

رسالة محمد



دانشکده علوم ریاضی

رشته ریاضی کاربردی، گرایش تحقیق در عملیات

رساله دکتری

بررسی مسائل مکان‌یابی چند وسیله‌ای معکوس پشتیبان

نگارنده: مرتضی نظری

استاد راهنما

دکتر جعفر فتحعلی

بهمن ۱۳۹۷

تقدیم به

خوشیدینام امام زمانم،

که جهان از نور وجودش روشن است و شقایق با به بهانه دیدنش جمعه بار انتظار می کشد. تقدیم به
همسرم

و

پدر و مادرم

خدا... ۱

به من زیستنی عطا کن که در لحظه مرگ، بر بی‌ثمری لحظه‌ای که برای زیستن گذشته است، حسرت نخورم و مُردنی عطا کن که بر بیهودگی‌ش، سوگوار نباشم. بگذار تا آن را، خود انتخاب کنم، اما آنچنان که تو دوست می‌داری.

تو می‌دانی و همه می‌دانند که شکنجه دیدن، به‌خاطر تو، زندانی کشیدن به‌خاطر تو و رنج بردن به پای تو، تنها لذت بزرگ زندگی من است. از شادی توست که من در دل می‌خندم، از امید‌رهایی توست که برق امید در چشمان خسته‌ام می‌درخشد و از خوشبختی توست که هوای پاک سعادت را در ریه‌هایم احساس می‌کنم. نمی‌توانم خوب حرف بزنم. نیروی شگفتی را که در زیر کلمات ساده و جمله‌های ضعیف و افتاده، پنهان کرده‌ام دریاب.

تو می‌دانی و همه می‌دانند که زندگی از تحمیل لبخندی بر لبان من، از آوردن برق امیدی در نگاه من، از برانگیختن موج شعفی در دل من، عاجز است.

تو، چگونه زیستن را به من بیاموز، چگونه مردن را خود خواهم آموخت. به من توفیق تلاش در شکست، صبر در نومیدی، رفتن بی‌همراه، جهاد بی‌سلاح، کار بی‌پاداش، فداکاری در سکوت، دین بی‌دنیا، مذهب بی‌عوام، عظمت بی‌نام، خدمت بی‌نان، ایمان بی‌ریا، خوبی بی‌نمود، گستاخی بی‌خامی، قناعت بی‌غرور، عشق بی‌هوس، تنهایی در انبوه جمعیت، و دوست داشتن بی‌آنکه دوست بداند، روزی کن.

اگر تنها ترین تنها شوم، باز خدا هست

او جانشین هم‌زمانش هست...

سپاس‌گزاری... پ

سپاس خداوندگار حکیم را که با لطف بی‌کران خود، آدمی را زیور عقل آراست. در آغاز وظیفه خود می‌دانم از زحمات بی‌دریغ استاد راهنمای خود، جناب آقای **دکتر جعفر فتحعلی**، صمیمانه تشکر و قدردانی کنم که از راهنمایی‌های ارزنده ایشان در راستای پیشبرد پژوهش، حاصل فراوان بردم و همواره شاگرد مکتب علم و انسانیت و منش والای ایشان هستم.

از همسر عزیزم، به پاس عاطفه سرشار و گرمای امیدبخش وجودش، که در این سردترین روزگاران، بهترین پشتیبان من بود نهایت سپاس و قدردانی را دارم. همچنین از اساتید محترمی که زحمت داوری این رساله را متقبل شده‌اند کمال تشکر و قدردانی را دارم.

مرتضی نظری

بهمن ۱۳۹۷

تعهد نامه

اینجانب مرتضی نظری دانشجوی دکتری رشته ریاضی کاربردی دانشکده علوم ریاضی دانشگاه صنعتی شاهرود، نویسنده رساله با عنوان بررسی مسائل مکان‌یابی چند وسیله‌ای معکوس پشتیبان، تحت راهنمایی دکتر جعفر فتحعلی متعهد می‌شوم:

- تحقیقات در این رساله توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های دیگر پژوهش‌گران، به مرجع مورد استفاده استناد شده است.
- مطالب این رساله، تا کنون توسط خود، یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ‌جا 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 به‌دست آوردن نتایج اصلی رساله تاثیرگذار بوده‌اند، در مقالات مستخرج از رساله رعایت می‌گردد.
- در تمام مراحل انجام این رساله، در مواردی که از موجود زنده (یا بافت‌های آنها) استفاده شده است، ضوابط و اصول اخلاقی رعایت شده است.
- در تمام مراحل انجام این رساله، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته (یا استفاده شده است)، اصل رازداری و اصول اخلاق انسانی رعایت شده است.

مرتضی نظری

بهمن ۱۳۹۷

مالکیت نتایج و حق نشر

- تمام حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده) متعلق به دانشگاه صنعتی شاهرود می‌باشد. این مطلب باید به نحو مقتضی، در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در این رساله بدون ذکر منبع مجاز نمی‌باشد.

چکیده

مکان‌یابی تسهیلات و تاسیسات از کاربردی‌ترین زمینه‌های تحقیقاتی است که در بسیاری از شاخه‌های علوم گسترده شده است و دارای کاربردهای بسیار زیادی است. امروزه مدل‌هایی از مکان‌یابی تاسیسات و تسهیلات که با شاخصه‌های دنیای واقعی تطابق بیشتری داشته باشند بیشتر مورد بررسی قرار گرفته است. به همین دلیل مکان‌یابی معکوس، مکان‌یابی پشتیبان، مکان‌یابی متعادل و آرمانی به دلیل تطابق بیشتری که در واقعیت دارند مورد توجه قرار گرفته‌اند. از این‌رو در این رساله ضمن معرفی مسائل مکان‌یابی کلاسیک p - میانه، مسائل مکان‌یابی اصلاحی مانند مسائل ۲- میانه پشتیبان در حالت نیمه ناخوشایند و ناخوشایند بر روی شبکه‌های درختی و مسئله چند وسیله‌ای پشتیبان آرمانی در فضای پیوسته را مورد بررسی قرار داده‌ایم و روش‌هایی کارآمد مبتنی بر اثبات قضایا را برای حل این مسائل ارائه کرده‌ایم. همچنین معکوس نوع حداقل هزینه و نوع محدودیت بودجه‌ای مسائل ۲- میانه پشتیبان را در فضای گسسته گراف‌ها و فضای پیوسته \mathbb{R}^2 مدل‌بندی ریاضی نموده و برای حل این مدل‌ها روش‌های تکراری شبه وایزفیلد و الگوریتم‌های فراابتکاری فرهنگی و بهینه‌سازی پرندگان را ارائه کرده‌ایم. در انتها نیز معکوس نوع محدودیت بودجه‌ای و حداقل هزینه یک مسئله ۲- سرویس‌دهنده متعادل را بر روی گراف‌ها مورد بررسی قرار داده و دو الگوریتم ابتکاری از مرتبه $O(n \log n)$ برای حل آن ارائه نموده‌ایم.

کلمات کلیدی: معکوس نوع محدودیت بودجه‌ای، معکوس نوع حداقل هزینه، مکان‌یابی پشتیبان، مکان‌یابی آرمانی، مکان‌یابی ۲- میانه متعادل.

لیست مقالات مستخرج از رساله

1. Nazari M., Fathali J., Reverse Backup 2-Median Problem with Variable Coordinate of Vertices, Journal of Operational Research in its Applications. (2018), 15 (2), 63-88.
2. Nazari M., Fathali J., Nazari M., Varedi Koulaei S. M., Inverse of Backup 2-Median Problems with Variable Edge Lengths and Vertex Weight on Trees and Variable Coordinates on the Plane. Production and Operation Management. (2018), 9 (2), 115-137.
3. Nazari M., Fathali J., Solution of Backup Multifacility Location Problem by Considering the Ideal Radius for each Customer, Journal of New Researches in Mathematics.(2018) submitted.
4. Nazari M., Fathali J., Mahdevar K., Semi-Obnoxious and Obnoxious Backup 2-Median Problems on Trees, OPSEARCH.(2018) submitted.
5. Nazari M., Fathali J., Inverse and reverse facility location problems with equality measures on a network, Discrete Optimization , (2018) submitted.
6. Omidi S., Fathali J., Nazari M., Inverse and reverse balanced facility location problems with variable edge lengths on trees, Theoretical Computer Science, (2018) submitted.
7. Nazari M., Fathali J., Goli M., Backup Reverse 2-Median Problem on Trees, the 10th International Conference on Iranian Operation Research Society, Babolsar, (2017).
۸. مرتضی نظری، جعفر فتحعلی، مسئله پشتیبان ۲- میانه نیمه ناخوشایند بر روی درخت، چهل و نهمین کنفرانس ریاضی ایران، تهران، (۱۳۹۷).
۹. مرتضی نظری، جعفر فتحعلی، مرتضی گلی، معکوس نوع محدودیت بودجه‌ای مسئله ۲- میانه متعادل بر روی گراف‌های عمومی: ارائه یک الگوریتم جدید با پیچیدگی زمانی $O(n^2)$ ، یازدهمین کنفرانس بین‌المللی انجمن ایرانی تحقیق در عملیات، کرمانشاه، (۱۳۹۷).

۱۰. مرتضی نظری، جعفر فتحعلی، ارائه یک الگوریتم جدید با پیچیدگی زمانی $O(n \log n)$ ، برای معکوس نوع حداقل هزینه مسئله مکان‌یابی ۲-میانه متعادل، یازدهمین کنفرانس بین‌المللی انجمن ایرانی تحقیق در عملیات، کرمانشاه، (۱۳۹۷).
۱۱. مرتضی نظری، جعفر فتحعلی، یک روش کارآمد برای حل مسائل مکان‌یابی پشتیبان چندوسیله‌ای آرمانی، چهارمین کنفرانس بین‌المللی مهندسی صنایع و سیستم‌ها، مشهد، (۱۳۹۷).
۱۲. مرتضی نظری، جعفر فتحعلی، ارائه یک مدل برنامه‌ریزی ریاضی برای مسئله مکان‌یابی ۲-میانه پشتیبان معکوس با روش تغییر طولی یال روی درخت، نهمین کنفرانس بین‌المللی انجمن ایرانی تحقیق در عملیات، شیراز، (۱۳۹۵).
۱۳. مرتضی نظری، جعفر فتحعلی، مسئله مکان‌یابی ۲-میانه پشتیبان معکوس با تغییر وزن رئوس روی درخت، اولین کنفرانس ملی بهینه‌سازی و تصمیم‌گیری، بابل‌سر، (۱۳۹۴).

فهرست مطالب

ق	فهرست تصاویر
ش	فهرست جداول
۱	۱ تعاریف و مفاهیم مقدماتی
۱	۱.۱ مفاهیم مقدماتی از آنالیز پیچیدگی
۵	۲.۱ تعاریف اولیه مربوط به گراف‌ها
۷	۳.۱ مقدماتی از روش‌های جستجو و مرتب‌سازی
۸	۱.۳.۱ روش جستجوی دودویی
۸	۲.۳.۱ مرتب‌سازی ادغامی
۱۰	۳.۳.۱ جستجوی عمقی و عرضی در گراف‌ها
۱۳	۲ مقدمه‌ای بر مسائل مکان‌یابی کلاسیک
۱۴	۱.۲ زمینه‌های عمده تحقیق در نظریه مکان‌یابی
۱۷	۲.۲ مولفه‌های اساسی مسائل مکان‌یابی
۱۷	۱.۲.۲ فضای مسائل مکان‌یابی
۱۹	۲.۲.۲ سرویس‌دهندگان
۲۰	۳.۲.۲ اندازه‌گیری فاصله
۲۱	۴.۲.۲ وزن
۲۲	۵.۲.۲ مشتریان
۲۲	۶.۲.۲ اهداف تصمیم‌گیرنده
۲۲	۷.۲.۲ تابع هدف‌های پرکاربرد مکان‌یابی
۲۳	۳.۲ برخی مسائل مهم مکان‌یابی
۲۳	۱.۳.۲ مسئله مکان‌یابی چند وسیله‌ای پیوسته
۲۵	۲.۳.۲ مسئله p -میانه و مدل ریاضی آن
۳۵	۳.۳.۲ مسئله مکان‌یابی p - مرکز

۳۹	۳	مقدمه‌ای بر مسائل مکان‌یابی اصلاحی
۳۹	۱.۳	مسائل مکان‌یابی پشتیبان
۴۰	۱.۱.۳	مسئله مکان‌یابی ۲- میانه پشتیبان
۴۱	۲.۱.۳	مسئله ۲- میانه پشتیبان روی درخت
۴۵	۲.۳	مسائل مکان‌یابی معکوس
۴۸	۱.۲.۳	مدل مکان‌یابی معکوس حداقل هزینه / حداکثر سود
۵۲	۲.۲.۳	مدل مکان‌یابی معکوس بودجه مقید
۵۴	۳.۲.۳	مدل‌های مکان‌یابی ترفیع و تنزیل
۵۵	۴.۲.۳	مروری بر برخی از مسائل مکان‌یابی معکوس
۶۵	۳.۳	مسئله مکان‌یابی آرمانی
۶۶	۱.۳.۳	مسئله مکان‌یابی وبر آرمانی
۶۸	۲.۳.۳	مسئله مکان‌یابی وبر با کمترین خطای مطلق
۶۹	۴	مسئله ۲- میانه پشتیبان نیمه‌ناخوشایند/ناخوشایند روی درخت
۶۹	۱.۴	مقدمه
۷۰	۲.۴	مسئله ۲- میانه پشتیبان نیمه‌ناخوشایند
۷۲	۱.۲.۴	چند خاصیت روی درخت‌ها
۷۴	۳.۴	مسئله ۲- میانه پشتیبان ناخوشایند
۷۹	۵	مسئله مکان‌یابی پشتیبان چندوسیله‌ای آرمانی
۷۹	۱.۵	مسئله مکان‌یابی پشتیبان چندوسیله‌ای آرمانی تحت نرم l_p
۸۴	۲.۵	الگوریتم شبه‌وایزفیلد
۸۶	۳.۵	نتایج محاسباتی
۹۱	۶	مسائل معکوس ۲- میانه پشتیبان
۹۲	۱.۶	معکوس نوع حداقل هزینه مسئله ۲- میانه پشتیبان با تغییر در وزن رؤس
۹۵	۲.۶	معکوس نوع حداقل هزینه مسئله ۲- میانه پشتیبان با تغییر در طول یال
۹۷	۳.۶	معکوس نوع محدودیت بودجه‌ای مسئله ۲- میانه پشتیبان با تغییر در وزن رؤس
۹۸	۴.۶	معکوس نوع محدودیت بودجه‌ای مسئله ۲- میانه پشتیبان با تغییر در طول یال
۹۹	۵.۶	معکوس مسئله ۲- میانه پشتیبان در فضای پیوسته با تغییر در مختصات نقاط

۱۰۰	۱.۵.۶ معکوس نوع محدودیت بودجه‌ای مسئله ۲- میانه پشتیبان در صفحه با نرم l_p
۱۰۵	۶.۶ الگوریتم‌های فراابتکاری
۱۰۶	۱.۶.۶ الگوریتم بهینه‌سازی پرندگان
۱۰۷	۲.۶.۶ الگوریتم فراابتکاری فرهنگی
۱۱۳	۷.۶ نتایج محاسباتی
۱۲۷	۷ معکوس مسئله مکان‌یابی متعادل
۱۲۸	۱.۷ تعریف مسئله
۱۲۹	۲.۷ معکوس نوع حداقل هزینه ۲- سرویس‌دهنده متعادل ۱.۲.۷ الگوریتم حل معکوس حداقل هزینه مسئله ۲- سرویس‌دهنده متعادل
۱۳۵	۳.۷ معکوس محدودیت بودجه‌ای ۲- سرویس‌دهنده متعادل
۱۴۱	مراجع

فهرست تصاویر

۳	مثال‌های ترسیمی برای نمادهای مجانبی Θ ، Ω و O .	۱.۱
۵	تصویری از مجموعه کلاس‌های تعریف شده.	۲.۱
۳۰	درختی با هفت راس و شش یال.	۱.۲
۳۰	درخت حاصل از حذف راس v_6 و یال (v_5, v_6) .	۲.۲
۳۱	درخت حاصل از حذف راس v_7 و یال (v_5, v_7) .	۳.۲
۳۱	درخت حاصل از حذف راس v_5 و یال (v_2, v_5) .	۴.۲
۳۲	درختی با شش راس و پنج یال.	۵.۲
۳۳	زیر درخت‌های حاصل از حذف یال (v_1, v_6) .	۶.۲
۳۳	زیر درخت‌های حاصل از حذف یال (v_1, v_2) .	۷.۲
۳۴	زیر درخت‌های حاصل از حذف یال (v_2, v_3) .	۸.۲
۳۴	زیر درخت‌های حاصل از حذف یال (v_2, v_5) .	۹.۲
۳۵	زیر درخت‌های حاصل از حذف یال (v_2, v_4) .	۱۰.۲
۳۷	درخت بدون وزن.	۱۱.۲
۶۰	درختی که رأس m ، 1 - میانه می‌باشد.	۱.۳
۶۱	زیر درخت‌های ریشه دار در رأس s .	۲.۳
۶۶	نمایش گرافیکی مسئله مکان‌یابی آرمانی.	۳.۳
۷۵	درخت T با ۴ راس.	۱.۴
	مقادیر تابع هدف برای سه جواب مسئله، نسبت به تغییرات $\rho \leq 0$.	۲.۴
۷۶	$\rho \leq 1$.	
۷۶	یک مسیر با ۵ راس.	۳.۴
	مقادیر تابع هدف به‌ازای جواب‌های $\{v_1, v_5\}$ و $\{v_2, v_5\}$ ، نسبت به	۴.۴
۷۷	تغییرات $0 \leq \rho \leq 1$.	
	نمودار نمودار سرعت همگرایی الگوریتم شبه-وایز فیلد ارائه شده	۱.۵
۸۹	در مقایسه با تعداد تکرار برای حالت $p = 2$.	

۱۰۸	تاثیر فرهنگ بر جامعه و بالعکس.	۱.۶
۱۱۲	نمایی از کارکرد الگوریتم فرهنگی.	۲.۶
	نقاط اولیه (مربع قرمز)، نقاط بهینه (ستاره مشکی) و نقاط ۲-	۳.۶
۱۱۳	میانه (نقاط بعلاوه آبی) برای $\rho = 0/3$ ، $n = 25$ و $B = 8^\circ$.	۴.۶
	نمودار مقایسه‌ای عملکرد روش‌های PSO، IPSO برای حالت $\rho =$	۵.۶
۱۱۴	$0/3$ ، $n = 25$ و $B = 8^\circ$.	۵.۶
	نمودار مقایسه‌ای عملکرد روش‌های PSO، IPSO برای حالت $\rho =$	۵.۶
۱۱۴	0 ، $n = 25$ و $B = 8^\circ$.	۵.۶
۱۲۱	مقادیر تابع هدف برحسب تعداد تکرار برای حالت $\beta = 18^\circ$.	۶.۶
	مختصات نقاط اولیه (مربع) و نقاط نهایی (دایره) برای حالت $p =$	۷.۶
۱۲۲	۲ و $\beta = 18^\circ$.	۸.۶
	مختصات نقاط اولیه (مربع) و نقاط نهایی (دایره) برای حالت $p =$	۸.۶
۱۲۲	۴ و $\beta = 18^\circ$.	۹.۶
	مختصات نقاط اولیه (مربع) و نقاط نهایی (دایره) برای حالت $p =$	۹.۶
۱۲۳	۵ و $\beta = 18^\circ$.	۱۰.۶
	مختصات نقاط اولیه (مربع) و نقاط نهایی (دایره) برای حالت $p =$	۱۰.۶
۱۲۳	۱۰ و $\beta = 18^\circ$.	۱۰.۶
	مختصات نقاط اولیه (مربع) و نقاط نهایی (دایره) برای حالت $p =$	۱۰.۶
۱۳۴	درخت T با ۹ راس	۱.۷
۱۳۴	یک مسیر با ۱۰ راس.	۲.۷
۱۳۸	گراف G با ۹ راس	۳.۷

فهرست جداول

۸۷	۱.۵	مختصات وسیله‌های موجود ($P_i = (a_i, b_i)$) و وزن بین سرویس‌دهنده‌های جدید و وسیله‌های موجود. ($w_{ji}, j = 1, \dots, 5$)
۸۸	۲.۵	وزن مربوط بین سرویس‌دهنده‌های جدید (v_{jl})
	۳.۵	نتایج بدست آمده برای $p = 2$ و $k = 4$ در حالت $i = 1, \dots, m$
۸۸		$R_i = 0/5$
	۴.۵	نتایج بدست آمده برای حالت $k = 4$ و $i = 1, \dots, m$ ، $R_i = 0/5$
۸۸		برای مقادیر مختلف p در 200 تکرار.
	۵.۵	مقایسه نتایج بدست آمده از الگوریتم حاضر با مقالات دیگر، نسبت به مقادیر مختلف p .
۸۹		
	۱.۶	تاثیرات ρ بر مقدار تابع هدف مسئله BR2MP برای حالت $B = 40$
۱۱۵		$n = 10$ و $p = 2$
	۲.۶	تاثیرات ρ بر مقدار تابع هدف مسئله BR2MP برای حالت $B = 35$
۱۱۵		$n = 50$ و $p = 2$
	۳.۶	تاثیرات ρ بر مقدار تابع هدف مسئله BR2MP برای حالت $\rho = 0/5$
۱۱۶		$n = 80$ و $B = 70$
	۴.۶	تاثیرات B ، بر مقدار تابع هدف مسئله BR2MP برای حالت $\rho = 0/3$
۱۱۷		$n = 30$ و $p = 3$
۱۱۸	۵.۶	مختصات نقاط (مشتری‌ها) به همراه وزن آن‌ها.
	۶.۶	نتایج شبیه‌سازی شده با بودجه $\beta = 180$ ، برای نرم‌های $p = 1, 2, 4, 5$
۱۱۹		
	۷.۶	مقادیر تابع هدف به‌ازای نرم‌های $p = 1, 2, 4, 5$ ، بر حسب تعداد تکرار
۱۲۱		
	۸.۶	مقایسه نتایج بدست آمده از روش‌های $IPSO$ ، PSO و CA ، با $n = 40$ ، $\beta = 75$ ، $\rho = 0/5$ و تعداد تکرار $t = 1000$ نسبت به نرم‌های $p = 1, 1/5, 2, 5, 10$
۱۲۴		

۹.۶	مقایسه نتایج بدست آمده از روش‌های <i>IPSO</i> ، <i>PSO</i> و <i>CA</i> با $n =$
	$\rho = 0.3$ ، $p = 2, 40$ و تعداد تکرار $t = 1000$ نسبت به $\beta =$
۱۲۴ $100, 300, 500, 700$
۱۰.۶	مقایسه نتایج بدست آمده از روش‌های <i>IPSO</i> ، <i>PSO</i> و <i>CA</i> با $n =$
	$\rho = 0.5$ ، $p = 2, 40$ و تعداد تکرار $t = 1000$ نسبت به $\beta =$
۱۲۴ $0, 0.3, 0.5, 0.8$
۱۱.۶	مقایسه نتایج بدست آمده از روش‌های <i>IPSO</i> ، <i>PSO</i> و <i>CA</i> با تعداد
۱۲۵	نقاط، بودجه و ضریب شکست‌های متفاوت.
۱۳۳ هزینه تغییر در وزن رئوس دخت T .
۱۳۳ هزینه تغییرات در وزن رئوس
۱۳۸ هزینه حاصل از تغییر وزن رئوس در گراف G .

فصل ۱

تعاریف و مفاهیم مقدماتی

تعاریف مقدماتی و پایه‌ای مورد نیاز فصل‌های بعدی این رساله در این فصل ارائه شده است. مطالب این فصل از مراجع [۳۸، ۶۴، ۹۲، ۱۰۲] برگرفته شده است.

۱.۱ مفاهیم مقدماتی از آنالیز پیچیدگی

در این قسمت تعاریف و مفاهیم مربوط به آنالیز پیچیدگی^۱ معرفی خواهد شد که در تفسیر زمان اجرای^۲ الگوریتم‌ها بکار می‌روند.

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

در بحث تجزیه و تحلیل الگوریتم‌ها به‌جای در نظر گرفتن رفتار دقیق توابع، رفتار مجانبی آن‌ها مورد بررسی قرار می‌گیرد. برای تخمین زمان اجرای الگوریتم و میزان

^۱Complexity

^۲Running time

پیچیدگی آن، از نمادهایی به نام نمادهای مجانبی استفاده می‌شود. این نمادها عبارتند از \mathcal{O} ، Θ و Ω که هر کدام طبق روش خاصی محاسبه می‌شوند و در تحلیل الگوریتم‌ها کاربرد فراوانی دارند.

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

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

به عبارتی دیگر می‌توان گفت، اگر $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \neq \infty$ ، آنگاه $f(n) = \mathcal{O}(g(n))$. اگر $f(n) = \mathcal{O}(g(n))$ ، آنگاه $\beta g(n)$ به عنوان کران بالای مجانبی برای تابع $f(n)$ تعریف می‌شود. شکل ۱.۱ الف، یک مثال برای نماد \mathcal{O} را نشان می‌دهد.

تعریف ۳.۱.۱. به ازای تابع مفروض $g(n)$ ، گویند $f(n) = \Omega(g(n))$ یا $f(n) \in \Omega(g(n))$ ، اگر یک مقدار حقیقی مثبت مانند β و یک عدد طبیعی مانند n_0 وجود داشته باشد به طوری که به ازای هر $n \geq n_0$ ، داشته باشیم:

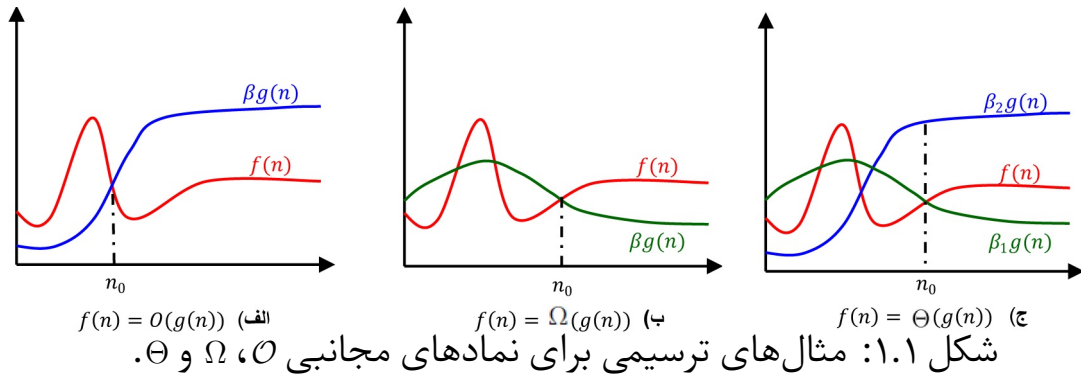
$$0 \leq \beta g(n) \leq f(n).$$

به عبارت دیگر می‌توان گفت که اگر $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \neq 0$ ، آنگاه $f(n) = \Omega(g(n))$. اگر $f(n) = \Omega(g(n))$ ، آنگاه $\beta g(n)$ به عنوان کران پایین مجانبی برای تابع $f(n)$ تعریف می‌شود. شکل ۱.۱ ب، یک مثال گرافیکی برای نماد Ω را نشان می‌دهد.

تعریف ۴.۱.۱. به ازای یک تابع $g(n)$ داده شده، گویند $f(n) = \Theta(g(n))$ یا $f(n) \in \Theta(g(n))$ ، اگر مقادیر حقیقی مثبت مانند β_1 ، β_2 و یک عدد طبیعی مانند n_0 وجود داشته باشد به طوری که به ازای هر $n \geq n_0$ ، داشته باشیم:

$$0 \leq \beta_1 g(n) \leq f(n) \leq \beta_2 g(n).$$

به عبارت دیگر، می‌توان گفت اگر $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \alpha$ ، آنگاه $f(n) = \Theta(g(n))$ که در آن α یک عدد ثابت مثبت است. اگر $f(n) = \Theta(g(n))$ ، آنگاه $\beta_1 g(n)$ را کران پایین مجانبی و $\beta_2 g(n)$ کران بالای مجانبی برای تابع $f(n)$ گویند. شکل ۱.۱ ج، یک مثال ترسیمی برای نماد Θ را نشان می‌دهد.



تعریف ۵.۱.۱. یک الگوریتم دارای زمان اجرای چندجمله‌ای است هرگاه $T(n) = O(n^k)$ ، که در آن k یک عدد طبیعی ثابت است.

تعریف ۶.۱.۱. یک الگوریتم دارای زمان اجرای شبه چندجمله‌ای است هرگاه پیچیدگی زمانی آن یک چندجمله‌ای از اندازه داده‌های ورودی و بزرگی آن‌ها باشد.

قضیه اصلی برای حل معادلات بازگشتی

قضیه اصلی، روشی برای حل معادلات بازگشتی به شکل

$$T(n) = aT(n/b) + f(n). \quad (1.1)$$

ارائه می‌کند، که در آن $a \geq 1$ و $b > 1$ اعداد ثابتی بوده و $f(n)$ یک تابع مثبت است. رابطه فوق بیانگر زمان اجرای یک الگوریتم با اندازه n می‌باشد که به a زیر مسئله با اندازه $\frac{n}{b}$ تقسیم شده است. زمان لازم برای تقسیم مسئله و ادغام نتایج زیر مسئله‌ها با تابع $f(n)$ بیان می‌شود.

قضیه ۱.۱.۱. (قضیه اصلی) معادله بازگشتی (۱.۱) را در نظر بگیرید.

• اگر برای یک عدد ثابت مانند $\varepsilon > 0$ داشته باشیم $f(n) = O(n^{\log_b a - \varepsilon})$ ، آنگاه خواهیم داشت $T(n) = \Theta(n^{\log_b a})$.

• اگر $f(n) = \Theta(n^{\log_b a})$ ، آنگاه $T(n) = \Theta(n^{\log_b a} \log n)$.

• اگر برای یک عدد ثابت مانند $\varepsilon > 0$ داشته باشیم $f(n) = \Omega(n^{\log_b a + \varepsilon})$ ، و نیز اگر برای مقدارهای به اندازه کافی بزرگ n ، یک عدد ثابت $c < 1$ وجود داشته باشد به طوری که $af(n/b) \leq cf(n)$ ، آنگاه $T(n) = \Theta(f(n))$.

یک مسئله را مسئله تصمیم‌گیری می‌گویند اگر همواره دارای جواب “بله” یا “خیر” باشد. یک مسئله بهینه‌سازی را می‌توان با اضافه کردن یک کران U (کران بالا یا پایین)، روی مقداری که باید بهینه شود و بررسی سؤالات زیر به یک مسئله تصمیم‌گیری تبدیل کرد:

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

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

قابل 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 چندجمله‌ای حل کند، آنگاه مسئله تصمیم‌گیری را نیز در زمان چندجمله‌ای حل خواهد کرد.

تعریف ۷.۱.۱. مسئله A به یک مسئله B در زمان چندجمله‌ای قابل تبدیل است، هرگاه یک تابع تبدیل R موجود باشد به طوری که:

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

۲. مقدار $R(x)$ در زمان چندجمله‌ای قابل محاسبه باشد.

به عبارتی دیگر، گویند مسئله A به مسئله B در زمان چندجمله‌ای قابل تبدیل است اگر یک الگوریتم چندجمله‌ای موجود باشد به طوری که هر نمونه از A را به یک نمونه از B با جواب‌های یکسان تبدیل کند. اگر یک مسئله A در زمان چندجمله‌ای به یک مسئله B قابل تبدیل باشد، آنگاه می‌نویسند:

$$A \leq_p B.$$

تعریف ۸.۱.۱. دسته \mathcal{P} ^۳، مجموعه‌ای از مسائل تصمیم‌گیری است که در زمان چندجمله‌ای قابل حل باشند.

^۳Deterministic Polynomial

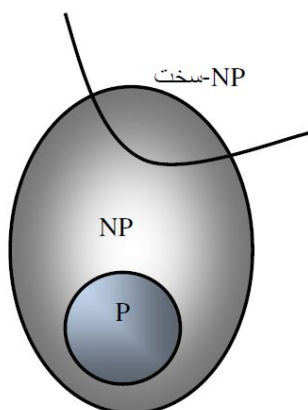
تعاریف اولیه مربوط به گراف‌ها ۵

تعریف ۹.۱.۱. کلاس NP^4 شامل آن دسته از مسائلی است که در زمان چندجمله‌ای قابل حل نمی‌باشند ولی می‌توان در یک زمان چندجمله‌ای ارزش “بله” یا “خیر” یک جواب کاندید را تشخیص داد.

تعریف ۱۰.۱.۱. گویند یک مسئله Q از نوع NP -سخت است اگر برای هر $Q' \in NP$ داشته باشیم:

$$Q' \leq_p Q.$$

شکل ۲.۱ تصویری از مجموعه کلاس‌های تعریف شده را نشان می‌دهد.



شکل ۲.۱: تصویری از مجموعه کلاس‌های تعریف شده.

۲.۱ تعاریف اولیه مربوط به گراف‌ها

تعریف ۱.۲.۱. (گراف). اگر V مجموعه‌ای متناهی و غیر تهی و E مجموعه همه زیر مجموعه‌های دو عضوی V باشد، در این صورت جفت $G = (V, E)$ ، یک گراف نامیده می‌شود. مجموعه‌های V و E ، به ترتیب مجموعه رئوس و مجموعه یال‌های گراف G می‌باشند. همچنین هر عضو V را یک راس یا گره و هر عضو E را یک یال می‌نامند. در نمایش تصویری یک گراف هر راس با یک نقطه و هر یال توسط یک پاره خط راست یا خمیده نشان داده می‌شود.

⁴Nondeterministic Polynomial

تعریف ۲.۲.۱. (رئوس مجاور). در گراف G اگر دو راس روی یک یال باشند آن‌ها را دو راس مجاور می‌نامند.

تعریف ۳.۲.۱. (طوقه). در گراف مفروض G یالی که راسی را به خودش وصل کند طوقه نامیده می‌شود.

تعریف ۴.۲.۱. (گراف ساده و گراف چندگانه). اگر در گرافی بیش از یک یال دو راس را به هم وصل کند گراف را چندگانه و آن یال‌ها را یال‌های چندگانه می‌نامند. گرافی که طوقه و یال چندگانه ندارد به گراف ساده معروف است.

تعریف ۵.۲.۱. (گراف وزن دار). اگر در گراف G به هر راس v یک عدد حقیقی $w(v)$ نسبت داده شود آن گراف را یک گراف وزن دار می‌نامند.

تعریف ۶.۲.۱. (زیر گراف). فرض کنید که $G = (V, E)$ یک گراف باشد. گراف $G' = (V', E')$ را زیر گراف G گویند هرگاه $V' \subset V$ و $E' \subset E$.

تعریف ۷.۲.۱. (درجه یک راس). درجه راس دلخواه v در گراف G عبارت است از تعداد یال‌هایی که از آن راس می‌گذرد. درجه راس v را با نماد $deg(v)$ نشان می‌دهند.

تعریف ۸.۲.۱. (گشت). یک گشت از G عبارت است از دنباله‌ای متناهی و ناتهی از رئوس و یال‌های G چون $W = v_0 e_1 v_1 e_2 v_2 \dots e_k v_k$ که به‌ازای $1 \leq i \leq k$ ، رئوس v_{i-1} و v_i دو سر یال e_i هستند. در این صورت گویند W یک گشت از v_0 تا v_k ، یا به عبارت دیگر یک (v_0, v_k) - گشت است.

تعریف ۹.۲.۱. (مسیر). یک مسیر از v_0 تا v_k در گراف G که با نماد $p(v_0, v_k)$ نشان داده می‌شود عبارت است از گشت $v_0 e_1 v_1 e_2 v_2 \dots e_k v_k$ به گونه‌ای که رئوس v_1, v_2, \dots, v_k و v_0 متمایز باشند. (که در نتیجه یال‌های e_1, e_2, \dots, e_k هم متمایز می‌شوند)

لازم به ذکر است که در این رساله با گراف‌های ساده سر و کار خواهیم داشت. از طرفی چون در گراف‌های ساده یال چندگانه نداریم می‌توان مسیر $p(v_0, v_k)$ را به صورت دنباله‌ای از رئوس $v_0 v_1 v_2 \dots v_k$ یا به صورت دنباله‌ای از یال‌های $e_1 e_2 \dots e_k$ نشان داد.

تعریف ۱۰.۲.۱. (طول مسیر). طول مسیر $p(v_0, v_k) = e_1 e_2 \dots e_k$ که با نماد $l(p(v_0, v_k))$ نشان داده می‌شود برابر است با مجموع طول تمام یال‌های تشکیل دهنده آن مسیر

یعنی:

$$l(p(v_0, v_k)) = \sum_{i=1}^k l(e_i).$$

تعریف ۱۱.۲.۱. (فاصله بین دو راس در گراف). اگر $u, v \in V$ دو راس دلخواه از گراف مفروض G باشند و $P[u, v]$ عبارت باشد از مجموعه همه مسیرهای بین دو راس u و v . در این صورت فاصله بین دو راس u و v که با نماد $d(u, v)$ نشان داده می‌شود برابر است با طول کوتاه‌ترین مسیر گذرنده از این رئوس، یعنی:

$$d(u, v) = \min_{p_i(u, v) \in P[u, v]} (l(p_i(u, v))).$$

تعریف ۱۲.۲.۱. (گراف همبند). یک گراف بدون جهت را همبند گویند، هرگاه بین هر دو راس دلخواه آن مسیری موجود باشد. در غیر این صورت آن را ناهمبند می‌نامند.

تعریف ۱۳.۲.۱. (دور). یک دور در گراف عبارت است از مسیری که نقطه ابتدایی و انتهایی آن یکی باشد.

تعریف ۱۴.۲.۱. (درخت). هر گراف همبند و بدون دور $G = (V, E)$ ، درخت نامیده می‌شود.

تعریف ۱۵.۲.۱. انقباض یال $e = (i, j)$ از گراف G . $G \setminus \{e\}$ ، گرافی است که در آن نه تنها یال e حذف می‌شود بلکه دو انتهای آن یعنی i و j بر هم منطبق می‌شوند. به عبارتی دیگر i و j را یک رأس در نظر می‌گیرند و هر یالی که به دو رأس i و j متصل می‌شود را به این رأس متصل می‌کنند.

تعریف ۱۶.۲.۱. (گراف انقباضی). گراف $\hat{G} = (\hat{V}, \hat{E})$ را انقباضی از گراف $G = (V, E)$ می‌نامند هرگاه با انقباض متوالی یال‌هایی از گراف G به وجود آمده باشد.

۳.۱ مقدماتی از روش‌های جستجو و مرتب‌سازی

در این قسمت مفاهیم مربوط به روش‌های جستجو و مرتب‌سازی معرفی خواهند شد.

۱.۳.۱ روش جستجوی دودویی

جستجوی دودویی یک روش برای یافتن یک مقدار عددی در یک آرایه مرتب داده شده است. فرض کنید آرایه داده شده A به صورت صعودی مرتب شده و x مورد جستجو باشد. در این روش، x با عضو وسط آرایه A مقایسه می‌شود. اگر x با این عضو برابر بود، آنگاه جستجو تمام می‌شود. اگر x از عضو وسط بزرگتر باشد، آنگاه جستجو در بخش بالایی آرایه و در غیر این صورت جستجو در بخش پایینی آرایه انجام می‌شود. این رویه به صورت بازگشتی تا یافتن عضو مورد نظر ادامه می‌یابد. الگوریتم جستجوی دودویی به شرح زیر است:

الگوریتم ۱.۳.۱. (مقدار عددی x را در آرایه مرتب A پیدا می‌کند)

شروع

گام ۱. قرار دهید $i = 1$ ، $j = n$ و $k = 0$.

گام ۲. مقدار $m = \lfloor \frac{i+j}{2} \rfloor$ را بیابید.

گام ۳. اگر $x = A[m]$ ، آنگاه قرار دهید $k = m$ و متوقف شوید.

گام ۴. اگر $x < A[m]$ ، آنگاه قرار دهید $j = m - 1$ ، در غیر این صورت قرار دهید $i = m + 1$.

گام ۵. اگر $i \leq j$ ، آنگاه به گام ۲ بروید، در غیر این صورت عضو x در آرایه A وجود ندارد و متوقف شوید.

پایان.

زمان اجرای الگوریتم جستجوی دودویی برابر $O(\log n)$ می‌باشد [۹].

۲.۳.۱ مرتب‌سازی ادغامی

مرتب‌سازی ادغامی^۵ یکی از سریع‌ترین روش‌های مرتب‌سازی موجود می‌باشد. این مرتب‌سازی بر اساس روش تقسیم و حل عمل کرده و الگوریتم آن به صورت زیر خلاصه می‌شود:

^۵Merge Sort

ابتدا آرایه مورد نظر به دو زیر آرایه با اندازه تقریباً یکسان تقسیم می‌شود. سپس، هر زیر آرایه مرتب می‌شود (این عمل به صورت بازگشتی با استفاده از خود مرتب‌سازی ادغامی انجام می‌شود). در نهایت، دو زیر آرایه مرتب با یکدیگر ادغام می‌شود. الگوریتم زیر این رویه را با جزئیات بیشتر بیان می‌کند. ورودی این الگوریتم یک آرایه شامل n عدد حقیقی a_1, \dots, a_n بوده و خروجی آن یک جایگشت $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ می‌باشد، به طوری که برای هر $i = 1, \dots, n-1$ داریم $a_{\pi(i)} \leq a_{\pi(i+1)}$.

الگوریتم ۲.۳.۱. (MERGE-SORT(a_1, \dots, a_n))

شروع

گام ۱. اگر $n = 1$ ، آنگاه قرار دهید $\pi(1) = 1$ و متوقف شوید.

گام ۲. قرار دهید

$$m = \lfloor \frac{n}{2} \rfloor,$$

$$\rho = \text{MERGE-SORT}(a_1, \dots, a_m),$$

$$\sigma = \text{MERGE-SORT}(a_{m+1}, \dots, a_n).$$

گام ۳. عملیات زیر را انجام دهید:

۱.۳ قرار دهید $k = 1$ و $l = 1$.

تا زمانی که $k \leq m$ و $l \leq n - m$ ، انجام دهید:

اگر $a_{\rho(k)} \leq a_{m+\sigma(l)}$ ، آنگاه قرار دهید $\pi(k+l-1) = \rho(k)$ و $k = k+1$.

در غیر این صورت، قرار دهید $\pi(k+l-1) = m + \sigma(l)$ و $l = l+1$.

۲.۳ تا زمانی که $k \leq m$ ، قرار دهید $\pi(k+l-1) = \rho(k)$ و $k = k+1$.

۳.۳ تا زمانی که $l \leq n - m$ ، قرار دهید $\pi(k+l-1) = m + \sigma(l)$ و $l = l+1$.

پایان.

بر اساس تعریف بازگشتی این الگوریتم، مرتب کردن n عضو نیاز به مرتب شدن دو زیر آرایه تقریباً $\frac{n}{2}$ عضوی دارد. پس از این مرحله، این دو زیر آرایه ادغام می‌شوند.

جهت ادغام دو زیر آرایه، که در مجموع با n عضو، حداکثر n مقایسه اتفاق خواهد افتاد. پس می‌توان نوشت:

$$T(n) = 2T\left(\frac{n}{2}\right) + n.$$

حال با استفاده از "قضیه اصلی" زمان اجرای $O(n \log n)$ برای مرتب‌سازی نتیجه می‌شود.

۳.۳.۱ جستجوی عمقی و عرضی در گراف‌ها

در نظریه گراف‌ها، جستجوی عرضی^۶، به اختصار BFS، و جستجوی عمقی^۷، به اختصار DFS، از الگوریتم‌های مهم پیمایش گراف‌ها می‌باشند و پایه و اساس بسیاری از الگوریتم‌های ترکیبیاتی هستند. گراف $G = (V, E)$ و رأس مفروض s را در نظر بگیرید. خروجی این جستجوها یک درخت با ریشه در رأس s است که شامل کلیه راس‌های قابل دسترس از s می‌باشد. رئوس گراف تحت پیمایش در طی جستجو جهت مشخص شدن وضعیت‌شان رنگ می‌شوند. هر رأس در ابتدا سفید بوده، وقتی برای اولین بار کشف می‌شود طوسی شده و وقتی کلیه راس‌های مجاور آن بررسی شدند، سیاه می‌شود.

جستجوی عرضی

روش جستجوی عرضی برای پیمایش گراف، جستجوی سطح به سطح گراف است. الگوریتم این روش جستجو از یک رأس به‌عنوان ریشه شروع می‌شود و آن را در سطح یک قرار می‌دهد. سپس در هر مرحله همه همسایه‌های راس‌های آخرین سطح دیده شده را که تا به حال دیده نشده‌اند بازدید می‌کند و آن‌ها را در سطح بعدی می‌گذارد. این روند زمانی متوقف می‌شود که همه همسایه‌های راس‌های آخرین سطح، قبلاً دیده شده باشند. الگوریتم، هر بار همه همسایه‌های یک رأس را بازدید کرده و سپس به سراغ رأس بعدی می‌رود و بنابراین گراف، سطح به سطح پیمایش خواهد شد. این روند آن قدر ادامه می‌یابد که همه گراف پیمایش شود.

الگوریتم ۳.۳.۱. $(BFS(G, s))$

شروع

گام ۱. به‌ازای هر رأس $u \in V$ قرار دهید $color(u) = white$.

^۶Breadth-First Search

^۷Depth-First Search

گام ۲. قرار دهید $Q = \emptyset$.

گام ۳. s را وارد صف Q کنید.

گام ۴. تا زمانی که $Q \neq \emptyset$ ، انجام دهید:

۱.۴ اولین عضو صف Q را خارج کنید و آن را u بنامید.

۲.۴ به‌ازای هر رأس مجاور u مانند v اگر $\text{color}(v) = \text{white}$ ، آنگاه قرار دهید

$$\text{color}(v) = \text{gray}$$

رأس v را به صف Q وارد کنید.

۳.۴ قرار دهید $\text{color}(u) = \text{black}$.

پایان.

جستجوی عمقی

روش جستجوی عمق اول برای پیمایش گراف، همان‌طور که از نامش پیداست جستجوی عمیق‌تر در گراف تا حد ممکن است. الگوریتم از ریشه شروع می‌شود و در هر مرحله همسایه‌های رأس جاری را از طریق یال‌های خروجی رأس جاری به ترتیب بررسی و به محض روبه‌رو شدن با همسایه دیده نشده تاکنون، به صورت بازگشتی برای آن رأس به عنوان رأس جاری اجرا می‌شود. در صورتی که همه همسایه‌ها قبلاً دیده شده باشند، الگوریتم عقب‌گرد می‌کند و اجرای الگوریتم برای رأسی ادامه می‌یابد که از آن به رأس جاری رسیده‌ایم. به عبارتی الگوریتم در حد امکان، به عمق بیشتر و بیشتر می‌رود و در مواجهه با بن‌بست به عقب بر می‌گردد. این روند آن‌قدر ادامه می‌یابد تا اینکه همه‌ی رأس‌های قابل دستیابی از ریشه دیده شوند. ابتدا رویه زیر را به صورت یک زیر الگوریتم بیان و سپس به بیان الگوریتم اصلی می‌پردازیم:

رویه $\text{DFS-VISIT}(G, v)$

۱. قرار دهید $\text{color}(u) = \text{gray}$.

۲. به‌ازای هر رأس مجاور u مانند v

اگر $\text{color}(v) = \text{white}$ ، آنگاه قرار دهید $\pi(v) = u$ ، $\text{DFS-VISIT}(G, v)$ را فراخوانی کنید و رأس v را به صف Q وارد کنید.

۳. قرار دهید $\text{color}(u) = \text{black}$.

الگوریتم ۴.۳.۱. $(DFS(G))$

شروع

گام ۱. به ازای هر رأس $u \in V$ قرار دهید $color(u) = white$.

گام ۲. به ازای هر رأس $u \in V$ ، اگر $color(u) = white$ ، آنگاه رویه $DFS-VISIT(G, u)$ را فراخوانی کنید.

پایان.

در دو روش جستجوی ارائه شده، هر یال گراف زمینۀ دقیقاً یک بار پیمایش می‌شود. همچنین هر رأس قابل دسترسی از ریشه دقیقاً یک بار بازدید خواهد شد. پس پیچیدگی زمانی هر دو جستجوی عمقی و عرضی برابر با $\mathcal{O}(|V(G)| + |E(G)|)$ می‌باشد.

فصل ۲

مقدمه‌ای بر مسائل مکان‌یابی کلاسیک

مسائل مکان‌یابی مسائلی هستند که وقتی به دنبال یافتن محل و نحوه استقرار بهینه یک یا چند سرویس‌دهنده بر اساس عوامل و متغیرهای مؤثر بر مکان‌یابی هستیم، با آن‌ها مواجه می‌شویم. معمولاً در این مسائل فرض بر این است که تعدادی مشتری (سرویس‌گیرنده یا نقطه تقاضا) موجودند و می‌خواهیم بهترین مکان برای استقرار سرویس‌دهندگان جدید را به‌گونه‌ای بیابیم که هزینه حمل و نقل، تأثیرات محیطی نامطلوب، زمان سرویس‌دهی، سود، کیفیت، نحوه توزیع سرویس و رعایت انصاف و برابری در آن، کسب بیشترین سهم در بازار و غیره، بسته به شرایط مسئله بهینه شوند.

مکان‌یابی تأسیسات و تسهیلات، از کاربردی‌ترین زمینه‌های تحقیقاتی می‌باشد که در بسیاری از شاخه‌های علوم گسترده شده است. از جمله این شاخه‌ها می‌توان به تحقیق در عملیات، علوم مدیریت و تخصیص منابع، مهندسی صنایع، جغرافیا، اقتصاد، علوم کامپیوتر، ریاضیات، بازاریابی، مهندسی برق، برنامه‌ریزی شهری، علوم نظامی و دیگر شاخه‌های مرتبط اشاره کرد.

از کاربردهای معروف مسائل مکان‌یابی می‌توان مکان‌یابی انبارها، کارخانه‌ها، ماشین‌آلات کارخانه، بیمارستان‌ها، نمایندگی‌های فروش، آتش‌نشانی‌ها، پاسگاه‌های پلیس، پایانه‌های مسافربری، اجزاء الکترونیکی در مدارها، آژیر خطر، چاه‌های نفت، تأسیسات نظامی، نیروگاه‌ها، آزمایشگاه‌ها و غیره را نام برد.

۱.۲ زمینه‌های عمده تحقیق در نظریه مکان‌یابی

شاید بتوان پیدایش مسئله مکان‌یابی را به زمانی نسبت داد که در قرن هفدهم میلادی فرما^۱ مسئله‌ای را به این صورت مطرح نمود: ”فرض کنید سه نقطه در صفحه داده شده است، نقطه چهارم را به گونه‌ای بیابید که مجموع فاصله‌های آن تا سه نقطه داده شده کمینه گردد.“ توریچلی^۲ در سال ۱۶۴۰ مشاهده کرد که اگر سه نقطه موجود را سه رأس 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. بدین دلیل، نقطه بهینه را نقطه توریچلی می‌نامند. در سال ۱۶۴۷ کاوالیری^۳ نشان داد که زاویه بین نقطه توریچلی و سه نقطه مورد نظر، ۱۲۰ درجه است. سیمپسون^۴ در سال ۱۷۵۰ ثابت کرد که اگر یک مثلث متساوی الاضلاع بر سه نقطه داده شده گذرانده شود و رئوس این مثلث به یکی از این سه نقطه که بر روی ضلع مقابل قرار گرفته وصل شود، محل تقاطع این سه خط، نقطه توریچلی را مشخص می‌کند. از این رو این سه خط را خطوط سیمپسون می‌نامند.

حالت مستثنایی که سه نقطه داده شده مثلثی را تشکیل دهند که در یک رأس زاویه بیشتر یا مساوی ۱۲۰ درجه داشته باشد، تا سال ۱۸۳۴ بدون حل باقی ماند، تا اینکه هینن^۵ ثابت کرد همان رأسی که دارای زاویه بیشتر یا مساوی ۱۲۰ درجه است، جواب بهینه مسئله می‌باشد. او همچنین ثابت کرد طول خطوط سیمپسون با هم مساوی هستند و طول آن‌ها با فاصله نقطه بهینه از سه نقطه داده شده نیز برابر است. فسبندر^۶ مفهوم دوگانی را برای مسئله فرما مطرح کرد. بعدها در سال ۱۹۶۷ کوهن^۷ به‌طور مبسوطی مفهوم دوگانی را مورد بررسی قرار داد و نتایج جالبی از تحقیقات وی منتشر گردید [۹۵].

نظریه مکان‌یابی به شکلی که امروزه مورد استفاده قرار می‌گیرد، با منتشر شدن کتاب آلفرد وبر در سال ۱۹۰۹ تحت عنوان *Über den Standort der Industrien* متولد شد [۱، ۱۴۱]. وبر در این کتاب نتایج تحقیقات خود را در مورد صنایع کارخانه‌ای ارائه کرد. در واقع می‌توان گفت که اولین تعریف مسئله مکان‌یابی به‌صورت کاربردی توسط وبر^۸ ارائه شده است [۱۴۱].

نظریه وبر دارای سه اصل اساسی است. اول اینکه محل مواد اولیه معلوم است.

¹Fermat

²Torricelli

³Cavalieri

⁴Simpson

⁵Heinen

⁶Fesbender

⁷Kuhn

⁸Weber

دوم اینکه مکان و میزان مصرف مشتریان مشخص است. تعداد این مکان‌ها متناهی است و جدا از هم هستند. هر تولید کننده بازار نامحدودی در اختیار دارد و امتیازات انحصاری از نظر انتخاب مکان وجود ندارد. سوم اینکه چندین مکان ثابت برای عرضه نیروی کار وجود دارد به طوری که این نیروی کار دارای قابلیت تحرک و جابجایی نبوده و عرضه آن نامحدود است.

اما مطالعات جدی بر روی مکان‌یابی از زمانی شروع شد که در سال ۱۹۶۴ حکیمی^۹، تابع هدف را به دو صورت کمترین مجموع^{۱۰} و مینیماکس^{۱۱} مطرح کرد [۷۵]. او همچنین مطالعاتی بر روی مسائل مکان‌یابی روی شبکه‌ها انجام داد. در واقع حکیمی کسی بود که روابط p -میان را ابداع و مسئله p -میان را معرفی نمود. از آن زمان به بعد مطالعات زیادی بر روی مسائل مکان‌یابی انجام گرفته است.

معمولاً در مسائل مکان‌یابی سرویس دهندگانی که می‌خواهیم مکان استقرار آن‌ها را بیابیم، مطلوب در نظر گرفته می‌شوند و تنها تأثیرات مثبت آن‌ها را به حساب می‌آوریم. بنابراین نزدیک بودن به آن‌ها نوعی جاذبه و دور بودن از آن‌ها نوعی دافعه تلقی می‌شود. در جوامع مدرن و پیشرفته امروزی با فعالیت‌ها و سرویس دهندگانی روبه‌رو هستیم که علاوه بر ارائه خدمات، تأثیرات منفی و مخربی نیز بر محیط پیرامون خود دارند. کارگاه‌های تولیدی، مثال آشکار این مورد هستند. کارگاه کالای مورد نیاز جامعه را تولید کرده و از طرف دیگر آلاینده محیط زیست است. مراکز ورزشی در عین حال که باید در دسترس باشند و امکانات ورزشی و تفریحی مناسب را ارائه دهند، می‌توانند تأثیرات منفی همچون سر و صدا، ازدحام جمعیت و یا مشکلاتی از این قبیل را نیز داشته باشند.

مطالعه مدل‌هایی که هر دو ویژگی را به‌طور همزمان در خود دارند، انطباق بیشتری با مسائل واقعی و کاربردی دارد و به سرعت در حال پیشرفت است. گلدمن و دیرینگ^{۱۲} [۷۱] و چرچ و گارفینکل^{۱۳} [۳۶] اولین کسانی بودند که مواردی را پیشنهاد کردند که بعدها مدل‌های مکان‌یابی مضر^{۱۴} نام گرفتند. در این دسته مدل‌ها، مشتریان فرض مطلوب بودن سرویس دهنده‌ها را کنار می‌گذارند و دیگر سعی در نزدیک‌تر بودن به آن‌ها را نداشته و تا حدی از آن‌ها دوری می‌کنند. به عنوان کاربردی از این مدل‌ها می‌توان به مکان‌یابی راکتورهای هسته‌ای، مراکز تصفیه آب و اماکن دفن زباله اشاره کرد. در حقیقت ویژگی‌های این نوع سرویس دهندگان نوعی دافعه ایجاد می‌کند و سبب می‌شود که مشتریان آن‌ها را از خود دور کنند.

⁹Hakimi

¹⁰Minisum

¹¹Minimax

¹²Golman and Dearing

¹³Church and Garfinkel

¹⁴Obnoxious

می‌توان گفت اولین طبقه‌بندی مدل‌های مختلف مکان‌یابی توسط هندلر^{۱۵} و میرچندانی^{۱۶} ارائه شد [۷۷]. پس از آن طبقه‌بندی‌های دیگری از جمله توسط هاماجر^{۱۷} و نیکل^{۱۸} انجام شد [۷۶]. همچنین بررسی‌هایی از کارهای انجام شده در این زمینه ارائه شد که از آن جمله تانسل^{۱۹} و همکاران [۱۳۲]، کرارپ^{۲۰} و پروزن^{۲۱} [۹۳]، هانسن^{۲۲} و همکاران [۷۸]، اون^{۲۳} و دسکین^{۲۴} [۱۱۵]، کاپانرا^{۲۵} [۳۵]، اسکاپارا^{۲۶} و اسکاتلا^{۲۷} [۱۲۴] لیستی از کارهای انجام شده در زمینه‌های مختلف مکان‌یابی را ارائه کرده‌اند. کارنت^{۲۸} و همکاران نیز کاربردهای مسائل مکان‌یابی را بیان نموده‌اند [۴۱].

به‌طور ویژه در دو دهه اخیر نیز مطالعات گوناگونی بر روی مسائل مکان‌یابی صورت گرفته است که به‌عنوان نمونه کامپیل^{۲۹} مسئله مکان‌یابی گسسته هاب را در سال ۱۹۹۴ مورد مطالعه قرار داد [۳۴]. در سال ۲۰۰۱ کلامروس^{۳۰} به مسئله تک وسیله‌ای با موانع چند وجهی پرداخت و یک الگوریتم قطعی و یک راه حل ابتکاری برای آن ارائه کرد [۹۰]. براتکورن^{۳۱} و همکاران در سال ۲۰۰۲ الگوریتم هوشمند سریعی برای رده وسیعی از مسائل مکان‌یابی پوشش ارائه کردند [۲۱]. همچنین یک رویکرد شاخه و قیمت در سال ۲۰۰۵ توسط سنی^{۳۲} و همکاران ارائه شد [۱۲۵]. زانگ^{۳۳} در سال ۲۰۰۶ الگوریتمی شبه حریصانه برای مسئله مکان‌یابی دو سطحی پیشنهاد کرد [۱۴۶]. در سال ۲۰۰۷ ساهین^{۳۴} و سورا^{۳۵} مدل‌های گوناگون مسئله مکان‌یابی سلسله‌مراتبی را مورد بررسی قرار دادند [۱۲۲]. بالسیک^{۳۶} و بیامون^{۳۷} در

¹⁵Handler¹⁶Mirchandani¹⁷Hamacher¹⁸Nickel¹⁹Tansel²⁰Krarup²¹Pruzan²²Hansen²³Owen²⁴Deskin²⁵Cappanera²⁶Scaparra²⁷Scutella²⁸Current²⁹Campbell³⁰Klamroth³¹Brotcorne³²Senne³³Zhang³⁴Sahin³⁵Sural³⁶Balcik³⁷Beamon

سال ۲۰۰۸ به مسئله مکان‌یابی برای مسئله کمک‌های بشر دوستانه پرداختند [۱۰]. در سال ۲۰۰۹ جمالیان و فتحعلی به مسئله مکان‌یابی با کمترین مقدار خطای مطلق پرداختند [۸۵]. در سال ۲۰۱۱ کالکسیکس^{۳۸} مسئله میانه چند وسیله‌ای با وزن‌های مثبت و منفی روی گراف را مورد مطالعه قرار داد [۸۷]. فرهانی و همکاران مسئله مکان‌یابی پوشش را در سال ۲۰۱۲ مورد مطالعه قرار دادند. در سال ۲۰۱۳ لی^{۳۹} به مسئله مکان‌یابی بدون ظرفیت پرداخت و الگوریتمی تقریبی برای حل آن ارائه نمود [۹۷]. در همین سال برسنو^{۴۰} الگوریتم شاخه و کران را برای مسئله مکان‌یابی رقابتی ارائه نمود [۱۶]. در سال ۲۰۱۴ رحمانی و میرحسینی الگوریتم ترکیبی ژنتیک کرم شب تاب را برای مسئله مکان‌یابی با ظرفیت معرفی کردند [۱۱۹]. هو^{۴۱} در سال ۲۰۱۵ به مسئله با ظرفیت تک منبع پرداخته و یک الگوریتم ابتکاری تکراری تابو را برای آن ارائه کرد [۸۲]. در سال ۲۰۱۶ مانتویا^{۴۲} و همکاران مسئله مکان‌یابی با ظرفیت چند محصوله را مورد مطالعه قرار دادند [۱۰۶]. در حال حاضر به دلیل افزایش روز افزون کاربردهای مسائل مکان‌یابی در زندگی روزمره، بسیاری از محققین و پژوهشگران به مطالعه در این شاخه از تحقیق در عملیات پرداخته‌اند [۱۱۸، ۴۵، ۱۳۵].

مسائل مکان‌یابی در توابع هدف، فواصل، تعداد و سبب سرویس‌دهنده‌هایی که باید تاسیس شوند و در چندین عامل دیگر متفاوت هستند. از این رو بسته به نوع هدف و با استفاده از هر یک از این شاخص‌های مختلف به مدل‌های مختلف مسئله مکان‌یابی دست خواهیم یافت.

۲.۲ مولفه‌های اساسی مسائل مکان‌یابی

در حالت کلی یک مسئله مکان‌یابی دارای عناصر و مولفه‌های اساسی‌ای می‌باشد که در ادامه به آن‌ها اشاره خواهد شد.

۱.۲.۲ فضای مسائل مکان‌یابی

معمولاً فضای مسائل مکان‌یابی یکی از سه دسته زیر است:

• فضای پیوسته

³⁸Kalcsics

³⁹Li

⁴⁰Beresnev

⁴¹Ho

⁴²Montoya

هرگاه فضای مکان تأسیسات و نقاط تقاضا پیوسته باشند، یعنی توسط متغیرهایی که به صورت پیوسته تغییر می‌کنند، نظیر مختصات مشخص شود، مدل را پیوسته می‌گویند. در این مدل‌ها مکان سرویس‌دهنده را باید در فضای d -بعدی \mathbb{R}^d پیدا نمود. اکثر مسائل مکان‌یابی پیوسته 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 مکان‌یابی پیوسته از ابزارهای هندسه و آنالیز (آنالیز تابعی که معمولاً از ویژگی‌های تحدب و تقعر استفاده می‌شود) استفاده می‌شود.

• فضای گسسته

مدل‌های مکان‌یابی گسسته مدل‌هایی هستند که در آن‌ها باید مکان سرویس‌دهندگان تنها روی نقاط از پیش تعیین شده پیدا شوند. 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 خوانده‌اند.

• فضای شبکه‌ای

به‌طور کلی مسائل مکان‌یابی بر روی فضاها شبکه‌ای رده‌ای دیگر از مسائل مکان‌یابی گسسته است که به جستجو برای یافتن بهترین مکان برای تاسیس سرویس‌دهنده‌ها روی شبکه می‌پردازد.

یعنی در این رده از مسائل، هدف پیدا کردن مجموعه‌ای از نقاط روی شبکه مفروض است به طوری که هزینه کل سرویس‌دهی این نقاط (سرویس‌دهنده‌ها) تا نقاط تقاضا (مشتری‌ها) کمینه شود.

در حوزه مکان‌یابی با فضای شبکه‌ای، مسائلی وجود دارند که در حالت کلی مسائلی NP -سخت هستند. از جمله این مسائل می‌توان مسئله مکان‌یابی p - میانه، مسئله مکان‌یابی p - مرکز، مسئله مکان‌یابی بدون ظرفیت و مسئله مکان‌یابی پوشش را نام برد. در سال ۱۹۶۸ تیتز^{۴۳} و بارت^{۴۴} به حل مسئله p - میانه روی گراف پرداختند و روشی جدید برای یافتن p - میانه راسی ارائه کردند [۱۳۴]. در سال ۱۹۹۱ کو^{۴۵} و لی^{۴۶} الگوریتمی با پیچیدگی زمانی $O(n)$ برای مسئله وزن‌دار ۲ - مرکز و ۳ - مرکز با فاصله مستطیلی ارائه کردند [۹۱]. کاروین^{۴۷} و حکیمی در سال ۱۹۷۹ نشان دادند که مسئله مکان‌یابی p - میانه روی شبکه مسئله‌ای NP -سخت است و الگوریتمی ارائه

⁴³Titz

⁴⁴Bart

⁴⁵Ko

⁴⁶Lee

⁴⁷Karvin

کردند که p – میانه درخت را می‌یابد [۸۸]. همچنین آن‌ها در سال ۱۹۷۹ الگوریتمی برای یافتن p – مرکز روی شبکه ارائه نمودند [۸۹]. در سال ۱۹۹۴ کامپیل در مقاله‌ای تحت عنوان ”فرمول‌بندی عدد صحیح مسئله مکان‌یابی گسسته هاب“، فرمول‌بندی عدد صحیح را برای چهار نوع از مسائل هاب شامل مسئله p – میانه هاب، مسئله p – مرکز هاب، مسئله بدون ظرفیت هاب و مسئله پوشش هاب ارائه نمود [۳۴]. در سال ۱۹۹۷ حدادی روشی ابتکاری ساده برای حل مسئله پوشش ارائه کرد [۷۴].

در همین سال روش‌های جستجوی همسایگی متعددی برای مسئله p – میانه توسط هانسن^{۴۸} و ملادونویچ^{۴۹} ارائه شده است [۷۹]. ارنست^{۵۰} و کریشنامورتی^{۵۱} در سال ۱۹۹۸ با انتشار مقاله‌ای به مسئله تخصیص چندوسیله‌ای p – میانه هاب بدون ظرفیت پرداختند و الگوریتمی قطعی و همچنین الگوریتمی ابتکاری برای حل آن ارائه کردند [۵۰]. ایبری^{۵۲} در سال ۲۰۰۱ روش کارامدی برای حل مسئله تخصیص تک وسیله‌ای p – هاب با سه یا چهار هاب ارائه کرد. وی در این مطالعه مثالی بزرگتر از سایر مثال‌های حل شده پیشین را حل کرد [۴۷]. فتحعلی در سال ۲۰۰۶ الگوریتم ژنتیک را برای حل مسائل p – میانه با وزن‌های مثبت و منفی ارائه کرد [۵۴].

در همین سال فتحعلی و تقی زاده الگوریتم جستجوی همسایگی را برای حل مسائل p – میانه با وزن‌های مثبت و منفی ارائه کردند [۵۷]. در سال ۲۰۰۷ گالینیر^{۵۳} و هرتز^{۵۴} مسئله پوششی را مورد بررسی قرار دادند و سه الگوریتم دقیق برای حل آن ارائه نمودند. همچنین آن‌ها حالت ابتکاری این سه الگوریتم را نیز ارائه نموده و مورد آنالیز قرار دادند [۷۲]. در طی این سال‌ها هر کدام از این مسائل به‌طور گسترده و توسط بسیاری از محققین و پژوهشگران مورد مطالعه و بررسی قرار گرفتند و امروزه نتایج بسیار جالب و ارزنده‌ای در هر یک از این زمینه‌ها ارائه شده است. به‌عنوان نمونه می‌توان به مراجع [۳۷، ۱۱۶، ۱۳۶] اشاره نمود. در ادامه دو نوع از پرکاربردترین مسائل مکان‌یابی گسسته معرفی خواهند شد.

۲.۲.۲ سرویس‌دهندگان

فرض کنید p تعداد سرویس‌دهنده‌ها باشد. حالت‌های زیر را در نظر بگیرید:

۱. $P=1$ ؛ در این حالت فقط یک فعالیت (سرویس‌دهنده) باید مستقر شود.

⁴⁸Hansen

⁴⁹Mladenovic

⁵⁰Ernst

⁵¹Hrishnamoorthy

⁵²Ebery

⁵³Galinier

⁵⁴Hertz

۲. $P > 1$ ؛ در این حالت علاوه بر پیدا کردن محل استقرار سرویس‌دهنده‌ها باید چگونگی تخصیص مشتریان به سرویس‌دهندگان را نیز مشخص شوند.

در حالتی که چند سرویس‌دهنده مکان‌یابی می‌شوند، ممکن است ارتباط بین سرویس‌دهندگان جدید نیز دارای اهمیت باشد. مدل‌هایی که این هدف را برآورده می‌کنند شامل فاصله وزن دار بین سرویس‌دهندگان جدید هستند. به‌عنوان مثالی از این مورد می‌توان به مکان‌یابی تأسیسات نظامی اشاره کرد که در آن پراکنده شدن و فاصله داشتن سرویس‌دهندگان جدید میزان ضرر و خسارت و آسیب‌پذیری را کاهش می‌دهد. در مکان‌یابی پمپ‌های گاز و یا فست‌فودها، تصمیم‌گیرنده سعی در پراکنده کردن آن‌ها دارد تا از ایجاد رقابت مستقیم آن‌ها با یکدیگر جلوگیری کند. بر مبنای این پارامتر دو نوع مدل مختلف وجود دارد:

۱. مدل‌هایی که تعداد سرویس‌دهندگان جدید در آن ثابت است.

۲. مدل‌هایی که تعداد سرویس‌دهندگان جدید در آن متغیر است.

مدل‌های نوع دوم پیچیده‌تر از مدل‌های نوع اول هستند، زیرا باید متغیر دیگری که نشان‌دهنده تعداد سرویس‌دهندگان است به مدل اضافه شود. در مدل‌های مکان‌یابی دو حالت رخ می‌دهد. در بسیاری از مدل‌ها فرض بر این است که تصمیم‌گیرنده با یک فضای خالی روبه‌رو می‌شود که هیچ سرویس‌دهنده‌ای در آن وجود ندارد. در حالتی دیگر که مکان‌یابی شرطی^{۵۵} نامیده می‌شود، تعدادی سرویس‌دهنده موجود هستند و هدف اضافه کردن تعداد دیگری است. حداقل سه نوع ارتباط مختلف بین سرویس‌دهندگان می‌تواند وجود داشته باشد:

۱. فقط فاصله بین مشتریان و سرویس‌دهندگان در نظر گرفته شود، مثلاً در مکان‌یابی چند محل دفن زباله در یک کشور.

۲. فقط فاصله بین سرویس‌دهندگان محاسبه شود، مثلاً مکان‌یابی سایت‌های موشکی در یک ناحیه بدون جمعیت.

۳. هر دو نوع فاصله ذکر شده در بالا در نظر گرفته شود، مثلاً مکان‌یابی تعدادی سایت هسته‌ای در یک ناحیه دارای جمعیت.

۳.۲.۲ اندازه‌گیری فاصله

فاصله یک معیار عددی برای تشخیص میزان دور بودن اشیا است. در مسائل مکان‌یابی، فرض می‌شود که ارتباط بین تأسیسات مستقیماً وابسته به موقعیت مکانی آن‌ها باشد

⁵⁵Conditional

و معمولاً به شکل تابعی از فاصله بیان می‌شود. بنابراین مفهوم فاصله نقشی محوری در مکان‌یابی پیوسته دارد. با توجه به موقعیت و شرایط مسئله اندازه‌های مختلفی برای ارزیابی فاصله به کار می‌روند. استفاده از روند نرم‌ها به عنوان تابع تخمین فاصله در یک مسئله مکان‌یابی با فضای پیوسته اولین بار توسط لاو و موریس^{۵۶} پیشنهاد شد [۹۹]. وارد و وندل^{۵۷} نیز استفاده از نرم‌های بلوکی^{۵۸} را مورد مطالعه قرار دادند [۱۳۹، ۱۴۰]. برای اندازه‌گیری فاصله بین مشتریان و سرویس‌دهندگان معمولاً از مترهای زیر استفاده می‌شود:

فاصله مینکوفسکی (نرم L_p):^{۵۹} این فاصله بین دو نقطه $x = (x_1, x_2)$ و $y = (y_1, y_2)$ در صفحه به صورت زیر تعریف می‌شود:

$$L_p(x, y) = (k_1|x_1 - y_1|^p + k_2|x_2 - y_2|^p)^{1/p}; \quad 1 \leq p \leq \infty.$$

فاصله مینکوفسکی برای مدل کردن مسائلی که فاصله در جهت‌های مختلف تقارن ندارد، مناسب است. در مسائل مکان‌یابی از بین خانواده L_p ها، فاصله اقلیدسی با $p=2$ و همچنین با $k_1 = k_2 = 1$ و فاصله مستطیلی (منهتن) با $p=1$ و $k_1 = k_2 = 1$ بیشتر مورد استفاده هستند. فاصله اقلیدسی برای مدل کردن فاصله‌های هوایی یا دریایی و فاصله مستطیلی برای مدل کردن فاصله‌های خیابانی (به شرط اینکه خیابان‌ها یک‌طرفه نباشند) مناسب هستند.

فاصله شبکه‌ای: برابر با کوتاه‌ترین مسیر بین دو نقطه روی شبکه است. برای محاسبه این فاصله الگوریتم‌های مختلفی پیشنهاد شده است که از آن جمله می‌توان الگوریتم‌های دایکسترا^{۶۰} و فلوید^{۶۱} را نام برد [۱۴].

۴.۲.۲ وزن

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

⁵⁶Love and Morris

⁵⁷Ward and Wendell

⁵⁸Block norms

⁵⁹Minkowsky

⁶⁰Dijkstra

⁶¹Floyd

۵.۲.۲ مشتریان

در تعدادی از مدل‌های مکان‌یابی نقش مشتری‌ها نیز منظور شده است. لذا لازم است که نحوه توزیع و میزان تقاضای آن‌ها مشخص باشد. برای مثال، ممکن است مشتریان به‌طور یکنواخت در یک ناحیه توزیع شده‌اند و یا در نقاط مخصوصی از فضا، مثلاً رؤوس 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 آن آزاد باشند که در آن صورت سؤال این است که آیا همیشه مشتریان به نزدیک‌ترین سرویس‌دهنده مراجعه می‌کنند یا عوامل دیگری در انتخاب سرویس‌دهنده توسط مشتری دخیل هستند.

۶.۲.۲ اهداف تصمیم‌گیرنده

مدل می‌تواند یک یا چند هدفه باشد. یعنی تصمیم‌گیرنده می‌تواند به دنبال مکان بهینه برای استقرار سرویس‌دهنده تحت یک هدف یا چند هدف به‌طور همزمان باشد. در بین مدل‌های تک هدفه حداقل سه دسته وجود دارند:

۱. مینیمم‌سازی تابعی از فاصله: کمترین مجموع^{۶۲}، مینیماکس^{۶۳}.
۲. ماکزیمم‌سازی تابعی از فاصله: بیشترین مجموع^{۶۴}، ماکزیمین^{۶۵}.
۳. تابع‌های ترکیبی از فاصله.

۷.۲.۲ تابع هدف‌های پرکاربرد مکان‌یابی

در این بخش، لیستی از توابع هدف معروف مکان‌یابی با توضیح مختصری درباره هر کدام ارائه شده است.

ماکزین^{۶۶}: نقطه‌ای روی شبکه که مجموع فاصله‌های وزن‌دار را بیشینه کند.

مرکز: نقطه‌ای روی شبکه که بیشترین فاصله بین سرویس‌دهنده و مشتریان را کمینه کند.

مرکز کلی^{۶۷}: نقطه‌ای روی شبکه که ماکزیمم فواصل بین آن و هر نقطه‌ای روی شبکه را کمینه کند. تفاوت این مسئله با مسئله مرکز در این است که جواب مسئله

⁶²Minisum

⁶³Minimax

⁶⁴Maxisum

⁶⁵Maximin

⁶⁶Maxian

⁶⁷General Center

مرکز کلی لزوماً یکی از رئوس شبکه نیست و می‌تواند هر جای آن از جمله روی یال‌ها نیز قرار گیرد.

مسئله چند وسیله‌ای وبر^{۶۸}: p نقطه روی صفحه که مجموع فواصل بین مشتری و نزدیک‌ترین سرویس‌دهنده را کمینه می‌کند.

میانه^{۶۹}: نقطه‌ای روی شبکه که مجموع فواصل بین سرویس‌دهنده و مشتریان را کمینه کند.

میانه کلی^{۷۰}: نقطه‌ای روی شبکه که مجموع فواصل بین آن و هر نقطه روی شبکه را کمینه کند. تفاوت این مسئله با مسئله میانه در این است که جواب مسئله مرکز کلی لزوماً یکی از رئوس شبکه نیست و می‌تواند هر جای آن از جمله روی یال‌ها نیز قرار گیرد.

p -میانه^{۷۱}: استقرار p نقطه روی شبکه به‌گونه‌ای که مجموع فواصل بین مشتری تا نزدیک‌ترین سرویس‌دهنده کمینه گردد.

p -مرکز^{۷۲}: استقرار p نقطه روی شبکه به‌گونه‌ای که بیشترین فاصله بین مشتری و سرویس‌دهنده‌ها کمینه گردد.

۳.۲ برخی مسائل مهم مکان‌یابی

در این بخش برخی مدل‌های مسائل مکان‌یابی پرکاربرد آمده است.

۱.۳.۲ مسئله مکان‌یابی چند وسیله‌ای پیوسته

به‌طورکلی، مسائل مکان‌یابی پیوسته (دو بعدی) به‌دنبال یافتن مکان یک یا چند سرویس‌دهنده بر روی صفحه هستند. شکل کلی این مسئله به‌صورت زیر است [۹۹]:

$$\min W(X) = \sum_{i=1}^m \sum_{j=1}^n w_{ij} d(X_i, A_j) + \sum_{i=1}^{m-1} \sum_{r=i+1}^m w_{ir} d(X_i, X_r). \quad (1.2)$$

که در آن $X_i = (a_{i1}, a_{i2})$, $i = 1, \dots, m$, $X = (X_1, X_2, \dots, X_m)$ مختصات مکان سرویس‌دهنده‌ها و $A_j = (a_{j1}, a_{j2})$, $j = 1, \dots, n$ مراکز تقاضا هستند.

و w_{ij} , $i = 1, \dots, m$, $j = 1, \dots, n$ و w_{ir} , $i = 1, \dots, m-1$, $r = i+1, \dots, m$ به‌ترتیب هزینه انتقال کالا از سرویس‌دهنده i - ام به مرکز تقاضای j - ام و هزینه انتقال

⁶⁸Multi facility Weber Problem

⁶⁹Median

⁷⁰General Median

⁷¹P-median

⁷²P-center

کالا از سرویس‌دهنده i - ام به سرویس‌دهنده r - ام هستند. همچنین $d(X_i, A_j)$ و $d(X_i, X_r)$ به ترتیب فاصله بین سرویس‌دهنده i - ام و مرکز تقاضای j - ام و فاصله بین دو مرکز سرویس‌دهی i - ام و r - ام هستند. اگر فرض شود که $m = 1$ ، در این صورت مسئله ۱.۲ به مسئله تک وسیله‌ای زیر تبدیل می‌شود:

$$\min W(X) = \sum_{j=1}^n w_j d(X, A_j). \quad (2.2)$$

مسئله اخیر یکی از شناخته شده‌ترین مسائل مکان‌یابی پیوسته است که به مسئله تک وسیله‌ای فرما - وبر معروف است. در این مسئله n نقطه در صفحه موجود است. هدف یافتن نقطه‌ای جدید در صفحه است به طوری که مجموع کل فواصل نقاط موجود تا نقطه بهینه کمینه شود [۹۹]. همچنین اگر هدف پیدا کردن X به گونه‌ای باشد که فاصله وزنی X تا دورترین نقطه موجود کمینه شود، آنگاه مسئله را مسئله تک وسیله‌ای مینیماکس گویند که به صورت زیر است:

$$\min \max_{i=1, \dots, n} w_i d(X, A_i).$$

در سال ۱۹۳۷ وایزفیلد^{۷۳} روشی تکراری برای حل مسئله فرما - وبر ارائه کرد [۱۴۲]. مهلی در سال ۱۹۵۸ برای اولین بار الگوریتم تکراری وایزفیلد را برای مسئله مکان‌یابی چند وسیله‌ای با نرم اقلیدسی گسترش داد [۹۹]. در سال ۱۹۶۴ فرانسیس^{۷۴} مسئله چند وسیله‌ای با نرم مستطیلی را مورد بررسی قرار داد [۶۰]. وسولوزکی^{۷۵} و لاو^{۷۶} در سال ۱۹۷۱ به مکان‌یابی در صفحه با فاصله مستطیلی در بین مقاصد نقاط و نواحی مستطیلی پرداختند [۱۴۴]. همچنین وسولوزکی در سال ۱۹۷۴ مسئله مکان‌یابی خط بهینه در صفحه را بیان نمود [۱۴۳]. در سال ۱۹۷۸ درزرنر^{۷۷} و وسولوزکی مسئله مکان‌یابی روی سطح کروی را بررسی کردند [۴۶]. همچنین در سال‌های اخیر نیز برخی از پژوهشگران و محققان مطالعه در این زمینه را ادامه دادند که منجر به حصول نتایج جدیدی شده است [۲۰، ۳۲].

در حالت گسسته مکان و مسائل جدید تنها می‌تواند در نقاط موجود قرار گیرند و حالت خاصی از آن وقتی است که نقاط موجود روی یک شبکه قرار داشته و مکان و مسائل جدید نیز باید نقطه‌ای از شبکه باشند. در این حالت وقتی تابع هدف به صورت کمترین مجموع و یا مینیماکس است، به ترتیب مسئله را مسئله میانه و مرکز می‌نامند.

⁷³Weiszfeld

⁷⁴Francis

⁷⁵Wesolowsky

⁷⁶Love

⁷⁷Drezner

همچنین اگر هدف پیدا کردن مکان p وسیله جدید باشد، آنگاه این مسائل به ترتیب مسائل p - میانه و p - مرکز نامیده می‌شوند.

۲.۳.۲ مسئله p - میانه و مدل ریاضی آن

مسئله p - میانه کاربردهای فراوانی دارد که از آن جمله می‌توان به مکان‌یابی مراکز شبکه ارتباطی کامپیوتری، مراکز توزیع کالا، مراکز اداری، مراکز نظامی، ایستگاه‌های اتوبوس و مراکز پستی اشاره کرد.

در مسئله p - میانه، میانه‌ها می‌توانند هر نقطه‌ای از شبکه انتخاب شوند، که در این حالت مسئله را مسئله p - میانه محض^{۷۸} گویند. اما اگر هدف پیدا کردن نقاط میانه تنها روی رأس‌ها باشد مسئله را مسئله p - میانه رأسی^{۷۹} می‌نامند. قضیه زیر که توسط حکیمی ارائه شد، بیان می‌کند که مجموعه جواب‌های p - میانه شامل یک جواب p - میانه رأسی است [۷۵].

قضیه ۱.۳.۲ (حکیمی). [۷۵]. حداقل یک مجموعه متشکل از p رأس در مجموعه جواب‌های p - میانه وجود دارد که یک جواب برای مسئله p - میانه باشد.

با توجه به خاصیت بهینگی رأسی فوق اگر مسئله به صورت شمارشی حل شود برای انتخاب p میانه از بین n رأس نیاز به $\binom{n}{p}$ انتخاب می‌باشد. مثلاً برای $p=1$ به n انتخاب نیاز است که برای هر رأس باید $F(X)$ را محاسبه شود یعنی به زمان $O(n^2)$ نیاز می‌باشد که در آن $n = |V|$. در حالت کلی برای p - های بزرگ پیچیدگی زمانی این روش زیاد شده و مسئله NP - سخت می‌باشد [۶۳].

روش‌های تقریبی مختلفی برای حل مسئله p - میانه 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.

مسئله p - میانه را می‌توان به صورت زیر به یک مسئله برنامه‌ریزی خطی صفر و یک تبدیل کرد. مدل برنامه‌ریزی صفر و یک و عدد صحیح مسئله p - میانه به صورت

⁷⁸ Absolute p -median

⁷⁹ Vertex p -median

زیر است [۴۳]:

$$\begin{aligned}
 (P_{11}) \quad & \min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \\
 \text{s.t} \quad & \\
 & \sum_{i=1}^n x_{ij} = w_j, \quad j = 1, \dots, n, \\
 & \sum_{i=1}^n y_i = p, \\
 & x_{ij} \leq y_i w_j, \quad i, j = 1, \dots, n, \\
 & x_{ij} \geq 0, y_i \in \{0, 1\}, \quad i, j = 1, \dots, n,
 \end{aligned}
 \tag{۳.۲}$$

که در آن

$$y_i = \begin{cases} 1 & \text{اگر راس } i - \text{ ام به‌عنوان سرویس‌دهنده انتخاب شود} \\ 0 & \text{در غیر این صورت} \end{cases}$$

و w_j ، میزان تقاضای مشتری روی راس j - ام است و d_{ij} ، فاصله بین راس i - ام و j - ام می‌باشد. همچنین x_{ij} ، $i, j = 1, \dots, n$ ، مقداری از تقاضای مشتری j - ام است که توسط سرویس‌دهنده i - ام برآورده می‌شود. در مدل (P_{11}) مجموعه قیود اول نشان می‌دهند که تمام نیاز هر مشتری باید پوشش داده شود. قید دوم بیان می‌کند که تعداد سرویس‌دهنده‌ها p است. مجموعه قیود سوم دلالت دارند بر اینکه ابتدا باید سرویس‌دهنده در مکان i - ام تاسیس شود و سپس سرویس‌دهی انجام شود. مدل (P_{11}) را می‌توان به‌صورت یک مدل صفر و یک زیر نیز فرمول‌بندی نمود [۴۳]:

$$\begin{aligned}
 (P_{12}) \quad & \min \sum_{i=1}^n \sum_{j=1}^n w_j d_{ij} x_{ij} \\
 \text{s.t} \quad & \\
 & \sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n, \\
 & \sum_{i=1}^n y_i = p, \\
 & x_{ij} \leq y_i, \quad i, j = 1, \dots, n, \\
 & x_{ij}, y_i \in \{0, 1\}, \quad i, j = 1, \dots, n.
 \end{aligned}$$

تفاوت این مدل و مدل قبل در تعریف متغیر x_{ij} ، $i, j = 1, \dots, n$ است که به صورت صفر و یک زیر تعریف می‌شود:

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

همچنین مجموعه قیود اول بیانگر آن است که مشتری مکان $j -$ ام فقط می‌تواند از یک راس سرویس‌دهی شود. بقیه قیود همانند حالت قبل هستند. یکی از روش‌های ابتکاری برای مسئله $p -$ میانه که توسط مارنزاناً^{۸۰} ارائه شده است روش جستجوی همسایگی^{۸۱} می‌باشد [۱۰۰].

این روش از یک جواب شدنی برای مسئله، یعنی مجموعه‌ای شامل مکان p وسیله شروع شده و نقاط تقاضا به نزدیک‌ترین وسیله نسبت داده می‌شود. مجموعه گره‌های نسبت داده شده به هر وسیله یک همسایگی اطراف آن وسیله تشکیل می‌دهند. در هر همسایگی، مسئله ۱ - میانه حل شده و وسیله به این مکان جدید تغییر مکان داده می‌شود. سپس هنگامی که برای تمام همسایگی‌ها مکان وسائل دوباره پیدا شدند، همسایگی‌های جدید را به دست آورده و عمل تکرار می‌شود، تا زمانی که دیگر تغییری در همسایگی‌ها یا مکان وسائل صورت نگیرد.

بر اساس روش فوق تیتز و بارت [۱۳۴] نیز الگوریتمی ارائه کرده‌اند که در آن برای تغییر مکان یک وسیله، دیگر رأس‌ها 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 مؤثر است. هانسن و ملادنویچ [۷۹] مسئله $p -$ میانه 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^{۸۲} و بر اساس روش تغییر مکان وسائل می‌باشد، در مرجع [۳۹] آمده است که در آن ابتدا نقطه ۱ - میانه به دست آمده و در مجموعه جواب قرار می‌گیرد. سپس در هر گام یک رأس به این مجموعه اضافه شده و بررسی می‌شود که آیا می‌توان رأس دیگری را جایگزین رأس‌های انتخاب شده کرد تا تابع هدف کاهش یابد یا خیر. در صورت مثبت بودن جواب، رأس را جایگزین کرده و دوباره عمل بررسی انجام می‌شود. پس از p گام یک جواب تقریبی برای مسئله به دست می‌آید. این الگوریتم یک مجموعه مانند S شامل p رأس را پیدا

⁸⁰Maranzana

⁸¹Neighborhood Search

⁸²Greedy

می‌کند که لزوماً جواب بهینه نیست ولی برای هر مجموعه دیگر مانند X که شامل p رأس از شبکه باشد اگر X و S دارای $p - 1$ رأس مشترک باشند، آنگاه داریم $F(S) \leq F(X)$.

از دیگر روش‌های ابتکاری که برای مسئله p -میانه به کار رفته است روش‌های الگوریتم ژنتیک^{۸۳} و جستجوی ممنوع^{۸۴} است. اولین کسانی که الگوریتم ژنتیک را برای مسئله p -میانه به کار بردند، هاسج^{۸۵} و گودچایلد^{۸۶} [۸۴] بودند. سپس دیبل^{۸۷} و دشام^{۸۸} [۴۴]، ارکوت^{۸۹} و همکاران [۴۹] و کورا^{۹۰} و همکاران [۴۰] روش‌هایی با استفاده از الگوریتم‌های ژنتیک برای مسئله p -میانه ارائه کردند. با استفاده از جستجوی تابو نیز مورنو^{۹۱} و همکاران [۱۰۷] و رلند^{۹۲} و همکاران [۱۲۱] روش‌هایی برای مسئله p -میانه ارائه کرده‌اند.

برای بررسی کارایی یک روش ابتکاری اگر بتوان کران‌هایی برای جواب بهینه یافت، آنگاه می‌توان تعیین کرد که آیا جواب یک روش ابتکاری به جواب بهینه نزدیک شده است یا خیر. یک روش برای تعیین کرانی برای جواب بهینه، روش کاهشی لاگرانژ^{۹۳} است. در این روش با انتقال یک یا چند محدودیت همراه با ضرایبی به نام ضرایب لاگرانژ، به تابع هدف یک مسئله جدید جایگزین مسئله اصلی می‌شود که جواب آن کرانی برای مسئله اصلی است. روش کاهشی لاگرانژ برای مسئله p -میانه در مراجع [۶۱] و [۴۳] آمده است. در مرجع [۷۷] نیز بر روی روش‌های تقریبی گوناگون (ابتکاری و برنامه‌ریزی ریاضی) که توسط افراد مختلف ارائه شده بررسی‌هایی به عمل آمده است.

برای مسئله p -میانه روی درخت، کریو^{۹۴} و حکیمی الگوریتمی دقیق با پیچیدگی زمانی $O(p^2 n^2)$ ارائه کرده‌اند [۸۸]. بعدها تمیر^{۹۵} روش حل آن‌ها را با الگوریتمی با پیچیدگی زمانی $O(pn^2)$ بهبود بخشید [۱۳۱].

⁸³Genetic Algorithm

⁸⁴Tabu Search

⁸⁵Hosage

⁸⁶Goodchild

⁸⁷Dibble

⁸⁸Densham

⁸⁹Erkut

⁹⁰Correa

⁹¹Moreno-perez

⁹²Rolland

⁹³Lagrangean Relaxation

⁹⁴Kariv

⁹⁵Tamir

مسئله ۱- میانه روی درخت

گلدمن در سال ۱۹۷۱ به مسئله یک میانه روی شبکه‌ها پرداخت و نشان داد که حل این مسئله با استفاده از روش شمارش دارای پیچیدگی زمانی $O(n^3)$ می‌باشد [۱۰۵]. همچنین وی برای حالتی که شبکه درخت باشد بر اساس سه لم زیر الگوریتم اکثریت وزنی را که دارای پیچیدگی زمانی $O(n)$ می‌باشد ارائه نمود.

لم ۱.۳.۲. اگر X درختی با n راس v_1, v_2, \dots, v_n مفروض باشد. همچنین اگر S و T دو زیر درخت حاصل از حذف یال (v_s, v_t) باشند که $s \in S$ ، $t \in T$ و

$$W(S) = \sum_{x \in S} w(x),$$

$$W(T) = \sum_{x \in T} w(x).$$

اگر $W(S) \geq W(T)$ آنگاه میانه m در زیر درخت S واقع می‌باشد.

برهان. به مرجع [۱۰۵] مراجعه شود.

□

لم ۲.۳.۲. فرض شود که S و T دو زیر درخت باشند که از حذف یال (s, t) از درخت X به دست آمده باشند. حال اگر $W(S) \geq W(T)$ آنگاه مسئله یافتن میانه روی درخت X تبدیل می‌شود به یافتن میانه روی زیر درخت N_S که در آن N_S همان درخت S است با این تفاوت که وزن راس s با $w(s) + W(T)$ جایگزین شده است.

برهان. به مرجع [۱۰۵] مراجعه شود.

□

لم ۳.۳.۲. اگر وزن راسی از یک درخت (گراف) بیشتر از نصف مجموع وزن‌های همه رئوس باشد آنگاه آن راس میانه درخت می‌باشد.

برهان. به مرجع [۱۰۵] مراجعه شود.

□

الگوریتم اکثریت وزنی

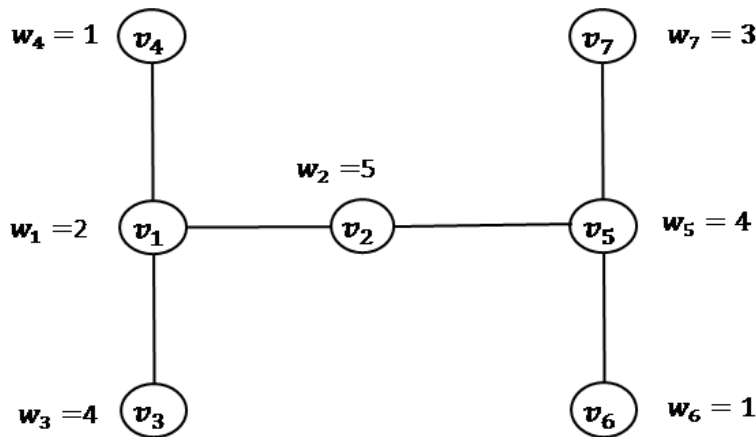
گام اول. اگر X شامل یک راس است توقف کنید همان راس جواب مسئله است.

گام دوم. به‌ازای هر راس انتهایی مانند v اگر $w(v) \geq \frac{W(X)}{۲}$ آنگاه به گام چهارم بروید. در غیر این صورت به گام سوم بروید.

گام سوم. فرض کنید u راس مجاور v باشد. وزن راس v را به راس u اضافه کرده و راس v و یال (u, v) را حذف کنید و به گام یک برگردید.

گام چهارم. توقف کنید. راس v جواب مسئله است.

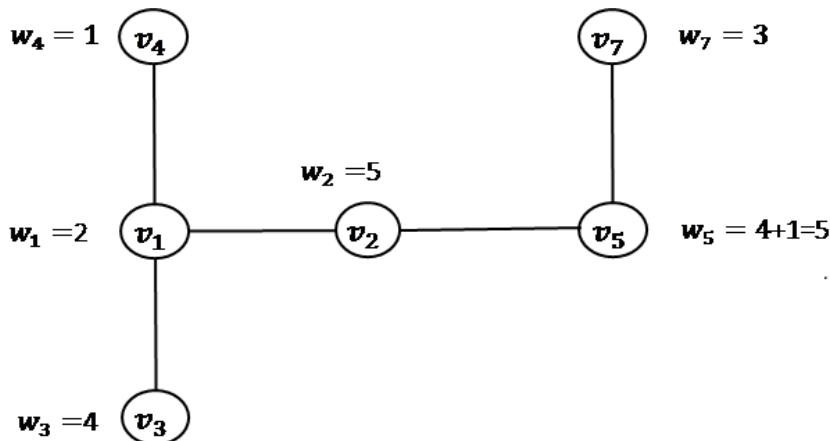
مثال ۱.۳.۲. درخت X با هفت راس و شش یال را در نظر بگیرید [شکل ۱.۲]: به



شکل ۱.۲: درختی با هفت راس و شش یال.

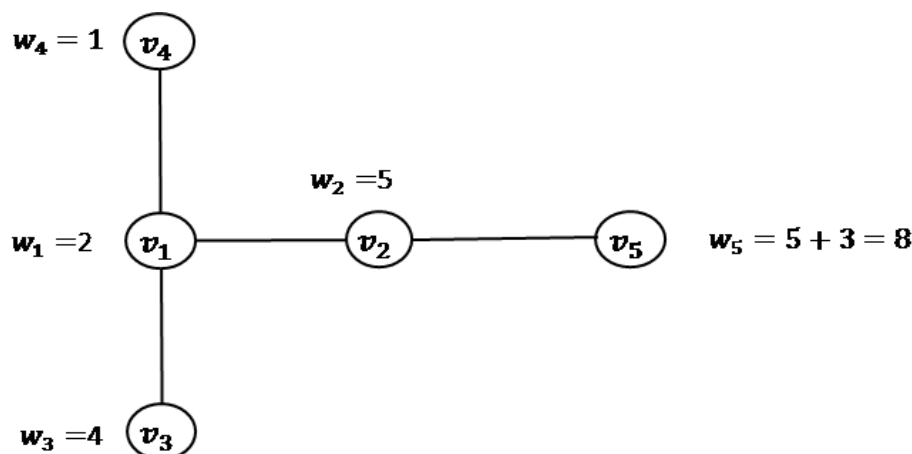
وضوح $W(X) = ۲۰$.

از راس دلخواه v_6 شروع می‌کنیم. از آنجایی که $w_6 \not\geq \frac{W(X)}{۲} = ۱۰$ وزن راس v_6 را به راس مجاور آن یعنی v_5 اضافه کرده و یال (v_5, v_6) را از درخت X حذف می‌کنیم. از این رو $w_5 = ۵$ و درخت X به صورت ۲.۲ تبدیل می‌شود.



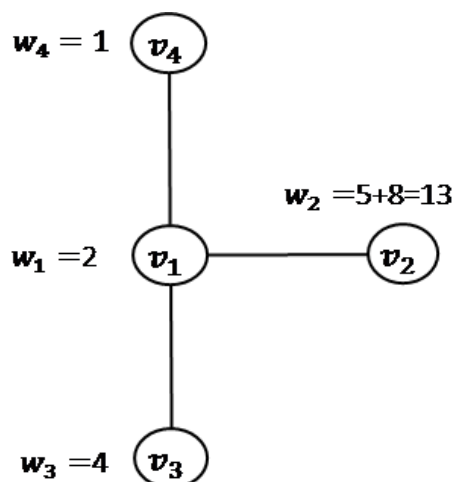
شکل ۲.۲: درخت حاصل از حذف راس v_6 و یال (v_5, v_6) .

حال به گام اول بر می‌گردیم و راس دلخواه v_7 را انتخاب می‌کنیم. از آنجایی که $w_7 \neq 10$ وزن راس v_7 را به راس مجاور آن یعنی v_8 افزوده و یال بین این دو راس را حذف می‌کنیم که درخت ۳.۲ با $w_8 = 8$ حاصل می‌شود.



شکل ۳.۲: درخت حاصل از حذف راس v_7 و یال (v_7, v_8) .

دوباره به گام اول بر می‌گردیم و راس v_8 را در نظر می‌گیریم. از آنجایی که $w_8 \neq 10$ وزن این راس را به راس مجاور آن یعنی v_2 افزوده و یال بین این دو راس را حذف می‌کنیم که درخت ۴.۲ با $w_2 = 13$ حاصل می‌شود. حال به گام اول بر می‌گردیم و



شکل ۴.۲: درخت حاصل از حذف راس v_8 و یال (v_2, v_8) .

راس v_2 را در نظر می‌گیریم. از آنجایی که $w_2 = 13 > 10$ توقف می‌کنیم و نتیجه می‌شود که راس v_2 میانه درخت X است.

مسئله ۲ - میانه روی درخت

مسئله ۲ - میانه روی شبکه را در نظر بگیرید. حل این مسئله با استفاده از روش شمارشی دارای پیچیدگی زمانی $O(n^3)$ می‌باشد اما برای حالتی که شبکه درخت باشد گلدمن الگوریتم خطی حذف یال را به صورت زیر ارائه نمود:

الگوریتم حذف یال [۱۰۵]

گام اول. فرض کنید $E = \{e_1, e_2, \dots, e_{n-1}\}$ مجموعه تمام یال‌های درخت X با n راس باشد. به ازای $i = 1, 2, \dots, n-1$ یال e_i را حذف می‌کنیم. در نتیجه دو زیر درخت حاصل می‌شود که به ترتیب $T_{\lambda_{e_i}}$ و $T_{\chi_{e_i}}$ می‌نامیم.

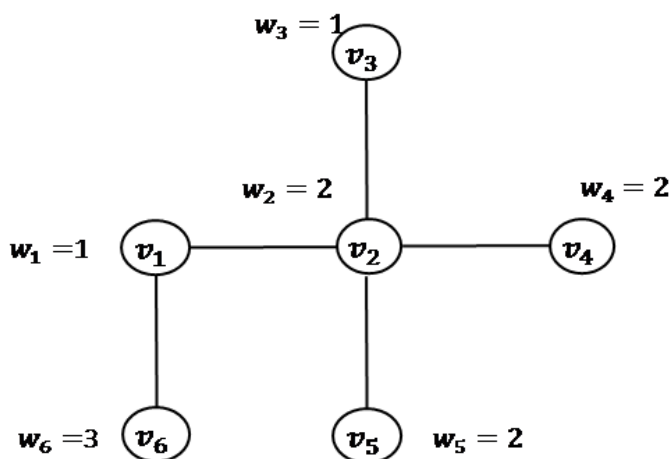
گام دوم. با استفاده از الگوریتم اکثریت وزنی در هر یک از دو زیر درخت $T_{\lambda_{e_i}}$ و $T_{\chi_{e_i}}$ یک میانه را می‌یابیم و به ترتیب $m_{\lambda_{e_i}}$ و $m_{\chi_{e_i}}$ نام‌گذاری می‌کنیم.

گام سوم. مقدار توابع هدف $f_{\lambda_{e_i}}$ و $f_{\chi_{e_i}}$ را برای هر دو زیر درخت $T_{\lambda_{e_i}}$ و $T_{\chi_{e_i}}$ به دست آورده و با هم جمع می‌کنیم تا هزینه نهایی برای مسئله اصلی به صورت زیر به دست آید:

$$f_{e_i} = f_{\lambda_{e_i}} + f_{\chi_{e_i}}.$$

گام چهارم. به ازای $i = 1, 2, \dots, n-1$ کمترین مقدار f_{e_i} را می‌یابیم. فرض کنید این کمترین مقدار با حذف یال e_k به دست آمده باشد. در این صورت $m_{\lambda_{e_k}}$ و $m_{\chi_{e_k}}$ میانه‌های درخت X می‌باشند و مقدار بهینه عبارت است از $f_{e_k} = f_{\lambda_{e_k}} + f_{\chi_{e_k}}$.

مثال ۲.۳.۲. درخت ۵.۲ با طول یال‌های واحد را در نظر بگیرید.



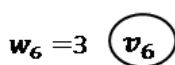
شکل ۵.۲: درختی با شش راس و پنج یال.

ابتدا فرض کنید یال بین دو راس v_6 و v_1 را حذف کنیم. با حذف این یال دو زیر درخت حاصل می‌شود که در شکل ۶.۲ نشان داده شده‌اند. با استفاده از الگوریتم اکثریت وزنی در هر یک از این دو زیر درخت یک میانه را می‌یابیم که به ترتیب عبارتند

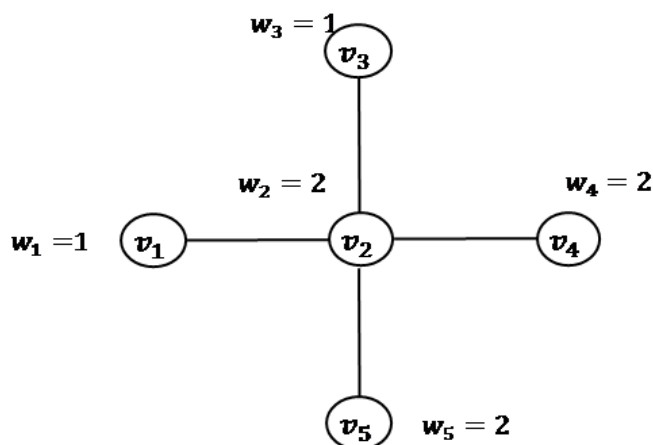
از راس v_6 در زیر درخت اول و راس v_2 در زیر درخت دوم. بنابراین تابع هزینه به صورت زیر است:

$$f(v_2, v_6) = f_1 + f_2 = 0 + 6 = 6$$

X_1 :



X_2 :



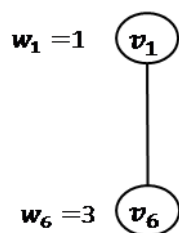
شکل ۶.۲: زیر درخت‌های حاصل از حذف یال (v_1, v_6) .

در مرحله بعد یال بین رئوس v_2 و v_1 را حذف کنیم. در این صورت دو زیر درخت به صورت شکل ۷.۲ به دست می‌آید. در نتیجه میانه‌ها عبارتند از رئوس v_2 و v_6 و مقدار تابع هدف برابر است با:

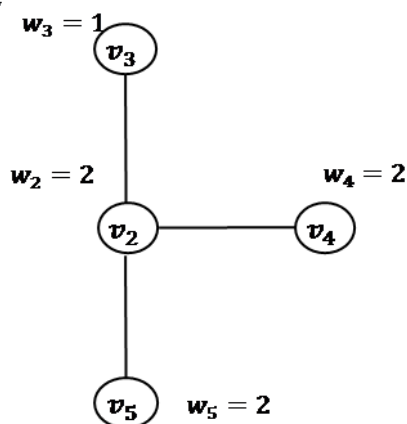
$$f(v_6, v_2) = f_1 + f_2 = 1 + 5 = 6.$$

حال با حذف یال بین رئوس v_2 و v_3 دو زیر درخت که در شکل ۸.۲ نشان داده شده‌اند

X_1 :



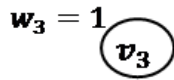
X_2 :



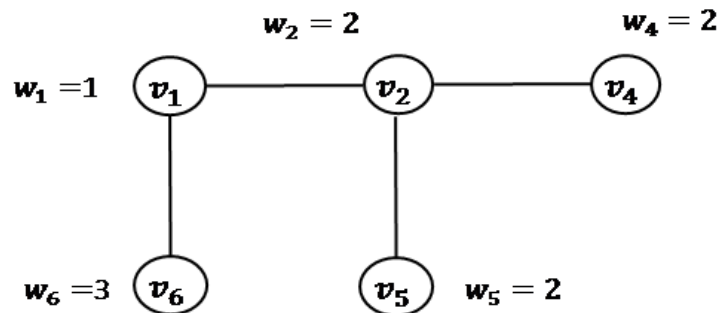
شکل ۷.۲: زیر درخت‌های حاصل از حذف یال (v_1, v_2) .

به دست می‌آیند. در این حالت میانه‌ها عبارتند از رئوس v_2 و v_3 و مقدار تابع هدف

X_1 :



X_2 :



شکل ۸.۲: زیر درخت‌های حاصل از حذف یال (v_2, v_3) .

به صورت زیر است:

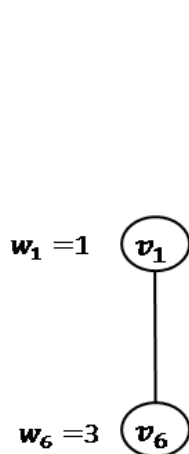
$$f(v_3, v_2) = f_1 + f_2 = 11 + 0 = 11.$$

از حذف یال بین رئوس v_5 و v_2 زیر درخت‌های شکل ۹.۲ نتیجه می‌شود. در این صورت میانه‌ها عبارتند از رئوس v_5 و v_2 و مقدار تابع هدف برابر است با:

$$f(v_5, v_2) = f_1 + f_2 = 10 + 0 = 10.$$

و در نهایت با حذف یال بین رئوس v_4 و v_2 شکل ۱۰.۲ نتیجه می‌شود که میانه‌ها

X_1 :



X_2 :

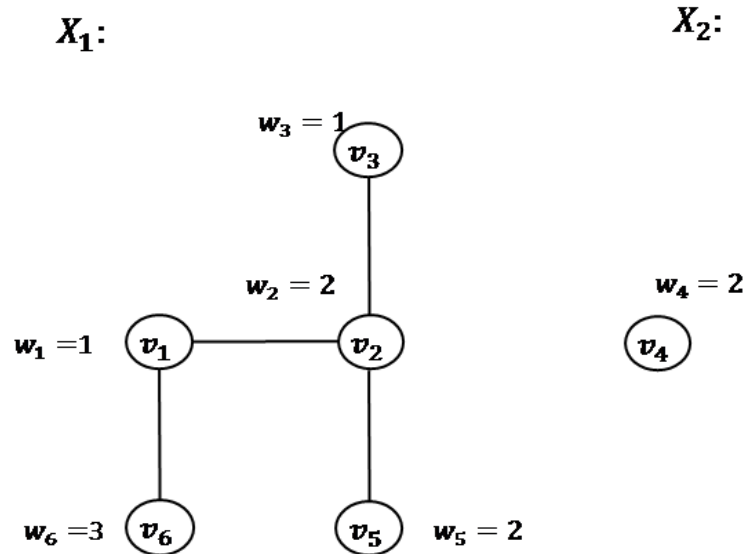


شکل ۹.۲: زیر درخت‌های حاصل از حذف یال (v_2, v_5) .

عبارتند از رئوس v_4 و v_2 و مقدار تابع هدف برابر است با:

$$f(v_4, v_2) = f_1 + f_2 = 10 + 0 = 10.$$

ملاحظه می‌شود که با حذف یال (v_1, v_6) کمترین مقدار تابع هزینه به دست می‌آید. لذا مجموعه رئوس $\{v_2, v_6\}$ عبارت است از مجموعه دو میانه درخت X .



۳.۳.۲ مسئله مکان‌یابی p - مرکز

مسئله مکان‌یابی p - مرکز عبارت است از تعیین مکان p سرویس‌دهنده روی شبکه به طوری که بیشترین فاصله وزن‌دار مشتری‌ها تا نزدیکترین سرویس‌دهنده کمینه شود. فرض کنید $X = \{x_1, \dots, x_p\}$ یک مجموعه شامل مکان p سرویس‌دهنده از شبکه N باشد. آنگاه در مسئله p - مرکز داریم:

$$\min_X \left(\max_{j=1, \dots, n} w_j D(v_j, X) \right). \quad (4.2)$$

که در آن

$$D(v_j, X) = \min_{x_i \in X} (d(v_j, x_i)).$$

مدل برنامه‌ریزی صفر و یک و عدد صحیح برای مسئله p - مرکز به صورت زیر است
[۴۳]:

$$\begin{aligned}
 (P_{13}) \quad & \min Z \\
 \text{s.t} \quad & \\
 & \sum_{i=1}^n x_{ij} = w_j, \quad j = 1, \dots, n, \\
 & \sum_{i=1}^n y_i = p, \\
 & x_{ij} \leq y_i w_j, \quad i, j = 1, \dots, n, \\
 & Z \geq \sum_{i=1}^n x_{ij} d_{ij}, \quad j = 1, \dots, n, \\
 & x_{ij} \geq 0, y_i \in \{0, 1\}, \quad i, j = 1, \dots, n,
 \end{aligned}$$

(۵.۲)

که در آن $y_i, i = 1, \dots, n$ و $x_{ij}, i, j = 1, \dots, n$ مانند مدل مسئله p - میانه تعریف می‌شوند. در مدل اخیر مجموعه قیود اول نشان می‌دهند که تمام نیاز هر مشتری باید پوشش داده شود. قید دوم بیان می‌کند که تعداد سرویس‌دهنده‌ها p است. مجموعه قیود سوم دلالت دارند بر اینکه ابتدا باید سرویس‌دهنده در مکان i - ام تاسیس شود و سپس سرویس‌دهی انجام شود.

در مدل (۵.۲) اگر $p = 1$ باشد آنگاه مسئله تبدیل به یافتن یک مرکز روی شبکه می‌شود. جواب بهینه به دست آمده از حل این مدل مرکز راسی نامیده می‌شود. کوکاینه و همکاران مسئله p - مرکز را بر روی شبکه درخت مورد مطالعه قرار دادند و الگوریتم خطی برای آن ارائه نمودند [۸۱]. سپس اسلیتر الگوریتم خطی برای این مسئله روی شبکه در سال ۱۹۸۲ ارائه نمود [۱۲۹]. همچنین هندلر در سال ۱۹۷۳ به حل این مسئله روی شبکه درخت بدون وزن پرداخت [۸۰]. در سال ۱۹۸۵ مینیکا به حل مسئله مسیر مرکزی زمانی که طول مسیر از مقدار خاص l بیشتر نباشد پرداخت [۱۰۴].

مسئله یک مرکز روی درخت بدون وزن

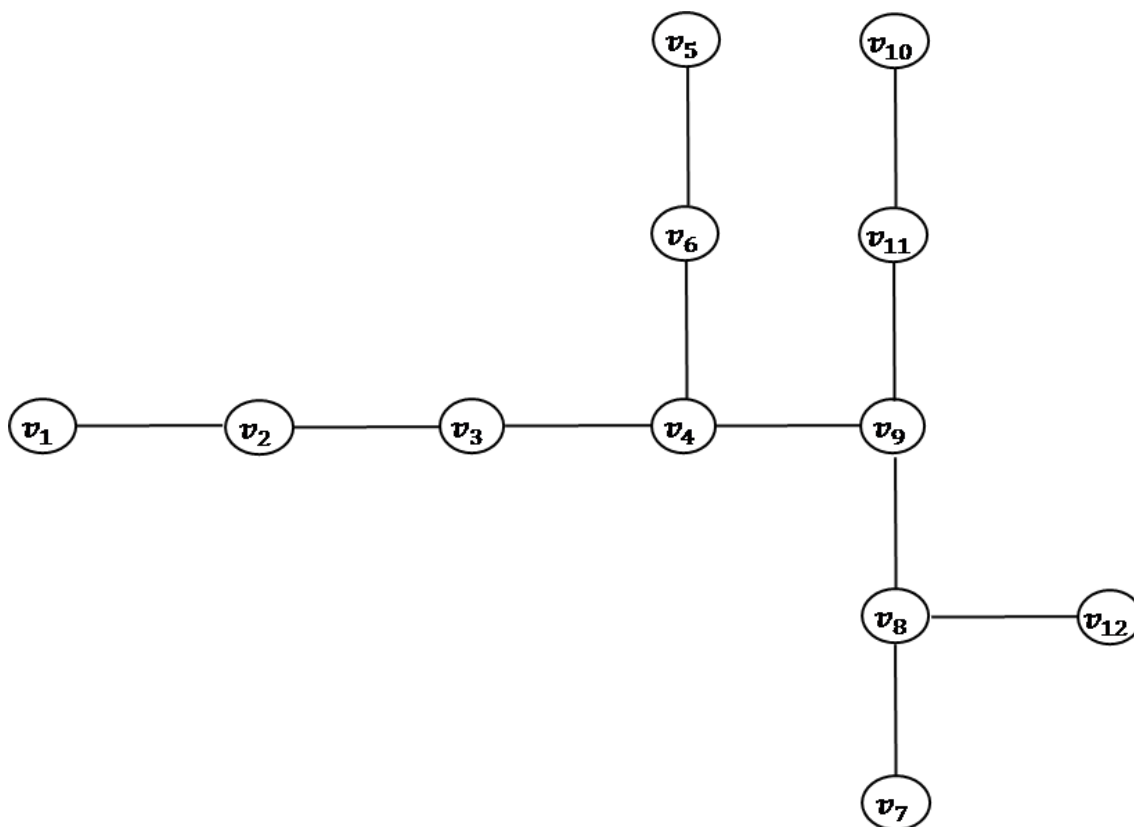
همان‌طور که گفته شد مسئله مرکز عبارت است از یافتن نقطه‌ای که فاصله دورترین نقطه در درخت از آن دارای کمترین مقدار ممکن باشد. درخت X متشکل از n راس v_1, v_2, \dots, v_n را در نظر بگیرید. هندلر الگوریتم زیر را برای یافتن مرکز ارائه نمود:

الگوریتم یک مرکز [۸۰]

- گام اول. راس دلخواه v_1 را در درخت X انتخاب کنید.
- گام دوم. دورترین راس از v_1 را بیابید و آن را v_2 بنامید.
- گام سوم. دورترین راس از v_2 را بیابید و آن را v_3 بنامید.
- گام چهارم. نقطه میانی مسیر بین دو راس v_2 و v_3 مرکز درخت X می‌باشد.

مثال ۳.۳.۲. درخت بدون وزن X نشان داده شده در شکل ۱۱.۲ با طول یال‌های واحد را در نظر بگیرید.

می‌خواهیم مرکز این درخت را بیابیم. بدین منظور بر اساس الگوریتم هندلر راس دلخواه v_8 را انتخاب می‌کنیم. دورترین راس از آن عبارت است از راس v_1 . همچنین دورترین راس از راس v_1 عبارت است از راس v_7 . در نهایت نقطه میانی مسیر بین رئوس v_1 و v_7 راس v_4 است. بنابراین راس v_4 مرکز درخت X می‌باشد.



شکل ۱۱.۲: درخت بدون وزن.

فصل ۳

مقدمه‌ای بر مسائل مکان‌یابی اصلاحی

در تقابل با مسائل مکان‌یابی کلاسیک ارائه شده، در سال‌های اخیر مسائل مکان‌یابی اصلاحی شامل مدل‌های معکوس، مدل‌های ترفیع/تنزیل، مدل‌های عدم قطعیت و مدل‌های مکان‌یابی آرمانی بسیار مورد توجه قرار گرفته‌اند که در ادامه به بررسی آن‌ها می‌پردازیم.

۱.۳ مسائل مکان‌یابی پشتیبان

نظریه مکان‌یابی به‌طور سنتی علاقه‌مند به مسائلی است که وزن سرویس‌دهنده‌ها و در دسترس بودن آن‌ها دقیقاً مشخص است. اما اغلب این امکان‌پذیر نیست که یک برآورد دقیق از تمامی این پارامترها داشته باشیم. بخش مهمی از داده‌های زندگی واقعی، اغلب شامل عدم قطعیت است و این پارامترها ممکن است با زمان تغییر کنند. بنابراین در سال‌های اخیر مدل‌های مکان‌یابی شامل عدم قطعیت بررسی شده و برخی حالت‌های آن‌ها تعریف و مورد مطالعه قرار گرفته است. شنایدر^۱ و دسکین^۲ [۱۳۰]، مدل قابلیت اطمینان را بررسی کردند که در این مدل بعضی اوقات ممکن است سرویس‌دهنده‌ها شکست بخورند و مشتریان اختصاص داده شده به

^۱Snyder

^۲Daskin

این سرویس‌دهنده‌ها مجبورند از سرویس‌دهنده‌های در حال کار سرویس بگیرند. وانگ وو و همکارانش^۳ مسئله ۲- میانه را همانند مدل قابلیت اطمینان بررسی کردند [۱۳۸]. در این مدل هر سرویس‌دهنده ممکن است با یک احتمال داده شده‌ای شکست بخورد. به محض این که یک سرویس‌دهنده شکست بخورد، با این فرض که سرویس‌دهنده‌ها نمی‌توانند به‌طور هم‌زمان شکست بخورند، سرویس‌دهنده دیگر مسئولیت سرویس‌دهی را به عهده می‌گیرد. آن‌ها این مدل را که در آن به دنبال یافتن یک زوج از رئوس بودند که مجموع فواصل وزنی از تمام رئوس به سرویس‌دهنده‌های در حال کار را کمینه کند، مسئله ۲- میانه پشتیبان^۴ نامیدند و یک الگوریتم با زمان $O(n \log n)$ برای آن پیشنهاد دادند که n تعداد رئوس درخت مورد بررسی است.

۱.۱.۳ مسئله مکان‌یابی ۲- میانه پشتیبان

فرض کنید که G گراف ساده بدون جهت $G = (V(G), E(G), w, l)$ باشد که $V(G)$ مجموعه رأس‌ها و $E(G)$ مجموعه یال‌های G ، $w : V(G) \rightarrow \mathbb{R}^+$ تابع وزنی رأس‌ها و $l : E(G) \rightarrow \mathbb{R}^+$ تابع طول یال‌ها باشد. یک میانه از G رأسی مانند m است که

$$\sum_{v \in V(G)} w(v) d_G(m, v) = \min_{u \in V(G)} \sum_{v \in V(G)} w(v) d_G(u, v). \quad (1.3)$$

برای ساده‌تر شدن نمایش $\sum_{v \in V} w(v) d_G(u, v)$ از نماد $D(V, u)$ با $V \subseteq V(G)$ استفاده می‌شود. همچنین فاصله بین دو رأس u و v با نماد $d(u, v)$ نمایش داده می‌شود. برای هر زوج رأس (u, v) قرار دهید $\Pi(u, v) = (U, V)$ که $U = \{x : d(x, u) \leq d(x, v)\}$ و $V = V(T) - U$. توجه کنید که در یک درخت هریک از مجموعه‌های U و V یک زیرگراف همبند را تولید می‌کنند که این دو مجموعه را می‌توان با حذف یال $e \in E(T)$ به‌دست آورد.

فرض شود ρ_1 و ρ_2 احتمال‌های شکست دو سرویس‌دهنده در مسئله ۲- میانه پشتیبان باشند. جواب مسئله با یک زوج رأس (u_1, u_2) نشان داده می‌شود که u_1 مکان سرویس‌دهنده با احتمال شکست ρ_1 و u_2 مکان سرویس‌دهنده با احتمال شکست ρ_2 است.

هدف مسئله ۲- میانه پشتیبان کمینه کردن مجموع فاصله از تمام رأس‌ها به مجموعه سرویس‌دهنده‌های در حال کار است که به‌صورت زیر نوشته می‌شود:

$$C_\rho(v_1, v_2) = (1 - \rho) \left(\sum_v w(v) d(v, \{v_1, v_2\}) \right) + \rho \left(\sum_v w(v) d(v, v_2) + \sum_v w(v) d(v, v_1) \right). \quad (2.3)$$

³Wang Wu

⁴Backup 2-Median Problem

۲.۱.۳ مسئله ۲- میانه پشتیبان روی درخت

درخت T و احتمال شکست ρ داده شده است. مسئله ۲- میانه پشتیبان روی درخت T یک زوج از رئوس (m_1, m_2) را جستجو می‌کند به گونه‌ای که تابع هدف زیر را کمینه کند:

$$\begin{aligned} C_\rho(v_1, v_2) &= (1 - \rho)(D(V_1, v_1) + D(V_2, v_2)) + \rho(D(V_1, v_1) + D(V_2, v_1)) \\ &\quad + \rho(D(V_1, v_2) + D(V_2, v_2)) \\ &= D(V_1, v_1) + \rho D(V_2, v_1) + D(V_2, v_2) + \rho D(V_1, v_2) \end{aligned} \quad (3.3)$$

که در آن $(V_1, V_2) = \prod(v_1, v_2)$ و $v_1, v_2 \in V(T)$ می‌باشد.

روش حل مسئله

در این قسمت مسئله ۲- میانه پشتیبان، روی درخت‌های با رأس‌های وزن‌دار بررسی شده است. فرض کنید درخت ورودی T در میانه خود یعنی m ریشه‌دار شده است. روش ارائه شده، مشابه روش حذف یال در مسئله ۲- میانه کلاسیک روی درخت است که در فصل دوم به صورت مختصر توضیح داده شده است. فرض کنید s_1, s_2 دو میانه T باشند. قرار دهید $(S_1, S_2) = \prod(s_1, s_2)$.

در مراجع [۷۰، ۹۴] نشان داده شده است که s_1 و s_2 باید به ترتیب میانه‌های زیردرخت‌های ایجاد شده توسط S_1 و S_2 باشند. در لم ۱.۱.۳ یک نتیجه مشابه برای مسئله ۲- میانه پشتیبان بیان شده است. برای یک یال $e = (x, y) \in E$ که پدر x پدر y است، فرض کنید $X(e)$ و $Y(e)$ دو زیر درخت به دست آمده از حذف یال e در درخت T باشند که x در $X(e)$ است و وزن‌های x و y به ترتیب برابر با $w(x) + \rho w(T_y)$ و $w(y) + \rho w(T - T_y)$ باشند. وزن تمام رأس‌های دیگر و طول تمام یال‌های باقی‌مانده تغییر داده نمی‌شوند. زیر درخت ریشه‌دار شده در u از $X(e)$ با $X_u(e)$ نمادگذاری می‌شود. به طور مشابه $Y_v(e)$ زیر درختی از $Y(e)$ است که در v ریشه‌دار شده است.

لم ۱.۱.۳. [۱۳۸] اگر (m_1, m_2) ۲- میانه پشتیبان روی رأس‌های مختلف درخت T باشد، آنگاه m_1 و m_2 به ترتیب میانه‌های $X(e)$ و $Y(e)$ هستند که $e = (x, y)$ یک یال مرکزی m_1 و m_2 است.

تعریف ۱.۱.۳. فرض کنید که $e = (x, y)$ یالی در $E(T)$ و پدر x پدر y باشد. با حذف یال e دو زیردرخت حاصل می‌شود. میانه زیردرخت شامل رأس x میانه بالا و میانه زیر

درخت شامل y میانه پایین یال e نامیده می‌شوند و به ترتیب با نمادهای m_L و m_U نشان داده می‌شوند.

الگوریتمی که در ادامه برای حل مسئله ارائه شده است بر پایه لم ۱.۱.۳ قرار دارد، در نتیجه باید به‌ازای هر یال e یک زوج میانه یافت شود.

قضیه ۱.۱.۳ [۱۳۸] رأس v میانه T است اگر و تنها اگر $w(T_v) \geq w(V)/2$ و $w(T_u) \leq w(V)/2$ برای هر فرزند u از v .

حال اگر V_L و V_U به ترتیب مجموعه رأس‌های $X(e)$ و $Y(e)$ باشند. تابع هدف به صورت تابع هزینه وابسته به یال بازنویسی می‌شود یعنی:

$$C'_\rho(e) = D(V_L, m_L) + \rho D(V_U, m_L) + D(V_U, m_U) + \rho D(V_L, m_U) \quad (4.3)$$

که در آن (m_L, m_U) زوج میانه‌های یال e هستند و $(V_L, V_U) = \Pi(m_L, m_U)$. الگوریتم یافتن ۲-میانه پشتیبان، پشتیبان ۲-میانه^۵ [۱۳۸]، نامیده می‌شود که به شکل زیر عمل می‌کند.

به جای محاسبه میانه‌های بالا و پایین یک یال به صورت همزمان، ابتدا میانه‌ی پایین تمام یال‌ها (خط ۳)، و سپس میانه بالای تمام یال‌ها (خط ۴) محاسبه می‌شود. یال‌های E با i اندیس‌گذاری شده و هریک از آن‌ها با e_i که $1 \leq i \leq n$ نمایش داده می‌شوند. پس از اجرای زیربرنامه‌های *UPPER-MEDIAN* و *LOWER-MEDIAN*، زوج میانه‌های هر یال e_i در $M_L[i]$ و $M_U[i]$ ذخیره می‌شود. بنابراین می‌توان هزینه وابسته به یال را به ازای هر زوج از میانه‌ها محاسبه و کمینه‌ترین هزینه وابسته به یال را پیدا کرد. (خط ۶ تا ۱۱).

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

شبه کد الگوریتم $BACKUP2MEDIAN(T, \rho)$

ورودی: درخت T و احتمال شکست ρ برای هر سرویس‌دهنده.
خروجی: ۲-میانه پشتیبان درخت ورودی.

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

⁵BACKUP2MEDIAN

- (۲) درخت T که در m ریشه‌دار شده‌است را در T^* قرار بده.
- (۳) زیربرنامه $LOWER - MEDIAN(T^*)$ را فراخوانی کرده و پس از اجرای آن تمام میانه‌های پایین یال‌ها را در آرایه M_L قرار بده.
- (۴) زیربرنامه $UPPER - MEDIAN(T^*)$ را فراخوانی کرده و پس از اجرای آن تمام میانه‌های بالای یال‌ها را در آرایه M_U قرار بده.
- (۵) متغیر opt را برابر با مقدار ∞ قرار بده.
- (۶) $i = 1$.
- (۷) اگر $C'_\rho(e_i) < opt$ آنگاه به گام (۸) برو در غیر این صورت به گام (۱۰) برو.
- (۸) مقدار $C'_\rho(e_i)$ را در متغیر opt قرار بده.
- (۹) مقدار متغیر i را در i^* ذخیره کن.
- (۱۰) $i = i + 1$.
- (۱۱) اگر $i < n$ به گام (۷) برو در غیر این صورت به گام (۱۲) برو.
- (۱۲) اگر $C'_\rho(e_{i^*}) > C'_\rho(m, m)$ آنگاه گام (۱۳) را اجرا کن در غیر این صورت به گام (۱۴) برو.
- (۱۳) توقف کنید، (m, m) ۲- میانه پشتیبان درخت ورودی T است.
- (۱۴) توقف کنید، $(M_L[i^*], M_U[i^*])$ ۲- میانه پشتیبان درخت ورودی T است.
- جزئیات محاسبه تابع هزینه وابسته به یال، میانه پایین و میانه بالا در مرجع [۱۳۸] آمده است.
- لم ۲.۱.۳. [۱۳۸] هزینه وابسته به یال یعنی C'_ρ برای تمام زوج میانه‌ها را می‌توان در زمان خطی محاسبه نمود.

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

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

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

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

(۲) مجموعه رأس‌ها را ترتیب پیمایش پس ترتیب که در فصل اول توضیح داده شده است، مرتب نموده و در (v_1, v_2, \dots, v_n) قرار دهید.

(۳) $i = 1$.

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

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

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

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

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

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

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

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

(۱۲) متغیر g را در $L[i]$ ذخیره کن.

(۱۳) $i = i + 1$.

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

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

در زیربرنامه $LOWER - MEDIAN$ ، همیشه میانه پایین در سنگین‌ترین زیردرخت نزدیک به ریشه جستجو می‌شود و با پیدا کردن آن با استفاده از قضیه ۱.۱.۳ جستجو پایان می‌یابد.

شبه‌کد زیربرنامه $UPPER - MEDIAN(T^*)$

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

خروجی: آرایه U که حاوی میانه‌های بالا به ازای تمام یال‌های T^* است.

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

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

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

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

- (۵) میانه زیردرخت T_{v_r} را در m_{v_r} ذخیره کن.
- (۶) مسیر از ریشه به m_{v_1} را در $Path_1$ ذخیره کن.
- (۷) مسیر از ریشه به m_{v_r} را نیز در $Path_r$ ذخیره کن.
- (۸) یال‌ها را به دلخواه مرتب نموده و در $(e_1, e_2, \dots, e_{n-1})$ قرار بده.
- (۹) $i = 1$.
- (۱۰) اگر $e_i \in T_{v_1}$ آنگاه به گام (۱۱) برو در غیراین صورت به گام (۱۲) برو.
- (۱۱) پایین‌ترین رأس v در مسیر $Path_r$ با $w(T_v) \geq \frac{w(X(e_i))}{r}$ را در $U[i]$ ذخیره کن.
- (۱۲) پایین‌ترین رأس v در مسیر $Path_1$ با $w(T_v) \geq \frac{w(X(e_i))}{r}$ را در $U[i]$ ذخیره کن.
- (۱۳) $i = i + 1$.
- (۱۴) اگر $i < n$ به گام (۱۰) برو در غیراین صورت به گام (۱۵) برو.
- (۱۵) آرایه U را برگردان.

قضیه ۲.۱.۳. [۱۳۸] الگوریتم $BACKUP_2MEDIAN$ ، ۲-میانه پشتیبان درخت T را در زمان $O(n \log n)$ محاسبه می‌کند.
برپایه لم فوق قضیه زیر نتیجه می‌شود:

قضیه ۳.۱.۳. [۱۳۸] احتمال‌های شکست ρ_1 و ρ_2 داده شده است. یک ۲-میانه پشتیبان برای درخت با رئوس وزن دار می‌تواند در زمان $O(n \log n)$ محاسبه شود.

۲.۳ مسائل مکان‌یابی معکوس

در این بخش، ابتدا یک بیان کلی از مسئله مکان‌یابی معکوس ارائه می‌شود و سپس انواع مسائل مکان‌یابی معکوس معرفی شده و خلاصه‌ای از کارهای مهم صورت گرفته مربوط به هر نوع معرفی می‌شوند.

در یک مدل مکان‌یابی معکوس برخی از پارامترهای ورودی سیستم (طول‌های یالی یا وزن‌های راسی گراف، وزن‌ها یا مختصات نقاط روی فضای حقیقی) با ارزان‌ترین (یا پرسودترین) روش ممکن تغییر داده می‌شوند تا اینکه موقعیت مکانی تسهیلات از قبل معین شده تحت پارامترهای جدید بهینه شده یا تا حد امکان بهبود یابند. در مدل‌های ترفیع / تنزیل، پارامترهای ورودی سیستم قبل از اقدام به مکان‌یابی تسهیلات، با توجه

به بودجه داده شده به شکل بهینه تغییر داده می‌شوند و سپس محل استقرار تاسیسات تحت پارامترهای جدید مکان‌یابی می‌شوند.

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

$$d : X \times X \rightarrow \mathbb{R}^+$$

باشد. این فضای متریک می‌تواند یک شبکه و یا فضای حقیقی $\mathbb{R}^k, k \geq 1$ باشد. همچنین، فرض کنید (LOC) یک مسئله مکان‌یابی روی این فضای متریک بوده و $\theta \in \mathbb{R}^n$ بردار پارامترهای مسئله (LOC) باشد. به‌عنوان مثال θ می‌تواند شامل وزن‌های راسی و یا طول‌های یالی روی شبکه‌ها باشد و یا وزن نقاط و یا مختصات نقطه‌ای روی فضای حقیقی $\mathbb{R}^k, k \geq 1$ باشد. به علاوه، فرض کنید S_θ مجموعه کلیه جواب‌های شدنی مسئله (LOC) تحت بردار پارامتر θ باشد و $f(\theta, S)$ تابع هدف مسئله (LOC) برای جواب شدنی $S \in S_\theta$ باشد. با این مفروضات می‌توان مسئله (LOC) را به صورت زیر نوشت.

$$\begin{aligned} \text{(LOC)} \quad & \min f(\theta, S) \\ \text{s.t.} \quad & S \in S_\theta. \end{aligned} \quad (۵.۳)$$

به علاوه فرض کنید $S^* \in S_\theta$ جوابی مفروض از مسئله مکان‌یابی (LOC) باشد. هدف مسائل مکان‌یابی معکوس عبارت است از اصلاح بردار پارامتر θ به $\tilde{\theta}$ با توجه به برخی محدودیت‌ها به ارزان‌ترین روش ممکن (پرسودترین روش ممکن)، به‌طوری 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 ترفیع (تنزیل) متناظر مسئله مکان‌یابی کلاسیک (LOC)، تغییر بردار پارامتر θ به $\tilde{\theta}$ است، به‌طوری که مقدار هدف مسئله (LOC) تحت بردار پارامتر جدید $\tilde{\theta}$ و با توجه به محدودیت بودجه تا حد ممکن بهتر (بدتر) شود. فرض کنید c_i^+ هزینه تحمیل شده p_i^+ سود حاصله) از افزایش i امین مولفه θ_i از بردار θ به اندازه یک واحد بوده و c_i^- هزینه تحمیل شده p_i^- سود حاصله) از کاهش θ_i 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 باشد. از طرفی، شاید نتوانیم که بردار θ را به‌طور دلخواه اصلاح کنیم. از این‌رو فرض کنید B مجموعه محدودیت‌هایی باشد که بردار پارامتر جدید $\tilde{\theta}$ باید در آن‌ها صدق کند. تابع هزینه $g(c, \tilde{\theta})$ تابع $g(p, \tilde{\theta})$ (سود) در حالت‌های مختلفی بررسی می‌شود. با در نظر گرفتن پارامتر $\tau = c$ برای تابع هزینه و $\tau = p$ برای تابع سود، برخی از حالت‌های این توابع مسائل مکان‌یابی معکوس بیشتر مورد توجه قرار گرفته‌اند عبارتند از :

• تابع هزینه (تابع سود) تحت نرم خطی:

$$g(\tau, \tilde{\theta}) = \sum_{i=1}^n \left(\tau_i^+ \max \{0, \tilde{\theta}_i - \theta_i\} + \tau_i^- \max \{0, \theta_i - \tilde{\theta}_i\} \right). \quad (۶.۳)$$

• تابع هزینه (تابع سود) تحت فاصله همینگ نوع جمعی:

$$g(\tau, \tilde{\theta}) = \sum_{i=1}^n \left(\tau_i^+ H(\tilde{\theta}_i, \theta_i) + \tau_i^- H(\theta_i, \tilde{\theta}_i) \right). \quad (۷.۳)$$

به طوری که فاصله همینگ $H(\tilde{\theta}, \theta)$ به صورت

$$H(\tilde{\theta}, \theta) = \begin{cases} 1 & \theta \neq \tilde{\theta}, \\ 0 & \theta = \tilde{\theta}, \end{cases}$$

تعریف می‌شود. تابع هزینه (تابع سود) تحت فاصله همینگ نوع تنگنا:

تابع هزینه (تابع سود) تحت نرم چبیشف:

$$g(\tau, \tilde{\theta}) = \max_{i=1, \dots, n} \max \left\{ \tau_i^+ \max \{0, \tilde{\theta}_i - \theta_i\}, \tau_i^- \max \{0, \theta_i - \tilde{\theta}_i\} \right\}. \quad (۸.۳)$$

• تابع هزینه (تابع سود) تحت نرم ترکیبی نوع تنگنا:

$$g(\tau, \tilde{\theta}) = \max_{i=1, \dots, n} \max \left\{ \tau_i^+ H(\tilde{\theta}_i, \theta_i) + \tau_i^+ \max \{0, \tilde{\theta}_i - \theta_i\}, \right. \\ \left. \tau_i^- H(\theta_i, \tilde{\theta}_i) + \tau_i^- \max \{0, \theta_i - \tilde{\theta}_i\} \right\}. \quad (۹.۳)$$

همچنین با توجه به هدف تصمیم‌گیرنده می‌توان توابع هزینه دیگری را نیز تعریف کرد. مجموعه B را به صورت زیر در نظر بگیرید:

$$B = \left\{ \tilde{\theta} : u^- \leq \tilde{\theta} \leq u^+, \varphi(\tilde{\theta}, \theta) \leq \beta, u^+, u^- \in \mathbb{R}^n \right\}.$$

که در آن بردار پارامتر اصلاح شده باید در قید کران بالا u^+ و کران پایین u^- صدق کند. همچنین قید $\varphi(\tilde{\theta}, \theta) \leq \beta$ ، به عنوان یک محدودیت اضافی برای اعمال نظرات تصمیم‌گیرنده در نظر گرفته شده است. مسئله مکان‌یابی معکوس دارای کاربردهای فراوانی می‌باشد که در اینجا به یک مورد اشاره می‌شود. ایستگاه قطار یک شهر را در نظر بگیرید که با گذشت زمان مکان آن از بهینگی خارج شده است و امکان انتقال آن به نقاط دیگر وجود ندارد. شهردار آن شهر می‌تواند برخی از پارامترهای سیستم شهری مانند خیابان‌ها یا خطوط حمل و نقل شهری و یا آرایش جمعیتی شهر را با کمترین هزینه تغییر دهد تا اینکه مکان کنونی ایستگاه قطار تبدیل به یک مکان ۱- میانه شبکه متناظر با آن شهر شود.

برخی از متداول‌ترین مدل‌های مکان‌یابی اصلاحی در نشریات پژوهشی معرفی شده‌اند که عبارتند از:

• مسائل مکان‌یابی معکوس حداقل هزینه / حداکثر سود^۶

^۶(Inverse location problems)

• مسائل مکان‌یابی معکوس بودجه مقید^۷

• مسائل مکان‌یابی ترفیع و تنزیل^۸

در ادامه به تشریح مختصر این مدل‌ها به همراه تاریخچه‌ای از کارهای صورت گرفته و برخی رویکردهای جواب این مدل‌ها پرداخته می‌شود.

۱.۲.۳ مدل مکان‌یابی معکوس حداقل هزینه/حداکثر سود

برای مسئله مکان‌یابی مفروض (LOC) در فضای متری (X, d) با بردار پارامتر θ ، مجموعه جواب‌های شدنی S_θ ، تابع هدف $f(\theta, S)$ و یک جواب شدنی مفروض $S^* \in S_\theta$ ، مسئله مکان‌یابی معکوس حداقل هزینه (حداکثر سود) متناظر به صورت زیر فرمول‌بندی می‌شود:

$$\begin{aligned} (\max) \min \quad & g(c, \tilde{\theta}) \\ \text{s.t.} \quad & f(\tilde{\theta}, S^*) \leq f(\tilde{\theta}, S) \quad \forall S \in S_{\tilde{\theta}}, \\ & \tilde{\theta} \in B, \end{aligned} \quad (10.3)$$

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

مروری بر مسائل مکان‌یابی میانه معکوس حداقل هزینه/حداکثر سود

در سال ۲۰۰۴ برکارد و همکاران مسئله ۱- میانه معکوس حداقل هزینه با تغییرات وزن مربوط به نقاط را در مرجع [۳۰] مورد بررسی قرار دادند. آن‌ها ابتدا مسئله p -میانه معکوس حداقل هزینه گسسته را در فضای متری (X, d) به صورت یک مدل برنامه‌ریزی خطی فرمول‌بندی کردند و با توجه به متناهی بودن تعداد قیود و متغیرهای این مدل گزاره زیر را نتیجه گرفتند. مسئله p -میانه معکوس حداقل هزینه گسسته، که در آن مشتریان ممکن است دارای هر وزن حقیقی باشند، در زمان چندجمله‌ای قابل حل است، با این شرط که مقدار p ثابت باشد و به‌عنوان یک پارامتر ورودی نباشد. توجه داشته باشید که بر اساس گزاره فوق، مسئله p -میانه ناخوشایند معکوس حداقل هزینه گسسته نیز در زمان چندجمله‌ای قابل حل است. آن‌ها در ادامه، مسئله ۱- میانه معکوس حداقل هزینه گسسته را روی درخت‌ها و صفحه مورد بررسی قرار داده و

⁷(Reverse location problems)

⁸(Up- and downgrading location problems)

الگوریتم‌هایی با زمان اجرای $O(n \log n)$ برای آن‌ها ارائه دادند. در سال ۲۰۰۸، برکارد و قلوبی^۹ مسئله فرما-وبر معکوس حداقل هزینه با تغییرات وزن راسی را در زمان $O(n \log n)$ حل نمودند [۲۵]. باروقی و همکاران مقاله‌ای در زمینه مسئله ۱- میانه معکوس حداقل هزینه با تغییر مختصات نقطه‌ای روی \mathbb{R}^d ارائه دادند که در آن ثابت نمودند در صورتی که فواصل بین نقاط با نرم خطی اندازه‌گیری شود، این مسئله \mathcal{NP} -سخت است. آن‌ها همچنین ثابت کردند که با فرض این که وزن نقاط یکسان باشند، مسئله ۱- میانه معکوس حداقل هزینه به d مسئله کوله‌پشتی کاهش یافته و در نتیجه در زمان $O(nd)$ قابل حل است. در ضمن ایشان ثابت نمودند که این مسئله تحت نرم چیبیشف در صفحه \mathcal{NP} -سخت بوده و با نرم اقلیدسی در زمان $O(nd)$ قابل حل است [۱۲].

برای مسئله ۱- میانه معکوس حداقل هزینه با تغییرات وزن راسی روی دور گراف‌ها، برکارد و همکاران [۳۱] نشان دادند که این مسئله را می‌توان به صورت یک مسئله برنامه‌ریزی خطی با متغیرهای کراندار فرمول‌بندی کرد که در این فرمول‌بندی ماتریس قیدها دارای ساختاری خاص است: ستون‌های این ماتریس را می‌توان به دو کلاس افراز کرد که در هر کلاس مولفه‌های ماتریس در هر سطر کاهش می‌یابند. آن‌ها از این خاصیت جالب بهره برده و مسئله را در زمان $O(n^2)$ حل نمودند. در سال ۲۰۰۸، قلوبی مسئله ۱- میانه معکوس حداقل هزینه با تغییرات وزن راسی روی درخت را که قبلاً توسط برکارد و همکاران برای آن الگوریتم $O(n \log n)$ ارائه شده بود، به صورت یک مدل مدل کوله‌پشتی پیوسته فرمول‌بندی کرده و از این رو نشان داد که این مسئله در زمان خطی قابل حل است [۶۲]. باروقی و همکاران [۱۳] نشان دادند که مسئله p - میانه معکوس حداقل هزینه با تغییرات طول یالی روی شبکه‌های کلی \mathcal{NP} -سخت است. با این وجود آن‌ها با تبدیل مسئله ۲- میانه معکوس حداقل هزینه روی درخت‌ها به دنباله متناهی از مدل‌های برنامه‌ریزی خطی نشان دادند که این مسئله در زمان چند جمله‌ای قابل حل است. علاوه بر این، الگوریتمی خطی برای ۲- میانه معکوس حداقل هزینه روی گراف‌های ستاره ارائه دادند. همان‌طور که قبلاً نیز اشاره شد، نتیجه ۳.۳.۲ نشان می‌دهد که 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. سپاسیان^{۱۰} و رهبرنیا^{۱۱} از این خاصیت جالب استفاده کرده و مسئله ۱- میانه معکوس حداقل هزینه روی درخت با تغییرات همزمان طول یال‌ها و وزن رئوس را در

^۹Galavii^{۱۰}Sepasian^{۱۱}Rahbarnia

زمان $O(n \log n)$ حل نمودند [۱۲۷]. گوان^{۱۲} و ژنگ^{۱۳} مسئله ۱- میانه معکوس حداقل هزینه با تغییرات وزن راسی روی درخت را تحت نرم چبیشف و فاصله همینگ مورد بررسی قرار دادند [۷۳]. آن‌ها نشان دادند که این مسئله تحت فاصله همینگ نوع جمعی معادل با یک مسئله کوله پشتی صفر و یک بوده و در نتیجه NP -سخت است. همچنین آن‌ها برای مسئله مذکور تحت نرم چبیشف الگوریتمی با زمان اجرای خطی و تحت فاصله همینگ- تنگنای وزن دار دو الگوریتم مختلف با زمان‌های $O(n \log n)$ و $O(n)$ ارائه دادند. در سال ۲۰۱۳، لانکشو^{۱۴} و همکاران مسئله ۱- میانه معکوس حداقل هزینه با تغییرات وزن یالی حقیقی را بررسی کردند. آن‌ها الگوریتمی خطی برای مسئله با اصلاحات وزن یالی نامنفی ارائه کردند. همچنین آن‌ها نشان دادند که مسئله ۱- میانه معکوس حداقل هزینه تحت فاصله همینگ در حالتی که وزن‌های یالی بهبود یافته درخت مجاز به اختیار مقادیر منفی باشند از کلاس مسائل NP -سخت است و الگوریتم‌هایی با زمان اجرای چندجمله‌ای برای مسئله تحت نرم‌های خطی و چبیشف پیشنهاد کردند [۹۸]. در سال ۲۰۱۶، نگوین^{۱۵} یک الگوریتم ترکیبیاتی با زمان $O(n \log n)$ برای مسئله ۱- میانه معکوس حداقل هزینه با تغییرات وزن‌های راسی روی گراف‌های بلوکی ارائه کرد که در آن فرض بر این است که طول یال‌ها در هر بلوک یکسان هستند [۱۱۰]. نگوین و چی^{۱۶} مسئله ۱- میانه معکوس با هزینه‌های غیر قطعی روی شبکه‌های درختی را مورد بررسی قرار دادند و الگوریتمی با زمان $O(n^2 \log n)$ برای بدست آوردن تابع توزیع معکوس جواب بهینه در هر سطح اطمینان $\alpha \in [0, 1]$ ارائه کردند. اخیراً علی‌زاده^{۱۷} و بخته^{۱۸} یک الگوریتم فراابتکاری گرم شب تاب تعمیم یافته برای تقریب جواب بهینه مسئله p - میانه معکوس حداقل هزینه با تغییرات وزن و مختصات نقاط در فضای k - بعدی حقیقی تحت نرم‌های مختلف ارائه کردند [۲].

در زمینه مسائل مکان‌یابی میانه ناخوشایند معکوس حداقل هزینه در سال ۲۰۰۷، گسنر^{۱۹} مسئله ۱- میانه ناخوشایند معکوس حداقل هزینه با تغییرات طول یالی را مورد بررسی قرار داد و ثابت نمود که این مسئله روی گراف‌های کلی NP -سخت است [۶۷]. وی در ادامه این مسئله را روی درخت‌ها در نظر گرفته و از طریق تبدیل مسئله به یک مسئله گردش با کمترین هزینه توانست روش حلی برای مسئله مورد نظر با زمان اجرای $O(n \log n)$ ارائه کند. قلوی در ادامه کار گسنر مسئله ۱- میانه ناخوشایند معکوس حداقل هزینه را با تغییرات وزن راسی روی یک مسیر در نظر گرفته و برای

¹²Guan¹³Zhang¹⁴Longshu¹⁵Nguyen¹⁶Chi¹⁷Alizadeh¹⁸Bakhteh¹⁹Gassner

آن الگوریتمی خطی ارائه داد. همچنین ایشان مسئله ۱- میانه ناخوشایند معکوس حداقل هزینه با تغییرات وزن راسی را روی دورگراف بررسی کرده و آن را به صورت یک مسئله برنامه‌ریزی خطی مدل‌بندی کرد [۶۲]. اخیراً نگوین [۱۱۴] مسئله p - میانه ناخوشایند معکوس حداقل هزینه با تغییرات طول یالی ($p \geq 2$) روی درخت را به صورت دنباله‌ای از مسائل مکان‌یابی ۲- میانه ناخوشایند معکوس حداقل هزینه بیان کرده و برای مسئله مکان‌یابی ۲- میانه ناخوشایند معکوس حداقل هزینه تحت نرم چبیشف و فاصله همینگ تنگنا الگوریتم‌هایی با زمان اجرای $O(n \log n)$ ارائه نمود. همچنین برای مسئله مکان‌یابی ۲- میانه ناخوشایند معکوس حداقل هزینه تحت نرم خطی روی گراف ستاره الگوریتمی با زمان اجرای خطی ارائه کرد.

مروری بر مسائل مکان‌یابی مرکز معکوس حداقل هزینه

اولین کاری که در زمینه مسائل مکان‌یابی مرکز معکوس حداقل هزینه صورت گرفت، در سال ۱۹۹۹ بود که در آن کای^{۲۰} و همکارانش ثابت کردند که مسئله ۱- مرکز معکوس حداقل هزینه با تغییرات طول یالی روی گراف‌های جهت‌دار بدون وزن NP -سخت است [۳۳]، حال آن که مسئله مکان‌یابی مرکز تحت مطالعه در زمان چندجمله‌ای قابل حل است. با توجه به این نکته، دیگر محققان در سال‌های بعد این مسئله را روی گراف‌های خاصی مانند درخت مورد بررسی قرار دادند. همان‌طور که در بخش مربوط به مسائل مکان‌یابی کلاسیک نیز اشاره شد، حکیمی ثابت نمود که خاصیت بهینگی راسی برای مسائل مکان‌یابی مرکز برقرار نمی‌باشد. لذا در بررسی مسئله مرکز معکوس حداقل هزینه باید دو حالت مرکز راسی معکوس و مرکز مطلق معکوس به‌طور جداگانه در نظر گرفته شوند. یانگ^{۲۱} و ژانگ در سال ۲۰۰۸، مسئله مرکز "راسی" معکوس حداقل هزینه با تغییرات طول یالی را روی درخت‌های بدون وزن در نظر گرفته [۱۴۵] و برای حالتی که طول کلیه 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 اشاره کردند که در حالتی که تغییر توپولوژی درخت نیز مجاز باشد، مسئله در زمان $O(n^3 \log n)$ قابل حل است. در ادامه کار این دو مولف، علی‌زاده و برکارد [۵] هر دو حالت راسی و مطلق مسئله مرکز معکوس حداقل هزینه را مورد بررسی قرار داده و زمان اجرای ارائه شده توسط یانگ و ژانگ را به $O(n^2)$ در حالت بدون تغییر توپولوژی و به $O(n^3)$ را در حالت دیگر بهبود دادند. علی‌زاده و برکارد در تحقیق بعدی [۶] مسئله مرکز معکوس حداقل هزینه را با ضرایب هزینه یکسان در نظر گرفته و هر دو حالت راسی و مطلق مسئله را در زمان $O(n \log n)$ حل کردند. قابل ذکر است که مسئله مرکز معکوس فقط در حالت افزایش طول یالی با هزینه‌های دلخواه

²⁰Cai²¹Yang

قبلاً توسط علی‌زاده و همکارانش در [۳] مورد بررسی قرار گرفته و برای آن روش حل $O(n \log n)$ ارائه شده بود. علی‌زاده و برکارد در سال ۲۰۱۲ مسئله مرکز “ناخوشایند” معکوس حداقل هزینه را با تغییرات طول یالی روی گراف‌های بدون وزن مورد بررسی قرار دادند [۴]. این کار اولین مطالعه انجام شده در زمینه مسائل مکان‌یابی مرکز ناخوشایند معکوس حداقل هزینه است. اخیراً نگوین و چسین^{۲۲} NP – سخت بودن مسئله ۱ – مرکز معکوس حداقل هزینه روی گراف‌های کاکتوس را اثبات کردند [۱۱۲]. علاوه بر این نگوین و آنه^{۲۳} یک الگوریتم با زمان $O(n^2)$ برای مسئله ۱ – مرکز روی درخت با تغییرات وزن راسی ارائه کردند [۱۱۱]. با توجه به ساختار خاص تابع هدف نگوین و سپاسیان برای مسئله ۱ – مرکز معکوس حداقل هزینه روی درخت‌ها تحت نرم چبیشف و فاصله همینگ با تغییرات توپولوژی الگوریتم‌هایی از زمان $O(n \log n)$ و بدون تغییرات توپولوژی الگوریتم‌هایی از زمان $O(n^2)$ ارائه کردند [۱۱۳]. در سال ۲۰۱۷، علی‌زاده و اعتماد^{۲۴} [۷] الگوریتم‌های ترکیبیاتی بهینه برای مسئله مرکز ناخوشایند راسی معکوس حداقل هزینه روی گراف‌های کلی ارائه کردند. به‌طور دقیق‌تر، آن‌ها نشان دادند که تحت افزایش طول‌های یالی مسئله مورد مطالعه در زمان خطی قابل حل است و برای حالت کاهش در طول یال‌ها الگوریتمی با زمان اجرای $O(n^3)$ ارائه دادند. اخیراً نگوین برای مسئله ۱ – مرکز راسی معکوس حداقل هزینه روی دور گراف‌ها الگوریتمی با زمان $O(n^2 \log n)$ پیشنهاد داده است [۱۰۹].

۲.۲.۳ مدل مکان‌یابی معکوس بودجه مقید

در این مسئله نیز فرض می‌شود $S^* \in S_\theta$ یک جواب شدنی مفروض مسئله مکان‌یابی (LOC) در فضای متری (X, d) تحت بردار پارامتر θ باشد. همچنین فرض می‌شود $B \in \mathbb{R}^+$ بودجه در نظر گرفته شده برای اصلاح پارامترهای θ بوده و B مجموعه محدودیت‌هایی باشد که پارامترهای اصلاح شده باید در آن‌ها صدق کنند. با فرض این که $f(\theta, S^*)$ تابع هدف مسئله مکان‌یابی کلاسیک (LOC) به‌ازای جواب شدنی S^* بوده و $g(c, \tilde{\theta})$ تابع هزینه مربوط به اصلاح پارامترهای θ به $\tilde{\theta}$ باشد، یک مسئله مکان‌یابی معکوس بودجه مقید به صورت زیر فرمول‌بندی می‌شود:

$$\begin{aligned} \min \quad & f(\tilde{\theta}, S^*) \\ \text{s.t.} \quad & g(c, \tilde{\theta}) \leq B, \\ & \tilde{\theta} \in B. \end{aligned} \tag{۱۱.۳}$$

²²Chassein

²³Anh

²⁴Etemad

توجه کنید که در این نوع مسئله معکوس هدف این نیست که یک جواب مفروض را تبدیل به جواب بهینه کنیم، بلکه می‌خواهیم مقدار تابع هدف مسئله کلاسیک را با توجه به محدودیت بودجه تا حد ممکن بهبود دهیم.

مروری بر مسائل مکان‌یابی میانه معکوس بودجه مقید

برمن و همکارانش برای اولین بار در سال ۱۹۹۲ مسئله ۱- میانه معکوس بودجه مقید روی یک درخت را مورد بررسی قرار داده و برای آن یک الگوریتم خطی ارائه دادند [۱۹]. در سال ۲۰۰۶، برکارد و همکارانش [۲۶] ثابت کردند که مسئله ۱- میانه معکوس بودجه مقید با تغییرات طول یالی روی گراف‌های کلی \mathcal{NP} -سخت است و برای دور گراف‌ها الگوریتم خطی ارائه دادند. مسئله ۲- میانه معکوس بودجه مقید روی درخت‌ها توسط برکارد و همکارانش در [۲۷] مورد بررسی قرار گرفت و برای آن یک الگوریتم با زمان $\mathcal{O}(n \log n)$ ارائه شد. در سال ۲۰۱۰ ونگ^{۲۵} و همکارش مسئله ۲- میانه معکوس بودجه مقید روی دورها را به مسئله ۳- میانه معکوس بودجه مقید روی یک مسیر تبدیل کرده و از این طریق نشان دادند که این مسئله در زمان چندجمله‌ای قابل حل است [۱۳۷]. در سال ۲۰۱۲، یانگ و همکاران [۸۶] مسئله ۱- میانه معکوس بودجه مقید با یک محدودیت اضافه روی شبکه‌های درختی را مورد بررسی قرار داده و نشان دادند که این مدل قابل تقسیم به دو زیر مسئله معادل بوده و این زیر مسئله‌ها را می‌توان به ترتیب توسط یک الگوریتم برش مینیمم و الگوریتم حریصانه حل نمود.

مروری بر مسائل مکان‌یابی مرکز معکوس بودجه مقید

اولین بار برمن^{۲۶} در سال ۱۹۹۴ ثابت کرد که مسئله مرکز معکوس بودجه مقید با تغییرات طول‌های یالی روی گراف‌های کلی \mathcal{NP} -سخت است [۱۷]. ژانگ و همکارانش [۱۴۷] مسئله مرکز معکوس بودجه مقید را روی درخت‌ها مورد بررسی قرار داده و الگوریتمی با زمان $\mathcal{O}(n^2 \log n)$ برای آن ارائه کردند. در سال ۲۰۱۵، نگوین مسئله ۱- مرکز معکوس با محدودیت بودجه روی درخت‌های وزن‌دار را مورد بررسی قرار داد و برای این مسئله تحت نرم خطی و با هزینه‌های یکسان الگوریتمی از زمان $\mathcal{O}(n^2)$ ارائه کرد [۱۰۸]. علی‌زاده و اعتماد [۸] مسئله ۱- مرکز ناخوشایند بودجه مقید را روی گراف‌های کلی مورد ارزیابی قرار دادند و برای این مسئله تحت نرم‌های مختلف رویکردهای جواب با زمان خطی ارائه نمودند. این نویسندگان در سال ۲۰۱۷ مسئله مکان‌یابی مرکز انتخابی معکوس نوع محدودیت بودجه روی درخت را مورد بررسی قرار دادند و برای این مسئله تحت نرم چبیشف و فاصله همینگ الگوریتم‌هایی با پیچیدگی زمانی چندجمله‌ای ارائه کردند [۵۱]. در مرجع [۵۲] مسئله مرکز ناخوشایند انتخابی

²⁵Wang

²⁶Berman

معکوس بودجه مقید روی دور گراف‌ها مورد بررسی قرار گرفت و برای این مسئله با تغییرات طول‌های یالی الگوریتم ترکیبیاتی با زمان $O(n \log n)$ ارائه شد. اخیراً، اعتماد و علی‌زاده [۵۳] مسئله مکان‌یابی مرکز ناخوشایند انتخابی معکوس بودجه مقید را روی شبکه‌های درختی بررسی کردند. آن‌ها برای این مسئله با تغییرات پیوسته و صحیح در طول‌های یالی الگوریتم‌هایی با زمان $O(n^2)$ پیشنهاد کردند. بعلاوه آن‌ها نشان دادند که اگر هزینه‌های تغییرات یکسان باشند زمان الگوریتم به $O(n)$ کاهش می‌یابد [۱۰۸].

۳.۲.۳ مدل‌های مکان‌یابی ترفیع و تنزیل

فرض کنید $B \in \mathbb{R}^+$ بودجه موجود برای اصلاح پارامترهای سیستم تحت مطالعه باشد. در یک مسئله ترفیع (تنزیل) متناظر مسئله مکان‌یابی کلاسیک (LOC)، هدف تغییر 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. لذا می‌توان این مسئله را به صورت زیر فرمول‌بندی کرد:

$$\begin{aligned} \min(\max) \quad & f(\tilde{\theta}, S) \\ \text{s.t.} \quad & g(c, \tilde{\theta}) \leq B, \\ & \tilde{\theta} \in B, S \in S_{\tilde{\theta}}, \end{aligned} \quad (۱۲.۳)$$

که در آن B مجموعه محدودیت‌هایی است که بردار اصلاح شده $\tilde{\theta}$ باید در آن‌ها صدق کند و $S_{\tilde{\theta}}$ مجموعه جواب‌های شدنی مسئله (LOC) تحت پارامترهای جدید $\tilde{\theta}$ است. توجه کنید که در این نوع مسائل مکان‌یابی برخلاف مسائل مکان‌یابی معکوس قبلی، هیچ جواب شدنی خاصی از مسئله مکان‌یابی کلاسیک (LOC) برای بهبود مشخص نمی‌شود. این نوع مسائل، ارتباط ضعیفتری با مسائل مکان‌یابی معکوس داشته ولی به دلیل اینکه در حوزه مسائل مکان‌یابی بهبود شبکه قرار می‌گیرند، می‌توان آن‌ها را جزو مسائل مکان‌یابی معکوس محسوب کرد.

مروری بر مسائل مکان‌یابی ترفیع و تنزیل

گسرن مسئله مکان‌یابی ترفیع و تنزیل ۱- میانه با تغییرات وزن راسی روی شبکه را در نظر گرفته و برای مسئله مکان‌یابی ترفیع ۱- میانه یک الگوریتم $O(n^2)$ ارائه نمود [۶۵]. وی همچنین ثابت نمود که مسئله مکان‌یابی تنزیل ۱- میانه در زمان چندجمله‌ای قابل حل می‌باشد. در ضمن برای گراف خاص درخت، با استفاده از یک خاصیت تقعر، الگوریتمی با زمان اجرای $O(n \log n)$ ارائه نمود.

بعلاوه وی در سال ۲۰۰۸ طی مطالعه‌ای دیگر، مسئله مکان‌یابی ترفیع و تنزیل ۱- مرکز با تغییرات وزن رئوس روی شبکه را بررسی کرده [۶۶] و نشان داد که مسئله ترفیع ۱- مرکز با معلوم بودن ماتریس فاصله در زمان $O(n^2)$ قابل حل است. در ضمن وی ثابت نمود که مسئله تنزیل ۱- مرکز روی گراف‌های کلی قویاً NP -سخت بوده ولی روی درخت‌ها در زمان $O(n^2)$ قابل حل است. در سال ۲۰۰۹، وی مسئله تنزیل ۱- میانه روی صفحه را تحت نرم منهتن مورد بررسی قرار داد [۶۸]. در این مقاله مسئله مذکور از دید نظریه بازی‌ها مورد بررسی قرار گرفته و برای آن یک الگوریتم با زمان اجرای $O(n \log^2 n)$ ارائه شده است. پلاستریا^{۲۷} مسئله ترفیع و تنزیل ۱- میانه را تحت نرم اقلیدسی و با فرض عدم قطعیت در وزن نقاط مورد بررسی قرار داد. وی نشان داد که مسئله ترفیع متعلق به خانواده مسائل مکان‌یابی- تخصیص پیوسته است و در مقابل مسئله تنزیل به یک مسئله بهینه‌سازی محدب اما مشتق ناپذیر کاهش می‌یابد. ایشان سپس با بررسی ویژگی‌های ساختاری از جواب‌های بهینه، الگوریتم‌های جواب چندجمله‌ای برای این مسئله ارائه نمود [۱۱۷]. در سال ۲۰۱۵، سپاسیان و رهبرنیا ثابت کردند که مسئله ترفیع p - میانه با تغییرات وزن رأسی روی گراف‌های کلی در زمان چندجمله‌ای قابل حل می‌باشد و برای همین مسئله روی یک مسیر با هزینه‌های یکسان یک الگوریتم خطی ارائه دادند [۱۲۸]. اخیراً سپاسیان [۱۲۶] مسئله ترفیع ۱- مرکز با تغییرات طول‌های یالی روی شبکه‌های درختی را بررسی کردند و نشان دادند که این مسئله در زمان چندجمله‌ای حل‌پذیر است. ایشان نشان دادند که مسئله روی درخت‌های بدون وزن در زمان $O(n^2 \log n^2)$ حل‌پذیر است و برای مسئله با هزینه‌های یکسان یک الگوریتم ترکیبیاتی با زمان $O(n^2 \log n)$ ارائه کردند.

۴.۲.۳ مروری بر برخی از مسائل مکان‌یابی معکوس

در این قسمت به بررسی برخی از مسائل مکان‌یابی معکوس پرکاربرد پرداخته می‌شود.

مسئله p - میانه معکوس

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 مثبت هستند). فرض کنید در فضای متریک X ، نقطه p مشخص 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 به اندازه یک واحد افزایش داده شود، متحمل هزینه نامنفی c_i شده و به‌طور مشابه اگر وزن w_i به اندازه یک واحد کاهش یابد، متحمل هزینه نامنفی d_i شده و

²⁷Plastria

البته باید توجه داشت که این افزایش و کاهش وزن مشتریان نمی‌تواند به‌طور 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. در ادامه مسئله p - میانه معکوس با وزن رئوس متغیر بیان می‌شود.

مسئله p - میانه معکوس با وزن رئوس متغیر

مسئله عبارت است از پیدا کردن وزن‌های جدید w_i^* ، $1 \leq i \leq n$ که در کران‌های داده شده قرار دارند، یعنی: $w_i \leq w_i^* \leq \bar{w}_i$ ، به‌گونه‌ای 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 کمینه شود. در مورد مسائل گسسته (مثلاً گراف‌ها) می‌توان مسئله p - میانه معکوس را به‌وسیله یک برنامه‌ریزی خطی توصیف کرد.

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 است. به علاوه، فرض کنید که مجموعه $V = \{v_1, v_2, \dots, v_m\}$ یک مجموعه متناهی شامل تمامی سرویس‌دهندگان ممکن باشد و $d(u, v)$ فاصله بین مشتری u و سرویس‌دهنده v باشد. حال اگر φ کلاس همه زیرمجموعه‌های $S \subseteq V$ با کاردینال $|S| = p$ 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:

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

حال فرض کنید مقداری که وزن مشتری i - ام یعنی w_i افزایش می‌یابد برابر p_i باشد و به‌طور مشابه مقداری که وزن مشتری i - ام کاهش می‌یابد برابر با q_i باشد آنگاه مسئله

p - میانه معکوس می‌تواند به صورت زیر نوشته شود:

$$\min \sum_{i=1}^n (c_i p_i + d_i q_i) \quad (۱۳.۳)$$

s.t. :

$$\sum_{i=1}^n (w_i + p_i - q_i) a_{iS} \leq 0, \quad \forall S \in \varphi, \quad (۱۴.۳)$$

$$w_i - q_i \geq \underline{w}_i, \quad i = 1, 2, \dots, n, \quad (۱۵.۳)$$

$$w_i + p_i \leq \bar{w}_i, \quad i = 1, 2, \dots, n, \quad (۱۶.۳)$$

$$p_i \geq 0, \quad i = 1, 2, \dots, n, \quad (۱۷.۳)$$

$$q_i \geq 0, \quad i = 1, 2, \dots, n, \quad (۱۸.۳)$$

که در مدل فوق:

$$a_{iS} = \min(d(u_i, s_1), d(u_i, s_2), \dots, d(u_i, s_p)) - \min_{v_k \in S} d(u_i, v_k).$$

در مدل برنامه‌ریزی فوق، عبارت (۱۴.۳) در واقع بیان می‌کند که اگر وزن‌ها اصلاح شوند، یعنی وزن رأس i - ام به $w_i + p_i - q_i$ تغییر پیدا کند، به نسبت این وزن‌های جدید، مجموعه از پیش تعیین شده $\{s_1, \dots, s_p\}$ به عنوان p - میانه مسئله معرفی شوند 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 (۱۴.۳) است. همچنین چون وزن‌ها کران دارند، بعد از اصلاح وزن‌ها باید وزن‌های جدید از این کران‌ها تبعیت کنند، پس داریم:

$$\underline{w}_i \leq w_i + p_i \leq \bar{w}_i, \quad \underline{w}_i \leq w_i - q_i \leq \bar{w}_i.$$

از روابط فوق به راحتی عبارات (۱۵.۳) و (۱۶.۳) 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، برنامه $\mathbb{O}\left(\binom{p}{m}\right)$ محدودیت دارد و بنابراین برای مسائل p - میانه نسبتاً کوچک قابل استفاده می‌باشد. توجه داشته باشید که با استفاده از فرض نامنفی بودن c_i و d_i ، می‌توان همیشه فرض کرد که در یکی از جواب‌های بهینه، یکی از مقادیر p_i و q_i برابر صفر است. با توجه به ملاحظات بالا، مدل فوق می‌تواند برای زمانی که وزن مشتریان منفی و یا

به صورت مثبت و منفی (آمیخته)، باشد به کار گرفته شود و در زمان چندجمله‌ای قابل حل است. به طور خلاصه داریم:

گزاره ۱.۲.۳ [۳۰]. 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 چندجمله‌ای قابل حل می‌باشد.

مسئله p - میانه معکوس با طول یال‌های متغیر

فرض کنید $G = (V, E)$ یک گراف متصل با مجموعه رئوس $V = \{v_1, \dots, v_n\}$ و مجموعه یال‌های E باشد که $|E| = m$ ، و یک ثابت p داده شده باشد (که $p \leq n$ می‌باشد). هر یال $e \in E$ دارای یک طول مثبت l_e است 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 به نسبت طول بردار l باشد. به علاوه فرض کنید که هر رأس $v \in V$ دارای یک وزن رأسی 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 رأس به نزدیک‌ترین سرویس‌دهنده کمینه شود که جواب بهینه آن یک p - میانه نامیده می‌شود. در قضیه ۱.۳.۲ بیان شد که یک جواب بهینه میان مجموعه رئوس قرار دارد، که این خاصیت، خاصیت بهینگی رأسی نامیده می‌شود.

هدف در مسئله p - میانه معکوس با طول یال‌های متغیر در واقع اصلاح طول یال‌ها با کمترین هزینه است که مجموعه رئوس از پیش تعیین شده $\{m_1, \dots, m_p\}$ یک p - میانه مسئله شود. فرض کنید که اگر l_e به اندازه یک واحد افزایش داده شود، متحمل هزینه نامنفی c_e^+ شده و به‌طور مشابه اگر l_e به اندازه یک واحد کاهش یابد، متحمل هزینه نامنفی c_e^- شده است. همچنین فرض کنید x_e^+ و x_e^- به ترتیب برابر با مقداری باشد که طول یال l_e افزایش و 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 بالای u_e^+ و u_e^- تبعیت می‌کنند). حال فرض کنید φ مجموعه همه زیرمجموعه‌های $S \subseteq V$ با کاردینال $|S| = p$ باشد. بنابراین مسئله p - میانه معکوس با طول یال‌های متغیر روی گراف G به صورت زیر بیان می‌شود:

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

• در گراف G به نسبت \tilde{l} ، مجموعه $\{m_1, \dots, m_p\}$ یک p - میانه مسئله شود، یعنی:

$$\sum_{v \in V} w(v) \min_{i=1, \dots, p} d_{\tilde{l}}(v, m_i) \leq \sum_{v \in V} w(v) \min_{v_k \in S} d_{\tilde{l}}(v, v_k), \quad \forall S \in \varphi.$$

• کران محدودیت‌ها برقرار باشد، یعنی برای هر $e \in E$ روابط زیر برقرار باشد:

$$\circ \leq x_e^- \leq u_e^-, \quad \circ \leq x_e^+ \leq u_e^+.$$

• تابع هزینه خطی زیر کمینه شود:

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

این فرمول‌بندی مسئله p - میانه معکوس یک مدل برنامه‌ریزی غیرخطی می‌باشد. هر چند در [۱۳] بیان شده است که اگر گراف یک درخت باشد، می‌توان یک فرمول‌بندی خطی برای آن ارائه داد.

مسئله ۱ - میانه معکوس روی درخت و روش حل

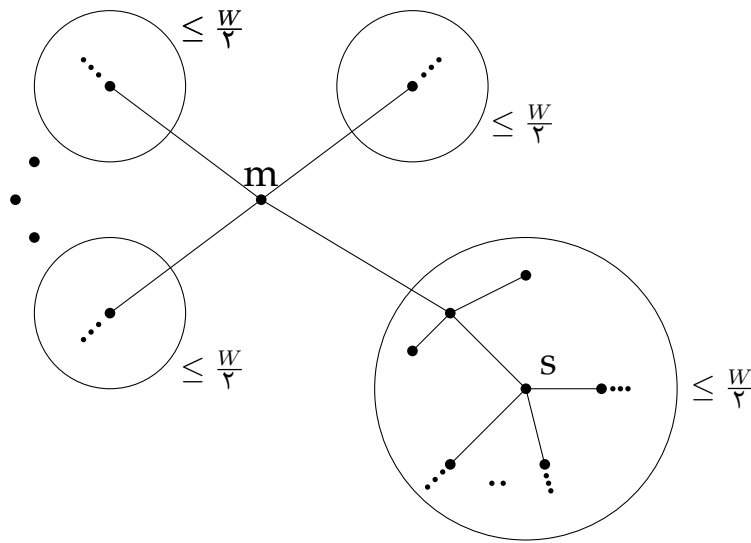
فرض کنید درخت $T = (V, E)$ با مجموعه رئوس V و مجموعه یال‌های E داده شده باشد که مجموعه رئوس (مشتریان) به صورت $V = \{v_1, v_2, \dots, v_n\}$ است، هر رأس v_i دارای وزن نامنفی w_i باشد و همچنین W مجموع وزن‌های رئوس این درخت باشد یعنی:

$$W = \sum_{i=1}^n w_i.$$

در مسئله ۱ - میانه معکوس روی درخت $T = (V, E)$ ، وزن رئوس درخت را باید چنان تغییر داد که یک نقطه از پیش تعیین شده، به‌عنوان ۱ - میانه مسئله واقع شود، که البته برای این تغییر وزن‌ها متحمل هزینه‌هایی می‌شویم. هدف در این مسئله در واقع کمینه کردن این هزینه‌ها است که مدل ریاضی آن در حالت کلی برای مسئله p - میانه معکوس روی گراف در بخش ۴.۲.۳ بیان شده است در ادامه روش حل مسئله ۱ - میانه معکوس، که در [۳۰] آمده است ارائه می‌شود.

فرض کنید رأس m ، یک ۱ - میانه درخت T به نسبت وزن‌های داده شده درخت باشد و رأس s که $s \neq m$ ۱ - میانه درخت به نسبت وزن‌های داده شده نباشد (شکل ۱.۳ را ببینید).

هدف این می‌باشد که رأس s به نسبت وزن‌های اصلاح شده به‌عنوان ۱ - میانه درخت واقع شود. اگر رأس s ، از درجه k باشد، با حذف k یال متصل به رأس s ، زیر درخت‌های T_1, \dots, T_k به دست می‌آید. یکی از این زیردرخت‌های تولید شده، شامل ۱ - میانه مسئله یعنی رأس m است، فرض کنید این زیر درخت، T_k باشد. از طرفی بنا بر قضیه ۱.۳.۲ و همچنین محک بهینگی که در ۳.۳.۲ بیان شد، نتیجه می‌شود که



شکل ۱.۳: درختی که رأس m ، ۱-میانه می‌باشد.

زیر درخت T_k وزنی بیشتر از نصف وزن کل درخت را دارد، پس بقیه زیر درخت‌ها باید وزنی کمتر از نصف وزن کل درخت داشته باشند یعنی:

$$w(T_i) < \frac{W}{3}, \quad \forall i = 1, \dots, k-1.$$

و همچنین:

$$w(T_k) > \frac{W}{3}.$$

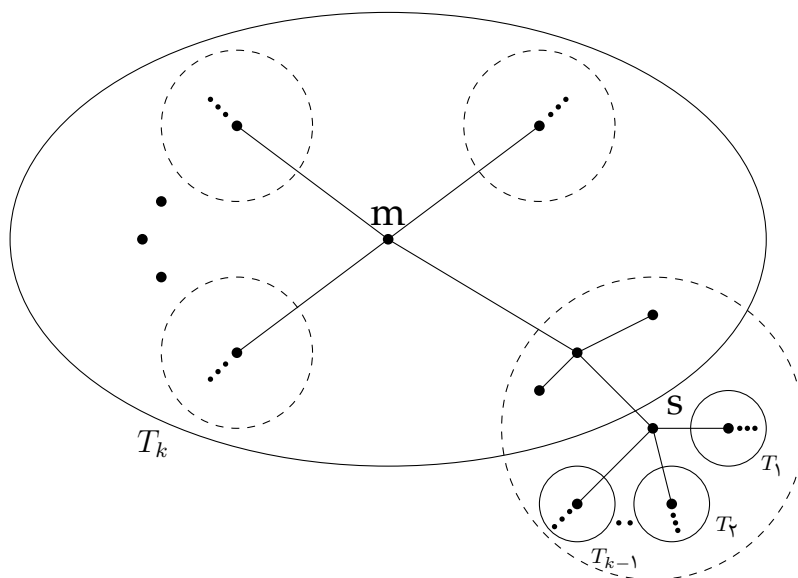
نتایج فوق در واقع از این حقیقت پیروی می‌کند که رأس m ، یک ۱-میانه است و رأس s در یکی از زیر درخت‌های با وزن $\leq \frac{W}{3}$ ، که در رأس m ریشه‌دار است، واقع شده است. حال چون زیر درخت‌های T_1, \dots, T_{k-1} در همان زیر درخت ریشه‌دار شده در m ، با وزن $\leq \frac{W}{3}$ قرار گرفته‌اند، آن‌ها باید وزنی کمتر از نصف وزن کل درخت داشته باشند. فرض کنید W_i وزن فعلی رأس s را مشخص کند و W_i مجموع وزن‌های فعلی رؤس در زیر درخت‌های T_i ؛ $i = 1, \dots, k$ را مشخص کند. در نظر بگیرید:

$$H = \sum_{i=0}^k \frac{W_i}{3}.$$

برای حل مسئله، باید وزن‌های w_i ، $i = 1, \dots, n$ تغییر داده شوند، پس در واقع وزن W_i و همچنین وزن زیر درخت‌های T_i ، یعنی W_i ، $i = 0, \dots, k$ تغییر پیدا می‌کنند. در روند حل مسئله، مقدار شکاف بهینگی^{۲۸} نقش مهمی را بازی می‌کند که به صورت زیر تعریف می‌شود:

$$D = W_k - H.$$

²⁸Optimality Gap



شکل ۲.۳: زیر درخت‌های ریشه دار در رأس s.

لم ۱.۲.۳. [۳۰]. اگر $W_1 \leq \frac{W}{2}, \dots, W_{k-1} \leq \frac{W}{2}$ آنگاه رأس s، ۱-میانه خواهد بود اگر و تنها اگر $D = \circ$.

برهان. می‌دانیم $D = \circ$ اگر و تنها اگر $W_k = \frac{W}{2}$ زیرا:

$$\begin{aligned} D = \circ &\iff W_k - H = \circ \\ &\iff W_k = \left(\frac{W_\circ}{2} + \dots + \frac{W_k}{2} \right) \\ &\iff W_k = \frac{W}{2} \end{aligned}$$

پس بنا بر لم ۳.۳.۲ در می‌یابیم که در این حالت رأس s ۱-میانه درخت خواهد بود.

□

با استفاده از مفروضات مسئله چون رأس ۱-میانه به نسبت وزن‌های موجود (یعنی رأس m) در زیر درخت T_k قرار دارد و همچنین رأس s، یک ۱-میانه درخت نیست، پس زیر درخت T_k وزنی بیشتر از نصف وزن درخت دارد، بنابراین بنا به تعریف شکاف بهینگی، D، مقداری مثبت است، برای اینکه رأس s به‌عنوان ۱-میانه مسئله واقع شود بنا بر لم ۱.۲.۳ باید مقدار D با این تغییر وزن‌های رئوس کم شده تا به مقدار صفر برسد.

اگر به اندازه δ واحد وزن رأس s یا یک رأس در زیر درخت‌های T_1, \dots, T_{k-1} تغییر داده شود، آنگاه مقدار H به اندازه $\frac{\delta}{2}$ تغییر می‌کند و بنا بر تعریف شکاف بهینگی، مقدار

D نیز به اندازه $\frac{\delta}{\rho}$ - تغییر می‌کند. بنابراین نتیجه می‌شود:

$$D_{new} = W_k - (H + \frac{\delta}{\rho}) = W_k - H - \frac{\delta}{\rho} = D - \frac{\delta}{\rho}.$$

همچنین اگر به اندازه δ واحد وزن یک رأس در زیر درخت T_k را تغییر داده شود، این تغییر وزن، مقدار وزن W_k را به اندازه δ و همچنین مقدار H را به اندازه $\frac{\delta}{\rho}$ تغییر می‌دهد، پس بنا بر تعریف شکاف بهینگی، مقدار D به اندازه $\frac{\delta}{\rho}$ تغییر می‌کند، بنابراین نتیجه می‌شود:

$$D_{new} = (W_k + \delta) - (H + \frac{\delta}{\rho}) = D + \delta - \frac{\delta}{\rho} = D + \frac{\delta}{\rho}.$$

حال بنا بر گفته‌های فوق، برای کاهش مقدار D با استفاده از تغییر وزن رئوس، می‌توان عملیات زیر را انجام داد:

- افزایش وزن رأس s ، یا رئوس واقع در زیر درختان T_i ، $i = 1, 2, \dots, k-1$.
- کاهش وزن رئوس واقع در زیر درخت T_k .

از طرف دیگر، هر عملیاتی که D را بیشتر می‌کند، باید به وسیله یک تغییر وزن در رئوس خنثی گردد (مقدار D کم شود) اما این کاهش و افزایش مقدار D که در واقع با استفاده از تغییر وزن رئوس اتفاق می‌افتد هزینه‌هایی را به دنبال دارد، در صورتی که هدف کاهش این هزینه‌ها است و همچنین از طرفی می‌توان مسئله را بدون افزایش مقدار D و خنثی کردن شکاف بهینگی بعد از آن، حل کرد. بنابراین عملیات فقط روی کاهش مقدار D ، با استفاده از موارد ذکر شده در بالا محدود می‌شود.

با توجه به ضرایب هزینه c و d که در ۴.۲.۳ بیان شد و همچنین با توجه به مطالب فوق، برای ریشه درخت (یعنی رأس s) و همچنین رئوس واقع در زیر درخت‌های T_i ، $i = 1, 2, \dots, k-1$ ضرایب هزینه c در نظر گرفته می‌شود. همچنین برای رئوس واقع در زیر درخت T_k ضرایب هزینه d در نظر گرفته می‌شود. با توجه به اینکه درخت T ، تعداد n رأس دارد، اگر φ یک جایگشت از مجموعه $\{1, 2, \dots, n\}$ باشد، که به‌طور صعودی ضرایب هزینه در یک لیست مرتب شده باشد، داریم:

$$r_{\varphi(1)} \leq r_{\varphi(2)} \leq \dots \leq r_{\varphi(n)}.$$

اگر رأس $v_{\varphi(l)}$ ، رأس s و یا یکی از رئوس واقع در زیر درخت‌های T_i ، $i = 1, 2, \dots, k-1$ باشد، آنگاه ضریب هزینه $r_{\varphi(l)}$ در واقع همان $c_{\varphi(l)}$ است و در غیر این صورت یعنی اگر $v_{\varphi(l)}$ متعلق به یکی از رئوس واقع در زیر درخت T_k باشد آنگاه ضریب هزینه $r_{\varphi(l)}$ در واقع همان $d_{\varphi(l)}$ است. حال فرض کنید که $r_{\varphi(1)} = c_1$. در این حالت برای کاهش مقدار شکاف بهینگی یعنی D می‌توان وزن w_1 را به اندازه p_1 افزایش داد. اما چون این افزایش وزن نمی‌تواند به‌صورت دلخواه باشد چون w_1 کراندار است. دو حالت زیر متمایز می‌شود:

حالت اول: این حالت وقتی است که وزن w_1 به اندازه p_1 افزایش می‌یابد ولی در حین این افزایش وزن قبل از اینکه به کران بالای وزن w_1 برسیم مقدار D برابر صفر شده و به جواب بهینه رسیده‌ایم. یعنی در این حالت داریم:

$$D = 0 \implies W_k - (H + \frac{p_1}{\gamma}) = 0 \implies W_k = H + \frac{p_1}{\gamma}.$$

بنابراین مقدار p_1 به صورت زیر می‌باشد:

$$p_1 = 2(W_k - H) = 2(W_k - \sum_{i=0}^k \frac{W_i}{\gamma}) = 2(\frac{1}{\gamma}W_k - \frac{1}{\gamma}\sum_{i=0}^{k-1} W_i) = W_k - \sum_{i=0}^{k-1} W_i.$$

که اگر مقدار فوق به وزن w_1 اضافه شود وزن جدید کمتر از کران بالای w_1 می‌شود. باید توجه کرد که در این حالت مقدار وزن رئوس در زیر درخت T_k بعد از اصلاح وزن‌ها دقیقاً برابر با نصف کل وزن درخت است. از طرفی بنا بر مطالب گفته شده $w(T_i) < \frac{W}{\gamma}; \forall i = 1, \dots, k-1$ است، پس بنا بر لم ۱.۲.۳ جواب فعلی بهینه است.

حالت دوم: این حالت وقتی است که وزن w_1 به اندازه p_1 افزایش داده می‌شود یعنی $w_1^* = w_1 + p_1$ و در این افزایش وزن به کران بالای وزن w_1 رسیده است. پس داریم:

$$p_1 = \bar{w}_1 - w_1.$$

ولی مقدار D هنوز برابر صفر نشده است و داریم:

$$D_{new} = D - \frac{p_1}{\gamma}.$$

بنابراین فعلاً به جواب بهینه نرسیده‌ایم، چون مقدار p_1 نمی‌تواند بیشتر از این افزایش پیدا کند، لذا مقدار p_1 به وزن w_1 اضافه می‌شود و $r_{\varphi(1)}$ از لیست حذف خواهد شد و باید به عنصر بعدی لیست، یعنی $r_{\varphi(2)}$ رفت.

حال صورت دیگر را در نظر می‌گیریم، فرض شود اولین عنصر لیست، d_1 باشد یعنی $r_{\varphi(1)} = d_1$ پس v_1 در زیر درخت T_k قرار دارد و برای کاهش شکاف بهینگی باید وزن این رأس به اندازه q_1 کم شود اما بنا بر کراندار بودن w_1 ، این کاهش وزن نمی‌تواند به صورت دلخواه انجام شود. مشابه مورد قبل دو مورد به صورت متمایز در نظر گرفته می‌شود:

حالت اول: این حالت وقتی است که وزن w_1 ، به اندازه q_1 کاهش می‌یابد ولی در حین این کاهش وزن قبل از اینکه به کران پایین وزن w_1 برسیم مقدار D برابر صفر شده و به جواب بهینه دست یافته‌ایم. یعنی در این حالت داریم:

$$D = 0 \implies (W_k - q_1) - (H - \frac{q_1}{\gamma}) = 0 \implies W_k - H - q_1 + \frac{q_1}{\gamma} = 0.$$

بنابراین مقدار q_1 به صورت زیر است:

$$q_1 = 2(W_k - H) = 2(W_k - \sum_{i=0}^k \frac{W_i}{2}) = 2(\frac{1}{2}W_k - \frac{1}{2} \sum_{i=0}^{k-1} W_i) = W_k - \sum_{i=0}^{k-1} W_i.$$

که اگر مقدار فوق از وزن w_1 کم شود، وزن جدید، بیشتر از کران پایین w_1 می‌باشد. توجه داریم که در این حالت مقدار وزن رئوس، در زیر درخت T_k بعد از اصلاح وزن‌ها دقیقاً برابر با نصف کل وزن درخت می‌باشد. از طرفی بنا بر مطالب گفته شده $w(T_i) < \frac{W}{2}; \forall i = 1, \dots, k-1$ پس بنا بر لم (۱.۲.۳) جواب فعلی بهینه است.

حالت دوم: این حالت وقتی است که وزن w_1 ، به اندازه q_1 کاهش می‌یابد، یعنی $w_1^* = w_1 - q_1$ و در این کاهش وزن به کران پایین وزن w_1 ، رسیده پس نتیجه می‌شود:

$$q_1 = w_1 - \underline{w}_1.$$

ولی مقدار D هنوز برابر صفر نشده است و داریم:

$$D_{new} = D - \frac{q_1}{2}.$$

لذا فعلاً به جواب بهینه نرسیده‌ایم، چون مقدار q_1 نمی‌تواند بیشتر از این کاهش پیدا کند، لذا مقدار q_1 از وزن w_1 کم می‌شود و $r_{\varphi(1)}$ ، از لیست حذف شده و باید به عنصر بعدی لیست، یعنی $r_{\varphi(2)}$ رفت.

در ملاحظات فوق مشاهده شد که می‌توان برای حل مسئله ۱- میانه معکوس روی یک درخت، با یک روش حریصانه پیش رفت، به این صورت که ابتدا لیست ضرایب هزینه به صورت صعودی مرتب شده، سپس اولین عنصر این لیست انتخاب شده و بر طبق مطالب گفته شده، وزن متناظر با آن را افزایش یا کاهش داد. اگر در این افزایش و کاهش با کران بالا یا پایین وزن متناظر مواجه شدیم، این عنصر را از لیست حذف کرده و عنصر بعدی لیست پردازش می‌شود. به محض اینکه به حالت اول در هر یک از دو مورد فوق رسیدیم، توقف می‌کنیم. رأس s به عنوان یک ۱- میانه درخت می‌باشد. البته باید در نظر گرفت که بنا بر مرتب‌سازی ضرایب هزینه، هر توزیع دیگر وزن‌ها که باعث شود رأس s یک ۱- میانه درخت واقع شود، هزینه‌ای بیشتر یا مساوی را خواهد داشت. در نتیجه، بنا بر مطالب گفته شده، برای رسیدن به جواب بهینه به الگوریتم زیر خواهیم رسید:

الگوریتم حل مسئله ۱- میانه معکوس روی درخت:

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

گام ۲: وزن زیر درختان ریشه‌دار شده در رأس s را محاسبه کنید.

- اگر همه زیر درخت‌ها وزنی کمتر یا مساوی $\frac{W}{4}$ داشتند، توقف کنید، وزن‌های داده شده از قبل بهینه بوده‌اند.
 - در غیر این صورت، زیردرخت با بیشترین وزن را T_0 و همچنین وزن این زیردرخت را W_0 بنامید.
- گام ۳:** برای $i = 1, 2, \dots, n$ ، اگر رأس v_i خود رأس s یا رأسی خارج از زیر درخت T_0 است، قرار دهید $r_i = c_i$ ، در غیر این صورت قرار دهید $r_i = d_i$.
- گام ۴:** قرار دهید $D = W_0 - \frac{W}{4}$ و $R = \{r_1, r_2, \dots, r_n\}$.
- گام ۵:** اگر $R \neq \emptyset$ ، آنگاه کوچک‌ترین عنصر R مثلاً r_i را پیدا کنید. در غیر این صورت یک جواب شدنی برای مسئله وجود ندارد.
- گام ۶:** اگر $r_i = c_i$ باشد، آنگاه قرار دهید $p_i = \min(2D, \bar{w}_i - w_i)$ ؛ $D = D - p_i/2$ و در غیر این صورت قرار دهید $q_i = \min(2D, w_i - \underline{w}_i)$ ؛ $D = D - q_i/2$.
- گام ۷:** اگر $D = 0$ ، آنگاه توقف کنید، جواب بهینه به دست آمده است، در غیر این صورت به گام ۵ برگردید.

گزاره ۲.۲.۳. [۳۰]. مسئله ۱- میانه معکوس روی درخت، به وسیله یک الگوریتمی از نوع حریمانه، در زمان $O(n \log n)$ قابل حل است.

۳.۳ مسئله مکان‌یابی آرمانی

در طول دهه‌های اخیر، تلاش‌های بسیار زیادی برای ایجاد مدل‌های مکان‌یابی ای که مشخصه‌های بیشتری از دنیای واقعی را در نظر می‌گیرند، انجام گرفته است. یکی مشخصه‌های پرکاربرد و مهمی که در نظریه‌های اخیر تحقیق در عملیات ظاهر شده است، مفهوم "مکان‌یابی آرمانی" است.

فتحعلی و همکاران [۵۹] برای اولین بار یک مسئله خاص از مسئله مکان‌یابی وبر را مطرح کردند. آن‌ها در این مسئله برای هر نقطه تقاضای p_i ، یک شعاع آرمانی را در نظر گرفتند و حداقل فاصله بین تسهیلات جدید و نقاط تقاضا را برابر با شعاع متناظر با هر نقطه تقاضا یعنی r_i ، تعیین کردند، از آنجایی که در واقعیت به ندرت مکانی برای تسهیلات جدید وجود دارد که فاصله آن تا نقاط تقاضای p_i ، دقیقاً r_i شود، لذا آن‌ها در این مدل به دنبال کمینه کردن مجموع وزنی مربعات خطا بودند. آن‌ها با استفاده از روش مربع بزرگ مربع کوچک الگوریتمی برای حل مسئله تحت نرم اقلیدسی پیشنهاد کردند. سپس جمالیان و فتحعلی [۸۵]، یک مدل برنامه‌ریزی خطی برای مسئله با هدف کمینه کردن مجموع وزنی قدرمطلق خطا ارائه کردند.

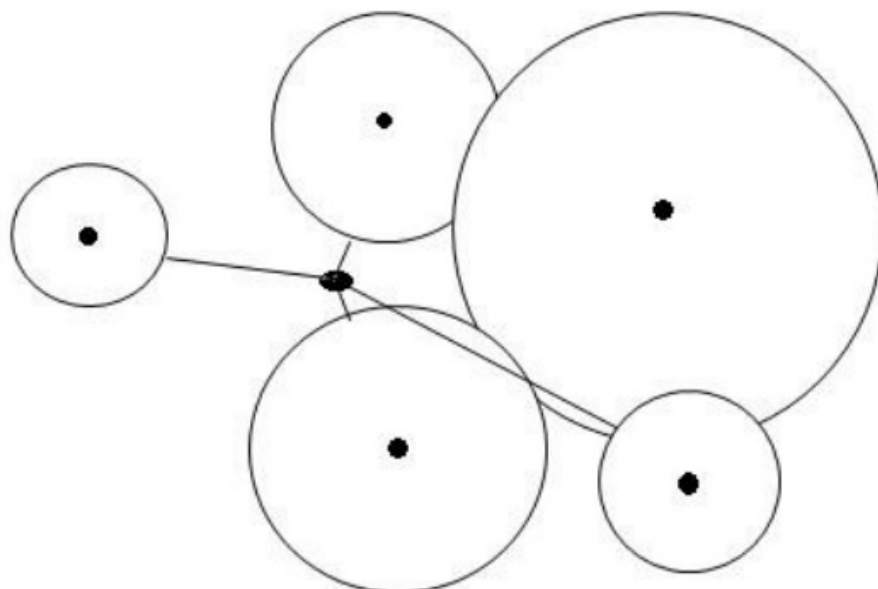
اخیرا فتحعلی و جمالیان [۵۶]، به بررسی مسئله کمینه کردن مجموع مربعات خطا پرداخته و آن را مسئله مکان‌یابی وبر مربعی آرمانی^{۲۹} (GSWLP) نامیدند. آن‌ها با استفاده از الگوریتم شبیه‌سازی پرندگان به حل مسئله فوق تحت نرم اقلیدسی پرداختند.

۱.۳.۳ مسئله مکان‌یابی وبر آرمانی

فرض کنید $p_i, i = 1, \dots, n$ ، نقاط موجود (مناطق / سرویس‌گیرنده‌ها)، $w_i, i = 1, \dots, n$ و $r_i, i = 1, \dots, n$ به ترتیب وزن‌ها و شعاع‌های متناظر با مناطق p_i باشند. همچنین فاصله نقطه مجهول تا منطقه p_i است. در این صورت یک مسئله مکان‌یابی وبر آرمانی به دنبال نقطه‌ای مانند x است به طوری که تابع هدف زیر را کمینه کند [۵۹]:

$$F(x) = \sum_{i=1}^n w_i (d(x, p_i) - r_i)^2. \quad (۱۹.۳)$$

در واقع مدل (۱۹.۳)، میزان خطای دوری و نزدیکی از کران‌های تعیین شده را کمینه می‌کند. شکل ۳.۳ نمایش گرافیکی مسئله مکان‌یابی آرمانی را برای پنج نقطه تقاضا با شعاع‌های آرمانی متفاوت را نشان می‌دهد.



شکل ۳.۳: نمایش گرافیکی مسئله مکان‌یابی آرمانی.

²⁹Goal Square Weber Location Problem,

به‌عنوان مثالی از این مسئله، اگر تمام نقاط روی دایره‌ای به شعاع r قرار داشته باشند و شعاع آرمانی مفروض برای هر سرویس‌دهنده برابر با r باشد، یعنی $\forall i \ r_i = r$ ، آنگاه مرکز دایره فوق جواب بهینه مسئله خواهد بود و مقدار تابع هدف برابر با صفر است.

با توجه به توجیه و تعبیر مسئله مکان‌یابی آرمانی، این مسئله کاربردهای زیادی در استقرار کارخانه‌ها، ماشین‌الات، نیروگاه‌ها، استادیوم‌ها و اماکن دفع زباله دارد. می‌توان گفت خواص زیر برای مسئله مکان‌یابی آرمانی (۱۹.۳)، تحت نرم اقلیدسی برقرار است.

لم ۱.۳.۳. [۵۹]. تابع هدف مسئله (۱۹.۳) محدب نیست.

تعریف ۱.۳.۳. [۵۹]. فرض کنید n نقطه p_1, \dots, p_n در فضای \mathbb{R}^2 موجود باشند و

$p_i = (a_i, b_i)$. تعریف می‌کنیم $RH_1 = (a_{min}, b_{min})$ ، $RH_2 = (a_{min}, b_{max})$ ، $RH_3 = (a_{max}, b_{min})$ و $RH_4 = (a_{max}, b_{max})$ که در آن

$$a_{min} = \min \{a_i - r_i | i = 1, \dots, n\},$$

$$a_{max} = \max \{a_i + r_i | i = 1, \dots, n\},$$

$$b_{min} = \min \{b_i - r_i | i = 1, \dots, n\},$$

$$b_{max} = \max \{b_i + r_i | i = 1, \dots, n\}.$$

قضیه ۱.۳.۳. [۵۹]. جواب بهینه مسئله (۱۹.۳) در پوسته مستطیلی نقاط RH_1 ، RH_2 ، RH_3 و RH_4 قرار دارد.

یک روش هندسی BSSS الگوریتم برای حل مسئله (۱۹.۳) توسط فتحعلی و همکاران در [۵۹] ارائه شده است. همچنین فتحعلی و همکاران در [۵۶] نیز با استفاده از شرط لازم بهینگی، یک روش شبه‌وایزفیلد را به‌صورت زیر برای حل مسئله (۱۹.۳) ارائه کرده‌اند.

$$\frac{\partial F}{\partial x} = 2 \sum_{i=1}^n w_i \left(1 - \frac{r_i}{d(x, p_i)}\right) (x - p_i). \quad (20.3)$$

پس از ساده شدن رابطه (۲۰.۳)، و با استفاده از روش نقطه ثابت، عبارت بازگشتی زیر به دست می‌آید:

$$x^{(k+1)} = \frac{\sum_{i=1}^n w_i \left(1 - \frac{r_i}{d(x^{(k)}, p_i)}\right) p_i}{\sum_{i=1}^n w_i \left(1 - \frac{r_i}{d(x^{(k)}, p_i)}\right)} \quad (21.3)$$

حال با استفاده از یک نقطه شروع اولیه مانند $x^{(0)}$ ، می‌توان به جواب بهینه مسئله (۱۹.۳) دست یافت.

۲.۳.۳ مسئله مکان‌یابی وبر با کمترین خطای مطلق

این مسئله در واقع همان مسئله (۱۹.۳) است، با این تفاوت که در جمله خطا به‌جای مجذور جملات از قدر مطلق آن‌ها استفاده شده است. این بخش به بررسی مسئله مورد نظر با خطای مطلق و نرم بلوکی به‌عنوان تابع فاصله که در مرجع [۸۵] آمده است، می‌پردازد.

با در نظر گرفتن فرضیات بخش قبل، مسئله وبر با کمترین خطای مطلق به‌صورت زیر فرموله می‌شود [۸۵]:

$$\min F(x) = \sum_{i=1}^n w_i |d(x, p_i) - r_i|. \quad (22.3)$$

لم ۱.۳.۳ و قضیه ۱.۳.۳ نیز برای مسئله (۲۲.۳) برقرار است. در حالتی که $d(\cdot, \cdot)$ یک نرم بلوکی باشد، مدل مسئله فوق به‌صورت زیر نوشته می‌شود:

$$\min F(x) = \sum_{i=1}^n w_i \left| \|x, p_i\|_B - r_i \right|. \quad (23.3)$$

لذا با توجه به تعریف نرم بلوکی می‌توان مسئله (۲۳.۳) را به‌صورت یک مسئله برنامه‌ریزی خطی تبدیل نمود یعنی [۸۵]:

برای هر $i = 1, 2, \dots, n$ قرار می‌دهیم:

$$\|x - p_i\|_B = \sum_{g=1}^r w_i (\lambda_{gi}^+ + \lambda_{gi}^-), \quad x - p_i = \sum_{g=1}^r (\lambda_{gi}^+ - \lambda_{gi}^-) b_g,$$

و

$$z_i^+ - z_i^- = \|x - p_i\|_B - r_i.$$

لذا می‌توان نوشت:

$$\min \sum_{i=1}^n w_i (z_i^+ + z_i^-) \quad (24.3)$$

$$\sum_{g=1}^r w_i (\lambda_{gi}^+ + \lambda_{gi}^-) - z_i^+ + z_i^- = r_i, \quad i = 1, \dots, n,$$

$$x - p_i = \sum_{g=1}^r (\lambda_{gi}^+ - \lambda_{gi}^-) b_g, \quad i = 1, \dots, n,$$

$$z_i^+, z_i^- \geq 0, \quad i = 1, \dots, n,$$

$$\lambda_{gi}^+, \lambda_{gi}^- \geq 0, \quad i = 1, \dots, n, \quad g = 1, 2, \dots, n.$$

توجه شود که مدل (۲۴.۳)، $2n + 2rn + 2$ متغیر و $3n$ محدودیت دارد.

فصل ۴

مسئله ۲- میانه پشتیبان نیمه ناخوشایند/ناخوشایند روی درخت

۱.۴ مقدمه

در سال‌های اخیر، سرویس‌دهندگان نیمه‌ناخوشایند^۱، یعنی سرویس‌دهنده‌هایی که مطلوب است به‌دور از بخشی از مشتریان مکان‌یابی شوند توجه بسیاری را به خود جلب کرده است.

برکارد و کراروپ اولین کسانی بودند که مدل مکان‌یابی با وزن‌های مثبت و منفی را روی شبکه‌ها مورد مطالعه قرار دادند [۲۹]. آن‌ها یک الگوریتم با زمان خطی را برای مسئله ۱- میانه روی کاکتوس، یعنی گراف همبندی که هیچ یک از دو دور در آن بیشتر از یک رأس مشترک نداشته باشند، بسط دادند. همچنین برکارد و همکارانش مسئله ۲- میانه نیمه‌ناخوشایند روی درخت‌ها را بررسی کردند [۲۲]. آن‌ها دو تابع هدف متفاوت را پیشنهاد دادند. اولین تابع هدف سعی دارد مجموع کمترین فاصله وزنی روی تمام رئوس (MWD) را کمینه کند درحالی‌که تابع هدف دوم مجموع وزنی کمترین فاصله از رأس v ، به سرویس‌دهنده را روی تمام رئوس v ، (WMD) کمینه

¹Semi obnoxious facility

می‌کند. برای مدل MWD ، آن‌ها تعدادی از قضایا را برای مسئله مورد نظر به کار برده، و برای یافتن ۲- میانه یک الگوریتم با زمان $O(n^2)$ روی درخت، یک الگوریتم با زمان $O(n \log n)$ روی ستاره و یک الگوریتم با زمان خطی روی مسیر مطرح کردند. اما برای مدل MWD ، آن‌ها تنها توانستند یک الگوریتم روی درخت که در زمان $O(n^3)$ اجرا می‌شود را بیابند که اگر میانه‌ها را تنها روی رئوس جستجو کنیم پیچیدگی زمانی آن به $O(n^2)$ کاهش می‌یابد. این نتایج توسط بنکوکزی و همکارانش بهبود یافتند [۱۵]. آن‌ها نشان دادند که برای مدل MWD ، ۲- میانه روی درخت با رئوس با وزن مثبت و منفی می‌تواند در زمان $O(n \log n)$ پاسخ داده شود. آن‌ها همچنین برای مدل WMD ، یک الگوریتم با زمان $O(nh \log^2 n)$ که h ارتفاع درخت را نشان می‌دهد، پیشنهاد دادند. از طرف دیگر زمانی که تنها وزن منفی وجود داشته باشد، برکارد و همکارانش نشان دادند که جواب بهینه مسئله ۲- میانه با تابع هدف MWD ، در زمان خطی به دست می‌آید [۲۴]. همچنین زمانی که شبکه مورد بررسی یک دور باشد، برکارد و هاتزل برای مسئله ۲- میانه، یک الگوریتم با زمان خطی برای مدل MWD و یک الگوریتم با زمان $O(n^2)$ برای مدل WMD به کار بردند [۲۸]. برای حالت $p = 3$ برکارد و فتحعلی، الگوریتمی با زمان $O(n^5)$ برای مدل WMD روی درخت و الگوریتمی با زمان $O(n^3)$ وقتی که میانه‌ها محدود به رئوس هستند، ارائه کردند [۲۳].

لذا با توجه به مسئله ۲- میانه پشتیبان روی درخت که در فصل سوم بیان شد و همچنین مسئله ۲- میانه با وزن مثبت - منفی که در مقدمه به آن اشاره کردیم، بر آن شدیم تا با ترکیب این دو مسئله، مسئله ۲- میانه پشتیبان نیمه‌ناخوشایند با مدل MWD را تعریف کنیم.

۲.۴ مسئله ۲- میانه پشتیبان نیمه ناخوشایند

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

$$W(T') = \sum_{v_i \in T'} w_i.$$

همچنین فرض کنید ρ_1 و ρ_2 احتمال شکست دو سرویس‌دهنده در مسئله ۲- میانه پشتیبان نیمه‌ناخوشایند باشند. برای ساده‌تر شدن مسئله فرض می‌کنیم که

$$\rho_1 = \rho_2 = \rho.$$

در مسئله ۲- میانه پشتیبان نیمه ناخوشایند، هدف کمینه کردن مجموع فاصله مورد انتظار از تمام رئوس به مجموعه سرویس دهنده‌های در حال کار است یعنی می‌خواهیم تابع هدف زیر را کمینه شود:

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

فرض کنید v_1 و v_2 پاسخ مسئله مورد بحث باشند. نقطه‌ای از T را با m نشان می‌دهیم که $m \in P(v_1, v_2)$ و $d(v_1, m) = d(v_2, m)$. فرض کنید $e(x, y)$ یالی در مسیر $P(v_1, v_2)$ باشد که شامل نقطه m است. بدون از دست دادن کلیت مسئله، فرض کنید رأس x روی مسیری که v_1 را به m متصل می‌کند قرار داشته باشد. علاوه بر این اگر m یک رأس باشد، فرض می‌کنیم که منطبق بر x است.

از طرفی چون برای هر رأس $v_i \in T_x$ و $v_j \in T_y$ داریم: $d(v_1, v_i) \leq d(v_2, v_i)$ و $d(v_1, v_j) > d(v_2, v_j)$ بنابراین می‌توان نتیجه گرفت که رئوس با وزن نامنفی در T_x و رئوس با وزن منفی در T_y ، توسط v_1 سرویس‌دهی می‌شوند. رئوس باقی‌مانده از v_2 سرویس می‌گیرند. (اگر $m = x$ باشد فرض می‌کنیم که x و تمامی رئوس دیگر که در فاصله یکسانی از m_1 و m_2 قرار دارند، از m_1 سرویس می‌گیرند.) بنابراین ما مجموعه رئوس V را به دو مجموعه رأسی V_1 و V_2 افراز می‌کنیم به‌طوری‌که:

$$V_1 = \{v | d(v, x) \leq d(v, y), w(v) \geq 0\} \cup \{v | d(v, x) \geq d(v, y), w(v) \leq 0\},$$

و

$$V_2 = V \setminus V_1.$$

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

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

در نتیجه تابع هدف مسئله ۲- میانه پشتیبان نیمه ناخوشایند را می‌توان به صورت زیر بازنویسی کرد:

$$F(v_1, v_2) = (1 - \rho)[D(V_1, v_1) + D(V_2, v_2)] + \rho[D(V_1, v_1) + D(V_2, v_1) + D(V_1, v_2) + D(V_2, v_2)] \\ = D(V_1, v_1) + \rho D(V_2, v_1) + \rho D(V_1, v_2) + D(V_2, v_2).$$

باید توجه کرد که مسئله ناخوشایند یک حالت خاص از مسئله نیمه‌ناخوشایند است که در آن وزن تمام رئوس منفی است. بنابراین در این مورد مجموعه‌های V_1 و V_2 را به صورت زیر تعریف می‌کنیم:

$$V_1 = \{v \in V \mid d(v, x) \geq d(v, y)\},$$

و

$$V_2 = \{v \in V \mid d(v, x) < d(v, y)\}.$$

۱.۲.۴ چند خاصیت روی درخت‌ها

یک خاصیت بسیار مهم برای یک مسئله ۲- میانه کلاسیک با وزن‌های نامنفی بر روی شبکه‌ها خاصیت بهینگی راسی است که در قضیه ۱.۳.۲ بیان نموده‌ایم. این خاصیت برای گراف‌هایی با وزن‌های دلخواه (مثبت و منفی) برقرار نمی‌باشد [۲۲]. با این حال برکارد و همکاران نشان دادند که در یک شبکه غیر جهت‌دار درختی این خاصیت برای مدل MWD برقرار است [۲۲].

ما در لم زیر خاصیت بهینگی راسی را برای یک مسئله ۲- میانه پشتیبان نیمه‌ناخوشایند بر روی یک درخت بیان می‌کنیم.

لم ۱.۲.۴. برای درخت T در مسئله ۲- میانه پشتیبان نیمه‌ناخوشایند یک جواب بهینه وجود دارد به طوری که تمام سرویس‌دهنده‌ها روی رئوس درخت T قرار دارند.

برهان.

به برهان خلف فرض کنید که هیچ جوابی وجود نداشته باشد که تمام سرویس‌دهنده‌ها روی رئوس درخت T قرار داشته باشند. فرض کنید (m_1, m_2) یک جواب بهینه مسئله روی درخت T است به طوری که، m_1 ، یک نقطه غیر رأسی روی درخت باشد. همچنین فرض کنید $e(v_r, v_q)$ یال شامل m_1 است و V_1 ، زیرمجموعه‌ای از رئوس درخت T است که توسط m_1 سرویس‌دهی می‌شوند. تابع هدف زیر که وابسته به m_1 است را در نظر می‌گیریم:

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

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

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

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

قرار دهید:

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

بنابراین داریم:

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

حال اگر $K \geq 0$ ، چون برای هر نقطه v_i روی قطعه یال $P(v_r, m_1)$ داریم، $d(v_r, v_i) < d(v_r, m_1)$ ، بنابراین نتیجه می‌شود که، $F(v_i, m_2) < F(m_1, m_2)$ ، که این تناقض است. و اگر $K < 0$ ، چون برای هر نقطه v_j روی قطعه یال $P(m_1, v_q)$ داریم، $d(v_r, v_j) > d(v_r, m_1)$ ، پس می‌توان نتیجه گرفت که، $F(v_q, m_2) < F(m_1, m_2)$ ، به‌طوریکه v_q و m_2 هر دو رأس هستند که این نیز تناقض است. بنابراین فرض خلف باطل و حکم اثبات می‌شود. \square

یک روش کارآمد برای حل مسئله ۲- میانه با وزن‌های مثبت و منفی بر روی درخت، روش حذف یال است که توسط برکارد و همکارانش ارائه شده است [۲۲]. همچنین این روش حذف یال برای پیدا کردن جواب مسئله ۲- میانه پشتیبان با وزن‌های مثبت بر روی درخت‌ها نیز توسط وانگ و همکارانش مورد استفاده قرار گرفته است [۱۳۸]. اما این روش حذف یال برای مسئله ۲- میانه پشتیبان با وزن مثبت و منفی بر روی درخت‌ها قابل اجرا نمی‌باشد. بنابراین با توجه به خاصیت بهینگی راسی که در لم ۱.۲.۴ آمده است می‌توان یک جواب بهینه مسئله ۲- میانه پشتیبان با وزن مثبت و منفی بر روی درخت‌ها را در زمان $\mathcal{O}(n^3)$ پیدا نمود.

۳.۴ مسئله ۲- میانه پشتیبان ناخوشایند.

در این قسمت مسئله ناخوشایند که حالت خاصی از یک مسئله نیمه ناخوشایند است را مورد بررسی قرار داده‌ایم. در یک مدل مکان‌یابی ناخوشایند هیچ راسی دارای وزن مثبت نیست.

درخت $T = (V, E)$ را که $V = \{v_1, v_2, \dots, v_n\}$ و $w(v)$ به ترتیب مجموعه رئوس و وزن هر راس می‌باشند را در نظر بگیرید. یک مسئله ناخوشایند بر روی درخت T ، مسئله‌ای است که در آن وزن تمام رئوس درخت T منفی باشد. به عبارت دیگر داشته باشیم $w(v) < 0, \forall v \in V$. در یک مسئله ناخوشایند ۲- میانه پشتیبان، به دنبال آن هستیم که تابع هدف زیر را کمینه کنیم:

$$F(v_1, v_2) = (1 - \rho) \sum_{v \in V, w(v) < 0} \min_{1 \leq i \leq 2} w(v) d(v, v_i) + \rho \left[\sum_{v \in V, w(v) < 0} w(v) d(v, v_1) + \sum_{v \in V, w(v) < 0} w(v) d(v, v_2) \right]. \quad (1.4)$$

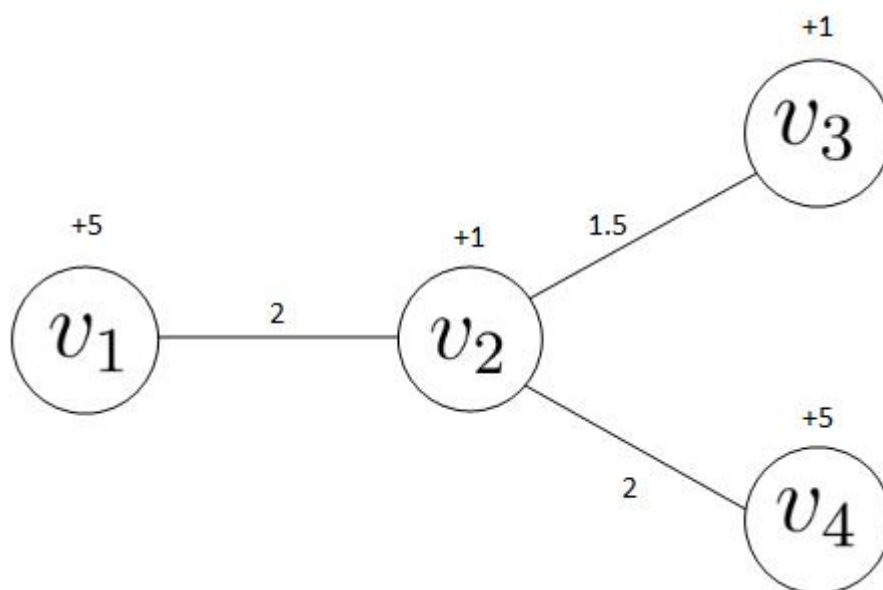
تابع هدف (۱.۴) معادل با یک مسئله ۲- ماکسین پشتیبان است. در یک مسئله ۲- ماکسین پشتیبان وزن تمام رئوس مثبت می‌باشد به عبارت دیگر $w(v) \geq 0, \forall v \in V$ ، و تابع هدف زیر باید بیشینه شود:

$$F_1(v_1, v_2) = (1 - \rho) \sum_{v \in V, w(v) \geq 0} \max_{1 \leq i \leq 2} w(v) d(v, v_i) + \rho \left[\sum_{v \in V, w(v) \geq 0} w(v) d(v, v_1) + \sum_{v \in V, w(v) \geq 0} w(v) d(v, v_2) \right]. \quad (2.4)$$

برکارد و همکارنش در مقاله [۲۴] نشان دادند که جواب مسئله ۲- ماکسین روی یک درخت، بر روی دو راس انتهایی قطر درخت قرار دارد. در مثال زیر نشان داده‌ایم که این خاصیت برای یک مسئله ۲- ماکسین پشتیبان برقرار نمی‌باشد.

مثال ۱.۳.۴. درخت T ، با ۴ راس به صورت شکل ۱.۴ را در نظر بگیرید. در حالتی که $\rho = 0$ با یک مسئله ۲- ماکسین کلاسیک مواجه هستیم، در این حالت جواب بهینه مسئله بر روی دو راس انتهایی بزرگترین مسیر درخت قرار دارد. لذا $\{v_1, v_4\}$ جواب بهینه مسئله در این حالت می‌باشد به طوری که، $F_1(v_1, v_4) = 45/5$.

اما در حالتی که $\rho > 0$ باشد مقدار تابع هدف مسئله به ازای جفت راس $\{v_1, v_4\}$



شکل ۱.۴: درخت T با ۴ راس.

به صورت زیر محاسبه می شود:

$$F_1(v_1, v_4) = (1 - \rho)45/5 + \rho(25/5 + 25/5) = 45/5 + 5/5\rho.$$

حال آن که با انتخاب دو راس $\{v_1, v_3\}$ به عنوان جواب برای مسئله ۲- میانۀ پشتیبان ناخوشایند به مقدار تابع هدفی به شکل زیر خواهیم رسید:

$$F_1(v_1, v_3) = (1 - \rho)43 + \rho(25/5 + 36/5) = 43 + 19\rho.$$

بنابراین اگر $0 < \rho \leq 1/185$ ، در این صورت $F_1(v_1, v_3) \geq F_1(v_1, v_4)$. بنابراین دو راس انتهایی قطر یک درخت نمی تواند جواب بهینه مسئله ۲- ماکسین پشتیبان باشد.

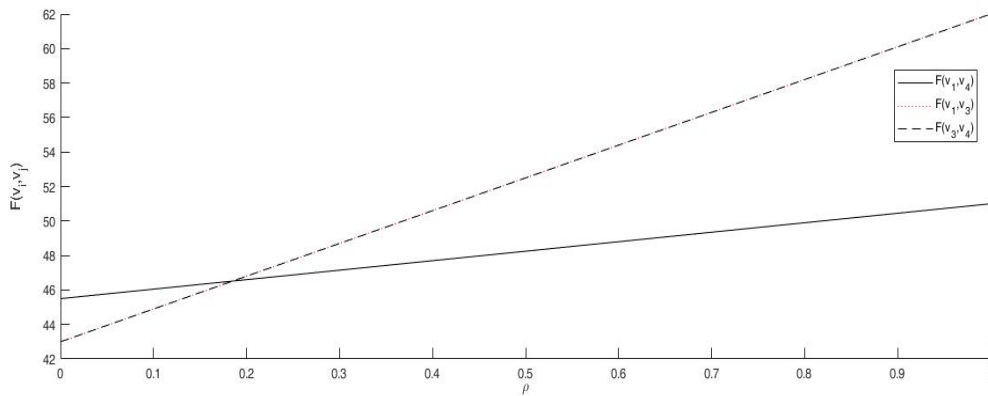
شکل ۲.۴ مقادیر تابع هدف مسئله را نسبت به سه جواب $\{v_1, v_3\}$ ، $\{v_1, v_4\}$ و $\{v_3, v_4\}$ نسبت به تغییرات پارامتر ρ نشان می دهد.

تینگ در مقاله [۱۳۳] نشان داد که جواب بهینه مسئله ۱- ماکسین بر روی برگ های درخت T قرار دارد. از طرفی همان طور که در این قسمت نیز ذکر شد جواب بهینه مساله ۲- ماکسین نیز بر روی دو راس انتهایی بزرگترین مسیر درخت T قرار دارد. مثال ۲.۳.۴ نشان می دهد که لزوماً جواب بهینه مسئله ۲- ماکسین پشتیبان بر روی برگ های درخت T قرار ندارد.

مثال ۲.۳.۴. مسیر نشان داده شده در شکل ۳.۴ را در نظر بگیرید.

مقدار تابع هدف به ازای جواب $\{v_1, v_5\}$ به صورت زیر بدست می آید:

$$F_1(v_1, v_5) = (1 - \rho)(7 + 52) + \rho(15 + 53) = 59 + 9\rho.$$



شکل ۲.۴: مقادیر تابع هدف برای سه جواب مسئله، نسبت به تغییرات $0 \leq \rho \leq 1$

از طرف دیگر، مقدار تابع هدف به ازای جواب $\{v_2, v_5\}$ نیز به صورت زیر است:

$$F_1(v_2, v_5) = (1 - \rho)(5 + 52) + \rho(18 + 53) = 57 + 14\rho.$$

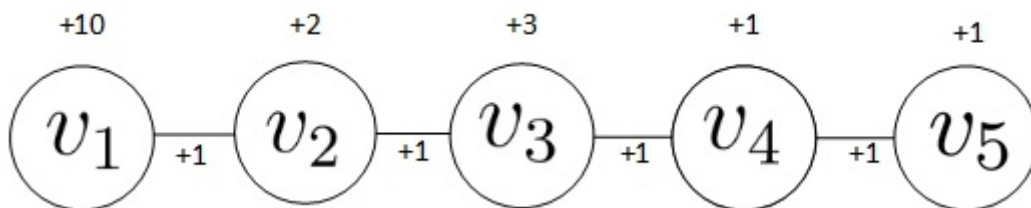
بنابراین اگر $57 + 14\rho \leq 59 + 9\rho$ یا $\frac{2}{5} < \rho \leq 1$ ، در این صورت نتیجه می شود که

$$F_1(v_2, v_5) \geq F_1(v_1, v_5).$$

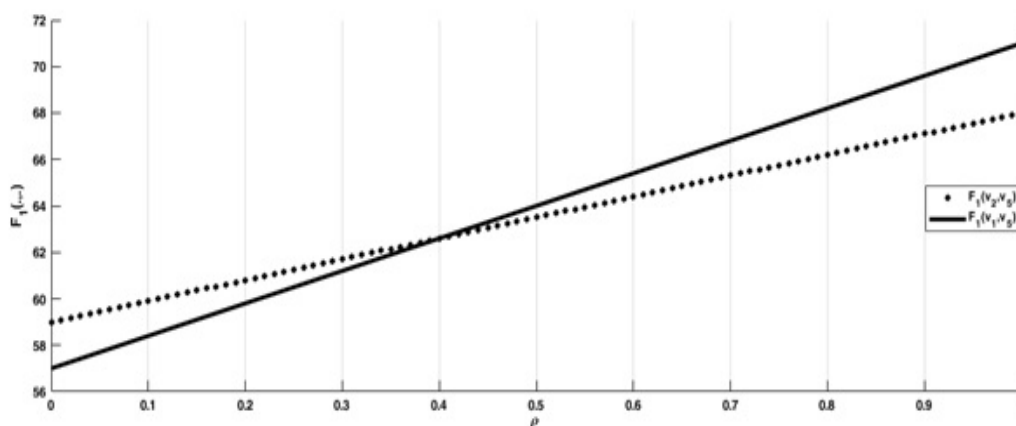
بنابراین راس انتهایی (برگ) v_1 ، نمی تواند به عنوان قسمتی از جواب بهینه مسئله

باشد. شکل ۴.۴ مقادیر تابع هدف را به ازای جواب های $\{v_1, v_5\}$ و $\{v_2, v_5\}$ نسبت به

مقادیر مختلف $0 \leq \rho \leq 1$ نشان می دهد.



شکل ۳.۴: یک مسیر با ۵ راس.



شکل ۴.۴: مقادیر تابع هدف به‌ازای جواب‌های $\{v_1, v_5\}$ و $\{v_2, v_5\}$ ، نسبت به تغییرات $0 \leq \rho \leq 1$

بنابراین با توجه به مطالب بالا برای پیدا کردن جواب بهینه مسئله ۲- ماکسین پشتیبان نیاز به (n) مقایسه تابع هدف داریم، بنابراین حل مسئله ۲- ماکسین پشتیبان دارای پیچیدگی زمانی ای معادل $O(n^3)$ است.

فصل ۵

مسئله مکان‌یابی پشتیبان چندوسیله‌ای آرمانی

۱.۵ مسئله مکان‌یابی پشتیبان چندوسیله‌ای آرمانی تحت

نرم l_p

فرض کنید می‌خواهیم فروشگاه‌های را در یک شهر احداث کنیم، با توجه به ساختار زندگی شهری، نحوه و توزیع جمعیت، هرچه فروشگاه به مرکز شهر و مناطق پرجمعیت نزدیک‌تر باشد از یک طرف دسترسی ساکنان افزایش یافته و متوسط هزینه حمل و نقل آن‌ها کاسته شده و تقاضای مشتریان به خوبی برآورده می‌شود و از طرف دیگر احداث فروشگاه در چنین جایگاه‌های هزینه‌های زیادی (خرید زمین و مالیات و غیره) در پی دارد، آلودگی صوتی و ترافیک ناشی از آن، تردد را دچار مشکل می‌کند که همگی از جمله آثار نامطلوب نزدیکی فروشگاه به مرکز شهر است. تصمیم‌گیری برای احداث چنین سرویس‌دهندگانی که در دنیای واقعی با آن روبه‌رو هستیم کار پیچیده و دشواری است. ایده‌ای که برای مدلسازی این مسائل پیشنهاد شده است این می‌باشد که برای دوری و نزدیکی از مرکز شهر یک کران مشترک (شعاع آرمانی) در نظر گرفته شود که

در نهایت به مسائل مکان‌یابی آرمانی می‌رسیم. با توجه به [۵۶] که در بخش ۳.۳ این رساله به آن اشاره کردیم، برای هر سرویس‌دهنده موجود P_i ، شعاع آرمانی R_i را در نظر می‌گیریم. فرض کنید که $P_i = (a_i, b_i)$ و $X_j = (x_j, y_j)$ باشند. با در نظر گرفتن تمام مفروضات بخش ۱.۳.۲ این رساله، یک مسئله مکان‌یابی پشتیبان چندوسیله‌ای آرمانی را می‌توان به صورت زیر مدل‌بندی ریاضی نمود:

$$\min F_R(X_1, \dots, X_m) = \sum_{t=1}^{k+1} \alpha_{t-1} f_R(X_t, \dots, X_m). \quad (1.5)$$

به طوری که

$$f_R(X_t, \dots, X_m) = \sum_{i=1}^n \sum_{j=t}^m w_{ij} (l_p(X_j, P_i) - R_i)^2 + \sum_{j=t}^{m-1} \sum_{l=j+1}^m v_{jl} l_p(X_j, X_l). \quad (2.5)$$

و α_i -ها احتمال شکست سرویس‌دهنده‌ها می‌باشند. قضیه زیر را در مورد مسئله (۱.۵) داریم:

قضیه ۱.۱.۵. جواب بهینه مسئله پشتیبان چندوسیله‌ای آرمانی، در پوسته گسترش یافته مستطیلی نقاط $P_i = (a_i, b_i)$ قرار دارد.

برهان. در نظر بگیرید که

$$\begin{cases} a_{\min} = \min\{a_i - R_i \mid i = 1, \dots, n\}, \\ a_{\max} = \max\{a_i + R_i \mid i = 1, \dots, n\}, \\ b_{\min} = \min\{b_i - R_i \mid i = 1, \dots, n\}, \\ b_{\max} = \max\{b_i + R_i \mid i = 1, \dots, n\}. \end{cases}$$

باشد. همچنین مجموعه نقاط $RH_3 = (a_{\min}, b_{\max})$ ، $RH_4 = (a_{\max}, b_{\min})$ و $RH_5 = (a_{\max}, b_{\max})$ را در نظر بگیرید. فرض کنید $X_j = (x_j, y_j)$ باشد و برای هر X_j ، $j = 1, \dots, n$ خارج از پوسته مستطیلی نقاط RH_3 ، RH_4 ، RH_5 و RH_6 قرار داشته باشد.

در حالت اول فرض کنید $x_j > a_{\max}$. بنابراین اگر قرار دهیم $X'_j = (a_{\max}, y_j)$ در این صورت داریم:

$$l_p(X_j, P_i) = (|x_j - a_i|^p + |y_j - b_i|^p)^{\frac{1}{p}} > (|a_{\max} - a_i|^p + |y_j - b_i|^p)^{\frac{1}{p}} = l_p(X'_j, P_i) > R_i$$

با توجه به رابطه بالا می‌توان نتیجه گرفت که:

$$\left((|x_j - a_i|^p + |y_j - b_i|^p)^{\frac{1}{p}} - R_i \right)^2 > \left((|a_{\max} - a_i|^p + |y_j - b_i|^p)^{\frac{1}{p}} - R_i \right)^2$$

بنابراین با توجه به مثبت بودن ضرایب w_{ij} ، داریم:

$$\begin{aligned} \sum_{i=1}^n \sum_{j=t}^m w_{ij} \left((|x_j - a_i|^p + |y_j - b_i|^p)^{\frac{1}{p}} - R_i \right)^2 &> \sum_{i=1}^n \sum_{j=t}^m w_{ij} \left((|a_{\max} - a_i|^p + |y_j - b_i|^p)^{\frac{1}{p}} - R_i \right)^2 \\ \rightarrow \sum_{i=1}^m \sum_{j=t}^m w_{ij} \left((|x_j - a_i|^p + |y_j - b_i|^p)^{\frac{1}{p}} - R_i \right)^2 &+ \sum_{j=t}^{m-1} \sum_{l=j+1}^m v_{jl} (|x_j - x_l|^p + |y_j - y_l|^p)^{\frac{1}{p}} > \\ \sum_{i=1}^n \sum_{j=t}^m w_{ij} \left((|a_{\max} - a_i|^p + |y_i - b_i|^p)^{\frac{1}{p}} - R_i \right)^2 &+ \sum_{j=t}^{m-1} \sum_{l=j+1}^m v_{jl} (|a_{\max} - x_l|^p + |y_j - y_l|^p)^{\frac{1}{p}} \end{aligned}$$

پس می‌توان گفت:

$$f_R(X_t, \dots, X_m) > f_R(X'_t, \dots, X'_m)$$

$$\rightarrow F = \sum_{t=1}^{k+1} \alpha_{t-1} f(X_t, \dots, X_m) > F_R = \sum_{t=1}^{k+1} \alpha_{t-1} f_R(X'_t, \dots, X'_m).$$

بنابراین $X_j = (x_j, y_j)$ نمی‌تواند جواب بهینه باشد.

برای حالت‌هایی که $x_j < a_{\min}$ ، $y_j < b_{\min}$ و $y_j > b_{\max}$ نیز به صورت کاملاً مشابه می‌توان اثبات نمود که نقاط X_j نمی‌توانند جواب‌های بهینه باشند. بنابراین نتیجه می‌شود که جواب بهینه مسئله در پوسته مستطیلی حاصل از نقاط RH_1, RH_2, RH_3 و RH_4 قرار دارد. □

لم ۱.۱.۵. مشتقات تابع هدف مسئله مکان‌یابی پشتیبان چندوسیله‌ای آرمانی، ممکن است در هر جایی از صفحه تعریف نشده باشند.

برهان. با مشتق‌گیری از تابع هدف می‌توان گفت در نقاطی از صفحه که مخرج مشتق رابطه (۱.۵) صفر می‌شود، مشتق تعریف نشده است. □

با توجه به لم ۱.۱.۵، برای اینکه تابع هدف مسئله (۱.۵) مشتق‌پذیر باشد قرار می‌دهیم:

$$f_R^h(X_t, \dots, X_m) = \sum_{i=1}^n \sum_{j=t}^m w_{ij} (l_p^h(X_j, P_i) - R_i)^2 + \sum_{j=t}^{m-1} \sum_{l=j+1}^m v_{jl} l_p^h(X_j, X_l). \quad (۳.۵)$$

به‌طوری‌که:

$$l_p^h(X_j, P_i) = \left(\left((x_j - a_i)^2 + \epsilon \right)^{\frac{p}{2}} + \left((y_j - b_i)^2 + \epsilon \right)^{\frac{p}{2}} \right)^{\frac{1}{p}}. \quad (۴.۵)$$

و

$$l_p^h(X_j, X_l) = \left(\left((x_j - x_l)^2 + \epsilon \right)^{\frac{p}{2}} + \left((y_j - y_l)^2 + \epsilon \right)^{\frac{p}{2}} \right)^{\frac{1}{p}}. \quad (۵.۵)$$

بنابراین با توجه به روابط بالا به جای بهینه شدن رابطه (۱.۵) به دنبال بهینه شدن رابطه زیر هستیم:

$$\min F_R^h(X_1, \dots, X_m) = \sum_{t=1}^{k+1} \alpha_{t-1} f_R^h(X_t, \dots, X_m). \quad (۶.۵)$$

رابطه (۶.۵) را تقریب رابطه (۱.۵) می‌نامیم. در لم زیر بیان می‌کنیم که تابع هدف مسئله تقریب زده شده، یعنی رابطه (۶.۵)، به تابع هدف مسئله اصلی یعنی رابطه (۱.۵) همگرا است.

لم ۲.۱.۵. تابع هدف تقریب زده شده‌ی $F_R^h(X_1, \dots, X_m)$ به تابع هدف مسئله اصلی یعنی $F_R(X_1, \dots, X_m)$ همگرا است و داریم:

$$(\epsilon \rightarrow 0) \Rightarrow \max\{|F_R^h(X_1, \dots, X_m) - F_R(X_1, \dots, X_m)|\} \rightarrow 0.$$

برهان. با توجه به (۲.۵) و (۳.۵) داریم:

$$\begin{aligned} f_R^h(X_t, \dots, X_m) - f_R(X_t, \dots, X_m) &= \\ &= \sum_{i=1}^n \sum_{j=t}^m w_{ij} \left(\left(\left((x_j - a_i)^2 + \epsilon \right)^{\frac{p}{2}} + \left((y_j - b_i)^2 + \epsilon \right)^{\frac{p}{2}} \right)^{\frac{1}{p}} - R_i \right)^2 \\ &\quad - \sum_{i=1}^n \sum_{j=t}^m w_{ij} \left((|x_j - a_i|^p + |y_j - b_i|^p)^{\frac{1}{p}} - R_i \right)^2 \\ &\quad + \sum_{j=t}^{m-1} \sum_{l=j+1}^m v_{jl} \left(\left((x_j - x_l)^2 + \epsilon \right)^{\frac{p}{2}} + \left((y_j - y_l)^2 + \epsilon \right)^{\frac{p}{2}} \right)^{\frac{1}{p}} \\ &\quad - \sum_{j=t}^{m-1} \sum_{l=j+1}^m v_{jl} (|x_j - x_l|^p + |y_j - y_l|^p)^{\frac{1}{p}}. \end{aligned}$$

با قرار دادن $s_1 = x_j - a_i$ ، $s_2 = y_j - b_i$ ، $s_3 = \epsilon^{\circ/\Delta}$ ، $s_4 = R_i$ و با استفاده از نامساوی مینکوفسکی داریم:

$$\begin{aligned} & \left(\left(\left((x_j - a_i)^r + \epsilon \right)^{\frac{p}{r}} + \left((y_j - b_i)^r + \epsilon \right)^{\frac{p}{r}} \right)^{\frac{1}{p}} - R_i \right)^2 \\ & - \left((|x_j - a_i|^p + |y_j - b_i|^p)^{\frac{1}{p}} - R_i \right)^2 = \\ & \left((s_1^r + s_2^r)^{\frac{p}{r}} + (s_2^r + s_3^r)^{\frac{p}{r}} \right)^{\frac{1}{p}} - (|s_1|^p + |s_2|^p)^{\frac{1}{p}} \\ & - 2s_4 \left(\left((s_1^r + s_2^r)^{\frac{p}{r}} + (s_2^r + s_3^r)^{\frac{p}{r}} \right)^{\frac{1}{p}} - (|s_1|^p + |s_2|^p)^{\frac{1}{p}} \right) \\ & \leq 2^{\frac{1}{p}} s_3^{\frac{p+1}{p}} + 2^{\frac{p+1}{p}} s_4 s_3 = 2^{\frac{1}{p}} \epsilon + 2^{\frac{p+1}{p}} s_4 s_3 = 2^{\frac{1}{p}} \epsilon + 2^{\frac{p+1}{p}} \epsilon^{\circ/\Delta} R_i. \end{aligned}$$

بنابراین می‌توان نتیجه گرفت:

$$\begin{aligned} & \sum_{i=1}^n \sum_{j=t}^m w_{ij} \left(\left(\left((x_j - a_i)^r + \epsilon \right)^{\frac{p}{r}} + \left((y_j - b_i)^r + \epsilon \right)^{\frac{p}{r}} \right)^{\frac{1}{p}} - R_i \right)^2 \\ & - \sum_{i=1}^n \sum_{j=t}^m w_{ij} \left((|x_j - a_i|^p + |y_j - b_i|^p)^{\frac{1}{p}} - R_i \right)^2 \leq \\ & 2^{\frac{1}{p}} \epsilon \left(\sum_{i=1}^n \sum_{j=t}^m w_{ij} \right) + 2^{\frac{p+1}{p}} \epsilon^{\circ/\Delta} \left(\sum_{i=1}^n \sum_{j=t}^m w_{ij} R_i \right). \end{aligned}$$

به‌طور مشابه نیز می‌توان اثبات کرد:

$$\begin{aligned} & \sum_{j=t}^{m-1} \sum_{l=j+1}^m v_{jl} \left(\left((x_j - x_l)^r + \epsilon \right)^{\frac{p}{r}} + \left((y_j - y_l)^r + \epsilon \right)^{\frac{p}{r}} \right)^{\frac{1}{p}} \\ & - \sum_{j=t}^{m-1} \sum_{l=j+1}^m v_{jl} (|x_j - x_l|^p + |y_j - y_l|^p)^{\frac{1}{p}} \leq \sum_{j=t}^{m-1} \sum_{l=j+1}^m (v_{jl} 2^{\frac{1}{p}} \epsilon). \end{aligned}$$

بنابراین با توجه به روابط بالا نتیجه می‌شود:

$$\begin{aligned} f_R^h(X_t, \dots, X_m) - f_R(X_t, \dots, X_m) & \leq 2^{\frac{1}{p}} \epsilon \left(\sum_{i=1}^n \sum_{j=t}^m w_{ij} \right) \\ & + 2^{\frac{p+1}{p}} \epsilon^{\circ/\Delta} \left(\sum_{i=1}^n \sum_{j=t}^m w_{ij} R_i \right) + 2^{\frac{1}{p}} \epsilon \left(\sum_{j=t}^{m-1} \sum_{l=j+1}^m v_{jl} \right). \end{aligned}$$

پس با توجه به رابطه (۶.۵) و (۱.۵) داریم:

$$\begin{aligned} \max\{|F_R^h(X_1, \dots, X_m) - F_R(X_1, \dots, X_m)|\} = \\ \sum_{t=1}^{k+1} \alpha_{t-1} (|F_R^h(X_t, \dots, X_m) - F_R(X_t, \dots, X_m)|) \leq \epsilon^{\frac{1}{p}} \left(\sum_{t=1}^{k+1} \alpha_{t-1} \left(\sum_{i=1}^n \sum_{j=t}^m w_{ij} \right) \right) \\ + \epsilon^{\frac{1}{p}} \left(\sum_{t=1}^{k+1} \alpha_{t-1} \left(\sum_{j=t}^{m-1} \sum_{l=j+1}^m v_{jl} \right) \right) + \epsilon^{\frac{p+1}{p}} \left(\sum_{t=1}^{k+1} \alpha_{t-1} \left(\sum_{i=1}^n \sum_{j=t}^m w_{ij} R_i \right) \right). \end{aligned}$$

با توجه به رابطه بالا زمانی که $\epsilon \rightarrow 0$ ، می‌توان نتیجه گرفت:

$$\max\{|F_R^h(X_1, \dots, X_m) - F_R(X_1, \dots, X_m)|\} \rightarrow 0$$

□

حال با توجه به این لم ۲.۱.۵ و مشتق‌پذیر بودن تابع هدف می‌توان یک روش شبه-وایز فیلد برای حل مسئله (۶.۵) ارائه داد.

۲.۵ الگوریتم شبه‌وایز فیلد

اگر شرط لازم بهینگی را برای رابطه (۶.۵) در نظر بگیریم در این صورت داریم:

$$\frac{\partial F_R^h(X_t, \dots, X_m)}{\partial x_r} = \begin{cases} \left\{ \sum_{i=1}^n \left(\frac{w_{ir}(Sa_n - R_i) \cdot (x_r - a_i)}{(Sa_n)^{p-1} Ra_n} \right) + \sum_{\substack{j=t \\ j \neq r}} \left(\frac{v_{rj}(x_r - x_j)}{Sx_{rj} Rx_{rj}} \right), & r = t, \dots, m, \right. \\ \left. 0, & r = 1, \dots, t-1, \right. \end{cases} \quad (7.5)$$

9

$$\frac{\partial F_R^h(X_t, \dots, X_m)}{\partial y_r} = \begin{cases} \left\{ \sum_{i=1}^n w_{ir} \frac{(Sa_n - R_i) \cdot (y_r - b_i)}{(Sa_n)^{p-1} Rb_n} + \sum_{\substack{j=t \\ j \neq r}} \frac{v_{rj}(y_r - y_j)}{Sx_{rj} Rr_j}, & r = t, \dots, m, \right. \\ \left. 0, & r = 1, \dots, t-1, \right. \end{cases} \quad (8.5)$$

به‌طوری‌که:

$$\begin{cases} Sa_n = \left(((x_r - a_i)^{\gamma} + \epsilon)^{\frac{p}{\gamma}} + ((y_r - b_i)^{\gamma} + \epsilon)^{\frac{p}{\gamma}} \right)^{\frac{1}{p}}, \\ Sx_{rj} = \left(((x_r - x_j)^{\gamma} + \epsilon)^{\frac{p}{\gamma}} + ((y_r - y_j)^{\gamma} + \epsilon)^{\frac{p}{\gamma}} \right)^{\frac{p-1}{p}}, \\ Ra_{ri} = ((x_r - a_i)^{\gamma} + \epsilon)^{\frac{\gamma-p}{\gamma}}, \\ Rb_n = ((y_r - b_i)^{\gamma} + \epsilon)^{\frac{\gamma-p}{p}}, \\ Rx_{rj} = ((x_r - x_j)^{\gamma} + \epsilon)^{\frac{\gamma-p}{p}}, \\ Ry_{rj} = ((y_r - y_j)^{\gamma} + \epsilon)^{\frac{\gamma-p}{p}}. \end{cases} \quad (9.5)$$

بنابراین چون داریم:

$$\frac{\partial F_R^h(X_1, \dots, X_m)}{\partial x_r} = \alpha_o \frac{\partial F_R^h(X_1, \dots, X_m)}{\partial r_r} + \dots + \alpha_k \frac{\partial F_R^h(X_{k+1}, \dots, X_m)}{\partial x_r}.$$

و

$$\frac{\partial F_R^h(X_1, \dots, X_m)}{\partial y_r} = \alpha_o \frac{\partial F_R^h(X_1, \dots, X_m)}{\partial y_r} + \dots + \alpha_k \frac{\partial F_R^h(X_{k+1}, \dots, X_m)}{\partial y_r}.$$

بنابراین می‌توان نتیجه گرفت:

$$\begin{aligned} \frac{\partial F_R^h(X_1, \dots, X_m)}{\partial x_r} = & \begin{cases} \sum_{t=1}^r \alpha_{t-1} \left(\sum_{i=1}^n \left(\frac{\gamma w_{ir} (Sa_n - R_i) \cdot (x_r - a_i)}{(Sa_n)^{p-1} Ra_{ri}} \right) + \sum_{\substack{j=t \\ j \neq r}} \left(\frac{v_{rj} (x_r - x_j)}{Sx_{rj} Rx_{rj}} \right) \right), & r = 1, \dots, k, \\ \sum_{t=1}^{k+1} \alpha_{t-1} \left(\sum_{i=1}^n \left(\frac{\gamma w_{ir} (Sa_n - R_i) \cdot (x_r - a_i)}{(Sa_n)^{p-1} Ra_{ri}} \right) + \sum_{\substack{j=t \\ j \neq r}} \left(\frac{v_{rj} (x_r - x_j)}{Sx_{rj} Rx_{rj}} \right) \right), & r = k+1, \dots, m, \end{cases} \\ & (10.5) \end{aligned}$$

و

$$\begin{aligned} \frac{\partial F_R^h(X_1, \dots, X_m)}{\partial y_r} = & \begin{cases} \sum_{t=1}^r \alpha_{t-1} \left(\sum_{i=1}^n \left(\frac{\gamma w_{ir} (Sa_n - R_i) \cdot (y_r - b_i)}{(Sa_n)^{p-1} Rb_{ri}} \right) + \sum_{\substack{j=t \\ j \neq r}} \left(\frac{v_{rj} (y_r - y_j)}{Sx_{rj} Ry_{rj}} \right) \right), & r = 1, \dots, k, \\ \sum_{t=1}^{k+1} \alpha_{t-1} \left(\sum_{i=1}^n \left(\frac{\gamma w_{ir} (Sa_n - R_i) \cdot (y_r - b_i)}{(Sa_n)^{p-1} Rb_{ri}} \right) + \sum_{\substack{j=t \\ j \neq r}} \left(\frac{v_{rj} (y_r - y_j)}{Sx_{rj} Ry_{rj}} \right) \right), & r = k+1, \dots, m, \end{cases} \\ & (11.5) \end{aligned}$$

لذا با مساوی صفر قرار دادن (۱۰.۵) و (۱۱.۵) و پیدا کردن x_r و y_r به رابطه تکراری زیر می‌رسیم:

$$(12.5) \quad x_r^{c+1} = \begin{cases} \frac{\sum_{t=1}^r \alpha_{t-1} \left(\sum_{i=1}^n \left(\frac{\Psi w_{ir}(Sa_{ri}^c - R_i) \cdot a_i}{(Sa_{ri}^c)^{p-1} Ra_{ri}^c} \right) + \sum_{j=t, j \neq r} \left(\frac{v_{rj} x_j^c}{Sx_{rj}^c Rx_{rj}^c} \right) \right)}{\sum_{t=1}^r \alpha_{t-1} \left(\sum_{i=1}^n \left(\frac{\Psi w_{ir}(Sa_{ri}^c - R_i)}{(Sa_{ri}^c)^{p-1} Ra_{ri}^c} \right) + \sum_{j=t, j \neq r} \left(\frac{v_{rj}}{Sx_{rj}^c Rx_{rj}^c} \right) \right)}, & r = 1, \dots, k, \\ \frac{\sum_{t=1}^{k+1} \alpha_{t-1} \left(\sum_{i=1}^n \left(\frac{\Psi w_{ir}(Sa_{ri}^c - R_i) \cdot a_i}{(Sa_{ri}^c)^{p-1} Ra_{ri}^c} \right) + \sum_{j=t, j \neq r} \left(\frac{v_{rj} x_j^c}{Sx_{rj}^c Rx_{rj}^c} \right) \right)}{\sum_{t=1}^{k+1} \alpha_{t-1} \left(\sum_{i=1}^n \left(\frac{\Psi w_{ir}(Sa_{ri}^c - R_i)}{(Sa_{ri}^c)^{p-1} Ra_{ri}^c} \right) + \sum_{j=t, j \neq r} \left(\frac{v_{rj}}{Sx_{rj}^c Rx_{rj}^c} \right) \right)}, & r = k+1, \dots, m, \end{cases}$$

و
(13.5)

$$y_r^{c+1} = \begin{cases} \frac{\sum_{t=1}^r \alpha_{t-1} \left(\sum_{i=1}^n \left(\frac{\Psi w_{ir}(Sa_{ri}^c - R_i) \cdot b_i}{(Sa_{ri}^c)^{p-1} Rb_{ri}^c} \right) + \sum_{j=t, j \neq r} \left(\frac{v_{rj} y_j^c}{Sx_{rj}^c Ry_{rj}^c} \right) \right)}{\sum_{t=1}^r \alpha_{t-1} \left(\sum_{i=1}^n \left(\frac{\Psi w_{ir}(Sa_{ri}^c - R_i)}{(Sa_{ri}^c)^{p-1} Rb_{ri}^c} \right) + \sum_{j=t, j \neq r} \left(\frac{v_{rj}}{Sx_{rj}^c Ry_{rj}^c} \right) \right)}, & r = 1, \dots, k, \\ \frac{\sum_{t=1}^{k+1} \alpha_{t-1} \left(\sum_{i=1}^n \left(\frac{\Psi w_{ir}(Sa_{ri}^c - R_i) \cdot a_i}{(Sa_{ri}^c)^{p-1} Rb_{ri}^c} \right) + \sum_{j=t, j \neq r} \left(\frac{v_{rj} y_j^c}{Sx_{rj}^c Ry_{rj}^c} \right) \right)}{\sum_{t=1}^{k+1} \alpha_{t-1} \left(\sum_{i=1}^n \left(\frac{\Psi w_{ir}(Sa_{ri}^c - R_i)}{(Sa_{ri}^c)^{p-1} Rb_{ri}^c} \right) + \sum_{j=t, j \neq r} \left(\frac{v_{rj}}{Sx_{rj}^c Ry_{rj}^c} \right) \right)}, & r = k+1, \dots, m, \end{cases}$$

که در آن منظور از c ، تکرار الگوریتم در هر مرحله می‌باشد. شرط توقف در روابط بازگشتی بالا را می‌توان $|F_h^c - F_h^{c-1}| < e$ در نظر گرفت به طوری که e حداکثر خطای الگوریتم تکراری بالا است.

۳.۵ نتایج محاسباتی

مثال ۱.۳.۵. در این قسمت نتایج یک مثال عددی با $n = 10$ و $m = 5$ که به صورت تصادفی تولید شده است نسبت به پارامترهای مختلفی چون p ، k و α_i ها آورده شده است. جداول ۱.۵ و ۲.۵ بیانگر مختصات و مسائل موجود و پارامترهای اولیه مسئله همچون وزن بین و مسائل موجود و سرویس‌دهنده‌های جدید (w_{ij}) و وزن بین هر دو سرویس‌دهنده جدید می‌باشد.

در جدول ۳.۵، نتایج حاصل از حل مسئله برای $k = 4$ و $p = 2$ برای حالات $R_i = 0/5$ و $\alpha_4 = 0/2$ ، $\alpha_3 = 0/4$ ، $\alpha_2 = 0/6$ ، $\alpha_1 = 0/8$ ، $\alpha_0 = 0/2$ $i = 1, \dots, n$ آمده است. در جدول ۴.۵ نیز نتایج حل مسئله مذکور نسبت به مقادیر

مختلف p ، در حالت $R_i = 0/5$ برای هر $i = 1, \dots, n$ آورده شده است. نمودار ۱.۵ نیز سرعت همگرایی روش شبه-وایز فیلد ارائه شده را در مقایسه با تعداد تکرار نشان می‌دهد. همچنین شرط توقف نیز $0/001 < |F_h^c - F_h^{c-1}|$ در نظر گرفته شده است. در مسئله فوق اگر تمامی R_i ها برابر با صفر در نظر گرفته شوند در این صورت با یک مسئله مکان‌یابی چندوسیله‌ای پشتیبان مواجه هستیم که این حالت مسئله توسط فتحعلی در مقاله [۵۵] مورد بررسی قرار گرفته است. در جدول ۵.۵ نتایج مقایسه‌ای مثال حاضر با مثال مقاله [۵۵] آورده شده است.

جدول ۱.۵: مختصات وسیله‌های موجود $(P_i = (a_i, b_i))$ و وزن بین سرویس‌دهنده‌های جدید و وسیله‌های موجود. $(w_{ji}, j = 1, \dots, 5)$

(i)	$P_i = (a_i, b_i)$	w_{1i}	w_{2i}	w_{3i}	w_{4i}	w_{5i}
۱	(۰, ۱۲)	۴	۳	۰	۱	۲
۲	(۲, ۱)	۰	۶	۰	۲	۳
۳	(۱۰, ۲)	۲	۰	۸	۳	۱
۴	(۶, ۱۲)	۰	۰	۱۰	۴	۵
۵	(۲۰, ۱۰)	۶	۸	۲	۱	۳
۶	(۵, ۲۰)	۵	۱	۶	۳	۴
۷	(۱۵, ۱۵)	۲	۴	۷	۲	۵
۸	(۲۲, ۵)	۷	۵	۰	۲	۴
۹	(۲۰, ۲۵)	۱	۲	۳	۴	۵
۱۰	(۲۵, ۲۵)	۱۸	۱	۵	۴	۲

جدول ۲.۵: وزن مربوط بین سرویس‌دهنده‌های جدید (v_{jl}) .

سرویس‌دهنده جدید	۱	۲	۳	۴	۵
۱	۰	۶	۱	۴	۵
۲	۶	۰	۴	۲	۳
۳	۱	۴	۰	۵	۲
۴	۴	۲	۵	۰	۸
۵	۵	۳	۲	۸	۰

جدول ۳.۵: نتایج بدست آمده برای $p = 2$ و $k = 4$ در حالت $R_i = 0/5, i = 1, \dots, m$

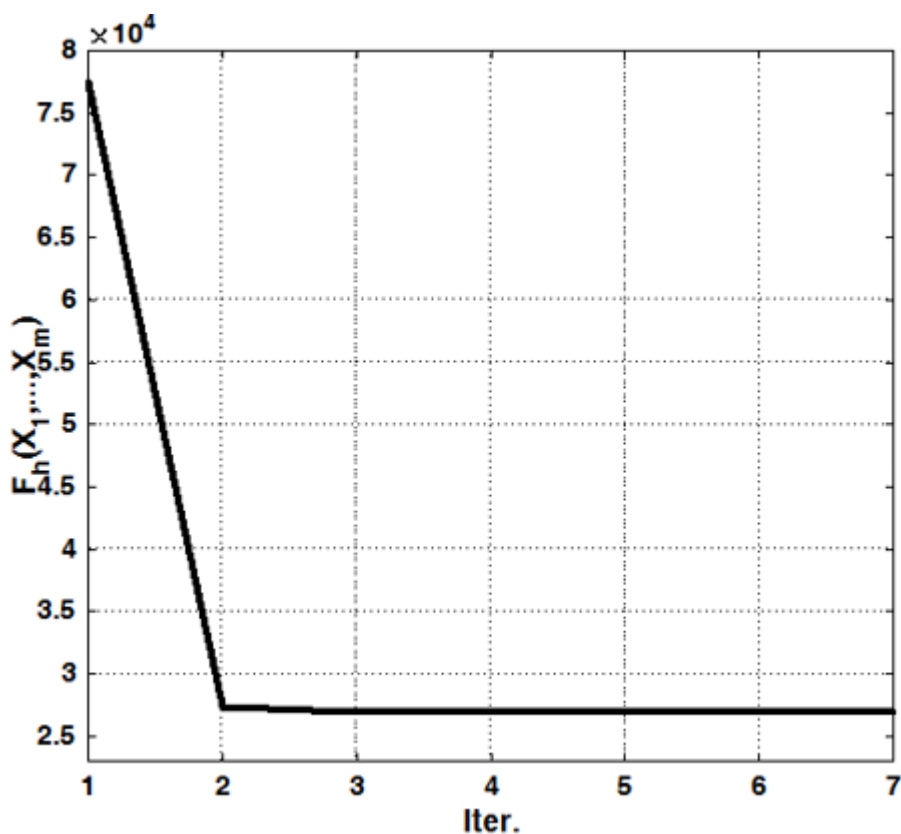
تکرار	۱	۲	...	۶	۷
X_1	(۳/۶۰, ۳/۳۱)	(۱۶/۱۱, ۱۴/۸۲)	...	(۱۸/۰۱, ۱۶/۶۰)	(۱۸/۰۱, ۱۶/۶۰)
X_2	(۲/۸۲, ۲/۰۶)	(۱۳/۱۱, ۹/۷۵)	...	(۱۳/۶۰, ۱۰/۲۱)	(۱۳/۶۰, ۱۰/۲۱)
X_3	(۳/۳۰, ۳/۸۴)	(۱۱/۵۸, ۱۳/۴۶)	...	(۱۲/۳۰, ۱۴/۱۵)	(۱۲/۳۰, ۱۴/۱۵)
X_4	(۱/۹۳, ۲/۱۱)	(۱۲/۱۳, ۱۳/۲۳)	...	(۱۳/۲۰, ۱۴/۳۵)	(۱۳/۲۰, ۱۴/۳۵)
X_5	(۲/۸۸, ۳/۰۸)	(۱۱/۷۷, ۱۲/۵۷)	...	(۱۲/۹۳, ۱۳/۸۶)	(۱۲/۹۳, ۱۳/۸۶)
F_R^h	۷۷۵۰۷/۱	۲۷۳۸۶/۱	...	۲۶۹۰۳/۹	۲۶۹۰۳/۹
F_R	۷۷۵۰۶/۶	۲۷۳۸۵/۷	...	۲۶۹۰۳/۵	۲۶۹۰۳/۵

جدول ۴.۵: نتایج بدست آمده برای حالت $k = 4$ و $R_i = 0/5, i = 1, \dots, m$ برای مقادیر مختلف p در ۲۰۰ تکرار.

تکرار	$p = 1$	$p = 2$	$p = 3$	$p = 10$
X_1	(۲۰/۲۷, ۱۷/۲۹)	(۱۸/۰۱, ۱۶/۶۰)	(۱۶/۷۵, ۱۶/۱۲)	(۱۷/۶۳, ۸/۰۹)
X_2	(۱۴/۱۳, ۹/۹۷)	(۱۳/۶۰, ۱۰/۲۱)	(۱۳/۴۴, ۱۱/۰۸)	(۱۱/۲۴, ۸/۳۳)
X_3	(۱۰/۵۷, ۱۵/۰۸)	(۱۲/۳۰, ۱۴/۱۵)	(۱۲/۳۳, ۱۳/۷۹)	(۱۹/۳۱, ۱۰/۸۷)
X_4	(۱۴/۰۸, ۱۴/۲۵)	(۱۳/۲۰, ۱۴/۳۵)	(۱۲/۹۹, ۱۴/۲۸)	(۱۷/۵۴, ۷/۰۵)
X_5	(۱۴/۱۰, ۱۴/۲۵)	(۱۲/۹۳, ۱۳/۸۶)	(۱۲/۹۹, ۱۴/۲۷)	(۷/۷۳, ۲۰/۱۱)
F_R^h	۴۹۳۱۳/۰	۲۷۳۸۶/۱	۲۲۶۵۷/۱	۳۲۱۰۷/۰
F_R	۴۹۳۰۹/۱	۲۷۳۸۵/۷	۲۲۶۵۶/۳	۳۲۱۰۶/۸

جدول ۵.۵: مقایسه نتایج بدست آمده از الگوریتم حاضر با مقالات دیگر، نسبت به مقادیر مختلف p .

$R_{i,p}$	F_R	فتحعلی [۵۵]
$R_i = \circ, p = 1$	۳۲۵۲/۳۶۱	۳۲۵۲/۳
$R_i = \circ, p = 2$	۲۴۲۱/۸۹۷	۲۴۴۳/۰
$R_i = \circ, p = 10$	۲۰۷۶/۸۱۲	۲۰۷۶/۹



شکل ۱.۵: نمودار نمودار سرعت همگرایی الگوریتم شبه-وایز فیلد ارائه شده در مقایسه با تعداد تکرار برای حالت $p = 2$.

فصل ۶

مسائل معکوس ۲ - میانه پشتیبان

در این قسمت مفاهیم و نمادگذاری‌های مورد نیاز در این فصل بیان می‌شود. فرض کنید $G = (V(G), E(G), w, l)$ یک گراف ساده و بدون جهت باشد که $V(G)$ ، مجموعه رأس‌ها، $E(G)$ ، مجموعه یال‌ها، $w : V(G) \rightarrow \mathbb{R}^+$ ، تابع وزنی رأس‌ها و $l : E(G) \rightarrow \mathbb{R}^+$ ، تابع طول یال‌های گراف G باشد. مجموع وزنی رأس‌های گراف G را با نماد $w(G)$ ، نمایش می‌دهیم به طوری که

$$w(G) = \sum_{v \in V(G)} w(v) d_G(u, v). \quad (1.6)$$

$d_G(u, v)$ فاصله بین دو رأس u و v در گراف G است و به صورت مجموع طولی کوتاه‌ترین مسیر بین دو رأس u و v ، تعریف می‌شود. اگر در گراف G داشته باشیم $U \subseteq V(G)$ ، در این صورت زیر گراف G شامل U را به صورت $G[U]$ ، نشان می‌دهیم. همچنین در درخت $T = (V, E)$ ، یک مسیر یکتا بین دو رأس u و v را به صورت $P[u, v]$ ، نمایش می‌دهیم.

برای هر زوج رأس (u, v) قرار می‌دهیم:

$$\Pi(u, v) = (U, V). \quad (2.6)$$

به طوری که $U = \{x | d(x, u) \leq d(x, v)\}$ و $V = V(T) - U$. با توجه به تعریف (۲.۶)، در یک درخت هر یک از مجموعه‌های U و V ، یک زیر گراف همبند را تولید می‌کنند. این

دو مجموعه با حذف یال $e \in E(T)$ به دست می آید که یال e یال مرکزی u و v است و همان نقطه میانی مسیر $P[u, v]$ است.

همان طور که قبلا در بخش ۱.۱.۳ این رساله اشاره کردیم، یک مسئله مکان یابی ۲- میانه پشتیبان به دنبال یک جفت راس مانند (m_1, m_2) می باشد به طوریکه تابع هدف زیر را کمینه کند [۱۳۸]:

$$F(m_1, m_2) = (1 - \rho) \sum_{v_i \in V} w(v_i) d(v_i, \{m_1, m_2\}) + \rho \sum_{v_i \in V} w(v_i) d(v_i, m_1) + \rho \sum_{v_i \in V} w(v_i) d(v_i, m_2). \quad (3.6)$$

در حالت معکوس می خواهیم با تغییر در پارامترهای اصلی مسئله همچون وزن نقاط، طول یال ها و یا مختصات نقاط و با کمترین هزینه ممکن دو رأس دلخواه m_1 و m_2 تبدیل به ۲- میانه پشتیبان شوند. در این قسمت معکوس نوع حداقل هزینه و نوع محدودیت بودجه ای مسئله ۲- میانه پشتیبان را در حالت گسسته با تغییر در وزن رئوس گراف و طول یال های گراف مورد بررسی قرار می دهیم.

۱.۶ معکوس نوع حداقل هزینه مسئله ۲- میانه پشتیبان

با تغییر در وزن رئوس

در معکوس نوع حداقل هزینه مسئله ۲- میانه پشتیبان^۱ (BI2MP) به روش تغییر در وزن رئوس، به دنبال آن هستیم تا با تغییر در وزن رئوس یک گراف و با کمترین هزینه، دو رأس از پیش تعیین شده (m_1, m_2) را بهینه نماییم. فرض کنید n تعداد رئوس گراف G باشد و c_i^+ هزینه متحمل شده از افزایش هر واحد وزن رأس v_i و c_i^- هزینه متحمل شده از کاهش هر واحد وزن رأس v_i باشند. همچنین p_i و q_i نیز به ترتیب میزان افزایش و کاهش وزن رئوس v_i یعنی w_i هستند. بنابراین هزینه متحمل شده از تغییر وزن رئوس که به دنبال کمینه شدن آن هستیم برابر با $\sum_{i=1}^n (c_i^+ p_i + c_i^- q_i)$ می باشد. از طرف دیگر

نیز زوج رأس (m_1, m_2) باید بهینه شوند. لذا باید برای هر $u_j, u_k \in V(G)$ نامساوی $F(m_1, m_2) \leq F(u_j, u_k)$ برقرار باشد. مقدار تابع هدف مسئله ۲- میانه پشتیبان می باشد که در رابطه (۳.۶) بیان شده است.

حال مسئله BI2MP با روش تغییر در وزن رئوس گراف به صورت زیر فرمول بندی می شود:

¹Backup Inverse 2 Median Problem

$$\min \sum_{i=1}^n (c_i^+ p_i + c_i^- q_i) \quad (۴.۶)$$

s.t.

$$\begin{aligned} (1 - \rho) \sum_{i=1}^n \hat{w}_i d(v_i, \{m_1, m_2\}) + \rho \sum_{i=1}^n \hat{w}_i d(v_i, m_1) + \rho \sum_{i=1}^n \hat{w}_i d(v_i, m_2) \leq \\ (1 - \rho) \sum_{i=1}^n \hat{w}_i d(v_i, \{u_j, u_k\}) + p \left(\sum_{i=1}^n \hat{w}_i d(v_i, u_j) + \sum_{i=1}^n \hat{w}_i d(v_i, u_k) \right), \quad \forall u_j, u_k \in V, \end{aligned} \quad (۵.۶)$$

$$\hat{w}_i = w_i + p_i - q_i, \quad i = 1, \dots, n, \quad (۶.۶)$$

$$p_i, q_i \geq 0, \quad i = 1, \dots, n, \quad (۷.۶)$$

مدل بالا یک مدل برنامه‌ریزی خطی و با تعداد محدودیت‌های از مرتبه $O(n^2)$ ، می‌باشد. مسئله بهینه‌سازی بالا، روی گراف‌های کلی، از نوع مسائل N -سخت است. لذا ما این مسئله را برای گراف‌های خاصی مانند درخت‌ها بررسی کرده‌ایم. در مسئله BI2MP، دو رأس (m_1, m_2) به‌عنوان جواب مسئله مشخص می‌باشند. لذا با توجه به لم ۱.۱.۳، m_1 و m_2 ، به‌ترتیب میانه‌های زیردرخت‌های $X(e)$ و $Y(e)$ می‌باشند. لذا طبق تعریف میانه بودن m_1 برای زیردرخت $X(e)$ ، داریم:

$$\begin{aligned} \sum_{v_i \in X(e) \setminus \{x\}} \hat{w}_i d(v_i, m_1) + (\hat{w}(x) + \rho \hat{w}(T_y)) d(x, m_1) \leq \sum_{v_i \in X(e) \setminus \{x\}} \hat{w}_i d(v_i, u_j) \\ + (\hat{w}(x) + \rho \hat{w}(T_y)) d(x, u_j), \quad \forall u_j \in V, \end{aligned} \quad (۸.۶)$$

به‌طور مشابه چون m_2 در زیردرخت $Y(e)$ میانه است لذا داریم:

$$\begin{aligned} \sum_{v_i \in Y(e) \setminus \{y\}} \hat{w}_i d(v_i, m_2) + (\hat{w}(y) + \rho \hat{w}(T - T_y)) d(x, m_2) \leq \sum_{v_i \in Y(e) \setminus \{y\}} \hat{w}_i d(v_i, u_k) \\ + (\hat{w}(y) + \rho \hat{w}(T - T_y)) d(y, u_k), \quad \forall u_k \in V, \end{aligned} \quad (۹.۶)$$

روابط (۸.۶) و (۹.۶) جایگزین رابطه (۵.۶) می‌شود. همچنین با جایگذاری کردن محدودیت (۶.۶) در روابط (۸.۶) و (۹.۶) و مقاداری عملیات ریاضی، مسئله BI2MP

را با تغییر در وزن رئوس درخت، به صورت زیر مدل بندی می شود:

$$\min \sum_{i=1}^n (c_i^+ p_i + c_i^- q_i) \quad (10.6)$$

s.t.

$$\sum_{v_i \in X(e)} (\gamma_{ij} p_i) + \rho \gamma_{xj} \sum_{v_i \in T_y} p_i - \sum_{v_i \in X(e)} (\gamma_{ij} q_j) - \rho \gamma_{xj} \sum_{v_i \in T_y} q_i \leq - \left(\sum_{v_i \in X(e)} (\gamma_{ij} w_i) + \rho \gamma_{xj} w(T_y) \right)_{\forall j \in n}, \quad (11.6)$$

$$\rho \gamma'_{yk} \sum_{v_i \in (T-T_y)} p_i + \sum_{v_i \in T_y} (\gamma'_{ik} p_i) - \rho \gamma'_{yk} \sum_{v_i \in (T-T_y)} q_i - \sum_{v_i \in T_y} (\gamma'_{ik} q_i) \leq - \left(\sum_{v_i \in T_y} (\gamma'_{ik} w_i) + \rho \gamma'_{yk} w(T-T_y) \right)_{\forall k \in n}, \quad (12.6)$$

$$p_i, q_i \geq 0, \quad i = 1, \dots, n. \quad (13.6)$$

در محدودیت های (۱۱.۶) و (۱۲.۶) تمامی ضرایب γ_{ij} ، γ_{xj} ، γ'_{ik} و γ'_{yk} مقادیر ثابتی هستند و داریم:

$$\begin{aligned} d(v_i, m_1) - d(v_i, u_i) &= \gamma_{ij}, \\ d(x, m_1) - d(x, u_i) &= \gamma_{xj}, \\ d(v_i, m_2) - d(v_i, u_k) &= \gamma'_{ik}, \\ d(y, m_2) - d(y, u_k) &= \gamma'_{yk}. \end{aligned} \quad (14.6)$$

بنابراین می توان این مسئله را به صورت زیر بازنویسی نمود:

$$\min \{ C^t X \mid AX \leq b, X \geq 0 \}. \quad (15.6)$$

که در آن:

$$\begin{cases} C = (c_1^+, c_1^-, \dots, c_n^+, c_n^-, \dots, c_n^+, c_n^-), \\ X^t = (p_1, p_2, \dots, p_n, q_1, q_2, \dots, q_n), \\ A = [B \mid -B]. \end{cases} \quad (16.6)$$

$$B = \begin{bmatrix} \left. \begin{matrix} \gamma_{i1} \\ \vdots \\ \gamma_{in} \end{matrix} \right\} & \left. \begin{matrix} \rho \gamma_{x1} \\ \vdots \\ \rho \gamma_{xn} \end{matrix} \right\} \\ \left. \begin{matrix} \rho \gamma'_{y1} \\ \vdots \\ \rho \gamma'_{yn} \end{matrix} \right\} & \left. \begin{matrix} \gamma'_{j1} \\ \vdots \\ \gamma'_{jn} \end{matrix} \right\} \end{bmatrix}. \quad (17.6)$$

بنابراین یک مسئله BI2MP با تغییر در وزن رئوس روی درخت را از یک مسئله برنامه‌ریزی خطی با تعداد محدودیت‌هایی از مرتبه $O(n^2)$ ، به یک مسئله برنامه‌ریزی خطی و حداکثر با $O(n) = 2n - 2$ محدودیت تبدیل کرده‌ایم.

۲.۶ معکوس نوع حداقل هزینه مسئله ۲- میانه پشتیبان با تغییر در طول یال

فرض کنید c_i^+ هزینه متحمل شده از افزایش هر واحد طول یال $e_i = (v_i, v_{i+1})$ و c_i^- هزینه متحمل شده از کاهش هر واحد طول یال e_i باشند. $x_{e_i}^+$ و $x_{e_i}^-$ به ترتیب میزان افزایش و کاهش طول یال e_i می‌باشند که l_{e_i} طول یال e_i است و $d^l(v_i, u_j)$ کوتاه‌ترین مسیر بین دو رأس (v_i, u_j) در گراف G می‌باشد. در این مسئله می‌خواهیم با کمترین هزینه ممکن از تغییر طول یال‌های گراف یعنی کمینه کردن $\sum_{i=1}^n (c_i^+ x_{e_i}^+ + c_i^- x_{e_i}^-)$ ، دو رأس دلخواه m_1 و m_2 ، تبدیل به ۲- میانه پشتیبان شوند. بنابراین مسئله BI2MP با تغییر در طول یال را به صورت زیر مدل می‌کنیم:

$$\min \sum_{i=1}^n (c_i^+ x_{e_i}^+ + c_i^- x_{e_i}^-) \quad (18.6)$$

s.t.

$$\begin{aligned} & (1 - \rho) \sum_{i=1}^n \hat{w}_i d^l(v_i, \{m_1, m_2\}) + \rho \sum_{i=1}^n \hat{w}_i d^l(v_i, m_1) + \rho \sum_{i=1}^n \hat{w}_i d^l(v_i, m_2) \\ & \leq (1 - \rho) \sum_{i=1}^n \hat{w}_i d^l(v_i, \{u_j, u_k\}) + \rho \left(\sum_{i=1}^n \hat{w}_i d^l(v_i, u_j) \sum_{i=1}^n \hat{w}_i d^l(v_i, u_k) \right), \quad \forall u_j, u_k \in V \end{aligned} \quad (19.6)$$

$$\hat{l}_i = l_i + x_{e_i}^+ - x_{e_i}^-, \quad \forall e_i \in E, \quad (20.6)$$

$$0 \leq x_{e_i}^+ \leq u_{e_i}^+, \quad \forall e_i \in E, \quad (21.6)$$

$$0 \leq x_{e_i}^- \leq u_{e_i}^-, \quad \forall e_i \in E. \quad (22.6)$$

مدل بالا یک مدل غیرخطی و با تعداد محدودیت‌های $O(n^2)$ است. مسئله بالا برای یک گراف کلی، از نوع یک مسئله NP -سخت است. ولی اگر گراف مورد نظر را یک درخت در نظر بگیریم می‌توان مدل بالا را به صورت یک مدل برنامه‌ریزی خطی بیان کرد.

فرض کنید $T = (V, E)$ ، یک درخت باشد که $V = \{v_1, \dots, v_n\}$ ، همچنین m_1 و m_2

دو رأسی باشند که می‌خواهیم بهینه شوند. از آنجایی که در معکوس مسئله ۲- میانه پشتیبان حاضر، m_1 و m_2 به‌عنوان جواب‌های مسئله از قبل مشخص هستند، بنابراین با توجه به لم ۱.۱.۳ می‌توان گفت، m_1 و m_2 به‌ترتیب میانه‌های زیر درخت‌های $X(e)$ و $Y(e)$ هستند. بنابراین با توجه به شرط میانه بودن m_1 برای زیر درخت $X(e)$ داریم:

$$\sum_{v_i \in X(e) \setminus \{x\}} w_i d^l(v_i, m_1) + (w(x) + \rho w(T_y)) d^l(x, m_1) \leq \sum_{v_i \in X(e) \setminus \{x\}} w_i d^l(v_i, u_j) + (w(x) + \rho w(T_y)) d^l(x, u_j), \forall u_j \in V, \quad (23.6)$$

از طرفی داریم:

$$\begin{cases} d^l(v_i, m_1) = d^l(v_i, m_1) + \sum_{e \in P(v_i, m_1)} (x_e^+ - x_e^-), \\ d^l(v_i, u_j) = d^l(v_i, u_j) + \sum_{e \in P(v_i, u_j)} (x_e^+ - x_e^-). \end{cases} \quad (24.6)$$

با قرار دادن (۲۴.۶) در شرط (۲۳.۶) داریم:

$$\sum_{v_i \in X(e)} w_i \left(\sum_{e_i \in P[v_i, m_1]} (x_{e_i}^+ - x_{e_i}^-) - \sum_{e_i \in P[v_i, u_j]} (x_{e_i}^+ - x_{e_i}^-) \right) + \rho w(T_y) \sum_{e_i \in P[x, m_1]} (x_{e_i}^+ - x_{e_i}^-) - \rho w(T_y) \sum_{e_i \in P[x, u_j]} (x_{e_i}^+ - x_{e_i}^-) \leq - \left(\sum_{v_i \in X(e)} w_i \gamma_{ij} + \rho w(T_y) \gamma_{xj} \right), \forall u_j \in V, \quad (25.6)$$

و به‌طور مشابه برای شرط میانه بودن m_2 ، در زیردرخت $Y(e)$ داریم:

$$\sum_{v_i \in Y(e)} w_i \left(\sum_{e_i \in P[v_i, m_2]} (x_{e_i}^+ - x_{e_i}^-) - \sum_{e_i \in P[v_i, u_k]} (x_{e_i}^+ - x_{e_i}^-) \right) + \rho w(T - T_y) \sum_{e_i \in P[y, m_2]} (x_{e_i}^+ - x_{e_i}^-) - \rho w(T_y) \sum_{e_i \in P[y, u_k]} (x_{e_i}^+ - x_{e_i}^-) \leq - \left(\sum_{v_i \in Y(e)} w_i \gamma'_{ik} + \rho w(T - T_y) \gamma'_{yk} \right), \forall u_k \in V, \quad (26.6)$$

که در آن:

$$\begin{cases} d^l(v_i, m_1) - d^l(v_i, u_j) = \gamma_{ij}, \\ d^l(x, m_1) - d^l(x, u_j) = \gamma_{xj}, \\ d^l(v_i, m_2) - d^l(v_i, u_k) = \gamma'_{ik}, \\ d^l(y, m_2) - d^l(y, u_k) = \gamma'_{yk}. \end{cases} \quad (27.6)$$

با قرار دادن (۲۵.۶) و (۲۶.۶) به‌جای محدودیت (۱۹.۶)، مسئله BI2MP با تغییر

طول یال به صورت زیر مدل بندی می‌شود:

$$\min \sum_{i=1}^n (c_i^+ x_{e_i}^+ + c_i^- x_{e_i}^-) \quad (28.6)$$

s.t.

$$\sum_{v_i \in X(e)} w_i \left(\sum_{e_i \in P[v_i, m_1]} (x_{e_i}^+ - x_{e_i}^-) - \sum_{e_i \in P[v_i, u_j]} (x_{e_i}^+ - x_{e_i}^-) \right) + \rho w(T_y) \sum_{e_i \in P[x, m_1]} (x_{e_i}^+ - x_{e_i}^-) - \rho w(T_y) \sum_{e_i \in P[x, u_j]} (x_{e_i}^+ - x_{e_i}^-) \leq - \left(\sum_{v_i \in X(e)} w_i \gamma_{ij} + \rho w(T_y) \gamma_{xj} \right), \forall u_j \in V, \quad (29.6)$$

$$\sum_{v_i \in X(e)} w_i \left(\sum_{e_i \in P[v_i, m_1]} (x_{e_i}^+ - x_{e_i}^-) - \sum_{e_i \in P[v_i, u_j]} (x_{e_i}^+ - x_{e_i}^-) \right) + \rho w(T_y) \sum_{e_i \in P[x, m_1]} (x_{e_i}^+ - x_{e_i}^-) - \rho w(T_y) \sum_{e_i \in P[x, u_j]} (x_{e_i}^+ - x_{e_i}^-) \leq - \left(\sum_{v_i \in Y(e)} w_i \gamma'_{ik} + \rho w(T - T_y) \gamma'_{yk} \right), \forall u_k \in V, \quad (30.6)$$

$$0 \leq x_{e_i}^+ \leq u_{e_i}^+, \forall e_i \in E(T), \quad (31.6)$$

$$0 \leq x_{e_i}^- \leq u_{e_i}^-, \forall e_i \in E(T), \quad (32.6)$$

مدل بالا یک مسئله بهینه‌سازی خطی روی یک مسیر و با حداکثر $2n = \mathcal{O}(n)$ محدودیت است. $u_{e_i}^-$ و $u_{e_i}^+$ نیز حداکثر میزان افزایش یا کاهش در طول یال هستند.

۳.۶ معکوس نوع محدودیت بودجه‌ای مسئله ۲- میانه

پشتیبان با تغییر در وزن رئوس

در این قسمت به بررسی معکوس نوع محدودیت بودجه‌ای مسئله ۲- میانه پشتیبان^۲ (BR2MP) با تغییر در وزن رئوس می‌پردازیم.

فرض کنید $B > 0$ ، بودجه مورد نظر برای تغییر پارامتر مسئله یعنی وزن رئوس (w_v) باشد، همچنین فرض کنید، m_1 و m_2 دو رأسی باشند که در یک مسئله پشتیبان، می‌خواهیم تبدیل به ۲- میانه شوند. یک مسئله معکوس از نوع محدودیت بودجه‌ای به دنبال این است که با بودجه‌ی $B > 0$ ، دو رأس دلخواه m_1 و m_2 را تا حد ممکن به بهینه شدن نزدیک کند. فاصله بین رأس v ، و رأس m_1 را با $d_1^k(v)$ ، و فاصله آن با رأس m_2 را با $d_2^k(v)$ ، نمایش می‌دهیم و p_v و q_v را به ترتیب میزان افزایش و کاهش وزن رأس

²Backup Reverse 2 Median Problem

v یعنی w_v تعریف می‌کنیم. همچنین هزینه متحمل شده از افزایش و کاهش وزن رأس v را به ترتیب با c_v^+ و c_v^- نمایش می‌دهیم، در این صورت مدل کلی مسئله BR2MP با تغییر در وزن رئوس به صورت زیر است:

$$\min(\lambda - \rho) \sum_{v \in V(G)} \min_{i=1,2} \{\hat{w}_v d_i^l(v)\} + \rho \left(\sum_{v \in V(G)} \hat{w}_v (d_1^l(v) + d_2^l(v)) \right) \quad (33.6)$$

s.t.

$$\hat{w}_v = w_v + p_v - q_v, \quad \forall v \in V(G), \quad (34.6)$$

$$\sum_{v \in V(G)} f_v(p_v, q_v) \leq B, \quad (35.6)$$

$$p_v, q_v \geq 0, \quad \forall v \in V(G). \quad (36.6)$$

تابع هدف (۳۳.۶) همان تابع هدف (۳.۶) برای یک مسئله ۲- میانه پشتیبان است که قرار است با تغییر در وزن رئوس یعنی تغییر w_v به \hat{w}_v تا حد امکان بهینه گردد. از طرف دیگر نیز مجاز هستیم تنها به اندازه بودجه B صرف تغییر وزن رئوس گراف کنیم. لذا داریم:

$$f_v(p_v, q_v) - c_v^+ p_v + c_v^- q_v = C \cdot (p_v, q_v), \quad (37.6)$$

که C یک بردار مثبت است. بدون اینکه به کلیت مسئله خللی وارد شود می‌توان فرض کرد $C = 1$. بنابراین محدودیت (۳۵.۶) را می‌توان به صورت زیر نیز بازنویسی نموده و به جای B با \tilde{B} کار کرد.

$$\sum_{v \in V(G)} p_v + q_v = \frac{B}{C} = \tilde{B}. \quad (38.6)$$

از آنجایی که m_1 و m_2 مقادیر مشخص هستند، بنابراین مقادیر $d_1^l(v)$ و $d_2^l(v)$ نیز برای هر $v \in V(G)$ نیز ثابت است. به وضوح مشخص است که تابع هدف (۳۳.۶) و محدودیت‌های (۳۴.۶) تا (۳۶.۶) خطی هستند. بنابراین با یک مسئله برنامه‌ریزی خطی روبرو هستیم.

۴.۶ معکوس نوع محدودیت بودجه‌ای مسئله ۲- میانه

پشتیبان با تغییر در طول یال

فرض کنید $B > 0$ بودجه مورد نظر برای تغییر پارامتر مسئله یعنی طول یال گراف باشد. همچنین فرض کنید m_1 و m_2 دو رأسی باشند که در یک مسئله پشتیبان

می‌خواهیم تبدیل به ۲- میانه شوند. به‌طور مشابه فاصله بین رأس v ، و رأس m_1 را با $d_1^l(v)$ و فاصله آن را با رأس m_2 را با $d_2^l(v)$ ، نمایش می‌دهیم. در این صورت مدل کلی مسئله BR2MP با تغییر در طول یال به‌صورت زیر است:

$$\min(\lambda - \rho) \sum_{v \in V(G)} \min_{i=1,2} \{w_v d_i^l(v)\} + \rho \left(\sum_{v \in V(G)} w_v (d_1^l(v) + d_2^l(v)) \right) \quad (39.6)$$

s.t.

$$\hat{l}_e = l_e + x_e, \quad \forall e \in E(G), \quad (40.6)$$

$$\sum_{v \in V(G)} f_e(x_e) \leq B, \quad (41.6)$$

$$0 \leq x_e \leq u_e, \quad \forall e \in E(G). \quad (42.6)$$

در روابط بالا x_e میزان تغییرات طول یالی است و u_e نیز حداکثر تغییرات طول یالی را بیان می‌کند. همچنین l_e ، طول یال $e = (v_i, v_{i+1})$ را نمایش می‌دهد. $\sum_{v \in V(G)} f_e(x_e)$ نیز تابع هزینه ناشی از تغییرات طول یال می‌باشد که می‌توان به‌صورت زیر در نظر گرفت:

$$\sum_{v \in V(G)} f_e(x_e) = \sum_{v \in V(G)} c_v x_e. \quad (43.6)$$

۵.۶ معکوس مسئله ۲- میانه پشتیبان در فضای پیوسته

با تغییر در مختصات نقاط

در این قسمت معکوس نوع محدودیت بودجه‌ای مسئله ۲- میانه پشتیبان (BR2MP) را به روش تغییر در مختصات نقاط مورد بررسی قرار می‌دهیم.

می‌خواهیم با تغییر در مختصات نقاط $X_i = (x_{i1}, \dots, x_{ik})$ ، دو نقطه دلخواه (از پیش تعیین شده) m_1 و m_2 را با استفاده از یک بودجه محدود تا حد امکان به حالت بهینه نزدیک کنیم. فرض کنید تغییر هر واحد مختصات نقاط X_i هزینه‌ی $C_i = (C_{i1}, \dots, C_{ik})$ را بر ما تحمیل کند، به‌طوری‌که هزینه متحمل شده از تغییر مختص $-t$ ام نقطه X_i است. به‌طور مشابه بردارهای $P_i^+ = (p_{i1}^+, \dots, p_{ik}^+)$ و $P_i^- = (p_{i1}^-, \dots, p_{ik}^-)$ را به ترتیب میزان افزایش و کاهش مختصات نقاط X_i تعریف می‌کنیم. بنابراین مسئله BR2MP همراه با تغییر در مختصات نقاط به‌صورت زیر مدل‌بندی

می شود:

Ω_1 :

$$\min(\lambda - \rho) \sum_{i=1}^n \left(w_i \min_{t=1,2} \{d(\hat{X}_i, m_t)\} \right) + \rho \sum_{i=1}^n \left(w_i d(\hat{X}_i, m_1) + d(\hat{X}_i, m_2) \right) \quad (44.6)$$

$$\text{s.t.} \quad (45.6)$$

$$X_i = X_i + P_i^+ - P_i^-, i = 1, \dots, n, \quad (46.6)$$

$$\sum_{i=1}^n f_d(P_i^+, P_i^-) \leq B, \quad (47.6)$$

$$P_i^+, P_i^- \geq 0, i = 1, \dots, n, \quad (48.6)$$

که $f_d(P_i^+, P_i^-)$ ، تابع هزینه ناشی از تغییر مختصات نقاط می باشد و به صورت زیر تعریف می شود:

$$f_d(P_i^+, P_i^-) = C_i^+ P_i^+ + C_i^- P_i^-. \quad (49.6)$$

مدل برنامه ریزی با تابع هدف (44.6) و محدودیت های (46.6) تا (48.6)، یک مسئله بهینه سازی غیرخطی است و به دلیل وجود عبارت $\min_{t=1,2} \{d(\hat{X}_i, m_t)\}$ در تابع هدف نامحدب است.

۱.۵.۶ معکوس نوع محدودیت بودجه ای مسئله ۲- میانه پشتیبان

در صفحه با نرم l_p .

ابرای نقاط (m_1, m_2) تعریف کنیم:

$$\prod(m_1, m_2) = (I_1, I_2). \quad (50.6)$$

به طوری که:

$$I_1 = \{i \mid d(X_i, m_1) \leq d(X_i, m_2)\}, \quad (51.6)$$

و

$$I_2 = \{1, 2, \dots, n\} \setminus I_1. \quad (52.6)$$

اگر $\xi = (\zeta_{i1}, \zeta_{in})$ میزان تغییرات مختصات نقاط $X_i = (x_{i1}, x_{i2})$ باشد، در این صورت با توجه به مجموعه های I_1 و I_2 ، مدل مسئله Ω_1 ، در صفحه به صورت زیر بازنویسی

می شود:

Ω_{γ} :

$$\begin{aligned} \min f(\zeta_{11}, \dots, \zeta_{1n}, \zeta_{12}, \dots, \zeta_{n2}) = & \\ & (1 - \rho) \sum_{i \in I_1} w_i (|x_i + \zeta_{i1} - a_1|^p + |y_i + \zeta_{i2} - b_1|^p)^{\frac{1}{p}} \\ & + (1 - \rho) \sum_{i \in I_{\gamma}} w_i (|x_i + \zeta_{i1} - a_{\gamma}|^p + |y_i + \zeta_{i2} - b_{\gamma}|^p)^{\frac{1}{p}} \\ & + \rho \sum_{i \in I_1 \cup I_{\gamma}} w_i (|x_i + \zeta_{i1} - a_1|^p + |y_i + \zeta_{i2} - b_1|^p)^{\frac{1}{p}} \\ & + \rho \left(\sum_{i \in I_1 \cup I_{\gamma}} w_i (|x_i + \zeta_{i1} - a_{\gamma}|^p + |y_i + \zeta_{i2} - b_{\gamma}|^p)^{\frac{1}{p}} \right) \end{aligned} \quad (53.6)$$

s.t.

$$\sum_{i=1}^n (\zeta_{i1} + \zeta_{i2}) \leq \beta, \quad (54.6)$$

$$\zeta_{i1}, \zeta_{i2} \in \mathbb{R}, \quad \forall i = 1, \dots, n. \quad (55.6)$$

حال دو لم زیر را می توان برای مدل فوق بیان کرد.

لم ۱.۵.۶. تابع هدف مسئله Ω_{γ} محدب است.

برهان. قرار می دهیم:

$$(\zeta_{i1}, a_1 - x_i, \zeta_{i2}, b_1 - y_i) = (s_1, s_2, s_3, s_4) = S$$

فرض کنید $f(S) = (|s_1 - s_2|^p + |s_3 - s_4|^p)^{\frac{1}{p}}$ و $0 < \lambda < 1$. در این صورت داریم:

$$\begin{aligned} f(\lambda S' + (1 - \lambda)S'') = & \\ & \left[|(\lambda s'_1 + (1 - \lambda)s''_1) - (\lambda s'_2 + (1 - \lambda)s''_2)|^p + |(\lambda s'_3 + (1 - \lambda)s''_3) - (\lambda s'_4 + (1 - \lambda)s''_4)|^p \right]^{\frac{1}{p}} \\ & = \left[|(\lambda(s'_1 - s'_2) + (1 - \lambda)(s''_1 - s''_2))|^p + |(\lambda(s'_3 - s'_4) + (1 - \lambda)(s''_3 - s''_4))|^p \right]^{\frac{1}{p}} \\ & \leq \left[|\lambda(s'_1 - s'_2)|^p + |\lambda(s'_3 - s'_4)|^p \right]^{\frac{1}{p}} + \left[|(1 - \lambda)(s''_1 - s''_2)|^p + |(1 - \lambda)(s''_3 - s''_4)|^p \right]^{\frac{1}{p}} \\ & = \lambda (|s'_1 - s'_2|^p + |s'_3 - s'_4|^p)^{\frac{1}{p}} + (1 - \lambda) (|s''_1 - s''_2|^p + |s''_3 - s''_4|^p)^{\frac{1}{p}} \\ & = \lambda f(S') + (1 - \lambda)f(S'') \end{aligned}$$

بنابراین تابع $f(S) = (|x_i + \zeta_{i1} - a_1|^p + |y_i + \zeta_{i2} - b_1|^p)^{\frac{1}{p}}$ محدب است.

لذا با توجه به مثبت بودن ضرائب w_i و ρ ، و با توجه به اینکه جمع تعداد شمارا تابع محدب، محدب می باشد، می توان نتیجه گرفت که عبارت:

$$(1 - \rho) \sum_{i \in I_1} w_i (|x_i + \zeta_{i1} - a_1|^p + |y_i + \zeta_{i2} - b_1|^p)^{\frac{1}{p}}$$

محدب است. به طور مشابه می توان ثابت نمود که قسمت های دیگر تابع هدف نیز محدب می باشند. بنابراین با توجه به اینکه جمع چند تابع محدب، محدب می باشد می توان نتیجه گرفت که تابع هدف مسئله Ω_2 نیز محدب است. \square

لم ۲.۵.۶. مشتقات تابع هدف مسئله Ω_2 ، در بعضی از نقاط، تعریف نشده است.

برهان. مشتقات تابع هدف در نقاطی که مخرج آن ها صفر می گردد، تعریف نشده هستند. \square

با توجه به لم ۲.۵.۶، برای آن که نقاط ناپیوستگی مشتقات جزئی تابع هدف مسئله Ω_2 را از بین ببریم، می توان مقدار ε ، که یک عدد کوچک و مثبت است را به رابطه (۵۳.۶) اضافه نمود تا مشتقات جزئی آن همواره پیوسته باشد. بنابراین تقریبی از تابع هدف مسئله Ω_2 ، به صورت زیر بدست می آید:

$$\begin{aligned} \Omega_2 : (BR2MPH) \\ \min f_h(\zeta_{11}, \dots, \zeta_{n1}, \zeta_{12}, \dots, \zeta_{n2}) = \\ (1 - \rho) \sum_{i \in I_1} w_i \left(\left((x_i + \zeta_{i1} - a_1)^2 + \varepsilon \right)^{\frac{p}{2}} + \left((y_i + \zeta_{i2} - b_1)^2 + \varepsilon \right)^{\frac{p}{2}} \right)^{\frac{1}{p}} \\ + (1 - \rho) \sum_{i \in I_2} w_i \left(\left((x_i + \zeta_{i1} - a_2)^2 + \varepsilon \right)^{\frac{p}{2}} + \left((y_i + \zeta_{i2} - b_2)^2 + \varepsilon \right)^{\frac{p}{2}} \right)^{\frac{1}{p}} \\ + \rho \sum_{i \in I_1 \cup I_2} w_i \left(\left((x_i + \zeta_{i1} - a_1)^2 + \varepsilon \right)^{\frac{p}{2}} + \left((y_i + \zeta_{i2} - b_1)^2 + \varepsilon \right)^{\frac{p}{2}} \right)^{\frac{1}{p}} \\ + \rho \sum_{i \in I_1 \cup I_2} w_i \left(\left((x_i + \zeta_{i1} - a_2)^2 + \varepsilon \right)^{\frac{p}{2}} + \left((y_i + \zeta_{i2} - b_2)^2 + \varepsilon \right)^{\frac{p}{2}} \right)^{\frac{1}{p}} \quad (56.6) \end{aligned}$$

وقتی $\varepsilon \rightarrow 0$ مقدار تابع هدف مسئله Ω_3 (BR2MPH)، با مقدار تابع هدف مسئله Ω_2 ، برابر می شود. این موضوع را می توان در لم ۳.۵.۶، مشاهده نمود.

لم ۳.۵.۶. وقتی $\varepsilon \rightarrow 0$ ، خطای حاصل از اضافه کردن مقدار ε به تابع هدف مسئله Ω_3 به صفر میل می کند و داریم:

$$\max\{f_h - f\} \leq M_\varepsilon = (1 + \rho) 2^{\frac{1}{p}} \varepsilon^{\frac{1}{p}} \left(\sum_{i=1}^n w_i \right). \quad (57.6)$$

برهان. قرار می دهیم، $s_1 = (x_i + \zeta_{i1} - a_1)$ ، $s_2 = (y_i + \zeta_{i2} - b_1)$ و $s_3 = \varepsilon^{1/p}$ باشد بنابراین داریم

$$\begin{aligned} & \left(\left((x_i + \zeta_{i1} - a_1)^2 + \varepsilon \right)^{\frac{p}{2}} + \left((y_i + \zeta_{i2} - b_1)^2 + \varepsilon \right)^{\frac{p}{2}} \right)^{\frac{1}{p}} = \left((s_1^2 + s_3^2)^{\frac{p}{2}} + (s_2^2 + s_3^2)^{\frac{p}{2}} \right)^{\frac{1}{p}} \\ & \leq \left((s_1 + s_3)^2 \right)^{\frac{p}{2}} + \left((s_2 + s_3)^2 \right)^{\frac{p}{2}} = ((s_1 + s_3)^p + (s_2 + s_3)^p)^{\frac{1}{p}} \end{aligned} \quad (58.6)$$

همچنین با استفاده از نامساوی مینکوفسکی می توان گفت

$$\begin{aligned} & ((s_1 + s_3)^p + (s_2 + s_3)^p)^{\frac{1}{p}} \leq (s_1^p + s_3^p)^{\frac{1}{p}} + (s_2^p + s_3^p)^{\frac{1}{p}} = (s_1^p + s_3^p)^{\frac{1}{p}} + 2^{\frac{1}{p}} \varepsilon^{\frac{1}{p}} \\ & = ((x_i + \zeta_{i1} - a_1)^p + (y_i + \zeta_{i2} - b_1)^p)^{\frac{1}{p}} + 2^{\frac{1}{p}} \varepsilon^{\frac{1}{p}} \end{aligned} \quad (59.6)$$

بنابراین داریم

$$\begin{aligned} & \left(\left((x_i + \zeta_{i1} - a_1)^2 + \varepsilon \right)^{\frac{p}{2}} + \left((y_i + \zeta_{i2} - b_1)^2 + \varepsilon \right)^{\frac{p}{2}} \right)^{\frac{1}{p}} \\ & - ((x_i + \zeta_{i1} - a_1)^p + (y_i + \zeta_{i2} - b_1)^p)^{\frac{1}{p}} \leq 2^{\frac{1}{p}} \varepsilon^{\frac{1}{p}} \end{aligned}$$

بنابراین نتیجه می شود

$$\begin{aligned} & \sum_{i \in I_1} w_i \left[\left(\left((x_i + \zeta_{i1} - a_1)^2 + \varepsilon \right)^{\frac{p}{2}} + \left((y_i + \zeta_{i2} - b_1)^2 + \varepsilon \right)^{\frac{p}{2}} \right)^{\frac{1}{p}} - \right. \\ & \left. ((x_i + \zeta_{i1} - a_1)^p + (y_i + \zeta_{i2} - b_1)^p)^{\frac{1}{p}} \right] \leq 2^{\frac{1}{p}} \varepsilon^{\frac{1}{p}} \sum_{i \in I_1} w_i. \end{aligned} \quad (60.6)$$

به طور مشابه می توان این اثبات را برای قسمت های دیگر تابع هدف (۵۶.۶) نیز بیان کرد. بنابراین داریم

$$\begin{aligned} f_h - f & \leq (1 - \rho) 2^{\frac{1}{p}} \varepsilon^{\frac{1}{p}} \sum_{i \in I_1} w_i + (1 - \rho) 2^{\frac{1}{p}} \varepsilon^{\frac{1}{p}} \sum_{i \in I_2} w_i + 2^{\frac{1}{p}} \varepsilon^{\frac{1}{p}} \rho \sum_{i \in I_1 \cup I_2} w_i + 2^{\frac{1}{p}} \varepsilon^{\frac{1}{p}} \rho \sum_{i \in I_1 \cup I_2} w_i \\ & = (1 + \rho) 2^{\frac{1}{p}} \varepsilon^{\frac{1}{p}} \left(\sum_{i \in I_1} w_i + \sum_{i \in I_2} w_i \right) = (1 + \rho) 2^{\frac{1}{p}} \varepsilon^{\frac{1}{p}} \sum_{i \in I_1 \cup I_2} w_i. \end{aligned} \quad (61.6)$$

□ در رابطه (۶۱.۶) وقتی $\varepsilon \rightarrow 0$ در این صورت خطا به صفر میل می کند.

در حالت خاصی که بودجه مورد نظر به اندازه کافی بزرگ باشد، تا بتوان به هر اندازه دلخواه نقاط را به سرویس دهنده‌ها نزدیک نمود (یعنی، $B \rightarrow \infty$)، می‌توان با استفاده از شرط لازم بهینگی و لم ۱.۵.۶ برای بهینه شدن مسئله Ω_3 ، یک الگوریتم بازگشتی ارائه نمود. لذا داریم:

$$\begin{aligned} \frac{\partial f_h}{\partial \zeta_{i1}} &= \frac{w_i(x_i + \zeta_{i1} - a_1)}{DxSx_{a1}} \Big|_{i \in I_1} + \rho \frac{w_i(x_i + \zeta_{i1} - a_1)}{DxSx_{a1}} \Big|_{i \in I_2} \\ &+ \frac{w_i(x_i + \zeta_{i1} - a_2)}{D'xSx_{a2}} \Big|_{i \in I_2} + \rho \frac{w_i(x_i + \zeta_{i1} - a_2)}{D'xSx_{a2}} \Big|_{i \in I_1}, \end{aligned} \quad (۶۲.۶)$$

و به‌طور مشابه:

$$\begin{aligned} \frac{\partial f_h}{\partial \zeta_{i2}} &= \frac{w_i(y_i + \zeta_{i2} - b_1)}{DxSy_{b1}} \Big|_{i \in I_1} + \rho \frac{w_i(y_i + \zeta_{i2} - b_1)}{DxSy_{b1}} \Big|_{i \in I_2} \\ &+ \frac{w_i(y_i + \zeta_{i2} - b_2)}{D'xSy_{b2}} \Big|_{i \in I_2} + \rho \frac{w_i(y_i + \zeta_{i2} - b_2)}{D'xSy_{b2}} \Big|_{i \in I_1}, \end{aligned} \quad (۶۳.۶)$$

از مساوی با صفر قرار دادن روابط (۶۲.۶) و (۶۳.۶) نتیجه می‌شود:

$$\Psi_{i1}^{k+1} = \begin{cases} \frac{\frac{w_i \alpha_{i1}}{Dx^k Sx_{a1}^k} + \frac{\rho w_i \alpha_{i2}}{D'x^k Sx_{a2}^k}}{\frac{w_i}{Dx^k Sx_{a1}^k} + \frac{\rho w_i}{D'x^k Sx_{a2}^k}}, & i \in I_1, \\ \frac{\frac{\rho w_i \alpha_{i1}}{Dx^k Sx_{a1}^k} + \frac{w_i \alpha_{i2}}{D'x^k Sx_{a2}^k}}{\frac{\rho w_i}{Dx^k Sx_{a1}^k} + \frac{w_i}{D'x^k Sx_{a2}^k}}, & i \in I_2, \end{cases} \quad (۶۴.۶)$$

و

$$\Psi_{i2}^{k+1} = \begin{cases} \frac{\frac{w_i \alpha'_{i1}}{Dx^k Sy_{b1}^k} + \frac{\rho w_i \alpha'_{i2}}{D'x^k Sy_{b2}^k}}{\frac{w_i}{Dx^k Sy_{b1}^k} + \frac{\rho w_i}{D'x^k Sy_{b2}^k}}, & i \in I_1, \\ \frac{\frac{\rho w_i \alpha'_{i1}}{Dx^k Sy_{b1}^k} + \frac{w_i \alpha'_{i2}}{D'x^k Sy_{b2}^k}}{\frac{\rho w_i}{Dx^k Sy_{b1}^k} + \frac{w_i}{D'x^k Sy_{b2}^k}}, & i \in I_2, \end{cases} \quad (۶۵.۶)$$

که:

$$\begin{cases} \alpha_{i1} = a_1 - x_i, \\ \alpha_{i2} = a_2 - x_i, \\ \alpha'_{i1} = b_1 - y_i, \\ \alpha'_{i2} = b_2 - y_i. \end{cases} \quad (۶۶.۶)$$

$$\begin{aligned}
 Dx &= \left(\left((x_i + \zeta_{i1} - a_1)^2 + \varepsilon \right)^{\frac{p}{q}} + \left((y_i + \zeta_{i2} - b_1)^2 + \varepsilon \right)^{\frac{p}{q}} \right)^{\frac{p-1}{p}}, \\
 D'x &= \left(\left((x_i + \zeta_{i1} - a_2)^2 + \varepsilon \right)^{\frac{p}{q}} + \left((y_i + \zeta_{i2} - b_2)^2 + \varepsilon \right)^{\frac{p}{q}} \right)^{\frac{p-1}{p}}, \\
 Sx_{a_1} &= \left(\left((x_i + \zeta_{i1} - a_1)^2 + \varepsilon \right)^{\frac{p}{q}} \right)^{\frac{2-p}{p}}, \\
 Sx_{a_2} &= \left(\left((x_i + \zeta_{i1} - a_2)^2 + \varepsilon \right)^{\frac{p}{q}} \right)^{\frac{2-p}{p}}, \\
 Sy_{b_1} &= \left(\left((y_i + \zeta_{i2} - b_1)^2 + \varepsilon \right)^{\frac{p}{q}} \right)^{\frac{2-p}{p}}, \\
 Sy_{b_2} &= \left(\left((y_i + \zeta_{i2} - b_2)^2 + \varepsilon \right)^{\frac{p}{q}} \right)^{\frac{2-p}{p}}.
 \end{aligned} \tag{۶۷.۶}$$

بنابراین وقتی که بودجه مورد نظر به اندازه کافی بزرگ باشد می‌توان از روابط بازگشتی (۶۴.۶) و (۶۵.۶) استفاده نمود. ما در اینجا وقتی بودجه مورد نظر محدود باشد مسئله Ω_2 را با الگوریتم‌های فراابتکاری حل نموده و نتایج را با هم مقایسه نموده‌ایم.

۶.۶ الگوریتم‌های فراابتکاری

الگوریتم‌های بهینه‌سازی، مقادیر متغیرهای طراحی را به گونه‌ای تغییر می‌دهند تا تابع هدف مورد نظر کمینه و یا بیشینه شود. روش‌های بهینه‌سازی را می‌توان به‌طور کلی به دو گروه روش‌هایی گرادینانی و روش‌های جستجو تقسیم نمود. روش‌های جستجو برای مسائل حجیم و در مواردی که مشتق گرفتن از تابع هدف دشوار است، کارایی زیادی دارند. همچنین نمی‌توان به صراحت در مورد برتری روش‌های تکاملی فوق نسبت به یکدیگر نظر داد و این امر طبق نظریه NFLT^۳ ([۸۳])، از مسئله‌ای به مسئله دیگر متفاوت است. در این فصل که با مسئله‌ای NP-سخت ([۱۰۳])، و به شدت غیرخطی همراه با قیود فراوان سروکار داریم، از روش‌های بهینه‌سازی پرندگان^۴ (PSO)، الگوریتم بهبودیافته بهینه‌سازی پرندگان^۵ (IPSO)، و الگوریتم فراابتکاری فرهنگی^۶ (CA) استفاده شده است.

برای حل یک مسئله به کمک این‌گونه الگوریتم‌ها، از یک مکانیسم تکرار ساده بهره گرفته می‌شود. این الگوریتم‌ها با یک جمعیت تصادفی اولیه از پاسخها شروع،

^۳No Free Lunch Theorem (NFLT)

^۴Particle Swarm Optimization

^۵Improve Particle Swarm Optimization (IPSO)

^۶Cultural algorithm

و در هر نسل تلاش می‌کنند تا نتایج را بهبود ببخشند. با الهام‌گیری از پدیده‌های مختلف طبیعی، الگوریتم‌های ابتکاری و فراابتکاری بسیاری از قبیل الگوریتم ژنتیک، مورچگان، زنبور عسل مصنوعی، حرکت ذرات و غیره معرفی شده‌اند. می‌توان مراحل عمومی این الگوریتم‌ها را به صورت زیر بیان نمود:

- (۱) ایجاد جمعیت تصادفی پاسخ اولیه.
- (۲) ارزیابی، مقایسه، رتبه‌بندی پاسخ‌ها و انتخاب بهترین‌ها.
- (۳) ترکیب جمعیت جدید با استفاده از فرایندهای طبیعی و باز تولید و ادغام با جمعیت قبل.
- (۴) بازگشت به مرحله (۲)، در صورت محقق نشدن شرایط مسئله.
- (۵) خروج در صورت محقق شدن شرایط مسئله.

۱.۶.۶ الگوریتم بهینه‌سازی پرندگان

PSO و IPSO از جمله روش‌های تکاملی و مدرن هستند که در سال ۱۹۹۵ توسط کندی و ابرهات با الهام از رفتار دسته جمعی موجوداتی مثل حشرات، زنبورها، مورچگان و پرندگان ایجاد شده است. هر عضو این گروه براساس اطلاعات و آگاهی خود و اطلاعات گروه حرکت می‌کند. این روش را می‌توان براساس گروه‌هایی از اعضا در نظر گرفت که به دنبال بهترین مقدار برای تابع هدف حرکت می‌کنند. هر کدام از این اعضا دارای دو مشخصه موقعیت و سرعت می‌باشند که دائماً تغییر کرده و اصلاح می‌شود. هر عضو در فضای طراحی مسئله گردش می‌کند و به دنبال نقطه بهینه می‌گردد، از سوی دیگر هر عضو بهترین موقعیت خود را نیز در نظر گرفته و در حافظه خود نگه می‌دارد. رد و بدل شدن اطلاعات میان این اعضا براساس بهترین نقاط برای هر عضو و بهترین نقطه تمام اعضا، موجب اصلاح مداوم موقعیت و سرعت هر عضو بر طبق روابط زیر می‌شود [۱۲۰]:

$$v_{ij}(t+1) = \omega_i v_{ij}(t) + c_1 r_{1j} [y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t) [\bar{y}_j(t) - x_{ij}(t)], \quad (68.6)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t), \quad (69.6)$$

$$y_i(t+1) = \begin{cases} y_i, & \text{اگر } f(x_i(t+1)) \geq f(y_i(t)), \\ x_i(t+1), & \text{اگر } f(x_i(t+1)) < f(y_i(t)), \end{cases} \quad (70.6)$$

$x_i(t)$ ، موقعیت فعلی ذره، v_i ، سرعت حرکت ذره، y_i ، بهترین موقعیتی که ذره تا به حال تجربه کرده است، \bar{y}_i ، بهترین مکانی است که تا کنون توسط ذرات مجاور یافت شده است.

c_1 ، ضریب یادگیری مربوط به تجارب شخصی هر ذره است که در این قسمت $c_1 = 2.02$ در نظر گرفته شده است. در مقابل c_2 ، ضریب یادگیری برای کل جامعه می‌باشد

که در اینجا $\alpha_2 = 2.02$ می‌باشد. از طرف دیگر r_1 و r_2 ، باعث می‌شود که نوع گوناگونی در جواب‌ها به وجود بیاید و به این شکل جستجوی کامل‌تری روی فضا انجام گیرد [۱۲۰]. در این قسمت از رساله در هر تکرار به صورت یک عدد تصادفی از توزیع نرمال در بازه $(0, 1)$ تولید شده و در هر تکرار به روز رسانی می‌شوند. بنابراین همان‌طور که مشاهده می‌شود موقعیت و سرعت هر ذره در هر مولفه برای $j = 1, 2, \dots, n$ ، به‌طور جداگانه به روز رسانی خواهد شد.

ω_i ، ضریب وزنی می‌باشد که به‌طور خطی از ω_{\max} تا ω_{\min} تغییر می‌کند. مقدار بالای ضریب وزنی، جستجوی کلی، و مقدار پایین آن، جستجوی محلی را برای تعیین نقطه بهینه انجام می‌دهد. بنابراین در طی مراحل تکرار روش برای یافتن نقطه بهینه، این ضریب به صورت خطی و طبق رابطه زیر کاهش می‌یابد تا این که در آخرین مرحله به کمترین مقدار خود می‌رسد [۱۲۰].

$$\omega_t = \omega_{\max} - \left(\frac{\omega_{\max} - \omega_{\min}}{t_{\max}} \right) \cdot t. \quad (71.6)$$

در اینجا $t_{\max} = 1000$ ، ماکزیمم مراحل تعیین شده برای تکرار مسئله، ω_{\max} و ω_{\min} ، به ترتیب برابر با ۰.۲ و ۰.۹، در نظر گرفته شده‌اند. اگر همه اعضا به یک نقطه نزدیک شوند، و یا فاصله اعضا از یکدیگر تا مقدار تعیین شده‌ای کاهش یابد، حل مسئله همگرا شده و نقطه بهینه به دست می‌آید.

در روش IPSO، ضریب وزنی ω_t ، به صورت زیر می‌باشد ([۴۲]:

$$\omega_t = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \left(\frac{\text{dist}_j}{\text{max}_{\text{dist}}} \right). \quad (72.6)$$

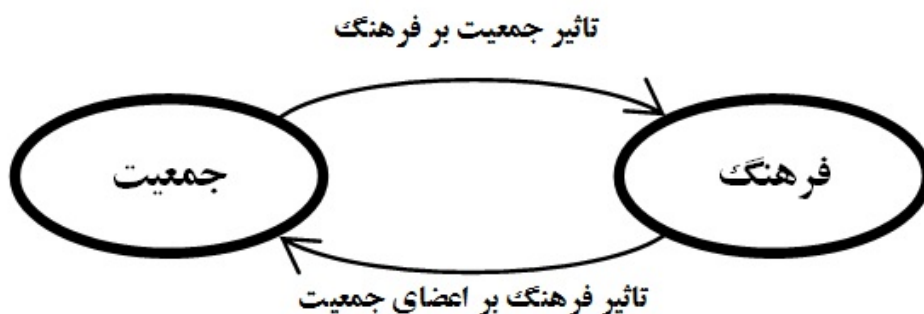
که، dist_j فاصله اقلیدسی عضو j -ام از بهترین نقطه \bar{y}_i و max_{dist} ، نیز بیشترین فاصله تا \bar{y}_i در بین همه اعضا می‌باشد [۴۲]. این مقادیر با توجه به مشخص بودن نقاط و بهترین جواب در هر تکرار، قابل محاسبه می‌باشند.

۲.۶.۶ الگوریتم فراابتکاری فرهنگی

الگوریتم‌های فرهنگی توسط رینولدز [۴۸]، مطرح گردید. این الگوریتم‌ها، تقابل فرهنگی و تاثیر فضای فرهنگ و فضای اجتماعی را با هم مورد بررسی قرار می‌دهند. تقابل و تعامل فرهنگ و اجتماع، منجر به ارائه یک راه حل برای حل مسائل بهینه‌سازی می‌شود. ایده اصلی الگوریتم‌های فرهنگی از تکامل فرهنگی گرفته شده است. در چگونگی تکامل و نقش افراد در اجتماع، در تولید و تغییر فرهنگ، نخبگان و افراد مشهور جامعه با وزن بیشتر، نقش و تاثیر مهمتر و بیشتری به‌طور مستقیم یا غیر مستقیم بر جامعه و فرهنگ خواهند داشت. به‌عنوان مثال، افراد ثروتمند جامعه با تکیه بر تمول مالی می‌توانند آداب و رسوم جاری جامعه (فضای باور جامعه) را تغییر

داده و با تبعیت سایر افراد جامعه (فضای جمعیت) از آن‌ها، رفته رفته به فرهنگ تبدیل شوند.

به‌طور کلی یک الگوریتم فرهنگی یک سیستم دوارثی می‌باشد که مبنای آن جستجو در فضای جمعیت (به نمایندگی از یک جزء ژنتیکی بر اساس نظریه تکاملی داروین) و فضای باور / فرهنگ (به نمایندگی از یک جزء فرهنگی) است. این فضای باور است که این الگوریتم را با الگوریتم‌هایی نظیر ژنتیک و سایر الگوریتم‌های تکاملی متمایز می‌کند. می‌توان گفت، فضای جمعیت اطلاعاتی را در مورد رفتارهای شخصی و ژنتیکی / فنوتیپی جامعه می‌دهد در حالی که فضای باور اطلاعات فرهنگی افراد جامعه را بیان می‌کند [۴۸]. هر دو فضا به‌صورت موازی با هم کار می‌کنند و دائماً در حال تاثیر بر یکدیگر هستند (شکل ۱.۶ را ببینید). این چرخه آن‌قدر ادامه می‌یابد تا فرهنگ به بهترین فرهنگ و اعضای جمعیت به بهترین اعضای جمعیت همگرا شوند. از طرف دیگر تغییراتی که در خود فرهنگ انجام می‌شود باعث می‌گردد تا فضای اعتقادی جامعه نیز جابجا گردد.



شکل ۱.۶: تاثیر فرهنگ بر جامعه و بالعکس.

مشخصه‌های الگوریتم فرهنگی

ما در این قسمت مشخصه‌های الگوریتم فرهنگی حاضر را بیان می‌کنیم. به‌طور کلی یک الگوریتم فرهنگی، در هر نسل ابتدا افراد را مانند الگوریتم ژنتیک در فضای جمعیت وارد کرده و توسط تابع شایستگی ارزیابی می‌کند. سپس توسط تابع پذیرش افرادی را که مناسب شکل دادن به فضای باور / فرهنگی هستند را انتخاب می‌کند و تجربیات پذیرفته شده افراد را برای ساختن و تغییر فضای باور / فرهنگ به کار می‌برد (در اینجا فرهنگ شبیه‌سازی می‌شود). فرهنگ ایجاد شده در فضای باور، بر روی تکامل جمعیت در فضای جمعیت تأثیر می‌گذارد. این تأثیر با تغییر دادن عملگر جهش و اعمال آن عملگر، در تولید فرزندان صورت می‌گیرد.

بنابراین یک الگوریتم فرهنگی دارای مشخصه‌های مختلفی همچون فضای جمعیت،

فضای باور / فرهنگ، تابع پذیرش، تنظیم اعتقادات / تابع تاثیر و انتخاب و ارزیابی می‌باشد.

● فضای جمعیت

این فضا در واقع فضای اصلی جمعیت می‌باشد و با مقدار دهی اولیه کار خود را شروع کرده و استخراج فرهنگ و ذخیره‌ی آن در فضای باور / فرهنگ در این قسمت انجام می‌گیرد. در این فصل، نقاط اولیه همان فضای اولیه مسئله می‌باشند که در هر تکرار تغییر می‌کنند. مولفه جمعیت یک الگوریتم فرهنگی، مشابه همین مولفه در یک الگوریتم ژنتیک است.

● ارزیابی و انتخاب

در هر تکرار باید اعضای جمعیت که در فضای جمعیت آن تکرار قرار دارند توسط تابعی ارزیابی شوند تا مشخص شود کدام یک از این اعضا نسبت به اعضای دیگر برتری دارد. ملاک برتری، کاهش دادن بیشتر تابع هدف مسئله است. بنابراین تابع ارزیابی را می‌توان همان تابع هدف مسئله معین نمود. این تابع ارزیابی، معیاری است برای انتخاب اعضایی که بیشترین کاهش تابع هدف را در پی دارند.

● فضای باور

فضای باور یک الگوریتم فرهنگی به بخش‌های مختلفی تقسیم می‌شود. این بخش‌ها نماینده حوزه‌های مختلف دانشی هستند که، جمعیت از فضای جستجو دارد. در فضای باور، تجربیات عمومی شده افراد موفق از فضای جمعیتی، به دست آمده و این تجارب در سراسر نسل و نسل‌های بعدی شکل گرفته و ذخیره می‌شود. لذا در واقع اطلاعات فرهنگ افراد در فضای باور مدل می‌شود. این تجارب بر تمامی نسل‌ها تأثیرگذار است و به نسل‌های آینده منتقل می‌گردد. در واقع این فضا برای هرس کردن فضای جمعیت مؤثر است.

هر فرد یک ذره در فضای جستجو است که فضای باور برای دور ساختن افراد از ناحیه‌های نامطلوب و سوق دادن آن‌ها به سمت ناحیه‌های امید بخش و نزدیک به جواب، به کار برده می‌شود. دانش‌های مختلفی نظیر دانش موقعیتی (وضعی)، دانش هنجار (معیار)، دانش توپوگرافی (مؤلفه ریختی)، دانش تاریخچه و دانش بازه برای تشکیل فضای باور تشکیل شده‌اند. در این فصل از دو دانش موقعیتی و دانش هنجاری برای تغییر فضای فرهنگ استفاده خواهیم کرد. فرض کنید $B(t)$ ، فضای باور در هر نسل باشد، داریم:

$$B(t) = (S(t), N(t)).$$

به طوریکه، $S(t)$ و $N(t)$ ، به ترتیب مولفه‌های موقعیتی (وضعی) و هنجاری فضای باور در نسل t -ام نامیده می‌شوند.

• دانش وضعی

بهترین جواب‌های هر نسل (و یا نسل جاری) را به عنوان مؤلفه وضعی در نظر می‌گیریم، نقطه‌ای که در فرهنگ به عنوان هدف در نظر گرفته می‌شود. در اینجا دانش موقعیتی (وضعی) شامل بهترین ذره سراسری است.

این دانش طبق رابطه زیر به روز می‌شود:

$$S(t) = y(t+1) = \begin{cases} \text{arc}\{f(x_i(t))\} & \text{اگر } (\min_{i=1, \dots, n_s} \{f(x_i(t))\} < f(y(t))), \\ y(t), & \text{در غیر این صورت} \end{cases} \quad (۷۳.۶)$$

که، $x_i(t)$ جواب مسئله در هر تکرار می‌باشد.

• دانش هنجاری

این منبع دانش، مجموعه بازه‌های خوب و امیدبخش را که از مجموعه‌ای از ذرات خوب استخراج شده است، برای هر بعد از مسأله نگهداری می‌کند. اگر فضا رقابتی باشد هنجارها شکل مناسبی به خود می‌گیرند و در صورت مراجعه به هنجارها، نهایتاً بهبود جواب‌ها حاصل می‌شود.

بنابراین اگر فرض کنیم پاسخ‌ها دارای چند مولفه باشند یعنی، $x = (x_1, \dots, x_D)$ ، که D ، معرف تعداد ابعاد مسئله است، برای هر یک از آن‌ها یک بازه تعریف می‌شود که به آن بازه هنجاری می‌گویند. این بازه، افراد موجه یا پذیرش شده را در فضای باور / فرهنگی، توصیف می‌کند. لذا برای هر متغیر یک فضای موجه و پذیرش شده داریم. مولفه هنجاری را با N ، نمایش می‌دهیم و به صورت $N = \{I_1, I_2, \dots, I_D\}$ ، بیان می‌کنیم. که در آن $I_i = (x_i^{\min}, x_i^{\max})$ و $x_i = (I_i, L_i, U_i)$ می‌باشد که، L_i ، با x_i^{\min} و U_i ، با x_i^{\max} ، مرتبط می‌باشند و L_i و U_i ، مقادیر تابع شایستگی در آن حدود می‌باشند. به عبارت دیگر L_i و U_i ، به ترتیب حد پایین مقدار یک ذره در تابع هدف و حد بالای یک ذره از در تابع هدف در بعد i -ام می‌باشد. در هر تکرار طبق این دانش فضای جستجو رفته رفته کوچکتر شده و به ناحیه‌های خوب نزدیکتر می‌شود.

• تابع پذیرش

تابع پذیرش تابعی است یک عضو از جمعیت را برای تاثیر بر روی فرهنگ انتخاب می‌کند، لذا این تابع مشخص می‌کند که چه افرادی می‌توانند بر فرهنگ تاثیرگذار باشند. افراد انتخاب شده توسط تابع پذیرش در هر تکرار فضای تابع پذیرش

را شکل می‌دهند. تابع پذیرش به دو روش استاتیک و دینامیک پیاده‌سازی می‌شود. در روش‌های استاتیک، معمولاً $n\%$ بهتر جمعیت انتخاب می‌شوند. در حالیکه در روش دینامیک، تعداد افراد موثر ثابت نبوده و قابل تغییر هستند، در این قسمت بر اساس منبع [۱۲۳]، از روش دینامیک و از رابطه‌ی (۷۴.۶) برای تابع پذیرش استفاده شده است.

$$n_B(t) = \frac{n_s \cdot \gamma}{t}. \quad (74.6)$$

به طوریکه n_s اندازه جمعیت است که در این فصل برای مثال ارائه شده $n_s = 50$ انتخاب شده است و γ ضریبی تجربی بین صفر و یک است که در این فصل $\gamma = 0.3$ فرض شده است. و t شمارنده تکرار است. که با افزایش تعداد تکرار $n_B(t)$ رو به کاهش می‌رود.

• تابع تاثیر / تنظیم اعتقادات

باورها در فضای باور برای تغییر دادن افراد و نزدیک کردن آن‌ها به باور سراسری (بهینه کل) به کار برده می‌شوند که این تغییرات با استفاده از تابع تاثیر تحقق می‌یابد [۴۸].

فضای باور با استفاده از عملگر جهش بر روی فضای جمعیت تاثیر می‌گذارد، این تاثیر از دو راه ممکن است یکی اندازه جهش و دیگری جهت جهش می‌باشد. در این فصل سعی شده است با توجه به مولفه وضعی و مولفه هنجاری از مرجع [۴۸]، تابع تاثیر استخراج شود. بنابراین در یک الگوریتم فرهنگی، برای اعمال تغییر در پاسخ‌ها می‌توانیم به روش‌های زیر عمل کنیم:

- ۱) استفاده از مولفه هنجاری.
 - ۲) استفاده از مولفه وضعی برای تعیین جهت حرکت.
 - ۳) استفاده از مولفه وضعی برای تعیین جهت و استفاده از مولفه هنجاری برای میزان حرکت.
 - ۴) استفاده از مولفه‌های هنجاری هم برای گام حرکت و هم جهت حرکت.
- در این فصل از روش چهارم برای تغییر در پاسخ‌ها استفاده شده است یعنی در هر تکرار داریم:

$$x'_{ij}(t) = \begin{cases} x_{ij}(t) + \delta_j \cdot |N(\circ, 1)|, & x_{ij}(t) \leq x_i^{\min}(t), \\ x_{ij}(t) - \delta_j \cdot |N(\circ, 1)|, & x_{ij}(t) \geq x_i^{\max}(t), \\ \text{در غیر این صورت} & \end{cases} \quad (75.6)$$

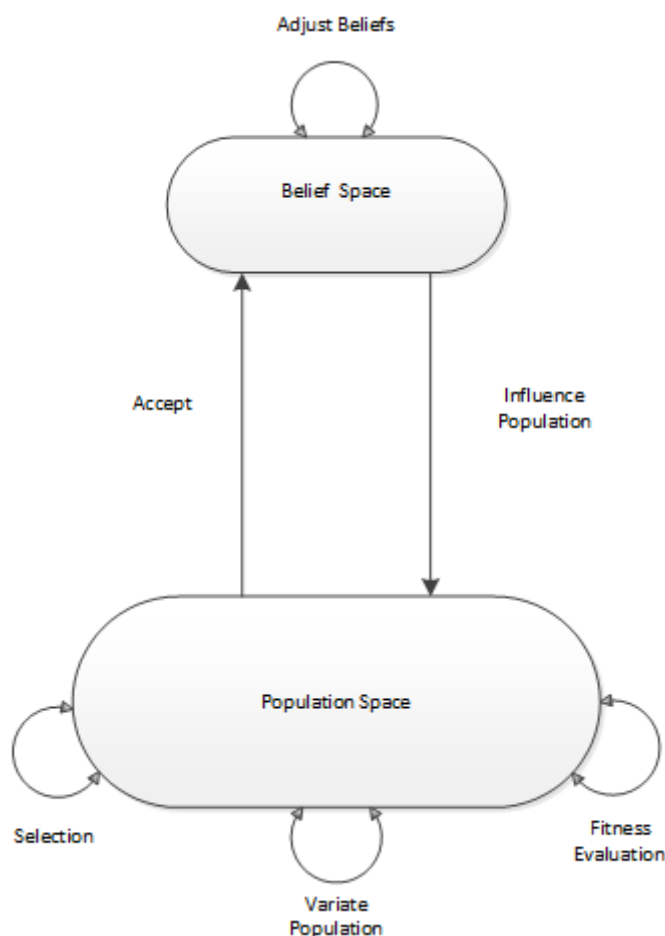
که در آن θ ضریب مقیاس‌گذاری است و برای آن مقداری برابر با $\theta = 0.5$ در نظر گرفته‌ایم. همچنین مقدار حرکت با $\delta_j \geq 0$ نمایش داده شده است. در این فصل برای

تعیین δ_j از طول بازه استفاده کرده‌ایم یعنی:

$$\delta_j(t) = \alpha[x_i^{\min}(t) - x_i^{\max}(t)]. \quad (۷۶.۶)$$

که $۰ < \alpha = ۰/۲ < ۱$. همچنین $N(۰, ۱)$ متغیری تصادفی با توزیع نرمال با میانگین صفر و انحراف معیار ۱ است. بنابراین می‌توان گام‌های الگوریتم فرهنگی حاضر را به صورت زیر بیان نمود.

- گام ۱: تولید جمعیت اولیه و ارزیابی آن‌ها.
 - گام ۲: آماده سازی فضای اعتقادی.
 - گام ۳: انتخاب اعضای جمعیت برای تاثیر در فرهنگ یا فضای اعتقادی.
 - گام ۴: تغییر مولفه‌های فضای اعتقادی توسط اعضای منتخب.
 - گام ۵: تولید پاسخ‌های جدید با تاثیر فرهنگ بر پاسخ‌های فعلی و ارزیابی آن‌ها.
 - گام ۶: بازگشت به مرحله ۳ در صورت برآورده نشدن شرایط مسئله.
- مراحل بالا را می‌توان در شکل ۲.۶ به وضوح مشاهده نمود.

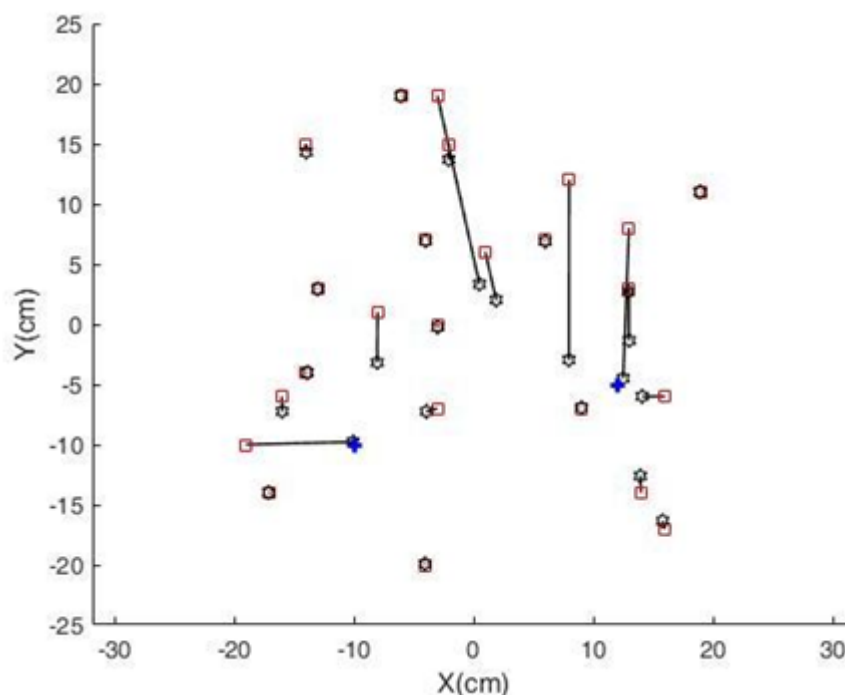


شکل ۲.۶: نمایی از کارکرد الگوریتم فرهنگی.

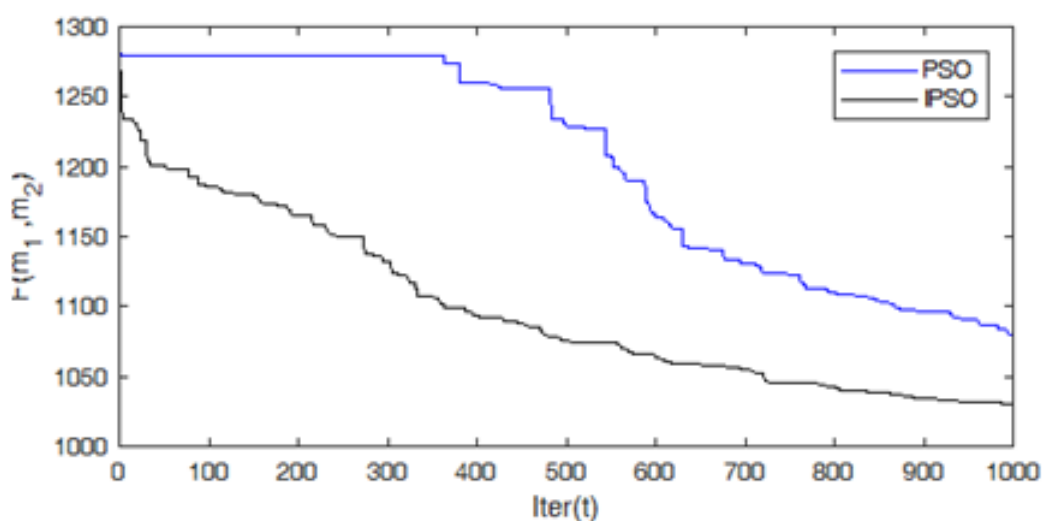
۷.۶ نتایج محاسباتی

در این قسمت مثال‌هایی عددی برای مسئله معکوس ۲-میانه پشتیبان در حالت پیوسته با روش تغییر در مختصات نقاط آورده شده است و اثرات مختلف تغییر در پارامترهای مختلف مسئله همچون تغییر در نرم p ، مقدار بودجه B ، و ضریب شکست ρ مورد بررسی قرار گرفته است. نرم افزار استفاده شده در این فصل، نرم افزار Matlab ورژن R2017b بوده است که توسط لپ تاپ مدل Fujitsu با مشخصات $Intel(R)Core^{TM}i5 - 2430M, CPU, 4, 00GB, RAM$ پیاده‌سازی شده است.

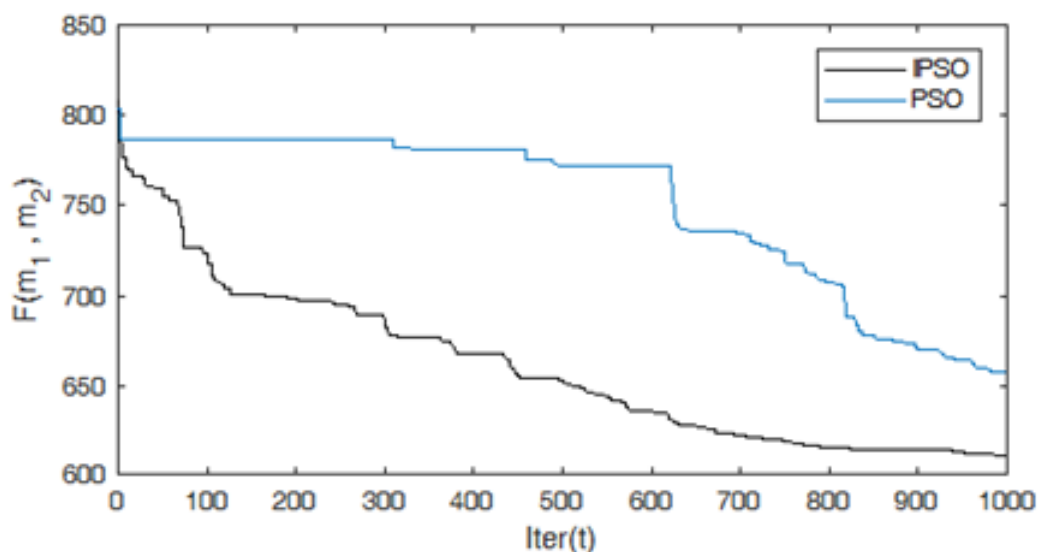
مثال ۱.۷.۶. در این مثال مسئله BR2MP با تغییر در مختصات نقاط در محیط پیوسته برای 20 مشتری با بودجه $B = 250$ و $\rho = 0.3$ در نرم اقلیدسی با روش‌های فراابتکاری PSO و IPSO حل شده است. نمودار ۳.۶ نحوه جابجایی نقاط اولیه (مشتریان) را نشان می‌دهد. و نمودار ۴.۶ و ۵.۶ نیز عملکرد الگوریتم‌های فراابتکاری PSO و IPSO استفاده شده برای $\rho = 0/3$ و $\rho = 0$ در این رساله را نشان می‌دهد. همان‌طور که در نمودار ۴.۶ و ۵.۶ مشاهده می‌کنید و در متن رساله نیز اشاره شد، عملکرد IPSO نسبت به PSO بهتر می‌باشد.



شکل ۳.۶: نقاط اولیه (مربع قرمز)، نقاط بهینه (ستاره مشکی) و نقاط ۲-میانه (نقاط بعلاوه آبی) برای $\rho = 0/3$ ، $n = 25$ و $B = 80$.



شکل ۴.۶: نمودار مقایسه‌ای عملکرد روش‌های PSO، IPSO برای حالت $\rho = 0/3$ ، $n = 25$ و $B = 80$.



شکل ۵.۶: نمودار مقایسه‌ای عملکرد روش‌های PSO، IPSO برای حالت $\rho = 0$ ، $n = 25$ و $B = 80$.

• اثرات ضریب شکست ρ ، بر روی مسئله BR2MP در حالت پیوسته به روش تغییر در مختصات

به‌طور کلی با توجه به تعریف یک مسئله ۲- میانه پشتیبان وقتی یکی از میانه‌ها

(سرویس‌دهنده‌ها) شکست بخورد مشتریان این سرویس دهنده مجبورند از سرویس دهنده دیگر سرویس بگیرند بنابراین این امر باعث افزایش مقدار تابع هدف خواهد شد. زمانی که $\rho = 0$ ، تابع هدف مسئله مذکور همان تابع هدف مسئله ۲- میانه معمولی است. (در رابطه (۳.۶)، ρ را معادل صفر در نظر بگیرید) ولی زمانیکه $\rho \neq 0$ ، دو جمله $\rho \sum_{v_i \in V} w(v_i)d(v_i, m_1)$ و $\rho \sum_{v_i \in V} w(v_i)d(v_i, m_2)$ به تابع هدف مسئله اضافه می‌شوند و با افزایش ρ این دو مقدار نیز افزایش می‌یابند. بنابراین با افزایش ضریب شکست ρ مقدار تابع هدف مسئله BR2MP نیز افزایش می‌یابد. این امر را می‌توان در جداول ۱.۶ و ۲.۶ مشاهده نمود.

مثال ۲.۷.۶. در این مثال دو شبیه‌سازی عددی با تعداد نقاط $n = 50$ و $n = 100$ آورده

شده است. جدول ۱.۶ مقدار تابع هدف با نرم اقلیدسی را به ازای $n = 100$ و بودجه $B = 40$ نسبت به مقادیر مختلف ρ ، نشان می‌دهد. جدول ۲.۶ نیز مقدار تابع هدف با نرم اقلیدسی را به ازای $n = 50$ و بودجه $B = 40$ نسبت به مقادیر مختلف ρ نشان می‌دهد.

جدول ۱.۶: تاثیرات ρ بر مقدار تابع هدف مسئله BR2MP برای حالت $B = 40$ ، $p = 2$ و $n = 100$.

ρ	F_{pso}	F_{ipso}	$CPU - Time_{pso}(s)$	$CPU - Time_{ipso}(s)$
0	68/81	70/04	63/74	32/23
0/3	166/90	166/56	50/59	31/10
0/5	225/74	225/44	39/28	33/29
0/7	287/37	287/49	38/17	47/93
0/9	346/46	345/40	39/58	48/74

جدول ۲.۶: تاثیرات ρ بر مقدار تابع هدف مسئله BR2MP برای حالت $B = 35$ ، $p = 2$ و $n = 50$.

ρ	F_{pso}	F_{ipso}	$CPU - Time_{pso}(s)$	$CPU - Time_{ipso}(s)$
0	63/02	571/99	123/05	121/75
0/3	$1/10E + 03$	$1/07E + 03$	91/32	99/81
0/5	$1/48E + 03$	$1/39E + 03$	92/94	134/18
0/7	$2/00E + 03$	$1/68E + 03$	129/42	108/35

مثال ۳.۷.۶. اثرات تغییر نرم p ، بر روی مسئله BR2MP در حالت پیوسته به روش تغییر در مختصات نقاط.

به طور کلی مقدار $d^p(X_i, m_t)$ (نرم l_p) با افزایش مقدار p کاهش پیدا می کند. بنابراین در یک بودجه ثابت B که $p < p'$ باشد، برای حالت p' نسبت به حالت p نقاط اولیه نزدیکتر به نقاط $\{m_1, m_2\}$ هستند. لذا بودجه ثابت B ، برای حالت p' نسبت به حالت p تابع هدف را بیشتر کاهش می دهد. بنابراین با افزایش مقدار p تابع هدف کاهش پیدا می کند. این امر را می توان در جدول ۳.۶ نیز مشاهده نمود. در جدول ۳.۶ یک مثال عددی با تعداد نقاط $n = 80$ و $\rho = 0.5$ و بودجه $B = 700$ نسبت به مقادیر مختلف p آورده شده است.

جدول ۳.۶: تاثیرات ρ بر مقدار تابع هدف مسئله BR2MP برای حالت $\rho = 0.5$ ، $B = 700$ و $n = 80$.

p	F_{pso}	F_{ipso}	$CPU - Time_{pso}(s)$	$CPU - Time_{ipso}(s)$
۱	$5/81E + 03$	$5/64E + 03$	۶۷۶/۰۹	۴۸۱/۰۶
۲	$۴/۷۱E + 03$	$۴/۶۵E + 03$	۳۵۳/۸۹	۳۷۸/۳۷
۵	$۴/۱۹E + 03$	$۳/۷۵E + 03$	۵۲۶/۱۴	۴۶۲/۰۶
۱۰	$۳/۸۳E + 03$	$۳/۳۹E + 03$	۲۶۲/۳۲۸	۶۵۷/۴۴

مثال ۴.۷.۶. اثرات تغییر بودجه B ، بر روی مسئله BR2MP در حالت پیوسته به روش تغییر در مختصات نقاط.

افزایش بودجه به این معنی است که می توان تغییرات بیشتری را در جابجایی مختصات نقاط اعمال کرد و این امر بیانگر این موضوع است که با افزایش بودجه، تابع هدف بهبود بخشیده می شود. به عبارت دیگر از لحاظ ریاضی با افزایش بودجه، فضای جوابی که محدودیت مسئله یعنی رابطه (۵۴.۶) ایجاد می کند افزایش یافته و این امر باعث بهبود مقدار تابع هدف، یعنی کاهش آن خواهد شد. جدول (۴.۶) اثرات تغییر بودجه را بر یک مسئله شبیه سازی شده برای حالت $\rho = 0.3$ ، $p = 3$ و $n = 30$ را نشان می دهد.

جدول ۴.۶: تاثیرات B ، بر مقدار تابع هدف مسئله BR2MP برای حالت $\rho = 0/3$ ،
 $p = 3$ و $n = 30$.

B	F_{ps0}	F_{ipso}	$CPU - Time_{ps0}(s)$	$CPU - Time_{ipso}(s)$
۱۰۰	۴۹۶/۹۰	۴۷۱/۷۹	۷۵۰/۸۸	۵۴۱/۸۰
۲۰۰	۳۲۶/۷۲	۳۱۴/۵۰	۱۲۰۷/۳۸	۶۸۲/۳۶
۵۰۰	۲۶۹/۴۸	۲۲۵/۴۳	۸۸۹/۹۱	۷۶۹/۵۲

مثال ۵.۷.۶. جدول ۵.۶، مشخصات نقاط اولیه و وزن مربوط به هر نقطه تقاضا را نشان می‌دهد. همچنین جواب‌های بدست آمده از حل مسئله نیز در جدول ۶.۶ قابل مشاهده است. جدول ۷.۶ نیز عملکرد الگوریتم فرهنگی ارائه شده را برای نرم‌های مختلف p بیان می‌کند. شکل ۶.۶، مقدار تابع هدف را بر حسب تعدا تکرار به‌ازای بودجه $\beta = 180$ نشان می‌دهد. همچنین شکل‌های ۷.۶ تا ۱۰.۶ مختصات نقاط اولیه (مربع) و نهایی (دایره) را به‌ازای مقادیر مختلف p نشان می‌دهد.

جدول ۵.۶: مختصات نقاط (مشتريها) به همراه وزن آن‌ها.

w_i	y_i	x_i	ردیف	w_i	y_i	x_i	ردیف
۷	۲/۵۳	۵/۲۷	۲۶	۵	۴/۴۳	۵/۵۳	۱
۸	۸/۶۱	۲/۰۳	۲۷	۶	۵/۰۸	۳	۲
۱۰	۲/۴۲	۶/۹۸	۲۸	۱	۶/۶۱	۹/۱۴	۳
۵	۰/۶۸	۷/۳۶	۲۹	۶	۷/۸	۲/۶۴	۴
۹	۰/۷۷	۶/۳۲	۳۰	۳	۶/۳۵	۴/۵۲	۵
۵	۰	۵/۷۸	۳۱	۴	۱/۶۲	۳/۱۳	۶
۱	۲/۰۶	۷/۴۱	۳۲	۱۰	۶/۹۷	۱/۶	۷
۹	۵/۰۴	۷/۳۸	۳۳	۸	۲/۶۸	۵/۲۴	۸
۴	۳/۳۱	۶/۲۸	۳۴	۲	۶/۳۱	۴/۸۷	۹
۳	۵/۶۱	۳/۹۶	۳۵	۶	۴/۷	۷/۵۵	۱۰
۷	۰/۳۴	۳/۳	۳۶	۶	۰/۴۶	۳/۳	۱۱
۹	۱/۰۹	۶/۳۱	۳۷	۹	۹/۹۳	۳/۲۳	۱۲
۹	۱/۴۴	۸/۹۱	۳۸	۳	۲/۰۲	۱/۶۹	۱۳
۳	۲/۳۳	۴/۹	۳۹	۷	۵/۲۵	۹/۰۸	۱۴
۴	۶/۶۶	۴/۲	۴۰	۲	۶/۶۸	۶/۶۵	۱۵
۱۰	۴/۴۹	۲/۶۷	۴۱	۴	۸/۳۹	۱/۶۳	۱۶
۳	۶/۲۷	۴/۰۱	۴۲	۴	۴/۲۸	۱/۳۲	۱۷
۲	۲/۴۴	۴/۳۵	۴۳	۱	۵/۳۹	۳/۰۴	۱۸
۸	۱/۴	۲/۵۵	۴۴	۱۰	۸/۳۲	۴/۶۶	۱۹
۳	۳/۳۷	۶/۱۶	۴۵	۹	۲/۷۸	۰/۱	۲۰
۱	۶/۵۹	۳/۹۱	۴۶	۹	۸/۸۵	۵/۲	۲۱
۴	۵/۹۴	۷/۷۶	۴۷	۱	۸/۵۴	۳/۱۸	۲۲
۸	۳/۳۳	۰/۸۷	۴۸	۸	۱/۹۷	۸/۰۵	۲۳
۷	۸/۸۲	۰/۰۹	۴۹	۱	۰/۳۴	۶/۲۵	۲۴
۸	۹/۵	۴/۲۲	۵۰	۳	۹/۹۲	۹/۷	۲۵

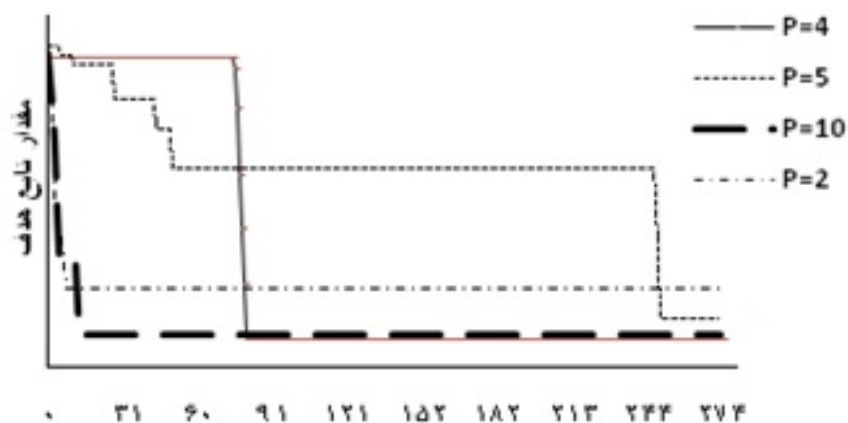
جدول ۶.۶: نتایج شبیه‌سازی شده با بودجه $\beta = 18^\circ$ ، برای نرم‌های $p = 2, 4, 5, 10$

ردیف	$p = 2$		$p = 4$		$p = 5$		$p = 10$	
	ζ_{i2}	ζ_{i1}	ζ_{i2}	ζ_{i1}	ζ_{i2}	ζ_{i1}	ζ_{i2}	ζ_{i1}
۱	۰/۹	۰/۹	۰/۰۴	۳/۳۴	۰/۰۸	۱/۹۱	۲/۱۶	۰/۲۳
۲	۰	۰	۰	۰	۰	۰	۰	۰
۳	۱/۵۳	۱/۵۳	۱/۱۷	۱/۱۷	۱/۱۴	۱/۱۴	۱/۴	۱/۴
۴	۰/۱۱	۱/۸۹	۰/۰۶	۳/۰۸	۰/۲۵	۲/۲۳	۰/۴	۳/۰۲
۵	۱/۳۶	۱/۳۶	۱/۴۸	۱/۴۸	۱/۴۶	۱/۴۶	۱/۵۱	۱/۵۱
۶	۰/۸۴	۲/۹۲	۱/۰۴	۳/۸۳	۰/۱۲	۳	۰/۹۳	۲/۶۹
۷	۱/۶۳	۱/۶۳	۰/۸۸	۱/۷۹	۰/۹۴	۲/۲۱	۱/۱۷	۲/۱
۸	۱/۶۶	۱/۶۶	۲/۲۸	۲/۲۸	۲/۲۵	۲/۲۵	۲/۴۴	۲/۴۴
۹	۳/۵۸	۱/۰۲	۱/۷۸	۱/۷۸	۱/۸	۱/۱۴	۲/۳۳	۱/۰۵
۱۰	۰	۰	۰	۰	۰	۰	۰	۰
۱۱	۱/۰۱	۴	۰/۹۹	۴/۱۷	۰/۹۴	۴/۵۵	۰/۵	۵/۱
۱۲	۱/۱۳	۴/۲	۱/۵۷	۴/۸۴	۰/۶۱	۵/۳۴	۰/۸۸	۴/۹۳
۱۳	۰/۷۴	۲/۶۷	۱/۰۶	۲/۸۴	۰/۷۳	۲/۸۲	۰/۵۱	۲/۶۲
۱۴	۱/۰۹	۱/۰۹	۱/۵۶	۰/۰۴	۱/۴۶	۰/۰۲	۱/۳۴	۰/۵۲
۱۵	۰/۹۷	۰/۹۷	۱/۳۷	۱/۳۷	۱/۲۲	۱/۲۲	۰/۴۳	۱/۴۱
۱۶	۰/۹	۳/۷۷	۰/۹۴	۳/۴۳	۰/۸۵	۲/۷۶	۰/۸۲	۲/۶۶
۱۷	۱/۴۷	۱/۴۷	۰/۶۹	۰/۶۹	۰/۸۲	۰/۸۲	۱/۰۶	۱/۰۶
۱۸	۰	۰	۰	۰	۰	۰	۰/۸۲	۰/۸۲
۱۹	۲/۷۵	۲/۷۵	۲/۷۹	۲/۷۹	۱/۹۱	۲/۸۷	۲/۷۵	۲/۷۵
۲۰	۲/۷	۲/۷	۲/۲۱	۲/۲۱	۲/۲۱	۲/۲۹	۲/۳۱	۲/۳۱
۲۱	۱/۴۷	۳/۱۴	۲/۱۵	۳/۸۲	۲/۲	۴/۳۱	۲/۸۶	۴/۲۳
۲۲	۱/۶۲	۳/۲۳	۰/۰۵	۳/۳۲	۰/۳۳	۴/۰۲	۰/۶۸	۳/۹۷
۲۳	۰/۱۶	۲/۶۶	۱/۳۶	۲/۹۷	۰/۵۶	۲/۷۲	۰/۶۴	۳/۱۳
۲۴	۱/۸	۳/۸۸	۰/۶۱	۵/۱۵	۱/۰۴	۴/۹۳	۱/۴۴	۴/۲۵

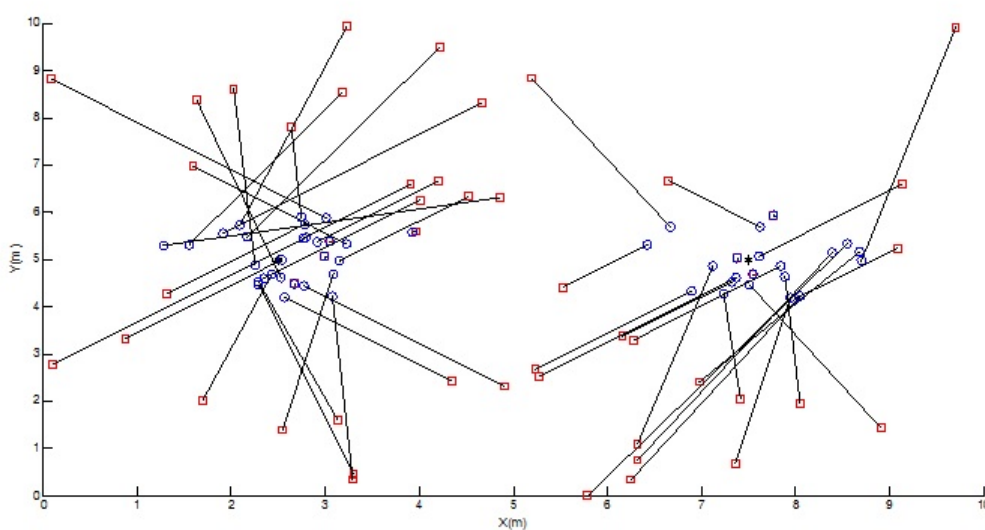
$p = 10$		$p = 5$		$p = 4$		$p = 2$		ردیف
ζ_{i2}	ζ_{i1}	ζ_{i2}	ζ_{i1}	ζ_{i2}	ζ_{i1}	ζ_{i2}	ζ_{i1}	
۵/۹۷	۱/۷۵	۴/۶۱	۲/۴۱	۵/۳	۲/۰۱	۴/۹۴	۱/۰۱	۲۵
۲/۸	۲/۸	۲/۳۱	۲/۳۱	۱/۸۲	۱/۸۲	۲/۱	۲/۱	۲۶
۳/۷	۰/۲۲	۳/۲۴	۱/۰۶	۳/۷۴	۰/۴۲	۳/۷۲	۰/۲۲	۲۷
۲/۴۱	۰/۴۸	۲/۳۸	۰/۰۵	۲/۰۹	۰/۶۳	۲/۷۵	۱/۷۱	۲۸
۴/۸	۰/۵۳	۴/۷۶	۰/۵۴	۴/۴۵	۰/۴۳	۳/۵۲	۰/۵۹	۲۹
۳/۸۹	۱/۲۴	۳/۸۶	۱/۱۲	۳/۶۷	۱/۰۹	۴/۳۸	۲/۰۶	۳۰
۴/۹۹	۲/۱۲	۴/۳۸	۱/۷	۴/۹۲	۲/۱۲	۵/۳۴	۲/۷۶	۳۱
۳/۱۷	۰/۶۵	۳/۵۵	۰/۱۴	۲/۵۷	۰/۵۸	۲/۲۲	۰/۱۸	۳۲
۰	۰	۰	۰	۰	۰	۰	۰	۳۳
۱/۳۴	۱/۳۴	۱/۲۱	۱/۲۱	۱/۴۸	۱/۴۸	۱/۵۶	۱/۵۶	۳۴
۰/۲۳	۱/۱۹	۰/۴۳	۱/۳۸	۰/۴۳	۲/۴۶	۰/۰۳	۰/۰۳	۳۵
۵/۰۸	۰/۶۳	۴/۴۶	۰/۹۵	۵/۰۹۷	۱/۰۳	۳/۸۸	۰/۲۲	۳۶
۳/۸۶	۰/۷۲	۵/۲۳	۱/۸۲	۳/۴۵	۰/۹۴	۳/۷۸	۰/۸۱	۳۷
۳/۶	۱/۷	۳/۰۶	۲/۰۹	۳/۷	۱/۳۲	۳/۰۴	۱/۴	۳۸
۲/۸۳	۲/۸۳	۲/۳۳	۲/۳۳	۲/۳۳	۲/۳۳	۲/۱۳	۲/۱۳	۳۹
۱/۲۵	۱/۲۵	۱/۶۹	۱/۶۹	۱/۴۶	۱/۴۶	۱/۲۹	۱/۲۹	۴۰
۰	۰	۰	۰	۰	۰	۰	۰	۴۱
۱/۵۹	۱/۵۹	۱/۵	۱/۵	۱	۱	۱/۶۶	۱/۶۶	۴۲
۲/۳	۲/۳	۲/۲	۲/۲	۲/۴۲	۲/۴۲	۱/۷۸	۱/۷۸	۴۳
۳/۰۷	۰/۴۲	۳/۵	۰/۵۵	۳/۷۲	۰/۷۳	۳/۳	۰/۵۴	۴۴
۱/۳۱	۱/۳۱	۱/۷۲	۱/۷۲	۲/۰۵	۲/۰۵	۱/۱۶	۱/۱۶	۴۵
۱/۷۵	۱/۷۵	۱/۴۹	۱/۴۹	۱/۱۷	۱/۱۷	۱/۱۵	۱/۱۵	۴۶
۰/۵۲	۰/۵۲	۰/۶۵	۰/۶۵	۰/۵۸	۰/۵۸	۰	۰	۴۷
۱/۵۹	۱/۵۹	۱/۹۷	۱/۹۷	۲/۱	۲/۱	۱/۶۶	۱/۶۶	۴۸
۳/۰۶	۳/۰۶	۳/۸	۱/۶۶	۳/۴۷	۱/۹۶	۲/۹۲	۲/۹۲	۴۹
۴/۴۲	۲/۱۵	۴/۴۷	۲/۱۹	۴/۱۲	۱/۶۴	۴/۰۱	۲/۰۵	۵۰

جدول ۷.۶: مقادیر تابع هدف به‌ازای نرم‌های $p = 2, 4, 5, 10$ ، بر حسب تعداد تکرار

ردیف	$p = 2$	$p = 4$	$p = 5$	$p = 10$
۱	۱۳۲/۸	۱۳۳/۲۱	۱۳۰/۱	۱۳۱/۱۱
۲	۱۱۹	۱۲۵/۱۲	۱۲۹/۲	۱۲۷/۰۹
۳	۱۰۲	۱۲۵/۱۲	۱۲۹/۲	۱۱۶/۳۴
۴	۸۰/۲۷	۱۲۵/۱۲	۱۲۹/۲	۱۰۴/۲۸
۵	۵۲/۱۲	۱۲۵/۱۲	۱۲۹/۲	۹۰/۶۰۳
⋮	⋮	⋮	⋮	⋮
۴۹۹	۳۲/۰۸	۱۱/۵۸۴	۱۹/۳۵	۱۲/۸۷
۵۰۰	۳۲/۰۸	۱۱/۵۸۴	۱۹/۳۵	۱۲/۸۷

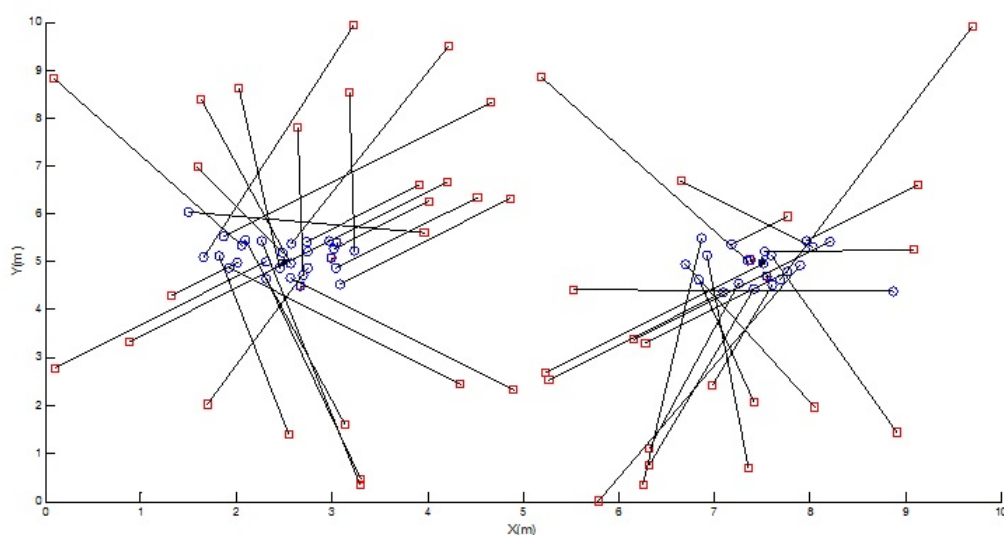


شکل ۶.۶: مقادیر تابع هدف بر حسب تعداد تکرار برای حالت $\beta = 180$



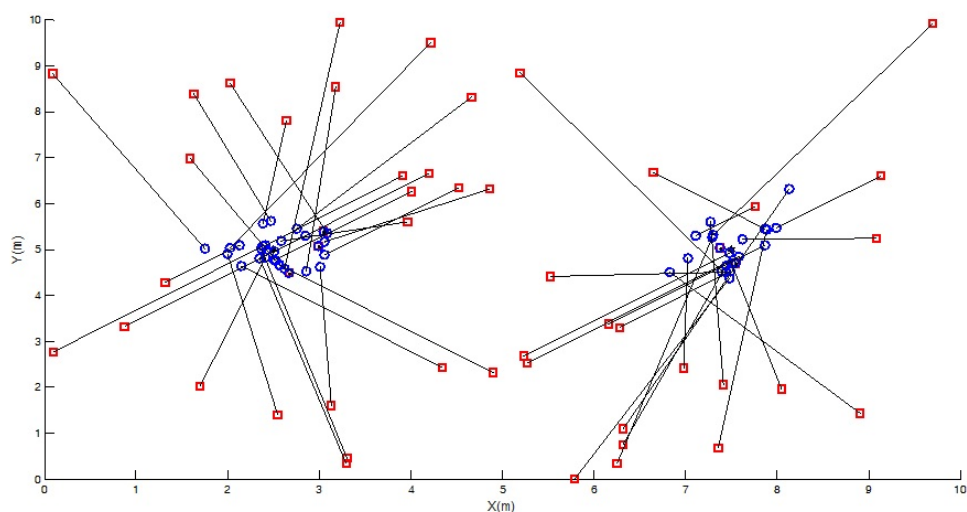
شکل ۷.۶: مختصات نقاط اولیه (مربع) و نقاط نهایی (دایره) برای حالت $p = 2$ و

$$\beta = 18^\circ$$



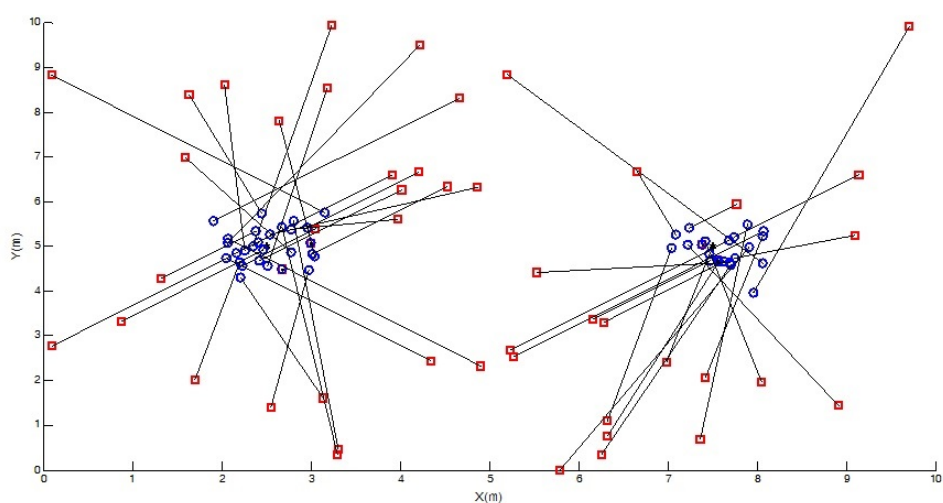
شکل ۸.۶: مختصات نقاط اولیه (مربع) و نقاط نهایی (دایره) برای حالت $p = 4$ و

$$\beta = 18^\circ$$



شکل ۹.۶: مختصات نقاط اولیه (مربع) و نقاط نهایی (دایره) برای حالت $p = 5$ و

$$\beta = 18^\circ$$



شکل ۱۰.۶: مختصات نقاط اولیه (مربع) و نقاط نهایی (دایره) برای حالت $p = 10$ و

$$\beta = 18^\circ$$

همچنین جهت بررسی کارایی الگوریتم فرهنگی، در ادامه نتایج این روش را با نتایج حاصل از دو الگوریتم فراابتکاری ازدحام ذرات (PSO) و الگوریتم بهبود یافته ازدحام ذرات (IPSO) مقایسه کرده‌ایم. برای این کار یک مسئله با $n = 40$ ، که داده‌های آن به‌طور تصادفی تولید شده‌اند با سه روش ذکر شده حل شده است. نتایج حاصل به ازای

بودجه‌های مختلف، ضریب شکست‌های متفاوت و تحت نرم‌های مختلف در جدول ۸.۶ تا ۱۰.۶ آمده است.

جدول ۸.۶: مقایسه نتایج بدست آمده از روش‌های $IPSO$ ، PSO و CA ، با $n = ۴۰$ ، $\beta = ۷۵۰$ ، $\rho = ۰/۵$ و تعداد تکرار $t = ۱۰۰۰$ نسبت به نرم‌های $۱، ۱/۵، ۲، ۵، ۱۰$ ، $p = ۱$.

$CPU_{Time}(CA)$	$CPU_{Time}(IPSO)$	$CPU_{Time}(PSO)$	CA	$IPSO$	PSO	p	Test#
۱۷۰/۱۵۶۴۴	۲۱۵/۶۶۷۲	۲۲۱/۵۹۵۷	۲/۲۶E+۰۳	۲/۵۸E+۰۳	۲/۷۶E+۰۳	۱	۱
۱۲۵/۳۲۵۶	۱۸۶/۹۴۴	۲۲۵/۵۵۱۷	۱/۹۲E+۰۳	۲/۲۵E+۰۳	۲/۴۰E+۰۳	۱/۵	۲
۲۰۰/۱۵۵	۲۰۵/۰۴۶۸	۱۸۷/۸۳۷۱	۱/۷۶E+۰۳	۲/۰۱E+۰۳	۲/۰۹E+۰۳	۲	۳
۱۸۹/۴۵۲۶	۲۲۶/۴۷۳۳	۲۱۶/۸۶۶۹	۱/۳۰E+۰۳	۱/۶۷E+۰۳	۱/۸۳E+۰۳	۵	۴
۱۴۷/۲۵۶۸	۲۳۷/۷۹۶	۱۹۸/۶۲۲۸	۱/۴۹E+۰۳	۱/۵۹E+۰۳	۱/۸۰E+۰۳	۱۰	۵

جدول ۹.۶: مقایسه نتایج بدست آمده از روش‌های $IPSO$ ، PSO و CA با $n = ۴۰$ ، $\beta = ۱۰۰، ۳۰۰، ۵۰۰، ۷۰۰$ نسبت به $\rho = ۰/۳$ ، $p = ۲$ و تعداد تکرار $t = ۱۰۰۰$.

$CPU_{Time}(CA)$	$CPU_{Time}(IPSO)$	$CPU_{Time}(PSO)$	CA	$IPSO$	PSO	β	Test#
۲۰۰/۴۱۹۹	۲۱۶/۸۶۷۸	۲۲۲/۴۳۲۲	۳/۱۵E+۰۳	۳/۰۹E+۰۳	۳/۱۷E+۰۳	۱۰۰	۶
۱۷۵/۲۳۶۵	۲۱۸/۶۵۳۹	۲۴۶/۵۷۳۶	۱/۳۸E+۰۳	۲/۴۴E+۰۳	۲/۵۵E+۰۳	۳۰۰	۷
۱۰۲/۱۲۵	۱۱۳/۱۲۵۹	۲۳۴/۹۷۳۲	۱/۰۸E+۰۳	۱/۸۲E+۰۳	۲/۰۰E+۰۳	۵۰۰	۸
۱۸۴/۱۲۵	۲۱۵/۰۶۸۹	۲۳۴/۱۸۴۶	۱/۲۲E+۰۳	۱/۳۶E+۰۳	۱/۴۸E+۰۳	۷۰۰	۹

جدول ۱۰.۶: مقایسه نتایج بدست آمده از روش‌های $IPSO$ ، PSO و CA با $n = ۴۰$ ، $\beta = ۵۰۰$ ، $p = ۲$ و تعداد تکرار $t = ۱۰۰۰$ نسبت به $\rho = ۰، ۰/۳، ۰/۵، ۰/۸$.

$CPU_{Time}(CA)$	$CPU_{Time}(IPSO)$	$CPU_{Time}(PSO)$	CA	$IPSO$	PSO	ρ	Test#
۲۱۶/۴۱۹۴	۲۲۵/۹۵۰۶	۲۲۹/۰۵۸۲	۹/۳۶E+۰۲	۸/۵۸E+۰۲	۹/۹۷E+۰۲	۰	۱۰
۷۵/۲۳۶۵	۱۱۳/۱۲۵۹	۲۳۴/۹۷۳۲	۹۵۴/۲۳۶۵	۱/۸۲E+۰۳	۲/۰۰E+۰۳	۰/۳	۱۱
۱۰۲/۱۲۵۱	۱۱۷/۷۸۱۱	۱۷۳/۹۲۸۴	۱۸۰۴/۲۳۶۵	۲/۵۰E+۰۳	۲/۸۸E+۰۳	۰/۵	۱۲
۱۱۴/۱۲۵۸	۱۳۶/۵۳۷۵	۲۲۱/۱۲۱۷	۳/۳۲E+۰۴	۳/۵۰E+۰۳	۳/۶۱E+۰۳	۰/۸	۱۳

در جدول ۱۱.۶ نتایج حاصل از سه روش ذکر شده برای مسائلی با تعداد نقاط مختلف و بودجه و ضریب شکست‌های متفاوت آمده است. نتایج نشان می‌دهد که الگوریتم فرهنگی هم از نظر کیفیت جواب و هم از نظر زمان محاسبه از هر دو روش دیگر بهتر عمل کرده است. در الگوریتم‌های ارائه شده جمعیت اولیه همان نقاط اولیه در نظر گرفته شده‌اند. البته جمعیت‌های مختلف دیگری نیز مورد بررسی قرار گرفتند ولی بهبود قابل توجهی حاصل نشد.

جدول ۱۱.۶: مقایسه نتایج بدست آمده از روش‌های *IPSO*، *PSO* و *CA* با تعداد

نقاط، بودجه و ضریب شکست‌های متفاوت.

$CPU_{Time}(CA)$	$CPU_{Time}(IPSO)$	$CPU_{Time}(PSO)$	<i>CA</i>	<i>IPSO</i>	<i>PSO</i>	ρ	β	n	<i>Test#</i>
۹۰/۸۲۱۳	۱۱۴/۵۱۰۵	۱۰۵/۶۵۴	۸/۳۳E+۰۲	۸/۵۳E+۰۲	۸/۵۴E+۰۲	۰	۱۸۰	۲۵	۱۴
۹۸/۶۷۱۴	۹۵/۲۴۸۴	۱۰۵/۱۷۲	۱/۰۱E+۰۳	۱/۹۲E+۰۳	۱/۹۷E+۰۳	۰/۵	۱۸۰	۲۵	۱۵
۱۴۰/۸۱۸	۱۵۸/۵۹۱۸	۲۱۳/۱۱۳	۲/۶۱E+۰۳	۲/۷۱E+۰۳	۲/۹۷E+۰۳	۰	۴۰۰	۵۰	۱۶
۱۱۰/۰۹۸۵	۲۲۱/۴۹۲۷	۱۱۷/۳۳۳۹	۴/۶۶E+۰۳	۵/۲۴E+۰۳	۵/۴۱E+۰۳	۰/۵	۴۰۰	۵۰	۱۷
۱۰۰/۰۹۸۵	۲۳۴/۵۳۵۹	۲۸۹/۱۴۴۶	۴/۴۲E+۰۳	۴/۸۴E+۰۳	۵/۰۶E+۰۳	۰	۸۰۰	۷۰	۱۸
۱۲۱/۸۳۱۹	۲۱۸/۰۲۵	۳۴۲/۵۷۸۴	۴/۴۴E+۰۳	۷/۲۱E+۰۳	۸/۲۲E+۰۳	۰/۳	۸۰۰	۷۰	۱۹
۱۲۵/۸۴۱۲	۴۲۵/۵۳۲	۳۹۵/۹۱۲۵	۱/۱۱E+۰۴	۱/۲۷E+۰۴	۱/۳۱E+۰۴	۰	۱۵۰۰	۱۰۰	۲۰
۱۳۲/۱۸۱۱	۲۶۱/۱۱۳۳	۲۷۴/۱۷۵۷	۱/۰۵E+۰۴	۱/۴۳E+۰۴	۱/۵۶E+۰۴	۰/۲	۲۰۰۰	۱۰۰	۲۱

فصل ۷

معکوس مسئله مکان‌یابی متعادل

مسائل مکان‌یابی متعادل در سال‌های اخیر بسیار مورد توجه قرار گرفته است. به‌طور کلی این‌گونه از مسائل به دنبال استقرار سرویس‌دهنده‌هایی است که با استقرار این سرویس‌دهنده‌ها، تعادل و برابری در سرویس‌دهی به نقاط تقاضا به حداکثر خود برسد. این موضوع مورد بررسی بسیاری از پژوهشگران قرار گرفته است. در این میان گاوالک^۱ و هادک^۲ تابع هدف متعادلی را مورد مطالعه قرار دادند به‌طوری‌که این تابع هدف، حداکثر اختلاف در فاصله از یک نقطه تقاضا به دورترین و نزدیکترین تسهیلات بود [۶۹]. برمن و همکاران، مسئله انتخاب p سرویس‌دهنده به‌طوری‌که بیشترین وزن اختصاص داده شده به هر سرویس‌دهنده کمینه شود را مورد مطالعه قرار دادند [۱۸]. مارین^۳ مسئله مکان‌یابی متعادل را که در آن به دنبال کمینه کردن اختلاف بین حداکثر و حداقل وزن اختصاص داده شده به سرویس‌دهنده‌ها بود را مورد مطالعه قرار داد [۱۰۱].

فتحعلی و زعفرانیه نیز یک الگوریتم چندجمله‌ای برای یک مسئله مکان‌یابی متعادل بر روی شبکه‌های درختی ارائه کردند [۵۸]. لانداته^۴ و مارین نیز یک مسئله

^۱Gavalec

^۲Hudec

^۳Marin

^۴Landete

مکان‌یابی را که در آن به دنبال کمینه کردن اختلاف مجموع وزن اختصاص داده شده به سرویس‌دهندگان بودند را مورد بررسی قرار دادند [۹۶]. همچنین بریتی^۵ و پیکلو^۶ برخی خواص را برای توصیف معیار تعادل در مسائل مکان‌یابی سرویس‌دهندگان ارائه کرده‌اند [۱۱].

۱.۷ تعریف مسئله

فرض کنید $G = (V, E)$ یک گراف بدون جهت باشد به‌طوری‌که V مجموعه رئوس گراف G ، E مجموعه یال‌های گراف G ، $|V| = n$ و $|E| = m$. هر راس v_i نیز دارای یک وزن نامنفی w_i است. w_i را می‌توان به‌عنوان تعداد مشتریان راس v_i در نظر گرفت. به‌عبارت دیگر می‌توان w_i را به‌عنوان میزان تقاضای راس v_i نیز تعبیر نمود. $d_{ij} = d(v_i, v_j)$ کوتاهترین مسیر بین دو راس v_i و v_j در گراف G است. برای هر $S \subset V$ قرار می‌دهیم:

$$W(S) = \sum_{v_i \in S} w_i.$$

در یک مسئله ۲- سرویس‌دهنده متعادل هدف کمینه کردن تعداد مشتریانی است که به سرویس‌دهنده با بیشترین مشتری تخصیص داده می‌شوند. به‌عبارت دیگر در یک مسئله ۲- سرویس‌دهنده متعادل هدف کمینه کردن رابطه زیر است:

$$\min_{X_p} \left(\max_{k=1,2} \sum_{i \in V_k, w_i \geq 0} w_i \right). \quad (1.7)$$

در رابطه (۱.۷)، $X_p = \{m_1, m_2\}$ ، مجموعه سرویس‌دهنده‌ها و V_1 و V_2 به‌ترتیب مجموعه رئوسی هستند که به m_1 و m_2 تخصیص می‌یابند و از روابط زیر به‌دست می‌آیند

$$V_1 = \{v_i; d(v_i, m_1) \leq d(v_i, m_2)\}, \quad (2.7)$$

$$V_2 = V \setminus V_1. \quad (3.7)$$

در حالت معکوس مسئله ۲- سرویس‌دهنده متعادل می‌خواهیم با تغییر در پارامتر وزن رئوس، دو راس از پیش تعیین شده m_1 و m_2 با کمترین هزینه ممکن یا بیشترین سود، بهینه شوند. در مسئله حاضر فرض بر این است که دو سرویس‌دهنده m_1 و m_2 معلوم هستند. بنابراین برای رسیدن به تعادل (بهینه شدن مسئله) باید وزن

⁵Barbati

⁶Piccolo

رئوس گراف تغییر کند. فرض کنید افزایش هر واحد وزن راس v_i هزینه‌ای معادل c_i^+ و کاهش هر واحد وزن راس v_i هزینه‌ای معادل c_i^- را داشته باشد. فرض کنید q_i^+ و q_i^- به ترتیب مقدار افزایش و کاهش وزن w_i باشند. همچنین برای میزان افزایش وزن w_i ، یک کران بالای u_i^+ را در نظر می‌گیریم. قرار می‌دهیم:

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

و برای هر $i = 1, \dots, n$ قرار می‌دهیم $\hat{w}_i = w_i + q_i^+ - q_i^-$. بنابراین دو تابع هدف زیر را در نظر می‌گیریم:

$$f_1(Q^+, Q^-) = \sum_{i=1}^n (c_i^+ q_i^+ + c_i^- q_i^-). \quad (4.7)$$

$$f_2(Q^+, Q^-) = \left| \sum_{i \in V_1} \hat{w}_i - \sum_{i \in V_2} \hat{w}_i \right|. \quad (5.7)$$

باید توجه داشت که مقدار بهینه تابع هدف f_2 برابر با صفر است و این در صورتی اتفاق می‌افتد که $W(V_1) = W(V_2)$.

رابطه (۴.۷) بیانگر هزینه کل ناشی از افزایش یا کاهش وزن رئوس v_i برای رسیدن به تعادل است و رابطه (۵.۷) بیانگر شرط تعادل است. از طرف دیگر بعضی اوقات محدودیت در بودجه باعث می‌شود که نتوان وزن رئوس را برای رسیدن به تعادل تنظیم نمود. از این رو در ادامه دو نوع از مسائل با بودجه محدود و نامحدود که به ترتیب مسائل معکوس محدودیت بودجه‌ای و حداقل هزینه نام دارند را بررسی خواهیم کرد.

۲.۷ معکوس نوع حداقل هزینه ۲- سرویس دهنده متعادل

در این قسمت فرض می‌کنیم که بودجه مورد نظر برای تنظیم وزن رئوس تا رسیدن به تعادل نامحدود باشد. این مسئله را معکوس نوع حداقل هزینه می‌نامیم. در حالت معکوس نوع حداقل هزینه به دنبال این هستیم تا با تغییر در وزن رئوس و با کمترین هزینه ممکن وزن رئوسی که به سرویس‌دهنده‌های m_1 و m_2 اختصاص داده می‌شوند به تعادل برسند. بنابراین می‌توان مدل این مسئله را به صورت زیر در نظر گرفت:

$$p_1 : \quad \min f_1 = \sum_{i=1}^n (c_i^+ q_i^+ + c_i^- q_i^-) \quad (6.7)$$

s.t.

$$\left| \sum_{i \in V_1} \hat{w}_i - \sum_{i \in V_2} \hat{w}_i \right| = 0, \quad (7.7)$$

$$\hat{w}_i = w_i + q_i^+ - q_i^-, \quad i = 1, 2, \dots, n. \quad (8.7)$$

$$0 \leq q_i^+ \leq u_i, \quad i = 1, 2, \dots, n. \quad (9.7)$$

$$0 \leq q_i^- \leq w_i, \quad i = 1, 2, \dots, n. \quad (10.7)$$

محدودیت‌های (۷.۷) و (۸.۷) را می‌توان با محدودیت زیر جایگزین نمود:

$$\sum_{i \in V_1} (w_i + q_i^+ - q_i^-) - \sum_{i \in V_2} (w_i + q_i^+ - q_i^-) = 0. \quad (11.7)$$

بنابراین مدل به یک مدل برنامه‌ریزی خطی تبدیل می‌شود. در ادامه یک الگوریتم از مرتبه $O(n \log n)$ را برای حل مسئله p_1 ارائه می‌دهیم.

۱.۲.۷ الگوریتم حل معکوس حداقل هزینه مسئله ۲ – سرویس‌دهنده متعادل

اگر $W(V_1) = W(V_2)$ ، بنابراین تعادل برقرار است و مسئله در حالت بهینه قرار دارد و نیازی به تغییر در وزن رئوس نیست. در غیر این صورت اگر $W(V_1) \neq W(V_2)$ بدون کاستن از کلیت فرض می‌کنیم $W(V_1) > W(V_2)$. بنابراین لم زیر را می‌توان بیان کرد.

لم ۱.۲.۷. برای رسیدن به یک جواب شدنی یا باید وزن مجموعه رئوس V_1 کاهش پیدا کند یا به وزن مجموعه رئوس V_2 اضافه شود.

قرار می‌دهیم

$$D = \{r_1, r_2, \dots, r_{2n}\}$$

که r_i ، یا c_i^+ یا c_i^- است که در آن

$$r_1 \leq r_2 \leq r_3 \leq \dots \leq r_{2n}.$$

برای پیدا کردن یک جواب با کمترین هزینه از r_1 شروع می‌کنیم که r_1 ممکن است c_k^+ یا c_k^- باشد. ابتدا حالت $r_1 = c_k^+$ را در نظر می‌گیریم. اگر $v_k \in V_2$ در این صورت باید

$W(V_2)$ افزایش پیدا کند لذا قرار می‌دهیم $q_k^+ = \min\{W(V_1) - W(V_2), u_k^+\}$. حال اگر $v_k \in V_1$ در این صورت به سراغ r_2 می‌رویم.
در حالت $r_1 = c_k^-$ ، اگر $v_k \in V_1$ در این صورت قرار می‌دهیم $q_k^- = \min\{W(V_1) - W(V_2), w_k\}$ و اگر $v_k \in V_2$ به سراغ r_2 می‌رویم. با ادامه این روش برای r_1 ، r_2 و تمامی r_{2n} ها به یک جواب بهینه می‌رسیم. شبه کد زیر الگوریتم حل مسئله را بیان می‌کند:

Algorithm [IE2FLP]⁷.

Input: The weighted graph G , two vertices m_1 and m_2 of G as location of facilities and the cost of increasing and decreasing of vertices weights.

Output: The new weights of vertices \hat{w}_i for balancing the weights of vertices which assigned to the facilities in m_1 and m_2 .

Initialization:

Set $V_1 = \{v_i \in V | d(v_i, m_1) \leq d(v_i, m_2)\}$ and $V_2 = V \setminus V_1$.

If $W(V_1) = W(V_2)$ then **Stop**, the current weights are optimal.

If $W(V_1) > W(V_2)$ then set $\hat{V}_1 = V_1$ and $\hat{V}_2 = V_2$,

Else set $\hat{V}_1 = V_2$ and $\hat{V}_2 = V_1$.

Sort the cost of changing vertices weights, i.e. c_1^+, \dots, c_n^+ and c_1^-, \dots, c_n^- , in an increasing order and called them r_1, \dots, r_{2n} .

Iteration counter $i := 0$.

(For any vertex v_i in \hat{V}_1 or \hat{V}_2 , let \hat{w}_i be the weight of v_i).

Set $f_1 := 0$ and for $i = 1, \dots, n$, $\hat{w}_i = w_i$.

Iteration step:

While the $W(\hat{V}_1) \neq W(\hat{V}_2)$ **do** the following:

1. **If** $r_i = c_k^+$ and $v_i \in \hat{V}_2$ **then** set
 - (a) $q_k^+ := \min\{W(\hat{V}_1) - W(\hat{V}_2), u_k^+\}$,
 - (b) $\hat{w}_k := w_k + q_k^+$,
 - (c) $f_1 := f_1 + r_i q_k^+$.

End if

2. **If** $r_i = c_k^-$ and $v_i \in \hat{V}_1$ **then set**

$$(a) \quad q_k^- := \min\{W(\hat{V}_1) - W(\hat{V}_2), w_k\},$$

$$(b) \quad \hat{w}_k := w_k - q_k^-,$$

$$(c) \quad f_1 := f_1 + r_i q_k^-.$$

End if

3. **Set** $i := i + 1$.

end while

از آنجایی که مراحل تکرار الگوریتم نیاز به زمانی از مرتبه $O(n)$ دارد، و برای مرتب کردن مقادیر هزینه‌ها به صورت غیر نزولی نیاز به زمانی از مرتبه $O(n \log n)$ می‌باشد، بنابراین پیچیدگی کلی الگوریتم ارائه شده از مرتبه $O(n \log n)$ خواهد بود.

قضیه ۱.۲.۷. معکوس نوع حداقل هزینه مسئله ۲- سرویس‌دهنده با در نظر گرفتن تعادل بین سرویس‌دهنده‌ها، در زمان $O(n \log n)$ قابل حل است.

برای نشان دادن کارکرد الگوریتم بالا مثال زیر را در نظر بگیرید.

مثال ۱.۲.۷. درخت نشان داده شده در شکل ۱.۷ را در نظر بگیرید (مثال ارائه شده در مقاله برمن [۱۸]). هزینه افزایش و کاهش هر واحد وزن رئوس درخت در جدول ۱.۷ آمده است. قرار دهید $v_3 = m_1$ و $v_6 = m_2$. در این صورت داریم:

$$V_1 = \{v_1, v_2, v_3, v_4, v_9\},$$

$$V_2 = \{v_5, v_6, v_7, v_8\},$$

که در آن $W(V_1) = 0/6$ و $W(V_2) = 0/4$. با مرتب کردن ضرائب هزینه به صورت صعودی داریم:

$$\begin{aligned} r_1 = c_1^- \leq r_2 = c_2^- \leq r_3 = c_3^+ \leq r_4 = c_4^+ \leq r_5 = c_5^+ \leq r_6 = c_6^- \leq r_7 = c_7^- \\ \leq r_8 = c_8^+ \leq r_9 = c_9^- \leq r_{10} = c_{10}^- \leq r_{11} = c_{11}^- \leq r_{12} = c_{12}^+ \leq r_{13} = c_{13}^+ \\ \leq r_{14} = c_{14}^- \leq r_{15} = c_{15}^+ \leq r_{16} = c_{16}^+ \leq r_{17} = c_{17}^- \leq r_{18} = c_{18}^+. \end{aligned}$$

جدول ۱.۷: هزینه تغییر در وزن رئوس دخت T .

v_i	w_i	c_i^+	c_i^-
v_1	۰/۰۵	۱	۰/۲
v_2	۰/۱	۰/۲۵	۰/۱
v_3	۰/۲	۰/۳	۱
v_4	۰/۱۵	۰/۷	۰/۶
v_5	۰/۱۵	۲	۰/۵
v_6	۰/۱	۱/۵	۰/۷
v_7	۰/۱	۰/۴	۱/۵
v_8	۰/۰۵	۱/۵	۱
v_9	۰/۱	۲	۲

بنابراین از $c_1^- = r_1$ شروع کرده و قرار می‌دهیم $q_2^- = 0$ ، $\hat{w}_2 = 0$ و $f_1 = 0/01$.
 بعد از ۳ مرحله تکرار الگوریتم مقادیر بهینه به صورت $q_7^+ = 0/05$ ، $\hat{w}_1 = 0$ ، $q_1^- = 0/05$ و $f_1 = 0/04$ و $\hat{w}_7 = 0/15$ ، $0/05$ به دست می‌آیند.

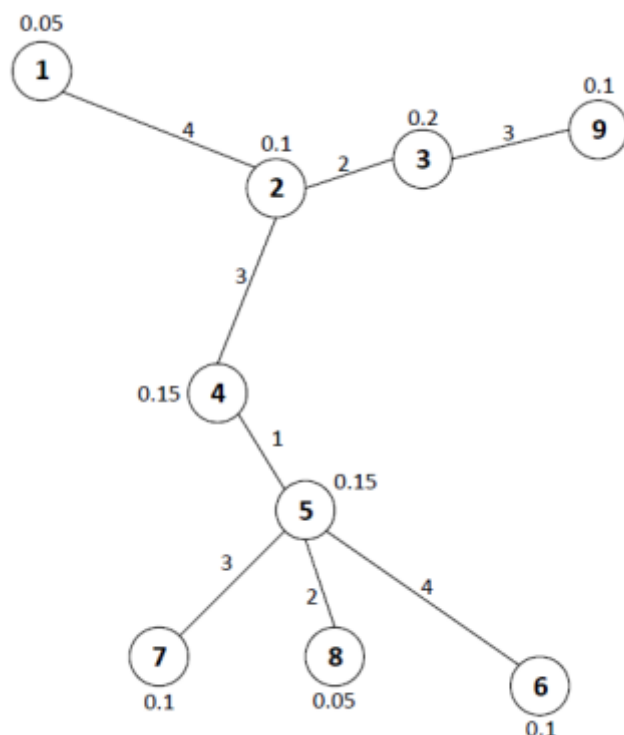
مثال ۲.۲.۷. یک مسیر با ۱۰ نقطه را (شکل ۲.۷) در نظر بگیرید. هزینه هر واحد تغییر در وزن رئوس نیز در جدول ۲.۷ بیان شده است. قرار می‌دهیم $\{m_1 = v_4, m_2 = v_9\}$.

جدول ۲.۷: هزینه تغییرات در وزن رئوس

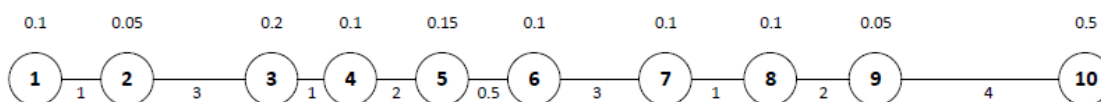
v_i	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
c_i^+	۰/۲	۰/۱	۰/۳	۰/۱	۰/۵	۰/۴	۰/۳	۰/۷	۰/۲	۰/۴
c_i^-	۰/۳	۰/۲	۰/۱	۰/۴	۰/۴	۰/۶	۰/۵	۰/۸	۰/۱	۰/۲

با مرتب کردن هزینه‌های تغییر در وزن رئوس به صورت صعودی داریم:

$$r_1 = c_1^+ \leq r_2 = c_2^+ \leq r_3 = c_3^- \leq r_4 = c_4^- \leq r_5 = c_1^+ \leq r_6 = c_9^+ \leq r_7 = c_2^- \leq r_8 = c_{10}^- \leq r_9 = c_3^+ \leq r_{10} = c_7^+ \leq r_{11} = c_1^- \leq r_{12} = c_6^+ \leq r_{13} = c_{10}^+ \leq r_{14} = c_4^- \leq r_{15} = c_5^- \leq r_{16} = c_5^+ \leq r_{17} = c_7^- \leq r_{18} = c_6^- \leq r_{19} = c_7^+ \leq r_{20} = c_8^-.$$



شکل ۱.۷: درخت T با ۹ راس



شکل ۲.۷: یک مسیر با ۱۰ راس.

$$V_1 = \{v_1, v_2, v_3, v_4, v_5, v_6\}$$

$$V_2 = \{v_7, v_8, v_9, v_{10}\}$$

بنابراین با اجرای الگوریتم IE2FLP، و با یک بار تکرار به تعادل می‌رسیم. در این صورت مقدار تغییرات وزن رئوس $q^+ = 0.05$ می‌باشد. این تغییر هزینه‌ای معادل 0.005 را به ما تحمیل می‌کند. حل مسئله بالا با استفاده از نرم افزار Matlab R2017a، مقدار $q^- = 0.05$ را می‌دهد که هزینه این تغییر نیز 0.005 می‌باشد.

۳.۷ معکوس محدودیت بودجه‌ای ۲ - سرویس‌دهنده متعادل

در حالت معکوس نوع محدودیت بودجه‌ای می‌خواهیم با استفاده از یک بودجه محدود ($B > 0$) و با تغییر در وزن رئوس تا حد امکان مسئله را به سمت بهینه شدن (متعادل) پیش ببریم. بنابراین هدف در حالت معکوس نوع محدودیت بودجه‌ای این است که با استفاده از بودجه محدود $B > 0$ تا حد امکان اختلاف مجموع وزن‌های اختصاص داده شده به سرویس‌دهنده‌های m_1 و m_2 تا حد امکان کمینه گردد. مدل این مسئله به صورت زیر نوشته می‌شود:

$$P_2 : \quad \min f_{\gamma} = \left| \sum_{i \in V_1} (w_i + q_i^+ - q_i^-) - \sum_{i \in V_2} (w_i + q_i^+ - q_i^-) \right| \quad (12.7)$$

s.t.

$$\sum_{i=1}^n (c_i^+ q_i^+ + c_i^- q_i^-) \leq B, \quad (13.7)$$

$$0 \leq q_i^+ \leq u_i, \quad i = 1, 2, \dots, n. \quad (14.7)$$

$$0 \leq q_i^- \leq w_i, \quad i = 1, 2, \dots, n. \quad (15.7)$$

قرار می‌دهیم:

$$y = \left| \sum_{i \in V_1} (w_i + q_i^+ - q_i^-) - \sum_{i \in V_2} (w_i + q_i^+ - q_i^-) \right|.$$

بنابراین مدل مسئله به مدل برنامه‌ریزی خطی زیر تبدیل می‌شود:

$$P_2 : \quad f_{\gamma} = \min y \quad (16.7)$$

s.t.

$$y \geq \sum_{i \in V_1} (w_i + q_i^+ - q_i^-) - \sum_{i \in V_2} (w_i + q_i^+ - q_i^-), \quad (17.7)$$

$$y \geq - \sum_{i \in V_1} (w_i + q_i^+ - q_i^-) + \sum_{i \in V_2} (w_i + q_i^+ - q_i^-), \quad (18.7)$$

$$\sum_{i=1}^n (c_i^+ q_i^+ + c_i^- q_i^-) \leq B, \quad (19.7)$$

$$0 \leq q_i^+ \leq u_i, \quad i = 1, 2, \dots, n. \quad (20.7)$$

$$0 \leq q_i^- \leq w_i, \quad i = 1, 2, \dots, n. \quad (21.7)$$

به‌طور مشابه معکوس نوع محدودیت بودجه‌ای مسئله ۲- سرویس‌دهنده متعادل را نیز می‌توان با یک الگوریتم از مرتبه $O(n \log n)$ حل نمود. در ادامه جزئیات این الگوریتم آورده شده است.

اگر $W(V_1) = W(V_2)$ در این صورت برای هر $i = 1, \dots, n$ و $q_i^+ = 0$ و $q_i^- = 0$ ، جواب بهینه مسئله است و نیازی به تغییر در وزن رئوس نیست. در غیراین صورت بدون کاستن از کلیت مسئله فرض می‌کنیم $W(V_1) > W(V_2)$. برای رسیدن به مقدار بهینه باید از وزن $W(V_1)$ کاسته شود یا به وزن $W(V_2)$ افزوده شود و این کار باید با کمترین هزینه و با محدودیت بودجه B انجام شود. با نمادگذاری مشابه قسمت ۲.۷ قرار می‌دهیم $D = \{r_1, r_2, \dots, r_n\}$ ، که مجموعه مرتب شده هزینه‌های c_1^+, \dots, c_n^+ و c_1^-, \dots, c_n^- باشد.

اگر $r_1 = c_k^+$ و $v_k \in V_2$ ، در این صورت قرار می‌دهیم:

$$q_k^+ = \min\left\{\frac{B}{c_k^+}, W(V_1) - W(V_2), u_k^+\right\}.$$

همچنین اگر $v_k \in V_1$ ، در این صورت نمی‌توان وزن راس v_k را افزایش داد زیرا با این کار از حالت بهینه (تعادل) دور خواهیم شد. بنابراین r_2 را در نظر می‌گیریم. از طرف دیگر در حالتی که $r_1 = c_k^-$ ، اگر $v_k \in V_1$ ، در این صورت قرار می‌دهیم:

$$q_k^- = \min\left\{\frac{B}{c_k^-}, W(V_1) - W(V_2), w_k\right\}.$$

و اگر $v_k \in V_2$ ، به سراغ r_2 خواهیم رفت. با ادامه این روش برای همه $2n$ تکرار، جواب بهینه به دست خواهد آمد. الگوریتم زیر می‌تواند برای حل معکوس نوع محدودیت بودجه‌ای مسئله ۲- سرویس‌دهنده متعادل به کار برده شود.

Algorithm [RE2FLP]⁸.

Input: The weighted graph G , two vertices m_1 and m_2 of G as location of facilities, the total budget B , and the cost of increasing and decreasing of vertices weights.

Output: The new weights of vertices \hat{w}_i for minimizing the difference weights of vertices which assigned to facilities in m_1 and m_2 .

Initialization:

Set $V_1 = \{v_i \in V | d(v_i, m_1) \leq d(v_i, m_2)\}$ and $V_2 = V \setminus V_1$.

If $W(V_1) = W(V_2)$ or $B = 0$ then **Stop**, the current weights are optimal.

If $W(V_1) > W(V_2)$ then set $\hat{V}_1 = V_1$ and $\hat{V}_2 = V_2$,

⁸Reverse Equality 2-Facility Location Problem

Else set $\hat{V}_1 = V_2$ and $\hat{V}_2 = V_1$.

Sort the cost of changing vertices weights, i.e. c_1^+, \dots, c_n^+ and c_1^-, \dots, c_n^- , in an increasing order and called them r_1, \dots, r_{2n} .

Iteration counter $i := 0$.

(For any vertex v_i in \hat{V}_1 or \hat{V}_2 , let \hat{w}_i be the weight of v_i).

Set $f_2 := W(\hat{V}_1) - W(\hat{V}_2)$ and for $i = 1, \dots, n$, $\hat{w}_i = w_i$.

Iteration step:

While the $W(\hat{V}_1) \neq W(\hat{V}_2)$ and $B \neq 0$, **do** the following:

1. **If** $r_i = c_k^+$ and $v_i \in \hat{V}_2$ **then** set

$$(a) \ q_k^+ := \min\left\{\frac{B}{c_k^+}, W(\hat{V}_1) - W(\hat{V}_2), u_k^+\right\},$$

$$(b) \ \hat{w}_k := w_k + q_k^+,$$

$$(c) \ B = B - c_k^+ q_k^+,$$

$$(d) \ f_2 := f_2 - q_k^+.$$

End if

2. **If** $r_i = c_k^-$ and $v_i \in \hat{V}_1$ **then** set

$$(a) \ q_k^- := \min\left\{\frac{B}{c_k^-}, W(\hat{V}_1) - W(\hat{V}_2), w_k\right\},$$

$$(b) \ \hat{w}_k := w_k - q_k^-,$$

$$(c) \ B = B - c_k^- q_k^-,$$

$$(d) \ f_2 := f_2 - q_k^-.$$

End if

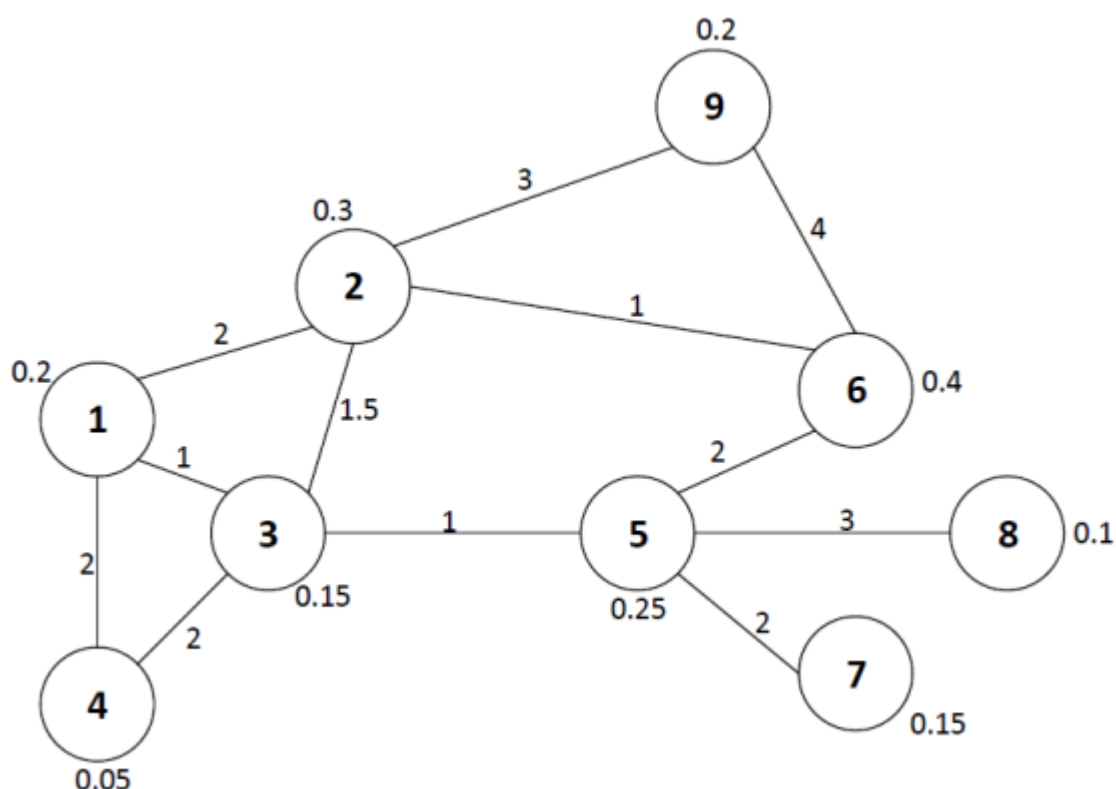
3. **Set** $i := i + 1$.

end while.

پیچیدگی زمانی این الگوریتم نیز مشابه الگوریتم IE2FLP است. بنابراین قضیه زیر را می‌توان در مورد الگوریتم RE2FLP بیان کرد.

قضیه ۱.۳.۷. معکوس نوع محدودیت بودجه‌ای مسئله ۲- سرویس‌دهنده با در نظر گرفتن تعادل بین سرویس‌دهنده‌ها، در زمان $\mathcal{O}(n \log n)$ قابل حل است.

مثال ۱.۳.۷. گراف G را به صورت شکل ۳.۷ در نظر بگیرید مجموعه هزینه‌های حاصل



شکل ۳.۷: گراف G با ۹ راس

از افزایش یا کاهش وزن هر راس در جدول ۳.۷ آمده است.

جدول ۳.۷: هزینه حاصل از تغییر وزن رئوس در گراف G .

v_i	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
c_i^+	۱	۱/۵	۱/۲	۱	۱/۳	۱/۱	۱/۲	۱/۳	۱/۲
c_i^-	۱	۱/۵	۱/۲	۱	۱/۳	۱/۱	۱/۲	۱/۳	۱/۲

فرض کنید بودجه مورد نظر برای تغییر در وزن رئوس $B = 3/10$ باشد و همچنین

$m_1 = v_2$ و $m_2 = v_5$. در این صورت داریم:

$$V_1 = \{v_1, v_2, v_6, v_9\},$$

$$V_2 = \{v_3, v_4, v_5, v_7, v_8\},$$

چون $W(V_1) = 1/1$ و $W(V_2) = 0/7$ ، لذا بعد از ۲ تکرار بودجه مورد نظر به صفر رسیده و داریم

$$q_1^- = 0/2, q_4^+ = 0/1, W(\hat{V}_1) = 0/9, W(\hat{V}_2) = 0/8, f_2 = 0/1.$$

مراجع

- [۱] سهامی ح، آمایش و مکانیابی، (۱۳۸۶) چاپ اول، دانشگاه صنعتی مالک اشتر، پژوهشکده مهندسی پدافند غیر عامل.
- [2] Alizadeh B., Bakhteh S., (2017), A modified firefly algorithm for general inverse p -median location problems under different distance norms, **Opsearch**, 54, 618-636.
- [3] Alizadeh B., Burkard R. E., (2009), Inverse 1-center location problems with edge length augmentation on trees, **Computing**, 86, 331–343.
- [4] Alizadeh B., Burkard R. E., (2012), A linear time algorithm for inverse obnoxious center location problems on networks, **Central European Journal of Operations Research**, 21, 585-594.
- [5] Alizadeh B., Burkard R. E., (2011), Combinatorial algorithms for inverse absolute and vertex 1-center location problems on trees, **Networks**, 58, 190–200.
- [6] Alizadeh B. , 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.
- [7] Alizadeh B., Etemad R., (2018), Optimal algorithms for inverse vertex obnoxious center location problems on graphs, **Theoretical Computer Science**, 707, 36-45.
- [8] Alizadeh B., Etemad R., (2016), Linear time optimal approaches for reverse obnoxious center location problems on networks, **Optimization**, 65, 2025-2036.

- [9] Alsuwaiyel M. H. (1999), **Algorithm: Design Techniques and Analysis**, World Scientific, New Jersey.
- [10] Balcik B., and Beamon B. M., (2008), Facility location in humanitarian relief, **International Journal of Logistics Research and Applications**, 11 (2), 101 - 121.
- [11] Barbati M., (2015), Piccolo C., Equality measures properties for location problems, **Optimization Letters**, 10, 903-920.
- [12] Baroughi Bonab F., Burkard R. E., Alizadeh B., (2010), Inverse median location problems with variable coordinates, **Central European Journal of Operations Research**, 18, 365-381.
- [13] Baroughi Bonab F., Burkard R. E., Gassner E., (2011), Inverse p -median problems with variable edge lengths, **Mathematical Methods of Operations Research**, 73, 263–280.
- [14] Bazaraa M.S., Sherali H., Jarvis J., (2005), **Linear programming and network flows**, third ed. John Wiley and Sons.
- [15] Benkoczi R., Breton D., Bhattacharya B., (2006), Efficient computation of 2-medians in a tree network with positive/negative weights. **Discrete Mathematics** 306, 1505-1516.
- [16] Beresnev V., (2013), Branch-and-bound algorithm for a competitive facility location problem, **Computers and Operations Research**, 40 (8), 2062-2070.
- [17] Berman O., IngcoD. I., Odoni A. R. , (1994), Improving the location of minmax facility through network modification, **Networks**, 24, 31-41.
- [18] Berman O., Drezner Z., Tamir A., Wesolowsky G.O., (2009), Optimal location with equitable loads, **Annals of Operations Research**, 167, 307-325.

- [19] Berman O., Ingco D. I., Odoni A. R., Improving the location of minisum facility through network modification, **Annals of Operations Research**, 40, 1-16.
- [20] Blanquero R., and Carrizosa E., (2009), Continuous location problems and Big Triangle Small Triangle: constructing better bounds, **Journal of Global Optimization**, 45 (3), 389-402.
- [21] Brotcorne L., Laporte G., and Semet F., (2002) Fast heuristics for large scale covering location problems, **Computers and Operations Research**, 29, 651 - 665 .
- [22] Burkard R.E., Çela E., and Dollani H., (2000), 2-Median in trees with pos/neg weights. **Discrete Appl. Math.**, 105, 51-71.
- [23] Burkard R.E., Fathali J., (2007), A polynomial time method for the pos/neg weighted 3-median problem on a tree, **Math. Methods Oper. Res.**, 65, 229-238.
- [24] Burkard R.E., Fathali J., Taghizadeh H., (2007), The p-maxian problem on a tree”, **Oper. Res. Lett.** 35, 331-335.
- [25] Burkard R. E., Galavii M., Gassner E., (2010), The inverse Fermat-Weber problem, **European Journal of Operation Research**, 206, 11-17.
- [26] Burkard R. E., Gassner E., Hatzl J., (2006), A linear time algorithm for the reverse 1-median problem on a cycle, **Networks**, 48, 16-23.
- [27] Burkard R. E., Gassner E., Hatzl J., (2008), Reverse 2-median problems on trees, **Discrete applied mathematic**, 156, 1963-1976.
- [28] Burkard R.E., Hatzl J., (2010), Median problems with positive and negative weights on cycles and cacti” , **J. Comb. Optim.**,20, 27-46.
- [29] Burkard, R.E. and Krarup J., (1998), A linear algorithm for the pos/neg-weighted 1-median problem on a cactus, **Computing**, 60: 193-215.

- [30] Burkard R. E., Pleschiutschnig C., Zhang J., (2004), Inverse median problems, **Discrete Optimization**, 1, 23–39.
- [31] Burkard R. E., Pleschiutschnig C., Zhang J., (2007), The inverse 1-median problem on a cycle, **Discrete Optimization**, 5, 242–253.
- [32] Cabello S., Díaz-Báñez J. M., Langerman S., Seara C., and Ventura I., (2010), Facility location problems in the plane based on reverse nearest neighbor queries, **European Journal of Operational Research**, 202, 99–106 .
- [33] Cai M. C., Yang X. G., Zhang J. Z., (1999), The complexity analysis of the inverse center location problem, **Journal of Global Optimization**, 15, 213–218.
- [34] Campbell J. F., (1994), Integer programming formulations of discrete hub location problems, **European Journal of Operational Research**, 72, 387 - 405.
- [35] Cappanera P., (1999), **A survey on obnoxious facility location problems**, Technical Report: TR-99-11, Department of Computer Science, University of Piza.
- [36] Charch R.L., Garfinkel R.S., (1978), Location an obnoxious facility on a network, **Trans. Sci.**, 12: 107-118.
- [37] Contreras I., Fernández E., (2012), General network design: A unified view of combined location and network design problems, **European Journal of Operational Research**, 219, 680-697.
- [38] Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. (2001), **Introduction to Algorithms**, MIT Press, Cambridge.
- [39] Cheriyan J., Ravi R., (1998), **Aproximation for Networks Problems**.
- [40] Corra E.S., Steiner M.T.A. Freitas A.A., Cornieri C., (2001), **A genetic algorithm for the p-median problem**, Available in Address: Cite-seer.nj.ncg.com/458099.html.

- [41] Current J., Daskin M., Schilling D., (1929), **Discrete network location models**, Forthcoming as Chapter 3 in: Facility Location Theory and Methods. Z.Drezner and H.Hamacher (eds.), 2001. Chicago University Press, Chicago, Illinois.
- [42] Das P. K., Behera H. S., Panigrahi B. K. (2016), A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. **Swarm and Evolutionary Computation**, 28, 14–28.
- [43] Daskin, M. S., (1995), **Network and Discrete Location: Models, Algorithms and Applications**, Wiley, NewYork.
- [44] Dibble C., Densham P.J., (1993), **Generating interesting alternatives in GIS and SDSS using genetic algorithms**, GIS/LIS Symposium, University of Nebraska, Lincoln.
- [45] Dominik Jena S., Cordeau J. F., and Gendron B. , (2017), Lagrangian Heuristics for Large-Scale Dynamic Facility Location with Generalized Modular Capacities, **INFORMS Journal on Computing**, 29 (2), 388 - 404 .
- [46] Drezner Z., and Wesolowsky G. O., (1978), Facility location on a sphere, **The Journal of the Operational Research Society**, 29 (10), 997-1004.
- [47] Ebery J., (2001), Solving large single allocation p-hub problems with two or three hubs, **European Journal of Operational Research**, 128, 447-458.
- [48] Engelbrecht P., (2007). **Computational Intelligence An Introduction**. Wiley, Second Edition.
- [49] Erkut E., Bozkaya B., Zhang J., (2001), **An effective genetic algorithm for the p-median problem**.

- [50] Ernst A. T. , and Hrishnamoorthy M., (1998), Exact and heuristic algorithms for the uncapacitated multiple allocation p - hub median problem, **European Journal of Operational Research**, 104, 100-112.
- [51] Etemad R., Alizadeh B., (2016), Some variants of reverse selective center location problem on trees under the Chebyshev and Hamming norms, **Yugoslav Journal of Operations Research**, 27, 367-384,
- [52] Etemad R., Alizadeh B., (2017), Combinatorial Algorithms for reverse selective undesirable center location problems on cycle graphs, **Journal of the Operations Research Society of China**, 5, 347-361.
- [53] Etemad R., Alizadeh B., (2018), Reverse selective obnoxious center location problems on tree graphs, **Mathematical Methods of Operations Research**, 87, 431-450.
- [54] Fathali J., (2006), A genetic algorithm for the p-median problem with pos/neg weights, **Applied Mathematics and Computation**, 183, 1071-1083.
- [55] Fathali, J., (2014), Backup multifacility location problem with l_p norm, **OPSEARCH**, 52, 382-391.
- [56] Fathali J., Jamalian A., (2017), Efficient methods for goal square Weber location problem, **Iranian Journal of Numerical Analysis and Optimization**, 7, 65-82.
- [57] Fathali J., and Taghizadeh Kakhki H., (2006), Solving the p-median problem with pos/neg weights by variable neighborhood search and some results for special cases, **European Journal of Operational Research**, 170, 440-462.
- [58] Fathali J., Zaferanieh M., (2018), The balanced 2-median and 2-maxian problems on a tree, arXiv:1803.10332 [math.OC].

- [59] Fathali J., Zaferanieh M. and Nezakati A., (2009), A BSSS algorithm for the location problem with minimum square error, **Advances in Operations Research**.
- [60] Francis R. L., (1946), On the location of multiple new facilities with respect to existing facilities, **The Journal of Industrial Engineering**, 15, 106-107.
- [61] Francis R., McGinnis Jr.L.F., White J.A., (1992), **Facility Layout and Location: An Analytical Approach**, Prentice Hall.
- [62] Galavii M., (2008), **Inverse 1-median problems**, Ph.D. Thesis, Institute of Optimization and Discrete Mathematics, Graz University of Technology, Graz, Austria.
- [63] Garey M.R., Johnson D.S., (1979), **Computers and intractability: A guid to the theory of NP-Completeness**, Freeman, San Francisco, California.
- [64] Garey M. R., Johnson D. S., (1979), **Computers and Intractability: A Guide to the Theory of NP-Completeness**, Freeman, San Francisco.
- [65] Gassner E., (2007), Up- and downgrading the 1-median in a network, **Technical Report**, 16, Graz University of Technology.
- [66] Gassner E., (2008), Up- and downgrading the 1-center in a network, **European Journal of Operational Research**, 198, 370–377 .
- [67] Gassner E., (2007), The inverse 1-maxian problem with edge length modification, **Journal of Combinatorial Optimization**, 16, 50–67.
- [68] 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.
- [69] Gavalec M., Hudec O., (1995), Balanced location on a graph, **Optimization**, 35, 367-372.
- [70] Gavish B., and Sridhar S., (1995), Computing the 2-median on tree networks is $O(n \log n)$ time. **Networks**, 26, 305-317.

- [71] Goldman, A.J., Dearing, P.M., (1975), **Concepts of optimal location for partially noxious facilities**, ORSA Bull. 23, 1, B31.
- [72] Galinier P., and Hertz A., (2007), Solution techniques for the large set covering problem, **Discrete Applied Mathematics**, 155, 312-326.
- [73] Guan X. C., Zhang B. W., (2012), Inverse 1-median problem on trees under weighted Hamming distance, **Journal of Global Optimization**, 54, 75–82.
- [74] Haddadi S., (1997), Simple lagrangian heuristic for the set covering problem, **European Journal of Operational research**, 99, 200-204.
- [75] Hakimi, S.L., (1964), Optimum location of switching centers and the absolute centers and medians of a graph, **Oper. Res**, 12: 450-459.
- [76] Hamacher H. W., Nickel S., (1998), classification of location models, **Location Sci.**, 6: 229-242.
- [77] Handler G. Y., Mirchandani P. B., (1979), **Location on Networks: Theory and Algorithms**, MIT Press, Cambridge.
- [78] Hansen P., Labbe M., Peeters D., Thisse J.F., (1987), Single facility location on networks, **Annals of Discrete Math**, 31: 113-146.
- [79] Hansen P., and Mladenovic N., (1997), Variable neighborhood search for the p-median, **Location Science**, 5 (4),207-226.
- [80] Handler G. Y. , (1973), Minimax location of a facility in a undirected tree network, **Transportation Science**, 7, 287-293.
- [81] Hedetniemi S. M. , Cockayne E. J., and Hedetniemi S. T., (1981), linear algorithm for finding the Jordan center and path center of a tree, **Trancs.SCI**, 15, 98-114.
- [82] Ho S. C., (2015), An iterated tabu search heuristic for the Single Source Capacitated Facility Location Problem, **Applied Soft Computing**, 27, 169 - 178 .

- [83] Ho, Y. C., Pepyne, D. L., (2001), Simple Explanation of the No Free Lunch Theorem of Optimization. **Proceeding of the 40th IEEE, Conf.** on Decision and Control, USA.
- [84] Hosage, M., Goodchild M.F., (1986), Discrete space location-allocation solutions from genetic algorithms, **Ann. Oper. Res**, 6: 35-46.
- [85] Jamalian A., and fathali J., (2009), Linear Programming for the Location Problem with Minimum Absolute Error, **World Applied Sciences Journal**, 7 (11), 1423 - 1427 .
- [86] Jianfang Y., Juan J., (2012), Reverse 1-median problem with constraint in trees, **The 2nd International Conference on Computer Application and System Modeling**.
- [87] Kalscsics J., (2011), The multi-facility median problem with pos/neg weights on general graphs, **Computers and Operations Research**, 38, 674 - 682 .
- [88] Karvin O., and Hakimi S.L., (1969) An algorithmic approach to network location problems; part 2, the p-medians, **SIAM Journal on Applied Mathematics**, 37, 539 - 560.
- [89] Karvin O., and Hakimi S. L., (1979), An algorithm approach to network location problems. Part I: The p-centers, **SIAM Journal of Applied Mathematics**, 37, 513 - 538.
- [90] Klamroth K., (2001), A reduction result for location problems with polyhedral barriers, **European Journal of Operational Research**, 130, 486 - 497 .
- [91] Ko M. T., and Lee R. C. T., (1991), On weighted rectilinear 2-center and 3-center problems, **Information Sciences**, 54, 169-190.
- [92] Korte B., Vygen J., (2012), **Combinatorial Optimization: Theory and Algorithms**, Springer, 5th Ed.

- [93] Krarup J., Pruzan P. M., (1983), The simple plant location problem: Survey and synthesis, **EJOR**, 12: 36-81.
- [94] Ku S. C. , Lu C. J., Wang B. F. , and Lin T. C., (2001), Efficient algorithms for two generalized 2-median problem on trees, , **Proceeding of the 12th International Symposium on Algorithms and Computation**, Christchurch, New Zealand, 768-778.
- [95] Kuhn, H.W., (1967), On a pair of dual nonlinear programs, **Nonlinear programming**, J.Abadie(ed.), North-Holland Publishing Co. Amesterdam, 37-54.
- [96] Landete M., Marin A., (2014), Looking for edge-equitable spanning trees, **Computers & Operations Research**, 41, 44-52.
- [97] Li S., (2013), A 1.488 approximation algorithm for the uncapacitated facility location problem, **Information and Computation**, 222, 45 - 58.
- [98] Longshu W., Joonwhoan L., Jianhua Z., Qin W., (2013), The inverse 1-median problem on tree networks with variable real edge lengths, **Mathematical Problems in Engineering**, 2013, 1-6.
- [99] Love R. F. , Morris J. D., and Wesolowsky G. O, (1988), **Facilities locations: Models and methods**, Publications in Operations Research Series, vol. 7. North-Holland, Amsterdam.
- [100] Maranzana F.E., (1964), On the location of supply points to minimize transport costs, **Oper. Res Quarterly**, 15: 261-270.
- [101] Marin A., (2011), The discrete facility location problem with balanced allocation of customers, **European Journal of Operational Research**, 210, 27-38.
- [102] Martello S., Toth P., (1990), **Knapsack Problems: Algorithms and Computer Implementations**, John Wiley, New York.

- [103] Megiddo N, Supowitz K, (1984), On the complexity of some common geometric location problems, **SIAM Journal on Computing**, 182-196.
- [104] Minika E., (1958), The optimal location of a path or tree in a tree network, **Networks**, 15, 309-321.
- [105] Mirchandani P. B, and Francis R. L., (1990), **Discrete location theory**, Wiley, New York.
- [106] Montoya A., Velez-Gallego M. C., and Villegas J. G., (2016), Multi-product capacitated facility location problem with general production and building costs, **NET-NOMICS: Economic Research and Electronic Networking**, 17 (1), 47 - 70 .
- [107] Moreno-Perez J.A., Moreno-Vega J.M., Meladenovic N., (1994), **Tabu search and simulated annealing in p-median problems**, Talk at the Canadian Operational Research Society Conference, Montral.
- [108] Nguyen K. T., (2016), Reverse 1-center problem on weighted trees, **A Journal of Mathematical Programming and Operations Research**, 65, 253-264.
- [109] Nguyen K. T., (2017), The inverse 1-center problem on cycles with variable edge lengths, **Central European Journal of Operations Research**, <https://doi.org/10.1007/s10100-017-0509-4>.
- [110] Nguyen K. T., (2016), Inverse 1-median problem on block graphs with variable vertex weights, **Journal of Optimization Theory and Applications** 168, 944-956.
- [111] Nguyen K. T., Anh L. Q., (2015), Inverse k -centrum problem on trees with variable vertex weights **Mathematical Methods of Operations Research**, 82, 19–30.
- [112] Nguyen K. T., Chassein A., (2015), Inverse eccentric vertex problem on networks, **Operations Research & Decision Theory**, 23, 687–698,

- [113] Nguyen K. T., Sepasian A. R., (2016), The inverse 1-center problem on trees with variable edge lengths under Chebyshev norm and Hamming distance **Journal of Combinatorial Optimization**, 32, 872-884.
- [114] Nguyen K. T., Vui P. T., (2016), The inverse p -maxian problem on trees with variable edge lengths, **Taiwanese Journal of Mathematics**, 20, 1437-1449.
- [115] Owen S.H., Daskin M.S., (1998), Strategic facility location: A review, **EJOR**, 111: 423-447.
- [116] Pereira M. A., Coelho L. C., Lorena L. A. N., and de L. C., (2015), A hybrid method for the Probabilistic Maximal Covering Location–Allocation Problem, **Computers and Operations Research**, 57, 51-59.
- [117] Plastria F., (2016), Up- and downgrading the euclidean 1-median problem and knapsack Voronoi diagrams, **Annals of Operations Research**, 246, 227-251.
- [118] Qin Z., and Gao Y., (2017), Uncapacitated p -hub location problem with fixed costs and uncertain flows, **Journal of Intelligent Manufacturing**, 28 (3), 705 - 716 .
- [119] Rahmani A., and MirHassani S.A., (2014), A hybrid Firefly-Genetic Algorithm for the capacitated facility location problem, **Information Sciences**, 283, 70 - 78 .
- [120] Rao S. S., (2009). **Engineering Optimization Theory and Practice (Fourth Edition)**. John Wiley and Sons, Inc.
- [121] Rolland E., Schilling D.A., Current J. R., (1997), An efficient tabu search heuristic for the p -median problem, **EJOR**, 96: 329-342.
- [122] Sahin G., and Sural H., (2007), A review of hierarchical facility location models, **Computers and Operations Research**, 34 (8), 2310 - 2331.

- [123] Saleem, Saleh M., (2011). Knowledge-Based Solution to Dynamic Optimization Problems using Cultural Algorithms. Ph.D. thesis, Wayne State University, Detroit, Michigan.
- [124] Scaparra M.P., Scutella M. G., (2001), **Facilities, locations, customers: Building blocks of location models: A survey**, Technical Report: tr-01-18, University of Piza, Italy.
- [125] Senne E. L. F., Lorena L. A. N., and Pereira M. A., (2005), A branch-and-price approach top - median location problems, **Computers and Operations Research**, 32, 1655 - 1664 .
- [126] Sepasian A. R., (2018), Upgrading the 1-center problem with edge length variables on a tree, **Discrete Optimization**, doi:10.1016/j.disopt.2018.02.002.
- [127] Sepasian A. R., Rahbarnia F., (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.
- [128] Sepasian A. R., Rahbarnia F., (2015), Upgrading p -median problem on a path, **Journal of Mathematical Modelling and Algorithms in Operations Research**, 14, 145-157.
- [129] Slater P. J., (1982), Locating central path in a graph, **Transportation Science**, 16, 1-18.
- [130] Snyder L. V., and Daskin M. S., (2005), Reliability models for facility location: the expected failure cost case, **Transport Sci**, 39, 400-416.
- [131] Tamir A., (1996), An $o(p n^2)$ algorithm for the p -median and related problems on tree graphs, **Oper. Res Letters**, 19: 59-64.

- [132] Tansel B. C., Francis R.L., Lowe T. J., (1983), Location on networks: A survey, **Management Sci.**, 29: 482-511.
- [133] Ting S.S. (1984), A linear-time algorithm for maximum facility location on tree networks, **Transportation Science**, 18, 76-84.
- [134] Teitz M. B., and Bart P., (1968), Heuristic methods for estimating the generalized vertex median of a weighted graph, **Operations Research**, 16, 955 - 961.
- [135] Teye C., HBell M. G., and JBliemer M. C., (2017), Urban intermodal terminals: The entropy maximising facility location problem, **Transportation Research Part B: Methodological**, 100, 64 - 81 .
- [136] Uno T. , and Katagiri H., (2008), Single- and multi-objective defensive location problems on a network, **European Journal of Operational Research**, 188 (1), 76-84.
- [137] Wang Q., Bai Y., (2010), An efficient algorithm for reverse 2-median problem on a cycle. **JNW**, 5, 1169-1176.
- [138] Wang H. L., Wu B. Y., and Chao K. M., (2009), The backup 2-center and backup 2-median problems on trees, **Networks**, 53, 39-49.
- [139] Ward J.E., Wendell R.E., (1980), A new norm for measuring distance which yields linear location problems, **Oper. Res.**, 28: 836-844.
- [140] Ward J.E., Wendell R.E., (1985), Using block norms for location modelling, **Operations Research**, 33: 1074-1090.
- [141] Weber, A., (1929), **Über den Standort der Industrien**, Tübingen, 1909; English Trans.: Theory of Location of Industries, (C.J.Friedrich, ed.and trans.), Chicago University Press, Chicago, Illinois.

- [142] Weiszfeld E., (1973), Sur le point pour lequel la somme des distances denpoints donnés est minimum, **Tôhoku Mathematical Journal**, 43, 355-386.
- [143] Wesolowsky G. O., (1974), Location of the median line for weighted points, **Environment and planning**, 7, 163-170.
- [144] Wesolowsky G. O., and Love R. F., Location of facilities with rectangular distance among point and area destinations, **Naval Research Logistics Quarterly**, 18, 83-90.
- [145] Yang X., Zhang J., (2008), Inverse center location problem on a tree, **Journal of System Science and Complexity**, 21, 651–664.
- [146] Zhang J., (2006), Approximating the two-level facility location problem via a quasi-greedy approach, **Mathematical Programming**, 108 (1), 159 - 176 .
- [147] Zhang J., Liu Z., Ma Z., (2000), Some reverse location problems, **European Journal of Operational Research** 124, 77-88.

Abstract

Location theory is an important field of the operations research and has many applications in the real world. In recent years, some new versions of the facility location models, called improvement facility location, involves the inverse location, backup facility location, balanced facility location and goal location models, have found an increasing interest. In this thesis some improvement p -median problems are considered. Efficient methods based on some properties of the desirable and undesirable backup 2-median problems on a tree and the multi-facility backup goal location problem in continuous space are presented. Moreover, the inverse models of the backup 2-median problem with minimum cost and limited budgeting in discrete and continuous space are considered. The Weiszfeld like, particle swarm optimization and cultural optimization algorithms are used to solve the inverse models. Finally, two $\mathcal{O}(n \log n)$ algorithms are presented for solving the inverse and reverse balanced 2-median problems.

keywords: Inverse Problem; Reverse Problem; Backup Facility Location; Goal Location; Balanced 2-Median Problem.



Shahrood University of Technology

Faculty Of Mathematical Sciences

PhD Thesis in: Operation Research

**Backup Inverse Multifacility Location
Problems**

By: Morteza Nazari

Supervisor

Dr. Jafar Fathali

February 2019