ترجمه مقاله درس الگوریتم پیشرفته
روش تقریبی برای درخت تصمیم دودویی بهینه

محمد مهدی حیدری
۹۴۲۳۰۴۵

۲۴ تیر ۱۳۹۸

چکیده

۱ مقدمه

در این مقاله مسئله تقریب درخت تصمبم دودویی را بررسی می‌کنیم. گری و جانسون مسئله درخت تصمیم را اینگونه تعریف می‌کنند: اگر مجموعه m آزمون دودویی $T = (T_1, ..., T_m)$ و مجموعه n شی $X = (X_1, ..., X_n)$ را به ما داده باشند، خروجی یک درخت دودویی است که در آن هر برگ با یکی از اعضای مجموعه X و هر گره داخلی درخت با یک آزمون از مجموعه T علامت‌گذاری شده است. اگر یک شی به یک آزمون پاسخ مثبت بدهد به شاخه راست گره مربوط به آن آزمون حرکت می‌کند و اگر پاسخ منفی باشد به شاخه چپ می‌رود. یک مسیر از ریشه درخت تا یک برگ به صورت یکتا یک شی با برچسب آن برگ را مشخص می‌کند. عمق یک برگ برابر با طول مسیر آن از ریشه درخت است. طول مسیر کل برای درخت برابر است با مجموع عمق تمام برگ‌های یک درخت. هدف مسئله درخت تصمیم این است که طول مسیر کل برای یک درخت را کمینه کند. یک شیوه معادل برای بیان مسئله این است که هر شی را یک رشته m ببینیم که بیت i ام آن نشان دهنده نتیجه‌ی آزمون i ام روی آن شی است که اگر جواب مثبت باشد بیت ۱ و در غیر این صورت ۰ است. در این مقاله از این شیوه از توصیف درخت تصمیم مثال‌هایی زده شده است که با اعضایی از مجموعه X آنها را نشان می‌دهیم. اگر هیچ دو رشته‌ای در مجموعه X یکسان نباشد، هر راه‌حل ممکن برای مسئله درخت تصمیم n برگ خواهد داشت. در این مقاله فرض می‌کنیم همیشه ورودی یک مجموعه از رشته‌های یکتاست چون پیدا کردن رشته‌های تکراری به راحتی در زمان چند جمله‌ای قابل انجام است. درخت‌های تصمیم کاربردهای طبیعی بسیاری دارند از جمله: تشخیص پزشکی (مجموعه آزمون، علائم بیماری است.)، طراحی آزمایش (آزمون‌ها، همان آزمایش‌هایی هستند که یک ویژگی را تعیین می‌کند.) در واقع هیاویل و رایوست اثبات کردند که مسئله درخت تصمیم NP-Complete است. چون تلاش بسیاری برای ارائه الگوریتمی که درخت تصمیم بهینه را تولید می‌کند انجام شده بود.

در این مقاله یک الگوریتم تقریبی $\ln n + 1$ برای مسئله درخت تصمیم ارائه می‌دهیم. همچنین نشان می‌دهیم که برای مسئله درخت تصمبم الگوریتم تقریبی زمان چند جمله‌ای وجود ندارد مگر اینکه P=NP باشد. با دانش فعلی ما بهترین حد بالا و پایین غیر بدیهی برای تقریب مسئله درخت تصمبم همین دو هستند. با توجه به قدمت زیاد و پرکاربرد بودن این مسئله، اینکه تلاش کمی برای ارائه روش تقریبی برای آن انجام شده به نظر اعجاب آور است. با این حال بررسی دقیق صورت مسئله مقداری توضیحات را ایجاب می‌کند: نام درخت تصمیم به یک مسئله مشابه با اندکی تفاوت نیز اشاره می‌کند که نام آنرا درخت تصمبم سازگار می‌گذاریم. مسئله اخیر برای تقریب بسیار سخت است. ورودی این مسئله n رشته دودویی است که با علامت مثبت و منفی برچسب گذاری شده است، طول هر کدام m است و نمونه‌های مسئله را تشکیل می‌دهد. خروجی یک درخت دودویی است که هر گره داخلی آن بیت i ام از نمونه‌ها را تست می‌کند و نمونه‌هایی که پاسخ ۱ دادند به شاخه راست و نمونه‌های با پاسخ ۰ را به شاخه چپ نگاشت می‌کند. هر برگ یکی از حالت‌های صحیح یا غلط را دارد. یک درخت تصمیم سازگار هر نمونه‌ی با برچسب مثبت را به یک برگ با برچسب صحیح و هر نمونه با برچسب منفی را به یک برگ با برچسب غلط نگاشت می‌کند. اندازه درخت در این حالت تعداد برگ‌ها است و مسئله درخت تصمیم سازگار به دنبال درخت تصمیمی می‌گردد که با کمترین اندازه با نمونه‌ها سازگاری داشته باشد.

آلکنویچ و همکارانش نشان دادند که برای هر ثابت نامنفی k نمی‌توان از طریق درخت تصمیم با اندازه s^k درخت تصمیم با اندازه s را تخمین زد مگر اینکه $\epsilon < 1$ وجود داشته باشد که کلاس NP زیرمجموعه $\text{DTIME}[2^{m^\epsilon}]$ باشد. این موضوع نتیجه کار هانوک و همکارانش را بهبود می‌دهد که نشان می‌داد هیچ تقریب $2^{\log^\delta s}$ برای درخت تصمیم با اندازه s وجود ندارد که $\delta < 1$ مگر اینکه کلاس NP شبه چند جمله‌ای باشد. این نتیجه برای وقتی که اندازه‌ی درخت $\Omega(n)$ باشد صادق است.

نتایج ما نشان می‌دهد که علیرغم ارتباط نزدیک مسئله درخت تصمیم و درخت تصمیم سازگار، این دو مسئله از نظر تقریب پذیری بسیار متفاوت هستند. درخت تصمیم سازگار برای هر ثابت c تقریب $c \ln n$ ندارد مگر اینکه P=NP باشد. این در حالی است که نتایج ما از وجود داشتن چنین تقریبی با ثابت c > 1 برای مسئله درخت تصمبم خبر می‌دهد. همچنین ما نشان می‌دهیم که حد پایین یادگیری درخت تصمیم از نوع سازگار برای وقتی که بخواهیم به جای تعداد برگ‌ها مجموع طول مسیرها را کمینه کنیم نیز برقرار است. لازم به ذکر است که در مسئله درخت تصمیم، اندازه درخت معیار مفیدی نیست چون هر جواب ممکن برای این مسئله n برگ دارد. بنابراین، تفاوت در ورودی و خروجی است که باعث تفاوت در پیچیدگی تقریب این دو مسئله می‌شود و نه معیار.

جای تعجب ندارد که تفاوت در پیچیدگی تقریب بین مسئله درخت تصمیم و درخت تصمیم سازگار به علاوه‌ی ابهام موجود در اسم درخت تصمیم باعث سردرگمی در ادبیات مسئله شده است. برای مثال در ارجا ع دوم مسئله درخت تصمیم با توجه به ورودی و خروجی همان مسئله تعریف شده ولی از نتایج منفی پژوهش هانوک و همکارانش استفاده شده است. بنابر ایم ما یکی از فعالیت‌های خود را جداسازی مسئله درخت تصمیم و درخت تصمیم سازگار از نظر پیچیدگی تقریب می‌دانیم.

مورت هر یک از مسائل درخت تصمیم و درخت تصمیم سازگار را نمونه‌های یکتایی از مسئله عمومی درخت تصمیم می‌داند که در آن هر یک از اعضا با یکی از k برچسب ممکن علامت گذاری شده است. با این فرض در مسئله درخت تصمیم این پارامتر k برابر با n است و هر عضو دقیقا یک برچسب دارد و در مسئله درخت تصمیم سازگار ۲ نوع برچسب داریم که برای هر برچسب می‌تواند چند عضو وجود داشته باشد. پس به نظر می‌آید محدودیت روی تنوع برچسب‌ها نقش اساسی در پیچیدگی تقریب در مسائل درخت تصمیم را دارد.

مسئله درخت تصمیم مشترکاتی با مسئله پوشش مجموعه‌ای (set cover) دارد. چون هر جفت از اعضا در یک درخت تصمیم معتبر دقیقا یک بار از هم جدا می‌شوند، می‌توانیم مسیر از ریشه تا یک برگ را به نوعی پوشش اعضا فرض کنیم. در این حالت هر برگ یک مسئله پوشش مجموعه‌ای را مشخص می‌کند که در آن باید n-1 عضو باقی مانده را با استفاده از مجموعه مناسبی از بیت‌ها یا به عبارتی آزمون‌ها پوشش دهیم. در واقع آنالیز ما از این مشاهده الهام گرفته است. با این حال در مسئله درخت تصمیم، n مجموعه‌ای که توسط برگ‌ها برای پوشش مجموعه‌ای معرفی می‌شوند مستقل نیستند. برای مثال بیتی که در ریشه یک درخت تصمیم دودویی بهینه وجود دارد، در همه‌ی n جواب مسئله پوشش مجموعه‌ای تکرار شده است. ولی به راحتی می‌توان نمونه‌هایی از درخت تصمیم ساخت که برای آن n مجموعه متناظر عضو مشترکی نداشته باشند. به طور دقیق‌تر اگر پاسخ n مسئله پوشش مجموعه‌ای با اندازه ۱ را که از هم مستقل هستند داشته باشیم، در زمان $\Theta(n^2)$ جواب متناظر آن در مسئله درخت تصمیم را پیدا می‌کنیم در حالی که ساخت درخت تصمیم بهینه هزینه‌ی $O(n \log n)$ دارد. در نتیجه فعل و انفعال بین مسائل پوشش مجموعه‌ای منحصر بفرد و مسئله درخت تصمیم، ظاهرا باعث تفاوت بنیادین بین این دو می‌شود.

مسئله پوشش مجموعه‌ای با کمترین مجموع نیز مشابه مسئله درخت تصمیم است. ورودی این مسئله مانند پوشش مجموعه‌ای است (مجموعه جهانی از اعضا X و مجموعه C که هر عضو آن یک زسر مجموعه از X باشد.) ، ولی خروجی یک ترتیب خطی از مجموعه‌های ۱ تا |C| است. اگر f(x) اندیس اولین مجموعه از ترتیب که x را پوشش می‌دهد به ما بدهد، هزینه این ترتیب $\sum_{x \in X} f(x)$ خواهد بود. این هزینه با هزینه‌ی درخت تصمیم متناظر مشابه است ولی یکسان نیست چون اعضای پوشش داده شده باید همچنان از هم جدا شوند و در نتیجه به هزینه افزوده می‌شود. اگر به طور حریصانه مجموعه‌ای را انتخاب کنیم که بیشترین اعضای پوشش داده نشده را پوشش بدهد، به پاسخی تقریبی با فاکتور ۴ از مسئله پوشش مجموعه‌ای با کمترین جمع می‌رسیم. این فاکتور تقریب تنگاتنگ است مگر اینکه P=NP برقرار باشد. مشابه مسئله پوشش مجموعه‌ای، می‌توانیم به درخت تصمیم مانند n نمونه از پوشش مجموعه‌ای با کمترین جمع نگاه کنیم، ولی مجددا این نمونه‌ها مستقل نیستند. پس مشکل ذاتی که برای مسئله پوشش مجموعه‌ای وجود داشت، در مسئله پوشش مجموعه‌ای با کمترین جمع نیز باقی می‌ماند.

در قسمت بعدی الگوریتم تقریبی خود برای درخت تصمیم را توصیف و آنالیز می‌کنیم. همچنین حالتی را که به هر آزمون t وزن داده‌شود نیز ملاحظه و می‌کنیم و نشان می‌دهیم که به فاکتور تقریب $\ln n + 1$ نقصی وارد نمی‌شود. در قسمت سوم نشان می‌دهیم که $\delta > 0$ پیدا می‌شود به طوری که مسئله درخت تصمیم تقریبی با فاکتور $1 + \delta$ نداشته باشد مگر اینکه P=NP باشد. علاوه بر این نشان می‌دهیم که کران پایین برای یادگیری درخت تصمیم سازگار برای مجموع طول مسیرهای خارجی نیز برقرار است. در آخر با بحث روی بعضی مسائل باز که فاصله‌ی بین کران کران بالا و پایین را شامل می‌شوند نتیجه گیری را انجام می‌دهیم.

۲ تقریب مسئله درخت تصمیم

با داشتن مجموعه‌ای از رشته‌های m بیتی به نام S ، انتخاب یک بیت i همواره اعضا را به دو مجموعه S^0 و S^1 تقسیم می‌کند که به ترتیب شامل رشته‌های با بیت i=0 و i=1 هستند. یک روش حریصانه این است که بیت i را طوری انتخاب کنیم که اندازه این دو مجموعه کمترین اختلاف را با هم داشته باشند یا به عبارتی مجموعه S را به متوازن‌ترین حالت ممکن بخش بندی کنند. الگوریتم حریصانه مقابل برای ساخت درخت تصمیم با مجموعه اعضای n عضوی به نام X را ملاحظه کنید:

```
GREEDY-DT(X)
if X = ∅
   then return NIL
   else Let i be the bit which most evenly partitions X into X^0 and X^1
        Let T be a tree node with left child left[T] and right child right[T]
        left[T] ← GREEDY-DT(X^0)
        right[T] ← GREEDY-DT(X^1)
        return T
```

شکل ۱: الگوریتم حریصانه برای ساختن درخت تصمیم

یک پیاده‌سازی سرراست این الگوریتم در زمان $O(mn^2)$. در حالی که این الگوریتم همیشه جواب بهینه را نمی‌دهد، آنرا با فاکتور $\ln n + 1$ تقریب می‌زند.

قضیه ۱ اگر X یک نمونه از درخت تصمیم با n عضو باشد و هزینه بهینه C^* باشد، آنگاه الگوریتم حریصانه درختی با هزینه‌ی حداکثر $(\ln n + 1)C^*$ تولید می‌کند.

اثبات با یک نمادگذاری آغاز می‌کنیم. فرض کنید T درخت تصمیمی با هزینه‌ی C باشد که الگوریتم حریصانه روی مجموعه X ساخته است. یک جفت عضو بدون ترتیب {x, y} (از این به بعد فقط یک جفت عضو) توسط گره داخلی S از هم جدا می‌شوند اگر x در یک شاخه و y در شاخه‌ی دیگر قرار بگیرد. به خاطر داشته باشید که در یک درخت تصمیم معتبر هر جفت عضو دقیقا یک بار از هم جدا می‌شوند. بالعکس هر گره داخلی S مجموعه‌ای به نام $\rho(S)$ تشکیل تعریف می‌کند که اعضای آن جفت‌هایی از اعضا هستند که توسط S از هم جدا شده‌اند. به این صورت:

$$\rho(S) = \{\{x,y\} | \{x,y\} \text{ is separated at } S\}$$

برای راحتی از S برای نشان دادن زیردرخت‌هایی که از S منشعب شده‌اند نیز استفاده می‌کنیم. فرض کنید S^+ و S^- دو فرزند S باشند به طوری که $|S^+| \geq |S^-|$. به یاد داشته باشید که $|S| = |S^+| + |S^-|$. تعداد مجموعه‌هایی که یک عضو به آن تعلق دارد، با طول مسیر آن از ریشه برابر است، پس هزینه T را می‌توان با جمع اندازه‌های هر مجموعه S نشان داد:

$$C = \sum_{S \in T} |S|$$

ما در آنالیز خود از روش بانکداری استفاده می‌کنیم تا هزینه‌ی کل درخت تصمیم حریصانه را بین تمام جفت‌های بدون ترتیبی که معرفی کردیم، پخش کنیم. چون هر مجموعه S به اندازه اندازه خود در هزینه‌ی کل سهیم است، ما سایز آنرا به طور یکنواخت بین $|S^+||S^-|$ جفت‌هایی از اعضا که در S از هم جدا شده‌اند تقسیم می‌کنیم. فرض کنید c_{xy} هزینه‌ای باشد که به هر جفت عضو {x, y} نسبت می‌دهیم که:

$$c_{xy} = \frac{۱}{S_{xy}^+} + \frac{۱}{S_{xy}^-}$$

در جمع سهیم هستند و هر گره y در Z^- به اندازه:

$$\sum_{x \in Z^+} \frac{۱}{|S_{xy} \cap Z^+|}$$

در جمع سهم دارد. برای روشن شدن موضوع، می‌توانیم Z را به عنوان یک گراف دو بخشی کامل ببینیم که Z^+ یک بخش گره‌های آن و Z^- بخش دیگر است. فرض کنید $b_{xy} = \frac{۱}{(|S_{xy} \cap Z^-|)}$ و $b_{yx} = \frac{۱}{(|S_{xy} \cap Z^+|)}$ باشد. می‌توانیم فرض کنیم که هر یال (x, y) که در آن $x \in Z^+$ و $y \in Z^-$ دو هزینه دارد: یکی مربوط به $x(b_{yx})$ و دیگری $y(b_{yx})$. بنابراین، هزینه‌ی کل هزینه‌ی Z حداکثر برابر با جمع تمام هزینه‌های b_{xy} و b_{yx} است. ما می‌توانیم هزینه‌ی کل را در ابتدا با محدود کردن تمام هزینه‌های مربوط به یک گره را محدود کنیم. به طور خاص ما ادعا میکنیم:

ادعا برای هر $x \in Z^+$ داریم:

$$\sum_{y \in Z^-} b_{xy} = \sum_{y \in Z^-} \frac{۱}{|S_{xy} \cap Z^-|} \leq H(|Z^-|)$$

اثبات   اگر Z^- m عضو داشته باشد، آنگاه فرض کنید $(y_1, ..., y_m)$ ترتیبی از Z^- باشد به طوری که هر چه یک عضو در درخت تصمیم حریصانه زودتر از x جدا شده باشد، در این ترتیب دیرتر ظاهر شده باشد (ترتیب معکوس و حالات تساوی به نحو دلخواهی شکسته شده باشد). این بدین معناست که y_1 آخرین و y_m اولین عضوی باشد که از x جدا شده است و به طور کلی y_{m-t+1} t امین عضوی است که از x جدا شده است. هنگامی که y_m از x جدا می‌شود، باید حداقل $|Z^-|$ عضو در S_{xym} وجود داشته باشد. با ترتیبی که ما در نظر گرفتیم اعضای باقی مانده در Z^- باید همچنان حضور داشته باشند پس: $Z^- \subseteq S_{xym}$. بنابراین هزینه‌ای که به گره x بخاطر یال (x, y_m) منسوب می‌شود، حداکثر $\frac{1}{|Z^-|}$ است و در حالت کلی وقتی y_t از x جدا می‌شود، حداقل t عضو از Z^- باقی می‌ماند پس هزینه b_{xyt} که به یال (x, y_t) نسبت داده شده حداکثر $\frac{1}{t}$ می‌شود. این بدین معناست که برای هر $x \in Z_+$:

$$\sum_{y \in Z^-} b_{xy} \leq H(|Z^-|)$$

که ادعا را ثابت می‌کند.

می‌توانیم همین استدلال را برای ادعایی مشابه برای همه‌ی اعضای موجود در Z^- استفاده کنیم. با داشتن این نامساوی‌ها خواهیم داشت:

$$\sum_{\{x,y\} \in \rho(Z)} \frac{۱}{|S_{xy} \cap Z^+|} + \frac{۱}{|S_{xy} \cap Z^-|} \leq |Z^+|H(|Z^-|) + |Z^-|H(|Z^+|)$$

$$< |Z^+|H(|Z|) + |Z^-|H(|Z|)$$

$$< |Z|H(|Z|) \ (since \ |Z^+| + |Z^-| = |Z|)$$

با تعویض این نتیجه با نامساوی ابتدایی، اثبات قضیه کامل می‌شود.

$$\sum_{Z \in T^*} \sum_{\{x,y\} \in \rho(Z)} c_{xy} \leq \sum_{Z \in T^*} |Z|H(|Z|) \leq \sum_{Z \in T^*} |Z|H(n) = H(n)C^* \leq (\ln n + ۱)C^*$$

۱.۲ حالت آزمون‌های وزن دار

در بسیاری از کاربردها ممکن است آزمون‌های مختلف هزینه‌های اجرای مختلفی داشته باشند. برای مثال در طراحی آزمایش یک آزمون تکی ممکن است تمییز دهنده‌ی خوبی برای اعضا باشد، ولی در عین حال پرهزینه باشد. اجرای چند آزمون کم‌هزینه می‌تواند هدف کلی را برآورده کند ولی هزینه‌ی کمتری به همراه داشته باشد. برای مدل کردن چنین سناریوهایی، ما به هر بیت k وزن w(k) را نسبت می‌دهیم و بدون سردرگمی می‌توانیم w(S) را برای بیتی که در گره S استفاده شده به کار ببریم. ما این مسئله را درخت تصمیم با آزمون‌های وزندار می‌نامیم. در بیان ریاضی مسئله اصلی، می‌توانیم فرض کنیم که هر آزمون وزن ۱ دارد پس هزینه معین کردن هر عضو همان طول مسیر از ریشه تا آن عضو می‌باشد. وقتی که آزمون‌ها وزن‌های غیر یکنواخت دارند، هزینه‌ی تعیین یک عضو برابر با جمع وزن‌های آزمون‌ها از ریشه تا آن عضو است که نام آن را هزینه مسیر می‌گذاریم. در نتیجه هزینه‌ی یک درخت تصمیم برابر با جمع هزینه‌ی مسیر هر عضو است. وقتی همه‌ی آزمون‌ها هزینه‌ي یکسانی دارند ما بیتی را انتخاب می‌کنیم که به مساوی‌ترین شکل ممکن اعضا را به دو گروه تفسیم می‌کند. به زبانی دیگر ما هزینه جفتی c_{xy} را کمینه می‌کنیم. با وزن‌های مساوی، هزینه‌ی یک گره داخلی همان اندازه S است در حالی که وقتی وزن‌ها نامساوی باشند، این هزینه |S| w(S) خواهد بود. بنابراین با فرض اینکه S ، x و y را از هم جدا می‌کند، هزینه‌ي جفتی می‌شود:

$$c_{xy} = \frac{w(S)}{|S^+|} + \frac{w(S)}{|S^-|}$$

و الگوریتم حریصانه‌ی ما به طور بازگشتی آن بیتی را انتخاب می‌کند که این مقدار را کمینه کند. این روند منجر به نتیجه‌ای معادل با قضیه ۱ برای درخت تصمیم با آزمون‌های وزندار می‌شود. یک پیاده‌سازی سرراست از این الگوریتم همچنان در زمان $O(mn^2)$ اجرا می‌شود.

قضیه ۲   الگوریتم حریصانه‌ای که به طور بازگشتی بیتی را انتخاب می‌کند که معادله قبل را کمینه کند، به تقریب $\ln n + 1$ مسئله درخت تصمیم با آزمون‌های وزن‌دار منجر می‌شود.

اثبات   با پیروی از قواعد اثباتی که برای قضیه ۱ ارائه شد، به نتیجه‌ی مطلوب می‌رسیم. مشاهده اساسی این است که انتخاب بیتی که معادله را کمینه کند به این نامساوی می‌انجامد:

$$c_{xy} \leq w(Z)(\frac{۱}{|S_{xy} \cap Z^+|} + \frac{۱}{|S_{xy} \cap Z^-|})$$

جون عبارت w(Z) از جمع به بیرون فاکتور گرفته می‌شود،

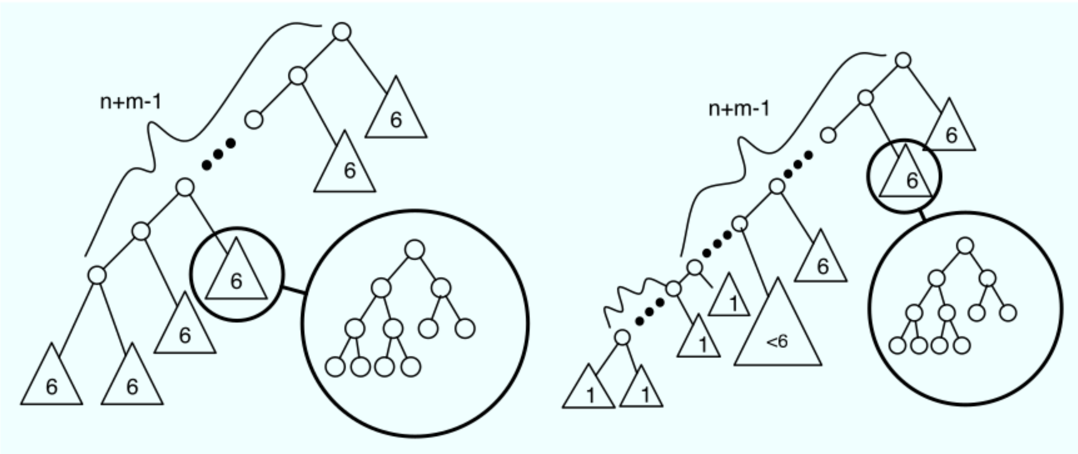
$$w(Z) \sum_{\{x,y\}\in\rho(Z)} c_{xy} \leq \sum_{Z\in T^*} w(Z)|Z|H(n) \leq (\ln n + ۱)C^*$$

می‌توانیم ادعای قبلی را اعمال کنیم و قضیه بدین صورت ادامه می‌یابد:

$$\sum_{Z\in T^*} \sum_{\{x,y\}\in\rho(Z)} c_{xy} \leq \sum_{Z\in T^*} w(Z)|Z|H(n) \leq (\ln n + ۱)C^*$$

در اینجا $C^* = \sum_{Z\in T^*} w(Z)|Z|$ هزینه درخت بهینه است.

یکی دیگر از افزونه‌های طبیعی مسئله درخت تصمیم به حساب آوردن وزن برای اعضا است. در اینجا هر مسیر با عضوی که مسیر به آن تعلق دارد وزندهی می‌شود. متاسفانه آنالیز ما برای این حالت برقرار نیست و در بخش ۵ بیشتر درباره آن بحث می‌کنیم.



شکل ۲: توپولوژی یک درخت بهینه

۳ تقریب درخت تصمیم مسئله‌ای سخت است.