دوفصلنامه محاسبات و سامانه های توزیع شده
سال پنجم، شماره اول، شماره پیاپی ۹، صفحه ۲۱ تا ۳۰، سال ۱۴۰۱

حل مسأله رنگ آمیزی جمعی گراف با استفاده از الگوریتم ابتکاری

مرتضی رجب زاده*[۱]، امین محمدنژاد[۲]، کوروش عشقی[۳]
[۱]استادیار، دانشکده مهندسی، مرکز آموزش عالی محلات، محلات، ایران.
[۲]محقق، دانشکده مهندسی، مرکز آموزش عالی محلات، محلات، ایران.
[۳]استاد، دانشکده مهندسی صنایع، دانشگاه صنعتی شریف، تهران، ایران.

چکیده

رنگ آمیزی جمعی گراف، اختصاص اعداد طبیعی به رئوس یگ گراف ساده می باشد، طوری که مجموع اعداد اختصاص داده شده به رئوس گراف، کمینه گردد. مهمترین کاربرد آن در حوزره زمانبندی می باشد. برای این مسأله که جزو مسائل NP-Hard[1] می باشد، تاکنون حل دقیقی ارائه نشده است. لذا در این پژوهش، یک الگوریتم ابتکاری مرکب، بر مبنای ایده شناسایی مجموعه های مستقل رئوس گراف و اختصاص کوچکترین عدد طبیعی در دسترس، برای بزرگترین مجموعه مستقل، توسعه داده شده است. الگوریتم پیشنهادی، بر روی گراف های موجود در کتابخانه های معروف گراف هایی که به صورت تصادفی تولید شده اند، آزمایش شده است. نتایج، نشان دهنده کارایی الگوریتم ارائه شده می باشد.

کلمات کلیدی: رنگ آمیزی جمعی گراف، مجموعه مستقل رئوس گراف، الگوریتم فراابتکاری.

تاریخچه مقاله:
تاریخ ارسال: ۱۴۰۰/۱۲/۱۸
تاریخ اصلاحات: ۱۴۰۱/۰۱/۰۴
تاریخ پذیرش: ۱۴۰۱/۰۳/۲۵
تاریخ انتشار: ۱۴۰۱/۰۵/۱۱

Keywords:
Graph Sum Colouring,
Independent Set,
Meta-heuristics

*ایمیل نویسنده مسئول:
rajabzadeh.m@gmail.com

Solving the Graph Sum Colouring Problem Using a Heuristic Algorithm

Morteza Rajabzadeh*[1], Amin Mohammadnejad[2], Kourosh Eshghi[3]

[1,2]Faculty of Engineering, Mahallat Institute of Higher Education, Mahallat, Iran
[3]Faculty of Industrial Engineering, Sharif University of Technology, Tehran, Iran

Abstract

Graph sum colouring is the assignment of natural numbers to the vertices of a simple graph so that the sum of the numbers assigned to the vertices of the graph is minimized. Its most important application is in the scheduling problems. Given that this is one of the NP-Hard problems, no exact solution has been provided so far. Therefore, in this study, a hybrid heuristic, based on the idea of identifying independent sets of graph vertices and assigning the smallest natural number available for the largest independent set has been developed. The proposed algorithm is tested on graphs in well-known libraries of randomly generated graphs. The results show the efficiency of the proposed algorithm.

Keywords: Graph Sum Colouring, Independent Set, Meta-heuristics.

[1]Non-deterministic Polynomial Hard problems

۱ – مقدمه

فرض کنید $G=(V,E)$ یک گراف ساده با مجموعه رئوس V و مجموعه یال های E باشد. رنگ آمیزی k رنگی گراف G نگاشت $C: V \to N$ می باشد، که به هر رأس گراف، یک عدد طبیعی اختصاص می دهد. اگر برای هر $uv \in E$ داشته باشیم $C(u) \neq C(v)$ آنگاه رنگ آمیزی $C(G)$ یک رنگ آمیزی مناسب k رنگی گراف G نامیده می شود. رنگ آمیزی مناسب گراف، در واقع افراز رئوس آن به مجموعه های $C^1,\dots,C^k$ می باشد که در آن، برای همه رئوس $v \in C^i$ داریم $C(v)=i$. عدد رنگی گراف که با $\chi(G)$ نشان داده می شود، برابر با کوچکترین عدد طبیعی k است، که به ازای آن، یک رنگ آمیزی مناسب k رنگی گراف، وجود داشته باشد. مجموع رنگی گراف G با رنگ آمیزی C به صورت زیر تعریف می شود:

$$Sum(C)=\sum_{v\in V} C(v) = 1\times|C^1| + 2\times|C^2| + \cdots + k\times|C^k|$$

مسأله رنگ آمیزی جمعی گراف، به دنبال پیدا کردن یک رنگ آمیزی مناسب، مانند C از رئوس گراف است، طوری که $Sum(C)$ کمینه گردد. این مقدار کمینه، جمع رنگی گراف نامیده شده و با $\sum(G)$ نشان داده می شود. تعداد رنگ های مورد استفاده در رنگ آمیزی، منجر به جمع رنگی گراف، قدرت گراف نامیده شده و با $s(G)$ نشان داده می شود. واضح است که عدد رنگی گراف، حد پایین قدرت گراف است، یا به عبارتی $s(G) \geq \chi(G)$.

کوبیکا و شونک[2] [۱] مفهوم جمع رنگی گراف را برای اولین بار معرفی کرده و ثابت نمودند که مسأله رنگ آمیزی جمعی گراف، یک مسأله NP-hard می باشد. برای برخی خانواده گراف ها، از جمله درخت ها، الگوریتم های با زمان چند جمله ای وجود دارد. در مرور این موارد، کوبیکا [۲] همچنین محاسبه حدود پایین جمع رنگی به صورت نظری را انجام داده است. روش های متعددی از منظر کاربردی و بر مبنای الگوریتم های ابتکاری و فراابتکاری جهت حل مسأله رنگ آمیزی جمعی گراف پیشنهاد شده است.

برخی از روش های ارائه شده بر مبنای استفاده از یک الگوریتم فراابتکاری، جهت کمینه کردن تابع هدف مسأله یعنی مجموع رنگی گراف می باشد. الگوریتم ژنتیک [۳]، الگوریتم حریصانه [۴]، الگوریتم جستجوی ممنوع [۵] و الگوریتم جستجوی محلی [۶]، از الگوریتم های فراابتکاری مورد استفاده در ادبیات موضوع می باشند. در برخی از روش های موجود در ادبیات موضوع نیز، از ترکیب دو یا چند الگوریتم فراابتکاری (به عنوان مثال الگوریتم جستجوی محلی و الگوریتم ژنتیک) استفاده شده است [۷]. در کنار این روش ها، الگوریتم های ابتکاری، بر مبنای استفاده از ویژگی های خاص مسأله رنگ آمیزی جمعی گراف (به عنوان مثال، شناسایی بزرگترین مجموعه مستقل رئوس گراف و تخصیص رنگ شماره ۱ به آن) استفاده شده است [۸-۱۱]. برای مرور جامع تر روش های مختلف موجود در ادبیات موضوع، به مرجع [۱۲] رجوع گردد.
دسته دیگر روش های موجود در ادبیات موضوع، بر محاسبه حدود پایین برای جمع رنگی گراف، متمرکز شده [۱۳, ۱۴] یا از روش های دقیق برنامه ریزی عدد صحیح، برای مدل سازی و حل مسأله رنگ آمیزی جمعی گراف استفاده کرده اند [۱۵, ۱۶].

روش های ارائه شده ادبیات موضوع، در مسائل با اندازه بزرگ (مسائل دنیای واقعی، عموماً اندازه بزرگی دارند)، کارایی پایینی دارند. دلیل این موضوع، عدم توجه همزمان این روش ها، به سه مؤلفه "توانایی جستجوی گسترده در فضای جواب"؛ "عمق بالای جستجو" (توانایی شناسایی جواب های بهینه محلی)؛ و "استفاده از اطلاعات هیوریستیکی مسأله تحت بررسی" است. لذا، در این پژوهش، سعی شده تا با ترکیب چند الگوریتم فراابتکاری با ویژگی های مکمل، شامل الگوریتم سیستم مورچگان، الگوریتم تبرید تدریجی و الگوریتم جستجوی محلی، کارایی الگوریتم در جستجوی فضای جواب، افزایش یافته و در کنار آن، با حذف مجموعه رئوس مستقل در هر مرحله از الگوریتم (اطلاعات هیوریستیکی)، با کاهش اندازه رئوس و یال های گراف، عملکرد الگوریتم، در مسائل با اندازه بزرگ بهبود یابد.
در این مقاله، الگوریتم ابتکاری GSC[3] برای حل مسأله رنگ آمیزی جمعی گراف، پیشنهاد شده است. در این الگوریتم، از مفهوم *بزرگترین مجموعه مستقل* استفاده شده است. برای این منظور، اندازه یک مجموعه مستقل به صورت تابع زیر تعریف می گردد:

$$\psi_M(S) = M \times |S| + |E(S, V-S)|$$

که در آن $E(S,V-S)$ نشان دهنده مجموعه یال های بین رئوس S و مجموعه رئوس $V-S$ است. M پارامتر الگوریتم می باشد. بنابراین در ادامه این مقاله، از ψ به جای ψ_M استفاده می شود.

[2]Kubicka and Schwenk

[3]Graph Sum Colouring

تعریف ۱. فرض کنید S و S' دو مجموعه مستقل گراف G باشند. گفته می شود S بزرگتر از S' است اگر و فقط اگر $\psi_M(S) > \psi_M(S')$. مجموعه مستقل S_B بزرگترین مجموعه مستقل نامیده می شود، اگر برای همه مجموعه های مستقل S داشته باشیم $\psi_M(S_B) > \psi_M(S)$.

تفاوت بین مجموعه مستقل ماکزیمم و بزرگترین مجموعه مستقل، در این است که برای مجموعه مستقل ماکزیمم، فقط اندازه مجموعه (تعداد رئوس) در نظر گرفته می شود، در حالی که برای بزرگترین مجموعه مستقل، هم اندازه مجموعه و هم تعداد یال های متصل به رئوس آن مجموعه و رئوس متمم آن مجموعه، به طور همزمان در نظر گرفته می شود.

در این پژوهش، علاوه بر مجموع رنگی، تابع هدف دیگری برای ارزیابی رنگ آمیزی ها استفاده می گردد:

$$g(C) = \sum_{i\in V(G)} C(i) - M_1 \times |C^1| - M_2 \times |E(C^1, V - C^1)|$$

که در آن، $E(C^1, V - C^1)$ نشان دهنده مجموعه یال های متصل به رئوس اولین کلاس رنگی می باشد، M_1 و M_2 اعداد صحیح و پارامترهای الگوریتم می باشند. این تابع، سه معیار را به صورت همزمان لحاظ می نماید: مجموع رنگی؛ اندازه مجموعه رئوس متعلق به اولین کلاس رنگی؛ و تعداد یال های متصل به این رئوس. بهترین رنگ آمیزی از منظر تابع هدف g، با C_g^* و اولین کلاس رنگی آن، با C_g^{*1} نشان داده می شود.

در ادامه، نتایج حل مسأله برای ۳۸ گراف آزمایشی در ادبیات موضوع ارائه شده است که نشان می‌دهد الگوریتم پیشنهادی با توجه به گزارش سایر روش های ابتکاری و فراابتکاری، به نتایج رقابتی دست یافته است. در واقع، رویکرد پیشنهادی، در بسیاری از موارد، به بهترین جواب های یافته شده توسط روش های موجود در ادبیات موضوع، رسیده یا توانسته برخی از آنها را بهبود بخشد. علاوه بر این، اهمیت و ضرورت اجرای الگوریتم GSC نشان داده شده است.

مابقی این مقاله به شرح زیر سازماندهی شده است:

در بخش بعدی، الگوریتم پیشنهادی GSC تشریح شده است. در بخش ۳، نتایج محاسباتی ارائه شده و با الگوریتم های موجود از ادبیات مقایسه شده است. در بخش ۴، مطالعه بیشتری در خصوص عملکرد الگوریتم GSC انجام شده و نهایتاً جمع بندی در بخش ۵ ارائه شده است.

۲ – الگوریتم پیشنهادی

در این بخش، الگوریتم پیشنهادی GSC برای حل مسأله رنگ آمیزی جمعی گراف تشریح می گردد. این الگوریتم از سه زیر الگوریتم تشکیل شده است. در ابتدا با استفاده از زیر الگوریتم سیستم مورچگان که BISANT نامیده می شود، بزرگترین مجموعه مستقل ممکن جستجو می گردد. سپس با استفاده از یک الگوریتم تبرید تدریجی، خروجی مرحله قبل، بهبود می یابد. این زیر الگوریتم، BISSA نامیده می شود. سومین زیر الگوریتم نیز، یک الگوریتم سیستم مورچگان می باشد که رئوس گراف را رنگ آمیزی (شماره گذاری با اعداد طبیعی) می نماید. این زیر الگوریتم که هسته اصلی الگوریتم پیشنهادی است، GCANT نامگذاری شده است. شبه کد الگوریتم پیشنهادی GSC به صورت زیر می باشد:

(الگوریتم-۱): الگوریتم GSC

ورودی: گراف $G = (V, E)$.

خروجی: رنگ آمیزی مناسب رئوس گراف G.

۱: تا زمانی که $G \neq \emptyset$:

۲: $\boldsymbol{S_B \leftarrow BISANT(G)}$.

۳: $S_B \leftarrow BISSA(G, S_B, |S_B|)$.

۴: $\{C^*, C_g^*\} \leftarrow GCANT(G, S_B)$.

۵: اگر $sum(C^*) < sum(C_{bs})$ آنگاه:

۶: $\boldsymbol{C_{bs} \leftarrow C^*}$.

۷: پایان حلقه.

۸: $\boldsymbol{G \leftarrow G - \{C_g^{*1}\}}$.

۹: پایان حلقه.

۱۰: پایان.

۲-۱- زیر الگوریتم BISANT

در این بخش، جزئیات زیر الگوریتم BISANT تشریح می گردد. BISANT یک الگوریتم سیستم مورچگان است. در این الگوریتم، در هر تکرار، تعداد m مورچه، به طور متوالی، جواب S را که یک مجموعه مستقل از رئوس گراف هستند می سازند. پس از اینکه یک جواب ساخته شد، فرمون[4] تمام رئوس تبخیر می شود و برای رئوس مجموعه ساخته شده، فرمون آزاد می شود. پس از هر بار تکرار، برای بهترین راه حل یافته شده، فرمون آزاد می شود. کد شبه الگوریتم BISANT به شرح زیر است:

(الگوریتم-۲): الگوریتم BISANT

ورودی: گراف $G = (V, E)$.

خروجی: مجموعه مستقل رئوس گراف G.

۱: فرمون رئوس را مقداردهی اولیه کن.

[4] Pheromone

۲: $\boldsymbol{iter \leftarrow 1}$

۳: تا زمانی که $iter < M$:

۴: $ant \leftarrow 1$.

۵: تا زمانی که $iter < M$:

۶: یک مجموعه مستقل مانند S را بساز.

۷: اگر $f(S) > f(S_B)$ آنگاه:

۸: $S_B \leftarrow S$.

۹: پایان حلقه.

۱۰: فرمون همه رئوس را تبخیر کن.

۱۱: برای رئوس $v \in S$ فرمون آزاد کن.

۱۲: $ant \leftarrow ant + 1$.

۱۳: پایان حلقه.

۱۴: برای همه رئوس عضو بزرگترین مجموعه مستقل پیدا شده، فرمون آزاد کن.

۱۵: $iter \leftarrow iter + 1$.

۱۶: پایان حلقه.

۱۷: پایان.

۲-۲- زیر الگوریتم BISSA

BISSA یک الگوریتم تبرید تدریجی است که در صورت امکان، بزرگترین مجموعه مستقل پیدا شده توسط الگوریتم BISANT را بهبود می بخشد. هدف این الگوریتم یافتن بزرگترین مجموعه مستقل از رئوس، با اندازه κ است. در ابتدا، κ برابر است با اندازه S_B، خروجی الگوریتم BISANT. در هر مرحله، الگوریتم به دنبال یافتن یک مجموعه مستقل، مانند S با اندازه κ است. اگر چنین مجموعه ای پیدا شد، κ یک واحد افزایش می یابد و مرحله بعدی شروع می شود. این روش تا زمانی ادامه می یابد که هیچ مجموعه مستقلی با اندازه κ پیدا نشود. شبه کد الگوریتم BISSA به شرح زیر است:

(الگوریتم-۳): الگوریتم BISSA

ورودی: گراف $G = (V, E)$ و مجموعه مستقل S_B.

خروجی: مجموعه مستقل بهبود یافته.

۱: $S \leftarrow S_B$.

۲: $\kappa \leftarrow |S_B|$.

۳: تا زمانی که شرایط خاتمه الگوریتم برقرار نیست دستورات زیر را اجراء کن:

۴: $T = T_{max}$.

۵: تا زمانی که $T > T_{min}$:

۶: $\boldsymbol{iter \leftarrow 1}$.

۷: تا زمانی که $iter < M$:

۸: همسایگی S' جواب S را تولید کن.

۹: $iter \leftarrow iter + 1$.

۱۰: اگر S' قابل پذیرش است آنگاه:

۱۱: $S \leftarrow S'$.

۱۲: اگر $f(S) = 0$ آنگاه ($f(S) = |E(S)|$):

۱۳: اگر $|E(S, V - S)| > |E(S_B, V - S_B)|$ آنگاه:

۱۴: $S_B \leftarrow S$

۱۵: پایان حلقه.

۱۶: پایان حلقه.

۱۷: $T \leftarrow T - \lambda$.

۱۸: پایان حلقه.

۱۹: $\kappa \leftarrow \kappa + 1$.

۲۰: پایان حلقه.

۲۱: پایان.

۳-۲- زیر الگوریتم GCANT

GCANT یک الگوریتم سیستم مورچگان است که برای رنگ آمیزی مناسب رئوس گراف استفاده می شود. در این الگوریتم، تمام رنگ آمیزی های ساخته شده، مناسب هستند (بین رئوس با رنگ یکسان، یالی وجود ندارد). به عبارت دیگر، الگوریتم، تنها مناطق شدنی فضای حل را جستجو می کند. پس از اینکه یک رنگ آمیزی ساخته شد، با استفاده از الگوریتم جستجوی محلی، در صورت امکان، جواب یافته شده بهبود می یابد. سپس، دنباله فرمون، از طریق تبخیر فرمون همه رئوس و آزادسازی فرمون برای جواب ساخته شده و بزرگترین مجموعه مستقل یافته شده، به‌روزرسانی می‌شود. پس از این که تعداد مشخصی از جواب ها ساخته شدند، برای بهترین جوابی که تاکنون پیدا شده است و همچنین بزرگترین مجموعه مستقل یافته شده، فرمون آزاد می شود. شبه کد الگوریتم GCANT به شرح زیر است:

(الگوریتم-۳): الگوریتم GCANT

ورودی: گراف $G = (V, E)$ و مجموعه مستقل S_B.

خروجی: رنگ آمیزی مناسب گراف.

۱: $S_B \leftarrow BISSA(G)$.

۲: دنباله فرمون ها را مقداردهی اولیه کن.

۳: $\boldsymbol{iter \leftarrow 1}$.

۴: تا زمانی که $iter < M$:

۵: $ant \leftarrow 1$.

۶: تا زمانی که $ant < m$:

۷: رنگ آمیزی مناسب C را بساز.

۸: رنگ آمیزی ساخته شده را با استفاده از الگوریتم جستجوی محلی بهبود بده.

۹: اگر $\psi(C^1) > \psi(S_B)$ آنگاه:

۱۰: $S_B \leftarrow C^1$.

۱۱: پایان حلقه.

۱۲: اگر $sum(C) < sum(C^*)$ آنگاه:

۱۳: $C^* \leftarrow C$.

۱۴: پایان حلقه.
۱۵: اگر $g(C) < g(C_g^*)$ آنگاه:
۱۶: $C_g^* \leftarrow C$.
۱۷: پایان حلقه.
۱۸: دنباله فرمون را تبخیر کن.
۱۹: برای جواب های ساخته شده C و S_B فرمون آزاد کن.
۲۰: $ant \leftarrow ant + 1$.
۲۱: پایان حلقه.
۲۲: برای بهترین جواب یعنی C^* فرمون آزاد کن.
۲۳: $iter \leftarrow iter + 1$.
۲۴: پایان حلقه.
۲۵: پایان.

در این الگوریتم، دنباله فرمون، یک ماتریس $n \times n$ است که در آن τ_{ij} نشان دهنده میزان فرمون مربوط به رنگ آمیزی رأس i با رنگ j است. در هر گام، یک کلاس رنگ توسط هر مورچه، ساخته

می شود. برای این منظور، یک رأس از مجموعه رئوس کاندید A به صورت تصادفی و با استفاده از قاعده انتخاب احتمالی انتخاب شده و به کلاس رنگ جاری منتقل می گردد. سپس تمام رئوس همسایه آن رأس، از مجموعه رئوس کاندید A حذف می گردند. این رویه تا زمانی که مجموعه A تهی گردد ادامه می یابد. در این مرحله، به رنگ جاری یک واحد اضافه شده و همه رئوسی که رنگ آمیزی نشده اند وارد مجموعه A می شوند. سپس مراحل فوق، تا رنگ آمیزی همه رئوس گراف، ادامه می یابد. قاعده انتخاب احتمالی که در بالا اشاره شد به صورت زیر می باشد:

$$P_{i,j}^A = \frac{\tau_{ij} \times {\eta_{i,A}}^{\beta}}{\sum_{k \in A} \tau_{kj} \times {\eta_{k,A}}^{\beta}}$$

که در آن $P_{i,j}^A$ احتمال انتخاب رأس i از مجموعه رئوس کاندید A زمانی که رنگ جاری j است بوده و $\eta_{i,A}$ اطلاعات هیوریستیکی الگوریتم است که به صورت زیر محاسبه می گردد:

$$\eta_{i,A} = \frac{d_{G_{UC}}(i)}{d_A(i)}$$

که در آن G_{UC} نشان دهنده زیر گراف القایی شامل رئوس رنگ نشده و $d_{G_{UC}}(i)$ و $d_A(i)$ به ترتیب نشان دهنده درجه (تعداد یال ها) رأس i در زیر گراف القایی و زیرگراف A می باشد.

منطق پشت این انتخاب، این واقعیت است که اگر یک رأس با تعداد رئوس همسایه کمتر، در لیست کاندید انتخاب شود، احتمال اینکه اندازه کلاس رنگ فعلی بزرگتر شود، زیادتر می شود. همچنین اگر یک رأس با تعداد رئوس همسایه بیشتری در مجموعه رئوس

بی رنگ انتخاب شود تا با رنگ فعلی رنگ آمیزی شود، پس از حذف این رأس از مجموعه رئوس بی رنگ، تعداد یال های بیشتری حذف شده و چگالی زیرگراف حاصل، کمتر می شود. این موضوع مطلوب است زیرا در این حالت، گرافی با چگالی کمتر و احتمالاً با مجموع رنگی کمتر حاصل می گردد.

پس از هر رنگ آمیزی، یک جستجوی محلی برای بهبود رنگ آمیزی اعمال می شود. این امر ضروری است، زیرا الگوریتم های سیستم مورچگان، دارای تنوع بالا (حرکت گسترده تر در فضای جواب) و تشدید کم (توانایی یافتن جواب بهینه محلی) هستند. از سوی دیگر، جستجوی محلی، شدت بالایی دارد. بنابراین، الگوریتم سیستم مورچگان، هنگامی که با الگوریتم جستجوی محلی ترکیب می شود، جواب های بهتری را تولید

می کند. الگوریتم جستجوی محلی پیشنهادی، از دو حرکت همسایگی تشکیل شده است:

- **همسایگی دو تعویضی:** این عملگر همسایگی، رنگ رأس v را از i به j که در آن $j < i - 1$، تغییر داده و سپس رنگ تمام رئوس همسایه آن را از j به کوچکترین رنگ ممکن تغییر می دهد.
- **همسایگی کاهشی:** این عملگر همسایگی، رنگ رأس v را از i به j که در آن $j \leq i$ ، تغییر می دهد. این عملگر همسایگی، در صورت امکان، رنگ یک رأس را کاهش می دهد، در غیر این صورت، رنگ آن بدون تغییر باقی می ماند.

در الگوریتم جستجوی محلی، ابتدا همه همسایگی‌های دو تعویضی یک رأس محاسبه می‌شوند و بهترین همسایه‌ای که مجموع رنگی کمتری دارد انتخاب می‌شود و الگوریتم، به همسایگی جدید حرکت می کند. پس از اعمال این رویه، برای همه رئوس گراف، همسایگی کاهشی برای همه رئوس اعمال می شود.

۳- نتایج

برای بررسی کارایی عملی الگوریتم ارائه شده GSC، این الگوریتم، روی 38 گراف آزمایشی از ادبیات موضوع تست شده است. این کار برای مقایسه الگوریتم ارائه شده با الگوریتم های قبلی در ادبیات موضوع می باشد. الگوریتم ارائه شده GSC، که در بخش قبل تشریح شد، با زبان برنامه نویسی C کد نویسی شده است. نتایج اجرای الگوریتم، روی گراف های نمونه در کتابخانه

گراف های تصادفی DIMACS و COLOR02 در ادامه آورده شده است.

در جداول (۱) و (۲)، به ترتیب نتایج عددی اجرای الگوریتم روی گراف های کتابخانه DIMACS و COLOR02 خلاصه شده است. در این جداول، در ستون اول اسامی گراف های مورد بررسی، در ستون دوم تعداد رئوس، در ستون سوم تعداد یال ها، در ستون چهارم نتایج الگوریتم پیشنهادی GSC، و در ستون آخر نیز بهترین جواب یافته شده در ادبیات موضوع آورده شده است.

از بین ۳۸ گراف مورد بررسی، در ۱۳ مورد، جواب های جدیدی پیدا شده است که از بهترین جواب یافته شده توسط همه الگوریتم های قبلی بهتر است. همچنین در ۱۹ مورد، جواب الگوریتم با جواب بهترین جواب یافته شده توسط الگوریتم های قبلی، برابر است. در مجموع، از بین ۳۸ گراف مورد بررسی، در ۳۲ مورد از گراف ها، یا جواب جدیدی به دست آمده و یا اینکه همان بهترین جواب قبلی تکرار شده است. در جدولهای (۱) و (۲) این جوابها با فونت برجسته متمایز شده اند.

(جدول ۱–): نتایج اجرای الگوریتم GSC روی ۲۰ گراف از کتابخانه DIMACS

نمونه	$\|V\|$	$\|E\|$	Sum^*	GSC Sum
dsjc125.1	۱۲۵	۷۳۶	۳۲۶	**۳۲۶**
dsjc125.5	۱۲۵	۳۸۹۱	۱۰۱۵	**۱۰۱۲**
dsjc125.9	۱۲۵	۶۹۶۱	۲۵۱۱	**۲۵۰۳**
dsjc250.1	۲۵۰	۳۲۱۸	۹۷۷	**۹۷۴**
dsjc250.5	۲۵۰	۱۵۶۶۸	۳۲۴۶	**۳۲۴۰**
dsjc250.9	۲۵۰	۲۷۸۹۷	۸۲۸۹	۸۴۰۰
dsjc500.1	۵۰۰	۱۲۴۵۸	۲۸۵۰	۲۸۶۰
dsjc500.5	۵۰۰	۶۲۶۲۴	۱۰۹۱۰	۱۰۹۶۵
dsjc500.9	۵۰۰	۱۱۲۴۳۷	۲۹۹۱۲	۳۰۳۱۵
le450_15a	۴۵۰	۶۱۶۸	۲۶۳۲	**۲۶۲۲**
le450_15b	۴۵۰	۸۱۶۹	۲۶۴۲	**۲۶۳۲**
le450_15c	۴۵۰	۱۶۶۸۰	۳۸۶۶	**۳۵۵۲**
le450_15d	۴۵۰	۱۶۷۵۰	۳۹۲۱	**۳۵۵۲**
le450_25a	۴۵۰	۸۲۶۰	۳۱۵۳	**۳۱۴۶**
le450_25b	۴۵۰	۸۲۶۳	۳۳۶۶	**۳۳۵۲**
le450_25c	۴۵۰	۱۷۳۴۳	۴۵۱۵	**۴۵۱۲**
le450_25d	۳۰۰	۱۷۴۲۵	۴۵۴۴	**۴۵۲۲**
flat300_28_0	۳۰۰	۲۱۶۹۵	۴۲۸۲	**۴۲۷۳**
flat300_26_0	۳۰۰	۲۱۶۳۳	۳۹۶۶	۳۹۷۳
flat300_20_0	۳۰۰	۲۱۳۷۵	۳۱۵۰	**۳۱۵۰**

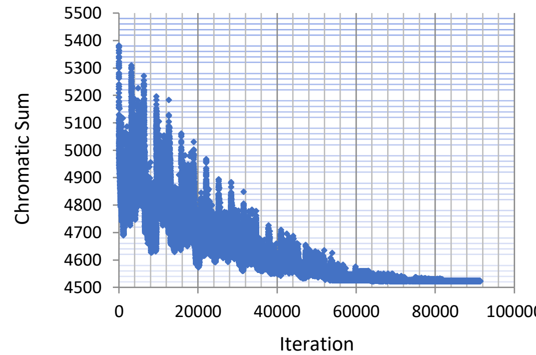
۴- بحث

علیرغم این واقعیت که الگوریتم GSC پیشنهادی ما از سه زیر الگوریتم تشکیل شده است، اما در یک زمان، اجرای معقول به جواب های بسیار خوبی می رسد، به این دلیل که زیر الگوریتم های پیشنهادی ما بسیار چابک هستند. علاوه بر این، ما زیر الگوریتم سیستم مورچگان GCANT را برای تعداد مشخصی از تکرارها اجرا می کنیم. به عبارت دیگر، به جای صرف زمان برای بهبود بی اهمیت در کیفیت جواب ها، الگوریتم را با زمان و کیفیت معقول اجرا می کنیم و سپس برخی از رئوس را از گراف حذف می کنیم و الگوریتم را روی گراف کوچکتری تکرار می کنیم. در این رویکرد، همگرایی الگوریتم را می توان به تعویق انداخت، اما می توان به بهبود قابل توجهی در مجموع رنگی دست یافت. روند همگرایی الگوریتم GSC در شکل (۱) ترسیم شده است.

یکی دیگر از ویژگی‌های مطلوب الگوریتم GSC این است که این الگوریتم می‌تواند از ایده رنگ‌آمیزی مجموعه‌های مستقل بزرگ با رنگ‌های کوچک استفاده کند و فضای راه حل را با کمک الگوریتم فراابتکاری سیستم مورچگان جستجو کند. همانطور که می دانیم، الگوریتم های فراابتکاری به طور سیستماتیک، فضای راه حل را برای یافتن راه حل های بهتر، جستجو می کنند.

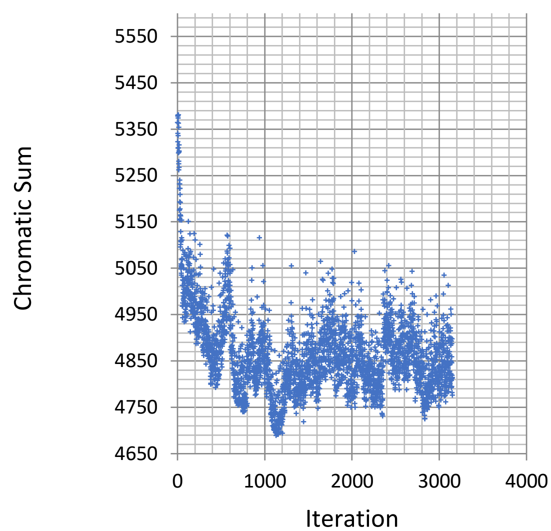
(جدول–۲): نتایج اجرای الگوریتم GSC روی ۱۸ گراف از کتابخانه COLOR02

نمونه	$\|V\|$	$\|E\|$	Sum^*	GSC Sum
anna	۱۳۸	۴۹۳	۲۷۶	۲۷۷
david	۸۷	۴۰۶	۲۳۷	**۲۳۷**
huck	۷۴	۳۰۱	۲۴۳	**۲۴۳**
jean	۸۰	۲۵۴	۲۱۷	**۲۱۷**
miles250	۱۲۸	۳۸۷	۳۲۵	**۳۲۵**
miles500	۱۲۸	۱۱۷۰	۷۰۹	**۷۰۸**
myciel3	۱۱	۲۰	۲۱	**۲۱**
myciel4	۲۳	۷۱	۴۵	**۴۵**
myciel5	۴۷	۲۳۶	۹۳	**۹۳**
myciel6	۹۵	۷۵۵	۱۸۹	**۱۸۹**
myciel7	۱۹۱	۲۳۶۰	۳۸۱	**۳۸۱**
queen5_5	۲۵	۱۶۰	۷۵	**۷۵**
queen6_6	۳۶	۲۹۰	۱۳۸	**۱۳۸**
queen7_7	۴۹	۴۷۶	۱۹۶	**۱۹۶**
queen8_8	۶۴	۷۲۸	۲۹۱	**۲۹۱**
games120	۱۲۰	۶۳۸	۴۴۳	**۴۴۳**
qg.order30	۹۰۰	۲۶۱۰۰	۱۳۹۵۰	**۱۳۹۵۰**
inithx.i.1	۸۶۴	۱۸۷۰۷	۳۶۷۶	**۳۶۷۶**

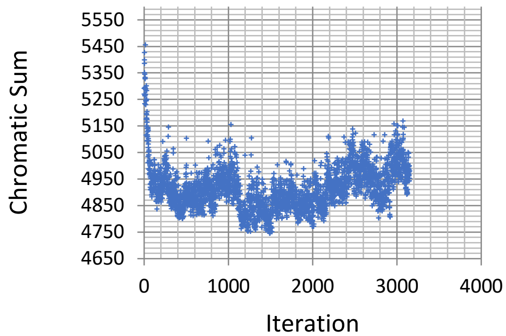


(شکل-۱): روند همگرایی الگوریتم GSC

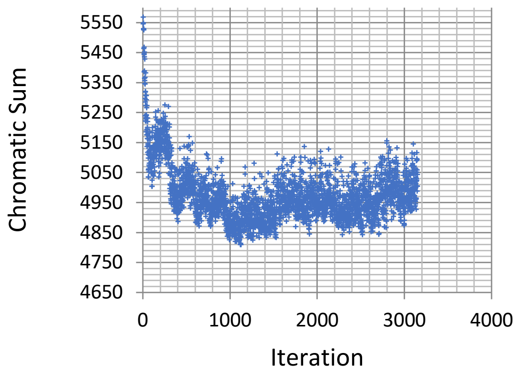
همانطور که در بخش ۲ ذکر شد، در الگوریتم GCANT، برای رئوس متعلق به بزرگترین مجموعه، فِرمون آزاد می شود. علاوه بر این، ما از اطلاعات هیوریستیکی و یک الگوریتم جستجوی محلی برای بهبود کارایی الگوریتم سیستم مورچگان استفاده کرده‌ایم. برای نشان دادن میزان کارآمدی این رویکرد، الگوریتم GCANT را بدون آنها اجرا می کنیم. نتایج را می توان در شکل (۲) تا شکل (۵) مشاهده کرد.



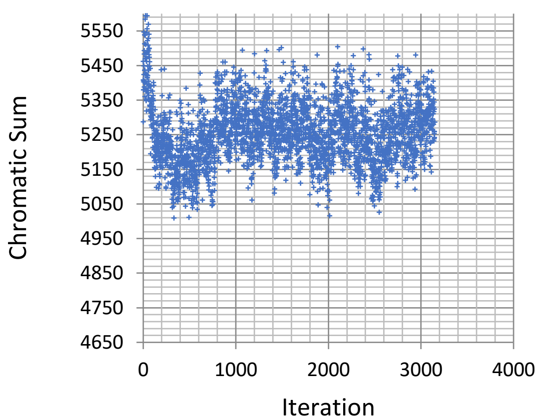
(شکل-۲): خروجی الگوریتم GCANT



(شکل-۳): خروجی الگوریتم GCANT بدون آزاد کردن فرمون برای بزرگترین مجموعه مستقل



(شکل-۳): خروجی الگوریتم GCANT بدون استفاده از اطلاعات هیوریستیکی



(شکل-۵): خروجی الگوریتم GCANT بدون استفاده از الگوریتم جستجوی محلی

۵- جمع بندی

در این تحقیق یک الگوریتم ابتکاری برای مسئله رنگ آمیزی جمعی گراف ارائه شد. ورودی این الگوریتم یک گراف دلخواه و خروجی آن نیز یک رنگ آمیزی رأسی گراف است که در آن، مجموع رنگی تا حد ممکن کمینه شده است. عملکرد این الگوریتم ابتکاری، روی گراف های تصادفی کتابخانه DIMACS و COLOR02 آزمایش شد و نتایج آن با الگوریتم های ادبیات موضوع مقایسه شد. با توجه به نتایج عددی، نشان داده شد که الگوریتم ارائه شده دارای کارایی بسیار خوبی است و در اکثر موارد به نتایج خوبی رسیده است.

با توجه به پیچیدگی فضای جواب مسئله، بعید به نظر می رسد که بتوان با استفاده از الگوریتم های فراابتکاری به نتایج بسیار بهتری دست یافت. پیشنهاد می شود که برای کارهای آینده، از الگوریتم های ترکیبی استفاده شود که عناصر زیر را شامل شود:

- شناسایی و استفاده از یک ویژگی کارای مربوط به ساختار جواب بهینه. شناسایی و استفاده از بزرگترین مجموعه مستقل رئوس گراف، کاری در این راستا بود که در این تحقیق انجام شد. ممکن است ویژگی های دیگری در ساختار جواب های بهینه یا زیر بهینه وجود داشته باشد. بایستی این ویژگی ها شناسایی شده و الگوریتم به سمت جواب هایی هدایت شود که دارای این ویژگی ها هستند. این امر، باعث افزایش کارایی و هدفمندی جستجو می شود.
- از یک یا دو الگوریتم فراابتکاری به عنوان هسته اصلی الگوریتم استفاده شود.
- در صورت امکان، از تکنیک های برنامه ریزی ریاضی از جمله برنامه ریزی عدد صحیح یا برنامه ریزی محدودیت برای افزایش عمق جستجو استفاده شود.

۶- منابع

[1]. *Kubicka, E. and A.J. Schwenk. An introduction to chromatic sums. in Proceedings of the 17th conference on acm annual computer science conference. 1989.*

[2]. *Kubicka, E., The chromatic sum of a graph: History and recent developments. International Journal of Mathematics and Mathematical Sciences, 2004. 2004(30): p. 1563-1573.*

[3]. *Kokosiński, Z. and K. Kwarciany. On sum coloring of graphs with parallel genetic algorithms. in International Conference on Adaptive and Natural Computing Algorithms. 2007. Springer.*

[4]. *Li, Y., et al. Greedy algorithms for the minimum sum coloring problem. in Logistique et transports. 2009.*

[5]. *Bouziri, H. and M. Jouini, A tabu search approach for the sum coloring problem. Electronic Notes in Discrete Mathematics, 2010. 36: p. 915-922.*

[6]. *Helmar, A. and M. Chiarandini. A local search heuristic for chromatic sum. in Proceedings of the 9th metaheuristics international conference. 2011.*

[7]. *Douiri, S.M. and S. Elbernoussi, New algorithm for the sum coloring problem. International Journal of Contemporary Mathematical Sciences, 2011. 6(10): p. 453-463.*

[8]. *Leighton, F.T., A graph coloring algorithm for large scheduling problems. Journal of research of the national bureau of standards, 1979. 84(6): p. 489-506.*

[9]. *Bar-Noy, A., et al., On chromatic sums and distributed resource allocation. Information and Computation, 1998. 140(2): p. 183-202.*

[10]. *Wu, Q. and J.-K. Hao, An effective heuristic algorithm for sum coloring of graphs. Computers & Operations Research, 2012. 39(7): p. 1593-1600.*

[11]. *Wu, Q., et al., Minimum sum coloring for large graphs with extraction and backward expansion search. Applied Soft Computing, 2018. 62: p. 1056-1065.*

[12]. *Jin, Y., J.-P. Hamiez, and J.-K. Hao, Algorithms for the minimum sum coloring problem: a review. Artificial Intelligence Review, 2017. 47(3): p. 367-394.*

[13]. *Moukrim, A., et al., Lower bounds for the minimal sum coloring problem. Electronic Notes in Discrete Mathematics, 2010. 36: p. 663-670.*

[14]. *Lin, W., et al., Computing lower bounds for minimum sum coloring and optimum cost chromatic partition. Computers & Operations Research, 2019. 109: p. 263-272.*

[15]. *Mrad, M., et al., A column generation-based lower bound for the minimum sum coloring problem. IEEE Access, 2020. 8: p. 57891-57904.*

[16]. *Delle Donne, D ,.et al., A branch-and-price algorithm for the minimum sum coloring*

problem. Discrete Applied Mathematics, 2021. 303: p. 39-56.

مرتضی رجب زاده مدرک کارشناسی خود را در رشته مهندسی صنایع در سال ۱۳۸۱ از دانشگاه شمال مازندران، مدرک کارشناسی ارشد خود را در رشته مهندسی صنایع در سال ۱۳۸۳ از دانشگاه ملی هوافضای خارکُف اوکراین و مدرک دکترا را در رشته مهندسی صنایع در سال ۱۳۹۲ از دانشگاه دولتی سُومی اوکراین اخذ کرده است. ایشان در حال حاضر عضو هیئت علمی مرکز آموزش عالی محلات می باشد. زمینه پژوهشی مورد علاقه ایشان عبارتند از: مدیریت کیفیت، مدیریت بهره وری، مدیریت ریسک، استانداردسازی و سیستم های مدیریت یکپارچه می باشد.

نشانه رایانامه ایشان عبارتند از:

rajabzadeh.m@gmail.com

امین محمدنژاد مدرک کارشناسی خود را در رشته مهندسی صنایع در سال ۱۳۸۸ از دانشگاه علم و صنعت ایران، مدرک کارشناسی ارشد خود را در رشته مهندسی صنایع در سال ۱۳۹۱ از دانشگاه صنعتی شریف و مدرک دکترا را در رشته مهندسی صنایع در سال ۱۴۰۰ از دانشگاه علم و صنعت ایران اخذ کرده است. ایشان در حال حاضر تحلیل گر مالی در شرکت های خصوصی می باشد. زمینه پژوهشی مورد علاقه ایشان عبارتند از: رنگ آمیزی جمعی گراف، بهینه سازی استوار، بهینه سازی تصادفی، ارزش گذاری سرمایه گذاری می باشد.

نشانه رایانامه ایشان عبارتند از:

mohammadnejad.daryani@gmail.com

کوروش عشقی مدرک کارشناسی خود را در رشته مهندسی صنایع در سال ۱۳۶۵ از دانشگاه صنعتی امیرکبیر، مدرک کارشناسی ارشد خود را در رشته مهندسی صنایع در سال ۱۳۷۱ از دانشگاه صنعتی شریف و مدرک دکترا را در رشته مهندسی صنایع در سال ۱۳۷۶ از دانشگاه تورنتو کانادا اخذ کرده است. ایشان در حال حاضر عضو هیئت علمی دانشگاه صنعتی شریف می باشد. زمینه پژوهشی مورد علاقه ایشان عبارتند از: نظریهٔ گراف، الگوریتم های متاهیورستیک، برنامه ریزی عدد صحیح می باشد.

نشانه رایانامه ایشان عبارتند از:

eshghi@sharif.edu

روش ارجاع به مقاله : م. رجب زاده، ا. محمدنژاد، ک. عشقی حل مسأله رنگ آمیزی جمعی گراف با استفاده از الگوریتم ابتکاری. دو فصلنامه و محاسبات و سامانه های توزیع شده. سال پنجم، شماره اول، شماره پیاپی ۹، صفحه ۲۲ تا ۳۰، سال۱۴۰۱.

How to cite: Morteza Rajabzadeh, Amin Mohammadnejad, Kourosh Eshghi. Solving the Graph Sum Colouring Problem Using a Heuristic Algorithm**.** Journal of Distributed Computing and Systems(JDCS), Vol 5, Issue 1, Page 22-30, 2022.