



دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

گزارش پروژه‌ی پایانی درس الگوریتم‌های شبکه‌های پیچیده

عنوان مقاله:

یک الگوریتم سریع جهت کشف انجمن‌ها در شبکه‌های
متغیر در گذر زمان (Temporal Networks)

دانشجو:

سید احمد نقوی نوزاد

ش-د: ۹۴۱۳۱۰۶۰

استاد درس:

دکتر امیر حائری

بهار ۱۳۹۵

سید علی

(۱) مقدمه:

بسیاری از سیستم‌های پیچیده نظیر شبکه‌های همکاری میان مؤلفین، شبکه‌های وبلاگ‌نویسان، شبکه‌های اجتماعی نظیر فیس‌بوک و توئیتر و مانند آن را می‌توان در قالب یک گراف متشکل از گره‌ها و اتصالات موجود میان آن‌ها نشان داد. مثلاً در مورد شبکه‌های همکاری میان مؤلفین، گره‌ها نمایانگر مؤلفین و یال‌ها معرف همکاری میان مؤلفین می‌باشند؛ در مورد شبکه‌های وبلاگ‌نویسان نیز گره‌ها بیانگر وبلاگ‌نویسان و یال‌ها هم معرف ابرلینک‌های موجود در وبلاگ‌ها می‌باشد و در مورد شبکه‌های اجتماعی هم، گره‌ها معرف مردم بوده و یال‌ها نیز نشان‌دهنده‌ی تعاملات اجتماعی میان افراد می‌باشد. ساختار انجمن‌های موجود در یک شبکه‌ی پیچیده، یکی از خصیصه‌های ذاتی آن می‌باشد که معرف مجاورت و قرابت گروه‌هایی متشکل از گره‌هایی می‌باشد که تمایل دارند تا اتصالات عمیق‌تری را درون خود گروه مربوطه برقرار کنند تا میان گروه‌های موجود؛ و البته پیمانی^۱ نیز معیاری می‌باشد که یک عدد مابین ۱- و ۱ بوده و معرف میزان چگالی اتصالات درون انجمن‌ها می‌باشد تا اتصالات میان آن‌ها، و جهت ارزیابی عملکرد الگوریتم‌های خوشه‌بندی و یا همان کشف انجمن به کار می‌رود. ضمن این تفاسیر در بسیاری از شبکه‌های پیچیده، زمان نقش مهمی را ایفا می‌نماید و به عبارتی این شبکه‌ها در طول زمان دستخوش تغییراتی در تعداد گره‌ها و البته اتصالات میان آن‌ها می‌شوند و بدین سبب شبکه‌های متغیر در گذر زمان نامیده می‌شوند. اگر بخواهیم روند رو به رشد موجودیت‌های درون این گونه شبکه‌ها را بهتر درک نمائیم، دانستن ساختار انجمن‌های موجود درون این شبکه‌ها می‌تواند بسیار کمک‌کننده بوده و به همین سبب رویکردهای متعددی تاکنون ارائه گشته‌اند که البته هر یک از مزایا و معایبی برخوردار می‌باشند. مثلاً در مورد روش‌های سنتی و مرسوم بسیاری که تا پیش از این استفاده می‌شده‌اند، بدین‌گونه است که کشف اجتماعات موجود در شبکه در هر مرحله یا بازه‌ی زمانی^۲، مستقل از مراحل زمانی پیشین می‌باشد و به همین سبب بسیاری از اطلاعات تاریخچه‌ای مربوط به اجتماعات موجود نادیده گرفته شده و بهره‌وری کار پایین می‌آید. در این مقاله روشی را جهت کشف اجتماعات در شبکه‌های متغیر در گذر زمان پیشنهاد می‌کنیم که در عین اینکه کیفیت و کارایی روند مربوطه را تضمین می‌نماید، از اطلاعات تاریخچه‌ای اجتماعات نیز بهره برده و علاوه بر این با توجه به نتایج آزمایشات انجام‌شده، نسبت به روش‌های سنتی از نظر زمان صرف‌شده جهت پردازش در CPU، بهبود ۶۹ درصدی را از خود نشان می‌دهد.

(۲) شرح روش و پارامترها:

در روش پیشنهادی در این مقاله، ما از الگوریتمی به نام روش بلاندل^۳ بهره می‌بریم که جهت کشف انجمن‌های موجود در یک شبکه‌ی ایستا^۴ معرفی گشته است. در این روش که از چند مرحله تشکیل می‌شود و هر مرحله نیز شامل دو فاز می‌باشد، ابتدا در مرحله‌ی اول و در فاز اول این مرحله، به هر کدام از گره‌های موجود در شبکه یک انجمن مجزا نسبت داده می‌شود و به عبارتی در ابتدای کار، ما به تعداد گره‌ها، انجمن خواهیم داشت. سپس در ادامه‌ی این فاز، هر گره را به انجمن یکی از همسایگانش نسبت داده و معیار پیمانی را به ازای ساختار انجمنی جدید حاصله محاسبه می‌نمائیم و در نهایت گره‌ی مربوطه را به انجمن آن همسایه‌ای نسبت می‌دهیم که به ازای این اطلاق، مقدار پیمانی حاصله بیشینه و البته مثبت^۵ باشد؛ و در ادامه این کار را به ازای هر کدام از گره‌ها به صورت ترتیبی انجام می‌دهیم. پس از پایان یک دور اطلاق به انجمن‌های همسایه به ازای همه‌ی گره‌ها، دوباره

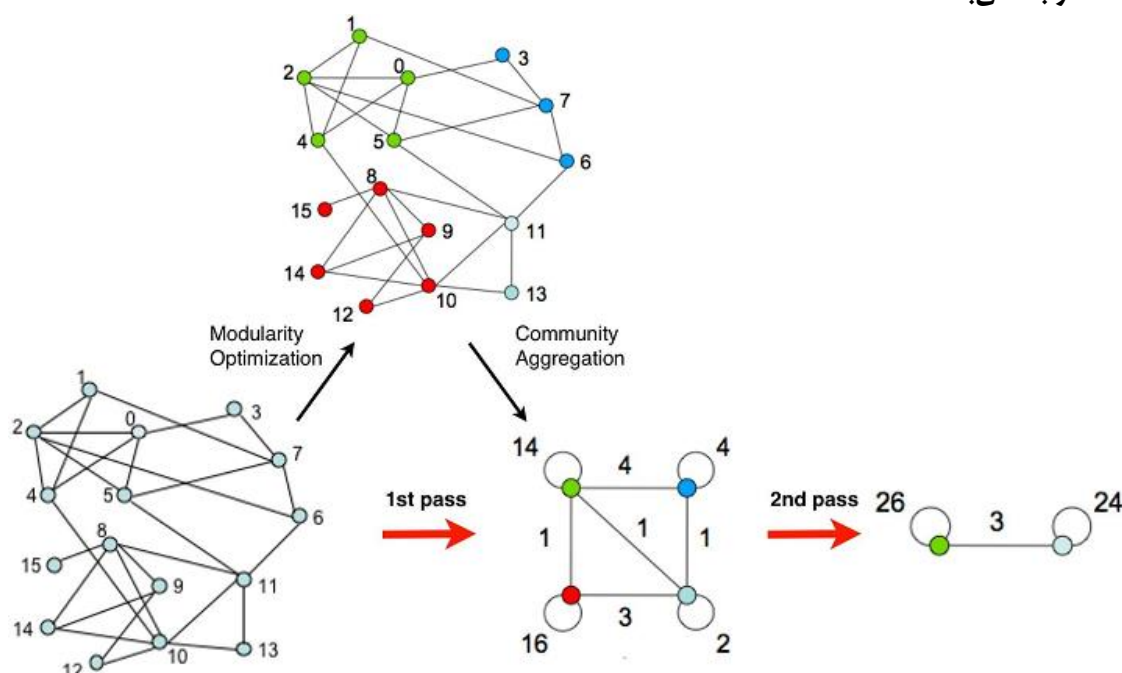
¹ Modularity

² Time step

³ Blondel method

⁴ Static Network

این کار را بر روی تمام گره‌ها به صورت تکراری انجام می‌دهیم و این رویه‌ی تریبی و تکراری^۵ زمانی خاتمه می‌یابد که در تمام یک مرحله، هیچ سود مثبت پیمانی حاصل نگردد. حال پس از پایان فاز اول در شبکه‌ی فعلی، به ازای هر انجمن، یک گره در شبکه‌ی جدید در نظر گرفته و مجموع وزن‌های اتصالات میان گره‌های موجود میان هر دو انجمن را به عنوان وزن یال مابین دو گره‌ی نماینده‌ی انجمن‌های مربوطه در گراف جدید در نظر می‌گیریم. مجموع وزن‌های اتصالات موجود درون هر انجمن نیز به عنوان وزن یک یال برگشتی و یا هم یال به خودی^۶ برای گره‌ی نماینده‌ی انجمن مربوطه در گراف جدید در نظر گرفته می‌شود. اکنون یک مرحله از الگوریتم بلاندل پایان یافته است و می‌توانیم گراف جدید حاصله از فاز دوم مرحله‌ی پیشین را دوباره با استفاده از الگوریتم بلاندل خوشه‌بندی نمائیم و این روند تا زمانی ادامه می‌یابد که تعداد انجمن‌های کشف‌شده در دو مرحله‌ی متوالی یکسان باشند. به عبارتی در هر مرحله از الگوریتم، یک ساختار انجمنی از انجمن‌های کشف‌شده در مرحله‌ی پیشین حاصل می‌شود که در نهایت به یک ساختار سلسله‌مراتبی^۷ از انجمن‌ها ختم می‌گردد. تصویر زیر نشانگر مراحل و فازهای قیدشده در بالا می‌باشد:



شکل ۱. دو مرحله از اجرای الگوریتم بلاندل بر روی یک گراف فرضی در سمت چپ که هر مرحله از دو فاز تشکیل شده است؛ فاز اول «بهینه‌سازی پیمانی» و فاز دوم «تجمیع انجمن‌ها» نامیده می‌شوند. در نهایت پس از اجرای دو مرحله از الگوریتم بر روی گراف مربوطه، گراف حاصل دارای دو انجمن خواهد بود.

و اما در روش پیشنهادی در این مقاله نیز، علاوه بر این که رسماً از الگوریتم بلاندل (که پیش از این روند کار آن قید گردید) استفاده می‌نمائیم، از رویه‌ی استفاده‌شده در این الگوریتم نیز جهت ساخت اصطلاحاً یک شبکه‌ی کوچک^۸ الگوبرداری نموده و سپس انجمن‌های موجود در این شبکه‌ی جدید را نیز با استفاده از همان الگوریتم بلاندل کشف می‌نمائیم.

با توجه به این که بسیاری از روش‌های سنتی جهت کشف انجمن‌ها در شبکه‌های پیچیده‌ی متغیر در گذر زمان، در هر مرحله‌ی زمانی به صورت کاملاً مستقل از مراحل پیشین انجمن‌ها را شناسایی می‌نموده‌اند و به عبارتی

⁵ Sequentially and repeatedly

⁶ Self-loop

⁷ Hierarchical structure

⁸ Small Network

اطلاعات تاریخی مربوط به انجمن‌ها در این گونه روش‌ها نادیده گرفته می‌شده‌اند، لذا استفاده از این روش‌ها در مورد شبکه‌های عظیم، سبب افزایش محاسبات زائد می‌گردد.

اما در مورد روش پیشنهادی در این مقاله باید گفت که علاوه بر این که تنها از ساختارهای انجمنی مرحله‌ی زمانی پیشین بهره می‌برد، تنها تغییرات ایجادشده در ساختار شبکه‌ی فعلی نسبت به مرحله‌ی قبل را در نظر می‌گیرد و در نتیجه سبب کاهش چشمگیر حجم محاسبات نسبت به روش‌های سنتی می‌گردد. به عبارتی در مورد دو مرحله‌ی زمانی متوالی، تنها تغییرات جزئی‌ای در تعداد یال‌ها ایجاد می‌شود که تأثیر چشم‌گیری در ساختار انجمن‌های موجود در گراف نداشته و به همین سبب ما نیز این‌گونه استنتاج می‌نمائیم که اگر اتصالات موجود میان گره‌های یک انجمن خاص در مرحله‌ی زمانی $t-1$ ، در مرحله‌ی زمانی t نیز برقرار باشند، این گره‌ها همچنان در همان انجمن پیشین حضور دارند و البته گره‌هایی که دچار تغییر درجه، چه از نوع افزایشی و یا کاهشی شده‌اند، هر یک در ساختار شبکه‌ی فعلی به عنوان یک گره‌ی مجزا در نظر گرفته شده و یک انجمن جدید به آن‌ها نسبت داده می‌شود. حال زمان آن رسیده است تا شبکه‌ی کوچک (small network) که پیش از این به آن اشاره گردید را از روی ساختار شبکه‌ی فعلی و با توجه به ساختار انجمنی جدید (که از روی ساختار انجمنی مرحله‌ی پیشین ساخته شد)، بسازیم. بدین منظور همانند آن‌چه در فاز دوم الگوریتم بلاندل اتفاق افتاد عمل نموده و هر انجمن را در ساختار شبکه‌ی کوچک جدید، به عنوان یک گره‌ی جدید در نظر گرفته و مجموع وزن‌های اتصالات میان گره‌های دو انجمن را به عنوان وزن یال میان گره‌های نماینده‌ی انجمن‌های مربوطه در نظر می‌گیریم و همین‌طور مجموع وزن‌های میان گره‌های درون یک انجمن خاص را به عنوان وزن یک یال برگشتی برای گره‌ی نماینده‌ی آن انجمن در شبکه‌ی کوچک جدید ثبت می‌کنیم. حال زمان آن رسیده تا از الگوریتم بلاندل جهت کشف انجمن‌ها در شبکه‌ی کوچک ساخته‌شده استفاده نمائیم و در نهایت طی یک ساختار سلسه‌مراتبی به هر کدام از گره‌های موجود در شبکه‌ی اصلی در مرحله‌ی زمانی فعلی، یک انجمن را نسبت دهیم. الگوریتم پویای کشف انجمن پیشنهادی در این مقاله که کلیات آن قید گردید، به صورت مرحله به مرحله در ادامه می‌آید:

(۱) ساختار انجمن‌های موجود در شبکه‌ی G_1 در مرحله‌ی زمانی یک را با استفاده از الگوریتم بلاندل تعیین می‌نمائیم.

(۲) برای شبکه‌ی G_t در مراحل زمانی $t = 2, 3, \dots, T$ به صورت زیر عمل می‌نمائیم:

أ) یک شبکه‌ی کوچک G_{t_new} را با توجه به ساختار شبکه‌ی موجود در G_t و البته اطلاعات انجمنی موجود در G_{t-1} می‌سازیم.

ب) انجمن‌های موجود در شبکه‌ی G_{t_new} را با استفاده از الگوریتم بلاندل کشف می‌نمائیم.

با توجه به این که در این روش پیشنهادی مانند روش‌های سنتی، از الگوریتم بلاندل استفاده می‌نمائیم، پیچیدگی زمانی هم‌چنان $O(m)$ خواهد بود که m برابر تعداد یال‌های موجود در شبکه می‌باشد. اما از آن‌جا که در این روش، یک شبکه‌ی کوچک ساخته می‌شود، زمان محاسبات شدیداً کاهش یافته و در نتیجه ضریب پیچیدگی زمانی در این روش نسبت به روش سنتی کوچک‌تر می‌باشد.

(۳) مجموعه داده‌های مورد استفاده

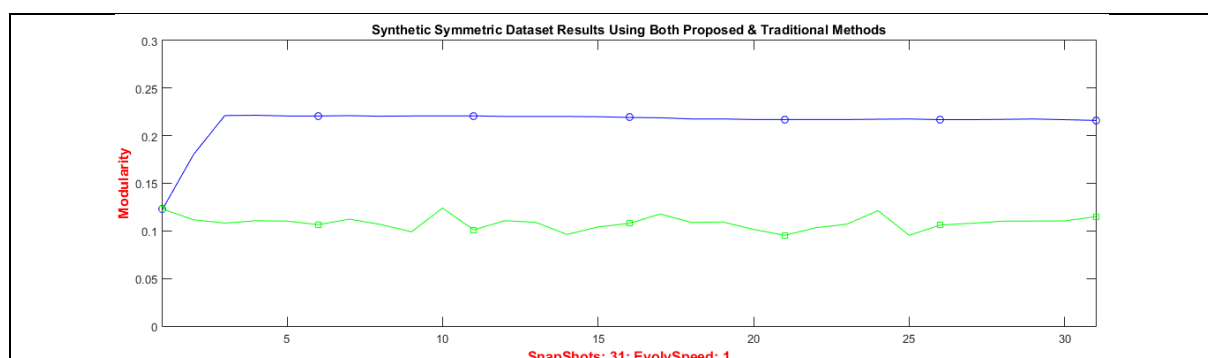
در این روش از مجموعه داده‌های واقعی و مصنوعی استفاده نموده‌ایم، که هر کدام از آن‌ها به صورت یک لیست یال^۹ متشکل از ۴ ستون می‌باشند. به این صورت که در هر ردیف، ستون اول نمایانگر گره‌ی مبدأ، ستون

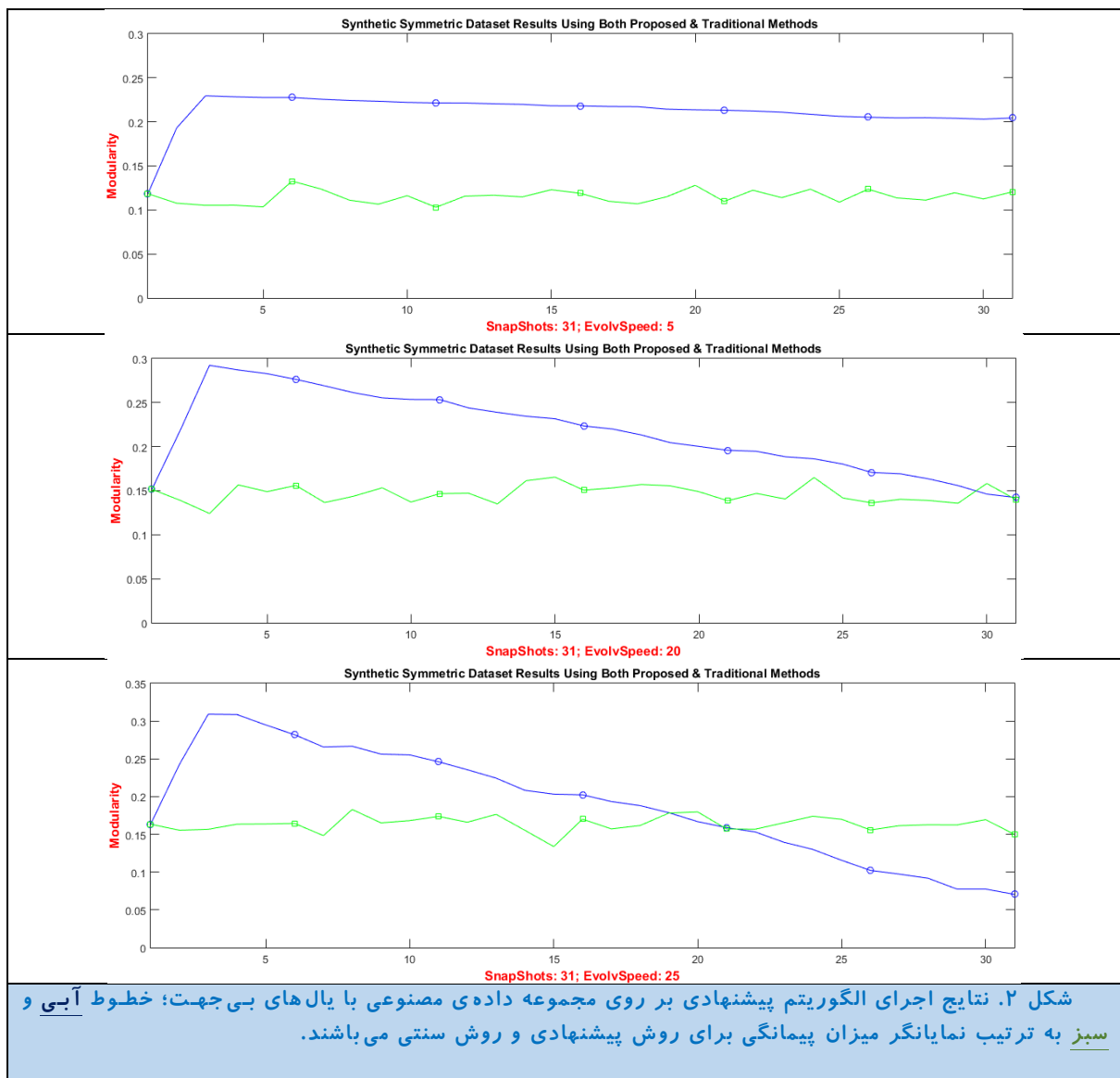
^۹ Edge-list

دوم نمایانگر گرهی مقصد، ستون سوم نمایانگر وزن یال مربوطه و ستون آخر نیز حاوی برچسب زمانی یال مربوطه بر حسب میلی ثانیه می باشد. در ابتدا بر طبق آن چه در مقاله قید شده، یال های برگشتی و البته گره هایی که درجه ی آن ها از یک حد آستانه ی مشخص $\sigma * \mu - 1$ (که μ و σ به ترتیب نشانگر میانگین و انحراف از معیار توزیع مربوط به درجات گراف می باشند) کمتر می باشد را حذف می نماییم (از این گره ها در متن مقاله تحت عنوان داده ی پرت^{۱۰} نام برده شده است). سپس برچسب زمانی هر یک را از میلی ثانیه به دقیقه تبدیل نموده و با توجه به این برچسب زمانی، از هر کدام از مجموعه داده ها با توجه سرعت های رشد متفاوت، تعداد ۳۱ عدد تصویر لحظه ای تهیه می نماییم. بدین ترتیب که مثلاً اگر یال های موجود در یک مجموعه داده، سرجمع در یک بازه ی زمانی [0,1000] دقیقه اتفاق افتاده باشند و بخواهیم تعداد ۳۱ عدد تصویر لحظه ای با سرعت رشد زمانی مثلاً ۵ دقیقه ای از مجموعه داده ی مربوطه تهیه نماییم، ابتدا مجموعه ی یال ها را بر حسب برچسب زمانی به صورت صعودی مرتب نموده و سپس یال های موجود در بازه ی زمانی [0,850] را به عنوان تصویر لحظه ای اول، یال های موجود در بازه ی زمانی [5,855] را به عنوان تصویر لحظه ای دوم و الی آخر در نظر می گیریم، به طوری که تصویر لحظه ای آخر شامل یال های موجود در بازه ی زمانی [150,1000] خواهد بود. حال این تصاویر لحظه ای نقش گراف های ما در مراحل زمانی مختلف را بازی خواهند کرد. در نهایت برای هر مجموعه داده، به ازای سرعت های رشد زمانی متفاوت، یک سری مجموعه داده شامل ۳۱ عدد تصویر لحظه ای از مجموعه داده ی اصلی تهیه می نماییم و به ازای هر سری، نتایج اجرای الگوریتم پیشنهادی را بر روی اعضای سری نمایش خواهیم داد.

۴) نتایج مربوط به پیاده سازی

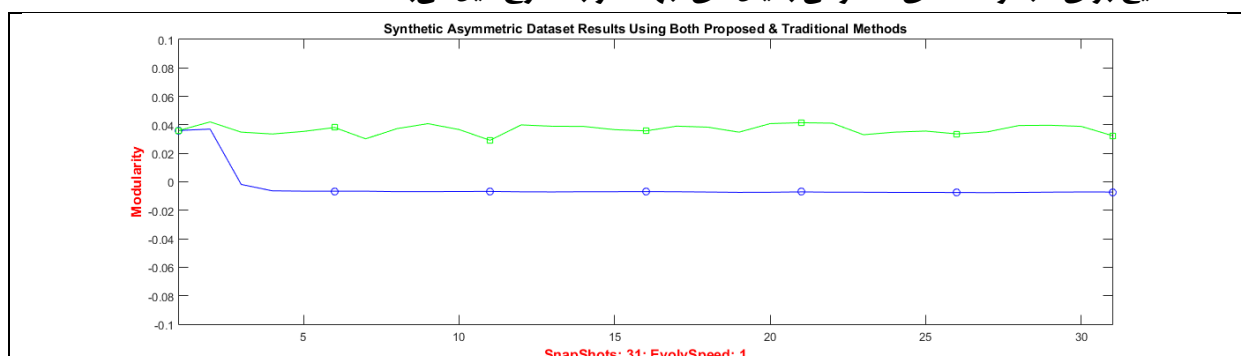
در این جا باید گفت که با توجه به بسیار حجیم بودن مجموعه داده های واقعی و در نتیجه طولانی بودن رویه ی اجرای الگوریتم پیشنهادی، قادر به کسب نتایج نهائی نگشتیم. اما در مورد مجموعه داده های مصنوعی، نتایج نهائی مشابه نتایج قید شده در مقاله برای مجموعه داده های واقعی می باشد، به این معنی که با افزایش سرعت رشد بازه های زمانی، میزان پیمانی کاهش چشم گیری دارد. نتایج مربوطه برای دو مجموعه داده ی مصنوعی که یکی دارای یال های جهت دار و دیگری بی جهت است، به شرح ذیل می باشند:

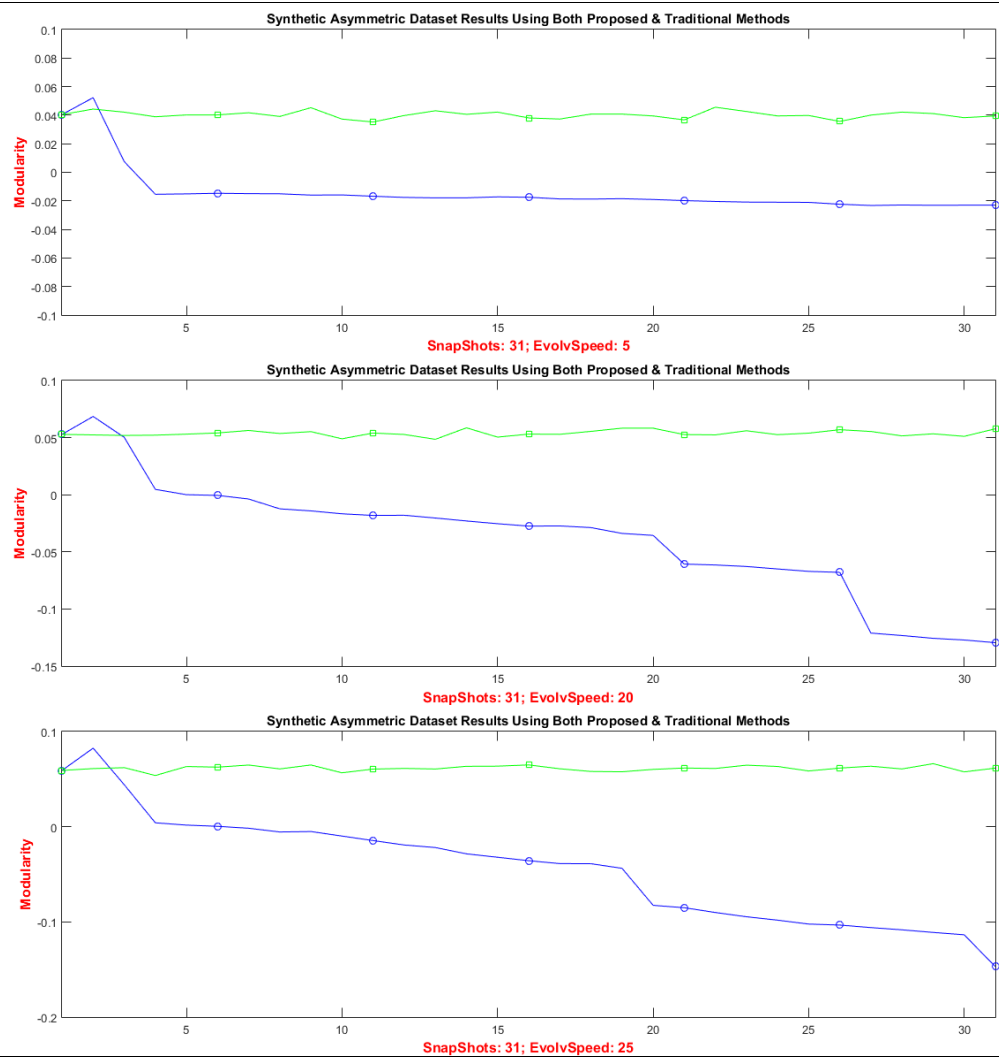




همانطور که مشاهده می‌شود، به ازای سرعت رشد اندک، شاهد افزایش ملایم میزان پیمانگی هستیم، ولی با افزایش سرعت رشد از ۱ دقیقه تا ۲۵ دقیقه، می‌بینیم که میزان پیمانگی فوراً رو به کاهش می‌گذارد، به طوری که برای سرعت رشد ۲۵ دقیقه‌ای و بعد از تصویر لحظه‌ای ۲۱، میزان پیمانگی برای روش پیشنهادی حتی از روش سنتی نیز کمتر شده است.

نتایج برای مجموعه‌داده‌ی مصنوعی با یال‌های جهت‌دار به شرح ذیل می‌باشد:





شکل ۲. نتایج اجرای الگوریتم پیشنهادی بر روی مجموعه داده‌ی مصنوعی با یال‌های جهت‌دار؛ خطوط آبی و سبز به ترتیب نمایانگر میزان پیمانی برای روش پیشنهادی و روش سنتی می‌باشند.

متأسفانه برای این مجموعه‌داده، نتایج بر طبق انتظار نشده و در تمامی موارد، الگوریتم پیشنهادی نسبت به روش سنتی بدتر عمل نموده و علت آن نیز در دست بررسی است!!!؟

- [1] He, Jialin, and Duanbing Chen. "A fast algorithm for community detection in temporal network." *Physica A: Statistical Mechanics and its Applications* 429 (2015): 87-94.
- [2] Blondel, Vincent D., et al. "Fast unfolding of communities in large networks." *Journal of statistical mechanics: theory and experiment* 2008.10 (2008): P10008.