





دانشکده علوم ریاضی

رساله دکتری تحقیق در عملیات

بررسی مسائل مکانیابی متعادل معکوس

نگارنده: شاهده امیدی نورآبادی

استاد راهنما

دکتر جعفر فتحعلی

استاد مشاور

دکتر مهرداد غزنوی

تیر ۱۴۰۰

۴۲۸-۲۲-۶
۱۳۰۵، ۶۳۰

شماره:

تاریخ:

ویرایش:

باسمه تعالی



مدیریت تحصیلات تکمیلی

پیوست شماره ۲

دانشکده: ریاضی

گروه: ریاضی کاربردی - تحقیق در عملیات

رساله دکتری خانم شاهده امیدی نورآبادی

شماره دانشجویی: ۹۴۰۰۰۹۵

تحت عنوان: بررسی مسائل مکانیابی متعادل معکوس

در تاریخ ۱۴۰۰/۰۶/۱۰ توسط کمیته تخصصی زیر جهت اخذ مدرک رساله دکتری ارزیابی گردید و با درجه عالی مورد پذیرش قرار گرفت.

امضاء	اساتید مشاور	امضاء	اساتید راهنما
	نام و نام خانوادگی: دکتر مهرداد غزنوی		نام و نام خانوادگی: دکتر جعفر فتحعلی
-----	نام و نام خانوادگی: -----	-----	نام و نام خانوادگی: -----

امضاء	نماینده تحصیلات تکمیلی	امضاء	اساتید داور
	نام و نام خانوادگی: دکتر سیدحیدر جعفری		نام و نام خانوادگی: دکتر محمدهادی نوری اسکندری
			نام و نام خانوادگی: دکتر اردشیر دولتی
			نام و نام خانوادگی: دکتر مسعود امان

نام و نام خانوادگی رئیس دانشکده: دکتر مهدی قونیند

تاریخ و امضاء و مهر دانشکده:



تقدیم

ماحصل آموخته‌هایم را تقدیم می‌کنم به آنان که مهر آسمانی‌شان آرام بخش آلام زمینی‌ام
است
به استوارترین تکیه‌گاهم، دستان پر مهر پدرم
به سبزترین نگاه زندگیم، چشمان سبز مادرم
که هرچه آموختم در مکتب عشق شما آموختم و هرچه بگو شتم قطره‌ای از دریای بی‌کران مهربانی‌تان
را پاس نتوانم بگویم.

امروز، هستی‌ام به امید شماست و فردا کلید باغ بهشتم رضای شما
باشد که حاصل تلاشم نسیم کوزه غبار خشکی‌تان را برزاید. بوسه بر دستان پرمهرتان

پدر و مادر عزیزم

ستایش و سپاس

سپاس خدای را که سخنوران، در ستودن او بمانند و شمارندگان، شمردن نعمت‌های او ندانند و کوشندگان، حق او را گزاردن نتوانند و سلام و درود بر محمد و خاندان پاک او، طاهران معصوم، هم آنان که وجودمان وامدار وجودشان است. پروردگارا! به هر اندازه‌ای که میان مردم مرا مرتبه می‌بخشی پیش خودم به همان مقدار خوارم کن و هر عزت ظاهری که برایم پدیدار می‌سازی به همان اندازه پیش نفسم برای من خواری باطن عطا فرما.

قبل از هر چیز بر خود لازم می‌دانم از استاد محترم جناب آقای دکتر فتحعلی، که مرا در نگارش این رساله یاری داده و از راهنمایی‌های ارزنده و مفید و دقیق خود، بهره‌مند نمودند، قدردانی نمایم و همچنین از استاد گرانقدر جناب آقای دکتر غزنوی که مشاوره این اثر را بر عهده گرفتند، نهایت تشکر و قدردانی را می‌دارم. از خواهر و برادران عزیزم که بهترین همراهان من در طول زندگیم بوده‌اند، نهایت سپاس و قدردانی را دارم. همچنین از اساتید محترمی که زحمت داوری این رساله را متقبل شدند کمال تقدیر و تشکر را دارم.

شاهده امیدی نورآبادی

تیر ۱۴۰۰

تعهد نامه

اینجانب شاهده امیدی نورآبادی دانشجوی دکتری رشته تحقیق در عملیات علوم ریاضی دانشگاه شاهرود، نویسنده پایان نامه با عنوان بررسی مسائل مکانیابی متعادل معکوس، تحت راهنمایی جعفر فتحعلی متعهد می شوم:

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش های دیگر پژوهش گران، به مرجع مورد استفاده استناد شده است.
- مطالب این پایان نامه، تا کنون توسط خود، یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارایه نشده است.
- حقوق معنوی این اثر، به دانشگاه صنعتی شاهرود تعلق دارد، و مقالات مستخرج با نام “دانشگاه صنعتی شاهرود” یا “Shahrood University of Technology” به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آوردن نتایج اصلی پایان نامه تاثیرگذار بوده اند، در مقالات مستخرج از پایان نامه رعایت می گردد.
- در تمام مراحل انجام این پایان نامه، در مواردی که از موجود زنده (یا بافت های آنها) استفاده شده است، ضوابط و اصول اخلاقی رعایت شده است.
- در تمام مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته (یا استفاده شده است)، اصل رازداری و اصول اخلاق انسانی رعایت شده است.

شاهده امیدی نورآبادی

تیر ۱۴۰۰

مالکیت نتایج و حق نشر

- تمام حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه های رایانه ای، نرم افزارها و تجهیزات ساخته شده) متعلق به دانشگاه صنعتی شاهرود می باشد. این مطلب باید به نحو مقتضی، در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در این پایان نامه بدون ذکر منبع مجاز نمی باشد.

چکیده

در این رساله مسائل مکانیابی معکوس و مکانیابی متعادل را با هم ترکیب کرده‌ایم. در یک مدل، مسئله مکانیابی متعادل معکوس با طول یال‌های متغیر با کمترین هزینه و بودجه محدود روی درخت در نظر گرفته‌ایم. هدف مسئله در حالت با کمترین هزینه، اصلاح طول یال‌ها با کمترین هزینه، به گونه‌ای است که اختلاف بین ماکزیمم و مینیمم وزن مشتری‌های تخصیص یافته به دو سرویس‌دهنده (میانه)، مینیمم شود. حالت دیگر این مسئله را با بودجه محدود در نظر گرفتیم. این بودجه را برای اصلاح طول یال‌ها طوری که اختلاف بین ماکزیمم و مینیمم وزن مشتری‌های تخصیص یافته به دو سرویس‌دهنده تا حد امکان کاهش یابد در نظر گرفته‌ایم. برای هر دو حالت دو الگوریتم با پیچیدگی زمانی $O(n \log n)$ ارائه کرده‌ایم. علاوه بر این مدل، مسئله مکانیابی تک سرویس‌دهنده متعادل معکوس را روی درخت معرفی کردیم و قصد داریم طول یال‌ها را با کمترین هزینه تغییر دهیم به طوری که اختلاف فاصله بین دورترین و نزدیکترین مشتری از میانه مینیمم شود. دو حالت را برای این مسئله در نظر گرفتیم حالتی که طول یال‌ها نامحدود و حالتی که طول یال‌ها محدود باشد. در حالت نامحدود نشان دادیم که مسئله می‌تواند برای حل به یک مسئله روی گراف ستاره‌ای کاهش یابد. سپس یک الگوریتم با پیچیدگی زمانی $O(n \log n)$ برای پیدا کردن جواب بهینه ارائه شد. برای حالت طول یال‌های محدود یک الگوریتم با پیچیدگی زمانی $O(n^2)$ ارائه شد. کلمات کلیدی: مسئله مکانیابی، مکانیابی معکوس، مکانیابی متعادل، مسئله میانه.

لیست مقالات مستخرج از پایان نامه

1. S. Omid, J. Fathali and M. Nazari, "Inverse and reverse balanced facility location problems with variable edge lengths on trees", **OPSEARCH** , 57 (2020), 261-273.
2. S. Omid, J. Fathali, "Inverse single facility location problem on a tree with balancing on the distance of server to clients", **JOURNAL OF INDUSTRIAL AND MANAGEMENT OPTIMIZATION**, (2020)

۳. شاهده امیدی نورابادی، جعفر فتحعلی، (۲۰۱۹)، "مسئله مکانیابی متعادل معکوس با تغییر همزمان وزن یال و وزن راس روی درخت"، دوازدهمین کنفرانس بین المللی انجمن ایرانی تحقیق در عملیات، دانشگاه علوم و فنون مازندران، بابل، ایران.

۴. شاهده امیدی نورابادی، جعفر فتحعلی، (۲۰۱۸)، "مسئله مکانیابی متعادل معکوس با تغییر وزن یال"، دومین سمینار ملی کنترل و بهینه سازی، دانشگاه صنعتی شاهرود، شاهرود، ایران.

فهرست مطالب

ف	فهرست تصاویر
ق	فهرست جداول
ش	پیشگفتار
۱	۱ مقدمه‌ای بر مسائل مکانیابی کلاسیک
۱	۱.۱ مفاهیم مقدماتی از آنالیز پیچیدگی
۴	۲.۱ مکانیابی
۵	۳.۱ انواع مسائل مکانیابی
۹	۴.۱ مسائل پرکاربرد مکانیابی
۱۱	۱.۴.۱ مسئله مکانیابی p -میانه
۱۶	۲.۴.۱ مسئله مکانیابی p -مرکز
۱۸	۵.۱ مسئله مکانیابی متعادل
۲۳	۲ مروری بر مسائل مکانیابی معکوس
۲۳	۱.۲ مقدمه
۲۴	۲.۲ مسائل مکانیابی اصلاحی
۲۶	۱.۲.۲ مدل مکانیابی معکوس با حداقل هزینه
۳۴	۲.۲.۲ مدل مکانیابی معکوس با محدودیت بودجه‌ای
۳۵	۳.۲.۲ مدل مکانیابی معکوس با تابع هدف کراندار
۳۶	۴.۲.۲ مدل مکانیابی ترفیع و تنزیل
۳۷	۳ مسائل مکانیابی متعادل معکوس
۳۸	۱.۳ مسائل مکانیابی متعادل معکوس با طول یال‌های متغیر
۳۹	۱.۱.۳ مدل معکوس با حداقل هزینه
۴۶	۲.۱.۳ مدل معکوس با محدودیت بودجه‌ای

۵۱	۴ مسائل مکانیابی ۱- میانه معکوس با تعادل فاصله
۵۱	۱.۴ تعریف مسئله
۵۲	۲.۴ مسئله با طول یال‌های نامحدود
۵۴	۳.۴ تعادل فاصله روی گراف ستاره‌ای
۵۸	۴.۴ مسئله با طول یال‌های محدود
۶۷	مراجع

فهرست تصاویر

۳		۱.۱	تصویری از مجموعه کلاس‌های تعریف شده
۴۱		۱.۳	درخت T و زیر درخت‌های T_1, T_2 و Tv_j
۴۴		۲.۳	درخت T با ۱۹ راس
۵۷		۱.۴	درخت T با ۱۰ راس

فهرست جداول

۱۹		جواب مسئله مکانیابی متعادل	۱.۱
۴۴		وزن راس‌ها در درخت T	۱.۳
۴۵		هزینه اصلاح طول یال‌ها در درخت T	۲.۳
۵۸		طول و هزینه اصلاح طول یال‌ها در درخت T	۱.۴
۶۴		کران طول یال‌ها در درخت T	۲.۴
۶۵		نتایج تکرار برای مثال ۱.۴.۴	۳.۴
۶۵		طول بهینه یال‌ها برای مثال ۱.۴.۴	۴.۴

پیشگفتار

نظریه مکانیابی حوزه غنی و ارزشمندی است که دامنه وسیعی از مسائل مختلف را بررسی کرده است و نقش بسیار برجسته‌ای را در تحقیق در عملیات ایفا می‌کند. علم مکانیابی به علت نیاز برای حل مسائل عملی از طریق چالش‌های تکنیکی، تئوریکی و نیز با توجه به مسائل برخاسته از رشته‌های دیگر از جمله اقتصاد، مدیریت، جغرافیا و فنی همواره رو به تکامل و پیشرفت بوده است. تصمیم‌گیری در مورد مکان تسهیلات یک عامل بسیار مهم در برنامه ریزی‌های استراتژیک برای کارهای بسیاری در بخش‌های خصوصی و عمومی است که در اکثر مواقع با منابع زیادی از سرمایه سروکار داشته و تاثیرات اقتصادی آن بلند مدت است. از این لحاظ مکان‌های انتخاب شده باید هم از لحاظ سیستم فعلی کارا باشند و هم با تغییر عوامل محیطی، تغییر جمعیت و عوامل دیگر کارایی خود را حفظ کنند. بنابراین مسائل مکانیابی نیازمند یک افق برنامه‌ریزی طولانی مدت هستند. در سال‌های اخیر تلاش‌ها برای توسعه و پیشرفت در زمینه مسائل مکانیابی از لحاظ تئوریکی، مدلسازی و روش‌های حل بسیار چشمگیر بوده است.

از جمله کاربردهای مسائل مکانیابی می‌توان به تعیین مکان‌های بهینه برای واحدهای اورژانس، آتش‌نشان‌ها، فرودگاه‌ها، تصفیه‌خانه‌ها و غیره اشاره کرد. در یک مدل مکانیابی کلاسیک، یک مجموعه از مشتریان روی یک گراف و یا یک فضای حقیقی چند بعدی مستقر شده و هدف این است که با توجه به معیارهای در نظر گرفته شده توسط تصمیم‌گیرنده بهترین مکان برای تاسیس سرویس‌دهنده‌ها روی آن سیستم تعیین شود.

در تقابل با مدل‌های مکانیابی کلاسیک، ممکن است در عمل یا موقعیت‌هایی مواجه شویم که در آنها علاوه بر مشتریان، سرویس‌دهنده‌ها نیز از قبل روی سیستم تحت مطالعه مستقر شده و با گذر زمان به دلیل تغییر ویژگی‌های سیستم کارایی خود را از دست داده‌اند. از طرفی جابجا کردن آنها به دلیل برخی محدودیت‌ها امکان‌پذیر نمی‌باشد در این حالت، تصمیم‌گیرنده معمولاً یکی از مسائل اصلاحی (بهبود شبکه) زیر را در نظر گرفته و حل می‌نماید:

- (الف) مسئله مکانیابی معکوس با حداقل هزینه: برخی از پارامترهای تاثیرگذار سیستم تحت مطالعه (مانند طول یال‌ها یا وزن راس‌ها روی گراف، وزن نقاط یا مختصات نقاط روی فضای حقیقی) با حداقل هزینه ممکن اصلاح شود تا اینکه مکان سرویس‌دهنده‌های از قبل معین با توجه به پارامترهای اصلاح شده بهینه گردند.

● (ب) مسئله مکانیابی معکوس با محدودیت بودجه‌ای: برخی از پارامترهای سیستم تحت مطالعه را با یک بودجه داده شده طوری اصلاح شود که مکان سرویس‌دهنده‌های تاسیس شده در سیستم اصلاح شده، تا حد ممکن بهبود یابند.

البته، می‌توان مدل‌های مختلف دیگری از این مسائل بهبود شبکه را مطرح نمود که همگی در حوزه مسائل مکانیابی معکوس قرار دارند.

مسائل مکانیابی متعادل در چند سال اخیر مورد توجه قرار گرفته است. این نوع مسائل مکانیابی، مکان سرویس‌دهنده‌ها را طوری انتخاب می‌کند که تعادل در سرویس‌دهی به نقاط تقاضا به حداکثر برسد.

هدف در این رساله تلفیق مسئله مکانیابی متعادل با مسئله مکانیابی معکوس است و به حل مسائلی که هم شرط تعادل و هم شرط کمترین هزینه در تغییر پارامترهای مسئله برای ایجاد جواب بهینه برقرار است، کمک می‌کند. به عبارتی دیگر در این رساله به دنبال بررسی و حل مسائل مکانیابی متعادل معکوس هستیم. مسائل مکانیابی متعادل معکوس موضوع جدیدی می‌باشد که اخیراً مورد توجه قرار گرفته است. در این مسائل یک جواب شدنی نامتعادل از پیش تعیین شده داریم، هدف این است که با اصلاح پارامترهای اساسی مسئله با کمترین هزینه، جواب نامتعادل را تا حد امکان متعادل کنیم و جواب بهینه شود. در این مسائل شرط تعادل، شرط بهینگی است.

برای مثال دو بانک را در سیستم شهری در نظر بگیرید تعداد مشتریانی که از بانک شماره ۱ سرویس می‌گیرند بیشتر از تعداد مشتریانی است که از بانک شماره ۲ سرویس می‌گیرند که باعث ایجاد خستگی برای کارمندان و مشتریان و صف‌های طولانی و در کل کاهش سرویس‌دهی در بانک ۱ می‌شود. یک راه حل موثر اصلاح برخی از پارامترهای تاثیرگذار از قبیل ایجاد سیاست‌های تشویقی در بانک ۲، کم کردن بخشی از خدمات در بانک ۱ و افزودن آن خدمات به بانک ۲ و ایجاد تسهیلات در بخش حمل و نقل برای سهولت دسترسی مشتریان به بانک ۲ به طوری که تعداد مشتریانی که از هر دو بانک سرویس می‌گیرند تا حدی متعادل باشند. سازماندهی این رساله به صورت زیر می‌باشد:

در فصل ۱، به یادآوری برخی از تعاریف و مفاهیم مقدماتی مورد نیاز می‌پردازیم و سپس مسائل مکانیابی کلاسیک معرفی شده و خلاصه‌ای از جدیدترین مطالعات انجام گرفته در زمینه این مسائل ارائه می‌گردد. در فصل ۲ با انواع مسائل مکانیابی اصلاحی، مسائل مکانیابی متعادل و تاریخچه این مسائل آشنا می‌شوید. در فصل ۳ یک مدل جدید مسئله مکانیابی متعادل معکوس با دو حالت (بودجه نامحدود و محدود) مطرح شده و الگوریتم‌هایی با پیچیدگی زمانی $O(n \log n)$ برای حل این دو حالت ارائه می‌شود. در فصل ۴ حالت خاصی از مسائل مکانیابی تک وسیله‌ای متعادل معکوس روی درخت در دو حالت (طول یال‌ها محدود و نامحدود) مطرح شده و الگوریتم‌های به ترتیب با پیچیدگی زمانی $O(n^2)$ و $O(n \log n)$ برای حل این مسائل ارائه می‌شود.

فصل ۱

مقدمه‌ای بر مسائل مکانیابی کلاسیک

در بخش اول این فصل، تعاریف مقدمات و پایه‌ای مورد نیاز برای فصل‌های بعدی این رساله بیان می‌شوند. مطالب این بخش از مراجع [۲۱] و [۳۲] برگرفته شده است.

۱.۱ مفاهیم مقدماتی از آنالیز پیچیدگی

در ابتدای این فصل تعاریف و مفاهیم مربوط به آنالیز پیچیدگی^۱ که در بیان زمان اجرای^۲ الگوریتم‌ها به کار می‌روند، معرفی خواهد شد.

تعریف ۱.۱.۱. زمان اجرای یک الگوریتم برابر با تعداد عملیات حسابی، مقایسه‌ای و جایگزینی در طول اجرای آن الگوریتم برای حل یک مسئله می‌باشد. اگر n نشانگر تعداد داده‌های ورودی الگوریتم باشد، آنگاه زمان اجرای آن معمولاً با $T(n)$ نمایش داده می‌شود.

در بحث تحلیل الگوریتم‌ها به جای در نظر گرفتن رفتار دقیق تابع، رفتار مجانبی آنها مورد بررسی قرار می‌گیرد. برای تخمین زمان اجرای یک الگوریتم و میزان پیچیدگی آن، از نمادهایی به نام نماد مجانبی استفاده می‌شود. یکی از این نمادها O است که طبق روش خاصی محاسبه می‌شود و در تحلیل الگوریتم‌ها کاربرد فراوانی دارد.

^۱complexity

^۲running time

تعریف ۲.۱.۱. برای تابع مفروض $g(n)$ ، گویند $f(n) = O(g(n))$ یا $f(n) \in O(g(n))$ ، اگر یک مقدار حقیقی مثبت β و یک عدد طبیعی n_0 وجود داشته باشد به طوری که به ازای هر $n \geq n_0$ رابطه

$$0 \leq f(n) \leq \beta g(n)$$

برقرار باشد. به عبارتی دیگر می‌توان گفت

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \neq \infty \implies f(n) = O(g(n))$$

اگر $f(n) = O(g(n))$ ، آنگاه $\beta g(n)$ به عنوان یک کران بالای مجانبی برای تابع $f(n)$ تعریف می‌گردد.

تعریف ۳.۱.۱. یک الگوریتم دارای زمان اجرای چند جمله‌ای است هرگاه $T(n) = O(n^k)$ ، که در آن k یک عدد طبیعی ثابت است.

تعریف ۴.۱.۱. یک الگوریتم دارای زمان اجرای شبه چند جمله‌ای است هرگاه پیچیدگی زمانی آن یک چند جمله‌ای از اندازه داده‌های ورودی و بزرگی آن‌ها باشد.

کلاس‌های P ، NP و NP -سخت

یک مسئله به عنوان مسئله تصمیم‌گیری شناخته می‌شود اگر همواره دارای جواب بله و خیر باشد. یک مسئله بهینه‌سازی را می‌توان با اضافه کردن کران U (کران بالا یا پایین)، روی مقداری که باید بهینه گردد و بررسی سوالات زیر به یک مسئله تصمیم‌گیری تبدیل کرد:

۱. برای مسئله مینیمم سازی: آیا یک جواب شدنی وجود دارد که مقدار تابع هدف برای آن حداکثر U باشد؟

۲. برای مسئله ماکزیمم سازی: آیا یک جواب شدنی وجود دارد که مقدار تابع هدف برای آن حداقل U باشد؟

لازم به ذکر است که اگر یک الگوریتم بتواند مسئله تصمیم‌گیری را حل کند، آنگاه مسئله بهینه‌سازی را نیز حل خواهد کرد. همچنین، اگر یک الگوریتم مسئله بهینه‌سازی را در زمان چند جمله‌ای حل کند، آنگاه مسئله تصمیم‌گیری را در زمان چند جمله‌ای حل خواهد کرد.

تعریف ۵.۱.۱. مسئله A به یک مسئله B در زمان چند جمله‌ای کاهش پذیر است هرگاه یک تابع کاهش R موجود باشد به طوری که

۱. به ازای هر داده ورودی x ، اگر x یک ورودی بله برای A باشد، آنگاه $R(x)$ یک ورودی بله برای B فراهم کند و برعکس. به طور مشابه اگر x یک ورودی خیر برای A باشد، آنگاه $R(x)$ یک ورودی خیر برای B فراهم کند و برعکس.

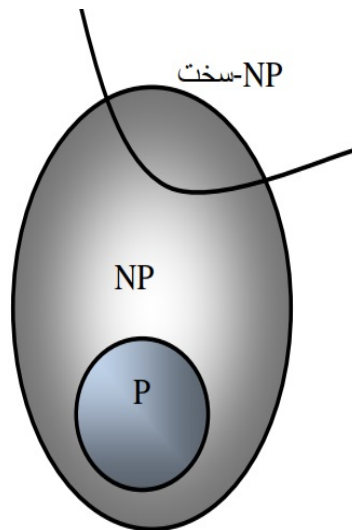
۲. مقدار R در زمان چندجمله‌ای محاسبه گردد.

به عبارت دیگر، گویند مسئله A به مسئله B در زمان چند جمله‌ای کاهش پذیر است اگر یک الگوریتم چندجمله‌ای موجود باشد به طوری که هر نمونه از A را به یک نمونه از B با جواب‌های یکسان تبدیل نماید. اگر مسئله A در زمان چند جمله‌ای به یک مسئله B کاهش پذیر باشد، در این صورت

$$A \leq_p B.$$

تعریف ۶.۱.۱. کلاس P ^۳ مجموعه‌ای از مسائل تصمیم‌گیری است که در زمان چندجمله‌ای قابل حل هستند.

تعریف ۷.۱.۱. کلاس NP ^۴ شامل آن دسته از مسائلی است که تضمینی برای حل آنها در زمان چند جمله‌ای وجود ندارد ولی می‌توان در یک زمان چندجمله‌ای ارزش بله یا خیر یک جواب کاندید را تشخیص داد.



شکل ۱.۱: تصویری از مجموعه کلاس‌های تعریف شده

تعریف ۸.۱.۱. گویند یک مسئله Q از نوع NP -سخت ^۵ است اگر برای هر $Q' \in NP$ داشته باشیم

$$Q' \leq_p Q.$$

تعریف ۹.۱.۱. درجه یک راس در گراف G برابر با تعداد یال‌هایی است که بر آن راس وارد شده‌اند.

³Deterministic Polynomail

⁴Nondeterministic Polynomail

⁵NP-hard

تعریف ۱۰.۱.۱. گراف G را همبند گویند هرگاه بین هر دو راس دلخواه آن یک مسیر وجود داشته باشد.

تعریف ۱۱.۱.۱. گرافی که در آن هر دو راس دلخواه دقیقاً با یک مسیر به هم متصل باشند، یک درخت نامیده می‌شود. به عبارت دیگر، هر گراف همبند بدون دور یک درخت نامیده می‌شود.

تعریف ۱۲.۱.۱. یک گراف ستاره، درختی با n راس است که یکی از رئوس آن دارای درجه $n-1$ بوده و درجه $n-1$ راس آن برابر ۱ باشند.

۲.۱ مکانیابی

مفهوم مکانیابی از گذشته‌های دور در تصمیم‌گیری‌هایی نظیر انتخاب بهترین غار برای زندگی، بهترین مکان برای احداث روستاها، بهترین مکان برای ساختن خانه‌ها و غیره حضور داشته و انسان‌ها این گونه مسائل را با روش‌های ابتکاری حل می‌کردند. چندین عامل باعث توجه فراوان به مسائل مکانیابی و به وجود آمدن تاریخچه عظیم آن شده است. اولاً، مسائل مکانیابی به طور مداوم در تمام سطوح برنامه‌ریزی‌های بشری از صنعت و انبارداری تا نهادهای دولتی حضور دارند. ثانیاً، تصمیم‌گیری‌های مکانی ماهیت استراتژیک دارند، یعنی اغلب آنها با منابع عظیمی از سرمایه سروکار داشته و تأثیرات اقتصادی آنها بلند مدت می‌باشد. سوماً، اکثر مدل‌های مکانیابی به سختی قابل حل هستند. حتی نمونه‌های بزرگ مربوط به پایه‌ای‌ترین مدل‌ها نیز دارای پیچیدگی محاسباتی بالایی هستند.

نهایتاً، شکل ساختاری مدل‌های مکانیابی (متغیرها، قیود و تابع هدف) بسته به نوع مسئله مکانیابی تحت مطالعه تعریف می‌شود، در نتیجه هیچ مدل جامعی برای مسائل مکانیابی وجود ندارد.

در آنچه امروز به عنوان مسئله مکانیابی شناخته می‌شود، مجموعه‌ای از نقاط به عنوان مکان مشتریان داده شده و هدف، پیدا کردن بهترین مکان برای یک یا چندین سرویس‌دهنده است به طوری که با توجه به شرایط موجود بتوانند به بهترین شکل به مشتریان سرویس‌دهی کنند. از جمله کاربردهای مسائل مکانیابی می‌توان به موارد زیر اشاره کرد:

- مکانیابی انبارها در یک زنجیره تامین برای مینیمم‌سازی میانگین زمانی دسترسی فروشگاه‌ها به این انبارها
- مکانیابی ترمینال‌ها، مراکز آموزش و ایستگاه‌های اتوبوس به منظور تسهیل در ارائه خدمات عمومی
- مکانیابی مراکز خرید به منظور ماکزیمم کردن سود حاصل از فروش محصولات

- مکانیابی کارخانه‌های مواد شیمیایی خطرناک و مکان تخلیه زباله برای به حداقل رساندن ضررهای ناشی از این‌ها به مردم منطقه
- مکانیابی خطوط هوایی و فرودگاه‌ها، سیستم‌های مخابرات و کارخانه‌های تولیدی
- مکانیابی مراکز خدمات دولت، ادارات، بانک‌ها و دفاتر خدماتی
- مکانیابی موسسات صنعت و تجاری برای به حداقل رساندن هزینه‌ها
- مکانیابی مراکز بهداشتی درمانی، مراکز انتقال خون و مراکز مراقبت اضطراری

۳.۱ انواع مسائل مکانیابی

در این بخش سعی شده تا حد امکان به معرفی مدل‌های مکانیابی کلاسیک پرکاربرد پرداخته و خلاصه‌ای از رویکردهای حل آنها، برای آشنایی بیشتر با مسائل مکانیابی، ارائه گردد.

مولفه‌های اساسی یک مسئله مکانیابی:

- فضای تصمیم: فضای تصمیم برای مکان سرویس‌دهنده‌ها و سرویس‌گیرنده‌ها به سه دسته تقسیم بندی می‌شود:

۱. فضای پیوسته:

هرگاه فضای مکان تاسیسات و نقاط تقاضا پیوسته باشد، یعنی توسط متغیرهایی که به صورت پیوسته تغییر می‌کند مشخص شود، مدل را پیوسته گویند. در این مدل‌ها مکان سرویس‌دهنده را در فضای n بعدی R^n پیدا می‌کنیم. اکثر مسائل مکانیابی در فضای حداقل دو بعدی تعریف می‌شوند. مسائل دو بعدی به خاطر ویژگی‌های هندسی و قابلیت ادراک بصری آنها بیشتر مورد توجه قرار می‌گیرند. در مواجهه با مسائل پیوسته از ابزارهای هندسه و آنالیز (آنالیز تابعی که معمولاً از ویژگی‌های تحدب و تقعر استفاده می‌شود) بهره می‌گیریم.

۲. فضای گسسته:

مدل‌های مکانیابی گسسته، مدل‌هایی هستند که در آنها باید مکان سرویس‌دهندگان را تنها روی نقاط از پیش تعیین شده پیدا کنیم. در این مدل‌ها، مجموعه‌ای متناهی از نقاط کاندیدا داده شده است که توسط متغیرهای گسسته نشان داده می‌شوند. در مدل‌های مکانیابی پیوسته، بر خلاف مدل‌های گسسته نمی‌توان لیست جامع و کاملی از نقاط در دسترس را ارائه داد. از این جهت مدل‌های پیوسته را مدل‌های ایجاد مکان و مدل‌های گسسته را مدل‌های انتخاب مکان نیز می‌نامند.

یکی از ابزارهای مفید در ریاضیات در مواجهه با مسائل بزرگ و پیوسته، ایده گسسته سازی فضا است. در مدل‌های گسسته، در حقیقت هر نقطه نماینده یک ناحیه از فضا است. لازم به ذکر است که در حالت پیوسته باید یک تابع برای تخمین فاصله بین نقاط انتخاب شود. که نحوه انتخاب این تابع مناسب، موضوع مقالات زیادی در زمینه مکانیابی پیوسته بوده است، در حالی که در حالت گسسته فاصله واقعی پیموده شده و یا از قبل محاسبه شده، مورد استفاده قرار می‌گیرد. بنابراین مدل‌های پیوسته به راحتی موقعیت واقعی و عملی را مدل می‌کنند و به سرعت اجرا می‌شوند. البته این نکته قابل ذکر است که به خاطر استفاده از تابع تخمین از دقت کمتری نسبت به انواع گسسته برخوردار هستند. البته انتخاب نوع فضا با توجه به شرایط مسئله به نظر تصمیم‌گیرنده بستگی دارد.

۳. شبکه‌ای:

مدل‌های شبکه‌ای مدل‌هایی هستند که در آن سرویس‌دهندگان روی رئوس یا یال‌های یک شبکه مستقر هستند. در حوزه مکانیابی با فضای شبکه‌ای، مسائلی وجود دارد که در حالت کلی NP سخت هستند ولی در گراف‌های خاص مانند درخت‌ها در زمان چند جمله‌ای قابل حل می‌باشند.

• سرویس‌دهندگان:

فرض کنید P تعداد سرویس‌دهنده باشد. حالت‌های $P = 1$ و $P > 1$ را در نظر می‌گیریم:

۱. $P = 1$ ، در این حالت فقط یک سرویس‌دهنده باید مستقر شود.
 ۲. $P > 1$ ، در این حالت علاوه بر پیدا کردن محل استقرار سرویس‌دهنده‌ها باید چگونگی تخصیص مشتریان به سرویس‌دهندگان نیز مشخص شود.
- در حالتی که چندین سرویس‌دهنده را مکانیابی می‌کنیم ممکن است ارتباط بین سرویس‌دهندگان جدید نیز دارای اهمیت باشد. مدل‌هایی که این هدف را برآورده می‌کنند شامل فاصله وزن‌دار بین سرویس‌دهندگان جدید می‌باشند. به عنوان مثال می‌توان به مکانیابی تاسیسات نظامی اشاره کرد که در آن پراکنده شدن و فاصله داشتن سرویس‌دهندگان جدید میزان ضرر و خسارت و آسیب پذیری را کاهش می‌دهد. همچنین در مکانیابی پمپ بنزین و یا فست فودها، تصمیم‌گیرنده سعی در پراکنده کردن آنها دارد تا از ایجاد رقابت مستقیم بین آنها با یکدیگر جلوگیری کند.

بر مبنای این پارامتر، دو نوع مدل مختلف وجود دارد:

- مدل‌هایی که تعداد سرویس‌دهندگان جدید در آن ثابت است
- مدل‌هایی که تعداد سرویس‌دهندگان جدید در آن متغیر است.

● مشتریان:

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

● اندازه فاصله:

فاصله یک معیار عددی برای تشخیص میزان دوری اشیاء است. در مسائل مکانیابی، فرض می‌کنیم که ارتباط بین تاسیسات مستقیماً وابسته به موقعیت مکانی آنها باشد و معمولاً به شکل تابعی از فاصله بیان می‌شود. بنابراین مفهوم فاصله نقش محوری در مکانیابی پیوسته دارد. با توجه به موقعیت و شرایط مسئله اندازه‌های مختلفی برای ارزیابی فاصله به کار می‌رود. استفاده از نرم‌ها به عنوان تابع تخمین فاصله در یک مسئله مکانیابی، اولین بار توسط لاو و موریس^۶ پیشنهاد شد [۴۹]. وارد و وندل^۷ نیز استفاده از نرم‌های بلوکی را مورد مطالعه قرار دادند [۷۲]. برای اندازه‌گیری فاصله بین مشتریان و سرویس‌دهندگان معمولاً از مترهای زیر استفاده می‌شود:

– نرم L_p که $1 \leq p < \infty$

این فاصله بین دو نقطه $x = (x_1, x_2)$ و $y = (y_1, y_2)$ در صفحه به صورت زیر تعریف می‌شود:

$$L_p(x, y) = (w_1|x_1 - y_1|^p + w_2|x_2 - y_2|^p)^{\frac{1}{p}}$$

– فاصله شبکه‌ای:

برابر کوتاهترین مسیر بین دو نقطه روی شبکه است. برای محاسبه این فاصله الگوریتم‌های مختلفی پیشنهاد شده است که از آن جمله می‌توان الگوریتم‌های دایکسترا^۸ و فلوید^۹ را نام برد [۸].

– فاصله همینگ^{۱۰}

^۶Love and Morris

^۷Ward and Wendell

^۸Dijkstra

^۹Floyd

^{۱۰}Hamming

• تابع معیار(هدف):

این تابع به فواصل میان مشتریان و سرویس‌دهنده‌ها وابسته بوده و کیفیت مکان‌های مورد نظر برای استقرار سرویس‌دهنده‌ها را معین می‌کند توابع معیار مهم عبارتند از:

– تابع Minisum

– تابع Minimax

– تابع Maxsum

– تابع Maxmin

– تابع Minmax-Minsum

– تابع Maxmin-Maxsum

• وزن‌ها:

برای هر مشتری یک وزن اختصاص داده می‌شود که بیانگر میزان اهمیت آن مشتری در مسئله مکانیابی می‌باشد.

ظرفیت سرویس‌دهنده:

یک سرویس‌دهنده ممکن است با ظرفیت سرویس محدود (یعنی سرویس‌دهنده مقدار معینی از تقاضای مشتری را تامین کند) و یا ظرفیت سرویس نامحدود باشد. (یعنی سرویس‌دهنده هر مقدار از تقاضای مشتری را تامین می‌کند).

همچنین سرویس‌دهنده‌های یک مسئله مکانیابی به دو نوع سرویس‌دهنده‌های خوشایند و ناخوشایند^{۱۱} طبقه‌بندی می‌گردد. سرویس‌دهنده‌های ناخوشایند به سرویس‌دهنده‌هایی اطلاق می‌گردد که برخلاف سرویس‌دهنده‌های خوشایند باید تا حد ممکن از مشتریان دور باشند. نمونه چنین سرویس‌دهنده‌هایی شامل تاسیسات هسته‌ای، تجهیزات نظامی، تصفیه خانه‌ها، مکان‌های تخلیه زباله و غیره می‌باشند. این نوع مسائل مکانیابی اولین بار در سال ۱۹۷۵ توسط گلدمن^{۱۲} [۳۷] معرفی شدند. برای مطالعات بیشتر در این زمینه به [۷۴] و [۵۳] مراجعه کنید.

بنابراین انواع مختلفی از مسائل مکانیابی وجود دارد که برخی از مهم‌ترین آنها عبارتند از: مسئله مکانیابی میانه، مسئله مکانیابی مرکز، مسئله مکانیابی سرویس‌دهنده بدون ظرفیت، مسئله پوششی، مسئله تخصیص درجه دوم، مسئله مکانیابی هاب، مسئله مکانیابی پشتیبان، مسئله مکانیابی سرویس‌دهنده ناخوشایند، مسئله مکانیابی سلسله مراتبی، مسئله مکانیابی رقابتی، مسئله مکانیابی انبار، مسئله مکانیابی چند معیاره، مسئله مکانیابی بامانع، مسئله مکانیابی متعادل، مسئله مکانیابی شرطی، مسئله مکانیابی تصادفی. مسئله مکانیابی میانه

¹¹Obnoxious

¹²Goldman

و مکانیابی مرکز و معکوس آنها، همچنین مسئله مکانیابی متعادل از مسائل مهم و پرکاربرد مکانیابی هستند که در بخش‌های بعدی به معرفی جامع آنها می‌پردازیم.

۴.۱ مسائل پرکاربرد مکانیابی

مسئله مکانیابی به صورت تئوری اولین بار توسط فرما^{۱۳} در قرن هفدهم به صورت زیر مطرح شد:

”فرض کنید سه نقطه در صفحه داده شده است، نقطه چهارم را طوری پیدا کنید که مجموع فواصل آن تا سه نقطه داده شده، کمینه گردد.“

اولین جواب این مسئله که به صورت هندسی بیان شده بود، توسط توریچلی^{۱۴} در سال ۱۶۴۰ مطرح گردید. توریچلی مشاهده کرد که اگر سه نقطه موجود را سه راس یک مثلث در نظر بگیریم محل برخورد دایره‌های محیطی مثلث‌های متساوی الاضلاعی که بر روی اضلاع مثلث اصلی و رو به بیرون آن ساخته می‌شوند جواب مسئله است. به همین دلیل این نقطه، معروف به نقطه توریچلی می‌باشد.

در ادامه، این مسئله با اختصاص وزن‌های مختلف به نقاط مفروض توسط وبر^{۱۵} تعمیم یافته به همین دلیل مسئله مذکور به مسئله فرما-وبر شهرت یافت. مسئله فرما-وبر به صورت زیر می‌باشد.

فرض کنید n نقطه $p_j = (p_{1j}, p_{2j})$ ، $j = 1, \dots, n$ در صفحه موجود باشند و وزن متناظر با نقطه p_j باشد. هدف یافتن نقطه‌ای مانند X است به طوری که مجموع وزنی فاصله نقطه X با نقاط موجود در صفحه کمینه شود. یعنی اگر فاصله X تا p_j را با $d(X, p_j)$ نمایش دهیم آنگاه مسئله به صورت زیر خواهد بود.

$$\min \sum_{j=1}^n w_j d(X, p_j)$$

مسئله فوق را مسئله تک وسیله‌ای^{۱۶} یا کمترین مجموع^{۱۷} نیز گویند. تابع هدف این مسئله محدب است. لذا هر جواب بهینه موضعی یک جواب بهینه سراسری است پس برای یافتن جواب این مسئله کافی است مشتق را برابر صفر قرار داده و نقطه مینیمم موضعی را بیابیم. به این ترتیب مختصات نقطه $X = (x, y)$ در روابط زیر صدق می‌کند.

$$x = \frac{\sum_{j=1}^n \frac{w_j p_{1j}}{d(X, p_j)}}{\sum_{j=1}^n \frac{w_j}{d(X, p_j)}}, \quad y = \frac{\sum_{j=1}^n \frac{w_j p_{2j}}{d(X, p_j)}}{\sum_{j=1}^n \frac{w_j}{d(X, p_j)}}$$

¹³Fermat

¹⁴Torricelli

¹⁵Weber

¹⁶Single facility location

¹⁷Minisum

از نتیجه بدست آمده در می‌یابیم که x و y به X وابسته هستند. در نتیجه با یک روش تکرار ساده روبرو می‌شویم که الگوریتم وایزفیلد نامگذاری شده است. در این الگوریتم با شروع از یک نقطه اولیه یک دنباله تکراری مسیر را بدست آورده و به سمت جواب بهینه حرکت می‌کنیم. برای مطالعه بیشتر به [۴۹] مراجعه کنید.

اگر هدف یافتن نقطه X به قسمی باشد که فاصله وزنی X تا دورترین نقطه‌ی موجود در صفحه مینیمم شود آنگاه مسئله به صورت زیر می‌باشد.

$$\min \max_{j=1, \dots, n} w_j d(X, p_j)$$

مسئله فوق را مسئله تک وسیله‌ای کمینه-بیشینه می‌نامند. حال اگر به جای پیدا کردن یک نقطه به دنبال یافتن چند نقطه باشیم یعنی چند مکان برای وسایل جدید به گونه‌ای پیدا کنیم که نزدیکترین نقطه را به نزدیکترین وسیله نسبت دهیم و بخواهیم مجموع وزنی فواصل را مینیمم کنیم، مسئله را مسئله چند وسیله^{۱۸} با کمترین مجموع و در حالت دیگر آن را مسئله چند وسیله‌ای کمینه-بیشینه گویند. یعنی اگر $X = \{x_1, \dots, x_n\}$ مجموعه وسایل جدید باشد که باید مکانیابی شود و

$$d(X, p) = \min_{x_i \in X} d(x_i, p)$$

آنگاه تابع هدف مسائل فوق به ترتیب به صورت زیر نوشته می‌شود.

$$\min \sum_{j=1}^n w_j d(X, p_j)$$

۹

$$\min \max_{j=1, \dots, n} w_j d(X, p_j)$$

واضح است که در حالت چند وسیله‌ای علاوه بر پیدا کردن مکان وسایل، مسئله تخصیص نقاط به وسایل جدید نیز مورد نظر است.

در بحث فوق اگر مکان وسایل جدید بتواند در هر نقطه‌ای از صفحه قرار بگیرد، آنگاه مسئله را مسئله مکانیابی پیوسته گویند. برای این حالت راه‌حلهایی برای زمانی که تابع فاصله یعنی $d(X, p)$ به صورت یکی از نرم‌های خطی، اقلیدسی و چپیشف می‌باشد در [۳۰] ارائه شده است. در حالت گسسته مکان وسایل جدید می‌تواند در نقاط موجود قرار گیرند و حالت خاصی از آن وقتی است که نقاط موجود روی یک شبکه قرار داشته و مکان وسایل جدید نیز باید نقطه‌ای از شبکه باشد. در این حالت وقتی تابع هدف به صورت کمترین مجموع و یا کمینه-بیشینه است، به ترتیب مسئله را مسئله میانه و مرکز می‌نامند. همچنین اگر هدف پیدا کردن مکان p وسیله جدید باشد، آنگاه این مسائل به ترتیب مسئله $-p$ میانه و $-p$ مرکز نامیده می‌شوند.

¹⁸Multi facility location

۱.۴.۱ مسئله مکانیابی p -میان

مسئله p -میان کاربردهای فراوانی دارد که از آن جمله می‌توان به مکانیابی مراکز شبکه‌های کامپیوتری، مراکز توزیع کالا، مراکز اداری و نظامی اشاره کرد. میان، نقطه‌ای از شبکه است که مجموع فواصل وزن دار سایر نقاط تا آن نقطه، مینیمم باشد. در مسئله p -میان، میانه‌ها می‌توانند هر نقطه‌ای از شبکه انتخاب شوند، که در این حالت مسئله را مسئله p -میان محض گویند. اما اگر هدف پیدا کردن نقاط میان تنها روی راس‌ها باشد مسئله را مسئله p -میان راسی نامند. در حالت کلی مسئله p -میان به صورت زیر تعریف می‌شود.

تعریف ۱.۴.۱. مسئله مکانیابی p -میان، انتخاب p مکان از بین n مکان موجود به عنوان سرویس دهنده است به گونه‌ای که مجموع وزن دار فاصله این p مکان تا سایر مکان‌های موجود کمینه گردد.

مگیدو^{۱۹} و سوپویتز^{۲۰} در سال ۱۹۸۴ ثابت کردند که مسئله p -میان در فضای R^k ، $k \geq 1$ با متر اقلیدسی و خطی یک مسئله NP -سخت است [۵۲]. در ادامه این کار، درزنر^{۲۱} الگوریتم کارایی برای مسئله ۲-میان روی صفحه تحت نرم اقلیدسی ارائه داد [۲۴]. مسئله مکانیابی میان روی شبکه در سال ۱۹۶۴ با انتشار مقاله حکیمی مورد توجه قرار گرفت. وی همچنین نشان داد که حداقل یک جواب بهینه برای مسئله p -میان وجود دارد که تنها شامل رأس‌های شبکه می‌باشد [۴۰].

گلدمن نشان داد که مکان ۱-میان روی درخت مستقل از طول یال‌های درخت است و تنها به وزن رأس‌ها وابسته می‌باشد و بر مبنای آن یک الگوریتم خطی برای این مسئله ارائه نمود [۳۶]. به علاوه برای مسئله ۲-میان روی شبکه‌های درختی یک الگوریتم با زمان اجرای $O(n \log n)$ توسط گاویش^{۲۲} و همکاران ارائه گردید [۳۵].

مدل ریاضی مسئله p -میان

فرض کنید n نقطه به عنوان مکان استقرار تقاضاها در صفحه موجود است. هر نقطه A_i ، $i = \{1, \dots, n\}$ دارای مختصات $A_i = (a_i, b_i)$ و وزن w_i می‌باشد. هدف پیدا کردن یک مجموعه p عضوی از نقاط A_i ها به گونه‌ای است که تابع هدف زیر کمینه گردد.

$$\min F(X) = \sum_{i=1}^n w_i d(A_i, X)$$

¹⁹Megiddo

²⁰Supowitz

²¹Drezner

²²Gavish

که $X = \{x_1, x_2, \dots, x_p\}$ مجموعه مکان سرویس‌دهنده‌ها است و

$$d(A_i, X) = \min d(A_i, x_j) \quad j = 1, \dots, p$$

به عبارتی در این حالت هدف کمینه کردن مجموع وزنی فاصله‌ها خواهد بود.

مسئله مکانیابی روی شبکه

فرض کنید یک گراف $G = (V, E)$ داده شده که راس V_i دارای وزن w_i و یال (i, j) دارای وزن d_{ij} است. در مسائل مکانیابی روی گراف هدف پیدا کردن مکان سرویس‌دهنده‌ها روی گراف می‌باشد به گونه‌ای که تابع هدف مرتبط با وزن رئوس و فاصله یال‌ها کمینه شود. قضیه زیر که توسط حکیمی [۴۰] ارائه شد، بیان می‌کند که مجموعه جواب‌های p -میانه شامل یک جواب p -میانه راسی است.

قضیه ۱.۴.۱. در شبکه $G = (V, E)$ همواره زیرمجموعه‌ای چون X_p^* از رئوس V وجود دارد که مکان p -میانه شبکه است.

با استفاده از خاصیت بهینگی راسی اگر بخواهیم مسئله را به روش شمارشی حل کنیم برای انتخاب p میانه از بین n راس نیاز به $\binom{n}{p}$ انتخاب داریم. مثلاً برای $p = 1$ به n انتخاب نیاز است که برای هر راس باید $F(x)$ را محاسبه کنیم یعنی به زمان $O(n^2)$ ، نیاز می‌باشد. که در آن $|V| = n$. در حالت کلی برای p های بزرگتر پیچیدگی زمانی این روش زیاد شده و مسئله NP -سخت [۳۲] می‌باشد.

مسئله p -میانه را می‌توان به صورت زیر به یک مسئله برنامه‌ریزی صفر و یک تبدیل کرد. فرض کنید به ازای $i = 1, \dots, n$ و $j = 1, \dots, n$ متغیرهای x_{ij} و y_j را به صورت زیر تعریف کنید.

$$x_{ij} = \begin{cases} 1 & \text{اگر مشتری } i \text{ توسط سرویس‌دهنده } j \text{ سرویس شود.} \\ 0 & \text{در غیر این صورت} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{اگر } A_j \text{ به عنوان سرویس‌دهنده انتخاب شود} \\ 0 & \text{در غیر این صورت} \end{cases}$$

بنابراین مسئله به صورت زیر خواهد بود:

$$\min \sum_{i=1}^n \sum_{j=1}^n w_i d(A_i, A_j) x_{ij}$$

s.t.

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (1.1)$$

$$\sum_{j=1}^n y_j = p \quad (2.1)$$

$$x_{ij} \leq y_j, \quad i, j = 1, \dots, n \quad (3.1)$$

$$x_{ij}, y_j \in \{0, 1\}, \quad i, j = 1, \dots, n \quad (4.1)$$

مجموعه قیود (۱.۱) بیانگر این هستند که هر مشتری فقط و فقط از یک سرویس‌دهنده سرویس می‌گیرد. قید (۲.۱) نشان می‌دهد که تعداد سرویس‌دهنده‌های تاسیس شده باید مساوی p باشد. مجموعه قیود (۳.۱) بیانگر این هستند که هر مشتری تنها می‌تواند از سرویس‌دهنده‌ای سرویس بگیرد که تاسیس شده باشد.

روش‌های تقریبی مختلفی برای حل مسئله p -میان به کار برده شده است که از آن جمله می‌توان روش‌های مبتنی بر برنامه‌ریزی خطی، روش‌هایی با استفاده از گراف‌ها و روش‌های ابتکاری را نام برد.

یکی از روش‌های ابتکاری برای مسئله p -میان که توسط مارنزان^{۲۳} [۵۰] ارائه شده است روش جستجوی همسایگی^{۲۴} می‌باشد. در این روش از یک جواب شدنی برای مسئله، یعنی مجموعه‌ای شامل مکان p وسیله شروع کرده و نقاط تقاضا را به نزدیکترین وسیله نسبت می‌دهیم. مجموعه گره‌های نسبت داده شده به هر وسیله یک همسایگی اطراف آن وسیله تشکیل می‌دهند. در هر همسایگی، مسئله ۱-میان را حل کرده و وسیله را به این مکان جدید تغییر مکان می‌دهیم. سپس هنگامی که برای تمام همسایگی‌ها مکان وسایل دوباره پیدا شدند، همسایگی‌های جدید را به دست آورده و عمل را تکرار می‌کنیم تا زمانی که دیگر تغییری در همسایگی‌ها یا مکان وسایل صورت نگیرد.

براساس روش فوق، تیتز^{۲۵} و بارت^{۲۶} نیز الگوریتمی ارائه کردند [۷۰] که در آن برای تغییر مکان یک وسیله، دیگر راس‌ها به ترتیب بررسی شده و در صورتی که تابع هدف بهبود یافت تغییر مکان انجام می‌گیرد. روند تا زمانی که دیگر بهبودی در جواب حاصل نشود ادامه می‌یابد. با این روش می‌توان یک جواب داده شده را بهبود بخشید. البته نحوه انتخاب مکان‌های جدید و جواب اولیه در سرعت الگوریتم موثر است.

²³Maranzana

²⁴Neighborhood Search

²⁵Teitz

²⁶Bart

هانسن^{۲۷} و ملادنویچ^{۲۸} [۴۳] مسئله p -میان را با روش جستجوی همسایگی متغیر حل کرده‌اند که در آن دو جواب که در k ($k \leq p$) عضو اختلاف دارند به عنوان همسایه در نظر گرفته می‌شوند.

یک روش ابتکاری نیز که به صورت الگوریتمی حریصانه^{۲۹} و براساس روش تغییر مکان وسایل می‌باشد، در [۲۰] آمده است که در آن ابتدا نقطه ۱-میان به دست آمده و در مجموعه جواب قرار می‌گیرد. سپس در هر گام یک راس به این مجموعه اضافه شده و بررسی می‌شود که آیا می‌توان راس دیگری را جایگزین راس‌های انتخاب شده کرد تا تابع هدف کاهش یابد یا خیر. در صورت مثبت بودن جواب، راس را جایگزین کرده و دوباره عمل بررسی انجام می‌شود. پس از p گام یک جواب تقریبی برای مسئله به دست می‌آید.

از دیگر روش‌های ابتکاری که برای مسئله p -میان به کار رفته است روش‌های الگوریتم ژنتیک^{۳۰} و جستجوی ممنوع^{۳۱} است. اولین کسانی که الگوریتم ژنتیک را برای مسئله p -میان به کار بردند، هاسج^{۳۲} و گودچایلد^{۳۳} در [۴۵] بودند. سپس دیبل^{۳۴} و دشام^{۳۵} [۲۳]، ارکوت^{۳۶} و همکاران [۲۹] و کورا^{۳۷} و همکاران [۲۲] روش‌هایی با استفاده از الگوریتم‌های ژنتیک برای مسئله p -میان ارائه کردند.

با استفاده از جستجو تابو نیز مورنو^{۳۸} همکاران در [۵۵] و رلند^{۳۹} و همکاران [۶۵] روش‌هایی برای مسئله p -میان ارائه کرده‌اند.

برای مسئله p -میان روی درخت، کریو^{۴۰} و حکیمی [۴۸] الگوریتم دقیق با پیچیدگی زمانی $O(p^2 n^2)$ ارائه کرده‌اند که آن را تمیر^{۴۱} [۶۹] به الگوریتم با پیچیدگی زمانی $O(p n^2)$ بهبود بخشیده است.

مسئله ۱- میان روی شبکه درختی

فرض کنید $T = (V, E)$ یک درخت با مجموعه رئوس V و مجموعه یال E که $|V| = n$ باشد. ۱-میان درخت T که با μ نشان می‌دهیم، عبارت است از جواب مسئله زیر

$$F(\mu) = \min f(x) = \sum_{i=1}^n w_i d(x, v_i)$$

²⁷Hansen

²⁸Meladenovic

²⁹Greedy

³⁰Genetic Algorithm

³¹Tabu Search

³²Hosage

³³Goodchild

³⁴Dibble

³⁵Densham

³⁶Erkut

³⁷Correa

³⁸Moreno-perez

³⁹Rolland

⁴⁰Kariv

⁴¹Tamir

تعریف ۲.۴.۱. فرض کنید T_1 زیر درختی از T باشد، مجموع وزن رئوس روی T_1 را با $W(T_1)$ نمایش می‌دهیم. به عبارت دیگر

$$W(T_1) = \sum_{v_i \in T_1} w_i \quad (5.1)$$

لم ۱.۴.۱. فرض کنید T_1 و T_2 دو زیر درخت از T باشند به قسمی که $T_1 \cup T_2 = T$ و $T_1 \cap T_2 = \emptyset$. اگر $\mu \in T_1$ اگر و تنها اگر

$$W(T_1) \geq W(T_2) \quad (6.1)$$

خاصیت فوق را که گلدمن ثابت کرده است [۲۶] خاصیت اکثریت وزنی می‌گویند.

لم ۲.۴.۱. اگر $W(T) \leq W(S)$ که T و S دو زیر درخت حاصل از حذف یال (s, t) در درخت اصلی هستند، آنگاه یافتن ۱-میانه متناظر است با یافتن ۱-میانه روی درخت N_S که N_S همان درخت S است با این تفاوت که وزن راس s برابر است با $w(s) + w(T)$

نتیجه ۱.۴.۱. فرض کنید T_1 یک زیر درخت از درخت مفروض T باشد. مکان ۱-میانه درخت T روی زیر درخت T_1 قرار دارد اگر و تنها اگر

$$W(T_1) \geq W(T)/2$$

این نتیجه اساس کار حل مسائل ۱-میانه معکوس روی شبکه‌های درخت می‌باشد. با استفاده از دو لم فوق و نتیجه، الگوریتم زیر که به الگوریتم اکثریت ^{۴۲} (گلدمن) مشهور است برای مسئله ۱-میانه ارائه گردید [۲۶].

الگوریتم گلدمن

● **گام ۱:** اگر درخت T شامل فقط یک راس است، همان راس مکان ۱-میانه است و به اجرای الگوریتم خاتمه دهید.

● **گام ۲:** یک راس انتهایی (برگ) مانند v_i انتخاب کن اگر $w(v_i) \geq W(T)/2$ ، آنگاه v_i مکان ۱-میانه درخت T است و به گام ۴ بروید. در غیر اینصورت به گام ۳ بروید.

● **گام ۳:** فرض کنید v_j راس متصل به v_i باشد، قرار دهید

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

و یال $e = (v_i, v_j)$ را از درخت T حذف کرده و درخت جدید را با T جایگزین کرده و به گام ۱ بروید.

⁴²Majority

● گام ۴: توقف، v_i ۱- میانه است.

واضح است که زمان اجرای الگوریتم برابر با $O(n)$ است.

مسئله ۲- میانه روی شبکه درختی

در مسئله ۲- میانه علاوه بر اینکه مکان سرویس‌دهنده‌ها باید انتخاب شود، نحوه تخصیص مشتری‌ها به سرویس‌دهنده نیز باید مشخص شود. برای یافتن مکان ۲- میانه روی درخت روش‌های مختلف از قبیل روش شمارش و روش حذف یال مطرح شده‌است. بنابراین با توجه به مطالبی که قبلاً گفته شد ملاحظه می‌شود که اگر بخواهیم مسئله ۲- میانه روی شبکه را به روش شمارشی حل کنیم به زمان $O(n^3)$ نیاز داریم. در زیر الگوریتمی ساده برای حل مسئله مکانیابی ۲- میانه روی درخت T مبتنی بر روش حذف یال ارائه می‌گردد.

الگوریتم حذف یال [۴۸]

در روش حذف یال به ازای هر یال مانند (i, j) درخت را به دو زیر درخت T_1 و T_2 افراز می‌کنیم و با استفاده از الگوریتم گلدمن، ۱- میانه را روی هر کدام از زیر درخت‌ها می‌یابیم. مقدار تابع هدف مسئله کلی برابر مجموع تابع هدف مسئله ۱- میانه روی T_1 و T_2 است. حال به ازای تمام یال‌ها این عمل را انجام می‌دهیم. جواب بهینه، کمترین مقدار تابع هدف بدست آمده از بین آنها است.

با توجه به اینکه الگوریتم به ازای هر یال $e \in E$ تکرار می‌شود و در هر تکرار مکان ۱- میانه زیر درخت‌های T_1 و T_2 در زمان $O(n)$ به دست می‌آید. لذا زمان اجرای الگوریتم حذف یال برابر با $O(n^2)$ می‌باشد.

روش فوق را میرچندانی و آدجیت [۵۴] تنها با حذف یال‌های خاصی بهبود بخشیدند.

۲.۴.۱ مسئله مکانیابی p -مرکز

تعریف ۳.۴.۱. مسئله مکانیابی p -مرکز، انتخاب p مکان از بین n مکان موجود به عنوان سرویس‌دهنده است به گونه‌ای که فاصله دورترین نقطه تقاضا تا p سرویس‌دهنده کمینه گردد.

حکیمی در [۴۰] ثابت نمود که خاصیت بهینگی راسی که قبلاً برای مسئله میانه ثابت نموده بود، برای مسئله مرکز برقرار نمی‌باشد. کریو و حکیمی ثابت کردند که مسئله p -مرکز روی شبکه‌های کلی NP -سخت است، حتی اگر شبکه مورد نظر مسطح غیروزن‌دار با طول‌های یالی واحد و ماکزیمم درجه راسی آن ۳ باشد [۴۸] کریو و حکیمی یک الگوریتم شمارشی با زمان اجرای $O(m^p n^{2p-1} \log n)$ برای شبکه‌های وزن‌دار ارائه کردند که در آن $|V| = n$ و $|E| = m$.

برای حالت غیروزن دار الگوریتمی با زمان اجرای $O(m^p n^{2p-1})$ ارائه کردند. در ادامه، تامیر مسئله p -مرکز روی شبکه‌های کلی را در زمان بهبود یافته $O(m^p n^p \log^2 n)$ برای حالت وزن دار و $O(m^p n^p \log n)$ برای حالت بدون وزن حل نمود [۶۸]. برای مسئله p -مرکز روی شبکه‌های درختی غیر وزن دار با $p > 1$ ، حکیمی الگوریتمی با زمان اجرای $O(n^{p-1})$ ارائه داد [۴۸].

مدل ریاضی مسئله p -مرکز

فرض کنید n نقطه به عنوان مکان استقرار تقاضاها در صفحه موجود است. هر نقطه A_i دارای مختصات $A_i = (a_i, b_i)$ و وزن w_i است. می‌خواهیم مکان استقرار p سرویس‌دهنده را در صفحه به گونه‌ای تعیین کنیم تا فاصله دورترین نقطه تقاضا تا p سرویس‌دهنده کمینه گردد.

$$\min_{i=1, \dots, n} \max w_i d(A_i, X)$$

که $X = \{x_1, x_2, \dots, x_p\}$ مجموعه مکان سرویس‌دهنده‌ها است و

$$d(A_i, X) = \min d(A_i, x_j), \quad j = 1, \dots, p$$

به‌طور مشابه با مدل‌بندی مسئله p -میانه، می‌توان مسئله p -مرکز را به صورت یک مسئله برنامه‌ریزی خطی صحیح مدل‌بندی کرد. برای این کار متغیر Z را به صورت

$$Z = \max_{i=1, \dots, n} w_i d_{ij} x_{ij}$$

در نظر می‌گیریم به ازای $i = 1, \dots, n$ و $j = 1, \dots, n$ متغیرهای x_{ij} و y_j را به صورت زیر تعریف می‌کنیم.

$$x_{ij} = \begin{cases} 1 & \text{اگر مشتری } i \text{ توسط سرویس‌دهنده } j \text{ سرویس شود.} \\ 0 & \text{در غیر این صورت} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{اگر } A_j \text{ به عنوان سرویس‌دهنده انتخاب شود} \\ 0 & \text{در غیر این صورت} \end{cases}$$

بنابراین مدل مسئله به صورت زیر خواهد بود:

$$\begin{aligned}
& \min Z \\
& s.t. \\
& \sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \\
& \sum_{j=1}^n y_j = p \\
& w_i d_{ij} x_{ij} \leq Z, \quad i, j = 1, \dots, n \\
& x_{ij} \leq y_j, \quad i, j = 1, \dots, n \\
& x_{ij}, y_j \in \{0, 1\}, \quad i, j = 1, \dots, n
\end{aligned}$$

۵.۱ مسئله مکانیابی متعادل

یکی از موضوعاتی که اخیراً در مکانیابی مطرح شده است، بحث عدالت یا تعادل در مکانیابی است. این نوع مسائل مکانیابی با قرار دادن سرویس‌دهنده‌ها به طوری که برابری در سرویس‌دهی به نقاط تقاضا حداکثر باشد، سروکار دارند. در ابتدای این بخش ابتدا مروری مختصر بر تاریخچه مسائل مکانیابی متعادل می‌کنیم و سپس بخش‌هایی از مقالات [۵۱] و [۹] به اختصار توضیح می‌دهیم.

در برخی از پژوهش‌ها عدالت برای مشتری‌ها مطرح است که اولین مقاله ارائه شده در این باره، مقاله ایرکوت^{۴۳} است که عدالت در فاصله برای مشتری‌ها مورد مطالعه قرار داد و یک چارچوب برای تعادل بخشیدن به نابرابری‌ها ارائه داد [۲۸]. بعد از آن مقالات زیادی در این رابطه منتشر شد. در پاره‌ای از پژوهش‌ها نیز عدالت را برای سرویس‌دهنده‌ها مورد بررسی قرار دادند. یکی از این مقالات، مقاله برمن^{۴۴} و همکاران است که یک جواب عادلانه را از منظر سرویس‌دهنده‌ها بررسی می‌کند بدین صورت که مقدار بار تخصیص یافته به هر سرویس‌دهنده متعادل گردد که بعداً به آن خواهیم پرداخت [۹]. اما دسته دیگر از مقالات مرتبط با عدالت نیز مطرح شد که در آن هم عدالت برای مشتری‌ها رعایت می‌شود و هم عدالت برای سرویس‌دهنده‌ها مد نظر قرار می‌گیرد. در مقاله‌ای که توسط مارین^{۴۵} مطرح شد، این موضوع مورد بحث قرار گرفته است [۵۱]. مکانیابی متعادل کاربردهای زیادی دارد از جمله در مکانیابی آنتن‌ها برای ارتباطات تلفن همراه یا در طراحی منطقه، جایی که مناطق جغرافیایی کوچک باشد، توزیع عادلانه تجهیزات و نیروی کار در مراکز خدماتی در کشورهای توسعه نیافته و ...

⁴³Erkut

⁴⁴Berman

⁴⁵Marín

فرض کنید $B = \{1, \dots, n\}$ و $A = \{1, \dots, m\}$ به ترتیب مجموعه‌های ارائه شده برای مشتری‌ها و مجموعه کاندید برای استقرار p سرویس‌دهنده باشد و فرض کنید $C = (c_{ij})_{i \in B, j \in A}$ ماتریس هزینه‌ها (هزینه‌های حمل و نقل، فاصله) باشد. هیچ محدودیت اضافی در ماتریس C نیست. هزینه‌ها می‌توانند منفی باشند) جواب مسئله مکانیابی متعادل به صورت یک مجموعه $X \subset A$ است که $|X| = p$ به طوری که هر مشتری که نزدیکترین سرویس‌دهنده تخصیص می‌یابد. یعنی فرض می‌کنیم که هر مشتری $i \in B$ از سرویس‌دهنده $j \in X$ سرویس دریافت می‌کند، به قسمی که $c_{ij} = \min_{k \in X} \{c_{ik}\}$. هدف، مینیمم کردن اختلاف بین مینیمم و ماکزیمم تعداد مشتری‌های تخصیص داده شده به هر سرویس‌دهنده در X است.

جدول ۱.۱: جواب مسئله مکانیابی متعادل

مقدار تابع هدف	(l, u)	1°	۴	۳	$\circ$	(x_1, x_2)
۲	$(1, 3)$	۲	۲	۲	۱	$(1, 2)$
۲	$(1, 3)$	۳	۳	۳	۱	$(1, 3)$
۲	$(1, 3)$	۴	۱	۱	۱	$(1, 4)$
$\circ$	$(2, 2)$	۳	۳	۲	۲	$(2, 3)$
۲	$(1, 3)$	۴	۲	۲	۲	$(2, 4)$
۲	$(1, 3)$	۴	۳	۳	۳	$(3, 4)$

مثال ۱.۵.۱. فرض کنید مجموعه $\{0, 3, 4, 1^\circ\}$ ، ۴ نقطه روی محور حقیقی متناظر با مجموعه مشتریان و مکان‌های کاندید برای استقرار سرویس‌دهنده‌ها باشد. فرض کنید c_{ij} فاصله اقلیدسی بین مشتری i و مکان j و $p = 2$ تعداد سرویس‌دهنده‌ها است. در این مثال ۶ حالت مختلف برای X وجود دارد که در جدول ۱.۱ نشان داده شده‌است. در جدول بیشترین و کمترین مشتری تخصیص داده شده به x_1, x_2 با l, u نشان داده شده‌است. در ۵ تا از این انتخاب‌ها، یک مشتری به یک سرویس‌دهنده اختصاص یافته و ۳ مشتری به سرویس‌دهنده دیگر تخصیص می‌یابد. مقدار تابع هدف برای این حالات $2 = 3 - 1$ است. در حالت دیگر یعنی $(2, 3)$ ، مقدار تابع هدف $\circ$ می‌شود. زیرا به هر سرویس‌دهنده ۲ مشتری تخصیص داده می‌شود و جواب کاملاً در حالت تعادل است.

مثال ۲.۵.۱. حالتی را در نظر بگیرید که $n = m = 3$ و ماتریس هزینه به صورت زیر است.

$$C = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

با توجه به تعداد سرویس‌دهنده‌ها، همه مشتری‌ها به یک سرویس‌دهنده اختصاص می‌یابد. به عبارتی دیگر اگر $p \geq 2$ ، یک سرویس‌دهنده به همه مشتری‌ها سرویس می‌دهد و باقی سرویس‌دهنده‌ها به هیچ مشتری سرویس نمی‌دهند. جواب بهینه برای این مثال، ۳ خواهد بود و جواب بهینه کاملاً نامتعادل است.

در ادامه یک فرمول برنامه‌ریزی عدد صحیح برای این نوع مسئله مکانیابی متعادل ارائه می‌شود. این فرمول استاندارد نامیده می‌شود زیرا از فرمول استاندارد مسئله p -میان به علاوه دو متغیر دیگر ساخته می‌شود (مینیمم و ماکزیمم مشتری تخصیص داده شده به هر سرویس‌دهنده). لذا مدل بندی ریاضی آن به صورت زیر می‌باشد.

$$x_{ij} = \begin{cases} 1 & \text{اگر مشتری } i \text{ توسط سرویس‌دهنده } j \text{ سرویس شود.} \\ 0 & \text{در غیر این صورت} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{اگر } A_j \text{ به عنوان سرویس‌دهنده انتخاب شود.} \\ 0 & \text{در غیر این صورت} \end{cases}$$

$$\min \quad u - l$$

s.t.

$$\sum_{j=1}^m y_j = p \quad (1.1)$$

$$\sum_{j=1}^m x_{ij} = 1 \quad \forall i \in B \quad (2.1)$$

$$x_{ij} \leq y_j \quad \forall i \in B, j \in A \quad (3.1)$$

$$u \geq \sum_{i=1}^n x_{ij} \quad \forall j \in A \quad (4.1)$$

$$l \leq \sum_{i=1}^n x_{ij} + n(1 - y_j) \quad \forall j \in A \quad (5.1)$$

$$\sum_{a \in A} c_{ia} x_{ia} + (M_i - c_{ij}) y_j \leq M_i \quad \forall i \in B, j \in A \quad (6.1)$$

$$x_{ij}, y_j \in \{0, 1\} \quad \forall i \in B, j \in A$$

که در مدل بالا $M_i = \max_{j \in A} \{c_{ij}\}$, $\forall i \in B$

محدودیت (۱.۱) تا (۳.۱) همان محدودیت‌های مدل p -میان است. محدودیت (۴.۱) و (۵.۱) در ارتباط با تابع هدف هستند. این محدودیت‌ها مقادیر قابل قبولی برای l, u بدست می‌آورند. محدودیت‌های (۱.۱) تا (۵.۱) نمی‌توانند تخصیص مشتری به نزدیکترین سرویس‌دهنده

را تضمین کنند برای ایجاد این شرایط محدودیت (۶.۱) اضافه شده است. با محدودیت (۶.۱) هر مشتری به نزدیکترین سرویس دهنده اختصاص می‌یابد.

در مقاله‌ای که توسط برمن و همکاران در سال ۲۰۰۸ ارائه شده [۹]، مسئله در نظر گرفته شده است که در آن، هدف پیدا کردن p مکان برای p سرویس دهنده به گونه‌ای است که وزن‌های تخصیص داده شده به هر سرویس دهنده تا حد امکان به یکدیگر نزدیک باشند. این مسئله به صورت زیر مدل‌بندی می‌شود.

فرض کنید $G = (V, E)$ یک گراف بدون جهت و همبند با مجموعه رئوس $V = \{v_1, v_2, \dots, v_n\}$ و مجموعه یال‌های E است. هر یال دارای طول مثبت و هر راس $v_i \in V$ دارای وزن تقاضای نامنفی w_i است. برای هر جفت از راس‌های v_j, v_i فرض کنید $d_{ij} = d(v_i, v_j)$ طول کوتاه‌ترین مسیر بین v_j و v_i است.

برای هر راس v_i و $\hat{V} \subset V$

$$d(v_i, \hat{V}) = d(\hat{V}, v_i) = \min_{v_j \in \hat{V}} \{d(v_j, v_i)\}$$

فرض کنید $S \subset V$ یک زیر مجموعه شامل p راس باشد. برای هر راس v_i فرض کنید

$$S^i = \{v_j \in S : d(v_j, v_i) = d(v_i, S)\}$$

یک تخصیص $-S$ شدنی، (π) ، یک تخصیصی است که در آن هر راس v_i به تعدادی راس در S^i تخصیص یابد.

اگر v_i به S^i $v_j \in S^i$ تخصیص یابد، داریم $\pi(i) = j$ و بار نهایی (وزن) تخصیص داده شده به v_j به صورت زیر تعریف می‌شود.

$$L_S^\pi(j) = \sum_{i: \pi(i)=j} w_i.$$

حداکثر بار π به صورت زیر تعریف می‌شود.

$$L_S^\pi = \max_{v_j \in S} \{L_S^\pi(j)\}.$$

یک تخصیص $-S$ شدنی π^* ، یک تخصیص $-S$ بهینه است اگر آن دارای کوچکترین ماکزیمم بار در میان تخصیص‌های S شدنی باشد و با L_S^* نشان می‌دهند.

برای این مسئله دو حالت در نظر گرفته شده است:

۱. حالت گسسته (P_1) : هدف پیدا کردن $S \subset V$ و $|S| = p$ به طوری که مینیمم ماکزیمم وزنی L_S^* باشد.

۲. حالت پیوسته (P_2) : تفاوتی که حالت پیوسته با حالت گسسته دارد این است که در حالت پیوسته تقاضای w_i راس v_i می‌تواند در میان راس‌های S^i توزیع شود و هدف پیدا کردن $S \subset V$ و $|S| = p$ به طوری که مینیمم ماکزیمم وزنی L_S^* باشد.

مسئله بالا به صورت برنامه‌ریزی صحیح فرمول‌بندی می‌شود. ابتدا در حالت گسسته :

$$x_{ij} = \begin{cases} 1 & \text{اگر } v_i \text{ به سرویس دهنده } j \text{ ام تخصیص یابد} \\ 0 & \text{در غیر این صورت} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{اگر سرویس دهنده در مکان } j \text{ باشد.} \\ 0 & \text{در غیر این صورت} \end{cases}$$

$$P_1 \min\{z\}$$

s.t.

$$\sum_{i=1}^n x_{ij} w_i \leq z \quad j = 1, \dots, n \quad (7.1)$$

$$\sum_{i=1}^n y_j = p \quad (8.1)$$

$$x_{ij} \leq y_j \quad i, j = 1, \dots, n \quad (9.1)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (10.1)$$

$$\sum_{k=1}^n d_{ik} x_{ik} + (F - d_{ij}) y_j \leq F \quad i, j = 1, \dots, n \quad (11.1)$$

$$x_{ij}, y_j = \{0, 1\} \quad i, j = 1, \dots, n \quad (12.1)$$

که F یک عدد خیلی بزرگ است که $F \geq \max_{i,j} \{d_{ij}\}$ برای حالت پیوسته (P_2) مدل مسئله دقیقاً مانند P_1 است فقط به جای محدودیت $y_{ij} \in \{0, 1\}$ محدودیت $y_{ij} \geq 0$ می‌نویسیم.

تابع هدف و محدودیت (۷.۱) شرط مینی ماکس را تضمین می‌کند. محدودیت (۸.۱) تا (۱۰.۱) همان محدودیت‌های مسئله p -میان است. محدودیت (۱۱.۱) مانند مدل قبل (برای مسئله‌های P_1 و P_2) هر مشتری را به نزدیکترین سرویس‌دهنده اختصاص می‌دهد. برمن و همکارانش [۹] در این مقاله نشان دادند که مسئله P_1 روی درخت NP -سخت است و برای مسئله P_2 و $p = 2$ یک الگوریتمی با پیچیدگی زمانی $O(n \log n)$ ارائه دادند.

فصل ۲

مروری بر مسائل مکانیابی معکوس

۱.۲ مقدمه

همانطور که بیان شد در یک مسئله مکانیابی کلاسیک با فرض معلوم بودن پارامترهای ورودی مسئله مانند هزینه‌ها، ظرفیت‌ها، مقدار تقاضای مشتریان و ... به دنبال یافتن یک مجموعه جواب بهین هستیم. اما برخلاف مسائل بهینه‌سازی کلاسیک، در یک مسئله معکوس یک جواب شدنی مسئله بهینه‌سازی تحت مطالعه مفروض است. هدف تغییر پارامترهای مسئله با صرف کمترین هزینه است به طوری که مکان سرویس‌دهنده‌های از پیش تعیین شده با پارامترهای جدید بهینه شوند و یا تا حد ممکن بهبود یابند. مسائل مکانیابی معکوس دارای کاربردهای فراوانی در مسائل عملی هستند.

برای مثال یک مرکز خرید در یک سیستم شهری را در نظر بگیرید. با گذر زمان ممکن است اقبال مردم (مشتریان) برای مراجعه به این مرکز خرید به دلایل مختلف (دور بودن آن از محل سکونت شهروندان، بالا بودن قیمت اجناس، پایین بودن کیفیت کالاهای عرضه شده و ...) کاهش پیدا کرده و مکان فعلی آن از بهینگی خارج شده (کاهش سود حاصل از فروش محصولات) در حالی که انتقال آن به مکان دیگر مقرون به صرفه نیست. در چنین حالتی یک راه‌حل موثر اصلاح برخی از پارامترهای تاثیرگذار از قبیل ایجاد تسهیلات در بخش حمل و نقل برای سهولت دسترسی مشتریان به مرکز خرید، ایجاد سیاست‌های تشویقی مانند تخفیف‌های مناسب برای کالاهای عرضه شده، افزایش کیفیت محصولات و ... با صرف کمترین هزینه

می‌باشد به‌طوری‌که مکان فعلی این مرکز خرید با توجه به پارامترهای جدید بهینه شود و یا تا حد ممکن بهبود یابد.

یا فرض کنید یک شبکه جاده‌ای و یک سرویس‌دهنده (مانند مرکز اورژانس) و تعدادی مشتری (مانند بیمارستان‌ها) در آن مفروض باشد. هدف این است که پارامترهای شبکه مانند طول جاده‌ها را با کمترین هزینه طوری اصلاح کنیم که مجموع فاصله سرویس‌دهنده تا مشتری‌ها مینیمم شود. این مسئله یک مسئله ۱-میانه معکوس است که در بخش بعدی مورد بررسی قرار می‌گیرد. در این فصل، ابتدا یک بیان کلی از مسائل مکانیابی معکوس را ارائه می‌کنیم. سپس انواع مسائل مکانیابی معکوس و خلاصه‌ای از کارهای مهم صورت گرفته مربوط به هر نوع معرفی می‌شوند.

در حوزه بهینه‌سازی معکوس در سال ۱۹۹۲ بورتین^۱ و توینت^۲ اولین کار تحقیقی برای مدل‌های کوتاه‌ترین مسیر معکوس ارائه کردند [۱۷]، [۱۸]. از آن زمان تاکنون مطالعات متعددی روی مدل‌های معکوس مسائل بهینه‌سازی، به‌ویژه مدل‌های مکانیابی معکوس انجام گرفته است. برای مطالعات بیشتر در این زمینه می‌توان به [۴۴]، [۶]، [۱۴] مراجعه کرد.

۲.۲ مسائل مکانیابی اصلاحی

در تقابل با مدل‌های کلاسیک ارائه شده، در سال‌های اخیر، مسائل مکانیابی اصلاحی از جمله مدل‌های معکوس و مدل‌های ترفیع/تنزیل بسیار مورد توجه قرار گرفته اند. در این بخش، ابتدا یک بیان کلی از مسئله مکانیابی اصلاحی ارائه می‌گردد. سپس انواع مسائل مکانیابی اصلاحی معرفی می‌شود و در آخر، رویکردهای جواب ارائه شده برای آنها مورد بررسی قرار می‌گیرد. در یک مدل مکانیابی معکوس، هدف اصلاح برخی پارامترهای ورودی سیستم (از قبیل طول‌های یالی یا وزن‌های راسی گراف، وزن‌ها یا مختصات نقاط روی فضای حقیقی و ...) با صرف کمترین هزینه است به طوری که مکان تسهیلات از قبل تعیین شده تحت پارامترهای جدید بهین شده یا تا حد امکان بهبود یابد. همچنین در یک مدل مکانیابی ترفیع/تنزیل ابتدا برخی از پارامترهای ورودی سیستم تحت یک بودجه معین اصلاح می‌شوند و سپس تسهیلات تحت پارامترهای جدید مکانیابی می‌شوند به طوری که مقدار هدف متناظر تا حد ممکن بهبود یابد/بدتر شود.

فرض کنید (X, d) یک فضای متریک با تابع فاصله زیر باشد.

$$d : X \rightarrow R^+$$

¹Burton

²Toint

این فضای متریک می تواند یک شبکه یا یک فضای حقیقی R^k باشد. همچنین فرض کنید (LOC) یک مسئله مکانیابی روی این فضای متریک باشد و $\theta \in R^n$ بردار پارامترهای مسئله (LOC) باشد. به علاوه S مجموعه جواب های شدنی مسئله (LOC) تحت بردار پارامتر θ بوده و $f(\theta, s)$ تابع هدف مسئله (LOC) برای جواب شدنی $s \in S$ باشد. با این مفروضات می توان مسئله مکانیابی (LOC) را به صورت زیر نوشت:

$$(LOC) \quad \min f(\theta, s) \\ s.t. \quad s \in S$$

به علاوه فرض کنید $s^* \in S$ یک جواب از پیش تعیین شده برای مسئله مکانیابی (LOC) باشد. هدف مسئله مکانیابی معکوس عبارت است از اصلاح بردار پارامتر θ به $\tilde{\theta}$ با توجه به برخی محدودیت ها به ارزان ترین روش ممکن (پرسودترین روش ممکن) به طوری که جواب شدنی مفروض s^* برای مسئله (LOC) تحت بردار پارامتر جدید $\tilde{\theta}$ بهینه شده یا تا حد ممکن بهبود یابد. هدف مسائل مکانیابی (ترفع، تنزیل) متناظر مسئله مکانیابی کلاسیک (LOC) تغییر بردار پارامتر θ به $\tilde{\theta}$ با توجه به محدودیت بودجه معین و سپس یافتن یک جواب بهینه برای مسئله (LOC) تحت بردار پارامتر جدید $\tilde{\theta}$ است به طوری که مقدار هدف متناظر تا حد ممکن بهتر (بدتر) شود. فرض کنید c_i^+ هزینه متحمل شده $(p_i^+ \text{ سود حاصله})$ از افزایش i امین مولفه θ_i از بردار θ به اندازه یک واحد بوده و c_i^- هزینه متحمل شده $(p_i^- \text{ سود حاصله})$ از کاهش θ به اندازه یک واحد باشد. همچنین فرض کنید $g(c, \tilde{\theta})$ تابع هزینه $g(p, \tilde{\theta})$ تابع سود اصلاح بردار پارامتر θ به $\tilde{\theta}$ با بردار هزینه c (بردار سود p) باشد. با توجه به این که بردار θ ممکن است قادر به اصلاح به طور دلخواه نباشد، از این رو فرض کنید β مجموعه محدودیت هایی باشد که بردار پارامتر جدید $\tilde{\theta}$ باید در آنها صدق کند. تابع هزینه $g(c, \tilde{\theta})$ تحت نرم های مختلفی مورد ارزیابی قرار می گیرد. برای ارائه یک رویکرد کلی فرض کنید پارامتر τ بصورت $\tau = c$ برای تابع هزینه و $\tau = p$ برای تابع سود تعریف شود. برخی از رایج ترین توابع هزینه به کار برده شده در مسائل مکانیابی اصلاحی به شرح زیر است:

• تابع هزینه $g(\tau, \tilde{\theta})$ تحت نرم خطی:

$$g(\tau, \tilde{\theta}) = \sum_{i=1}^n (\tau_i^+ \max\{0, \tilde{\theta}_i - \theta_i\} + \tau_i^- \max\{0, \theta_i - \tilde{\theta}_i\})$$

• تابع هزینه $g(\tau, \tilde{\theta})$ تحت نرم چپیشف:

$$g(\tau, \tilde{\theta}) = \max_{i=1, \dots, n} \max\{\tau_i^+ \max\{0, \tilde{\theta}_i - \theta_i\}, \tau_i^- \max\{0, \theta_i - \tilde{\theta}_i\}\}$$

- تابع هزینه $g(\tau, \tilde{\theta})$ تحت فاصله همینگ جمعی:

$$g(\tau, \tilde{\theta}) = \sum_{i=1}^n (\tau_i^+ H(\tilde{\theta}_i - \theta_i) + \tau_i^- H(\theta_i - \tilde{\theta}_i))$$

- تابع هزینه $g(\tau, \tilde{\theta})$ تحت فاصله همینگ تنگنا:

$$g(\tau, \tilde{\theta}) = \max_{i=1, \dots, n} \max\{\tau_i^+ H(\tilde{\theta}_i - \theta_i), \tau_i^- H(\theta_i - \tilde{\theta}_i)\}$$

به طوری که فاصله همینگ به صورت زیر تعریف می شود:

$$H(\nu) = \begin{cases} 1 & \nu > 0 \\ 0 & \nu \leq 0 \end{cases}$$

و به علاوه مجموعه β در حالت کلی می تواند به صورت زیر تعریف شود.

$$\beta = \{\tilde{\theta}, u^- \leq \tilde{\theta} \leq u^+, u^+, u^- \in R^n\}$$

که در آن بردار پارامترهای اصلاح شده باید در محدودیت کران بالا u^+ و کران پایین u^- صدق کند. برخی از مدل های مکانیابی معکوس عبارتند از

- مدل مکانیابی معکوس با حداقل هزینه^۳
- مدل مکانیابی معکوس با محدودیت بودجه ای^۴
- مدل مکانیابی معکوس با تابع هدف- کران دار^۵
- مدل مکانیابی ترفیع و تنزیل^۶

۱.۲.۲ مدل مکانیابی معکوس با حداقل هزینه

برای مسئله مکانیابی مفروض (LOC) در فضای متریک (X, d) با بردار پارامتر θ ، مجموعه جواب های شدنی S ، تابع هدف $f(\theta, s)$ و یک جواب شدنی مفروض $s^* \in S$ ، مسئله مکانیابی معکوس با حداقل هزینه متناظر به صورت زیر فرمول بندی می گردد:

$$\begin{aligned} \min \quad & g(c, \tilde{\theta}) \\ \text{s.t.} \quad & f(\tilde{\theta}, s^*) \leq f(\tilde{\theta}, s) \quad \forall s \in S \\ & \tilde{\theta} \in \beta \end{aligned}$$

³minimum cost inverse location problem

⁴Reverse location problem or Budget-Constrained Inverse location problem

⁵Objective-Bounded Inverse location problem

⁶Up-and Downgrading location problem

که در آن $g(c, \bar{\theta})$ تابع هزینه مربوط به تغییر بردار پارامتری از θ به $\bar{\theta}$ بوده و β مجموعه محدودیت‌هایی است که بردار اصلاح شده $\bar{\theta}$ باید در آنها صدق کند. این مسئله بسته به نحوه تعریف تابع هزینه $g(c, \bar{\theta})$ و نوع مسئله مکانیابی کلاسیک (LOC) یک مسئله بهینه‌سازی خطی یا غیرخطی است.

دو مدل از مسائل مکانیابی معروف، مسائل p -میانه و p -مرکز است که قبلاً معرفی شدند. در این مسائل مکانیابی سرویس‌دهنده‌ها به وزن مشتریان و فاصله بین مشتریان و سرویس‌دهندگان وابسته می‌باشند.

برای مسائل مکانیابی در گراف‌ها، موقعیت مشتریان بر روی رئوس گراف منطبق شده است و فاصله بین دو نقطه، در واقع کوتاه‌ترین مسیر بین آن دو نقطه در گراف به نسبت طول یال‌ها داده شده می‌باشد و در مورد مسائل مکانیابی در صفحه، فاصله معمولاً به وسیله نرم l_1 یا l_2 یا l_∞ اندازه‌گیری می‌شود.

بنابراین اگر یک نگرش معکوس را در نظر بگیریم، در مورد مسائل روی گراف‌ها، پارامترهایی که در این بین می‌توانند اصلاح شوند، وزن مشتریان یا طول یال‌ها (و یا هر دو) و در مورد مسائل روی صفحه، وزن مشتریان یا موقعیت‌ها (و یا هر دو) می‌باشد.

همانطور که بیان شد، در مسئله p -میانه معکوس روی گراف‌ها، هدف اصلاح کردن پارامترهای مسئله (وزن مشتریان و یا طول یال‌ها) می‌باشد که p راس از پیش تعیین شده، p -میانه مسئله شوند.

۱- مسئله p -میانه معکوس با تغییر وزن رئوس

فرض کنید (X, d) یک فضای متریک با تابع متر d باشد و n نقطه مشتری $p_1, p_2, \dots, p_n$ که به ترتیب دارای وزن‌های $w_1, w_2, \dots, w_n$ می‌باشند، داده شده باشد (که در مورد کلاسیک، وزن‌های w_i ، $i = 1, 2, \dots, n$ مثبت هستند). در فضای متریک X ، نقطه p ، $s_1, s_2, \dots, s_p$ را مشخص می‌کنیم و هدف این است که با اصلاح وزن مشتریان، مجموعه این نقاط از پیش تعیین شده، p -میانه مسئله شود.

فرض کنید که اگر وزن w_i را به اندازه یک واحد افزایش دهیم، متحمل هزینه نامنفی c_i^+ می‌شویم و به طور مشابه اگر وزن w_i را به اندازه یک واحد کاهش دهیم، متحمل هزینه نامنفی c_i^- می‌شویم و البته توجه داریم که این افزایش و کاهش وزن مشتریان نمی‌تواند به طور دلخواه انجام شود و وزن مشتریان کراندار می‌باشد و از یک کران بالا و یک کران پایین تبعیت می‌کند. پس باید این اصلاح وزن مشتریان طوری صورت بگیرد که وزن اصلاح شده در بین کران‌های داده شده قرار داشته باشد. بنابراین مسئله p -میانه معکوس با تغییر وزن رئوس به صورت زیر بیان می‌کنیم:

مسئله عبارت است از پیدا کردن وزن‌های جدید w_i^* ، $i = 1 \leq n$ که در کران‌های داده شده قرار دارند، یعنی: $\underline{w}_i \leq w_i^* \leq \bar{w}_i$ به گونه‌ای که مجموعه نقاط از پیش تعیین شده $\{s_1, s_2, \dots, s_p\}$ به نسبت وزن‌های جدید یک p -میانه مسئله شوند و هزینه کل برای تغییر

وزن‌ها مینیمم شود. در مورد مسائل گسسته (مثلا گراف‌ها) می‌توان مسئله p -میانه معکوس را به وسیله یک برنامه ریزی خطی توصیف کنیم.

فرض کنید $U = \{u_1, u_2, \dots, u_n\}$ یک مجموعه متناهی باشد که عناصر آن مشتری‌ها باشند. هر مشتری u_i دارای وزن w_i می‌باشد. به علاوه، فرض کنید که مجموعه $V = \{v_1, v_2, \dots, v_m\}$ یک مجموعه متناهی شامل تمام سرویس‌دهندگان ممکن باشد و $d(u, v)$ فاصله بین مشتری u و سرویس‌دهنده v باشد. حال اگر φ مجموعه تمام زیر مجموعه‌های p عضوی از V باشد. شرط لازم و کافی برای اینکه مجموعه $\{s_1, s_2, \dots, s_p\}$ به نسبت وزن‌های w_i , $i = 1, \dots, n$ یک p -میانه مسئله شود این است که:

$$\sum_{i=1}^n w_i \min_{k=1, \dots, p} d(u_i, s_k) \leq \sum_{i=1}^n w_i \min_{v_k \in S} d(u_i, v_k), \forall S \in \varphi$$

حال فرض کنید مقداری که وزن مشتری i ام یعنی w_i افزایش می‌یابد برابر p_i و به طور مشابه مقداری که وزن مشتری i ام کاهش می‌یابد برابر q_i باشد آنگاه مسئله p -میانه معکوس به صورت زیر است.

$$\min \sum_{i=1}^n (c_i p_i + d_i q_i)$$

s.t.

$$\sum_{i=1}^n (w_i + p_i - q_i) a_{iS} \leq 0 \quad \forall S \in \varphi \quad (1.2)$$

$$w_i - q_i \geq \underline{w}_i \quad i = 1, \dots, n \quad (2.2)$$

$$w_i + p_i \leq \overline{w}_i \quad i = 1, \dots, n \quad (3.2)$$

$$p_i \geq 0, \quad q_i \geq 0 \quad i = 1, \dots, n \quad (4.2)$$

که در مدل فوق

$$a_{iS} = \min(d(u_i, s_1), \dots, d(u_i, s_p)) - \min_{v_k \in S} d(u_i, v_k)$$

در مدل برنامه‌ریزی فوق، عبارت (۱.۲) در واقع بیان می‌کند که اگر وزن‌ها اصلاح شوند، یعنی وزن راس i ام به $w_i + p_i - q_i$ تغییر پیدا کند، به نسبت این وزن‌های جدید، مجموعه از پیش تعیین شده $\{s_1, s_2, \dots, s_p\}$ به عنوان p -میانه مسئله معرفی می‌شوند. یعنی:

$$\sum_{i=1}^n (w_i + p_i - q_i) \min_{k=1, \dots, p} d(u_i, s_k) \leq \sum_{i=1}^n (w_i + p_i - q_i) \min_{v_k \in S} d(u_i, v_k)$$

که این عبارت در واقع همان رابطه (۱.۲) است. همچنین چون وزن‌ها کراندارند، بعد از اصلاح وزن‌ها باید وزن‌های جدید از این کران‌ها تبعیت کنند، پس داریم:

$$\underline{w}_i \leq w_i - q_i \leq \overline{w}_i \quad \underline{w}_i \leq w_i + p_i \leq \overline{w}_i$$

از روابط فوق به راحتی عبارت (۲.۲) و (۳.۲) نتیجه می‌شود. بنابراین مسائل p -میانه معکوس گسسته مشروط بر اینکه p معلوم باشد و یک پارامتر ورودی نباشد، در زمان چند جمله‌ای قابل حل می‌باشد. با توجه به برنامه‌ریزی فوق، مدل $O\left(\binom{n}{p}\right)$ محدودیت دارد و بنابراین برای مسائل p -میانه نسبتاً کوچک قابل استفاده می‌باشد.

توجه داشته باشید که با فرض نامنفی بودن d_i و c_i ، در یکی از جواب‌های بهینه، یکی از مقادیر q_i و p_i برابر صفر می‌باشد. با توجه به ملاحظات بالا، مدل فوق می‌تواند برای زمانی که وزن مشتریان منفی، یا به صورت مثبت و منفی (آمیخته) و یا مسئله p -میانه ناخوشایند معکوس به کار گرفته شود، در زمان چند جمله‌ای قابل حل باشد. به طور خلاصه داریم:

گزاره ۱.۲.۲. [۱۳]. مسئله p -میانه معکوس گسسته، که مشتریان ممکن است هر وزن حقیقی داشته باشند، مشروط بر اینکه p ثابت باشد و یک پارامتر ورودی نباشد، در زمان چند جمله‌ای قابل حل می‌باشد.

۲- مسئله p -میانه معکوس با طول یال‌های متغیر

فرض کنید $G = (V, E)$ یک گراف همبند با مجموعه رئوس $V = \{v_1, v_2, \dots, v_n\}$ و مجموعه یال‌های E باشد که $|E| = m$ و یک ثابت p داده شده باشد ($p \leq n$). هر یال $e \in E$ دارای یک طول مثبت l_e می‌باشد و $d_l(v_i, v_j)$ مشخص کننده طول کوتاه‌ترین مسیر از $v_i \in V$ به $v_j \in V$ در گراف G نسبت به طول بردار l باشد. به علاوه فرض کنید که هر راس $v \in V$ دارای یک وزن راسی مثبت $w(v)$ باشد. در مسئله p -میانه کلاسیک هدف مکانیابی p سرویس‌دهنده می‌باشد (یعنی روی رئوس و یا یال‌ها) که مجموع فاصله وزنی از هر راس به نزدیکترین سرویس‌دهنده مینیمم شود که جواب بهینه آن یک p -میانه نامیده می‌شود. در قضیه (۱.۴.۱) بیان شد که یک جواب بهینه میان مجموعه رئوس قرار دارد، که این خاصیت، خاصیت بهینگی راسی نامیده می‌شود.

هدف در مسئله p -میانه معکوس با طول یال‌های متغیر در واقع اصلاح طول یال‌ها با مینیمم هزینه می‌باشد. که مجموعه رئوس از پیش تعیین شده $\{s_1, s_2, \dots, s_p\}$ یک p -میانه مسئله شود. فرض کنید اگر l_e را به اندازه یک واحد افزایش دهیم، متحمل هزینه نامنفی c_e^+ می‌شویم و به طور مشابه اگر l_e را به اندازه یک واحد کاهش دهیم، متحمل هزینه نامنفی c_e^- می‌شویم. همچنین فرض کنید x_e^+ و x_e^- به ترتیب برابر با مقداری باشد که طول یال l_e افزایش یا کاهش پیدا می‌کند (البته توجه داریم که این افزایش و کاهش نمی‌تواند به طور دلخواه انجام شود و فرض بر این است که x_e^+ و x_e^- به ترتیب از کران‌های بالا u_e^+ و u_e^- تبعیت می‌کنند). حال فرض کنید اگر φ مجموعه همه‌ی زیر مجموعه‌های $S \subseteq V$ با اندازه $|S| = p$ باشد. می‌توانیم مسئله p -میانه معکوس با طول یال‌های متغیر روی گراف G را به صورت زیر بیان کنیم:

اصلاح طول یال‌های $e \in E$ به l_e به $\tilde{l}_e = l_e + x_e^+ - x_e^-$ که سه خاصیت زیر را برقرار کند:

- در گراف G نسبت به $\tilde{l}_e$ ، مجموعه $\{s_1, s_2, \dots, s_p\}$ یک p -میانۀ مسئله شود، یعنی:

$$\sum_{i=1}^n w_i \min_{k=1, \dots, p} d_{\tilde{l}_e}(v_i, s_k) \leq \sum_{i=1}^n w_i \min_{v_k \in S} d_{\tilde{l}_e}(v_i, v_k) \quad \forall S \in \varphi$$

که در آن $d_{\tilde{l}_e}(u, v)$ فاصله جدید بین u, v است.

- کران محدودیت‌ها برقرار باشد، یعنی برای هر $e \in E$ داشته باشیم:

$$0 \leq x_e^- \leq u_e^- \quad 0 \leq x_e^+ \leq u_e^+$$

- تابع هزینه خطی زیر مینیمم شود.

$$\min \sum_{e \in E} (c_e^+ x_e^+ + c_e^- x_e^-)$$

که در آن E مجموعه یال‌های گراف است. مدل فوق در حالت کلی غیر خطی است که می‌توان در بعضی حالات به مدل خطی تبدیل کرد. البته در [۷] اگر گراف یک درخت باشد، یک مدل خطی برای این مسئله ارائه شده است.

مروری بر مسائل میانه معکوس با حداقل هزینه

در سال ۲۰۰۴ بورکارد^۷ و همکارانش مسئله ۱- میانه معکوس با تغییرات وزن مربوط به نقاط را مورد مطالعه قرار دادند. آنها ابتدا مسئله p - میانه گسسته معکوس را در فضای متریک (X, d) به صورت یک مدل برنامه‌ریزی خطی فرمول‌بندی کردند و با توجه به متناهی بودن تعداد قیود و متغیرهای این مدل نتیجه گرفتند که مسئله p - میانه معکوس گسسته با حداقل هزینه، که در آن مشتریان ممکن است دارای هر وزن حقیقی باشند، در زمان چند جمله‌ای قابل حل است، با این شرط که p ثابت بوده و به عنوان یک پارامتر ورودی نباشد و در ادامه مسئله ۱- میانه معکوس روی درخت و صفحه مورد بررسی قرار داده و یک الگوریتم شبه حریصانه در زمان $O(n \log n)$ برای آنها ارائه داد [۱۴]. گوان^۸ و ژنگ^۹ مسئله ۱- میانه معکوس با تغییرات وزن راسی روی درخت را تحت فاصله همینگ مورد مطالعه قرار دادند. آنها نشان دادند که این مسئله تحت فاصله همینگ معادل با یک مسئله کوله پشتی صفر و یک بوده و در نتیجه NP - سخت است [۳۹]. همچنین برای مسئله مذکور تحت نرم‌های همینگ تنگنا و چپیشف الگوریتم‌هایی با زمان اجرای خطی ارائه دادند.

در سال ۲۰۱۵ نگوین^{۱۰} و چاسین^{۱۱} مسئله ۱- میانه مرتب معکوس را روی درخت‌های غیر

⁷Burkard

⁸Guan

⁹Zhang

¹⁰Nguyen

¹¹Chassein

وزن دار مطالعه کردند. آنها ثابت نمودند که این مسئله تحت نرم همینگ جمعی NP - سخت بوده ولی تحت نرم‌های چپیشف و همینگ تنگنا با استفاده از روش جستجوی دودویی در زمان $O(n^2 \log n)$ حل می‌شود [۶۲]. مسئله ۱- میانه معکوس با تغییرات وزن راسی روی گراف‌های بلوکی در [۶۱] بررسی شده و برای حالتی که طول یال‌های هر بلوک گراف، یکسان در نظر گرفته شوند یک الگوریتم ترکیبیاتی با زمان اجرای $O(n \log n)$ پیشنهاد گردید. اخیراً نگوین و چی^{۱۲} مسئله ۱- میانه معکوس را با هزینه‌های غیر قطعی روی درخت مورد مطالعه قرار داده و برای به دست آوردن اطلاعات در مورد هزینه بهینه هر سطح اطمینان $\alpha \in [0, 1]$ یک مدل ارائه دادند [۶۳].

با توجه به این که نظریه مکانیابی معکوس یک مبحث جدید در زمینه مسائل بهینه‌سازی است. تعداد تحقیقات صورت گرفته در این زمینه در ایران اندک است که در ادامه مورد اشاره قرار می‌گیرد. در سال ۲۰۰۸ گلوی در پایان‌نامه خود مسئله ۱- میانه معکوس با تغییرات وزن راسی روی درخت که قبلاً توسط بورکارد و همکاران برای آن الگوریتم $O(n \log n)$ ارائه شده بود، به صورت یک مدل کوله پشتی پیوسته فرمول‌بندی کرده و نشان داد که این مسئله در زمان خطی قابل حل است [۳۱]. وی همچنین با همکاری بورکارد و گسسر^{۱۳} در سال ۲۰۱۰ مسئله فرما-وبر معکوس با تغییرات وزن نقاط را در حالت هزینه یکسان مورد بررسی قرار داده و برای آن الگوریتمی خطی ارائه داد [۱۶]. در سال ۲۰۱۱ باروقی و همکارانش نشان دادند که مسئله p - میانه معکوس با تغییرات طول یالی روی شبکه‌های کلی NP - سخت است. لذا آنها شبکه‌های خاصی چون درخت‌ها و گراف‌های ستاره را در نظر گرفته و الگوریتم‌های چند جمله‌ای برای آنها ارائه دادند [۷]. این مولف در تحقیقی دیگر با همکارانش مقاله‌ای در زمینه مسئله ۱- میانه معکوس با تغییر مختصات نقطه‌ای ارائه داد که در آن ثابت نمود که در صورتی که فواصل بین نقاط با نرم خطی اندازه‌گیری شود، این مسئله NP - سخت است. آنها همچنین ثابت کردند که با فرض این که وزن نقاط یکسان باشند، مسئله ۱- میانه معکوس به d مسئله کوله پشتی کاهش یافته و در نتیجه در زمان $O(nd)$ قابل حل است. در ضمن ایشان ثابت نمودند که این مسئله تحت نرم چپیشف در صفحه NP - سخت بوده و با نرم اقلیدسی در زمان $O(nd)$ قابل حل است [۶]. همان طور که می‌دانیم، مکان ۱- میانه درخت مستقل از طول یال‌ها است. اما وقتی طول یالی از درخت به صفر کاهش می‌یابد، توپولوژی درخت تغییر یافته و از این طریق ممکن است مکان ۱- میانه درخت تحت الشعاع قرار بگیرد. در سال ۲۰۱۳ سپاسیان و رهبرنیا از این خاصیت جالب استفاده کردند و مسئله ۱- میانه معکوس روی درخت را با تغییرات همزمان وزن راسی و طول یال‌ها در زمان $O(n \log n)$ حل نمودند [۶۶].

۳- مسئله ۱ میانه معکوس درخت با تغییر وزن رئوس

فرض کنید درخت $T = (V, E)$ داده شده است. می‌خواهیم راس s با تغییر وزن رئوس

¹²Chi

¹³Gassner

تبدیل به ۱-میانه درخت شود. فرض کنید راس m ، ۱-میانه درخت فعلی باشد و $m \neq s$. فرض کنید راس s از درجه K باشد. بنابراین K زیر درخت ماکسیمال از حذف یال‌های متصل به s تولید می‌شود، که آنها $T_1, T_2, \dots, T_K$ می‌نامیم. یکی از این زیر درخت‌ها شامل m است. فرض کنید m درون زیر درخت T_K باشد. باتوجه به خاصیت اکثریت وزنی داریم.

$$W(T_i) < w/2 \quad i = 1, \dots, K-1$$

$$W(T_K) \geq w/2$$

که در آن w وزن کلی درخت است و $W(T_i)$ مجموع وزن رئوس درخت T_i است. فرض کنید w وزن فعلی راس s باشد. تعریف می‌کنیم

$$H = \sum_{i=1}^K W(T_i)/2 = w/2$$

که در آن T_0 درختی تنها شامل راس s است. برای اینکه s نقطه ۱-میانه شود باید وزن s و زیر درخت‌های T_i ، $i = 1, \dots, K$ را تغییر دهیم در حقیقت باید وزن درخت T_K کم شود و وزن سایر زیر درخت‌ها افزایش یابد. شکاف بهینگی را به صورت زیر تعریف می‌کنیم:

$$D = W(T_K) - H$$

لم ۱.۲.۲. اگر $W(T_1) \leq w/2, \dots, W(T_{K-1}) \leq w/2$ آنگاه راس s ، ۱-میانه است اگر و تنها اگر شکاف بهینگی D مساوی صفر باشد [۱۳].

بر اساس لم بالا، بورکارد و همکارانش [۱۳] الگوریتمی را ارائه دادند که در این الگوریتم ابتدا وزن رئوس بر اساس هزینه متناظر کاهش یا افزایش به صورت صعودی مرتب شده و با شروع از کم هزینه‌ترین وزن راسی سعی در به صفر رساندن شکاف بهینگی می‌شود. از این رو آنها نتیجه گرفتن که مسئله ۱-میانه معکوس با حداقل هزینه روی درخت را می‌توان با استفاده از الگوریتم شبه حریصانه در زمان $O(n \log n)$ حل نمود. از لم بالا برای حل انواع دیگری از مسائل ۱-میانه معکوس روی درخت استفاده شده است.

الگوریتم ۱-میانه معکوس روی درخت

۱. وزن کل درخت یعنی $w = \sum_{i=1}^n w_i$ را محاسبه کنید.

۲. وزن زیر درخت‌های با ریشه‌ی s ، را محاسبه کنید (یعنی $T_1, \dots, T_K$). اگر تمام $T_1, \dots, T_K$ دارای وزن کمتر یا مساوی $w/2$ هستند، توقف کنید. s بهینه است.

در غیر اینصورت زیر درخت با بیشترین وزن را T_0 می‌نامیم و وزن آن را w_0 در نظر بگیرید.

۳. به ازای $i = 1, \dots, n$ اگر راس v_i خود راس s یا راس خارج از T_0 است، قرار دهید $r_i = c_i$ در غیر اینصورت قرار دهید $r_i = d_i$.

۴. قرار دهید $D = w_0 - w/2$ و $R = \{r_1, r_2, \dots, r_n\}$.

۵. اگر $R = \emptyset$ ، مسئله جواب ندارد و اگر $R \neq \emptyset$ ، فرض کنید کوچکترین عضو r_i ، R باشد.

۶. اگر $r_i = c_i$ آنگاه قرار دهید $p_i = D - c_i/2$ که $D = D - p_i/2$

$$p_i = \min\{2D, \bar{w}_i - w_i\}$$

در غیر این صورت قرار دهید $q_i = D - q_i/2$ که $D = D - q_i/2$

$$q_i = \min\{2D, w_i - \underline{w}_i\}$$

۷. اگر $D = 0$ توقف کنید، جواب بهینه بدست آمده، در غیر اینصورت به گام ۵ بروید.

مروری بر مسائل مرکز معکوس با حداقل هزینه

اولین کاری که در زمینه مسائل مرکز معکوس با حداقل هزینه صورت گرفت، در سال ۱۹۹۹ بود که در آن کای^{۱۴} و همکاران ثابت کردند که مسئله ۱- مرکز معکوس با تغییرات طول یالی روی شبکه‌های جهتدار بدون وزن NP - سخت است [۱۹]. همچنین NP - سختی این مسئله روی شبکه‌های بدون جهت نیز اخیراً توسط نگوین و چاسین در [۶۰] ثابت شده است. با توجه به این نکته، دیگر محققان در سال‌های بعد این مسئله را روی شبکه‌های خاصی مانند درخت مورد مطالعه قرار دادند. همانطور که در بخش مربوط به کلاسیک نیز اشاره شد، حکیمی ثابت نمود که خاصیت بهینگی راسی برای مسائل مرکز برقرار نمی‌باشد. لذا در بررسی مسئله مرکز معکوس باید دو حالت مرکز راسی معکوس و مرکز مطلق معکوس به طور جداگانه در نظر گرفته شوند. در سال ۲۰۰۸ یانگ^{۱۵} و ژانگ مسئله مرکز راسی معکوس با تغییرات طول یالی را روی درخت‌های بدون وزن در نظر گرفته [۷۳] و برای حالتی که طول تمام یال‌های اصلاح شده مثبت باقی بمانند، یعنی حالتی که در آن تغییر توپولوژی درخت مجاز نباشد، یک روش حل $O(n^2 \log n)$ ارائه دادند. آنها همچنین نشان دادند که در حالتی که تغییر توپولوژی درخت مجاز باشد، مسئله در زمان $O(n^3 \log n)$ قابل حل است. در ادامه کار این دو محقق علی زاده و بورکار در دو حالت راسی و مطلق مسئله مرکز معکوس را مورد مطالعه قرار داده [۳] و زمان اجرای ارائه شده توسط یانگ و ژانگ را به $O(n^2)$ در حالت بدون تغییر توپولوژی و $O(n^3)$ در حالت دیگر بهبود دادند. مسئله مرکز معکوس با ضرایب هزینه یکسان در [۴] بررسی شده و هر دو حالت راسی و مطلق آن در زمان $O(n \log n)$ حل شد. قابل ذکر است که مسئله مرکز معکوس فقط در حالت افزایش طول یالی با هزینه‌های دلخواه قبلاً توسط علی‌زاده و همکاران

¹⁴Cai

¹⁵Yang

در [۲] مورد بررسی قرار گرفته و برای آن روش حل $O(n \log n)$ ارائه شده بود. اخیراً نگوین و سپاسیان مسئله مرکز معکوس با حداقل هزینه را روی درخت‌های وزن دار تحت نرم چپیشف و همینگ مورد مطالعه قرار داده [۵۹] و نشان دادند که این مسائل بدون تغییر توپولوژی در زمان $O(n \log n)$ قابل حل هستند. آنها همچنین روش‌های حل با زمان $O(n^2)$ برای حالت کلی ارائه دادند. سپس نگوین و آن^{۱۶} برای مسئله ۱- مرکز معکوس با حداقل هزینه با تغییرات وزن راسی روی درخت‌ها یک الگوریتم $O(n^2)$ طراحی کردند [۵۸].

۲.۲.۲ مدل مکانیابی معکوس با محدودیت بودجه‌ای

در یک مسئله مکانیابی معکوس با محدودیت بودجه، یک بودجه معین B برای هزینه اصلاح پارامتر θ در نظر گرفته می‌شود. فرض کنید $s^* \in S$ یک جواب شدنی از پیش تعیین شده برای مسئله مکانیابی (LOC) باشد و $f(\theta, s^*)$ تابع هدف مسئله مکانیابی کلاسیک به ازای جواب شدنی s^* بوده و $g(c, \tilde{\theta})$ تابع هزینه متناظر با اصلاح پارامتر θ به $\tilde{\theta}$ باشد و β مجموعه محدودیت‌هایی باشد که پارامترهای اصلاح شده باید در آنها صدق کنند. بنابراین یک مسئله مکانیابی معکوس با محدودیت بودجه به صورت زیر فرمول‌بندی می‌گردد:

$$\begin{aligned} \min \quad & f(\tilde{\theta}, s^*) \\ \text{s.t.} \quad & g(c, \tilde{\theta}) \leq B \\ & \tilde{\theta} \in \beta \end{aligned}$$

در مسئله معکوس با محدودیت بودجه هدف بهینه کردن جواب مفروض s^* نیست، بلکه هدف این است که مقدار تابع هدف مسئله به ازای جواب شدنی s^* با توجه به محدودیت بودجه تعیین شده تا حد ممکن بهبود یابد.

مروری بر مسائل میانه معکوس با محدودیت بودجه‌ای

برمن و همکارانش برای اولین بار در سال ۱۹۹۲ مسئله ۱- میانه معکوس با محدودیت بودجه‌ای را مورد بررسی قرار داده و برای این مسئله روی شبکه‌های درختی یک الگوریتم خطی ارائه کردند [۱۱]. بورکارد و همکارانش در سال ۲۰۰۶ ثابت کردند که مسئله ۱- میانه معکوس با محدودیت بودجه‌ای با تغییرات طول یالی NP -سخت است و برای دورها الگوریتم خطی ارائه دادند [۱۳]. مسئله ۲- میانه معکوس با محدودیت بودجه‌ای روی درخت‌ها توسط بورکارد و همکارانش مورد مطالعه قرار گرفته و برای آن یک الگوریتم $O(n \log n)$ ارائه شده است [۱۵].

در سال ۲۰۱۰، ونگ^{۱۷} و بای^{۱۸} مسئله ۲- میانه معکوس با محدودیت بودجه‌ای روی دورها را به مسئله ۳- میانه معکوس با محدودیت بودجه‌ای روی یک مسیر تبدیل کرده و از این طریق نشان دادند که این مسئله در زمان چند جمله‌ای قابل حل است [۷۱]. مسئله ۱- میانه معکوس با محدودیت بودجه‌ای با یک قید اضافی روی درخت‌ها توسط ژانگ فن^{۱۹} و ژان^{۲۰} مورد بررسی قرار گرفته و ثابت شد که این مدل قابل تقسیم به دو زیر مسئله معادل بوده و این زیر مسئله‌ها را می‌توان به ترتیب توسط یک الگوریتم مینیمم برش و الگوریتم حریصانه حل نمود [۴۶].

مروری بر مسائل مرکز معکوس با محدودیت بودجه‌ای

در سال ۱۹۹۴، برمن ثابت کرد که مسئله مرکز معکوس با تغییرات طول یالی روی شبکه‌های بدون وزن، NP - سخت است [۱۰]. ژانگ و همکاران مسئله مرکز معکوس را روی درخت‌ها مورد مطالعه قرار داده و روشی با پیچیدگی محاسباتی $O(n^2 \log n)$ برای آن طراحی کردند [۷۵]. شایان ذکر است که رویکرد جواب ارائه شده توسط ژانگ و همکاران در سال‌ها بعد توسط علیزاده و بورکارد در [۳] برای کاهش ارتفاع یک زیر درخت ریشه‌دار از درخت تحت مطالعه مورد استفاده قرار گرفت. آنها با توجه به ساختار خاص شبکه N ، نشان دادند که این مسئله در زمان $O(n^2)$ قابل حل است. اخیراً، نگوین یک الگوریتم با زمان $O(n^2)$ برای حل مسئله مکانیابی مرکز معکوس با محدودیت بودجه‌ای روی درخت‌های وزن‌دار با تغییرات طول یالی ارائه داد که در آن تمام ضرایب هزینه یک عدد یکسان در نظر گرفته شده‌اند [۵].

۳.۲.۲ مدل مکانیابی معکوس با تابع هدف کراندار

مسئله مکانیابی (LOC) با تابع هدف $f(\theta, s)$ را در نظر بگیرید. فرض کنید s^* یک جواب از پیش تعیین شده برای مدل (LOC) باشد. به علاوه فرض کنید $\lambda \in R^+$ یک کران بالا روی مقدار هدف مسئله (LOC) است. در یک مسئله مکانیابی معکوس با تابع هدف کراندار، هدف اصلاح پارامتر θ به $\tilde{\theta}$ است به طوری که تابع هزینه $g(c, \tilde{\theta})$ مینیمم گردد و مقدار هدف متناظر با جواب مفروض s^* تحت بردار پارامتر اصلاح شده $\tilde{\theta}$ حداکثر برابر λ شود این مسئله به صورت زیر فرمول‌بندی می‌گردد:

$$\begin{aligned} \min \quad & g(c, \tilde{\theta}) \\ \text{s.t.} \quad & f(\tilde{\theta}, s^*) \leq \lambda \\ & \tilde{\theta} \in \beta \end{aligned}$$

¹⁷Wang

¹⁸Bai

¹⁹Jainfang

²⁰Juan

۴.۲.۲ مدل مکانیابی ترفیع و تنزیل

فرض کنید $B \in R^+$ بودجه تخصیص یافته برای اصلاح بردار پارامتر θ باشد. در یک مسئله ترفیع (تنزیل) متناظر مسئله مکانیابی کلاسیک (LOC)، هدف اصلاح بردار پارامتر θ به $\tilde{\theta}$ است به طوری که مقدار هدف مسئله (LOC) تحت بردار پارامتر جدید $\tilde{\theta}$ و با توجه به محدودیت بودجه تا حد ممکن بهتر (بدتر) شود. بنابراین می توان این مسئله را به صورت زیر فرمول بندی کرد:

$$\begin{aligned} \min(\max) \quad & f(\tilde{\theta}, s) \\ \text{s.t.} \quad & g(c, \tilde{\theta}) \leq B \\ & \tilde{\theta} \in \beta, s \in S \end{aligned}$$

که در آن β مجموعه محدودیت هایی است که بردار اصلاح شده $\tilde{\theta}$ باید در آن صدق کند و S مجموعه جواب های شدنی مسئله (LOC) تحت پارامترهای جدید $\tilde{\theta}$ است. لازم به ذکر است که برخلاف مسائل مکانیابی معکوس عنوان شده، در مسئله مکانیابی ترفیع (تنزیل) هیچ جواب از پیش تعیین شده ای مشخص نمی شود و هدف اصلاح پارامترهای شبکه و سپس یافتن جواب بهینه برای مسئله تحت پارامترهای اصلاح شده است. این نوع مسائل، ارتباط ضعیف تری با مسائل معکوس داشته ولی به دلیل اینکه در حوزه مسائل بهبود شبکه قرار می گیرند، می توان آنها را از نوع مسائل معکوس محسوب کرد.

مروری بر مسائل مکانیابی ترفیع و تنزیل

گسرن مسئله ترفیع و تنزیل ۱- میانه را با تغییرات وزن راسی روی شبکه ها در نظر گرفته و برای مسئله ترفیع ۱- میانه یک الگوریتم $O(n^2)$ ارائه نمود. وی همچنین ثابت نمود که مسئله تنزیل ۱- میانه در زمان چند جمله ای قابل حل می باشد. در ضمن برای گراف خاص درخت، الگوریتمی با زمان اجرای $O(n \log n)$ ارائه نمود [۳۳]. مسئله تنزیل ۱- میانه روی صفحه تحت نرم خطی در [۳۴] مورد مطالعه قرار گرفته و برای آن الگوریتمی با زمان اجرای $O(n \log^2 n)$ بر مبنای نظریه بازی ها پیشنهاد شد. پلاستریا^{۲۱} مسئله ترفیع و تنزیل ۱- میانه را روی فضای اقلیدسی با وزن های غیر قطعی در سال ۲۰۱۴ مورد مطالعه قرار داده و برای آن الگوریتم های چند جمله ای طراحی نمود [۶۴]. سپاسیان و رهبرنیا ثابت کردند که مسئله ترفیع p - میانه با تغییرات وزن راسی روی گراف های کلی در زمان چند جمله ای قابل حل می باشد در ضمن برای همین مسئله روی یک مسیر با هزینه های یکسان یک الگوریتم خطی ارائه دادند [۶۷].

²¹Plastria

فصل ۳

مسائل مکانیابی متعادل معکوس

فرض کنید $G = (V, E)$ یک گراف بدون جهت باشد. به طوری که V ، مجموعه رئوس گراف G و E مجموعه یال‌های گراف G ، $|V| = n$ و $|E| = m$ باشد. هر راس v_i دارای یک وزن نامنفی w_i می‌باشد که این w_i را می‌توان به دو حالت زیر تعبیر نمود.

- w_i بیانگر میزان اهمیت راس v_i در مسئله مکانیابی باشد.
- w_i را می‌توان تعداد مشتریان راس v_i در نظر گرفت به عبارت دیگر می‌توان w_i را به عنوان میزان تقاضای راس v_i دانست.

از طرفی هر یال $e_{ij} \in E$ دارای وزن d_{ij} است، که این وزن یال می‌تواند به صورت فاصله، هزینه، زمان سفر و ... بین دو راس در نظر گرفته شود. همان طور که از قبل گفتیم، در مسائل متعادل معکوس، یک جواب شدنی نامتعادل داریم. هدف این است که با تغییر پارامترهای مسئله که در اینجا وزن راس‌ها یا وزن یال‌ها و یا تغییر هم‌زمان هر دو، با کمترین هزینه، جواب را تا حد امکان متعادل کنیم.

یکی از کارهایی که در این زمینه انجام شده و ایده ما برای شروع این موضوع بود، مسئله ۲-میانه متعادل معکوس روی درخت همراه با تغییر وزن رئوس است. در مسئله فرض بر این است که دو سرویس‌دهنده m_1 و m_2 معلوم هستند. بنابراین برای رسیدن به تعادل (بهینه شدن مسئله) باید وزن رئوس گراف تغییر کند. از طرفی تغییر (افزایش یا کاهش) هر واحد وزن w_i برای ما هزینه‌های معادل c_i^+ یا c_i^- را در برخواهد داشت. هدف در این مسئله این

است که وزن رئوس با کمترین هزینه اصلاح شود به طوری که اختلاف وزن‌های کلی رئوس تخصیص داده شده به هر دو سرویس‌دهنده تا حد امکان کمینه شود.

برای رسیدن به تعادل باید وزن رئوس w_i تغییر نماید. بنابراین اگر q_i^+ و q_i^- به ترتیب مقدار افزایش و کاهش در وزن رئوس باشند، وزن نهایی رئوس بعد از به تعادل رسیدن از رابطه $\hat{w}_i = w_i + q_i^+ - q_i^-$ بدست می‌آید. V_1 و V_2 مجموعه رئوسی می‌باشند که به ترتیب به x_1 و x_2 تخصیص داده شده‌اند. c_i^+ و c_i^- به ترتیب هزینه‌های متحمل شده از افزایش و کاهش هر واحد وزن راس v_i یعنی w_i هستند. قرار دهیم

$$Q^+ = \{q_1^+, q_2^+, \dots, q_n^+\}, \quad Q^- = \{q_1^-, q_2^-, \dots, q_n^-\}$$

در این صورت در یک مسئله ۲- میانه متعادل معکوس با تغییر وزن رئوس حاضر، به دنبال کمینه کردن توابع زیر هستیم:

$$f_1(Q^+, Q^-) = \sum_{i=1}^n (c_i^+ q_i^+ + c_i^- q_i^-). \quad (1.3)$$

$$f_2(Q^+, Q^-) = \left| \sum_{i \in V_1} \hat{w}_i - \sum_{i \in V_2} \hat{w}_i \right|. \quad (2.3)$$

رابطه (۱.۳) بیانگر هزینه کل ناشی از افزایش یا کاهش وزن رئوس تحت نرم خطی، برای رسیدن به تعادل است و رابطه (۲.۳) بیانگر شرط تعادل وزنی سرویس‌گیرنده‌ها است. در این مقاله مسئله را با دو مدل (کمترین هزینه و بودجه محدود) در نظر گرفتند و الگوریتم‌هایی برای این حل این دو مدل ارائه شده است. برای اطلاعات بیشتر به [۵۶] مراجعه کنید.

۱.۳ مسائل مکانیابی متعادل معکوس با طول یال‌های متغیر

فرض کنید

$$V_1 = \{v_i \in V \mid d(v_i, m_1) \leq d(v_i, m_2)\}$$

و $V_2 = V \setminus V_1$ به ترتیب مجموعه رئوسی هستند که به سرویس‌دهنده‌های m_1 و m_2 اختصاص داده می‌شوند. در مسئله مکانیابی ۲- میانه متعادل معکوس با حداقل هزینه، قصد داریم طول یال‌ها را با کمترین هزینه اصلاح کنیم به طوری که اختلاف وزن‌های کلی رئوس در V_1 و V_2 کمینه شود. برای هر یال e_i ، فرض کنید هزینه افزایش هر واحد l_i ، c_i^+ و هزینه کاهش هر واحد l_i ، c_i^- است. همچنین q_i^+ و q_i^- به ترتیب مقادیر افزایش و کاهش طول یال l_i است. این

افزایش و کاهش طول یال e_i به طور دلخواه انجام نمی‌شود و فرض بر این است که q_i^+ و q_i^- به ترتیب از کران‌های بالا u_i^+ و u_i^- تبعیت می‌کنند. قرار دهیم

$$Q^+ = \{q_1^+, q_2^+, \dots, q_m^+\}, \quad Q^- = \{q_1^-, q_2^-, \dots, q_m^-\}.$$

همچنین برای $i = 1, \dots, m$ ، قرار دهید $\hat{l}_i = l_i + q_i^+ - q_i^-$. توجه کنید که با تغییر طول یال‌ها، مجموعه رئوس در V_1 و V_2 ممکن است تغییر کند. بنابراین V_1 و V_2 تغییر یافته را به ترتیب با $\hat{V}_1$ و $\hat{V}_2$ نشان می‌دهیم.

در مدل معکوس با حداقل هزینه با تغییر طول، کمینه کردن دوتابع هدف زیر را در نظر می‌گیریم.

$$f_1(Q^+, Q^-) = \sum_{i=1}^m (c_i^+ q_i^+ + c_i^- q_i^-). \quad (1.3)$$

$$f_2(Q^+, Q^-) = \left| \sum_{i \in \hat{V}_1} w_i - \sum_{i \in \hat{V}_2} w_i \right|. \quad (2.3)$$

اگر در این مدل، محدودیت بودجه داشتیم، نمی‌توانیم طول یال‌ها را برای کمینه کردن هر دوتابع هدف بالا تغییر دهیم، در این حالت مسئله را با بودجه محدود در نظر می‌گیریم که به آن مدل معکوس با محدودیت بودجه‌ای می‌گویند. در بخش‌های بعدی مدل‌ها و الگوریتم‌های حل مسئله معکوس با کمترین هزینه و محدودیت بودجه‌ای ارائه شده است.

۱.۱.۳ مدل معکوس با حداقل هزینه

در این بخش فرض می‌کنیم بودجه نامحدود است، به عبارت دیگر مدل معکوس با حداقل هزینه را در نظر می‌گیریم. در مدل معکوس با حداقل هزینه قصد داریم طول یال‌ها را با کمترین هزینه تغییر دهیم، طوری که وزن رئوس تخصیص داده شده به سرویس‌دهنده‌های m_1 و m_2 در حالت تعادل باشند.

اگر $W(V_1) = W(V_2)$ ، پس سرویس‌دهنده‌ها در حالت تعادل هستند و طول یال‌ها تغییر نمی‌کنند. در غیر این صورت بدون از دست دادن کلیت، فرض کنید $W(V_1) > W(V_2)$. پس برای بدست آوردن جواب شدنی، از طریق تغییر طول بعضی از یال‌ها، مجموعه رئوسی که به m_1 تخصیص می‌یابد یعنی V_1 باید کم شوند، یا مجموعه رئوسی که به m_2 تخصیص می‌یابد یعنی V_2 باید زیاد شوند. مشاهده کردیم که با اصلاح طول یال‌های خارج از مسیر بین m_1 و m_2 ، $W(V_1)$ و $W(V_2)$ بدون تغییر خواهند بود.

فرض کنید z یک نقطه میانی در مسیر بین m_1 و m_2 باشد و $e_z = [v_j, v_{j+1}]$ یال شامل است. همچنین T_1 و T_2 دو زیر درخت به دست آمده از حذف یال e_z ، طوری که $v_j \in T_1$ و $v_{j+1} \in T_2$ باشد. در واقع V_1 و V_2 به ترتیب مجموعه رئوس در T_1 و T_2 هستند. فرض کنید $P = \{e_{h_1}, \dots, e_{h_z}, \dots, e_{h_t}\}$ یال‌های روی مسیر بین m_1 و m_2 باشد و P_1, P_2 به صورت زیر تعریف شده‌اند.

$$P_1 = P \cap T_1, \quad P_2 = (P \cap T_2) \cup e_z$$

چون فرض کردیم $W(V_1) > W(V_2)$ ، پس خاصیت زیر برقرار است.

لم ۱.۱.۳. برای به دست آوردن یک تخصیص متعادل، کفایت طول یال‌ها در مسیرهای P_1 و P_2 را به ترتیب افزایش و کاهش دهیم.

پس برای یافتن یک تخصیص متعادل با کمترین هزینه، طول یال‌ها در P_1 افزایش و یا طول یال‌ها در P_2 را کاهش می‌دهیم. به عبارت دیگر در تلاش هستیم راس v_j را به m_2 اختصاص دهیم. برای رسیدن به این هدف با کمترین هزینه برای $i = 1, \dots, t$ در نظر بگیرید

$$r_i = \begin{cases} c_{h_i}^+ & \text{if } e_{h_i} \in P_1 \\ c_{h_i}^- & \text{if } e_{h_i} \in P_2 \end{cases} \quad (3.3)$$

$$D = \{r_1, r_2, \dots, r_t\} \quad (4.3)$$

که r_i در عبارت فوق $c_{h_i}^+$ یا $c_{h_i}^-$ است طوری که

$$r_1 \leq r_2 \leq r_3 \leq \dots \leq r_t$$

با r_1 شروع می‌کنیم. حالت‌های زیر ممکن است اتفاق بیافتد:

۱. اگر $r_1 = c_{h_k}^+$ (یعنی $e_{h_k} \in P_1$)، پس طول یال e_{h_k} به اندازه $q_{h_k}^+ = \min\{|d(v_j, m_1) - d(v_j, m_2)| + \epsilon, u_{h_k}^+\}$ که ϵ مقدار خیلی کوچک مثبت است، باید افزایش یابد

۲. اگر $r_1 = c_{h_k}^-$ (یعنی $e_{h_k} \in P_2$)، پس طول یال e_{h_k} به اندازه $q_{h_k}^- = \min\{|d(v_j, m_1) - d(v_j, m_2)| + \epsilon, u_{h_k}^-\}$ ، باید کاهش یابد.

این روش را برای r_2 و r_3 و در نهایت r_t ادامه می‌دهیم تا زمانی که فاصله جدید بین v_j و m_1 بیشتر از v_j و m_2 شود و بنابراین v_j به m_2 اختصاص یابد. فرض کنید که $e_z, e_{z_1} \in P$ دو یال

مجاور v_j روی P است و Tv_j زیر درختی از T که از حذف یال e_z و e_{z1} بدست می‌آید که راس v_j را شامل می‌شود (شکل ۱.۳ را ببینید). قرار می‌دهیم

$$\hat{T}_1 = T_1 \setminus Tv_j, \quad \hat{T}_2 = T_2 \cup Tv_j.$$

و $\hat{V}_1$ و $\hat{V}_2$ به ترتیب مجموعه رئوسی در $\hat{T}_1$ و $\hat{T}_2$ تعریف می‌کنیم. اگر

$$|W(\hat{V}_1) - W(\hat{V}_2)| < |W(V_1) - W(V_2)|$$

تابع هدف متعادل بهبود می‌یابد و با راس بعدی در T_1 ادامه می‌دهیم و تلاش می‌کنیم با تغییر طول یال‌ها در P به همین طریق، راس بعدی را به m_2 اختصاص دهیم. در غیر این صورت الگوریتم پایان می‌یابد. در نتیجه از لم زیر برای پایان الگوریتم (شرط توقف) می‌توانیم استفاده کنیم.

لم ۲.۱.۳. فرض کنید T_1 و T_2 به ترتیب زیر درخت‌های شامل رئوس اختصاص یافته به m_1 و m_2 در تکرار فعلی است. همچنین V_1 و V_2 را به ترتیب مجموعه رئوس در T_1 و T_2 در نظر بگیرید. $v \in T_1$ راس مجاور با T_2 است و $\hat{V}_1 = V_1 \setminus \{v\}$, $\hat{V}_2 = V_2 \cup \{v\}$. اگر

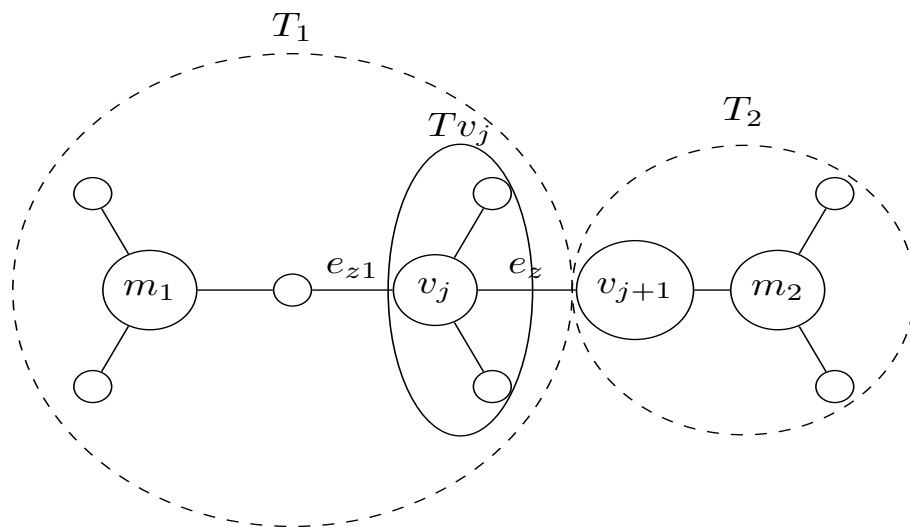
$$|W(\hat{V}_1) - W(\hat{V}_2)| \geq |W(V_1) - W(V_2)|$$

بنابراین تابع هدف متعادل با تغییر مجموعه‌های V_1 و V_2 بهبود نخواهد یافت.

در هر تکرار الگوریتم، ملاحظه زیر را در نظر گرفتیم.

ملاحظه ۱

اگر طول یک یال در یک تکرار افزایش یافت پس طول آن یال در تکرارهای بعدی نمی‌تواند کاهش یابد و اگر طول یک یال در یک تکرار کاهش یابد پس طول آن یال در تکرارهای بعدی



شکل ۱.۳: درخت T و زیر درخت‌های T_1, T_2 و Tv_j

نمی‌تواند افزایش یابد.

این ایده‌ها ما را به الگوریتم زیر سوق دادند.

الگوریتم IB2FLP

ورودی:

درخت وزن دار T ، دو راس m_1 و m_2 از درخت T به عنوان مکان سرویس‌دهنده‌ها طوری که $W(V_1) > W(V_2)$ که V_1 و V_2 به ترتیب مجموعه رؤس اختصاص یافته به m_1 و m_2 هستند. همچنین هزینه افزایش و کاهش وزن یال‌ها داده شده است.

خروجی:

مقدار بهینه تابع هدف f_1 و طول جدید یال‌های $\hat{e}_i$ برای تعادل وزن رؤسی که به سرویس‌دهنده‌های m_1 و m_2 اختصاص یافته‌اند، داده شده است.

مقداردهی اولیه:

- فرض کنید $P_{m_1, m_2} = \{v_{a_1} = m_1, \dots, v_{a_{t+1}} = m_2\}$ مسیر بین m_1 و m_2 باشد.
- z نقطه میانی بین m_1 و m_2 و $e_{h_z} = [v_{a_j}, v_{a_{j+1}}]$ یال شامل z را در نظر بگیرید.
- $P = \{e_{h_1}, \dots, e_{h_z}, \dots, e_{h_t}\}$ را مجموعه‌ای از یال‌ها روی P_{m_1, m_2} ، $P_1 = P \cap T_1$ و $P_2 = (P \cap T_2) \cup e_{h_z}$ در نظر بگیرید.
- $\hat{T}_2 = T_2 \cup T v_{a_j}$ ، $\hat{T}_1 = T_1 \setminus T v_{a_j}$ و فرض کنید $\hat{V}_1$ و $\hat{V}_2$ به ترتیب مجموعه رؤسی در $\hat{T}_1$ و $\hat{T}_2$ باشند.
- $D = \{r_1, r_2, \dots, r_t\}$ که $i = 1, \dots, t$ به صورت (۳.۳) تعریف می‌شوند و به صورت دنباله صعودی مرتب شده‌اند.
- $f_1 := \circ$ ، $b := a_j$ قرار دهید.

گام تکراری:

تا زمانی که $|W(\hat{V}_1) - W(\hat{V}_2)| < |W(V_1) - W(V_2)|$ ، کارهای زیر را انجام دهید.

$$1. \quad S = s_2 - s_1 \text{ و } s_2 := d(v_b, m_2), \quad s_1 := d(v_b, m_1), \quad i := 1$$

۲. تا زمانی که $S > \epsilon$ کارهای زیر را انجام دهید.

(a) اگر $r_i = c_{h_k}^+$ و $e_{h_k} \in P_1$ قرار دهید.

$$i. \quad q_{h_k}^+ := \min\{S + \epsilon, u_{h_k}^+\}$$

$$ii. \quad \hat{l}_{h_k} := l_{h_k} + q_{h_k}^+$$

$$iii. \quad f_1 := f_1 + r_i q_{h_k}^+$$

$$iv. \quad \text{اگر } q_{h_k}^+ = u_{h_k}^+ \text{، قرار دهید } S = S - u_{h_k}^+$$

در غیراین صورت $S = \epsilon$ و $c_{h_k}^+$ را از لیست حذف کنید.

* حلقه خاتمه یافت.

حلقه خاتمه یافت.

(b) اگر $r_i = c_{h_k}^-$ و $e_{h_k} \in P_\uparrow$ ، قرار دهید.

$$q_{h_k}^- := \min\{S + \epsilon, u_{h_k}^-\} \quad .i$$

$$\hat{l}_{h_k} := l_{h_k} - q_{h_k}^- \quad .ii$$

$$f_\uparrow := f_\uparrow + r_i q_{h_k}^- \quad .iii$$

.iv اگر $q_{h_k}^- = u_{h_k}^-$ ، قرار دهید $S = S - u_{h_k}^-$.

در غیر این صورت $S = \epsilon$ و $c_{h_k}^-$ را از لیست حذف کنید.

* حلقه خاتمه یافت.

حلقه خاتمه یافت.

$$i := i + 1 \quad (c)$$

حلقه خاتمه یافته است.

۳. $j = j - 1$ و قرار دهید و $c_{h_z}^-$ در D قرار دهید و $P_\uparrow = P_\uparrow \cup \{e_{h_z}\}$ ، $P_\downarrow = P_\downarrow \setminus \{e_{h_z}\}$ ، $V_\uparrow = \hat{V}_\uparrow$ ، $V_\downarrow = \hat{V}_\downarrow$.

۴. $b = a_j$ ، $h_z = h_z - 1$ ، رئوس در Tv_b را از $\hat{V}_\downarrow$ حذف کرده و همه را به $\hat{V}_\uparrow$ اضافه کنید.

پایان الگوریتم.

پیچیدگی الگوریتم

برای محاسبه پیچیدگی زمانی الگوریتم، توجه کنید که همه راس‌ها از زیر درخت Tv_b (به جز v_b) در مسیر $p_{m_\downarrow, m_\uparrow}$ قرار ندارند و چون تخصیص مجدد در زمان خطی در تعدادی از راس‌ها در Tv_b انجام می‌شود بنابراین زمان اجرا به

$$\sum_{(v_i, v_t) \in P_{m_\downarrow, m_\uparrow}} O(|V(Tv_i)|) = O(n),$$

محدود می‌شود و چون مجموعه‌های $V(Tv_t)$ مجزا هستند همچنین در بدترین حالت زمان $O(m \log m)$ برای مرتب کردن یال‌ها نیاز است و در یک درخت $n = m + 1$ است بنابراین پیچیدگی زمانی الگوریتم $O(n \log n)$ است.

قضیه ۱.۱.۳. مسئله مکانیابی ۲- میانه متعادل معکوس با کمترین هزینه با طول یال‌های متغیر روی درخت در زمان $O(n \log n)$ می‌تواند حل شود.

برای توضیح این الگوریتم مثال زیر را در نظر بگیرید.

مثال ۱.۱.۳. درخت T در شکل ۲.۳ را در نظر بگیرید. عدد روی یال‌ها طول یال را مشخص می‌کند. وزن راس‌ها در جدول ۱.۳ داده شده است. $m_1 = v_3$ و $m_2 = v_{14}$ ، در این مثال فرض شده است حداقل طول یال‌ها صفر و کران بالا برای تغییر طول یال‌ها ۳ در نظر گرفته شده است. پس

$$V_{\mathcal{F}} = \{v_{1\mathcal{F}}, v_{1\mathcal{A}}, \dots, v_{1\mathcal{Q}}\}$$

که $W(V_1) = 41$ و $W(V_2) = 16$ است. هزینه‌های افزایش و کاهش طول یال‌ها روی مسیر بین m_1 و m_2 در جدول ۲.۳ داده شده است. بنابراین

$$P_1 = \{e_1 = (v_3, v_6), e_2 = (v_6, v_7), e_3 = (v_7, v_8), e_4 = (v_8, v_{10}), e_5 = (v_{10}, v_{13})\}$$

و $P_2 = \{e_6 = (v_{13}, v_{14})\}$ است. با مرتب کردن هزینه‌های اصلاح طول یال‌ها خواهیم داشت

$$r_1 = c_{\mathfrak{F}}^+ \leq r_{\Upsilon} = c_1^+ \leq r_{\Psi} = c_{\Upsilon}^+ \leq r_{\mathfrak{F}} = c_{\mathfrak{E}}^- \leq r_{\Delta} = c_{\Delta}^+ \leq r_{\mathfrak{E}} = c_{\Psi}^+. \quad (\Delta.3)$$

در اولین گام باید رئوس در Tv_{13} را با تغییر طول بعضی از یال‌های روی مسیر بین m_1 و m_2 از V_1 به V_2 انتقال دهیم. با توجه به هزینه‌ها با $c_1^+ = r_1$ شروع می‌کنیم و مقادیرهای $l_1^+ = 2 + \epsilon$, $q_1^+ = 1 + \epsilon$ و $f_1 = \frac{1}{6}(1 + \epsilon)$ به دست می‌آید. پس رئوس در Tv_{13} به m_2 تخصیص می‌یابد.

جدول ۱.۳: وزن راس‌ها در درخت T

[illegible]

جدول ۲.۳: هزینه اصلاح طول یال‌ها در درخت T

e_i	l_i	c_i^+	c_i^-
$e_1 = (v_3, v_6)$	۲	۰/۸	۱/۲
$e_2 = (v_6, v_7)$	۱	۱/۰	۰/۴
$e_3 = (v_7, v_8)$	۱	۲/۰	۱/۴
$e_4 = (v_8, v_{10})$	۱	۰/۶	۱/۱
$e_5 = (v_{10}, v_{13})$	۱	۱/۳	۰/۵
$e_6 = (v_{13}, v_{14})$	۷	۰/۹	۱/۰

در گام دوم دیدیم که $W(V_1) = ۴۰$, $W(V_2) = ۱۷$ ، پس برای ایجاد تعادل بایستی راس v_{10} با تغییر طول یال‌ها در P به m_2 تخصیص بیابد. در نهایت بعد از سه تکرار جواب بهینه با $W(V_2) = ۳۱$ و $W(V_1) = ۲۶$, $f_1 = ۷ + ۳/۱\epsilon$ که $V_2 = \{v_8, v_9, \dots, v_{19}\}$ و $V_1 = \{v_1, v_2, \dots, v_7\}$ به دست می‌آید.

۲.۱.۳ مدل معکوس با محدودیت بودجه‌ای

در این بخش حالتی را که بودجه محدود و مساوی $B > 0$ است را در نظر می‌گیریم. این بودجه برای اصلاح طول یال‌ها صرف می‌شود طوری که اختلاف بین وزن رئوس تخصیص یافته به m_1 و m_2 تا حد امکان کوچک شود.

نمادهای مشابه با حالت معکوس با حداقل هزینه را در نظر بگیرید. فرض کنید z نقطه میانی بین m_1 و m_2 ، $e_z = [v_j, v_{j+1}]$ یالی شامل z و T_1, T_2 دو زیر درختی که از حذف یال e_z طوری که $v_j \in T_1$ و $v_{j+1} \in T_2$ به دست آمده باشد. همچنین $\hat{T}_1, \hat{T}_2$ دو زیر درخت از T با طول یال‌ها اصلاح شده هستند و شامل رئوسی هستند که به ترتیب به m_1 و m_2 اختصاص می‌یابند. توجه کنید که هر راس به نزدیکترین سرویس‌دهنده اختصاص می‌یابد. بنابراین ممکن است مجموعه رئوس در $\hat{T}_1$ با مجموعه رئوس در T_1 تفاوت داشته باشد. این خاصیت برای $\hat{T}_2$ و T_2 نیز ممکن است برقرار باشد.

بنابراین مدل با محدودیت بودجه‌ای می‌تواند به صورت زیر نوشته شود:

$$P_2 : \quad \min f_2 = \left| \sum_{i \in \hat{T}_1} w_i - \sum_{i \in \hat{T}_2} w_i \right|$$

$$s.t.$$

$$\sum_{i=1}^m (c_i^+ q_i^+ + c_i^- q_i^-) \leq B.$$

$$0 \leq q_i^+ \leq u_i^+, \quad i = 1, \dots, m.$$

$$0 \leq q_i^- \leq u_i^-, \quad i = 1, \dots, m.$$

توجه کنید که در مدل معکوس با حداقل هزینه، مینیمم هزینه کلی اصلاح طول یال‌ها یعنی مینیمم f_1 در (۱.۳)، را هم به عنوان تابع هدف در نظر گرفتیم. در صورتی که در این مدل تابع تعادل یعنی f_2 در (۲.۳) به عنوان تابع هدف در نظر می‌گیریم. در ادامه، در حالت مشابه با مدل معکوس با حداقل هزینه، الگوریتمی با پیچیدگی زمانی $O(n \log n)$ ارائه می‌شود.

فرض کنید V_1 و V_2 به ترتیب مجموعه رئوس در T_1 و T_2 است. اگر $W(V_1) = W(V_2)$ پس جواب $q_i^+ = 0, q_i^- = 0$ برای $i = 1, \dots, m$ بهینه و بنابراین وزن یال‌ها بدون تغییر باقی می‌مانند. در غیر این صورت بدون از دست دادن کلیت، فرض کنید $W(V_1) > W(V_2)$. بنابراین برای یافتن یک جواب بهینه، طول یال‌ها در T_1 افزایش یا طول یال‌ها در T_2 باید کاهش یابد. با در نظر گرفتن نمادهایی مشابه در قسمت قبل، فرض کنید $P = \{e_{h_1}, \dots, e_z, \dots, e_{h_t}\}$ یال‌های روی مسیر بین m_1 و m_2 باشد و $P_2 = (P \cap T_2) \cup e_z$ ، $P_1 = P \cap T_1$ همچنین s_2, s_1 به ترتیب فاصله بین v_j, m_1 و v_j, m_2 است.

فرض کنید $W(V_1) > W(V_2)$ ، بنابراین مقدار $W(V_1) - W(V_2)$ را تا حد امکان با توجه به بودجه

کاهش می‌دهیم. مانند حالت قبل با افزایش طول یال‌های روی P_1 یا کاهش طول یال‌ها روی P_2 تلاش می‌کنیم v_j به m_2 اختصاص یابد. دوباره قرار می‌دهیم

$$D = \{r_1, r_2, \dots, r_t\}$$

که برای $r_i, i = 1, \dots, t$ به صورت (۳.۳) تعریف شده است و

$$r_1 \leq r_2 \leq r_3 \leq \dots \leq r_t$$

با r_1 شروع می‌کنیم و حالت‌های زیر را در نظر می‌گیریم.

۱. اگر $r_1 = c_{h_k}^+$ ، $r_1 \in P_1$ پس طول یال e_{h_k} به اندازه $q_{h_k}^+ = \min\{|d(v_j, m_1) - d(v_j, m_2)| + \epsilon, u_{h_k}^+, \frac{B}{c_{h_k}^+}\}$ افزایش می‌یابد.

۲. اگر $r_1 = c_{h_k}^-$ ، $r_1 \in P_2$ پس طول یال e_{h_k} به اندازه $q_{h_k}^- = \min\{|d(v_j, m_1) - d(v_j, m_2)| + \epsilon, u_{h_k}^-, \frac{B}{c_{h_k}^-}\}$ کاهش می‌یابد.

این روش را برای r_2, r_3 و در نهایت r_t ادامه می‌دهیم تا راس v_j به m_2 اختصاص یابد. پس اگر $|W(\hat{V}_1) - W(\hat{V}_2)| < |W(V_1) - W(V_2)|$ تابع هدف می‌تواند بهبود یابد و با راس‌های بعدی در زیر درختی که وزن بیشتری دارد، ادامه می‌دهیم و تلاش می‌کنیم با تغییر طول یال‌ها در P رئوس را به m_2 تخصیص دهیم. در غیر این صورت الگوریتم به پایان می‌رسد. این ایده ما را به الگوریتم زیر سوق داد.

الگوریتم [RB2FLP]

ورودی:

درخت وزن‌دار T دو راس m_1 و m_2 از درخت T به عنوان مکان سرویس‌دهنده‌ها که در آن $W(V_1) > W(V_2)$ و V_1, V_2 به ترتیب مجموعه رئوس اختصاص یافته به m_1 و m_2 هستند. همچنین هزینه افزایش و کاهش طول یال‌ها و بودجه کلی B داده شده است.

خروجی:

مقدار بهینه تابع هدف f_2 و طول‌های جدید یال‌های $\hat{e}_i$ برای مینیمم کردن اختلاف وزن رئوس تخصیص یافته به سرویس‌دهنده‌های m_1 و m_2 داده می‌شود.

مقداردهی اولیه:

- مسیر بین m_1 و m_2 $P_{m_1, m_2} = \{v_{a_1} = m_1, \dots, v_{a_{t+1}} = m_2\}$ قرار دهید.
- z نقطه میانی بین m_1 و m_2 و $e_{h_z} = [v_{a_j}, v_{a_{j+1}}]$ یال شامل z را در نظر بگیرید.
- $P = \{e_{h_1}, \dots, e_{h_z}, \dots, e_{h_t}\}$ را مجموعه‌ای از یال‌ها روی P_{m_1, m_2} و $P_1 = P \cap T_1$ و $P_2 = (P \cap T_2) \cup e_{h_z}$ در نظر بگیرید.

• $\hat{T}_1 = T_1 \cup Tv_{a_j}$, $\hat{T}_1 = T_1 \setminus Tv_{a_j}$ و فرض کنید $\hat{V}_1$ و $\hat{V}_2$ به ترتیب مجموعه رئوسی در $\hat{T}_1$ و $\hat{T}_2$ باشند.

• $D = \{r_1, r_2, \dots, r_t\}$ که $i = 1, \dots, t$, r_i به صورت (۳.۳) تعریف شده، را به صورت دنباله صعودی مرتب کنید.

• $b := a_j$

گام تکراری:

تا زمانی که $|W(\hat{V}_1) - W(\hat{V}_2)| < |W(V_1) - W(V_2)|$ و $B > 0$ کارهای زیر را انجام دهید.

$$1. S = s_2 - s_1 \text{ و } i := 1, s_1 := d(v_b, m_1), s_2 := d(v_b, m_2).$$

۲. تا زمانی که $S > 0$ و $B > 0$ کارهای زیر را انجام دهید

(a) اگر $r_i = c_{h_k}^+$ و $e_{h_k} \in P_1$ ، قرار دهید.

$$i. q_{h_k}^+ := \min\{S + \epsilon, u_{h_k}^+, \frac{B}{c_{h_k}^+}\}$$

$$ii. B := B - c_{h_k}^+ q_{h_k}^+$$

iii. اگر $(B \neq 0)$ یا $(B = 0 \text{ and } q_{h_k}^+ = S + \epsilon)$ ، بنابراین

$$A. \hat{l}_{h_k} := l_{h_k} + q_{h_k}^+$$

$$B. \text{ اگر } q_{h_k}^+ = u_{h_k}^+ \text{، قرار دهید } S = S - u_{h_k}^+.$$

در غیراین صورت $S = \epsilon$ و $c_{h_k}^+$ را از لیست حذف کنید.

★ حلقه خاتمه یافت

حلقه خاتمه یافت.

حلقه خاتمه یافته است.

(b) اگر $r_i = c_{h_k}^-$ و $e_{h_k} \in P_2$ ، قرار دهید.

$$i. q_{h_k}^- := \min\{S + \epsilon, u_{h_k}^-, \frac{B}{c_{h_k}^-}\}$$

$$ii. B := B - c_{h_k}^- q_{h_k}^-$$

iii. اگر $(B \neq 0)$ یا $(B = 0 \text{ and } q_{h_k}^- = S + \epsilon)$ ، بنابراین

$$A. \hat{l}_{h_k} := l_{h_k} - q_{h_k}^-$$

$$B. \text{ اگر } q_{h_k}^- = u_{h_k}^- \text{، قرار دهید } S = S - u_{h_k}^-.$$

در غیراین صورت $S = \epsilon$ و $c_{h_k}^-$ را از لیست حذف کنید.

* حلقه خاتمه یافت.

حلقه خاتمه یافته است.

حلقه خاتمه یافت.

$$i := i + 1 \quad (c)$$

حلقه خاتمه یافته است.

۳. $j = j - 1$ قرار دهید و D در $c_{h_z}^-$, $P_2 = P_2 \cup \{e_{h_z}\}$, $P_1 = P_1 \setminus \{e_{h_z}\}$, $V_2 = \hat{V}_2$, $V_1 = \hat{V}_1$

۴. $h_z = h_z - 1$, $b = a_j$ و رئوس در Tv_b از $\hat{V}_1$ را حذف کرده و همه را به $\hat{V}_2$ اضافه کنید.

پایان الگوریتم.

قضیه ۲.۱.۳. مسئله مکانیابی ۲-میانه متعادل معکوس با محدودیت بودجه‌ای با طول یال‌های متغیر روی درخت در زمان $O(n \log n)$ قابل حل است.

□

برهان. مشابه حالت کمترین هزینه، ثابت می‌شود.

مثال ۲.۱.۳. درخت T را در شکل ۲.۳ در نظر بگیرید. وزن راس‌ها و طول یال‌ها و هزینه اصلاح طول یال‌ها مشابه با مثال ۱.۳.۴ است.

$V_2 = \{v_{14}, \dots, v_{19}\}$ و $V_1 = \{v_1, v_2, \dots, v_{13}\}$, $\epsilon = 0/2$, $B = 2$, $m_2 = v_{14}$, $m_1 = v_3$ مجموعه رئوسی که به m_2 و m_1 تخصیص می‌یابند. همچنین $W(V_2) = 16$, $W(V_1) = 41$ و $f_2 = 25$ است. هزینه اصلاح طول یال‌ها را به صورت (۵.۳) مرتب کنید.

در اولین گام، بایستی راس v_{13} را از مجموعه V_1 به V_2 انتقال دهیم. با انتخاب $r_1 = c_4^+$ و جایگذاری $q_4^+ = 1 + \epsilon = 1/2$

به دست می‌آوریم. پس راس $\hat{l}_4 = 2 + \epsilon = 2/2$, $f_1 = 0/6(1 + \epsilon) = 0/72$, $B = 2 - f_1 = 1/28$ به m_2 تخصیص می‌یابد و $f_2 = W(V_1) - W(V_2) = 23$ می‌شود.

در گام دوم راس v_{10} با کاهش طول یال l_5 و افزایش طول یال l_4 به m_2 تخصیص می‌دهیم. بنابراین $q_5^+ = 1/2$, $\hat{l}_4 = 3/4$, $f_1 = 1/94$, $B = 0/06$ بدست می‌آوریم. سرانجام بودجه کافی برای ادامه مسئله نداریم و الگوریتم با $f_2 = 19$ پایان می‌یابد.

فصل ۴

مسائل مکانیابی ۱ – میانه معکوس با تبادل فاصله

۱.۴ تعریف مسئله

همانطور که قبلاً اشاره شد کاربرد مدل‌های مکانیابی معکوس زمانی است که با وضعیتی روبرو می‌شویم که مکان سرویس‌دهنده موجود است و به دلایل اقتصادی و ... نمی‌توانیم مکان سرویس‌دهنده را تغییر دهیم. در این حالت ممکن است وزن راس‌ها یا طول یال‌ها را روی شبکه تا حد امکان تغییر دهیم طوری که مکان سرویس‌دهنده بهینه شود. در حالتی خاص مدلی که در این فصل بررسی کردیم می‌تواند در وضعیتی که یک شبکه‌ای جاده‌ای و مکان سرویس‌دهنده داده شده است، کاربرد داشته باشد. هدف از حل این مسئله این است که شبکه ترافیک را اصلاح کنیم طوری که سرویس‌دهی به مشتریان تا حد امکان متعادل گردد.

درخت T را با مجموعه رئوس V , $|V| = n$ و مجموعه یال‌ها E در نظر بگیرید. برای هر $e_i \in E$, $l_i \geq 0$ طول یال e_i است طول مسیر بین v_i و v_j در درخت T را با $d(v_i, v_j)$ نشان می‌دهیم. همچنین $x \in V$ مکان سرویس‌دهنده در درخت T است. در مسئله مکانیابی متعادل معکوس قصد داریم طول یال‌ها را با کمترین هزینه تغییر دهیم طوریکه تفاوت فاصله بین دورترین و نزدیکترین سرویس‌گیرنده از سرویس‌دهنده مینیمم شود. فرض کنید هزینه افزایش هر واحد یال e_i , $c_i^+ \geq 0$ و هزینه کاهش هر واحد یال e_i , $c_i^- \geq 0$ است. مقدارهای q_i^+ و q_i^- به ترتیب مقادیر افزایش و کاهش طول l_i است. برای هر $e_i \in E$, قرار دهید $\hat{l}_i = l_i + q_i^+ - q_i^-$. فرض شده است که برای هر یال $e_i \in E$, طول جدید یال نامنفی است یعنی $\hat{l}_i \geq 0$ یا $q_i^- \leq l_i$ است. فرض کنید

$$f_1 = \sum_{e_i \in E} (c_i^+ q_i^+ + c_i^- q_i^-). \quad (1.4)$$

$$f_2 = \max_{v_i \in V} \hat{d}(v_i, x) - \min_{v_i \in V} \hat{d}(v_i, x). \quad (2.4)$$

بنابراین در این مدل، از بین طول یال‌هایی اصلاح شده که تابع f_2 را مینیمم می‌کند، باید اصلاحاتی که تابع f_1 را مینیمم می‌کند، پیدا کنیم. در توابع بالا $\hat{d}(u, x)$ طول مسیر بین راس u و x با در نظر گرفتن طول یال‌ها اصلاح شده است.

۲.۴ مسئله با طول یال‌های نامحدود

در این بخش برای هر $e_i \in E$, طول یال‌ها را یعنی $\hat{l}_i \geq 0$ را نامحدود در نظر می‌گیریم. هم‌زمان مینیمم f_2 با کمترین هزینه را در نظر بگیرید. واضح است که در این حالت مقدار بهینه f_2 ، صفر است، یعنی طول همه مسیرها از x تا بقیه رئوس برابر هستند. بنابراین مدل زیر را می‌توانیم در نظر بگیریم.

$$\min f_1 = \sum_{e_i \in E} (c_i^+ q_i^+ + c_i^- q_i^-) \quad (1.4)$$

s.t.

$$\max_{v_i \in V} \hat{d}(v_i, x) - \min_{v_i \in V} \hat{d}(v_i, x) = 0. \quad (2.4)$$

$$\hat{l}_i \geq 0 \quad e_i \in E \quad (3.4)$$

بنابراین ما به دنبال یافتن یک جواب بهینه برای این مدل مسئله هستیم. فرض کنید P_1 و P_2 به ترتیب مجموعه همه مسیرها از x به دورترین و نزدیکترین راس در

$T \setminus \{x\}$ باشند. دقت کنید که P_1 و P_2 ثابت نیستند و در طول الگوریتم مرتبا تغییر خواهند کرد.

ملاحظات زیر را در نظر بگیرید.

ملاحظه ۱.۲.۴. یک جواب شدنی با کاهش طول بعضی از یال‌ها روی مسیرهایی در P_1 و یا افزایش طول بعضی از یال‌ها روی مسیرهایی در P_2 می‌تواند به دست آید.

ملاحظه ۲.۲.۴. اگر دو مسیر $p_i \in P_1$ و $p_j \in P_2$ وجود داشته باشد، طوریکه p_i شامل p_j باشد آنگاه هر تغییری روی یال‌ها p_j نمی‌تواند جواب شدنی را بهبود بخشد.

فرض کنید Adj مجموعه‌ای از یال‌های مجاور x باشد. در لم زیر، نشان می‌دهیم که اگر یک مسیر در P_2 بخشی از یک مسیر در P_1 باشد، بنابراین در جواب بهینه طول همه یال‌های غیر مجاور x صفر است.

لم ۱.۲.۴. اگر مسیری در P_1 وجود داشته باشد که شامل مسیری در P_2 باشد، آنگاه در جواب بهینه طول هر یال $e \notin Adj$ برابر با صفر است.

برهان. فرض کنید $p_i \in P_1$ شامل $p_j \in P_2$ باشد. پس با در نظر گرفتن ملاحظات ۱.۲.۴ و ۲.۲.۴ باید طول یال‌ها درون $p_i \setminus p_j$ را کاهش دهیم. آنگاه برای به دست آوردن جواب شدنی، طول همه یال‌های غیر مجاور x را مساوی صفر قرار می‌دهیم. $\square$

همچنین لم زیر را نیز می‌توانیم نتیجه بگیریم.

لم ۲.۲.۴. اگر فقط مجاز به کاهش طول یال‌ها باشید. آنگاه برای به دست آوردن یک جواب بهینه کافیست قرار دهید.

$$\hat{l}_i = \begin{cases} 0 & \text{if } e_i \notin Adj \\ Lm & \text{if } e_i \in Adj \end{cases} \quad (4.4)$$

که Lm طول هر مسیر در P_2 است. در این حالت هزینه پیدا کردن یک جواب بهینه به صورت

$$f_1 = \sum_{e_i \in Adj} c_i^-(l_i - Lm) + \sum_{e_i \notin Adj} c_i^- l_i. \quad (5.4)$$

محاسبه می‌شود.

برهان. چون فقط مجاز به کاهش طول یال هستیم بنابراین مشابه لم ۱.۲.۴ باید در جواب بهینه طول همه یال‌های غیر مجاور x مساوی صفر باشد. همچنین طول یال‌هایی که مجاور x هستند و در P_2 نیستند، باید به Lm کاهش بیابند. $\square$

لم ۳.۲.۴. اگر فقط مجاز به افزایش طول یال‌ها باشید. آنگاه برای بدست آوردن یک جواب شدنی روی درخت کافیت طول یال‌های مجاور x را افزایش دهیم.

در قضیه زیر نشان می‌دهیم که برای حل این مسئله روی درخت، کافیت این مسئله را روی یک زیر درخت کوچکتر حل کنیم.

لم ۴.۲.۴. حل این مسئله روی یک درخت با پیدا کردن یک جواب بهینه روی زیر درختی که از x و راس‌های مجاورش القا شده باشد، هم ارز است.

برهان. این برهان با این واقعیت نتیجه‌گیری می‌شود که برای به دست آوردن یک جواب شدنی باید طول بعضی از یال‌ها روی مسیرها در P_1 و P_2 به ترتیب کاهش و افزایش دهیم. در نهایت به یک مسیری در P_2 که بخشی از یک مسیر در P_1 است، خواهیم رسید. بنابراین با استفاده از لم ۱.۲.۴ در جواب بهینه باید طول همه یال‌های غیر مجاور x صفر شود.

□

فرض کنید T_x یک زیر درخت از T باشد که از x و راس‌های مجاورش القا شده باشد. در واقع T_x یک گراف ستاره‌ای است. بنابراین در بخش بعدی روی پیدا کردن یک جواب بهینه روی یک گراف ستاره‌ای متمرکز می‌شویم و نشان می‌دهیم جواب بهینه در زمان $O(n_x \log n_x)$ پیدا می‌شود که n_x تعداد یال‌ها T_x است. قضیه زیر برای این مسئله روی درخت نتیجه می‌شود.

قضیه ۱.۲.۴. فرض کنید d درجه x در درخت T باشد. آنگاه مسئله مکانیابی ۱- میانه معکوس با تعادل فاصله از مشتری‌ها به سرویس‌دهنده در زمان $O(d \log d)$ می‌تواند حل می‌شود.

توجه کنید که چون $d \leq n$ ، بنابراین $O(n \log n)$ کران بالای پیچیدگی روی درخت است.

۳.۴ تعادل فاصله روی گراف ستاره‌ای

فرض کنید T یک گراف ستاره‌ای و x راسی از گراف T با درجه بزرگتر از ۱ باشد. به این علت، این مدل را روی گراف ستاره‌ای در نظر گرفتیم که از طریق لم ۴.۲.۴ حل این مسئله روی درخت با حل این مسئله روی گراف ستاره‌ای اشاره شده در لم، هم ارز است. مجموعه از طولانی‌ترین و کوتاه‌ترین یال‌ها را در T به ترتیب با P_3 و P_4 در نظر بگیرید. برای بدست آوردن یک جواب بهینه روی T ، فرض کنید

$$d_1 < d_2 < \dots < d_r$$

یک ترتیب صعودی از فاصله بین رئوس T و x است. برای رسیدن به یک جواب شدنی از اقدامات زیر می‌توان استفاده کرد.

۱. طول هر یال P_3 که $j = 1, \dots, n_1$, $e_j \in P_3$ را کاهش دهیم. در این حالت برای تضمین اینکه همه یال‌ها در P_3 در تکرار بعدی، طولانی‌ترین یال در T باقی بمانند، بایستی به طور همزمان طول همه‌ی آن‌ها را کاهش دهیم. دقت کنید که طول هر یال در P_3 ، d_r است که باید به d_{r-1} کاهش دهیم. این کاهش، شدنی بودن را به اندازه $K_1 = d_r - d_{r-1}$ بهبود می‌دهد. هزینه این کاهش از

$$C_1 = \sum_{e_j \in P_3} c_j^- K_1. \quad (1.4)$$

محاسبه می‌شود.

۲. طول همه یال‌ها در P_4 افزایش یابد. در این حالت طول هر یال $e_i \in P_4$ که برابر با d_1 است تا d_2 افزایش پیدا می‌کند. بنابراین شدنی بودن به اندازه

$$K_2 = d_2 - d_1 \quad (2.4)$$

بهبود می‌یابد. هزینه این افزایش از

$$C_2 = \sum_{e_i \in P_4} c_i^+ K_2. \quad (3.4)$$

محاسبه می‌شود. در هر تکرار الگوریتم یکی از این دو حالت را انتخاب می‌کنیم. حالتی که بیشترین بهبود شدنی و کمترین هزینه را شامل می‌شود، انتخاب می‌گردد. درنظر بگیرید

$$\frac{K_1}{C_1} = \frac{1}{\sum_{e_j \in P_3} c_j^-}$$

و

$$\frac{K_2}{C_2} = \frac{1}{\sum_{e_i \in P_4} c_i^+}$$

بنابراین یک حالت با توجه به رابطه زیر انتخاب می‌شود.

$$\min \left\{ \sum_{e_i \in P_4} c_i^+, \sum_{e_j \in P_3} c_j^- \right\}$$

این ایده‌ها ما را به الگوریتم زیر برای یافتن یک جواب بهینه روی گراف ستاره‌ای سوق می‌دهد.

الگوریتم ISFBDS

ورودی: گراف ستاره‌ای T که x به عنوان راسی از T با درجه بزرگتر از ۱ و مکان سرویس‌دهنده، همچنین هزینه افزایش و کاهش طول یال‌ها داده شده است.

خروجی: طول جدید یال‌ها $\hat{l}_i$ طوریکه $f_2 = 0$ باشد.

مقداردهی اولیه:

- مجموعه $D = \{d_1, d_2, \dots, d_r\}$ که طول یال‌های T است را به صورت صعودی مرتب کنید. یعنی $(d_1 < d_2 < \dots < d_r)$.

• مجموعه‌ای از یال‌هایی که طول آنها d_1 و d_r است را به ترتیب با P_4 و P_3 نشان دهید.

• $s := 1, t := r, f_2 := d_r - d_1, f_1 := 0$.

• قرار دهید

$$C_1 := \sum_{e_j \in P_3} c_j^-$$

$$C_2 := \sum_{e_j \in P_4} c_j^+$$

• گام تکراری

• تا زمانی که $f_2 > 0$ کارهای زیر را انجام دهید.

۱. اگر $C_1 < C_2$ آنگاه

(a) برای هر یال $e_j \in P_3$ ، قرار دهید $l_j := d_{t-1}$.

$$f_1 := f_1 + C_1(d_t - d_{t-1}) \quad (b)$$

$$f_2 := f_2 - (d_t - d_{t-1}) \quad (c)$$

$$P := \{e_i \in T \mid l_i = d_{t-1}\} \quad (d)$$

$$P_3 := P_3 \cup P \quad (e)$$

$$C_1 := C_1 + \sum_{e_j \in P} c_j^-$$

$$t := t - 1 \quad (f)$$

در غیر این صورت

(a) برای هر یال $e_j \in P_4$ ، قرار دهید $l_j := d_{s+1}$.

$$f_1 := f_1 + C_2(d_{s+1} - d_s) \quad (b)$$

$$f_2 := f_2 - (d_{s+1} - d_s) \quad (c)$$

$$P := \{e_i \in T \mid l_i = d_{s+1}\} \quad (d)$$

$$P_4 := P_4 \cup P \quad (e)$$

$$C_2 := C_2 + \sum_{e_j \in P} c_j^+$$

$$s := s + 1 \quad (f)$$

حلقه پایان یابد.

الگوریتم پایان می‌یابد.

برای محاسبه پیچیدگی زمانی الگوریتم، فرض کنید تعداد راس‌های گراف ستاره‌ای T ، n_t باشد. بنابراین طول یال‌ها در T می‌توانند در زمان $O(n_t \log n_t)$ مرتب شوند. در بدترین حالت، ممکن به رفتن از طول یک یال از d_1 به d_2 و سپس d_3 و در نهایت d_{n_t} یا برعکس نیاز داشته باشیم. پس در این حالت، گام تکراری در زمان $O(n_t)$ تکرار خواهد شد. پس پیچیدگی زمانی الگوریتم $O(n_t \log n_t)$ است.

قضیه ۱.۳.۴. مسئله مکانیابی تک سرویس‌دهنده معکوس با تبادل روی فاصله از مشتری‌ها به سرویس‌دهنده روی گراف ستاره‌ای می‌تواند در زمان $O(n_t \log n_t)$ حل شود.

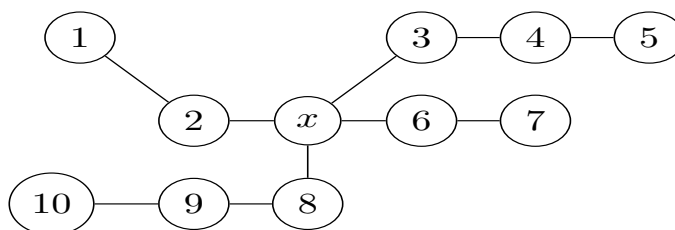
برای توضیح الگوریتم ارائه شده، مثال زیر را در نظر بگیرید.

مثال ۱.۳.۴. درخت T را در شکل ۱.۴ در نظر بگیرید، طول یال‌ها و هزینه افزایش و کاهش طول یال‌ها در جدول ۱.۴ داده شده است. با استفاده از لم ۴.۲.۴ کفایت طول یال‌های $l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, l_{10}$ را به صفر کاهش دهیم و یک جواب روی گراف ستاره‌ای بدست آمده از درخت T پیدا کنیم. هزینه کاهش درخت به گراف ستاره‌ای ۷۴ است. بنابراین فاصله رئوس گراف ستاره‌ای فوق با x را به ترتیب صعودی به صورت زیر مرتب می‌کنیم.

$$1 < 3 < 5$$

بنابراین $P_4 = \{e_6\}$ و $f_1 = 0, f_2 = 4, P_2 = \{e_3\}$ و چون $C_1 = 2 < C_2 = 5$ پس $f_1 = 4, f_2 = 2$ و $l_3 = 3$ است.

در تکرار بعدی $P_4 = \{e_6\}$ و $P_2 = \{e_3, e_2, e_8\}$ پس $C_1 = 2 + 1 + 2$ و $C_2 = 5$ است. چون $C_1 = C_2$ پس می‌توانیم طول یال‌ها در P_2 را کاهش یا طول یال در P_4 را افزایش دهیم. فرض کنید قرار دهیم $l_6 = 3$ در این حالت $f_1 = 14, f_2 = 0$ و الگوریتم متوقف می‌شود. هزینه نهایی پیدا کردن جواب بهینه روی درخت اصلی ۸۸ است.



شکل ۱.۴: درخت T با ۱۰ راس

۴.۴ مسئله با طول یال‌های محدود

در این بخش حالتی را در نظر می‌گیریم که طول یال‌ها محدود هستند. یعنی برای هر $e_i \in E$ که $\circ < \underline{l}_i \leq \hat{l}_i \leq \bar{l}_i$ به ترتیب کران بالا و پایین طول یال e_i است. در این حالت مقدار بهینه f_2 ممکن است صفر نشود.

فرض کنید $T = (V, E)$ یک گراف بدون جهت باشد. نمادهای P_1, P_2, Lm و Adj مشابه با بخش ۲.۴ تعریف می‌شوند. بنابراین ملاحظات ۱.۲.۴ و ۲.۲.۴ و لم ۳.۲.۴ برای حالتی که طول یال‌های محدود است، برقرار است. چون فرض کردیم $\circ < \underline{l}_i$ پس ملاحظه زیر نیز برقرار است.

ملاحظه ۱.۴.۴. هر مسیری در P_2 تنها شامل یک یال است که آن یال مجاور x است.

لم ۱.۴.۴. در حالت طول یال‌های محدود، اگر یک مسیر در P_2 وجود داشته باشد که شامل بخشی از یک مسیر در P_1 باشد. آنگاه در جواب بهینه یک مسیر $p_i \in P_1$ موجود است به طوری که برای هر یال $e_r \in p_i \setminus Adj$ داریم.

$$\hat{l}_r = \underline{l}_r$$

برهان. فرض کنید مسیری در P_2 وجود دارد که شامل بخشی از یک مسیر در P_1 است. با استفاده از ملاحظه ۲.۲.۴ بایستی طول یال‌هایی را که فقط روی مسیرهایی که در P_1 هستند، کاهش دهیم. پس طول یال‌هایی که مجاور x نیستند، در همه مسیرهای در P_1 را کاهش

جدول ۱.۴: طول و هزینه اصلاح طول یال‌ها در درخت T .

e_i	$\underline{l}_i$	c_i^+	c_i^-
$e_1 = (v_1, v_2)$	۳	۱	۲
$e_2 = (v_2, x)$	۳	۳	۱
$e_3 = (x, v_3)$	۵	۲	۲
$e_4 = (v_3, v_4)$	۲	۱	۳
$e_5 = (v_4, v_5)$	۸	۱	۴
$e_6 = (x, v_6)$	۱	۵	۳
$e_7 = (v_6, v_7)$	۲	۳	۱
$e_8 = (x, v_8)$	۳	۵	۲
$e_9 = (v_8, v_9)$	۴	۳	۴
$e_{10} = (v_9, v_{10})$	۳	۴	۴

می‌دهیم، این کاهش تا زمانی است که طول همه یال‌ها از یک مسیر در P_1 که در Adj نیستند، به کران پایین خودشان می‌رسند. $\square$

قبل از توضیح الگوریتم، به تعریف زیر نیاز داریم.

تعریف ۱.۴.۴. یک مسیر p در P_1 را اشباع شده می‌نامیم، اگر طول همه‌ی یال‌ها در p در کران پایین خودشان باشند. به عبارت دیگر

$$\forall e_t \in p, l_t = \underline{l}_t$$

و همچنین یک مسیر q در P_2 را اشباع شده می‌نامیم، اگر طول همه‌ی یال‌ها در q در کران بالا خودشان باشند. به عبارت دیگر

$$\forall e_r \in q, l_r = \bar{l}_r$$

ابتدا فرض کنید مسیری در P_1 وجود ندارد که شامل مسیری در P_2 باشد. مشابه با بخش ۳.۴ فرض کنید

$$d_1 < d_2 < \dots < d_r$$

یک ترتیب صعودی از فاصله بین رئوس T و x باشد. برای رسیدن به یک جواب شدنی دو حالت زیر می‌تواند استفاده شود.

۱. **کاهش.** طول هر یال روی هر مسیر در P_1 ، $j = 1, \dots, n_1$ که $|P_1| = n_1$ را کاهش دهیم. در این حالت برای تضمین اینکه همه مسیرها در P_1 طولانی‌ترین مسیر از x باقی بمانند، به طور همزمان طول همه‌ی آنها را کاهش می‌دهیم. اگر یک مسیر اشباع شده در P_1 وجود داشته باشد پس نمی‌توانیم طول طولانی‌ترین مسیر را کاهش دهیم بنابراین این حالت (کاهش طول مسیرها در P_1) را کنار می‌گذاریم. در غیر این صورت، در هر مسیر $p_j \in P_1$ ، $j = 1, \dots, n_1$ ، یالی با کمترین هزینه کاهش را انتخاب می‌کنیم و طول آن را کاهش می‌دهیم. توجه کنید که اگر این یال در Adj باشد، طول آن از Lm نمی‌تواند کمتر باشد. بنابراین برای $j = 1, \dots, n_1$ قرار دهید

$$h_j = \text{Argmin}\{c_h^- \mid e_h \in p_j, l_h > \underline{l}_h\}$$

و

$$RL_j = \begin{cases} \min\{l_{h_j} - Lm, l_{h_j} - \underline{l}_{h_j}\} & \text{if } e_{h_j} \in Adj \\ l_{h_j} - \underline{l}_{h_j} & \text{if } e_{h_j} \notin Adj \end{cases} \quad (1.4)$$

بنابراین طول هر e_{h_j} را به مقدار K_1 کاهش می‌دهیم که

$$K_1 = \min\{d_r - d_{r-1}, RL_1, \dots, RL_{n_1}\},$$

مقدار بهبود شدنی است. این کاهش به هزینه زیر نیاز دارد.

$$C_{P_1} = \sum_{p_j \in P_1} c_{h_j}^- K_1. \quad (2.4)$$

۲. افزایش. طول همه یال‌ها در مسیرهای درون P_2 را افزایش می‌دهیم. در این حالت طول هر یال e_t روی مسیر $p_i \in P_2$ با d_1 برابر است. توجه کنید که p_i فقط شامل یال e_t است. بنابراین شدنی بودن به اندازه

$$K_2 = \min\{d_2 - d_1, \bar{l}_1 - l_1, \dots, \bar{l}_{n_2} - l_{n_2}\}, \quad (3.4)$$

که $|P_2| = n_2$ ، بهبود می‌یابد. هزینه این افزایش طول یال از طریق فرمول زیر محاسبه می‌شود.

$$C_{P_2} = \sum_{p_i \in P_2, e_t \in p_i} c_t^+ K_2. \quad (4.4)$$

در هر تکرار باید حالتی که بیشترین بهبود شدنی و کمترین هزینه را شامل می‌شود را انتخاب کرد. در روشی مشابه با بخش ۳.۴ حالتی متناظر با

$$\min\left\{\sum_{p_j \in P_1} c_{h_j}^-, \sum_{p_i \in P_2, e_t \in p_i} c_t^+\right\}.$$

انتخاب می‌شود.

در این حالت یعنی حالتی که مسیری در P_1 وجود ندارد که شامل مسیری در P_2 باشد، اگر شرایط زیر برقرار باشد، الگوریتم پایان می‌یابد.

(A). یک یال مجاور x در P_2 به کران بالای خودش برسد. به عبارت دیگر یک مسیر اشباع شده در P_2 وجود دارد.

(B). یک مسیر $p_r \in P_1$ که همه یال‌های غیر مجاور x از p_r به کران پایین خودشان برسند و طول یال در $p_r \cap Adj$ به کران پایین خودشان یا Lm برسد. به عبارت دیگر فرض کنید $e_a \in p_r$ یال مجاور x باشد، پس برای هر

$$e_t \in p_r, e_t \neq e_a, \hat{l}_t = \underline{l}_t$$

$$l_a = \max\{\underline{l}_a, Lm\}$$

اکنون حالتی را در نظر بگیرید که مسیری در P_1 شامل مسیری در P_2 باشد. بنابراین در تکرار فعلی و تکرارهای بعد فقط کاهش طول یال در نظر گرفته می‌شود. اگر شرایط (B) برقرار باشد الگوریتم پایان می‌یابد. این ایده ما را به الگو زیر سوق می‌دهد.

الگوریتم ISFBDB

ورودی:

درخت $T = (V, E)$ با طول یال‌های محدود و همچنین هزینه افزایش و کاهش طول یال‌ها داده شده است و راس x به عنوان مکان سرویس‌دهنده در نظر گرفته شده است.

خروجی

طول جدید یال‌ها $\hat{l}_i$

مقدار دهی اولیه

مجموعه‌هایی از طولانی‌ترین و کوتاه‌ترین مسیر بین x و رئوس دیگر درخت T را به ترتیب P_1 و P_2 بنامید.

اولین و دومین راسی که بیشترین فاصله را با x دارند، پیدا می‌کنیم، فاصله آن‌ها را با x به ترتیب با d_{max1} و d_{max2} بنامید.

اولین و دومین راسی که کمترین فاصله را با x دارند، پیدا می‌کنیم، فاصله آن‌ها را با x به ترتیب با d_{min1} و d_{min2} بنامید.

فرض کنید

$$Adj = \{e_i \in E \mid \text{است } x \text{ مجاور } e_i\}.$$

قرار دهید $f_1 := 0$, $f_2 := d_{max1} - d_{min1}$, $Lm := d_{min1}$, $n_1 := |P_1|$ و $n_2 := |P_2|$.

گام تکراری

۱. اگر یک مسیر اشباع شده در P_1 و یک مسیر اشباع شده در P_2 وجود داشته باشد، پس الگوریتم پایان می‌یابد و طول یال‌های فعلی بهینه است.

۲. اگر یک مسیر اشباع شده در P_2 وجود داشته باشد و شرط (B) برقرار باشد. توقف کنید و طول یال‌های فعلی بهینه است.

۳. اگر یک مسیر اشباع شده در P_1 وجود داشته باشد، پس

(a) قرار دهید

$$C_{P_2} := \sum_{p_i \in P_2, e_t \in p_i} c_t^+$$

(b) با استفاده از (۳.۴) مقدار K_2 را محاسبه کنید.

(c) برای هر یال e_t روی هر مسیر $p_i \in P_2$ ، قرار دهید $l_t := l_t + K_2$.

$$f_1 := f_1 + C_{P_1} K_2 \quad (d)$$

$$f_2 := f_2 - K_2 \quad (e)$$

$$d_{min1} = d_{min1} + K_2 \quad (f)$$

در غیر این صورت

(a) اگر مسیری در P_1 وجود داشته باشد که شامل مسیری در P_2 باشد. آنگاه

(i) اگر شرط (B) برقرار باشد، پس الگوریتم متوقف می شود و طول فعلی یال ها بهینه است.

(ii) برای هر $p_j \in P_1$ ، $j = 1, \dots, n_1$ قرار دهید

$$h_j := \text{Argmin}\{c_h^- | e_h \in p_j, l_h > \underline{l}_h\},$$

$$C_{P_1} := \sum_{p_j \in P_1} c_{h_j}^-$$

(iii) قرار دهید $L_m := d_{min1}$ و $K_1 := \min\{d_r - d_{r-1}, RL_1, \dots, RL_{n_1}\}$ که برای RL_j ، $j = 1, \dots, n_1$ در رابطه (۱.۴) تعریف شده است.

(iv) برای $j = 1, \dots, n_1$ قرار دهید $l_{hj} := l_{hj} - K_1$.

$$f_1 := f_1 + C_{P_1} K_1 \quad (v)$$

$$f_2 := f_2 - K_2 \quad (vi)$$

$$d_{max1} = d_{max1} - K_1 \quad (vii)$$

در غیر این صورت

(i) برای هر $p_j \in P_1$ ، $j = 1, \dots, n_1$ قرار دهید

$$h_j := \text{Argmin}\{c_h^- | e_h \in p_j, l_h > \underline{l}_h\}$$

$$C_{P_2} := \sum_{p_i \in P_2, e_t \in p_i} c_t^+ \text{ و } C_{P_1} := \sum_{p_j \in P_1} c_{h_j}^- \quad (ii)$$

(iii) اگر $C_{P_1} < C_{P_2}$ پس

(A) قرار دهید $K_1 := \min\{d_r - d_{r-1}, RL_1, \dots, RL_{n_1}\}$ ، که برای $RL_j, j = 1, \dots, n_1$ در رابطه (۱.۴) تعریف شده است.

(B) برای $j = 1, \dots, n_1$ قرار دهید. $l_{hj} := l_{hj} - K_1$

(C) $f_1 := f_1 + C_{P_1} K_1$

(D) $f_2 := f_2 - K_2$

(E) $d_{max_1} = d_{max_1} - K_1$

در غیر این صورت

(A) با استفاده از (۳.۴) مقدار K_2 را محاسبه کنید.

(B) برای هر یال e_t روی هر مسیر $p_i \in P_2$ ، قرار دهید $l_t := l_t + K_2$

(C) $f_1 := f_1 + C_{P_2} K_2$

(E) $f_2 := f_2 - K_2$

(F) $d_{min_1} = d_{min_1} + K_2$

(b) اگر $f_2 = 0$ توقف کنید. جواب بهینه به دست آمده است.

در غیر این صورت

(i) P_2 و P_1 که به ترتیب مجموعه‌هایی از طولانی‌ترین و کوتاه‌ترین مسیر بین x و رئوس در درخت T است را پیدا کنید.

(ii) d_{min_2} و d_{max_2} که به ترتیب دومین رئوسی هستند که بیشترین و کمترین فاصله را با x دارند، را پیدا کنید.

(iii) به مرحله ۱ بروید.

الگوریتم پایان یافته است.

برای محاسبه پیچیدگی الگوریتم ISFBDB، توجه کنید طولانی‌ترین و کوتاه‌ترین مسیرها در زمان $O(n)$ می‌تواند محاسبه شود ([۴۲]، [۱۲]) و هر گام الگوریتم حداکثر زمان $O(n)$ نیاز دارد. در هر تکرار f_r از d_r به d_{r-1} کاهش می‌یابد یا یک طول یال به کران پایین یا بالای خودش می‌رسد. بنابراین گام تکراری الگوریتم در حداکثر زمان $O(n)$ تکرار می‌شود، در نتیجه پیچیدگی زمانی الگوریتم $O(n^2)$ است.

قضیه ۱.۴.۴. مسئله مکانیابی تک سرویس‌دهنده معکوس تعادل روی فاصله در حالت طول یال‌های محدود روی درخت در زمان $O(n^2)$ حل می‌شود.

مثال ۱.۴.۴. درخت T را در مثال ۱.۳.۴ در نظر بگیرید. کران‌های بالا و پایین طول یال‌ها در جدول ۲.۴ آورده شده است.

فاصله دورترین و نزدیکترین راس به x به ترتیب ۱۵ و ۱ است. بنابراین $p_1 = \{e_3, e_4, e_5\}$ که $P_1 = \{p_1\}$ و $d_{max1} = 15$ ، $d_{min1} = 1$ ، $f_1 = 14$ ، $P_1 = \{p_1\}$ و $p_2 = \{e_6\}$ در تکرار اول مسیری اشباع شده‌ای وجود ندارد و مسیر p_1 شامل مسیر p_2 نیست. بنابراین در این تکرار، C_{P_1} و C_{P_2} محاسبه می‌کنیم. از بین یال‌ها در p_1 ، یال e_3 کمترین هزینه اصلاح را دارد بنابراین $h_1 = 3$ و $RL_1 = \min\{l_3 - d_{min1}, l_3 - \underline{l}_3\} = 4$ است. $C_{P_1} = c_3^-$ و $C_{P_2} = c_6^+$ چون $C_{P_1} = 2 < C_{P_2} = 5$ پس

$$K_1 = \min\{15 - 10, 4\} = 4, \quad l_3 = 1, \quad f_1 = 8, \quad f_2 = 10$$

جدول ۲.۴: کران طول یال‌ها در درخت T

e_i	l_i	$\underline{l}_i$	$\bar{l}_i$
$e_1 = (v_1, v_2)$	۳	۱	۴
$e_2 = (v_2, x)$	۳	۲	۶
$e_3 = (x, v_3)$	۵	۱	۷
$e_4 = (v_3, v_4)$	۲	۱	۵
$e_5 = (v_4, v_5)$	۸	۴	۱۰
$e_6 = (x, v_6)$	۱	۱	۴
$e_7 = (v_6, v_7)$	۲	۱	۵
$e_8 = (x, v_8)$	۳	۱	۷
$e_9 = (v_8, v_9)$	۴	۲	۸
$e_{10} = (v_9, v_{10})$	۳	۱	۶

جدول ۳.۴: نتایج تکرار برای مثال ۱.۴.۴

$iteration$	P_1	P_2	C_{P_1}	C_{P_2}	f_1	f_2
۱	$\{p_1\}$	$\{p_2\}$	$c_3^- = 2$	$c_6^+ = 5$	۸	۱۰
۲	$\{p_1\}$	$\{p_2, p_3\}$	$c_4^- = 3$	—	۱۱	۹
۳	$\{p_1, p_4\}$	$\{p_2, p_3\}$	$c_5^- + c_8^- = 6$	—	۲۳	۷
۴	$\{p_1, p_4\}$	$\{p_2, p_3\}$	$c_5^- + c_9^- = 8$	—	۳۹	۵

جدول ۴.۴: طول بهینه یال‌ها برای مثال ۱.۴.۴

e_i	$optimal \ l_i$	$\underline{l}_i$	$\overline{l}_i$
$e_1 = (v_1, v_2)$	۳	۱	۴
$e_2 = (v_2, x)$	۳	۲	۶
$e_3 = (x, v_3)$	۱	۱	۷
$e_4 = (v_3, v_4)$	۱	۱	۵
$e_5 = (v_4, v_5)$	۴	۴	۱۰
$e_6 = (x, v_6)$	۱	۱	۴
$e_7 = (v_6, v_7)$	۲	۱	۵
$e_8 = (x, v_8)$	۱	۱	۷
$e_9 = (v_8, v_9)$	۲	۲	۸
$e_{10} = (v_9, v_{10})$	۳	۱	۶

در تکرار دوم مسیر $p_3 = \{e_3\}$ به P_2 اضافه می‌شود. این مسیر یک مسیر اشباع شده است، اما چون شرط B برقرار نیست، الگوریتم پایان نمی‌یابد. از طرفی چون p_1 شامل p_3 است پس فقط مقدار C_{P_1} را به دست می‌آوریم. چون $h_1 = 4$ پس l_4 باید کاهش یابد. بنابراین $C_{P_1} = c_4^- = 3$ و $K_1 = \min\{11-10, 2-1\} = 1$ پس $l_4 = 1$, $f_1 = 11$, $f_2 = 9$ است.

در تکرار سوم مسیر $p_4 = \{e_8, e_9, e_{10}\}$ به P_1 اضافه می‌شود. مسیر p_3 یک مسیر اشباع شده است و شرط B برقرار نیست و p_1 شامل p_3 است پس دوباره فقط مقدار C_{P_1} را محاسبه می‌کنیم. چون $h_1 = 5$, $h_4 = 8$ پس l_5, l_8 باید کاهش یابد. بنابراین $RL_1 = 4$, $RL_4 = 2$, $C_{P_1} = c_5^- + c_8^- = 6$, $K_1 = \min\{10-7, 4, 2\} = 2$, $l_5 = 6$ و $f_2 = 7$ به دست می‌آید.

در تکرار چهارم $f_1 = 39$, $l_1 = 2$, $l_5 = 4$ و $f_2 = 5$ بدست خواهیم آورد. بنابراین $p_1 = \{e_3, e_4, e_5\}$ یک مسیر اشباع شده و جواب فعلی بهینه است. نتایج تکرارها در جدول ۳.۴ آورده شده است.

طول یال‌ها در جواب بهینه در جدول ۴.۴ آورده شده است.

پیشنهادهای

برای انجام تحقیقات آتی، مطالعه و بررسی موارد ذیل پیشنهاد می‌گردد:

- مطالعه و بررسی مسئله ۲- میانه متعادل در حالت پیوسته با محدودیت بودجه و ارائه روش حل.
- مطالعه و بررسی مسئله مکانیابی مذکور با در نظر گرفتن تابع هزینه تحت نرم چپیشف و نرم همینگ و ارائه الگوریتم برای حل.
- مطالعه و بررسی مسائل مکانیابی متعادل معکوس با شرطهای تعادل دامنه تغییرات، تعادل واریانس.
- مطالعه و بررسی مسئله مکانیابی متعادل معکوس روی صفحه و شبکه‌های دیگر.
- ترکیب مدل‌های دیگر معکوس با مسائل مکانیابی متعادل، مطالعه و بررسی مدل‌های ترکیبی.

مراجع

- [1] Ahuja R.K. and Orlin J.B. (2001), "Inverse optimization", **Operations Research**, 49,771-783
- [2] Alizadeh B. and Burkard R.E. (2009), "Inverse 1-center location problems with edge length augmentation on trees", **Computing**, 86,331-343
- [3] Alizadeh B. and Burkard R.E. (2011), "Combinatorial algorithms for inverse absolute and vertex 1-center location problems on trees", **Networks**, 58,190-200
- [4] Alizadeh B. and Burkard R.E. (2011), "Uniform-cost inverse absolute and vertex center location problems with edge length variations on trees", **Discrete Applied Mathematics**, 159,706-716
- [5] Alsuwaiyel M.H. (1999), "**Algorithm: Design Techniques and Analysis**", World Scientific, New Jersey.
- [6] Baroughi Bonab F, Burkard R.E. and Alizadeh B. (2010), "Inverse median location problems with variable coordinates", **Central European Journal of Operations Research**, 18,365-381
- [7] Baroughi Bonab F, Burkard R.E. and Gassner E. (2011), "Inverse. p-median problems with variable edge lengths", **Mathematical Methods of Operations Research**, 73,263-280
- [8] Bazaraa M. S, Sherali H. and Jarvis J. (2005), "Linear programming and network flows", **third ed. John Wiley and Sons**
- [9] Berman O, Drezner Z, Tamir A, and Wesolowsky G.O. (2009), "Optimal location with equitable loads", **Ann Oper Res**, 167,307-326.

-
- [10] Berman O, Ingco D.I. and Odoni A.R. (1994), " Improving the location of minmax facility through network modification", **Networks**, 24,31-41
 - [11] Berman O, Ingco D.I. and Odoni A. R. (1992), " Improving the location of minisum facility through network modification", **Annals of Operations Research**, 40,1-16
 - [12] Bulterman R.W, Van-Der-Sommen F.W, Zwaan G, Verhoeff T, Van-Gasteren A.J. M and Feijen W.H.J. (2002), " On computing a longest path in a tree", **Information Processing Letters**, 81,93-96
 - [13] Burkard R.E, Pleschiutchnig C. and Zhang J. (2004), " Inverse median problems", **Discrete Optimization**, 1,23–39.
 - [14] Burkard R.E, Gassner E. and Hatzl J. (2006), " A linear time algorithm for the reverse 1-median problem on a cycle", **Networks**, 48, 16-23.
 - [15] Burkard R.E, Gassner E. and Hatzl J. (2008), " Reverse 2-median problems on trees", **Discrete applied mathematic**, 156, 1963-1976.
 - [16] Burkard R. E, Galavii M. and Gassner E. (2010), " The inverse Fermat-Weber problem", **European Journal of Operation Research**, 206,11-17.
 - [17] Burton D. and Toint Ph.L. (1992), " On an instance of the inverse shortest paths problem", **Mathematical Programming**, 53,45–61
 - [18] Burton D. and Toint Ph.L. (1994), " On the use of an inverse shortest paths algorithm for recovering linearly correlated costs", **Mathematical Programming**, 63,1–22.
 - [19] Cai M.C, Yang X.G. and Zhang J.Z. (1999), " The complexity analysis of the inverese center location problem", **Journal of Global Optimization**, 15,213–218
 - [20] Cheriyan J. and Ravi R. (1998)," **Aproximation for Networks Problems**", University of Waterloo press, Canada
 - [21] Cormen T.H, Leiserson C.E, Rivest R.L. and Stein C. (2001), "**Introduction to Algorithms**", MIT Press, Cambridge

- [22] Corra E.S , Steiner M.T.A, Freitas A.A. and Cornier C. (2001), " A genitic algorithm for the p - median problem" , Available in Address: seer.nj.ncg.com/458099.html.bridge.

Cite
- [23] Dibble C. and Densham P.J. (1993), " **Generating interesting alternatives in GIS and SDSS using genetic algorithms, GIS/LIS Symposium**", University of Nebraska, Lincoln.
- [24] Drezner Z. (1984), " The planar two-center and two-median problems", **Transportation Science**, 18,351-61.
- [25] Drezner Z. (1992), " A note on the Weber location problem", **Annals of Operations Research**, 40,153–161.
- [26] Drezner Z. (2013), " The fortified Weiszfeld algorithm for solving the Weber problem", **IMA Journal of Management Mathematics**, 26,1–9.
- [27] Eiselt H.A. (2011), "**Foundation of Location Analysis**", Springer Verlag, New York
- [28] Erkut E. (1993), " Inequality measures for location problems", **Location Science**, 1,199–217
- [29] Erkut E, Bozkaya B. and Zhang J. (2001), " An effective genetic algorithm for the p-median problem", **Preprint**
- [30] Francis R. , McGinnis Jr.L.F. and White J.A. (1992), " Facility Layout and Location: An Analytical Approach", **Prentice Hall**
- [31] Galavii M. (2010), " The inverse 1-median problem on a tree and on a path", **Electronic Notes in Discrete Mathematics**, 36, 1241-1248
- [32] Garey M.R. and Johnson D.S. (1979), "**Computers and intractability: A guid to the theory of NP-Completeness: Freeman**" , San Francisco, California.
- [33] Gassner E. (2007), " Up- and downgrading the 1-median in a network", **Technical Report**, Graz University of Technology

-
- [34] Gassner E. (2009), " A game-theoretic approach for downgrading the 1-median in the plane with Manhattan metric", **Annals of Operations Research** , 172, 393–404
 - [35] Gavish B. and Sridhar S. (1995), " Computing the 2-median on tree networks in $O(n \log n)$ time", **Networks**, 26,305-317
 - [36] Goldman A.J. (1971), " Optimal center location in simple networks", **Transportation Science**, 5,212-221.
 - [37] Goldman A.J. and Dearing P.M. (1975), " Concepts of optimal location for partially noxious facilities", **ORSA Bulletin**, 23,B31.
 - [38] Goldman A.J. (1979), " Optimal center location in simple networks", **transportation Sci**, 5, 212–221.
 - [39] Guan X.C. and Zhang B.W. (2012), " Inverse 1-median problem on trees under weighted Hamming distance", **Journal of Global Optimization**, 54,75–82.
 - [40] Hakimi S. L. (1964), " Optimum location of switching centers and the absolute centers and medians of a graph" , **Operations Research**, 12,450-459.
 - [41] Hakimi S.L. (1965), " Optimum distribution of switching centers in a communication network and some related graph theoretic problems", **Operations Research**, 13,462–45.
 - [42] Handler G.Y. (1973), " Minimax location of a facility in an undirected tree networks", **Transportation Sci**, 7,287-293 .
 - [43] Hansen P. and Meladenovic N. (1997), " Variable neighborhood search for the p-median", **Location SCi**, 207-226.
 - [44] Heuberger C. (2004), " Inverse combinatorial optimization: A survey on problems, methods, and results", **Journal of Combinatorial Optimization**, 8,329-361
 - [45] Hosage M. and Goodchild M.F. (1986), " Discrete space location-allocation solutions from genetic algorithms", **Ann. Oper. Res**, 6 35-46.

- [46] Jianfang Y. and Juan J. (2012), " Reverse 1-median problem with constraint in trees", The 2nd International Conference on Computer Application and System Modeling.
- [47] Jeger M. and Kariv O. (1985), " Algorithms for finding p-centers on a weighted tree (for relatively small p)", **Networks**, 15,381–389
- [48] Kariv O, Hakimi S.L. (1979), " An algorithmic approach to network location problems . Part I: The p-centers", **SIAM Journal of Applied Mathematics**, 37,513–538
- [49] Love R.F, Morris J.G. and Wesolowsky G.O. (1988), " **Facilities Location: Models and Methods**", Elsevier, New York
- [50] Maranzana F.E. (1964)," On the location of supply points to minimize transport costs", **Oper. Res Quarterly**, 15,261-270.
- [51] Marín A.(2011) , " The discrete facility location problem with balanced allocation of customers", **European journal of operational research**, 210,27-38.
- [52] Megiddo N. and Supowitz K. (1984), " On the complexity of some common geometric location problems", **SIAM Journal on Computing**, 13,182-196.
- [53] Mirchandani B.P. and Francis R.L. (1990), " **Discrete Location Theory**", John Wiley New York.
- [54] Mirchandani B.P. and Qudjit A. (1980), " Localizing 2-median on probabilistic and deterministic tree networks", **Networks**, 10, pp,329–350.
- [55] Moreno-Perez J.A, Moreno-Vega J.M. and Meladenovic N. (1994), "Tabu search and simulated annealing in p-median problems", Talk at the Canadian Operational Research Society Conference, Montral.
- [56] Nazari M. and Fathali J. (2019), " Inverse and reverse 2-facility location problems with equality measures on a network" **Iranian Journal of Mathematical Sceiences and Informatices**
- [57] Nguyen K.T, Hamacher H.W. and Burkard R.E. (2014), " **Inverse Location Theory with Ordered Median Function and Other Extensions**", epubli GmbH, Berlin.

- [58] Nguyen K.T. and Anh L.Q. (2015), "Inverse k-centrum problem on trees with variable vertex weights", **Mathematical Methods of Operational Research**, 82,19-30.
- [59] Nguyen K.T. and Sepasian A.R. (2015), "The inverse 1-center problem on trees with variable edge lengths under Chebyshev norm and Hamming distance", **Journal of Combinatorial Optimization**, DOI: 10.1007/s10878-015-9907-5.
- [60] Nguyen K.T. and Chassein A. (2015), "Inverse eccentric vertex problem on networks", **Central European Journal of Operations Research**, 23,687-698 .
- [61] Nguyen K.T. (2016), "Inverse 1-median problem on block graphs with variable vertex weights", **Journal of Optimization Theory and Applications**, 168,944-957.
- [62] Nguyen K.T, Chassein A. (2015), "The inverse convex ordered 1-median problem on trees under Chebyshev norm and Hamming distance", **European Journal of Operational Research**, 247,774-781.
- [63] Nguyen K.T. and Chi N.T.L. (2016) ," A model for the inverse 1-median problem on trees under uncertain costs", **Opuscula Mathematica**, 36,513-523.
- [64] Plastria F. (2014), "Up- and downgrading the euclidean 1-median problem and knapsack Voronoi diagrams", **Annals of Operations Research**, DOI: 10.1007/s10479-014-1587-5.
- [65] Rolland E, Schilling D.A. and Current J.R. (1997)," An efficient tabu search heuristic for the p-median problem", **EJOR**, 96,329-342.
- [66] Sepasian A.R. and F. Rahbarnia. (2015), "An $O(n \log n)$ algorithm for the inverse 1-median problem on trees with variable vertex weights and edge reductions", **Optimization**, 64,595-602.
- [67] Sepasian A.R. and F. Rahbarnia. (2015), "Upgrading p-Median Problem on a Path", **Journal of Mathematical Modelling and Algorithms in Operations Research**, 14,145-157.

- [68] Tamir A. (1988), " Improved complexity bounds for center location problems on networks by using dynamic data structures", **SIAM Journal on Discrete Mathematics**, 1,377–396.
- [69] Tamir A. (1996), " An $O(pn^2)$ algorithm for the p-median and related problems on tree graphs", **Operations Research Letters**, 19,59-64.
- [70] Teitz M.B. and Bart P. (1968), " Heuristic methods for estimating generalized vertex median of a weighted graph", **Oper. Res**, 16,955-961.
- [71] Wang Q. and Bai Y. (2010), " An Efficient Algorithm for Reverse 2-median Problem on A Cycle", **Journal of Networks**, 5,1169-1176.
- [72] Ward J. E and Wendell R.E. (1980)," A new norm for measuring distance which yields linear location problems", **Operation Research**, 28,836–844.
- [73] Yang X. and Zhang J. (2008), " Inverse center location problem on a tree", **Journal of System Science and Complexity**, 21,651–664.
- [74] Zanjirani R. and Hekmatfar M. (2009), "**Facility location: Concepts, Models, Algorithms and Case Studies**", Physica-Verlag, Berlin.
- [75] Zhang J, iu Z . and Ma Z. (2000), " Some reverse location problems", **European Journal of Operational Research**, 124,77–88.

Aabstract

In this thesis, we combine inverse location and balanced location problem. in a model the inverse and reverse balanced facility location problems with considering the variable edge lengths. The aim of the inverse problem is modifying the length of edges with minimum cost, such that the difference between the maximum and minimum weights of clients, allocated to the given facilities is minimized. On the other hand, the reverse case of the balanced facility location problem considers the modifying the lengths of edges with a given budget constraint, such that the difference between the maximum and minimum weights of vertices, allocated to the given facilities is reduced as much as possible. Two algorithms with time complexity $O(n \log n)$ are presented for the inverse and reverse balanced 2-facility location problems.

in addition, we introduce a case of inverse single facility location problem on a tree and we want to modify in the length of edges with minimum cost, such that the difference of distances between the farthest and nearest clients to a given facility is minimized. Two cases are considered: bounded and unbounded nonnegative edge lengths. In the unbounded case, we show the problem can be reduced to solve the problem on a star graph. Then an $O(n \log n)$ algorithm is developed to find the optimal solution. For the bounded edge lengths case an algorithm with time complexity $O(n^2)$ is presented.

Keywords: Facility location problem, Inverse Facility location, Balanced location, Median problem.



Faculty Of Mathematical Sciences

PhD Thesis in Operations of Research

inverse balanced location problems

By: Shahede Omid Noorabadi

Supervisor:

Jafar Fathali

Advisor:

Mehrdad Ghzanvi

July 2021