نشریه علمی "پدافند الکترونیکی و سایبری"

سال هشتم، شماره ۴، زمستان ۱۳۹۹، ص ۱۵-۱

علمی– پژوهشی

روش توزیعی تشخیص انجمن در شبکه‌های اجتماعی بزرگ بر اساس انتشار برچسب

محمد حسینی[۱]، امین‌اله مه‌آبادی[۲*]

۱- کارشناس‌ارشد، گروه مهندسی کامپیوتر، دانشگاه شاهد، ۲- استادیار، گروه مهندسی کامپیوتر، دانشگاه شاهد

(دریافت: ۱۳۹۸/۰۳/۱۳، پذیرش: ۱۳۹۹/۰۵/۱۵)

چکیده

تشخیص انجمن‌های هم‌پوشان در شبکه‌های اجتماعی بسیار بزرگ با عامل‌های هوشمند یک مساله سخت و مهم است که قدرت تشخیص و تحلیل آن شبکه‌ها را از حالت بی‌درنگِ برخط خارج می‌کند. همپوشانی انجمن‌ها در کنار افزایش ابعاد و ارتباطات این شبکه‌ها به چالش‌های پیچیدگی زمان زیاد جستجوی انجمن‌ها و افزایش طاقت‌فرسای حافظه مصرفی منجر می‌شود که از قابلیت کنترل سریع آن‌ها می‌کاهد. ارائه روش‌های توزیعی مقیاس‌پذیر تصادفی و عامل‌گرا، بر اساس انتشار برچسب در شبکه‌های بسیار بزرگ و پیچیده به کاهش زمان جستجو و تسریع تشخیص کمک می‌کند. این مقاله روش توزیعی نوین مقیاس‌پذیر عامل‌گرا برای تشخیص انجمن‌های هم‌پوشان بر اساس انتشار برچسب توانسته با محدودسازی انتشار پیام و استفاده از معیارهای جدید بر روی معماری چندهسته‌ای، به پیچیدگی خطی زمان اجرا و حافظه مصرفی دست یابد. روش پیشنهادی با آزمون بر روی مجموعه داده‌های بسیار بزرگ شبکه‌های اجتماعی، از نظر زمان اجرا در شبکه‌های بزرگ تا ۹ برابر تسریع و از نظر پیمانه‌ای از ٪۳ تا ٪۱۰۰ بهبود دارد و در یافتن انجمن‌های هم‌پوشان بسیار دقیق و سریع عمل می‌کند.

کلیدواژه‌ها: شبکه‌های اجتماعی، پردازش توزیعی، تشخیص انجمن‌های همپوشان، الگوریتم انتشار برچسب

۱- مقدمه

شبکه‌های اجتماعی بعد از ظهور شبکه‌های واقعی مانند Facebook و Twitter گسترش بسیار یافتند و وارد کاربردهای واقعی زندگی بشر شدند. دو خاصیت مهم شبکه‌های اجتماعی، موجودیت‌ها[۱] و رابطه بین آن‌ها است. موجودیت‌ها می‌توانند افراد و رابطه بین آن‌ها می‌تواند دوستی باشد. مجموعه‌ای از موجودیت‌های شبکه اجتماعی که شدت تعاملات در آن بیشتر از دیگر موجودیت‌های پراکنده باشد انجمن تشکیل می‌دهند. بعضی از موجودیت‌ها ممکن است عضو چندین انجمن باشند. لذا انجمن‌ها مستقل یا بدون عضو مشترک و همپوشان یا دارای عضو مشترک هستند. تشخیص انجمن یک مساله سخت و مهم در مطالعه ساختار شبکه و فهم ویژگی‌های برای استخراج اطلاعات شبکه است. با شناسایی انجمن‌ها و افراد متعلق به چند انجمن، می‌توان سرعت انتشار اطلاعات در شبکه را افزایش داد. شناسایی انجمن‌های همپوشان و فنون تشخیص آن‌ها از محورهای مطالعاتی جدید محسوب می‌شود [۱].

۱-۱- مدل انجمن

در تئوری گراف، انجمن‌ها در شبکه‌های اجتماعی را می‌توان با گراف دارای تراکم زیاد نشان داد. از لحاظ آماری تراکم یال‌های زیرگراف‌های انجمن نسبت به تراکم گراف تصادفی با همان تعداد یال و گره بیشتر باشد، در این صورت انجمن‌ها یا زیرگراف‌ها مستقل از هم هستند و اعضای مشترک ندارند. گراف G(V;E) شامل گره‌ها و یال‌های بسیاری است. گراف دارای مجموعه غیرتهی V رأس و مجموعه E یال است. برای هر یال G=(V,E),e ∈E گره‌های u و v را انتهای یال می‌گویند. زیرگراف H = (V′, E′) از گراف G = (V,E)، گرافی که گره‌های آن زیرمجموعه گره‌های گراف G و یال‌های آن زیرمجموعه یال‌های G باشند. جزء متصل[۲] به زیرگرافی گفته می‌شود که تمام گره‌های آن به هم متصل هستند. k_i^{in} را درجه ورودی، k_i^{out} را درجه خروجی و k_i را درجه گره i گویند. ماتریس همسایگی گراف G با نام A(G) با یک ماتریس n × n تعریف می‌شود که در صورت اتصال گره‌های i و j مقدار $A_{ij} = 1$ و در غیر آن مقدار $A_{ij} = 0$ است. به مجموعه‌ای از زیرگراف‌های گراف G که تمام گره‌های گراف را پوشش می‌دهند گراف‌پوششی G گویند. گراف‌های پوششیG همان انجمن‌های هم‌پوشان $C = \{c_i; c_i \subset G, \coprod c_i = G\}$ هستند. هر گره در گراف‌های پوششی می‌تواند عضو چند زیرگراف باشد که برای هر گره بردار تعلق به‌صورت $\beta_i = \{b_{ic}; c \epsilon C, \sum b_{ic} = 1\}$ تعریف می‌شود.

* رایانامه نویسنده مسئول: mahabadi@shahed.ac.ir

[1] Entities

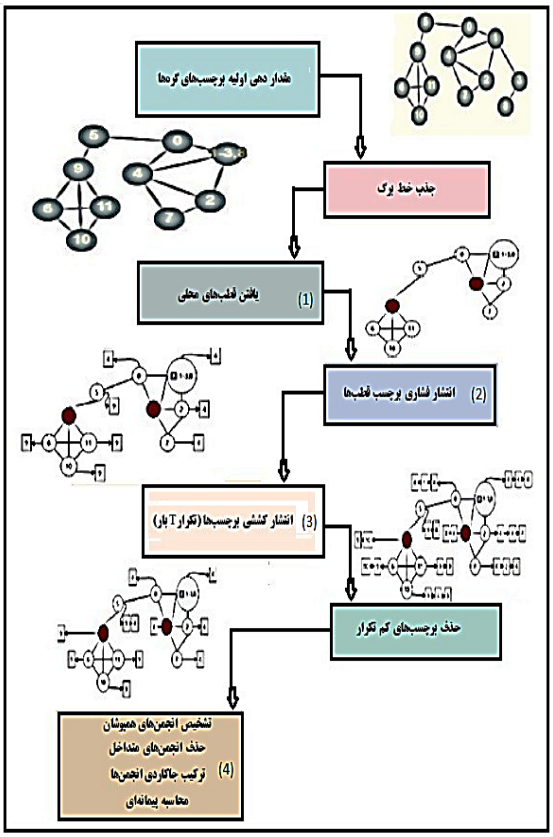
[2] Connected Components

۱-۲- تشخیص انجمن

تشخیص انجمن، تمرکز بر مناطقی از گراف که دارای استقلال نسبی هستند را فراهم می‌کند [۲]. از روش‌های سنتی شناسایی انجمن‌ها، پارتیشن‌بندی[1] گره‌های شبکه به تعدادی انجمن از پیش تعیین‌شده است [۳]. غلظت ارتباط افراد پارتیشن با معیار ضریب خوشه‌بندی اندازه‌گیری می‌شود که برابر نسبت یال‌های موجود در گروه به کل یال‌های ممکن در آن گروه است [۴].

تشخیص انجمن هم‌پوشان: انجمن‌های دنیای واقعی هم‌پوشان هستند. روش‌های تشخیص انجمن هم‌پوشان، گراف را به بخش‌های کاملاً مجزا تفکیک نمی‌کنند و میزان یافتن هم‌پوشانی در روش‌های مختلف با هم تفاوت دارند. برای تشخیص انجمن‌های هم‌پوشان روش لوین [۵] ارائه شد. همچنین چندین الگوریتم چندسطحی با خوشه‌بندی یال‌ها ارائه شد که خواص شبکه، انجمن و گره را در سطوح مختلف بین کرد [۶]. برای حل مشکل همپوشانی روش‌های انتشار اطلاعات مطرح شد. روش خطی تشخیص انجمن به نام انتشار برچسب ارائه شد که با استفاده از ساختار شبکه و بدون نیاز به دانش قبلی از انجمن‌ها، به تشخیص آن‌ها می‌پردازد [۷]. این روش الگوریتم یادگیر شبه-متمرکز است. ابتدا هر گره برچسب منحصر به فردی اختیار کرده و سپس گره‌ها با بررسی همسایگان خود، با بیشترین تکرار برچسب، همسایگان خود را انتخاب می‌کنند. روش انتشار برچسب با افزودن تابع بیشینه‌سازی ارائه شد که مشکل عدم خروج از بیشینه محلی داشت [۸]. برای خروج از دام این بیشینه محلی، در هر مرحله بعد از انتشار برچسب، انجمن‌های یافته‌شده با معیار پیمانه‌ای، با هم ترکیب و انجمن بزرگ‌تری تشکیل دادند [۹].

تشخیص موازی انجمن هم‌پوشان: به‌دلیل گسترش شبکه‌های اجتماعی و گسترش کاربران آن‌ها، شبکه‌های بسیار پیچیده تشکیل شد که دارای گره‌ها و یال‌های بسیار هستند. تحقیقات اخیر برای این شبکه‌های بزرگ و پیچیده، به ارائه روش‌های سریع تشخیص انجمن متمرکز شده‌اند. آن‌ها زمان جستجوی برای اجرا را دنبال می‌کنند تا مقیاس‌پذیری آن روش‌ها قابل بیان باشد. روش‌های مقیاس‌پذیر از ساختارهای توزیعی و موازی بهره می‌برند. اکنون طراحی الگوریتم‌های موازی که با استفاده از امکانات سخت‌افزاری بتوانند به مقیاس‌پذیری و اجرای سریع‌تر کمک کنند از اهمیت ویژه‌ای برخوردارند [۱۰-۱۳]. همگام‌سازی داده‌ها در محاسبات موازی از چالش‌های خطی‌سازی این روش‌ها است. یکی از روش‌های مهم خطی برای تشخیص انجمن‌های هم‌پوشان، روش انتشار برچسب گوینده شنونده است که بر اساس تکرار انتشار اطلاعات، به شناسایی انجمن‌ها می‌پردازد. این روش در دو ساختار معماری چندماشینی [۱۳] و چندهسته‌ای [۱۴] پیاده‌سازی می‌شوند.



شکل (۱): روندنمای روش پیشنهادی SLPA-PP.

۱-۳- روش پیشنهادی و ارزیابی

روندنمای روش پیشنهادی SLPA-PP[2] در سه مرحله اساسی و مطابق شکل (۱) به شناسایی سریع و دقیق انجمن‌های همپوشان در شبکه می‌پردازد. در مرحله۱ (*یافتن قطب‌ها*) با دریافت گراف ورودی و بر اساس اطلاعات سراسری گراف، در یک پالایش، خط برگ‌ها شناسایی و جذب گره‌های جدشان می‌شوند. سپس گره‌های شبه‌قطب به تعداد k گره متوالی انتخاب می‌شوند. در مرحله ۲ (*فشار برچسب*) گره‌های شبه قطب بر اساس اطلاعات محلی، نفوذ خود را از طریق انتشار برچسب به همسایگانشان اعمال می‌کنند و با استفاده از *ضریب تعریفی انتشار قطب* کنترل می‌شوند. در مرحله ۳ (*تشخیص انجمن*) هر گره بر اساس اطلاعات انجمن‌ها با دریافت اطلاعات برچسب همسایگانش، برچسب کششی خود را انتخاب می‌کند. در این مرحله انجمن‌هایی که زیرمجموعه انجمن‌های دیگر هستند باید از لیست انجمن‌ها موجود خارج شوند. با بررسی انجمن‌ها و استفاده از

[1] Partitioning

[2] Speaker Listener Propagation Push Pull Algorithm (SLPA-PP)

ضریب تعریفی تشابه جاکارد، انجمن‌های مشابه ترکیب و از طریق الگوریتم گوینده-شنونده و محاسبه ارزش پیمانه‌ای، انجمن‌های همپوشان انتخاب می‌شوند. نهایتاً خروجی الگوریتم، لیست زیرگراف‌های پوششی را به‌عنوان انجمن‌های هم‌پوشان ارائه می‌دهد.

ایده اصلی روش مقیاس‌پذیر پیشنهادی استفاده از تئوری اطلاعات برای انتشار موازی اطلاعات در شبکه و کنترل مراحل انتشار است. در گام اول، اعمال نفوذ گره‌های شبه‌قطب در حلقه اول انتشار برچسب به همسایگان خودشان است. در روش‌های قبلی، انتشار برچسب تمامی گره‌ها به‌صورت تصادفی صورت می‌گرفت ولی در روش پیشنهادی انتشار اطلاعات به‌صورت هدایت‌شده و فقط از طریق گره‌های دارای نفوذ بیشتر انجام می‌گیرد. گراف شبکه به k دسته تقسیم و در هر دسته گرهی که بیشترین درجه نسبت به همسایگانش را دارد به‌عنوان گره شبه قطب انتخاب و برچسب خود را به همسایگانش تحمیل می‌کند. در گام دوم انتشار برچسب قطب‌ها به‌صورت تصادفی منتشر و از سرایت آن به قطب‌های دیگر از طریق معیار HPP جلوگیری می‌شود. نهایتاً در گام سوم با جذب برچسب‌های انتشاری و پردازش آن‌ها، تشخیص بهینه انجمن‌های هم‌پوشان صورت می‌گیرد.

روش پیشنهادی قدرت مطلوب تشخیص انجمن‌های هم‌پوشان در شبکه‌های بزرگ و پیچیده را دارد. این روش با تمرکز بر کاهش فضای جستجو و توزیع موازی آن و استفاده مرحله‌ای از اطلاعات سراسری گراف، اطلاعات انجمن‌ها و اطلاعات محلی از روش‌های قبلی دقیق‌تر، سریع‌تر و کاراتر است. این روش با توسعه موازی‌سازی چندهسته‌ای [۱۴] در موازی‌سازی نخ‌ها بهبود ایجاد کرده و الگوریتم سریع‌تری ارائه می‌دهد. همچنین ساختار داده‌ای مناسب با استفاده از TBB[1] اینتل ارایه‌داده [۱۵] که شبکه‌های بزرگ را با موازی‌سازی بیشتری، سریع‌تر پردازش کند. همچنین همراه بهینه‌سازی ساختار ذخیره‌سازی داده‌ها، بارگذاری شبکه در حافظه به‌صورت موازی انجام می‌شود. همچنین با تغییراتی در الگوریتم سنتی گوینده شنونده و با استفاده از ساختارهای داده‌ای مناسب چندنخی و کاهش مراحل یافتن برچسب، سرعت اجرای الگوریتم را افزایش داده است.

این روش با انتشار فشاری برچسب قطب‌ها، سرعت اجرا را بیشتر کرده و با اعمال معیار پیمانه‌ای نیکوشیا برای انجمن‌های هم‌پوشان، نشان داد که کارایی الگوریتم پیشنهادی بیشتر از روش‌های سنتی است. با ترکیب انجمن‌های مشابه توسط تشابه جاکارد، معیار پیمانه‌ای را بیشتر کرده و الگوریتم جدیدی برای پردازش توزیعی با روش انتشار برچسب گوینده شنونده فشاری-کششی ارائه داده است که در آن فشار برچسب از سوی قطب‌ها و کشش برچسب از سوی سایر گره‌ها انجام می‌شود. نتایج آزمایش‌ها بر روی داده‌های استاندارد نشان می‌دهد که این روش در مقایسه با روش‌های قبلی، دارای ساختار موازی چندهسته‌ای، پیچیدگی زمانی خطی، تسریع ارزشمند، بهبود قابل ملاحظه‌ای پیمانه‌ای و کاهش نسبی حافظه مصرفی است.

۱-۴- انگیزه و نوآوری

محرک روش پیشنهادی همان روش موجود گوینده-شنونده و بر اساس تئوری اطلاعات است. روش‌های قبلی تشخیص انجمن هم‌پوشان، انجمن‌های شبیه به هم می‌یابند. انتشار برچسب به‌صورت کاملاً تصادفی در کل گراف انجام می‌شود. همچنین برچسب‌ها در برگ‌ها نیز منتشر می‌شوند که علاوه بر چالش صرف زمان زیاد، موجب تشخیص برگ‌ها در انجمن‌های ناهمبند می‌شوند که در مرحله نهایی الگوریتم این گره‌ها باید از انجمن‌های ناهمبند حذف شوند. گراف به‌صورت متوالی به حافظه بارگذاری می‌شود که معیاری برای اندازه‌گیری کارایی این روش‌ها ارایه نشده است. چالش‌های حل این مساله به روش پیاده‌سازی، معماری اجرا، داده‌های عظیم، مقیاس‌پذیری روش، تشخیص انجمن‌های هم‌پوشان، سرعت‌عمل تشخیص، کندی فضای جستجو و دقت تشخیص بر می‌گردد. به‌طور خلاصه نوآوری‌های ارائه‌شده در این مقاله به شرح زیر است.

- طراحی الگوریتم توزیعی تشخیص انجمن‌های همپوشان در شبکه‌های اجتماعی بزرگ و پیچیده،
- مدل‌سازی عامل‌گرای توزیعی و افزایش قدرت مقیاس‌پذیری الگوریتم پیشنهادی،
- کاهش پیچیدگی زمانی و حافظه‌ای الگوریتم با تمرکز بر کاهش فضای جستجو و کنترل مراحل انتشار برچسب،
- استفاده از ساختار امن چندنخی اینتل TBB و بارگذاری موازی شبکه به حافظه،
- ترکیب انجمن‌های شبیه به هم با معیار تشابه جاکارد و افزایش پیمانه‌ای‌بودن نتایج اجرای الگوریتم،
- اجرای الگوریتم با زمان‌بندی نخ‌ها در openMP به‌صورت ایستا، پویا با تعداد کار خودکار، پویا با تعداد کار ثابت و هدایت‌شده و یافتن زمان‌بندی بهینه پویای کارها، و
- کاربرد سه معیار جدید قطب‌محلی، ضریب انتشار، تشابه جاکارد در ترکیب انجمن‌ها.

در ادامه در بخش ۲ پیش‌نیازها و کارهای مرتبط را مرور و

[1] Threading Building Block (TBB)

روش‌های تشخیص انجمن‌های هم‌پوشان و غیر هم‌پوشان را بررسی و در نهایت روش‌های موازی‌سازی الگوریتم‌های تشخیص انجمن ارائه می‌شود. در بخش ۳ روش پیشنهادی تشریح می‌گردد. در بخش ۴ نتایج آزمایش‌های تجربی و مشکلات فنی پیاده‌سازی الگوریتم بیان می‌شود. در بخش۵ نتیجه‌گیری و کارهای آینده ارائه می‌شود.

۲- کارهای مرتبط

در این بخش، به مبانی مورد نیاز تشخیص انجمن‌های هم‌پوشان و کارهای مرتبط اشاره می‌شود. انواع الگوریتم‌های مهم و مزایا و معایب آن‌ها ارائه می‌شود و در نهایت الگوریتم‌های مهم تشخیص انجمن‌های هم‌پوشان و مقایسه می‌گردند.

۲-۱- تشخیص انجمن

تشخیص انجمن از مؤلفه‌های مهم در مطالعه ساختار شبکه و فهم ویژگی‌های آن است. انجمن به‌عنوان یک گروه از اشخاص تعریف‌شده به‌طوری‌که شامل تراکم شدید درون‌گروهی و تراکم کم بین گروه‌ها است [۱۶]. انجمن (خوشه‌ها یا پیمانه‌های گراف) گروهی از رئوس هستند که خواص مشترک یا نقش مشابهی دارند [۱۷]. تشخیص انجمن، تمرکز بر مناطقی از گراف است که دارای استقلال نسبی هستند. افراد یک گروه تمایل بیشتری به ارتباط دارند و دوستان دوست، با احتمال بیشتری دوست فرد خواهند بود که به این خاصیت، ساختار انجمن قوی گفته می‌شود. غلظت ارتباط افراد گروه با معیار ضریب خوشه‌بندی اندازه‌گیری می‌شود که برابر نسبت یال‌های موجود در گروه به‌کل یال‌های ممکن در آن گروه است [۴].

تشخیص دقیق انجمن‌ها در گراف، مساله سخت[1] است [۱۸] و عموم روش‌های تشخیص انجمن تقریبی هستند. باید معیاری برای تعیین کیفیت نتیجه الگوریتم‌ها موجود باشد. ابتدا معیار اطلاعات متقابل[2] از تئوری اطلاعات ارائه شد که رابطه اطلاعاتی بین دو متغیر تصادفی X و Y را بیان می‌کند. این معیار نشان می‌دهد که اگر X را بشناسیم چقدر اطلاعات راجع به Y داریم و برعکس. معیار اطلاعات متقابل هنجار[3] با این ایده ارائه شد که اگر دو گروه دارای رأس با شباهت زیاد باشند، یک گروه برای پی بردن به گروه دیگر به اطلاعات کمی نیاز دارد که با ترکیب آنتروپی شانون[4] با معیار اطلاعات متقابل فرموله می‌شود.

در شبکه‌های اجتماعی واقعی تعداد ثابت انجمن با اندازه‌های یکسان کمتر اتفاق می‌افتد و بیشتر انجمن‌ها هم‌پوشانی وجود دارد. در روش‌های تشخیص انجمن هم‌پوشان، گراف را به تکه‌های کاملاً مجزا تفکیک نمی‌کنند ولی میزان یافتن هم‌پوشانی در روش‌های مختلف با هم تفاوت دارند. هم‌پوشانی انجمن‌ها از چالش‌های مهم تشخیص انجمن است. تشخیص انجمن هم‌پوشان از الگوریتم نفوذ دسته آغاز [۱۹] ، COPRA [۲۰]، CONGA[5] [۲۱]، CONGO[6] [۲۰] و EAGLE[7] [۲۲] ارائه شد.

۲-۲- تشخیص انجمن‌های هم‌پوشان

تشخیص انجمن‌های هم‌پوشان دارای روش‌های مهم بسیاری از پیمانه‌ای، خوشه‌بندی، انتشار برچسب، بذر تا انتشار برچسب است که به بیان آن‌ها می‌پردازیم.

روش پیمانه‌ای: روش اکتشافی جستجوی محلی لوین [۵] برای تشخیص انجمن هم‌پوشان تعمیم یافت. تعداد انجمن‌ها از قبل معلوم نبود و باید در هر مرحله به‌صورت پویا مشخص می‌شد. همچنین با هم‌پوشانی زیاد دو انجمن ترکیب و یک انجمن بزرگ‌تر ایجاد شد. همچنین تابع بهینه‌سازی لوین فقط برای تشخیص انجمن‌های مجزا مناسب بود. از آنجا که تابع بهینه‌سازی برای انجمن‌های هم‌پوشان بر اساس مثلث‌ها برای انجمن‌هایی که هم‌پوشانی خوبی دارند مناسب است [۲۳] لذا تعمیم wcc[8] به wocc[9] برای انجمن‌های هم‌پوشان و گسترش پیمانه Q^E برای انجمن‌های هم‌پوشان دارای هم‌پوشانی کم انجام شد که تابع الگوریتم با توجه به ساختار گراف به‌صورت پویا تصمیم به انتخاب یکی از دو معیار فوق می‌کند.

روش خوشه‌بندی یال: برای تشخیص انجمن هم‌پوشان، الگوریتمی چندسطحی به روش خوشه‌بندی یال‌ها از مرتبه $O\left(N\overline{k^2} + \overline{N_{com}^2}\right)$ ارائه شد که در آن k درجه مورد انتظار گره‌ها و N_{com} اندازه انجمن است [۶]. در روش نفوذ دسته، زیرگراف کامل تعیین و به‌عنوان مرکز انجمن گسترش می‌یابد. این روش با تعریف گرافیست نیمه‌کامل به‌نام گراف آشنایی تعمیم یافت و همسایه‌های محلی هر گراف را به‌عنوان هسته‌های انجمن در نظر گرفته شد. در تعمیم روش به‌جای تعلق گره به‌دسته‌ها، تعلق آن به حلقه‌های دوستی[10] بیان شد. حلقه دوستی به مجموعه‌ای از یال‌ها که از دسته آشنایی برگرفته شده‌اند و شرایط انجمن را دارند، گفته می‌شود.

تعریف انجمن در سه سطح گره، انجمن و شبکه ارائه می‌شود. تشخیص انجمن توسط خوشه‌بندی یال‌ها انجام می‌شود که هر

[1] None-deterministic Polynomial Time Hard
[2] Mutual Information
[3] Normalized Mutual Information (NMI)
[4] Shannon Entropy

[5] Cluster Overlap Newman Girvan Algorithm (CONGA)
[6] CONGA Optimized
[7] Agglomerative hierarchicAl clusterinG based on maximaL cliquE (EAGLE)
[8] Weighted Community Clustering (WCC)
[9] Weighted Overlapping Community Clustering (WOCC)
[10] Friendship Circles

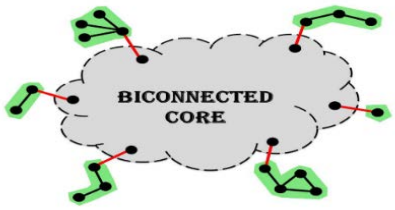
گره می‌تواند متعلق به چندین انجمن باشد. خصوصیت اصلی در سطح انجمن آن است که چگالی یال‌های انجمن بیشتر از چگالی یال‌های گراف باشد که این ویژگی در سطح گره برآورده نمی‌شود. ویژگی مجموعه انجمن‌ها در سطح شبکه آن است که گره‌های خوشه‌بندی نشده نداشته باشد و کمترین تعداد انجمن را برای پوشش شبکه تشکیل دهد. از معیار *F-Score* برای تعیین کیفیت تشخیص انجمن استفاده می‌شود. دقت[1] با رابطه (۱) بیان می‌شود و نسبت تشخیص درست انجمن‌های به‌کل انجمن‌های تشخیصی را محاسبه می‌کند. فراخوانی[2] با رابطه (۲) بیان و نسبت تعداد انجمن‌های تشخیص داده‌شده به‌کل انجمن‌های واقعی را محاسبه می‌کند [۶].

$$Precision = \frac{T_p}{T_p + F_p} \quad (۱)$$

$$F = \frac{2 * Precision * Recall}{Precision + Recall} \quad (۲)$$

روش انتشار برچسب یال: روش تشخیص انجمن یالی[3] از اطلاعات هم‌پوشان استفاده می‌کند. با این حال نتایج این الگوریتم‌ها مانند انجمن‌های گره محور بهینه نیست. لئو در [۲۴] روشی با ترکیب انتشار برچسب و تشخیص انجمن یالی با نام ELPA[4] ارائه کرد. در آن کارایی روش انتشار برچسب با مزایای روش‌یالی ترکیب و در نتیجه هم انجمن‌های یالی و هم انجمن‌های گره‌ای را تشخیص داد. ایده اصلی الگوریتم ELPA ایجاد انجمن بر اساس یال‌های متصل به هم و متراکم بر مبنای برچسب آن‌ها است. هر یال برچسب خود را بر اساس برچسب یال‌های همسایه‌هایش به‌روز می‌کند. الگوریتم ELPA دارای چهار مرحله راه‌اندازی، انتشار برچسب یال، انتشار برچسب گره، و شناسایی پل‌ها است. در مرحله شناسایی پل، یال‌ها به‌عنوان پل بین دو انجمن، مشخص می‌شوند. روش ELPA نسبتاً ساده و پایدار، بدون پارامتر ورودی و فقط بر اساس ساختار شبکه و بدون دانش قبلی از شبکه عمل می‌کند.

روش متورم‌سازی بذرها: ونگ در [۲۵] روش تشخیص انجمن هم‌پوشان را با متورم‌سازی بذرهای انتخابی به همسایگانش ارائه داد. ایده اصلی آن یافتن بذرهای اولیه مناسب و گسترش آن‌ها با روش حریصانه و استفاده از معیار انجمن است. مهم‌ترین قدم، متورم‌سازی بذرها برای دربرگرفتن همسایگان مجاورش است. در راهبردهای قبلی انتخاب بذرها کم و دور از هم صورت گرفت و معمولاً دسته‌های بزرگ به‌عنوان بذر انتخاب می‌شدند که محاسبات سنگینی داشت. در این روش با الگوریتم چندسطحی k-means چندین خوشه محاسبه و سپس توسط تابع فاصله مربوطه، گره مرکزی هر خوشه انتخاب شد. هر گره می‌تواند به چند انجمن تعلق داشته باشد یا عضو هیچ انجمنی نباشد. الگوریتم دارای چهار مرحله پالایش، بذرپاشی، گسترش بذر و انتشار است. در مرحله پالایش، مناطقی از گراف که به‌وضوح مستقل هستند حذف می‌شوند. در مرحله بذرپاشی، گره‌های مناسب از گراف پالایش‌شده انتخاب می‌شوند. برای گسترش بذر، از طرح خوشه‌بندی رتبه صفحه شخصی استفاده می‌شود. در آخر گسترش انجمن‌ها به مناطقی از گراف که در مرحله پالایش حذف شد، صورت می‌گیرد.



شکل (۲): زیرگراف‌های هسته و حاشیه [۲۵].

تقسیم گراف به زیرگراف‌های هسته و حاشیه، پالایش گراف نامیده می‌شود (شکل(۲)). زیرگراف‌های هسته همبند هستند و گراف‌های حاشیه دارای یک یال، به این زیرگراف متصل هستند. زیرگراف‌های حاشیه جزو مناطق هم‌پوشان نیستند. بر اساس اطلاعات آماری، زیرگراف‌های حاشیه مستقل از هم هستند و اندازه آن‌ها بزرگ نیست. همچنین فاصله اندازه گراف‌های هسته و حاشیه زیاد است. یافته‌های آماری مبین ایجاد گراف مناسب برای تشخیص انجمن هم‌پوشان در مرحله پالایش است. یافتن گره‌های پخش در خوشه‌هایی که هدایت خوبی دارند، انتخاب بذر نام دارد. انجام این کار محاسبات سنگینی ندارد. احتمال خروج از خوشه با یک حرکت تصادفی از داخل خوشه را هدایت خوشه می‌گویند. کمی این احتمال مبین یک خوشه با کیفیت است. این کار با انتخاب گره‌های با بیشترین درجه و همچنین خوشه‌بندی سلسله‌مراتبی بالا به پایین مراکز گراکلوس[5] انجام می‌شود [۳۳].

بسط آن‌ها برای رسیدن به خوشه‌ها با داشتن گره‌های بذر و با فن رتبه صفحه‌شخصی که به حرکت تصادفی[6] با شروع مجدد معروف است، انجام می‌شود. گسترش بذر توسط چندنخ و به‌صورت موازی از بهبودهای این الگوریتم است. رتبه صفحه شخصی با احتمال α به گره تصادفی دیگر حرکت می‌کند و به احتمال (1-α) به یکی از بذرهای منتخب بر می‌گردد. در صورت وجود چندین بذر، انتخاب بذر برگشتی معمولاً با توزیع تصادفی یکنواخت است. بنابراین گره‌های نزدیک به بذر، با احتمال بیشتری دیده می‌شوند که بیشتر به انجمن‌های دنیای واقعی نزدیک هستند. در واقع بردار رتبه صفحه‌شخصی به برش گراف و

[1] Precision
[2] Recall
[3] Link Community
[4] Edge Label Propagation Algorithm (ELPA)
[5] Graclus Centers
[6] Random Walk

روش‌های خوشه‌بندی شباهت دارد. بعد از یافتن انجمن‌های دارای رتبه‌بندی شخصی در زیرگراف هسته، انجمن‌های یافت‌شده به زیر گراف‌های حاشیه که در مرحله پالایش حذف شد گسترش می‌یابند. گسترش انجمن‌ها ساده است و زیرگراف‌های حاشیه که با یک یال به هسته وصل هستند به تمام انجمن‌هایی که شامل یک گره از یال هستند افزوده می‌شوند.

روش انتشار برچسب: روش خطی تشخیص انجمن به نام انتشار برچسب توسط راگاوان [۷] ارائه شد که با استفاده از ساختار شبکه و بدون نیاز به دانش قبلی از انجمن‌ها، به تشخیص می‌پردازد. ابتدا هر گره برچسب منحصر به فردی اختیار و سپس با بررسی همسایگان خود، بیشترین تکرار برچسب موجود در همسایگان را انتخاب می‌کند. در صورت وجود چندین برچسب با بیشینه مساوی، یکی از بیشینه‌ها به‌صورت تصادفی انتخاب می‌شود. برای پایان الگوریتم، روش انتشار غیر هم‌زمان را ارائه داد که در هر مرحله با برخی از برچسب‌های فعلی و برچسب‌های مرحله قبلی همسایگانش به‌روزرسانی می‌شود. این الگوریتم هنگامی‌که تمام گره‌ها، دارای برچسبی باشند که متعلق به گره‌های دیگر نیز هست پایان می‌یابد. گره‌های دارای برچسب یکسان ممکن است همبند نباشند لذا یک مرحله پایانی جهت جداسازی انجمن‌های همبند به الگوریتم اضافه شد. پیچیدگی الگوریتم خطی $O(n)$ و هر تکرار از مرتبه $O(m)$ است. تخمین تعداد تکرار حلقه سخت است ولی به ادعای راگاوان بعد از ۵ تکرار ۹۵ درصد از گره‌ها دارای برچسب صحیح هستند. پردازش گراف‌های همبند این لگوریتم از مرتبه $O(n+m)$ است.

باربر در [۸] انتشار برچسب را به‌صورت مساله بهینه‌سازی مدل کرد و با بیشینه‌سازی تابع هدف به یافتن انجمن‌ها پرداخت. این تابع هدف فقط تعداد یال‌های اتصال‌دهنده برچسب‌های یکسان را بیشینه می‌کرد. این تصور دارای عیب مفهومی است که با این بیشینه‌سازی، به انجمن‌های بهتری نتوان رسید. با درنظر گرفتن کل گره‌ها به‌عنوان یک انجمن، به بیشینه مقدار عمومی تابع هدف رسید که ناخوشایند بود لذا باید از رسیدن به انجمن یکتا با درنظر گرفتن جریمه در تابع بهینه‌سازی جلوگیری می‌کرد. بهینه‌سازی LPA فقط با سمت راست رابطه (۳) توسط LPA صورت گرفت که موجب افتادن به دام بیشینه محلی بود لذا یکی از عیوب آن جستجوهای محلی است. با تغییر الگوریتم LPA، دو الگوریتم جدید ارائه شد. LPAm با بیشینه‌سازی محلی تابع پیمانه‌ای Q سروکار داشت و LPAb برای شبکه‌های دوجزئی با درنظر گرفتن تابع پیمانه‌ای تغییریافته Q^B مناسب بود.

$$H = \frac{1}{2}(\sum_{v\neq x}\sum_{u\neq x} A_{uv}\delta(l_u, l_v) - A_{xx}) + \sum_{u=1}^{n} A_{ux}\delta(l_u, l_x) \quad (۳)$$

$$H' = H - \lambda G \quad (۴)$$

در الگوریتم استاندارد LPA، اگر چند برچسب همسایه و خود گره دارای بیشینه فراوانی یکسانی باشند، برچسب گره بدون تغییر می‌ماند. در غیر آن به‌صورت تصادفی یکی از بیشینه همسایه‌ها به‌عنوان برچسب غالب انتخاب می‌شود. با تغییر الگوریتم در شرایط فوق، به‌جای بی‌تغییر نگه‌داشتن برچسب، باز هم تصادفی یکی را انتخاب کرد لذا بهبودی حاصل گردید و امکان حرکت تصادفی در بین همسایه‌ها بیشتر شد. افزایش حالت تصادفی در الگوریتم با LPAr[1] بیان شد. برای جلوگیری از انتشار برچسب به‌کل گراف و تجزیه به انجمن‌های دارای جمع درجه یکسان جهت تابع پیمانه‌ای، جریمه در نظر گرفته شد. برای شبکه‌های دوجزئی، تابع پیمانه‌ای Q^b ارائه شد.

با وجود تشابه فرمولی، تفاوت مفهومی در تابع پیمانه‌ای مورد استفاده در الگوریتم LPAm با تابع پیمانه‌ای استاندارد وجود دارد. انتخاب مقدار λ در رابطه(۴) با مقادیر اختیاری می‌تواند رفتار الگوریتم را تغییر دهد ولی تابع تعریف‌شده می‌تواند دو انجمن با مجموع درجات داخلی یکسان تولید کند. به عبارت دیگر در مقایسه با مدل تصادفی، در تشخیص انجمن‌هایی با اندازه‌های مختلف با مشکل مواجه است. الگوریتم انتشار برچسب با در نظر گرفتن تابع پیمانه‌ای، LPAm نام گرفت و دارای عیب گیرافتادن در بیشینه محلی است. این الگوریتم به +LPAm توسعه یافت تا از به دام افتادن در بیشینه محلی فرار کند.

انتشار برچسب با بیشینه‌سازی پیمانه‌ای، ناپایدار است و در دام بیشینه محلی می‌افتد. روشی برای خروج از این دام در [۹] ارائه شد که در هر مرحله بعد از انتشار برچسب، انجمن‌های یافته‌شده که ترکیب‌شان سبب افزایش پیمانه‌ای‌شدن است، با هم ترکیب و انجمن بزرگ‌تری را تشکیل دهند تا الگوریتم از تله‌بیشینه محلی خارج شود. برای خروج از این تله، از روش ترکیب انجمن‌ها به‌صورت حریصانه استفاده شد. با ترکیب انجمن‌هایی که بیشترین افزایش را در تابع پیمانه‌ای دارند از دام بیشینه محلی خارج و الگوریتم LPAm ادامه می‌یابد زیرا ممکن است در تله محلی دیگری افتاده باشد. الگوریتم تا وقتی‌که تغییر در تابع پیمانه‌ای ایجاد نشود به کار خود ادامه می‌دهد. الگوریتم +LPAm در اجراهای مختلف با کمتر از 5% انحراف، پایدارتر از الگوریتم LPA است. بلاندل[۲۶] الگوریتمی شبیه به +LPAm طراحی کرد که در دو مرحله انجمن‌ها را تشخیص دهد. اول تا رسیدن به حداکثر پیمانه‌ای محلی و سپس در مرحله بعد، انجمن‌ها به‌عنوان گره فرض و شبکه جدید دوباره جهت تشخیص انجمن انتخاب شود. به‌صورت سلسله مراتبی انجمن‌ها تشخیص داده شد تا مشکل محدودیت تجزیه حل شود. در سطوح پایین،

[1] LPA Random (LPAR)

انجمن‌های کوچک و در سطوح بالا، انجمن‌های بزرگ‌تر مشخص شدند. سرعت الگوریتم تقریباً خطی است که بیشتر محاسبات آن در مرحله اول انجام می‌گیرد. معمولاً در کمتر از ۵ سطح، کل انجمن‌ها مشخص می‌شوند.

الگوریتم COPRA: الگوریتم COPRA[1] توسط گرگوری برای تشخیص انجمن‌های هم‌پوشان ارائه شد [۲۰]. او الگوریتم RAK[2] را برای یافتن انجمن‌های هم‌پوشان گسترش داد تا هر گره حداکثر بتواند به v انجمن تعلق یابد که v پارامتر ورودی این الگوریتم است. هر گره می‌تواند به چندین انجمن متعلق باشد لذا برچسب آن باید بتواند چندین شناسه انجمن را در خود نگه دارد. در صورت پیاده‌سازی این روش کلیه گره‌ها به کلیه انجمن‌ها می‌پیوندند. برای برچسب‌های گره‌ها از دوتایی (c,b) استفاده شد که c شناسه انجمن و b ضریب‌تعلق به آن انجمن است. جمع تعلقات هر گره به انجمن‌های مربوطه باید برابر ۱ باشد. تابع تعلق $b_t(c,x)$ میزان تعلق گره x به انجمن c را در تکرار t نشان می‌دهد. این تعلق همیشه وابسته به تعلقات گره‌های همسایه x در مرحله قبل، طبق رابطه (۵) است و در آن $N(x)$ مجموعه همسایه‌های x است.

$$b_t(c,x) = \frac{\sum_{y \in N(x)} b_{t-1}(c,y)}{|N(x)|} \qquad (۵)$$

روش فوق به تعداد گره‌ها، انجمن تشخیص داده و همه گره‌ها در انتها دارای یک برچسب می‌شوند که روش فوق مناسبی نیست. نیاز به‌روشی برای داشتن چند انجمن است بدون آن‌که تمام گره‌ها را شامل شود. لذا در هر انتشار، برچسب‌هایی که کمتر از مقدار آستانه‌ای تعلق را نشان دهند از لیست حذف می‌شوند. مقدار آستانه برابر $1/v$ در نظر گرفته می‌شود که v پارامتر ورودی است. v حداکثر تعداد انجمن‌هایی است که گره می‌تواند به آن‌ها تعلق داشته باشد. درصورت بیشتر شدن تعداد انجمن‌هایی که مقدار آستانه آن‌ها بیش از v شود، از بزرگ‌ترین تعلق‌ها به تعداد v انتخاب شده و بقیه حذف می‌گردند. در صورتی که همه تعلقات به انجمن‌ها کمتر از مقدار آستانه باشد، تصادفی یکی از همسایه‌ها انتخاب می‌شود. به‌دلیل انتخاب تصادفی الگوریتم، این انجمن‌ها غیرقطعی هستند. بعد از حذف دوتایی‌های اضافی، میزان تعلق سایر دوتایی‌های گره در مقدار ثابت ضرب می‌شود تا مجموع آن‌ها به مقدار ۱ برسد.

یکی از مشکلات ارثی الگوریتم RAK، بزرگ‌شدن بیش از حد اندازه انجمن‌ها است. برای حل آن، بعد از هر انتشار، اگر برچسب x دو بار در برچسب‌های گره x وجود داشته باشد تمام برچسب‌های x حذف و فقط برچسبx باقی می‌ماند تا سبب عدم انتشار گره x به انجمن‌های دیگر شود. این عمل در بدترین حالت بهبود ایجاد می‌کند است ولی در مواردی که اندازه انجمن باید بزرگ‌تر باشد دچار ضعف می‌شود. بعضی مواقع انتشار برچسب‌ها به حالت پایداری نمی‌رسد و داخل حلقه‌ای می‌افتاد که باعث تکرار حالت‌های برچسب است. تعداد انجمن‌ها در الگوریتم با n شروع و با حذف برچسب انجمن‌ها در هر مرحله، تعداد انجمن‌ها کاهش می‌یابد تا به یک انجمن برسد. برای خروج از حلقه تکرار، اندازه انجمن‌ها در مرحله قبلی و فعلی مقایسه و در صورت عدم تغییر تعداد و اندازه انجمن‌ها، الگوریتم خاتمه می‌یابد. بعد از اتمام مرحله انتشار برچسب، گره‌های شامل برچسب c در گروه c قرار دارند. گرچه انجمن‌های یافت‌شده می‌توانند زیرمجموعه یا مساوی هم باشند. برای سرعت بیشتر، هنگام ساخت انجمن‌ها، رابطه زیرمجموعه بودن بررسی و انجمن‌های زیرمجموعه حذف می‌شوند. همانند الگوریتم RAK، الگوریتم COPRA نیز انجمن‌های غیرهمبند تولید می‌کند که در مرحله پایانی باید به انجمن‌های همبند شکسته شوند. پیچیدگی الگوریتم برای شبکه تنک برابر $O(v^3 n) + O(vn\ log\ (v))$ است و با کوچک‌بودن v تقریباً خطی است.

الگوریتم SLPA[3]: روش انتشار برچسب گوینده-شنونده [۲۷]، تعمیم روش انتشار برچسب LPA[4] [۷] است. در LPA هر گره دارای یک برچسب است که آن را با توجه به برچسب‌های گره‌های همسایه خود به‌روزرسانی می‌کند و در صورت همگرایی‌روش، انجمن‌های غیرهم‌پوشان تعیین می‌شوند. در روش گوینده-شنونده انجمن‌های هم‌پوشان با پردازش چندین برچسب هر گره تعیین می‌شوند. این روش، شبیه‌سازی تعامل انسان‌ها در گفت‌وشنود یعنی یک گره گوینده (تولید اطلاعات) و دیگری شنونده (مصرف اطلاعات) است. برخلاف روش‌های قبلی، هر گره دارای حافظه‌ای از برچسب‌ها است که در تصمیم‌گیری استفاده می‌شود تا انجمن‌های بزرگ‌تری را تشخیص دهند. ابتدا هر گره نماینده یک انجمن و دارای یک برچسب واحد است. الگوریتم با پیمایش گره‌های شبکه، انجمن‌ها را مشخص می‌کند و نیازی به تعیین تعداد آن‌ها در الگوریتم نیست. به حافظه هر گره در هر گام یکی افزوده و با محدود کردن اندازه حافظه به یک، الگوریتم به انتشار برچسب معمولی تبدیل می‌شود. در تکرارهای بیش از ۲۰ بدون توجه به اندازه شبکه، خروجی نسبتاً پایداری تولید می‌شود. الگوریتم با توجه به انتخاب تصادفی گره‌های همسایه، غیرقطعی است.

[1] Community Overlap PRopagation Algorithm (COPRA)
[2] Raghavan, Albert and Kumara (RAK)
[3] Speaker Listener Propagation Algorithm (SLPA)
[4] Label Propagation Algorithm (LPA)

با اتمام الگوریتم، اطلاعات ذخیره‌شده پردازش و انجمن‌ها تعیین می‌شوند. حافظه برچسب گره‌ها به‌احتمال تبدیل و از آنجا که برچسب گره‌ها نماینده انجمن‌ها هستند، این احتمال طبیعتاً نشانگر میزان قدرت ارتباط برچسب‌ها به انجمن‌ها است. برای تشخیص انجمن‌های مجزا، مقدار احتمال کمتر از ۰/۵ حذف و مابقی گروه‌بندی می‌شوند. اگر گره‌ای دارای بیش از یک برچسب باشد متعلق به چند انجمن و عضو هم‌پوشان انجمن‌ها است. ماهیت روش انتشار برچسب به‌گونه‌ای است که بهتر می‌توان آن را موازی ساخت. کوزمین در [۱۳]، کیاو در [۱۴] و ونگ در [۲۸] به موازی‌سازی الگوریتم گوینده- شنونده پرداختند. همچنین ژنگ در [۲۹] روشی مشابه [۲۸] ارائه داد که با نگاشت و کاهش به انتشار برچسب پرداخت.

الگوریتم توزیعی SLPA: در این روش توزیعی مبتنی بر پیام[1]، شبکه به چند بخش جدا تقسیم و هر بخش به یک پردازنده تخصیص می‌یابد. هر پردازنده، فهرست برچسب گره‌های محلی خود را به دیگر پردازنده‌ها می‌فرستد به‌طوری‌که آن‌ها بتوانند گره‌های غیرمحلی خود را به‌روزرسانی کنند [۱۳]. این الگوریتم از حداقل هم‌زمان‌سازی داده‌ها استفاده می‌کند و هیچ تکرار داده‌ای ندارد. هر پردازنده تنها داده مربوط به خود را در حافظه نگه می‌دارد و ارسال و دریافت داده‌ها را غیر هم‌زمان انجام می‌دهد. پردازنده‌ها قبل از ارسال درخواست‌های بیشتر ارسال و دریافت داده، اطمینان می‌یابند که درخواست‌های دریافت و ارسال قبلی آن‌ها به پایان رسیده باشد. افزایش اندازه داده‌ها، عملکرد کلی سیستم را محدود نمی‌کند. بعلاوه، نتایج نشان‌داده که با افزایش داده‌ها، تسریع[2] الگوریتم موازی افزایش می‌یابد.

این الگوریتم، فهرست گره‌های موجود در شبکه را می‌پیماید. هر گره i به‌صورت تصادفی یکی از همسایگانش را انتخاب می‌کند. همسایه به‌صورت تصادفی یک برچسب از فهرست برچسب‌ها انتخاب و آن را به گره درخواست‌کننده می‌فرستد. گره i فهرست برچسب‌های محلی خود را با برچسب دریافتی به‌روز می‌کند. این فرآیند برای همه گره‌های شبکه تکرار می‌شود. پس از اتمام، فهرست گره‌ها به‌صورت تصادفی آمیخته و همان پردازش برای همه گره‌ها دوباره تکرار می‌شود. پس از تکرار و پردازش انتشار برچسب، هر گره فهرستی از برچسب‌ها به طول t در اختیار دارد زیرا هر گره در هر تکرار یک برچسب دریافت می‌کند. پس از تکمیل پردازش تکرارها، پردازش بر روی فهرست برچسب‌ها انجام و انجمن‌ها استخراج می‌شوند.

هر پردازنده با گره‌های خود به‌عنوان گره‌های‌محلی و سایر گره‌های اختصاصی به سایر پردازنده‌ها، شبکه خود را می‌سازد. گره‌های مربوط به سایر پردازنده‌ها گره‌های غیرمحلی نامیده می‌شوند. یک سمت یال‌های متصل به گره‌های غیرمحلی به گره محلی وصل هستند. هر پردازنده پس از ایجاد یک شبکه محلی، SLPA اصلاح‌شده را بر روی آن اجرا می‌کند. در پایان هر تکرار j ($j \leq t$) در SLPA هر پردازشگر p، فهرست برچسب گره‌های محلی خود را به دیگر پردازنده‌ها می‌فرستد به‌طوری‌که آن‌ها بتوانند گره‌های غیرمحلی خود را به‌روزرسانی کنند. همچنین پردازنده p، گره‌های محلی تکراری خود را از روی فهرست برچسب‌هایی که از سایر پردازنده‌ها دریافت کرده به‌روز می‌کند.

از کتابخانه افرازبندی زلتان[3] برای ایجاد تعادل بار افرازهای موازی استفاده می‌شود. n تعداد گره‌های شبکه‌محلی در یک پردازنده و $Labels_i$ فهرستی از برچسب‌های موجود توسط گره i است. نسخه موازی SLPA پیچیدگی خود را تغییر نداده لذا الگوریتم از نظر زمانی خطی باقی ماند. در پایان هر اجرا، محاسبه زمان اجرای کل و همچنین زمان سپری‌شده در تبادل پیام MPI اندازه‌گیری می‌شود. تسریع روش با رابطه (۶) و بازدهی آن با رابطه (۷) محاسبه می‌گردد. پس از آن‌که همه پردازنده‌ها، برچسب‌ها را تبادل کنند، اجرای تبادل برچسب اصلی به پایان می‌رسد. هر پردازنده بیشترین تکرار برچسب را در فهرست برچسب هر گره می‌یابد. این برچسب‌ها نماینده انجمن‌ها هستند و در یک فایل خروجی درج می‌شوند. از همه فایل‌های تولیدی توسط پردازنده‌ها، برچسب‌های منحصر به‌فرد استخراج و آن برچسب‌ها نماینده انجمن‌های هم‌پوشان هستند.

$$Speedup = \frac{T1}{Tn} \quad (۶)$$

$$Efficiency = \frac{Speedup}{p} \quad (۷)$$

الگوریتم موازی چندنخی SLPA: با استفاده از نخ‌ها، از هم‌زمان‌سازی انتظار-کار[4] استفاده می‌شود. هر نخ عمل انتشار برچسب‌های مربوط به زیرمجموعه گره‌های اختصاصی خود را انجام می‌دهد. برای تخصیص زیرگره‌ها به هر نخ باید شبکه به چند بخش شکسته شود. شکستن شبکه بر اساس توابع مختلف هدف دارای روش‌های متفاوت است. شکستن شبکه به‌روشی انجام می‌شود که هر نخ به میزان تقریباً یکسانی به پردازش نیاز داشته باشد. انتخاب برچسب برای هر گره با دریافت برچسب‌های همسایگان گره و انتخاب یک برچسب از بین آن‌ها انجام می‌شود. هر نخ، در هر مرحله همسایگان یکسانی را برای

[1] Message Passing Interface (MPI)
[2] Speed-up
[3] Zoltan Partitioning
[4] Busy-waiting

انتخاب برچسب پردازش می‌کند تا زمان یکسان برای پردازش تمام نخ‌ها حاصل شود. این‌گونه بخش‌بندی گراف در عمل به‌دلیل نامعین بودن زمان‌بندی نخ‌ها و تغییر در زمان شروع نخ‌ها کارآمد نیست. به‌عبارت دیگر، هم‌زمان‌سازی کامل نخ‌ها بعد از پردازش هر نخ، ضروری است و چنین همگام‌سازی با درنظر گرفتن محدودیت سرعت اجرا در گراف‌های واقعی، می‌تواند بهبود یابد. به‌جای چنین هم‌زمان‌سازی که موجب کاهش سرعت است، برای هر نخ جهت پردازش تعداد مساوی از همسایگان در هر گام ارسال می‌شود.

برای این کار نیاز است که جمع درجات ورودی گره‌های همسایه تمام نخ‌ها یکسان باشد و با این ترفند هر نخ تعداد گره مساوی را پردازش و جهت انجام کارش نیاز دارد. با درنظرگرفتن این مشکلات، هم‌زمان‌سازی نخ‌ها بعد از انجام هر تکرار کفایت می‌کند تا نخ‌ها در یک مرحله اجرا شوند. با تخصیص گره‌های محلی به هر نخ، امکان پردازش گره‌های همسایه در نخ‌های دیگر کمتر است. در صورتی که همسایه‌های یک گره در بیش از یک نخ پردازش شوند، وابستگی اجرای نخ‌ها در یک بردار نگهداری می‌شود تا در هم‌زمان‌سازی استفاده شود. هر نخ قبل از اجرا مطمئن است که بیش از یک مرحله جلوتر از عقب‌ترین نخ نیست.

الگوریتم موازی SLPA-OMP: کیاو در مقاله [۱۴] به موازی‌سازی الگوریتم گوینده-شنونده *SLPA-OMP*[1] پرداخته و با بهینه‌سازی محاسبات‌موازی و دسترسی به حافظه، به الگوریتم سریع‌تری دست‌یافت. او دریافت که تخصیص و آزادسازی حافظه، زمان‌بری زیادی دارد و همچنین اجرای دستورات شنونده دارای بیشترین زمان‌بری است. برای موازی‌سازی از ساختار openMP استفاده کرد که چندین گره به‌عنوان شنونده و هم‌زمان در حال اجرا داشت. به‌دلیل وجود همسایه‌های متفاوت، نخ‌های اجرایی در زمان‌های مختلف کار خود را به پایان می‌رسانند. الگوریتم مرتب‌سازی برای انتخاب بیشترین برچسب برای هر گره، از $O(mlogm)$ به $O(m)$ کاهش یافت. خطی‌بودن و قابلیت استفاده آن در گراف‌های بزرگ از مزایای الگوریتم، و بارگذاری کل گراف و نتایج میانی در حافظه از معایب اصلی آن است. شکال اساسی و کندی روش‌های قبلی در بارگذاری کل گراف در حافظه پردازنده است. برای مراحل مرتب‌سازی و محاسبات میانی الگوریتم‌ها، در گراف‌های بزرگ نیاز به رایانه‌های با حافظه بسیار زیاد (۳۲ تا ۶۴ گیگابایت) دارد.

[1] Open Multi-Processing (OMP)

جدول(۱): مقایسه الگوریتم‌های انتشار برچسب تشخیص انجمن.

الگوریتم	رویکرد و مشخصات
MMOCD[2] [۶]	خوشه‌بندی یال در سه سطح: گره، انجمن و شبکه
SCD [۳۰]	بهینه‌سازی معیار ضریب خوشه‌بندی
CDCFGI[3] [۳۱]	گروه‌های دوستی، بیشینه‌سازی مسیرهای مستقل بین گره‌ها
LPA [۷]	انتشار برچسب گره، غیرهم‌پوشان، نامعین
LPAm [۸]	انتشار برچسب گره، غیرهم‌پوشان، بهینه‌سازی پیمانه
LPAm+ [۹]	انتشار برچسب گره، غیرهم‌پوشان، بهینه‌سازی پیمانه، بیشینه سراسری
FUCLN[4] [۲۶]	بهینه‌سازی پیمانه، ترکیب انجمن
ELPA [۲۴]	انتشار برچسب یال، هم‌پوشانی کم، معین
COPRA[۲۰]	انتشار برچسب گره، هم‌پوشانی با پارامتر v
SLPA [۲۷]	انتشار برچسب گره، بدون پارامتر، هم‌پوشان، نامعین
SLPA-MT [۱۳]	انتشار برچسب گره، بدون پارامتر، هم‌پوشان، نامعین، چندنخی
SLPA-MPI [۱۳]	انتشار برچسب گره با تبادل پیام، بدون پارامتر، هم‌پوشان، نامعین، چند پردازشی، تبادل پیام همگام
POM-SLPA [۱۴]	انتشار برچسب گره، بدون پارامتر، هم‌پوشان، نامعین، چندنخی با **openMP**
Proposed SLPA-PP	انتشار برچسب‌فشاری‌وکششی: بدون پارامتر، نامعین، هم‌پوشان، هم‌پوشانی زیاد، چندنخی، **openMP**، تقلیل‌حافظه، بهینه‌سازی پیمانه، ماشین چندهسته‌ای

مقایسه الگوریتم‌های تشخیص‌انجمن با روش پیشنهادی در جدول (۱) ارائه شده است. الگوریتم‌های انتشار برچسب، خطی هستند و برای داده‌های بزرگ قابل کاربرد هستند. الگوریتم‌های انتشار برچسب SLPA بدون پارامتر و برای تشخیص انجمن‌های هم‌پوشان به‌کاربرده می‌شوند. جدیدترین الگوریتم‌های با مبنای SLPA که ارایه شده است [۳۵-۳۶]. این تحقیقات در جهت امتیازدهی گره‌ها و کاهش بحث انتخاب تصادفی اقداماتی کرده‌اند ولی برای مقایسه عادلانه هیچ‌گونه اطلاعات پیچیدگی زمان و حافظه مصرفی آن‌ها گزارش نشده است .

۳- الگوریتم پیشنهادی

الگوریتم پیشنهادی SLPA-PP بر اساس رویکرد گوینده-شنونده جهت تشخیص انجمن‌های هم‌پوشان با ترکیب ایده‌های جدید ارائه می‌شود که کارآمدتر از روش‌های قبلی SLPA است. ایده اصلی انتشار برچسب بر اساس الگوریتم گوینده- شنونده با الهام از "تعامل بین انسان‌ها و به خاطر سپردن پیام در صورت شنیدن بیش از یک بار از اطرافیان و سپس انتقال آن به دیگران" ارائه شده است. برای مرحله انتشار فشاری برچسب، مفهوم قطب‌محلی و ضریب برچسب تعریف شده است. برای انتشار بهینه برچسب در مرحله کششی مفهوم خط برگ و جذب خط‌برگ جهت بهبود پیمانه‌ای بودن نتایج انجمن‌ها ارایه شده است. الگوریتم جدیدی برای ترکیب انجمن‌های مشابه با معیار تشابه جاکارد دو انجمن ارائه شد. همچنین برای ارزیابی انجمن‌ها و افزایش میزان

[2] Modular Multiscale Approach to Overlapping Community Detection (MMOCD)
[3] Overlapping Community DetectionCollective Friendship Group Inference (CDCFGI)
[4] Fast Unfolding of Communities in Large Networks (FUCLN)

پیمانه‌ای‌بودن نتایج، معیار جدید پیمانه‌ای ارائه گردیده است. ماهیت الگوریتم پیشنهادی انتشار برچسب به‌گونه‌ای طراحی‌شده که امکان اجرای موازی آن میسر باشد.

با استفاده از الگوریتم اولیه انتشار برچسب که برای تشخیص انجمن‌های مجزا و با اصلاحاتی در اندازه حافظه نگهداری برچسب الگوریتم جدید طراحی و برای تشخیص انجمن‌های هم‌پوشان ارائه داده‌ایم. با الهام از ایده انتشار برچسب و ایده‌های جدید قطب‌های محلی و فشار برچسب قطب‌ها، الگوریتم پیشنهادی را کامل‌تر کرده‌ایم. این الگوریتم دارای چهار مرحله اصلی انتخاب قطب، فشار برچسب، کشش برچسب و ترکیب انجمن‌ها است. جزئیات مفهومی آن در شکل (۱) و الگوریتم آن در الگوریتم (۱) آمده و در ادامه به‌شرح آن می‌پردازیم.

انتخاب قطب: ابتدا گره‌های گراف درهم آمیخته می‌شوند. از هر k گره متوالی یکی از گره‌ها به‌عنوان قطب‌محلی انتخاب می‌شود. قطب‌محلی دارای بیشترین درجه در بین k گره است به شرطی که در همسایگانش گره‌ای با درجه بزرگ‌تر وجود نداشته باشد.

فشار برچسب: بعد از انتخاب قطب‌ها، مرحله فشار انتشار برچسب انجام می‌شود. در این مرحله، قطب‌ها نفوذ خود را به گره‌های دیگر اعمال می‌کنند. گره‌های قطب، برچسب خود را به همسایگان منتشر می‌کنند. ضریب انتشار برچسب قطب را برای جلوگیری از انتشار برچسب به قطب‌های هم‌جوار تعریف کرده‌ایم. در صورت کمتر بودن این ضریب از مقدار آستانه الگوریتم، برچسب قطب به همسایه منتشر می‌شود. نسبت درجه گره همسایه قطب به درجه قطب را ضریب انتشار قطب به همسایه می‌نامیم.

الگوریتم (۱): الگوریتم پیشنهادی SLPA-PP

Input: G (V, E), HPP, K, T, r
Output: C={c_i; U c_i=G}
1- For each n in nodes do
  Set unique label to n.memory
2- Shuffle network nodes
3- For each k sequence of nodes do
  Add node n to HubNodes where with degree of n max in k nodes and all neighbor's degree of n less than degree of n
4- Parallel for each node n in HubNodes do
  For all nb in neighbors of n do
  If degree(n)/degree(nb)<HPP then
  Push label of n to memory of nb
5- Parallel for i=1 to T do
  Shuffle network nodes
  Fore each n in nodes do
  If memory(n)<I then
  For each nb in neighbors of n do
  LabelQueue.add (random label from nb's memory)
  n.memory.add (LabelQueue.mustFrequent)
6- For each node n in network do
  If all labels in memory of n are less frequent than r then
  Remove all except greatest frequent labels of n.memory
  Else
  Remove labels of n.memory where frequents less than r
7- Remove label from node.memory where no path exists from node to any nodes that have the label in their memory
8- Remove small community that is sub community of larger one.
9- Merge each two communities where jaccard similarity of them greater than JACSIM

کشش برچسب: بعد از اعمال نفوذ گره‌های قطب در انتشار برچسب و انتقال برچسب‌شان به همسایه‌های خود، هر گره برچسب‌های موجود در همسایه‌هایش را به‌صورت تصادفی دریافت و بیشترین تکرار برچسب‌های دریافتی را به برچسب‌های حافظه خود اضافه می‌کند. در صورتی که چندین برچسب دریافتی دارای بیشینه تکرار باشند از میان آن‌ها یکی به‌صورت تصادفی انتخاب می‌کند. هر گره به تعداد برچسب دریافتی از مرحله فشاری، در انتشار برچسب شرکت نمی‌کند تا سایر گره‌ها که برچسب کمتری دارند به دریافت برچسب از همسایگان خود ادامه دهند. روش انتشار برچسب گوینده-شنونده فشاری کششی پیشنهادی در الگوریتم (۱) ارائه شده است.

ترکیب انجمن‌ها: در الگوریتم سنتی انتشار برچسب، بعد از اتمام مرحله انتشار برچسب، انجمن‌های یافت‌شده پردازش می‌شوند تا انجمن‌هایی که زیرمجموعه انجمن‌های دیگر هستند از لیست انجمن‌ها خارج شوند. در این مقاله با تعمیم این پردازش به محاسبه تشابه جاکارد دو انجمن و ترکیب انجمن‌هایی که تشابه جاکارد آن‌ها بیش از مقدار آستانه‌ای ورودی الگوریتم است نتایج انجمن‌های یافته‌شده را بهبود می‌دهیم.

معیارهای ارزیابی: دو معیار اصلی برای تعیین کیفیت انجمن‌های یافت‌شده در دو حالت از قبل شناخته‌بودن و نبودن انجمن‌ها استفاده می‌شود. از معیار NMI[1] برای داده‌گانی که انجمن‌های واقعی آن‌ها معلوم است استفاده می‌شود [۳۲] که با توجه به داده‌گان این مقاله از این معیار کاربرد ندارد. برای داده‌هایی که دارای انجمن‌های مشخص نیستند از شاخص M^{ov}[2] استفاده می‌شود [۳۳]. معمولاً ساختارهایی که دارای انجمن هستند معیار پیمانه‌ای بیشتری دارند. معیار پیمانه‌ای برای تعیین کیفیت انجمن‌های مجزا ارائه می‌شود و سپس برای انجمن‌های هم‌پوشان تعمیم می‌یابد. ایده معیار پیمانه‌ای، بر اساس بیشتر بودن تعداد یال‌های زیرگراف از تعداد یال‌های مرتب چینش‌شده تصادفی است. در انجمن‌های هم‌پوشان هر گره می‌تواند به چند انجمن تعلق داشته باشد لذا فهرستی از ضرایب‌تعلق به تعداد انجمن‌ها برای هر گره نگهداری می‌شود تا جمع این ضرایب برابر ۱ شود. رابطه (۸) محاسبه تابع پیمانه‌ای برای انجمن‌های هم‌پوشان را نشان می‌دهد. که در آن A ماتریس همسایگی گراف، r_{ijc} ضریب تعلق یال i به j در انجمن c، K_i^{Out} درجه خروجی گره i، K_j^{In} درجه ورودی گره j، s_{ijc} احتمال تعلق یال ورودی از i به j ضرب‌در احتمال تعلق یال خروجی از j به i در مدل پوچ مرجع[3] و m تعداد یال‌های گراف است .

$$Q_{ov} = \frac{1}{m}\sum_{c\in C}\sum_{i,j\in V}\left[r_{ijc}A_{ij} - s_{ijc}\frac{K_i^{Out}K_j^{In}}{m}\right] \qquad (۸)$$

[1] Normalized Mutual Information (MMI)
[2] Modularity Overlapping (MO)
[3] Reference Null Model (RNM)

پیچیدگی الگوریتم: مرحله مقداردهی اولیه از مرتبه $O(n)$ است که n تعداد گره‌های شبکه است. مرحله انتخاب قطب‌محلی برای هر گره با تعداد درجه آن مقایسه انجام می‌شود که از مرتبه $O(n\bar{k})$ که $\bar{k}$ میانگین درجه گره است به $O(m)$ تبدیل می‌شود. حلقه تکرار خارجی با مقدار ثابت و کوچک T کنترل می‌شود که برحسب تجربه آن را ۲۰ فرض می‌کنیم. حلقه داخلی برای هر گره یک‌بار با نقش شنونده $O(\bar{k})$ و یک‌بار با نقش گوینده $O(1)$ اجرا می‌شود و k همان درجه گره است. مرحله پس‌پردازش با توجه به حافظه هر گره به مقدار T از مرتبه $O(Tn)$ است و در نتیجه مرتبه الگوریتم $O(Tn\bar{k})$ یا $O(Tm)$ است که m تعداد یال‌های گراف می‌باشد.

۴- نتایج آزمایش‌های تجربی

در این بخش نتایج آزمایش‌های تجربی بر اساس داده‌های مورد استفاده و معیارهای استاندارد برای تحلیل و پیاده‌سازی ارائه می‌شود.

۴-۱- تنظیمات و مجموعه داده‌ها

سخت‌افزار مورد استفاده برای اجرای الگوریتم‌ها Core i3 با فرکانس $2.13GH$ دو هسته‌ای و چهار هسته منطقی، حافظه سریع سه‌سطحی $128KB$، $512KB$ و $3KB$، حافظه تصادفی DDR3 به میزان 6GB و سیستم عامل ویندوز ۱۰ با معماری ۶۴ بیتی است. برای ترجمه برنامه‌های زبان C++ به زبان ماشین از مترجم مایکروسافت و محیط برنامه‌نویسی ویژوال استودیو ۲۰۱۵ استفاده شده است. ترجمه با استفاده از فعال‌سازی openMP و بهینه‌سازی فلگ O3 همراه شده است. زبان برنامه‌نویسی سطح بالای C++ برای نوشتن برنامه انتخاب، ذخیره و بازیابی داده‌های گراف و نتایج در فایل سیستم صورت می‌گیرد. برای موازی‌سازی حلقه‌های اجرایی برنامه از کتابخانه OpenMP با پیاده‌سازی توسط مترجم C و FORTRAN استفاده و برای موازی‌سازی بهتر از کتابخانه امن و از نخ‌های TBB شرکت اینتل بهره‌برداری شده است.

از مجموعه داده‌های واقعی شبکه‌های اجتماعی [۳۴] برای آزمون و اجرای الگوریتم‌ها استفاده و اجرای الگوریتم‌ها بر گراف‌های بدون جهت و بدون وزن انجام شده است. گراف‌های داده‌ای با حذف یال‌های برگشتی به خود گره و افزودن یال‌های معکوس در فقدان یال در گراف اصلی، به گراف‌های بدون جهت تبدیل شده‌اند. خواص داده‌های مورد استفاده در جدول (۲) بیان‌شده و داده‌ها از شبکه‌های واقعی اجتماعی Live Journal و Twitter انتخاب‌شده است. تعداد گره‌های شبکه‌ها از ۴۰ هزار تا بیش از یک‌میلیون گره و تعداد یال‌ها از ۱۰۰ هزار تا ۱۰ میلیون یال است.

جدول (۲): مجموعه داده‌های آزمایشی

کد	نام گراف	تعداد گره	تعداد یال	میانگین درجه
UT5	youtube.top5000.cmty	۳۹۸۴۱	۱۲۱۵۰۴	۶/۱۰
UTA	youtube.all.cmty	۵۲۶۷۵	۱۸۹۱۷۲	۷/۱۸
SLD	Slashdot0902	۸۲۱۶۸	۱۰۰۸۴۶۰	۲۴/۵۵
LJ5	lj.top5000.cmty	۸۴۴۳۸	۲۰۱۰۵۸	۴/۷۶
UTU	youtubeungraph	۱۱۳۴۸۹۰	۵۹۷۵۲۴۸	۱۰/۵۳
LJA	ljallcmty	۱۱۴۷۹۴۸	۱۰۸۸۳۳۵۰	۱۸/۹۶

۴-۲- تحلیل آزمایش‌ها

در برنامه‌سازی موازی برای دست‌یابی به کارایی مناسب عوامل بسیاری از قبیل الگوریتم پیشنهادی، معیارهای اندازه‌گیری، روش‌های پیاده‌سازی و تنظیمات آن‌ها مؤثر هستند که در ادامه به آن‌ها و اثراتشان می‌پردازیم.

۴-۲-۱- مقایسه با روش سنتی SLPA

تنظیمات مختلف برای پارامترهای رزرو و ضریب درهم‌ساز برای روش پیشنهادی در مقایسه با روش سنتی درنظر گرفته شده که نتایج اجرای آن در جدول (۳) به ثانیه آمده است. اجرای الگوریتم با ۴ نخ، آستانه انتخاب برچسب ۰/۳ و در ۲۰ تکرار انجام شده است. با بررسی نتایج مشاهده شد که ساختار درهم‌ساز برای گراف‌های بزرگ، با مقدار رزرو اولیه ۸ و برای گراف‌های کوچک بدون رزرو با سرعت بیشتری اجرا می‌شود. در اجراهای مختلف، تمام پیاده‌سازی‌های پیشنهادی در بدترین حالت بهتر از روش سنتی است.

جدول (۳): مقایسه زمان اجرای SLPA سنتی و روش پیشنهادی با بهبود جدول hash (ثانیه)

روش	Hash parameters		داده		
			lja	Utu	sld
سنتی	ندارد		۵۱/۱۸	۳۰/۱۷	۳/٤۷
	Reserve	Fill Factor			
پیشنهادی	۰	۰	۴۰/۷۸	۲۴/۷۵	۲/۲۸
	۸	۰/۲۵	۳۶/۳۵	۲۴/۹۶	۲/۸۱
	۱۶	۰/۲۵	۳۹/۷۸	۲۵/۵۷	۲/۴۱
	۳۲	۰/۲۵	۴۰/۲۴	۲۵/۶۷	۲/۷۱

۴-۲-۲- روش‌های موازی‌سازی

نتایج آزمایش روش پیشنهادی و روش‌های پیاده‌سازی آن از تنظیم تعداد نخ‌ها تا بررسی معیارهای پیشنهادی در زیر ارائه‌شده است.

تنظیم تعداد نخ‌ها: با پیاده‌سازی موازی به‌صورت چندنخی[1] (از ۱ نخ تا ۱۶ نخ)، نتایج نشان می‌دهد که تأثیر تعداد نخ‌ها بر سرعت زیاد است و بهترین زمان اجرا در ۴ نخ به‌دست می‌آید. این تعداد با تعداد پردازنده‌های منطقی رایانه برابر می‌باشد.

[1] Multi-threading

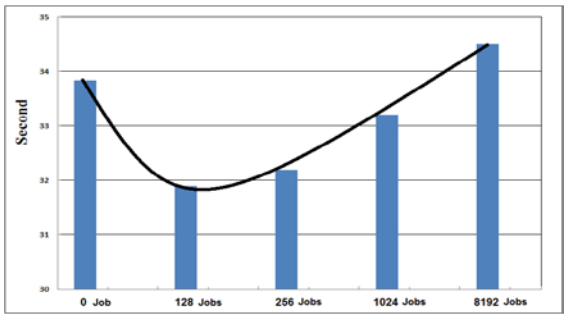
زمان‌بندی اجرای نخ‌ها: با بررسی سه روش زمان‌بندی اجرای نخ‌ها در openMP را به همراه تنظیمات مختلف و زمان‌بندی‌های ایستا، پویا و هدایت‌شده (با مقادیر اولیه مشخص با تعداد [1]کار اولیه) در نتایج بهبود مناسب مشاهده شد. در بیشتر حالت‌ها، زمان‌بندی پویا بر زمان‌بندی‌های دیگر برتری دارد و زمان‌بندی پویا در تمام حالات بهتر از زمان‌بندی ایستا عمل می‌کند. این امر به‌دلیل محاسبات نامتقارن بین گره‌های متفاوت است و هزینه تعویض نخ به‌صورت پویا کمتر از زمان انتظار اجرای نخ‌ها به‌صورت ایستا است. زمان‌بندی پویا با مقدار اولیه تعداد کار ۱۲۸ و ۲۵۶ بهتر از زمان‌بندی پویای خودکار عمل می‌کند. در شبکه بزرگ LJA زمان‌بندی هدایت‌شده با مقدار اولیه ۱۰۲۴ کار، جواب بهینه دارد و تنها در یک مجموعه داده بهتر عمل می‌کند. نتایج زمان اجرای الگوریتم انتشار برچسب با تخصیص پویا و ایستای زمان‌بندی نخ‌ها در جدول (۴) با ۱۰حلقه تکرار آمده است. در نتایج مشهود است که زمان‌بندی‌پویا در تمام حالات بهتر از زمان‌بندی ایستا عمل می‌کند. این امر به‌دلیل محاسبات نامتقارن بین گره‌های متفاوت است و هزینه کم تعویض نخ‌ها است. در شبکه‌های بزرگ LJa و UTU اجرای الگوریتم با زمان‌بندی پویای ۱۲۸تایی به ترتیب ۵٪ و ۶٪ بهبود در اجرای الگوریتم نشان می‌دهد.

جدول (۴): مقایسه زمان اجرای الگوریتم پیشنهادی، زمان‌بندی‌های پویا، ثابت و دستی (ثانیه)

روش انتشار	ضریب انتشار قطب		
	۱	۰،۹	۰،۴
بی جذب خط برگ	۰،۱۳	۰،۱۸	۰،۲۹
با جذب خط برگ	۰،۱۴	۰،۳۶	۰،۲۷

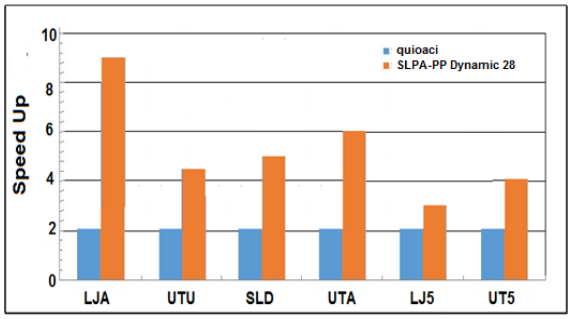
نتایج پیاده‌سازی نشان داد که می‌توان با تغییر تنظیمات پیش‌فرض زمان‌بندی اجرای نخ‌های پویا، حداقل ۵٪ بهبود در اجرای الگوریتم در داده‌های بزرگ ایجاد کرد. در بدترین حالت‌ها نیز با ثابت بودن زمان‌بندی پویا در اندازه‌های ۱۲۸یا ۲۶۵ کار نتیجه‌ای بهتر از حالت پیش¬فرض در مقایسه با پیاده‌سازی کیاوسی حاصل می‌شود. با بررسی نتایج زمان‌بندی هدایت‌شده در شکل (۳) نشان می‌دهد که این روش تنها در یک حالت (۱۲۸کار) برای بزرگ‌ترین گراف‌ها مجموعه LJA و UTU دارای کارایی بهتر نسبت به سایر زمان‌بندی‌های اجرای نخ دارد. با افزایش تعداد کارها به بیش از ۱۲۸ دارای رشد خطی است.

[1] Job



شکل (۳): تأثیر تعداد کارها در زمان‌بندی پویا

ضریب انتشار قطب: با مقایسه ضریب انتشار قطب با مقدار ورودی الگوریتم، در صورت کوچک‌تر بودن این ضریب از مقدار آستانه، گره قطب برچسب خود را به همسایه به‌صورت فشاری منتقل و در غیر آن منتقل نخواهد کرد. بررسی اجراها برای داده‌های مختلف نشان می‌دهد که ضریب بین ٪۶۰ تا ٪۸۰ دارای نتایج بهتری نسبت به انتشار اجباری به همه همسایگان است. با کمتر شدن ضریب انتشار قطب از ۱۰۰ به ۸۰، ضریب نفوذ قطب به گره‌های شبه‌قطب کمتر و از انتشار بیش از حد برچسب در گره‌های دیگر جلوگیری می‌شود. بهبود نتایج حاصل نسبت به پیاده‌سازی قبلی توسط کیاوسی در شکل (۴) نشان می‌دهد که در گراف‌های بزرگ ۴ تا ۹ برابر نسبت به پیاده‌سازی قبلی تسریع ایجاد شده است.



شکل (۴): میزان تسریع روش پیشنهادی نسبت به‌روش سنتی

بهبود معیار پیمانه‌ای: با درج تأثیر معیارهای جذب خط برگ، دسته انتخاب قطب‌محلی، ضریب‌انتشار و تشابه جاکارد پیشنهادی برای ترکیب انجمن‌ها بر معیار پیمانه‌ای، نتایج پیمانه‌ای بهینه و معمولی حاصل از اجرای الگوریتم انتشار برچسب با در نظر گرفتن دو قطب انتشار برای داده زاکاری در جدول (۵) نشان داده‌شده است. نتایج گویای آن است که در هر دو حالت جذب‌شدن و نشدن خط‌برگ، ضریب انتشار کمتر از ۱۰۰ تأثیر زیادی در بهبود معیار پیمانه‌ای دارد ولی مقدار بیشینه پیمانه‌ای در حالت جذب خط‌برگ به‌دست آمده است. نتایج تأثیر ضریب‌انتشار قطب برای داده زاکاری در شکل (۶) نشان داده‌شده است. تابع پیمانه‌ای بیشینه برای ضرایب‌انتشار ۰/۶ و ۰/۷ اتفاق

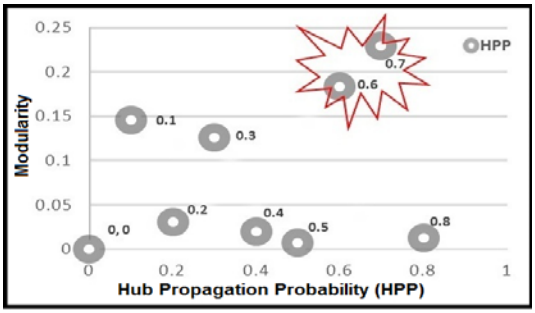
افتاده است. همچنین در جدول (۶) تأثیر ترکیبی ضریب‌انتشار را با دسته انتخاب قطب به تفکیک جذب‌شدن و نشدن خط‌برگ برای داده Utu بیان‌شده است. در این گراف بیشینه تابع پیمانه‌ای در حالت بدون جذب خط برگ اتفاق افتاده است. با این حال مشاهده می‌شود که جذب خط‌برگ تأثیر بیشتری در بهبود ضرایب انتشار به تابع پیمانه‌ای داشته است.

جدول (۵): پیمانه نیکوشیا در داده زاکاری با ضریب انتشار قطب بهینه

Method	Batch Size	Ut5	Lj5	Uta	Sld	Utu	Ija
Dynamic	Auto	۰,۵۴	۰,۹۲	۱,۴۲	۳,۲۵	۲۷,۵۳	۳۹,۳۳
	۱۲۸	۰,۵۴	۰,۹۰	۱,۳۹	۳,۲۲	**۲۵,۶۶**	۳۷,۳۱
	۲۵۶	**۰,۵۴**	**۰,۸۹**	۱,۳۶	**۳,۲۱**	۲۶,۲۶	۳۷,۸۷
	۱۰۲۴	۰,۶۲	۱,۱۷	**۰,۹۸**	۳,۴۷	۲۶,۹۵	۴۰,۲۶
	۸۱۹۲	۰,۸۳	۱,۲۹	۱,۰۹	۳,۹۴	۲۱,۲۷	**۳۸,۵۰**
Static	Auto	۰,۵۷	۰,۹۴	۱,۴۷	۳,۵۷	۲۸,۴۷	۷۲,۳
	۱۲۸	۰,۵۶	۰,۹۳	۱,۵	۳,۶۴	۲۸,۷۲	۷۲,۸
	۲۵۶	۰,۵۹	۰,۹۹	۱,۳۹	۳,۶۹	۲۸,۳۴	۶۵
	۱۰۲۴	۰,۷۶	۱,۱۷	۱,۰۶	۳,۴۶	۲۹,۳۹	۳۹,۷۶
	۸۱۹۲	۰,۹۷	۱,۳۸	۱,۱۸	۴,۰۷	۲۷,۹۹	۴۷,۲۴
Guided	۱۰۲۴	۰,۵۷	۰,۹۰	۱,۴۲	۳,۲۶	۳۴,۲۸	**۳۵,۳۶**

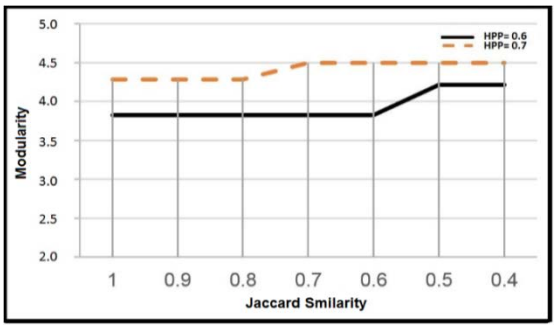
جدول (۶): پیمانه نیکوشیا در UT5 با دسته قطب و ضرایب انتشار مختلف.

روش انتشار	دسته قطب	ضریب انتشار قطب				
		۱	۰,۹	۰,۸	۰,۷	۰,۶
بی‌جذب خط برگ	۰	۰,۲۷۳	۰,۲۶۱	۰,۲۷۵	۰,۲۷۰	۰,۲۵۷
	۱۰۲۴	۰,۲۷۴	۰,۲۷۹	۰,۲۸۱	۰,۲۶۹	۰,۲۸۵
با جذب خط برگ	۰	۰,۲۵۷	۰,۲۵۶	۰,۲۷۷	۰,۲۶۳	۰,۲۵۶
	۱۰۲۴	۰,۲۶۹	۰,۲۶۲	۰,۲۸۳	۰,۲۷۲	۰,۲۸۴



شکل(۵): تأثیر ضریب انتشار بر تابع پیمانه‌ای در داده زاکاری.

ترکیب انجمن‌ها: نسبت تعداد گره‌های مشترک دو انجمن به مجموع گره‌های دو انجمن را تشابه جاکارد دو انجمن است. هر چه اندازه دو انجمن به هم نزدیک باشد و تعداد گره‌های مشترک آن‌ها بیشتر باشد مقدار تشابه به ۱ نزدیک‌تر است. در الگوریتم‌های سنتی انتشار برچسب گوینده شنونده، بعد از تشخیص انجمن‌ها، زیرمجموعه بودن انجمن‌های یافت‌شده مورد بررسی قرار می‌گیرد و در صورت زیرمجموعه کامل‌بودن، انجمن کوچک‌تر حذف می‌شود. ما با تعریف ضریب تشابه جاکارد، عملیات زیرمجموعه را به عملیات مشابه‌بودن تغییر داده‌ایم. در صورتی که تشابه جاکارد دو انجمن بیش از مقدار ورودی الگوریتم باشد، دو انجمن در هم ادغام می‌شوند. تنظیمات مختلف برای ترکیب انجمن‌ها با تشابه جاکارد از ۵۰٪ تا ۹۰٪ برای داده زاکاری بررسی شد که ترکیب انجمن‌ها در صورت تشابه بیش از ۷۰٪ بهتر از سایر حالت‌ها به بهترشان کیفیت انجمن‌ها کمک کرد. با یافتن حالت بهینه ضریب‌انتشار ۰/۶ و ۰/۷ برای داده زاکاری به بررسی تأثیر ضریب تشابه جاکارد به تابع پیمانه‌ای پرداخته شد که در شکل (۶) نشان داده‌شده است. طبق داده‌های این شکل بیشترین تأثیر ضریب جاکارد، در ترکیب انجمن‌هایی است که تشابه آن‌ها بیش از ۶۰٪ باشد. هرچه آستانه ضریب ترکیب انجمن‌ها کمتر انتخاب شود انجمن‌های مستقل‌تر با هم ترکیب خواهند شد.



شکل(۶): تأثیر ضریب تشابه جاکارد بر تابع پیمانه‌ای در داده زاکاری.

۴-۳- ارزیابی روش پیشنهادی

در الگوریتم گوینده- شنونده سنتی برای انتخاب بیشینه تکرار برچسب‌های دریافتی، عمل مرتب‌سازی انجام می‌شود که با مقایسه و نگهداری بیشینه برچسب دریافتی، جایگزین شده و عمل مرتب‌سازی را با یافتن بیشینه آن در زمان دریافت برچسب، تغییر دادیم. برای موازی‌سازی بیشتر و اخذ نتایج بهتر از ساختار امن درهم‌ساز اینتل استفاده و در موازی‌سازی حلقه‌های برنامه openMP از زمان‌بندی پویای ۱۲۸کار بهره بردیم. ایده جدید فشار برچسب از سمت قطب‌ها به همسایگانشان را در الگوریتم گوینده-شنونده اضافه کردیم. نکته کلیدی ایده این الگوریتم، اعمال نفوذ گره‌های شبه‌قطب در حلقه اول انتشار برچسب به همسایگان خود است. نتایج نشان می‌دهد که این ایده در سرعت اجرای الگوریتم و بهبود نتایج تشخیص انجمن، افزایش سرعت بهتری از روش‌های قبلی ارائه داده است.

برای تنظیم رفتار انتشار برچسب قطب‌ها و همچنین انتخاب قطب‌ها راه‌کاری ارائه دادیم. انتخاب قطب‌ها خیلی دقیق نیست و با پیمایش تعداد دسته‌ای از گره‌ها به‌عنوان ورودی الگوریتم صورت می‌گیرد. این گره در بین دسته خود (نه کل گراف)

بیشینه درجه را دارد که در آن را قطب‌محلی نامیدیم. مفهومی به‌نام ضریب‌انتشار قطب (نسبت درجه گره همسایه به درجه گره قطب) را تعریف و ارائه دادیم. با بررسی انتشار قطب و تنظیمات مختلف آستانه انتشار ضریب قطب، نتایج تجربی نشان داد با ضریب انتشار قطب برابر۰/۶، مقدار بهینه بیشینه در تابع پیمانه‌ای رخ می‌دهد. سپس برای بهبود کیفیت انجمن‌های یافت‌شده مفهوم خطبرگ را تعریف شد. گره‌های انتهایی گراف که به‌صورت درختی بی‌برگ منتهی می‌شوند را شناسایی و با جذب گره‌های خطبرگ قبل از انتشار برچسب، به کیفیت بهتری از تابع پیمانه‌ای در بیشتر داده‌ها دست یافتیم. سپس برای تشخیص کیفیت انجمن‌های یافت‌شده در داده‌هایی که انجمن‌های واقعی آن‌ها مشخص نیست از تابع پیمانه‌ای نیکوشیا (تعمیم تابع پیمانه‌ای برای انجمن‌های هم‌پوشان) استفاده کردیم.

با این انتخاب نتایج کیفیت انجمن‌های بیشتر از الگوریتم‌های سنتی است. برای بهبود نتایج، مفهوم جدیدی به‌نام تشابه جاکارد دو انجمن (نسبت تعداد گره‌های مشترک دو انجمن به جمع تعداد گره‌های دو انجمن) تعریف کردیم. در الگوریتم پیشنهادی، پارامتری جهت آستانه ترکیب انجمن‌ها ارائه شد و از آن برای ترکیب انجمن‌هایی که تشابه جاکاردی آن‌ها بیشتر از مقدار ورودی الگوریتم باشد بهره بردیم. نتایج تجربی نشان داد که با ترکیب انجمن‌هایی که ضریب جاکارد آن‌ها بیشتر از ۰/۶ باشد نتایج بهتری در کیفیت انجمن‌های تشخیصی به وجود می‌آید. در نهایت با کاهش پیچیدگی زمانی و حافظه‌ای با افزایش حجم شبکه به بهبود خطی پیچیدگی زمانی الگوریتم و حافظه دست یافتیم.

۵- نتیجه‌گیری

در این مقاله، روش توزیعی مقیاس‌پذیر تشخیص انجمن‌های هم‌پوشان بر اساس انتشار برچسب بر بستر موازی چندنخی در سیستم چندهسته‌ای ارائه و بر روی میلیون‌ها گره و اتصال ارزیابی شد. زمان بارگذاری شبکه و پردازش آن بسیار کاهش یافت و با تعریف معیارهای جدید، تسریع ارزشمندی در تشخیص سریع و دقیق ایجاد شد. با استفاده مرحله‌ای از اطلاعات محلیِ (گره و همسایگان) و اطلاعات سراسریِ (انجمن و گراف)، همچنین هم‌زمانی جستجوها، پیاده‌سازی موازی و زمان‌بندی تسک‌های مهم، ضمن داشتن دقت زیاد تشخیص انجمن‌ها با قابلیت پیمانه‌ای بالا، از قدرت بالای کنترل بر جلوگیری از تکرار و پخش توالی برچسب‌ها برخوردار است. با بزرگ‌شدن شبکه از زمان و حافظه مصرفی خطی برخوردار شد. چالش اصلی در ارجاع تصادفی به یک گره در هر مرحله و سپس ارجاع تصادفی به برچسب‌های دریافتی آن گره است. در صورت عدم بارگذاری کل گراف در حافظه با مشکل کمبود حافظه اصلی با حافظه‌ثانویه مواجه و سرعت اجرای الگوریتم به‌شدت پایین می‌آید. در کارهای آینده ضمن تلاش برای رفع چالش فوق، الگوریتم پیشنهادی گوینده شنونده را بر بستر OPEN-MPI با افزودن تابع پیمانه‌ای نیکوشیا پیاده‌سازی خواهیم کرد. همچنین روش مکاشفه‌ای هدایت انتشار برچسب در حلقه‌های انتشار ارائه خواهیم کرد تا به بهبود تابع پیمانه‌ای بیانجامد. سپس با شکستن گراف شبکه به زیرگراف‌های کاملاً مستقل و اجرای موازی عملیات انتشار و تشخیص محلی بر پایه اطلاعات سراسری به توزیع داده‌ها و بخش‌بندی حافظه برای تسریع بیشتر کمک خواهیم کرد.

۶- مراجع

[1] T. Aynaud and J. L. Guillaume, "Multi-step community detection and hierarchical time segmentation in evolving networks," Proceedings of the 5th SNA-KDD workshop, 2011.

[2] W. M. Campbell, C. K. Dagli, and C. J. Weinstein, "Social Network Analysis with Content and Graphs," Lincoln Laboratory Journal, vol. 20, no. 1, 2013.

[3] Y.-Y. Ahn, J. P. Bagrow, and S. J. n. Lehmann, "Link communities reveal multiscale complexity in networks," vol. 466, no. 7307, p. 761, 2010.

[4] L. Tang and H. Liu, "Community detection and mining in social media," Synthesis Lectures on Data Mining and Knowledge Discovery, vol. 2, no. 1, pp. 130-137, 2010.

[5] [5] Y. Cohen, D. Hendler, and A. Rubin, "Node-centric detection of overlapping communities in social networks," International Conference and School on Network Science, pp. 1-10, 2017.

[6] [6] M. Brutz and F. G. Meyer, "A Modular Multiscale Approach to Overlapping Community Detection," arXiv preprint arXiv: 1501.05623, 2015.

[7] [7] U. N. Raghavan, R. Albert, and S. J. P. E. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," Phys. Rev. E., vol. 76, no. 3, p. 036106, Sep 2007.

[8] [8] M. J. Barber and J. W. J. P. R. E. Clark, "Detecting network communities by propagating labels under constraints," Phys. Rev. E., vol. 80, no. 2, p. 026129, Aug. 2009.

[9] [9] X. Liu, T. J. P. A. S. M. Murata, and I. Applications, "Advanced modularity-specialized label propagation algorithm for detecting communities in networks," Physica A: Statistical Mechanics and its Applications, vol. 389, no. 7, pp. 1493-1500, 2010.

[10] C. L. Staudt and H. Meyerhenke, "Engineering parallel algorithms for community detection in massive networks," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 1, pp. 171-184, 2016.

[11] S. Moon, J.-G. Lee, M. Kang, M. Choy, and J. w. Lee, "Parallel community detection on large graphs with

Transactions on Knowledge and Data Engineering, vol. 28, no. 5, pp. 1272-1284, 2016.

[26] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," Journal of Statistical Mechanics: Theory and Experiment, vol. 2008, no. 10, p. P10008, 2008.

[27] A. Hollmann and et al., "Tight controlled expression and secretion of Lactobacillus brevis SlpA in Lactococcus lactis," Biotechnol Letter, vol. 34, no.7, pp. 1275-81, 2012.

[28] T. Wang, X. Qian, and X. Wang, "HLPA: A hybrid label propagation algorithm to find communities in large-scale networks," 7th International Conference on Awareness Science and Technology (iCAST) IEEE, pp. 135-140, 2015.

[29] Q. Zhang, Q. Qiu, W. Guo, K. Guo, and N. J. C. N. Xiong, "A social community detection algorithm based on parallel grey label propagation," Computer Networks, vol. 107, pp. 133-143, 2016.

[30] A. Prat-Perez, D. Dominguez-Sal, and J. L. Larriba-Pey, "High quality, scalable and parallel community detection for large real graphs," 23rd International Conference on World Wide Web, Seoul, Korea, pp. 225-236, 2014.

[31] B. S. Rees and K. B. Gallagher, "Overlapping community detection by collective friendship group inference," International Conference on Advances in Social Networks Analysis and Mining IEEE, pp. 375-379, 2010.

[32] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," Physical Review E, vol. 80, no. 5, pp. 056117, Nov. 2009.

[33] V. Nicosia, G. Mangioni, V. Carchiolo, and M. J. J. O. S. M. T. Malgeri, and Experiment, "Extending the definition of modularity to directed graphs with overlapping communities," Journal of Statistical Mechanics: Theory and Experiment, vol. 03, p. P03024, 2009.

[34] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection, June 2014," URL: http://snap. stanford. edu/data, 2014.

[35] IB. El Kouni, W. Karoui, LB. Romhdane, "Node Importance based Label Propagation Algorithm for overlapping community detection in networks," Expert Systems with Applications, pp.113020, 2019.

[36] D. Jin, B. Li, P. Jiao, D. He, H. Shan, W. Zhang, "Modeling with Node Popularities for Autonomous Overlapping Community Detection," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 11, no. 3, pp.1-23, 2020.

MapReduce and GraphChi," Data & Knowledge Engineering, vol. 104, pp. 17-34, 2016.

[12] C. Li, Y. Tang, H. Lin, C. Yuan, and H. Mai, "Parallel overlapping community detection algorithm in complex networks based on label propagation," SCIENTIA SINICA Informationis, vol. 46, no. 2, pp. 212-227, 2016.

[13] K. Kuzmin, S. Y. Shah, and B. K. Szymanski, "Parallel overlapping community detection with SLPA," International Conference on Social Computing. IEEE, pp. 204-212, 2013.

[14] Q. Yuchen, W. Haixia, and W. Dongsheng, "Parallelizing and optimizing overlapping community detection with speaker-listener Label Propagation Algorithm on multi-core architecture," IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), pp. 439-443, 2017.

[15] A. Kukanov and M. J. J. I. T. J. Voss, "The Foundations for Scalable Multi-Core Software in Intel Threading Building Blocks," Intel Technology Journal, vol. 11, no. 4, 2007.

[16] S. Kim, "Community Detection in Directed Networks and its Application to Analysis of Social Networks," Dissertation, The Ohio State University, 2014.

[17] S. Fortunato, "Community detection in graphs," Physics Reports, vol. 486, no. 3-5, pp. 75-174, 2010.

[18] J. Su and T. C. Havens, "Fuzzy community detection in social networks using a genetic algortihm," International Conference on Fuzzy Systems IEEE, pp. 2039-2046, 2014.

[19] B. Adamcsek, G. Palla, I. J. Farkas, I. Derenyi, and T. Vicsek, "CFinder: locating cliques and overlapping modules in biological networks," Bioinformatics, vol. 22, no. 8, pp. 1021-1023, 2006.

[20] S. Gregory," Finding overlapping communities in networks by label propagation," New Journal of Physics, vol. 12, no. 10, p. 103018, 2010.

[21] S. Gregory, "An algorithm to find overlapping community structure in networks," European Conference on Principles of Data Mining and Knowledge Discovery, pp. 91-102, Springer 2007.

[22] H. Shen, X. Cheng, K. Cai, and M. B. Hu," Detect overlapping and hierarchical community structure in networks," Physica A: Statistical Mechanics and its Applications, vol. 388, no. 8, pp. 1706-1712, 2009.

[23] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," Knowledge and Information Systems, vol. 42, no. 1, pp. 181-213, 2013.

[24] W. Liu, X. Jiang, M. Pellegrini, and X. J. S. r. Wang, "Discovering communities in complex networks by edge label propagation," Scientific Reports 6.1, vol. 6, no. 1, pp. 1-10, 2016.

[25] J. J. Whang, D. F. Gleich, and I. S. Dhillon, "Overlapping community detection using neighborhood-inflated seed expansion," IEEE

Journal of Electronical & Cyber Defence
Vol. 8, No. 4, 2021, Serial No. 32

A Distributed Approach to Community Detection in Large Social Networks Based on Label Propagation

M. Hosseini, A. Mahabadi*

*Shahed University

(Received: 03/06/2019, Accepted: 05/08/2020)

ABSTRACT

Detection of overlapping communities in large complex social networks with intelligent agents, is an NP problem with great time complexity and large memory usage and no simultaneous online solution. Proposing a novel distributed label propagation approach can help to decrease the searching time and reduce the memory space usage. This paper presents a scalable distributed overlapping community detection approach based on the label propagation method by proposing a novel algorithm and three new metrics to expand scalability and improve modularity through agent-based implementation and good memory allocation in a multi-core architecture. The experimental results of large real datasets over the state-of-the-art SLPA approach show that the execution time speeds up by 900% and the modularity improves by 3% to 100% thus producing fast and accurate detection of overlapped communities.

Keywords: Social Networks, Distributed Processing, Overlapping Community Detection, Label Propagation Algorithm

* Corresponding Author Email: mahabadi@shahed.ac.ir