مدل كردن الگوريتم‌هاي كلوني مورچه‌ها با استفاده ازاتوماتاهاي يادگير

فردين ابدالي محمدي[*]   محمد رضا ميبدي[†]

چكيده

الگوريتم‌هاي كلوني مورچه‌ها[1] گروهي از الگوريتم‌هاي بهينه‌سازي مي‌باشند كه از كاوش مورچه‌ها براي غذا در طبيعت الهام گرفته‌اند. در اين الگوريتم‌ها تعدادي عامل كه همان مورچه‌ها مي‌باشند به منظور يافتن راه حل مناسب فضاي راه حل‌ها را جستجو مي‌كنند. از طرف ديگر اتوماتاي يادگير يك مدل انتزاعي است كه تعداد محدودي عمل را مي‌تواند انجام دهد. هر عمل انتخاب شده توسط محيطي احتمالي ارزيابي مي‌گردد و پاسخي به اتوماتاي يادگير داده مي‌شود. اتوماتاي يادگير از اين پاسخ استفاده نموده و عمل خود را براي مرحله بعد انتخاب مي‌كند و از اين طريق كارايي خود را بهبود مي‌بخشد. در اين مقاله در ابتدا نشان داده مي‌شود كه الگوريتم‌هاي كلوني مورچه‌ها را مي‌توان با مجموعه‌اي از اتوماتاهاي يادگير كه با يكديگر در تعامل هستند مدل كرد و سپس با استفاده از مجموعه‌اي از اتوماتاهاي يادگير الگوريتمي براي حل مسالة مسيريابي در شبكه‌هاي كامپيوتري ارايه مي‌گردد. از طريق شبيه‌سازي كامپيوتري كارايي الگوريتم ارايه شده مورد بررسي قرار مي‌گيرد.

كلمات كليدي: كلوني مورچه‌ها، اتوماتاهاي يادگير، مدلسازي، مسيريابي در شبكه‌هاي كامپيوتري

Modeling Ant Colony Algorithms Using Learning Automata

Fardin Abdali Mohammadi   Mohammad Reza Meybodi

Abstract

Ant colony algorithms are a group of heuristic optimization algorithms that have been inspired by ants foraging for food. In these algorithms there are some agents, the ants, that for finding the suitable solution, search the solution space. On the other hand, Learning Automata is an abstract model that can do finite actions. Each selected action is evaluated by a random environment and the environment makes a response back to Learning Automata. Using this response, Learning Automata select its action for the next step and improves its efficiency. In this paper we show that ant colony algorithms can be modeled by a group of cooperating Learning Automata and then using a group of set of cooperating learning automata an algorithm for solving the routing problem in computer networks has been proposed. Computer simulations have been conducted to show the efficiency of the proposed method.

Keywords: **Ant Colony, Learning Automata, Modeling, Routing in Computer Networks.**

۱- مقدمه

مدل اتوماتاهاي يادگير كه قادر به يادگيري در محيطهاي تصادفي هستند اولين بار توسط ستلين[2] جهت مدل كردن رفتار سيستم‌هاي بيولوژيك مطرح شد. اتوماتاهاي يادگيري در مواردي استفاده مي‌شوند كه تهية اطلاعات در باره محيط هزينه‌بر و يا اندازه‌گيري آنها غير ممكن باشد. اتوماتاي يادگير يك مدل انتزاعي است كه تعداد محدودي عمل را مي‌تواند انجام دهد. هر عمل انتخاب شده توسط محيطي احتمالي ارزيابي مي‌گردد و پاسخي به اتوماتاي يادگير داده مي‌شود. اتوماتاي يادگير از اين پاسخ استفاده نموده و عمل خود را براي مرحله بعد انتخاب مي‌كند و از اين طريق كارايي خود را بهبود مي‌بخشد[۲].

---

[*] عضو هيات علمي، دانشكده مهندسي كامپيوتر ، دانشگاه لرستان، ايران

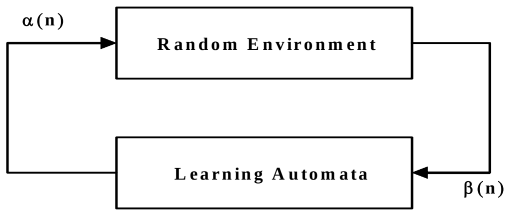
[†] عضو هيات علمي، دانشكده مهندسي كامپيوتر و فناوري اطلاعات، دانشگاه صنعتي اميركبير، تهران، ايران

مورچه‌ها در طبیعت در هنگام یافتن غذا و برگشت به طرف لانه مقداري فرمون[۳] روي مسير قرار مي‌دهند. اين عمل به ساير مورچه‌ها كمك مي‌كند تا منابع غذايي را شناسايي كنند. با توجه به اينكه فرمون قرار داده شده روي زمين به آرامي تبخير مي‌شود فرمون مسير بايد همواره بهنگام شود. لذا با گذشت زمان فرمون روي مسيرهاي طولاني كمتر شده و فرمون مسيرهاي كوتاه‌تر بيشتر مي‌شود. اين عوامل باعث فراموش شدن مسيرهاي طولاني و تقويت مسيرهاي كوتاهتر مي‌شود. مسايل كلوني مورچه‌ها معمولا توسط یک گراف مدل مي‌شود. مسيرها توسط يالهاي گراف و مقدار فرمون هر يال وزن آن يال مي‌باشد. هر مورچه سعي مي‌كند با شروع از یک گره آغازين از طريق پيمايش گراف كوتاهترين دور را در گراف پيدا كند. مورچه‌ها در هر گره بر اساس یک احتمال (احتمال انتخاب مسير) " گرهٔ بعدي" را انتخاب كرده و بر اساس فرمول "بهنگام‌سازي فرمون" باعث تغيير مقدر فرمون يال‌هاي مسير خود مي‌شوند. با اين مكانيزم هر مورچه سعي مي‌كند كوتاهترين دور را در گراف پيدا كند[۱]. در اين مقاله در ابتدا نشان داده مي‌شود كه الگوريتم‌هاي كلوني مورچه‌ها را مي‌توان با مجموعه‌اي از اتوماتاهاي يادگير كه با يكديگر در تعامل هستند مدل كرد. سپس با استفاده از مجموعه‌اي از اتوماتاهاي يادگير الگوريتمي براي حل مسالهٔ مسيريابي در شبكه‌هاي كامپيوتري ارايه مي‌گردد. از طريق شبيه-سازي كامپيوتري كارايي الگوريتم ارايه شده مورد بررسي قرار مي‌گيرد.

در ادامه اين مقاله در بخش ۲ ابتدا به معرفي اجمالي اتوماتاهاي يادگير مي‌پردازيم. در بخش ۳ كلوني مورچه‌ها به طور مختصر شرح داده ميشود. در بخش ٤ روش مدل كردن كلوني مورچه‌ها توسط مجموعه اي از اتوماتاهاي يادگير مورد بررسي قرار مي‌گيرد. در بخش ٥ با استفاده هر از دو رويكرد اتوماتاهاي يادگير و كلوني مورچه‌ها به حل مسالهٔ مسيريابي در شبكه‌هاي كامپيوتري ميپردازيم. بخش نهايي مقاله نتيجه گيري مي‌باشد.

۲- اتوماتاهاي يادگير[٤]

اتوماتاي يادگير يك ماشين با حالات محدود است[٥] كه ميتواند تعدادي محدود عمل را انجام دهد. هر عمل انتخاب شده توسط محيطي احتمالي ارزيابي مي‌گردد و پاسخي به اتوماتاي يادگير داده مي‌شود. اتوماتاي يادگير از اين پاسخ استفاده مي‌نمايد وعمل خود را براي مرحله بعد انتخاب مي كند.. در طي اين فرايند، اتوماتاي يادگير ياد مي‌گيرد كه چگونه بهترين عمل را انتخاب نمايد. شكل (۱) ارتباط بين اتوماتاي يادگير و محيط را نشان مي دهد. محيط[٦] را مي توان توسط سه تايي $E \equiv \{\alpha, \beta, c\}$ نشان داد كه در آن $\alpha = \{\alpha_1, ..., \alpha_r\}$ مجموعه وروديها، $\beta = \{\beta_1, ..., \beta_m\}$ مجموعه خروجيها و $c = \{c_1, ..., c_r\}$ مجموعه احتمالهاي جريمه مي باشد. هرگاه β مجموعه دو عضوي باشد محيط از نوع P مي باشد. در چنين محيطي $\beta_1 = 1$ به عنوان جريمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته مي شود. در محيط Q، $\beta(n)$ مي تواند به طور گسسته يک مقدار از مقادير محدود در فاصله $[0,1]$ و در محيط از نوع S، $\beta(n)$ هر مقدار در فاصله $[0,1]$ را اختيار كند. c_i احتمال اينكه عمل α_i نتيجه نا مطلوب داشته باشد، مي باشد. در محيط ايستا مقادير c_i بدون تغيير مي مانند، حال آنكه در محيط غير ايستا اين مقادير در طي زمان تغيير مي كنند.



شكل (۱): ارتباط بين اتوماتاي يادگير و محيط

اتوماتاهاي يادگير به دو گروه با ساختار ثابت و با ساختار متغير تقسيم بندي ميگردند. در ادامه به شرح مختصري درباره اتوماتاهاي يادگير با ساختار متغيير[۷] و اتوماتاهاي يادگير با ساختار ثابت[۸] مي پردازيم.

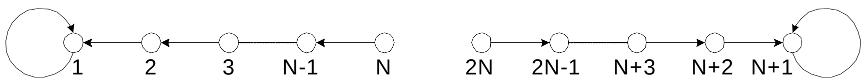
اتوماتاي يادگير با ساختار ثابت: اتوماتاي يادگير با ساختار ثابت توسط ۵ تايي $\{\alpha, \beta, F, G, \phi\}$ نشان داده مي‌شود كه $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه اعمال اتوماتا، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه ورودي‌هاي اتوماتا، $F : \phi \times \beta \rightarrow \phi$ تابعي كه بر اساس پاسخ محيط، وضعيت جديد را مي‌يابد، $G : \phi \rightarrow \alpha$ تابع خروجي كه وضعيت كنوني را به خروجي بعدي مي‌نگارد و $\phi(n) \equiv (\phi_1, \phi_2, ..., \phi_k\}$ مجموعه وضعيت‌هاي داخلي اتوماتاي يادگير مي‌باشد. چند نمونه از اتوماتاهاي يادگير با ساختار ثابت كه در اين مقاله از آنها استفاده كرده‌ايم در زير معرفي مي‌گردد:

اتوماتاي يادگير $L_{2N,2}$: اين اتوماتا تعداد پاداش‌ها و جريمه‌هاي دريافت شده براي هر عمل را نگهداري كرده و تنها زماني كه تعداد جريمه‌ها بيشتر از پاداش‌ها مي‌گردد، عمل ديگر را انتخاب مي‌كند. نمودار تغيير وضعيت اين اتوماتاي يادگير مطابق شكل (۲) مي‌باشد.

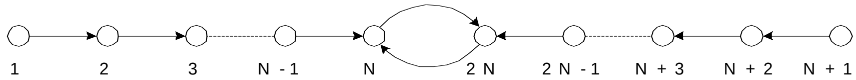
اتوماتاي يادگير $G_{2N,2}$: در اين اتوماتاي يادگير بر خلاف $L_{2N,2}$، عمل α_2 حداقل N بار انجام مي‌گردد (پس از گرفتن N جريمـه) تـا اينكـه در نهايت عمل α_1 دوباره انتخاب شود. گراف تغيير وضعيت اين اتوماتاي يادگير براي پاسخ مطلوب مانند اتوماتاي يادگير $L_{2N,2}$ بـوده و بـراي پاسـخ نامطلوب مطابق شكل (۳) مي باشد.

اتوماتاي يادگير ***Krinsky***: اين اتوماتاي يادگير زماني كه پاسخ محيط نامطلوب است، مانند اتوماتاي يادگير $L_{2N,2}$ رفتار مي‌كند. اما براي پاسـخ مطلوب هر وضعيت $\phi_i (i = 1,2,3,..., N)$ بـه وضـعيت ϕ_1 و هـر وضـعيت $\phi_i (i = N+1, N+2,...,2N)$ بـه وضـعيت ϕ_{N+1} مـي رود. بنابراين هميشه N پاسخ نامطلوب متوالي لازم است تا اتوماتا يادگير عمل خود را تغيير دهد. نمودار تغيير وضعيت اين اتوماتاي يادگير بـراي پاسـخ نامطلوب مانند اتوماتاي يادگير $L_{2N,2}$ بوده و براي پاسخ مطلوب مطابق شكل (٤) مي باشد.

اتوماتاي يادگير ***Krylov***: در اين اتوماتاي يادگير زمانيكه پاسخ محيط مطلوب است، تغيير وضعيت مانـنـد اتوماتـاي يـادگير $L_{2N,2}$ ميباشـد. امـا زمانيكه پاسخ محيط نامطلوب مي‌باشد، هر وضعيت $\phi_i (i \neq 1, N, N+1, 2N)$ با احتمال 0.5 به وضعيت ϕ_{i+1} و با احتمال 0.5 به وضعيت ϕ_{i-1} مطابق شكل (۵) منتقل مي شود.

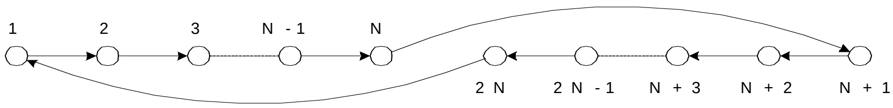


الف- پاسخ مطلوب $\beta = 0$



ب- پاسخ نا مطلوب $\beta = 1$

شكل (۲): نمودار تغيير وضعيت اتوماتاي يادگير $L_{2N,2}$

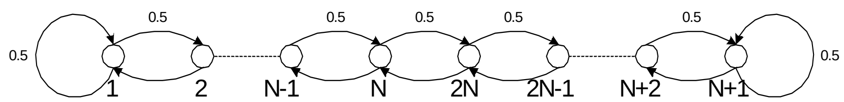


پاسخ نا مطلوب $\beta = 1$

شكل (۳): نمودار تغيير وضعيت اتوماتاي يادگير $G_{2N,2}$

پاسخ مطلوب $\beta = 0$

شكل (٤): نمودار تغيير وضعيت اتوماتاي يادگير***Krinsky***



پاسخ نا مطلوب $\beta = 1$

شكل (۵): نمودار تغيير وضعيت اتوماتاي يادگير ***Krylov***

اتوماتاي يادگير با ساختار متغير: اتوماتاي يادگير با ساختار متغير توسط ٤ تائي $\{\alpha, \beta, p, T\}$ نشان داده مي شود كه در آن $\alpha \equiv \{\alpha_1, \alpha_2, \cdots, \alpha_r\}$ مجموعه عمل هاي اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \cdots, \beta_m\}$ مجموعه وروديهاي اتوماتا، $p \equiv \{p_1, p_2, \cdots, p_r\}$ بردار احتمال انتخاب هر يك از عملها، و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوريتم يادگيري مي باشد. در اين نوع از اتوماتاها، اگر عمل α_i در مرحله n ام انتخاب شود و پاسخ مطلوب از محيط دريافت نمايد، احتمال $p_i(n)$ افزايش يافته و ساير احتمالها كاهش مي‌يابند. و براي پاسخ نامطلوب احتمال $p_i(n)$ كاهش يافته و ساير احتمالها افزايش مي‌يابند. در هر حال، تغييرات به گونه‌اي صورت مي گيرد تا حاصل جمع $p_i(n)$ ها همواره ثابت و مساوي يك باقي بماند. الگوريتم زير يك نمونه از الگوريتمهاي يادگيري خطي براي اتوماتاي يادگير با ساختار متغيير است.

الف- پاسخ مطلوب

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \qquad (۱)$$
$$p_j(n+1) = (1-a)\,p_j(n) \qquad j \neq i \quad \forall j$$

ب- پاسخ نامطلوب

$$p_i(n+1) = (1-b)\,p_i(n) \qquad (۲)$$
$$p_j(n+1) = \frac{b}{r-1} + (1-b)\,p_j(n) \qquad j \neq i \ \forall j$$

در روابط فوق، a پارامتر پاداش و b پارامتر جريمه مي باشد. با توجه به مقادير a و b سه حالت را مي توان در نظر گرفت. زمانيكه a و b با هم برابر باشند، الگوريتم را L_{RP} [۹] مي ناميم. زمانيكه b از a خيلي كوچكتر باشد، الگوريتم را $L_{R\varepsilon P}$ [۱۰] مي ناميم. زمانيكه b مساوي صفر باشد، الگوريتم را L_{RI} [11] مي ناميم. براي اطلاعات بيشتر در باره اتوماتاهاي يادگير با ساختار متغير و اتوماتاهاي يادگير با ساختار ثابت مانند $L_{2N,2}$ ، $G_{2N,2}$ ، كرينسكي[۱۲] و كرايلو[۱۳] كه در اين مقاله از آنها استفاده شده است مي توان به [2,9,10,26,27] مراجعه كرد. اتوماتاهاي يادگير داراي كاربردهاي فراواني از جمله كنترل ترافيک در شبكه هاي مخابراتي[2,11]، مهندسي شبكه هاي عصبي[12-19,24,28]، حل مسايل مشكل[20-24] و فشره سازي اطلاعات[18] مي‌باشد.

۳- الگوريتم‌هاي كلوني مورچه‌ها

در اين بخش ويژگيهاي عمومي الگوريتمهاي كلوني مورچه‌ها شرح داده مي‌شود. مورچه‌هاي طبيعي در هنگام يافتن غذا و برگشت به طرف لانه مقداري فرمون روي مسير قرار مي‌دهند. اين عمل به ساير مورچه‌ها كمك مي‌كند تا مكان منابع غذايي را شناسايي كنند. با گذشت زمان فرمون روي مسيرهاي طولاني كمتر شده و فرمون مسيرهاي كوتاه‌تر بيشتر مي‌شود كه اين باعث فراموش شدن مسيرهاي طولاني و تقويت مسيرهاي كوتاهتر مي‌شود. الگوريتم‌هاي كلوني مورچه‌ها از ٤ مولفه تشكيل شده است كه عبارتند از: گراف مسالة، مورچه‌ها، مكانيزم انتخاب گره بعدي و مكانيزم بهنگام‌سازي فرمون. در ادامه اين بخش اين مولفه ها شرح داده مي‌شود.

گراف مساله: مسايل كلوني مورچه‌ها معمولا توسط یک گراف مدل مي‌شوند. مسير حركت مورچه‌ها توسط يالهاي گراف و مقدار فرمون هر يال وزن آن يال مي‌باشد. هر مورچه سعي مي‌كند با شروع از یک گره آغازين از طريق پيمايش گراف، كوتاهترين دور را در گراف پيدا كند. مورچه‌ها در هر گره بر اساس یک احتمال (احتمال انتخاب مسير) "گرة بعدي" را انتخاب كرده و بر اساس فرمول "بهنگام‌سازي فرمون" باعث تغيير فرمون يال-هاي مسير خود مي‌شوند. با اين مكانيزم هر مورچه سعي مي‌كند كوتاهترين دور گراف را پيدا كند. در الگوريتم‌هاي كلوني مورچه‌ها هر مورچه در ابتداي الگوريتم بصورت تصادفي در يك گره قرار داده مي‌شود. هر مورچه داراي يك حافظه است كه راه‌حل‌هاي جزيي كه تا به حال بدست آمده‌اند را در خود نگه مي‌دارد (حافظه در ابتدا حاوي گره آغازين است). با شروع از گره آغازين، هر مورچه به ترتيب از يك گره به گرة ديگر حركت مي-كند. در گرة i ، مورچة شمارة k ، گره بعدي j را كه توسط آن مورچه پيموده نشده است با احتمال بدست آمده از تابع احتمال انتخاب مسير محاسبه مي‌كند. ساختن راه‌حل، بعد از اينكه هر مورچه يك دور كامل زد انجام مي‌شود.

مكانيزم انتخاب مسير: مورچه‌ها براي انتخاب گرة بعدي از مقادير فرمون‌ها و فاصلة بين گره‌ها استفاده مي‌كنند. هر چه فاصلة بين گرة جاري تا يك گره كمتر و مقدار فرمون بين آنها بيشتر باشد، احتمال انتخاب گره بيشتر مي‌شود. به عبارت ديگر فرمول انتخاب گرة بعدي رابطة مستقيمي با مقدار فرمون روي يال و عكس فاصلة بين گره‌ها دارد. عكس فاصلة بين دو گره را ديد[۱٤] مي‌ناميم و آنرا $\eta_{ij} = 1/d_{ij}$ تعريف مي‌كنيم. احتمال عبور از گره i به گره j طبق رابطه (۳) تعريف ميشود. پارامترهاي α و β به كاربر اجازه مي‌دهند بر روي نسبت اهميت مقدار دنبالة فرمون به ديد، كنترل داشته باشد.

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{j_k=1}^{n}[\tau_{ij_k}(t)]^{\alpha}.[\eta_{ij_k}]^{\beta}} \qquad (۳)$$

بهنگام سازي فرمون: بعد از اينكه هر مورچه يك دور توليد كرد مقدار دنبالة فرمون روي يال‌ها بهنگام مي‌شود. در كلوني مورچه‌ها اين عمل ابتدا با كاهش مقدار فرمون با يك فاكتور ثابت (ضريب تبخير) و سپس قرار دادن فرمون توسط هر مورچه روي يال‌هايي كه در دور خود پيموده انجام مي‌شود. ضريب تبخير كه مقداري بين صفر و يك دارد براي اجتناب از انباشتن بي‌حد دنبالة فرمون است و الگوريتم را قادر مي‌سازد تا تصميم‌هاي بدي را كه قبلاً گرفته شده است فراموش كند. روي يال‌هايي كه توسط مورچه‌ها انتخاب نشده‌اند شدت فرمون با تعداد تكرارها بصورت

نمايي كاهش مي‌يابد. الگوريتم به مرور مورچه‌هاي بيشتري را در طول مسيرهاي كوتاهتر هدايت مي‌كند و در نتيجه مقدار فرمون قرار داده شده روي اين مسيرها بالا خواهد رفت كه اين باعث افزايش احتمال انتخاب مسيرهاي كوتاهتر در آينده خواهد شد.

$\iota_{ij}(t+1)$ را شدت دنباله روي يال بين گره‌هاي i و j در لحظة t+1 مي‌ناميم كه با فرمول (٤) محاسبه مي‌شود.

$$\iota_{ij}(t+1) = \rho \cdot \iota_{ij}(t) + \Delta\iota_{ij}(t,t+1) \qquad (٤)$$

محاسبه مي‌شود كه در آن ρ ضريب تبخير است و $\Delta\iota_{ij}(t,t+1) = \sum_{k=1}^{m} \Delta\iota_{ij}^{k}(t,t+1)$ كه $\Delta\iota_{ij}^{k}(t,t+1)$ مقدار دنباله فرمون است كـه روي يال بين گره‌هاي i و j توسط مورچة k در زمان بين t و t+1 قرار داده مي‌شود. مقدار دنباله فرمون در زمان 0، $\iota_{ij}(0)$، مي‌تواند مقداري تـصادفي باشد (عموماً مقداري كوچک). انتخابهاي مختلفي در رابطه با چگونگي محاسبة $\Delta\iota_{ij}^{k}(t,t+1)$ و زمان بهنگام سازي $\iota_{ij}(t)$ وجود دارد كه منجر به نسخه‌هاي مختلفي از الگوريتم كلوني مورچه‌اي مي‌گردد[1,5,6].

۴- مدل كردن كلوني مورچه‌ها توسط اتوماتاهاي يادگير

در اين بخش الگوريتم‌هاي كلوني مورچه‌ها را با استفاده از مجموعه‌اي از اتوماتاهاي يادگير مدل مي‌كنيم. همانطور كه قبلا اشاره شد الگوريتم‌هاي كلوني مورچه‌ها از ٤ مولفه گراف مساله، مورچه‌ها، مكانيزم انتخاب گره بعدي و مكانيزم بهنگام سازي فرمون تشكيل شده است. در ادامه معادل هر يک از اين مولفه‌ها در مدل اتوماتاهاي يادگير شرح داده مي‌شود.

گراف مساله: گراف مساله مجموعه‌اي از اتوماتاهاي يادگير است و بصورت زير تعريف مي‌شود:

$G \equiv \{NLA_1, NLA_2, ..., NLA_n\}$

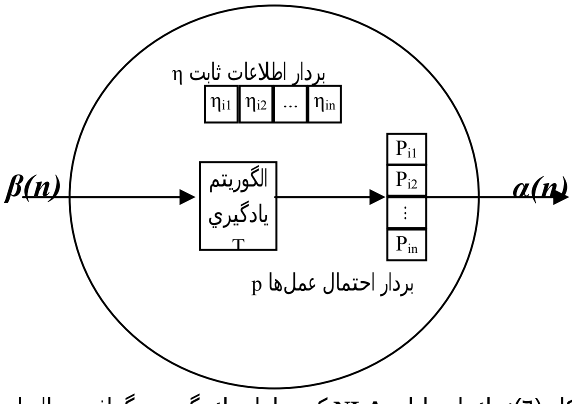
اتوماتاي يادگير هر گره بصورت 5 تايي زير تعريف مي‌شود:

$NLA \equiv \{\alpha, \beta, p, T, \eta\}$

كه:

- $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه عمل‌هاي اتوماتاي يادگير مي‌باشد كه معادل با يال‌هايي است كه از گره متناظر با آن اتوماتا در گراف مساله خارج مي‌شود.
- $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه ورودي‌هاي اتوماتاي يادگير.
- $p \equiv \{p_1, p_2, ..., p_r\}$ بردار احتمال انتخاب هر يك از عمل‌ها كه معادل بردار احتمال انتخاب مسير در كلوني مورچه‌ها مي‌باشد..
- $p(n+1) = T[\beta(n), p(n)]$ الگوريتم يادگير براي اتوماتاهاي يادگير است كه بردار احتمال انتخاب مسير را با توجه به دورهايي كه اتوماتاهاي يادگير توليد كرده‌اند بروز مي‌كند. (در كلوني مورچه‌ها اين احتمالات توسط تغيير مقادير فرمون‌ها تغيير مي‌كنند.)
- $\eta \equiv \{\eta_1, \eta_2, ..., \eta_r\}$ ماتريس اطلاعات مسالة است كه فاصلة گرة جاري تا هر يك از گره‌هاي ديگر در آن ذخيره مي‌شود.

هر عمل اتوماتاي NLA مانند عمل α_i معادل يك يال در گراف كلوني مورچه‌ها است. براي مثال α_i معادل يالي است كه به گرة iام متصل است. بردار η مقادير فاصلة بين گره‌ها را در خود ذخيره كرده است و در الگوريتم يادگيري نقشي ندارد. اين بردار را بردار اطلاعات ثابت مساله ناميده‌ايم. اندازة بردارهاي اتوماتاهاي يادگير با هم برابر است و مساوي است با تعداد گره‌هاي گراف مساله و انديس j هر بردار اطلاعات مربوط به گرة jام را در خود دارد. براي مثال عنصر jام بردار η مسافت گرة جاري تا گرة j را در خود ذخيره مي‌كند و عنصر jام بردار p نيز احتمال انتخاب گرة jام را نشان مي‌دهد.. شكل (٦) يك گره از گراف را نشان مي‌دهد.



شكل (٦): يك اتوماتاي **NLA** كه معادل يك گره در گراف مساله است

مكانيزم انتخاب گره بعدي در اتوماتاي **NLA**: در الگوريتمهاي كلوني مورچه‌ها انتخاب گره بعدي به مقادير فرمون‌ها و مقادير ثابت مساله (فاصلة بين گره‌ها) بستگي دارد. در بخش ۳ اين مكانيزم شرح داده شد. انتخاب هر گره باعث افزايش فرمون مسير مربوطه و تقويت آن مسير مي‌گردد. مكانيزم انتخاب شهر بعدي در كلوني مورچه‌ها به اين صورت است كه ابتدا براي هر يالي كه از گرة جاري خارج مي‌شود يك احتمال محاسبه مي‌شود. مجموع احتمال تمام يالهاي خارج شونده از يك گره برابر با ۱ مي‌باشد. سپس بر اساس احتمال‌هاي محاسبه شده، يك يال بصورت تصادفي انتخاب مي‌گردد. در مدل اتوماتاهاي يادگير هر NLA داراي يك بردار احتمال مي‌باشد كه هر عنصر اين بردار احتمال انتخاب يال متناظر با آن عنصر را نشان مي‌دهد. انتخاب گرة بعدي در مدل اتوماتاهاي يادگير معادل انتخاب یک عمل توسط يكي از اتوماتاهاي يادگير مي‌باشد..

الگوريتم يادگيري اتوماتاي **NLA**: در الگوريتم‌هاي كلوني مورچه‌ها عمل يادگيري با تغيير مقدار فرمون يال‌ها صورت مي‌گيرد به اين صورت كه هر چه فرمون روي يك يال بيشتر باشد احتمال انتخاب آن يال در مراحل بعدي بيشتر مي‌شود. در اتوماتاي يادگير عمل يادگيري با افزايش و يا كاهش احتمالهاي انتخاب عمل‌هاي اتوماتاهاي يادگير با توجه به نتيجه عمل بر طبق الگوريتم يادگيري صورت مي‌گيرد.
در الگوريتم‌هاي كلوني مورچه‌ها مقدار پاداشي كه به هر يال داده مي‌شود به نسبت طول دوري است كه مورچه توليد كرده است. اگر طول دور توليد شده توسط مورچه كوتاهتر باشد مقدار پاداش يا فرمون بيشتري بر روي هر يال در دور طي شده قرار خواهد گرفت و هر چه طول دور توليد شده بيشتر باشد مقدار فرمون به جاي گذاشته بر روي هر يال دور كمتر خواهد بود. در مدل اتوماتاهاي يادگير پاداش دادن و يا جريمه كردن اعمال هر NLA باعث تغيير بردار احتمال عمل‌هاي آن NLA و در نتيجه تغيير احتمال انتخاب يال متناظر در مراحل بعد خواهد شد. مقدار اولية بردار احتمال اعمال هر اتوماتاي NLA در ابتدا بر اساس فرمول (۵) محاسبه مي‌گردد.

$$p_{ij} = \frac{\eta_{ij}^{\beta}}{\sum_{l=1}^{n} \eta_{il}^{\beta}} \qquad (5)$$

p_{ij} احتمال انتخاب عمل jام در اتوماتاي NLA iام است و η_{ij} فاصله بين گره i تا گره j در گراف مساله مي‌باشد و β پارامتري است بين صفر و یک كه اهميت نسبي فاصله‌ها را مشخص مي‌كند.. فرمول يادگيري اتوماتاي NLA را براي پاداش عمل iام مي‌توان بصورت زير نوشت:

$$p_i(n+1) = \frac{p_i(n) + \Delta p}{1 + \Delta p} \qquad \text{(٦- الف)}$$

$$p_j(n+1) = \frac{p_j(n)}{1 + \Delta p} \quad 1 \le j \le n, j \ne i \qquad \text{(٦- ب)}$$

كه $k \succ 0$ $\Delta p = \frac{k}{f(c)}$ و $f(c)$ طول دوري است كه مورچة cام توليد كرده است. ثابت k نرخ يادگيري است و ۰/۱ در نظر گرفته مي‌شود. مقدار k بايد به اندازة كافي بزرگ باشد تا در عمليات بهنگام‌سازي تاثير داشته باشد و آنقدر كوچك باشد تا همگرايي را مختل نكند. بهنگام سازي بعد از اينكه دوري تعيين شد بدين صورت انجام مي‌گيرد كه احتمال اعمال انتخابي اتوماتاهاي يادگير در طول آن دور طبق فرمول ٦ بهنگام مي شود. اين عمل مشابه بهنگام سازي فرمون‌ها در الگوريتم Ant-Cycle مي‌باشد[7].

۵- مطالعة موردي: مسيريابي در شبكه‌هاي كامپيوتري

مسيريابي در شبكه‌هاي كامپيوتري توسط مسيريابها صورت مي‌گيرد كه هدايت بسته‌ها در شبكه را به سمت مقصدهاي مورد نظر به عهده دارند. براي اينكه مسيريابها به درستي كار كنند و بسته‌ها سريع‌تر به مقصد برسند مسيرياب‌ها در تعامل با يكديگر سعي مي‌كنند كه كوتاهترين مسيرها به سمت مقصدهاي مختلف را در هر زمان بيابند. در الگوريتم‌هاي مسيريابي قطعي با توجه به آدرس مقصد موجود در بسته و اطلاعات موجود در مسيرياب يك مسير براي ارسال بسته تعيين مي‌شود. اگر بسته‌اي با مقصد مشابه دوباره وارد مسيرياب شود و اطلاعات مسيرياب تغييري نكرده باشد همان مسير قبلي براي ارسال بسته انتخاب مي‌شود. حال آنكه در مسيريابي غير قطعي مانند مسيريابي توسط كلوني مورچه‌ها ممكن است دو مسير متفاوت براي دو بسته با مقصد يكسان انتخاب شود. در اين بخش مسالة مسيريابي در شبكه‌هاي كامپيوتري را با مدل كلوني مورچه-ها و همچنين مدل اتوماتاهاي يادگير انجام داده و از طريق شبيه‌سازي كارايي آنها را مورد بررسي قرار مي‌دهيم.

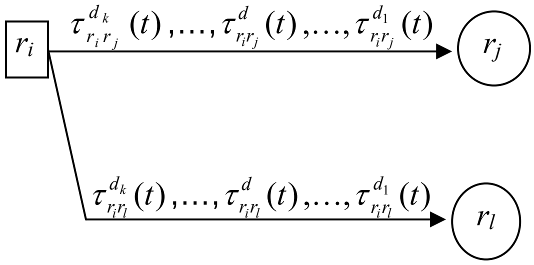
۵-۱- الگوريتم مسيريابي كلوني مورچه‌اي و مولفه‌هاي آن

در الگوريتم مسيريابي توسط كلوني مورچه‌ها شبكه كامپيوتري همان گراف مساله است و هر بسته به منزلة يك مورچه است. الگوريتم انتخاب مسير با توجه به هزينة زماني عبور يك بسته به مسيرياب همسايه (معادل فاصله در كلوني مورچه‌ها) و مقادير مكاشفه‌اي (فرمون)، يك مسير براي ارسال بسته پيدا مي‌كند. هزينة زماني ارسال بسته به مسيرياب مجاور را مي‌توان بر اساس بار خط ارتباطي مسيرياب و پهناي باند خط محاسبه كرد. مقادير فرمون هم با عبور بسته‌ها تغيير مي‌يابد و مقدار فرمون‌ها در مسيرياب‌ها نگهداري مي‌شود. با توجه به هزينة مسير انتخاب شده براي بسته‌ها، مقدار فرمون روي خطوط تغيير پيدا مي‌كند.

الگوريتم‌هاي مسيريابي توسط كلوني مورچه‌ها از ۳ مولفه تشكيل شده است كه عبارتند از: جدول مسيريابي، مكانيزم انتخاب مسيرياب بعدي و مكانيزم بهنگام‌سازي اطلاعات مسيرياب. در ادامه اين بخش اين مولفه ها شرح داده مي‌شود.

جدول مسيريابي: در الگوريتم‌هاي مسيريابي قطعي، هر يال هزينه‌اي دارد كه مي‌تواند با توجه به سرعت و ازدحام آن خط محاسبه شود. الگوريتم مسيريابي قطعي بر مبناي اين هزينه‌ها كوتاهترين مسيرها را تعيين مي‌كند. در الگوريتم‌هاي مسيريابي مبتني بر كلوني مورچه‌ها اگر يك يال در طول مسيرهاي متفاوت به n مسيرياب (مقصد) متفاوت قرار گيرد براي هر مسيرياب يك مقدار فرمون جداگانه بر روي آن يال نگهداري مي‌كند. اگر گراف مساله كامل باشد، هر يال (خط ارتباطي در شبكه) براي $|R|-1$ مقصد ممكن اطلاعات فرمون را نگهداري مي‌كند. كه $|R|$ تعداد مسيرياب‌هاي شبكه است. يعني اگر يك بسته در مسيرياب $r_i \in R$ قرار داشته باشد و بخواهد به مسيرياب مقصد d برود، تابع انتخاب مسير، احتمال عبور را بر اساس مقدار فرمون مربوط به مقصد d روي يالهايي كه از مسيرياب r_i خارج شده‌اند محاسبه مي‌كند (مانند حركت مورچه‌ها در ACO).

مقدار دنبالة $\tau^{d}_{r_i r_j}(t)$ را مقدار فرمون روي خط (r_i, r_j) در زمان t كه به مسيرياب مقصد d منتهي مي‌شود تعريف مي‌كنيم. در الگوريتم‌هاي مسيريابي مبتني بر كلوني مورچه‌ها هر سطر در جدول مسيريابي، مقدار فرمون يك يال خروجي در خود دارد. شكل (۷) اين مساله را نشان مي‌دهد.



شكل (۷): مقدار فرمون‌هاي روي يالها در مسيريابي كلوني مورچه‌ها

مكانيزم انتخاب مسيرياب بعدي: قانون انتخاب گره بعدي تقريباً شبيه قانون استفاده شده در ACO است. احتمال انتخاب مسيرياب بعدي r_j در زمان t در مسيرياب r_i براي بستة k كه به مقصد d مي‌رود طبق رابطة (۷) محاسبه مي‌شود.

$$p^{k}_{r_i r_j}(t) = \begin{cases} \dfrac{1}{\sum\limits_{r_l \in J^{T_k}_{r_i}} 1} & if\ \chi \le p_r \\ \dfrac{\left[\tau^{d}_{r_i r_j}(t)/b_\tau\right]^{\alpha} . \left[\eta_{r_i r_j}\right]^{\beta}}{\sum\limits_{r_l \in J^{T_k}_{r_i}} \left[\tau^{d}_{r_i r_l}(t)/b_\tau\right]^{\alpha} . \left[\eta_{r_i r_l}\right]^{\beta}} & if\ \chi \succ p_r \end{cases} \qquad (۷)$$

كه $J^{T_k}_{r_i}$ همسايگي N_{r_i} است كه مسيريابهاي موجود درليست مسير T_k بستة k در آن نيستند و χ يك متغير تصادفي با توزيع يكنواخت در بازة [0,1] است. پارامترهاي α و β اهميت نسبي مقدار فرمون $\tau^{d}_{r_i r_j}(t)$ و مقدار مكاشفه‌اي $\eta_{r_i r_j}(t)$ را مشخص مي‌كنند. براي مقدار مكاشفه‌اي محلي از تخمين هزينه استفاده از خط $w_k = (r_i, r_j)$ در مسيرياب r_i در زمان t استفاده مي‌كنيم كه با $\eta_{r_i r_j}(t)$ نشان داده مي‌شود:

$$\eta_{r_i r_j}(t) = \eta_{w_k}(t) = d(r_i, r_j).loadOut(r_i, r_j, t) + d(r_i, r_j) \qquad (۸)$$

تابع $loadOut$، حجم اطلاعات منتظر ارسال به مسيرياب بعدي را محاسبه مي‌كند. $\eta_{r_i r_j}(t)$، تخمين زمان مورد نياز براي حركت از صف خروجي r_i و خط w_k است. تخميني از مقدار $\eta_{r_i r_j}(t)$ با استفاده از اطلاعات موجود در مسيرياب محاسبه مي‌شود. $0 \le p_r \le 1$ احتمال

انتخاب مسيرياب بعدي بطور تصادفي و $1-p_r$ احتمال انتخاب مسيرياب با استفاده از رابطة ۵ مي‌باشد. اين روش انتخاب براي مسيرياب باعث كشف مسيرهاي جديد مي‌گردد. احتمال $p^d_{r_ir_j}(t)$ اگر r_j در T_k باشد (يعني قبلاً از اين مسيرياب عبور كرده باشد) صفر است.

مكانيزم بهنگام‌سازي اطلاعات مسيرياب: بهنگام سازي اطلاعات مسيرياب با استفاده از دو رابطه انجام مي‌گيرد كه يكي براي قرار دادن فرمون روي تمام مسيرياب‌هاي يك مسير مورچه و دومي براي بهنگام‌سازي فرمون يك مسيرياب وقتي كه يك بسته گم مي‌شود مي‌باشند. قرار دادن و بهنگام‌سازي فرمون بعد از اتمام مسيريابي بستة kام صورت مي‌گيرد. براي هر $r_i \in T_k$ مجموعة $\{\tau^d_{r_ir_j}(t) \mid (r_i, r_j) \in W\}$ با استفاده از فرمول (۹) انجام مي‌گيرد.

$$\tau^d_{r_ir_j}(t) = \begin{cases} b_\tau & if\ \tau_{unbound} < b_\tau \\ 1-b_\tau & if\ \tau_{unbound} > 1-b_\tau \\ \tau_{unbound} & else \end{cases} \qquad (۹)$$

پارامتر b_τ حداقل و حداكثر مقدار مجاز فرمون‌ها را نشان مي‌دهد. بازة مجاز مقادير فرمون‌ها براي $\tau^d_{r_ir_j}(t)$ عبارت است از $[b_\tau, 1-b_\tau]$. مقدار b_τ بايد در بازة [0,1/2] باشد. $\tau_{unbound}(t)$ مقدار جديد فرمون است و با رابطة (۱۰) محاسبه مي‌شود:

$$\tau_{unbound}(t) = \begin{cases} \tau^d_{r_ir_j}(t) + \tau_{upd}\tau^d_{r_ir_j}(t) & if\ \tau^d_{r_ir_j}(t) \le 0.5 \\ \tau^d_{r_ir_j}(t) + \tau_{upd}(1-\tau^d_{r_ir_j}(t)) & if\ \tau^d_{r_ir_j}(t) \succ 0.5 \end{cases} \qquad (۱۰)$$

مقدار τ_{upd} از رابطة (۱۱) بدست مي‌آيد:

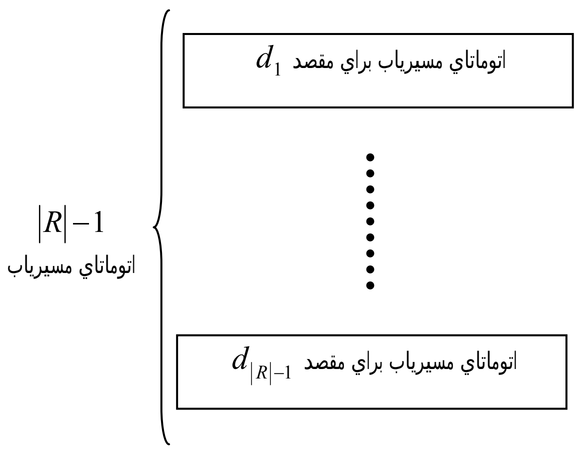
$$\tau_{upd} = \frac{k}{length\,(p,t)} \qquad (۱۱)$$

كه ثابت k نرخ يادگيري و $length(p,t)$ زمان صرف شده براي مسيريابي است. در صورت گم شدن بسته در فرمول ۷, $\tau_{upd} = n_\tau$ مي‌باشد. مقدار پارامتر n_τ بايد در بازه [0,1] باشد.

۵-۲- مسير يابي در شبكه اي كامپيوتري توسط مدل اتوماتاهاي يادگير

در اين بخش يک الگوريتم مسيريابي با استفاده از مدل اتوماتاهاي يادگير ارايه مي‌گردد. در ادامه معادل هر يک از مولفه‌هاي الگوريتم مسيريابي مبتني بر كلوني مورچه‌ها در مدل اتوماتاي يادگير شرح داده مي‌شود.

جدول مسيريابي: اطلاعات جدول مسيريابي توسط اتوماتاهاي يادگير حفظ مي‌شود. تغيير ساختار اتوماتاهاي يادگير به منزله تغيير جدول مسيريابي است. اگر گراف شبكه كامل باشد در هر مسيرياب $r_i \in R$ تعداد $|R|-1$ اتوماتاي يادگير وجود دارد كه به آنها اتوماتاهاي يادگير مسيرياب مي‌گوييم. هر اتوماتاي يادگير اطلاعات مسيريابي يك مقصد مشخص را در خود نگه مي‌دارد. به عبارت ديگر به جاي اينكه هر خط جداگانه اطلاعات مقصدهاي مختلف را نگهداري كند، اطلاعات هر مقصد مشخص مانند d توسط يك اتوماتاي يادگير حفظ مي‌شود. هر يك از اعمال اين اتوماتاها متناظر با یک مسيرياب است كه بسته مي‌تواند از طريق آن به سمت مقصد مورد نظر ارسال گردد. تعداد اعمال اتوماتاهاي يادگير برابر با تعداد خطوط متصل به مسيرياب است. هر اتوماتاي يادگير سعي مي‌كند در فرآيند مسيريابي بهترين مسير را براي هر بسته پيدا كند. شكل (۸) جدول مسيريابي را نشان مي‌دهد. جدول مسيريابي در شبكة (R,W) براي هر مسيرياب $r_i \in R$ كه به مجموعه‌اي از m مسيرياب $\{r_j \in R \mid (r_i, r_j) \in W\}$ متصل است شامل m اتوماتاي مسيرياب NLA است.



شكل (۸): جدول مسيريابي

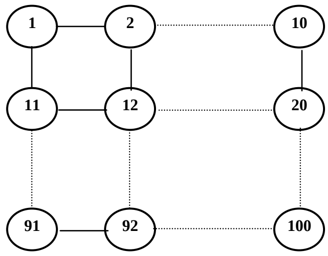
مکانیزم انتخاب مسیریاب بعدي: اگر یک بسته که به مقصد d مي‌رود وارد مسیریاب i شود آنگاه اتوماتايي که وظیفه مسیریابي بسته‌ها براي مقصد d را دارد یک عمل را انتخاب مي‌کند. عمل انتخاب شده یک خط خروجي است که بسته از طریق آن مي‌تواند به سوي مقصد مورد نظر هدایت شود.

مکانیرم بهنگام‌سازي اطلاعات مسیریاب‌ها : بهنگام‌سازي مقادیر بردار احتمالات بعد از اینکه مسیریابي یک بسته به پایان رسید مطابق فرمول ۶ صورت مي‌گیرد. در اینجا $f(c)$ هزینه مسیریابي بسته مورد نظر است.

در فرمول یادگیري اتوماتاي NLA پارامتري به نام k وجود دارد که به آن نرخ یادگیري مي‌گوییم. پارامتر k براي هر اتوماتاي یادگیر با استفاده از یک اتوماتاي یادگیر دیگري که در آن گره قرار دارد با توجه به کارایي مسیریابي تطبیق داده مي‌شود[8].

۳-۵- نتایج شبیه‌سازي

شبیه‌سازي‌ها بر روي يك شبكه‌اي از مسيرياب‌ها مطابق شكل (۹) از مسيرياب‌ها انجام شده است. اين شبكه متشكل از ۱۰۰ مسيرياب است. تأخير براي هر خط ۱۰ واحد در نظر گرفته شده است. هر مسيرياب براي هر خط خروجي يك بافر دارد و تنها داراي يك صف ورودي است.



شكل (۹): شبكة استفاده شده براي شبيه‌سازيها

مسيرياب ۲۳ ارسال كنندة بسته‌ها و مسيرياب ۸۵ دريافت كنندة بسته‌ها است. ظرفيت صف ورودي مسيرياب ارسال كننده خيلي زياد در نظر گرفته شده است تا در صورتي كه نرخ ارسال بسته‌ها توسط اين مسيرياب افزايش پيدا كند اين مسيرياب توانايي حفظ و سپس ارسال آنها را داشته باشد. زمان پردازش بسته در داخل مسيرياب ارسال كننده ۷ واحد در نظر گرفته شده است. صف ورودي مسيرياب دريافت كننده ۱۰ بسته ظرفيت دارد و زمان پردازش بسته‌ها در آن ۱ واحد است. براي ساير مسيرياب‌ها اندازة صف ورودي و صف خروجي ۱۰ است و زمان پردازش بسته‌ها در آنها ۱۵ واحد در نظر گرفته شده است.

شبيه سازي الگوريتم مسيريابي توسط كلوني مورچه‌ها : سه آزمايش متفاوت انجام گرفته است. آزمايش اول با در نظر گرفتن مقادير بالا براي پارامترهاي مساله ($\alpha = 0.9$ و $\beta = 0.9$) و آزمايش دوم با در نظر گرفتن مقادير پايين براي پارامترهاي مساله ($\alpha = 0.5$ و $\beta = 0.5$) انجام شده است. آزمايش سوم براي بررسي تاثير مقدار جريمه گم شدن بسته‌ها در كارايي مسيريابي انجام گرفته است. نتايج اين شبيه‌سازي‌ها در جدول (۱) آمده است. با توجه به نتايج در صد گم شدن بسته‌ها در آزمايش اول بسيار زياد است و از بيشتر بسته‌ها به علت پر بودن صف ورودي صرف‌نظر مي‌شود.

نتايج آزمايش دوم نشان مي‌دهد و مقايسة آن با نتايج آزمايش اول نشان مي‌دهد كه الگوريتم كلوني مورچه‌ها زماني كه مقدار پارامترهاي آن بالا نباشد بهتر كاراي بهتري را از خود نشان مي‌دهد. در آزمايش سوم پارامتر جريمه افزايش داده شده است. همانطور كه نتايج جدول (۱) نشان مي‌دهد اين تغيير باعث تغيير محسوسي در كارايي الگوريتم مسيريابي نمي‌گردد.

جدول (۱): نتايج آزمايش‌هاي الگوريتم مسيريابي مبتني بر كلوني مورچه‌ها

پارامترهاي شبيه‌سازي شده	نتيجة آزمايش:		
	اول	دوم	سوم
تعداد كل بسته‌ها	10000	5000	5000
تعداد كل بسته‌هاي از بين رفته	3431	2866	2849
بسته‌هايي كه شمارندة آنها تمام شده	0	0	0
بسته‌هاي از بين رفته به علت صف ورودي پر	3050	1286	851
بسته‌هاي از بين رفته به علت صف خروجي پر	216	0	0
بسته‌هاي از بين رفته به علت نبودن مسير	165	1580	1990
ميانگين زمان مورد نياز براي مسيريابي	452.68	672.9	789.38

شبيه سازي الگوريتم مسيريابي توسط اتوماتاهاي يادگير: نتايج اين شبه‌سازي در جداول (۲) آمده است. تعداد بسته‌هايي كه گم شده است نسبت به آزمايش قبل بسيار كمتر شده است. زمان ميانگين زمان مورد نياز براي مسيريابي هر بسته نيز كاهش يافته است. در جدول (۳) الگوريتم ارايه شده با بهترين نتيجة الگوريتم مسيريابي كلوني مورچه‌ها مقايسه شده است. با توجه به جدول (۳) مشخص است كه تعداد كل بسته‌هاي از بين رفته در الگوريتم مسيريابي توسط اتوماتاهاي يادگير بسيار كمتر شده است. همچنين ميانگين زمان رسيدن بسته‌هاي شبكه به مقصد نيز كاهش يافته است.

جدول ۲- نتايج آزمايش الگوريتم مسيريابي مبتني بر اتوماتاهاي يادگير

تعداد كل بسته‌ها	5000
تعداد كل بسته‌هاي از بين رفته	362
بسته‌هايي كه شمارندة آنها تمام شده	0
بسته‌هاي از بين رفته به علت صف ورودي پر	216
بسته‌هاي از بين رفته به علت صف خروجي پر	0
بسته‌هاي از بين رفته به علت نبودن مسير	146
ميانگين زمان مورد نياز براي مسيريابي	396.12

جدول (۳): مقايسة الگوريتم مبتني بر اتوماتاهاي يادگير با الگوريتم مبتني بر كلوني مورچه‌ها

پارامترهاي آزمايش	الگوريتم ارايه شده	الگوريتم كلوني مورچه‌ها
تعداد كل بسته‌ها	5000	5000
تعداد كل بسته‌هاي از بين رفته	362	2866
بسته‌هايي كه شمارندة آنها تمام شده	0	0
بسته‌هاي از بين رفته به علت صف ورودي پر	216	1286
بسته‌هاي از بين رفته به علت صف خروجي پر	0	0
بسته‌هاي از بين رفته به علت نبودن مسير	146	1580
ميانگين زمان مورد نياز براي مسيريابي	396.12	672.9

۶- نتيجه گيري

در اين مقاله ابتدا نشان داده شد كه الگوريتم‌هاي كلوني مورچه‌ها را مي‌توان با مجموعه‌اي از اتوماتاهاي يادگير كه با يكديگر در تعامل هستند مدل كرد. سپس با استفاده از مجموعه‌اي از اتوماتاهاي يادگير، الگوريتمي براي حل مسالة مسيريابي در شبكه‌هاي كامپيوتري ارايه گرديد. از طريق شبيه‌سازي كامپيوتري، الگوريتم مسيريابي مبتني بر اتوماتاهاي يادگير با الگوريتم مسيريابي مبتني بر كلوني مورچه‌ها مقايسه گرديد. نتايج شبيه-سازيها برتري رويكرد اتوماتاهاي يادگير را حداقل براي مطالعة موردي در نظر گرفته شده در اين مقاله نشان داده است.

مراجع

[1] Dorigo, M., Stuetzle T., "The Ant Colony Optimization Metaheuristic, Algorithms, Applications and Advances", Technical Report IRIDIA/2000-32, Belgium: Université Libre de Bruxelles, 2000.
[2] Narendra, K., Thathachar, M., Learning Automata: An Introduction, Prentice Hall 1989.
[3] Sutton, R. S., Barto, A. G., "Reinforcement Learning: An Introduction", Cambridge, MA: MIT Press 1998.
[4] Dorigo M., Di Caro, G., "AntNet, A Mobile Agents Approach to Adoptive Routing", Tech.Rep.IRIDIA/9712, Universit Libre de Bruxelles, Belgium, 1997.
[5] Colorni, A., Dorigo, M., Maniezzo, V., "The Ant System: An Autocatalytic Optimizing Process", Tech. Rep. 91-016 Revised, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1991.
[6] Colorni, A., Dorigo, M., Maniezzo, V., Positive Feedback as a Search Strategy, Tech. Rep.91-016, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1991.
[7] Colorni, A., Dorigo, M., Maniezzo, V., "The Ant System: Optimization by a Colony of Cooperating Agents", IEEE Transactions on Systems, Man, and Cybernetics – Part B, Vol. 26, No. 1 pp. 29–41, 1996.
[8] Meybodi M.R., Mohammadi, F.A., Optimizing ACS Using Learning Automata. Technical Report, Softcomputing Lab, Amirkabir University of Tech., 2004.
[9] Meybodi, M. R. "Results on Strongly Absolutely Expedient Learning Automata", Proc. Of OU Inference Conf. 86, ed. D. R. Mootes and Butrick, Athens, Ohio, Ohio Univ. Press, pp. 197-204, 1987.

[10] Mars, P., Chen, J. R. and Nambiar, R. Learning Algorithms: Theory and Application in Signal Processing, Control, and Communications, CRC press, New York, 1998.
[11] Mars, P., Narendra, K. S and Chrystall, M., "Learning Automata Control of Computer Communication Networks", Proc. of Third Yale Workshop on Applications of Adaptive Systems Theory, Yale University, 1983.
[12] Adibi, P., Meybodi M. R. and Safabakhsh, R., "Unsupervised Learning of Synaptic Delays based on Learning Automata in an RBF-Like Network of Spiking Neurons for Data Clustering", Journal of Neurocomputing, No. 64, Elsevier Publishing Company, pp. 335-357, 2005.
[13] Mashoufi, B., Meybodi, M. R., Motamedi, S. A. and Menhaj, M. B., "Adaptive Survival Algorithm", International Journal of Engineering Science, Vol. 15, No. 3, pp. 1-16, 2004.
[14] Meybodi, M. R. and Beigy, H., "A Note on Learning Automata Based Schemes for Adaptation of BP Parameters", Journal of Neurocomputing, Vol. 48, No. 4, pp. 957-974, October 2002.
[15] Meybodi, M. R. and Beigy, H., "New Learning Automata Based Algorithms for Adaptation of Backpropagation Algorithm Parameters'', International Journal of Neural System, Vol. 12, No. 1, PP. 45-67, 2002.
[16] Beigy, H. and Meybodi, M. R., "Adaptation of Momentum Factor and Steepness Parameter in Backpropagation Using Fixed Structure Learning Automata, " International Journal of Science and Technology (Iranica Saientia), Vol. 8, No. 4, pp. 250-264, Fall 2001.
[17] Beigy, H. and Meybodi, M. R., "Backpropagation Algorithm Adaptation Parameters Using Learning Automata", International Journal of Neural System, Vol. 11, No. 11, No. 3, pp. 219-228, 2001.
[18] 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.
[19] Meybodi, M. R. and Beigy, H., "New Class of Learning Automata Based Scheme for Adaptation of Backpropagation Algorithm Parameters", International Journal of Neural Systems, vol. 12, No. 1, pp. 45-67, 2002.
[20] Oommen, B. J. and Ma, D. C. Y., "Deterministic Learning Automata Solutions to the Equipartitioning Problem", IEEE Trans. on Computers, No. 37, No. 1, pp. 2-13,1988.
[21] Oommen, B. J., Valivet, R. S. and Zgierski, J. R., "An Adaptive Learning Solution to the Keyboard Optimization Problem", IEEE Trans. on Systems, Man, and Cybernetics, Vol. 21, No. 6, pp. 1608-1618, 1991.
[22] Oommen B. J. and de St Croix, E. V., " Graph Partitioning Using Learning Automata", IEEE Trans. on Computers, No. 45, No. 2, pp. 195-208, 1996.
[23] Thathachar M. A. L. and Sastry, P. S., "Learning Optimal Discriminant Functions Through a Cooperative Game of Automata", IEEE Trans. Syst., Man and Cybern., Vol. SMC-27, pp.73-85, 1987.
[24] Beigy, H. and Meybodi, M. R., "Solving the Graph Isomorphism Problem Using Learning Automata", In Proceedings of 5th Annual International Computer Society of Iran Computer Conference, CISCC-2000, pp. 402-415, 2000.
[25] Beigy, H. and Meybodi, M. R., "Backpropagation Algorithm Adaptation Parameters using Learning Automata, International Journal of Neural Systems, Vol. 11, No. 3, pp. 219-228, 2000.
[26] Meybodi, M. R. and Lakshmivarhan, S., "Optimality of a General Class of Learning Algorithm", Information Science, Vol. 28, pp. 1-20, 1982.
[27] Meybodi, M. R. and Lakshmivarhan, S., "On a Class of Learning Algorithms which have a Symmetric Behavior Under Success and Failure", Springer-Verlag, Lecture Notes in Statistics, pp. 145-155, 1984.
[28] Mashoufi, B., Menhaj, M. B., Motamedi, S. A. and Meybodi, M. R., "Introducing an Adaptive VLR Algorithm Using Learning Automata for Multilayer Perceptron", DEICE Transactions on Information and Systems, Vol. E86-D, No. 3, pp. 495-609, March 2003.

زیرنویس‌ها

[1] Ant Colony
[2] Tsetlin
[3] Pheromone
[4] Learning Automata
[5] Finite State Machine
[6] Environment
[7] Variable Structure Learning Automata
[8] Fixed Structure Learning Automata
[9] Linear Reward Penalty
[10] Linear Reward Epsilon Penalty
[11] Linear Reward Inaction
[12] Krinsky
[13] Krylov
[14] Visibility