

حل مسئله یافتن بزرگ‌ترین کلیک به روش بهینه‌سازی اکستریمال با پارامتر τ

محدثه گریوانی^۱، مجید وفايي جهان^۲، سيد جواد سيد مهدوي چابک^۳

^۱دانشجوی کارشناسی ارشد مهندسی کامپیوتر -نرم افزار، دانشگاه آزاداسلامی -واحدمشهد،مشهد، bj.gerivani@gmail.com

^۲استادیار گروه کامپیوتر -نرم افزار، دانشگاه آزاداسلامی -واحدمشهد،مشهد، vafaeijahan@mshdiau.ac.ir

^۳استادیار گروه کامپیوتر -سخت افزار، دانشگاه آزاداسلامی -واحدمشهد،مشهد، mahdavi@mshdiau.ac.ir

چکیده - مسئله یافتن بزرگ‌ترین کلیک، از جمله مسائل NP -Complete است که به یافتن بزرگ‌ترین زیرگراف کامل در یک گراف بدون جهت اشاره دارد. در این مقاله روشی مبتنی بر الگوریتم فرا اکتشافی بهینه‌سازی اکستریمال پیشنهاد شده است که رؤس کم ارزش را با احتمال بیشتری از بزرگ‌ترین کلیک فعلی حذف می‌کند، در نتیجه رؤس با ارزش بیشتر، احتمال حضور بیشتری نیز خواهند داشت. با هر جابجایی جزئی، تغییرات زیادی در کلیک ایجاد می‌شود و مرحله به مرحله به سمت یافتن بزرگ‌ترین کلیک پیش می‌رود. نتایج حاصل از اجرای روش پیشنهادی روی گراف‌های $DIMACS$ ، پاسخ‌های مطلوبی را از نظر اندازه کلیک‌های یافت شده و سرعت همگرایی در مقایسه با سایر روش‌ها نشان می‌دهد. در نهایت، روش پیشنهادی و پاسخ‌ها، با استفاده از نمودارهای همگرایی و آزمون آماری t صحت سنجی شده‌اند. کلیدواژه-بزرگ‌ترین کلیک، بهینه‌سازی اکستریمال، کلیک، گراف.

که زیرمجموعه کلیک دیگری نیست ولی بزرگ‌ترین کلیک، یک کلیک ماکسیمال است که دارای بزرگ‌ترین اندازه است [۱]. از کاربردهای مسئله یافتن بزرگ‌ترین کلیک (MCP^*)، می‌توان به آنالیز خوشه‌ها، بازیابی اطلاعات، شبکه‌های متحرک و شبکه اجتماعی اشاره داشت [۱،۲].

اما MCP ، یکی از مسائلی است که به عنوان مسئله NP -Complete شناخته شده است [۱،۳]. بنابراین زمانی بهتر از نمایی برای حل دقیق آن وجود ندارد. اکثر الگوریتم‌های دقیق شناخته شده، بر پایه الگوریتم انشعاب و تحدید هستند. این الگوریتم، ابتدا به صورت بازگشتی اندازه کلیک را به کمک اضافه کردن گره‌ها از یک مجموعه کاندید، توسعه می‌دهد. سپس

۱- مقدمه

در علم کامپیوتر، مسئله کلیک^۱، اشاره به هر مسئله یافتن زیرگراف‌های کامل (کلیک) در یک گراف دارد؛ یعنی مجموعه‌ای از عناصر که دو به دو به هم متصل هستند. یک گراف بدون جهت $G=(V, E)$ را داریم که V مجموعه رؤس و $E \subseteq V \times V$ مجموعه‌ای از یال‌هاست؛ یک کلیک، مجموعه‌ای از رؤس $C \subseteq V$ است.

می‌بایست بین بزرگ‌ترین کلیک^۲ و کلیک ماکسیمال، تمایز قائل شد. یک کلیک ماکسیمال (مجموعه مستقل)، کلیکی است

³ Maximum Clique Problem

⁴Branch-and-bound

¹ Clique

² Maximum Clique

توسط این روش و زمان اجرا در این الگوریتم، بسیار مطلوب هستند.

با توجه به اهمیت زمان بدست آوردن جواب بهینه در چنین مسائلی، در این مقاله سعی بر این است که با استفاده از روش بهینه سازی اکسترمال^{۱۱} با پارامتر τ ، MCP را حل کنیم. الگوریتم پیشنهادی EO-MaxClique به نحوی شبیه سازی شده است که مبتنی بر ایده اصلی الگوریتم EO، با شروع از یک کلیک اولیه، در هر مرحله با استفاده از یک تابع ارزشیابی، ارزیابی محلی و سپس مرتب سازی رئوس صورت گرفته و آن گاه رئوس کم ارزش حذف می شوند و رئوس تصادفی دیگری به کلیک فعلی اضافه می شوند. این عملیات به طور مکرر تا یافتن بزرگترین کلیک ادامه می یابد. به علت استفاده از پارامتر τ ، احتمال ماندن در بهینه های محلی رفع می شود. ویژگی مهم این روش، همگرایی سریع به سمت پاسخ است که دلیل آن، بهبود تنها یک نمونه کلیک در تمام مراحل اجرای الگوریتم می باشد. در بخش ۲ معرفی MCP آمده است. بخش ۳ و ۴، اختصاص به حل MCP با استفاده از EO دارد. برای بررسی کارآمدی الگوریتم پیشنهادی در حل MCP، از ۳۷ گراف Benchmark^{۱۲} که در DIMACS برای این مسئله استفاده

آمده،

شد

(<http://cs.hbg.psu.edu/benchmarks/clique.html>). نتایج حاصل از تنظیم پارامتر τ و نتایج حاصل از زمان اجرای الگوریتم پیشنهادی EO-MaxClique و مقایسه آن با الگوریتم HGA، در بخش ۵ و ۶ آمده است. نتیجه گیری نهایی نیز در بخش ۷ قابل مشاهده است.

۲- تعریف مسئله

در MCP، گراف G را داریم که دارای n رأس

توسط هرس مبتنی بر اعمال یک محدودیت ساده، گره های بدون استفاده درخت جست و جو را جدا می کند. این الگوریتم با افزایش تعداد رئوس، کارایی خود را از دست می دهد [۴،۵]. یکی دیگر از راه حل های پیش رو برای حل MCP، استفاده از الگوریتم ژنتیک (GA^۵) بود. کاتر و پارک [۷] نشان دادند که استفاده از یک GA ساده برای حل MCP مناسب نیست و به این موضوع اشاره داشتند که GA ها برای حل این مسئله، نیاز به ترکیب با تکنیک های دیگری برای بهبود کارایی شان دارند. در الگوریتم ژنتیک مبتنی بر اکتشاف (HGA^۶) پیشنهادی توسط مارچیوری، یک رویه فرا اکتشافی حریمانه، ابتدا زیرمجموعه گره ها را با اضافه کردن گره های تصادفی انتخاب شده توسعه می دهد؛ سپس آن را به یک کلیک کاهش می دهد و نهایتاً آن را به بزرگترین کلیک بسط می دهد [۸،۹]. الگوریتم EA/G^۷ مبتنی بر الگوریتم ژنتیک مارچیوری و به همراه عملگر جهش راهنمایی شده جدیدی ارائه شد که از اطلاعات آماری عمومی مرتبط با اطلاعات محلی استفاده می کرد. نتایج آزمایشگاهی اجرای EA/G، حاکی از خروجی های بهتر آن نسبت به HGA بود [۱۲]. الگوریتم Ant-Clique نیز نمونه ای از حل MCP به روش کلونی مورچگان (ACO^۸) است که با استفاده از یک اکتشاف حریمانه متوالی، کلیک های ماکسیمال را با افزودن مکرر رئوس به کلیک های جزئی، ایجاد می کند [۱۳،۱۴]. این رئوس، براساس دنباله های فرومون باقی مانده در مسیر حرکت مورچه های قبلی، انتخاب می شوند. الگوریتم دو مرحله ای IEA/PTS^۹، که مبتنی بر جست و جوی ممنوعه^{۱۰} است، یکی از بهترین الگوریتم های پیشنهادی برای حل MCP بود که از یک استراتژی تکاملی مؤثر دو مرحله ای کشف مسئله و یافتن راه حل، استفاده می کرد [۱۵،۱۶]. اندازه راه حل های تولید شده

⁵ Genetic Algorithm

⁶ Heuristic Genetic Algorithm

⁷ Evolutionary Algorithm With Guided Mutation

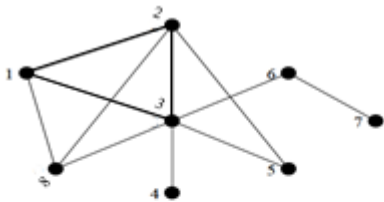
⁸ Ant Colony Optimization

⁹ Evolutionary Algorithm with Probabilistic Tabu Search

¹⁰ Tabu Search

¹¹ Extremal Optimization

از مقداردهی اولیه رئوس در یک گراف فرضی را در شکل ۱ مشاهده می‌نمایید.



رئوس کلیک: ۱-۲-۳

آرایه ایجاد شده در این مرحله:

شماره رأس	۱	۲	۳	۴	۵	۶	۷	۸
مقدار	۳	۳	۳	۹	۱۰	۱۱	۱۲	۱۳

شکل ۱. مقداردهی اولیه به رئوس گراف

۳-۲- تابع برازندگی محلی

هسته اصلی الگوریتم EO، یافتن تابع برازندگی محلی است. کافی است در هر مرحله این تابع را فقط برای رئوس موجود در کلیک بررسی کنیم و رئوس با برازندگی کم را با رئوس جدید، جایگزین کنیم. پس مقدار تابع برازندگی برای رئوسی که عضو کلیک نیستند، برابر ۱ خواهد بود.

اگر تعداد رئوس مجاور تمام رئوس در ابتدای الگوریتم، فقط یک بار محاسبه شود، کافی است در هر مرحله تابع ارزیابی برای رئوس عضو کلیک، به صورت معادله (۱) مقداردهی شود:

$$\lambda(v_i) = \begin{cases} 1 - \frac{(C-1)}{\text{Adjacent}(v_i)} & \text{if } v_i \in \text{Clique} \\ 1 & \text{o.w.} \end{cases} \quad (1)$$

که در رابطه فوق، آرایه Clique، همان کلیک ایجاد شده در مرحله قبل است؛ C تعداد رئوس (عناصر) Clique و $\lambda(v_i)$ مقدار برازندگی رأس v_i است. بعد از یافتن برازندگی تمام رئوس، طبق الگوریتم اکستریمال، کافی است این مقادیر برازندگی را از کوچک به بزرگ مرتب کرده و کوچک‌ترین مقدار را که به عبارتی بدترین رأس می‌باشد را انتخاب کرده و با رأسی تصادفی تعویض کنیم [۱۷، ۱۸، ۱۹، ۲۰]. در روش پیشنهادی، ابتدا مجاورت رأس تصادفی با تمام رئوس کلیک بررسی شده و در صورت تأیید، همین روال در مورد مجاورهای این رأس تصادفی نیز تکرار می‌شود و رئوس مورد تأیید، یکی یکی به رئوس بزرگ‌ترین کلیک ما در این مرحله، افزوده می‌شوند.

$V = \{V_1, \dots, V_n\}$ است و از هر رأس، یال‌هایی عبور می‌کند؛ مجموعه یال‌های عبوری از رأس i ، برابر $E_i = \{e_{i1}, \dots, e_{im}\}$ است که $E \subseteq V \times V$ هدف، یافتن بزرگ‌ترین زیرگراف کاملی است که در G وجود دارد.

۳- یافتن بزرگ‌ترین کلیک با الگوریتم اکستریمال

الگوریتم EO، الگوریتمی فرا اکتشافی است که از مدل بیولوژیکی بک-اسنپن الهام گرفته شده است [۱۸]. در این مدل، گونه‌هایی که در یک گراف قرار دارند، دارای مقادیر برازندگی^{۱۳} در بازه [۰، ۱] هستند. در هر مرحله، گونه‌ای که کمترین مقدار برازش را دارد، برای به روز رسانی تصادفی انتخاب می‌شود و مقدار تابع ارزیابی‌اش را با یک مقدار تصادفی در بازه [۰، ۱] عوض می‌کند. بنابراین همه گونه‌های متصل به «ضعیف‌ترین» برازندگی، خود را با اعداد تصادفی جایگزین می‌کنند. بعد از چند مرحله، سیستم به یک وضعیت همبسته به نام SOC^{۱۴} می‌رسد [۱۹]. در این حالت، تقریباً تمامی گونه‌ها، به یک برازندگی بالاتر از یک حد آستانه مشخص می‌رسند و فقط همسایه ضعیف‌تر می‌تواند مقدار برازندگی خود را کاهش دهد. این فعالیت، بهمین^{۱۵} ایجاد می‌کند.

۳-۱- روش حل

تفاوت روش فرا اکتشافی EO با GA این است که علاوه بر نیاز به دانستن برازندگی کلی جواب، باید برازندگی محلی هر یک از سلول‌ها را هم بدانیم. برای حل مسئله، فرض می‌کنیم که آرایه‌ای یک بعدی از رئوس گراف داریم که در آن، رئوسی که عضو کلیک در حال رشد هستند، مقداری برابر با اندازه کلیک فعلی دارند و سایر رئوس، مقداری برابر یک شمارنده در حال افزایش، با شروع از عدد «شماره آخرین رأس + ۱» دارند. نمونه‌ای

¹² Species

¹³ Fitness

¹⁴ Self-Organized Criticality

¹⁵ Avalanches

۴- اجرای الگوریتم

$$k = (1 + (n^{1-\tau} - 1) \cdot \text{rand}())^{\frac{1}{1-\tau}} \quad (2)$$

که در رابطه بالا k ، به شماره کلیک اشاره دارد که از لیست مرتب شده رتبه‌ها انتخاب شد؛ n تعداد رئوس گراف و $\text{rand}()$ تابع تولید عدد تصادفی در بازه $[0, 1]$ است.

مقدار پارامتر τ ثابت است؛ برای $\tau \rightarrow \infty$ ، الگوریتم، فضای جست‌وجو را محدود کرده و به عبارتی سخت‌گیرانه‌تر به دنبال پاسخ می‌گردد و برای $\tau \rightarrow 1$ ، فضای جست‌وجو افزایش یافته و از بهینه محلی فرار می‌کند و این خود، باعث بروز بهمن خواهد شد. بنابراین انتظار داریم که الگوریتم پیشنهادی، دارای توزیع قدرت کم $P_k \propto k^{-\tau}$ باشد که در آن $1 \leq k \leq n$ است [۲۴، ۲۵]. بعد از آزمایشات انجام شده، مقدار $\tau = 1.4$ اتخاذ شد.

۵- شبیه سازی و نتایج

الگوریتم پیشنهادی، به تعداد ۳۰ مرتبه روی ۳۷ گراف DIMACS اجرا شد. نتایج حاصل از پیاده سازی را در جدول ۱ می‌توان یافت که به منظور بررسی اندازه بزرگ‌ترین کلیک‌های بدست آمده و مقایسه زمان اجرا، مقادیر متناظر از اجرای الگوریتم HGA نیز در آن گنجانده شده است. الگوریتم EO-MaxClique، توسط نرم افزار Matlab و روی سیستم اینتل با پردازنده 2.6 GHz و الگوریتم HGA، روی AMD 2400 (2GHz) پیاده سازی شده‌اند [۱۲]. همان طور که در جدول ۱ مشاهده می‌شود، از نظر اندازه بزرگ‌ترین کلیک بدست آمده، الگوریتم EO-MaxClique، به خوبی الگوریتم HGA عمل نمی‌کند؛ اما همان طور که پیش از این ذکر شد، هدف از پیشنهاد روش EO، کاهش زمان رسیدن به پاسخ است که با مشاهده زمان اجرای دو الگوریتم و مقایسه آن در جدول ۱، می‌توان ادعا کرد که به هدف مورد نظر دست یافته‌ایم. به عنوان مثال در گراف MANN-a27، الگوریتم HGA، ۱۲۶ را به عنوان بهترین جواب با میانگین ۱۲۵.۶ و انحراف معیار ۰.۵، در زمان ۸.۸ ثانیه بدست آورده است. اما الگوریتم EO-MaxClique، ۱۲۵ را به عنوان بهترین جواب با میانگین ۱۲۴.۳ و انحراف معیار ۰.۸ در زمان ۰.۳۱ ثانیه بدست آورده است. اگر بخواهیم این اختلاف را به صورت عددی بیان کنیم، الگوریتم پیشنهادی در ۳۷ گراف شبیه سازی شده، به طور میانگین در زمانی ۲۵۱ برابر کمتر از الگوریتم HGA، به پاسخ رسیده است. این اختلاف زمانی حتی با احتساب اختلاف قدرت

روش پیشنهادی مبتنی بر الگوریتم بهینه سازی اکستریمال برای یافتن بزرگ‌ترین کلیک در هر گراف مفروض، دارای مراحل زیر است:

Algorithm 1: EO-MaxClique

1. Create initial clique and set the Ingraph array;
2. Calculate the total fitness and set $Clique = Ingraph, MaxClique = Size(Ingraph)$;
3. DO{
4. Iteration ++;
5. Calculate the local fitness (λ_i) for each vertice according to Eq. (1) ;
6. Sort the vertices (Species) according to their local fitness with saving the k(th) rank of each specie;
7. Select the vertices with power-low distribution in Eq. (2) based on calculated λ_i ;
8. Check the adjacency of Clique vertices with the new vertice and it's neighbors, If True: add them to the Clique;
9. If $C > MaxClique$, Set $MaxClique = C$;
10. } While (Termination)
11. Output the MaxClique;

در الگوریتم بالا، آرایه Ingraph، کلیک اولیه است که بر اساس آزمایش‌های انجام شده، ۳ رأس دارد که به صورت کاملا تصادفی انتخاب می‌شوند و اندازه بزرگ‌ترین کلیک که از این رئوس بدست می‌آید نیز در یک روال ساده قابل محاسبه است. مقدار MaxClique، اندازه بزرگ‌ترین کلیک یافت شده در هر مرحله است که نهایتاً خروجی الگوریتم خواهد بود. توضیح سایر متغیرها قبلاً در بخش ۳-۲ ارائه شده است.

۴-۱- تنظیم پارامتر τ

برای فرار از بهینه محلی و پوشش فضای جست‌وجوی بیشتری از هر گراف، به جای اینکه در هر مرحله بدترین رأس انتخاب شود، τ مین رأس بد انتخاب شود؛ یعنی رأسی که رتبه k آن با استفاده از معادله ۲ محاسبه شده است، برای تعویض انتخاب می‌شود.

پردازنده‌ها باز هم قابل درنگ است.

$$\begin{cases} H_0 : \mu_0 = Mean \\ H_1 : \mu_1 < Mean \end{cases} \quad (1)$$

که فرض H_0 نشان دهنده میانگین برابر با مقدار ستون Mean و فرض H_1 نشان دهنده مقادیر میانگین کوچک‌تر از Mean است. مقدار P-Value متناظر با هر آزمون نیز در این جدول آمده است [۲۸،۲۹]. همان طور که ملاحظه می‌شود، در تمام موارد، مقدار آن بزرگ‌تر از 0.05 است و این به معنی پذیرش فرضیه H_0 است [۳۰]. یعنی اثبات کردیم که پاسخ‌های بدست آمده، به طور مطلوبی به سمت مقادیر آمده در ستون Mean از جدول I میل می‌کنند.

۷- نتیجه گیری

در این مقاله راه حلی فرا اکتشافی برای مسئله یافتن بزرگ‌ترین کلیک مطرح می‌شود که مبنای آن، روش بهینه سازی اکستریمال است. نتایج آزمایشگاهی حاصل از اجرای روش پیشنهادی روی ۳۷ گراف DIMACS، حاکی از همگرایی سریع پاسخ‌ها به سمت مقادیر مطلوب و در نتیجه کاهش زمان اجرای الگوریتم است که علت آن، بهبود تنها یک نمونه (به جای کار روی جمعیتی از نمونه‌ها در الگوریتم ژنتیک) در طی حل مسئله است. در نتیجه شاهد برتری الگوریتم EO-MaxClique نسبت به الگوریتم مورد مقایسه یعنی HGA از نظر زمان اجرای الگوریتم هستیم. محاسبات نشان داد که زمان رسیدن به پاسخ در الگوریتم پیشنهادی، ۲۵۱ برابر بهتر از HGA است. اما در مورد بزرگ‌ترین کلیک و میانگین کلیک‌های بدست آمده، در اکثر موارد، برتری با الگوریتم HGA است. به‌کارگیری خاصیت بهمن گونه در روش EO و تنظیم پارامتر τ ، باعث شد که این الگوریتم از بهینه‌های محلی فرار کند و این خود باعث یافتن سریع پاسخ‌ها شد. در نهایت، از دو روش برای اثبات مطلوب بودن پاسخ‌ها استفاده شد. ابتدا با استفاده از رسم نمودار همگرایی پاسخ‌های یکی از گراف‌های DIMACS، نشان دادیم که پاسخ‌ها به صورت یکنواخت همواره در مسیر حل مسئله، در حال بهبود بوده و به سمت پاسخ مناسب همگرا هستند؛ و بعد با استفاده از آزمون آماری ۹۵٪ و در جهت بررسی صحت الگوریتم پیشنهادی، اثبات کردیم که تعداد بزرگ‌ترین کلیک‌های بدست آمده در هر گراف، به سمت مقدار مطلوبی میل می‌کند.

۶- بررسی همگرایی و پایداری الگوریتم پیشنهادی

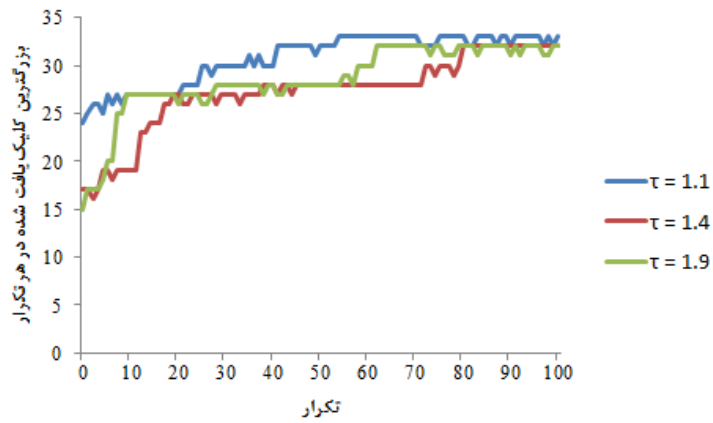
سه موضوع در الگوریتم پیشنهادی قابل بررسی است؛ مقدار پارامتر τ ، همگرایی و پایداری. پارامتر τ برای پیاده سازی تمامی گراف‌ها در این روش، عدد ثابت ۱.۴ را گرفته است. تعداد تکرار در هر بار اجرای الگوریتم، ۱۰۰۰ بوده است. نمودار شکل ۲، مقایسه ای از نتایج اجرای الگوریتم پیشنهادی روی گراف P-Hat300-3 با مقادیر $\tau = \{1.1, 1.4, 1.9\}$ است. برای نمایش واضح‌تر، شکل ۲ (آ) ۱۰۰ تکرار اول و شکل ۲ (ب) تمامی ۱۰۰۰ تکرار را نمایش می‌دهند. این آزمایش که روی اکثر گراف‌های DIMACS انجام شد، نشان دهنده این است که $\tau = 1.4$ مقدار مناسبی برای فرار الگوریتم از بهینه‌های محلی است.

برای بررسی همگرایی و یکنواختی پاسخ‌های بدست آمده، الگوریتم پیشنهادی را روی گراف P_Hat300-3، ۳ مرتبه اجرا کرده و نمودار همگرایی کلیک‌های بدست آمده در هر مرحله آن را در قالب شکل ۳ ارائه دادیم. در اینجا نیز برای حصول وضوح بیشتر، شکل ۳ را در دو قسمت نمایش داده‌ایم که شکل ۳ (آ) ۱۰۰ تکرار اول و شکل ۳ (ب) تمامی ۱۰۰۰ تکرار را نمایش می‌دهند.

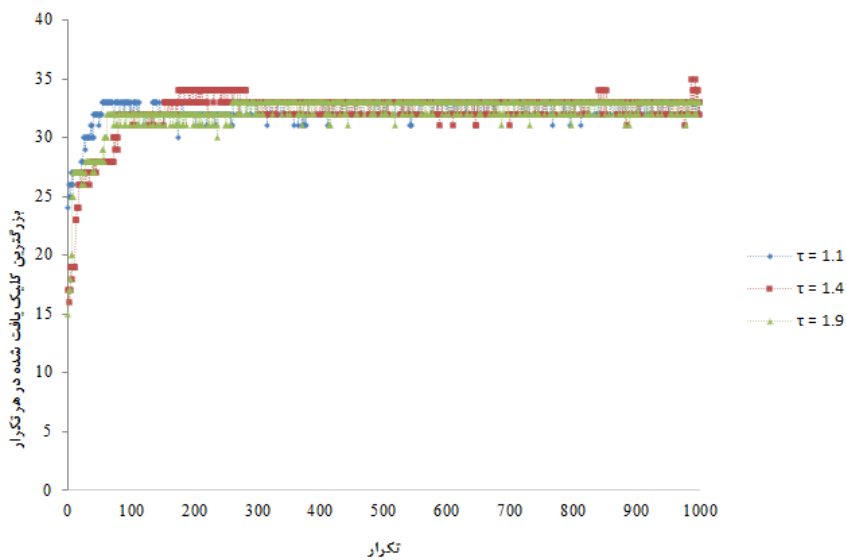
همان طور که مشخص است، پاسخ‌ها به سمت مقدار مطلوبی مانند ۳۵ همگرا هستند که البته انتخاب مقدار مناسب برای پارامتر τ نیز به وجود همگرایی بیشتر، کمک کرده است؛ در مسیر حل مسئله، شاهد بهبود یکنواخت پاسخ‌ها هستیم که این موضوع نیز نمایانگر پایداری پاسخ‌هاست.

۶-۱- آزمون آماری

تاکنون بررسی جدول I نشان داد که میانگین پاسخ‌های بدست آمده از اجرای الگوریتم EO-MaxClique نزدیک به مقادیر DIMACS هستند. برای اثبات این موضوع، در ادامه جدول I، از آزمون آماری t ۹۵٪ برای تمامی گراف‌ها استفاده شد [۲۷] که فرض‌های آن به صورت معادله (۳) بوده است. در ستونی با نام Mean از جدول I مقادیری نزدیک به مقدار DIMACS هر گراف قرار داده شده که آزمون نسبت به آن ستون انجام شده است.

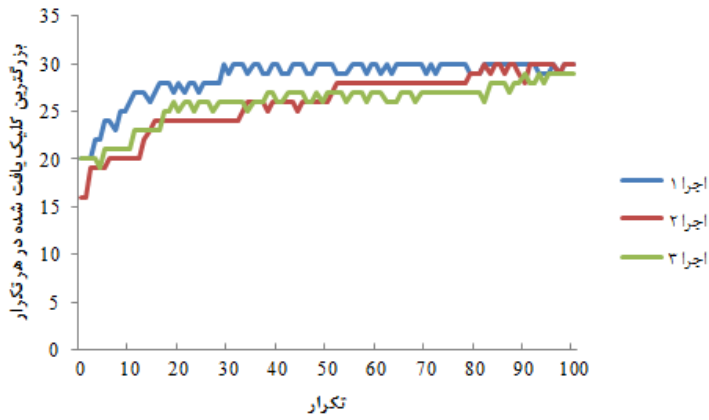


شکل ۲ (آ)

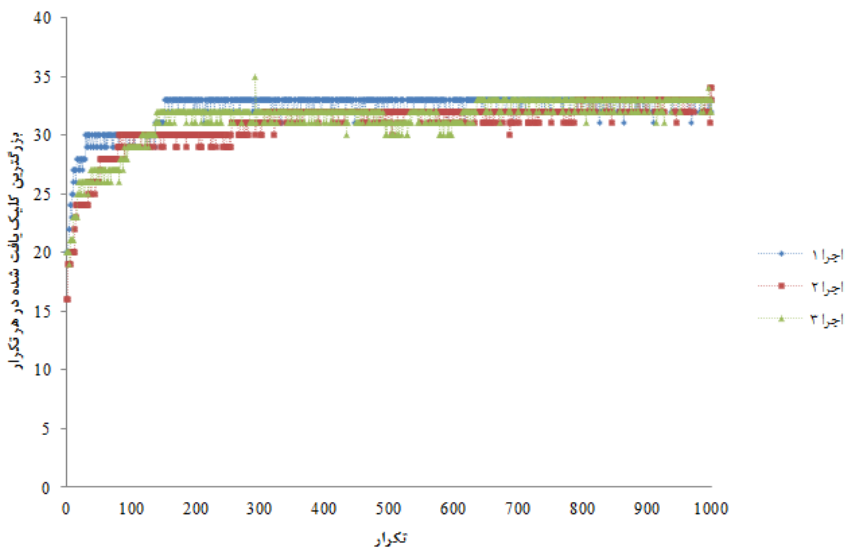


شکل ۲ (ب)

شکل ۲: مقایسه اجرای الگوریتم پیشنهادی با مقادیر مختلف τ ، روی گراف P_Hat300-3، ۲(آ): ۱۰۰۰ تکرار اول، ۲(ب): تمام ۱۰۰۰ تکرار.



شکل ۳ (آ)



شکل ۳ (ب)

شکل ۳: بررسی همگرایی الگوریتم پیشنهادی روی گراف P_Hat300-3، ۳(آ): ۱۰۰۰ تکرار اول، ۳(ب): تمام ۱۰۰۰ تکرار.

- Theory via Clique Finding ,” IEEE Transactions On Systems, Man, and Cybernetics—part B: Cybernetics, vol. 38, no. 3, 2008.
- [16] M. Gendreau and P. Soriano and L. Salvail, ”Solving the maximum clique problem using a tabu search approach,Annals of Operations Research,” Volume 41, Number 4, 1993.
- [17] S. Boettcher and A. G. Percus, ”Extremal Optimization: Methods derived from Co-Evolution,” in Proceedings of the Genetic and Evolutionary Computation Conference, 1999.
- [18] S. Boettcher and A. G. Percus, ”Nature’s way of optimizing Artificial Intelligence,” 119, 275-286, 2000.
- [19] S. Boettcher and A. G. Percus, ”Optimization with Extremal Dynamics,” Phys. Rev. Lett. 86, 5211-5214, 2001.
- [20] S. Boettcher and A. G. Percus, ”ExtremalOptimization: An Evolutionary Local-Search Algorithm,”CoRR cs.NE/0209030, 2002.
- [21] S. Boettcher and A. G. Percus, ”Extremal Optimization for Graph Partitioning,” Phys. Rev. E, 64, 2001.
- [22] M. El .Menai and M.Batouche, ”Efficient Initial Solution to Extremal Optimization Algorithm for Weighted MAXSAT Problem,” Springer-Verlag Berlin HeidelbergLNAI 2718, pp. 592-603, 2003.
- [23] G. Zeng and Y. Lu, Z. et al, ”Modified extremal optimization for the hard maximum satisfiability problem,” J Zhejiang Univ-Sci C (Comput& Electron) 2011.
- [24] M. VafaeiJahan and M.R. AkbarzadehTotonchi, ”Hybrid local search algorithm via evolutionary avalanches for spin glass based portfolio selection,” Egyptian Informatics Journal, Vol. 13, Issue 2, pp:65-73, 2012.
- [25] M. VafaeiJahan and M.R. AkbarzadehTotonchi, ”Extremal Optimization vs. Learning Automata: Strategies for Spin Selection in Portfolio Selection Problems,” Applied Soft Computing, In Press, Corrected Manuscript, Available online 7 June 2012.
- [26] S. Boettcher, ”Extremal Optimization: Heuristics Via Coevolutionary Avalanches,” Computer Simulations, 2000.
- [27] G.K. Kanji, ”100Statistical test,” Sage Publications Ltd, Third edition published 2006.
- [28] P. W. Zehna, ”Excel Manual,” 6/E, Neil A. Weiss, Arizona State University, pp. 119-131, 2002.
- [29] L. A. Pace, ”The Excel 2007 Data & Statistics Cookbook: A Point-and-Click! Guide,” Published in the United States of America,pp. 43-46, 2007.
- [30] M. Nosratabadi and M. VafaeiJahan and M.R. AkbarzadehTotonchi and M. Gharejanloo, ”Fuzzy Job Shop Scheduling Using Extremal Optimization Algorithm,” Iranian Fuzzy systems Conference, University of Sistan and Baluchestan, 2011.
- [1] I. M. Bomze and M. Budinich and P. M. Pardalos and M. Pelillo,The Maximum Clique Problem, Handbook of Combinatorial Optimization, Kluwer Academic Publishers,pp. 2-14, 1999.
- [2] I. M. Bomzea and M. Budinichb and M. Pelilloc and C. Rossic, ”Annealed replication: a new heuristic for the maximum clique problem,” Discrete Applied Mathematics 121 27–49, 2002.
- [3] L. Engebresten and J. Holmerin, ”Clique Is Hard to approximate Within n,” Royal Institute of Technology Sweden, 2000.
- [4] R. Carraghan and P.M. Pardalos. An exact algorithm for the maximum clique problem. Operations Research Letters 9:375–382, 1990.
- [5] L. Babel and G. Tinhofer, ”A Branch and Bound Algorithm for the Maximum Clique Problem,” ZOR-Methods andModels of Operations Research, 34: 207-217, 1990.
- [6] J.Konc and D.Jane`zi`, ”An improved branch and bound algorithm for the maximum clique problem,” MatchCommun. Math. Comput. Chem. 58 569-590, 2007.
- [7] K. Park and B. Carter, ”On the effectiveness of genetic search in combinatorial optimization,” Comput. Sci. Dept., Boston Univ., Boston, MA, Tech. Rep. BU-CS-94-010, 1994.
- [8] E. Marchiori, ”A simple Heuristic Based Genetic Algorithm for the Maximum Clique Problem,” ACM Symposium on Applied Computing, pp. 366-373, 1998.
- [9] E. Marchiori, ”Genetic, iterated and multistart local search for the maximum clique problem,” in Applications of Evolutionary Computing. Berlin, Germany: Springer-Verlag, 2002, LNCS 2279, pp. 112–121.
- [10] k. katayama and A. kohmura and k. kohmoto and H. Minamihara, ”Memetic algorithm with strategic controller for the maximum clique problem,” Proceedings of the ACM Symposium on Applied Computing, 2011.
- [11] D. C.Porumbel and J-K Hao and P. Kuntz, ”Spacing Memetic Algorithms,” ACM 978-1-4503-0557, 2011.
- [12] Q. Zhang and J. ong Sun, and E. Tsang, ”An Evolutionary Algorithm With Guided Mutation for the Maximum Clique Problem,” IEEE Transactions On Evolutionary Computation, vol. 9, no. 2, april 2005.
- [13] S. Fenet and Ch. Solnon, ”Searching for maximum cliques with ant colony optimization,” ACM, Proceedings of the international conference on Applications of evolutionary computing, 2003.
- [14] Ch.Solnonand D. Bridge, ”An Ant Colony Optimization Meta-Heuristic for Subset Selection Problems,” pp. 1-23, 2005.
- [15] P.Guturu and S. Memberand R. Dantu, ”An Impatient Evolutionary Algorithm With Probabilistic Tabu Search for Unified Solution of Some NP-Hard Problems in Graph and Set

جدول 1: مقایسه دو الگوریتم HGA و EO-MaxClique به همراه نتایج آزمون آماری t روی الگوریتم EO-MaxClique. Best: اندازه بزرگترین کلیک یافت شده، avg: میانگین اندازه بزرگترین کلیک یافت شده، std: انحراف معیار اندازه بزرگترین کلیک یافت شده، time: کمترین زمان اجرای هر الگوریتم بر حسب ثانیه، T-Statistic: نتایج آزمون آماری الگوریتم EO-MaxClique

Graph	HGA				EO-MaxClique				T-Statistic			DIMACS
	Best	Avg	Std	time	Best	Avg	Std	time	t	P-Value	Mean	
C125.9	34	33.8	0.4	1.3	34	31.1	1.2	0.01	1.3693	0.091	31.4	34
C250.9	44	42.8	0.7	2.4	42	38.8	1.8	0.07	1.5215	0.069	39.3	44
C500.9	56	52.2	1.6	4.9	50	45.4	2.0	0.21	1.6432	0.056	46	57
C1000.9	66	61.6	2.1	17.4	51	45.9	1.6	0.10	1.3693	0.091	46.3	68
c2000.9	72	68.2	2.4	39.3	63	59.6	1.9	1.66	1.4414	0.080	60.1	80
DSJC500.5	13	12.2	0.4	3.5	12	11.5	0.4	0.02	1.3693	0.091	11.6	13
DSJC1000.5	14	13.5	0.5	9.2	14	12.8	0.5	0.06	1.0954	0.141	12.9	15
C2000.5	15	14.2	0.4	23.6	15	13.8	0.5	0.20	1.0954	0.141	13.9	16
C4000.5	16	15.4	0.5	50.8	15	13.9	0.4	0.17	1.3693	0.091	14	18
MANN-a27	126	125.6	0.5	8.8	125	124.3	0.8	0.31	1.3693	0.091	124.5	126
MANN-a45	343	342.4	0.5	74.3	339	331.1	11.7	7.95	1.6853	0.051	334.7	345
MANN-a81	1097	1096.3	0.6	720.5	993	802	63.9	46.13	1.2857	0.06	820	1100
brock200-2	12	10.5	0.7	1.4	10	9.6	0.4	0.01	1.3693	0.091	9.7	12
brock200-4	16	15.4	0.5	1.4	16	14.9	0.5	0.01	1.0954	0.141	15	17
brock400-2	24	22.5	0.7	3.0	24	21.6	0.8	0.02	1.3693	0.091	21.8	25
brock400-4	25	23.6	0.8	3.0	24	21.7	0.7	0.03	1.5649	0.064	21.5	24
brock800-2	21	19.3	0.6	7.6	20	17.7	0.7	0.04	1.5649	0.064	17.9	24
brock800-4	20	18.9	0.5	7.8	19	17.9	0.8	0.05	1.3693	0.091	18.1	26
gen200...44	44	39.7	1.6	19.9	37	34.4	1.3	0.02	1.6853	0.051	34.8	44
gen200...55	55	50.8	6.4	24.0	40	37.5	1.3	0.01	1.6853	0.051	37.9	55
gen400...55	51	49.7	1.2	38.9	47	43.9	1.8	0.05	1.5215	0.069	44.4	55
gen400...65	65	53.7	7.4	41.8	48	42.5	0.2	0.05	0	0.500	42.5	65
gen400...75	75	60.2	12.1	38.3	47	44.7	1.2	0.04	1.3693	0.091	45	75
hamming8-4	16	16	0.0	1.8	16	15.6	1.1	0.00	1.4938	0.073	15.9	16
hamming10-4	40	37.7	1.9	14.6	37	33.3	1.6	0.03	1.3693	0.091	33.7	40
keller4	11	11	0.0	1.2	11	10.8	0.4	0.01	1.3693	0.091	10.9	11
keller5	27	26	0.8	8.3	27	22.1	1.3	0.02	1.6853	0.051	22.5	27
keller6	56	51.8	1.5	56.0	48	43.6	1.8	1.47	1.5215	0.069	44.1	59
p-hat300-1	8	8	0.0	2.0	8	7.5	0.4	0.01	1.3693	0.091	7.6	8
p-hat300-2	25	25	0.0	2.2	25	24.9	0.1	0.02	0	0.500	24.9	25
p-hat300-3	36	34.6	0.9	2.5	35	32.6	0.8	0.03	1.3693	0.091	32.8	36
p-hat700-1	11	9.8	0.9	5.0	11	8.8	0.6	0.02	0.9129	0.184	8.9	11
p-hat700-2	44	43.5	0.8	7.5	43	40.9	1.3	0.10	1.6853	0.051	41.3	44
p-hat700-3	62	60.4	0.1	9.2	61	58.2	1.2	0.12	1.3693	0.091	58.5	62
p-hat1500-1	11	10.8	0.4	14.5	11	10.5	0.4	0.14	1.3693	0.091	10.6	12
p-hat1500-2	65	63.8	1.0	23.4	64	60	1.7	0.67	1.6109	0.059	60.5	65
p-hat1500-3	94	92.1	0.9	28.8	88	83.8	2.1	0.89	1.5649	0.064	84.4	94