بسم الله الرحمن الرحیم

دانشکده ریاضی
گروه ریاضی کاربردی

مسأله مکانیابی p- ماکسین

دانشجو

حبیبه قاسمی

استاد راهنما

دکتر جعفر فتحعلی

استاد مشاور

دکتر میثم علیشاهی

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

بهمن۹۱

سالها خواندم و خواندم کتاب دیگران
تا که امروز بتوانم مطلبی آرم به زبان

همه حاصل عمرم همین هاست همین
پیشکش به مهر مادر و پدرم با دل و جان

پیشکش به یاری همسرم و صبر و دعاش
در تک تک لحظه ها به من داد توان

تقدیم به استاد گرانقدر و صبور
که آموخت به من آنچه بیاموخت به جان

تقدیر وتشکر

سپاس همواره زیبنده اوست که در آدمی میل به آموختن نهاد و سپس سپاس شایسته نام آنان است که در لباسی از تار مهر و پود علم به من آموختند چگونه بیندیشم.

در به پایان رسیدن این تحقیق بعد از تشکر از خدای خود، لازم می دانم از رهنمود های بسیار ارزشمند استاد راهنما وگرانقدر جناب آقای دکتر جعفر فتحعلی که زحمت هدایت و راهنمایی بنده را بر عهده داشتند و مرا در هر مرحله از تحقیق یاری نمودند وهمچنین از جناب آقای دکتر میثم علیشاهی که زحمت مشاوره بنده را تقبل فرموده کمال تشکر وقدردانی را داشته باشم، باشد که در تمام عرصه های فرارو، موفقیت های لازم را کسب نمایند.

چکیده

در سالهای اخیر، علاقه مندی زیادی در مورد مسائل مکانیابی تسهیلات مضر به وجود آمده است که در این مسأله یک یا چند سرویس دهنده تا آنجا که امکان داشته باشد دور از مشتریان قرار می گیرند. مسأله p-ماکسین یکی از مسائل مکانیابی تسهیلات مضر می باشد. این پایان نامه با هدف بررسی مسأله p-ماکسین روی درخت و گراف بازه ای و راههایی که برای حل آن ارائه شده، صورت گرفته است.

در ابتدا مسائل مکانیابی را تعریف کرده و به بررسی مسأله ۱-ماکسین روی درخت می پردازیم. در فصل سوم مسأله p-ماکسین روی درخت را مطرح می کنیم و نشان می دهیم که جواب بهینه این مسأله به وسیله دو برگ از طولانی ترین مسیر در درخت به دست می آید، سپس نتیجه به دست آمده را به مسأله p-ماکسین روی درخت تعمیم می دهیم.

در فصل چهارم به معرفی گراف های بازه ای می پردازیم و مسأله p-ماکسین روی گراف های بازه ای که هر بازه دارای یک وزن مثبت است را بررسی می کنیم. ابتدا یک الگوریتم خطی برای حل مسأله ۱-ماکسین وزن دار ارائه می کنیم. برای مسأله ۲-ماکسین، نشان می دهیم که دو بازه با می نیمم نقطه پایانی راست و ماکزیمم نقطه پایانی چپ یک جواب بهینه هستند که می توان آنرا به مسأله p-ماکسین تعمیم داد.

کلمات کلیدی : نظریه مکانیابی، مسأله p-میانه، مسأله p-ماکسین، تسهیلات مضر، گراف بازه ای.

فهرست مطالب

فصل اول : مسائل مکانیابی

فصل دوم :الگوریتم خطی برای مسأله ۱-ماکسین روی درخت

فصل سوم : مسأله p-ماکسین روی درخت

فصل چهارم : مسأله p-ماکسین روی گراف های بازه ای

فهرست اشکال

فهرست جداول

فصل اول
مسائل مکانیابی

۱.۱ مقدمه

نظریه مکانیابی یکی از شاخه های اساسی تحقیق در عملیات است و نقش مهمی را در آن ایفا می کند. شاید بتوان پیدایش مسأله مکانیابی را به زمانی نسبت داد که در قرن هفدهم فرما[1] مسأله زیر را مطرح کرد: فرض کنید سه نقطه در صفحه داده شده است، نقطه چهارم را به گونه ای بیابید که مجموع فاصله های آن تا سه نقطه داده شده می نیمم شود. توریچلی[2] در ۱۶۴۰ این مسأله را حل کرده است بدین دلیل نقطه بهینه را نقطه توریچلی می نامند. اولین تعریف مسأله مکانیابی به صورت کاربردی در ۱۹۰۹ توسط وبر[3] [۲۸] ارائه شد. اما مطالعات جدی بر روی این مسأله از زمانی شروع شد که در سال ۱۹۶۴ حکیمی[4] [۲] تابع هدف را به صورت کمترین مجموع و مینیماکس مطرح کرد. او همچنین بررسی هایی بر روی مسائل مکانیابی روی شبکه ها انجام داد. اولین طبقه بندی مدلهای مختلف مکانیابی توسط هندلر[5] و میرچندانی[6] [۳۱] ارائه شد. پس از آن، طبقه بندی های دیگری از جمله توسط هاماچر[7] و نیکل[8][۳۰] انجام شد. همچنین بررسی هایی از کارهای انجام شده در این زمینه ارائه شد که از آن جمله تانسل[9] و همکاران [۳۲]، کرارپ[10] و پروزن[11] [۳۳]، هانسن[12] و همکاران[۳۴]، اون[13] و دسکین[14][۳۶]، لیستی از کارهای انجام شده در زمینه های مختلف ارائه کرده اند. کارنت[15] و همکاران نیز کاربردهای مسائل مکانیابی را [۳۵] بیان نموده اند.

---

[1] *Fermat*

[2] *Torricelli*

[3] *Weber*

[4] *Hakimi*

[5] *handler*

[6] *Michandani*

[7] *hamacher*

[8] *Nickel*

[9] *Tansel*

[10] *Krarup*

[11] *Pruzan*

[12] *Hansen*

[13] *Owen*

۲.۱ تعاریف اولیه

تعریف ۱,۱. فرض کنید V مجموعه ای متناهی و غیرتهی و E زیرمجموعه ای از مجموعه تمام زیرمجموعه های دوعضوی V است. در این صورت به جفت $G=(V,E)$ یک گراف می گوییم. هر عضو V را یک رأس و هر عضو E را یک یال می نامیم.

تعریف ۲,۱. گراف G را کامل نامیم هرگاه بین هر جفت از رئوس آن یالی موجود باشد.

تعریف ۳,۱. درجه یک رأس در یک گراف عبارت است از تعداد یالهایی که از آن رأس می گذرد.

تعریف ۴,۱. مسیر بسته ای که ابتدا و رأس های داخلی آن متمایز باشند را یک دور در گراف می نامیم.

تعریف ۵,۱. گراف G را همبند گوییم اگر هر دو رأس آن با کوتاهترین مسیر در G متصل شوند.

تعریف ۶,۱. یک گراف فاقد دور و همبند را درخت نامیم.

تعریف ۷,۱. برای گراف همبند G رأس v یک رأس برشی است اگر $G - v$ ناهمبند باشد.

تعریف ۸,۱. زیرگراف همبند ماکسیمال تولید شده در G که فاقد رأس برشی باشد، بلوک نامیده می شود.

تعریف ۹,۱. گراف G یک گراف بلوکی است هرگاه هر بلوک در G کامل باشد یعنی هر دو رأس در یک بلوک وابسته باشند.

---

[14] *Daskin*

[15] *Current*

تعریف ۱٫۱۰. به دوری با اندازه ی بزرگ تر از ۳ که هیچ یالی بین رئوس غیر متوالی در آن دور نباشد، حفره گویند.

تعریف ۱٫۱۱. فرض کنیم مجموعه ای از بازه های باز داریم. اگر این بازه ها را به عنوان رئوس و اتصال دو رأس را، به شرط ناتهی بودن اشتراک بازه های متناظر، یال ها در نظر بگیریم، گرافی می توان رسم کرد که به آن گراف بازه ای می گوییم.

تعریف ۱٫۱۲. کاکتوس گرافی است که در آن هر دو دور حداکثر دارای یک رأس مشترک باشند.

۳.۱ مسائل مکانیابی

مسأله مکانیابی ابتدا توسط فرما مطرح شد و ما آن را در مقدمه بیان کردیم. تعمیمی از آن، که به مسأله فرما-وبر معروف است، به این صورت است که فرض کنید n نقطه $p_1, p_2,\dots,p_n$ در صفحه موجودند و به ترتیب دارای وزنهای $w_1, w_2,\dots,w_n$ هستند، می خواهیم نقطه ای مانند x را به گونه ای بیابیم که مجموع وزنی فاصله x تا نقاط موجود می نیمم شود. یعنی اگر فاصله x تا p را به صورت $d(x,p)$ نمایش دهیم آنگاه مسأله به صورت زیر خواهد بود :

$$min\sum_{i=1}^{n} w_i d(x,p_i)$$

مسأله فوق به مسأله تک وسیله ای با کمترین مجموع معروف است. همچنین اگر هدف پیدا کردن x به گونه ای باشد که فاصله وزنی x تا دورترین نقطه موجود می نیمم شود، آنگاه مسأله را مسأله تک وسیله ای مینیماکس گویند که به صورت زیر می باشد ،

$$min \max_{i=1,..n} w_i d(x,p_i)$$

اگر به جای یک نقطه به دنبال چند نقطه باشیم یعنی چند مکان برای وسایل جدید به گونه ای بیابیم که نزدیکترین نقطه را به نزدیکترین وسیله نسبت داده و بخواهیم مجموع وزنی فاصله ها را می نیمم کنیم، مسأله را مسأله چند وسیله ای با کمترین مجموع و در حالت دیگر آنرا مسأله چند وسیله ای مینیماکس گویند. در این حالت اگر $X=\{x_1,x_2,\ldots,x_k\}$ مجموعه وسایل جدید باشد که باید مکانیابی شوند و $d(X,p)=\min_{x_i\in X} d(x_i,p)$ آنگاه تابع هدف در مسائل کمترین مجموع و مینیماکس به ترتیب می تواند به صورت زیر نوشته شود،

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

و

$$\min G(X)=\max_{i=1,\ldots,n} w_i d(X,p_i)$$

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

در بحث فوق اگر مکان وسایل جدید بتواند در هر نقطه ای از صفحه قرار گیرد آنگاه مسأله را مسأله مکانیابی پیوسته گویند. برای این حالت راه حلهایی برای وقتی که تابع فاصله یعنی $d(x,p)$ به صورت یکی از نرمهای خطی اقلیدسی و چبیشف می باشد در[۱] ارائه شده است.

در حالت گسسته یک شبکه مانند $N=(V,E)$ موجود است که تابع فاصله روی آن تعریف می شود و مکان وسایل جدید نیز باید نقطه ای از شبکه باشد. در این حالت وقتی تابع هدف به صورت کمترین مجموع یا مینیماکس است، به ترتیب مسأله را مسأله میانه یا مرکز می نامند. همچنین اگر هدف پیدا کردن مکان p وسیله جدید باشد، آنگاه این مسائل به ترتیب مسأله p-میانه یا p-مرکز نامیده می شوند.

اولین افرادی که به مسأله مکانیابی گسسته با تابع هدف کمترین مجموع توجه کردند کوهن[۱۶] و هامبرگر[۱۷][۴]، حکیمی[۲]، مان[۱۸] [۵] و بالینسکی[۱۹] [۳] بودند. اما این حکیمی بود که اصطلاح p-میانه را ابداع کرد.

۴.۱ مسأله p-میانه

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

مسأله p-میانه روی شبکه به صورت زیرمی باشد. گراف $G=(V,E)$ را که یک گراف همبند بدون جهت با مجموعه رئوس V، $|V|=n$ و مجموعه یال E، $|E|=m$ باشد، در نظر بگیرید به گونه ای که هر رأس v_i از آن متناظر با یک مشتری و دارای وزن نامنفی w_i می باشد. همچنین $X=\{x_1,x_2,\dots,x_p\}$ مجموعه ای شامل p متغیر که باید مکانیابی شوند باشد، $d(x,v)$ طول کوتاهترین مسیر بین نقطه x و رأس v باشد، آنگاه مسأله p-میانه می تواند به صورت زیر نوشته شود :

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

در واقع در مسأله p-میانه ما می خواهیم یک مجموعه X شامل p سرویس دهنده روی یال ها یا رئوس G به گونه ای بیابیم که مجموع فاصله وزنی از همه مشتری ها تا نزدیک ترین سرویس دهنده می نیمم شود.

---

[16] Kuehn

[17] Hamburger

[18] Manne

[19] Balinski

در مسأله فوق $x \in X$ می تواند هر نقطه ای از شبکه باشد، یعنی ممکن است یک رأس یا یک نقطه از G باشد که در این حالت مسأله را مسأله p- میانه محض گویند. اما اگر هدف پیدا کردن نقاط میانه روی رأسها باشد، مسأله را مسأله p- میانه رأسی نامند.

در مسأله p-میانه کلاسیک همه وزنهای w_i نا منفی هستند. کریو[۲۰] و حکیمی [۶] نشان دادند که مسأله p- میانه کلاسیک در گراف ها یک مسأله NP- سخت می باشد. برای مواردی که گراف یک درخت باشد، این نویسندگان الگوریتمی با زمان $O(p^2n^2)$ طراحی کردند. بعدا تمیر[۲۱][۷] پیچیدگی زمانی مسأله p-میانه روی درخت را به $O(pn^2)$ بهبود بخشید.

۱. ۵ مسأله میانه با وزن منفی روی رأس ها

در مسأله p-میانه اغلب فرض بر این است که وزن رئوس مثبت باشد، اما حالتی را درنظر بگیرید که ایجاد وسیله جدید روی شبکه برای مشتریها ناخوشایند است و دور بودن وسیله جدید از بعضی نقاط و نزدیکی به بعضی نقاط دیگر مورد نظر است. در این حالت می توان برای حل مسأله به نقاطی که باید از وسیله جدید دور باشند وزن منفی داد و به بقیه وزن مثبت و سپس مسأله را حل کرد. از کاربردهای این مسأله می توان به تعیین مکان نیروگاههای هسته ای، انبارهای نظامی و انبارهای زباله اشاره کرد.

بورکارد[۲۲] و کرارپ [۱۵] مسأله ۱-میانه با وزن مثبت و منفی روی شبکه را معرفی کردند و نشان دادند که این مسأله می تواند بطور خطی روی کاکتوس حل شود. همچنین برای شبکه های با وزن مثبت و منفی بورکارد و دلانی[۲۳] [۹] و [۸] مسأله مکانیابی را در حالت کلی و مرکز بررسی کرده

---

[20] *Kariv*

[21] *Tamir*

[22] *Burkard*

[23] *Dollani*

و الگوریتم هایی برای حالتهای مختلفی از آنها ارائه کرده اند. در حالتی که هدف پیدا کردن بیش از یک نقطه روی شبکه است بورکارد و همکاران[۱۱]دو نوع تابع هدف تعریف کرده و روشهایی برای حل مسأله ۲-میانه با وزن مثبت و منفی روی درخت ارائه کرده اند.

فرض کنید $G=(V,E)$ یک شبکه با n رأس باشد که رئوس آن دارای وزن مثبت یا منفی باشند. برای تخصیص یک رأس به یک میانه می توان به دو صورت عمل کرد : ۱- اگر وزن رأس مثبت بود آن را به نزدیکترین وسیله اختصاص دهیم و اگر منفی بود به دورترین وسیله. ۲- بدون توجه به علامت وزن رأس آنرا به نزدیکترین وسیله اختصاص دهیم. دو تابع هدف متناظر با این دو حالت به ترتیب به صورت زیر می باشند :

$$F_1(X)=\sum_{i=1}^{n}\min_{1\le j\le p} w_i d(x_j,v_i) \qquad (P_1) \tag{۱ .۱}$$

$$F_2(X)=\sum_{i=1}^{n} w_i \min_{1\le j\le p} d(x_j,v_i)\ . \qquad (P_2) \tag{۲ .۱}$$

مسائل p-میانه متناظر با توابع فوق به ترتیب عبارت است از پیدا کردن مجموعه X^* با حداکثر p نقطه به گونه ای که

$$P_1 : F_1(X^*)=\min_{x\in G} F_1(X) \tag{۳ .۱}$$

$$P_2 : F_2(X^*)=\min_{x\in G} F_2(X) \tag{۴ .۱}$$

توجه داریم که اگر تمام وزنها مثبت باشد آنگاه $F_1(X)=F_2(X)$ و هر دو مسأله تبدیل به مسأله p-میانه کلاسیک می شوند.

در مدل (P_1) مجموع از می نیمم فاصله وزنی در سراسر $X\subseteq G$ با $|X|=P$ می نیمم شده است و در مدل (P_2) مجموع وزنی می نیمم فاصله روی سراسر $X\subseteq G$ با $|X|=P$ می نیمم شده

است. بورکارد و همکاران [۱۱] مدل (P_1) و (P_2) را برای وزنهای مخلوط که تعدادی از وزنها مثبت و تعدادی منفی هستند بررسی کرده اند. آنها نشان دادند که مدل (P_1) دارای خاصیت بهینگی رأسی است (به این معنی که مجموعه X یک زیرمجموعه از مجموعه رئوس V است)، اگر G یک درخت باشد، اما مدل (P_2) برای وزنهای مخلوط یا منفی دارای این خاصیت نیست حتی اگر G یک مسیر باشد. برای حل مسأله ۲-میانه با وزنهای مثبت و منفی مدل (P_1) بورکارد و همکاران [۱۱] برای درخت یک الگوریتم با زمان $O(n^2)$ و برای مسیر یک الگوریتم با زمان خطی ارائه کردند. برای مدل (P_2) آنها یک الگوریتم با زمان $O(n^3)$ برای حل ۲-میانه روی یک درخت مطرح کردند. اگر شبکه یک مسیر باشد یا میانه به مجموعه رئوس محدود شود، آنها نشان دادند که پیچیدگی به $O(n^2)$ کاهش می یابد.

۱. ۶ مسأله P-ماکسین

یک مسأله مکانیابی با وزنهای رئوس منفی، مسأله مکانیابی تسهیلات مضر نامیده می شود در این مسأله سرویس دهنده دارای اثرات ناخوشایندی مانند سروصدا، آلودگی و ... می باشد. در مسأله p سرویس دهنده مضر ما می خواهیم p سرویس دهنده را به گونه ای قرار دهیم که مجموع فاصله وزنی منفی از مشتریان تا سرویس دهنده ها می نیمم شود. این متناظر است با این که مجموع فاصله وزنی مثبت از مشتریان تا سرویس دهنده ها ماکزیمم شود. از این رو این مسأله، مسأله p-ماکسین نامیده می شود.

مسأله p-ماکسین روی شبکه به صورت زیر تعریف می شود، گراف $G=(V,E)$ با مجموعه رئوس V و $|V|=n$ و مجموعه یالهای E، $|E|=m$ را در نظربگیرید. هر رأس v_i دارای وزن w_i میباشد که ما فرض می کنیم همه وزنهای w_i مثبت هستند. ما می خواهیم یک مجموعه $X=\{x_1,x_2,\ldots,x_p\}$ از p سرویس دهنده را روی گراف G به گونه ای بیابیم که مجموع فاصله وزنی مشتریان تا دورترین

سرویس دهنده ماکزیمم شود. که در این صورت مسأله p-ماکسین می تواند به صورت زیر نوشته شود،

$$\max F(X) = \sum_{i=1}^{n} \max_{1\le j\le p} (w_i d(x_j, v_i))$$

$$= \sum_{i=1}^{n} w_i \max_{1\le j\le p} d(x_j, v_i)$$

زلینکا[24] [۱۳] نشان داد که مجموعه برگ ها در یک درخت یک جواب بهینه برای مسأله ۱-ماکسین به دست می آورد و تینگ[25] [۱۲] یک الگوریتم با زمان خطی برای مسأله ۱-ماکسین ارائه داد. همچنین به بورکارد و کرارپ [۱۵] نگاه کنید. برای مسأله ۱-ماکسین در شبکه های عمومی، چرچ[26] و گارفینکل[27] [۱۴] یک الگوریتم با زمان $O(mnlogn)$ ارائه دادند. آنها نشان دادند که جواب بهینه به یک مجموعه از رئوس گراف G می تواند محدود شود. همچنین در گام های بعدی مشاهده کردند اگر گراف، یک درخت باشد، جواب بهینه در برگ ها اتفاق می افتد. تمیر [۱۹]پیچیدگی زمانی الگوریتم رابه $O(mn)$ بهبود بخشید.

بورکارد و همکاران [۱۸] ثابت کردند یک مسأله p-ماکسین روی یک درخت در زمان خطی قابل حل می باشد. آنها ابتدا بیان کردند که مسأله ۲-ماکسین روی یک درخت با مدل (P_1) از مسأله ۲-میانه مضر معادل است. بعد نشان دادند که یک جواب بهینه به وسیله دوبرگ از طولانی ترین مسیر درخت به دست می آید که نتیجه به p-ماکسین قابل تعمیم است. این در حالی است که برای مدل (P_2) برای مسأله p-میانه با وزن منفی جواب بهینه ممکن است هیچ برگی را شامل نشود. در ادامه با

---

[24] Zelinka

[25] Ting

[26] Church

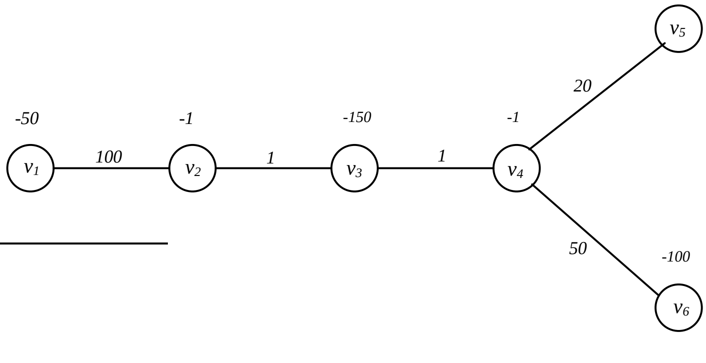
[27] Garfinkel

ذکر مثالی به بررسی این مورد می پردازیم. کانگ[28] و چنگ[29] [۱۶] و [۱۷] نیز یک الگوریتم با زمان خطی برای حل مسأله p-ماکسین روی گراف های بلوکی و گراف های بازه ای با طول یال واحد ارائه کردند.

۱. ۷ مدل P_2 برای مسأله p- میانه روی یک درخت با وزن منفی

مدل (P_2) برای $p>1$ با وزن رئوس منفی روی یک درخت را درنظر می گیریم. بر طبق چرچ و گارفینکل [۱۴] و بورکارد و همکاران [۱۱] کافی است مسأله ۱-میانه متناظر را حل کنیم و همه p سرویس دهنده را در این نقطه قرار دهیم. یک جواب بهینه با p ، $p>1$ مکان متفاوت وجود ندارد. به عبارت دیگر، اگر ما یک سرویس دهنده را در مکان ۱-میانه و $p-1$ سرویس دهنده باقی مانده را به اندازه ε فاصله حول جواب بهینه مسأله ۱-میانه قرار دهیم و ε را به سمت صفر میل دهیم، آنگاه جواب بهینه مسأله p- میانه به جواب بهینه مسأله ۱-میانه نزدیک می شود. به همین دلیل، ما فرض می کنیم مدل (P_2) با این محدودیت که مکان های X باید از p رأس متفاوت انتخاب شوند، حل شود. برای مدل (P_2) مکان های بهینه لزوما در برگ ها اتفاق نمی افتد. مثال۱. ۱ (شکل۱. ۱) نشان می دهد که یک جواب بهینه لزوما در برگ های درخت قرار نگرفته است.

مثال ۱. ۱. [۱۸]. درخت در شکل (۱. ۱) را در نظر بگیرید. می خواهیم مکان بهینه برای مسأله ۲-میانه را روی آن بررسی کنیم.



[28] Kang

[29] Cheng

شکل (۱.۱) جواب بهینه مدل P_2 هیچ برگی از درخت را شامل نمی شود.

$$F(v_1,v_2)=\sum_{i=1}^{6} w_i \min\big(d(v_i,v_1),d(v_i,v_2)\big)$$

$$= w_1 \min\big(d(v_1,v_1),d(v_1,v_2)\big)+ w_2 \min\big(d(v_2,v_1),d(v_2,v_2)\big)$$

$$+ w_3 \min\big(d(v_3,v_1),d(v_3,v_2)\big)+ w_4 \min\big(d(v_4,v_1),d(v_4,v_2)\big)$$

$$+ w_5 \min\big(d(v_5,v_1)),d(v_1,v_2)\big)+ w_6 \min\big(d(v_6,v_1),d(v_6,v_2)\big)$$

$$=(-50\times 0)+(-1\times 0)+(-150\times 1)+(-1\times 2)+(-20\times 22)+(-100\times 52)=-5792$$

و به همین ترتیب برای بقیه رئوس محاسبه می کنیم.

جدول (۱.۱) مقادیر F

۲- میانه	F
(v_1 , v_2)	-5792
(v_1 , v_3)	-5523
(v_1 , v_4)	-5552
(v_1 , v_5)	-10192

(v_1 , v_6)	-9152
(v_2 , v_3)	-10521
(v_2 , v_4)	-10550
(v_2 , v_5)	-10352
(v_2 , v_6)	-5592
(v_3 , v_4)	-10451
(v_3 , v_5)	-10152
(v_3 , v_6)	-5472
(v_4 , v_5)	-10252
(v_4 , v_6)	-5652
(v_5 , v_6)	-9292

با توجه به مقادیر به دست آمده در جدول (۱. ۱) ، $F(v_2 , v_4)$ کمترین مقدار به دست آمده را دارا می باشد. در نتیجه $\{v_2 , v_4\}$ با مقدار $F(v_2 , v_4)=-10550$ جواب بهینه مسأله ۲-میانه می باشد، همان طور که مشاهده می کنید ، v_2 , v_4 جزء برگ های درخت نمی باشند.

فصل دوم

۱- الگوریتم خطی برای مسأله ماکسین روی درخت

۲.۱ مقدمه

در این فصل مسأله تعیین مکان بهینه یک سرویس دهنده را روی شبکه درخت n-رأسی $T=(V,E,\lambda,w)$ بررسی می کنیم، به طوری که تابع زیر ماکسیمم شود[۱۲].

$$f(x)=\sum\{w(v)d(x,v),v\in V\} \qquad (۲.۱)$$

برای هر $e\in E$، $\lambda(e)>0$ به معنای طول یال می باشد، در حالی که برای هر $v\in V$، $w(v)\geq 0$ به معنای وزن رئوس است. (فرض میکنیم که همه $w(v)=0$ نباشد). $d(x,y)$ به معنای اندازه کوتاهترین مسیر بین هر دو نقطه $x,y\in T$ می باشد.

با توجه به [۱۴] یک راه حل طبیعی برای مسأله بیشترین مجموع یا ۱-ماکسین روی یک درخت، ارزیابی تابع f در هر برگ و انتخاب بیشترین مقدار به دست آمده برای تابع f می باشد.

۲.۲ روش حل مسأله

در این بخش تعاریف ضروری، نمادگذاری ها و ۴ لم که برای توجیه الگوریتم مورد استفاده قرار میگیرد، ارائه می شود. منظور از شبکه در این بخش، شبکه درخت $T=(V,E,\lambda,w)$ میباشد. درجه هر رأس v در T یا به عبارت دیگر تعداد یالهای واقع شده در v به وسیله $d(v)$ نمایش داده میشود. بنابراین مجموعه برگ ها یا رئوس آویزان از T می تواند به صورت زیر تعریف شود،

$$L=\{v\in V:d(v)=1\}.$$

در اینجا تعریف می کنیم،

$$f_{max}=max\{f(x):x\in T\}$$

بنابر نتایج بدست آمده از چرچ و گارفینکل [۱۴] (مبنی بر این که جواب بهینه مسأله مکان یابی تسهیلات مضر در درخت، در برگ ها رخ می دهد)، تعریف بالا می تواند مجدداً به صورت لم زیر بیان شود،

لم ۲. ۱. [۱۲] : رأس $v \in L$ وجود دارد که *f(x)* را ماکزیمم می کند.

برای بدست آوردن نتایج دقیق تر از لم ۲. ۱، برای هر $v \in V$ تعریف میکنیم،

$$\Delta(v) = max\{ f(u) - f(v) : u \in L - \{v\}\} \qquad (۲. ۲)$$

فرض کنید که *u(v)* نشان دهنده برگ *u* باشدکه در فرمول (۲. ۲) به ماکسیمم دست می یابد. (توجه کنید *L-{v}* تهی نیست بنابر فرضی که $|V|>1$).

لم ۲. ۲. [۱۲] :

a) اگر $\Delta(v) < 0$، آنگاه و $v \in L$ مقدار f را ماکزیمم می کند به عبارت دیگر $f(v) = f_{max}$

b) اگر $\Delta(v) \geq 0$، آنگاه *u(v)* مقدار f را ماکزیمم می کند به عبارت دیگر

$$f(u(v)) = f(v) + \Delta(v) = f_{max}$$

اثبات

a) با توجه به تعریف f_{max} و لم ۲. ۱ داریم،

$$f_{max} = max\{ f(u) : u \in L\}$$

فرض کنید $\Delta(v) < 0$. اگر $v \in V - L$ ، آنگاه داریم،

$$f(v) > max\{ f(u) : u \in L\}$$

یعنی $f(v) > f_{max}$ که این تناقض است. حال اگر $v \in L$، با توجه به اینکه $\Delta(v) < 0$ داریم،

$$f(v) > max\{ f(u) : u \in L - \{v\}\}$$

و با توجه به تعریف f_{max} داریم،

$f(v)=f_{max}$

و نتیجه به دست می آید.

b) فرض کنید $\Delta v \geq 0$ برای هر $u \in L-\{v\}$، داریم،

$$f(u(v))= f(v)+[f(u(v))- f(v)]\geq f(v)+[f(u)- f(v)]= f(u)$$

(رابطه بالا با توجه به تعریف $u(v)$ به دست می آید).

اگر $v \in V-L$ در این صورت $f(u(v))=max\{ f(u):u \in L\}$ یعنی $f(u(v))= f_{max}$ که نتیجه به دست می آید. حال اگر $v \in L$ با توجه به اینکه $\Delta v \geq 0$ داریم، $f(u(v))\geq f(v)$ که این با رابطه $f(u(v))=max\{ f(u):u \in L-\{v\}\}$ نتیجه می دهد که $f(u(v))= f_{max}$. □

در حقیقت ایده ی اساسی الگوریتم، ریشه دار کردن درخت T، یعنی؛ یک رأس از T را به طور اختیاری به عنوان ریشه ی درخت در نظر می گیریم و درخت را با توجه به رأس در نظر گرفته شده به عنوان ریشه r، ریشه دار می کنیم؛ تعیین کردن $\Delta(r)$ و $u(r)$ با زمان $o(n)$ با به کار گرفتن یک ساختار بازگشت پذیر مناسب و سپس استفاده از لم ۲. ۲ برای شناختن r و یا $u(r)$ به عنوان مکان بهینه خواهد بود. برای تدارک الگوریتم، هر رأسی از v را به دلخواه در نظر می گیریم و فرض می کنیم همه یالهایی که شامل v هستند به جز یکی از آنها از درخت T حذف شوند، زیرگراف باقیمانده ناهمبند خواهد شد اگر $d(v)>1$ ، یکی از مؤلفه ها از زیرگراف های ایجاد شده که شامل v می باشد، یک شاخه اولیه در v نامیده خواهد شد. به تعداد درجه رأس v یعنی $d(v)$، ما از این زیر درخت ها خواهیم داشت که اجتماع آنها T را به ما می دهد، و هر دو از آنها در $\{v\}$ متقاطعند. در اینجا خود $\{v\}$ را نیز به عنوان یک شاخه اولیه در v در نظر می گیریم. حال یک شاخه درخت در v را، یک زیر درخت از T که اجتماع یک یا چندین شاخه اولیه است، تعریف می کنیم. یک شاخه درخت B را در نظر بگیرید، مجموعه رئوس آن را به وسیله $V(B)=B\bigcap V$ نشان می دهیم.

برای هر v و هر شاخه درخت B در v، $\Delta(v)$ و $u(v)$ از لم ۲. ۲ را به وسیله تعریف زیر تعمیم می دهیم.

$$\Delta(v,B)=max\{ f(u)-f(v):\ u\in L\bigcap V(B)-\{v\}\} \qquad (۲. ۳)$$

فرض کنید $u(v,B)$ هر برگی باشد که در (۲. ۳) به ماکسیمم دست می یابد. توجه کنید که $L\bigcap V(B)-\{v\}$ تهی خواهد بود اگر و تنها اگر $B=\{v\}$، در این صورت $\Delta(v,B)=(-\infty)$ و ما قرار می دهیم $u(v,B)=v$.

در آخر ما تابع f را به زیرمجموعه V' از V اختصاص می دهیم، به عبارت دیگر

$$f(x,v')=\sum\{ w(v)d(x,v): v\in V'\},$$

و همچنین تابع w را به طور افزایشی به زیرمجموعه ها تعمیم می دهیم،

$$w(V')=\sum\{ w(v): v\in V'\}.$$

قرار دهید $W=w(V)$. با این نمادسازی ها، ما می توانیم در دو لم بعدی، مطالبی را عنوان کنیم که برای الگوریتم بازگشتی مورد نیاز است.

لم ۲. ۳. [۱۲] **(ادغام)** : برای هر رأس v، فرض کنید B',B'' دو شاخه درخت در v باشند، که مجزا باشند به جز در رأس v، و B یک شاخه درخت در v به گونه ای باشد که $B=B'\bigcup B''$. (شکل (۲. ۱) را ببینید.)

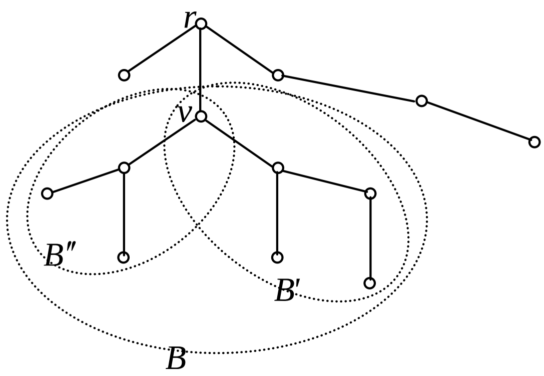
اگر $\Delta(v,B')\geq\Delta(v,B'')$ آنگاه $\Delta(v,B)$ و $u(v,B)$، $w(V(B))$ و $f(v,V(B))$ به وسیله مقادیر مربوط به B',B'' به دست می آید. یعنی،

$$a)\ \Delta(v,B)=\Delta(v,B')\ \ ,\ \ u(v,B)=u(v,B'),$$

$b)\, w(V(B)) = w(V(B')) + w(V(B'')) - w(v),$

$f(v,V(B)) = f(v,V(B')) + f(v,V(B'')).$

اثبات



شکل(۲. ۱) درخت با دو شاخه درخت B' و B'' که مجزا هستند به جز در رأس v

وشاخه درخت B به طوریکه $B = B' \cup B''$.

a) با توجه به تعریف $\Delta(v,B)$ داریم،

$$\Delta(v,B') = max\{\, f(u) - f(v):\ u \in L \cap V(B') - \{v\}\}$$

$$\Delta(v,B'') = max\{\, f(u) - f(v):\ u \in L \cap V(B'') - \{v\}\}$$

درصورت مسأله فرض کردیم که $\Delta(v,B') \geq \Delta(v,B'')$ و این بدین معنی است که فاصله برگ ها در شاخه درخت B' تا رأس v از فاصله برگ ها در شاخه درخت B'' تا رأس v بیشتر است و چون در $\Delta(v,B)$ ماکسیمم این فاصله را می خواهیم، داریم،

$$\Delta(v,B) = \Delta(v,B')$$

$$u(v,B) = u(v,B')$$

b) با توجه به تعریف $f(v,V(B)),\ w(V(B))$ داریم،

$$w(V(B))=\sum\{w(v):v\in V(B)\}$$

$$f(v,V(B))=\sum\{w(u)d(v,u)\ :\ u\in V(B)\}$$

با توجه به اینکه $B=B'\cup B''$ و چون در رأس v در شاخه درخت B' و B'' مشترک هستند، داریم،

$$w(V(B))=\sum\{w(v):v\in V(B)\}$$

$$=\sum\{w(v):v\in V(B')\}+\sum\{w(v):v\in V(B'')\}-w(v)$$

$$=w(V(B'))+w(V(B''))-w(v)$$

$$f(v,V(B))=\sum\{w(u)d(v,u)\ :\ u\in V(B)\}$$

$$=\sum\{w(u)d(v,u)\ :\ u\in V(B')\}+\sum\{w(u)d(v,u)\ :\ u\in V(B'')\}$$

$$=f(v,V(B'))+f(v,V(B'')).\qquad\square$$

لم ۲. ۴. [۱۲] : (بسط). فرض کنید B یک شاخه درخت در q ، $e=(q,p)$ یک یال که در B نیست، و $B^*=B\cup e$ یک شاخه درخت در p باشد. آنگاه مقادیر $f(p,V(B^*)),\ w(V(B^*)),u(p,B^*),\ \Delta(p,B^*)$ به وسیله مقادیر مربوط به (q, B) به دست می آید. یعنی،

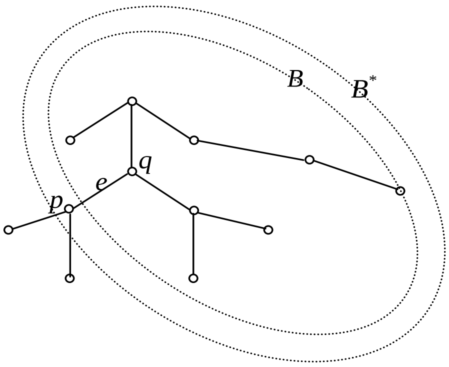
$$a)\ \ \Delta(p,B^*)=[W-2w(V(B))]\lambda(e)+\begin{cases}\Delta(q,B) & if\quad q\in V-L\\ 0 & if\quad q\in L\end{cases}$$

$b)\ u(p, B^*) = u(q, B),$

$c)\ w(V(B^*)) = w(V(B)) + w(p),$

$d)\ f(p, V(B^*)) = f(q, V(B)) + w(V(B))\lambda(e).$

برای درک بیشتر شکل (۲. ۲) را در نظر بگیرید.



شکل (۲. ۲) درخت با دو شاخه درخت B و B^* به طوریکه $B^* = B \cup e$.

توجه داشته باشید که شاخه درخت B در q باید به گونه ای انتخاب شود که $B^* = B \cup e$ یک شاخه درخت در p باشد.

اثبات

برای اثبات (a) و (b)، ابتدا فرض کنید $q \in V - L$. آنگاه با توجه به اینکه $B^* = B \cup e$ رابطه زیر برقرار است،

$$L \cap V(B) - \{q\} = L \cap V(B^*) - \{p\}$$

از آنجایی که $u(q, B)$ یک برگ u انتخاب شده از مجموعه بالا است که به بیشترین مقدار $f(u)$ دست یافته است، رابطه (b) صحیح است.

از طرفی با توجه به تعریف $\Delta(p,B^*)$ داریم،

$$\Delta(p,B^*)=max\{f(u)-f(p):\ u\in L\cap V(B^*)-\{p\}\}$$

$$=max\{f(u)-f(q)+f(q)-f(p):\ u\in L\cap V(B)-\{q\}\}$$

$$=max\{f(u)-f(q):\ u\in L\cap V(B)-\{q\}\}-[f(p)-f(q)]$$

$$=\Delta(q,B)-[f(p)-f(q)].$$

$$\Delta(p,B^*)=\Delta(q,B)-[f(p)-f(q)] \qquad (۲. ۴)$$

تفاضل سمت راست رابطه یعنی $(f(p)-f(q))$ می تواند به صورت زیر ارزیابی شود،

$$w(V(B))\lambda(e)-[W-w(V(B))]\lambda(e)$$

دلیل برابری دو رابطه به این دلیل است که رئوسی که در $V(B)$ هستند با مجموع وزنی $w(V(B))$، به رأس q به اندازه $\lambda(e)$ تا رأس p نزدیکترند و بقیه رئوس، با مجموع وزنی $(W-w(V(B)))$، به رأس p نسبت به رأس q به اندازه $\lambda(e)$ نزدیکترند. در واقع رئوسی که در $V(B)$ هستند، $d(v,p)-d(v,q)$ برای آنها برابر $\lambda(e)$ و برای بقیه رئوس این اختلاف برابر $-\lambda(e)$ می باشد یعنی،

$$f(p)-f(q)=w(V(B))\lambda(e)-[W-w(V(B))]\lambda(e)=[2w(V(B))-W]\lambda(e)$$

با جایگذاری در رابطه (۲. ۴) داریم،

$$\Delta(p,B^*)=\Delta(q,B)-[f(p)-f(q)]=[W-2w(V(B))]\lambda(e)+\Delta(q,B)$$

و در نتیجه رابطه (a) به دست می آید.

حال فرض کنید $q \in L$، آنگاه e تنها یالی است که در q واقع شده است، در این صورت $B=\{q\}$ و $B^*=e$. از آنجایی که $L \cap V(B^*)-\{p\}=\{q\}$ در نتیجه $u(p,B^*)=q$ و از آنجاییکه $u(q,B)=q$ و در نتیجه $u(p,B^*)=u(q,B)$. از طرفی،

$$\Delta(p,B^*)=max\{f(u)-f(p):\ u \in L \cap V(B^*)-\{p\}\}$$

$$=\{f(q)-f(p)\}$$

$$=[W-2w(V(B))]\lambda(e)$$

که رابطه (a) نیز نتیجه می شود.

c) برای اثبات قسمت c، از آنجایی که $B^*=B \cup e$ لذا مجموعه رئوس B^* از مجموعه رئوس B و رأس p تشکیل شده است، در نتیجه رابطه c بدیهی است.

d) با توجه به تعاریف و با توجه به اینکه فاصله q تا p برابر $\lambda(e)$ می باشد، داریم،

$$f(p,V(B^*))=\sum\{w(v)d(v,p)\ :\ v \in V(B^*)\}$$

$$=\sum\{w(v)[d(v,q)+\lambda(e)]\ :\ v \in V(B)\}$$

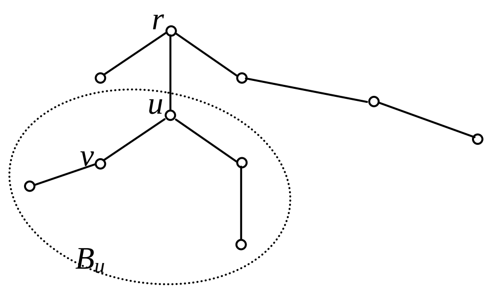
$$=\sum\{w(v)d(v,q)\ :\ v \in V(B)\}+\lambda(e)\sum\{w(v):\ v \in V(B)\}$$

$$=f(q,V(B))+\lambda(e)w(V(B)).\quad \square$$

۲.۳ الگوریتم

همان طور که قبلا ملاحظه کردید، در الگوریتم فرض می کنیم که رأس r از T به عنوان ریشه انتخاب شده است، این انتخاب به دلخواه است. حال با به کار گرفتن اصطلاحات علمی داریم،

اگررأس u بلافاصله جلوتر از رأس v روی مسیر از r به v باشد،آنگاه u پدر v است که با $\pi(v)$ نشان می دهیم و v یکی از فرزندان u است. یک نسل از u، رئوسی مانند v هستند چنان که u روی مسیر از r به v باشد. ما V_u را مجموعه رئوس شامل u و همه نسل هایش قرار می دهیم و B_u را زیرگرافی از T که نتیجه شده به وسیله V_u در نظر میگیریم. توجه کنید که B_u یک شاخه درخت در u است. برای مثال $B_r=T$.



شکل(۲. ۳) زیرگراف B_u شامل رأس u و همه نسل هایش.

الگوریتم یک پیمایش از برگ به ریشه را که از درخت ریشه دار T ساخته شده است، دربردارد. اگر $c_1,...,c_{d(r)}$ را به عنوان فرزندان r در نظر بگیریم، آنگاه پیمایش به وسیله برنامه بازگشتی می تواند به صورت زیر توصیف شود :

۱- پیمایش رئوس زیردرخت های ریشه دار شده در $c_1,...,c_{d(r)}$

۲- پیمایش خود r

ما فرض میکنیم که رئوس بر طبق موقعیتشان در این پیمایش شماره گذاری شوند، بنابراین رأس۱ یک برگ، رأس n ریشه r است و ... . با این شماره گذاری داریم، (منظور از u، شماره رأس می باشد.)

$v \in V_u$ اگر و تنها اگر $u-|V_u|<v\le u$.

قبل از ارائه الگوریتم به طور رسمی، توضیحاتی را ارائه می دهیم. محاسبات شامل n-1 مرحله است، هر کدام متناظر با عبور از گام های ۱ تا ۶ است که در زیر توضیح داده می شود. در q امین مرحله، رأس q اطلاعات خودش را به پدر خود $\pi(q)$ گزارش می دهد. این اطلاعات شامل $w(V_q)$, $\Delta(q,B_q)$, $u(q,B_q)$ و $f(q,V_q)$ است. برای بدست آوردن این اطلاعات در $\pi(q)$ از فرمول های لم ۲. ۴ استفاده می شود، با این فرض که $p=\pi(q)$. همان طور که هر فرزند از رأس v اطلاعات خود را به v گزارش می دهد، اطلاعات جدید باید به طور مناسب با آن اطلاعاتی که قبلاً گزارش شده از فرزندان v ادغام شود، که این فرآیند با استفاده از فرمول های لم ۲. ۳ انجام می شود و در آن B' با یالهایی از v که فرزندان قبلاً اطلاعات خود را گزارش داده اند متناظر است و B'' با یالی از v که جدیداً فرزند آن گزارش داده، متناظر است.

مهم است که به یاد داشته باشیم در پیمایش برگ به ریشه، از رأس q خواسته نمی شود که به پدرش قبل از اینکه اطلاعات خودش کامل شود، گزارش دهد. یعنی قبل از اینکه همه فرزندان q به q گزارش دهند.

در آخرین مرحله، ما تعیین خواهیم کرد $u(n,B_n)=u(r), \Delta(n,B_n)=\Delta(r,T)$ و $w(V_r)=W$ و $f(r,V_r)=f(r)$. اگر $\Delta(r)<0$، آنگاه به وسیله لم ۲. ۲، r خودش یک مکان بهینه است، و مقدار بهینه $f_{max}=f(r)$ در دست است.

اگر $\Delta(r)\geq 0$، آنگاه به وسیله لم ۲. ۲ رأس شناخته شده $u(r)$ یک مکان بهینه است و $f_{max}=f(r)+\Delta(r)$ است.

ما اکنون الگوریتم را به طور رسمی ارائه می دهیم. الگوریتم شامل ۴ آرایه به طول n به صورت (Δ,u,w',f') است. در q امین مرحله ازالگوریتم مقادیر $\Delta(q,B'_q)$ $u(q,B'_q)$, $w(v(B'))$, $f(q,v(B'_q))$ به دست خواهد آمد، که B'_q یک شاخه درخت

در q است که در طی الگوریتم تغییر میکند. هر B'_q در $\{q\}$ مقدار دهی اولیه می شود؛ برای هر فرزند k از q، درkامین مرحله از الگوریتم، B'_q به وسیله اتصال با شاخه ابتدایی جدید در q که با یال (k,q) متناظر است گسترش می یابد. زمانی که qامین مرحله رسیده است، B'_qبه مقدار نهایی خودش B_q گسترش یافته است، و بنابراین در qامین ورود از ۴ آرایه اطلاعات مناسبی به دست می آید تا به $\pi(q)$ گزارش شود.

الگوریتم

گام صفر : مقدار دهی اولیه با قرار دادن $\Delta(q)=(-\infty)$، $w'(q)=w(q),\ u(q)=q$، $f'(q)=0$ برای $q=1, 2, ..., n$. سپس قرار دهید $q=0$.

گام اول : قرار دهید $q:=q+1$ و $\lambda=\lambda(q,\pi(q))$.

گام دوم : قرار دهید $\Delta''=[W-2w'(q)]\lambda$. اگر $q\in V-L$، بگذارید $\Delta'':=\Delta''+\Delta(q)$.

گام سوم: بگذارید $u''=u(q)$ $w''=w'(q)+w(\pi(q)),$ و $f''=f'(q)+w'(q)\lambda$.

گام چهارم : اگر $\Delta''>\Delta(\pi(q))$ بگذارید $\Delta(\pi(q)):=\Delta''$، $u(\pi(q)):=u''$.

گام پنجم : بگذارید $w'(\pi(q)):=w'(\pi(q))+w''-w(\pi(q))$

$f'(\pi(q)):=f'(\pi(q))+f''$.

گام ششم : اگر $q=n-1$ متوقف شوید، در غیر اینصورت به گام اول بروید.

قضیه۲. ۱. [۱۲] : الگوریتم بالا $\Delta(r)$ و $u(r)$ را با زمان $o(n)$ به درستی محاسبه می کند.

اثبات

از آنجایی که الگوریتم *n-1* بار از میان گام های ۱ تا ۶ به ترتیب عبور می کند و هر بار نیازمند زمان *o(1)* می باشد، در نتیجه پیچیدگی زمانی الگوریتم *o(n)* می باشد.

صحت خروجی الگوریتم با استفاده از توضیحات داده شده در بخش سوم و لم های ۲. ۳ و ۲. ۴ قابل اثبات می باشد. گام های ۲ و ۳ متناظر با لم ۲. ۴ می باشند، فرض می کنیم $p=\pi(q)$، در واقع اطلاعات زیر درخت با توجه به یال اضافه شده، بسط داده می شود، که این کار به وسیله لم ۲. ۴ صورت می گیرد. یعنی،

$$\varDelta'' = \varDelta(p) = [W - 2w(V(B))]\lambda(q,p) + \begin{cases} \varDelta(q) & \text{if} \quad q \in V - L \\ 0 & \text{if} \quad q \in L \end{cases}$$

$\lambda(q,p)$ در گام اول در هر مرحله به دست آمده و برابر λ می باشد و $w(q)$ در گام صفر برابر $w'(q)$ فرض شده است،

$$\varDelta'' = \varDelta(p) = [W - 2w'(q)]\lambda + \begin{cases} \varDelta(q) & \text{if} \quad q \in V - L \\ 0 & \text{if} \quad q \in L \end{cases}$$

$$u'' = u(p) = u(q)$$

$$w'' = w(q) + w(p) = w'(q) + w(\pi(q))$$

$$f'' = f(q) + w(q)\lambda = f'(q) + w'(q)\lambda$$

گام های ۴ و ۵ متناظر با لم ۲. ۳ می باشند که در این گام ها اطلاعاتی که قبلا به دست آمده با اطلاعات جدید ادغام و به روز رسانی می شود که این کار به وسیله لم ۲. ۳ صورت می گیرد. □

در ارائه رسمی الگوریتم، گام های ۲ تا ۵ بر اساس لم های ۲. ۳ و ۲. ۴ فرمول بندی شده اند. برای اجرا در سیستم، بهره وری بیشتر به وسیله حذف محاسبات f'', w'' در گام۳ به دست می آید. (بنابراین، این مقادیر کمکی زیاد لازم نیستند)، به جای فرمول های گام ۵ قرار می دهیم،

$$w'(\pi(q)) := w'(\pi(q)) + w'(q)$$

$$f'(\pi(q)) := f'(\pi(q)) + f'(q) + w'(q)\lambda$$

در واقع با جایگذاری فرمول های مربوط به گام ۳ درفرمول های گام ۵ روابط بالا به دست آمده است،

$$w'(\pi(q)) = w'(\pi(q)) + w'' - w(\pi(q))$$

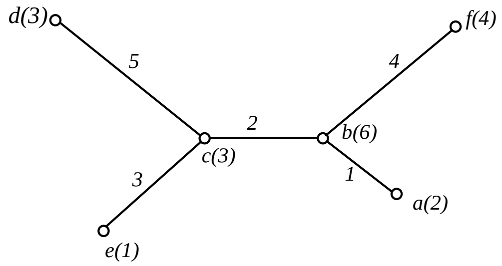
$$= w'(\pi(q)) + w'(q) + w(\pi(q)) - w(\pi(q))$$

$$= w'(\pi(q)) + w'(q)$$

$$f'(\pi(q)) = f'(\pi(q)) + f'' = f'(\pi(q)) + f'(q) + w(q)\lambda\ .$$

۲. ۴ مثال عددی

در اینجا الگوریتم را با مثالی روشن می کنیم. درخت T با ۶ رأس را در شکل (۲. ۴) در نظر بگیرید. وزن هر رأس v یعنی $w(v)$ کنار هر رأس داخل پرانتز مشخص شده، و طول هر یال کنار یال متناظر داده شده است. با توجه به اینکه فاصله هر رأس تا رأس دیگر داده شده، می توانیم مقادیر تابع هدف برای هر کدام از رئوس را محاسبه کنیم.



شکل (۲. ۴) یک درخت ساده با وزن ها و فاصله های بین رئوس

$d(a,b) = 1$	$d(b,c) = 2$	$d(c,e) = 3$
$d(a,c) = 3$	$d(b,d) = 7$	$d(c,f) = 6$
$d(a,d) = 8$	$d(b,e) = 5$	$d(d,e) = 8$
$d(a,e) = 6$	$d(b,f) = 4$	$d(d,f) = 11$
$d(a,f) = 5$	$d(c,d) = 5$	$d(e,f) = 9$

$$f(a)=\sum_{i=1}^{6} w_i d(a,v_i)=w(b)d(a,b)+w(c)d(a,c)+$$

$$+w(e)d(a,e)+w(d)d(a,d)+w(f)d(a,f)=$$

$$=(6\times 1)+(3\times 3)+(1\times 6)+(3\times 8)+(4\times 5)=65$$

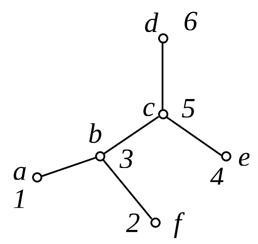
و به همین ترتیب برای بقیه رئوس به صورت زیر محاسبه می کنیم.

$$f(b)=50,\ f(c)=60,\ f(d)=125,\ f(e)=111,\ f(d)=94$$

ازآنجاییکه بر طبق لم ۲. ۱، ماکسیمم مقدار برای f در یک برگ اتفاق می افتد، با توجه به مقادیر به دست آمده رأس d مکان بهینه برای سرویس دهنده مضر در این مسأله است. (با توجه به لم ۲. ۱، جواب بهینه به مجموعه برگ های $\{a,d,e,f\}$ محدود می شود.)

حال می خواهیم این مسأله را با توجه به الگوریتم حل کنیم. در اینجا یک بار مسأله را با فرض $r=d$ و بار دیگر با فرض $r=b$ حل می کنیم.

فرض می کنیم $r=d$ داریم،



شکل (۲. ۵) درخت دریشه دار شده با فرض $r=d$

ابتدا رئوس را با توجه به پیمایش برگ به ریشه شماره گذاری می کنیم، q شماره هر رأس v است. حال با توجه به گام های الگوریتم داریم،

گام صفر :

$\Delta(a)=-\infty$	$u(a)=a$	$w'(a)=w(a)$	$f'(a)=0$	$q=1$
$\Delta(f)=-\infty$	$u(f)=f$	$w'(f)=w(f)$	$f'(f)=0$	$q=2$
$\Delta(b)=-\infty$	$u(b)=b$	$w'(b)=w(b)$	$f'(b)=0$	$q=3$
$\Delta(e)=-\infty$	$u(e)=e$	$w'(e)=w(e)$	$f'(e)=0$	$q=4$
$\Delta(c)=-\infty$	$u(c)=c$	$w'(c)=w(c)$	$f'(c)=0$	$q=5$
$\Delta(d)=-\infty$	$u(d)=d$	$w'(d)=w(d)$	$f'(d)=0$	$q=6$

گام اول : $\lambda=\lambda(a,\pi(a))=\lambda(a,b)=1,\ q:=1$

گام دوم : $\Delta''=[W-2w'(a)]\lambda=[19-(2\times 2)]\times 1=15$

گام سوم : $u''=u(a)=a$

گام چهارم : $u(b):=a,\ \Delta(b):=15,\ 15>\Delta(b)=-\infty$

گام پنجم : $w'(b) := w'(b) + w'(a) = 6 + 2 = 8$

$$f'(b) := f'(b) + f'(a) + w'(a) \times 1 = 0 + 0 + 2 = 2$$

گام ششم : به گام اول بر می گردیم.

گام اول : $\lambda = \lambda(f, \pi(f)) = \lambda(f, b) = 1,\ q := 2$

گام دوم : $\Delta'' = [W - 2w'(f)] \times 4 = [19 - (2 \times 4)] \times 4 = 44$

گام سوم : $u'' = u(f) = f$

گام چهارم : $u(b) := f,\ \Delta(b) := 44,\ 44 > \Delta(b) = 15$

گام پنجم : $w'(b) := w'(b) + w'(f) = 8 + 4 = 12$

$$f'(b) := f'(b) + f'(f) + w'(f) \times 4 = 2 + 0 + 4 \times 4 = 18$$

گام ششم : به گام اول بر می گردیم.

گام اول : $\lambda = \lambda(b, \pi(b)) = \lambda(b, c) = 2,\ q := 3$

گام دوم : $\Delta'' = [W - 2w'(b)] \times 2 = [19 - (2 \times 12)] \times 2 = -10$، حال چون $q \in V - L$ است

داریم : $\Delta'' = \Delta'' + \Delta(b) = -10 + 44 = 34$

گام سوم : $u'' = u(b) = f$

گام چهارم : $u(c) := f,\ \Delta(c) := 34,\ 34 > \Delta(c) = -\infty$

گام پنجم : $w'(c) := w'(c) + w'(b) = 3 + 12 = 15$

$$f'(c) := f'(c) + f'(b) + w'(b) \times 2 = 0 + 18 + 12 \times 2 = 42$$

گام ششم : به گام اول بر می گردیم.

گام اول : $\lambda = \lambda(e,\pi(e)) = \lambda(e,c) = 3,\ q := 4$

گام دوم : $\Delta'' = [W - 2w'(e)] \times 3 = [19 - (2 \times 1)] \times 3 = 51$

گام سوم : $u'' = u(e) = e$

گام چهارم : $u(c) := e,\ \Delta(c) := 51,\ 51 > \Delta(c) = 34$

گام پنجم : $w'(c) := w'(c) + w'(e) = 15 + 1 = 16$

$$f'(c) := f'(c) + f'(e) + w'(e) \times \lambda = 42 + 0 + 1 \times 3 = 45$$

گام ششم : به گام اول بر می گردیم.

گام اول : $\lambda = \lambda(c,d) = 5,\ q := 5$

گام دوم : $\Delta'' = [W - 2w'(c)] \times 5 = [19 - (2 \times 16)] \times 5 = -65$ حال چون $q \in V - L$ است

$$\Delta'' = \Delta'' + \Delta(c) = -65 + 51 = -14$$

گام سوم : $u'' = u(c) = e$

گام چهارم : $u(d) := e,\ \Delta(d) := -14,\ \ -14 > \Delta(d) = -\infty$

گام پنجم : $w'(d) := w'(d) + w'(c) = 3 + 16 = 19$

$$f'(d) := f'(d) + f'(c) + w'(c) \times \lambda = 0 + 45 + 16 \times 5 = 125$$

گام ششم : $q=5$ متوقف می شویم .

حال با توجه به محاسبات انجام گرفته و مقادیر به دست آمده $\Delta(d)=-14<0$ و $f(d)=125$ با استفاده از لم ۲. ۲، رأس d جواب بهینه و $f_{max}=125$ است.

مقادیر به دست آمده از الگوریتم در جدول ۲. ۱ آورده شده است. ستون π به پدر هر رأس مربوط می شود. بخش "بسط" از خروجی گام ۲ در مرحله q به دست آمده و بخش "ادغام" و "به روز رسانی" از خروجی گام ۴ و گام ۵ به دست می آید.

جدول (۲. ۱) مقادیر بدست آمده از الگوریتم با فرض **r=d**

r	q	یال جدید		بسط			ادغام		به روز رسانی		درخت جدید
		v	π	$\Delta(v)$	$u(v)$	Δ''	$\Delta(\pi)$	$u(\pi)$	$w'(\pi)$	$f'(\pi)$	v_π
d	1	a	b	$-\infty$	a	15	15	a	8	2	b,a
	2	f	b	$-\infty$	f	44	44	f	12	18	b,a,f
	3	b	c	44	f	34	34	f	15	42	b,a,f,c
	4	e	c	$-\infty$	e	51	51	e	16	45	b,a,f,c,e
	5	c	d	51	e	-14	-14	e	19	125	b,a,f,c,e,d

حال فرض می کنیم $r=b$ باشد، با انجام مرحله به مرحله الگوریتم مقادیر زیر به دست می آید که در جدول ۲. ۲ آورده شده است.

جدول (۲.۲) مقادیر بدست آمده از الگوریتم با فرض **r=b**

r	q	یال جدید		بسط			ادغام		به روز رسانی		درخت جدید
		v	π	$\Delta(v)$	$u(v)$	Δ''	$\Delta(\pi)$	$u(\pi)$	$w'(\pi)$	$f'(\pi)$	v_π
b	1	a	b	$-\infty$	a	15	15	a	8	2	a,b
	2	d	c	$-\infty$	d	65	65	d	6	15	d,c
	3	e	c	$-\infty$	e	51	65	d	7	18	d,c,e
	4	c	b	65	d	75	75	d	15	34	a,b,d,c,e
	5	f	b	$-\infty$	f	44	75	d	19	50	a,b,d,c,e,f

با توجه به مقادیر به دست آمده $\Delta(b)=75$، $f(b)=50$ و $u(b)=d$ ، با توجه به لم ۲. ۲، رأس d جواب بهینه و $f_{max} = f(b)+\Delta(b)=125$.

فصل سوم

مسأله p–ماکسین روی یک درخت

۳. ۱ مقدمه

در این فصل ما ابتدا مسأله ۲-ماکسین روی یک درخت را مطرح می کنیم که با مدل P_1 از مسأله ۲-میانه مضر معادل است. سپس نشان میدهیم که یک جواب بهینه این مسأله به وسیله دو برگ از طولانی ترین مسیر در درخت به دست می آید. بنابراین این مسأله در زمان خطی می تواند حل شود و جواب مستقل از وزنهای واقعی است.

نتیجه به آسانی می تواند به مسأله p-ماکسین تعمیم داده شود، در حالیکه مسأله کلاسیک p-میانه روی یک درخت در زمان $O(pn^2)$ [۷] می تواند حل شود، مسأله p-ماکسین در زمان خطی حل شده است و جواب برای تمام وزنهای ممکن رئوس یکسان است[۱۸] .

۳. ۲ مفروضات مسأله

$T=(V,E)$ یک درخت با n رأس را در نظربگیرید. ما فرض می کنیم که هر رأس v_i دارای یک وزن مثبت w_i باشد. در این بخش مسأله p-ماکسین را مورد بررسی قرار می دهیم که با مدل P_1 از مسأله p-میانه با وزن منفی معادل است.

همانطور که در فصل اول اشاره شد، هدف پیدا کردن یک زیرمجموعه X از V با $|X|=p$ می باشد بطوریکه تابع زیر ماکسیمم شود. در اینجا فرض بر این است که نقاط بهینه روی رأس قرار دارند.

$$F(X)=\sum_{i=1}^{n}\max_{1\le j\le p}(w_i d(x_j,v_i))$$

$$=\sum_{i=1}^{n}w_i\max_{1\le j\le p}d(x_j,v_i)$$

با در نظر گرفتن مسئله ۲-ماکسین شروع می کنیم. هدف یافتن دو رأس u, v (در واقع می خواهیم مکان بهینه برای دو سرویس دهنده را روی رئوس درخت بیابیم) به طوری که

$$F(u,v)=\sum_{i=1}^{n} w_i \ \ max(d(v_i,u), d(v_i,v)) \qquad (۳. ۱)$$

ماکسیمم شود.

برای هر دو رأس u , v از درخت T , $P(u,v)$ را مسیر یکتا در درخت برای اتصال رأس u به رأس v در نظر می گیریم. نقطه میانی روی مسیر را با m_{uv} نمایش می دهیم. این نقطه میانی روی یک یال می افتد، مثلا فرض کنید نقطه میانی دو رأس u ,v روی یال $[r,s]$ باشد، که ما فرض می کنیم رأس r به رأس u از s نزدیکتر باشد. اگر m_{uv} روی یکی از رأس ها قرار بگیرد، فرض می کنیم که با رأس r منطبق باشد. با حذف یال $[r, s]$، درخت به دو زیر درخت تجزیه می شود، زیردرخت چپ را با $L(u, v)$ و زیردرخت راست را با $R(u, v)$ نمایش می دهیم . توجه داشته باشید که زیر درخت چپ $L(u, v)$ همواره شامل r و زیر درخت راست $R(u, v)$ همواره شامل s است.

برای هر زیر درخت از T مانند $T'=(V',E')$ و برای هر رأس از v مانند x ، $f(T'|x)$ را به صورت زیر تعریف می کنیم،

$$f(T'|x)=\sum_{v\in V'} w_i d(v_i,x). \qquad (۳. ۲)$$

از آنجایی که در مدل ۲-ماکسین مشتریان به وسیله دورترین سرویس دهنده سرویس دهی می شوند، برای هر جفت u ,v از رئوس می توانیم نشان دهیم،

$$F(u,v)= f(R(u,v)|u)+ f(L(u,v)|v). \qquad (۳. ۳)$$

با توجه به تعریف $F(u,v)$ داریم،

$$F(u,v)=\sum_{v_i \in V} w_i \ \ max(d(v_i,u),d(v_i,v))$$

بعد از حذف یال $[r,s]$ و تقسیم شدن درخت به دو زیر درخت، رئوس v_i یا به $L(u,v)$ متعلق هستند یا به $R(u,v)$، در نتیجه،

$$F(u,v)=\sum_{v_i \in L(u,v)} w_i \ \ max(d(v_i,u),d(v_i,v))+\sum_{v_i \in R(u,v)} w_i \ \ max(d(v_i,u),d(v_i,v))$$

حال چون در مسأله ۲-ماکسین ، انتخاب دورترین سرویس دهنده مد نظر است، پس،

$$, \ \ \max_{v_i \in R(u,v)}(d(v_i,u),d(v_i,v))=d(v_i,u) \ \ \max_{v_i \in L(u,v)}(d(v_i,u),d(v_i,v))=d(v_i,v)$$

در نتیجه،

$$\begin{aligned} F(u,v) &= \sum_{v_i \in R(u,v)} w_i d(v_i,v)+\sum_{v_i \in R(u,v)} w_i d(v_i,u) \\ &= f(L(u,v)|v)+f(R(u,v)|u). \end{aligned}$$

قضیه۳. ۱. [۱۸]، فرض کنید $p(a, b)$ طولانی ترین مسیر درخت T با رئوس انتهایی b , a باشد. بنابراین $X:=\{a, b\}$ یک جواب بهینه برای مسأله ۲- ماکسین روی درخت است.

اثبات

فرض کنید b, a دو نقطه پایانی از طولانی ترین مسیر در T باشد. از آنجایی که مسأله ۲- ماکسین روی درختان دارای خاصیت بهینگی رأسی است به عبارت دیگر، حداقل یک مجموعه

متشکل از ۲ رأس در T وجود دارد که یک جواب برای مسأله ۲- ماکسین باشد، به[۱۱] نگاه کنید، کافی است نشان دهیم که برای هر $u,v\in V$

$$F(a,b)\geq F(u,v)\ .$$

از آنجایی که $p(a,b)$ طولانی ترین مسیر با نقطه میانی m_{ab} می باشد، واضح است که برای هر $x\in L(a,b)$،

$$d(x,m_{ab})\leq d(a,m_{ab}) \qquad (۳. ۴)$$

و برای هر $y\in R(a,b)$،

$$d(y,m_{ab})\leq d(b,m_{ab})$$

حال نشان می دهیم که، برای همه $x\in L(a,b)$،

$$f(R(a,b)|a)\geq f(R(a,b)|x) \qquad (۳. ۵)$$

با توجه به رابطه (۳. ۴) و اینکه $w_i\geq 0$ هستند، برای هر $x\in L(a,b)$ داریم،

$$f(R(a,b)|x)=\sum_{v_i\in R(a.b)}w_id(v_i,x)\ \leq\ \sum_{v_i\in R(a.b)}w_id(v_i,a)=f(R(a,b)|a)$$

در گام بعدی ما نشان می دهیم برای هر جفت از رئوس x و y در $L(a,b)$ ،

$$d(x,y)\leq d(x,b) \qquad (۳. ۶)$$

بر اساس نامساوی مثلث داریم،

$$d(x,y)\leq d(x,m_{ab})+d(m_{ab},y) \qquad x,y\in L(a,b)$$

و با توجه به نامساوی (۳. ۴) داریم،

$$d(m_{ab},y)\le d(m_{ab},a)=d(m_{ab},b)$$

که از این ها رابطه (۳. ۶) نتیجه می شود. یعنی،

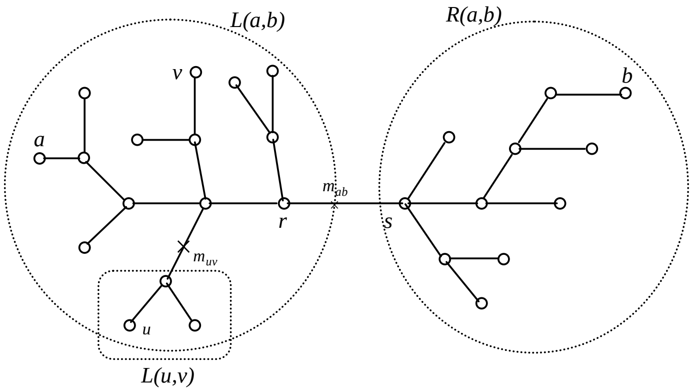
$$\begin{aligned} d(x,y) &\le d(x,m_{ab})+d(m_{ab},y)\le d(x,m_{ab})+d(m_{ab},b) \\ &= d(x,b). \end{aligned}$$

حال دو رأس u و v را از درخت T در نظربگیرید به طوری که m_{uv} نقطه میانی مسیر $p(u, v)$ باشد. ما دو حالت را در نظر می گیریم.

اول اینکه، $u,v\in L(a,b)$

ثانیا، $u\in L(a,b) \quad \& \quad v\in R(a,b)$

حالت اول،



شکل (۳. ۱) زیر درخت (L(a ,b و (R(a ,b از درخت T که با حذف یال [r ,s] به دست می آید و زیر درخت (L(u ,v با فرض $u,v\in L(a,b)$.

بگیرید $u,v\in L(a,b)$، (شکل (۳. ۱) را ببینید.) فرض میکنیم $d(v,b)\le d(u,b)$. این بدین معنی است که $d(v,m_{ab})\le d(u,m_{ab})$ و از طرفی چون $u,v\in L(a,b)$ داریم،

$$L(u,v) \subseteq L(a,b). \tag{۳. ۷}$$

$$R(a,b) \subseteq R(u,v). \tag{۳. ۸}$$

با توجه به رابطه (۳. ۶) (وقتی $x \in L(a,b)$، x یا در $L(u,v)$ قرار می گیرد و یا در $R(u,v) \setminus R(a,b)$.)، برای هر $x \in L(u,v)$ داریم،

$$d(x,v) \le d(x,b)$$

و برای هر $x \in R(u,v), x \notin R(a,b)$،

$$d(x,u) \le d(x,b)$$

که از دو رابطه بالا رابطه زیر نتیجه می شود،

$$f(L(a,b)|b) \ge f(L(u,v)|v) + \left[f(R(u,v)|u) - f(R(a,b)|u) \right] \tag{۳. ۹}$$

$$\begin{aligned} f(L(a,b)|b) &= \sum_{v_i \in L(a,b)} w_i d(v_i,b) = \sum_{v_i \in L(u,v)} w_i d(v_i,b) + \sum_{v_i \in R(u,v)-R(a,b)} w_i d(v_i,b) \\ &\ge \sum_{v_i \in L(u,v)} w_i d(v_i,v) + \sum_{v_i \in R(u,v)-R(a,b)} w_i d(v_i,u) \\ &= f(L(u,v)|v) + \left[f(R(u,v)|u) - f(R(a,b)|u) \right]. \end{aligned}$$

با توجه به نامساوی (۳. ۵) چون $u \in L(a,b)$ داریم،

$$f(R(a,b)|a) \ge f(R(a,b)|u) \tag{۳. ۱۰}$$

که نشان می دهد،

$$F(a,b) = f(L(a,b)|b) + f(R(a,b)|a) \ge$$

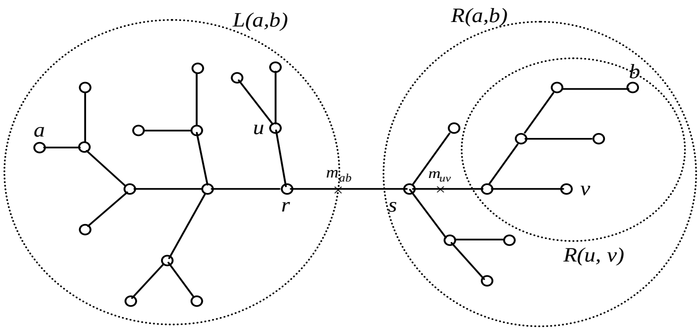
$$f(L(u,v)|v)+\left[f(R(u,v)|u)-f(R(a,b)|u)\right]+f(R(a,b)|u)$$

$$=f(L(u,v)|v)+f(R(u,v)|u)$$

$$=F(u,v).$$

$$\Rightarrow F(a,b)\geq F(u,v).$$

حالت دوم، بگیرید $u\in L(a,b)$ و $v\in R(a,b)$. (شکل (۳. ۲) را ببینید.)



شکل (۳. ۲) زیر درخت (R(u ,v با فرض $u\in L(a,b)$ و $v\in R(a,b)$ و $m_{uv}\in R(a,b)$.

اگر m_{uv} روی همان یالی بیفتد که m_{ab} قرار دارد (نه منطبق با رأس s)، بنابراین،

$$L(a,b)=L(u,v) \qquad , \qquad R(a,b)=R(u,v).$$

از آنجایی که،

(زیرا $u\in L(a,b)$) $\qquad d(u,m_{ab})\leq d(a,m_{ab})$

(زیرا $v \in R(a,b)$) $d(v,m_{ab}) \le d(b,m_{ab})$

داریم،

$$f(R(a,b)|a) \ge f(R(a,b)|u) = f(R(u,v)|u)$$

$$f(L(a,b)|b) \ge f(L(a,b)|v) = f(L(u,v)|v)$$

$$\Rightarrow f(R(a,b)|a) + f(L(a,b)|b) \ge f(R(u,v)|u + f(L(u,v)|v)$$

$$F(a,b) \ge F(u,v).$$

اگر m_{uv} روی همان یالی که m_{ab} قرار دارد نیفتد، ما فرض می کنیم $m_{uv} \in R(a,b)$.

این نشان می دهد که $L(a,b) \subseteq L(u,v)$ $R(u,v) \subseteq R(a,b)$.

از آنجایی که برای هر رأس $x \in L(a,b)$، $d(x,v) \le d(x,b)$ داریم،

(۳. ۱۱)

$f(L(a,b)|b) \ge f(L(a,b)|v)$ بعلاوه، ما برای هر رأس $x \in R(a,b)$ داریم،

(۳. ۱۲) $d(x,u) \le d(x,a)$

(۳. ۱۳) $d(x,v) \le d(x,a)$

هر رأس $x \in R(a,b)$ یا در $R(u,v)$ قرار می گیرد یا در $L(u,v) \backslash L(a,b)$ ، حال با توجه به روابط (۳. ۱۲) و (۳. ۱۳) می توانیم نشان دهیم که،

$$f(R(a,b)|a) \ge f(R(u,v)|u) + [f(L(u,v)|v) - f(L(a,b)|v])$$

$$f(R(a,b)|a)=\sum_{v_i\in R(a,b)} w_i d(v_i,a)$$

$$=\sum_{v_i\in R(u,v)} w_i d(v_i,a)+\sum_{v_i\in L(u,v)-L(a,b)} w_i d(v_i,a)$$

$$\geq \sum_{v_i\in R(u,v)} w_i d(v_i,u)+\sum_{v_i\in L(u,v)-L(a,b)} w_i d(v_i,v)$$

$$=f(R(u,v)|u)+[f(L(u,v)|v)-f(L(a,b)|v)]$$

$$\Rightarrow f(R(a,b)|a)\geq f(R(u,v)|u)+[f(L(u,v)|v)-f(L(a,b)|v]). \qquad (۳. ۱۴)$$

با توجه به دو رابطه (۳. ۱۱) و (۳. ۱۴) داریم،

$$F(a,b) = f(R(a,b)|a) + f(L(a,b)|b)$$

$$\geq f(R(u,v)|u)+[f(L(u,v)|v)-f(L(a,b)|v)]+f(L(a,b)|v)$$

$$=f(R(u,v)|u) + f(L(u,v|v)=F(u,v). \quad \square$$

نتیجه۳. ۲. [۱۸] : مجموعه X مسأله ی ۲-ماکسین به وزنهای مثبت رئوس w_i بستگی ندارد. همه وزنهای مثبت ، ۲-ماکسین یکسانی را حاصل میکنند.

اکنون ما نشان خواهیم داد که مقدار تابع هدف بهینه برای $p>2$ با حالتی که $p=2$ معادل است.

لم ۳. ۳. [۱۸] : فرض کنید X یک مجموعه شامل p رأس مجزا در درخت T باشد. اگر X شامل دو رأس a,b به گونه ای باشد که $p(a,b)$ طولانی ترین مسیر در درخت T باشد، آنگاه X یک جواب بهینه برای مسأله p –ماکسین روی درخت T است.

اثبات

فرض کنید $X=\{x_1,x_2,\dots,x_p\}$ یک جواب بهینه برای مسأله p- ماکسین باشد.

فرض کنید که $a,b\in X$ دو نقطه از T هستند به طوریکه،

$$d(a,b)=\max_{x_i,x_j\in X} d(x_i,x_j) \tag{۳. ۱۵}$$

همانند قبل فرض کنید m_{ab} نقطه میانی مسیر بین a,b باشد، مانند اثبات قضیه ۳. ۱ ما می توانیم نشان دهیم که $d(x,b)\geq d(x,x_i)$ برای هر رأس $x\in L(a,b)$ و هر $x_i\in X$ و برای هر رأس $x\in R(a,b)$ و هر $x_i\in X$، $d(x,a)\geq d(x,x_i)$.

بنابراین همه رئوس در $L(a,b)$ به b اختصاص داده شده اند و مشابها، همه رئوس در $R(a,b)$ به a اختصاص داده شده اند. به عبارتی دیگر دورترین مکان در درخت برای تمام رئوس واقع شده در $L(a, b)$ رأس b و برای تمام رئوس واقع شده در $R(a, b)$ دورترین رأس به آنها a می باشد. بنابراین :

$F(X)=f(L(a,b)|b)+f(R(a,b)|a).$

این نشان می دهد که هر مسأله p- ماکسین با $p\geq 2$ می تواند به مسأله ۲- ماکسین کاهش بیابد. بر طبق قضیه ۳. ۱ جواب بهینه مسأله ۲-ماکسین به وسیله طولانی ترین مسیر در درخت T بدست می آید و در نتیجه لم ۳. ۳ اثبات می شود. □

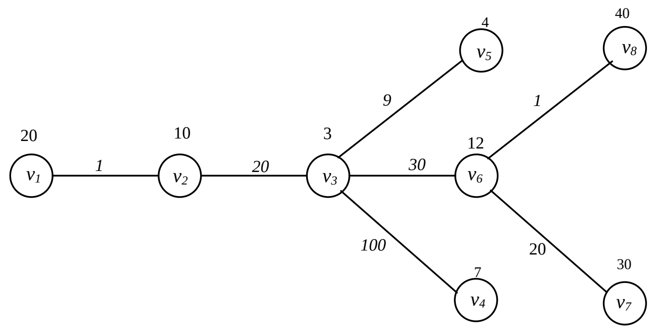
بر طبق هندلر [۲۰] طولانی ترین مسیر در درخت می تواند در زمان خطی یافت شود. بنابراین داریم،

قضیه ۳. ۴. [۱۸]: مسأله p-ماکسین روی یک درخت می تواند در زمان خطی حل شود.

اثبات

یک جواب بهینه برای مسأله p-ماکسین روی درخت T به وسیله طولانی ترین مسیر روی T به دست می آید. از آنجایی که یافتن طولانی ترین مسیر روی درخت در زمان خطی به دست می آید، در نتیجه مسأله p-ماکسین روی یک درخت در زمان خطی حل می شود. $\square$

مثال ۳. ۱. درخت در شکل (۳. ۳) را در نظر بگیرید. وزن هر رأس و فاصله ی بین رئوس داده شده است.



شکل (۳. ۳) یک درخت، با وزن و فاصله ی بین رئوس

با توجه به فواصل داده شده، $P(V_4,V_7)$ طولانی ترین مسیر روی درخت در شکل (۳. ۳) می باشد، بنابر قضیه۳. ۱، مجموعه ی $X:=\{V_4,V_7\}$ جواب بهینه برای مسأله ۲-ماکسین با مقدار تابع $F(V_4,V_7)=11156$ روی درخت می باشد.

فصل چهارم

مسأله p- ماکسین روی گراف های بازه ای

۱.۴ مقدمه

در این فصل مسأله p-ماکسین روی گراف های بازه ای که هر بازه دارای یک وزن مثبت است را بررسی می کنیم. ابتدا یک الگوریتم خطی که شبیه الگوریتم بسپامیاتنیخ[30] [۲۱] برای مسأله ۱- میانه بدون وزن روی گراف های بازه ای، برای حل مسأله ۱- ماکسین وزن دار ارائه می شود [۱۷].

برای مسأله ۲-ماکسین، نشان می دهیم که دو بازه به ترتیب با می نیمم نقطه پایانی راست و ماکسیمم نقطه پایانی چپ یک جواب بهینه هستند که می توان آن را به مسأله p-ماکسین تعمیم داد.

۲.۴ گراف های بازه ای

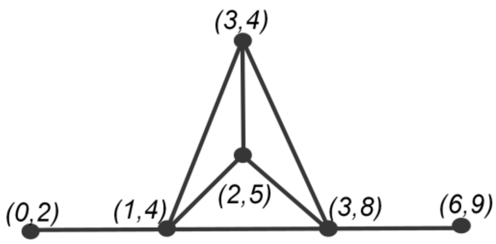
گراف $G=(V,E)$ یک گراف بازه ای است اگر یک مجموعه S از بازه های حقیقی I_i وجود داشته باشد به طوریکه یک تناظر یک به یک بین رئوس $v_i \in V$ و $I_i \in S$ برقرار باشد، در این صورت یک یال $(v_i, v_j) \in E$ اگر و تنها اگر $I_i \cap I_j \neq \phi$. یک مجموعه بازه ای S، یک مدل از گراف بازه ای G نامیده می شود.

مثال ۴. ۱. به عنوان مثال می خواهیم گراف بازه ای متناظر با بازه های زیر را رسم کنیم،

$(6,9), (3,8), (3,4), (2,5), (1,4), (0,2)$

[30] Bespamyatnikh

دو بازه $(0,2)$, $(1,4)$ اشتراک ناتهی دارند، لذا رأس های متناظر این دو بازه را با یک یال به هم وصل می کنیم. ولی دو بازه $(0,2)$, $(2,5)$ اشتراکشان تهی است، پس رأس های متناظر این دو بازه به هم وصل نمی شوند. به این ترتیب با همین استدلال نمودار گراف بازه ای شش بازه فوق به صورت زیر در می آید.



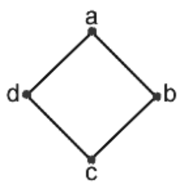
شکل (۴. ۱) گراف بازه ای متناظر با شش بازه

سوألی که پیش می آید این است که چه گراف‌هایی می‌توانند بازه‌ای باشند؟

قضیه۴. ۱. [۳۷] : یک گراف اگر دارای حفره باشد حتماً بازه‌ای نمی‌باشد.

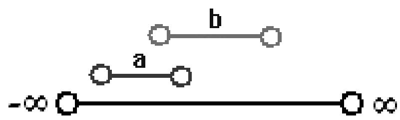
اثبات

فرض می کنیم دور مرتبه ۴ در شکل (۴. ۲) خود یک گراف یا قسمتی از یک گراف باشد،



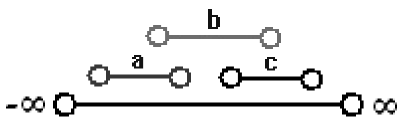
شکل (۴. ۲) یک دور از مرتبه ۴

نشان می دهیم این گراف و یا گرافی شامل این دور بازه ای نمی باشد. به برهان خلف فرض می کنیم این گراف یا گراف شامل این دور بازه ای باشد. روی محور اعداد حقیقی برای هر یک از رأس ها بازه ای به صورت زیر در نظر می گیریم، چون a با b مجاور است باید روی محور اعداد بازه های متناظر با این دو رأس دارای اشتراک باشند مطابق شکل(۴. ۳) ،



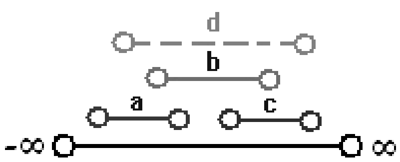
شکل (۴. ۳) دو بازه متناظر با رأس b،a روی محور اعداد حقیقی

از طرفی c نیز با b مجاور است و با a مجاور نمی باشد پس بازه ی متناظر با c با بازه ی متناظر b اشتراک دارد ولی با بازه متناظر a اشتراک ندارد. مطابق شکل (۴. ۴)،



شکل (۴. ۴) بازه های متناظر با رئوس c،b،a روی محور اعداد حقیقی

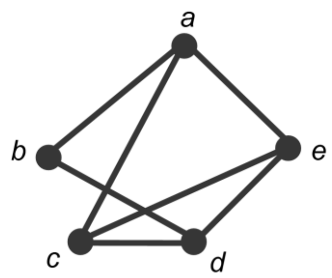
حال چون d هم با a و هم با c مجاور است پس بازه ی متناظر با راس d باید به گونه ای اشد که هم با a و هم با c اشتراک داشته باشد و این تناقض است چرا که در این صورت d با b هم اشتراک پیدا می کند در حالی که از b به d یالی رسم نشده است.



شکل (۴. ۵) بازه های متناظر با رئوس d،c،b،a روی محور اعداد حقیقی

پس فرض خلف باطل و حکم برقرار است .

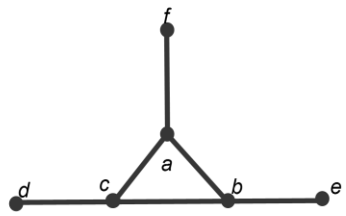
به عنوان مثال در گراف شکل (۴. ۶)، $abde$ یک حفره محسوب می شود و لذا گراف همان طور که گفته شد بازه ای نمی باشد.



شکل (۴. ۶) گراف شامل حفره بازه ای نمی باشد.

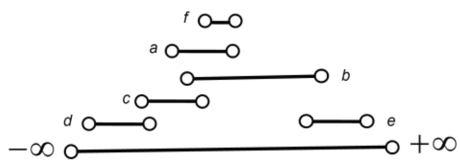
نکته ۴. ۱. اگر در گرافی حفره مشاهده نشد نمی توان نتیجه گرفت لزوما گراف بازه ای است.

به عنوان مثال گراف شکل (۴. ۷) دارای حفره نمی باشد ولی در عین حال بازه ای نیز نمی باشد.

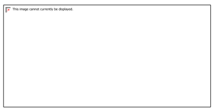


شکل (۴. ۷) گرافی که شامل حفره نیست، لزوما بازه ای نیست.

زیرا بر طبق شکل (۴. ۸)، f هر کجا باشد به هر حال باید به یک بازه ی دیگر به غیر از a وصل شود.



شکل (۴. ۸) بازه های متناظر با رئوس a،b،c،d،e،f روی محور اعداد حقیقی

نکته ۴. ۲. اگر در گراف بازه  ای یک n ضلعی $(n \geq 4)$ وجود داشته باشد، حداقل یکی از قطرهای n ضلعی باید یالی از گراف باشد در غیر این صورت گراف بازه ای نمی باشد .

نکته ۴. ۳. اگر قسمتی از گراف بازه ای نباشد کل آن گراف بازه ای نیست.

به عنوان مثال گراف شکل (۴. ۹) بازه ای نیست، زیرا چهار ضلعی بدون قطر دارد.

شکل (۹،۴) چهار ضلعی بدون قطر بازه ای نیست.

گراف های بازه ای توسط محققان بسیاری [۲۶، ۲۵، ۲۴، ۲۳، ۲۲] مطالعه شده اند، گراف های بازه ای ابزار مهمی در بسیاری از حوزه های کاربردی مانند باستان شناسی، زیست شناسی، زمان بندی و کنترل ترافیک به حساب می آیند [۲۷، ۱۰]. از این رو بسیاری از محققان به مسائل مکانیابی روی گراف های بازه ای، علاقه مندند [۲۹، ۲۱].

۴. ۳ تعریف مسأله

بدون کاستن از کلیت مسأله، رئوس G را به عنوان بازه ها در نظر می گیریم. فرض می کنیم هر بازه $I_i \in S$ متناظر با یک مشتری و دارای یک وزن مثبت $w(I_i)$ باشد و پیدا کردن مکان بهینه برای سرویس دهنده به بازه ها محدود شود. همچنین فرض می کنیم هر یال در E دارای طول واحد

باشد. کوتاهترین مسیر بین بازه I_i, I_j را با $P(I_i, I_j)$ و طول مسیر بین I_i, I_j به عبارت دیگر تعداد یالها روی $P(I_i, I_j)$ را با $d(I_i, I_j)$ نمایش می دهیم .

مسأله p-ماکسین روی گراف های بازه ای به صورت زیر تعریف می شود،

هدف یافتن یک مجموعه $\bar{S}=\{I_{i_1},...I_{i_p}\}$ (I_{i_k}, $1\le k\le p$ را میانه ی ماکسیمم می نامیم) به طوریکه تابع هدف زیر ماکسیمم شود.

$$F(\bar{s})=\sum_{k=1}^{n} w(I_k)\max_{1\le j\le p} d(I_k, I_{i_j}).$$

۴.۴ برخی خواص مسأله

فرض کنید مجموعه S یک مدل بازه ای برای گراف بازه ای G باشد که شامل n بازه است. هر بازه $I_i\in S$ به وسیله نقطه پایانی چپ خود a_i و نقطه پایانی راست خود b_i ، یعنی $I_i=[a_i, b_i]$، $a_i\le b_i$ تعریف می شود. بدون کاستن از کلیت مسأله، فرض می کنیم دو بازه مجزا در S نقطه پایانی یکسانی ندارند و برای هر $a_i<b_i, i=1,2,...,n$. L را آرایه ای درجه بندی شده شامل نقاط پایانی بازه ها در S در نظر بگیرید. یعنی $L=\{p_1, p_2,...,p_{2n}\}$، که p_j یا a_i و یا b_i برای بازه $I_i\in S$ می باشد. توجه کنید که به وسیله L به آسانی می توانیم لیست همه باز ها در S را درجه بندی شده به وسیله a'_{is} (به ترتیب b'_{is}) به دست آوریم. پس از این، فرض می کنیم که بازه ها در S به گونه ای انتخاب شده باشند که $b_i<b_j$ اگر و تنها اگر $i<j$. s و t را به ترتیب کوچکترین نقطه پایانی چپ و بزرگترین نقطه پایانی راست باز های موجود در L درنظر بگیرید. پس $s=a_l$ که $l\in\{1,...,n\}$ و $t=b_n$.

بدون کاستن از کلیت مسأله، فرض می کنیم که $\bigcup_{i=1}^{n} I_i$ با بازه ی $[s,t]$ معادل است به عبارت دیگر اجتماع بازه ها در S یک مؤلفه متصل مانند $[s,t]$ روی اعداد حقیقی می باشد.

چن[31] [۲۳] یک تابع جانشین روی بازه ها برای حل مسأله کوتاهترین مسیر روی گراف های بازه ای معرفی کرده است. بسپامیاتینخ [۲۱] از ایده چن برای تعریف یک تابع جانشین راست و یک تابع جانشین چپ برای یک عدد صحیح q به صورت زیر استفاده کرد.

برای هر نقطه q روی اعداد حقیقی،

$RSUC(q) = I_i \in S$ اگر و تنها اگر $\{I_j$ شامل q است $| b_j\}$ $b_i = max$

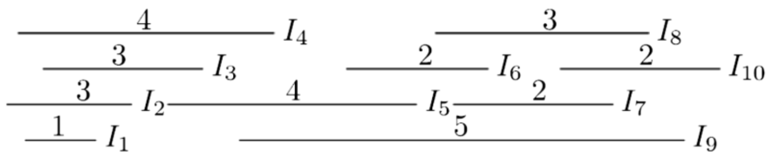
و $LSUC(q) = I_i \in S$ اگر و تنها اگر $\{I_j$ شامل q است $| a_j\}$ $a_i = min$.

برای یک بازه ی I_i، $RSUC(I_i) = RSUC(b_i)$ و $LSUC(I_i) = LSUC(a_i)$. توجه کنید از آنجایی که $\bigcup_{i=1}^{n} I_i$ یک مؤلفه متصل است، $RSUC(I_i) = I_i$ نشان می دهد که $i=n$ و $LSUC(I_i) = I_i$ نشان می دهد که I_i بازه ای با کوچکترین نقطه پایانی چپ می باشد. بعلاوه، i امین تابع جانشین راست نقطه q یعنی $RSUC(q,i)$ برای هر عدد صحیح i به صورت $RSUC(RSUC(..RSUC(q,0)))$ تعریف می شودکه $RSUC$ ، $i+1$ بار محاسبه می شود و $RSUC(q,0) = q$. مشابهاً، ما می توانیم $LSUC(q,i)$ را به صورت $LSUC(LSUC(..LSUC(q,0)))$ تعریف کنیم. برای مثال گراف بازه ای با ده رأس را در نظر بگیرید، بازه های متناظر با رئوس گراف

---

[31] Chen

بازه ای روی محور اعداد حقیقی در شکل (۴. ۱۰)، که مجموعه ای از ۱۰ بازه است آورده شده است، داریم،



شکل (۴. ۱۰) مجموعه ای از ۱۰ بازه ، شماره روی هر بازه، وزن بازه است.

$$RSUC(I_1)=I_4 \qquad RSUC(I_4)=I_9 \qquad RSUC(I_9)=I_{10}$$

$$LSUC(I_{10})=I_9 \qquad LSUC(I_9)=I_4 \qquad LSUC(I_4)=I_2$$

$$RSUC(I_1,3)=RSUC(RSUC(RSUC(RSUC(I_1,0))))$$

$$=RSUC(RSUC(RSUC(I_1)))$$

$$=RSUC(RSUC(I_4))$$

$$=RSUC(I_9)=I_{10}$$

$$LSUC(I_{10},3)=LSUC(LSUC(LSUC(LSUC(I_{10},0))))$$

$$=LSUC(LSUC(LSUC(I_{10})))$$

$$=LSUC(LSUC(I_9))$$

$$=LSUC(I_4)=I_2$$

چن [۲۳] یک ساختار درخت پایه *(RSUC-TREE)* با استفاده از تابع جانشین راست *RSUC* ساخته است که در این ساختار هر گره یا رأس متناظر با یک بازه در S می باشد. برای هر $I_i \in S$ ، $1 \le i \le n$ ، $RSUC(I_i)$ پدر گره ی متناظر با بازه ی I_i در *RSUC-TREE* می باشد. مشابهاً ما ساختار درخت دیگری *(LSUC-TREE)* با استفاده از تابع جانشین چپ *LSUC* می سازیم که در آن هر گره یا رأس متناظر با یک بازه در S می باشد و $LSUC(I_i)$ پدر گره ی متناظر با بازه ی I_i به استثنای بازه ی I_i با کوچکترین نقطه پایانی چپ است.

با استفاده از ساختار *RSUC-TREE* ، چن و همکاران [۲۳] تعدادی لم مهم برای محاسبه مسافت بین هر دو بازه در S مطرح کردند. در اینجا دو بازه دارای اشتراک I_i, I_j را با $I_i \sim I_j$ و هر دوبازه غیر مشترک را با $I_i \nsim I_j$ نمایش می دهیم.

مستقیماً از تعریف تابع جانشین راست، می توانیم به راحتی نتایج زیر را به دست آوریم. توجه داشته باشید که تمامی قضییه ها و لم ها و نتایج گفته شده در این فصل بر روی گراف های بازه ای قابل اثبات است.

نتیجه ۴. ۲. [۱۷] : برای هر دو بازه I_i و I_j با $i < j$ به عبارت دیگر $b_i < b_j$، (طبق فرضی که در بخش چهارم کردیم.) کوتاهترین مسیر $p(I_i,I_j)=\{I_i,RSUC(I_i,1),(RSUC(I_i,d-1),I_j\}$ به طول d وجود دارد.

یک مدل بازه ای S برای گراف G و p_{2n} بزرگترین نقطه در L مفروض است، برای هر نقطه ی $p_i \in L$ ما آن را با مقدار $p_{2n}-p_i$ جایگزین می کنیم، آنگاه یک مدل بازه ای جدید S' برای G به دست می آوریم. در این صورت، $RSUC-TREE$ از مدل جدید با *LSUC-TREE* از مدل قبل معادل است . به طور مشابه، می توانیم نتیجه ی زیر را به دست آوریم.

نتیجه ۴. ۳. [۱۷] : برای هر دو بازه ی I_i و I_j در S با $a_i < a_j$، کوتاهترین مسیر $p(I_i,I_j)=\{I_j, LSUC(I_j,1),\ldots,LSUC(I_j,d-1), I_i\}$ به طول d وجود دارد.

لم ۴. ۴. [۱۷] : (۱) اگر سه بازه ی I_i,I_h و I_j در S با $b_h < b_i$ ، $b_h < b_j$ و $a_i < a_j$ وجود داشته باشد، آنگاه،

$$d(I_h,I_i) \le d(I_h,I_j);$$

(۲) اگر سه بازه ی I_j,I_i,I_h در S با $a_h > a_i$، $a_h > a_j$، $b_i < b_j$ وجود داشته باشد، آنگاه،

$$d(I_h,I_j) \le d(I_h,I_i).$$

اثبات

(۱) با استفاده از نتیجه ۴. ۲، دو عدد صحیح k و l وجود دارند به طوری که،

$$p(I_h,I_i)=\{I_h,RSUC(I_h,1),\ldots,RSUC(I_h,k-1),I_i\};$$

$$p(I_h,I_j)=\{I_h,RSUC(I_h,1),\ldots,RSUC(I_h,l-1),I_j\}.$$

از آنجایی که طبق فرض مسأله $a_i < a_j$ واضح است که $k \le l$ بنابراین $d(I_h,I_i) \le d(I_h,I_j)$.

(۲) با استفاده از نتیجه ۴. ۳ ، برای دوبازه ی I_h و I_i، چون $a_i < a_h$ و برای دو بازه ی I_j و I_h چون $a_j < a_h$ دو عدد صحیح l, k وجود دارند به طوری که،

$$p(I_h,I_i)=\{I_i,LSUC(I_i,1),\ldots,LSUC(I_i,k-1),I_h\};$$

$$p(I_h,I_j)=\{I_j,LSUC(I_j,1),\ldots,LSUC(I_j,l-1),I_h\}.$$

از آنجایی که طبق فرض مسأله $b_i < b_j$ لذا $k \geq l$ و در نتیجه $d(I_h,I_j) \leq d(I_h,I_i)$. □

نتیجه ۴. ۵. [۱۷] : اگر بازه I_i مشمول در I_j باشد، به عبارت دیگر $a_j < a_i < b_i < b_j$ ، آنگاه برای هر بازه دیگر I_h $(I_h \neq I_i, I_j)$، $d(I_h,I_j) \leq d(I_h,I_i)$.

اثبات

برای اثبات سه حالت زیر را در نظر می گیریم،

حالت اول. اگر $I_h \sim I_j$، آنگاه $d(I_h,I_j) = 1 \leq d(I_h,I_i)$ ؛

حالت دوم. اگر $I_i \nsim I_j$ و $b_h < a_j$، آنگاه واضح است که $b_h < b_j,\ b_h < b_i$ و $a_j < a_i$. با استفاده از (۱) در لم ۴. ۴ ، نتیجه می گیریم که $d(I_h,I_j) \leq d(I_h,I_i)$.

حالت سوم. اگر $I_i \nsim I_j$ و $a_h > b_j$، آنگاه واضح است که $a_h > a_i$ ، $a_h > a_j$ و $b_i < b_j$. با استفاده از (۲) در لم ۴. ۴، داریم که $d(I_h,I_j) \leq d(I_h,I_i)$. □

یک بازه در S را کمینه می نامیم اگر آن بازه شامل هیچ بازه ی دیگری در S نباشد. نتیجه ۴. ۵ نشان می دهد که ، هر بازه I_j که کمینه نیست نباید برای یک میانه ی ماکسیمم کاندیدا شود زیرا بازه ی کمینه ی I_i مشمول در I_j می تواند به عنوان میانه ی ماکسیمم جایگزین I_j شود بدون آنکه مقدار تابع هدف کاهش یابد.

۴. ۵ مسأله ۱- ماکسین روی گراف بازه ای

در این بخش ما مسأله ۱- ماکسین روی گراف های بازه ای را درنظر می گیریم، یعنی یک میانه ماکسیمم را در یک بازه به گونه ای قرار دهیم که مجموع فاصله وزنی از بقیه بازه ها ماکسیمم شود. ابتدا تعدادی نماد جدید برای مجموع وزن ها برای نقطه $q \in L$ تعریف می کنیم.

$$W_{aL}(q)=\sum_{I_j\in S|a_j<q} w(I_j) \qquad ; \qquad W_{bL}(q)=\sum_{I_j\in S|b_j<q} w(I_j) \qquad ;$$

$$W_{aR}(q)=\sum_{I_j\in S|a_j>q} w(I_j) \qquad ; \qquad W_{bR}(q)=\sum_{I_j\in S|b_j>q} w(I_j) \qquad .$$

حال فرمول های تکراری زیر را برای محاسبه $W_{aL}(q)$، $W_{bL}(q)$، $W_{aR}(q)$ و $W_{bR}(q)$ برای نقطه $q \in L$ مطرح می کنیم.

۴. ۵. ۱ قاعده تکراری

$$1) W_{aL}(p_1)=0;$$

$$W_{aL}(p_{i+1})=\begin{cases} W_{aL}(p_i)+w(I_j) & p_i=a_j \quad \text{اگر} \\ W_{aL}(p_i) & p_i=b_j \quad \text{اگر} \end{cases}$$

برای تعدادی بازه I_j در S ؛

$$2)\ W_{bL}(p_1)=\ldots=W_{bL}(p_l)=0 \qquad p_l=\min\{b_1,b_2,\ldots,b_n\};$$

$$W_{bL}(p_{i+1})=\begin{cases} W_{bL}(p_i)+w(I_j) & p_i=b_j \quad \text{اگر} \\ W_{bL}(p_i) & p_i=a_j \quad \text{اگر} \end{cases}$$

برای تعدادی بازه I_j در S ؛

$$3)W_{aR}(p_{2n})=\ldots=W_{aR}(p_l)=0 \qquad p_l=max\{a_1,a_2,\ldots,a_n\};$$

برای تعدادی بازه I_j در S ؛

$$W_{aR}(p_i)=\begin{cases} W_{aR}(p_{i+1})+w(I_j) & p_{i+1} = \text{ اگر } \\ W_{aR}(p_{i+1}) & p_{i+1} = \text{ اگر } \end{cases}$$

$$4)W_{bR}(p_{2n})=0;$$

برای تعدادی بازه I_j در S ؛

$$W_{bR}(p_i)=\begin{cases} W_{bR}(p_{i+1})+w(I_j) & p_{i+1}=b_j \text{ اگر } \\ W_{bR}(p_{i+1}) & p_{i+1}=a_j \text{ اگر } \end{cases}$$

با استفاده از قاعده تکراری می توانیم $W_{aL}(q)$، $W_{bL}(q)$، $W_{aR}(q)$ و $W_{bR}(q)$ را برای همه $q\in L$ در زمان $O(n)$ محاسبه کنیم.

مشابه ایده ی بسپامیاتیخ [۲۱]، دو تابع $F_{aR}(q),F_{aL}(q)$ را برای نقطه $q\in L$ که نقطه q یک نقطه پایانی بازه I_i است، تعریف می کنیم،

$$F_{aL}(q)=\sum_{I_j\in S|a_j<q} w(I_j)d(I_j,I_i),$$

$$F_{aR}(q)=\sum_{I_j\in S|a_j>q} w(I_j)d(I_j,I_i).$$

بنابراین برای هر کاندیدای میانه ی ماکسیمم I_i ، تابع هدف می تواند به صورت زیر فرمول بندی شود،

$$F(I_i)=\sum_{I_j\in S} w(I_j)d(I_j,I_i)=F_{aL}(a_i)+F_{aR}(a_i).$$

به عبارت دیگر برای یافتن یک جواب بهینه از مسأله ۱- ماکسین، باید $F_{aL}(q)$ و $F_{aR}(q)$ را برای هر نقطه $q \in L$ محاسبه کنیم و بازه I_k به طوری که $F(I_k)$ ماکسیمم شود را بیابیم.

لم ۴. ۶. [۱۷] : برای هر بازه I_i با نقطه پایانی چپ a_i و نقطه پایانی راست b_i داریم،

$$F_{aR}(b_i) = F_{aR}\left(b_{RSUC(I_i)}\right) + 2W_{aR}(b_i) - W_{aR}\left(b_{RSUC(I_i)}\right);$$

$$F_{aR}(a_i) = F_{aR}(b_i) + W_{aR}(a_i) - W_{aR}(b_i);$$

$$F_{aL}(a_i) = F_{aL}\left(a_{LSUC(I_i)}\right) + W_{aL}(a_i) + W_{bL}(a_i) - W_{aL}\left(b_{LSUC(I_i)}\right);$$

$$F_{aL}(b_i) = F_{aL}(a_i) + W_{aL}(b_i) - W_{aL}(a_i) - w(I_i).$$

اثبات

ابتدا درستی فرمول قسمت اول را نشان می دهیم. اگرنقطه $q = b_n$، آنگاه $\{I_j \in S | a_j > b_n\} = \phi$ بنابراین $F_{aR}(b_n) = 0$. برای b_i های دیگر، $i=1,2,...,n-1$ می توانیم مجموعه ی بازه ی $\{I_j \in S | a_j > b_i\}$ به دو مجموعه ی مجزای $\{I_j \in S | a_j > b_{RSUC(I_i)}\}$ و $\{I_j \in S | b_i < a_j < b_{RSUC(I_i)}\}$ تقسیم کنیم که $b_{RSUC(I_i)}$ نقطه پایانی راست $RSUC(I_i)$ می باشد.

برای هر بازه ی I_j در $\{I_j \in S | a_j > b_{RSUC(I_i)}\}$ ، با استفاده از نتیجه ۴. ۲ می توانیم بنویسیم،

$d(I_j, I_i) = d(I_j, RSUC(I_i)) + d(RSUC(I_i), I_i) = d(I_j, RSUC(I_i)) + 1$. برای هر بازه I_j در $\{I_j \in S | b_i < a_j < b_{RSUC(I_i)}\}$ ، با توجه به اینکه $a_j < b_{RSUC(I_i)}$ و $I_j \sim RSUC(I_i)$ و با توجه به اینکه $b_i < a_j$ در نتیجه $I_i \nsim I_j$.

از این رو $d(I_j, I_i) = d(I_j, RSUC(I_i)) + d(RSUC(I_i), I_i) = 1 + 1 = 2$

بنابراین،

$$F_{aR}(b_i)=\sum_{I_j\in S|a_j>b_i} w(I_j)d(I_j,I_i)$$

$$=\sum_{I_j\in S|a_j>b_{RSUC(I_i)}} w(I_j)d(I_j,I_i)+\sum_{I_j\in S|b_i<a_j<b_{RSUC(I_i)}} w(I_j)d(I_j,I_i)$$

$$=\sum_{I_j\in S|a_j>b_{RSUC(I_i)}} w(I_j)(d(I_j,RSUC(I_i))+1)+2\sum_{I_j\in S|b_i<a_j<b_{RSUC(I_i)}} w(I_j)$$

$$=F_{aR}\left(b_{RSUC(I_i)}\right)+W_{aR}\left(b_{RSUC(I_i)}\right)+2\left(\sum_{I_j\in S|a_j>b_i} w(I_j)-\sum_{I_j\in S|a_j>b_{RSUC(I_i)}} w(I_j)\right)$$

$$=F_{aR}\left(b_{RSUC(I_i)}\right)+2W_{aR}(b_i)-W_{aR}\left(b_{RSUC(I_i)}\right).$$

برای اثبات درستی قسمت دوم، مجموعه ی بازه ی $\{I_j\in S|a_j>a_i\}$ برای $i=1,2,...,n$ را می توانیم به دو مجموعه مجزای $\{I_j\in S|a_j>b_i\}$ و $\{I_j\in S|a_i<a_j<b_i\}$ تقسیم کنیم. برای مجموعه $\{I_j\in S|a_i<a_j<b_i\}$ چون $I_i\sim I_j$ لذا $d(I_i, I_j)=1$.

بنابراین،

$$F_{aR}(a_i)=\sum_{I_j\in S|a_j>a_i} w(I_j)d(I_j,I_i)$$

$$=\sum_{I_j\in S|a_j>b_i} w(I_j)d(I_j,I_i)+\sum_{I_j\in S|a_i<a_j<b_i} w(I_j)d(I_j,I_i)$$

$$=F_{aR}(b_i)+\left(\sum_{I_j\in S|a_j>a_i} w(I_j)-\sum_{I_j\in S|a_j>b_i} w(I_j)\right)$$

$$= F_{aR}(b_i) + W_{aR}(a_i) - W_{aR}(b_i)\ .$$

برای اثبات قسمت سوم، مجموعه بازه ی $\{ I_j \in S \mid a_j < a_i \}$ را می توانیم به دو مجموعه ی $\{ I_j \in S \mid a_j < a_{LSUC(I_i)} \}$ و $\{ I_j \in S \mid a_{LSUC(I_i)} < a_j < a_i \}$ تقسیم کنیم که مجموعه ی $\{ I_j \in S \mid b_j < a_i < a_j \}$ در هر دوی آنها مشترک است. $a_{LSUC(I_i)}$ نقطه پایانی چپ $LSUC(I_i)$ است. برای هر بازه I_j در $\{ I_j \in S \mid a_j < a_{LSUC(I_i)} \}$ با استفاده از نتیجه ۴. ۳ داریم،

$$d(I_j, I_i) = d(I_i, LSUC(I_i)) + d(LSUC(I_i), I_j)$$

$$= 1 + d(LSUC(I_i)), I_j).$$

برای هر بازه ی I_j در $\{ I_j \in S \mid a_{LSUC(I_i)} < a_j < a_i \}$، با توجه به اینکه $a_{LSUC(I_i)} < a_j < a_i$، $I_j \sim LSUC(I_i)$ و $I_i \nsim I_j$. بنابراین $d(I_j, I_i) = 2$.

برای هر بازه I_j در $\{ I_j \in S \mid a_j < a_i < b_j \}$، با توجه به اینکه $a_j < a_i < b_j$ لذا $I_i \sim I_j$ و بنابراین،

$$d(I_i, I_j) = 1$$

با توجه به روابط بدست آمده در بالا ،

$$F_{aL}(a_i) = \sum_{I_j \in S \mid a_j < a_i} w(I_j) d(I_j, I_i)$$

$$= \sum_{I_j \in S \mid a_j < a_{LSUC(I_i)}} w(I_j) d(I_j, I_i) + \sum_{I_j \in S \mid a_{LSUC(I_i)} < a_j < a_i} w(I_j) d(I_j, I_i) - \sum_{I_j \in S \mid a_j < a_i < b_j} w(I_j) d(I_j, I_i)$$

$$= \sum_{I_j \in S \mid a_j < a_{LSUC(I_i)}} w(I_j) \big(1 + d(LSUC(I_i), I_j)\big) + 2 \sum_{I_j \in S \mid a_{LSUC(I_i)} < a_j < a_i} w(I_j) - \sum_{I_j \in S \mid a_j < a_i < b_j} w(I_j)$$

$$= F_{aL}(a_{LSUC(I_i)}) + W_{aL}(a_{LSUC(I_i)}) + 2\left[\sum_{I_j \in S \mid a_j < a_i} w(I_j) - \sum_{I_j \in S \mid a_j < a_{LSUC(I_i)}} w(I_j)\right] - \left[\sum_{I_j \in S \mid a_j < a_i} w(I_j) - \sum_{I_j \in S \mid b_j < a_i} w(I_j)\right]$$

$$= F_{aL}(a_{LSUC(I_i)}) + W_{aL}(a_{LSUC(I_i)}) + 2W_{aL}(a_i) - 2W_{aL}(a_{LSUC(I_i)}) - W_{aL}(a_i) + W_{bL}(a_i)$$

$$= F_{aL}(a_{LSUC(I_i)}) + W_{aL}(a_i) + W_{bL}(a_i) - W_{aL}(a_{LSUC(I_i)}).$$

اثبات قسمت چهارم مشابه اثبات سه قسمت قبل می باشد. □

لم ۴. ۶ این مطلب را می رساند که مقادیر $F_{aR}(q)$ و $F_{aL}(q)$ می تواند در زمان ثابتی محاسبه شود. برای محاسبه ی مقدار تابع هدف $F(I_i)$ با توجه به فرمول های لم ۴. ۶ مقادیر $F_{aR}(a_i)$ و $F_{aL}(a_i)$ را برای نقطه پایانی چپ a_i به دست می آوریم که در زمان ثابتی محاسبه می شود. از آنجایی که n نقطه پایانی چپ داریم، $\{F(I_i)\}_{i=1,\dots,n}$ در زمان $O(n)$ محاسبه می شود. بنابراین می توانیم ماکسیمم را در زمان $O(n)$ به دست آوریم. لذا قضیه ی زیر را داریم،

قضیه ۴. ۷. [۱۷] : مسأله ۱- ماکسین روی گراف های بازه ای که نقاط پایانی درجه بندی شده اند می تواند در زمان $O(n)$ حل شود.

مثال ۴. ۲. [۱۷] . با ذکر یک مثال ازشکل (۴. ۱۰) فرمول های تکراری مجموع وزنها و فرمول های لم ۴. ۶ را روشن تر می کنیم. در اینجا برای یافتن جواب بهینه برای مسأله ۱-ماکسین در شکل (۴. ۱۰) مطابق فرمول های تکراری مجموع وزنها، ابتدا مقادیر $W_{aL}(q)$، $W_{bL}(q)$، $W_{aR}(q)$ و $W_{bR}(q)$ را برای تمامی نقاط پایانی بازه ها به دست می آوریم. سپس با توجه به فرمول های لم ۴. ۶ مقادیر $F_{aL}(q)$، $F_{bL}(q)$، $F_{aR}(q)$ و $F_{bR}(q)$ را محاسبه می کنیم. در آخر با توجه به مقادیر به

دست آمده، $F(I_i)$ را برای هر بازه I_i را به دست آورده و از میان آنها مقدار ماکسیمم را انتخاب می کنیم که جواب بهینه است.

ابتدا مقادیر W را برای نقطه p_1 که متناظر نقطه پایانی a_2 است به دست می آوریم ،

$$W_{aL}(a_2) = \sum_{I_j \in S | a_j < a_2} w(I_j) = 0$$

زیرا $\{ I_j \in S | a_j < a_2 \} = \phi$

$$W_{bL}(a_2) = \sum_{I_j \in S | b_j < a_2} w(I_j) = 0$$

زیرا $\{ I_j \in S | b_j < a_2 \} = \phi$

$$\begin{aligned} W_{aR}(a_2) &= \sum_{I_j \in S | a_j > a_2} w(I_j) = w(I_1) + w(I_3) + w(I_4) + w(I_5) + w(I_6) + w(I_7) \\ &+ w(I_8) + w(I_9) + w(I_{10}) = 1+3+4+4+5+2+2+2+3 = 26 \end{aligned}$$

$$W_{bR}(a_2) = \sum_{I_j \in S | b_j > a_2} w(I_j) = \sum_{j=1}^{10} w(I_j) = 29$$

حال مقادیر W را برای نقطه پایانی دیگر به عنوان مثال a_{10} به دست می آوریم،

$$W_{aL}(a_{10}) = \sum_{I_j \in S | a_j < a_{10}} w(I_j) = \sum_{j=1}^{9} w(I_j) = 27$$

$$\begin{aligned} W_{bL}(a_{10}) &= \sum_{I_j \in S | b_j < a_{10}} w(I_j) = w(I_1) + w(I_2) + w(I_4) + w(I_5) + w(I_6) + w(I_3) \\ &= 1+3+3+4+2+4 = 17 \end{aligned}$$

$$W_{aR}(a_{10})=\sum_{I_j\in S|a_j>a_{10}} w(I_j)=0$$

زیرا $\{I_j\in S|a_j>a_{10}\}=\phi$

$$W_{bR}(a_{10})=\sum_{I_j\in S|b_j>a_{10}} w(I_j)=w(I_7)+w(I_8)+w(I_9)+w(I_{10})=12$$

به همین ترتیب برای بقیه نقاط پایانی بازه ها به دست می آوریم که مقادیر آن در جدول(۴. ۱) آمده است.

جدول (۴. ۱) مجموع وزنها برای هر نقطه $q\in L$

نقطه	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
نقطه پایانی	a_2	a_4	a_1	a_3	b_1	b_2	a_5	b_3	a_9	b_4
W_{aL}	0	3	7	8	11	11	11	15	15	20
W_{bL}	0	0	0	0	0	1	4	4	7	7
W_{aR}	26	22	21	18	18	18	14	14	9	9
W_{bR}	29	29	29	29	28	25	25	22	22	18
نقطه	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}	p_{18}	p_{19}	p_{20}
نقطه پایانی	a_6	b_5	a_8	a_7	b_6	a_{10}	b_7	b_8	b_9	b_{10}
W_{aL}	20	22	22	25	27	27	29	29	29	29
W_{bL}	11	11	15	15	15	17	17	19	22	27
W_{aR}	7	7	4	2	2	0	0	0	0	0
W_{bR}	18	14	14	14	12	12	10	7	2	0

حال مقادیر F_{aL}, F_{aR} را برای نقاط پایانی چپ و نقاط پایانی راست بازه ها به دست می آوریم، برای نمونه این مقادیر را برای b_1 به دست می آوریم،

$$F_{aR}(b_1) = F_{aR}\left(b_{RSUC(I_1)}\right) + 2W_{aR}(b_1) - W_{aR}\left(b_{RSUC(I_1)}\right)$$

$$= F_{aR}(b_4) + 2\times 18 - W_{aR}(b_4)$$

$$= F_{aR}(b_4) + 36 - 9$$

$$= F_{aR}\left(b_{RSUC(I_4)}\right) + 2W_{aR}(b_4) - W_{aR}\left(b_{RSUC(I_4)}\right) + 27$$

$$= F_{aR}(b_9) + 2\times 9 + 0 + 27$$

$$= F_{aR}\left(b_{RSUC(I_9)}\right) + 2W_{aR}(b_9) - W_{aR}\left(b_{RSUC(I_9)}\right) + 45$$

$$= F_{aR}(b_{10}) + 0 - 0 + 45 = 45$$

$$F_{aL}(b_1) = F_{aL}(a_1) + W_{aL}(b_1) - W_{aL}(a_1) - w(I_1)$$

$$= F_{aL}(a_{LSUC(I_1)}) + W_{aL}(a_1) + W_{bL}(a_1) - W_{aL}(a_{LSUC(I_1)}) + 3$$

$$= F_{aL}(a_2) + 7 + 0 - 0 + 3 = 10$$

و به همین ترتیب برای بقیه نقاط به دست می آوریم که مقادیر آن در جدول ۴. ۲ آمده است.

جدول ۴. ۲ مقادیر توابع F_{aL} ,F_{aR} و مقدار تابع F(I)

نقطه پایانی راستی	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
F_{aR}	45	45	35	18	14	4	0	0	0	0
F_{aL}	10	8	12	16	22	40	49	48	31	51
نقطه پایانی چپ	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
F_{aR}	48	53	39	31	21	9	2	4	9	0
F_{aL}	7	0	8	3	15	35	47	44	22	51
بازه	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
$F(I)$	55	53	47	34	36	44	49	48	31	51

با توجه به مقادیر به دست آمده در جدول ۴. ۲، جواب بهینه بازه I_1 است.

۴. ۶ مسأله p- ماکسین $(p \geq 2)$ روی گراف های بازه ای

در این بخش به بررسی مسأله p-ماکسین $(p \geq 2)$ می پردازیم، یعنی p میانه ی ماکسیمم را در بازه های کمینه طوری قرار دهیم که مجموع فاصله وزنی از همه بازه ها تا دورترین میانه ی ماکسیمم، ماکسیمم شود.

با در نظر گرفتن مسأله ۲- ماکسین شروع می کنیم، هدف یافتن دو بازه ی کمینه I_X و I_Y به طوری که تابع زیر ماکسیمم شود،

$$F(I_X,I_Y)=\sum w(I_j)max\{d(I_j,I_X),d(I_j,I_Y)\}.$$

برای محاسبه $F(I_X,I_Y)$ برای دو بازه ی کمینه ی I_X و I_Y داده شده، کافی است مجموعه بازه ای S را به دو زیر مجموعه مجزا S_X و S_Y تقسیم کنیم به طوری که برای هر بازه $I_j \in S_X$ ، $d(I_j,I_X) \geq d(I_j,I_Y)$ و برای هر بازه ی $I_j \in S_Y$ ، $d(I_j,I_Y) \geq d(I_j,I_X)$ باشد.

دو کاندیدای میانه ماکسیمم I_X و I_Y کمینه هستند لذا نمی توانند مشمول یکدیگر باشند. بودن کاستن از کلیت مسأله، فرض می کنیم $a_x < a_y$ و $b_x < b_y$. روش افراز زیر را با استفاده از دو مورد مشخص شده که به فاصله بین I_X و I_Y بستگی دارد مطرح می کنیم. توجه کنید $d=d(I_X,I_Y)$. از این روش به منظور افراز مجموعه ی S به دو زیرمجموعه مجزای S_X و S_Y استفاده می کنیم.

۴. ۶. ۱ روش افراز

۱- اگر d زوج باشد، آنگاه p_{XY} را نقطه پایانی چپ $LSUC\left(I_Y,\frac{d}{2}-1\right)$ قرار دهید و $S_X=\{I_j \in S \mid b_j > p_{XY}\}$ و $S_Y=\{I_j \in S \mid b_j < p_{XY}\}$.

۲- اگر d فرد باشد، آنگاه p_{XY} را نقطه پایانی راست $RSUC\left(I_X,\frac{d-1}{2}\right)$ قرار دهید و $S_X=\{I_j \in S \mid b_j > p_{XY}\}$ و $S_Y=\{I_j \in S \mid b_j \leq p_{XY}\}$.

واضح است که $S_X \cap S_Y=\phi$ و $S_X \cup S_Y=S$. در اینجا p_{XY} را نقطه منقسم از بازه I_X و I_y می نامیم.

واضح است که $p_{XY}=\begin{cases} b_x & d=1 \\ a_y & d=2 \end{cases}$ و برای $d \geq 3$ $b_x < p_{XY} < a_y$.

لم های زیر در [۲۱] بیان شده است.

لم ۴. ۸. [۱۷] : فرض کنید بازه های I_X و I_Y نا مجاور باشند با $b_x < b_y$ و p_{XY} نقطه مقسم از I_Y, I_X باشد، برای هر بازه ی I_j با $a_x < a_j$ و $b_j < b_y$ (به عبارت دیگر I_j مشمول در $\left[a_x, b_y\right]$ است)، اگر $b_j < p_{XY}$ آنگاه $d(I_j, I_X) \le d(I_j, I_Y)$ و اگر $b_j > p_{XY}$، آنگاه $d(I_j, I_Y) \le d(I_j, I_X)$.

اکنون باید نشان دهیم که S_X و S_Y در روش افراز دقیقاً زیر مجموعه هایی هستند که ما می خواهیم داشته باشیم.

لم ۴. ۹. [۱۷] : اگر S_Y, S_X زیر مجموعه هایی از S که در روش افراز توضیح داده شده، باشند، آنگاه $d(I_j, I_X) \ge d(I_j, I_Y)$ برای هر بازه $I_j \in S_X$ و برای هر بازه $I_j \in S_Y$، $d(I_j, I_Y) \ge d(I_j, I_X)$.

اثبات

ابتدا به وسیله دو حالت درنظر گرفته شده، اثبات می کنیم برای هربازه $I_j \in S$ با $b_j > p_{XY}$، $d(I_j, I_X) \ge d(I_j, I_Y)$.

حالت اول. $b_j > b_y > p_{XY}$. اگر $a_j < b_y$. آنگاه $I_j \sim I_Y$. بنابراین $d(I_j, I_Y) = 1$ و لذا $d(I_j, I_X) \ge 1 = d(I_j, I_Y)$. در غیر اینصورت اگر $a_j > b_y$ آنگاه $a_j > a_y$، ازآنجایی که قبلاً فرض کردیم $a_x < a_y$ و $b_x < b_y$ لذا $a_j > a_x$. حال چون $a_j > a_y$، $a_j > a_x$ و $b_x < b_y$، شرایط قسمت ۲ لم (۴. ۴) برقرار است و لذا با توجه به آن لم $d(I_j, I_X) \ge d(I_j, I_Y)$ است.

حالت دوم. $p_{XY} < b_j < b_y$. آنگاه فقط هنگامی که $d=1$ و $d=2$ است نقطه پایانی چپ I_j یعنی a_j می تواند از a_x کمتر باشد. که در اینصورت $I_j \sim I_X$ و $I_j \sim I_Y$ ، به عبارت دیگر $d(I_j, I_X) = d(I_j, I_Y) = 1$.

اگر $d \geq 3$، آنگاه $a_j > a_X$. زیرا در غیر اینصورت با $d = d(I_X, I_Y)$ متناقض است. چون $a_j > a_x$ و $b_j < b_y$ لذا بازه I_j مشمول در $[a_x, b_y]$ است.

برای هر بازه I_j مشمول در $[a_x, b_y]$، اگر $d=1$ ، آنگاه $I_j \sim I_y$. بنابراین $d(I_j, I_X) \geq 1 = d(I_j, I_Y)$. در غیر اینصورت با استفاده از لم ۴. ۸ چون $b_j > p_{XY}$ بنابراین $d(I_j, I_Y) \leq d(I_j, I_X)$.

حال در قسمت دوم اثبات می کنیم که برای $I_j \in S_Y$ رابطه $d(I_j, I_Y) \geq d(I_j, I_X)$ برقرار است.

اگر d فرد باشد و $b_j = p_{XY}$، آنگاه $RSUC\left(I_X, \frac{d-1}{2}\right) = I_j$. بنابراین $d(I_j, I_X) = \frac{d-1}{2}$. ازطرفی چون $b_j = p_{XY}$ و $d(I_j, I_X) = \frac{d-1}{2}$ حداقل فاصله بین بازه I_Y و I_X برابر $\frac{d+1}{2}$ است و لذا

$d(I_j, I_Y) > \frac{d-1}{2} = d(I_j, I_X)$.

حال به بررسی بازه هایی می پردازیم که در آن $b_j < p_{XY} < a_y < b_y$. اگر $b_j < p_{XY} < b_y$، دو حالت اتفاق می افتد،

حالت اول. $a_X < a_j$ آنگاه چون $b_j < b_y$ ، I_j مشمول در $[a_x, b_y]$ می باشد. اگر $d=1$، آنگاه $p_{XY} = b_x$ و لذا $b_j < p_{XY} = b_x$ و طبق فرض $a_x < a_j$ بنابراین $I_j \sim I_X$، بنابراین $d(I_j, I_Y) \geq 1 = d(I_j, I_X)$. اگر $d > 1$ داریم $I_X \nsim I_Y$. بنابر لم ۴. ۸ چون I_j مشمول در $[a_x, b_y]$ است و $b_j < p_{XY}$ بنابراین $d(I_j, I_X) \leq d(I_j, I_Y)$.

حالت دوم. $a_j < a_x$، دو حالت را بررسی می کنیم. اگر $b_j < b_x$، آنگاه بنابر قسمت (۱) از لم ۴. ۴ چون $b_j < b_x$ و $b_j < b_y$ و از طرفی طبق فرضی که در بخش ۴. ۶ کردیم $a_x < a_y$، بنابراین $d(I_j,I_Y) \ge d(I_j,I_X)$. اگر $b_x < b_j < p_{XY}$ آنگاه چون $a_j < a_x$، بازه I_X مشمول در I_j است و لذا $d(I_j,I_X)=1$، بنابراین $d(I_j,I_Y) \ge 1 = d(I_j,I_X)$. □

مجموعه S' را یک زیر مجموعه از مجموعه بازه ای S و بازه $I \in S$ را در نظر بگیرید. برای راحتی قرار دهید $F_{S'}(I) = \sum_{I_j \in S'} w(I_j) d(I_j,I)$. بنابراین بنابرلم ۴. ۹ داریم،

$$F(I_X,I_Y) = \sum_{j=1}^{n} w(I_j) max\{ d(I_j,I_X), d(I_j,I_Y)\}$$

$$= \sum_{I_j \in S_X} w(I_j) d(I_j,I_X) + \sum_{I_j \in S_Y} w(I_j) d(I_j,I_Y)$$

$$= F_{S_X}(I_X) + F_{S_Y}(I_Y).$$

دو بازه ی M_1 و M_2 با $b_{M_1} = min\{ b_1,\ldots,b_n\} = b_1$ و $a_{M_2} = max\{ a_1,\ldots,a_n\}$ را در نظر بگیرید.

لم ۴. ۱۰. [۱۷]: دو بازه ی M_1 و M_2 هر دو کمینه هستند.

اثبات

به برهان های خلف فرض کنیم M_1 کمینه نباشد یعنی M_1 شامل بازه ی دیگر مانند I با نقطه پایانی چپ a و نقطه پایانی راست b باشد. چون بازه I مشمول در بازه ی M_1 است، پس $a_{M_1} < a < b < b_{M_1}$. که این متناقض است با اینکه $b_{M_1} = min\{ b_1,\ldots,b_n\}$ بنابراین برهان خلف باطل

و M_1 کمینه است. اثبات برای M_2 مشابه M_1 است. به برهان خلف فرض می کنیم M_2 کمینه نباشد یعنی M_2 شامل بازه ی دیگر مانند I با نقطه پایانی چپ a و نقطه پایانی راست b باشد. بنابراین $a_{M_2} < a < b < b_{M_2}$ که این متناقض با $a_{M_2} = max\{ a_1,\ldots,a_n \}$ است. پس فرض خلف باطل و M_2 نیز کمینه است. $\square$

لم ۴. ۱۱. [۱۷] : فاصله بین دو بازه ی M_2, M_1، ماکسیمم فاصله روی گراف باز ه ای G است.

اثبات

برای اثبات کافی است نشان دهیم که برای هر دوبازه I_Y, I_X در S داریم، $d(M_1,M_2) \geq d(I_X,I_Y)$.

اگر $I_X \sim I_Y$ آنگاه $d(M_1,M_2) \geq 1 = d(I_X,I_Y)$. در ادامه به بررسی اینکه $I_X \nsim I_Y$ می پردازیم. نشان می دهیم که $d(I_X,M_2) \geq d(I_X,I_Y)$ برای هر دو بازه I_X و I_Y در S.

طبق تعریف M_2 ، $a_{M_2} > a_y$ حال اگر $b_y > b_{M_2}$ آنگاه M_2 مشمول در I_Y است. با استفاده از نتیجه ۴. ۵ برای هر بازه ی دیگری مانند I_X داریم، $d(I_X,M_2) \geq d(I_X,I_Y)$. اگر $b_y < b_{M_2}$، طبق فرضی که قبلاً کردیم، داریم $b_x < b_y < b_{M_2}$ و با توجه به تعریف M_2 داریم $a_{M_2} > a_y$ آنگاه با استفاده از قسمت (۱) از لم ۴. ۴ چون $b_x < b_y$ و $b_x < b_{M_2}$ و $a_{M_2} > a_y$ داریم، $d(I_X,M_2) \geq d(I_X,I_Y)$.

از آنجایی که طبق تعریف بازه های M_1 و M_2 داریم، $a_{M_2} > a_{M_1}$، $a_{M_2} > a_x$ و $b_x > b_{M_1}$، با استفاده از قسمت (۲) از لم ۴. ۴ داریم، $d(M_2,M_1) \geq d(M_2,I_X)$، بنابراین اثبات کامل می شود. $\square$

طبق لم ۴. ۱۱ واضح است اگر $d(M_1,M_2)=1$ ، آنگاه برای دو بازه I_Y,I_X در S، $d(I_X,I_Y)=1$. لذا هر جفت از بازه ها مقدار تابع هدف یکسانی به صورت $\sum_{I_j\in S} w(I_j)$ را دارا می باشند. بنابراین ما فرض می کنیم که $d(M_1,M_2)>1$.

قضیه ۴. ۱۲. [۱۷] : $\{M_1,M_2\}$ با نقطه منقسم $P_{M_1M_2}$ یک جواب بهینه برای مسأله ۲-ماکسین روی گراف بازه ای G است.

اثبات

برای نشان دادن درستی قضیه، کافی است اثبات کنیم که برای هر دو بازه I_Y,I_X در S، $F(M_1,M_2)\geq F(I_X,I_Y)$. بدون از دست دادن کلیت مسأله فرض می کنیم که $b_x<b_y$. ابتدا مجموعه S_{M_2},S_{M_1} را که در آن $p_{M_1M_2}$ نقطه منقسم M_1 و M_2 می باشد را به صورت زیر تعریف می کنیم،

$$S_{M_1}=\{I_j\in S \mid b_j>p_{M_1M_2}\},\quad S_{M_2}=\{I_j\in S \mid b_j\leq p_{M_1M_2}\}$$

مشابه اثبات لم ۴. ۱۱، می توانیم به دست آوریم،

$$d(I_X,I_Y)\leq d(I_X,M_2)\quad;\quad d(I_X,I_Y)\leq d(I_Y,M_1). \qquad (۴. ۱)$$

برای هر بازه $I_X\in S_{M_2}$،

$$F_{S_{M_1}}(I_X)\leq F_{S_{M_1}}(M_1), \qquad (۴. ۲)$$

برای هر بازه $I_Y\in S_{M_1}$،

$$F_{S_{M_2}}(I_Y) \le F_{S_{M_2}}(M_2), \tag{۴. ۳}$$

و برای هر جفت از بازه های $I_X, I_Y \in S_{M_1}$،

$$d(I_X, I_Y) \le d(I_X, M_2) \le d(I_X, M_1); \tag{۴. ۴}$$

$$d(I_X, I_Y) \le d(I_Y, M_1). \tag{۴. ۵}$$

مشابهاً برای دوبازه $I_X, I_Y \in S_{M_2}$،

$$d(I_X, I_Y) \le d(I_X, M_2); \tag{۴. ۶}$$

$$d(I_X, I_Y) \le d(I_Y, M_1) \le d(I_Y, M_2). \tag{۴. ۷}$$

از آنجایی که p_{XY} نقطه منقسم بازه I_Y, I_X است، دو حالت ممکن زیر را در نظر می گیریم،

حالت اول، $I_X, I_Y \in S_{M_2}$. آنگاه با توجه به تعریف S_{M_2} و روش افراز داریم، $p_{XY} < p_{M_1M_2}$, $S_X \supset S_{M_1}$ و $S_Y \subset S_{M_2}$.

از آنجایی که $S_Y \subset S_{M_2}$ و $S_X \supset S_{M_1}$ و $S_{M_1} \cup S_{M_2} = S_X \cup S_Y = S$، داریم،

$$S_X - S_{M_1} = S_{M_2} - S_Y \subset S_{M_2}$$

چون فرض کردیم $I_X, I_Y \in S_{M_2}$ طبق رابطه (۴. ۶) و (۴. ۷) داریم،

(توجه داشته باشید که هر $I_j \in S_Y$، چون $S_Y \subset S_{M_2}$، $I_j \in S_{M_2}$ نیز می باشد.)

$$F_{S_Y}(I_Y) = \sum_{I_j \in S_Y} w(I_j) d(I_j, I_Y) \le \sum_{I_j \in S_Y} w(I_j) d(I_j, M_2) = F_{S_Y}(M_2)$$

$$F_{S_Y}(I_Y) \le F_{S_Y}(M_2) \quad . \tag{۴. ۸}$$

از طرفی چون $S_X - S_{M_1} = S_{M_2} - S_Y$،

$$F_{S_x - S_{M_1}}(I_X) = \sum_{I_j \in S_X - S_{M_1}} w(I_j) d(I_j, I_X) = \sum_{I_j \in S_{M_2} - S_Y} w(I_j) d(I_j, I_X) = F_{S_{M_2} - S_Y}(I_X).$$

از آنجایی که $S_{M_2} - S_Y \subset S_{M_2}$، رابطه زیر را با توجه به رابطه (۴. ۶) و (۴. ۷) می توانیم به دست آوریم،

$$F_{S_{M_2} - S_Y}(I_X) = \sum_{I_j \in S_{M_2} - S_Y} w(I_j) d(I_j, I_X) \le \sum_{I_j \in S_{M_2} - S_Y} w(I_j) d(I_j, M_2) = F_{S_{M_2} - S_Y}(M_2).$$

$$\Rightarrow F_{S_X - S_{M_1}}(I_X) = F_{S_{M_2} - S_Y}(I_X) \le F_{S_X - S_{M_1}}(M_2) = F_{S_{M_2} - S_Y}(M_2). \tag{۴. ۹}$$

با استفاده از روابط (۴. ۲) و (۴. ۸) و (۴. ۹) می توانیم به دست آوریم،

$$\begin{aligned} F(I_X, I_Y) &= F_{S_X}(I_X) + F_{S_Y}(I_Y) \\ &= F_{S_{M_1}}(I_X) + F_{S_X - S_{M_1}}(I_X) + F_{S_Y}(I_Y) \\ &\le F_{S_{M_1}}(M_1) + F_{S_{M_2} - S_Y}(M_2) + F_{S_Y}(M_2) \\ &= F_{S_{M_1}}(M_1) + F_{S_{M_2}}(M_2) \\ &= F(M_1, M_2). \end{aligned}$$

حالت دوم. $I_X \in S_{M_2}$ و $I_Y \in S_{M_1}$. اگر $p_{XY} = p_{M_1 M_2}$، آنگاه $S_X = S_{M_1}$ و $S_Y = S_{M_2}$.

با استفاده از روابط (۴. ۲) و (۴. ۳) داریم،

$$F(I_X,I_Y)=F_{S_X}(I_X)+F_{S_Y}(I_Y)$$

$$=F_{S_{M_1}}(I_X)+F_{S_{M_2}}(I_Y)$$

$$\le F_{S_{M_1}}(M_1)+F_{S_{M_2}}(M_2)$$

$$=F(M_1,M_2).$$

اگر $p_{XY}\neq p_{M_1M_2}$ ، بدون کاستن از کلیات مسأله فرض می کنیم که $p_{XY}>p_{M_1M_2}$. در این صورت $S_X\subset S_{M_1}$ و $S_Y\supset S_{M_2}$ ، از آنجایی که $S_X\subset S_{M_1}$ و $I_X\in S_{M_2}$ رابطه (۴. ۲) نشان می دهد که $F_{S_X}(I_X)\le F_{S_X}(M_1)$. برای زیر مجموعه ی $S_Y-S_{M_2}=S_{M_1}-S_X$ با استفاده از روابط (۴. ۴) و (۴. ۵) داریم،

$$\Rightarrow F_{S_Y-S_{M_2}}(I_Y)=F_{S_{M_1}-S_X}(I_Y)\le F_{S_Y-S_{M_2}}(M_1)=F_{S_{M_1}-S_X}(M_1).$$

با استفاده از دو رابطه به دست آمده در بالا و رابطه (۴. ۳) داریم،

$$F(I_X,I_Y)=F_{S_X}(I_X)+F_{S_Y}(I_Y)$$

$$=F_{S_X}(I_X)+F_{S_Y-S_{M_2}}(I_Y)+F_{S_{M_2}}(I_Y)$$

$$\le F_{S_X}(M_1)+F_{S_{M_1}-S_X}(M_1)+F_{S_{M_2}}(M_2)$$

$$=F_{S_{M_1}}(M_1)+F_{S_{M_2}}(M_2)$$

$$\Box= F(M_1,M_2).$$

از اثبات قضیه ۴. ۱۲ واضح است که میانه های ۲-ماکسین بهینه ی $\{M_1,M_2\}$، به وزن های مثبت بازه ها بستگی ندارد. بنابراین نتیجه زیر را داریم،

نتیجه ۴. ۱۳. [۱۷] : همه وزنهای مثبت بازه ها در گراف بازه ای، میانه های ۲- ماکسین بهینه ی یکسانی را به ما می دهد.

قضیه ۴. ۱۴. [۱۷] : فرض کنید مجموعه $\bar{S}$ یک زیر مجموعه از S شامل p بازه مجزا باشد. اگر $\bar{S}$ شامل M_1 و M_2 باشد، آنگاه $\bar{S}$ یک جواب بهینه از مسأله p-ماکسین $(p \geq 2)$ است.

اثبات

بگیرید $\bar{S}=\{M_1=I_{i_1},I_{i_2},\ldots,I_{i_p}=M_2\}$ یک جواب بهینه برای مسأله p ـماکسین باشد. بنابراین،

$$d(M_1,M_2)=\max_{1\leq k,\,1\leq p}\{d(I_{i_k},I_{i_l})\}.$$

همانند اثبات قضیه ۴. ۱۲، برای هربازه $I_j \in S_{M_1}$ ، $I_{i_k} \in S_{M_1}$ یا $I_{i_k} \in S_{M_2}$، $d(I_j,I_{i_k}) \leq d(I_j,M_1)$. بنابراین همه بازه ها در S_{M_1} به M_1 اختصاص داده می شوند (یعنی دورترین فاصله به بازه ها در S_{M_1}، بازه M_1 است) و مشابها همه بازه ها در S_{M_2} به M_2 اختصاص داده می شوند. از این رو

$$F(\bar{S})=F_{S_{M_1}}(M_1)+F_{S_{M_2}}(M_2)=F(M_1,M_2).$$

بنابراین برای هر زیر مجموعه $\bar{S}'$ شامل p بازه مجزا، داریم،

$$F(\bar{S})=F(M_1,M_2)\geq F(\bar{S}'). \quad \square$$

از آنجایی که M_1 و M_2 با استفاده از لم ۴. ۱۰ کمینه هستند و می توانند با محاسبه به ترتیب $\min_{1\le i\le n}\{b_i\}$ و $\max_{1\le i\le n}\{a_i\}$ در زمان $O(n)$ به دست آیند، قضیه زیرا را داریم،

قضیه ۴. ۱۵. [۱۷] : مسأله p-ماکسین روی گراف های بازه در زمان خطی قابل حل است.

مثال ۴. ۳.[۱۷] . به عنوان مثال دو میانه های ماکسیمم را در شکل (۴. ۱۰) به دست می آوریم. برای این کار کافی است دو بازه ی کمینه ی M_1 و M_2 را بیابیم. مطابق شکل (۴. ۱۰) $Min\{b_1,\dots,b_{10}\}=b_1$ لذا $M_1=I_1$ و $Max\{a_1,\dots,a_{10}\}=a_{10}$، لذا $M_2=I_{10}$، دو میانه ماکسیمم با نقطه منقسم $P_{M_1M_2}=b_4$ هستند. مقدار تابع هدف برابر است با $F(M_1,M_2)=74$.

روش به دست آوردن نقطه منقسم $P_{M_1M_2}$ و مقدار تابع هدف $F(M_1,M_2)$ به صورت زیر است،

$$d(M_1,M_2)=\{I_1,RSUC(I_1,1),RSUC(I_1,2),I_{10}\}=\{I_1,I_4,I_9,I_{10}\}=3$$

چون d فرد است داریم،

$$RSUC(M_1,1)=I_4$$

لذا $P_{M_1M_2}=b_4$

$$S_{M_1}=\{I_j\in S \mid b_j>b_4\}=\{I_5,I_6,I_7,I_8,I_9,I_{10}\}$$

$$S_{M_2}=\{I_j\in S \mid b_j\le b_4\}=\{I_4,I_3,I_2,I_1\}$$

$$F(M_1,M_2)=F_{S_{M_1}}(M_1)+F_{S_{M_2}}(M_2)=\sum_{I_j\in S_{M_1}}w(I_j)d(I_j,I_1)+\sum_{I_j\in S_{M_2}}w(I_j)d(I_j,I_{10})=74$$

واژه نامه انگلیسی به فارسی

1-median Problem	مسأله ۱-میانه
2-maxian Problem	مسأله ۱-ماکسین
Absolute p-median	*p*- میانه محض
Block Graphs	گراف های بلوکی
Branch-Tree	شاخه درخت
Center	مرکز
Classical p-median Problem	مسأله کلاسیک *p*-میانه
Componet	مؤلفه
Divided Point	نقطه منقسم
Elementary Branch	شاخه اولیه
Extention	بسط
Generation	نسل
Interval Graphs	گراف های بازه ای
Left Sub Tree	زیر درخت چپ
Location Problem	مسأله مکانیابی
Maxian Median	میانه ماکزیمم
Median	میانه
Merging	ادغام
Minimal	کمینه
Minimax	مینیماکس

کمترین مجموع	*Minisum*
مسأله چند وسیله ای مینیماکس	*Multi Facility Minimax Problem*
مسأله چند وسیله ای با کمترین مجموع	*Multi Facility Minisum Problem*
NP-سخت	*NP-hard*
مسأله مکانیابی تسهیلات مضر	*Obnoxious Facility Location Problem*
مسأله *p*-مرکز	*P-center Problem*
مسأله *p*-ماکسین	*P-maxian Problem*
مسأله *p*-میانه	*P-median Problem*
پیمایش برگ به ریشه	*Post-order*
زیر درخت راست	*Right Subtree*
مسأله تک وسیله ای مینیماکس	*Single Facility Minimax Problem*
مسأله تک وسیله ای با کمترین مجموع	*Single Facility Minisum Problem*
p- میانه رأسی	*Vertex P-median*

مراجع

[1] Francis R., McGinnis Jr.L.F and white J.A., Facility Layout and location : An Analytical Approach, Prentice Hall, 1992.

[2] Hakimi S.L., Optimum location of switching centers and the absolute centers and medians of a graph, Operations Research 12(1964) 450-459.

[3] Balinski M.L., Integer programming: Methods, uses, computation, Management Sci. 12 (1965) 253-313.

[4] Kuehn A.A. and Hamburger M.J., A heuristic program for location ware houses, Management Sci. 9(1963) 643-666.

[5] Manne A., Plant location under economics of scale-decentralization and computation, Management Sci. 11(1964) 213-235.

[6] O. Kariv, S.L. Hakimi, An algorithmic approach to network location problems. Part II p-medians, SIAM J. Appl. Math. 37 (1979) 539–560.

[7] A. Tamir, An O (pn^2) algorithm for the p-median and related problems on tree graphs, Oper. Res. Lett. 19 (1996)59–64.

[8] Burkard R.E. and Dollani H., Center problems with pos/neg weights on trees, SFB Report No.215, Institute of math, B, Technical University Graz, February, 2001.

[9] Burkard R.E. and Dollani H., Robust location problems with pos/neg weights on trees Networks, 38 (2001) 102-113.

[10] S. Irani, V. Leung, Scheduling with conflicts on bipartite interval graphs, J. Scheduling 6 (2003) 287–307.

[11] R.E. Burkard, E. Çela, H. Dollani, 2-median in trees with pos/neg weights, Discrete Appl. Math. 105 (2000) 51–71.

[12] S.S. Ting, A linear-time algorithm for maxisum facility location on tree networks, Transp. Sci. 18 (1984) 76–84.

[13] B. Zelinka, Medians and peripherians of trees, Arch. Math. 4 (1968) 87–95.

[14] R.L. Church, R.S. Garfinkel, Locating an obnoxious facility on a network, Transp. Sci. 12 (1978) 107–118.

[15] R.E. Burkard, J. Krarup, A linear algorithm for the pos/neg weighted median problem on a cactus, Computing 60 (1998) 193–215.

[16] L.Y. Kang, Y.K. Cheng, The p-maxian problem on block graphs, J. Comb. Optim. 20 (2010) 131–141.

[17] L.Y. Kang, Y.K. Cheng, The p-maxain problem on interval graph, Discrete Applied Mathematics. 158 (2010) 1986-1993.

[18] R.E. Burkard, J. Fathali, H.T. Kakhki, The p-maxian problem on a tree, Oper. Res. Lett. 35 (2007) 331–335.

[19] A. Tamir, Obnoxious facility location on graphs, SIAM Journal on Discrete Mathematics 4 (1991) 550–567.

[20] G.Y. Handler, Minimax location of a facility in an undirected tree networks, Transp. Sci. 7 (1973) 287–293.

[21] S. Bespamyatnikh, B. Bhattacharya, M. Keil, D. Kirkpatrick, M. Segal, Efficient algorithms for centers and medians in interval and circular-arc graphs, Networks 29 (2002) 144–152.

[22] M.A. Bonuccelli, Dominating sets and domination number of circular-arc graphs, Discrete Appl. Math. 12 (1985) 203–213.

[23] D. Chen, D.T. Lee, R. Sridhar, C. Sekharam, Solving the all-pair shortest path query on interval and circular-arc graphs, Networks 31 (1998) 249–258.

[24] M.C. Golumbic, Interval graphs and related topics, Discrete Math. 55 (1985) 113–121.

[25] R. Laskar, J. Pfaff, S.M. Hedetniemi, S.T. Hedetniemi, On the algorithmic complexity of total domination, SIAM J. Algebr. Discrete Methods 5 (1984) 420–425.

[26] M.S. Lin, Fast and simple algorithms to count the number of vertex covers in an interval graph, Inform. Process. Lett. 102 (2007) 143–146.

[27] M.M. Halldorsson, G. Kortsarz, H. Shachnai, Sum coloring interval and k-claw free graphs with application to scheduling dependent jobs, Algorithmica 37 (2003) 187–209.

[28] Weber A., Uber den Standort der Industrient. Tubingen, 1909; English Trans.: Theory of location of Industries, (C.J.Friedrich, ed. and trans.) Chicago University press, Chicago, Illinois, 1929.

[29] T.C.E. Cheng, L.Y. Kang, C.T. Ng, An improved algorithm for the p-center problem on interval graphs with unit lengths, Comput. Oper. Res. 34 (2007) 2215–2222.

[30] Hamacher H.W. and Nickel S., "classification of location models." Location Sci., 6(1998)229-242.

[31] Handler G.Y. and Mirchandani P.B., Location On Networks: Theory and Algorithms. MIT Press, Cambridge, 1979.

[32] Tansel B.C., Francis R.L. and Low T.J., Location on networks: Asurvey; Management Sci., 29(1983) 482-511.

[33] Krarup J. and Pruzan P.M., The simple plant location problem: survey and synthesis. EJOR, 12 (1983) 36-81.

[34] Hansen P., Labbe M., Peeters D. and J.F. Thisse J.F., Single facility location on networks. Annals of Discrete Math., 31(1987) 113-146.

[35] Current J., Daskin M. and Schilling D., 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, 1929.

[36] Owen S.H. and Daskin M.S., "Strategic facility location: A review." EJOR, 111 (1998) 423-447.

[37] Bondy J.A. and Murty U.S.R., Graph Theory , Springer Reading, DOI 10.1007/978-1-84628-970-5, (2007).

Abstract

In recent years, there has been an increasing interest in obnoxious facility location problems where one or more facilities are to be placed as far away as possible from the clients. P-maxian problem is one of obnoxious facilities location problems. This dissertation treats in p-maxian problem on tree and interval graph and presents solutions for its solving.

At first define location problems and investigate 1-maxian problem on tree. In third section consider p-maxian problem on tree and show that the leaves of a longest path provide an optimal solution for the 2-maxian problem (and the p-maxianproblem, $p>2$*) on a tree.*

In fourth section introduce interval graphs and investigate p-maxian problem on interval graphs where each interval has a positive weight. First we propose a linear time algorithm to solve the weighted 1-maxian problem. For a 2-maxian problem, we show that two intervals with the minimum right endpoint and the maximum left endpoint, respectively, are an optimal solution. It can be easily extended to the p-maxian problem ($p \geq 3$*).*

Keywords: Location theory; p-maxian; p-median problem; Obnoxious facilities; interval graph.