آدرس ایمیل : leile563@yahoo.com
۲دکتری تخصصی ، ریاضی کابردی – تحقیق در عملیات /استاد
آدرس ایمیل:hsalehi@ut.ac.ir
۳ دکتری تخصصی /ریاضی کابردی – تحقیق در عملیات /استاد
آدرس ایمیل :zohrebandian@yahoo.com

چکیده

در شبکه های جریان عبور جریان از کمان های(مسیرها) شبکه دارای هزینه می باشد. در این گونه شبکه ها، مسئله جریان با مینیمم هزینه جریانی است که ضمن انتقال مقدار لازم جریان از گره های تولیدی(منبع) به گره های مصرف(مقصد) دارای حد اقل هزینه نیز باشد. این مسئله در شبکه های معمولی با الگوریتم های متعددی حل گردیده است. اما در شبکه های مسطح نیاز به بررسی و مطالعه بیشتری دارد. شبکه های مسطح به گونه ای طراحی می شوند که اگر روی یک سطح دو بعدی(مثلاً کاغذ) رسم شوند کمان ها همدیگر را قطع نمی کنند. به علاوه هر گراف مسطح به عنوان گراف اولیه دارای دوگان به صورت یک شکل هندسی نیز می باشد. در این مسئله از گراف دوگان به عنوان یک وسیله میانی استفاده می گردد. تابع هدف این مسائل سرجمع کل هزینه های انتقال جریان بین گره ها می باشد. قیود این مسئله تعادل جریان در گره ها است. به این معنی که حجم جریان های ورودی گره ها، حجم جریان های خروجی گره ها و مقدار جریان تولید/ مصرف گره ها دارای تعادل (بالانس) می باشد، می‌دهد.

واژه‌های کلیدی: شبکه های مسطح، دوگان، مینیمم هزینه

۱- مقدمه

مسئله شبکه جریان با کمترین هزینه، مسئله‌ی اصلی در علوم کامپیوتر و بهینه‌سازی ترکیبی است، و یکی از مهم ترین مسائل جریان شبکه است و بسیاری از کاربردها و الگوریتم‌ها برای این مسئله وجود دارد آهوجا[1] و همکاران [۱] خولر[2] و همکاران[۱۷] در سال۱۹۹۳ ساختار فضای راه‌حل مسئله گردش حداقل هزینه مسطح را مورد مطالعه قرار دادند. آنها چند روشی که ممکن است منجر به الگوریتم کارایی برای مسئله شود، پیشنهاد کردند. یکی از روش‌هایی که آنها پیشنهاد کردند، فرمول‌بندی گراف دوگانه از مسئله جریان حداقل هزینه است که ما به دنبال آن هستیم. چمبرز[3]و همکاران[۳] در سال ۲۰۰۹ الگوریتم حداکثر جریان را برای گراف‌های دسته‌ی کراندار، و همچنین یک الگوریتم گردش حداقل هزینه برای پیدا کردن گردشی با حداقل هزینه در میان جریانات موردنظر ویژگی‌های توپولوژیکی معین ارائه داده‌اند. این الگوریتم همچنین برای برنامه نویسی خطی دوگان مسئله در گراف دوگان هندسی استفاده شده است ایمای و ایوانو[4] [۱۵] الگوریتمی مبتنی بر روش نقطه درونی برای برنامه‌ریزی خطی ارائه داده‌اند که در آن $o(n^{1.594}\sqrt{logn}\log(n \max\{c,u\}))$ زمان در حال اجرا و نیز $o(nlogn)$ فضای مورد نیاز است. اخیرا، ویدیاناتان[5] و آهوجا [۲۷] مسئله جربان حداقل هزینه در چرخه را مورد مطالعه قرار داده‌اند.که یک مورد خاص از یک شبکه مسطح است، آنها $o(nlogn)$ را الگوریتم زمان برای دور دو طرفه $o(n)$ را الگوریتم زمان برای چرخه یک سویه نام‌گذاری کرده‌اند. کورنلسون و کارنباهور[6] [۱۴] مسئله در زمینه‌ی طراحی گراف را مورد مطالعه قرار داده‌اند. کورنلسون و همکاران[۵] مسئله ای در زمینه پردازش تصویر را مورد مطالعه قرار داده‌اند. و $o(n^{\frac{3}{2}}u\sqrt{\delta})$ الگوریتم برای مسئله وقتی که همه‌ی کمان‌ها، ظرفیت جدایی ناپذیر متناهی دارند، ارائه داده‌اند. این الگوریتم اخیر به شبکه‌ها اجازه می‌دهد تا یک راس اضافی که مسطح بودن را نقض می‌کند، داشته باشند.

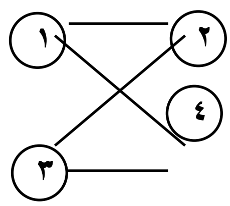
در این پایان نامه به ترتیب به بیان فصل های زیر می پردازیم.

در فصل اول، تعاریف کلی و مقدماتی مورد نیاز مسئله را بیان می کنیم. در فصل دوم به کاربردهای دوگان مسئله جریان با مینیمم هزینه در ریاضی خواهیم پرداخت. تعدادی از حالات خاص جریان با کمترین هزینه نقش اساسی را در تئوری گراف و به طور عام تر در شبکه های جریان بازی می کند. در این فصل به دو مورد آن اشاره کرده و می گوییم که چرا این دو حالت خاصی از مسئله ی جریان با کمترین هزینه است. به علاوه این دو کاربرد رابطه ی عمیقی با مسئله ی جریان با کمترین هزینه دارد زیرا که جنبه های مشترکی با هر دوی این مسائل دارد. تعدادی از الگوریتم های جریان با کمترین هزینه به طور زیادی جهت حل مسئله ی عمومی به این دو کاربرد وابسته اند. در فصل سوم ساختار(ساختمان)داده برای نگهداری مجموعه‌ای از درخت های با رئوس ناهمبند تحت دو عملگر مطرح می‌شود، یک عملگر اتصال که دو درخت را با افزودن یک یال ترکیب کرده و به یک درخت تبدیل می‌کند، و دیگری عملگر برش که هر درخت را با حذف یک یال، به دو درخت تقسیم می‌کند. هر عملگر به زمان$o(logn)$ نیاز دارد. فصل چهارم با یک مسئله جریان باحداقل هزینه در شبکه های معمولی و مسئله جریان گردشی با حداقل هزینه در یک شبکه مسطح شروع می کنیم.سپس مسئله متناظر در گراف دوگان هندسی را که به صورت مسئله برنامه ریزی خطی تشریح می کنیم. سپس دو تبدیل را به کار می‌بریم، یکی تبدیل مبتنی بر دوگان هندسی از نمودار-های مسطح و دیگری تبدیل دوگان برنامه‌ریزی خطی. در نتیجه، یک مسئله جریان با حداقل هزینه در شبکه مسطح مرتبط است که قیدهای متعادل توسط هزینه‌هایی از مسئله‌ی اصلی تعریف می‌شوند که چنین هزینه هایی با ظرفیت های مسئله اصلی تعریف شده‌اند.

تعاریف ومفاهیم مورد نیاز

۱) **گراف:** یک مجموعه غیرتهی از اشیا به نام رأس و مجموعه‌ای از یال ها که رأس ها را به هم وصل می‌کند را گراف نامند. مجموعه رأس‌ها را با N و مجموعه یال‌ها را با A و گراف را با$G(N,A)$ نشان می‌دهند. برای نمایش تصویری گراف، از نقطه یا دایره به جای رأس و نیز از کمان یا خط به جای یال استفاده می‌شود.

۲) گراف غیرجهت‌دار: به گرافی که در آن یال ها جهت‌دار نیستند، گراف غیر جهت دار گویند که یال های آن زوج های نامرتب از گره های متمایز می باشند.



شکل۱-۱: گراف غیر جهت دار

N={۱,۲,۳,٤} , $A = \{(۱,۲),(۱,٤),(۲,۳),(۳,٤)\}$

۳) شبکه غیرجهت‌دار: گراف غیرجهت‌دار $G(N, A)$ یک شبکه غیرجهت‌دار است هرگاه در آن توابع عرضه/تقاضا، ظرفیت، هزینه انتقال و جریان به ترتیب به صورت زیر تعریف شده باشند:

- عرضه/تقاضا $b : N \longrightarrow R$
- ظرفیت $u : A \longrightarrow R^+ \cup \{۰\}$
- هزینه انتقال $c : A \longrightarrow R$
- جریان $x : A \longrightarrow R^+ \cup \{۰\}$

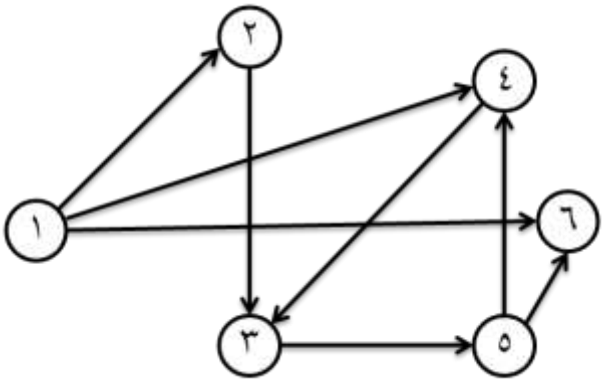
$u\left((i,j)\right) = u_{ij}$ را ظرفیت کمان، (i,j) $c\left((i,j)\right) = c_{ij}$ را هزینه انتقال یک واحد جریان از کمان (i,j) و $b(i) = b_i$ را عرضه یا تقاضای گره‌ی i و $x\left((i,j)\right) = x_{ij}$ را جریان روی کمان (i,j) می‌نامند.

توجه : اگر $b_i > ۰$ آنگاه گره i را گره عرضه می نامند و اگر $b_i < ۰$ آنگاه گره i را گره تقاضا می نامند و اگر $b_i = ۰$ آنگاه گره i را گره میانی می نامند.

توجه : گره میانی گره ی است که در آن تقاضا وعرضه موجود نیست وفقط به عنوان یک رابطه استفاده می شود.

۴) گراف و شبکه جهت‌دار: تعریفی مشابه گراف و شبکه غیرجهت‌دار دارد با این تفاوت که در این حالت، همه یال ها دارای جهت بوده و کمان نامیده می شوند،کمان زوج های مرتبی از گره های متمایز هستند یعنی $A \subseteq N \times N$.

شکل (۱-۲) مثالی از یک گراف جهت‌دار با مجموعه رئوس $V = \{۱,۲,۳,٤,٥,٦\}$ و مجموعه کمان های $A = \{(۱,۲),(۱,٤),(۱,٦),(۲,۳),(۳,٥)(٤,۳),(٥,٤),(٥,٦)\}$ است.



گراف جهتدار با شش گره و هشت کمان

۵) **زیرگراف:** گراف $G=(N,A)$ را در نظر بگیرید فرض کنید $N' \subset N$ و $A' \subset A$ آنگاه $G'=(N',A')$ را یک زیر گراف $G=(N,A)$ می نامند به گونه ای که اگر، $(i,j) \in A'$ آن گاه i و j در N' باشند.

۶) **گراف مسطح** : گراف مسطح گرافی است که بتوان آن را در یک صفحه به گونه ای رسم کرد که یال ها فقط در نقاط انتهایی همدیگر را قطع کنند، به عبارت دیگر یال ها روی یک صفحه همدیگر را قطع نکنند.

۷) **شبکه ی مسطح:** شبکه های مسطح به گونه ای طراحی می شوند که اگر روی یک سطح دو بعدی رسم شوند کمان ها همدیگر را قطع نمی کنند.

۸) **زنجیر:** در شبکه غیرجهت‌دار $G(N,A)$، یک زنجیراز گره i، به i_p دنباله ای از یال های P=$\{(i_{\cdot},i_{۱}),(i_{۱},i_{۲}),\dots,(i_{p-۱},i_p\}$ است بگونه ای که در آن گره اولیه ی هر یال گره پایانی یال قبلی خود است.

۹) **مسیر:** اگر تمام یال های یک زنجیر در یک جهت باشند آنگاه به زنجیر یک مسیر می گویند.

۱۰) **ریشه ی درخت:** گره r را در نظر بگیرید، اگر یک مسیر جهت دار موجود باشد که گره r را به هر گره دیگر درخت وصل کند، در این صورت r را ریشه درخت می نامند.

۱۱) **مسیر مینیمال:** مسیری که فاقد گره یا کمان تکراری باشد.

۱۲) **دور:** زنجیری که گره ابتدا وانتهای آن برهم منطبق باشند دور نامیده می شود. مسیری که گره های ابتدا و انتهایی آن بر هم منطبق باشند دور جهت دار نامیده می شود (یک شبکه را بدون دور گوییم هر گاه شامل هیچ دور جهتداری نباشد).

۱۳) **گراف همبند:**گراف $G=(V,A)$را درنظر بگیرید. اگر بین هردوگره دراین گراف، حداقل یک مسیر موجود باشـد، آنگـاه گراف G، همبند است.

۱۴) **درخت:** گراف همبندی که فاقد دور باشد درخت نامیده می شود.

۱۵) **درخت پوشا یا فراگیر:** درختی است که شامل همه ی گره های گراف باشد، یعنی یک زیر گراف همبند بدون دور پوشا است.

۱۶) **گره مبداء و مقصد:** در بخشی از مسائل شبکه های جریان یک یا چند گره حاوی کالا(جریانی) هستند که باید به گره(گره های) دیگری انتقال یابد. گره های دسته اول را گره(گره های) مبداء و گره های دسته دوم را گره(گره های) مقصد نامند.

۱۷) **گره پدر(father) و فرزند(son)** : گره ای که ابتدای کمان است را گره پدر(father) و گره ای که انتهای کمان است را گره فرزند(son) می نامند.

۱۸) **برش:** شبکه$G(N,A)$ ، را در نظر بگیرید فرض کنید , $\bar{X}=N\setminus X$ $\varnothing\neq X\subset N$ به مجموعه یال هایی که یک انتهایشان (ابتدا یا انتها آن ها) در X و انتهای دیگر آن ها در$\bar{X}$ است را برش$(\bar{X},X)$ می نامند که به صورت زیر تعریف می شود.

$$(X,\bar{X})=\{(i,j)\mid i\in X,\ j\in\bar{X}\}$$

۱۹) **برش مینیمال:** یک برش را مینیمال گوییم هر گاه هر زیر مجموعه سره ی آن، یک برش نباشد. برش های مینیمال را به اختصار با MC نمایش می دهیم.

۲۰) **هزینه تخفیف یافته کمان** (i,j) : اگربرای هر گره $i\in V$،$\pi(i)$پتانسیل گره i باشد و c_{ij} هزینه انتقال یک واحد جریان را ازگره i به گره j درطول کمان $(i,j)\in A$ نشان دهد، برای یک مجموعه داده شده از پتانسیل گره ها مانند$\pi=(\pi(۱),\pi(۲),\dots,\pi(n))$، هزینه کاهش یافته مربوط به کمان(i,j) را باc_{ij}^{π} نمایش می دهیم وبه صورت $c_{ij}^{\pi}=c_{ij}-\pi(i)+\pi(j)$ تعریف می کنیم.

۲۱) **اندازه یک داده با مقدار x:** اندازه یک داده با مقدار x را به دو صورت می‌توان تعریف کرد:

۱) اندازه آن را خود xتعریف کرد.
۲) اندازه آن را$\log x$تعریف کرد.

بنابر دلایل خاصی تعریف دوم کاراتر است. دلیل اصلی با توجه به کار کامپیوتر بیان می‌شود. اکثر کامپیوترها اعداد را به صورت دودویی و در مکان های حافظه با اندازه ثابت بیت ذخیره می‌کنند و نمایش دودویی x، به $\log x$ بیت احتیاج دارد بنابراین مطلوب‌تر آن است که اندازه داده xرا $\log x$ تعریف کنیم.

۲۲) ظرفیت باقیمانده یک کمان: ظرفیت باقیمانده کمان $(i,j) \in A$ را با r_{ij} نشان می دهیم هر گاه $\boldsymbol{x}$ جریان جاری در شبکه باشد آنگاه $r_{ij} = (u_{ij} - x_{ij}) + x_{ji}$ تعریف می‌شود یا به عبارتی دیگر r_{ij} بیشترین جریان اضافی است که می تواند از گره i به گره j ارسال شود.

ظرفیت باقیمانده r_{ij} دارای دو مولفه است:

۱) $(u_{ij} - x_{ij})$ که ظرفیت استفاده نشده کمان (i,j) است

۲) x_{ji} که جریان فعلی روی کمان(j,i) است و می توان این جریان x_{ji} را برای افزایش جریان از گره i به j حذف کرد.

۲۳) شبکه باقیمانده: فرض کنید G=(N,A) یک شبکه جریان باشد و $\boldsymbol{x}$ جریان شدنی در G باشد؛ شبکه باقی مانده $G(x)$ را به صورت زیر از G می‌سازیم.

۱) تمام گره های G در $G(x)$ هستند.
۲) اگر یال (i,j) در G باشد و $x_{ij} < u_{ij}$ آنگاه یال (i,j) با ظرفیت $r_{ij} = u_{ij} - x_{ij}$ را در $G(x)$ جای می‌دهیم.
۳) اگر یال (i,j) در G و $x_{ij} > ۰$ باشد آن‌گاه یال (j,i) با ظرفیت $r_{ji} = x_{ij}$ را در $G(x)$ جای می‌دهیم.

توجه: اگر $0 < x_{ij} < u_{ij}$ آنگاه یال (i,j) در G منجر به دو یال در$G(x)$ می شود.

مفاهیم پایه‌ای الگوریتم‌ها

در این بخش، به طور اجمالی به معرفی مفاهیمی چون الگوریتم، روش‌های اندازه‌گیری کارایی الگوریتم و پیچیدگی الگوریتم می‌پردازیم.

الگوریتم

یک الگوریتم، در واقع یک تولیدکننده‌ی گام به گام جواب مسئله است. گام‌های مختلف یک الگوریتم در حالت کلی عبارتند از:

۱) گام‌های تخصیص (از قبیل تخصیص مقادیر به متغیرها)
۲) گام‌های عملیات (از قبیل جمع، ضرب، تقسیم و...)
۳) گام‌های منطق (از قبیل مقایسه دو عدد و ...)

یک الگوریتم ممکن است یک مسئله را به سرعت و مسئله دیگر را به کندی حل کند، بنابر این باید زمان اجرای الگوریتم‌ها را اندازه بگیریم تا بتوانیم بهترین الگوریتم‌ها را انتخاب کنیم.

سه روش برای اندازه‌گیری کارایی یک الگوریتم وجود دارد.

۱) **آنالیز تجربی:** در این روش برنامه کامپیوتری الگوریتم را نوشته و در کلاس‌های مختلفی از مسائل کارایی، آن را امتحان می‌کنیم.
۲) **آنالیز میانگین حالت:** در این روش تعداد گام‌های مورد انتظار الگوریتم را با انتخاب یک تابع توزیع احتمال، تخمین می‌زنیم.

۳) **آنالیز بدترین حالت:** در این روش با محاسبه حداکثر گام‌های ممکن، یک کران بالا برای تعداد گام ها یا الگوریتم تعیین می‌کنیم.

روش سوم بیشتر از بقیه مورد استفاده قرار می‌گیرد.

پیچیدگی الگوریتم

زمانی که یک الگوریتم برای اجرا صرف می‌کند، ، به نوع و اندازه ورودی‌ها وابسته است. از طرفی مسائل بزرگ‌تر به زمان حل بیشتری نیاز دارند و نیز مسائل مختلف با اندازه‌های یکسان زمان‌های اجرای متفاوتی دارند.

تابع پیچیدگی زمانی یک الگوریتم، بیشترین زمان مورد نیاز برای حل انواع مسائل را بازای اندازه ورودی داده شده مشخص می‌کند. به عنوان مثال، اگر تابع پیچیدگی زمانی یک الگوریتم بازای مقدار ثابت C، برابر Cmn باشد، در این صورت حداکثر زمان اجرای مورد نیاز برای حل هر مسئله شبکه جریان با n گره و m کمان Cmn است.

مسئله ماکزیمم جریان در شبکه‌های $s-t$

اگر شبکه‌ای با n گره و m کمان، که هر کمان دارای کران بالای $U_{ij}(U_{ij} < \infty)$ و کران پایین L_{ij} است، در نظر گرفته شود و در این شبکه گره مبدا با s و گره مقصد با t معرفی شده باشد، آنگاه هدف مسئله ماکزیمم جریان، یافتن جریانی در این شبکه است به طوری که مقدار جریان خروجی از گره s (و مقدار جریانی که به گره t وارد می‌شود)، بدون تجاوز کردن از قیود ظرفیت، بیشینه گردد. بعضی از الگوریتم های ارائه شده برای این مسئله عبارتند از: الگوریتم مسیرهای افزایشی

f در شبکه مقدار جریان از گره s به گره t است که از مبدا خارج شده ودقیقا همان مقدار جریان وارد مقصد می شود. مسئله ماکزیمم جریان به صورت زیر فرمول‌بندی می‌شود:

$$
\begin{aligned}
&Max \quad f \\
&s.t \quad \sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = \begin{cases} f & i = s \\ 0 & i \in V - \{s,t\} \\ -f & i = t \end{cases} \qquad (1-1) \\
&\quad\quad x_{ij} \le u_{ij} \\
&\quad\quad x_{ij} \ge 0 \qquad (i,j) \in A
\end{aligned}
$$

باید توجه داشت که x_{ij}، مقدار جریان از گره i به گره j است.

یکی از الگوریتم هایی که مسئله ماکزیمم جریان را حل می کند، الگوریتم مسیر های افزایشی است که به صورت زیر بیان می‌شود:

در ابتدا باید شبکه ویژگی‌های زیر را داشته باشد:

۱) شبکه همبند باشد.

۲) شبکه جهت‌دار باشد.

۳) شبکه دارای یک مسیر با ظرفیت نامتناهی از s به t نباشد.

۴) ظرفیت کمان‌ها، اعداد صحیح باشند.

الگوریتم مسیرهای افزایشی

۱) یک جواب شدنی اولیه مثل $x = ۰$ در نظر گرفته می‌شود.
۲) شبکه باقیمانده $G'(x)$ ساخته می‌شود.
۳) اگر در شبکه باقیمانده مسیری مانند P از s به t موجود بود، به گام ۴ بروید. در غیر این صورت توقف کنید. جریان فعلی، بهینه است.
٤) $۰ < \Delta = min\{r_{ij} | (i,j) \in P\}$ را در نظر بگیرید.
۵) Δ واحد جریان را روی مسیر P در G اضافه کنید (جریان حاصل، باز هم شدنی باقی می ماند). و به گام ۲ برگردید.

توجه: در شبکه G

۱) اگر $(i,j) \in P$,$(i,j) \in G$ در این صورت جریان برابر است با $x_{ij} = x_{ij} + \Delta$
۲) اگر $(i,j) \in P$,$(j,i) \in G$ در این صورت جریان روی کمان برابر است با $x_{ji} = x_{ji} - \Delta$

لازم به ذکر است که چون شبکه های غیرجهتدار، قابل تبدیل‌شدن به شبکه های جهت‌دار هستند، بنابراین هر الگوریتم ماکزیمم جریان که برای شبکه های جهتدار ساخته شده است، قابل اجرا روی شبکه های غیرجهتدار (با تبدیل آنها به شبکه های جهتدار) نیز است

مسئله جریان با کمترین هزینه

فرض کنید $G = (N,A)$ شبکه جریانی با n گره و m کمان باشد. کالا (جریان) باید از گره‌های عرضه به گره‌های تقاضا ارسال شوند به شرطی که تقاضاها برآورده شده و هزینه ارسال این کالا کمینه شود. x_{ij} ها متغیرهای تصمیم هستند که مقدار جریان روی کمان (i,j) را مشخص می‌کنند.

فرمول‌بندی این مساله به صورت زیر می‌باشد:

$$min \quad \sum_{(i,j)\in A} C_{ij}x_{ij}$$

$$s.t \quad \sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(i,j)\in A\}} x_{ji} = b(i) \quad , \qquad \forall i \in V \qquad (۲-۱)$$

$$۰ \le x_{ij} \le u_{ij}$$

محدودیت‌های فوق شروط شدنی بودن نامیده می‌شوند.

در ابتدا برای حل اینگونه مسائل باید شرط $\sum_{i\in V} b(i) = ۰$ برقرار باشد، در غیر این صورت با اضافه نمودن گره‌ها و کمان های مصنوعی این شرط برقرار می‌شود. محدودیت اول را محدودیت تعادل جریان و محدودیت دوم را محدودیت کران جریان می‌نامند. عبارت $\sum_{\{j:(i,j)\in A\}} x_{ij}$ بیانگر مجموع جریان های خروجی از گره iاُم و $\sum_{\{j:(i,j)\in A\}} x_{ji}$ بیانگر مجموع جریان ورودی به گرهiاُم می‌باشد و مقدار $\sum_{(i,j)\in A} C_{ij}x_{ij}$ مقدار تابع هدف به ازای جریان x می‌باشد. برای رسیدن به جواب بهینه الگوریتم های مختلفی از جمله الگوریتم حذف دور منفی و سیمپلکس شبکه استفاده می‌شوند. اگر $Cycle = (i_a, i_۱, \dots, i_k, i_a)$ یک دور جهت‌دار در شبکه G باشد هزینه آن با $\sum_{(i,j)\in Cycle} C_{ij}x_{ij}$ تعریف می‌گردد.

الگوریتم سیمپلکس شبکه

۱) یک درخت پوشای شدنی (یک پایه شدنی) در نظر بگیرید.
۲) مقدار جریان های پایه ای (مقدار جریان کمان های درخت) را محاسبه کنید.
۳) $w_m = ۰$, و برای هر کمان پایه ای(i,j)، $w_i - w_j - c_{ij} = ۰$، قرار داده و پتانسیل گره های دیگر را بدست آورید.

۴) برای کمان های غیرپایه ای $w_i - w_j - c_{ij}$ را تعیین کنید.

۵) اگر بازای هرکمان غیر پایه ای (i,j)، $w_i - w_j - c_{ij} \leq ۰$ توقف، جواب فعلی بهینه است.

۶) اگر $w_p - w_q - c_{pq} > ۰$ کمان (p وq) را به درخت اضافه کنید.

۷) مقدار جریان $x_{pq} = \Delta$ قرار دهید و بیشترین مقدار Δ را مثلا Δ^* با توجه به جهت کمان های روی دور ایجاد شده تعیین نمائید.

۸) Δ^* را روی کمان های دور تحمیل کنید.

۹) کمانی که جریان آن صفر است از درخت خارج می شود.

۱۰) به قدم ۳ بروید.

کوتاه ترین مسیر در شبکه جریان

شبکه ی G با m گره و n یال و هزینه c_{ij} مربوط به هر یال (i,j) درG مفروض است. مسئله کوتاه ترین مسیر عبارت است از پیدا کردن کوتاه ترین(کم هزینه ترین) مسیر از گره ی ۱ به گره m درG. هزینه ی مسیر مجموع هزینه های یال های تشکیل دهنده ی آن مسیر است. مسئله کوتاه ترین مسیر را می توان در قالب شبکه های جریان طراحی کرد. به این صورت که اگر بخواهیم در یک شبکه یک واحد جریان را از گره ی ۱ به گره m با کمترین هزینه بفرستیم، از این رو $b_۱ = ۱$ و$b_m = -۱$ و $b_i = ۰$ به ازای m یا ۱ ≠i مدل ریاضی آن به صورت زیر تنظیم می گردد.

$$\min \sum_{i=۱}^{m} \sum_{j=۱}^{m} c_{ij} x_{ij}$$

$$\text{s.t} \quad \sum_{j=۱}^{m} x_{ij} - \sum_{k=۱}^{m} x_{ki} = \begin{cases} ۱ & i = ۱ \\ ۰ & i \neq ۱ \text{یا} m \\ -۱ & i = m \end{cases}$$

$$x_{ij} = ۰ \text{ یا } ۱ \qquad i,j = ۱,۲,\dots,m$$

مقدار جریان روی کمان ها در این حالت باید صفر یا یک باشد و نشان دهنده ی آن است که هر یال در مسیر هست یاخیر.

نکته: اگر به جای $x_{ij} = ۰,۱$ قرار دهیم $x_{ij} \geq ۰$ آنگاه مسئله به یک مسئله برنامه ریزی خطی عادی تبدیل می شود و به صورت زیر ارائه می شود.

$$\min \sum_{i=۱}^{m} \sum_{j=۱}^{m} c_{ij} x_{ij}$$

$$\text{s.t} \quad \sum_{j=۱}^{m} x_{ij} - \sum_{k=۱}^{m} x_{ki} = \begin{cases} ۱ & i = ۱ \\ ۰ & i \neq ۱ \text{یا} m \\ -۱ & i = m \end{cases}$$

$$x_{ij} \geq ۰ \qquad i,j = ۱,۲,\dots,m$$

مدل برنامه ریزی خطی فوق را به راحتی می توان با استفاده از الگوریتم سیمپلکس شبکه حل نمود.

برنامه ریزی خطی

تخصیص منابع محدود به فعالیتهای تعریف شده جهت افزایش بازدهی و یافتن بهترین راه حل بهینه را برنامه ریزی خطی می گویند. در واقع برنامه ریزی خطی نوع ساده ای از مدل برنامه ریزی ریاضی می باشد که بهترین گزینه را از میان روشهای ممکن انتخاب می کند و در آن تابع هدف و محدودیت ها همگی به صورت خطی نمایش داده می شوند.

بخش های اصلی مدل برنامه ریزی خطی بصورت زیر است:

۱) تابع هدف : حداکثر کردن یا حداقل کردن عملکرد مدل می باشد.

۲) محدودیت های کارکردی (قیود): بیانگر محدودیت های منابع جهت رسیدن به اهداف مدل می باشد که به صورت بزرگتر مساوی (≥) کوچکتر مساوی (≤) و یا مساوی (=) نمایش داده می شود .

۳) محدودیت های غیر کارکردی یا متغیرهای تصمیم : نشان دهنده مقدار عملکرد یا سطح یک فعالیت بوده و با x_j() نمایش داده می شود این متغیرها می توانند بصورت مثبت، منفی و یا آزاد در علامت مورد استفاده قرار گیرند.

پویا

برنامه ریزی پویا با به کارگیری فرایند های نظام گرا، ترکیبی از تصمیمات متوالی را معین میکند که به ماکزیمم شدن راندمان (کارایی) محاسبات منتهی میگردد. وقتی برنامه ریزی پویا برای حل یک مسئله به کار میرود، تصمیم گیری های چند مرحله ای برای دنباله ای از مسائل اتخاذ می گردد. یعنی روش برنامه ریزی پویا یک مسئله ی تصمیم گیری چند مرحله ای را به دنباله ای از مسائل یک مرحله ای تبدیل می کند . از این رو یک مسئله N متغیره به N مسئله یک متغیره تبدیل می گردد، که با حل پی در پی این مسائل، مسئله اصلی حل خواهد شد . امتیاز این عمل در آن است که مسائل جزئی در مقایسه با مسئله اصلی، مسائلی بسیار ساده و کوچک هستند. بر خلاف برنامه ریزی خطی، چارچوب استانداردی جهت فرمول کردن مسئله برنامه ریزی پویا وجود ندارد. در واقع، آنچه برنامه ریزی پویا انجام میدهد، ارائه روش کلی جهت حل این نوع مسائل است. آنچه در کل می توان درباره به کارگیری برنامه ریزی پویا می توان گفت، آن است که در هر مورد باید معادلات و روابط ریاضی مخصوصی که با شرایط مسئله منطبق است، نوشته و به کار گرفته شوند.

دوگان مسئله جریان با مینیمم هزینه

دوگان جریان با حداقل هزینه چندین توضیح مختلف دارد. از آنجا که همه الگوریتم ها برای حل مسئله اولیه می تواند عامل بالقوه گره اولیه $(\pi(i))$ و متغیرهای α_{ij} را ایجاد کند. حل مسئله اولیه همیشه مسائل دوگان و اولیه را حل می کند.

مشابه این مورد در حل مسئله دوگان عمدتا مسئله اولیه را نیز حل می کند. اکثر الگوریتم ها برای حل مسائل جریان شبکه به طور واضح و یا ضمنی از مشخصه های متغیرهای دوگان استفاده می کنند. چون آنها گره های پتانسیل هستند که در هر قسمت استفاده می شوند. و جزئی از برنامه خطی دوگان هستند. مخصوصا اینکه مسئله دوگان عاملی ایجاد می کند که بتوانیم جواب دوگان عملی پیدا کنیم که مقدار تابع مشابهی مثل جواب اولیه داده شده داشته باشد. از قضیه دوگان قوی می توان فهمید که بدون انجام محاسبات اضافی و بدون در نظر گرفتن جواب های مطلوب دیگر بایستی جواب های اولیه مطلوب و بهینه باشند. این روند در تایید تفکر قوی در بهینه سازی شبکه و طرحی کلی و جامع است. همواره در بسیاری از پیشرفت های قبلی از آن استفاده کردیم و در موارد زیادی هم دوباره آنها را خواهیم دید.

در مسائل جریان شبکه مسائل دوگان و اولیه از طریق کوتاهترین مسیر اصلی و مسائل جریان ماکزیمم به هم مرتبط هستند. در حقیقت این روابط کمک می کنند تا اهمیت اساسی این دو مسئله اصلی در مورد تئوری جریان شبکه و الگوریتم ها را دریابیم.

ساختارداده‌ای برای درخت های پویا

ساختار(ساختمان)داده برای نگهداری مجموعه‌ای از درخت های با رئوس ناهمبند تحت دو عملیات(عملگر)[1] مطرح می‌شود.یک عملگر اتصال که دو درخت را با افزودن یک یال ترکیب کرده و به یک درخت تبدیل می‌کند.و دیگری عملگر برش که هر درخت را با حذف یک یال، به دو درخت تقسیم می‌کند.هر عملگر به زمان $o(\log n)$ نیاز دارد. با استفاده از این ساختمان داده، می توان الگوریتم‌های سریع جدیدی برای مسائل زیر به دست آورد.

۱) محاسبه‌ی نزدیک‌ترین اجداد مشترک[2].

۲) حل مسائل گوناگون شبکه جریان که شامل پیدا کردن جریان های ماکزیمم، جریان های انسداد و جریان های دوری است.

۳) محاسبه‌ی انواع خاصی ازدرخت های فراگیر مینیمم.

۴) پیاده سازی الگوریتم شبکه ای سیمپلکس برای جریانهای با مینیمم هزینه.

مسئله ی دوم بهترین کاربرد را در بین مسائل دیگر دارد، که از آن الگوریتمی با زمان o(mnlogn) برای پیدا کردن ماکزیمم جریان در یک شبکه با n رأس و m یال به دست می آید که خود از قبل سریعترین الگوریتم شناخته شده برای گراف های اسپارس[۳] می باشد. مجموعه‌ای از درخت های ریشه‌ای با رأس های ناهمبند داده شده است. می‌خواهیم درخت‌ها را با ساختار(ساختمان) داده‌ای نمایش دهیم که این امکان را می‌دهد تا اطلاعات درخت‌ها را به راحتی خلاصه کنیم و به سهولت این ساختمان را به منظور انعکاس تغییرات ناشی از سه عملگر زیر بروز رسانی نماییم.

عملگر link(v, w) : اگرv ریشه‌ی یک درخت و w رأس درخت دیگری باشد، به وسیله ی اضافه نمودن یال (v, w) دو درخت را به هم متصل می کند و با این کار w را والد v می کند.

عملگر cut (v) : اگرگره v ریشه‌ی یک درخت نباشد، درختی که شاملv است را با حذف یالی که از v به والد(پدر) خودش می‌رود، به دو درخت تقسیم می کند.

عملگر evert(v): درختی را که شامل رأسv است، با تبدیلv به ریشه‌ی درخت، درخت را واژگون(پشت و رو) می نماید. بنابراین ساختمان داده‌ای را ارائه می‌دهیم که مسائل درخت‌های پویا را حل کند. در این فصل دو نوع ساختمان داده، را ارائه می دهیم که نوع اول، کران زمانی(o(logn را برای هر عملگروقتی که زمان روی بدترین حالت دنباله ای از عملگرها هدر رفته است دارد. و نوع دوم که کمی پیچیده تر است و بدترین حالت کران زمانo(logn) را برای هر عملیات اجرایی (عملگر) دارد.

نکته: اگر f و g توابعی بر حسب x باشند، علامت$f(x)$ همان(o(g(x) است، به این معنا که مقادیر ثابت و مثبتc_1 وc_2 موجود است که $f(x) \leq c_1 g(x) + c_2 . \forall x;$

بنابراین ساختمان داده را برای طراحی الگوریتم سریع جدیدی برای مسائل نظریه‌ی گراف زیر استفاده می‌کنیم.

۱) محاسبه‌ی نزدیک‌ترین اجداد مشترک در زمان (o(logn برای هر عملگر.

۲) محاسبه‌ی انواع مختلفی از شبکه های جریان که شامل جریان های ماکزیمم در زمان (o(nmlogn) و جریان های انسداد در زمان(o(nmlogn) و جریان های دوری در زمان (o(mlogn)است(هنگام بحث در مورد مسائل گراف منظور از n تعداد راس ها و m تعداد یال های روی گراف است).

۳) محاسبه‌ی انواع خاصی ازدرخت های پوشا مینیمم در زمان(o(mlogn

۴) پیاده سازی الگوریتم شبکه ای سیمپلکس برای انتقال مسائل به گونه ای که به ازای هر مرحله، بروز رسانی جواب درخت به زمان (o(logn نیاز داشته باشد.

می توان اجرای ساختمان داده ی که مطرح کردیم و انجام تمرین های تجربی برای مشخص کردن ارزش اجرایی آن ها، اگر ممکن باشد را مد نظر قرار داد. سلیتور یک الگوریتم درخت پویا را برای جریان های شبکه ای ماکزیمم مطرح کرد، که توسط یک عامل ثابت(در حدود دو) از الگوریتم دینیتس کند تر است البته به جز در مثال گراف های خاص ساخته شده در بدترین حالت. این مورد نشان می دهد که در گراف های تولید شده به صورت تصادفی تنها یک کسر کوچک می تواند باعث تضعیف الگوریتم دینیتس شود.

برخی تجربه ها که در آنها روش به الگوریتم سیمپلکس اعمال شده می تواند ارزنده باشد. نسخه ی هدر رفت زمانی ساختمان داده ی ما از درخت های دودویی پیش مقداری شده ی محلی استفاده می کند. در حالی که نسخه ی بدترین حالت از درخت های دودویی پیش مقداری شده ی سراسری استفاده می کند.

درخت های سراسری پیش مقدار شده، در روش هدر رفت زمانی هم کار می کنند اما درخت های پیش مقدار شده ی محلی در نسخه ی بدترین حالت کار نمی کنند. تمرین بیشتر برای گسترش و توضیح ساختمان داده ارزشمند است. اخیرا محققین نوع جدیدی از درخت تحقیقی را پیدا کرده اند که آن را درخت تحقیقی خود پیوند[۴۰] نامیده اند. می توانند جایگزین درخت های دودویی پیش فرض شده در ساختمان داده ی ارائه شده در بخش۳-۳ و ۳-۴ شود که توسط تارجان در [۴۱] ارائه شده است.

دوگان جریان با مینیمم هزینه در شبکه های مسطح

مسئله جریان با مینیمم هزینه یکی از مسائل مهم در علوم کامپیوتر است. همچنین یکی از مهمترین مسائل مطرح شده در شبکه های جریان است. هدف از ارائه فصل چهارم بررسی جریان با مینیمم هزینه درشبکه های مسطح[۱۶] است. در مسئله جریان با مینیمم هزینه هر یال شبکه یک هزینه دارد و باید یک جریان درست و صحیح که دارای کمترین هزینه است را در این شبکه یافت. جریان درست وصحیح جریانی است که محدودیت ظرفیت، کران پایین یال ها و همچنین تعادل[۱۷] رئوس را حفظ کند. در حقیقت جریان در شبکه های جریان یک تابع حقیقی به صورت $\mathcal{F}: v \times v \rightarrow R$ است که باید محدودیت ها روی رئوس و یال ها را تامین نماید. هر گراف مسطح یک گراف مسطح هندسی دوگان دارد به طوری که هریال از گراف اولیه به یک رأس دوگان و همچنین هر رأس اولیه به یک یال دوگان مربوط می شود. هر مسئله ی برنامه ریزی خطی یک مسئله ی برنامه ریزی خطی دوگان دیگر دارد بطوریکه هر محدودیت اولیه به یک متغیر دوگان وهر متغیر اولیه به یک محدودیت دوگان، مربوط می شود. مسئله اولیه دارای جواب محدود و بهینه است اگر و فقط اگرمسئله ی دوگان یک جواب داشته باشد. با مسئله ی جریان گردشی[۱۸] با مینیمم هزینه در شبکه های مسطح این فصل را شروع می کنیم.

ابتدا مسئله را در گراف دوگان هندسی مطرح کرده و سپس دوگان برنامه ریزی خطی رابرای این مسئله پیدا میکنیم. هزینه های مسئله ی اصلی محدودیت تعادل رئوس را در مسئله ی جدید تعریف می کند و همچنین ظرفیت های یال ها در مسئله ی اصلی، هزینه ها را در مسئله ی جدید تعریف می کنند. در این فصل سادگی ساختار دوگان هندسی گراف های مسطح بیرونی[۱۹]را به کار برده ویک الگوریتم برای مسئله ی جریان با مینیمم هزینه برای این نوع گراف ها ارائه داده که در زمان o(n$log^{۲}$n) (n تعداد رئوس) به نتیجه می رسد.

دو حالت وجود دارد که مسئله ی ورودی دارای جواب بهینه ی محدود نشود.

حالت اول: ممکن است یک حلقه با هزینه ی منفی در شبکه ی ورودی وجود داشته باشد.در این حالت الگوریتم nU+1 واحد جریان در طی این حلقه ارسال می کند، زیرا که ظرفیت تمام یال ها را به nU+1 رسیده است این تنها حالتی است که یک یال دارای nU+1 واحد جریان است بنابراین این حالت به راحتی قابل تشخیص است.

حالت دوم: جواب منطقی و درستی برای مسئله وجود ندارد. این حالت را این گونه تشخیص می دهیم که تمام نسخه های v_o را از $\widehat{T}$ برش می دهیم و $e_f(v_o)$ همچنان بزرگ تر از صفر است. هر عملیات درخت پویا به زمان o(log^n) نیاز دارد و در هر تکرار یک یال از درخت حذف می شود بنابراین زمان کل اجراء جهت توازن (تعادل) v_o و v_c، o(nlog^n) است. الگوریتم o(log^n) سطح بازگشت دارد بنابراین زمان اجراء کل برابر o(n$log^{۲}n$) است.

نکته: مسئله ی جریان با مینیمم هزینه در شبکه های جریان مسطح بیرونی با n رأس می تواند در زمان o(n$log^{۲}n$) حل شود.

منابع

1) A. V. Karzanovd,Dtermining the maximal flow in a network by the method of preflows, Sovief Math. Dokl. 15 (1974), 434437.

2) A. V. AHO, J. E. HOPCROFT, AND J. D. ULLMAN, On finding lowest common ancestors in trees SIAMJ. Comput. 5 (1975), 115-132.

3) Ahuja, R. K., Magnanti, T. L., Orlin, J. B.: Network Flows: Theory, Algorithms and Applications. Prentice-Hall, New Jersey (1993).

4) Bland, R. G., Jensen, D. L.: On the computational behavior of a polynomial-time network flow algorithm. Math. Program. 54, 1–39 (1992)

5) C. V. CHVATAL“,L inear Programming,” Freeman, San Francisco, 1983, to appear

6) Chambers, E. W., Erickson, J., Nayyeri, A.: Homology flows, cohomology cuts. In: Proceed- ings of the 41st annual ACM Symposium on Theory of Computing, pp. 273–282. ACM, New York (2009)

7) Cornelsen, S., Karrenbauer, A.: Accelerated bend minimization. J. Graph. Appl. 16, 635–650 (2012)

8) Cornelsen, S., Karrenbauer, A., Li, S.: Leveling the grid. In: Bader, D. A., Mutzel, P. (eds.) ALENEX 2012, pp. 45–54. SIAM (2012).

9) D. E. KNUTH, “The Art of Computer Programming, Vol. 3: Sorting and Searching,” Addison-Wesley, Reading, Mass., 1974.

10) D. MAIER, An efficient method for storing ancestor information in trees, SIAM J. Comput. 8 (1979), 599-618.

11) D. D. SLEATOR, “An O(nm log n) Algorithm for Maximum Network Flow,” Tech. Rep. STAN-CS- 80-83 1, Computer Science Department, Stanford University, Stanford, Calif., 1980

12) D. D. SLEAT~R AND R. E. TARIAN, A data structure for dynamic trees, in “Proc. Thirteenth Annual ACM Symp. on Theory of Computing,” pp. 114-122, 1981.

13) D. D. SLEATOR AND R. E. TARIAN, Self-adjusting binary trees, in “Proc. Fifteenth Annual ACM Symp. on Theory of Computing,” pp. 235-245, 1983.

14) D. HAREL AND R. E. TARJAN, Fast algorithms for finding nearest common ancestors, SIAM J. Cornput., to appear.

15) Daitch, S. I., Spielman, D. A.: Faster approximate lossy generalized flow via interior point algorithms. In: Proceedings of the 40th annual ACM Symposium on Theory of Computing, pp. 451–460. ACM, New York (2008)

16) E. A. DINITS, An algorithm for the solution of a problem of maximal flow in a network with power estimation, Soviet Math. Dokl. I1 (1970), 1277-1280.

17) Edmonds, J., Karp, R. M.: Theoretical improvements in algorithmic efficiency for network flow problems. J. ACM 19, 248–264 (1972)

18) Fleischner, H. J., Geller, D. P., Harary, F.: Outerplanar graphs and weak duals. J. Indian Math. Soc. 38, 215–219 (1974). Cited in [24]

19) Frederickson, G. N.: Planar graph decomposition and all pairs shortest paths. J. ACM 38, 162–204 (1991)

20) Hopcroft, J., Tarjan, R.: Efficient Planarity Testing. J. ACM 21, 549–568 (1974)

21) Hassin, R.: Maximum flow in (s,t) planar networks. Inf. Process. Lett. 13, 107 (1981)

22) Hassin, R., Johnson, D. B.: An O(nlog2 n) algorithm for maximum flow in undirected planar networks. SIAM J. Comput. 14, 612–624 (1985)

23) Henzinger, M. R., Klein, P., Rao, S., Subramania, S.: Faster shortest-path algorithms for planar graphs. J. Comput. Syst. Sci. 55, 3–23 (1997)

24) Henzinger, M. R., King, V.: Randomized fully dynamic graph algorithms with polylogarithmic time per operation. J. ACM 46, 502–516 (1999).

25) Haim Kaplan and Yahav Nussbaum Min-Cost Flow Duality in Planar Networks

26) Imai, H., Iwano, K.: Efficient sequential and parallel algorithms for planar minimum cost flow. In: Asano T., Ibaraki T., Imai H., Nishizeki T. (eds.) SIGAL 1990. LNCS, vol. 450, pp. 21–30. Springer, Heidelberg (1990).

27) J. NIEVERGELT AND F. M. REINGOLDB, inary search trees of bounded balance, SIAM J. Cornput. 2 (1973), 33-43.

28) Khuller, S., Naor, J., Klein, P.: The lattice structure of flow in planar graphs. SIAM J. Disc. Math. 63, 477–490 (1993)

29) L. G. GUIBAS AND R. SEDGEWICK, A dichromatic framework for balanced trees, in "Proc. Nineteenth Annual IEEE Symp. on Foundations of Computer Science," pp. 8-21, 1978.

30) Miller, G. L., Naor, J.: Flow in planar graphs with multiple sources and sinks. SIAM J. Comput. 24, 1002–1017 (1995)

31) Mozes, S., Wulff-Nilsen, C.: Shortest paths in planar graphs with real lengths in O(nlog2 n/loglogn) time. In: ESA 2010. LNCS, vol. 6347, pp. 206–217. Springer, Heidel- berg (2010)

32) Orlin, J. B.: A faster strongly polynomial minimum cost flow algorithm. Operations Research 41, 338–350 (1993).

33) R. E. TARJAN, Efficiency of a good but not linear set union algorithm, J. Assoc. Cornput. Mach. 22 (1975), 215-225.

34) R. E. TARJAN,A pplications of path compression on balanced trees, 1. Assoc. Cornput. Mach. 26 (1979), 690-715.

35) R. E. TARIAN, “Data Structures and Network Algorithms,” Society for Industrial and Applied Mathematics, Philadelphia, Penn., to appear.

36) R. E. TARJAN AND J. VAN LEEUWEN, Worst-case analysis of set union algorithms, J. Assoc. Cornput. Mach., to appear.

37) Ro¨ck, H.: Scaling techniques for minimal cost network flows. In: Discrete Structures and Algorithms: Proceedings of the Worskshop WG 79, pp. 181–191. Carl Hansen, Munich (1980)

38) S. W. BENT, D. D. SLEATOR, AND R. E. TARJAN, Biased 2-3 trees, in “Proc. Twenty-First Annual IEEE Symp. on Foundations of Computer Science,” pp. 248-254, 1980.

39) S. W. BENT, D. D. SLEATOR, AND R. E. TARJAN, “Biased Search Trees,” SIAM J. Cornput., to appear

40) Sleator, D. D., Tarjan, R. E.: A data structure for dynamic trees. J. Comput. Syst. Sci. 26, 362–391 (1983)

41) Schrijver, A.: Combinatorial Optimization : Polyhedra and Efficiency. Algorithms and Com- binatorics 24. Springer (2004)

42) Sys lo, M. M.: Characterizations of outerplanar graphs. Discrete Math. 26, 47–53 (1979)

43) Tarjan, R. E.: Dynamic trees as search trees via Euler tours, applied to the network simplex algorithm. Math. Program. 78, 169–188 (1997)

44) Vaidyanathan, B., Ahuja, R. K.: Fast algorithms for specially structured minimum cost flow problems with applications. Oper. Res. 58, 1681–1696 (2010)

45) Y. SHILOACH, “An O(nZ log2 I) Maximum-Flow Algorithm,” Tech. Rep. STAN-CS-78-702, Computer Science Department, Stanford University, Stanford, Calif., 1978.

46) Z. GALIL AND A. NAAMAD, An Q(EVlog* V) algorithm for the maximal flow problem, J. ComputSystem Sci. 21 (1980), 203-217.