یک الگوریتم موازی برای رنگ آمیزی گراف
برای اتوماتای سلولی دو بعدی

محمدرضا میبدی   شهرام گلزاری

دانشکده مهندسی کامپیوتر   آزمایشگاه سیستم ها نرم افزاری

دانشگاه صنعتی امیرکبیر

تهران–ایران

چکیده : در این مقاله یک الگوریتم موازی برای رنگ‌آمیزی گراف توری بر مبنای CA دو بعدی ارائه می‌شود. در این الگوریتم در ابتدا گراف توری در CA نگاشت می شود.و سپس بوسیله تراکنشهای ساده و محلی ومنطبق با خواص حاکم بر اتوماتای سلولی، گراف با استفاده از حداقل تعداد رنگ، رنگ آمیزی می شود.

کلمات کلیدی: اتوماتای سلولی، گراف توری، الگوریتمهای رنگ آمیزی گراف، الگوریتمهای موازی

۱- مقدمه

رنگ‌آمیزی گراف یکی از مسائل معروف و کاربردی نظریه گرافها می‌باشد. در این مسأله هدف رنگ نمودن تمام گره‌های گراف به صورتی می‌باشد که هیچ دو گره همسایه دارای رنگ یکسان نباشند. رنگ‌آمیزی گراف در زمینه‌های فراوانی نظیر حل معادلات دیفرانسیل با مشتقات جزئی، الگوریتم‌های موازی گوس–سایدل برای حل معادلات جبری غیرخطی، زمانبندی فرآیندها و استخراج قسمت‌های موازی الگوریتم‌ها کاربرد دارد[12][10][4][2]. الگوریتمهای ترتیبی برای رنگ‌آمیزی گراف بطور گسترده مورد مطالعه قرار گرفته است[12][11][2]. در این مطالعات اثبات شده است که هر گراف مسطح را می‌توان با استفاده از چهار رنگ، رنگ‌آمیزی نمود در صورتی که برای رنگ‌آمیزی گرافهای غیر مسطح نیاز به تعداد بیشتری رنگ می‌باشد[1]. مسأله رنگ‌آمیزی گراف با حداقل مقدارK (کمترین تعداد رنگهای ممکن) یک مسأله NP-Complete است. یک مسأله ساده تر رنگ‌آمیزی گراف با استفاده از تعداد کم رنگها می‌باشد. برای مسأله رنگ‌آمیزی با تعداد کم رنگ گرچه الگوریتمهای ترتیبی متعددی با زمان چند جمله ای گزارش شده است ولی الگوریتمهای موازی زیادی گزارش نشده است. یک الگوریتم موازی برای این مساله توسط Luby ارائه شده است که یک گراف دارای n گره را در متوسط زمانی O(log n) در مدل PRAM رنگ‌آمیزی می‌نماید[12]. در[10] پیچیدگی زمانی الگوریتم گزارش شده در [12] بهبود داده شده است.

در این مقاله یک الگوریتم موازی برای رنگ‌آمیزی گراف توری برای اتوماتای سلولی[1] دو بعدی ارائه می‌شود. در این الگوریتم در ابتدا گراف توری در اتوماتای سلولی(CA) نگاشت می شود.و سپس بوسیله تراکنشهای ساده و محلی ومنطبق با خواص حاکم بر اتوماتای سلولی، گراف با استفاده از حداقل تعداد رنگ، رنگ آمیزی می شود. ادامه مقاله بصورت زیر سازماندهی شده است مفاهیم اصلی اتوماتای سلولی در بخش ۲ تشریح می شود و سپس در بخش ۳ ، الگوریتمهای رنگ آمیزی گراف را مرور خواهیم نمود. در بخش ٤ به شرح مفصل الگوریتم پیشنهادی می پردازیم. در این قسمت ابتدا ایده الگوریتم را شرح داده و سپس قوانین CA را طراحی می نماییم. در بخش ٥ بحثی در باره نگاشت کردن گرافهای غیر توری در CA را خواهیم داشت. آخرین قسمت مقاله نتیجه گیری میباشد.

۲- اتوماتای سلولی

اتوماتای سلولی یک شبکهٔ منظم از ماشینهای با حالت محدود که کدام از انها یک سلول نامیده میشود میباشد. هر کدام از این سلولها طبق الگوی ثابت و یکسانی با بعضی از سلولهای مجاور خویش در ارتباط میباشند. این ارتباط محلی بوده و برای تمام سلولها یکسان است. تمام سلولهایی که با یک سلول در ارتباط میباشند همسایگان ان سلول را تشکیل می دهند. هر سلول در هر لحظه از زمان می تواند یکی از حالتهای موجود در مجموعه حالت خود را اختیار نماید. مجموعه حالت برای تمامی سلولها یکسان می باشد. در هر لحظه از زمان حالت تمامی سلولها به صورت همزمان و بر اساس یک قانون یکسان تغییر می کند. این قانون خود تابعی از حالتهای همسایگان سلول بوده و بنابراین در هر لحظه از زمان، حالت بعدی هر سلول به حالت فعلی تمامی همسایگانش بستگی دارد. این شبکه از یک پیکره بندی اولیه شروع به کار کرده، در هر مرحله زمانی با اعمال قانون به تمامی سلولها پیکره بندی بهنگام شده. از این طریق با گذشت زمان شبکه رفتار های خاص و پیچیده از خود نشان میدهد.

با توجه به توضیحات فوق، وجه تمایز CA نسبت به دیگر شبکه های اتوماتایی عبارت است از : ساده بودن ساختار، محلی بودن ارتباطات بین سلولها، برقراری الگوی ارتباطی یکسان برای تمام سلولها، بهنگام سازی همزمان سلولها، بهنگام سازی سلولها توسط یک قانون یکسان و تولید رفتارهای خاص و پیچیده از مجموعه سلولها. با توجه به این خصوصیات در اکثر مستندات از CA به عنوان ساختاری موازی، محلی و یکسان نام برده شده است که براحتی می تواند رفتارهایی را که دارای این سه ویژگی هستند، شبیه سازی نماید[22][18][9][7]. امروزه CA در زمینه های متعددی از جمله تولید الگوهای تصادفی، نظریه محاسبات، مدلسازی سیستم های فیزیکی و بیولوژیکی و محاسبات کاربردی مورد استفاده قرار گرفته است[22][20][18][16][13][12][9][8][6].

یک نمونه ازCA هایی که امروزه متداول می باشد، CA دو بعدی است[15]. در این ساختار سلولها در یک شبکه دو بعدی قرار دارند. یکی از نکات مهم در این CA نوع همسایگی می باشد. دو نوع همسایگی مهم در این ساختار عبارتند از همسایگی von Neumann و همسایگی Moore . در همسایگی Moore برای هر سلول هشت سلول همسایه و در همسایگی von Neumann چهار سلول همسایه در نظر گرفته می شود(شکل ۱).

اگرچه اثبات شده است که CA یک مدل محاسباتی عمومی می باشد[5][3] ولی اغلب در عمل از این مدل برای شبیه سازی پدیده های فیزیکی پویا استفاده شده و این مدل بیش از آنکه مورد توجه دانشمندان علم کامپیوتر قرار گیرد، توجه پژوهشگران سایر علوم مانند فیزیک و بیولوژی را به خود مشغول نموده است. آنها تکامل تدریجی

---

[1] Cellular Automata (CA)

پدیده های فیزیکی را با استفاده از قوانین CA بصورت گام به گام شبیه سازی می نمایند. در صورتی که طبق تعریف یک مدل محاسباتی مکانیزمی برای توصیف تکامل تدریجی پدیده ها نمی باشد، بلکه ساختاری است که برای انجام محاسبات برروی داده های ورودی بکار می رود. در این مقاله به CA از دید یک ماشین محاسباتی عمومی نگریسته شده و الگوریتمی برای حل مساله رنگ امیزی گراف توسط CA ارایه شده است. برای حل مسایل محاسباتی به یک ساختمان داده نیاز میباشد. این ساختمان داده شامل ورودی و خروجی و یک رویه برای تبدیل ورودی به خروجی می باشد. مراحل پردازش و تبدیل ورودی به خروجی در CA توسط انتقال حالتهای CA پیاده سازی می شود. هر چند در تعریف CA استاندارد، حافظه منظور نشده است ولی اگر CA خواسته باشد نقش یک ماشین محاسبه گر عمومی را بازی کند اولاً هر سلول اتوماتا نیاز به تعدادی حافظه برای نگهداری مقادیر ورودی و خروجی داشته و ثانیا اتوماتا قابلیت خواندن مقادیر ورودی و قابلیت نوشتن مقادیر(از/ به) حافظه را داشته باشد.

همسایگی Moor همسایگی von Neumann

شکل ۱ : دو نوع همسایگی در CA دوبعدی

تعریف ارایه شده در زیر که در این مقاله بعنوان تعریف CA در نظر گرفته شده است، اصلاح شده تعریف اتوماتای میلی[۲] [8] می باشد.CA یک ۷ تایی به صورت $(Q, d, V, \Sigma, \Delta, \delta, \lambda)$ می باشد که در آن:

۱- Q مجموعه حالاتی است که هر سلول می تواند اختیار کند.

۲- d ابعاد فضای سلولی را مشخص می نماید. اگر $d = 2$ باشد در این صورت یک CA دو بعدی خواهیم داشت.

۳- برای هر سلول x در CA ، آرایه $V(x)=\{x+v_0, x+v_1, \cdots, x+v_k\}$ مشخص کننده k+1 همسایه ای می باشد که بصورت مستقیم با سلول در ارتباطند.

٤- Σ الفبای ورودی CA می باشد.

۵- Δ الفبای خروجی CA می باشد.

٦- δ تابع انتقال است که بفرم $\delta:(Q\times\Sigma^n)^{k+1}\rightarrow Q$ می باشد. بر اساس تابع انتقال حالت بعدی هر سلول به حالت و مقادیر حافظه ای ورودی تمامی همسایگان ان سلول در مرحله فعلی بستگی دارد. n تعداد رجیسترهای ورودی و خروجی هر سلول است .

۷- λرابطه مبدل[۳] است که زیر مجموعه متناهی از $(Q\times\Sigma^n)^{k+1}\times\Delta^n$ می باشد . این مبدل مقدار هر حافظه خروجی سلول را با توجه به حالت و مقادیر حافظه ای ورودی همسایگانش مشخص می سازد. در این جا هر سلول CA در همان حافظه هایی می نویسد که از آنها می خواند و در نتیجه $\Sigma=\Delta$ می باشد.

---

۲ Mealy Automaton

۳ Transducer Relation

۳- الگوریتم‌های رنگ‌آمیزی موازی گراف

سیاست رنگ‌آمیزی گره‌ها بستگی به هدف از رنگ‌آمیزی دارد. اگر هدف رنگ‌آمیزی، رنگ‌آمیزی بصورت بهینه باشد معمولاً کوچکترین رنگ در اختیار انتخاب می‌شود. رنگ در اختیار رنگی است که تا کنون توسط هیچ کدام از گره‌های همسایه استفاده نشده است. اگر هدف رنگ‌آمیزی بصورت متوازن باشد، رنگ کمترین استفاده شده در اختیار به کار برده می‌شود. این رنگ رنگی است که تا بحال کمتر در گراف مورد استفاده قرار گرفته و همچنین توسط هیچ یک از همسایگان گره نیز انتخاب نشده باشد. در این صورت توازنی بین رنگهای گراف بوجود می‌آید.

در این بخش به معرفی چند الگوریتم موازی و محلی برای رنگ آمیزی گراف می پردازیم. الگوریتم‌های موازی گراف بر اساس این اصل ساده بنا شده اند که هر مجموعه مستقل از گره هارا می‌توان به صورت موازی رنگ‌آمیزی نمود. یک مجموعهٔ مستقل مجموعه ای از گره‌هاست که هیچ دوتایی از آنها در مجاورت همدیگر نیستند. چگونگی رنگ‌آمیزی گراف بر اساس این ایده در شکل ۲ نشان داده شده است. در این جا U نشانگر مجموعه گره‌های رنگ نشده گراف می‌باشد. تفاوت الگوریتم‌های موازی در چگونگی انتخاب و رنگ‌آمیزی مجموعه‌های مستقل می‌باشد. الگوریتمهای که از بزرگترین مجموعه مستقل استفاده میکند بصورت کلی توسط الگوریتم شکل ۲ بیان می شوند.

```
U:=V
While ( |U| > 0 ) do in parallel
      choose as independent set I from U
      color all vertices in I
      U:=U-I
end do
```

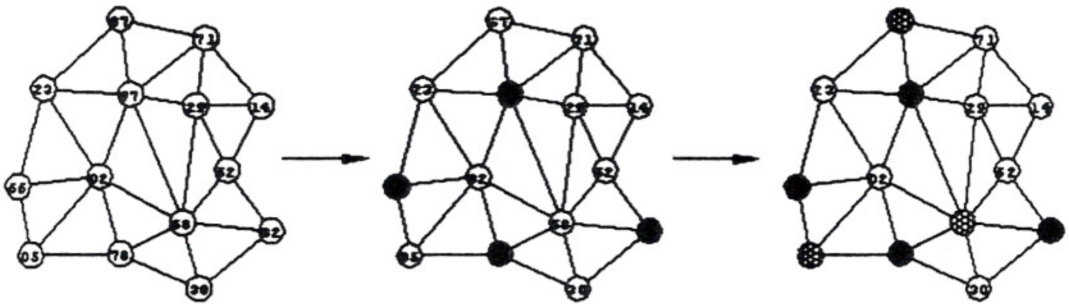
شکل ۲: رنگ‌آمیزی گراف بصورت موازی

در ادامه به بررسی الگوریتمهایی می‌پردازیم که از مفهوم مجموعه مستقل استفاده نموده و در ضمن محلی می‌باشند. یعنی برای رنگ‌آمیزی هر گره تنها به اطلاعات گره‌های همسایه نیاز دارند.

بزرگترین مجموعه مستقل[۴]: الگوریتم بزرگترین مجموعه مستقل (MIS) یک گراف را بصورت تکراری و با یافتن بزرگترین مجموعه مستقل از گره‌هادر گراف رنگ‌آمیزی می‌نماید[12]. الگوریتم در مرحله اول بزرگترین مجموعه مستقل از گراف را پیدا نموده و به همه آنها رنگی یکسان اختصاص داده و سپس آنها را از گراف حذف می‌نماید. در مرحله بعد الگوریتم یک MIS جدید یافته و رنگ بعدی را به گره‌های آن اختصاص می‌دهد و این کار را تا هنگامی‌که تمامی‌گره‌های گراف رنگ‌آمیزی شوند ادامه می‌دهد. یک نسخه از الگوریتم موازی MIS توسط Luby ارائه شده است [12].

---

[۴] Maximal Independent Set



شکل ۳ : رنگ‌آمیزی گراف با الگوریتم جونز-پلاسمن

الگوریتم جونز-پلاسمن[۵]: جونز و پلاسمن یک الگوریتم موازی رنگ‌آمیزی مبتنی بر الگوریتم MIS موازی ارائه نموده اند[10]. در الگوریتم آنها نیازی به یافتن یک جایگشت تصادفی از گره‌ها به منظور ایجاد یک مجموعه مستقل نمی‌باشد. در این الگوریتم یک مجموعه از وزنهای تصادفی یکتا در ابتدای کار ساخته می‌شود و در خلال رنگ‌آمیزی از آن استفاده می‌شود. برای ساختن این مجموعه به هر گره یک عدد تصادفی یکتا نسبت می‌دهیم. این الگوریتم دارای تفاوتهایی نسبت به الگوریتم MIS موازی می‌باشد. این الگوریتم تنها یکبار در ابتدای کار یک مجموعه مستقل را به صورت موازی می‌یابد. تفاوت دیگر این است که به گره‌های مجموعه مستقل رنگ یکسان و مشابه انتساب داده نمی‌شود و در عوض گره‌ها با کوچکترین رنگ در اختیار رنگ‌آمیزی می‌شوند. در این الگوریتم ابتدا به هر یک از گره‌ها وزنی تصادفی انتساب داده می‌شود. حال در هر مرحله گره‌هایی که دارای بیشترین وزن در میان همسایگان رنگ نشده خود باشند، با استفاده از کوچکترین رنگ در اختیار رنگ‌آمیزی می‌شوند و این روال تا رنگ شدن همه گره‌ها ادامه می‌یابد. رنگ‌آمیزی نهایی مشابه آنچه که الگوریتم آزمند تولید می‌کند، می‌باشد. در شکل ۳ با یک مثال نحوه رنگ‌آمیزی گراف توسط این الگوریتم نشان داده شده است.

الگوریتم بزرگترین درجه ابتدا[۶]: این الگوریتم بسیار شبیه به الگوریتم جونز پلاسمن می‌باشد[19]. در این الگوریتم در عوض بکارگیری وزنهای تصادفی برای ایجاد مجموعه‌های مستقل، از درجه هر گره به عنوان وزن گره استفاده می‌شود و اعداد تصادفی فقط برای حل مشکل برخورد بین گره‌های همسایه دارای درجه مساوی استفاده می‌شوند. در این الگوریتم گره‌ها بصورت تصادفی رنگ‌آمیزی نمی‌شوند بلکه به ترتیب نزولی درجه گره رنگ‌آمیزی می‌شوند.

الگوریتم کوچکترین درجه آخر[۷]: الگوریتم آخر کوچکترین درجه تلاش می‌کند که بر اساس الگوریتم بزرگترین درجه اول و بکارگیری یک سیستم وزن دهی پیچیده تر به رنگ امیزی بهتری دست یابد[14]. این الگوریتم در دو فاز عمل می‌کند: فاز وزن‌دهی و فاز رنگ‌آمیزی. فاز وزن‌دهی با یافتن تمام گره‌های با درجه برابر با کوچکترین درجه گراف (d) شروع می‌شود. به این گره‌ها وزن فعلی انتساب داده می‌شود و از گراف حذف می‌شوند و بنابراین درجه همسایگانشان را تغییر می‌دهند. این الگوریتم در هر مرحله گره‌های با درجه d را حذف می‌کند و در هر مرحله یک واحد به وزن اضافه می‌نماید. هنگامی‌که هیچ گره با درجه d در گراف باقی نماند الگوریتم به جستجوی گره‌های با درجه d+1 می‌پردازد. این روند تا هنگامی‌ادامه می‌یابد که به تمام گره‌ها وزنی نسبت داده شود. هنگامی که انتساب

---

[۵] Jones-Plassman Algorithm

[۶] Largest Degree First Algorithm

[۷] Smallest Degree Last Algorithm

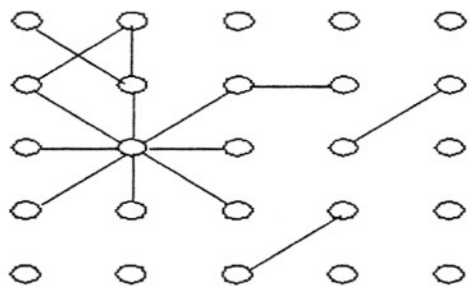
وزنها انجام گرفت رنگ‌آمیزی همانند الگوریتم‌های جونز-پلاسمن و اول بزرگترین درجه ادامه می‌یابد. در فاز رنگ‌آمیزی هر گره به تمام همسایگان رنگ نشده اش می‌نگرد و هنگامی‌که دریابد بالاترین وزن را دارد خودش را با بکارگیری کوچکترین رنگ در حوزه همسایگی اش رنگ می‌نماید.

٤- یک الگوریتم مبتنی بر CA برای رنگ‌آمیزی گراف

در این قسمت الگوریتم پیشنهادی را شرح می دهیم. ابتدا ایده الگوریتم را بیان نموده و سپس نحوه نگاشت گراف توری در CA را توضیح می دهیم. در ادامه قوانین مطلوب برای انجام الگوریتم را طراحی می نماییم. در این الگوریتم ابتدا به هر گره از گراف یک عدد تصادفی نسبت می‌دهیم. الگوریتم بصورت موازی عمل نموده به این صورت که در هر گام تمام گره‌های رنگ نشده عدد تصادفی مربوط به خود را با عدد تصادفی گره‌های رنگ نشدهدر مجاورت خود مقایسه می‌کنند. گره در صورتی رنگ می‌شود که عدد تصادفی آن از تمام اعداد تصادفی مربوط به گره‌های رنگ نشده مجاور بزرگتر باشد. رنگ گره برابر با کوچکترین رنگی خواهد بود که توسط گره‌های رنگ شده مجاور اختیار نشده است. این عمل تا زمانیکه تمام گره ها رنگ نشده اند ادامه می‌یابد. از آنجا که تمام عملیات این الگوریتم ماهیت محلی دارند در صورتی که گراف توری را در CA نگاشت کرد پیاده سازی الگوریتم براحتی قابل انجام میباشد.

٤-١- پیاده سازی الگوریتم توسط CA

در این قسمت نحوه پیاده سازی الگوریتم با همسایگی Moore شرح داده می‌شود. به این منظور ابتدا گراف بایستی در CA نگاشت شود. گروهی از گرافها که به نام گراف های توری موسوم هستند را می‌توان بسادگی در CA دوبعدی نگاشت نمود. یک گراف را یک گراف توری مینامیم اگر بتوان آن را در یک شبکه توری (شبکه ای توری که در ان هر ند به هشت همسایه خود متصل است) نگاشت کرد بطوریکه هر دو گره در این گراف که در مجاورت همدیگر قرار دارند به دو ند در مجاورت همدیگر در شبکه توری نگاشت شوند(شکل ٤).



شکل٤ : گراف توری

گراف توری را می‌توان به سادگی در CA دو بعدی با همسایگی Moore نگاشت نمود. هر گره از گراف به یکی از سلولهای CA نگاشت میشود. البته هر سلول CA متناظر با یک گره گراف نمی‌باشد. سلولهایی را که متناظر با هیچکدام از گره های گراف نیستند را با حالت None-Node مشخص می‌نماییم.

حال باید مکانیسمی‌بیابیم که توسط آن مشخص شود میان کدام گره‌ها لبه وجود دارد. برای مشخص نمودن لبه‌ها می‌توان از هشت متغیر حافظه ای یک بیتی N ، S ، E ، W ، NE ، NW ، SE و SW استفاده نمود. یک بودن هر یک از این متغیرها در سلول نشان دهنده آن است که لبه ای بین این سلول با به ترتیب همسایگان شمالی، جنوبی،

شرقی، غربی، شمال شرقی، شمال غربی، جنوب شرقی و جنوب غربی وجود دارد و صفر بودن این بیت‌ها نشان دهنده عدم وجود لبه در این موارد می‌باشد. بدیهی است در مورد سول های None-Node مقدار این متغیرهای حافظه ای بی اهمیت است.

فرض می‌کنیم یک گراف توری در یک CA دوبعدی با همسایگی Moore نگاشت شده است. حال به شرح الگوریتم پیشنهادی میپردازیم. برای هر سلول یک متغیر حافظه ای بنام Number می‌گیریم که در وضعیت اولیه (در ابتدای الگوریتم) به این متغیر یک عدد تصادفی نسبت داده می‌شود. حالت سلولهای متناظر با گره‌های گراف در وضعیت اولیه U-Node می‌باشد. بدین معنی که هیچ کدام از گره های گراف هنوز رنگ نشده اند. در طول اجرای الگوریتم حالت هر گره ای که رنگ شود به C-Node تغییر می‌یابد. به منظور رنگ‌آمیزی گره‌ها برای هر سلول یک متغیر حافظه ای دیگر به نام Color تعریف می‌کنیم که معرف رنگی است که آن گره با آن رنگ‌آمیزی شده است.

CA به این طریق عمل می‌کند که در هر گام هر سلول U-Node متغیر Number مربوط به همسایه‌های با حالت U-Node خود را بررسی می‌نماید. اگر همه آنها کوچکتر از متغیر Number خود سلول باشند سلول می‌بایستی رنگ‌آمیزی شود. شماره رنگ سلول برابر با کوچکترین شماره غیر منفی خواهد بود که توسط همسایه‌های با حالت C-Node سلول اختیار نشده باشد. پس از رنگ شدن گره حالت آن به C-Node تبدیل می‌شود. در غیر این صورت (یعنی متغیر Number بزرگترین مقدار در بین همسایه‌های U-Node نباشد) حالت سلول بدون تغییر باقی خواهد ماند. در ضمن اجرای الگوریتم حالت سلولهای C-Node و None-Node بدون تغییر می‌ماند. این الگوریتم تا زمانی ادامه پیدا خواهد کرد که تمامی‌سلولهای با حالت U-Node حالت خود را به C-Node تغییر دهند.

ساختار هر سلول: در این الگوریتم هر سلول دارای متغیرهای حافظه ای زیر می‌باشد.

- متغیر State که حالت هر سلول را مشخص می‌کند و می‌تواند مقادیر زیر را انتخاب نماید:
  - None-Node : سلولهایی که متناظر با هیچ گره ای از گراف نمی باشند دارای حالت one-Node می‌باشند.
  - U-Node : سلولهای متناظر با گره‌های رنگ نشده دارای حالت U-Node می‌باشند.
  - C-Node : سلولهای متناظر با گره‌های رنگ شده دارای حالت C-Node می‌باشند.
- متغیر Number: این متغیر در وضعیت اولیه یک عدد تصادفی اختیار نموده و تا پایان الگوریتم بدون تغییر باقی می‌ماند.
- متغیر N : اگر لبه ای بین گره متناظر با سلول و گره متناظر با همسایه شمالی سلول وجود داشته باشد متغیرN دارای مقدار یک و در غیر این صورت دارای مقدار صفر خواهد بود.
- متغیر S : اگر لبه ای بین گره متناظر با سلول و گره متناظر با همسایه جنوبی سلول وجود داشته باشد متغیر S دارای مقدار یک و در غیر این صورت دارای مقدار صفر خواهد بود.
- متغیر W : اگر لبه ای بین گره متناظر با سلول و گره متناظر با همسایه غربی سلول وجود داشته باشد متغیر W دارای مقدار یک و در غیر این صورت دارای مقدار صفر خواهد بود.
- متغیر E : اگر لبه ای بین گره متناظر با سلول و گره متناظر با همسایه شرقی سلول وجود داشته باشد متغیر E دارای مقدار یک و در غیر این صورت دارای مقدار صفر خواهد بود.
- متغیر NW : اگر لبه ای بین گره متناظر با سلول و گره متناظر با همسایه شمال غربی سلول وجود داشته باشد متغیر NW دارای مقدار یک و در غیر این صورت دارای مقدار صفر خواهد بود.

- متغیر NE : اگر لبه ای بین گره متناظر با سلول و گره متناظر با همسایه شمال شرقی سلول وجود داشته باشد متغیر NE دارای مقدار یک و در غیر این صورت دارای مقدار صفر خواهد بود.
- متغیر SW : اگر لبه ای بین گره متناظر با سلول و گره متناظر با همسایه جنوب غربی سلول وجود داشته باشد متغیر SW دارای مقدار یک و در غیر این صورت دارای مقدار صفر خواهد بود.
- متغیر SE : اگر لبه ای بین گره متناظر با سلول و گره متناظر با همسایه جنوب سلول وجود داشته باشد. متغیر SE دارای مقدار یک و در غیر این صورت دارای مقدار صفر خواهد بود.

وضعیت اولیه CA: زمانیکه یک گراف توریدر CA نگاشت میشود سلولهای متناظر با هر گره گراف حالت U-Node و بقیه سلولها حالت Non-Node دارند. متغیر Number در وضعیت اولیه در مورد سلولهای U-Node یک مقدار تصادفی خواهد داشت و در مورد سلولهای Non-Node بی اهمیت است. مقدار متغیر Non-Node در مورد همه سلولها در وضعیت اولیه بی اهمیت است. مقدار متغیرهای N ، S ، E ، W ، NE ، NW ، SE و SW در مورد سلولهای U-Node معرف لبه‌های متصل به گره می‌باشد.

وضعیت نهایی CA : در وضعیت نهایی تمام سلولهای متناظر با گره‌های گراف رنگ‌آمیزی شده اند. به این معنا که مقدار متغیر State آنها به C-Node تغییر پیدا کرده است. مقدار متغیر Color هر سلول آنها معرف رنگ گره متناظر با آن سلول خواهد بود.

قوانین CA:

- اگر سلول دارای حالت Non-Node باشد، در گام بعد سلول بدون تغییر باقی می‌ماند.
- اگر سلول دارای حالت C-Node باشد، در گام بعد سلول بدون تغییر باقی می‌ماند.
- اگر سلول دارای حالت U-Node باشد حالتهای زیر متصور است:

•اگر متغیر Number مربوط به سلول از متغیر Number مربوط به یکی از سلولهایU-Node همسایه کوچکتر باشد، حالت سلول در گام بعد بدون تغییر باقی می‌ماند.

•اگر متغیر Number سلول از متغیر Number تمام سلولهای U-Node همسایه اش بزرگتر باشد آنگاه حالت سلول C-Node شده و مقدار متغیر Color آن برابر با کوچکترین عدد مثبتی خواهد بود که توسط همسایه‌های با حالت C-Node سلول اختیار نشده باشد.

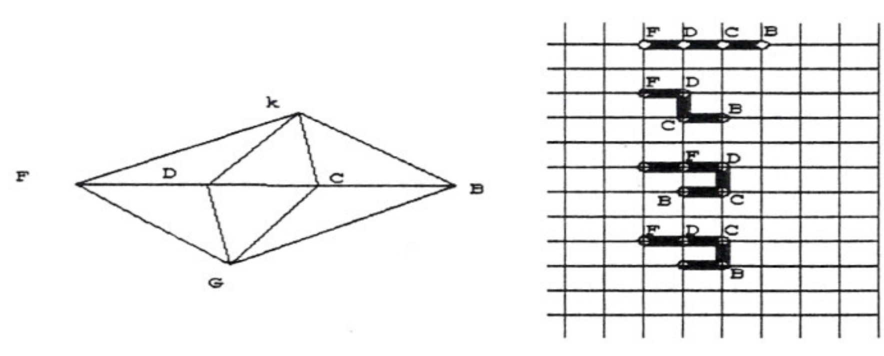
پیچیدگی الگوریتم: بدترین حالت الگوریتم از نظر زمانی هنگامی‌رخ می‌دهد که در هر گام فقط یک گره گراف رنگ شود. در این صورت برای رنگ‌آمیزی یک گراف دارای n گره به n گام زمانی نیاز خواهیم داشت. حال باید دید که ایا در این الگوریتم چنین حالتی پیش می‌آید یا نه؟ در صورتی که هر گره گراف دارای هشت همسایه بوده و اعداد تصادفی مربوط به گره‌ها به صورت مارپیچی مرتب شده باشند، حالت فوق پیش می‌آید. در این صورت در گام اول فقط گره دارای بزرگترین عدد تصادفی، در گام دوم گره دارای دومین عدد تصادفی بزرگ،... و در گام آخر فقط گره دارای کوچکترین عدد تصادفی رنگ می‌شود و بدین ترتیب برای رنگ نمودن یک گراف توری دارای n گره به n گام زمانی نیاز می‌باشد. لذا پیچیدگی الگوریتم برای رنگ‌آمیزی گراف‌های توری نگاشت شده در یک CA دوبعدی $n \times n$ از مرتبه n^2 خواهد بود. برای کم نمودن این زمان می‌توان از روش بکار برده شده در الگوریتم مرتب سازی گزارش در [23] استفاده نمود و خاتمه الگوریتم را با توجه به داده‌های ورودی مشخص نمود.

۵- نگاشت مش‌های مثلثی[۸] بر روی CA دو بعدی

ما تلاش نمودیم تا انواع دیگر گرافها را بر روی CA نگاشت نماییم و برای این کار مش‌های مثلثی را انتخاب نمودیم. مش مثلثی یک گراف مسطح است که از اتصال گرافهای متصل که دارای سه گره میباشند حاصل شده باشند. مش مثلثی در زمینه‌هایی نظیر شبیه‌سازی جاذبه ذرات و حل کننده‌های معادلات با مشتقات جزئی کاربرد دارد. برای نگاشت نمودن مش در CA دو بعدی آن دسته از مش‌ها مثلثی را در نظر می‌گیریم که حداکثر درجه هر گره برابر هشت باشد. برای نگاشت نمودن مش‌های دیگر می‌توان از CA با ابعاد بیشتر استفاده نمود. بسیاری از مش‌های مثلثی را می‌توان در CA دوبعدی نگاشت نمود. ولی مش‌هایی نیز وجود دارند که قابل نگاشت در CA نمی‌باشند. یک نمونه از این مش‌ها در شکل ۵-الف نشان داده شده است. در این مش هر دو گره G و K با چهار گره C، D، F وB در ارتباط هستند. در شکل ۵-ب تمامی‌حالتهای ممکن برای نگاشت گره‌هایC، D، F وB در CA نمایش داده شده است. در هیچ کدام از حالتهای ذکر شده نمی‌توان محلی مناسب را برای گره G یا K به صورتی پیدا نمود که با هر چهار گره مورد نظر در ارتباط باشند و در نتیجه میتوان نتیجه گرفت که هر مش مثلثی الزاماً قابل نگاشت در CA نمی‌باشد. پیدا کردن شرایط لازم و کافی برای قابل نگاشت بودن مش‌های مثلثی در CA بنظر مساله جالبی است که ارزش فکر کردن دارد.

۶- نتیجه گیری

در این مقاله الگوریتمی مبتنی بر اتوماتای سلولی برای رنگ آمیزی گراف توری ارائه شد. الگوریتم پیشنهادی که ساده ومحلی و منطبق با خواص حاکم بر CA می باشد یک گراف توری را با استفاده از حداقل تعداد رنگ، رنگ آمیزی می نماید.



ب

الف

شکل ۵

۷-مراجع

[1] K. Apple and W. Haken, "Every Planar Map is Four Colourable," Illinois Journal of Mathematics, Vol. 21, p. 429, 1977.

[2] D. Brelaz, "New Methods to Color the Vertices of a Graph," Communications of the ACM, Vol. 22, p. 251, 1979.

[3] W. Burks, Essays on Cellular Automata, Urbana, IL: University of Illinois Press, 1970.

[4] T. F. Coleman, and J. J. More, "Estimation of Sparse Jacobian Matrices and Graph Coloring Problems," SIAM Journal of Numerical Analysis, Vol. 20, p. 187, 1983.

---

[۸] Tringulated mesh

[5] J. H. Conway, E. Berlekamp and R. Guy, Winning Ways for Your Mathematical Plays, Vol. 2, Academic Press, 1982.

[6] K. Culik, L. Hurd, and S. Yu, "Computation Theoritic Aspects of Cellular Automata," Physica D, Vol. 45, pp. 357-378, 1990.

[7] D. Farmer, T. Toffoli and S. Wolfram, Cellular Automata, Proceedings of an Interdisciplinary Workshop, Amsterdam, North Holland, 1984.

[8] L. Gordillo and V. Luna, "Parallel Sort on a Linear Array of Cellular Automata, " IEEE Transaction on Computers, pp. 1904-1910, 1994.

[9] A. H. Gutowitz, Cellular Automata, Cambridge, MA: MIT Press, 1990.

[10] M. T. Jones and P. E. Plassmann, "A Parallel Graph Coloring Heuristic," SIAM Journal of Scientific Computing, Vol. 14, p. 654, 1993.

[11] R. J. Lipton and R. E. Miller, "A Batching Method for Coloring Planar Graphs," Information Processing Letters, Vol. 7, p. 185, 1978.

[12] M. Luby, "A Simple Parallel Algorithm for the Maximal Independent Set Problem," SIAM Journal on Computing, Vol. 4, p. 1036, 1986.

[13] M. Mitchel, "Computation in Cellular Automata: A Selected Review," Technical Report, Santa Fe Institute, Santa Fe, New Mexico, 1996.

[14] D. W. Matula, G. Marble and J. D. Issacson, Graph Coloring Algorithms, Acaddemic Press, New York, 1972.

[15] N. Packard, "Two-Dimensional Cellular Automata," Journal of Statistical Physics, Vol.30, pp. 901-942, 1985.

[16] P. Sarkar, "Brief History of Cellular Automata," ACM Computing Surveys, Vol. 32, No. 1, 2000.

[17] N. A. Sherwani, Algorithm for VLSI Physical Design Automation, Western Michigan University Kluwer Academic Publishers, 1993.

[18] T. Toffoli and N. Margolus, Cellular Automata Machines: A New Environment for Modeling, Cambridge, MA: MIT Press, 1987.

[19] D. J. A. Welsh, and M. B. Powell, "An Upper Bound for the Chromatic Number of a Graph and its Application to Timetabling Problems," Computing Journal, Vol. 10, 1967.

[20] S. Wolfram, "Statistical Mechanics of Cellular Automata," Review of Modern Physics, Vol. 55, pp. 601-644, 1983.

[21] S. Wolfram, "Computation Theory of Cellular Automata," Communication in Mathematical Physics, Vol. 96, pp. 15-57, 1984.

[22] S. Wolfram, Theory and Applications of Cellular Automata, Singapore: World Scientific, 1986.

[23] M. R. Meybodi and S. Golzari, A Parallel Sorting Algorithm for Two Dimensional Cellular Automata, Technical Report, Computer Engineering Department, Amirkabir University, 2001

دهمین کنفرانس مهندسی برق ایران

۲۴ الی ۲۶ اردیبهشت ۱۳۸۱

مجموعهٔ مقالات

الکترونیک، کامپیوتر و مهندسی پزشکی

جلد اول

گروه مهندسی برق
دانشکده فنی
دانشگاه تبریز