سیزدهمین کنفرانس ملی انجمن کامپیوتر ایران
جزیره کیش، خلیج فارس، ایران ۱۹ الی ۲۱ اسفند ۱۳۸۶
انجمن کامپیوتر ایران
Computer Society of Iran
دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر
پوستر

تعیین درخت پوشای کمینه در گرافهای تصادفی بکمک اتوماتای یادگیر توزیع شده

جواد اکبری ترکستا نی
دانشکدهٔ مهندسی کامپیوتر
دانشگاه آزاد اسلامی
اراک ایران

محمد رضا میبدی
دانشکدهٔ مهندسی کامپیوتر و فن آوری اطلاعات
دانشگاه صنعتی امیر کبیر
تهران ایران

چکیده: در این مقاله یک الگوریتم مبتنی بر اتوماتای یادگیر توزیع شده بمنظور تعیین درخت پوشای کمینهٔ یک گراف تصادفی که تابع توزیع وزن یالهای آن از قبل شناخته شده نمی باشند پیشنهاد می گردد. در این الگوریتم، سعی می شود با حداقل تعداد نمونه گیری از یالهای گراف درخت پوشای کمینه شناسایی شود. با انتخاب مناسب پارامترهای اتوماتای یادگیر توزیع شده الگوریتم پیشنهادی قادر است درخت پوشای کمینه را با احتمالی نزدیک به یک انتخاب نماید. به منظور ارزیابی الگوریتم پیشنهادی، تعداد نمونه های گرفته شده توسط الگوریتم پیشنهادی با تعداد نمونه های موردنیاز به روش نمونه گیری استاندارد مقایسه شده است. آزمایشها نشان داده است که تعداد نمونه های گرفته شده توسط الگوریتم پیشنهادی بمراتب از تعداد نمونه های گرفته شده به روش نمونه گیری استاندارد کمتر است.

کلمات کلیدی: گراف تصادفی، درخت پوشای کمینه، ، اتوماتاهای یادگیر، اتوماتای یادگیر توزیع شده

۱. مقدمه

یک گراف تصادفی[1] را می توان بوسیلهٔ سه تایی $G=<\mathrm{V},\mathrm{E},F>$ تعریف کرد بگونه ای که در آن $\mathbf{V}=\{v_1,v_2,...,v_n\}$ مجموعهٔ رئوس گراف و $\mathbf{E}\subseteq\mathbf{V}\times\mathbf{V}$ مجموعهٔ یالهای گراف در نظر گرفته شود. ماتریس $F_{n\times n}$ ماتریس توزیعهای احتمالی برای مشخصه ای از یالهای گراف میباشد. درایهٔ f_{ij} از این ماتریس توزیع احتمالی[2] مقادیر مشخصهٔ یال (v_i,v_j) می باشد. زیرگراف تصادفی $G'=<\mathrm{V}',E',F>$ را درخت پوشای تصادفی گراف G میگوییم اگر این زیر گراف یک گراف تصادفی متصل از گراف G بطوریکه $\mathrm{V}'=\mathrm{V}$ و $\mathrm{E}'\subseteq\mathrm{E}$ باشد. اگر $T=\{t_1,t_2,t_3,\mathbf{K}\}$ مجموعهٔ درخت های پوشای گراف تصادفی G و $\overline{L}\,t_i$ متوسط وزن درخت پوشای $t_i\in T$ باشد در این صورت، درخت پوشای $t^*\in T$ درخت پوشای کمینهٔ تصادفی[3] گراف G نامیده میشود اگر[11] $\overline{L}\,t^*=\min_{\forall t_i\in T}\{\overline{L}\,t_i\}$.

روشهای مختلفی مانند روش حریصانه و برنامه سازی پویا برای حل مسالهٔ کوتاه ترین در خت پوشا در یک گراف قطعی[4] ارایه گردیده است. الگوریتم پریم و کراسکال میتوانند مسالهٔ درخت پوشای کمینه در یک گراف قطعی را در زمان چندجمله ای[5] حل نمایند[9,10]. اما هنگامی که در یک شبکهٔ تصادفی وزن هر یک از یالهای گراف یک متغیر تصادفی گسسته با مؤلفه های آماری نامعین باشد تعیین درخت پوشای کمینه کار چندان آسانی نمی باشد. در این شرایط عامل یادگیر بایستی در یک محیط غیر قطعی و نامعین طی یک فرایند یادگیری پویا و تکرار پذیر نسبت به شناسایی مشخصه های آماری محیط و توزیع حاکم بر وزن یالهای گراف اقدام نماید و بر اساس توزیع و متوسط وزنی یالها، درخت پوشای کمینه را شناسایی نماید[12, 13].

الگوریتم پیشنهادی، ابتدا بکمک روش نمونه گیری استاندارد، متوسط تعداد نمونه های اخذ شده از هر یال گراف مشروط به آنکه پارامتر خطای نمونه گیری در یک بازهٔ اطمینان از مقدار معینی کمتر باشد، تعیین می گردد. سپس نشان داده می شود که متوسط مجموع دفعات نمونه گیری از گراف در الگوریتم پیشنهادی، همواره به مقدار قابل توجهی از متوسط تعداد نمونه های اخذ شده در روش نمونه گیری استاندارد کمتر است. در ادامهٔ این مقاله و در بخش ۲ به ترتیب اتوماتاهای یادگیر و اتوماتاهای یادگیر توزیع شده به اختصار شرح داده میشود. در بخش ۳ الگوریتم پیشنهادی و در بخش ۴ نیز نتایج آزمایشها آمده است. بخش ۵ نتیجه گیری مقاله می باشد.

---

1 Stochastic Graph
2 Probability Density Function
3 Stochastic Minimal Spanning Tree
4 Deterministic Graph
5 Polynomial Time

۲. اتوماتاهای یادگیر[۶]

اتوماتای یادگیر[4,5,7,8]، ماشینی است که می‌تواند تعدادی متناهی عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می‌شود و نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به اتوماتا داده می‌شود و اتوماتا از این پاسخ در انتخاب عمل بعدی تأثیر می‌گیرد. هدف نهایی این است که اتوماتا یاد بگیرد تا از بین اعمال خود، بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداکثر برساند. کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل ۱ نشان داده شده است.

محیط را می‌توان توسط سه تایی $E \equiv \{a, b, c\}$ نشان داد که در آن $a \equiv \{a_1, a_2, ..., a_r\}$ مجموعه ورودی‌ها، $b \equiv \{b_1, b_2, ..., b_m\}$ مجموعه خروجی‌ها و $c \equiv \{c_1, c_2, ..., c_r\}$ مجموعه احتمال‌های جریمه می‌باشد. هرگاه b مجموعه‌ای دو عضوی باشد، محیط از نوع P است. در محیط از نوع Q، $b(n)$ می‌تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله [۰.۱] را اختیار کند و در محیط از نوع S، $b(n)$ متغیر تصادفی در فاصله [۰.۱] است. c_i احتمال این‌که عمل a_i نتیجه نامطلوب داشته باشد میباشد.

شکل(۱): ارتباط بین اتوماتای یادگیر و محیط

اتوماتای یادگیر با ساختار متغیر را می‌توان توسط چهارتایی $\{a, b, p, T\}$ نشان داد که a مجموعه اعمال اتوماتا، b مجموعه ورودی‌های اتوماتا، $p = \{p_1, ..., p_r\}$ بردار احتمال انتخاب هریک از عمل‌ها و $p(n+1) = T[a(n), b(n), p(n)]$ الگوریتم یادگیری می‌باشد. الگوریتم زیر یک نمونه از الگوریتم‌های یادگیری خطی است. فرض می‌کنیم عمل a_i در مرحله n ام انتخاب شود.

- پاسخ مطلوب از محیط

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1-a)p_j(n) \quad \forall j \quad j \neq i \end{aligned} \tag{۱}$$

- پاسخ نامطلوب از محیط

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n) \\ p_j(n+1) &= (b/r-1) + (1-b)p_j(n) \quad \forall j \quad j \neq i \end{aligned} \tag{۲}$$

در روابط (۱) و (۲)، a پارامتر پاداش و b پارامتر جریمه می باشند[1,3,4,5,13].

اتوماتای یادگیر توزیع شده[۷]: اتوماتای یادگیر توزیع شده یک شبکه از اتوماتاهای یادگیر است که بمنظور حل یک مسالهٔ خاص به صورت گروهی با یکدیگر همکاری می نمایند[5]. اتوماتای یادگیر توزیع شده را می توان توسط چند تایی $< V, E, T, V_0 >$ نمایش داد که در آن V معرف مجموعهٔ رئوس گراف متناظر با مجموعهٔ اتوماتا های یادگیر A. $E \subset A \times A$ معرف یالهای گراف و E_i متناظر با مجموعهٔ عمل های اتوماتا A_i بگونه ای که یال E_{ij} متناظر است با عمل a_{ij} از اتوماتای A_i و با عمل a_{ji} از اتوماتای A_j. $T = \{T^1, T^2, ..., T^n\}$ معرف مجموعهٔ الگوریتم های یادگیر و V_0 گرهٔ ریشه را در اتوماتای یادگیر توزیع شده نشان می دهد[6,7].

ابتدا اتوماتای یادگیر ریشه یک عمل را انتخاب می نماید. انتخاب یال مذکور موجب فعال سازی اتوماتایی که در انتهای دیگر یال قرار دارد می شود. این اتوماتا نیز عملی را انتخاب و این سلسله مراتب فعال سازی تا فعال سازی اتوماتای برگ ادامه پیدا می کند. عمل انتخابی توسط اتوماتای برگ به محیط اعمال می گردد و پاسخ محیط به تمامی اتوماتاهای فعال شده در مسیر اتوماتای ریشهٔ تا اتوماتای برگ برگشت داده می شود. هر یک از اتوماتاها بر اساس پاسخ دریافتی ازمحیط، بکمک الگوریتم یادگیری، بردار احتمالات خود را بروز می نمایند[4,8].

۳. الگوریتم پیشنهادی

گام ۱. در مرحلهٔ نخست از الگوریتم، یک شبکه از اتوماتاهای یادگیر را مطابق آنچه در بخش های قبل تشریح گردید، متناظر با گراف تصادفی وزن دار ورودی $G = < V, E, W >$ در نظر می گیریم.

گام ۲. با شروع از اتوماتای آغازین A_1، فعال سازی اتوماتاها آغاز می گردد. چنانچه در هر مرحله اتوماتا به هر دلیلی همچون بروز حلقه نتواند عملی را انتخاب نماید، گام۲ با انتخاب یکی از اعضاء مجموعهٔ اتوماتاهای فعال نشده، به عنوان اتوماتای آغازین جدید، ادامه می یابد.

گام۳. مرحلهٔ دوم از الگوریتم تا زمانی که تعداد یالهای انتخابی توسط گروه اتوماتاهای فعال شده کمتر از $(n-1)$ یال باشد و یا تمامی اتاماتا ها فعال شده باشند، ادامه پیدا می کند.

گام ۴. در این گام طول درخت پوشای انتخابی در مرحلهٔ t ام یعنی $Lt_i(t)$، محاسبه و با مقدار T_k (مقدار آستانهٔ[۸] تعیین شده برای مرحلهٔ k ام) مقایسه می گردد. مقدار آستانه عبارتست از متوسط طول درخت های انتخابی تا این مرحله و بکمک رابطهٔ $T_k = (k-1)T_{k-1} + L(t_i)/k$ تعیین می گردد.

گام ۵. محیط با توجه به نتیجهٔ گام ۴، چنانچه $L\tau_i(k) \leq T_k$ باشد، آنگاه پاسخ محیط مطلوب بوده و تمامی

---

[6] Learning Automata

[7] Distributed Learning Automata

[8] Threshold

اتوماتا های فعال شده بردار احتمال عمل خود را با دریافت پاداش از محیط اصلاح می نمایند. در غیر این صورت تمامی اتوماتاهای فعال شده از طرف محیط جریمه دریافت خواهند کرد.

گام ۶. عمل پیمایش درخت های پوشای گراف تا زمانی که حاصل ضرب احتمالات انتخاب یال های پیمایش شده از یک مقدار معین کمتر بوده و یا تعداد درخت های پیمایش شده از مقدار از پیش تعیین شده ای کمتر باشد، ادامه پیدا می کند. درختی که در آخرین پیمایش و قبل از خاتمهٔ الگوریتم انتخاب گردد به عنوان درخت پوشای بهینه در نظر گرفته می شود. شبه کد و جزییات دقیق الگوریتم در نسخهٔ کامل مقاله آمده است.

به منظور جلوگیری از الگوی بروز حلقه در فرایند فعال سازی اتوماتا ها، حذف عمل اتوماتا در الگوریتم پیشنهادی به صورت زیر صورت می گیرد. با فرض آنکه مسیر $p_{i,j}=\{E_{(i,k)},p_{k,l},E_{(l,j)}\}$ بازاء تمامی $k,l\in V$ به شرط آنکه $i\neq l,j$ و $k\neq j$ دنبالهٔ یالهایی باشد که راس v_i را به v_j متصل نماید و $p_j^i(t)$ و $q_j^i(t)$ بترتیب احتمالات انتخاب یال $E_{(i,j)}$ و مسیر $p_{i,j}$ در مرحلهٔ t ام باشند، حال اگر یال $E_{(i,j)}$ در t امین مرحله از الگوریتم توسط اتوماتای A_i انتخاب گردد. در این صورت می بایستی احتمال انتخاب یال $E_{(k,i)}$ بازاء هر مسیر $p_{j,k}$ به سمت صفر میل نماید[7].

نمونه گیری استاندارد: برای بدست آوردن تعداد نمونه برداریهـای های مورد نیاز از هر یال از گراف ورودی، بگونه ای کـه بتـوان تضـمین کرد انحراف متوسط طول نمونه های اخذ شده از متوسط واقعـی آن بـا نرخ اطمینان مطلوبی کمتر از ε باشد، از قضیه زیر استفاده می شود.

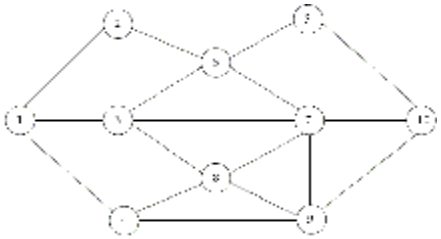
قضیه: فرض کنید $x_1,x_2,...,x_n$ دنباله ای از متغیر های تصادفی مستقل با توزیع یکسان باشند، بگونه ای که $E(x_i)=\mu$ و $Var(x_i)=\sigma^2$. با فرض آنکه متوسط μ با اطمینان $1-\delta$ در بازهٔ $\bar{x}\pm\frac{\sigma}{\sqrt{n\delta}}$ قرار داشته باشد. آنگاه بازاء هر مقدار دلخواه وبه قدر کافی کوچک برای ε، وجود دارد n_0ای بگونه ای که بازاء تمامی مقادیر $n\geq n_0$ خواهیم داشت[2].

$$\forall\varepsilon\,\exists n_0\ \forall n\geq n_0\,;\quad p\{|\bar{x}_n-\mu|\mathbf{p}\,\varepsilon\}\ \mathbf{f}\ 1-\delta$$

۴. نتایج آزمایشها

در هر آزمایش ابتدا بکمک روش نمونه گیری استاندارد حداقل تعداد نمونه های مورد نیاز از هریال، بگونه ای که متوسط نمونه های اخذ شده بازاء مقدار معین و به قدرکافی کوچک پارامتر خطای ε با احتمال $1-\delta$ در بازهٔ $(\mu-\varepsilon,\mu+\varepsilon)$ قرار داشته باشد، تعیین می گردد. سپس تعداد نمونه گیریهای مورد نیاز در این روش با تعداد نمونه های اخذ شده توسط الگوریتم پیشنهادی مقایسه میگردد و نشان داده میشود که تعداد نمونه گیری های انجام گرفته توسط الگوریتم پیشنهادی به مراتب کمتر از تعداد نمونه گیریهای انجام شده به روش استاندارد می باشد. برای پیدا کردن یکی از درختهای پوشای کمینه میتوان با اجرای الگوریتم کراسکال بر روی گراف، که در ان طول هر یال متوسط طولهای نمونه گیری شده از آن یال میباشد، درخت پوشای کمینه را بدست آورد.

الگوریتم پیشنهادی بر روی گراف تصادفی G(شکل ۲)که از مرجع [5] اقتباس گردیده است، آزمایش می گردد. (تابع توزیع احتمالی طول یالها در این مرجع آمده است). بازهٔ اطمینان بین ۵۰٪ تا ۹۹٪ تغییر داده شده و برای هر مقدار مجموع تعداد نمونه های مورد نیاز از یالها تعیین گردیده است که نتایج آن در جدول ۲ آمده است.



شکل(۲): گراف تصادفی G

همچنین تعداد نمونه های مورد نیاز از هر یال بازاء بازه های اطمینان ۵۰٪، ۷۰٪، ۹۰٪ و۹۹٪ و با در نظر گرفتن $\varepsilon=0.001$، در جدول ۱ ارایه گردیده است. تعداد نمونه های بدست آمده از این طریق با تعداد نمونه های گرفته شده توسط الگوریتم پیشنهادی مقایسه میشود. الگوریتم پیشنهادی که از الگوریتم یادگیری L_{R-I} استفاده میکند برای مقادیر مختلف پارامتر پاداش برای گراف شکل ۲ آزمایش گردیده است. نتایج ارایه شده در جدول ۳ متوسط نتایج بدست آمده برای ۱۰۰ بار تکرار آزمایش میباشند. الگورینم زمانی خاتمه می یابد که احتمال انتخاب درخت پوشای انتخاب شده در یک تکرار به بیش از ۰.۹۵ برسد.

	Confidence Interval			
Edge	50	70	90	99
(1,2)	228	399	342	400
(1,3)	238	205	234	416
(1,4)	423	372	570	427
(2,3)	566	615	621	492
(2,5)	221	303	408	377
(2,6)	231	253	297	480
(3,4)	198	223	299	421
(3,6)	180	223	272	481
(3,7)	166	176	238	1189
(3,8)	159	225	222	460
(4,8)	168	234	350	1303
(4,9)	159	191	262	920
(5,6)	209	232	291	496
(5,7)	227	212	290	369
(5,10)	124	170	349	1795
(6,7)	214	188	367	381
(7,8)	488	579	431	576
(7,9)	186	171	252	619
(7,10)	288	299	257	364
(8,9)	189	225	231	525
(9,10)	235	240	250	298

جدول(۱): متوسط تعداد نمونه های اخذ شده از هر یال در روش نمونه گیری استاندارد برای بازه های اطمینان مختلف

کاهش مقدار پارامتر a در الگوریتم پیشنهادی موجب افزایش نرخ همگرایی الگوریتم به پاسخ بهینه میگردد ولی درمقابل باعث افزایش تعداد نمونه گیریها از یالهای گراف تصادفی میگردد. این مساله که چه رابطه ای بین تغییرات پارامتر a و تغییرات بازهٔ اطمینان در نمونه گیری استاندارد وجود دارد دست مطالعه میباشد. همانطور که در جداول مشاهده میشود تعداد نمونه های گرفته شده توسط الگوریتم پیشنهادی به مراتب کمتر از تعداد نمونه های مورد نیاز در روش استاندارد میباشد.

Confidence Interval	Total Number of Samples
50	5105
55	5063
60	5398
65	5304
70	5747
75	5522
80	5505
85	6050
90	6844
92.5	7062
95	7632
97.5	9181
99	12797

جدول(۲): متوسط تعداد کل نمونه ها ذر نمونه گیری استاندارد

در جدول ۳ مشاهده می شود که با کاهش مقدار پارامتر پاداش سرعت همگرایی الگوریتم به مقدار قابل توجهی کاهش و از سوی دیگر صحت همگرایی افزایش می یابد. آزمایشها نشان داده است که با انتخاب مقدار مناسب $a \leq 0.008$ برای پارامتر پاداش، الگوریتم میتواند همواره به یکی از درختهای پوشای کمینه همگرا شود.

Learning Parameter	Total Number of Samples	Number of traveling Optimal tree	Convergence Rate
0.007	9066	5236	100
0.008	8850	4346	100
0.009	6789	3214	98.89
0.01	5576	2854	97.78
0.03	1122	291	91.76
0.04	881	282	87.78
0.05	608	206	78.89
0.06	375	125	78.33
0.07	258	91	76.67
0.08	181	65	75
0.09	154	59	70
0.1	138	42	68.33

جدول(۳): متوسط تعداد کل نمونه های اخذ شده توسط الگوریتم پیشنهادی

۵. نتیجه گیری

در این مقاله یک الگوریتم مبتنی بر اتوماتای یادگیر توزیع شده بمنظور تعیین درخت پوشای کمینه در یک گراف تصادفی که در آن تابع توزیع وزنهای یالهای گراف تصادفی از قبل شناخته شده نمیباشند پیشنهاد گردید. به منظور ارزیابی الگوریتم پیشنهادی، تعداد نمونه های گرفته شده توسط الگوریتم پیشنهادی با تعداد نمونه های موردنیاز به روش نمونه گیری استاندارد مقایسه گردید و نشان داده شد که تعداد نمونه های گرفته شده توسط الگوریتم پیشنهادی بمراتب از تعداد نمونه های گرفته شده به روش نمونه گیری استاندارد کمتر است.

مراجع

[1] K. Vrbeek, A. Nowe, M. Peeters and K. Tuyls, "Multi-Agent Coordination in Tree Structured Multi-Stage Games", Proc. of Fourth Symposium on Adaptive Agents and Multi-Agent Systems, 2003.
[2] A. Papoulis, " Probability, Random Variables, and Stochastic Processes", 3rd Edition, New York, McGraw-Hill, 1991.
[3] P. S.Sastry, V. V.Phansalkar and M.A.L. Thathachat, "Decentralized Learning of Nash Equilibria in Multi-Person Stochastic Games with Incomplete Information", IEEE Transactions on Systems, Man and Cybernetics, Vol.24, pp.769-777, May 1994.
[4] M. A. L. Thathachat, P. S. Sastry, " A Hierarchical System of Learning Automata That Can Learn The Globally Optimal Path", Information Science, Vol.42, pp.743-766, 1997.
[5] H. Beigy, M. R. Meybodi, "Utilizing Distributed Learning Automata to Solve Stochastic Shortest Path Problems", International Journal of Uncertainty, Fuzziness

[9] P. Spira, A. Pan, "On Finding and Updating Spanning Trees and Shortest Paths", SIAM Journal on Computing, Vol. 4, No. 3, pp. 375–380, 1975.
[10] D. Frigioni, A. Marchetti-Spaccamela, U. Nanni, "Fully Dynamic Algorithms for Maintaining Shortest Paths Trees", Journal of Algorithms, Vol. 34, pp. 251–281, 2000.
[11] P. Narvaez, K. Siu, H. Tzeng, "New Dynamic Algorithms for Shortest Path Tree Computation", IEEE/ACM Transactions on Networking, Vol. 8, No. 6, pp. 734–746, 2000.
[12] H. Ishii, S. Shiode and T. Nishida, "Stochastic Spanning Tree Problem", Discrete Applied Mathematics, Vol. 3, pp. 263-273, 1981.
[13] K. S. Narendra and K. S. Thathachar, "Learning Automata: An Introduction", New York, Printice-Hall, 1989.
and Knowledge-Based Systems, Vol.14, pp. 591-615, Oct 2006.
[6] H. Beigy, “Intelligent Channel Assignment in Cellular Networks: A Learning Automata Approach”, PhD Thesis, Computer Engineering Department, Amir Kabir University of Technology, Tehran, Iran, 2006.
[7] M. A. L. Thathachar and B. R. Harita, "Learning Automata with Changing Number of Actions", IEEE Transactions on Systems, Man, and Cybernetics, vol. SMG17, pp. 1095-1100, Nov. 1987.
[8] M. A. L. Thathachat, V.V.Phansalkar, "Convergence of Teams and Hierarchies of Learning Automata in Connectionist Systems", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 24, pp.1459-1469, Nov. 995.