Journal of Soft Computing and Information Technology (JSCIT)
Babol Noshirvani University of Technology, Babol, Iran
Journal Homepage: jscit.nit.ac.ir
Volume 9, Number 3, Fall 2020, pp. 244-254
Received: 07/06/2019, Revised: 05/27/2020, Accepted: 07/29/2020

Provide an Optimal Model for Finding the Shortest Estimated Paths with Full Graph Coverage

Shekoofe Bostan[1], Ali Mohammad Zare Bidoki[2*]
1- Department of Computer Engineering, Yazd University, Yazd, Iran.
2*- Department of Computer Engineering, Yazd University, Yazd, Iran.
[1] sbostan@stu.yazd.ac.ir, [2*] alizareh@yazd.ac.ir

Corresponding author address: Ali Mohammad Zare Bidoki, Faculty of Computer Engineering, Yazd University, Yazd, Iran, Post Code : 89195-741.

Abstract- Due to the increasing volume of information in social networks and the web, the need for efficient and fast algorithms for analyzing graph content is felt more than ever. One of the most important operations in a graph is to find the shortest path between two nodes, which can have different applications in routing and communication. Classic algorithms are very slow and computationally expensive, nearly impossible, so algorithms using approximation approaches are often used based on Landmark nodes. In this study, four landmark models are introduced. Using innovative methods, landmark nodes are selected for each nodes cluster, the shortest paths are pre-computed and the results are Hashing for direct access. Hence, a fast, efficient and precise result retrieval is possible when an online query is executed. The proposed methods cover the entire graph can reduce the error rate by 0.0016.

Keywords- Shortest Path, Landmark, Approximate Error, Graph, Cluster.

مجله علمی پژوهشی رایانش نرم و فناوری اطلاعات
دانشگاه صنعتی نوشیروانی بابل
صفحه مجله: jscit.nit.ac.ir
جلد ۹، شماره ۳، پائیز ۱۳۹۹ ، صفحه ۲۴۴-۲۵۴
دریافت: ۱۳۹۸/۰۴/۱۵، بازنگری: ۱۳۹۹/۰۳/۰۷، پذیرش: ۱۳۹۹/۰۵/۰۸

ارائه مدلی بهینه جهت یافتن کوتاهترین مسیرهای تخمینی با پوشش کامل گراف

شکوفه بستان [۱]، علی‌محمد زارع‌بیدکی*[۲]
۱- گروه مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران.
*۲- گروه مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران.
[1] sbostan@stu.yazd.ac.ir, [2*] alizareh@yazd.ac.ir

* نشانی نویسنده مسئول: علی‌محمد زارع‌بیدکی ، یزد، دانشگاه یزد، ساختمان فنی ۱، گروه مهندسی کامپیوتر، کد پستی: ۷۴۱-۸۹۱۹۵

چکیده- با توجه به افزایش حجم اطلاعات در شبکه‌های اجتماعی و فضای وب، نیاز به الگوریتم‌های سریع برای آنالیز محتوای گراف بیش از پیش احساس می شود. یکی از مهمترین عملیات‌ها در گراف، یافتن کوتاهترین مسیر بین دو گره است که می‌تواند کاربردهای مختلفی در مسیریابی و ارتباطات داشته باشد. الگوریتم‌های کلاسیک برای حل این مسئله بسیار کند و استفاده از آن‌ها عملا غیرممکن است، بنابراین می‌توان ازالگوریتم‌های تخمینی استفاده کرد که اغلب مبتنی بر لندمارک هستند. در این مقاله چهار مدل تخمینی مبتنی بر لندمارک معرفی می‌گردد که با استفاده از روش‌های ابتکاری، گره‌های لندمارک به صورت برون‌خط انتخاب می‌گردند. همچنین از یک الگوریتم ابتکاری برای خوشه‌بندی گره‌ها استفاده شده و سپس کوتاهترین مسیرها در هر خوشه محاسبه می‌گردد، همچنین از داده‌ساختار هش استفاده می‌شود تا دسترسی به گره‌ها به صورت مستقیم صورت پذیرد و در زمان اجرای پرس‌وجو به صورت برخط، با سرعت و دقت بالا مورد استفاده قرار گیرد. روش‌های پیشنهادی با هدف پوشش کل گراف می‌تواند خطای قابل محاسبه را به 0/0016 کاهش دهد.

واژه‌های کلیدی: کوتاهترین مسیر، لندمارک، خطای تقریبی، گراف، خوشه.

۱- مقدمه

یکی از مسائل اساسی در نظریه گراف‌ها، یافتن کوتاهترین مسیر بین هر دو گره انتخابی است. در طی سال‌های اخیر الگوریتم‌های فراوانی مطرح گردیدند که بر روی گراف‌های کوچک و متوسط در زمان کم و دقت بالا اجرا می‌شوند اما با توجه به اهمیت گراف در دنیای امروز، مسئله مقیاس‌پذیری گراف به عنوان یک چالش مهم مطرح می‌گردد. شبکه‌های اجتماعی بزرگی همچون فیسبوک، توییتر، LinkedIn و حتی فضای وب، همگی مبتنی بر گراف هستند اما هر چه گراف بزرگتر می‌شود سرعت اجرای الگوریتم‌های دقیقی همچون عرض‌اول[1] و دایکسترا[2] کاهش و در نهایت استفاده از آن‌ها غیرممکن خواهد شد. بنابراین الگوریتم‌های تخمینی مطرح می‌گردد که می‌تواند در مقیاس وسیع گراف با سرعت بالا و دقت قابل قبول مورد استفاده قرار گیرد. روش‌های تخمینی ارائه شده به منظور یافتن کوتاهترین مسیر بین دو گره تصادفی، اغلب مبتنی بر لندمارک هستند به این صورت که گرهی به عنوان گره واسط بین سایر گره‌ها در نظر گرفته می‌شود و کوتاهترین مسیرهای تقریبی از این گره می‌گذرد. یکی از چالش‌های اصلی در روش‌های مبتنی بر لندمارک، انتخاب گره‌های مناسب به عنوان لندمارک است که بیشترین ارتباط را با سایر گره‌ها داشته باشند و کوتاه‌ترین مسیرهای موجود از آن گره‌ها بگذرند. در واقع با توجه به این نکته که گراف وب جهت‌دار است و تعداد لندمارک‌ها بسیار کمتر از گره‌ها هستند بنابراین ضرورت انتخاب گره لندمارک مناسبی که بتواند منجر به پوشش حداکثری سایر گره‌ها شود حائز اهمیت است زیرا فاصله هر دو گره بر اساس فاصله آن‌ها با لندمارک‌ها بدست می‌آید و اگر گره به هیچ کدام از لندمارک‌ها متصل نباشد در واقع آن گره

در دسترس نیست. بنابراین دقت، سرعت و پوشش در روش‌های تخمینی مبتنی بر لندمارک، ارتباط مستقیمی با گره‌های لندمارک انتخابی دارد که می‌تواند منجر به افزایش کارایی و اثربخشی شود.

۲- تعاریف اولیه

۲-۱- لندمارک

یافتن کوتاهترین مسیر با استفاده از الگوریتم‌های دقیقی همچون عرض‌اول و دایکسترا در گراف‌های حجیم عملا غیرممکن است. بنابراین نیاز به الگوریتمی که بتواند به سرعت، کوتاهترین مسیر ممکن را تشخیص دهد احساس می‌شود.

با داشتن گراف $G = (V.E)$ ، تعداد گره‌ها را با $n = |V|$ و تعداد یال‌ها را با $m = |E|$ نشان می‌دهیم. بنابراین مسیر $\pi_{s.t}$ با طول $l = |\pi_{s.t}|$ بین دو راس $s.t \in V$ به صورت رابطه ۱ بیان می‌گردد:

$$\pi_{s.t} = (s.u_1.\dots.u_{l-1}.t) \qquad (۱)$$

به صورتی که $\{(s.u_1).\dots.(u_{l-1}.t)\} \subseteq E$ و $\{u_1.\dots.u_{l-}1\} \subseteq V$ است. برای محاسبه مسیر بین دو گره $s.v$ در صورت داشتن کوتاهترین مسیر $\pi_{s.t}$ و $\pi_{t.v}$ می‌توان با محاسبه $\pi_{s.v} = \pi_{s.t} +$ به کوتاهترین مسیر تخمینی بین دو گره مذکور رسید. بنابراین $d(s.v)$ به عنوان تخمینی از کوتاهترین مسیر بین دو گره ذکر شده در نظر گرفته می‌شود.

۲-۲- انتخاب لندمارک

انتخاب گره مناسب به عنوان لندمارک، تاثیر مستقیم بر روی دقت دارد اما از آن به عنوان NP-Hard یاد می‌شود زیرا نمی‌توان یک روش واحد در نظر گرفت که روی تمام گراف‌ها درست جواب بدهد. درواقع ساختار و رفتار گره‌ها و یال‌های گراف در تعیین شیوه مناسب انتخاب لندمارک‌ها بسیار تاثیرگذار خواهد بود. در اغلب روش‌های موجود از رویکرد تصادفی و یا روش‌های مبتنی بر بیشترین درجه استفاده می‌شود که در روش‌های پیشنهادی در این مقاله، بیشترین درجه به عنوان مبنای کار اما با رویکردی متفاوت در نظر گرفته شده است. نکته قابل توجه، تعداد گره‌های انتخابی به عنوان لندمارک است زیرا هرچه تعداد گره‌های لندمارک افزایش یابد منجر به کاهش خطای ناشی از تخمین کوتاهترین مسیر بین گره‌های مبدا و مقصد و در نتیجه باعث افزایش دقت می‌شود اما هزینه بیشتری را در برمی‌گیرد بنابراین باید بین هزینه و زمان تعادل برقرار شود و این کار مستلزم انتخاب روشی مناسب جهت یافتن بهترین گره‌های لندمارک و نحوه استفاده از آن‌هاست.

۲-۳- خطای تخمینی

با فرض داشتن |p| به عنوان طول کوتاهترین مسیر واقعی بین دو گره $u.v$ به صورت $|p| = dist(u.v)$ و همچنین محاسبه طول کوتاهترین مسیر تخمینی تحت عنوان |q|، می توان خطای تخمینی این مسیر را از طریق رابطه 2 محاسبه نمود:

$$error(q) = \frac{|q| - |p|}{|p|} = \frac{|q| - dist(u.v)}{dist(u.v)} \in [0.\infty] \qquad (۲)$$

در این پژوهش، با محاسبه خطای ناشی از هر مدل، میزان دقت و کارایی مورد بررسی قرار می‌گیرد. به منظور ارزیابی، از دیتاست فارسی محک و دیتاست بین‌المللی گوگل استفاده شده که در قسمت ۱-۶ به اختصار به توضیح آن خواهیم پرداخت.

۳- مرور ادبیات

الگوریتم دایکسترا یکی از الگوریتم‌های موفق در زمینه پیمایش گراف به منظور یافتن کوتاهترین مسیرهاست که در سال ۱۹۵۹ توسط دانشمند هلندی ارائه گردید، اما برای استفاده در گراف‌های حجیم قابل استفاده نیست. همچنین الگوریتم‌های دیگری همچون عرض‌اول و فلویدوارشال ارائه گردیدند که مشابه دایکسترا به صورت دقیق به محاسبه کوتاهترین مسیرها می‌پردازند و عملا برای گراف-های حجیم کاربردی ندارند[۲]. به منظور استفاده از روش‌های تخمینی مورد استفاده در گراف‌های حجیم، کاستیلو از روشی مبتنی بر لندمارک استفاده کرد که در آن شیوه انتخاب لندمارک بر مبنای معیارهای مرکزیت گره‌هاست[۳] همچنین گولدبرگ از مدلی مبتنی بر لندمارک استفاده کرد که در آن فضای جستجو بر مبنای طول مسیر هرس[۳] می‌شود[۴]. او همچنین در روشی دیگر با انتخاب زیرمجموعه‌ای از گراف و استفاده از الگوریتم دایکسترا و A* به صورت ابتکاری، سعی کرد به صورت برون‌خط به محاسبه کوتاهترین مسیرها بپردازد [5]. کوین گرنت مدل جدیدی تخت عنوان LPI[4] که مبتنی بر لندمارک است ارائه کرد به این صورت که هر گره لندمارک، کوتاهترین مسیر از خود تا سایر گره‌ها را ذخیره می‌کند سپس بر اساس اشتراک نقاط بین لندمارک‌ها، کوتاهترین مسیر بین دو گره بدست می‌آید[۶].

یکی از الگوریتم‌های موفق در این زمینه الگوریتم TreeSketch است که توسط گوبیچف وهمکارانش معرفی و به نتایج ارزشمندی دست یافت. این الگوریتم شامل فاز برون خط به صورت پیش‌پردازش و فاز برخط برای دریافت کوئری و یافتن کوتاهترین مسیر بین دو گره مبدا و مقصد است. با فرض داشتن V به عنوان مجموعه نقاط گراف

جهت‌دار، در صورتی که $n = |V|$ و $r := \lfloor log(n) \rfloor$ باشد، $r + 1$ مجموعه از گره‌ها به عنوان گره‌های seed در نظر گرفته و $s_0, s_1, s_2 \dots s_r$ نامیده می‌شود. سپس برای هر مجموعه s_i و هر گره v از $v \in V$ کوتاهترین مسیرهای ممکن با استفاده از الگوریتم عرض-اول محاسبه می‌گردد. در گام بعد برای هر گره، نزدیک‌ترین seed به عنوان لندمارک l_1 در نظر گرفته می‌شود و همچنین از گره v به مجموعه s_i کوتاهترین مسیر با کمک الگوریتم عرض‌اول محاسبه می‌شود و نزدیکترین seed به عنوان لندمارک l_2 در نظر گرفته می-شود. در نهایت گره‌های l_1 و l_2 به عنوان لندمارک‌های گره v تعیین می‌گردند. فرایند انتخاب گره‌های seed و محاسبه کوتاهترین مسیرها $\mathcal{K}$ بار تکرار می‌شود بنابراین برای هر گره $v \in V$ به $2r\mathcal{K}$ لندمارک و مسیر (به همراه هزینه محاسبه الگوریتم عرض‌اول برای هر مجموعه seed) بدست می‌آید.

در واقع مجموعه گره‌های لندمارک هر گره متفاوت از گره دیگر است بنابراین داده‌های بدست آمده برای هر گره v شامل $2r\mathcal{K}$ لندمارک و مسیر با نام sketch(v) ذخیره می‌گردد. در فاز اجرای برخط، با دریافت پرس‌وجوی کاربر، sketch های دو گره مبدا و مقصد جستجو می‌شوند. گره مبدا به عنوان ریشه درخت در نظر گرفته می‌شود و بر اساس مسیرهای یافته، $TreeSketch(v)$ شکل می‌گیرد. همین ساختار برای گره مقصد نیز اعمال می‌شود. در صورتی که L مجموعه لندمارک‌های دو $sketch$ را در برگیرد، برای هر لندمارک مشترک در مجموعه L ، مسیر از گره مبدا به آن لندمارک و همچنین مسیر از آن لندمارک به گره مقصد را یافته و این کار برای تمامی لندمارک‌های مشترک انجام می‌گیرد و در یک صف اولویت به صورت نزولی بر اساس کوتاهترین مسیر ذخیره می-گردد. سپس به منظور بهبود نتایج در گام بعد، حلقه‌های ایجاد شده در مسیر حذف می‌شوند و مسیرهای میانبر در مسیر بدست آمده، شناسایی می‌گردند.[۷].

همچنین لیلی‌چاوو از مجموعه‌ای از درختان پوشا[۵] برای یافتن تخمینی کوتاهترین مسیر استفاده کرد به این صورت که با استفاده از توزیع گراف روی چندین ماشین، و ایجاد درختان پوشا در هر ماشین، کوتاهترین مسیرها محاسبه می‌گردد و سپس نتایج با هم ترکیب می‌شود[۸].

یکی دیگر از الگوریتم‌های مطرح در این حوزه توسط کیاوو و همکارش ارائه گردید ، آنها در الگوریتم خود از شیوه متفاوتی برای انتخاب گره لندمارک استفاده کردند که وابسته به کوئری کاربر است. در واقع در این روش که برگرفته از ایده الگوریتم sketch است، به جای آنکه گره‌های لندمارک از قبل و به صورت کلی و عمومی انتخاب شوند، بعد از دریافت پرس‌وجوی کاربر، با انتخاب لندمارک-های محلی، سعی می‌شود کوتاهترین مسیرها بین آن گره‌ها بدست آید. در این مقاله سه الگوریتم GLS[6] ، LLS[7] ، و LS[8] ارائه گردید که نتایج ارزشمندی به همراه داشت. در الگوریتم GLS بر اساس مرکزیت گره‌ها، ۵۰ گره به عنوان لندمارک در نظر گرفته شد و کوتاهترین مسیر بین گره‌ها و لندمارک‌ها محاسبه گردید. اما در الگوریتم LLS بعد از دریافت کوئری کاربر و یافتن مسیر بین گره مبدا و مقصد از طریق لندمارک‌های بدست آمده از راهکار قبلی، در صورتی که گره مشترکی بین گره مبدا و مقصد و لندمارک وجود داشته باشد، آن گره به عنوان لندمارک محلی انتخاب می‌شود، به عنوان مثال اگر هدف یافتن کوتاهترین مسیر بین گره a و b باشد و با کمک لندمارک l_1 به گره a و b مسیر وجود داشته باشد و در این میان در هر دو مسیر به صورت مشترک گره c دیده شود پس c به عنوان لندمارک محلی انتخاب می‌شود. در واقع هر دفعه گره لندمارک به عنوان ریشه درخت در نظر گرفته می‌شود و مسیر از ریشه یا برگ که گره مبدا یا مقصد است طی می‌شود . در واقع در این روش، هدف یافتن دورترین گره از ریشه یعنی لندمارک است که در مسیر رسیدن به گره مبدا و مقصد مشترک باشد که به عنوان لندمارک محلی در نظر گرفته شود تا به دقت قابل قبولی برسد. در راهکار سوم الگوریتم LS معرفی می‌گردد که از جستجوی محلی برای محدودکردن فضای جستجوی برخط استفاده می‌کند که نتایج بهتری نسبت به دو الگوریتم قبلی خود یعنی GLS و LLS ارائه می‌دهد. همچنین در پیاده‌سازی الگوریتم‌های موجود، از فشرده-سازی گراف به منظور کاهش فضای ذخیره‌سازی و بهبود کارایی استفاده شده است[۹].

وولدیمر و همکارانش با استفاده از تعریف گره‌های لندمارک از قبل تعریف شده و هرس مسیرهای طولانی، راه‌حلی ارائه دادند[۱۰]. فنگ و همکارش الگوریتم جدیدی مبتنی بر لندمارک بر اساس زیرمجموعه‌ای از گراف و به صورت محلی تحت عنوان LSQ[۹] ارائه کردند که منجر به افزایش دقت و کاهش خطای تخمینی می-شود[۱۱]. همچنین کیگ‌دوون یک الگوریتم جدید بر مبنای PLL[۱۰] بر اساس هرس کردن لندمارک‌های برچسب‌گذاری شده به عنوان هاب[۱۱] ارائه کرد که در دو سطح به صورت موازی از روش درختی SPT[۱۲] به منظور یافتن کوتاهترین مسیر بین گره‌ها استفاده می‌کند[۱۲].

دو الگوریتم TreeSketch و LS در قسمت ۶-۳ به عنوان دو الگوریتم مطرح با الگوریتم‌های ابتکاری این مقاله از نظر کارایی و عملکرد مقایسه می‌گردند.

۴- روش‌های پیشنهادی

در این پژوهش چهار مدل ابتکاری جهت یافتن کوتاهترین مسیر بین دو گره تصادفی، پیاده‌سازی و مورد بررسی قرار گرفته است. در هر روش از تکنیک خوشه‌بندی ابتکاری استفاده شده و گره‌های لندمارک به شیوه متفاوت از یکدیگر و به صورت برون‌خط انتخاب شده‌اند. در واقع فرایند محاسباتی به دو فاز تقسیم می‌گردد:

الف) فاز پیش‌پردازش یا برون‌خط[۱۳]

۱- انتخاب گره لندمارک به شیوه مرسوم (الگوریتم اول و دوم) و یا ابتکاری (الگوریتم سوم و چهارم)

۲- ایجاد خوشه به شیوه ابتکاری

۳- محاسبه کوتاهترین مسیرها در هر خوشه مابین گره‌ها و همچنین نسبت به گره‌ها و لندمارک(ها)

۴- محاسبه کوتاهترین مسیر بین لندمارک‌ها

ب) فاز برخط

۱- دریافت پرس‌وجو شامل دو گره مبدا و مقصد جهت یافتن کوتاهترین مسیر

۲- یافتن خوشه هر گره

۳- محاسبه کوتاهترین مسیر بهینه بر اساس مسیرهای بدست آمده از گره به مرکز خوشه، سایر لندمارک‌ها و سایر گره‌ها

۴-۱- فاز پیش‌پردازش[۱۴]

با توجه به اهمیت زمان در تخمین کوتاهترین مسیر بین رئوس در کاربردهای مختلف، می‌توان از یک فاز پیش‌پردازش استفاده کرد و بخشی از محاسبات را به صورت برون‌خط انجام داد تا در زمان اجرا بتوان تنها با استفاده از یک یا چند پرس‌وجوی ساده، در کمترین زمان ممکن به نتیجه مطلوب رسید. بنابراین مهمترین قسمت در این پژوهش که انتخاب گره لندمارک وخوشه‌بندی است، به صورت برون‌خط و از قبل محاسبه می‌گردد.

۴-۲- راهکار اول (SPLL[15])

۴-۲-۱- انتخاب گره لندمارک

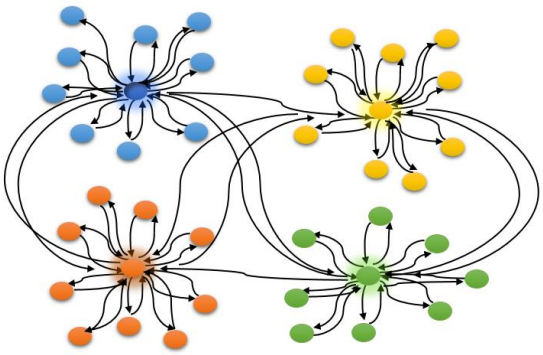
یکی از پارمترهای مهم در انتخاب گره لندمارک، درجه ورودی و خروجی آن گره است زیرا هرچه درجه ورودی یک گره بزرگتر باشد یعنی آن گره اهمیت بیشتری داشته که توسط صفحات زیادی به آن اشاره شده، همچنین درجه خروجی یک صفحه بیانگر میزان دسترسی آن گره به صفحات دیگر است که اگر بزرگ باشد می‌توان آن گره را به عنوان هاب در نظر گرفت که صفحات زیادی را می‌شناسد و به آن‌ها دسترسی دارد. بنابراین به منظور انتخاب گره لندمارک، در ابتدا بیشترین درجه ملاک انتخاب قرار می‌گیرد و در این روش ۲۰ گره لندمارک با بیشترین درجه ورودی و خروجی انتخاب می‌گردند.

۴-۲-۲- خوشه‌بندی

در این مرحله به منظور خوشه‌بندی، به تعداد لندمارک‌های بدست آمده از مرحله قبل، خوشه ایجاد می‌گردد و هر لندمارک به عنوان مرکز هر خوشه در نظر گرفته می‌شود. بنابراین با وجود ۲۰ گره لندمارک، ۲۰ خوشه ایجاد می‌گردد، سپس به ازای هر گره از گراف، کوتاهترین مسیر آن گره با تمامی لندمارک‌ها محاسبه و گره در خوشه‌ای قرار می‌گیرد که به مرکز آن خوشه نزدیک‌تر باشد. در نهایت، بعد از انجام این مرحله برای تمامی گره‌های درون گراف وب، تمام گره‌ها در خوشه مناسب قرار می‌گیرند. با توجه به این شیوه ابتکاری، می‌توان دریافت که تعداد گره‌ها در هر خوشه متفاوت است و ممکن است در یک خوشه تعداد گره‌های بیشتری قرار گرفته باشند که می‌تواند به دلیل ارتباط بیشتر آن لندمارک به عنوان مرکز خوشه با تعداد زیادی از گره‌ها باشد.

۴-۲-۳- محاسبه کوتاهترین مسیر

در این مرحله کوتاهترین مسیر درون هر خوشه مابین گره‌های داخل آن خوشه محاسبه می‌گردد. این مسیر می‌تواند مستقل از لندمارک درون آن خوشه باشد و یا ممکن است مسیر مورد نظر از لندمارک آن خوشه گذر کند. بنابراین کوتاهترین مسیر بین تمامی گره‌های درون هر خوشه با یکدیگر و با لندمارک داخل آن خوشه به عنوان مرکز خوشه، محاسبه و ذخیره می‌گردد. شکل ۱ بیانگر نمایی از الگوریتم اول با نام SPLL است که نشان می‌دهد لندمارک هر خوشه به عنوان یک لندمارک محلی، در ارتباط نزدیک با گره‌های آن خوشه



شکل ۱: نمایی از SPLL

است. مجموع زمان اجرای برخط برای ۱۰ هزار جفت گره به ترتیب برابر با ۹/۰۱۵ و ۹/۰۲۱ میلی‌ثانیه به ازای مجموعه دادگان گوگل و محک محاسبه می‌گردد ضمن اینکه خطای تخمینی برابر با ۰/۰۰۶۲ و ۰/۰۰۴۰ دست می‌آید.

۴-۳- راهکار دوم (SPGL[16])

در این روش مشابه SPLL ، ۲۰ گره لندمارک از روی کل گراف انتخاب و سپس خوشه‌بندی بر اساس قرار دادن هر لندمارک به عنوان مرکز هر خوشه صورت می‌پذیرد. سپس گره‌ها بر اساس میزان نزدیکی به هر مرکز خوشه، در خوشه مناسب قرار می‌گیرند. اما تفاوت SPGL با SPLL در این است که در انتها، برای هر خوشه، ۱۹ لندمارک دیگر یعنی مراکز سایر خوشه‌ها، به عنوان گره، به آن خوشه تزریق می‌شود. در واقع در این راهکار در هر خوشه ۲۰ لندمارک وجود دارد اما تنها یکی از آن‌ها، به عنوان مرکز خوشه در ارتباط نزدیک با تمامی گره‌های آن خوشه است و سایر لندمارک‌های اضافه شده به این خوشه، به منظور ایجاد مسیرهای بهتر نسبت به الگوریتم SPLL و در نتیجه افزایش دقت در یافتن مسیرهای بهینه-تر، به هر خوشه تزریق می‌شوند. سپس کوتاهترین مسیر بین هر دو گره داخل هر خوشه محاسبه می‌گردد.

بدیهی است که با توجه به افزودن تمامی گره‌های لندمارک به هر خوشه، کوتاهترین مسیر بدست آمده برای هر دو گره در آن خوشه، می‌تواند از طریق مرکز آن خوشه و یا هر یک از ۱۹ لندمارک تزریق شده به آن خوشه و یا حتی مستقل از لندمارک‌ها و از طریق سایر گره‌های درون همان خوشه باشد. بنابراین SPGL منجر به افزایش دقت نسبت به SPLL با توجه به امکان انتخاب‌های متعدد برای مسیرهای پیش‌رو و انتخاب بهترین و کوتاهترین مسیر ممکن بین دو گره مورد نظر خواهد بود.

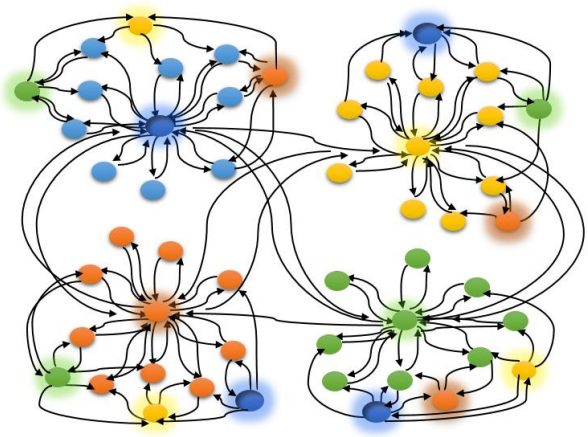
شکل ۲ نمایی از SPGL را در مقیاس کوچک با چهار لندمارک نشان می‌دهد.

مجموع زمان اجرای برخط الگوریتم SPGL برابر با ۹/۰۹۲ و ۹/۱۰۱ به ازای مجموعه دادگان گوگل و محک است ضمن اینکه متوسط خطای تخمینی به ۰/۰۰۴۲ و ۰/۰۰۲۷ کاهش می‌یابد که به دلیل افزودن مراکز سایر خوشه‌ها در هر خوشه و در نتیجه امکان انتخاب مسیرهای بهینه‌تر است.

۴-۴- راهکار ابتکاری با هدف پوشش کل گراف

۴-۴-۱- انتخاب گره لندمارک

SPGL توانست با دقت قابل قبول، کوتاهترین مسیرها بین هر دو گره تصادفی را بیابد اما در صورتی که هدف پوشش تمامی گره‌های درون گراف باشد می‌توان از الگوریتم ابتکاری جدیدی برای انتخاب



شکل ۲: نمایی از SPGL

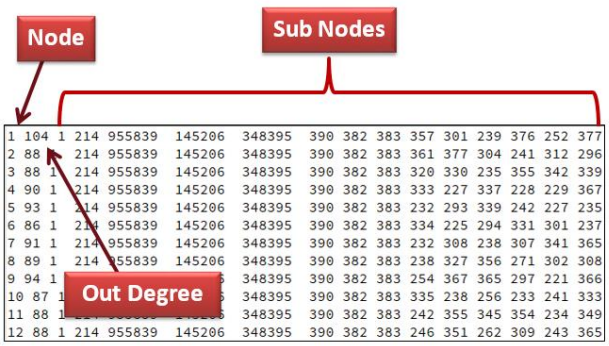
گره لندمارک استفاده کرد تا هم پوشش بهتری از کل گراف داشته باشد و همچنین میزان خطای ناشی از تخمین کوتاه‌ترین مسیر بین دو گره را کاهش دهد.

مراحل ذیل روال انتخاب گره لندمارک در این راهکار با استفاده از یک شیوه نوین و ابتکاری است:

گام اول: ابتدا بر اساس هر گره در گراف، درجه خروجی آن گره و همچنین مجموعه گره‌های زیرمجموعه[17] یا فرزند آن گره بدست می‌آید و در فایلی مشابه شکل ۳ ذخیره می‌شود سپس گره‌ها بر اساس بیشترین درجه خروجی به صورت نزولی مرتب می‌شوند.

گام دوم: گره اول با بیشترین درجه به عنوان لندمارک در نظر گرفته می‌شود و به گره‌های زیرمجموعه‌ی آن گره، برچسب "دیده شد[18]" اضافه می‌شود.

گام سوم: الگوریتم با مشاهده گره بعدی در صورتی که تمام گره‌های زیرمجموعه‌ی آن گره قبلا دیده شده، از آن گره گذر می‌کند ولی

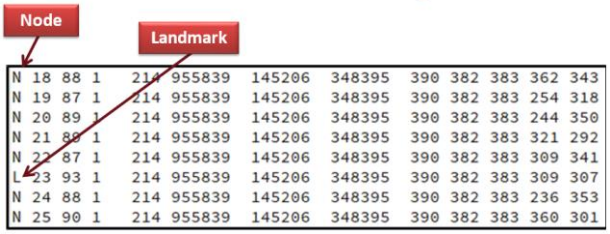


شکل ۳ : بخشی از فایل خروجی

در صورت مشاهده‌ی تنها یک زیرمجموعه‌ی دیده نشده، آن گره را به عنوان لندمارک جدید به مجموعه لندمارک‌های خود اضافه می-کند.

در دور دوم، تمام مراحل ۱ تا ۳ بر مبنای درجه ورودی محاسبه می‌گردد با این تفاوت که به جای گره‌های فرزند، از گره‌های والد استفاده شده است و در صورتی که خود گره، برچسب "دیده شد " از دور اول بدست نیاورده باشد والدهای او بررسی می‌شود و در صورتی که هیچ والدی برچسب "دیده شد" نداشته باشد در این صورت آن گره نیز به مجموعه لندمارک‌ها اضافه می‌شود وگرنه از آن گره گذر می‌شود که در شکل ۴ قابل مشاهده است.

در این روش کلیه گره‌های گراف توسط لندمارک‌ها پوشش داده می‌شوند اما تعداد لندمارک‌هایی بیشتری بدست می‌آید که به دلیل



شکل ۴ : تفکیک گره‌ها از لندمارک‌ها در گراف

شرایط یکسان بین مدل‌های مذکور، ۲۰ لندمارک اول از مجموع لندمارک‌های بدست آمده، انتخاب و مورد استفاده قرار می‌گیرد.

در مرحله بعد خوشه‌بندی بر اساس لندمارک‌های بدست آمده از این روش به عنوان مراکز خوشه و با هدف پوشش کامل گراف، در نظر گرفته می‌شود و گره‌ها در خوشه مناسب بر اساس کمترین فاصله به مرکز آن خوشه، قرار می‌گیرند. سپس کوتاهترین مسیر بین هر دو گره درون هر خوشه مشابه قبل محاسبه می‌گردد. این الگوریتم SPLLC[19] نامیده می‌شود و نتایج حاصل از این راهکار، دقت بیشتری به همراه دارد ضمن اینکه، پوشش بهتری از تمامی گره‌ها نسبت به SPLL و SPGL بدست می‌آورد که به دلیل انتخاب لندمارک‌ها با الگوریتم جدید ابتکاری است. در نتیجه تمامی گره‌ها در این راهکار، در صورت وجود مسیر، به هم دسترسی و ارتباط خواهند داشت و کوتاهترین مسیر بین هر دو گره حتی با ارتباطات بسیار اندک و حتی دور از یکدیگر، به سهولت محاسبه و در دسترس خواهد بود.

اما رویکرد SPGLC[20] ، مشابه الگوریتم SPGL از تزریق گره‌های لندمارک سایر خوشه‌ها در هر خوشه و محاسبه کوتاهترین مسیر بین گره‌های درون هر خوشه از طریق لندمارک یا بدون آن‌هاست.

میزان خطای تخمینی بدست آمده در SPLLC برابر با ۰/۰۰۴۹ و ۰/۰۰۳۱ به ترتیب برای دو مجموعه دادگان گوگل و محک است ضمن اینکه مجموع زمان اجرای برخط آن به ۹/۰۰۳ و ۹/۰۱۰ می‌رسد که بیانگر افزایش دقت نتایج بدست آمده نسبت به الگوریتم SPLL با ساختار مشابه است اما در SPGLC مجموع زمان اجرای برخط برابر با ۹/۰۸۵ و ۹/۰۸۸ است در حالی‌که خطای بدست آمده به ۰/۰۰۳۳ و ۰/۰۰۱۶ کاهش می‌یابد. در نتیجه SPGLC بهترین نتایج را در بین الگوریتم‌های پیشنهادی ما فراهم می‌سازد ضمن اینکه پوشش کاملی از کل گراف ارائه می‌دهد که قابل توجه است.

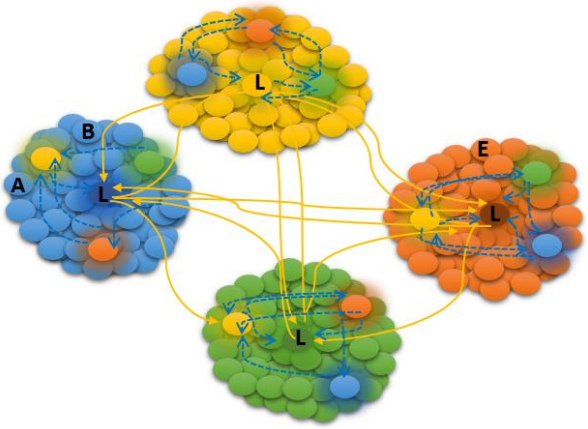
با توجه به اینکه تعداد گره‌های لندمارک و همچنین تعداد خوشه‌ها در چهار الگوریتم پیشنهادی یکسان است در نتیجه مدت زمان محاسبه و میزان پیچیدگی در فاز پیش‌پردازش، تفاوت چندانی ندارد. به منظور بهبود نتایج می‌توان نشان داد هر چه تعداد گره‌های لندمارک بیشتر باشد دقت افزایش می‌یابد زیرا منجر به افزایش تعداد مسیرهای موجود بین گره‌ها می‌گردد. اما توجه به این نکته ضروری است که تعداد لندمارک‌ها با زمان و پیچیدگی محاسباتی ارتباط مستقیم دارد پس به نوعی باید بین تعداد لندمارک‌ها و هزینه محاسباتی و زمان توازن برقرار کرد. به بیان دیگر با توجه به کاربرد مسئله می‌توان در بعضی موارد، دقت را بر هزینه ارجعیت داد و با انجام محاسبات بیشتر در زمان طولانی‌تر، به دقت بیشتری دست یافت.

۵- فاز برخط(پرس‌وجو)

با توجه به شکل ۵ ، در زمان جستجو، دوحالت ممکن است رخ دهد.

الف) اگر گره مبدا و مقصد مشابه گره A و B ، در یک خوشه قرار داشته باشند:

- در این حالت بعد از جستجو در خوشه مربوطه، کوتاهترین مسیر که ممکن است از طریق مرکز خوشه و یا از طریق سایر



شکل ۵ : نمایی از ارتباط گره‌ها

لندمارک‌های تزریق شده به آن خوشه و یا مستقل از لندمارک-ها و از طریق گره‌های درون آن خوشه بدست آمده باشد برگردانده می‌شود.

ب) اگر گره مبدا و مقصد مشابه گره A و E در دو خوشه مجزا باشند:

- در این صورت ابتدا گره A و E به صورت مستقل در خوشه مربوط به خود جستجو و کوتاهترین مسیر هریک با مرکز خوشه خود استخراج می‌شود. در نهایت مجموع این دو مسیر به عنوان یک مسیر کاندید محاسبه می‌گردد.
- در صورتی که از الگوریتم‌های SPGL یا SPGLC استفاده شده باشد، به دلیل افزودن تمامی لندمارک‌ها در هر خوشه، کوتاهترین مسیر هر یک از دو گره A و E با مرکز خوشه دیگری که به عنوان گره لندمارک در خوشه آن گره اضافه شده به صورت مستقل بدست می‌آید. در نتیجه در این مرحله هم دو مسیر کاندید جدید بدست می‌آید.
- در نهایت کوتاهترین مسیر از بین مسیرهای کاندید بدست آمده مطابق رابطه ۳ تعیین می‌گردد.

$$\begin{aligned}\text{Min}(\ \text{A} \rightarrow\ & \text{CentroidLandmark}_A \\ & \rightarrow \text{CentroidLandmark}_E \\ & \rightarrow E\ .\end{aligned} \qquad (۳)$$

$$\text{A} \rightarrow\ \text{CentroidLandmark}_A \rightarrow E\ .$$

$$\text{A} \rightarrow\ \text{CentroidLandmark}_E \rightarrow E\ )$$

۶- نتایج تجربی

۶-۱- دیتاست

در این پژوهش از دیتاست فارسی محک به عنوان بستری برای انجام آزمایش‌های مختلف و بررسی الگوریتم‌ها استفاده شده است. همچنین از دیتاست بین‌المللی گوگل نیز برای مقایسه نتایج در وب فارسی و انگلیسی و ارزیابی نتایج با سایر الگوریتم‌ها استفاده گردیده است.

دیتاست فارسی محک شامل یک پیکره‌ی استاندارد یک میلیون سندی و قسمت‌های متعدد دیگری است. مجموعه محک وب دات‌آی‌آر توسط گروه تحقیقاتی پایگاه‌داده‌ی دانشگاه تهران و با حمایت مرکز تحقیقات مخابرات ایران تهیه شده است[۱] و دیتاست گوگل، توسط کمپانی گوگل جهت استفاده در مسابقه‌ی برنامه-نویسی گوگل تهیه و منتشر گردید[۱۳] که در هر دو مجموعه، توزیع درجه ورودی و خروجی صفحات وب از قانون قدرت[۲۱] پیروی می‌کند و به صورت رابطه ۴ بیان می‌شود.

$$p(x) = x^{-\gamma} \qquad (۴)$$

جدول ۱ اطلاعات تکمیلی از دو دیتاست مورد استفاده را بیان می-کند. همان‌گونه که از این جدول مشخص است، تعداد یال‌ها که با $|E|$ نشان داده می‌شود، در وب فارسی بسیار بیشتر از وب انگلیسی

جدول ۱: توصیف دیتاست‌ها

عنوان	محک (فارسی)	گوگل
$\|V\|$	۹۹۷۴۶۲	۸۷۵۷۱۳
$\|E\|$	۴۰۹۲۴۴۲۹	۵۱۰۵۰۳۹
متوسط درجه ورودی	۴۷	۷
متوسط درجه خروجی	۴۷	۶
آلفای درجه ورودی	۱.۳۹۳۴۴	۱.۴۶۹۵۷
آلفای درجه خروجی	۱.۲۴۷۵۶	۱.۴۰۳۶۴
سیگما درجه ورودی	۰.۰۰۰۴۲	۰.۰۰۰۵۵
سیگما درجه خروجی	۰.۰۰۰۲۶	۰.۰۰۰۴۶
زمان اجرای الگوریتم عرض‌اول	۱۱ ثانیه	۴ ثانیه

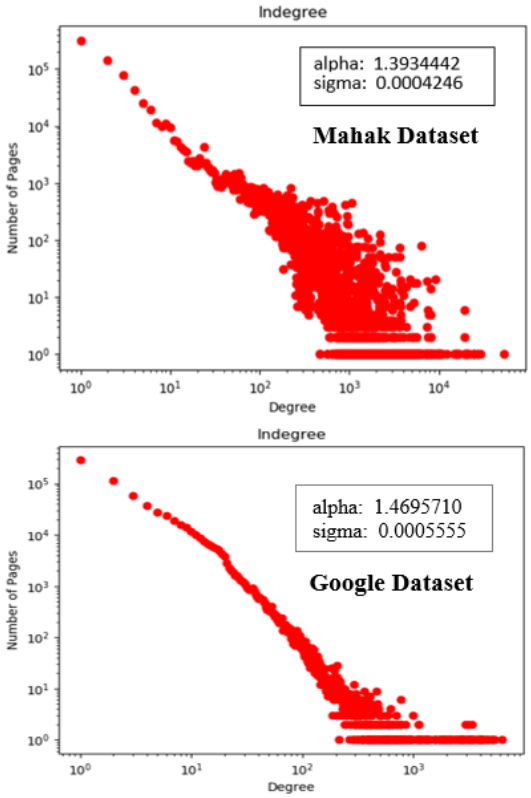
است و این یعنی درجه صفحات در گراف وب فارسی به مراتب، بیشتر از وب انگلیسی است. در نتیجه در صورت استفاده از الگوریتم دقیق عرض‌اول، مدت زمان پاسخگویی بر روی دیتاست محک برابر با ۱۱ ثانیه و برای دیتاست گوگل، ۴ ثانیه خواهد بود. همچنین شکل ۶ و ۷ به ترتیب بیانگر نمودار احتمال توزیع درجه ورودی و خروجی صفحات وب است .

بنابراین با توجه به این‌که گراف وب از قانون قدرت پیروی می‌کند، در صورتی که بتوان از الگوریتم‌های ابتکاری که مبتنی بر درجه‌ی ورودی و خروجی صفحات وب هستند، جهت یافتن بهترین گره‌های لندمارک استفاده کرد، می‌توان به نتایج ارزشمندی دست یافت که منجر به یافتن کوتاهترین مسیرهای بهینه بین دو گره در کمترین زمان ممکن شود.

۶-۲- تحلیل و مقایسه راهکارهای پیشنهادی

جدول ۲ خلاصه‌ای از مراحل انجام کار برای هر الگوریتم و مدت زمان اجرای فرایندها در فاز پیش‌پردازش به صورت برون‌خط و همچنین اجرای پرس‌وجوها به صورت برخط را نشان می‌دهد ضمن این‌که خطای تخمینی و پوشش گراف نیز مطابق جدول۲ می‌تواند در انتخاب مدل مناسب جهت کاربردهای مختلف تعیین کننده باشد. به دلیل آن‌که انتخاب گره لندمارک در وب فارسی و انگلیسی

بر اساس چهار الگوریتم پیشنهادی، مبتنی بر درجه ورودی و خروجی گره‌ها است لذا می‌توان از مدل‌های فوق در هر دو دیتاست

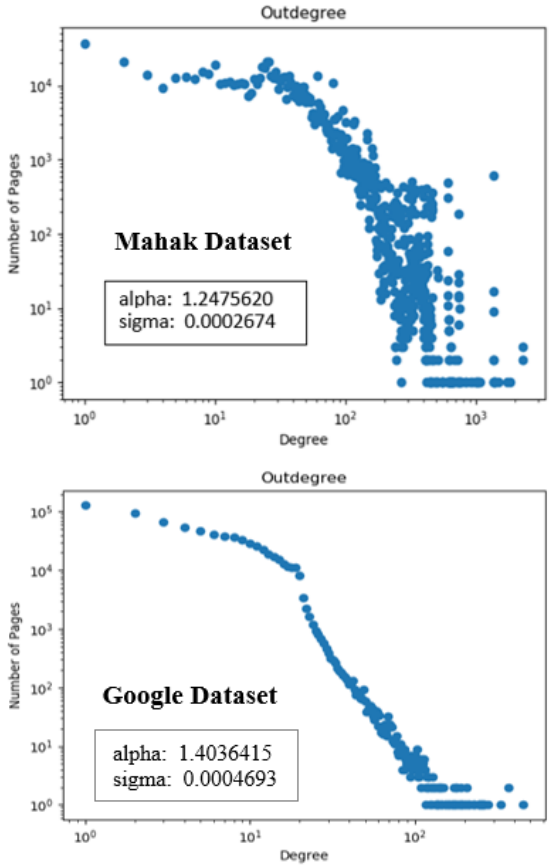


شکل ۶ : نمودار پیروی گرافها از توزیع قانون قدرت برای درجه ورودی

استفاده کرد. در واقع با توجه به این‌که مشخصه مشترک گراف وب، پیروی درجه صفحات وب از قانون قدرت است، لذا رفتار مشابهی بین صفحات وب وجود دارد که می‌تواند در وب فارسی و انگلیسی با تفاوت‌های رفتاری جزئی بیان گردد. بنابراین استفاده از الگوریتم-های پیشنهادی ما در گراف‌های مشابه که مبتنی بر قانون قدرت باشند امکان‌پذیر است. به همین دلیل به منظور ارزیابی و مقایسه کارایی الگوریتم‌های پیشنهادی با سایر الگوریتم‌های موجود، از گراف وب گوگل در کنار گراف فارسی محک استفاده شد تا تاثیر الگوریتم‌های ابتکاری بر روی دادگان غیرفارسی و بین‌المللی بررسی گردد و کارای و عملکرد آن‌ها با سایر الگوریتم‌های مطرح در این حوزه، مورد ارزیابی قرار گیرد.

با توجه به نتایج جدول ۲، کوتاهترین مسیر بهینه بین ۱۰ هزار جفت گره به واسطه چهار الگوریتم پیشنهادی، در زمان تقریبی ۹ میلی‌ثانیه بر روی هر یک از دو دیتاست محک و گوگل به صورت مستقل محاسبه و ارائه می‌گردد. همچنین متوسط خطای تخمینی بدست آمده در الگوریتم‌های فوق، در بدترین حالت ۰/۰۰۶۲ و در بهترین حالت به ۰/۰۰۱۶ می‌رسد که قابل توجه است.

در الگوریتم پیشنهادی اول با نام SPLL، ۲۰ گره لندمارک بر مبنای بیشترین درجه در کل گراف انتخاب گردید و سپس با استفاده از تکنیک خوشه‌بندی به صورت ابتکاری، هر گره در خوشه مناسب بر اساس میزان نزدیکی به مرکز آن خوشه که یکی از این ۲۰ لندمارک بود، قرار گرفت. در واقع با وجود ۲۰ گره لندمارک، ۲۰ خوشه ایجاد گردید و هر لندمارک در مرکز یک خوشه جای گرفت و سپس گره‌ها بر اساس فاصله از مرکز خوشه، در خوشه مناسب قرار داده شدند. سپس درون هر خوشه مستقل از سایر خوشه‌ها، کوتاهترین مسیر



شکل ۷ : نمودار پیروی گرافها از توزیع قانون قدرت برای درجه خروجی

بین هر دو گره محاسبه و ذخیره گردید. لازم به ذکر است مسیر بدست آمده مابین دو گره درون یک خوشه می‌تواند از طریق لندمارک درون آن خوشه و یا مستقل از آن محاسبه گردیده باشد. با توجه به اینکه تمامی این مراحل در فاز پیش‌پردازش و به صورت برون‌خط محاسبه می‌گردد لذا در فاز برخط و در زمان دریافت پرس‌وجوی کاربر، تنها کافیست دو گره مورد نظر جستجو گردند زیرا در صورتی که هر دو گره در یک خوشه باشند کوتاهترین مسیر آنها قبلا در فاز پیش‌پردازش محاسبه شده و دیگر نیازی به محاسبه اضافی نیست. اما در صورتی که دو گره در دو خوشه مجزا قرار گرفته باشند لذا کوتاهترین مسیر هر گره با مرکز خوشه خود را یافته و از

مجموع فاصله آن‌ها بر اساس فاصله دو لندمارک نسبت به یکدیگر استفاده کرده و کوتاهترین مسیر بین دو گره پرس‌وجو، محاسبه و ارائه می‌گردد. در نهایت مجموع زمان اجرای برخط بر روی دادگان گوگل و محک به ۹/۰۱۵ و ۹/۰۲۱ میلی‌ثانیه و متوسط خطای تخمین ۰/۰۰۶۲ و ۰/۰۰۴۰ بدست می‌آید.

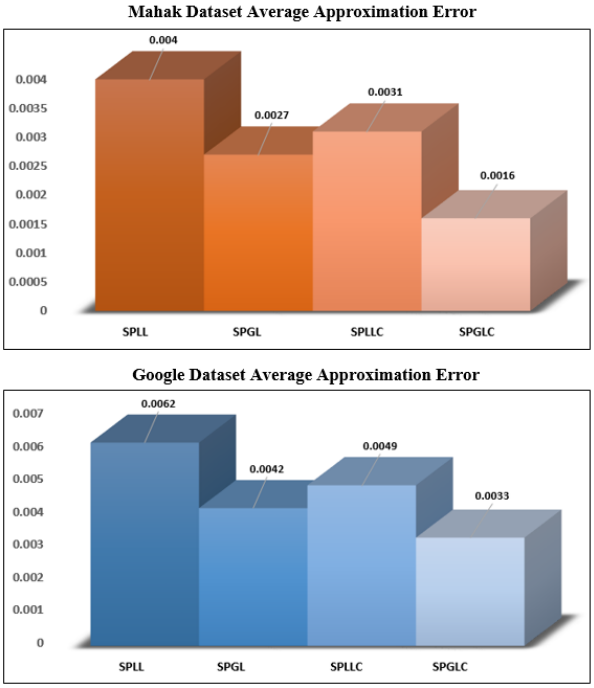
جدول ۲: خلاصه‌ای از مدل‌های پیشنهادی

راهکار	زمان اجرای برون-خط (دقیقه)	متوسط زمان‌اجرای برخط (۱۰۰۰۰ جفت گره) (بر حسب میلی ثانیه)		متوسط خطای تخمینی		پوشش کل گراف
		گوگل	محک	گوگل	محک	
SPLL	۱۲۰	۹.۰۱۵	۹.۰۲۱	۰.۰۰۶۲	۰.۰۰۴۰	عدم پوشش
SPGL	۱۴۰	۹.۰۹۲	۹.۱۰۱	۰.۰۰۴۲	۰.۰۰۲۷	عدم پوشش
SPLLC	۱۳۰	۹.۰۰۳	۹.۰۱۰	۰.۰۰۴۹	۰.۰۰۳۱	پوشش کل گراف
SPGLC	۱۴۵	۹.۰۸۵	۹.۰۸۸	۰.۰۰۳۳	۰.۰۰۱۶	پوشش کل گراف

در الگوریتم دوم یعنی SPGL ، از تزریق مراکز سایر خوشه‌ها به هر خوشه به منظور افزایش دقت در دسترسی به تعداد مسیرهای بیشتر و امکان انتخاب مسیرهای بهینه‌تر استفاده شده است. این ساختار می‌تواند منجر به افزایش زمان پیش‌پردازش در محاسبه کوتاهترین مسیرها بین گره‌های درون هر خوشه و مراکز سایر خوشه‌ها گردد اما با مجموع زمان اجرای ۹/۰۹۲و ۹/۱۰۱ به ازای دو مجموعه دادگان گوگل و محک، به خطای تخمینی ۰/۰۰۴۲ و ۰/۰۰۲۷ می-رسد که نسبت به الگوریتم اول، پیشرفت خوبی محسوب می‌شود و این به دلیل یافتن مسیرهای بهتر با کمک گره‌های لندمارک سایر خوشه‌ها در هر خوشه است. در الگوریتم سوم یعنی SPLLC با هدف دستیابی به پوشش بهتری از هر دو گراف وب، از الگوریتم ابتکاری جدیدی به منظور انتخاب گره‌های لندمارک استفاده شد. دلیل استفاده از الگوریتم جدید ابتکاری در انتخاب گره‌های لندمارک به شکلی متفاوت از روش‌های رایج، دستیابی به پوشش بهتر برای یکسری از گره‌هاست که به دلیل ارتباطات کمتر با سایر گره‌ها، اغلب در دسترس سایرین نیستند ولی تا به حال در هیچ مقاله یا الگوریتمی، به آن پرداخته نشده و اغلب معیار ارزیابی بر اساس جفت گره‌های انتخاب شده از قبل به صورت نمونه و حذف این نوع گره‌ها، صورت می‌پذیرد و این مسئله نادیده گرفته می‌شود. اما در SPLLC و SPGLC این موضوع بررسی و تا حد امکان مرتفع گردید. در واقع ساختار اصلی این دو الگوریتم مشابه دو الگوریتم SPLL وSPGL است با این تفاوت که به جای انتخاب گره‌های لندمارک از شیوه‌های مرسوم، از یک الگوریتم ابتکاری جدید با هدف پوشش حداکثری تمامی گره‌های گراف استفاده شده است که در بخش ۴-۴-۱ توضیح کاملی از آن ارائه گردید. الگوریتم SPLLC به مجموع زمان اجرای برخط ۹/۰۰۳ و ۹/۰۱۰ و متوسط خطای تخمینی ۰/۰۰۴۹ و ۰/۰۰۳۱ به ازای دو مجموعه دادگان گوگل و محک می‌رسد که از نظر خطای تخمینی بهتر از شیوه مشابه در الگوریتم SPLL است. همچنین مجموع زمان اجرای الگوریتم SPGLC برابر با۹/۰۸۵ و ۹/۰۸۸ است در حالی که به خطای تخمینی ۰/۰۰۳۳ و ۰/۰۰۱۶ دست می‌یابد. همانگونه که مشاهده می‌شود الگوریتم SPGLC بهترین نتایج را از نظر خطای تخمینی نسبت به سایر الگوریتم‌های پیشنهادی ارائه داده است که به دلیل پوشش بهتر گراف و همچنین افزودن مراکز سایر خوشه‌ها به هر خوشه است.

نکته قابل توجه در نتایج بدست آمده از الگوریتم‌های پیشنهادی، کاهش خطای تخمینی مجموعه دادگان فارسی محک نسبت به گوگل است که می‌تواند به دلیل ارتباطات بیشتر گره‌ها نسبت به یکدیگر در مجموعه دادگان فارسی محک باشد. در واقع توجه به این نکته ضروری است که درجه گره‌ها در وب فارسی مطابق جدول ۱ تقریبا هفت برابر وب انگلیسی است لذا می‌توان به این نتیجه رسید که الگوریتم‌های پیشنهادی ما، علاوه بر ارائه نتایج بسیار خوب در وب انگلیسی، بر روی گراف وب فارسی به نتایج ارزشمند و بسیار دقیقی دست می‌یابند که قابل توجه است ضمن اینکه در خصوص الگوریتم‌های مورد مقایسه در بخش مرور ادبیات و همچنین قسمت ۶-۳ یعنی TreeSketch و LS، به عملکرد بهتر این الگوریتم‌ها بر روی گراف‌های با درجه کم مانند شبکه‌های اجتماعی و مانند آن اشاره شده و می‌تواند این مساله، کارایی آن‌ها را به نوعی تحت الشعاع قرار دهد.

شکل ۸ متوسط خطای تخمینی بین چهار الگوریتم ارائه شده را نشان می‌دهد که با توجه به نتایج بدست آمده، خطای تخمینی SPGL و SPGLC از سایر الگوریتم‌های پیشنهادی کمتر است که به دلیل افزودن مجموعه لندمارک‌های بدست آمده از کل گراف درون هر خوشه است.



شکل ۸ : متوسط خطای تخمینی مدل‌های پیشنهادی

۶-۳- مقایسه الگوریتم‌های پیشنهادی با سایر روش‌ها

در جدول ۳ الگوریتم‌های پیشنهادی با دو الگوریتم موفق ذکر شده در بخش مرور ادبیات بر مبنای دیتاست گوگل مقایسه گردید.

در الگوریتم TreeSketch تمامی مسیرهای موجود بین هر گره و لندمارک‌ها در فاز پیش‌پردازش، محاسبه و تحت عنوان sketch هر گره ذخیره می‌شود، سپس با دریافت پرس‌وجوی حاوی گره مبدا و مقصد در زمان اجرای برخط، با جستجوی sketch گره مبدا و مقصد، تمامی مسیرهایی که گره مبدا با استفاده از گره لندمارک به گره مقصد متصل می‌شود را بدست آورده و با شناسایی مسیرهای میان‌بر در زمان اجرای برخط، نتایج قابل قبولی ارائه می‌دهد.

بعد از الگوریتم TreeSketch الگوریتم دیگری ارائه شد که از ایده الگوریتم TreeSketch استفاده کرد و توانست نتایج آن را بهبود بخشد. این الگوریتم با سه مدل ارائه گردید که هریک بهبود یافته مدل قبلی محسوب می‌شد و در نهایت الگوریتم LS به عنوان الگوریتم برتر برمبنای لندمارک محلی و وابسته به پرس‌وجوی کاربر معرفی شد. در این الگوریتم با انتخاب لندمارک‌های محلی برای هر گره بعد از دریافت پرس‌وجوی کاربر در فاز برخط، کوتاهترین مسیر بین دو گره مبدا و مقصد تعیین می‌گردد. در واقع تعدادی لندمارک مبتنی بر درجه گره‌ها در فاز پیش‌پردازش انتخاب می‌شود و سپس در فاز برخط در صورتی که در مسیر رسیدن از گره مبدا به مقصد به واسطه یک لندمارک، گره‌ای را بیابد که در هر دو مسیر مبدا و مقصد نسبت به لندمارک، مشترک باشد، آن گره را به عنوان لندمارک محلی در نظر می‌گیرد و از این پس، فاصله بین دو گره از طریق آن محاسبه می‌گردد.

جدول ۳: مقایسه الگوریتم‌های پیشنهادی با سایر الگوریتم‌ها

دیتاست گوگل		
الگوریتم ها	متوسط زمان اجرای برخط (۱۰۰۰۰ جفت گره) (بر حسب میلی ثانیه)	متوسط خطای تخمینی
SPGL	۹.۰۹۲	۰.۰۰۴۲
SPGLC	۹.۰۸۵	۰.۰۰۳۳
TreeSketch	۳.۵۴۹	۰.۰۰۴۸
LS	۲.۷۲۹	۰.۰۰۴۶

با توجه به جدول ۳، مجموع زمان اجرای دو الگوریتم TreeSketch و LS به ترتیب برابر با ۳/۵۴۹ و ۲/۷۲۹ میلی‌ثانیه برای ۱۰ هزار جفت گره‌ی پرس‌وجوست در صورتی که دو الگوریتم پیشنهادی ما یعنی SPGL وSPGLC به متوسط زمان اجرای ۹/۰۹۲ و ۹/۰۸۵ میلی‌ثانیه می‌رسد که به نتایج مورد نظر نزدیک است با این تفاوت که متوسط خطای تخمینی دو الگوریتم TreeSketch و LS به ترتیب برابر ۰/۰۰۴۸ و ۰/۰۰۴۶ است در صورتی که خطای تخمینی دو الگوریتم ما به ۰/۰۰۴۲ و ۰/۰۰۳۳ می‌رسد که بیانگر دقت بالای الگوریتم‌های پیشنهادی، نسبت به الگوریتم‌های فوق است. ضمن اینکه به عملکرد بهتر دو الگوریتم TreeSketch و LS، بر روی گراف-های با درجه کم مانند شبکه‌های اجتماعی و مانند آن در مقاله‌های مذکور اشاره شده است. اما با در نظر گرفتن اطلاعات گراف‌ها در جدول یک، متوسط درجه گره‌ها در گراف وب گوگل بین ۶ و ۷ و در وب فارسی محک برابر ۴۷ است که تقریبا ۷ برابر بیشتر است لذا کارایی این دو الگوریتم در گراف وب فارسی محک با این حجم از ارتباطات بین گره‌ها، قطعا کاهش می‌یابد و تحت تاثیر روابط بین گره‌ها قرار خواهد گرفت، در صورتی که در الگوریتم‌های پیشنهادی ما، هر چه ارتباطات بین گره‌ها بیشتر باشد منجر به افزایش کارایی و نتایج دقیق‌تر می‌شود که نتایج آن بر روی مجموعه دادگان فارسی محک در جدول ۲ قابل مشاهده است [۷] و [۹] .

۷- نتیجه‌گیری و کارهای آینده

در این پژوهش روش‌های مختلف یافتن کوتاهترین مسیر تخمینی بین هر دو گره تصادفی مورد بررسی قرار گرفت، سپس چهار راهکار ابتکاری مبتنی بر لندمارک ارائه گردید که با در نظر گرفتن شیوه متفاوت در انتخاب گره لندمارک در هر مدل و خوشه‌بندی گره‌ها، نتایج جالب و قابل توجهی بدست می‌آید که می‌تواند در صورت نیاز، به پوشش کاملی از گراف نیز منجر شود. در واقع در این پژوهش با استفاده از یک تکنیک خوشه‌بندی به صورت ابتکاری و افزودن مراکز سایر خوشه‌ها به هر خوشه، مدل ابتکاری جدیدی معرفی می‌گردد که دقت و پوشش کامل گراف را تحت تاثیر قرار می‌دهد و می‌تواند در کاربردهای مختلف بسیار مفید و کاربردی باشد.

به عنوان کارهای آینده می‌توان از روش‌های مختلفی به منظور بهبود مدل‌های فوق استفاده نمود به عنوان مثال با هرس‌کردن فضای جستجو، می‌توان با توجه به طول مسیرهای بدست آمده و اعمال محدودیت روی آن، مسیرهای طولانی را نادیده گرفت همچنین استفاده از سایر روش‌های خوشه‌بندی در کنار الگوریتم ابتکاری ما، می‌تواند تاثیر بیشتری در ارتباط اسناد، کاهش خطا و سرعت رسیدن به نتایج داشته باشد.

مراجع

[۱] درودی، ا.، برادران هاشمی، ه.، آل احمد، ا.، زارع بیدکی، ع. م.، حبیبیان، ا. ح.، مهدیخانی، ف.، شاکری، آ.، و رهگذر، م. (۱۳۸۷). مجموعه محک استاندارد برای تحقیقات بازیابی اطلاعات وب فارسی. (شماره گزارش: ۱۳۸۷۰۲DBRG-TR-). گروه تحقیقاتی پایگاه داده: دانشگاه تهران.

[2] Madkour, A., Aref, W. G., Rehman, F. U., Rahman, M. A., & Basalamah, S. (2017). A survey of shortest-path algorithms. *arXiv preprint arXiv:1705.02044*.

[3] Potamias, M., Bonchi, F., Castillo, C., & Gionis, A. (2009, November). Fast shortest path distance estimation in large networks. In Proceedings of the 18 th ACM conference on Information and knowledge management (pp. 867-876). ACM.

[4] Goldberg, A. V., & Harrelson, C. (2005, January). Computing the shortest path: A search meets graph theory. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms* (pp.156-165). Society for Industrial and Applied Mathematics.

[5] Goldberg, A. V. (2007, January). Point-to-point shortest path algorithms with preprocessing. In *International Conference on Current Trends in Theory and Practice of Computer Science* (pp. 88-102). Springer, Berlin, Heidelberg.

[6] Grant, K., & Mould, D. (2008, July). LPI: Approximating shortest paths using landmarks. In *Workshop on Artificial Intelligence in Games* (p. 45).

[7] Gubichev, A., Bedathur, S., Seufert, S., & Weikum, G. (2010). Fast and accurate estimation of shortest paths in large networks. In *19th ACM Conference on Information and Knowledge Management* (pp. 499-508). ACM.

[8] Cao, L., Zhao, X., Zheng, H., & Zhao, B. Y. (2011). Atlas: Approximating shortest paths in social graphs. *Computer Science Department*, U. C. Santa Barbara.

[9] Qiao, M., Cheng, H., Chang, L., & Yu, J. X. (2012). Approximate shortest distance computing: A query-dependent local landmark scheme. *IEEE Transactions on Knowledge and Data Engineering*, 26(1), 55-68.

[10] Floreskul, V., Tretyakov, K., & Dumas, M. (2014, May). Memory-efficient fast shortest path estimation in large social networks. In *Eighth International AAAI Conference on Weblogs and Social Media*.

[11] Feng, C., & Deng, T. (2018, October). More Accurate Estimation of Shortest Paths in Social Networks. In ۲۰۱۸ *International Symposium on Distributed Computing and Applications for Business Engineering and Science* (DCABES) (pp. 314-317). IEEE.

[12] Dong, Q., Lakhotia, K., Zeng, H., Karman, R., Prasanna, V., & Seetharaman, G. (2018, September). A Fast and Efficient Parallel Algorithm for Pruned Landmark Labeling. In 2018 *IEEE High Performance extreme Computing Conference (HPEC)*(pp.1-7) . IEEE.

[13] J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney. (2009). Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. Internet Mathematics 6(1) 29--123.

زیرنویس‌ها:

[1] Breadth First Search
[2] Dijkstra
[3] Prune
[4] Landmark Pathfinding between Intersections
[5] Spanning trees
[6] Global Landmark Scheme
[7] Local Landmark Scheme
[8] Local Search
[9] Local Subgraph Query
[10] Pruned Landmark Labeling
[11] Hub
[12] Shortest-Path Trees
[13] Offline
[14] Preprocess
[15] Shortest Path Local Landmark
[16] Shortest Path Global Landmark
[17] Subset Node
[18] Seen
[19] Shortest Path Local Landmark Full Coverage
[20] Shortest Path Global Landmark Full Coverage
[21] Power Law