فرهنگ و اندیشهٔ ریاضی
سال ۳۷، شمارهٔ ۶۳ (پاییز و زمستان ۱۳۹۷) صص. ۱۲۹ تا ۱۴۷

رتبه‌بندی رأس‌های گراف

حسن حیدری و سید محمود طاهری

چکیده

یک مسئلهٔ مهم در نظریهٔ گراف، علوم کامپیوتر و شبکه‌های اجتماعی، مشخص کردن اهمیت رأس‌های یک گراف (یا گره‌های یک شبکه) است. بدین منظور، معیارها و روش‌های گوناگونی پیشنهاد شده است. یکی از این روش‌ها، رتبه‌بندی است که بر پایهٔ گام‌برداریِ تصادفی بنا شده است. هدف ما در این مقاله، توضیح الگوریتم رتبه‌بندی به دو شکل متمرکز و توزیع‌شده است. به این منظور، نخست مفهوم رتبه‌بندی و الگوریتم محاسبهٔ آن را به‌صورت متمرکز توضیح می‌دهیم. سپس یک الگوریتم رتبه‌بندی توزیع‌شده مبتنی بر شبیه‌سازی مونت‌کارلو را که در $O(\log n)$ دور با احتمال زیاد پایان می‌پذیرد، تشریح می‌کنیم. همچنین کاربردهایی از این الگوریتم را در مسیریابیِ بسته‌ها و زمان‌بندیِ کارها در سیستم‌های توزیع‌شده بیان می‌کنیم.

۱. سرآغاز و انگیزه

در شبکه‌ها (گراف‌ها[۱]) به‌ویژه در شبکهٔ جهانی وب[۲]، شبکه‌های اجتماعی[۳] و شبکه‌های زیستی[۴] و همچنین در سیستم‌های توزیع‌شده[۵] به‌منظور تعیین اهمیت گره‌ها از *معیارهای مرکزیت*[۶] استفاده می‌شود. یک معیار مرکزیت تابعی است که به هر کدام از گره‌های شبکه متناسب با اهمیت آن گره، عددی حقیقی

عبارات و کلمات کلیدی. روش مونت‌کارلو؛ گام‌برداریِ تصادفی؛ معیارهای مرکزیت؛ سیستم‌های توزیع‌شده.

[۱] در نوشتارهای مربوط به شبکهٔ جهانی وب و شبکه‌های اجتماعی، از واژه‌های شبکه، گره و لینک (پیوند) استفاده می‌شود در حالی که در متن‌های ریاضی به‌ترتیب، واژه‌های گراف، رأس و یال به‌کار گرفته می‌شوند. در این مقاله، از هر دو مجموعه واژه‌های مذکور، حسبِ مورد استفاده کرده‌ایم.

[۲] world wide web [۳] social networks [۴] biological networks [۵] distributed systems [۶] centrality

نسبت می‌دهد. معیارهای مرکزیت را بر اساس روش‌های گوناگون از قبیل مرکزیت مبتنی بر درجه[۱]، بردار ویژه[۲]، رتبه‌بندی[۳] و نزدیکی[۴] تعریف می‌کنند [۱۳]. در این میان، روش رتبه‌بندی که برای اولین بار توسط برین و پیج در سال ۱۹۹۸ به‌منظور تعیین اهمیت صفحه‌های وب و در حوزهٔ الگوریتم‌های وب مطرح شد و موجب شکل‌گیریِ شرکت گوگل گردید، به‌دلیل کاربردهای گسترده‌ای که بعدها در سایر حوزه‌های علوم کامپیوتر پیدا کرد، از اهمیت ویژه‌ای برخوردار است [۴]. در این مقاله، به معرفی و بررسی این روش خواهیم پرداخت ولی پیش از ادامهٔ بحث و به‌منظور تشریح اهمیت رتبه‌بندی گره‌های یک شبکه در مسائل کاربردی، توجه خواننده را به مثال‌های زیر جلب می‌کنیم.

مثال ۱.۱ (کاربرد در شبکهٔ وب). اگر صفحه‌های وب را معادل با گره‌های شبکه و اَبَرلینک‌هایی[۵،۶] را که بین صفحه‌های وب پیوند ایجاد می‌کنند، لینک‌های شبکه در نظر بگیریم، شبکهٔ ایجاد شده را *شبکهٔ وب* می‌نامیم. *پیمایش‌کنندگان*[۷] صفحه‌های وب، با جستجوی یک عبارت در موتورهای جستجو مانند گوگل یا بینگ، غالباً با صدها یا هزاران صفحه مواجه می‌شوند که بررسی همهٔ این صفحه‌ها زمان‌بر و در بسیاری اوقات ناممکن است. لذا باید صفحه‌های مذکور به شیوه‌ای مناسب مرتب شوند و صفحه‌هایی که اهمیت بیشتر دارند، در ابتدای نتیجهٔ جستجو قرار بگیرند.

مثال ۲.۱ (کاربرد در امور نظامی). اگر سربازان و فرماندهان یک لشگر نظامی را گره‌های شبکه و ارتباط‌های بین آنها را لینک‌های شبکه در نظر بگیریم، شبکهٔ ایجادشده را یک *شبکهٔ نظامی* می‌نامیم. به‌منظور انجام حملهٔ نظامی و رساندن بیشترین آسیبِ ممکن به دشمن، می‌توانیم همهٔ نیروی‌های دشمن را مورد حمله قرار دهیم. واضح است که این عملیات پرهزینه و زمان‌بر خواهد بود. از آنجا که نابودی گره‌های با اهمیت (فرماندهان) باعث از هم گسیختگی شبکه (لشگر) خواهد شد، لذا به‌جای حمله به همهٔ گره‌های شبکه، می‌توانیم تنها به گره‌های مهم حمله کنیم. برای این کار، نیاز است که نخست گره‌های مهم را شناسایی کنیم.

مثال ۳.۱ (کاربرد در شبکه‌های اجتماعی LinkedIn و ResearchGate). این دو شبکه به‌ترتیب در حوزه‌های «کسب و کار» و «پژوهش» فعالیت می‌کنند. اعضای این شبکه‌های اجتماعی، مهارت‌ها و

---

[۳] توجه کنید که در نوشتارهای علوم شبکه، واژهٔ رتبه‌بندی معادل PageRank است. در این مقاله، هرگاه از ترکیب «رتبه‌بندی رأس‌های گراف» یا «رتبه‌بندی گره‌های شبکه» استفاده می‌کنیم، منظور معنای عام رتبه‌بندی (مرتب‌سازی بر اساس اهمیت) است وگرنه منظور از PageRank، یک روش ویژه برای رتبه‌بندی رأس‌های گراف است.

[۶] فرض کنیم در صفحهٔ وب A باشیم. تمام اجزایی از A را که کلیک بر روی آنها، موجب انتقال به صفحه‌های وب دیگر می‌گردد، *اَبَرلینک* می‌نامیم. اگر چنین جزئی در A وجود داشته باشد و با کلیک بر روی آن، به صفحهٔ وب B انتقال یابیم، می‌گوییم A یک *اَبَرلینک خروجی* به B دارد.

[۱] degree centrality [۲] eigenvector centrality [۴] closeness centrality [۵] hyperlink [۷] surfer

توانایی‌های خود را در شبکهٔ اجتماعی وارد می‌کنند. همچنین می‌توانند مهارت‌ها و توانایی‌های واردشده توسط سایر اعضا را تأیید[۱] کنند. برای مقایسهٔ توانایی‌های چند تن از اعضای شبکهٔ اجتماعی، می‌توانیم از لیست مهارت‌های ایشان و تأییدیه‌های مهارت‌ها استفاده کنیم. روشن است که بررسی تعداد تأییدیه‌های یک مهارت، به‌تنهایی نمی‌تواند معیاری مناسب برای مقایسه باشد، چراکه اعتبار افراد تأییدکننده نیز مهم است. لذا مسئلهٔ مقایسهٔ اعضا، مسئله‌ای ساده و بدیهی نیست. اگر اعضا را رأس‌های گراف و تأییدیه‌ها را یال‌های گراف در نظر بگیریم، می‌توانیم برای حل این مسئله از یک سازوکار رتبه‌بندی رأس‌های گراف، استفاده کنیم [۱۶].

اکنون به موضوع رتبه‌بندی باز می‌گردیم. رتبه‌بندی، مبتنی بر یک زنجیر مارکوف است که در حالت متمرکز[۲]، با به‌توان رساندنِ ماتریس انتقال[۳] زنجیر مارکوف محاسبه می‌شود. حتی اگر انجام عملیات مذکور به‌صورت توزیع‌شده بر روی مدل محاسباتی متراکم[۴] [۱۴] مقدور باشد، از نظر زمانی و ارتباطی، بسیار پُرهزینه خواهد بود. لذا محاسبهٔ توزیع‌شدهٔ رتبه‌بندی بر روی مدل محاسباتی متراکم، مستلزم استفاده از روشی متفاوت با روش متمرکز است. هزینهٔ زمانی و ارتباطیِ گام‌برداریِ تصادفی در سیستم‌های توزیع‌شده، نسبت به عملیات توان‌رسانیِ ماتریس انتقال، بسیار کمتر است [۱۸]. بر همین اساس، روشی جدید مبتنی بر شبیه‌سازی مونت‌کارلو برای محاسبهٔ توزیع‌شدهٔ رتبه‌بندی ارائه گردیده است [۲، ۱۷].

موضوع این مقاله، تشریح رتبه‌بندی رأس‌های یک گراف و بررسی یک الگوریتم متمرکز و توزیع‌شده برای این مسئله است. در بخش ۲، معیار مرکزیت، معیار مرکزیت مبتنی بر درجه، زنجیر مارکوف، گام‌برداریِ تصادفی، رتبه‌بندی و الگوریتم متمرکزِ محاسبهٔ رتبه‌بندی را توضیح می‌دهیم. در بخش ۳، تعریف سیستم‌های توزیع‌شده و مدل محاسباتی مورد استفاده را بیان می‌کنیم و کاربردهایی از محاسبهٔ رتبه‌بندی را در زمان‌بندی کارها و مسیریابی بسته‌ها در سیستم‌های توزیع‌شده توضیح می‌دهیم. در بخش ۴، برای محاسبهٔ رتبه‌بندی، الگوریتمی را شرح می‌دهیم که بر پایهٔ شبیه‌سازی مونت‌کارلو است [۲] و از آن برای ارائهٔ الگوریتم توزیع‌شدهٔ رتبه‌بندی استفاده می‌کنیم.

۲. مفاهیم و نتایج مقدماتی

در این بخش، مفاهیم کلیدی این مقاله (مرکزیت، رتبه‌بندی و گام‌برداریِ تصادفی) را همراه با برخی مفاهیم مرتبط با آنها بیان و تشریح می‌کنیم.

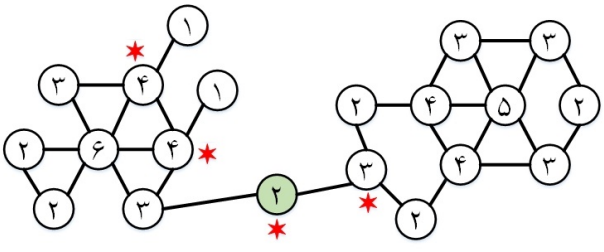
۱.۲. مرکزیت. حجم عظیمی از پژوهش‌هایی که در حوزهٔ شبکه‌ها انجام می‌شوند، دربارهٔ مفهوم مرکزیت است. به گره‌هایی از شبکه که دارای اهمیت بیشتری هستند، *گره‌های مرکزی* می‌گوییم. در واقع

[۱] endorse [۲] centralized [۳] transition matrix [۴] congest

مفهوم مرکزیت معادل با این پرسش است: «کدام گره‌های شبکه، مرکزی هستند؟» تعریف رسمی معیار مرکزیت را در ادامه بیان می‌کنیم.

تعریف ۱.۲. فرض کنیم $G = (V, E)$ یک گراف باشد که در آن، V و E به‌ترتیب مجموعهٔ رأس‌ها و یال‌های گراف هستند. هر تابع $C : V \to \mathbb{R}$ یک *معیار مرکزیت* C نامیده می‌شود که در آن، $C(v)$ اهمیت گره $v \in V$ را نشان می‌دهد.

به‌منظور تعیین مرکزیت، معیارهای متفاوتی ارائه شده است. یکی از رایج‌ترین معیارهای مرکزیت، *معیار مبتنی بر درجه* است که در آن، اهمیت هر گره متناسب با درجهٔ آن گره است. غالباً در شبکه‌ها گره‌هایی که درجهٔ بزرگتری دارند، اهمیت بیشتری نیز دارند. اما عکس این موضوع همیشه صادق نیست. به بیان دیگر، گره‌های با درجهٔ کوچک، الزاماً کم‌ارزش نیستند. برای مثال، در شکل ۱ با اینکه درجهٔ گره خاکستری کوچک است ولی چون حذف آن موجب ناهمبند شدن شبکه می‌شود، از اهمیت زیادی برخوردار است. گفتنی است که به رأس‌هایی مانند رأس‌های ستاره‌دار در شکل ۱ که حذف آنها موجب ناهمبند شدن گراف می‌شود، *نقاط مَفصَلی*[۱] یا *رأس‌های برشی*[۲] می‌گوییم. در نظر داشته باشید که گراف شکل ۱، بی‌جهت



شکل ۱. مثالی از معیار مرکزیت مبتنی بر درجه. عددهایی که در هر گره نوشته شده است، درجهٔ آن گره را نشان می‌دهند که معادل با اهمیت آن گره در معیار مرکزیت مبتنی بر درجه است. گره‌های ستاره‌دار، مَفصَلی (برشی) هستند. گره خاکستری با اینکه درجهٔ کوچکی دارد، به‌دلیل مَفصَلی (برشی) بودن، اهمیت زیادی دارد.

است و چون اکثر شبکه‌ها جهتدار هستند، می‌توانیم معیار مبتنی بر درجه را در گراف‌های جهتدار و بر اساس درجه‌های ورودی و خروجی تعریف کنیم. همان‌طور که در شکل ۱ پیدا است، معیار مرکزیت مبتنی بر درجه، در برخی از موارد، ارزش و اهمیت گره‌ها را به‌درستی تعیین نمی‌کند (مانند گره خاکستری). برای مطالعهٔ معیارهای مرکزیت و بررسی آنها، [۱۳] را مطالعه کنید.

۲.۲. زنجیر مارکوف و گام‌برداریِ تصادفی. پیش از بررسی روش رتبه‌بندی، مطالبی را دربارهٔ فرآیندهای تصادفی از [۷، ۱۱] یادآوری می‌کنیم. یک فرآیند تصادفی خانواده‌ای از متغیرهای تصادفی مانند

[۱] articulation points [۲] cut vertices

$X = \{X_t : t \in T\}$ با مجموعهٔ اندیس‌گذار T (مجموعهٔ زمان) است که هر کدام از این متغیرهای تصادفی، فضای نمونه‌ای Ω را به یک مجموعهٔ S می‌نگارد. مجموعهٔ S را *فضای حالت* و هر یک از عضوهای آن را یک *حالت* می‌نامیم. معمولاً برای مجموعهٔ اندیس‌گذار T و فضای حالت S این موردها در نظر گرفته می‌شود: (الف) $T = \{0, 1, 2, ...\}$ یا $T = [0, \infty)$؛ (ب) $S = \mathbb{N}$ یا $S = \mathbb{R}$. به این ترتیب، در یک تقسیم‌بندی کلی، فرآیندهای تصادفی به چهار رده تقسیم می‌شوند: زمان گسسته–حالت گسسته؛ زمان گسسته–حالت پیوسته؛ زمان پیوسته–حالت گسسته و زمان پیوسته–حالت پیوسته. یک فرآیند تصادفی زمان گسسته–حالت گسسته را *زنجیر مارکوف* می‌نامیم هر گاه به‌ازای هر $n \in \mathbb{N}$ و هر $i_0, i_1, ..., i_{n-1}, i, j \in S$ داشته باشیم

$$P(X_{n+1} = j | X_0 = i_0, \ldots, X_{n-1} = i_{n-1}, X_n = i) = P(X_{n+1} = j | X_n = i)$$

به این شرط که $P(X_0 = i_0, \ldots, X_{n-1} = i_{n-1}, X_n = i) > 0$. این رابطه را *ویژگی مارکوفی* می‌خوانیم و معنای آن این است که احتمال شرطی پیشامد آینده ($\{X_{n+1} = j\}$) فقط به پیشامد حال ($\{X_n = i\}$) وابسته است و به پیشامدهای گذشته ($\{X_0 = i_0, \ldots, X_{n-1} = i_{n-1}\}$) وابسته نیست. به بیان دیگر، آینده به‌شرط حال، از گذشته مستقل است. برای $i, j \in S$ احتمال‌های شرطی $P(X_{n+1} = j | X_n = i)$ را احتمال‌های انتقال یک‌مرحله‌ای، یا به‌طور ساده *احتمال‌های انتقال* می‌نامیم. اگر احتمال‌های انتقال به n بستگی نداشته باشند، فرآیند را همگن می‌گوییم و احتمال انتقال از حالت i به حالت j را با p_{ij} نشان می‌دهیم. سپس احتمال‌های انتقال را در یک ماتریس $n \times n$ قرار می‌دهیم و آن را ماتریس احتمال‌های انتقال می‌نامیم.

برای $i, j \in S$ می‌گوییم حالت j در *دسترسِ* حالت i است و می‌نویسم $i \to j$ اگر زنجیر با شروع از i، با احتمال مثبت به حالت j برسد. دو حالت i و j را در دسترس یکدیگر می‌نامیم، اگر $i \to j$ و $j \to i$ و می‌نویسیم $i \leftrightarrow j$. یک زنجیر مارکوف را *تحویل‌ناپذیر* گوییم اگر به‌ازای هر $i, j \in S$ داشته باشیم $i \leftrightarrow j$.

فرض کنیم p_{ij} احتمال انتقال یک‌مرحله‌ای از حالت i به حالت j و $p_{ij}^{(n)}$ احتمال انتقال nمرحله‌ای از i به j باشد. دورهٔ تناوب حالت i را با $d(i)$ نشان می‌دهیم که عبارت است از بزرگترین مقسوم‌علیه مشترک همهٔ عددهای صحیح $n \geq 1$ که در آنها $p_{ii}^{(n)} > 0$. به بیان دیگر،

$$d(i) = \gcd\{n \geq 1 : \ p_{ii}^{(n)} > 0\}.$$

اگر $\{n \geq 1 : p_{ii}^{(n)} > 0\} = \varnothing$، آنگاه $d(i)$ را ۱ در نظر می‌گیریم. اگر $d(i) = 1$، حالت i را *نامتناوب* و اگر $d(i) > 1$، حالت i را *متناوب* با دورهٔ تناوب $d(i)$ می‌خوانیم.

فرض کنیم گراف $G = (V, E)$ بی‌جهت، متناهی و همبند باشد. درجهٔ رأس u را با نماد $\deg(u)$ نشان می‌دهیم. گام‌برداریِ تصادفی همگن روی گراف G، یک زنجیر مارکوف است که دنباله‌ای از انتقال‌ها بین رأس‌های گراف را نشان می‌دهد طوری که احتمال انتقال از رأس u به هر رأس همسایهٔ آن برابر با $\frac{۱}{\deg(u)}$ است.

در یک گراف جهتدار، وقتی می‌نویسیم (u, v) منظورمان این است که یالی از رأس u به رأس v وجود دارد. مجموعهٔ همهٔ رأس‌هایی مانند v را که (u, v)، با $V_O(u)$ نشان می‌دهیم. گام‌برداریِ تصادفی همگن، مشابه گراف‌های بی‌جهت، روی گراف‌های جهتدار نیز قابل تعریف است با این تفاوت که اگر (u, v)، آن‌گاه احتمال انتقال از u به v برابر با $\frac{۱}{|V_O(u)|}$ است.

۳.۲. رتبه‌بندی. روش رتبه‌بندی در ابتدا برای تعیین اهمیت صفحه‌های وب مطرح شد. یک پیمایش‌کنندهٔ صفحه‌های وب را در حال مشاهدهٔ صفحهٔ وب A در نظر بگیرید. پس از مشاهدهٔ A، یکی از چهار پیشامد زیر ممکن است رخ دهد:

(۱) پیمایش‌کننده به پیمایش صفحه‌های وب (وِبگردی) پایان می‌دهد؛
(۲) پیمایش‌کننده از طریق یکی از اَبَرلینک‌های خروجی A وارد قسمتی دیگر از A شود؛
(۳) پیمایش‌کننده از طریق یکی از اَبَرلینک‌های خروجی A، وارد صفحهٔ B شود؛
(۴) پیمایش‌کننده وارد صفحهٔ C شود که جزء لینک‌های خروجی A نیست.

در حالتی که پیمایش‌کننده از A وارد قسمتی دیگر از A می‌شود (حالت دوم)، این گونه در نظر می‌گیریم که A دو بار دیده شده است. اگر پیمایش‌کننده از A وارد B (یا C) گردد (حالت سوم و چهارم)، صفحهٔ A و B (یا C) هر کدام یک بار دیده شده‌اند. حال تعداد زیادی پیمایش‌کننده را در نظر بگیرید که برای هر کدام از آنها در هر لحظه یکی از چهار پیشامد ذکر شده رخ می‌دهد. یکی از روش‌های تعیین اهمیت صفحه‌ها این است که بگوییم اهمیت صفحه‌ها متناسب با تعداد پیمایش‌کنندگانی است که وارد آنها شده‌اند. در واقع رتبه‌بندی، مدل‌سازیِ ریاضیِ رفتار پیمایش‌کنندگان است و بر پایهٔ شیوه بالا، اهمیت صفحه‌های وب را می‌سنجد. در این روش، صفحه‌هایی که دارای لینک‌های ورودی بیشتری هستند، توسط پیمایش‌کنندگانِ بیشتری مشاهده می‌شوند. لذا به آنها رتبهٔ بالاتری نسبت داده می‌شود. البته سهم همهٔ لینک‌های ورودی در رتبهٔ یک گره به یک اندازه نیست و تأثیر لینک‌های ورودی از صفحه‌های با رتبهٔ بالا نسبت به صفحه‌های با رتبه پایین، بیشتر است.

برای محاسبهٔ رتبه‌بندی، شبکهٔ وب را توسط گراف جهتدار $D = (V, A)$ که در آن، V و A به‌ترتیب، مجموعهٔ رأس‌ها و یال‌ها هستند، مدل‌سازی می‌کنیم. در گراف D یک تناظر یک‌به‌یک بین مجموعهٔ رأس‌ها و صفحه‌های وب برقرار است و از رأس u به رأس v یک یال وجود دارد اگر و تنها اگر

صفحهٔ متناظر با رأس u دارای یک اَبَرلینک خروجی به صفحهٔ متناظر با رأس v باشد. فرض می‌کنیم $|V| = n$. برای هر گراف D، ماتریس مربعی $\tilde{P}_{n\times n}$ را به‌صورت زیر می‌سازیم:

$$\tilde{p}_{v,u} = \begin{cases} \begin{cases} \frac{1}{|V_O(v)|} & u \in V_O(v) \\ 0 & \text{اگرنه} \end{cases} & |V_O(v)| > 0 \\ \frac{1}{n} & |V_O(v)| = 0 \end{cases} \qquad (1.2)$$

از ماتریس $\tilde{P}$ برای ساخت ماتریس انتقال یک زنجیر مارکوف استفاده می‌کنیم.

اکنون تعریف رسمی رتبه‌بندی را ارائه می‌کنیم.

تعریف ۲.۲. گیریم $D = (V, A)$ یک گراف جهتدار باشد. *توزیع مانای*[1] زنجیر مارکوفی را که فضای حالت آن مجموعهٔ V و ماتریس انتقال آن $P = \epsilon\tilde{P} + (1-\epsilon)(1/n)J$ باشد، *رتبه‌بندی* می‌گوییم. در اینجا J یک ماتریس مربعی $n \times n$ با درایه‌های ۱ و ϵ عددی حقیقی در بازه $(0, 1)$ است که احتمال جذب در یک گره (باقی ماندن در گره) را نشان می‌دهد.

قضیه ۳.۲ ([۱۲]). *فرض کنیم P ماتریس انتقال یک زنجیر مارکوف باشد. اگر P نامتناوب*[2] *و تحویل‌ناپذیر*[3] *باشد، آن‌گاه یک بردار سطری π (که به آن توزیع مانای زنجیر مارکوف می‌گوییم) با شرایط $\pi P = \pi$ و $\pi\underline{1} = 1$ وجود دارد که در آن، $\underline{1}$ برداری ستونی است که همهٔ درآیه‌های آن ۱ هستند.*

به‌عبارت دیگر، به‌ازای هر حالت i $(i = 1, 2, \ldots, n)$، حد $\lim_{t\to\infty} p_{ji}^{(t)} \geq 0$ وجود دارد و از j $(j = 1, 2, \ldots, n)$ مستقل است. این حد را با π_i نشان می‌دهیم و اینها درآیه‌های π را تشکیل می‌دهند.

قضیه ۴.۲ ([۲]). *ماتریس انتقال P در تعریف ۲.۲، در شرایط قضیهٔ ۳.۲ صدق می‌کند.*

برای محاسبهٔ رتبه‌بندی، قضیهٔ ۳.۲ دو روش ارائه می‌کند. یکی محاسبهٔ ماتریس انتقال P و حل معادلهٔ $\pi P = \pi$ و دیگری، محاسبهٔ ماتریس انتقال و به‌توان رساندنِ آن. برای مثال، در شکل ۲ گراف یک شبکهٔ وب متشکل از ۵ صفحه را به‌همراه رتبهٔ رأس‌های آن مشاهده می‌کنید که رتبهٔ رأس‌ها با روش رتبه‌بندی محاسبه شده‌اند. شیوهٔ محاسبهٔ رتبه‌بندی با استفاده از تعریف ۲.۲ و قضیهٔ ۳.۲ گام‌به‌گام به‌صورت زیر است:

روش اول)

(۱) قرار می‌دهیم $\epsilon = 0/85$ (در ادامه دلیل این کار را توضیح می‌دهیم).

[1] stationary distribution [2] aperiodic [3] irreducible

(۲) ماتریس $\tilde{P}$ را بر اساس رابطهٔ (۱.۲) به‌صورت زیر می‌سازیم:

$$\tilde{P} = \begin{bmatrix} ۰ & \frac{۱}{۲} & ۰ & \frac{۱}{۲} & ۰ \\ ۰ & ۰ & ۱ & ۰ & ۰ \\ ۱ & ۰ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۰ & ۱ \\ \frac{۱}{۵} & \frac{۱}{۵} & \frac{۱}{۵} & \frac{۱}{۵} & \frac{۱}{۵} \end{bmatrix}.$$

(۳) برای ساختن P بر اساس تعریف و به‌صورت زیر عمل می‌کنیم:

$$P = \epsilon\tilde{P} + (۱-\epsilon)(۱/n)J = \begin{bmatrix} ۰/۰۳۰ & ۰/۴۵۵ & ۰/۰۳۰ & ۰/۴۵۵ & ۰/۰۳۰ \\ ۰/۰۳۰ & ۰/۰۳۰ & ۰/۸۸۰ & ۰/۰۳۰ & ۰/۰۳۰ \\ ۰/۸۸۰ & ۰/۰۳۰ & ۰/۰۳۰ & ۰/۰۳۰ & ۰/۰۳۰ \\ ۰/۰۳۰ & ۰/۰۳۰ & ۰/۰۳۰ & ۰/۰۳۰ & ۰/۸۸۰ \\ ۰/۲۰۰ & ۰/۲۰۰ & ۰/۲۰۰ & ۰/۲۰۰ & ۰/۲۰۰ \end{bmatrix}.$$

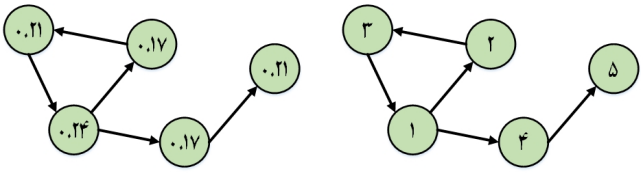
(۴) برای محاسبهٔ بردار مانا (رتبه‌بندی)، بنابر قضیهٔ ۳.۲، معادلهٔ $\pi = \pi P$ را تشکیل می‌دهیم که از آن، دستگاه معادلات خطی زیر به‌دست می‌آید:

$$\begin{cases} ۰/۰۳۰x_۱ + ۰/۰۳۰x_۲ + ۰/۸۸۰x_۳ + ۰/۰۳۰x_۴ + ۰/۲۰۰x_۵ = x_۱ \\ ۰/۴۵۵x_۱ + ۰/۰۳۰x_۲ + ۰/۰۳۰x_۳ + ۰/۰۳۰x_۴ + ۰/۲۰۰x_۵ = x_۲ \\ ۰/۰۳۰x_۱ + ۰/۸۸۰x_۲ + ۰/۰۳۰x_۳ + ۰/۰۳۰x_۴ + ۰/۲۰۰x_۵ = x_۳ \\ ۰/۴۵۵x_۱ + ۰/۰۳۰x_۲ + ۰/۰۳۰x_۳ + ۰/۰۳۰x_۴ + ۰/۲۰۰x_۵ = x_۴ \\ ۰/۰۳۰x_۱ + ۰/۰۳۰x_۲ + ۰/۰۳۰x_۳ + ۰/۸۸۰x_۴ + ۰/۲۰۰x_۵ = x_۵ \end{cases}$$

(۵) با حل دستگاه معادلات بالا و استفاده از شرط $\pi \underline{۱} = ۱$ (به‌طور معادل، $\sum_{i=۱}^{۵} x_i = ۱$) بردار مانا (رتبه‌بندی) به‌دست می‌آید:

$$\pi \approx \begin{bmatrix} ۰/۲۴۳ & ۰/۱۶۹ & ۰/۲۰۹ & ۰/۱۶۹ & ۰/۲۰۹ \end{bmatrix}$$

روش دوم) مراحل ۱، ۲ و ۳ از روش اول را تکرار می‌کنیم.



(آ) گراف D. (ب) رتبهٔ رأس‌های گراف D.

شکل ۲. در شکل ۲(آ)، گراف D متناظر با یک شبکهٔ وب که متشکل از ۵ صفحه است و در شکل ۲(ب)، رتبهٔ رأس‌های گراف D که با روش رتبه‌بندی محاسبه گردیده، نشان داده شده است.

(۴) توان دوم P را محاسبه می‌کنیم. داریم

$$P^{۲}=\begin{bmatrix} ۰/۰۶۱ & ۰/۰۴۸ & ۰/۴۲۲ & ۰/۰۴۸ & ۰/۴۲۲ \\ ۰/۷۸۳ & ۰/۰۴۸ & ۰/۰۶۱ & ۰/۰۴۸ & ۰/۰۶۱ \\ ۰/۰۶۱ & ۰/۴۰۹ & ۰/۰۶۱ & ۰/۴۰۹ & ۰/۰۶۱ \\ ۰/۲۰۵ & ۰/۱۹۲ & ۰/۲۰۵ & ۰/۱۹۲ & ۰/۲۰۵ \\ ۰/۲۳۴ & ۰/۱۴۹ & ۰/۲۳۴ & ۰/۱۴۹ & ۰/۲۳۴ \end{bmatrix}.$$

(۵) باید $P^{۳}$ و $P^{۴}$ و ... را نیز پیدا کنیم. هرچه توان بزرگتری از ماتریس انتقال را محاسبه کنیم، به تقریب دقیق‌تری از بردار توزیع مانا دست پیدا می‌کنیم. در اینجا تا توان صدم را محاسبه کرده‌ایم:

$$P^{۱۰۰}=\begin{bmatrix} ۰/۲۴۳ & ۰/۱۶۹ & ۰/۲۰۹ & ۰/۱۶۹ & ۰/۲۰۹ \\ ۰/۲۴۳ & ۰/۱۶۹ & ۰/۲۰۹ & ۰/۱۶۹ & ۰/۲۰۹ \\ ۰/۲۴۳ & ۰/۱۶۹ & ۰/۲۰۹ & ۰/۱۶۹ & ۰/۲۰۹ \\ ۰/۲۴۳ & ۰/۱۶۹ & ۰/۲۰۹ & ۰/۱۶۹ & ۰/۲۰۹ \\ ۰/۲۴۳ & ۰/۱۶۹ & ۰/۲۰۹ & ۰/۱۶۹ & ۰/۲۰۹ \end{bmatrix}$$

(۶) بنابر قضیهٔ ۳.۲، یکی از سطرهای ماتریس $P^{۱۰۰}$ را به‌عنوان π در نظر می‌گیریم. توجه دارید که مهم نیست کدام سطر را به‌عنوان π انتخاب کنیم.

پیچیدگی محاسباتیِ روش اول با استفاده از روش حذفی گاوس، از مرتبهٔ $O\big(n^{۳}(\log n)^{۲}\big)$ است که در آن، n تعداد گره‌های شبکه است [۵]. همچنین پیچیدگی محاسباتیِ روش دوم، با استفاده از روش ضرب ماتریس‌های آهو و همکاران [۱]، از مرتبهٔ $O(n^{۲/۳۷۳})$ است. لذا پیچیدگی محاسباتیِ روش دوم نسبت به

روش اول کمتر است و به همین دلیل، گوگل برای محاسبهٔ رتبه‌بندی صفحه‌های وب از روش دوم استفاده می‌کند [۱۰].

در جدول ۱، تعداد دورهای[۱] اجرای روش دوم (در دور iام، $P^{(i)}$ را محاسبه می‌کنیم) را به‌ازای چند مقدارِ ϵ از [۱۰] آورده‌ایم. با افزایش ϵ، دقت الگوریتم افزایش می‌یابد. همان‌طور که از جدول پیدا است، با افزایش ϵ، تعداد دورهای اجرای الگوریتم نیز افزایش می‌یابد. گفتنی است که شرکت گوگل مقدار ۰٫۸۵ را به‌طور تجربی برای ϵ در نظر گرفته است.

جدول ۱. تاثیر ϵ (احتمال جذب) بر تعداد دورهای اجرای الگوریتم رتبه‌بندی [۱۰].

تعداد دورهای اجرای الگوریتم	ϵ	تعداد دورهای اجرای الگوریتم	ϵ
۲۱۹	۰/۹۰۰	۳۴	۰/۵۰۰
۴۴۹	۰/۹۵۰	۸۱	۰/۷۵۰
۲۲۹۲	۰/۹۹۰	۱۰۴	۰/۸۰۰
۲۳۰۱۵	۰/۹۹۹	۱۴۲	۰/۸۵۰

۳. سیستم‌های توزیع‌شده : مفاهیم و نتایج مقدماتی

در بخش قبل، روش‌های محاسبهٔ متمرکز رتبه‌بندی را بررسی کردیم. چون الگوریتم مذکور روی یک سیستم متمرکز (یا سیستم تک‌پردازنده‌ای[۲]) اجرا می‌شود، آن را متمرکز می‌نامیم. در مقابل سیستم‌های متمرکز، سیستم‌های توزیع‌شده قرار دارند [۶، ۱۴]. الگوریتم رتبه‌بندی بر روی سیستم‌های توزیع‌شده نیز قابل محاسبه است. پیش از بررسی الگوریتم رتبه‌بندی توزیع‌شده، تعریف سیستم‌های توزیع‌شده را بیان می‌کنیم.

تعریف ۱.۳ ([۶]). یک سیستم توزیع‌شده مجموعه‌ای از گره‌های[۳] محاسباتی است که توسط یک شبکهٔ ارتباطی به هم وصل شده‌اند و درصدد انجام کاری مشترک هستند.

منظور از گره محاسباتی در تعریف بالا، هر وسیله‌ای است که قابلیت پردازش و نگهداری داده‌ها را داشته باشد. بنابراین رایانه‌های شخصی، حسگرهای یک شبکهٔ حسگر بیسیم[۴]، پردازنده‌های یک سیستم چندپردازنده‌ای و یا تلفن‌های همراه را می‌توانیم گره‌های محاسباتی یک سیستم توزیع‌شده در نظر بگیریم. توجه به این نکته مهم است که در تعریف بالا، در سیستم‌های توزیع‌شده، واحدهای محاسباتی درصدد انجام کاری مشترک هستند.

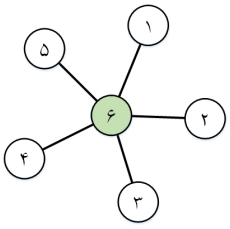
[۱] iterations [۲] single processor system [۳] node [۴] wireless sensor network

مثال ۲.۳. گراف G را به‌صورت شکل ۳(آ) در نظر می‌گیریم که در آن، هر گره معادل با یک واحد محاسباتی (برای سادگی یک رایانه) در یک سیستم توزیع‌شده است. قصد داریم تا مجموع اعداد فهرست $\sigma = [۱, ۲, \ldots, ۱۰]$ را محاسبه کنیم. بدین منظور، σ را در سیستم توزیع‌شده توزیع و سپس مجموع آن را حساب می‌کنیم. فرض می‌کنیم σ بر روی رایانهٔ شماره ۶ (شکل ۳(ب)) قرار داشته باشد. به‌جای اینکه کل محاسبات را توسط رایانهٔ ۶ انجام دهیم (محاسبهٔ متمرکز)، رایانهٔ ۶ فهرست σ را به ۵ زیرفهرست (چون ۵ واحد محاسباتی دیگر وجود دارد) می‌شکند و هر زیرفهرست را به یکی از واحدهای محاسباتی دیگر ارسال می‌کند (شکل ۳(ج)). در ادامه گره‌های ۱، ۲، ۳، ۴ و ۵ مجموع اعداد فهرست‌های خودشان را حساب و نتیجه را به گره ۶ ارسال می‌کنند (شکل ۳(د)). در پایان، گره ۶ مجموع عددهای رسیده از گره‌های دیگر را به‌عنوان خروجی اعلام می‌کند و کار پایان می‌یابد.

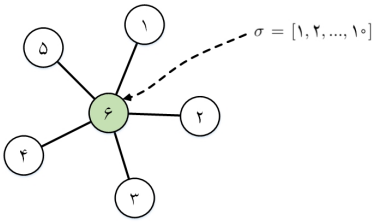
۱.۳. مدل‌سازی سیستم توزیع‌شده. به‌منظور بررسی یک سیستم توزیع‌شده، باید آن را به‌صورت ریاضی مدل‌سازی کنیم. سیستم توزیع‌شده را با گراف بدون وزن، بی‌جهت و همبند $G = (V, E)$ که در آن، $|V| = n$ و $|E| = m$ مدل‌سازی می‌کنیم. فرض می‌کنیم که هر گره دارای شناسه‌ای یکتا است که با $O(\log n)$ بیت قابل کدگذاری است (برای مثال، به هر گره، عددی یکتا از بازهٔ $[۱, n]$ تخصیص می‌دهیم و یادآوری می‌کنیم که نمایش دودویی عدد cn که در آن، c عددی ثابت است، $O(\log n)$ بیت دارد). هر گره از شناسهٔ همسایه‌های خود و تعداد کل گره‌های سیستم توزیع‌شده (n) مطلع است ولی از توپولوژیِ شبکه مطلع نیست. هر گره تنها از طریق ارسال پیام با همسایه‌های خود ارتباط برقرار می‌کند و حافظه اشتراکی[۱] (نوعی از حافظه که چند واحد محاسباتی به‌صورت اشتراکی از آن استفاده می‌کنند) وجود ندارد. سیستم توزیع‌شده همگام[۲] است و تمام پیام‌هایی که توسط گره v در دور[۳] rام ارسال شده‌اند، در پایان همان دور به مقصد می‌رسند (برای مثال، در شکل ۳ در آغاز دور اول، فهرست σ به گره ۶ ارسال می‌شود و پیش از شروع دور دوم، این فهرست توسط گره ۶ تحویل گرفته می‌شود. پس از شروع دور دوم، گره ۶ فهرست σ را به چند زیرفهرست می‌شکند و زیرفهرست‌ها را به سایر گره‌ها ارسال می‌کند. پیش از شروع دور سوم، زیرفهرست‌ها توسط سایر گره‌ها تحویل گرفته می‌شوند. پس از شروع دور سوم، هر گره مجموع عددهای فهرست خود را محاسبه می‌کند و نتیجه را برای گره ۶ ارسال می‌کند و این نتایج پیش از شروع دور چهارم توسط گره ۶، تحویل گرفته می‌شوند. در پایان، پس از شروع دور چهارم، گره ۶ محاسبات نهایی را انجام می‌دهد. لذا الگوریتم در ۴ دور خاتمه می‌یابد).

فرض می‌کنیم که *خرابی فرآیند*[۴] و یا *خرابی لینک*[۵] نداریم. سیستم توزیع‌شده از مدل متراکم که در هر دور، هر گره تنها مجاز به ارسال $O(\log n)$ بیت بر روی هر کدام از لینک‌های خود است، پیروی می‌کند. به‌منظور سنجش کارایی الگوریتم، از معیار زمان اجرا که برابر با تعداد دورهای الگوریتم و پیچیدگیِ پیامی

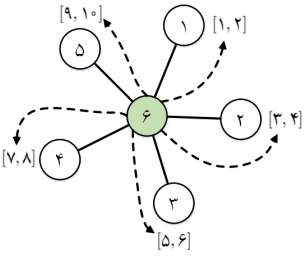
[۱] shared memory [۲] synchronous [۳] round [۴] process failure [۵] link failure



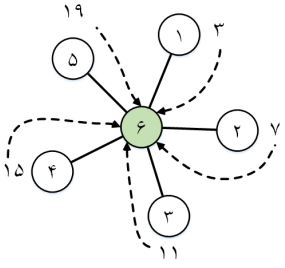
(آ) گراف G



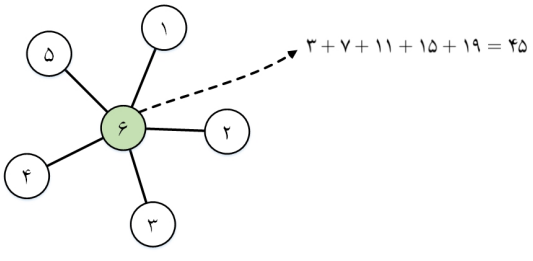
(ب) فهرست σ برای انجام محاسبات به گره ۶ ارسال می‌شود.



(ج) گره ۶، فهرست σ را به چند زیرفهرست می‌شکند و هر زیرفهرست را به یک واحد محاسباتی دیگر ارسال می‌کند.



(د) هر گره مجموع اعداد فهرست خود را محاسبه می‌کند و نتیجه را برای گره ۶ ارسال می‌کند.



(ه) گره ۶، مجموع اعدادی را که از سایر گره‌ها دریافت کرده است، محاسبه می‌کند.

شکل ۳. مثالی از یک محاسبهٔ توزیع‌شده. یک فهرست در سیستم توزیع و مجموع آن محاسبه می‌شود.

استفاده می‌کنیم که برابر است با تعداد پیام‌های مبادله شده ضرب در طول هر پیام. شایان ذکر است که این دو معیار از مهم‌ترین معیارهای سنجش الگوریتم‌های توزیع‌شده هستند [۱۴].

۲.۳. کاربرد رتبه‌بندی گره‌های یک سیستم توزیع‌شده. یکی از مسائل مطرح در حوزهٔ سیستم‌های توزیع‌شده، *توازن بار* است. اگر در انجام محاسبات توزیع‌شده، بار محاسباتی به گونه‌ای متوازن بین واحدهای محاسباتی توزیع شود، می‌گوییم توازن بار روی داده است. در مثال ۲.۳، بار محاسباتی، زیرفهرست‌هایی است که توسط گره ۶ به گره‌های ۱ تا ۵ تخصیص داده می‌شود و اگر طول زیرفهرست‌ها برابر یا تقریباً برابر باشد، توازن بار روی داده است. هدف از توازن بار، استفادهٔ بهینه از منابع، کاهش زمان پاسخ و از بین بردن سربارِ محاسباتیِ واحدهای محاسباتی است. تا به حال راه‌حل‌های بسیاری برای ایجاد توازن بار در سیستم‌های توزیع ارائه شده است [۹، ۱۵]. در ادامه، سه کاربرد از رتبه‌بندی گره‌های یک سیستم توزیع‌شده را در مسائل توازن بار، مسیریابی سریع بسته‌ها و زمان‌بندی کارها ارائه می‌کنیم.

مثال ۳.۳ (کاربرد در توازن بار). در محاسبات توزیع‌شده، گره‌ها علاوه بر انجام محاسباتی که به آنها واگذار شده است، با گره‌های دیگر نیز در تعامل هستند. برای مثال، در شکل ۳ فرض می‌کنیم هر کدام از گره‌ها در حال انجام محاسبات موضعی[۱] هستند. گره ۳، قصد ارسال یک پیام را به گره ۵ دارد. از آنجا که گره‌های با رتبهٔ بالا با تعداد زیادی از گره‌های دیگر در ارتباط هستند، قرار دادنِ بار محاسباتیِ سنگین بر روی آن گره‌ها موجب می‌شود تا توانایی پاسخ‌گویی به درخواست سایر گره‌ها را نداشته باشند. بنابراین توازن بار را می‌توانیم به گونه‌ای تعریف کنیم که متناسب با عکس رتبهٔ گره‌ها باشد.

مثال ۴.۳ (کاربرد در مسیریابی). ساختار و توپولوژی بسیاری از شبکه‌ها دائماً در حال تغییر است. لذا برای ارسال پیام از یک گره به گره دیگر، لازم است که مسیریابی به‌صورت *پویا*[۲] انجام شود. از آنجا که گره‌های با رتبهٔ بالا، با تعداد زیادی از گره‌های دیگر در ارتباط هستند، یک گره برای اینکه پیامی را انتقال دهد، از بین همسایه‌های خود به همسایه‌های با رتبهٔ بالاتر اولویت بیشتری می‌دهد و از این طریق موجب افزایش سرعت رسیدن پیام به مقصد می‌شود.

مثال ۵.۳ (کاربرد در زمان‌بندیِ کارها). چون در سیستم‌های رایانه‌ای در یک بازهٔ زمانی کوتاه نیاز به پردازش‌های بسیار است، اولویت‌بندی کارها در این سیستم‌ها امری ضروری است. چارچوب هدوپ[۳] به‌منظور ذخیره و پردازش توزیع‌شده به‌کار می‌رود. یکی از روش‌های نوین برای اولویت‌بندی کارها در چارچوب هدوپ، استفاده از رتبهٔ کارها است [۸].

اکنون که با برخی از کاربردهای رتبه‌بندی گره‌های یک سیستم توزیع‌شده آشنا شدیم، روش محاسبهٔ رتبه‌بندی را به‌صورت توزیع‌شده بررسی می‌کنیم. نخست از الگوریتم متمرکز رتبه‌بندی که آن را در بخش ۳.۲ توضیح دادیم، برای محاسبهٔ رتبه‌بندی به‌صورت توزیع‌شده استفاده می‌کنیم. گره‌های سیستم توزیع‌شده مطابق مدل‌سازی ارائه‌شده در بخش ۱.۳، از توپولوژی شبکه مطلع نیستند. بنابراین در قدم اول، همهٔ گره‌ها

[۱] local [۲] dynamic [۳] Hadoop framework

باید اطلاعات خود و همسایه‌هاشان را برای گره‌های دیگر ارسال کنند تا ماتریس $\tilde{P}$ (۱.۲) ساخته شود. اگر تعداد دورهای لازم برای ساختن ماتریس $\tilde{P}$ را در یک سیستم توزیع‌شدهٔ همگام و دارای مدل متراکم، با T نشان دهیم، رابطهٔ $O(\mathrm{diam}) \ll T$ برقرار خواهد بود که در آن، diam اندازهٔ قطر گراف است [۱۹، ۲۰]. چون توپولوژیِ بسیاری از شبکه‌ها دائماً در حال تغییر است، در فاصله‌های زمانی کوتاهی باید هزینهٔ مذکور را برای ساختن $\tilde{P}$ بپردازیم. با آنکه باید قدم‌های دیگری را نیز تا محاسبهٔ رتبه‌بندی طی کنیم، هزینهٔ زمانی قدم اول آنقدر زیاد است که پیاده‌سازی آن مقرون به صرفه نیست. در بخش بعدی، برای محاسبهٔ توزیع‌شدهٔ رتبه‌بندی روشی را که با روش متمرکز محاسبهٔ رتبه‌بندی تفاوت دارد و دارای پیچیدگی $O(\log |V|)$ دور است، ارائه می‌کنیم.

۴. الگوریتم مبتنی بر شبیه‌سازی مونت‌کارلو و الگوریتم توزیع‌شدهٔ رتبه‌بندی

همان‌طور که در بخش ۲ گفتیم، رتبه‌بندی، توزیع مانای زنجیر مارکوفی است که فضای حالت آن، مجموعهٔ رأس‌های گراف و ماتریس انتقال آن، $P = \epsilon\tilde{P} + (۱ - \epsilon)(۱/n)J$ است. توزیع مانا با توان رساندنِ P محاسبه می‌شود. قضیهٔ زیر بیان می‌کند که با استفاده از فرآیند گام‌برداریِ تصادفی می‌توان رتبه‌بندی را با روش شبیه‌سازی مونت‌کارلو محاسبه کرد که این کار موجب ساده شدن و کاهش پیچیدگیِ محاسبهٔ آن می‌گردد.

قضیه ۱.۴ ([۲]). *فرض کنید گام‌بردارِ تصادفی سادهٔ r حرکت خود را از یک گره تصادفی در شبکه آغاز کند و بعد از ورود به هر گره v، با احتمال $\epsilon - ۱$ به کار خود پایان دهد (در گره v جذب شود) و با احتمال ϵ زنده بماند و در صورت زنده ماندن، یکی از همسایه‌های گره v را به‌تصادف و با توزیع یکنواخت انتخاب کند و وارد آن شود. در این صورت احتمال اینکه r در گره i به کار خود پایان دهد (جذب شود)، برابر با π_i (در قضیهٔ ۳.۲) است.*

بر اساس قضیهٔ ۱.۴، می‌توانیم الگوریتم‌هایی را برای محاسبهٔ رتبه‌بندی بر پایهٔ شبیه‌سازی مونت‌کارلو ارائه کنیم. پنج الگوریتم در [۲] برای این منظور ارائه شده است که در این مقاله تنها به یکی از این الگوریتم‌ها می‌پردازیم. این الگوریتم با سرعت بیشتری نسبت به چهار الگوریتم دیگر به رتبه‌بندی همگرا می‌شود (الگوریتم ۱). در ادامه الگوریتم ۱ را به‌طور دقیق و رسمی به‌صورت یک الگوریتم توزیع‌شده در الگوریتم ۲ بیان می‌کنیم. خروجی الگوریتم ۱ (به‌طور معادل، الگوریتم ۲) با احتمال زیاد به رتبه‌بندی همگرا می‌شود [۱۷]. پیش از بررسی پیچیدگیِ زمانی و پیامیِ الگوریتم ۱ (به‌طور معادل، الگوریتم ۲)، تعریف امید ریاضی و پیشامد با احتمال زیاد و دلیل استفاده از آن را بیان می‌کنیم.

الگوریتم ۱ الگوریتم رتبه‌بندی بر پایهٔ شبیه‌سازی مونت‌کارلو [۲]

ورودی: گراف $G = (V, E)$ که در آن $|V| = n$ است.

۱: از هر یک از رأس‌ها، $\log n$ گام‌بردار تصادفی را ایجاد می‌کنیم.
۲: هر گام‌بردار بعد از ورود به هر گره v، با احتمال $\epsilon - ۱$ به کار خود پایان می‌دهد و با احتمال ϵ زنده می‌ماند. در صورت زنده ماندن، یکی از همسایه‌های گره v را به‌تصادف و با توزیع یکنواخت انتخاب می‌کند و وارد آن می‌شود.
۳: هر گره یک شمارنده (C) دارد که تعداد گام‌بردارهایی را که وارد آن شده‌اند، می‌شمارد.
۴: بعد از آنکه تمام گام‌بردارها به کار خود پایان دادند، رتبهٔ گره v_i را از رابطهٔ $\frac{C_{v_i}}{\sum_{j=۱}^{n} C_{v_j}}$ محاسبه می‌کنیم.

گوییم متغیر تصادفی گسسته X با تابع احتمال $P_X(x)$ دارای امید ریاضی است اگر $\sum_x |x| P_X(x)$ همگرا باشد و در این صورت امید ریاضی X این‌گونه تعریف می‌شود:

$$E(X) = \sum_x x P_X(x).$$

برای تعیین پیچیدگیِ یک الگوریتم تصادفی؛ مثلاً پیچیدگیِ زمانی، می‌توانیم امید ریاضی زمان اجرای آن را محاسبه کنیم. توجه کنید که امید ریاضی ممکن است عددی کوچک باشد ولی حالت‌هایی وجود داشته باشند که پیچیدگیِ زمانی زیادی دارند. لذا در بیشتر مسائل علوم کامپیوتر در مورد محاسبهٔ پیچیدگیِ زمانی یک الگوریتم، به محاسبهٔ امید ریاضی بسنده نمی‌کنند و یک حکم با احتمال زیاد را در مورد پیچیدگی زمانی بیان و ثابت می‌کنند. در علوم کامپیوتر برای مسئله‌ای با اندازهٔ ورودی n[۱] گوییم پیشامد A به احتمال زیاد رخ می‌دهد اگر $P(A) \geq ۱ - \frac{۱}{n^c}$ که در آن، c عددی ثابت است. روشن است که با رشد n، احتمال رخ دادنِ این پیشامد به ۱ میل می‌کند.

قضیه ۲.۴. *فرض کنیم متغیر تصادفی X تعداد گام‌های یک گام‌بردارِ تصادفی در الگوریتم ۱ (به‌طور معادل، الگوریتم ۲) تا جذب در یک گره باشد. در این صورت $E(X) = \frac{۱}{۱-\epsilon}$.*

اثبات. یک گام بردارِ تصادفی مطابق ۱.۴ بعد از ورود به هر گره v، با احتمال $\epsilon - ۱$ جذب می‌شود (به کار خود پایان می‌دهد). بنابراین اگر موفقیت را برابر با جذب یک گام‌بردارِ تصادفی تعریف کنیم، متغیر تصادفی X دارای توزیع هندسی $Ge(۱ - \epsilon)$ خواهد بود. لذا امید ریاضی آن $\frac{۱}{۱-\epsilon}$ است. □

[۱] در رتبه‌بندی رأس‌های یک گراف، اندازهٔ ورودی را تعداد رأس‌های گراف در نظر می‌گیریم و یا در مثال مرتب‌سازی فهرستی از عددها، اندازهٔ ورودی، تعداد عددهای موجود در آن فهرست منظور می‌شود.

الگوریتم ۲ الگوریتم رتبه‌بندی توزیع‌شده [۱۷]

Input (for every node): $|V| = n$, number of walks $K = \log n$ from each node, reset probability ϵ.
Output: PageRank of each node.

[Each node v starts $\log n$ walks. All walks keep moving in parallel until they terminate. The termination probability of each walk is ϵ.]

1: Initially each node v in G creates $\log n$ messages (called coupons) and $C = C_1, C_2, \ldots, C_{\log n}$. Each node also maintains a counter ζ_v (for counting visits of random walks to it).
2: **while** there is at least one (alive) coupon **do**
3: This is i-th round. Each node v holding at least one coupon does the following: Consider each coupon C held by v which is received in the $(i-1)$-th round. Generate a random number $r \in [0, 1]$.
4: **if** $r < \epsilon$ **then**
5: Terminate the coupon C.
6: **else**
7: Select a outgoing neighbor uniformly at random, say u. Add one coupon counter number to T_u^v where the variable T_u^v indicates the number of coupons chosen to move to the neighbor u from v in the i-th round.
8: **end if**
9: Send the coupon's counter number T_u^v to the respective outgoing neighbors u.
10: Every node u adds the total counter number ($\sum_{v \in N(u)} T_u^v$) to ζ_u.
11: **end while**
12: Each node outputs its PageRank as $\frac{\zeta_v \epsilon}{n \log n}$.

قضیه ۳.۴. *الگوریتم۱ (به‌طور معادل، الگوریتم۲) به احتمال زیاد در $O(\log n)$ گام (دور) پایان می‌یابد.*

اثبات. اگر تعداد گام‌هایی را که گام‌بردار iام ($1 \le i \le n \log n$) تا جذب شدن برمی‌دارد با متغیر تصادفی X_i ($x_i = ۱, ۲, ...$) نشان دهیم، بنابر نامساوی مارکوف داریم

$$Pr(X_i > \log n) \le \frac{E(e^{tX_i})}{e^{t \log n}} = \frac{E(e^{tX_i})}{n^t}. \qquad (۱.۴)$$

از سوی دیگر، چون X_i توزیع هندسی دارد، پس

$$E(e^{tX_i}) = \sum_{i=1}^{\infty} e^{it}\epsilon(1-\epsilon)^{i-1} = \frac{\epsilon}{1-\epsilon}\sum_{i=1}^{\infty}\big(e^t(1-\epsilon)\big)^i. \qquad (۲.۴)$$

سری طرف راست رابطهٔ بالا به‌ازای $1 \le t \le \log(\frac{1}{1-\epsilon})$ همگرا است. الگوریتم، زمانی پایان می‌یابد که تمام گام‌بردارها جذب شوند. برای اینکه نشان دهیم همهٔ گام‌بردارها در $O(\log n)$ گام با احتمال زیاد جذب می‌شوند، باید نشان دهیم رابطهٔ زیر به‌ازای یک c ثابت برقرار است:

$$\begin{aligned} &Pr\big(\max\{X_1, X_2, \ldots, X_{n\log n}\} \le \log n\big) \\ &= Pr(\{X_1 \le \log n\} \cap \{X_2 \le \log n\} \cap \cdots \cap \{X_{n\log n} \le \log n\}) \ge 1 - n^{-c}. \end{aligned}$$

ولی به‌ازای $1 \le t \le \log(\frac{1}{1-\epsilon})$ بنابر (۱.۴)،

$$\begin{aligned} &Pr(\{X_1 \le \log n\} \cap \{X_2 \le \log n\} \cap \cdots \cap \{X_{n\log n} \le \log n\}) \\ &\quad = 1 - Pr(\{X_1 > \log n\} \cup \{X_2 > \log n\} \cup \cdots \cup \{X_{n\log n} > \log n\}) \\ &\quad \ge 1 - \sum_{i=1}^{n\log n} \frac{E(e^{tX_i})}{n^t} = 1 - \frac{c' n\log n}{n^t} \end{aligned}$$

که در آن، c' یک عدد ثابت است. چون به‌ازای یک c ثابت و ϵهای بزرگتر از $\frac{1}{2}$، $\frac{c' n\log n}{n^t} \le \frac{1}{n^c}$ (لگاریتم‌ها در مبنای ۲ هستند)، اثبات کامل می‌گردد. □

قضیه ۴.۴. *پیچیدگی پیامیِ الگوریتم ۱ (به‌طور معادل، الگوریتم ۲) از مرتبهٔ $O\big(m\log^2(n)\big)$ است.*

اثبات. الگوریتم ۱ دارای ویژگی مارکوفی است. به بیان دیگر، فرض کنیم گام‌بردارهای rw_1، rw_2، ...، rw_k در دور rام وارد رأس $v \in V$ بشوند. ادامهٔ الگوریتم از رأسِ آغازی و همچنین مسیری که گام‌بردارهای rw_1، rw_2، ...، rw_k تا رأس v پیموده‌اند، مستقل است. لذا لزومی به تفکیک گام‌بردارهای rw_1، rw_2، ...، rw_k از هم نداریم. فرض کنیم $u \in V$ همسایهٔ رأس v باشد. در دور rام کافی است تنها یک عدد که حاوی تعداد گام‌بردارهای رونده از رأس v به رأس u است، از رأس v به رأس u ارسال شود. بنابراین در هر دور، بر روی هر یال تنها دو پیام ارسال می‌گردد که هر کدام حاوی یک عدد کوچکتر یا مساوی n هستند و طول آنها $O(\log n)$ است. بنابر قضیهٔ ۳.۴، الگوریتم پس از $O(\log n)$ دور به احتمال زیاد پایان می‌پذیرد. لذا پیچیدگی پیامیِ الگوریتم برابر با $O\big(m\log^2 n\big)$ است. □

بنابر قضیه‌های ۳.۴ و ۴.۴، الگوریتم ۱ (به‌طور معادل، الگوریتم ۲) به‌ترتیب، دارای پیچیدگیِ زمانی و پیچیدگیِ پیامی $O(\log n)$ و $O(m\log^2 n)$ است. به‌دلیل پیچیدگی زمانیِ لگاریتمی، این الگوریتم قابلیت اجرا بر روی سیستم‌های توزیع‌شده با منابع محدود را دارد. همچنین چون طول پیام‌هایی که در هر دور بر روی هر لینک تبادل می‌شوند از مرتبهٔ $O(\log n)$ است، این الگوریتم بر روی سیستم‌های توزیع‌شده با مدل متراکم نیز قابل پیاده‌سازی است. به‌دلیل استفاده از گام‌برداریِ تصادفی در این الگوریتم، در صورتی که در طی اجرای آن، تعداد کمی از گره‌ها یا لینک‌های ارتباطی خراب شوند، تفاوت چندانی در نتیجهٔ الگوریتم ایجاد نمی‌شود. بنابراین یکی دیگر از سودمندی‌های این الگوریتم مقاوم بودن آن در برابر خرابی گره‌ها و لینک‌ها است.

از آنجا که بسیاری از شبکه‌های دنیای واقعی از *توزیع توانی*[۱] و مدل *مستقل از مقیاس*[۲] پیروی می‌کنند [۳]، علاقه‌مندان می‌توانند الگوریتم توزیع‌شدهٔ رتبه‌بندی را طوری تغییر دهند که برای اجرا بر روی مدل مذکور بهینه شود. همچنین استفاده از گام‌برداریِ تصادفی را در محاسبهٔ پارامترها و ساختارهای دیگر شبکه مانند *بی‌نظمی*[۳] و شناسایی پل‌ها و *حلقه‌ها*[۴] برای پژوهش‌های آینده پیشنهاد می‌کنیم.

مراجع

[1] Aho, A. V., Hopcroft, J. E., Ullman, J. D., *The Design and Analysis of Computer Algorithms*, Addison-Wesley, New York, 1974.

[2] Avrachenkov, K., Litvak, N., Nemirovsky, D., Osipova, N., Monte-Carlo methods in PageRank computation: when one iteration is sufficient, *SIAM Journal on Numerical Analysis*, **45** (2007), 890–904

[3] Barabási, A. L., *Network Science*, Cambridge University Press, Cambridge, 2016.

[4] Brin, S., Page, L., The anatomy of a large-scale hypertextual web search engine, *Computer Networks and ISDN Systems*, **30** (1998), 107–117.

[5] Dixon, J. D., Exact solution of linear equations using p-adic expansions, *Numerische Mathematik*, **40** (1982), 137–141.

[6] Erciyes, K., *Distributed Graph Algorithms for Computer Networks*, Springer-Velrag, New York, Berlin, 2013.

[7] Grimmett, G., Stirzaker, D., *Probability and Random Processes*, Oxford University Press, London, 2001.

[8] Huang, W., Yang, H., A Hadoop job scheduling algorithm based on PageRank, *Metallurgical and Mining Industry*, **8** (2015), 420–425.

[۱] power law distribution [۲] scale free [۳] entropy [۴] cycles

[9] Kim, H., Veciana, G., Yang, X., Venkatachalam, M., Distributed α-optimal user association and cell load balancing in wireless networks, *IEEE/ACM Transactions on Networking*, **20** (2012), 177–190.

[10] Langville, A. N., Meyer, C. D., *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, 2012.

[11] Lawler, F. G., *Introduction to Stochastic Processes*, Chapman and Hall/CRC, 2006.

[12] Mitzenmacher, M., Upfal, E., *Probability and Computing: An Introduction to Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press, London, 2005.

[13] Newman, M., *Networks: An Introduction*, Oxford University Press, London, 2010.

[14] Peleg, D., *Distributed Computing: A Locality-Sensitive Approach*, SIAM, 2000.

[15] Penmatsa, S., Chronopoulos, A.T., Game-theoretic static load balancing for distributed systems, *Journal of Parallel and Distributed Computing*, **71** (2011), 537–555.

[16] Pérez-Rosés, H., Sebé, F., Ribó, J. M., Endorsement deduction and ranking in social networks, *Computer Communications*, **73** (2016), 200–210.

[17] Sarma, A. D., Molla, A., Pandurangan, G., Upfal, E., Fast distributed PageRank computation, *Theoretical Computer Science*, **561** (2015), 113–121.

[18] Sarma, A. D., Molla- A., Nanongkai, D., Pandurangan, G., Tetali, P., Distributed random walks, *Journal of the ACM*, **60** (2013), 1–31.

[19] Shi, S., Yu, J., Yang, G., Wang, D, Distributed page ranking in structured P2P networks, *International Conference on Parallel Processing* (2003), Kaohsiung, 179–186.

[20] Zhu, Y., Ye, S., Li, X., Distributed pagerank computation based on iterative aggregation-disaggregation methods, *CIKM Proceedings of the 14th ACM International Conference on Information and Knowledge Management* (2005), Bremen, 578–585.

حسن حیدری: دانشگاه تهران، دانشکدهٔ فنی، گروه علوم مهندسی
رایانامه: h_heydari@ut.ac.ir

سید محمود طاهری: دانشگاه تهران، دانشکدهٔ فنی، گروه علوم مهندسی
رایانامه: sm_taheri@ut.ac.ir