تکلیف سری دوم

درس نظریه محاسبه
دانشکده ریاضی. دانشگاه صنعتی خواجه نصیرالدین طوسی. بهار ۱۴۰۲

۱. گراف غیرجهت دار $G = (V, E)$ داده شده است. به هر راس گراف یک وزن مثبت نسبت داده شده است.

$$w : V \to \mathbb{R}^+$$

می‌خواهیم یک مجموعه مستقل با بیشترین وزن در گراف G پیدا کنیم. برای این منظور یک الگوریتم حریصانه را به کار می‌بندیم. الگوریتم هر بار یک راس با بیشترین وزن را برمی‌دارد و آن را به مجموعه مستقل اضافه می‌کند. سپس راس مورد نظر و رئوس مجاور آن را حذف می‌کند و این کار را ادامه می‌دهد تا اینکه راسی باقی نماند.

- فرض کنید S مجموعه مستقلی باشد که الگوریتم حریصانه پیدا می‌کند و T یک مجموعه مستقل دلخواه در G باشد. نشان دهید برای هر $v \in T$ دو حالت وجود دارد. یا داریم $v \in S$ یا یک راس $u \in S$ وجود دارد بطوریکه u مجاور v است و $w(v) \leq w(u)$.

  اثبات با برهان خلف. فرض کنید راس $v \in T$ وجود دارد که در S نیست و راس مجاوری هم در S ندارد. این با سیاست الگوریتم حریصانه در تضاد است. الگوریتم حریصانه v را انتخاب نکرده چون حتما یک همسایه بهتر از آن را قبلا انتخاب کرده است و لذا v حتما باید یک راس مجاور با وزن بیشتر در S داشته باشد.

- فرض کنید Δ بیشترین درجه در G باشد. نشان دهید ضریب تقریب الگوریتم حریصانه $\frac{1}{\Delta}$ است.

  فرض کنید T یک مجموعه مستقل ماکزیمم باشد و S جواب الگوریتم حریصانه باشد. نشان می‌دهیم

  $$\Sigma_{u \in S}\, w(u) \geq \frac{1}{\Delta} \Sigma_{v \in T}\, w(v)$$

  بنا به استدلال قسمت اول، برای هر عضو $v \in T$ می‌توان یک همسایه (یا خودش) در S پیدا کرد که از v سنگینتر است. چون درجه گراف Δ است به هر عضو $u \in S$ حداکثر Δ عضو از T نسبت داده می‌شود که وزنشان از $w(u)$ کمتر است. پس $\Sigma_{v \in T}\, w(v)$ حداکثر Δ برابر $\Sigma_{u \in S}\, w(u)$ است.

۲. نشان دهید اگر $NP = P$ آنگاه مسئله MAX-SAT را می‌توان در زمان چندجمله‌ای حل کرد.

در مسئله MAS-SAT می‌خواهیم یک مقداردهی پیدا کنیم که بیشترین تعداد عبارت را در فرمول ϕ ارضا کند. توجه کنید، نسخه تصمیم گیری مسئله MAX-SAT (آیا برای فرمول داده شده یک مقداردهی وجود دارد که حداقل k عبارت را ارضا کند؟) در NP است. چون NP=P پس الگوریتمی وجود دارد که نسخه تصمیم گیری مسئله MAX-SAT را حل می‌کند. با این الگوریتم می‌توان ماکزیمم تعداد عبارت ارضاشدنی فرمول ϕ را پیدا کرد. برای هر $k = 1, 2, \ldots, \ldots$ سوال مورد نظر را می‌پرسیم و بالاترین k را پیدا می‌کنیم.

فرض کنید $s(\phi)$ ماکزیمم تعداد عبارت ارضاشدنی در فرمول ϕ باشد. با داشتن $s(\phi)$ می‌توان مقداردهی هدف را پیدا کرد. برای این منظور، متغیر دلخواه x_i را انتخاب می‌کنیم و به آن مقدار $True$ می‌دهیم. اگر

با True کردن x_i، فرمول حاصل باز هم به اندازه $s(\phi)$ عبارت ارضاشدنی داشت، مقدار $x_i = True$ را ثابت می‌کنیم و سراغ یک متغیر دیگر می‌رویم. در غیر این صورت $x_i = False$ را ثابت می‌کنیم و سراغ یک متغیر دیگر می‌رویم. با ادامه این روش مقداردهی هدف بدست می‌آید.

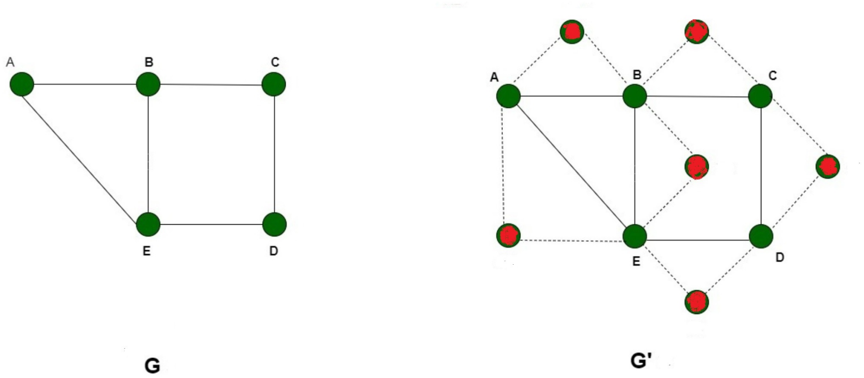
۳. در مسئله Dominating Set گراف ساده $G = (V, E)$ و عدد صحیح k داده شده است. می‌خواهیم بدانیم آیا زیرمجموعه‌ای از رئوس $S \subseteq V$ با اندازه حداکثر k وجود دارد که هر راس $u \in V$ یا درون S باشد یا همسایه‌ای داخل S داشته باشد. نشان دهید که Dominating Set یک مسئله NP-Complete است.

روشن است که Dominating Set عضو کلاس NP است. برای اثبات NP-Complete بودن آن نشان می‌دهیم

$$\text{Vertex Cover} \leq_p \text{Dominating Set}$$

به عبارت دیگر نشان می‌دهیم، اگر یک الگوریتم برای حل مسئله Dominating Set وجود داشته باشد، می‌توانیم با استفاده از آن و یک تقلیل چند جمله‌ای مسئله Vertex Cover را حل کنیم. فرض کنید گراف G و پارامتر k ورودی مسئله Vertex Cover باشند. آیا گراف G یک پوشش راسی با اندازه حداکثر k راس دارد؟

با داشتن $G = (V, E)$، یک گراف جدید می‌سازیم. برای هر یال $e = (u, v)$ در G یک راس x_e اضافه می‌کنیم و x_e را به دو سر یال e، یعنی رئوس u و v وصل می‌کنیم. این کار را برای هر یال گراف G انجام می‌دهیم. نام گراف حاصل را G' می‌گذاریم. شکل زیر یک نمونه از تبدیل G به G' را نشان می‌دهد. رئوس اضافه شده با رنگ قرمز مشخص شده‌اند.



ادعا می‌کنیم

$$|VC(G)| = |DS(G')|$$

ابتدا دقت کنید اگر $S \subseteq V$ یک پوشش راسی برای G باشد، آنگاه S یک مجموعه Dominating برای G' نیز خواهد بود. چون برای هر یال $e = (u, v)$ در گراف G داریم $u \in S$ یا $v \in S$ و یا هر دو. لذا راس اضافه شده x_e هم حتما یک همسایه در S دارد. از این نتیجه می‌شود

$$|DS(G')| \leq |VC(G)|$$

از طرف دیگر فرض کنید S' یک مجموعه Dominating برای G' باشد. نشان می‌دهیم مجموعه $V \supseteq S$ وجود دارد که پوشش راسی برای G است بطوریکه $|S'| = |S|$. مجموعه S' را در نظر

بگیرید. فرض کنید یال $e = (u, v)$ توسط S' پوشش داده نشده است. چون رئوس u و v در S' نیستند و S' یک مجموعه Dominating است پس لزوما x_e باید در S' باشد. در چنین حالتی به جای انتخاب x_e یکی از دو راس u یا v انتخاب می‌کنیم و در S' می‌گذاریم. بدین ترتیب یک مجموعه Dominating برای G' بدست می‌آید که شامل هیچکدام از رئوس اضافه شده (قرمز رنگ) نیست. علاوه بر این مجموعه حاصل یک پوشش راسی برای G است. نتیجه می‌شود

$$|VC(G)| \leq |DS(G')|$$

۴. یک الگوریتم با زمان $O(2^k \text{poly}(n))$ برای مسئله Dominating Set پیشنهاد دهید.

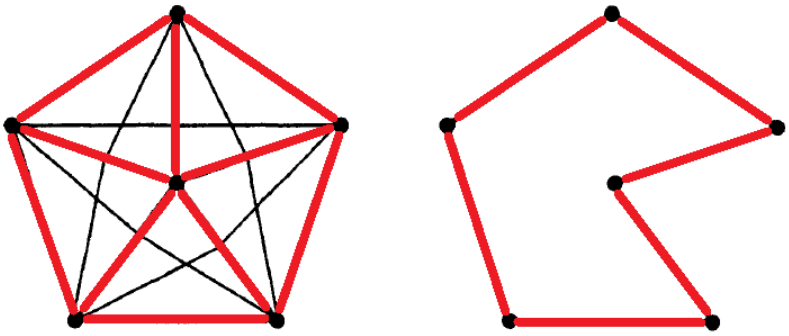
علارغم تشابه با مسئله Vertex Cover چنین الگوریتمی برای مسئله Dominating Set تا کنون پیدا نشده است و اعتقاد بر این است که چنین الگوریتمی وجود ندارد.

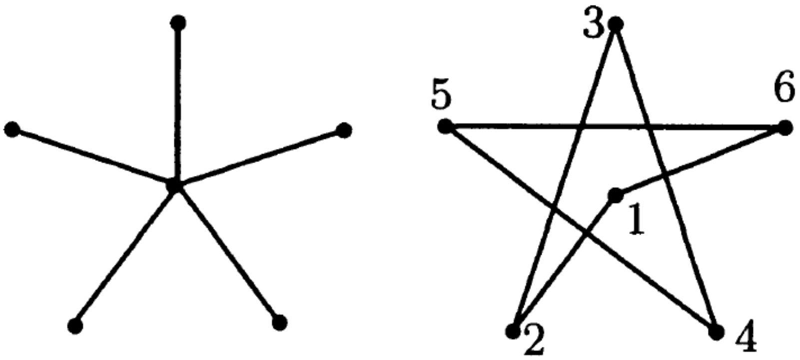
۵. یک الگوریتم با زمان چند جمله‌ای با ضریب تقریب حداقل $\frac{1}{2}$ برای مسئله $MAX-SAT$ پیشنهاد دهید. استدلال کنید که چرا ضریب تقریب الگوریتم شما حداقل $\frac{1}{2}$ است.

فرمول ϕ با n متغیر و m عبارت داده شده است. مقداردهی دلخواه A برای فرمول ϕ را در نظر بگیرید. فرض کنید مقداردهی $\bar{A}$ متمم مقداردهی A باشد. یعنی اگر $A(x) = True$ آنگاه $\bar{A}(x) = False$ و بالعکس. عبارت (جمله) دلخواه C در فرمول ϕ را در نظر بگیرید. روشن است اگر A عبارت C را ارضا نکند، حتما $\bar{A}$ آن را ارضا می‌کند (و بالاعکس.) دقت کنید ممکن است A و $\bar{A}$ هر دو عبارت C را ارضا کنند (اما امکان ندارد هر دو آن را ارضا نکنند.) از این مشاهده ساده نتیجه می‌شود، یکی از دو مقداردهی A یا $\bar{A}$ حداقل نصف عبارات ϕ را ارضا می‌کند. لذا یک الگوریتم ساده با ضریب تقریب $\frac{1}{2}$ برای مسئله MAX-SAT این است که یک مقداردهی دلخواه و متمم آن را بررسی کنیم. یکی از آنها (شاید هر دو) حداقل نصف عبارات را ارضا می‌کند. آن مقداردهی را به عنوان جواب الگوریتم گزارش می‌کنیم.

۶. مثالی از مسئله Metric TSP بزنید که الگوریتم با ضریب تقریب 2 که در کلاس ارائه شد، یک جواب غیر بهینه برای این مثال پیدا کند. آیا می‌توانید مثالی بزنید که هزینه جواب بهینه تقریبا $\frac{1}{2}$ هزینه جواب الگوریتم باشد؟

مثال زیر برگرفته از کتاب الگوریتمهای تقریبی نوشته Vazirani است. یک گراف کامل با n راس را در نظر بگیرید. طول هر یال 1 یا 2 است. لذا فاصله ها خاصیت متریک بودن را دارند. در شکل زیر، بالا سمت چپ، یک نمونه برای $n = 6$ نشان داده شده است. یالهای با رنگ قرمز طول 1 و بقیه یالها طول 2 دارند. اگر n تعداد راسها باشد، طول تور بهینه n است. یک نمونه تور بهینه در قسمت سمت راست نشان داده شده است.

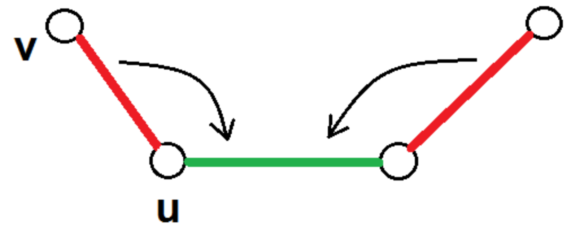




فرض کنید الگوریتم در قدم اول یک درخت فراگیر کمینه به شکل ستاره در گراف پیدا کند. قسمت پایین شکل سمت چپ را ببینید. از این درخت، نهایتا یک تور بدست می‌آید شامل دو یال سبک با وزن 1 و n-۲ یال سنگین با وزن 2. عدد روی رئوس در شکل ترتیب ملاقات راسها را نشان می‌دهد. لذا طول تور بدست آمده $2n-2$ است که تقریبا 2 برابر طول تور بهینه است.

۷. در مسئله Maximum Matching گراف ساده $G=(V,E)$ داده شده است. هدف پیدا کردن بیشترین تعداد یال در G است که اشتراک راسی با هم نداشته باشند. یک الگوریتم حریصانه برای مسئله Maximum Matching می‌تواند بدین صورت باشد. الگوریتم هر بار یک یال دلخواه $e \in E$ از گراف را انتخاب می‌کند و سپس یالهای مجاور e را از گراف حذف می‌کند. این کار دوباره با گراف باقیمانده تکرار می‌شود تا اینکه یالی در گراف باقی نماند. نشان دهید تعداد یالهای انتخاب شده توسط الگوریتم حریصانه حداقل $\frac{1}{2}$ برابر جواب بهینه است.

فرض کنید A تطابق بهینه و B تطابقی باشد که توسط الگوریتم حریصانه بدست آمده است. یال $e=(u,v)$ در مجموعه A را در نظر بگیرید. دو حالت داریم: یال e توسط الگوریتم حریصانه انتخاب شده است. در این حالت $e \in B$. در حالت دیگر e در B حضور ندارد. در این حالت الگوریتم حریصانه حتما یالی دیگری را انتخاب کرده که شامل u یا v است و به همین دلیل e امکان انتخاب پیدا نکرده است (یال سبز رنگ در شکل زیر توسط الگوریتم حریصانه انتخاب شده است.)



لذا برای هر یال بهینه (یال قرمز رنگ)، می‌توان یک یال حریصانه (سبز رنگ) را مسئول دانست. از طرف دیگر، هر یال حریصانه مسئول حداکثر دو یال بهینه است. پس تعداد یالهای حریصانه حداقل نصف تعداد یالهای بهینه است.

۸. نشان دهید اگر در مسئله 3-SAT این محدودیت را اعمال کنیم که هر متغیر حداکثر 4 بار در فرمول ظاهر شود، مسئله NP-Complete باقی خواهد ماند. اما اگر هر متغیر حداکثر 2 بار ظاهر شود مسئله جزو کلاس P خواهد بود.

قسمت اول. فرض کنید متغیر x در k عبارت آمده است. در عبارت اول، اسم متغیر را z_1، در عبارت دوم اسم متغیر را z_2 و به همین ترتیب در هر عبارت یک اسم جدید برای متغیر x قرار می‌دهیم. سپس عبارات زیر را به فرمول اضافه می‌کنیم.

$$(z_1 \to z_2) \wedge (z_2 \to z_3) \wedge \ldots \wedge (z_k \to z_1)$$

که معادل اضافه کردن عبارات زیر است.

$$(\neg z_1 \vee z_2) \wedge (\neg z_2 \vee z_3) \wedge \ldots \wedge (\neg z_k \vee z_1)$$

این کار باعث می‌شود همه متغیرهای z_i، در صورت صدق پذیر بودن فرمول، هم مقدار باشند. در فرمول جدید هر متغیر حداکثر 3 بار ظاهر شده است. اما بعضی از عبارات دو تایی هستند. برای اینکه عبارت $(u \vee v)$ را به سه تایی تبدیل کنیم. کافی است متغیر جدید w را اضافه کنیم و به جای $(u \vee v)$ عبارات $(u \vee v \vee w) \wedge (u \vee v \vee \neg w)$ را قرار دهیم. با این کار هر متغیر حداکثر 4 بار ظاهر می‌شود.

قسمت دوم. فرض کنید ϕ یک فرمول منطقی به شکل CNF باشد. اگر متغیر x فقط یک بار ظاهر شده و یا هر دو باری که ظاهر شده هماهنگ باشند (هر دو مثبت یا هر دو بصورت منفی باشند) در این حالت با مقداردهی x می‌توان عبارات شامل آن را ارضا کرد و آنها را حذف کرد. اما اگر x یک بار بصورت مثبت و بار دیگر بصورت منفی ظاهر شده باشد، برای مثال $(x \vee C) \wedge (\neg x \vee C')$ را داشته باشیم. در این حالت، با خیال راحت می‌توانیم x را حذف کنیم چون $(x \vee C) \wedge (\neg x \vee C')$ معادل با $(C \vee C')$ است. پس هر بار یا یک عبارت را حذف می‌کنیم و یا یک متغیر را حذف می‌کنیم. در نهایت اگر به موردی مانند $(x) \wedge (\neg x)$ رسیدیم نتیجه می‌گیریم که فرمول صدق پذیر نبوده، در غیر اینصورت اگر همه چیز حذف شد، آنگاه فرمول داده شده صدق پذیر است.