دوفصلنامه محاسبات و سامانه‌های توزیع شده
www.ijdcs.ir

ارائه‌ی روش جدید یافتن هسته جوامع برای بهبود الگوریتم انتشار برچسب در شبکه‌های اجتماعی

حمید روغنی*[1]، عسگرعلی بویر[1]

[1] دانشگاه شهید مدنی آذربایجان، تبریز، ایران

چکیده

کشف و تحلیل اطلاعات پنهان در شبکه‌های پیچیده، مخصوصا شبکه‌های اجتماعی توجه بسیاری از محققان را به خود جلب کرده است. تشخیص جوامع ابزاری بسیار مهم برای پی بردن به اطلاعات شبکه و ساختار آن می‌باشد. روش‌های بسیاری در این زمینه ارائه شده است اما همچنان ارائه‌ی روش‌های جدید با دقت و کیفیت بالا یکی از مباحث جذاب و چالش برانگیز در زمینه‌ی تشخیص جوامع است. الگوریتم انتشار برچسب (LPA)، یکی از الگوریتم‌های مشهور در زمینه‌ی تشخیص جوامع است. وجود ماهیت تصادفی و عدم ثبات در نتایج این الگوریتم، سبب شده است که LPA الگوریتم غیر قابل اطمینانی باشد. تشخیص درست گره‌های هسته و اتخاذ یک ترتیب مشخص از گره‌ها بر اساس اهمیتشان، پخش صحیح‌تر برچسب گره‌های هسته و بهبود نحوه‌ی انتخاب برچسب صحیح مباحثی هستند که در این مقاله به بررسی آنها می‌پردازیم. در این مقاله الگوریتم پیشنهادی که ترکیبی از روش پخش برچسب گره‌های هسته و روش انتشار برچسب است با انتخاب گره‌های هسته و پخش برچسب آنها و بهبود نحوه‌ی بروزرسانی برچسب گره‌ها، توانسته است دقت و کارایی الگوریتم LPA را بهبود بخشد. نتایج آزمایش‌های انجام شده بر روی دیتاست‌های استاندارد و مصنوعی و مقایسه نتایج حاصل، بیانگر عملکرد بهتر الگوریتم پیشنهادی نسبت به الگوریتم LPA و NIB-LPA است.

کلمات کلیدی: تشخیص جوامع، پخش دو سطحی برچسب، گره هسته، تاثیر برچسب، شبکه‌های اجتماعی

Email: **h.roghani@azaruniv.ac.ir**

A New Core Discovering Method for Improving Label Propagation Algorithm in Social Networks

Hamid Roghani[*1], Asgarali Bouyer[1]

[1]Azarbaijan Shahid Madani University, Tabriz, Iran

تاریخچه مقاله:

تاریخ ارسال: ۹۸/۰۹/۲۰

تاریخ اصلاحات: ۹۸/۱۱/۱۶

تاریخ پذیرش: ۹۸/۱۲/۰۲

تاریخ انتشار: ۹۸/۱۲/۲۰

Keywords:

Community detection
2-Level Label diffusion
Core node
Label influence
Social networks

Abstract

Discovering and analyzing of hidden information in complex networks and especially social networks has attracted the attention of many researchers. Community detection is a very important tool for understanding network information and its structure. Many approaches have been proposed, but proposing new methods with high accuracy and high quality are still one of the most exciting and challenging topics in community detection field. Label Propagation Algorithm (LPA) is one of the most popular algorithms in the field of community detection. The random nature and instability in the results of this algorithm make the LPA unreliable. Determining the core nodes and adopting a specific order of nodes based on their importance, more accurate diffusion of the core node label, and improving the way of selecting the correct label, are the main topics that we will discuss in this paper. In this paper, the proposed method which is combination of label diffusion of core nodes and label propagation method, has been able to improve the accuracy and efficiency of LPA by selecting and diffusing the core nodes label and improving the label update strategy. The results of experiments performed on standard and synthetic datasets and comparing the results show that the proposed algorithm performs better than the LPA and NIB-LPA algorithms.

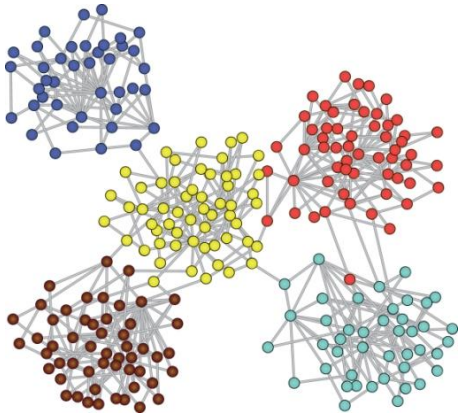
روش ارجاع به مقاله: ارائه‌ی روش جدید یافتن هسته جوامع برای بهبود الگوریتم انتشار برچسب در شبکه‌های اجتماعی، ح.روغنی، ع.علی بویر، دو فصلنامه محاسبات و سامانه‌های توزیع شده، سال دوم، شماره دوم، شماره پیاپی ۴، سال ۱۳۹۸، ص ۲۳-۵۵

۱ – مقدمه

امروزه با افزایش ابزارهای تولید داده و سهولت دست-یابی به نرم افزارهای شبکه‌های اجتماعی، داده‌های بسیاری در حال تولید هستند. وجود ساختار انجمنی در شبکه‌های پیچیده، یکی از ویژگی‌های بارز آنها است. کشف این ساختار پایه و اساس دسترسی به اطلاعات ارزشمند نهفته در شبکه‌های پیچیده[1] است، به عنوان مثال تشخیص جوامع[2] در شبکه‌های اجتماعی، اساسی-ترین ابزار برای تحلیل روابط بین افراد متفاوت است. شبکه‌های اطلاعاتی، شبکه‌های استناد به مقالات و شبکه‌های اجتماعی نمونه‌هایی از شبکه‌های پیچیده هستند. شبکه‌های پیچیده به دلیل ویژگی‌هایی که دارند، متفاوت از گراف‌های تصادفی هستند. این نوع شبکه‌ها ویژگی‌هایی مانند ضریب خوشه‌بندی بالا، پدیده‌ی جهان کوچک، توزیع درجات به صورت قانون توانی[3]، ساختار انجمنی و ... دارند که آنها را از سایر شبکه‌های تصادفی متمایز می‌سازد. یکی از ویژگی‌های مهم شبکه‌های پیچیده، وجود جوامع در آنها است. با استفاده از روش‌های تشخیص جوامع می‌توان ساختار شبکه را کشف کرد. یک جامعه، زیرگرافی از گراف اصلی است که تعداد اتصالات بین گره‌های داخل آن جامعه زیاد بوده و اتصالات بین گره‌های داخل یک جامعه با گره‌های سایر جوامع بسیار کم باشد. شکل (۱) نمونه‌ای از یک شبکه با ۵ جامعه را نشان می‌دهد. الگوریتم LPA یکی از الگوریتم‌های تشخیص جوامع است که به دلیل سادگی الگوریتم و پیچیدگی زمانی نزدیک به خطی، توجه بسیاری به خود جلب کرده است [۱]. این الگوریتم به دلیل رفتار تصادفی در انتخاب ترتیب گره‌ها و نحوه‌ی انتخاب و به‌روز رسانی برچسب گره‌ها زمانی که چندین برچسب فراوانی یکسانی دارند، سبب شده است نتایج غیر قابل اطمینانی ارائه دهد. مسئله‌ی مهم دیگری که باید به آن توجه داشت این است که گره‌های موجود در یک انجمن شباهت‌هایی باهم دارند. تشابه بین دو گره، معیاری است که میزان رابطه و یا نزدیکی بین دو گره را اندازه‌گیری می‌کند [۲]. تشابه سراسری اطلاعات جامع و کلی‌تری از شبکه را ارائه می-کند اما از طرفی محاسبه آن زمانبر است. در مقابل، تشابه محلی، اطلاعات محلی را در زمان کمتری محاسبه می‌کند و می‌تواند نتایج قابل قبولی ارائه دهد [۳،۴]. الگوریتم LPA به هیچ وجه اهمیت گره‌ها و تشابه بین آنها را در نظر نمی‌گیرد و تنها معیار آن برای قرار دادن گره‌ها در یک جامعه، انتخاب فراوان‌ترین برچسب است. همچنین منطقی است پس از مشخص شدن گره‌های هسته، آن گره‌ها برچسب خود را به گره‌های اطراف خود پخش کنند که این عمل در الگوریتم LPA وجود ندارد. راه‌حل مشکل ماهیت تصادفی الگوریتم LPA، انتخاب یک ترتیب مشخص و ثابت برای گره‌ها است، به طوری که گره‌هایی که اهمیت بیشتری دارند و گره هسته[4] محسوب می‌شوند، زودتر از بقیه گره‌ها برچسب بگیرند و برچسب خود را به همسایه‌های مستقیم و حتی همسایه‌های غیر مستقیم خود پخش کنند تا گره‌های کم اهمیت‌تر برچسب گره با اهمیت را دریافت کنند. همچنین با تغییر نحوه‌ی انتخاب برچسب و توجه به تشابه گره‌ها و تاثیر

[1] Complex network
[2] Community detection
[3] Power law
[4] Core node

متفاوت برچسب جوامع مختلف،



(شکل-۱): نمونه‌ای از یک شبکه با ۵ جامعه

می‌توان موثرترین برچسب جامعه را انتخاب کرد. الگوریتم NIB-LPA [۵] یکی از الگوریتم‌های مبتنی بر انتشار برچسب است که از تجزیه K-shell برای تعیین اهمیت گره‌ها استفاده می‌کند. این الگوریتم با انتخاب یک ترتیب ثابت از گره‌ها بر اساس مقدار اهمیت به دست آمده از تجزیه K-shell و مجموع اهمیت و تاثیر برچسب‌ها، الگوریتم LPA را بهبود داده است و نتایج پایدارتری نسبت به LPA دارد. روش پیشنهادی ارائه شده در این مقاله روشی است که هم از مزایای پخش برچسب گره هسته و هم از مزایای روش انتشار برچسب بهره گرفته است و الگوریتم LPA و NIB-LPA را بهبود داده است.

در بخش دوم برخی از روش‌های ارائه شده در زمینه‌ی تشخیص جوامع بیان می‌شود. در بخش سوم، الگوریتم-های LPA، NIB-LPA و روش پیشنهادی توضیح داده می‌شوند، سپس در بخش چهارم نتایج حاصل از آزمایش الگوریتم‌ها بر روی دیتاست‌های مختلف مورد بررسی و ارزیابی قرار می‌گیرد و در بخش پنجم نتیجه‌گیری کلی از روش پیشنهادی بیان می‌شود.

۲ – کارهای انجام شده‌ی پیشین

از اولین روش‌های تشخیص جوامع، روش‌های بخش-بندی گراف است. این نوع روش‌ها گراف را به g قسمت از پیش تعیین شده تقسیم می‌کند به طوری که تعداد یال‌های بین جوامع کمینه باشد و از یک تابع برای ارزیابی کیفیت تقسیم استفاده می‌شود تا یال‌های بین جوامع کمینه باشد [۳, ۶, ۷].

روش‌های مبتنی بر ماژولاریتی از دیگر روش‌های تشخیص جوامع هستند که از معیاری به نام ماژولاریتی برای سنجش کیفیت جوامع و ادغام گره‌ها استفاده می کند. در مقاله [۸] روشی مبتنی بر ماژولاریتی و الگوریتم سلسله مراتبی تجمعی ارائه شده است. در این روش ابتدا هر گره به عنوان یک جامعه مستقل است و به صورت تکراری گره‌هایی که ادغام آنها مقدار تابع ماژولاریتی را افزایش دهد باهم ادغام می‌شوند.

در مقاله [۹] روشی اکتشافی مبتنی بر ماژولاریتی ارائه شده است. در این روش، هر گره به عنوان یک جامعه مستقل در نظر گرفته می‌شود. سپس طی یک فرایند تکراری گره‌هایی که ادغام آنها معیار ماژولاریتی را افزایش دهد، باهم ادغام می‌شوند. ادغام گره‌ها تا زمانی که هیچ بهبودی حاصل نشود ادامه می‌یابد. پس از تشخیص جوامع اولیه، جوامعی که ادغام آنها سبب افزایش ماژولاریتی می‌شود باهم ادغام می‌شوند. این روش به الگوریتم Louvain معروف است.

روش‌های سلسله مراتبی دسته‌ای دیگر از روش‌های تشخیص جوامع هستند. در این نوع روش‌ها با استفاده از یک معیار تشابه، به صورت تکراری گره‌هایی که تشابه

بیشتری نسبت به هم دارند باهم ادغام می‌شوند و یا سعی می‌شود با استفاده از معیاری یال‌هایی از شبکه حذف شود تا جوامع نهایی به دست آیند.

نیومن الگوریتمی تقسیمی مبتنی بر بینیت[5] در مقاله [۱۰] ارائه داده است. در این روش با استفاده از الگوریتم سلسله مراتبی تقسیمی و معیار بینیت جوامع موجود کشف می‌شوند.

مقاله [۱۱] ، از الگوریتم سلسله مراتبی تقسیمی برای حذف یال‌ها با بیشترین مقدار مرکزیت اطلاعات استفاده می‌کند.

در مقاله [۱۲] از الگوریتم Louvain و از یک معیار تشابه برای سنجش میزان تشابه دو گره استفاده شده است. در این الگوریتم ابتدا تشابه بین گره‌های متصل به هم سنجیده می‌شود. هر گره به عنوان یک جامعه مستقل بوده و گره به جامعه‌ای که انتقال آن سبب افزایش مقدار ماژولاریتی می‌شود، انتقال می‌یابد. پس از تشکیل جوامع اولیه، جوامعی که ادغام آن‌ها ماژولاریتی را افزایش دهد، باهم ادغام می‌شوند.

روش‌های مبتنی بر انتشار برچسب[6] معمولا به دلیل پایین بودن پیچیدگی زمانی الگوریتم و همچنین سهولت انجام برخی بهبودها برای افزایش دقت و کیفیت الگوریتم، دارای محبوبیت بیشتری هستند.

مقاله [۱۳] با استفاده از اهمیت گره‌ها و تاثیر متفاوت برچسب جوامع، الگوریتم LPA را بهبود داده است. این روش که به الگوریتم LPA-NI معروف است، با استفاده از شبکه‌ی بیزین، اهمیت گره‌ها را به دست می‌آورد و گره‌ها به ترتیب اهمیتشان انتخاب می‌شوند، به این ترتیب مشکل انتخاب تصادفی گره در الگوریتم LPA بهبود می‌یابد. همچنین این الگوریتم با انتخاب موثرترین برچسب زمانی که چندین برچسب با فراوانی یکسان وجود داشته باشد، مشکل انتخاب تصادفی برچسب در الگوریتم LPA را بهبود داده است. الگوریتم LPA-NI بسیار شبیه الگوریتم NIB-LPA [۵] است با این تفاوت که در الگوریتم NIB-LPA از مفهوم K-shell برای تعیین اهمیت گره‌ها استفاده شده است.

مقاله [۱۴] با در نظر گرفتن تاثیر متفاوت گره‌ها بر روی هم یک ترتیب ثابت و معنی‌دار از گره‌ها انتخاب می‌کند. گره‌ها بر اساس تاثیرشان انتخاب می‌شوند تا گره‌های با اهمیت بیشتر، در ابتدا برچسبشان تعیین شود. در این روش برای به‌روز رسانی برچسب گره‌ها، ابتدا برچسب با بیشترین مجموع تاثیر انتخاب می‌شود و در صورتی که چندین برچسب با تاثیر برابر وجود داشته باشد، از معیار دیگری که در مقاله توضیح داده شده است استفاده می‌کند. این روش به الگوریتم LPA-Intimacy معروف است که یکی از الگوریتم‌های ارائه شده با دقت و پایداری نسبتا مناسب برای بهبود الگوریتم LPA است.

مقاله [۱۵] روشی مبتنی بر انتشار برچسب با هسته جوامع[7] است. هر گره با توجه به اهمیتی که در شبکه دارد یک وزن دریافت می‌کند و گره‌ها بر اساس وزنی که دارند انتخاب شده و برچسبشان را به‌روز رسانی می‌کنند. گره‌هایی که به مرکز جامعه نزدیک هستند وزن بیشتر و گره‌هایی که از مرکز جامعه دورتر هستند، وزن کمتری

[5] Betweenness
[6] Label propagation
[7] Community kernel

خواهند داشت. اولین گام، کشف هسته جوامع است. به ازای هر هسته اولیه‌ی کشف شده، گرهی با بیشترین اتصال به آن انتخاب شده و به جامعه‌ی آن افزوده می‌شود. در به‌روز رسانی و انتخاب برچسب جدید، مجموع وزن‌های گره‌های متصل به گره انتخاب شده، که در یک جامعه هستند محاسبه می‌شود، و هر جامعه‌ای که مقدار بیشتری داشته باشد برچسب آن انتخاب می‌شود.

مقاله [۱۶] با استفاده از معیار درجه‌ی عضویت گره‌ها، تعداد مراکز جوامع و ساختار اصلی مشخص می‌شود و گره‌های همسایه‌ای که مقدار درجه‌ی عضویت آنها با جامعه اطراف خود بیشتر از حد آستانه باشد به آن جامعه افزوده می‌شوند. سپس با استفاده از الگوریتم LPA، گره-هایی که به جامعه‌ای تخصیص داده نشده‌اند، برچسب جدید می‌گیرند.

با در نظر گرفتن نقاط ضعف الگوریتم LPA، الگوریتم NIB-LPA و الگوریتم پیشنهادی برای بهبود نقاط ضعف آن مورد بررسی و آزمایش قرار گرفته‌اند. هر یک از دو الگوریتم ذکر شده با اتخاذ یک ترتیب مشخص و بهبود نحوه‌ی انتخاب و به‌روز رسانی برچسب گره‌ها، نقاط ضعف الگوریتم LPA را بهبود داده‌اند.

۳ – روش پیشنهادی

در این بخش در قسمت ۳-۱ الگوریتم انتشار برچسب(LPA) و در بخش ۳-۲ الگوریتم NIB-LPA که بهبودی بر الگوریتم LPA است بیان می‌شود. جزئیات الگوریتم پیشنهادی نیز در بخش ۳-۳ به طور کامل توضیح داده می‌شود.

۳-۱ الگوریتم LPA

الگوریتم انتشار برچسب یا همان LPA (Label Propagation Algorithm)، اولین الگوریتم مبتنی بر انتشار برچسب است که در سال ۲۰۰۷ توسط راگاون ارائه شد [۱, ۳]. این الگوریتم به دلیل پیچیدگی زمانی نزدیک به خطی و همچنین سادگی الگوریتم و سهولت فهم آن، جز الگوریتم‌های مورد توجه است اما به دلیل وجود ماهیت تصادفی در آن، نتایج ثابتی ارائه نمی‌دهد. مراحل این الگوریتم به صورت زیر است:

۱) در ابتدای الگوریتم تمامی گره‌ها برچسب منحصر به فردی دریافت می‌کنند.
۲) هر بار یک گره به صورت تصادفی از بین گره‌های شبکه انتخاب می‌شود.
۳) گره انتخاب شده برچسب جامعه‌ای که بیشترین فراوانی را دارد به عنوان برچسب جدید خود انتخاب می‌کند. اگر چندین برچسب با بیشترین فراوانی وجود داشته باشد، گره مورد نظر یکی را به صورت تصادفی انتخاب می‌کند.
۴) مراحل ۲ و ۳ برای تمامی گره‌ها تکرار خواهد شد تا شرط پایان الگوریتم ایجاد شود.
۵) شرط پایان الگوریتم به این صورت است که یا مقداری به عنوان تکرار برای الگوریتم در نظر گرفته می‌شود و با پایان آن الگوریتم خاتمه می-یابد و یا زمانی که برچسب گره‌ها تغییری نکند، الگوریتم به پایان می‌رسد.
۶) پس از اتمام الگوریتم گره‌هایی که برچسب یکسان دارند در یک جامعه قرار می‌گیرند.

همانطور که از الگوریتم مشخص است این الگوریتم برای انتخاب ترتیب گره‌ها و همچنین انتخاب برچسب، یک رویکرد تصادفی دارد که باعث می‌شود الگوریتم نتایج ثابت و پایداری در هر اجرا ارائه ندهد، لذا به صورت کامل

قابل اطمینان نیست. باید در نظر داشت که در یک شبکه گره‌ها اهمیت متفاوتی دارند و گره‌هایی که در هسته شبکه قرار دارند اهمیت بیشتری نسبت به گره‌های حاشیه‌ای دارند. با توجه به این نکته، برای کسب نتایج دقیق‌تر لازم است که گره‌هایی که در هسته‌ی شبکه قرار دارند و با بقیه‌ی گره‌ها اتصالات بیشتری دارند، ابتدا برچسب دریافت کنند، چون این گره‌ها توانایی بیشتری در انتشار و پخش برچسب خود به سایر گره‌ها دارند. از طرفی برای کسب نتایج دقیق‌تر باید تا حد ممکن از انتخاب تصادفی خودداری کرد تا مناسب‌ترین برچسب انتخاب شود.

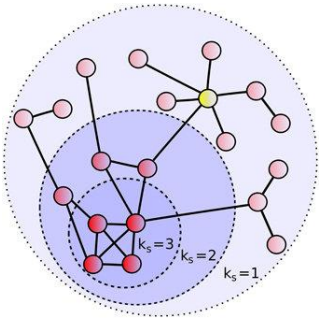
۳-۲ الگوریتم NIB-LPA

الگوریتم NIB-LPA [۵] یکی دیگر از الگوریتم‌های مبتنی بر انتشار برچسب است که از مفهوم تجزیه K-shell و مرتب‌سازی گره‌ها بر اساس مقدار اهمیت به دست آمده برای هر گره استفاده می‌کند. این الگوریتم با استفاده از مجموع اهمیت برچسب جوامع مختلف، نقاط ضعف الگوریتم LPA در انتخاب گره و به‌روز رسانی برچسب را بهبود داده است. طبق ایده‌ی مطرح شده در الگوریتم NIB-LPA، در شبکه‌های اجتماعی گره‌هایی که در هسته شبکه هستند اهمیت بیشتری نسبت به سایر گره‌ها، مخصوصا گره‌های حاشیه‌ای دارند و این گره‌های با اهمیت توانایی بیشتری در تعیین برچسب کلی جوامع موجود دارند، به همین خاطر معیار اهمیت گره را، بالا بودن مقدار K-shell به دست آمده می‌داند. یک K-shell یک زیرگراف همبند ماکزیمال است که در آن درجه هر گره حداقل k است[۵]. نمونه‌ای از یک گراف با ۳ پوسته، در شکل (۲) نشان داده شده است.

با استفاده از رابطه (۱) مقدار اهمیت برای هر گره مشخص می‌شود. سپس گره‌ها به صورت نزولی مرتب-سازی می‌شوند.

$$NI(i) = Ks(i) + \alpha \cdot \sum_{j \in N(i)} \frac{Ks(j)}{d(j)}$$

(فرمول-۱)مقدار اهمیت برای هر گره



(شکل-۲): عملکرد K-shell روی یک شبکه ۳ پوسته‌ای

در رابطه (۱)، $NI(i)$ مقدار اهمیت گره i، $Ks(i)$ مقدار K-shell به دست آمده برای گره i، α یک پارامتر قابل تنظیم بین ۰ و ۱ است که میزان تاثیر همسایه‌ها بر روی مقدار اهمیت گره i است. $Ks(j)$ مقدار K-shell برای گره j که همسایه‌ی گره i است و $d(j)$ هم درجه‌ی گره j است. $N(i)$ نیز مجموعه همسایه‌های گره i است. همچنین برای رفع ایراد انتخاب تصادفی برچسب در الگوریتم LPA زمانی که چندین برچسب با فراوانی یکسان وجود داشته باشد، الگوریتم NIB-LPA با استفاده از رابطه (۲) تاثیر هر برچسب را محاسبه می‌کند.

$$LI(l) = \sum_{j \in N^1(i)} \frac{NI(j)}{d(j)}$$

(فرمول-۲)تاثیر برچسب

در رابطه (۲)، $LI(l)$ نشان دهنده‌ی تاثیر هر برچسب، $NI(j)$ مقدار اهمیت گره j است که قبلا از رابطه (۱) محاسبه شده است و $d(j)$ نیز درجه‌ی گره j است. موثرترین برچسب جامعه از رابطه (۳) محاسبه می‌شود.

$$C_i = \arg_{l \in l_{max}} \max LI(l)$$

(فرمول-۳)

که در رابطه (۳) منظور از l_{max}، مجموعه‌ای از برچسب‌ها است که بیشترین فراوانی را در بین سایر برچسب‌ها دارند.

مراحل الگوریتم NIB-LPA به صورت زیر است:

۱) تخصیص برچسب منحصر به فرد به تمامی گره‌های شبکه.
۲) محاسبه مقدار اهمیت هر گره با استفاده از رابطه (۱) و مرتب‌سازی نزولی مقادیر.
۳) شروع الگوریتم و انتخاب گره‌ها از با اهمیت‌ترین گره به سمت کم اهمیت‌ترین گره و به‌روز رسانی برچسب گره‌ها. در هر مرحله از الگوریتم، یک گره با بیشترین مقدار اهمیت انتخاب می‌شود و برچسب آن باید به‌روز رسانی شود. در به‌روز رسانی برچسب یک گره، برچسبی که بیشترین فراوانی را داشته باشد، آن را به گره مورد نظر تخصیص می‌دهیم. اگر چندین برچسب فراوانی یکسانی داشته باشند، با استفاده از رابطه (۲) مقدار تاثیر هر برچسب محاسبه می‌شود و از رابطه (۳) موثرترین برچسب انتخاب می‌شود.
۴) اگر برچسب گره‌ها تغییری نکند، الگوریتم به پایان می‌رسد، در غیر این صورت دوباره مرحله‌ی به‌روز رسانی برچسب گره‌ها تکرار می‌شود.
۵) پس از پایان الگوریتم، گره‌هایی که برچسب یکسان دارند، در یک جامعه قرار می‌گیرند.

۳-۳- الگوریتم پیشنهادی

روش پیشنهادی در این مقاله ترکیبی از ایده‌ی پخش برچسب گره هسته و انتشار برچسب است که از مفاهیم تشخیص گره هسته و پخش برچسب گره هسته، تشابه محلی بین گره‌ها و تاثیر متفاوت برچسب جوامع استفاده می‌کند. روش پیشنهادی با بهره‌گیری از مفاهیم پخش برچسب گره هسته و انتشار برچسب، سعی می‌کند رویکردی متفاوت‌تر از سایر روش‌ها داشته باشد و جوامعی دقیق‌تر و نزدیک به جوامع واقعی پیدا کند. روش پیشنهادی با ترکیب دو معیار برای محاسبه‌ی اهمیت گره‌ها، ایجاد یک ترتیب مشخص و ثابت و پخش برچسب گره‌های هسته و همچنین تغییر نحوه‌ی به‌روز رسانی برچسب گره‌ها، از حالت تصادفی به حالت انتخاب بر اساس تاثیر برچسب، عملکرد و نتایج الگوریتم LPA را بهبود داده است. در الگوریتم LPA عملا پخش برچسبی وجود ندارد که گرهی پس از اینکه برچسبی دریافت کرد آن را پخش کند و فقط هر گره برچسب خود را به‌روز رسانی می‌کند و این کار سبب می‌شود که گره‌هایی که در هسته هستند نتوانند برچسب خود را به سایر گره‌های کم اهمیت پخش کنند و حتی امکان دارد یک گره هسته، برچسب یک گره حاشیه‌ای را دریافت کند. به همین دلیل منطقی است که پس از تعیین گره‌های با اهمیت، ابتدا آنها صاحب برچسب شوند و برچسب خود را پخش کنند

تا گره‌های همسایه که اهمیت کمتری دارند آن را دریافت کنند و چارچوب اصلی جامعه سریع‌تر مشخص شود و بتوان در مرحله‌ی انتخاب برچسب، تا حد امکان از شکست تساوی‌ها[8] زمانی که چندین برچسب فراوانی و یا تاثیر برابر دارند، خودداری کرد. از طرفی قسمت به‌روز رسانی برچسب گره‌ها در الگوریتم NIB-LPA، که در قسمت ۳-۲ ذکر شده است، شامل ضعفی در انتخاب برچسب موثر است. رابطه (۲) برای انتخاب برچسب موثر فقط اهمیت گره‌های اطراف یک گره را در نظر می‌گیرد و اصلا توجهی به تعداد همسایه‌های مشترک بین دو گره و مقدار تشابه آنها ندارد. طبق تعریف جامعه، گره‌هایی که با هم تشابه بیشتر دارند باید در یک انجمن قرار گیرند، به همین دلیل وجود همسایه‌های مشترک زیاد بین دو گره و همچنین در نظر گرفتن اهمیت گره‌ها در کنار آن، می‌تواند معیاری مناسب برای انتخاب برچسب مناسب باشد، چون دو گرهی که با هم همسایه مشترک بیشتری داشته باشند به این معنی است که این دو گره با هم ارتباط بیشتر دارند و تمایل بیشتری برای قرار گرفتن در یک جامعه دارند، اما وجود گره با اهمیت در اطراف یک گره حتما به معنی شباهت بیشتر آن‌ها و دلیلی بر قرار گرفتن در یک جامعه نیست، که این موضوع را الگوریتم NIB-LPA در نظر نمی‌گیرد. همچنین الگوریتم NIB-LPA تنها یک مرحله برای شکست تساوی قرار داده است، که ممکن است در اکثر مواقع این یک مرحله کافی نباشد و افزودن چندین مرحله برای شکست تساوی، سبب انتخاب دقیق‌تر برچسب شود.

در الگوریتم پیشنهادی در ابتدا گره‌ها بر اساس اهمیتشان مرتب‌سازی می‌شوند تا بدین صورت یک ترتیب ثابت برای انتخاب گره‌ها ایجاد شود. همچنین چندین مرحله برای شکست تساوی زمانی که چندین برچسب با فراوانی و یا تاثیر یکسان وجود داشته باشد، در نظر گرفته شده است.

گره‌های هسته، گره‌هایی هستند که هم درجه‌ی بالایی دارند و در مرکز جوامع قرار می‌گیرند و هم تعداد اتصالات و تشابه بین آنها و سایر گره‌ها بالا است. در واقع گره‌های هسته علاوه بر اینکه با یک گره ارتباط دارند، با همسایه‌های آن گره نیز ممکن است ارتباط بالایی داشته باشند. به همین دلیل برای انتخاب گره هسته، شاید فقط در نظر گرفتن یک معیار کافی نباشد و استفاده‌ی همزمان از دو معیار می‌تواند به انتخاب دقیق‌تر گره هسته کمک کند. به همین دلیل در روش پیشنهادی از مجموع دو معیار ارائه شده برای تعیین امتیاز کلی هسته بودن یک گره استفاده می‌کنیم. گرهی که نسبت درجه‌ی آن بر تعداد کل یال‌های ممکن در شبکه، بیشتر باشد، می‌تواند به احتمال زیادی یک گره مرکزی و هسته باشد که با بقیه گره‌ها ارتباط بیشتری دارد. برای محاسبه‌ی اهمیت یک گره از نظر درجه و تعداد یال‌ها، از رابطه (۴) استفاده می‌شود.

$$\mathrm{NC}_{deg}(\mathrm{A}) = \frac{\deg(\mathrm{A})}{\frac{\mathrm{N} * (\mathrm{N}-1)}{2}}$$

(فرمول- ۴) محاسبه‌ی اهمیت یک گره از نظر درجه و تعداد یال‌ها

در رابطه (۴)، $\mathrm{NC}_{deg}(\mathrm{A})$ همان $\mathrm{node_coreness}_{degree}(\mathrm{A})$ است که نشان دهنده‌ی مقدار اهمیت یک گره از نظر درجه و تعداد یال‌های اختصاص داده به خود است. در رابطه (۴)، $deg(A)$ نشان دهنده‌ی درجه‌ی گره A و N تعداد گره‌های کل

[8] Tie break

شبکه است. رابطه (۴) نشان می‌دهد که درجه یک گره چه کسری از کل یال‌های ممکن را به خود اختصاص داده است. علاوه بر رابطه‌ی (۴)، باید معیار دیگری برای سنجش مقدار امتیاز هسته بودن یک گره از نظر تشابه و انسجام بین گره با گره‌های همسایه وجود داشته باشد. در این مقاله از معیار جاکارد [۱۷] برای سنجش تشابه بین دو گره متصل به هم استفاده می‌کنیم. این معیار نسبت تعداد همسایه‌های مشترک بین دو گره بر تعداد اجتماع همسایه‌های دو گره را به عنوان معیار تشابه در نظر می‌گیرد. هر اندازه این مقدار بیشتر باشد دو گره مشابه‌تر هستند. گرهی که مجموع تشابه‌های آن با همسایه‌هایش بیشتر باشد، به این معنی است که این گره یک موجودیت محبوب بین سایر گره‌ها است که توانسته است اتصالات بیشتری با همسایه‌های همسایه‌ی خود داشته باشد. برای محاسبه مقدار تشابه از رابطه (۵) استفاده می‌کنیم.

$$\mathrm{Sim}_{\mathrm{I,J}}^{\mathrm{Jaccard}} = \frac{|\mathrm{N_I} \cap \mathrm{N_J}|}{|\mathrm{N_I} \cup \mathrm{N_J}|}$$

(فرمول- ۵) مقایسه تشابه بین ۲ گره

که در رابطه (۵) $\mathrm{Sim}_{\mathrm{I,J}}^{\mathrm{Jaccard}}$، نشان دهنده‌ی تشابه (similarity) بین گره I و همسایه آن گره J است. N_I مجموعه همسایه‌های گره I و N_J مجموعه همسایه‌های گره J است. طبق رابطه‌ی (۶)، مقدار امتیاز هسته بودن یک گره از نظر مجموع تشابه بین آن گره با همسایه‌هایش محاسبه می‌شود.

$$\mathrm{NC}_{sim}(\mathrm{A}) = \sum_{j \in \mathrm{N_A}} sim_{A,j}$$

(فرمول-۶)

در رابطه‌ی (۶)، $\mathrm{NC}_{sim}(\mathrm{A})$ همان $\mathrm{node_coreness}_{similarity}(\mathrm{A})$ است که نشان دهنده‌ی امتیاز هسته بودن گره A از لحاظ مجموع تشابه آن با تمامی همسایه‌هایش است. $sim_{A,j}$ نیز نشان دهنده‌ی تشابه بین گره A و گره j است که از رابطه (۵) محاسبه می‌شود. $\mathrm{N_A}$ مجموعه همسایه‌های گره A است. هر اندازه مجموع تشابه برای یک گره بیشتر باشد، آن گره احتمال زیادی دارد که یک گره هسته باشد. در نهایت برای محاسبه‌ی مقدار نهایی امتیاز هسته بودن یک گره، از رابطه (۷) استفاده می‌کنیم.

$$\mathrm{NC}_{total}(\mathrm{A}) = \mathrm{NC}_{deg}(A) + \mathrm{NC}_{sim}(A)$$

(فرمول-۷)

طبق رابطه‌ی (۷)، $\mathrm{NC}_{total}(\mathrm{A})$ همان $\mathrm{node_coreness}_{total}(\mathrm{A})$ است که مقدار امتیاز هسته بودن یک گره هم از لحاظ تعداد یال‌ها و درجه گره و هم از لحاظ مجموع تشابه گره را نشان می‌دهد. $\mathrm{NC}_{deg}(A)$ همان اهمیت گره از لحاظ درجه گره است که از رابطه (۴) محاسبه می‌شود و $\mathrm{NC}_{sim}(A)$ نشان دهنده‌ی اهمیت گره از لحاظ مجموع تشابه آن است که از رابطه (۶) به دست می‌آید. هر اندازه این رابطه بیشتر باشد گره مورد نظر ارزش بیشتری دارد و به احتمال زیاد یک گره هسته است. پس از محاسبه رابطه (۷) برای تمامی گره‌های شبکه، آنها را به صورت نزولی مرتب‌سازی می‌کنیم تا الگوریتم از گره‌های با اهمیت بیشتر شروع کند و برچسب آنها را انتشار دهد.

در قسمت پخش و انتشار برچسب از گره‌های هسته به همسایه‌هایشان، برای انتخاب دقیق‌تر گره‌های همسایه و جلوگیری از انتشار نادرست برچسب، از معیار تشابه بین دو گره استفاده می‌شود تا برچسب به درستی به گره

جدید انتشار یابد. به هنگام به‌روز رسانی برچسب یک گره، اگر چندین برچسب با فراوانی یکسان وجود داشته باشد، باید برچسب جامعه‌ای که بیشترین تشابه را با گره مورد نظر دارد، انتخاب کنیم. تشابه در نظر گرفته شده برای این بخش مشابه معیار جاکارد است، با این تفاوت که تعداد همسایه‌های مشترک بین دو گره ضرب در اهمیت گره همسایه می‌شود تا مقدار تشابه به اندازه‌ی اهمیت گره همسایه تقویت شود، در واقع هم تشابه بین دو گره و هم اهمیت گره‌ها در نظر گرفته می‌شود و این کار سبب می‌شود زمانی که مقدار تشابه بین گره مورد نظر با چند گره همسایه دیگر، برابر باشد، گره همسایه‌ای که اهمیت بیشتر دارد، برچسبش انتخاب شود. اهمیت برچسب مبتنی بر تشابه از رابطه (۸) محاسبه می‌شود.

$$\mathrm{LI(l)} = \sum_{\mathrm{j}\in \mathrm{N^l(i)}} \frac{|\mathrm{N_i} \cap \mathrm{N_j}|}{|\mathrm{N_i} \cup \mathrm{N_j}|} . \mathrm{NC}_{total}(\mathrm{j})$$

(فرمول-۸)

که در رابطه (۸)، i همان گرهی است که می‌خواهد برچسبش به‌روز رسانی شود و j همسایه‌ی گره i است، $LI(l)$ تاثیر هر برچسب جامعه، N_i و N_j به ترتیب همسایه‌های گره i و j هستند و $NC_{total}(\mathrm{j})$ نیز مقدار اهمیت یا همان امتیاز هسته بودن گره j است که از رابطه (۷) محاسبه شده است. پس از محاسبه‌ی مقدار تاثیر برچسب و تشابه گره‌های جوامع با گره مورد نظر، موثرترین برچسب جامعه از رابطه (۹) محاسبه می‌شود.

$$\mathrm{C_i} = \mathrm{arg_{l\in l_{max}}} \max \mathrm{LI(l)}$$

(فرمول-۹)

که در رابطه (۹) منظور از l_{max}، مجموعه‌ای از برچسب‌ها با بیشترین مقدار فراوانی در بین سایر برچسب‌ها است. در صورتی که پس از محاسبه‌ی رابطه (۹)، جوامعی وجود داشته باشند که مقدار تاثیر برچسب آنها باهم برابر باشد، از رابطه (۱۰) مقدار مجموع امتیاز هسته بودن(مجموع اهمیت گره‌ها از نظر درجه و یال اختصاص داده شده به خود) گره‌های آن جامعه که با گره مورد نظر ارتباط دارند، محاسبه خواهد شد.

$$\mathrm{CS(C_x)} = \sum_{\mathrm{j}\in \mathrm{N(i)}, \mathrm{j}\in \mathrm{C_x}} \mathrm{NC}_{deg}(\mathrm{j})$$

(فرمول-۱۰)

در رابطه (۱۰)، $CS(C_x)$ نشان دهنده‌ی مجموع اهمیت گره‌های جامعه x که با گره i (گره مورد نظر) ارتباط دارند، است و $\mathrm{NC}_{deg}(j)$ همان اهمیت گره j است که از رابطه (۴) محاسبه می‌شود. برای انتخاب برچسب جامعه‌ای با بیشترین مجموع امتیاز گره‌ها، از رابطه (۱۱) استفاده می‌کنیم.

$$\mathrm{C_i} = \arg\max \mathrm{CS(C_x)}$$

(فرمول-۱۱)

که در رابطه (۱۱)، $\mathrm{C_i}$ نشان دهنده‌ی موثرترین و با اهمیت‌ترین جامعه در بین سایر جوامع اطراف گره مورد نظر است. مراحل الگوریتم پیشنهادی به صورت زیر است:

۱) تخصیص برچسب صفر به تمامی گره‌های شبکه.
۲) محاسبه‌ی مقدار اهمیت و امتیاز هسته بودن هر گره با استفاده از رابطه (۷) و مرتب‌سازی نزولی گره‌ها بر اساس اهمیتشان.
۳) شروع الگوریتم و انتخاب با اهمیت‌ترین گره و اجرای الگوریتم. الگوریتم پیشنهادی شامل فاز پخش برچسب از گره‌های هسته به سمت گره-های همسایه و همچنین فاز انتخاب گره‌هایی با

برچسب صفر و به‌روز رسانی برچسب آنها است. پس از پایان فاز پخش برچسب از گره هسته به سمت همسایه‌های آن، سراغ گره‌هایی با برچسب صفر خواهیم رفت.

الف) پخش برچسب گره هسته به همسایه‌های مستقیم: مرحله‌ی پخش برچسب از گره هسته به سمت گره‌های همسایه، برای گره‌هایی که مقدار اهمیت آنها بیشتر از میانگین اهمیت گره-های شبکه باشد، اجرا می‌شود. اگر گره هسته، برچسبی به غیر از صفر داشته باشد (قبلا برچسب دریافت کرده باشد) سراغ گره هسته‌ی دیگری خواهیم رفت. پس از انتخاب یک گره هسته، شماره‌ی گره هسته به عنوان برچسب آن انتخاب می‌شود سپس اگر همسایه‌های مستقیم آن، برچسب صفر داشته باشند، گره هسته برچسب خود را به آنها انتشار خواهد داد. در غیر این صورت باید بررسی شود که برچسب گره هسته‌ی جدید موثرتر است یا برچسب گره هسته قبلی که گره مورد نظر(گره همسایه‌ی گره هسته جدید) آن را دریافت کرده است؟ ممکن است گره همسایه‌ی گره هسته‌ی جدید، که قبلا برچسب یک گره هسته‌ی دیگر را دریافت کرده است، همسایه‌ی مستقیم گره هسته‌ی قبلی نباشد، پس به همین دلیل اگر یال مستقیمی بین گره هسته قبلی با گرهی که برچسب دارد، وجود نداشته باشد، برچسب گره هسته‌ی جدید به همسایه‌ی آن انتقال می‌یابد و در صورتی که هر دو گره هسته(گره هسته‌ی جدید و گره هسته‌ی قبلی)، با گرهی که برچسب دارد، یال مستقیم داشته باشد، باید مقدار تشابه از رابطه (۵) برای هر یک از گره‌های هسته با گره همسایه، که می‌خواهد برچسب جدید بگیرد محاسبه شود. اگر مقدار تشابه گره هسته‌ی جدید با همسایه‌اش بیشتر از مقدار تشابه گره هسته قبلی باشد، برچسب گره هسته‌ی جدید به همسایه‌اش انتشار خواهد یافت.

ب) پخش برچسب گره هسته به همسایه‌های غیر مستقیم (همسایه‌های سطح۲): پس از اینکه برچسب گره هسته‌ی جدید به همسایه‌های مستقیمش(همسایه‌های سطح۱) انتشار یافت، برای گسترش بیشتر برچسب گره هسته و کاهش تعداد شکست تساوی در مراحل بعدی، برچسب گره هسته پس از پخش به گره‌های همسایه‌ی مستقیم خود، یک سطح دیگر پخش خواهد یافت. از بین همسایه‌های گره هسته‌ی جدید، که برچسب یکسان با برچسب گره هسته دارند، همسایه‌ای که بیشترین اهمیت را دارد و مقدار اهمیت آن بیشتر از میانگین اهمیت کل گره‌های شبکه باشد، انتخاب می‌شود و برچسب گره هسته را به همسایه‌های خودش که برچسب صفر دارند، پخش می‌کند. مراحل (الف) و (ب) را از با اهمیت‌ترین گره شروع کرده و تا زمانی که مقدار اهمیت گره هسته‌ی انتخاب شده از میانگین اهمیت گره‌های شبکه بیشتر باشد، ادامه می‌دهیم. این مرحله فقط یکبار انجام می-شود و نیازی به تکرار ندارد.

۴) پس از اتمام مرحله ۳، گره‌هایی که برچسب صفر دارند باید به همان ترتیب مرتب شده بر اساس اهمیتشان، بررسی شوند و برچسب خود را به‌روز رسانی کنند. گره انتخاب شده باید برچسب

جامعه‌ای با بیشترین فراوانی را به عنوان برچسب خود انتخاب کند. به این دلیل از معیار بیشترین فراوانی استفاده می‌شود که در مراحل قبلی گره-های هسته برچسب خود را ۲ سطح انتشار داده-اند و به احتمال زیاد برچسب آنها فراوانی بیشتری نسبت به سایر گره‌های کم اهمیت دارد، پس منطقی است که گره، در ابتدا برچسب با بیشترین فراوانی را انتخاب کند. در صورتی که چندین برچسب با بیشترین مقدار فراوانی وجود داشته باشد باید از رابطه (۸) تاثیر هر برچسب جامعه محاسبه شود(جامعه‌ای که بیشترین تشابه را با گره داشته باشد) و طبق رابطه (۹) موثرترین برچسب جامعه انتخاب شود. اگر دوباره در حالت شکست تساوی قرار بگیریم و چندین جامعه با مقدار تاثیر یکسان وجود داشته باشد، با استفاده از رابطه (۱۰) مجموع اهمیت گره‌های هر جامعه (از نظر درجه و یال‌های اختصاص داده شده به خود) که با گره انتخاب شده، یال داشته باشند محاسبه شده و طبق رابطه (۱۱) جامعه‌ای که مجموع اهمیت بیشتری داشته باشد، انتخاب می‌شود. در نهایت اگر چندین جامعه با مجموع اهمیت برابر وجود داشته باشد، یکی را به تصادف انتخاب می‌کنیم.

۵) پس از خاتمه‌ی مرحله‌ی ۴، ممکن است تعدادی از گره‌ها همچنان برچسب صفر داشته باشند. در صورت وجود گره‌هایی با برچسب صفر این مرحله اجرا خواهد شد. در این مرحله، گره‌هایی که برچسب صفر دارند، به ترتیب اهمیتشان به‌روز رسانی می‌شوند. به ازای هر گره که برچسب صفر دارد مقدار تشابه آن با گره‌های همسایه‌اش که در جوامع مختلف هستند، از رابطه (۵) محاسبه می‌شود و برچسب جامعه‌ای را که بیشترین تشابه را با گره داشته باشد، انتخاب می‌کند. در صورتی که چندین جامعه با مقدار تشابه برابر وجود داشته باشد، برچسب اولین جامعه را انتخاب می-کنیم و از انتخاب تصادفی خودداری می‌نماییم.

۶) پس از اتمام مرحله ۵، شاید به ندرت گره‌هایی وجود داشته باشند که هنوز برچسب صفر دارند. این می‌تواند به این دلیل باشد که احتمالا در شبکه برخی جوامع کوچک(مثلا با ۱، ۲ و یا ۳ گره و...) وجود داشته باشد که کاملا از سایر جوامع جدا هستند و هیچگونه یالی با آنها ندارند و از طرفی هم هیچکدام از گره‌های این جوامع اهمیت بیشتری از میانگین اهمیت گره‌های شبکه نداشته باشد، در نتیجه در هیچکدام از مراحل قبلی برچسب مناسب خود را دریافت نکرده باشند، به همین دلیل این مرحله برای برچسب‌دهی به جوامع جدا و کوچک در نظر گرفته شده است که در صورت وجود گره‌هایی با برچسب صفر اجرا خواهد شد. در این مرحله گرهی که برچسب صفر دارد، شماره‌ی خود را به عنوان برچسب جدید خود انتخاب می‌کند و آن را به تمامی همسایه‌های خود که برچسب صفر دارند انتشار می‌دهد.

۷) شرط پایان الگوریتم به این صورت است که یا برچسب گره‌ها تغییری نکند و یا تعداد تکرارهای در نظر گرفته شده پایان یابد. لازم به ذکر است که تکرار الگوریتم فقط از مرحله‌ی ۴ تا مرحله‌ی ۶ انجام می‌شود و نیازی به تکرار مرحله ۱ نیست.

پس از پایان الگوریتم، گره‌هایی که برچسب یکسان دارند در یک جامعه قرار می‌گیرند.

شبه کد مربوط به مراحل الگوریتم پیشنهادی در الگوریتم ۱ تا الگوریتم ۳ بیان شده است.

الگوریتم۱- الگوریتم اصلی

Input: Network G (V, E)
Output: Final communities
1) **Execute Algorithm 2**
2) **While** iteration ≠ maximum iteration OR there is no change in labels **Do**
   - Call Algorithm 3

الگوریتم۲- پخش برچسب گره هسته

Input: Network G (V, E)
Output: All labeled core nodes with some other labeled nodes
1) Set all nodes label to zero
2) Calculate equation (7) and sort nodes descending
3) **For** all nodes v_i that importance (v_i) ≥ Average (all nodes importance)
   I. **Give a label to the core node and diffuse its label to its direct neighbors according to the diffusion rules**
4) **Select** v_z ∈ direct neighbors of core node with maximum importance which label (v_z)=label (v_i)
If importance (v_z) ≥ Average (all nodes importance)
Then for all v_k ∈ neighbors (v_z)
If label (v_k) =0 Then label (v_k)=label (v_z)

الگوریتم۳- انتشار برچسب برای گره‌های بدون برچسب

Input: All unlabeled nodes (nodes with label=0)
Output: Some labeled nodes
1) **For** all v_i ∈ sorted_nodes which label (v_i)=0
   - Select the most appropriate label an assign it to the v_i
2) **else If** ∃ v_i ∈ sorted_nodes which label (v_i)=0
   - Calculate equation (5) for all communities around v_i
   - Then select a community label which has the highest similarity with v_i
3) **else If** ∃ v_i ∈ sorted_nodes which label (v_i)=0
   - Label (v_i)=ID (v_i)
   - Then give v_i's label to all neighbors with label=0

۴-۳- پیچیدگی زمانی

در این بخش به تحلیل پیچیدگی زمانی الگوریتم پیشنهادی و سایر الگوریتم‌های مورد آزمایش می‌پردازیم. در یک شبکه، n تعداد گره‌ها، m تعداد یال‌ها و k نیز نشان دهنده‌ی درجه هر گره فرض می‌شود. پیچیدگی زمانی الگوریتم LPA برابر $O(m)$ است. الگوریتم NIB-LPA برای محاسبه‌ی اهمیت گره از تجزیه‌ی K-shell که پیچیدگی زمانی آن برابر $O(m)$ و مرتب-سازی گره‌ها که برابر $O(n\log n)$ است، استفاده می-کند که در کل پیچیدگی آن برابر $O(n\log n)$ است. الگوریتم LPA-Intimacy برای محاسبه‌ی اهمیت گره‌ها از معیار تاثیر استفاده می‌کند که محاسبه‌ی آن مستلزم اشتراک‌گیری بین مجموعه همسایه‌های دو گره است که پیچیدگی آن برابر $O(nk^2)$ است. چون این روش از مرتب‌سازی نیز استفاده کرده است، در کل پیچیدگی آن برابر $O(n\log n)$ است.

در الگوریتم پیشنهادی نیز برای محاسبه‌ی اهمیت یک گره باید مجموع تشابه آن که پیچیدگی برابر $O(nk^2)$ دارد استفاده شود. مرتب‌سازی گره‌ها برابر $O(n\log n)$ است. فاز پخش برای x تا گره که اهمیت بیشتر از میانگین دارند اجرا می‌شود که پیچیدگی آن برابر $O(xk)$ است. برای P تا گره باقی مانده که برچسب نگرفته‌اند ممکن است ۳ بار مرحله انتخاب برچسب مناسب اجرا شود که برابر $O(3Pk)$است. سپس در مراحل بعدی نیز به ازای q گرهی که برچسب ندارند باید مناسب‌ترین برچسب انتخاب شود که پیچیدگی آن برابر

$O(qk)$ است. اما باید در نظر داشت که چون در مرحله‌ی پخش برچسب گره‌های هسته، بیشتر گره‌ها صاحب برچسب می‌شوند، عملا مقدار p و q خیلی کوچک خواهد بود. در کل پیچیدگی زمانی الگوریتم پیشنهادی برابر $O(n\log n)$ است. باید توجه داشت که هرچند پیچیدگی زمانی LPA کمتر از الگوریتم پیشنهادی است و دو الگوریتم NIB-LPA و LPA-Intimacy پیچیدگی برابر با الگوریتم پیشنهادی دارند، اما به دلیل تعداد مقایسات بالا برای هر گره طی چندین تکرار، سرعت اجرای آنها بسیار پایین‌تر از الگوریتم پیشنهادی است و نتایج آزمایش‌های انجام شده این ادعا را ثابت می‌کند.

۴ – نتایج تجربی

آزمایش‌های متعددی بر روی چندین دیتاست واقعی و مصنوعی توسط الگوریتم پیشنهادی و الگوریتم‌های LPA، NIB-LPA، LPA-Intimacy انجام شده است. برای بررسی و مقایسه‌ی نتایج به دست آمده، از جداول و نمودارهایی استفاده شده است که در بخش‌های بعدی به طور کامل بیان خواهند شد. برای انجام تمامی آزمایش‌ها از یک کامپیوتر ثابت، با پردازنده‌ی Core-i5(3.40 GHz) با حافظه‌ی 12 GB و در محیط برنامه نویسی MATLAB(2017) و با سیستم عامل ویندوز ۱۰ انجام شده است.

۴-۱ معرفی دیتاست‌ها

از ۱۳ دیتاست واقعی و ۳ نوع دیتاست مصنوعی LFR[9] که هر کدام ۸ نوع پارامتر μ دارند، برای انجام آزمایش‌ها و بررسی عملکرد الگوریتم‌ها استفاده شده است. اطلاعات مربوط به دیتاست‌های واقعی در جدول (۱) نمایش داده شده است.

(جدول-۱): مشخصات مربوط به ۱۳ دیتاست استفاده شده در آزمایش‌ها

دیتاست	تعداد گره	تعداد یال	تعداد جوامع
Karate	۳۴	۷۸	۲
Dolphins	۶۲	۱۵۹	۲
Polbooks	۱۰۵	۴۴۱	۳
Football	۱۱۵	۶۱۳	۱۲
Email	۱۱۳۳	۵۴۵۱	نامشخص
Netscience	۱۵۸۹	۲۷۴۲	نامشخص
Power	۵۲۴۲	۱۴۴۹۰	نامشخص
CA-GRQC	۵۲۴۲	۱۴۴۹۰	نامشخص
Collaboration	۸۳۶۱	۱۵۷۵۱	نامشخص
PGP	۱۰۶۸۰	۲۴۳۱۶	نامشخص
Condmat-2003	۳۱۱۶۳	۱۲۰۰۲۹	نامشخص
Email-Enron	۳۶۶۹۲	۱۸۳۸۳۱	نامشخص
Condmat-2005	۴۰۴۲۱	۱۷۵۶۹۱	نامشخص

LFR [۱۸]، یکی از برنامه‌های مشهور برای ایجاد دیتاست‌های مصنوعی است که از آن برای ایجاد دیتاست‌های مصنوعی در آزمایش‌ها استفاده شده است. LFR شامل پارامترهای قابل تنظیم برای ایجاد دیتاست‌هایی مشابه با ویژگی‌های دیتاست‌های جهان واقعی است. پارامترهای مربوط به ساخت دیتاست‌های مصنوعی در

[9] Lancichinetti–Fortunato–Radicchi

LFR، در جدول (۲) و مشخصات دیتاست‌های ایجاد شده، در جدول (۳) نشان داده شده است.

(جدول-۲): پارامترهای مربوط به دیتاست‌های LFR

پارامترها	توضیح
N	تعداد گره‌های جامعه
Min K	حداقل درجه گره‌ها
Max K	حداکثر درجه گره‌ها
μ	ضریب اختلاط جوامع[10]
Min c	تعداد گره‌ها در کوچک‌ترین جامعه
Max c	تعداد گره‌ها در بزرگ‌ترین جامعه
γ	توان پارامترها برای توزیع درجه گره‌ها
β	توان پارامترها برای توزیع اندازه جوامع

(جدول-۳): مشخصات دیتاست‌های مصنوعی LFR ایجاد شده

پارامترها	LFR1	LFR2	LFR3
N	۵۰۰۰	۱۰۰۰۰	۲۰۰۰۰
Min K	۲۰	۲۰	۲۰
Max K	۵۰	۵۰	۵۰
Min c	۱۰	۲۰	۲۰
Max c	۵۰	۱۰۰	۱۰۰
γ	۲	۲	۲
β	۱	۱	۱
μ	۰,۸-۰,۱	۰,۸-۰,۱	۰,۸-۰۱

۲-۴ پارامترهای ارزیابی

با توجه به ماهیت تصادفی الگوریتم LPA، ممکن است این الگوریتم در هر اجرا نتایج متفاوتی داشته باشد و از طرفی برای به دست آوردن نتایج قابل قبول، بهتر است الگوریتم چندین دور تکرار شود تا گره‌ها تقریبا برچسب خود را کمتر تغییر دهند و به نوعی به یک همگرایی برسند. به همین دلیل در آزمایش‌های انجام شده هر چهار الگوریتم LPA، LPA-Intimacy، NIB-LPA و الگوریتم پیشنهادی تنها یکبار اجرا شده و ۱۰ تکرار داشته‌اند (به عنوان مثال در الگوریتم LPA یکبار ترتیب تصادفی گره‌ها ایجاد شده و الگوریتم ۱۰ بار با همان ترتیب تصادفی، تکرار شده است). با توجه به نتایج آزمایش‌های انجام شده در مقاله‌ی [۵] که برای تعیین مقدار صحیح پارامتر α برای الگوریتم NIB-LPA است، در این مقاله نیز مقدار α را برای الگوریتم NIB-LPA برابر ۰,۲ ($\alpha = 0.2$) در نظر می‌گیریم.

از معیارهای NMI[11]، F-measure و ماژولاریتی برای ارزیابی و مقایسه‌ی دیتاست‌ها استفاده شده است. در دیتاست‌هایی که تعداد جوامع آنها از پیش مشخص است (دیتاست‌های دارای گراندتروس[12])، از معیار NMI و F-measure و برای دیتاست‌هایی که تعداد جوامع مشخصی ندارند، از معیار ماژولاریتی برای ارزیابی استفاده می‌شود.

معیار NMI، معیاری برای ارزیابی تشابه بین نتایج حاصل از جوامع کشف شده و افراز واقعی جامعه است. هر اندازه مقدار به ۱ نزدیک باشد، الگوریتم دقت بیشتری دارد [۱۹]. مقدار NMI از رابطه (۱۲) محاسبه می‌شود.

[10] Mixing parameter
[11] Normalized mutual information
[12] Ground truth

$$\mathrm{NMI(A,B)} = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} N_{ij}\log\left(\frac{N_{ij}N}{N_i N_j}\right)}{\sum_{i=1}^{C_A} N_i \log\left(\frac{N_i}{N}\right)+\sum_{j=1}^{C_B} N_j \log\left(\frac{N_j}{N}\right)}$$ (فرمول-۱۲)

در رابطه (۱۲)، A نشان دهنده‌ی جوامع واقعی و B جوامع کشف شده توسط الگوریتم است. C_A تعداد جوامع واقعی، C_B تعداد جوامع کشف شده، N تعداد کل گره-های موجود در شبکه، N_{ij} برابر تعداد گره‌های مشترک بین جامعه واقعی i در مجموعه A با جامعه کشف شده j در مجموعه B، N_i برابر مجموع سطر i در ماتریس N_{ij} و N_j برابر مجموع ستون j در ماتریس N_{ij} است.

معیار F-measure، معیاری برای اندازه‌گیری و ارزیابی میزان دقت نتایج حاصل از آزمایش الگوریتم بر روی یک دیتاست است. F-measure به عنوان میانگین هارمونی Precision و Recall تعریف می‌شود [۲۰, ۲۱]. از F-measure اغلب در مسائل مربوطه به کلاس-بندی برای ارزیابی دقت الگوریتم استفاده می‌شود. F-measure از رابطه‌ی (۱۳) محاسبه می‌شود.

$$F-measure = 2 * \frac{precision * recall}{precision + recall}$$

(فرمول-۱۳)

در رابطه (۱۳) معیار Precision و Recall به ترتیب از رابطه (۱۴) و رابطه (۱۵) محاسبه می‌شود.

$$Precision = \frac{TP}{TP + FP}$$

(فرمول-۱۴)

در رابطه‌ی (۱۴) منظور از TP همان نتایج درست مثبت و FP همان نتایج نادرست مثبت است.

$$Recall = \frac{TP}{TP + FN}$$

(فرمول-۱۵)

در رابطه (۱۵) نیز منظور از FN همان نتایج نادرست منفی است.

معیار ماژولاریتی، معیاری برای سنجش کیفیت افراز و چگالی جوامع به دست آمده است. ماژولاریتی در واقع تراکم یال‌های داخلی یک جامعه نسبت به یال‌های بین جوامع را می‌سنجد. مقدار این رابطه از مقایسه‌ی تعداد یال‌های داخل یک جامعه نسبت به مقدار مورد انتظار برای یک شبکه تصادفی از همان اندازه و درجه، به دست می‌آید. هر اندازه این مقدار به ۱ نزدیک باشد، افراز خوبی انجام شده است [۸]. رابطه (۱۶) نحوه‌ی محاسبه‌ی معیار ماژولاریتی را نشان می‌دهد.

$$\mathrm{Q} = \frac{1}{2\mathrm{m}}\sum_{\mathrm{i,\,j}\,\epsilon\,\mathrm{V}}\left(\mathrm{A_{ij}} - \frac{\mathrm{d_i d_j}}{2\mathrm{m}}\right) * \delta(\mathrm{c_i}, \mathrm{c_j})$$

(فرمول-۱۶)

در رابطه (۱۶)، m نشان دهنده‌ی تعداد یال‌های موجود در شبکه و A ماتریس همسایگی گره‌های شبکه است. اگر بین گره i و گره j یالی وجود داشته باشد $A_{ij} = 1$ و در غیر این صورت برابر صفر خواهد بود. c_i و c_j به ترتیب برچسب گره i و گره j است. $\delta(\mathrm{c_i}, \mathrm{c_j})$ همان دلتای کرونکر[۱۳] است و در صورتی که دو گره

[13] Kronecker delta

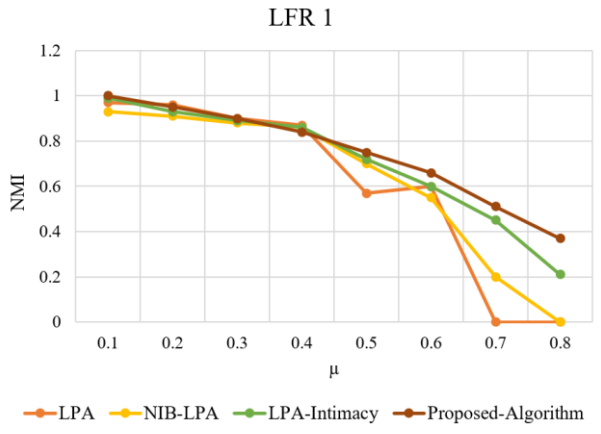
برچسب یکسان داشته باشند $\delta(c_i, c_j) = 1$ و در غیر این صورت برابر صفر خواهد بود.

یکی از مباحث مطلوب در الگوریتم‌های مبتنی بر انتشار برچسب، کاهش تعداد شکست تساوی است. شکست تساوی زمانی که چندین برچسب فراوانی و یا تاثیر یکسانی داشته باشند اتفاق می‌افتد. شکست تساوی اغلب برای انتخاب دقیق‌تر برچسب جامعه استفاده می‌شود، اما از طرفی هم سبب افزایش زمان اجرای الگوریتم می‌شود. در قسمت آزمایش‌ها، علاوه بر معیارهای NMI، F-measure و ماژولاریتی، از یک معیار دیگر با عنوان شکست تساوی(Tie break) برای مقایسه‌ی تعداد دفعاتی که الگوریتم باید از بین چندین برچسب که فراوانی و یا تاثیر یکسانی دارد، یکی را انتخاب کند، استفاده می‌کنیم. پایین بودن دفعات شکست تساوی می‌تواند نشان دهنده‌ی قاطعیت الگوریتم در پخش برچسب‌ها باشد، چون هر اندازه تعداد شکست تساوی کم باشد، به این معنی است که الگوریتم توانسته برچسب را درست پخش کند و در انتخاب برچسب گره‌های دیگر کمتر حالتی پیش آمده است که الگوریتم مابین چندین برچسب یکی را انتخاب کند. ممکن است در الگوریتمی چندین مرحله برای انجام شکست تساوی در نظر گرفته شده باشد، اما اگر مراحل اولیه تشخیص جوامع به خوبی صورت گیرد، تعداد شکست تساوی نیز کمتر خواهد بود. عدم وجود مرحله‌ی شکست تساوی سبب کاهش دقت الگوریتم می‌شود و اتخاذ مراحل زیاد برای شکست تساوی سبب زمانگیر بودن اجرای الگوریتم می‌شود، پس بهتر است مراحل اولیه‌ی تعیین برچسب دقت کافی داشته باشند تا آنجایی که ممکن است سراغ مرحله شکست تساوی نرویم تا سرعت الگوریتم کاهش نیابد.

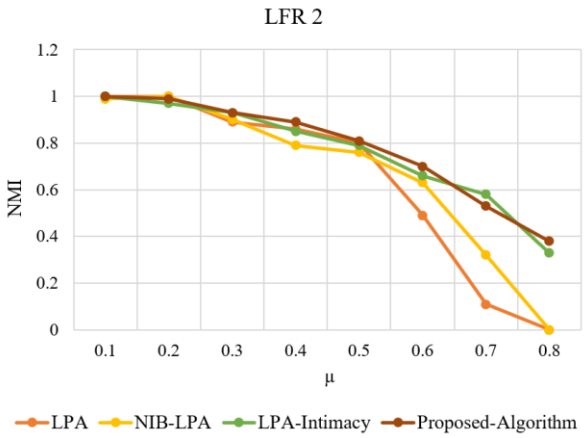
همچنین در یک آزمایش دیگر، الگوریتم‌های مورد آزمایش از لحاظ سرعت همگرایی و رسیدن به یک نتیجه پایدار و هم از لحاظ زمان اجرای الگوریتم مورد آزمایش قرار می‌گیرند تا عملکرد و سرعت اجرای آنها نسبت به هم مورد ارزیابی قرار گیرد.

۴-۳ نتایج به دست آمده

شکل‌های (۳)، (۴) و (۵) مقادیر NMI به دست آمده از آزمایش‌های انجام شده توسط الگوریتم پیشنهادی، الگوریتم LPA، NIB-LPA و LPA-Intimacy را بر روی دیتاست‌های مصنوعی با ۵۰۰۰، ۱۰۰۰۰ و ۲۰۰۰۰ گره نشان می‌دهد. مقادیر موجود در جدول‌ها و شکل‌ها، نتایج حاصل از ۱۰ تکرار اجرای هر یک از الگوریتم‌های LPA، NIB-LPA، LPA-Intimacy و الگوریتم پیشنهادی است. در آزمایش‌های انجام شده، هر الگوریتم ۱ بار اجرا شده و به تعداد ۱۰ تکرار ادامه می‌یابد و پس از پایان دهمین تکرار، تعداد جوامع، مقدار NMI و مقدار ماژولاریتی محاسبه می‌شود. نتایج حاصل برای مقادیر NMI، F-measure و ماژولاریتی در آزمایش‌های انجام شده بر روی دیتاست‌های واقعی در جدول‌های (۴)، (۵) و (۶) نمایش داده شده است. همچنین تعداد شکست تساوی‌ها در هر مرحله از تکرار محاسبه شده و پس از پایان ۱۰ تکرار، میانگین تعداد شکست تساوی نیز محاسبه می‌شود. نتایج حاصل از مقایسه‌ی شکست تساوی در شکل‌های (۱۱) تا (۱۸) نمایش داده شده است.



(شکل-۳): مقایسه نتایج اجرای ۴ الگوریتم LPA، NIB-LPA، LPA-Intimacy و الگوریتم پیشنهادی بر روی دیتاست LFR1 با ۵۰۰۰ گره

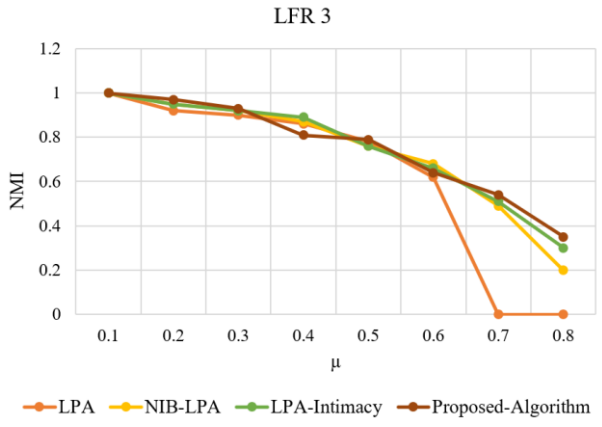


(شکل-۴): مقایسه نتایج اجرای ۴ الگوریتم LPA، NIB-LPA، LPA-Intimacy و الگوریتم پیشنهادی بر روی دیتاست LFR2 با ۱۰۰۰۰ گره

در شکل (۳) زمانی که $0.1 \leq \mu \leq 0.3$ است در نقاط $\mu = 0.1$ و $\mu = 0.3$، الگوریتم پیشنهادی بالاترین مقدار NMI را نسبت به الگوریتم LPA-Intimacy، NIB-LPA و LPA دارد و تنها زمانی که $\mu = 0.2$ است، LPA با تفاوت بسیار اندکی NMI بیشتری نسبت به الگوریتم پیشنهادی دارد. در بازه‌ی $0.1 \leq \mu \leq 0.3$، الگوریتم NIB-LPA کمترین مقدار NMI را دارد. زمانی که $\mu = 0.4$ است، الگوریتم LPA، NIB-LPA و LPA-Intimacy به ترتیب مقدار NMI بیشتری نسبت به الگوریتم پیشنهادی دارند. اما در بازه‌ی $0.5 \leq \mu \leq 0.8$، الگوریتم LPA و NIB-LPA کاهش قابل توجهی در مقدار NMI دارند و در این بازه الگوریتم پیشنهادی بهترین مقدار NMI را دارد و مقدار NMI الگوریتم پیشنهادی در کل این بازه بیشتر از الگوریتم LPA-Intimacy است. زمانی که $0.7 \leq \mu \leq 0.8$ است، الگوریتم LPA مقدار NMI صفر و زمانی که $\mu = 0.8$ است، الگوریتم NIB-LPA مقدار صفر دارند در حالی که الگوریتم پیشنهادی در این بازه مقدار غیر صفر و در حدود ۰,۴ دارد.

با توجه به نتایج موجود در شکل (۴)، در بازه‌ی $0.1 \leq \mu \leq 0.2$، تقریبا هر سه الگوریتم پیشنهادی، LPA، NIB-LPA و LPA-Intimacy نتایج بسیار نزدیکی به هم دارند و تنها الگوریتم LPA-Intimacy با تفاوت اندکی کمترین مقدار را دارد. در $\mu = 0.1$ الگوریتم پیشنهادی و در $\mu = 0.2$، NIB-LPA با اختلاف بسیار اندکی بالاترین مقدار NMI را دارند. در حالت کلی در بازه‌ی $0.3 \leq \mu \leq 0.8$، الگوریتم پیشنهادی به طور مطلق مقدار NMI بالاتری نسبت به ۳ الگوریتم LPA، NIB-LPA و LPA-Intimacy دارد و فقط زمانی که $\mu = 0.7$ است، الگوریتم LPA-Intimacy مقدار NMI بهتری دارد. با توجه به شکل (۴) می‌توان مشاهده کرد که به ازای $\mu > 0.5$ الگوریتم LPA و به ازای $\mu > 0.6$ الگوریتم NIB-LPA کاهش چشمگیری در مقدار NMI دارند، اما الگوریتم پیشنهادی در کل آزمایش‌های انجام شده کاهش شدیدی

در انتقال از یک بازه به یک بازه‌ی دیگر ندارد و تقریبا شیب تغییرات ملایم‌تری دارد. از بین الگوریتم‌های مورد آزمایش می‌توان گفت که الگوریتم LPA-Intimacy تقریبا رفتار مشابهی با الگوریتم پیشنهادی دارد اما الگوریتم پیشنهادی مقادیر NMI بهتری نسبت به آن دارد. در حالت کلی می‌توان مشاهده کرد زمانی که شبکه پیچیدگی و تعداد اتصال‌های بالایی دارد و تفکیک‌پذیری جوامع سخت است، الگوریتم پیشنهادی می‌تواند جوامع را با دقت قابل قبولی کشف کند و عملکرد بهتری نسبت به دو الگوریتم دیگر دارد. زمانی که $\mu = 0.8$ است، هر دو الگوریتم LPA و NIB-LPA مقدار NMI=0 دارند اما الگوریتم پیشنهادی در $\mu = 0.8$ دوباره مقدار NMI غیر صفر دارد. شکل (۵)، نشان دهنده‌ی مقادیر NMI به دست آمده در دیتاست LFR3 با ۲۰۰۰۰ گره است. این دیتاست بزرگ‌ترین دیتاست موجود در دیتاست‌های مصنوعی LFR ایجاد شده برای انجام آزمایش‌ها است. به ازای $\mu = 0.1$، هر چهار الگوریتم نتایج برابری دارند. زمانی که $0.2 \leq \mu \leq 0.3$ است، الگوریتم پیشنهادی بالاترین مقدار NMI را نسبت به سه الگوریتم دیگر کسب کرده است و LPA در این بازه کمترین مقدار NMI را دارد. فقط در نقاط $\mu = 0.4$ و $\mu = 0.6$، الگوریتم NIB-LPA و LPA-Intimacy مقدار بهتری نسبت به الگوریتم پیشنهادی دارد و در سایر نقاط، به ازای $\mu = 0.5$، $\mu = 0.7$ و $\mu = 0.8$ الگوریتم پیشنهادی بهترین مقدار NMI را کسب کرده است. به ازای $\mu = 0.7$



(شکل-۵): مقایسه نتایج اجرای ۴ الگوریتم LPA، NIB-LPA، LPA-Intimacy و الگوریتم پیشنهادی بر روی دیتاست LFR3 با ۲۰۰۰۰ گره

و $\mu = 0.8$، الگوریتم LPA مقادیر صفر دارد. زمانی که $\mu = 0.8$ است، الگوریتم پیشنهادی با اختلاف قابل توجهی، NMI بهتری نسبت به الگوریتم NIB-LPA دارد. مقایسه های انجام شده نشان می‌دهد که الگوریتم پیشنهادی زمانی که دیتاست وضوح کمتری دارد و تفکیک جوامع سخت‌تر است، نتایج قابل قبولی ارائه می-دهد.

جدول (۴) مقایسه‌ای بین نتایج حاصل از آزمایش چهار الگوریتم LPA، NIB-LPA، LPA-Intimacy و الگوریتم پیشنهادی بر روی دیتاست‌های واقعی را با در نظر گرفتن معیار NMI نشان می‌دهد. همانطور که از جدول (۴) مشخص است، الگوریتم پیشنهادی در دیتاست‌های Karate و Dolphins و Polbooks بیشترین مقدار NMI را کسب کرده است. در دیتاست-های Karate و Dolphins، الگوریتم پیشنهادی اختلاف چشمگیری در مقدار NMI به دست آورده با دو الگوریتم LPA و NIB-LPA دارد و در دیتاست Karate مقدار برابر و در دیتاست Dolphins و

Polbooks مقدار بیشتری نسبت به الگوریتم LPA-Intimacy دارد. در دو دیتاست Karate و Dolphins پس از الگوریتم پیشنهادی، الگوریتم LPA-Intimacy

(جدول-۴): مقایسه‌ی نتایج الگوریتم پیشنهادی با الگوریتم‌های LPA، NIB-LPA و LPA-Intimacy بر روی دیتاست‌های واقعی بر اساس معیار NMI

دیتاست	LPA	NIBLPA	LPA-Intimacy	الگوریتم پیشنهادی
Karate	۰,۸۱	۰,۵۸	۱	۱
Dolphins	۰,۵۲	۰,۵۰	۰,۶۲	۰,۶۵
Polbooks	۰,۴۹	۰,۵۴	۰,۵۴	۰,۵۹
Football	۰,۸۱	۰,۷۲	۰,۸۴	۰,۷۶

بیشترین مقادیر را دارد. در دیتاست Football الگوریتم LPA-Intimacy بیشترین مقدار NMI را دارد و الگوریتم پیشنهادی پس از الگوریتم LPA، سومین مقدار NMI را دارد. در هر ۴ دیتاست موجود در جدول (۴)، الگوریتم پیشنهادی مقدار NMI بیشتری از الگوریتم NIB-LPA کسب کرده است. در حالت کلی الگوریتم پیشنهادی در بین ۴ دیتاست گراندتروس‌دار، در ۳ مورد بیشترین مقدار را کسب کرده است. باید توجه داشت که الگوریتم LPA-Intimacy یکی از الگوریتم‌های با دقت بالا است که برای بهبود الگوریتم LPA ارائه شده است و الگوریتم پیشنهادی ما توانسته است مقادیر بهتری نسبت به الگوریتم LPA-Intimacy کسب کند.

الگوریتم LPA در دیتاست‌های Karate، Dolphins و Football مقدار NMI بهتری نسبت به الگوریتم NIB-LPA دارد. اما از آنجایی که الگوریتم LPA ماهیت تصادفی بسیار بالایی دارد، نتایج آن قابل اطمینان نیست و در اجراهای متفاوت ممکن است نتایج مختلفی داشته باشد. در طی اجرای تکراری الگوریتم (تکرار ۱۰ دور الگوریتم)، مشاهده می‌شد که الگوریتم پیشنهادی در اکثر تکرارها نتایج ثابت و یکسانی ارائه می‌دهد، در حالی که الگوریتم LPA بیشترین تغییرات در جواب‌های به دست آمده

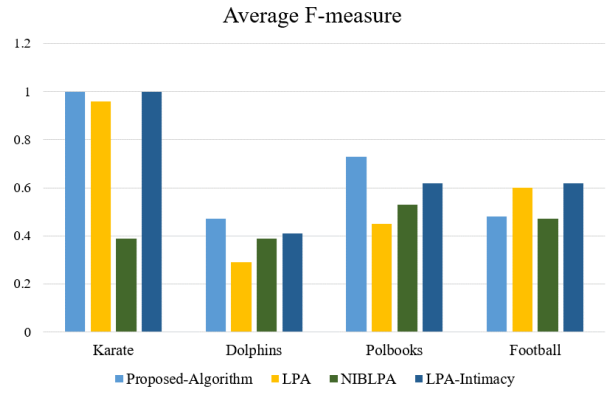
(جدول-۵): مقایسه‌ی نتایج الگوریتم پیشنهادی با الگوریتم‌های LPA، NIB-LPA و LPA-Intimacy بر روی دیتاست‌های واقعی بر اساس معیار F-measure

دیتاست	LPA	NIBLPA	LPA-Intimacy	الگوریتم پیشنهادی
Karate	۰,۹۶	۰,۳۹	1	۱
Dolphins	۰,۳۰	۰,۳۸	۰,۴۱	۰,۴۷
Polbooks	۰,۴۵	۰,۵۳	۰,۶۲	۰,۷۳
Football	۰,۶	۰,۴۷	۰,۶۲	۰,۴۸

در هر تکرار را داشت. همچنین الگوریتم NIB-LPA نیز در اجراهای مختلف تغییراتی در نتایج داشت ولی ناپایداری آن نسبت به LPA کمتر بود.

جدول (۵) نشان دهنده‌ی مقادیر F-measure به دست آمده از انجام آزمایش‌ها توسط ۴ الگوریتم بر روی چهار دیتاست گراندتروس‌دار است.

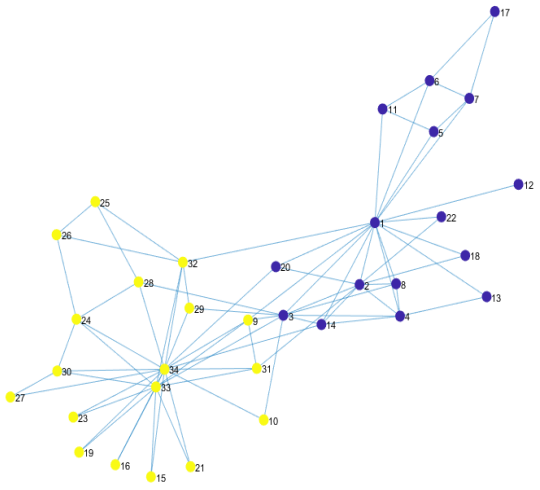
شکل (۶) مقایسه‌ی بین مقادیر F-measure را به صورت واضح‌تری نشان می‌دهد.



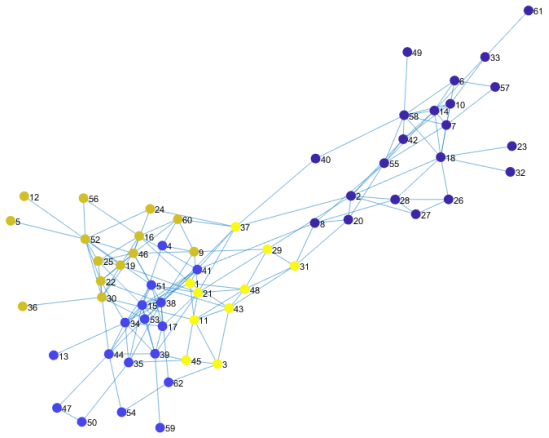
(شکل-۶): مقایسه‌ی نتایج الگوریتم پیشنهادی با الگوریتم LPA، NIB-LPA و LPA-Intimacy بر روی دیتاست‌های گراندتروس‌دار با استفاده از معیار F-measure

طبق جدول (۵)، الگوریتم پیشنهادی در دیتاست‌های Karate، Dolphins و Polbooks بالاترین مقدار F-measure را کسب کرده است. در دیتاست Football، الگوریتم LPA-Intimacy بهترین مقدار F-measure را به دست آورده است و پس از آن، الگوریتم LPA بیشترین مقدار NMI را دارد و الگوریتم پیشنهادی سومین مقدار F-measure را کسب کرده است. الگوریتم NIB-LPA در هیچکدام از دیتاست‌ها نتوانسته است مقداری بهتر از الگوریتم پیشنهادی کسب کند، اما در دیتاست‌های Dolphins و Polbooks توانسته است مقدار بیشتری از LPA به دست آورد. الگوریتم LPA-Intimacy نیز در تمامی موارد مقدار بهتری نسبت به الگوریتم LPA و NIB-LPA کسب کرده است.

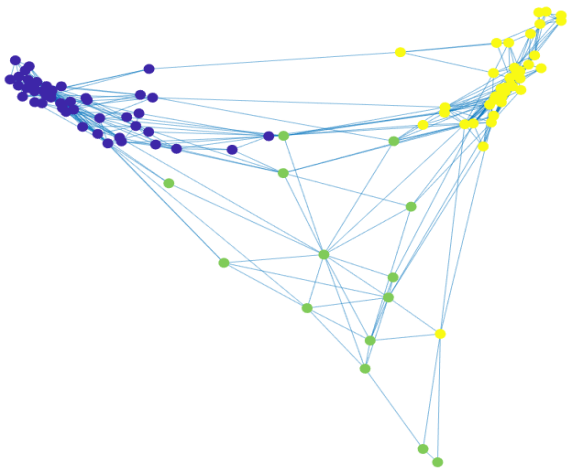
شکل‌های (۷) تا (۱۰) جوامع کشف شده توسط الگوریتم پیشنهادی در دیتاست‌های Karate، Dolphins، Polbooks و Football را نشان می‌دهد.



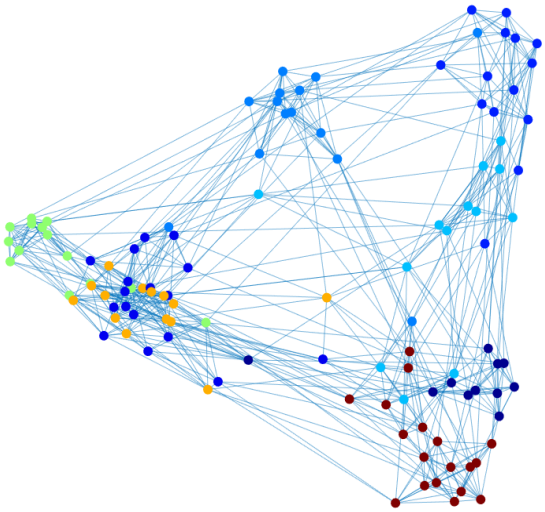
(شکل-۷): جوامع کشف شده توسط الگوریتم پیشنهادی در دیتاست Karate



(شکل-۸): جوامع کشف شده توسط الگوریتم پیشنهادی در دیتاست Dolphins



(شکل-۹): جوامع کشف شده توسط الگوریتم پیشنهادی در دیتاست Polbooks



(شکل-۱۰): جوامع کشف شده توسط الگوریتم پیشنهادی در دیتاست Football

جدول (۶) نشان دهنده‌ی نتایج ماژولاریتی به دست آمده از اجرای الگوریتم پیشنهادی، الگوریتم LPA، NIB-LPA و LPA-Intimacy بر روی ۱۳ دیتاست جهان واقعی است. بالا بودن مقدار ماژولاریتی به دست آمده نشان دهنده‌ی افراز خوب و با کیفیت توسط الگوریتم است. در دیتاست‌هایی که تعداد جوامع از پیش مشخص شده‌ای ندارند از معیار ماژولاریتی برای ارزیابی کیفیت جوامع کشف شده استفاده می‌کنیم.

در دیتاست‌های Karate، Dolphins، Polbooks و Football که گراندتروس دارند عملا مقایسه‌ی الگوریتم‌ها با استفاده از معیار ماژولاریتی بی‌معنی است، چون در این نوع دیتاست‌ها مقایسه‌ی اصلی بر اساس معیار NMI و F-measure است که در جدول‌های (۴) و (۵) به مقایسه‌ی آنها پرداختیم. ممکن است ماژولاریتی طبیعی یک دیتاست گراندتروس‌دار پایین باشد و پیدا کردن ماژولاریتی بالا بی فایده باشد، چون ما به دنبال یافتن ساختار واقعی جوامع هستیم تا اینکه ماژولاریتی بالایی کسب کنیم. به عنوان مثال در دیتاست Karate الگوریتم پیشنهادی بالاترین مقدار NMI را دارد که برابر ۱ است و ماژولاریتی واقعی این دیتاست ۰,۳۷۱۵ است که الگوریتم پیشنهادی آن را به درستی به دست آورده است، هرچند که مقدار ماژولاریتی الگوریتم پیشنهادی کمتر از LPA و NIB-LPA است. در مواقعی که گراندتروس وجود نداشته باشد، از معیار ماژولاریتی برای مقایسه‌ی کیفیت جوامع استفاده می‌کنیم. هر اندازه این معیار بالا باشد، جوامع منسجم‌تری یافت شده است. در دیتاست Email، الگوریتم پیشنهادی بالاترین مقدار ماژولاریتی را کسب کرده است و پس از الگوریتم پیشنهادی، الگوریتم NIB-LPA و LPA-Intimacy دومین مقادیر بالای ماژولاریتی را دارد. در دیتاست Netscience، NIB-LPA و LPA-Intimacy با ماژولاریتی برابر ۰,۸۹، بالاترین مقادیر را دارند و LPA با مقدار ۰,۸۸ دومین مقدار ماژولاریتی را کسب کرده است. الگوریتم پیشنهادی در دیتاست‌های Power و CA-GRQC بیشترین مقدار ماژولاریتی را دارد. در دیتاست Collaboration، الگوریتم پیشنهادی و LPA هر دو نتیجه یکسان و بیشتر از NIB-LPA دارند و الگوریتم LPA-Intimacy پس از الگوریتم پیشنهادی و LPA، دومین مقدار ماژولاریتی را دارد. در دیتاست PGP الگوریتم پیشنهادی با ۰,۰۲ اختلاف از LPA، دومین مقدار ماژولاریتی را دارد و مقادیری بیشتر از NIB-LPA و LPA-Intimacy به دست آورده است. در دیتاست Condmat-2003 الگوریتم LPA-Intimacy و الگوریتم پیشنهادی به ترتیب با مقادیر ۰,۶۷ و ۰,۶۵ بیشترین مقدار ماژولاریتی بین سایر

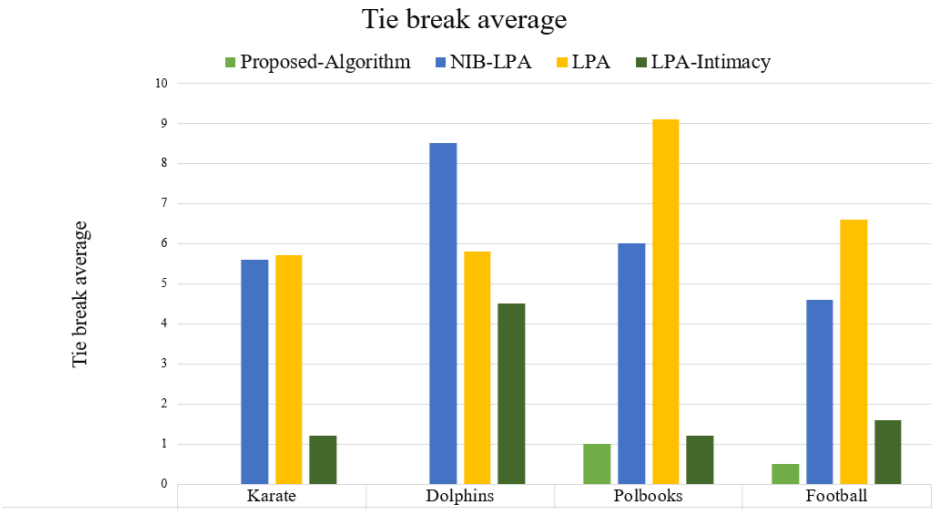
الگوریتم‌ها را دارند. در دیتاست Email-Enron، الگوریتم پیشنهادی با اختلاف قابل ملاحظه‌ای بیشترین ماژولاریتی را دارد. حتی الگوریتم دقیقی مانند LPA-Intimacy در این دیتاست مقدار پایینی به دست آورده است، در حالی که الگوریتم پیشنهادی با مقدار ماژولاریتی ۰,۴۷ توانسته است افراز خوبی در این دیتاست داشته باشد. در دیتاست Condmat-2005 بیشترین مقدار ماژولاریتی متعلق به الگوریتم LPA-Intimacy است و پس از آن الگوریتم پیشنهادی ماژولاریتی بالایی نسبت به دو الگوریتم LPA و NIB-LPA دارد. نکته‌ی قابل توجه اینکه در دیتاست Condmat-2005 که بزرگ‌ترین دیتاست مورد آزمایش است، الگوریتم پیشنهادی ماژولاریتی ۰,۵۹ به دست آورده است در حالی که الگوریتم NIB-LPA مقدار ماژولاریتی ۰,۲۵ دارد، که در مقایسه باهم، الگوریتم NIB-LPA نتیجه‌ی بسیار کمی به دست آورده است. در دو دیتاست Condmat-2003 و Condmat-2005 الگوریتم پیشنهادی دومین مقدار ماژولاریتی پس از الگوریتم LPA-Intimacy را دارد اما باید توجه داشت که تنها ۰,۰۲ اختلاف بین آنها وجود دارد و الگوریتم پیشنهادی توانسته است به خوبی با الگوریتم دقیقی مانند LPA-Intimacy رقابت کند و نتایج بهتر و یا نزدیک به آن کسب کند. در حالت کلی، در بین ۱۳ دیتاست مورد آزمایش، الگوریتم پیشنهادی در ۸ مورد ماژولاریتی بهتری دارد که ۳ مورد آن مربوط به دیتاست‌های گراندتروس‌دار است و با اینکه در دیتاست‌های Karate، Dolphins و Polbooks ماژولاریتی پایینی دارد، اما با توجه به مقدار NMI به دست آمده، متوجه می‌شویم این مقدار ماژولاریتی صحیح و بیشتر نزدیک به ماژولاریتی واقعی دیتاست‌ها است. الگوریتم NIB-LPA نیز در بین ۱۳ دیتاست فقط در ۱ مورد، الگوریتم LPA در ۲ مورد و الگوریتم LPA-Intimacy در ۵ مورد ماژولاریتی بیشتر کسب کرده اند.

برای مقایسه‌ی تعداد شکست تساوی سه الگوریتم، آزمایش دیگری بر روی ۱۳ دیتاست واقعی انجام شد و تعداد شکست تساوی الگوریتم‌های مورد آزمایش مورد مقایسه و ارزیابی قرار گرفت. در الگوریتم LPA زمانی که چندین برچسب جامعه فراوانی یکسانی داشته باشند، شکست تساوی انجام می‌شود. در الگوریتم NIB-LPA نیز در ۲ قسمت شکست تساوی رخ می‌دهد، زمانی که چندین برچسب با فراوانی یکسان وجود دارد و زمانی که چندین جامعه با مقدار مجموع K-shell برابر وجود داشته باشد. در الگوریتم LPA-Intimacy نیز زمانی که مجموع تاثیر چندین جامعه برابر باشد و زمانی که چندین جامعه با مقدار tightness برابر وجود داشته باشد شکست تساوی انجام می‌شود. در الگوریتم پیشنهادی نیز زمانی که چندین برچسب فراوانی یکسان داشته باشند، زمانی که چندین برچسب مقدار تاثیر یکسانی داشته باشند و زمانی که مجموع اهمیت گره‌های جوامع مختلف، یکسان باشد، باید شکست تساوی انجام شود و سپس موثرترین برچسب انتخاب شود. با توجه به اینکه در الگوریتم پیشنهادی مراحل شکست تساوی بیشتر از سه الگوریتم دیگر است، اما نتایج نشان می‌دهد که الگوریتم پیشنهادی تعداد شکست تساوی کمتری نسبت به الگوریتم LPA، NIB-LPA و LPA-Intimacy دارد. میانگین تعداد شکست تساوی‌ها در ۱۰

تکرار، برای هر ۳ الگوریتم در دیتاست‌های گراندتروس‌دار در شکل (۱۱) و برای دیتاست‌های بدون گراندتروس، در شکل (۱۲) نشان داده شده است. همچنین نتایج تعداد شکست به دست آمده در هر تکرار از الگوریتم برای هر سه روش، در۴ دیتاست گراندتروس‌دار و دو دیتاست بزرگ Email-Enron و Condmat_2005، در شکل (۱۳) تا شکل (۱۸) نمایش داده شده است.

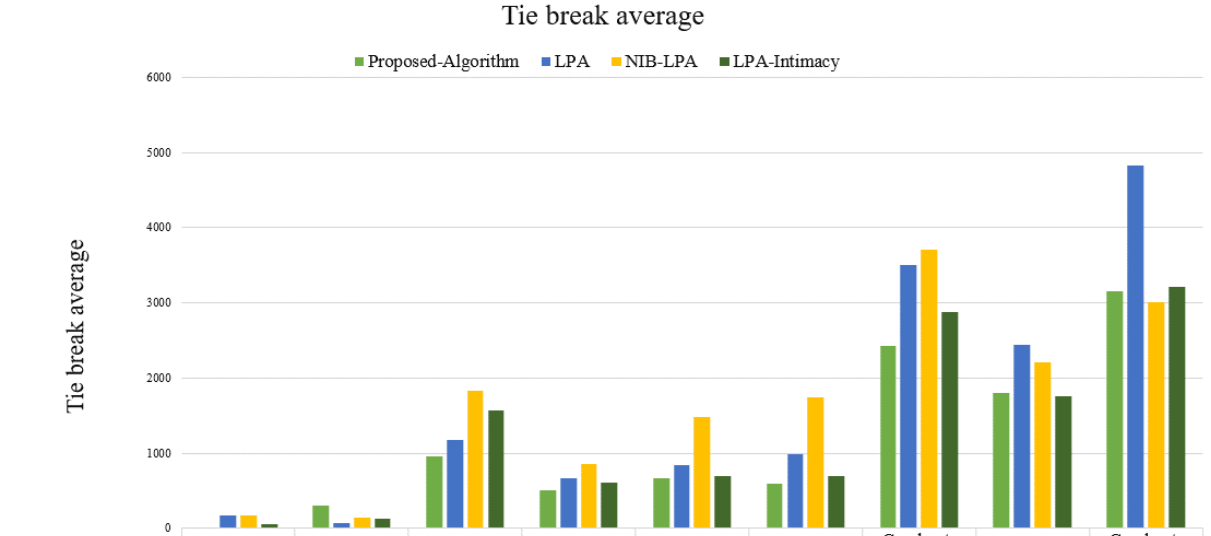
(جدول-۶): ارزیابی الگوریتم پیشنهادی بر روی دیتاست‌های واقعی بر اساس معیار ماژولاریتی (Q)

	Q				CN			
دیتاست	LPA	NIBLPA	LPA-Intimacy	الگوریتم پیشنهادی	LPA	NIBLPA	LPA-Intimacy	الگوریتم پیشنهادی
Karate	۰,۳۸	۰,۴۰	۰,۳۷۱۵	۰,۳۷۱۵	۲	۵	۲	۲
Dolphins	۰,۵۰	۰,۴۳	۰,۵۰	۰,۵۱۴۸	۴	۵	۴	۴
Polbooks	۰,۴۷	۰,۵۵	۰,۴۸	۰,۴۹	۳	۴	۳	۳
Football	۰,۵۹	۰,۵۰	۰,۶۰	۰,۵۲	۹	۸	۱۰	۸
Email	۰,۳۱	۰,۴۲	۰,۴۲	۰,۴۶	۳۶	۱۳۹	۱۸۲	۴۱
Netscience	۰,۸۸	۰,۸۹	۰,۸۹	۰,۸۶	۳۲۶	۳۶۴	۳۲۹	۴۴۲
Power	۰,۷۱	۰,۶۲	۰,۷۲	۰,۷۵	۵۷۹	۱۳۴۱	۴۹۳	۳۴۹
CA-GRQC	۰,۷۲	۰,۷۴	۰,۷۴	۰,۷۸	۷۳۱	۹۱۴	۶۳۹	۷۰۵
Collaboration	۰,۷۴	۰,۷۱	۰,۷۲	۰,۷۴	۱۸۶۳	۲۳۶۲	۱۸۴۳	۲۰۳۸
PGP	۰,۸۰	۰,۷۶	۰,۷۶	۰,۷۸	۹۷۵	۱۸۷۰	۶۲۹	۲۷۴
Condmat-2003	۰,۵۸	۰,۵۲	۰,۶۷	۰,۶۵	۳۵۶۷	۳۶۱۹	۲۶۹۱	۲۷۶۹
Email-Enron	۰,۲۶	۰,۱۴	۰,۱۶	۰,۴۷	۱۹۳۴	۲۱۴۳	۱۴۵۳	۲۰۶۸
Condmat-2005	۰,۵۲	۰,۲۵	۰۶۱	۰,۵۹	۴۴۲۶	۳۵۴۸	۲۹۱۱	۳۰۷۸



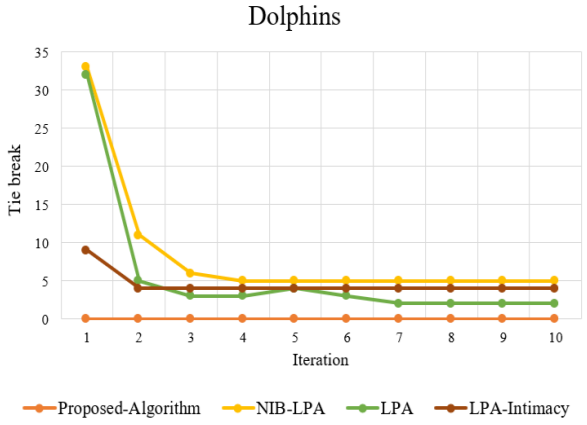
	Karate	Dolphins	Polbooks	Football
Proposed-Algorithm	0	0	1	0.5
NIB-LPA	5.6	8.5	6	4.6
LPA	5.7	5.8	9.1	6.6
LPA-Intimacy	1.2	4.5	1.2	1.6

(شکل-۱۱): مقایسه میانگین تعداد شکست تساوی برای الگوریتم‌های LPA، NIB-LPA، LPA-Intimacy و الگوریتم پیشنهادی در دیتاست‌های گراندتروس‌دار- در قسمت‌هایی که نمودار الگوریتم پیشنهادی(نمودار سبز کم رنگ) مشاهده نمی‌شود، مقدار میانگین صفر دارد.

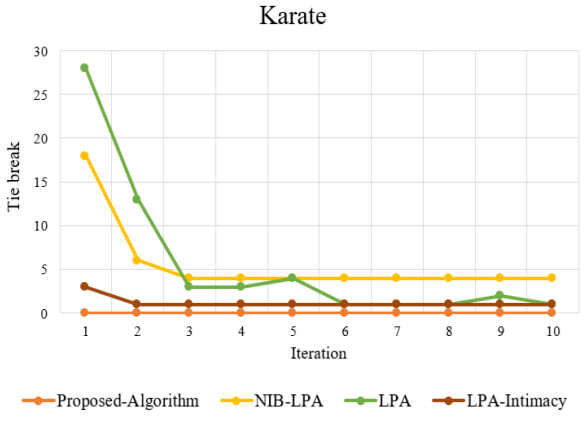


	Email	Netscience	Power	CA-GRQC	Collaboration	PGP	Condmat-2003	Email-Enron	Condmat-2005
Proposed-Algorithm	17	310	965	505	674	592	2422.0	1803.0	3158
LPA	172	78	1183	663	848	993	3507	2440	4819
NIB-LPA	174	148	1833	862	1480	1744	3703	2210	3009
LPA-Intimacy	51	124	1567	612	691	696	2883	1762	3213

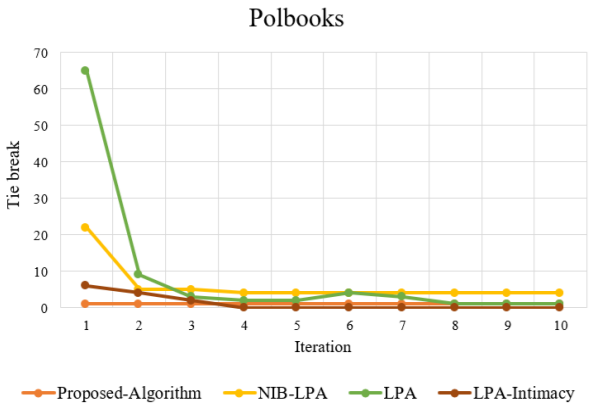
(شکل-۱۲): مقایسه میانگین تعداد شکست تساوی برای الگوریتم‌های LPA، NIBLPA، LPA-Intimacy و الگوریتم پیشنهادی در دیتاست‌های بدون گراندتروس



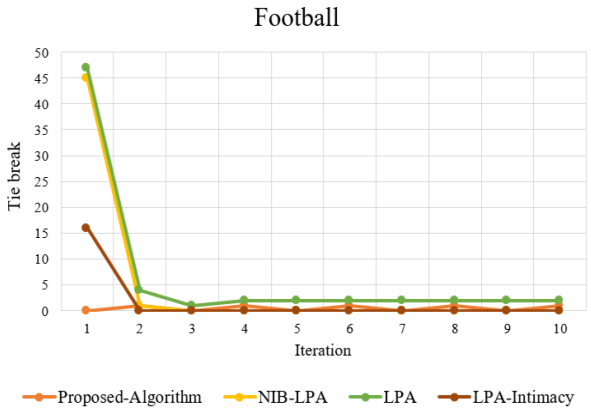
(شکل-۱۴): مقایسه تعداد شکست تساوی ۴ الگوریتم LPA، NIB-LPA، LPA-Intimacy و الگوریتم پیشنهادی در هر تکرار الگوریتم در دیتاست Dolphins



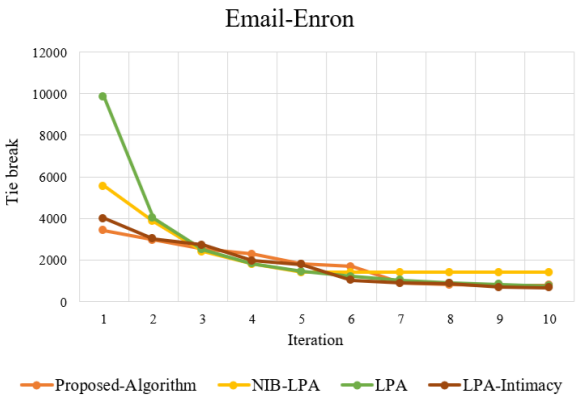
(شکل-۱۳): مقایسه تعداد شکست تساوی ۴ الگوریتم LPA، NIB-LPA، LPA-Intimacy و الگوریتم پیشنهادی در هر تکرار الگوریتم در دیتاست Karate



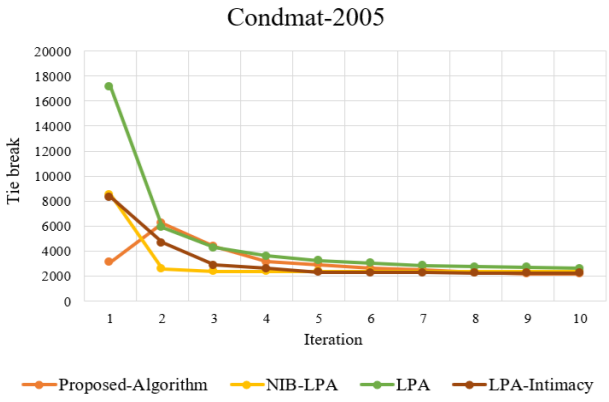
(شکل-۱۵): مقایسه تعداد شکست تساوی ۴ الگوریتم LPA، NIB-LPA، LPA-Intimacy و الگوریتم پیشنهادی در هر تکرار الگوریتم در دیتاست Polbooks



(شکل-۱۶): مقایسه تعداد شکست تساوی ۴ الگوریتم LPA، NIB-LPA، LPA-Intimacy و الگوریتم پیشنهادی در هر تکرار الگوریتم در دیتاست Football



(شکل-۱۷): مقایسه تعداد شکست تساوی ۴ الگوریتم LPA، NIB-LPA، LPA-Intimacy و الگوریتم پیشنهادی در هر تکرار الگوریتم در دیتاست Email-Enron



(شکل-۱۸): مقایسه تعداد شکست تساوی ۴ الگوریتم LPA، NIB-LPA، LPA-Intimacy و الگوریتم پیشنهادی در هر تکرار الگوریتم در دیتاست Condmat-2005

طبق شکل (۱۱) الگوریتم پیشنهادی در دیتاست‌های Karate و Dolphins، در تمامی ۱۰ تکرار هیچگونه مرحله‌ی شکست تساوی انجام نداده است و میانگین کل آن در ۱۰ تکرار برابر صفر است در حالی که الگوریتم‌های NIB-LPA، LPA و LPA-Intimacy به ترتیب میانگین شکست تساوی ۵,۶، ۵,۷ و ۱,۲ در دیتاست Karate دارند. وجود میانگین تعداد شکست تساوی برابر صفر بیانگر دقت الگوریتم پیشنهادی در انتخاب و انتشار درست برچسب است که توانسته است برچسب‌ها را طوری پخش کند که در انتخاب برچسب مناسب، چندین برچسب با فراوانی یا تاثیر یکسان وجود نداشته باشند. کاهش تعداد شکست تساوی، سبب افزایش سرعت الگوریتم می‌شود، زیرا از بررسی‌های اضافی جلوگیری می‌کند و از طرفی پایداری نتایج الگوریتم را افزایش می‌دهد و در اجراهای مختلف، جواب‌های یکسان حاصل می‌شود. در دیتاست‌های Polbooks و Football الگوریتم پیشنهادی به ترتیب میانگین ۱ و ۰,۵ دارد که بهترین نتیجه در مقایسه با سه روش دیگر است. در دیتاست Football الگوریتم پیشنهادی به صورت نوسانی شکست تساوی برابر ۰ و ۱ دارد به همین دلیل مقدار میانگین کل برابر ۰,۵ است. در دیتاست‌های Polbooks و Football، LPA بیشترین مقدار شکست تساوی را دارد. با مقایسه نتایج در شکل (۱۲) می‌توان مشاهده کرد که به غیر از دیتاست Netscience، Email-Enron و Condmat-2005، الگوریتم پیشنهادی در سایر دیتاست‌ها کمترین میانگین تعداد شکست تساوی را دارد. در این ۳ دیتاست ذکر شده، الگوریتم پیشنهادی در دیتاست Netscience بیشترین مقدار شکست تساوی و در دیتاست Email-Enron پس از LPA-Intimacy و دیتاست Condmat-2005 پس از NIB-LPA کمترین مقدار شکست تساوی را دارد.

برای مقایسه‌ی سرعت همگرایی و زمان اجرای الگوریتم‌ها، آزمایش دیگری انجام شد که در آن تمامی الگوریتم‌های مورد آزمایش به تعداد ۱۰ تکرار اجرا شدند و شرطی اتخاذ شد که اگر الگوریتم در ۲ تکرار متوالی نتایج یکسانی داشته باشد (یعنی برچسب گره‌ها تغییر نکند) الگوریتم پایان می‌پذیرد. زمانی که الگوریتم به پایان می‌رسد، به عنوان زمان اجرای الگوریتم در نظر گرفته می‌شود چراکه ممکن است الگوریتمی در ۴ تکرار پایان پذیرد و برخی ممکن است در ۱۰ تکرار هنوز به اتمام نرسند. جدول (۷) نشان دهنده‌ی شماره تکراری است که الگوریتم به همگرایی رسیده است و جدول (۸) زمان اجرای الگوریتم‌ها را نشان می‌دهد.

(جدول-۷): مقایسه سرعت همگرایی الگوریتم‌ها بر روی ۱۳ دیتاست مورد آزمایش- اعداد موجود در جدول شماره‌ی تکراری را که در آن الگوریتم همگرا شده است را نشان می‌دهد

دیتاست	الگوریتم پیشنهادی	LPA	NIBLPA	LPA-Intimacy
Karate	۳	۷	۶	۴
Dolphins	۴	۱۰	۶	۴
Polbooks	۴	۹	۵	۴
Football	۵	۱۰	۴	۵
Email	۴	۱۰	۹	۱۰
Netscience	۴	۱۰	۵	۹
Power	۱۰	۱۰	۱۰	۱۰

CA-GRQC	۱۰	۱۰	۱۰	۱۰
Collaboration	۱۰	۱۰	۱۰	۱۰
PGP	۱۰	۱۰	۱۰	۱۰
Condmat-2003	۱۰	۱۰	۱۰	۱۰
Email-Enron	۱۰	۱۰	۱۰	۱۰
Condmat-2005	۱۰	۱۰	۱۰	۱۰

(جدول-۸): مقایسه‌ی زمان اجرای الگوریتم‌های مورد آزمایش بر روی ۱۳ دیتاست مختلف. زمان‌ها بر اساس ثانیه هستند.

دیتاست	الگوریتم پیشنهادی	LPA	NIBLPA	LPA-Intimacy
Karate	۰,۱۵	۰,۹	۰,۱۵	۰,۱۷
Dolphins	۰,۱۳	۰,۱۴	۰,۱۵	۰,۲
Polbooks	۰,۱۹	۰,۱۴	۰,۱۶	۰,۲۵
Football	۰,۲	۰,۱۵	۰,۱۴	۰,۲۷
Email	۱,۵	۱,۹	۱,۹	۲,۶
Netscience	۱,۷	۱,۶	۱,۳	۱,۸
Power	۷	۶	۷	۷,۶
CA-GRQC	۱۰	۹	۹	۱۲
Collaboration	۱۱,۶	۱۳	۱۴	۱۶
PGP	۱۹	۲۱	۲۲	۲۵
Condmat-2003	۸۸	۱۸۶	۱۹۳	۲۲۰
Email-Enron	۱۲۹	۳۲۱	۳۱۴	۲۸۸
Condmat-2005	۱۳۰	۳۵۰	۳۴۹	۳۸۰

همانطور که از جدول (۷) مشخص است الگوریتم پیشنهادی در حالت کلی پایین‌ترین تعداد تکرار را برای رسیدن به یک نتیجه پایدار داشته است. به جز دیتاست Football که در آن الگوریتم NIB-LPA کمترین تعداد تکرار برای رسیدن به همگرایی را دارد، در بقیه دیتاست‌ها الگوریتم پیشنهادی بهترین عملکرد را دارد.

از دیتاست Power تا دیتاست Condmat-2005 تمامی الگوریتم‌ها ۱۰ تکرار را تکمیل کرده‌اند و نتوانسته‌اند قبل از ۱۰ تکرار همگرا شوند. پس مقایسه‌ی اصلی از دیتاست Karate تا Netscience خواهد بود که الگوریتم پیشنهادی توانسته است سریع‌تر از بقیه همگرا شود. در دیتاست‌هایی که در ۱۰ تکرار الگوریتم‌ها نتوانسته‌اند همگرا شوند، بهتر است آنها را از لحاظ زمان اجرا مقایسه کرد. جدول (۸) زمان اجرای الگوریتم‌ها را نشان می‌دهد. در دیتاست Karate تا دیتاست CA-GRQC تقریبا الگوریتم‌ها زمان اجرای نزدیک به هم دارند و اختلاف آنها در حدود چند صدم ثانیه است. در دیتاست Collaboration و PGP الگوریتم پیشنهادی کمترین زمان اجرا را دارد. تفاوت اصلی بین عملکرد و سرعت الگوریتم‌ها در دیتاست‌های Condmat-2003، Email-Enron و Condmat-2005 ایجاد می‌شود. کاملا واضح است که الگوریتم پیشنهادی در این ۳ دیتاست بزرگ عملکرد بسیار خوبی داشته است و تفاوت بسیار زیادی با بقیه روش‌ها در زمان اجرا دارد. به عنوان مثال در دیتاست Condmat-2003 الگوریتم پیشنهادی در ۸۸ ثانیه ۱۰ تکرار را انجام داده است اما الگوریتم LPA-Intimacy در ۲۲۰ ثانیه توانسته ۱۰ تکرار انجام دهد که ۱۳۲ ثانیه تفاوت بین زمان اجرای LPA-Intimacy و الگوریتم پیشنهادی وجود دارد. همچنین الگوریتم LPA که به دلیل پیچیدگی محاسباتی پایین معروف است، در این ۳ دیتاست زمان اجرای بیشتری از الگوریتم پیشنهادی دارد و حتی در دیتاست Condmat-2005 در حالی که الگوریتم پیشنهادی آن را در ۱۳۰ ثانیه اجرا کرده است، الگوریتم

LPA با ۲۲۰ ثانیه اختلاف، در ۳۵۰ ثانیه توانسته است به پایان برساند و این کاملا نشان‌دهنده این است که الگوریتم پیشنهادی عملکرد و سرعت بالایی نسبت به تمام الگوریتم‌های مورد آزمایش دارد. الگوریتم LPA-Intimacy هر چند که در آزمایش‌های انجام شده در برخی موارد نتایج نزدیک به الگوریتم پیشنهادی دارد، اما زمان اجرای آن مخصوصا در دیتاست‌های بزرگ خیلی بیشتر از الگوریتم پیشنهادی است.

۵– نتیجه‌گیری و پژوهش‌های آتی

در این مقاله یک روش ترکیبی مبتنی بر پخش برچسب گره هسته و انتشار برچسب ارائه شد که با در نظر گرفتن اهمیت گره‌ها و تاثیر متفاوت برچسب جوامع مختلف و همچنین با بهره‌گیری از تشابه‌های بین گره‌ها و تغییر نحوه‌ی به‌روز رسانی برچسب گره‌ها، نتایج بهتری نسبت به سه الگوریتم LPA، NIB-LPA و LPA-Intimacy به دست آورد. روش پیشنهادی با در نظر گرفتن دو معیار برای محاسبه‌ی امتیاز هسته بودن یک گره و پخش برچسب گره هسته به ۲ سطح از همسایه‌ها، سبب شد که دقت الگوریتم نسبت به دو الگوریتم دیگری که مورد مقایسه هستند بیشتر باشد و همچنین تعداد شکست تساوی به طور قابل ملاحظه‌ای کاهش یابد. همچنین تغییر نحوه‌ی انتخاب برچسب و توجه به تشابه‌های محلی گره‌ها و انتخاب موثر برچسب زمانی که چندین برچسب با فراوانی یکسان وجود دارد، عملکرد الگوریتم LPA را بهبود بخشید. آزمایش‌هایی بر روی دیتاست‌های واقعی و مصنوعی انجام شد و عملکرد الگوریتم پیشنهادی در مقایسه با الگوریتم LPA، NIB-LPA و LPA-Intimacy مورد ارزیابی قرار گرفت. با ارزیابی نتایج، مشاهده شد که الگوریتم پیشنهادی عملکرد، سرعت اجرا، همگرایی سریع و نتایج بهتری نسبت به الگوریتم LPA، NIB-LPA و LPA-Intimacy دارد و از طرفی الگوریتم پیشنهادی از هیچ پارامتر قابل تنظیمی استفاده نمی‌کند. با توجه به ساختار روش پیشنهادی، این روش پایداری بیشتری نسبت به سه الگوریتم دیگر دارد و اغلب در اجراهای متفاوت نتایج یکسانی ارائه می‌دهد. با ارائه‌ی معیارهای دقیق‌تر برای انتخاب گره‌های هسته و روش‌های جدیدتر برای به‌روز رسانی برچسب گره‌ها، می‌توان نتایج بهتری حاصل نمود.

۶ – مراجع

[1] Raghavan, Usha Nandini, Réka Albert, and Soundar Kumara. "Near linear time algorithm to detect community structures in large-scale networks." Physical review E 76, no. 3 (2007): 036106.

[2] Saoud, Bilal, and Abdelouahab Moussaoui. "Node similarity and modularity for finding communities in networks." Physica A: Statistical Mechanics and its Applications 492 (2018): 1958-1966.

[3] Fortunato, Santo. "Community detection in graphs." Physics reports 486, no. 3-5 (2010): 75-174.

[4] Zhang, Peng, Dan Qiu, An Zeng, and Jinghua Xiao. "A comprehensive comparison of network similarities for link prediction and spurious link elimination." Physica A: Statistical

Mechanics and its Applications 500 (2018): 97-105.

[5] Zhang, Peng, Dan Qiu, An Zeng, and Jinghua Xiao. "A comprehensive comparison of network similarities for link prediction and spurious link elimination." Physica A: Statistical Mechanics and its Applications 500 (2018): 97-105.

[6] Barnes, Earl R. "An algorithm for partitioning the nodes of a graph." SIAM Journal on Algebraic Discrete Methods 3, no. 4 (1982): 541-550.

[7] Kernighan, Brian W., and Shen Lin. "An efficient heuristic procedure for partitioning graphs." The Bell system technical journal 49, no. 2 (1970): 291-307.

[8] Newman, Mark EJ. "Fast algorithm for detecting community structure in networks." Physical review E 69, no. 6 (2004): 066133.

[9] Blondel, Vincent D., Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. "Fast unfolding of communities in large networks." Journal of statistical mechanics: theory and experiment 2008, no. 10 (2008): P10008.

[10] Newman, Mark EJ, and Michelle Girvan. "Finding and evaluating community structure in networks." Physical review E 69, no. 2 (2004): 026113.

[11] Fortunato, Santo, Vito Latora, and Massimo Marchiori. "Method to find community structures based on information centrality." Physical review E 70, no. 5 (2004): 056104.

[12] Xi, Jingke, Wenwei Zhan, and Zhixiao Wang. "Hierarchical community detection algorithm based on node similarity." Int J Database Theory Appl 9, no. 6 (2016): 209-218.

[13] Zhang, Xian-Kun, Jing Ren, Chen Song, Jia Jia, and Qian Zhang. "Label propagation algorithm for community detection based on node importance and label influence." Physics Letters A 381, no. 33 (2017): 2691-2698.

[14] Kong, Hanzhang, Qinma Kang, Chao Liu, Wenquan Li, Hong He, and Yunfan Kang. "An improved label propagation algorithm based on node intimacy for community detection in networks." International Journal of Modern Physics B 32, no. 25 (2018): 1850279.

[15] Lin, Zhen, Xiaolin Zheng, Nan Xin, and Deren Chen. "CK-LPA: Efficient community detection algorithm based on label propagation with community kernel." Physica A: Statistical Mechanics and its Applications 416 (2014): 386-399.

[16] Gui, Chun, Ruisheng Zhang, Zhili Zhao, Jiaxuan Wei, and Rongjing Hu. "LPA-CBD an improved label propagation algorithm based on community belonging degree for community detection." International Journal of Modern Physics C 29, no. 02 (2018): 1850011.

[17] Jaccard, Paul. "Étude comparative de la distribution florale

dans une portion des Alpes et des Jura." Bull Soc Vaudoise Sci Nat 37 (1901): 547-579.

[18] Lancichinetti, Andrea, Santo Fortunato, and Filippo Radicchi. "Benchmark graphs for testing community detection algorithms." Physical review E 78, no. 4 (2008): 046110.

[19] Danon, Leon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. "Comparing community structure identification." Journal of Statistical Mechanics: Theory and Experiment 2005, no. 09 (2005): P09008.

[20] Chinchor, N. "MUC-4 Evaluation Metrics (1992)." Proceedings of the Fourth Message Understanding Conferene: 22-29.

[21] Sammut, Claude, and Geoffrey I. Webb, eds. Encyclopedia of machine learning. Springer Science & Business Media, 2011.

حمید روغنی مدرک کارشناسی خود را در رشته‌ی مهندسی فناوری اطلاعات در سال ۱۳۹۶ از دانشگاه شهید مدنی آذربایجان اخذ کرده است. وی در حال حاضر در مقطع کارشناسی ارشد رشته‌ی مهندسی فناوری اطلاعات در دانشگاه شهید مدنی آذربایجان مشغول به تحصیل است. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارت‌اند از: تحلیل شبکه‌های اجتماعی و پیچیده، پردازش موازی و توزیع شده و تحلیل داده‌های بزرگ است.

عسگرعلی بویر دوره دکترای مهندسی کامپیوتر را در دانشگاه صنعتی مالزی (UTM) در سال ۱۳۹۰ به اتمام رسانده و موفق به اخذ درجه دکترا در رشته کامپیوتر از دانشگاه مذکور گردید. و هم اکنون دانشیار دانشکده فناوری اطلاعات و مهندسی کامپیوتر دانشگاه شهید مدنی آذرربایجان می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: تحلیل شبکه های اجتماعی پیچیده، رایانش ابری گرید و داده کاوی می باشد.