

الگوریتم مبتنی بر اتوماتاهای یادگیر برای مسئله پوشش مجموعه

آیلین موسویان^{۱*}، علیرضا رضوانیان^۲، محمدرضا میبیدی^۳

*مسئول مکاتبات:

^۱دانشجوی کارشناسی ارشد، دانشکده مهندسی برق، رایانه و فناوری اطلاعات، دانشگاه آزاد اسلامی واحد قزوین، قزوین، ایران

A.mousavian@qiau.ac.ir

^۲دانشجوی دکتری، آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)، تهران، ایران

a.rezvanian@aut.ac.ir

^۳استاد، آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)، تهران، ایران

mmeybodi@aut.ac.ir

چکیده

مسئله پوشش مجموعه یکی از مسائل NP-Hard است که در کاربردهای مختلفی مانند شبکه‌های ارتباطی مورد استفاده قرار می‌گیرد. هدف از پوشش مجموعه، یافتن یک زیرمجموعه به گونه ایست که اجتماع اعضای این زیر مجموعه، کل مجموعه را پوشش دهد. پوشش مجموعه-ای در نظریه گراف به دو صورت پوشش راس و پوشش یال می‌باشد. با توجه به آنکه برای مسئله پوشش مجموعه نمی‌توان جواب دقیقی در زمان چندجمله‌ای پیدا کرد، بنابراین روش‌های هیوریستیکی مختلفی نیز برای حل آن ارائه شده است. در این مقاله یک الگوریتم مبتنی بر اتوماتاهای یادگیر برای حل مسئله پوشش مجموعه پیشنهاد شده است. در الگوریتم پیشنهادی، هر یک از رئوس گراف به یک اتوماتای یادگیر مجهز می‌شوند که دارای دو عمل حضور یا عدم حضور راس متناظر در مجموعه پوشش است. با توجه به همکاری میان اتوماتاهای یادگیر، در هر مرحله بردار احتمال اعمال اتوماتاهای یادگیر به‌روز می‌شود و این روند به‌طور تکراری ادامه می‌یابد تا آنکه در خاتمه‌ی الگوریتم، مجموعه پوشش نزدیک به بهینه بدست آید. جهت ارزیابی الگوریتم پیشنهادی مبتنی بر اتوماتاهای یادگیر، از دادگان آزمایشی معروف DIMACS استفاده شده است. نتایج شبیه‌سازی در مقایسه با سایر روش‌های متداول برای آزمایشات مختلف حاکی از موفقیت الگوریتم پیشنهادی است

کلمات کلیدی:

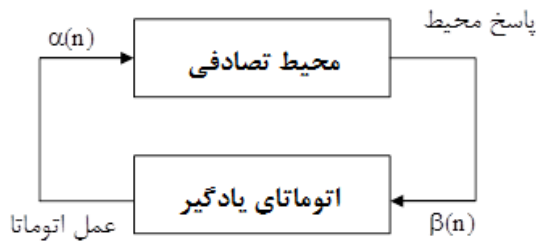
اتوماتای یادگیر، اتوماتای یادگیر توزیع شده، مسائل سخت، مجموعه پوشش، مجموعه پوشش راسی.

۱) مقدمه

را پوشش می‌دهد. تعداد رئوس موجود در پوشش راسی، اندازه آن را مشخص می‌کند که پوشش راسی با کمترین کاردینالیتی به عنوان پوشش راسی کمینه شناخته می‌شود. مساله پوشش راسی کمینه دارای پیچیدگی نمایی است و جزو مسائل NP کامل می‌باشد [۴]، [۵]، بنابراین یافتن راه‌حلی که بتواند در زمان چندجمله‌ای به جواب برسد، تقریباً ناممکن است [۶]، [۷].

راه‌حل‌های مختلفی برای حل مسئله پوشش مجموعه ارائه شده است. بسیاری از راه‌حل‌ها مبتنی بر تکنیک‌های برنامه-ریزی هستند [۸]، اما با توجه دشواری این روش‌ها، رویکردهایی مبتنی بر الگوریتم‌های تقریبی نیز وجود دارد [۹]، [۱۰]. در [۱۱] دو الگوریتم تقریبی برای پوشش مجموعه پیشنهاد شده است. در این تحقیق ابتدا با استفاده از درخت

مسئله پوشش مجموعه به عنوان یکی از مسائل معروف در حوزه نظریه محاسبات در علوم کامپیوتر شناخته می‌شود. کارپ برای اولین بار در سال ۱۹۷۲ مسئله پوشش مجموعه را به عنوان یکی از مسائل NP-Complete معرفی کرد [۱]. از مسئله پوشش مجموعه برای حل مسائل مختلفی در کاربردهایی چون شبکه‌های کامپیوتری، سیستم‌های حمل و نقل، زمان‌بندی، بیوانفورماتیک و مسیریابی استفاده شده است [۲]، [۳]. پوشش مجموعه به صورت زیرمجموعه‌ای از اعضا است، به طوریکه اجتماع اعضای این زیر مجموعه کل مجموعه را شامل شود. پوشش مجموعه در نظریه گراف به دو صورت پوشش راس و پوشش یال است. در پوشش راسی هدف یافتن مجموعه‌ای از رئوس است که کلیه یال‌های گراف



شکل ۱: ارتباط بین اتوماتای یادگیر و محیط

اتوماتای یادگیر با ساختار متغیر را می‌توان چهارتایی $\{\alpha, \beta, p, T\}$ نشان داد که $\alpha = \{\alpha_1, \dots, \alpha_r\}$ مجموعه عمل‌های اتوماتا، $\beta = \{\beta_1, \dots, \beta_m\}$ مجموعه ورودی‌های اتوماتا، $p = \{p_1, \dots, p_r\}$ بردار احتمال هر یک از اعمال بوده و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد. روابط (۱) و (۲) که در ادامه ذکر شده است چگونگی اعمال تغییرات در بردار احتمال در مرحله n ام با توجه به میزان ارزیابی عملکرد α_i را نشان می‌دهد. در صورتی که پاسخ مورد نظر مطلوب ارزیابی گردد رابطه (۱) را خواهیم داشت:

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

همچنین در صورتی که پاسخ نامطلوب ارزیابی گردد رابطه (۲) را خواهیم داشت:

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

بطوریکه در روابط (۱) و (۲) پارامتر a پارامتر پاداش و b پارامتر جریمه می‌باشند. با توجه به مقادیر مختلف a و b می‌توان سه حالت مختلف را متصور شد. در حالتیکه a و b با هم برابر باشند، الگوریتم را L_{RP} گویند، در زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را L_{REP} می‌نامند و در صورتیکه پارامتر جریمه (b) برابر صفر باشد، الگوریتم را L_{RI} می‌گویند.

۲-۱) اتوماتای یادگیر توزیع شده

اتوماتای یادگیر توزیع شده (DLA) [۱۷] شبکه‌ای از چند اتوماتای یادگیر است که برای حل یک مساله مشخص با یکدیگر همکاری می‌کنند. در این شبکه از اتوماتاهای یادگیر همکار در هر زمان تنها یک اتوماتا فعال است. تعداد اعمال قابل انجام توسط یک اتوماتا در DLA برابر با تعداد اتوماتاهایی است که به این اتوماتا متصل شده‌اند. انتخاب یک عمل توسط

اشتاینر کمینه، سعی در ایجاد بیشترین پوشش با کمترین تعداد راس را در مجموعه دارد، وی هم چنین روشی مبتنی بر تکنیک حریصانه با کمترین تعداد راس پوششی ارائه نموده است. روش‌های مبتنی بر تکنیک شاخه و حد برای این مساله ارائه شده است، اما از پیچیدگی بسیار بالایی برخوردار است [۱۲]. در میان روش‌های ارائه شده برای حل مسائل پوشش مجموعه، الگوریتم‌های هیوریستیکی از اقبال بیشتری برخوردار بوده‌اند. جانسون یک روش حریصانه برای حل این مسئله پیشنهاد کرد. در این روش، در ابتدا یک مجموعه تهی در نظر گرفته می‌شود و در هر گام با انتخاب بهترین کاندید که دارای درجه‌ی بیشینه است ادامه می‌یابد و با رسیدن به حالتی که دیگر پوششی وجود نداشته باشد، خاتمه می‌یابد [۱۳]. روش‌های هیوریستیکی دیگری به صورت الگوریتم ژنتیک [۱۴]، الگوریتم کولونی مورچه [۱۵]، روش‌های ترکیبی بهبود یافته با الگوریتم‌های محلی برای حل مساله فوق ارائه شده است [۶].

در این مقاله، از اتوماتای یادگیر برای حل مسئله پوشش مجموعه در گراف استفاده شده است. در الگوریتم پیشنهادی شبکه‌ای از اتوماتاهای یادگیر متناظر با گراف موردنظر ایجاد می‌شود. براساس همکاری میان اتوماتاهای یادگیر و مکانیزم یادگیری که برای هر یک از اتوماتون‌ها وجود دارد، الگوریتم پیشنهادی قادر است پاسخ‌های نزدیک به بهینه را بدست آورد.

در ادامه در این مقاله، ابتدا شرح مختصری درباره‌ی اتوماتای یادگیر و اتوماتای یادگیر توزیع شده در بخش ۲ ارائه می‌شود. در بخش ۳ الگوریتم پیشنهادی مبتنی بر اتوماتای یادگیر توزیع شده شرح داده می‌شود. در بخش ۴ نتایج آزمایشات انجام شده مورد بررسی قرار می‌گیرد و بخش ۵ شامل نتیجه‌گیری خواهد بود.

۲) اتوماتای یادگیر

اتوماتای یادگیر یک مدل انتزاعی از یک ماشین حالات محدود است که می‌تواند تعداد محدودی عمل را انجام دهد. اعمال بطور تصادفی انتخاب شده و توسط محیطی احتمالی ارزیابی می‌گردد و پاسخی به صورت پاداش یا جریمه به اتوماتای یادگیر داده می‌شود. اتوماتای یادگیر از این پاسخ استفاده نموده و عمل بعدی خود را انتخاب می‌کند. در طی این فرایند، اتوماتای یادگیر، یاد می‌گیرد که چگونه بهترین عمل را از بین اعمال مجاز خود انتخاب نماید. شکل ۱ ارتباط بین اتوماتای یادگیر و محیط را نشان می‌دهد [۱۶].

می‌گیرد. یال‌های متصل به هر نود مجموعه اقدام‌های هر اتوماتای یادگیر می‌باشد. تعداد اقدام‌های اتوماتا با مجموع یال‌های نود معادل است.

الگوریتم پیشنهادی به طور خلاصه در ۲ قدم زیر ارائه شده است.

(۱) مقداردهی اولیه بردار احتمال اقدام اتوماتاهای یادگیر

یک یک مجموعه پوشش به صورت تهی

(۲) تکرار مراحل زیر تا رسیدن به شرایط توقف

a. انتخاب یکی از اعمال اتوماتای یادگیر بر مبنای

بردار احتمال اقدام

b. پیمایش مسیری از گراف و تشکیل مجموعه

پوشش اولیه

c. ارزیابی محیط و به‌روز رسانی بردار احتمال

(۳) پایان

در ابتدا فرض می‌شود همه‌ی اتوماتاها غیر فعال هستند. در اولین تکرار اتوماتای منتخب به صورت تصادفی گزینش شده و فعال می‌شود، سپس به مجموعه اتوماتاهای پوشش‌دهنده اضافه می‌گردد. سپس همسایه‌های اتوماتای فعال شده به مجموعه همسایگی اضافه می‌شود. اتوماتای فعال شده با توجه به احتمال اقدام‌های موجود (یال‌های متصل به اتوماتا) عملی را انتخاب می‌کند. (در تکرار نخست این انتخاب تصادفی است، زیرا احتمال تمام یالها برابر است) این تکرار تا رسیدن به آستانه‌ای از پوشش رئوس گراف ادامه می‌یابد. در این فرایند مسیر اتوماتای انتخاب شده در یک مجموعه‌ی جداگانه قرار می‌گیرد تا اتوماتاهای فعال شده به ترتیب انتخاب مشخص باشند.

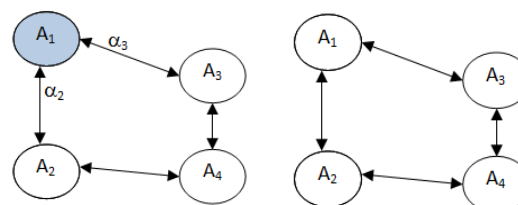
بعد از هر تکرار مسیر مربوط به تشکیل مجموعه پوشش راسی در دو تکرار متوالی می‌شود و نحوه تغییرات در مسیرهای طی شده به عنوان معیاری برای ارزیابی توسط محیط مورد استفاده قرار می‌گیرد. برای ارزیابی بهبود مسیر از رابطه زیر در دو تکرار متوالی استفاده شده است.

(۳)

به طوریکه، p_{vi} نیز بردار احتمال راس v_i متعلق به 1 مسیر طی شده در تکرار k ام است.

بنابراین اتوماتاهای که میزان همگرایی بیشتری در طول مسیر دارند، مورد پاداش قرار می‌گیرد در غیر اینصورت اتوماتاهای مورد نظر جریمه می‌شود. مکانیزم یادگیری براساس توابع معرفی شده در بخش ۲ اعمال می‌شود. برای توقف الگوریتم

اتوماتای یادگیر در این شبکه باعث فعال شدن اتوماتای یادگیر متصل شده به اتوماتای یادگیر متناظر با این عمل می‌گردد. به عبارت دیگر انتخاب یک عمل توسط یک اتوماتای یادگیر در این شبکه متناظر با فعال شدن یک اتوماتای یادگیر دیگر در این شبکه است. یک اتوماتای یادگیر توزیع شده را می‌توان با یک گراف جهت‌دار مدل کرد. به صورتی که مجموعه گره‌های آن مجموعه‌ای از اتوماتای یادگیر و یال‌های خروجی هر گره مجموعه اعمال متناظر با اتوماتای یادگیر متناظر با آن گره است. هنگامی که اتوماتا یکی از اعمال خود را انتخاب می‌کند، اتوماتایی که در دیگر انتهای یال متناظر با آن عمل قرار دارد، فعال می‌شود. بعنوان مثال در شکل زیر هر اتوماتا ۲ عمل دارد. اگر اتوماتای A_1 عمل α_3 خود را انتخاب کند، آنگاه اتوماتای A_3 فعال خواهد شد. در گام بعد، اتوماتای A_3 یکی از اعمال خود را انتخاب می‌کند که منجر به فعال شدن یکی از اتوماتاهای متصل به A_3 می‌شود.

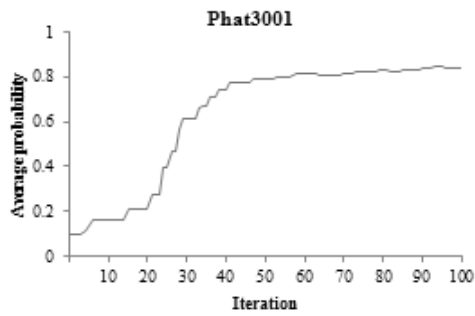


شکل ۲: اتوماتای یادگیر توزیع شده

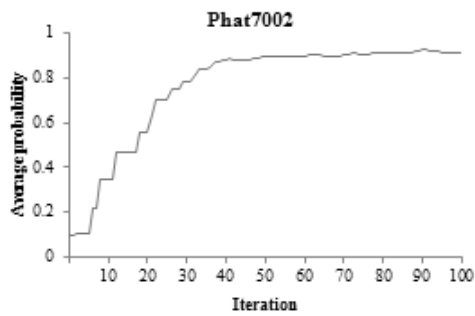
بصورت رسمی، یک اتوماتای توزیع شده با n اتوماتای یادگیر توسط یک گراف (A, E) تعریف می‌شود که $A = \{A_1, \dots, A_n\}$ مجموعه اتوماتا و $E \subset A \times A$ مجموعه لبه‌های گراف است بطوریکه لبه (i, j) متناظر با عمل α_j از اتوماتای A_i است. اگر بردار احتمال اعمال اتوماتای یادگیر A_j با $\underline{p}_m^j = \{p_1^j, p_2^j, \dots, p_r^j\}$ نشان داده شود، آنگاه p_m^j احتمال انتخاب عمل α_m از اتوماتای یادگیر A_j را نشان می‌دهد که احتمال انتخاب لبه خروجی (j, m) از میان لبه‌های خروجی گره j می‌باشد.

۳) الگوریتم پیشنهادی

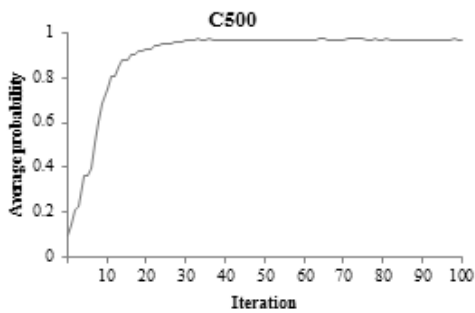
در الگوریتم پیشنهادی برای حل مسئله پوشش راسی، ورودی الگوریتم گراف غیرجهت‌دار $G(V, E)$ و خروجی الگوریتم مجموعه پوشش راسی می‌باشد. در ابتدا شبکه‌ای از اتوماتون-های یادگیر که متناظر با گراف مساله می‌باشد ایجاد می‌گردد به طوریکه در این شبکه در هر راس یک اتوماتای یادگیر قرار



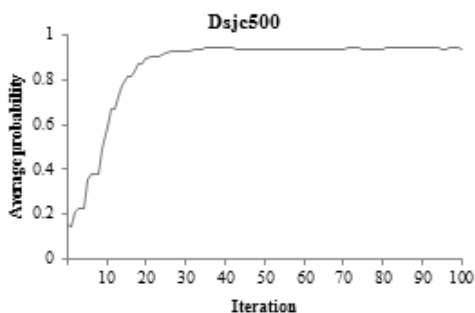
(الف)



(ب)



(پ)



(ت)

شکل ۳: رفتار همگرایی الگوریتم پیشنهادی براساس احتمال اعمال

می‌توان تعداد تکرار مشخصی از اجرای الگوریتم را در نظر گرفت. شبه کد مربوط به نحوه به روز رسانی اعمال در الگوریتم پیشنهادی به صورت زیر است.

Algorithm Updating action probability vector

If ($\pi_k(l) > \pi_{k-1}(l)$)

The chosen actions in path l by automata are rewarded

Else

The chosen actions in path l by automata are penalized

End if

Update action probability vector

شکل ۴: شبه کد به‌روزرسانی اتوماتای یادگیر برای پوشش مجموعه

به این ترتیب اتوماتا در تکرارهای متوالی یاد می‌گیرد که چگونه اتوماتا با بهترین عمل (بیشترین پوشش) را انتخاب کند. این اعمال تا زمانی که به یک مقدار بهینه نزدیک شود ادامه می‌یابد.

در بخش بعدی شبیه‌سازی‌هایی برای ارزیابی الگوریتم پیشنهادی ترتیب داده شده است.

(۴) نتایج آزمایشات

جهت ارزیابی الگوریتم پیشنهادی، از مجموعه آزمایشی گراف-های DIMACS استفاده شده است. مشخصات گراف‌های مورد استفاده در جدول (۱) ارائه شده است.

جدول ۱: گراف‌های منتخب DIMACS برای انجام آزمایشات

نام گراف	تعداد رئوس	تعداد یال
C500	۵۰۰	۱۱۲۳۳۲
Dsjc500	۵۰۰	۱۲۵۲۴۸
Mann27	۳۷۸	۷۰۵۵۱
P_hat3001	۳۰۰	۱۰۹۹۳
P_hat3003	۳۰۰	۳۳۳۹۰
P_hat7002	۷۰۰	۱۲۱۷۲۸

در کلیه‌ی آزمایشات اتوماهااتای یادگیر از نوع L_{RI} با نرخ یادگیری ۰/۱ لحاظ شده است.

نمونه همگرایی الگوریتم پیشنهادی برای مجموعه پوشش راسی براساس حاصلضرب بردار احتمال برای ۱۰۰ قدم از اجرای الگوریتم به طور متوسط در شکل ۲ نمایش داده شده است.

(۵) نتیجه‌گیری

در این مقاله یک الگوریتم جدید برای مسئله پوشش مجموعه با استفاده از اتوماتای یادگیر توزیع شده ارائه شد. چنانچه که ذکر شد با توجه به پیچیدگی زمانی این مسئله، الگوریتم‌های هیوریستیک با شروع از یک مجموعه تصادفی از موفقیت بیشتری برخوردار بوده‌اند. در الگوریتم پیشنهادی با استفاده از شبکه‌ای از اتوماتاهای یادگیر، در هر گام مجموعه پوشش ایجاد شده و با استفاده از مکانیزم یادگیری، راه‌حل‌ها بهبود یافته و نتایج مطلوبی برای پوشش مجموعه در حالت راسی بدست آمده است. با توجه پیچیدگی نمایی مسئله پوشش، نمی‌توان در یک زمان منطقی، به جواب بهینه دست یافت، در نتایج آزمایشات نشان داده شد که الگوریتم پیشنهادی، نسبت به روش‌های دیگر از بهبود نسبی برخوردار است.

(۶) مراجع

- [1] R. M. Karp, "Reducibility among combinatorial problems," *Complex. Comput. Comput.*, vol. 40, no. 4, pp. 85–103, 1972.
- [2] G. Lee and A. T. Murray, "Maximal covering with network survivability requirements in wireless mesh networks," *Comput. Environ. Urban Syst.*, vol. 34, no. 1, pp. 49–57, 2010.
- [3] S. Bouamama, C. Blum, and A. Boukerram, "A population-based iterated greedy algorithm for the minimum weight vertex cover problem," *Appl. Soft Comput.*, vol. 12, no. 6, pp. 1632–1639, 2012.
- [4] P. Horak and K. McAvaney, "On covering vertices of a graph by trees," *Discrete Math.*, vol. 308, no. 19, pp. 4414–4418, 2008.
- [5] S. Chiba and S. Fujita, "Covering vertices by a specified number of disjoint cycles, edges and isolated vertices," *Discrete Math.*, vol. 313, no. 3, pp. 269–277, Feb. 2013.
- [6] S. Richter, M. Helmert, and C. Gretton, "A stochastic local search approach to vertex cover," in *KI 2007: Advances in Artificial Intelligence*, Springer, 2007, pp. 412–426.
- [7] E. Halperin, "Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs," *Siam J. Comput.*, vol. 31, no. 5, pp. 1608–1623, 2002.
- [8] C. E. Lemke, H. M. Salkin, and K. Spielberg, "Set covering by single-branch enumeration with linear-programming subproblems," *Oper. Res.*, vol. 19, no. 4, pp. 998–1022, 1971.
- [9] J. Cardinal, M. Karpinski, R. Schmied, and C. Viehmann, "Approximating subdense instances of covering problems," *Electron. Notes Discrete Math.*, vol. 37, pp. 297–302, 2011.

نمودار همگرایی الگوریتم پیشنهادی براساس تغییرات بردار احتمال اعمال برای گراف‌های مختلف در شکل ۳ نمایش داده شده است. میزان تغییرات احتمال اعمال برای میانگین ۱۰ بار اجرای مختلف به ازای ۱۰۰ چرخه نمایش داده شده است. با توجه به شکل ۳، سیر صعودی نمودار بردار احتمال نشان دهنده همگرایی به عمل بهینه است. نکته قابل توجه در این نمودارها همگرایی سریع الگوریتم پیشنهادی است. حال الگوریتم پیشنهادی در مقایسه با روش‌های معروف EWLS [۱۸] و EWCC [۶] در جداول ۲ و ۳ مورد مقایسه قرار گرفته است. در جدول ۲، تعداد رؤوس مجموعه پوشش مشخص کننده است. در جدول ۳، تعداد تکرار لازم برای رسیدن به پاسخ قابل قبول مشخص شده است.

جدول ۱: نتایج پوشش برای الگوریتم پیشنهادی و روش‌های دیگر

نام گراف	EWCC	EWLS	روش پیشنهادی
C500	۵۷	۵۷	۳
Dsjc500	۱۳	۱۳	۷
Mann27	۱۲۶	۱۲۶	۲
P_hat3001	۸	۸	۱۷
P_hat3003	۳۶	۳۶	۴
P_hat7002	۴۴	۴۴	۱۶

جدول ۲: نتایج تکرار برای الگوریتم پیشنهادی و روش‌های دیگر

نام گراف	EWCC	EWLS	روش پیشنهادی
C500	۹۹۲۰۳	۱۱۵۹۸	۱۰۰
Dsjc500	۲۳۴۴	۳۱۷۹	۱۰۰
Mann27	۸۱۰۵	۷۱۵۵	۱۰۰
P_hat3001	۱۴۸	۲۱۵	۱۱۵
P_hat3003	۶۷۹	۷۹۴	۱۰۰
P_hat7002	۲۱۲	۲۲۲	۱۲۸

با توجه به جدول ۲ و ۳، الگوریتم پیشنهادی در مقایسه با الگوریتم‌های دیگر از موفقیت قابل توجهی از نظر اندازه پوشش و تعداد تکرار الگوریتم برخوردار است. نتایج جدول ۲ و ۳ و شکل ۳ حاکی از عملکرد مطلوب الگوریتم پیشنهادی در مقایسه با روش‌های EWLS، EWCC می‌باشد. چنانچه مشاهده می‌شود، در الگوریتم پیشنهادی به نسبت الگوریتم‌های دیگر تعداد نودهای پوشش‌دهنده بسیار پایین آمده که این نتیجه حاصل اجرای تکرار کمتری نسبت به الگوریتم‌های مذکور بوده است.

- [15] R. Jovanovic and M. Tuba, "An ant colony optimization algorithm with improved pheromone correction strategy for the minimum weight vertex cover problem," *Appl. Soft Comput.*, vol. 11, no. 8, pp. 5360–5366, 2011.
- [16] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*. Printice-Hall, 1989.
- [17] H. Beigy and M. R. Meybodi, "Utilizing distributed learning automata to solve stochastic shortest path problems," *Int. J. Uncertain.Fuzziness Knowl.Based Syst.*, vol. 14, no. 5, p. 591, 2006.
- [18] S. Cai, K. Su, and A. Sattar, "Local search with edge weighting and configuration checking heuristics for minimum vertex cover," *Artif. Intell.*, vol. 175, no. 9, pp. 1672–1696, 2011.
- [10] N. Bansal, A. Caprara, and M. Sviridenko, "A new approximation method for set covering problems, with applications to multidimensional bin packing," *Siam J. Comput.*, vol. 39, no. 4, pp. 1256–1278, 2009.
- [11] Z. Zhang, X. Gao, and W. Wu, "Algorithms for connected set cover problem and fault-tolerant connected set cover problem," *Theor. Comput.Sci.*, vol. 410, no. 8, pp. 812–817, 2009.
- [12] K. V. R. Kumar and D. Garg, "Choosing the Efficient Algorithm for Vertex Cover Problem," 2009.
- [13] D. S. Johnson, "Approximation algorithms for combinatorial problems," *J. Comput. Syst. Sci.*, vol. 9, no. 3, pp. 256–278, 1974.
- [14] J. E. Beasley and P. C. Chu, "A genetic algorithm for the set covering problem," *Eur. J. Oper. Res.*, vol. 94, no. 2, pp. 392–404, 1996.