

حل مسئله بزرگترین برش در گراف با استفاده از اتماتای یادگیر سلوی

محمد رضا میدبی

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی امیرکبیر

تهران ایران

mmeybodi@aut.ac.ir

مهدى عنایت زارع

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی امیرکبیر

تهران ایران

enayatzare@aut.ac.ir

الگوریتم‌ها می‌توان به الگوریتم‌های سه‌نی^۳ و گونزالس^۴(SG,[۳])، هوف‌میستر^۵ و فمن^۶(الگوریتم ترکیبی)[۴] و ژئومنس^۷ و ویلیامسون^۸(GW,[۵]) اشاره کرد. نمونه‌هایی دیگر از این الگوریتم‌ها را می‌توان در [۶]، [۷] و [۸] مشاهده نمود. علاوه بر الگوریتم‌های تقریبی، الگوریتم‌های مکاشفه‌ای و تصادفی متعددی نیز برای مسئله بزرگترین برش گزارش شده است. در این الگوریتم‌ها تلاش بر این است که با استفاده از روش‌های مکاشفه‌ای، راه حلی سریع و یا دقیق‌تر برای این مسئله ارایه کنند. در این مقاله یک الگوریتم مبتنی بر اتماتای یادگیر [۹] و [۱۰] مشاهده کرد. در این مقاله یک الگوریتم مبتنی بر اتماتای یادگیر سلوی^۹ برای حل مسئله بزرگترین برش در گراف پیشنهاد می‌گردد. الگوریتم پیشنهادی با الگوریتم‌های تقریبی SG، GW، ترکیبی و نیز الگوریتم ژنتیک مقایسه شده است. طبق نتایج به دست آمده الگوریتم پیشنهادی نتایج بهتری را در مقایسه با الگوریتم‌های فوق الذکر تولید می‌کند.

ادامه مقاله به این صورت سازماندهی شده است. در بخش‌های ۲ تا ۴ به ترتیب به معرفی اجمالی اتماتای سلوی^{۱۰}، اتماتاهای یادگیر^{۱۱} و اتماتای یادگیر سلوی می‌پردازم. در بخش ۵ الگوریتم پیشنهادی شرح داده می‌شود و در بخش ۶ نتایج آزمایش‌ها ارایه می‌گردد. بخش نهایی مقاله نتیجه‌گیری می‌باشد.

۲- اتماتای سلوی

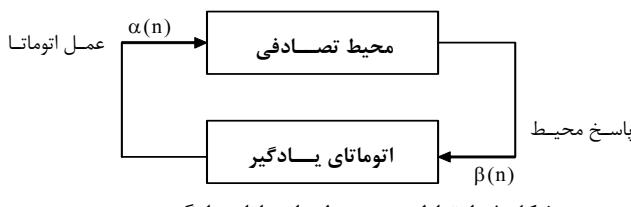
اتماتای سلوی (CA) [۱۲] [۱۳] در اواخر دهه ۱۹۴۰ توسط ون‌نیومن^{۱۲} مطرح و پس از او توسط ریاضی‌دانی بنام اولام^{۱۳} به عنوان مدلی برای بررسی رفتار سیستم‌های پیچیده پیشنهاد شد. اتماتای سلوی در حقیقت سیستم‌های دینامیکی گستته‌ای هستند که رفتارشان کاملاً بر اساس ارتباط محلی استوار است. در اتماتای سلوی، فضای بصورت یک شبکه تعريف می‌گردد که به هر خانه از آن یک سلول گفته می‌شود. زمان بصورت گسسته پیش می‌رود و قوانین آن بصورت سرتاسری است که از طریق آن در هر مرحله هر سلول، وضعیت جدید خود را با در نظر گرفتن همسایه‌های مجاور خود بدست می‌آورد. قوانین اتماتای سلوی، نحوه تأثیر پذیرفتن سلول از سلول‌های همسایه خود را مشخص می‌کند. یک سلول را همسایه سلول دیگر گوییم هرگاه بتواند

چکیده: مسئله بزرگترین برش در گراف دارای کاربردهای فراوانی از جمله طراحی مدارهای مجتمع متراکم و فیزیک آماری می‌باشد. بزرگترین برش در گراف عبارت است از افزار مجموعه رأس‌های گراف به دو زیرمجموعه غیرمشترک به گونه‌ای که تعداد (وزن) یال‌هایی که یکسر آنها در یک زیرمجموعه و سر دیگران در زیرمجموعه دیگر قرار گرفته است، بیشینه شود. مسئله بزرگترین برش یکی از مسائل NP-Complete می‌باشد و به همین دلیل الگوریتم‌های تقریبی متعددی برای حل آن ارایه شده است. در این مقاله یک الگوریتم مبتنی بر اتماتای یادگیر سلوی برای حل این مسئله پیشنهاد می‌گردد. الگوریتم پیشنهادی با الگوریتم‌های تقریبی سه‌نی، ژئومنس و ترکیبی و نیز الگوریتم ژنتیک مقایسه شده است. طبق نتایج به دست آمده الگوریتم پیشنهادی نتایج بهتری را در مقایسه با الگوریتم‌های فوق الذکر تولید می‌کند.

واژه‌های کلیدی: مسئله بزرگترین برش، الگوریتم‌های تقریبی، اتماتای یادگیر سلوی.

۱- مقدمه

بزرگترین برش^۱ گراف که دارای کاربردهای متعددی از جمله طراحی مدارهای مجتمع متراکم و فیزیک آماری می‌باشد[۱] عبارت است از افزار رأس‌های گراف به دو قسمت به گونه‌ای که تعداد (وزن) یال‌هایی که یکسر آنها در هر کدام از این قسمتها قرار گرفته است، بیشینه شود. بطور رسمی، در یک گراف ساده G با مجموعه رأس‌های V مجموعه‌ی یال‌های E و وزن‌های نامنفی W_{ij} روی یال‌های $\{i, j\} \in E$ ، مسئله بزرگترین برش عبارت است از یافتن مجموعه‌های S_1 و S_2 از رأس‌ها به قسمی که اولاً $V = S_1 \cup S_2$ ، ثانیاً $S_1 \cap S_2 = \emptyset$ و ثالثاً وزن برش (S_1, S_2) بیشینه شود. منظور از وزن برش (S_1, S_2) وزن یال‌هایی است که یک سرشناس در S_1 و سر دیگران در S_2 قرار دارد. برای سادگی معمولاً فرض می‌کنند برای هر $i, j \in E$. $W_{ij}=1$. مسئله بزرگترین برش یکی از مسائل NP-Complete می‌باشد[۲] و به همین دلیل الگوریتم با پیچیدگی چندجمله‌ای برای آن وجود ندارد. با توجه به این که در بسیاری از کاربردها یافتن یک راه حل نزدیک به بهینه نیز می‌تواند مفید باشد،



محیط را می‌توان توسط سه‌تایی $E = (\alpha, \beta, c)$ نشان داد که در آن $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودی‌ها، $\beta = \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه احتمال‌های جریمه مجموعه خروجی‌ها و $c = \{c_1, c_2, \dots, c_r\}$ مجموعه احتمال‌های جریمه می‌باشد. هرگاه β مجموعه دو عضوی باشد، محیط از نوع P می‌باشد. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می‌شود. در محیط از نوع Q و در محیط از طور گسته یک مقدار از مقادیر محدود در فاصله $[0, 1]$ احتمال این $\beta(n)$ متغیر تصادفی در فاصله $[0, 1]$ است. c ایستا است که عمل α_i نتیجه نامطلوب داشته باشد. در محیط ایستا مقادیر c بدون تغییر می‌مانند، حال آنکه در محیط غیرایستا این مقادیر در طی زمان تغییر می‌کنند.

اتوماتای یادگیر با ساختار ثابت توسط پنج‌تایی $(\alpha, \beta, F, G, \phi)$ نشان داده می‌شود که در آن $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عمل‌های اتماتا، $\beta = \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودی‌های اتماتا، $\phi = \{\phi_1, \phi_2, \dots, \phi_s\}$ مجموعه وضعیت‌های داخلی اتماتا، $F: \phi \times \beta \rightarrow \phi$ تابع تولید وضعیت جدید اتماتا و $G: \phi \rightarrow \alpha$ تابع خروجی می‌باشد که وضعیت کنونی اتماتا را به خروجی بعدی می‌نگارد.

اتوماتای یادگیر با ساختار متغیر را می‌توان توسط چهارتایی (α, β, p, T) نشان داد که $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عمل‌های اتماتا، $\beta = \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودی‌های اتماتا، $p = \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب هریک از عمل‌ها و $T[\alpha(n), \beta(n), p(n)] = p(n+1)$ الگوریتم یادگیری می‌باشد. الگوریتم زیریک نمونه از الگوریتم‌های یادگیری خطی است. فرض کنید عمل α_i در مرحله n ام انتخاب شود.

- پاسخ مطلوب

$$p_i(n+1) = p_i + a[1 - p_i(n)]$$

$$p_j(n+1) = (1-a)p_j(n) \quad \forall j \neq i$$

- پاسخ نامطلوب

$$p_i(n+1) = (1-b)p_i(n)$$

$$p_j(n+1) = (b/r - 1) + (1-b)p_j(n) \quad \forall j \neq i$$

آن را در یک مرحله و بر اساس قانون حاکم تحت تأثیر قرار دهد. در بدست آوردن وضعیت کنونی سلول علاوه بر وضعیت قبلی سلول‌های همسایه، می‌توان وضعیت قبلی خود سلول را نیز دخالت داد.

تعريف ۱: اتماتای سلولی d بعدی یک

چندتایی $CA = (Z^d, \phi, N, F)$ است به طوری که:

Z^d یک شبکه از d تایی‌های مرتب از اعداد صحیح

می‌باشد. این شبکه می‌تواند یک شبکه متناهی، نیمه متناهی یا نامتناهی باشد.

$\phi = \{1, \dots, m\}$ یک مجموعه متناهی از حالت‌ها می‌باشد.

$N = \{\bar{x}_1, \dots, \bar{x}_m\}$ یک زیر مجموعه متناهی

از Z^d می‌باشد که بردار همسایگی خوانده می‌شود. بردار

همسايگي، موقعیت نسبی همسایگان را برای هر سلول u

در شبکه سلولی به صورت زیر مشخص می‌کند:

$$N(u) = \{u + \bar{x}_i \mid i = 1, \dots, m\}$$

تابع $N(u)$ دو شرط زیر را ارضا می‌کند:

$$\forall u \in Z^d \Rightarrow u \in N(u)$$

$$\forall u, v \in Z^d \Rightarrow u \in N(v) \wedge v \in N(u)$$

$F: \phi^{\overline{m}} \rightarrow \phi$ قانون محلی CA می‌باشد

در مدل‌سازی سیستم‌های فیزیکی و بیولوژیکی، گاهی لازم است که قوانین را بصورت احتمالی در نظر بگیریم. رفتار احتمالی را می‌توان به عنوان نویز در سیستم تعبیر نمود. یکی از اشکالات اتماتای سلولی، تعیین فرم قطعی قوانین مورد نیاز برای یک کاربرد خاص است. راه حل‌های متفاوتی در برخورد با این مشکلات به نظر می‌رسد. یکی از این راه حل‌ها احتمالاتی کردن قوانین می‌باشد. به این ترتیب که تمام قانون‌های امکان‌پذیر را در نظر بگیریم و برای فعل شدن آنها احتمالی در نظر بگیریم. اما مشکلی که باز گریبان گیر ما خواهد شد آن است که شناسایی همین احتمال‌ها در سیستم‌های ناشناخته عملی نمی‌باشد. پس باید به سمتی حرکت کنیم که به نحوی خود ابزار با گذشت زمان بتواند قوانین مناسب را استخراج کند.

۳- اتماتای یادگیر

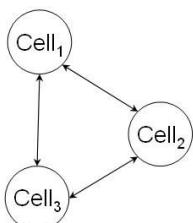
اتوماتای یادگیر [۱۴]، ماشینی است که می‌تواند تعدادی متناهی عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می‌شود و نتیجه ارزیابی در قالب پاسخی مثبت یا منفی به اتماتا داده می‌شود و اتماتا از این پاسخ در انتخاب عمل بعدی تأثیر می‌گیرد. هدف نهایی این است که اتماتا یاد بگیرد تا از بین اعمال خود بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حد اکثر برساند. کارکرد اتماتای یادگیر در تعامل با محیط، در شکل ۱ مشاهده می‌شود.

عمل بهنگام سازی ساختار اتوماتاهای موجود در اتوماتای یادگیر سلولی توسط الگوریتم یادگیری انجام می‌شود.

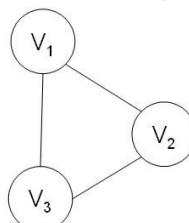
اتوماتای یادگیر سلولی را یکنواخت می‌گوییم، اگر برای تمام سلول‌ها، تابع همسایگی، قانون محلی و اتوماتاهای یادگیر یکسان باشند در غیر این صورت اتوماتای یادگیر سلولی غیریکنواخت نامیده می‌باشد. نوع دیگر اتوماتای یادگیر سلولی، اتوماتای یادگیر سلولی باز^{۱۴} (OCLA) می‌باشد[۱۸]. در OCLA علاوه بر محیط محلی^{۱۵} یک محیط سراسری^{۱۶} نیز برای آن در نظر گرفته شده است[۱۸][۱۹]. در OCLA دادن جریمه و یا پاداش به عمل انتخاب شده توسط یک سلول علاوه بر اعمال انتخابی توسط همسایگانش به پاسخ محیط سراسری نیز بستگی دارد. در[۱۹] اثبات شده است این مدل همانند CLA بسته[۱۷]، برای قوانین جابجایی پذیر، می‌تواند به نقاط بهینه محلی همگرا شود. در صورتی که اتصالات بین اتوماتاهای یادگیر منظم (مانند آرایه یک بعدی و یا آرایه دو بعدی) نباشد به آن اتوماتای یادگیر سلولی نامنظم^{۱۷}[۲۰] گفته می‌شود. اگر بروزرسانی سلول‌ها به صورت همگام^{۱۸} و در غیر این صورت نامهمگام^{۱۹}[۲۱][۲۲] نامیده می‌شود. برای بعضی از کاربردهای اتوماتای یادگیر سلولی می‌توان به [۱۶-۲۴] مراجعه نمود.

۵- الگوریتم پیشنهادی

در این بخش یک الگوریتم تقریبی مبتنی بر اتوماتای یادگیر سلولی برای حل مسئله بزرگترین برش در گراف پیشنهاد می‌گردد. برای این الگوریتم در ابتدا یک اتوماتای یادگیر سلولی غیریکنواخت همگام نامنظم متناظر با گراف مسئله ایجاد می‌شود. بطور مثال برای گراف شکل ۲-الف اتوماتای یادگیر سلولی نامنظم شکل ۲-ب ایجاد می‌شود. هر یک از اتوماتاهای یادگیر در اتوماتای یادگیر سلولی دارای دو عمل "عضو شدن در مجموعه S_1 " و "عضو شدن در مجموعه S_2 " می‌باشد. شماره هر سلول در اتوماتای یادگیر سلولی شماره رأس متناظر با آن سلول در گراف مسئله می‌باشد (از ۱ تا n). اتوماتای یادگیر در هر سلول از نوع L_{RP} با ضریب پاداش و جریمه ۱، ۰، ۰ می‌باشد. در ابتدا احتمال انتخاب اعمال هر یک از اتوماتاهای یادگیر مساوی و برابر $1/2$ در نظر گرفته می‌شود.



شکل ۲-ب- اتوماتای یادگیر سلولی متناظر با شکل ۲-الف



شکل ۲-الف- یک گراف بدون جهت بدون وزن

در روابط بالا، a پارامتر پاداش و b پارامتر جریمه می‌باشند. با توجه به مقادیر a و b سه حالت زیر را می‌توان در نظر گرفت. زمانی که a و b با هم برابر باشند، الگوریتم را L_{RP} می‌نامیم، زمانی که b از a خیلی کوچکتر باشد، الگوریتم را L_{REP} می‌نامیم. و زمانی که b مساوی صفر باشد الگوریتم را L_{RI} می‌نامیم[۱۵].

۴- اتوماتای یادگیر سلولی

بسیاری از مسایل را نمی‌توان با استفاده از یک اتوماتای یادگیر تنها حل کرد، بلکه قدرت اصلی اتوماتای یادگیر زمانی آشکار می‌شود که آنها به صورت دسته‌جمعی به کار روند. با توجه به این مسئله و ضعفهای عنوان شده برای اتوماتای سلولی، با ترکیب این دو مدل، مدل جدیدی با نام اتوماتای یادگیر سلولی ایجاد گردید[۱۶]. در زیر تعریف فرمال اتوماتای یادگیر سلولی ارائه شده است[۱۷].

تعریف ۲: اتوماتای یادگیر سلولی d بعدی یک

چندتایی $CLA = (Z^d, \phi, A, N, F)$ است به طوری که:

- Z^d یک شبکه از d تایی‌های مرتب از اعداد صحیح می‌باشد.

این شبکه می‌تواند یک شبکه متناهی، نیمه متناهی یا متناهی باشد.

- ϕ یک مجموعه متناهی از حالت‌ها می‌باشد.

- A ، یک مجموعه از اتوماتاهای یادگیر است که هر یک از آنها به یک سلول از اتوماتای سلولی نسبت داده می‌شود.

- $\{ \bar{x}_1, \dots, \bar{x}_{\bar{m}} \} = N$ یک زیرمجموعه متناهی از Z^d می‌باشد که بردار همسایگی نامیده می‌شود.

- $F: \underline{\phi}^{\bar{m}} \rightarrow \underline{\beta}$ قانون محلی اتوماتای یادگیر سلولی می‌باشد

به طوری که $\underline{\beta}$ مجموعه مقادیر است که می‌تواند به عنوان سیگنال تقویتی پذیرفته شود.

عملکرد اتوماتای یادگیر سلولی را می‌توان به شرح زیر بیان کرد. در هر لحظه هر اتوماتای یادگیر در اتوماتای 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 اتوماتای یادگیر سلولی پاداش داده و یا جریمه می‌شود. با توجه به اینکه عمل انتخاب شده پاداش گرفته و یا جریمه شده است، اتوماتا رفتار خود را تصحیح کرده و ساختار داخلی اتوماتا بهنگام می‌گردد. معمولاً عمل بروزرسانی تمام اتوماتها به صورت همزمان انجام می‌شود. بعد از بروزرسانی، هر اتوماتا در اتوماتای 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 می‌یابد.

✓ اگر درجه آن سلول از درجه تمام سلول‌های همسایه مذکورش بیشتر باشد آنگاه عمل انتخابی پاداش می‌گیرد و در غیر این صورت جریمه می‌شود.

- در غیر این صورت (وقتی که عمل سلول با عمل هیچ‌کدام از همسایگانش برابر نباشد و یا عمل سلول با عمل بعضی از همسایگانش برابر باشد و هیچ‌کدام از آن همسایه‌ها عمل خود را تغییر نداده باشند) آنگاه عمل انتخابی پاداش می‌گیرد.
- شیوه کد این الگوریتم در شکل ۳ نشان داده شده است.

۶- نتایج آزمایشات

گراف‌های مورد بررسی گراف‌های موجود در مجموعه استاندارد DIMACS [۲۵] با تعداد ۱۰، ۲۵، ۵۰، ۱۰۰، ۱۵۰ و ۲۰۰ گره می‌باشند که يال‌های آنها به صورت تصادفی و با احتمال ۵٪ انتخاب شده‌اند. نتایج حاصل از الگوریتم پیشنهادی با نتایج الگوریتم‌های سه‌نی و گونزالس (SG) [۳]، ژئومنس و ویلیامسون (GW) [۵]، الگوریتم ترکیبی (Combinatorial) [۴] و ژنتیک (Genetic) [۱۱] مقایسه شده است. معیارهای مقایسه اندازه برش به دست آمده و زمان اجرا بر حسب ثانیه می‌باشد. برای اجرای الگوریتم‌ها یک کامپیوتر شخصی با پردازنده اینتل P4 2.4GHz و با ۵۱۲ مگابایت حافظه اصلی مورد استفاده قرار گرفته است. هر یک از نتایج که در جدول ۱ و ۲ و نمودارهای ۴ و ۵ گزارش شده است، متوسط ۵۰ بار اجرا می‌باشد.

همان‌گونه که نمودارها نشان می‌دهند، الگوریتم پیشنهادی از نظر اندازه برش حاصل از تمام الگوریتم‌های مورد بحث بهتر عمل می‌کند و لی زمان اجرای آن در مقایسه با الگوریتم‌های ترکیبی، سه‌نی و ژنتیک بیشتر است. نکته قابل توجه در مورد الگوریتم پیشنهادی این است که با وجود این که این الگوریتم نتایج بهتری نسبت به الگوریتم ژئومنس از نظر اندازه برش تولید می‌کند، زمان اجرای آن نیز در مقایسه با این الگوریتم بهتر می‌باشد.

در اتوماتای یادگیر سلولی که برای حل این مسئله در نظر گرفته شده است دادن پاداش یا جریمه به عمل انتخابی توسط یک سلول، علاوه بر اعمال انتخاب شده توسط سلول و همسایگانش، به اعمال انتخاب شده در مرحله قبل توسط سلول و همسایگانش نیز بستگی دارد. در اولین مرحله از الگوریتم اتوماتای یادگیر هر سلول از بین اعمال "عضو شدن در مجموعه ۱" و یا "عضو شدن در مجموعه ۲" یکی را انتخاب می‌کند. عمل انتخابی یک سلول در این مرحله به عنوان عمل قبلی این سلول برای مرحله بعد محسوب می‌شود. در ادامه الگوریتم، در هر مرحله هر سلول از اتوماتای یادگیر سلولی یکی از اعمال خود را انتخاب می‌کند و سپس عمل انتخاب شده توسط سلول طبق قانونی که در ادامه توضیح داده می‌شود پاداش گرفته و یا جریمه می‌شود و این مراحل آنقدر ادامه می‌یابد تا در یک مرحله هیچ سلولی جریمه نشود. در پایان این مرحله اندازه برش حاصل به این صورت بدست می‌آید. ابتدا برای هر سلول تعداد سلول‌های همسایه‌ای که عمل انتخابی‌شان متفاوت با عمل آن سلول می‌باشد تعیین می‌گردد. نصف مجموع تعداد این گونه همسایه‌ها برای تمام سلول‌ها، اندازه برش می‌باشد.

هر سلول برای دادن پاداش و یا جریمه به صورت زیر عمل می‌کند
(قانون حاکم بر اتوماتای یادگیر سلولی):

▪ اگر عمل سلول با عمل قبلی آن متفاوت باشد آنگاه:

- اگر عمل سلول با اعمال قبلی بعضی از همسایگانش متفاوت باشد و عمل حداقل یکی از این همسایه‌ها با عمل قبلی اش یکسان باشد، آنگاه عمل انتخابی پاداش دریافت می‌کند.

▪ اگر عمل سلول با اعمال قبلی بعضی از همسایگانش متفاوت باشد ولی هیچ‌کدام از آن همسایه‌ها عمل خود را تغییر نداده باشند آنگاه:

- ✓ اگر درجه سلول از درجه تمام سلول‌های همسایه مذکورش بیشتر باشد، عمل انتخابی پاداش می‌گیرد و در غیر این صورت جریمه می‌شود.

▪ اگر عمل سلول با اعمال قبلی تمام همسایگانش یکسان باشد، آنگاه

- ✓ اگر تعداد همسایگانی از آن سلول که عمل خود را تغییر داده‌اند بیشتر از همسایگانی باشد که عمل خود را تغییر نداده‌اند، عمل انتخابی پاداش می‌گیرد و در غیر این صورت جریمه می‌شود.

▪ اگر عمل سلول همان عمل قبلی آن باشد، آنگاه:

- اگر عمل سلول با اعمال بعضی از همسایگانش برابر باشد و عمل حداقل یکی از این همسایه‌ها با عمل قبلی اش متفاوت باشد، آنگاه:

Algorithm CLA-Max-Cut

Input: Graph G(V,E)

Output: the size of the Max-Cut

Begin

Construct an irregular CLA for input graph

Repeat

For all cells do in parallel

Select an action

If (the cell's action is not equal to the cell's old action)

If (the cell's action is equal to the old action of some of its neighboring cells)

If (there exists at least one neighbor that has not change its action)

Reward the cell's action

Else

If (the cell's degree is greater than the degree of all its neighboring cells)

Reward the cell's action

Else

Penalize the cell's action

End if

End if

//the cell's action is not equal to the old action of any of its neighboring cell

If (the number of neighbors that have changed their actions is greater than those that have not changed their actions)

Reward the cell's action

Else

Penalize the cell's action

End if

End if

//the cell's action is equal to the cell's old action

If (the cell's action is equal to the action of some of neighboring cells)

If (there exists at least one neighbor that has changed its action)

If (the cell's degree is greater than the degree of all neighboring cells)

Reward the cell's action

Else

Penalize the cell's action

End if

//there does not exist any neighbor that has changed its action

Reward the cell's action

End if

//the cell's action is not equal to the actions of all neighboring cells

Reward the cell's action

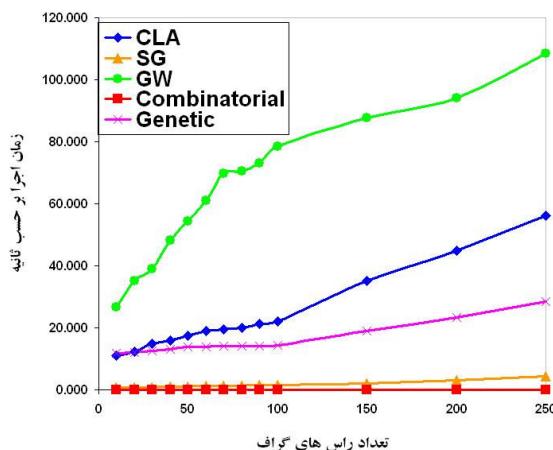
End if

End if

Until Reward the action of all cells

Return the size of the Max Cut

End



شکل ۵- مقایسه الگوریتم پیشنهادی با الگوریتم‌های سهندی، زئوننس، ترکیبی و ژنتیک بر اساس زمان اجرا بر حسب ثانیه

جدول ۱- اندازه برش به دست آمده برای الگوریتم‌های مختلف برای گراف‌هایی با تعداد راس‌های مختلف

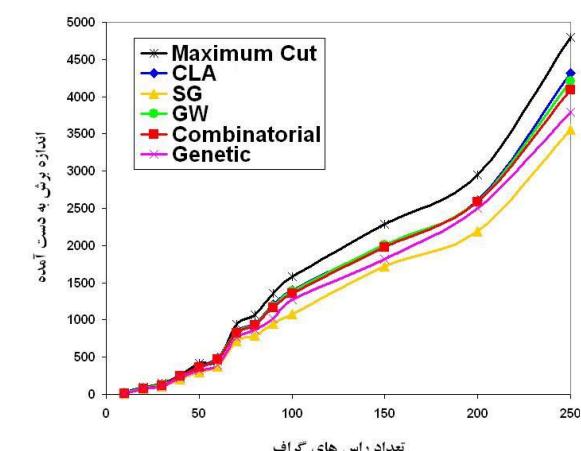
الگوریتم	اندازه برش	تعداد راس‌ها				
		CLA	ژنتیک	ترکیبی	GW	SG
۱۰	۳۳	۲۵	۱۹	۲۲	۲۸	۱۴
۲۵	۱۰۹	۹۷	۸۴	۸۵	۹۵	۷۷
۵۰	۴۷۰	۴۲۰	۳۴۴	۳۸۴	۴۱۱	۳۰۸
۱۰۰	۱۸۲۳	۱۶۰۴	۱۱۸۵	۱۳۹۰	۱۵۹۵	۱۰۰۸
۱۵۰	۲۴۴۹	۲۲۰۴	۱۹۰۲	۱۹۵۶	۲۱۴۲	۱۵۸۶
۲۰۰	۳۱۴۰	۲۸۶۶	۲۵۴۰	۲۶۹۰	۲۷۴۷	۱۹۹۷
۲۵۰	۵۰۲۳	۴۵۰۱	۳۹۹۲	۴۰۷۸	۴۳۹۵	۳۴۴۰

جدول ۲- زمان اجرا بر حسب ثانیه برای الگوریتم‌های مختلف برای گراف‌هایی با تعداد راس‌های مختلف

الگوریتم	تعداد راس‌ها	الگوریتم				
		CLA	ژنتیک	ترکیبی	GW	SG
۵۰	۱۰۲۳	۱۷.۴۲۳	۱۳.۸۰۵	۰.۰۱۱	۵۴.۲۴۴	۱.۰۲۳
۱۰۰	۱.۶۴۸	۲۲.۰۹۸	۱۴.۲۶۸	۰.۰۱۳	۷۸.۲۶۲	۱.۶۴۸
۱۵۰	۲.۱۳۴	۳۵.۱۸۴	۱۸.۹۰۸	۰.۰۲۸	۸۷.۷۶۷	۲.۱۳۴
۲۰۰	۳.۰۹۸	۴۴.۸۲۵	۲۳.۰۲۵	۰.۰۵۵	۹۴.۰۲۳	۳.۰۹۸
۲۵۰	۴.۲۴۴	۵۶.۰۹۴	۲۸.۰۴۵	۰.۰۸۹	۱۰۸.۵۸۷	۴.۲۴۴

۸- مراجع

- [1] F. Barahona, M. Grotschel, M. Junger and G. Reinelt, "An Application of Combinatorial Optimization to Statistical Physics and Circuit Layout Design", Oper. Res., Vol. 36, pp. 493-513, 1988.
- [2] R. Karp, "Reducibility among combinatorial problems", Complexity of computer computations, pp. 85-104, 1972.
- [3] S. Sahni and T. Gonzalez, "P-Complete Approximation Problems", Journal of ACM, Vol. 23, No. 3, pp. 555-565, 1976.
- [4] T. Hofmeister and H. Lefmann, "A Combinatorial Design Approach to MAXCUT," Proceedings of the 13th Symposium on Theoretical Aspects of Computer Science, pp. 441-452, 1996.
- [5] M. X. Goemans and D. P. Williamson, "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming", Journal of ACM, Vol. 42, No. 6, pp. 1115-1145, 1995.
- [6] P. M. Vitanyi, "How Well Can a Graph be n-Colored?", Disc. Math. Vol. 34, pp. 69-80, 1981.
- [7] S. Poljak and D. Turžík, "A Polynomial Algorithm for Constructing a Large Bipartite Subgraph With an Application to a Satisfiability Problem", Can. J. Math, Vol. 34, pp. 519-524, 1982.
- [8] D. J. Haglin and S. M. Venkatesan, "Approximation and Intractability Results for the Maximum Cut Problem and its Variants", IEEE Trans. Comput., Vol. 40, pp. 110-113, 1991.



شکل ۴- مقایسه الگوریتم پیشنهادی با الگوریتم‌های سهندی، زئوننس، ترکیبی و ژنتیک بر اساس اندازه برش به دست آمده

- [18] H. Beigy and M. R. Meybodi, "Open Synchronous Cellular Learning Automata", Advances in Complex Systems, 2007, to appear.
- [19] H. Beigy and M. R. Meybodi, "Open Synchronous Cellular Learning Automata", Proceedings of the 8th world Multi-conference on Systemic, Cybernetics and Informatics(SCI2004), pp. 9-15, Orlando, Florida, USA, 2004.
- [20] 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.
- [21] H. Beigy and M. R. Meybodi "Asynchronous Cellular Learning Automata", Proceedings of 10th Annual CSI Computer Conference Iran, Telecommunication Research Center, Tehran, Iran, pp. 271-280, 2005.
- [22] H. Beigy and M. R. Meybodi, "Asynchronous Cellular Learning Automata", Automatica, Journal of International Federation of Automatic Control, 2007, to appear.
- [23] 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.
- [24] 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, 2004.
- [25] <http://dimacs.rutgers.edu/Challenges/Seventh/Instances>
- [9] P. Festa, P. M. Pardalos, M. G. C. Resende and C. C. Ribeiro, "Randomized Heuristics for the MAX-CUT Problem", Optimization Methods and Software, Vol. 17, No. 6, pp. 1033-1058, 2002
- [10] O. Dolezal, T. Hofmeister and H. Lefmann, "A Comparison of Approximation Algorithms for the Maxcut-Problem", Manuscript, Universitat Dortmund, Lehrstuhl Informatik 2, Dortmund, Germany, 1999.
- [11] A. Khurti, Th. Bäck and J. Heitkötter, "An Evolutionary Approach to Combinatorial Optimization Problems", Proceeding of 22nd Annual ACM Computer Science Conference, Ed. D. Sizmar, pp. 66-73, ACM Press, New York, 1994.
- [12] S. Wolfram, "Cellular Automata", Los Alamos Science, Vol. 9, pp. 2-21, fall 1983.
- [13] S. Wolfram, "Universality and complexity in cellular automata", Physica D, No. 10, pp. 1-35, Jan. 1984.
- [14] K. S. Narendra and M.A.L. Thathacha, "Learning Automata: An Introduction", Prentice Hall, 1989.
- [15] M. A. L. Thathachar and P. S. Sastry, "Varieties of Learning Automata: An Overview", IEEE Transaction on Systems, Man and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, pp. 711-722, 2002.
- [16] M. R. Meybodi, H. Beigy and 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.
- [17] H. Beigy and M. R. Meybodi, "A Mathematical Framework for Cellular Learning Automata", Advanced in Complex Systems, Vol. 7, No. 3&4, pp. 294-319, 2004.

¹ Maximum Cut

² Approximation

³ Sahni

⁴ Gonzalez

⁵ Hofmeister

⁶ Lefmann

⁷ Geomans

⁸ Williamson

⁹ Cellular Learning Automata (CLA)

¹⁰ Cellular Automata (CA)

¹¹ Learning Automata (LA)

¹² Von Neumann

¹³ Ulam

¹⁴ Open Cellular Learning Automata

¹⁵ Local Environment

¹⁶ Global Environment

¹⁷ Irregular Cellular Learning Automata

¹⁸ Synchronous Cellular Learning Automata

¹⁹ Asynchronous Cellular Learning Automata