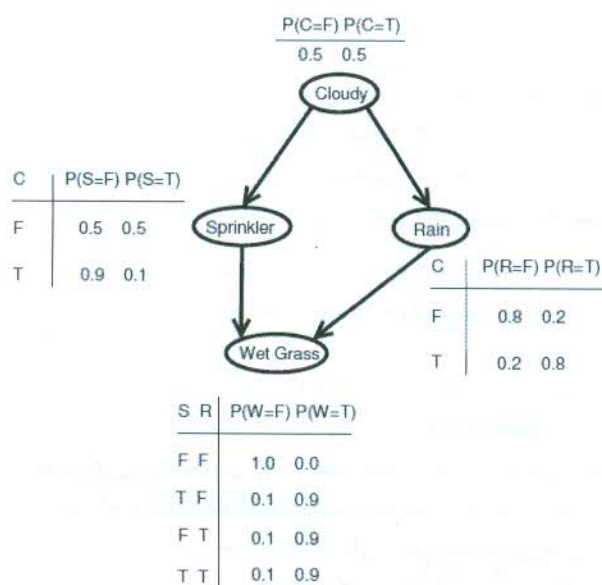


یادگیری ساختاری شبکه‌های بیزی یک رهیافت مبتنی بر آتاماتهای یادگیر

محمدرضا ملاخلیلی میبدی و محمدرضا میبدی



شکل ۱: نمونه‌ای از یک شبکه بیزی.

نشان‌دهنده ارتباط وابستگی مستقیم بین این متغیرها است [۲]. برای هر متغیر $x_i \in \vec{V}$ یک خانواده از توزیع‌های شرطی $p(x_i | Pa(x_i))$ موسوم به پارامتر داریم که $Pa(x_i)$ مجموعه والد‌های متغیر x_i را نشان می‌دهد. توزیع تجمعی روی \vec{V} را می‌توان از طریق این توزیع‌های شرطی به دست آورد [۳]

$$p(x_1, x_2, \dots, x_n) = \prod_{i=1}^n p(x_i | Pa(x_i)) \quad (1)$$

که (۱) نشان‌دهنده تجزیه توزیع توأم است. روابط وابستگی / استقلال این تجزیه را ممکن می‌کنند به شکل گرافیکی در ساختار شبکه بیزی کد می‌شوند. به بیان دیگر، وجود یا عدم وجود ارتباطات مستقیم میان جفت‌های متغیرها، وابستگی یا عدم وابستگی مستقیم بین متغیرها را نشان می‌دهند. نمونه‌ای از یک شبکه بیزی و روابط استقلال شرطی میان متغیرها را در شکل ۱ مشاهده می‌کنید.

مسئله یادگیری یک شبکه بیزی را می‌توان به این صورت توصیف کرد: یک مجموعه آموزشی $D = \{\vec{V}^1, \vec{V}^2, \dots, \vec{V}^m\}$ از نمونه‌هایی از \vec{V} داده شده است. شبکه بیزی BN را به گونه‌ای بیابید که بهترین تطابق با داده‌های D را داشته باشد. این مسئله در رده مسایل NP-Hard رده‌بندی می‌شود [۴] و لذا باید از روش‌های مکاشفه‌ای آن را حل کرد.

رهیافت معمول برای حل این مسئله استفاده از یک تابع امتیازدهی f است که بر اساس داده‌های آموزشی می‌تواند هر یک از شبکه‌های ممکن را ارزیابی و امتیازدهی کند. بر اساس این امتیاز، جست و جو برای یافتن

چکیده: یکی از مسایل جالب در هوش مصنوعی ساخت شبکه بیزی بر اساس نمونه‌هایی از داده‌ها است؛ یعنی فرض کنید یک شبکه بیزی N روی مجموعه متغیرهای V مفروض است. هدف، ساخت یک شبکه بیزی - استخراج مجموعه‌ای از روابط علت / معلولی - میان مجموعه متغیرها بر اساس نمونه‌هایی که از N استخراج شده و بدون در اختیار داشتن N است. از این مسأله در متون با عنوان یادگیری ساختاری شبکه بیزی یاد می‌شود. یکی از روش‌های مهم در یادگیری ساختاری شبکه‌های بیزی با استفاده از داده‌های نمونه، استفاده از معیارهای مبتنی بر امتیاز برای ارزیابی میزان برازندگی یک ساختار بیزی مفروض با داده‌های نمونه و جست و جو در میان ساختارهای ممکن است. جست و جو برای یافتن یک ساختار مناسب برای شبکه بیزی که بیشترین سازگاری را با نمونه‌ها داشته باشد غالباً از طریق جست و جو در فضای ساختارها با استفاده از تکنیک‌های جست و جوی استاندارد یا الهام‌گرفته از طبیعت نظیر تپه‌نوردی حریمانه، الگوریتم‌های ژنتیک، شبیه‌سازی حرارتی یا الگوریتم تبرید، بهینه‌سازی کلونی مورچه‌ها و نظایر آن صورت می‌گیرد.

در این مقاله یک روش جدید مبتنی بر آتاماتای یادگیر برای یادگیری ساختاری شبکه بیزی ارائه شده است. در این روش آتاماتای یادگیر به عنوان یک ابزار جستجوی تصادفی مورد استفاده قرار می‌گیرد. از ویژگی‌های روش جدید پیشنهادی جستجوی هم‌زمان در فضای جایگشت‌های ممکن از متغیرها (فضای ترتیب متغیرها) و فضای ساختارها (فضای DAGها) است. ضمن بررسی ریاضی الگوریتم پیشنهادی، روش جدید روی تعدادی از شبکه‌های نمونه مورد آزمایش قرار گرفته است.

کلیدواژه: یادگیری ساختار شبکه بیزی، آتاماتای یادگیر.

۱- مقدمه

یک شبکه بیزی - یا آن گونه که در ادبیات این حوزه مصطلح است، شبکه باور^۱ یا شبکه علی^۲ ابزاری برای بازنمایی دانش است که امکان مدیریت مؤثر ارتباطات وابستگی / استقلال را میان متغیرهای تصادفی تشکیل‌دهنده دامنه مسأله‌ای که مدل کردن آن مد نظر است فراهم می‌کند [۱]. این نمایش از دو مؤلفه تشکیل شده است: ساختار و پارامترها. ساختار شبکه بیزی یک گراف جهت‌دار و فاقد دور $G = (\vec{V}, E)$ است که در آن $\vec{V} = \{x_1, x_2, \dots, x_n\}$ مجموعه گره‌هایی است که متغیرهای سیستم را نشان می‌دهند و $E \subset (\vec{V} \times \vec{V})$ یک مجموعه از یال‌ها و

این مقاله در تاریخ ۱۰ اسفند ماه ۱۳۹۲ دریافت و در تاریخ ۱۲ آذر ماه ۱۳۹۳ بازنگری شد.

محمدرضا ملاخلیلی میبدی، دانشگاه آزاد اسلامی، واحد میبد، گروه کامپیوتر، میبد، (email: mollakhalili@maybodiau.ac.ir)

محمدرضا میبدی، آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، (email: mmeybodi@aut.ac.ir)

1. Belief Network

2. Causal Network

طبیعت [۱۱] مجدداً تلاش‌هایی برای حل این مسأله صورت گرفته است [۱۲] تا [۱۸].

نگاه دیگری که به روش‌های پیشنهادی برای حل مسأله یادگیری ساختاری می‌توان داشت، به نحوه بررسی تناسب و برازندگی ساختارهای گوناگون برمی‌گردد [۱]. در گروهی از این روش‌ها از آزمون‌های استقلال شرطی [۱۹] و در گروهی دیگر از روش‌های مبتنی بر امتیازدهی [۶] تا [۸] برای تعیین میزان برازندگی یک ساختار با داده‌های نمونه استفاده شده و در برخی نیز ترکیبی از هر دو مورد استفاده قرار گرفته است [۲۰].

از منظر دیگر می‌توان الگوریتم‌های مورد استفاده برای یادگیری ساختار یک شبکه بیزی را به دو گروه تقسیم کرد. گروهی که در فضای DAG‌های مختلف ممکن به جست و جو می‌پردازند [۹]، [۱۸] و [۲۱] و گروهی دیگر که در میان ترتیب‌های مختلف متغیرها به جستجوی بهترین ساختار می‌پردازند [۲۲] تا [۲۴]. گروه دوم الگوریتم‌ها در مقایسه با گروه اول فضای جست و جوی محدودشده‌تری دارند و بنابراین اگر یک ترتیب کلی از متغیرها مفروض باشد، پیدا کردن بهترین ساختار به مسأله انتخاب بهترین والد‌ها برای هر گره تقلیل می‌یابد. اما اگر ترتیب شناخته‌شده نباشد می‌توانیم به جای جست و جو روی DAG‌های مختلف، روی مرتب‌سازی‌های مختلف جست و جو را انجام دهیم.

برخی از نویسندگان نشان داده‌اند که جست و جو در فضای جایگشت‌های متغیرها از جست و جو در فضای ساختارها مناسب‌تر و الگوریتم به دست آمده کارآمدتر و دقیق‌تر است [۲۴] و [۲۵].

۲-۱ جست و جو در فضای جایگشت‌ها

یک شبکه بیزی $G = (\vec{V}, E)$ را می‌توان با بردارهای $G = (\vec{G}_1, \vec{G}_2, \dots, \vec{G}_n)$ از مجموعه والد‌های هر یک از رئوس متناظر با مجموعه متغیرهای $\vec{V} = \{x_1, x_2, \dots, x_n\}$ توصیف کرد. به عبارت دیگر \vec{G}_i زیرمجموعه‌ای از \vec{V} است که یال‌هایی از آنها به سمت x_i وجود دارد. به عنوان مثال شبکه $G = (\{\{4\}, \{1, 3\}, \{1\}, \{\}\})$ متناظر با یک DAG با مجموعه یال‌های $4 \rightarrow 1, 1 \rightarrow 2, 3 \rightarrow 2, 1 \rightarrow 3$ است که این شبکه را در شکل ۲ مشاهده می‌کنید.

علاوه بر این از مفهوم ترتیب متغیرها استفاده می‌کنیم. مجموعه متغیرهای V با یک ترتیب مشخص مثلاً $Ord = (4, 1, 3, 2)$ را در نظر بگیرید. i امین عنصر در یک ترتیب را با Ord_i نمایش می‌دهیم. مثلاً در ترتیب مشخص شده $Ord_4 = 1$ است.

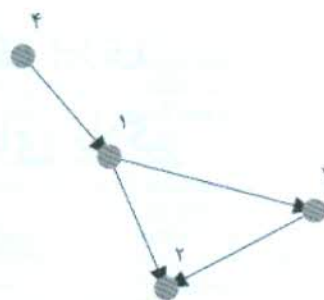
تعریف ۱ (سازگاری شبکه بیزی با یک ترتیب): یک شبکه بیزی $G = (\vec{G}_1, \vec{G}_2, \dots, \vec{G}_n)$ را با ترتیب Ord سازگار گویند هر گاه

$$\forall i, \vec{G}_i \subseteq \bigcup_{j=1}^{i-1} Ord_j \quad (4)$$

بر اساس این تعریف زمانی که تمام والد‌های یک گره در ترتیب Ord قبل از آن گره قرار گیرند شبکه را با ترتیب مفروض سازگار گویند. به عنوان مثال شبکه $G = (\{\{4\}, \{1, 3\}, \{1\}, \{\}\})$ در شکل ۲ با ترتیب $\{4, 1, 3, 2\}$ سازگار است.

در هر DAG حداقل یک گره فاقد یال خروجی وجود دارد، گره‌ای که والد هیچ گره دیگری نیست و این گره‌ها را Sink شبکه گویند. وجود Sink‌ها ناشی از فاقد دور بودن DAG است. هر مسیر جهت‌دار در DAG بایستی طول محدودی داشته باشد و پایان یک مسیر با حداکثر طول به یک Sink ختم می‌شود.

در [۲۶] یک ره یافت مبتنی بر جست و جوی بهترین ترتیب برای شبکه‌های بیزی با تعداد متغیرهای کم پیشنهاد شده است. این روش،



شکل ۲: نمونه‌ای از یک DAG سازگار با ترتیب $\{4, 1, 3, 2\}$ و متناظر با شبکه $G = (\{\{4\}, \{1, 3\}, \{1\}, \{\}\})$.

بهترین شبکه صورت می‌گیرد. معیارهای امتیازدهی متعددی اعم از بیزی و غیر بیزی در متون پیشنهاد شده و قابل استفاده است [۵] تا [۸]. مهم‌ترین ویژگی مورد انتظار از یک شاخص f تجزیه‌پذیری آن در شرایط وجود داده‌های کامل است، یعنی تابع امتیازدهی می‌تواند به صورت زیر تجزیه شود

$$f(G : D) = \sum_{i=1}^n f(x_i, Pa(x_i) : N_{x_i, Pa(x_i)}) \quad (2)$$

که $N_{x_i, Pa(x_i)}$ آماره متغیر x_i و $Pa(x_i)$ در داده‌های D است. به عبارت دیگر تعداد نمونه‌هایی از D که با x_i و $Pa(x_i)$ مطابقت دارند. تجزیه‌پذیری این معیار از نقطه نظر الگوریتم‌های جست و جو و یادگیری فوق‌العاده با اهمیت است. یک پروسه جست و جوی محلی که یک یال را در هر گام تغییر می‌دهد (مثلاً یالی را به مجموعه یال‌های یک شبکه موجود می‌افزاید یا آن را حذف می‌کند) بایستی بتواند به گونه‌ای مؤثر میزان تغییرات پدیدآمده ناشی از این تغییر را ارزیابی کند که تنها در صورت تجزیه‌پذیری تابع امتیازدهی امکان‌پذیر است.

ادامه این مقاله به این صورت سازمان‌دهی شده است: در بخش ۲ به بررسی مفصل پیشینه موضوع پرداخته‌ایم. بخش ۳ به توضیح کوتاهی درباره‌ی آتاماتای یادگیر اختصاص داده شده است. در بخش ۴ روش جدید پیشنهادی تشریح شده و بخش ۵ به بررسی همگرایی روش جدید پیشنهادی اختصاص یافته است. در بخش ۶ به بررسی تجربی الگوریتم جدید پیشنهادی پرداخته‌ایم و در نهایت بخش ۷ به نتیجه‌گیری و پیشنهادهایی برای کارهای آتی اختصاص یافته است.

۲-۲ بررسی کارهای صورت گرفته

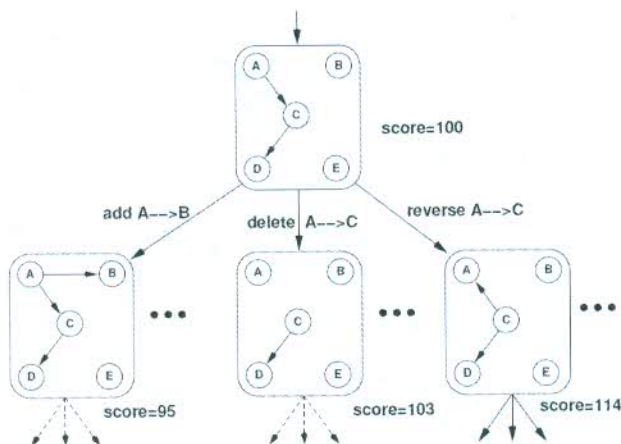
ثابت شده است که $r(n)$ تعداد ساختارهای مختلف برای یک شبکه بیزی با n گره از طریق رابطه بازگشتی (۳) قابل محاسبه است [۹]

$$r(n) = \sum_{k=1}^n (-1)^{k+1} \frac{n!}{(n-k)!k!} r^{k(n-k)} r(n-k) = n^{r(n)} \quad (3)$$

بدیهی است جست و جو برای یافتن جواب در چنین فضای جواب بزرگی ناممکن است. بنابراین بایستی از روش‌های مکاشفه‌ای برای جست و جو و یافتن یک بهترین جواب یا نزدیک‌ترین جواب بهره گرفت.

الگوریتم‌های متعددی که برای حل مسأله یادگیری ساختاری شبکه بیزی پیشنهاد شده را می‌توان از منظرهای گوناگون دسته‌بندی کرد.

از نظر زمانی بخش عمده‌ای از تلاش‌های صورت گرفته برای حل این مسأله به دو دهه پیش برمی‌گردد که با دیدگاه آماری به حل مسأله نگاه شده است [۶]، [۸] و [۱۰]. با پیدایش روش‌های بهینه‌سازی الگو گرفته از



شکل ۴: فرایند جست و جوی مورد استفاده توسط الگوریتم جست و جوی تپهنوردی [۲۸] (صفحه ۱۹، شکل ۲ و ۳).

امتیازات هر گراف را در این همسایگی محاسبه می‌کند. سپس گرافی در همسایگی را که بیشینه امتیاز را کسب کند به عنوان گراف بعدی برای جست و جوی مرحله بعد انتخاب می‌کند.

در شبکه‌های بیزی مجموعه همسایگی را می‌توان مجموعه گراف‌هایی تعریف کرد که با گراف فعلی در کم یا زیاد شدن یک یال یا تغییر جهت یک یال تفاوت دارند (شکل ۴). ما در بخش شبیه‌سازی‌ها از نسخه‌هایی از این الگوریتم به نام‌های GS^1 ، HC^1 و GS^2 استفاده خواهیم کرد [۲۷] که شمای کلی الگوریتم در شکل ۳ نشان داده شده است.

۲-۴ الگوریتم B

الگوریتم B یک روش ساخت شبکه بیزی به شیوه مکاشفه‌ای حریصانه است [۹] که این روش با یک DAG خالی شروع می‌کند (یک شبکه فاقد یال). در هر گام یالی که بتواند بیشترین افزایش را در معیار امتیاز f ایجاد کند و موجب ایجاد حلقه در گراف نشود را به ساختار می‌افزاید. این الگوریتم تا زمانی که افزودن یک یال معتبر موجب افزایش در امتیاز نشود ادامه می‌یابد. این الگوریتم در زمره الگوریتم‌هایی است که فضای ساختارها را برای یافتن ساختار بهینه جست و جو می‌کنند.

در این الگوریتم $A[i, j]$ یک ماتریس مجاورت است که مقادیر مختلف $f(x_i, Pa(x_i) \cup \{x_j\}) - f(x_i, Pa(x_i))$ را ذخیره می‌کند. این مقدار، بهبود به دست آمده ناشی از افزودن یال جهت‌دار $x_j \rightarrow x_i$ را نشان می‌دهد. یال‌هایی که افزودن آنها در گراف باعث پیدایش سیکل جهت‌دار می‌شود با انتساب مقدار $-\infty$ به $A[i, j]$ مشخص می‌شوند. در هر گام بعد از افزودن یال معتبر $x_j \rightarrow x_i$ الگوریتم، یال‌هایی که افزودن آنها ممنوع است را از لیست یال‌ها حذف می‌کند و سپس با توجه به این که $f(x_i, Pa(x_i))$ تغییر کرده است، مقدار جدید $A[i, k]$ را برای تمامی یال‌های معتبر $x_k \rightarrow x_i$ محاسبه می‌کند. پیچیدگی این به روز رسانی $O(n^2)$ است. شبه‌کد این روش در شکل ۵ نشان داده شده است. در [۹] ترکیبی از الگوریتم B و روش بهینه‌سازی کلونی مورچه‌ها برای انجام جست و جو در فضای ساختارهای شبکه بیزی ارائه شده است. در الگوریتم پیشنهادی مسأله به شکل DAG‌هایی با n گره مدل شده است. حالت G_n یک DAG با گره‌های $x_i \in \bar{V}$ و دقیقاً h یال است. مورچه به شکل تزییدی جواب را با شروع از یک گراف خالی G می‌سازد. این کار با افزودن یال $x_i \rightarrow x_j$ به حالت فعلی G_n به دست می‌آید

ورودی: مجموعه داده‌های D

خروجی: DAG متناظر با شبکه بیزی B

```

 $E \leftarrow \emptyset$ ;
 $T \leftarrow \text{ProbabilityTables}(E, D)$ ;
 $B \leftarrow (u, E, T)$ ;
 $\text{Score} \leftarrow -\infty$ ;
repeat
     $\text{maxScore} \leftarrow \text{Score}$ ;
    foreach attribute pair  $(X, Y)$ 
        foreach  $E' \subset \{E \cup \{X \rightarrow Y\}, E - \{X \rightarrow Y\}\}$ 
             $E - \{X \rightarrow Y\} \cup \{Y \rightarrow X\}$  do
                 $T' \leftarrow \text{ProbabilityTables}(E', D)$ 
                 $B' \leftarrow (u, E', T')$ 
                 $\text{newScore} \leftarrow \text{Score}(B', D)$ 
                if  $\text{newScore} > \text{Score}$ 
                     $B \leftarrow B'$ 
                     $\text{Score} \leftarrow \text{newScore}$ 
            end if
        end for
    end for
until  $(\text{Score} \leq \text{maxScore})$ 

```

شکل ۳: الگوریتم جست و جوی تپهنوردی Hill Climbing.

ورودی: مجموعه داده‌های D و n : تعداد متغیرها

خروجی: ماتریس مجاورت DAG متناظر با شبکه بیزی در قالب ماتریس مجاورت A_{adj}

```

repeat
    Select two indexes  $(i, j)$  such that  $(i, j) = \arg \max_{i, j} A[i, j]$ ;
    if  $A[i, j] > 0$ 
         $Pa(x_i) = Pa(x_i) \cup \{x_j\}$ ;
    end if
     $A[i, j] = -\infty$ ;
    forall  $x_a \in \text{Ancestors}(x_i) \cup \{x_j\}$  and
         $x_b \in \text{Descendants}(x_i) \cup \{x_j\}$ 
         $A[a, b] = -\infty$ ;
    end for
    for  $k \leftarrow 1 : n$ 
        if  $A[i, j] > -\infty$ 
             $A[i, k] = f(x_i, Pa(x_i) \cup \{x_k\}) - f(x_i, Pa(x_i))$ ;
        end if
    end for
until

```

شکل ۵: الگوریتم B.

جست و جو را در میان تمام ترتیب‌های ممکن و تمام مجموعه والد‌های ممکن برای یک متغیر انجام می‌دهد.

۲-۲ جست و جو در فضای ساختارها

رهیافت اصلی که برای حل مسأله یافتن ساختار شبکه بیزی بر اساس داده‌ها وجود دارد، جست و جو در فضای گراف‌های جهت‌دار فاقد دور (DAG) ممکن میان متغیرها است. با توجه به بزرگی چنین فضایی از جواب‌ها، هیوریستیک‌هایی برای بهینه‌کردن فرایند جست و جو لازم است. در ادامه به برخی از تلاش‌هایی که در این زمینه صورت گرفته است اشاره خواهیم کرد.

۳-۲ جست و جوی حریصانه

جست و جوی حریصانه مشهورترین هیوریستیک بهینه‌سازی است که با یک گراف اولیه شروع می‌کند. بر اساس یک همسایگی تعریف‌شده

ورودی: D مجموعه نمونه‌ها، n : تعداد متغیرها، u حداکثر تعداد والد‌های یک گره و Ord یک ترتیب از متغیرها
خروجی: ماتریس مجاورت DAG متناظر با شبکه بیزی در قالب بردار Parents که لیست والد‌های هر گره را مشخص می‌کند.

for $i \leftarrow 1:n$

$\pi_i \leftarrow \phi$;

(π_i والد‌های گره i را نشان می‌دهد)

$P_{old} = f_{K2}(x_i, \pi_i)$; (۶) مطابق با

$OKToProceed \leftarrow true$;

while $OKToProceed$ and $|\pi_i| < u$

Let z be the node in $Pred(x_i) - \pi_i$ that maximizes $f_{K2}(i, \pi_i \cup \{z\})$;

$z = \arg \max_{z \in Pred(x_i) - \pi_i} f_{K2}(x_i, \pi_i \cup \{z\})$;

$P_{new} = f_{K2}(x_i, \pi_i \cup \{z\})$;

if $P_{new} > P_{old}$

$P_{old} \leftarrow P_{new}$;

$\pi_i \leftarrow \pi_i \cup \{z\}$;

else

$OKToProceed \leftarrow false$;

end if

end while

Parents (x_i) $\leftarrow \pi_i$;

end for

شکل ۷: الگوریتم K2.

که r_i تعداد مقادیر مختلف ممکن متغیر x_i ، q_i تعداد پیکربندی‌های (نمونه‌های) مختلف برای متغیرها در $Pa(x_i)$ ، N_{ijk} تعداد نمونه‌ها در D که متغیر x_i در آن k امین مقدار را دارد و $Pa(x_i)$ در z امین پیکربندی‌اش قرار گرفته و $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$ است.

فرض کنید $P(G)$ یک توزیع یکنواخت داشته باشد. با محاسبه $\log(P(G, D))$ به جای $P(G, D)$ داریم

$$f_{K2}(G, D) = \sum_{i=1}^n f_{K2}(x_i, Pa(x_i); N_{x_i, Pa(x_i)}) \quad (۶)$$

که در آن

$$f_{K2}(x_i, Pa(x_i); N_{x_i, Pa(x_i)}) = \sum_{j=1}^{q_i} \left(\log \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} + \sum_{k=1}^{r_i} \log(N_{ijk}!) \right) \quad (۷)$$

شبه کد K2 را در الگوریتم شکل ۷ مشاهده می‌کنید.

۲-۶ جست و جوی حریصانه در فضای هم‌ارز مارکوف

کارهای جدیدتری که در حوزه یادگیری ساختار شبکه بیزی از طریق جست و جوی حریصانه ساختارها صورت گرفته است، علاقه‌مندی به جست و جوی حریصانه هم‌ارزی مارکوف را نشان می‌دهند.

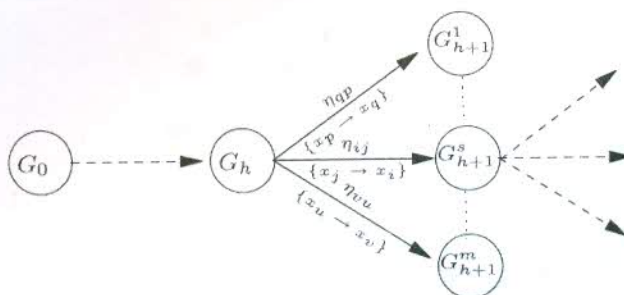
تعریف ۲: دو DAG را هم‌ارز گوئیم اگر مجموعه وابستگی/استقلال شرطی یکسانی داشته باشند. به عنوان مثال اگر X به شرط Y از Z مستقل باشد ($X \perp Z | Y$) در این صورت

$$P(X, Y, Z) = P(X)P(Y|X)P(Z|Y) =$$

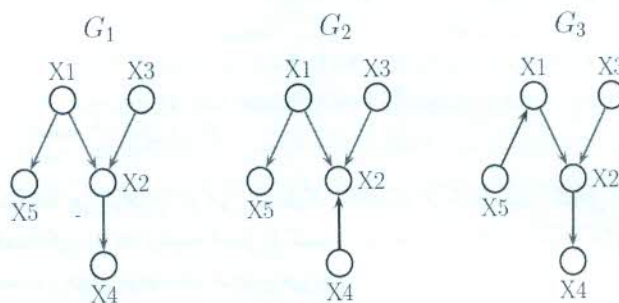
$$P(X|Y)P(Y)P(Z|Y) = P(X|B)P(Y|Z)P(Z)$$

در شکل ۸ سه نمونه DAG مشاهده می‌کنید که دو تای آنها با یکدیگر هم‌ارز هستند.

در شکل ۹ مشاهده می‌کنید که می‌توان با حذف جهت برخی یال‌ها DAG را به صورتی بدون ابهام به شکل غیر جهت‌دار نشان داد.



شکل ۶: نسخه تصادفی الگوریتم B بر اساس بهینه‌سازی کلونی مورچه [۹].



شکل ۸: DAGهای G_1 و G_3 هم‌ارز مارکوفی هستند اما با G_2 هم‌ارز نیستند [۲۹] (صفحه ۹۱۵، شکل ۲۶.۵).

یعنی $G_{h+1} = G_h \cup \{x_j \rightarrow x_i\}$. جواب نهایی حالت G_h است که در آن مورچه فاز ساخت را متوقف می‌کند. هیوریستیک انتخاب‌شده در راستای افزودن یالی است که بیشترین افزایش را در معیار f ایجاد می‌کند و بنابراین

$$\eta_{ij} = f(x_i, Pa(x_i) \cup \{x_j\}) - f(x_i, Pa(x_i)) \quad (۵)$$

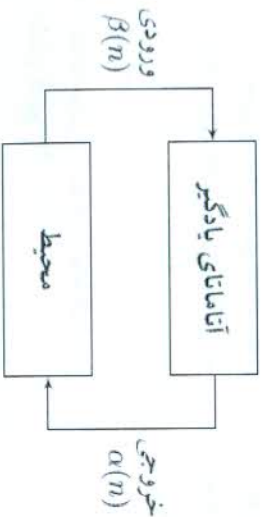
انتخاب می‌شود. فرمونها با ماتریس τ مدل می‌شوند که τ_{ij} سطح فرمون موجود روی یال از j به i را ذخیره می‌کند. اگر τ_{ij} سطح فرمون یال $x_j \rightarrow x_i$ و G^+ بهترین گراف یافته‌شده باشد مقدار فرمون بر اساس رابطه $\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}$ به روز رسانی می‌شود و در آن $\Delta\tau_{ij} = \tau_{ij}$ است اگر $x_j \rightarrow x_i \notin G^+$ و $\Delta\tau_{ij} = 1/f(G^+, D)$ است اگر $x_j \rightarrow x_i \in G^+$ باشد (شکل ۶ را ملاحظه کنید).

کمان بعدی که بایستی به گراف G اضافه شود به شیوه‌ای مشابه با الگوریتم B انتخاب می‌شود با این تفاوت که تصمیم‌گیری به شکل تصادفی انجام می‌گیرد و یال با بیشترین $[\tau_{ij}][\eta_{ij}]^{\rho}$ انتخاب شده و به ساختار شبکه اضافه می‌شود. در [۲۴] از الگوریتم‌های ژنتیک به عنوان ابزاری برای تولید جایگشت‌ها و جست و جوی حریصانه ساختارهای شبکه استفاده شده است.

۲-۵ الگوریتم K2

ایده اصلی الگوریتم K2 پیشنهادی بر اساس احتمال ساختار بر اساس داده‌های موجود است. الگوریتم K2 احتمالاً یکی از بهترین الگوریتم‌هایی است که تاکنون برای یادگیری ساختار شبکه‌های بیزی ابداع شده است. این الگوریتم از یک معیار امتیازدهی بیزی استفاده می‌کند که توزیع احتمال توأم یک شبکه بیزی G و مجموعه نمونه داده‌های D را اندازه می‌گیرد. جزئیات مفصل این الگوریتم در [۶] ارائه شده است. این معیار با کمک رابطه زیر تعریف می‌شود

$$P(G, D) = P(G) \prod_{i=1}^n \prod_{j=a}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!$$



شکل ۱۰: آنامانای یادگیر و محیط.

الگوریتم یادگیری یک رابطه بازگشتی است که برای انجام تغییرات و به روز رسانی در بردار احتمال اقدام‌های آنامانا در یک آنامانای یادگیر تصادفی با ساختار متغیر مورد استفاده قرار می‌گیرد. فرض کنید یک آنامانای یادگیر تصادفی ساختار متغیر در زمان k از میان مجموعه اقدام‌های α عمل $\alpha_i(k)$ را انتخاب کرده باشد. همچنین فرض کنید بردار احتمال انتخاب اقدام‌های آنامانا را با $p(k)$ نمایش داده‌ایم. اگر a و b پارامترهایی باشند که به ترتیب میزان افزایش یا کاهش احتمالات اقدام‌ها را مشخص می‌کنند و r تعداد اقدام‌های قابل انجام توسط آنامانای یادگیر باشد، بردار $p(k)$ توسط الگوریتم یادگیری خطی ارائه‌شده در روابط زیر به روز رسانی می‌شود. مقدار a را پارامتر پاداش و b را پارامتر جریمه می‌نامند

$$p_j(k+1) = \begin{cases} (1-a)p_j(k) + a, & j = i \\ (1-a)p_j(k), & \forall j \neq i \end{cases} \quad (9)$$

$$p_j(k+1) = \begin{cases} (1-b)p_j(k), & j = i \\ (1-b)p_j(k) + \frac{b}{r-1}, & \forall j \neq i \end{cases} \quad (10)$$

رابطه (۹) زمانی مورد استفاده قرار می‌گیرد که عمل $\alpha_i(k)$ منجر به دریافت پاداش از محیط شده باشد و (۱۰) زمانی مورد استفاده قرار می‌گیرد که این عمل به دریافت جریمه از محیط منجر شود. در (۹) و (۱۰) اگر $a = b$ باشد روابط یادگیری خطی را الگوریتم L_{R-CR} ، اگر $a \gg b$ باشد آن را L_{R-CR} و اگر $b = 0$ باشد آن را L_{R-CR} می‌نامند.

۳-۱ آنامانای یادگیر با مجموعه اقدام‌های متغیر

یک آنامانای یادگیر با مجموعه اقدام متغیر، آنامانایی است که در آن تعداد اقدام‌های موجود هر آنامانا در طول زمان تغییر می‌کند [۳۴]. فرض کنید مجموعه اقدام‌های آنامانا را با $\{\alpha_1, \alpha_2, \dots, \alpha_r\}$ و مجموعه اقدام‌های قابل انجام توسط آنامانا در زمان n را با $A(k) \subseteq \alpha$ نشان دهیم

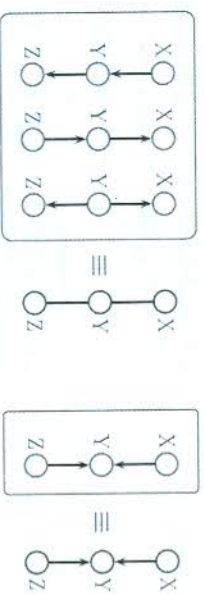
$$A(k) \in \{A_1, A_2, \dots, A_m\}, \quad m = r^{n-1} \quad (11)$$

انتخاب زیرمجموعه $A(k)$ از میان مجموعه اقدام‌های قابل انجام توسط آنامانا، غالباً توسط شرایط بیرونی تحمیل می‌شود. می‌توان این گونه فرض کرد که این زیرمجموعه از اقدام‌های آنامانا را که از این به بعد مجموعه اقدام‌های فعال آنامانا نامیده می‌شود، توسط یک عامل بیرونی و با توزیع احتمال

$$\psi_i(k) = \{\psi_1(k), \psi_2(k), \dots, \psi_m(k)\} \quad (12)$$

$$\hat{p}_i(k) = \text{Prob}[\alpha(k) = \alpha_i | A(k)], \quad \alpha_i \in A(k) \quad (13)$$

انتخاب شود. اگر تعریف کنیم



شکل ۹: نمایش DAGهای هم‌راز مارکوف به شکل PDAG [۳۹] (صفحه ۹۱۴ شکل ۳۶۶).

تعریف ۴: یک کمان را معکوس‌پذیر گویند اگر معکوس کردن آن منجر به ایجاد گرافی هم‌راز با گراف اولیه شود. فضای PDAGهای کامل و CPDAGها یا گراف‌های پایه به عنوان مجموعه تمام گراف‌های فاقد دور جزئی جهت‌دار تعریف می‌شود که تنها دارای بال‌های فاقد جهت و کمان‌های (بال‌های جهت‌دار) معکوس‌ناپذیر است.

تعریف ۴: یک معیار امتیازدهی را هم‌راز گویند اگر روی DAGهای هم‌راز مقادیر یکسانی تولید کند.

نویسندگان در [۳۰] نشان دادند که احتمال همگرا شدن جست و جو در این فضا با امتیازهای هم‌راز نسبت به جست و جو در فضای DAGها بیشتر است. این مفهوم توسط [۳۲] و [۳۲] در روش‌های جست و جوی ساختاری پیشنهاد شده است. نویسنده در [۳۳] یک روش جست و جوی هم‌رازی حریصانه (GES) پیشنهاد داده که از CPDAGها برای نمایش کلاس‌های هم‌رازی مارکوف استفاده کرده است. این روش در دو مرحله عمل می‌کند: مرحله اول با یک گراف خالی شروع می‌شود و افزودن کمان‌ها به آن آغاز می‌شود. افزودن کمان‌ها تا زمانی که موجب بهبود امتیاز شود ادامه پیدا می‌کند. در مرحله دوم سعی می‌کند کمان‌های نامرتب را حذف کند.

۳-۲ آنامانای یادگیر

آنامانای یادگیر تصادفی یک واحد تصمیم‌گیرنده تطبیقی است که فرایند یادگیری در آن از طریق تعاملش با محیط صورت می‌گیرد. آنامانای یادگیر، مجموعه‌ای از اقدام‌های قابل انجام دارد و این اقدام‌ها به تصادف و بر اساس یک بردار توزیع احتمال، انتخاب شده و به عنوان ورودی به محیط اعمال می‌شوند. محیط، اقدام انجام‌شده را به کمک یک سیگنال تقویتی بازخوردی مورد ارزیابی قرار می‌دهد. آنامانای یادگیر بر اساس سیگنال بازخوردی حاصل، بردار توزیع احتمال انتخاب اقدام‌ها را به روز رسانی می‌کند. هدف آنامانا، پیدا کردن اقدام بهینه در میان مجموعه اقدام‌های قابل انجام است، اقدامی که بیشترین پاداش را از محیط دریافت کند.

ارتباط آنامانای تصادفی با محیط در شکل ۱۰ نشان داده شده است. از این مجموعه به همراه الگوریتم یادگیری تحت عنوان آنامانای یادگیر تصادفی نام برده می‌شود. به این ترتیب آنامانای یادگیر تصادفی را می‌توان با چهارتایی (۸) تعریف کرد

$$SLA \equiv \{\alpha, \beta, p, T, c\} \quad (8)$$

که در این رابطه $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه اقدام‌های آنامانا/مجموعه ورودی‌های محیط و $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودی‌های آنامانا/مجموعه خروجی‌های محیط را نشان می‌دهد. $T \equiv \varphi \rightarrow \alpha$ بردار احتمال اقدام‌های آنامانا، $p \equiv \{p_1, p_2, \dots, p_r\}$ الگوریتم یادگیری و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمالات جریمه و معرف محیط است.

این قواعد تابعی از سطح فعالیت فعلی هر آتاماتا و سطح فعالیت آتاماتاهای مجاور است و بسته به مسأله‌ای که eDLA برای حل آن به کار گرفته می‌شود، می‌تواند متغیر باشد.

$S^* = (s_1^*, s_2^*, \dots, s_n^*)$ وضعیت آغازین سیستم نامیده می‌شود. مجموعه شرایط پایانی است. $F = \{S^F | S^F = (s_1^F, s_2^F, \dots, s_n^F)\}$ مجموعه شرایطی را بر حسب سطح فعالیت آتاماتاهای مشخص می‌کند که در صورت تحقق حداقل یکی از آنها سیستم به وضعیت پایانی منتقل می‌شود. F حداقل یک عضو بدیهی دارد که در آن سطح فعالیت تمام آتاماتاهای off است.

C یک تابع خاص است که بر اساس سطح فعالیت آتاماتاهای و مسأله‌ای که eDLA برای حل آن طراحی شده است، آتاماتایی را از میان مجموعه آتاماتاهای موجود در eDLA انتخاب می‌کند. C را تابع آتش^۱ می‌نامیم.

هر آتاماتای یادگیر در eDLA متناسب با درجه‌اش در گراف ارتباطی (در حالتی که گراف ارتباطی G فاقد جهت باشد) و یا درجه خروجی‌اش در گراف ارتباطی (در حالتی که گراف ارتباطی G جهت‌دار باشد) اقدام برای انجام‌دادن دارد.

نحوه انجام کار eDLA به این صورت است که eDLA از وضعیت S^* آغاز کرده و بر اساس مجموعه قواعد P نسبت به تغییر سطح فعال‌بودن آتاماتاهای اقدام می‌کند. تغییر سطح فعال‌بودن آتاماتاهای تا زمان حل کامل مسأله ادامه پیدا می‌کند. در هر زمان آتاماتایی که در سطح فعالیت Fire قرار دارد، اقدامی را انتخاب کرده و آن را به محیط اعمال می‌کند. ضمن این که بر اساس قواعد ارتباطی، مجموعه‌ای از آتاماتاهای مجاور با آتاماتای Fire که در پایین‌ترین سطح فعال‌بودن (Passive) قرار دارند، ارتقای سطح می‌یابند (به سطح Active). هر آتاماتا پس از Fire شدن و انجام اقدام به وضعیت Off تغییر وضعیت می‌دهد. انتخاب آتاماتای Fire از میان مجموعه آتاماتاهای Active صورت می‌گیرد و غالباً به صورت تصادفی انجام می‌شود.

از eDLA تا کنون برای حل برخی مسایل استفاده شده است. نویسنده در [۳۵] مسأله یادگیری پارامتری شبکه‌های بیزی به کمک آتاماتای یادگیر توزیع‌شده توسعه‌یافته مورد بررسی قرار گرفته است. در [۳۸] روشی مبتنی بر آتاماتای یادگیر برای حل مسأله بهینه‌سازی جایگشت ارائه شده است. در [۳۹] از روش پیشنهادی در [۳۸] برای حل مسأله درخت پوشای کمینه تصادفی استفاده شده است. در [۴۰] نویسنده از eDLA برای نمونه‌گیری در شبکه‌های اجتماعی استفاده کرده است. در [۴۱] ساختار eDLA برای حل مسایل یافتن زیرگراف بهینه در گراف‌های تصادفی به صورت عام مورد بررسی و عملکرد آن به صورت دقیق مورد ارزیابی قرار گرفته است.

۴- الگوریتم پیشنهادی

الگوریتم پیشنهادی که در این بخش ارائه می‌شود به جای جست و جو در فضای DAGها به جست و جو در فضای جایگشت‌ها پرداخته و از روش امتیازدهی K_2 برای ارزیابی جایگشت ایجادشده در هر مرحله استفاده می‌کند. همان گونه که قبلاً متذکر شدیم، فضای جایگشت‌ها نسبت به فضای DAGها محدودتر است. علاوه بر این انتخاب معیار K_2 در این مقاله به دلیل کاهش سربار محاسباتی از طریق ذخیره‌کردن نتایج جزئی میانی (Cahce کردن) و کاهش سربار محاسباتی روش پیشنهادی

(احتمال انتخاب عمل از میان مجموعه اقدام‌های فعال، به شرط آن که مجموعه اقدام‌های فعال آتاماتا برابر با $A(k)$ باشد) در این صورت

$$\hat{p}_i(k) = \frac{p_i(k)}{K(k)} \quad (۱۴)$$

در (۱۴) $K(k)$ مجموع احتمالات تمام اقدام‌های فعال (اقدام‌های عضو $A(k)$) است و داریم

$$K(k) = \sum_{\alpha_i \in A(k)} p_i(k) \quad (۱۵)$$

که در آن

$$p_i(k) = \text{Prob}[\alpha(k) = \alpha_i] \quad (۱۶)$$

احتمال انتخاب اقدام α_i در میان مجموعه تمام اقدام‌های آتاماتا (اعم از این که فعال باشد یا نباشد) است. بدین ترتیب نحوه انتخاب اقدام توسط آتاماتا یا مجموعه اقدام متغیر بدین صورت است که فرض کنید آتاماتا مجموعه اقدام‌های فعال $A(k)$ را داشته باشد. ضریب نرمال‌کننده $K(k) = \sum_{\alpha_i \in A(k)} p_i(k)$ برای این مجموعه از اقدام‌های فعال محاسبه شده و بردار احتمال انتخاب اقدام‌های آتاماتا مطابق با رابطه بالا نرمال‌سازی می‌شود (به گونه‌ای که مجموع احتمال انتخاب اقدام‌های فعال همچنان ۱ باشد). پس از این آتاماتا بر اساس بردار جدید $\hat{p}_i(k)$ یکی از اقدام‌های (فعال) خود را انتخاب کرده و به محیط اعمال می‌کند. در مرحله به روز رسانی نیز بردار $\hat{p}_i(k)$ مطابق با الگوریتم یادگیری مورد استفاده توسط آتاماتا به روز رسانی می‌شود و سپس با استفاده مجدد از رابطه بالا بردار $p_i(k) = \hat{p}_i(k) \cdot K(k)$ به دست می‌آید.

نشان داده شده در صورتی که الگوریتم یادگیری مورد استفاده L_{R-1} باشد، این روش ویژگی‌های ϵ -optimality و absolute expediency را دارد [۳۴].

۳-۲ آتاماتای یادگیر توزیع‌شده توسعه‌یافته

آتاماتای یادگیر توزیع‌شده توسعه‌یافته شبکه‌ای است از آتاماتاهای یادگیر مرتبط با هم که تحت نظارت مجموعه‌ای از قواعد ارتباطی که ناظر بر ترتیب و نحوه فعالیت آنهاست برای حل یک مسأله خاص با یکدیگر همکاری دارند [۳۵] تا [۳۷]. در آتاماتای یادگیر توزیع‌شده توسعه‌یافته، هر آتاماتا دارای یک سطح فعال‌بودن است و زمان و نحوه تغییر سطح فعالیت آتاماتاهای به تبعیت از شرایط مسأله، تغییر می‌کند. در آتاماتای یادگیر توزیع‌شده توسعه‌یافته در هر زمان تنها یک آتاماتا در سطحی قرار می‌گیرد که قادر به انجام اقدام و اعمال آن بر محیط است. به صورت فرمال می‌توان یک آتاماتای یادگیر توزیع‌شده توسعه‌یافته را به صورت $eDLA = \{A, E, S, P, S^*, F, C\}$ تعریف کرد که در آن $E \subseteq A \times A$ مجموعه‌ای از آتاماتاهای یادگیر است، $A = \{1, 2, \dots, n\}$ مجموعه ارتباطات سلسله مراتبی میان اعضا A بوده و $G = (A, E)$ گراف ارتباطی نامیده می‌شود.

$S = \{s_1, s_2, \dots, s_n\}$ مجموعه‌ای از مقادیر موسوم به سطح فعالیت است که سطح فعال‌بودن هر آتاماتایی متعلق به A را مشخص می‌کند. هر یک از s_i ها می‌توانند یکی از مقادیر Active, Passive, Fire, Off باشند که آنها را به ترتیب با Pa, Ac, Fi و Of نمایش می‌دهیم. Fire بالاترین سطح فعالیت و Passive و Off پایین‌ترین سطوح فعالیت هستند با این تفاوت که اگر آتاماتا به سطح فعالیت off برسد، دیگر هرگز تغییر وضعیت نمی‌دهد اما در حالت Passive امکان تغییر وضعیت وجود دارد. مجموعه‌ای محدود از قواعد حاکم بر سطح فعالیت آتاماتاهای است.

وضعیت Active قرار می‌گیرند. آتاماتاها به ترتیب از ریشه Fire شده و با انجام اقدام، آتاماتای بعدی که بایستی Fire شود را معین می‌کنند. آتاماتاها از نوع تصادفی با تعداد اقدام متغیر هستند [۳۴]. با انجام یک اقدام توسط هر آتاماتا، سایر آتاماتاها یی که در وضعیت Active قرار دارند مجموعه اقدام‌های خود را به گونه‌ای تغییر می‌دهند که موجب ایجاد حلقه در جواب نهایی نشود. این کار با بلاک کردن اقدامی که موجب پیدایش حلقه خواهد شد صورت می‌گیرد.

محاسبه امتیاز جایگشت: با استفاده از جایگشت تعیین شده در مرحله قبل و داده‌ها نمونه موجود یک DAG سازگار با جایگشت بر اساس الگوریتم K2 ایجاد و امتیاز آن بر اساس (۶) و (۷) محاسبه می‌شود. eDLA از این مقدار برای تعیین پاداش یا جریمه اقدام‌های صورت گرفته توسط آتاماتاها استفاده می‌کند. این کار با مقایسه مقدار امتیاز با یک مقدار آستانه صورت می‌گیرد. مقدار آستانه پیشینه مقدار امتیاز DAG‌های قبلی را نشان می‌دهد.

ارزیابی اقدام انجام شده توسط eDLA: بر اساس مقایسه انجام شده در مرحله قبل کلیه آتاماتاها یی eDLA جریمه یا پاداش داده می‌شوند. اگر امتیاز جایگشت ایجاد شده از مقدار آستانه بیشتر باشد، کلیه آتاماتاها پاداش داده می‌شوند وگرنه جریمه می‌شوند. الگوریتم جریمه یا پاداش می‌تواند یکی از الگوریتم‌های یادگیری خطی مورد بحث در بخش قبلی باشد.

به روز رسانی مقدار آستانه مقایسه و آماده سازی eDLA برای اجرای بعدی: مقدار آستانه به روز رسانی می‌شود. کلیه آتاماتاها دوباره به وضعیت Active برمی‌گردند و تمامی اقدام‌های بلاک شده در مراحل قبل به وضعیت قابل انجام برمی‌گردند. eDLA برای اجرای بعدی آماده می‌شود. چنان که ملاحظه می‌شود در این الگوریتم از ساختار eDLA برای هدایت جایگشت‌ها به سمت بهترین جایگشت که بتواند معیار K2 را بیشینه کند استفاده می‌کنیم. شبه کد روش جدید پیشنهادی در الگوریتم شکل ۱۱ ارائه شده است. نمونه‌ای از اجرا را نیز در شکل ۱۲ مشاهده می‌کنید.

۵- بررسی ریاضی روش پیشنهادی

در این قسمت به بررسی درستی عملکرد الگوریتم پیشنهادی در یافتن جایگشت بهینه‌ای که منجر به یک DAG سازگار با داده‌های نمونه با بیشینه امتیاز گردد می‌پردازیم. برای این منظور فرض کنید δ_i یک جایگشت باشد و d_i احتمال آن را نشان دهد که ساختار DAG القایی توسط این جایگشت بر اساس الگوریتم K2 بیشینه امتیاز را کسب کرده و توسط محیط پاداش داده شود. $c_i = 1 - d_i$ احتمال بهینه نبودن ساختار القایی حاصل از جایگشت ایجاد شده توسط الگوریتم ۴ را نشان می‌دهد. ساختار کلی اثبات به این صورت است: در مرحله اول نشان می‌دهیم که احتمال انتخاب جایگشت بهینه (حاصل ضرب احتمال انتخاب اقدام‌ها متناظر با عناصر جایگشت و تشکیل دهنده مسیر همیتونی در گراف eDLA) برای مقادیر به اندازه کافی بزرگ از تکرار الگوریتم یک زیرمارتینگل است. در مرحله دوم کران‌هایی برای این احتمال برقرار کرده و نشان می‌دهیم با انتخاب مقدار مناسب برای پارامتر نرخ یادگیری در الگوریتم L_{R-1} توسط هر یک از آتاماتاها در ساختار eDLA می‌توان این کران‌ها را به اندازه دلخواه به مقدار ۱ نزدیک کرد. روش استفاده شده در این بخش همان است که در [۴۱] مورد استفاده قرار گرفته است.

لم ۱: اگر $d_i(k)$ و $c_i(k)$ احتمالات مربوط به جریمه و تنبیه شدن جایگشت σ_i حاصل شده در k امین تکرار را نشان دهند و $q(k)$ بر اساس الگوریتم پیشنهادی ارائه شده تغییر کند، در این صورت امید ریاضی

ورودی: D : مجموعه نمونه‌ها، n : تعداد متغیرها، u : حداکثر تعداد والد‌های یک گره، آستانه‌های K و P_{TH}
خروجی: ماتریس مجاورت DAG متناظر با شبکه بیزی در قالب بردار Parents که لیست والد‌های هر گره را مشخص می‌کند.

Construct $G = (V, V \times V)$ with $V = \{0, 1, 2, \dots, n\}$;

Construct an eDLA from graph G ;

$k \leftarrow 0, \Pi \leftarrow \phi, TH_k \leftarrow -\infty$;

repeat

$P \leftarrow \phi; A \leftarrow V; F \leftarrow \phi; O \leftarrow \phi$;

Let u denotes the position in permutation and initially set to 1;

Select $u = \{0\} \in A$;

$F \leftarrow F \cup u; A \leftarrow A - F$;

while $|\Pi| \leq n$

$e = (u, \pi(u)) = \text{selectaction}(u)$;

$\Pi \leftarrow \Pi \cup \pi(u)$;

All LA in A prunes its action-set

$O \leftarrow O \cup F; F \leftarrow \phi; u \leftarrow \pi(u)$;

$F \leftarrow F \cup u$;

end while

if $|\Pi| = n$ **then**

if $K2(D, n, u, \Pi) > TH_k$ **then**

Reward all $LA \in O$;

else

penalize all $LA \in O$;

end if

$TH_k = \max(TH_k, K2(D, n, u, \Pi))$;

else

Do nothing;

end if

$k \leftarrow k + 1$;

$q_\pi = Pr(\Pi) = \prod_{\pi(u) \in \Pi} p_{\pi(u)}^*$;

Enable all disables actions of $LA \in O$;

until $(q_\pi > P_{TH} \vee k > K)$

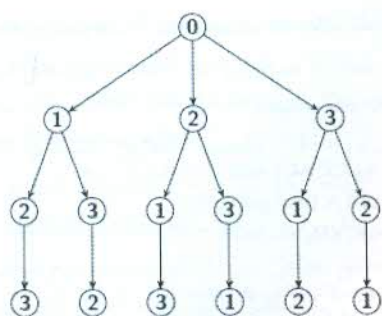
شکل ۱۱: الگوریتم پیشنهادی مبتنی بر eDLA برای یادگیری ساختار شبکه بیزی.

صورت گرفته است. جست و جو در فضای جایگشت‌ها با نگاشت یک جایگشت به یک مسیر همیتونی در یک گراف کامل صورت می‌گیرد. برای این کار از روش پیشنهادی در [۳۸] و [۴۱] استفاده شده است.

برای این منظور یک گراف کامل فاقد جهت با تعداد $n+1$ رأس (یک رأس بیشتر از تعداد متغیرها) به عنوان گراف ارتباطی eDLA ایجاد می‌شود. هر یک از رؤس گراف ایجاد شده متناظر با یکی از متغیرها و یک رأس به عنوان ریشه است. هر یال $e_{(i,j)}$ که رؤس i و j را در گراف به یکدیگر متصل می‌کند متناظر با اقدام α_i^j از آتاماتای LA_i یا اقدام α_j^i از آتاماتای LA_j است. در هر بار اجرا، eDLA از آتاماتای متناظر با رأس ریشه شروع کرده و بر اساس بردار احتمال انتخاب اقدام‌های آن اقدامی را انجام می‌دهد. این کار باعث تغییر وضعیت یکی از آتاماتاها یی واقع شده در سمت دیگر یال متناظر با اقدام انجام شده به وضعیت Fire می‌شود. Fire شدن آتاماتاها تا زمانی که کلیه آتاماتاها Fire شوند ادامه پیدا می‌کند. با پایان این پروسه یک جایگشت (ترتیب) از متغیرها به دست می‌آید. بر اساس جایگشت ایجاد شده بهترین DAG بر اساس معیار K2 ایجاد می‌شود. مقدار K2 متناظر با جایگشت ایجاد شده به عنوان معیاری برای ارزیابی اقدام‌های صورت گرفته توسط آتاماتاها استفاده می‌شود.

به صورت خلاصه می‌توان یک دور اجرای الگوریتم را متشکل از گام‌های زیر دانست:

تشکیل یک جایگشت توسط eDLA: در ابتدای کار تمامی آتاماتاها در



شکل ۱۳: درخت جایگشت‌های روی سه عنصر با شماره‌های ۱، ۲ و ۳.

جدول ۱: نحوه تعریف احتمال انتخاب هر جایگشت برای جایگشت روی سه عنصر.

جایگشت	احتمال
$\sigma_1 = (1, 2, 3)$	$q_1(k) = p_1^-(k)p_1^+(k)$
$\sigma_2 = (1, 3, 2)$	$q_2(k) = p_1^-(k)p_2^+(k)$
$\sigma_3 = (2, 1, 3)$	$q_3(k) = p_2^-(k)p_1^+(k)$
$\sigma_4 = (2, 3, 1)$	$q_4(k) = p_2^-(k)p_2^+(k)$
$\sigma_5 = (3, 1, 2)$	$q_5(k) = p_3^-(k)p_1^+(k)$
$\sigma_6 = (3, 2, 1)$	$q_6(k) = p_3^-(k)p_2^+(k)$

$$p_i^+ \uparrow p_i^- \downarrow = \{p_i^-(1-a) + a\} \left\{ p_i^+ \left(1 - \frac{a}{p_i^-(1-a) + a} \right) \right\} = \quad (21)$$

$$\{p_i^-(1-a) + a\} p_i^+ - a p_i^+ = p_i^+ p_i^-(1-a) = q_i(1-a)$$

و

$$p_i^+ \uparrow p_i^+ \uparrow = \{p_i^-(1-a) + a\} \times \left\{ p_i^+ \left(1 - \frac{a}{p_i^-(1-a) + a} \right) + \frac{a}{p_i^-(1-a) + a} \right\} = \quad (22)$$

$$\{p_i^-(1-a) + a\} p_i^+ - a p_i^+ + a =$$

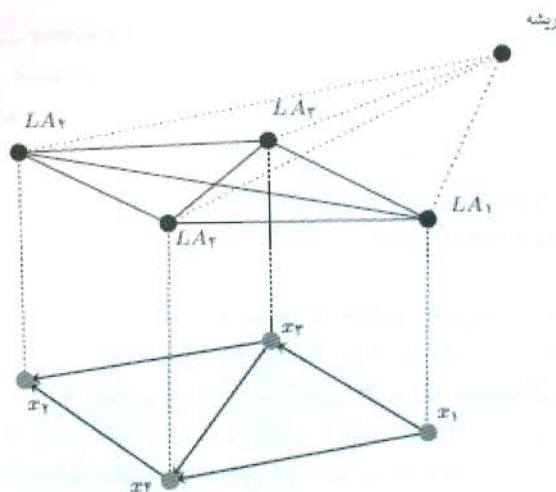
$$p_i^+ p_i^-(1-a) + a = q_i(1-a) + a$$

با جایگذاری مقادیر (۲۰) تا (۲۲) در (۱۹) خواهیم داشت

$$E[q_i(k+1)|q(k)] = q_i(k) \{c_i(k)q_i(k) + d_i(k)q_i(k)(1-a)\} + q_i(k) \{c_i(k)q_i(k) + d_i(k)q_i(k)(1-a)\} + q_i(k) \{c_i(k)q_i(k) + d_i(k)q_i(k)(1-a)\} + q_i(k) \{c_i(k)q_i(k) + d_i(k)\{q_i(k)(1-a) + a\}\} + q_i(k) \{c_i(k)q_i(k) + d_i(k)q_i(k)(1-a)\} + q_i(k) \{c_i(k)q_i(k) + d_i(k)q_i(k)(1-a)\} \quad (23)$$

با ساده‌سازی جبری (۲۳) داریم

$$E[q_i(k+1)|q(k)] = q_i(k) \{q_i(k) - a d_i(k)q_i(k)\} + q_i(k) \{q_i(k) - a d_i(k)q_i(k)\} + q_i(k) \{q_i(k) - a d_i(k)q_i(k)\} + q_i(k) \{q_i(k) - a d_i(k)q_i(k) + a d_i(k)\} + q_i(k) \{q_i(k) - a d_i(k)q_i(k)\} + q_i(k) \{q_i(k) - a d_i(k)q_i(k)\} = q_i(k) - \{a q_i(k) \sum_{j=1, j \neq i}^r q_j(k) d_j(k)\} + a d_i(k)q_i(k) \{1 - q_i(k)\} = q_i(k) + a q_i(k) \sum_{j=1, j \neq i}^r q_j(k) \{d_i(k) - d_j(k)\} \quad (24)$$



شکل ۱۴: تناظر بین eDLA و متغیرها و مثالی از اجرای الگوریتم پیشنهادی: eDLA مسیر $0 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 2$ را انتخاب می‌کند که به ترتیب $Ord = \{1, 3, 4, 2\}$ نگاشت می‌شود. K2 بر اساس این ترتیب DAG سازگار $G = \{\{\}, \{1, 3, 4\}, \{1\}, \{3, 1\}\}$ را به عنوان ساختار با امتیاز بیشینه تعیین می‌کند.

شرطی $E[q_i(k+1) - q_i(k) | q(k)]$ مطابق با (۱۷) تعریف می‌شود

$$E[q_i(k+1) - q_i(k) | q(k)] \geq a q_i(k) \sum_{j=1}^r q_j(k) [d_i(k) - d_j(k)] \quad (17)$$

اثبات: برای اثبات، یک جایگشت روی سه عنصر را در نظر گرفته و محاسبات را انجام می‌دهیم. اگر از الگوریتم ۴ استفاده شده باشد، ۶ جایگشت ممکن برای این سه عنصر از طریق طی کردن مسیرهای مشخص شده در درخت شکل ۱۳ ساخته می‌شوند.

احتمال متناسب به هر یک از ۶ جایگشت این مثال در جدول ۱ نشان داده شده است. ضمناً فرض می‌کنیم مطابق آنچه که در [۴۲] پیشنهاد شده است نرخ یادگیری در هر یک از سطوح مطابق با (۱۸) تغییر می‌کند

$$a_{\pi_{i,j}}(k) = \frac{a_{\pi_{i,j-1}}(k)}{p_{\pi_{i,j-1}}(k+1)} \quad (18)$$

در (۱۸) مقدار نرخ یادگیری در زمان k برای j آتاماتای واقع در مسیر π_i را نشان می‌دهد [۴۳]. ما در ادامه، محاسبات را برای جایگشت σ_i انجام خواهیم داد

$$E[q_i(k+1)|q(k)] = \sum_{\beta=1}^r \sum_{j=1}^r E[q_i(k+1)|q(k), \alpha_j, \beta] \times P[\beta | \alpha_j] \times P[\alpha_j | q(k)] = q_i(k) \{c_i(k)q_i(k) + d_i(k)p_i^- \downarrow p_i^+ \downarrow\} + q_i(k) \{c_i(k)q_i(k) + d_i(k)p_i^- \downarrow p_i^+ \downarrow\} + q_i(k) \{c_i(k)q_i(k) + d_i(k)p_i^+ \uparrow p_i^- \downarrow\} + q_i(k) \{c_i(k)q_i(k) + d_i(k)p_i^+ \uparrow p_i^+ \uparrow\} + q_i(k) \{c_i(k)q_i(k) + d_i(k)p_i^- \downarrow p_i^+ \downarrow\} + q_i(k) \{c_i(k)q_i(k) + d_i(k)p_i^- \downarrow p_i^+ \downarrow\} \quad (19)$$

منظور از \downarrow و \uparrow در (۱۹) به ترتیب کاهش و افزایش در مقدار احتمال یک اقدام بر اساس روش L_{R-I} است اما

$$p_i^- \downarrow p_i^+ = p_i^-(1-a)p_i^+ = q_i(1-a) \quad (20)$$

در نتیجه

$$E[q_+(k+1)|q(k)] - q_+(k) = aq_+(k) \sum_{\substack{j=1 \\ j \neq i}}^r q_j(k) \{d_+(k) - d_j(k)\} \quad (25)$$

و حکم ثابت است. ■

لم ۲: با فرض این که جایگشت σ_i جواب مسأله یافتن جایگشت بهینه باشد، در این صورت مقدار $\Delta q_i(k+1) = E[q_i(k+1) - q_i(k) | q(k)]$ همواره مثبت است.

اثبات: متوسط تغییر در مقدار $q_i(k)$ برای جایگشت σ_i که از ضرب مقادیر احتمال اقدام‌های متناظر با یال‌های تشکیل‌دهنده جایگشت σ_i تشکیل می‌شود مطابق با لم قبل برابر است با

$$\Delta q_i(k+1) = E[q_i(k+1) - q_i(k) | q(k)] \geq aq_i(k) \sum_{\substack{j=1 \\ j \neq i}}^r q_j(k) \{d_i(k) - d_j(k)\} \quad (26)$$

چون جایگشت σ_i بهینه است بنا بر قانون قوی اعداد بزرگ برای مقادیر بزرگ k مقدار $\{d_i(k) - d_j(k)\}$ برای هر $j \neq i$ مثبت است و در نتیجه سمت راست (۲۶) مثبت و بنابراین برای مقادیر بزرگ k طرف راست (۲۶) مقداری مثبت است و در نتیجه $\Delta q_i(k+1) \geq 0$ ■

گزاره ۱: $\lim_{k \rightarrow \infty} q_i(k) = q_i^* \in \{0, 1\}$ با احتمال ۱ وجود دارد.

لم ۲ بیان‌کننده آن است که $\{q_i(k)\}$ یک زیرمارتینگل است. با استفاده از قضایای حدی مارتینگل نتیجه می‌شود که $\lim_{k \rightarrow \infty} q_i(k) = q_i^*$ با احتمال ۱ وجود دارد. علاوه بر این اگر فرض کنیم $q_i(k) \neq 0$ و $q_i(k) \neq 1$ در این صورت با یک احتمال ناصفر به ازای تمام مقادیر k داریم $q_i(k) = q_i(k+1)$ و بنابراین الزاماً $q_i^* \in \{0, 1\}$ ■

گزاره ۱ نشان می‌دهد که احتمال انتخاب جایگشت بهینه یکی از دو مقدار ۰ یا ۱ است. به عبارت دیگر الگوریتم پیشنهادی همواره همگرا می‌شود اما احتمال این که به جوابی غیر از جواب بهینه همگرا شود وجود دارد. در این بخش نشان خواهیم داد که تحت شرایطی و با انتخاب پارامترهای مناسب برای الگوریتم یادگیری می‌توان احتمال همگرایی الگوریتم پیشنهادی به جواب بهینه را به اندازه کافی به ۱ نزدیک کرد. برای این کار از روش پیشنهادی [۴۴] که در موارد مشابه نیز مورد استفاده قرار گرفته است ([۴۳] و [۴۵]) استفاده می‌کنیم.

فرض کنید $V_r = \{e_1, e_2, \dots, e_r\}$ مجموعه حالت‌های جاذب فرایند $\{q(k)\}$ باشد و $q^* \in V_r$ حالتی را نشان دهد که $q(k)$ به آن همگرا می‌شود. تعریف می‌کنیم

$$\Gamma_i[q] = \Pr[q^* = e_i | q(\cdot) = q] \quad (27)$$

$\Gamma_i[q]$ احتمال همگرایی الگوریتم به جایگشت σ_i با شروع از بردار احتمال q را نشان می‌دهد. فرض کنید

$$S_r = \{q | q \geq 0, \sum_{j=1}^r q_j = 1, j = 1, 2, \dots, r\} \quad (28)$$

فرض کنید $C(S_r): S_r \rightarrow \mathbb{R}$ فضای حالت تمام توابع مشتق‌پذیر پیوسته حقیقی مقدار با مشتق محدود تعریف شده روی S_r باشد که \mathbb{R} یک خط حقیقی است. اگر $g[q] \in C(S_r)$ باشد، الگوریتم ۴ پیشنهادی نظیر یک عملگر U میانگین $Ug[q] = E[g(q(k+1)) | q(k) = q]$ عمل می‌کند که در این رابطه $E[\cdot]$ نشان‌دهنده امید ریاضی است. نشان

داده شده است [۴۴] و [۴۵] که U عملگری خطی است و توابع نامنفی را نامنفی نگه می‌دارد. یعنی

$$\text{if } g[q] \geq 0 \Rightarrow Ug[q] \geq 0 \quad \forall q \in S_r$$

نسبت به عملگر U می‌توان دو گروه از توابع موسوم به زیر- و ابر-منظم را تعریف کرد. اگر $Ug[q] \geq g[q]$ تابع $g[q]$ را زیرمنظم و اگر $Ug[q] \leq g[q]$ تابع $g[q]$ را ابرمنظم گویند.

نشان داده شده است [۴۴] که $\Gamma_i[q]$ تنها جواب پیوسته $U\Gamma_i[q] = \Gamma_i[q]$ با شرایط مرزی زیر است

$$\Gamma_i[e_j] = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (29)$$

اما حل معادله تابعی $U\Gamma_i[q] = \Gamma_i[q]$ برای تعیین احتمال انتخاب جایگشت بهینه بر حسب احتمال اولیه توسط الگوریتم پیشنهادی، کار سختی است. برای این کار از یک کران پایین برای این تابع استفاده می‌کنیم. از گزاره قبلی می‌دانیم که احتمال انتخاب جایگشت بهینه $q_i^* \in \{0, 1\}$ است و تعریف می‌کنیم

$$\Phi_i[x, q] = \frac{\exp \frac{-xq_i - 1}{a}}{\exp \frac{-x}{a} - 1} \quad (30)$$

که در آن x بایستی انتخاب شود. واضح است که تابع تعریف شده در (۳۰) شرایط مرزی تعریف شده در (۲۹) را ارضا می‌کند. نشان خواهیم داد که $\Phi_i[x, q]$ یک تابع زیرمنظم بوده و لذا قادر است که یک کران پایین برای $\Gamma_i[q]$ تعریف کند. می‌توان نشان داد که $\Phi_i[x, q]$ وقتی زیرمنظم است که $\theta_i[x, q] = \exp(-xq_i/a)$ ابرمنظم باشد $\exp(-x/a) - 1$ همواره منفی است).

لم ۳: اگر σ_i جایگشت بهینه $\theta_i[x, q] = \exp(-xq_i/a)$ باشد که در آن q_i احتمال انتخاب جایگشت σ_i و a نرخ یادگیری مورد استفاده در الگوریتم است، در این صورت x ای وجود دارد که $\theta_i[x, q]$ ابرمنظم خواهد بود.

اثبات: برای تعیین شرایطی که تحت آن شرایط، تابع $\theta_i[x, q]$ یک تابع ابرمنظم است، از تعریف ابرمنظم و محاسبه مقدار امید ریاضی استفاده می‌کنیم. فرض کنید جایگشت σ_i جواب بهینه باشد، در این صورت

$$U\theta_i[x, q] = E[\exp \frac{-xq_i(k+1)}{a} | q(k) = q] \quad (31)$$

برای محاسبه مقدار امید ریاضی در (۳۱) به این ترتیب عمل می‌کنیم

$$\begin{aligned} E[\exp \frac{-xq_i(k+1)}{a} | q(k) = q] &= \sum_{j=1}^r E[\exp \frac{-xq_i(k+1)}{a} | q(k) = q, \sigma_j] q_j = \\ &= \sum_{j=1}^r \exp \frac{-xq_i(k+1)}{a} \times q_j \times d_j + \\ &= \sum_{j=1}^r \exp \frac{-xq_i(k)}{a} \times q_j \times (1 - d_j) \end{aligned} \quad (32)$$

رابطه (۳۲) در حقیقت تغییرات در احتمال انتخاب جایگشت مورد نظر σ_i را تحت هر نوع شرایطی و در یک گام از اجرای الگوریتم پیشنهادی مورد بررسی قرار می‌دهد. در هر گام الگوریتم پیشنهادی یک جایگشت نظیر σ_j را انتخاب می‌کند. این جایگشت ممکن است توسط محیط با سیگنال تقویتی تنبیه جریمه شود که در این صورت با توجه به استفاده از الگوریتم

$$D(p_n^m) = \frac{\prod_{e(m,n) \in \sigma_i \cap \sigma_j} \{p_n^m + a(1 - p_n^m)\}}{\prod_{e(m,n) \in \sigma_i \cap \sigma_j} \{p_n^m(1 - a)\}} \geq 1 \quad (37)$$

با توجه به نزولی بودن تابع e^{-x} داریم

$$\exp\left\{-\frac{x}{a} q_i(1-a)\right\} \frac{\prod_{e(m,n) \in \sigma_i \cap \sigma_j} \{p_n^m + a(1 - p_n^m)\}}{\prod_{e(m,n) \in \sigma_i \cap \sigma_j} \{p_n^m(1-a)\}} \leq \quad (38)$$

$$\exp\left\{-\frac{x}{a} q_i(1-a)\right\}$$

از (۳۶) و (۳۸) به (۳۹) می‌رسیم

$$U\theta_i[x, q] - \theta_i[x, q] \leq q_i d_i \times \exp\left\{-\frac{x}{a} (q_i + a(1 - q_i))\right\} + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j \times \exp\left\{-\frac{x}{a} q_i(1-a)\right\} - \theta_i[x, q] \sum_{j=1}^r q_j d_j \quad (39)$$

اما

$$\theta_i[x, q] \sum_{j=1}^r q_j d_j = q_i d_i \theta_i[x, q] + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j \theta_i[x, q] \quad (40)$$

فلذا (۴۱) به دست می‌آید و تعریف می‌کنیم

$$V[x] = \begin{cases} \frac{\exp(x) - 1}{x}, & x \neq 0 \\ 1, & x = 0 \end{cases} \quad (42)$$

لذا

$$U\theta_i[x, q] - \theta_i[x, q] \leq -x q_i \theta_i[x, q] ((1 - q_i) d_i V[-x(1 - q_i)] - \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j V[x q_i]) \quad (43)$$

بر منظم بودن $\theta_i[x, q]$ است، لذا شرط لازم برای

$$(1 - q_i) d_i V[-x(1 - q_i)] \geq \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j V[x q_i] \quad (44)$$

یا

$$\sum_{j=1}^r \exp\left\{-\frac{x q_i(k+1)}{a}\right\} q_j d_j = q_i d_i \exp\left\{-\frac{x \prod_{e(m,n) \in \sigma_i} \{p_n^m + a(1 - p_n^m)\}}{a}\right\} + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j \exp\left\{-\frac{x \prod_{e(m,n) \in \sigma_i \cap \sigma_j} \{p_n^m + a(1 - p_n^m)\}}{a} \prod_{e(m,n) \in \sigma_i - \sigma_j} \{p_n^m(1-a)\}\right\} \quad (45)$$

$$U\theta_i[x, q] - \theta_i[x, q] \leq q_i d_i (\exp\left\{-\frac{x}{a} (q_i + a(1 - q_i))\right\} - \theta_i[x, q]) + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j (\exp\left\{-\frac{x}{a} q_i(1-a)\right\} - \theta_i[x, q]) = \quad (46)$$

$$\theta_i[x, q] (q_i d_i \exp\{-x(1 - q_i)\} - 1 + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j (\exp\{x q_i\} - 1)) =$$

$$-x q_i \theta_i[x, q] ((1 - q_i) d_i \frac{\exp\{-x(1 - q_i)\} - 1}{-x(1 - q_i)} - \sum_{\substack{j=1 \\ j \neq i}}^r q_j \times d_j \frac{\exp\{x q_i\} - 1}{x q_i})$$

L_{R-1} توسط آتاماتاها، احتمال انتخاب جایگشت‌ها هیچ تغییری نمی‌کند. حالت دیگر آن است که این جایگشت توسط محیط با سیگنال تقویتی پاداش، مورد تشویق قرار گیرد. در این حالت کلیه انتخاب‌هایی که منجر به انتخاب σ_j شده است پاداش داده خواهند شد (برای مثال جدول ۱ را ببینید). در این حالت چنانچه جایگشتی در هیچ یک از مکان‌های اشتراکی با σ_j نداشته باشد کاملاً جریمه خواهد شد اما جایگشت‌هایی که در برخی از مکان‌ها با σ_j اشتراک دارند هم جریمه و هم پاداش داده خواهند شد.

با ساده‌سازی (۳۲) به (۳۳) و (۳۴) می‌رسیم که در این روابط $p_n^m \downarrow = p_n^m(1-a)$ و $p_n^m \uparrow = p_n^m(1-a) + a$

$$\sum_{j=1}^r \exp\left\{-\frac{x q_i(k)}{a}\right\} \times q_j \times (1 - d_j) = \theta_i[x, q] \sum_{j=1}^r q_j \times (1 - d_j) = \theta_i[x, q] - \theta_i[x, q] \sum_{j=1}^r q_j \times d_j \quad (34)$$

در (۳۳) داریم

$$\prod_{e(m,n) \in \sigma_i - \sigma_j} \{p_n^m(1-a)\} = \frac{\prod_{e(m,n) \in \sigma_i - \sigma_j} \{p_n^m(1-a)\} \prod_{e(m,n) \in \sigma_i \cap \sigma_j} \{p_n^m(1-a)\}}{\prod_{e(m,n) \in \sigma_i \cap \sigma_j} \{p_n^m(1-a)\}} = \quad (35)$$

$$\frac{\prod_{e(m,n) \in \sigma_i} \{p_n^m(1-a)\}}{\prod_{e(m,n) \in \sigma_i \cap \sigma_j} \{p_n^m(1-a)\}}$$

با توجه به $q_i = \prod_{e(m,n) \in \sigma_i} p_n^m$ تا (۳۵) خواهیم داشت

$$U\theta_i[x, q] = q_i d_i \times \exp\left\{-\frac{x}{a} (q_i + a(1 - q_i))\right\} + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j \exp\left\{-\frac{x q_i(1-a)}{a}\right\} D(p_n^m) + \theta_i[x, q] - \theta_i[x, q] \sum_{j=1}^r q_j \times d_j \quad (36)$$

اما روشن است که

(۳۳)

(۴۱)

۶- بررسی تجربی

۱-۶ آزمایش ۱: بررسی عملکرد الگوریتم

در این آزمایش برای بررسی عملکرد الگوریتم از یک شبکه بیزی ساده مطابق آنچه که در شکل ۱ مشاهده می‌کنید استفاده شده است. برای انجام آزمایش از یک نمونه ۱۰۰ تایی انتخاب شده از داده‌های تولید شده بر اساس این شبکه استفاده شده است. شاخص $K2$ برای این نمونه انتخاب شده از داده‌ها در DAG اصلی، $f_{K2}(D, G_{original}) = -216.004$ بود. مطابق با الگوریتم ۴ کار جست و جوی بهترین جایگشت انجام شده است. مبنای ارزیابی اقدام‌های آتاماتا امتیاز $K2$ کسب شده توسط بهترین DAG سازگار با ترتیب پیشنهادی توسط eDLA بوده است. از الگوریتم یادگیری L_{R-1} با نرخ یادگیری ۰/۰۲ برای پاداش اقدام‌های آتاماتا استفاده شده است. کار جست و جوی یافتن بهترین ترتیب ۵۰ بار تکرار شده است. شبکه ایجاد شده بر اساس این الگوریتم را در شکل ۱۴ مشاهده می‌کنید. بر اساس داده‌های نمونه، این ساختار نیز دارای امتیاز $K2$ با مقدار -216.004 می‌باشد.

۲-۶ آزمایش ۲: بررسی عملکرد مقایسه‌ای

برای انجام این آزمایش‌ها از شبکه ASIA [۴۶]، ICU-ALARM [۴۷] و INSURANCE استفاده کرده‌ایم. کلیه شبیه‌سازی‌ها در Matlab صورت گرفته است. فرایند انجام کار بدین صورت بوده است که از شبکه اصلی یک نمونه با تعداد مشخص گرفته شده است. بر اساس این نمونه مشخص و با استفاده از یک معیار امتیازدهی- که در آزمایش‌های این سری از معیار $K2$ استفاده شده است- فرایند جست و جوی یافتن بهترین جایگشت توسط الگوریتم پیشنهادی صورت گرفته است. برای مقایسه از الگوریتم‌های gs، hc و gs2 استفاده شده است.

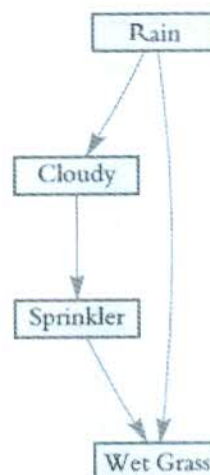
در نخستین آزمایش از شبکه ASIA استفاده شده است. این شبکه دارای ۸ گره و ۸ کمان است. از روی شبکه اصلی یک نمونه ۵۰۰ تایی نمونه‌گیری شده است. سپس از الگوریتم پیشنهادی و الگوریتم‌های hc، gs و gs2 خواسته شده تا یادگیری ساختار شبکه را به کمک داده‌های نمونه انجام دهند. علاوه بر این بر اساس ترتیب اصلی گره‌ها و داده‌های نمونه و بر اساس الگوریتم $K2$ نیز کار یادگیری ساختار شبکه بیزی انجام شده است که ساختار حاصل را Optimal نامیده‌ایم. دلیل انجام این کار آن است که داده‌های نمونه‌گیری شده بر اساس معیار $K2$ نتوانند معرف ساختار اصلی باشند. کلیه مقایسه‌های بین ساختارهای حاصل از هر یک از الگوریتم‌ها با ساختار شبکه Optimal صورت گرفته است. در این آزمایش ساختار شبکه Optimal با ساختار شبکه اصلی یکسان است.

برای انجام مقایسه الگوریتم‌های مختلف با الگوریتم پیشنهادی از معیارهای زیر استفاده شده است:

(۱) زمان اجرا: مدت زمان صرف شده توسط الگوریتم برای یافتن ساختار بر اساس داده‌های نمونه و معیار $K2$

(۲) امتیاز: امتیاز کسب شده توسط شبکه ایجاد شده توسط هر الگوریتم بر اساس معیار $K2$. هر چقدر این امتیاز بزرگ‌تر و به امتیاز شبکه Optimal نزدیک‌تر باشد الگوریتم عملکرد بهتری دارد.

(۳) مقایسه ساختارها: برای مقایسه بین ساختار ایجاد شده توسط هر الگوریتم و ساختار شبکه Optimal از سه شاخص مرتبط با ساختار استفاده کرده‌ایم. تعداد کمان‌های با طرفین و جهت درست، تعداد کمان‌های با طرفین درست و جهت نادرست و تعداد کمان‌های با طرفین و جهت نادرست.



شکل ۱۴: نتیجه اجرای الگوریتم پیشنهادی با کمک داده‌های تولید شده از شبکه بیزی شکل ۱ و بازتولید ساختار شبکه بر اساس این داده‌ها.

$$\frac{V[-x(1-q_i)]}{V[xq_i]} \geq \frac{\sum_{j=1, j \neq i}^r q_j d_j}{(1-q_i) d_i} \quad (45)$$

در (۴۵) داریم

$$\frac{\sum_{j=1, j \neq i}^r q_j \times d_j}{(1-q_i) d_i} = \frac{\sum_{j=1, j \neq i}^r q_j \times d_j}{\sum_{j=1, j \neq i}^r q_j \times d_i} = \frac{\sum_{j=1, j \neq i}^r q_j \times \frac{d_j}{d_i}}{\sum_{j=1, j \neq i}^r q_j} \quad (46)$$

لذا

$$\min_{j \neq i} \frac{d_j}{d_i} \leq \frac{\sum_{j=1, j \neq i}^r q_j \times \frac{d_j}{d_i}}{\sum_{j=1, j \neq i}^r q_j} \leq \max_{j \neq i} \frac{d_j}{d_i} \quad (47)$$

با جایگذاری (۴۷) در (۴۵) شرط ابرمنظم بودن به (۴۸) تقلیل می‌یابد

$$\frac{V[-x(1-q_i)]}{V[xq_i]} \geq \max_{j \neq i} \frac{d_j}{d_i} \quad (48)$$

تعریف می‌کنیم $H[x] = \ln V[x]$. با توجه به محدب بودن $H[x]$ می‌توان نشان داد [۴۴]

$$\frac{1}{V[x]} \leq \frac{V[-x(1-q_i)]}{V[xq_i]} \leq V[-x] \quad (49)$$

از آنجا که $1/V[x]$ تابعی پیوسته و یکنوا کاهشی با $V[0]=1$ است مقدار $x=x^*$ ای وجود دارد که برای تمام مقادیر در بازه $x \in (0, x^*)$ داریم $\max_{j \neq i} (d_j/d_i) = 1/V[x^*] \leq 1/V[x]$. با انتخاب $x=x^*$ نابرابری (۴۵) ارضا خواهد شد که در این صورت $\theta_i[x, q]$ ابرمنظم خواهد بود. ■

قضیه ۱: فرض کنید احتمال متناظر با جایگشت بهینه σ_i را مرحله k ام نشان دهد. اگر بردار $q(k) = [q_i(k)]_{i=1, \dots, r}$ مطابق با الگوریتم پیشنهادی تغییر کند، در این صورت $\lim_{n \rightarrow \infty} q_i(n) = 1$.

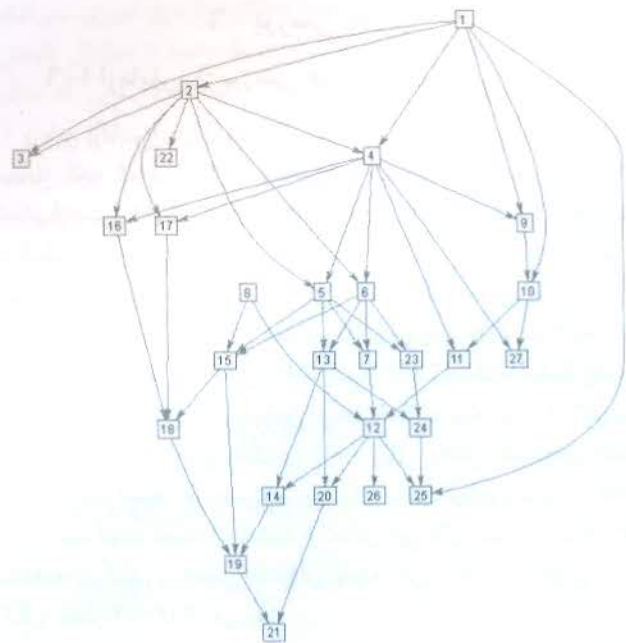
اثبات: با توجه به لم ۳ و تعریف تابع $\Phi_i[x, q]$ مشاهده می‌شود که به ازای هر $\epsilon > 0$ مفروض، ثابت مثبت $\alpha^* < 1$ به گونه‌ای موجود است که نابرابری $1 - \epsilon \leq \Phi_i[x, q] \leq \Gamma_i[q]$ برقرار است. این به آن معناست که $\lim_{n \rightarrow \infty} q_i(n) = 1$ و به این ترتیب اثبات قضیه ۱ کامل می‌شود. ■

جدول ۲: نتایج شاخص‌های مقایسه‌ای برای شبکه ASIA.

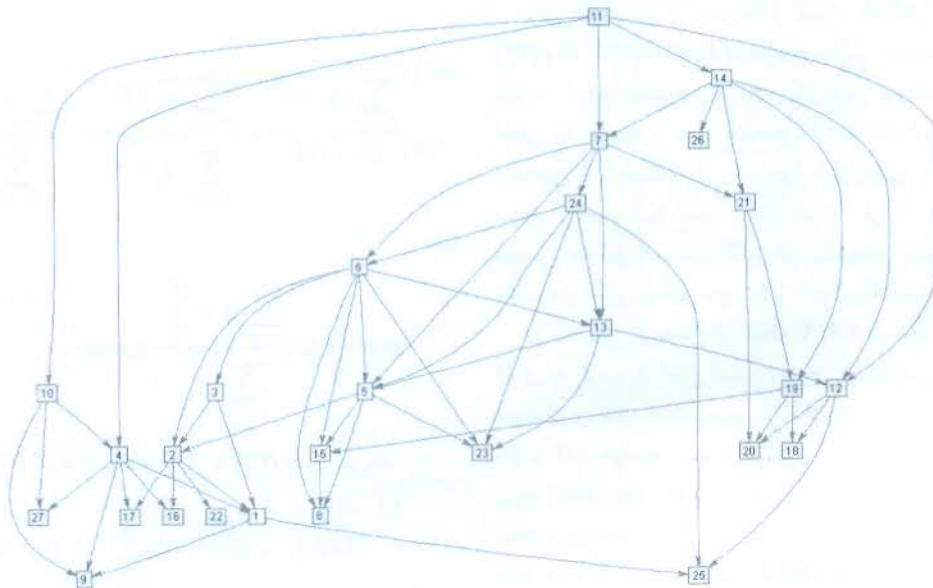
شاخص	HC	GS	GS۲	eDLA
Time	۲۱۷/۴۲	۲۲۱/۳۲	۲۲۹/۱۶	۱۲۷/۳۰
Score _{diff}	-۴۰	-۴۰	-۴۰	۰
n _{correct}	۴	۴	۴	۷
n _{direrr}	۲	۲	۲	۱
n _{error}	۳	۳	۳	۰

جدول ۳: نتایج شاخص‌های مقایسه‌ای برای شبکه INSURANCE با تعداد نمونه ۱۰۰۰۰.

شاخص	HC	GS	GS۲	eDLA
Time	۱۴۶۹۹	۱۲۹۸۶	۱۲۴۷۷	۵۰۸۲/۷
Score _{diff}	-۲۵۹۰	-۲۵۹۰	-۲۲۱۲	-۲۱۷۰
n _{correct}	۳۰	۳۰	۳۰	۲۰
n _{direrr}	۱۰	۱۰	۱۰	۲۰
n _{error}	۴	۴	۴	۱۹



شکل ۱۵: ساختار شبکه Insurance مورد استفاده.



شکل ۱۶: ساختار DAG القایی حاصل از ترتیب ایجاد شده توسط الگوریتم پیشنهادی برای شبکه بیزی Insurance.

جست و جو در DAG روی شبکه بیزی Insurance (شکل ۱۵) مورد بررسی قرار گرفته است. این شبکه دارای ۲۷ گره و ۵۲ یال می‌باشد که ساختار شبکه بیزی ایجاد شده برای داده‌های نمونه‌ای این آزمایش را در شکل ۱۶ و نتایج شاخص‌های مقایسه‌ای آن را در جدول ۳ مشاهده می‌کنید. در شکل ۱۷ نیز ساختار شبکه بیزی پیشنهاد شده به کمک الگوریتم GS بر اساس همین داده‌های نمونه را مشاهده می‌کنید. بر اساس نتایج جدول مشاهده می‌شود که اولاً الگوریتم پیشنهادی به لحاظ زمان اجرا نسبت به الگوریتم‌های جست و جو در فضای ساختارها عملکرد بسیار بهتری دارد. علاوه بر این امتیاز ساختار القایی حاصل از جایگزشت ایجاد شده توسط الگوریتم پیشنهادی در مقایسه با الگوریتم‌های دیگر مورد بررسی در این مقاله بزرگ‌تر و به مقدار واقعی نزدیک‌تر است اما از نقطه نظر ساختار پیشنهادی توسط الگوریتم‌ها به نظر می‌رسد ساختار پیشنهادی توسط الگوریتم مبتنی بر eDLA نسبت به سایر ساختارها پیچیده‌تر است. مشابه چنین آزمایشی روی شبکه ICU-ALARM با ۱۰۰۰۰ نمونه

نتایج مقایسه برای این آزمایش را در جدول ۲ مشاهده می‌کنید. در این جدول شاخص Time زمان اجرای الگوریتم بر حسب ثانیه و شاخص Score_{diff} اختلاف امتیاز به دست آمده برای ساختار پیشنهادی الگوریتم از امتیاز K2 به دست آمده توسط شبکه Optimal را نشان می‌دهد. امتیاز شبکه اصلی 1.0×10^{-12} بوده است و تعداد کمان‌های با طرفین و جهت درست در ساختار به دست آمده توسط هر الگوریتم، n_{error} تعداد کمان‌های با جهت نادرست و $n_{correct}$ تعداد کمان‌های با جهت و طرفین نادرست را نشان می‌دهند.

بر اساس نتایج این جدول مشاهده می‌شود که الگوریتم پیشنهادی که بر اساس ترکیبی از جست و جو در فضای جایگزشت‌ها و جست و جوی در فضای DAG‌ها به کمک شاخص K2 است نسبت به روش‌های مبتنی بر جست و جو در فضای ساختار هم از نظر زمان اجرا و هم از نظر درستی ساختار پیشنهادی عملکرد خوبی دارد. در آزمایشی مشابه عملکرد الگوریتم پیشنهادی در برابر الگوریتم‌های

جدول ۴: نتایج شاخص‌های مقایسه‌ای برای شبکه ICU-ALARM

شاخص	ICU-ALARM			
	HC	GS	GS _Y	eDLA
Time	۴۰۵۰۰	۴۱۰۰۰	۴۰۰۰۰	۱۰۵۰۰
Score _{diff}	-۱۱۱۰	-۱۱۱۰	-۱۱۱۰	-۷۴۰
n _{correct}	۳۲	۳۲	۳۲	۳۳
n _{direct}	۱۲	۱۲	۱۲	۱۱
n _{error}	۷	۷	۷	۲۲

۷- نتیجه‌گیری

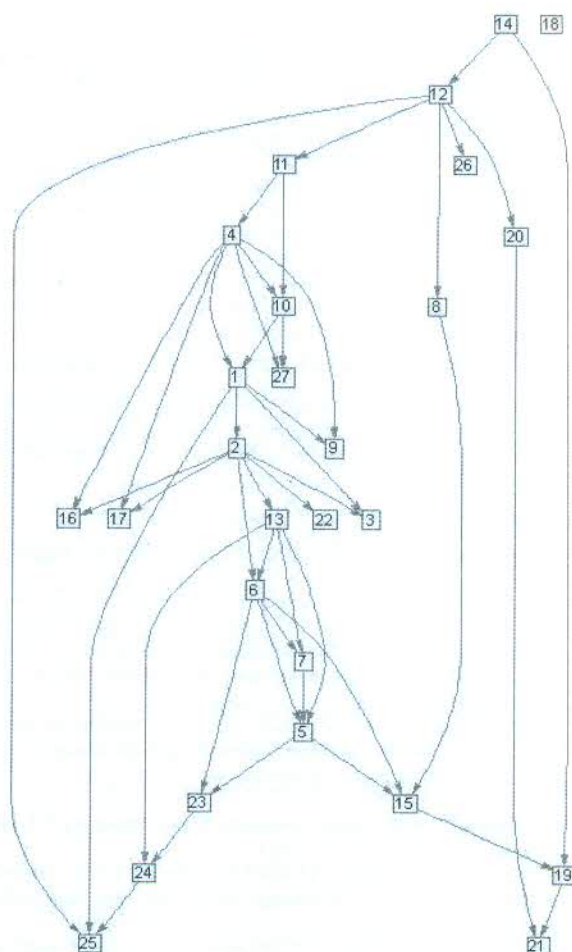
در این مقاله یک الگوریتم جدید مبتنی بر امتیاز برای یادگیری شبکه‌های بیزی مورد بررسی قرار گرفت. نوآوری این الگوریتم در استفاده از اجتماع از آتاماتا‌های یادگیر به عنوان واحدهای هوشمند محاسباتی ساده برای حل این مسأله است. این اجتماع از آتاماتاها با استفاده از بردارهای احتمال انتخاب اقدام‌های خود، الگوریتم را به سمت یافتن جایگشت مناسبی که بتواند بر اساس داده‌ها یک شبکه بیزی با بیشترین امتیاز معیار K₂ را کسب کند، هدایت می‌کردند. از مزایای این روش، سادگی روش پیشنهادی به دلیل سادگی عملکرد آتاماتا‌های یادگیر است. بررسی‌های تجربی روی شبکه‌های نمونه نشان‌دهنده عملکرد مناسب الگوریتم پیشنهادی در مقایسه با روش‌های جست و جوی دیگری است که در این مقاله مورد بررسی قرار گرفته‌اند.

در تحقیقات آتی، هدف ما استفاده از آتاماتاها برای یادگیری ساختار شبکه بیزی است. برای این هدف به دنبال یافتن هیوریستیک‌هایی به منظور بهبود الگوریتم پیشنهادی فعلی و نیز استفاده از آتاماتاها برای جست و جوی فضای ساختارها هستیم. علاوه بر این جست و جوی PDAG و استفاده از امتیازهای هم‌ارز نیز بخشی از تحقیقات آتی ما را در این راستا تشکیل می‌دهد.

مراجع

- [1] L. de Campos and J. Puerta, "Stochastic local algorithms for learning belief networks: searching in the space of the orderings," in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, vol. 2143, pp. 228-239, 2001.
- [2] D. Heckerman, A Tutorial on Learning with Bayesian Networks, in *Innovations in Bayesian Networks*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 33-82, 1996.
- [3] T. Nielsen and F. Jensen, *Bayesian Networks and Decision Graphs*, 2nd Ed. Springer, 2009.
- [4] D. M. Chickering, "Learning Bayesian networks is NP-complete," in *Learning from Data*, Springer, pp. 121-130, 1996.
- [5] W. Buntine, "A guide to the literature on learning probabilistic networks from data," *IEEE Trans. Knowl. Data Eng.*, vol. 8, no. 2, pp. 195-210, Apr. 1996.
- [6] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Mach. Learn.*, vol. 9, no. 4, pp. 309-347, 1992.
- [7] D. Heckerman and D. M. Chickering, "Learning Bayesian networks: the combination of knowledge and statistical data metrics for belief networks," *Mach. Learn.*, vol. 20, no. 3, pp. 197-243, 1995.
- [8] W. Lam and F. Bacchus, "Learning Bayesian belief networks: an approach based on the MDL principle," *Comput. Intell.*, vol. 10, no. 3, pp. 269-293, Aug. 1994.
- [9] L. De Campos and J. Fernandez-Luna, "Ant colony optimization for learning Bayesian networks," *Int. J. Approx. Reason.*, vol. 31, no. 3, pp. 291-311, 2002.
- [10] D. Chickering, D. Geiger, and D. Heckerman, "Learning Bayesian networks: search methods and experimental results," in *Proc. 5th Conf. Artif. Intell. Stat.*, pp. 112-128, 1995.

[۱۱] ف. مریخیات، الگوریتم‌های بهینه‌سازی الهام‌گرفته از طبیعت، نص، ۱۳۹۱.



شکل ۱۷: ساختار پیشنهادشده توسط الگوریتم جست و جوی حریصانه gs برای شبکه بیزی Insurance.

انجام شده است. از روی شبکه واقعی یک نمونه ۱۰۰۰۰ تایی گرفته شده است. سپس بر اساس ترتیب واقعی متغیرها ساختار شبکه Optimal با استفاده از معیار بیزی K₂ محاسبه شده است. شبکه واقعی دارای ۴۶ یال بوده و شبکه Optimal دارای ۴۷ یال و امتیاز $1.0^5 \times 10.96 - 1$ است. سپس بر اساس داده‌های نمونه‌گیری شده از هر یک از الگوریتم‌ها خواسته شده تا کار یادگیری ساختار شبکه بیزی را انجام دهند. نتایج مقایسه‌ای را در جدول ۴ مشاهده می‌کنید.

نکته بارزی که در این شبکه نیز به چشم می‌خورد آن است که الگوریتم پیشنهادی مبتنی بر eDLA به دلیل جست و جوی فضای جایگشت‌ها امتیاز بهتر و زمان کمتری دارد. اما به نظر می‌رسد ساختار به دست آمده از روش مبتنی بر eDLA در مقایسه با دیگر روش‌ها دارای ساختار پیچیده‌تری بوده و از ساختار اصلی فاصله بیشتری دارد.

در این آزمایش علاوه بر الگوریتم‌های مزبور از الگوریتم ges نیز برای جست و جوی فضای PDAG استفاده شد. نتایج نشان داد که این الگوریتم قادر است با نمونه ۱۰۰۰۰ تایی مزبور کار یادگیری ساختاری شبکه بیزی را در زمانی برابر با ۲۴۲۰۰ ثانیه انجام دهد که نسبت به روش‌های جست و جوی حریصانه در فضای ساختارها حدود ۵۰٪ کاهش نشان می‌دهد اما در مقایسه با زمان اجرای الگوریتم پیشنهادی مبتنی بر eDLA بیش از ۲ برابر زمان اجرای الگوریتم پیشنهادی این مقاله است. اما ساختار پیشنهادی روش ges نسبت به سایر روش‌ها هم امتیاز بیشتری را کسب می‌کند $1.0^5 \times 1.085$ و هم ساختار حاصل بسیار به ساختار شبکه اصلی نزدیک است.

و مهندسی کامپیوتر ایران، ب- مهندسی کامپیوتر، دوره ۱۲، شماره ۲، صص. ۱۲۶-۱۱۹، زمستان ۱۳۹۳.

- [36] M. R. Mollakhalili Meybodi and M. R. Meybodi, *Extended Distributed Learning Automata: A New Method for Solving Stochastic Graph Optimization Problems*, arXiv Prepr. arXiv:1308.2772, Aug. 2013.
- [37] M. R. Mollakhalili Meybodi and M. R. Meybodi, "Extended distributed learning automata: an automata-based framework for solving stochastic graph optimization problems," *Appl. Intell.*, vol. 41, no. 2, pp. 923-940, 2014.
- [38] M. R. Mollakhalili Meybodi and M. R. Meybodi, "Solving stochastic permutation optimization using monte carlo sampling: a learning automata based framework," *Submitt. to Appl. Soft Comput.*, 2013.
- [۳۹] م. ملاخلیلی میبدی و م. میبدی، حل مسأله درخت پوشای کمینه تصادفی از طریق بهینه‌سازی جایگشت: یک رهیافت مبتنی بر اتوماتای یادگیر توزیع‌شده، ۱۳۹۵.
- [40] A. Rezvanian and M. R. Meybodi, "A new learning automata-based sampling algorithm for social networks," *Int. J. Commun. Syst.*, p. n/a-n/a, 2015.
- [۴۱] م. ملاخلیلی میبدی و م. میبدی، "یک چارچوب مبتنی بر اتوماتای یادگیر توزیع‌شده توسعه‌یافته برای حل مسأله یافتن زیرگراف بهینه تصادفی،" *نشریه مهندسی برق و مهندسی کامپیوتر ایران، ب- مهندسی کامپیوتر، دوره ۱۲، شماره ۲، صص. ۹۷-۸۵، زمستان ۱۳۹۳.*
- [42] M. Thathachar and K. Ramakrishnan, "A hierarchical system of learning automata," *IEEE Trans. Syst. Man. Cybern.*, vol. 11, no. 3, pp. 236-241, 1981.
- [43] H. Beigy and M. R. Meybodi, "Utilizing distributed learning automata to solve stochastic shortest path problems," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 14, no. 5, pp. 591-615, Oct. 2006.
- [44] F. Norman, "On the linear model with two absorbing," *J. Math. Psychol.*, vol. 5, pp. 225-241, 1968.
- [45] S. Lakshmivarahan and M. Thathachar, "Bounds on the convergence probabilities of learning automata," *IEEE Trans. Syst. Man, Cybern.-Part A Syst. Humans*, vol. 6, no. 11, pp. 756-763, 1976.
- [46] V. Lepar and P. P. Shenoy, "A comparison of lauritzen-spiegelhalter, hugin, and shenoy-shafer architectures for computing marginals of probability distributions," in *Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence, UAI'98*, pp. 328-337, 1998.
- [47] R. M. C. G. F. C. Ingo A. Beinlich Henri Jacques Suermondt, "The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks," 1989.

محمدرضا ملاخلیلی میبدی تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد و دکتری کامپیوتر به ترتیب در سال‌های ۱۳۸۰ و ۱۳۸۲ و ۱۳۹۳ در دانشگاه‌های شهید بهشتی تهران، صنعتی امیرکبیر و دانشگاه آزاد اسلامی واحد علوم و تحقیقات به پایان رسانده است و هم‌اکنون استادیار گروه مهندسی کامپیوتر دانشگاه آزاد اسلامی واحد میبد است. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: شبکه‌های کامپیوتری و نرم‌افزاری، محاسبات نرم و کاربردهای آن.

محمدرضا میبدی تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد اقتصاد به ترتیب در سال‌های ۱۳۵۲ و ۱۳۵۶ از دانشگاه شهید بهشتی و در مقاطع کارشناسی ارشد و دکتری علوم کامپیوتر به ترتیب در سال‌های ۱۳۵۹ و ۱۳۶۲ از دانشگاه اوکلاه‌مای آمریکا به پایان رسانده است و هم‌اکنون استاد دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر می‌باشد. نام‌برده قبل از پیوستنش به دانشگاه صنعتی امیرکبیر در سال‌های ۱۳۶۲ الی ۱۳۶۴ استادیار دانشگاه میشیگان غربی و در سال‌های ۱۳۶۴ الی ۱۳۷۰ دانشیار دانشگاه اوهایو در ایالات متحده آمریکا بوده است. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: الگوریتم‌های موازی، پردازش موازی، محاسبات نرم و کاربردهای آن، شبکه‌های کامپیوتری و مهندسی نرم‌افزار.

- [12] K. Shibata, H. Nakano, and A. Miyauchi, "A learning method for dynamic Bayesian network structures using a multi-objective particle swarm optimizer," *Artif. Life Robot.*, vol. 16, no. 3, pp. 329-332, Dec. 2011.
- [13] K. Salama and A. Freitas, "ABC-miner: an ant-based Bayesian classification algorithm," *Swarm Intell.*, 2012.
- [14] T. Wang and J. Yang, "A heuristic method for learning Bayesian networks using discrete particle swarm optimization," *Knowl. Inf. Syst.*, vol. 24, no. 2, pp. 269-281, Aug. 2009.
- [15] P. C. Pinto, A. Nagele, M. DeJori, T. A. Runkler, and J. M. C. Sousa, "Using a local discovery ant algorithm for Bayesian network structure learning," *IEEE Trans. Evol. Comput.*, vol. 13, no. 4, pp. 767-779, Aug. 2009.
- [16] R. Daly, Q. Shen, and S. Aitken, "Using ant colony optimization in learning Bayesian network equivalence classes," in *Proc. UKCI*, pp. 111-118, 2006.
- [17] T. Brouard, A. Delaplace, and H. Cardot, *Evolutionary Methods for Learning Bayesian Network Structures*, 2008.
- [18] J. Lee, W. Chung, E. Kim, and S. Kim, "A new genetic approach for structure learning of Bayesian networks: matrix genetic algorithm," *Int. J. Control. Autom. Syst.*, vol. 8, no. 2, pp. 398-407, Apr. 2010.
- [19] L. de Campos and J. Huete, "On the use of independence relationships for learning simplified belief networks," *Int. J. Intell. Syst.*, vol. 12, no. 7, pp. 495-522, 1997.
- [20] S. Acid and L. M. de Campos, "A hybrid methodology for learning belief networks: BENEDICT," *Int. J. Approx. Reason.*, vol. 27, no. 3, pp. 235-262, Sep. 2001.
- [21] R. Blanco, I. Inza, and P. Larranaga, "Learning Bayesian networks in the space of structures by estimation of distribution algorithms," *Int. J. Intell. Syst.*, vol. 18, no. 2, pp. 205-220, Feb. 2003.
- [22] M. Teyssier and D. Koller, "Ordering-Based Search: A Simple and Effective Algorithm for Learning Bayesian Networks," arXiv:1207.1429, 2012.
- [23] W. Hsu, H. Guo, B. Perry, and J. Stilson, "A Permutation Genetic Algorithm for Variable Ordering in Learning Bayesian Networks from Data," *GECCO*, 2002.
- [24] P. Larranaga, C. M. H. Kuijpers, R. H. Murga, and Y. Yurramendi, "Learning Bayesian network structures by searching for the best ordering with genetic algorithms," *IEEE Trans. Syst. Man, Cybern.-Part A Syst. Humans*, vol. 26, no. 4, pp. 487-493, Jul. 1996.
- [25] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *J. Comput. Biol.*, vol. 7, no. 3-4, pp. 601-620, Jan. 2000.
- [26] T. Silander and P. Myllymaki, *A Simple Approach for Finding the Globally Optimal Bayesian Network Structure*, arXiv Prepr. arXiv:1206.6875, Jun. 2012.
- [27] O. Francois, *BNT Structure Learning Package: Documentation and Experiments*, 2004.
- [28] D. Margaritis, *Learning Bayesian Network Model Structure from Data*, Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, 2003.
- [29] K. P. Murphy, *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning Series)*, the MIT Press, 2012.
- [30] P. Munteanu and M. Bendou, "The EQ framework for learning equivalence classes of Bayesian networks," in *Proc. 2001 IEEE Int. Conf. on Data Mining, ICDM'01*, pp. 417-424, San Jose, CA, USA, 2001.
- [31] D. Chickering, "Learning equivalence classes of Bayesian-network structures," *J. Mach. Learn. Res.*, vol. 2, no. 3, pp. 445-498, 2002.
- [32] R. Castelo and T. Kocka, "On inclusion-driven learning of Bayesian networks," *J. Mach. Learn. Res.*, vol. 4, pp. 527-574, 2003.
- [33] D. M. Chickering, "Optimal structure identification with greedy search," *J. Mach. Learn. Res.*, vol. 3, pp. 507-554, 2003.
- [34] M. A. Thathachar and B. Harita, "Learning automata with changing number of actions," *IEEE Trans. Syst. Man, Cybern.-Part A Syst. Humans*, vol. 17, no. 6, pp. 1095-1100, Nov./Dec. 1987.
- [۳۵] م. ملاخلیلی میبدی و م. میبدی، "یک الگوریتم جدید مبتنی بر اتوماتای یادگیر توزیع‌شده توسعه‌یافته برای یادگیری پارامتری شبکه بی‌زی،" *نشریه مهندسی برق*



نشریه

مهندسی برق و

مهندسی کامپیوتر

ایران

نشریه‌ای علمی خاص مباحث نظری و کاربردی در عرصه مهندسی برق و کامپیوتر

نشریه علمی - پژوهشی

ب - مهندسی کامپیوتر

<http://www.ijece.org/farsi>

شماره ۱

سال ۱۴

بهار ۱۳۹۵

- | | | |
|----|--|--|
| ۱ | جابه‌جایی اجسام توسط یک سیستم چندرباتی توزیع‌شده با استفاده از یک معماری ترکیبی | طاہر حکمت‌فر و الیس مسیحی |
| ۱۶ | تخمین ضرایب علیت در نقشه راهبردی سازمان به کمک آموزش نقشه شناختی فازی با الگوریتم جستجوی گرانشی | علی جهان‌بیگی، منصور شیخان و محسن روحانی |
| ۲۷ | یادگیری ساختاری شبکه‌های بیزی یک رهیافت مبتنی بر آتاماتاهای یادگیر | محمدرضا ملاخلیلی میبدی و محمدرضا میبدی |
| ۴۱ | آشکارسازی چندتایی صورت انسان در تصاویر اینترنتی با استفاده از ترکیب فیلترهای گابور و شبکه عصبی | رضا محمدیان و محمود محلوچی |
| ۵۳ | ادغام شبکه‌های عصبی بر اساس یادگیری با همبستگی منفی در بازشناسی برون خط کلمات دست‌نویس | سیدعلی اصغر عباس‌زاده آرانی و احسان‌اله کبیر |
| ۶۳ | استفاده از یک روش خوشه‌بندی و محاسبه شهرت منفی هر میزبان به منظور تشخیص بات‌نت‌ها با استفاده از ترافیک DNS | رضا شریف‌نیا، دیزینی و آناهیتا منافی مورکانی |
| ۷۳ | طبقه‌بندی تصاویر ابرطیفی با استفاده از تحلیل ممیز خطی فضای خوشه و مجموعه نمونه‌های آموزشی کوچک | مریم ایمانی و حسن قاسمیان |
| ۸۲ | آشکارسازی آغاز در تکنوازی تار بر پایه انرژی و فرکانس پایه | بهرآز فرخی و احسان‌اله کبیر |

۸۹

تقدیر و تشکر

NASHRIYYAH -I

MUHANDISI -I BARQ VA MUHANDISI -I KAMPYUTAR -I

A Journal Devoted to the Theoretical and Experimental Aspects of Computer Engineering

IRAN

B: MUHANDISI -I KAMPYUTAR

SPRING 2016

VOLUME 14

NUMBER 1

NMBMKI

(ISSN 1682-3745)

<http://www.ijece.org>

ABSTRACTS