مجله تحقیق در عملیات در کاربردهای آن، سال نوزدهم، شماره دوم (پیاپی ۷۳)،تابستان ۰۱، ص ص ۹۱-۷۷
شاپا ۷۲۸۶-۲۲۵۱

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

بهاره آزاد همپا[1]، فهیمه باروقی*[2]
۱- دانش‌آموخته 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 می‌کند که دارای مینیمم وزن ترکیبی $\max w(e)+\sum c(e)$ است و در آن $w(e)$ وزن و $c(e)$ هزینه یال $e\in E$ می‌باشند. این مساله در زمان $O(m \log n)$ حل می‌شود که در آن m تعداد یال‌ها و n تعداد رئوس گراف می‌باشد. 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 می‌شود. سپس بردار وزن w به $\bar{w}$ اصلاح می‌شود به‌طوری‌که 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 وزن‌دار نوع جمعی با اصلاح بردار وزن نوع تنگنا می‌باشد. ابتدا مساله را فرمول‌بندی کرده و سپس یک الگوریتم ترکیبیاتی با زمان اجرای $O(m \log n)$ برای حل آن پیشنهاد می‌شود. در آخر یک مثال عددی برای نشان دادن کارایی روش پیشنهادی ارایه می‌شود.

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

۱ مقدمه

مساله درخت فراگیر ماکس + سام کلاسیک یکی از مسایل پرکاربرد در بهینه‌سازی می‌باشد. یک نمونه از کاربردهای مسایل درخت فراگیر ماکس + سام کلاسیک طراحی خطوط جاده‌ای بین شهری است. در تقابل با مساله درخت فراگیر ماکس + سام کلاسیک مساله درخت فراگیر ماکس + سام معکوس مطرح است که در سال‌های اخیر مورد توجه محققان قرار گرفته است و در زمینه‌هایی چون توموگرافی کامپیوتری، انتشار موج لرزه‌ای، برنامه‌ریزی 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 یال‌ها است. در همان سال ژانگ و همکاران [۲]

* عهده‌دار مکاتبات
آدرس الکترونیکی: baroughi@sut.ac.ir ; baroughi22454@yahoo.com

برای حل مساله درخت فراگیر مینیمم معکوس با محدودیت‌های بلوکی یک الگوریتم در زمان چندجمله‌ای[2] پیشنهاد دادند. پس از آن ایری، سکالینگام و همکاران [۳] درخت فراگیر مینیمم معکوس را تحت نرم l_1، نرم l_1 وزن‌دار 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. اهوجا 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[3] وزن‌دار نوع جمعی و مساله درخت فراگیر مینیمم معکوس مقید تحت فاصله همینگ تنگنا[4] 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_∞ ارایه دادند. همچنین یک الگوریتم با زمان اجرای چندجمله ای قوی برای مساله درخت فراگیر تنگنا ی معکوس تحت فاصله همینگ توسط لیو و همکاران [۹,۱۰] در سال ۲۰۰۹ ارایه گردید. سپس گوان و پاردالوس [۱۱] مساله درخت فراگیر ماکس + سام معکوس[5] را با اصلاح بردار هزینه نوع جمعی تحت نرم l_∞ وزن‌دار در نظر گرفتند و مساله درخت فراگیر ماکس + سام معکوس بی‌کران را به‌عنوان یک مساله بهینه‌سازی ترکیبیاتی کسری خطی فرمول‌بندی کردند و یک روش گسسته نیوتن را برای حل آن توسعه دادند. به‌علاوه 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 وزن‌دار با اصلاح بردار هزینه نوع جمعی را فرمول‌بندی کردند و الگوریتم تولید ستون را در زمان $O(m \log n)$ برای حل آن ارایه 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[6] را برای آن ارایه دادند.

در این مقاله، مساله درخت فراگیر ماکس + سام معکوس تحت فاصله همینگ وزن‌دار نوع جمعی با اصلاح بردار وزن نوع تنگنا برای اولین بار در نظر گرفته می‌شود و الگوریتمی برای حل آن ارایه می‌شود. سازمان‌دهی مقاله به‌صورت 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

---

[2] Polynomial
[3] Hamming distance
[4] Bottleneck
[5] Max + Sum Spanning Tree
[6] 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 به‌علاوه یک درخت باشد.

مجموعه درخت‌های فراگیر یک گراف G با نماد $\Gamma(G)$ نشان داده می‌شود. بنابراین هزینه جمعی درخت فراگیر T به‌صورت

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

و وزن تنگنای درخت فراگیر T به‌صورت

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

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 به‌صورت زیر تعریف می‌شود:

$$(D_G)_{i,j} = \begin{cases} \deg(i), & i = j, \\ -۱, & i \neq j,\ (i,j) \in E, \\ ۰, & \text{otherwise.} \end{cases}$$

۲-۲ مساله درخت فراگیر مینیمم کلاسیک

تعریف ٤. [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. به‌عبارت 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[7] است که دارای زمان اجرای $O(m\log n)$ می‌باشد. این الگوریتم کشف رئوس را از یک رأس دلخواه مانند r شروع می‌کند و تا زمانی ادامه می‌یابد که تمام رئوس کشف شود. نحوه ی کشف رئوس بدین صورت است که در هر تکرار از الگوریتم یک یال پیمایش می‌شود، به‌طوری‌که این یال درخت را به یک رأس کشف نشده وصل می‌کند.

۲-۳ 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 می‌باشد. به‌عبارت دیگر هدف حل مدل بهینه‌سازی

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

است. برای پیدا کردن درخت فراگیر تنگنا ابتدا کل زیردرخت‌های فراگیر گراف Gرا یافته و سپس ماکزیمم وزن هر درخت فراگیر را به‌دست آورید. درخت متناظر با کمترین مقدار ماکزیمم وزن را درخت فراگیر تنگنا می‌نامند.

۲-۴ 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 می‌شود.

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

$$\begin{array}{ll} \min\limits_{T\in\Gamma(G)} & \left\{ \max\limits_{e\in T} w(e) + \sum\limits_{e\in T} c(e) \right\} \\ \text{s.t.} & T\in\Gamma(G). \end{array}$$

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

- $\text{maketree}(v)$: درخت جدیدی ایجاد می‌کند که شامل رأس v بوده و قبلاً در هیچ درختی نبوده است.
- $\text{findroot}(v)$: ریشه درختی را برمی‌گرداند که شامل رأس v باشد.
- $\text{findmaxcost}(v)$: یالی با بالاترین مقدار هزینه روی مسیری از v به ریشه فعلی را برمی‌گرداند.

[7] Prim

[DOR: 20.1001.1.22517286.2022.19.2.5.7]
[DOI: 10.52547/jamlu.19.2.77]

- $\text{evert}(v)$: درخت شامل رأس v را در v ریشه‌دار می‌کند.
- $\text{link}(v,w)$: دو درخت مجزا شامل رأس v و w را با افزودن یال (v,w) ترکیب می‌کند و رأس v را به‌عنوان ریشه درخت در نظر می‌گیرد.
- $\text{cut}(v)$: درخت شامل رأس v (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

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

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

گام ۲. یال‌های $E(w^0)$ را به‌صورت $e_1, e_2, \ldots, e_r$ به‌ترتیب صعودی برحسب $c(e)$ها اندیس‌گذاری مجدد کنید که در آن

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

گام ۳. یال‌های $e\in\left[E-E(w^0)\right]$ را به‌ترتیب صعودی بر حسب $w(e)$ها به‌صورت $e_{r+1}, e_{r+2}, \ldots, e_m$ اندیس‌گذاری مجدد کنید.

گام ۴. به‌ازای هر $v\in V$، $\text{maketree}(v)$ را اجرا کنید.

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

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

فرض کنید e_i دارای رأس‌های انتهایی x و y باشد.

اگر $\text{findroot}(x)=\text{findroot}(y)$، آن‌گاه قرار دهید $i=i+1$، در غیر این‌صورت اجرا کنید:

$\text{link}(x,y)$

قرار دهید $T=T\cup\{e_i\}$.

گام ۷. قرار دهید

$$\begin{aligned}
&\text{best-tree-info}=w^0,\\
&\text{cost}-\text{tree}=\sum_{e\in T}c(e),\\
&\text{best}-\text{obj}=w^0+(\text{cost}-\text{tree}).
\end{aligned}$$

گام ۸. به‌ازای $i=1,2,\ldots,m-r$ 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 $\text{evert}(v)$ را اجرا کنید.

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

[DOR: 20.1001.1.22517286.2022.19.2.5.7]
[DOI: 10.52547/jamlu.19.2.77]

اگر $c(e) \succ c(e_{r+i})$،آن‌گاه 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 $cost-tree + w(e_{r+i}) \prec best-obj$، آن‌گاه 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 به‌دست آمده را برابر T^* قرار دهید.

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

پایان

لازم به‌ذکر 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 می‌شود [۱۷,۱۶]. به‌علاوه مرتب‌سازی هزینه‌ها و وزن‌ها نیز در زمان

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

انجام می‌شود[۱۸] در نهایت یافتن درخت فراگیر مینیمم در زمان $O(m \log n)$ انجام می‌شود. 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 وزن‌دار نوع جمعی با اصلاح بردار وزن از نوع تنگنا

در این بخش مساله MSST معکوس تحت فاصله همینگ وزن‌دار نوع جمعی با اصلاح بردار وزن نوع تنگنا $(IMSST_{SH})$[۸] بیان و فرمول‌بندی می‌شود و سپس به بررسی خواص مساله MSST معکوس تحت فاصله همینگ پرداخته می‌شود.

[8] Invers Max + Sum Spanning Tree

[DOR: 20.1001.1.22517286.2022.19.2.5.7]

[DOI: 10.52547/jamlu.19.2.77]

۳-۱ بیان مساله و فرمول‌بندی ریاضی

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

$$H(\bar{w}(e),w(e))=\begin{cases} ۱, & \bar{w}(e)\neq w(e), \\ ۰, & \text{otherwise.} \end{cases}$$

در این‌صورت مساله $IMSST_{SH}$ را می‌توان به‌صورت

$$\begin{aligned} \min \quad & \sum_{e\in E} q(e)H(\bar{w}(e),w(e)) \\ \text{s.t.} \quad & \bar{W}^b(T_0)+C^s(T_0)\le \bar{W}^b(T)+C^s(T) \qquad \forall T\in\Gamma(G), \\ & w(e)-l(e)\le \bar{w}\le w(e)+u(e) \qquad \forall e\in E, \end{aligned} \tag{۱}$$

فرمول‌بندی کرد که در آن $q(e)\ge ۱$ هزینه تغییرات به‌ازای هر $e\in E$ و همچنین $l(e),u(e)\ge ۰$ دامنه تغییرات است. باید l به‌گونه‌ای انتخاب شود که به‌ازای 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 می‌توان به‌صورت

$$\begin{aligned} \min \quad & \sum_{i\in|\Gamma|} q(e)H(\alpha'(e_i),۰) \\ \text{s.t.} \quad & W^b(T_0)+C^s(T_0)+\alpha'(e_0)\le W^b(T_i)+C^s(T_i)+\alpha'(e_i), \qquad \forall T_i\in\Gamma(G),\ i=۱,\dots |\Gamma|, \\ & -l(e_i)\le \alpha'(e_i)\le +u(e_i), \qquad \forall i=۰,۱,\dots |\Gamma|, \end{aligned} \tag{۲}$$

بازنویسی کرد.

قضیه ٦. مساله $IMSST_{SH}$ یک جواب بهینه دارد، به‌طوری‌که

$$\begin{cases} \alpha'(e_0)\le ۰, & e_0\in T_0, \\ \alpha'(e_i)\ge ۰, & e_i\notin T_0. \end{cases}$$

به‌عبارت دیگر می‌توان یک جواب بهینه را با کاهش ماکزیمم وزن یال‌های درختی و با افزایش ماکزیمم وزن یال‌های غیردرختی به‌دست آورد.

برهان . فرض کنید α'_1 جواب بهینه مدل (۲) باشد. برای یال‌های درختی دو حالت زیر را در نظر بگیرید:
حالت اول:

$$0 \prec \alpha'_1\left(e_{0_j}\right) \leq u\left(e_j\right), \quad e_{0_j} \in T_0.$$

حالت دوم:

$$-l(e) \leq \alpha'_1(e_0) \leq 0, \quad e_0 \in T_0.$$

و برای تمام یال‌های غیردرختی

$$0 \leq \alpha'_1(e_i) \leq u(e_i).$$

در این‌صورت

$$\begin{cases} \alpha'_2\left(e_{0_j}\right) = -\alpha'_1\left(e_{0_j}\right), & e_{0_j} \in T_0,\ 0 \leq \alpha'_1\left(e_{0_j}\right) \leq u\left(e_{0_j}\right), \\ \alpha'_2(e_0) = \alpha'_1(e_0), & e_0 \in T_0, \\ \alpha'_2(e_i) = \alpha'_1(e_i), & e_i \in T_i. \end{cases}$$

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

$$\begin{aligned} & W^b(T_0) + C^s(e_0) + \alpha'_2(e_0) = W^b(T_0) + C^s(T_0) + \alpha'_1(e_0) \\ & \leq W^b(T) + C^s(T) + \alpha'_1(e_i) = W^b(T) + C^s(T) + \alpha'_2(e_i). \end{aligned} \tag{۳}$$

اما اگر در حالت اول صدق کند، می‌توان نوشت:

$$\begin{aligned} & W^b(T_0) + C^s(T_0) + \alpha'_2\left(e_{0_j}\right) = W^b(T_0) + C^s(T_0) - \alpha'_1\left(e_{0_j}\right) \\ & = W^b(T_0) + C^s(T_0) + \alpha'_1(e_0) \leq W^b(T) + C^s(T) + \alpha'_1(e_i) \\ & = W^b(e_i) + C^s(T) + \alpha'_2(e_i). \end{aligned} \tag{۴}$$

با توجه به روابط (۳) و (۴) می‌توان نتیجه گرفت که α_2 یک جواب شدنی برای مساله (۲) است. همچنین بدیهی است که مقادیر عددی α'_1 و α'_2 یکسان هستند. از این رو α'_2 جواب بهینه مساله (۲) نیز است. این استدلال را می‌توان به‌طور مشابه برای یال‌های غیردرختی نیز به کار برد. □

محدودیت‌های مدل (۲) را می‌توان به‌صورت

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

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 به‌صورت

$$\beta(e)=\begin{cases} -\alpha(e_0), & e_0\in W^b(T_0),\\ \\ \alpha(e_i), & e_i\in W^b(T_i),\ i=۱,\dots|\Gamma|. \end{cases} \tag{۶}$$

تعریف کنیم، آن‌گاه با استفاده از (۵) و قضیه ۵ و (۶) مدل $IMSST_{SH}$ را می‌توان به‌صورت

$$\begin{aligned}
\min \quad & \sum_{e\in E} q(e)H(\beta(e),۰) \\
\text{s.t.} \quad & \beta(e_i)-\beta(e_0)\ge f(w,c,T_0)-f(w,c,T), \quad \forall T\in\Gamma(G),\ i=۱,\dots|\Gamma|,\\
& ۰\le\beta(e_0)\le l(e), \qquad e_0\in T_0,\\
& ۰\le\beta(e_i)\le u(e), \qquad e_i\in T_i,\ i=۱,\dots|\Gamma|.
\end{aligned} \tag{۷}$$

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

$$\tilde{w}(e)=\begin{cases} w(e)-l(e), & e_0\in W^b(T_0),\\ \\ w(e)+u(e), & e_i\in W^b(T_i),\ i=۱,\dots|\Gamma|. \end{cases} \tag{۸}$$

لم زیر شرط شدنی بودن مساله را بیان می‌کند.

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

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

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

برهان. اثبات لم بدیهی است زیرا بنا به تعریف $\tilde{w}(e)$ تمام یال‌ها دارای ماکزیمم تغییرات وزنی هستند و اگر درخت نابهینه T_0 بعد از ماکزیمم تغییرات در وزن یال‌ها بهینه نشود، مساله نشدنی خواهد بود. □

[DOR: 20.1001.1.22517286.2022.19.2.5.7]
[DOI: 10.52547/jamlu.19.2.77]

۲-۳ الگوریتم حل

در این بخش یک الگوریتم برای یافتن جواب بهینه مساله $IMSST_{SH}$ ارایه می‌شود. ابتدا مقادیر $\{q(e) \mid e \in E\}$ به‌صورت صعودی

$$q\left(e_{i_1}\right) \leq \ldots \leq q\left(e_{i_\tau}\right)$$

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 $IMSST_{SH}$ ارایه داد.

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

شروع

گام۱. با استفاده از الگوریتم ۱ زیر درخت بهینه گراف $G(V,E,c,\tilde{w})$ را بیابید و آن را $\tilde{T}$ بنامید.

- اگر $f\left(\tilde{w},c,T_0\right)=f\left(\tilde{w},c,\tilde{T}\right)$، آن‌گاه نمونه شدنی است و به گام ۲ بروید.
- اگر $f\left(\tilde{w},c,T_0\right)\succ f\left(\tilde{w},c,\tilde{T}\right)$، آن‌گاه نمونه نشدنی است و به گام ۶ بروید.

گام۲. قرار دهید $k=۱$ و $Q=\varnothing$، هزینه تغییرات یال‌های e_i که

$$e_i = W^b\left(T_i\right), \qquad i=۰,\ldots,|\Gamma|,$$

را به‌صورت صعودی

$$q\left(e_{i_1}\right) \leq \ldots \leq q\left(e_{i_\tau}\right)$$

مرتب کنید. یال e_i با کوچک‌ترین مقدار q که قبلاً انتخاب‌نشده را انتخاب کرده و قرار دهید $Q = Q \cup e_i$.

گام۳. اگر $e_i \in T_0$، آن‌گاه قرار دهید

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

در غیر این‌صورت قرار دهید

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

گام۴. یک درخت فراگیر ماکس + سام بهینه $\bar{T}$ روی گراف $G\left(V,E,c,\bar{w}_{(k)}\right)$ پیدا کنید.

- اگر $f\left(\bar{w}_{(k)},c,T_0\right)=f\left(\bar{w}_{(k)},c,\bar{T}\right)$، آن‌گاه T_0 یک درخت بهینه بوده و به گام ۵ بروید.
- اگر $f\left(\bar{w}_{(k)},c,T_0\right)\succ f\left(\bar{w}_{(k)},c,\bar{T}\right)$، آن‌گاه قرار دهید $k=k+۱$ و به گام ۲ برگردید.

گام ۵. مقدار بهینه برابر است با

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

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

[DOR: 20.1001.1.22517286.2022.19.2.5.7]
[DOI: 10.52547/jamlu.19.2.77]

$$\beta^*(e)=\begin{cases} l(e), & e\in T_0\cap Q,\\ u(e), & e\notin T_0\cap Q,\\ ۰, & e\notin Q, \end{cases} \qquad (۹)$$

در نتیجه جواب بهینه مدل (۱) به‌صورت

$$\bar{w}^*(e)=\begin{cases} w(e)-l(e), & e\in T_0\cap Q,\\ w(e)+u(e), & e\notin T_0\cap Q,\\ w(e), & e\notin Q. \end{cases} \qquad (۱۰)$$

خواهد بود.

پایان

لازم به‌ذکر 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، آن‌گاه 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.

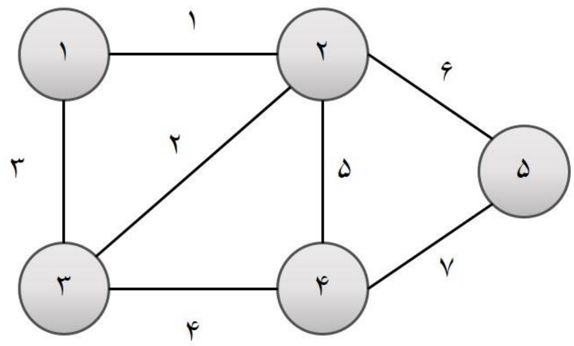
برهان. گام اول الگوریتم ۲ دارای پیچیدگی زمانی $O(m\log n)$ است که در آن یک مساله MSST حل می‌شود [۱۱]. در گام دوم عملیات مرتب‌سازی صورت می‌گیرد که دارای پیچیدگی زمانی $O(c\log c)$ است [۱۹]. توجه داشته باشید که c تعداد دورهای گراف G به اضافه یک است. گام سوم الگوریتم در زمان $O(۱)$ انجام می‌شود. گام چهارم 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

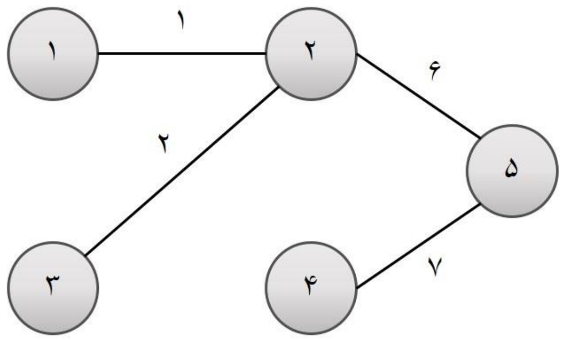
گراف همبند G در شکل ۱ و درخت نابهینه T_0 در شکل ۲ را در نظر بگیرید. اطلاعات مربوط به هر یال گراف G در جدول ۱ آورده شده است. هدف اصلاح بردار وزن $w(e)$ به $\bar{w}(e)$ است، به‌طوری که T_0 به یک درخت فراگیر ماکس + سام بهینه تبدیل شود.

[DOR: 20.1001.1.22517286.2022.19.2.5.7]

[DOI: 10.52547/jamlu.19.2.77]



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 می‌گیرد که در آن

$$\tilde{w} = (۱, ۱, ۷, ۱۳, ۱۷, ۱, ۲),$$

با توجه به مدل (۷) فرمول‌بندی مساله به‌صورت زیر است،

$$\begin{aligned}\min\quad & (۳H(\beta_۱,۰)+۷H(\beta_۲,۰)+۵H(\beta_۳,۰)+۴H(\beta_۴,۰)+۲H(\beta_۵,۰)+۶(\beta_۶,۰)\\ & +۸H(\beta_۷,۰))\\ \text{s.t.}\quad & \beta_۱+\beta_۵\ge ۱\ ,\\ & \beta_۱+\beta_۶\ge ۱\ ,\\ & \beta_۱+\beta_۷\ge -۳\ ,\\ & ۰\le\beta_۱\le ۸\ ,\\ & ۰\le\beta_۲\le ۲\ ,\\ & ۰\le\beta_۳\le ۵\ ,\\ & ۰\le\beta_۴\le ۸\ ,\\ & ۰\le\beta_۵\le ۹\ ,\\ & ۰\le\beta_۶\le ۶\ ,\\ & ۰\le\beta_۷\le ۴\ ,\end{aligned}$$

لیست مرتب‌شده به‌صورت

$$q_{۵_۱}\le q_{۱_۲}\le q_{۶_۳}\le q_{۷_۴}$$

به‌دست می‌آید. یال ۵ به دلیل داشتن مینیمم هزینه تغییرات انتخاب می‌شود و چون یال درختی نیست، وزن آن افزایش می‌یابد:

$$\bar{w}_۵=w_۵+u_۵=۱۷.$$

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 می‌کنیم. فرمول‌بندی جدید مساله با توجه به بردار وزن جدید به‌صورت زیر است

$$\begin{aligned}\min\quad & (۳H(\beta_۱,۰)+۷H(\beta_۲,۰)+۵H(\beta_۳,۰)+۴H(\beta_۴,۰)+۲H(\beta_۵,۰)+۶(\beta_۶,۰)\\ & +۸H(\beta_۷,۰))\\ \text{s.t.}\quad & \beta_۱+\beta_۵\ge -۷\ ,\\ & \beta_۱+\beta_۶\ge ۱\ ,\\ & \beta_۱+\beta_۷\ge -۳\ ,\\ & ۰\le\beta_۱\le ۸\ ,\\ & ۰\le\beta_۲\le ۲\ ,\\ & ۰\le\beta_۳\le ۵\ ,\\ & ۰\le\beta_۴\le ۸\ ,\\ & ۰\le\beta_۵\le ۹\ ,\\ & ۰\le\beta_۶\le ۶\ ,\\ & ۰\le\beta_۷\le ۴\ .\end{aligned}$$

[DOR: 20.1001.1.22517286.2022.19.2.5.7]
[DOI: 10.52547/jamlu.19.2.77]

لیست مرتب‌شده به‌صورت

$$q_{۵_۱} \le q_{۱_۲} \le q_{۶_۳} \le q_{۷_۴}$$

به‌دست می‌آید. یال ۱ انتخاب می‌شود به دلیل این‌که در تکرار قبل یال ۵ انتخاب شده است و چون یال درختی است، وزن آن کاهش می‌یابد:

$$\bar{w}_۱ = w_۱ + l_۱ = ۱.$$

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

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

بنابراین جواب بهینه مدل (۷) به‌صورت

$$\beta^* = \left(۸,۰,۰,۰,۹,۰,۰\right)^T,$$

و جواب بهینه مدل (۱) به‌صورت

$$\bar{w}^* = \left(۱,۳,۲,۵,۱۷,۷,۶\right)^T,$$

به‌دست می‌آید.

٤ نتیجه‌گیری

در این مقاله برای اولین بار، مساله درخت فراگیر ماکس + سام معکوس تحت فاصله همینگ وزن‌دار نوع جمعی با اصلاح بردار وزن نوع تنگنا در نظر گرفته و فرمول‌بندی شد. کاربردهای فراوان این مساله در زندگی واقعی انگیزه‌ای برای بررسی و حل این مساله ایجاد کرد. نشان داده شد که با کاهش ماکزیمم وزن یال‌های درختی و با افزایش ماکزیمم وزن یال‌های غیردرختی می‌توان 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 چندجمله‌ای $O\left(m \log n\right)$ می‌باشد.

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.

[DOR: 20.1001.1.22517286.2022.19.2.5.7]

[DOI: 10.52547/jamlu.19.2.77]