



یافتن کلیک وزن دار بیشینه با استفاده از اتاماتای یادگیر سلوالی

فرناد بزرگی^۱، علیرضا رضوانیان^۲ و محمدرضا میدی^۳

F_bozorgi@qiau.ac.ir

۱ دانشجوی دکتری، آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ir.a.rezvaninan@aut.ac.ir

۲ استاد تمام، آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، mmeybodi@aut.ac.ir

می شود، که به هر راس آن یک وزن اختصاص می یابد. این وزن می تواند فاصله، نوع ارتباطات، ترافیک یا هر چیز دیگری تعریف شود. به طوری که $E = \{e_1, \dots, e_k\} \subseteq V \times V$ مجموعه رئوس، $\{v_1, \dots, v_n\}$ مجموعه یال های گراف و W وزن مربوط به راس در مجموعه اعداد حقیقی است. کلیکی با حداقل وزن از میان تمام کلیک های موجود را کلیک با وزن بیشینه گویند که آن را می توان به صورت کلیک کاندید^۷ در نظر گرفت و نشان داد، که به ازای هر γ داریم $W(\gamma) \geq W(\gamma^*)$. به طوری که در حالتی که وزن ها بر روی رئوس باشند: $W(\gamma) = \sum_{i \in \gamma} w_i$ که به کلیکی با حداقل وزن رئوس^۸ اطلاق می شود [۷، ۸].

الگوریتم مبتنی بر کلونی مورچه (روشی که برای یافتن کلیک بیشینه از یک کلیک خالی شروع و در هر مرحله راسی را به کلیک اولیه می افزاید) را پیشنهاد کردند [۹]. همچنین [۱۰] (الگوریتم جستجوی محلی و واکنشی دار)^۹ بر پایه جستجوی منوعه که بر اصل جریمه

دهی استوار است، معروف به RLS را ارائه کردند.

در این مقاله، الگوریتمی مبتنی بر اتاماتای یادگیر سلوالی نامنظم برای حل مسئله کلیک وزن دار بیشنهادی پیشنهاد می گردد. الگوریتم پیشنهادی با الگوریتم های ارائه شده در [۱۱] مقایسه می گردد. نتایج آزمایش ها نشان می دهد که الگوریتم پیشنهادی نتایج بهتری را تولید می کند. ادامه مقاله به این صورت سازمان دهی شده است: در بخش ۲ به معرفی اجمالی اتاماتای یادگیر سلوالی و اتاماتای یادگیر سلوالی نامنظم می پردازیم. در بخش ۳ الگوریتم پیشنهادی مبتنی بر اتاماتای یادگیر سلوالی نامنظم شرح داده می شود و در بخش ۴ الگوریتم ارائه شده با الگوریتم های [۱۱] مقایسه می گردد. بخش نهایی مقاله نتیجه گیری است.

چکیده- در این مقاله، الگوریتمی مبتنی بر اتاماتای یادگیر سلوالی نامنظم برای حل مسئله کلیک وزن دار بیشنهادی پیشنهاد می گردد. در الگوریتم پیشنهادی، برای ساخت اتوماتون های یادگیر معادل با گراف وزن دار و بدون جهت G ، به ازای هر رأس معادل با آن یک اتوماتون یادگیر سلوالی در نظر گرفته می شود، که هر سلوال آن مجدهز به یک اتوماتون یادگیر است و در ابتدا احتمال انتخاب اقدام هر یک از اتوماتون های یادگیر به طور مساوی مقداردهی می شود. در مراحل بعدی بردار احتمال با توجه به الگوریتم یادگیری بروزرسانی می شود. در این مسئله فرض بر این است که وزن ها بر روی رئوس قرار می گیرند. بر اساس نتایج مقایسات انجام شده، الگوریتم پیشنهادی نتایج قابل قبولی را ارائه می دهد.

کلمات کلیدی- اتاماتای یادگیر سلوالی، اتاماتی یادگیر سلوالی نامنظم، اقدام، گراف وزن دار

۱- مقدمه

مسئله کلیک به عنوان یکی از مسائل بنیادی در نظریه گراف و این مسئله جزء مسائل NP- کامل است که در کاربردهای مختلف مهندسی مورد استفاده قرار گرفته است و برای حل آن الگوریتم های متعددی توسط محققین ارائه شده است [۱]. یکی از زیرگراف های مهم در گراف، کلیک بیشینه است که به عنوان یکی از مسائل کلاسیک در نظریه گراف شناخته می شود. کلیک بیشینه امروزه در طیف وسیعی از کاربردها مانند شبکه های موردنی متحرک^۱ [۲]، توصیه فروش^۲ [۳]، بیوانفورماتیک^۳ [۴]، شناسایی الگو^۴ [۵] و آنالیز شبکه های اجتماعی^۵ [۶] مورد استفاده قرار می گیرد و راهکارهای متعددی جهت حل و بهینه کردن آن ارائه شده است.

به طور کلی در دنیای واقعی مسائل زیادی با گراف وزن دار مطرح می شود، یک گراف وزن دار به صورت (V, E, W) نشان داده

¹ Mobile Adhoc Network

² Shopping Recommendation

³ Bioinformatics

⁴ Pattern Recognition

⁵ Social Networks Analysis

⁶ Maximum Vertex Weighted Clique(MVWC) problem

⁷ Reactive



۲- اتاماتای یادگیر سلولی

اتوماتای یادگیر سلولی^۸، یک مدل ریاضی برای سیستم‌هایی با اجزای ساده است، بطوری که رفتار هر جزء براساس رفتار همسایگانش و نیز تجربیات گذشته‌اش تعیین و اصلاح می‌شود. اجزای ساده تشکیل دهنده این مدل، از طریق کنش و واکنش با یکدیگر رفتار پیچیده‌ای از خود نشان می‌دهند، بنابراین از آن می‌توان در مدل‌سازی بسیاری از مسائل بهره برد. این مدل در [۱۲] بصورت ریاضی مورد تحلیل قرار گرفته و رفتار همگرایی آن مطالعه شده است. هر اتاماتای یادگیر سلولی، از یک اتاماتای سلولی تشکیل شده است که هر سلول آن به یک یا چند اتاماتای یادگیر مجهز است که حالت این سلول را مشخص می‌سازد. قانون محلی در محیط حاکم است و این قانون تعیین می‌کند که آیا عمل انتخاب شده توسط یک اتاماتا در سلول باید پاداش داده شود و یا اینکه جریمه شود. دادن پاداش و یا جریمه باعث به هنگام رسانی ساختار اتاماتای یادگیر سلولی بهمنظور نیل به یک هدف مشخص می‌گردد. ایده اصلی اتاماتای یادگیر سلولی، که زیر مجموعه‌ای از اتاماتای یادگیر سلولی تصادفی محاسبه می‌شود، استفاده از اتاماتای یادگیر برای محاسبه احتمال انتقال حالت در اتاماتای سلولی تصادفی است.

۱-۲ اتاماتای یادگیر سلولی نامنظم

اتاماتای یادگیر سلولی نامنظم [۱۳]، یک اتاماتای یادگیر سلولی است با حذف این محدودیت که اتاماتای یادگیر سلولی در این نوع، دارای ساختار شبکه‌ای نیست. این نوع اتاماتای یادگیر سلولی، برای کاربردهایی مثل شبکه‌های حسگر پیسمیم، سیستم‌های شبکه‌ای آزاد و کاربردهای مبتنی بر گراف که بر اساس یک شبکه‌ی مستطیلی نمی‌توانند مدل شوند، بسیار مؤثر است. یک اتاماتای یادگیر سلولی نامنظم، به صورت یک گراف بدون جهت تعریف می‌شود، بطوری که هر رأس آن منتظر با یک سلول است که دارای یک اتاماتای یادگیر است. اتاماتای یادگیر که در هر سلول وجود دارد، حالت (عمل) آن سلول را بر اساس بردار احتمال عمل مربوط به آن سلول، تعیین می‌کند. شبیه اتاماتای یادگیر سلولی، قانونی وجود دارد که اتاماتای یادگیر سلولی نامنظم، تحت آن عمل می‌کند و قانون اتاماتای یادگیر سلولی و اعمال انتخاب شده بوسیله‌ی اتاماتاهای یادگیر همسایه برای یک اتاماتای یادگیر خاص، بردار احتمال اعمال مربوط به آن اتاماتای یادگیر را بروز می‌کند.

۳- الگوریتم پیشنهادی

۱-۳ ساخت اتوماتون‌های یادگیر سلولی

جهت نگاشتن یک گراف به اتوماتای یادگیر سلولی، از اتوماتای یادگیر سلولی نامنظم با ساختار متغیر استفاده می‌کنیم و برای ساخت اتوماتون‌های یادگیر معادل با گراف وزن‌دار و بدون جهت G، به ازای هر رأس معادل با آن یک اتوماتون یادگیر سلولی در نظر گرفته می‌شود، که هر سلول آن مجهز به یک اتوماتون یادگیر است. اتوماتای یادگیر در هر سلول از نوع RP با ضریب پاداش و جریمه ۰ است. تعداد اقدام هر یک از اتوماتون‌های یادگیر در اتوماتای یادگیر سلولی^۹ برابر درجه نوD در گراف است و هرکدام از اقدام‌ها که به صورت $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$ نشان داده می‌شود، برای اتاماتای یادگیر برابر با انتخاب یکی از رئوس در نظر گرفته می‌شوند. سیگنال تقویتی β تابعی متناظر با پیدا کردن کلیک وزن‌دار پیشینه را بررسی می‌کند. وضعیت CLA هم به صورت یک رشته از اعداد مشخص شده که مشخص کننده کلیک است و در تمام CLA‌ها همه اتاماتاهای به طور همزمان فعالیت می‌کنند. در ابتدا احتمال انتخاب اقدام هر یک از اتوماتون‌های یادگیر مساوی و برابر یک تقسیم بر تعداد گره‌های همسایه است، به این صورت که احتمال اقدام‌های اتوماتون‌های یادگیر بر روی یال‌ها قرار می‌گیرد. در مراحل بعدی بردار احتمال با توجه به الگوریتم یادگیری بروزرسانی می‌شود. در ابتدا می‌باشد برای هر اتوماتون یادگیر مقادیر اولیه‌ای به هر احتمال اقدام تخصیص داد، حال با در نظر گرفتن گراف G = (V, E, W) بردار احتمال به صورت (p_i) ^{۱۰} نشان داده می‌شود که n برابر شماره تکرار الگوریتم است، بنابراین p_i برابر بردار احتمال اولیه راس نام است. مقدار بردار احتمال اولیه در جاهایی که بین رأس آن و زام یالی وجود دارد برابر D_i / D و در غیر اینصورت برابر صفر است، به طوری که D_i برابر درجه راس آن است.

۲-۳ پیدا کردن کلیک توسط اتوماتون‌های یادگیر

پس از ساخته شدن اتوماتون‌های یادگیر در این مرحله نحوه جستجوی کلیک با وزن پیشینه ارائه می‌شود. هر سلول در اتوماتای یادگیر سلولی با سلول دیگر همسایه است اگر و تنها اگر بین راس متناظر با آن سلول در گراف مساله با راس متناظر با سلول دیگر، یالی وجود داشته باشد و یا به عبارت دیگر مجاور باشد (قانون همسایگی). در طول اجرای الگوریتم هر اتوماتای یادگیر به طور همزمان و متناسب با بردار احتمال انتخاب عمل خود، یکی از اقدام‌هایش (یکی از همسایه‌ها) را انتخاب می‌نماید و این انتخاب را در رشته خود ذخیره می‌کند، اگر وزن رشته ایجاد شده بزرگترین وزن را در میان تمام رشته‌های هم طول تولید شده توسط سایر LA‌ها داشت، این اقدام در رشته ایجاد شده توسط LA باقی می‌ماند

⁸ Cellular Learning Automata (CLA)



a از a خیلی کوچکتر باشد، الگوریتم^{۱۰} L_{REP} و اگر b صفر باشد، الگوریتم^{۱۱} L_{RI} نامیده می‌شود.

۴-۳ شرط توقف

شرط توقف میانگین‌گیری بردار احتمالات است، به این صورت که اگر در یک تعداد گام مشخص، مثلاً ۲۰ گام، میانگین احتمال‌های الگوریتم بهبودی نداشت، حلقه خارجی این الگوریتم متوقف می‌شود.

میانگین احتمالات از رابطه زیر محاسبه می‌شود:

$$\text{Mean Prob} = \frac{\sum_{i=1}^{n_c} P_{\text{clique}(i)}}{n_c} \quad (3)$$

در معادله (۳)، $P_{\text{clique}(i)}$: جمع احتمال‌های اقدام‌های شرکت کننده در تولید کلیک CLA نام و n_c : تعداد CLA ‌ها است.

در پایان هر بار جستجوی کلیک و اجرای یک دور از الگوریتم، کلیک بدست آمده با بهترین کلیک در تمامی تکرارهای الگوریتم تا مرحله جاری از لحاظ وزنی با توجه به حالت وزنی مبتنی بر رأس بر اساس رابطه (۴) مقایسه شده است.

$$W(\gamma) = \sum_{i \in \gamma} w_i \quad (4)$$

به طوری که در معادله (۴)، $(\gamma) W$ مجموع وزنی کلیک کاندید γ و w_i وزن رأس i است.

الگوریتم پیشنهادی شامل ۴ مرحله است:

۱. ساخت اتوماتون‌های یادگیر سلولی
۲. پیدا کردن کلیک توسط هر کدام از اتوماتون‌های یادگیر با توجه به بردار احتمال اقدام‌های آنها
۳. بروزرسانی بردار احتمالات (پاداش و جرمیه)
۴. بازگشت به مرحله ۲ تا رسیدن به شرط توقف

۴-نتایج شبیه‌سازی

پس از تفسیر الگوریتم پیشنهادی جهت محاسبه کلیک با وزن پیشینه، حال برای ارزیابی الگوریتم پیشنهادی از ۱۲ مجموعه داده مختلف پایگاه داده استاندارد DIMACS (از گراف‌های استاندارد مرکز ریاضیات کسیته و علوم کامپیوتر^{۱۲}) استفاده شده است. با توجه به این نکته که این دادگان غیر وزن‌دار هستند جهت ارزیابی می‌بایست آنها را به گراف‌های

و در غیر این صورت از رشته تولید شده حذف می‌شود. این کار تا زمانی ادامه پیدا می‌کند که دیگر انتخابی برای LA‌ها باقی نماند، یا رشته‌ها از یک دیگر تغییری نداشته باشند و یا اینکه تعداد حداقل حلقه داخلی فرا برسد. لازم به ذکر است اگر یک LA اقدامی را انتخاب می‌کند و آن را در رشته خود قرار می‌دهد اقدام‌هایی که با گره متناظر با این اقدام همسایه نیستند از لیست اقدام‌های مجاز این LA حذف می‌شوند (احتمال آنها در بردار احتمال برابر صفر می‌شود). این موضوع باعث می‌شود که رشته انتخابی توسط هر LA همواره کلیک باقی بماند.

۳-۳ بروزرسانی بردار احتمالات

در این مرحله بر اساس قوانین تعریف شده برای CLA پاداش یا جرمیه تعیین شده و اتوماتای یادگیر بردار احتمال انتخاب عمل خود را متناسبًا بروز می‌نماید. پس از تکرار الگوریتم هر کدام از LA‌ها یک رشته تولید کرده‌اند که این رشته نماینده کلیک است اگر وزن هر رشته از بزرگترین وزن تولید شده توسط همان LA در تکرارهای قبل بزرگتر باشد کلیه اقدام‌هایی که باعث شده‌اند این مقدار وزنی حاصل شود پاداش می‌گیرند و در غیر این صورت اگر وزن رشته کوچکتر از وزن قبلی باشد، کلیه اقدام‌هایی که باعث شده‌اند این مقدار وزنی حاصل شود جرمیه می‌شوند. این روند تا زمانی که شرط توقف فرا برسد ادامه پیدا می‌کند. نحوه گرفتن پاداش و جرمیه به صورت زیر در معادله‌های ۱ و ۲ توضیح داده شده است.

پاسخ مطلوب از محیط:

$$\begin{aligned} P_i(n+1) &= P_i(n) + a[1 - P_i(n)] \\ P_j(n+1) &= (1 - a)P_j(n) \quad \forall j, j \neq i \end{aligned} \quad (1)$$

پاسخ نامطلوب از محیط:

$$\begin{aligned} P_i(n+1) &= (1 - b)P_i(n) \\ P_j(n+1) &= \frac{b}{r-1} + (1 - b)P_j(n) \quad \forall j, j \neq i \end{aligned} \quad (2)$$

که در روابط فوق، متغیر r تعداد اکشن‌های اتاماتای یادگیر، متغیر a به عنوان پارامتر پاداش و متغیر b به عنوان پارامتر جرمیه در نظر گرفته می‌شود. با توجه به مقادیر a و b سه حالت را می‌توان در نظر گرفت. اگر مقادیر a و b برابر باشند، اتوماتای یادگیر^۹ L_{RP} ، چنانچه

10 Linear Reward Epsilon Penalty

11 Linear Reward Inaction

12 Center for Discrete Mathematics and Theoretical Computer Science

9 Linear Reward Penalty



Brock200-4	2107	2107	2107	2107
DSJC500.5	1725	1725	1725	1725
Hamming8_4	1472	1472	1472	1472

همان طور که در جدول (۱) مشاهده می شود الگوریتم پیشنهادی برای دیتاست های C250.9, C500.9, MANN_a27 و P_hat700-2 دارای حداکثر وزن بالاتری نسبت به الگوریتم های SSDALW و MSDLAW است.

در ارزیابی دوم جهت نشان دادن توانایی الگوریتم پیشنهادی، متوسط وزن بدست آمده برای 3° تکرار مجزا، انحراف معیار برای بهترین وزن های کلیک ها و همچنین زمان اجرای الگوریتم در اجراهای مجزا محاسبه شده است در جدول (۲)، میانگین و انحراف معیار به ترتیب به صورت $\mu \pm \sigma$ گزارش شده است. در این جدول در ستون اول نام دادگان استاندارد، در ستون دوم برای الگوریتم پیشنهادی به ترتیب زمان اجرا بر حسب ثانیه و میانگین وزن کلیک بدست آمده ذکر شده است، در ستون سوم و چهارم نیز به همین ترتیب زمان اجرا و میانگین وزن کلیک برای الگوریتم های SSDALW و MSDLAW آورده شده است.

جدول ۲: مقایسه متوسط وزن، انحراف معیار و متوسط زمان برای الگوریتم های MSDLAW, SSDALW, MCPCLAW

	MCPCLAW		SSDALW		MSDLAW	
Graph	Time $\mu \pm \sigma$	Weight $\mu \pm \sigma$	Time $\mu \pm \sigma$	Weight $\mu \pm \sigma$	Time $\mu \pm \sigma$	Weight $\mu \pm \sigma$
Brock200-2	8±2	1428±0	16±6	1418. 1±16	10±3	1428±0
C125.9	15±7	2522±7	25±11	2504.2±13	19±8	2521.9±8
C250.9	10±4	5035. 3±79	17±8	4850.2±99	12±4	5023.2±44
C500.9	44±27	6590±58	58±35	6228.0±124	47±29	6422.5±66
Keller4	4±0	1152.9±0.49	10±.1	1146.5±18	7±0	1152.8±0.5
MANN_a27	64±32	12166±2	87±41	12127. 4±4	71±35	12134.4±5
P_hat300-1	19±6	1057±0	23±12	1056.8±0.7	20±7	1057±0
P_hat300-2	21±9	2481.7±16	25±13	2304. 7±94	22±10	2441.2±38
P_hat700-2	74±40	4990±78	92±56	4637.5±159	80±44	4830.9±99

وزن دار تغییر دهیم. این تغییرات با استی برابر گراف های وزن دار با وزن معادل با رئوس صورت گیرد. جهت این تبدیل راهکاری توسط پولان [۱۴] ارائه شده است که در این نوشتار به طور مشابه از این روش بهره گرفته شده است. این راهکار به صورتی است که برای گراف های وزن دار با وزنی بر روی رئوس از رابطه $i \mod 200 + 1$ جهت ایجاد وزن برای هر راس V_i استفاده شده است. برای الگوریتم پیشنهادی رویکردهای مختلف یادگیری L_{RP} , L_{REP} , L_{RI} و L_{RRI} مورد آزمایش قرار گرفته است با توجه به بررسی های انجام شده، برای الگوریتم پیشنهادی رویکرد یادگیری L_{RP} از نتایج بهتری برخوردار بود به همین دلیل برای مقداردهی به پارامترهای الگوریتم پیشنهادی از معیار L_{RP} با مقادیر پاداش و جریمه به میزان ۰.۱ استفاده شده است. مقادیر پاداش و جریمه بر اساس آزمون و خط، این مقدار در نظر گرفته شده است. الگوریتم پیشنهادی به تعداد تکراری برابر ۱۰۰۰ و در ۳۰ اجرای مجزا صورت پذیرفته است. در الگوریتم پیشنهادی جهت ارزیابی نتایج در حالتی که وزن ها بر روی رئوس قرار دارند، الگوریتم پیشنهادی با [۱۱] مقایسه شده است.

در اولین آزمایش ابتدا ارزیابی ها بر اساس وزن موجود بر روی رئوس انجام شده است به طوری که محاسبه بهترین وزن از لحظه اندازه، متوسط وزن و انحراف معیار وزن های بدست آمده به ازای تکرارهای مجزا نشان داده شده است. جدول (۱) وزن بیشینه بدست آمده به وزن بیشینه واقعی و میانگین وزن کلیک بدست آمده ذکر شده است، در ستون سوم و چهارم نیز به همین ترتیب زمان اجرا و میانگین وزن کلیک برای الگوریتم های MSDLAW, SSDALW نشان می دهد همانطور که مشاهده می شود، الگوریتم پیشنهادی در مقایسه با الگوریتم های MSDLAW, SSDALW دارای نتایج مطلوب تری است.

جدول ۱: مقایسه نتایج حداکثر وزن الگوریتم پیشنهادی با MCPCLAW, SSDALW, MSDLAW

	Global Weight	MCPCLAW	SSDALW	MSDLAW
Graph	Best Weight	Max Weight	Max Weight	Max Weight
Brock200-2	1428	1428	1428	1428
C125.9	2529	2529	2529	2529
C250.9	5092	5072	5060	5060
C500.9	6955	6755	6493	6579
Keller4	1153	1153	1153	1153
MANN_a27	12281	12180	12139	12145
P_hat300-1	1057	1057	1057	1057
P_hat300-2	2487	2487	2487	2487
P_hat700-2	5290	5026	5020	5012



[7] Y. Xu-Hua, J. Feng-Ling, C. Sheng-Yong, and W. Wan-Liang, "Modeling evolution of weighted clique networks," Communications in Theoretical Physics, vol. 56, p. 952, 2011.

[8] X. Dolgui, A. Kovalyov, "Combinatorial design of a minimum cost transfer line," Journal Omega, Elsevier, Vol. 40, pp.31-41, January 2012.

[9] C. Solnon, S. Fenet, "A study of Aco capabilities for solving the maximum clique problem," Journal of Heuristics, Vol. 12, pp.155-180, May 2006.

[10] R. Battiti, M. Protasi, "Reactive local search for the maximum clique problem," Journal Algorithmica, Springer, Vol. 29, pp.610-637, December 2001.

[11] محمد سلیمانی پوری "حل تقریبی مسئله ماکزیمال در شبکه‌های پیچیده با استفاده از اتماتای یادگیر" پایان‌نامه، دانشگاه برق، رایانه، فناوری اطلاعات، گروه کامپیوتر، بهار ۱۳۹۲.

[12] H. Beigy, M.R. Meybodi, "Cellular learning automata with multiple learning automata in each cell and its applications," IEEE Transactions on, Vol.40, pp.54-65, Feb 2010.

[13] M. Asnaashari, MR. Meybodi, "Irregular Cellular Learning Automata and Its Application to Clustering in Sensor Networks," Proceedings of 15th Conference on Electrical Engineering (15th ICEE), Vol. on Communication, 2007.

[14] W. Pullan, "Approximating the maximum vertex/edge weighted clique using local search," Journal of Heuristics, vol. 14, pp. 117-134, 2008.

Brock200-4	12 ± 8	2105 ± 13	20 ± 13	2034.8 ± 49	13 ± 9	2101.9 ± 8
DSJC500-5	28 ± 11	1703.8 ± 20	35 ± 17	1654.6 ± 35	30 ± 12	1697.4 ± 24
Hamming 8_4	9 ± 0	1472 ± 0	12 ± 6	1447.4 ± 18	10 ± 0.1	1472 ± 0

همان‌طور که از نتایج جدول (۲) پیداست، الگوریتم پیشنهادی کاملاً نتایج قابل رقابتی را ارائه می‌دهد، پایین بودن مقدار انحراف معیار حاکی از نزدیکی وزن‌های محاسبه شده است که بیانگر همگرایی الگوریتم به جواب‌های بهتر است و از نظر میانگین وزنی بسیار بهتر از الگوریتم‌های ارائه شده در [۱۱] می‌باشد.

۵-نتیجه گیری

در این مقاله، مسئله کلیک بیشینه مورد بررسی قرار گرفته شده است و الگوریتمی برای یافتن کلیک بیشینه، مبتنی بر اتماتای یادگیر سلولی ارائه گردیده است. در این روش به هر گره از گراف یک اتماتای یادگیر اخلاص‌داده شده که نرخ پاداش و جریمه برابر ۰.۱ است. در نهایت جهت مقایسه کارایی، روش ارائه شده را بر روی استاندارد DIMACS ارزیابی کرده و نتایج الگوریتم پیشنهادی نشان دهنده برتری این الگوریتم نسبت به [۱۱] است.

مراجع

- [1] M. Garey, D. Johnson, "Computers and intractability: a guide to the theory of NP-completeness," Journal WH Freeman & Co., San Francisco, Vol. 6, pp. 145-159, March 1985.
- [2] Q. Chen , Z. M, Fadlullah, X. Lin & N. Kato, " A clique-based secure admission control scheme for mobile ad hoc networks (MANETs)," Journal of Network and Computer Applications, Vol. 34, pp.1827-1835, November 2011.
- [3] Q. Yang, P. Zhou, H. Zhang & J. Zhang, "Clique Discovery Based on User Similarity for Online Shopping Recommendation," Information Technology Journal, Vol. 10, pp.1587-1593, 2011.
- [4] D. Fukagawa, T. Tamura, A. Takasu, E. Tomita, & T. Akutsu, "A clique-based method for the edit distance between unordered trees and its application to analysis of glycan structures," BMC bioinformatics, Vol. 12, pp.1-13, 2011.
- [5] S. Mimaroglu, M. Yagci, "Cliques for combining multiple clusterings," Expert Systems with Applications, Vol. 39, pp.1889-1901, February 2012.
- [6] J. Pattillo, N. Youssef, & S. Butenko, "Clique Relaxation Models in Social Network Analysis," in Handbook of Optimization in Complex Networks, Vol. 58, pp. 143-162, 2012.