ارائه الگوریتمی جهت تسریع روش تکرار سیاست در راستی-آزمایی فرآیندهای تصمیم مارکوف با استفاده از یادگیری ماشین

محمدصادق محققی[1*]، استادیار

[1] دانشکده علوم ریاضی ـ دانشگاه ولیعصر رفسنجان ـ رفسنجان ـ ایرانـ mohagheghi@vru.ac.ir

چکیده:فرآیندهای تصمیم مارکوف (MDPs) در هوش مصنوعی و راستی‌آزمایی رسمی برای مدلسازی سیستم‌های کامپیوتری که دارای رفتارهای تصادفی و غیرقطعی هستند استفاده می‌شوند. دو دسته مهم از ویژگی‌هایی که در وارسی مدل احتمالاتی استفاده می-شوند شامل احتمال بهینه (بیشنه یا کمینه) رسیدن به حالت هدف و پاداش‌انباشته شده بهینه تا رسیدن به هدفهستند. تکرار مقدار و تکرار سیاست دو روش عددی تکراری شناخته شده برای تقریب مقادیر بهینه مورد نیاز هستند. یکی از ایرادات اساسی این روش‌ها زمان اجرایی بالای آنهاست. در این مقاله روشی جدید برای تسریع همگرایی به سیاست بهینه ارائه می‌شود که زمان اجرایی روش تکرار سیاست را کاهش می‌دهد. این روش بر پایه استفاده از یادگیری ماشین برای تخمین یک سیاست نزدیک به بهینه است. برای هر کلاس از مدل‌های ,MDP تعدادی مدل کوچک را برای مرحله آموزش و ساخت دسته‌بند در نظر می‌گیریم.دسته‌بند ساخته شده در فرآیند یادگیری، برای پیش‌بینی کنش بهینه هر حالت MDPداده شده به کار می‌رود تا یک سیاست نزدیک به بهینه را ارائه دهد. این دسته‌بند برای MDP های بزرگ از همان دسته مدل‌ها قابل استفاده است تا با پیشنهاد یک سیاست مناسب، زمان مصرفی کل را کاهش دهد. پیاده سازی روش ارائه شده در وارسی‌گر مدل PRISM نشان می‌دهد زمان اجرا به طور میانگین ۵۰ درصد کاهش می-یابد.

واژه‌های کلیدی: راستی‌آزمایی صوری، وارسی مدل احتمالاتی، فرآیندهای تصمیم مارکوف، تکرار سیاست، دسترس‌پذیری بهینه، یادگیری ماشین.

*** نویسنده مسئول، ایمیل نویسنده**

Accelerating policy iteration for verifying Markov decision processes using machine learning

MohammadsadeghMohagheghi[1*], Assistant professor,

[1,] Department of Computer science, Vali-e-Asr University of Rafsanjan,, Rafsanjan, Iran, mohagheghi@vru.ac.ir

Abstract: Markov decision processes (MDPs) are used in artificial intelligence and formal verification to model computer systems with nondeterministic and stochastic behaviors. Optimal reachability probabilities and expected accumulated rewards are two main classes of properties that are used in probabilistic model checking. Value iteration and policy iteration are two well-known iterative numerical methods to approximate the optimal values. A main challenge of these approaches is their high running time. In this paper, a new method is proposed for accelerating the convergence to the optimal policy. This method is based on using machine learning to estimate optimal policies. For each class of MDP models, we consider several small models for the training and construction step of the classifier model. The classifier model is used to predict the optimal actions of each MDP model and for suggesting a near optimal policy for large models of the same MDP class. Implementing the proposed method in the PRISM model checker shows a 50% improvement in the average runtime.

Keywords: *Formal verification, probabilistic model checking, Markov decision processes, maximal reachability probabilities, policy iteration, machine learning.*

*** corresponding author, Email of the corresponding author**

۱. مقدمه

وارسی مدل احتمالاتی یک روش راستی‌آزمایی رسمی[۱] برای تحلیل ویژگی‌های کمی و کیفی سیستم‌های کامپیوتری تصادفی است. در این روش سیستم‌های تغییروضعیت احتمالاتی[۲] برای مدلسازی سیستم‌های کامپیوتری مورد بررسی استفاده می‌شوند [1]. زنجیره‌های مارکوف درمدلسازی سیستم‌هایی که رفتاری کاملا احتمالاتی دارند به کار می‌روند. از سوی دیگر فرآیندهای تصمیم مارکوف[۳](MDPs) برای مدلسازی سیستم‌هایی که جنبه‌هایی از رفتارهای احتمالاتی و غیرقطعی را توامان دارند استفاده می‌شوند [1,2]. به عبارتی در وارسی مدل احتمالاتی از MDP ها به عنوان توسعه‌ای بر سیستم‌های تغییر وضعیت که مفاهیم عدم قطعیت و احتمالات را نیز در بر دارند استفاده می‌شود عدم قطعیت غالبا زمانی استفاده می‌شود که در مدلسازی با مسائلی مانند "در نظر گرفتن تصمیم‌گیری توسط سیستم" و یا "عدم وجود اطلاعات لازم در مورد بخش*ایی از آن*" مواجهیم. منطق درخت محاسبات احتمالاتی[۴] (PCTL) یک منطق زمانی شناخته شده برای توصیف ویژگی‌های مورد نیاز در وارسی مدل احتمالاتی است. راستی‌آزمایی ویژگی‌های PCTL در MDP ها معمولا به محاسبه احتمالات دسترس‌پذیری بهینه (کمینه یا بیشینه) یا پاداش مورد انتظار بهینه نیاز دارد. در این موارد نیاز به انجام محاسبات عددی برای وارسیویژگی‌های مورد نظر است. یکی از این روش‌های عددی فرآیند وارسی مدل را بهساخت و حل یک مساله برنامه ریزی خطی (LP)تبدیل می‌کند. هرچند پیچیدگی زمان مصرفی این روش بر حسب اندازه مدل چندجمله‌ای است اما کارایی آن برای حل بیشتر نمونه‌های مسائل واقعی پایین است[1-3].تکرار ارزش[۵] و تکرار سیاست[۶] دو روش شناخته شده دیگر برای تخمین احتمالات دسترس‌پذیری یا پاداش مورد انتظار بهینه هستند که کارایی بیشتری نسبت به روش LPدارند [2]. PRISM و STORM مشهورترین ابزارهای وارسی‌گر مدل احتمالاتی هستند که از روش‌های تکرار ارزش و تکرار سیاست پشتیبانی می‌کنند [4,5].

نیاز بالا به زمان یا حافظه معمولاً کارایی و امکان‌سنجی وارسی مدل را در همه انواع آن محدود می‌کند [2,3]. دلیل اصلی این محدودیت نیاز به ذخیره کل اطلاعات مدل در حافظه اصلی است. علاوه بر این، در مورد وارسی مدل احتمالی، محاسبات عددی بسیار زمان بر است. طیف وسیعی از رویکردها برای مقابله با این مشکلات پیشنهاد شده‌اند(برای مرور تعدادی از آنها به [3,5,6] مراجعه کنید). در بیشتر

[۱] Formal verification

[۲] Probabilistic transition systems

[۳] Markov decision processes

[۴] Probabilistic computation tree logic

[۵] Value iteration

[۶] Policy iteration

موارد، این رویکردها عمومی هستند و می‌توانند در هر روش عددی تکراری اعمال شوند. در این مقاله، بر روی روش تکرار سیاست (PI) تمرکز شده و رویکردی برای بهبود عملکرد آن ارائه می‌شود. این رویکرد متکی بر استفاده از مفهوم سیاست برای حل مساله عدم قطعیت در مدل-های MDP است. برای هر سیاست (به عنوان یک نگاشت از حالت‌ها به کنش‌ها[1])، یک زنجیره مارکوف گسسته[2] (DTMC) القایی به نام DTMC خارج قسمت تولید می‌شود. سپس از یک روش تکراری برای محاسبه (یا تقریب) احتمال دسترس‌پذیری به حالت‌های هدف استفاده می‌شود [7]. سیاست‌ها بر اساس رویکرد حریصانه بهبود می‌یابند، که در آن بهترین کنش هر حالت انتخاب می‌شود. دلیل اصلی در نظر گرفتن PI در این مقاله این است که نیازی به داشتن تمام اطلاعات یک مدل در هر تکرار نیست و در مقابل، با در نظر گرفتن DTMC های القایی مقادیر حالت‌ها به‌روز می‌شوند. از این رو تعداد کل محاسبات و مدت زمان اجرا در هر تکرار کاهش می‌یابد.

یک عیب PI در مقایسه با سایر روش‌های عددی تکراری این است که تعداد کل تکرارهای آن به دقت سیاست‌های محاسبه شده بستگی دارد[2]یعنی هر قدر که سیاست‌های انتخابی به سیاست بهینه نزدیکتر باشند و تعداد بیشتری از کنش‌های بهینه را انتخاب کنند تعداد کل تکرارها کمتر می-شود. به طور کلی، هیچ رویکرد ساده‌ای برای تقریب یک سیاست اولیه مناسب برای مدل MDP داده شده وجود ندارد [2,6]. روش PI استاندارد یک سیاست دلخواه (یا تصادفی) را به عنوان سیاست اولیه انتخاب می‌کند و سیاست‌ها را تا رسیدن به یک مورد ثابت بهبود می‌دهد. برای هر DTMC القایی،تکرارها تا زمان برآورده سازی یک معیار توقف از پیش تعریف شده ادامه می‌یابند. در تکرار سیاست اصلاح شده[3] تعداد محدودی از تکرارها برای بروزرسانی مقادیر هر DTMC القایی اعمال می‌شود.

به عنوان نوآوری اصلی این مقاله برای بهبود عملکرد تکرار سیاست، روشی با استفاده از یادگیری ماشین و درخت تصمیم برای تقریبیک سیاست مناسب برای مدل MDP داده شده ارائه می‌شود. این روش مساله یافتن سیاست بهینه را به یک مساله دسته بندی[4] تبدیل می‌کند که در آن به هر حالت یک کنش از دسته مربوطه نسبت داده می‌شود. برای مرحله آموزش از چندین نمونه کوچکتر از مدل MDP داده شده استفاده می‌شود. رویکرد مورد نظر، محاسبه سیاست بهینه برای هر یک از این مدل‌های کوچک برای آموزش دسته‌بند انتخاب شده (در این مقاله درخت تصمیم) است. در این رویکرد، حالت‌های MDP داده شده به عنوان ورودی در نظر گرفته می‌شوند. به ازای هر حالت، دسته‌بند(درخت تصمیم) ساخته شده یکی از کنش‌های MDP را به عنوان کنش بهینه پیشنهاد می‌دهد.

روش ارائه شده در این مقاله به عنوان پیش‌پردازش استفاده می‌شود. این روش برای یک مدل MDP بزرگ و ویژگیPCTL داده شده یک سیاست اولیه مناسب را بدون هیچ تکرار اضافی فراهم کند.استفاده از سیاست اولیه مناسب (در مقابل سیاست تصادفی) می‌تواند به کاهش تعداد تکرارهای لازم تا همگرایی به جواب بهینه شود. ضمنا با داشتن یک سیاست اولیه مناسب، این امکان به وجود می‌آید که از برخی تکنیک‌های شناخته شده برای DTMCها در

[1] Actions

[2] Discrete-time Markov chain

[3] Modified policy iteration

[4] Classification

جهت کاهش فضای حالات و افزایش کارایی استفاده کرد. به عنوان مثال می‌توان به روش شبیه‌سازی دوگانه[1] [8] و ترتیب توپولوژیکی برای DTMCها [9,10] اشاره کرد. در روش پیشنهادی، امیدواریم بیشترین تعداد تکرار را برای اولین DTMC داشته باشیم و آن را به مدل معادل کوچکتر تقلیل دهیم تا تعداد کل محاسبات کاهش یابد. این در حالی است که در روش تکرار سیاست استاندارد یا اصلاح شده، معمولا سیاست آغازین از دقت پایینی برخوردار است و استفاده از تکنیک‌های کاهشی برای DTMC های القایی دارای سربار زمان مصرفی بالایی خواهد بود.

روش ارائه شده در این مقاله در وارسی‌گر مدل PRISM[2] پیاده‌سازی شده و بر روی تعدادی از مثال‌های استاندارد مورد محک قرار گرفته است. نتایج تجربی از بهبود تقریبی تا بیش از ۹۰ درصد در کاهش تعداد تکرارها و زمان اجراها حکایت دارد.

۱-۱ مروری بر کارهای انجام شده

یادگیری ماشین یکی از روش‌های رایج هوش مصنوعی برای حل کارای بسیاری از مسائل در حوزه های گوناگون علوم است [22,24]. استفاده از یادگیری ماشین برای بهبود کارایی وارسی مدل احتمالاتی در سالیان اخیر مورد توجه تعدادی از پژوهشگران قرار گرفته است. استفاده از الگوریتم‌های مبتنی بر یادگیری با تمرکز بر الگوریتم‌های برنامه‌ریزی پویای بلادرنگ[3] در [9] بررسی شده است. در این روش تمرکز بر حالت‌های مهمتر و به‌روز رسانی مقادیر آنها و صرف نظر از حالت‌های کم اهمیت است. یکی از ایرادات این کار وابستگی آن به ساختارگرافی مدلMDP است به شکلی که ممکن است برای دسته‌ای از مدلها سربار محاسبات بیشتر از روش-های استاندارد شود. در [6] از روش‌های یادگیری عمیق با تاکید بر ایده‌های یادگیری تقویتی استفاده شده است تا دسته-ای از مسائل زمانبندی که با MDP ها مدل شده اند به شکل موثر حل شوند. یادگیری ماشین برای تعریف MDP ها در محیط‌های جدید در [11] استفاده شده است. در این کار احتمالات تغییر وضعیت طی یک فرآیند یادگیری تخمین زده می‌شوند. تکنیک‌های سریعتر جبر خطی عددی در [12] استفاده شده است تا زمان محاسبات تکراری و عددی کاهش یابد. در [13] از درونیابی برای تخمین سیاست بهینه استفاده شده است. هر چند ایده اصلی مقاله جاری که در بخش‌های بعد توضیح داده می‌شود نیز به تخمین سیاست بهینه می‌پردازد اما تفاوت اصلی آن با [13] در استفاده از دسته‌بندی و درخت تصمیم است. ضمنا روش ارائه شده در [13] محدود به یک مطالعه موردی[4] است در حالی که روش ما عمومیت دارد. در [14] از یادگیری ماشین برای تخمین مجموعه‌های مورد نیاز در وارسی مدل کیفی برای MDP استفاده شده است.علاوه بر این پژوهش‌هایی در دو دهه گذشته برای کاهش زمان مصرفی روش‌های تکراری عددی مورد استفاده در وارسی مدل احتمالاتی ارائه شده‌اند. برخی از این روش‌ها بر روی MDP هایی با ساختار خلوت تمرکز دارند [4,7,10,20] و برخی دیگر برای مدل‌هایی با ساختار غیرخلوت توسعه یافته‌اند [12,19,21]. هرچند این روش-های بهبود یافته قادرند با کاهش زمان مصرفی محاسبات کارایی روش‌های وارسی مدل احتمالاتی را بهبود دهند اما کابرد آنها به مدل‌های با اندازه استاندارد که حافظه کامپیوتر قادر به نگهداری اطلاعات آنها است محدود می‌شوند. از

[1] bisimulation

[2] Probabilistic Symbolic model checker

[3] Real-time dynamic programming

[4] Case study

طرف دیگر تمرکز اصلی پژوهش ما بر روی توسعه روش‌های آماری وارسی مدل برای نمونه‌های بزرگی است که روش‌های یاد شده قادر به اعمال نیستند.

همچنین کاربردهای متنوعی از DTMC ها و MDP ها در [15-18,23] ارائه شده است .

۲. تعریف‌های مربوط به مساله

در این بخش به مرور تعریف‌های مربوطه برای فرآیندهای تصمیم مارکوف، وارسی مدل احتمالاتی و یادگیری ماشین پرداخته می‌شود. توضیحات نظری بیشتر در [1-3,11] ارائه شده‌اند. در این مقاله از $Dist(S)$ برای مجموعه همه توزیع احتمالات گسسته روی فضای S داده شده استفاده می‌شود. هر توزیع احتمال تابعی به صورت $p:S\to[0,1]$. است که در آن $\sum_{s\in S} p(s)=1$.

تعریف ۱. یک فرآیند تصمیم مارکوف (MDP)به صورت چندتایی $M=(S,s_0,Act,\delta,G)$ تعریف می‌شود که S مجموعه متناهی از حالت‌های مدل، $s_0\in S$ حالت آغازین، Act مجموعه‌ای متناهی از کنش‌هاست به شکلی که برای هر حالت $s\in S$ یک یا چند کنش از Act به عنوان کنش‌های مجاز شناخته می‌شوند و با $Act(s)$ نشان داده می‌شوند، $\delta:S\times Act\to Dist(S)$ یک تابع تغییروضعیت است و $G\subset S$ یک مجموعه غیرتهی هدف است. منظور از $\delta(s,\alpha,s')$ احتمال رفتن از حالت s به حالت بعدی s' با استفاده از کنش α است. هر تغییروضعیت یک سه تایی به صورت (s,α,s') است اگر $\delta(s,\alpha,s')>0$ برقرار باشد. اندازه M که با $|M|$ نشان داده می‌شود به شکل مجموع تعداد حالت‌ها و تغییر وضعیت‌های مدل تعریف می‌شود.

MDP ها به طور گسترده در ریاضیات، علوم کامپیوتر، اقتصاد و سایر و علوم برای مدلسازی سیستم‌هایی که رفتارهای غیرقطعی و تصادفی دارند به کار می‌روند [10]. در وارسی مدل MDP ها به عنوان سیستم‌های تغییروضعیت با توضیع‌های احتمالاتی به کار می‌روند. برای هر حالت $s\in S$ و کنش مجاز $\alpha\in Act(s)$ از $Post(s,\alpha)$ برای مجموعه حالت‌های بعد از $s\in S$ با کنش α استفاده می‌شود. همچنین از $Post(s)$ برای مجموعه حالت‌های بلافاصله پس از s استفاده می‌شود:

$$Post(s,\alpha)=\{s'\in S\mid \delta(s,\alpha,s')>0\} \quad (1)$$

$$Post(s)=\{\cup_{\alpha\in Act(s)}\mid Post(s,\alpha)\} \quad (2)$$

یک زنجیره مارکوف زمان-گسسته (DTMC) یک MDP است که در آن به ازای هر حالت دقیقا یک کنش مجاز تعریف می‌شود. یک مسیر در M یک اجرای ممکن از سیستم مدل شده است که به صورت دنباله‌ای ناتهی (متناهی یانامتناهی) از حالت‌ها و کنش‌ها به شکل $\pi=s_0\xrightarrow{\alpha_0}s_0\xrightarrow{\alpha_0}\dots$ تعریف می‌شود که به ازای هر مقدار $i\geq 0$ داریم $\alpha_i\in Act(s_i)$ و $s_{i+1}\in Post(s_i,\alpha_i)$. از $Path_s$ برای مجموعه همه مسیرهای نامتناهی M که از حالت $s\in S$ شروع می‌شوند‌استفاده می‌کنیم. از $FPath_s$نیزبرای زیرمجموعه مسیرهای متناهی آن استفاده می‌کنیم. ضمنا از $\pi(i)$ برای نمایش حالت i+1-ام از مسیر π استفاده می‌شود یعنی $\pi(i)=s_i$.

مثال ۱- برای درک بهتر نقش MDPها در وارسی مدل احتمالات رباتی را در نظر بگیرید که در یک محیط شامل سه اتاق عمل می‌کند. ربات دارای سلول‌های خورشیدی است که

مقدار محدودی شارژ را فراهم می‌کنند. وظیفه ربات جستجو و پیدا کردن یک شیئ مشخص در یکی از اتاق‌ها است.

مبدا	مقصد	کنش	احتمال
1	4	s	.03
1	5	s	.07
1	2	s	.9
1	4	f	.15
1	5	f	.15
1	2	f	.7
2	4	s	.08
2	5	s	.02
2	3	s	.8
2	1	s	.1
2	4	f	.09
2	5	f	.06
2	3	f	.7
2	1	f	.15
3	4	s	.06
3	5	s	.04
3	3	s	.7
3	2	s	.2
3	4	f	.07
3	5	f	.08
3	3	F	.6
3	2	F	.25
4	4	D	1
5	5	D	1

جدول ۱- نمایش اطلاعات تغییروضعیت MDP مثال ۱

فرض کنید ربات از قبل نمی‌داند شیئ در کدام اتاق قرار دارد و احتمال وجود آن در هر اتاق یک سوم است. ربات در هر یک از اتاق‌ها می‌تواند در یکی از دو وضعیت آهسته یا سریع کار کند. در وضعیت سریع احتمال یافتن شیئ بیشتر خواهد بود اما احتمال اتمام شارژ نیز بیشتر می‌شود. مدت زمان حضور در هر اتاق یک ساعت است. در صورتی که قبل از اتمام جستجوی یک اتاق شارژ ربات تمام شود ماموریت شکست خورده گزارش می‌شود. از طرفی در صورتی که قبل از اتمام شارژ، یک ساعت جستجو به اتمام برسد ربات می‌تواند به اتاق بعدی گام بگذارد.همچنین به دلیل وجود برخی شرایط محیطی در صورت عدم یافتن شیئ مورد نظر احتمال بازگشت به خانه قبل نیز وجود دارد. برای مدلسازی رفتار این ربات یک MDP در نظر گرفته شده کهاطلاعات هر یک از تغییروضعیت‌های ممکن آن در جدول ۱ ارائه شده است. این MDP شامل ۵ حالت است که حالت‌های ۱ تا ۳ نشان دهنده قرار داشتن در اتاق‌های سه گانه است. حالت ۴ نشان دهنده اتمام شارژ و در نتیجه شکست جستجو و حالت ۵ نشان دهنده یافتن شیئ مورد نظر است.در وارسی مدل این ربات علاقه‌مند به یافتن بیشترین احتمال رسیدن به حالت هدف (یافتن شیئ مورد جستجو) هستیم. در این مثال حالت اول به عنوان شروع در نظر گرفته می‌شود. مقادیر احتمالات بر اساس شرایط محیطی تخمین زده شده است.

۲-۱ سیاست

مفهوم سیاست یک سازوکار استاندارد برای حل انتخاب‌های غیرقطعی در مدلهای MDP است که با استفاده از آن می‌توان به تحلیل رفتارهای احتمالاتی این مدل‌ها پرداخت. در این مقاله تنها سیاست‌های قطعی و بی حافظه[1] در نظر گرفته می‌شوند که برای محاسبه ویژگی‌های ارائه شده در PCTL (که در ادامه توضیح داده می‌شوند)کافی هستند [2,9].

تعریف ۲ (سیاست) یک سیاست قطعی برای یک MDP مثل M به صورت تابع $\sigma : FPath_s \to Act$ تعریف می‌شود که برای هر مسیر متناهی به شکل $\pi = s_0 \xrightarrow{\alpha_0} s_0 \xrightarrow{\alpha_0} \ldots \xrightarrow{\alpha_{i-1}} s_i$ یک کنش مجاز $\alpha_i \in Act(s_i)$ را انتخاب می‌کند. یک سیاست σ بی حافظه نامیده می‌شود اگر تنها به آخرین حالت مسیر وابسته باشد. از Adv_M برای مجموعه همه سیاست‌های قطعی M استفاده می‌شود.

[1] Memory-less and deterministic

برای سادگی در ادامه کار برای کنش‌های هر حالت مثل s تنها از ترتیب آن کنش استفاده می‌شود. برای هر حالت $s \in S$ قرارداد می‌کنیم $\sigma(s)=0$ اگر سیاست σ اولین کنش M را انتخاب کند، $\sigma(s)=1$ اگر سیاست σ دومین کنش M را انتخاب کند و به همین صورت برای سایر کنش‌ها نگاشت متناسب تعریف می‌شود.برای MDP مثال ۱ یک سیاست ممکن است به حالت‌های ۱ و ۲ کنش s (معادل با حرکت آهسته) و برای حالت سوم کنش f را انتخاب کند.

برای هر سیاست σ یک DTMC متناظر (که به آن DTMC خارج قسمت می‌گویند) قابل تعریف است که می‌توان از آن برای تحلیل رفتار سیستم تحت σ استفاده کرد. محاسبه احتمال دسترس‌پذیری بیهنه برای یک MDP داده شده به مساله محاسبه یک سیاست بهینه که احتمال رسیدن به یکی از حالت‌های هدف با شروع از s_0 را کمینه یا بیشینه می‌کند، کاهش می‌یابد [1,7].

تعریف ۳- (DTMC خارج قسمت) برای MDP داده شده M و سیاست معین و بی حافظه σ (نه لزوما بهینه)، DTMC خارج قسمت به صورت $M^\sigma=(S,s_0,P)$ تعریف می‌شود که در آن S و s_0 همانند M تعریف می‌شود و $P:S\times S\to[0,1]$ تابع تغییروضعیت متناظر بوده و به صورت $P(s,s')=\delta(s,\sigma(s),s')$ تعریف می‌شود[2]. گاه به P ماتریس تغییر وضعیت نیز می‌گویند. از $Path_s^\sigma$ برای مجموعه همه مسیرهای متناهی M^σ با شروع از حالت s استفاده می‌شود.

2-2 احتمالات دسترس‌پذیری

در فرایند وارسی مدل از انواع منطق زمانی[1] برای توصیف ویژگی‌های مورد نظر استفاده می‌شود. توسعه‌های احتمالاتی منطق زمانی مانند PCTL و LTL احتمالی برای توصیف خصوصیاتی استفاده می‌شوندکه لازم است در برابر ساختارهای احتمالی مورد بررسی قرار گیرند[1,2,11]. یک کلاس مهم از خصوصیات PCTL برای MDP ها مجموعه احتمالات دسترس‌پذیری بهینه[2] است. به عنوان مثال، حداقل یا حداکثر احتمال رسیدن به یکی از حالت‌های G در هنگام شروع از حالت $s \in S$ به صورت زیر تعریف می‌شود:

$$P_s^{\min}(G)=\inf_{\sigma\in Adv_M} p_s^\sigma(G) \quad (۳)$$

$$P_s^{\max}(G)=\sup_{\sigma\in Adv_M} p_s^\sigma(G) \quad (۴)$$

که در رابطه‌های فوق $p_s^\sigma(G)$ به عنوان فضای احتمال روی $Path_s^\sigma$ تعریف می‌شود:

$$p_s^\sigma(G)=\Pr ob_s^\sigma(\{\pi\in Path_s^\sigma \mid \exists i.\pi(i)\in G\}) \quad (5)$$

برای سادگی در ادامه این مقاله احتمالات دسترس‌پذیری بیشینه را در نظر می‌گیریم. احتمالات دسترس‌پذیری کمینه به شکل مشابه محاسبه می‌شوند. محاسبه احتمالات دسترس-پذیری بیشینه به محاسبه سیاست بهینه σ^* کاهش می‌یابد که معادله زیر (معروف به معادله *بلمن*) را ارضا کند:

$$p_s^{\sigma^*}(G)=\begin{cases}1 & if\ s\in G\\ \max\limits_{\alpha\in Act(s)}\Big(\sum\limits_{s'\in Post(s,\alpha)} P(s,\alpha,s').p_{s'}^{\sigma^*}(G)\Big) & if\ s\notin G\end{cases}$$

به شکل مشابه سیاست بهینه سیاستی است که بهترین کنش‌ها را به هر حالت نسبت دهد:

[1] Temporal logic

[2] Optimal reachability probabilities

$$\sigma^*(s) = \arg\max_{\alpha \in Act(s)} \Big(\sum_{s' \in Post(s,\alpha)} P(s,\alpha,s').p_{s'}^{\sigma^*}(G) \Big) \quad (۷)$$

بیشتر وارسی‌گرهای مدل احتمالاتی مانند PRISM و STORM از روش‌های عددی تکراری مانند تکرار ارزش و تکرار سیاست برای حل معادله‌های بلمن استفاده می‌کنند[4,5].برای ربات مثال ۱ احتمال بیشینه رسیدن به حالت هدف پس از اعمال روش تکرار ارزش در PRISM برابر با 0.5131 و احتمال کمینه رسیدن به حالت هدف برابر با ۰,۳۸۹۴ به دست آمده است. یعنی در بهترین عملکرد روبات (با انتخاب بهترین سیاست) با احتمال ۰,۵۱۳۱ قبل از اتمام شارژ ربات قادر خواهد بود شیئ مورد نظر را پیدا کند. در ادامه این بخش به مرور روش تکرار سیاست پرداخته می‌شود. جزئیات بیشتر در مورد این روش‌ها و شرایط همگرایی آنها در [2,7] ارائه شده‌است.

۲-۳ تکرار سیاست

ایده روش تکرار سیاست تولید دنباله‌ای از سیاست‌های بی‌حافظه σ_i و ساخت DTMC خارج قسمت برای هر یک از این سیاست‌ها است. شمای کلی این روش در الگوریتم ۱ ارائه شده است. در این الگوریتم از یک بردار $\bar{x}$ برای مقادیر استفاده می‌شود به شکلی که برای هر $s \in S$ از x_s برای ذخیره مقادیر تخمین زده شده برای $p_s^{\sigma}(G)$ استفاده می‌شود. پس از مقدارگذاری اولیه $\bar{x}$ در خط ۲ و انتخاب سیاست اولیه σ_1 این الگوریتم از یک روش تکراری استاندارد (مانند گاوس–سایدل [2]) برای تقریب احتمال دسترس‌پذیری هر یک از حالت‌ها در DTMC متناظر استفاده می‌کند.

پس از برآورده ساختن شرط توقف محاسبات برای هر یک از DTMC های متناظر با سیاست‌های σ_i، الگوریتم از یک روش حریصانه برای محاسبه یک سیاست بهبود یافته σ_{i+1} در خطهای ۸ تا ۱۰ استفاده کرده و محاسبات را در چرخه دیگری از تکرارها ادامه می‌دهد.

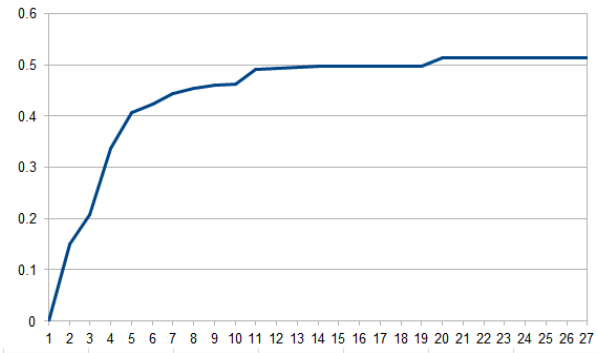
Algorithm 1. Policy Iteration for $p_s^{\max}(G)$

Input: An MDP $M = (S, s_0, Act, \delta, G)$

Output: $p_s^{\max}(G)$ for all $s \in S$?

1. For all $s \in S$ do
2. $x_s = \begin{cases} 1 & \text{if } s \in G \\ 0 & \text{otherwise} \end{cases}$
3. endfor
4. i=1;
5. Select an initial policy σ_1;
6. do
7. Compute $x_s = p_s^{\sigma}(G)$ for all $s \in S$
8. foreach $s \in S$? do
9. $\sigma_{i+1}(s) = \arg\max_{\alpha \in Act(s)} \sum_{s' \in S} \delta(s,\alpha,s').x_{s'}(G)$;
10. endfor;
11. $i = i + 1$;
12. while $\sigma_{i+1} \neq \sigma_i$
13. Return $(x_s)_{s \in S}$;

تکرار سیاست زمانی به پایان می‌رسد که دو سیاست متوالی یکسان باشند و شرط $\sigma_i = \sigma_{i+1}$ برقرار باشد. سیاست پایانی را می‌توان به عنوان تقریبی خوب از سیاست بهینه در نظر گرفت. البته ممکن است سیاست پایانی پس از اتمام محاسبات بهینه نباشد اما معمولا چنین وضعیتی برقرار است [2].به عنوان یک مثال عددی از اعمال روش تکرار سیاست برای تعیین بیشینه احتمال رسیدن به حالت هدف برای روبات مثال یک مراحل اجرا در شکل ۱ نشان داده شده است.



شکل ۱. مقادیر عددی احتمال بیشینه رسیدن به حالت هدف

دراین شکل محور افقی نشان دهنده شماره تکرار و محور عمودی نشان دهنده مقدار محاسبه شده در آن تکرار است. در واقع این نمودار چگونگی همگرا شدن به مقدار ۰,۵۱۳۱ را در تکرارهای مختلف نشان می‌دهد. برای این مثال روش تکرار سیاست دو مرتبه به اصلاح تکرار نیاز دارد. به ازای سیاست اولیه بعد از ۱۱ تکرار و به ازای سیاست دوم بعد از ۹ تکرار (در تکرار بیستم) اصلاح سیاست انجام می‌شود. نهایتا چون سیاست سوم سیاست بهینه است تکرار سیاستدر تکرار بیست و هفتم به مقدار مورد نظر همگرا شده و خاتمه می یابد.

۲-۴ تکرار سیاست اصلاح شده

یک چالش تکرار سیاست، همگرایی دیرهنگام آن به سیاست بهینه برای تعدادی از مدل‌های MDP است. نسخه استاندارد این روش از یک خط سیاست دلخواه (تصادفی) شروع می-شود و ممکن است برای همگرایی به روش‌های بهینه به چندین اصلاح سیاست نیاز داشته باشد. نسخه اصلاح شده تکرار سیاست (که در ادامه مقاله MPI نامیده می شود) برای بروزرسانی هر سیاست پس از حداکثر تعداد مشخصی تکرار (به عنوان مثال ۱۰۰ تکرار) کارمی‌کند. در نتیجه تعداد محدودی از تکرارها را برای DTMC خارج قسمت مربوطه اعمال می‌کند تا مقادیر حالت‌ها را بروز کند. یک مزیت MPI این است که از تکرارهای زیاد محاسبه برای DTMCهای اولیه جلوگیری می‌کند. با این حال MPI ممکن است سعی کند سیاست‌ها را حتی در موارد بهینه نیز بهبود بخشد. در این حالت، در مقایسه با تکرار سیاست استاندارد، ممکن است تعداد کلی اصلاحات سیاست افزایش یابد. در بخش ۳ و به عنوان نوآوری اصلی این مقاله ما روشی را پیشنهاد می‌کنیم که تعداد کلی تکرار روش MPI را کاهش دهد.

۲-۵ یادگیری ماشین

یادگیری ماشین روشی است که در آن از مجموعه‌ای از نمونه‌های اولیه برای آموزش استفاده می‌شود و هدف آن است که با تکنیک‌های مبتنی بر ریاضیات و احتمالات یک تعمیم از آن برای نمونه‌های جدید به دست آید تا امکان پیش‌بینی درمورد هر نمونه جدید ممکن فراهم آید [6]. در دسته‌بندی با نظارت[1]چندین ویژگی برای نمونه‌ها در نظر گرفته شده و از یک تابع برچسب‌گذاری براینگاشت هر یک از نمونه‌ها به دسته مورد نظر استفاده می‌شود. وظیفهیادگیری ماشین تحت نظارت تعریف رابطه بین ویژگی‌های مجموعه نمونه با برچسب‌ها است.

بسته به نوع مساله مورد استفاده و ماهیت داده‌ها، روش‌های مختلفی در دهه‌های گذشته برای یادگیری ماشین توسعه داده شده‌اند. برای مساله دسته‌بندی، بسته به اینکه داده‌ها خطی تفکیک‌پذیر باشند یا نه دسته‌بندهای متفاوتی ارجحیت خواهند داشت. به طور ویژه برای مساله مورد بررسی در این مقاله، به دلیل تفکیک خطی‌ناپذیر بودن داده‌ها بهتراست از روش‌های مناسب این رده مانند درخت تصمیم استفاده شود.

۲-۶ زبان مدلسازی PRISM

برای ارائه یک توصیف سطح بالا از یک سیستم کامپیوتری با رفتار تصادفی چندین زبان مدلسازی ارائه شده است. در این مقاله زبان مدلسازی PRISM به عنوان یک مورد شناخته شده (که در بیشتر ابزارهای وارسی مدل احتمالاتی استفاده می-شود) در نظر گرفته می‌شود. در PRISM هر مدل به

[1] Supervised

صورتمجموعه‌ای از ماژول‌ها تعریف می‌شود. هر ماژول شامل یک یا چندین متغیر با مقادیر صحیح با دامنه مقادیر تعریف شده است. حد بالای مقادیر مجاز هر متغیر را می‌توان به طور صریح و یا بر اساس یک پارامتر مشخص کرد. همچنین ممکن است تعدادی متغیر سراسری تعریف شوند. برای نام-گذاری متغیرها در حالت کلی از var_i استفاده می‌شود: var_1 برای متغیر اول، var_2 برای متغیر دوم و مانند آن. تعدادی دستور چگونگی تغییر مقادیر متغیرها را مشخص می‌کنند. توصیف سطح بالای یک مدل (که مدل برنامه نام دارد) به یک سیستم تغییروضعیت سطح پایین (که در مورد مدل‌های غیرقطعی MDP است) ترجمه می‌شود.

یک متغیر را *پارامتری* گوییم اگر دامنه آن با یک پارامتر برنامه مشخص شود. در نتیجه تعدادی مدل MDP با اندازه متفاوت بر اساس مقادیر گوناگون پارامتر ساخته خواهند شد.

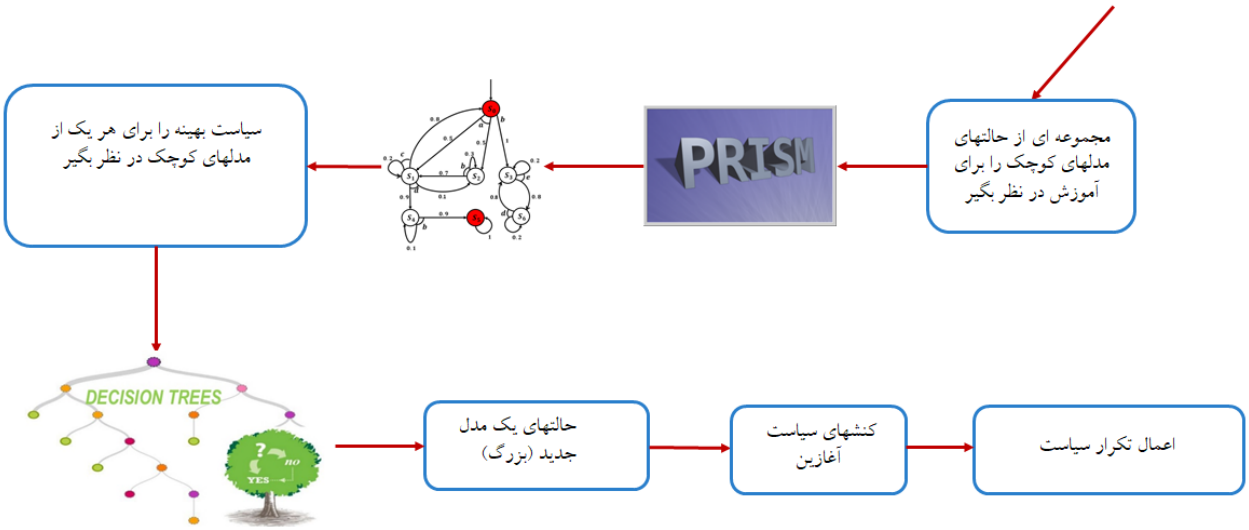
۳ روش ارائه شده برای بهبود تکرار سیاست

تکرار سیاست (PI) یکی از روش‌های استاندارد برای محاسبه احتمال دسترس‌پذیری بهینه برای یکMDP داده شده‌هاست. دقت سیاست اولیه σ_1 تاثیر قابل توجهی در کارایی این روش می‌تواند داشته باشد. انتخاب کنش‌های غیربهینه برای حالت‌ها در بیشتر موارد می‌تواند تاثیر منفی در همگرایی این روش به مقادیر واقعی داشته باشد [15] . در حالت کلی سیاست‌ها به تدریج به سیاست بهینه σ^* همگرا می‌شوند. در نتیجه دقت σ_1می‌تواند در تعداد کلی تکرارها و زمان مصرفی کلی روش تکرار سیاست موثر باشد. به عنوان نوآوری اصلی این مقاله و برای تخمین یک سیاست مناسب و استفاده از این تخمین به عنوان سیاست آغازین روشی مبتنی بر یادگیری ماشین معرفی و توسعه داده می‌شود. در ادامه این مقاله از عنوان "تکرار سیاست مبتنی بر یادگیری" که به اختصار LBPI (مخفف learning-based policy iteration) نامیده می‌شودبه عنوان نوآوری اصلی این مقاله که ایده اصلی آن درشکل ۲ نشان داده شده است، استفاده می‌کنیم. برای ارزیابی دقت سیاست داده شده σ زمان مصرفی محاسبات و دقت پاسخ نهایی در نظر گرفته می‌شود.

۳-۱ استفاده از یادگیری ماشین برای تخمین سیاست بهینه

یک روش مناسب برای کاهش تعداد تکرارهای روش PI و MPI استفاده از یک سیاست نزدیک به بهینه σ^+ به عنوان سیاست اولیه σ_1 است. به عبارت دیگر علاقه‌مند به یافتن سیاست نزدیک به بهینه σ^+ هستیم که برای اکثر حالت‌های $s \in S$ کنش‌هایی که σ^+ انتخاب می‌کند با کنش بهینه یکسان باشد یعنی شرط $\sigma^+(s) = \sigma^*(s)$ برای بخش عمده حالت‌ها برقرار باشد. برای یک MDP داده شده، روش ارائه شده برای محاسبه σ^+به عنوان تخمینی از سیاست بهینه استفاده از یک دسته‌بند برای نگاشت هر یک از حالت‌ها به کنش بهینه معادل است. از این حقیقت استفاده می‌شود که اکثر مدل‌های برنامه از یک یا چند پارامتر استفاده می‌کنند که مقدار این پارامتر(ها) در اندازه مدل تاثیرگذار است. مجموعه-ای از n پارامتر $p_1, p_2, ..., p_n$ از یک مدل برنامه PRISM و بردار $\overline{Vals}$از مقدارگذاری برای این پارامترها را در نظر بگیرید. قرارداد می‌کنیم که از $P(p_1 = val_1, ..., p_n = val_n)$ برای MDP متناظری استفاده کنیم که درآن هر یک از پارامترهای p_i دارای مقدار val_i باشد. به عنوان مثال از $P(K = 6)$ استفاده می‌شود تا MDP متناظر با مدلی را که در آن مقدار پارامتر برابر ۶ است نشان دهد.

-



شکل ۲- شمای کلی روش پیشنهادی

ایده کلی روش LBPI در شکل ۲ توضیح داده شده است. ورودی الگوریتم یک مدل برنامه به زبان PRISM است. با در نظر گرفتن مقادیر کوچک مختلف برای پارامترهای این برنامه مجموعه‌ای از حالت‌ها به دست می‌آیند. سپس با توجه به حالت‌های هدف، با اجرای PRISM مقادیر احتمال دسترس-پذیری بهینه و سیاست‌های بهینه به دست می‌آیند. با در نظر گرفتن مجموعه حالت‌های مدل‌های مورد نظر و کنش‌های بهینه متناظر هر حالت به عنوان ورودی مرحله آموزش، با استفاده از یادگیری ماشین یک درخت تصمیم به دست خواهد آمد. استفاده از این درخت تصمیم برای حالت‌های یک مدل جدید می‌تواند یک سیاست خوب را به عنوان سیاست شروع روش تکرار سیاست پیشنهاد دهد.

روش کلی یادگیری ماشین استفاده از مجموعه‌ای از داده‌های نمونه برای مرحله آموزش است. نمونه‌های مساله دارای مجموعه‌ای از ویژگی‌ها هستند که وضعیت هر یک از این نمونه‌ها را نشان می‌دهند. در مساله مورد نظر ما چندین مدل MDP کوچک به عنوان نمونه‌های مورد استفاده در نظر گرفته می‌شوند.این MDP های کوچک با در نظر گرفتن مقادیر کوچکتر $\overline{Val\_i}$ (نسبت به $\overline{Vals}$)ساخته می-شوند.همچنین هر متغیر برنامه به عنوان یکی از ویژگی‌های مساله در نظر گرفته می‌شود. در نتیجه فضای حالات مدل‌های MDP نمونه به عنوان ورودی برای مرحله آموزش در نظر گرفته می‌شوند. سیاست بهینه هر یک از MDP های نمونه تعیین کننده برچسب هر یک ازحالت‌ها و دسته متناظر آن است. به عنوان مثال اگر سیاست بهینه برای یک حالت داده شده، اولین کنش را به عنوان کنش بهینه انتخاب کند آنگاه آن حالت در دسته نخست قرار می‌گیرد و به همین ترتیب سایر دسته‌ها برای حالت‌های مختلف در نظر گرفته می‌شوند. به طور دقیق‌تر با درنظر گرفتن k مدل MDP کوچک $M_1,M_2,\ldots,M_k$در اولین گام شکل ۲و سیاست‌های بهینه معادل آنها $\sigma_1^*,\sigma_2^*,\ldots,\sigma_k^*$ که در گام بعد محاسبه شده‌اند و یک مدل MDP بزرگ M_{k+1} به عنوان ورودی، باید یک دسته‌بند مناسب اعمال شود. این دسته بند به شکلی است که سیاست $\sigma^+ \cong \sigma_{k+1}^*$ به عنوان تقریب مناسبی ارائه شود که در آن برای بیشتر حالت‌های s از M_{k+1} داشته باشیم $\sigma^+(s) \cong \sigma_{k+1}^*(s)$. برای مدل برنامه داده شده P و

مجموعه‌ای از n پارامتر آن، k بردار $\overline{vals_1}, \overline{vals_2}, ..., \overline{vals_k}$ از مقادیر کوچک برای این پارامترها را در نظر گرفته می‌شوند (بخش نخست شکل ۲) تا k مدل MDP به شکل $M_1, M_2, ..., M_k$ را به عنوان نمونه‌های آموزش به دست آیند. هر مدل M_i به صورت $M_i = P(\overline{p} = \overline{vals_i})$ ساخته می‌شود که در آن $\overline{p} = \overline{vals_i}$ به معنای انتساب مقادیر $\overline{vals_i}$ به بردار پارامترهای $\overline{p}$ است. برای ایجاد تمایز میان مجموعه حالت-های مدل‌های مختلف و ارائه تعریف غیرمبهم از اشیاء ورودی، تعدادی ویژگی اضافه در نظر گرفته می‌شوند. با هدف تعمیم بهتر دسته بند آموزش یافته، برای هر متغیر پارامتری مثل var_j با در نظر گرفتن $bound_j$ به عنوان حد بالای مقادیر آن (که تابعی از پارامتر p_j است) یک ویژگی اضافه به صورت $bound_j - \text{var}_j$(تفاوت مقدار متغیر با حد بالای آن) تعریف می‌شود. یک سیاست دلخواه σ هر یک از حالت‌های $s \in S/G$ را به دسته متناظر آن نگاشت می‌کند.

از یک روش تکراری استاندارد می‌توان برای محاسبه سیاست بهینه هر یک از مدل‌های M_i استفاده کرد. هر چند یک روش عددی تکراری نمی‌تواند تضمین کند که سیاست دقیقا بهینه را محاسبه کند اما می‌توان با در نظر گرفتن یک مقدار ε کوچک از روش تکرار بازه [7] برای محاسبه سیاست بهینه با حداکثر خطای ε استفاده نمود. با داشتن مدل‌های MDP کوچک برای مرحله آموزش انتظار داریم زمان اجرای کمی برای مرحله آموزش در مقایسه با زمان لازم برای یافتن سیاست بهینه برای یک مدل بزرگ داشته باشیم.

برای درک بهتر روش ارائه شده شکل ۲ را در نظر بگیرید که قسمتی از یک سیاست بهینه را به عنوان نگاشتی از ویژگی‌ها (متغیرها)ی مدل برای ۳ مدل از یکMDPنمونه را نشان می-دهد که برنامه آن شامل ۵ متغیر است.همچنین ویژگی اضافه استفاده شده مقدار $bound - counter$است که در ششمین ستون نشان داده شده است.به ازای هر مقدارگذاری برای ویژگی‌ها شماره دسته متناظر در شکل 3 نمایش داده شده است. چنین نگاشتی از ویژگی‌ها به برچسب کنش‌ها برای آموزش یک دسته‌بند استفاده می‌شود.در این مثال هر حالت (مقدارگذاری ویژگی‌ها) به یکی از سه دسته نسبت داده می-شود و انتظار داریم دسته‌بند آموزش دیده بتواند دسته متناظر با هر حالت (کنش معادل) را برای یک مدل بزرگ داده شده پیش‌بینی کند. در این مقاله از درخت تصمیم برای دسته بندی استفاده می‌شود.دلیل اصلی علت انتخاب این دسته بند این است داده‌های مورد استفاده در این مقاله به راحتلی تفکیک پذیر خطی نیستند و از طرفی نتایج تجربی (بخش ۴) نشان می‌دهد این دسته بند با دقت خوبی می‌تواند کنش‌های بهینه را برای مدل‌های داده شده پیش‌بینی کند.هر چند تضمینی برای یافتن سیاست بهینه در روش ارائه شده در این قسمت وجود ندارد اما این روش می‌تواند یک تخمین مناسب برای اولین سیاست برای روش‌های PI و MPI ارائه دهد و از تعداد کلی تکرارها (در مقایسه با روش‌های استاندارد) بکاهد.باتوجه به اینکه تمرکز اصلی این مقاله بر ارائه یک تخمین مناسب برای اولین تکرار است، برای ادامه کار از روش PI استاندارد استفاده می‌شود.

برای درک بهتر چگونگی استفاده از روش ارائه شده یک بار دیگر MDP مثال ۱ را در نظر بگیرید. یاداوری می‌شود هدف از استفاده از فرآیندهای تصمیم مارکوف در این مقاله، مدلسازی مسائلی است که جنبه‌هایی از عدم قطعیت و احتمالات را با یکدیگر دارند. در صورتی که مدل حاصل به اندازه‌ای باید که حافظه جاری سیستم قادر به تحلیل آن باشد می‌توان از روش‌های استاندارد وارسی مدل احتمالاتی استفاده کرد. در مورد روبات مثال ۱در صورتی که اندازه محیط به میزانی باشد که MDP حاصل دچار مشکل انفجار حالات

نشود می‌تواند از روش‌های مثل PI برای تحلی آن استفاده کرد. اما در صورتی که MDP بیش از اندازه بزرگ باشد (که موضوع اصلی مقاله جاری است)، ابتدا با افزایش سطح انتزاع و کوچک کردن محیط عملیاتی می‌توان چندین MDP استاندارد (که حافظه جاری سیستم قادر به نگهداری اطلاعات آنهاست) به دست آود.

```
3,1,0,3,1,12  ->  2
3,1,1,0,0,12  ->  1
3,1,1,1,0,12  ->  1
3,1,1,1,1,12  ->  0
3,1,1,2,0,12  ->  1
.
.
.
3,1,0,3,1,16  ->  2
3,1,1,0,0,16  ->  1
3,1,1,1,0,16  ->  1
3,1,1,1,1,16  ->  0
3,1,1,2,0,16  ->  1
.
.
.
3,1,0,3,1,20  ->  2
3,1,1,0,0,20  ->  1
3,1,1,1,0,20  ->  1
3,1,1,1,1,20  ->  0
3,1,1,2,0,20  ->  1
.
.
.
```

شکل ۳. نگاشتی از ویژگی‌های متناظر با هر حالت به کنش‌های بهینه. برای سادگی شماره حالت نشان داده نشده است.

۳-۲ تحلیل دقت روش LBPI

معیارهای زیادی را می‌توان برای دقت یک سیاست در روش پیشنهادی این مقاله ارائه داد. روش‌های مورد استفاده استاندارد (مانند روش‌های استفاده شده در [7,9,10,15]) مقایسه زمان مصرفی و یا تعداد تکرارهای روش استاندارد و یک روش بهبود یافته است. از طرف دیگر یک روش کلی در یادگیری ماشین برای تحلیل دقت یک دسته بند، مقایسه تعداد پیش‌بینی‌های نادرست بر روی داده‌های تست است. اما در مورد مساله وارسی مدل احتمالاتی تعداد پیش‌بینی‌های نادرست الزاما نشان از معیار مناسبی برای دقت یک سیاست پیش نهادی نیست. به عنوان مثال ممکن است احتمال دسترس‌پذیری حالتی مانند s_i به طور مستقیم یا غیر مستقیم بر روی مقادیر تعداد زیادی از حالت‌های دیگر تاثیرگذار باشد. در چنین حالتی انتخاب یک کنش نادرست برای s_i می‌تواند بر روی مقادیر سایر حالت‌های وابسته (حتی در صورت انتخاب حالت بهینه برای آنها) تاثیر منفی بگذارد. برای تعریف یک معیار مناسب برای دقت محاسبات سیاست پیشنهادی σ^+، احتمالات دسترس‌پذیری محاسبه شده برای حالت‌های DTMC خارج قسمت $M^{\sigma+}$ را با مقادیر محاسبه شده با استفاده از روش تکرار بازه مقایسه می‌کنیم. برای هر حالت $s \in S$ و سیاست مورد نظر σ از $precision(s,\sigma)$ به عنوان دقت مقدار محاسبه شده برای s تحت σ استفاده شده و به صورت زیر تعریف می‌شود:

$$precision(s,\sigma) = \frac{\min(p_s^{\sigma}(G), p_s^{II}(G))}{\max(p_s^{\sigma}(G), p_s^{II}(G))}$$

توجیه این تعریف این است که برای دسترس‌پذیری کمینه و برای یک سیاست غیربهینه مقدار تخمین زده شده ممکن است بیشتر از مقدار واقعی باشد. در نتیجه مقدار کمتر در صورت کسر و مقدار بیشتر در مخرج کسر قرار می‌گیرد. بر اساس این تعریف و برای سیاست ارائه شده σ^+ توسط روش مبتنی بر یادگیری ماشین، سه کلاس برای دقت مقادیر محاسبه شده در نظر می‌شود:

$$C_1 = \{s \in S^? \mid precision(s,\sigma) \geq .99\},$$

$$C_2 = \{s \in S^? \mid .99 > precision(s,\sigma) \geq .9\},$$

$$C_3 = \{s \in S^? \mid .9 > precision(s,\sigma)\}.$$

طبق این تعریف‌ها C_1 کلاس همه حالت‌هایی است که مقادیر محاسبه شده برای احتمالات دسترس‌پذیری آنها تحت سیاست σ^+ با مقدار دقیق آن در دو رقم نخست بعد از اعشار یکسان است. C_2 شامل همه حالت‌هایی است مقدار محاسبه شده برای آنها در اولین رقم معنادار بعد از اعشار با مقدار دقیق یکسان باشد. هر اندازه که درصدر بیشتری از حالت‌ها به کلاس C_1 تعلق داشته باشند، دقت سیاست σ^+ بیشتر بوده و امیدواریم با تعداد تکرار کمتری به مقادیر مورد نظر همگرا شویم.

۴- نتایج تجربی

برای اعتبار سنجی و بررسی کارایی روش پیشنهادی LBPI و مقایسه آن با روش‌های PI و MPI پنج کلاس از MDP های استاندارد به عنوان بررسی موردی[1] در نظر می‌شوند. این پنج کلاس شامل Consensus, Zeroconf, Mer, Wlan, Firewire از مجموعه بررسی‌های موردی وارسی‌گر PRISM انتخاب شده‌اند که به طور گسترده در کارهایی از این دست استفاده شده‌اند [2,3,7,9,15-17]. برای هر دسته یک پارامتر برای ساخت MDP های گوناگون در نظر می‌شوند. در جدول ۱ برخی اطلاعات مربوط به این دسته از مدل‌ها نمایش داده شده‌اند. در این جدول نام مدل‌ها و مقادیر به کار رفته برای پارامترها در مرحله آموزش ارائه شده است. تعداد متغیرهای برنامه نشان می‌دهد برنامه PRISM هر دسته از مدل‌های شامل چه تعداد متغیر است. همچنین تعداد کل حالت‌های به کار رفته در مرحله آموزش برای هر دسته از مدل‌ها رائه شده است.

نام مدل (پارامتر)	پارامترها برای آموزش	تعداد متغیرهای برنامه	تعداد کل حالت‌ها برای آموزش
Consensus(K)	3,4,5	9	129408
Zeroconf(K)	2,3,4	22	577128
Mer(n)	30,40,50	21	1450110
Wlan(TTM)	40,50,06	13	713532
Firewire(ddl)	500,600, 700	3	510732

برای پیاده سازی روش LBPI از وارسی‌گر مدل احتمالاتی PRISM که یک نرم افزار متن باز است استفاده شده است. این پیاده‌سازی با استفاده از ساختمان داده خلوت مورد استفاده در PRISM و با استفاده از زبان C انجام شده است. هر چند مولفه‌های سطح بالاتر PRISM با استفاده از JAVA توسعه داده شده‌اند اما برای بخش‌های مربوط به محاسبات عددی آن به طور عمده از C استفاده شده است. علاوه بر این برای ساخت درخت تصمیم از زبان پایتون و کتابخانه DecisionTreeClassifier از مجموعه sckitlearn استفاده شده است.

پیاده‌سازی LBPI انجام شده اطلاعات مربوط به هر سیاست را (که توسط درخت تصمیم تولید شده) از فایل می‌خواند و از آن به عنوان سیاست شروع استفاده و محاسبات عددی را مطابق الگوریتم ۱ ادامه می‌دهد. برای هماهنگی بین هر مدل و اطلاعات سیاست متناظر از یک برنامه shell استفاده شده است. محاسبات مربوط به ساخت درخت تصمیم در ویندوز و سایر محاسبات در اوبونتو ۲۰ و بر روی کامپیوتری با ۸ گیگابایت حافظه اصلی و پردازنه اینتل corei-7 انجام شده‌اند.

[1] Case study

با استفاده از مجموعه نمونه‌های آموزشی و با به کارگیری درخت تصمیم، برای هر دسته از مدل‌های پنجگانه یک درخت تصمیم متناظر ایجاد می‌شود. برای بررسی تاثیر روشLBPI بر کاهش زمان اجرایی، برای هر دسته از مسائل، تعدادی مدل بزرگتر از مدل‌های آموزشی در نظر گرفته شده و سیاست بهینه با استفاده از درخت تصمیم ساخته شده است.

به منظور اعتبارسنجی و تحلیل تاثیرگذاری روش پیشنهادی این مقاله بر روی کاهش زمان اجرایی روش PI، تعداد گام-های لازم برای همگرایی پاسخ‌ها و نیز زمان اجرا برای هر یک از نمونه‌های موردی گزارش و بررسی می‌شود. همچنین از آنجایی که تعداد تکرارها در روش‌های تکراری استاندارد بر روی دقت محاسبات تاثیر مستقیم دارد، میزان خطای ناشی از عدم بهینگی سیاست آغازین را نیز به عنوان معیاری دیگر در نظر می‌گیریم.

در جدول ۲ نتایج حاصل از اجرای روش PI با استفاده از سیاست‌های ساخته شده پیشنهادی توسط درخت تصمیم با روش‌های PI استاندارد و MPI مقایسه شده است. در تمام اجراها از $\varepsilon = 10^{-6}$ به عنوان مقدار آستانه همگرایی جواب‌ها استفاده شده است.همچنین در تمام موارد زمان اجرایی بر حسب ثانیه داده شده است. در این جدول نخست نام کلاس-های مدل و سپس مقادیرپارامترهای مورد استفاده برای ساخت مدل‌ها و نیز تعداد حالت‌های هر یک از مدل‌های ساخته شده گزارش شده است. برای مقایسه کارایی روش LBPI با دو روش دیگر تعداد تکرارهای انجام شدهو نیز زمان مصرفی تا همگرایی جواب‌ها ارائه شده است. همچنین دقت روش LBPI با استفاده از شاخص بخش ۲-۳ در جدول ۲ آمده است. بر اساس این شاخص مشخص شده است که چه درصدی از کل فضای حالات هر مدل به کلاس C_1 (با دقت بالای ۹۹٪)، چه درصدی به کلاس C_2 و چه درصدی به کلاس C_3 تعلق دارد.علاوه بر این میزان خطای سیاست پیشنهادی در ستون خطای LBPI گزارش شده است. برای محاسبه این خطا ابتدا روش تکرار بازه بر روی هر مدل اعمال شده است که مقدار دقیق احتمال بهینه رسیدن به حالت هدف با شروع از s_0 را تا ۶ رقم اعشار می‌دهد. سپس روش تکرار بازهبر روی DTMC خارج قسمت حاصل از اعمال سیاست پیشنهادی σ^+ اجرا می‌شود. مقدار گزارش شده از این روش دقت σ^+در تعیین احتمال دسترس‌پذیری بیشینه تا ۶ رقم اعشار را نشان می‌دهد. با مقایسه این مقدار با مقدار محاسبه شده نخست، دقت نسبی روش LBPIبه دست می‌آید که بر این اساس خطای LBPI در جدول ۲ گزارش شده است. از آنجا که در روش PI و MPI معمولا از یک سیاست تصادفی به عنوان سیاست آغازین استفاده می‌شود و با یک نمونه نمی-توان نتیجه‌گیری کرد، برای تحلیل بهتر روش LBPI، روش PI را با در نظر گرفتن ۲۰ سیاست تصادفی تحلیل کرده‌ایم. به ازای هر سیاست تصادفی مقدار عددی احتمال دسترس‌پذیری را در DTMC خارج قسمت محاسبه کرده و میانگین مقادیر و بهترین جواب را در نظر گرفته و مقدار خطای نسبی هر یک را در جدول ۲ (تحت عنوان RPI) گزارش کرده‌ایم.

با توجه به نتایج تجربی، در بیشتر موارد روش LBPI قادر به کاهش تعداد کل تکرارها و زمان محاسبات است. در مورد کلاس Consensus روش MPI تقریبا دوبرابر سریع‌تر از روش PI بوده و روش پیشنهادی LBPIبین ۲۵ تا ۳۰ درصدتعداد تکرارها و زمان مصرفی را در مقایسه با روشMPI کاهش می‌دهد. برای همه مدل‌های این دسته سیاست پیشنهادی درخت تصمیم قادر است مقادیر مورد نظر را با خطای کمتر از ۱ درصد محاسبه کند و تمام حالت‌ها در کلاس C_1 قرار می‌گیرند.

بـرای مـدل‌هـای کـلاس Zeroconf درحـالی کـه تعـداد تکرارهای روش‌های PI و MPI به یکـدیگر نزدیکنـد روش LBPI قادر به کاهش تعداد تکرارها و زمان مصرفی تا حـدود ۹۰ درصد است. هر چند در این روش دقت محاسبات انجـام شده مبتنی بر سیاست پیشنهادی چندان زیاد نیسـت و بخـش قابل توجهی از فضای حالات هر مدل به کلاس C_3 تعلق دارد اما تعداد کل محاسبات کاهش چشم‌گیری می‌یابد.در این مثال خطای روش LBPI با افزایش K کمی بیشتر می‌شود. در حالی که سیاست‌های تصادفی به طور میـانگین دقـت پـایینی بـرای نمونه‌های این دسته دارند امکان رسیدن به یک سیاست خوب با در نظر گرفتن تعدادی سیاست تصادفی افزایش مـی‌یابـد و در نتیجه خطای بهترین جواب کم است.

باید دقت داشت روشLBPI از سیاست پیش‌بینی شده توسط درخت تصمیم تنها برای سیاست آغازین اسـتفاده مـی‌کنـد و پس از انجام محاسبات تکراری بر روی DTMC ساخته شده، برای بهبود سیاست از روشی مشابه الگوریتم ۱ اسـتفاده مـی-کند. در نتیجه دقت محاسبات پس از پایان اجرای کامل LBPI (و نه تنها سیاست آغازین)مشابه دقت محاسـبات روش‌هـای PI و MPIاست.

برای مدل‌هـای Mer و Wlan تعـداد تکرارهـا و نیـز زمـان مصرفی در روش LBPI تقریبا نصف تعداد تکرارهـا و زمـان مصرفی روش‌های PI و MPI است. در مورد مدل‌های کلاس Mer تعداد قابل توجهی از حالت‌ها به کلاس C_3 تعلق دارند. برای این دسته از مسائل و نیز دسته Zeroconf مـی‌تـوان از روش‌های قدرتمندتر یادگیری مثل روش یادگیری عمیق برای بهبود دقت محاسبات استفاده کرد. برای مدل‌هـای firewire تعداد تکرارهای روش‌هـای PI و MPI بـا تعـداد تکرارهـای روش LBPI تقریبا برابر است. در این دسته از مسائل سیاست نخسـت کـه توسـط PRISM بـرای روش‌هـای PI و MPI انتخاب می‌شود تا حد زیادی به سیاست بهینه نزدیک است و در نتیجه تعداد تکرارها در روش مبتنی بـر یـادگیری چنـدان کاهش نمی‌یابد.

جدول ۲ نتایج حاصل از اجرای روش پیشنهادی و مقایسه آن با روش‌های تکرار سیاست استاندارد و تکرار سیاست بهبود یافته

نام مدل	پارامتر	تعداد حالت	تعداد تکرار PI	تعداد تکرار MPI	تعداد تکرار LBPI	زمان اجرای PI	زمان اجرای MPI	زمان اجرای LBPI	درصد حالت-های C1	درصد حالت-های C2	درصد حالت-های C3	خطای LBPI	میانگین خطای RPI	کمترین خطای RPI
Consensus	16	166016	83775	44193	29693	372	219	152	100	0	0	<.0001	.6452	.5813
	24	247936	145698	85243	57659	653	419	308	100	0	0	.0013	.718	.6219
	32	329856	268844	138449	90996	1342	720	519	100	0	0	.0045	.7649	.6548
	40	411776	395012	221453	128254	1961	1252	811	100	0	0	.0179	.7826	.6693
	6	798471	1254	1045	80	19	17.2	4.1	97.6	1.2	1.2	.0004	.2317	.0327
	10	3001911	1881	1742	147	93	87	7.8	78.9	7.5	13.6	.0098	.2519	.0428

.0315	.2484	.018	48.5	0	51.5	10.5	121	125	160	1846	1889	4427159	14	Zeroconf
.03258	.261	.0211	51.1	7.5	41.4	15.8	144	141	144	1989	1912	5477150	18	
.0785	.0872	.0178	31.9	21.9	46.2	9.4	17	17	43	83	83	1773664	300	Mer
.0763	.0921	.0181	31.9	21.9	46.2	17	31	31	43	83	83	2955064	500	
.0592	.1049	.0183	31.9	22	46.1	28	47	47	43	84	84	4431814	750	
.0674	.112	.0184	32	21.9	46.1	36	63	63	43	86	86	5908564	1000	
.0069	.0573	.0003	0.2	0.4	99.4	9.2	16	16	161	279	279	1920725	100	Wlan
.0094	.0829	.0004	0.6	0.9	98.5	14	31	31	179	480	480	2192135	110	
.0121	.0921	.0006	0.8	1.2	98	17	40	40	210	585	585	2463145	120	
.0158	.0895	.001	0.8	1.3	97.9	19	56	56	235	795	795	2733747	130	
.3416	.3755	<.0001	0	0	100	9.3	9.2	9.2	987	985	985	412062	1000	Firewire
.3284	.4119	<.0001	0	0	100	25	25	25	1475	1473	1473	717562	1500	
.3402	.4058	.0007	.01	.02	99.7	33	33	33	2006	2005	2005	1023062	2000	
.3593	.4319	.0008	0.2	0.3	99.5	42	42	42	2494	2493	2493	1328562	2500	

۵- نتیجه‌گیری

در این مقاله، با استفاده از یادگیری ماشین و درخت تصمیم روشی جدید برای ارائه یک تخمین مناسب از سیاست اولیهبرای یک فرآیند تصمیم مارکوف ارائه شد. هدف اصلی این روش کاهش زمان مصرفیروش PI در وارسی مدل برای سیستم‌های مدل شده با MDP ها است. نتایج تجربینشان می‌دهد تعداد تکرارها و در نتیجه زمان مصرفی روش PI با استفاده از ایده ارائه شده تا ۹۰ درصد کاهش می‌یابد در حالی که دقت محاسبات این روش کاهش چندانی نمییابد. برای پژوهش‌های آینده می‌توان از سایر روش‌ها مانند شبکه‌های عصبی عمیق برای ارائه یکسیاست نزدیک بهبهینه استفاده کرد. همچنین با در نظر گرفتن مدل-های بزرگ می‌توان امکان بسط روش ارائه شده در راستای وارسی مدل آماری را مورد بررسی قرار داد.

مراجع

[1] C. Baier, J.P. Katoen. Principles of model checking. MIT press, 2008.

[2] V. Forejt, M. Kwiatkowska, G. Norman, D. Parker. "Automated verification techniques for probabilistic systems." In International school on formal methods for the design of computer, communication and software systems, pp. 53-113. Springer, Berlin, Heidelberg, 2011.

[3] J.P. Katoen. "The probabilistic model checking landscape." In Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, pp. 31-45. 2016.

[4] C. Hensel, S. Junges, J.P. Katoen, T. Quatmann, M. Volk. The probabilistic model checker storm. International Journal on Software Tools for Technology Transfer. pp. 1-22. Jul 2021

[5] M. Kwiatkowska, G. Norman, D. Parker. PRISM 4.0: Verification of Probabilistic Real-time Systems. In Proc. 23rd International Conference on Computer Aided Verification (CAV'11), volume 6806 of LNCS, pp. 585-591, Springer, 2011

[6] T. Gros, H. Hermanns, J. Hoffmann, M. Klauck, M. Steinmetz. Deep Statistical Model Checking. In: Gotsman A., Sokolova A. (eds) Formal Techniques for Distributed Objects,

Components, and Systems. FORTE 2020. Lecture Notes in Computer Science, vol 12136. 2020

[7] C. Baier, J. Klein, L. Leuschner, D. Parker, S. Wunderlich. Ensuring the reliability of your model checker: interval iteration for markov decision processes. In: CAV 2017. LNCS, vol. 10426, pp. 160180. Springer, Cham 2017.

[8] C. Baier, P.R. DArgenio, H. Hermanns. On the probabilistic bisimulation spectrum with silent moves. Acta Informatica. vol. 57, pp. 465-512. 2020.

[9] T. Brázdil, C. Krishnendu, C. Martin, F. Vojtěch, J. Křetínský, M. Kwiatkowska, D. Parker, M. Ujma. "Verification of Markov decision processes using learning algorithms." In International Symposium on Automated Technology for Verification and Analysis, pp. 98-114. Springer, Cham, 2014.

[10] L. Gui, J. Sun, S. Song, Y. Liu, J.S. Dong. SCC-Based Improved Reachability Analysis for Markov Decision Processes. Int. Conf. Formal Engineering Methods, Luxembourg, 35 November, Volume 8829 of Lecture Notes in Computer Science, pp. 171186. Springer, Berlin. 2014.

[11] Bouchekir, Redouane, and Mohand Cherif Boukala. "Toward implicit learning for the compositional verification of Markov decision processes." In International Conference on Verification and Evaluation of Computer and Communication Systems, pp. 200-217. Springer, Cham, 2018.

[12] Shlakhter, Oleksandr, and Chi-Guhn Lee. "Accelerated modified policy iteration algorithms for Markov decision processes." Mathematical Methods of Operations Research 78, no. 1 pp. 61-76. 2013.

[13] A. Rataj, B. Wona-Szczeniak. Extrapolation of an Optimal Policy using Statistical Probabilistic Model Checking. Fundamenta Informaticae 157, no.4 pp. 443-461. 2018.

[14] M. Mohagheghi, K. Salehi. "Machine Learning and Disk-based Methods for Qualitative Verification of Markov Decision Processes." In ICTERI Workshops, pp. 74-88. 2020.

[15] M. Kwiatkowska, D. Parker, H. Qu. Incremental Quantitative Verification for Markov Decision Processes. IEEE/IFIP 41st Int. Conf. Dependable Systems &Networks (DSN), Hong Kong, China, IEEE CS Press. pp. 359-370. June 2011

[16] A. Hartmanns, M. Klauck, D. Parker, T. Quatmann, E. Ruijters. The quantitative verification benchmark set. In: Vojnar, T., Zhang, L. (eds.) TACAS.LNCS, vol. 11427, pp. 344350. Springer, Cham. 2019

[17] F. Ciesinski, C. Baier, M. Größer, J. Klein. "Reduction techniques for model checking Markov decision processes." In Fifth International Conference on Quantitative Evaluation of Systems, pp. 45-54. IEEE, 2008.

[18] کاهش تاخیر مبتنی ,طاهره صالحی – محسن معدنی – مژده مهدوی ,بر فرآیند تصمیم گیری مارکوف در شبک های بی سیم بی خودرویی چهارمین کنفرانس بین المللی مطالعات نوین در علوم کامپیوتر 1396 –و فناوری اطلاعات

[19] M. Mohagheghi, J. Karimpour, and A. Isazadeh. "Prioritizing methods to accelerate probabilistic model checking of discrete-time Markov models." *The Computer Journal* 63, no. 1 (2020): 105-122.

[20] M. Mohagheghi, J. Karimpour, and Ayaz Isazadeh. "Improving Modified Policy Iteration for Probabilistic Model Checking." *Computer Science* 23, no. 1 (2022).

[21] J. Karimpour, A. Isazadeh, M. Mohagheghi, and K. Salehi. "Improved Iterative Methods for Verifying Markov Decision Processes." In *Fundamentals of Software Engineering: 6th International Conference, FSEN 2015, Tehran, Iran, April 22-24, 2015. Revised Selected Papers*, pp. 207-214. Springer International Publishing, 2015.

[۲۲] نجفی س.، نوفرستی س.، "تعیین برش زمانی در الگوریتم زمانبندی نوبت گردشی با استفاده از یادگیری ماشین"، مجله محاسبات نرم، جلد ۱۰، شماره ۲، ص. ۴۳-۳۲، ۱۴۰۰.

[۲۳] صادقیان ر.، "تعیین نقطه تعادل در بازی‌های پویای مارکفی گسسته دونفره با احتمالات انتقال صرفا تحت تاثیر استراتژی‌های رقیب"، مجله محاسبات نرم، جلد ۱۱، شماره ۱، ۱۴۰۱

[۲۴] ویسی ه.، قایدشرف ه.، ابراهیمی م.، "بهبود کارایی الگوریتم‌های یادگیری ماشین در تشخیص بیماری‌های قلبی با بهینه‌سازی داده‌ها و ویژگی‌ها"، مجله محاسبات نرم، جلد ۸، شماره ۱، ص. ۸۵-۷۰، ۱۳۹۸