

لینک های مفید



عضویت
در خبرنامه



کارگاه های
آموزشی



سرویس
ترجمه تخصصی
STRS



فیلم های
آموزشی



بلاگ
مرکز اطلاعات علمی



سرویس های
ویژه



یک الگوریتم نوین جهت رنگ آمیزی گراف با استفاده از اتوماتای یادگیر

علی اکبر دادجویان

دانشگاه آزاد اسلامی واحد شبستر

Aliakbar_z2004@yahoo.com

عباس میرزایی ثمرین

بورسیه هیات علمی دانشگاه آزاد اسلامی واحد اردبیل

a.mirzaei@iaut.ac.ir

حبیب مطیع قادر

دانشگاه آزاد اسلامی واحد تبریز - باشگاه پژوهشگران جوان

Habib_moti@yahoo.com

چکیده - رنگ آمیزی گراف یکی از مسائل Np-Complete به شمار می‌رود. یکی از کاربردهای این مسئله رنگ آمیزی نقشه‌ها است. در این مقاله یک الگوریتم جدید برای رنگ آمیزی گراف با استفاده از اتوماتای یادگیر پیشنهاد می‌شود. فرآیند یادگیری با تعدادی از اتوماتاهای تصادفی شروع می‌شود. هر اتوماتا به تنهایی نمایش دهنده یک رنگ آمیزی تصادفی می‌باشد. با تکرار فرآیند یادگیری رنگ آمیزی بهبود می‌یابد. در مقایسه با الگوریتم‌های ژنتیک، الگوریتم پیشنهادی سریعتر به جواب نزدیک به بهینه می‌رسد. دلیل آن این است که الگوریتم‌های ژنتیک به دنبال کروموزوم بهینه از میان جمعیت‌ها هستند و به جایگاه ژن‌ها در کروموزومها اهمیت داده نمی‌شود ولی در الگوریتم پیشنهادی سعی می‌شود تا با استفاده از اتوماتای مهاجرت اشیاء جایگاه بهینه ژن‌ها مشخص شود. الگوریتم پیشنهادی در تعداد مراحل تکرار یادگیری کمتری در مقابل تعداد نسل الگوریتم ژنتیک به جواب بهینه نزدیک می‌شود. نتایج شبیه‌سازی حاصل از الگوریتم پیشنهادی بر روی گراف‌های مطرح با الگوریتم ژنتیک مورد مقایسه قرار گرفته است.

واژه‌های کلیدی: اتوماتای یادگیر، اتوماتای مهاجرت اشیاء، رنگ آمیزی گراف.

۱- مقدمه

پاداش رفته رفته جایگاه مناسب ژن‌ها در کروموزومها را پیدا می‌کند. هدف از الگوریتم پیشنهادی پیدا کردن جایگاه مناسب برای ژن‌ها با استفاده از اتوماتای یادگیر است. در الگوریتم پیشنهادی هر کروموزوم معادل یک اتوماتا و هر ژن معادل یک اقدام^۲ از اتوماتا می‌باشد. هر اتوماتا نمایش دهنده یک رنگ آمیزی تصادفی می‌باشد. الگوریتم پیشنهادی با تکرار فرآیند یادگیری جایگاه مناسب ژن‌ها را بهبود می‌بخشد.

در ادامه در بخش ۲ مسئله رنگ آمیزی گراف توضیح داده می‌شود. سپس در بخش ۳ اتوماتای یادگیر معرفی شده و در بخش ۴ الگوریتم جدید پیشنهاد می‌شود. در نهایت در بخش ۵ نتایج حاصل از شبیه سازی الگوریتم پیشنهادی مورد ارزیابی قرار می‌گیرد.

۲- شرح مسئله

یک گراف به صورت دوتایی $G=(V, E)$ نشان داده می‌شود که در آن V مجموعه گره‌ها و $E \subseteq V \times V$ مجموعه یال‌ها است. مسئله رنگ آمیزی گراف جزو مسائل NP-Complete به شمار می‌رود. در این گونه مسائل با افزایش تعداد گره‌های گراف، زمان مورد نیاز برای پیدا کردن راه حل بهینه، به صورت نمایی رشد پیدا می‌کند. به همین دلیل برای اینکه راه حل نزدیک به بهینه برای اینگونه مسائل پیدا شود، به ناچار از روشهای

گراف‌ها یکی از مهمترین ابزارهایی هستند که کاربردهای فراوانی را دارند، یکی از مهمترین کاربردهای گراف‌ها، مسئله رنگ آمیزی گراف‌ها می‌باشد. در این مسئله برای گراف G به تعداد V گره و E یال وجود دارد. هدف از رنگ آمیزی گراف این است که به هر کدام از گره‌های گراف یک رنگ اختصاص یابد، بطوری‌که هیچ دو گره مجاور دارای رنگ یکسان نبوده و از حداقل تعداد رنگ‌ها برای رنگ آمیزی استفاده شود. در میان روش‌های مکاشفه‌ای^۱ غیرقطعی، الگوریتم ژنتیک یکی از مهمترین الگوریتم‌هایی است که برای رنگ آمیزی گراف مورد استفاده قرار گرفته است. الگوریتم ژنتیک کار خود را با تعدادی جمعیت تصادفی شروع کرده و با تکرار تولید نسل به دنبال کروموزوم شایسته‌تر می‌گردد. در الگوریتم ژنتیک هدف پیدا کردن شایسته‌ترین کروموزوم از میان جمعیت‌ها است و به جایگاه ژن‌ها در کروموزومها هیچ اهمیتی داده نمی‌شود. اگر بتوان جایگاه مناسب ژن‌ها در کروموزومها را مشخص کرد در این صورت می‌توان در تعداد نسل‌های بسیار کمتری به جواب نزدیک به بهینه رسید. برای پیدا کردن جایگاه مناسب ژن‌ها در کروموزومها می‌توان از اتوماتای یادگیر استفاده نمود. اتوماتای یادگیر با طی فرآیند یادگیری و با اعمال عملگرهای جریمه و

² Action

¹ Heuristic

برابر است با تعداد گره‌های گراف). مجموعه وضعیت‌ها و N عمق حافظه برای اتوماتا می‌باشد. مجموعه وضعیت‌های این اتوماتا به K زیر مجموعه $\{\phi_1, \phi_2, \dots, \phi_N\}$ و $\{\phi_{N+1}, \phi_{N+2}, \dots, \phi_{2N}\}$ و ... و $\{\phi_{(K-1)N+1}, \phi_{(K-1)N+2}, \dots, \phi_{KN}\}$ افزاز می‌شود و هر کدام از گره‌ها بر اساس این که در کدام وضعیت قرار داشته باشند دسته بندی می‌گردند. در مجموعه وضعیت‌های اقدام k ، به وضعیت $\phi_{(k-1)N+1}$ وضعیت داخلی و به وضعیت ϕ_{kN} ، وضعیت مرزی گفته می‌شود. گرهی که در وضعیت $\phi_{(k-1)N+1}$ قرار دارد گره با اهمیت بیشتر و گرهی که در وضعیت ϕ_{kN} گره با اهمیت کمتر نامیده می‌شود. در شکل ۱ شبه کد مربوط به الگوریتم پیشنهادی نمایش داده شده است.

Algorithm GCLA(Graph Coloring using Learning Automata)
 Input G : Graph
 Output S : A near optimal coloring for G
 Method
 1. Generate an initial population, P, of randomly created coloring OMA for G
 2. For Generation = 1 to the Maximum no. generations, M, do
 3. For each automaton, A, in P do
 4. Select an action, α_i randomly
 5. Apply relation 1 described in section 4.2.2 to reward or penalize α_i
 6. if rewarded and $j < N$, upgrade its state from ϕ_{ij} to ϕ_{ij+1}
 7. else if penalized and $j > 1$ downgrade its state from ϕ_{ij+1} to ϕ_{ij}
 8. else if penalized and $j = 1$ swap the action α_i with α_j such that satisfy color α_i
 End

شکل ۱. شبه کد الگوریتم پیشنهادی

در حالت کلی الگوریتم پیشنهادی از دو بخش زیر تشکیل شده است:

۱. تولید جمعیت اولیه
۲. عملگر جریمه و پاداش

۴-۱- تولید جمعیت اولیه

در ابتدا P اتوماتای تصادفی ایجاد شده و سپس در هر اتوماتا گره‌های گراف به اقدام‌های اتوماتاها تخصیص داده می‌شوند. پس از تخصیص گره‌ها به اقدام‌های اتوماتا الگوریتم شکل ۲ به تمام اتوماتاها اجرا می‌شود. با اجرای الگوریتم شکل ۲ به هر کدام از اقدام‌ها یک عدد تصادفی اختصاص داده می‌شود. هر عدد نشان دهنده یک رنگ می‌باشد.

Procedure ACA(Assign Colors to Actions)
 Input N: Nodes
 Output A: Automata
 Method
 1. For $i=1$ to N do
 2. assign i to α_i
 3. For $i=1$ to $N/2$
 4. $x1$ =random number
 5. $x2$ = random number
 6. If $(x1 \neq x2)$ then
 7. Swap α_{x1} value with α_{x2}
 End

شکل ۲. انتساب رنگ‌ها به اقدام‌ها

جستجوی تصادفی که زمان اجرای آن‌ها به صورت خطی است استفاده می‌شود. راه حل‌های جستجوی تصادفی تضمینی در به دست آوردن بهینه‌ترین جواب ندارند، ولی می‌توانند جواب‌های نزدیک به بهینه را به دست آورند. در این مساله هدف از بهترین جواب عبارت است از حداقل تعداد رنگ‌های مورد نیاز جهت رنگ آمیزی گراف بطوری که هیچ دو گرهی هم-رنگ نباشند.

۳- اتوماتای یادگیر

یک اتوماتای یادگیر یک مدل انتزاعی است که به طور تصادفی یک عمل از مجموعه متناهی عمل‌های خود را انتخاب کرده و بر محیط اعمال می‌کند. محیط عمل انتخاب شده توسط اتوماتای یادگیر را ارزیابی کرده و نتیجه ارزیابی خود را توسط یک سیگنال تقویتی به اتوماتای یادگیر اعلام می‌کند. اتوماتای یادگیر با استفاده از عمل انتخاب شده سیگنال تقویتی وضعیت داخلی خود را بروز کرده و سپس عمل بعدی خود را انتخاب می‌کند [1].

محیط را می‌توان توسط سه تایی $E = \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودی‌ها، $\beta = \{\beta_1, \beta_2, \dots, \beta_r\}$ 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، مجموعه β دارای تعداد متناهی عضو می‌باشد و در محیط از نوع S، مجموعه β دارای تعداد نامتناهی عضو می‌باشد. c_i احتمال جریمه شدن عمل α_i است. اتوماتاهای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم‌بندی می‌گردند [2][3][4][5][6].

۴- الگوریتم پیشنهادی

برای رنگ‌آمیزی یک گراف با n گره با استفاده از m رنگ، m^n رنگ-آمیزی مختلف وجود دارد در صورتیکه از اتوماتای یادگیر برای حل مسئله رنگ‌آمیزی گراف استفاده شود، اتوماتای یادگیر باید m^n اقدام داشته باشد که تعداد زیاد اقدام‌ها باعث کاهش سرعت همگرایی می‌شود و به همین منظور از اتوماتای مهاجرت اشیا^۴ توسط اومن^۵ و ما^۶ پیشنهاد شده است که یکی از آن‌ها اتوماتای مهاجرت اشیا مبتنی بر اتوماتای تستلین^۷ می‌باشد. الگوریتم ارایه شده از اتوماتای تستلین برای حل مسئله رنگ آمیزی گراف استفاده می‌کند.

در این اتوماتا $\alpha = \{\alpha_1, \dots, \alpha_k\}$ مجموعه اقدام‌های مجاز برای اتوماتای یادگیر است. این اتوماتا k اقدام دارد (تعداد اقدام‌های این اتوماتا

³ Action

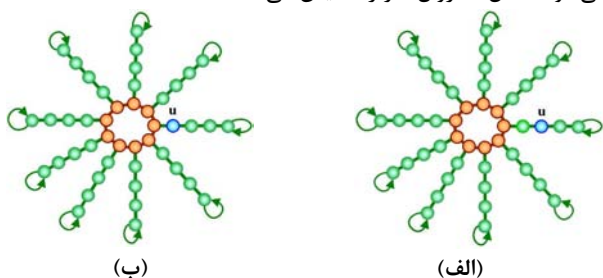
⁴ Object Migrating Automata(OMA)

⁵ Oommen

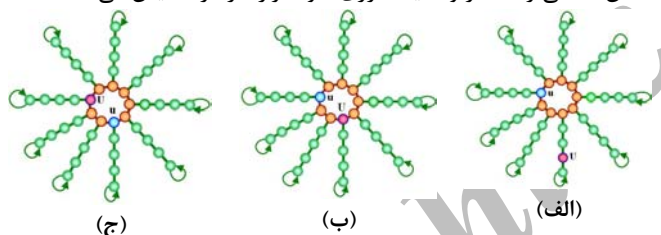
⁶ Ma

⁷ Tsetline

در صورتی که اقدام مورد نظر در وضعیت غیر مرزی قرار داشته باشد، با اعمال عمل جریمه وضعیت آن اقدام به سمت وضعیت مرزی تغییر داده می‌شود. شکل ۶ روال کار را نمایش می‌دهد.



شکل ۶. (الف) وضعیت گره u قبل از جریمه و (ب) وضعیت گره u بعد از جریمه در صورتی که اقدام مورد نظر در وضعیت مرزی قرار داشته باشد، در این صورت مقدار تخصیص داده شده به اقدام مورد نظر (رنگ تخصیص داده شده به اقدام) با عدد تخصیص داده شده به اقدامی جایگزین می‌شود که پس از جایگزینی آن عدد هیچ دو گره مجاور هم‌رنگ نباشد. در صورتی که اقدام جایگزین شونده در وضعیت مرزی خود قرار داشته باشد در این صورت مقدار خود را به اقدام جریمه شونده جایگزین می‌کند ولی در صورتی که اقدام جایگزین شونده در وضعیت مرزی خود قرار نداشته باشد، در این صورت در ابتدا به وضعیت مرزی خود انتقال داده شده و پس از آن مقدار خود را به اقدام جریمه شونده جایگزین می‌کند. شکل ۷ نحوه جریمه شدن اقدامی را که در وضعیت مرزی خود قرار دارد را نمایش می‌دهد.



شکل ۷. (الف) وضعیت گره u قبل از جریمه، (ب) انتقال گره u به وضعیت مرزی و (ج) جایگزینی u به جای u

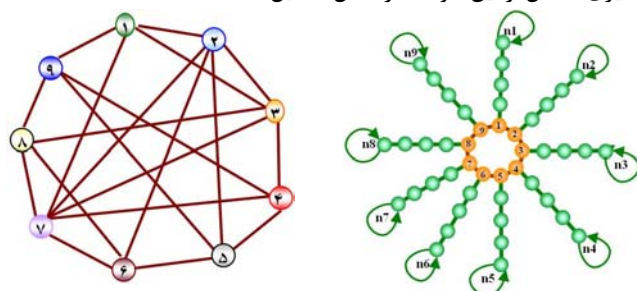
در این بخش مشخص می‌شود که اعمال جریمه و پاداش بر چه اساسی صورت می‌گیرد. یکی از مهمترین کارهایی که در اتوماتاهای یادگیر مطرح است تعیین رابطه‌ای مناسب برای عملگر جریمه و پاداش است. با در نظر گرفتن گره n_i از گراف با استفاده از رابطه ۱ مشخص می‌شود که این گره مورد عمل جریمه قرار گیرد یا مورد عمل پاداش. با در نظر گرفتن مفروضات زیر داریم:

δ : عبارت است از مجموع تمام رنگ‌های گره‌هایی که با گره n در ارتباط هستند.

C : عبارت است از تعداد یال‌های وابسته به گره n_i .

E : عبارت است از تعداد کل یال‌های گراف.

در شکل ۳ یک اتوماتای تصادفی که نمایش دهنده یک رنگ آمیزی تصادفی می‌باشد، نمایش داده شده است. همانطور که مشخص است رنگ آمیزی حاصل از این اتوماتا در شکل ۴ قابل مشاهده است.



شکل ۳. یک اتوماتای تصادفی

شکل ۴. رنگ آمیزی حاصل از اتوماتا

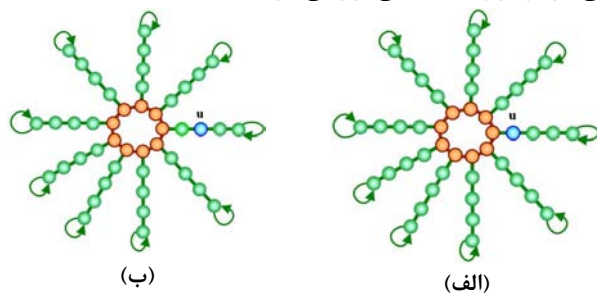
۴-۲- عملگر جریمه و پاداش

در هر یک از اتوماتاها یک اقدام به صورت تصادفی انتخاب شده و آن اقدام پاداش یا جریمه می‌شود. در اثر پاداش دادن یا جریمه کردن یک اقدام، وضعیت آن اقدام در مجموعه وضعیت‌های اقدام مورد نظر، تغییر می‌کند. اگر یک اقدام در وضعیت مرزی قرار داشته باشد، جریمه شدن آن باعث تغییر اقدام آن و در نتیجه باعث ایجاد رنگ آمیزی جدیدی می‌شود. عملگر جریمه و پاداش با توجه به نوع اتوماتای یادگیر متفاوت خواهد بود.

۴-۲-۱- عملگر پاداش

زمانی که به یک اقدام پاداش داده می‌شود، ممکن است آن اقدام در وضعیت داخلی و یا در وضعیت غیرداخلی قرار داشته باشد. در صورتی که اقدام مورد نظر در وضعیت غیرداخلی قرار داشته باشد در این صورت با پاداش دهی به آن اقدام، وضعیت آن به سمت وضعیت داخلی حرکت می‌کند. شکل ۵ نحوه پاداش دهی به اقدامی که در وضعیت غیر داخلی قرار دارد را نمایش می‌دهد.

در صورتی که اقدام مورد نظر در وضعیت داخلی خود قرار داشته باشد، در این صورت با اعمال عمل پاداش هیچ تغییری در وضعیت آن اقدام صورت نمی‌گیرد و در وضعیت فعلی قرار می‌گیرد.

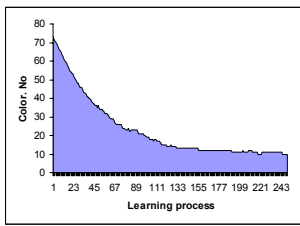


شکل ۵. (الف) وضعیت گره u قبل از پاداش و (ب) وضعیت گره u بعد از پاداش

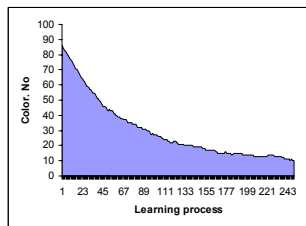
۴-۲-۲- عملگر جریمه

موقعی که یک اقدام مورد عمل جریمه قرار می‌گیرد، دو حالت ممکن است رخ دهد:

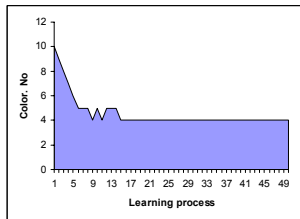
۱. اقدام در وضعیت غیر مرزی قرار داشته باشد
۲. اقدام در وضعیت مرزی قرار داشته باشد.



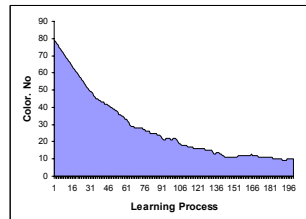
گراف ۲



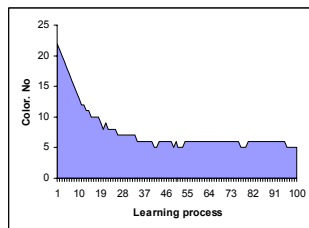
گراف ۱



گراف ۴



گراف ۳



گراف ۵

شکل ۱۰. سیر یادگیری الگوریتم پیشنهادی برای گراف‌های شکل ۸

۶- نتیجه گیری

در این مقاله یک الگوریتم غیر قطعی مبتنی بر اتوماتای یادگیر برای حل مسئله رنگ آمیزی گراف پیشنهاد گردید. این الگوریتم با استفاده از عملگرهای جریمه و پاداش سعی در به حداقل رساندن تعداد رنگ‌های مورد نیاز برای رنگ آمیزی گراف را دارد. در این مقاله از اتوماتای مهاجرت اشیا مبتنی بر اتوماتای تستلین برای یادگیری استفاده شده است. در نهایت الگوریتم پیشنهادی برای گرافهای مطرح با الگوریتم ژنتیک مورد بررسی و ارزیابی قرار گرفته است. نتایج آزمایشات به عمل آمده نحوه اجرای الگوریتم پیشنهادی را نمایش می‌دهد. همان‌طور که در نتایج شبیه‌سازی‌ها قابل مشاهده است الگوریتم پیشنهادی خیلی سریع‌تر از الگوریتم ژنتیک به جواب بهینه نزدیک می‌شود.

مراجع

- [1] K. S. Narendra and M. A. L. Thathachar, "Learning Automata: An introduction", Prentice Hall, 1989.
- [2] Meybodi, M. R. and Beigy, H., "New Class of Learning Automata Based Scheme for Adaptation of Backpropagation Algorithm Parameters", Proc. Of EUFIT-98, Sep. 7-10, Aachen, Germany, pp. 339-344, 1998.
- [3] Oommen, B. J. and Ma, D. C. Y., "Deterministic Learning Automata Solution to the Keyboard Optimization Problem", IEEE Trans. On Computers, Vol. 37, No. 1, pp. 2-3, 1988.
- [4] Beigy, H. and Meybodi, M. R., "Optimization of Topology of neural Networks Using Learning Automata", Proc. Of 3th Annual Int. Computer Society of Iran Computer Conf. CSICC-98, Tehran, Iran, pp. 417-428, 1999.
- [5] Beigy, H. and Meybodi, M. R. "Optimization of Topology of Neural Networks Using Learning Automata", Proc. Of 3th Annual Int.

$$n_i \begin{cases} \text{Reward} & \text{if } (\delta/c \leq c/E) \\ \text{Penalize} & \text{if } (\delta/c > c/E) \end{cases}$$

رابطه ۱: شرط جریمه و پاداش شدن اقدام n_i

۵- نتایج شبیه سازی شده

برای اجرا و آزمایش نتایج حاصل از الگوریتم پیشنهادی یک شبیه‌ساز پیاده‌سازی شده است. این شبیه‌ساز گراف را در قالب یک فایل متنی در یافت کرده و سپس، پس از اجرای الگوریتم پیشنهادی تعداد حداقل رنگ-های مورد نیاز برای رنگ آمیزی گراف ورودی را برمی‌گرداند. در این بخش الگوریتم پیشنهادی با الگوریتم ژنتیک مورد بررسی و مقایسه قرار داده می‌شود. با در نظر گرفتن گراف‌های شماره ۱ تا ۵ در جدول شکل ۸ نتایج حاصل از الگوریتم ژنتیک و الگوریتم پیشنهادی قابل مشاهده است. همان‌طور که مشاهده می‌شود نتایجی را که الگوریتم پیشنهادی تولید می‌کند، نتایج تقریباً بهتری از نتایج الگوریتم ژنتیک هستند.

شماره گراف	تعداد گره	تعداد یال	الگوریتم ژنتیک	الگوریتم پیشنهادی	مرجع گراف
۱	۸۷	۴۰۶	۱۱	۱۰	[8]
۲	۷۴	۳۰۱	۱۱	۱۰	[9]
۳	۸۰	۲۵۴	۱۰	۹	[10]
۴	۱۱	۲۰	۴	۴	[11]
۵	۲۳	۷۱	۵	۵	[12]

شکل ۸. مقایسه الگوریتم ژنتیک و الگوریتم پیشنهادی برای گراف‌های مطرح

پارامترهای الگوریتم پیشنهادی که برای شبیه سازی گراف‌های شکل ۸ بکار رفته شده در جدول شکل ۹ نمایش داده شده‌اند.

شماره گراف	جمعیت اولیه	تکرار	عمق حافظه
۱	۳۰۰	۲۵۰	۵
۲	۲۵۰	۲۵۰	۵
۳	۲۵۰	۲۰۰	۵
۴	۱۰۰	۵۰	۵
۵	۱۰۰	۱۰۰	۵

شکل ۹. پارامترهای الگوریتم پیشنهادی

شکل ۱۰ سیر یادگیری الگوریتم پیشنهادی را برای ۵ گراف نمایش داده شده در شکل ۸ را نمایش می‌دهند. همان‌طور که در این اشکال مشاهده می‌شود، الگوریتم پیشنهادی در تعداد تکرارهای کمتری به جواب بهینه نزدیک می‌شود، یا به عبارتی سرعت همگرایی بالایی دارد.

Computer Society of Iran Computer Conf. CSICC-98, Tehran, Iran, pp. 417-428, 1999.

- [6] Hashim, A.A., Amir, S. and Mars, p. "Application of Learning Automata to Data Compression, In Adaptive and Learning Systems", K. S. Narendra (Ed), New York: Plenum Press, pp. 229-234, 1986.
- [7] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Reading, MA: Addition-Wesley, 1989.
- [8] <http://mat.gsia.cmu.edu/COLOR/instances/david.col>
- [9] <http://mat.gsia.cmu.edu/COLOR/instances/huck.col>
- [10] <http://mat.gsia.cmu.edu/COLOR/instances/jean.col>
- [11] <http://mat.gsia.cmu.edu/COLOR/instances/mycie13.col>
- [12] <http://mat.gsia.cmu.edu/COLOR/instances/mycie14.col>

Archive of SID

لینک های مفید



عضویت
در خبرنامه



کارگاه های
آموزشی



سرویس
ترجمه تخصصی
STRS



فیلم های
آموزشی



بلاگ
مرکز اطلاعات علمی



سرویس های
ویژه