مجله کنترل جلد ۳، شماره ۱، بهار ۱۳۸۸، صفحه ۱۱-۱

بررسی یادگیری تقویتی و خواص سیاست بهینه در مسائل جدولی با استفاده از روش‌های کنترل دیجیتال

سید مصطفی کلامی هریس [۱]، ناصر پریز [۲]، محمدباقر نقیبی سیستانی [۳]

[۱] فارغ التحصیل کارشناسی ارشد برق- کنترل، دانشگاه فردوسی مشهد، دانشکده‌ی مهندسی، گروه مهندسی برق، sm.kalami@gmail.com

[۲] دانشیار، دانشگاه فردوسی مشهد، دانشکده‌ی مهندسی، گروه مهندسی برق، گرایش کنترل، n-pariz@um.ac.ir

[۳] استادیار، دانشگاه فردوسی مشهد، دانشکده‌ی مهندسی، گروه مهندسی برق، گرایش کنترل، mb-naghibi@um.ac.ir

چکیده: فرآیند تصمیم‌گیری مارکوف یا MDP، یکی از مسائلی است که دارای کاربردهای وسیعی در زمینه‌های مختلف علمی، مهندسی، اقتصادی و مدیریت است. بسیاری از فرآیندهای تصمیم‌گیری، دارای خاصیت مارکوف می‌باشند و به صورت یک مسأله‌ی تصمیم‌گیری مارکوف قابل بیان هستند. یادگیری تقویتی یکی از رویکردهایی است که برای حل MDP به کار می‌رود، و به نوبه‌ی خود از برنامه‌ریزی پویا یا DP استفاده می‌کند. در این نوشتار الگوریتم ارزیابی سیاست، که در بحث یادگیری تقویتی و DP برای حل MDP به کار می‌رود، به صورت معادله‌ی دینامیکی یک سیستم دیجیتال یا گسسته-زمان بازنویسی شده است. به این ترتیب این امکان به وجود آمده است که بتوان با بهره‌گیری از روش‌های موجود در کنترل دیجیتال، به بررسی خواص معادلات به دست آمده پرداخت و تحلیل مناسبی از رفتار عامل یادگیرنده، تحت سیاست‌های مختلف، به عمل آورد. روش مذکور برای تحلیل دو مسأله‌ی جدولی استفاده شده است و سپس نتایج کلی در خصوص مسائل جدولی بیان و اثبات شده‌اند. به عنوان مثال، نتایج به دست آمده نشان می‌دهند که سیاست بهینه برای هر مسأله‌ی جدولی، در چارچوب کنترل دیجیتال، به صورت یک سیستم مرده نَوِش یا Dead Beat قابل توصیف است.

کلمات کلیدی: برنامه‌ریزی پویا، سیستم‌های کنترل دیجیتال، فرآیندهای تصمیم‌گیری مارکوف، کنترل تصادفی، یادگیری تقویتی.

Abstract: Markov Decision Process (MDP) has enormous applications in science, engineering, economics and management. Most of decision processes have Markov property and can be modeled as MDP. Reinforcement Learning (RL) is an approach to deal with Markov Decision Processes. RL methods are based on Dynamic Programming (DP) algorithms, such as Policy Evaluation, Policy Iteration and Value Iteration. In this paper, policy evaluation algorithm is represented in the form of a discrete-time dynamical system, namely a Discrete-Time Control system. Hence, using Discrete-Time Control methods, behavior of agent and properties of various policies, can be analyzed. Two grid-world problems are solved and analyzed using this approach. Therefore general case of grid-world problems is addressed, and some important results are obtained for this type of problems, For example, equivalent dynamical system of an optimal policy for a grid-world problem, is always a dead-beat system in the framework of Discrete-Time Control systems.

Keywords: Dynamic Programming, Discrete-Time Control Systems, Markov Decision Process, Reinforcement Learning, Stochastic Control.

۱- مقدمه

مسائل جدولی[1]، نوع خاص از فرآیند تصمیم‌گیری مارکوف[2] یا MDP هستند. فرآیندهای تصمیم‌گیری مارکوف برای اولین بار توسط بلمن در سال ۱۹۵۷ و سپس توسط هُوارد در سال ۱۹۶۰ معرفی گردید و مورد بررسی قرار گرفت [1]. اولین کاربرد مشخصی که برای MDP ثبت شده است، استفاده از آن در سازمان‌دهی راه‌های ایالت آریزونا در سال ۱۹۷۸ بوده است [1]. همچنین کاربردهایی نظیر مدیریت حیات وحش، مدیریت تولید و کارخانه، انبارداری و حمل و نقل، از جمله کاربردهایی

[1] Grid-world Problems
[2] Markov Decision Process

هستند که برای MDP پیشنهاد شده‌اند [1]. فهرست کاملی از کاربردهای MDP و مدل‌های نیمه مشاهده‌پذیر MDP در نوشته‌هایی توسط پوترمن [1]، کاساندرا [2]، پایت [3] ، هو و همکارانش [18] و سو و همکارانش [19] آمده است. یافتن یک سیاست بهینه در مسأله‌ای که به صورت MDP مدل شده است، یکی از مباحثی است که در نظریه‌ی بهینه‌سازی و کنترل بسیار مورد توجه بوده است [1,5,7,18,19]. برای حل چنین مسأله‌ای از شیوه‌های برنامه‌ریزی خطی[1] و برنامه‌ریزی پویا[2] استفاده شده است و تغییرات متعددی در این الگوریتم‌ها به وجود آمده‌اند تا سرعت پاسخ‌دهی مناسبی برای این روش‌ها تأمین شود [1,5,8,18,19].

روش‌های اولیه‌ای که برای حل این مسأله با استفاده از برنامه‌ریزی پویا مورد استفاده قرار گرفته‌اند، به نام‌های ارزیابی سیاست[3]، تکرار سیاست[4] و تکرار ارزش[5] شناخته می‌شوند و در منابعی چون [4]، [5]، [6] و [7] مورد بررسی قرار گرفته‌اند. این روش‌ها پاسخ دقیقی را برای هر مسأله‌ی تصمیم‌گیری MDP به دست می‌دهند و می‌توان به قدر نیاز، جواب نهایی را به جواب واقعی نزدیک کرد. وجود جواب برای این روش‌ها، با استناد به قضایای مربوط به آنالیز توابع در فضای‌های برداری اندازه‌پذیر، به خصوص قضیه‌ی نقطه‌ی ثابت [17]، با روش‌های مختلفی مورد بررسی قرار گرفته است [6,7,16]. شرایط دقیقی که برای وجود جواب در یک مسأله‌ی MDP که توسط برنامه‌ریزی پویا حل می‌شود، به خوبی مورد مطالعه قرار گرفته‌اند و گردآوری شده‌اند [5,6,7,16]. چیزی که بلمن از آن به عنوان نفرین ابعاد یاد کرده است، یک مشکل جدی در زمینه‌ی استفاده از این روش‌ها به وجود آورده است، که باعثِ به وجود آمدن روش‌های تقریبی و سریع‌تر شده است که معمولا در قالب مباحث یادگیری تقویتی[6] [5,7,9] و یا برنامه‌ریزی عصبی-پویا[7] [4,8,9] مطرح می‌شوند.

روش‌های مورد استفاده برای بهتر کردن فرآیند حل MDP به چهار گروه اصلی قابل تقسیم می‌باشند. روش‌های گروه اول از ویژگی‌های ساختاری مسأله، برای تسهیل فرآیند به دست آوردن جواب یا سیاست بهینه، استفاده می‌کنند [11,12,26]. روش‌های بعدی، بر خلاف روش‌های گروه اول، منجر به پیدا شدن جواب بهینه برای مسأله نمی‌شوند. بلکه در این روش‌ها، جواب‌های به دست آمده جواب‌های شبه بهینه هستند و با فرض‌هایی که منجر به ساده شدن مسأله‌ی اصلی شده‌اند، به دست آمده‌اند [10,15]. روش‌های گروه دوم، خود به دو زیر گروه اصلی قابل تقسیم هستند. در روش‌های زیر گروه اول، از مدل‌های ساده شده برای حل مسأله استفاده می‌شود و به این ترتیب می‌توان یک سیاست شبه بهینه را برای مسأله‌ی اصلی پیدا نمود [10,15]. در روش‌های زیر گروه دوم، ساختار سیاست مجهول، به شکلی خاص فرض می‌شود و به این ترتیب نوعی ساده‌سازی در فرآیند حل مسأله به وجود می‌آید [4,6,10,15]. روش‌های گروه سوم، از تقریب توابع ارزش حالت، توابع ارزش حالت-عمل و معادلات برنامه‌ریزی پویا به دست آمده‌اند. در این روش‌ها از شیوه‌هایی همچون تجمیع حالات[8]، نمایش بر اساس توابع پایه[9] و استخراج خواص[10] استفاده شده است [4,5,7,9,10,13,14]. روش‌های گروه چهارم، همانند روش کلاسیک تکرار سیاست، در فضای سیاست تعریف و به کار برده می‌شوند. این روش‌ها، سعی بر این دارند که با شیوه‌های خاصی، شکل‌گیری سیاست بهینه را سریع‌تر کنند و به این ترتیب فرآیند حل مسأله سریع‌تر شود.

فرآیند حل یک MDP، در بحث یادگیری تقویتی، به نام یادگیری شناخته می‌شود و شکل‌گیری شیوه‌های تصمیم‌گیری جدید، همواره در اثر جمع‌آوری اطلاعات جدید می‌باشد. یکی از شیوه‌هایی که برای تسریع فرآیند یادگیری پیشنهاد شده است، استفاده از شیوه‌ی باز-استعمال است که برای تسریع فرآیند یادگیری و همچنین طبقه‌بندی اطلاعات به دست آمده از تجارب قبلی مورد استفاده قرار گرفته است [20-23]. همچنین با بهینه کردن ساختار الگوریتم تکرار سیاست نیز، نتایج مناسبی به دست آمده‌اند که می‌توان برای نمونه به [1]،[6]،[7]،[19]، [24]، [25] و [26] اشاره نمود.

روش‌های بسیاری برای بهبود عملکرد الگوریتم‌های برنامه‌ریزی پویا و همچنین الگوریتم‌های یادگیری تقویتی در محیط‌های مارکوف، ابداع شده‌اند. هر کدام از این روش‌ها با رویکردی خاص، قصد دارند در کمترین تعداد تکرار و کمترین زمان به یک پاسخ بهینه یا شبه بهینه، دسترسی پیدا کنند. مشکل اصلی در بسیاری از مسائل، نبودن شناخت کافی در مورد چگونگی یک پاسخ بهینه است. برای تشخیص یک سیاست بهینه و انجام مقایسه میان دو سیاست، معیاری سریع و ساده وجود ندارد. به این ترتیب که برای ارزیابی یک سیاست، حتما می‌بایست سیاست مذکور، توسط عامل یادگیرنده مورد استفاده قرار بگیرد و بر اساس خروجی به دست آمده، در مورد خوبی یا بدی آن سیاست، قضاوت شود. تا کنون، علی رغم تلاش‌های بسیاری که برای بهبود عملکرد الگوریتم‌های مرتبط با حل مسأله‌ی MDP انجام شده‌اند، روشی برای تحلیلِ ریاضیِ فرآیند حلِ MDP ارائه نشده است. به همین دلیل، معیاری غیر از الگوریتم‌های زمان‌بَر برای تحلیل عملکرد عامل یادگیرنده، که از سیاستی خاص پیروی می‌کند، وجود ندارد.

در این نوشتار فرآیند حل مسائل MDP با استفاده از روش DP، به صورت یک دینامیک گسسته-زمان یا دیجیتال بیان شده است. سیستم دیجیتالی که به دست می‌آید با روش حلی که برای مسأله ارائه شده

[1] Linear Programming
[2] Dynamic Programming
[3] Policy Evaluation
[4] Policy Iteration
[5] Value Iteration
[6] Reinforcement Learning
[7] Neuro-Dynamic Programming

[8] State Aggregation
[9] Basis Function Representation
[10] Feature Extraction

است، متناظر است و می‌توان با تحلیل خصوصیات کنترلی این سیستم، خواص روش حل متناظر با آن را مشخص نمود و کیفیت جواب نهایی را حدس زد. استفاده از این معادل‌سازی، این امکان را به وجود می‌آورد که بتوان به تحلیل عملکرد یادگیری تقویتی در محیط‌های مارکوف پرداخت.

سایر بخش‌های این مقاله به صورت زیر می‌باشند. در بخش ۲، یادگیری تقویتی و ایده‌ی اصلی آن به صورت اجمالی توضیح داده می‌شوند. در بخش ۳، تعاریف ابتدایی در مورد فرآیندهای تصمیم‌گیری مارکوف و روش‌های برنامه‌ریزی پویا برای حل این نوع از مسائل، مورد بررسی قرار می‌گیرند. در بخش ۴، الگوریتم ارزیابی سیاست به صورت یک سیستم دینامیکی گسسته-زمان بیان می‌شود. در بخش ۵، دو مسأله‌ی نمونه با استفاده از مطالب بخش ۴ و روش‌های تحلیل و کنترل سیستم‌های دینامیکی گسسته-زمان مورد بررسی قرار گرفته‌اند. بخش ۶ نیز، حاوی بیان نتایج کلی در مورد مسائل جدولی و سیاست بهینه‌ی مرتبط با این نوع از مسائل است.

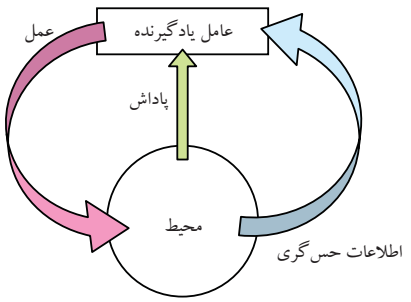
۲- یادگیری تقویتی

هدف اصلی از یادگیری، یافتن شیوه‌ای برای عملکرد در حالات مختلف است که این شیوه در مقایسه با سایرین، با در نظر گرفتن معیارهایی، بهتر است. معمولا این شیوه‌ی عملکرد، از نظر ریاضی، به صورت نگاشتی از فضای حالات به فضای اعمال، قابل بیان است. هنگامی می‌توان گفت یادگیری اتفاق افتاده است که، عاملی بر اساس تجربیاتی که کسب می‌کند به نحوی دیگر، و به احتمال زیاد بهتر، عمل کند. در این صورت می‌بایست نحوه‌ی عملکرد عامل در اثر کسب اطلاعات جدید، متفاوت از نحوه‌ی عملکرد در زمان‌های قبل از کسب این اطلاعات و تجارب باشد.

در یادگیری تقویتی، هدف اصلی از یادگیری، انجام دادن کاری و یا رسیدن به هدفی است، بدون آنکه عامل یادگیرنده، با اطلاعات مستقیم بیرونی تغذیه شود. در این روش، تنها مسیر اطلاع‌رسانی به عامل، از طریق یک سیگنال پاداش یا جریمه می‌باشد. تنها چیزی که از طریق سیگنال پاداش به عامل فهمانده می‌شود، این است که آیا تصمیم مناسبی گرفته است یا نه؟ در بسیاری از حیوانات، یادگیری تقویتی، تنها شیوه‌ی یادگیری مورد استفاده است. همچنین یادگیری تقویتی، بخشی اساسی از رفتار انسان‌ها را تشکیل می‌دهد. هنگامی که دست ما در مواجهه با حرارت می‌سوزد، ما به سرعت یاد می‌گیریم که این کار را بار دیگر تکرار نکنیم. لذت و درد مثالهای خوبی از پاداش‌ها و جرایمی هستند که الگوهای رفتاری ما و بسیاری از حیوانات را تشکیل می‌دهند.

در یادگیری تقویتی، هیچ گاه به عامل گفته نمی‌شود که عمل صحیح در هر وضعیت چیست، و فقط به وسیله‌ی معیاری، به عامل گفته می‌شود که یک عمل چقدر خوب یا چقدر بد است. عامل موظف است، با در دست داشتن این اطلاعات، یاد بگیرد که بهترین عمل کدام است. این موضوع، بخشی از نقاط قوت خاص یادگیری تقویتی است. از این طریق، مسائل پیچیده‌ی تصمیم‌گیری در اغلب اوقات می‌توانند با فراهم کردن کمترین میزان اطلاعات مورد نیاز برای حل مسأله، حل شوند. در این شیوه از یادگیری، حتی در برخی موارد، ماهیت مسأله و هدف از حل آن نیز به طور کامل و مستقیم به عامل تفهیم نمی‌شود. سیگنال پاداش، به طور ضمنی نحوه‌ی عملکرد مناسب را به عامل نشان می‌دهد و هدف از حل مسأله را مشخص می‌کند. در این حالت، هدف عامل از یادگیری به بیشینه کردن میزان پاداش دریافتی در بازه‌ای از زمان، تغییر می‌کند. این دو هدف (نحوه‌ی عملکرد مناسب و بیشینه کردن پاداش‌ها) با هم مترادف هستند و برآورده شدن هرکدام، دیگری را نیز برآورده خواهد کرد. به این طریق، عامل نحوه‌ی عملکرد مناسب را با تمرکز بر پاداش‌های دریافتی، یاد می‌گیرد. این امر به این صورت محقق می‌شود که، نگاشتی میان حالات و اعمال قابل انجام توسط عامل، پیدا می‌شود. این نگاشت که به نام سیاست شناخته می‌شود، به عامل می‌گوید که در مواجهه با حالات مختلف، چه عمل یا اعمالی را انجام دهد. تبعیت از یک سیاست خوب، قطعا عامل را به نتیجه‌ای مناسب خواهد رساند.

یک مسأله‌ی یادگیری تقویتی نوعی، در شکل ۱ نشان داده شده است. عامل یادگیرنده از طریق حس‌گرها، توصیفی از حالت محیط اطرافش را به دست می‌آورد. اطلاعات مربوط به محیط در قالب اطلاعات حس‌گری به عامل داده می‌شوند. هنگامی که عامل ، عملی را انجام می‌دهد، پاداشی را دریافت می‌کند که می‌تواند بسته به خوبی یا بدی عمل، پاداشی مثبت یا منفی باشد.



شکل ۱ - یک مسأله‌ی یادگیری تقویتی و نحوه‌ی تعامل محیط و عامل

۳- فرآیندهای تصمیم‌گیری مارکوف

بخش اعظمی از کارهای تحقیقاتی انجام شده بر روی یادگیری تقویتی، توأم با این فرض بوده‌اند که، تعامل بین عامل و محیط اطرافش را می‌توان به صورت یک فرآیند تصمیم‌گیری مارکوف یا MDP گسسته-زمان مدل‌سازی کرد. یک فرآیند تصمیم‌گیری مارکوف، فرآیندی تصادفی و گسسته-زمان می‌باشد که اغلب به صورت دسته‌ی

چهار تایی $(\mathbb{S},\mathbb{A},\mathcal{P},\mathcal{R})$ تعریف می‌شود. اجزای یک فرآیند تصمیم‌گیری مارکوف، عبارتند از:

- یک زمان‌سنج سراسری به صورت $t=0,1,\ldots,T$ برای شمارش زمانِ گسسته. (T می‌تواند نامحدود باشد.)
- $\mathbb{S}$ نشان دهنده‌ی فضای حالت فرآیند می‌باشد و مشتمل بر تمام حالات ممکنی است که عامل تصمیم گیرنده در آن‌ها قرار می‌گیرد و ملزم به تصمیم‌گیری در این حالات می‌باشد. فرض بر این است که $\mathbb{S}$ مجموعه‌ای محدود به صورت $\mathbb{S}=\{s^1,s^2,\ldots,s^n\}$ می‌باشد.
- $\mathbb{A}$ نشان دهنده‌ی فضای اعمال قابل انتخاب، برای عامل تصمیم‌گیرنده است. این مجموعه حاوی انتخاب‌ها و یا تصمیمات ممکن در هر حالت، برای عامل می‌باشد. فرض بر این است که $\mathbb{A}$ مجموعه‌ی محدودی به صورت $\mathbb{A}=\{a^1,a^2,\ldots,a^m\}$ می‌باشد.
- $\mathcal{P}$ نحوه‌ی انتقال و تحول حالات را مدل می‌کند. اگر فرآیند در حالت s باشد، و عمل a توسط عامل انتخاب شود، احتمال تغییر حالت فرآیند به حالت s' به صورت $\mathcal{P}^a_{ss'}$ تعریف می‌شود. به عبارت دیگر داریم:

$$\mathcal{P}^a_{ss'}=\Pr\{s'|s,a\} \qquad (۱)$$

در حالت کلی $\mathcal{P}$ نگاشتی به صورت $\mathcal{P}:\mathbb{S}\times\mathbb{A}\times\mathbb{S}\to[0,1]$ است. این مدل، دارای خاصیت مارکوف می‌باشد. یعنی احتمال فوق، صرفا به حالت و عمل اخیر بستگی دارد و کاملا مستقل از خاطره‌ی قبلی عامل می‌باشد. اگر s_t و a_t به ترتیب نشان دهنده‌ی حالت فرآیند و عمل انتخاب شده در زمان گسسته‌ی t باشند، آن‌گاه خاصیت مارکوف به صورت رابطه‌ی زیر قابل توصیف می‌باشد:

$$\mathcal{P}^{a_t}_{s_ts_{t+1}}=\Pr\{s_{t+1}|s_t,a_t\}=\Pr\{s_{t+1}|s_t,a_t,s_{t-1},a_{t-1},\ldots\} \qquad (۲)$$

- $\mathcal{R}$ تابع تعریف کننده‌ی امید ریاضی پاداش می‌باشد. اگر عامل در حالت s باشد و با انجام عمل a به حالت s' برود، مقدار پاداشی که دریافت می‌کند به صورت عددی تصادفی و با امید ریاضی $\mathcal{R}^a_{ss'}$ تعریف می‌شود. در حالت کلی تابع پاداش، نگاشتی به صورت $\mathcal{R}:\mathbb{S}\times\mathbb{A}\times\mathbb{S}\to\mathbb{R}$ می‌باشد. تابع $\mathcal{R}$ نیز دارای خاصیت مارکوف می‌باشد و مقدار $\mathcal{R}^a_{ss'}$ صرفا به حالت فعلی (یعنی s) و عمل فعلی (یعنی a) و حالت بعدی (یعنی s') بستگی دارد و کاملا مستقل از حالات یا اعمال قبلی می‌باشد. به عبارت دیگر:

$$\begin{aligned}\mathcal{R}^{a_t}_{s_ts_{t+1}}&=\mathbb{E}\{r_{t+1}|s_{t+1},s_t,a_t\}\\&=\mathbb{E}\{r_{t+1}|s_{t+1},s_t,a_t,s_{t-1},a_{t-1},\ldots\}\end{aligned} \qquad (۳)$$

فرض کنید عامل در زمان یا مرحله‌ی t در حالت s_t قرار دارد و عمل $a_t\in\mathbb{A}$ را انجام می‌دهد. عامل با احتمال $\mathcal{P}^{a_t}_{s_ts'}$ در زمان $t+1$ به حالت s' می‌رود و خواهیم داشت: $s_{t+1}=s'$. ضمنا عامل در نتیجه‌ی این عمل، پاداشی اسکالر به اندازه‌ی r_{t+1} خواهد گرفت که در حالت کلی، کمیتی تصادفی و با امید ریاضی $\mathcal{P}^{a_t}_{s_ts_{t+1}}$ می‌باشد.

احتمال انتخاب عمل a از طرف عامل، هنگامی که در حالت s قرار دارد، با نگاشتی به صورت $\pi:\mathbb{S}\times\mathbb{A}\to[0,1]$ تعریف می‌شود و می‌توان نوشت:

$$\Pr\{a_t=a|s_t=s\}=\pi(s,a) \qquad (۴)$$

نگاشت π با نام سیاست شناخته می‌شود و مجهول اصلی یک مسأله‌ی یادگیری تقویتی و هر مسأله‌ی تصمیم‌گیری می‌باشد [1,5,7,14].

برای مقایسه‌ی سیاست‌های مختلف با یکدیگر، می‌توان معیاری را برای سنجش آن‌ها تعریف نمود. این معیار، مقداری است که سیاست در هر حالت از فرآیند برمی‌گرداند و به عنوان خروجی[1] سیاست در حالت مذکور از آن یاد می‌شود. خروجی یک سیاست، میزانی از پاداش است که در اثر اتخاذ تصمیمات متوالی و با تبعیت از آن سیاست به دست آمده است. برای هر کدام از حالت‌ها، ارزشی در نظر گرفته می‌شود که برابر با امید ریاضی خروجی است که با شروع کردن از هر حالت و تبعیت از یک سیاست خاص به دست می‌آید. در حالت کلی، منظور از حل یک مسأله‌ی یادگیری تقویتی، پیدا کردن سیاست π^* است به نحوی که مقدار خروجی سیاست و یا ارزش هر کدام از حالت‌ها، بیشینه شوند [5,7,14]. روش‌های متفاوتی برای تعریف خروجی وجود دارند. روشی که در اکثر کاربردها معمول است و در این مقاله نیز مورد توجه قرار گرفته است، تعریف خروجی به صورت تنزیلی[2] می‌باشد. اگر ضریب تنزیل به صورت $\gamma\in[0,1]$ باشد، خروجی تنزیلی به صورت زیر خواهد بود:

$$z_t=r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\cdots=\sum_{k=0}^{\infty}\gamma^k r_{t+k} \qquad (۵)$$

در این حالت، ارزش حالت s به صورت زیر قابل بیان است:

$$\begin{aligned}V^\pi(s)&=\mathbb{E}^\pi\{z_{t+1}|s_t=s\}\\&=\sum_{a\in\mathbb{A}}\pi(s,a)\sum_{s'\in\mathbb{S}}\mathcal{P}^a_{ss'}\left(\mathcal{R}^a_{ss'}+\gamma V^\pi(s')\right)\end{aligned} \qquad (۶)$$

که در آن، منظور از $\mathbb{E}^\pi$، عملگر امید ریاضی است. اندیس‌های π نیز، صرفا برای تاکید بر این که عامل از سیاست π پیروی می‌کند، نوشته شده‌اند. رابطه‌ی (۶)، به معادله‌ی (بهینگی) بلمن معروف است [1,4,14,18,19].

یکی از روش‌هایی که برای حل این معادله و یافتن مقدار ارزش تمام حالات استفاده می‌شود، بازگشتی کردن این معادله است. این روش که مبتنی بر قضیه‌ی نقطه‌ی ثابت[3] [17] است، پیشنهاد می‌کند که معادله‌ی (۶) به صورت زیر بازنویسی شود:

[1] Return
[2] Discounted
[3] Fixed Point Theorem

$$V_{k+1}^{\pi}(s)=\sum_{a\in\mathbb{A}}\pi(s,a)\sum_{s'\in\mathbb{S}}\mathcal{P}_{ss'}^{a}\left(\mathcal{R}_{ss'}^{a}+\gamma V_{k}^{\pi}(s')\right) \quad (۷)$$

که در آن $V_k^{\pi}(s)$، تخمین kاُم از مقدار واقعی $V^{\pi}(s)$ است. با توجه به این که $|\gamma|<1$ $|\gamma|<1$ است، می‌توان استدلال کرد که $V_{k+1}^{\pi}(s)$ با یک نگاشت انقباضی[1] [17] به $V_k^{\pi}(s)$ مرتبط است. طبق قضیه‌ی نقطه‌ی ثابت [17]، این نگاشت دارای نقطه‌ی ثابتِ منحصر به فردی است که جواب معادله‌ی (۶) نیز می‌باشد. با توجه به معادله‌ی (۷)، جواب معادله‌ی (۶) به صورت زیر خواهد بود:

$$V^{\pi}(s)=\lim_{k\to\infty}V_{k}^{\pi}(s) \quad (۸)$$

فرآیند محاسبه‌ی $V^{\pi}(s)$ برای تمام حالت‌ها، در بحث یادگیری تقویتی و برنامه‌ریزی پویا به نام ارزیابی سیاست [1,5,7,14] معروف است.

۴- مدل‌سازی الگوریتم ارزیابی سیاست

فرض کنید برداری به صورت

$$v^{\pi}=\begin{bmatrix}V^{\pi}(s^1) & V^{\pi}(s^2) & \cdots & V^{\pi}(s^n)\end{bmatrix}^{T} \quad (۹)$$

تعریف شده باشد. این بردار حاوی ارزش تمام حالات یک مدل است. در این صورت می‌توان رابطه‌ی بازگشتی (۷) را، به شکل زیر برای تمام حالت بازنویسی کرد و آن را به صورت یک معادله تبدیل نمود:

$$v_{k+1}^{\pi}=\gamma\mathcal{P}v_{k}^{\pi}+\mathcal{R}=\gamma\begin{bmatrix}\mathcal{P}_{s^1s^1} & \cdots & \mathcal{P}_{s^1s^n}\\ \vdots & \ddots & \vdots\\ \mathcal{P}_{s^ns^1} & \cdots & \mathcal{P}_{s^ns^n}\end{bmatrix}v_{k}^{\pi}+\begin{bmatrix}\mathcal{R}_{s^1}\\ \vdots\\ \mathcal{R}_{s^n}\end{bmatrix} \quad (۱۰)$$

که در آن داریه‌های ماتریس‌های $\mathcal{P}$ و $\mathcal{R}$ عبارتند از:

$$\mathcal{P}_{ss'}=\sum_{a\in\mathbb{A}}\pi(s,a)\mathcal{P}_{ss'}^{a} \quad (۱۱)$$

و

$$\mathcal{R}_{s}=\sum_{a\in\mathbb{A}}\sum_{s'\in\mathbb{S}}\pi(s,a)\mathcal{P}_{ss'}^{a}\mathcal{R}_{ss'}^{a}=\sum_{a\in\mathbb{A}}\pi(s,a)\mathcal{R}_{s}^{a} \quad (۱۲)$$

می‌توان معادله‌ی (۱۰) را به صورت زیر بازنویسی کرد:

$$v_{k+1}^{\pi}=\gamma\mathcal{P}v_{k}^{\pi}+\mathcal{R}u_{k} \quad (۱۳)$$

که در آن فرض شده است که u_k به ازای تمام مقادیر $k\geq 0$، برابر با واحد باشد، که همان تعریف تابع پله‌ی واحد [27] می‌باشد. معادله‌ی (۱۳) معادله‌ی حالت یک سیستم گسسته-زمان یا دیجیتال [27] می‌باشد که متغیرهای حالت آن، ارزش‌های مربوط به حالات فرآیند تصمیم‌گیری می‌باشند. ماتریس حالت این سیستم، از ترکیب اطلاعات مربوط به محیط در قالب $\mathcal{P}_{ss'}^{a}$، اطلاعات مربوط به سیاست در قالب $\pi(s,a)$، و ضریب تنزیل به دست آمده است. بردار وروی این سیستم $\mathcal{R}$ می‌باشد که اطلاعات مربوط به محیط، سیاست و پاداش‌ها را در بر دارد. طبق قرارداد، ورودی این سیستم، همواره برابر با پله‌ی واحد در نظر گرفته می‌شود. شرط پایداری سیستم فوق، عبارت است از این که، همه‌ی مقادیر ویژه‌ی ماتریس $\gamma\mathcal{P}$، که قطب‌های سیستم توصیف شده با (۱۰) هستند ، در داخل دایره‌ی واحد قرار بگیرند [27]. برای تحقق این شرط، می‌بایست ضریب تنزیل γ در نامساوی زیر صدق کند:

$$\gamma<\frac{1}{\max\limits_{1\leq i\leq n}\left|\lambda_i(\mathcal{P})\right|}=\frac{1}{\rho(\mathcal{P})} \quad (۱۴)$$

که در آن، $\lambda_i(\mathcal{P})$ نشان دهنده‌ی مقدار ویژه[2]ی iاُم، و $\rho(\mathcal{P})$ نیز شعاع طیفی[3] ماتریس $\mathcal{P}$ می‌باشد. لذا مشاهده می‌شود که شرط $\gamma\leq 1$، الزاما تضمین کننده‌ی همگرایی سری تعریف شده در معادله‌ی (۵) و یا وجود جواب محدود برای (۶) نمی‌باشد. رابطه‌ی (۱۴)، شرط دقیق‌تری برای γ بیان می‌کند. اگر تمام مقادیر ویژه‌ی ماتریس $\mathcal{P}$، درون دایره‌ی واحد باشند، آن‌گاه سری (۵)، به ازای برخی از مقادیر γ نیز، که بزرگتر از یک هستند، همگرا خواهد بود.

۵- حل و بررسی دو مسأله‌ی نمونه

در این بخش، با استفاده از مطالب مطرح شده در بخش قبل، دو مسأله‌ی نمونه مورد حل و بررسی قرار می‌گیرند. مشاهداتی که در حل این دو مسأله انجام گرفته است، راهگشای نتیجه‌گیری‌های کلی در خصوص مسائل جدولی و مسائل مشابه هستند.

۵-۱- مسأله‌ی اول

یک مسأله‌ی جدولی را، به صورت نشان داده شده در شکل ۲، در نظر بگیرید. عاملی (مثلا یک روبات) در یکی از خانه‌های سفید رنگ این جدول قرار دارد. عامل در هر حالتی می‌تواند به سمت چپ یا راست حرکت کند. هنگامی که عامل به یکی از خانه‌های خاکستری برسد، حرکت او متوقف می‌شود. حرکت به چپ یا راست، پاداشی به اندازه‌ی ۱- در پی دارد که در واقع هزینه‌ای است که عامل برای حرکت کردن می‌پردازد. هدف از حل مسأله، پیدا کردن شیوه‌ای برای حرکت است که عامل از هر کدام از حالات، در کمترین تعداد حرکت به یکی از خانه‌های هدف برساند.

s^0	s^1	s^2	s^3	s^0

شکل ۲ - جدول مربوط به مثال مورد بررسی در بخش ۵-۱

جدول ۱ - احتمال انتخاب حرکات در حالات مختلف برای $\pi^{(p)}$

$\pi^{(p)}$	چپ (L)	راست (R)
s_1	$1-p$	p

[1] Contraction Mapping

[2] Eigenvalue
[3] Spectral Radius

s_2	$\frac{1}{2}$	$\frac{1}{2}$
s_3	p	$1-p$

سیاستی به صورت نشان داده شده در جدول ۱ را برای مسأله‌ی حاضر در نظر بگیرید. توجه کنید که $p \in [0,1]$ پارامتری است که تغییر مقدار آن، باعث ایجاد سیاست‌های مختلف برای مسأله‌ی مورد بررسی می‌شود. به این ترتیب سیاست $\pi^{(\frac{1}{2})}$، یک سیاست کاملا تصادفی و با احتمال مساوی برای حرکت‌های چپ و راست است. سیاست $\pi^{(0)}$ نیز، یک سیاست بهینه برای این مسأله است و عاملی که از این سیاست تبعیت کند، از هر حالت، در کم‌ترین تعداد حرکت به یکی از خانه‌های هدف خواهد رسید. سیستم دینامیکی معادل با ارزیابی سیاست $\pi^{(p)}$، به صورت زیر خواهد بود:

$$\pi^{(p)} : v_{k+1}^{\pi^{(p)}} = \begin{bmatrix} 0 & p & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & p & 0 \end{bmatrix} v_k^{\pi^{(p)}} + \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} u_k \qquad (۱۵)$$

تابع تبدیل معادل با سیستم توصیف شده با معادله‌ی حالت فوق عبارت است از:

$$G(z) = \frac{1}{z^2 - p} \begin{bmatrix} -(z+p) \\ -(z+1) \\ -(z+p) \end{bmatrix} \qquad (۱۶)$$

قطب‌های سیستم فوق $+\sqrt{p}$ و $-\sqrt{p}$ هستند. هنگامی که $p = 0$ باشد، یعنی عامل از سیاست $\pi^{(0)}$ تبعیت کند، تمام قطب‌های سیستم فوق در مبدأ صفحه‌ی z قرار می‌گیرند. چنین سیستمی در مبحث کنترل دیجیتال، به نام سیستم مرده نَوش[1] شناخته می‌شود. یک سیستم مرده نَوشِ درجه n، در مقایسه با سایر سیستم‌های درجه n، سریع‌ترین پاسخ ممکن را دارد و پاسخ پله‌ی آن، دقیقا در n واحد زمانی گسسته، به مقدار نهایی می‌رسد [27]. از معادله‌ی حالت (۱۵) بر می‌آید که ورودی سیستم مورد بررسی، سیگنال u_k و خروجی آن v_k^{π} است. اگر رابطه‌ی بین ورودی و خروجی را با استفاده از تبدیل z بنویسیم، خواهیم داشت:

$$V(z) = G(z)U(z) \qquad (۱۷)$$

که در آن $V(z)$ و $U(z)$، به ترتیب تبدیل z سیگنال‌های v_k^{π} و u_k هستند. مقدار نهایی v_k^{π}، جواب الگوریتم ارزیابی سیاست است. طبق قضیه‌ی مقدار نهایی برای سیگنال‌های گسسته، مقدار نهایی v_k^{π} به صورت زیر قابل محاسبه است:

$$\begin{aligned} v^{\pi} &= \lim_{k\to\infty} v_k^{\pi} = \lim_{z\to 1} (1 - z^{-1}) V(z) \\ &= \lim_{z\to 1} (1 - z^{-1}) G(z) U(z) \end{aligned} \qquad (۱۸)$$

با توجه به این که سیگنال u_k، همواره برابر با پله‌ی واحد در نظر گرفته می‌شود، داریم:

$$v^{\pi} = \lim_{z\to 1} (1 - z^{-1}) G(z) \frac{1}{1 - z^{-1}} = \lim_{z\to 1} G(z) = G(1) \qquad (۱۹)$$

لذا برای سیستم توصیف شده با معادله‌ی حالت (۱۵)، که معادل با ارزیابی سیاست $\pi^{(p)}$ است، مقدار نهایی متغیرهای حالت به صورت زیر قابل محاسبه هستند:

$$v^{\pi} = \begin{bmatrix} -\frac{1+p}{1-p} & -\frac{2}{1-p} & -\frac{1+p}{1-p} \end{bmatrix}^T \qquad (۲۰)$$

این مقادیر نهایی، نشان دهنده‌ی متوسط هزینه‌ای هستند که عامل با شروع از هر یک از حالات، برای رسیدن به خانه‌های هدف، می‌پردازد. کمترین مقدار هزینه‌ای که پرداخت می‌شود، به ازای $p = 0$ به دست می‌آید. سیاست معادل با این مقدار، $\pi^{(0)}$ است که یک سیاست بهینه برای این مسأله است.

در محاسبات انجام شده، مقدار ضریب تنزیل γ برابر با یک در نظر گرفته شده است. اگر γ را در محاسبات وارد کنیم، معادله‌ی حالت به دست آمده، به صورت زیر خواهد بود:

$$\pi^{(p)} : v_{k+1}^{\pi^{(p)}} = \gamma \begin{bmatrix} 0 & p & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & p & 0 \end{bmatrix} v_k^{\pi^{(p)}} + \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} u_k \qquad (۲۱)$$

شرط لازم و کافی برای همگرایی الگوریتم ارزیابی سیاست، پایداری سیستم فوق است. این سیستم در صورتی پایدار است که همه‌ی مقادیر ویژه‌ی ماتریس حالت آن در داخل دایره‌ی واحد قرار بگیرند [27]. مقادیر ویژه‌ی ماتریس حالت سیستم توصیف شده با معادله‌ی حالت (۲۲)، عبارتند از:

$$\lambda_1 = 0 \qquad \lambda_{2,3} = \pm\gamma\sqrt{p} \qquad (۲۲)$$

لذا شرط پایداری این سیستم و همچنین همگرایی الگوریتم ارزیابی سیاست $\pi^{(p)}$ به صورت زیر است:

$$\gamma\sqrt{p} < 1 \quad \Rightarrow \quad \gamma < \frac{1}{\sqrt{p}} \qquad (۲۳)$$

شرط فوق بیان می‌کند که اگر سیاست مورد ارزیابی $\pi^{(0)}$ باشد، الگوریتم به ازای تمام مقادیر γ همگرا خواهد بود.

۵-۲- مسأله‌ی دوم

جدولی به صورت شکل ۳ را در نظر بگیرید. در این مسأله نیز، عامل در یکی از خانه‌های سفید رنگ جدول قرار دارد، و می‌بایست با حرکت در یکی از چهار جهتِ بالا، پایین، چپ و راست، خود را به یکی از دو خانه‌ی هدف، که با رنگ خاکستری مشخص شده‌اند، برساند. حرکت

[1] Dead Beat

در هر جهت، پاداشی به اندازه‌ی ۱- در پی دارد، که این پاداش، نشان دهنده‌ی هزینه‌ای است که عامل برای هر حرکت می‌پردازد. حرکت‌هایی که باعث خارج شدن عامل از جدول می‌شوند، بر موقعیت عامل تاثیری ندارند و محل عامل را تغییر نمی‌دهند. عامل باید یاد بگیرد که با دریافت بیشترین پاداش (پرداخت کمترین جریمه)، خود را به یکی از خانه‌های هدف برساند. اگر چنین کاری محقق شود، عامل توانسته است با کمترین تعداد حرکت، به هدف برسد. این مسأله به صورت یک فرآیند تصمیم‌گیری مارکوف، قابل بیان می‌باشد و می‌توان برای حل آن، از روش ارزیابی سیاست استفاده کرد. برای استفاده از روش ارزیابی سیاست، ارزش اولیه‌ی هر کدام از خانه‌ها، برابر با صفر در نظر گرفته می‌شود [1,5,7,14]. طبق قضیه‌ی نقطه‌ی ثابت، نتیجه‌ی نهایی، مستقل از ارزش اولیه‌ی خانه‌ها می‌باشد و الگوریتم همواره به یک نقطه‌ی منحصر به فرد در فضای جستجو، همگرا می‌شود [17].

s^0	s^1	s^2	s^3
s^4	s^5	s^6	s^7
s^8	s^9	s^{10}	s^{11}
s^{12}	s^{13}	s^{14}	s^0

شکل ۳ – جدول مربوط به مسأله‌ی مورد بررسی در بخش ۵-۲

سیاستی که تاپایان حل مسأله مورد استفاده قرار گرفته است، سیاست تصادفی است. به این معنی که، در همه‌ی خانه‌های جدول، احتمال حرکت به تمام جهات، مساوی و همگی برابر با یک چهارم یا ۰/۲۵ می‌باشد. در شکل ۴، چند مرحله از حل تکراری معادله‌ی بلمن، توسط معادله‌ی (۷)، نشان داده شده است. با استفاده از نتایج مربوط به هر مرحله، می‌توان سیاستی را پایه‌ریزی کرد. به این ترتیب که، عامل می‌بایست در هر خانه از جدول، به سمت خانه‌هایی حرکت کند که بیشترین ارزش را دارند. سیاستی که با استفاده از ارزش‌های به دست آمده در مرحله‌ی k اُم به دست می‌آید، به صورت π_k نشان داده شده است. π_0 همان سیاست تصادفی است. π_∞ نیز سیاستی است که با استفاده از ارزش‌های نهایی به دست می‌آید. π_3 و تمام سیاست‌های بعد از آن، همگی معادل هستند و $\pi_3 = \pi_\infty$ می‌باشد. فرض کنید با استفاده از هر کدام از سیاست‌های به دست آمده، معادله‌ی سیستم معرفی شده در معادله‌ی (۱۰) محاسبه شوند، و ماتریس $\mathcal{P}$ در معادله‌ی (۱۰) برای سیاست π_i به صورت $\mathcal{P}_i$ باشد. استفاده از اندیس i، صرفا به دلیل جلوگیری از تداخل اندیس‌ها در معادله‌ی (۱۰) می‌باشد. شعاع طیفی هر کدام از ماتریس‌های مذکور محاسبه شده‌اند و عبارتند از:

$$\begin{aligned} &\rho(\mathcal{P}_0) \simeq 0.9468, \quad \rho(\mathcal{P}_1) \simeq 0.8431, \\ &\rho(\mathcal{P}_2) = 0.5, \qquad \rho(\mathcal{P}_3) = \rho(\mathcal{P}_\infty) \simeq 0 \end{aligned} \tag{۲۳}$$

به وضوح دیده می‌شود که هر چه قدر سیاست به کار رفته در ایجاد مدل، بهینه‌تر باشد، اندازه‌ی بزرگترین مقدار ویژه‌ی ماتریس حالت نیز کوچک‌تر می‌شود. به خصوص به ازای سیاست π_3 یا همان π_∞، تمامی مقادیر ویژه برابر با صفر هستند. به این ترتیب، حد بالای ضریب تنزیل γ، برای همگرایی سری (۵)، به ازای سیاست‌های π_0 تا π_3 به ترتیب عبارت است از: ۱/۰۵۶۲، ۱/۱۸۶۱، ۲ و ∞. به عبارت دیگر، سری تعریف شده با معادله‌ی (۵)، به شرط پیروی از سیاست $\pi_3 = \pi_\infty$، به ازای تمام مقادیر γ، همگرا خواهد بود.

از طرفی، قطب‌های سیستم (۱۰)، با مقادیر ویژه‌ی ماتریس $\gamma\mathcal{P}$ برابر هستند. مشاهده می‌شود که قطب‌های سیستم معادل با سیاست π_∞، همگی در مبدا قرار دارند. به عبارت دیگر، این سیستم نیز یک سیستم مرده نَوِش است. هدف از حل مسأله‌ی فوق نیز، رسیدن به یکی از خانه‌های هدف در کمترین تعداد حرکت می‌باشد. لذا کاملا طبیعی است که پاسخ بهینه، متناظر با یک سیستم مرده نَوِش باشد، که سریع‌ترین پاسخ را در بین سیستم‌های هم‌درجه‌اش دارد. می‌توان استدلال کرد که، الگوریتم ارزیابی سیاست برای π_∞، حد اکثر در ۱۴ تکرار همگرا می‌شود و پس از آن، هیچ تغییری در ارزش حالات ایجاد نخواهد شد.

با تعریف ارزش همه‌ی حالات به عنوان خروجی، می‌توان تابع تبدیل این سیستم‌ها را به صورت زیر به دست آورد:

$$G_i(z) = (zI - \gamma\mathcal{P}_i)^{-1}\mathcal{R}_i \tag{۲۵}$$

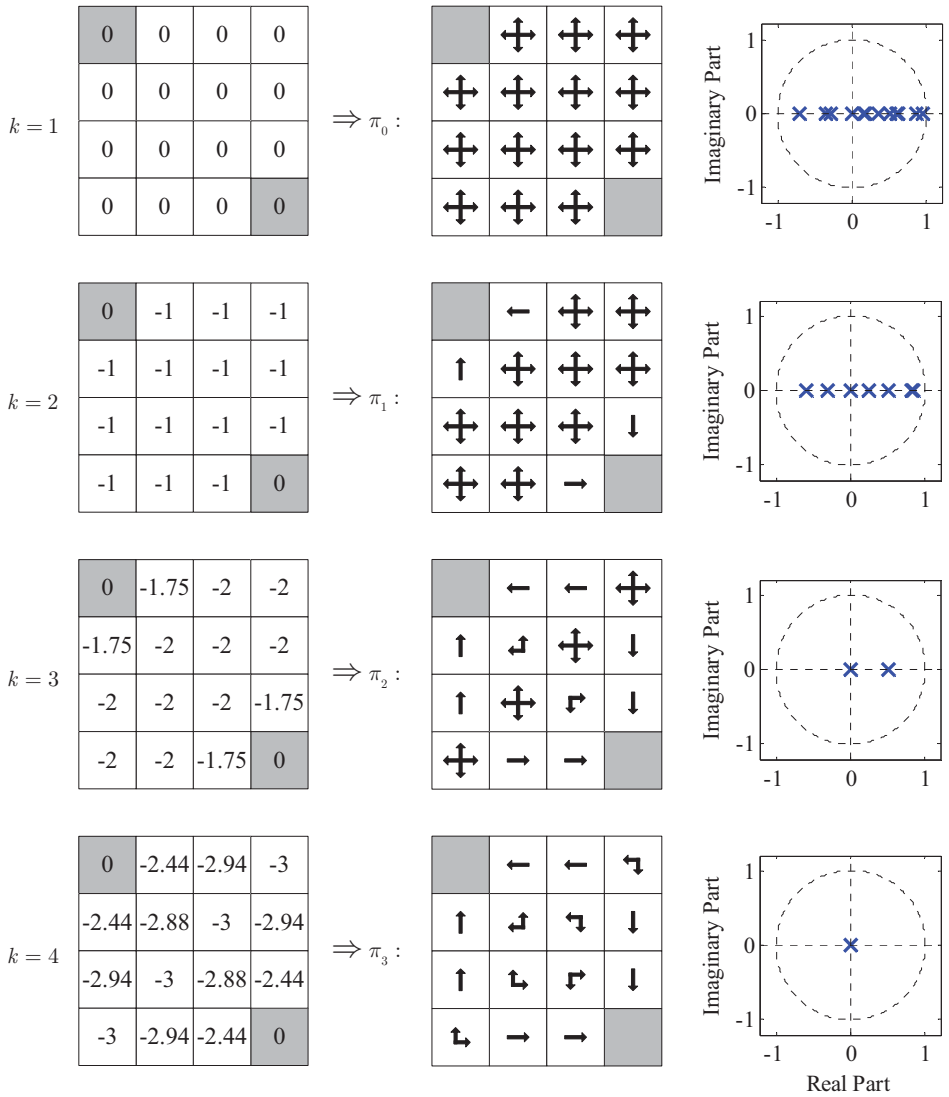
تابع تبدیل فوق، متناظر با سیستمی با یک ورودی و ۱۴ خروجی می‌باشد. هر کدام از خروجی‌ها، متناظر با ارزش یکی از خانه‌های جدول مربوط به مسأله‌ی مورد بررسی می‌باشند.

به عنوان نمونه، پاسخ فرکانسی هر یک از سیستم‌ها را به ازای s^3 در شکل ۵ مشاهده می‌کنید. با توجه به تقارن موجود در مسأله، این پاسخ فرکانسی، مربوط به خانه‌ی s^{12} نیز می‌باشد. توجه کنید که درجه‌بندی محور عمودی، به صورت دِسی بِل (dB) انتخاب نشده است و مقادیر نشان داده شده، مقادیر واقعی هستند.

با توجه به این که در کنترل دیجیتال، رابطه‌ی فرکانس موهومی ω با فرکانس مختلط z به صورت $z = e^{j\omega T_s}$ است، در شکل ۵، فرکانس موهومی ω به بازه‌ی $[0,\pi]$ محدود شده است. منظور از T_s، زمان نمونه‌برداری سیستم دیجیتال است. توجه نمایید که پاسخ فرکانسی سیستم، که تبدیل فوریه‌ی گسسته از سیگنالی گسسته است، یک سیگنال متناوب و پیوسته است و دوره‌ی تناوب آن 2π می‌باشد و به دلیل حقیقی بودن سیستم، تابعی زوج بر حسب ω می‌باشد [27].

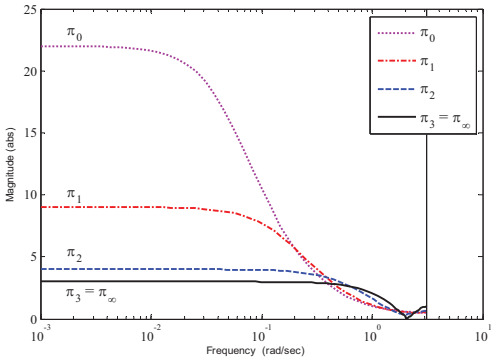
پاسخ فرکانسی نشان داده شده در شکل ۵، حاوی اطلاعات مهمی در مورد محیط و سیاست به کار رفته از طرف عامل، می‌باشد. در این شکل، عملکرد حالت ماندگار سیستم در خانه‌ی s^3، با توجه به پاسخ فرکانسی، قابل مشاهده است. عملکرد حالت ماندگار این سیستم‌ها، متناظر با مقدار پاسخ فرکانسی در فرکانس صفر است. دیده می‌شود که

دامنه‌ی پاسخ فرکانسی در فرکانس صفر، به ازای سیاست‌های π_0، π_1، π_2 و π_3 به ترتیب برابر با ۲۲، ۹، ۴ و ۳ است.



شکل ۴ – مراحل الگوریتم ارزیابی سیاست برای π_0 و سیاست‌های استخراج شده از اطلاعات هر مرحله به همراه نمودار قطب‌های سیستم دینامیکی معادل با ارزیابی هر کدام از سیاست‌ها

این مقادیر، متوسط هزینه‌ای هستند که عامل برای رسیدن به هر یک از خانه‌های هدف و با شروع از s^3 می‌پردازد. با توجه به تعریف تابع پاداش برای این مسأله، مقدار جریمه برابر با تعداد حرکت‌هایی است که عامل برای رسیدن به خانه‌های هدف، انجام می‌دهد. مشاهده می‌شود که به ازای π_3، کمترین تعداد حرکت‌ها به دست آمده است. در شکل ۵ مشاهده می‌شود که سیستم‌های معادل با سیاست‌های بهتر، پهنای باند وسیع‌تری دارند. با توجه به این که پهنای باند، معیاری از سرعت پاسخ‌دهی هر سیستم است، می‌توان استدلال کرد که، برای سیاست‌های بهتر، سرعت پاسخ‌دهی سیستم‌ها بیشتر است. این نکته همان چیزی است که در مورد مسأله‌ی حاضر، از یک سیاست خوب انتظار داریم.



شکل ۵ – پاسخ فرکانسی سیستم‌های تبعیت کننده از سیاست‌های π_0 تا π_3 در خانه‌ی s^3 از جدول نشان داده شده در شکل

۶- بررسی جامع مسائل جدولی

نتایج به دست آمده در این بخش، به مسائل جدولی و یا مسائل مشابه مربوط می‌شوند. خواص مشترک مسائلی که نتیجه‌گیری‌های این بخش در مورد آن‌ها صدق می‌کنند، عبارتند از:

- می‌بایست مسأله در قالب فرآیند تصمیم‌گیری مارکوف قابل توصیف باشد.
- یک یا چندین حالت نهایی[1] وجود داشته باشند که با رسیدن به آن حالات، وظیفه‌ی عامل تمام می‌شود.
- تمام حرکات، هزینه‌ای به صورت پاداش منفی داشته باشند.
- هدف از حل مسأله، یافتن شیوه‌ای برای رسیدن به حالت نهایی است که مستلزم پرداخت کمترین هزینه، و یا اخذ کمترین میزان پاداش منفی، باشد. هزینه‌ی کل به صورت مجموع تنزیلی پاداش‌های دریافت شده توسط عامل، تعریف می‌شود.

یک مسأله‌ی جدولی را، با یک یا چند خانه‌ی هدف در نظر بگیرید. از نظر مدل مارکوفی، تمام خانه‌های هدف، به عنوان یک حالت واحد در نظر گرفته می‌شوند. ارزش حالتی که متناظر با خانه‌های هدف است، همواره برابر با صفر است. لذا جهت اختصار، ارزش این حالت از بردار ارزش حالت حذف گردیده است.

بدون کاسته شدن از کلیت مسأله، فرض شده است که، هر حرکتی، هزینه‌ای دارد که به صورت پاداشی به اندازه‌ی ۱- مدل شده است. همچنین ضریب تنزیل به صورت $\gamma = 1$ در نظر گرفته شده است. هر یک از خانه‌های عادی جدول، معادل با یک حالت مانند s هستند. برای هر حالت s، قطعا می‌توان مسیری به سمت یکی از خانه‌های هدف پیدا کرد که مستلزم دریافت کمترین جریمه باشد. این مسیر دارای کمترین تعداد حرکت ممکن است و برای حالت s، این تعداد حرکت با $m(s)$ نمایش داده می‌شود. با در نظر گرفتن فرض‌های یاد شده، می‌توان قضیه‌ای را در خصوص مسائل جدولی و به صورت زیر بیان نمود.

قضیه. سیستم دینامیکی معادل با ارزیابی سیاست برای سیاست بهینه‌ی یک مسأله‌ی جدولی، دارای خواص زیر است:

- یک سیستم مرده نَوِش است و مولفه‌ای از آن که متناظر با حالت s است، دقیقا $m(s)$ قطب دارد که همگی در مبدأ صفحه‌ی z قرار دارند.
- هنگامی که $\gamma = 1$ اختیار می‌شود، صفرهای مولفه‌ی مربوط به حالت s، به همراه نقطه‌ی $z = 1$، همگی ریشه‌های $m(s)$اُم واحد هستند و محیط دایره‌ی واحد را به $m(s)$ قسمت مساوی تقسیم می‌کنند.
- این سیستم در مقایسه با سیستم‌های معادل با ارزیابی سیاست‌های دیگر، سریع‌ترین پاسخ ممکن را دارد.
- این سیستم به ازای تمامی مقادیر ضریب تنزیل γ، پایدار است. به عبارت دیگر، الگوریتم ارزیابی سیاست برای سیاست بهینه‌ی یک مسأله‌ی جدولی، مستقل از مقدار γ، همواره همگرا است.
- اندازه‌ی پاسخ فرکانسی در فرکانس موهومی $\omega = 0$، برای این سیستم، کمترین مقدار ممکن را دارد. در صورتی که $\gamma = 1$ باشد، اندازه‌ی پاسخ فرکانسی متناظر با حالت s، در فرکانس $\omega = 0$ یا $z = 1$، برابر با $m(s)$ خواهد بود. این مقدار برابر با قرینه‌ی ارزش نهایی حالت s است.

□

اثبات. اگر عاملی از سیاست بهینه تبعیت کند، با شروع از حالت s، مسیر بهینه‌ای را به سمت خانه‌های هدف طی خواهد کرد، دقیقا با انجام $m(s)$ حرکت به خانه‌ی هدف خواهد رسید، و نهایتا پاداشی به اندازه‌ی $-m(s)$ دریافت خواهد نمود.

اگر $\gamma = 1$ در نظر گرفته شود، ارزش خانه‌ی s، پس از انجام دادن حرکت kاُم، برابر با مجموع پاداش‌هایی است که تا زمان انجام حرکت kاُم، توسط عامل دریافت شده است. به این ترتیب، ارزش حالت s، به شرطی که عامل از سیاست بهینه تبعیت کند، به صورت تابعی از زمان گسسته قابل تعریف است:

$$v_s^*[k] = \begin{cases} 0 & ,k < 0 \\ -k & ,0 \le k \le m(s) \\ -m(s) & ,k > m(s) \end{cases} \qquad (۲۶)$$

توجه کنید که زمان در تعریف فوق، متناظر با شماره‌ی تکرار در الگوریتم ارزیابی سیاست می‌باشد. تابع فوق، یکی از خروجی‌های سیستم معادل با ارزیابی سیاست بهینه می‌باشد. تبدیل z تابع ارزش فوق، عبارت است از:

$$\mathcal{Z}\left\{v_s^*[k]\right\} = V_s^*(z) = -\sum_{k=1}^{m(s)} kz^{-k} - m(s)\sum_{k=m(s)+1}^{\infty} z^{-k} \qquad (۲۷)$$

از طرفی، طبق مطالب مطرح شده، ورودی سیستم معادل با ارزیابی سیاست، همواره برابر با پله‌ی واحد اختیار می‌شود. لذا تابع تبدیل متناظر با ارزش حالت s، با توجه به رابطه‌ی زیر قابل محاسبه است:

$$V_s^*(z) = G_s^*(z)U(z) = \frac{G_s^*(z)}{1 - z^{-1}} \qquad (۲۸)$$

لذا تابع تبدیل متناظر با حالت s، یعنی $G_s^*(z)$، عبارت است از:

$$G_s^*(z) = (1 - z^{-1})V_s^*(z) \qquad (۲۹)$$

با جایگذاری عبارت به دست آمده برای $V_s^*(z)$ در رابطه‌ی اخیر، $G_s^*(z)$ به صورت زیر به دست خواهد آمد:

$$\begin{aligned} G_s^*(z) &= -\sum_{k=1}^{m(s)} z^{-k} = -\frac{z^{m(s)-1} + \cdots + z + 1}{z^{m(s)}} \\ &= -\frac{z^{m(s)} - 1}{z^{m(s)}(z-1)} \end{aligned} \qquad (۳۰)$$

[1] Terminal State

مشاهده می‌شود که، تابع تبدیل متناظر با حالت s، دقیقا $m(s)$ قطب دارد که همگی در مبدأ صفحه‌ی z قرار دارند. به عبارت دیگر، $G_s^*(z)$ همواره یک سیستم مرده نَوِش است. همچنین صفرهای تابع تبدیل $G_s^*(z)$، همواره بر روی دایره‌ی واحد قرار دارند و همگی ریشه‌های $m(s)$اُم واحد هستند. توجه کنید که $z=1$، صفر تابع تبدیل $G_s^*(z)$ نمی‌باشد. نقطه‌ی $z=1$ و صفرهای تابع تبدیل $G_s^*(z)$، محیط دایره‌ی واحد را به $m(s)$ قسمت مساوی تقسیم می‌کنند. مطالب مطرح شده، مربوط به یک مولفه‌ی اختیاری از تابع تبدیل سیستم معادل با ارزیابی سیاست بهینه بودند. همه‌ی مولفه‌های این تابع تبدیل، مرده نَوِش هستند و بنابراین تمام قطب‌های این تابع تبدیل در مبدأ صفحه‌ی z قرار دارند. با توجه به نظریه‌ی کنترل سیستم‌های گسسته-زمان، سیستم مرده-نوش، سریع‌ترین پاسخ ممکن را در بین تمام سیستم‌های هم‌درجه دارد [27]. همچنین با توجه به این که قطب‌های سیستم در مبدأ صفحه‌ی z قرار دارند، و با توجه به نحوه‌ی تاثیر ضریب γ بر روی محل قطب‌ها در رابطه‌ی (۱۳)، می‌توان استدلال کرد که γ تاثیری بر محل قطب‌های سیستم ندارد و از این رو، سیستم تعریف شده با (۳۰)، مستقل از مقدار γ، همواره پایدار است.

اگر عامل یادگیرنده از سیاست بهینه پیروی نکند، مقدار تابع ارزش آن در تمام لحظات، قطعا کمتر از یا مساوی با مقدار تابع ارزش مربوط به سیاست بهینه خواهد بود. فرض کنید سیاست π، نشان دهنده‌ی سیاستی باشد که در تمام حالت‌ها دقیقا مثل سیاست بهینه باشد و فقط در یک حالت به خصوص مانند s'، متفاوت از سیاست بهینه باشد. همچنین فرض کنید، عاملی که از این سیاست تبعیت می‌کند، با شروع از حالت s، پس از k_0 حرکت، به s' می‌رسد. در این حالت می‌توان تابع ارزش حالت s را در حرکت kاُم و به شرط تبعیت از سیاست π، به صورت زیر تعریف نمود:

$$v_s^\pi[k]=\begin{cases}0 & ,k<0\\ -k & ,0\le k<k_0\\ -k-\sum_{i=k_0}^{k}\delta_i & ,k\ge k_0\end{cases} \qquad (۳۱)$$

که در آن، δ_i مقداری از جریمه‌ی اضافه‌ای است که در اثر عدم تبعیت از سیاست بهینه، در حرکت k_0 دریافت می‌شود. اگر تبدیل z مربوط به این تابع ارزش محاسبه شود و طبق رابطه‌ی (۲۹)، تابع تبدیل سیستم محاسبه شود، خواهیم داشت:

$$G_s^\pi(z)=-\sum_{k=1}^{k_0-1}z^{-k}-\sum_{k=k_0}^{\infty}(1+\delta_k)z^{-k} \qquad (۳۲)$$

سیستم فوق فقط به شرطی می‌تواند مرده-نوش باشد که بتوان k_1 را پیدا کرد، به نحوی که:

$$\forall k>k_1,\ \ 1+\delta_k=0 \qquad (۳۳)$$

در غیر این صورت سیستم مرده-نوش نیست و کندتر از سیستم (۳۰) عمل خواهد کرد. البته درجه‌ی توابع تبدیل (۳۲) و (۳۰) نیز یکسان نخواهد بود و حتی در صورت مرده-نوش بودن (۳۲)، باز هم سیستم (۳۰) دارای سرعت پاسخ‌دهی بیشتری خواهد بود.

برای سیستم (۳۲)، اندازه‌ی پاسخ فرکانسی در $\omega=0$ یا $z=1$ بزرگتر از سیستم (۳۰) است و داریم:

$$\begin{aligned}&G_s^\pi(1)=-(k_0-1)-\sum_{k=k_0}^{\infty}(1+\delta_k)\\ &G_s^\pi(1)\le -m(s)-\sum_{k=k_0}^{\infty}\delta_k\le -m(s)=G_s^*(1)\\ &\Rightarrow \left|G_s^\pi(1)\right|\ge\left|G_s^*(1)\right|\end{aligned} \qquad (۳۴)$$

■

۷- نتیجه‌گیری

در این مقاله با ارائه‌ی مروری بر یادگیری تقویتی، فرآیندهای مارکوف و برنامه‌ریزی پویا، معادلات مربوط به حل یک فرآیند مارکوف با استفاده از برنامه‌ریزی پویا، به صورت یک دینامیک گسسته-زمان جمع‌بندی و دوباره‌نویسی شدند. این روش برخورد، امکان بررسی فرآیند حل یک مسأله‌ی یادگیری تقویتی در محیط مارکوف را، در قالب یک سیستم دیجیتال فراهم می‌آورد. به این ترتیب، می‌توان شیوه‌های مرسوم در کنترل دیجیتال را برای تحلیل یک فرآیند یادگیری استفاده نمود. موضوع بحث این نوشتار، بر روی مسائلی موسوم به مسائل جدولی می‌باشد. نتایج حاکی از آن هستند که یک سیاست بهینه برای این نوع از مسائل، در قالب کنترل دیجیتال به صورت یک سیستم مرده نَوِش قابل توصیف می‌باشد. تعمیم این نتیجه به انواع دیگر مسائل و تعریف دوگانی بین فضای تصمیم‌گیری و فضای سیستم‌های کنترل دیجیتال، از مطالعات و تحقیقات تکمیلی در ادامه‌ی این مقاله هستند.

مراجع

[1] M. L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, Wiley, 2005.
[2] A. Cassandra, "Exact and Approximate Algorithms for Partially Observable Markov Decision Processes," as Ph.D. Thesis, Brown University, 1998.
[3] L. Pyeatt, "Integration of Partially Observable Markov Decision Processes and reinforcement Learning for Simulated Robot Navigation," as Ph.D. Thesis, Colorado State University, 1999.
[4] D. P. Bertsekas and J. N. Tsitsiklis, Neuro-Dynamic Programming, Athena Scientific, 1996.

[5] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, The MIT Press, 1998.
[6] A. Lew and H. Mauch, Dynamic Programming: A Computational Tool, Springer-Verlag, Berlin, 2007.
[7] S. I. Reynolds, "Reinforcement Learning with Exploration," as Ph.D. Thesis, School of Computer Science, The University of Birmingham, UK, 2002.
[8] B. Van Roy, "Neuro-Dynamic Programming: Overview and Recent Trends," chapter of, E. A. Feinberg and A. Schwartz, Handbook of Markov Decision Processes: Methods and Applications, Kluwer Academic, 2002.
[9] J. Si et al., Handbook of Learning and Approximate Dynamic Programming, Wiley Inter-Science, 2004.
[10] Hyeong Soo Chang et al, "A survey of some Simulation-Based Algorithms for Markov Decision Processes," in Communications in Information and Systems, Vol. 7, No. 1, pp. 59-92, 2007.
[11] J. E. Smith and K. F. Mc Cardle, "Structural Properties of Stochastic Dynamic Programs," in Operations Research, Vol. 50, pp. 796-809, 2002.
[12] M. C. Fu et al., "Monotone optimal policies for queuing staffing problem," in Operations Research, Vol. 46, pp. 327-331, 2000.
[13] R. Givan et al. "Bounded Markov Decision Processes," in Artificial Intelligence, Vol. 122, pp. 71-109, 2000.
[14] L. P. Kaelbling, M. L. Littman and A. W. Moore, "Reinforcement Learning: A Survey," Journal of Artificial Intelligence Research, Vol. 4, pp. 237-285, 1996.
[15] G. J. Gordon, "Approximate Solution to Markov Decision Processes," Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, 1999.
[16] D. P. de Farias and B. Van Roy, "On the Existance of Fixed Points for Approximate Value Iteration and Temporal-Difference Learning," in Journal of Optimization theory and Applications, Vol. 105, No. 3, pp. 589-608, 2000.
[17] H. Royden, Real Analysis (3rd Edition), Prentice Hall, 1988.
[18] Qiying Hu and Wuyi Yue, Markov Decision Processes with Their Applications, Springer Science+Busines Media, LLC, 2008.
[19] Heyeong Soo Chang et al., Simulation-based Algorithms for Markov Decision Processes, Springer-verlag, London, 2007.
[20] F. Fernandez and M. Veloso, "Exploration and Policy Reuse," as Technical Report, School of Computer Science, Carnegie Mellon University, 2005.
[21] F. Fernandez and M. Veloso, "Probabilistic Reuse of Past policies," as Technical Report, School of Computer Science, Carnegie Mellon University, 2005.
[22] F. Fernandez and M. Veloso, "Building a Library of Policies through Policy Reuse," as Technical Report, School of Computer Science, Carnegie Mellon University, 2005.
[23] D. S. Bernstein, "Reusing Old Policies to Accelerate Learning on New Markov Decision Processes," as Technical Report, Department of Computer Science, University of Massachusetts, Amherst Tech. Rep. No. 99-26, 1999.
[24] N. L. Zhang and W. Zhang, "Speeding Up the Convergence of Value Iteration in Partially Observable Markov Decision Processes," in Journal of Artificial Intelligence Research, Vol. 14, pp. 29-51, 2001.
[25] E. A. Hansen, "An Improved Policy Iteration for Partially Observable Markov Decision Processes," in Proceedings of 10th Neural Information Processing Systems Conference, 1997.
[26] B. Sallans, "Reinforcement Learning for Factored Markov Decision Processes," as Ph.D. Thesis, Graduate Department of Computer Science, University of Toronto, 2002.
[27] K. Ogata, Discrete-Time Control Systems (2nd Edition), Prentice Hall, 1994.