

مساله درخت فراگیر ماکس + سام معکوس تحت فاصله همینگ وزن دار نوع جمعی با اصلاح بردار وزن

بهاره آزاد همپا^۱، فهیمه باروقی^{۲*}

۱- دانش آموخته کارشناسی ارشد، دانشگاه صنعتی سهند، گروه ریاضی کاربردی، تبریز، ایران

۲- دانشیار، دانشگاه صنعتی سهند، گروه ریاضی کاربردی، تبریز، ایران

رسید مقاله: ۲۰ خرداد ۱۴۰۰

پذیرش مقاله: ۲۶ دی ۱۴۰۰

چکیده

مساله درخت فراگیر ماکس + سام، یک درخت فراگیر T^* را در گراف G پیدا می کند که دارای مینیمم وزن ترکیبی $\max w(e) + \sum c(e)$ است و در آن $w(e)$ وزن و $c(e)$ هزینه یال $e \in E$ می باشند. این مساله در زمان $O(m \log n)$ حل می شود که در آن m تعداد یال ها و n تعداد رئوس گراف می باشد. در مساله درخت فراگیر ماکس + سام معکوس یک درخت فراگیر مفروض از گراف G که یک درخت فراگیر ماکس + سام بهینه نیست، در نظر گرفته می شود. سپس بردار وزن W به \bar{W} اصلاح می شود به طوری که درخت مفروض به یک درخت فراگیر ماکس + سام بهینه تبدیل گردد. هدف این است که هزینه تغییرات $\|\bar{W} - W\|$ تحت فاصله همینگ مینیمم شود. در این مقاله هدف ارایه یک روش جدید برای حل مساله درخت فراگیر ماکس + سام معکوس تحت فاصله همینگ وزن دار نوع جمعی با اصلاح بردار وزن نوع تنگنا می باشد. ابتدا مساله را فرمول بندی کرده و سپس یک الگوریتم ترکیبیاتی با زمان اجرای $O(m \log n)$ برای حل آن پیشنهاد می شود. در آخر یک مثال عددی برای نشان دادن کارایی روش پیشنهادی ارایه می شود.

کلمات کلیدی: مساله درخت ماکس + سام، بهینه سازی معکوس، فاصله همینگ.

۱ مقدمه

مساله درخت فراگیر ماکس + سام کلاسیک یکی از مسایل پر کاربرد در بهینه سازی می باشد. یک نمونه از کاربردهای مسایل درخت فراگیر ماکس + سام کلاسیک طراحی خطوط جاده ای بین شهری است. در تقابل با مساله درخت فراگیر ماکس + سام کلاسیک مساله درخت فراگیر ماکس + سام معکوس مطرح است که در سال های اخیر مورد توجه محققان قرار گرفته است و در زمینه هایی چون توموگرافی کامپیوتری، انتشار موج لرزه ای، برنامه ریزی ترافیک برای ارتباطات با سرعت بالا، تحلیل تلفیقی در بازاریابی و غیره کاربرد دارد. در سال ۱۹۹۶، پونن و نایر [۱] یک الگوریتم برای حل مساله درخت فراگیر ماکس + سام (MSST) در زمان $O(m \log n)$ ارایه دادند که در آن n تعداد رئوس و m تعداد یال ها است. در همان سال ژانگ و همکاران [۲]

* عهده دار مکاتبات

برای حل مساله درخت فراگیر مینیمم معکوس با محدودیت‌های بلوکی یک الگوریتم در زمان چندجمله‌ای^۲ پیشنهاد دادند. پس از آن ایری، سکالینگام و همکاران [۳] درخت فراگیر مینیمم معکوس را تحت نرم l_1 ، نرم l_1 وزن دار و نرم l_∞ به ترتیب در زمان‌های $O(n^3)$ ، $O(n^2 m \log(nc))$ و $O(n^2)$ حل کردند که در آن c ماکزیمم هزینه و m تعداد یال‌ها است. اهوچا و اورلین [۴] در سال ۲۰۰۰ پیچیدگی زمانی مساله درخت فراگیر مینیمم معکوس تحت نرم l_1 را با استفاده از ساختار ویژه مساله جریان مینیمم به $O(n^2 \log n)$ کاهش دادند. در سال ۲۰۰۳، هوچباوم [۵] پیچیدگی زمانی مساله درخت فراگیر مینیمم معکوس تحت نرم l_1 را به $O(n^{1.5} \log n \log c)$ کاهش داد. در سال‌های ۲۰۰۵ و ۲۰۰۶، ژانگ و همکاران [۶، ۷] مساله درخت فراگیر مینیمم معکوس تحت فاصله همینگ^۳ وزن دار نوع جمعی و مساله درخت فراگیر مینیمم معکوس مقید تحت فاصله همینگ تنگنا^۴ را در نظر گرفتند. آن‌ها با استفاده از مسایل پوشش رأسی با وزن مینیمم روی گراف دوبخشی مسایل مورد اشاره را در زمان‌های چندجمله‌ای قوی حل کردند. در سال ۲۰۰۷، یانگ و ژانگ [۸] الگوریتم‌هایی با زمان اجرای چندجمله‌ای قوی برای مساله درخت فراگیر تنگنا معکوس تحت نرم l_1 و نرم l_∞ ارائه دادند. همچنین یک الگوریتم با زمان اجرای چندجمله‌ای قوی برای مساله درخت فراگیر تنگنا ی معکوس تحت فاصله همینگ توسط لیو و همکاران [۹، ۱۰] در سال ۲۰۰۹ ارائه گردید. سپس گوان و پاردالوس [۱۱] مساله درخت فراگیر ماکس + سام معکوس^۵ را با اصلاح بردار هزینه نوع جمعی تحت نرم l_∞ وزن دار در نظر گرفتند و مساله درخت فراگیر ماکس + سام معکوس بی کران را به عنوان یک مساله بهینه‌سازی ترکیباتی کسری خطی فرمول‌بندی کردند و یک روش گسسته نیوتن را برای حل آن توسعه دادند. به علاوه آنها ثابت کردند که مساله درخت فراگیر ماکس + سام معکوس نامقید و مقید می‌تواند در زمان $O(n)$ حل شوند که در هر تکرار یک مساله درخت فراگیر ماکس + سام حل می‌شود.

در سال ۲۰۱۷، گوان، پاردالوس و ژانگ [۱۲] مساله درخت فراگیر ماکس + سام معکوس تحت نرم l_1 وزن دار با اصلاح بردار هزینه نوع جمعی را فرمول‌بندی کردند و الگوریتم تولید ستون را در زمان $O(m \log n)$ برای حل آن ارائه دادند. در همان سال گوان، هی، پاردالوس و ژانگ [۱۳] مساله درخت فراگیر ماکس + سام معکوس تحت فاصله همینگ با اصلاح بردار هزینه نوع جمعی را در نظر گرفته و یک الگوریتم جستجوی دودویی^۶ را برای آن ارائه دادند.

در این مقاله، مساله درخت فراگیر ماکس + سام معکوس تحت فاصله همینگ وزن دار نوع جمعی با اصلاح بردار وزن نوع تنگنا برای اولین بار در نظر گرفته می‌شود و الگوریتمی برای حل آن ارائه می‌شود. سازمان‌دهی مقاله به صورت زیر می‌باشد: بخش ۲ به معرفی مسایل درخت فراگیر کلاسیک می‌پردازد. در بخش ۳ مساله درخت فراگیر ماکس + سام معکوس تحت فاصله همینگ نوع جمعی با اصلاح بردار وزن نوع تنگنا در نظر

^۲ Polynomial

^۳ Hamming distance

^۴ Bottleneck

^۵ Max + Sum Spanning Tree

^۶ Binary search

گرفته شده و ویژگی‌های این مساله بررسی می‌شود. سپس یک الگوریتم جدید برای حل مساله پیشنهاد می‌شود. در آخر، نتایج حاصل از اجرای الگوریتم پیشنهادی روی یک مثال ارایه می‌شود.

۲ تعاریف و مفاهیم مقدماتی

در این بخش ابتدا مفاهیم اساسی، تعاریف و قضایایی از درخت‌های فراگیر ارایه می‌شود. سپس به بررسی برخی از مسایل معکوس پرداخته می‌شود.

۲-۱ درخت‌های فراگیر

تعریف ۱. [۱۴] فرض کنید $G(V, E, c, w)$ یک گراف همبند با مجموعه رأسی V و مجموعه یالی E باشد که در آن m تعداد یال‌ها و n تعداد رئوس گراف می‌باشد. فرض کنید برای هر یال $e \in E$ یک هزینه $c(e)$ و یک وزن $w(e)$ اختصاص داده شود. یک زیرگراف T از گراف G ، یک درخت فراگیر روی G نامیده می‌شود، هر گاه T یک زیرگراف فراگیر از G بوده و به علاوه یک درخت باشد. مجموعه درخت‌های فراگیر یک گراف G با نماد $\Gamma(G)$ نشان داده می‌شود. بنابراین هزینه جمعی درخت فراگیر T به صورت

$$C^s(T) = \sum_{e \in T} c(e),$$

و وزن تنگنای درخت فراگیر T به صورت

$$W^b(T) = \max_{e \in T} w(e),$$

تعریف می‌شود.

تعریف ۲. [۱۴] درجه یک رأس $i \in V$ برابر است با تعداد یال‌هایی از G که به رأس i متصل هستند و با نماد $\deg(i)$ نشان داده می‌شود.

قضیه ۳. [۱۵] تعداد درخت‌های فراگیر گراف G برابر با $\det(D_G)$ است که در آن D_G به صورت زیر تعریف می‌شود:

$$(D_G)_{i,j} = \begin{cases} \deg(i), & i = j, \\ -1, & i \neq j, (i,j) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

۲-۲ مساله درخت فراگیر مینیمم کلاسیک

تعریف ۴. [۱۴] فرض کنید G یک گراف با هزینه یالی $c(e)$ باشد. در یک مساله درخت فراگیر مینیمم روی G هدف پیدا کردن درخت فراگیر T با مینیمم هزینه جمعی $(C^s(T))$ می‌باشد. به عبارت دیگر هدف حل مدل بهینه‌سازی

$$\begin{array}{ll} \min & C^s(T) \\ \text{s.t.} & T \in \Gamma(G) \end{array}$$

است. جواب بهینه مساله فوق را یک درخت فراگیر مینیمم گویند.

یک روش برای یافتن درخت فراگیر مینیمم، استفاده از الگوریتم پریم^۷ است که دارای زمان اجرای $O(m \log n)$ می باشد. این الگوریتم کشف رئوس را از یک رأس دلخواه مانند r شروع می کند و تا زمانی ادامه می یابد که تمام رئوس کشف شود. نحوه ی کشف رئوس بدین صورت است که در هر تکرار از الگوریتم یک یال پیمایش می شود، به طوری که این یال درخت را به یک رأس کشف نشده وصل می کند.

۳-۲ مساله درخت فراگیر تنگنای کلاسیک

فرض کنید G یک گراف با وزن یالی $w(e)$ باشد. در مساله درخت فراگیر تنگنا هدف پیدا کردن درخت فراگیر T با مینیمم وزن تنگنا می باشد. به عبارت دیگر هدف حل مدل بهینه سازی

$$\begin{array}{ll} \min & W^b(T) \\ \text{s.t.} & T \in \Gamma(G), \end{array}$$

است. برای پیدا کردن درخت فراگیر تنگنا ابتدا کل زیردرخت های فراگیر گراف G را یافته و سپس ماکزیمم وزن هر درخت فراگیر را به دست آورید. درخت متناظر با کمترین مقدار ماکزیمم وزن را درخت فراگیر تنگنا می نامند.

۴-۲ مساله درخت فراگیر ماکس + سام کلاسیک

فرض کنید $G(V, E, c, w)$ یک گراف همبند باشد که به هر یال $e \in E$ ، یک هزینه $c(e)$ و یک وزن $w(e)$ تخصیص داده می شود.

تعریف ۵. [۱۴] دریک مساله MSST هدف پیدا کردن یک درخت فراگیر $T \in \Gamma(G)$ است، به طوری که جواب بهینه مدل زیر باشد:

$$\begin{array}{ll} \min_{T \in \Gamma(G)} & \left\{ \max_{e \in T} w(e) + \sum_{e \in T} c(e) \right\} \\ \text{s.t.} & T \in \Gamma(G). \end{array}$$

در سال ۱۹۸۳، استیلر و تارجان [۱۶] یک الگوریتم برای مساله MSST پیشنهاد کردند. فرض کنید:

• $\text{maketree}(v)$: درخت جدیدی ایجاد می کند که شامل رأس v بوده و قبلاً در هیچ درختی نبوده است.

• $\text{findroot}(v)$: ریشه درختی را برمی گرداند که شامل رأس v باشد.

• $\text{findmaxcost}(v)$: یالی با بالاترین مقدار هزینه روی مسیری از v به ریشه فعلی را برمی گرداند.

⁷ Prim

- $evert(v)$: درخت شامل رأس v را در v ریشه‌دار می‌کند.
- $link(v, w)$: دو درخت مجزا شامل رأس v و w را با افزودن یال (v, w) ترکیب می‌کند و رأس v را به عنوان ریشه درخت در نظر می‌گیرد.
- $cut(v)$: درخت شامل رأس v (ریشه نیست) را به دو درخت با حذف یال خروجی از رأس v افراز می‌کند.

الگوریتم ۱. مساله درخت فراگیر ماکس + سام را حل می‌کند.

شروع

گام ۱. مساله تنگنای

$$\min_{T \in I(G)} \max_{e \in T} (w_e),$$

را حل کنید و مقدار بهینه را برابر w^0 قرار دهید [۱۴].

گام ۲. یال‌های $E(w^0)$ را به صورت $e_1, e_2, \dots, e_r, \dots, e_m$ به ترتیب صعودی بر حسب $c(e)$ ها اندیس گذاری مجدد کنید که در آن

$$E(w^0) = \{e \in E \mid w(e) \leq w^0\}.$$

گام ۳. یال‌های $e \in [E - E(w^0)]$ را به ترتیب صعودی بر حسب $w(e)$ ها به صورت $e_m, \dots, e_{r+2}, e_{r+1}$ اندیس گذاری مجدد کنید.

گام ۴. به ازای هر $v \in V$ ، $maketree(v)$ را اجرا کنید.

گام ۵. قرار دهید $T = \emptyset$ ، $i = 1$.

گام ۶. تا زمانی که $|T| = |V| - 1$ ، عملیات زیر را انجام دهید:

فرض کنید e_i دارای رأس‌های انتهایی x و y باشد.

اگر $findroot(x) = findroot(y)$ ، آن‌گاه قرار دهید $i = i + 1$ ، در غیر این صورت اجرا کنید:

$$link(x, y)$$

$$T = T \cup \{e_i\}$$

گام ۷. قرار دهید

$$best - tree - info = w^0,$$

$$cost - tree = \sum_{e \in T} c(e),$$

$$best - obj = w^0 + (cost - tree).$$

گام ۸. به ازای $i = 1, 2, \dots, m - r$ عملیات زیر را انجام دهید:

یال e_{r+i} را انتخاب کنید و فرض کنید e_{r+i} دارای رأس‌های انتهایی v و w باشد.

عملیات $evert(v)$ را اجرا کنید.

فرض کنید $e = findmaxcost(v)$ که دارای رأس ابتدایی u است.

اگر $c(e) > c(e_{r+i})$ ، آن گاه عملیات زیر را انجام دهید:

$cut(u)$ را اجرا کنید،

$link(v, w)$ را اجرا کنید،

قرار دهید $T = T - e + e_{r+i}$ ،

قرار دهید $cost - tree = cost - tree + c(e_{r+i}) - c(e)$ ،

اگر $cost - tree + w(e_{r+i}) < best - obj$ ، آن گاه قرار دهید

$$best - obj = cost - tree + w(e_{r+i}),$$

$$best - tree - info = w(e_{r+i}).$$

گام ۹. قرار دهید $w^* = best - tree - info$.

گام ۱۰. درخت فراگیر به دست آمده را برابر T^* قرار دهید.

گام ۱۱. T^* و $best - obj$ را برگردانید.

پایان

لازم به ذکر است که الگوریتم ۱ دارای زمان اجرای $O(m \log n)$ است که در آن $n = |V|$ و $m = |E|$ و پیدا کردن یک درخت فراگیر تنگنا در زمان $O(m \log n)$ انجام می شود [۱۷، ۱۶]. به علاوه مرتب سازی هزینه ها و وزن ها نیز در زمان

$$O(r \log r) + O((m-r) \log(m-r)),$$

انجام می شود [۱۸] در نهایت یافتن درخت فراگیر مینیمم در زمان $O(m \log n)$ انجام می شود. بنابراین پیچیدگی زمانی الگوریتم ۱ برابر است با

$$O(m \log n) + O(r \log r) + O((m-r) \log(m-r)) = O(m \log n).$$

۳ مساله MSST معکوس تحت فاصله همینگ وزن دار نوع جمعی با اصلاح بردار وزن از نوع تنگنا

در این بخش مساله MSST معکوس تحت فاصله همینگ وزن دار نوع جمعی با اصلاح بردار وزن نوع تنگنا $(IMSST_{SH})^{\wedge}$ بیان و فرمول بندی می شود و سپس به بررسی خواص مساله MSST معکوس تحت فاصله همینگ پرداخته می شود.

⁸ Invers Max + Sum Spanning Tree

۳-۱ بیان مساله و فرمول بندی ریاضی

فرض کنید T_0 یک درخت فراگیر مفروض گراف $G(V, E, c, w)$ باشد، به طوری که یک جواب شدنی برای مساله $IMSST_{SH}$ است، هدف اصلاح بردار وزن w است به طوری که T_0 جواب بهینه مساله $IMSST_{SH}$ گردد و این تغییرات تا حد امکان مینیمم شود. به عبارت دیگر T_0 یک درخت بهینه برای گراف $G(V, E, c, \bar{w})$ تحت فاصله همینگ وزن دار نوع جمعی گردد. فرض کنید

$$H(\bar{w}(e), w(e)) = \begin{cases} 1, & \bar{w}(e) \neq w(e), \\ 0, & \text{otherwise.} \end{cases}$$

در این صورت مساله $IMSST_{SH}$ را می توان به صورت

$$\begin{aligned} \min \quad & \sum_{e \in E} q(e) H(\bar{w}(e), w(e)) \\ \text{s.t.} \quad & \bar{W}^b(T_0) + C^s(T_0) \leq \bar{W}^b(T) + C^s(T) \quad \forall T \in \Gamma(G), \\ & w(e) - l(e) \leq \bar{w} \leq w(e) + u(e) \quad \forall e \in E, \end{aligned} \quad (1)$$

فرمول بندی کرد که در آن $q(e) \geq 1$ هزینه تغییرات به ازای هر $e \in E$ و همچنین $l(e), u(e) \geq 0$ دامنه تغییرات است. باید l به گونه ای انتخاب شود که به ازای هر یال درختی، مقدار l از اختلاف وزن آن یال با مینیمم وزن بیشتر باشد. فرض کنید $w(e_i)$ ماکزیمم وزن درخت T_i باشد و

$$\alpha'(e_i) = \bar{w}(e_i) - w(e_i).$$

با توجه به تعریف α' مساله $IMSST_{SH}$ را می توان به صورت

$$\begin{aligned} \min \quad & \sum_{i \in |\Gamma|} q(e) H(\alpha'(e_i), 0) \\ \text{s.t.} \quad & W^b(T_0) + C^s(T_0) + \alpha'(e_0) \leq W^b(T_i) + C^s(T_i) + \alpha'(e_i), \quad \forall T_i \in \Gamma(G), i=1, \dots, |\Gamma|, \\ & -l(e_i) \leq \alpha'(e_i) \leq +u(e_i), \quad \forall i=0, 1, \dots, |\Gamma|, \end{aligned} \quad (2)$$

بازنویسی کرد.

قضیه ۶. مساله $IMSST_{SH}$ یک جواب بهینه دارد، به طوری که

$$\begin{cases} \alpha'(e_0) \leq 0, & e_0 \in T_0, \\ \alpha'(e_i) \geq 0, & e_i \notin T_0. \end{cases}$$

به عبارت دیگر می توان یک جواب بهینه را با کاهش ماکزیمم وزن یال های درختی و با افزایش ماکزیمم وزن یال های غیردرختی به دست آورد.

برهان . فرض کنید α'_i جواب بهینه مدل (۲) باشد. برای یال های درختی دو حالت زیر را در نظر بگیرید:
حالت اول:

$$\circ \prec \alpha'_i(e_{0_j}) \leq u(e_j), \quad e_{0_j} \in T_0.$$

حالت دوم:

$$-l(e) \leq \alpha'_i(e_0) \leq \circ, \quad e_0 \in T_0.$$

و برای تمام یال های غیردرختی

$$\circ \leq \alpha'_i(e_i) \leq u(e_i).$$

در این صورت

$$\begin{cases} \alpha'_i(e_{0_j}) = -\alpha'_i(e_{0_j}), & e_{0_j} \in T_0, \circ \leq \alpha'_i(e_{0_j}) \leq u(e_{0_j}), \\ \alpha'_i(e_0) = \alpha'_i(e_0), & e_0 \in T_0, \\ \alpha'_i(e_i) = \alpha'_i(e_i), & e_i \in T_i. \end{cases}$$

اگر ماکزیمم وزن انتخاب شده برای درخت T_0 در حالت دوم صدق کند، آن گاه خواهیم داشت:

$$\begin{aligned} W^b(T_0) + C^s(e_0) + \alpha'_i(e_0) &= W^b(T_0) + C^s(T_0) + \alpha'_i(e_0) \\ &\leq W^b(T) + C^s(T) + \alpha'_i(e_i) = W^b(T) + C^s(T) + \alpha'_i(e_i). \end{aligned} \quad (3)$$

اما اگر در حالت اول صدق کند، می توان نوشت:

$$\begin{aligned} W^b(T_0) + C^s(T_0) + \alpha'_i(e_{0_j}) &= W^b(T_0) + C^s(T_0) - \alpha'_i(e_{0_j}) \\ &= W^b(T_0) + C^s(T_0) + \alpha'_i(e_0) \leq W^b(T) + C^s(T) + \alpha'_i(e_i) \\ &= W^b(e_i) + C^s(T) + \alpha'_i(e_i). \end{aligned} \quad (4)$$

با توجه به روابط (۳) و (۴) می توان نتیجه گرفت که α_i یک جواب شدنی برای مساله (۲) است. همچنین بدیهی است که مقادیر عددی α'_i و α'_i یکسان هستند. از این رو α'_i جواب بهینه مساله (۲) نیز است. این استدلال را می توان به طور مشابه برای یال های غیردرختی نیز به کار برد. \square

محدودیت های مدل (۲) را می توان به صورت

$$\begin{aligned}
 W^b(T_0) + C^s(T_0) + \alpha'(e_0) &\leq W^b(T_i) + C^s(T_i) + \alpha'(e_i) \\
 \Leftrightarrow \alpha'(e_0) - \alpha'(e_i) &\leq W^b(T_i) + C^s(T_i) - W^b(T_0) - C^s(T_0) \\
 \Leftrightarrow \alpha'(e_i) - \alpha'(e_0) &\geq W^b(T_0) + C^s(T_0) - W^b(T_i) - C^s(T_i) \\
 &= \alpha'(e_i) - \alpha(e_0) \geq f(w, c, T_0) - f(w, c, T_i)
 \end{aligned} \tag{۵}$$

ساده کرد که در آن

$$f(w, c, T) = \max_{e \in T} w(e) + \sum_{e \in T} c(e) = W^b(T) + C^s(T).$$

حال اگر $\beta(e)$ را به صورت

$$\beta(e) = \begin{cases} -\alpha(e_0), & e_0 \in W^b(T_0), \\ \alpha(e_i), & e_i \in W^b(T_i), i = 1, \dots, |\Gamma|. \end{cases} \tag{۶}$$

تعریف کنیم، آن گاه با استفاده از (۵) و قضیه ۵ و (۶) مدل $IMSST_{SH}$ را می توان به صورت

$$\min \sum_{e \in E} q(e) H(\beta(e), \cdot) \tag{۷}$$

$$\text{s.t. } \beta(e_i) - \beta(e_0) \geq f(w, c, T_0) - f(w, c, T), \quad \forall T \in \Gamma(G), i = 1, \dots, |\Gamma|,$$

$$\cdot \leq \beta(e_0) \leq l(e), \quad e_0 \in T_0,$$

$$\cdot \leq \beta(e_i) \leq u(e), \quad e_i \in T_i, i = 1, \dots, |\Gamma|.$$

بازنویسی کرد. حال شدنی بودن مساله $IMSST_{SH}$ را بررسی می کنیم. فرض کنید

$$\tilde{w}(e) = \begin{cases} w(e) - l(e), & e_0 \in W^b(T_0), \\ w(e) + u(e), & e_i \in W^b(T_i), i = 1, \dots, |\Gamma|. \end{cases} \tag{۸}$$

لم زیر شرط شدنی بودن مساله را بیان می کند.

لم ۷. فرض کنید \tilde{T} زیردرخت فراگیر ماکس + سام بهینه روی گراف $G(V, E, c, \tilde{w})$ باشد.

(۱) اگر $f(\tilde{w}, c, T_0) = f(\tilde{w}, c, \tilde{T})$ ، آن گاه مساله $IMSST_{SH}$ شدنی است.

(۲) اگر $f(\tilde{w}, c, T_0) > f(\tilde{w}, c, \tilde{T})$ ، آن گاه مساله $IMSST_{SH}$ نشدنی است.

برهان. اثبات لم بدیهی است زیرا بنا به تعریف $\tilde{w}(e)$ تمام یال ها دارای ماکزیمم تغییرات وزنی هستند و اگر

درخت نابینه T_0 بعد از ماکزیمم تغییرات در وزن یال ها بهینه نشود، مساله نشدنی خواهد بود. \square

۲-۳ الگوریتم حل

در این بخش یک الگوریتم برای یافتن جواب بهینه مساله $IMSST_{SH}$ ارائه می‌شود. ابتدا مقادیر

$$\{q(e) \mid e \in E\}$$

$$q(e_{i_1}) \leq \dots \leq q(e_{i_t})$$

مرتب‌سازی می‌شود که در آن τ در هر تکرار از الگوریتم برابر تعداد دورهای گراف G به اضافه یک است.

حال گراف $G(V, E, c, w)$ ، بردار هزینه تغییرات q و بردارهای دامنه تغییرات l و u را در نظر بگیرید.

الگوریتم ۲ را می‌توان برای حل مساله $IMSST_{SH}$ ارائه داد.

الگوریتم ۲. مساله $IMSST_{SH}$ را حل می‌کند.

شروع

گام ۱. با استفاده از الگوریتم ۱ زیر درخت بهینه گراف $G(V, E, c, \tilde{w})$ را بیابید و آن را \tilde{T} بنامید.

• اگر $f(\tilde{w}, c, T_0) = f(\tilde{w}, c, \tilde{T})$ ، آن‌گاه نمونه‌شدنی است و به گام ۲ بروید.

• اگر $f(\tilde{w}, c, T_0) > f(\tilde{w}, c, \tilde{T})$ ، آن‌گاه نمونه‌نشده‌ای است و به گام ۶ بروید.

گام ۲. قرار دهید $k=1$ و $Q = \emptyset$ ، هزینه تغییرات یال‌های e_i که

$$e_i = W^b(T_i), \quad i = 1, \dots, |\Gamma|,$$

را به صورت صعودی

$$q(e_{i_1}) \leq \dots \leq q(e_{i_t})$$

مرتب کنید. یال e_i با کوچک‌ترین مقدار q که قبلاً انتخاب نشده را انتخاب کرده و قرار دهید

$$Q = Q \cup e_i$$

گام ۳. اگر $e_i \in T_0$ ، آن‌گاه قرار دهید

$$\bar{w}_{(k)}(e_i) = w(e_i) - l(e_i),$$

در غیر این صورت قرار دهید

$$\bar{w}_{(k)}(e_i) = w(e_i) + u(e_i).$$

گام ۴. یک درخت فراگیر ماکس + سام بهینه \bar{T} روی گراف $G(V, E, c, \bar{w}_{(k)})$ پیدا کنید.

• اگر $f(\bar{w}_{(k)}, c, T_0) = f(\bar{w}_{(k)}, c, \bar{T})$ ، آن‌گاه T_0 یک درخت بهینه بوده و به گام ۵ بروید.

• اگر $f(\bar{w}_{(k)}, c, T_0) > f(\bar{w}_{(k)}, c, \bar{T})$ ، آن‌گاه قرار دهید $k = k + 1$ و به گام ۲ برگردید.

گام ۵. مقدار بهینه برابر است با

$$q^* = \sum q(e_i),$$

که در آن $e_i \in Q$. جواب بهینه مدل (۷) نیز برابر است با

$$\beta^*(e) = \begin{cases} l(e), & e \in T_0 \cap Q, \\ u(e), & e \notin T_0 \cap Q, \\ , & e \notin Q, \end{cases} \quad (9)$$

در نتیجه جواب بهینه مدل (۱) به صورت

$$\bar{w}^*(e) = \begin{cases} w(e) - l(e), & e \in T_0 \cap Q, \\ w(e) + u(e), & e \notin T_0 \cap Q, \\ w(e), & e \notin Q. \end{cases} \quad (10)$$

خواهد بود.

پایان

لازم به ذکر است که در هر تکرار از الگوریتم ۲ تغییر ماکزیمم وزن در برخی از زیر درخت‌ها باعث تغییر برخی متغیرها و در نتیجه تغییر محدودیت‌ها خواهد شد. بنابراین در هر تکرار از الگوریتم یک مساله جدید حل خواهد شد.

بدیهی است که q^* جواب بهینه مساله $IMSST_{SH}$ است، زیرا اگر مساله شدنی باشد، آن گاه گام ۴ الگوریتم زمانی متوقف می‌شود که وزن ترکیبی درخت T_0 برابر وزن ترکیبی درخت بهینه است و در عین حال در هر تکرار از گام ۲ الگوریتم کوچک‌ترین مقدار $q(e)$ انتخاب شده باشد. همچنین بدیهی است که $\beta^*(e)$ در محدودیت‌ها صدق می‌کند، زیرا با توجه به مساله (۷) مقدار $\beta^*(e)$ از $l(e)$ کمتر و از $u(e)$ بیشتر نمی‌شود.

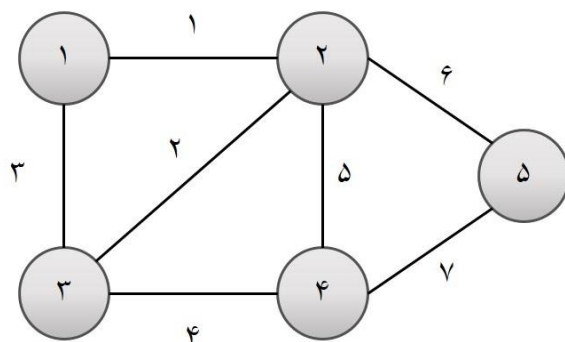
قضیه ۸. الگوریتم ۲ مساله درخت فراگیر ماکس + سام معکوس تحت فاصله همینگ وزن‌دار نوع جمعی با اصلاح بردار وزن نوع تنگنا را در زمان $O(m \log n)$ حل می‌کند.

برهان. گام اول الگوریتم ۲ دارای پیچیدگی زمانی $O(m \log n)$ است که در آن یک مساله $MSST$ حل می‌شود [۱۱]. در گام دوم عملیات مرتب‌سازی صورت می‌گیرد که دارای پیچیدگی زمانی $O(c \log c)$ است [۱۹]. توجه داشته باشید که c تعداد دورهای گراف G به اضافه یک است. گام سوم الگوریتم در زمان $O(1)$ انجام می‌شود. گام چهارم دارای پیچیدگی زمانی $O(m \log n)$ است، زیرا در هر تکرار یک مساله $MSST$ حل می‌شود. بنابراین پیچیدگی زمانی الگوریتم ۲ برابر با $O(m \log n)$ است. □

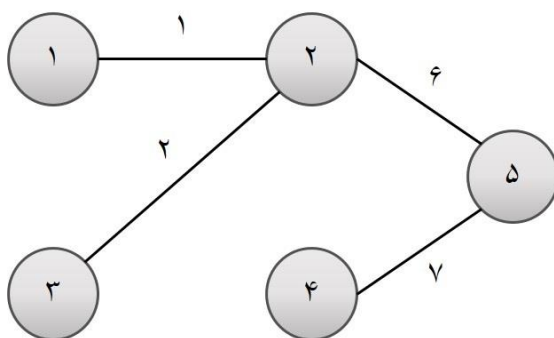
۳-۳ مثال عددی

گراف همبند G در شکل ۱ و درخت نابینه T_0 در شکل ۲ را در نظر بگیرید. اطلاعات مربوط به هر یال گراف G در جدول ۱ آورده شده است. هدف اصلاح بردار وزن $w(e)$ به $\bar{w}(e)$ است، به طوری که T_0 به یک درخت فراگیر ماکس + سام بهینه تبدیل شود.

آزاد همپا و باروقی، مساله درخت فراگیر ماکس + سام معکوس تحت فاصله همینگ وزن دار نوع جمعی با اصلاح بردار وزن



شکل ۱. گراف G مربوط به مثال عددی



شکل ۲. درخت T_0 مربوط به مثال عددی

جدول ۱. اطلاعات مربوط به گراف G

شماره یال	$w(e)$	$c(e)$	$q(e)$	l	u
۱	۹	۵	۳	۸	۷
۲	۳	۴	۷	۲	۶
۳	۲	۶	۵	۱	۵
۴	۵	۸	۴	۳	۸
۵	۸	۲	۲	۷	۹
۶	۷	۳	۶	۶	۷
۷	۶	۱	۸	۴	۵

گراف G دارای ۲۱ زیردرخت فراگیر است. در ابتدا با حل مساله MSST روی گراف $G(V, E, c, \tilde{w})$

شدنی بودن مساله مورد بررسی قرار می گیرد که در آن

$$\tilde{w} = (1, 1, 7, 13, 17, 1, 2),$$

با توجه به مدل (۷) فرمول بندی مساله به صورت زیر است،

$$\begin{aligned}
 \min \quad & (\mathfrak{r}H(\beta_{\lambda},^{\circ}) + \nu H(\beta_{\tau},^{\circ}) + \delta H(\beta_{\tau},^{\circ}) + \mathfrak{r}H(\beta_{\epsilon},^{\circ}) + \mathfrak{r}H(\beta_{\delta},^{\circ}) + \epsilon(\beta_{\epsilon},^{\circ}) \\
 & + \lambda H(\beta_{\nu},^{\circ})) \\
 \text{s.t.} \quad & \beta_{\lambda} + \beta_{\delta} \geq 1, \\
 & \beta_{\lambda} + \beta_{\epsilon} \geq 1, \\
 & \beta_{\lambda} + \beta_{\nu} \geq -3, \\
 & 0 \leq \beta_{\lambda} \leq 8, \\
 & 0 \leq \beta_{\tau} \leq 2, \\
 & 0 \leq \beta_{\tau} \leq 5, \\
 & 0 \leq \beta_{\epsilon} \leq 8, \\
 & 0 \leq \beta_{\delta} \leq 9, \\
 & 0 \leq \beta_{\epsilon} \leq 6, \\
 & 0 \leq \beta_{\nu} \leq 4,
 \end{aligned}$$

لیست مرتب شده به صورت

$$q_{\delta_{\lambda}} \leq q_{\lambda} \leq q_{\epsilon} \leq q_{\nu}$$

به دست می آید. یال ۵ به دلیل داشتن مینیمم هزینه تغییرات انتخاب می شود و چون یال درختی نیست، وزن آن افزایش می یابد:

$$\bar{w}_{\delta} = w_{\delta} + u_{\delta} = 17.$$

در گام ۴ با حل مساله MSST روی گراف $G(V, E, c, \bar{w})$ داریم:

$$f(\bar{w}, c, T_0) \succ f(\bar{w}, c, \bar{T})$$

بنابراین الگوریتم را تکرار می کنیم. فرمول بندی جدید مساله با توجه به بردار وزن جدید به صورت زیر است

$$\begin{aligned}
 \min \quad & (\mathfrak{r}H(\beta_{\lambda},^{\circ}) + \nu H(\beta_{\tau},^{\circ}) + \delta H(\beta_{\tau},^{\circ}) + \mathfrak{r}H(\beta_{\epsilon},^{\circ}) + \mathfrak{r}H(\beta_{\delta},^{\circ}) + \epsilon(\beta_{\epsilon},^{\circ}) \\
 & + \lambda H(\beta_{\nu},^{\circ})) \\
 \text{s.t.} \quad & \beta_{\lambda} + \beta_{\delta} \geq -7, \\
 & \beta_{\lambda} + \beta_{\epsilon} \geq 1, \\
 & \beta_{\lambda} + \beta_{\nu} \geq -3, \\
 & 0 \leq \beta_{\lambda} \leq 8, \\
 & 0 \leq \beta_{\tau} \leq 2, \\
 & 0 \leq \beta_{\tau} \leq 5, \\
 & 0 \leq \beta_{\epsilon} \leq 8, \\
 & 0 \leq \beta_{\delta} \leq 9, \\
 & 0 \leq \beta_{\epsilon} \leq 6, \\
 & 0 \leq \beta_{\nu} \leq 4.
 \end{aligned}$$

لیست مرتب شده به صورت

$$q_{\Delta_1} \leq q_{\Delta_2} \leq q_{\Delta_3} \leq q_{\Delta_4}$$

به دست می آید. یال ۱ انتخاب می شود به دلیل این که در تکرار قبل یال ۵ انتخاب شده است و چون یال درختی است، وزن آن کاهش می یابد:

$$\bar{w}_1 = w_1 + l_1 = 1.$$

در گام ۴ با حل مساله MSST روی گراف $G(V, E, c, \bar{w})$ داریم:

$$f(\bar{w}, c, T_0) = f(\bar{w}, c, \bar{T})$$

بنابراین جواب بهینه مدل (۷) به صورت

$$\beta^* = (\lambda, \circ, \circ, \circ, \circ, \circ, \circ)^T,$$

و جواب بهینه مدل (۱) به صورت

$$\bar{w}^* = (1, 3, 2, 5, 17, 7, 6)^T,$$

به دست می آید.

۴ نتیجه گیری

در این مقاله برای اولین بار، مساله درخت فراگیر ماکس + سام معکوس تحت فاصله همینگ وزن دار نوع جمعی با اصلاح بردار وزن نوع تنگنا در نظر گرفته و فرمول بندی شد. کاربردهای فراوان این مساله در زندگی واقعی انگیزه ای برای بررسی و حل این مساله ایجاد کرد. نشان داده شد که با کاهش ماکزیمم وزن یال های درختی و با افزایش ماکزیمم وزن یال های غیردرختی می توان به جواب بهینه دست یافت. سپس شرط شدنی بودن مساله با بیان لم ۷ ثابت گردید. در آخر، یک الگوریتم جدید ترکیبیاتی برای حل مساله ارایه شد که با توجه به قضیه ۸ دارای زمان اجرای چند جمله ای $O(m \log n)$ می باشد.

منابع

- [1] Punnen, A. P., Nair, K. P. K. (1996). An $O(m \log n)$ algorithm for the max+ sum spanning tree problem. European Journal of Operational Research, 89(2), 423-426.
- [2] Zhang, J., Liu, Z., Ma, Z. (1996). On the inverse problem of minimum spanning tree with partition constraints. Mathematical methods of operations research, 44(2), 171-187.
- [3] Sokkalingam, P. T., Ahuja, R. K., Orlin, J. B. (1999). Solving inverse spanning tree problems through network flow techniques. Operations Research, 47(2), 291-298.
- [4] Ahuja, R. K., Orlin, J. B. (2000). A faster algorithm for the inverse spanning tree problem. Journal of Algorithms, 34(1), 177-193.
- [5] Hochbaum, D. S. (2003). Efficient algorithms for the inverse spanning-tree problem. Operations Research, 51(5), 785-797.
- [6] He, Y., Zhang, B., Yao, E. (2005). Weighted inverse minimum spanning tree problems under Hamming distance. Journal of Combinatorial Optimization, 9(1), 91-100.
- [7] Hang, B., Zhang, J., He, Y. (2006). Constrained inverse minimum spanning tree problems under the bottleneck-type Hamming distance. Journal of Global Optimization, 34(3), 467-474.

- [8] Yang, X., Zhang, J. (2007). Some inverse min-max network problems under weighted l_1 and l_∞ norms with bound constraints on changes. *Journal of Combinatorial Optimization*, 13(2), 123-135.
- [9] Liu, L., Yao, E. (2007). Inverse min-max spanning tree problem under the weighted sum-type Hamming distance. In *International Symposium on Combinatorics, Algorithms, Probabilistic and Experimental Methodologies*. Springer, Berlin, Heidelberg, 375-383.
- [10] Liu, L., Wang, Q. (2009). Constrained inverse min-max spanning tree problems under the weighted Hamming distance. *Journal of Global Optimization*, 43(1), 83-95.
- [11] Guan, X., Pardalos, P. M., Zuo, X. (2015). Inverse Max+ Sum spanning tree problem by modifying the sum-cost vector under weighted l_∞ Norm. *Journal of Global Optimization*, 61(1), 165-182.
- [12] Guan, X., He, X., Pardalos, P. M., Zhang, B. (2017). Inverse max + sum spanning tree problem under Hamming distance by modifying the sum-cost vector. *Journal of Global Optimization*, 69(4), 911-925.
- [13] Guan, X., Pardalos, P. M., Zhang, B. (2017). Inverse max+ sum spanning tree problem under weighted l_1 norm by modifying the sum-cost vector. *Optimization Letters*, 1-13.
- [14] Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2001). *Introduction to algorithms*. MIT press.
- [15] Even, S. (1973). *Algorithmic combinatorics* (Vol. 190). New York: Macmillan.
- [16] Sleator, D. D., Tarjan, R. E. (1983). A data structure for dynamic trees. *Journal of computer and system sciences*, 26(3), 362-391.
- [17] Galil, Z., Schieber, B. (1988). On finding most uniform spanning trees. *Discrete Applied Mathematics*, 20(2), 173-175.
- [18] Aho, A. V., Hopcroft, J. E. (1974). *The design and analysis of computer algorithms*. Pearson Education India.
- [19] Alsuwaiyel, M. H. (1999). *Algorithms: Design Techniques And Analysis*. World Scientific.