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

محمد مهدی حیدری
۹۴۲۳۰۴۵

۲۵ 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 علامت گذاری 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 DTIME[2^{m^ϵ}] 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 ارجاع 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 $\delta + 1$ 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)
1  if X = ∅
2     then return NIL
3     else Let i be the bit which most evenly partitions X into X^0 and X^1
4          Let T be a tree node with left child left[T] and right child right[T]
5          left[T] ← GREEDY-DT(X^0)
6          right[T] ← GREEDY-DT(X^1)
7          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 یدون 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:

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

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 تقسیم 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:

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

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:

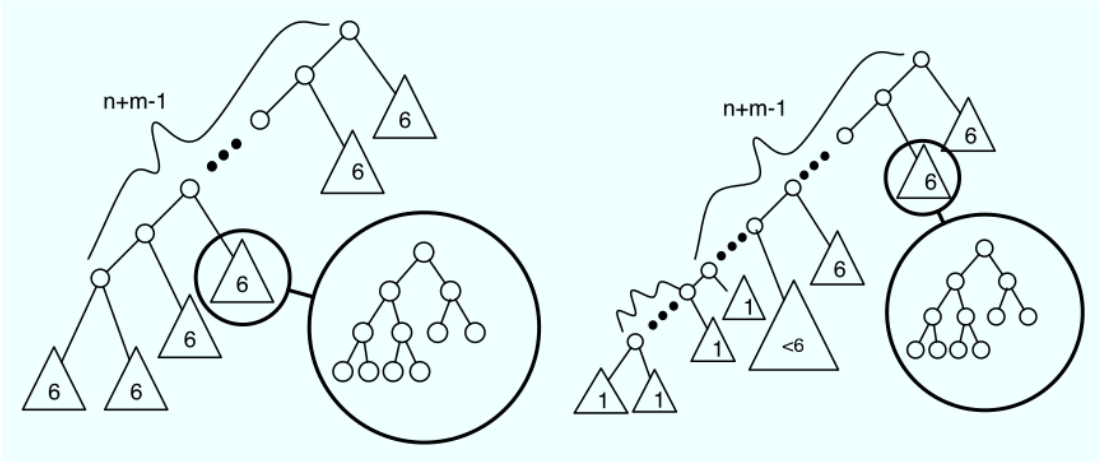
$$\sum_{Z \in T^*} \sum_{\{x,y\} \in \rho(Z)} c_{xy} \leq \sum_{Z \in T^*} w(Z)|Z|H(n) \leq (\ln n + ۱)C^*$$

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.

۳ تقریب درخت تصمیم مسئله‌ای سخت است.

در این بخش نشان می‌دهیم که ثابت جهانی $\delta > 0$ وجود دارد به طوری که درخت تصمیم هیچ تقریبی با فاکتور $(1 + \delta)$ نداشته باشد مگر اینکه، P=NP باشد. این مطلب بدون واسطه ایجاب می‌کند که اگر درخت تصمیم PTAS داشته باشد،



شکل ۲: توپولوژی یک درخت بهینه

آنگاه P=NP خواهد بود. ما مسئله‌ای به نام MAX-3SAT5 را به مسئله درخت تصمیم با حفظ فاصله (gap preserving) تبدیل می‌کنیم که توسط فیج اینگونه تعریف شده است:

ورودی: مجموعه‌ای از n متغیر $X = \{x_1, ..., x_n\}$ و m عبارت منطقی $C = C_1, ..., C_m$ که در آن هر عبارت دقیقا سه حرف دارد (حرف یک متغیر منطقی یا نفی آن است)، هیچ متغیری بیش از یک بار در یک عبارت ظاهر نمی‌شود و هر متغیر دقیقا در ۵ عبارت ظاهر می‌شود. به یاد داشته باشید که $m = \frac{5n}{3}$

خروجی: بیشینه تعداد عبارت‌هایی که با یک سری مقدار معین برای متغیرها می‌تواند خوشنود شود.

فیج نشان می‌دهد که برای $\epsilon > 0$ این مسئله NP-Hard است که بین فرمول‌های 3SAT5 که قابل خوشنود شدن هستند و آنهایی که حداکثر $(1-\epsilon)|C|$ عبارت آنها همزمان خوشنود می‌شود، تفکیک قائل شویم. از این رو ما توجه خود را به نمونه‌هایی جلب می‌کنیم که یا خوشنود پذیر هستند یا آنهایی که با هر مقداری که به متغیرها داده شود حداقل $\epsilon|C|$ عبارت دارند که خوشنود نشده.

ایده این است که 3SAT5 را به مسئله پوشش کاهش بدهیم و سپس آنرا با مسئله درخت تصمیم کاهش دهیم. نوع مسئله پوشش اهمیت دارد: با در نظر گرفتن طول مسیر خارجی کل، هزینه تابعی از عمق برگ‌های درخت است. پس برای کنترل هزینه باید روی عمق برگ‌ها کنترل پیدا کنیم. ما نشان می‌دهیم چگونه می‌توان نمونه‌های مسئله 3SAT5 را به نمونه‌های مسئله پوشش مجموعه‌ای تبدیل کنیم به طوری که هر مجموعه آن اندازه ۶ داشته باشد و هر نمونه خوشنود پذیر 3SAT5 دقیقا یک پوشش داشته باشد و هر نمونه خوشنودی ناپذیر برای پوشش پیدا کردن به تعدادی مجموعه با فاکتور ثابت نیاز داشته باشد.

کاهش مسئله از 3SAT5 به مسئله پوشش مجموعه‌ای با اندازه محدود، از اعمال تغییراتی به دست می‌آید که سیلینگ از آن استفاده کرد تا نشان دهد مسئله MinDT (مسئله‌ای بسیار شبیه ولی متمایز از درخت تصمیم پایدار)، PTAS ندارد.

۱.۳ کاهش مسئله 3SAT5 به پوشش مجموعه‌ای

با داشتن فرمول 3SAT5 به نام ϕ که n متغیر و m عبارت دارد، ما تبدیل زمان چند جمله‌ای $f(\phi)$ را تعریف می‌کنیم که خروجی آن یک نمونه (U,Z) برای مسئله پوشش مجموعه‌ای است، U مجموعه‌ای از اعضا است و Z مجموعه‌ای از زیرمجموعه‌های U است. فرض کنید Y(i) اندیس ۵ عبارتی باشد که در آنها x_i ظاهر شده. در ابتدا مجموعه اعضا را تعریف می‌کنیم. برای هر متغیر x_i ۱۱ عضو درست می‌کنیم: y_i ، a_{ij} (برای هر j در Y(i)) و b_{ij} (مجددا برای هر j در Y(i)). برای هر عبارت C_j این ۳ عضو را به وجود می‌آوریم: c_{j1} ، c_{j2} و c_{j3}. در کل $11n + 3m = 16n$ عضو وجود دارد پس $|U| = 16n$.

اکنون مجموعه‌ها را تعریف می‌کنیم. برای هر متغیر x_i دو مجموعه می‌سازیم:

$$S_i^a = \{y_i\} \cup \{a_{ij} | j \in Y(i)\} \;\; S_i^b = \{y_i\} \cup \{b_{ij} | j \in Y(i)\}$$

اسم این مجموعه‌ها را مجموعه‌های متغیر می‌گذاریم چون از متغیرهای فرمول ساخته شده‌است. برای هر عبارت C_j ، ۷ مجموعه به نام مجموعه‌های عبارت می‌سازیم. هر مجموعه را به صورت مقابل می‌سازیم: فرض کنید x_{u1} ، x_{u2} و x_{u3} متغیرهایی باشند که در عبارت j ظاهر شده‌اند. برای هر انتساب مقدار z(x) که باعث خوشنود شدن محلی می‌شوند (دقیقا ۷ عدد هستند)، این مجموعه را تشکیل می‌دهیم:

$$S_j^z = \{c_{j۱}, c_{j۲}, c_{j۳}\} \cup \{d_۱, d_۲, d_۳\} \;\; where \;\; d_k = b_{ukj} \; if \;\; z(x_{u_k}) = ۱ \;\; else \;\; a_{ukj}$$

برای مثال اگر $C_j = (x_1 \vee \bar{x_2} \vee x_3)$ آنگاه ۸ مقداردهی محلی برای متغیرها وجود دارد. مقداردهی $x_1 = 1, x_2 = 0, x_3 = 0$ عبارت را خوشنود می‌کند پس مجموعه $S_j^{100} = \{c_{j1}, c_{j2}, c_{j3}\} \cup \{b_{1j}, a_{2j}, a_{3j}\}$ شامل می‌شود. با این حال مقداردهی $x_1 = 0, x_2 = 1, x_3 = 0$ نمی‌تواند عبارت را خوشنود کند پس مجموعه $S_j^{010} = \{c_{j1}, c_{j2}, c_{j3}\} \cup \{a_{1j}, b_{2j}, a_{3j}\}$ شامل نمی‌شود. چون برای هر عبارت ۷ مقداردهی وجود دارد که آن را به صورت محلی خوشنود کند، در کل 7m مجموعه در Z وجود دارد. به خاطر داشته باشید که هر مجموعه دقیقا ۶ عضو دارد. کلید تبدیل مسئله این است که a_{ij} مربوط به مقداردهی مثبت به x_i و b_{ij} مربوط به مقداردهی منفی به x_i است. مجموعه‌های عبارت شامل اعضایی هستند که منفی مقداردهی ما هستند، پس وقتی یک فرمول خوشنودشدنی است، همیشه می‌توان آنرا با n+m مجموعه پوشش داد. با این حال وقتی هیچ مقداردهی موجب خوشنود شدن وجود نداشته باشد مجموعه‌های بیشتری نیاز است. در نتیجه به رغم اینکه تبدیل ارائه شده با چیزی که سیلینگ ارائه داده است یکسان نیست قضیه زیر همچنان برقرار است:

قضیه ۳ اگر ϕ یک فرمول خوشنود شدنی 3SAT5 باشد، آنگاه جوابی به اندازه m+n برای $f(\phi)$ وجود دارد. اگر برای هر مقداردهی به ϕ حداکثر $(1-\epsilon)m$ عبارت بتوانند همزمان خوشنود شوند آنگاه اندازه‌ی جواب $f(\phi)$ حداقل $m + n + \frac{\epsilon n}{3}$ خواهد بود.

اثبات واضح است که n+m مجموعه برای پوشش U لازم است ولی وقتی ϕ خوشنود شدنی است، این تعداد کافی نیز هست. اگر z یک مقداردهی باشد که موجب خوشنود شدن ϕ شود، آنگاه برای هر متغیر x_i اگر $z(x_i) = 1$ بود S_i^a را انتخاب کنید و اگر $z(x_i) = 0$ بود S_i^b را انتخاب کنید. برای هر عبارت j ، S_j^z را انتخاب کنید که مجموعه‌ای برای عبارت j است که متناظر مقداردهی z است. واضح است که تمام اعضای y_i و c_j پوشش داده می‌شود. همچنین تمام اعضای a_{ij} نیز پوشش داده می‌شود چون اگر $z(x_i) = 1$ آنگاه S_i^a و اگر $z(x_i) = 0$ آنگاه S_j^z آنرا پوشش می‌دهد. موارد b_{ij} متقارن هستند پس n+m مجموعه کافی است. اگر z منجر به خوشنود شدن فرمول نشود آنگاه مجموعه‌های بیشتری نیاز خواهد بود. برای دیدن این موضوع، فرض کنید Y جوابی برای مسئله پوشش مجموعه‌ای باشد. فرض کنید 'Y زیر مجموعه‌ای از Y باشد به طوری که برای هر i ، Y' شامل اولین رخدادی از یک مجموعه باشد که y_i ، و برای همه‌ی j ها اولین رخدادی از یک مجموعه باشد که c_j را پوشش می‌دهد. مجموعه‌های متغیر از 'Y یک مقداردهی z را برای متغیرها تعریف می‌کنند. با این مقداردهی حداقل ϵm عبارت خوشنود نمی‌شوند. فرض کنید C_j چنین عبارتی باشد، فرضا (x_1, x_2, x_3). چون این عبارت خوشنود نشده است $z(x_i) = 0$ برای $i \in [1, 3]$. پس همه‌ی b_i برای $i \in [1, 3]$ توسط مجموعه‌های متغیرهای پوشش داده شده‌اند. ولی مجموعه عبارت j نمی‌تواند a_i های باقی مانده را پوشش دهد چون توسط ساختن وجود ندارد. برای پوشش دادن اعضای باقی مانده، حداقل یک مجموعه متغیر دیگر لازم است. این پتانسیل پوشش دادن حداکثر ۵ عضو پوشش داده نشده را دارد. پس اگر ϕ خوشنودشدنی نباشد حداقل به $\frac{\epsilon n}{3} = \frac{\epsilon m}{5}$ مجموعه بیشتر نیاز داریم. برای جزئیات بیشتر می‌توانید مرجع ۱۲ را مطالعه کنید. به خاطر داشته باشید وقتی ϕ خوشنود شدنی باشد پوشش مجموعه‌ای پوشش دقیق خواهد بود به این معنا که هیچ عضوی در دو مجموعه وجود ندارد.

اکنون ما یک تبدیل فضا نگهدار به نام g تعریف می‌کنیم که نمونه‌های پوشش مجموعه‌ای U,Z که از تبدیل f (تبدیل قبلی) داده شده به نمونه‌های X درخت تصمیم تبدیل می‌کند. فرض کنید $|U| = n'$ و $|Z| = m'$. برای هر عضو u در U یک رشته دودویی با طول '4m بسازید. 'm بیت اول با 'm مجموعه Z متناظر است پس نام آنرا بیت‌های Z می‌گذاریم.در یک رشته برای عضو u بیت i Z ام برابر با ۱ خواهد بود اگر و فقط اگر u در مجموعه Z_i موجود خواهد بود. به بیانی دیگر، 'm بیت اول نشان‌دهنده عضویت در 'm مجموعه‌ی Z است. '3m بیت آخری به ما اجازه می‌دهد اعضا را از هم تشخیص دهیم. به یاد آورید که هر مجموعه شش عضو دارد پس ما یک بیت Z را در گره T درخت تصمیم استفاده می‌کنیم. سپس حداکثر ۶ عضو ممکن است در شاخه ۱ T به دنبال هم بیایند. می‌خواهیم شکل زیردرختی را که از این اعضا تشکیل می‌شود، کنترل کنیم. یعنی می‌خواهیم تا جای ممکن زیردرخت کامل تشکیل دهیم. بنابراین هر مجموعه Z_i در Z ۳ بیت دیگر به

هر رشته اضافه می‌کند.این بیت‌ها برای همه‌ی اعضایی که در Z_i ظاهر نشده‌اند ۰ هستند. برای آن اعضایی که در Z_i ظاهر شده باشند، بیت‌ها را جوری نسبت می‌دهیم که درخت شبه‌کامل با ارتفاع ۳ همیشه قابل تشکیل باشد. (برای مثال نیمی از اعضا بیت ۱ و نیمی دیگر بیت ۰ دارند و الی آخر). در کنار هم این n' رشته نمونه‌های مسئله درخت تصمیم را دربر دارند. توجه کنید که تعداد اعضا عوض نمی‌شود و تبدیل چه در n' و چه در m' چند جمله‌ای است.

ادعا اگر ϕ یک فرمول 3SAT5 خوشنود شدنی باشد، آنگاه $g(f(\phi))$ دارای درختی با هزینه‌ی حداکثر $\frac{64n^2}{3}+\frac{152n}{3}-6$ است.

اثبات تمام کاری که باید انجام دهیم این است که درخت آبشارگونه‌ای که در شکل ۲ مشاهده کردید، بسازیم. بیت‌های سمت چپ نشان‌دهنده پوشش مجموعه‌ای بهینه است به جز ۳ بیت انتهایی (که برای تمییز دادن مجموعه باقی‌مانده نهایی استفاده شده) . انتخاب $n+m-1=n+\frac{5n}{3}-1$ مجموعه به هزینه زیر منجر می‌شود:

$$C=(-۶)+\sum_{i=۱}^{n+۵/۳n}۶i+۱۶=\frac{۶۴n^۲}{۳}+\frac{۱۵۲n}{۳}-۶$$

در واقع می‌توانیم نشان دهیم که C حداقل هزینه نیز هست. برای درک این موضوع فرض کنید درخت T متناظر با یک پوشش زیر–بهینه (sub-optimal) از اعضا است. این درخت حداقل عمق n+m+3 دارد. چون هر بیت حداکثر ۶ عضو را جدا می‌کند، می‌دانیم T همچنین درختی آبشارگونه است ولی بعضی از زیر درخت‌های آن، کمتر از ۶ عضو دارند. به سادگی می‌توان نشان داد حرکت یک برگ که در قسمت پایین‌تری از درخت قرار دارد به بالا، هزینه‌ی کلی را کاهش می‌دهد. دلیل این اتفاق این است که درخت دودویی کامل جمع طول مسیرهای خارجی را کمینه می‌کند. بنابراین مفید است که تعداد زیر درخت‌های به اندازه ۶ که در ارتفاع‌های بالاتری قرار دارند را بیشتر کنیم. چون درخت متناظر با پوشش مجموعه‌ای بهینه تماما از زیردرخت‌های با اندازه‌ی ۶ تشکیل شده است، کمترین هزینه را دارد. تنها چیزی که باقی مانده تا نشان دهیم این است که g فاصله‌ی موجود در هزینه را وقتی که پوشش مجموعه‌ای حداقل $n+m+\frac{\epsilon n}{3}$ مجموعه نیاز دارد، حفظ می‌کند.

ادعا اگر ϕ یک فرمول 3SAT5 باشد، به طوری که برای هر مقداردهی به آن، حداکثر $(1-\epsilon)|C|$ عبارت همزمان خوشنود شود، آنگاه هر جواب $g(f(\phi))$ حداقل هزینه $C+\frac{n^2\epsilon^2}{18}+\frac{n\epsilon}{6}-1$ دارد.

اثبات از قضیه ۳ می‌دانیم حداقل $\frac{n\epsilon}{3}$ مجموعه بیشتر برای پوشش دادن تمام اعضا لازم است. با دانستن اینکه درخت باید عمق حداقل $n+m+\frac{n\epsilon}{3}-1$ برای n های به اندازه کافی بزرگ داشته باشد و هر بیت می‌تواند حداکثر باعث تمییز ۶ عضو شود، کمترین هزینه برای درختی که این قیود را رعایت کند چقدر است؟ در بهترین حالت هر یک از $\frac{n\epsilon}{3}$ گره داخلی اضافه فقط ۱ عضو را جدا می‌کند و (n+m) گره داخلی دیگر $n'-\frac{n\epsilon}{3}+1$ عضو را جدا می‌کند. در بهترین حالت تا جای ممکن این زیردرخت‌ها اندازه ۶ دارند. این مسئله در قسمت سمت راست شکل ۲ روشن شده است. راه دیگر برای فکر کردن به این درخت این است که با درخت بهینه از ادعای ۱.۳ شروع کنیم و به طور مکرر یک برگ از ارتفاع بالاتر حذف و به ارتفاع پایین‌تر اضافه می‌کنیم. این کار معادل با حرکت دادن یک برگ است. این روند راه مناسبی برای آنالیز هزینه است چون، اولین برگی که حرکت داده می‌شود حداقل ۱ واحد به هزینه اضافه می‌کند، دومین برگ حداقل ۲ واحد و همین طور تا آخر. بنابراین هزینه درخت بهینه برای حالت فاصله (gap) حداقل $C+1+2+...+\frac{n\epsilon}{3}-1=C+\frac{n^2\epsilon^2}{18}+\frac{n\epsilon}{6}-1$ خواهد بود.

با ترکیب ادعای ۱.۳ و ۱.۳ به نتیجه مطلوب می‌رسیم:

قضیه ۴ تنها در حالتی $\delta>0$ وجود دارد به طوری که مسئله درخت تصمیم الگوریتمی با فاکتور تقریب $(1+\delta)$ داشته باشد که P=NP.

اثبات فرض کنید برای هر $\delta>0$ مسئله درخت تصمیم در زمان چند جمله‌ای با فاکتور $(1+\delta)$ تقریب زده شود. فرض کنید $0<\epsilon<1$ داده شده‌باشد، طوری که فاصله‌ای به اندازه ϵ در فرمول‌های خوشنودپذیر و خوشنودنشدنی 3SAT5 وجود داشته باشد. $\delta<\frac{\epsilon^2}{912}$ انتخاب کنید. حال هر نمونه خوشنودپذیر حداکثر هزینه‌ی $(1+\delta)C<C+\frac{n^2\epsilon^2}{18}+\frac{n\epsilon}{6}-1$ برای هر $n>\frac{6}{\epsilon}$ دارد. با ادعای ۱.۳ این یک روند زمان چندجمله‌ای برای تشخیص فرمول‌های خوشنودشدنی 3SAT5 می‌دهد که تناقض است مگر اینکه P=NP.

۲.۳ سختی تقریب درخت سازگار تحت طول مسیر خارجی کلی

آلکنوویچ و همکارانش نشان دادند که ممکن نیست بتوان درخت تصمیم با اندازه k را با درخت تصمیم با اندازه s^k برای هر ثابت $k \geq 0$ تقریب زد مگر اینکه برای یک $\epsilon < 1$ کلاس NP داخل کلاس $DTIME[2^{m^\epsilon}]$ قرار بگیرد. در اینجا درخت تصمیم به نوع سازگار آن اشاره می‌کند و معیار اندازه درخت است. در این بخش نشان می‌دهیم این نتایج سختی برای کمینه کردن طول کلی مسیر خارجی در درخت تصمیم سازگار نیز برقرار است. قضیه ما روی مشاهده‌ای تکیه می‌کند که اگر I یک نمونه از درخت تصمیم پایدار باشد و طول کل مسیر خارجی آن s باشد، آنگاه اندازه‌ی درخت I حداقل $\Omega(\sqrt{s})$ است. در غیر این صورت، درخت با اندازه کمتر طول کلی مسیر خارجی کمتری دارد که تناقض است. حالتی که کمترین طول کلی مسیر خارجی s با کمترین اندازه $\Omega(\sqrt{s})$ متناظر است مربوط به درخت آبشار گونه است. یعنی درختی که در هر عمق یک برگ دارد و در آخرین قسمت دو برگ دارد.

قضیه ۵ اگر یک تقریب s^k به ازای یک ثابت k>0 برای درخت تصمیم با طول مسیر خارجی کل s وجود داشته باشد، آنگاه برای یک $\epsilon < 1$ کلاس NP داخل کلاس $DTIME[2^{m^\epsilon}]$ قرار دارد.

اثبات فرض کنید I یک نمونه از درخت تصمیم سازگار با کمترین طول مسیر خارجی کلی $s = r^2$ باشد. در نتیجه حداقل اندازه درخت $\Omega(r)$ خواهد بود. حال اگر یک تقریب s^k برای یک مقداری از k وجود داشته باشد، آنگاه یک تقریب $\Omega(r^{2k}) = r^{k'}$ به ازای یک ثابت k' برای درخت تصمیم سازگار تحت کمترین اندازه درخت موجود است که این یک تناقض است.

۴ حدود پایین جدید برای تقریب درخت تصمیم کمینه

یک مسئله مشابه درخت تصمیم سازگار، پیدا کردن درخت‌های تصمیم کوچک و معادل است. این مسئله MinDT یا درخت تصمیم کمینه، به عنوان ورودی درخت تصمیم T روی n متغیر را می‌گیرد که از نوع سازگار است(یعنی یک تابع از $\{0,1\}^n$ به $\{0,1\}$) و به دنبال کوچکترین درخت تصمیم T' می‌گردد (کوچکترین مجددا از نظر تعداد برگ) که از نظر عملکرد معادل T باشد. در اینجا عملکرد معادل به این معناست که T و T' یک تابع را روی رشته‌های دودویی به طول n محاسبه می‌کنند. زانتما و بودلاندر ابتدا این مسئله را تعریف و اثبات کردند که NP-hard است. سیلینگ نشان داد این مسئله الگوریتم تقریبی با فاکتور ثابت ندارد مگر اینکه P=NP باشد. این نتایج منفی از یک ویژگی درخت تصمیم سرچشمه می‌گیرد که خودش را بهبود می‌دهد و اگر نتایج مرجع ۷ را مربع کنیم همچنان برقرار می‌ماند. این ویژگی که خودش را بهبود می‌دهد در اینجا بررسی می‌شود، چگونگی تعمیم آن نشان داده می‌شود و سپس آرگومان‌ها و تکنیک‌های مرجع ۷ اعمال می‌شود تا نشان دهیم هیچ الگوریتم تقریبی $2^{\log^\delta s}$ ای وجود ندارد (در اینجا s اندازه درخت بهینه و $\delta < 1$ است)، مگر اینکه $NP \subseteq DTIME[2^{ploylog\ n}]$

لم ۲ فرض کنید f و g توابع دودویی به ترتیب روی متغیرهای دودویی مستقل n و m باشند. اگر $MIN(f \oplus g)$ اندازه کوچکترین درخت تصمیمی باشد که $f \oplus g$ را محاسبه می‌کند، آنگاه $MIN(f \oplus g) = MIN(f).MIN(g)$ خواهد بود. به طور مشابه، اگر T درختی با اندازه‌ی |T| باشد که $f \oplus g$ را محاسبه می‌کند، آنگاه درخت‌های T_f و T_g که به ترتیب f و g را محاسبه می‌کنند، در زمان چندجمله‌ای قابل ساخت هستند به طوری که $|T_f|.|T_g| \leq |T|$ باشد.

اثبات لم۲ T را به ناحیه‌های f و ناحیه‌های g بر اساس بزرگترین بخش‌های پیوسته درخت تقسیم می‌کند که بیت‌های را به طور انحصاری از f یا g بررسی می‌کنند. ایده این است که نواحی نزدیک برگ‌ها ممکن است با والدهای خود عوض شوند و که نواحی پیوسته بزرگتری را شکل دهند تا اینکه T_f و T_g به راحتی قابل تشخیص باشند. تعمیم این نتیجه به جزئیات بیشتری نیاز دارد: