

حل مسئله رنگ آمیزی گراف با استفاده از اتماماتای یادگیر سلوالی

محمد رضا میبدی	جود اکبری ترکستانی	علیرضا انعامی عراقی
دانشکده مهندسی کامپیوتر و فناوری اطلاعات	دانشکده مهندسی کامپیوتر	دانشکده مهندسی کامپیوتر
دانشگاه صنعتی امیر کبیر	دانشگاه آزاد اسلامی	دانشگاه آزاد اسلامی
تهران ایران	اراک ایران	فراهان ایران
mmeybodi@aut.ac.ir	j-akbari@iau-arak.ac.ir	alireza_enami@yahoo.com

خطی، استخراج قسمتهای موازی الگوریتم‌ها و انتساب فرکانس می‌باشد [2-5]. با توجه به اهمیت مسئله رنگ آمیزی گراف و کاربردهای فراوان آن، الگوریتم‌های فراوانی برای یافتن یک رنگ آمیزی مجاز در گراف، پیشنهاد شده است؛ از جمله، الگوریتم‌های دقيق [6,7]، الگوریتم‌های توییع شده [8,9]، الگوریتم‌های موازی [10,11]، الگوریتم‌های تقریبی [12-14] و الگوریتم‌های اکتشافی [15-17]. کارامیا و همکارش [15] الگوریتم HCD را پیشنهاد کردند که یک الگوریتم جستجوی محلی با قوانین اولویت می‌باشد که از تکنیک‌های جستجوی آگاهانه بهره می‌برد. این الگوریتم در نسخه بهینه سازی کار می‌کند و هنگامی که تعداد تکرارهای داده شده، بدست آمد؛ الگوریتم متوقف می‌شود. فانابیکی و همکارش [16] الگوریتم MIPS-CLR را ارائه کردند که در نسخه بهینه سازی کار می‌کند، اما نیازمند یک مقدار ورودی به عنوان هدف (k_{init}) برای رنگ آمیزی گراف می‌باشد. اگر الگوریتم قادر به حل مسئله با k_{init} رنگ نباشد، آنگاه این مقدار ورودی بصورت پویا تغییر می‌یابد. مالاگوتی و همکارانش [17] الگوریتم MMT را ارائه کردند که بر اساس یک راهکار فوق اکتشافی کار می‌کند که دارای دو فاز است: فاز اول مبتنی بر یک الگوریتم تکاملی است در حالی که فاز دوم یک فاز بعد از بهینه سازی است که مبتنی بر SCF می‌باشد. در این مقاله، الگوریتمی مبتنی بر اتماماتای یادگیر سلوالی نامنظم، برای حل مسئله رنگ آمیزی اصلی گراف پیشنهاد می‌گردد. در این الگوریتم از درجه یک سلول (رأس منتظر با آن سلول) برای دادن جریمه و پاداش استفاده می‌شود و درجه یک سلول بدین صورت تعریف می‌شود: تعداد سلولهای همسایه آن، که عمل (رنگ) متفاوتی انتخاب کرده اند. الگوریتم بدین صورت است که هر یک از اتماماتاهای یادگیر در اتماماتای یادگیر سلوالی، یک عمل از مجموعه اعمال خود را با توجه به بردار احتمال مربوطه انتخاب می‌کند. اگر عمل انتخابی یک سلول، متفاوت از اعمال انتخابی برای سلولهای همسایه باشد، در این صورت به عمل انتخابی این سلول پاداش داده می‌شود، ولی اگر عمل انتخاب شده توسط یک سلول، توسط بعضی از همسایگانش انتخاب شده باشد، در این صورت آن سلول فقط در صورتی پاداش می‌گیرد که درجه آن سلول از درجه تمام سلولهای همسایه همنگ با آن سلول، بیشتر باشد و در غیر این صورت سلول جریمه می‌شود. الگوریتم تا زمانی ادامه می‌یابد که تمام سلولها پاداش بگیرند.

چکیده: مسئله رنگ آمیزی رئوس گراف عبارت است از انتساب یک رنگ به هر یک از رئوس گراف، بگونه‌ای که به هر دو رأس مجاور دلخواه از گراف، رنگ‌های متفاوتی اختصاص داده شود. مسئله رنگ آمیزی گراف از مسائل بفرنج می‌باشد و به همین دلیل الگوریتم‌های متعددی که جوابهای قابل قبول در زمان کوتاه تولید می‌کنند، برای حل آن گزارش شده است. در این مقاله با استفاده از اتماماتای یادگیر سلوالی نامنظم، الگوریتمی برای حل این مسئله ارایه شده است. الگوریتم پیشنهادی، با الگوریتم‌های کارامیا، فانابیکی و مالاگوتی مقایسه شده است. طبق نتایج بدست آمده بر روی گرافهای نمونه، مشاهده می‌گردد که الگوریتم پیشنهادی نتایج به مرتبه بهتری در زمانی قبل قبول را تولید می‌کند.

كلمات کلیدی: مسئله رنگ آمیزی گراف، الگوریتم‌های اکتشافی، اتماماتای یادگیر سلوالی.

1- مقدمه

مسئله رنگ آمیزی اصلی رئوس¹ گراف($G=(V,E)$) که در آن V مجموعه رئوس و $E \subseteq V$ مجموعه یالهای گراف G می‌باشد؛ نکاشتی است بصورت $S: C:V \rightarrow W$ بطوری که برای دو رأس همسایه v و w داریم $C(v) \neq C(w)$. اعضای S را رنگها تشکیل می‌دهند. کوچکترین عدد صحیح k برای G در نگاشت $\{1,2,\dots,k\} \rightarrow C:V$ عدد رنگی G نامیده می‌شود و با (G) نشان داده می‌شود. یک گراف G ، k -رنگی نامیده می‌شود اگر $k(G) = k$ باشد، آنگاه G را k -رنگ پذیر می‌نامیم. یافتن عدد رنگی گراف برای گرافهای k -رنگ پذیر که در آن $k \geq 3$ یک مسئله بفرنج می‌باشد [1]. رنگ آمیزی مجاز عبارتست از انتساب یک رنگ به رئوس گراف؛ بطوری که رئوس همسایه دارای رنگ متفاوتی باشند و رنگ آمیزی بهینه گراف برابر با حداقل تعداد رنگ مورد نیاز برای یک رنگ آمیزی مجاز می‌باشد.

مسئله رنگ آمیزی گراف دارای کاربردهای فراوانی از جمله مسایل زمان‌بندی، مسئله تخصیص ثبات، حل معادلات دیفرانسیل با مشتقه جزیی، الگوریتم‌های موازی گاوس-سایدل برای حل معادلات جبری غیر

¹ Pure Vertex Coloring Problem(VCP)

² NP-Complete

عمل بهنگام‌سازی ساختار اتماتاهای موجود در اتماتاتی یادگیر سلولی توسط الگوریتم یادگیری انجام می‌شود. برای بعضی از کاربردهای اتماتاتی یادگیر سلولی می‌توان به [18-22] مراجعه نمود.

اتماتاتی یادگیر سلولی نامنظم: اتماتاتی یادگیر سلولی نامنظم [22]، یک اتماتاتی یادگیر سلولی است با حذف این محدودیت که اتماتاتی یادگیر سلولی در این نوع، دارای ساختار شبکه ای نمی‌باشد. این نوع اتماتاتی یادگیر سلولی، برای کاربردهایی مثل شبکه های حسگر بسیم، سیستمهای شبکه ای آزاد و کاربردهای مبتنی بر گراف که بر اساس یک شبکه مستطیلی نمی‌توانند مدل شوند، بسیار مؤثر است. یک اتماتاتی یادگیر سلولی نامنظم، بصورت یک گراف بدون جهت تعریف می‌شود؛ بطوری که هر رأس آن متناظر با یک سلول است که دارای یک اتماتاتی یادگیر می‌باشد. اتماتاتی یادگیر که در هر سلول وجود دارد، حالت(عمل) آن سلول را بر اساس بردار احتمال عمل مربوط به آن سلول، تعیین می‌کند. شبیه اتماتاتی یادگیر سلولی، قانونی وجود دارد که اتماتاتی یادگیر سلولی نامنظم، تحت آن عمل می‌کند. قانون اتماتاتی یادگیر سلولی و اعمال انتخاب شده بوسیله اتماتاتاهای یادگیر همسایه برای یک اتماتاتی یادگیر خاص، بردار احتمال اعمال مربوط به آن اتماتاتی یادگیر را بروز می‌کند.

3- الگوریتم پیشنهادی برای حل مساله رنگ‌آمیزی گراف

مبتنی بر اتماتاتی یادگیر سلولی نامنظم

به دلیل اینکه اغلب موقع اتماتاتی یادگیر سلولی به صورت منظم یعنی مانند آرایه‌های یک بعدی و دو بعدی دیده می‌شود، ولی اتماتاتی یادگیر سلولی می‌تواند نامنظم نیز باشد یعنی سلول‌های این اتماتاتی می‌توانند از نظر تعداد، نوع و شعاع همسایگی‌شان با هم تفاوت داشته باشند. به چنین اتماتاتی، اتماتاتی یادگیر سلولی نامنظم گفته می‌شود. جهت نگاشتن یک گراف به اتماتاتی یادگیر سلولی، ما از اتماتاتی یادگیر سلولی نامنظم استفاده می‌کنیم؛ برای این منظور، به ازای هر رأس از گراف، یک سلول از اتماتاتا را در نظر می‌گیریم و هر سلول در اتماتاتی یادگیر سلولی با سلول دیگر همسایه است اگر و تنها اگر رأس متناظر با آن سلول در گراف مسأله با رأس متناظر با سلول دیگر، یا داشته و یا به عبارت دیگر مجاور باشد. مجموعه رنگ‌ها، مجموعه اعمال هر یک از اتماتاتاهای یادگیر می‌باشد. اتماتاتی یادگیر در هر سلول از نوع L_{RP} با ضرایب پاداش و جریمه 0/01 می‌باشد. در ابتدا احتمال انتخاب اعمال هر یک از اتماتاتاهای یادگیر مساوی و برابر با یک تقسیم بر تعداد رنگ‌های مورد استفاده می‌باشد.

در [23] نشان داده شده است که اگر Δ بزرگترین درجه یک رأس در گراف باشد در این صورت Δ تعداد رنگ‌های مورد نیاز برای رنگ آمیزی مجاز گراف کوچکتر و یا مساوی $\Delta+1$ می‌باشد یعنی $c \leq \Delta+1$ ، بنابراین تعداد اعضای مجموعه عمل در این الگوریتم برابر $\Delta+1$ خواهد بود.

الگوریتم پیشنهادی با الگوریتمهای کارامیا [15]، فانابیکی [16] و ملاگوتی [17] مقایسه می‌گردد. نتایج آزمایشها نشان می‌دهد که از نظر تعداد رنگ الگوریتم پیشنهادی نتایج بهتری را تولید می‌کند، اما از نظر زمان اجرا، الگوریتمهای اکتشافی فوق عملکرد بهتری دارند. ادامه مقاله به این صورت سازماندهی شده است: در بخش 2 به معرفی ۴ اجمالی اتماتاتی یادگیر سلولی³ و اتماتاتی یادگیر سلولی نامنظم می‌پردازیم. در بخش 3 الگوریتم پیشنهادی مبتنی بر اتماتاتی یادگیر سلولی نامنظم شرح داده می‌شود و در بخش 4 الگوریتم ارایه شده با الگوریتمهای کارامیا، فانابیکی و ملاگوتی مقایسه می‌گردد. بخش نهایی مقاله نتیجه‌گیری می‌باشد.

2- اتماتاتی یادگیر سلولی

اتماتاتی یادگیر سلولی: اتماتاتی یادگیر سلولی d بعدی [18-19] یک چندتایی ($CLA = (Z^d, f, A, N, F)$ است به طوریکه:

- Z^d یک شبکه از d تایی‌های مرتب از اعداد صحیح می‌باشد. این شبکه می‌تواند یک شبکه متناهی، نیمه متناهی یا متناهی باشد.

• f یک مجموعه متناهی از حالتها می‌باشد.

• A ، یک مجموعه از اتماتاتاهای یادگیر است که هر یک از آنها به یک سلول از اتماتاتی سلولی نسبت داده می‌شود.

• $N = \{\bar{x}_1, \dots, \bar{x}_m\}$ یک زیر مجموعه متناهی از Z^d می‌باشد که بردار همسایگی نامیده می‌شود.

• $F: \underline{f}^{\bar{m}} \rightarrow \underline{b}$ قانون محلی اتماتاتی یادگیر ساولی می‌باشد به طوریکه \underline{b} مجموعه مقادیری است که می‌تواند به عنوان سیگنال تقویتی پذیرفته شود.

عملکرد اتماتاتی یادگیر سلولی را می‌توان به شرح زیر بیان کرد. در هر لحظه هر اتماتاتی یادگیر در اتماتاتی یادگیر سلولی یک عمل از مجموعه اعمال خود را انتخاب می‌کند. این عمل می‌تواند بر اساس مشاهدات قبلی و یا به صورت تصادفی انتخاب شود. عمل انتخاب شده با توجه به اعمال انتخاب شده توسط سلولهای همسایه و قانون حاکم بر اتماتاتی یادگیر سلولی پاداش داده و یا جریمه می‌شود. با توجه به اینکه عمل انتخاب شده پاداش گرفته و یا جریمه شده است، اتماتاتا رفتار خود را تصحیح کرده و ساختار داخلی اتماتاتا بهنگام انجام می‌گردد. عموماً عمل بروزرسانی، هر اتماتاتا در اتماتاتی یادگیر سلولی دوباره یک عمل از مجموعه اعمال خود را انتخاب کرده و انجام می‌دهد. فرآیند انتخاب عمل و دادن پاداش و یا جریمه تا زمانی که سیستم به حالت پایدار برسد و یا یک معیار از قبل تعریف شدهای برقرار شود، ادامه می‌باید.

³ Cellular Learning Automata(CLA)

⁴ Irregular Cellular Learning Automata(ICLA)

گام پنجم: بررسی خاتمه الگوریتم.

در صورتی که تمام سلولها پاداش بگیرند، الگوریتم خاتمه می یابد و در غیر این صورت اجرای الگوریتم از گام دوم ادامه می یابد.

شبه کد الگوریتم پیشنهادی در شکل ۱ آورده شده است.

Algorithm CLA

Input : Graph G(V,E)

Output : The number of colors needs to be chosen for coloring the graph

Procedure Saturation degree(cell):

Determines the number of neighbors for a given cell, which they choose different action.

Begin

Construct an irregular CLA isomorphic to the input graph

Repeat

For all cells do in parallel

Each cell chooses one of its actions according to its action probability vector

If (The chosen action is different from those chosen by all its neighbors)

Reward the chosen action

Else

If (The saturation degree(cell) is greater than those of all its neighbors)

Reward the chosen action

Else

Penalize the chosen action

End if

End if

Until all cells are rewarded by the random Environment

Return the number of used colors

End.

شکل ۱- شبه کد الگوریتم پیشنهادی

4- نتایج آزمایشات

برای اجرای الگوریتمها، یک کامپیوتر شخصی با پردازنده اینتل P4 2.4 GHz و با 512 مگابایت حافظه اصلی و ویندوز XP مورد استفاده قرار گرفته است و روی نمونه های پیشنهاد شده در استاندارد DIMACS تست شده است [24,25]. نتایج حاصل از الگوریتم پیشنهادی با نتایج الگوریتمهای کارامیا [15]، فانابیکی [16] و مالاگوتی [17] مقایسه شده است. معیارهای ارزیابی الگوریتمها، تعداد رنگهای استفاده شده برای رنگ آمیزی (C) و زمان اجرا بر حسب ثانیه (T) می باشد. در رنگ آمیزی اصلی گراف، همانگونه که از نتایج آزمایشات مشهود است، اجرای الگوریتمها بر روی 42 نمونه پیشنهاد شده در استاندارد DIMACS نشان می دهد که الگوریتم فانابیکی دارای نتایج بهتری نسبت به الگوریتم کارامیا می باشد و همچنین الگوریتم مالاگوتی همواره نتایج بهتر یا برابر الگوریتم کارامیا و فانابیکی تولید می کند، بجز در مورد latin-square-10 که الگوریتم فانابیکی

در این الگوریتم از درجه یک سلول (رأس متناظر با آن سلول) برای دادن جریمه و پاداش استفاده می شود و درجه یک سلول بدین صورت تعریف می شود: تعداد سلولهای همسایه آن، که عمل (رنگ) متفاوتی انتخاب کرده اند. الگوریتم بدین صورت است که هر یک از اتماتاهای یادگیر در اتماتای یادگیر سلولی، یک عمل از مجموعه اعمال خود را که تعداد آنها برابر است با $\Delta + 1$ با توجه به بردار احتمال مربوطه انتخاب می کند. اگر عمل انتخابی یک سلول، متفاوت از اعمال انتخابی برای سلولهای همسایه باشد؛ در این صورت به عمل انتخابی این سلول، توسط بعضی از همسایگانش انتخاب شده باشد؛ در این صورت آن سلول فقط در صورتی پاداش می گیرد که درجه آن سلول از درجه تمام سلولهای همسایه همنگ با آن سلول، بیشتر باشد و در غیر این صورت سلول جریمه می شود. الگوریتم تا زمانی ادامه می یابد که تمام سلولها پاداش بگیرند. البته در جهت بهبود الگوریتم در بخش 4 شرط خاتمه را تغییر داده ایم. مراحل الگوریتم پیشنهادی را می توان در گامهای زیر تشریح کرد:

گام اول: ساخت یک اتماتای یادگیر سلولی نامنظم همراه با گراف ورودی.

در این گام به هر یک از رئوس گراف یک سلول متناظر می شود و دو سلول در صورتی همسایه هستند که رئوس متناظر آنها در گراف ورودی نیز همسایه باشند، همچنین به هر سلول یک اتماتای یادگیر اختصاص می یابد.

گام دوم: انتخاب تصادفی عملها توسط اتماتای یادگیر هر سلول.

در این گام هر یک از اتماتاهای یادگیر یک عمل از مجموعه اعمال خود را با توجه به بردار احتمال مربوطه انتخاب می کند.

گام سوم: ارزیابی رنگ آمیزی مجاز.

رنگ آمیزی مجاز عبارتست از انتساب یک رنگ به هر یک از رئوس گراف بگونه ای که به هر دو رأس مجاور دلخواه از گراف رنگ های متفاوتی اختصاص داده شود.

گام چهارم: پاداش دادن و جریمه کردن عمل انتخابی اتماتاهای و بروز رسانی بردار احتمالات.

اگر عمل انتخابی یک سلول، متفاوت از اعمال انتخابی برای سلولهای همسایه باشد؛ در این صورت، به عمل انتخابی این سلول پاداش داده می شود، ولی اگر عمل انتخاب شده توسط یک سلول، توسط بعضی از همسایگانش انتخاب شده باشد؛ در این صورت آن سلول، فقط در صورتی پاداش می گیرد که درجه (تعداد سلولهای همسایه آن که عمل متفاوتی انتخاب کرده اند) آن سلول از درجه تمام سلولهای همسایه همنگ با آن سلول، بیشتر باشد و در غیر این صورت سلول جریمه می شود.

الگوریتمهای مورد مقایسه می باشد. همانطور که ملاحظه می شود الگوریتم پیشنهادی در 8 مورد باعث کاهش تعداد رنگ می شود و در 6 مورد نیز تعداد رنگ افزایش می یابد و در 28 مورد نیز تعداد رنگ استفاده شده تفاوتی نمی کند.

نتیجه بهتری را تولید می کند. هر یک از نتایج که در جدول 1 گزارش شده است، متوسط 50 بار اجرا می باشد. ستونهای اول (|V|) و دوم (|E|) در جدول 1 به ترتیب برابر تعداد رئوس و تعداد یالهای گرافهای نمونه می باشد. ستون آخر (diff.) نیز تفاضل بین تعداد رنگ بدست آمده توسط الگوریتم پیشنهادی و بهترین تعداد رنگ بدست آمده توسط

جدول 1- تعداد رنگ و مدت زمان لازم برای رنگ آمیزی گراف به کمک الگوریتمهای مختلف

graph	V	E	best	proposed algorithm		Caramia[15]		Fanabiki[16]		Malaguti[17]		diff.
				T	C	T	C	T	C	T	C	
DSJC125.1	125	736	5	54.15	5	0	6	0	5	21	5	0
DSJC125.5	125	3891	17	119.71	17	0	19	1	17	122	17	0
DSJC125.9	125	6961	44	149.5	42	0	44	0	44	121	44	-2
DSJC250.1	250	3218	8	89.19	9	2	9	5	8	21	8	1
DSJC250.5	250	15668	28	183.63	28	1	31	14	28	117	28	0
DSJC250.9	250	27897	72	215.02	74	3	75	31	72	89	72	2
DSJC500.1	500	12458	12	163.14	12	3	14	84	12	210	12	0
DSJC500.5	500	62624	48	427.33	48	141	53	349	49	388	48	0
DSJC500.9	500	112437	127	593.15	126	73	132	480	127	433	127	-1
DSJC1000.1	1000	49629	20	312.01	20	140	22	90	21	260	20	0
DSJC1000.5	1000	249826	84	931.81	85	531	95	4658	88	8407	84	1
DSJC1000.9	1000	440449	225	1035.27	225	857	241	1565	228	3234	225	0
DSJR500.1	500	3555	12	81.53	12	2	12	0	12	25	12	0
DSJR500.1C	500	121275	85	211.61	85	7	85	6	85	88	85	0
DSJR500.5	500	58862	122	137.92	121	269	129	276	122	163	122	-1
le450-15a	450	8168	15	169.53	15	42	16	1	15	0	15	0
le450-15b	450	8169	15	173.21	14	19	16	1	15	0	15	-1
le450-15c	450	16680	15	191.19	15	213	17	11	15	3	15	0
le450-15d	450	16750	15	243.59	16	68	18	5	15	4	15	1
le450-25c	450	17343	25	268.02	25	59	27	7	26	1321	25	0
le450-25d	450	17425	25	294.53	25	74	27	1	26	436	25	0
le450-5a	450	5714	5	129.73	5	6	8	1	5	0	5	0
le450-5b	450	5734	5	142.19	5	0	8	2	5	0	5	0
le450-5d	450	9757	5	192.5	5	0	7	3	5	0	5	0
r125.1	125	209	5	23.15	5	0	5	0	5	0	5	0
r125.1c	125	7501	46	112.62	45	0	46	0	46	20	46	-1
r125.5	125	3838	36	87.19	36	17	36	0	36	21	36	0
r250.1	250	867	8	62.83	8	0	8	0	8	26	8	0
r250.1c	250	30227	64	215.02	65	0	64	2	64	21	64	1
r250.5	250	14849	65	163.17	65	5	68	16	65	64	65	0
r1000.1	1000	14378	20	121.4	20	13	20	0	20	37	20	0
r1000.1c	1000	485090	98	859.31	98	173	99	557	98	518	98	0
r1000.5	1000	238267	234	615.27	233	727	256	1345	237	753	234	-1
school1	385	19095	14	113.5	14	0	14	0	14	0	14	0
school1-nsh	352	14612	14	94.25	14	1	14	1	14	21	14	0
latin-square-10	900	307350	99	726.43	98	687	107	938	99	5156	101	-1

flat300-20-0	300	21375	20	622.06	20	2	20	2	20	21	20	0
flat300-26-0	300	21633	26	741.5	26	1	35	1	26	36	26	0
flat300-28-0	300	21695	31	714.18	32	38	33	133	31	212	31	1
flat1000-50-0	1000	245000	50	863.14	50	551	92	14	50	1417	50	0
flat1000-60-0	1000	245830	60	927.01	58	562	93	59	60	3645	60	-2
flat1000-76-0	1000	246708	83	1042.19	83	293	94	2499	87	7325	83	0

ضرایب پاداش و جریمه متفاوت انجام داده و نتایج حاصل از آنها را با هم مقایسه کردایم. گراف‌های استفاده شده در این آزمایشات، همان گراف‌های مجموعه DIMACS [24,25] می‌باشد. معیار ارزیابی تعداد رنگ‌های استفاده شده برای رنگ‌آمیزی گراف(C) می‌باشد. آزمایشات ما روی گراف‌های ذکر شده و با ضرایب پاداش و جریمه ۰/۰, ۰/۰/۱, ۰/۱ و ۳/۰ انجام شده و نتایج حاصل از ۵۰ بار اجرا در جدول ۲ گزارش شده است. همانگونه که از نتایج آزمایشات مشهود است هر چه ضرایب پاداش و جریمه بزرگتر می‌شود تعداد رنگ‌ها افزایش می‌یابد و بالعکس زمان اجرا کاهش می‌یابد.

1-4- نتایج آزمایشات انجام شده جهت بهبود الگوریتم

ارایه شده

در جهت بهبود الگوریتم ارایه شده، تغییراتی را در چند زمینه مختلف در الگوریتم ایجاد کردیم و حالت‌هایی را که الگوریتم نتایج بهتری تولید می‌کند، را مشخص نمودیم. این آزمایشات در دو جنبه مختلف صورت گرفته است که در ادامه به آنها خواهیم پرداخت.

انتخاب ضرایب پاداش و جریمه بهینه: در الگوریتم ارایه شده، اتماتای یادگیر از نوع L_{RP} با ضرایب پاداش و جریمه ۰/۰/۱ می‌باشد. ما در جهت یافتن ضرایب پاداش و جریمه بهینه، آزمایشات مختلفی را با

جدول 2- تعداد رنگ و مدت زمان لازم برای رنگ آمیزی گراف به کمک الگوریتم پیشنهادی و با ضرایب پاداش و جریمه متفاوت

graph	0.01			0.1			0.2			0.3			graph	0.01			0.1			0.2					
	T	C	C	C	C	C	T	C	C	C	C	C		T	C	C	C	C	C	C	C	C			
DSJC125.1	54.15	5	5	5	5	5	le450-5a	129.73	5	5	5	5	7	DSJC125.5	119.71	17	17	17	18	le450-5b	142.19	5	6	7	9
DSJC125.5	149.5	42	43	47	47	47	le450-5d	192.5	5	5	5	5	5	DSJC125.9	89.19	9	9	11	12	r125.1	23.15	5	7	8	11
DSJC125.9	183.63	28	29	32	32	35	r125.1c	112.62	45	47	51	51	55	DSJC250.1	215.02	74	78	79	79	r125.5	87.19	36	39	40	47
DSJC250.1	427.33	48	51	51	52	52	r250.1	62.83	8	10	13	13	17	DSJC250.5	593.15	126	131	134	134	r250.1c	215.02	65	66	69	75
DSJC250.5	931.81	85	89	91	97	97	r250.5	163.17	65	65	70	70	73	DSJC500.1	312.01	20	20	21	24	r1000.1	121.4	20	21	29	38
DSJC500.1	1035.27	225	225	229	235	235	r1000.1c	859.31	98	110	105	105	113	DSJC500.5	81.53	12	12	13	15	school1	113.5	14	14	16	20
DSJC500.5	211.61	85	86	86	87	87	school1-nsh	94.25	14	15	19	19	20	DSJR500.1	137.92	121	123	124	127	latin-square-10	726.43	98	99	111	118
DSJR500.1	169.53	15	15	15	15	15	flat300-20-0	622.06	20	20	22	22	27	le450-15a	173.21	14	14	16	19	flat300-26-0	741.5	26	28	29	29
DSJR500.5	191.19	15	17	18	22	22	flat300-28-0	714.18	32	39	35	35	45	le450-15c	243.59	16	19	20	23	flat1000-50-0	863.14	50	62	53	59
le450-15d	268.02	25	26	27	29	29	flat1000-60-0	927.01	58	58	62	62	67	le450-25c	294.53	25	26	26	30	flat1000-76-0	1042.19	83	85	89	92
le450-25d																									

سلولها پاداش بگیرند، همچنین باید حاصلضرب احتمال انتخاب هر عمل توسط اتماتای مربوط به آن سلول، برای یک رنگ آمیزی مجاز در گراف از یک مقدار آستانه بیشتر شود. ما در جهت تعیین حد آستانه

انتخاب شرط خاتمه بهینه: در الگوریتم ارایه شده شرط خاتمه اینست که تمام سلولها پاداش بگیرند، در جهت یک رنگ آمیزی بهینه؛ شرط خاتمه بدین شکل اصلاح می‌شود که علاوه بر اینکه بایستی تمام

روی گراف‌های ذکر شده و با حددهای آستانه ۰/۶, ۰/۴ و ۰/۸ انجام شده و نتایج حاصل از ۵۰ بار اجرا در جدول ۳ گزارش شده است. همانگونه که از نتایج آزمایشات مشهود است هر چه حددهای آستانه بزرگتر می‌شود، تعداد رنگها به تعداد رنگ بهینه نزدیکتر می‌شود اما بر عکس زمان اجرا افزایش می‌یابد.

بهینه، آزمایشات مختلفی را با حددهای آستانه متفاوت انجام داده و نتایج حاصل از آنها را با هم مقایسه کردیم. گراف‌های استفاده شده در این آزمایشات همان گراف‌های مجموعه DIMACS [24,25] می‌باشد. معیار ارزیابی، تعداد رنگ‌های استفاده شده (C) و همچنین زمان اجرا بر حسب ثانیه (T) برای رنگ‌آمیزی گراف می‌باشد. آزمایشات ما

جدول ۳- تعداد رنگ و مدت زمان لازم برای رنگ آمیزی گراف به کمک الگوریتم پیشنهادی و با حددهای آستانه متفاوت

graph	pure		0.2		0.4		0.6		0.8	
	T	C	T	C	T	C	T	C	T	C
<i>DSJC125.1</i>	54.15	5	59.5	5	68.31	5	77.45	5	86.21	5
<i>DSJC125.5</i>	119.71	17	123.42	17	131.02	17	137.2	17	149.53	16
<i>DSJC125.9</i>	149.5	42	151.7	42	162	42	178.3	42	192.7	42
<i>DSJC250.1</i>	89.19	9	95.41	9	113.7	8	139.11	8	154.61	8
<i>DSJC250.5</i>	183.63	28	192.62	28	213.55	28	229.73	28	261.15	26
<i>DSJC250.9</i>	215.02	74	223.19	74	261.9	74	283.07	72	291.84	72
<i>DSJC500.1</i>	163.14	12	169.43	12	172.58	12	181.89	12	201.02	12
<i>DSJC500.5</i>	427.33	48	431.55	48	445.62	48	463.31	48	491.73	48
<i>DSJC500.9</i>	593.15	126	602.14	126	631.51	126	659.48	126	693.19	126
<i>DSJC1000.1</i>	312.01	20	321.16	20	361.94	20	391.5	20	415.62	20
<i>DSJC1000.5</i>	931.81	85	942.7	84	949.32	84	976.01	84	1011.19	83
<i>DSJC1000.9</i>	1035.27	225	1051.93	225	1082.09	225	1131.59	225	1192.43	225
<i>DSJR500.1</i>	81.53	12	95.27	12	123.25	12	159.03	12	185.41	12
<i>DSJR500.1C</i>	211.61	85	219.5	85	245.43	85	281.71	84	305.16	84
<i>DSJR500.5</i>	137.92	121	152.18	121	178.5	121	192.94	121	237.81	121
<i>le450-15a</i>	169.53	15	175.13	15	192.05	15	237.13	15	281.95	15
<i>le450-15b</i>	173.21	14	181.12	14	205.24	14	214.2	14	263.2	14
<i>le450-15c</i>	191.19	15	201.35	15	231.07	15	289.11	15	304.35	15
<i>le450-15d</i>	243.59	16	253.07	16	265.14	16	281.07	16	297.37	14
<i>le450-25c</i>	268.02	25	274.16	25	293.26	25	314.55	25	371.19	25
<i>le450-25d</i>	294.53	25	299.17	24	329.91	24	355.09	24	392.5	24
<i>le450-5a</i>	129.73	5	135.48	5	158.34	5	182.85	5	185.61	5
<i>le450-5b</i>	142.19	5	159.2	5	181.55	5	197.26	4	217.27	4
<i>le450-5d</i>	192.5	5	203.37	5	256.41	5	264.15	5	289.91	5
<i>r125.1</i>	23.15	5	35.5	5	52.43	5	89.61	5	121.13	5
<i>r125.1c</i>	112.62	45	129.24	45	155.55	45	191.92	45	209.12	45
<i>r125.5</i>	87.19	36	95.71	36	115.14	36	127.53	36	161.53	36
<i>r250.1</i>	62.83	8	71.55	8	99.16	8	110.21	8	170.07	8
<i>r250.1c</i>	215.02	65	241.9	65	281.7	65	301.19	64	314.61	64
<i>r250.5</i>	163.17	65	165.85	65	192.39	65	237.5	65	243.71	65
<i>r1000.1</i>	121.4	20	137.26	20	151.72	20	162.18	20	192.51	20
<i>r1000.1c</i>	859.31	98	861.17	98	891.5	98	701.13	98	734.49	98
<i>r1000.5</i>	615.27	233	618.49	233	652.18	233	679.12	233	693.23	233
<i>school1</i>	113.5	14	125.31	14	167.31	14	185.53	14	207.9	14
<i>school1-nsh</i>	94.25	14	98.61	14	129.12	14	142.7	14	151.52	14
<i>latin-square-10</i>	726.43	98	731.15	98	782.53	98	809.61	98	813.34	98

flat300-20-0	622.06	20	625.48	20	631.24	20	672.19	20	712.2	20
flat300-26-0	741.5	26	749.02	26	782.07	26	801.59	26	831.73	26
flat300-28-0	714.18	32	721.91	32	753.41	32	793.02	31	814.5	29
flat1000-50-0	863.14	50	873.26	50	891.62	49	922.35	49	973.42	49
flat1000-60-0	927.01	58	932.91	58	982.19	58	1012.73	58	1054.17	58
flat1000-76-0	1042.19	83	1059.33	83	1081.34	83	1127.19	83	1186.09	83

[11] M. Luby, "A Simple Parallel Algorithm for the Maximal Independent Set Problem", SIAM Journal on Computing, Vol. 4, p. 1036, 1989.

[12] A. Blum, "Algorithms for Approximate Graph Coloring", PhD. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1991.

[13] D. Karger, R. Motwani, M. Sudan, "Approximate Graph Coloring by Semidefinite Programming", Journal of the Association for Computing Machinery, Vol. 45, No. 2, pp. 246-265, 1998.

[14] E. Halperin, R. Nathaniel, U. Zwick, "Coloring k-Colorable Graphs using Smaller Palettes", In Proceedings of SODA, pp. 319-326, 2001.

[15] M. Caramia and P. Dell'Olmo,"A Fast and Simple Local Search for Graph Coloring", Proc. of the 3d Workshop on Algorithm Engineering WAE'99, Lecture Notes in Computer Science, pp. 319-313, 1999.

[16] N. Funabiki and T. Higashino, "A Minimal-State Processing Search Algorithm for Graph Coloring Problems", IEICE Trans, Fundamentals, pp. 1420-1430, 1990.

[17] E. Malaguti, M. Monaci, and P. Toth, "A Metaheuristic Approach for the Vertex Coloring Problem", Technical Report OR/05/3, DEIS University of Bologna, 2005.

[18] M.R. Meybodi, H. Beigy, M. Taherkhani, "Cellular Learning Automata and Its Applications", Journal of Science and Technology, University of Sharif, No. 25, pp. 54-77, Autumn/Winter 2003-2004.

[19] H. Beigy and M.R. Meybodi, "A Mathematical Framework for Cellular Learning Automata", Advanced in Complex Systems, Vol. 7, No. 3 and 4, pp. 294-319, 2004.

[20] M.R. Meybodi and M.R. Kharazmi, "Cellular Learning Automata and Its Application to Image Processing", Journal of Amirkabir, Vol. 14, No. 56A, pp. 1101-1126, 2004.

[21] M.R. Meybodi and F. Mehdipour, "Application of Cellular Learning Automata with Input to VLSI Placement", Journal of Modarres, University of Tarbeit Modarres, Vol. 16, pp. 81-95, summer 2004.

[22] M. Asnaashari and M.R. Meybodi, "Irregular Cellular Learning Automata and Its Application to Clustering in Sensor Networks", Proceedings of 15th Conference on Electrical Engineering (15th ICEE), Volume on Communication, Telecommunication Research Center, Tehran, Iran, May 15-17, 2007.

[23] R. Diestel, "Graph Theory", 3rd Edition, Springer-Verlag, New York, 2005.

[24] <ftp://dimacs.rutgers.edu/pub/challenge/graph/>.

[25] D.S. Johnson and M.A. Trick (eds.), "Cliques, Coloring, and Satisfiability: 2nd DI-MACS Implementation Challange, 1993", DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1996.

5- نتیجه گیری

در این مقاله الگوریتمی برای حل مساله رنگ آمیزی گراف با استفاده از اثباتاتی یادگیر سلولی نامنظم پیشنهاد گردید. الگوریتم پیشنهادی با الگوریتم های کارامیا، فانابیکی و ملاگوتی مقایسه شد که بر طبق نتایج به دست آمده الگوریتم پیشنهادی نتایج بهتری را نسبت به الگوریتم های فوق الذکر تولید می کند. همچنین تغییراتی در الگوریتم پیشنهادی صورت گرفت تا نتایج حاصل از الگوریتم در تعداد رنگ و همچنین در زمان اجرا بهبود یابد؛ این تغییرات عبارت بودند از انتخاب ضرایب پاداش و جریمه متفاوت و تغییر شرط خاتمه.

مراجع

- [1] R. Karp, "Reducibility among Combinatorial Problems", Complexity of computer computations, pp. 85-104, 1972.
- [2] T.A. Redl, "A Study of University Timetabling that Blends Graph Coloring with the Satisfaction of Various Essential and Preferential Conditions", PhD. Thesis, Rice University, 2004.
- [3] W.K. Hale, "Frequency Assignment: Theory and Applications", Proceeding of IEEE, Vol. 68, No. 12, pp. 1497-1514, 1980.
- [4] T. Park and C.Y. Lee, "Application of the Graph Coloring Algorithm to the Frequency Assignment Problem", Journal of the Operations Research Society of Japan, Vol. 39, No. 2, pp. 258-265, 1996.
- [5] V. Maniezzo and R. Montemanni, "An Exact Algorithm for the Min-Interference Frequency Assignment Problem", Tech. Report WP-CO0003, Scienze dell'Informazione, University of Bologna, Cesena, Italy, 2000.
- [6] K.H. Rozen and et al, "Handbook of Discrete and Combinatorial Mathematics", United States of America, CRC Press, 2000.
- [7] D.J.A. Welsh and M.B. Powell, "An Upper Bound for the Chromatic Number of a Graph and its Application to Timetabling Problems", Computer Journal, Vol. 10, pp. 85-86, 1967.
- [8] B. Awerbuch and et al., "Network Decomposition and Locality in Distributed Computation", in Proceedings of the 30th Symposium on Foundations of Computer Science(FOCS), pp. 364-369, 1989.
- [9] N. Linial, "Locality in Distributed Graph Algorithms", SIAM Journal on Computing, Vol. 21, No. 1, pp. 193-201, 1992.
- [10] M.T. Jones and P.E. Plassmann, "A Parallel Graph Coloring Heuristic", SIAM Journal of Scientific Computing, Vol. 4, p. 654, 1993.