Tabriz Journal of Electrical Engineering (TJEE), vol. 55, no. 2, Summer 2025 Serial no. 112
DOI: 10.22034/tjee.2024.59859.4790

An Actor-Critic Deep Reinforcement Learning Framework for Multi-objective Sequential Decision-making

Mohammad Amir Rezaei Gazik, Mehdy Roayaei*,

Electrical and Computer Engineering Department, Tarbiat Modares University, Tehran, Iran
E-mails: m_rezaeigazik@modares.ac.ir; mroayaei@modares.ac.ir

Abstract
Sequential decision making describes a situation where the decision maker makes successive observations of a process before a final decision is made. In real-world scenarios, multi-objective sequential decision-making problems have been common and pose multiple challenges for researchers in decision-making. Most studies in this area have traditionally focused on single-objective situations or converted multi-objective problems into single-objective ones by combining objectives into a single goal. In this article, a multi-objective deep reinforcement learning framework called "MACA," based on the actor-critic method is presented, to optimize and balance multiple conflicting objectives in dynamic environments over time. This framework learns different policies for various objectives and eventually converges them to a global optimal policy. This framework, is evaluated in the domain of recommender systems for two conflicting objectives: accuracy (the desirability of recommended items for users) and fairness (the selection of recommended items from all categories); and, compared with other recent multi-objective reinforcement learning methods. Experimental results on the benchmark problem (recommender systems) demonstrate that this framework outperforms previous works in terms of performance (the accuracy was 92.5% with a fairness score of 96.5% on the Kiva dataset, and 93.1% accuracy with a fairness score of 97.6% on the MovieLens dataset), convergence time, and memory consumption. Moreover, the proposed framework is scalable with respect to the number of objectives and enables optimization of the variable number of objectives.

Keywords
Deep reinforcement learning, Recommender system, Actor-Critic, Multi-objective decision making.

Introduction

Multi-objective optimization problems involve two or more conflicting objectives, meaning that improving one objective leads to a reduction in another. Multi-objective optimization has been applied in various domains, including recommender systems, economics, chemical engineering, and transportation. Although traditional learning approaches have mainly focused on single-objective settings, in reality, many problems have multiple conflicting objectives. Such problems can be modeled as "Multi-Objective Markov Decision Processes" (MOMDP) and can be solved using Reinforcement Learning. In these scenarios, the agent learns to make decisions that achieve a balance between the conflicting objectives, ultimately reaching a set of policies that represent the best compromise among the multiple goals.

Proposed Work and Methodology (including comprision, simulation/experimental results and discusion)

In the proposed method, which is based on the actor-critic approach, where a single policy is learned by default, we have modified the architecture to a multi-objective reinforcement learning framework. This architecture consists of N parallel actor-critic blocks managed by a global attention layer (where N is the number of objectives). In the proposed architecture, for each objective, there exists a local actor-critic block that tries to learn a policy to maximize its corresponding objective. Additionally, a global attention layer is present, providing unique feedback to the local blocks to manage them. In other words, the global attention layer receives the outputs from the local blocks, compares these outputs with each other, determines the performance of each local block concerning each objective, and provides separate feedback to each local block to update their policies based on this feedback. Moreover, in each local block, there is a local critic that provides separate rewards for all objectives to the agent of that block.
For evaluating the proposed framework and comparing it with previous works, we have chosen the Item Recommendation Problem as the benchmark problem. Two datasets (Kiva and MovieLens) containing feedback (including user acceptances or rejections and ratings given by users to various items) over different time periods are given. The objective is to find a policy for the recommender system to optimize recommendations to users regarding the given objectives (accuracy and fairness).
Results showed that the proposed method, due to concurrently learning different policies in parallel, has a good exploration capability in the policy space. The generated policy in both objectives outperforms previous multi-objective methods. Additionally, due to the parallelization of actor-critic blocks, it exhibits faster convergence speed.

Conclusion

In this article, we presented a framework based on deep reinforcement learning for optimizing multi-objective sequential decision-making problems. The proposed method extends the actor-critic approach and includes separate actor-critic components for learning optimal policies for various objectives. The framework ultimately returns a global optimal policy by converging the policies of different blocks. The presented method is generalizable to problems with any number of objectives. Results showed that learning multiple policies concurrently can improve the exploration capability and efficiency of the reinforcement learning algorithm.

یک چارچوب یادگیری تقویتی عمیق عامل-منتقد برای تصمیم‌گیری متوالی چند هدفه

محمد امیر رضایی گزیک
کارشناسی ارشد، گروه مهندسی کامپیوتر، دانشگاه تربیت مدرس، تهران، ایران
مهدی رعایائی
استادیار، گروه مهندسی کامپیوتر، دانشگاه تربیت مدرس، تهران، ایران

چکیده
تصمیم‌گیری متوالی، شرایطی را توصیف می‌کند که در آن تصمیم‌گیرنده قبل از اینکه تصمیم نهایی گرفته شود، به صورت پیاپی مشاهداتی از یک فرآیند انجام می‌دهد. در کاربردهای دنیای واقعی، مسائل تصمیم‌گیری متوالی چند هدفه رایج بوده و چالش‌های متعددی را برای پژوهشگران فراهم می‌کند. بیشتر پژوهش‌ها در این حوزه به طور سنتی بر روی موقعیت‌هایی با یک هدف تمرکز داشته‌اند و یا با ترکیب هدف‌ها به یک هدف واحد، مسئله چند هدفه را به مسئله یک هدفه تبدیل کرده‌اند. در این مقاله، یک چارچوب یادگیری تقویتی عمیق چند هدفه، "MACA" بر اساس روش عامل-منتقد ارائه شده است تا در محیط‌های پویا، هدف‌های متعارض چندگانه را در طول زمان بهینه کرده و تعادل بخشد. این چارچوب به ازای اهداف مختلف، سیاست‌های مختلفی را یاد گرفته و در نهایت این سیاست‌ها را به یک سیاست بهینه‌ی سراسری همگرا می‌کند. برای ارزیابی این چارچوب، روش پیشنهادی در مسئله‌ی سیستم‌های توصیه‌گر و برای دو هدف متناقض صحت تصمیم‌گیری (مورد پسند بودن اقلام توصیه شده برای کاربران) و انصاف (انتخاب شدن اقلام توصیه شده از همه‌ی دسته‌ها) پیاده‌سازی و با سایر روش‌های اخیر یادگیری تقویتی چند هدفه مقایسه شده است. نتایج آزمایشی روی مسئله‌ی محک (سامانه‌های توصیه‌گر) نشان می‌دهد که این چارچوب نسبت به کارهای قبلی نتایج بهتری از نظر عملکرد (صحت ۹۲.۵ و انصاف ۹۶.۵ در مجموعه داده Kiva و صحت ۹۳.۱ و انصاف ۹۷.۶ در مجموعه داده MovieLens)، زمان همگرایی و مصرف حافظه دارد. همچنین، چارچوب پیشنهادی نسبت به تعداد اهداف مقیاس‌پذیر بوده و بهینه‌سازی تعداد متغیر اهداف را امکان‌پذیر می‌کند.

کلمات کلیدی
یادگیری تقویتی عمیق، سیستم‌های توصیه‌گر، عامل-منتقد، تصمیم‌گیری متوالی چند هدفه.

نام نویسنده مسئول: مهدی رعایائی
ایمیل نویسنده مسئول: mroayaei@modares.ac.ir

تاریخ ارسال مقاله: ۱۴۰۲/۰۵/۰۳
تاریخ(های) اصلاح مقاله: ۱۴۰۲/۰۹/۲۳
تاریخ پذیرش مقاله: ۱۴۰۳/۰۲/۲۴

۱- مقدمه

تصمیم‌گیری متوالی[1]، شرایطی را توصیف می‌کند که در آن تصمیم‌گیرنده قبل از اینکه تصمیم نهایی گرفته شود، به صورت پیاپی مشاهداتی از یک فرآیند انجام می‌دهد. رویکرد متداول برای حل چنین مسائلی، استفاده از یادگیری تقویتی[2] است. یادگیری تقویتی نوعی از یادگیری ماشینی است که در آن عامل (Agent) با تعامل با محیط، تصمیم‌گیری‌های خود را یاد می‌گیرد. عامل بر اساس اقدام‌های (Action) خود بازخوردی از محیط در قالب پاداش (Reward) دریافت می‌کند که اجازه می‌دهد تصمیم‌گیری‌های خود را در طول زمان بهبود دهد. هدف از یادگیری تقویتی یافتن سیاستی (Policy) بهینه است که منجر به حداکثر کردن پاداش کلی کسب‌شده توسط عامل می‌شود.

مسائل بهینه‌سازی چند هدفه شامل دو یا چند هدف بهینه‌سازی هستند که با یکدیگر در تضادند؛ به این معنی که بهبود یک هدف منجر به کاهش هدف دیگر می‌شود. بهینه‌سازی تصمیم‌گیری متوالی چند هدفه در حوزه‌های مختلفی از جمله سامانه‌های توصیه‌گر، اقتصاد، مهندسی شیمی و حمل و نقل به کار گرفته شده است. بنابراین، روش‌های بهینه‌سازی چند هدفه متعددی برای کنترل مسائل بهینه‌سازی چند هدفه پیشنهاد شده‌اند [۱-۴].

یک روش مرسوم برای مدیریت هدف‌های چندگانه، یکپارچه‌سازی آن‌ها در یک تابع هدف واحد است. با این حال، یک ترکیب وزن‌دار از اهداف فقط زمانی معتبر است که هدف‌ها با یکدیگر در تضاد نباشند. علاوه بر این، یک ترکیب از پیش تعیین شده در محیط‌های پویا انعطاف‌پذیری لازم را نخواهد داشت.

اگرچه بیشتر تحقیقات در حوزه یادگیری تقویتی شامل استفاده از یک مقدار پاداش عددی تک‌مولفه برای تعریف هدف عامل است، اما مطالعاتی نیز در استفاده از چند هدف در محیط‌های یادگیری تقویتی انجام شده است [۵-۷].

به همین ترتیب، مسائل تصمیم‌گیری متوالی با چند هدف به صورت طبیعی در دنیای واقعی پدیدار می‌شوند و چالش‌های منحصر به فردی را برای تحقیقات یادگیری تصمیم‌گیری ایجاد می‌کنند که عمدتاً برای حل آن‌ها بر روی تنظیمات تک‌هدفه تمرکز شده است [۸].

با وجود این که رویکردهای یادگیری عموماً بر تنظیمات تک‌هدفه تمرکز داشته‌اند، در واقعیت بسیاری از مسائل دارای هدف‌های چندگانه هستند که ممکن است با هم متعارض باشند. به عنوان مثال، عاملی که می‌خواهد عملکرد یک سرور برنامه وب را بیشینه کند، ممکن است تمایل داشته باشد همزمان مصرف انرژی را نیز کمینه سازد. چنین مسائلی می‌توانند به عنوان فرآیندهای

[1] Sequentai Decision Making

[2] Reinforcement learning

تصمیم‌گیری چند هدفه مارکوف[3] (MOMDP) مدل شوند و با استفاده از یادگیری تقویتی چند هدفه حل شوند [۸]. MOMDP ها در زمینه‌های مختلفی قابل مشاهده هستند، از جمله سیستم‌های توصیه‌گر [۹]، رانندگی شهری [۱۰]، کنترل ربات‌ها [۱۱] و کنترل ترافیک [۱۲].

الگوریتم‌های یادگیری تقویتی را می‌توان به دو دسته جدولی و تقریبی تقسیم کرد. الگوریتم‌های جدولی به دلیل استفاده از جدولی به عنوان حافظه برای ذخیره اطلاعات توابع ارزش (Value Function) نیاز به حافظه‌ی بالا دارند که این موضوع در محیط‌های با فضای حالت بزرگ ناکارآمد خواهد بود و عملی نیست. اگر چه ایده‌ی استفاده از یادگیری تقویتی برای تصمیم‌گیری متوالی جدید نیست و حدود دو دهه است که وجود دارد، اما استفاده از آن عمدتاً به دلیل مشکل مربوط به مقیاس‌پذیری الگوریتم‌های سنتی یادگیری تقویتی عملی نبوده است. از زمان معرفی یادگیری تقویتی عمیق [۱۳]، یک روند جدید در این حوزه ظهور کرده است که امکان استفاده از یادگیری تقویتی در مسائل تصمیم‌گیری با فضای حالت (State) و عمل بزرگ را فراهم کرده است. روش‌های یادگیری تقویتی عمیق به عنوان راهکارهایی برای رفع مشکل روش‌های سنتی ارائه شدند. در این روش‌ها که از شبکه‌های عصبی عمیق استفاده می‌شود، تنها نیاز به ذخیره‌ی شبکه عصبی و بازپخش تجربه[4] است [۱۴].

در مجموع، چالش‌های اصلی مسئله‌ی تصمیم‌گیری متوالی چند هدفه را می‌توان به شرح زیر خلاصه کرد:

- در مسائل چند هدفه، هدف‌ها با یکدیگر در تضاد هستند و نمی‌توان آن‌ها را همزمان بیشینه کرد. بنابراین، روش پیشنهادی باید تعادلی بین هدف‌های مختلف را به دست آورد.
- چون مسئله متوالی است، محیط ممکن است به طور مداوم تغییر کنن. بنابراین، روش پیشنهادی باید به طور پویا به به‌روزرسانی سیاست در طول زمان بپردازد.
- چون معمولاً اولویت هدف‌های مختلف نسبت به یکدیگر از پیش مشخص نیست، هیچ سیاست بهینه‌ی یکتایی وجود ندارد. بنابراین، تولید مجموعه‌ای از سیاست‌های بهینه مطلوب خواهد بود.

نوآوری‌های این مقاله را می‌توان به شرح زیر خلاصه کرد:

- مسئله توازن هدف‌های متعارض چند هدفه در یک مسئله‌ی تصمیم‌گیری متوالی به عنوان یک مسئله تصمیم‌گیری چند هدفه مارکوف فرموله شده است.
- یک چارچوب یادگیری تقویتی چند هدفه‌ی عمومی ارائه شده است که برای هر تعداد هدف قابل تعمیم است.
- چارچوب پیشنهادی برای محیط‌های پویا که به طور مداوم در تغییر است، طراحی شده است.
- روش عامل-منتقد به گونه‌ای تعمیم داده شده است که به صورت همزمان چندین سیاست را یاد می‌گیرد و آن‌ها را ترکیب می‌کند تا یک سیاست یکتا را که تمام هدف‌ها را در یک زمان به تعادل می‌رساند، به دست آورد.
- چارچوب پیشنهادی شامل چندین ماژول موازی است که به صورت همزمان اجرا می‌شوند و از این رو به لحاظ زمانی کارآمد است.
- چارچوب پیشنهادی در کاربرد واقعی ارزیابی شده تا برتری آن نشان داده شود.

قسمت باقی‌مانده این مقاله به شکل زیر سازماندهی شده است. در بخش ۶، کارهای مرتبط را ارائه خواهیم داد. بخش ۷ چارچوب پیشنهادی را ارائه می‌دهد. بخش ۸ نتایج ارزیابی را ارائه خواهد داد. در نهایت، بخش ۹ نتیجه‌گیری و کارهای آتی را ارائه خواهد کرد.

۲- کارهای مرتبط

با وجود رویکردهای مختلف در سیستم‌های توصیه‌گر [۱۶-۱۵] ، با توجه به این که تمرکز این مقاله بر روی مسائل تصمیم‌گیری چند هدفه با رویکرد یادگیری تقویتی است، در این بخش کارهای مرتبط با این حوزه را مورد بررسی قرار داده‌ایم.

لیو و همکاران [۱۷] یک روش مبتنی بر عامل-منتقد برای توازن دو هدف، انصاف (Fairness) و صحت (Accuracy)، در سیستم‌های توصیه‌گر پیشنهاد کردند. در روش پیشنهادی آن‌ها، یک عامل وجود دارد که براساس ترجیحات (preferences) کاربر و وضعیت معیار انصاف، توصیه‌هایی متغیر با زمان[5] را انجام می‌دهد، و همچنین یک منتقد وجود دارد که مقدار ارزش مرتبط با اقدام‌های عامل را تخمین می‌زند تا از این طریق به تشویق یا متوقف کردن توصیه اقلامی (item) خاص کمک کند.

ویرینگ و دو جونگ [۱۸] یک الگوریتم برنامه‌ریزی پویای چندهدفه مبتنی برای محاسبه سیاست‌های پارتو بهینه[6] برای مسائل تصمیم‌گیری متوالی چند هدفه قطعی (Deterministic) پیشنهاد کردند. برای مقابله با مشکل تابع وزن‌دهی اهداف که می‌تواند دلخواه باشد و پس از حل مسئله توسط عامل ارائه شود، روش پیشنهادی در طول اجرای خود، سیاست‌های پارتو-بهینه ایستا را ذخیره می‌کنند.

خمیس و گما [۱۹] مسئله کنترل ترافیک چند هدفه (که تابع چند هدف شامل کمینه کردن زمان انتظار سفر، کل زمان سفر و زمان انتظار در تقاطع است) را به عنوان یک سیستم چند هدفه مدلسازی کرده و یک روش یادگیری تقویتی چند هدفه برای حل آن پیشنهاد کردند.

یانگ و همکاران [۲۰] روشی چند هدفه مبتنی بر عامل-منتقد برای بهینه‌سازی پیشنهاد قیمت در سیستم‌های مناقصه بلادرنگ (Real-Time) ارائه دادند. در روش پیشنهادی آن‌ها، عامل‌ها بر اساس هدف متناظر خود به صورت ناهمروند (Asynchronous) یک شبکه سراسری (Global) را به‌روزرسانی می‌کنند تا به یک سیاست پایدار (Robust) برسند. مدل آن‌ها از چندین شبکه محلی عامل-منتقد با هدف‌های مختلف برای تعامل با یک محیط یکسان استفاده می‌کند و سپس براساس مقدار پاداش‌های مختلف شبکه سراسری را به‌روزرسانی می‌کند. اگرچه مدل آن‌ها از چندین بلوک عامل-منتقد استفاده می‌کند، اما یک سیاست سراسری در طول تعاملات در حال یادگیری است.

ریموند و همکاران [۲۱] یک مدل یادگیری تقویتی چند هدفه با استفاده از روش عامل-منتقد پیشنهاد کردند. در روش آن‌ها، منتقد یک توزیع چند متغیره روی بازه‌ها را یاد می‌گیرد که با پاداش‌های تجمعی ترکیب شده است و بر روی یک تابع سود غیرخطی عمل می‌کند.

نگوین و همکاران [۲۲] دو روش یادگیری تقویتی چند هدفه بر اساس شبکه عصبی عمیق DQN پیشنهاد کردند. در روش پیشنهادی آن‌ها، چندین نخ (Thread) برای امکان یادگیری عامل‌ها با سیاست‌های چندگانه به صورت موازی پیاده‌سازی شده‌اند، با این حال، سیاست‌های یادگرفته‌شده از هم مستقل هستند.

یکی از نقاط اشتراک بین اکثر کارهای گذشته، استفاده از روش عامل-منتقد در این کارها است که نشان از کارایی این روش نسبت به سایر روش‌های یادگیری تقویتی دارد. همچنین، شباهت دیگر بین روش‌های گذشته را می‌توان یادگیری یک سیاست برای بهینه‌سازی چندین هدف دانست که در نتیجه تعریف پاداش را تسهیل می‌کند اما نتایج نشان می‌دهد که قادر به اکتشاف کارا

[3] Multi-Objective Markov Decision Process
[4] Experience replay
[5] Time-varying recommendation
[6] Pareto-optimal

در فضای حالت سیاست برای یافتن یک سیاست مناسب در تمامی اهداف نمی‌باشند.

در کل، تفاوت‌های اصلی بین چارچوب پیشنهادی ما و اکثر مطالعات مشابه ذکرشده به صورت زیر خلاصه می‌شوند:

- بر خلاف برخی از کارهای گذشته مانند [۱۷] که همه اهداف به عنوان یک هدف ترکیب می‌شوند، در چارچوب پیشنهادی ما، هر عامل در هر گام زمانی یک *بردار* پاداش دریافت می‌کند نه یک مقدار عددی.
- چارچوب پیشنهادی، چندین سیاست را به طور همزمان یاد می‌گیرد که به ما کمک می‌کند فضای سیاست‌ها را به طور کارآمدتری کاوش (Explore) کنیم.
- چارچوب پیشنهادی از یک لایه‌ی توجه (Attention) به عنوان مکانیزم نظارتی میان بلوک‌های محلی استفاده می‌کند.
- در چارچوب مبتنی بر عامل-منتقد که توسط یانگ و همکاران پیشنهاد شده است [۲۰]، برای نظارت بر عامل‌ها، یک منتقد جداگانه برای هر هدف وجود دارد. اما، در چارچوب پیشنهادی ما، یک منتقد سراسری وجود دارد که بر همه‌ی عامل‌ها نظارت می‌کند.
- در چارچوب مبتنی بر عامل-منتقد که توسط ریموند و همکاران پیشنهاد شده است [۲۱]، یک عامل و یک منتقد به عنوان یک توزیع چندمتغیره در نظر گرفته می‌شود و مسئله با استفاده از یک رویکرد تک‌هدفه حل می‌شود. با این حال، چارچوب پیشنهادی ما یک رویکرد چند هدفه است و از چندین بلوک عامل-منتقد استفاده می‌کند که هر یک مرتبط با یکی از اهداف است.
- تفاوت میان روش پیشنهادی ما و روش پیشنهادی نوینگن و همکاران [۲۲] این است که روش آن‌ها فاقد بلوک نظارت سراسری بر عامل‌ها است.

۳- روش پیشنهادی

در این بخش روش پیشنهادی با نام روش[۷] MACA ارائه می‌گردد. در ابتدا مسئله به صورت یک مسئله‌ی چند هدفه فرآیند تصمیم‌گیری مارکف مدل می‌شود. سپس چارچوب پیشنهادی ارائه شده و جزئیات آن تشریح می‌گردد.

۳-۱- فرموله کردن مسئله

یک فرآیند تصمیم‌گیری مارکوف چند هدفه، یک فرآیند تصمیم‌گیری مارکف است که تابع پاداش آن به جای یک مقدار عددی به صورت برداری از N پاداش است، یک پاداش به ازای هر هدف (N تعداد اهداف را مشخص می‌کند). ما مسئله‌ی تصمیم‌گیری متوالی چند هدفه را به صورت یک MOMDP مدل می‌کنیم. تعریف پنج‌تایی MOMDP به شکل $(\mathcal{S},\mathcal{A},\mathcal{T},\gamma,\vec{R})$ است که در آن:

- فضای حالت $\mathcal{S}$: مجموعه متناهی از حالت‌ها است،
- فضای اقدام $\mathcal{A}$: مجموعه متناهی از اقدام‌ها است،
- تابع گذار $\mathcal{T}$: تابعی به شکل $\mathcal{T}:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,1]$ است که نشان می‌دهد اگر در حالت کنونی s اقدام a انجام شود، حالت بعدی s' با احتمال $\mathcal{T}(s,a,s')$ تعیین می‌شود.
- تابع پاداش $\vec{R}$: این تابع به صورت $\vec{R}:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}^N$ تعریف می‌شود و مشخص می‌کند که با انجام اقدام a در حالت s و رفتن به حالت s'، بردار N بعدی پاداش بدست می‌آید. در فرآیند تک هدفه، $N=1$ است در حالی که در MOMDP، $N>1$ و فضای هدف چند بعدی می‌شود.
- γ: ضریب کاهش (Discount) است که اهمیت نسبی پاداش‌های آنی (Immediate) را مشخص می‌کند.

۳-۲- چارچوب پیشنهادی

در این بخش، چارچوب پیشنهادی یادگیری تقویتی عمیق چند هدفه‌ی MACA معرفی می‌شود که بر پایه‌ی معماری عامل-منتقد است. روش‌های یادگیری تقویتی به سه دسته تقسیم می‌شوند. *روش‌های مبتنی بر ارزش* از تابع ارزش استفاده می‌کنند و سپس عملی که ارزش بیشتری دارد انتخاب می‌گردد. روش‌های مبتنی بر ارزش، نیاز به نمونه‌های کمتری دارند و پایدارتر هستند، اما هنگامی که فضای اقدام بزرگ است زمان محاسباتی بالایی خواهند داشت. *روش‌های مبتنی بر سیاست* به صورت مستقیم یک سیاست را یاد می‌گیرند که حالت فعلی عامل را به عنوان ورودی دریافت کرده و اقدام را به عنوان خروجی تولید می‌کند. این روش‌ها عموماً همگرایی سریع‌تری دارند.

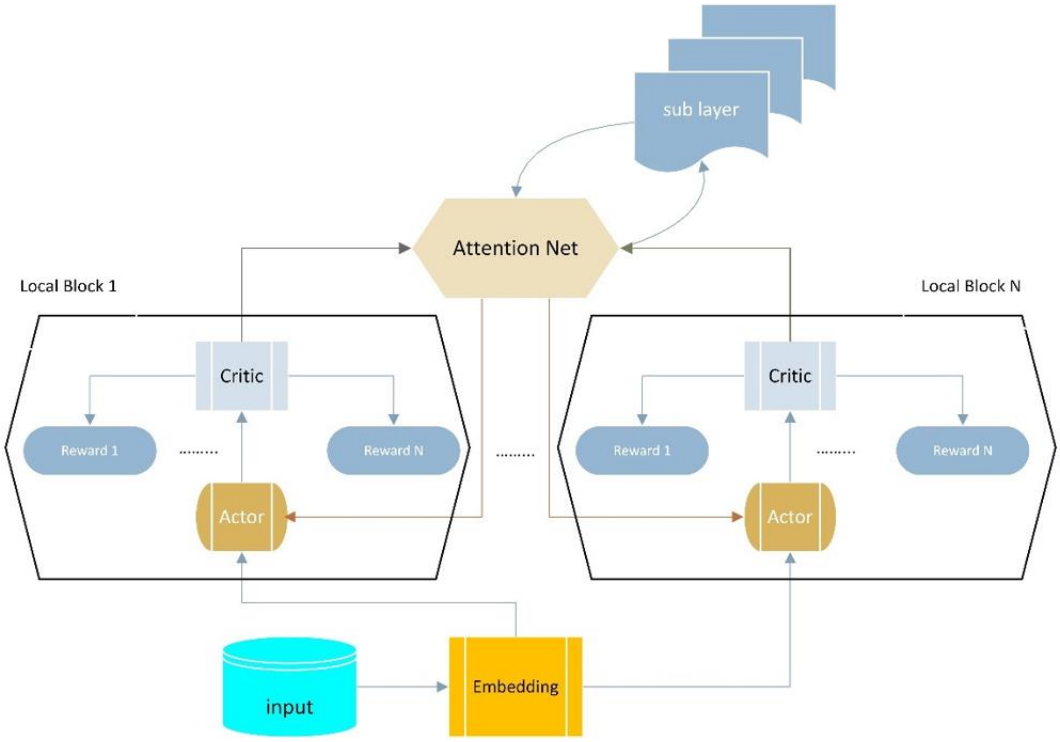
روش‌های *عامل-منتقد* (Actor-Critic) این دو روش را با هم ترکیب می‌کنند تا از مزیت‌های هر دو روش بهره ببرند. عامل، نمایانگر سیاست است و بر اساس حالت کنونی، اقدام‌هایی را پیشنهاد می‌دهد (با استفاده از شبکه سیاست)، در حالی که منتقد تابع ارزش را تخمین می‌زند (با استفاده از شبکه Q) و بازخوردی در مورد کیفیت اقدام‌های عامل ارائه می‌دهد. این ترکیب به عامل امکان می‌دهد که سیاست‌های بهتری یاد بگیرد و در محیط‌های پیچیده تصمیمات بهتری با اطلاعات بیشتری بگیرد. ما چارچوب عامل-منتقد را که به صورت پیش‌فرض یک سیاست یاد می‌گیرد، به یک معماری یادگیری تقویتی چند هدفه تغییر داده‌ایم.

معماری پیشنهادی، همان‌طور که در شکل ۱ نشان داده شده است، شامل N بلوک عامل-منتقد موازی است که توسط یک لایه توجه سراسری مدیریت می‌شود. همچنین لازم به ذکر است ورودی قبل از رسیدن به بلوک‌های محلی تعبیه (embedding) می‌شود تا تمام ورودی‌ها دارای اندازه یکسان و فشرده باشند.

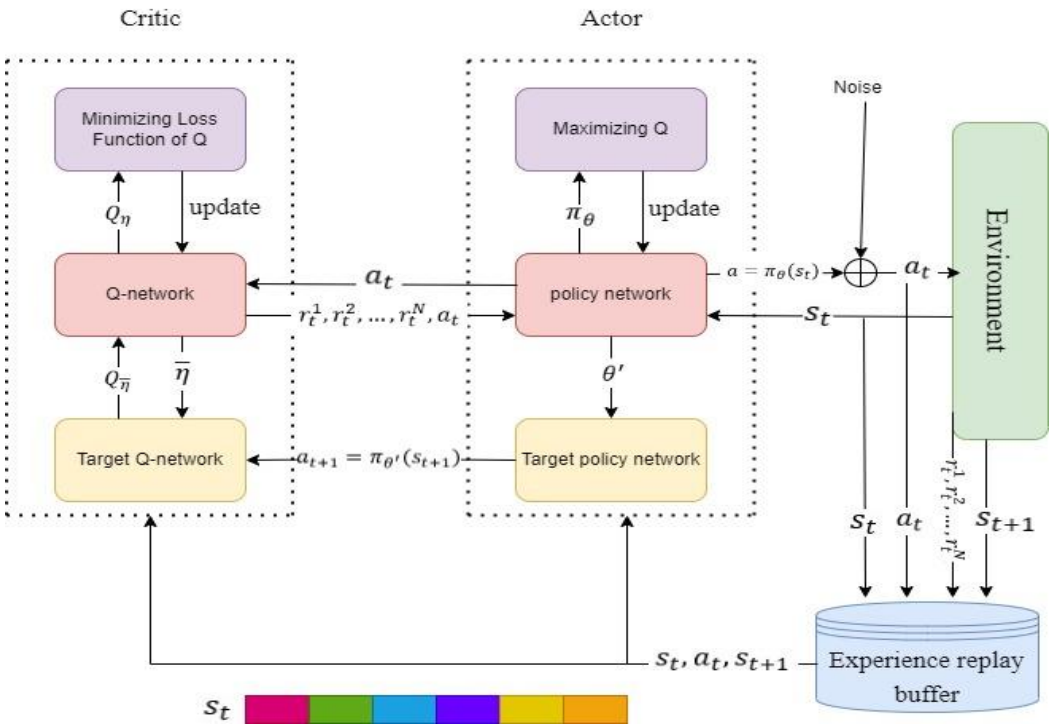
در معماری پیشنهادی، برای هر هدف، یک بلوک محلی عامل-منتقد وجود دارد که سعی می‌کند سیاستی را یاد بگیرد تا هدف مربوط به خود را بیشینه کند. همچنین، یک لایه توجه سراسری وجود دارد که بازخورد منحصر به فردی به بلوک‌های محلی ارائه می‌دهد تا آن‌ها را مدیریت کند. به عبارت دیگر، لایه توجه سراسری خروجی بلوک‌های محلی (اقدام‌های انتخاب شده برای حالت فعلی) را دریافت می‌کند و خروجی‌ها را با یکدیگر مقایسه کرده و تعیین می‌کند که هر بلوک محلی چه عملکردی در هر یک از هدف‌ها داشته است و بازخورد جداگانه‌ای به هر بلوک محلی ارائه می‌دهد تا بلوک محلی سیاست خود را بر اساس این بازخورد به‌روز کند.

همچنین، در هر بلوک محلی، یک منتقد محلی وجود دارد که پاداش‌های جداگانه‌ای را برای تمام اهداف به عامل آن بلوک ارائه می‌دهد. هر بلوک محلی سعی می‌کند تصمیماتی را اتخاذ کند که همه اهداف خود را بیشینه کند، اما وزن بیشتری به یک هدف (هدف مخصوص آن بلوک) داده می‌شود. اجزای اصلی الگوریتم پیشنهادی به تفصیل در ادامه توصیف شده‌اند.

[7] Multi-objective Actor-Critic Algorithm



شکل ۱: معماری روش پیشنهادی MACA



شکل ۲: چارچوب بلوک محلی

لایه توجه سراسری

لایه توجه سراسری شامل N زیر-لایه است که مبتنی بر مکانیزم توجه [۲۳] بوده و به شرح زیر عمل می‌کند. در هر تکرار، هر بلوک محلی به لایه توجه سراسری یک اقدام ارائه می‌دهد که در پاسخ به حالت فعلی است. لایه سراسری تمامی اقدام‌های بلوک‌های محلی را جمع کرده و این اقدام‌ها را بر اساس پاداش در اهداف مختلف رتبه‌بندی می‌کند. سپس هر زیرلایه براساس اینکه متناظر با کدام بلوک محلی است، به بررسی رتبه‌بندی آن بلوک می‌پردازد تا ببیند بلوک محلی i در هر هدف در کدام رتبه قرار دارد و همچنین میزان شباهت بلوک محلی i را با دیگر بلوک‌های محلی بدست می‌آورد تا متوجه شود که بلوک محلی i به چه میزان باید تغییر کند تا در تکرار بعدی در رتبه بالاتری قرار بگیرد. در نهایت، این زیرلایه بازخوردی به بلوک محلی i می‌دهد که با نماد f_i نشان می‌دهیم که در رابطه (۱) به زیر تعریف شده است:

$$f_i = \sum_{j \neq i} \alpha_j u(V k_j(s_t, a_t)) \quad (۱)$$

که در آن s_t و a_t به ترتیب حالت فعلی و اقدام بلوک محلی j و k_j یک تابع پرسپترون چند لایه (MLP) است. u یک تابع ReLU غیر خطی است. V آرایه‌ای از مقادیر است که هرمقدار آن با v_i نشان داده می‌شود که حاوی تابع ارزش برای بلوک محلی i است. همچنین α_j وزن توجهی است که اقدام بلوک محلی i را با اقدام بلوک محلی j مقایسه می‌کند و مقدار شباهت این دو اقدام را به دست می‌آورد.

بلوک محلی

در هر بلوک محلی، یک معماری عامل-منتقد وجود دارد. هر بلوک محلی سیاست خاص خود را دارد که تمام اهداف را در نظر می‌گیرد. در هر بلوک محلی، یک منتقد وجود دارد که برای هر هدفی یک پاداش محاسبه می‌کند. منتقد در هر بلوک محلی تلاش می‌کند تا تابع ضرر را به حداقل برساند. تابع ضرر، معیاری بر ای سنجش مناسب بودن مدل از نظر قابلیت پیش‌بینی ارزش‌های جدید است و به صورت تفاضل بین بازخورد ارائه شده توسط منتقد و تابع ارزش در وضعیت فعلی عامل تعریف می‌شود (رابطه (۲)):

$$L = \sum_{i=1}^{N} E[(Q_\eta^i(s_t, a_t) - V^i(s_t))^2] \quad (۲)$$

که در آن $Q_\eta^i(s_t, a_t)$ ارزشی است که منتقد در بلوک محلی i پیش‌بینی می‌کند و $V^i(s_t)$ مقدار تابع ارزش در حالت فعلی عامل در بلوک محلی i است. تابع ارزش برابر با جمع پاداش‌ها و تفاوت بین بازخورد شبکه هدف Q و ارزش سیاست هدف در حالت s_{t+1} است که مطابق رابطه (۳) محاسبه می‌شود:

$$V^i(s_t) = \sum_{j=1}^{N} r_t^j + \gamma E\left[Q_{\bar{\eta}}^i(s_{t+1}, a_{t+1}) - \alpha \log\left(\pi_{\theta_i'}(s_{t+1}|a_{t+1})\right)\right] \quad (۳)$$

که در آن α نرخ یادگیری، r_t^j پاداش هدف j در زمان t است. همچنین $\bar{\eta}$ پارامتر منتقد هدف و θ' پارامتر سیاست هدف است. s_{t+1} حالات بعدی است که از فضای حالات $\mathcal{S}$ گرفته شده است. a_{t+1} اقدام در زمان $t+1$ است. تابع بازخورد در معادله (۴) بیان می‌شود:

شکل ۲، معماری بلوک محلی چارچوب پیشنهادی را نشان می‌دهد که شامل چهار شبکه، یعنی یک شبکه Q، یک شبکه سیاست، یک شبکه Q هدف، و یک شبکه سیاست هدف است.

$$Q_\eta^i(s_t, a_t) = d_i(k_i(s_t, a_t), f_i) \quad (۴)$$

که در آن d تابع MLP دو لایه و η پارامتر وزن برای منتقد در بلوک محلی i است و به شکل معادله (۵) بیان می‌شود:

$$\eta \leftarrow \eta + \alpha \frac{1}{B} \sum_t \left[\left(V^i(s_t) - Q_\eta^i(s_t, a_t)\right) \nabla_\eta Q_\eta^i(s_t, a_t)\right] \quad (۵)$$

که در آن B اندازه دسته (batch) است. $\pi_i(\theta)$ سیاست بلوک محلی i است که استراتژی است که عامل تصمیم‌گیرنده در پاسخ به وضعیت فعلی، برای اقدام بعدی خود درنظر می‌گیرد و به شکل رابطه (۶) تعریف می‌شود:

$$\pi_i(\theta) = E\left[\nabla_{\theta_i} \log\left(\pi_{\theta_i}(s_t, a_t)\right)\left(-\alpha log\left(\pi_{\theta_i}(s_t|a_t)\right) + Q_\eta^i(s_t, a_t) - f_i\right)\right] \quad (۶)$$

که در آن هر عامل در بلوک محلی، سیاست خود را با دریافت بازخورد از لایه توجه سراسری به روز می‌کند تا روند بهتری داشته باشد. هر بلوک محلی در هر زمان اقدام خود را به لایه توجه سراسری ارسال می‌کند و منتظر بازخورد این لایه می‌ماند. معماری یک بلوک محلی با جزئیات بیشتر در شکل ۲ و شبه کد اجرای آن در شکل ۳ آمده است.

Algorithm *Local_block b*

input: Actor learning rate α_θ, Critic learning rate α_η, discount factor γ, batch size B and reward function r_i

```
{
    t = 0 ;
    replay buffer P = 0 ;
    Initialize W for block b
    While (s != s_T){
      s = s_t ;
      a = a_t ;
      For i = 1 ....N do {
          Calculate reward r_i^t based on the feedback of the user;
          r_t[i] = r_i^t ;   /* The rewards here are given by the
          critics of each local block as feedback to the actor of
          that local block. r_t is the vector where
          rewards are stored. */
       }
       Save (s , a , s_{t+1} , r_t) to P;
       s = s_{t+1} ;
       t = t + 1 ;
    }
    for i = 1 ....N do {
       Q_η^i(s_t, a_t) = d_i(k_i(s_t, a_t), f_i) ;
       Calculate Equation (3);
    }
    Calculate the error function expressed in Equation (2);
    π_i(θ) = E[∇_{θ_i} log(π_{θ_i}(s_t, a_t)) (−α log(π_{θ_i}(a_t|s_t)) +
    Q_η^i(s_t, a_t) − f_i)] ;
    η ← η + α (1/B) Σ_t [(V^i(s_t) − Q_η^i(s_t, a_t)) ∇_η Q_η^i(s_t, a_t)];
}
```

شکل ۳: شبه‌کد بلوک محلی

در شکل ۲، شبکه Q هدف، $Q_{\bar{\eta}}$ و شبکه سیاست هدف، $\pi_{\theta'}$ نام دارد. آن‌ها به ترتیب کپی‌هایی با تأخیر زمانی، شبکه Q و شبکه سیاست هستند. شکل ۲

۴- نتایج و بحث

در این بخش روش پیشـنهادی مورد ارزیابی قرار می‌گیرد. ابتدا مسئله محک و اهداف معرفی می‌شوند. سپس مجموعه‌دادها و معیارهای ارزیابی معرفی شده و در نهایت نتایج روش پیشنهادی با روش‌های اخیر مقایسه می‌گردد.

۴-۱- تعریف مسئله محک و اهداف

ما برای ارزیابی چارچوب پیشنهادی و مقایسه‌ی آن با کارهای گذشته، مسئله توصیه اقلام[8] به کاربران را به عنوان مسئله محک (Benchmark) انتخاب کرده‌ایم:

تعریف (توصیه اقلام توسط سیستم‌های پیشنهاددهنده): یک مجموعه‌داده شامل بازخورد (شامل پذیرش یا رد کاربران و امتیازهایی که کاربران به اقلام تخصیص می‌دهند) از n کاربر به اقلام مختلف در زمان‌های مختلف داده‌شده است. هدف یافتن یک سیاست برای سیستم توصیه‌گر جهت بهینه‌سازی توصیه به کاربران نسبت به N هدف داده‌شده است.

۴-۲- مجموعه داده‌ها

برای ارزیابی از مجموعه داده‌های MovieLens [۲۴] و Kiva.org [۲۵] استفاده می‌کنیم. مجموعه داده اول، MovieLens، مجموعه‌داده‌های عمومی برای سیستم‌های توصیه‌گر است که شامل کاربران، آیتم‌ها و تعاملات کاربر-آیتم می‌باشد (امتیازات کاربران به آیتم‌ها در بازه ۱ تا ۵ است). فیلم‌های موجود در مجموعه داده بر اساس ژانر دسته‌بندی شده‌اند.

Kiva.org یک پلتفرم جمع‌آوری سرمایه (Crowdfundng) آنلاین است که وام‌های کوچک به افراد نیازمند در سراسر جهان ارائه می‌دهد. Kiva سعی می‌کند تساوی در دسترسی به سرمایه در مناطق مختلف ایجاد کند تا وام‌ها از هر منطقه احتمال مناسبی برای تامین شدن داشته باشند. این مجموعه داده، با بهره‌گیری از تکنیک‌های پیش‌پردازش استفاده شده در [۲۶]، شامل لاگ‌های تراکنش‌های اعطای وام در طول یک دوره شش ماهه تهیه شده است. این سایت لیست‌های کوتاهی از وام‌های پیشنهادی به کاربران برای حمایت ارائه می‌دهد و وام‌دهندگان به این پیشنهادها پاسخ می‌دهند (امتیازات وام‌دهندگان به وام‌ها در بازه ۱ تا ۶ است). وام‌های مجموعه‌های داده بر اساس کشورها دسته‌بندی شده‌اند. مشخصات مجموعه داده‌ها در جدول ۱ نمایش داده شده است.

جدول ۱: خصوصیات مجموعه‌های داده

مجموعه داده	کاربران	آیتم‌ها	تعاملات
MovieLens 100K	۹۴۳ کاربر	۱۶۸۲ فیلم	۱۰۰۰۰۰ امتیاز
Kiva	۱۷۸۷۸۸ وام‌دهنده	۱۱۳۷۳۸ وام	۸۵۳۲۶۹ تراکنش

که ۷۰ درصد از داده‌ها برای آموزش و ۳۰ درصد از داده‌ها برای آزمون استفاده شده است.

۴-۳- معیارهای ارزیابی

چارچوب‌های پیشنهادی ارائه شده را بر اساس دو هدف میزان صحت و انصاف در توصیه اقلام از همه‌ی گروه‌ها و همچنین دو معیار عملکردی سرعت همگرایی و حافظه‌ی مصرفی، ارزیابی می‌کنیم. هدف صحت، نشان‌دهنده‌ی میزان رضایت‌مندی کاربران نسبت به اقلام پیشنهادی توسط سیستم توصیه‌گر است. صحت لیست اقلام توصیه شده برای کاربر j را با معیار نرخ تبدیل[9] در رابطه (۷) اندازه‌گیری می‌کنیم:

$$CVR = \frac{\sum_{k=1}^{T} y_{j,a_k}}{T}, \quad 0 < CVR < 1 \tag{۷}$$

جایی که y_{j,a_k} بازخورد کاربر j برای قلم a_k در لیست توصیه شده است و T نشان‌دهنده زمان بیشتر.

از معیار PropFair [۲۷] برای اندازه‌گیری منصفانه بودن لیست توصیه‌ها استفاده می‌شود. اقلام قابل توصیه به گروه‌های مختلف تعلق دارند (ژانر برای مجموعه‌داده‌ی MovieLens و کشور برای مجموعه‌داده‌ی Kiva). منصفانه بودن لیست توصیه بدین معنی است که اقلام توصیه شده از گروه‌های مختلف بوده و تا حد ممکن اقلام همه‌ی گروه‌ها شانس انتخاب شدن داشته باشند. منصفانه بودن لیست توصیه برای کاربر j در رابطه (۸) تعریف شده است:

$$PropFair = \sum_{i=1}^{l} \log\left(1 + k_{i,j}^{T}\right), \quad 0 < PropFair < 1 \tag{۸}$$

که در آن $k_{i,j}^{T}$ بردار تخصیص است که در ادامه (رابطه (۹)) تعریف شده است. اقلام بر اساس ارتباطشان به l گروه $G \in (g_1, g_2, \dots, g_l)$ تقسیم می‌شوند. $L_i(a)$ نشان می‌دهد که آیا قلم a عضوی از گروه g_i است یا خیر. اگر $a \in g_i$ آنگاه $L_i(a) = 1$، در غیر این صورت $L_i(a) =$. بردار تخصیص $k_{i,j}^{t}$ نشان‌دهنده نسبت بازخورد کاربر j به اقلام گروه i به بازخورد کاربر j به اقلام همه گروه‌ها تا زمان t است:

$$k_{i,j}^{t} = \frac{\sum_{k=1}^{t} y_{j,a_k} L_i(a_k)}{\sum_{i'=1}^{l} \sum_{k=1}^{t} y_{j,a_k} L_{i'}(a_k)} \tag{۹}$$

معیار سرعت همگرایی نشان می‌دهد که سیستم در کدام مرحله از الگوریتم به همگرایی می‌رسد. هنگامی‌که همه اهداف در طول تعداد مراحل مشخصی تغییر نمی‌کنند، فرض می شود که الگوریتم همگرا شده است. همچنین پارامترهای آزمون در جدول ۲ آورده شده است.

جدول ۲: پارامترهای آزمون

Parameter	Value
number of epochs	100
batch size	128
embedding dimension	100
learning rate	0.001
discount factor	0.9
hidden dimension	128
replay buffer size	1000000

۴-۴- نتایج

ما چارچوب پیشنهادی خود را با چهار روش یادگیری تقویتی چندهدفه مقایسه می‌کنیم. این روش‌ها به‌روزترین روش‌های ارائه شده در این زمینه با بهترین نتایج می‌باشند:

- MCSP. در این مقاله، از چندین منتقد با یک عامل برای بهینه‌سازی چند هدفه استفاده شده است [۲۲].
- MoTiAC. در این مقاله، معماری عامل-منتقد چند هدفه برای مسئله مناقصه در زمان بلادرنگ (Real-time bidding) استفاده شده است [۲۰].
- MoCAC. در این مقاله، رویکرد عامل-منتقد دسته‌ای چندهدفه ارائه شده است [۲۱].
- FairRec. در این مقاله، چارچوب توصیه‌گر آگاه از انصاف

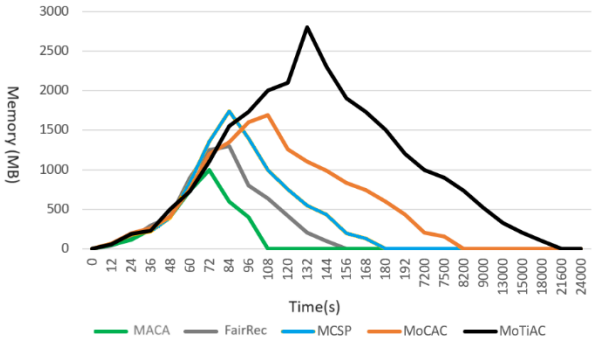
[8] Item Recommendation

[9] Conversion rate (CVR)

(Fairnes-aware) به کمک یادگیری تقویتی ارائه داده است [۱۷].

مقایسه زمان اجرا

شکل ۴، مقایسه روش‌های در نظر گرفته شده را بر اساس زمان همگرایی و میزان استفاده از حافظه نشان می‌دهد. نتایج نشان می‌دهد که چارچوب پیشنهادی نسبت به کارهای قبلی از نظر زمان همگرایی و استفاده از حافظه بهتر عمل کرده است. لایه توجه سراسری به رسیدن سریع‌تر به نقطه همگرایی کمک می‌کند. همچنین تمام بلوک‌های محلی مستقل هستند و به صورت موازی اجرا می‌شوند. بنابراین، منجر به زمان اجرای کمتر می‌شود. از سوی دیگر به دلیل استفاده از تعبیه ورودی توسط لایه تعبیه در چارچوب پیشنهادی، فضای حافظه کمتری مصرف شده است.



شکل ۴: مقایسه زمان همگرایی و حافظه مصرفی

مقایسه صحت

جدول ۳ صحت روش‌های در نظر گرفته شده را نشان می‌دهد. این جدول نشانه‌گر این است که چارچوب پیشنهادی دقیق‌تر از دیگر روش‌ها است. دلیلش این است که علاوه بر یک لایه توجه سراسری، منتقد انحصاری دیگری در هر بلوک محلی وجود دارد که منحصراً برای هر هدف آموزش دیده شده است و به عامل آن بلوک محلی بازخورد می‌دهد.

جدول ۳: مقایسه صحت لیست پیشنهادی

	Kiva	MovieLens 100k
Method	Accuracy	Accuracy
MCSP	0.779	0.887
MoCAC	0.630	0.830
MoTiAC	0.700	0.857
FairRec	0.690	0.870
MACA	**0.925**	**0.931**

معیار انصاف

این مقایسه در جدول ۴ نشان داده شده است. همان‌طور که مشخص است روش پیشنهادی در هدف دوم، انصاف، عملکرد بهتری نسبت به سایر روش‌ها داشته است. دلیل آن این است که هر بلوک محلی مسئول بهینه‌سازی یک هدف مشخص است. بنابراین سیاست‌های مختلفی برای اهداف متناقض یادگیری شده و در طول زمان به هم همگرا می‌گردند. همچنین، پاداش برای همه اهداف هم در بلوک‌های محلی و هم در لایه توجه سراسری در هر مرحله محاسبه می‌شود. همچنین در یک لایه توجه، بازخورد منحصر به فردی به هر یک از بلوک‌های محلی می‌دهد تا بلوک‌های محلی بهبود یابند.

جدول ۴: مقایسه معیار انصاف

	Kiva	MovieLens
Method	ProbFair	ProbFair
MCSP	0.790	0.710
MoCAC	0.710	0.653
MoTiAC	0.770	0.709
FairRec	0.866	0.799
MACA	**0.965**	**0.976**

هزینه برقراری انصاف

اگر چارچوب پیشنهادی خود را به صورت تک هدفی اجرا کنیم، به نتایج نشان داده شده در جدول ۵ خواهیم رسید.

جدول ۵: مقایسه چارچوب پیشنهادی چند هدفه با تک هدفه

	Kiva		MovieLens	
Method	PropFair	Accuracy	PropFair	Accuracy
Single-objective	0.801	0.929	0.756	0.940
MACA	0.965	0.925	0.976	0.931

نتایج نشان می دهد که با کاهش کمتر از ۱ درصدی صحت (هدف اول)، بین ۱۵ تا ۲۰ درصد انصاف (هدف دوم) بیشتری نسبت به حالت تک هدفه به دست آورده‌ایم. این نتایج نشان می‌دهد چارچوب پیشنهادی کاوش مناسبی در فضای سیاست‌ها برای ایجاد تعادل بین دو هدف متضاد داشته است.

۵- نتیجه‌گیری و کارهای آینده

در این مقاله، چارچوبی مبتنی بر یادگیری تقویتی عمیق جهت بهینه‌سازی مسائل تصمیم‌گیری متوالی چند هدفه ارائه دادیم. روش پیشنهادی تعمیم‌سازی از روش عامل-منتقد است که دارای مولفه‌های عامل-منتقد مجزا برای یادگیری سیاست‌های بهینه برای اهداف مختلف است. این چارچوب در نهایت با همگرا کردن سیاست‌های بلوک‌های مختلف، یک سیاست بهینه سراسری را به عنوان خروجی برمی‌گرداند. روش ارائه شده قابل تعمیم به مسائل با هر تعداد هدف می‌باشد.

برای ارزیابی از مسئله سیستم‌های توصیه‌گر اقلام استفاده شده است و دو مجموعه داده برای این کار مورد استفاده قرار گرفته‌اند. در این مسئله دو هدف صحت لیست توصیه شده (مورد پسند بودن اقلام توصیه شونده برای کاربر) و انصاف (انتخاب اقلام از همه گروه‌های مختلف) مورد ارزیابی قرار گرفت. نتایج نشان داد که روش ارائه شده به دلیل همزمانی یادگیری سیاست‌های مختلف به طور موازی، قابلیت کاوش مناسبی در فضای سیاست‌ها دارد و سیاست تولید شده در هر دو هدف نسبت به روش‌های چند هدفه قبل برتری یافته است. ضمن این که به دلیل موازی‌سازی بلوک‌های عامل-منتقد سرعت همگرایی بالاتری نیز دارد.

همچنین نتایج نشان داد که چنانچه از روش بهینه‌سازی چند هدفه مناسب استفاده گردد، با میزان کاهش بسیار کم در هدف اول (۱٪ در صحت)، افزایش قابل توجهی در اهداف دیگر (۱۵-۲۰٪ در انصاف) قابل دستیابی خواهد بود.

در ادامه، پیشنهاد می‌شود روش ارائه شده برای کاربردهای چند هدفه دیگر نظیر بهینه‌سازی مسیریابی ربات، پیش‌بینی بازار سهام، تخصیص منابع در رایانش ابری و غیره مورد ارزیابی بیشتر قرار گیرد.

مراجع

[1] N. Gunantara, “A review of multi-objective optimization: Methods and its applications”, Cogent Engineering, vol. 5, no. 1, pp. 1–16, 2018.

[2] A. Konak, D.W. Coit, and E.S. Alice, “Multi-objective optimization using genetic algorithms: A tutorial”, Reliability engineering & system safety, pp. 992–1007, 2006.

[3] Q. Q. Wang, Z. D. Li, W. W. Wang, C. L. Zhang, L. Q. Chen, and L. Wan, “Multi-objective optimization design of wheat centralized seed feeding device based on particle swarm optimization (PSO) algorithm”, International Journal of Agricultural and Biological Engineering, vol. 13, no. 6, pp. 76–84, 2020.

[4] B. Razaghi, M. Roayaei, and N. M. Charkari, “On the Group-Fairness-Aware Influence Maximization in Social Networks”, IEEE Transactions on Computational Social Systems, 2022.

[5] M. A. Wiering, M. Withagen, and M. M. Drugan, “Model-based multi-objective reinforcement learning”, IEEE symposium on adaptive dynamic programming and reinforcement learning, pp. 1-6, 2014.

[6] R. Yang, X. Sun, and K. Narasimhan, “A generalized algorithm for multi-objective reinforcement learning and policy adaptation” Advances in Neural Information Processing Systems, vol. 32, 2019.

[7] J. Xu, Y. Tian, P. Ma, D. Rus, S. Sueda, and W. Matusik, “Prediction-guided multi-objective reinforcement learning for continuous robot control”, 37th International Conference on Machine Learning (ICML), pp. 10538–10547, 2020.

[8] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley, “A Survey of Multi-Objective Sequential Decision-Making”, Journal of Artificial Intelligence Research, vol. 48, pp. 67–113, 2013.

[9] M. Rodriguez, C. Posse, and E. Zhang, “Multiple objective optimization in recommender systems”, Proceedings of the sixth ACM conference on Recommender systems, pp. 11-18. 2012.

[10] C. Li and K. Czarnecki, “Urban Driving with Multi-Objective Deep Reinforcement Learning”, Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, pp. 359-367, 2019.

[11] Z. Miao, J. Yu, J. Ji, and J. Zhou, “Multi-objective region reaching control for a swarm of robots”. Automatica, vol. 103, pp. 81–87, 2019.

[12] J. Jin and X. Ma, “A Multi-Objective Agent-Based Control Approach with Application in Intelligent Traffic Signal System”, IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 10, pp. 3900–3912, 2019.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, “Playing Atari with Deep Reinforcement Learning”, arXiv preprint arXiv:1312.5602, 2013.

[14] T. T. Nguyen, N. D. Nguyen, P. Vamplew, S. Nahavandi, R. Dazeley, C. P. Lim, “A multi-objective deep reinforcement learning framework”, Engineering Applications of Artificial Intelligence, vol. 96, p. 103915, 2020.

[15] N. M. Tabatabaei, S.R. Mortezaei, S. Shargh, & B. Khorshid, "Solving multi-objective optimal power flow using multi-objective electro search algorithm." International Journal on Technical and Physical Problems of Engineering, 9, pp. 1-8, 2017.

[16] N. Tabrizi, E. Babaei, & M. Mehdinejad, M. “An interactive fuzzy satisfying method based on particle swarm optimization for multi-objective function in reactive power market”, Iranian Journal of Electrical and Electronic Engineering, 12(1), 65-72, 2016.

[17] W. Liu, F. Liu, R. Tang, B. Liao, G. Chen, and P. A. Heng, “Balancing Between Accuracy and Fairness for Interactive Recommendation with Reinforcement Learning” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12084, pp. 155–167, 2020.

[18] M. A. Wiering, E. D. De Jong, “Computing optimal stationary policies for multi-objective Markov decision processes”, Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning, pp. 158–165, 2007.

[19] M. Khamis, W. Gomaa, “Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework”, Engineering Applications of Artificial Intelligence, vol. 29, pp. 134-151, 2014.

[20] J. Lu, H. Liu, Q. Chen, G. Chen, “MoTiAC : Multi-objective Actor-Critics for Real-time Bidding in Display Advertising”, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 20-37, 2022.

[21] M. Reymond, C. Hayes, D. M. Roijers, D. Steckelmacher, A. Nowé, “Actor-Critic Multi-Objective Reinforcement Learning for Non-Linear Utility Functions”, Autonomous Agents and Multi-Agent Systems, vol. 37, no. 2, 2023.

[22] N. D. Nguyen, T. T. Nguyen, P. Vamplew, R. Dazeley, S. Nahavandi, “A Prioritized objective actor-critic method for deep reinforcement learning”, Neural Computing and Applications, vol. 33, no. 16, pp. 10335–10349, 2021.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I Polosukhin, “Attention is all you need”, Advances in neural information processing systems, vol. 30, 2017.

[24] F. M. Harper, J. A. Konstan, “The MovieLens Datasets: History and Context”, ACM Transactions on Interactive Intelligent Systems, vol.5, 2016.

[25] V. Gajjala, R. Gajjala, A. Birzescu, S. Anarbaeva, “Microfinance in online space: a visual analysis of kiva.org”, Development in Practice, vol. 21, no. 6, pp. 880–893, 2011.

[26] Burke R. Liu W, “Personalizing fairness-aware re-ranking”, arXiv preprint arXiv:1809.02921, 2018.

[27] T. Mahmood, F. Ricci, “Learning and adaptivity in interactive recommender systems”, ACM International Conference Proceeding Series, vol. 258, pp. 75–84, 2007.