مجله علمی پژوهشی «علوم و فناوری‌های پدافند غیرعامل»
سال سوم، شماره۲، تابستان ۱۳۹۱؛ ص ۱۵۸-۱۴۵

معرفی الگوریتم خلاصه‌سازی معناگرای SIGS برای گراف‌های عظیم‌الجثه

محمد مهدی ارسنجانی*[۱]، محمد رضا کنگاوری[۲]

۱- دانشجوی کارشناسی ارشد، ۲- استادیار، دانشکده کامپیوتر، دانشگاه علم و صنعت ایران

(دریافت: ۱۳۹۰/۰۹/۱۲، پذیرش: ۱۳۹۱/۰۵/۲۰)

چکیده

امروزه گراف‌ها به‌طور گسترده در بسیاری از حوزه‌ها از جمله نرم‌افزار، شبکه، وب، شیمی، زیست، ژنتیک و حتی مخابرات و جامعه‌شناسی برای مدل‌سازی و پردازش داده‌ها استفاده می‌شوند. حجیم بودن و پیچیدگی زیاد گراف‌های داده، یکی از مهم‌ترین چالش‌ها در این زمینه است که کار استخراج اطلاعات و دانش مورد نیاز از میان مجموعه‌ای از داده‌ها را بسیار مشکل می‌سازد. در چنین شرایطی، استفاده از الگوریتم‌های خلاصه‌سازی گراف می‌تواند راه‌حل مناسبی باشد. در مقاله حاضر، الگوریتمی نو برای خلاصه‌سازی گراف‌ها ارائه شده که قادر است از یک گراف برحسب نیاز کاربران، خلاصه‌های مختلف با جزئیات متفاوت تولید کند. به‌علاوه، کاربر قادر است سطح خلاصه‌سازی را نیز کنترل کند. الگوریتم معرفی‌شده، با استفاده از پایگاه‌داده Neo4j که یکی از انواع پایگاه‌های داده‌ای غیررابطه‌ای است پیاده‌سازی شده است. همچنین آزمون‌هایی به‌وسیله داده‌های آزمایشگاهی و واقعی برای ارزیابی الگوریتم انجام گرفته است که نشان می‌دهد خلاصه‌های تولیدشده، از کیفیت مناسبی برخوردار هستند. ضمن آنکه این الگوریتم از لحاظ مقیاس‌پذیری و کارایی از نمونه مشابه خود کیفیت بهتری ارائه می‌کند.

کلیدواژه‌ها: گراف، خلاصه‌سازی معناگرا، گراف خلاصه، مقیاس‌پذیری، کارایی.

Introducing a Novel Algorithm SISG to Semantically Summarize Massive Graphs

M. M. Arsanjani*, M. R. Kangavari

Department of Computer, Iran University of Science and Technology
(Received: 12/03/2011; Accepted: 08/10/2012)

Abstract

Nowadays graphs are widely used in many domains such as software, network, web, chemistry, biology and even communication and sociology to modelling and data processing. In many applications, graphs are very large and complex. So understanding the structure and extracting useful information from them is become more challenging. Here, graph summarization algorithms could be a suitable solution. In this paper, a new graph summarization algorithm has been proposed which is able to produce different summaries from different points of view from one graph regarding to user's interested subjects. Also users can control the resolution of produced summaries. Moreover, the algorithm is developed using Neo4j database which is one of NoSQL databases. Also, the algorithm using different laboratorial and real data sets is tested. The results show that the produced summaries are in high quality position and also the efficiency and scalability of the algorithm is better that the similar one.

Keywords: Graph, Semantical Summarization, Summary Graph, Scalability, Efficiency.

* Corresponding author E-mail: m.arsanjani@gmail.com

۱. مقدمه

امروزه حجم و تنوع داده‌های موجود در جهان با سرعت چشم‌گیری در حال افزایش است. به‌همین دلیل، یافتن داده‌های مفید از میان انبوهی از داده ها و استخراج اطلاعات و دانش مورد نیاز از میان آنها، به چالشی اساسی برای محققین در زمینه‌های مختلف تبدیل شده است. در همین زمینه، مسئله یافتن راه‌حلی مناسب برای مدل‌سازی، نمایش و تحلیل داده های حجیم و متنوع نیز موضوعی بسیار مهم به‌شمار می رود. در میان تمام ساختارهای موجود، گراف ها با توجه به خواص منحصر به‌فرد خود، محبوبیت خاصی در میان محققین پیدا کرده‌اند.

با استفاده از گراف، می‌توان داده‌های بسیار متنوع با پیچیدگی زیاد را به‌سادگی نمایش داد. سادگی و معنادار بودن گراف و همچنین قدرت این ساختار در نمایش روابط پیچیده میان موجودیت ها، در کنار قابلیت انعطاف‌پذیری و مقیاس‌پذیری آن، این ساختار را به مدلی مناسب برای نمایش داده‌های متنوع و حجیم با روابط پیچیده تبدیل کرده است، به‌گونه‌ای که امروزه در بیشتر حوزه‌ها، از گراف به‌عنوان مدل نمایش داده‌ها استفاده می‌شود. گراف وب، گراف فراخوانی‌ها در یک نرم‌افزار، نمودارهای UML، گراف اتصالات شبکه‌های ارتباطی، گراف اعضای یک شبکه اجتماعی و گراف ترکیبات شیمیایی و زیستی تنها برخی از نمونه های کاربرد گراف ها در حوزه‌های مختلف علمی هستند.

یکی از چالش های اساسی در زمینه گراف ها، بزرگ شدن حجم گراف داده‌ها است که گاه شامل میلیون‌ها رأس و یال است. این موضوع، فهم گراف و یافتن داده های مورد نیاز را برای کاربران بسیار مشکل می‌سازد، به‌گونه‌ای که گاه حتی بسیاری از برنامه‌های کامپیوتری نیز به‌سختی قادر به پردازش و تحلیل این‌گونه گراف ها هستند [۱].

گراف‌های داده‌ای موجود در فضای وب و اینترنت مانند گراف وب و گراف‌های شبکه‌های اجتماعی، نمونه‌های واضحی از این گراف‌ها هستند که امروزه با توجه به گسترش روزافزون داده های موجود در وب، مورد توجه بسیاری از محققین در حوزه های مختلف قرار گرفته‌اند.

موتورهای جستجو از جمله مشهورترین نرم‌افزارهایی هستند که نیاز مبرمی به توانایی کاوش و استخراج اطلاعات گوناگون از این گراف ها دارند. در واقع این نرم افزارها، برای تحلیل بهتر صفحات وب و اولویت‌بندی صحیح‌تر آنها برای پاسخ به پرس‌وجوهای کاربران، نیازمند راه‌حل‌هایی برای پردازش و تحلیل این گراف‌های عظیم هستند. کاوش گراف‌های وب و شبکه‌های اجتماعی، همچنین در حوزه‌های دیگری مانند امنیت، سیاست و حتی فرهنگ از اهمیت بالایی برخوردار است[۴-۲].

امروزه با به‌وجود آمدن مفاهیم جدیدی مانند جنگ نرم، فضای سایبری، تهاجم فرهنگی، تأثیرات رسانه و امثال اینها، فضایی مانند اینترنت که گستره‌ای جهانی داشته و از تأثیرگذارترین رسانه‌های دنیای امروز محسوب می‌شوند، بیش از پیش مورد توجه دولت‌ها و نهادهای گوناگون در سراسر دنیا قرار گرفته‌اند. در این میان، دولت‌هایی مانند دولت ایالات متحده آمریکا یا نهادهایی مانند موساد و سی‌آی‌ای که از پشتیبانی ابزارهایی بسیار قوی در فضای اینترنت مانند موتورهای جستجوی گوگل و یاهو یا شبکه‌های اجتماعی گسترده مانند فیس‌بوک و تویتر بهره می‌برند، قدرت بیشتری نیز در کنترل این فضا و تأثیرگذاری بر افکار مخاطبان این رسانه جهان‌گستر دارا می‌باشند.

تلاش‌های گسترده کشورهایی مانند روسیه، چین و کره‌جنوبی در تولید موتورهای جستجوی بومی و شبکه‌های اجتماعی ملی که با هدف دستیابی به استقلال و افزایش میزان کنترل و نیز تأثیرگذاری در فضای وب انجام گرفته است خود گواه دیگری بر اهمیت بسیار بالای داده‌های موجود در اینترنت است. راه‌اندازی موتورهای جستجوی ملی و شبکه‌های مجازی اینترنتی مانند یندکس[1] (روسیه)، بایدو[2] (چین) و نِیور[3] (کره‌جنوبی) به‌وسیله برخی از این کشورها نمونه‌هایی از این تلاش هستند. حتی بسیاری از دولت‌ها و نهادها می‌کوشند تا با صرف مبالغ بسیار هنگفت، تنها گوشه‌ای از داده‌هایی را که موتورهای جستجوی مطرح دنیا مانند گوگل و یاهو یا شبکه های اجتماعی نظیر فیس‌بوک یا تویتر در اختیار دارند را به‌دست‌آورند و با کاوش و تحلیل آنها، فضای فرهنگی و سیاسی جامعه خود را بهتر شناخته و برای هدایت آن برنامه‌ریزی کنند.

امروزه حتی شاهد هستیم که فضاهای اینترنتی مانند وبلاگ‌ها و شبکه های اجتماعی، به ابزاری برای برقراری ارتباط و اطلاع رسانی و بیدارسازی میان اعضای قیام های مردمی در سرتاسر دنیا تبدیل شده اند که حتی با وجود اعمال محدودیت های فراوان و ایجاد خفقان‌های گسترده از سوی دولت‌ها، هم‌چنان این قیام‌ها را زنده و مؤثر نگه داشته است و از طرف دیگر اینترنت را به مخزنی بزرگ از اطلاعات مفید درباره این ملت ها و قیام هایشان تبدیل کرده است. در چنین شرایطی، در اختیار داشتن ابزارهای توانمند برای گردآوری داده از اینترنت و تحلیل و کاوش این داده ها با هدف استخراج اطلاعات، دانش مفید و موردنیاز، از ضروری‌ترین نیازهای دولت‌ها و نهادهای گوناگون محسوب می‌شود.

علاوه‌بر داده‌های سیاسی و فرهنگی، امروزه بسیاری از جدیدترین مطالب علمی دنیا در حوزه‌های گوناگون نیز از طریق اینترنت قابل دسترسی است. مقالات و کتب علمی، پس از گذشت اندک زمانی از انتشار، روی اینترنت قرار می‌گیرند. تمام دانشگاه‌ها و انجمن‌های علمی معتبر دنیا در اینترنت دارای سایت‌های مخصوص هستند و از این طریق اطلاعات ارزنده‌ای درباره آخرین یافته‌ها و دستاوردهای خود را عرضه می‌کنند. بنابراین جامعه‌ای که قصد دارد وارد رقابت جدی در فضای علمی دنیا شود، می‌بایست حداکثر بهره‌برداری را از داده‌های موجود در این فضا انجام دهد [۵].

از طرفی به‌طور تقریبی تمام داده‌های موجود در فضای اینترنت از

[1] http:// www.yandex.ru
[2] http:// http://www.baidu.com
[3] http:// http://www.naver.com

مدل گراف تبعیت می‌کنند. صفحات وب و پیوندهای میان آنها، شبکه‌های اجتماعی و اعضای آنها و روابط میان این اعضا، دانشگاه‌ها و مراکز علمی و پیوندها و مطالب مشترک میان آنها، همگی انواعی از مدل کلی گراف هستند که در حوزه اینترنت از حجم و پیچیدگی قابل ملاحظه‌ای برخوردارند.

اینجاست که توانایی پردازش، تحلیل و کاوش گراف های عظیم و پیچیده، اهمیت خود را در حوزه‌هایی مانند کنترل فضای سایبری، جنگ نرم، تهاجم فرهنگی، رقابت‌ها، تحقیقات علمی و حتی تحلیل و شناخت تفکرات مردم و سایر اقوام و ملل نشان می دهد. واضح است که بیشتر این زمینه ها با حوزه پدافند غیرعامل در ارتباط هستند. بنابراین افزایش قدرت پردازش و کاوش داده‌ها در این حوزه‌ها، منجر به کسب دانشی صحیح تر و به‌تبع آن افزایش دقت و سرعت در تصمیم‌گیری‌های مربوط به حوزه‌های مورد نظر و درنهایت افزایش سرعت پیشرفت در این زمینه‌ها خواهد شد.

اما همان‌طور که گفته‌شد، چالش اساسی در این مسیر، حجم بسیار زیاد داده‌های موجود و پیچیدگی روابط میان موجودیت‌ها است که مدل‌های گرافی داده‌ها را به گراف‌هایی عظیم‌الجثه و بسیار پیچیده تبدیل می‌کند که پردازش و کاوش آن‌ها با استفاده از روش‌های معمول داده‌کاوی امکان‌پذیر نبوده و یا مستلزم صرف هزینه‌های بسیار بالا است.

در چنین شرایطی می‌توان با حذف داده‌های غیر مرتبط و ساده‌تر کردن ساختار گراف، کاربر را در یافتن اطلاعات مورد نیازش یاری کرد. به‌عبارت دیگر، تولید یک خلاصه از گراف اولیه که حاوی اطلاعات مورد نیاز کاربر بوده و نیز اطلاعات و خصوصیات مهم موجود در گراف اولیه را دربر داشته باشد، راه‌حلی مناسب برای چالش مذکور محسوب می‌شود.

در واقع بسیاری از الگوریتم‌های کاوش گراف[1] نیز با چنین هدفی ایجاد شده‌اند. الگوریتم‌های کاوش الگوهای متناوب[2] نمونه‌ای از الگوریتم‌های خلاصه‌سازی هستند که زیرگراف‌های تکرارشونده را در یک یا مجموعه‌ای از گراف‌ها یافته و به کاربر عرضه می‌کنند[۱۳-۶]. الگوریتم‌های خوشه‌بندی گراف[3] نیز نمونه‌ای دیگر از الگوریتم‌هایی هستند که به‌نوعی سعی در تهیه خلاصه‌ای از ساختار گراف دارند [۳ و ۱۶-۱۴]. این الگوریتم‌ها در واقع با یافتن رئوس مشابه در گراف، آنها را در یک خوشه قرار داده و درنهایت مجموعه‌ای از خوشه‌ها را به کاربر عرضه می‌کنند.

اما با وجود گذشت زمان قابل توجهی از زمان پیدایش الگوریتم‌های کاوش گراف و نیز با وجود تنوع کارهای انجام‌گرفته در این حوزه، الگوریتم‌های مذکور در زمینه خلاصه‌سازی گراف هم‌چنان ضعف‌ها و کاستی‌های قابل توجهی دارند. اول آن‌که خروجی این الگوریتم‌ها غالباً گراف نیست. برای مثال، نتیجه حاصل از یک الگوریتم کاوش الگوهای متناوب، مجموعه‌ای از زیرگراف های تکرارشونده در یک یا چند گراف است. یا نتیجه به‌دست آمده از اجرای یک الگوریتم خوشه‌بندی، مجموعه‌ای از خوشه‌هاست که هر یک تعدادی از رئوس گراف اولیه را دربر دارند. دوم آن‌که، الگوریتم های مذکور به ازای هر گراف، تنها قادر به تولید یک مجموعه جواب هستند.

این در حالی است که در بسیاری مواقع، کاربران مختلف نیاز به استخراج اطلاعات متنوع از یک گراف دارند. بنابراین در چنین مواقعی مجبور خواهند بود هر یک به‌طور جداگانه الگوریتم متناسب با نیاز خود را یافته و آن را روی گراف اجرا کنند. سوم و از همه مهم‌تر آنکه، غالب الگوریتم‌های کاوش گراف، با فرض امکان بارگذاری تمام ساختار گراف اولیه روی حافظه اصلی طراحی شده‌اند [۹]. در حالی که در مسائل واقعی، گراف ها غالباً آنقدر حجیم و بزرگ هستند که بارگذاری تمام ساختار آنها روی حافظه در عمل امکان‌پذیر نیست. بنابراین الگوریتم‌های کاوش گراف، نمی‌توانند راه‌حل مناسبی برای موضوع خلاصه‌سازی گراف‌های بزرگ باشند.

خلاصه‌سازی به‌معنای تولید نسخه مختصرتری از یک شیء است که با وجود حجم کمتر، نکات و خصوصیات مهم و اصلی موجود در شیء اولیه را دربر داشته باشد[۱۷]. بنابراین منظور از یک الگوریتم خلاصه‌سازی گراف، الگوریتمی است که بتواند یک گراف عظیم‌الجثه را به‌گونه‌ای خلاصه کند که حاصل، گرافی ساده‌تر و کوچک‌تر باشد و در عین حال تمامی خصوصیات و اطلاعات برجسته موجود در گراف اولیه را داشته باشد. همچنین این خلاصه می بایست با توجه به اطلاعات مورد نیاز کاربر تولید شده باشد. به‌عبارت دیگر، الگوریتم خلاصه‌سازی می‌بایست متناسب با نیاز کاربران مختلف، خلاصه‌های مختلفی از یک گراف را تولید کند.

در مقاله حاضر، در پی ارائه یک الگوریتم خلاصه سازی هستیم به‌گونه‌ای که بتواند گراف‌های عظیم را برحسب نیاز کاربران به‌گونه‌ای خلاصه کند که در ابتدا، خروجی، یک گراف ساده‌تر و کوچک‌تر از گراف اولیه باشد. سپس کاربر بتواند با توجه به اطلاعات مورد نیاز خود، خلاصه‌های مختلفی از یک گراف تولید کند، به‌طوری‌که هرکدام حاوی داده‌ها و اطلاعات مورد علاقه کاربر باشد. ضمن آنکه الگوریتم مذکور می‌بایست از لحاظ سرعت و کارایی در حد قابل قبولی بوده و بتواند گراف‌های بزرگ با هزاران و حتی میلیون‌ها رأس را بدون نیاز به سخت‌افزارهای سنگین و خاص‌منظوره و در زمان مناسبی خلاصه کند. بر این اساس، در این مقاله، پس از بررسی اجمالی کارهای انجام‌گرفته در زمینه خلاصه‌سازی گراف که تا حدودی به اهداف فوق دست‌یافته‌اند، الگوریتم خلاصه‌سازی SIGS[4] به‌عنوان یک الگوریتم مناسب برای خلاصه سازی گراف که قادر است تمام اهداف فوق را به‌نحو احسن ارضا کند، معرفی شده است.

در ادامه این مقاله و در بخش ۲، مروری اجمالی بر کارهای انجام‌شده در حوزه خلاصه‌سازی گراف‌ها صورت گرفته است. پس از آن در بخش ۳، مبانی اولیه و طراحی الگوریتم خلاصه‌سازی SIGS ارائه شده است. سپس در بخش ۴، چارچوب و بستری مناسب برای

[1] Graph Mining
[2] Frequent Pattern Mining
[3] Graph Clustering
[4] Semantic Based Interactive Graph Summarization

پیاده‌سازی و اجرای یک الگوریتم خلاصه‌سازی معرفی شده که الگوریتم SIGS با استفاده از آن پیاده‌سازی شده است. همچنین در بخش ۵، الگوریتم SIGS به‌وسیله داده‌های واقعی آزمایش شده و نتیجه‌های حاصل از آزمون ها به‌همراه تحلیل آنها به بحث گذاشته شده است. درنهایت در بخش‌های ۶ و ۷ یک جمع‌بندی از مباحث مطرح شده به‌همراه بحث مختصری پیرامون مسائل باقی‌مانده و پیشنهاداتی برای کارهای آتی عنوان شده است.

۲. پیشینه تحقیق

امروزه با توجه به گسترش کاربرد گراف و به‌وجود آمدن گراف های عظیم، موضوع خلاصه‌سازی گراف به‌طور جدی مورد توجه قرار گرفته و به حوزه ای جدی برای تحقیق و کار بدل شده است. به‌طورکلی می‌توان خلاصه‌سازی گراف را به دو صورت درنظر گرفت: خلاصه‌سازی ساختاری و خلاصه‌سازی معناگرا.

در خلاصه سازی ساختاری، ویژگی ها و اطلاعات به‌دست آمده از ساختار گراف، مانند همسایگی ها، درجه رئوس و چگالی زیرگراف ها مبنای تولید خلاصه هستند. بنابراین با توجه به ثابت بودن ساختار گراف که منجر به ثابت بودن خصوصیات ساختاری نیز می‌شود، الگوریتم های خلاصه‌سازی ساختاری، به ازای هر گراف و با توجه به مبنای انتخاب شده توسط طراحان الگوریتم، تنها قادر هستند از هر گراف، یک خلاصه تولید کنند. الگوریتم های کاوش الگوهای متناوب و خوشه‌بندی گراف را می‌توان از انواع چنین خلاصه‌سازی‌هایی دانست.

نمونه‌هایی دیگر از الگوریتم‌های خلاصه‌سازی ساختاری گراف هستند که بیشتر به الگوریتم‌های فشرده‌سازی شبیه هستند [۲۲-۱۸]. برخلاف الگوریتم‌های کاوش گراف، خروجی این الگوریتم‌ها، یک گراف خلاصه ساخته شده از گراف اولیه است که برای بررسی مناسب‌تر بوده و ذخیره آن نیز ساده‌تر است. همچنین در گزارش دیگری [۱۹] گراف خلاصه‌ای تولید مشده که برای پاسخ به پرس وجوهای گوناگون با توجه به گراف اولیه مناسب است. اما اشکال عمده این الگوریتم ها این است که از صفات موجود در رئوس و یال های گراف که حاوی اطلاعات مهم و قابل توجهی هستند، صرف‌نظر می‌کنند. بنابراین نمی‌توانند برای کاربردهای مختلف، خلاصه‌های متنوعی تولید کنند.

اما خلاصه‌سازی معناگرا به‌معنای تولید خلاصه‌های مختلف با توجه به معانی و اطلاعات مورد نیاز کاربر است، به‌طوری‌که هر خلاصه علاوه‌بر خصوصیات اصلی و برجسته گراف اولیه، اطلاعات مورد نظر کاربر را نیز در خود داشته باشد. بر این اساس، یک الگوریتم خلاصه‌سازی معناگرا، قادر است از یک گراف، خلاصه‌های مختلفی تولید کند. الگوریتم های ارائه شده توسط مرجع‌های [۲۵-۲۳] از جمله معدود الگوریتم‌هایی هستند که خلاصه‌سازی معناگرا را در پیش گرفته‌اند. در واقع حوزه خلاصه‌سازی معناگرا یک حوزه بسیار جوان محسوب می‌شود که هنوز کارهای متنوعی در آن انجام نگرفته است. لازم به‌ذکر است که الگوریتم kSNAP [۲۵] بسیار شبیه به الگوریتم SIGS است، با این تفاوت که الگوریتم SIGS روش متفاوتی را برای تجزیه ابر رأس‌ها در پیش گرفته که منجر به تولید خلاصه‌هایی با کیفیت بالاتری می‌شود. همچنین چارچوب استفاده شده برای پیاده‌سازی SIGS آن را به لحاظ کارایی و مقیاس‌پذیری از kSNAP برتر می‌سازد. لازم به‌ذکر است که الگوریتم ارائه شده در مرجع [۲۴] مکمل و در ادامه کار مرجع [۲۵] است که تنها بررسی اولیه صفات رئوس را برای کاربر ساده‌تر ساخته است و به‌صورت خودکار انجام می‌دهد و درعمل در معیارهای استفاده شده برای خلاصه تولید شده، اثری ندارد. بنابراین برتری‌های SIGS نسبت به kSNAP در کیفیت خلاصه‌های تولید شده درمورد محصول مرجع [۲۴] نیز صادق خواهد بود.

الگوریتم‌های نمایش گراف را نیز می‌توان نوعی از الگوریتم‌های خلاصه‌سازی برشمرد. مروری اجمالی بر انواع این الگوریتم‌ها در مرجع [۲۶] آمده است. در واقع برخی از این الگوریتم‌ها برای نمایش گراف‌ها، خلاصه‌هایی ساختاری از آن‌ها تولید می‌کنند که کاربر قادر است در فرآیندی محاوره‌ای[1]، با زیاد و کم کردن دقت[2] نمایش گراف، میزان جزئیات قابل نمایش را تنظیم نماید.

۳. الگوریتم خلاصه‌سازی SIGS

گراف $G = (V, E)$ مجموعه‌ای از رئوس (موجودیت‌ها) و یال‌ها است که روابط میان رئوس را نشان می‌دهند. هر رأس و یال می‌تواند دارای صفات مختلفی باشد. صفاتی مانند *وزن* و *جهت* ازجمله شناخته شده‌ترین صفاتی هستند که به‌طور معمول یال‌ها اختیار می‌کنند. گراف‌های وزن‌دار و جهت‌دار گراف‌هایی هستند که یال‌های آنها به‌ترتیب دارای صفات وزن و جهت باشند. در ادامه بحث، مجموعه صفات رئوس گراف با $A = \{a_1, a_2, \dots, a_n\}$ نمایش داده می‌شود. هر رأس v از گراف G به ازای هر صفت a_i از مجموعه A یک مقدار دارد که با $a_i(v)$ نشان داده می‌شود. این مقدار می‌تواند پوچ نیز باشد.

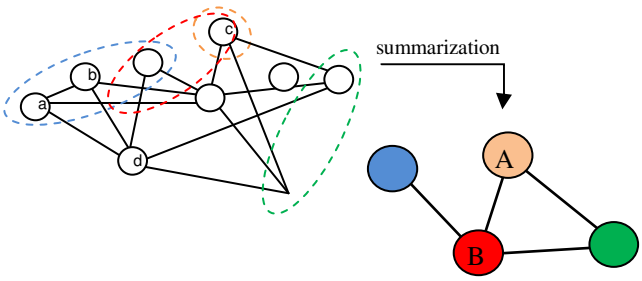
صفات رئوس و مقادیر آنها ازجمله مهم‌ترین اجزای گراف هستند که معانی مختلفی را در خود نهفته دارند. به‌عنوان مثال در گراف یک شبکه اجتماعی، هر رأس نشان‌دهنده یک عضو است که می‌تواند صفاتی مانند *سن*، *محل زندگی*، *سطح تحصیلات* و *شغل* داشته باشد. حال در چنین گرافی، خلاصه‌های مختلف می‌توانند برای کاربردهای مختلف مفید باشند.

برای مثال، می‌توان برای تحلیل‌های اجتماعی خلاصه‌ای براساس رابطه اعضاء با توجه به *سن* و *سطح تحصیلات* آنها تهیه کرد. همچنین خلاصه‌ای که براساس رابطه اعضاء با توجه به *محل زندگی* آنها باشد، می‌تواند در تحلیل‌های آماری مفید واقع شود. الگوریتم‌های خلاصه‌سازی معناگرا نیز با هدف ایجاد توانایی تولید خلاصه‌های مختلف با توجه به موضوع و اطلاعات مورد نظر کاربر طراحی شده‌اند.

[1] Interactive
[2] Resolution

الگوریتم SIGS نیز از همین نوع است. همچنین الگوریتم SIGS بر انواع گراف‌های همبند و غیر همبند، جهت‌دار و بدون جهت و وزن‌دار یا غیر وزن‌دار قابل اعمال است. اما در مقاله حاضر و به جهت سادگی، گراف‌ها بدون جهت و غیر وزن‌دار فرض شده‌اند.

گراف $G_S = (V_S, E_S)$ یک گراف خلاصه از گراف $G = (V, E)$ است اگر هر رأس از V_S یک اَبَر رأس متشکل از چند رأس عضو V و هر یال از E_S نیز یک اَبَر یال[1] متشکل از چند یال عضو E باشد. شکل (۱) نمونه‌ای از گراف اولیه و یک گراف خلاصه متناظر با آن را نشان می دهد. هر رأس از گراف خلاصه یک ابر رأس متشکل از رئوسی است که با خط چین همرنگ با ابر رأس مربوطه مشخص شده‌اند.



شکل ۱. تولید گراف خلاصه (سمت رأست) از گراف اولیه (سمت چپ)

۳-۱. خلاصه‌سازی به‌وسیله گروه‌بندی رئوس

در این روش، رئوس گراف ورودی براساس میزان مشابهت در گروه‌هایی قرار داده می‌شوند که هر گروه یک رأس از گراف خلاصه نهایی را تشکیل می‌دهد. سپس از روی ساختار یال‌های گراف اولیه، یال‌های میان گروه‌ها محاسبه می‌شود. این یال‌ها در واقع روابط میان رئوس در گراف خلاصه هستند. بر این اساس، در گراف خلاصه حاصل از چنین روشی، هر رأس یک ابر رأس از گراف اولیه و هر یال نیز یک ابر یال است. الگوریتم SIGS نیز برهمین اساس عمل می‌کند. در الگوریتم SIGS، مشابهت رئوس با توجه به مقادیر برخی صفات آنها و همسایگان مشترکشان محاسبه می‌شود. برای شرح دقیق عملکرد این الگوریتم لازم است ابتدا چند تعریف به‌طور دقیق‌تر ارائه شود.

به هر بخش‌بندی[2] Φ روی گراف G یک گروه‌بندی رئوس گفته می‌شود که در ادامه این مقاله به‌جهت سادگی و اختصار، *گروه‌بندی* نامیده خواهد شد. به هر بخش در Φ یک *گروه* گفته می‌شود و اندازه گروه‌بندی Φ برابر با تعداد گروه‌های آن است. گروه‌بندی Φ را مسلط بر گروه‌بندی Φ' می‌نامیم اگر و تنها اگر هر گروه در Φ' زیرمجموعه گروهی در Φ باشد. همچنین گروه‌بندی Φ، یک گروه‌بندی سازگار با مجموعه صفات A است و با Φ^A نمایش داده می‌شود، اگر مقادیر صفات موجود در A در تمام رئوس عضو هر گروه مساوی باشد. به عبارت دیگر Φ^A یک گروه‌بندی سازگار با صفات A است اگر:

$$\forall u, v \in V, if\ \Phi(u) = \Phi(v), then\ \forall a_i \in A, a_i(u) = a_j(v)$$

واضح است که به ازای هر زیرمجموعه از مجموعه صفات رئوس یک گراف، می‌توان گروه‌بندی‌های سازگار متعددی تولید کرد. اما قابل اثبات است که به ازای هر زیرمجموعه از مجموعه صفات رئوس یک گراف، یک گروه‌بندی سازگار وجود دارد که مسلط بر سایر گروه‌بندی‌ها است [۲۵]. این گروه‌بندی را گروه‌بندی سازگار با صفات بیشینه نامیده و با Φ^A_{max} نشان می‌دهیم.

در گروه‌بندی Φ، میان دو گروه G_i و G_j رابطه E_Φ وجود دارد اگر و تنها اگر میان برخی از رئوس عضو G_i و G_j در گراف اولیه رابطه E وجود داشته باشد. در این حالت، وزن رابطه E_Φ، درصد رأس‌هایی از دو گروه G_i و G_j است که با یکدیگر رابطه E دارند. به عنوان مثال در یک گروه‌بندی Φ روی گراف بدون جهت G، وزن رابطه میان دو گروه G_i و G_j از رابطه زیر قابل محاسبه است:

$$w^E_{i,j} = \frac{|u \in G_i, \exists (u,v) \in E\ that\ v \in G_j| + |u \in G_j, \exists (u,v) \in E\ that\ v \in G_i|}{|G_i| + |G_j|}$$

در یک گروه‌بندی Φ^A، مقادیر صفات مجموعه A در تمامی رئوس هر گروه یکسان است اما ممکن است روابط آنها یکسان نباشد. در واقع به‌طور معمول مجموعه همسایه‌های رئوس هر گروه برابر نیستند. این موضوع به‌طور عموم منجر به این می‌شود که روابط میان گروه‌ها، وزنی غیر از صفر یا صد داشته باشند.

در چنین حالتی گروه‌بندی Φ^A سازگار در روابط نیست. به‌عبارت دیگر رئوس هر گروه، گروه‌های همسایه یکسانی ندارند. برهمین اساس گروه‌بندی $\Phi^{A,E}$ را تعریف می‌کنیم که یک گروه‌بندی سازگار در صفات A و رابطه E روی گراف $G = (V, E)$ است. در چنین گروه‌بندی، رأس‌های هر گروه، علاوه‌بر اینکه در مقادیر صفات مجموعه A با یکدیگر معادل هستند، روابط یکسانی نیز با سایر گروه‌ها دارند. یعنی مجموعه همسایگان تمامی رئوس در هر گروه برابر است.

البته لازم به‌ذکر است که مجموعه همسایگان هر رأس در سطح گروه‌ها محاسبه می‌شود. به‌عبارت دیگر رئوس هر گروه با رأس‌هایی از گروه‌های مشابه همسایگی دارند. در واقع در یک گروه‌بندی سازگار با صفات و رابطه، به ازای هر دو گروه G_i و G_j، اگر برخی از رئوس گروه G_i با برخی از رئوس گروه G_j همسایه باشند، آنگاه سایر رئوس G_i نیز باید با حداقل یک رأس از G_j همسایه باشند. به‌عبارت ساده‌تر می‌توان گفت وزن تمام روابط در یک گروه‌بندی $\Phi^{A,E}$ برابر با صفر یا صد است. واضح است که به ازای گراف G و مجموعه صفات A می‌توان گروه‌بندی‌های سازگار در صفات و رابطه متفاوتی ارائه کرد. اما می‌توان اثبات کرد که به ازای هر گراف G و مجموعه صفات A، یک گروه‌بندی سازگار در صفات و رابطه وجود دارد که مسلط بر تمام گروه‌بندی‌های سازگار در صفات و رابطه است [۲۵]. چنین گروه‌بندی را گروه‌بندی سازگار در صفات و رابطه بیشینه نامیده و با $\Phi^{A,E}_{max}$ نمایش می‌دهیم.

با توجه به تعاریف فوق، عملگر خلاصه‌سازی به‌وسیله گروه‌بندی رئوس به‌صورت زیر قابل تعریف است:

[1] Super Edge
[2] Partitioning

«عملگر خلاصه‌سازی به‌وسیله گروه‌بندی رئوس، یک گراف و زیرمجموعه‌ای از صفات رئوس آن را به‌عنوان ورودی دریافت و $\Phi_{max}^{A,E}$ معادل ورودی را به‌عنوان گراف خلاصه محاسبه می‌کند.»
گروه‌های موجود در $\Phi_{max}^{A,E}$ در واقع رأس‌های گراف خلاصه و روابط موجود در آن، همان یال‌های گراف خلاصه هستند.
در ادامه این نوشته، این عملگر به‌جهت اختصار "خ.ب.گ" نامیده خواهد شد. واضح است که خلاصه محاسبه شده توسط عملگر "خ.ب.گ" به‌طور کامل به صفات انتخابی توسط کاربر به عنوان ورودی وابسته است. بنابراین کاربر می تواند با انتخاب مجموعه صفات مناسب، خلاصه های مورد نظر خود را تولید کند. این روش به‌صورت تقریبی همان روش خلاصه سازی است که در بخش‌های قبلی با عنوان خلاصه‌سازی معناگرا به آن اشاره شد. اما خلاصه تولید شده به‌وسیله عملگر *"خ.ب.گ"*، ایرادات و کاستی‌هایی دارد که می‌بایست برطرف شوند تا بتوان خلاصه تولید شده را به‌عنوان یک خلاصه مطلوب ارزیابی کرد. در بخش بعد به این کاستی ها و راه حل رفع آنها تا رسیدن به الگوریتم مطلوب اشاره خواهد شد.

۳-۲. گروه‌بندی منعطف و تعیین وضوح گراف خلاصه

یکی از مشکلاتی که عملگر *"خ.ب.گ"* دارد این است که به‌طور معمول هنگامی که به گراف‌های عظیم و پیچیده اعمال می‌شود، خلاصه تولید شده توسط آن بیش از اندازه بزرگ است. این موضوع ناشی از پایبندی سفت و سخت این عملگر به سازگاری خلاصه نهایی در صفات و رابطه است.
در واقع همان‌طور که در بخش قبل گفته شد، عملگر "خ.ب.گ" خلاصه‌ای تولید می‌کند که همه رئوس موجود در هر گروه از لحاظ مقادیر صفات و همسایگان با یکدیگر برابر باشند. از طرفی در گراف‌های بزرگ، به‌طور معمول تفاوت میان رئوس بسیار زیاد است. بنابراین رئوس کمی را می‌توان یافت که علاوه‌بر شباهت در مقادیر صفات انتخاب شده، همسایگان یکسانی نیز داشته باشند.
علاوه‌بر تنوع ذاتی این‌گونه گراف‌ها، وجود اختلالات و خطاهای کوچک در زمان مدل‌سازی آنها نیز خود در عدم یکسان بودن و شباهت کامل میان رئوس این گراف‌ها مؤثر است. به‌همین دلیل، پس از اعمال عملگر "خ.ب.گ" بر چنین گراف‌هایی، خلاصه تولید شده نیز خود تعداد زیادی رأس و رابطه خواهد داشت و حتی بسیاری از رئوس گراف اولیه به‌دلیل عدم سازگاری کامل در صفات و رابطه با سایر رئوس، در گروه‌های تک رأسی قرار خواهند گرفت. به‌همین دلیل لازم است برای افزایش میزان خلاصه‌سازی و همچنین کیفیت گراف خلاصه، مدل گروه‌بندی را منعطف‌تر طراحی کنیم.
با توجه به مطالب فوق، می‌توان با حذف شرط سازگاری در روابط، بهترین گروه‌بندی روی گراف ورودی برحسب مجموعه صفات A را یک گروه‌بندی سازگار در صفات دانست، به‌طوری که کمترین فاصله را تا $\Phi_{max}^{A,E}$ داشته باشد. برهمین اساس فاصله گروه‌بندی Φ^A با $\Phi_{max}^{A,E}$ با استفاده از فرمول زیر قابل محاسبه است:

$$\alpha\left(\Phi^{A}\right)=\frac{\sum_{G_i,G_j\in\Phi^A}\left(\delta_{E,G_j}(G_i)+\delta_{E,G_i}(G_j)\right)}{|\{(G_i,G_j)|p_i(j)\neq 100\}|} \quad (۱-۳)$$

$$\delta_{E,G_j}(G_i)=\begin{cases}p_j(i) & p_j(i)<50\\ 100-p_j(i) & otherwise\end{cases} \quad (۲-۳)$$

در فرمول (۳-۲)، $p_j(i)$ درصد مشارکت رئوس گروه G_j در رابطه گروهی (G_i, G_j) را نشان می‌دهد. واضح است که در یک گروه‌بندی $\Phi_{max}^{A,E}$، این مقدار همواره برابر صفر یا صد است. بنابراین فرمول (۳-۱) در واقع مجموع درصد ناسازگاری روابط در گروه‌بندی Φ^A را که برحسب تعداد روابط ناسازگار نرمال شده نشان می‌دهد. به‌عبارت دیگر اگر درصد مشارکت رأس‌های گروه در یک رابطه کمتر از ۵۰ باشد، اندازه آن به‌عنوان اقلیت مخالف و اگر درصد مشارکت بیشتر از ۵۰ باشد، اختلاف آن با ۱۰۰ را به‌عنوان اقلیت درنظر می‌گیرد و در محاسبه فاصله استفاده می‌کند. با توجه به پارامتر α، گروه‌بندی به $\Phi_{max}^{A,E}$ نزدیک‌تر است که α آن کمتر باشد. بنابراین در فرآیند خلاصه‌سازی نیز باید همواره به‌دنبال گروه‌بندی رفت که کمترین فاصله ممکن تا مدل ایده‌آل $\Phi_{max}^{A,E}$ را داشته باشد. این ایده در واقع یکی از اصول استفاده شده در SIGS است. از طرفی واضح است که با به‌کارگیری پارامتر درصد مشارکت رئوس گروه‌ها، به شباهت‌ها و تفاوت‌های رفتاری رئوس یک گروه اهمیت خاصی داده شده است.
شبه کد مربوط به الگوریتم SIGS در جدول (۱) نشان داده شده است.

جدول ۱. شبه کد الگوریتم SIGS

```
INPUT:
  Graph G(V,E); Attribute set A; Summary size k;
OUTPUT:
  A summary graph S(V',E');
SIGS ALGORITHM:
// computing maximum A-compatible grouping
1. Sort and Groups nodes
      based on values of attributes in A
2. for each group Gi
3.     create a list of associated nodes
4.     compute adjLst of each node in Gi
          with other groups
5.     create and compute an adjLst of Gi
          with other groups
6.     let Φc as current grouping
// spliting groups to reach to higher resolution
7. else if |Φc| < k
8.     compute splt_fact for each group in Φc
9.     build a heap on the splt_fact value
          of each group
10.    while |Φc| < k
11.        get groups with splt_fact
              in MAX_VARIANCE_INTERVAL
12.        select Gi with max alien_nodes
13.        split Gi into two Based on neighbor
              Gj = arg(alien_nodes)
14.        update the heap and adjLst of each group
//creating summary graph and returning
15.form the summary graph S(V',E') using Φc
16.return S(V',E')
```

الگوریتم SIGS گراف اولیه را به‌همراه مجموعه‌ای از صفات رئوس آن (A) و اندازه مورد نظر کاربر برای خلاصه نهایی (k) به‌عنوان ورودی دریافت کرده و گراف خلاصه‌ای به اندازه مورد نظر کاربر تولید می‌کند. در واقع خروجی الگوریتم SIGS یک گروه‌بندی سازگار با

صفات A است که براساس حداقل فاصله با گروه‌بندی ایده‌آل $\Phi_{max}^{A,E}$ با توجه به پارامتر α تولید شده است.

الگوریتم SIGS از دو قسمت اصلی تشکیل شده است. در قسمت اول (خطوط ۶-۱) گروه‌بندی اولیه Φ^A براساس مجموعه صفات ورودی به‌دست می‌آید. سپس در صورتی که Φ^A از اندازه مورد نظر کاربر کوچک‌تر باشد، نیاز است که بعضی از گروه‌ها به گروه‌های کوچک‌تری شکسته شوند. در چنین حالتی باید گروهی برای تجزیه انتخاب شود که بیشترین تأثیر را در پارامتر δ دارد. همچنین، گروه مورد نظر باید به‌گونه‌ای تجزیه شود که بیشترین مقدار کاهش در α را ایجاد کند تا گروه‌بندی حاصل به Φ_{max}^{A} نزدیک‌تر شود. از طرفی با توجه به فرمول (۳-۱) می‌دانیم که روابطی که درصد مشارکت یک یا هر دو رأس آنها نزدیک به ۵۰ باشد، بیشترین تأثیر را در α ایجاد می‌کنند. بنابراین ابتدا باید گروه‌هایی را که رابطه‌ای با درصد مشارکت نزدیک به ۵۰ دارند، استخراج و سپس از میان آنها بهترین گزینه را برای تجزیه انتخاب کنیم. بعد از آن نیز باید گروه مورد نظر را براساس رابطه ناسازگارش تجزیه کنیم تا علاوه‌بر نزدیک‌تر شدن به اندازه مطلوب برای خلاصه، مقدار α نیز کاهش یابد. به این منظور پارامتر splt_fact به‌صورت زیر تعریف می‌شود:

$$splt\_fact(G_i) = max_j\{(|50 - p_i(j)|)^{-1}\} \qquad (۳-۳)$$

در حقیقت splt_fact به ازای هر گروه، عکس درصد ناسازگارترین رابطه آن را نشان می‌دهد. در قسمت دوم الگوریتم SIGS (خطوط۱۴-۷)، ابتدا این مقدار برای تمام گروه‌ها محاسبه می‌شود. سپس (خط ۱۱) تمامی گروه‌هایی که رابطه‌ای با درصد مشارکت نزدیک ۵۰ دارند، انتخاب می‌شوند. برای این انتخاب یک بازه با نام MAX_VARIANCE_INTERVAL تعریف شده است که در ابتدا می‌تواند درصدهای بین ۳۵ تا ۷۵ را شامل شود. سپس در صورت عدم وجود گروهی با چنین رابطه‌ای، این بازه در هر مرحله ۵٪ از هر طرف گسترده خواهد شد. بعد از انتخاب گروه‌های اولیه، از میان آنها گروهی که بیشترین تعداد رأس را در رابطه ناسازگارش دارد، برای تجزیه انتخاب می‌شود (خط ۱۲). این کار به‌وسیله پارامتر alien_nodes که به‌صورت زیر تعریف شده انجام می‌گیرد:

$$alien\_nodes(G_i) = p_i(\arg[splt\_fact(G_i)]) * |G_i| \qquad (۴-۳)$$

در واقع alien_nodes تعداد رئوس مشارکت کننده در رابطه ناسازگار را برای گروه G_i نشان می‌دهد. پس از انتخاب بهترین گروه برای تجزیه، لازم است که گروه مذکور براساس رابطه ناسازگارش به دو گروه تجزیه شود (خطوط ۱۴-۱۳). در هر مرحله از قسمت دوم، هر گروه فقط براساس یک رابطه که بیشترین ناسازگاری را دارد و فقط به دو گروه جدا شکسته می‌شود. علت اصلی این امر، اطمینان از رعایت اصل «مشابه‌ها باید همواره باهم باشند» یا KEAP[1] است.

پس از رسیدن به گروه‌بندی مطلوب، حال الگوریتم SIGS گراف خلاصه را براساس این گروه‌بندی ساخته و به‌عنوان خلاصه نهایی به کاربر برمی‌گرداند (خطوط ۱۶-۱۵).

[1] Keap Equivalen Always Together

۳-۳.پیچیدگی زمانی الگوریتم SIGS

یکی از مشکلات عمده الگوریتم‌های پردازش گراف، پیچیدگی زمانی و محاسباتی بالای این الگوریتم‌ها است که آنها را در مواجهه با گراف‌های عظیم بسیار ناکارآمد و گاه حتی غیرقابل استفاده می‌کند. برهمین اساس، یکی از پارامترهای مهم در طراحی الگوریتم SIGS، پیچیدگی زمانی این الگوریتم بوده است. بنابراین در بخش حاضر، به محاسبه و بررسی پیچیدگی زمانی الگوریتم SIGS پرداخته شده است. در این محاسبات فرض بر آن است که تعداد صفات انتخابی توسط کاربر، نسبت به اندازه و درجه گراف اولیه و همچنین اندازه گراف خلاصه نهایی بسیار کوچک است.

با توجه به جدول (۱)، الگوریتم SIGS از دو قسمت مجزا تشکیل شده است. قسمت اول (خطوط ۶-۱) برای محاسبه گروه‌بندی سازگار با صفات A است که در آن، زمان لازم برای گروه‌بندی اولیه رئوس (خط ۱) $O(|V|log|V|)$ و زمان اجرای حلقه for معادل $O(|E|+|V|)$ است. در نتیجه زمان اجرای این قسمت، در کل برابر با $O(|V|log|V| + |E|)$ است. در قسمت دوم، با توجه به اختلاف اندازه گروه‌بندی سازگار با صفات و اندازه مورد نظر کاربر، تا رسیدن به اندازه مطلوب، گروه‌ها شکسته می‌شوند.

در این بخش محاسبه splt_fact و ساختن heap مربوط به آن به $O(k_0^2 + logk_0)$ زمان نیاز دارد که در آن k_0 اندازه گروه‌بندی اولیه سازگار با صفات است. علاوه‌بر آن، هزینه هر بار اجرای حلقه while نیز در بدترین حالت برابر با $O(|E| + k_i + logk_i)$ است که در آن k_i اندازه گروه‌بندی در iامین اجرای حلقه است. واضح است که k_i همواره مرز پایین k و با k قابل پوشش است. بنابراین هزینه زمانی کل حلقه مذکور در بدترین حالت $O(k^2 + klogk + k|E|)$ خواهد بود. اگر فرض کنیم که k_0 نیز حد زیرین k باشد، آنگاه هزینه اجرای کل این بخش (خطوط ۱۴-۷) نیز معادل همین میزان است.

در بخش آخر الگوریتم نیز (خطوط ۱۶-۱۵) که گراف خلاصه تولید می‌شود، هزینه زمانی برابر با $O(k^2)$ است. بر این اساس هزینه اجرای الگوریتم SIGS به‌صورت زیر خواهد بود:

$$O(|V|log|V| + k|E| + k^2 + klogk)$$

با این محاسبات روشن می‌شود که هزینه زمانی اجرای الگوریتم SIGS نسبت به اندازه گراف (|V|) و درجه آن (|E|) یک چند جمله‌ای است.

۴-۳. مقیاس‌پذیری و کارایی الگوریتم

علاوه‌بر زمان اجرا، کارایی و مقیاس‌پذیری الگوریتم‌هایی که با گراف‌های بزرگ و عظیم‌الجثه سروکار دارند نیز از معیارهای مهم ارزیابی و موفقیت این الگوریتم‌ها محسوب می‌شود. غالب الگوریتم‌هایی که تاکنون ارائه شده‌اند نیز قادر به پاسخ‌گویی در حد مطلوبی در این زمینه نیستند. حتی نمونه ارائه شده در مرجع [۲۵] که شباهت زیادی به الگوریتم SIGS دارد برای خلاصه کردن یک گراف با یک میلیون رأس و تولید خلاصه‌ای با اندازه هزار، آن هم در حالتی که تمام ساختار داده خلاصه‌سازی و همچنین گراف خلاصه

در حافظه اصلی باشد، زمانی معادل ۸۰۰۰ ثانیه نیاز دارد. این درحالی است که در نمونه‌های واقعی مانند گراف وب یا گراف شبکه‌های اجتماعی و یا حتی گراف ترکیبات شیمیایی و زیستی، اندازه گراف‌ها گاه به چند صد میلیون و حتی میلیارد رأس نیز می‌رسد. واضح است که در این حالت، علاوه‌بر اینکه وجود تمام ساختار داده خلاصه‌سازی و گراف خلاصه در حافظه امکان‌پذیر نیست، زمان اجرای الگوریتم نیز بسیار زیاد خواهد بود.

در طراحی سطح بالای الگوریتم SIGS نیز که در جدول (۱) ارائه شد، برخی نقاط بحرانی از منظر کارایی و مقیاس‌پذیری به‌چشم می‌خورند. از جمله این نقاط قسمت اول الگوریتم است. جایی‌که در آن لازم است تمام رئوس گراف ورودی براساس مقادیر صفات انتخابی مرتب شده و در گروه‌های مربوطه قرار گیرند. نقطه دیگری که مسئله مقیاس‌پذیری در آن به‌شدت اهمیت پیدا می‌کند، ساختار داده‌ای است که برای تولید و نگهداری داده‌های مرتبط با خلاصه‌سازی و گراف خلاصه (مانند گروه‌ها و اطلاعات هرکدام) مورد استفاده قرار می‌گیرد.

همچنین مدل گراف و سازوکار استفاده شده برای ذخیره و بازیابی گراف ورودی نیز خود از نکات و نقاط چالش برانگیز مقیاس‌پذیری و کارایی این الگوریتم محسوب می‌شود. از طرفی چالش‌های مربوط به کارایی و مقیاس‌پذیری این قسمت‌ها بیشتر در قالب روش پیاده‌سازی و چارچوب و بستر مورد استفاده برای تولید و اجرای الگوریتم قابل حل هستند. همچنین خود الگوریتم با توجه به محاسبات بسیار ساده و پیچیدگی زمانی مطلوب، هیچ چالش ذاتی در زمینه‌های مذکور در سطح طراحی نداشته و به‌نظر می‌رسد چالش‌ها و نگرانی‌های عنوان شده نیز با انتخاب چارچوب و بستر مناسب و پیاده‌سازی درست الگوریتم قابل حل باشند. به‌همین جهت در بخش‌های آینده که مرتبط با روش پیاده‌سازی و آزمون الگوریتم SIGS خواهد بود، به این موضوع بیشتر پرداخته خواهد شد.

۴. پیاده‌سازی الگوریتم

با توجه به اهمیت کیفیت پیاده‌سازی الگوریتم SIGS در پارامترهای کارایی و مقیاس‌پذیری، در بخش حاضر به معرفی بستر و چارچوبی مناسب برای پیاده‌سازی الگوریتم SIGS پرداخته خواهد شد. پیکربندی تشریح شده در این بخش، در حالت کلی، یک ترکیب مناسب برای تولید الگوریتم‌هایی با خصوصیات، معیارها و شرایط مشابه SIGS است.

۴-۱. مدل گراف و پایگاه‌داده

همان‌طور که گفته شد، بزرگی اندازه گراف‌ها یکی از مهم‌ترین چالش‌های کار با آنها است. برای غلبه بر چنین چالشی، علاوه‌بر طراحی صحیح الگوریتم‌ها، می‌بایست با درنظر گرفتن موضوع مقیاس‌پذیری و کارایی، از ساختاری مناسب برای مدل کردن گراف و نیز از بستر و چارچوبی مناسب برای ذخیره و بازیابی گراف استفاده کرد. بدین منظور در این مقاله، از پایگاه‌داده‌ای با نام Neo4j برای پیاده‌سازی SIGS استفاده شده که در واقع یکی از انواع پایگاه‌داده‌های NoSQL بوده و مخصوص ذخیره و بازیابی داده‌های گرافی است [۲۷].

توانایی بالای این پایگاه‌داده‌ها در ذخیره و بازیابی داده‌های حجیم که الگوی[1] متفاوتی نیز دارند، آنها را به‌گونه‌ای محبوب از پایگاه‌داده‌ها بدل کرده است که امروزه در بسیاری از سیستم‌های بازیابی اطلاعات، مانند موتورهای جستجو کاربرد فراوان پیدا کرده‌اند. به‌طور خلاصه می‌توان گفت، برای کار با داده‌های حجیمی که الگوی آنها یک ماتریس تنک[2] است، چنین پایگاه‌داده‌هایی بسیار مناسب‌تر از انواع رابطه‌ای هستند.

به‌طور کلی می‌توان پایگاه‌های داده‌ای غیر رابطه‌ای را به چهار نوع تقسیم کرد که عبارت است از: *ستون محور*، *کلید/مقدار*، *گرافی* و *سند محور*. در انواع گرافی (که Neo4j یکی از بهترین آنها است)، دو نوع مجودیت رأس و یال قابل تعریف می‌باشند که هرکدام یک شناسه و تعدادی صفت از انواع مختلف دارند. همچنین رئوس می‌توانند به‌وسیله یال‌ها به‌یکدیگر متصل شوند. در واقع هر رأس تعدادی یال و هر یال دو رأس دارد. در پایگاه‌های داده‌ای غیر رابطه‌ای، جستجوها فقط براساس شناسه موجودیت‌ها انجام می‌گیرد. به‌عبارت دیگر، از آنجا که به ازای صفات، شاخصی تولید نمی‌شود، انجام پرس‌وجو براساس مقادیر صفات بسیار پرهزینه است.

یکی از خصوصیات پایگاه‌داده Neo4j، استفاده از شاخص‌گذار لوسن[3] برای تولید شاخص روی مقادیر صفات رئوس و یال‌ها است. البته استفاده از لوسن به‌عنوان یک ابزار جانبی به‌وسیله پایگاه‌داده Neo4j تنها برای مواقع ضروری است، چراکه علاوه بر افزایش هزینه حافظه موردنیاز، محلی بودن لوسن نیز درحالتی که Neo4j به‌صورت توزیع‌شده اجرا می‌شود مشکلات کارایی را به‌دنبال خواهد داشت. به‌همین دلیل بهتر است از این ابزار تنها در مواقع ضروری استفاده شود.

با توجه به مطالب فوق، ضرورت استفاده از پایگاه‌داده‌ای نظیر Neo4j برای پیاده‌سازی الگوریتم‌های پردازش گراف مشخص شده است. این مورد به‌ویژه در مورد الگوریتم‌هایی مانند SIGS که به‌طور ذاتی با گراف‌های عظیم سروکار دارند، به‌طریق اولی احساس می‌شود.

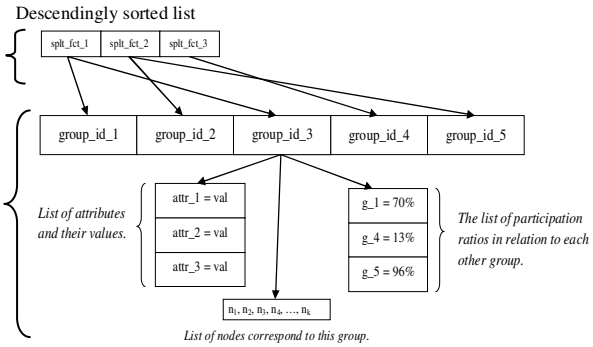
۴-۲. معماری و ساختار داده‌ای استفاده شده

یکی از ضعف‌های Neo4j عدم پشتیبانی از یال‌های برگشتی است. یک یال برگشتی، یالی است که هر دو سر آن روی یک رأس باشد. این مشکل در الگوریتم SIGS به‌وسیله تعریف یک رأس با نام loop حل شده است. در واقع به ازای هر یال برگشتی روی رأس v، یک رأس loop ایجاد شده و رأس v به‌وسیله یالی از نوع loop-edge به آن متصل می‌شود. سایر خصوصیات گراف‌های استفاده شده در

[1] Schema
[2] Sparse Matrix
[3] Lucene

SIGS، یعنی گراف ورودی و گراف خلاصه، تابع مدل Neo4j بوده و برای نگهداری و پردازش آنها از Neo4j در کد SIGS استفاده شده است. در واقع هرجا که در الگوریتم نیاز به تولید و کار با گراف باشد، از این پایگاه‌داده استفاده خواهد شد.

علاوه‌بر گراف ورودی و گراف خلاصه نهایی، لازم است در حین اجرای الگوریتم و در مراحل مختلف، اطلاعات مورد نیاز الگوریتم نیز در یک ساختار داده مناسب نگهداری و مدیریت شود تا درنهایت، گراف خلاصه به‌وسیله آن تولید شود. الگوی ساده و سطح‌بالای این ساختار داده در شکل (۲) نشان داده شده است.



شکل ۲. شمای سطح بالای ساختار داده‌ای استفاده شده برای خلاصه‌سازی

همان‌طورکه مشاهده می‌شود، این ساختار از دو بخش اصلی تشکیل شده است. بخش اول یک لیست مرتب شده از مقادیر مختلف splt_fact است که هرکدام به لیستی از ابر رأس‌هایی اشاره می‌کنند که دارای splt_fact هستند. به‌وسیله این ساختار می‌توان به‌سادگی ابر رأس‌هایی را که بیشترین splt_fact را دارند، شناسایی کرد. بخش دوم این ساختار داده برای نگهداری اطلاعات مربوط به ساختار گراف خلاصه حین فرآیند خلاصه‌سازی است. این بخش در واقع لیستی از ابر رأس‌ها است که هر یک به‌وسیله یک شناسه یکتا مشخص شده‌اند. از طرفی هر ابررأس سه لیست دارد. اول، لیستی از شناسه رئوسی از گراف اولیه که متعلق به این ابررأس هستند. دوم، لیستی از صفات مربوطه و مقادیر آنها که در مرحله تولید مدل سازگار با صفات به‌دست آمده‌اند. و درنهایت هر ابررأس لیستی از گروه‌های همسایه به‌همراه درصد میزان مشارکت رئوس خود در رابطه با هر گروه همسایه را در اختیار دارد که در محاسبه پارامترهای α و splt_fact بسیار پرکاربرد است. این ساختار در بسیاری از مسائل معمولی قابل بارگذاری در حافظه اصلی سیستم‌های ساده است. اما در صورت بزرگ شدن گراف خلاصه یا کمبود حافظه نیز هم‌چنان قابل استفاده است. در واقع الگوریتم SIGS طوری پیاده‌سازی شده که می‌تواند لیست مربوط به ابر رأس‌ها را به‌وسیله سازوکار نهان‌سازی، به‌صورت مشترک در حافظه اصلی و جانبی سیستم ذخیره و مدیریت کند. به این وسیله، ساختار داده فوق نیز به‌طور کامل مقیاس‌پذیر و همچنین کارا برای فرآیند خلاصه‌سازی الگوریتم SIGS خواهد بود. لازم به‌ذکر است که در طول فرآیند خلاصه‌سازی، لازم است به هر رأس از گراف اولیه صفتی با عنوان «شناسه گروه» اضافه شود که حاوی شناسه ابررأس دربردارنده آن رأس در گراف خلاصه باشد. این صفت می‌بایست در هر مرحله از الگوریتم و پس از هر بار تجزیه گروه‌ها بروزرسانی شود. همچنین محتوای لیست‌های مربوط به رئوس متعلق به ابررأس و همسایگی‌ها و درصد مشارکت هر گروه نیز در هر مرحله از الگوریتم از روی گراف اولیه بروزرسانی خواهد شد.

۵. آزمون‌ها، تحلیل نتایج و ارزیابی الگوریتم

تمامی آزمون‌های انجام‌گرفته در این بخش به‌وسیله یک سیستم با مشخصات سخت‌افزاری نشان داده‌شده در جدول (۲) انجام گرفته است. ضمن آنکه یادآوری می‌شود الگوریتم SIGS به زبان جاوا نوشته شده و برای ذخیره و بازیابی گراف‌ها از پایگاه‌داده Neo4j درحالت جاسازی شده استفاده شده است.

جدول ۲. سخت‌افزارهای استفاده شده جهت اجرا و محک الگوریتم SIGS

CPU	*Core2 Quad 2.66GHz*
RAM	*4 GB*
Operating System	*Ubuntu Desktop 10.0.4*
File System	*ext 4*

۵-۱. داده‌های استفاده شده

برای انجام مجموعه آزمون‌هایی جامع روی SIGS و بررسی آن از جهات مختلف، درمجموع چهار مجموعه داده گرافی استفاده شده است که به قرار زیر می‌باشند:

- گراف سازگار با صفات و روابط: برای بررسی اولیه الگوریتم SIGS و تضمین صحت عملکرد آن، گرافی با اندازه ۲۰۰۰۰ رأس تولید شد. هر رأس از گراف مذکور صفتی با نام name دارد که در مجموع می‌تواند ۵ مقدار مختلف داشته باشد. همچنین گراف تولید شده، مدل سازگار با صفات (صفت name) و رابطه‌ای به اندازه ۶۵ دارد.
- گراف وبلاگ‌های سیاسی: این گراف از طریق اینترنت[1] تهیه شده و شامل مجموعه‌ای از ۱۴۹۰ وبلاگ سیاسی ایالات متحده و یال‌های میان آنها است (در مجموع ۱۹۰۸۷ یال). به‌عبارت دیگر می‌توان گفت این گراف یک زیرگراف کوچک از گراف وب است. هر رأس از گراف مذکور یک وبلاگ سیاسی است که صفتی با عنوان جناح دارد که می‌تواند آزادیخواه یا محافظه‌کار باشد. همچنین هر یال در این گراف نشانه وجود یک پیوند میان دو وبلاگ است. این گراف برای بررسی کیفیت خلاصه‌های تولیدشده توسط SIGS استفاده خواهد شد.
- گراف همکاری نویسندگان: این گراف با استفاده از اطلاعات موجود در مجموعه داده‌ای DBLP که به‌طور رایگان قابل دسترسی است[2]

[1] http://www-personal.umich.edu/~mejn/netdata/polblogs.zip
[2] http://kdl.cs.umass.edu/databases/dblp-data.xml.gz

ایجاد شده است. با توجه به داده‌های موجود در این گراف، گرافی با عنوان گراف همکاری نویسندگان ساخته شد. در این گراف هر رأس یک نویسنده و هر یال میان دو نویسنده بیانگر حداقل یک همکاری میان آن‌دو در نوشتن یک سند در گراف DBLP است. همچنین هر رأس دارای یک صفت عددی به‌نام تعداد مقالات است که تعداد مقالات نوشته شده توسط آن نویسنده را نشان می‌دهد. این گراف در مجموع ۴۵۶۰۲۰ رأس و ۱۲۳۲۸۹۷ یال دارد.

- مجموعه گراف‌های شبیه‌سازی شده: برای آزمون‌های کارایی و مقیاس‌پذیری از گراف‌های شبیه‌سازی شده‌ای که به‌وسیله یک نمونه تولیدکننده تصادفی گراف[1] تولید شده‌اند (که یک پیاده‌سازی از مرجع [۲۸] است)، استفاده شده است. با توجه به این موضوع که اغلب گراف‌های واقعی از الگوی توزیع توانی در درجه رئوس خود پیروی می‌کنند [۲۹]، گراف‌های تولید شده در این مجموعه نیز براساس همین قانون و با درجه میانگین ۴/۵ و در اندازه‌های ۲۰۰۰۰، ۵۰۰۰۰، ۱۰۰۰۰۰، ۲۰۰۰۰۰، ۵۰۰۰۰۰، ۸۰۰۰۰۰ و ۱۰۰۰۰۰۰ رأس تولید شدند.

۵-۲. آزمون صحت عملکرد

هدف از این آزمون بررسی دو موضوع است. اول اینکه آیا الگوریتم درست پیاده‌سازی شده است یا خیر. دوم اینکه آیا پارامترهای الگوریتم SIGS درجهت رسیدن به مدل سازگار با صفات و روابط عمل می‌کنند یا خیر. برهمین اساس، الگوریتم SIGS به‌وسیله گراف سازگار با صفات و روابط به‌عنوان ورودی و نیز صفت name و اندازه خلاصه بی‌نهایت به اجرا گذاشته شد. نتیجه حاصل از این اجرای آزمایشی در جدول (۳) نشان داده شده است. همان‌طور که مشاهده می‌شود. الگوریتم SIGS مطابق با انتظار توانست بدون طی مرحله اضافی و با انجام ۶۰ عمل تجزیه، به مدل سازگار با صفات و روابط دست‌یابد.

در جدول مذکور، ستون اول (Split number) شماره مرحله تجزیه است. ستون دوم (Split node ID) شماره رأسی را که قرار است تجزیه شود نشان می‌دهد. ستون سوم (Neighbor node ID) شماره رأس مجاوری را که تجزیه براساس آن همسایگی انجام می‌شود، شامل می‌شود و ستون آخر مقدار پارامتر آلفا را پس از هر تجزیه نشان می‌دهد.

۵-۳. آزمون کیفیت خروجی

در این آزمون، گراف وبلاگ‌های سیاسی به‌عنوان ورودی به الگوریتم SIGS داده شده تا براساس صفت جناح که یک صفت دو مقداری است، گراف‌های خلاصه‌ای با اندازه‌های ۲ (مدل سازگار با صفات) تا ۷ تولید شود. برخی از خلاصه‌های به‌دست آمده از اجرای این آزمون در شکل (۳) نشان داده شده‌اند.

[1] http://www-rp.lip6.fr/~latapy/FV

جدول ۱. خروجی حاصل از اجرای الگوریتم SIGS در آزمون صحت الگوریتم

attribute compatible grouping was constructed. the alpha parameter of current grouping is: 23.916666666666664			
breaking groups ...			
Split number	Split node ID	Neighbor node ID	Alpha (after split)
1	3	5	21.833333333333336
2	4	5	20.063492063492067
3	5	8	18.782051282051285
4	6	11	12.013888888888888
5	1	7	14.273333333333335
6	2	15	11.826815695151858
7	16	13	12.685856286974902
8	15	19	12.35337953958974
9	19	12	12.33021621044529
10	20	12	9.537588785509216
11	21	10	9.91276631012019
12	27	13	10.53169801132214
13	28	18	8.96919801132214
14	29	12	8.085506451570067
15	17	14	9.202351918818154
16	35	7	8.961567572113363
17	37	12	8.593619824517141
18	39	13	7.85546878398325
19	8	18	8.286632221848457
20	10	26	8.193239126365562
21	11	24	8.329245858212094
22	9	22	8.540551304369162
23	12	48	11.580286597604754
24	51	24	12.095087112405267
25	49	52	10.337255355563121
26	47	52	8.943553659940685
27	52	46	8.432493450801218
28	54	58	6.923351639738666
29	13	45	11.907885385272815
30	62	30	11.907885385272815
31	43	65	11.907885385272815
32	45	67	6.047560059930172
33	42	26	7.0868022131258135
34	18	71	12.832852841309501
35	73	30	8.517290848686198
36	75	44	8.983359476232692
37	70	77	6.202629492600419
38	26	72	10.567079432195714
39	81	79	6.359023640207569
40	63	72	12.637288583509509
41	85	79	6.088638321196457
42	44	80	15.391384778012684
43	89	79	5.258425568959083
44	14	34	5.355179704016912
45	7	92	4.885231048021744
46	22	48	9.20296420801982
47	97	58	4.782679403508523
48	48	24	5.630113061862303
49	50	101	6.980829528857839
50	24	102	16.824869482676792
51	105	96	14.585883312933499
52	107	46	7.890365448504984
53	46	104	15.03875968992248
54	111	96	26.253229974160202
55	103	112	27.433247200689056
56	100	114	25.98939208486332
57	106	112	13.842746400885938
58	113	108	6.046511627906979
59	96	104	50.0
60	123	112	00.0

همان‌طور که در شکل (۳) نیز مشاهده می‌شود، اولین خلاصه تولید شده در واقع همان مدل سازگار با صفات و دارای دو رأس است. این گراف در شمایی سطح بالا و کلی، تعداد و رفتار وبلاگ‌های آزادیخواه و محافظه‌کار را نشان می‌دهد. به سادگی می‌توان فهمید که روابط درون دسته‌ای هر گروه از وبلاگ‌ها بسیار قوی‌تر از روابط میان دسته‌های مختلف است.

از این مرحله به بعد، با افزایش اندازه و جزییات گراف خلاصه، گراف‌های جدیدی ایجاد می‌شوند که پیچیدگی بیشتری داشته است، ضمن آنکه اطلاعات خاصی را به کاربر ارائه نمی‌دهند. بلکه تنها با افزایش جزئیات و پیچیدگی‌ها، کار درک گراف و کسب دانش کلی را با مشکل مواجه می‌سازد. بنابراین به‌نظر می‌رسد برای کاربر عادی که قصد فهم ساختار کلی گراف و رفتار نویسندگان را دارد، گراف خلاصه با اندازه ۵، گزینه مناسبی است. از طرفی گراف‌های خلاصه بزرگ‌تر می‌توانند به‌عنوان ورودی الگوریتم‌های کاوش گراف استفاده شوند.

موتورهای جستجو و نیز سیستم‌های رصد فضای وب از جمله برنامه‌هایی هستند که برای بررسی فضای وب نیاز مبرمی به این‌گونه خلاصه‌سازی‌ها دارند. در واقع گراف وب و ارتباطات موجود میان اعضای آن، منبع اطلاعاتی بسیار مفیدی برای سیستم‌هایی که فضای امنیتی و سیاسی جوامع را رصد می‌کنند، محسوب می‌شود. خلاصه‌های تولید شده در این آزمون خود نمونه‌هایی هستند که می‌توانند به‌سادگی اطلاعات مناسبی از فضای سیاسی حاکم برسمتی از کشور مذکور را در زمان خاص به کاربران انتقال دهند. همچنین خلاصه‌های تولید شده از شبکه‌های اجتماعی و خبرگزاری‌ها نیز چنین موقعیتی دارند. به‌علاوه در سیستم‌هایی مانند موتورهای جستجو، برای رتبه‌بندی صفحات وب نیز وجود خلاصه‌های بزرگ‌تر از قسمت‌های بزرگ‌تری از گراف وب بسیار مفید و ضروری است.

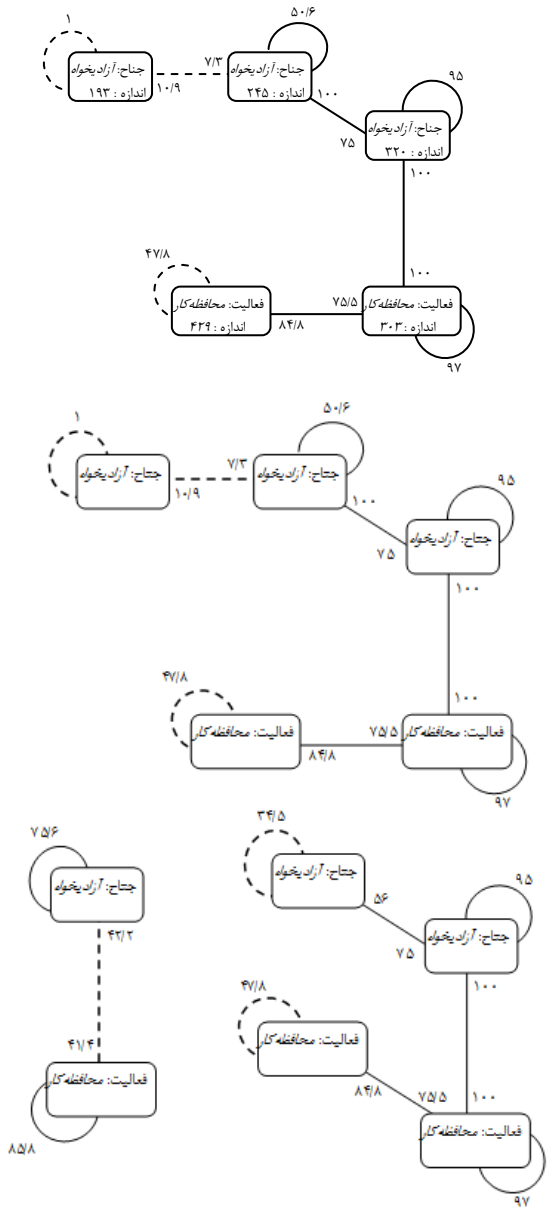
۵-۴. آزمون کیفیت خروجی (گراف همکاری نویسندگان)

هدف از این آزمون، خلاصه کردن گراف همکاری نویسندگان براساس صفت *تعداد مقالات* هر نویسنده و سپس بررسی گراف خلاصه تولیدشده است. ازطرفی بررسی اولیه نشان می‌دهد که صفت مورد نظر در مجموع ۲۴۳ مقدار مختلف دارد.

در نتیجه مدل اولیه سازگار با صفات که در فاز اول الگوریتم خلاصه‌سازی تولید خواهد شد، دارای ۲۴۳ رأس است که برای یک گراف خلاصه اندازه بزرگی است. بنابراین و به‌منظور کوچک کردن دامنه مقادیر صفت ورودی، صفت جدیدی با نام فعالیت به رئوس گراف افزوده شد.

مقدار این صفت با توجه به صفت تعداد مقالات تعیین شد. براین اساس، اگر تعداد مقالات نویسنده‌ای کمتر از ۵ باشد، فعالیت او "کم"، اگر تعداد مقالات بین ۵ و ۱۵ باشد "متوسط" و اگر بیش از ۱۵ مقاله باشد فعالیت مقدار "زیاد" را به خود اختصاص خواهد داد.

پس‌از درج صفت فوق در رئوس گراف، الگوریتم SIGS با ورودی صفت *فعالیت* روی گراف همکاری نویسندگان اجرا شد تا خلاصه‌هایی با اندازه‌های ۷-۳ تولید کند.



شکل ۳. گراف‌های خلاصه تولید شده توسط SIGS برای گراف وبلاگ‌های سیاسی

شکل (۴) برخی گراف‌های خلاصه تولید شده به‌وسیله الگوریتم SIGS در اجراهای مختلف را نشان می‌دهد. همان‌طور که در شکل (۴) نیز مشخص است، اولین خروجی، گراف خلاصه‌ای به اندازه ۳ است که در واقع همان مدل سازگار با صفات است. این گراف یک شمای کلی از رفتار نویسندگان مختلف در گراف DBLP را نشان می‌دهد. به‌طور خلاصه از این گراف می‌توان فهمید که نویسندگان فعال، به‌طور تقریبی با تمام دیگر نویسندگان همکاری داشته‌اند. از طرفی نویسندگانی که فعالیت آن‌ها کم بوده به‌صورت تقریبی همکاری ضعیفی با سایر نویسندگان داشته‌اند. اما در عوض همکاری آنها با خودشان درصد قابل توجهی است.

درباره نویسندگانی که فعالیت آنها متوسط بوده است نیز نکته جالبی به چشم می‌خورد. در واقع این دسته از نویسندگان با سایر گروه‌ها رابطه نزدیک به ۱۰۰ دارند، درحالی که همکاری آنها با هم‌نوعان

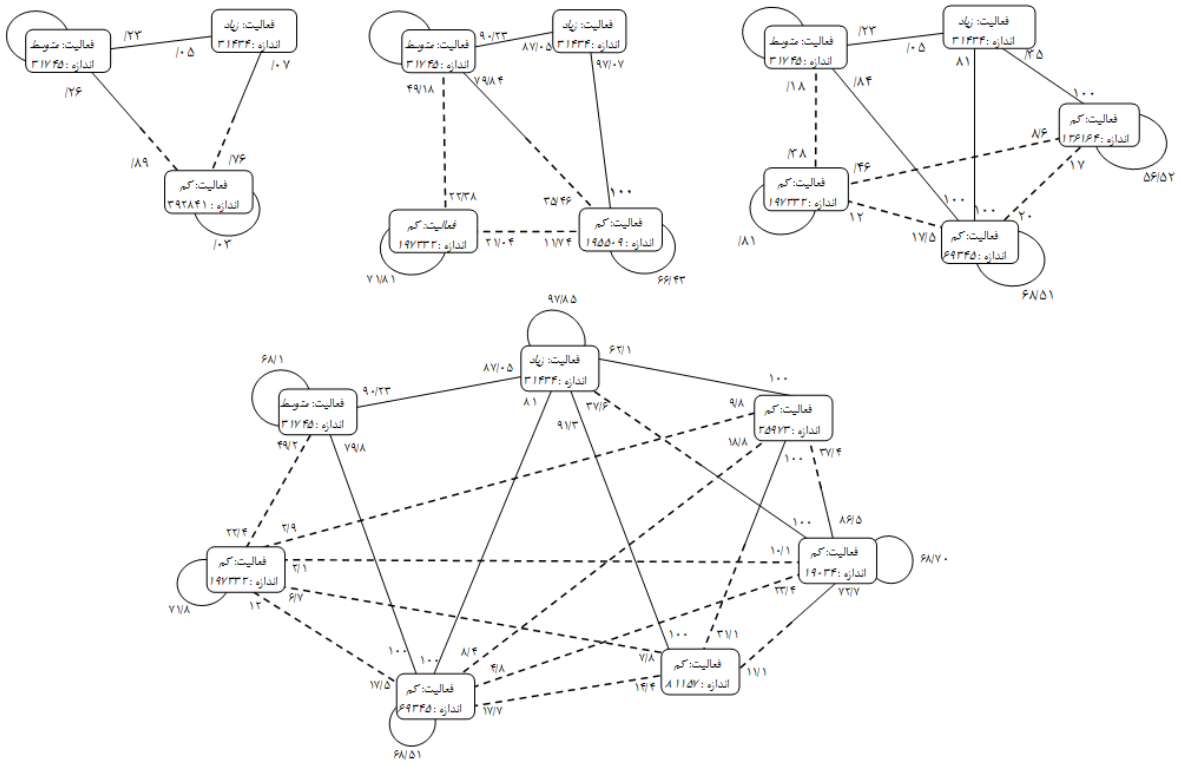
خودشان کمتر است. در واقع این دسته از نویسندگان بیشتر به همکاری با سایرین تمایل نشان داده‌اند. از طرفی در این گراف جزئیات کافی وجود ندارد. در واقع رأس مربوط به نویسندگان کم فعالیت شامل نزدیک به ۳۹۳هزار رأس از گراف اولیه است که به‌طور تقریبی ۸۶٪ کل گراف است. بنابراین نیاز است که برای کسب اطلاعات بیشتر از رفتار این دسته بزرگ، خلاصه‌ای با اندازه بزرگ‌تر تولید شود.در گراف خلاصه با اندازه ۴، دو رأس مربوط به نویسندگان کم فعال وجود دارد که یکی به‌هیچ وجه با نویسندگان فعال مرتبط نبوده و دیگری رابطه کامل با نویسندگان فعال دارد.

نکته جالب دیگر در مورد این دو رأس این است که رابطه رأس دوم با نویسندگان با فعالیت متوسط نیز بیشتر از رأس اولی است. همچنین ارتباط این دو رأس جدید با یکدیگر نیز بسیار ضعیف است که تأییدی برای تجزیه آن‌دو به‌وسیله الگوریتم می‌باشد.

بعد از انجام تجزیه بعدی و افزایش اندازه گراف خلاصه، مشاهده می‌شود که باز یکی از رئوس مربوط به نویسندگان کم فعال تجزیه شده است. گراف خلاصه حاصل از این تجزیه نیز اطلاعات جالبی را در اختیار می‌گذارد.

براساس آنچه از این گراف قابل استفاده است، گروه نویسندگان کم فعالیت را می‌توان به سه دسته تقسیم کرد. دسته اول نویسندگانی هستند که همکاری بسیار قوی با نویسندگان فعال و نیز با فعالیت متوسط دارند. تعداد این دسته در اقلیت است. پس از آنها دسته دیگری از نویسندگان کم فعالیت هستند که همکاری آنها با نویسندگان فعال بسیار قوی است ولی هیچ رابطه‌ای با نویسندگان با فعالیت متوسط ندارند. تعداد این نویسندگان از دسته قبل بیشتر است. اما دسته سوم که اکثریت نویسندگان کم فعالیت را تشکیل می‌دهند، رابطه ضعیفی با دسته‌های دیگر دارند. اما رابطه این دسته از نویسندگان کم فعال با خودشان قوی‌تر از دو دسته دیگر است. نکته جالب دیگری که درباره گراف خلاصه با اندازه ۵ به چشم می‌خورد این است که برخلاف خلاصه‌های قبلی (و بر اثر تصمیمات صحیح SIGS در تجزیه رئوس) تمام ارتباط‌ها یا از هر دوطرف ضعیف یا قوی بوده است و رابطه‌ای که از یک‌طرف قوی و از طرف دیگر ضعیف‌باشد، وجود ندارد که این می‌تواند معیاری برای دستیابی به خلاصه‌ای با اندازه مناسب تلقی شود.

گراف‌های خلاصه به‌دست آمده در مراحل بعدی بزرگ‌تر شده اما همان‌طور که در شکل (۴) نیز یکی از آنها نشان داده‌شده است، این گراف‌ها اطلاعات خاصی را نسبت به آنچه در گراف خلاصه با اندازه ۵ به‌دست آمده بود، در اختیار نمی‌گذارد، بلکه تنها با افزایش جزئیات و پیچیدگی‌ها، کار درک گراف و کسب دانش کلی را با مشکل مواجه می‌سازند. بنابراین به‌نظر می‌رسد برای کاربر عادی که قصد فهم ساختار کلی گراف و رفتار نویسندگان را دارد، گراف خلاصه با اندازه ۵ گزینه مناسبی است. اما ازطرفی گراف‌های خلاصه بزرگ‌تر می‌توانند به‌عنوان ورودی الگوریتم‌های کاوش گراف استفاده شوند.

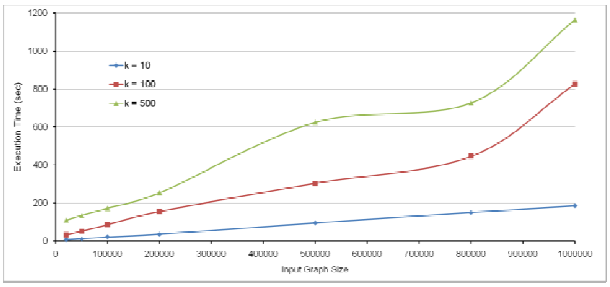


شکل ۴. گراف‌های خلاصه تولیدشده توسط SIGS

۵-۵. آزمون کارایی و مقیاس‌پذیری

کارایی و مقیاس‌پذیر بودن ازجمله مهم‌ترین پارامترهای لازم برای یک الگوریتم خلاصه‌سازی است. برهمین اساس، در این بخش با استفاده از مجموعه *گراف‌های شبیه‌سازی شده* که در اندازه‌های مختلف تولید شده‌اند، کارایی الگوریتم SIGS مورد ارزیابی قرار گرفته است.

شکل (۵) نمودار زمان اجرای الگوریتم SIGS برای خلاصه‌سازی گراف‌های مذکور را برای خلاصه‌هایی با اندازه‌های ۱۰، ۱۰۰ و ۵۰۰ نشان می‌دهد.همان‌طور که مشاهده می‌شود، الگوریتم SIGS موفق شده است گرافی با اندازه یک میلیون رأس را در زمانی نزدیک به ۸۰۰ ثانیه (کمتر از ۱۴ دقیقه) به خلاصه‌ای به اندازه ۱۰۰ تبدیل نماید. این درحالی است که آزمون کارایی که در مرجع [۲۵] انجام شده نشان می‌دهد الگوریتم kSNAP همین کار را در زمانی نزدیک به ۳۰۰۰ ثانیه (حدود ۵۰ دقیقه) انجام داده است. این موضوع مؤید افزایش قابل توجه کارایی و سرعت الگوریتم SIGS نسبت به نمونه مشابه در مرجع [۲۵] است. از طرفی از شکل (۵) مشخص است که الگوریتم SIGS توانسته است گراف ۱۰۰۰۰۰۰تایی را در زمانی کمتر از ۱۲۰۰ ثانیه (۲۰ دقیقه) به خلاصه‌ای با اندازه ۵۰۰ تبدیل کند. این درحالی است که برای اجرای الگوریتم، تنها از یک سیستم استفاده می‌شده است. بنابراین مقیاس‌پذیری الگوریتم SIGS نیز در حد قابل قبولی ارزیابی می‌شود. درواقع پایگاه داده Neo4j که زیربنای اصلی کار با گراف در نمونه پیاده‌سازی شده الگوریتم SIGS است، در این آزمون به‌صورت غیر توزیع‌شده استفاده شده است. این در حالی است که با افزایش سیستم‌ها و اجرای توزیع‌شده پایگاه‌داده Neo4j می‌توان سرعت و مقیاس‌پذیری را افزایش نیز داد.



شکل ۱. نمودار زمان اجرای SIGS برای گراف‌ها و خلاصه‌های مختلف

۶. نتیجه‌گیری

امروزه با توجه به افزایش چشم‌گیر حجم و پیچیدگی داده‌های موجود در حوزه ها و موضوعات مختلف، استفاده از ساختاری مناسب برای نمایش و پردازش این داده‌ها چالش و مسئله‌ای بسیار مهم محسوب می شود. برهمین اساس، گراف به‌عنوان ساختاری ساده، سطح بالا، قابل‌فهم و انعطاف‌پذیر به‌عنوان راه‌حلی مناسب برای مدل‌سازی، نمایش و پردازش داده‌ها معرفی شده است. این ساختار می تواند انواع موجودیت‌ها و روابط میان آنها را به‌سادگی نشان دهد. از طرفی با گسترش و افزایش استفاده از گراف‌ها در حوزه‌های مختلف، موضوع کاوش گراف به موضوعی مهم بدل شده که امروزه توجه بسیاری را به خود معطوف ساخته است. حوزه هایی مانند نرم‌افزار (گراف ها و نمودارهای استفاده شده برای مدل کردن و تحلیل ساختار نرم‌افزارها)، شبکه (گراف توپولوژی شبکه یا شبیه‌سازی ترافیک)، شیمی (گراف ترکیبات عناصر)، زیست (گراف ترکیبات زیستی) و حتی جامعه‌شناسی (گراف شبکه های اجتماعی و روابط اعضای جوامع) تنها برخی از حوزه‌هایی هستند که گراف‌ها در آن برای کار با داده‌ها استفاده می‌شوند.

با توجه به حجم زیاد داده‌ها در هر زمینه، گراف‌های داده‌ای، اغلب به مدل‌هایی بسیار بزرگ و پیچیده تبدیل می‌شوند که یافتن و استخراج اطلاعات مورد نیاز کاربران از آنها بسیار دشوار است. حتی الگوریتم‌های کاوش گراف نیز در مواجهه با گراف‌های عظیم‌الجثه، کارایی خود را از دست می‌دهند. در چنین موقعیت‌هایی خلاصه‌سازی و تولید یک گراف خلاصه که از گراف اولیه ساده‌تر بوده و در عین حال خصوصیات و داده‌های مهم موجود در گراف اولیه را شامل باشد، راه‌حل مناسبی محسوب می‌شود.از یک منظر می‌توان خلاصه سازی گراف را به دو روش انجام داد: روش ساختاری و روش معناگرا. در روش ساختاری، خلاصه‌سازی براساس خصوصیات ساختاری گراف، مانند همسایگی میان رئوس یا چگالی زیرگراف‌های مختلف انجام می‌گیرد. در چنین روشی، گراف خلاصه حاوی اطلاعات برجسته موجود در ساختار گراف است. اما درعین حال به ازای هر گراف تنها می‌توان یک خلاصه به‌دست آورد که مستقل از اطلاعات و دانش مورد نیاز کاربران مختلف است. روش دوم خلاصه‌سازی است که خلاصه‌سازی معناگرا نام دارد، می‌کوشد تا خلاصه‌ای از گراف اولیه متناسب با نیاز کاربر تولید کند که در آن اطلاعات مورد نیاز کاربر برجسته و اطلاعات نامرتبط حتی الامکان کم‌رنگ شده باشند. بنابراین روش خلاصه‌سازی معناگرا این امکان را به کاربر می‌دهد که از یک گراف، خلاصه‌های متعدد و متفاوت تولید کرده و نیازهای خود را تاحد قابل قبولی مرتفع سازد.براساس مطالب فوق، در مقاله حاضر روشی نو برای خلاصه‌سازی معناگرای گراف‌ها ارائه شده است. این الگوریتم که SIGS نام گرفته، قادر است گراف‌ها را براساس اطلاعات موردنیاز کاربر خلاصه کرده و میزان جزئیات و اندازه گراف خلاصه را بنابه درخواست کاربر تغییر دهد. از طرفی با استفاده از بستر مناسب و خاص منظوره برای کار با گراف‌ها، الگوریتم SIGS به‌گونه‌ای پیاده‌سازی شده تا از لحاظ پارامترهای کارایی و مقیاس‌پذیری از نمونه‌های مشابه خود برتر بوده و پاسخ‌گوی نیاز کاربران معمولی نیز (که اغلب به سخت‌افزارهای پیشرفته دسترسی نداشته و در عین حال زمان و حوصله زیادی نیز ندارند) باشد. همچنین الگوریتم SIGS به‌وسیله داده های متنوع مورد ارزیابی قرار گرفت. بررسی و ارزیابی نتیجه‌های آزمون‌ها نشان می‌دهد که خلاصه‌های تولید شده توسط SIGS حاوی اطلاعات مورد انتظار بوده و از کیفیت قابل قبولی برخوردار هستند. علاوه‌بر این، الگوریتم SIGS از لحاظ کارایی و مقیاس‌پذیری نسبت به نمونه‌های مشابه خود برتر است.

با وجود کارهای انجام گرفته در حوزه خلاصه‌سازی گراف‌ها، هم‌چنان مسائلی در این زمینه وجود دارند کـه نیازمنـد تـلاش و کـار بیشـتر هستند. از جملـه ایـن مسـائل مـی‌تـوان بـه موضـوع خلاصـه‌سـازی گراف‌های پویا اشاره کرد. این گراف‌ها، گراف‌هایی هستند که ساختار آنها به مرور زمان تغییر می‌کند. گراف تماس‌ها در یک شبکه تلفنـی یا رشته‌های گرافی[1] مانند گـراف فراخـوانی‌هـای زمـان اجـرای یـک نرم‌افـزار نمونـه‌هـایی از ایـن‌گونـه گـراف‌هـا هسـتند. امـا در حـوزه خلاصه‌سازی معناگرای گراف‌های ایستا و در ادامه کار انجام گرفته در مقاله حاضر و به‌عنوان کارهای آتی نیز می‌توان مـواردی را پیشـنهاد کـرد. بـرای نمونـه مـی‌تـوان گفـت کـه اسـتفاده از تکنیـک‌هـای هستان‌شناسی و وجود یک آنتولـوژی دربـاره صـفات رئـوس گـراف، می‌تواند الگوریتم خلاصه‌سازی را در انتخاب گروه‌های مناسـب بـرای تجزیه بیشتر راهنمایی کنـد. همچنـین درصـورت وجـود آنتولـوژی، الگوریتم می‌تواند در صورت نیاز صفات هم‌معنی و مـرتبط بـا صـفات انتخابی کاربر را نیز در خلاصه‌سازی دخالت دهد. طراحی و تولید یک زبان محاوره میان کاربر و الگوریتم SIGS که به کاربر امکان افزایش و کاهش اندازه گراف خلاصه را بدون نیاز بـه اجـرای مجـدد الگـوریتم می‌دهد نیز می‌توانـد از جملـه دیگـر کارهـای آتـی باشـد. از طرفـی همان‌طور که گفته شد، خلاصه‌سازی گـراف‌هـای پویـا و رشـته‌هـای گرافی می‌تواند دورنمای ایده‌آلی بـرای هـر الگـوریتم خلاصـه‌سـازی گراف باشد. بنابراین افزودن قابلیت خلاصه‌سازی گراف‌های پویـا نیـز به‌عنوان یکی از کارهای آتی این پروژه پیشنهاد می‌شود.

۷. مراجع

[1] Chaoji1, V.; Al Hasan, M.; Salem, S.; Besson, J.; Zaki, M. "ORIGAMI: A Novel and Effective Approach for Mining Representative Orthogonal Graph Patterns."; Statistical Analysis and Data Mining 2008, 1, 67-84.

[2] Ivancsy, R.; Vajk, I. "Frequent Pattern Mining in Web Log Data."; Acta Politechnica Hungarica 2006, 3, 77-90.

[3] Abello, J.; Resende, M. G.; Sudarsky, S. "Massive Quasi-Clique Detection."; In Proc. of the 5th Latin American Symposium on Theoretical Informatics (LATIN) 2002, 598-612.

[4] Murata, T. "Graph Mining Approaches for the Discovery of Web Communities."; In Proc. of the First International Workshop on Mining Graphs, Trees and Sequences 2003, 199-208.

[5] Ting, I. H. "Web-Mining Applications in E-Commerce and E-Services."; Online Information Review 2008, 32, 129-132.

[6] Yan, X.; Han, J. "Gspan: Graph-Based Substructure Pattern Mining."; In Proc. of 2002 International Conference on Data Mining (ICDM'02) 2002, 721–724.

[7] Kudo, T.; Maeda, E.; Matsumoto, Y. "An Application of Boosting to Graph Classification."; Advances in Neural Information Processing Systems 18 (NIPS'04) 2004, 729-736.

[8] Nijssen, S.; Kok, J. "A Quickstart in Frequent Structure Mining Can Make a Difference."; In Proc. of 2004 ACM SIGKDD International Conference on Knowledge Discovery in Databases (KDD'04) 2004, 647–652.

[9] Holder, L. B.; Cook, D. J.; Djoko, S. "Substructure Discovery in the Sub-Due System."; In Proc. of AAAI'94 Workshop Knowledge Discovery in Databases (KDD'94) 1994, 169–180.

[10] Huan, J.; Wang, W.; Prins, J. "Efficient Mining of Frequent Subgraph in The Presence of Isomorphism."; In Proc. of 2003 International Conference on Data Mining (ICDM'03) 2003, 549–552.

[11] Han, J.; Pei, J.; Yin, Y.; Mao, R. "Mining Frequent Patterns Without Candidate Generation: A Frequent-Pattern Tree Approach."; Data Mining and Knowledge Discovery 2004, 8, 53-87.

[12] Saigo, H.; Krämer, N.; Tsuda, K. "Partial Least Squares Regression for Graph Mining."; In Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) 2008, 578-586.

[13] Chen, C.; Lin, C. X.; Fredrikson, M.; Christodorescu, M.; Yan, X.; Han, J. "Mining Graph Patterns Efficiently Via Randomized Summaries."; In Proc. of VLDB Conference 2009, 742-753.

[14] Flake, G. W.; Tarjan, R. E.; Tsioutsiouliklis, K. "Graph Clustering and Minimum Cut Trees."; Internet Mathematics 2003, 1, 385-408.

[15] Newman, M.; Girvan, M. "Finding and Evaluating Community Structure in Networks."; Physical Review E, 2004, No. 69:026113.

[16] Andersen, R.; Chellapilla, K. "Finding Dense Subgraphs with Size Bounds."; In Proc. of 6th Int. Workshop on Algorithms and Models for the Web-Graph, Springer-Verlag, 2009, 25–37.

[17] Wikipedia, "Summary."; Internet: http://en.wikipedia.org/wiki/Summary, Dec. 15, 2010.

[18] Navlakha, S.; Rastogi, R.; Shrivastava, N. "Graph Summarization with Bounded Error."; In Proc. of the ACM SIGMOD International Conference on Management of Data (SIGMOD'08) 2008, 419–432.

[19] LeFevre, K.; Terzi E. "GraSS: Graph Structure Summarization."; In Proc. of the SDM 2010, 454-465.

[20] Chen, C.; Lin, C.; Fredrikson, M.; Christodorescu, M.; Yan, X.; Han, J. "Mining Graph Patterns Efficiently *via* Randomized Summaries."; In Proc. of VLDB Endowment 2009, 2, 742-753.

[21] Aggarwal, C. C.; Xie, Y.; Yu, P. "GConnect: A Connectivity Index for Massive Disk-Resident Graphs."; In Proc. of VLDB Endowment 2009, 2, 862-873.

[22] Aggarwal, C. C.; Wang, H. "Graph Data Management and Mining: A Survey of Algorithms and Applications."; in Managing and Mining Graph Data, C. C. Aggarwal, and H. Wang, London: Springer 2010, 13-68.

[23] Chen, C.; Yan, X.; Zhu, F.; Han, J.; Yu, P. S. "Graph OLAP: Towards Online Analytical Processing on Graphs."; In Proc. of 8th Int. IEEE Conference on Data Mining (ICDM) 2008, 103–112.

[24] Zhang, N.; Tian, Y.; Patel, J. M. "Discovery-Driven Graph Summarization."; In Proc. of the IEEE International Conference on Data Engineering (ICDE'10) 2009, 880-891.

[25] Tian, Y.; Hankins, R. A.; Patel, J. M. "Efficient Aggregation for Graph Summarization."; In Proc. of 2008 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'08) 2008, 567–580.

[26] Battista, G.; Eades, P.; Tamassia, R.; Tollis, I. "Graph Drawing: Algorithms for the Visualization of Graphs."; Prentice Hall 1999, 385-408.

[27] Wikipedia, "NoSQL."; Internet: http://en.wikipedia.org/wiki/NoSQL Mar. 21, 2011.

[28] Viger, F.; Latapy, M. "Random Generation of Large Connected Simple Graphs with Prescribed Degree Distribution."; In Proc. Of The 11-Th International Conference on Computing and Combinatorics COCOON'05 2005, 440-449.

[29] Newman, M. "The Structure and Function of Complex Networks."; SIAM Review 2003, 45, 167-256.

[1] Graph Streams