نشریه علمی"پدافند الکترونیکی و سایبری"
سال هفتم، شماره ۳، پاییز ۱۳۹۸، ص ۱۱۱-۱۰۵

ساخت تجزیه درختی گراف‌ها با استفاده از الگوریتم رقابت استعماری جهت استفاده در تسهیم راز

میثم رجعتی باویل علیایی[۱]، محمدرضا هوشمند اصل[۲*]

۱- دانشجوی دکتری، ۲- دانشیار، گروه علوم کامپیوتر، آزمایشگاه محاسبات کوانتومی و رمزنگاری، دانشگاه یزد، ایران

(دریافت: ۹۷/۰۹/۲۵، پذیرش: ۹۷/۱۲/۱۴)

چکیده

تسهیم راز، یعنی به اشتراک گذاشتن داده محرمانه میان تعدادی شرکت‌کننده، به‌طوری‌که زیرمجموعه‌های مشخصی (مجاز) از آنها قادر به بازیابی آن داده، باشند ولی زیرمجموعه‌های غیرمجاز قادر به بازیابی اطلاعات مرتبط با آن نباشند. روش‌های متعدد برای تسهیم راز ارائه شده است. از جمله این روش‌ها، تسهیم راز مبتنی‌بر مجموعه احاطه‌گر و احاطه‌گر یالی است. در روش مبتنی بر احاطه‌گر یالی، نیاز است که تمام مجموعه‌های احاطه‌گر یالی برای گراف به‌دست آید. یافتن تمام مجموعه‌های احاطه‌گر یالی برای گراف یک مسئله NP-کامل است. به‌سادگی می‌توان تمام مجموعه‌های احاطه‌گر یالی یک گراف داده شده را با استفاده از تجزیه درختی گراف آن و الگوریتم برنامه‌نویسی پویا به‌دست آورد. ساخت تجزیه درختی یک گراف با عرض درختی محدود، از زمان چندجمله‌ای است. اما در حالت کلی محاسبه عرض درختی و ساختن تجزیه درختی با حداقل عرض، یک مسئله NP-کامل است. هدف ما در این مقاله، استفاده از الگوریتم رقابت استعماری برای ساخت تجزیه درختی گراف‌ها است که می‌تواند به‌صورت موازی پیاده‌سازی شود. بنابراین، روش پیشنهادی علاوه‌بر این که روش نوینی برای پیاده‌سازی طرح تسهیم راز است، می‌تواند زمان اجرارا در حالت موازی تا ۵٪ کاهش دهد.

کلیدواژه‌ها: تسهیم راز، مجموعه احاطه‌گر یالی، تجزیه درختی و الگوریتم رقابت استعماری

۱- مقدمه

دانش رمزنگاری با امنیت اطلاعات در ارتباط است. با توجه به استفاده گسترده از اینترنت، نیاز به امنیت اطلاعات نیز افزایش می‌یابد. لذا، به‌کارگیری روش‌های جدیدبرای ابداع روش‌های مختلف در حوزه رمزنگاری جهت محافظت از اطلاعات خصوصی افراد یا سازمان‌ها ضروری می‌باشد. ازجمله این روش‌ها، استفاده از مفهوم مجموعه احاطه‌گر و تعمیم‌های آن در حوزه‌های مختلف ازجمله، رأی‌گیری الکترونیکی مبتنی‌بر تسهیم راز [۱]، شبکه‌های بی‌سیم [۲] و طرح تسهیم چندراز[۳] است.

مجموعه احاطه‌گر برای گراف یک زیرمجموعه از رأس‌ها است، به‌طوری‌که هر گره یا درون این مجموعه است یا همسایه‌ای در این مجموعه دارد. به‌دلیل ویژگی‌ها و تنوع کاربردهای احاطه‌گرها، تعمیم‌های مختلفی از آنها برای مدل‌سازی پدیده‌های مختلف و بررسی مسائل مرتبط با آن مورد پژوهش قرار گرفته است. به‌عنوان مثال می‌توان به‌مجموعه احاطه‌گر مخلوط [۴]، مجموعه احاطه‌گر خارجی [۵] اشاره نمود. مجموعه احاطه‌گر برای گراف در زمینه مسیریابی و رایانش‌های توزیع‌شده یا خوشه‌ای کاربرد دارد. به‌طورمشابه، مفهوم مجموعه احاطه‌گر یالی برای گراف G که عبارت است از یک زیرمجموعه از یال‌ها به‌طوری‌که هر یال یا درون آن مجموعه قرار دارد یا همجوار با عضوی از آن مجموعه است.

مسئله یافتن مجموعه احاطه‌گریالی نیز کاربردهای زیادی در حوزه رمزنگاری از جمله روش تسهیم راز دارد. تسهیم راز عبارت است از به اشتراک گذاشتن یک راز میان مجموعه‌ای از شرکاء با اختصاص مقداری محرمانه به نام سهم به هر یک از آنها، به‌گونه‌ای که زیرمجموعه‌ای مجاز از شرکاء با استفاده از سهم‌های خود بتوانند راز را بازیابی نمایند. الگوریتم‌های با رویکردهای مختلف برای تسهیم راز ارائه شده است، نمونه‌ای از این‌گونه الگوریتم‌ها شامل تسهیم راز برمبنای خم بیضوی، مبتنی‌بر ضرب ماتریس‌ها، بصری، درون‌یابی لاگرانژ و مبتنی بر ماتریس تصویر هستند. آل- سیدی و همکارانش [۱] روشی مبتنی‌بر مجموعه احاطه‌گر یالی برای تسهیم راز ارائه دادند که این روش و روش مبتنی‌بر مجموعه احاطه‌گر رأسی روش‌های تسهیم‌راز مبتنی‌بر نظریه گراف هستند. مجموعه احاطه‌گریالی با کمترین تعداد یال به‌عنوان ساختار دسترسی کمینه برای بازیابی کلید مخفی

* رایانامه نویسنده مسئول: hooshmandasl@yazd.ac.ir

استفاده می‌شود. طرح تسهیم راز پیشنهاد شده شامل سه مرحله است. در مرحله اول به تنظیم مقادیر اولیه پرداخته شده است. در مرحله دوم تجزیه رمز طراحی شده است. در این طراحی تمام مجموعه احاطه‌گریالی گراف تولیدشده و از آنها برای تولید رمز استفاده می‌شود. در مرحله نهایی بازسازی رمز به‌نحوی صورت می‌پذیرد که عناصر انتخاب شده گراف را پوشش بدهند.

برای به‌دست آوردن مجموعه احاطه‌گریالی، می‌توان از تجزیه درختی گراف داده‌شده و الگوریتم برنامه‌نویسی پویا استفاده نماییم. اگرچه برای ساخت تجزیه درختی یک گراف زمانی که عرض درختی آن محدود باشد، الگوریتم زمان چندجمله‌ای وجود دارد [۶]. امّا درحالت کلی محاسبه عرض درختی و ساختن تجزیه درختی با حداقل عرض، یک مسئله NP-کامل است. لذا در این مقاله، روشی جهت محاسبه تجزیه درختی گراف با الگوریتم ابتکاری ارائه خواهد شد.

ایده عرض درخت در سال ۱۹۸۴ توسط رابرتسون و سیمور ارائه شده است. آنها ثابت کردند که این مفهوم معیار اندازه‌گیری خوبی از سختی ذاتی یک مسئله NP-کامل در گراف است و شباهت گراف به یک درخت را نشان می‌دهد [۷]. تجزیه درختی با عرض درختی محدود از دیدگاه نظری جالب است، زیرا تمام مسائل NP-کامل که می‌توانند به‌صورت منطق مرتبه دوم بیان شوند، می‌توانند به لحاظ نظری در زمان خطی حل شوند [۸]. علاوه‌بر این، اگر بتوان یک مسئله را به یک مسئله دیگر کاهش داد به‌طوری که مسئله دوم را بتوان به‌صورت منطق مرتبه دوم نوشت در نتیجه الگوریتمی وجود دارد که این ویژگی را در زمان چندجمله‌ای برای گراف‌هایی با عرض درختی ثابت بررسی کند [۹].

تحقیقات نشان داده است برای حل مسئله‌هایNP-کامل روی گراف‌هایی با عرض درختی ثابت، الگوریتم زمان چندجمله‌ای وجود دارد. مهمترین روشی که برای حل این مسئله‌ها به‌کار می‌رود، برنامه‌نویسی پویا می‌باشد. در این روش نیاز است ابتدا یک تجزیه درختی برای گراف ورودی ایجاد شود و سپس برای هر یک از گره‌های تجزیه درختی یک جواب جزئی به‌دست می‌آید و با پیمایش پایین به بالای تجزیه درختی، جواب نهایی در ریشه یافت خواهد شد [۱۰].

فومین و همکارانش [۱۱] در سال ۲۰۰۴ یک الگوریتم دقیق برای پیدا کردن عرض درختی گراف در حالت کلی در زمان $O^*(1.9601^n)$ ارائه دادند که n تعداد رأس‌ها درگراف ورودی است. اما ایراد روش ارائه داده شده این بود که نیاز به فضای نمایی داشت امّا درسال ۲۰۰۸، آنها توانستند در زمان $O^*(2.9512^n)$ فضای ذخیره‌سازی را به چندجمله‌ای کاهش دهند.

برای حالتی که عرض درختی مشخص نیست ولی یک کران بالا برای آن وجود دارد، یک الگوریتم خطی برای آن ارائه شده است. امّا پیچیدگی الگوریتم هنوز برحسب مقدار عرض درختی نمایی است و قابلیت این الگوریتم در استفاده آن برای مقادیر بسیار کوچک عرض درختی است. از این‌رو، تلاش برای یافتن الگوریتم‌های کارآمد برای پارامتر ثابت در حالی‌که از نظر پیاده‌سازی نیز عملی باشد، باقی مانده است.

اگر چه ساخت یک تجزیه درختی بهینه در زمان معقول معمولاً غیرممکن است، امّا الگوریتم‌های زیادی وجود دارند که عرض درخت را به‌صورت تقریبی برمی‌گردانند. به‌عنوان مثال بوتلندر و همکارانش درسال ۲۰۱۶ یک الگوریتم تقریبی با فاکتور ۵ ارائه دادند[۱۲].

روش‌های اکتشافی حد بالای خوبی را ارائه می‌دهند. این روش‌ها اغلب خیلی سریع‌تر از الگوریتم‌های تقریبی هستند. روش‌های اکتشافی بسیاری برای یافتن حد بالای عرض درختی گراف وجود دارد که اغلب آنها از ایده مثلثی‌سازی کردن گراف بهره می‌برند اما شیوه پیاده‌سازی آنها متفاوت است. الگوریتم‌های مبتنی‌بر الگوریتم ژنتیک، الگوریتم تبرید شبیه‌سازی شده، الگوریتم جستجوی ممنوعه، جستجوی محلی و الگوریتم بهینه‌سازی مورچه نمونه‌ای ازالگوریتم‌های فرااکتشافی در این زمینه هستند [۱۳].

در این مقاله از الگوریتم رقابت استعماری بهره می‌بریم تا تجزیه درختی گراف‌هایی که عرض درختی آنها نامشخص هستند را به‌دست آوریم.

در ادامه در بخش ۲، مفاهیم اولیه نظیر تعریف تجزیه درختی و ایده ترتیب حذف شرح داده خواهد شد. سپس در بخش ۳، الگوریتم پیشنهادی را بیان خواهیم نمود و در بخش ۴، نتایج عملی آن را نشان خواهیم داد.

۲- مفاهیم اولیه

در این بخش، مفاهیم اولیه نظیر تعریف تجزیه درختی و ایده ترتیب حذف شرح داده خواهد شد.

۲-۱- عرض درختی

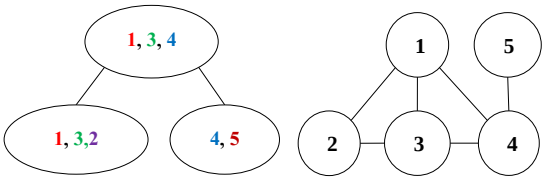
گراف $G = (V, E)$ را در نظر بگیرید. یک تجزیه درختی گراف G زوج مرتب $(\{X_i \mid i \in I\}, T)$ است به‌طوری که هر رأس X_i یک زیر مجموعه‌ای از V است و آن را بسته نامند و T یک درخت با مجموعه I به‌عنوان مجموعه رئوس درخت T می‌باشد، خواص زیرباید برقرار باشد [۱۱]:

۱. $\bigcup_{i \in I} X_i = V$،

۲. برای هر یال $\{u,v\} \in E$ یک $i \in I$ وجود دارد به‌طوری‌که، $u, v \in X_i$،

۳. برای هر $v \in V$ مجموعه $I_v = \{i \in I \mid v \in X_i\}$ یک زیردرخت همبند از T است.

عرض تجزیه درختی برابر $\max\{|X_i| \mid i \in I\} - 1$ است و عرض درختی گراف G برابر کمترین مقدار عرض درختی در میان تمام تجزیه‌های درختی گراف است. شکل (۱) گراف $G1$ و یکی از تجزیه‌های درختی متناظر با آن که عرض آن برابر دو می‌باشد را نشان می‌دهد. هر رأس و هر یال حداقل در یکی از بسته‌های تجزیه درختی ظاهر شده است. همچنین، هر رأسی از گراف را که در نظر بگیریم، بسته‌هایی که این رأس در آنها ظاهر شده است، تشکیل یک زیردرخت می‌دهند.



ب) تجزیه درختی‌گراف $G1$ آ) گراف $G1$

شکل (۱): گراف $G1$ و یکی از تجزیه‌های درختی متناظر با آن.

۲-۲- ایده ترتیب حذف رأس‌ها جهت محاسبه عرض درختی گراف درختی

یک ترتیب حذف برای گراف G یک تابع یک‌به‌یک به‌صورت $g: V \to \{1, 2, \dots, n\}$ است. ترتیب حذف g کامل نامیده می‌شود، هرگاه برای هر $v \in V$ مجموعه تمام همسایه‌های رأس v که در این ترتیب بزرگتر از v هستند، $\{u | u \in N_g(v) \wedge N_g(u) > N_g(v)\}$، تشکیل یک خوشه بدهد. قضیه زیر ارتباط بین مفهوم ترتیب حذف کامل را با عرض درختی نشان می‌دهد.

گراف وتری گرافی است که هر دور به طول چهار یا بیشتر از آن شامل وتر باشد. گراف وتری را گراف مثلثی‌سازی شده گویند. هرگاه در روند حذف رأس‌ها در ترتیب حذف، تمام همسایه‌های رأسی که قرار است حذف شود را به یکدیگر متصل نموده و سپس رأس را حذف نمایند و این روند را ادامه دهند گراف مثلثی‌سازی شده حاصل می‌گردد.

قضیه ۱ ([۱۴]): گراف G مفروض است،

۱. گراف G مثلثی‌سازی شده است اگر و تنها اگر G دارای ترتیب حذف کامل باشد.

۲. گراف G دارای ترتیب حذف کامل است اگر و تنها اگر یک تجزیه درختی برای گراف وجود داشته باشد به‌طوری‌که هر یک از بسته‌های گراف یک خوشه تشکیل دهند.

نتیجه: با توجه به قضیه ۱ عرض یک تجزیه درختی برابر است با بزرگترین خوشه گراف مثلثی‌سازی شده منهای یک، در نتیجه عرض درختی یک گراف برابر است با یافتن گراف مثلثی‌سازی شده‌ای که اندازه بزرگترین خوشه آن، کمترین اندازه را در میان تمام گراف‌های مثلثی‌سازی شده گراف داشته باشد.

۳- الگوریتم پیشنهادی

الگوریتم پیشنهادی جهت تسهیم راز مبتنی‌بر احاطه‌گر یالی با استفاده از الگوریتم رقابت استعماری به شرح زیر می‌باشد. ابتدا با استفاده از الگوریتم رقابت استعماری یک تجزیه درختی برای گراف ورودی ایجاد می‌شود. در مرحله دوم، با استفاده از الگوریتم تولید مجموعه احاطه‌گر مخلوط در [۴] و با در نظر گرفتن فقط پوشش یالی در مراحل اجرای الگوریتم،مجموعه احاطه‌گریالی متناظر با تجزیه درختی داده شده در زمان چندجمله‌ای محاسبه می‌کنیم. با استفاده از مجموعه احاطه‌گر به‌دست‌آمده و الگوریتم تسهیم راز آل_ سیدی [۱]، یک ساختار تسهیم راز در زمان چندجمله‌ای حاصل می‌شود. لذا، زمان پیاده‌سازی این ساختار تسهیم راز وابسته به زمان اجرای الگوریتم رقابت استعماری در مرحله اول است. این مراحل در الگوریتم ۱ مشخص شده است.

الگوریتم تسهیم راز مبتنی بر احاطه‌گر یالی با استفاده از الگوریتم رقابت استعماری

ورودی: گراف

۱. تولید مقدار عرض درختی گراف داده شده با استفاده از الگوریتم رقابت استعماری که در بخش ۳-۲ شرح داده خواهد شد. در این الگوریتم برای محاسبه عرض درختی از نتیجه قضیه ۱ استفاده شده است.

۲. تولید تجزیه درختی گراف مفروض با استفاده از الگوریتم تولید درخت تجزیه در [۶].

۳. با استفاده از درخت تجزیه تولید شده در مرحله ۲ و به‌کارگیری الگوریتم احاطه‌گر مخلوط محدود شده به یال در [۴]، مجموعه احاطه‌گر یالی حاصل می‌شود.

۴. از مجموعه احاطه‌گر یالی تولید شده در مرحله ۳ برای تولید ساختار تسهیم راز با استفاده از الگوریتم تسهیم راز آل- سیدی [۱]، استفاده می‌شود.

الگوریتم ۱: الگوریتم تسهیم راز مبتنی‌بر احاطه‌گر یالی با استفاده از الگوریتم رقابت استعماری

۳-۱- امنیت و مزیت طرح

امنیت این ساختار مبتنی بر امنیت تسهیم راز آل- سیدی است که بر اساس نرخ اطلاعات تعیین می‌گردد. در طرح آل- سیدی این نرخ اطلاعات نسبت به طرح‌های مشابه مانند طرح تسهیم راز مبتنی‌بر مجموعه احاطه‌گر رأسی [۱۵] و ساختار تجزیه‌ای طرح تسهیم راز استین‌سون [۱۶] دارای کمترین کران است. علاوه‌بر این، در الگوریتم پیشنهادی در مرحله سوم می‌توانیم تمام مجموعه‌های احاطه‌گر بهینه با کمترین تعداد یال را شمارش نماییم، به‌طوری‌که هر کدام از این مجموعه‌های به‌دست‌آمده یک سیستم تسهیم راز را ارائه خواهد داد. لذا با استفاده از این مزیت، می‌توانیم سیستم تسهیم راز چندگانه بر مبنای احاطه‌گر یالی را پیاده‌سازی نماییم. این شیوه و بحث امنیت آن به‌عنوان یک رهیافت جدید برای پژوهش‌های آتی پیشنهاد می‌گردد.

۳-۲- الگوریتم رقابت استعماری

الگوریتم‌های بهینه‌سازی فرامکاشفه‌ای یک نمونه از الگوریتم‌های تقریبی می‌باشند که در دامنه وسیعی از مسئله‌های موجود در رشته‌های مختلف محبوبیت پیدا کرده‌اند، زیرا آنها نیازی به اطلاعات اضافی ندارند، همچنین ساده و آسان پیاده‌سازی می‌شوند و می‌توانند از بهینه محلی دور بمانند. روش‌های فرامکاشفه‌ای از رفتار انسان و یا طبیعت الهام می‌گیرند. رویکردهایی مانند الگوریتم رقابت استعماری از رفتار انسان الهام می‌گیرند. الگوریتم رقابت استعماری در سال ۲۰۰۷ توسط آتشپز و همکارانش ارائه شد [۱۷]. این الگوریتم از رقابت بین امپراتوری‌ها برایتصاحب مستعمره‌ها الهام گرفته شده است. این روش از سه مرحله تشکیل شده است. ما قصد داریم با متناسب کردن الگوریتم رقابت استعماری برای محیط‌های گسسته، از آن برای تولید تجزیه درختی استفاده کنیم.

۳-۲-۱- ایجاد کشورها و تشکیل امپراتوری‌ها

مشابه سایر الگوریتم‌های فرامکاشفه‌ای مبتنی‌بر جمعیت، الگوریتم رقابت استعماری نیز با یک جمعیت تصادفی شروع می‌شود که شامل N جمعیت اولیه است. برای نشان‌دادن یک فرمول مناسب برای نمونه اولیه، یک آرایه $1 \times n$ است که کشور نام دارد و آن کشور را به‌صورت آرایه $country = [p_1, p_2, \dots, p_n]$ معرفی شده است. هر کشور یک جایگشت از اعداد یک تا n است که در آن n تعداد رأس‌ها در گراف در نظر گرفته شده را نشان می‌دهد. به‌عنوان مثال [۵، ۶، ۳، ۷، ۲،۴، ۱] یک کشور برای گرافی با هفت رأس است که معرف یک تجزیه درختی برای گراف است.

یک کشور در واقع یک تجزیه درختی برای گراف است، از این‌رو، تابع هزینه کشور را عرض تجزیه درختی معرفی می‌کنیم. هزینه یک کشور کران بالای عرض درخت را نشان می‌دهد و با ارزیابی تابع هدف f تعیین می‌شود. هزینه هر کشور با استفاده از تابع هزینه $Cost = f(country)$ محاسبه می‌شود. ورودی تابع هزینه، یک کشور (تجزیه درختی) است و خروجی آن عرض تجزیه درختی مربوط به آن کشور است.

سپس، ما یک جمعیت اولیه با اندازه N_{pop} که شامل امپراتورها و مستعمره‌ها است را تولید می‌کنیم. کشورهای قدرتمند به تعداد N_{imp} با کمترین هزینه امپراتور نامیده می‌شوند و کشورهای باقی مانده با اندازه $N_{col} = N_{pop} - N_{imp}$ به‌عنوان مستعمره‌ها در نظر گرفته می‌شوند.

تشکیل امپراتوری‌های اولیه به‌وسیله تخصیص مستعمره‌ها به امپراتورها با توجه به قدرت امپراتورها آغاز می‌شود. تعداد مستعمره‌هایی که امپراتورها به‌دست می‌آورند، متناسب است با قدرت آنها. هزینه عادی هر یک از امپراتورها توسط $NC_n = \max_i(Cost_i) - Cost_n$ تعیین می‌گردد. قدرت نرمال شده هر امپراتور به‌صورت $Pow_j = \frac{NC_j}{\sum_i NC_i}$ محاسبه می‌شود. تعداد مستعمره‌های امپراتوری توسط $N_p = \lfloor Pow_j \times N_{col} \rfloor$ تعیین می‌شود. مستعمره‌ها به‌طور تصادفی در امپراتورها برای ایجاد امپراتوری‌ها توزیع می‌شوند.

۳-۲-۲- تکامل مستعمره‌ها

تکامل مستعمره‌ها شامل سه فرآیند جذب، انقلاب و جایگزین موقعیت امپراتور با بهترین مستعمره آن است.

در فرآیند جذب، مستعمره‌ها به سمت امپراتوری‌های مربوطه حرکت خواهند کرد. حرکت به سمت امپراتوری در واقع مرحله تکامل است. روند جذب را به‌صورت زیر تعریف می‌کنیم. یک کشور با امپراتوری مربوطه مرتبط است و کشورهای جدیدی را که بیشتر شبیه امپراتوری هستند، می‌سازد. نسبت انقلاب را یک عدد تصادفی در نظر می‌گیریم، این عدد یک عدد تصادفی است که توسط توزیع تصادفی در فاصله [0,1] به‌عنوان احتمال ترکیب بین یک مستعمره و امپراتوری آن ایجاد شده است.

حال یک عملگر ترکیب جدید بین یک امپراتور و یک مستعمره معرفی می‌کنیم. فاصله بین امپراتور و مستعمره را اختلاف تعداد عنصرهای مشابه در k عنصر اول در نظر گرفته می‌شود، حتی اگر در نظم متفاوت باشند. عدد k به‌صورت تصادفی انتخاب می‌شود. این عدد بین ۱ و $n-1$ است. هدف ترکیب، تولید یک کشور جدید است که همان فاصله بین امپراتور و مستعمره را داشته باشد. ابتدا همه k عنصر اول همان مستعمره را مبادله می‌کنیم و عناصر باقی مانده را به‌صورت تصادفی انتخاب می‌کنیم. فرض کنید که $I_{k+1}I_{k+2} \dots I_n$ قسمت دوم امپراتور است و $C_{k+1}C_{k+2} \dots C_n$ بخش دوم کشور باشد،

لیست $I_{k+1}C_{k+1}I_{k+2}C_{k+2}$ ... I_nC_n را در نظر می‌گیریم و از آن برای ایجاد کشور جدید استفاده می‌کنیم. از اولین عنصر لیست شروع کرده و اگر عنصر به کشور جدید تعلق نداشته باشد، آن را به کشور اضافه می‌کنیم.

فرآیند انقلاب برای افزایش توانایی الگوریتم‌ها برای جلوگیری از همگرایی زود هنگام و فرار از بهینه محلی استفاده می‌شود. انقلاب به‌صورت زیر در نظر گرفته می‌شود. در هر یک از امپراتوری‌ها، بعضی از مستعمره‌ها به‌صورت تصادفی بر اساس نرخ انقلاب انتخاب می‌شوند و سپس با امپراتورهای مربوطه ترکیب شده و کشورهای جدید تولید می‌شوند. فرآیند تولید یک کشور جدید با انتخاب چندین موقعیت در عناصر مستعمره و با یک سکه برای هر موقعیت، به‌طور تصادفی انجام می‌شود. عناصر مستعمره در این موقعیت‌ها به کشور جدید اضافه می‌شوند. عناصر باقیمانده برحسب ترتیب عناصر امپراتور وارد می‌شوند.

پس از آن‌که عملیات ترکیب و انقلاب در مستعمره‌های امپراتوری انجام شد، توابع هزینه‌های مستعمره‌ها با عملکرد هزینه امپراتور مقایسه می‌شوند. اگر مستعمره بهتر از امپراتور خود باشد، نقش امپراتور را با آن مستعمره عوض می‌شود.

۳-۲-۳- تکامل مستعمره‌ها

هسته الگوریتم رقابت استعماری، رقابت بین امپراتوری‌ها است. در این مرحله، ابتدا، هزینه کل هر امپراتوری محاسبه می‌شود. قدرت امپراتوری بر اساس قدرت امپراتوری و کسری از قدرت مستعمرات آن محاسبه می‌شود تا امپراتوری‌ها را براساس کمترین هزینه‌های آنها رتبه‌بندی کنیم. هزینه کل هر امپراتوری با معادله (۱) محاسبه می‌شود.

$$T.C._n = Cost(imperialist_n) + \xi.mean(Cost(colonies\ of\ empire_n)) \quad (۱)$$

$T.C._n$ هزینه کل امپراتوری nام و ξ یک عدد مثبت کمتر از یک است و معمولاً نزدیک به صفر است. توجه می‌شود که قدرت امپراتوری عمدتاً تحت تاثیر قدرت امپراتوری آن قرار می‌گیرد و قدرت مستعمره‌ها بسیار کم و تقریباً ناچیز است. بنابراین، مقدار کم ξ تأکید بیشتری بر نفوذ قدرت امپراتور در تعیین قدرت کل امپراتوری دارد، درحالی که مقدار زیاد ξ نشان‌دهنده تأثیر قدرت متوسط مستعمره‌ها در تعیین قدرت کل امپراتوری است.

درنهایت، امپراتوری آماده است تا در جریان اصلی رقابت امپراتوری‌ها شرکت کند. در طول رقابت بین کشورهای امپراتوری، امپراتوری‌های ضعیف به‌تدریج فرو می‌پاشند. این به این معنی است که امپراتوری‌ها در رقابت برای کنترل یک یا چندتا از ضعیف‌ترین مستعمره‌ها که متعلق به ضعیف‌ترین امپراتوری‌ها هستند، برای افزایش قدرت خود رقابت می‌کنند. امپراتوری‌های قوی‌تر شانس بیشتری برای داشتن ضعیف‌ترین مستعمره‌ها دارند. برای انجام این فرآیند، باید با توجه به معادله (۲) هزینه کلی نرمال شده هر امپراتوری ارزیابی شود.

$$N.T.C._j = T.C._j - \max_i\{T.C._i\} \quad (۲)$$

موفقیت هر امپراتوری توسط P_{E_j} محاسبه می‌شود.

$$P_{E_j} = \frac{N.T.C._j}{\sum_{i=1}^{N_{imp}} N.T.C._i} \quad (۳)$$

به‌طوری‌که: $\sum_{i=1}^{N_{imp}} P_i = 1$.

یک سازوکار شبیه چرخ رولت در ژنتیک برای توزیع ضعیف‌ترین مستعمره‌ها در امپراتوری‌ها بر اساس احتمال مالکیت آنها استفاده می‌شود. با این تفاوت که الگوریتم رقابت استعماری یک سازوکار توزیع جدید را معرفی می‌کند که نیاز به محاسبات کمتر دارد. این سازوکار تنها نیاز به یک تابع چگالی احتمال دارد که باعث می‌شود سریع‌تر از روش معمول چرخ رولت در ژنتیک کار کند.

با تکرار این روند، پس از رقابت امپراتوری‌ها، امپراتوری‌های ضعیف‌تر مستعمرات خود را از دست می‌دهند و امپراتوری‌های قدرتمند مستعمرات بیشتری را به‌دست می‌آورند. بنابراین، قدرت بیشتری را به‌دست می‌آورند و امپراتوری‌های ضعیف را با گذشت زمان از بین می‌برند.

الگوریتم رقابت استعماری همچون سایر الگوریتم‌های تکاملی از یک روند تکراری پیروی می‌کند تا زمانی‌که معیارهای ماندگاری مانند تعداد دقیق تکرار یا زمان اجرای از پیش تعریف شده به پایان برسد. معیار توقف ایده آل این است که تنها یک امپراتوری باقی بماند.

۴- ارزیابی

برای شبیه‌سازی با استفاده از نرم‌افزار متلب پارامترهای زیر در-نظر گرفته شده است.

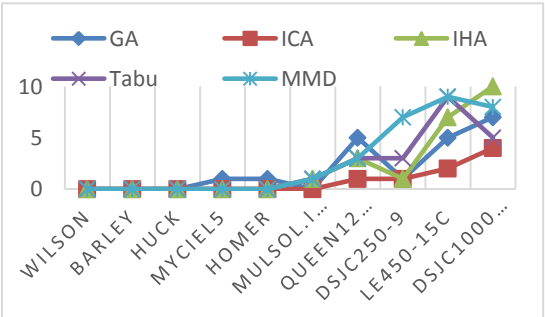
جدول (۱): پارامترهای الگوریتم

اسم پارامتر	مرحله	اندازه پارامتر
N_{pop}	تشکیل امپراتوری‌ها	$2 \times \lvert V \rvert$
N_{imp}	تشکیل امپراتوری‌ها	$\%10 \times N_{pop}$
Assimilationcoefficient	تکامل	۳
Revolution rate	انقلاب	۰/۳
Damp ratio	انقلاب	۰/۹۵
ξ	رقابت	۰/۱
Competition pressure	رقابت	۰/۱۰

نتایج حاصل با اجرای الگوریتم پیشنهادی روی رایانه‌ای با مشخصات پردازنده i7, 1.73GHz و حافظه رم ۶ gb نتایج جدول (۲) با صدبار تکرار حاصل شده است. در جدول (۲)، بهترین جواب الگوریتم پیشنهادی و میانگین جواب‌های حاصل از صدبار تکرار، برای برخی گراف‌های معروف نمایش داده شده است. برای مقایسه روش پیشنهادی (ICA)، با سایر روش‌ها، اختلاف خروجی الگوریتم‌های GA، IHA، Tabu و MMD با عرض درختی گراف‌های جدول (۲)، در شکل (۲) نشان داده شده است.

جدول (۲): نتایج عملی

	گراف	رأس	یال	عرض درختی	بهترین	میانگین
۱	Wilson	۲۱	۲۷	۳	۳	۳
۲	Barley	۴۸	۱۲۶	۷	۷	۷
۳	Huck	۷۴	۶۰۲	۱۰	۱۰	۱۰
۴	Myciel5	۴۷	۲۳۶	۱۹	۱۹	۱۹
۵	Homer	۵۶۱	۳۲۵۸	۳۱	۳۱	۳۱
۶	Mulsol.i.1	۱۳۸	۳۹۲۵	۵۰	۵۰	۵۰
۷	Queen 12-12	۱۴۴	۲۵۹۶	۱۰۳	۱۰۳	۱۰۳٫۶
۸	DSJC 250-9	۲۵۰	۲۷۸۹۷	۲۴۳	۲۴۳	۲۴۴٫۳
۹	LE 450-15C	۴۵۰	۱۶۶۸۰	۳۵۰	۳۵۱	۳۵۱٫۳
۱۰	DSJC 1000.9	۱۰۰۰	۴۴۹۴۴۹	۹۹۱	۹۹۵	۹۹۶٫۳



شکل (۲): اختلاف خروجی الگوریتم‌های GA، ICA، IHA، Tabu و MMD با عرض درختی برای گراف‌های جدول (۲).

۵- نتیجه‌گیری

یکی از روش‌های ساخت تسهیم راز استفاده از مجموعه احاطه‌گر یا احاطه‌گر یالی است. در این شیوه، پیش فرض ساخت تسهیم راز مبتنی بر مجموعه احاطه‌گر بهینه داده شده است امّا به‌دست آوردن مجموعه احاطه‌گر، یک مسئله NP-کامل است. لذا روش‌های متعددی برای ساخت این نوع مجموعه‌ها متناسب با ساختار گراف پیشنهاد شده است. یکی از شیوه‌های اساسی برای ساخت مجموعه احاطه‌گر بهینه، استفاده از عرض درختی گراف مفروض است. امّا به‌طور کلی محاسبه عرض درختی گراف در زمان چندجمله‌ای امکان‌پذیر نمی‌باشد. از این‌رو، در این پژوهش با استفاده از الگوریتم رقابت استعماری روشی جهت به‌دست آوردن عرض درختی گراف مفروض ارائه گردیده است که یک پیش‌پردازش جهت ساخت سیستم تسهیم راز است. به‌عبارتی با داشتن عرض درختی گراف می‌توان در زمان چندجمله‌ای درخت تجزیه گراف را تولید نموده و سپس با استفاده از برنامه نویسی پویا در زمان چندجمله‌ای یک مجموعه احاطه‌گر یالی بهینه با کمترین مقدار عدد احاطه‌گر را تولید نماییم و در آخر با استفاده از مجموعه احاطه‌گر یالی تولید شده و الگوریتم آل‌سیدییک ساختار تسهیم راز متناظر با مجموعه داده شده ارائه می‌گردد. مزیت روش پیشنهادی در این است که می‌تواند به‌صورت موازی پیاده‌سازی شود. بنابراین، علاوه‌بر این‌که روش پیشنهادی، روش نوینی برای پیاده‌سازی طرح تسهیم راز است، زمان اجرا را نیز کاهش داده ودر کمترین زمان قادر به یافتن جواب بهینه است.

۶- منابع

[1] N. M. G. Al-Saidi and MM. Abdulhadi, "E--Voting System based on Secret Sharing Scheme," Engineering and Technology, vol. 35(1), pp. 13-18, 2017.

[2] C. Jing, H. Kun, D. Ruiying, Z. Minghui, X. Yang, and Y. Quan, "Dominating set and network coding-based routing in wireless mesh networks," IEEE Transactions on Parallel and Distributed Systems, vol. 26(2), pp. 423-433, 2015.

[3] A. R. Mirghadri and F. S. Sangtajan, "An Efficient Visual Multi-Secret Sharing Scheme," Electronical & Cyber Defence, vol. 3(4), pp. 1-9, 2016.(In Persian)

[4] M. Rajaati, M. R. Hooshmandasl, M. Dinneen, and A. Shakiba, "On fixed-parameter tractability of the mixed domination problem for graphs with bounded tree-width," Discrete Mathematics & Theoretical Computer Science, vol. 20(2), pp. 1-25, 2018.

[5] M. Hashemipour, M. R. Hooshmandasl, and A. Shakiba, "On outer-connected domination for graph products," arXiv preprint arXiv: 1708.00188, 2017.

[6] H. L. Bodlaender, "A linear-time algorithm for finding tree-decompositions of small treewidth," SIAM Journal on computing, vol. 25(6), pp. 1305–1317, 1996.

[7] N. Robertson, and P. D. Seymour, "Graph minors. iii. plannar tree-width," Combinatorial Theory, Series B, vol. 36(1), pp. 49–64, 1984.

[8] B. Courcelle, "Fly-automata for checking monadic second-order properties of graphs of bounded tree-width," Electronic Notes in Discrete Mathematics, vol. 50, pp. 3–8, 2015.

[9] H. L. Bodlaender and B. V. A. Fluiter, "Reduction algorithms for graphs of small treewidth," Information and Computation, vol. 167(2), pp. 86–119, 2001.

[10] H. L. Bodlaender, "Treewidth: Algorithmic techniques and results," In International Symposium on Mathematical Foundations of Computer Science, Springer, pp. 19–36, 1997.

[11] F. Fomin, D. Kratsch, I. Todinca, and Y. Villanger, "Exact algorithms for treewidth and minimum fill-in," SIAM Journal on Computing, vol. 38(3), pp. 1058-1079, 2008.

[12] H. L. Bodlaender, P. G. Drange, M. S. Dregi, F. V. Fomin, D. Lokshtanov, and M. Pilipczuk, "A c^{kn} 5-approximation algorithm for treewidth," SIAM Journal on Computing, vol. 45(2), pp. 317–378, 2016.

[13] T. Hammerl, N. Musliu, and W. Schafhauser, "Metaheuristic algorithms and tree decomposition," Springer Handbook of Computational Intelligence, pp. 1255-1270, 2015.

[14] H. L. Bodlaender and A. M.C.A. Koster, "Treewidth computations I. Upper bounds," Information and Computation, vol. 208, pp. 259–275, 2010.

[15] N. M. G. Al-Saidi, N. A. Rajab, M. R. Md Said, and K. A. Kadhim, "Perfect Secret Sharing based on Vertex Dominating Set," Computer Mathematics, Vol. 92(9), 2015.

[16] D. R. Stinson, "Decomposition Constructions for Secret Sharing Schemes," IEEE, Transactions information theory, vol. 40(1), 1994.

[17] E. Atashpaz-Gargari and C. Lucas, "Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition" IEEE Cong. Evol. Comput, Singapore, pp. 4661–4667, 2007.

Journal of Electronical & Cyber Defence
Vol. 7, No. 3, 2019, Serial No. 27

Generating Tree Decomposition of Graphs with Imperialist Competitive Algorithm for Use in Secret Sharing Scheme

M. Rajaati Bavil Olyaei, M. R. Hooshmandasl*

*Yazd University

(Received: 30/07/2018, Accepted: 13/10/2018)

ABSTRACT

Secret sharing refers to methods of distributing a secret amongst a group of participants, each of whom is assigned with a share of the secret. The secret can be reconstructed only when a sufficient number, of possibly different types of shares are combined together. Different secret sharing methods have been presented, such as secret sharing schemes based on dominating set and edge dominating set. In edge dominating set method, it is required that all of the edge dominating sets are obtained for the graph, which is a NP-complete problem. All of the edge dominating sets can be easily obtained, using tree decomposition of the graph and dynamic programming. Although generating tree decomposition of a graph with finite treewidth can be solved in polynomial time, but it is shown to be NP-complete for general graphs. In this paper, to generate tree decompositions of general graphs, we use the notion of Imperialist Competitive Algorithm (ICA) which can be applied in parallel. Therefore, the proposed method, in addition to being a new method for implementation of the secret sharing scheme, can reduce runtime by up to five percent in parallel.

Keywords: Secret Sharing, Edge dominating set, Tree decomposition, Imperialist Competitive Algorithm (ICA)

* Corresponding Author Email: hooshmandasl@yazd.ac.ir