

الگوریتم مبتنی بر خوشه بندی سلسله مراتبی برای رنگ آمیزی گراف ها

روح اله اعتمادی^۱؛ نصراله مقدم چرکری^۲

چکیده

رنگ آمیزی گراف عبارت است از برچسب گذاری هر راس بطوری که رئوس مجاور برچسب یکسانی نداشته باشند. و مینیمم رنگ آمیزی برای یک گراف بکار گیری کمترین برچسب ممکن می باشد. مسئله رنگ آمیزی گراف با کمترین رنگ ممکن بعنوان یک مسئله NP-Hard شناخته شده است. در این مقاله الگوریتم جدیدی مبتنی بر خوشه بندی سلسله مراتبی برای حل مسئله رنگ آمیزی گراف با حداقل رنگ ممکن ارائه شده است. نتایج تجربی بدست آمده از اجرای الگوریتم پیشنهادی روی گراف های تست DIMACS نشان دهنده کارایی الگوریتم پیشنهادی نسبت به الگوریتم های مورد مقایسه می باشد.

کلمات کلیدی

رنگ آمیزی گراف - خوشه بندی - داده کاوی .

A Hierarchical Clustering Based Algorithm for coloring graphs

R.etemadi , N.mogaddan charkari

ABSTRACT

Coloring a graph involves labeling each node so that adjacent nodes have different labels. A minimum coloring of a graph is a coloring that uses as few different labels as possible. The problem of coloring graph with the minimum number of colors is well known to be NP-Hard. In this paper, a new algorithm based on Hierarchical clustering is presented for the graph coloring problem with the minimum number of colors. Experimental results on DIMACS graphs Benchmark instances show improvements over existing algorithms for the graph coloring problem.

KEYWORDS

Graph Coloring-Clustering-Data Mining.

۱. مقدمه

مسئله k -رنگ آمیزی گراف عبارت است از تخصیص یک رنگ از k رنگ مختلف به هر یک از رئوس گراف بطوری که هیچ دو زوج راس مجاور رنگ یکسانی نداشته باشند. عدد رنگی گراف نیز کوچکترین مقدار k می باشد که بتوان با آن تعداد رنگ، گراف مورد نظر را رنگ آمیزی کرد ، و آن را با $\chi(G)$ نمایش می دهند. از دیدگاه الگوریتمی ، روشن است که مسئله رنگ آمیزی گراف با کمترین تعداد رنگ ممکن یک مسئله NP-Hard و دشوار می باشد [۲]، [۴]، [۳]. از مهمترین کاربردهای رنگ آمیزی گراف تخصیص ثبات در کامپایلرها [۷]، [۶]، [۵]، مسائل زمانبندی [۸]، [۹] و دسوار می باشد [۱۱] و الگوریتم های بلام [۲]، کارگر [۴] و نیز هالپرین [۱۲]، ANTCOL [۲۳] و بزرگ زاده [۱] و MMGC [۲۵] و ABAC [۲۶] از جمله آنهاست.

^۱ مربی ، دانشگاه آزاد اسلامی واحد بناب، دانشکده برق و کامپیوتر ، Roohalahetemadi@gmail.com

^۲ استاد یار، دانشگاه تربیت مدرس، دانشکده فنی مهندسی ، Charkari@modarres.ac.ir

مقاله حاضر در هشت بخش تنظیم شده است. در بخش ۲ تعریف مسئله رنگ آمیزی گراف آورده شده است. در بخش ۳ و ۴ مفهوم خوشه بندی و ارتباط آن را با مسئله رنگ آمیزی گراف ذکر گردیده است. در بخش ۵ الگوریتم پیشنهادی توضیح داده شده است. نتایج ارزیابی و نتیجه گیری نیز در بخش های ۶ و ۷ ارائه شده است.

۲. تعریف مسئله

گراف غیر جهدار $G = (V, E)$ که در آن V مجموعه رئوس و E مجموعه یالها می باشد را در نظر بگیرید. K -رنگ آمیزی گراف G عبارت است از نگاشت رئوس گراف G به K مجموعه بصورت $c: V \rightarrow \{1, 2, 3, \dots, K\}$ که در آن $\forall (u, v) \in E: c(u) \neq c(v)$ باشد و یا به عبارت دیگر رئوسی از گراف G که با هم مجاور باشند رنگ یکسانی نداشته باشند. عدد رنگی گراف نیز کوچکترین مقدار k می باشد که بتوان با آن تعداد رنگ گراف مورد نظر را رنگ کرد، و آن را با $\chi(G)$ نمایش می دهند. پر واضح است که تعداد حالت هایی که برای k -رنگ آمیزی یک گراف با n راس باید بررسی شود برابر است با $\frac{k^{n+1} - 1}{k - 1}$ و از دیدگاه الگوریتمی، روشن است که مسئله رنگ آمیزی گراف با کمترین تعداد رنگ یک مسئله NP-Hard و دشوار می باشد [۲]، [۴]، [۳]. پس هدف ما در این مقاله ارائه راهکاری ابتکاری و البته تقریبی برای رنگ آمیزی گراف G با کمترین رنگ ممکن و هزینه زمانی کمتر می باشد.

۳. خوشه بندی

فرایند گروه بندی مجموعه ای از اشیاء فیزیکی یا انتزاعی به دسته های شبیه به هم، خوشه بندی نامیده می شود و هر یک از گروه های ایجاد شده را یک خوشه می نامند. اشیاء موجود در داخل یک خوشه بیشترین شباهت را با هم دیگر داشته و با اشیاء گروه های بعدی بیشترین تفاوت را دارند [۱۳]، [۱۴].

تعریف: مجموعه $X = \{x_1, x_2, \dots, x_n\}$ شامل n شی را در نظر بگیرید، هدف از خوشه بندی گروه بندی اشیاء در k خوشه بصورت $C = \{C_1, C_2, \dots, C_k\}$ می باشد بطوری که هر خوشه دارای شرایط زیر می باشد:

- 1) $C_1 \cup C_2 \cup \dots \cup C_k = X$
- 2) $C_i \neq \emptyset \quad i = 1 \dots k$
- 3) $C_i \cap C_j = \emptyset$

با توجه به تعریف بالا تعداد حالت های مختلف برای خوشه بندی n شی به k خوشه برابر است با: [۱۵]

$$NW(n, k) = \frac{1}{k!} \sum_{i=0}^k (-1)^i \binom{k}{i} (k-i)^n \quad (1)$$

در بیشتر روشها مقدار تعداد خوشه ها یعنی k توسط کاربر تعیین می شود. از رابطه (۱) می توان فهمید که حتی اگر k معلوم باشد پیدا کردن بهترین حالت خوشه بندی کار ساده ای نیست. همچنین تعداد راه های خوشه بندی n شی به K خوشه تقریباً بصورت $k^n/k!$ افزایش می یابد بنابراین مسئله یافتن بهترین حالت برای خوشه بندی n شی به k خوشه، یک مسئله NP-Complete و بغرنج بشمار می آید و باید توسط تکنیک هایی بصورت پهنه حل گردد [۱۶].

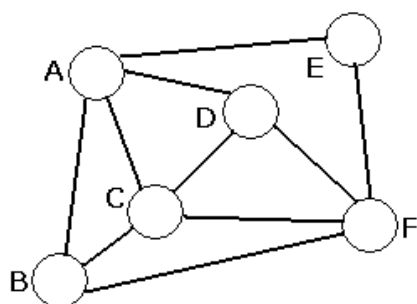
۴. ارتباط خوشه بندی با k -رنگ آمیزی گراف

با توجه به تعاریف ارائه شده در قسمت ۲ و ۳ برای k -رنگ آمیزی گراف و خوشه بندی می توان به این نتیجه رسید که k -رنگ آمیزی گراف به راحتی قابل تبدیل به مسئله خوشه بندی می باشد. و می توان از الگوریتم ها و تکنیک هایی موجود برای خوشه بندی اشیاء برای رنگ آمیزی گراف استفاده نمود.

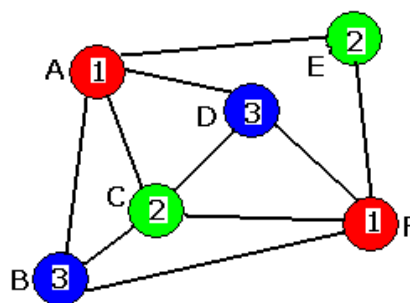
با توجه به مفهوم خوشه بندی می توان مسئله k -رنگ آمیزی گراف را بصورت زیر تعریف کرد:

تعریف k-رنگ آمیزی گراف $G(V, E)$ با n راس بر اساس مفهوم خوشه بندی : مجموعه $V = \{v_1, v_2, \dots, v_n\}$ شامل n راس را در نظر بگیرید ، هدف از خوشه بندی گروه بندی اشیاء موجود در مجموعه V در k خوشه بصورت $C = \{c_1, c_2, \dots, c_k\}$ می باشد بطوری که هر خوشه دارای شرایط زیر می باشد:

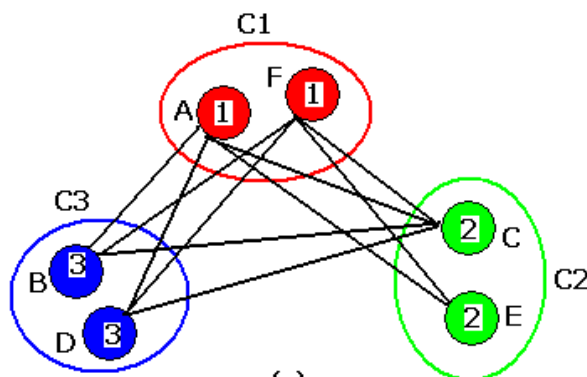
- 1) $C_1 \cup C_2 \cup \dots \cup C_k = V$
- 2) $C_i \neq \emptyset \quad i = 1, 2, \dots, k$
- 3) $C_i \cap C_j = \emptyset$
- 4) $\forall u, v \in C_i : (u, v) \notin E, 1 \leq i \leq k$



(a)



(b)



(c)

شکل (۱): ارتباط بین رنگ آمیزی گراف و خوشه بندی

برای مثال می خواهیم گراف شکل (۱-a) را رنگ آمیزی نمائیم. در شکل (۱-b) یک رنگ آمیزی ممکن با سه رنگ ۱، ۲، ۳ دیده می شود. با توجه به تعریف بالا برای رنگ آمیزی گراف بر اساس خوشه بندی می توان گراف مورد نظر را به سه خوشه مجزا خوشه بندی کرد و بعد از خوشه بندی برای هر خوشه یک رنگ دلخواه را انتخاب کرد (شکل ۱-c).

با توجه به تعریف بالا با تعیین k خوشه از رئوس و رنگ آمیزی هر خوشه با یک رنگ می توان مسئله رنگ آمیزی گراف را حل کرد. حال الگوریتم های زیادی برای خوشه بندی اشیاء ارائه داده شده است. یکی از فرایندهای مهم کشف الگو ، خوشه بندی می باشد، که الگوریتم ها و تکنیک های مختلفی در زمینه داده کاوی را می پوشانند. بطور کلی الگوریتم ها و تکنیک های خوشه بندی ارائه شده را می توان به پنج دسته تقسیم بندی کرد: روشهای پارتیشن بندی، روشهای سلسله مراتبی ، روشهای مبتنی بر تراکم ، روشهای مبتنی بر Grid و روشهای مبتنی بر مدل. که برای آشنایی با این الگوریتم ها می توان به منابع [۱۳]، [۱۴]، [۱۵]، [۱۶]، [۱۷]، [۱۸] ، [۱۹] ، [۲۰]، [۲۱]، [۲۲] مراجعه کرد ، که توصیف هر یک از آنها خارج از مجال این بحث بوده ولی هر کدام راه کار بهینه ای را برای حل مسئله خوشه بندی پیشنهاد داده اند.

۵. ارائه الگوریتم پیشنهادی

با فرض گراف غیر جهت دار $G(V, E)$ که در آن V مجموعه رئوس و E مجموعه یالها می باشد، تعاریف زیر را قبل از ارائه الگوریتم پیشنهادی در نظر می گیریم:

تعریف ۵-۱- مجموعه رئوس هم رنگ کاندید یک راس: برای یک راس دلخواه مثل v تعریف می شود و عبارت است از تمامی رئوسی از گراف G که با راس v مجاور نیستند.

تعریف ۵-۲- معیار تشابه گره ها: برای یک گره دلخواه v با گره دلخواه دیگر u تعریف می شود و عبارت است از تعداد رئوسی از مجموعه رئوس هم رنگ کاندید راس u که با گره v مجاور نیستند.

تعریف ۵-۳- مجموعه خوشه های هم رنگ کاندید یک خوشه: برای خوشه مفروض C تعریف می شود و عبارت است از مجموعه تمام خوشه هایی مثل $SC = \{C_1, C_2, \dots, C_t\}$ که در تعریف زیر صدق کند:

$$\forall v \in C_i, \forall u \in C : edge(v, u) \notin E, 1 \leq i \leq t, C_i \in SC$$

و یا به عبارت دیگر هیچ راسی از خوشه C با هیچ کدام از رئوس موجود در خوشه های $SC = \{C_1, C_2, \dots, C_t\}$ مجاور نباشد.

تعریف ۵-۴- برخورد: می گوئیم خوشه C_1 با خوشه C_2 برخورد دارد اگر حداقل یک راس در خوشه C_2 موجود باشد که با یکی از رئوس خوشه C_1 مجاور باشد.

تعریف ۵-۵- معیار تشابه خوشه ها: بین دو خوشه مثل خوشه C_1 با C_2 تعریف می شود و عبارت است از تعداد خوشه از مجموعه خوشه های هم رنگ کاندید برای خوشه C_2 که با خوشه C_1 برخورد ندارند.

✓ الگوریتم پیشنهادی:

با توجه به تعاریف بالا الگوریتم پیشنهادی برای رنگ آمیزی گراف G که بر اساس الگوریتم های سلسله مراتبی خوشه بندی ارائه می شود. همانطور که در شکل (۲) دیده می شود الگوریتم پیشنهادی دارای مراحل زیر می باشد.

- ۱- هر کدام از رئوس گراف G را بعنوان یک خوشه در نظر بگیر
- ۲- تا رسیدن به تعداد معین خوشه ها و یا عدم امکان کاهش تعداد خوشه ها مرحله زیر را تکرار کن
- ۲-۱- عدد تصادفی k را بین ۱ و تعداد خوشه ها ایجاد کن.
- ۲-۲- مجموعه خوشه های هم رنگ کاندید خوشه k ام را پیدا کن.
- ۲-۳- برای تمامی خوشه های موجود در مجموعه هم رنگ کاندید معیار تشابه با خوشه k ام را محاسبه کن.
- ۲-۴- خوشه ای که بیشترین تشابه را با خوشه k ام دارد، با خوشه k ام ترکیب کن.

اگر در الگوریتم ارائه شده دقت شود، در هر مرحله خوشه هایی را که بیشترین تشابه را با هم دارند ترکیب می کنیم. اگر به معیار تشابه خوشه ها دقت شود، واضح است که هدف از این معیار تشابه این است که خوشه هایی برای ترکیب در هر مرحله انتخاب شوند که خوشه حاصل از ترکیب، کمترین برخورد را با اجتماع مجموعه های هم رنگ کاندید هر دو خوشه داشته باشد تا به کمترین تعداد خوشه ها در الگوریتم خوشه بندی دست یابیم. که نتایج حاصل بیانگر این مطلب است.

۶. نتایج تجربی

الگوریتم ارائه شده با زبان برنامه سازی C#.net کد شده و روی سیستم کامپیوتری پنتیوم ۴ با پردازنده ۱.۷ GHz و حافظه اصلی با ظرفیت ۲۵۶MB اجرا شده است. برای ارزیابی الگوریتم پیشنهادی از گراف های استاندارد DIMACS [۲۴] استفاده شده است. برای ارزیابی الگوریتم های رنگ آمیزی گراف، بیشتر محققان از گراف های تست مسابقه DIMACS سال ۲۰۰۲ استفاده می کنند. البته اغلب این گرافها قبل از آن نیز در دسته های مختلفی معرفی شده بوده اند. برخی از گراف های این مجموعه بصورت تصادفی و با استفاده از برنامه های تولید کننده گراف ساخته شده اند مانند سری DSJ از David Johnson و برخی نیز از مسائل کاربردی نتیجه می شوند، بعنوان مثال گراف های سری SCH از Gray و Lewandaski که مربوط به مسئله زمانبندی کلاس در هر یک از حالات بدون فضای خالی و با فضای خالی است و یا گراف های سری LAT، که مربوط به مساله مربع لاتین می باشد. همچنین در این مجموعه سری REG بر مبنای مساله تخصیص ثبات در کامپایلر ها ایجاد شده است [۱]. مشخصات بعضی از این گراف های آزمون که در این مقاله استفاده شده، در جدول (۱) آورده شده است.

INPUT: $V = \{v_1, v_2, \dots, v_n\}$ (set of vertices Graph $G(V, E)$)

OUTPUT: $C = \{c_1, c_2, \dots, c_k\} (\forall (u, v) \in E, u \in C_i, v \in C_j \Rightarrow i \neq j, 1 \leq i, j \leq k)$

```

set ofCluster KGCBC()
{
    create n cluster and assign  $n v_i \rightarrow C_i$ 
    while(can merge clusters)
    {
         $k = \text{randomNumber}(1, \text{countOfcluster})$ ;
        setOfCluster SCK = find_candidate_sameColor( $C_k$ );
        IndexOfCluster h = select_cluster_withMaxSimalerity(SCK);
        Merge_Cluster( $k, h$ );
    }
    Return SetofCluster;
}

```

شکل (۲): الگوریتم پیشنهادی برای رنگ آمیزی گراف مبتنی بر الگوریتم خوشه بندی سلسله مراتبی

جدول (۱): مشخصات گراف های آزمون استفاده شده

نام گراف نمونه	تعداد رئوس	تعداد یالها	عدد رنگی
DSJ۱۲۵.۱	۱۲۵	۷۳۶	۵
DSJ۱۲۵.۵	۱۲۵	۳۸۹۱	۱۷
Le۴۵۰_۵a	۴۵۰	۵۷۱۴	۵
Le۴۵۰_۲۵a	۴۵۰	۸۲۶۰	۲۵
School۱	۳۸۵	۱۹۰۹۵	۱۴
School۱_nsh	۳۵۲	۱۴۶۱۲	۱۴
DSJC۲۵۰.۰.۱	۲۵۰	۳۲۱۸	۸
Le۴۵۰_۲۵c	۴۵۰	۱۷۳۴۳	۲۵
DSJC۵۰۰.۰.۱	۵۰۰	۳۵۵۵	۱۲
Queen۱۵_۱۵	۲۲۵	۱۰۳۶۰	۱۵

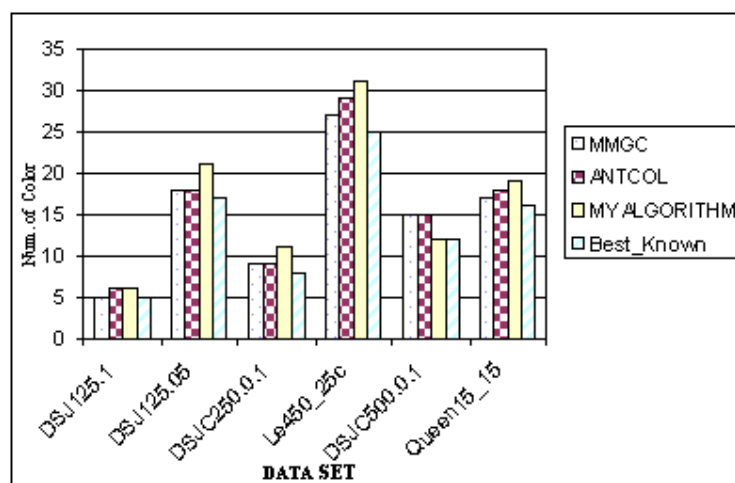
نتایج بدست آمده از الگوریتم پیشنهادی را با الگوریتم های EA_{1_a}, EA_{2_a}, EA₃, EA_{1_b} و EA_{2_b} از منبع [۱] و با الگوریتم های ANTCOL [۲۳] و الگوریتم MMGC [۲۵] و الگوریتم ABAC [۲۶] مورد مقایسه قرار گرفته و بصورت جداول (۲) و (۳) و (۴) گزارش شده است. لازم به توضیح است که در جدول (۲) نتایج بدست آمده از اجرای الگوریتم های MMGC و ANTCOL روی سیستم کامپیوتری پنتیوم ۴ با پردازنده ۱.۸GHz و حافظه اصلی ۲۵۶MB که در منبع [۲۳] گزارش شده است، استفاده گردیده است. در جدول (۴) نتایج حاصل از پیاده سازی الگوریتم ABAC با C++ و اجرای آن بر روی سیستمی کامپیوتر پنتیوم ۴ با پردازنده ۳.۲GHz و ۱GB Ram و سیستم عامل LINUX و پیاده سازی الگوریتم پیشنهادی با C#.net۲۰۰۵ و اجرای آن بر روی کامپیوتر پنتیوم ۴ با پردازنده ۳.۰۸GHz و ۵۱۲MB Ram و سیستم عامل XP گزارش شده است.

جدول (۲) نتایج اجرای الگوریتم پیشنهادی و الگوریتم های EA_{1_a}, EA_{2_a}, EA₃, EA_{1_b} و EA_{2_b}

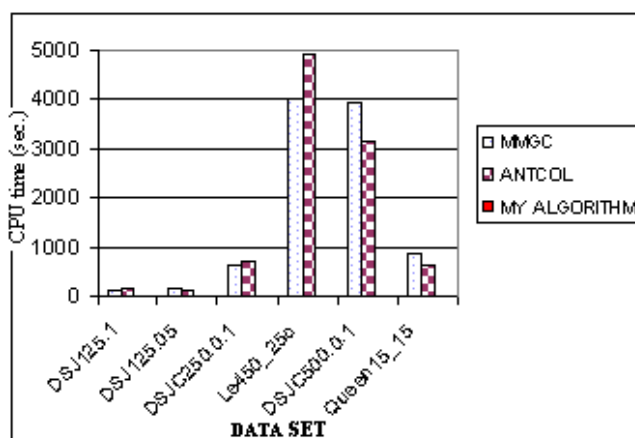
Algorithms							
Data SET	Best - Known	EA _{1_a}	EA _{2_a}	EA ₃	EA _{1_b}	EA _{2_b}	My Algorithm
DSJ۱۲۵.۱	۵	۷	۷	۷	۶	۶	۶
DSJ۱۲۵.۵	۱۷	۲۱	۲۲	۲۲	۲۲	۲۲	۲۱
Le۴۵۰_۵a	۵	۱۳	۱۳	۱۲	۱۳	۱۰	۱۰
Le۴۵۰_۲۵a	۲۵	۲۷	۲۷	۲۷	۲۶	۲۶	۲۵
School۱	۱۴	۳۹	۳۶	۳۶	۳۲	۲۶	۲۰
School۱_nsh	۱۴	۳۵	۳۶	۳۵	۳۴	۲۵	۲۸

جدول (۳) نتایج اجرای الگوریتم پیشنهادی و الگوریتم های MMGC و ANTCOL

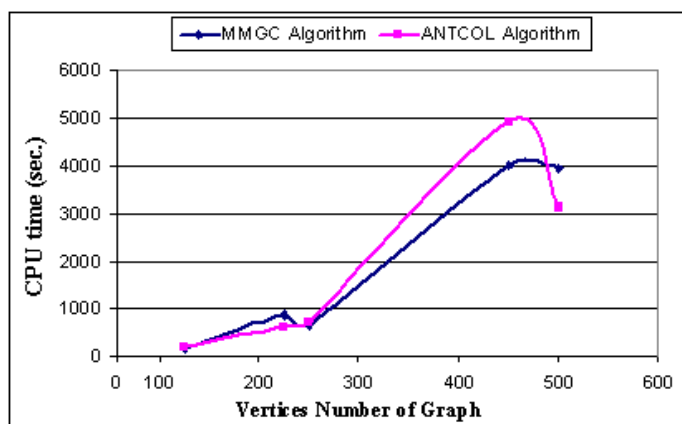
Algorithms							
Data SET	Best - Known	MMGC		ANTCOL		My Algorithm	
		K	CPU time(sec.)	k	CPU time(sec.)	k	CPU time(sec.)
DSJ۱۲۵.۱	۵	۵	۱۵۵	۶	۱۷۳	۶	۰.۴
DSJ۱۲۵.۰۵	۱۷	۱۸	۱۸۶	۱۸	۱۲۵	۲۱	۰.۴
DSJC۲۵۰.۰.۱	۸	۹	۶۵۰	۹	۷۱۳	۱۱	۲
Le۴۵۰_۲۵c	۲۵	۲۷	۴۰۰.۲	۲۹	۴۹۱۸	۳۱	۶.۲
DSJC۵۰۰.۰.۱	۱۲	۱۵	۳۹۴۲	۱۵	۳۱۲۵	۱۲	۵.۴
Queen۱۵_۱۵	۱۶	۱۷	۸۴۵	۱۸	۶۲۴	۱۹	۱.۸



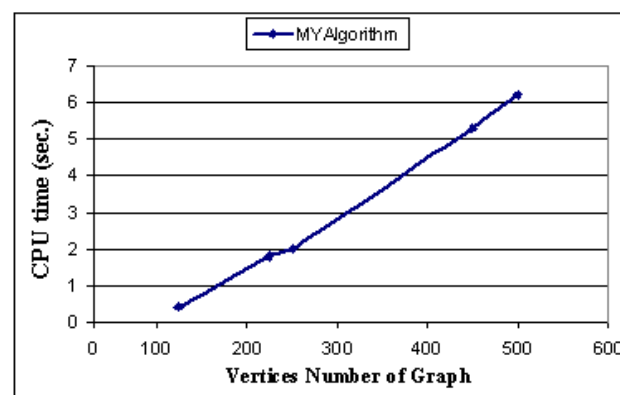
نمودار (۱) : مقایسه تعداد رنگ های بدست آمده حاصل از اجرای الگوریتم پیشنهادی و الگوریتم های MMGC و ANTCOL روی گراف های نمونه



نمودار (۲) : مقایسه اجرای الگوریتم پیشنهادی و الگوریتم های MMGC و ANTCOL روی گراف های نمونه بر اساس زمان اجرا (توجه شود زمان اجرای الگوریتم پیشنهادی بعلت ناچیز بودن در مقایسه با الگوریتم های دیگر در نمودار دیده نمی شود و تقریبا قابل چشم پوشی است)



(a)

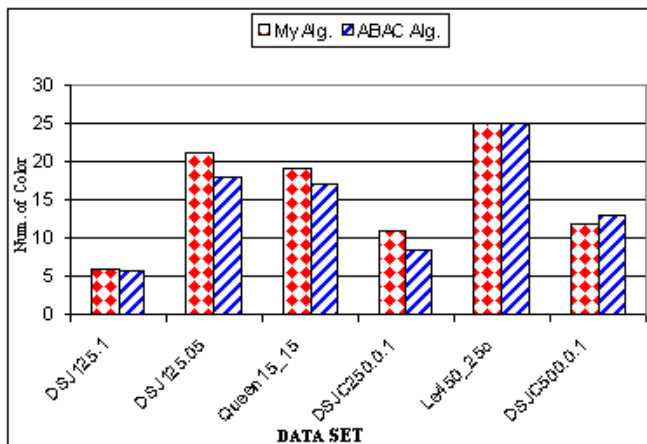


(b)

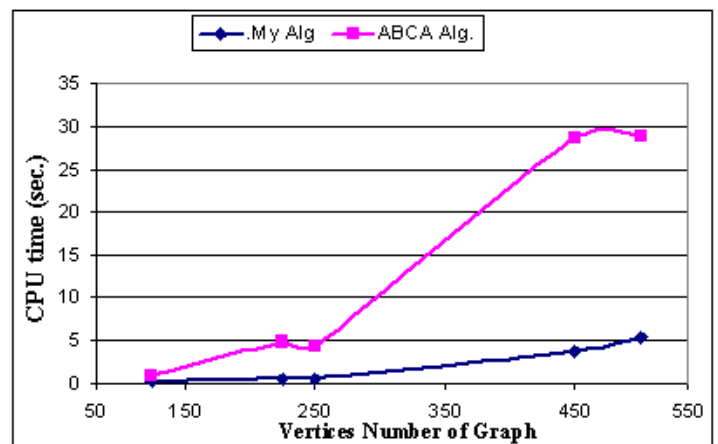
نمودار (۳) : نتایج حاصل از تغییرات زمان اجرای الگوریتم پیشنهادی و الگوریتم های MMGC و ANTCOL بر اساس افزایش تعداد رئوس گراف ها

جدول (۴) نتایج اجرای الگوریتم پیشنهادی و الگوریتم ABAC

Data SET	Best - Known	Algorithms			
		My Algorithm		ABAC (۵۰ runs of ABAC on each instance)	
		K	CPU time(sec.)	Avg. k	Avg. time(sec.)
DSJ۱۲۵.۱	۵	۶	۰.۱	۵.۷	۰.۹۲
DSJ۱۲۵.۰۵	۱۷	۲۱	۰.۱	۱۷.۸	۱.۶۹
Queen۱۵_۱۵	۱۶	۱۹	۰.۵	۱۷	۴.۹
DSJC۲۵۰۰.۱	۸	۱۱	۰.۶	۸.۵	۴.۳۳
Le۴۵۰_۲۵c	۲۵	۲۵	۳.۸	۲۵	۲۸.۷۱
DSJC۵۰۰۰.۱	۱۲	۱۲	۵.۴	۱۳	۲۸.۹۲



(a)



(b)

نمودار (۴) : نتایج حاصل از تغییرات زمان اجرا و تعداد رنگ های حاصل از اجرای الگوریتم پیشنهادی و الگوریتم ABAC

۷. نتیجه گیری

در این مقاله الگوریتم جدیدی برای رنگ آمیزی گراف مبتنی بر خوشه بندی سلسله مراتبی ارائه داده شده است. نتایج اجرای الگوریتم پیشنهادی و مقایسه آن با الگوریتم های انتخابی در جداول (۲) و (۳) و (۴) و نمودار های (۱)، (۲)، (۳) و (۴) ارائه شده است. همانطور که از نتایج ارائه شده مشخص است، کارایی الگوریتم پیشنهادی از نظر زمان خیلی قابل توجه می باشد. گرچه الگوریتم پیشنهادی در یک کامپیوتر مشابه با پردازنده کندتر (۱.۷GHz نسبت به ۱.۸ GHz و ۳.۰۸GHz نسبت به ۳.۲GHz) اجرا شده است ولی با در نظر گرفتن زمانهای بدست آمده از اجرای الگوریتم پیشنهادی روی گراف های استاندارد انتخاب شده و مقایسه آن با زمان اجرای الگوریتم های انتخابی اختلاف خیلی فاحشی را بدست می دهد (نمودارهای ۲، ۳ و ۴). اگر به نمودار ۱ و جداول (۲) و (۳) نیز دقت شود الگوریتم پیشنهادی برای رنگ آمیزی گراف بهتر عمل می کند و این نیز می تواند دلیل دیگری بر کارا بودن الگوریتم پیشنهادی نسبت به الگوریتم های انتخابی برای مقایسه باشد. مزیت دیگر الگوریتم پیشنهادی این است که با در نظر گرفتن گرافهایی با تعداد رئوس بیشتر (با ۵۰۰ راس) کارایی الگوریتم تغییر نمی کند و حتی در بعضی نمونه ها

بهتر عمل می کند. در نمودار شکل (۳) زمان اجرای الگوریتم ها را بر اساس افزایش گره ها نشان داده ایم. به دلیل اختلاف زیاد زمان اجرای الگوریتم پیشنهادی با الگوریتم های مورد مقایسه آن را در نمودار جداگانه ای (شکل ۳-b) نمایش داده ایم. همانطور که در نمودار ها دیده می شود تغییرات زمان اجرای الگوریتم های MMGC و ANTCOL بصورت نمایی است و با افزایش تعداد رئوس گراف زمان اجرا خیلی افزایش می یابد ولی اگر به نمودار الگوریتم پیشنهادی توجه کنیم در می یابیم که تغییرات زمان اجرای آن با افزایش تعداد گره های نمونه گرافها، تقریباً بصورت خطی و با یک ضریب ثابت افزایش پیدا می کند. در جدول (۴) و نمودار (۴) الگوریتم پیشنهادی را با الگوریتم ABAC مورد مقایسه قرار داده ایم همانطور که دیده می شود الگوریتم پیشنهادی سریعتر می باشد و همانند نمودار (۳-b) با افزایش تعداد گره های گراف زمان اجرا تقریباً بصورت خطی تغییر می کند و زمان اجرای الگوریتم ABCA بصورت تقریباً نمایی افزایش پیدا می کند (نمودار ۴-b). اگر دو الگوریتم را بر اساس تعداد رنگ های حاصل مورد مقایسه قرار دهیم می بینیم که الگوریتم پیشنهادی در بعضی نمونه ها بهتر از الگوریتم ABCA عمل کرده است و در بقیه، تعداد رنگ نزدیک به الگوریتم ABCA را بدست آورده است. و این خود نشاندهنده کارایی الگوریتم پیشنهادی می باشد.

۸. مراجع

- [۱] بزرگ زاده، محمد علی؛ عباد زاده، محمد مهدی؛ "استفاده از الگوریتم های تکاملی با ساختار کروموزم دو پارچه برای حل مساله رنگ آمیزی گراف"، هفتمین کنفرانس سیستمهای فازی، دانشگاه فردوسی مشهد، ۱۳۸۴.
- [۲] Blum, A., "Algorithms for Approximate Graph Coloring", PhD. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, ۱۹۹۱.
- [۳] D.S. Johnson, C.R. Aragon, L.A. McGoach, C. Schevon, "Optimization by simulated annealing: an experimental evaluation". Part ۲, graph coloring and number partitioning, Operations Research, vol. ۳۹(۳), pp. ۳۷۸-۴۰۶, ۱۹۹۱.
- [۴] Karger, D., Motwani, R., Sudan, M., "Approximate Graph Coloring by Semidefinite Programming", Journal of the Association for Computing Machinery, Vol. ۴۵, No. ۲, pp. ۲۴۶-۲۶۵, ۱۹۹۸.
- [۵] P. Briggs, K.D. Cooper, K. Kennedy, and L. Torczon. "Coloring heuristics for register allocation", In Proceedings of the SIGPLAN ۸۹ Conference on Programming Language Design and Implementation, pp. ۲۷۵-۲۷۴, ۱۹۸۹.
- [۶] G.J. Chaitin. "Register allocation and spilling via graph coloring", In Proceedings of the SIGPLAN ۸۹ Conference on Compiler Construction, pp. ۹۸-۱۰۱, ۱۹۸۲.
- [۷] G.J. Chaitin, M.A. Auslander, A.K. Chandra, J. Cocke, M.E. Hopkins, and P.W. Markstein. "Register allocation via coloring" Computer Languages, ۶: ۴۷-۵۷, ۱۹۸۱.
- [۸] D.C. Wood. "A Technique for Coloring a Graph Applicable to Large-Scale Optimization Problems" Computer Journal, ۱۲: ۳۱۷, ۱۹۶۹.
- [۹] D. de Werra, "An Introduction to Timetabling", European Journal of Operational Research, ۱۹(۱۹۸۵), ۱۵۱-۱۶۲.
- [۱۰] D. Brelaz, "New methods to color vertices of a graph", Communications of ACM, vol. ۲۲, pp. ۲۵۱-۲۵۶, ۱۹۷۹.
- [۱۱] F.T. Leighton, "A graph coloring algorithm for large scheduling problems", Journal of Research of the National Bureau Standard, vol. ۸۴, pp. ۴۸۹-۵۰۵, ۱۹۷۹.
- [۱۲] Halperin, E., Nathaniel, R., Zwick, U., "Coloring k-Colorable Graphs Using Smaller Palettes", In Proceedings of SODA, pp. ۳۱۹-۳۲۶, ۲۰۰۱.
- [۱۳] J. Han, and M. Kamber, "Data Mining: Concepts and Techniques", San Francisco: Morgan Kaufmann, ۲۰۰۱.
- [۱۴] F. Keller, "Clustering", Computer University Saarlandes, Tutorial Slides.
- [۱۵] G.L. Liu, "Introduction to Combinatorial Mathematics", McGraw-Hill, ۱۹۶۸.
- [۱۶] E.R. Hruschka, N.F.F. Ebecken, "A genetic algorithm for cluster analysis", Intelligent Data Analysis ۷(۱) ۱۵-۲۵, ۲۰۰۳.
- [۱۷] Maria Irene Miranda, "Clustering methods and algorithms"
<http://www.cse.iitb.ac.in/dbms/Data/Courses/CS۶۲۲/۱۹۹۹/clustering/dbms.html>
- [۱۸] Brian T. Luke: "K-Means Clustering", Tutorial Slides, <http://fconyx.ncifcrf.gov/~lukeb/kmeans.html>
- [۱۹] Andrew Moore: "K-means and Hierarchical Clustering", Tutorial Slides, <http://www-۲.cs.cmu.edu/~awm/tutorials/kmeans.html>
- [۲۰] Hee-su Kim, Sung-bae Cho, "An Efficient Genetic Algorithm With Less Fitness Evaluation By Clustering", IEEE, p. ۸۸۷-۸۸۹, ۲۰۰۱.
- [۲۱] A. R. Web, "Statistical Pattern Recognition", John Wiley & Sons, ۲۰۰۲.
- [۲۲] Q. He, "A Review of Clustering Algorithms as Applied in IR", Graduate School of Library and Information Science University of Illinois at Urbana-Champaign, ۱۹۹۹.
- [۲۳] E. salari, K. Eshghi, "An ACO Algorithm for the Graph Coloring Problem", Int.J.Contemp.Math.Sciences, Vol. ۳, ۲۰۰۸, no. ۶, ۲۹۳-۳۰۴.
- [۲۴] <http://cs.hbg.psu.edu/txn۱۳۱/INSTANCES/graphcoloring.html>.

- [25] C. Morgenstern, H. Shapiro, "**Coloration Neighborhood Structure for General Graph Coloring**", In Proceedings of the first annual ACM-SIAM Symposium on Discrete Algorithms, 226-235, 1990.
- [26] Thang N. Bui, ThanhVu H. Nguyen, Chirag M. Patel, Kim-Anh T. Phan," **An ant-based algorithm for coloring graphs "**, Discrete Applied Mathematics 156 (2008) 190 – 200.