

حل مسئله مکان یابی پوشش بیشینه با استفاده از اتوماتای یادگیر در شرایط تصادفی بودن فاصله نودها

مینا قوی پور^۱، محمدرضا میبدی^۲

^۱دانشکده کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران، mina_ghavipour@aut.ac.ir

^۲دانشکده کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران، mmeybodi@aut.ac.ir

چکیده - مسئله مکان یابی پوشش بیشینه یک مسئله NP_hard می باشد که هدف از آن، قرار دادن تعداد مشخصی سرویس دهنده در شبکه است به گونه ای که ماکزیمم جمعیت در یک شعاع سرویس از پیش تعیین شده، پوشش داده شوند. این مقاله مسئله پوشش بیشینه را در شرایط تصادفی مورد بررسی قرار می دهد. در این شرایط فرض می شود که فاصله نودها از یکدیگر، متغیرهای تصادفی با توزیع احتمال نامشخص است. راهکار پیشنهادی توسط مقاله از اتوماتای یادگیر ساختار متغیر برای حل مسئله کمک گرفته است. جهت ارزیابی کارایی الگوریتم پیشنهادی، از روش نمونه گیری استاندارد استفاده می شود. نتایج شبیه سازی ها نشان می دهد که این الگوریتم نسبت به روش نمونه گیری استاندارد با توجه به تعداد نمونه های گرفته شده، بسیار بهتر عمل کرده است.

کلید واژه - اتوماتای یادگیر، بیشینه پوشش، عدم قطعیت فاصله، گراف تصادفی، مکان یابی سرویس دهنده

به راه حل بهینه دست یابند. به این خاطر، در کاربردهای واقعی و برای مسائل بزرگ، از روش های مکاشفه ای استفاده می گردد. رسنده [2] از یک الگوریتم مکاشفه ای تحت عنوان رویه جستجوی تطبیقی تصادفی حریصانه^۲ برای حل مسئله مکان یابی پوشش بیشینه استفاده کرده است. این الگوریتم یک پیکربندی مکان خوب و نه الزاماً بهینه را برای مسئله می یابد. لورنا [3] و همکارش روش مکاشفه ای لاگرانژی/سارگیت^۳ ارائه شده برای حل مسئله p-میانه^۴ را برای حل مسئله MCLP بکار برده اند. آنها ضرایب فاصله در مدل خطی مسئله p-میانه را به گونه ای تطبیق دادند که نشان دهنده میزان تقاضای یک جمعیت باشند. با این روش، این امکان فراهم شد که بتوان از الگوریتم های p-میانه با حداقل تغییر، برای حل مسئله MCLP استفاده کرد. اکسیا و همکارانش [4] کارایی الگوریتم های مکاشفه ای: افزودن حریصانه^۵، الگوریتم تعویض^۶، الگوریتم ژنتیک، الگوریتم جستجوی مبتنی بر منع^۷ و الگوریتم بازپخت شبیه سازی شده^۸ برای حل مسئله مکان یابی پوشش بیشینه را مورد بررسی قرار دادند.

۱- مقدمه

مسئله مکان یابی پوشش بیشینه^۱ (MCLP) برای اولین بار توسط چرچ و همکارش [1] مطرح شد. در این مسئله سعی بر آن است که با توجه به محدود بودن سرمایه و در اختیار داشتن تعداد مشخصی سرویس دهنده، بهترین پیکربندی که بیشترین پوشش را برای نودهای متقاضی سرویس فراهم می کند، یافت شود. این در حالیست که برای سرویس دهنده ها یک ماکزیمم فاصله پوشش هم در نظر گرفته می شود، به طوریکه نود متقاضی سرویس تنها در صورتی که در فاصله پوشش یک سرویس دهنده قرار داشته باشد، سرویس داده خواهد شد. به صورت کلی الگوریتم های ارائه شده برای حل مسئله مکان یابی پوشش بیشینه، به دو دسته الگوریتم های دقیق و الگوریتم های مکاشفه ای تقسیم می شوند. الگوریتم های دقیق که معمولاً از روش های برنامه ریزی خطی و انشعاب و برش کمک می گیرند، تنها برای مسائل کوچک می توانند در زمان قابل قبولی

مبتنی بر اتوماتای یادگیر را برای حل مسئله MCLP در شرایط تصادفی ارائه می کند. جهت ارزیابی کارایی این الگوریتم از تکنیک نمونه گیری استاندارد استفاده می شود. همانطور که نشان داده خواهد شد، الگوریتم پیشنهادی با توجه به نرخ نمونه گیری بهتر از روش نمونه گیری استاندارد عمل می کند.

مقاله به صورت زیر سازماندهی شده است. فرمول بندی ریاضی از مسئله MCLP در بخش بعد ارائه می گردد. در بخش ۲ معرفی مختصری از اتوماتای یادگیر خواهیم داشت. الگوریتم پیشنهادی برای حل مسئله پوشش بیشینه تصادفی و نتایج شبیه سازی به ترتیب در بخش های ۳ و ۴ ارائه شده و بخش آخر نتیجه گیری مقاله می باشد.

۲- فرمول بندی مسئله

فرض کنید $G = \langle V, E, W \rangle$ یک گراف تصادفی باشد که در آن، V مجموعه رئوس، E مجموعه یال ها و W وزن یال های گراف را شامل می شود. به طوریکه وزن یال (i, j) یک متغیر تصادفی مثبت با تابع توزیع احتمال w_{ij} می باشد. از طرفی، به هر یک از نودهای گراف، یک وزن مثبت اختصاص داده شده است. مسئله مکان یابی پوشش بیشینه با وزن یال تصادفی به صورت زیر فرموله می شود:

$$\max \sum_{i=1}^n a_i y_i \quad (1)$$

$$\text{S.T.} \sum_{j=1}^n x_j = P$$

$$\sum_{\{j: w_{ij} \leq S\}} x_j \geq y_i \quad i \in V$$

$$x_j, y_i \in \{0, 1\} \quad \forall i, j \in V$$

به طوریکه n تعداد نودهای گراف، P تعداد از پیش تعیین شده سرویس دهنده ها، S شعاع پوشش مطلوب و a_i نشان دهنده وزن نود i ام می باشد. مقدار مورد انتظار وزن یال بین نود i ام و سرویس دهنده j ام با w_{ij} نشان داده می شود.

مسئله MCLP در هر دو حوزه خصوصی و عمومی کاربرد دارد. به عنوان مثال، در بخش خصوصی می توان از این مسئله برای تعیین مکان شعبه های بانک [5] و در بخش عمومی برای تعیین مکان بیمارستان ها [6]، محل استقرار آمبولانس ها [7] و غیره استفاده کرد. در بسیاری از کاربردهای دنیای واقعی بر حسب نیاز، از محدودیت های مسئله پوشش بیشینه کاسته شده و به این ترتیب مدل های جدیدی از این مسئله ارائه گردیده است. در مدل MCLP با پوشش جزئی با استفاده از یک تابع پوشش مشخص می شود که چه نسبتی از تقاضای نود متقاضی سرویس، که در یک فاصله خاص از سرویس دهنده قرار دارد، پوشش داده شود [8, 9]. در مدل پیشنهادی توسط برمن [10] و همکارانش برای آن دسته از نودهای متقاضی سرویس که مضر هستند و نباید پوششی برای آن ها فراهم گردد، وزن منفی در نظر گرفته شده است. به این ترتیب، از آنجا که هدف مسئله MCLP ماکزیمم کردن تقاضایی است که پوشش داده می شود، سرویس دهنده ها باید به گونه ای تعیین مکان شوند که حتی الامکان نودهای با وزن منفی در شعاع پوشش آن ها قرار نگیرند.

مسئله پوشش بیشینه در شرایط تصادفی هم مورد توجه قرار گرفته است. شاهنقی و همکارش شرایطی را در نظر گرفتند که در آن فاصله نودها از یکدیگر متغیر تصادفی با تابع چگالی نرمال باشد و یک الگوریتم هوشمند ترکیبی را برای حل آن ارائه کردند [11]. درنزر و همکارش [12] شعاع سرویس کامل و شعاع سرویس جزئی را در مسئله MCLP با پوشش جزئی به صورت متغیرهای تصادفی فرض کردند. سرا و همکارش [13] مسئله MCLP با عدم قطعیت تقاضا را با استفاده از روش سناریو کلاسیک مورد بررسی قرار دادند. در این روش، الگوهای مختلفی از تقاضا در سناریوهای متفاوت مطرح می شود.

در این مقاله مسئله پوشش بیشینه در شرایط تصادفی بودن فاصله نودها مورد توجه قرار گرفته است. برخلاف کارهای انجام شده تاکنون، در اینجا فرض می شود که فاصله نودها از یکدیگر، یک متغیر تصادفی با توزیع احتمال ناشناخته است و تنها نمونه هایی از این توزیع ها در اختیار می باشد. به این ترتیب، یک تکنیک یادگیری تقویتی مورد نیاز است که بتواند با تعامل با این محیط ناشناخته به حل بهینه دست یابد. این مقاله الگوریتمی

محیط، شامل تعداد متناهی از مقادیر در بازه $[a, b]$ باشد، در این صورت این مدل Q-Model نامیده می‌شود. در نهایت، اگر خروجی محیط متغیر تصادفی پیوسته با مقادیر در بازه $[a, b]$ باشد، به این مدل، S-Model گفته می‌شود.

اتوماتای یادگیر به دو دسته اتوماتای یادگیر ساختار ثابت و اتوماتای یادگیر ساختار متغیر تقسیم می‌شود. اتوماتای یادگیر با ساختار متغیر را می‌توان توسط یک ۴ تایی $\{\alpha, \beta, p, T\}$ نشان داد که در آن، α مجموعه‌ی خروجی‌های اتوماتا، β مجموعه‌ی ورودی‌های آن، p بردار احتمال انتخاب هر یک از اقدام‌ها و T الگوریتم یادگیری می‌باشد. ایده اصلی تمام الگوریتم‌های یادگیری به این صورت است که اگر اتوماتای یادگیر در تکرار n ام، اقدام α_i را انتخاب کرده و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ متناظر با آن اقدام افزایش یافته و احتمال سایر اقدام‌ها کاهش می‌یابد (معادله (۲)). بالعکس، در صورت نامطلوب بودن پاسخ محیط، احتمال اقدام α_i کاهش یافته و احتمال سایر اقدام‌های اتوماتا افزایش داده می‌شود (معادله (۳)).

$$p_i(n+1) = p_i(n) + a \cdot [1 - p_i(n)] \quad (2)$$

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

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

$$p_j(n+1) = \frac{b}{r-1} + (1 - b) \cdot p_j(n) \quad \forall j \neq i \quad (3)$$

در روابط بالا، a پارامتر پاداش و b پارامتر جریمه می‌باشد. اگر مقادیر a و b با یکدیگر برابر باشند، روش یادگیری L_{RP} نامیده می‌شود. در این شرایط، نرخ به روزرسانی بردار احتمال بدون توجه به پاسخ محیط، در هر لحظه از زمان یکسان می‌باشد. اگر پارامتر یادگیری b مساوی با صفر باشد، روش یادگیری L_{RI} نامیده می‌شود. این بدین معنی است که احتمال اقدام‌ها تنها در صورتی که پاسخ محیط مطلوب باشد، به روز خواهد شد. در نهایت اگر $a \ll b$ باشد، روش یادگیری L_{REP} نامیده خواهد شد.

اگر سرویس‌دهنده‌ای در نود j ام قرار دارد

$$x_j = \begin{cases} 1 \\ 0 \end{cases}$$

در غیر این صورت

اگر نود i ام پوشش داده می‌شود

$$y_i = \begin{cases} 1 \\ 0 \end{cases}$$

در غیر این صورت

هدف مسئله این است که تقاضای پوشش داده شده کل، در شرایطی که از تعداد P سرویس‌دهنده با شعاع پوشش مشخص S استفاده شده است، ماکزیمم گردد. بر اساس محدودیت اول دقیقاً P نود باید به عنوان سرویس‌دهنده انتخاب شوند. محدودیت دوم مشخص می‌کند که هر نود متقاضی سرویس تنها در صورتی پوشش داده می‌شود که سرویس‌دهنده‌ای در فاصله کمتر مساوی S نسبت به آن قرار داشته باشد. محدودیت سوم متغیرهای x_j, y_i را محدود به داشتن مقدار دودویی می‌کند.

۳- اتوماتای یادگیر

اتوماتای یادگیر تصادفی^۹ یک واحد تصمیم‌گیری تطبیقی و ابزاری نیرومند در حل مسائل بهینه‌سازی محسوب می‌شود- [14] [19]. اتوماتای تصادفی تلاش می‌کند که بدون داشتن هیچگونه اطلاعاتی از اقدام بهینه، راه‌حلی برای مسئله بیابد. بر این اساس، اتوماتا اقدامی را به صورت تصادفی انتخاب و بر محیط اعمال می‌کند. پس از دریافت پاسخ محیط، اتوماتا احتمال اقدام‌های خود را به روز کرده و این فرآیند تا یافتن عمل بهینه تکرار می‌شود. اتوماتای تصادفی که به این صورت کارایی خود را بهبود می‌بخشد، اتوماتای یادگیر نامیده می‌شود [20].

محیط توسط یک سه‌تایی $\{\alpha, \beta, c\}$ نشان داده می‌شود که در آن، α مجموعه ورودی‌های محیط، β مجموعه خروجی‌های آن و c مجموعه احتمالات جریمه می‌باشد، به طوریکه هر c_i متناظر با یک اقدام α_i از مجموعه اعمال است. مدل‌های مختلفی بر مبنای پاسخ محیط تعریف می‌شود. در P-Model، محیط مقادیر صفر و یک را به عنوان خروجی اختیار می‌کند. صفر به معنی پاداش و یک به معنی جریمه است. اگر مجموعه خروجی

۴- الگوریتم پیشنهادی

محاسبه و New_{sol} نامیده می شود.

گام سوم: مقدار New_{sol} و Cur_{sol} مقایسه می شود و اگر بهبودی حاصل شده بود، عمل انتخابی اتوماتای فعال LA_i پاداش گرفته و راه حل جدید به عنوان راه حل جاری در نظر گرفته می شود. در غیر این صورت، اتوماتای LA_i جریمه می شود.

گام چهارم: اگر DC نشان دهنده مجموعه نودهای سرویس دهنده در راه حل جاری، Γ مجموعه اتوماتاهای فعال و $p_i(\alpha'_i)$ احتمال اقدام انتخابی توسط اتوماتای LA_i باشد، در این صورت، احتمال اینکه مجموعه DC در تکرارهای بعدی تغییر نکند یعنی هریک از اتوماتاها در صورت فعال شدن، مجدداً خودش را برای قرارگیری سرویس دهنده کاندید نماید، به صورت زیر محاسبه می شود:

$$Prob_{DC} = \frac{1}{P} \sum_{LA_i \in \Gamma} p_i(\alpha'_i = v_i) \quad (5)$$

گام پنجم: گام های دو تا چهار تکرار می شوند تا زمانیکه مقدار $Prob_{DC}$ بزرگتر یا مساوی مقدار آستانه τ شود و یا تعداد تکرارهای الگوریتم بیشتر یا مساوی مقدار Max_{iter} گردد.

۵- نتایج شبیه سازی

جهت ارزیابی کارایی الگوریتم پیشنهادی، آزمایشات بر روی گراف های تصادفی Alex2-B و Alex3-B که از منبع [21] اقتباس شده اند، انجام گرفت. نمودار این گراف ها در شکل ۱ و ۲ و تابع توزیع احتمال وزن یال های آن ها در جداول ۱ و ۲ قابل مشاهده می باشد. وزن رئوس گراف ها به صورت تصادفی با توزیع یکنواخت در بازه بسته [1,3] تولید شده است.

الگوریتم یادگیری که برای به روزرسانی بردار احتمال اقدام های اتوماتاها استفاده شده است، الگوریتم L_{RI} می باشد. همچنین مثال در نظر گرفته شده بر روی هر دو گراف جهت آزمایش الگوریتم، مکان یابی تعداد ۲ سرویس دهنده با شعاع سرویس مشخص ۱۴ است به گونه ای که ماکزیمم پوشش فراهم شود. جهت ارزیابی کارایی الگوریتم پیشنهادی، تعداد نمونه های اخذ شده توسط این الگوریتم با تعداد نمونه های مورد نیاز به روش نمونه گیری استاندارد مقایسه می شود. روش نمونه گیری

در این بخش، الگوریتم پیشنهادی برای حل مسئله MCLP بر روی گراف های تصادفی ارائه می شود. ورودی الگوریتم، گراف تصادفی وزن دار $G = \langle V, E, W \rangle$ و پیکربندی با پوشش بیشینه خروجی الگوریتم را تشکیل می دهد. در گراف G ، متناظر با هر نود v_i یک اتوماتای یادگیر ساختار متغیر LA_i در نظر گرفته می شود که تعداد اقدام های آن برابر با $deg(v_i) + 1$ می باشد. به بیانی دیگر، مجموعه اقدام های اتوماتای LA_i شامل یک اقدام متناظر با نود v_i و یک اقدام متناظر با هریک از نودهای همسایه v_i می باشد:

$$\alpha_i = \{v_j | \forall j; j = i \text{ or } (v_i, v_j) \in E\} \quad (4)$$

در ابتدای کار، احتمال تمامی اقدام های اتوماتای LA_i یکسان و برابر با $1/(deg(v_i) + 1)$ در نظر گرفته می شود. مراحل الگوریتم پیشنهادی ۱ به شرح زیر است:

گام اول: ابتدا از هر یال e_{ij} از گراف G یک نمونه گرفته می شود. نمونه گرفته شده مقدار اولیه w_{ij} را تشکیل می دهد. سپس، تعداد P نود به تصادف به عنوان سرویس دهنده انتخاب می گردد. به ازای هر سرویس دهنده، مجموع وزن نودهایی از گراف که توسط آن سرویس دهنده پوشش داده می شوند، محاسبه می شود. مجموع پوشش فراهم شده توسط این P سرویس دهنده، پوشش پیکربندی جاری را تشکیل می دهد که با Cur_{sol} نشان داده می شود.

گام دوم: یکی از اتوماتاهایی که نود متناظرشان در راه حل جاری سرویس دهنده می باشد، به صورت تصادفی انتخاب و فعال می شود. اتوماتای فعال شده (LA_i) از بین نودهای همسایه که سرویس دهنده نیستند به علاوه نود v_i ، که اقدام های فعال اتوماتا را تشکیل می دهند، یکی (v_j) را با توجه به بردار احتمال مربوطه انتخاب و به عنوان کاندیدی برای قرارگیری سرویس دهنده پیشنهاد می کند. از یال (v_i, v_j) نمونه گرفته شده و مقدار w_{ij} و در نتیجه Cur_{sol} به روز می شود. نود v_j به همراه سایر سرویس دهنده های راه حل جاری به استثنای نود v_i ، یک پیکربندی جدید را فراهم می کند که به عنوان راه حل جدید در نظر گرفته می شود. مقدار پوشش فراهم شده توسط راه حل جدید

همه یال های گراف در نظر گرفته می شود، به طوریکه $\varepsilon_0 = \varepsilon/k$

جدول ۱: تابع توزیع احتمال وزن یال های گراف تصادفی G_1

یال	وزن	احتمال
e_1	{2, 8, 12}	{0.9, 0.08, 0.02}
e_2	{6, 18, 21}	{0.88, 0.1, 0.02}
e_3	{17, 35, 50}	{0.75, 0.2, 0.05}
e_4	{3, 7, 10}	{0.68, 0.25, 0.07}
e_5	{9, 13, 20}	{0.82, 0.15, 0.03}
e_6	{21, 33, 45}	{0.85, 0.09, 0.06}
e_7	{18, 27, 36}	{0.94, 0.05, 0.01}
e_8	{8, 14, 18}	{0.77, 0.13, 0.10}
e_9	{15, 21, 25}	{0.76, 0.22, 0.02}
e_{10}	{25, 38, 50}	{0.8, 0.12, 0.08}
e_{11}	{20, 30, 40}	{0.87, 0.09, 0.04}
e_{12}	{10, 17, 30}	{0.95, 0.03, 0.02}
e_{13}	{4, 19, 15}	{0.75, 0.14, 0.11}
e_{14}	{15, 28, 45}	{0.79, 0.15, 0.06}
e_{15}	{8, 14, 25}	{0.85, 0.1, 0.05}

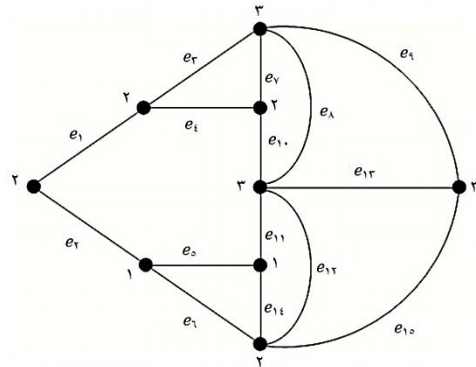
جدول ۲: تابع توزیع احتمال وزن یال های گراف تصادفی G_2

یال	وزن	احتمال
e_1	{2, 8, 12}	{0.9, 0.08, 0.02}
e_2	{10, 24, 35}	{0.85, 0.12, 0.03}
e_3	{6, 18, 24}	{0.88, 0.1, 0.02}
e_4	{17, 35, 50}	{0.75, 0.2, 0.05}
e_5	{3, 7, 10}	{0.68, 0.25, 0.07}
e_6	{12, 22, 30}	{0.85, 0.11, 0.04}
e_7	{10, 19, 24}	{0.80, 0.14, 0.06}
e_8	{4, 6, 10}	{0.75, 0.17, 0.08}
e_9	{3, 7, 12}	{0.84, 0.13, 0.03}
e_{10}	{21, 33, 45}	{0.85, 0.09, 0.06}
e_{11}	{8, 14, 18}	{0.77, 0.13, 0.10}
e_{12}	{15, 21, 25}	{0.76, 0.22, 0.02}
e_{13}	{5, 10, 12}	{0.65, 0.23, 0.12}
e_{14}	{18, 27, 36}	{0.94, 0.05, 0.01}
e_{15}	{25, 38, 50}	{0.8, 0.12, 0.08}
e_{16}	{20, 30, 40}	{0.87, 0.09, 0.04}
e_{17}	{4, 19, 15}	{0.75, 0.14, 0.11}
e_{18}	{9, 13, 20}	{0.82, 0.15, 0.03}
e_{19}	{15, 28, 45}	{0.79, 0.15, 0.06}
e_{20}	{10, 17, 30}	{0.95, 0.03, 0.02}
e_{21}	{8, 14, 25}	{0.85, 0.1, 0.05}

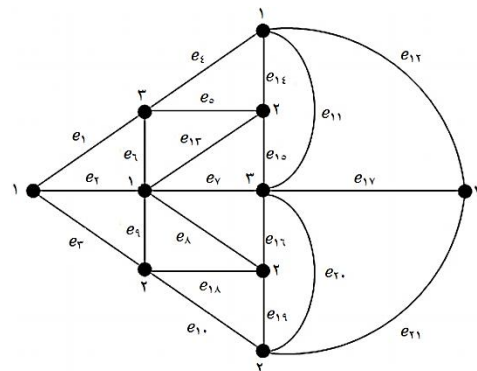
استاندارد، حداقل تعداد نمونه های مورد نیاز از هر یال گراف تصادفی را مشخص می کند به طوریکه با احتمال $1 - \varepsilon$ ، انحراف میانگین نمونه های اخذ شده از یال گراف از مقدار مورد انتظار وزن آن، کمتر از مقدار δ باشد. به بیانی دیگر:

$$\forall \delta \exists n_0 \forall n \geq n_0 \quad p\{|\bar{x}_n - \mu| < \delta\} > 1 - \varepsilon \quad (6)$$

که در آن، ε پارامتر خطا، δ یک عدد مثبت به اندازه کافی کوچک، μ و \bar{x}_n به ترتیب میانگین واقعی و میانگین نمونه ها می باشند. مقدار δ در تمامی آزمایشات برابر با ۰.۰۰۱ در نظر گرفته شده است.



شکل ۱: گراف تصادفی G_1 با وزن یال تصادفی [21]



شکل ۲: گراف تصادفی G_2 با وزن یال تصادفی [21]

حداقل تعداد نمونه های مورد نیاز از هر یک از یال های هر دو گراف Alex2-B و Alex3-B به ازای سطوح اطمینان ۰.۵ تا ۰.۹۹ به روش نمونه گیری استاندارد تعیین شده و به ترتیب در جداول ۳ و ۴ قابل مشاهده می باشد.

در صورتی که بخواهیم یک سطح اطمینان $1 - \varepsilon$ را برای راه حل بهینه داشته باشیم، نیاز به یک سطح اطمینان $1 - \varepsilon_i$ برای هر یال e_i گراف داریم به طوریکه $\sum_{i=1}^k \varepsilon_i = \varepsilon$. که در آن، k نشان دهنده تعداد یال های پوشش داده شده در راه حل بهینه می باشد. بر این اساس، یک سطح اطمینان یکسان $1 - \varepsilon_0$ برای

به ازای مقادیر مختلف پارامتر یادگیری، متوسط تعداد کل نمونه های گرفته شده از گراف، متوسط تعداد نمونه های گرفته شده از یال های بین سرویس دهنده ها و نودهای پوشش داده شده در راه حل بهینه، متوسط تعداد دفعات همگرا شدن الگوریتم به راه حل بهینه و متوسط تعداد تکرارهای الگوریتم برای ۱۰۰ بار اجرای مستقل، برای هر دو گراف در جداول ۵ و ۶ آورده شده است. همچنین، تعداد کل نمونه های مورد نیاز از هر یک از گراف ها در روش نمونه گیری استاندارد نیز به ازای سطوح اطمینان مختلف در همین جداول قابل مشاهده می باشد. در شبیه سازی ها مقدار τ برابر با ۰.۹ و مقدار Max_{iter} برابر با ۵۰۰۰۰ در نظر گرفته شده است.

همانطور که از نتایج شبیه سازی می توان فهمید، کاهش مقدار پارامتر یادگیری موجب افزایش تعداد تکرارهای الگوریتم و در نتیجه افزایش تعداد نمونه گیری های صورت گرفته از گراف شده است. تعداد نمونه های بیشتر، نرخ همگرایی بالاتری را به دنبال دارد. همانطور که مشاهده می کنید در گراف G_1 نرخ همگرایی الگوریتم به ازای تعداد کل ۳۵۶ نمونه به ۹۸٪ رسیده است. اما متأسفانه در این مقدار ثابت مانده و کاهش مقدار پارامتر یادگیری نتوانسته است تغییری در آن ایجاد کند و از یک جایی به بعد (به ازای پارامتر یادگیری ۰.۰۰۰۵) تعداد نمونه های الگوریتم از روش نمونه گیری استاندارد بیشتر شده است. با توجه به شرط خاتمه الگوریتم، می توان نتیجه گرفت که در ۲٪ مواقع الگوریتم بر روی این گراف در بهینه محلی متوقف می شود. در گراف G_2 ، الگوریتم توانسته است با تعداد کل ۴۰۹ نمونه به نرخ همگرایی ۱۰۰٪ دست یابد. این درحالیست که تعداد نمونه های مورد نیاز در روش نمونه گیری استاندارد بر روی این گراف، ۱۰۳۴۳ نمونه برای همگرایی ۹۹٪ می باشد. به این ترتیب، به صورت کلی می توان گفت که الگوریتم بر روی هر دو گراف موفق شده است که با تعداد نمونه های بسیار کمتری نسبت به روش نمونه گیری استاندارد، به نرخ های همگرایی بالایی دست یابد.

۶- نتیجه گیری

در این مقاله مسئله مکان یابی پوشش بیشینه در شرایطی که وزن یال ها متغیرهای تصادفی با توزیع نامشخص باشند، مورد

جدول ۳: حداقل تعداد نمونه های مورد نیاز از یال های گراف G_1 در روش نمونه گیری استاندارد

سطح اطمینان									yal
0.99	0.95	0.9	0.85	0.8	0.75	0.7	0.6	0.5	
295	271	347	228	176	196	202	229	187	e_1
240	312	203	216	258	297	282	296	224	e_2
479	382	358	416	438	434	527	375	476	e_3
286	206	162	176	253	211	223	196	192	e_4
281	268	166	210	233	184	181	246	184	e_5
306	270	334	265	256	329	307	308	342	e_6
326	197	194	216	187	252	112	159	238	e_7
286	224	321	230	252	189	199	197	207	e_8
262	227	214	219	216	188	265	249	182	e_9
445	409	417	440	378	419	225	336	429	e_{10}
266	353	281	298	332	343	215	201	204	e_{11}
216	254	233	279	210	171	237	173	183	e_{12}
382	280	346	329	285	317	395	311	241	e_{13}
313	441	344	354	351	280	429	283	289	e_{14}
272	204	256	283	289	235	228	279	206	e_{15}
4655	4298	4176	4159	4114	4045	4027	3838	3784	جمع کل

جدول ۴: حداقل تعداد نمونه های مورد نیاز از یال های گراف G_2 در روش نمونه گیری استاندارد

سطح اطمینان									yal
0.99	0.95	0.9	0.85	0.8	0.75	0.7	0.6	0.5	
524	449	471	249	275	394	288	273	315	e_1
608	499	473	552	481	397	519	473	459	e_2
481	434	401	344	293	436	338	401	460	e_3
479	376	401	476	395	413	351	514	346	e_4
670	869	679	580	482	601	551	557	751	e_5
464	452	286	320	372	268	284	304	364	e_6
441	384	492	459	326	328	292	365	359	e_7
412	385	353	318	319	248	240	268	279	e_8
516	392	390	544	448	386	438	421	390	e_9
658	446	428	407	352	405	467	392	304	e_{10}
424	491	342	364	479	390	333	317	229	e_{11}
463	550	406	428	438	580	490	410	524	e_{12}
456	388	302	347	426	371	265	294	305	e_{13}
500	388	434	353	377	337	443	410	400	e_{14}
459	378	483	466	372	333	357	248	293	e_{15}
593	555	509	482	552	474	692	507	705	e_{16}
463	533	512	441	504	529	434	439	398	e_{17}
393	442	444	434	340	347	342	254	327	e_{18}
385	454	420	509	464	469	351	412	503	e_{19}
546	536	666	556	505	706	530	765	704	e_{20}
400	421	410	343	373	494	463	400	337	e_{21}
10343	9829	9310	8981	8582	8917	8478	8433	8761	جمع کل

جدول ۶: متوسط تعداد کل نمونه‌ها، متوسط تعداد نمونه‌های گرفته شده از راه حل بهینه، متوسط تعداد تکرارها و نرخ همگرایی الگوریتم پیشنهادی و تعداد کل نمونه‌های مورد نیاز در روش نمونه‌گیری استاندارد از گراف G_2

روش نمونه‌گیری استاندارد		الگوریتم پیشنهادی				
تعداد کل نمونه‌ها	سطح اطمینان	میانگین تعداد نمونه‌ها از MCLP	میانگین تعداد کل نمونه‌ها	نرخ همگرایی	تعداد تکرارها	پارامتر یادگیری
8761	۰.۵	141	380	87	39244	۰.۰۹
8433	۰.۶	121	321	89	40266	۰.۰۸
8478	۰.۷	98	236	91	41153	۰.۰۷
8917	۰.۷۵	124	247	92	41959	۰.۰۶
8582	۰.۸	125	232	94	43076	۰.۰۵
8981	۰.۸۵	99	177	96	44645	۰.۰۴
9310	۰.۹	120	189	98	46438	۰.۰۳
9829	۰.۹۵	163	230	99	48086	۰.۰۲
10343	۰.۹۹	307	409	100	49569	۰.۰۱

- [2] T. Drezner, Z. Drezner and Z. Goldstein, "A stochastic gradual cover location problem," *Naval Research Logistics*, vol. 57, pp. 367-372, 2010.
- [3] D. Serra, S. Ratick and C. ReVelle, "The maximum capture problem with uncertainty," *Environ. Planning B: Planning Des.*, Vol. 23, pp. 49-59, 1996.
- [4] J. Akbari Torkestani and M.R. Meybodi, "Finding minimum weight connected dominating set in stochastic graph based on learning automata," *Information Sciences*, Vol. 200, pp. 57-77, 2012.
- [5] J. Akbari Torkestani and M.R. Meybodi, "Learning automata based algorithms for solving stochastic minimum spanning tree problem," *Journal of Applied Soft Computing*, Vol. 11, pp. 4064-4077, 2011.
- [6] J. Akbari Torkestani and M.R. Meybodi, "Learning automata-based algorithms for finding minimum weakly connected dominating set in stochastic graphs," *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, Vol. 18, pp. 721-758, 2010.
- [7] J. Akbari Torkestani and M.R. Meybodi, "A new vertex coloring algorithm based on variable action-set learning automata," *Journal of Computing and Informatics*, Vol. 29, pp. 1001-1020, 2010.
- [8] H. Beigy and M.R. Meybodi, "Utilizing distributed learning automata to solve stochastic shortest path problems," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 14, pp. 591-615, 2006.
- [9] B.J. Oommen and E.V. de St. Croix, "Graph partitioning using learning automata," *IEEE Transactions on Computers*, vol. 45, pp. 195-208, 1995.
- [10] K.S. Narendra and M.A.L. Thathachar, *Learning automata: an introduction*, Prentice-Hall, New York, 1989.
- [11] K.R. Hutson and D.R. Shier, "Bounding distributions for the weight of a minimum spanning tree in stochastic networks," *Operations Research*, vol. 53, pp. 879-886, 2005.

توجه قرار گرفت و الگوریتمی مبتنی بر اتوماتای یادگیر برای حل آن ارائه گردید. به منظور ارزیابی کارایی الگوریتم پیشنهادی از روش نمونه‌گیری استاندارد استفاده شد. مقایسه این الگوریتم با روش نمونه‌گیری استاندارد از نقطه نظر تعداد نمونه‌گیری نشان داد که این الگوریتم توانایی رسیدن به نرخ‌های همگرایی بالا با تعداد نمونه‌های بسیار کمتر از روش نمونه‌گیری استاندارد را دارا می‌باشد.

- ¹ Maximal covering location problem
² Greedy randomized adaptive search procedure (GRASP)
³ Lagrangian/Surrogate constraint
⁴ p-median problem
⁵ Greedy-add algorithm
⁶ Interchange algorithm
⁷ Tabu search algorithm (TS)
⁸ Simulated annealing algorithm (SA)
⁹ Stochastic learning automata
¹⁰ Linear reward-penalty
¹¹ Linear reward-inaction
¹² Linear reward-epsilon Penalty

مراجع

- [1] R.L. Church and C.S. ReVelle, "The maximal covering location problem," *Papers in Regional Science*, Vol. 32, pp. 101-118, 1974.

جدول ۵: متوسط تعداد کل نمونه‌ها، متوسط تعداد نمونه‌های گرفته شده از راه‌حل بهینه، متوسط تعداد تکرارها و نرخ همگرایی الگوریتم پیشنهادی و تعداد کل نمونه‌های مورد نیاز در روش نمونه‌گیری استاندارد از گراف G_1

روش نمونه‌گیری استاندارد		الگوریتم پیشنهادی				
تعداد کل نمونه‌ها	سطح اطمینان	میانگین تعداد نمونه‌ها از MCLP	میانگین تعداد کل نمونه‌ها	نرخ همگرایی	تعداد تکرارها	پارامتر یادگیری
3784	۰.۵	۲۱۶	۳۵۶	۹۸	۸۱۲	۰.۰۰۹
3838	۰.۶	۲۷۵	۴۵۰	۹۸	۱۰۴۰	۰.۰۰۷
4027	۰.۷	۳۸۱	۶۱۹	۹۸	۱۴۴۸	۰.۰۰۵
4045	۰.۷۵	۶۲۸	۱۰۱۴	۹۸	۲۴۰۱	۰.۰۰۳
4114	۰.۸	۱۸۶۴	۲۹۸۸	۹۸	۷۱۶۵	۰.۰۰۱
4159	۰.۸۵	۲۰۶۹	۳۳۱۷	۹۸	۷۹۵۹	۰.۰۰۰۹
4176	۰.۹	۲۶۵۸	۴۲۵۶	۹۸	۱۰۲۲۶	۰.۰۰۰۷
4298	۰.۹۵	۳۷۱۴	۵۹۴۸	۹۸	۱۴۳۱۱	۰.۰۰۰۵
4655	۰.۹۹	۶۱۹۴	۹۸۹۹	۹۸	۲۳۸۴۳	۰.۰۰۰۳

- [12] M.G.C. Resende, "Computing approximate solutions of the maximum covering problem with GRASP", *Journal of Heuristics*, vol. 4, pp. 161-177, 1998.
- [13] L.A.N. Lorena and M.A. Pereira, "A lagrangean/surrogate heuristic for the maximal covering location problem using hillsman's edition," *International Journal of Industrial Engineering*, vol. 9, pp. 57-67, 2002.
- [14] L. Xia, M. Xie, W. Xu, J. Shao, W. Yin and J. Dong, "An empirical comparison of five efficient heuristics for maximal covering location problems," *IEEE International Conference on Service Operations, Logistics and Informatics Chicago, IL*, 2009.
- [15] J.T. Pastor, "Bicriterion programs and managerial locations: application to the banking sector," *Journal of the Operational Research Society*, vol. 45, pp. 1351-1362, 1994.
- [16] J.F. Alegre, A. Avarez, S. Casado and J.A. Pacheco, "Use of memetic algorithms to solve a stochastic location model: health resources for diabetics in some provinces of castilla-león," *Rect@*, vol. Actas_13, p. 20, 2005.
- [17] D. Eaton, M. Hector, V. Sanchez, R. Latingua and J. Morgan, "Determining ambulance deployment in santo domingo, dominican republic," *Journal of the Operational Research Society*, vol. 37, pp. 113-126, 1986.
- [18] O. Karasakal and E.K. Karasakal, "A maximal covering location model in the presence of partial coverage," *Computers and Operations Research*, vol. 31, pp. 1515-1526, 2004.
- [19] O. Berman and D. Krass, "The generalized maximal covering location problem," *Computers and Operations Research*, vol. 29, pp. 563-591, 2002.
- [20] O. Berman, Z. Drezner and G.O. Wesolowsky, "The maximal covering problem with some negative weights," *Geographical Analysis*, vol. 41, pp. 30-42, 2009.
- [21] K. Shahanaghi and V.R. Ghezavati, "Efficient solution procedure to develop maximal covering location problem under uncertainty (using ga and simulation)", *International Journal of Industrial Eng. & Production Research*, vol. 19, pp. 21-29, 2008.