

## یک الگوریتم مسیر یاب ماز مبتنی بر اتوماتای سلولی دو بعدی

محمدرضا میبدی شهram گلزاری

ازمايشگاه سيسitem هاي نرم افزاري

دانشکده مهندسي کامپیوتر

دانشگاه صنعتی امیرکبیر

تهران- ایران

**چکیده:** در این مقاله یک الگوریتم مسیریاب ماز مبتنی بر اتوماتای سلولی (CA) دو بعدی ارائه شده است. هدف از این الگوریتم یافتن مسیری با طول کمینه از سلول مبدأ به سلول مقصد است به طوری که مسیر از موانع عبور ننماید. الگوریتم دارای دو فاز اکتشاف و رديابي مجدد می باشد. قوانین اتوماتا به صورتی طراحی شده اند که در فاز اکتشاف از سلول مبدأ موجی منتشر شود و در حین انتشار از سلولهایی که می گذرد ردی از خود بر جای گذارد. در فاز رديابي مجدد از سلول مقصد شروع کرده و رد به جای گذاشته توسط موج را دنبال کرده تا به سلول مبدأ برسیم که مسیر طی شده در این فاز مسیر مطلوب خواهد بود. الگوریتم پیشنهادی ساده بوده و تمام تراکنشهای آن محلی و منطبق با خواص حاکم بر اتوماتای سلولی می باشد. این الگوریتم مسیر مطلوب را در یک CA دو بعدی  $m \times m$  در  $O(m^2)$  گام زمانی می یابد.

**كلمات کلیدی:** اتوماتای سلولی، مسیریابی، الگوریتمهای مسیریاب ماز، طراحی فیزیکی، الگوریتم موازی

### - مقدمه

الگوریتمهای مسیریاب ماز کوتاهترین مسیر بین دو نقطه مبدأ<sup>(S)</sup> و مقصد<sup>(T)</sup> را در یک گراف توری مستطیلی مسطح<sup>(3)</sup> پیدا می کنند. به منظور انجام مسیریابی توسط این الگوریتمها ابتدا کل طرح (فضای که می خواهیم در آن عملیات مسیریابی انجام دهیم) را بسادگی توسط گراف توری مدل می نمایند. برای انجام این کار کل طرح را بوسیله مجموعه ای از سلولهای مربعی با سطح واحد که در یک آرایه دو بعدی قرار گرفته اند، نمایش می دهند. درست مثل اینکه یک توری روی طرح انداخته باشیم.

حال با استفاده از این سلولها و ارتباطات بین آنها گراف را بدست می اوریم. بدین صورت، که هر سلول  $i,j$  بوسیله یک گره  $v_{i,j}$  در گراف نمایش داده می شود. گره های متناظر با سلولهایی که شامل مانع هستند را گره های مسدود و بقیه گره ها را گره های غیر مسدود می نامند. در صورتی که سلولهای  $i,j$  و  $i,k$  با هم همسایه باشند بین گره های  $v_{i,j}$  و  $v_{i,k}$  یک گراف لبه وجود دارد. وزن هر لبه در گراف برابر با یک در نظر گرفته می شود بجز لبه های مرتبط با گره های مسدود که وزن این گونه گره ها برابر صفر در نظر گرفته می شود. با توجه به توضیحات فوق شکل ۱-۱

گراف توری مربوط به طرح شکل ۱-۱ را نمایش می دهد.

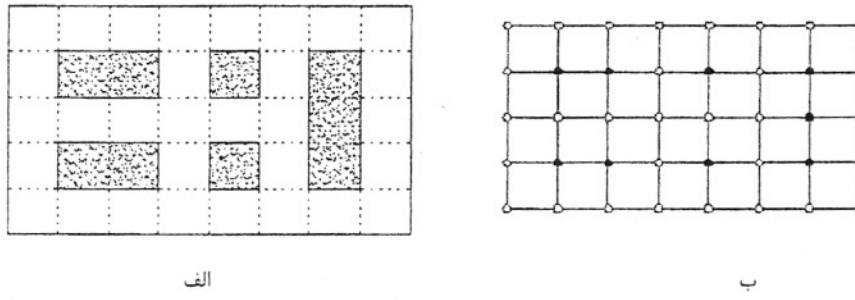
در گراف توری مستطیلی مسطح هر گره با چهار گره همسایه در ارتباط می باشد. گره های متناظر با نواحی از طرح که می توان عمل مسیر یابی را از طریق آنها انجام داد را با حالت غیرمسدود و بقیه گرد ها را با حالت مسدود نمایش می دهند. هدف الگوریتمهای مسیریاب ماز، پیدا کردن مسیری بین گره های مبدأ و مقصد می باشد بطوری که این مسیر از هیچ یک از گره های با حالت مسدود نگذشته و در ضمن کمترین طول را نیز

 Source<sup>۱</sup>

 Target<sup>۲</sup>

 Planner Rectangular Grid Graph<sup>۳</sup>

دارا بشد، این دسته الگوریتمها دارای دو فاز اکتشاف<sup>۱</sup> و ریدیابی مجدد<sup>۲</sup> می‌باشند و الگوریتم با فاز اکتشاف آغاز می‌شود. در این فاز از گره مبدأ شروع کرده، تمام مسیرهای ممکن را که از آن آغاز می‌شوند در نظر گرفته و تمامی آنها را ادامه می‌دهیم تا تکی از آن مسیرها به گره مقصد برسد.



شکل ۱: گراف توری

هنگامی که گره مقصد مورد دسترسی قرار گرفت، فاز ریدیابی مجدد آغاز می‌شود. در این فاز با استفاده از روش بازگشت به عقب<sup>۳</sup> شخص می‌شود که کدام گره‌های گراف در مسیر متصل کننده گره‌های مبدأ و مقصد قرار دارند. پس از انجام این فاز الگوریتم خاتمه می‌یابد. برای پیاده‌سازی فاز ریدیابی مجدد لازم است که در فاز اکتشاف اطلاعاتی در مورد مسیرها در هر یک از گره‌ها قرار گیرد و سپس با استفاده از این اطلاعات فاز ریدیابی مجدد انجام می‌گیرد[15, 7, 8, 11]. الگوریتمها مسیریاب ماز در یافتن مسیر ربات و فاز مسیریابی از طراحی فیزیکی VLSI کاربرد دارند[15, 8].

اولین الگوریتم از این دسته الگوریتمها توسط Lee معرفی شده که کوتاهترین مسیر را در یک گراف توری با ابعاد  $w \times h$  در زمان  $O(h \times w)$  می‌یابد[8] و پس از آن الگوریتم‌های متعددی بر مبنای الگوریتم Lee و برای بهمود در نحوه گسترش مسیر و زمان اجرای<sup>۴</sup> الگوریتم گزارش شده است که از آنچه می‌توان به الگوریتم‌های [14] و Hadlock [7] اشاره نمود. یکی از خصوصیات جالب الگوریتم‌های مسیریاب ماز، مستر بودن نوعی توازنی در آنها می‌باشد. این نکته موجب شده است که این الگوریتمها بر احتی بر روی ساختارهای موازی نگاشت شده و در عین ثابت بودن پیچیدگی الگوریتم، زمان اجرای الگوریتم به دلیل انجام شدن الگوریتم توسط سخت‌افزار مناسب و کم شدن تعداد دستورالعملهای انجام شده توسط هر بردازنده تا حد زیادی کاهش یابد. تعدادی از این نگاشتها توسط Massara و Sagar گزارش شده است[19].

در این مقاله یک الگوریتم مسیریاب ماز مبتنی بر CA ارائه می‌شود. الگوریتم مطرح شده بسیار ساده می‌باشد. در این الگوریتم هر سلول CA دستورالعملهای بسیار اندکی را اجرا می‌کند که موجب کم شدن زمان اجرای الگوریتم می‌شود. ساختار مورد استفاده نیز دارای خواصی نظری سادگی، توازنی و محلی بودن می‌باشد که متناسب با الگوریتم‌های مسیریاب ماز است. در این الگوریتم ابتدا طرح را برروی CA دو بعدی نگاشت نموده و سپس با استفاده از قوانین ساده و محلی مناسب فازهای مسیریابی را پیاده سازی می‌نماییم. در این مقاله ابتدا در بخش ۲ در مورد مفاهیم اساسی CA صحبت خواهیم نمود و سپس در بخش ۳ به تشریح الگوریتم و نحوه پیاده سازی آن توسط CA می‌پردازیم.

## ۲-اتومات‌ای سلولی (CA)

یک شبکه منظم از ماشینهای با حالت محدود را در نظر بگیرید. به هر یک از این ماشینهای سلول<sup>۵</sup> می‌گوییم. هر کدام از این سلولها طبق الگوی ثابت و یکسانی با بعضی از سلولهای مجاور خویش در ارتباط هستند. این ارتباط محلی بوده و برای تمام سلولها یکسان است. خود سلول و تمام سلولهایی که با آن در ارتباط هستند، مجموعه همسایگان<sup>۶</sup> سلول را تشکیل می‌دهند. هر سلول در هر لحظه از زمان می‌تواند یکی از حالت‌های موجود در مجموعه حالت را اختیار نماید. مجموعه حالت برای تمامی سلولها یکسان می‌باشد. در هر لحظه از زمان حالت تمامی سلولها به صورت همزمان و بر اساس یک قانون تغییر می‌کند. 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 و با گذشت زمان شبکه یک رفتار پیچیده و جالب از خود تولید می‌نماید.

<sup>۱</sup>	Exploration
<sup>۲</sup>	Retrace
<sup>۳</sup>	Back Tracking
<sup>۴</sup>	Run Time
<sup>۵</sup>	Cell
<sup>۶</sup>	Neighborhood

با توجه به توصیحات فوق، وجه تأثیر CA نسبت به دیگر شکه های اقامتی عبارت است از: ساده بودن از این طبقات، محلى بودن از این طبقات، بین سلولها، برقراری الگوی ارتیاطی یکسان را تمام سلولها، بهینگام سازی همزمان سلولها، پرسپکت قانون یکسان و تولید رفتارهای جالب و پیچیده از سلولهای ساده. با توجه به این خصوصیات در اکثر مستندات از CA به عنوان ساختاری مسازی، محلی و یکسان: نام مستعدی از جمله تولید الگوهای تصادفی، نظریه محاسبات، مدل‌سازی سیستم های فیزیکی و بیولوژیکی و محاسبات کاربردی مورد استفاده قرار گرفته است [۱-۱۸].

نکات مهم در این نوع حساباتی می باشد. دونوع حساباتی عبارتند از حساباتی Moore و حساباتی von Neumann. در حساباتی Moore برای هر سلول هشت سلول همسایه و در حساباتی von Neumann چهار سلول همسایه در نظر گرفته می شود. در هر دو حساباتی سلول فوارمگر فرنئر در مرکز حساباتی بهینگام می شود. Packard CA تاکنون با استفاده از روش‌های مختلفی اثبات شده است که CA یک مدل محاسباتی عمومی می باشد.<sup>[10]</sup> ولی در عمل از این مدل اغلب برای شنبه سازی پدیده های فیزیکی برویا استفاده شده و این مدل بیش از آنکه مورد توجه دانشمندان علم کامپیوتر قرار گیرد. توجه پژوهشگران سایر علم مانند فیزیک و بیولوژی را به خود مشغول نموده است. آنها تکامل تدریجی پدیده های فیزیکی را با استفاده از قوانین CA بصورت گام به گام شنبه سازی می نمایند. در صورتی که طبق تعریف یک مدل محاسباتی مکانیزمی برای توصیف تکامل تدریجی پدیده های نامنی باشد، یا که

مہماں بھری Moor مہماں بھری von Neumann

شکل ۲: دو نوع همسایگی در CA در بعدی

در این مقاله به CA از دید یک ماشین محاسباتی عمومی نگریسته شده و الگوریتمی برای حل یک مسئله توسعه CA ارائه شده است. برای حل مسائل محاسباتی به یک ساختمان داده و یک روند نیاز می باشد. ساختمان داده ورودی و خروجی را در خود یک داشته و از رویه برای تبدیل ورودی به خروجی استفاده می شود. مراحل پردازش و تبدیل ورودی به خروجی در CA توسعه انتقال حالتی CA بیانه سازی می شود. هر چند در تعریف CA استاندارد، حافظه مظظر نشده است و اگر CA خواسته باشد نتشی یک ماشین محاسبه گر عمومی را بازی کند او لامر سلول ایتمان نیاز به تعدادی حافظه برای نگهداری مقادیر ورودی و خروجی داشته و ثانیاً اتوماتا باید قابلیت خواندن مقادیر ورودی و قابلیت نوشتن اضافه شده در نظر گرفته شده است. اصلاح شده مقادیر خروجی (از آبه) حافظه را داشته باشد. تعریف ارائه شده در زیر که در این مقاله بعنوان تعريف CA در این مقابله به CA از دید یک ماشین محاسباتی عمومی نگریسته شده و الگوریتمی برای حل یک مسئله توسعه CA ارائه شده است. برای

CA بک ایتائی به صورت  $(Q, d, V, \Sigma, \Delta, \delta, \lambda)$  می باشد که در آن:

۵- د باند فضای سلولی را مشخص می نماید. اگر  $2 = d$  باشد در این صورت یک CA دو بعدی خواهد داشت.

- ۳- برای هر سلول  $X$  در  $\mathcal{A}$  مشخص کنند  $V(X) = \{X + v_0, X + v_1, \dots, X + v_k\}$  ای می باشد که

**بصورت مسعيّم بـ سهول در اینستی:**

$\Delta - \Delta$  المبتدئ حروجي A می باشد.

١٢٣

نوجه به حالت و مقدار حافظه ای و رودی همسایگانش مشخص می سازد. در این جا هر سطول CA در همان حافظه هایی می نویسد که از آنها می باشد. این مبدل مقدار هر حافظه خروجی سطول را با  $\Delta = \sum_{i=1}^n Q_i$  می باشد. است که زیر مجموعه ممتهنی از  $\Delta$  می باشد.

### ۳- یک الگوریتم مسیریاب ماز برای CA دو بعدی

#### ۱-۳- ایده الگوریتم

اساس الگوریتم پیشنهادی مانند دیگر الگوریتم‌های مسیریاب ماز، الگوریتم در فاز اکتشاف از نقطه مبدأ موجی منتشر می‌شود. این موج در حین انتشار از محلهایی که می‌گذرد، ردی از خود بر جای می‌گذارد. در محلهایی که موج با مانع (سلولهای مسدود) برخورد می‌کند، حرکت موج از آن نقاط متوقف شده در حالی که حرکت موج از نقاط دیگر ادامه می‌یابد. زمانی که موج به مقصد می‌رسد فاز اکتشاف به پایان رسیده است. ولی موج همچنان به حرکت خود ادامه می‌دهد.

در فاز رديابي مجدد اين بار از مقصد شروع كرده و رد به جاي گذاشته توسيط موج را دنبال می کنيم تا به مبدأ برسيم. مسیر طی شده در فاز رديابي مجدد همان مسیر مطلوب خواهد بود که کوتاهترین مسیر از مبدأ تا مقصد می باشد. با توجه به آنکه هر دو فاز اکتشاف و رديابي مجدد با تراکنشهای محلی انجام پذير است، لذا با توجه به خصوصیت اصلی CA (محلي بودن تراکنشها) پياده سازی الگوریتم توسيط CA براحتی امكان پذير می باشد.

#### ۲-۳- پياده سازی الگوریتم توسيط CA

در اين قسمت نحوه پياده سازی الگوریتم توسيط CA تشریح می شود. به اين منظور می بايست ابتدا کل طرح در يك CA دو بعدی و با همسایگی Von Neumann نگاشت شود. در اين روش کل طرح به صورت مجموعه ای از سلوهای با مساحت واحد در نظر گرفته می شود. سلوهایی که موانع طرح را می پوشانند با حالت اولیه Block(B) در نظر می گيريم. حالت سلوهای متناظر با نقاط مبدأ و مقصد را بترتیب برابر Source(S) و Target(T) قرار داده و حالت بقیه سلوهای Free(F) قرار می دهیم. بدیهی است که مسیر موردنظر از میان سلوهای با حالت F می گذرد. بنابراین برای هر سلول به يك متغیر حافظه ای بنام State که نشاندهنده حالت آن سلول می باشد نیاز داریم. از اين به بعد منظور از سلول B، سلول S، سلول T و سلول F بترتیب سلول با حالت Block، سلول با حالت Source، سلول با حالت Target و سلول با حالت Free خواهد بود.

همانطور که توضیح داده شد در فاز اکتشاف می بايست از سلول مبدأ موجی تولید شود. برای پياده سازی تولید موج می توان بصورت زیر عمل نمود. در گام اول موج ياد شده تنها به سلوهای همسایه با سلول مبدأ می رسد. حال سلوهای F که موج به آنها رسیده است حالت خود را به تغییر می دهند. در گام بعد موج تمام سلوهای F که دارای يك همسایه M هستند را در بر می گيرد که اين سلوهای نیز حالت خود را M می کنند. عمل انتشار موج به همین مثال ادامه يافته و با اين روش فاز اکتشاف پياده سازی می شود. بنا به سناریوی فوق می توان انتشار موج را توسيط قانون ساده زير پياده سازی نمود: هر سلول F که دارای حداقل يك همسایه M یا S باشد، حالت خود را به M تغیير می دهد.

همانطور که در ایده الگوریتم ذکر شد، موج می بايست در محلهایی که از آن عبور می کند، ردی از خود بر جای گذارد تا در فاز رديابي مجدد، مسیر با توجه به آن ایجاد شود. در صورتی که سناریوی فوق اين عمل را انجام نداده و تنها موج را منتشر می کند. حال باید طرحی آنديشيد که همگام با انتشار موج، رد انتشار موج نیز در سلوهای ذخیره گردد.

همانطور که ديديم در هر مرحله که موج گسترش يبدا می کند تعدادی از سلوهای F تبدیل به M می شوند، در حقیقت سلوهای F که دارای همسایه های M یا S هستند، تبدیل به M می شوند. لذا برای ذخیره کردن رد موج کافی است بدانیم کدام همسایه سلول F، حالت M یا S داشته و اين اطلاع را در يك متغیر حافظه ای نگهداري کنيم. به اين منظور از يك متغیر حافظه ای بنام Direction استفاده می کنيم. اين متغير می تواند يكی از مقادير Up,Down,Left,Right Null را اختیار نماید. در وضعیت اولیه مقدار اين متغیر برای همه سلوهای F و T برابر Null و برای بقیه سلوهایها بی اهمیت است.

حال هر سلول F که دارای همسایه M یا S باشد، چک می کند آن همسایه در کدام طرف سلول قرار دارد و با توجه به آن متغير Direction خود را مقدار می دهد. بدین صورت که اگر همسایه M یا S در بالاپایین، چپ یا راست سلول باشد متغير Left بترتیب مقادير Up و Right را اختیار می نماید. در صورتی که سلول دارای بیش از يك همسایه M باشد متغير Direction آن به دلخواه مقدار متناظر با يکی از همسایه های M را اختیار می کند. سلول T نیز عملی مشابه با سلول F انجام می دهد، با اين تفاوت که حالت آن تغیير نمی کند و فقط متغير Direction آن مقدار می گيرد.

زناني که موج به سلول T می رسد (متغير Direction اين سلول مقدار مخالف Null می گيرد)، فاز اکتشاف به پایان رسیده و فاز رديابي مجدد شروع می شود. در اين فاز می بايست مسیر بصورت گام به گام از سلول T تا سلول S مشخص شده و سلوهای واقع شده در مسیر حالت Path(P) بخود بگيرند. حال ببينيم اين مسیر چگونه ایجاد می گردد. برای توضیح اين مطلب ابتدا مفهوم زير را تعریف می کنيم:

\*تعریف: نشانه رفتن سلول A به سلول B: گوییم سلول A به سلول B نشانه رفته است اگر يکی از موارد زير اتفاق افتاده باشد:

\*سلول A همسایه پایینی سلول B باشد و مقدار Direction سلول A برابر Up باشد.

\*سلول A همسایه بالای سلول B باشد و مقدار Direction سلول A برابر Down باشد.

\*سلول A همسایه سمت چپ سلول B باشد و مقدار Direction سلول A برابر Right باشد.

\*سلول A همسایه سمت راست سلول B باشد و مقدار Direction سلول A برابر Left باشد.

در فاز ریدیابی مجدد اگر سلول T یا P دارای یک همسایه M باشد و در ضمن به طرف آن همسایه M نیز نشانه رفته باشد، آنگاه آن همسایه M حالت P به خود می‌گیرد. نتیجه اعمال مطلب فوق به این صورت خواهد بود: زمانی که متغیر Direction مربوط به سلول T مقدار مخالف Null گرفت (یعنی زمانی که موج به سلول T رسید) فاز ریدیابی مجدد آغاز گشته است. بنابراین در اولین گام از فاز جدید، آن سلول M که سلول T به طرف آن نشانه رفته است حالت P به خود می‌گیرد و در گام بعد نیز آن سلول M که سلول P جدید به آن نشانه رفته است تبدیل به P می‌شود و این عمل تا رسیدن به سلول S ادامه می‌باید. سلولهای P مسیر مورد نظر را تشکیل می‌دهند. زمانی الگوریتم عملانجام یافته تلقی می‌شود که مسیر مورد نظر به سلول S برسد.(یعنی سلول S دارای یک همسایه T یا P باشد)

### ۳-۳- ساختار هر سلول

در الگوریتم پیشنهادی هر سلول دارای دو متغیر حافظه ای بنامهای State و Direction می‌باشد.

♦ متغیر State: حالت هر سلول را مشخص می‌کند و می‌تواند یکی از مقادیر زیر را اختیار نماید:

- سلول مبدأ دارای حالت S می‌باشد.

- سلول مقصد دارای حالت T می‌باشد.

- Block(B) : سلولهایی که مسیر نمی‌تواند از آنها گذر کند دارای حالت B می‌باشد.

- Free(F) : سلولهایی که مسیر می‌تواند از آنها بگذرد، دارای حالت F می‌باشد.

- Mark(M) : سلولهایی که موج به آنها می‌رسد دارای حالت M می‌شوند.

- Path(P) : سلولهایی که در مسیر نهایی قرار می‌گیرند دارای حالت P می‌شوند.

♦ متغیر Direction : این متغیر رد انتشار موج در هر سلول را مشخص می‌کند و می‌تواند یکی از مقادیر زیر را انتخاب نماید.

- Up : سلولهایی که موج از طریق همسایه بالایی به آنها رسیده است، مقدار متغیر Direction آنها برابر با Up می‌شود.

- Down : سلولهایی که موج از طریق همسایه پایینی به آنها رسیده است، مقدار متغیر Direction آنها برابر با Down می‌شود.

- Left : سلولهایی که موج از طریق همسایه سمت چپ به آنها رسیده است، مقدار متغیر Direction آنها برابر با Left می‌شود.

- Right : سلولهایی که موج از طریق همسایه سمت راست به آنها رسیده است، مقدار متغیر Direction آنها برابر با Right می‌شود.

- Null : سلولهایی که موج به آنها نرسیده است، مقدار متغیر آنها برابر با Null است.

### ۴-۳- وضعیت اولیه CA

در وضعیت اولیه می‌بایست طرح بر روی CA نگاشت شود. به این منظور سلولهایی که در آنها بلاک قرار دارد، دارای حالت B می‌شوند. سلولهای متناظر با مبدأ و مقصد بترتیب حالت S و T بخود می‌گیرند و حالت بقیه سلولها خواهد بود. مقدار متغیر Direction در وضعیت اولیه برای سلولهای F و T برابر Null و برای بقیه سلولها بی اهمیت خواهد بود.

### ۵-۳- وضعیت نهایی CA

در پایان الگوریتم سلولهایی که بر روی مسیر قرار دارند حالت P خواهند داشت. سلولهایی که در وضعیت اولیه دارای حالت‌های T یا S باشند تا وضعیت نهایی بدون تغییر باقی می‌مانند.

### ۶-۳- قوانین CA

• اگر سلولی دارای حالت S باشد، در گام بعد سلول بدون تغییر باقی می‌ماند.

• اگر سلولی دارای حالت B باشد، در گام بعد سلول بدون تغییر باقی می‌ماند.

• اگر سلول دارای حالت F باشد، وضعیت های زیر برای آن متصور است:

- هیچ یک از همسایگان سلول M یا S نباشد، در این صورت سلول بدون تغییر باقی می‌ماند.

- فقط یکی از همسایگان سلول حالت M یا S داشته باشد، در این صورت حالت سلول M شده و سلول به سمت همسایه مورد نظر نشانه می‌رود.

توضیح: اگر همسایه با حالت S یا M در یکی از جهات بالا بین چپ یا راست سلول قرار داشته باشد، متغیر Direction سلول بترتیب مقداری Right, Down, Up و Left را اختیار خواهد نمود.

• بیش از یک همسایه سلول دارای حالت M یا S باشد. در این صورت حالت سلول M شده و سلول به دلخواه یکی از همسایگان دارای حالت M یا S را انتخاب نموده و به آن نشانه می‌رود.

• اگر سلول دارای حالت T باشد، از قوانین سلول با حالت F پیروی کرده با این تفاوت که حالت سلول بدون تغییر باقی می‌ماند.

• اگر سلول دارای حالت M باشد، وضعیت های زیر برای آن متصور است:

- هیچ کدام از همسایگان سلول دارای حالت P یا T نباشد، در این صورت حالت سلول بدون تغییر باقی می‌ماند.

- یکی از همسایگان سلول دارای حالت P یا T باشد، در این صورت سلول تست می کند که آیا همسایه مذکور به سمت سلول نشانه رفته است یا نه. در صورت مثبت بودن جواب حالت سلول P شده و در غیر این صورت سلول حالت سلول بدون تغییر باقی می ماند.
- اگر سلول دارای حالت P باشد، در ۳ام بعد سلول بدون تغییر باقی می ماند.

### ۷-۳- پیچیدگی الگوریتم

مسیر از مبدأ تا مقصد در یک CA دو بعدی  $m \times m$  به هیچ طریقی نمی تواند طولی بیش از  $m^2$  داشته باشد، زیرا مسیر از هر سلول بیش از یکبار نمی تواند عبور کند و تعداد سلوهای بیش از  $m^2$  می باشد و لذا پیچیدگی الگوریتم از مرتبه  $O(m^2)$  می باشد.

### ۸-۳- اثبات الگوریتم

می خواهیم اثبات کنیم مسیری که توسط الگوریتم فوق ایجاد می شود کوتاهترین مسیر بین مبدأ و مقصد خواهد بود. برای این منظور ابتدا قضایا زیر را اثبات می نماییم. در این قضایا منظور از فاصله یک سلول تا سلول مبدأ، طول کوتاهترین مسیر از آن سلول تا سلول مبدأ می باشد.

**قضیه ۱:**  $P(t)$ : اولاً تمام سلوهایی که در وضعیت اولیه دارای حالت F هستند و فاصله آنها تا سلول مبدأ ۰ می باشد، تا ۳ام  $t-1$  دارای حالت F بوده و در ۳ام  $t$  حالت M را به خود می گیرند.

**ثانیاً:** تمام سلوهایی که در ۳ام  $t-1$  دارای حالت F بوده و در ۳ام  $t$  حالت M به خود می گیرند، نسبت به سلول مبدأ دارای فاصله ۱ می باشند.  
برهان: قضیه فوق را توسط استقرآ تام ثابت می کنیم:

پایه استقرار:  $P(1)$  صادق است. زیرا اولاً روشی است که سلوهایی که در وضعیت اولیه ( $t=0$ ) حالت F دارند و فاصله آنها تا سلول مبدأ برابر یک است همگی همسایه سلول مبدأ هستند و طبق قوانین CA در ۳ام یک حالت M بخود می گیرند. و ثانیاً سلوهایی که در ۳ام صفر (وضعیت اولیه) دارای حالت F هستند و در ۳ام یک حالت M بخود می گیرند، طبق قوانین CA لزوماً همسایه سلول مبدأ می باشند (در غیر اینصورت حالت M بخود نمی گرفتند) و فاصله همسایگان سلول مبدأ بیش از ۱ نیز برابریک می باشد.

فرض استقرار:  $P(1) \wedge P(2) \wedge P(3) \wedge \dots \wedge P(k)$

حکم استقرار:  $P(k+1)$

ابتدا قسمت اول حکم استقرار را ثابت می کنیم: سلوهایی که در وضعیت اولیه حالت F دارند و فاصله آنها تا سلول مبدأ  $k+1$  است، بطور قطعی تا ۳ام  $k$  حالت F دارند زیرا در غیر این صورت عددی مانند  $r$  ( $0 < r < k-1$ ) خواهیم داشت که آن سلول تا ۳ام  $k-r-1$  بوده و در ۳ام  $M$  شده است. در این صورت طبق قسمت ثانیاً گزاره  $P(k-r)$ ، فاصله آن سلول تا سلول مبدأ  $k-r$  می باشد. در حالی که فاصله سلول تا سلول مبدأ برابر  $k+1$  است. پس ثابت می شود که سلول تا ۳ام  $k$  حالت F دارد.

حال باید ثابت کنیم که در ۳ام  $k+1$  حالت سلول M خواهد شد. روشی است که سلولی که فاصله اش نسبت به سلول مبدأ  $k+1$  می باشد (مانند سلول A) حداقل یکی از همسایگانش دارای فاصله  $k$  نسبت به سلول مبدأ می باشد. طبق قسمت اولاً  $P(k)$  این همسایه در ۳ام  $k$  حالت M به خود گرفته و بنابراین چون سلول A در مرحله  $k+1$  دارای یک همسایه با حالت M می باشد طبق قوانین CA حالت M به خود می گیرد. پایان اثبات قسمت اولاً  $(P(k+1))$ .

حال می بایست قسمت ثانیاً از حکم استقرار ثابت شود:

فرض می کنیم سلول A در ۳ام  $k$ ، دارای حالت F بوده و در ۳ام  $k+1$  حالت M به خود گرفته است. این وضعیت تنها زمانی اتفاق می افتد که در ۳ام  $k$  یکی از همسایگان سلول A (مانند سلول B) دارای حالت M باشد. زیرا در غیر این صورت طبق قوانین CA حالت سلول A برابر M نمی شود.

می توان نشان داد که فاصله سلول A تا سلول مبدأ عددی بزرگتر از  $k$  است. زیرا در غیر این صورت عددی مانند  $r$  ( $0 < r < k-1$ ) وجود دارد که فاصله سلول A تا سلول مبدأ برابر با  $k-r$  است. در آن صورت طبق قسمت اولاً گزاره  $P(k-r)$ ، در ۳ام  $k-r$  حالت سلول A می شود. حالت سلول A تا ۳ام  $k$  نیز M باقی می ماند. در حالی که طبق فرض در ۳ام  $k$  دارای حالت F می باشد.

همچنین می توان نشان داد که فاصله سلول B تا سلول مبدأ عددی کوچکتر یا مساوی  $k$  می باشد. زیرا فرض کرده بودیم که در ۳ام  $k$  سلول B دارای حالت M است. در این صورت عددی مانند  $r$  ( $0 < r < k-1$ ) وجود دارد که سلول B تا ۳ام  $k-r-1$  دارای حالت F بوده و در ۳ام  $k-r$  حالت M به خود گرفته است. بنابراین طبق قسمت ثانیاً گزاره  $P(k-r)$  فاصله آن سلول تا سلول مبدأ عددی کوچکتر یا مساوی  $k$  است.

از طرف دیگر سلول A و سلول B همسایه می باشند و روشی است که اختلاف فاصله دو سلول همسایه تا سلول مبدأ برابر یک می باشد. از آنجا که فاصله سلول A تا سلول مبدأ عددی بزرگتر از  $k$  و فاصله سلول B تا سلول مبدأ عددی کوچکتر یا مساوی  $k$  است، تنها در صورتی سلول A و B می توانند با هم همسایه باشند که فاصله سلول B تا مبدأ برابر  $k$  و فاصله سلول A تا مبدأ برابر  $k+1$  باشد. که این همان اثبات قسمت ثانیاً حکم استقرارا می باشد.

**تعريف ۲:** رسیدن موج به یک سلول در زمان  $t$ : گوییم موج در زمان  $t$  به سلول A رسیده است اگر یکی از وضعیتها زیر اتفاق بیفتد:

- ۱- سلول A در زمان  $t-1$  دارای حالت F باشد و در زمان  $t$  حالت M بخود گیرد.

۲- سلول A در زمان  $t-1$  دارای حالت T بوده و مقدار متغیر Direction آن برابر Null باشد و در زمان  $t$  متغیر Direction آن مقداری غیر از Null داشته باشد.

قضیه ۲: اگر موج در گام  $t$  به سلول T برسد، مسیر ایجاد شده توسط فاز ردبایی مجدد دارای طول ۱ خواهد بود.

برهان: طبق قضیه ۱ تمام سلولهای T یا F که در فاصله  $d$  از سلول مبدأ قرار دارند، در گام  $t$  موج از طریق یکی از همسایگانشان به آنها می‌رسد و بنابراین سلول به سوی همسایه منتقل کننده موج نشانه می‌رود. روشن است همسایه‌ای که سلول به آن نشانه رفته در زمان  $t-1$ ، شده و بنابراین فاصله اش تا سلول مبدأ برابر  $-d$  است.

بر اساس استدلال فوق اگر موج در گام  $t$  به سلول T برسد، آنگاه سلول M که فاصله اش تا سلول مبدأ برابر  $-1$  است اشاره می‌کند و این سلول M در اولین گام از فاز ردبایی مجدد، حالت P به خود می‌گیرد. این سلول نیز به نوبه خود به سلولی که فاصله اش تا سلول مبدأ  $-2$  است اشاره می‌کند و در گام بعد این سلول نیز P می‌شود و این عمل تا رسیدن به سلول S ادامه می‌یابد که  $t$  گام طول خواهد کشید و در هر گام یک سلول M، p خواهد شد و بنابراین مسیر ایجاد شده دارای طول  $t$  خواهد بود. حال می‌خواهیم اثبات نماییم که مسیر ایجاد شده توسط الگوریتم کوتاهترین مسیر است. با استفاده از قضایای فوق این مطلب به سادگی قابل اثبات است. اگر کوتاهترین مسیر از سلول مبدأ تا سلول مقصد دارای طول  $d$  باشد طبق قضیه ۱ موج در زمان  $t$  به سلول مقصد می‌رسد و طبق قضیه ۲ اگر موج در زمان  $t$  به سلول مقصد برسد، مسیر ایجاد شده توسط فاز ردبایی مجدد دارای طول  $t$  است که همان کوتاهترین مسیر است.

### ۹-۳- ردیابی الگوریتم

در شکل ۳، چگونگی تغییر حالت سلولها در هنگام اجرای الگوریتم نمایش داده شده است. در این شکل از حروف S, T, P, M, T, S بترتیب برای نمایش سلولهای با حالت Target, Source, Path, Mark و Free می‌باشد. در این شکلها برای نمایش مقادیر Right, Left, Up, Down و Up متفاوت Direction سلولها بترتیب از نمادهای  $\rightarrow, \leftarrow, \downarrow, \uparrow$  استفاده شده است. در صورتی که مقدار متغیر Direction سلولی برابر Null باشد، هیچ کدام از این نمادها را نخواهد داشت.

### ۴- نتیجه گیری

در این مقاله یک الگوریتم مسیریاب مازمینتی بر اutomاتای سلولی (CA) دو بعدی ارائه شد. این الگوریتم مسیری با طول کمینه از سلول مبدأ به سلول مقصد را می‌یابد به طوری که مسیر از موانع عبور ننماید. الگوریتم پیشنهادی ساده بوده و تمام تراکنشهای آن محلی و منطبق با خواص حاکم بر اutomاتای سلولی می‌باشد. هر سلول اutomاتا نیز ساختار ساده ای داشته و در هر لحظه از زمان تنها به محتویات سلولهای همسایه دسترسی دارد. این الگوریتم مسیر مطلوب را در یک CA دو بعدی  $m \times m$  در  $O(m^2)$  گام زمانی می‌یابد.

### ۵- مراجع

- [1] W. Burks, Essays on Cellular Automata, Urbana, IL: University of Illinois Press, 1970.
- [2] J. H. Conway, E. Berlekamp, and R. Guy, Winning Ways for Your Mathematical Plays, Vol. 2, Academic Press, 1982.
- [3] K. Culik, L. Hurd, and S. Yu, "Computation Theoretic Aspects of Cellular Automata," Physica D, Vol. 45, pp. 357-378, 1990.
- [4] D. Farmer, T. Toffoli, and S. Wolfram, Cellular Automata Proceedings of an Interdisciplinary Workshop, Amsterdam, North Holland, 1984.
- [5] L. Gordillo, and V. Luna, "Parallel Sort on a Linear Array of Cellular Automata," IEEE Transaction on Computers, pp. 1904-1910, 1994.
- [6] A. H. Gutowsky, Cellular Automata, Cambridge, MA: MIT Press, 1990.
- [7] F. O. Hadlock, "A Shortest Path Algorithm for Grid Graph," Networks, 1977.
- [8] C. Y. Lee, "An Algorithm for Path Connections and its Applications," IRE Transactions on Electronic Computers, 1961.
- [9] M. Mitchel, "Computation in Cellular Automata: A Selected Review," Technical Report, Santa Fe Institute, Santa Fe, New Mexico, 1996.
- [10] N. Packard, "Two-Dimensional Cellular Automata," Journal of Statistical Physics, Vol. 30, pp. 901-942, 1985.
- [11] Y. Pan, Y.C. Hsu, and W. J. Kubitz, "A Path Selection Global Router," Proceedings of Design Automation Conference, 1987.
- [12] P. Sarkar, "Brief History of Cellular Automata," ACM Computing Surveys, Vol. 32, No. 1, 2000.
- [13] N. A. Sherwani, Algorithm for VLSI Physical Design Automation, Western Michigan University Kluwer Academic Publishers, 1993.



مجموعه مقالات  
هفتمین  
کنفرانس سالانه  
انجمن کامپیوتر ایران

۱۳۸۰ تا ۹ اسفند ماه

مرکز تحقیقات مخابرات ایران



انجمن کامپیوتر ایران  
Computer Society of Iran

