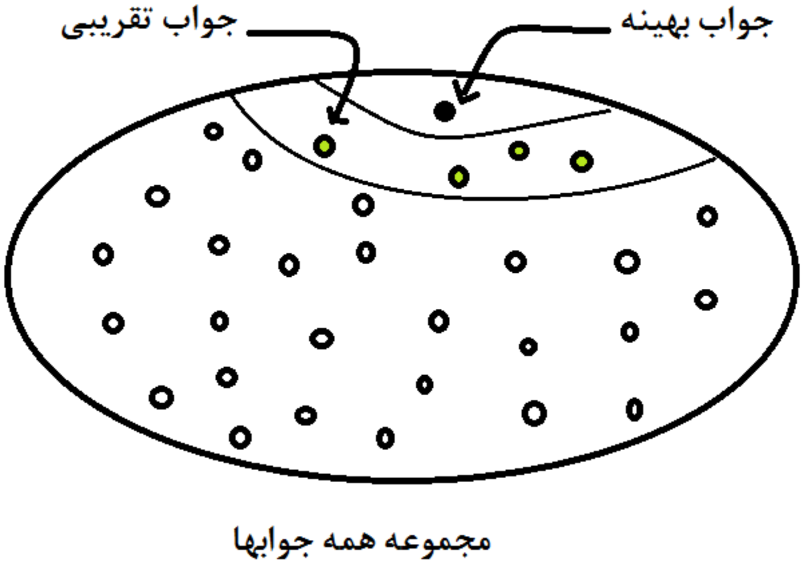
مواجه با سختی مسائل NP

در مواجه با چالشی که مسائل NP-Hard ایجاد کرده‌اند، دو راه پیش رو داریم.

راه اول این است که سعی کنیم الگوریتمهای سریعتر (اگرچه زمان اجرای نمایی هم داشته باشند) برای مسائل NP-Hard پیدا کنیم. برای مثال، الگوریتم بدیهی برای مسئله ۳-SAT که کل مقداردهی های ممکن را امتحان می‌کند زمان اجرایش حداقل $\Omega(2^n)$ دارد. آیا می‌توانیم برای $\alpha < 2$ الگوریتمی بهتر با زمان اجرای α^n برای مسئله ۳-SAT پیدا کنیم؟ جواب این سوال مثبت است و در این درس به معرفی یکی از این الگوریتمها می‌پردازیم.

راه دوم این است که قید جواب دقیق را بزنیم و به جوابی تقریبی برای یک مسئله NP-Hard قناعت کنیم. اینجا باید منظورمان از یک جواب تقریبی روشنتر بیان کنیم. اکثر مسائل NP-Hard در واقع مسائل بهینه سازی هستند. برای مثال در نسخه بهینه سازی مسئله Vertex-Cover می‌خواهیم کمترین تعداد رئوس را انتخاب کنیم که همه یالهای گراف را پوشش دهد. در نسخه بهینه سازی مسئله SAT که با عنوان MAX-SAT شناخته شده است می‌خواهیم یک مقداردهی پیدا کنیم که بیشترین تعداد جمله در فرمول داده شده را ارضا کند. حال یک الگوریتم تقریبی با ضریب تقریب $c \leq 1$ برای مسئله MAX-SAT الگوریتم با زمان چند جمله‌ای است که حداقل $c\beta$ جمله را ارضا کند زمانی که β بیشترین تعداد جملاتی است که صدق پذیر هستند. دقت کنید برای مسائلی که دنبال جواب کمینه هستیم (مثل مسئله Vertex-Cover) ضریب تقریب c عددی بزرگتر از 1 است.



۱ یک الگوریتم بهتر برای مسئله ۳-SAT

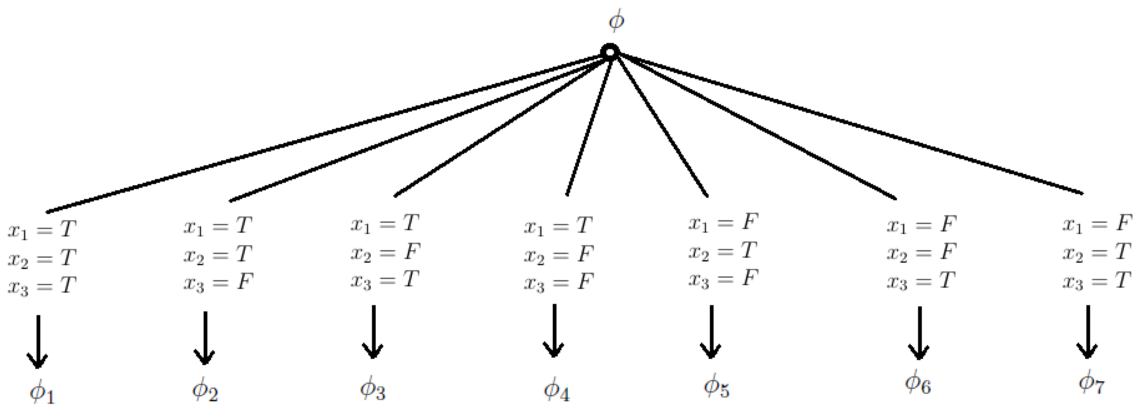
در مسئله SAT یک فرمول ϕ داده شده که در قالب CNF است (ترکیب عطفی از جملاتی که خود ترکیب فصلی از تعدادی لیترال هستند) و پرسش این است که آیا مقداردهی برای متغیرها وجود دارد که همه جملات را تصدیق کند؟

$$(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_5} \vee x_6) \wedge (x_3 \vee \overline{x_6} \vee x_4) \wedge (x_4 \vee x_5 \vee x_6).$$

می‌دانیم که مسئله ۳-SAT (یک نسخه از مسئله SAT است که در آن هر جمله ترکیب فصلی دقیقا ۳ لیترال است) یک مسئله NP-Complete است. امید چندانی نداریم که یک الگوریتم با زمان چند جمله‌ای برای این مسئله پیدا کنیم. الگوریتم بدیهی که همه مقداردهی های ممکن را چک می‌کند زمان اجرایش حداقل 2^n است. در این قسمت به معرفی یک الگوریتم برای مسئله ۳-SAT می‌پردازیم که زمان اجرایش اندکی بهتر از راهبرد بدیهی است. در واقع زمان اجرایش متناسب با 1.913^n است. این الگوریتم با نام الگوریتم ۷ شناخته شده است. وجه تسمیه آن بزودی روشن خواهد شد.

الگوریتم ۷. یک ایده ابتکاری برای تسریع حل مسئله ۳-SAT و بطور کلی SAT این است که یک متغیر را انتخاب کنیم که مقداردهی آن تکلیف تعداد زیادی جمله را مشخص کند (آنها را تصدیق کند) و لذا این جملات را از گود خارج کند. بدین ترتیب به فرمول کوچکتری برسیم که حل آن آسانتر باشد. این ایده را تکرار می‌کنیم و در فرمول باقیمانده یک متغیر خوب را برمی‌داریم و آن را مقداردهی می‌کنیم. کار را ادامه می‌دهیم تا اینکه به جایی برسیم که حل مستقیم فرمول باقیمانده آسان باشد. اگر به سنگ خوردیم شاید لازم باشد برگردیم و مقداردهی های دیگر را امتحان کنیم. این روش کلی به نام DPLL (حروف اول نویسندگان مقاله‌ای در این راستا) شناخته شده است.

الگوریتم ۷ یک نسخه ساده از این تکنیک است که مخصوص مسئله ۳-SAT طراحی شده است. فرض کنید ϕ فرمول داده شده باشد. در این الگوریتم، یکی از جملات فرمول که تصدیق نشده‌اند (در شروع کار هیچ جمله‌ای تصدیق نشده) انتخاب می‌شود. فرض کنید $(x_1 \vee x_2 \vee x_3)$ جمله انتخاب شده باشد. این جمله را به چند طریق می‌توان تصدیق کرد؟ به ۷ طریق زیر (وجه تسمیه الگوریتم!):



الگوریتم هر ۷ مقداردهی بالا را امتحان می‌کند. دقت کنید با مقداردهی $x_i = T$ هر جمله‌ای که x_i بصورت مثبت در آن ظاهر شده تصدیق می‌شود و می‌توان آن را از فرمول حذف کرد. از طرف دیگر، در جمله‌ای که x_i بصورت نقیض (منفی) ظاهر شده باشد، متغیر x_i از آن جمله حذف می‌شود. با انتخاب هر کدام از هفت حالت بالا، یک فرمول متمایز بدست می‌آید. این پروسه با فرمول حاصل بصورت بازگشتی تکرار می‌شود (جمله‌ای سه تایی که تصدیق نشده انتخاب می‌شود و همه ۷ مقداردهی ممکن برای آن امتحان می‌شود). اگر موقعیتی پیش بیاید که فرمول حاصل جمله ۳ تایی نداشته باشد، می‌توان آن را بطور مستقیم در زمان چندجمله‌ای حل کرد (مسئله ۲-SAT بر خلاف ۳-SAT یک الگوریتم با زمان چندجمله‌ای دارد!)

تحلیل زمان اجرای الگوریتم ۷. از آنجا که الگوریتم ۷ یک الگوی بازگشتی را دنبال می‌کند (الگوریتم را برای یک فرمول کوچکتر دوباره فراخوانی می‌شود) برای تحلیل زمان اجرای آن از یک رابطه بازگشتی استفاده می‌کنیم. دقت کنید که با انتخاب یک جمله ۳ تایی تصدیق نشده و انتخاب یک مقداردهی، سه متغیر از فرمول کم می‌شود. الگوریتم هفت مقداردهی مختلف را امتحان می‌کند. پس اگر $T(n)$ زمان اجرا برای فرمولی با n متغیر باشد داریم

$$T(n) \leq 7T(n-3) \leq 7^2T(n-6) \leq \cdots \leq 7^iT(n-3i)$$

همچنین بدیهی است که

$$T(0) = 0$$

پس عمق بازگشت حداکثر $\frac{n}{3}$ است. لذا بدست می‌آید

$$T(n) \leq 7^{n/3} = (7^{1/3})^n \leq 1.913^n$$

دقت کنید، برای سادگی، در تحلیل بالا قدمهایی جزئی در الگوریتم که زمان چندجمله‌ای دارند را از تحلیل حذف کرده‌ایم. در واقع نتیجه دقیقتر این است که

$$T(n) \leq \mathrm{poly}(n) 1.913^n$$

۲ یک الگوریتم تقریبی برای مسئله Min-Vertex-Cover

در مسئله Min-Vertex-Cover گراف $G = (V, E)$ با n راس و m یال داده شده و می‌خواهیم مجموعه رئوس $S \subseteq V$ در این گراف را پیدا کنیم که همه یالهای گراف را پوشش دهد (هر یالی حداقل یک انتهایش در S باشد). علاوه بر این می‌خواهیم S کمترین تعداد رئوس را داشته باشد. روشن است اگر الگوریتمی برای این مسئله داشته باشیم می‌توانیم نسخه تصمیم گیری مسئله Vertex-Cover را در زمان چندجمله‌ای حل کنیم. لذا با فرض $P \neq NP$ این مسئله در زمان چندجمله‌ای قابل حل نیست.

فرض کنید اندازه پوشش راسی بهینه برای گراف G را با $VC(G)$ نمایش دهیم. در این قسمت یک الگوریتم تقریبی برای مسئله Min-Vertex-Cover را معرفی می‌کنیم که در زمان چندجمله‌ای کار می‌کند و همواره یک پوشش راسی برای G پیدا می‌کند که اندازه آن حداکثر ۲ برابر $VC(G)$ است. به عبارت دیگر ضریب تقریب الگوریتم ۲ است.

۱. با استفاده از یک الگوریتم حریصانه، در قدم اول، یک تطابق ماکسیمال در گراف G پیدا کن. فرض کن M تطابق بدست آمده باشد.

۲. $S \leftarrow \emptyset$

۳. برای هر $(u, v) \in M$ دو راس u و v را به S اضافه کن.

۴. مجموعه S را به عنوان جواب گزارش کن.

لم : مجموعه S یک پوشش راسی برای گراف G است. همچنین داریم $|S| \leq 2VC(G)$.

اثبات: برای اثبات گزاره اول، فرض کنید S یک پوشش راسی برای G نباشد. پس یال $e \in E$ وجود دارد که توسط S پوشش داده نشده. اما یال e را می‌توان به مجموعه M اضافه کرد بدون اینکه مشکلی پیش بیاید. در نتیجه M نمی‌تواند یک تطابق ماکسیمال باشد. تناقض.

برای اثبات گزاره دوم، توجه کنید چون M یک تطابق در گراف G است پس حتما $VC(G) \geq |M|$. چون هر پوشش راسی برای G باید حداقل یکی از دو انتهای یالهای داخل M را بردارد. از آنجا که $|S| = 2|M|$ پس نتیجه می‌گیریم $|S| \leq 2VC(G)$. $\square$

این را هم اضافه کنیم که الگوریتمی که ارائه کردیم در زمان چند جمله‌ای کار می‌کند (پیدا کردن یک تطابق ماکسیمال در زمان $O(m)$ قابل انجام است.) لذا نتیجه می‌گیریم که الگوریتم ارائه شده یک الگوریتم تقریبی با ضریب تقریب ۲ برای مسئله Min-Vertex-Cover است.

در نهایت جالب است بدانیم که اثبات شده الگوریتم حریصانه برای مسئله Min-Vertex-Cover (یعنی الگوریتمی که در مرحله راسی را برمی‌دارد که بیشترین تعداد یال پوشش داده نشده را پوشش دهد) ضریب تقریب $\log n$ دارد. علاوه بر این تاکنون الگوریتم با زمان چندجمله‌ای با ضریب تقریب بهتر از ۲ برای مسئله Min-Vertex-Cover پیدا نشده.