

یادگیری ساختاری شبکه‌های بیزین

یک رهیافت مبتنی بر آتاماتاهای یادگیر

محمد رضا میبیدی

آزمایشگاه محاسبات نرم، دانشگاه صنعتی امیرکبیر

mmezbodi@aut.ac.ir

محمد رضا ملاخلیلی میبیدی

دانشکده فنی و مهندسی، دانشگاه آزاد اسلامی، واحد میبد

mollakhalili@maybodiau.ac.ir

چکیده - یکی از مسائل جالب در هوش مصنوعی ساخت شبکه بیزی بر اساس نمونه‌هایی از داده‌ها است؛ یعنی فرض کنید یک شبکه بیزین 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 جست‌وجو در میان ساختارهای ممکن است. جست‌وجو برای یافتن یک ساختار مناسب برای شبکه بیزی که بیشترین سازگاری را با نمونه‌ها داشته باشد غالباً از طریق جست‌وجو در فضای ساختارها با استفاده از تکنیک‌های جست‌وجوی استاندارد یا الهام گرفته از طبیعت نظیر تپه‌نوردی حریرصانه، 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 جست‌وجوی تصادفی مورد استفاده قرار می‌گیرد. از ویژگی‌های روش جدید پیشنهادی جست‌وجوی 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. کلید واژه‌ها - یادگیری ساختار شبکه بیزی، آتاماتای یادگیر.

۱ مقدمه

عبارت ۱ نشان‌دهنده تجزیه توزیع توأم است. روابط وابستگی/استقلالی که این تجزیه را ممکن می‌کند 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.

مسأله یادگیری یک شبکه بیزی را می‌توان به این صورت توصیف کرد: یک مجموعه آموزشی $\{V^1, V^2, \dots, V^m\}$ از D از نمونه‌هایی از V داده شده است. شبکه بیزین BN را به‌گونه‌ای بیابید که بهترین تطابق با داده‌های D را داشته باشد. این مسأله در رده مسائل NP-Hard رده‌بندی می‌شود [۱۲] و لذا بایستی از روش‌های مکاشفه‌ای^۳ آن را حل کرد.

رهیافت معمول برای حل این مسأله 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، جست‌وجو برای یافتن بهترین شبکه صورت می‌گیرد. معیارهای امتیازدهی متعددی اعم از بیزی و غیربیزی در متون پیشنهاد شده و قابل استفاده است [۹] [۱۶] [۲۳] [۲۷].

مهم‌ترین ویژگی موردانتظار از یک شاخص f تجزیه‌پذیری^۴ آن در

یک شبکه بیزین - 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 میان متغیرهای تصادفی تشکیل‌دهنده دامنه مسأله‌ای که مدل کردن آن مدنظر است را فراهم می‌کند [۱۹]. این نمایش از دو مؤلفه تشکیل شده است: ساختار و پارامترها.

ساختار شبکه بیزی یک گراف جهت‌دار و فاقد دور $G = (V, E)$ است که در آن $V = \{x_1, x_2, \dots, x_n\}$ مجموعه گره‌هایی است که متغیرهای سیستم را نشان می‌دهند و $E \subset (V \times V)$ یک مجموعه از یال‌ها و نشان‌دهنده ارتباط وابستگی مستقیم بین این متغیرها است [۲۲]. برای هر متغیر $x_i \in V$ 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 جمعی روی V را می‌توان از طریق این توزیع‌های شرطی به دست آورد [۳۷]:

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

^۱Heuristic

^۴Decomposability

^۱Belief Network

^۲Causal Network

الگوریتم‌های متعددی که برای حل مسأله یادگیری ساختاری شبکه بیزی پیشنهاد شده است را می‌توان از منظرهای گوناگون دسته‌بندی کرد. از نظر زمانی بخش عمده‌ای از تلاش‌های صورت گرفته برای حل این مسأله به دو دهه پیش بر می‌گردد که با دیدگاه آماری به حل مسأله نگاه شده است [۲۷][۱۶][۱۵]. با پیدایش روش‌های بهینه‌سازی الگوگرفته از طبیعت [۱] مجدداً تلاش‌هایی برای حل این مسأله صورت گرفته است [۴۶][۴۱][۴۰][۳۹][۳۰][۸][۱۷].

نگاه دیگری که به روش‌های پیشنهادی برای حل مسأله یادگیری ساختاری می‌توان داشت، به نحوه بررسی تناسب و برازندگی ساختارهای گوناگون برمی‌گردد [۱۹]. در گروهی از این روش‌ها، از آزمون‌های استقلال شرطی [۱۸] و در گروهی دیگر از روش‌های مبتنی بر امتیازدهی [۱۶][۲۷][۲۳] برای تعیین میزان برازندگی یک ساختار با داده‌های نمونه استفاده شده است. در برخی نیز، ترکیبی از هر دو [۵] مورد استفاده قرار گرفته است.

از منطری 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 جست‌وجو می‌پردازند [۱۰][۳۰][۷] و گروهی دیگر که در میان ترتیب‌های مختلف متغیرها به جست‌وجوی بهترین ساختار می‌پردازند [۴۳][۲۴][۲۹]. گروه دوم الگوریتم‌ها در مقایسه با گروه اول فضای جست‌وجوی محدود شده‌تری دارند. 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 شناخته شده نباشد می‌توانیم به جای جست‌وجو روی DAG‌های مختلف، روی مرتب‌سازی‌های مختلف جست‌وجو را انجام دهیم. برخی از نویسندگان نشان داده‌اند که جست‌وجو در فضای جایگشت‌های متغیرها از جست‌وجو در فضای ساختارها مناسب‌تر و الگوریتم به دست‌آمده کارآمدتر و دقیق‌تر است [۲۹][۲۱].

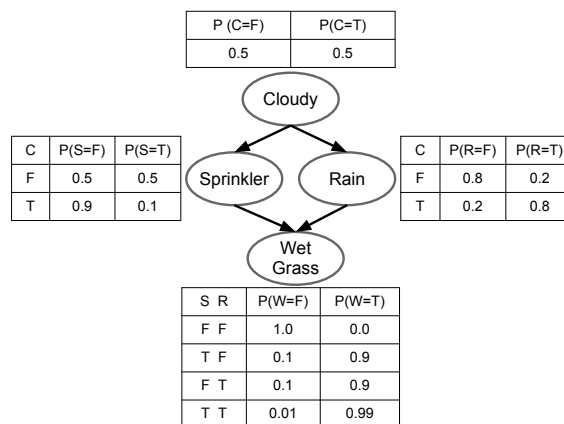
۱.۲ جست‌وجو در فضای جایگشت‌ها

یک شبکه بیزین $G = (V, E)$ را می‌توان با بردارهای $G = (G_1, G_2, \dots, G_n)$ از مجموعه والد‌های هر یک از رؤس متناظر با مجموعه متغیرهای $V = \{x_1, x_2, \dots, x_n\}$ توصیف کرد. به عبارت دیگر G_i زیرمجموعه‌ای از 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 $4 \rightarrow 1, 1 \rightarrow 3, 3 \rightarrow 1$ است. 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 بگیرد. i امین عنصر در یک ترتیب را با Ord_i نمایش می‌دهیم. مثلاً در ترتیب مشخص شده $Ord_2 = 1$ است.

تعریف ۱ (سازگاری شبکه بیزین با یک ترتیب). یک شبکه بیزین $G = (G_1, G_2, \dots, G_n)$ را با ترتیب Ord سازگار گویند، هرگاه

$$\forall i, G_i \subseteq \bigcup_{j=1}^{i-1} Ord_j \quad (3)$$



شکل ۱: نمونه‌ای از یک شبکه بیزی

شرایط وجود داده‌های کامل است؛ یعنی تابع امتیازدهی می‌تواند به صورت زیر تجزیه شود:

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

که در آن $N_{x_i, Pa(x_i)}$ 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 نقطه‌نظر الگوریتم‌های جست‌وجو و یادگیری فوق‌العاده با اهمیت است. یک پروسه جست‌وجوی 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. در بخش ۴ روش جدید پیشنهادی تشریح شده است. 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 بررسی کارهای صورت گرفته

ثابت شده است که $r(n)$ ، تعداد ساختارهای مختلف برای یک شبکه بیزین با n گره از طریق رابطه بازگشتی زیر قابل محاسبه است [۱۰]

$$r(n) = \sum_{k=1}^n (-1)^{k+1} \frac{n!}{(n-k)!k!} 2^{k(n-k)} r(n-k) = n^{2^{O(n)}}$$

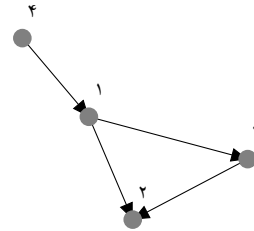
بدیهی است جست‌وجو برای یافتن جواب در چنین فضای جواب بزرگی ناممکن است. بنابراین بایستی از روش‌های مکاشفه‌ای برای جست‌وجو و یافتن یک بهترین جواب یا نزدیک‌ترین جواب بهره گرفت.

الگوریتم ۱ الگوریتم جست و جوی تپه‌نوردی Hill Climbing

ورودی: مجموعه داده‌های D
 خروجی: DAG متناظر با ساختار شبکه بیزی B

```

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



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

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. 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 [۴۲] یک رهیافت مبتنی بر جست و جوی بهترین ترتیب برای شبکه‌های بیزی با تعداد متغیرهای کم پیشنهاد شده است. این روش جست و جوی را در میان تمام ترتیب‌های ممکن و تمام مجموعه والد‌های ممکن برای یک متغیر انجام می‌دهد.

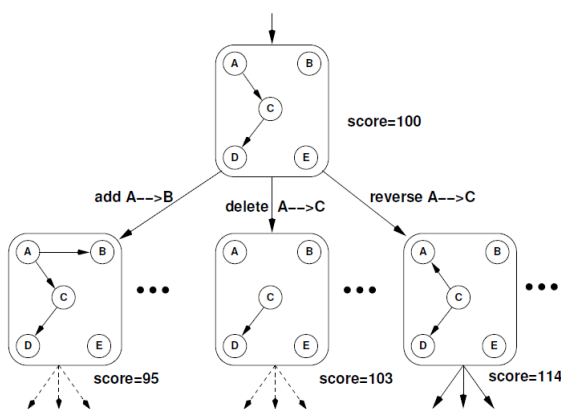
۲.۲ جست و جوی فضای ساختارها

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

جست و جوی حریصانه

جست و جوی حریصانه مشهورترین هیوریستیک بهینه‌سازی است. این روش با یک گراف اولیه شروع می‌کند. بر اساس یک همسایگی تعریف شده، 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 بیزین مجموعه همسایگی رامی‌توان مجموعه گراف‌هایی تعریف کرد که با گراف فعلی در کم یا زیاد شدن یک یال یا تغییر جهت یک یال تفاوت دارند (شکل ۳). ما در بخش شبیه‌سازی‌ها از نسخه‌هایی از این الگوریتم به نام‌های GS، HC (Hill Climbing) و GS۲ استفاده خواهیم کرد [۲۰].



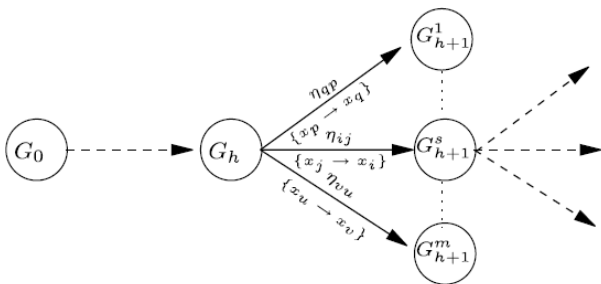
شکل ۳: فرآیند جست و جوی مورد استفاده توسط الگوریتم جست و جوی تپه‌نوردی [۲۲، صفحه ۱۹، شکل ۳]

الگوریتم B

G_h است که در آن مورچه فاز ساخت را متوقف می‌کند. هیوریستیک انتخاب شده در راستای افزودن یالی است که بیشترین افزایش را در معیار f ایجاد می‌کند. بنابراین

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

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 $x_i \rightarrow x_j$ و G^+ بهترین گراف یافته شده باشد مقدار فرمون بر اساس رابطه $(1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}$ به روزرسانی می‌شود و در آن $\Delta\tau_{ij} = \tau_{ij}$ است اگر $x_j \rightarrow x_i \notin G^+$ و $\Delta\tau_{ij} = 0$ است اگر $x_j \rightarrow x_i \in G^+$ باشد. کمان بعدی که



شکل ۴: AntB نسخه تصادفی الگوریتم B بر اساس بهینه‌سازی کلونی مورچه [۱۰]

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 $[\tau_{ij}][\eta_{ij}]^\beta$ انتخاب شده و به ساختار شبکه اضافه می‌شود. در [۲۸] از الگوریتم‌های ژنتیک به عنوان ابزاری برای تولید جایگشت‌ها و جست‌وجو در ساختارهای شبکه استفاده شده است.

الگوریتم K۲

ایده اصلی الگوریتم K۲ بیشینه‌کردن احتمال ساختار بر اساس داده‌های موجود است. الگوریتم K۲ 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 [۱۶] ارائه شده است. این معیار با کمک رابطه زیر تعریف می‌شود.

$$P(G, D) = P(G) \prod_{i=1}^n \prod_{j=a}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \quad (۴)$$

در این رابطه 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 j 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

الگوریتم B یک روش ساخت شبکه بیزین به شیوه مکاشفه‌ای حریصانه است [۱۰]. 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 جست‌وجو 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 بروزرسانی $O(n^2)$ است. شبه‌کد این روش در الگوریتم ۲ نشان داده شده است.

الگوریتم ۲ الگوریتم B

ورودی: مجموعه داده‌های D و n : تعداد متغیرها
خروجی: ماتریس مجاورت DAG متناظر با شبکه بیزی در قالب ماتریس مجاورت $A_{n \times n}$

repeat

Select two indexes (i, j) such that $(i, j) = \arg \max_{i_1, j_1} A[i_1, j_1]$;

if $A[i, j] > 0$ then

$Pa(x_i) = Pa(x_i) \cup \{x_j\}$;

end if

$A[i, j] = -\infty$;

for all $x_a \in \text{Ancestors}(x_j) \cup \{x_j\}$ and $x_b \in \text{Descendants}(x_i) \cup \{x_i\}$ do

$A[a, b] = -\infty$;

end for

for $k \leftarrow 1 : n$ do

if then $A[i, j] > -\infty$

$A[i, k] = f(x_i, Pa(x_i) \cup \{x_k\}) - f(x_i, Pa(x_i))$;

end if

end for

until $\forall i, j (A[i, j] \leq 0 \text{ or } A[i, j] = -\infty)$;

در [۱۰] ترکیبی از الگوریتم B و روش بهینه‌سازی کلونی مورچه‌ها برای انجام جست‌وجو در فضای ساختارهای شبکه بیزی ارائه شده است. در الگوریتم پیشنهادی مسأله به شکل DAG‌هایی با n گره مدل شده است. حالت G_h یک DAG با گره‌های $x_i \in \mathbf{V}$ و دقیقاً h یال است. مورچه به شکل تزییدی جواب را با شروع از یک گراف خالی G می‌سازد. این کار با افزودن یال $x_j \rightarrow x_i$ به حالت فعلی G_h به دست می‌آید یعنی $G_{h+1} = G_h \cup \{x_j \rightarrow x_i\}$. جواب نهایی حالت

داریم: $\log(P(G, D))$ به جای $P(G, D)$

$$f_{K2}(G, D) = \sum_{i=1}^n f_{K2}(x_i, Pa(x_i); N_{x_i, Pa(x_i)}) \quad (5)$$

که در آن:

$$f_{K2}(x_i, Pa(x_i); N_{x_i, Pa(x_i)}) = \sum_{j=1}^{q_i} \left(\log \left(\frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \right) + \sum_{k=1}^{r_i} \log(N_{ijk!}) \right) \quad (6)$$

شبهکد $K2$ را در الگوریتم ۳ مشاهده می‌کنید.

الگوریتم ۳ الگوریتم $K2$

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  $i \leftarrow 1 : n$  do
   $\pi_i \leftarrow \phi$ ; // نشان می‌دهد
   $P_{old} = f_{K2}(x_i, \pi_i)$ ; // مطابق با رابطه ۶
  OKToProceed  $\leftarrow$  true;
  while OKToProceed and  $|\pi_i| < u$  do
     $z = \arg \max_{z \in Pred(x_i) - \pi_i} f_{K2}(x_i, \pi_i \cup \{z\})$ 
     $P_{new} = f_{K2}(x_i, \pi_i \cup \{z\})$ ;
    if  $P_{new} > P_{old}$  then
       $P_{old} \leftarrow P_{new}$ ;
       $\pi_i \leftarrow \pi_i \cup \{z\}$ ;
    else
      OKToProceed  $\leftarrow$  false;
    end if
  end while
  Parents( $x_i$ )  $\leftarrow$   $\pi_i$ ;
end for

```

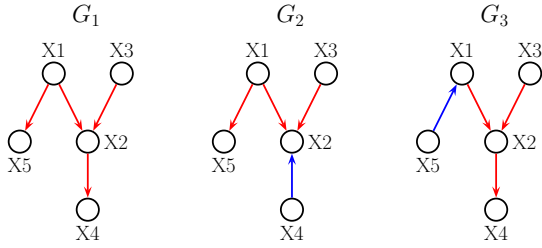
جست‌وجوی حریصانه در فضای هم‌ارز مارکوف

کارهای جدیدتری که در حوزه یادگیری ساختار شبکه بیزی از طریق جست‌وجو در فضای ساختارها صورت گرفته است، علاقه‌مندی به جست‌وجو در فضای هم‌ارز مارکوف را نشان می‌دهند.

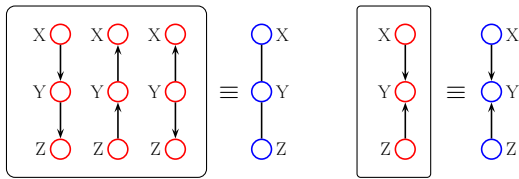
تعریف ۲. دو DAG را هم‌ارز گوئیم اگر مجموعه وابستگی/استقلال شرطی یکسانی داشته باشند. به عنوان مثال اگر X به شرط Y از Z مستقل باشد $(X \perp Z | Y)$ در این صورت

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

در شکل ۵ سه نمونه DAG مشاهده می‌کنید که دوتای آنها با یکدیگر هم‌ارز هستند.



شکل ۵: DAGهای $G1$ و $G3$ هم‌ارز مارکوفی هستند اما با $G2$ هم‌ارز نیستند [۲۶، ۵، شکل ۹۱۵]



شکل ۶: نمایش DAGهای هم‌ارز مارکوف به شکل PDAG [۲۶، ۶، صفحه ۹۱۶، شکل ۲۶، ۶]

در شکل ۶ مشاهده می‌کنید که می‌توان با حذف جهت برخی یال‌ها DAG را به صورتی بدون ابهام به شکل غیرجهت‌دار نشان داد.

تعریف ۳. یک کمان را معکوس پذیر گوئیم اگر معکوس کردن آن منجر به ایجاد گرافی هم‌ارز 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 یال‌های فاقد جهت کمان‌های (یال‌های جهت‌دار) معکوس‌ناپذیر است

تعریف ۴. یک معیار امتیازدهی را هم‌ارز گوئیم اگر روی DAGهای هم‌ارز مقادیر یکسانی تولید کند

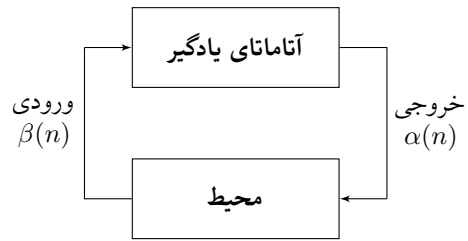
نویسندگان در [۳۵] نشان داده‌اند که احتمال همگرا شدن جست‌وجو در این فضا با امتیازهای هم‌ارز نسبت به جست‌وجو در فضای DAGها بیشتر است. این مفهوم توسط [۱۴] و [۱۱] در روش‌های جست‌وجوی ساختاری پیشنهاد شده است. نویسنده در [۱۳] یک روش جست‌وجوی هم‌ارزی حریصانه (GES) پیشنهاد داده است که از CPDAGها برای نمایش کلاس‌های هم‌ارزی 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 آتاماتای یادگیر

آتاماتای یادگیر تصادفی یک واحد تصمیم‌گیرنده تطبیقی است که فرآیند یادگیری در آن، از طریق تعاملش با محیط صورت می‌گیرد. آتاماتای یادگیر، مجموعه‌ای از اقدام‌های قابل انجام دارد. این اقدام‌ها به تصادف

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

رابطه ۸ زمانی مورد استفاده قرار می‌گیرد که عمل $\alpha_i(k)$ منجر به دریافت پاداش از محیط شده باشد و ۹ زمانی مورد استفاده قرار می‌گیرد که این عمل به دریافت جریمه از محیط منجر شده باشد. در روابط ۸ و ۹ اگر $a=b$ باشد روابط یادگیری خطی را الگوریتم L_{R-P} می‌نامند. اگر $a \gg b$ باشد آن را $L_{R-\varepsilon P}$ و اگر $b=0$ باشد آن را L_{R-I} می‌نامند.



شکل ۷: آتاماتای یادگیر و محیط

۱.۳ آتاماتای یادگیر با مجموعه اقدام‌های متغیر

یک آتاماتای یادگیر با مجموعه اقدام متغیر، آتاماتایی است که در آن تعداد اقدام‌های موجود هر آتاماتا در طول زمان تغییر می‌کند [۲۵]. فرض کنید مجموعه اقدام‌های آتاماتا را با $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ و مجموعه اقدام‌های قابل انجام توسط آتاماتا در زمان n را با $A(k) \subseteq \alpha$ نشان دهیم

$$A(k) \in \{A_1, A_2, \dots, A_m\} \quad m = 2^{n-1}$$

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

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

انتخاب می‌شود.
اگر تعریف کنیم

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

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

$$\hat{p}_i(k) = p_i(k)/K(k) \quad (10)$$

در رابطه ۱۰، $K(k)$ مجموع احتمالات تمام اقدام‌های فعال (اقدام‌های عضو $A(k)$) است و داریم

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

که در آن $p_i(k) = Prob[\alpha(k) = \alpha_i]$ احتمال انتخاب اقدام α_i در میان مجموعه تمام اقدام‌های آتاماتا (اعم از اینکه 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.

و بر اساس یک بردار توزیع احتمال، انتخاب شده و به عنوان ورودی به محیط اعمال می‌شوند. محیط، اقدام انجام شده را به کمک یک سیگنال تقویتی بازخوردی، مورد ارزیابی قرار می‌دهد. آتاماتای یادگیر بر اساس سیگنال بازخوردی حاصل، بردار توزیع احتمال انتخاب اقدام‌ها را به روزرسانی می‌کند. هدف آتاماتا، پیدا کردن اقدام بهینه در میان مجموعه اقدام‌های قابل انجام است؛ اقدامی که بیشترین پاداش را از محیط دریافت کند. ارتباط آتاماتای تصادفی با محیط در شکل ۷ نشان داده شده است. از این مجموعه به همراه الگوریتم یادگیری تحت عنوان آتاماتای یادگیر تصادفی نام برده می‌شود. به این ترتیب آتاماتای یادگیر تصادفی را می‌توان با چهارتایی ψ تعریف کرد.

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

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

الگوریتم یادگیری یک رابطه بازگشتی است که برای انجام تغییرات و به روزرسانی در بردار احتمال اقدام‌های آتاماتا در یک آتاماتای یادگیر تصادفی با ساختار متغیر مورد استفاده قرار می‌گیرد. فرض کنید یک آتاماتای یادگیر تصادفی ساختار متغیر در زمان k از میان مجموعه اقدام‌ها α عمل $\alpha_i(k)$ را انتخاب کرده باشد. همچنین فرض کنید بردار احتمال انتخاب اقدام‌های آتاماتا 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 آتاماتای یادگیر باشد، بردار $p(k)$ توسط الگوریتم یادگیری خطی ارائه شده در روابط زیر به روزرسانی می‌شود. مقدار a را پارامتر پاداش و b را پارامتر جریمه می‌نامند

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

ضریب نرمال کننده $K(k) = \sum_{\alpha_i \in A(k)} p_i(k)$ برای این مجموعه از اقدام‌های فعال محاسبه شده و بردار احتمال انتخاب اقدام‌های آتاماتا مطابق با رابطه بالا نرمال سازی 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 به‌روزرسانی نیز بردار $\hat{p}_i(k)$ مطابق با الگوریتم یادگیری مورد استفاده توسط آتاماتا به‌روزرسانی می‌شود و سپس با استفاده مجدد از رابطه بالا بردار $K(k)$ را $p_i(k) = \hat{p}_i(k)$ به دست می‌آید. نشان داده شده است [۴۵] که در صورتی که الگوریتم یادگیری مورد استفاده L_{R-I} باشد، این روش ویژگی‌های ε -optimality و absolute expediency را دارد.

۲.۳ آتاماتای یادگیر توزیع شده توسعه یافته

آتاماتای یادگیر توزیع شده توسعه یافته 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 [۳۳]. در آتاماتای یادگیر توزیع شده توسعه یافته، هر آتاماتا دارای یک سطح فعال بودن است و زمان و نحوه تغییر سطح فعالیت آتاماتاها به تبعیت از شرایط مسأله، تغییر می‌کند. در آتاماتای یادگیر توزیع شده توسعه یافته، 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 توزیع شده توسعه یافته را به صورت $A = \{A, E, S, P, S^*, F, C\}$ تعریف کرد که در آن $A = \{1, 2, \dots, n\}$ مجموعه‌ای از آتاماتاهای یادگیر است. $E \subseteq A \times A$ مجموعه ارتباطات سلسله مراتبی میان اعضا A است. $G = (A, E)$ گراف ارتباطی نامیده می‌شود.

$S = \{s_1, s_2, \dots, s_n\}$ مجموعه‌ای از مقادیر موسوم به سطح فعالیت است که سطح فعال بودن هر آتاماتایی متعلق به A را مشخص می‌کند. هر یک از s_i ها می‌توانند یکی از مقادیر $Active$ ، $Passive$ ، Off یا Fi باشند که آنها را به ترتیب با Ac ، Pa ، Of و Fi نمایش می‌دهیم. $Fire$ ، بالاترین سطح فعالیت و $Passive$ و off پایین ترین 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.

P : مجموعه‌ای محدود از قواعد حاکم بر سطح فعالیت آتاماتاها است. این قواعد تابعی از سطح فعالیت فعلی هر آتاماتا و سطح فعالیت آتاماتاها مجاور است و بسته به مسأله‌ای که $eDLA$ برای حل آن به کار گرفته می‌شود، می‌تواند متغیر باشد.

$S^0 = (s_1^0, s_2^0, \dots, s_n^0)$ وضعیت آغازین سیستم نامیده می‌شود. $F = \{S^F | S^F = (s_1^F, s_2^F, \dots, s_n^F)\}$ مجموعه شرایط پایانی نامیده می‌شود. مجموعه F شرایطی را برحسب سطح فعالیت آتاماتاها 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 آتاماتاها off است.

C : یک تابع خاص است که بر اساس سطح فعلی فعالیت آتاماتاها و مسأله‌ای که $eDLA$ برای حل آن طراحی شده است، آتاماتایی را از

میان مجموعه آتاماتاها موجود در $eDLA$ انتخاب می‌کند. C را تابع آتش یا Fire Function 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 انجام دادن دارد.

نحوه انجام کار $eDLA$ به این صورت است که $eDLA$ از وضعیت S^0 آغاز کرده و بر اساس مجموعه قواعد P نسبت به تغییر سطح فعال بودن آتاماتاها اقدام می‌کند. تغییر سطح فعال بودن آتاماتاها 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 این‌که، بر اساس قواعد ارتباطی، مجموعه‌ای از آتاماتاها مجاور با آتاماتای $Fire$ که در پایین ترین سطح فعال بودن ($Passive$) 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 آتاماتاها $Active$ صورت می‌گیرد و غالباً به صورت تصادفی انجام می‌شود.

از $eDLA$ تاکنون برای حل برخی مسائل استفاده شده است. نویسنده در [۳] مسأله یادگیری پارامتری شبکه‌های بیزی به کمک آتاماتای یادگیر توزیع شده توسعه یافته مورد بررسی قرار داده است. در [۳۴] روشی مبتنی بر آتاماتای یادگیر برای حل مسأله بهینه‌سازی جایگشت ارائه شده است. در [۲] از روش پیشنهادی در [۳۴] برای حل مسأله درخت پوشای کمینه تصادفی استفاده شده است.

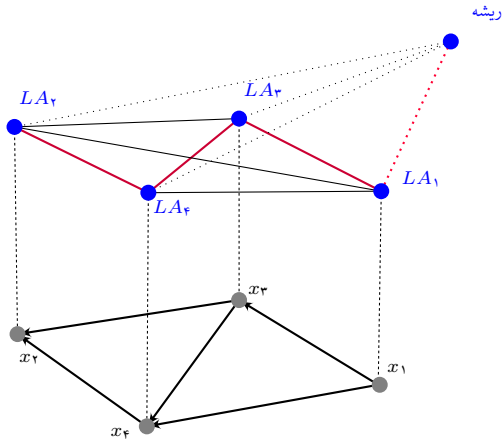
۴ روش پیشنهادی

در این بخش به بررسی روش پیشنهادی یادگیری ساختار شبکه بیزی با استفاده از آتاماتای یادگیر خواهیم پرداخت.

۱.۴ الگوریتم پیشنهادی

الگوریتم پیشنهادی که در این بخش ارائه می‌شود به جای جست‌وجو در فضای DAG ها به جست‌وجو در فضای جایگشت‌ها پرداخته و از روش امتیازدهی K_2 برای ارزیابی جایگشت ایجاد شده در هر مرحله استفاده می‌کند. همانگونه که قبلاً متذکر شدیم، فضای جایگشت‌ها نسبت به فضای DAG ها محدودتر است. علاوه بر این انتخاب معیار K_2 در این مقاله به دلیل کاهش سربار محاسباتی از طریق ذخیره کردن نتایج جزئی میانی (کردن $Cache$) و کاهش سربار محاسباتی روش پیشنهادی صورت گرفته است. جست‌وجو در فضای جایگشت‌ها با نگاشت یک جایگشت به یک مسیر هامیلتونی در یک گراف کامل صورت می‌گیرد. برای این‌کار از روش پیشنهادی در [۳۴] [۴] استفاده شده است.

برای این منظور یک گراف کامل فاقد جهت با تعداد $n+1$ راس (یک راس بیشتر از تعداد متغیرها) به عنوان گراف ارتباطی $eDLA$ ایجاد می‌شود. هر یک از رئوس گراف ایجاد شده متناظر با یکی از متغیرها و یک رأس به عنوان ریشه است. هر یال $e_{(i,j)}$ که رئوس i و j را در گراف به یکدیگر متصل می‌کند متناظر با اقدام α_i^j از آتاماتای LA_j یا اقدام α_j^i از آتاماتای LA_i است. در هر بار اجرا، $eDLA$ از آتاماتای متناظر



شکل ۸: تناظر بین eDLA و متغیرها و مثالی از اجرای الگوریتم ۴: eDLA مسیر $2 \rightarrow 3 \rightarrow 4 \rightarrow 1$ را انتخاب می‌کند که به ترتیب $Ord = \{2, 3, 4, 1\}$ نگاشت می‌شود. بر اساس این ترتیب DAG سازگار $G = \{\{1\}, \{3, 4\}, \{1\}, \{3, 1\}\}$ را به عنوان ساختار با امتیاز بیشینه تعیین می‌کند.

الگوریتم ۴ الگوریتم پیشنهادی مبتنی بر eDLA برای یادگیری ساختار شبکه بیزین

ورودی: D : مجموعه نمونه‌ها، n : تعداد متغیرها، u : حداکثر تعداد والدهای یک گره، آستانه‌های P_{TH} و K ؛

خروجی: ماتریس مجاورت DAG متناظر با شبکه بیزی در قالب بردار Parents که لیست والدهای هر گره را مشخص می‌کند.

Construct $G = (V, V \times V)$ by $V = \{0, 1, 2, \dots, n\}$
 Construct an eDLA from graph G ;
 $k \leftarrow 0, \Pi \leftarrow \phi, TH_k \leftarrow -\infty$;
repeat
 $P \leftarrow \phi; A \leftarrow V; F \leftarrow \phi; O \leftarrow \phi$;
 Select $u = \{0\} \in A$;
 $F \leftarrow F \cup u; A \leftarrow A - F$;
 while $|\Pi| \leq n$ **do**
 $e = (u, \pi(u)) = \text{selectaction}(u)$;
 $\Pi \leftarrow \Pi \cup \pi(u)$;
 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 $|\Pi| \stackrel{?}{=} n$ **then**
 if $K2(D, n, u, \Pi) > TH_k$ **then**
 reward all $LA \in O$;
 else
 penalize all $LA \in O$;
 end if
 $TH_k = \text{Max}(TH_k, K2(D, n, u, \Pi))$;
 else
 Do nothing;
 end if
 $k \leftarrow k + 1$;
 $q_\pi = Pr(\Pi) = \prod_{\pi(u) \in \Pi} p_{\pi(u)}^u$;
 Enable all disables actions of $LA \in O$;
until $(q_\pi > P_{TH} \vee k > K)$

با رأس ریشه شروع کرده و بر اساس بردار احتمال انتخاب اقدام‌های آن اقدامی را انجام می‌دهد. این کار باعث تغییر وضعیت یکی دیگر از آتاماتاها به وضعیت Fire و واقع شده در سمت دیگر یال متناظر با اقدام انجام شده 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 مورد استفاده قرار می‌گیرد.

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

۱. تشکیل یک جایگشت توسط eDLA: در ابتدای کار تمامی

آتاماتاها در 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 [۴۵] با انجام یک اقدام توسط هر آتاماتا، سایر آتاماتاهایی 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.

۲. 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 روابط ۶ و ۵ 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 بیشینه مقدار امتیاز DAG‌های قبلی را نشان می‌دهد.

۳. ارزیابی اقدام انجام شده توسط eDLA: بر اساس مقایسه انجام

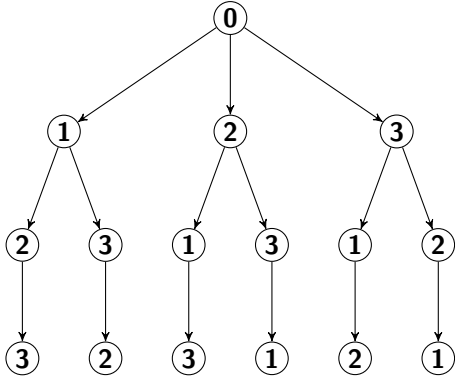
شده در مرحله قبل کلیه آتاماتاهای eDLA جریمه یا پاداش داده می‌شوند. اگر امتیاز جایگشت ایجاد شده از مقدار آستانه بیشتر باشد، کلیه آتاماتاها پاداش داده می‌شوند وگرنه جریمه می‌شوند. الگوریتم جریمه یا پاداش می‌تواند یکی از الگوریتم‌های یادگیری خطی مورد بحث در بخش قبلی باشد.

۴. به روزرسانی مقدار آستانه مقایسه و آماده‌سازی eDLA برای

اجرای بعدی: مقدار آستانه به روزرسانی می‌شود. کلیه آتاماتاها دوباره به وضعیت Active بر می‌گردند و تمامی اقدام‌های بلاک شده در مراحل قبل به وضعیت قابل انجام بر می‌گردند. eDLA برای اجرای بعدی آماده می‌شود.

چنانکه ملاحظه می‌شود در این الگوریتم از ساختار eDLA برای هدایت جایگشت‌ها به سمت بهترین جایگشت که بتواند معیار $K2$ را بیشینه کند استفاده می‌کنیم. شبکه‌کد روش جدید پیشنهادی در الگوریتم ۴ ارائه شده است.

۵ بررسی ریاضی روش پیشنهادی



شکل ۹: درخت جایگشت‌های روی سه عنصر با شماره‌های ۱ و ۲ و ۳

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

جایگشت	احتمال
$\sigma_1 = (1, 2, 3)$	$q_1(k) = p_{\downarrow}(k)p_{\downarrow}^{\downarrow}(k)$
$\sigma_2 = (1, 3, 2)$	$q_2(k) = p_{\downarrow}(k)p_{\downarrow}^{\downarrow}(k)$
$\sigma_3 = (2, 1, 3)$	$q_3(k) = p_{\downarrow}(k)p_{\downarrow}^{\downarrow}(k)$
$\sigma_4 = (2, 3, 1)$	$q_4(k) = p_{\downarrow}(k)p_{\downarrow}^{\downarrow}(k)$
$\sigma_5 = (3, 1, 2)$	$q_5(k) = p_{\downarrow}(k)p_{\downarrow}^{\downarrow}(k)$
$\sigma_6 = (3, 2, 1)$	$q_6(k) = p_{\downarrow}(k)p_{\downarrow}^{\downarrow}(k)$

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 را نشان دهد. $c_i = 1 - d_i$ 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 هامیلتونی در گراف eDLA) برای مقادیر به اندازه کافی بزرگ از تکرار الگوریتم یک زیرمارتینگل است. در مرحله دوم کرانه‌هایی برای این احتمال برقرار کرده و نشان می‌دهیم با انتخاب مقدار مناسب برای پارامتر نرخ یادگیری در الگوریتم L_{R-I} 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 [۴] مورد استفاده قرار گرفته است.

لم ۱. اگر $d_i(k)$ و $c_i(k)$ احتمالات مربوط به جریمه شدن و تنبیه شدن شدن جایگشت σ_i حاصل شده در k امین تکرار را نشان دهند و بردار احتمالات انتخاب هریک از جایگشت‌ها $\mathbf{q}(k)$ ، بر اساس الگوریتم ۴ تغییر کند، در این صورت امید ریاضی شرطی

$$E[q_{\sigma}(k+1)|\mathbf{q}(k)] \quad (۱۳)$$

$$\begin{aligned} &= \sum_{j=1}^{\rho} E[q_{\sigma}(k+1)|\mathbf{q}(k), \alpha_j] P[\alpha_j|\mathbf{q}(k)] \\ &= q_1(k) \{c_1(k)q_{\sigma}(k) + d_1(k)p_{\downarrow}^{\downarrow} \downarrow p_{\downarrow}^{\downarrow}\} \\ &+ q_2(k) \{c_2(k)q_{\sigma}(k) + d_2(k)p_{\downarrow}^{\downarrow} \downarrow p_{\downarrow}^{\downarrow}\} \\ &+ q_3(k) \{c_3(k)q_{\sigma}(k) + d_3(k)p_{\downarrow}^{\downarrow} \uparrow p_{\downarrow}^{\downarrow} \downarrow\} \\ &+ q_4(k) \{c_4(k)q_{\sigma}(k) + d_4(k)p_{\downarrow}^{\downarrow} \uparrow p_{\downarrow}^{\downarrow} \uparrow\} \\ &+ q_5(k) \{c_5(k)q_{\sigma}(k) + d_5(k)p_{\downarrow}^{\downarrow} \downarrow p_{\downarrow}^{\downarrow}\} \\ &+ q_6(k) \{c_6(k)q_{\sigma}(k) + d_6(k)p_{\downarrow}^{\downarrow} \downarrow p_{\downarrow}^{\downarrow}\} \end{aligned}$$

منظور از \downarrow و \uparrow در رابطه ۱۳ به ترتیب کاهش و افزایش در مقدار احتمال یک اقدام بر اساس روش یادگیری L_{R-I} است. اما:

$$p_{\downarrow}^{\downarrow} \downarrow p_{\downarrow}^{\downarrow} = p_{\downarrow}^{\downarrow}(1-a)p_{\downarrow}^{\downarrow} = q_{\sigma}(k)(1-a)$$

$$\begin{aligned} &p_{\downarrow}^{\downarrow} \uparrow p_{\downarrow}^{\downarrow} \downarrow \\ &= \{p_{\downarrow}^{\downarrow}(1-a) + a\} \left\{ p_{\downarrow}^{\downarrow} \left(1 - \frac{a}{p_{\downarrow}^{\downarrow}(1-a) + a} \right) \right\} \\ &= \{p_{\downarrow}^{\downarrow}(1-a) + a\} p_{\downarrow}^{\downarrow} - ap_{\downarrow}^{\downarrow} \\ &= p_{\downarrow}^{\downarrow} p_{\downarrow}^{\downarrow} (1-a) = q_{\sigma}(k)(1-a) \end{aligned}$$

و

$$E[q_i(k+1) - q_i(k)|\mathbf{q}(k)]$$

مطابق با رابطه ۱۱ تعریف می‌شود:

$$E[q_i(k+1) - q_i(k)|\mathbf{q}(k)] \geq \quad (۱۱)$$

$$aq_i(k) \sum_{\substack{j=1 \\ j \neq i}}^r q_j(k) [d_i(k) - d_j(k)]$$

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 ۹ 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 [۴۴] پیشنهاد شده است نرخ یادگیری در هر یک از سطوح مطابق با رابطه ۱۲ تغییر می‌کند:

$$a_{\pi_{i,j}}(k) = \frac{a_{\pi_{i,j-1}}(k)}{p_{\pi_{i,j-1}}^{\pi_{i,j}}(k+1)} \quad (۱۲)$$

در رابطه ۱۲، $a_{\pi_{i,j}}(k)$ مقدار نرخ یادگیری در زمان k برای j امین آتاماتای واقع در مسیر π_i را نشان می‌دهد [۶]. ما در ادامه، محاسبات را برای جایگشت σ_4 انجام خواهیم داد.

باشد، در این صورت مقدار

$$\Delta q_i(k+1) = E[q_i(k+1) - q_i(k) | \mathbf{q}(k)]$$

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:

$$\begin{aligned} \Delta q_i(k+1) &= E[q_i(k+1) - q_i(k) | \mathbf{q}(k)] \quad (16) \\ &\geq a q_i(k) \sum_{\substack{j=1 \\ j \neq i}}^{\rho} q_j(k) \{d_i(k) - d_j(k)\} \end{aligned}$$

چون جایگشت σ_i بهینه است بنابراین 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 رابطه ۱۶ مثبت و بنابراین برای مقادیر بزرگ k طرف راست ۱۶ 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.

اثبات. 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 $q_i(k) = q_i(k+1)$. 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 [۲۸] که در موارد مشابه نیز مورد استفاده قرار گرفته است ([۶][۲۶]) استفاده می‌کنیم.

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

$$\Gamma_i[\mathbf{q}] = \Pr[\mathbf{q}^* = e_i | \mathbf{q}(0) = \mathbf{q}]$$

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

$$S_r = \left\{ \mathbf{q} \mid q_j \geq 0, \sum_{j=1}^r q_j = 1, j = 1, 2, \dots, r \right\}$$

فرض کنید $R : S_r \rightarrow C(S_r)$ فضای حالت تمام توابع مشتق‌پذیر

$$\begin{aligned} &p_i^* \uparrow p_i^* \uparrow \\ &= \{p_i^*(1-a) + a\} \left\{ p_i^* \left(1 - \frac{a}{p_i^*(1-a) + a} \right) \right\} \\ &+ \{p_i^*(1-a) + a\} \left\{ \frac{a}{p_i^*(1-a) + a} \right\} \\ &= \{p_i^*(1-a) + a\} p_i^* - a p_i^* + a \\ &= p_i^* p_i^*(1-a) + a = q_{\mathcal{F}}(k)(1-a) + a \end{aligned}$$

با جایگذاری مقادیر بالا در رابطه ۱۳ خواهیم داشت:

$$\begin{aligned} E[q_{\mathcal{F}}(k+1) | \mathbf{q}(k)] & \quad (14) \\ &= q_1(k) \{c_1(k)q_{\mathcal{F}}(k) + d_1(k)q_{\mathcal{F}}(k)(1-a)\} \\ &+ q_2(k) \{c_2(k)q_{\mathcal{F}}(k) + d_2(k)q_{\mathcal{F}}(k)(1-a)\} \\ &+ q_3(k) \{c_3(k)q_{\mathcal{F}}(k) + d_3(k)q_{\mathcal{F}}(k)(1-a)\} \\ &+ q_{\mathcal{F}}(k) \{c_{\mathcal{F}}(k)q_{\mathcal{F}}(k) + d_{\mathcal{F}}(k) \{q_{\mathcal{F}}(k)(1-a) + a\}\} \\ &+ q_5(k) \{c_5(k)q_{\mathcal{F}}(k) + d_5(k)q_{\mathcal{F}}(k)(1-a)\} \\ &+ q_6(k) \{c_6(k)q_{\mathcal{F}}(k) + d_6(k)q_{\mathcal{F}}(k)(1-a)\} \end{aligned}$$

با ساده‌سازی جبری ۱۴ داریم:

$$\begin{aligned} E[q_{\mathcal{F}}(k+1) | \mathbf{q}(k)] & \quad (15) \\ &= q_1(k) \{q_{\mathcal{F}}(k) - a d_1(k)q_{\mathcal{F}}(k)\} \\ &+ q_2(k) \{q_{\mathcal{F}}(k) - a d_2(k)q_{\mathcal{F}}(k)\} \\ &+ q_3(k) \{q_{\mathcal{F}}(k) - a d_3(k)q_{\mathcal{F}}(k)\} \\ &+ q_{\mathcal{F}}(k) \{q_{\mathcal{F}}(k) - a d_{\mathcal{F}}(k)q_{\mathcal{F}}(k) + a d_{\mathcal{F}}(k)\} \\ &+ q_5(k) \{q_{\mathcal{F}}(k) - a d_5(k)q_{\mathcal{F}}(k)\} \\ &+ q_6(k) \{q_{\mathcal{F}}(k) - a d_6(k)q_{\mathcal{F}}(k)\} \\ &= q_{\mathcal{F}}(k) - \left\{ a q_{\mathcal{F}}(k) \sum_{\substack{j=1 \\ j \neq \mathcal{F}}}^{\rho} q_j(k) d_j(k) \right\} \\ &+ a d_{\mathcal{F}}(k) q_{\mathcal{F}}(k) \{1 - q_{\mathcal{F}}(k)\} \\ &= q_{\mathcal{F}}(k) + a q_{\mathcal{F}}(k) \sum_{\substack{j=1 \\ j \neq \mathcal{F}}}^{\rho} q_j(k) \{d_{\mathcal{F}}(k) - d_j(k)\} \end{aligned}$$

در نتیجه:

$$\begin{aligned} E[q_{\mathcal{F}}(k+1) | \mathbf{q}(k)] - q_{\mathcal{F}}(k) &= \\ a q_{\mathcal{F}}(k) \sum_{\substack{j=1 \\ j \neq \mathcal{F}}}^{\rho} q_j(k) \{d_{\mathcal{F}}(k) - d_j(k)\} \end{aligned}$$

و حکم ثابت است.

□

لم ۲. با فرض این که جایگشت σ_i جواب مسأله یافتن جایگشت بهینه

صورت

$$U\theta_i[x, \mathbf{q}] = E \left[e^{-\frac{xq_i(k+1)}{a}} \middle| \mathbf{q}(k) = \mathbf{q} \right] \quad (19)$$

برای محاسبه مقدار امید ریاضی در رابطه ۱۹ به این ترتیب عمل می‌کنیم:

$$\begin{aligned} U\theta_i[x, \mathbf{q}] &= E \left[e^{-\frac{xq_i(k+1)}{a}} \middle| \mathbf{q}(k) = \mathbf{q} \right] \quad (20) \\ &= \sum_{j=1}^r E \left[e^{-\frac{xq_i(k+1)}{a}} \middle| \alpha_j, \mathbf{q}(k) = \mathbf{q} \right] q_j(k) \\ &= \sum_{j=1}^r q_j(k) d_j(k) e^{-\frac{xq_i(k+1)}{a}} \\ &\quad + \sum_{j=1}^r q_j(k) (1 - d_j(k)) e^{-\frac{xq_i(k)}{a}} \end{aligned}$$

رابطه بالا در حقیقت تغییرات در احتمال انتخاب جایگشت موردنظر σ_i را تحت هر نوع شرایطی و در یک گام از اجرای الگوریتم پیشنهادی مورد بررسی قرار می‌دهد. در هرگام 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 L_{R-I} 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 رابطه ۲۰ به روابط ۲۱ و ۲۲ می‌رسیم:

$$\begin{aligned} &\sum_{j=1}^r \exp(-xq_i(k+1)/a) q_j d_j \\ &= q_i d_i \exp \left(-x \prod_{e(m,n) \in \sigma_i} \{p_n^m \uparrow\} / a \right) + \quad (21) \\ &\sum_{j=1, j \neq i}^r q_j d_j \exp \left(-x \prod_{e(m,n) \in \sigma_i \cap \sigma_j} \{p_n^m \uparrow\} \prod_{e(m,n) \in \sigma_i - \sigma_j} \{p_n^m \downarrow\} / a \right) \end{aligned}$$

که در آن $p_n^m \downarrow = p_n^m(1-a)$ و $p_n^m \uparrow = p_n^m(1-a) + a$ است.

پیوسته حقیقی مقدار با مشتق محدود تعریف شده روی S_r باشد که R یک خط حقیقی است. اگر $g[\mathbf{q}] \in C(S_r)$ باشد، الگوریتم ۴ پیشنهادی نظیر یک عملگر U با میانگین

$$Ug[\mathbf{q}] = E[g[\mathbf{q}(k+1)] | \mathbf{q}(k) = \mathbf{q}]$$

عمل می‌کند که در این رابطه E نشان‌دهنده امید ریاضی است.

نشان داده شده است [۳۸] [۲۶] که U عملگری خطی است و توابع نامنفی را نامنفی نگه می‌دارد. یعنی

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

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

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

$$\Gamma_i[\mathbf{e}_j] = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (17)$$

اما حل معادله تابعی $U\Gamma_i[\mathbf{q}] = \Gamma_i[\mathbf{q}]$ برای تعیین احتمال انتخاب جایگشت بهینه برحسب احتمال اولیه توسط الگوریتم پیشنهادی، کار سختی است. برای این کار از یک کران پایین برای این تابع استفاده می‌کنیم. از گزاره ۱ می‌دانیم که احتمال انتخاب جایگشت بهینه $q_i^* \in \{0, 1\}$ است. تعریف می‌کنیم:

$$\Phi_i[x, \mathbf{q}] = \frac{e^{-\frac{xq_i}{a}} - 1}{e^{-\frac{x}{a}} - 1} \quad (18)$$

که در آن x بایستی انتخاب شود. واضح است که تابع تعریف شده در رابطه ۱۸ شرایط مرزی تعریف شده در ۱۷ را ارضا می‌کند. نشان خواهیم داد که $\Phi_i[x, \mathbf{q}]$ یک تابع زیرمنظم بوده و لذا قادر است که یک کران پایین برای $\Gamma_i[\mathbf{q}]$ تعریف کند. می‌توان نشان داد که $\Phi_i[x, \mathbf{q}]$ وقتی زیرمنظم است که $\theta_i[x, \mathbf{q}] = e^{-\frac{xq_i}{a}}$ ابرمنظم باشد. (زیرا تحت شرط مثبت بودن a همواره $e^{-\frac{x}{a}} \leq 1$ و در نتیجه مخرج کسر همواره منفی است)

لم ۳. اگر σ_i جایگشت بهینه و $\theta_i[x, \mathbf{q}] = e^{-\frac{xq_i}{a}}$ 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 $\theta_i[x, \mathbf{q}]$ ابرمنظم خواهد بود

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

از روابط ۲۴ و ۲۵ به رابطه ۲۶ می‌رسیم:

$$\begin{aligned}
 U\theta_i[x, \mathbf{q}] - \theta_i[x, \mathbf{q}] &\leq \quad (26) \\
 q_i(k) d_i(k) \exp \left\{ \frac{-x}{a} \{q_i(k) + a(1 - q_i(k))\} \right\} \\
 + \sum_{\substack{j=1 \\ j \neq i}}^r q_j(k) d_j(k) \times \exp \left\{ \frac{-x}{a} q_i(k)(1 - a) \right\} \\
 - \theta_i[x, \mathbf{q}] \sum_{j=1}^r q_j(k) d_j(k)
 \end{aligned}
 \quad (22) \quad \sum_{j=1}^r q_j(k) (1 - d_j(k)) e^{-\frac{xq_i(k)}{a}} =$$

$$\begin{aligned}
 \theta_i[x, \mathbf{q}] \sum_{j=1}^r q_j(k) \times (1 - d_j(k)) = \\
 \theta_i[x, \mathbf{q}] - \theta_i[x, \mathbf{q}] \sum_{j=1}^r q_j(k) \times d_j(k)
 \end{aligned}$$

در رابطه ۲۱ داریم:

اما

$$\begin{aligned}
 \theta_i[x, \mathbf{q}] \sum_{j=1}^r q_j(k) d_j(k) \\
 = q_i(k) d_i(k) \theta_i[x, \mathbf{q}] + \sum_{\substack{j=1 \\ j \neq i}}^r q_j(k) d_j(k) \theta_i[x, \mathbf{q}]
 \end{aligned}
 \quad (23) \quad \prod_{e_{(m,n)} \in \sigma_i - \sigma_j} \{p_n^m (1 - a)\}$$

$$\begin{aligned}
 &= \frac{\prod_{e_{(m,n)} \in \sigma_i - \sigma_j} \{p_n^m (1 - a)\} \prod_{e_{(m,n)} \in \sigma_i \cap \sigma_j} \{p_n^m (1 - a)\}}{\prod_{e_{(m,n)} \in \sigma_i \cap \sigma_j} \{p_n^m (1 - a)\}} \\
 &= \frac{\prod_{e_{(m,n)} \in \sigma_i} \{p_n^m (1 - a)\}}{\prod_{e_{(m,n)} \in \sigma_i \cap \sigma_j} \{p_n^m (1 - a)\}}
 \end{aligned}$$

فلذا:

$$\begin{aligned}
 U\theta_i[x, \mathbf{q}] - \theta_i[x, \mathbf{q}] &\leq \\
 q_i d_i (\exp \{ -\frac{x}{a} (q_i + a(1 - q_i)) \} - \theta_i[x, \mathbf{q}]) + \\
 \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j (\exp \{ -\frac{x}{a} q_i (1 - a) \} - \theta_i[x, \mathbf{q}]) = \\
 \theta_i[x, \mathbf{q}] \left(\frac{q_i d_i \exp \{ -x(1 - q_i) \} - 1 + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j (\exp \{ xq_i \} - 1)}{\sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j (\exp \{ xq_i \} - 1)} \right) = \\
 -xq_i \theta_i[x, \mathbf{q}] \left(\frac{(1 - q_i) d_i \frac{\exp \{ -x(1 - q_i) \} - 1}{-x(1 - q_i)} - 1}{-\sum_{\substack{j=1 \\ j \neq i}}^r q_j \times d_j \frac{\exp \{ xq_i \} - 1}{xq_i}} \right)
 \end{aligned}
 \quad (24) \quad U\theta_i[x, \mathbf{q}] =$$

با توجه به اینکه $q_i = \prod_{e_{(m,n)} \in \sigma_i} p_n^m$ و روابط ۱۹ و ۲۰ و ۲۱ و ۲۲ و ۲۳ خواهیم داشت:

$$\begin{aligned}
 \theta_i[x, \mathbf{q}] \left(\frac{q_i d_i \exp \{ -x(1 - q_i) \} - 1 + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j (\exp \{ xq_i \} - 1)}{\sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j (\exp \{ xq_i \} - 1)} \right) = \\
 -xq_i \theta_i[x, \mathbf{q}] \left(\frac{(1 - q_i) d_i \frac{\exp \{ -x(1 - q_i) \} - 1}{-x(1 - q_i)} - 1}{-\sum_{\substack{j=1 \\ j \neq i}}^r q_j \times d_j \frac{\exp \{ xq_i \} - 1}{xq_i}} \right)
 \end{aligned}
 \quad (27) \quad \begin{aligned}
 &= q_i(k) d_i(k) \times \exp \left\{ \frac{-x}{a} (q_i(k) + a(1 - q_i(k))) \right\} \\
 &+ \sum_{\substack{j=1 \\ j \neq i}}^r q_j(k) d_j(k) \exp \left\{ \frac{-xq_i(k)(1 - a)}{a} D(p_n^m) \right\} \\
 &+ \theta_i[x, \mathbf{q}] - \theta_i[x, \mathbf{q}] \sum_{j=1}^r q_j(k) \times d_j(k)
 \end{aligned}$$

تعریف می‌کنیم:

$$V[x] = \begin{cases} \frac{e^x - 1}{x} & x \neq 0 \\ 1 & x = 0 \end{cases}$$

فلذا

$$\begin{aligned}
 U\theta_i[x, \mathbf{q}] - \theta_i[x, \mathbf{q}] &\leq \\
 -xq_i \theta_i[x, \mathbf{q}] \left(\frac{(1 - q_i) d_i V[-x(1 - q_i)] - 1}{-\sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j V[xq_i]} \right)
 \end{aligned}
 \quad (28)$$

اما روشن است که:

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

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

$$\begin{aligned}
 \exp \left\{ \frac{-x}{a} q_i(k)(1 - a) D(p_n^m) \right\} &\leq \\
 \exp \left\{ \frac{-x}{a} q_i(k)(1 - a) \right\}
 \end{aligned}
 \quad (25)$$

شرط لازم برای ابرمنظم بودن $\theta_i[x, \mathbf{q}]$ آن است که :

$$(1 - q_i(k))d_i(k)V[-x(1 - q_i(k))] \geq \sum_{\substack{j=1 \\ j \neq i}}^r q_j(k)d_j(k)V[xq_i(k)] \quad (29)$$

یا

$$\frac{V[-x(1 - q_i(k))]}{V[xq_i(k)]} \geq \frac{\sum_{j=1, j \neq i}^r q_j(k)d_j(k)}{(1 - q_i(k))d_i(k)} \quad (30)$$

در رابطه ۳۰ داریم:

$$\frac{\sum_{j=1, j \neq i}^r q_j(k)d_j(k)}{(1 - q_i(k))d_i(k)} = \frac{\sum_{j=1, j \neq i}^r q_j(k)d_j(k)}{\sum_{j=1, j \neq i}^r q_j(k)d_i(k)} = \frac{\sum_{j=1, j \neq i}^r q_j(k) \frac{d_j(k)}{d_i(k)}}{\sum_{j=1, j \neq i}^r q_j(k)} \quad (31)$$

اما:

$$\min_{j \neq i} \left(\frac{d_j^*}{d_i^*} \right) \leq \frac{\sum_{j=1, j \neq i}^r q_j(k) \frac{d_j(k)}{d_i(k)}}{\sum_{j=1, j \neq i}^r q_j(k)} \leq \max_{j \neq i} \left(\frac{d_j^*}{d_i^*} \right) \quad (32)$$

با جایگذاری ۳۲ در ۳۰ شرط ابرمنظم بودن به ۳۳ تحویل می‌شود:

$$\frac{V[-x(1 - q_i(k))]}{V[xq_i(k)]} \geq \max_{j \neq i} \left(\frac{d_j^*}{d_i^*} \right) \quad (33)$$

تعریف می‌کنیم $H[x] = \ln V[x]$. با توجه به محدب بودن $H[x]$ می‌توان نشان داد [۳۸]

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

از آنجا که $\frac{1}{V[x]}$ 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 $\max_{j \neq i} \left(\frac{d_j^*}{d_i^*} \right) = \frac{1}{V[x^*]} \leq \frac{1}{V[x]}$ با انتخاب $x = x^*$ نابرابری ۳۰ ارضا خواهد شد که در این صورت $\theta_i[x, \mathbf{q}]$ ابرمنظم خواهد بود. □

قضیه ۴ (همگرایی الگوریتم). فرض کنید $q_i(k)$ احتمال متناظر با جایگشت بهینه σ_i را در مرحله k ام الگوریتم ۴ نشان دهد. اگر بردار

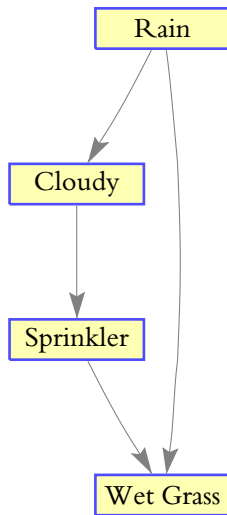
$\mathbf{q}(k) = [q_i(k)]_{i=1 \dots r}$ مطابق با این الگوریتم تغییر کند، در این صورت $\lim_{n \rightarrow \infty} q_i(n) \stackrel{a.s.}{=} 1$

اثبات. با توجه به لم ۳ و تعریف تابع $\Phi_i[x, \mathbf{q}]$ مشاهده می‌شود که به ازای هر $\varepsilon > 0$ مفروض، ثابت مثبت $1 - \varepsilon < a^*$ به گونه‌ای موجود است که نابرابری $1 - \varepsilon \leq \Phi_i[x, \mathbf{q}] \leq \Gamma_i[\mathbf{q}]$ برقرار است. این به آن معناست که $\lim_{n \rightarrow \infty} q_i(n) \stackrel{a.s.}{=} 1$. به این ترتیب اثبات قضیه ۴ کامل می‌شود. □

۶ بررسی تجربی

۱.۶ آزمایش شماره 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 عدد $f_{K2}(D, G_{original}) = 216/0.04 -$ بود. مطابق با الگوریتم ۴ کار جست‌وجوی 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 $L_R - I$ با نرخ یادگیری ۰,۰۲ برای پاداش اقدام‌های آتاماتا استفاده شده است. کار جست‌وجو برای یافتن بهترین ترتیب ۵۰ بار تکرار شده است. شبکه ایجاد شده بر اساس این الگوریتم را در شکل ۱۰ مشاهده می‌کنید. بر اساس داده‌های نمونه، این ساختار نیز دارای امتیاز $K2$ با مقدار $216/0.04 -$ می‌باشد.



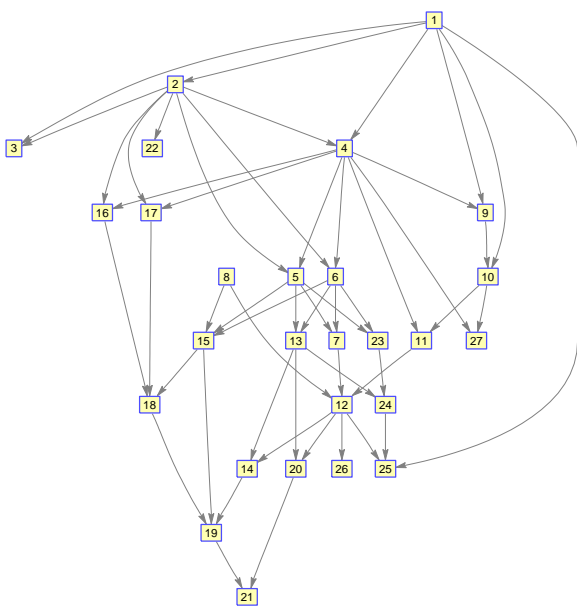
شکل ۱۰: نتیجه اجرای الگوریتم ۴ با کمک داده‌های تولید شده از شبکه بیزین شکل ۱ و بازتولید ساختار شبکه بر اساس این داده‌ها

۲.۶ آزمایش شماره ۲: بررسی عملکرد مقایسه‌ای

برای انجام آزمایش‌های این بخش از شبکه‌های ASIA [۳۱] و ICU-ALARM [۲۵] و INSURANCE استفاده کرده‌ایم. کلیه شبیه‌سازی‌ها در Matlab صورت گرفته است. فرآیند انجام کار بدین

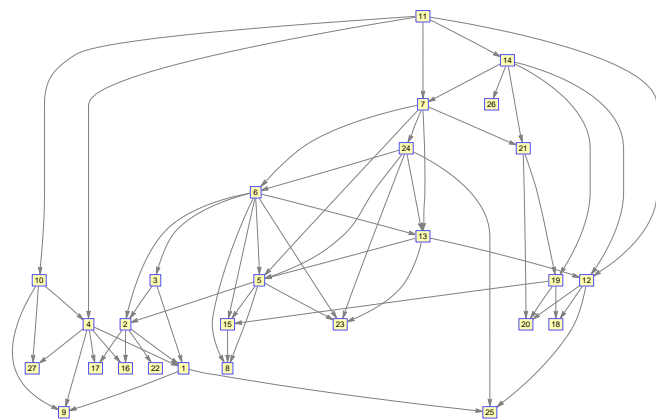
را نشان می‌دهد. امتیاز شبکه اصلی $10^5 \times 1/1208 -$ بوده است. $n_{correct}$ تعداد کمان‌های با طرفین و جهت درست در ساختار به دست آمده توسط هر الگوریتم را نشان می‌دهد. به همین ترتیب شاخص n_{direrr} تعداد کمان‌های با جهت نادرست و n_{error} تعداد کمان‌های با جهت و طرفین نادرست را نشان می‌دهند.

بر اساس نتایج این جدول مشاهده می‌شود که الگوریتم پیشنهادی که بر اساس ترکیبی از جست‌وجو در فضای جایگشت‌ها و جست‌وجوی در فضای DAG‌ها به کمک شاخص K2 است نسبت به روش‌های مبتنی بر جست‌وجو در فضای ساختار هم از نظر زمان اجرا و هم از نظر درستی ساختار پیشنهادی عملکرد خوبی دارد.



شکل ۱۱: ساختار شبکه Insurance مورد استفاده

در آزمایشی مشابه عملکرد الگوریتم پیشنهادی در برابر الگوریتم‌های جست‌وجو در DAG روی شبکه بیزی Insurance (شکل ۱۱) مورد بررسی قرار گرفته است. این شبکه دارای ۲۷ گره و ۵۲ یال می‌باشد. نتایج شاخص‌های مقایسه‌ای را در جدول ۳ مشاهده می‌کنید



شکل ۱۲: ساختار DAG القایی حاصل از ترتیب ایجاد شده توسط الگوریتم پیشنهادی برای شبکه بیزی Insurance

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 - فرآیند جست‌وجو برای یافتن بهترین جایگشت توسط الگوریتم ۴ پیشنهادی صورت گرفته است. برای مقایسه از الگوریتم‌های hc، gs 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 ۵۰۰۰۰ تایی نمونه‌گیری شده است. سپس از الگوریتم پیشنهادی و الگوریتم‌های hc، gs و gs2 خواسته شده است 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 Op-timal صورت گرفته است. در این آزمایش ساختار شبکه Optimal 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.

نتایج مقایسه برای این آزمایش را در جدول ۲ مشاهده می‌کنید. در این

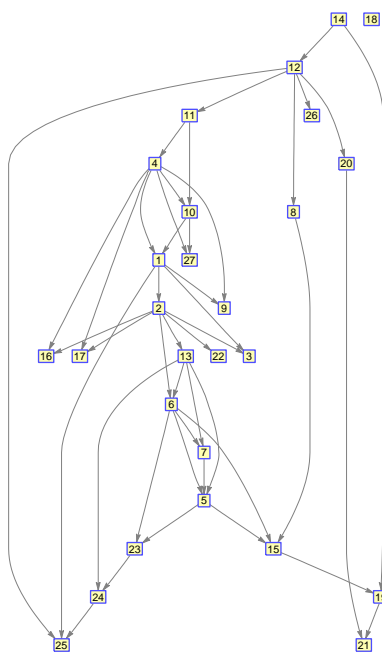
جدول ۲: نتایج شاخص‌های مقایسه‌ای برای شبکه ASIA

شاخص	ASIA			
	HC	GS	GS2	eDLA
Time	۲۱۷,۴۲	۲۲۱,۳۲	۲۲۹,۱۶	۱۲۷,۳۰
Score _{dif}	-۴۰	-۴۰	-۴۰	۰
$n_{correct}$	۴	۴	۴	۷
n_{direrr}	۲	۲	۲	۱
n_{error}	۳	۳	۳	۰

جدول شاخص Time زمان اجرای الگوریتم بر حسب ثانیه را نشان می‌دهد. شاخص Score_{dif} اختلاف امتیاز به دست آمده برای ساختار پیشنهادی الگوریتم از امتیاز K2 به دست آمده توسط شبکه Optimal

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

INSURANCE				
شاخص	HC	GS	GS۲	eDLA
<i>Time</i>	۱۴۶۹۹	۱۲۹۸۶	۱۲۴۷۷	۵۰۸۲,۷
<i>Score_{dif}</i>	-۲۵۹۰	-۲۵۹۰	-۲۲۱۲	-۲۱۷۰
<i>n_{correct}</i>	۳۰	۳۰	۳۰	۲۰
<i>n_{direrr}</i>	۱۰	۱۰	۱۰	۲۰
<i>n_{error}</i>	۴	۴	۴	۱۹



شکل ۱۳: ساختار پیشنهادی شده توسط الگوریتم جست‌وجوی حریصانه gs برای شبکه بیزی Insurance

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 $1.05 \times 10^5 -$ و هم ساختار حاصله بسیار به ساختار شبکه اصلی نزدیک است.

۷ نتیجه‌گیری

در این مقاله یک الگوریتم جدید مبتنی بر امتیاز برای یادگیری شبکه‌های بیزین مورد بررسی قرار گرفت. نوآوری این الگوریتم در استفاده از اجتماع از آتاماتاها یادگیر به عنوان واحدهای هوشمند محاسباتی ساده، برای حل این مسئله 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 بیزین با بیشترین امتیاز معیار K_2 را کسب کند، هدایت می‌کند. از مزایای این روش، سادگی روش پیشنهادی به دلیل سادگی عملکرد آتاماتاها یادگیر است. بررسی‌های تجربی روی شبکه‌های نمونه نشان‌دهنده عملکرد مناسب الگوریتم پیشنهادی در مقایسه با روش‌های جست‌وجوی 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 جست‌وجو در فضای ساختارها هستیم. علاوه بر این جست‌وجو در PDAGها و استفاده از امتیازهای هم ارز نیز بخشی از تحقیقات آتی ما را در این راستا تشکیل می‌دهد.

بر اساس نتایج جدول ۳ مشاهده می‌شود که اولاً الگوریتم پیشنهادی به لحاظ زمان اجرا نسبت به الگوریتم‌های جست‌وجو در فضای ساختارها عملکرد بسیار بهتری دارد. علاوه بر این امتیاز ساختار القایی حاصل از جایگشت 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 انجام شده است. از روی شبکه واقعی یک نمونه ۱۰۰۰۰ تایی گرفته شده است. سپس بر اساس ترتیب واقعی متغیرها ساختار شبکه Op-timal با استفاده از معیار بیزین K_2 محاسبه شده است. شبکه واقعی دارای ۴۶ یال بوده است. شبکه Optimal دارای ۴۷ یال و امتیاز $1.05 \times 10^5 -$ است. سپس بر اساس داده‌های نمونه‌گیری شده از هر یک از الگوریتم‌ها خواسته شده تا کار یادگیری ساختار شبکه بیزین را انجام دهند. نتایج مقایسه‌ای را در جدول ۴ مشاهده می‌کنید.

جدول ۴: نتایج شاخص‌های مقایسه‌ای برای شبکه ICU-ALARM

ICU-ALARM				
شاخص	HC	GS	GS۲	eDLA
<i>Time</i>	۴۰۵۰۰	۴۱۰۰۰	۴۰۰۰۰	۱۰۵۰۰
<i>Score_{dif}</i>	-۱۱۱۰	-۱۱۱۰	-۱۱۱۰	-۷۴۰
<i>n_{correct}</i>	۳۲	۳۲	۳۲	۳۳
<i>n_{direrr}</i>	۱۲	۱۲	۱۲	۱۱
<i>n_{error}</i>	۷	۷	۷	۲۲

نکته بارزی که در این شبکه نیز به چشم می‌خورد آن است که الگوریتم پیشنهادی مبتنی بر eDLA به دلیل جست‌وجو در 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 جست‌وجو در فضای PDAGها 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 جست‌وجوی حریصانه در فضای ساختارها حدود ۵۰٪ کاهش نشان می‌دهد اما در مقایسه با زمان اجرای الگوریتم پیشنهادی

- [15] CHICKERING, D., GEIGER, D., AND HECKERMAN, D. Learning Bayesian networks: Search methods and experimental results. *proceedings of fifth conference on artificial intelligence and statistics* (1995), 112–128. 2
- [16] COOPER, G. F., AND HERSKOVITS, E. A Bayesian method for the induction of probabilistic networks from data. *Machine learning* 9, 4 (1992), 309–347. 1, 2, 4
- [17] DALY, R., SHEN, Q., AND AITKEN, S. Using ant colony optimization in learning Bayesian network equivalence classes. *Proceedings of the 2006 UK Workshop on ...* (2006). 2
- [18] DE CAMPOS, L., AND HUETE, J. On the use of independence relationships for learning simplified belief networks. *International Journal of Intelligent Systems* 12 (1997), 495–522. 2
- [19] DE CAMPOS, L., AND PUERTA, J. Stochastic local algorithms for learning belief networks: Searching in the space of the orderings. *Symbolic and Quantitative Approaches to ...* (2001), 228–239. 1, 2
- [20] FRANCOIS, O. BNT structure learning package: documentation and experiments. tech. rep., Technical Report FRE CNRS 2645). Laboratoire PSI, Universit et INSA de Rouen, 2004. 3
- [21] FRIEDMAN, N., AND KOLLER, D. Being Bayesian about network structure. *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence* (2000), 201–210. 2
- [22] HECKERMAN, D. A tutorial on learning with Bayesian networks. Tech. Rep. November, Microsoft, 1996. 1
- [23] HECKERMAN, D., AND CHICKERING, D. M. Learning Bayesian Networks : The Combination of Knowledge and Statistical Data Metrics for Belief Networks .: *Machine learning* 20, 3 (1995), 197–243. 1, 2
- [24] HSU, W., GUO, H., PERRY, B., AND STILSON, J. A Permutation Genetic Algorithm For Variable Ordering In Learning Bayesian Networks From Data. *GECCO* (2002). 2
- [25] INGO A. BEINLICH HENRI JACQUES SUERMONDT, R. M. C. G. F. C. The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks. Tech. Rep. KSL-88-84, Knowledge Systems, AI Laboratory, 1989. 13
- [26] LAKSHMIVARAHAN, S., AND THATHACHAR, M. Bounds on the Convergence Probabilities of Learning Automata. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 6, 11 (1976), 756–763. 10, 11
- [27] LAM, W., AND BACCHUS, F. Learning Bayesian Belief Networks: an Approach Based on the MDL Principle. *Computational Intelligence* 10, 3 (Aug. 1994), 269–293. 1, 2
- [28] LARRAÑAGA, P., AND POZA, M. Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. *Pattern Analysis and ...* 18, 9 (1996), 912–926. 4
- [1] مریخیبیات، فرشاد. الگوریتم‌های بهینه‌سازی الهام گرفته از طبیعت. نص، ۱۳۹۱. ۲
- [۲] ملاخلیلی‌مبیدی، محمدرضا، و مبیدی، محمدرضا. حل مسأله درخت پوشای کمینه تصادفی از طریق بهینه‌سازی جایگشت: یک رهیافت مبتنی بر آتاماتای یادگیر توزیع شده. گزارش طرح تحقیقی، ۱۳۹۲. ۷
- [۳] ملاخلیلی‌مبیدی، محمدرضا، و مبیدی، محمدرضا. یک الگوریتم جدید مبتنی بر آتاماتای یادگیر توزیع شده توسعه یافته برای یادگیری پارامتری شبکه بی‌زی. نشریه مهندسی برق و مهندسی کامپیوتر ایران (۱۳۹۲)، ۸–۱۷. ۷
- [۴] ملاخلیلی‌مبیدی، محمدرضا، و مبیدی، محمدرضا. یک چارچوب مبتنی بر آتاماتای یادگیر توزیع شده توسعه یافته برای حل مسأله یافتن زیرگراف بهینه تصادفی. نشریه مهندسی برق و مهندسی کامپیوتر ایران (۱۳۹۳)، ۱۵–۱۹. ۷، ۹
- [5] ACID, S., AND DE CAMPOS, L. M. A hybrid methodology for learning belief networks: BENEDICT. *International Journal of Approximate Reasoning* 27, 3 (Sept. 2001), 235–262. 2
- [6] BEIGY, H., AND MEYBODI, M. R. Utilizing Distributed Learning Automata To Solve Stochastic Shortest Path Problems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 14, 5 (Oct. 2006), 591–615. 9, 10
- [7] BLANCO, R., INZA, I., AND LARRANAGA, P. Learning Bayesian networks in the space of structures by estimation of distribution algorithms. *International Journal of Intelligent Systems* 18, 2 (Feb. 2003), 205–220. 2
- [8] BROUARD, T., DELAPLACE, A., AND CARDOT, H. Evolutionary Methods for Learning Bayesian Network Structures. *Open access cdn.intechopen.com*, November (2008). 2
- [9] BUNTINE, W. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering* 8, 2 (Apr. 1996), 195–210. 1
- [10] CAMPOS, L. D., AND FERNANDEZ-LUNA, J. Ant colony optimization for learning Bayesian networks. *International Journal of Approximate Reasoning* 31 (2002), 291–311. 2, 4
- [11] CASTELO, R., AND KOCKA, T. On inclusion-driven learning of Bayesian networks. *The Journal of Machine Learning Research* 4 (2003), 527–574. 5
- [12] CHICKERING, D. M. Learning Bayesian networks is NP-complete. in *Learning from data*. Springer, 1996, pp. 121–130. 1
- [13] CHICKERING, D. M. Optimal structure identification with greedy search. *The Journal of Machine Learning Research* 3 (2003), 507–554. 5
- [14] CHICKERING, D. Learning equivalence classes of Bayesian-network structures. *The Journal of Machine Learning Research* 2 (2002), 445–498. 5

- [42] SILANDER, T., AND MYLLYMAKI, P. A simple approach for finding the globally optimal Bayesian network structure. *arXiv preprint arXiv:1206.6875* (2012). [3](#)
- [43] TEYSSIER, M., AND KOLLER, D. Ordering-Based Search: A Simple and Effective Algorithm for Learning Bayesian Networks. *arXiv:1207.1429* (2012). [2](#)
- [44] THATHACHAR, M., AND RAMAKRISHNAN, K. A Hierarchical System of Learning Automata. *IEEE Transactions On Systems, Man, And Cybernetics* 11, 3 (1981), 236–241. [9](#)
- [45] THATHACHAR, MAL AND HARITA, B. Learning automata with changing number of actions. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 17, 6 (1987), 1095–1100. [6](#), [7](#), [8](#)
- [46] WANG, T., AND YANG, J. A heuristic method for learning Bayesian networks using discrete particle swarm optimization. *Knowledge and Information Systems* 24, 2 (Aug. 2009), 269–281. [2](#)
- [29] LARRANAGA, P., KUIJPERS, C., MURGA, R., AND YURRAMENDI, Y. Learning Bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 26, 4 (July 1996), 487–493. [2](#)
- [30] LEE, J., CHUNG, W., KIM, E., AND KIM, S. A new genetic approach for structure learning of Bayesian networks: Matrix genetic algorithm. *International Journal of Control, Automation and Systems* 8, 2 (Apr. 2010), 398–407. [2](#)
- [31] LEPAR, V., AND SHENOY, P. P. A Comparison of Lauritzen-Spiegelhalter, Hugin, and Shenoy-Shafer Architectures for Computing Marginals of Probability Distributions. in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)* (1998), Morgan Kaufmann, pp. 328–337. [13](#)
- [32] MARGARITIS, D. Learning Bayesian network model structure from data. [3](#)
- [33] MEYBODI, M. R. M., AND MEYBODI, M. R. Extended Distributed Learning Automata: A New Method for Solving Stochastic Graph Optimization Problems. *arXiv preprint arXiv:1308.2772* (Aug. 2013). [7](#)
- [34] MEYBODI, M. R. M., AND MEYBODI, M. R. Solving Stochastic Permutation Optimization Using Monte Carlo Sampling : A Learning Automata Based Framework. tech. rep., 2013. [7](#)
- [35] MUNTEANU, P., AND BENDOU, M. The EQ framework for learning equivalence classes of Bayesian networks. in *Proceedings 2001 IEEE International Conference on Data Mining*, IEEE Comput. Soc, pp. 417–424. [5](#)
- [36] MURPHY, K. P. *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series)*. The MIT Press, 2012. [5](#)
- [37] NIELSEN, T., AND JENSEN, F. *Bayesian networks and decision graphs*, 2 ed. . Springer, 2009. [1](#)
- [38] NORMAN, F. On the Linear Model with Two Absorbing. *Journal of Mathematical Psychology* 5 (1968), 225–241. [10](#), [11](#), [13](#)
- [39] PINTO, P. C., NAGELE, A., DEJORI, M., RUNKLER, T. A., AND SOUSA, J. A. M. C. Using a Local Discovery Ant Algorithm for Bayesian Network Structure Learning. *IEEE Transactions on Evolutionary Computation* 13, 4 (Aug. 2009), 767–779. [2](#)
- [40] SALAMA, K., AND FREITAS, A. ABC-miner: an ant-based bayesian classification algorithm. *Swarm Intelligence* (2012). [2](#)
- [41] SHIBATA, K., NAKANO, H., AND MIYAUCHI, A. A learning method for dynamic Bayesian network structures using a multi-objective particle swarm optimizer. *Artificial Life and Robotics* 16, 3 (Dec. 2011), 329–332. [2](#)