

معیاری جدید برای بخش‌بندی سیستم‌های پردازش گراف مبتنی بر بلوک

مسعود ساغریچیان و مرتضی علی‌پور لنگوری

موجود در این شبکه‌ها، نظریه گراف‌ها به عنوان ابزاری اساسی جهت نمایش و پردازش چنین شبکه‌هایی مورد استفاده قرار گرفته است. اغلب گراف‌هایی که با آنها روبه‌رو هستیم عمدتاً بسیار بزرگ هستند. به عنوان مثال گراف شبکه اجتماعی فیس‌بوک طبق گزارش سال ۲۰۲۱ دارای ۲/۸۹ میلیارد گره بوده است.

در راستای تحلیل چنین گراف‌های بزرگی، سیستم‌های پردازش گراف مختلفی توسعه داده شده است. برخی از این سیستم‌ها به صورت متمرکز و روی یک ماشین، فرایند پردازش را انجام می‌دهند [۴] و [۵]، اما درصد قابل توجهی از سیستم‌های پردازش گراف از رویکرد توزیع‌شدگی بهره گرفته‌اند [۶] تا [۱۰]. سیستم‌های پردازش گراف مبتنی بر بلوک نسبت به سایر روش‌های پردازش گراف توزیع‌شده، کارایی بهتری را از حیث میزان ترافیک و همچنین تعداد گلوگاه‌های همگام‌سازی از خود نشان داده‌اند [۱۱] تا [۱۶]. به منظور پردازش توزیع‌شده، ابتدا باید گراف به قسمت‌هایی بخش‌بندی شود و هر بخش توسط یک ماشین در سیستم توزیع‌شده مورد پردازش قرار گیرد. با توجه به توزیع یال‌ها در این گراف‌ها و همچنین ارتباطات زیاد بین گره‌ها، بخش‌بندی گراف به صورت تصادفی منجر به تولید پارتیشن‌هایی می‌شود که ارتباط بین گره‌های پارتیشن‌های مختلف بسیار زیاد است. در نتیجه یک بخش‌بند گراف باید بتواند به دو چالش اساسی توجه نماید: (۱) کمینه‌سازی تعداد یال‌های برشی و (۲) بیشینه‌کردن تعادل بار بین پارتیشن‌ها. اغلب روش‌های ارائه‌شده برای بخش‌بندی گراف تنها به این دو معیار توجه نموده‌اند و سعی کرده‌اند که بخش‌بندی نهایی را با این دو معیار مورد ارزیابی قرار دهند. از سوی دیگر، فرایند بخش‌بندی باید بر روی گراف اصلی که گراف بزرگی است، اعمال شود. برخی از روش‌ها نظیر «متیس» و توسعه‌های آن، فرایند بخش‌بندی را به صورت متمرکز با استفاده از یک ماشین انجام می‌دهند [۱۷] تا [۲۰]. مزیت این دسته از روش‌ها آن است که چون الگوریتم، دانش کاملی از کل گراف را دارا می‌باشد، عمدتاً کیفیت بخش‌بندی تولیدشده بسیار بهتر است. مشکل این دسته از روش‌ها آن است که برای پردازش گراف‌های بزرگ، نیازمند ماشینی با منابع محاسباتی بسیار بالا هستند. همچنین زمان اجرای بالا هم از دیگر نقاط ضعف آنها است. برای غلبه بر این مشکل روش‌های دیگری فرایند بخش‌بندی را به صورت توزیع‌شده انجام می‌دهند [۲۱] تا [۲۴].

در این مقاله نشان داده شده که معیارهای سنتی مورد استفاده در بخش‌بندی گراف برای سیستم‌های پردازش گراف مبتنی بر بلوک لازم است، اما کافی نیست. هدف اصلی این مقاله، استخراج معیارهایی پایه‌ای است که می‌تواند کارایی این دسته از سیستم‌ها را بهبود بخشد. برای نیل به این هدف، ابتدا ویژگی‌های مشترک سیستم‌های پردازش گراف مبتنی بر بلوک تبیین شده است. بر اساس این ویژگی‌ها، دو معیار جدید که تأثیر

چکیده: به واسطه قدرت و سادگی، سیستم‌های پردازش گراف مبتنی بر بلوک در سال‌های اخیر مورد توجه ویژه‌ای قرار گرفته‌اند. اغلب این سیستم‌ها از روش‌های بخش‌بندی عمومی و همه‌منظوره جهت تولید پارتیشن‌های مورد نیاز خود استفاده می‌کنند. همین امر منجر شده که کارایی این سیستم‌ها محدود شود. برای رفع این مشکل الگوریتم‌های خاص‌منظوره‌ای برای بخش‌بندی این دسته از سیستم‌ها ارائه شده است، اما مشکل این دسته از روش‌ها آن است که همچنان معیارهای سنتی نظیر تعداد یال برشی و تعادل بار به عنوان تابع هدف این روش‌ها مد نظر قرار گرفته است. این در حالی است که قدرت سیستم‌های پردازش گراف مبتنی بر بلوک به واسطه ویژگی‌های منحصر به فردی است که در طراحی این دسته از سیستم‌ها مد نظر قرار گرفته است. به همین جهت در این مقاله، ویژگی‌های ذاتی و اساسی این دسته از سیستم‌ها مورد توجه قرار گرفته و با توجه به این خواص، دو معیار جدید به عنوان معیار تابع هدف بخش‌بندی، معرفی شده است. بر اساس تحقیقات انجام‌گرفته، روش پیشنهادی اولین الگوریتم بخش‌بندی است که قطر گراف سطح بالا و اندازه گره‌های گراف سطح بالای حاصل از بخش‌بندی را به عنوان تابع هدف در نظر گرفته می‌گیرد. ارزیابی روش پیشنهادی بر روی مجموعه داده‌های واقعی نشان داد که روش پیشنهادی به طور مؤثری قادر به کاهش قطر گراف سطح بالای حاصل از بخش‌بندی نسبت به سایر الگوریتم‌های بخش‌بندی متداول می‌باشد. به علاوه، یال برشی حاصل از روش پیشنهادی بسیار نزدیک به یکی از معروف‌ترین روش‌های بخش‌بندی متمرکز، متیس می‌باشد. از آنجا که قطر گراف سطح بالا رابطه مستقیمی با تعداد سوپرستپ‌های مورد نیاز در سیستم‌های پردازش گراف بلوکی دارد، روش پیشنهادی با کاهش آن قادر به افزایش کارایی این دسته از روش‌ها خواهد شد.

کلیدواژه: بخش‌بندی، مبتنی بر بلوک، گراف، قطر.

۱- مقدمه

با افزایش سرعت رشد شبکه‌هایی نظیر شبکه‌های اجتماعی، گراف وب و شبکه‌های نظیر به نظیر، اندازه این شبکه‌ها در سال‌های اخیر با رشد بالایی روبه‌رو بوده است. از آنجا که این شبکه‌ها در حوزه‌های مختلفی اعم از سیاسی [۱]، روان‌شناسی [۲]، جامعه‌شناسی [۳] و نظایر آنها مورد توجه است، استخراج ویژگی‌های این شبکه‌ها و همچنین تحلیل آنها به یک حوزه جذاب تحقیقاتی تبدیل شده است. با توجه به ذات ارتباطات

این مقاله در تاریخ ۱۵ شهریور ماه ۱۴۰۰ دریافت و در تاریخ ۱۵ اسفند ماه ۱۴۰۰ بازنگری شد.

مسعود ساغریچیان (نویسنده مسئول)، گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه الزهراء، تهران، ایران، (email: m.sagharichian@alzahra.ac.ir)

مرتضی علیپور لنگوری، دپارتمان کامپیوتر و نرم‌افزار، دانشگاه مکماستر، همیلتون، کانادا، (email: alipoum@mcmaster.ca)

می‌کنند [۳۴] و [۳۵]. بخش‌بندی موازی برای معماری‌های چند هسته‌ای در [۱۷] و مبتنی بر دیسک در [۳۶] معرفی شده است. این دسته از الگوریتم‌ها با هدف کاهش یال برشی و افزایش تعادل بار پارتیشن‌ها طراحی شده‌اند. دسته دیگری از الگوریتم‌ها بر بخش‌بندی گراف‌های پویا و به صورت آنلاین تمرکز کرده‌اند. مهم‌ترین ایده‌های استفاده‌شده در این حوزه، استفاده از روش‌های تعادل بار، درهم‌سازی و حریصانه می‌باشد [۳۷]. روش Fennel [۳۸] چارچوبی را فراهم آورده که مسأله بخش‌بندی جریان را به صورت یک تابع سراسری فرموله کرده است. روش [۲۳] و Spinner [۲۲] به ارائه روش بخش‌بندی بر اساس معیارهای یال-برشی و تعادل بار ارائه نموده است. روش GraphGrind [۲۴] و [۲۰] با تمرکز بیشتر بر تعادل بار به ارائه یک روش بخش‌بندی گراف به صورت موازی عمل نموده است.

اغلب سیستم‌های پردازش گراف بلوکی نظیر «بلوجل» [۱۱]، Giraph++ [۳۹] و Grape [۴۰] از الگوریتم‌های بخش‌بندی مبتنی بر یال برشی استفاده می‌کنند. به عنوان مثال، الگوریتم ورونوی که برای سیستم بلوجل ارائه شده است، بر اساس معیار یال برشی بلوکی عمل می‌کند. در [۱۴] برای شناسایی ویژگی‌های سیستم‌های پردازش گراف، از گراف‌های تحت آزمون گراف‌های سطح بالا مشابه آنچه در این مقاله ارائه شده است، ساخته شده است. در این مقاله، تأثیر رویکردهای مختلف بخش‌بندی موجود بر روی گراف سطح بالا مورد ارزیابی قرار گرفته است. مرجع [۴۱] به ارائه یک مقایسه عملی بر روی تعدادی از الگوریتم‌های بخش‌بندی پرداخته است.

در سال‌های اخیر با توجه به موفقیت سیستم‌های پردازش گراف مبتنی بر بلوک و همچنین کارایی پایین بخش‌بندی مبتنی بر یال، الگوریتم‌های بخش‌بندی مختلفی برای این منظور ارائه شده است. روش [۴۲] برای بهبود عملکرد سیستم‌های پردازش گراف مبتنی بر بلوک، از اصلاح بخش‌بندی به صورت پویا بهره می‌برد. در این روش، بلوک‌ها در زمان اجرا به زیربخش‌هایی، شکسته و برخی از این زیربخش‌ها به پارتیشن‌های دیگر منتقل می‌شوند. مشکل این روش آن است که انتقال تعداد زیادی گره، هم‌زمان با اجرای الگوریتم، ترافیک زیادی را به سیستم وارد نموده و بهره‌وری را تحت تأثیر قرار می‌دهد. روش Drone [۴۳] به جای استفاده از الگوریتم‌های مبتنی بر یال برشی برای بخش‌بندی گراف، از رهیافت برش گره استفاده می‌کند. تفاوت روش پیشنهادی با الگوریتم‌های موجود آن است که هیچ یک از روش‌های موجود، قطر گراف سطح بالای حاصل از بخش‌بندی را به عنوان معیار بخش‌بندی در نظر نمی‌گیرند. روش پیشنهادی بر اساس بررسی‌های انجام‌گرفته، اولین روشی است که با در نظر گرفتن معیارهای جدیدی نظیر قطر گراف سطح بالای حاصل از بخش‌بندی پارتیشن‌های ایجادشده و انحراف معیار اندازه گره‌های سطح بالا، به تولید یک بخش‌بندی با توجه به ویژگی‌های ذاتی سیستم‌های پردازش گراف بلوکی سعی می‌نماید.

۳- روش پیشنهادی

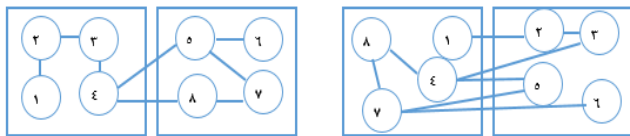
مسأله بخش‌بندی گراف را می‌توان به صورت تقسیم‌کردن گراف به تعداد m گروه مستقل تعریف نمود، به طوری که گروه‌های نهایی بر اساس یک سری معیارهای بهینه‌سازی از هم تفکیک شده باشند. از آنجا که هدف اصلی الگوریتم پیشنهادی، بهبود کارایی سیستم‌های پردازش گراف مبتنی بر بلوک است، معیار بهینه‌سازی باید بر اساس ویژگی‌های این سیستم‌ها تعریف شود. به همین منظور در بخش ۳-۱ عوامل تأثیرگذار بر موفقیت الگوریتم بخش‌بندی برای این دسته از سیستم‌ها معرفی شده

چشم‌گیری بر کیفیت پارتیشن‌ها برای سیستم‌های مبتنی بر بلوک دارند تعریف شده است: (۱) قطر گراف سطح بالا و (۲) متوسط انحراف معیار اندازه گره‌های سطح بالا. منظور از گره و گراف سطح بالا، گراف بلوکی حاصل از فرایند بخش‌بندی می‌باشد که مفصلاً در بخش‌های بعدی معین شده است. بر اساس این معیارها، یک روش جدید به منظور بخش‌بندی گراف ارائه شده که به دنبال بهینه‌سازی پارتیشن‌های تولیدی بر اساس دو معیار پیشنهادی می‌باشد: (۱) کمینه‌سازی قطر گراف سطح بالا و (۲) کمینه‌سازی متوسط انحراف معیار گره‌های سطح بالا. بخش‌هایی از الگوریتم پیشنهادی که با گراف اصلی سروکار دارند، به صورت توزیع‌شده و قسمت‌هایی که با گراف سطح بالای تولیدی (اندازه آن بسیار کوچک‌تر از گراف اصلی است) سروکار دارند به صورت متمرکز، پیاده‌سازی شده‌اند. نتایج ارزیابی‌ها بر روی مجموعه داده‌های واقعی نشان دادند که نه تنها پارتیشن‌های تولیدی روش پیشنهادی از حیث قطر گراف سطح بالا بسیار کوچک‌تر از پارتیشن‌های تولیدی روش‌های متیس و «ورونوی» هستند، بلکه میزان یال برشی و تعادل بار پارتیشن‌های حاصل از روش پیشنهادی (که جزء اهداف ثانویه الگوریتم پیشنهادی و اهداف اصلی روش‌های تحت آزمون است)، بسیار نزدیک به متیس که بهترین روش بخش‌بندی متمرکز است، می‌باشد. به طور خلاصه نوآوری‌های این مقاله عبارت هستند از:

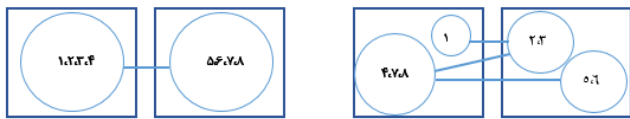
- استخراج و معرفی معیارهایی جدید برای بخش‌بندی سیستم‌های پردازش گراف بلوکی که می‌توانند کارایی این دسته از سیستم‌ها را بهینه‌تر نمایند.
- ارائه یک روش بخش‌بندی مبتنی بر معیارهای معرفی‌شده در این مقاله
- ارزیابی کیفی پارتیشن‌های تولیدشده از روش پیشنهادی سایر بخش‌های این مقاله به شرح زیر است: بخش ۲ به مروری بر کارهای مرتبط می‌پردازد و در بخش ۳، معیارها و روش بخش‌بندی پیشنهادی به ترتیب معرفی می‌شوند. بخش ۴، نتایج حاصل از ارزیابی روش پیشنهادی را با روش‌های موجود نشان می‌دهد و بخش ۵ هم نتیجه‌گیری و کارهای تحقیقاتی آتی را بیان می‌نماید.

۲- پیشینه پژوهش

بخش‌بندی به عنوان یکی از مسائل کلاسیک در بسیاری از شاخه‌های علوم کامپیوتر نظیر VLSI [۲۵] و پردازش توزیع‌شده [۲۶] شناخته می‌شود و ارائه یک بخش‌بندی بهینه، یک مسأله NP-Hard است [۲۷]. الگوریتم‌های بخش‌بندی ارائه‌شده را می‌توان به گروه‌هایی طبقه‌بندی نمود. گروه اول بر مبنای جابه‌جایی گره‌ها، عمل بخش‌بندی را انجام می‌دهد که مهم‌ترین الگوریتم این دسته، الگوریتم KL [۲۸] می‌باشد، اما این رهیافت برای گراف‌های بزرگ مقیاس مناسب نیست. این بدان جهت است که این دسته از الگوریتم‌ها نیازمند منابع و زمان زیاد به منظور شناسایی بهترین جابه‌جایی‌ها هستند. برای گراف‌های بزرگ، رویکردهای چندسطحی ارائه شدند [۲۹] تا [۳۱] و الگوریتم چندسطحی از ۲ فاز تشکیل می‌شود. در فاز اول سعی می‌گردد تا حد امکان، پیچیدگی و اندازه گراف کوچک شود. سپس این گراف کوچک‌شده با استفاده از الگوریتم‌های نیمه‌بهینه نظیر KL بخش‌بندی می‌شود. در فاز دوم، نتیجه بخش‌بندی به گراف اصلی اعمال می‌گردد. دسته دیگر الگوریتم‌ها بر مبنای ماکسیمم جریان عمل می‌نمایند [۳۲] و [۳۳]. دسته بعدی از الگوریتم‌ها بر مبنای الگوریتم‌های اکتشافی و حریصانه و همچنین الگوریتم‌های ژنتیک عمل



شکل ۲: دو نمونه از بخش‌بندی گراف شکل ۱.



شکل ۳: گراف سطح بالای حاصل از بخش‌بندی‌های شکل ۲.

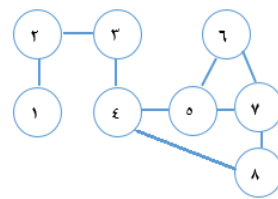
باید تمامی گره‌های سطح بالا را پیمایش نماید که همان قطر گراف سطح بالا می‌باشد. این فرایند برای محاسباتی نظیر رتبه‌بندی صفحه که محاسبه‌ای تکراری است، در بدترین حالت به صورت ضربی از قطر گراف سطح بالا سوپرستپ نیاز دارد. در نتیجه اولین معیار تأثیرگذار پیشنهادی بر بخش‌بندی سیستم‌های مبتنی بر بلوک، کاهش قطر گراف سطح بالای حاصل از بخش‌بندی می‌باشد.

به منظور فهم بهتر تأثیر قطر گراف سطح بالای حاصل از بخش‌بندی بر کارایی این سیستم‌ها، گراف شکل ۱ را در نظر بگیرید. دو بخش‌بندی مختلف برای این گراف در شکل ۲ ارائه شده است. همان طور که دیده می‌شود، تعداد سوپرستپ مورد نیاز برای محاسبه کوتاه‌ترین مسیر از گره ۱ به سایر گره‌ها برای شکل ۲ بخش‌بندی چپ برابر ۲ و برای بخش‌بندی راست برابر ۴ می‌باشد. علت اختلاف تعداد سوپرستپ‌ها، تفاوت در قطر گراف سطح بالای این دو بخش‌بندی است که برای بخش‌بندی‌های چپ و راست در شکل ۲ به ترتیب برابر ۱ و ۳ می‌باشد. گراف سطح بالای این دو بخش‌بندی در شکل ۳ آمده است.

۲-۱-۳ اندازه گره‌های سطح بالا

همان طور که گفته شد، محاسبه یک بلوک درون همان سوپرستپ انجام می‌گیرد. حال فرض کنید که گراف طوری بخش‌بندی شده که گره سطح بالای A دارای ۱۰۰۰ گره و دو گره سطح بالای دیگر (B و C) هر یک دارای تنها ۵ گره باشند. در این حالت برای دو گره سطح بالای B و C چون تعداد کمی گره دارند، محاسبه درون سوپرستپی خیلی سریع انجام می‌شود و هر دو منتظر پایان محاسبه درون سوپرستپی گره سطح بالای A می‌شوند. این امر منجر می‌گردد که استفاده مؤثر از منابع سیستم توزیع شده انجام نشود و محاسبه به صورت نامتعادل باشد، به طوری که ماشین‌ها منتظر تکمیل محاسبه یک ماشین می‌شوند. در این مقاله تعداد گره‌های یک گره سطح بالا، اندازه آن گره نام‌گذاری می‌گردد $(|B_p(i)|)$. از طرفی هر قدر اندازه گره سطح بالا بزرگ‌تر باشد، تعداد گره بیشتری را درون یک سوپرستپ به روز رسانی می‌کند. از طرف دیگر، هر قدر اندازه گره‌های سطح بالا نامتعادل‌تر باشد، بهره‌وری محاسبات توزیع شده کاهش می‌یابد. بر این اساس، معیار دوم به این صورت تعریف شده است: انحراف معیار اندازه گره‌های سطح بالا هر قدر کمتر باشد، کیفیت بخش‌بندی برای سیستم‌های مبتنی بر بلوک بهتر خواهد بود.

به منظور فهم بهتر موضوع فرض کنید که گرافی با ۱۰۰۰ رأس به ۴ پارتیشن و هر یک با اندازه ۲۵۰ تقسیم‌بندی شده باشد. حال گراف سطح بالای حاصل از بخش‌بندی اول دربرگیرنده ۸ گره سطح بالا به اندازه ۱۲۵ باشد و برای بخش‌بندی دوم، یک گره سطح بالا به اندازه ۲۵۰ و ۱۵۰ گره سطح بالا به اندازه ۵ تولید نماید. بدیهی است که بخش‌بندی اول از لحاظ این معیار مناسب‌تر است. طبق معیار تعریف شده در این مقاله، انحراف معیار گره‌های سطح بالا برای بخش‌بندی اول برابر ۰ و



شکل ۱: مثالی از یک گراف.

است. در بخش ۳-۲ الگوریتم پیشنهادی که بر اساس این عوامل طراحی شده است، معرفی می‌گردد و در بخش ۳-۳ با استفاده از یک مثال چگونگی کارکرد این روش توضیح داده خواهد شد.

۱-۳ عوامل تأثیرگذار بر کیفیت پارتیشن‌ها برای سیستم‌های مبتنی بر بلوک

به طور کلی گراف $G=(V, E)$ به تعداد p پارتیشن تقسیم‌بندی شده و هر پارتیشن بر روی یک ماشین مورد پردازش قرار می‌گیرد. هر پارتیشن p دربرگیرنده زیرمجموعه‌ای از گره‌های گراف V_p به همراه یال‌های خروجی از این گره‌ها E_p می‌باشد. در سیستم‌های مبتنی بر بلوک، اجزای همبند هر پارتیشن به عنوان یک بلوک مد نظر قرار می‌گیرند و محاسبه به ازای هر بلوک تعریف می‌شود. مجموعه B_p بیانگر بلوک‌های پارتیشن p است که به فرم $\{CC(P)\}$ تعریف می‌گردد. از نماد $B_p(i)$ برای نمایش i امین بلوک در پارتیشن p استفاده می‌شود. در نتیجه می‌توان از گراف اولیه G ، یک گراف سطح بالای $HL(G)$ تعریف نمود به طوری که گره‌های آن، اجزای همبند پارتیشن‌های آن (از این پس در این مقاله به آنها گره‌های سطح بالا گفته می‌شود) و یال‌های آن، یال‌های بین گره‌های همبند پارتیشن‌ها (از این پس در این مقاله به آنها یال‌های سطح بالا گفته می‌شود) هستند. به طور دقیق‌تر رئوس این گراف برابر B_p به ازای تمام p ها می‌باشد. اگر یال‌های گراف G را به صورت $E=(u, v)$ نشان دهیم و داشته باشیم $B_p(i, u)=B_q(j, v)$ ، یال‌های $HL(G)$ زیرمجموعه‌ای از E هستند که $i=j$ یا $i=q$ یا $p=q$ است. در هر سوپرستپ ابتدا محاسبه در سطح گره‌های سطح بالا انجام می‌شود. هر گره سطح بالا می‌تواند پیام‌هایی را برای گره‌های سطح بالای دیگر ارسال نماید که در سوپرستپ بعدی در اختیار گره سطح بالای مقصد قرار می‌گیرد. محاسبه، زمانی خاتمه می‌یابد که تمامی گره‌های سطح بالا در حالت غیر فعال باشند و هیچ پیامی هم در سطح شبکه وجود نداشته باشد [۱۱]. بر اساس این فرایند در سیستم‌های مبتنی بر بلوک، در این بخش دو عامل تأثیرگذار بر کیفیت بخش‌بندی این سیستم‌ها معرفی می‌گردد: قطر گراف سطح بالا و اندازه گره‌های سطح بالا.

۱-۱-۳ قطر گراف سطح بالا

از آنجا که هر سوپرستپ در سیستم‌های پردازش گراف بلوکی نیازمند همگام‌سازی سراسری است، به عنوان یک گلوگاه در نظر گرفته می‌شود. بنابراین کاهش تعداد سوپرستپ‌های مورد نیاز به منظور تکمیل محاسبه می‌تواند تأثیر بسیار مهمی در زمان اجرای این سیستم‌ها داشته باشد. می‌توان نشان داد که تعداد سوپرستپ مورد نیاز برای انجام یک محاسبه در سیستم‌های پردازش گراف، رابطه مستقیمی با قطر گراف سطح بالا $Diameter(HL(G))$ دارد. مثلاً برای محاسبه کوتاه‌ترین مسیر از یک گره به تمامی گره‌های دیگر، در بدترین حالت به اندازه قطر گراف سطح بالا سوپرستپ مورد نیاز است. این بدان جهت است که باید مقدار مینیمم فاصله از گره a که در گره سطح بالای A قرار دارد به گره b که در گره سطح بالای B قرار دارد، ارسال گردد. در بدترین حالت، این مقدار

برای بخش‌بندی دوم برابر ۱۹/۹ است که نشان‌دهنده برتری بخش‌بندی اول می‌باشد.

۳-۲ الگوریتم بخش‌بندی پیشنهادی

بر اساس معیارهای معرفی شده، الگوریتمی توزیع شده جهت بخش‌بندی گراف برای سیستم‌های مبتنی بر بلوک ارائه گردیده است. ابتدا گراف ورودی به صورت تصادفی به تعداد پارتیشن مورد نیاز، بخش‌بندی و هر بخش به یک ماشین تخصیص داده می‌شود. سپس گراف سطح بالای این بخش‌بندی ساخته می‌گردد. چگونگی ساخت این گراف به صورت توزیع شده در بخش ۳-۲-۱ توضیح داده خواهد شد. از آنجا که این گام با گراف اصلی سروکار دارد و بسیار بزرگ است، این گام به صورت توزیع شده پیاده‌سازی گردیده است. در گام بعدی، گره‌های سطح بالا بر اساس معیارهای پیشنهادی ادغام می‌شوند. به فرایند ادغام در بخش ۳-۲-۲ پرداخته خواهد شد. در گام آخر که در بخش ۳-۲-۳ اشاره شده است، شماره پارتیشن گره‌های سطح بالای باقیمانده به‌روزرسانی می‌شود. دو گام پایانی روش پیشنهادی، گراف سطح بالا را که بسیار کوچک‌تر از گراف اولیه می‌باشد، پردازش می‌کند و در نتیجه این دو گام به صورت متمرکز پیاده‌سازی شده‌اند.

۳-۲-۱ اندازه گره‌های سطح بالا

هر ماشین، پارتیشن مربوط به خود را بررسی و اجزای همبند پارتیشن خود را فارغ از سایر پارتیشن‌ها محاسبه می‌کند. به عبارت بهتر، هر جزء همبندی درون یک پارتیشن دربرگیرنده گره‌هایی است که حداقل یک مسیر بین آنها وجود دارد و تمامی گره‌های آن مسیر درون همان پارتیشن باشند. هر یک از اجزای همبند، یک گره سطح بالا برای این پارتیشن محسوب گردیده و به هر گره سطح بالا شناسه‌ای منحصر به فرد تخصیص داده می‌شود. برای آن که شناسه‌ها بین پارتیشن‌ها هم منحصر به فرد باشند، شناسه به صورت ترکیب شماره پارتیشن و یک عدد منحصر به فرد افزایشی (با شروع از ۱) تعیین می‌گردند. مثلاً جزء همبند سوم از پارتیشن دوم، شناسه $B_p(2_3)$ را خواهد داشت. سپس هر ماشین اطلاعات مربوط به گره‌های سطح بالای خود را با بقیه ماشین‌ها به اشتراک می‌گذارد. بدین منظور از پایگاه داده ردیس که یک پایگاه داده NoSQL مقیم در حافظه و از نوع کلید-مقدار می‌باشد، استفاده شده است. از آنجا که ردیس امکان نگهداری اطلاعات درون حافظه و به صورت توزیع شده را فراهم می‌کند، از این پایگاه داده استفاده شده است. فرمت قرارگیری اطلاعات درون ردیس به صورت زیر است:

- کلید = شناسه گره گراف ورودی (v)
- مقدار = شناسه گره سطح بالا ($B_p(i)$)

هنگامی که تمامی ماشین‌ها، اطلاعات گره‌های سطح بالای خود را درون ردیس قرار دهند، هر ماشین می‌تواند به اطلاعات گره‌های سطح بالای سایر ماشین‌ها دسترسی داشته باشد. در نتیجه هر ماشین می‌تواند رابطه بین گره‌های سطح بالای پارتیشن خود را با گره‌های سطح بالای سایر پارتیشن‌ها محاسبه نماید. هر ماشین به ازای گره‌های سطح بالای ماشین‌های دیگر که حداقل با یکی از گره‌های سطح بالای پارتیشن خود ارتباط دارد، ۲ پارامتر را محاسبه می‌کند:

- درجه چسبندگی: تعداد گره‌های سطح بالایی که با این گره سطح بالا ارتباط دارند

$$Cohesion_Deg(B_p(i), B_q(j))$$

$$= |\{v | v \in B_p(i), (v, u) \in E, u \in B_q(j)\}| \quad (1)$$

- میزان چسبندگی: تعداد یال‌های برشی که با این گره سطح بالا ارتباط دارند

$$Cohesion_Val(B_p(i), B_q(j)) = |\{(v, u) | v \in B_p(i), (v, u) \in E, u \in B_q(j)\}| \quad (2)$$

پس از آن که این پارامترها محاسبه شد، هر ماشین به ازای هر گره سطح بالا رکوردی در ردیس ایجاد می‌کند:

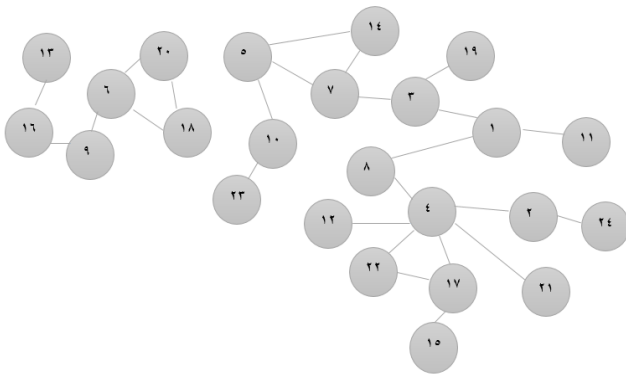
- کلید = شناسه گره سطح بالای ماشین خود
- مقدار = شناسه گره سطح بالای ماشین دیگر، درجه چسبندگی، میزان چسبندگی

بنابراین خروجی این گام، لیست مجاورت گره‌های سطح بالای پارتیشن‌ها به همراه دو مقدار محاسبه شده برای چسبندگی می‌باشد.

سربار ارتباطی و محاسباتی گام اول، پس از آن که گره‌ها به صورت تصادفی بخش‌بندی شدند، ابتدا بایستی اجزای همبند هر پارتیشن محاسبه گردد. با توجه به ذات تصادفی بودن بخش‌بندی انتظار می‌رود که در هر پارتیشن، $|V|/P$ گره و $|E|/P$ یال وجود داشته باشد. بنابراین پیچیدگی زمانی محاسبه اجزای همبند برای هر پارتیشن برابر $O(|V| + |E|/P)$ است. سپس هر ماشین به ازای هر گره، شماره گره سطح بالای خود را باید درون ردیس درج کند. این بخش نیازمند $O(|V|/P)$ پیام در شبکه به ازای هر پارتیشن می‌باشد. هر پارتیشن بایستی اطلاعات گره‌های سطح بالای یال‌هایی را که با آنها ارتباط دارند، محاسبه نماید. در بدترین حالت تمامی یال‌ها با سایر پارتیشن‌ها خواهند بود و در نتیجه این گام نیازمند $O(|E|/P)$ سربار ارتباطی و محاسباتی به ازای هر پارتیشن است. محاسبه فاکتورها به ازای هر گره سطح بالای نیازمند $O(|E|/P)$ سربار محاسباتی به ازای هر پارتیشن است. با فرض آن که تعداد گره‌های سطح بالا برابر H باشد، در پایان این گام، قرار گرفتن اطلاعات گره‌های محاسباتی نیازمند $O(H)$ سربار ارتباطی است. در نتیجه گام اول که به صورت توزیع شده اجرا می‌گردد، پیچیدگی زمانی‌اش از مرتبه $O(|V| + |E|)$ و سربار ارتباطی آن از مرتبه $O(|V|)$ می‌باشد. دلیل حذف H از سربار ارتباطی آن است که تعداد گره‌های سطح بالا خیلی کمتر از تعداد کل گره‌های گراف می‌باشد.

۳-۲-۲ ادغام گره‌های سطح بالا

شبه‌کد این گام در شکل ۴ نشان داده شده است. در این گام که به صورت متمرکز بر روی یک ماشین اجرا می‌شود، ابتدا لیست مجاورت گراف سطح بالای تولیدشده در مرحله قبل را که در ردیس قرار گرفته است، می‌خواند (خط ۱ شبه‌کد). همان‌طور که گفته شد، هر رکورد این لیست نگاشتی است از شناسه گره سطح بالا به اطلاعات استخراج شده در مرحله قبل. سپس گره‌های سطح بالا را بر اساس فاکتورهای محاسبه شده (درجه چسبندگی یا میزان چسبندگی) به صورت نزولی مرتب می‌کند (خط ۲ از شبه‌کد). پیچیدگی زمانی این بخش $O(H \log H)$ می‌باشد. در بخش ارزیابی، تأثیر هر یک از این دو پارامتر بر کیفیت بخش‌بندی مورد بررسی قرار گرفته و البته می‌توان از ترکیب هر دو معیار هم استفاده نمود. کار از گره‌های با بیشترین مقدار فاکتور محاسبه شده، شروع می‌شود. دو گره سطح بالا تنها در صورتی می‌توانند با هم ادغام شوند که مجموع اندازه آنها (مجموع تعداد گره‌های این ۲ گره سطح بالا) از حداکثر اندازه مجاز یک پارتیشن تجاوز نکند (خطوط ۸ تا ۱۱ در شبه‌کد). برای پرهیز از انحراف معیار بالا در اندازه گره‌های سطح بالا (در نظر گرفتن معیار دوم پیشنهادی)، این گام در چند سطح انجام می‌شود. در هر سطح، حداکثر



شکل ۵: گراف نمونه.

کمک درجه عدم تعادل، شانس ادغام بیشتر گره‌های سطح بالا ایجاد می‌شود. البته با توجه به آن که این درجه عموماً کم تنظیم می‌شود، انتظار داریم گره‌های سطح بالای کوچک بتوانند با گره‌های سطح بالای دیگر ادغام شوند. در نتیجه اگرچه هدف، کاهش قطر گراف سطح بالاست، انتظار می‌رود با ادغام گره‌های سطح بالای کوچک‌تر، انحراف معیار اندازه گره‌های سطح بالا نیز کاهش یابد.

فرایند به این صورت خواهد بود که گره‌های سطح بالا بر اساس اندازه به صورت نزولی مرتب می‌شوند. هر گره سطح بالا به پارتیشن‌های تخصیص داده می‌شود که قطر گراف سطح بالا را کاهش دهد. اگر دو گره سطح بالایی که به یک پارتیشن تخصیص داده شده‌اند دارای رابطه باشند، طبیعتاً با یکدیگر ادغام می‌شوند. برای نیل به این هدف از رویکرد حریصانه استفاده شده است. ابتدا پارتیشن‌های کاندیدایی که گره سطح بالای فعلی می‌تواند با آنها ادغام شود، تعیین می‌گردند. پارتیشن‌های کاندیدا، پارتیشن‌هایی هستند که اگر گره سطح بالا به آن پارتیشن اضافه شود، اندازه این پارتیشن از حداکثر اندازه مجاز برای پارتیشن به علاوه درجه عدم تعادل بیشتر نشود. سپس بهترین پارتیشن انتخاب می‌شود. بهترین پارتیشن، پارتیشنی است که قطر سطح بالای جدید آن کمترین باشد. اگر بیش از یک پارتیشن بهترین وجود داشته باشد، پارتیشنی که فضای خالی کمتری دارد انتخاب می‌شود. از آنجا که محاسبه قطر گراف سطح بالا باید به صورت مکرر انجام پذیرد، از روش محاسبه قطر پویا تحت عنوان DiameterMonitoring که در [۴۴] معرفی گردیده است، بهره گرفته شده است. این الگوریتم که توسط نویسندگان این مقاله ارائه شده است، با اضافه و حذف گره و یال جدید به صورت کارا قطر گراف را به‌روزرسانی می‌کند.

۳-۳ شرح الگوریتم پیشنهادی به کمک یک مثال عملی

در این بخش برای توضیح بهتر الگوریتم پیشنهادی، گام‌های الگوریتم به کمک یک مثال تشریح می‌گردد. گراف شکل ۵ را در نظر بگیرید که دربرگیرنده ۲۴ رأس و ۲۵ یال است و از دو جزء همبند تشکیل شده و قطر آن ۹ می‌باشد. فرضاً بخواهیم این گراف را به ۴ پارتیشن که هر یک دارای ۶ رأس هستند، بخش‌بندی کنیم. شکل ۶ نمونه‌ای از بخش‌بندی تصادفی این گراف را نشان می‌دهد. همان طور که دیده می‌شود، تعداد یال‌های برشی برابر ۲۰ (۸۰٪ تعداد یال‌ها) می‌باشد. گراف سطح بالای این بخش‌بندی شامل ۱۹ گره و قطر ۷ است.

هر ماشین در گام اول همان طور که گفته شد، گره‌های سطح بالای پارتیشن خود را محاسبه و اطلاعات مربوط را در ردیس وارد می‌کند. اطلاعاتی که توسط ماشین ۱ در ردیس نوشته می‌شود در جدول ۱ ارائه شده است. سپس هر ماشین با توجه به اطلاعات گره‌های سطح بالای

Function: Merge

Input:

Pref: Preferred measure
S: The number of graph vertices
P: The number of partitions
L: The number of levels

Output:

Updated HL_Graph at redis

```
01: [HL_Vertex_Id → (Size, Cohesion_Deg, cohesion_Val)] HL_Graph = Read_From_Redis()
02: Sort(HL_Graph according to Pref)
03: Level = 1
04: for l=1 to L
05:   Max_Partition_Size = (S * l) / (P * L)
06:   foreach vertex v in HL_Graph
07:     foreach vertex vv in HL_Graph
08:       if (v != vv and (v.Size + vv.Size) < Max_Partition_Size)
09:         merge vv into v
10:       update v factors
11:     remove vv from HL_Graph
12: update HL_Graph into redis
```

شکل ۴: شبه‌کد گام ادغام.

اندازه مجاز یک پارتیشن از رابطه زیر محاسبه می‌شود که در آن S اندازه کل گره‌های گراف، l شماره سطح فعلی، P تعداد کل پارتیشن‌ها و L تعداد کل سطوح می‌باشد (خطوط ۳ تا ۵ در شبه‌کد). پیچیدگی زمانی این گام برابر $O(H^*L)$ است

$$?? = \frac{S \times l}{P \times L} \quad (3)$$

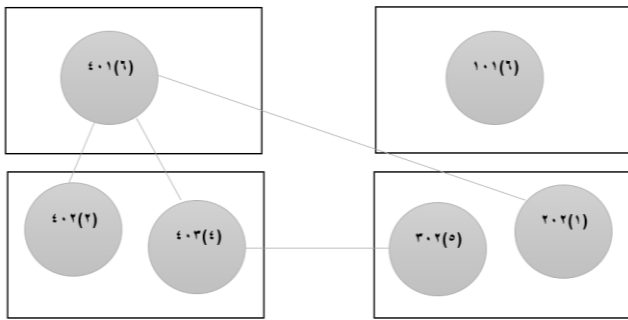
مثلاً اگر تعداد گره‌های گراف برابر ۱۰۰ و تعداد پارتیشن‌ها برابر ۱۰ باشد و بخواهیم فرایند ادغام در دو سطح انجام شود، حداکثر اندازه مجاز یک پارتیشن در سطح اول برابر ۵ و در سطح دوم برابر ۱۰ خواهد شد.

این گام در راستای هر دو معیار تعریف‌شده در این مقاله عمل می‌کند: (۱) هر قدر تعداد گره سطح بالای بیشتری ادغام گردد، قطر گراف سطح بالا کاهش پیدا می‌کند. (۲) قراردادن شرط سخت‌گیرانه حداکثر سایز و همچنین سطح‌بندی ادغام در آغاز سعی می‌کند که گره‌های سطح بالای کوچک‌تر را ادغام نماید و از ایجاد گره‌های سطح بالای خیلی بزرگ پرهیز کند. این امر باعث می‌شود که انحراف معیار اندازه گره‌های سطح بالا نیز کاهش پیدا کند. این گام که به صورت متمرکز اجرا می‌شود، سربار محاسباتی از مرتبه $O(H^*L)$ را داراست.

۳-۲-۳ انتساب شماره پارتیشن جدید به گره‌های سطح بالای ادغام‌شده

در گام قبل سعی شد که قطر گراف سطح بالا و همچنین انحراف معیار اندازه گره‌های سطح بالا به صورت هم‌زمان کمینه شود. در این گام باید شماره پارتیشن گره‌های سطح بالا اصلاح گردند. رویکرد در نظر گرفته شده در این گام بر اساس معیار پیشنهادی کاهش قطر گراف سطح بالا می‌باشد. به عبارت دیگر، هدف این گام آن است که بتواند با تحمل کردن کمی عدم تعادل در پارتیشن‌ها، قطر گراف سطح بالای حاصل‌شده از بخش‌بندی جدید را کمتر سازد. برای نیل به این هدف، برای هر پارتیشن یک مقدار ثابت تحت عنوان درجه عدم تعادل در نظر گرفته می‌شود. این مقدار، بیانگر حداکثر تعداد گره‌هایی است که یک پارتیشن می‌تواند بیش از اندازه مجازش داشته باشد.

فرض کنید که کل گره‌های گراف ۱۰۰۰ و تعداد پارتیشن و درجه عدم تعادل برابر با ۱۰ باشد. در نتیجه به جای این که هر پارتیشن حداکثر ۱۰۰ گره را بتواند در خود جای دهد، حداکثر ۱۱۰ گره را می‌تواند در بر گیرد. به



شکل ۸: گراف سطح بالای حاصل از گام سوم بخش‌بندی پیشنهادی.

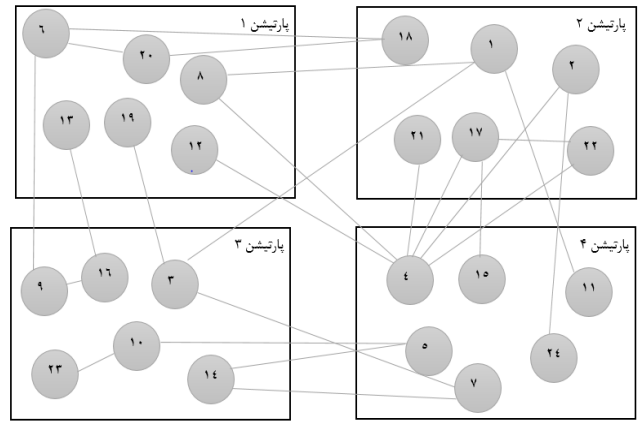
جدول ۲: رابطه گره‌های سطح بالای پارتیشن ۱ با سایر گره‌های سطح بالا.

کلید = شناسه گره سطح بالا	مقدار = { میزان چسبندگی، درجه چسبندگی، شناسه گره سطح بالا }
$P1_1$	{ $P2_1, 1, 2$ }
$P1_5$	{ $P2_2, 1, 1$ }
$P1_1$	{ $P3_1, 2, 2$ }
$P1_2$	{ $P3_1, 2, 2$ }
$P1_3$	{ $P3_2, 1, 1$ }
$P1_4$	{ $P4_1, 2, 2$ }
$P1_5$	{ $P4_1, 2, 2$ }

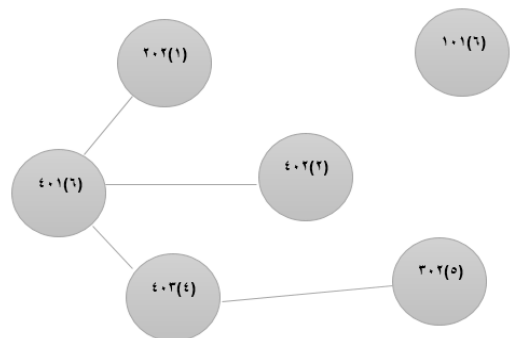
گام سوم با اصلاح شماره پارتیشن‌ها به همراه اضافه‌نمودن درجه عدم تعادل، سعی در کاهش بیشتر قطر گراف سطح بالا می‌نماید. در این مثال گام سوم را با دو مقدار مختلف برای درجه عدم تعادل نشان می‌دهیم. یک بار درجه عدم تعادل را ۰ لحاظ کردیم، به طوری که تمام پارتیشن به صورت کاملاً متعادل باشند. خروجی این بخش‌بندی در شکل ۸ نشان داده شده است. در این حالت قطر گراف سطح بالا برابر ۳ و تعداد یال‌های برشی برابر ۴ می‌باشد. به علاوه اگر درجه عدم تعادل را ۳ در نظر بگیریم (یعنی هر پارتیشن حداکثر ۹ گره را بتواند در خود جای دهد)، قطر گراف سطح بالا برابر ۲ و تعداد یال‌های برشی برابر ۳ می‌باشد.

۴- تجزیه و تحلیل یافته‌ها

نحوه آزمون و ارزیابی روش پیشنهادی در این بخش ارائه می‌شود. از آنجا که روش بخش‌بندی پیشنهادی، روشی توزیع‌شده می‌باشد از ۲۵ ماشین به عنوان سیستم توزیع‌شده استفاده گردیده است. هر ماشین دارای ۲ هسته ۳ گیگاهرتز و ۸ گیگابایت حافظه است. روش پیشنهادی با دو روش بخش‌بندی مقایسه شده است: (۱) روش ورونوی [۱۱] که روشی توزیع‌شده است. از آنجا که این روش نیز به طور خاص‌منظوره برای سیستم‌های گرافی ارائه شده است، معیار خوبی برای مقایسه با روش پیشنهادی می‌باشد. (۲) روش متیس [۱۸] که روشی متمرکز است. از آنجا که این روش تنها فرایند بخش‌بندی را بر روی یک ماشین انجام می‌دهد، در نتیجه مقیاس‌پذیر نیست و در عمل برای بخش‌بندی گراف‌های بزرگ مورد استفاده قرار نمی‌گیرد. اما به واسطه ذات متمرکز بودن و داشتن اطلاعات کامل از گراف، توانایی تولید بخش‌بندی با کیفیت بسیار بالا را دارد. در نتیجه مقایسه کیفیت روش پیشنهادی با این روش می‌تواند اطلاعات مفیدی از کیفیت روش پیشنهادی ارائه نماید. با توجه به این که روش متیس تنها بر روی یک ماشین اجرا می‌شود، برای این که بتوانیم آن را بر روی مجموعه داده‌های آزمون مورد ارزیابی قرار دهیم، از یک سیستم ۸ هسته‌ای با ۴۰ گیگابایت حافظه برای متیس استفاده گردیده و تعداد پارتیشن‌ها در تمامی موارد برابر با ۴۰ در نظر گرفته شده است.



شکل ۵: بخش‌بندی تصادفی از گراف شکل ۵.



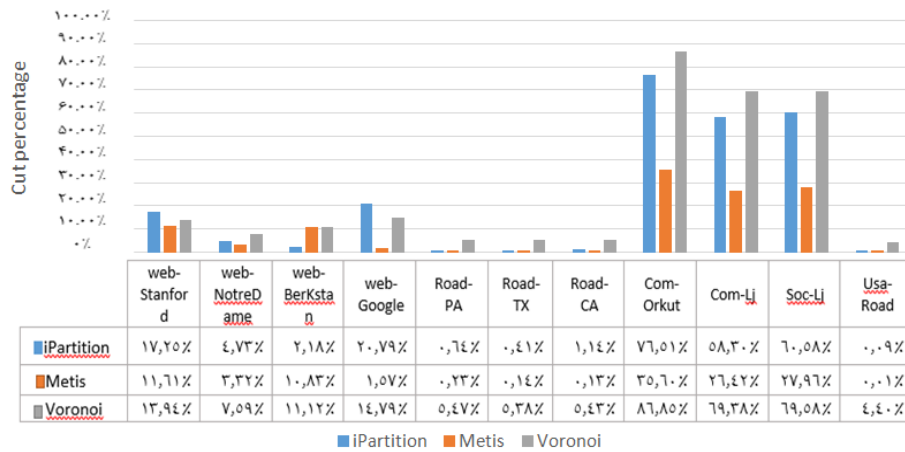
شکل ۷: گراف سطح بالای ادغام‌شده از گام دوم بخش‌بندی پیشنهادی.

جدول ۱: اطلاعات گره‌های سطح بالای پارتیشن ۱.

کلید = شناسه گره	مقدار = شناسه گره سطح بالا
۶	$P1_1$
۲۰	$P1_1$
۸	$P1_2$
۱۳	$P1_3$
۱۹	$P1_4$
۱۲	$P1_5$

سایر ماشین‌ها، درجه چسبندگی و میزان چسبندگی گره‌های سطح بالای خود را با سایر گره‌های سطح بالا محاسبه می‌نماید. این اطلاعات برای پارتیشن ۱ در جدول ۲ آورده شده است. مثلاً گره‌های ۱، ۳، ۴، ۹، ۱۶ و ۱۸ یال‌هایی با پارتیشن ۱ دارند. از آنجایی که این گره‌ها به ترتیب متعلق به گره‌های سطح بالای $P2_1$ ، $P2_2$ ، $P3_1$ ، $P3_1$ ، $P3_2$ و $P4_1$ هستند، در نتیجه ماشین ۱ بایستی رابطه گره‌های سطح بالای خود با این گره‌های سطح بالا را محاسبه نماید. به عنوان مثال چون گره‌های ۹ و ۱۶ متعلق به گره سطح بالای $P3_1$ هستند، در نتیجه درجه چسبندگی و میزان چسبندگی برابر با ۲ و ۲ خواهد بود.

در گام دوم با داشتن گراف سطح بالا، فرایند ادغام گره‌های سطح بالا آغاز می‌شود. همان طور که گفته شد، فرایند ادغام می‌تواند بر اساس ۲ پارامتر انجام شود که در این مثال از پارامتر میزان چسبندگی برای ادغام استفاده نموده‌ایم. ابتدا گره‌های سطح بالا بر اساس این پارامتر به صورت نزولی مرتب می‌شوند. در این مثال فرایند ادغام را در ۲ سطح انجام می‌دهیم و در نتیجه، حداکثر اندازه پارتیشن در سطح اول برابر ۳ و در سطح دوم برابر ۶ خواهد بود. گراف سطح بالای ادغام‌شده در شکل ۷ آمده است. اعداد نشان داده شده در آکولاد به ازای هر گره سطح بالا، اندازه آن گره سطح بالا (تعداد گره‌های آن) را نشان می‌دهند.



شکل ۹: مقایسه درصد یال‌های برشی حاصل از بخش‌بندی‌های مختلف.

۵ گراف، میزان یال برشی روش پیشنهادی بسیار نزدیک به متیس است. برای گراف Web-Berkstan تعداد یال برشی روشی پیشنهادی بسیار کمتر از متیس نیز هست.

۲-۴ قطر گراف سطح بالای حاصل از بخش‌بندی

همان طور که گفته شد، تعداد سوپرستپ‌های مورد نیاز در سیستم‌های پردازش گراف، رابطه مستقیمی با قطر گراف سطح بالای حاصل از بخش‌بندی دارد. از آنجا که هر سوپرستپ نیازمند گام همگام‌سازی است، کاهش تعداد سوپرستپ‌ها می‌تواند زمان اجرای محاسبات را در این سیستم‌ها کاهش دهد. در این آزمون، قطر گراف سطح بالای حاصل از الگوریتم‌های بخش‌بندی مختلف مورد مقایسه قرار گرفته و شکل ۱۰ نسبت به روش ورونوی بهتر عمل می‌کند. به علاوه این روش نسبت به روش متیس که اطلاعات کل گراف را داراست، دارای قطر گراف سطح بالای کمتری است و در برخی موارد نیز بسیار نزدیک هم هستند.

۳-۴ تعادل بار بین بخش‌ها

در سیستم‌های توزیع‌شده، عدم تعادل بار باعث می‌شود ماشین‌هایی که کارشان را انجام داده‌اند، منتظر پایان کار ماشین‌های دیگر بمانند و در نتیجه، عدم بهره‌وری مناسب از منابع منجر به کاهش کارایی می‌شود. در این آزمون از تعداد گره‌های هر بخش به عنوان معیار تعادل بار استفاده شده است. همان طور که گفته شد در سیستم پیشنهادی، امکان تنظیم حداکثر میزان عدم تعادل در بین بخش‌ها وجود دارد. در تمام آزمایش‌ها درجه عدم تعادل در سیستم پیشنهادی ۴ درصد در نظر گرفته شده و این در حالی است که برای سایر الگوریتم‌ها چنین تنظیمی وجود ندارد. شکل ۱۱ نتایج این آزمون را نمایش می‌دهد. یکی از نقاط ضعف اصلی ورونوی آن است که برای برخی از گراف‌ها، میزان عدم تعادل بالای ۵۰ درصد می‌باشد که می‌تواند کارایی سیستم توزیع‌شده را بسیار کاهش دهد. شایان ذکر است که می‌توان در روش پیشنهادی برای رسیدن به ۱۰٪ تعادل بار، درجه عدم تعادل را ۰ در نظر گرفت.

۵- نتیجه‌گیری و کارهای آتی

هدف سیستم‌های پردازش گراف مبتنی بر بلوک، غلبه بر مشکل محلیت موجود در سیستم‌های مبتنی بر «پریدل» می‌باشد. اگرچه این سیستم‌ها کارایی بسیار بالاتری نسبت به سایر سیستم‌های پردازش گراف دارند، اما از آنجا که بخش‌بندی‌های استفاده‌شده در این سیستم‌ها منطبق

جدول ۳: مجموعه داده‌های گرافی تحت آزمون.

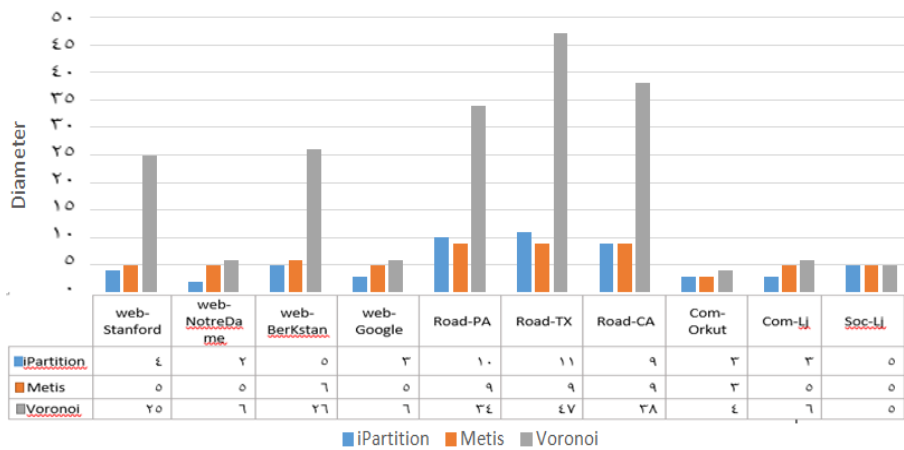
Number of Edges	Number of Vertices	Graph Name
۱۹۹۲۶۳۶	۲۸۱۹۰۳	Web-Stanford
۱۰۹۰۱۰۸	۳۲۵۷۲۹	Web-NotreDame
۶۶۴۹۴۷۰	۶۸۵۲۳۰	Web-Berkstan
۴۳۲۲۰۵۱	۸۷۵۷۱۳	Web-Google
۱۵۴۱۸۹۸	۱۰۸۸۰۹۲	Road-PA
۱۹۲۱۶۶۰	۱۳۷۹۹۱۷	Road-TX
۲۷۶۶۶۰۷	۱۹۵۶۲۰۶	Road-CA
۱۱۷۱۸۵۰۸۳	۳۰۷۳۴۴۱	Com-Orkut
۳۴۶۸۱۱۸۹	۳۹۹۷۹۶۲	Com-Lj
۴۲۸۵۱۲۳۷	۴۸۴۶۶۰۹	Soc-Lj
۲۸۸۵۴۳۱۲	۲۳۹۴۷۳۴۷	Usa-Road

جدول ۳ گراف‌های مورد استفاده به منظور ارزیابی سیستم پیشنهادی را نشان می‌دهد. در انتخاب مجموعه داده سعی گردیده که گراف‌ها متعلق به حوزه‌ها و کاربردهای مختلف باشند. این مجموعه داده از [۴۵] قابل دسترس است. از آنجا که گره‌ها در گام اول روش پیشنهادی به صورت تصادفی بخش‌بندی می‌شوند، در راستای کاهش وابستگی به چگونگی کیفیت بخش‌بندی تصادفی، هر آزمون ۱۰ بار (با بخش‌بندی تصادفی مختلف) اجرا گردیده و نتایج ارائه‌شده، میانگین نتایج حاصل از این ۱۰ آزمون می‌باشد.

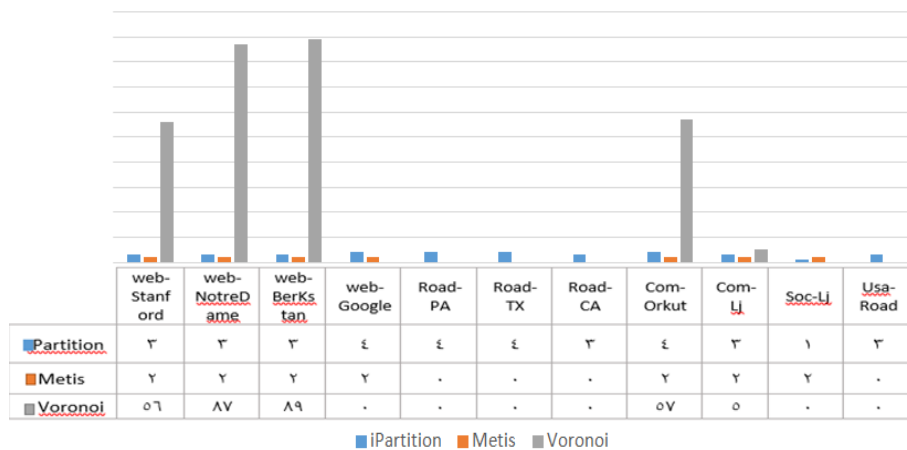
۱-۴ تعداد یال‌های برشی

همان طور که گفته شد، یکی از معیارهایی که توسط اغلب روش‌های بخش‌بندی گراف مورد توجه قرار گرفته است، تعداد یال‌های برشی حاصل از بخش‌بندی می‌باشد. این بدان جهت است که هر چه تعداد یال برشی بین بخش‌ها کمتر باشد، تعداد پیام‌های کمتری بین بخش‌ها رد و بدل شده و بنابراین میزان ترافیک شبکه کاهش می‌یابد. از آنجا که تعاملات شبکه به عنوان یک گلوگاه در سیستم‌های پردازش گراف است، کاهش ترافیک شبکه، کارایی سیستم‌های پردازش گراف را افزایش می‌دهد.

شکل ۹ درصد یال‌های برشی را برای گراف‌های مختلف نشان می‌دهد. از آنجا که متیس روشی متمرکز است و اطلاعات کل گراف را داراست و با هدف اصلی کاهش تعداد یال‌های برشی طراحی شده است، حداقل تعداد یال برشی را در بین الگوریتم‌های دیگر داراست. روش پیشنهادی اگرچه با هدف کاهش قطر گراف سطح بالا طراحی شده است، اما در تمامی موارد نسبت به الگوریتم ورونوی، تعداد یال برشی کمتری دارد.



شکل ۱۰: مقایسه قطر گراف سطح بالای حاصل از بخش‌بندی‌های مختلف.



شکل ۱۱: مقایسه تعادل بار بین بخش‌های حاصل از بخش‌بندی‌های مختلف.

قطر گراف سطح بالا ارائه نماید. به علاوه رهیافت‌های بخش‌بندی پویا به منظور مهاجرت گره‌های سطح بالا برای بخش‌بندی گراف‌های پویا مد نظر قرار خواهند گرفت. گره‌هایی که دارای درجه خروجی بسیار بالایی هستند، منجر به تولید یال‌های برشی زیادی می‌شوند. برخی روش‌های بخش‌بندی برای کاهش تأثیر چنین گره‌هایی، آنها را در بیش از یک پارتیشن تکثیر می‌کنند. این بهینه‌سازی را می‌توان به روش پیشنهادی نیز به منظور کاهش میزان یال برشی اضافه نمود.

مراجع

- [1] M. Ulizko, E. Antonov, A. Artamonov, et al., "Graph visualization of the characteristics of complex objects on the example of the analysis of politicians," in *Proc. 30th Int. Conf. on Computer Graphics and Machine Vision*, 9 pp., St. Petersburg, Russia Federation, 22-25 Sept. 2020.
- [2] J. S. Yeom, et al., "Overcoming the scalability challenges of epidemic simulations on blue waters," in *Proc. IEEE 28th Int. Parallel and Distributed Processing Symp.*, pp. 755-764, Phoenix, AZ, USA, 19-23 May 2014.
- [3] R. Hess, V. H. Visschers, and M. Siegrist, "Risk communication with pictographs: the role of numeracy and graph processing," *Judgment and Decision Making*, vol. 6, no. 3, pp. 263-274, Apr. 2011.
- [4] V. Agarwal, F. Petrini, D. Pasetto, D. A. Bader, "Scalable graph exploration on multicore processors," in *Proc. of the ACM/IEEE International Conf. for High Performance Computing, Networking, Storage and Analysis*, 11 pp., New Orleans, LA, USA, 13-19 Nov. 2010.
- [5] Y. Wang, Y. Shangdi, L. Dhulipala, Y. Gu, and J. Shun, "GeoGraph: a framework for graph processing on geometric data," *ACM SIGOPS Operating Systems Review*, vol. 55, no. 1, pp. 38-46, 2021.

بر ویژگی‌های آنها نیست، در نتیجه حداکثر کارایی حاصل نمی‌شود. به همین منظور در این مقاله با توجه به ویژگی‌های مشترک موجود در این سیستم‌ها، دو معیار جدید از نقطه‌نظر بخش‌بندی ارائه شد. نشان داده شد که قطر گراف سطح بالا و همچنین انحراف معیار اندازه گره‌های سطح بالا می‌توانند تأثیر چشم‌گیری در کاهش ترافیک شبکه و همچنین تعداد سوپراستپ‌های مورد نیاز به منظور تکمیل محاسبات در این سیستم‌ها داشته باشند. با توجه به این معیارها، یک الگوریتم بخش‌بندی توزیع‌شده در این مقاله پیشنهاد گردید. ارزیابی روش پیشنهادی بر روی گراف‌های واقعی نشان داد که روش پیشنهادی اگرچه با هدف کاهش قطر گراف سطح بالای حاصل از بخش‌بندی، بهینه شده و قطر گراف سطح بالا را بسیار کاهش می‌دهد، تعداد یال برشی و همچنین تعادل بار بسیار بهتری نسبت به روش‌های بخش‌بندی ارائه‌شده برای سیستم‌های مبتنی بر بلوک از جمله ورونوی دارد. به علاوه میزان یال برشی و همچنین تعادل بار بسیار نزدیکی با روش‌های نسبتاً دقیق متمرکز نظیر متیس داراست.

به عنوان کارهای آینده در نظر داریم که نتیجه بخش‌بندی حاصل از روش پیشنهادی را بر روی سیستم‌های پردازش گراف مبتنی بر بلوک نظیر «ایکس پریبل» و بلوکل اعمال کرده و نتایج آن را با طیف بیشتری از الگوریتم‌های بخش‌بندی مقایسه نماییم. یکی از مشکلات الگوریتم‌های بخش‌بندی موجود آن است که بایستی تعداد پارتیشن مورد نیاز به عنوان ورودی الگوریتم داده شود. یکی از تغییراتی که می‌توان برای الگوریتم پیشنهادی به عنوان کار آینده در نظر گرفت، آن است که الگوریتم بتواند با توجه به تعداد پارتیشن و حداکثر قطر گراف، تقابلی بین این دو در نظر گرفته و سعی کند که یک بخش‌بندی با حداکثر تعداد پارتیشن و حداقل

- [28] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *The Bell System Technical J.*, vol. 49, no. 2, pp. 291-307, Feb. 1970.
- [29] Y. G. Saab, "An effective multilevel algorithm for bisecting graphs and hypergraphs," *IEEE Trans. on Computers*, vol. 53, no. 6, pp. 641-652, Jun. 2004.
- [30] L. Sun and M. Leng, "An effective multi-level algorithm based on simulated annealing for bisecting graph," in *Proc. Int. Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 12 pp., Ezhou, China, 27-29 Aug. 2007.
- [31] C. Aykanat, B. B. Cambazoglu, and B. Uçar, "Multi-level direct k-way hypergraph partitioning with multiple constraints and fixed vertices," *J. of Parallel and Distributed Computing*, vol. 68, no. 5, pp. 609-625, May 2008.
- [32] R. Khandekar, S. Rao, and U. Vazirani, "Graph partitioning using single commodity flows," *J. of the ACM*, vol. 56, no. 4, pp. 385-390, 2009.
- [33] H. Meyerhenke, B. Monien, and S. Schamberger, "Graph partitioning and disturbed diffusion," *Parallel Computing*, vol. 35, no. 10-11, pp. 544-569, Oct. 2009.
- [34] P. Chardaire, M. Barake, and G. P. McKeown, "A probe-based heuristic for graph partitioning," *IEEE Trans. on Computers*, vol. 56, no. 12, pp. 1707-1720, Dec. 2007.
- [35] R. Zamprogno and A. R. Amaral, "An efficient approach for large scale graph partitioning," *J. of Combinatorial Optimization*, vol. 13, no. 4, pp. 289-320, May 2007.
- [36] A. Trifunovic and W. J. Knottenbelt, "Towards a parallel disk-based algorithm for multilevel k-way hypergraph partitioning," in *Proc. IEEE 18th Int. Parallel and Distributed Processing Symp.*, pp. 236-243, Santa Fe, NM, USA, 26-30 Apr. 2004.
- [37] I. Stanton and G. Kliot, "Streaming graph partitioning for large distributed graphs," in *Proc. of the 18th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 1222-1230, Beijing, China, 12-16 Aug. 2012.
- [38] C. Tsourakakis, C. Gkantsidis, B. Radunovic, M. Vojnovic, "Fennel: streaming graph partitioning for massive scale graphs," in *Proc. of the 7th ACM Int. Conf. on Web Search and Data Mining*, pp. 333-342, New York, NY, USA, 24-26 Feb. 2014.
- [39] Y. Tian, A. Balmin, S. A. Corsten, S. Tatikonda, and J. McPherson, "From 'think like a vertex' to 'think like a graph'," *Proc. of the VLDB Endowment*, vol. 7, no. 3, pp. 193-204, Nov. 2013.
- [40] W. Fan, J. Xu, Y. Wu, W. Yu, and J. Jiang, "GRAPE: parallelizing sequential graph computations," *Proc. of the VLDB Endowment*, vol. 10, no. 12, pp. 1889-1892, Aug. 2017.
- [41] S. Verma, L. M. Leslie, Y. Shin, and I. Gupta, "An experimental comparison of partitioning strategies in distributed graph processing," *Proc. of the VLDB Endowment*, vol. 10, pp. 493-504, Jan. 2017.
- [42] Y. Kim, M. Bae, and S. Oh, "Dynamic block reassignment for load balancing of block centric graph processing systems," *KIPS Trans. on Software and Data Engineering*, vol. 7, no. 5, pp. 177-188, 2018.
- [43] X. Wen, S. Zhang, and H. You, *DRONE: A Distributed Subgraph-Centric Framework for Processing Large Scale Power-law Graphs*, arXiv preprint arXiv:1812.04380, 2018.
- [44] M. Sagharichian, M. A. Langouri, and H. Naderi, "A fast method to exactly calculate the diameter of incremental disconnected graphs," *World Wide Web*, vol. 20, no. 2, pp. 399-416, Mar. 2017.
- [45] Stanford University, *Stanford Large Network Dataset Collection*, Available from: <http://snap.stanford.edu/data>.
- [6] G. Malewicz, et al., "Pregel: a system for large-scale graph processing," in *Proc. of the ACM SIGMOD Int. Conf. on Management of Datapp*. 135-146, Indianapolis, IN, USA, 6-10 Jun. 2010.
- [7] S. Salihoglu and J. Widom, "GPS: a graph processing system," in *Proc. of the 25th Int. Conf. on Scientific and Statistical Database Management*, Article No.: 22, 12 pp., Baltimore, MA, USA, 29-31 Jul. 2013.
- [8] J. E. Gonzalez, et al., "Graphx: Graph processing in a distributed dataflow framework," in *Proc. 11th USENIX Symp. on Operating Systems Design and Implementation, OSDI'14*, pp. 599-613, Broomfield, CO, USA, 6-8 Oct. 2014.
- [9] S. Hong, et al., "PGX.D: a fast distributed graph processing engine," in *Proc. of the Int. Conf. for High Performance Computing, Networking, Storage and Analysis*, 12 pp., Austin, TX, USA, 15-20 Nov. 2015.
- [10] R. R. McCune, T. Weninger, and G. Madey, "Thinking like a vertex: a survey of vertex-centric frameworks for large-scale distributed graph processing," *ACM Computing Surveys*, vol. 48, no. 2, Article No.: 25, 39 pp., Nov. 2015.
- [11] D. Yan, J. Cheng, Y. Lu, and W. Ng, "Blogel: a block-centric framework for distributed computation on real-world graphs," *Proc. of the VLDB Endowment*, vol. 7, no. 14, pp. 1981-1992, Oct. 2014.
- [12] M. Sagharichian, H. Naderi, and M. Haghjoo, "ExPregel: a new computational model for large-scale graph processing," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 17, pp. 4954-4969, Dec. 2015.
- [13] S. Aridhi, A. Montresor, and Y. Velegrakis, "BLADYG: a novel block-centric framework for the analysis of large dynamic graphs," in *Proc. of the ACM Workshop on High Performance Graph Processing*, pp. 39-42, Kyoto, Japan, 31-31 May 2016.
- [14] R. Dindokar, N. Choudhury, and Y. Simmhan, "A meta-graph approach to analyze subgraph-centric distributed programming models," in *Proc. IEEE Int. Conf. on Big Data*, pp. 37-47, Washington, DC, USA, 5-8 Dec. 2016.
- [15] A. Quamar and A. Deshpande, "NScaleSpark: subgraph-centric graph analytics on Apache Spark," in *Proc. of the 1st ACM SIGMOD Workshop on Network Data Analytics*, Article No.: 5, 8 pp., San Francisco, CA, USA, 1-1 Jul. 2016.
- [16] M. Sagharichian and H. Naderi, "Intelligent and independent processes for overcoming big graphs," *The J. of Supercomputing*, vol. 73, no. 4, pp. 1438-1466, Apr. 2017.
- [17] X. Sui, D. Nguyen, M. Burtcher, and K. Pingali, "Parallel graph partitioning on multicore architectures," in *Proc. Int. Workshop on Languages and Compilers for Parallel Computing*, pp. 246-260, Houston, TX, USA, 7-9 Oct. 2010.
- [18] D. LaSalle and G. Karypis, "Multi-threaded graph partitioning," *IEEE 27th Int. Symp. on Parallel and Distributed Processing*, pp. 225-236, Cambridge, MA, USA, 20-24 May 2013.
- [19] M. Naumov and T. Moon, *Parallel Spectral Graph Partitioning*, Tech. Rep., NVIDIA Tech. Rep. 2016.
- [20] H. Meyerhenke, P. Sanders, and C. Schulz, "Parallel graph partitioning for complex networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 28, no. 9, pp. 2625-2638, Sept. 2017.
- [21] F. Rahimian, A. H. Payberah, S. Girdzijauskas, M. Jelasity, and S. Haridi, "JA-BE-JA: A distributed algorithm for balanced graph partitioning," in *Proc. IEEE 7th Int. Conf. on Self-Adaptive and Self-Organizing Systems*, pp. 51-60, Philadelphia, PA, USA, 9-13 Sept. 2013.
- [22] C. Martella, D. Logothetis, A. Loukas, and G. Siganos, "Spinner: scalable graph partitioning in the cloud," in *Proc. IEEE 33rd International Conf. on Data Engineering, ICDE'17*, pp. 1083-1094, San Diego, CA, USA, 19-22 Apr. 2017.
- [23] M. Onizuka, T. Fujimori, and H. Shiokawa, "Graph partitioning for distributed graph processing," *Data Science and Engineering*, vol. 2, no. 1, pp. 94-105, Mar. 2017.
- [24] J. Sun, H. Vandierendonck, and D. S. Nikolopoulos, "Graphgrind: addressing load imbalance of graph partitioning," in *Proc. of the Int. Conf. on Supercomputing*, Article No.: 16, 10 pp., Chicago, IL, USA, 14-16 Jun. 2017.
- [25] C. J. Alpert and A. B. Kahng, "Recent directions in netlist partitioning: a survey," *Integration*, vol. 19, no. 1-2, pp. 1-81, 1995.
- [26] B. Hendrickson and T. G. Kolda, "Graph partitioning models for parallel computing," *Parallel Computing*, vol. 26, no. 12, pp. 1519-1534, 2000.
- [27] T. N. Bui and C. Jones, "Finding good approximate vertex and edge partitions is NP-hard," *Information Processing Letters*, vol. 42, no. 3, pp. 153-159, May 1992.

مسعود ساغریچیان تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد و دکتری مهندسی کامپیوتر به ترتیب در سال‌های ۱۳۸۸ و ۱۳۹۰ و ۱۳۹۵ از دانشگاه علم و صنعت ایران به پایان رسانده است و هم‌اکنون استاد دانشکده مهندسی کامپیوتر دانشگاه الزهراء می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: کلان داده‌ها، پردازش گراف و شبکه‌های یادگیری عمیق.

مرتضی علیپور لنگروی در سال ۱۳۹۵ مدرک کارشناسی ارشد مهندسی کامپیوتر خود را از دانشگاه علم و صنعت ایران دریافت نمود و در حال حاضر دانشجوی دکتری دانشگاه مکتستر کانادا می‌باشد. زمینه‌های علمی مورد علاقه نام‌برده عبارتند از: پردازش توزیع‌شده و یادگیری ماشینی.