





دانشکده علوم ریاضی
گروه ریاضی کاربردی

پایان نامه

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

عنوان

مساله مکان یابی پشتیبان

استاد راهنما

دکتر جعفر فتحعلی

استاد مشاور

دکتر نادر جعفری راد

پژوهشگر

کبری مهدور

بهمن ۱۳۹۱

نام خانوادگی دانشجو: مهدور

نام: کبری

عنوان: مساله مکان‌یابی پشتیبان

استاد راهنما: دکتر جعفر فتحعلی
استاد مشاور: دکتر نادر جعفری راد

مقطع تحصیلی: کارشناسی ارشد رشته: ریاضی کاربردی گرایش: تحقیق در عملیات

دانشگاه: صنعتی شاهرود تاریخ فارغ‌التحصیلی: بهمن ۱۳۹۱
دانشکده علوم ریاضی تعداد صفحات: ۷۸

واژگان کلیدی: مکان‌یابی، میانه، پشتیبان، نیمه‌ناخوشایند، وزن مثبت و منفی

چکیده

تئوری مکان‌یابی به‌طور سنتی علاقه‌مند به مسایلی است که وزن سرویس‌دهنده‌ها و در دسترس بودن آن‌ها دقیقاً مشخص است. اما اغلب یک برآورد دقیق از تمامی این پارامترها امکان‌پذیر نیست. بخش مهمی از داده‌های زندگی واقعی، اغلب شامل عدم قطعیت است و این پارامترها ممکن است با زمان تغییر کنند. بنابراین در سال‌های اخیر مدل‌های مکان‌یابی شامل عدم قطعیت بررسی شده، برخی حالت‌های آن‌ها تعریف و مورد مطالعه قرار گرفته‌است. یک حالت خاص از این مساله مدل قابلیت اطمینان است که در این مدل بعضی اوقات ممکن است سرویس‌دهنده‌ها شکست بخورند و مشتریان اختصاص داده‌شده به این سرویس‌دهنده‌ها مجبورند از سرویس‌دهنده‌های در حال کار سرویس بگیرند. مساله ۲- میانه نیز همانند مدل قابلیت اطمینان بررسی شده که در آن هر سرویس‌دهنده ممکن است با یک احتمال معین شکست بخورد. مساله پشتیبان ۲- میانه همانند مدل قابلیت اطمینان است که در آن به‌دنبال یافتن یک زوج از رئوس هستیم که مجموع فاصله وزنی از تمام رئوس به سرویس‌دهنده‌های در حال کار را مینیمم کند. با توجه به اهمیت موضوع، مدل‌های مختلفی که تا کنون در این زمینه مطرح شده‌اند را مورد مطالعه قرار داده و با ارایه الگوریتمی به بررسی مساله پشتیبان ۲- میانه نیمه ناخوشایند می‌پردازیم.

اگر تنها ترین تنها شوم، باز خدا هست
او جانشین همه نداشته های من است...

تقدیم به کوه های کرانمایه زندگیم،
پدر و مادرم

آنان که ناتوان شدند تا به توانایی برسم
و موهایشان سپید گشت تا رو سفید شوم،
و همسر مهربانم

کسی که آفتاب را به زندگی ام هدیه نمود
و عاشقانه سوخت تا کرم با نخش وجود و روشنگر را هم باشد.

شکر و قدردانی

با نام آنکه خلق کرد هستی را از رحمت بی‌انتهای خویش تا انسان در قدوم خلقتش به پهنای گسترده‌ای به بزرگی جهان خیمه زند و به او عقل داد تا عظمت خلقت را دریابد.

اکنون که به یاری خداوند متعال این دوره تحصیلیم را به پایان رسانده‌ام، هرچند واژه‌ها را یارای آن نیست که لطف و محبت آنانی را که در تمام دوران زندگیم جرعه‌نوش دریای مهر و محبت‌شان بوده‌ام به تصویر بکشم، اما به رسم ادب و احترام بوسه بر دستان‌شان زده و بر خود واجب می‌دانم زحمات بی‌شائبه پدر و مادر عزیزم، صبر و فداکاری همسر مهربانم و یاری برادر و خواهران نازنینم را که همواره راه گشای مشکلاتم در تمام مراحل زندگی بوده‌اند، ارج نهاده و مراتب تشکر قلبی و باطنی را از مهربانی‌های آنان ابراز دارم.

با تمام وجود و در اوج فروتنی از مقام شامخ استاد گرانمایه جناب آقای دکتر فتحعلی که در نهایت صبر و بزرگواری تمامی سعی و تلاش خود را در جهت اعتلای واقعی ارزش‌های آموزشی در کالبد هدایت‌ها و رهنمودها نسبت به اینجانب مبذول فرمودند، کمال قدردانی را می‌نمایم و با امتنان بیکران از مساعدت‌های جناب آقای دکتر جعفری‌راد و جناب آقای دکتر ناظمی و تشکر خالصانه از تمامی کسانی که به نوعی مرا در به انجام رساندن این مهم یاری نموده‌اند، در برابر وجود گرامی‌شان زانوی ادب بر زمین می‌نهم.

کبری‌مدور
بهمن ۱۳۹۱

فهرست مطالب

۱	مقدمه ای بر مسایل مکانیابی	۱
۲	۱.۱ مقدمه	۱.۱
۳	۲.۱ اجزا مسائل مکانیابی	۲.۱
۳	۱.۲.۱ فضا	۱.۲.۱
۸	۳.۱ تعاریف و مفاهیم اولیه	۳.۱
۱۱	۴.۱ مساله p-میانه	۴.۱
۱۴	۵.۱ مساله پشتیبان ۲-میانه	۵.۱
۱۴	۶.۱ مساله ۲-میانه نیمه ناخوشایند	۶.۱
۱۶	۲ مساله پشتیبان ۲-میانه روی درخت های با رئوس وزن دار	۲
۱۷	۱.۲ مقدمه	۱.۲
۱۷	۲.۲ تعریف مساله و نماد گذاری	۲.۲
۱۸	۳.۲ روش حل مساله	۳.۲
۳۰	۳ مساله پشتیبان ۲-میانه روی گراف های بلوکی	۳
۳۱	۱.۳ مقدمه	۱.۳
۳۱	۲.۳ تعاریف اولیه	۲.۳
۳۳	۳.۳ مساله ۱-میانه روی گراف های بلوکی	۳.۳
۳۶	۴.۳ مساله پشتیبان ۲-میانه روی گراف های بلوکی	۴.۳
۴۷	۴ مساله ۲-میانه با وزن مثبت-منفی روی درخت	۴
۴۸	۱.۴ مقدمه	۱.۴
۴۸	۲.۴ تعریف مساله و نمادگذاری	۲.۴
۵۱	۳.۴ مساله ۲-میانه با وزن مثبت-منفی روی درخت	۳.۴

۶۱	۵	مساله پشتیان ۲- میانه نیمه ناخوشایند روی درخت
۶۲	۱.۵	مقدمه
۶۲	۲.۵	تعریف مساله
۶۳	۳.۵	چند خاصیت روی درخت ها
۶۶	۴.۵	مساله پشتیان ۲- میانه نیمه ناخوشایند روی مسیر
۶۹	۵.۵	مثال
۷۰		مراجع
۷۳		واژه‌نامه فارسی به انگلیسی
۷۵		واژه‌نامه انگلیسی به فارسی

فصل ۱

مقدمه ای بر مسایل مکانیابی

۱.۱ مقدمه

مسایل مکان‌یابی مسایلی هستند که وقتی به دنبال یافتن محل و نحوه استقرار بهینه یک یا چند فعالیت (سرویس دهنده^۱) بر اساس عوامل و متغیرهای موثر بر مکان‌یابی هستیم با آن‌ها روبه‌رو می‌شویم. معمولاً در این مسایل فرض بر این است که تعدادی مشتری (سرویس گیرنده، نقطه ثابت، نقطه تقاضا) موجودند و می‌خواهیم بهترین مکان برای استقرار سرویس‌دهندگان جدید را به گونه‌ای بیابیم که هزینه حمل و نقل، تأثیرات محیطی نامطلوب، زمان سرویس‌دهی، سود، کیفیت، نحوه توزیع سرویس و رعایت انصاف و برابری در آن، کسب بیشترین سهم در بازار و ... بسته به شرایط مساله، بهینه شوند.

مکان‌یابی تسهیلات و تاسیسات از کاربردی‌ترین زمینه‌های تحقیقاتی است که در بسیاری از شاخه‌های علوم گسترده شده‌است. از جمله این شاخه‌ها می‌توان به تحقیق در عملیات، علوم مدیریت، مهندسی صنایع، جغرافیا، اقتصاد، علوم کامپیوتر، ریاضیات، بازاریابی، مهندسی برق، برنامه‌ریزی شهری، علوم نظامی و دیگر شاخه‌های مرتبط اشاره کرد. از کاربردهای معروف مسایل مکان‌یابی می‌توان مکان‌یابی انبارها، کارخانه‌ها، ماشین‌آلات کارخانه‌ها، بیمارستان‌ها، نمایندگی‌های فروش، آتش نشانی‌ها، پاسگاه‌های پلیس، پایانه‌های مسافری، اجزا الکترونیکی در مدارها، آژیر خطر، چاه‌های نظامی، تاسیسات نظامی، نیروگاه‌ها، آزمایشگاه‌ها و غیره را نام برد. مسایل ترکیبی از مسیریابی و مکان‌یابی نیز در این حوزه مورد مطالعه و بررسی قرار می‌گیرند.

زمان پیدایش مسایل مکان‌یابی به اوایل قرن هفدهم و مساله‌ای که فرما^۲ مطرح کرد برمی‌گردد. مساله‌ای که فرما مطرح کرد به این صورت است: "یک مثلث را در نظر بگیرید که وزن هریک از رئوس آن ۱+ باشد. نقطه چهارم را به گونه‌ای بیابید که مجموع فواصل وزنی آن تا رئوس مثلث کمینه گردد." توریکلی^۳ مشاهده کرد که محل برخورد دایره‌های محیطی مثلث‌های متساوی‌الاضلاعی که بر روی اضلاع مثلث اصلی و رو به بیرون آن ساخته می‌شوند، جواب مساله است. نظریه مکان‌یابی به شکلی که امروزه مورد استفاده قرار می‌گیرد، با منتشر شدن کتاب آلفرد وبر^۴ در سال ۱۹۰۹ با عنوان *Vber den standortder industrien* متولد شد [۲]. وبر در این کتاب نتایج تحقیقات خود در مورد صنایع کارخانه‌ای را ارائه کرد. بیست سال قبل از ارزیابی تئوری مکان‌یابی صنعتی توسط وبر، لانهارد^۵ به‌طور جزئی‌تر به این امر همت گماشته بود. لانهارد کوشش کرد نشان دهد که چگونه می‌توان مکان‌یابی بهینه را در شرایط ساده‌ای که دو منبع مواد اولیه وجود دارند و یک بازار به وسیله رئوس یک مثلث نشان داد (شکل (۱.۱)) [۲].

نظریه وبر دارای سه اصل اساسی است. اول این که محل مواد اولیه معلوم است. دوم این که مکان و میزان مصرف مشتریان مشخص است. تعداد این مکان‌ها متناهی است و جدا از هم هستند. هر تولیدکننده

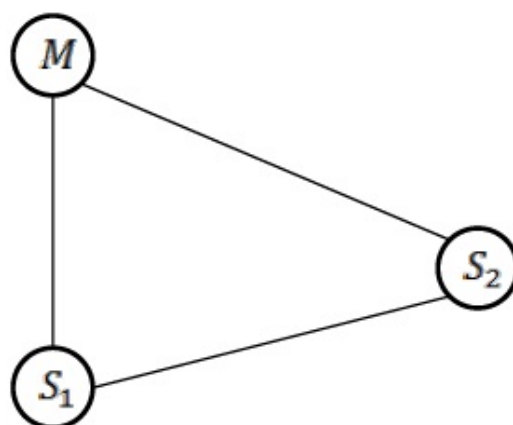
^۱ Facility

^۲ Fermat

^۳ Torricelli

^۴ Alfred weber

^۵ Launhardt



شکل ۱.۱: مدل لانهارد با دو منبع و یک بازار

بازار نامحدودی در اختیار دارد و امتیازات انحصاری از نظر انتخاب مکان وجود ندارد. سوم این که چندین مکان ثابت برای عرضه نیروی کار وجود دارد به طوری که این نیروی کار دارای قابلیت تحرک و جابه جایی نبوده و عرضه آن نامحدود است.

مطالعات جدی بر روی مکان یابی از زمانی شروع شد که در سال ۱۹۶۴ حکیمی [۳۰]، تابع هدف را به دو صورت کم ترین مجموع و مینیماکس مطرح کرد. او همچنین مطالعاتی بر روی مسایل مکان یابی روی شبکه انجام داد که نقطه شروع مکان یابی گسسته مدرن است. معمولاً در مسایل مکان یابی سرویس دهنده هایی که می خواهیم مکان استقرار آن ها را بیابیم، مطلوب در نظر گرفته می شوند و تنها تاثیرات مثبت شان را به حساب می آوریم.

۲.۱ اجزا مسائل مکانیابی

در این بخش ترکیبی از طبقه بندی هایی که از مسایل مکان یابی و تعیین ویژگی های آن ها و اجزای شان صورت گرفته است، ارائه می شود.

۱.۲.۱ فضا

معمولاً فضای^۶ی مسایل مکان یابی یکی از سه دسته زیر است:

- (۱) پیوسته: هرگاه فضای مکان تاسیسات و نقاط تقاضا پیوسته باشد یعنی توسط متغیرهایی که به صورت پیوسته تغییر می کنند، نظیر مختصات مشخص شود، مدل را پیوسته می گویند. در این مدل ها مکان سرویس دهنده ها را در فضای d بعدی $\mathbb{R}^d$ پیدا می کنیم. اکثر مسایل مکان یابی در فضای حداقل دو بعدی تعریف می شوند. مسایل دوبعدی به خاطر ویژگی های هندسی و قابلیت ادراک بصری شان بیشتر

^۶Space

مورد توجه قرار می گیرند. در مواجهه با مسایل پیوسته از ابزارهای هندسی و آنالیز (آنالیز تابعی، که معمولاً از ویژگی های تحذب و تفعر استفاده می شود) بهره می بریم. عمدتاً مسایل مکانیابی طبیعت و توصیف هندسی دارند ولی مطالعه و روند حل آنها هم شامل بررسی های هندسی و هم آنالیزی است. در کاربردهای عملی از فضای پیوسته به دو شکل استفاده می شود. مسطح، مثل مدارهای مجتمع در مدارهای الکتریکی و الکترونیکی، یک تکه کاغذ، میز کار و یا یک تکه زمین و کروی برای نواحی مثل قاره ها و حتی کره زمین. در این حالت ها برای مشخص کردن نقاط احتیاج به دو مولفه داریم. در مسایلی که با مکانیابی در ساختمان زیر آب و یا در هوا روبه رو هستیم، باید مشخصه سومی که ارتفاع یا عمق را نشان می دهد را هم به حساب آوریم. مسایل یک بعدی عمدتاً زمانی رخ می دهند که مکانیابی را روی یک خط (مستقیم یا دارای انحنا) انجا می دهیم. مثل کشیدن یک بزرگراه یا خط آهن.

(۲) **گسسته:** مدل های مکانیابی گسسته مدلهایی هستند که در آنها باید مکان سرویس دهندگان را تنها روی نقاط از پیش تعیین شده پیدا کنیم. در این مدل ها مجموعه ای متناهی از نقاط کاندیدا داده شده است که توسط متغیرهای گسسته نشان داده می شوند. در مدل های مکانیابی پیوسته بر خلاف مدل های گسسته نمی توان لیست جامع و کاملی از نقاط در دسترس را ارایه داد. از این جهت مدل های پیوسته را مدل های ایجاد مکان و مدل های گسسته را مدل های انتخاب مکان نیز خوانده اند. یکی از ابزارهای مفید در ریاضیات در مواجهه با مسایل بزرگ و پیوسته ایده گسسته سازی فضا است. یکی از دلایل استفاده از این ایده ساده تر شدن و کاهش فضای جواب به تعداد متناهی مجموعه و کاربردهای ویژه در بعضی حالات خاص از مساله است. در مدل های گسسته، در حقیقت هر نقطه نماینده یک ناحیه از فضا است. قابل توجه است که در حالت پیوسته باید یک تابع برای تخمین فاصله بین نقاط انتخاب شود که نحوه انتخاب یک تابع مناسب، خود موضوع مقالات زیادی در زمینه مکانیابی پیوسته بوده است ولی در حالت گسسته، فاصله واقعی پیموده شده و یا از قبل محاسبه شده، مورد استفاده قرار می گیرد. بنابراین مدل های پیوسته به راحتی موقعیت های عملی و واقعی را مدل می کنند و به سرعت اجرا می شوند زیرا به بانک های اطلاعاتی عظیم مسافت ها و همچنین محاسبه کوتاه ترین مسیر احتیاجی ندارند. این نکته قابل ذکر است که به خاطر استفاده تابع تخمین دقت کمتری نسبت به انواع گسسته دارند. البته انتخاب نوع فضا به نظر تصمیم گیرنده بستگی دارد. بعضی اوقات ممکن است مزایای مدل های پیوسته سبب شود که از دقت آن تاحدی صرف نظر کرد. مثلاً در یک شهر ممکن است که انتخاب فاصله مستطیلی^۷ (منهتن) دقت بسیار زیادی در تخمین فاصله واقعی بین دو نقطه از آن را داشته باشد. عمدتاً شهرهایی که دارای خیابان های عمود بر هم و توری شکل هستند در این دسته قرار می گیرند.

^۷ Rectilinear

(۳) **شبکه‌ای:** مدل‌های شبکه‌ای مدل‌هایی هستند که در آن سرویس دهندگان روی رئوس و یا یال‌های یک شبکه مستقر می‌شوند. در حوزه مکان‌یابی با فضای شبکه‌ای، مسایلی وجود دارند که در حالت کلی NP -سخت هستند ولی در گراف‌های خاص مثل درخت یا k -درخت‌ها در زمان‌های حداکثر از درجه چندجمله‌ای قابل حل هستند. علاوه بر موارد ذکر شده ممکن است فضایی که باید سرویس دهنده در آن قرار گیرد با تعریف نواحی ممنوع محدود شود. نواحی ممنوع نواحی هستند که سرویس دهنده نباید در آن‌ها قرار گیرد. این مناطق می‌توانند قوانین منطقه‌بندی یا سایر محدودیت را مدل کنند.

تعداد سرویس دهنده (یا فعالیت) هایی که باید مستقر شوند (p)

فرض کنید p تعداد سرویس دهنده‌ها باشد. حالت‌های $p = 1$ و $p > 1$ را در نظر می‌گیریم:

(۱) $p = 1$ ؛ در این حالت فقط یک فعالیت (سرویس دهنده) باید مستقر شود.

(۲) $p > 1$ ؛ در این حالت علاوه بر پیدا کردن محل استقرار سرویس دهنده‌ها باید چگونگی تخصیص مشتریان به سرویس دهنده‌ها را نیز مشخص کنیم.

در حالتی که چند سرویس دهنده را مکان‌یابی می‌کنیم، ممکن است ارتباط بین سرویس دهنده‌های جدید نیز دارای اهمیت باشد. مدل‌هایی که این هدف را برآورده می‌کنند شامل فاصله وزن‌دار بین سرویس دهنده‌های جدید می‌باشند. به عنوان مثالی از این مورد می‌توان به مکان‌یابی تاسیسات نظامی اشاره کرد که در آن پراکنده شدن و فاصله داشتن سرویس دهنده‌های جدید میزان ضرر و خسارت و آسیب‌پذیری را کاهش می‌دهد. در مکان‌یابی پمپ‌های گاز یا فست فودها تصمیم‌گیرنده سعی در پراکنده کردن آن‌ها دارد تا از ایجاد رقابت مستقیم آن‌ها با یک‌دیگر جلوگیری کند.

تعداد سرویس دهنده‌های موجود

بر مبنای این پارامتر دو نوع مدل مختلف وجود دارد:

(۱) مدل‌هایی که تعداد سرویس دهنده‌های جدید در آن ثابت است.

(۲) مدل‌هایی که تعداد سرویس دهنده‌های جدید در آن متغیر است.

مدل‌های نوع دوم پیچیده‌تر از نوع اول هستند، زیرا باید متغیری دیگر که نشان‌دهنده تعداد سرویس دهنده‌ها است را به مدل اضافه کنیم. در مدل‌های مکان‌یابی دو حالت رخ می‌دهد. در بسیاری از مدل‌ها فرض بر این است که تصمیم‌گیرنده با یک فضای خالی روبه‌رو می‌شود که هیچ سرویس دهنده‌ای در آن وجود ندارد. در حالتی دیگر که آن را مکان‌یابی شرطی^۸ می‌نامیم، تعدادی سرویس دهنده موجود هستند و هدف اضافه کردن تعدادی دیگر است.

^۸Conditional location

اهداف تصمیم گیرنده

مدل می تواند یک یا چند هدفه باشد. یعنی تصمیم گیرنده می تواند به دنبال مکان بهینه برای استقرار سرویس دهنده تحت یک هدف یا چند هدف به طور هم زمان باشد. در بین مدل های تک هدفه حداقل سه دسته وجود دارند:

(۱) مینیم سازی تابعی از فاصله: کمترین مجموع^۹، مینیماکس^{۱۰}

(۲) ماکزیم سازی تابعی از فاصله: بیشترین مجموع^{۱۱}، ماکزیمین^{۱۲}

(۳) تابع های ترکیبی از فاصله

اندازه گیری فاصله

فاصله^{۱۳} یک معیار عددی برای تشخیص میزان دور بودن اشیا است. در مسایل مکانیابی فرض می کنیم که ارتباط بین تاسیسات مستقیما وابسته به موقعیت مکانی آنها باشد و معمولا به شکل تابعی از فاصله بیان می شود. بنابراین مفهوم فاصله نقشی محوری در مکانیابی پیوسته دارد. با توجه به موقعیت و شرایط مساله اندازه های مختلفی برای ارزیابی فاصله به کار می روند. استفاده از روند نرم ها^{۱۴} به عنوان تابع تخمین فاصله در یک مساله مکانیابی با فضای پیوسته اولین بار توسط لاو و موریس [۲۳] پیشنهاد شد. وارد و وندل [۴۸] نیز استفاده از نرم های بلوکی^{۱۵} را مورد مطالعه قرار دادند.

محدودیت های فاصله

در بسیاری از مدل ها محدودیت^{۱۶} فاصله وجود دارد. محدودیت های فاصله بر دو نوع هستند:

(۱) کران بالا برای فاصله: در این مدل ها سرویس دهنده (ها) به اندازه کافی دور از هم قرار می گیرند ولی تا حدی که در دسترس نیز باشند.

(۲) کران پایین برای فاصله: در این مدل ها سرویس دهنده (ها) تا حد امکان نزدیک هم هستند اما نه بیش از یک حد معین.

^۹ Minisum

^{۱۰} Minimax

^{۱۱} Maxisum

^{۱۲} maximin

^{۱۳} Distance

^{۱۴} Round norm

^{۱۵} Block norms

^{۱۶} Restrictions

وزن

فاصله‌ها در تابع هدف می‌توانند دارای وزن‌های ثابت یا متغیر، یکسان یا مختلف باشند. گاهی برای مشخص کردن میزان تقاضا در هر نقطه ثابت (مشتریان) از وزن‌ها استفاده می‌شود. در مسایل شبکه‌ای، وزن یال می‌تواند نشان‌دهنده طول مسیر، زمان پیمایش یال، میزان مخاطره و هزینه استفاده از یال و آرایه خدمات و مواردی از این قبیل باشد.

ارتباط بین سرویس دهنده‌ها

حداقل سه نوع ارتباط مختلف بین سرویس دهنده‌ها می‌تواند وجود داشته باشد:

- (۱) فقط فاصله بین مشتریان و سرویس دهنده‌ها در نظر گرفته شود، مثلاً در مکان‌یابی چندمحل دفن زباله در یک کشور.
- (۲) فقط فاصله بین سرویس دهندگان محاسبه شود، مثلاً مکان‌یابی سایت‌های موشکی در یک ناحیه بدون جمعیت.
- (۳) هر دو نوع فاصله ذکر شده در بالا در نظر گرفته شود، مثلاً مکان‌یابی تعدادی سایت هسته‌ای در یک ناحیه دارای جمعیت.

مشتریان

در تعدادی از مدل‌های مکان‌یابی نقش مشتریان^{۱۷} نیز منظور شده است. لذا لازم است نحوه توزیع و میزان تقاضای آن‌ها را بدانیم. برای مثال ممکن است فرض کنیم مشتریان به‌طور یکنواخت در یک ناحیه توزیع شده‌اند و یا در نقاط خاصی از فضا، مثلاً رئوس یک شبکه قرار دارند و میزان تقاضا تابعی از فاصله باشد. مشتریان ممکن است به یک سرویس دهنده اختصاص یابند و یا ممکن است در انتخاب آن آزاد باشند که در این صورت سوال این است که آیا همیشه مشتریان به نزدیک‌ترین سرویس دهنده مراجعه می‌کنند یا عوامل دیگری در انتخاب سرویس دهنده توسط مشتریان دخیل هستند.

تابع هدف

در این قسمت دسته‌های اصلی تابع‌های هدف^{۱۸} در مدل‌های موجود مکان‌یابی را معرفی می‌کنیم. اکثر این توابع پیشنهادی هستند. یک راه، تفکیک سرویس دهنده‌ها بر اساس نوع سرویس‌شان به سرویس دهنده‌ها است که به سرویس دهنده‌های عمومی و خصوصی تقسیم می‌شوند. اغلب موارد خصوصی متناظر با کمترین مجموع و موارد عمومی متناظر با مینیماکس است، زیرا هدف کمترین مجموع سرویس دهنده‌ها را به گونه

^{۱۷}Clients

^{۱۸}Objective Functions

ای مکانیابی می کند که مجموع هزینه های حمل و نقل مینیمم شود. در مکانیابی سرویس دهنده های عمومی مانند استخر یا کتابخانه، می خواهیم مکانی را پیدا کنیم که به همه مشتریان نزدیک باشد. مورد دیگری که در تابع هدف مد نظر قرار می گیرد، میزان مطلوبیت سرویس دهنده برای مشتریان است. سرویس دهنده ها از دید مشتریان یکی از سه حالت زیر را دارند:

(۱) مطلوب: در این حالت سرویس گیرنده می خواهد تا جایی که امکان دارد سرویس دهنده به او نزدیک باشد، یعنی سعی در جذب سرویس دهنده دارد.

(۲) نامطلوب (مضر): سرویس گیرنده می خواهد تا جایی که امکان دارد از سرویس دهنده فاصله بگیرد، یعنی سعی در دفع آن دارد.

(۳) تا حدی مطلوب و تا حدی نامطلوب: در این حالت سرویس گیرنده می خواهد به سرویس دهنده نزدیک باشد ولی نه خیلی نزدیک. مثلاً یک سوپر مارکت را در نظر بگیرید، از یک طرف مشتریان می خواهند که برای راحتی در امر خرید به فروشگاه نزدیک باشند و از طرف دیگر ترافیک، شلوغی و سروصدای ناشی از فروشگاه، آن را نامطلوب و ناخوشایند می سازد. در چنین موقعیتی سرویس گیرنده سه انتخاب دارد:

- با اعمال یک محدودیت روی فاصله بین شان سرویس دهنده را دفع کند، یعنی سرویس دهنده تا حد امکان نزدیک باشد ولی حداقل به اندازه فاصله از پیش تعیین شده فاصله داشته باشد.
- با اعمال یک محدودیت روی فاصله مابین شان سرویس دهنده را جذب کند، یعنی سرویس دهنده تا حد امکان دور باشد ولی حداکثر به اندازه فاصله از پیش تعیین شده فاصله داشته باشد.
- ساخت یک هدف ترکیبی جاذبه-دافعه، مثلاً ترکیب محدودیتی از دو هدف جاذب و دافع.

۳.۱ تعاریف و مفاهیم اولیه

تعاریف و مفاهیم گراف

در این بخش مفاهیم مربوط به گراف که در این پایان نامه با آن ها سروکار داریم را بیان می کنیم. خواننده برای دیگر قضایای گراف گراف که در این بخش بیان نشده می تواند به هراری^{۱۹} [۳۲] مراجعه کند. فرض کنید $N = (V(N), E(N))$ یک گراف ساده بدون جهت با مجموعه رئوس $V(N)$ و مجموعه یال های $E(N)$ باشد.

تعریف ۱.۳.۱. اگر گراف N یک گراف وزن دار باشد، برای یک رأس $v \in V(N)$ وزن آن را با $w_N(v)$ و مجموع وزن رئوس گراف N را با $W(N) = \sum_{v \in V(N)} w_N(v)$ نشان می دهیم.

^{۱۹}Harary

تعریف ۲.۳.۱. برای هر دو رأس u و v در گراف N $P_N[u, v]$ کوتاه‌ترین مسیر از رأس u به v می‌باشد.

تعریف ۳.۳.۱. گراف N را همبند ^{۲۰} گویند هرگاه بین هر دو رأس دلخواه u و v از N ، مسیری در N موجود باشد. درغیراین صورت N را ناهمبند نامند.

تعریف ۴.۳.۱. اگر u و v در N همبند باشند، آنگاه مجموع طول یال‌های موجود در کوتاه‌ترین مسیر بین u و v در N را فاصله‌ی بین u و v گوئیم و با $d_N(u, v)$ نشان می‌دهیم. مسیر بین دو رأس N ، که بیشترین فاصله را از هم داشته باشند، مسیر قطری N نامیم و با $diam(N)$ نمایش می‌دهیم.

تعریف ۵.۳.۱. فاصله میان رأس v و زیرمجموعه $U \subseteq V(N)$ را با $d_N(v, U)$ نمادگذاری می‌کنیم به طوری که $d_N(v, U) = \min_{u \in U} d_N(u, v)$.

تعریف ۶.۳.۱. برای هر دو رأس u و v در گراف N نقطه میانی ^{۲۱} مسیر $P[u, v]$ را با m_{uv} نشان می‌دهیم که $d_N(u, m_{uv}) = d_N(v, m_{uv})$. یال e یال مرکزی u و v نامیده می‌شود اگر شامل نقطه میانی مسیر $P[u, v]$ باشد.

تعریف ۷.۳.۱. برای گراف همبند N رأس v یک رأس برشی ^{۲۲} است اگر تمام یال‌های متصل به رأس v را در گراف N حذف کنیم، زیرگراف حاصل که همان $N - v$ است، ناهمبند باشد.

تعریف ۸.۳.۱. زیرگراف همبند ماکسیمال تولید شده در N که فاقد رأس برشی باشد، بلوک ^{۲۳} نامیده می‌شود.

تعریف ۹.۳.۱. گرافی که هر دو رأس متمایز آن مجاور باشد را گراف کامل گوئیم. گراف کامل با n رأس را با K_n نشان می‌دهیم.

تعریف ۱۰.۳.۱. گراف N یک گراف بلوکی است هرگاه هر بلوک در N کامل باشد، یعنی هر دو رأس در یک بلوک توسط یالی به هم متصل باشند. فرض کنید $B(N) = \{B_1, B_2, \dots, B_l\}$ مجموعه بلوک‌های N باشد. بدیهی است $l \leq n - 1$ و تساوی زمانی برقرار خواهد بود که N یک درخت با $|V(N)| > 1$ باشد. در این پایان‌نامه فرض می‌کنیم که در گراف بلوکی هر دو یال در یک بلوک B دارای طول یکسان می‌باشند.

تعریف ۱۱.۳.۱. گراف بازه‌ای ^{۲۴} گرافی است که رئوس آن متناظر با بازه‌های باز اعداد حقیقی است و رئوسی به هم وصل می‌شوند که بازه‌های متناظر با آن‌ها اشتراک داشته باشند. اگر یک گراف دارای حفره یعنی دوری با اندازه بزرگ‌تر از ۳ باشد که هیچ یالی بین رئوس غیر متوالی در آن دور نباشد، حتماً بازه‌ای نیست.

^{۲۰} Connected

^{۲۱} Mid point

^{۲۲} Cut-vertex

^{۲۳} Block

^{۲۴} Interval graph

تعریف ۱۲.۳.۱. گراف N را یک گراف مسطح^{۲۵} گوئیم هر گاه بتوان آن را روی سطح صاف (و یا روی سطح کره) به گونه ای رسم کرد که هیچ دو یالی یکدیگر را قطع نکنند.

تعریف ۱۳.۳.۱. هر درخت T گره خاصی به نام ریشه^{۲۶} دارد که کلیه گره های دیگر درخت در پایین آن قرار دارند. گره ریشه والدی ندارد. هر درخت تنها شامل یک گره ریشه است.

تعریف ۱۴.۳.۱. برای رأس x در درخت ریشه دار T ، تعداد رأس های موجود روی مسیر $P_T[x, r]$ را $level(x)$ می نامیم. اگر ریشه را در سطح یک فرض کنیم برحسب اینکه یک گره نسبت به ریشه در چه ردیفی باشد شماره سطح می گیرد.

تعریف ۱۵.۳.۱. ارتفاع^{۲۷} درخت برابر با بیشترین سطح گره ها در درخت یا سطح دورترین برگ است. ارتفاع درختی که تنها گره ریشه را دارد صفر است.

تعریف ۱۶.۳.۱. اگر در درخت ریشه دار T رأس x مجاور y باشد و $level(x) = level(y) + 1$ آنگاه y پدر x است که با $par(x)$ نشان داده می شود و x فرزند y می باشد و با $C(y)$ نمادگذاری می شود.

تعریف ۱۷.۳.۱. اگر T ریشه دار باشد، زیردرخت ریشه دار شده در $u \in V(T)$ با T_u نشان داده می شود که یک درخت تولید شده با تمام فرزندان u است.

تعریف ۱۸.۳.۱. برای هر دو رأس u و v در درخت ریشه دار T اگر u روی مسیر $P_T[v, r]$ باشد و $level(v) = level(u) + 2$ آنگاه u پدر سویی رأس v می باشد که آن را با نماد $DVP(v)$ نمایش می دهیم.

تعریف ۱۹.۳.۱. برای هر زوج راس (u, v) در گراف N قرار دهید $\Pi(u, v) = (U, V)$ که

$$U = \{x : d(x, u) \leq d(x, v)\}, \quad V = V(T) - U.$$

در واقع دو مجموعه U و V افراز رأسی مجموعه رئوس $V(N)$ می باشند.

تعریف ۲۰.۳.۱. در پیمایش یک درخت، می خواهیم با حرکت روی یال های آن، همه گره های درخت را یک بار ملاقات کنیم. در روش پیمایش پس ترتیب^{۲۸} ابتدا فرزندان، سپس خود گره ملاقات می شود. در این پیمایش از سمت چپ ترین رأس که در پایین ترین سطح قرار دارد، شروع کرده و به سمت راست حرکت می کنیم. پس از پیمایش تمام فرزندان یک گره، خود گره را ملاقات کرده و دوباره به سمت چپ ترین گره که در پایین ترین سطح قرار دارد و تاکنون ملاقات نشده است، برمی گردیم. این روند را تا پیمایش ریشه انجام می دهیم.

^{۲۵} Planar graph

^{۲۶} Root

^{۲۷} Height

^{۲۸} Postorder

۴.۱ مساله p-میانه

در این بخش از مرجع [۱] استفاده شده است. فرض کنید $N = (V(N), E(N))$ یک گراف بدون جهت با وزن رئوس مثبت و طول یال‌های مثبت باشد. حال باید p سرویس‌دهنده را روی یال‌ها یا رأس‌های N مکان‌یابی کنیم به گونه‌ای که مجموع فاصله وزنی رئوس تا نزدیک‌ترین سرویس‌دهنده مینیمم شود. مساله p - میانه^{۲۹} کاربردهای فراوانی دارد که از آن جمله می‌توان به مکان‌یابی مراکز شبکه‌های ارتباطی کامپیوتری، مراکز توزیع کالا، مراکز اداری، مراکز نظامی، ایستگاه‌های اتوبوس و مراکز پستی اشاره کرد.

در مساله p - میانه، میانه‌ها می‌توانند هر نقطه‌ای از شبکه انتخاب شوند، که در این حالت مساله را مساله p - میانه محض^{۳۰} می‌گویند. اما اگر هدف پیدا کردن میانه تنها روی رأس‌ها باشد مساله را مساله p - میانه رأسی^{۳۱} نامند. در ادامه هر جا از اصطلاح «مساله p - میانه» استفاده شد منظور مساله p - میانه محض است.

مساله p - میانه خاصیت بهینگی رأسی^{۳۲} دارد به این معنا که ادعا می‌کند یک جواب بهینه موجود است که تمام سرویس‌دهنده‌ها را بر روی رئوس N مکان‌یابی می‌کند. با توجه به این خاصیت اگر بخواهیم مساله را به صورت شمارشی حل کنیم برای انتخاب p - میانه از n رأس نیاز به $\binom{n}{p}$ انتخاب داریم، مثلاً برای $p = ۱$ به n انتخاب نیاز است که برای هر رأس باید مقدار تابع هدف را محاسبه کنیم یعنی به زمان $O(n^۲)$ نیاز می‌باشد که در آن $n = |V|$. در نتیجه مساله p - میانه برای هر $p \in N$ ثابت روی گراف‌های عمومی را می‌توان با محاسبه مقدار تابع هدف برای هر جواب شدنی با زمان چندجمله‌ای پاسخ داد. اما زمانیکه p بخشی از ورودی است، کریو^{۳۳} و حکیمی^{۳۴} [۳۵] نشان دادند که مساله p - میانه حتی برای گراف‌های مسطح^{۳۵} با ماکزیمم درجه ۳، NP - سخت است. بنابراین بسیاری از بررسی‌ها به کلاس‌های خاصی از گراف‌ها اختصاص داده می‌شود که الگوریتم‌هایی با کارایی بیشتر به دست می‌آید. به‌ویژه برای شبکه درختی تامیر^{۳۶} [۴۵]، یک الگوریتم با پیچیدگی زمانی $O(pn^۲)$ پیشنهاد داد که تا کنون سریع‌ترین الگوریتم برای این نوع مساله بوده است. برای p ثابت و کوچک در درخت‌ها، هوآ^{۳۷} [۳۳] و گلدمن^{۳۸} [۲۷] به‌طور مستقل برای $p = ۱$ یک الگوریتم با پیچیدگی زمانی خطی به دست آوردند و گاویش^{۳۹} و سرایدر^{۴۰} [۲۶] الگوریتمی با پیچیدگی زمانی $O(n \log n)$ برای $p = ۲$ طراحی کردند. علاوه بر این نتایج بسیار خوب دیگری

^{۲۹} p-Median problem^{۳۰} Pure p-Median^{۳۱} Vertex p-Median^{۳۲} Vertex Optimality Property^{۳۳} Karive^{۳۴} Hakimi^{۳۵} Planar Graphs^{۳۶} Tamir^{۳۷} Hua^{۳۸} Goldman^{۳۹} Gavish^{۴۰} Sridhar

برای کاکتوس، چرخ و گراف های دیگر موجود است. بورکارد^{۴۱} و کراروپ^{۴۲} [۱۵] مساله ۱-میانه روی کاکتوس را با استفاده از الگوریتمی با زمان خطی پاسخ دادند. هاتزل^{۴۳} [۲۴] یک الگوریتم با پیچیدگی زمانی $O(n^2)$ برای مساله ۲-میانه روی کاکتوس و یک الگوریتم با زمان خطی برای مساله ۱-میانه روی چرخ ارائه کرد. اگر وزن تمام رئوس منفی باشد، مساله میانه متناظر با مساله p -ماکسین خواهد بود. در مرجع [۱۲]، بورکارد و همکارانش نشان دادند که مساله p -ماکسین روی درخت می تواند در زمان خطی پاسخ داده شود. کنگ^{۴۴} و چنگ^{۴۵} همین مساله را روی گراف های بلوکی که عمومی تر از درخت ها هستند بررسی کردند و یک الگوریتم با زمان خطی پیشنهاد دادند [۳۴]. اخیرا کنگ و چنگ [۱۸] مساله p -ماکسین را روی گراف های بازه ای با زمان خطی حل کردند. زمانی که تنها یک بخش از وزن های رئوس منفی هستند مساله میانه، مساله p -میانه با وزن مثبت-منفی نامیده می شود. در ادامه مساله ۱-میانه و ۲-میانه که حالت هایی از مساله p -میانه هستند را توضیح می دهیم.

مساله ۱-میانه

[۹] علاوه بر خاصیت بهینگی رأسی که در حالت کلی برای مساله p -میانه برقرار است، در حالت $p = 1$ یک خاصیت دیگر روی درخت ها برقرار می باشد، که به خاصیت اکثریت^{۴۶} معروف است. این خاصیت که توسط گلدمن در [۲۷] ارائه شده به صورت زیر می باشد،

قضیه ۱.۴.۱. فرض کنید T یک درخت باشد و T_1 و T_2 دو زیر درخت متناظر با افرازی از رأس های T باشند. آنگاه T_1 شامل یک جواب برای مساله ۱-میانه است اگر و تنها اگر داشته باشیم،

$$\sum_{\{i: v_i \in T_1\}} w_i \geq \frac{1}{2} \sum_{\{i: v_i \in T\}} w_i$$

با استفاده از قضیه فوق الگوریتمی با پیچیدگی زمانی $O(|V|)$ برای مساله ۱-میانه روی درخت ها به صورت زیر در [۳۸] ارائه شده است که در آن $W(V)$ مجموع وزن رأس های درخت می باشد.

الگوریتم اکثریت

(۱) اگر T شامل تنها یک رأس است، توقف کنید، همان رأس نقطه یک میانه است.

(۲) برای هر رأس انتهایی مانند v_i ، اگر $w(v_i) \geq \frac{W(V)}{2}$ آنگاه به گام ۴ بروید، در غیر این صورت به گام ۳ بروید.

^{۴۱} Burkard

^{۴۲} Krarup

^{۴۳} Hatzl

^{۴۴} Kang

^{۴۵} Cheng

^{۴۶} Majority property

(۳) فرض کنید v_j رأس مجاور v_i باشد، و یال (v_i, v_j) را حذف کنید و قرار دهید

$$w(v_i) = w(v_i) + w(v_j)$$

و به گام ۱ بروید.

(۴) توقف کنید، v_i یک ۱-میانه است.

مساله ۲-میانه

با توجه به مطالبی که قبلاً گفته شد ملاحظه می‌شود که اگر بخواهیم مساله ۲-میانه روی شبکه ^{۴۷} را به روش شمارشی حل کنیم به زمان $O(n^3)$ نیاز داریم. برای این مساله در حالتی که شبکه به صورت درخت است می‌توان از روش حذف یال ^{۴۸} استفاده کرد که در ادامه توضیح آن آمده است.

روش حذف یال [۳۵]: در این روش با حذف یک یال از درخت T آن را به دو زیردرخت T_1 و T_2 تقسیم می‌کنیم. در هر یک از این دو زیردرخت، نقاط ۱-میانه را می‌یابیم و مقدار تابع هدف را به ازای آن‌ها محاسبه می‌کنیم. سپس یال دیگری از درخت را حذف کرده و ۱-میانه‌ها را در دو زیردرخت حاصل به دست می‌آوریم. عمل حذف را برای تمام یال‌های درخت انجام می‌دهیم. نقاط ۱-میانه دو زیردرخت متناظر با حذف یالی که تابع هدف به ازای آن‌ها کمترین مقدار را دارد، نقاط ۲-میانه برای درخت خواهد بود.

مشاهده می‌شود که در این روش ابتدا عمل تخصیص و سپس عمل انتخاب میانه‌ها انجام می‌شود. در حالی که در روش شمارشی ابتدا میانه‌ها انتخاب می‌شوند بعد عمل تخصیص صورت می‌گیرد. در روش فوق چون $n - 1$ یال در هر درخت وجود دارد و به ازای هر یال برای محاسبه ۱-میانه نیاز به زمان $O(n)$ داریم، بنابراین این روش در کل در زمان $O(n^2)$ می‌تواند انجام شود. روش فوق را میرچندانی و آدجیت [۴۰] تنها با حذف یال‌های خاصی بهبود بخشیدند. روش آن‌ها بر قضیه زیر که اثبات آن را می‌توان در [۳۵] یافت، استوار است.

قضیه ۲.۴.۱. فرض کنید m نقطه ۱-میانه برای درخت T و m_1 و m_2 نقاط ۲-میانه برای آن باشند. در این صورت m روی مسیر بین m_1 و m_2 قرار دارد.

البته روش میرچندانی و آدجیت در بدترین حالت دارای همان پیچیدگی زمانی $O(n^3)$ است ولی در اکثر مسایل، تعداد یال‌هایی که برای حذف در نظر گرفته می‌شوند خیلی کمتر از روش معمول است. با استفاده از روش حذف یال، گاوش و اسرایدر [۲۶] الگوریتمی با پیچیدگی زمانی $O(n \log n)$ برای مساله ۲-میانه روی درخت ارائه کرده‌اند.

^{۴۷}Network

^{۴۸}Edge Deletion Method

۵.۱ مساله پشتیبان ۲-میانه

تئوری مکانیابی به طور سنتی علاقه مند به مسائلی است که وزن سرویس دهنده ها و در دسترس بودن آنها دقیقاً مشخص است. اما اغلب این امکان پذیر نیست که یک برآورد دقیق از تمامی این پارامترها داشته باشیم. بخش مهمی از داده های زندگی واقعی، اغلب شامل عدم قطعیت است و این پارامترها ممکن است با زمان تغییر کنند. بنابراین در سال های اخیر مدل های مکان یابی شامل عدم قطعیت بررسی شده، برخی حالت های آن ها تعریف و مورد مطالعه قرار گرفته است. اشنایدر و دسکین [۴۳]، مدل قابلیت اطمینان را بررسی کردند که در این مدل بعضی اوقات ممکن است سرویس دهنده ها شکست بخورند و مشتریان اختصاص داده شده به این سرویس دهنده ها مجبورند از سرویس دهنده های در حال کار سرویس بگیرند. وانگ وو و همکارانش [۴۹] [۴۷] مساله ۲-میانه را همانند مدل قابلیت اطمینان بررسی کردند در حالی که هر سرویس دهنده ممکن است با یک احتمال داده شده شکست بخورد. به محض این که یک سرویس دهنده شکست بخورد، با این فرض که سرویس دهنده ها نمی توانند به طور هم زمان شکست بخورند، سرویس دهنده دیگر مسئولیت سرویس دهی را به عهده می گیرد. آن ها این مدل را که در آن به دنبال یافتن یک زوج از رئوس هستیم که مجموع فواصل وزنی از تمام رئوس به سرویس دهنده های در حال کار را مینیمم کند، مساله پشتیبان ۲-میانه^{۵۰} نامیدند و یک الگوریتم با زمان $O(n \log n)$ پیشنهاد دادند که n تعداد رئوس درخت مورد بررسی است. در فصل (۲) به طور مفصل الگوریتم و روش حل مساله را مطرح کرده ایم. همچنین چنگ و کنگ

مساله پشتیبان ۲-میانه را روی گراف های بلوکی بررسی کرده و الگوریتمی با زمان $O(n \log n + m)$ ارائه کردند که در فصل (۳) این پایان نامه آورده شده است. n تعداد رئوس و m تعداد یال های گراف بلوکی مورد بررسی است.

۶.۱ مساله ۲-میانه نیمه ناخوشایند

در سال های اخیر، سرویس دهندگان نیمه ناخوشایند^{۵۱}، یعنی سرویس دهنده هایی که مطلوب است به دور از بخشی از مشتریان مکانیابی شوند توجه بسیاری را به خود جلب کرده است. در شبکه ها، اولین مدل مکانیابی با وزن های مثبت و منفی توسط بورکارد و کراروپ [۱۵] ارایه شد. آن ها یک الگوریتم با زمان خطی را برای مساله ۱-میانه روی کاکتوس یعنی گراف همبندی که هیچ یک از دو دور در آن بیشتر از یک رأس مشترک نداشته باشند، بسط دادند. بورکارد و همکارانش [۱۱] مساله ۲-میانه نیمه ناخوشایند روی درخت ها را بررسی کردند. آن ها دو تابع هدف متفاوت را پیشنهاد دادند. اولین تابع هدف سعی دارد مجموع کمترین فاصله وزنی روی تمام رئوس (MWD) را مینیمم کند در حالی که تابع هدف دوم مجموع وزن

^{۴۹}Wang Wu

^{۵۰}Backup 2-Median Problem

^{۵۱}Semi obnoxious facility

مینیمم‌ترین فاصله از v به سرویس دهنده را روی تمام رئوس v ، (WMD) مینیمم می‌کند. برای مدل MWD ، آن‌ها تعدادی از قضایای مساله را به کار برده و برای یافتن ۲-میانه یک الگوریتم با زمان $O(n^2)$ روی درخت، یک الگوریتم با زمان $O(n \log n)$ روی ستاره و یک الگوریتم با زمان خطی روی مسیر مطرح کردند. اما برای مدل MWD ، آن‌ها تنها توانستند یک الگوریتم روی درخت که در زمان $O(n^3)$ اجرا می‌شود را بیابند که اگر میانه‌ها را تنها روی رئوس جستجو کنیم پیچیدگی زمانی آن به $O(n^2)$ کاهش می‌یابد. این نتایج توسط بنکوکزی و همکارانش [۹] بهبود یافتند. آن‌ها نشان دادند که برای مدل MWD ، ۲-میانه روی درخت با رئوس با وزن مثبت و منفی می‌تواند در زمان $O(n \log n)$ پاسخ داده شود. آن‌ها همچنین برای مدل WMD ، یک الگوریتم با زمان $O(nh \log^2 n)$ که h ارتفاع درخت را نشان می‌دهد، پیشنهاد دادند. زمانی که تنها وزن منفی داشته باشیم، بورکارد و همکارانش [۱۳] نشان دادند که جواب بهینه مساله ۲-میانه با تابع هدف MWD ، در زمان خطی به دست می‌آید. زمانی که شبکه مورد بررسی یک دور باشد، بورکارد و هاتزل [۱۴] برای مساله ۲-میانه یک الگوریتم با زمان خطی برای مدل MWD و یک الگوریتم با زمان $O(n^2)$ برای مدل WMD به کار بردند. برای حالت $p = 3$ بورکارد و فتحعلی [۱۲]، الگوریتمی با زمان $O(n^5)$ برای مدل WMD روی درخت و الگوریتمی با زمان $O(n^3)$ وقتی که میانه‌ها محدود به رئوس هستند، ارائه کردند.

فصل ۲

مساله پشتيان ۲-ميانه روي درخت هاي با
رئوس وزن دار

۱.۲ مقدمه

در این فصل مساله پشتیبان ۲-میانه روی درخت‌های با رئوس وزن‌دار بررسی می‌کنیم که وزن رئوس مثبت می‌باشد. با بیان این‌که خاصیت بهینگی رأسی برای پشتیبان p -میانه نیز برقرار است، همان روش حذف یال که برای مساله ۲-میانه کلاسیک در فصل (۱) بیان شده را می‌توان به کار برد. برپایه همین روش وانگ و همکارانش [۴۷] الگوریتم $BACKUP_2MEDIAN$ را که شامل دو زیربرنامه $LOWER-MEDIAN$ و $UPPER-MEDIAN$ می‌باشد را با پیچیدگی زمانی $o(n \log n)$ برای مساله پشتیبان ۲-میانه روی درخت پیشنهاد دادند که n تعداد رئوس‌های درخت مورد بررسی می‌باشد.

تعاریف و نمادگذاری‌های استفاده شده در این فصل در بخش بعد آورده شده است. الگوریتم‌های پیشنهادی برای حل مساله در بخش آخر آورده شده است. در انتهای فصل نیز مثالی برای این الگوریتم بیان کرده‌ایم.

۲.۲ تعریف مساله و نماد گذاری

در این بخش مساله ۲-میانه و نمادهای استفاده شده در این فصل را تعریف می‌کنیم. فرض کنید G گراف ساده بدون جهت $G = (V(G), E(G), w, l)$ باشد که $V(G)$ مجموعه رئوسها و $E(G)$ مجموعه یال‌های G ، $w : V(G) \rightarrow R^+$ تابع وزنی رئوسها و $l : E(G) \rightarrow R^+$ تابع طول یال‌ها می‌باشد.

یک میانه از G ، رأسی مانند m است که

$$\sum_{v \in V(G)} w(v) d_G(m, v) = \min_{u \in V(G)} \sum_{v \in V(G)} w(v) d_G(u, v).$$

حال ما برای ساده‌تر شدن نمایش $\sum_{v \in V} w(v) d_G(u, v)$ از نماد $D(V, u)$ با $V \subseteq V(G)$ استفاده می‌کنیم. در ادامه، فاصله بین دو رأس u و v همیشه همان فاصله در درخت ورودی T است. به منظور ساده نویسی، برای فاصله بین u و v به جای $d_T(u, v)$ از $d(u, v)$ استفاده می‌شود.

برای هر زوج رأس (u, v) قرار دهید $\Pi(u, v) = (U, V)$ که $U = \{x : d(x, u) \leq d(x, v)\}$ و $V = V(T) - U$. توجه کنید که در یک درخت، هریک از U و V یک زیرگراف همبند را تولید می‌کنند که این دو مجموعه را با حذف یال $e \in E(T)$ می‌توان به دست آورد.

فرض کنید ρ_1 و ρ_2 احتمال‌های شکست دو سرویس‌دهنده در مساله پشتیبان ۲-میانه باشد. جواب مساله با یک زوج رأس (u_1, u_2) نشان داده می‌شود که u_1 مکان سرویس‌دهنده با احتمال شکست ρ_1 و u_2 مکان سرویس‌دهنده با احتمال شکست ρ_2 می‌باشد.

هدف مساله پشتیبان ۲-میانه مینیمم کردن مجموع فاصله از تمام رئوسها به مجموعه سرویس‌دهنده‌های

در حال کار است که به صورت زیر نوشته می شود،

$$C_\rho(v_1, v_2) = (1 - \rho) \left(\sum_v w(v) d(v, \{v_1, v_2\}) \right) \\ + \rho \left(\sum_v w(v) d(v, v_2) + \sum_v w(v) d(v, v_1) \right).$$

پس از ساده سازی با استفاده از نماد $D(V, u)$ که $V \subseteq V(G)$ و $u \in V(G)$ ، تابع هدف مساله به صورت زیر تعریف می شود.

تعریف ۱.۲.۲. (پشتیبان ۲- میانه روی درخت). درخت T و احتمال شکست ρ داده شده است. مساله پشتیبان ۲- میانه یک زوج از رتوس (m_1, m_2) را جستجو می کند به گونه ای که تابع هدف زیر را مینیمم کند:

$$C_\rho(v_1, v_2) = (1 - \rho) (D(V_1, v_1) + D(V_2, v_2)) + \rho (D(V_1, v_1) + D(V_2, v_1)) \\ + \rho (D(V_1, v_2) + D(V_2, v_2)) \\ = D(V_1, v_1) + \rho D(V_2, v_1) + D(V_2, v_2) + \rho D(V_1, v_2) \quad (1.2)$$

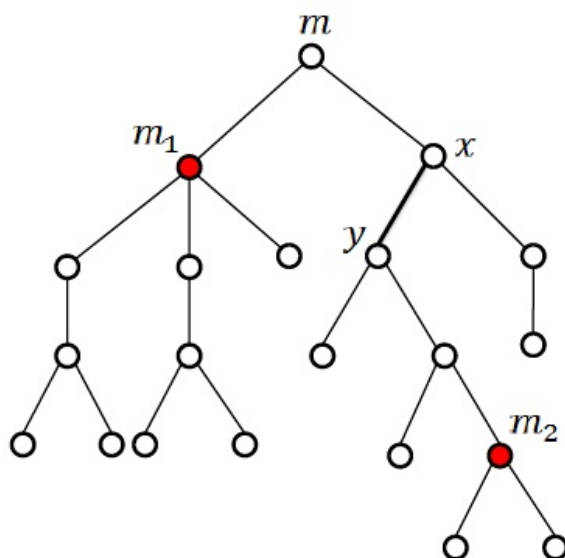
که در آن $v_1, v_2 \in V(T)$ و $(V_1, V_2) = \Pi(v_1, v_2)$ می باشد.

۳.۲ روش حل مساله

در این قسمت ما مساله پشتیبان ۲- میانه را روی درخت های با رأس های وزن دار بررسی می کنیم. درخت ورودی T را در میانه خود یعنی m ریشه دار می کنیم. روش ما مشابه روش حذف یال در مساله ۲- میانه کلاسیک روی درخت است که در فصل (۱) به صورت مختصر توضیح داده شد. فرض کنید s_1, s_2 دو میانه T باشند. قرار دهید $(S_1, S_2) = \Pi(s_1, s_2)$.

در مراجع [۲۶، ۳۶] نشان داده شده است که s_1 و s_2 باید به ترتیب میانه های زیردرخت های ایجاد شده توسط S_1 و S_2 باشند. در لم ۱.۳.۲ یک نتیجه مشابه را برای پشتیبان ۲- میانه بیان شده است. برای یک یال $e = (x, y) \in E$ که x پدر y است، فرض کنید $X(e)$ و $Y(e)$ دو زیر درخت به دست آمده از حذف یال e در درخت T باشند که x در $X(e)$ است و وزن های x و y به ترتیب برابر با $w(x) + \rho w(T_y)$ و $w(y) + \rho w(T - T_y)$ باشد. وزن تمام رأس های دیگر و طول تمام یال های باقی مانده را تغییر نمی دهیم. زیر درخت ریشه دار شده در u از $X(e)$ را با $X_u(e)$ نمادگذاری می کنیم. به طور مشابه $Y_v(e)$ زیر درختی از $Y(e)$ است که در v ریشه دار شده است.

لم ۱.۳.۲. [۴۷] اگر (m_1, m_2) پشتیبان ۲- میانه روی رأس های مختلف درخت T باشد، آنگاه m_1 و m_2 به ترتیب میانه های $X(e)$ و $Y(e)$ هستند که $e = (x, y)$ یک یال مرکزی m_1 و m_2 است. (شکل (۱.۲))



شکل ۱.۲: درخت T با میانه m و پشتیبان ۲-میانه (m_1, m_2) که با حذف یال $e = (x, y)$ به دست آمده است.

برهان. مجموعه رأس‌های $X(e)$ و $Y(e)$ را به ترتیب V_1 و V_2 در نظر بگیرید. با استفاده از تعریف، $d(v, m_1) \leq d(v, m_2)$ برای هر $v \in V_1$ و $d(v, m_2) \leq d(v, m_1)$ برای هر $v \in V_2$. در ادامه ثابت می‌کنیم m_1 میانه $X(e)$ است. اثبات برای m_2 مشابه است بنابراین آن را حذف کردیم. به برهان خلف فرض می‌کنیم m_1 میانه $X(e)$ نیست. فرض کنید s میانه $X(e)$ باشد. با استفاده از تعریف میانه داریم:

$$\sum_{v \in V_1} w(v)d(v, s) < \sum_{v \in V_1} w(v)d(v, m_1)$$

اما

$$\begin{aligned} \sum_{v \in V_1} w(v)d(v, s) &= \sum_{v \in V_1, v \neq x} w(v)d(v, s) + (w(x) + \rho w(V_2))d(x, s) \\ &= \sum_{v \in V_1} w(v)d(v, s) + \rho w(V_2)d(x, s) \end{aligned}$$

در نتیجه

$$D(V_1, s) + \rho w(V_2)d(x, s) < D(V_1, m_1) + \rho w(V_2)d(x, m_1) \quad (2.2)$$

همچنین داریم:

$$\begin{aligned} D(V_T, s) &= \sum_{v \in V_T} w(v) d(v, s) \\ &= \sum_{v \in V_T} w(v) (d(v, x) + d(x, s)) \\ &= \sum_{v \in V_T} w(v) d(v, x) + d(x, s) \sum_{v \in V_T} w(v) \\ &= D(V_T, x) + w(V_T) d(x, s) \end{aligned}$$

بنابراین می توان نوشت:

$$\begin{aligned} D(V_1, s) + \rho D(V_T, s) &= D(V_1, s) + \rho w(V_T) d(x, s) + \rho D(V_T, x) \\ &< D(V_1, m_1) + \rho w(V_T) d(x, m_1) + \rho D(V_T, x) \\ &= D(V_1, m_1) + \rho D(V_T, m_1) \end{aligned}$$

اما

$$\begin{aligned} c_\rho(m_1, m_T) &= D(V_1, m_1) + \rho D(V_T, m_1) + D(V_T, m_T) + \rho D(V_1, m_T) \\ &> D(V_1, s) + \rho D(V_T, s) + D(V_T, m_T) + \rho D(V_1, m_T) \geq c_\rho(s, m_T) \end{aligned}$$

□

که تناقض است

تعریف ۲.۳.۲. فرض کنید $e = (x, y)$ یالی در $E(T)$ و x پدر y باشد. با حذف یال e دو زیردرخت حاصل می شود. در ادامه میانه زیردرخت شامل رأس x را میانه بالا و میانه زیردرخت شامل y را میانه پایین یال e می نامیم و به ترتیب با نمادهای m_U و m_L نشان می دهیم.

الگوریتمی که در ادامه برای حل مساله ارائه می شود بر پایه لم ۱.۳.۲ قرار دارد، در نتیجه باید به ازای هر یال e یک زوج میانه بیابیم. در ادامه یک قضیه ساده از میانه در درخت بیان می کنیم.

قضیه ۳.۳.۲. [۴۷] رأس v میانه T است اگر و تنها اگر $w(T_v) \geq w(V)/2$ و $w(T_u) \leq w(V)/2$ برای هر فرزند u از V .

حال فرض کنید V_L و V_U به ترتیب مجموعه رأس های $X(e)$ و $Y(e)$ باشند. ما تابع هدف را به صورت تابع هزینه وابسته به یال بازنویسی می کنیم:

$$c'_\rho(e) = D(V_L, m_L) + \rho D(V_U, m_L) + D(V_U, m_U) + \rho D(V_L, m_U) \quad (۳.۲)$$

که (m_L, m_U) زوج میانه‌های یال e هستند و $(V_L, V_U) = \prod(m_L, m_U)$. الگوریتم یافتن پشتیبان ۲-میانه [۴۷]، $BACKUP\text{MEDIAN}$ نامیده می‌شود که به شکل زیر عمل می‌کند.

به جای محاسبه میانه‌های بالا و پایین یک یال به صورت همزمان، ابتدا میانه‌ی پایین تمام یال‌ها را (خط ۳) و سپس میانه بالای تمام یال‌ها را (خط ۴) محاسبه می‌شود.

یال‌های E را با i اندیس‌گذاری نموده و هریک از آن‌ها با e_i که $1 \leq i \leq n$ نمایش می‌دهیم. پس از اجرای زیربرنامه‌های $LOWER - MEDIAN$ و $UPPER - MEDIAN$ ، زوج میانه‌های هر یال e_i در $M_L[i]$ و $M_U[i]$ ذخیره می‌شود. بنابراین می‌توان هزینه وابسته به یال را به ازای هر زوج از میانه‌ها محاسبه و مینیمم‌ترین هزینه وابسته به یال را پیدا کرد. (خط ۶ تا ۱۱).

توجه کنید که دو سرویس‌دهنده ممکن است در یک رأس قرار بگیرند، بنابراین نیاز است که این مورد را نیز بررسی شود. بدیهی است که اگر دو سرویس‌دهنده در یک رأس قرار گیرند، m یعنی میانه درخت T ، بهترین انتخاب است. بنابراین $c_\rho(m, m)$ با هزینه وابسته به یال بهترین زوج از میانه‌ها مقایسه می‌شود و پشتیبان ۲-میانه مینیمم‌ترین آن‌هاست. (خط ۱۲ تا ۱۳).

شبه کد الگوریتم $BACKUP\text{MEDIAN}(T, \rho)$

ورودی: درخت T و احتمال شکست ρ برای هر سرویس‌دهنده.

خروجی: پشتیبان ۲-میانه درخت ورودی.

(۱) میانه درخت T را در متغیر m قرار بده.

(۲) درخت T که در m ریشه‌دار شده‌است را در T^* قرار بده.

(۳) زیربرنامه $LOWER - MEDIAN(T^*)$ را فراخوانی کرده و پس از اجرای آن تمام میانه‌های پایین یال‌ها را در آرایه M_L قرار بده.

(۴) زیربرنامه $UPPER - MEDIAN(T^*)$ را فراخوانی کرده و پس از اجرای آن تمام میانه‌های بالای یال‌ها را در آرایه M_U قرار بده.

(۵) متغیر opt را برابر با مقدار ∞ قرار بده.

(۶) $i = 1$

(۷) اگر $C'_\rho(e_i) < opt$ آن گاه به گام (۸) برو در غیراین صورت به گام (۱۰) برو.

(۸) مقدار $C'_\rho(e_i)$ را در متغیر opt قرار بده.

(۹) مقدار متغیر i را در i^* ذخیره کن.

$$i = i + 1 \quad (10)$$

(۱۱) اگر $i < n$ به گام (۷) برو در غیراین صورت به گام (۱۲) برو .

(۱۲) اگر $C'_\rho(e_{i^*}) > C_\rho(m, m)$ آن گاه گام (۱۳) را اجرا کن در غیراین صورت به گام (۱۴) برو .

(۱۳) توقف کنید، (m, m) پشتیبان ۲- میانه درخت ورودی T است.

(۱۴) توقف کنید، $(M_L[i^*], M_U[i^*])$ پشتیبان ۲- میانه درخت ورودی T است.

جزئیات محاسبه تابع هزینه وابسته به یال، میانه پایین و میانه بالا در ادامه توضیح داده شده است.

لم ۴.۳.۲. [۴۷] هزینه وابسته به یال یعنی c'_ρ برای تمام زوج میانه ها را می توان در زمان خطی محاسبه نمود.

برهان. برای یک رأس $u \in V(T_v)$ ، مجموع فاصله های $D(V(T_v), u)$ که

$$V(T_v) = V - (V - V(T_v))$$

را می توان با استفاده از فرمول زیر محاسبه کرد،

$$D(V(T_v), u) = D(V, u) - D(V - V(T_v), u)$$

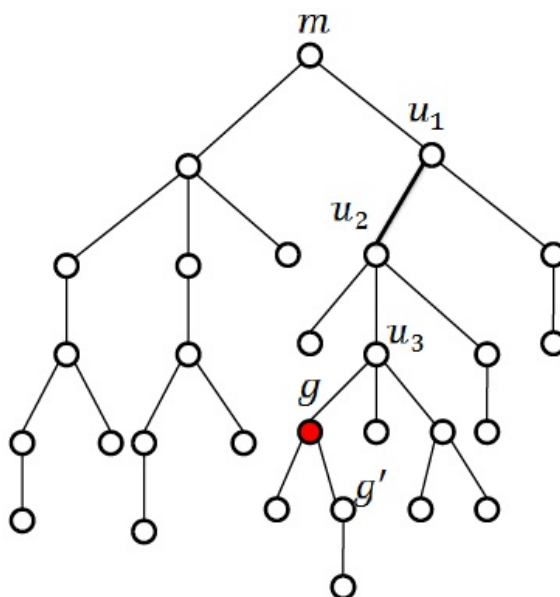
از آنجایی که $D(V - V(T_v), u)$ به صورت زیر تعریف می شود،

$$\begin{aligned} D(V - V(T_v), u) &= \sum_{x \in V - V(T_v)} w(x) d(x, u) \\ &= \sum_{x \in V - V(T_v)} w(x) (d(x, v) + d(v, u)) \\ &= \sum_{x \in V - V(T_v)} w(x) d(x, v) + d(v, u) \sum_{x \in V - V(T_v)} w(x) \\ &= D(V - V(T_v), v) + w(V - V(T_v)) d(v, u) \end{aligned}$$

می توان نوشت

$$\begin{aligned} D(V(T_v), u) &= D(V, u) - \left(D(V - V(T_v), v) + w(V - V(T_v)) \right) d(v, u) \\ &= D(V, u) - \left(D(V, v) - D(V(T_v), v) + (w(V) - w(T_v)) \right) (d(u, m) - d(v, m)). \end{aligned}$$

مرحله پیش پردازش مقادیر $d(v, m)$ ، $D(V(T_v), v)$ ، $D(V, v)$ و $w(T_v)$ را محاسبه می کند. پیش پردازش برای تمام رأس های V تنها به زمان خطی نیاز دارد، بنابراین $D(V(T_v), u)$ می تواند در زمان ثابت برای هر فرزند u از V محاسبه شود.

شکل ۲.۲: درخت T و میانه زیردرخت $Y(e)$

برای هر یال $e = (u_1, u_2)$ که u_1 پدر باشد، قرار دهید $V_U = V - V_L$ مقدار تابع هدف برابر است با،

$$\begin{aligned} c'_\rho(e) &= D(V_L, m_L) + \rho D(V_U, m_L) + D(V_U, m_U) + \rho D(V_L, m_U) \\ &= D(V_L, m_L) + \rho(D(V, m_L) - D(V_L, m_L)) + D(V, m_U) - (\lambda - \rho)D(V_L, m_U) \end{aligned}$$

چون

$$\begin{aligned} D(V_L, m_U) &= \sum_{x \in V_L} w(x) d(x, m_U) \\ &= \sum_{x \in V_L} w(x) (d(x, u_2) + d(u_2, m) + d(m, m_U)) \\ &= \sum_{x \in V_L} w(x) d(x, u_2) + (d(u_2, m) + d(m, m_U)) \sum_{x \in V_L} w(x) \\ &= D(V_L, u_2) + w(V_L) (d(u_2, m) + d(m, m_U)) \end{aligned}$$

بنابراین

$$\begin{aligned} c'_\rho(e) &= D(V_L, m_L) + \rho(D(V, m_L) - D(V_L, m_L)) + D(V, m_U) \\ &\quad - (\lambda - \rho)D(V_L, u_2) + w(V_L) (d(u_2, m) + d(m, m_U)) \end{aligned}$$

که می‌تواند در زمان ثابت محاسبه شود. زمان پیش‌پردازش $o(n)$ است. از آنجایی که $n - 1$ یال داریم، پس از پیش‌پردازش می‌توان فرمول بالا را برای محاسبه هزینه وابسته به یال برای تمام زوج میانه‌ها در زمان $o(n)$ استفاده کرد. بنابراین کل زمان مورد نیاز $o(n)$ است. $\square$

الگوریتم $LOWER - MEDIAN$ که در ادامه بیان شده، میانه‌ی پایین تمام یال‌ها را در زمان خطی محاسبه می‌کند. صحت و درستی این ادعا به صورت مستقیم از لم ۵.۳.۲ نتیجه می‌شود.

لم ۵.۳.۲. [۴۷] برای هر یال $e = (u_1, u_2)$ که u_1 پدر u_2 است، میانه g از $Y(e)$ در مسیر $P[g', u_2]$ قرار دارد که g' میانه $Y(e')$ به‌ازای یال $e' = (u_2, u_3)$ است و u_3 فرزند u_2 با

$$w(T_{u_3}) = \max_{v \in C(u_2)} w(T_v)$$

است. (شکل (۲.۲))

برهان. چون g میانه $Y(e)$ است بنابر قضیه ۳.۳.۲ داریم: $w(Y_g(e)) \geq w(Y(e))/2$ ، از آنجایی که در بین فرزندان u_2 ، T_{u_3} زیردرختی با بیشترین وزن است پس g باید در $T_{u_3} \cup u_2$ مستقر شود. با استفاده از قضیه ۳.۳.۲ کافی است نشان دهیم که برای هر $x \notin P[g', u_2]$ ،

$$w(Y_x(e)) < w(Y(e))/2.$$

برای هر فرزند x از g' ، که $x \neq u_3$ و $x \neq u_2$ ، چون g' میانه $Y(e')$ است، می‌توان نتیجه گرفت که $w(Y_x(e)) = w(Y_x(e')) < w(Y(e'))/2$ با استفاده از تعریف داریم،

$$w(Y(e')) = w(T_{u_3}) + \rho w(T - T_{u_3})$$

$$< w(T_{u_3}) + \rho w(T - T_{u_3})$$

$$= w(Y(e))$$

بنابراین $w(Y_x(e)) < w(Y(e))/2$.

برای رأس‌هایی که در مسیر $P[g', u_2]$ نیستند و همچنین رأس‌هایی که فرزند g' نیستند، پارامتر مشابه برقرار است و لذا لم نیز برقرار می‌باشد. $\square$

شبه کد زیربرنامه $LOWER - MEDIAN(T^*)$

ورودی: درخت T^* که در میانه خود ریشه‌دار شده‌است.

خروجی: آرایه L که حاوی میانه‌های پایین به ازای تمام یال‌های T^* است.

(۱) برای هر رأس v از درخت T^* مقدار $w(T_v)$ که همان زیردرخت ریشه‌دار شده در رأس v است، را محاسبه کن. وزن $w(T_v)$ برابر $w(v) + \sum_{u \in C(v)} w(u)$ می‌باشد.

(۲) مجموعه رأس‌ها را ترتیب پیمایش پس ترتیب که در فصل (۱) توضیح دادیم، مرتب نموده و در $(v_1, v_2, \dots, v_n)$ قرار دهید.

$$i = 1 \quad (۳)$$

(۴) یال بین v_i و پدرش را در متغیر e_i قرار دهید.

(۵) اگر v_i یک برگ است به گام (۶)، در غیراین صورت به گام (۷) برو.

(۶) رأس v_i را در آرایه $L[i]$ قرار بده.

(۷) مقدار $w(T_{v_i}) + \rho w(T - T_{v_i})$ را در متغیر λ قرار بده.

(۸) فرزندی از v_i مانند u را که $w(T_u) = \max_{x \in C(v_i)} w(T_x)$ در متغیر u قرار بده.

(۹) میانه زیردرخت $Y((v_i, u))$ را در متغیر g قرار بده.

(۱۰) اگر $w(T_g) < \lambda/2$ و $g \neq v_i$ ، به گام (۱۱)، در غیراین صورت به گام (۱۲) برو.

(۱۱) پدر رأس g را جایگزین g کرده و به گام (۱۰) برو.

(۱۲) متغیر g را در $L[i]$ ذخیره کن.

(۱۳) $i = i + 1$

(۱۴) اگر $i < n$ به گام (۴) برو در غیراین صورت به گام (۱۵) برو.

(۱۵) آرایه L شامل میانه‌های پایین تمام یال‌ها می‌باشد.

در زیربرنامه $LOWER - MEDIAN$ ، همیشه میانه پایین را در سنگین‌ترین زیردرخت نزدیک به ریشه جستجو می‌کنیم و با پیدا کردن آن با استفاده از قضیه ۳.۳.۲ جستجو پایان می‌یابد. در این الگوریتم هیچ‌گونه محاسبات تکراری نداریم و می‌توانیم لم زیر را نتیجه بگیریم.

لم ۶.۳.۲ [۴۷] فرض کنید T یک درخت ریشه‌دار در میانه خود یعنی m باشد. میانه‌های $Y(e)$ برای تمام یال‌های e می‌تواند در زمان $o(n)$ محاسبه شود.

الگوریتم $UPPER - MEDIAN$ ، میانه بالای تمام یال‌ها را در زمان $o(n \log n)$ محاسبه می‌کند. صحت و درستی آن در ادامه به صورت مستقیم از لم ۷.۳.۲ نتیجه می‌شود.

لم ۷.۳.۲ فرض کنید T یک درخت ریشه‌دار در میانه خود یعنی m باشد و فرض کنید T_1 و T_2 سنگین‌ترین و دومین سنگین‌ترین زیردرخت‌های ریشه‌دار در فرزندان m باشند. برای هر یال e ، میانه $X(e)$ در مسیر بین m و میانه T_i است که $i = 2$ اگر e در T_1 باشد و در غیراین صورت $i = 1$.

برهان. فرض کنید که یال حذف شده در T_1 باشد. ما نیاز داریم یک رأس v بیابیم که در قضیه ۳.۳.۲ صدق کند یعنی به ازای تمام فرزندان u از V ، $w(X_v(e)) \geq \frac{w(X(e))}{4}$ و $w(X_u(e)) < \frac{w(X(e))}{4}$. چون m میانه درخت T است، می‌توان نتیجه گرفت که هیچ شاخه یا انشعابی از m با وزن بیشتر از $\frac{w(T)}{4}$ وجود ندارد. بنابراین $\frac{w(T)}{4} \geq w(T - T_1) \geq \frac{w(X(e))}{4}$ و بدیهی است که میانه $X(e)$ باید در $T - T_1$ باشد. از

آنجایی که تنها رأس هایی که ممکن است وزن شان از $\frac{w(X(e))}{4}$ بزرگتر باشد، رأس های بین m و میانه T_2 هستند، بنابراین کافی است که تنها این رأس ها را بررسی کنیم.

در موردی که حذف یال در زیردرخت دیگری از T_1 رخ دهد، اثبات مشابه است لذا آن را ذکر نکردیم.

□

با استفاده از لم ۷.۳.۲، ما می توانیم میانه $X(e)$ را با استفاده از جستجوی دودویی روی مسیر بین میانه T و میانه شاخه ای با بیشترین وزن از زیردرخت به دست آمده بیابیم.

شبه کد زیر برنامه $UPPER - MEDIAN(T^*)$

ورودی: درخت T^* که در میانه خود ریشه دار شده است.

خروجی: آرایه U که حاوی میانه های بالا به ازای تمام یال های T^* است.

(۱) برای هر رأس v از درخت T^* مقدار $w(T_v)$ را محاسبه کن.

(۲) سنگین ترین شاخه از m را در T_{v_1} قرار بده.

(۳) دومین سنگین ترین شاخه از m را در T_{v_2} قرار بده.

(۴) میانه زیردرخت T_{v_1} را در m_{v_1} قرار بده.

(۵) میانه زیردرخت T_{v_2} را در m_{v_2} ذخیره کن.

(۶) مسیر از ریشه به m_{v_1} را در $Path_1$ ذخیره کن.

(۷) مسیر از ریشه به m_{v_2} را نیز در $Path_2$ ذخیره کن.

(۸) یال ها را به دلخواه مرتب نموده و در $(e_1, e_2, \dots, e_{n-1})$ قرار بده.

(۹) $i = 1$

(۱۰) اگر $e_i \in T_{v_1}$ آن گاه به گام (۱۱) برو در غیر این صورت به گام (۱۲) برو.

(۱۱) پایین ترین رأس v در مسیر $Path_2$ با $w(T_v) \geq \frac{w(X(e_i))}{4}$ را در $U[i]$ ذخیره کن.

(۱۲) پایین ترین رأس v در مسیر $Path_1$ با $w(T_v) \geq \frac{w(X(e_i))}{4}$ را در $U[i]$ ذخیره کن.

(۱۳) $i = i + 1$

(۱۴) اگر $i < n$ به گام (۱۰) برو در غیر این صورت به گام (۱۵) برو.

(۱۵) آرایه U را برگردان.

لم ۸.۳.۲. [۴۷] فرض کنید T یک درخت ریشه‌دار در میانه خود، m باشد. میانه‌های $X(e)$ برای تمام یال‌های e را می‌توان در زمان $o(n \log n)$ محاسبه نمود.

حال ما درستی و همچنین پیچیدگی الگوریتم بیان شده را نشان می‌دهیم. با توجه به لم‌های ۵.۳.۲ و ۷.۳.۲ ما می‌دانیم که زوج میانه‌های تمام یال‌ها به درستی محاسبه شده‌اند. بنابراین اثبات این که پشتیبان ۲-میانه یا یک میانه از درخت و یا یک زوج میانه با مینیمم‌ترین هزینه وابسته به یال است، باقی‌مانده است.

لم ۹.۳.۲. [۴۷] برای هر زوج از میانه‌های (m_L, m_U) از e ، $C'_\rho(e) \geq C_\rho(m_L, m_U)$ برقرار است. علاوه‌براین، مینیمم مقدار دو تابع روی تمام زوج میانه‌ها یکسان است.

برهان. فرض کنید $e = (u_1, u_2)$ که پدر است و $(V_1, V_2) = \Pi(m_L, m_U)$. برای چهار زیرمجموعه از رأس‌های $(V(T - T_{u_2}), V(T_{u_2}), V_1, V_2)$ تنها دو امکان وجود دارد: یا $V_1 \subseteq V(T_{u_2})$ یا $V_2 \subseteq V(T - T_{u_2})$. دو مورد مشابه هستند، بنابراین تنها حالت $V_1 \subseteq V(T_{u_2})$ را بررسی می‌کنیم.

از روابط زیر می‌توان نتیجه گرفت که $C'_\rho(m_L, m_U) \leq C'_\rho(e)$ و

$$\begin{aligned} C'_\rho(m_L, m_U) &= D(V_1, m_L) + \rho D(V_2, m_L) + D(V_2, m_U) + \rho D(V_1, m_U) \\ &= D(V_1, m_L) + D(V(T_{u_2}) - V_1, m_U) + D(V_2 - V(T_{u_2}), m_U) \\ &\quad + \rho(D(V_1, m_U) + \rho(D(V(T_{u_2}) - V_1, m_L) + D(V_2 - V(T_{u_2}), m_L))) \\ &\leq D(V_1, m_L) + D(V(T_{u_2}) - V_1, m_L) + D(V_2 - V(T_{u_2}), m_U) \\ &\quad + \rho(D(V_1, m_U) + \rho(D(V(T_{u_2}) - V_1, m_U) + D(V_2 - V(T_{u_2}), m_L))) \\ &\leq D(V(T_{u_2}), m_L) + D(V - V(T_{u_2}), m_U) + \rho(D(V(T_{u_2}), m_U) \\ &\quad + \rho(D(V - V(T_{u_2}), m_L)) \\ &= C'_\rho(e) \end{aligned}$$

بنابر لم ۱.۳.۲، می‌دانیم که پشتیبان ۲-میانه روی رأس‌های مختلف S' باید زوجی از میانه‌های یال e^* باشد. بنابراین $C_\rho(s') = C'_\rho(e^*)$ مینیمم مقدار دو تابع است. $\square$

حال پیچیدگی زمانی الگوریتم $BACKUP_2MEDIAN$ را بررسی می‌کنیم. طبق قضیه ۳.۳.۲ گام (۱) زمان $o(n)$ نیاز دارد. گام (۲)، درخت را به فرم ریشه‌دار در زمان $o(n)$ بازچینی می‌کند. گام (۳) نیز بنابر لم ۶.۳.۲، زمان $o(n)$ نیاز دارد و گام (۴) همان‌گونه که در لم ۸.۳.۲ اثبات شد به زمان $o(n \log n)$ نیاز دارد. گام (۱۲)، $C_\rho(m, m)$ را محاسبه و یک مقایسه در زمان ثابت انجام می‌دهد. بر پایه تجزیه و تحلیل فوق، قضیه زیر را داریم.

قضیه ۱۰.۳.۲. [۴۷] الگوریتم $BACKUP_2MEDIAN$ ، پشتیبان ۲- میانه درخت T را در زمان $o(n \log n)$ محاسبه می کند.

برای حالتی که $\rho_1 \neq \rho_2$ مراحل زیر می توان انجام داد.
ابتدا احتمال شکست ρ با ρ_1 و ρ_2 جایگزین می شود و تابع هدف را به فرم زیر تعریف می شود:

$$C_{\rho_1, \rho_2}((v_1, v_2)) = (1 - \rho_1)(D(V_1, v_1) + \rho_2(D(V_2, v_1))) \\ + (1 - \rho_2)((D(V_2, v_2) + \rho_1 D(V_1, v_2)))$$

که $(V_1, V_2) = \prod(v_1, v_2)$.

مشابه $X(e)$ و $Y(e)$ ما $X^i(e)$ و $Y^i(e)$ را برای $e = (x, y) \in E(T)$ که x پدر باشد، تعریف می کنیم.
درحالی که وزن های x و y برای $i, j \in 1, 2$ و $i \neq j$ برابر با $w(x) + \rho_i w(T_y)$ و $w(y) + \rho_i w(T - T_y)$ است. بنابراین درحالتی که $\rho_1 \neq \rho_2$ برای یک یال دو زوج از میانه ها وجود دارد.
همچنین هزینه وابسته به یال را به فرم زیر دوباره تعریف می کنیم:

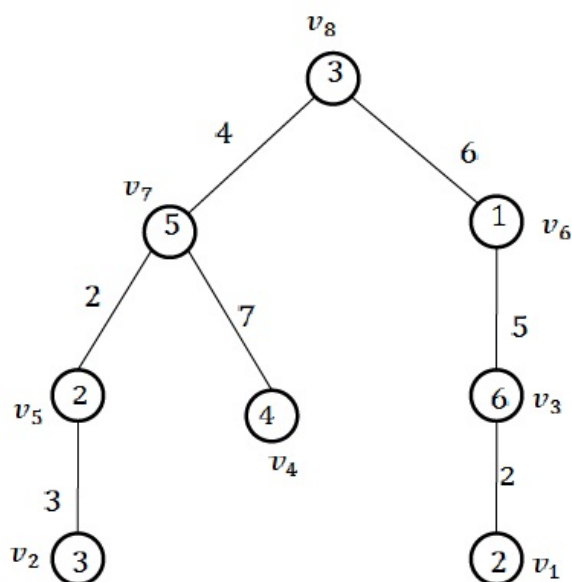
$$C'_{\rho_1, \rho_2}(e) = (1 - \rho_1)D(V(T_{u_1}), m_L) + (1 - \rho_2)D(V - V(T_{u_2}), m_U) \\ + \rho_1(1 - \rho_2)D(V(T_{u_2}), m_U) + \rho_2(1 - \rho_1)D(V - V(T_{u_1}), m_L)$$

که (m_L, m_U) یک زوج از میانه های یال $e = (u_1, u_2)$ است که u_1 پدر است. سرانجام ما کمترین هزینه وابسته به یال تمام میانه های بالا و پایین را با $C_{\rho_1, \rho_2}(m, m)$ مقایسه می کنیم و یک پشتیبان ۲- میانه را پیدا می کنیم.

برپایه تحلیل فوق قضیه زیر را داریم:

قضیه ۱۱.۳.۲. [۴۷] احتمال های شکست ρ_1 و ρ_2 داده شده است. یک پشتیبان ۲- میانه برای درخت با رتوس وزن دار می تواند در زمان $o(n \log n)$ محاسبه شود.

مثال ۱۲.۳.۲. درختی با ۸ رأس مشابه آنچه در شکل ۳.۲ نشان داده شده است، در نظر بگیرید. اگر $\rho = 0$ پشتیبان ۲- میانه (v_7, v_3) است، که جواب مساله ۲- میانه کلاسیک با وزن مثبت-منفی نیز است. اگر $\rho = 0.5$ پشتیبان ۲- میانه (v_8, v_2) است که از حذف یال (v_2, v_5) به دست آمده است. برای این مساله مقدار تابع هدف که از حذف بعضی یال ها به دست آمده، به ازای $\rho = 0$ و $\rho = 0.5$ به ترتیب در جدول های (۱.۲) و (۲.۲) آورده شده است.



شکل ۳.۲: درخت T با ۸ رأس و میانه v_7 .

یال حذف شده	m_U	m_L	$W(X(e))$	$W(Y(e))$	$F(m_U, m_L)$
(v_1, v_3)	v_7	v_1	۲۴	۲	۱۵۹
(v_2, v_5)	v_8	v_2	۲۳	۳	۱۷۴
(v_3, v_6)	v_7	v_3	۱۸	۸	۷۳
(v_4, v_7)	v_8	v_4	۲۲	۴	۱۵۷
(v_5, v_7)	v_8	v_5	۲۱	۵	۱۷۱
(v_6, v_8)	v_7	v_6	۱۷	۹	۱۰۳
(v_8, v_7)	v_7	v_8	۱۴	۱۲	۱۰۹

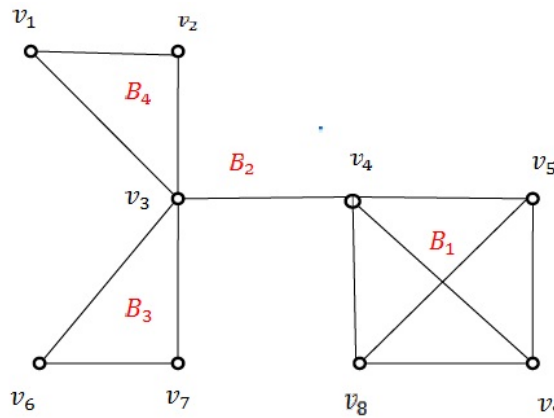
جدول ۱۰.۲: برای حالت $\rho = 0/0$ تابع هدف و میانه های زیر درخت های به دست آمده از حذف یال درخت شکل (۳.۲).

یال حذف شده	m_U	m_L	$W(X(e))$	$W(Y(e))$	$F(m_U, m_L)$
(v_1, v_3)	v_7	v_1	۲۵	۱۴	۳۷۴
(v_2, v_5)	v_8	v_2	۲۴.۵	۱۴.۵	۲۳۹.۵
(v_3, v_6)	v_7	v_3	۲۲	۱۷	۳۷۲.۷۵
(v_4, v_7)	v_8	v_4	۲۴	۱۵	۴۷۵
(v_5, v_7)	v_8	v_5	۲۳.۵	۱۷.۵	۲۷۵.۲۵
(v_6, v_8)	v_7	v_6	۲۱.۵	۱۷.۵	۴۰۷.۷۵
(v_8, v_7)	v_7	v_8	۲۰	۱۹	۲۹۵.۵

جدول ۲۰.۲: برای حالت $\rho = 0/5$ تابع هدف و میانه های زیر درخت های به دست آمده از حذف یال درخت شکل (۳.۲).

فصل ۳

مساله پشتیان ۲- میانه روی گراف های بلوکی



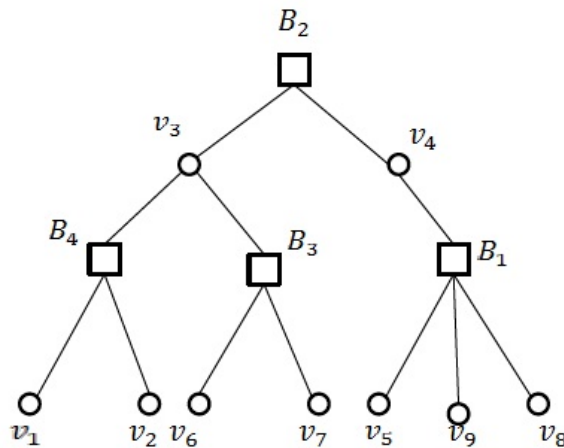
شکل ۱.۳: گراف بلوکی G با چهار بلوک $\{B_1, B_2, B_3, B_4\}$

۱.۳ مقدمه

در این فصل ما مساله پشتیبان ۲- میانه روی گرافهای بلوکی را بررسی می‌کنیم. مساله p - میانه روی شبکه‌های عمومی NP - سخت است، در نتیجه به‌منظور حل این مساله اغلب از ساختاری که الگوریتم‌هایی با کارایی بهتر دارند، استفاده می‌شود. بنابراین چنگ و کنگ [۱۹] برای حل مساله پشتیبان ۲- میانه روی گرافهای بلوکی از ساختاری که چن و همکارانش [۱۷] برای تبدیل گراف بلوکی به درخت ارایه دادند، استفاده کرده و با ساخت یک پیکربندی وزن‌دار از گراف بلوکی، الگوریتمی با پیچیدگی زمانی $O(n \log n + m)$ برای این مساله پیشنهاد دادند. n و m به‌ترتیب تعداد رئوس و تعداد یال‌های گراف بلوکی مورد بررسی می‌باشند. در بخش اول نمادهای استفاده شده در این فصل و روش ساخت یک پیکربندی وزن‌دار از گراف بلوکی را بیان می‌کنیم. سپس مساله ۱- میانه روی گرافهای بلوکی را مطرح و با استفاده از نتایج بیان شده برای این مساله، مساله پشتیبان ۲- میانه روی گرافهای بلوکی را بررسی می‌کنیم. برای ساده‌تر شدن تعریف مساله از ابتدای فصل احتمال شکست دو سرویس‌دهنده یکسان در نظر گرفته می‌شود. در قسمت انتهایی فصل مساله را برای حالتی که $\rho_1 \neq \rho_2$ باشد، تعریف خواهیم کرد.

۲.۳ تعاریف اولیه

در این بخش برخی نمادها و اصطلاحات علمی این فصل معرفی می‌شود. در این فصل از نماد $G = (V(G), E(G))$ برای نشان دادن یک گراف بلوکی با $|V(G)| = n$ و $|E(G)| = m$ استفاده می‌کنیم. با این فرض که هر دو یال در یک بلوک طول یکسان دارند، $l_G(B_j)$ طول هر یال در بلوک B_j ، $j = 1, 2, \dots, l$ ، است. شکل (۱.۳) یک گراف بلوکی G با چهار بلوک را نشان می‌دهد که بلوک‌های آن B_1, B_2, B_3 و B_4 هستند. در الگوریتمی که چنگ و کنگ [۱۹] برای مساله پشتیبان ۲- میانه روی گراف بلوکی ارایه دادند، همانند



شکل ۲.۳: پیکربندی گراف بلوکی G

روش مطرح شده در مرجع [۱۷] از گراف بلوکی G یک پیکربندی وزن دار مانند $S = (V(S), E(S))$ می سازند که $V(S) = V(G) \cup B(G)$ و $E(S) = \{(v, B) \mid v \in V(B), v \in V(G), B \in B(G)\}$. برای تمایز قایل شدن بین رئوس S و G ، یک رأس در S را گره نامیده می شود. در نتیجه گره x یا یک رأس در $V(G)$ و یا یک بلوک در $B(G)$ است. اگر گره x رأسی در G باشد، وزن آن برابر با $w_S(x) = w_G(x)$ است، در غیر این صورت $w_S(x) = 0$. بنابراین می توان به راحتی تساوی $W(S) = W(G)$ را نتیجه گرفت. طول هر یال $e = (v, B)$ در $E(S)$ را برابر با $l_S(e) = \frac{1}{2}l_G(B)$ در نظر بگیرید. پیکربندی وزن دار 1 گراف شکل (۱.۳) در شکل (۲.۳) نشان داده شده است. توجه کنید که اگر گراف بلوکی G همبند باشد، پیکربندی S یک درخت خواهد بود. علاوه بر این اگر $|V(G)| > 1$ ، در نتیجه برگ های S رئوس G هستند. آهو 2 و همکارانش [۴] نشان دادند که پیکربندی گراف بلوکی G با روش جستجوی اولین مسیر می تواند در زمان خطی $O(n + m)$ ساخته شود. پیکربندی S به عنوان یک درخت ریشه دار با ریشه r در نظر گرفته می شود.

تعریف ۱.۲.۳. برای هر دو رأس u و v ، اگر u روی مسیر بحرانی $P_S[v, r]$ باشد و $level(v) = level(u) + 2$ آنگاه ما u را پدر سوئی رأس v می نامیم و آن را با نماد $DVP(v)$ نمایش می دهیم.

برای مثال در شکل (۲.۳)، بلوک B_2 ریشه S است. همچنین با توجه به شکل داریم: $v_3 = par(B_3)$ ، $C(B_4) = \{v_1, v_2\}$ ؛ $par(B_4) = \{v_3, v_4\}$ و $v_4 = DVP(v_5) = DVP(v_8) = DVP(v_9)$. اگر یال $(x, y) \in E(S)$ با $x = par(y)$ را حذف کنیم، S به دو زیردرخت همبند تجزیه می شود که زیردرخت شامل گره y را با S_y و زیردرخت دیگر را با S_y^c نشان می دهیم. مجموعه های $V^G(S_y)$ و $V^G(S_y^c)$ ، مجموعه رئوس زیردرخت های S_y و S_y^c هستند. با توجه به این ساختار S ، لم زیر را خواهیم داشت.

لم ۲.۲.۳. [۱۹] برای هر دو رأس u و v در $V(G)$ ، $d_G(u, v) = d_S(u, v)$.

¹Weighted Skelton²Aho

برهان. با توجه به ساختار S بدیهی است که برای هر دو گره x و y در $V(S)$ روی مسیر $PS[x, y]$ رئوس و بلوک ها به صورت تناوبی قرار دارند. بنابراین برای دو رأس u و v ، فرض می کنیم که $PS[u, v] = (u = v_i, B_j, v_{i+1}, B_{j+1}, \dots, B_g, v_h = v)$ پس مسیر $PG[u, v]$ در G را می توان به صورت $PG[u, v] = (u = v_i, v_{i+1}, \dots, v_h = v)$ نوشت. حال چون هر دو یال در یک بلوک دارای طول یکسان هستند، همچنین برای هر یال $e = (v, B)$ در $E(S)$ ، داریم: $l_S(v, B) = \frac{1}{p} l_G(B)$ در نتیجه لم برقرار است. $\square$

۳.۳ مساله ۱- میانه روی گراف های بلوکی

به منظور حل مساله پشتیبان ۲- میانه روی گراف بلوکی G ، در این قسمت ابتدا مساله ۱- میانه روی همان گراف G را در نظر می گیریم [۱۹].

مساله ۱- میانه روی گراف N ، به دنبال یافتن یک رأس m^* است به گونه ای که مجموع کلی وزن فاصله تمام رئوس تا m^* ، مینیمم باشد. برای یک زیرگراف N' از N و یک رأس v بر N تعریف می کنیم:

$$F(N', v) = \sum_{u \in V(N')} w_{N'}(u) d_{N'}(u, v) \quad (۱.۳)$$

بنابراین میانه m^* در $F(N, m^*) = \min_{u \in V(N)} F(N, u)$ صدق می کند.

لم ۱.۳.۳. [۱۹] برای هر رأس $v \in V(G)$ ، $F(S, v) = F(G, v)$.

برهان. با استفاده از تعریف فوق داریم:

$$F(S, v) = \sum_{u \in V(S)} w_S(u) d_S(u, v) = \sum_{u \in V(G)} w_G(u) d_G(u, v) = F(G, v)$$

معادله دوم به صورت مستقیم از لم ۲.۲.۳ و این حقیقت که وزن های گره های بلوکی در S صفرند، پیروی می کند. $\square$

گراف بلوکی G داده شده است. وزن های پیکربندی S می تواند در زمان $O(n + m)$ ایجاد شود. همچنین مقادیر $F(S, x)$ و $W(S_x)$ برای تمام گره های $V(S)$ در زمان $O(n + l) = O(n)$ محاسبه می شود. با استفاده از لم ۱.۳.۳، می توان $F(G, v)$ را برای تمام رئوس در $V(G)$ حساب کرده و مینیمم را در زمان $O(n)$ بیابیم. بنابراین قضیه زیر را داریم:

قضیه ۲.۳.۳. [۱۹] مساله ۱- میانه در گراف بلوکی می تواند در زمان $O(n + m)$ حل شود.

به صورت مستقیم از لم ۱.۳.۳ می توان نتیجه گرفت که اگر میانه پیکربندی S'_s یک رأس باشد، آنگاه رأس متناظر آن یک میانه در گراف بلوکی G است. در ادامه ارتباط مساله ۱- میانه با گراف بلوکی G و پیکربندی وزن دار آن S را با مثالی که میانه پیکربندی S ، بلوک B^* است، نشان می دهیم. در این جا ما پیکربندی S را در میانه B^* ریشه دار می کنیم. فرض کنید $C(B^*) = \{u_1, u_2, \dots, u_k\}$.

در ادامه با بیان یک لم نشان می دهیم که خاصیت اکثریت که برای مساله ۱- میانه روی درخت برقرار است، برای مساله ۱- میانه روی پیکربندی وزن دار به دست آمده از گراف بلوکی نیز برقرار می باشد.

لم ۳.۳.۳. [۱۹] گره x میانه پیکربندی S است اگر و تنها اگر $W(S_x) \geq W(S)/2$ و $W(S_y) \leq W(S)/2$ برای هر فرزند x مانند y .

بنابر لم ۳.۳.۳ می توان رابطه زیر را نوشت: $W(S_{u_i}) \leq W(S)/2 = W(G)/2$. حال به منظور تعیین میانه گراف بلوکی G ، بیان لم زیر کافی است.

لم ۴.۳.۳. [۱۹] برای هر فرزند u_i از B^* که $i = 1, 2, \dots, k$ ، میانه پیکربندی S است، داریم:

$$F(S, u_i) = F(S, B^*) + l_G(B^*)(1/2W(G) - W(S_{u_i}))$$

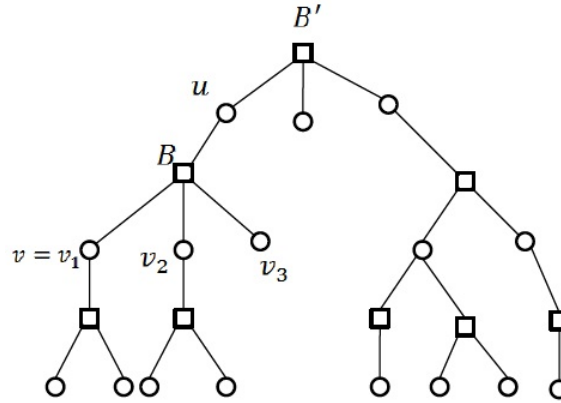
برهان. بدیهی است که مجموعه رأسی $V(G)$ می تواند به k مجموعه رأسی مجزا تجزیه شود به گونه ای که $V(G) = \bigcup_{i=1}^k V^G(S_{u_i})$. بنابراین می توان نتیجه گرفت $W(G) = \sum_{i=1}^k W(S_{u_i})$. برای هر رأس $P_S[v, B^*]$ ، $v \in V^G(S_{u_i})$ باید از بلوک B^* بگذرد و برای هر رأس $v \in V^G(S_{u_j})$ ، $j \neq i$ ، $P_S[v, B^*]$ از رأس u_i می گذرد. بنابراین می توان رابطه زیر را نوشت،

$$d_S(v, u_i) = \begin{cases} d_S(v, B^*) + 1/2 l_G(B^*) & v \in V^G(S_{u_j}), j \neq i \\ d_S(v, B^*) - 1/2 l_G(B^*) & v \in V^G(S_{u_i}) \end{cases} \quad (2.3)$$

با استفاده از رابطه ۲.۳ و این اصل که وزن هر گره بلوکی در S صفر است، می توان نتیجه گرفت که،

$$\begin{aligned} F(S, u_i) &= \sum_{j=1, j \neq i}^k \sum_{v \in V^G(S_{u_j})} w_S(v) d_S(v, u_i) + \sum_{v \in V^G(S_{u_i})} w_S(v) d_S(v, u_i) \\ &= \sum_{j=1, j \neq i}^k \sum_{v \in V^G(S_{u_j})} w_S(v) (d_S(v, B^*) + \frac{1}{2} l_G(B^*)) + \\ &\quad \sum_{v \in V^G(S_{u_i})} w_S(v) (d_S(v, B^*) - \frac{1}{2} l_G(B^*)) \\ &= \sum_{j=1}^k \sum_{v \in V^G(S_{u_j})} w_S(v) d_S(v, B^*) + \frac{1}{2} l_G(B^*) \sum_{j=1, j \neq i}^k \sum_{v \in V^G(S_{u_j})} w_S(v) \\ &\quad - \frac{1}{2} l_G(B^*) \sum_{v \in V^G(S_{u_i})} w_S(v) \pm \frac{1}{2} l_G(B^*) \sum_{v \in V^G(S_{u_i})} w_S(v) \\ &= F(S, B^*) + l_G(B^*)(\frac{1}{2} W(G) - W(S_{u_i})) \end{aligned}$$

□



شکل ۳.۳: پیکربندی وزن دار از گراف بلوکی G که $u = DVP(v)$

لم ۵.۳.۳. [۱۹] برای هر دو رأس u و v که $u = DVP(v)$ داریم:

$$F(S, v) = F(S, u) + l_G(B)(W(G) - W(S_v) - W(S_B))$$

که $B = par(v) \in C(u)$.

برهان. واضح است که بلوک B روی مسیر $P_S[u, v]$ قرار دارد و همچنین رئوس u و v هر دو در B هستند. فرض کنید $C(B) = \{v = v_1, v_2, \dots, v_t\}$ ، بنابراین مجموعه رئوس $V(G)$ می تواند به $t+1$ مجموعه مجزا تقسیم شوند به طوری که $V(G) = (\bigcup_{i=1}^t V^G(S_{v_i})) \cup V^G(S_B^c)$. برای هر رأس $x \in V^G(S_{v_i})$ ، $P_S[x, u]$ از v می گذرد. برای هر رأس $x \in V^G(S_B^c)$ ، $P_S[x, v]$ روی u قرار دارد و برای هر رأس $x \in V^G(S_{v_i})$ ، $i = 1, 2, \dots, t$ و $P_S[x, v]$ هر دو از بلوک B می گذرند. (شکل ۳.۳) را ببینید.) بنابراین

$$d_S(x, v) = \begin{cases} d_S(x, u) - l_G, & x \in V^G(S_v); \\ d_S(x, B) + \frac{1}{t} l_G(B) = d_S(x, u) & x \in V^G(S_{v_i}), i = 2, \dots, t; \\ d_S(x, u) + l_G, & x \in V^G(S_B^c). \end{cases} \quad (3.3)$$

با استفاده از رابطه ۳.۳ () می توان نتیجه گرفت که،

$$\begin{aligned} F(S, v) &= \sum_{i=1}^t \sum_{x \in V^G(S_{v_i})} w_S(x) d_S(x, v) + \sum_{x \in V^G(S_B^c)} w_S(x) d_S(x, v) \\ &= \sum_{i=2}^t \sum_{x \in V^G(S_{v_i})} w_S(x) d_S(x, u) + \sum_{x \in V^G(S_v)} w_S(x) (d_S(x, u) - l_G(B)) \\ &\quad + \sum_{x \in V^G(S_B^c)} w_S(x) (d_S(x, u) + l_G(B)) \\ &= F(S, u) - l_G(B)W(S_v) + l_G(B)W(S_B^c) \\ &= F(S, u) + l_G(B)(W(G) - W(S_v) - W(S_B)). \end{aligned}$$

□ که معادله آخر از رابطه $W(G) = W(S_B) + W(S_B^c)$ نتیجه گرفته می شود.

حال با توجه به مطالب بیان شده تا کنون، می توان قضیه زیر را که اساسی ترین قضیه در مساله ۱- میانه روی گراف بلوکی است، بیان کرد. با استفاده از قضیه زیر می توانیم میانه گراف بلوکی G را زمانی که میانه پیکربندی S یک بلوک است، تعیین کنیم.

قضیه ۶.۳.۳. [۱۹] اگر میانه پیکربندی S ، بلوک B^* باشد و B^* ریشه S باشد، آنگاه فرزند u^* از B^* که در رابطه $W(S_{u^*}) = \max_{1 \leq i \leq k} W(S_{u_i})$ صدق می کند، میانه گراف بلوکی G است که $C(B^*) = \{u_1, u_2, \dots, u_k\}$

برهان. با استفاده از لم (۴.۳.۳)، فرض کنید $F(S, u^*) = \min_{1 \leq i \leq k} F(S, u_i)$ باشد. چون $W(S_{u^*}) = \max_{1 \leq i \leq k} W(S_{u_i})$ پس نشان می دهیم که برای هر رأس $v \in V^G(S_{u_i})$ ، $F(S, v) \geq F(S, u_i)$

برای هر دو رأس u و v با $u = DVP(V)$ باید $u_i \in C(B^*)$ ، $i = 1, \dots, k$ وجود داشته باشد به قسمی که u و v هر دو در S_{u_i} باشند. بنابراین با استفاده از لم (۳.۳.۳) می توان نوشت،

$$W(S_v) \leq W(S_B) \leq W(S_u) \leq W(S_{u_i}) \leq W(G)/2$$

علاوه بر این $W(G) - W(S_v) - W(S_B) \geq 0$ و $F(S, u) \leq F(S, v)$ با استفاده از روش استقرا داریم:

$$F(S, u_i) = \min_{v \in V^G(S_{u_i})} F(S, v)$$

$$\begin{aligned} F(S, u^*) &= \min_{i=1, \dots, k} F(S, u_i) \\ &= \min_{i=1, \dots, k} \min_{v \in V^G(S_{u_i})} F(S, v) \\ &= \min_{v \in V(G)} F(S, v) \end{aligned} \quad (4.3)$$

طبق لم (۱.۳.۳):

$$F(G, u^*) = \min_{v \in V(G)} F(G, v).$$

□

۴.۳ مساله پشتیبان ۲- میانه روی گراف های بلوکی

در این قسمت ما مساله پشتیبان ۲- میانه را روی گراف های بلوکی مطرح می کنیم. در مرجع [۲۱] نشان داده شده است که خاصیت بهینگی رأسی برای مساله پشتیبان p - میانه روی شبکه های عمومی برقرار است. بنابراین تنها چگونگی مکان یابی دو سرویس دهنده بر روی رئوس را بررسی می شود. فرض کنید ρ_1 و ρ_2 احتمال شکست دو سرویس دهنده در مساله پشتیبان ۲- میانه باشند. جواب این مساله با یک زوج رأس $\{u_1, u_2\}$ تعیین

می شود که به این معنی است که سرویس دهنده ها با احتمال شکست ρ_1 و ρ_2 به ترتیب روی رأس های u_1 و u_2 قرار می گیرند. جهت ساده سازی تعریف مساله، فرض می کنیم $\rho_1 = \rho_2 = \rho$ باشد. برای حالتی که $\rho_1 \neq \rho_2$ است، روش حل مساله مشابه همین روش می باشد. زمانی که احتمال های شکست یکسان هستند، ترتیب رئوس در جواب تاثیری ندارد اما برای سازگاری با حالتی که $\rho_1 \neq \rho_2$ باشد، ما هم چنان جواب ها را به صورت یک زوج مرتب از رئوس نشان می دهیم.

برای هر دو رأس u و v فرض کنید $\Pi(v, u) = (V, U)$ که $V \cap U = \emptyset$ و $V \cup U = V(G)$. علاوه بر این برای هر رأس $x \in V$ ، $d_G(x, v) \leq d_G(x, u)$ و برای هر رأس $y \in U$ ، $d_G(y, v) \leq d_G(y, u)$ می باشد. حال ما (V, U) را افزاز رأسی $\{v, u\}$ می نامیم. در یک مساله پشتیبان ۲- میانه، تابع هدف هر دو رأس u و v با یک احتمال شکست داده شده ρ به صورت زیر نشان داده می شود:

$$C_\rho(v, u) = (1 - \rho) \sum_{x \in V(G)} w_G(x) d(x, \{v, u\}) + \rho \left(\sum_{x \in V(G)} w(x) d(x, v) + \sum_{x \in V(G)} w(x) d(x, u) \right)$$

با استفاده از تعریف (۳.۴) می توان تابع فوق را به شکل زیر ساده نمود:

$$C_\rho(v, u) = F(G[V], v) + \rho F(G[U], v) + F(G[U], u) + \rho F(G[V], u) \quad (5.3)$$

که $\Pi(v, u) = (V, U)$ و هدف این مساله یافتن یک زوج رأس $\{m_1, m_2\}$ است به قسمی که،

$$C_\rho(m_1, m_2) = \min_{u, v \in V(G)} C_\rho(v, u)$$

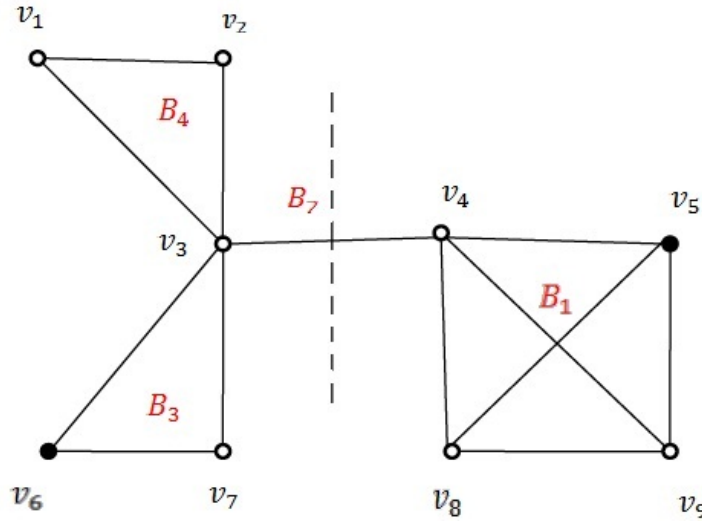
توجه کنید که اگر هر دو سرویس دهنده در یک رأس مکان یابی شوند، آنگاه این رأس در گراف G باید ۱- میانه باشد. برای یک رأس v در بلوک B ، $\{v, B\}$ زوج رأس-بلوک نامیده می شود.

اگر تمام یال های متصل به v در بلوک B را حذف کنیم، آنگاه گراف G به دو زیرگراف تجزیه می شود. زیرگراف شامل v با $G(v, B)$ و زیرگراف دیگر با $G^c(v, B)$ نشان داده می شود. مجموعه رئوس دو زیرگراف بالا به ترتیب با $V(v, B)$ و $V^c(v, B)$ نمادگذاری می شود. علاوه بر این دو گراف $\tilde{G}(v, B)$ و $\tilde{G}^c(v, B)$ را می توان ساخت که $\tilde{G}(v, B)$ و $\tilde{G}^c(v, B)$ به ترتیب با $G(v, B)$ و $G[V^c(v, B) \cup \{v\}]$ یکریخت^۳ باشند. همچنین وزن v در $\tilde{G}(v, B)$ برابر با $w_G(v) + \rho W(G^c(v, B))$ و وزن v در $\tilde{G}^c(v, B)$ برابر با $\rho W(G(v, B))$ است. وزن تمام رئوس دیگر و طول تمام یال های دیگر تغییر داده نمی شود.

برای دو رأس u و v ، نقطه میانی مسیر $P_G(u, v)$ را با نماد m_{uv} نشان می دهیم. فرض کنید m_{uv} روی یال (s, t) در بلوک B باشد، که رأس s به u نزدیک تر از v است. اگر m_{uv} بر یک رأس واقع شود، فرض می کنیم که m_{uv} بر s منطبق است.

لم ۱.۴.۳. [۱۹] $(V(s, B), V^c(s, B)) = \Pi(u, v)$ یا $(V^c(t, B), V(t, B)) = \Pi(u, v)$ برای هر دو رأس u و v در $V(G)$.

^۳ Isomorphic



شکل ۴.۳: $G(v_3, B_7)$ و $G^c(v_3, B_7)$

برهان. چون نقطه میانی m_{uv} روی یال (s, t) قرار دارد، داریم:

$$d_G(u, s) + d_G(s, m_{uv}) = d_G(m_{uv}, t) + d_G(t, v). \quad (۶.۳)$$

چون فرض کردیم که رأس s به u نزدیکتر است تا به v داریم

$$d_G(u, s) \leq d_G(v, s). \quad (۷.۳)$$

برای هر رأس x در زیرگراف $G(s, B)$ ، مسیر $P_G[x, v]$ باید از s بگذرد. بنابراین $d_G(x, v) = d_G(x, s) + d_G(s, v)$ با استفاده از نامساوی مثلث و رابطه ۷.۳ خواهیم داشت:

$$d_G(x, u) \leq d_G(x, s) + d_G(s, u) \leq d_G(x, s) + d_G(s, v) = d_G(x, v)$$

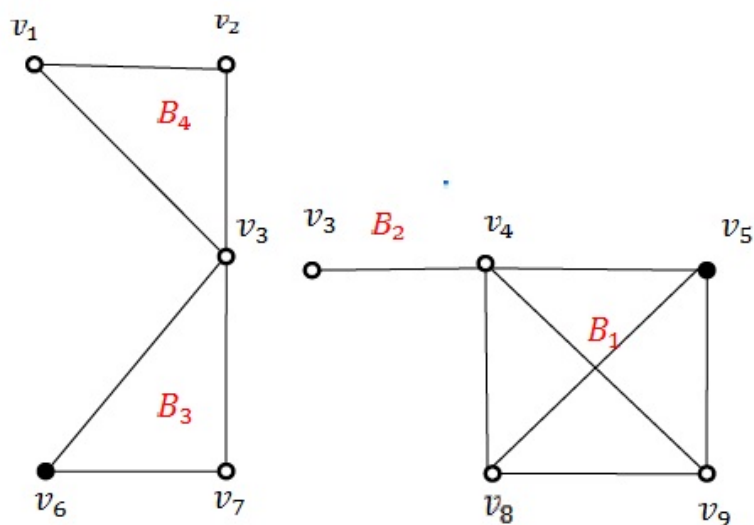
به طور مشابه می توان ثابت کرد که برای هر رأس y بر $G(t, B)$ داریم: $d_G(y, v) \leq d_G(y, u)$. در ادامه حکم را برای رئوس روی $G(w, B)$ که $w \in V(B)$ و $w \neq s, t$ در نظر می گیریم. واضح است که برای هر رأس z بر $G(w, B)$ ، مسیر $P_G[z, v]$ (به ترتیب $P_G[z, u]$) از w و s می گذرد. (به ترتیب w و t). بنابر این فرض که هر دو یال در یک بلوک طول یکسان دارند می توان گفت که اگر آنگاه $d_G(s, m_{uv}) \leq d_G(m_{uv}, t)$

$$d_G(z, u) = d_G(z, w) + d_G(w, s) + d_G(s, u) \geq d_G(z, w) + d_G(w, t) + d_G(t, v) = d_G(z, v)$$

در غیر این صورت $d_G(z, u) < d_G(z, v)$. بنابراین

$$d_G(u, v) = \begin{cases} (V(s, B), V^c(s, B)), & d_G(s, m_{uv}) \leq d_G(m_{uv}, t) \\ (V^c(t, B), V(t, B)), & \text{در غیر این صورت} \end{cases}$$

□



$\tilde{G}(v_3, B_2)$

شکل ۵.۳: $\tilde{G}^c(v_3, B_2)$

بنابر لم (۱.۴.۳) می دانیم که برای هر دور رأس u و v باید یک رأس r و بلوک B با $r \in V(B)$ موجود باشد به قسمی که: $(V(r, B), V^c(r, B)) = \Pi(u, v)$ یا $(V^c(r, B), V(r, B)) = \Pi(u, v)$. این زوج رأس-بلوک $\{r, B\}$ زوج رأس-بلوک مرکزی u و v نامیده می شود.

قضیه ۲.۴.۳. [۱۹] فرض کنید $\{m_1, m_2\}$ پشتیبان ۲- میانه روی رئوس مختلف گراف بلوکی G باشد. بدون کاستن از کلیت فرض می کنیم $(V(r, B), V^c(r, B)) = \Pi(m_1, m_2)$ که $\{r, B\}$ زوج رأس-بلوک مرکزی m_1 و m_2 است. آنگاه m_1 و m_2 در روابط زیر صدق می کنند:

$$F(\tilde{G}(r, B), m_1) = \min_{v \in V(r, B)} F(\tilde{G}(r, B), v) \quad (۸.۳)$$

$$F(\tilde{G}^c(r, B), m_2) = \min_{v \in V^c(r, B)} F(\tilde{G}^c(r, B), v) \quad (۹.۳)$$

برهان. در اینجا ما فقط قسمت (۹.۳) را اثبات می کنیم. اثبات قسمت (۸.۳) مشابه همین اثبات است. به برهان خلف فرض می کنیم m_2 در رابطه (۹.۳) صدق نمی کند و z را رأسی در نظر بگیرید که $F(\tilde{G}^c(r, B), z)$ در $V^c(r, B)$ کمترین مقدار باشد، آنگاه

$$\begin{aligned} F(\tilde{G}^c(r, B), z) &= \sum_{x \in \{v^c(r, B) \cup r\}} w(x) d_G(x, z) \\ &= \sum_{x \in v^c(r, B)} w(x) d_G(x, z) + \rho W(G(r, B)) d_G(r, z) \\ &= F(G^c(r, B), z) + \rho W(G(r, B)) d_G(r, z) \\ &< F(G^c(r, B), m_2) + \rho W(G(r, B)) d_G(r, m_2) \end{aligned}$$

چون برای هر رأس $x \in V(r, B)$ و $y \in V^c(r, B)$ ، مسیر $P_G[x, y]$ باید از r عبور کند، در نتیجه $d_G(x, z) = d_G(x, r) + d_G(r, z)$ بنابراین

$$\begin{aligned} F(G^c(r, B), z) + \rho F(G(r, B), z) &= F(G^c(r, B), z) + \rho W(G(r, B))d_G(r, z) + \rho F(G(r, B), r) \\ &< F(G^c(r, B), m_1) + \rho W(G(r, B))d_G(r, m_1) + \rho F(G(r, B), r) \\ &= F(G^c(r, B), m_1) + \rho F(G(r, B), m_1) \end{aligned}$$

در نتیجه

$$\begin{aligned} C_\rho(m_1, m_2) &= F(G(r, B), m_1) + \rho F(G^c(r, B), m_1) + F(G^c(r, B), m_2) + \rho F(G(r, B), m_2) \\ &> F(G(r, B), m_1) + \rho F(G^c(r, B), m_1) + F(G^c(r, B), z) + \rho F(G(r, B), z) \\ &\geq C_\rho(m_1, z) \end{aligned}$$

از آخرین نامساوی می توان چنین نتیجه گرفت که $(V(r, B), V^c(r, B))$ ممکن است افراز رأسی m_1 و z نباشد که این تناقض است. لذا فرض خلف باطل و حکم برقرار است. $\square$

با استفاده از قضیه (۲.۴.۳) می توان دید که

$$F(\tilde{G}(r, B), m_1) = \min_{v \in V(r, B)} F(\tilde{G}(r, B), v)$$

و

$$F(\tilde{G}^c(r, B), m_2) \geq \min_{v \in \tilde{V}^c(r, B)} F(\tilde{G}^c(r, B), v)$$

بنابراین فقط m_1 میانه $\tilde{G}(r, B)$ است و چون $V^c(r, B) \subset \tilde{V}(r, B)$ ، رأسی است که $F(\tilde{G}^c(r, B), m_2)$ در $V^c(r, B)$ کمترین مقدار را دارد. در نتیجه m_2 را میانه نزدیک $\tilde{G}^c(r, B)$ می نامیم. فرض کنید m و m^c میانه و میانه نزدیک $\tilde{G}(v, B)$ و $\tilde{G}^c(v, B)$ به ازای زوج رأس-بلوک $\{v, B\}$ باشد. حال تابع هدف (۵.۳) را به فرم تابع هزینه وابسته به زوج رأس-بلوک بازنویسی می کنیم،

$$\begin{aligned} C'_\rho(v, B) &= F(G(v, B), m) + \rho F(G^c(v, B), m) + F(G^c(v, B), m^c) + \rho F(G(v, B), m^c) \\ &= F(\tilde{G}(v, B), m) + F(\tilde{G}^c(v, B), m^c) + \rho F(G, v) \end{aligned} \quad (۱۰.۳)$$

قضیه ۳.۴.۳. [۱۹] برای هر زوج رأس (m, m^c) از $\{v, B\}$ ، نامساوی $C'_\rho(v, B) \geq C_\rho(m, m^c)$ برقرار است.

برهان. فرض کنید $(V_1, V_2) = \Pi(m, m^c)$. برای چهار زیرمجموعه رأسی V_1, V_2, V_3, V_4 ، $V^c(v, B)$ و $V(v, B)$ ، تنها دو امکان وجود دارد: یا $V_1 \subseteq V(v, B)$ یا $V_1 \subseteq V^c(v, B)$. در اینجا ما تنها حالتی که $V_1 \subseteq V(v, B)$

را بررسی می کنیم. اثبات حالت دیگر مشابه است.

$$\begin{aligned}
 C_\rho(m, m^c) &= F(G[V_1], m) + \rho F(G[V_2], m) + F(G[V_2], m_c) + \rho F(G[V_1], m_c) \\
 &= F(G[V_1], m) + F(G^c(v, B), m_c) + F(G(v, B) - G[V_1], m^c) + \rho F(G[V_1], m^c) \\
 &\quad + \rho F(G^c(v, B), m) + F(G(v, B) - G[V_1], m) \\
 &\leq F(G[V_1], m) + F(G^c(v, B), m_c) + F(G(v, B) - G[V_1], m) + \rho F(G[V_1], m^c) \\
 &\quad + \rho F(G^c(v, B), m) + F(G(v, B) - G[V_1], m^c) \\
 &= C'_\rho(v, B)
 \end{aligned}$$

□

از قضیه های (۲.۴.۳) و (۳.۴.۳) می دانیم که $C_\rho(m_1, m_2) = \min_{\{v, B\}} C'_\rho(v, B)$ وزن های پیکربندی S داده شده است. با استفاده از قضیه (۲.۳.۳) m ، $F(\tilde{G}(v, B), m)$ و m^c ، $F(\tilde{G}(v, B), m^c)$ را به ترتیب با زمان های $O(|\tilde{V}(v, B)| + |\tilde{E}(v, B)|)$ و $O(|\tilde{V}(v, B)| + |\tilde{E}(v, B)|)$ محاسبه کرد، که $O(n + m) = O(|\tilde{V}(v, B)| + |\tilde{E}(v, B)| + |\tilde{V}^c(v, B)| + |\tilde{E}^c(v, B)|)$ هم چنین می توانیم در مرحله پیش پردازش برای تمام رئوس v ، $F(G, v) = F(S, v)$ را در زمان خطی محاسبه کنیم. بنابراین برای یک زوج رأس-بلوک داده شده $\{v, B\}$ ، $C'_\rho(v, B)$ می تواند در زمان $O(n + m)$ محاسبه شود. چون یک زوج رأس-بلوک متناظر با یک یال در $E(S)$ است و $|E(S)| = n + l - 1 = O(n)$ و $C'_\rho(v, B)$ برای تمامی زوج های رأس-بلوک $\{v, B\}$ می تواند در زمان $O(n^2 + nm)$ محاسبه شود و برای رسیدن به جواب می توانیم یک زوج رأس-بلوک با کمترین مقدار را در زمان $O(n)$ انتخاب کنیم. در نتیجه یک الگوریتم با پیچیدگی زمانی $O(n^2 + nm)$ خواهیم داشت. به منظور کاهش بیشتر زمان، نشان می دهیم که با استفاده از الگوریتم $BACKUP_2MEDIAN(T, \rho)$ که در فصل ۲ آورده شده، می توان از محاسبات تکراری در جستجو برای یافتن زوج رأس بهینه از میان تمام زوج های رأس-بلوک روی درخت اجتناب نمود.

مشابه روش الگوریتم $BACKUP_2MEDIAN(T, \rho)$ ، پیکربندی S را در میانه خود یعنی m^* ریشه دار می کنیم. برای یک یال $e = (x, y) \in E(S)$ با $x = par(y)$ فرض کنید $X(e)$ و $Y(e)$ دو زیردرخت به دست آمده از حذف یال e از S با x در $X(e)$ و افزایش وزن های x و y با وزن کل زیردرخت دیگر ضرب در ρ باشند. یعنی داریم: $w_{X(e)}(x) = w_S(x) + \rho W(S_y)$ و $w_{Y(e)}(Y) = w_S(y) + \rho(W(G) - W(S_y))$. وزن سایر رئوس و طول تمام یال ها تغییر نمی کند. چون $x = par(y)$ ، پس میانه m^* روی زیردرخت $X(e)$ قرار می گیرد. فرض کنید m_U و m_L به ترتیب میانه های $X(e)$ و $Y(e)$ باشند. به طور مشابه، دو رأس v_U و v_L را به ترتیب میانه راسی $X(e)$ و $Y(e)$ تعریف می کنیم چنان که

$$F(X(e), v_U) = \min_{u \in V^G(X(e))} F(X(e), u)$$

و

$$F(Y(e), v_L) = \min_{u \in V^G(Y(e))} F(Y(e), u).$$

لم ۴.۴.۳. [۱۹] فرض کنید $e = (v, B)$ یالی در $E(S)$ باشد.

(۱) اگر $B = \text{par}(v)$ باشد، آنگاه v_L میانه $\tilde{G}(v, B)$ با تابع هدف $F(\tilde{G}(v, B), v_L) = F(Y(e), v_L)$ است و v_U میانه نزدیک $\tilde{G}^c(v, B)$ با $\frac{\rho}{4} l_G(B) W(S_v)$ است $F(\tilde{G}^c(v, B), v_U) = F(X(e), v_U)$ است.

(۲) اگر $v = \text{par}(B)$ باشد، آنگاه v_L میانه نزدیک $\tilde{G}^c(v, B)$ با $F(\tilde{G}^c(v, B), v_L) = F(Y(e), v_L)$ است و v_U میانه $\tilde{G}(v, B)$ با $\frac{\rho}{4} l_G(B) (W(G) - W(S_B))$ است $F(\tilde{G}(v, B), v_U) = F(X(e), v_U)$ است.

برهان. در اینجا تنها حالتی که اگر $B = \text{par}(v)$ باشد، v_U میانه نزدیک $\tilde{G}^c(v, B)$ است، اثبات می شود. سایر قسمت ها به روش مشابه اثبات می شود. چون $B = \text{par}(v)$ ، می توان نوشت، $V^G(S_v) = V^G(Y(e)) = V^c(v, B)$ و $V(v, B) = V^G(S_v^c) = V^G(X(e)) = V^c(v, B)$ و $W(S_v^c) = W(S_v)$ و $W(G(v, B)) = W(S_v)$ بنابراین $W(G^c(v, B)) = W(S_v)$ می باشد. با استفاده از ساختار $X(e)$ ، می دانیم که وزن تمام بلوک ها به جز بلوک B صفر است که $W_{x(e)}(B) = \rho W(S_v) = \rho W(G(v, B))$. بنابراین برای هر رأس u بر $X(e)$ ،

$$\begin{aligned} F(X(e), u) &= \sum_{x \in V^G(X(e))} w_S(x) d_S(x, u) + W_{x(e)}(B) d_S(B, u) \\ &= \sum_{x \in V^c(v, B)} w_G(x) d_G(x, u) + \rho W_{S_v}(B) (d_G(v, u) - \frac{1}{4} l_G(B)) \\ &= F(\tilde{G}^c(v, B), u) - \frac{\rho}{4} l_G(B) W(S_v). \end{aligned}$$

با استفاده از تعریف v_U داریم:

$$\begin{aligned} F(\tilde{G}^c(v, B), v_U) &= F(X(e), v_U) + \frac{\rho}{4} l_G(B) W(S_v) \\ &= \min_{u \in V^G(X(e))} \left(F(X(e), u) + \frac{\rho}{4} l_G(B) W(S_v) \right) \\ &= \min_{u \in V^c(v, B)} F(\tilde{G}^c(v, B), u) \end{aligned}$$

بنابراین v_U میانه نزدیک $\tilde{G}^c(v, B)$ با مقدار تابع هدف زیر است: $F(X(e), v_U) + \frac{\rho}{4} l_G(B) W(S_v)$. □

به منظور محاسبه $C'_\rho(v, B)$ برای تمام زوج های رأس-بلوک $\{v, B\}$ ، ما باید برای تمام یال های $e = (v, B)$ بر S ، v_L و v_U را به ترتیب از $X(e)$ و $Y(e)$ بیابیم. واضح است اگر m_U یا m_L یک رأس باشد، آنگاه $v_U = m_U$ یا $v_L = m_L$. در ادامه، چگونگی تعیین v_L و v_U زمانی که m_L و m_U بلوک هایی

در S هستند، را بررسی می کنیم. فرض کنید که $d(m_U)$ و $d(m_L)$ به ترتیب درجات m_U و m_L در $X(e)$ و $Y(e)$ باشند. فرض کنید $X^1(e), X^2(e), \dots, X^{d(m_U)}(e)$ مولفه هایی باشند که با حذف m_U در $X(e)$ به دست آمده اند. همچنین $W(X^i(e)) = \sum_{v \in V(X^i(e))} w_{X(e)}(v)$ ، مجموع وزن تمام گره ها در $X^i(e)$ ، $i = 1, 2, \dots, d(m_U)$ است. به طور مشابه زیردرخت های $Y^i(e)$ و با وزن های $d(m_L)$ ، $1, 2, \dots, d(m_U)$ را داریم.

لم ۵.۴.۳ [۱۹] فرض کنید $e = (x, y)$ یالی در $E(S)$ با $x = \text{par}(y)$ باشد. فرض کنید که میانه های m_U و m_L بلوک باشند. بنابراین

(۱) میانه رأسی v_U از $X(e)$ ، j -امین همسایه از m_U در $X(e)$ ، $j = 1, 2, \dots, d(m_U)$ است که در $W(X^j(e)) = \max_{1 \leq i \leq d(m_U)} W(X^i(e))$ صدق می کند.

(۲) میانه رأسی v_L از $Y(e)$ ، k -امین همسایه از m_L در $Y(e)$ ، $k = 1, 2, \dots, d(m_L)$ است که در $W(Y^k(e)) = \max_{1 \leq i \leq d(m_L)} W(Y^i(e))$ صدق می کند.

برهان. ما تنها صحت حالتی که v_U بر $X(e)$ است را نشان می دهیم. قسمت دوم لم با روش مشابه اثبات می شود. به منظور تحلیل ساده تر، زیردرخت $X(e)$ را که در میانه خود m_U ریشه دار کرده ایم را با $X'(e)$ نشان می دهیم. در نتیجه همسایه های m_U در $X(e)$ تمام فرزندان m_U در $X'(e)$ هستند. فرض کنید $C'(m_U) = \{u_1, u_2, \dots, u_{d(m_U)}\}$ مجموعه فرزندان m_U در $X'(e)$ باشند. مشابه لم (۵.۳.۳) و (۵.۳.۳)، به راحتی برای یک فرزند u_i در $C'(m_U)$ می توان نوشت،

$$F(X'(e), u_i) = F(X'(e), m_U) + l_G(m_U) \left(\frac{1}{p} W(X'(e)) - W(X'_{u_i}(e)) \right) \quad (۱۱.۳)$$

برای دو رأس u و v با $u = DVP(v)$ بر $X'(e)$ ؛

$$F(X'(e), v) = F(X'(e), u) + l_G(B') \left(W(X'(e)) - W(X'_{B'}(e)) - W(X'_v(e)) \right)$$

که B' بلوک شامل u و v است.

چون $X'(e)$ تنها در ریشه اش با $X(e)$ متفاوت است، برای هر گره v بر $X(e)$ و $W(X'(e)) = W(X(e))$ داریم، $F(X(e), v) = F(X'(e), v)$. همچنین از نماد $X'_v(e)$ برای نمایش زیرگراف ماکسیمال $X'(e)$ که در گره v ریشه دار شده است، استفاده می کنیم. بدیهی است که $W(X'_{u_i}(e)) = W(X^i(e))$. با استفاده از لم (۳.۳.۳) و اثبات قضیه (۶.۳.۳)، نتیجه می گیریم که رأس میانه v_U ، فقط j -امین همسایگی m_U بر $X(e)$ است همان گونه که $W(X^j(e)) = \max_{1 \leq i \leq d(m_U)} W(X^i(e))$. $\square$

با استفاده از لم (۵.۴.۳) می دانیم که کلید یافتن v_U بر $X(e)$ و v_L بر $Y(e)$ ، محاسبه تمام مجموع وزن های $W(X^i(e))$ ، $i = 1, \dots, d(m_U)$ و $W(Y^j(e))$ ، $j = 1, \dots, d(m_L)$ است، به این شرط که m_U و m_V به دست آمده باشند.

لم ۶.۴.۳. ([۴۷]) فرض کنید S یک درخت ریشه دار در میانه خود m^* باشد و S_1 و S_2 سنگین ترین و دومین سنگین ترین زیردرخت ریشه دار در فرزند m^* باشند. برای هر یال e ، میانه $X(e)$ ، در مسیری بین m^* و میانه S_i است که $i = 2$ اگر e در S_1 باشد. در غیر این صورت $i = 1$ است.

در پیکربندی S با ریشه m^* برای هر یال $e = (x, y)$ فرض می کنیم $C(m_U) = \{x_1, x_2, \dots, x_p\}$ و $C(m_L) = \{y_1, y_2, \dots, y_p\}$ باشد. x^* و y^* را به ترتیب فرزندان $C(m_U)$ و $C(m_L)$ در نظر بگیرید که در $W(S_{y^*}) = \max_{1 \leq j \leq q} W(S_{y_j})$ و $W(S_{x^*}) = \max_{1 \leq i \leq p} W(S_{x_i})$ صدق می کنند. به ویژه اگر m_U تنها ریشه m^* باشد، بنابراین فرزندی از m^* را با $\hat{x}$ نمادگذاری می کنیم که $S_{\hat{x}}$ ، دومین سنگین ترین زیردرخت ریشه دار در فرزند m^* است. بر پایه لم فوق نتیجه گیری زیر کاملاً صحیح خواهد بود.

نتیجه ۷.۴.۳. [۱۹] فرض کنید $e = (x, y)$ با $x = \text{par}(y)$ یک یال در $E(S)$ باشد. اگر $m_U \neq m^*$ ، آنگاه

$$\max_{i=1, \dots, d(m_U)} \{W(X^i(e))\} = \max\{W(S_{x^*}), W(G) - (1 - \rho)W(S_y) - W(S_{m_U})\}. \quad (۱۲.۳)$$

در غیر این صورت وجود دارد $x_l \in C(m_U)$ به قسمی که e در S_{x_l} است. اگر $x_1 = x^*$ ، آنگاه

$$\max_{i=1, \dots, d(m_U)} \{W(X^i(e))\} = \max\{W(S_{\hat{x}}), W(S_{x^*}) - (1 - \rho)W(S_y)\}. \quad (۱۳.۳)$$

در غیر این صورت

$$\max_{i=1, \dots, d(m_U)} \{W(X^i(e))\} = W(S_{x^*})$$

نتیجه ۸.۴.۳. [۱۹] فرض کنید $e = (x, y)$ با $x = \text{par}(y)$ یک یال در $E(S)$ باشد. اگر $m_L \neq y$ ، آنگاه

$$\max_{j=1, \dots, d(m_L)} \{W(Y^j(e))\} = \max\{W(S_{y^*}), \rho W(G) + (1 - \rho)W(S_y) - W(S_{m_L})\}. \quad (۱۴.۳)$$

در غیر این صورت

$$\max_{j=1, \dots, d(m_L)} \{W(Y^j(e))\} = W(S_{y^*}).$$

در مرحله پیش پردازش، باید ابتدا پیکربندی وزن دار S را در زمان $O(n+m)$ بسازیم. و ما می توانیم میانه m^* در گراف G را بیابیم و $F(G, m^*)$ را در زمان $O(n+m)$ محاسبه کنیم. علاوه بر این $C\rho(m^*, m^*)$ را می توان در زمان ثابت به دست آورد. به عبارت دیگر ما باید پشتیبان ۲- میانه بر رئوس مختلف را به دست آوریم. ما می توانیم تمام مجموع های وزنی $\{W(S_x)\}_{x \in V(S)}$ را در زمان $O(n)$ روی S محاسبه کنیم. همچنین برای هر گره $x \in V(S)$ زمان لازم برای محاسبه $W(S_{u_*}) = \max_{u \in C(x)} \{W(S_u)\}$ ، $O(n)$ است. به ویژه برای ریشه m^* ، $W(S_{\hat{u}}) = \max_{u \neq u^* \in C(x)} \{W(S_u)\}$ می تواند در زمان $O(n)$ به دست آید. با

استفاده از الگوریتم $BACKUP_2MEDIAN$ ، به ازای هر یال $e \in E(S)$ یافتن میانه های m_L و m_U زمان $O(n \log n)$ نیاز دارد. برای هر یال $e = (x, y)$ در $E(S)$ با $x = par(y)$ ، با استفاده از معادلات نتیجه های (۷.۴.۳) و (۸.۴.۳) $\max_{i=1, \dots, d(m_U)} \{W(X^i(e))\}$ و $\max_{j=1, \dots, d(m_L)} \{W(Y^j(e))\}$ می توانند در زمان ثابت پاسخ داده شوند. علاوه بر این ما می توانیم رأس های میانه v_L ، v_U و مقادیر تابع هدف آنها $F(X(e), v_U)$ و $F(Y(e), v_L)$ را با استفاده از لم (۵.۴.۳) در زمان ثابت به دست آوریم. چون هر یال $e \in E(S)$ متناظر با یک زوج رأس-بلوک $\{v, B\}$ است، $F(\tilde{G}(s, B), m)$ و $F(\tilde{G}^c(s, B), m^c)$ مقدار $C_{\rho'}(v, B)$ با استفاده از لم ۴.۴.۳ و رابطه ۱۰.۳ می تواند در زمان ثابت محاسبه شود. بنابراین ما می توانیم $C_{\rho'}(v, B)$ را برای تمام زوج های رأس-بلوک $\{v, B\}$ به دست آوریم و کمترین مقدار را برای تعیین پشتیبان ۲- میانه روی رئوس مختلف در زمان $O(n)$ بیابیم. با مقایسه با $C_{\rho}(m^*, m^*)$ می توانیم جواب بهینه را به دست آوریم. بنابراین کل زمان لازم $O(n \log n + m)$ است و قضیه زیر را داریم.

قضیه ۹.۴.۳ [۱۹] اگر احتمال شکست $\rho_1 = \rho_2 = \rho$ ، پشتیبان ۲- میانه ای گراف بلوکی G می تواند در زمان $O(n \log n + m)$ پاسخ داده شود، که هر دویال در یک بلوک طول یکسان دارند.

در حالتی که $\rho_1 \neq \rho_2$ ، مساله به فرم زیر نشان داده می شود:

برای یک زوج رأس $\{u, v\}$ ، بدون از دست دادن کلیت فرض می کنیم که $(V(r, B), V^c(r, B)) = \Pi(u, v)$. ابتدا تابع هدف $\{u, v\}$ با احتمال شکست $\{\rho_1, \rho_2\}$ به صورت زیر بازنویسی می شود:

$$C_{\rho_1, \rho_2}(u, v) = (1 - \rho_1) \left(F(G(r, B), u) + \rho_2 F(G^c(r, B), u) \right) + (1 - \rho_2) \left(F(G^c(r, B), v) + \rho_1 F(G(r, B), v) \right)$$

مشابه ساختار ارایه شده در فصل (۲) برای $e = (x, y) \in E(S)$ با $x = par(y)$ و $X_i(e)$ و $Y_j(e)$ را تعریف می کنیم به طوری که وزن x و y به ترتیب برابر است با $w_S(x) + \rho_j W(S_y)$ و $w_S(y) + \rho_i (W(S) - W(S_y))$ برای $i, j \in \{1, 2\}, i \neq j$. بنابراین در حالتی که $\rho_1 \neq \rho_2$ برای یک یال دو زوج از میانه ها وجود دارد. علاوه بر این زیرگراف های $\tilde{G}_i(v, B)$ و $\tilde{G}_j^c(v, B)$ را متناظر با $G(v, B)$ و $G^c(v, B)$ تعریف می کنیم که وزن v در $\tilde{G}_i(v, B)$ و $\tilde{G}_j^c(v, B)$ به ترتیب برابر $w_G(v) + \rho_j W(G^c(v, B))$ و $\rho_i W(G(v, B))$ برای $i, j \in \{1, 2\}$ و $i \neq j$ است. در نتیجه برای یک زوج رأس-بلوک، دو زوج میانه و میانه نزدیک $\{m, m^c\}$ موجود است. بنابراین تابع زوج رأس-بلوک $\{v, B\}$ به فرم زیر نوشته می شود:

$$C_{\rho_1, \rho_2}(v, B) = (1 - \rho_1) \left(F(G(v, B), m) + \rho_2 F(G^c(v, B), m) \right) + (1 - \rho_2) \left(F(G^c(v, B), m^c) + \rho_1 F(G(v, B), m^c) \right) \\ = (1 - \rho_1) F(\tilde{G}_1(v, B), m) + (1 - \rho_2) F(\tilde{G}_2^c(v, B), m^c) + (\rho_2 - \rho_1 \rho_2) F(G, v) + (\rho_1 - \rho_2) F(G(v, B), v)$$

در پیکربندی S اگر $B = par(v)$ باشد، آنگاه $F(G(v, B), v) = F(S_v, v)$ خواهد بود. درغیراین صورت $F(G(v, B), v) = F(S, v) - F(S_B, B) - \frac{1}{4}l_G(B)W(S_B)$ بنابراین می توانیم الگوریتم فوق را با تعدیل احتمال شکست برای یافتن پشتیان ۲- میانه به کار ببریم.

قضیه ۱۰.۴.۳. مساله پشتیان ۲- میانه روی گراف بلوکی که هر دو یال در یک بلوک دارای طول یکسان باشند، می تواند در زمان $O(n \log n + m)$ پاسخ داده شود.

فصل ۴

مساله ۲- میانه با وزن مثبت- منفی روی
درخت

۱.۴ مقدمه

در این فصل ما مساله ۲-میانه را روی شبکه‌هایی که وزن بخشی از رئوسشان منفی است، بررسی می‌کنیم. در این حالت مساله متناظر، مساله ۲-میانه با وزن مثبت-منفی نامیده می‌شود. بورکارد و همکارانش ابتدا دو تابع هدف متفاوت برای این مساله پیشنهاد کردند [۱۱]. مدل‌های MWD و WMD را طراحی و به ترتیب الگوریتم‌هایی با پیچیدگی زمانی $O(n^2)$ و $O(n^3)$ روی درخت زمانی که $p = 2$ است ارائه دادند. پس از آن پیچیدگی الگوریتم‌ها روی درخت‌ها زمانی که $p = 2$ است توسط بنکوکزی [۸]^۱، بنکوکزی و برتون^۲ و بهاتاچاریا [۹]^۳ به ترتیب به $O(n \log n)$ و $O(nh \log^2 n)$ بهبود یافت، که h ارتفاع درخت است. بورکارد و هاتزل [۲۴] یک الگوریتم با پیچیدگی زمانی $O(n)$ برای مدل MWD و یک الگوریتم با پیچیدگی زمانی $O(n^2)$ برای مدل WMD روی سیکل زمانی که $p = 2$ ، پیشنهاد کردند. چنگ و همکارانش [۲۰] نیز مساله ۱-میانه با وزن مثبت-منفی روی درخت‌ها را با زیردرخت شکل یافته مشتریان پاسخ دادند.

در این فصل ما مساله ۲-میانه با وزن مثبت-منفی که بورکارد و همکارانش مطرح کردند را با مدل MWD بررسی می‌کنیم. آن‌ها برای حل این مساله از الگوریتم ارایه شده برای مساله ۱-میانه با وزن مثبت-منفی روی کاکتوس که توسط بورکارد و کراروپ [۱۵] مطرح شده، استفاده کردند و الگوریتمی با زمان $O(n^2)$ ارایه کردند که n تعداد رئوس درخت مورد بررسی است. ما پس از تعریف مساله این الگوریتم را بیان می‌کنیم. در قسمت پایانی فصل نیز الگوریتمی که بورکارد و همکارانش [۱۱] برای مساله ۲-میانه با وزن مثبت-منفی روی مسیر پیشنهاد کردند را بیان می‌کنیم.

۲.۴ تعریف مساله و نمادگذاری

فرض کنید $N = (V, E, c, w)$ یک شبکه همبند به گونه‌ای باشد که V مجموعه رئوس با $|V| = n$ و $\{v_1, v_2, \dots, v_n\}$ مجموعه یال‌های E ، $c: E \rightarrow R^+$ تابع طول یال‌ها و $w: V \rightarrow R$ تابع وزن رئوس شبکه N باشد که در این مساله وزن رأس‌ها دلخواه است. یک نقطه در N یا یک رأس $v \in V$ یا یک نقطه داخلی یال $e \in E$ است. فرض کنید p سرویس‌دهنده در p نقطه از N مستقر شده‌اند که با مجموعه $X_p = \{x_1, x_2, \dots, x_p\}$ نشان داده می‌شوند. دو تابع هدف می‌توان برای مساله بیان کرد، فرض کنید

$$F_1(X_p) = \sum_{i=1}^n \min_{1 \leq j \leq p} w_i d(v_i, x_j) \quad (1.4)$$

^۱ Benkokzi

^۲ Breton

^۳ Bhattacharya

$$F_p(X_p) = \sum_{i=1}^n w_i \min_{1 \leq j \leq p} d(v_i, x_j) \quad (2.4)$$

که تابع هدف‌های (۱.۴) و (۲.۴) به ترتیب همان مدل‌های MWD و WMD ارایه شده توسط کراروپ است که در فصل (۱) بیان کردیم. بنابراین مسایل p -میانه به صورت زیر است:

$$(P1) \quad \text{یک مجموعه } X^* \text{ با حداکثر } p \text{ نقطه در } N \text{ را به گونه‌ای بیابید که،}$$

$$F_1(X^*) = \min_{X \subseteq N, |X| \leq p} F_1(X)$$

$$(P2) \quad \text{یک مجموعه } X^* \text{ با حداکثر } p \text{ نقطه در } N \text{ را به گونه‌ای بیابید که،}$$

$$F_p(X^*) = \min_{X \subseteq N, |X| \leq p} F_p(X)$$

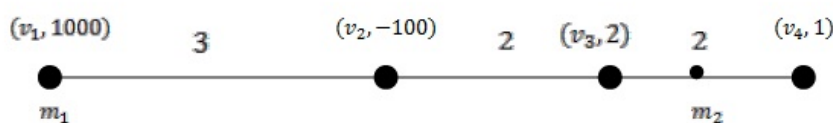
هر دو مساله (P1) و (P2) می‌توانند مساله ۱-میانه تعریف شده در [۱۵] را بررسی کنند. برای $p = 1$ هر دو تابع هدف F_1 و F_p با تابع هدف مطرح شده در [۱۵] یکسان هستند. توجه کنید اگر تمام وزن‌های w_i ، $1 \leq i \leq n$ مثبت باشند، آن‌گاه به ازای تمام X ، $F_1 = F_p$ و هر دو مساله همانند مساله p -میانه کلاسیک هستند.

به منظور کلاس‌بندی مسایل، از فهرست ارایه شده توسط هامپر و نیگل [۳۱] استفاده می‌کنیم. مطابق این جدول مساله (P1) را با نماد $\sum_1 |N| w_i \leq O \mid d(V, N) \mid$ نشان می‌دهیم. اولین بخش نمادگذاری، تعداد تسهیلات را نشان می‌دهد. دومین قسمت نشان می‌دهد که ما یک مساله روی شبکه N داریم. سومین قسمت تعیین کننده وزن رئوس است که در این مساله وزن رأس‌ها دلخواه است. قسمت چهارم نوع فاصله‌ای که برای مساله استفاده می‌شود را نشان می‌دهد و قسمت آخر هم بیان می‌کند که تابع هدف میانه همان تابع هدف تعریف شده در ۳.۴ می‌باشد. به همین صورت مساله (P2) با نماد $\sum_p |N| w_i \leq O \mid d(V, N) \mid$ نمایش داده می‌شود. به جای N اگر شبکه مورد بررسی یک درخت T و یا یک مسیر P باشد در قسمت دوم، از T و P استفاده می‌کنیم. اگر نیاز باشد که میانه‌ها تنها روی رئوس شبکه مورد بررسی مکان‌یابی شوند، به جای قسمت چهارم باید از نماد $d(V, V)$ استفاده شود.

همان‌طور که در فصل (۱) نیز بیان کردیم یک قضیه اساسی در مساله p -میانه کلاسیک روی یک شبکه با وزن‌های نامنفی وجود دارد که خاصیت بهینگی رأسی نام دارد. در مرجع [۱۵] در قالب یک مثال برای مساله ۱-میانه نشان داده شده است که این قضیه وقتی وزن‌های w_i دلخواه باشند برقرار نیست. اما اگر شبکه مورد بررسی یک درخت باشد و از تابع هدف (۳.۴) استفاده کنیم، خاصیت بهینگی رأسی برقرار است.

قضیه ۱.۲.۴. [۱۱] خاصیت بهینگی رأسی برای مساله $\sum_1 |T| w_i \leq O \mid d(V, T) \mid$ روی درخت T برقرار است.

برهان. فرض کنید مجموعه $X^* = \{m_1, m_2, \dots, m_p\}$ یک جواب بهینه $\sum_1 |T| w_i \leq O \mid d(V, T) \mid$ با کمترین نقطه غیررأسی باشد. اگر تمام m_i ها، $1 \leq i \leq p$ رأس باشند، حکم برقرار است. فرض کنید



شکل ۱.۴: خاصیت بهینگی رأسی برای مساله $P2$ برقرار نیست.

$m_k \in X^*$ یک نقطه غیر رأسی درخت باشد. فرض کنید (v_r, v_q) یالی از درخت T و شامل m_k باشد. فاصله m_i از v_r را با t نشان می‌دهیم. فرض کنید V_k مجموعه‌ای از رئوس V باشند که از m_k سرویس می‌گیرند یعنی مجموعه رئوس $v_i \in V$ که در نامساوی $\min_{1 \leq j \leq p} w_i d(v_i, m_j) = w_i d(v_i, m_k)$ صدق می‌کنند.

فرض کنید m_k می‌تواند در طول یال (v_r, v_q) حرکت کند درحالی‌که سایر میانه‌ها ثابت هستند. تابع هدف زیر که به m_k وابسته است را در نظر بگیرید:

$$F'_1(X^*) = \sum_{v_i \in V_k} w_i d(v_i, m_k) + \sum_{v_i \in V - V_k} \min_{1 \leq j \leq p, j \neq k} w_i d(v_i, m_j)$$

همان‌گونه که m_k در طول (v_r, v_q) حرکت می‌کند، فاصله $d(v_i, m_k)$ برای هر $v_i \in V_k$ یک تابع خطی از t است. بنابراین $F'_1(X^*)$ یک تابع خطی در t است همان‌گونه که m_k در طول (v_r, v_q) حرکت می‌کند، و می‌توان نتیجه گرفت که این تابع مینیمم مقدار خود را زمانی می‌پذیرد که m_k بر v_r یا v_q منطبق باشد. بدون کاستن از کلیت فرض کنید که اگر m_k منطبق بر v_r باشد، F'_1 مینیمم مقدار خود را می‌پذیرد و $X^* \setminus \{m_k\} \cup \{v_r\}$ را با نماد $\tilde{X}$ نشان دهید. واضح است که $F_1(\tilde{X}) \leq F'_1(X^*) = F_1(X^*)$. از تعریف F_1 داریم $F_1(\tilde{X}) \leq F'_1(\tilde{X})$. در نتیجه $F_1(\tilde{X}) \leq F_1(X^*)$. نامساوی آخر بر این دلالت دارد که $\tilde{X}$ یک جواب بهینه برای مساله p - میانه مورد بررسی با نقطه غیر رأسی کمتر از X^* است. اما این با این فرض که X^* یک جواب بهینه با کمترین نقطه غیر رأسی است، در تناقض است. $\square$

مثال (۲.۲.۴) نشان می‌دهد که خاصیت بهینگی رأسی برای مساله $\sum p \mid T \mid w_i \leq O \mid d(V, T) \mid$ حتی اگر $p = 2$ و T یک مسیر باشد، برقرار نیست.

مثال ۲.۲.۴. مسیر نشان داده شده در شکل ۱.۴ را در نظر بگیرید. زوج‌های نشان داده شده در شکل به ترتیب رأس و وزن آن را نشان می‌دهد. به دلیل وزن خیلی زیاد v_1 یکی از میانه‌ها در جواب بهینه باید در رأس v_1 قرار بگیرد. به دلیل وزن خیلی کم v_2 ، در جواب بهینه دومین میانه باید حداقل ۳ واحد از v_2 فاصله داشته باشد یعنی دومین میانه در نقطه m_2 و یا در سمت راست آن قرار بگیرد. نقطه m_2 در شکل نشان داده شده است. به محض این که دومین میانه در این شرط صدق کند، سهم رأس v_2 در تابع هدف بیشتر از این به مکان دومین میانه وابسته نیست. بنابراین جواب بهینه باید برای دومین میانه مکانی در سمت راست m_2 و یا منطبق بر آن انتخاب کند که با در نظر گرفتن v_3 و v_4 بهینه باشد. بدیهی است که در این جا تنها یک نقطه وجود دارد که همان نقطه m_2 نشان داده شده در شکل است.

۳.۴ مساله ۲- میانه با وزن مثبت- منفی روی درخت

مساله $\sum_1 |d(V, T)| \mid w_i \leq O \mid p \mid T$ را در نظر بگیرید. یک جواب شذنی شامل دو (نه لزوماً مجزا) نقطه m_1 و m_2 در T است. به دلیل یکنوایی^۴ تابع هدف F_1 ، همواره یک جواب بهینه با دو نقطه مجزای m_1 و m_2 وجود دارد. بنابراین بدون کاستن از کلیت، در مساله P_1 تنها جواب‌های شذنی شامل دو نقطه مجزا را بررسی می‌کنیم. نقطه میانی m_1 و m_2 را با m نشان می‌دهیم. فرض کنید (a, b) یالی در مسیر $P[m_1, m_2]$ باشد که شامل نقطه m است. بدون کاستن از کلیت فرض کنید رأس a در مسیری که m_1 و m را به هم متصل می‌کند قرار دارد. اما اگر m یک رأس باشد، بدون کاستن از کلیت فرض می‌کنیم بر a منطبق است.

فرض کنید T_a و T_b دو زیر درخت از T باشند که به ترتیب شامل a و b هستند و از حذف یال (a, b) به‌دست آمده‌اند. چون برای هر رأس $v_i \in T_a$ ، $d(m_1, v_i) \leq d(m_2, v_i)$ و برای هر رأس $v_j \in T_b$ ، $d(m_2, v_j) < d(m_1, v_j)$ است، می‌توان نتیجه گرفت تمام رئوس با وزن نامنفی در T_a و تمام رئوس با وزن منفی در T_b توسط m_1 سرویس‌دهی می‌شوند. رئوس باقی‌مانده نیز از m_2 سرویس می‌گیرند. (اگر $m = a$ فرض کنید که a و تمامی رئوس دیگر که در فاصله یکسانی از m_1 و m_2 قرار دارند، از m_1 سرویس می‌گیرند.)

بنابراین ما می‌توانیم مانند مساله ۲- میانه کلاسیک برای مساله $\sum_1 |d(V, T)| \mid w_i \leq O \mid p \mid T$ روش حذف یال را به‌کار ببریم. (مرجع [۳۹] را ببینید.) به ازای هر یال $e = (a, b)$ ، دو درخت T_e^1 و T_e^2 را می‌سازیم که تنها در وزن رئوس‌شان با درخت اصلی متفاوت هستند. وزن رئوس درخت T_e^1 را به صورت زیر تعریف می‌کنیم:

$$w_i^1 = \begin{cases} w_i & \text{if } ([v_i \in T_a, w_i \geq 0] \text{ or } [v_i \in T_b, w_i \leq 0]) \\ 0 & \text{if } ([v_i \in T_a, w_i < 0] \text{ or } [v_i \in T_b, w_i > 0]) \end{cases}$$

وزن رئوس درخت T_e^2 را نیز به شکل زیر تعریف می‌کنیم:

$$w_i^2 = \begin{cases} w_i & \text{if } ([v_i \in T_a, w_i \leq 0] \text{ or } [v_i \in T_b, w_i \geq 0]) \\ 0 & \text{if } ([v_i \in T_a, w_i > 0] \text{ or } [v_i \in T_b, w_i < 0]) \end{cases}$$

بنابراین با توجه به این که خاصیت بهینگی رأسی برای مساله $\sum_1 |d(V, T)| \mid w_i \leq O \mid p \mid T$ برقرار است می‌توان با استفاده از الگوریتم خطی بورکارد و کراروپ [۱۵] که در ادامه آورده ایم، ۱- میانه با وزن مثبت- منفی را برای هر درخت T_e^1 و T_e^2 به دست آورد. جواب بهینه این دو مساله یک جواب بهینه برای مساله $\sum_1 |d(V, T)| \mid w_i \leq O \mid p \mid T$ است. جواب بهینه مساله آخر را می‌توان از میان زوج جواب‌های بهینه مساله ۱- میانه با وزن مثبت- منفی به‌دست‌آمده از حذف بعضی یال‌های $(a, b) \in E$ که در بالا نیز توضیح داده شد، به‌دست‌آورد.

^۴ Monotonicity

مساله ۱- میانه با وزن مثبت- منفی روی درخت

در این بخش روش به دست آوردن جواب مساله ۱- میانه روی درخت با وزن مثبت- منفی که در [۱۵] آمده است را بیان می کنیم. برای این منظور به صورت زیر عمل می کنیم.

برای هر رأس T وزن آن $w(x)$ را نیز به آن نسبت می دهیم، و به جز ریشه درخت T ، طول $c(x, y)$ از یال (x, y) بین x و پدر آن y است.

تعریف ۱.۳.۴. رئوسی از درخت T که هنوز پردازش روی آن ها صورت نگرفته است را رئوس زنده می نامیم. در ابتدا ما راست ترین رأس زنده از بین رئوس زنده را که در بالاترین سطح است انتخاب می کنیم. دستورات پردازش به این صورت است که هر زمان یک رأس x انتخاب شد، ما در زیردرختی از T که در x ریشه دار شده است، زیرمجموعه $suc(x)$ از تمام رئوسی که در سطح بالاتر از x قرار دارند را پردازش می کنیم. برای هر رأس x در T قرار دهید

$$w^+(x) = \sum_{j \in suc(x)} w(j)$$

و

$$w^-(x) = -w(x) + \sum_{j \notin suc(x)} w(j).$$

یال $(x, y) \in T$ که y پدر x است را در نظر بگیرید. نماد $\Delta_T(x, y)$ را به صورت زیر تعریف می کنیم.

$$\Delta_T(x, y) = f(y) - f(x)$$

حال با توجه به این تعریف بررسی می کنیم که اگر ما از x به سمت پدرش y حرکت کنیم، آیا مقدار تابع هدف تغییر می کند؟

چون هر زوج (u, v) از رئوس در درخت T با یک مسیر واقعی متصل هستند، $d(u, v)$ فاصله کوتاه ترین مسیر بین u و v باید برابر با طول مسیر باشد. برای هر رأس k ، $k \in suc(x)$ ، $d(k, x) + c(x, y) = d(k, y)$ و همچنین اگر $k \notin suc(x)$ ، $d(k, x) - c(x, y) = d(k, y)$ سهم رأس k در $\Delta_T(x, y)$ دقیقاً برابر است با $w(k)c(x, k)$. مشابهاً، برای هر رأس k ، $k \notin suc(x) \cup \{x\}$ ، نقش آن در $\Delta_T(x, y)$ ، $-w(k)c(x, y)$ خواهد بود. در نهایت سهم خود رأس x ، $w(x)c(x, y)$ است. فرض کنید مجموعه $C(y) \setminus suc(x) \cup \{x\}$ را با نماد A نشان دهیم.

$$\begin{aligned}
f(y) &= \sum_{k \in C(y)} w(k)d(y, k) \\
&= \sum_{k \in \text{suc}(x)} w(k)d(y, k) + w(x)c(x, y) + \sum_{k \in A} w(k)d(y, k) \\
&= \sum_{k \in \text{suc}(x)} w(k)(d(k, x) + c(x, y)) + w(x)c(x, y) + \sum_{k \in A} w(k)(d(k, x) - c(x, y)) \\
&= \sum_{k \in \text{suc}(x)} w(k)d(k, x) + (w^+(x) + w(x) - w^-(x))c(x, y) + \sum_{k \in A} w(k)d(k, x)
\end{aligned}$$

و

$$f(x) = \sum_{k \in \text{suc}(x)} w(k)d(x, k) + \sum_{k \in A} w(k)d(x, k)$$

بنابراین

$$\Delta_T(x, y) = f(y) - f(x) = (w^+(x) + w(x) - w^-(x))c(x, y) \quad (۳.۴)$$

توجه کنید که W_T وزن کل درخت T و برای هر رأس x در درخت T برابر $w^+(x) + w(x) + w^-(x)$ است. رابطه (۳.۴) را می توان به شکل زیر نوشت:

$$\Delta_T(x, y) = \left(2(w^+(x) + w(x)) - W_T \right) c(x, y) \quad (۴.۴)$$

برای هر رأس x در T اطلاعات زیر را نیاز داریم. (قرار داد می کنیم که یک مجموع با مجموعه اندیس صفر برابر ۰ است.)

• $S(x)$: مجموعه فرزندان x . اگر x یک برگ باشد $S(x) = \emptyset$.

• $w(x)$: وزن رأس x .

• $w^+(x)$: وزن فرزندان x . برای $w^+(x)$ رابطه بازگشتی زیر را داریم:

$$w^+(x) = \sum_{s \in S(x)} (w^+(s) + w(s))$$

• $c(x, y)$: طول یال (x, y) از x به پدر آن که یکتا و تعیین شده است، به شرط این که x مخالف ریشه r باشد.

• $\Delta_T(x, y)$: تغییرات مقدار تابع هدف با حرکت از x به سوی پدرش y ، که آنرا در رابطه (۴.۴) تعریف کردیم و برابر است با

$$\Delta_T(x, y) = \left(2(w^+(x) + w(x)) - W_T \right) c(x, y).$$

- $\Delta_T^*(x)$: اختلاف بین مقدار تابع هدف y ، پدر x و یک رأس مانند $suc(x) \cup \{x\}$ با کمترین مقدار تابع هدف. اگر x تنها یک فرزند s داشته باشد، رابطه بازگشتی زیر را داریم:

$$\Delta_T^*(x) = \Delta_T(x, y) + \max(\circ, \Delta_T^*(s))$$

اگر x چند فرزند داشته باشد باید دقت کنیم، زیرا ما تنها یک رأس با کمترین مقدار می خواهیم. بنابراین داریم

$$\Delta_T^*(x) = \Delta_T(x, y) + \max_{s \in S(x)} (\circ, \Delta_T^*(s)). \quad (۵.۴)$$

اگر x ریشه r باشد، داریم:

$$\Delta_T^*(r) = \max_{s \in S(r)} (\circ, \Delta_T^*(s)) \quad (۶.۴)$$

- $\beta_T(x)$: نام رأسی با کمترین مقدار تابع هدف در مجموعه $suc(x) \cup \{x\}$. واضح است که برای هر برگ e داریم $\beta_T(e) = e$. در غیر این صورت رابطه بازگشتی زیر را داریم:

$$\beta_T(x) = \begin{cases} x, & \text{اگر } \Delta_T^*(s) \leq \circ \text{ برای تمام } s \in S(x) \\ \beta_T(s^*) & \text{اگر } \Delta_T^*(s^*) = \max_{s \in S(x)} \{\Delta_T^*(s) : \Delta_T^*(s) > \circ\} \end{cases} \quad (۷.۴)$$

- $\varphi(x)$: سهم زیردرخت ریشه دار شده در x در مقدار تابع هدف در ریشه درخت T .

$$\varphi(x) = \sum_{s \in S(x)} \varphi(s) + (w^+(x) + w(x))c(x, y)$$

که y پدر x است. اگر x ریشه باشد، داریم:

$$\varphi(x) = \sum_{s \in S(x)} \varphi(s)$$

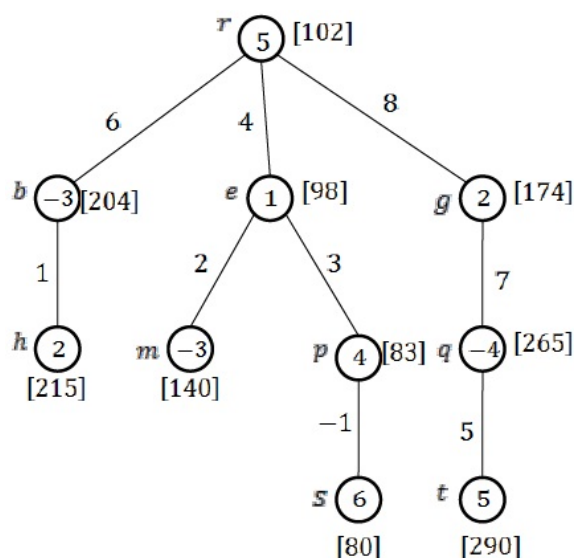
فرض کنید $f_T(x)$ مقدار تابع هدف درخت T باشد. برای ریشه r از درخت T داریم:

$$f_T(r) = \varphi(r).$$

مقدار بهینه تابع هدف در درخت T برابر است با

$$f_T^* = \varphi(r) - \Delta_T^*(r) \quad (۸.۴)$$

، که این مقدار بهینه در $\beta_T(r)$ به دست می آید.



شکل ۲.۴: درخت ریشه‌دار شده در r . عددهای داخل دایره وزن رئوس $w(x)$ ، و عددهای دیگر طول یال‌ها $c(x, y)$ را نشان می‌دهند. اعداد درون براکت نیز مقدار $f_T(x)$ را برای هر رأس x نشان می‌دهند.

به طور خلاصه، ما باید ابتدا یک دنباله از رئوس را که با برگ‌ها شروع می‌شود تعیین کنیم به گونه‌ای که فرزندان x قبل از x پردازش شوند. سپس وزن کل درخت T را محاسبه می‌کنیم. پس از آن با اولین رأس در دنباله شروع کرده و $w^+(x)$ ، $\Delta_T(x, y)$ ، $\Delta_T^*(x)$ ، $\beta_T(x)$ و $\varphi(x)$ را تا زمانی که به ریشه برسیم محاسبه می‌کنیم. در نهایت برای ریشه r مقدار تابع هدف را با استفاده از رابطه (۸.۴) محاسبه می‌کنیم.

مثال ۲.۳.۴. درخت نشان داده شده در شکل (۲.۴) را در نظر بگیرید. وزن کل درخت برابر ۱۵ است. با استفاده از الگوریتم فوق جدول زیر (۳.۴) را خواهیم داشت. جواب بهینه در رأس s با $f_T^* = \varphi(a) - 8$ به دست آمده است.

تعداد عملیات‌های (جمع، ضرب، مقایسه) مورد نیاز برای یافتن جواب بهینه برای درخت متناسب با تعداد رئوس درخت است، چون هر رأس را تنها یک بار ملاقات می‌کنیم و مقدار تخمینی برای بررسی هر رأس ثابت است، بنابراین روش در زمان خطی اجرا می‌شود.

حال فرض کنید $\{m_1, m_2\}$ جواب بهینه مساله $\sum_1 |T| \mid w_i \leq O \mid d(V, T) \mid$ باشد. در نتیجه m_1 جواب بهینه مساله $\sum_1 |T_e^1| \mid w_i \leq O \mid d(V, T_e^1) \mid$ و m_2 جواب بهینه مساله $\sum_1 |T_e^2| \mid w_i \leq O \mid d(V, T_e^2) \mid$ خواهد بود. این درست است زیرا در غیر این صورت ما برای مساله $\sum_1 |T| \mid w_i \leq O \mid d(V, T) \mid$ یک جواب با تابع هدف کمتر خواهیم داشت. در واقع، فرض کنید m'_1 و m'_2 به ترتیب جواب بهینه مساله‌های $\sum_1 |T_e^1| \mid w_i \leq O \mid d(V, T_e^1) \mid$ و $\sum_1 |T_e^2| \mid w_i \leq O \mid d(V, T_e^2) \mid$ باشند.

x	$S(x)$	$w(x)$	$w^+(x)$	$c(x, y)$	$\Delta_T(x, y)$	$\Delta_T^*(x)$	$\beta_T(x)$	$\varphi(x)$
t	—	۵	۰	۵	-۲۵	-۲۵	t	۲۵
s	—	۶	۰	-۱	۳	۳	s	-۶
m	—	-۳	۰	۲	-۴۲	-۴۲	m	-۶
h	—	۲	۰	۱	-۱۱	-۱۱	h	۲
q	t	-۴	۵	۷	-۹۱	-۹۱	q	۳۲
p	s	۴	۶	۳	۱۵	۱۸	s	۲۴
g	q	۲	۷	۸	-۷۲	-۷۲	g	۵۶
e	m, p	۱	۷	۴	۴	۲۲	s	۵۰
b	h	-۳	۲	۶	-۱۰۲	-۱۰۲	b	-۴
r	b, e, g	—	—	—	—	۲۲	s	۱۰۲

و فرض کنید f_1 و f_2 به ترتیب تابع هدف های این دو مساله باشند. بنابراین باید نامساوی زیر برقرار باشد:

$$F_1(m_1, m_2) = f_1(m_1) + f_2(m_2) \geq f_1(m'_1) + f_2(m'_2) = F_1(m'_1, m'_2)$$

در نتیجه جواب $\{m'_1, m'_2\}$ تابع هدف با مقدار کمتر از مقدار بهینه $F_1(m_1, m_2)$ دارد. بنابراین جواب بهینه مساله $\sum_1 |T| \mid w_i \leq O \mid d(V, T) \mid$ را می توانیم از میان زوج جواب های بهینه مساله ۱- میانه با وزن مثبت- منفی از روش حذف بعضی یال های $(a, b) \in E$ به دست آوریم. چون $n - 1$ امکان برای انتخاب یال حذفی داریم، الگوریتمی با زمان $O(n^2)$ به دست می آوریم. پس نتیجه زیر را خواهیم داشت.

قضیه ۳.۳.۴. [۱۱] مساله $\sum_1 |T| \mid w_i \leq O \mid d(V, T) \mid$ که T درختی با n رأس است، می تواند در زمان $O(n^2)$ پاسخ داده شود.

فرض کنید $f_e^1(v)$ مقدار تابع هدف مساله $\sum_1 |T_e^1| \mid w_i \leq O \mid d(V, T_e^1) \mid$ و $f_e^2(v)$ مقدار تابع هدف مساله $\sum_1 |T_e^2| \mid w_i \leq O \mid d(V, T_e^2) \mid$ در میانه v باشد، درخت های T_e^1 و T_e^2 از درخت T با روش حذف یال $e = (a, b)$ که توضیح داده شد به دست آمده اند. فرض کنید p_e^1 و p_e^2 به صورت زیر باشند:

$$p_e^1 \in \operatorname{argmin}_{v \in T_a} f_e^1(v)$$

$$p_e^2 \in \operatorname{argmin}_{v \in T_b} f_e^2(v)$$

با استفاده از قضیه (۳.۴) می توانیم نتیجه زیر را بیان کنیم.

لم ۴.۳.۴. [۱۱] زوج رأس (p_e^1, p_e^2) که

$$f_{e^*}^1(p_{e^*}^1) + f_{e^*}^2(p_{e^*}^2) = \min_{e \in N} (f_e^1(p_e^1) + f_e^2(p_e^2))$$

جواب بهینه مساله $\sum_1 |T| \mid w_i \leq O \mid d(V, T) \mid$ هستند.

در حالت‌های خاصی از درخت مانند مسیر، الگوریتم را می‌توان بهبود بخشید. از نظریه آئولتا و همکارانش [۵] برای سرعت بخشیدن به محاسبات ۲- میانه استفاده می‌شود. یعنی در هر گام از اطلاعات به‌دست‌آمده در گام‌های قبلی استفاده می‌شود.

الگوریتم خطی برای مساله روی یک مسیر

فرض کنید p یک مسیر با n رأس $\{v_1, v_2, \dots, v_n\}$ باشد. همانند قبل به ازای هر یال (v_i, v_{i+1}) درخت‌های $T_{(v_i, v_{i+1})}^1$ و $T_{(v_i, v_{i+1})}^2$ را می‌سازیم. برای ساده‌تر شدن نمایش، درخت‌های $T_{(v_i, v_{i+1})}^1$ و $T_{(v_i, v_{i+1})}^2$ را به ترتیب با T_i^1 و T_i^2 نشان می‌دهیم. نماد $f_i^1(v)$ ($f_i^2(v)$) را برای تابع هدف مساله $\sum_1 |T_i^1| w_i \leq O |d(V, T_i^1)|$ ($\sum_2 |T_i^2| w_i \leq O |d(V, T_i^2)|$) با میانه مکان‌یابی شده در رأس v نشان می‌دهیم. g_i^1, g_i^2 و p_i^1, p_i^2 را به صورت زیر تعریف می‌کنیم:

$$g_i^1 = \min_{1 \leq j \leq i} f_i^1(v_j)$$

$$g_i^2 = \min_{i \leq j \leq n} f_i^2(v_j)$$

$$p_e^1 \in \operatorname{argmin}_{1 \leq j \leq i} f_e^1(v)$$

$$p_e^2 \in \operatorname{argmin}_{i \leq j \leq n} f_e^2(v)$$

توجه کنید که برای تمام رأس‌های v_j از T_i^1 با وزن مثبت $i \leq j$ است. به‌طورمشابه، برای تمام رأس‌های v_j از T_i^2 با وزن مثبت $j > i$ است. طبق لم (۴.۳.۴) مقدار بهینه مساله $\sum_1 |P| w_i \leq O |d(V, P)|$ به فرم زیر خواهد بود:

$$g_{i_o}^1 + g_{i_o}^2 = \min_{1 \leq i \leq n-1} (g_i^1 + g_i^2) \quad (9.4)$$

و $\{p_{e_o}^1, p_{e_o}^2\}$ جواب بهینه هستند. حال با استفاده از ساختاری ویژه مینیمم (۹.۴) را در زمان خطی محاسبه می‌کنیم. نمادهای زیر را تعریف می‌کنیم:

$$WL^+(v_i) = \sum_{\{j | 1 \leq j \leq i, w_j > 0\}} w_j,$$

$$WL^-(v_i) = \sum_{\{j | 1 \leq j < i, w_j < 0\}} w_j,$$

$$WR^+(v_i) = \sum_{\{j | i \leq j \leq n, w_j > 0\}} w_j,$$

$$WL^-(v_i) = \sum_{\{j | i < j \leq n, w_j < 0\}} w_j.$$

وزن درخت‌های T_i^1 و T_i^2 ، $1 \leq i \leq n-1$ به شکل زیر است:

$$W(T_i^1) = WL^+(v_i) + WR^-(v_i),$$

$$W(T_i^2) = WL^+(v_{i+1}) + WR^+(v_{i+1})$$

حال می‌توان قضایای زیر را برای تابع‌های f_i^1 و f_i^2 نتیجه گرفت. ما تنها تابع f_i^1 را صریحا بررسی می‌کنیم. نتایج به‌دست‌آمده برای f_i^2 نیز برقرار است.

همان‌گونه که بورکارد و کراووپ نیز بیان کردند [۱۵]، برای هر دو رأس مجاور v_j و v_{j+1} با $1 \leq j+1 \leq j$ در درخت T_i^1 داریم:

$$\begin{aligned} \Delta f_i^1(v_j) &= f_i^1(v_{j+1}) - f_i^1(v_j) \\ &= \left(\sum_{k=1}^j w_k^1 - \sum_{k=j+1}^n w_k^1 \right) c(v_j, v_{j+1}) \\ &= (2WL^+(v_j) - W(T_i^1))c(v_j, v_{j+1}) \end{aligned} \quad (10.4)$$

حال به‌سادگی می‌توان دید که تابع f_i^1 (f_i^2) تعدادی خاصیت یکنوایی خوب دارد که در زیر بیان شده است.

قضیه ۵.۳.۴. [۱۱] اگر زوج (i_0, j_0) وجود داشته باشد به‌قسمی که $j_0 < i_0$ و $\Delta f_{i_0}^1(v_{j_0}) \geq 0$ ، آن‌گاه برای هر j که $j_0 \leq j < i_0$ ، داریم $\Delta f_{i_0}^1(v_j) \geq 0$. به عبارت دیگر، به ازای $1 \leq j \leq i_0$ حداکثر یک مینیمم محلی دارد.

برهان. چون $0 \leq c(v_j, v_{j+1})$ و $W(T_i^1(v_i))$ ثابت است، همچنین برای هر j که $j_0 \leq j < i_0$ داریم $WL + (v_j) > WL + (v_{j_0})$ بنابراین

$$\Delta_{i_0}^1(v_j) = [2WL^+(v_j) - W(T_{i_0}^1(v_{i_0}))]c(v_j, v_{j+1}) \geq 0$$

□

قضیه ۶.۳.۴. [۱۱] اگر زوج (i_0, j_0) وجود داشته باشد به‌قسمی که $j_0 < i_0$ و $\Delta f_{i_0}^1(v_{j_0}) \leq 0$ ، آن‌گاه برای هر i که $i_0 < i$ ، داریم $\Delta f_i^1(v_{j_0}) \leq 0$. به عبارت دیگر، برای هر $i_0 \leq i$ مینیمم $f_i^1(v_j)$ برای $1 \leq j < i$ سمت راست مینیمم $f_{i_0}^1(v_j)$ قرار دارد.

برهان. در این حالت چون j_0 ثابت و i متغیر است، در نتیجه $WL^+(v_{j_0})$ ثابت است. واضح است که برای هر i به‌قسمی که $i_0 < i$ می‌توان نوشت: $W(T_i^1(v_i)) \geq W(T_{i_0}^1(v_{i_0}))$ بنابراین

$$\Delta f_i^1(v_{j_0}) = [2WL^+(v_{j_0}) - W(T_i^1(v_i))]c(v_{j_0}, v_{j_0+1}) \leq 0$$

□

برپایه لم (۴.۳.۴) و قضیه‌های (۵.۳.۴) و (۶.۳.۴) می‌توانیم الگوریتم زیر را نتیجه بگیریم:

*

الگوریتم

(۱) برای هر رأس v در درخت T ، $WL^+(v)$ ، $WL^-(v)$ ، $WR^+(v)$ و $WR^-(v)$ را محاسبه کنید. (قدم به قدم از چپ به راست و از راست به چپ حرکت کنید.) برای هر i ، $2 \leq i \leq n$ ، $d(v_1, v_i)$ و برای هر i ، $1 \leq i \leq n-1$ ، $d(v_n, v_i)$ را محاسبه کنید. $f_1^1(v_1)$ و $f_{n-1}^2(v_n)$ را محاسبه کنید.

(۲) برای هر i ، $1 \leq i \leq n-1$ ، با حرکت از چپ به راست و اجرای زیر برنامه $Left-to-Right$ ، g_i^1 را محاسبه و p_i^1 را بیابید.

(۳) برای هر i ، $1 \leq i \leq n-1$ ، با حرکت از راست به چپ و اجرای زیر برنامه $Right-to-Left$ ، g_i^2 را محاسبه و p_i^2 را بیابید.

$$(4) \quad g_{i_o}^1 + g_{i_o}^2 = \min_{1 \leq i \leq n-1} (g_i^1 + g_i^2) \quad \text{را محاسبه کنید.}$$

(۵) خروجی: $\{p_{i_o}^1, p_{i_o}^2\}$ یک جواب بهینه مساله $\sum_1 |P| w_i \leq O |d(V, P)|$ است.

در ادامه زیر برنامه $Left-to-Right$ را معرفی می شود. زیر برنامه $Right-to-Left$ نیز به طور مشابه مقدار g_i^2 محاسبه کرده و نقطه p_i^2 را تعیین می کند. این زیر برنامه نیز مقدار f_i^2 را می گیرد و اندیس i را در حلقه for کاهش می دهد. طبق رابطه (۱۰.۴) از نماد زیر استفاده می شود،

$$\Delta f_i^1(v_k) = f_i^1(v_{k+1}) - f_i^1(v_k)$$

*

زیر برنامه $Left-to-Right$

(۱) قرار دهید $k = 1$ ، $g_1^1 = f_1^1(v_1)$ ، $p_1^1 = v_1$ ، $i = 2$.

(۲) قرار دهید $|w_i| d(v_k, v_i) + f_{i-1}^1(v_k) = f_i^1(v_k)$ و به گام (۳) بروید.

(۳) اگر $\Delta f_i^1(v_k) \leq 0$ و $k < i$ ، به گام (۴) بروید در غیر این صورت به گام (۶) بروید.

$$(4) \quad f_i^1(v_{k+1}) = f_i^1(v_k) + \Delta f_i^1(v_k)$$

(۵) $k = k + 1$ و به گام (۳) بروید.

(۶) قرار دهید $p_i^1 = v_k$ و $g_i^1 = f_i^1(v_k)$.

(۷) قرار دهید $i = i + 1$.

(۸) اگر $i \leq n - 1$ به گام (۲) بروید در غیراین صورت به گام (۹) بروید.

(۹) پایان.

برای جمع‌بندی مطالب فوق قضیه زیر را داریم:

قضیه ۷.۳.۴. [۱۱] برای یک مسیر P الگوریتم فوق مساله $\sum_i |w_i| \leq O(d(V, P))$ را بادر نظر گرفتن تعداد رئوس در زمان خطی پاسخ می‌دهد.

برهان. با توجه به لم ۲ و قضیه‌های (۵.۳.۴) و (۶.۳.۴) می‌توان نتیجه گرفت که الگوریتم به‌درستی عمل می‌کند. برای این که پیچیدگی زمانی آن را اثبات کنیم کافی است نشان دهیم که زیربرنامه *Left-to-Right* (*Right-to-Left*) در زمان خطی اجرا می‌شود. فرض کنید گام i اجرا شده و P_{i-1}' را رأسی قرار دهید که تابع f_{i-1}' را مینیمم می‌کند. طبق قضیه (۶.۳.۴)، مینیمم f_i' در مسیر اتصال P_{i-1}' و v_i قرار دارد. طبق قضیه (۵.۳.۴) این مینیمم اولین رأس v_k در مسیر اتصال P_{i-1}' و v_i است که $\Delta f_i'(v_k) > 0$. بنابراین در گام i -ام ما $f_i'(v_s)$ (یا به طور معادل $\Delta f_i'(v_s)$) را برای تمام رئوس v_s میان P_{i-1}' و v_k محاسبه می‌کنیم. مجموعه رئوس شامل در این محاسبات در گام‌ها مختلف الگوریتم مجزا هستند. بنابراین روی هم رفته میزان محاسبات برای تمام گام‌ها بیشتر از محاسبات Δ و مقایسه زوج رأس‌ها است. در پیش‌پردازش محاسبات هر Δ زمان ثابت نیاز دارد و این اثبات را کامل می‌کند. $\square$

فصل ۵

مساله پشتيان ۲-ميانه نيمه ناخوشايند روى
درخت

۱.۵ مقدمه

با توجه به مساله پشتیبان ۲- میانه روی درخت که در فصل (۲) بیان کردیم و همچنین مساله ۲- میانه با وزن مثبت- منفی که در فصل (۴) مطرح کردیم، برآن شدیم تا با ترکیب این دو مساله، مساله پشتیبان ۲- میانه نیمه ناخوشایند با مدل MWD را تعریف کنیم. در این فصل با استفاده از الگوریتمی که برای یافتن ۲- میانه در پیوندهای گراف کاکتوس که این پیوندها نیز ساختار درختی دارند، روش حل را برای مساله پشتیبان ۲- میانه نیمه ناخوشایند تعمیم داده و الگوریتمی با زمان $O(n^2)$ برای مساله ارایه می‌دهیم که n تعداد رئوس درخت مورد بررسی است. در بخش پایانی فصل نیز مساله پشتیبان ۲- میانه نیمه ناخوشایند روی مسیر را مطرح کرده و با استفاده از الگوریتم بیان شده در [۱۱]، توانستیم مساله را در زمان خطی پاسخ دهیم.

۲.۵ تعریف مساله

فرض کنید $T = (V, E)$ یک درخت با n رأس باشد. هر رأس v_i در درخت T وزن دلخواه w_i دارد. فرض کنید $d(x, y)$ و $P[x, y]$ به ترتیب کوتاهترین مسیر بین دو رأس x و y و فاصله آن‌ها باشد. به ازای هر یال T_v فرض کنید T_v زیردرختی از T باشد که از حذف یال $e(v, u) \in T$ ایجاد شده به قسمی که $v \in T_v$ و $u \in T_u = T \setminus T_v$. همچنین برای هر زیردرخت $T' \subseteq T$ تعریف می‌کنیم: $W(T') = \sum_{v_i \in T'} w_i$.

فرض کنید ρ_1 و ρ_2 احتمال شکست دو سرویس دهنده در مساله پشتیبان ۲- میانه نیمه ناخوشایند باشد. در این فصل نیز برای ساده تر شدن نمایش فرض می‌کنیم $\rho_1 = \rho_2 = \rho$. در مساله پشتیبان ۲- میانه نیمه ناخوشایند، هدف مینیم کردن مجموع فاصله مورد انتظار از تمام رئوس به مجموعه سرویس دهنده‌های در حال کار است یعنی ما می‌خواهیم تابع هدف زیر را مینیم کنیم:

$$F(v_1, v_2) = (1 - \rho) \left[\sum_{v \in V} \min_{1 \leq i \leq 2} w(v) d(v, v_i) \right] + \rho \left[\sum_{v \in V} w(v) d(v, v_1) + \sum_{v \in V} w(v) d(v, v_2) \right]$$

فرض کنید v_1 و v_2 پاسخ مساله مورد بحث باشند. نقطه‌ای از T را با m نشان دهید که $m \in P(v_1, v_2)$ و $d(v_1, m) = d(v_2, m)$. فرض کنید $e(x, y)$ یالی در مسیر $P(v_1, v_2)$ باشد که شامل نقطه m است. بدون از دست دادن کلیت فرض کنید رأس x روی مسیری که v_1 را به m متصل می‌کند قرار دارد. علاوه بر این اگر m یک رأس باشد، ما فرض می‌کنیم که منطبق بر x است. چون $d(v_1, v_i) \leq d(v_2, v_i)$ و $d(v_1, v_j) > d(v_2, v_j)$ برای هر رأس $v_i \in T_X$ و $v_j \in T_Y$ ، می‌توان نتیجه گرفت که رئوس با وزن نامنفی در T_X و رئوس با وزن منفی در T_Y توسط v_1 سرویس‌دهی می‌شوند. رئوس باقی‌مانده از v_2 سرویس می‌گیرند. (اگر $m = x$ فرض کنید که x و تمامی رئوس دیگر که در فاصله یکسانی از m_1 و m_2 قرار دارند، از m_1 سرویس می‌گیرند.)

بنابراین ما مجموعه رئوس V را به دو مجموعه رأسی V_1 و V_2 افراز می کنیم که

$$V_1 = \{v | d(v, x) \leq d(v, y), w(v) \geq \circ\} \cup \{v | d(v, x) \geq d(v, y), w(v) \leq \circ\},$$

و

$$V_2 = V \setminus V_1$$

و $e(x, y)$ یالی است که شامل نقطه میانی مسیر بین v_1 و v_2 است.

فرض کنید

$$D(V, u) = \sum_{v \in V} w(v) d(v, u)$$

در نتیجه تابع هدف مساله پشتیبان ۲- میانه نیمه ناخوشایند را می توان به فرم زیر بازنویسی کرد:

$$\begin{aligned} F(v_1, v_2) &= (1 - \rho)[D(V_1, v_1) + D(V_2, v_2)] \\ &+ \rho[D(V_1, v_1) + D(V_2, v_1) + D(V_1, v_2) + D(V_2, v_2)] \\ &= D(V_1, v_1) + \rho D(V_2, v_1) + \rho D(V_1, v_2) + D(V_2, v_2). \end{aligned}$$

۳.۵ چند خاصیت روی درخت ها

لم ۱.۳.۵. برای یک درخت T در مساله پشتیبان ۲- میانه نیمه ناخوشایند یک جواب بهینه وجود دارد که تمام سرویس دهنده ها را روی رئوس درخت T مکان یابی می کند.

برهان. به برهان خلف فرض کنید که هیچ جوابی وجود ندارد که تمام سرویس دهنده ها را روی رئوس T مکان یابی کند. فرض کنید (m_1, m_2) یک جواب بهینه مساله روی درخت T باشند که m_1 نقطه غیر رأسی روی درخت است. فرض کنید $e(v_r, v_q)$ یال شامل m_1 باشد. فرض کنید V_1 زیرمجموعه ای از رئوس T باشد که توسط m_1 سرویس دهی می شوند. تابع هدف زیر که وابسته به m_1 است را در نظر بگیرید:

$$F(m_1, m_2) = D(V_1, m_1) + \rho D(V_2, m_1) + \rho D(V_1, m_2) + D(V_2, m_2)$$

که $V_2 = V \setminus V_1$. در نتیجه داریم:

$$\begin{aligned} D(V_1, m_1) + \rho D(V_2, m_1) &= \sum_{v \in V_1 \cap T_{v_r}} w(v)(d(v, v_r) + d(v_r, m_1)) + \sum_{v \in V_1 \setminus T_{v_r}} w(v)(d(v, v_r) - d(v_r, m_1)) \\ &+ \rho \left[\sum_{v \in V_2 \cap T_{v_r}} w(v)(d(v, v_r) + d(v_r, m_1)) + \sum_{v \in V_2 \setminus T_{v_r}} w(v)(d(v, v_r) - d(v_r, m_1)) \right] \\ &= \sum_{v \in V_1 \cap T_{v_r}} w(v) d(v, v_r) + \sum_{v \in V_1 \setminus T_{v_r}} w(v) d(v, v_r) \end{aligned}$$

$$\begin{aligned}
 & + \rho \left[\sum_{v \in V_1 \cap T_{v_r}} w(v) d(v, v_r) + \sum_{v \in V_2 \setminus T_{v_r}} w(v) d(v, v_r) \right] \\
 & + \left(\sum_{v \in V_1 \cap T_{v_r}} w(v) - \sum_{v \in V_1 \setminus T_{v_r}} w(v) + \rho \left[\sum_{v \in V_1 \cap T_{v_r}} w(v) - \sum_{v \in V_2 \setminus T_{v_r}} w(v) \right] \right) d(v_r, m_1) \\
 & = D(V_1, v_r) + \rho D(V_2, v_r) \\
 & + \left[2 \left(\sum_{v \in V_1 \cap T_{v_r}} w(v) + \rho \sum_{v \in V_2 \cap T_{v_r}} w(v) \right) - w(T_{V_1}) - \rho w(T(V_2)) \right] d(v_r, m_1).
 \end{aligned}$$

قرار دهید

$$K = \left[2 \left(\sum_{v \in V_1 \cap T_{v_r}} w(v) + \rho \sum_{v \in V_2 \cap T_{v_r}} w(v) \right) - w(T_{V_1}) - \rho w(T(V_2)) \right]$$

بنابراین

$$D(V_1, m_1) + \rho D(V_2, m_1) = D(V_1, v_r) + \rho D(V_2, v_r) + K d(v_r, m_1)$$

حال اگر $K \geq 0$ چون برای هر نقطه x روی قطعه یال $P(v_r, m_1)$ داریم $d(v_r, x) < d(v_r, m_1)$ ، بنابراین $F(v_i, m_2) < F(m_1, m_2)$. که این تناقض است.

اگر $K < 0$ چون برای هر نقطه y روی قطعه یال $P(m_1, v_q)$ داریم $d(v_r, y) > d(v_r, m_1)$ ، پس می‌توان نتیجه گرفت $F(v_q, m_2) < F(m_1, m_2)$ که v_q و m_2 هر دو رأس هستند که این نیز تناقض است. $\square$

بر پایه لم (۱.۳.۵)، ما می‌توانیم از روش حذف یال که در مساله ۲- میانه با وزن مثبت منفی نیز به کار رفته‌است، استفاده کنیم. به ازای هر یال $e = (x, y)$ دو درخت T_e^1 و T_e^2 را می‌سازیم که تنها در وزن رئوسشان با درخت T متفاوت هستند. وزن رئوس در درخت T_e^1 را به صورت زیر تعریف می‌کنیم:

$$w_i^1 = \begin{cases} w_i & \text{if } ([v_i \in T_x, w_i \geq 0] \text{ or } [v_i \in T_y, w_i \leq 0]) \\ 0 & \text{if } ([v_i \in T_x, w_i < 0] \text{ or } [v_i \in T_y, w_i > 0]) \end{cases}$$

همچنین وزن رئوس در درخت T_e به صورت زیر خواهد بود:

$$w_i^2 = \begin{cases} w_i & \text{if } ([v_i \in T_x, w_i \leq 0] \text{ or } [v_i \in T_y, w_i \geq 0]) \\ 0 & \text{if } ([v_i \in T_x, w_i > 0] \text{ or } [v_i \in T_y, w_i < 0]) \end{cases}$$

فرض کنید $T_e^1(x)$ و $T_e^2(y)$ دو درخت باشند که به ترتیب با T_e^1 و T_e^2 مشابه هستند با این تفاوت که وزن x در $T_e^1(x)$ را برابر با $w(x) + \rho w(T_e^2)$ و وزن y در درخت $T_e^2(y)$ را برابر با $w(y) + \rho w(T_e^1)$ باشد.

لم ۲.۳.۵. فرض کنید (m_1, m_2) جواب بهینه مساله پشتیبان ۲- میانه نیمه ناخوشایند روی درخت T باشد. پس m_1 و m_2 به ترتیب جواب مساله ۱- میانه نیمه ناخوشایند روی درخت‌های $T_e^1(x)$ و $T_e^2(y)$ هستند که $e = (x, y)$ یال شامل نقطه میانی m_1 و m_2 است.

برهان. ثابت می کنیم m_1 میانه $T_e^1(x)$ است. اثبات برای m_2 نیز مشابه است. به برهان خلف فرض کنید که m_1 میانه $T_e^1(x)$ نباشد. فرض کنید s میانه $T_e^1(x)$ باشد. بنابر تعریف میانه داریم:

$$\begin{aligned} \sum_{v \in V_1} w(v)d(v, s) &= \sum_{v \in V_1, v \neq x} w(v)d(v, s) + (w^1(x) + \rho w(T_e^2))d(x, s) \\ &= D(V_1, s) + \rho w(T_e^2)d(x, s) < D(V_1, m_1) + \rho w(T_e^2)d(x, m_1). \end{aligned} \quad (۱.۵)$$

در نظر بگیرید اگر $v \in V_2$ و $w(v) \geq 0$ آن گاه $d(v, s) = d(v, x) + d(x, s)$ و همچنین اگر $v \in V_2$ و $w(v) < 0$ آن گاه $d(v, s) = d(v, x) - d(x, s)$. در نتیجه

$$\begin{aligned} D(V_2, s) &= \sum_{v \in V_2} w(v)d(v, s) \\ &= \sum_{v \in V_2, w(v) \geq 0} w(v)(d(v, x) + d(x, s)) + \sum_{v \in V_2, w(v) < 0} w(v)(d(v, x) - d(x, s)) \\ &= \sum_{v \in V_2, w(v) \geq 0} w(v)d(v, x) + \sum_{v \in V_2, w(v) < 0} w(v)d(v, x) \\ &\quad + \left[\sum_{v \in V_2, w(v) \geq 0} w(v) - \sum_{v \in V_2, w(v) < 0} w(v) \right] d(x, s) \\ &\quad + \sum_{v \in V_2, w(v) < 0} w(v)d(x, s) - \sum_{v \in V_2, w(v) < 0} w(v)d(x, s) \\ &= \sum_{v \in V_2} w(v)d(v, x) + [w(T_e^2) - 2 \sum_{v \in V_2, w(v) < 0} w(v)]d(x, s) \\ &= D(V_2, x) + [w(T_e^2) - 2 \sum_{v \in V_2, w(v) < 0} w(v)]d(x, s). \end{aligned} \quad (۲.۵)$$

حال با استفاده از (۱.۵) و (۲.۵) داریم:

$$\begin{aligned} &D(V_1, s) + \rho D(V_2, s) \\ &= D(V_1, s) + \rho w(T_e^2)d(x, s) + \rho D(V_2, x) - 2\rho \sum_{v \in V_2, w(v) < 0} w(v)]d(x, s) \\ &< D(V_1, m_1) + \rho w(T_e^2)d(x, m_1) + \rho D(V_2, x) - 2\rho \sum_{v \in V_2, w(v) < 0} w(v)]d(x, s) \\ &= D(V_1, m_1) + \rho D(V_2, m_1) \end{aligned}$$

بنابراین

$$\begin{aligned} F(m_1, m_2) &= D(V_1, m_1) + \rho D(V_2, m_1) + \rho D(V_1, m_2) + D(V_2, m_2) \\ &> D(V_1, s) + \rho D(V_2, s) + \rho D(V_1, m_2) + D(V_2, m_2) \geq F(s, m_2) \end{aligned}$$

□

که تناقض است.

بنابراین جواب بهینه مساله پشتیبان ۲-میانه نیمه ناخوشایند روی درخت T را می‌توان از میان زوج جواب‌های بهینه مساله ۱-میانه با وزن مثبت-منفی که از حذف بعضی یال‌های $e(x, y) \in T$ به دست می‌آیند، پیدا نمود. چون $n - 1$ امکان برای انتخاب یال حذفی وجود دارد بنابراین یک الگوریتم با زمان $O(n^2)$ خواهیم داشت.

قضیه ۳.۳.۵. مساله پشتیبان ۲-میانه نیمه ناخوشایند روی درخت T ، می‌تواند در زمان $O(n^2)$ پاسخ داده‌شود.

برهان. در مساله پشتیبان ۲-میانه نیمه ناخوشایند روی مسیر، از آنجایی که مساله ۱-میانه با وزن مثبت-منفی روی مسیر پیچیدگی زمانی خطی دارد، می‌توان نتیجه گرفت با الگوریتم ارایه شده مساله در زمان خطی پاسخ داده می‌شود. $\square$

۴.۵ مساله پشتیبان ۲-میانه نیمه ناخوشایند روی مسیر

مساله پشتیبان ۲-میانه نیمه ناخوشایند را روی یک مسیر در نظر بگیرید. فرض کنید P یک مسیر با n رأس $v_1, v_2, \dots, v_n$ باشد. درخت‌های به‌دست‌آمده از حذف یال $e_i = (v_i, v_{i+1})$ مشابه آن‌چه در قسمت قبل گفته شد، با T_i^1 و T_i^2 نشان می‌دهیم. همچنین تابع هدف‌های مساله ۱-میانه روی درخت‌های T_i^1 و T_i^2 را به ترتیب $f_i^1(v)$ و $f_i^2(v)$ در نظر بگیرید. مشابه آن‌چه در مرجع [۱۱] گفته شده، نمادگذاری‌های زیر را تعریف می‌کنیم:

$$WL^+(v_i) = \sum_{\{j | 1 \leq j \leq i, w_j > 0\}} w_j,$$

$$WL^-(v_i) = \sum_{\{j | 1 \leq j < i, w_j < 0\}} w_j,$$

$$WR^+(v_i) = \sum_{\{j | i \leq j \leq n, w_j > 0\}} w_j,$$

$$WR^-(v_i) = \sum_{\{j | i < j \leq n, w_j < 0\}} w_j.$$

فرض کنید دو درخت $T_i^1(v_i)$ و $T_i^2(v_{i+1})$ به ترتیب مشابه T_i^1 و T_i^2 باشند با این تفاوت که وزن v_i در $T_i^1(v_i)$ برابر با $w(v_i) + \rho w(T_i^2)$ و وزن v_{i+1} در $T_i^2(v_{i+1})$ برابر با $w(v_{i+1}) + \rho w(T_i^1)$ است. وزن درخت‌های T_i^1 و T_i^2 ، $1 \leq i \leq n - 1$ به صورت زیر خواهد بود:

$$W(T_i^1) = WL^+(v_i) + WR^-(v_i),$$

$$W(T_i^2) = WL^+(v_{i+1}) + WR^+(v_{i+1})$$

و

$$W(T_i^1(v_i)) = WL^+(v_i) + WR^-(v_i) + \rho w(T_i^2),$$

$$W(T_i^{\vee}) = WL^+(v_{i+1}) + WR^+(v_{i+1})\rho w(T_i^{\vee})$$

همچنین

$$\begin{aligned}\Delta f_i^{\vee}(v_j) &= f_i^{\vee}(v_{j+1}) - f_i^{\vee}(v_j) \\ &= [2WL^+(v_j) - \rho W(T_i^{\vee}) - W(T_i^{\vee})]F(v_j, v_{j+1}) \\ &= [2WL^+(v_j) - W(T_i^{\vee}(v_i))]c(v_j, v_{j+1})\end{aligned}$$

حال قضایای زیر که در فصل (۴) برای مساله ۲- میانه با وزن مثبت- منفی بیان شد، در این جا نیز برقرار است.

قضیه ۱.۴.۵. اگر زوج (i_0, j_0) وجود داشته باشد به قسمی که $j_0 < i_0$ و $\Delta f_{i_0}^{\vee}(v_{j_0}) \geq 0$ ، آن گاه برای هر j که $j_0 \leq j < i_0$ ، داریم $\Delta f_{i_0}^{\vee}(v_j) \geq 0$. به عبارات دیگر، $f_{i_0}^{\vee}(v_j)$ به ازای $1 \leq j \leq i_0$ حداکثر یک مینیمم محلی دارد.

برهان. چون $c(v_j, v_{j+1}) \geq 0$ و $W(T_i^{\vee}(v_i))$ ثابت است، همچنین برای هر j که $j_0 \leq j < i_0$ داریم $WL + (v_j) > WL + (v_{j_0})$ بنابراین

$$\Delta_{i_0}^{\vee}(v_j) = [2WL^+(v_j) - W(T_{i_0}^{\vee}(v_{i_0}))]c(v_j, v_{j+1}) \geq 0$$

□

قضیه ۲.۴.۵. اگر زوج (i_0, j_0) وجود داشته باشد به قسمی که $j_0 < i_0$ و $\Delta f_{i_0}^{\vee}(v_{j_0}) \leq 0$ ، آن گاه برای هر i که $i_0 < i$ ، داریم $\Delta f_i^{\vee}(v_{j_0}) \leq 0$. به عبارات دیگر، برای هر i که $i_0 \leq i$ مینیمم $f_i^{\vee}(v_j)$ ، $1 \leq j < i$ سمت راست مینیمم $f_{i_0}^{\vee}(v_j)$ ، $1 \leq j < i_0$ قرار دارد.

برهان. در این حالت چون j_0 ثابت و i متغیر است، در نتیجه $WL^+(v_{j_0})$ ثابت است. واضح است که برای هر i به قسمی که $i_0 < i$ می توان نوشت: $W(T_i^{\vee}(v_i)) \geq W(T_{i_0}^{\vee}(v_{i_0}))$ بنابراین

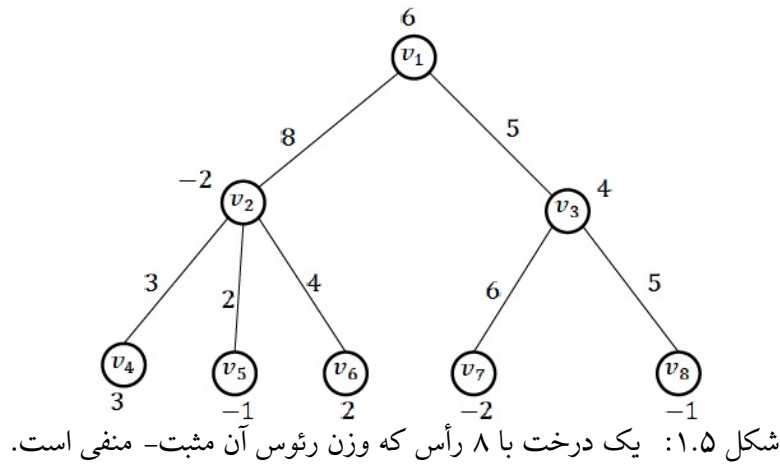
$$\Delta f_i^{\vee}(v_{j_0}) = [2WL^+(v_{j_0}) - W(T_i^{\vee}(v_i))]c(v_{j_0}, v_{j_0+1}) \leq 0$$

□

دوباره مانند فصل (۴) برای هر i ، $1 \leq i \leq n$ ، ما می توانیم دو زیربرنامه $Left_to_Right$ و $Right_to_left$ را برای یافتن میانه های $m_i^{\vee}$ و $m_i^{\wedge}$ تابع هدف آن ها به ترتیب روی درخت های $T_i^{\vee}$ و $T_i^{\wedge}$ بیابیم. بنابراین

$$\begin{aligned}F(m_i^{\vee}, m_i^{\wedge}) &= f_i^{\vee}(m_i^{\vee}) + f_i^{\wedge}(m_i^{\wedge}) \\ &+ \rho \left[\sum_{k=1}^n w_k^{\vee} d(v_k, m_i^{\vee}) + \sum_{k=1}^n w_k^{\wedge} d(v_k, m_i^{\wedge}) \right],\end{aligned}$$

و جواب بهینه را می توان از الگوریتم زیر به دست آورد.



الگوریتم $Semi - obnoxiousBackup_2 - med$

ورودی: یک مسیر p با وزن مثبت-منفی و احتمال شکست $0 \leq \rho \leq 1$ برای هر سرویس دهنده.

خروجی: پشتیبان ۲-میانه روی مسیر ورودی.

۱. قرار دهید $i = 1, opt = \infty$.

۲. زیر برنامه $Left - to - Right$ را اجرا کرده و جواب را در آرایه MP_1 ذخیره کن.

۳. زیر برنامه $Right - to - Left$ را اجرا کرده و جواب را در آرایه MP_2 ذخیره کن.

۴. اگر $F(MP_1[i], MP_2[i]) < opt$ آن گاه به گام (۵) برو در غیر این صورت به گام (۹) برو.

۵. $opt = F(MP_1[i], MP_2[i])$.

۶. $i^* = i$.

۷. $i = i + 1$.

۸. اگر $i \leq n - 1$ آن گاه به گام (۴) برو در غیر این صورت به گام (۹) برو.

۹. $(MP_1[i^*], MP_2[i^*])$ پشتیبان ۲-میانه مساله روی مسیر ورودی است.

۱۰. پایان.

قضیه ۳.۴.۵. مساله پشتیبان ۲-میانه نیمه ناخوشایند روی مسیر، می تواند در زمان خطی پاسخ داده شود.

یال حذف شده	m_i^1	m_i^2	$W(T_i^1)$	$W(T_i^2)$	f_i^1	f_i^2	$F(m_i^1, m_i^2)$
(v_3, v_8)	v_1	v_8	۱۴	۵-	۶۴.۵	۷۸-	۴.۶۵
(v_3, v_7)	v_1	v_7	۱۳	-۵	۵۲.۵	-۸۱	-۸/۷۱
(v_2, v_6)	v_1	v_6	۱۳	-۴	۴۹.۸	-۸۲	-۱۹/۳
(v_2, v_5)	v_1	v_6	۱۴	-۵	۶۳	-۶۷/۶	۷.۳
(v_2, v_4)	v_1	v_4	۱۲	-۳	۴۱.۶	-۸۹/۶	-۳۴/۸۷
(v_1, v_3)	v_2	v_3	۸	۱	۹.۸	-۴۱	-۲۲/۶۶
(v_1, v_2)	v_3	v_4	۷	-۱۰	۲	-۴۸/۹	-۴۰/۲۸

جدول ۱.۵: برای حالت $\rho = ۰/۱$ تابع هدف و میانه‌های زیر درخت‌های به دست آمده از حذف یال درخت شکل (۱.۵).

ρ	(m^1, m^2)	$F(m^1, m^2)$
۰	(v_3, v_4)	-۶۲
۰.۱	(v_3, v_4)	-۴۰/۲۸
۰.۳	(v_1, v_4)	۷.۵۷
۰.۵	(v_1, v_4)	۴۶.۲۵
۰.۷	(v_1, v_2)	۴۸.۹۴
۰.۹	(v_1, v_2)	۱۲۴.۰۲

جدول ۲.۵: پشتیبان‌های ۲-میانه و مقدار تابع هدف آن‌ها به ازای مقادیر مختلف ρ روی درخت شکل (۱.۵).

۵.۵ مثال

درختی با ۸ رأس که در شکل ۱.۵ نشان داده شده است را در نظر بگیرید. اگر $\rho = ۰$ ، پشتیبان ۲-میانه (v_4, v_3) است، که جواب مساله ۲-میانه کلاسیک با وزن مثبت-منفی نیز است. اگر $\rho = ۰/۱$ ، پشتیبان ۲-میانه (v_6, v_1) است که از حذف یال (v_2, v_6) به دست آمده است. برای این مساله مقدار تابع هدف که از حذف بعضی یال‌ها به دست آمده در جدول ۱.۵ آورده شده است. جدول ۲.۵ شامل پشتیبان‌های ۲-میانه و مقادیر تابع هدف آن‌ها به ازای مقادیر مختلف ρ می‌باشد.

مراجع

- [۱] تقی زاده، فتحعلی، "مساله p -میان روی شبکه"، دانشگاه صنعتی شاهرود، مجموعه مقالات سی و چهارمین کنفرانس ریاضی ایران، (۱۳۸۲)، ۶۵-۷۴.
- [۲] سهامی، "آمایش و مکان یابی" چاپ اول، دانشگاه صنعتی مالک اشتر، پژوهشکده پدافند غیر عامل، (۱۳۸۶).
- [۳] جمالیان، "مکان یابی با جاذبه و دافعه"، دانشگاه صنعتی شاهرود، دانشکده ریاضی، (۱۳۸۸).
- [4] A. V. Aho, J. E. Hopcroft and J. D. Ullman, "Dynamic and static algorithms for optimal placement of resources in a tree", *Theoret. Comput. Sci.* 165 (1996) 441-461.
- [5] V. Auletta, D. Parente, G. Persiano, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, (1974).
- [6] I. Averbakh and O. Berman, "Minimax regret median location on a network under uncertainty", *INFORMS J comput* 12 (2003), 97- 103.
- [7] I. Averbakh and O. Berman, "An improved algorithm for the minimax regret problem on tree", *Networks* 41 (2000), 104- 110.
- [8] R. Benkoczi, "Cardinality constrained facility location problems in tree." Ph. D. Thesis, Simon Fraser University, (2004).
- [9] R. Benkoczi, D. Breton, B. Bhattacharya, "Efficient computation of 2-medians in a tree network with positive/negative weights." *Discrete Mathematics* 306 (2006), 1505-1516.
- [10] J. A. Bondy and U. S. R. Murty, *Graph Theory*, Springer, Reading, DOI: 10.1007/978-1-84628-970-5, (2007)
- [11] R.E. Burkard, E. Çela and H. Dollani, "2-Median in trees with pos/neg weights." *Discrete Appl. Math.*, 105 (2000), 51-71.
- [12] R.E. Burkard, J. Fathali, "A polynomial time method for the pos/neg weighted 3-median problem on a tree", *Math. Methods Oper. Res.*, 65:(2007) 229-238.
- [13] R.E. Burkard, J. Fathali, H. Taghizadeh, "The p -maxian problem on a tree", *Oper. Res. Lett.* 35:(2007) 331-335.
- [14] R.E. Burkard, J. Hatzl, "Median problems with positive and negative weights on cycles and cacti", *J. Comb. Optim.*, (2010) 20: 27-46.
- [15] R.E. Burkard, and J. Krarup, "A linear algorithm for the pos/neg-weighted 1-median problem on a cactus", *Computing*, 60: 193-215, (1998).
- [16] B. T. Chen and C. S. Lin, "Minimax-regret robust 1-median location on tree", *Networks* 31 (1998), 93-103.
- [17] M. L. Chen and R. L. Francis, and T. J. Lowe, "The 1-center problem: Exploiting block structure", *Trans Sci.* 22 (1988), 259-269.

- [18] Y.K. Cheng and L. Y. Kang, "The p-maxian problem on interval graphs", *Discrete Appl. Math* 158 (2010), 1986-1993.
- [19] Y.K. Cheng, L. Y. Kang, H. Yan "The backup 2-median problem on block graphs", *Optimization Methods and Software* 164 (2010).
- [20] Y.K. Cheng and L. Y. Kang, and C. H. Lu, "The pos/neg-weighted 1-median problem on tree graphs with subtree-shaped customers", *Theor. Comput. Sci.* 411 (2010), 1038-1044.
- [21] Y.K. Cheng and L. Y. Kang, and M. F. Luo, "The backup p-median problem on network", *manuscript*.
- [22] G. Cornuejols, M. L. Fishrer, G. L. Nemhauser, "Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms", *Manage. Sci.* 23 (1977) 789-810.
- [23] Z. Drenznar, Facility location: A Survey of Application and Methods, *Springer-Verlag*, New York, Inc.
- [24] J. Hatzl, "Median problem on wheels and cactus graphs", *Computing* 80 (2007), 377-393.
- [25] L. Hua, "Applications of mathematical methods for wheat harvesting". *Chinese Math* 2: (1962) 77-91.
- [26] B. Gavish and S. Sridhar, "Computing the 2-median on tree networks is $O(n \log n)$ time." *Networks*, 26 (1995), 305-317.
- [27] A.J. Goldman, "Optimal center location in simple networks." *Transportation Sci.*, 5 (1979) 212-221.
- [28] Y. Gurevich, L. Stockmeyer, and U. Vishkin, "Solving NP-hard problems on graphs that are almost trees and an application to facility location problems", *J ACM* 31 (1984), 459- 473.
- [29] R. L. Francis, and J. A. White "Facility Layout and Location", *Prentice-Hall*, Englewood Cliffs NJ, (1994).
- [30] S. L. Hakimi, "Optimum location of switching centers and absolute centers and medians of graph", *Operation Research* 12 (1964), 450- 459.
- [31] H. W. Hamacher, S. Nickel, "Classification of location models", *Location Sci.* 6 (1998), 229-242.
- [32] F. Harary, Graph theory, Addison-Wesley, Reading, Massachusetts, (1969).
- [33] L. Hua, "Applications of mathematical methods for wheat harvesting." *Chin Math* 2, (1962) 77-91.
- [34] L. Y. Kang and Y. K. Cheng, "The p-maxian problem on block graphs", *J. Comb. Optim.* 20 (2010) 131-141.
- [35] O. Kariv and S. L. Hakimi, "An algorithmic approach to network location problems. Part II: p-medians." *SIAM J. Appl. Math.*, 37 (1979) 539-560.
- [36] S. C. Ku, C. J. Lu, B. F. Wang, and T. C. Lin, "Efficient algorithms for two generalized 2-median problem on trees", *Proceeding of the 12th International Symposium on Algorithms and Computation*, Christchurch, New Zealand, (2001), pp. 768-778.
- [37] N. Meggido, "Linear-time algorithms for linear programming in R^n and related problems", *SIAM J comput* 12 (1983), 759-776.
- [38] B. P. Mirchandani, R. Francis "Discret Location Theory", Wiley, New York, (1990), pp. 55-117.

- [39] B. P. Mirchandani, "The p-median problem and generalization", in: B. P. Mirchandani, R. L. Francis (Eds), *Discrete Location Theory*, Wiley, New York, (1990), pp. 55-117.
- [40] B. P. Mirchandani, A. Oudjit, "Localizing 2-medians on probabilistic and deterministic tree networks.", *Networks* 10 (1980), 329-350.
- [41] N. Meggido, A. Tamir, E. Zemel, and R. Chandrasekaran, "An $O(n \log^3 n)$ algorithm for the kth longest path in a tree with applications to location problems", *SIAM J comput* 10 (1981), 328-337.
- [42] L. V. Snyder, "Facility location under uncertainty: A review", *IIE Tran*, 38 (2006) 537-554 .
- [43] L. V. Snyder and M. S. Daskin, "Reliability models for facility location: the expected failure cost case", *Transport Sci*, 39 (2005) 400-416 .
- [44] L. V. Snyder and M. S. Daskin, "Stochastic p-robust location problems", *IIE Trans*, 38 (2006) 971-985 .
- [45] A. Tamir, "An $O(pn^3)$ algorithm for the p-median and related problems on tree graphs." *Operations Research Letters*, 19 (1996) 59-64.
- [46] B. F. Wang, S. C. Ku and K. H. Shi, "Cost-optimal parallel algorithms for the tree bisector problem and applications", *IEEE Trans Parallel Distributed Syst*, 12, (2001), pp. 888-898 .
- [47] H. L. Wang, B. Y. Wu and K. M. Chao, "The backup 2-center and backup 2-median problems on trees", *Networks*, 53, (2009), pp. 39-49 .
- [48] J. E. Ward, R. E. Wendell, "Using block norms for location modelling", *Operation Research* 33 (1985), 1074-1090.
- [49] B. Y. Wu and K. M. Chao, *Spanning trees and optimization problems*, Chapman and Hall/CRC Press, Boca Raton, Florida (2004).

واژه‌نامه فارسی به انگلیسی

Probability	احتمال
Vertex Partition	افراز رأسی
Backup	پشتیبان
Time Complexity	پیچیدگی زمانی
Preprocessing	پیش پردازش
Objective Function	تابع هدف
Decompose	تجزیه کردن
Facility	تسهیلات
Optimal Solution	جواب بهینه
Feasible Solution	جواب شدنی
Majority Property	خاصیت اکثریت
Vertex Optimality Property	خاصیت بهینگی رأسی
Classify	رده‌بندی
Edge Deletion Method	روش حذف یال
Subset	زیرمجموعه
Network	شبکه
Distance	فاصله
Interval graph	گراف بازه ای
Block Graph	گراف بلوکی
Planar Graph	گراف مسطح
Node	گره
Adjucent	مجاور

Disjoint	مجزا
Restriction	محدودیت
Path	مسیر
Diameter Path	مسیر قطری
Client	مشتری
Location	مکان‌یابی
Median	میانه
Triangle Inequalit	نامساوی مثلث
Semi-Obnoxious	نیمه ناخوشایند
Connected	همبند
Isomorphic	یک‌ریخت
Monotonicity	یک‌نوایی

واژه‌نامه انگلیسی به فارسی

Adjucent	مجاور
Backup	پشتیبان
Block Graph	گراف بلوکی
Classify	رده‌بندی
Client	مشتری
Connected	همبند
Decompose	تجزیه کردن
Diameter Path	مسیر قطری
Disjoint	مجزا
Distance	فاصله
Edge Deletion Method	روش حذف یال
Feasible Solution	جواب شدنی
Facility	تسهیلات
Interval graph	گراف بازه ای
Isomorphic	یک‌ریخت
Location	مکان‌یابی
Median	میانه
Majority Property	خاصیت اکثریت
Monotonicity	یکنوایی
Network	شبکه
Node	گره
Objective Function	تابع هدف

Optimal Solution	جواب بهینه
Planar Graph	گراف مسطح
Preprocessing	پیش پردازش
Probability	احتمال
Path.....	مسیر
Semi-Obnoxious.....	نیمه ناخوشایند
Subset	زیرمجموعه
Time Complexity	پیچیدگی زمانی
Triangle Inequalit	نامساوی مثلث
Vertex Optimality Property	خاصیت بهینگی رأسی
Vertex Partition	افراز رأسی

Surname: Mahdevar

Name: Kobra

Title: Backup Facility Location Problem

Supervisor: Dr. Jafar Fathali

Advisor: Dr. Nader Jafari Rad

Degree: Master of Science

Subject: Faculty Mathematics

Field: Operation Research

Shahrood University of Technology

Faculty Of Mathematical Sciences

Date: February 2013

Number of pages: 78

Keywords: Location; Median; Backup; Semi-Obnoxious; positive and negative weight

Abstract

Traditionally, location theory has been concerned with the problem in which the facility weights and their availability are known precisely. However, it is often impossible to have an accurate estimate of all these parameters. Real-life data often involve a significant portion of uncertainty, and these parameters may change with time. Thus, location models involving uncertainty have attracted increasing research efforts in recent years. Several cases of uncertainty have been defined and studied. One case of this problem is the reliability model where servers may sometimes fail, and the clients originally allocated to these servers have to request service from functioning servers.

The 2-median problem considered as a reliability-based formulation, in which each server may fail with a given probability. They called this model as the backup 2-median problem and proposed an The backup 2-median problem, asks for a pair of vertices which minimize the expected sum of distances from all vertices to the set of functioning servers. Because of important of this field, we have study many models presented in this area, then we consider the backup Semi-Obnoxious 2-median problem and propose an algorithm for this problem.



Shahrood University of Technology
Faculty Of Mathematical Sciences

Dissertation Submitted in Partial
Fulfillment of The Requirements For The
Degree of Master of Science in
Faculty Mathematics

Backup Facility Location Problem

Supervisor

Dr. Jafar Fathali

Advisor

Dr. Nader Jafari Rad

by

Kobra Mahdevar

February 2013