حل مساله بزرگترین برش در گراف با استفاده از اتوماتای یادگیر سلولی

مهدی عنایت زارع محمدرضا میبدی

آزمایشگاه محاسبات نرم
دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیرکبیر
تهران ایران

چکیده: مساله بزرگترین برش در گراف دارای کاربردهای فراوانی از جمله طراحی مدارهای مجتمع متراکم و فیزیک آماری می‌باشد. بزرگترین برش در گراف عبارت است از افراز مجموعه رأس‌های گراف به دو زیرمجموعه غیرمشترک به‌گونه‌ای که تعداد (وزن) یال‌هایی که یک‌سر آنها در یک زیر مجموعه و سر دیگرشان در زیر مجموعه دیگر قرار گرفته است, بیشینه شود. مساله بزرگترین برش یکی از مسایل NP-Complete می‌باشد و به‌همین دلیل الگوریتم‌های تقریبی متعددی برای حل آن ارایه شده است. در این مقاله یک الگوریتم مبتنی بر اتوماتای یادگیر سلولی برای حل این مساله پیشنهاد می‌گردد. الگوریتم پیشنهادی با الگوریتم‌های تقریبی سهنی, ژئومنس و نیز الگوریتم‌های ترکیبی و ژنتیک مقایسه شده است. طبق نتایج به‌دست آمده الگوریتم پیشنهادی نتایج بهتری را در مقایسه با الگوریتم‌های فوق‌الذکر تولید می‌کند.

کلمات کلیدی: مساله بزرگترین برش، الگوریتم‌های تقریبی، اتوماتای یادگیر سلولی

Solving Maximum Cut Problem Using Cellular Learning Automata

M. Enayatzare M. R. Meybodi

Computer Engineering and Information Technology Department
Amirkabir University of Technology
Tehran Iran

Abstract: The Maximum cut problem has applications in many fields including VLSI circuit design and statistical physics. Maximum cut is to partitioning the vertex set of graph into two disjoint part so that the number (or weight) of edges joining vertices in different parts as large as possible. Maximum cut is known to be NP-Complete and therefore many approximation algorithms are reported for solving this problem. In this paper, we proposed an algorithm for solving this problem using cellular learning automata is proposed. In order to study the performance of the proposed algorithm computer simulations have been conducted and the results are compared with the results obtained for Sehni, Geomens, Combinatorial and Genetic algorithms. The results show the superiority of the proposed algorithm over the existing algorithms.

Keywords: Maximum Cut Problem, Approximate Algorithms, Cellular Learning Automata

۱- مقدمه

بزرگترین برش[1] گراف که دارای کاربردهای متعددی از جمله طراحی مدارهای مجتمع متراکم و فیزیک آماری می‌باشد[۲] عبارت است از افراز رأس‌های گراف به دو قسمت به‌گونه‌ای 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 مجموعه‌ی رأس‌های V, مجموعه‌ی یال‌های E و وزن‌های نامنفی W_{ij} روی یال‌های $\{i, j\} \in E$, مساله بزرگترین برش عبارت است از یافتن مجموعه‌های S_1 و S_2 از راس‌ها به‌قسمی‌که اولاً $V = S_1 \cup S_2$, ثانیاً $S_1 \cap S_2 = \phi$ و ثالثاً وزن برش

[1] Maximum Cut

(S_1, S_2) بیشینه شود. منظور از وزن برش (S_1, S_2) وزن یال‌هایی است که یک سرشان در S_1 و سر دیگرشان در S_2 قرار دارد. برای سادگی معمولاً فرض می‌کنند برای هر $\{i, j\} \in E$,W_{ij}=1. مساله بزرگترین برش یکی از مسایلNP-Complete می‌باشد[۱] و به‌همین دلیل الگوریتم با پیچیدگی چندجمله‌ای برای آن وجود ندارد. با توجه به این‌که در بسیاری از کاربردها یافتن یک راه حل نزدیک به بهینه نیز می‌تواند مفید باشد, الگوریتم‌های تقریبی[۱] متعددی برای حل آن ارایه شده است. از جمله این الگوریتم‌ها می‌توان به الگوریتم‌های سهنی[۲][۳], ویتانی[۳][۴], پولژاک[۴][۵], هاگلین[۵][۶], هوف‌میستر[۶][۷] و ژئومنس[۷][۸] اشاره کرد. علاوه بر الگوریتم‌های تقریبی, الگوریتم‌های مکاشفه‌ای و تصادفی متعددی نیز برای مسأله بزرگترین برش ارایه شده است. در این الگوریتم‌ها تلاش بر این است که با استفاده از روش‌های مکاشفه‌ای[۸], راه‌حلی سریع و یا دقیق‌تر برای این مساله ارایه کنند. نمونه‌ای از این الگوریتم‌ها را می‌توان در [۹], [۱۰], [۱۱] و [۱۲] مشاهده کرد. در این مقاله یک الگوریتم مبتنی بر اتوماتای یادگیر سلولی[۹] برای حل مساله بزرگترین برش در گراف پیشنهاد می‌گردد. الگوریتم پیشنهادی با الگوریتم‌های تقریبی سهنی, ژئومنس و نیز الگوریتم‌های ترکیبی و ژنتیک مقایسه شده است. طبق نتایج به‌دست آمده الگوریتم پیشنهادی نتایج بهتری را در مقایسه با الگوریتم‌های فوق‌الذکر تولید می‌کند.

ادامه مقاله به این صورت سازمان‌دهی شده است. در بخش‌های ۲ تا ۴ به ترتیب به معرفی اجمالی اتوماتای سلولی[۱۰], اتوماتاهای یادگیر[۱۱] و اتوماتای یادگیر سلولی می‌پردازیم. در بخش ۵ الگوریتم پیشنهادی شرح داده می‌شود و در بخش ۶ نتایج آزمایش‌ها ارایه می‌گردد. بخش نهایی مقاله نتیجه‌گیری می‌باشد.

۲- اتوماتای سلولی

اتوماتای سلولی (CA) [۱۳] [۱۴] 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. اتوماتای سلولی در حقیقت سیستم‌های دینامیکی گسسته‌ای هستند که رفتارشان کاملاً بر اساس ارتباط محلی استوار است. در اتوماتای سلولی، فضا بصورت یک شبکه تعریف می‌گردد که به هر خانه 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 اتوماتای سلولی، نحوه تأثیر پذیرفتن سلول از سلول‌های همسایه خود را مشخص می‌کند. یک سلول را همسایه سلول دیگر گوئیم هرگاه بتواند آن را در یک مرحله و براساس 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.

تعریف ۱ اتوماتای سلولی d بعدی یک چندتایی $CA = (Z^d, \phi, N, F)$ است به‌طوری‌که:

- Z^d یک شبکه از d تایی‌های مرتب از اعداد صحیح می‌باشد. این شبکه می‌تواند یک شبکه متناهی، نیمه متناهی یا نامتناهی باشد.
- $\phi = \{1,...,m\}$ یک مجموعه متناهی از حالت‌ها می‌باشد.
- $N = \{\overline{x}_1,...,\overline{x}_{\overline{m}}\}$ ، $\overline{x}_i \in Z^d$ ، یک زیر مجموعه متناهی از Z^d می‌باشد که بردار همسایگی خوانده می‌شود. بردار همسایگی، موقعیت نسبی همسایگان را برای هر سلول u در شبکه سلولی به صورت زیر مشخص می‌کند:

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

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

---

[1] Approximation
[2] Sehni
[3] Vitanyi
[4] Poljak
[5] Hoglin
[6] Hofmeister
[7] Geomens
[8] Heuristic
[9] Cellular Learning Automata (CLA)
[10] Cellular Automata (CA)
[11] Learning Automata (LA)
[12] Von Neumann
[13] Ulam

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

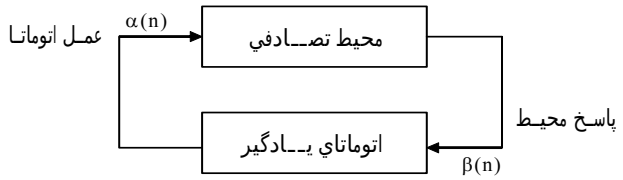
$\forall u,v \in Z^d \Rightarrow u \in N(v) \wedge v \in N(u)$

- $F : \underline{\phi}^{\overline{m}} \rightarrow \underline{\phi}$ 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 اتوماتای سلولی، تعیین فرم قطعی قوانین مورد نیاز برای یک کاربرد خاص است. راه‌حل‌های متفاوتی در برخورد با این مشکلات به نظر می‌رسد. یکی از این راه‌حل‌ها احتمالاتی کردن قوانین می‌باشد. به این ترتیب که تمام قانون‌های امکان‌پذیر را در نظر بگیریم و برای فعال شدن آنها احتمالی در نظر بگیریم. اما مشکلی که باز گریبان‌گیر 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- اتوماتای یادگیر

اتوماتای یادگیر[۱۵]، 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 اتوماتا داده می‌شود و اتوماتا از این پاسخ در انتخاب عمل بعدی تاثیر می‌گیرد. هدف نهایی این است که اتوماتا یاد بگیرد تا از بین اعمال خود بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداکثر برساند. کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل ۱ مشاهده می‌شود.



شکل ۱- ارتباط بین محیط و اتوماتای یادگیر

محیط را می‌توان توسط سه‌تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه ورودی‌ها، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه خروجی‌ها و $c \equiv \{c_1, c_2, ..., c_r\}$ 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، $\beta(n)$ می‌تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله [۰،۱] و در محیط از نوع S، $\beta(n)$ متغیر تصادفی در فاصله [۰،۱] است. c_i احتمال این است که عمل α_i نتیجه نامطلوب[1] داشته باشد. در محیط ایستا[2] مقادیر c_i بدون تغییر می‌مانند، حال آنکه در محیط غیرایستا[3] 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 $\alpha \equiv \{\alpha_1, \alpha_2, \cdots, \alpha_r\}$ مجموعه عمل‌های اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \cdots, \beta_m\}$ مجموعه ورودی‌های اتوماتا، $\phi \equiv \{\phi_1, \phi_2, \cdots, \phi_s\}$ مجموعه وضعیت‌های داخلی اتوماتا، $F : \phi \times \beta \rightarrow \phi$ تابع تولید وضعیت جدید اتوماتا و $G : \phi \rightarrow \alpha$ تابع خروجی می‌باشد که وضعیت کنونی اتوماتا را به خروجی بعدی می‌نگارد.

اتوماتای یادگیر با ساختار متغیر را می‌توان توسط چهارتایی $\{\alpha, \beta, p, T\}$ نشان داد که $\alpha = \{\alpha_1, ..., \alpha_r\}$ مجموعه عمل‌های اتوماتا، $\beta = \{\beta_1, ..., \beta_m\}$ مجموعه ورودی‌های اتوماتا، $p = \{p_1, .., p_r\}$ بردار احتمال انتخاب هریک از عمل‌ها و

[1] Unfavorable
[2] Stationary
[3] Non-Stationary

$p(n+1)=T[\alpha(n),\beta(n),p(n)]$ الگوریتم یادگیری می‌باشد. الگوریتم زیریک نمونه از الگوریتم‌های یادگیری خطی است. فرض کنید عمل α_i در مرحله nام انتخاب شود.

- پاسخ مطلوب

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

- پاسخ نامطلوب

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

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. زمانی‌که a و b با هم برابر باشند، الگوریتم را L_{RP} می نامیم، زمانی‌که b از a خیلی کوچک‌تر باشد، الگوریتم را $L_{R\varepsilon P}$ می‌نامیم. و زمانی‌که b مساوی صفر باشد الگوریتم را L_{RI} می‌نامیم[۱۶].

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 بکار روند. با توجه به این مساله و ضعف‌های عنوان شده برای اتوماتای سلولی، با ترکیب این دو مدل، مدل جدیدی با نام اتوماتای یادگیر سلولی ایجاد گردید[۱۷]. در زیر تعریف فرمال اتوماتای یادگیر سلولی ارائه شده است[۱۸].

تعریف ۲ (اتوماتای یادگیر سلولی): اتوماتای یادگیر سلولی d بعدی یک چندتایی $CLA=(Z^d,\phi,A,N,F)$ است به طوری‌که:

- Z^d یک شبکه از dتایی‌های مرتب از اعداد صحیح می‌باشد. این شبکه می‌تواند یک شبکه متناهی، نیمه متناهی یا نامتناهی 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.
- $N=\{\overline{x}_1,...,\overline{x}_{\overline{m}}\}$ یک زیر مجموعه متناهی از Z^d می‌باشد که بردار همسایگی نامیده می‌شود.
- $F:\underline{\phi}^{\overline{m}}\rightarrow\underline{\beta}$ قانون محلی اتوماتای یادگیر سلولی می‌باشد به طوری‌که $\underline{\beta}$ مجموعه مقادیری 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 اتوماتاها 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. عمل بهنگام‌سازی ساختار اتوماتاهای موجود در اتوماتای یادگیر سلولی توسط الگوریتم یادگیری انجام می‌شود.

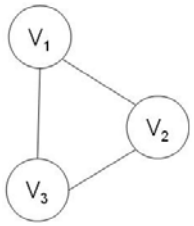
اتوماتای یادگیر سلولی را یکنواخت می‌گوییم، اگر برای تمام سلول‌ها، تابع همسایگی، قانون محلی و اتوماتاهای یادگیر یکسان باشند در غیر این صورت اتوماتای یادگیر سلولی غیریکنواخت نامیده می‌باشد. نوع دیگر اتوماتای یادگیر سلولی، اتوماتای یادگیر سلولی باز (OCLA) می‌باشد[۲۴]. در OCLA علاوه بر محیط محلی[1] یک محیط سراسری[2] نیز برای آن در نظر گرفته شده است[۲۰] [۲۴]. 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[۲۰] اثبات شده است این مدل همانند **CLA** بسته[۱۷] [۱۸]، برای قوانین جابجایی‌پذیر، می‌تواند به نقاط بهینه محلی همگرا شود. در

[1] Local Environment
[2] Global Environment

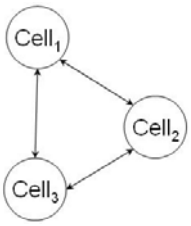
صورتی‌که اتصالات بین اتوماتاهای یادگیر منظم (مانند آرایه یک بعدی و یا آرایه دو بعدی) نباشد به آن اتوماتای یادگیر سلولی نامنظم[1][۲۳] گفته می‌شود. اگر بروزرسانی سلول‌ها به صورت همگام برای تمامی سلول‌ها صورت بگیرد، اتوماتای یادگیر سلولی همگام[2] و در غیر این صورت ناهمگام[3][۱۹] [۲۵] نامیده می‌شود. برای بعضی از کاربردهای اتوماتای یادگیر سلولی می‌توان به [۲۵-۱۷] مراجعه نمود.

۵- الگوریتم پیشنهادی

در این بخش یک الگوریتم تقریبی مبتنی بر اتوماتای یادگیر سلولی برای حل مساله بزرگترین برش در گراف پیشنهاد می‌گردد. برای این الگوریتم در ابتدا یک اتوماتای یادگیر سلولی غیریکنواخت همگام نامنظم متناظر با گراف مساله ایجاد می‌شود. بطور مثال برای گراف شکل۲-الف اتوماتای یادگیرسلولی نامنظم شکل ۲-ب ایجاد می‌شود. هر یک از اتوماتهای یادگیر در اتوماتای یادگیر سلولی دارای دو عمل "عضو شدن در مجموعه S_1" و "عضو شدن در مجموعه S_2" می‌باشند. شماره هر سلول در اتوماتای یادگیر سلولی شماره راس متناظر با آن سلول در گراف مساله می‌باشد (از ۱ تا n). اتوماتای یادگیر در هر سلول از نوع L_{RP} با ضریب پاداش و جریمه ۰,۰۱ می‌باشد. در ابتدا احتمال انتخاب اعمال هر یک از اتوماتاهای یادگیر مساوی و برابر ۱/۲ در نظر گرفته می‌شود.



شکل ۲- الف- یک گراف بدون‌جهت بدون وزن



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

در اتوماتای یادگیر سلولی که برای حل این مساله 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 S_1" و یا "عضو شدن در مجموعه S_2" 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 به‌صورت 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 قبلی‌اش یکسان باشد, انگاه 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 این‌صورت جریمه می‌شود.
  - اگر عمل سلول با اعمال قبلی تمام همسایگانش یکسان باشد, آنگاه

[1] Irregular Cellular Learning Automata
[2] Synchronous Cellular Learning Automata
[3] Asynchronous Cellular Learning Automata

✓ 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 این‌صورت 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 قبلی‌اش متفاوت باشد, آنگاه:

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

- در غیر این‌صورت (وقتی‌که 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.

شبه کد این الگوریتم در شکل ۳ نشان داده شده است.

```
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
                Else //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
            Else //cell's action is equal to the cell's old action
                If (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
                    Else //There does not exist any neighbor that has changed its action
                        Reward the cell's action
                    End if
                Else //cell's action is not equal to the actions of all neighboring cells
                    Reward the cell's action
                End if
```

```
                End if
            Until Reward action of all cells
            Return the size of the Max Cut
End
```

شکل ۳- الگوریتم پیشنهادی

۶-نتایج آزمایش‌ها

گراف‌های مورد بررسی گراف‌های موجود در مجموعه استاندارد DIMACS [۲۶] با تعداد ۱۰, ۲۵, ۵۰, ۱۰۰, ۱۵۰ و ۲۰۰ گره می‌باشند که یال‌های آنها به‌صورت تصادفی و با احتمال ۰,۵ انتخاب شده‌اند. نتایج حاصل از الگوریتم پیشنهادی با نتایج الگوریتم‌های سهنی[۳], ژئومنس[۸], الگوریتم ترکیبی[۱۱] و ژنتیک[۱۲] 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.

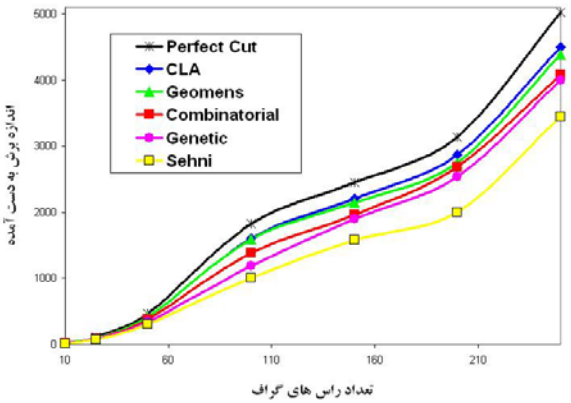
همان‌گونه که نمودارها نشان می‌دهند, الگوریتم پیشنهادی از نظر اندازه برش حاصل از تمام الگوریتم‌های مورد بحث بهتر عمل می‌کند ولی زمان اجرای در مقایسه با الگوریتم‌های ترکیبی, سهنی و ژنتیک بیشتر است. نکته قابل توجه در مورد الگوریتم پیشنهادی این است که با وجود این‌که این الگوریتم نتایج بهتری نسبت به الگوریتم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

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

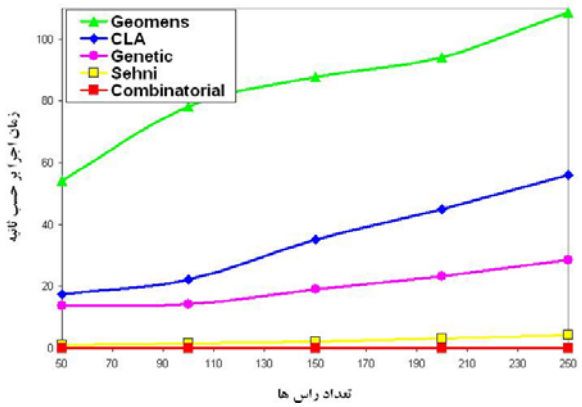
الگوریتم / تعداد راس‌ها	اندازه برش	سهنی	ژئومنس	ترکیبی	ژنتیک	CLA
۱۰	۳۳	۱۴	۲۸	۲۲	۱۹	۲۵
۲۵	۱۰۹	۷۷	۹۵	۸۵	۸۴	۹۷
۵۰	۴۷۰	۳۰۸	۴۱۱	۳۸۴	۳۴۴	۴۲۰
۱۰۰	۱۸۲۳	۱۰۰۸	۱۵۹۵	۱۳۹۰	۱۱۸۵	۱۶۰۴
۱۵۰	۲۴۴۹	۱۵۸۶	۲۱۴۲	۱۹۵۶	۱۹۰۲	۲۲۰۴
۲۰۰	۳۱۴۰	۱۹۹۷	۲۷۴۷	۲۶۹۰	۲۵۴۰	۲۸۶۶
۲۵۰	۵۰۲۳	۳۴۴۰	۴۳۹۵	۴۰۷۸	۳۹۹۲	۴۵۰۱

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

الگوریتم / تعداد راس‌ها	سهنی	ژئومنس	ترکیبی	ژنتیک	CLA
۵۰	۱,۰۲۳	۵۴,۲۴۴	۰,۰۱۱	۱۳,۸۰۵	۱۷,۴۲۳
۱۰۰	۱,۶۴۸	۷۸,۳۶۲	۰,۰۱۳	۱۴,۲۶۸	۲۲,۰۹۸
۱۵۰	۲,۱۳۴	۸۷,۷۶۷	۰,۰۲۸	۱۸,۹۰۸	۳۵,۱۸۴
۲۰۰	۳,۰۹۸	۹۴,۰۲۳	۰,۰۵۵	۲۳,۲۵	۴۴,۸۲۵
۲۵۰	۴,۲۴۴	۱۰۸,۵۸۷	۰,۰۸۹	۲۸,۴۵	۵۶,۰۹۴



شکل ۴- مقایسه الگوریتم پیشنهادی با الگوریتم‌های سهنی، ژئومنس، ترکیبی و ژنتیک بر اساس اندازه برش به‌دست آمده



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

۷- نتیجه‌گیری

در این مقاله با استفاده از اتوماتای یادگیر سلولی نامنظم یک الگوریتم برای حل مساله بزرگترین برش گراف پیشنهاد گردید. الگوریتم پیشنهادی با الگوریتم‌های تقریبی سهنی و ژئومنس و نیز الگوریتم‌های ترکیبی و ژنتیک مقایسه شده است. طبق نتایج به‌دست آمده الگوریتم پیشنهادی نتایج بهتری را نسبت به الگوریتم‌های فوق‌الذکر تولید می‌کند.

مراجع

[1] R. Karp, "Reducibility among combinatorial problems", Complexity of computer computations, pp. 85-104, 1972.

[2] 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.

[3] S. Sahni and T. Gonzalez, “P-Complete Approximation Problems”, Journal of ACM, Vol. 23, No. 3, pp. 555-565, 1976.

[4] P. M. Vitanyi, “How Well Can a Graph is n-Colored?”, Disc. Math. Vol. 34, pp. 69-80, 1981.

[5] S. Poljak and D. Turźik, “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.

[6] 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.

[7] T. Hofmeister and H. Lefmann, “A Combinatorial Design Approach to MAXCUT,” In Proceedings of the 13th Symposium on Theoretical Aspects of Computer Science, pp. 441-452, 1996.

[8] M. X. Goemans and D. P. Williamson, “Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming”, Journal of Association for Computing Machinery, Vol. 42, No. 6, pp. 1115-1145, 1995.

[9] P. Festa, P. M. Pardalos, M. G. C. Resende and C. C. Ribero, “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] T. Hofmeister and H. Lefmann, “A Combinatorial Design Approach to Max Cut”, Random Structures & Algorithms, Vol. 9, pp. 163-175, 1996.

[12] 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.

[13] S. Wolfram, "Cellular Automata", Los Alamos Science, Vol. 9, pp. 2-21, fall 1983.

[14] S. Wolfram, "Universality and complexity in cellular automata", Physica D, No. 10, pp. 1-35, Jan. 1984.

[15] K. S. Narendra and M.A.L. Thathacha, "Learning Automata: An Introduction", Prentice Hall, 1989.

[16] 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.

[17] 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.

[18] 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.

[19] H. Beigy and M. R. Meybodi “Asynchronous Cellular Learning Automata”, In Proceedings of 10th Annual CSI Computer Conference Iran, Telecommunication Research Center, Tehran, Iran, pp. 271-280, Feb. 2005.
[20] H. Beigy and M. R. Meybodi, “Open Synchronous Cellular Learning Automata”, In Proceedings of the 8th world Multi-conference on Systemic, Cybernetics and Informatics(SCI2004), pp. 9-15, Orlando, Florida, USA, July 2004.
[21] 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.
[22] 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.
[23] 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.
[24] H. Beigy and M. R. Meybodi, "Open Synchronous Cellular Learning Automata" Advances in Complex Systems, 2007, to appear.
[25] H. Beigy and M. R. Meybodi, "Asynchronous Cellular Learning Automata" Automatica, Journal of International Federation of Automatic Control, 2007, to appear.
[26] http://dimacs.rutgers.edu/Challenges/Seventh/Instances