

یک الگوریتم ترکیبی (کلونی مورچه‌ها + اتوماتاهای یادگیر) برای حل مساله درخت اشتاینر ایستا

محمد رضا میبیدی
دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیرکبیر
mmeybodi@aut.ac.ir

سمیرا نوفرستی
دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیرکبیر
Samira_noferesty@yahoo.com

چکیده- در این مقاله یک الگوریتم ترکیبی که از ترکیب مدل کلونی مورچه‌ها و مدل اتوماتاهای یادگیر حاصل شده است برای حل مساله درخت اشتاینر ایستا پیشنهاد می‌گردد. الگوریتم پیشنهادی، الگوریتم کلونی مورچه‌ها برای حل مساله درخت اشتاینر می‌باشد که در آن از اتوماتاهای یادگیر برای تطبیق پارامترهای اهمیت نسبی فرمون روی مسیرها، ضریب تبخیر فرمون و پارامتر انتخاب مسیر استفاده شده است. در الگوریتم پیشنهادی هر گروه از مورچه‌ها به سه اتوماتای یادگیر مجهز شده است که هر اتوماتای یادگیر عهده‌دار تطبیق یکی از پارامترها می‌باشد. هر گروه از مورچه‌ها پارامترهای خاص خود را دارد و مورچه‌های هر گروه برای انتخاب گره بعدی و بهنگام‌سازی فرمون از پارامترهای گروه خود استفاده می‌کنند. نتایج آزمایشهای انجام شده نشان می‌دهد که این الگوریتم در مقایسه با الگوریتم کلونی مورچه‌ها نتایج بهتری را تولید می‌کند.

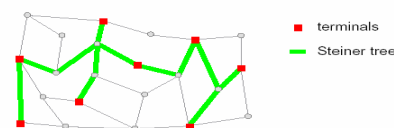
کلیدواژه: اتوماتاهای یادگیر، تطبیق پارامترها، کلونی مورچه‌ها، مساله درخت اشتاینر

۱- مقدمه

روشهایی مانند برنامه‌نویسی پویا و الگوریتمهای برش‌وانشعاب استفاده می‌کنند [۱]. این الگوریتمها به دلیل زمان‌اجرای نامایی مناسب کاربردهای عملی نمی‌باشند. به همین دلیل الگوریتمهای تقریبی متعددی برای حل این مساله گزارش شده است. بسیاری از این الگوریتمها درخت اشتاینر را از طریق ساختن یک درخت پوشای کمینه می‌سازند [۲].

از جمله روشهای تقریبی کارا برای حل مساله اشتاینر الگوریتمهای تکرارشونده^۱ هستند. در این روشها رسیدن به یک پاسخ بهینه تضمین نمی‌شود اما در اغلب موارد جوابهای تقریبی قابل‌قبولی تولید می‌شود. از جمله الگوریتمهای تکرارشونده می‌توان به الگوریتمهای ژنتیکی [۳ و ۴]، الگوریتمهای مبتنی بر اتوماتاهای سلولی [۵] و الگوریتم کلونی مورچه‌ها [۶] اشاره کرد. در این مقاله یک الگوریتم تکرارشونده ترکیبی که از ترکیب مدل کلونی مورچه‌ها و مدل یادگیری اتوماتاهای یادگیر حاصل شده است برای حل مساله درخت اشتاینر پیشنهاد می‌گردد.

مساله درخت اشتاینر یکی از معروفترین مسائل بهینه‌سازی ترکیبی است که کاربردهای فراوانی از قبیل مسیریابی در شبکه‌های کامپیوتری، طراحی مدارهای VLSI، طراحی شبکه‌های پلیس امداد و ایستگاه‌های پستی دارد. مساله درخت اشتاینر عبارت است از یافتن درختی با حداقل هزینه که مجموعه‌ای از نقاط خاص به نام ترمینال را به هم متصل کند. هزینه درخت برابر مجموع هزینه یالهای آن می‌باشد. برای کاهش هزینه درخت از گره‌های دیگری غیر از ترمینالها استفاده می‌شود که این گره‌ها، نقاط اشتاینر نامیده می‌شوند. مثالی از مساله درخت اشتاینر در شکل ۱ آمده است.



شکل ۱: درخت اشتاینر در گراف

مساله درخت اشتاینر یک مساله NP_Complete است. الگوریتمهای قطعی گزارش شده برای حل این مساله از

¹ Iterative

کوتاهتر مقدار فرمون بیشتری خواهند داشت. حرکت مورچه‌های بعدی به صورت تصادفی انجام می‌شود. با این وجود احتمال انتخاب مسیری که مقدار فرمون بیشتری در آن قرار دارد بیشتر است. این عمل به سایر مورچه‌ها کمک می‌کند تا مسیرهای کوتاهتر به سمت منابع غذایی را شناسایی کنند.

در [۶] روشی مبتنی بر کلونی مورچه‌ها برای حل مساله درخت اشتاینر معرفی شده است. راه‌حل ارائه شده به این صورت عمل می‌کند که در هر تکرار از الگوریتم روی هر گره ترمینال یک مورچه قرار داده می‌شود. مورچه‌ها به طور همزمان شروع به حرکت می‌کنند. هرگاه مورچه‌ای به گرهی رسید که قبلاً توسط مورچه دیگری طی شده است، مسیرهای طی شده توسط این دو مورچه ادغام می‌شوند. به این معنا که مورچه جاری حذف و درخت طی شده توسط آن به درخت مورچه دیگر اضافه می‌شود. در نهایت هرگاه تنها یک مورچه باقی ماند درخت حاصل تقریبی از درخت اشتاینر این گراف است. کل این فرآیند به صورت تکراری و به دفعات زیاد انجام می‌شود.

برای انتخاب یک گره توسط یک مورچه به صورت زیر عمل می‌شود. مورچه m که در گره i قرار دارد، گره j را به عنوان گره بعدی انتخاب می‌کند مشروط بر این که $(i,j) \in CE$ و گره j در لیست گره‌های طی شده توسط این مورچه نباشد. برای تسریع ادغام مورچه‌ها، به هر گره j یک پتانسیل مربوط به مورچه m به صورت زیر نسبت داده می‌شود:

$$\psi_j^{(m)} = \min_k \{d(j,k)\} \quad k \in \bigcup_{m' \neq m} T^{m'} \quad (1)$$

که در آن m' یک مورچه غیر از m بوده و $T^{m'}$ نشان دهنده لیست گره‌های طی شده توسط مورچه m و $d(j,k)$ کوتاهترین مسیر از گره j به گره k در گراف می‌باشد. با توجه به تعریف فوق میزان مطلوبیت گره j برای انتخاب توسط مورچه m بنابر رابطه زیر تعریف می‌شود:

$$\eta_{i,j}^{\{m\}} = \frac{1}{c(i,j) + \lambda \psi_j^{\{m\}}} \quad (2)$$

که در آن λ یک پارامتر ثابت و $c(i,j)$ نشان دهنده هزینه یال (i,j) است. اگر غلظت فرمون کمان (i,j) را با τ_{ij} نشان داده شود، احتمال انتخاب گره j توسط مورچه m به صورت زیر به انجام می‌شود:

الگوریتم‌های کلونی مورچه‌ها دارای پارامترهای متعددی می‌باشند. این پارامترها در تصمیم‌گیری مورچه‌ها دخیل هستند به نحوی که همگرایی و کارایی الگوریتم تا حد زیادی وابسته به آنها می‌باشد. معمولاً مقادیر مناسب برای این پارامترها با توجه به نوع مساله از طریق آزمایش و خطا تعیین می‌شود. در [۷] مطالعاتی درباره تأثیر پارامترهای الگوریتم کلونی مورچه‌ها در حل مساله فروشنده دوره‌گرد انجام گرفته است. نتایج به دست آمده نشان می‌دهند که الگوریتم کلونی مورچه‌ها زمانیکه مقادیر پارامترهای آن در بازه‌های خاصی قرار دارند نتایج بهتری تولید می‌کند.

تا آنجا که نگارندگان این مقاله اطلاع دارند دو روش برای برای تطبیق پارامترهای سیستم کلونی مورچه‌ها گزارش شده است. در [۸] نشان داده شده است که استفاده از الگوریتم‌های ژنتیکی برای تطبیق پارامترهای الگوریتم کلونی مورچه‌ها می‌تواند کارایی را به مقدار زیادی بهبود بخشد. در [۹] از اتوماتاهای یادگیر برای تطبیق پارامترهای الگوریتم کلونی مورچه‌ها در حل مساله فروشنده دوره‌گرد استفاده شده است و نشان داده شده است که این روش از کارایی بالاتری برخوردار است.

الگوریتم پیشنهادی در این مقاله، الگوریتم کلونی مورچه‌ها برای حل مساله درخت اشتاینر گزارش شده در [۶] می‌باشد که در آن از اتوماتاهای یادگیر برای تطبیق پارامترها استفاده شده است. نتایج آزمایشات انجام شده نشان می‌دهد که این الگوریتم در مقایسه با الگوریتم کلونی مورچه‌ها نتایج بهتری را تولید می‌کند.

ادامه مقاله بدین صورت سازماندهی شده است. در بخش ۲ به اختصار سیستم کلونی مورچه‌ها و ایده حل مساله اشتاینر با استفاده از این سیستم معرفی می‌شود. در بخش ۳ اتوماتاهای یادگیر و انواع آن بیان می‌شود. در بخش ۴ الگوریتم پیشنهادی و در بخش ۵ نتایج شبیه‌سازی‌های ارائه می‌شود. بخش پایانی مقاله نتیجه‌گیری می‌باشد.

۲- الگوریتم کلونی مورچه‌ها

مورچه‌ها در طبیعت در هنگام یافتن غذا و برگشت به طرف لانه ماده‌ای فرار به نام فرمون روی مسیری که پیمایش می‌کنند قرار می‌دهند. فرمون به جای گذاشته شده بر روی زمین به آرامی تبخیر می‌شود. به همین دلیل مسیرهای

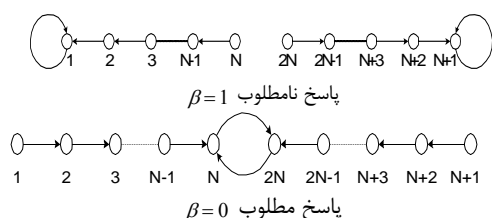
و $\beta_i = 0$ به عنوان پاداش در نظر گرفته می‌شود. c_i احتمال اینکه عمل α_i نتیجه نامطلوب داشته باشد می‌باشد.

اتوماتاهای یادگیر به دو گروه اتوماتای یادگیر با ساختار ثابت^۲ و اتوماتای یادگیر با ساختار متغیر^۳ تقسیم می‌شوند که در ادامه به معرفی آنها می‌پردازیم.

۳-۱- اتوماتاهای یادگیر با ساختار ثابت

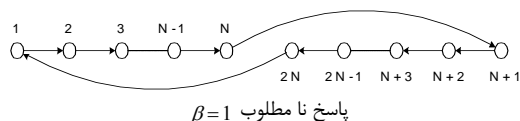
اتوماتای یادگیر با ساختار ثابت توسط ۵ تایی $LA \equiv \{\alpha, \beta, F, G, \phi\}$ نشان داده می‌شود که $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عملها، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودی‌ها، $F \equiv \phi \times \beta \rightarrow \phi$ تابعی که بر اساس پاسخ محیط وضعیت جدید اتوماتای یادگیر را تعیین می‌کند، $G \equiv \phi \rightarrow \alpha$ تابعی که وضعیت کنونی را به خروجی بعدی نگاشت می‌کند و $\phi(n)$ مجموعه وضعیت‌های داخلی اتوماتای یادگیر می‌باشند. در ادامه چند نمونه از اتوماتاهای یادگیر با ساختار ثابت که در این مقاله از آنها استفاده شده است معرفی می‌گردد.

اتوماتای یادگیر $L_{2N,2}$: این اتوماتا تعداد پاداش‌ها و جریمه‌های دریافت شده برای هر عمل را نگهداری کرده و تنها زمانی که تعداد جریمه‌ها بیشتر از پاداش‌ها می‌گردد، عمل دیگر را انتخاب می‌کند. نمودار تغییر وضعیت این اتوماتا مطابق شکل ۲ می‌باشد.



شکل ۲: نمودار تغییر وضعیت اتوماتای یادگیر $L_{2N,2}$

اتوماتای یادگیر $G_{2N,2}$: در این اتوماتای یادگیر عمل α_2 حداقل N بار انجام می‌گردد تا اینکه در نهایت عمل α_1 دوباره انتخاب شود. گراف تغییر وضعیت این اتوماتا برای پاسخ مطلوب مانند اتوماتای $L_{2N,2}$ و برای پاسخ نامطلوب مطابق شکل ۳ می‌باشد.



^۱Fixed Structure

^۲Variable Structure

$$p(i, j) = \frac{[\tau_{i,j}]^\alpha [\eta_{i,j}^{\{m\}}]^\beta}{\sum_{k \in T^{\{m\}}} [\tau_{i,k}]^\alpha [\eta_{i,k}^{\{m\}}]^\beta} \quad (3)$$

که α و β پارامترهای ثابت هستند. هر اتوماتای یادگیر با احتمال $1-q$ گر z را با احتمال $p(j)$ انتخاب و با احتمال q گر z را طبق رابطه زیر انتخاب می‌کند:

$$j = \operatorname{argmax} \{p(i, j)\} \quad (4)$$

دو قاعده بهنگام‌سازی فرمون مطرح است. برای بهنگام‌سازی محلی هرگاه مورچه‌ای از گر i به گر z حرکت کند غلظت فرمون یال (i, z) به کمک رابطه زیر بهنگام می‌شود:

$$\tau_{i,z}^{new} = (1-\rho)\tau_{i,z}^{old} + \rho\delta \quad (5)$$

که در آن ρ و σ پارامترهای ثابت هستند. قاعده بهنگام‌سازی سراسری به صورت زیر تعریف می‌شود:

$$\tau_{i,j}^{new} = \begin{cases} (1-\rho)\tau_{i,j}^{old} + \rho\delta/c & c \leq c_{\min} \\ (1-\rho)\tau_{i,j}^{old} & c > c_{\min} \end{cases} \quad \forall i, j \in E_S \quad (6)$$

که در آن S درخت بدست آمده از این مرحله با هزینه E_S ، c_{\min} هزینه بهترین درخت بدست آمده تا قبل از این مرحله، ρ_1 پارامتر ثابت تبخیر سراسری و σ_1 میزان اصلاح سراسری است. این قاعده به این معناست که اگر هزینه درخت جدید بیشتر از c_{\min} باشد یک تبخیر سراسری بر روی یالهای آن انجام می‌شود که به معنای جریمه است. اما اگر درخت جدید هزینه‌ای کمتر یا مساوی c_{\min} داشته باشد تشویق می‌شود. در این مقاله برای تطبیق مقادیر پارامترهای α ، δ و q از اتوماتاهای یادگیر استفاده شده است.

۳- اتوماتاهای یادگیر

اتوماتای یادگیر [۱۰] یک مدل انتزاعی است که می‌تواند تعداد محدودی عمل را انجام دهد. هر عمل انتخاب شده توسط محیطی تصادفی ارزیابی می‌گردد و پاسخی به اتوماتای یادگیر داده می‌شود. اتوماتای یادگیر با استفاده از این پاسخ عمل خود را برای مرحله بعد انتخاب می‌کند.

محیط: محیط تصادفی را می‌توان توسط سه‌تایی $E \equiv \{\alpha, \beta, c\}$ تعریف نمود که $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودی‌ها، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجی‌ها و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمال‌های جریمه شدن می‌باشند. هرگاه β_i دو مقداری باشد $\beta_i = 1$ به عنوان جریمه

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

ب- پاسخ نامطلوب

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

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

که پارامتر پاداش و b پارامتر جریمه می‌باشد. با توجه به مقادیر a و b سه حالت مختلف را می‌توان در نظر گرفت. زمانیکه a و b با هم برابر باشند، الگوریتم را L_{RP} ^۴، زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را L_{REp} ^۵ و زمانیکه b مساوی صفر باشد، الگوریتم را L_{RI} ^۶ می‌نامند.

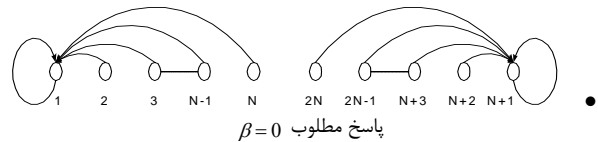
۴- الگوریتم پیشنهادی

در این بخش یک الگوریتم ترکیبی که از ترکیب مدل کلونی مورچه‌ها و مدل یادگیری اتوماتاهای یادگیر حاصل شده است برای حل مساله درخت اشتاینر پیشنهاد می‌گردد. الگوریتم پیشنهادی، الگوریتم کلونی مورچه‌ها برای حل مساله درخت اشتاینر گزارش شده در [۶] می‌باشد که در آن از اتوماتاهای یادگیر برای تطبیق پارامترهای α ، β ، ρ و q استفاده شده است که α اهمیت نسبی میدان دید، β اهمیت نسبی فرمون، ρ ضریب تبخیر فرمون و q فرمول انتخاب مسیر می‌باشد. این پارامترها در تصمیم‌گیری مورچه‌ها و در نتیجه در کارایی الگوریتم نقش بسزایی دارند.

در الگوریتم پیشنهادی گروههایی از مورچه‌ها که در هر گروه به تعداد ترمینالهای گراف مورچه قرار دارد، وجود دارد. در هر تکرار به ترتیب یک گروه از مورچه‌ها وارد عمل می‌شود. بعد از این که همه گروهها یک بار بررسی شدند مجدداً تکرار الگوریتم با گروه اول انجام می‌شود. هر گروه از مورچه‌ها به سه اتوماتای یادگیر مجهز شده است. هر اتوماتا برای تعیین مقدار یکی از پارامترهای تصمیم‌گیری به کار می‌رود. در واقع هر گروه از مورچه‌ها پارامترهای خاص خود را دارد و مورچه‌های هر گروه برای انتخاب گره بعدی و بهنگام‌سازی فرمون از پارامترهای گروه خود استفاده می‌کنند. مجموعه مقادیر مجاز هر پارامتر بصورت زیر تعریف شده است:

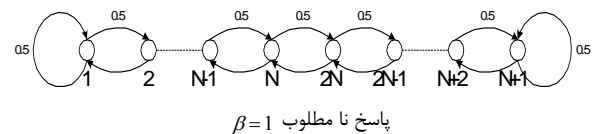
شکل ۳: نمودار تغییر وضعیت اتوماتای یادگیر $G_{2N,2}$

• **اتوماتای یادگیر Krinsky:** این اتوماتای یادگیر زمانی که پاسخ محیط نامطلوب است، مانند اتوماتای $L_{2N,2}$ رفتار می‌کند. اما برای پاسخ مطلوب مطابق شکل ۴ عمل می‌کند. بنابراین همیشه N پاسخ نامطلوب متوالی لازم است تا اتوماتای یادگیر عمل خود را تغییر دهد.



شکل ۴: نمودار تغییر وضعیت اتوماتای یادگیر Krinsky

• **اتوماتای یادگیر Krylov:** در این اتوماتای یادگیر زمانیکه پاسخ محیط مطلوب است، تغییر وضعیت مانند اتوماتای $L_{2N,2}$ می‌باشد. اما زمانیکه پاسخ محیط نامطلوب می‌باشد، هر وضعیت $(i \neq 1, N, N+1, 2N)$ با احتمال 0.5 به وضعیت ϕ_{i+1} و با احتمال 0.5 به وضعیت ϕ_{i-1} مطابق شکل ۵ منتقل می‌شود:



شکل ۵: نمودار تغییر وضعیت اتوماتای Krylov

۳-۲- اتوماتاهای یادگیر با ساختار متغیر

اتوماتای یادگیر با ساختار متغیر توسط δ تایی $LA \equiv \{\alpha, \beta, p, T, c\}$ نشان داده می‌شود که $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عمل‌های اتوماتای یادگیر، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودی‌های اتوماتای یادگیر، $p \equiv \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب عملها، $T \equiv p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری و $c \equiv \{c_1, c_2, \dots, c_r\}$ جریمه اعمال می‌باشند. اگر در اتوماتای یادگیر عمل α_i در مرحله n ام انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالها کاهش می‌یابند و برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته و سایر احتمالها افزایش می‌یابند. تغییرات به گونه‌ای صورت می‌گیرد تا حاصل جمع $p_i(n)$ ها همواره ثابت و مساوی یک باقی بماند.

الف- پاسخ مطلوب

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

^۴Linear Reward Penalty

^۵Linear Reward Epsilon Penalty

^۶Linear Reward Inaction

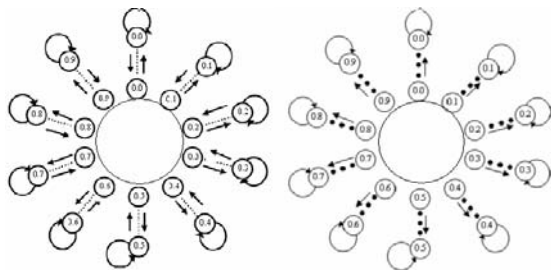
۱۴/۱۹	۴/۶۵	۰/۷۰	۰	۰	۰/۷۰	۰	۸۶	۱۰
۷/۰۴۴	۱/۹۵۹	۲۷۶	۰/۱۹۷	۰/۴۲۵	۰/۵۳۲	۰/۲۱۵		متوسط خطا

ستون اول جدول شماره گراف در مجموعه B مسائل بیسلی، ستون دوم هزینه درخت اشتاینر بهینه برای هر گراف و ستونهای بعدی درصد خطای نسبی حاصل از ۱۰ بار اجرای الگوریتم پیشنهادی با اتوماتاهای یادگیر مختلف را نشان می‌دهد. برای محاسبه خطای نسبی از رابطه زیر استفاده می‌شود:

$$relative - error = \frac{T_* - T_{opt}}{T_{opt}} \quad (11)$$

که T_* هزینه درخت حاصل از اجرای الگوریتم و T_{opt} هزینه درخت اشتاینر بهینه می‌باشد در شبیه‌سازی‌های انجام شده تعداد گروههای مورچه‌ها برابر ۱۰ و حداکثر عمق مجاز برای اتوماتاهای یادگیر با ساختار ثابت ۵ در نظر گرفته شده است. در مورد اتوماتای یادگیر با ساختار متغیر نیز احتمال اولیه عملها یکسان در نظر گرفته شده است. مقدار پارامتر پاداش در اتوماتاهای L_{RI} ، L_{RP} و L_{REP} برابر ۰/۳ و مقدار پارامتر جریمه در L_{REP} برابر ۰/۰۱ و تعداد تکرار الگوریتم ۳۰۰ در نظر گرفته شده است. همان طور که مشخص است اتوماتای یادگیر krylov نتایج بهتری را در مقایسه با سایر اتوماتاهای یادگیر تولید کرده است.

با توجه به نتایج بدست آمده برای تطبیق پارامترهای الگوریتم کلونی مورچه‌ها از اتوماتای یادگیر krylov استفاده شده است. شکل ۶ اتوماتای یادگیر تطبیق پارامتر ρ را نشان می‌دهد. اتوماتاهای یادگیر برای تطبیق پارامترهای دیگر مشابه این اتوماتا می‌باشند.



ب: پاسخ نامطلوب

الف: پاسخ مطلوب

شکل ۶: تغییر وضعیت اتوماتای یادگیر krylov برای تطبیق پارامتر ρ

در جدول ۲ درصد خطای نسبی الگوریتم ACSLA به همراه درصد خطای نسبی الگوریتم ACS و درصد خطای نسبی تعدادی از الگوریتمهای حل مساله اشتاینر که در [۴]

$$\alpha \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$$

$$\rho \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$$

$$Q_0 \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$$

از آنجا که پارامترهای β و α به همراه هم اهمیت نسبی فرمون در مقابل فاصله را نشان می‌دهند، مقدار β از فرمول $\beta = 0.5 - \alpha$ تعیین می‌شود. مقداری که یک عمل از یک اتوماتای یادگیر اختیار می‌کند یکی از مقادیر مجاز پارامتر متناظر با این اتوماتای یادگیر می‌باشد.

در شروع الگوریتم، اعمال انتخاب شده توسط اتوماتاهای یادگیر (پارامترهای مورچه‌ها) و عمق اعمال انتخاب شده (برای اتوماتاهای با ساختار ثابت) برای هر گروه از مورچه‌ها بصورت تصادفی انتخاب می‌شود. در هر تکرار با توجه به درخت بدست آمده مقادیر پارامترها بهنگام می‌شود. اگر هزینه درخت بدست آمده از این تکرار از هزینه بهترین درخت بدست آمده تا این مرحله کمتر بود هر سه اتوماتای یادگیر آن گروه پاداش می‌گیرند و در غیر این صورت هر سه اتوماتا جریمه می‌شوند. ارتباط بین گروههای مورچه‌ها به صورت غیر مستقیم و از طریق مقدار فرمونی که بر روی بالهای پیمایش شده بر جای می‌گذارند انجام می‌گیرد.

۵- نتایج شبیه‌سازی

به منظور نشان دادن کارایی الگوریتم پیشنهادی (ACSLA) از گرافهای مجموعه B از دسته مسائل بیسلی^۷ [۱۱] استفاده شده است. در آزمایش اول برای تطبیق پارامترهای الگوریتم ACS از اتوماتاهای یادگیر مختلف استفاده شده است. نتایج بدست آمده در جدول ۱ نشان داده شده است.

جدول ۱: مقایسه نتایج حاصل از الگوریتم ACSLA با استفاده از

اتوماتاهای یادگیر مختلف

شماره گراف	هزینه بهینه	L	G	Krinsky	Krylov	LRP	LR _{RI}	L _I
۱	۸۲	۰	۰	۰	۰	۰	۰	۰
۲	۸۳	۰	۰	۰	۰	۱/۶۹	۰	۳/۸۵
۳	۱۳۸	۰	۰	۰	۰	۰	۰	۸/۸۴
۴	۵۹	۰	۰	۰	۰	۰	۱/۳۵	۱۱/۱۸
۵	۶۱	۰	۰	۰	۰	۰	۰	۵/۳۴
۶	۱۲۲	۰/۹۷	۱/۶۳	۱/۳۱	۱/۲۳	۰/۸۲	۰/۸۲	۵/۵۷
۷	۱۱۱	۰/۳۶	۰	۰	۰	۰	۰/۱۸	۱۸/۰۱
۸	۱۰۴	۰	۱/۳۵	۲/۱۲	۰/۱۹	۰	۲/۵	۰/۳۸
۹	۲۲۰	۰/۸۲	۱/۶۴	۱/۰۹	۰/۵۵	۰/۵۵	۰/۰۹	۳/۱۸

⁷Beasley

پیشنهادی می‌تواند برای گرافهای پویا نیز استفاده شود که در گزارش دیگری به آن پرداخته خواهد شد.

مراجع

- [1] Koch, T., and Martin, A. Solving Steiner Tree Problems in Graphs to Optimality. *Networks*, Vol. 32, 207-232, 1998.
- [2] Diane, M., and Plesnik, J. Three New Heuristics for the Steiner Problem in Graphs. *Acta Math.*, Vol. LX, 105-121, 1999.
- [3] Esbensen, H. Computing Near-Optimal Solutions to the Steiner Problem in a Graph Using a Genetic Algorithm. *Networks*, 26 (1995), 173-185.
- [4] Ding, S., and Ishii, N. An Online Genetic Algorithm for Dynamic Steiner Tree Problem. *Symposium on Computational Geometry*, 337-343, 1995.
- [5] Adamatzky, A. Computing in Nonlinear Media and Automata Collectives. *IOP Publishing*, ISBN 0 7503 0751 X, 2001.
- [6] م. تشکری هاشمی، پ. ادیبی، ع. جهانیان، و ع. نوراله، حل مساله درخت اشتاینر پویا به کمک سیستم کلونی مورچه‌ها، در مجموعه مقالات نهمین کنفرانس سالانه انجمن کامپیوتر ایران، ۱۳۸۲.
- [7] Coloni, A., Dorigo, M., and Maniezzo, V. An Investigation of some Property of an Ant Colony. *Parallel Problem Solving from Nature*, No. 2, Elsevier Science Publication, 1992.
- [8] Marcin, L. P., and Tony, W. Using Genetic Algorithms to Optimize ACS-TSP. <http://citeseer.nj.nec.com/>, AT: march 2005.

[9] ف. ابدالی، م. ر. میبیدی، تطبیق پارامترهای الگوریتم کلونی مورچه‌ها با استفاده از اتوماتاهای یادگیر، در مجموعه مقالات دهمین کنفرانس سالانه انجمن کامپیوتر ایران، مرکز تحقیقات مخابرات، تهران، ایران، ۱۳۸۴.

- [10] Sutton, R.S., Barto, A.G. Reinforcement Learning: An introduction, Cambridge. *MIT Press*, 1998.
- [11] Beasley, J. E. OR-Library: Distributing Test Problems by Electronic Mail. *Operational Research. SOC.* Vol. 41, No. 11, 1096-1072, 1990.
- [12] Rayward-smith, V. J., and Clare, A. On Finding Steiner vertices. *Networks*, Vol. 16, 283-294, 1986.

معرفی شده‌اند، نشان داده شده است. جزئیات الگوریتمهای ADH^۸ و SPH^۹ در مرجع [۱۲] آورده شده است. این نتایج حاصل میانگین ۱۰ بار اجرای الگوریتم با ۵۰۰ تکرار در هر اجراست. همان طور که مشاهده می‌شود در اکثر موارد الگوریتم ACSLA به نتایج بهتری رسیده است.

جدول ۲: مقایسه الگوریتم ACSLA و تعدادی از الگوریتمهای گزارش شده برای حل مساله اشتاینر

شماره گراف	هزینه بهینه	SPH	ADH	ACS	ASCLA
۱	۸۲
۲	۸۳
۳	۱۲۸
۴	۵۹	۵/۰۸	۵/۰۸	۲/۷	.
۵	۶۱
۶	۱۲۲	۳/۲۸	۱/۶۴	۱/۸	۱/۱۴
۷	۱۱۱
۸	۱۰۴
۹	۲۲۰	.	.	۰/۰۵	۰/۸۱
۱۰	۸۶	۴/۶۵	۴/۶۵	۳/۸۴	۰/۷۰
۱۱	۸۸	۲/۲۷	۲/۲۷	۱/۵۹	۱/۳۶
۱۲	۱۷۴	.	.	۰/۱۱	.
۱۳	۱۶۵	۷/۸۸	۲/۲۴	.	۱/۶۳
۱۴	۲۳۵	۲/۵۵	۰/۴۳	۰/۰۹	۰/۰۹
۱۵	۳۱۸
۱۶	۱۲۷	۳/۱۵	.	۰/۱۶	۰/۱۶
۱۷	۱۳۱	۳/۸۲	۳/۰۵	۱/۲۲	۱/۰۶
۱۸	۲۱۸	۱/۸۳	.	۱/۴۲	۰/۰۹

۶- نتیجه گیری

در این مقاله از ترکیب مدل کلونی مورچه‌ها و مدل اتوماتاهای یادگیر الگوریتمی برای حل مساله درخت اشتاینر ایستای بهینه پیشنهاد گردید. به منظور نشان دادن کارایی الگوریتم پیشنهادی از گرافهای مجموعه B از دسته مسائل بیسلی استفاده شده است. نتایج آزمایشهای انجام شده نشان داد که الگوریتم پیشنهادی در مقایسه با الگوریتم کلونی مورچه‌ها از کارایی بالاتری برخوردار است. یکی از ویژگی‌های مهم الگوریتم پیشنهادی این است که به دلیل تطبیق مقادیر پارامترهای آن، کیفیت نتایج تولید شده مستقل از گراف مساله می‌باشد. بدلیل این ویژگی، الگوریتم

⁸ Average Distance Heuristic

⁹ Shortest Path Heuristic