شماره: ....
تاریخ: ....
پیوست: ....

Intelligent Systems Scientific Society of Iran
انجمن سیستم های فازی ایران
دانشگاه صنعتی مالک اشتر

دومین کنگره مشترک سیستم های فازی و هوشمند ایران
۷-۹ آبان ماه ۱۳۸۷
2nd Joint Congress on Fuzzy and Intelligent Systems
28-30 October 2008

حل تقریبی مساله دسته ماکزیمال (Maximal Clique) با استفاده از آتاماتای یادگیر توزیع شده

مهدی قربعلی پور درو، محمدرضا میبدی، سعید شیری قیداری

دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران

{ghorbalipoor, mmeybodi, shiry}@aut.ac.ir

چکیده - دسته ماکزیمال در یک گراف، مجموعه‌ای از رئوس می‌باشد که در آن هر دو راس دلخواه با هم مجاور بوده و به علاوه زیر مجموعه هیچ دسته بزرگتری نمی‌باشد. این مساله *NP-hard* بوده و الگوریتم‌های تقریبی متعددی برای آن ارائه شده است. آتاماتای یادگیر یک ابزار جستجوی عمومی بوده و برای حل تعدادی از مسائل *NP-hard* به کار برده شده است. در این مقاله با استفاده از آتاماتای یادگیر توزیع شده الگوریتمی برای حل مساله دسته ماکزیمال ارائه شده و سپس کارایی این الگوریتم روی تعدادی از نمونه مساله‌های دسته ماکزیمال آزمایش گردیده و با بعضی روشهای موجود مقایسه شده است. نتایج این مقایسات حاکی از کاراتر بودن این روش نسبت به روشهای موجود می‌باشد.

کلید واژه- دسته ماکزیمال، آتاماتای یادگیر، آتاماتای یادگیر توزیع شده

۱- معرفی

گراف همبند و غیر جهتدار $G=<V,E>$ که درآن مجموعه رئوس گراف $G=\{v_1,v_2,...,v_n\}$ و $E=\{e_1,e_2,...,e_m\}$ مجموعه یالهای گراف می‌باشد را در نظر بگیرید. زیر مجموعه $W\subseteq V$ ، یک دسته از G نامیده می‌شود هرگاه به ازای هر دو عضو دلخواه $v,w\in V$ ، یال $e=\{v,w\}$ متعلق به E باشد. یک دسته ماکزیمال دسته‌ای از گراف با بیشترین کاردینالیتی می‌باشد.

این مساله یک مساله بهینه‌سازی NP-hard بوده [1] و در [10] نشان داده شده که اگر $P\neq NP$ باشد آنگاه هیچ الگوریتم با زمان چند جمله‌ای به ازای هر $0<\varepsilon$ قادر به تقریب دسته ماکزیمال با ضریبی از $n^{1-\varepsilon}$ نیست.

برای حل دقیق مساله MC روشهایی مانند الگوریتمهای شاخه و قید ارائه شده‌اند [11 ,12] ولی کارایی این روشها محدود به گرافهای کوچک ویا گرافهای اسپارس می‌باشد. از اینرو الگوریتمهای تقریبی زیادی برای حل این مساله ارائه شده است. از جمله می‌توان به جستجوی پراکنده [13]، کلونی مورچه‌ها[14] ،شبکه های عصبی[15]، ARH [۱] [4] و COH [۲] [16] اشاره کرد.

از کاربردهای مربوط به دسته ماکزیمم می‌توان به مواردی نظیر انتخاب پروژه، دسته‌بندی، تحمل نقص [۳] ، کدگذاری، بینایی ماشین، شبکه‌های مبایل،آنالیز خوشه‌بندی [۴]، آنالیز بیولوژیکی، اقتصاد، بازیابی اطلاعات، انتقال سیگنال و بیو شیمی اشاره کرد. برای یافتن اطلاعات بیشتر درباره کاربردهای مساله دسته ماکزیمم به [3,17] مراجعه کنید.

حل مساله دسته ماکزیمال ارتباط تنگاتنگی با حل دو مساله NP-hard به نامهای پوشش راسی مینیمال [۵] و مجموعه مستقل ماکزیمال [۶] دارد و هر الگوریتمی که برای حل یکی از این مسائل به کار برده شود می‌تواند برای حل دو مساله دیگر نیز به کار برده شود. این مطلب از معادل بودن سه گزاره زیر نتیجه می‌شود:

محل برگزاری: تهران. بزرگراه شهید بابایی. لویزان. دانشگاه صنعتی مالک اشتر
دبیرخانه: اصفهان. شاهین شهر. خیابان فردوسی . دانشگاه صنعتی مالک اشتر . مجتمع علوم کاربردی . مرکز تحقیقات ریاضی
ص.پ: ۸۳۱۴۵/۱۱۵ تلفن: ۵۲۳۰۱۶۰-۰۳۱۲ فاکس: ۵۲۳۰۱۶۱-۰۳۱۲
Web site: http://isfs.mut-es.ac.ir E-mail: mrc@mut-es.ac.ir

لف- A یک دسته ماکزیمال از G است.

ب- A یک مجموعه مستقل ماکزیمال از $\overline{G}$ (مکمل G) است.

پ- $V-A$ یک پوشش راسی مینیمال از $\overline{G}$ است.

نتیجه منطقی گزاره‌های بالا این است که هر الگوریتمی که برای حل هر کدام از این سه مساله ارائه شود می‌تواند با الگوریتمهای ارائه شده برای دو مساله دیگر نیز مقایسه شود.

در این مقاله یک الگوریتم جدید مبتنی بر اتوماتای یادگیر توزیع شده برای حل مساله مساله دسته ماکزیمال ارائه می-دهیم. معیار جدیدی برای تعیین پاداش و جریمه به آتاماتای یادگیر معرفی می‌کنیم و نتایج به دست آمده از به کارگیری این الگوریتم را در مقایسه با الگوریتمهای گزارش شده ARH [4]، CMPF [5] [7]، PCAOP [6] [8] و الگوریتم گزارش شده در [2] ارائه می‌دهیم.

ادامه مقاله به صورت زیر سازمان داده شده است. در بخش ۲ آتاماتاهای یادگیر معرفی می شود و الگوریتم یادگیری خطی و دنباله رو خطی را برای این نوع از آتاماتاها ارائه می دهیم و سپس الگوریتم یادگیری پیشنهادی خود را برای آتاماتاهای یادگیر معرفی می کنیم. در بخش ۳ آتاماتای یادگیر توزیع شده را شرح می دهیم و در بخش ۴ الگوریتم پیشنهادی را برای حل مساله دسته ماکزیمال بیان می کنیم. در بخش ۵ نتایج آزمایشی آورده خواهد شد و بخش آخر نتیجه گیری می باشد.

۲- آتاماتاهای یادگیر

آتاماتای یادگیر یک مدل انتزاعی است که تعداد محدودی عمل را می‌تواند انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی شده و پاسخی به آتاماتای یادگیر داده می‌شود. آتاماتای یادگیر از این پاسخ استفاده نموده و عمل خود را برای مرحله بعد انتخاب می‌کند.

محیط را می‌توان توسط سه تایی $E=\{\alpha,\beta,c\}$ نشان داد که در آن $\alpha=\{\alpha_1,\alpha_2,...,\alpha_r\}$ مجموعه ورودیها، $\beta=\{\beta_1,\beta_2,...,\beta_m\}$ مجموعه خروجیها و $c=\{c_1,c_2,...,c_r\}$ مجموعه احتمالهای جریمه می‌باشد. هرگاه β مجموعه ای 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. c_i احتمال آنکه عمل α_i نتیجه نامطلوب داشته باشد می‌باشد.

آتاماتای یادگیر با ساختار متغیر: آتاماتای یادگیر با ساختار متغیر توسط ۴ تایی $\{\alpha,\beta,p,T\}$ نشان داده می-شود که در آن $\alpha=\{\alpha_1,\alpha_2,...,\alpha_r\}$ مجموعه عملهای آتاماتا، $\beta=\{\beta_1,\beta_2,...,\beta_m\}$ مجموعه ورودیهای آتاماتا، $p=\{p_1,p_2,...,p_r\}$ بردار احتمال انتخاب هر یک از عملها و $p(n+1)=T[\alpha(n),\beta(n),p(n)]$ الگوریتم یادگیری می-باشد. در این نوع آتاماتا اگر عمل α_i در مرحله n-ام انتخاب شود و پاسخ مطلوب از محیط دریافت کند احتمال $p_i(n)$ افزایش یافته و سایر احتمالها کاهش می‌یابند. برای پاسخ نامطلوب ، احتمال $p_i(n)$ کاهش یافته و سایر احتمالها افزایش می‌یابند.

۲-۱- الگوریتم یادگیری پیشنهادی

الگوریتم یادگیری که برای آتاماتای یادگیر استفاده می کنیم ترکیبی از دو الگوریتم یادگیری خطی و دنباله رو[9] می باشد. این دو الگوریتم دارای ضعف و قوتهای خاص خود هستند. در الگوریتم یادگیری خطی به میزان نسبت پاداش به انتخاب عملها توجه نمی‌شود ولی این مزیت را دارد که احتمال انتخاب یک عمل هرچند ممکن است خیلی کوچک باشد ولی هرگز صفر نخواهد شد. از طرفی در الگوریتم یادگیری دنباله رو هر چند که به میزان نسبت پاداش به انتخاب عملها توجه می‌شود ولی این ضعف را دارد که به طور ناخواسته احتمال انتخاب بعضی از عملها را صفر می-کند. الگوریتم یادگیری که آنرا معرفی می‌کنیم نقاط ضعف دو نوع الگوریتم یادگیری قبلی را پوشش می‌دهد و از نقاط قوت هر دو روش استفاده می‌کند.

محل برگزاری: تهران. بزرگراه شهید بابایی. لویزان. دانشگاه صنعتی مالک اشتر
دبیرخانه: اصفهان. شاهین شهر. خیابان فردوسی . دانشگاه صنعتی مالک اشتر . مجتمع علوم کاربردی . مرکز تحقیقات ریاضی
ص.پ: ۸۳۱۴۵/۱۱۵ تلفن: ۵۲۳۰۱۶۰-۰۳۱۲ فاکس: ۵۲۳۰۱۶۱-۰۳۱۲
Web site: http://isfs.mut-es.ac.ir E-mail: mrc@ mut-es.ac.ir

الگوریتم یادگیری پیشنهادی مانند الگوریتم یادگیری دنباله رو عمل می‌کند با این تفاوت که نحوه پاداش عملهای آتاماتای به صورت زیر تغییر خواهد کرد.

الف- اگر $\beta(t)=0$ آنگاه

$$p_i(t+1)=(1-a)\,p_i(t) \quad \forall i\neq m$$
$$p_m(t+1)=(1-a)\,p_m(t)+a \qquad (۱)$$

ب- در غیر این صورت

$$p_i(t+1)=p_i(t) \quad \forall i \qquad (۲)$$

در عبارات بالا a پارمتر پاداش نامیده می‌شود که نرخ پاداش را برای عملهای آتاماتا تعیین می‌کند.

۳- آتاماتای یادگیر توزیع شده

آتاماتای یادگیر توزیع شده شبکه‌ای است از آتاماتای یادگیر که برای حل مساله خاصی با یکدیگر همکاری می‌کنند. در این شبکه ازآتاماتای همکار در هر زمان یک آتاماتا فعال است و تعداد اعمال قابل انجام توسط یک آتاماتا در DLA برابر با تعداد آتاماتاهای یادگیر متصل به این آتاماتا می‌باشد. انتخاب یک عمل توسط آتاماتا در این شبکه، باعث فعال شدن آتاماتای متصل شده به این آتاماتا و متناظر با این عمل می‌گردد. به عبارت معادل، انتخاب یک عمل توسط یک آتاماتا در این شبکه متناظر با فعال شدن یک آتاماتای دیگر در این شبکه است[8].

ساختار DLA را می توان به صورت گراف $G(V,E)$ که در آن $V=\{LA_1,LA_2,...,LA_n\}$ و $E\subseteq V\times V$ است، در نظر گرفت. وجود یال (LA_i,LA_j) در این گراف بدان معناست که انتخاب عمل α_j^i توسط LA_i باعث فعال شدن LA_j می‌گردد.

۴- الگوریتم پیشنهادی

در این بخش الگوریتم پیشنهادی خود را برای حل مسال دسته ماکزیمال ارائه می‌دهیم. ابتدا دو تعریف کوتاه زیر را ارائه می‌دهیم که به درک راحت الگوریتم کمک می‌کند.

مجموعه راسهای مجاور راس $v\in V$ از گراف G برابر است با $N(v)=\{w\in V\mid (v,w)\in E\}$.

مجموعه راسهای غیر مجاور راس $v\in V$ از گراف G برابر است با $\overline{N}(v)=\{w\in V\mid w\neq v,(v,w)\notin E\}$.

گراف G را در نظر بگیرید. به وضوح کلیه راسهایی از V که دارای درجه $n-1$ می‌باشند عضو دسته ماکزیمال نیز هستند. فرض کنید که $G'=<V',E'>$ زیر گرافی از G باشد که با حذف کلیه راسهای درجه $n-1$ از گراف G به دست آمده باشد. در این صورت G' هیچ راسی از درجه $|V'|-1$ نخواهد داشت زیرا در غیر این صورت این راس در گراف G باید راسی از درجه $n-1$ می‌بوده است و قبلا حذف می‌گردیده است و این تناقض با تعلق آن به V' دارد.

در صورتی که الگوریتم ما بتواند دسته ماکزیمال را در G' بیابد آنگاه این الگوریتم می‌تواند دسته ماکزیمال را در G نیز بیابد. از اینرو بدون کاستن کلیات فرض می‌کنیم که گراف همبند و غیر جهتدار $G=<V,E>$ دارای راس از درجه $n-1$ نباشد و الگوریتم پیشنهادی را روی این گراف اجرا خواهیم کرد.

اولین مرحله الگوریتم ساخت اتوماتای یادگیر توزیع شده متناظر با گراف مساله می باشد. به این صورت که به هر راس $v\in V$ از گراف G آتاماتای یادگیر LA_v را متناظر می‌کنیم و مجموعه عملهای LA_v را فعال سازی مجموعه آتاماتاهای $\{LA_w\mid w\in N(v)\}$ در نظر می‌گیریم.

پس از ساختن DLA متناظر با گراف مساله، در گام نخست یکی از راسهای گراف به صورت تصادفی انتخاب (مثلا v) و به مجموعه MC اضافه می‌شود و سپس آتاماتاهای یادگیر متناظر با مجموعه $\overline{N}(v)$ غیر فعال می‌شوند و اقدامهای متناظر با انتخاب این آتاماتاها از لیست اقدامهای آتاماتاهای یادگیر فعال باقی مانده غیر فعال می‌گردد. آتاماتای LA_v به نوبه خود یکی از اقدامهایش را بر طبق بردار احتمال اقدامها انتخاب کرده و راس متناظر با اقدام انتخاب شده را به MC اضافه کرده و سپس آتاماتاهای یادگیر متناظر با مجموعه راسهای غیر مجاور این گره ، غیر فعال می‌گردند و در لیست

محل برگزاری: تهران. بزرگراه شهید بابایی. لویزان. دانشگاه صنعتی مالک اشتر
دبیرخانه: اصفهان. شاهین شهر. خیابان فردوسی . دانشگاه صنعتی مالک اشتر . مجتمع علوم کاربردی . مرکز تحقیقات ریاضی
ص.پ: ۸۳۱۴۵/۱۱۵ تلفن: ۵۲۳۰۱۶۰-۰۳۱۲ فاکس: ۵۲۳۰۱۶۱-۰۳۱۲
Web site: http://isfs.mut-es.ac.ir E-mail: mrc@ mut-es.ac.ir

اقدامهای آتاماتاهای یادگیر فعال باقی مانده غیر فعال می‌شوند. این فرایند انتخاب تا غیر فعال شدن همه آتاماتاهای یادگیر ادامه می‌یابد.

هنگامی که کلیه آتاماتاها غیر فعال شدند کاردینالیتی مجموعه MC حساب شده و با کاردینالیتی مجموعه OPT(MC) که در ابتدا تهی است مقایسه می‌گردد. در صورتی‌که کاردینالیتی MC فعلی از کاردینالیتی OPT(MC) بیشتر باشد OPT(MC) برابر با MC گرفته می‌شود.

در صورتیکه کاردینالیتی MC فعلی از کاردینالیتی OPT(MC) بزرگتر یا مساوی باشد با فرض اینکه $MC=\{w_1,w_2,...,w_k\}$ و به ازای $i=1,...,k-1$ آتاماتای LA_{w_i} آتاماتای $LA_{w_{i+1}}$ را فعال کرده باشد به عمل $\alpha_{w_{i+1}}^{w_i}$ (عمل مربوط به LA_{w_i}، متناظر با فعالسازی آتاماتای $LA_{w_{i+1}}$) به ازای $i=1,...,k-1$ پاداش داده می‌شود. سپس مجموعه های MC برابر تهی گرفته می‌شود و کلیه آتاماتاهای یادگیر شبکه و مجموعه اقدامهای آنها فعال می‌شوند و سپس یک تکرار جدید صورت می‌گیرد. این تکرار تا برقراری یک شرط خاتمه ادامه می‌یابد.

در اینجا توضیح دو مطلب ضروری به نظر می‌رسد. یکی طریقه انتخاب یک اقدام(عمل) توسط آتاماتاهای یادگیر موجود در شبکه و دیگری شرط خاتمه الگوریتم است که در زیر توضیح داده می‌شود.

۴-۱- طریقه انتخاب یک اقدام توسط آتاماتاهای یادگیر موجود درشبکه

فرض کنید که آتاماتای یادگیر LA_v دارای مجموعه اقدامهای $\alpha=\{\alpha_1,\alpha_2,...,\alpha_r\}$ باشد و بردار احتمال متناظر با این اقدامها $p=\{p_1,p_2,...,p_r\}$ باشد. فرض کنید در لحظه‌ای که آتاماتا می‌خواهد اقدامی از مجموعه اقدامهای خود انتخاب کند فقط مجموعه اقدامهای $\eta=\{\eta_1,\eta_2,...,\eta_k\}\subseteq\alpha$ فعال باشند و بردار احتمال متناظر با این اقدامها $\{q_1,q_2,...,q_k\}\subseteq p$ باشد. شاید ساده ترین روش برای انتخاب یک اقدام این باشد که قرار دهیم $\bar{q}_i=\frac{q_i}{\sum_{j=1..k} q_j}$ و با توجه به بردار احتمال $\bar{q}=\{\bar{q}_1,\bar{q}_2,...,\bar{q}_k\}$ یکی از عملها را انتخاب کنیم. متاسفانه این کار موجب تولید نتایج ضعیفی می‌شود. راه حلی که برای مقابله با این مشکل پیشنهاد کرده‌ایم استفاده از یک بردار احتمال تغییر یافته به جای بردار احتمال واقعی می‌باشد. به این صورت که با فرض اینکه $\{v_1,v_2,...,v_k\}$ راسهای متناظر با مجموعه اقدامهای η باشد از بردار احتمال تغییر یافته $\overline{mq}=\{\overline{mq}_1,\overline{mq}_2,...,\overline{mq}_k\}$ استفاده کنیم که در آن $\overline{mq}_i$ به صورت زیر تعریف می‌شود.

$$\overline{mq}_i=\frac{(\bar{q}_i/(n-1-\deg_G(v_i)))^{\beta}}{\sum_{j=1,...,k}(\bar{q}_j/(n-1-\deg_G(v_j)))^{\beta}} \qquad (۳)$$

در این رابطه $\beta\ge 1$ یک نرمال کننده بوده و موجب می‌شود که احتمال انتخاب اقدامهایی که دارای $\bar{q}_i$ بیشتر و درجه $\deg_G(v_i)$ بیشتر باشند بالاتر رود و احتمال انتخاب اقدامهایی که دارای $\bar{q}_i$ کمتر و درجه $\deg_G(v_i)$ کمتر باشند پایین تر بیاید.

مقدار مورد نظر برای پارامتر β در گرافهای مختلف متفاوت است نتایج آزمایشی ما نشان داد که بهترین بازه برای β، ۵ تا ۷ می‌باشد.

۴-۲- شرط خاتمه الگوریتم

فرض کنید که $OPT(MC)=\{w_1,w_2,...,w_k\}$ جدیدترین MC به دست آمده باشد و به ازای $i=1,...,k-1$ آتاماتای LA_{w_i} آتاماتای $LA_{w_{i+1}}$ را فعال کرده باشد و احتمال انتخاب اقدام $\alpha_{w_{i+1}}^{w_i}$ توسط آتاماتای LA_{w_i} به ازای $i=1,...,k-1$ برابر با $P(\alpha_{w_{i+1}}^{w_i})$ باشد. تعریف می‌کنیم

$$P(w_i)=\max_{w_j\in OPT(MIS)-\{w_i\}} P(\alpha_{w_j}^{w_i})$$

و

$$P(MIS)=\prod_{i=1,...,k} P(w_i)\ .$$

معیاری که برای شرط خاتمه الگوریتم در نظر می‌گیریم

محل برگزاری: تهران. بزرگراه شهید بابایی. لویزان. دانشگاه صنعتی مالک اشتر
دبیرخانه: اصفهان. شاهین شهر. خیابان فردوسی . دانشگاه صنعتی مالک اشتر . مجتمع علوم کاربردی . مرکز تحقیقات ریاضی
ص.پ: ۸۳۱۴۵/۱۱۵ تلفن: ۰۳۱۲-۵۲۳۰۱۶۰ فاکس: ۰۳۱۲-۵۲۳۰۱۶۱
Web site: http://isfs.mut-es.ac.ir E-mail: mrc@ mut-es.ac.ir

این است که یا تعداد تکرارها از یک عدد ثابتی مانند *MaxIteration* بیشتر باشد و یا اینکه $P(MC)$ از مقدار ثابتی مانند *Threshold* بیشتر باشد.

۵- نتایج آزمایشی

الگوریتم پیشنهادی بر روی تعدادی از گرافهای نمونه موجود در [9] اجرا گردید. نتایج بدست آمده با نتایج بدست آمده برای الگوریتمهای گزارش شده ARH [4]، CMPF [5]، PCAOP [6] و الگوریتم گزارش شده در [2] مقایسه گردید. نتایج مقایسه حاکی از برتری الگوریتم پیشنهادی نسبت به روشهای قبلی می‌باشد.

جدول (۱) نتایج حاصل از اجرای الگوریتم پیشنهادی را روی تعدادی از گرافها ی نمونه نشان می‌دهد.

در جدول (۲) نتایج اجرای الگوریتم پیشنهادی ما با نتایج اجرای الگوریتم ارائه شده در [2] برای تعدادی از گرافها مقایسه شده است. این مقایسه نشان دهنده کاراتر بودن این الگوریتم نسبت به الگوریتم ارائه شده در [2] می‌باشد.

در جدول (۳) الگوریتم پیشنهادی با سه الگوریتم ARH ، CMPF و PCAOP مقایسه شده است.

از جدول (۲) دیده می‌شود که الگوریتم پیشنهادی دارای برتری محسوس نسبت به الگوریتم ارائه شده در [2] داشته و با توجه به جدول (۳) از CMPF و ARH جوابهای بهتری تولید کرده است. الگوریتم پیشنهادی تقریبا دارای زمان اجرای برابر با الگوریتم ارائه شده در [2] می‌باشد و از آنجا که در [2] نشان داده شده است که الگوریتم موجود در آن دارای سرعت اجرایی بهتری ازPCAOP می‌باشد نتیجه می‌شود که الگوریتم پیشنهادی نیز نسبت به PCAOP دارای زمان اجرای بهتری بوده و از اینرو به این الگوریتم نیز ترجیح داده می‌شود.

جدول ۱:نتایج اجرای الگوریتم پیشنهادی روی تعدادی از گرافها

متوسط تکرارها برای رسیدن به جواب متوسط	جواب متوسط	بهترین جواب	دسته ماکزیمم	تعداد راسها	نام گراف
۸	۱۶	۱۶	۱۶	۴۵	MANN_a9
۳۰۰۰	۱۲۵	۱۲۶	۱۲۶	۳۷۸	MANN_a27
۱۰۰	۱۱	۱۱	۱۱	۱۷۱	Keller4

جدول ۲: مقایسه الگوریتم پیشنهادی با الگوریتم ارائه شده در [۲]

جواب متوسط الگوریتم	جواب متوسط در [2]	بهترین جواب الگوریتم	بهترین جواب در [2]	دسته ماکزیمم	تعداد راسها	نام گراف
۱۶	۱۴	۱۶	۱۶	۱۶	۴۵	MANN_a9
۱۲۵	۱۱۵	۱۲۶	۱۲۶	۱۲۶	۳۷۸	MANN_a27
۱۱	۱۱	۱۱	۱۱	۱۱	۱۷۱	Keller4

جدول۳: مقایسه الگوریتم پیشنهادی با بعضی الگوریتم‌های موجود

بهترین جواب در PCAOP	بهترین جواب در CMPF	بهترین جواب در ARH	بهترین جواب الگوریتم	نام گراف
۱۶	۱۶	۱۶	۱۶	MANN_a9
۱۲۶	۱۱۷	۱۲۵	۱۲۶	MANN_a27
۱۱	۸	۱۱	۱۱	Keller4

۶- نتیجه گیری

در این مقاله ابتدا مساله NP-hard دسته ماکزیمال را معرفی و ارتباط آن با دو مساله NP-hard پوشش راسی مینیمال و مجموعه مستقل ماکزیمال را بیان کردیم. الگوریتم یادگیری جدیدی برای آتاماتاهای یادگیر موجود در شبکه DLA ارائه نمودیم و الگوریتمی مبتنی بر آتاماتای یادگیر توزیع شده برای حل مساله دسته ماکزیمال پیشنهاد کردیم. الگوریتم پیشنهادی با تعدادی از الگوریتمهای گزارش شده مقایسه گردید. نتایج آزمایشی حاکی از کاراتر بودن این روش نسبت به الگوریتمهای موجود می‌باشد.

مراجع

[1] M. Garey and D. Johnson, *“Computers and Intractability: A Guide to the Theory of NP completeness”,* Freeman, San Francisco, CA, 1979.

[2] M. Alipour and M. R. Meybodi, *"Solving Maximal independent Set Problem Using Distributed Learning Automata"*, Proceedings of 14th Iranian Electrical Engineering Conference(ICEE2006), Amirkabir University, Tehran, Iran, May 16-18, 2006.

[3] P. M. Pardalos, J. Rappe, and M. G. C. Resende, *"An Exact Parallel Algorithm for the Maximum Clique Problem"*, 1998.

[4] I. M. Bomze, M. Budinich, M. Pelillo and Claudio, *“Annealed Replication: A New Heuristic for the Maximum Clique Problem”*, 2000.

[5] J. Abello, *“Finding independent sets in a graph using continuous multivariable polynomial formulations”*, J. Global Optim., Vol. 21, pp. 111-137, 2001.

[6] J. L. Wong, F. Koushanfar, S. Meguerdichian and M. Potkonjak, *“A probabilistic constructive approach to optimization problems”*, Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design, November 04-08, San Jose, California, 2001.

[7] B. J. Oomen and M. Agache, *“Acomparsion of Continues and Discretized Pursuit Learning Scheme.”,* Technical Report, Carleton University, Ottawa, 2002.

[8] M. R. Meybodi and H. Beigy, *"Solving Stochastic Path Problem Using Distributed Learning Automata"*, Proceedings of The Sixth Annual International CSI Computer Conference, CSICC2001, Isfahan, Iran, pp. 70-86, Feb. 20- 22, 2001.

[9] http://mat.gsia.cmu.edu/COLOR04/INSTANCES/

[10] J. Håstad, *"Clique is* hard *to approximate within* $n^{1-\varepsilon}$*"*, ActaMath., Vol. 182, pp. 105–142, 1999.

[11] P .R. J. Östergård, *"A fast algorithm for the maximum clique problem"*, Discrete Appl. Math., Vol. 120, pp. 195–205, 2002.

[12] E. Tomita and T. Seki, *"An efficient branch-and-bound algorithm for finding a maximum clique"*, in: Proc. 4th Internat. Conf.on Discrete Mathematics and Theoretical Computer Science, in: Lecture Notes in Comput. Sci., Springer, Berlin, Vol. 2731, pp. 278–289, 2003.

[13] L. Cavique, C. Rego and I. Themido, *"A scatter search algorithm for the maximum clique problem"*, in: Essays and Surveys in Metaheuristics, Kluwer, Dordrecht, pp. 227–244, 2001.

[14] S. Fenet and C. Solnon, *"Searching for maximum cliques with ant colony optimization"*, in: Applications of Evolutionary Computing, in: Lecture Notes in Comput. Sci., Springer, Berlin, Vol. 2611, pp. 236–245, 2003.

[15] N. Funabiki, Y. Takefuji and K.C. Lee, *"A neural network model for finding a near-maximum clique"*, Parallel Distrib. Comput., Vol.14, No. 3, pp. 340–344, 1992.

[16] S. Burer, R. D. C. Monteiro and Y. Zhang, *"Maximum stable set formulations and heuristics based on continuous optimization"*, Math. Programming, Vol. 94, pp. 137–166, 2002.

[17] I. M. Bomze, M. Budinich, P.M. Pardalos and M. Pelillo, "*The maximum clique problem*", in: D. Z. Du and P.M. Pardalos (Eds.), Handbook of Combinatorial Optimization (suppl. vol. A), Kluwer, Dordrecht, pp. 1–74, 1999.

---

1 Annealed Replication Heuristic
2 Continuous Optimization Heuristics
3 fault tolerance
4 cluster analysis
5 Minimal Vertex Cover
6 Maximal Independent Set
7 continuous multivariable polynomial formulations
8 probabilistic constructive approach to optimization problems
9 persuit