مقاله پژوهشی

محاسبه میانگین درجه رنگی گراف‌ها در زمان زیرخطی

محمدعلی آبام و محمدرضا بهرامی

چکیده: گراف‌ها یکی از ساختارهای مهم و پرکاربرد در ذخیره‌سازی داده‌ها هستند. برخی اوقات رئوس گراف‌ها دربردارنده ویژگی‌هایی است که محاسبه میزان اثر آنها بر روی گراف از اهمیت بسزایی برخوردار است. در این نوشتار برخی از این ویژگی‌ها را به کمک رنگ‌ها و درجه رنگی مدل کرده و حل بسیار سریع مسئله را به کمک دو الگوریتم زیرخطی که نیازی به مشاهده همه اطلاعات ندارد، مورد بررسی قرار می‌دهیم. در روش اول فرض می‌کنیم اطلاعات هر رأس از گراف و ویژگی‌های آن را می‌دانیم و سپس یک الگوریتم تقریبی با ضریب $2+\varepsilon$ به ازای ε داده‌شده برای آن ارائه می‌دهیم. سپس در بخش بعدی این فرض را کنار گذاشته و نشان می‌دهیم همچنان می‌توان به چنین تقریبی دست یافت در حالی که امید ریاضی زمان اجرای الگوریتم ارائه‌شده زیرخطی است.

***کلیدواژه:* الگوریتم‌های زمان- زیرخطی، الگوریتم‌های تقریبی، الگوریتم‌های گراف، درجه رنگی، میانگین درجه.**

۱- مقدمه

یکی از مهم‌ترین داده‌ساختارهای مورد استفاده برای ذخیره‌سازی اطلاعات، گراف‌ها هستند. ماهیت گراف‌ها باعث ایجاد ویژگی‌هایی می‌شود که خود در تحلیل داده بسیار مؤثر هستند. امروزه به وفور از این مدل برای نگهداری اطلاعات استفاده می‌شود که این امر سبب شده است ویژگی‌های گراف‌ها در عمل مهم و جذاب باشد. خواص ساده‌ای مانند میانگین درج[1]، همبندی[2]، اندازه پوشش رأسی کمینه[3] و ... اطلاعات ارزشمندی را درباره ساختار گراف فراهم می‌کنند.

با توجه به آنچه ذکر شد اهمیت گراف‌ها و استخراج ویژگی‌های آنها بر کسی پوشیده نیست اما طی سال‌های اخیر و با توجه به رشد روزافزون سرعت تولید و تغییر داده در جهان و نیاز روزافزون به تحلیل اطلاعات و بهره‌بردن از نتایج آنها، نیاز به الگوریتم‌های کارا برای یافتن سریع این ویژگی‌ها بیش از پیش احساس می‌شود.

همواره ارائه یک الگوریتم از مرتبه خطی برای یک مسئله به عنوان یک استاندارد طلایی از موفقیت در حل آن قلمداد شده است. با این حال تغییراتی که به آن اشاره شد، ما را وادار می‌کند تا پارادایم حل مسئله خود را تغییر دهیم، زیرا در بسیاری از موارد حتی یک الگوریتم خطی نیز زمان زیادی را مصرف می‌کند. به طوری که یا این زمان به واقع زیاد است و یا زمانی که محاسبه ما پایان می‌پذیرد، دیگر جواب به دست آمده معتبر نیست و بنابراین در عمل کار بیهوده‌ای انجام داده‌ایم [۱].

حقایق فوق، این سؤال را به وجود می‌آورد که آیا می‌توان تنها به کمک مقدار بسیار کمی از داده‌ها پردازش را انجام داد؟ در صورتی که بخواهیم بخشی از داده‌ها را در نظر نگیریم، ناچار به استفاده از تقریب‌زدن یا عملکرد تصادفی هستیم؛ زیرا گاهی اوقات که شرح آن در بالا گذشت، به دست آوردن یک جواب تقریبی و یا احتمالاتی و در نتیجه غیر دقیق در زمان کوتاه، ارزشمندتر از یافتن جوابی دقیق، اما دیرهنگام است. هر چند گاهی اوقات چنین کاری غیر ممکن یا همراه با محدودیت‌هایی است.

در این نوشتار بر حل یک مسئله از نظریه گراف که می‌تواند جهت حل مسایل دیگری از این دست مورد استفاده قرار بگیرد تمرکز می‌کنیم.

۱-۱ تعریف مسئله و اهمیت آن

هدف ما، محاسبه میانگین درجه رنگی در یک گراف رنگ‌آمیزی شده است. ابتدا به طور رسمی میانگین درجه را تعریف کرده تا مفهوم میانگین درجه رنگی به وضوح برای یک گراف مشخص شود. گرافی را که به هر رأس آن یک رنگ تخصیص داده شده است یک گراف رنگی می‌نامیم.

انگیزه تعریف گراف‌های رنگی بررسی ارتباط بین ویژگی‌های متمایز در یک ساختار می‌باشد. به عنوان مثال شهری را در نظر بگیرید که مراکز مهم آن توسط جاده‌هایی به هم وصل شده‌اند. هر مرکز، ممکن است خدماتی را مثل بیمارستان، آتش‌نشانی و ... ارائه دهد. حال فرض کنید می‌خواهیم بدانیم در یک نقطه حداکثر به چند خدمت مجزا دسترسی داریم. یا مثلاً در یک شبکه اجتماعی افرادی با ویژگی‌ها یا تخصص‌های متمایز، حساب کاربری دارند. حضور آنها سؤالاتی را پیرامون میزان دسترسی و ارتباط افراد عضو به متخصصین مختلف، ایجاد می‌کند. مثال‌هایی از این دست، مثل دسترسی‌های متمایز در یک شبکه برای مسیریاب[4] و یا بررسی تعداد مشاغل مجاور به یک فرد و ... فراوانند.

تعریف ۱) میانگین درجه: در گراف G، میانگین درجه گراف d^* به این صورت تعریف می‌شود

$$d^* = \frac{1}{n}\sum_{v\in V}\deg(v) = \frac{m}{n} \qquad (1)$$

که در آن m تعداد یال‌ها، n تعداد رئوس گراف و همچنین $\deg(v)$ نشان‌دهنده درجه یک رأس می‌باشد. برای تعریف درجه رنگی رئوس یک گراف ابتدا لازم است گراف رنگی را تعریف کنیم. گرافی را رنگی می‌گوییم که به هر رأس آن یک رنگ نسبت داده شده باشد. تعریف ۲ به دقت این موضوع را مشخص می‌کند.

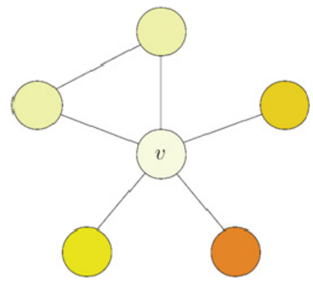
تعریف ۲) تابع رنگ‌آمیزی: تابع $\varphi: S \to V$ را یک تابع رنگ‌آمیزی از مجموعه رئوس V به مجموعه رنگ S می‌نامیم. حال به کمک تعریف

این مقاله در تاریخ ۲۲ اسفند ماه ۱۳۹۸ دریافت و در تاریخ ۲۲ خرداد ماه ۱۴۰۰ بازنگری شد.

محمدعلی آبام (نویسنده مسئول)، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران، (email: abam@sharif.edu).

محمدرضا بهرامی، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران، (email: mrbahrami@ce.sharif.edu).

1. Average Degree
2. Connectivity
3. Minimum Vertex Cover Size
4. Router



شکل ۱: درجه رنگی و میانگین آن. درجه رنگی رأس v برابر ۴ بوده و میانگین درجه رنگی برابر ۱٫۸۳ = ۶/(۱+۲+۱+۲+۱+۴) است.

۲ درجه رنگی را تعریف می‌کنیم.

تعریف ۳) درجه رنگی: برای هر رأس $v \in V$ از گراف G و تابع رنگ‌آمیزی φ، تعداد رنگ‌های متمایز همسایه رأس v را درجه رنگی آن رأس می‌نامیم و به طور دقیق‌تر داریم $\mu(v) = |A|$ که $A = \{c \in S | \exists u \in adj[v], \varphi(u) = c\}$ است. حال می‌توانیم میانگین درجه رنگی را برای یک گراف رنگی تعریف کنیم.

تعریف ٤) میانگین درجه رنگی: میانگین درجه رنگی گراف G با تابع φ بر روی رئوس آن برابر است با

$$\mu^*_{G,\varphi} = \frac{1}{n}\sum_{v \in V}\mu(v) \tag{۲}$$

که به اختصار با μ^* نشان می‌دهیم.

شکل ۱ برای یک گراف رنگی، درجه رنگی و میانگین آن را نمایش می‌دهد. هدف ما محاسبه احتمالی و تقریبی درست (۲) است. در واقع جواب ما باید با احتمال δ داده شده خطایی بیش از ε داده‌شده نداشته باشد.

۲-۱ ابزارها و مدل‌های نمونه‌برداری

هنگامی که با ساختاری مانند گراف مواجه هستیم، این مسئله پیش می‌آید که به چه اطلاعاتی از گراف دسترسی داریم؟ البته این مسئله برای هر مجموعه بزرگی از داده‌ها مطرح است، زیرا با افزایش حجم داده‌ها، افزونگی[1] داده‌ها از دیدگاه غیر تئوری به راحتی قابل چشم‌پوشی نیست. فرض ما بر روی اطلاعاتی که در اختیارمان است علاوه بر محدودکردن کارایی الگوریتم می‌تواند بر روی حل و یافتن حد پایین برای مسئله نیز اثرگذار باشد.

در این بخش فرضیات خود را از مدلی که قصد داریم مسئله را در آن حل نماییم مرور می‌کنیم. این فرضیات شامل نوع دسترسی ما به داده‌ساختارهای لازم برای حل مسئله است.

با توجه به این که عمده تمرکز ما بر روی گراف‌هاست و همچنین همه الگوریتم‌هایی که ارائه خواهیم کرد، به نوعی نیاز به نمونه‌برداری خواهند داشت، در این بخش، نمونه‌برداری از ساختار گراف را مورد بررسی قرار می‌دهیم. پیش از هر چیز لازم است انواع نمونه‌برداری را مورد بررسی قرار داده و سپس فرضیات خود را بیان کنیم.

نوع و پیچیدگی نمونه‌برداری به میزان نسبتاً زیادی به نوع ذخیره‌سازی و داده‌ساختار به کار رفته برای نگهداری اطلاعات بستگی دارد و بنابراین هر پرس‌وجو به صورت ضمنی، پیش‌فرض‌هایی را درباره ساختار ذخیره‌سازی و مسایل مشابه ایجاد می‌کند. پس باید دقت کرد که فرضیات به چه میزان محدودیت ایجاد می‌کنند.

1. Redundancy

نمونه‌برداری رأسی، شاید پایه‌ای‌ترین نمونه‌برداری ممکن باشد که در این نمونه‌برداری، یک رأس به صورت تصادفی انتخاب می‌شود. اغلب اوقات این تصادف به صورت تصادفی یکنواخت[2] (UAR) است و گاهی اوقات به شکل وزن‌دار خطی[3] که البته نوع آخر، نیاز به پیش‌فرض‌های پیچیده‌تری دارد. به هر حال، این نوع از نمونه‌برداری بسیار ساده و در عمده کاربردها امکان‌پذیر است. به طور دقیق‌تر:

تعریف ۵) نمونه‌برداری رأسی:

- تصادفی یکنواخت: $\forall u, v \in V : \Pr[u] = \Pr[v]$
- وزن‌دار خطی: $\forall v \in V : \Pr[v] = \deg(v)/2m$

در اینجا V نشان‌دهنده مجموعه رئوس گراف و $\Pr[v]$ احتمال انتخاب رأس v می‌باشد. البته باید توجه داشت که انجام این نمونه‌برداری، به خصوص در مورد مسایلی که جنبه عملی دارند، همواره ساده نیست [۲].

نوع دیگری از نمونه‌برداری از گراف، نمونه‌برداری یالی است و به این منظور لازم است یال‌ها را در داده‌ساختاری مناسب داشته باشیم. به علاوه باید تعداد آنها را نیز بدانیم. در صورتی که این امکان در گراف فراهم باشد، امکان نمونه‌برداری وزن‌دار خطی نیز برای ما فراهم است. اگر هر یال را به شکل تصادفی یکنواخت انتخاب کنیم و سپس به صورت تصادفی یکنواخت یکی از رئوسش را در نظر بگیریم احتمال انتخاب هر رأس برابر است با

$$\Pr(v) = \sum_{e=1}^{\deg(v)} \frac{1}{2}\Pr(e) = \sum_{e=1}^{\deg(v)} \frac{1}{2m} = \frac{\deg(v)}{2m} \tag{۳}$$

انواع دیگری از نمونه‌برداری، مثل نمونه‌برداری از زیرگراف‌های خاص نیز قابل تعریف است که به علت عدم امکان‌پذیری عملی آنها از تعریفشان چشم‌پوشی می‌کنیم. حال می‌توانیم انواع پرس‌وجوهای مختلف را بر روی گراف‌ها تعریف کنیم. این تعاریف به طور مفصل‌تر در [۳] تا [۵] قابل پیگیری است.

پرس‌وجوی درجه[4]: یکی از متداول‌ترین و اولین پرس‌وجوهای قابل تعریف بر روی گراف است. این پرس‌وجو به این صورت است که هر بار یک رأس را به صورت تصادفی انتخاب کرده و درجه آن رأس را مورد سؤال قرار می‌دهیم.

پرس‌وجوی همسایه[5]: پرس‌وجویی است که طی آن بررسی می‌شود همسایه iام رأس دلخواه v چه رأسی است؟ آیا اصلاً این رأس وجود دارد؟ به کمک $\log n$ پرس‌وجوی همسایه می‌توان پرس‌وجوی درجه را شبیه‌سازی کرد که این پرس‌وجو، طبیعی‌ترین پرس‌وجو در مسایل گراف است.

پرس‌وجوی درجه رنگی[6]: برای رأس مورد پرس‌وجوی $v \in V$ در گراف رنگی $G = (V, E)$، در زمان $O(1)$ مقدار μ^* بازگردانده می‌شود.

پرس‌وجوی رنگ همسایه[7]: این پرس‌وجو می‌گوید که آیا در همسایگی رأس v، یک رأس مانند u که رنگ آن i باشد موجود است یا خیر.

پرس‌وجوی لیست رنگ همسایه[8]: این پرس‌وجو، لیست رنگ‌های مجاور یک رأس v را به ما می‌دهد. این مورد، قوی‌تر از دو پرس‌وجوی

2. Uniformly at Random
3. Linear Weighted
4. Degree Query
5. Neighbor Query
6. Color Degree Query
7. Neighbor Color Query
8. List of Neighbor's Color Query

دیگر است.

این پرس‌وجوها اهمیتی اساسی در نوع و حدود الگوریتم‌هایی که ارائه می‌کنیم دارند و بنابراین لازم است به نوع انتخابمان توجه کنیم.

۱-۳ داده‌ساختارهای گراف‌های رنگی

همواره داده‌ساختار انتخابی، بر زمان اجرا و نوع الگوریتمی که ارائه می‌دهیم، اثرگذار است. علاوه بر نحوه ذخیره‌سازی، اطلاعات کمکی که ممکن است در کنار گرافمان فراهم باشد یا نباشد، می‌تواند اثر بسزایی بر روی زمان اجرا یا حد پایین برای بهترین جواب داشته باشد. بنابراین لازم است نوع اطلاعات و نحوه ذخیره‌سازیشان را نیز مشخص کنیم.

جدای از این که رنگ هر رأس در آن ذخیره شده است، می‌توان فرضیاتی را نیز به مسئله افزود. به طور مثال می‌توان فرض کرد که هر رنگ لیست رئوسی که به آن رنگ هستند را داراست. در واقع افراز رئوس به رنگ‌ها را به طور جداگانه داریم. اطلاع آخر تعبیر جالبی دارد؛ ممکن است قصد داشته باشیم میزان دسترس‌پذیربودن منابع را محاسبه کنیم، در این صورت چنین لیستی در حکم مناطقی است که یک نوع مرکز (مثلاً آتش‌نشانی) می‌تواند به آنها خدمت ارائه کند. با ازدیاد نوع خدمات یا همان رنگ‌ها، چنین اطلاعاتی می‌توانند برخی اوقات بسیار مفید و گاهی هم بی‌اهمیت باشند.

۲- کارهای پیشین

در این بخش مروری داریم بر کارهای مشابه انجام‌شده مرتبط با گراف‌ها و گراف‌های رنگی که در زمان زیرخطی جواب را پیدا می‌کنند. با توجه به آن که به دنبال میانگین درجه رنگی هستیم، اولین رویکرد، نگاه‌کردن به دنباله درجه رئوس گراف است. فرض کنید دنباله‌ای از n عدد نامنفی داریم و قصد داریم مجموع یا میانگین آنها را به دست آوریم. این مسئله کاربردهای زیادی در علوم کامپیوتر و آمار دارد. مسلماً محاسبه دقیق مجموع یک دنباله، هزینه‌ای معادل دیدن همه اعضا دارد و طبیعی است که بخواهیم یک تخمین معقول از مجموع اعضای دنباله را به سرعت به دست آوریم.

به طور سنتی، از روش نمونه‌برداری یکنواخت تصادفی استفاده می‌شود. واضح است که بدون در نظر گرفتن هیچ شرطی، این نوع نمونه‌برداری نمی‌تواند کمک شایانی به حل مسئله بکند. به عنوان مثال دو ورودی متفاوت را در نظر بگیرید: در هر دو ورودی همه عناصر برابر صفر هستند ولی در ورودی دوم، یکی از عناصر برابر عددی بزرگ است. واضح است که با احتمال بالا[1] عنصر یادشده دیده نخواهد شد. بنابراین با نمونه‌برداری، تصادفی یکنواخت و بدون هیچ محدودیتی نمی‌توان مجموع یا میانگین این دو دنباله را از هم تشخیص داد، حال آن که تفاوت چشم‌گیری بین مجموع آن دو وجود دارد. به طور دقیق‌تر قادر نیستیم با خطای ضربی میانگین را محاسبه کنیم. اما با افزودن چند فرض می‌توان جوابی برای مسئله یافت (یک جواب در [۶] آمده است). اگر اعداد را به بازه [۰،۱] محدود کنیم، می‌توان به کمک $O((1/\varepsilon^2)\log(1/\delta))$ نمونه، یک تقریب از میانگین با خطای جمعی ε با احتمال δ به دست آورد که با توجه به داشتن تعداد نمونه‌ها، می‌توان مجموع را نیز محاسبه کرد. نمونه‌برداری یکنواخت و تصادفی بر روی ورودی‌هایی که اعداد، بازه‌های بزرگی را تشکیل می‌دهند، ضعیف عمل می‌کند. اما اگر قادر باشیم از نمونه‌برداری خطی وزن‌دار استفاده کنیم، می‌توانیم به نتایج بهتری دست یابیم. در [۷] به کمک نمونه‌بردای وزن‌دار خطی، الگوریتمی با خطای ضربی و با استفاده از $\tilde{O}(\sqrt{n})$ نمونه ارائه شده که n تعداد اعداد می‌باشد. به طور خلاصه، این روش تبدیل مسئله اصلی به مسئله‌ای است که اعداد به تعدادی سطل خالی یا هم‌اندازه افراز شده‌اند. در حالت جدید، صرفاً باید تعداد سطل‌های خالی را شمارش کنیم و به این منظور از تناقض روز تولد[2] استفاده می‌شود. به این صورت که سعی می‌کنیم، تعداد نمونه‌های لازم را برای آن که به تکرار برخورد کنیم بیابیم. همچنین اگر از نمونه‌برداری تصادفی یکنواخت و خطی وزن‌دار به صورت توأمان استفاده کنیم، می‌توان تعداد نمونه‌های مورد نیاز را به $U=\{v_a,\ldots,u_n\}$ کاهش داد. همچنین در [۷] ثابت شده که تعداد نمونه‌های مورد نیاز برای تخمین مجموع با استفاده از نمونه‌بردای وزن‌دار خطی از مرتبه $\Omega(\sqrt{n})$ است. همچنین با هر ترکیب از نمونه‌بردارهای وزن‌دار عمومی نیز، حتماً به $\Omega(\sqrt[3]{n})$ نمونه نیاز داریم. لازم به ذکر است که نماد $\tilde{O}$ که به نام اوی بزرگ نرم مشهور است به شکل زیر تعریف می‌شود [۸]

$$\tilde{O}(g(n)) = \{f(n) : \exists c,k,n_0 \in \mathbb{N},\ \forall n \in \mathbb{N},\ n > n_0 \Rightarrow 0 < f(n) < cg(n)(\log n)^k\} \quad (۴)$$

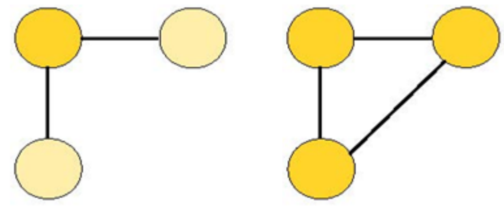
روشن است که به این ترتیب استفاده خاصی از خاصیت گراف‌ها نکرده‌ایم بلکه به دنباله درجات گراف به دید یک دنباله عادی نگریسته‌ایم در حالی که دنباله درجات گراف اطلاعات بیشتری نسبت به یک دنباله معمولی در خود جای داده است. در سال ۲۰۰۴، Fiege به کمک یک قضیه، روشی برای بهره‌بردن از این خاصیت گراف ارائه کرد [۹]. وی الگوریتمی برای محاسبه میانگین درجه گراف ارائه داده و ثابت کرد ضریب تقریب آن $2+\varepsilon$ است. در این الگوریتم به ازای ε داده‌شده، تعدادی رأس از مرتبه $O(\varepsilon^{-1}\sqrt{n})$ به صورت تصادفی یکنواخت انتخاب می‌شوند و میانگین درجه آنها به عنوان میانگین درجه کل گراف گزارش می‌شود. نشان داده شده که این روس با احتمال حداقل ۳/۴ خطای تقریبی $2+\varepsilon$ دارد.

علاوه بر روش فوق، می‌توان از پرس‌وجوی همسایه نیز بهره برد و به نتایج دیگری دست یافت. در [۴] به کمک پرس‌وجوی همسایه با احتمال حداقل ۲/۳ یک جواب $(1+\varepsilon)$– تقریب برای مسئله میانگین درجه ارائه گردیده که امید ریاضی زمان اجرای این الگوریتم $\tilde{O}(\sqrt{n})$ است. همچنین نشان داده شده که زمان اجرای این الگوریتم، بهینه است. در [۴] نشان داده شد که هر الگوریتم با ضریب تقریب ثابت برای مسئله میانگین درجه، نیاز به $\Omega(\sqrt{n})$ پرس‌وجو از یک گراف دارد، حتی اگر امکان استفاده از پرس‌وجوی درجه و پرس‌وجوی همسایه به طور هم‌زمان فراهم باشد. همچنین هر الگوریتم $(1+\varepsilon)$– تقریب نیاز به $\tilde{O}(\sqrt{n}/\varepsilon)$ نمونه دارد. همچنین به کمک پرس‌وجوی همسایه، قادر هستیم که رئوس دسته‌های کم‌جمعیت را که در افزایش درجه رئوس دسته پرجمعیت مؤثر هستند نیز در تخمینمان دخیل کنیم تا به تقریب بهتر دست یابیم. ایده اصلی این الگوریتم از [۱۰] گرفته شده است. نسخه‌ای از الگوریتم که در اینجا ارائه می‌دهیم، به یک حد پایین برای میانگین درجه رئوس نیاز دارد، اما با اندکی تغییر می‌توان این فرض را نیز حذف کرد. این نسخه از الگوریتم در [۱۱] آمده و توضیح مفصل چگونگی حذف این وابستگی به کمک ترفند میانه در [۴] موجود است.

روش دیگر که در [۲] ارائه شده است، با فرض در اختیار داشتن حد بالای $U \ll n$ از میانگین درجه گراف، به کمک $O(\log U \log\log U)$ نمونه، مسئله را حل می‌کند. البته در این راه فرض می‌شود که می‌توان به

1. With High Probability

2. Birthday Paradox



(الف) (ب)

شکل ۲: (الف) گرافی با ۳ رأس، ۳ یال و میانگین درجه ۱ و (ب) گرافی با ۳ رأس، ۲ یال و میانگین درجه ۱.

کمک یک توزیع مفروض نیز از گراف نمونه‌برداری کرد. همچنین نمونه‌برداری وزن‌دار خطی نیز مجاز است.

۳- محاسبه میانگین درجه رنگی گراف

۳-۱ حل به کمک دنباله درجه رنگی

در این بخش یک حل ساده را به کمک قضایای مطرح‌شده در [۶] و [۷] بررسی می‌کنیم. به این بخش از آن جنبه توجه داریم که رنگ‌آمیزی و دنباله درجه رنگی یک گراف تا حدی اطلاعات موجود در آن گراف را از بین می‌برد. یعنی عملاً مسئله را با دید یک تابع کلی گسسته مورد بررسی قرار می‌دهیم.

مسئله محاسبه مجموع دنباله (یا همان میانگین) را برای حالت کلی تعریف کرده بودیم، حال آن که در اینجا حالت خاصی از آن را مشاهده می‌کنیم. در واقع گراف $G=(V,E)$ و تابع φ به کمک یکدیگر دنباله $A=\{a_1,\dots,a_n\}$ را تشکیل داده‌اند که $a_i=\mu(v)$ است. در حقیقت، برخلاف حالت عمومی که تابعی با دامنه n عنصر داریم که روی مجموعه $\{1,\dots,n-1\}$ تعریف شده است، در اینجا همان تابع بر اساس قواعد رنگ‌آمیزی، مجموعه برد $\{1,\dots,l\}$ دارد که l تعداد رنگ‌های مجموعه S (برد تابع φ) می‌باشد. در صورتی که تعداد رنگ‌ها به اندازه تعداد رئوس باشد، ممکن است تصور این باشد که دنباله A، همان دنباله حالت کلی است. نکته جالب آن است که در حالتی که هر رأس رنگ متمایز دارد، دنباله به دست آمده دنباله درجات یک گراف است که برای آن نتایج جالبی در قسمت قبلی ارائه شد.

اگر بخواهیم از راه حل‌های موجود در این قضیه استفاده کنیم، یا باید مسئله را به کمک روشی که در [۶] ارائه گردیده و در قسمت قبل توضیح داده شده است حل کنیم یا با استفاده از نمونه‌برداری وزن‌دار خطی. برای استفاده از حل اول، لازم است اعداد به بازه [۰،۱] منتقل شوند که با توجه به محدودبودن بازه، این امکان وجود دارد و مسئله با خطای جمعی قابل حل است. البته لازم به ذکر است که این خطا با افزایش چشم‌گیری هنگام تبدیل بازه مواجه خواهد شد. در واقع اگر بخواهیم حداکثر $\sqrt{n}$ واحد با عدد واقعی فاصله داشته باشیم، نیاز است تا $\Omega(n)$ نمونه انتخاب کنیم که مورد نظر ما نیست.

اما به کمک نمونه‌برداری وزن‌دار خطی مسئله قابل حل است. می‌توانیم فرض کنیم که نمونه‌برداری برای این گراف در اختیار داریم که رئوس را با احتمال متناسب با درجه‌شان انتخاب می‌کند و با این فرض حل مسئله با کمک روش ارائه‌شده در [۷] امکان‌پذیر است. اما شبیه‌سازی نمونه‌برداری خطی وزن‌دار به کمک نمونه‌برداری تصادفی، جای تأمل دارد. در صورتی که تعداد یال‌های گراف را بدانیم یا تقریبی از آن را داشته باشیم و یا به دست آوریم، می‌توان هر رأس را با توجه به وزنش انتخاب کرد. به طور خلاصه، این رویه به این صورت عمل می‌کند که یک یال به صورت تصادفی انتخاب شده و با احتمال یکسان یکی از دو سرش انتخاب می‌شود. با این وجود ممکن است رأسی که درجه آن از مرتبه $\theta(n)$ است، درجه رنگی ثابتی داشته باشد که به این ترتیب رویه مذکور به تنهایی برای نمونه‌برداری خطی وزن‌دار، بی‌فایده است.

۳-۲ کاهش مسئله میانگین درجه رنگی به میانگین درجه در گراف

قبل از آن که به تبدیل مسئله‌مان بپردازیم، شکل ۲ را مورد بررسی قرار می‌دهیم. در این شکل، دو گراف مشابه با تعداد رئوس برابر به تصویر کشیده شده است. میانگین درجه رنگی در هر دو شکل برابر ۱ است. پیش از ادامه این توضیح، ویژگی مشابه در گراف‌های غیر رنگی را کمی دقیق‌تر بررسی می‌کنیم. در دو گراف ساده با میانگین درجه برابر، هرچند ناصحیح باشد، حاصل‌ضرب میانگین درجه در تعداد رئوس گراف، برابر تعداد یال‌ها خواهد شد که عددی صحیح است. به علاوه دو گراف با میانگین درجه یکسان و تعداد رئوس برابر، تعداد یال‌هایشان برابر است. اما در مورد گراف‌های رنگی این موضوع صحیح نیست. شکل‌های ۲- الف و ۲- ب، دو گراف با دو تابع رنگی متفاوت را نشان می‌دهند که میانگین درجه رنگیشان هم یکی است اما تعداد یال‌هایشان با هم برابر نیست.

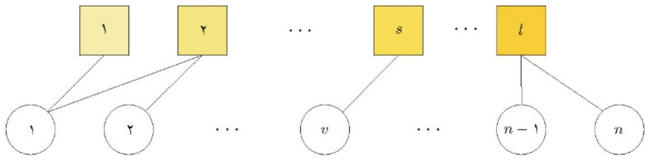
این مسئله ناشی از تأثیر یال‌ها در درجه رنگی هر رأس است. به طور مثال اگر در شکل ۲- الف یال مورب را حذف کنیم، اثری بر روی درجه رنگی هیچ رأسی ندارد و در نتیجه، میانگین درجه رنگی نیز دچار تغییر نمی‌شود. اما تصور کنید در شکل مجاور آن (شکل ۲- ب)، بین همان دو رأس، یال موربی ایجاد کنیم که در این صورت درجه رنگی هر یک از آن دو رأس یکی اضافه می‌شود و در نتیجه ۲/۳ به میانگین درجه گراف افزوده می‌گردد. این مسئله به علت آن است که یک رنگ فقط یک بار بر روی درجه رنگی رأس مجاورش اثر می‌گذارد. بنابراین اگر رأسی (مثل v) با رنگ ۱ مجاور رأسی (مثل u) با رنگ ۲ باشد، در صورتی که رأسی (مثل w) با رنگ ۱ که مجاوری با رنگ ۲ ندارد، مجاور u باشد، درجه رنگی u تغییر نمی‌کند در حالی که درجه رنگی w یک واحد اضافه می‌شود. اما اگر w همسایه‌ای به رنگ ۲ (به جز u) داشته باشد، آن گاه یال (w,u) تأثیری در درجه رنگی نداشته و در نتیجه، در میانگین درجه رنگی نیز بی‌تأثیر است.

در اینجا تبدیلی را مطرح می‌کنیم که مسئله میانگین درجه رنگی را تبدیل به مسئله میانگین درجه گراف کند. گراف ساده $G=(V,E)$ را با مجموعه رئوس $V=\{v_1,\dots,v_n\}$ و مجموعه یال‌های $E=\{e_1,\dots,e_n\}$ و همچنین تابع $\varphi:S\to V$ را با مجموعه $S=\{s_1,\dots,s_n\}$ شامل ۱ رنگ متمایز در نظر بگیرید. همان طور که قبلاً تعریف کردیم، این گراف و تابع φ تشکیل یک گراف رنگی را می‌دهند. حال با توجه به این گراف، گراف $\tilde{G}=(\tilde{V},\tilde{E})$ را تعریف می‌کنیم که مجموعه رئوس این گراف $\tilde{V}=V\bigcup U$ می‌باشد که رئوس مجموعه $V=\{v_1,\dots,v_n\}$ نماینده رئوس اصلی گراف G و رئوس مجموعه $U=\{u_1,\dots,u_n\}$ هر یک نماینده رنگ‌های متمایز مجموعه S می‌باشند. مجموعه یال‌های گراف $\tilde{G}$ را به این صورت تعریف می‌کنیم

$$\begin{aligned}&\forall v\in V, s\in S, (v,s)\in\tilde{E}\\&\Leftrightarrow \exists u\in V, \varphi(u)=s, (u,v)\in E\end{aligned} \qquad (۵)$$

در واقع، در گراف $\tilde{G}$ هر رأس از گراف G به رنگ‌هایی متصل است که در گراف G همسایه‌ای با آن رنگ دارد.

همان طور که در شکل ۳ مشخص است، گراف تبدیل‌یافته یک گراف دوبخشی است که رئوس آن شامل دو بخش رئوس اصلی گراف و



شکل ۳: گراف تبدیل‌یافته $\tilde{G}$ برای درجه‌ی رنگی. در اینجا رأس v در گراف G همسایه‌ای از رنگ s دارد.

ورودی: تعداد رئوس n، تعداد رنگ‌ها l، گراف G، لیست L شامل تعداد رئوس همسایه یک‌رنگ

خروجی: میانگین درجه رنگی گراف G

۱) S را برابر با صفر قرار بده.

۲) به ازای $i=۰$ تا $t=\sqrt{n+l}$:

۳) یک عدد تصادفی a بین ۱ تا $n+l$ انتخاب کن.

۴) اگر $a \leq n$:

۵) درجه v_a از گراف G را به S اضافه کن.

۶) در غیر این صورت:

۷) مقدار $L[a-n]$ راه به S اضافه کن.

۸) $\frac{S}{t} \to S$.

۹) مقدار $\frac{S(n+l)}{n}$ را برگردان.

شکل ۴: الگوریتم ۱- محاسبه میانگین درجه رنگی با اطلاع کامل.

رنگ‌های به کار رفته می‌باشند. در اینجا قصد نداریم گرافمان را به این گراف تبدیل کنیم، زیرا این کار باعث می‌شود هر الگوریتمی از مرتبه $\Omega(n)$ باشد. بلکه این گراف صرفاً مدلی برای انجام تحلیل برخی از ایده‌های مطرح‌شده برای حل این مسئله است.

گزاره ٦: درجه هر رأس $v \in \tilde{V}$ که معادل $v \in V$ است برابر $\mu(v)$ در گراف G است.

اثبات: طبق تعریف گراف $\tilde{G}$ و همین طور (۴)، هر رأس در صورتی که همسایه‌ای از هر رنگ s داشته باشد، به آن متصل می‌شود و بنابراین درجه آن برابر تعداد رنگ‌هایی است که در G با آنها همسایه است که این همان تعریف ۳ می‌باشد.

نتیجه ۷: اگر میانگین درجه رئوس گراف $\tilde{G}$ را با $\tilde{d}^*$ نشان دهیم، میانگین درجه رنگی گراف G با l رنگ، برابر است با

$$\mu = \frac{\tilde{d}^*}{n}(l+n) \qquad (۶)$$

که در واقع همان میانگین درجه رئوس $\{v_۱,\ldots,v_n\}$ در گراف $\tilde{G}$ است. بنابراین اگر بتوانیم میانگین درجه گراف $\tilde{G}$ را به دست آوریم، عملاً توانسته‌ایم میانگین درجه رنگی گراف G را نیز به دست آوریم. مسئله‌ای که در اینجا مطرح است یافتن اطلاعات رئوس $U=\{u_۱,\ldots,u_n\}$ است. در بخش‌های آینده با فرض داشتن و نداشتن این اطلاعات سعی در حل مسئله داریم.

۳-۳ حل مسئله با اطلاع کامل

با توجه به تبدیل صورت‌گرفته، در این بخش قصد داریم مسئله را با این فرض حل کنیم که اطلاعات کامل گراف $\tilde{G}$ را در اختیار داریم. همان طور که پیشتر گفتیم، این مسئله مانند بررسی میزان دسترسی‌پذیری منابع در یک گراف است که هر منبع با رنگی مشخص شده و به عنوان مثال می‌توان از مراکز آتش‌نشانی به عنوان یکی از منابع نام برد. منطقی است فرض کنیم که هر رأس می‌تواند به همه همسایگانش خدمات‌رسانی

ورودی: تعداد رئوس n، تعداد رنگ‌ها l، گراف G

خروجی: میانگین درجه رنگی گراف G

۱) S را برابر با صفر قرار بده.

۲) به ازای $i=۰$ تا $t=\sqrt{n+l}$:

۳) یک عدد تصادفی a بین ۱ تا $n+l$ انتخاب کن.

۴) اگر $a \leq n$:

۵) درجه v_a از گراف G را به S اضافه کن.

۶) در غیر این صورت:

۷) $S += findNeighborCount(G, a-n)$.

۸) $\frac{S}{t} \to S$.

۹) مقدار $\frac{S(n+l)}{n}$ را برگردان.

شکل ۵: الگوریتم ۲- محاسبه میانگین درجه رنگی با اطلاع محدود.

کند. همچنین در هر رأس فقط یک منبع می‌تواند فعالیت کند. به علاوه، می‌توان فرض کرد که هر منبع یا خدمت، لیستی از مراکزی را که می‌تواند به آنها خدمت ارائه دهد داشته باشد که طول این لیست، همان درجه رأس u_i متناظر با آن منبع است. با کمک این فرض عملاً درجه همه رئوس گراف $\tilde{G}$ را داریم. به این ترتیب می‌توان با تقریب $۲+\varepsilon$ این مسئله را در زمان $O(\sqrt{n+l})$ به کمک نمونه‌برداری تصادفی حل کرد. شیوه حل در شکل ۴ نوشته شده است.

به کمک آنچه در بخش ۲ در مورد روش Fiege در [۹] گفتیم، می‌دانیم الگوریتم انتخاب چند رأس و محاسبه میانگین آنها، میانگین درجه گراف $\tilde{G}$ را با تقریب $۲+\varepsilon$ محاسبه می‌کند که این روش تا خط ۸ الگوریتم ۱ نشان داده شده است. سپس در خط آخر الگوریتم میانگین درجه رنگی، به کمک نتیجه ۷ محاسبه شده و خروجی داده می‌شود. به این ترتیب می‌توان یک الگوریتم $۲+\varepsilon$- تقریب را برای این مسئله با فرض در اختیار داشتن اطلاعات همه رئوس ارائه کرد.

نتیجه ۸: با انتخاب $O(\sqrt{n+l})$ رأس و اطلاع از میانگین درجه رنگی هر کدام، می‌توان میانگین درجه رنگی یک گراف را با تقریب $۲+\varepsilon$ محاسبه کرد.

۴-۳ حل مسئله با اطلاع محدود

در اینجا با فرضی محدودتر ولی واقعی‌تر به مسئله نگاه می‌کنیم. فرض می‌کنیم اطلاعات رئوس U را به طور آماده در اختیار نداریم، یعنی نمی‌دانیم که یک رنگ دلخواه در مجاورت چند رأس متمایز قرار دارد. از طرفی می‌دانیم می‌توان این اطلاع را در زمان $O(n)$ در بدترین حالت برای هر رأس به دست آورد. هرچند این زمان خطی بوده و مطلوب ما نیست ولی می‌توان به کمک روش ساده‌ای از آن برای محاسبه میانگین درجه در زمان زیرخطی بهره برد.

در اینجا فرض می‌کنیم تعداد رنگ‌هایمان (l) از مرتبه $O(\log n)$ است، با این تفاوت که پرسیدن درجه یک رأس این بار همواره از مرتبه $O(۱)$ نیست. در واقع اگر یکی از رئوس مجموعه U را انتخاب کنیم، نیازمند انجام کاری از مرتبه $O(n)$ هستیم. این موضوع به طور دقیق‌تر در شکل ۵ آمده است. تابع $findNeighborCount()$ یک گراف و یک رنگ را دریافت کرده و تعداد رئوسی را که همسایه‌ای به آن رنگ دارند بازمی‌گرداند. همان طور که اشاره شد، می‌توان این تابع را به گونه‌ای پیاده کرد که در زمان $O(n)$ کار کند، اما سؤالی که در اینجا قصد داریم به آن پاسخ دهیم، تأثیر این تغییر بر روی زمان اجرای کل الگوریتم است.

تحلیل الگوریتم ۲

در واقع الگوریتم ۲ از یک نوع پرس‌وجو (پرس‌وجوی درجه) با دو رویه متفاوت در خطوط ۵ و ۷ استفاده می‌کند. زمان اجرای رویه خط ۵ از $O(1)$ و رویه خط ۷ از $O(n)$ است. از طرفی احتمال انتخاب یک رأس از مجموعه V که همان رئوس گراف G هستند، $n/(n+l)$ می‌باشد ولی احتمال انتخاب یکی از رئوس مجموعه U برابر $l/(n+l)$ است و بنابراین

$$\begin{aligned}\Pr(a \le n) = \Pr(v_a \in V) = \frac{n}{n+l} = \frac{n}{n+\log n}\\ \Pr(a > n) = \Pr(v_a \in U) = \frac{l}{n+l} = \frac{\log n}{n+\log n}\end{aligned} \tag{7}$$

همان طور که مشخص است تعداد پرس‌وجوها $O(\sqrt{n+l})$ است. اما برخلاف الگوریتم قبل، پیچیدگی الگوریتم تنها با پیچیدگی پرس‌وجوها مرتبط نیست بلکه چون هر پرس‌وجو پیچیدگی زمانی متفاوتی دارد، پیچیدگی الگوریتم با مجموع پیچیدگی پرس‌وجوها مربوط است. با توجه به توزیع احتمال بین دو دسته رئوس که در (۶) دیدیم می‌توان امید زمان اجرای الگوریتم ۲ (در اینجا A) را محاسبه کرد

$$E[TIME(A)] = E[t\,Q(i)] \tag{8}$$

در معادله بالا $Q(i)$ زمان اجرای پرس‌وجوی iام است و بنابراین داریم

$$E[TIME(A)] = tE[Q(i)] \tag{9}$$

به کمک آنچه در (۶) گفتیم می‌توان (۸) را به شکل زیر بازنویسی کرد

$$\begin{aligned}E[TIME(A)] &= t(\Pr(a \le n)\times 1 + \Pr(a > n)\times O(n))\\ &= t\left(\frac{n}{n+\log n} + \frac{\log n}{n+\log n}O(n)\right)\\ &= t\left(\frac{n}{n+\log n} + \frac{c' n\log n}{n+\log n}\right)\end{aligned} \tag{10}$$

که c' یک مقدار ثابت است. با توجه به مقدار t داریم

$$\begin{aligned}E(TIME(A)) &\le c\sqrt{l+n}\,(O(1)+O(\log n))\\ &= O(\sqrt{l+n})\log n = \tilde{O}(\sqrt{n})\end{aligned} \tag{11}$$

بنابراین امید ریاضی زمان اجرای الگوریتم کماکان زیرخطی خواهد ماند.

قضیه ۹: متوسط زمان اجرای الگوریتم ۲ از مرتبه $\tilde{O}(\sqrt{n})$ بوده و ضریب تقریب آن $2+\varepsilon$ است.

اثبات: توجه داریم که نمونه‌برداری عملاً از گراف $\tilde{G}$ انجام می‌شود و بنابراین با توجه به توضیحات بخش ۳-۲ و نتیجه ۷، ضریب تقریب الگوریتم همان $2+\varepsilon$ خواهد بود. همچنین همان طور که در (۱۰) نشان دادیم، امید ریاضی زمان اجرا از مرتبه $\tilde{O}(\sqrt{n})$ می‌باشد.

۴- جمع‌بندی

در این نوشتار مسئله میانگین درجه رنگی برای یک گراف را مورد بررسی قرار دادیم و دو روش با تقریب $2+\varepsilon$ برای حل این مسئله در زمان زیرخطی ارائه کردیم. روش اول مبتنی بر اطلاع کامل از وضعیت گراف بود که در زمان $O(\sqrt{n+l})$ قادر به تقریب میانگین درجه رنگی با حداکثر خطای $2+\varepsilon$ می‌باشد. همچنین به کمک الگوریتم ۲ با انتخاب $O(\sqrt{n+l})$ رأس و در زمان متوسط $\tilde{O}(\sqrt{n})$ می‌توانیم به همان ضریب تقریب دست یابیم.

مراجع

[1] R. Rubinfeld and A. Shapira, "Sublinear time algorithms," *SIAM J. on Discrete Mathematics*, vol. 25, no. 4, pp. 1562-15882011.

[2] A. Dasgupta, R. Kumar, and T. Sarlos, "On estimating the average degree," in *Proc. 23rd Int. Conf. on World Wide Web, WWW'14,* pp. 795-806, Seoul, South Korea, 7-11 Apr. 2014.

[3] O. Goldreich, "Introduction to testing graph properties," Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation, pp. 470-506, 2011.

[4] O. Goldreich and D. Ron, "Approximating average parameters of graphs," *Random Structures and Algorithms*, vol. 32, no. 4, pp. 473-493, 2008.

[5] O. Goldreich and D. Ron, "Estimating simple graph parameters in sublinear time," *Encyclopedia of Algorithms*, pp. 650-653, 2016.

[6] R. Canetti, G. Even, and O. Goldreich, "Lower bounds for sampling algorithms for estimating the average," *Information Processing Letters*, vol. 53, no. 1, pp. 17-25, 13 Jan. 1995.

[7] R. Motwani, R. Panigrahy, and Y. Xu, "Estimating sum by weighted sampling," in *Proc. Int. Colloquium on Automata, Languages, and Programming, ICALP'07*, pp. 53-64, Wroclaw, Poland, 9-13 Jul. 2007.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd Edition, MIT Press, 2009.

[9] U. Feige, "On sums of independent random variables with unbounded variance and estimating the average degree in a graph," *SIAM J. on Computing*, vol. 35, no. 4, pp. 964-984, 2006.

[10] T. Kaufman, M. Krivelevich, and D. Ron, "Tight bounds for testing bipartiteness in general graphs," *SIAM J. on Computing*, vol. 33, no. 6, pp. 1441-1483, 2004.

[11] O. Goldreich and D. Ron, "On estimating the average degree of a graph," *Electronic Colloquium on Computational Complexity*, vol. 11, Article 13, 9 pp., 2004.

محمدعلی آبام در ۳۶امین المپیاد جهانی ریاضیات موفق به کسب مدال برنز شد. او مدرک کارشناسی و کارشناسی ارشد خود را از دانشگاه صنعتی شریف به ترتیب در سال‌های ۱۳۷۸ و ۱۳۸۰ دریافت کرد. سپس در سال ۱۳۸۶ دکتری خود را از دانشگاه TU Eindhoven دریافت کرد و از سال ۱۳۹۰ به عنوان عضو هیأت علمی به دانشکده‌ی مهندسی کامپیوتر دانشگاه صنعتی شریف ملحق شد. زمینه‌های مورد علاقه‌ی پژوهشی وی شامل هندسه‌ی محاسباتی و گسسته، الگوریتم‌های داده‌های حجیم و الگوریتم‌های تقریبی و تصادفی می‌باشد.

محمدرضا بهرامی مدرک کارشناسی خود را از دانشگاه شهید بهشتی در سال ۱۳۹۴ دریافت کرد. سپس در سال ۱۳۹۶ مدرک کارشناسی ارشد خود را از دانشگاه صنعتی شریف اخذ نمود. در حال حاضر وی دانشجوی دکتری مهندسی کامپیوتر در دانشگاه صنعتی شریف است و به پژوهش در زمینه‌ی الگوریتم‌های گراف، الگوریتم‌های تصادفی و تقریبی برای داده‌های بزرگ علاقمند است.