مجله تحقیق در عملیات در کاربردهای آن، سال سیزدهم، شماره دوم (پیاپی ۴۹)، تابستان ۹۵، ص ص ۱۱۱-۱۰۱
شاپا ۷۲۸۶-۲۲۵۱

یک الگوریتم خطی برای مساله پیداکردن هسته درخت‌های بازه‌ای وزن‌دار

سمانه متولی اشکذری[1]، **جعفر فتحعلی**[2*]
۱- دانشجوی دکترای، دانشگاه صنعتی شاهرود، دانشکده علوم ریاضی، شاهرود، ایران
۲- دانشیار، دانشگاه صنعتی شاهرود، دانشکده علوم ریاضی، شاهرود، ایران

رسید مقاله: ۱۸ آذر ۱۳۹۴
پذیرش مقاله: ۲۵ اردیبهشت ۱۳۹۵

چکیده
در این مقاله ابتدا گراف‌های بازه‌ای را تعریف و سپس مساله‌ی پیداکردن هسته روی گراف‌های بازه‌ای و درخت‌های بازه‌ای را بررسی می‌کنیم. یک هسته در یک گراف بازه‌ای، مسیری از بازه‌های متصل به هم است که مجموع فاصله‌های تمام بازه‌ها تا این مسیر کمینه شود. نشان می‌دهیم بازه‌هایی که روی هسته‌ی یک درخت قرار دارند نمی‌توانند بازه‌ای غیرماکسیمال باشند. سپس الگوریتمی با پیچیدگی زمانی $O(n)$ برای پیداکردن هسته‌ی یک درخت بازه‌ای ارایه می‌دهیم.

کلمات کلیدی: گراف بازه‌ای، هسته درخت، مکانیابی.

۱ مقدمه
در جوامع امروزی، بهینه‌سازی در زمینه‌های مختلف اهمیت فراوانی برای صاحبان سرمایه و مدیران کارخانه‌ها دارد. پرکاربردبودن این مسایل توجه بسیاری از محققان را به خود جلب کرده و تحقیقات فراوانی در این زمینه انجام شده‌است. یکی از مسایل مهم بهینه‌سازی، مساله‌ی مکانیابی است. این مساله انواع مختلفی دارد. اولین طبقه‌بندی مدل‌های مختلف مکانیابی توسط هندلر و میرچندانی [۱] ارایه شد. پس از آن طبقه‌بندی‌های دیگری از جمله توسط هاماخر و نیکل [۲] انجام شد.

در یک شبکه با n راس، مساله‌ی پیداکردن مکان p سرویس‌دهنده روی شبکه به‌طوری‌که مجموع وزنی فاصله رئوس تا نزدیک‌ترین سرویس‌دهنده مینیمم شود، مساله‌یp-میانه نامیده می‌شود. هسین و تمیر [۳] مساله p-میانه را روی یک مسیر با وزن‌های مثبت در نظر گرفتند و برای حل آن الگوریتمی خطی با پیچیدگی زمانی

* عهده‌دار مکاتبات
آدرس الکترونیکی: fathali@shahroodut.ac.ir

$O\left(pn\right)$ ارایه کردند. مایمانی [۴] و فتحعلی و همکاران [۵] مساله مکانیابی میانه را روی گراف‌های خاص بررسی کردند.

در مساله مکانیابی سرویس‌دهنده، سرویس‌دهنده می‌تواند به صورت یک مسیر در یک شبکه نیز مطرح شود. بنابراین مساله عبارت است از پیداکردن مسیری از یک شبکه که مجموع فاصله‌های وزنی رئوس شبکه تا این مسیر کم‌ترین مقدار ممکن باشد. واضح است که برخلاف مساله p-میانه که سرویس‌دهنده‌ها بر روی چند نقطه‌ی جدا از هم قرار دارند، در این مساله سرویس‌دهنده یک وسیله‌ی بهم‌پیوسته است که بر روی یک مسیر قرار می‌گیرد. چنین مسیری را هسته‌ی شبکه و مساله را مساله‌ی هسته می‌نامند. مساله‌ی پیداکردن هسته روی شبکه یا درخت به طور جداگانه قابل بررسی و دارای خواص متفاوتی است. پیداکردن هسته یک درخت توسط مورگان و اسلیتر [۶] و بکر [۷] بررسی و الگوریتم‌هایی خطیبرای حل آن پیشنهاد شده‌است. هم‌چنین حکیمی و همکاران [۸] در سال **۱۹۹۳** نشانداد که پیداکردن مسیر بر روی شبکه یک مساله‌ی NP-سخت است. زعفرانیه و فتحعلی [۹] این مساله را روی شبکه‌های درختی با وزن مثبت و منفی مورد بررسی قرار دادند. سرویس‌دهنده‌ها در مسایل مکانیابی را می‌توان به صورت درخت نیز در نظرگرفت. این مساله توسط تمیر و همکاران [۱۰] در سال **۲۰۰۵** بررسی شد.

مساله‌ی هسته مقید حالت دیگری از این مساله است که در آن طول مسیر مورد نظر باید از مقدار مشخصی مانند l کم‌تر شود. این حالت از مساله که به مساله‌ی l-هسته‌معروف است توسط نویسندگان مختلفی از جمله مراجع [۱۱-۱۴] مورد بررسی قرارگرفته‌است. هم‌چنین اگر هدف، یافتن زیردرختی با حداکثر k برگ باشد، مساله را مساله‌ی k-هسته می‌نامند. پنگ و همکاران [۱۵] دو الگوریتم با پیچیدگی زمانی $O\left(kn\right)$و $O\left(n\log n\right)$ برای یافتن k-هسته‌ی یک درخت ارایه کردند.

حال اگر هدف یافتن زیردرختی با حداکثر k برگ و با قطر حداکثر l باشد که مجموع فاصله‌های رئوس درخت از زیردرخت انتخاب شده کمینه شود، مساله را مساله‌ی (K,L)-هسته می‌نامند که در آن قطر درخت، ماکزیمم فاصله بین دو رأس از درخت است. الگوریتمی با پیچیدگی $O\left(n^{۲}logn\right)$ برای حل این مساله بر روی درخت‌های بدون وزن و درختهای با وزن مثبت توسط بکر وهمکارانش [۱۴] در سال **۲۰۰۲** ارایه شده‌است. متولی و همکاران [۱۶] الگوریتم آن‌ها را به حالت نیمه ناخوشایند یعنی درخت با وزن مثبت و منفی توسعه دادند.

در این مقاله ما به بررسی مساله هسته بر روی گراف‌های بازه‌ای می‌پردازیم. یک گراف بازه‌ای مجموعه‌ای از بازه‌ها است که این بازه‌ها می‌توانند متناظر با رئوس یک گراف باشند به‌طوری‌که اگر یالی بین رأس v_i و v_j برقرار باشد، دوبازه متناظر با آن‌ها I_i و I_j دارای اشتراک هستند. گراف‌های بازه‌ای دارای ساختار توپولوژی مناسبی هستند. هم‌چنین ابزار مهمی در زمینه‌های کاربردی از جمله باستان‌شناسی، زیست‌شناسی، زمان‌بندی و کنترل ترافیک هستند [۱۷و۱۸]. به همین‌دلیل محققان زیادی علاقه‌مند به مطالعه بر روی گراف‌های بازه‌ای شدند (مراجع [۱۹-۲۱] را ببینید). مساله ۱-میانه روی گراف‌های بازه‌ای با وزن مثبت توسط بسپامیاتنیخ و همکارانش [۱۹] بررسی شد. آن‌ها الگوریتمی خطی برای حل این مساله پیشنهاد کردند. چنگ و کانگ [۲۲] به بررسی

مساله p-ماکسین روی گراف‌های بازه‌ای پرداخته‌اند. قاسمی نیز در [۲۳] مساله‌ی P-ماکسین روی گراف‌های بازه‌ای که هر بازه دارای یک وزن مثبت است را بررسی کرده است. درخت‌های بازه‌ای حالت خاصی از گراف‌های بازه‌ای هستند که از هر بازه به بازه‌ی دیگر فقط یک مسیر وجود دارد و این گراف فاقد دور است. ساختار داده‌ای درخت‌های بازه‌ای در [۲۴] توسط ربیعی بررسی شده‌است.

ما در این مقاله ابتدا هسته را در یک گراف بازه‌ای تعریف کرده، سپس با ارایه خواصی برای این مساله روی درخت‌های بازه‌ای الگوریتمی خطی برای پیداکردن هسته درخت بازه‌ای ارایه می‌کنیم.

۲ مساله‌ی میانه بازه‌ای

فرض کنید G=(V,E) یک گراف بامجموعه رئوس V و مجموعه یال‌های E باشد. در یک گراف بازه‌ای هر رأس v_i تناظر یک به یک با بازه I_i دارد و یال $\left(v_i, v_j\right) \in E$ در صورتی وجوددارد که $I_i \cap I_j \neq \varnothing$. مساله p-میانه در یک گراف بازه‌ای عبارت‌است‌از یافتن p بازه یا سرویس‌دهنده روی شبکه با n بازه به عنوان نقاط تقاضا، به‌طوری‌که مجموع فاصله‌های n-p بازه‌ی دیگر تا p بازه موردنظر کمینه شود.

در این مقاله طول هر یال را یک درنظرگرفته و اگر رأس v_i دارای وزن w_i باشد طول بازه I_i را w_i در نظر می‌گیریم. بنابراین یک گراف بازه‌ای مجموعه‌ای از بازه‌ها است که اگر دوبازه I_i و I_j با هم اشتراک داشته باشند، یالی بین رأس v_i و v_j برقرار است. هم‌چنین فرض‌کنید $C\left(I_i, I_j\right)$ کوتاه‌ترین مسیر بازه‌های متصل I_i و I_j باشد. تعداد یال‌های روی $C\left(I_i, I_j\right)$ را فاصله بین I_i و I_j درنظرگرفته و با $d\left(I_i, I_j\right)$ نشان می‌دهیم. بنابر تعاریف فوق مساله p-میانه در گراف‌های بازه‌ای عبارت است‌از پیداکردن مجموعه $I=\left\{I_{i_1}, I_{i_2}, \ldots, I_{i_p}\right\}$ به طوری‌که تابع هدف زیر کمینه شود:

$$F\left(I\right)=\sum_{k=1}^{n} w\left(I_k\right) \min_{1 \leq j \leq p} d\left(I_k, I_{i_j}\right) \tag{۱}$$

فرض کنید $I_i=\left[a_i, b_i\right]$ که a_i و b_j به ترتیب نقاط انتهایی چپ و راست بازه I_i هستند. فرض می‌کنیم هیچ دو بازه‌ای در مجموعه بازه‌های I نقاط انتهایی یکسانی ندارند و $a_i < b_i$. نقاط انتهایی بازه‌ها را مرتب کرده و در بردار L قرار می‌دهیم. بنابراین

$$L=\left\{p_1, p_2, \ldots, p_{2n}\right\}$$

که p_i ها در آن یا a_i و یا b_i هستند. بنابراین در این بردار اگر $i > j$ آن گاه $b_i < b_j$ و اگر انتهای بازه‌ی I_i ابتدای بازه‌ی I_j باشد آن‌گاه در بردار L، b_i قبل از a_i قرار می‌گیرد. چنگ و کانگ [۷] نشان داده‌اند که این برچسب‌گذاری در زمان $O\left(nlogn\right)$ به‌دست می‌آید. اگر s کوچک‌ترین نقطه پایانی چپ و t بزرگ‌ترین نقطه پایانی راست باشد، آن‌گاه قرار می‌دهیم $s=a_1$، $t=b_n$ و $L \in \left\{1,2,\ldots,n\right\}$. هم‌چنین فرض می‌کنیم $\bigcup_{i=1}^{n} I_i=[s,t]$.

تعریف ۱: بازه‌ی I_i را در I ماکسیمال گوئیم، هرگاه مشمول هیچ بازه‌ی دیگری نباشد.

تعریف۲: بازه‌ی I_i را در I مینیمال گوئیم، هرگاه شامل هیچ بازه‌ی دیگری نباشد و بازه‌ی ماکسیمالی مانند I_j وجود داشته باشد که شامل آن باشد.

برای ادامه بحث نیاز به لم زیر داریم که اثبات آن در [۲۱] آمده است.

لم ۱: اگر بازه‌ی I_iدر بازه‌ی I_j قرار داشته‌باشد یعنی $a_j < a_i < b_i < b_j$ آن‌گاه برای هر بازه‌ی دیگر مانند I_h $(I_h \neq I_i, I_j)$ داریم:

$$d\left(I_h, I_j\right) \le d\left(I_h, I_i\right)$$

قضیه ۱: هیچ بازه‌ی مینیمالی نمی‌تواند کاندیدی برای جواب مساله ۱-میانه باشد.

اثبات: با توجه به لم ۱ اگر بازه‌ی I_i در بازه‌ی I_j قرار داشته باشد یا $a_j < a_i < b_i < b_j$آن‌گاه برای هر بازه‌ی دیگر I_h $(I_h \neq I_i, I_j)$ داریم:

$$d\left(I_h, I_j\right) \le d\left(I_h, I_i\right)$$

بنابراین چون فاصله بازه‌های دیگر به I_jکم‌تر از I_i است و برای مینیمم‌شدن تابع $F\left(I\right)$ کم‌ترین‌فاصله لازم است. بازه‌ی I_j می‌تواند جایگزین بازه I_i شود. بنابراین بازه I_i نمی‌تواند به عنوان کاندیدی برای مساله ۱-میانه درنظرگرفته شود.□

اگر $q \in L$ یک نقطه پایانی ازبازه‌ی I_i باشد. تعاریف زیر را در نظربگیرید:

$$I_i \cap I_j \neq \varnothing$$

هم‌چنین توابع $F_{aL}(q)$ و $F_{aR}(q)$ را به صورت زیر تعریف می‌کنیم:

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

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

بنابراین همان‌طور که در [۱۱] آمده‌است، برای یک کاندید ۱-میانه تابع هدف می‌تواند به‌صورت زیر فرمول‌بندی شود:

$$F(I_i) = \sum_{k=1}^{n} w\left(I_k\right) d\left(I_k, I_i\right) = F_{aL}\left(a_i\right) + F_{aR}\left(a_i\right) \tag{۲}$$

به‌طوری‌که بازه‌ای که تابع فوق را کمینه کند ۱-میانه است. هم‌چنین یانگ چنگ در [۱۱] نشان داده‌است که مساله‌ی ۱-ماکسین و ۱-میانه روی گراف‌های بازه‌ای که نقاط پایانی آن‌ها مرتب شده‌اند در زمان $O\left(n\right)$ قابل حل هستند.

۳ مساله‌ی پیداکردن هسته‌ی یک درخت بازه‌ای

هسته‌ی یک گراف مسیری است در گراف که مجموع فاصله‌ها از تمام رئوس تا این مسیر کمینه شود. بنابراین در یک گراف بازه‌ای یک هسته عبارت‌است‌از مجموعه‌ای از بازه‌های متصل بهم مانند $I = \left\{I_{i_1}, I_{i_2}, \ldots, I_{i_p}\right\}$ که

مجموع فاصله‌ها از تمام بازه‌ها به این مسیر بازه‌ای کمینه شود. لذا با توجه به تعاریف فوق تابع هدف در این مساله را می‌توان به‌صورت زیر درنظر گرفت:

$$F\left(I\right)=\sum_{k=1}^{n} w\left(I_k\right)\min_{1\le j\le p} d\left(I_k,I_{i_j}\right) \tag{۳}$$

به‌طوری‌که $\forall u\in\{1,2,\dots,p-1\}$ و $I_{i_u}\cap I_{i_{u+1}}\neq\varnothing$.

در ادامه به بررسی مساله پیداکردن هسته روی یک درخت بازه‌ای می‌پردازیم.

قضیه ۲: در یک درخت بازه‌ای اگر دو بازه‌ی I_i و I_j که $a_i\le a_j$ و $b_i\le b_j$ با هم اشتراک داشته و یالی بین رئوس متناظر با آن‌ها موجود باشد، هیچ بازه‌ی، I_h $(I_h\neq I_i,I_j)$ دیگری نمی‌تواند وجودداشته باشد به‌طوری که

$$a_j\le a_h<b_h\le b_i\,.$$

اثبات: فرض کنید بازه‌ی I_h ی وجود دارد که $I_h\neq I_i,I_j$ و $a_j\le a_h<b_h\le b_i$. از آن‌جا که $a_j\le a_h$ و $a_h<b_h\le b_i$ و $b_i\le b_j$ بنـابراین $a_j<a_h\le b_j$ یعنـی بـازه‌ی I_h بـا I_j دارای اشـتراک اسـت و $I_j\cap I_h\neq\varnothing$. هم‌چنین چون $a_j\le a_h$ و $a_i\le a_j$ بنابراین $a_i\le a_h$ می‌باشد و چون $a_h<b_h\le b_i$ لذا $a_h\le b_i$ درنتیجه $a_i<b_h\le b_i$ یعنی $I_i\cap I_h\neq\varnothing$. از طرفی طبق فـرض $I_i\cap I_j\neq\varnothing$ بنـابراین $I_i\cap I_h\cap I_j\neq\varnothing$ که نشان می‌دهد گراف دارای دور است و با فرض درخت بودن گراف تناقض دارد.□

قضیه۳: اگر $I=\{I_1,I_2,\dots,I_n\}$ مجموعه‌ی بازه‌ها در درخـت بـازه‌ای G و $I_p=\{I_{i_1},I_{i_2},\dots,I_{i_p}\}$ هسته‌ی درخت G باشد. آن‌گاه

$$F(I_p)=F_{aL}(a_{i_1})+F_{aR}(a_{i_1})-\sum_{j=1}^{p-1}w_{aR}(b_{i_j})-\sum_{j=2}^{p}w(I_{i_j})$$

اثبـات: مـی‌دانـیم $F(I_{i_1})=F_{aL}(a_{i_1})+F_{aR}(a_{i_1})$ بـا اضـافه کـردن بـازه‌هـای $I_{i_2},I_{i_3},\dots,I_{i_p}$ بـه I_{i_1} مسـیر $I_p=\{I_{i_1},\,I_{i_2},\dots,I_{i_p}\}$ تشکیل می‌شود. به وضوح $F(I_p)<F(I_{i_1})$. فرض کنید

$$F(I_p)=F(I_{i_1})-E$$

که $E>0$. ثابت می‌کنیم:

$$E=\sum_{j=1}^{p-1}w_{aR}(b_{i_j})+\sum_{j=2}^{p}w(I_{i_j})$$

برای بازه‌های I_j که $b_{i_1}<a_j<b_j<b_{i_2}$، چون بازه‌ی I_j با I_{i_1} دارای اشتراک نیست و با I_{i_2} اشتراک دارد، پس $d(I_j,I_{i_1})=2$ و $d(I_j,I_{i_2})=1$. از طرفـی مجمـوع وزن‌هـای بـازه‌هـای بـین I_{i_1} و I_{i_2} بـه انـدازه‌ی $w_{aR}(I_{i_1})-w_{aR}(I_{i_2})$ می‌باشد و از آن‌جا که فاصله تا I_{i_1}، ۲ واحـد اسـت کـه یـک واحـد از آن فاصلـه تـا I_{i_2} می‌باشد. بنابراین از مقدار $F(I_{i_1})$ به اندازه‌ی $w_{aR}(I_{i_1})-w_{aR}(I_{i_2})$ کاسته می‌شود. هم‌چنین برای بازه‌های I_j که $b_{i_2}<a_j<b_j<b_{i_3}$، چون بازه‌ی I_j با I_{i_2} دارای اشتراک نیست و با I_{i_3} اشتراک دارد، پس $d(I_j,I_{i_1})=3$ و $d(I_j,I_{i_3})=1$ و $d(I_{i_1},I_{i_3})=2$. از طرفـی مجمـوع وزن‌هـای بـازه‌هـای بـین I_{i_2} و I_{i_3} بـه انـدازه‌ی

$w_{aR}(I_{i_2})-w_{aR}(I_{i_3})$ می‌باشد و از آن‌جا که فاصله تا I_{i_1}، ۳ واحد است که یک واحد از آن فاصله تا I_{i_3} می‌باشد و $d(I_{i_1},I_{i_3})=2$. بنابراین $F(I_{i_1})$ به اندازه‌ی $2[w_{aR}(I_{i_2})-w_{aR}(I_{i_3})]$ کاهش می‌یابد. با ادامه این روند برای بازه‌های I_j که $b_{i_{p-1}}<a_j<b_j<b_{i_p}$، داریم: $d(I_j,I_{i_1})=p$ و $d(I_j,I_{i_k})=1$ که به اندازه‌ی $(p-1)[w_{aR}(I_{i_{p-1}})-w_{aR}(I_{i_p})]$ و برای بازه‌های I_j که $b_{i_p}<a_j$، اگر $d(I_j,I_{i_p})=k$، چون $d(I_j,I_{i_1})=p+k-1$ بنابراین به اندازه $(p-1)w_{aR}(I_{i_p})$ از مقدار $F(I_{i_1})$ کم می‌شود. از طرف دیگر خود بازه‌های $I_{i_2},I_{i_3},\ldots,I_{i_p}$ که در مسیر قرار گرفته شده‌اند نیز مقدار $F(I_{i_1})$ را کاهش می‌دهند این بازه‌ها در محاسبات قبل به طور کامل محاسبه نمی‌شوند. مثلاً بازه‌ی I_j در

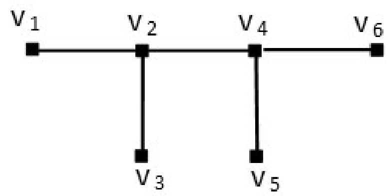
$(j-2)[w_{aR}(I_{i_{j-2}})-w_{aR}(I_{i_{j-1}})]$

در نظر گرفته می‌شود اما $d(I_j,I_{i_1})=j-1$ بنابراین برای هر بازه‌ی I_j، $j=2,3,\ldots,p$ به اندازه‌ی $w_{aR}(I_{i_j})$ باید از مقدار $F(I_p)$ کم کنیم. لذا داریم:

$$F(I_p)=F_{aL}(a_{i_1})+F_{aR}(a_{i_1})-\left[w_{aR}(b_{i_1})-w_{aR}(b_{i_2})\right]$$
$$-2\left[w_{aR}(b_{i_2})-w_{aR}(b_{i_3})\right]-\ldots-(p-1)[w_{aR}(b_{i_{p-1}})-w_{aR}(b_{i_p})]-pw_{aR}(b_{i_p})=$$
$$F_{aL}(a_{i_1})+F_{aR}(a_{i_1})-\sum_{j=1}^{p-1}w_{aR}(b_{i_j})-\sum_{j=2}^{p}w(I_{i_j}).$$

مثال زیر کاربرد قضیه ۳ را نشان می دهد.

مثال ۱: گراف نشان داده شده در شکل ۱ همراه با وزن رئوسی که در جدول۱ آمده است را در نظر بگیرید. بازه‌ای متناظر با گراف شکل ۱ در شکل ۲ آمده است.



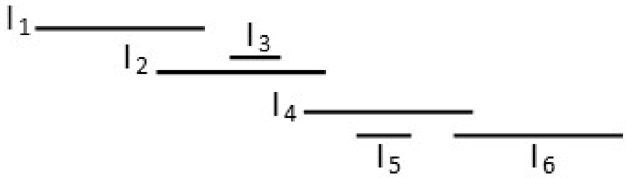
شکل ۱. گراف بازه‌ای

جدول ۱. وزن رئوس گراف شکل ۱

W_1	W_2	W_3	W_4	W_5	W_6
۲	۲	۱	۲	۱	۲

برای بازه‌ی I_2، مقدار تابع هدف به صورت زیر محاسبه می‌شود.

$$F(I_2)=F_{aL}(a_2)+F_{aR}(a_2)=2+9=11$$



شکل ۲. بازه‌های متناظر با گراف شکل ۱

حال اگر مسیر $\mathrm{I}=\{I_2,I_4,I_5\}=\{I_{i_1},I_{i_2},I_{i_3}\}$ را در نظر بگیریم. داریم:

$$F(I)=F_{aL}(a_2)+F_{aR}(a_2)-\sum_{j=1}^{2}w_{aR}(b_{i_j})-\sum_{j=2}^{3}w(I_{i_j})=11-3-3=5$$

بنابراین با استفاده از قضیه ۳ اگر مقدار تابع هدف را در ابتدای مسیر داشته باشیم با استفاده از مجموع وزن‌ها با محاسبات کم‌تری می‌توان مقدار تابع هدف را برای مسیر مورد نظر به دست آورد. هم‌چنین با استفاده از این قضیه می‌توان تغییرات تابع هدف را در زمان اضافه و کم شدن بازه‌ها به راحتی به‌دست‌آورد.

نتیجه ۱: اگر I_1 بازه‌ای در درخت بازه‌ای G و $F(I_1)=F_{aL}(a_1)+F_{aR}(a_1)$ و $I_p=\{I_{i_1},I_{i_2}\}$ مسیری از درخت G باشد. آن‌گاه

$$F(I_p)=F(I_1)-w_{aR}(b_1)-w(I_2)$$

قضیه ٤: اگر $I_p=\{I_{i_1},I_{i_2},\dots,I_{i_p}\}$ یک هسته‌ی درخت G باشد هیچ یک از بازه‌های $I_{i_2},I_{i_3},\dots,I_{i_{p-1}}$ بازه مینیمال نیستند.

اثبات: طبق قضیه۱ اگر بازه‌ی I_{i_j} ، $j=2,3,\dots,\mathrm{p}$ یک بازه‌ی مینیمال باشد بازه ماکسیمالی مشمول آن مانند I_{i_k} وجود دارد که فاصله‌ی بازه‌های دیگر تا آن از I_{i_j} کم‌تر است. بنابراین I_{i_k} می‌تواند جایگزین I_{i_j} باشد.□

نتیجه ۲: اگر $I_p=\{I_{i_1},I_{i_2},\dots,I_{i_p}\}$ یک هسته‌ی درخت G باشد هیچ یک از بازه‌های $I_{i_1},I_{i_2},\dots,I_{i_p}$ نمی‌توانند بازه‌ای غیرماکسیمال باشند. به عبارت دیگر تمام بازه‌های روی هسته ماکسیمال هستند.

قضیه ۵: اگر $I=\{I_1,I_2,\dots,I_n\}$ مجموعه‌ی بازه‌ها در درخت بازه‌ای G و $I_p=\{I_{i_1},I_{i_2},\dots,I_{i_p}\}$ مسیری از درخت G باشد و $I_k=\{I_{i_k},I_{i_{k+1}},\dots,I_{i_p}\}$ که $1<k<p$ آن‌گاه

$$F(I_k)=F(I_p)-F_{aL}(a_{i_1})-F_{aR}(a_{i_1})+F_{aL}(a_{i_k})+F_{aR}(a_{i_k})-\sum_{j=1}^{k-1}w_{aR}(b_{i_j})-\sum_{j=2}^{k}w(I_{i_j})$$

اثبات: از قضیه ۳ داریم:

$$F(I_p)=F_{aL}(a_{i_1})+F_{aR}(a_{i_1})-\sum_{j=1}^{p-1}w_{aR}(b_{i_j})\left(b_{i_j}\right)-\sum_{j=2}^{p}w(I_{i_j})$$

و

$$F(I_k)=F_{aL}(a_{i_k})+F_{aR}(a_{i_k})-\sum_{j=k}^{p-1}w_{aR}(b_{i_j})-\sum_{j=k+1}^{p}w(I_{i_j})$$

برای یافتن اختلاف $F(I_p)$ و $F(I_k)$، تفاضل دو رابطه‌ی فوق را به دست می‌آوریم.

$$F(I_p)-F(I_k)=F_{aL}(a_{i_1})+F_{aR}(a_{i_1})-\sum_{j=1}^{p-1}w_{aR}(b_{i_j})-\sum_{j=2}^{p}w(I_{i_j})-$$
$$[F_{aL}(a_{i_k})+F_{aR}(a_{i_k})-\sum_{j=k}^{p-1}w_{aR}(b_{i_j})-\sum_{j=k+1}^{p}w(I_{i_j})]=F_{aL}(a_{i_1})+F_{aR}(a_{i_1})-F_{aL}(a_{i_k})$$

بنابراین

$$F(I_k)=F(I_p)-F_{aL}(a_{i_1})-F_{aR}(a_{i_1})+F_{aL}(a_{i_k})+F_{aR}(a_{i_k})-\sum_{j=1}^{k-1}w_{aR}(b_{i_j})-\sum_{j=2}^{k}w(I_{i_j}).$$

۳-۱ الگوریتم

با استفاده از قضایا و نتایج فوق می‌توان الگوریتمی برای به‌دست‌آوردن هسته‌ی یک درخت بازه‌ای پیشنهاد کرد. به‌این صورت که اگر سمت چپ‌ترین بازه I_1 باشد، $F(I_1)$ را محاسبه می‌کنیم. طبق قضیه‌ی ۴ و نتیجه‌ی ۲ بازه‌های مینیمال نمی‌توانند کاندیدی برای قرارگرفتن در هسته باشند و بازه‌های ماکسیمال هسته را تشکیل می‌دهند. بنابراین برای ساختن هسته از I_1 شروع کرده و از بین بازه‌هایی است که شامل b_1 هستند بازه‌ی ماکسیمال I_2 را که بزرگ‌ترین نقطه‌ی انتهایی b_2 را دارد انتخاب و آن را $RSUC\left(I_1\right)$ نامیده و به مسیر $I_p=\{I_1\}$ اضافه می‌کنیم. بنابراین $RSUC\left(I_1\right)=I_2$ که در آن همانند [۶] برای هر نقطه q روی خط حقیقی $RSUC\left(q\right)=I_k$ اگر $b_k=\max\{b_i|\mathrm{q}\in I_i\}$ و برای هر بازه I_j تعریف می‌کنیم $RSUC(I_j)=RSUC(b_j)$. بنابراین طبق نتیجه‌ی ۱ داریم:

$$F(I_p)=F(I_1,I_2)=F(I_1)-w_{aR}(b)-w(I_2)$$

حال $RSUC\left(I_2\right)$ را یافته و آن را I_3 نامیده و به مسیر اضافه می‌کنیم. با ادامه‌ی این روند رابطه‌ی بازگشتی زیر را به‌دست می‌آوریم.

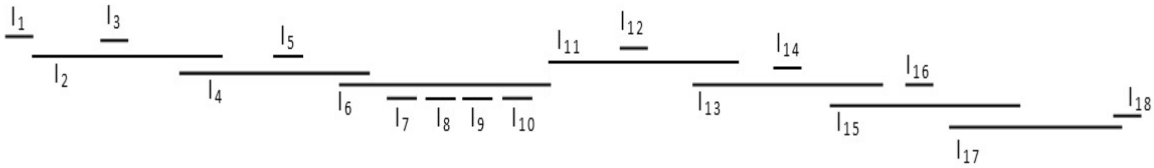
$$F(I_p)=F(I_1,I_2,\ldots,I_k)=F(I_1,I_2,\ldots,I_{k-1})-w_{aR}(b_{k-1})-w(I_k) \quad (۴)$$

اضافه شدن بازه‌ها به مسیر را تا سمت راست‌ترین بازه ادامه می‌دهیم. مسیر به‌دست‌آمده هسته می‌باشد.

قضیه ٦: مساله‌ی پیداکردن هسته‌ی یک درخت بازه‌ای که نقاط پایانی آن‌ها مرتب شده‌اند، می‌توانند در زمان $o(n)$ حل شود.

اثبات: فرض کنیم $I=\{I_1,I_2,\ldots,I_n\}$ مجموعه‌ی بازه‌ها در درخت بازه‌ای G باشد. با توجه به الگوریتم ارایه شده، در بدترین حالت اگر n بازه‌ی ماکسیمال متصل به هم داشته باشیم، هسته شامل تمام بازه‌ها است. مقدار $F(I_n)$ می‌تواند در زمان ثابت محاسبه شود و چون n بازه داریم، مساله‌ی پیداکردن هسته‌ی یک درخت بازه‌ای که نقاط پایانی آن‌ها مرتب شده‌اند، می‌تواند در زمان $o(n)$حل شود.□

مثال۲: بازه‌های شکل ۳ را در نظر بگیرید. که وزن رئوس در جدول ۲ داده‌شده‌است.



شکل ۳. بازه‌های مثال ۲

جدول ۲. وزن رئوس متناظر با بازه‌های شکل ۳

$w_{۱}$	$w_{۲}$	$w_{۳}$	$w_{۴}$	$w_{۵}$	$w_{۶}$	$w_{۷}$	$w_{۸}$	$w_{۹}$	$w_{۱۰}$
۰/۱	۱	۰/۱	۱	۰/۱	۱	۰/۱	۰/۱	۰/۱	۰/۱

$w_{۱۱}$	$w_{۱۲}$	$w_{۱۳}$	$w_{۱۴}$	$w_{۱۵}$	$w_{۱۶}$	$w_{۱۷}$	$w_{۱۸}$
۱	۰/۱	۱	۰/۱	۱	۰/۱	۱	۰/۱

برای به‌دست‌آوردن هسته‌ی آن طبق الگوریتم از سمت چپ‌ترین بازه شروع می‌کنیم. بنابراین

$$F(I_p)=F(I_۱)=۳۲/۷$$

سپس برای اضافه‌کردن بازه‌ی بعدی به مسیر، بازه‌ی $I_۲$ را انتخاب می‌کنیم زیرا $RSUC\left(I_۱\right)=I_۲$.

$$F(I_p)=F(I_۱,I_۲)=F(I_۱)-w_{aR}(b_۱)-w(I_۲)=۳۲/۷-۷-۱=۲۴/۷$$

داریم: $RSUC\left(I_۲\right)=I_۴$ بنابراین بازه‌ی $I_۴$ به مسیر اضافه می‌شود.

$$F(I_p)=F(I_۱,I_۲,I_۴)=F(I_۲)-w_{aR}(b_۲)-w(I_۴)=۲۴/۷-۵/۹-۱=۱۷/۸$$

به‌همین‌ترتیب در هر مرحله RSUC آخرین بازه را به‌دست‌آورده و به مسیر اضافه می‌کنیم. این روند را تا رسیدن به سمت راست‌ترین بازه یعنی $I_{۱۸}$ ادامه می‌دهیم.

$$F(I_p)=F(I_۱,I_۲,I_۴,I_۶)=F(I_۴)-w_{aR}(b_۴)-w(I_۶)=۱۷/۸-۴/۸-۱=۱۲$$

$$F(I_p)=F(I_۱,I_۲,I_۴,I_۶,I_{۱۱})=F(I_۶)-w_{aR}(b_۶)-w(I_{۱۱})=۱۲-۳/۴-۱=۷/۶$$

$$F(I_p)=F(I_۱,I_۲,I_۴,I_۶,I_{۱۱},I_{۱۳})=F(I_{۱۱})-w_{aR}(b_{۱۱})-w(I_{۱۳})=۷/۶-۲/۳-۱=۴/۳$$

$$F(I_p)=F(I_۱,I_۲,I_۴,I_۶,I_{۱۱},I_{۱۳},I_{۱۵})=F(I_{۱۳})-w_{aR}(b_{۱۳})-w(I_{۱۵})=۴/۳-۱/۲-۱=۲/۱$$

$$F(I_p)=F(I_۱,I_۲,I_۴,I_۶,I_{۱۱},I_{۱۳},I_{۱۵},I_{۱۷})=F(I_{۱۵})-w_{aR}(b_{۱۵})-w(I_{۱۷})=۲/۱-۰/۱-۱=۱$$

$$F(I_p)=F(I_۱,I_۲,I_۴,I_۶,I_{۱۱},I_{۱۳},I_{۱۵},I_{۱۷},I_{۱۸})=F(I_{۱۷})-w_{aR}(b_{۱۷})-w(I_{۱۸})=۱-۰/۱=۰/۹$$

بنابراین هسته‌ی گراف بازه‌ای فوق مسیر $P=\left\{I_۱,I_۲,I_۴,I_۶,I_{۱۱},I_{۱۳},I_{۱۵},I_{۱۷},I_{۱۸}\right\}$ است و $F(I_p)=۰/۹$ می‌باشد.

٤ نتیجه‌گیری

در این مقاله به بررسی مساله پیداکردن هسته روی درخت‌های بازه‌ای پرداختیم و نشان دادیم بازه‌هایی که روی هسته‌ی یک درخت قرار دارند نمی‌توانند بازه‌ای غیر ماکسیمال باشند. سپس الگوریتمی با پیچیدگی زمانی $o(n)$ برای پیداکردن هسته‌ی یک درخت بازه‌ای ارایه دادیم. در مطالعات بعدی می‌توان هسته‌ی درخت با محدودیت طول در نظرگرفت. بنابراین هدف یافتن هسته‌ای با طول L روی درخت بازه‌ای است. هم‌چنین می‌توان مساله را از یافتن هسته به (K,L)-هسته تغییر داد که در این حالت هدف یافتن زیردرختی با طول حداکثر L و تعداد برگ‌های حداکثرK است که مجموع فاصله‌های تمام بازه‌ها تا این زیردرخت کمینه شود.

منابع

[۲۱] امینی، ف.، (۱۳۸۸). گراف‌های بازه‌ای کاوشگر. مجله تحقیق در عملیات درکاربردهای آن، ۳(۲۲)، ۷۰-۶۱.

[۲۳] قاسمی، ح.، (۱۳۹۱)، مساله مکانیابی p-ماکسین، پایان‌نامه کارشناسی‌ارشد، دانشگاه صنعتی شاهرود، دانشکده ریاضی.

[۲۴] ربیعی، ا.، (۱۳۹۴)، گراف بازه‌ای و کاربردهایش، پایان‌نامه کارشناسی‌ارشد، دانشگاه صنعتی شاهرود، دانشکده ریاضی.

[1] Handler, G. Y., Mirchandani, P. B., (1979). Location On Network. Theory And Algorithms, Cambridg.

[2] Hamacher, H. W., Nickel, S., (1998). Classification of location models. Location science, 6, 229-242.

[3] Hassin, R., Tamir, A., (1991). Improved complexitybounds for location problems on the real line. Operation Research Letters, 10, 395-402.

[4] Maimani, H. R., (2008). Median and center of zero-divisor graph of commutative semigroups. Iranian Journal of Mathematical Sciences and Informatics, 3, 69-76.

[5] Fathali, J., Jafari-Rad, N., Sherbaf, S. R., (2014). The p-median and p-center problems on bipartite graphs. Iranian Journal of Mathematical Sciences and Informatics, 9, 37-43.

[6] Morgan, C. A., Slater, P. J., (1980). A linear algorithm for a core of atree. Journal of Algorithms, 1, 247-258.

[7] Becker, R. I., (1990). Inductive algorithms on finite trees. Quaest Math., 13, 165-181.

[8] Hakimi, S. L., Schmeichel, E. F., Labbe, M., (1993). On locating path or tree shaped facilities on network. Networks, 23, 543-555.

[9] Zaferanieh, M, Fathali, J., (2012). Finding a core of a tree with pos/neg weight. Math. Meth. Oper. Res., 26, 147-160.

[10] Tamir, A., Puerto, J., Mesa, J. A., Rodriguez-chia, A. M., (2005). Conditional location of path and tree shaped facilities on trees, Journal of Algorithms, 56, 50-75.

[11] Peng, S., Lo, W., (1996). Efficient algorithms for finding a core of a tree with a specified length, Journal of Algorithms, 20, 445-458.

[12] Minieka, E., Patel, N. H., (1983). On finding the core of a treewith a specified length. J. Alg., 345-352.

[13] Becker, R. I., Chang, Y. I., Lari, I., Scozzari, A., and Storchi, G., (2002). Finding the l-core of a tree, Discrete Applied Mathematics, 118, 25-42.

[14] Becker, R. I., Lari, I., Storchi, G., Scozzari, A., (2002). Efficient algorithms for finding the (k, l)-core of tree networks. Networks, 40, 208-215.

[15] Peng, S. T., Stephens, A. B., Yesha, Y., (1993). Algorithms for a core and k-tree core of a tree. Journal of Algorithms, 15, 143-159.

[16] Motevalli, S., Fathali, J., Zaferanieh, M., (2015). An efficient algorithm for finding the semi-obnoxious (k,l)-core of a tree. Journal of Mathematical Modeling, 3, 129-144.

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

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

[19] Bespamyatnikh, S., Bhattacharya, B., Keil, M., Kirkpatrick, D., Segal, M., (2002). Efficient algorithms for centers and medians in interval and circular-arc graphs. Networks, 29, 144-152.
[20] Cheng, T. C. E., Kang, L. Y., Ng, C. T., (2007). An improved algorithm for the p-center problem on interval graphs with unit lengths. Comput. Oper. Res., 34, 2215-2222.
[22] Cheng, Y., Kang, L., (2010). The p-maxian problem on interval graphs. Discrete Applied Mathematics, 158, 1986-1993.