فصلنامه علمی - پژوهشی فرماندهی و کنترل، سال هشتم، شماره چهار، زمستان ۱۴۰۳

ارائه یک مدل ترکیبی از مدلسازی پیشرفته ریاضی و یادگیری تقویتی به منظور بهبود اثربخشی فرماندهی و تصمیم‌گیری در صحنه نبرد

محسن شکربیگی*[۱]، سامان عرفا[۲]

تاریخ دریافت: ۱۴۰۳/۱۱/۲۸ تاریخ پذیرش: ۱۴۰۴/۰۷/۲۸

چکیده

امروزه بهره‌گیری از مدل‌سازی ریاضی و روش‌های هوش مصنوعی در تحلیل و شبیه‌سازی صحنه‌های نبرد، نقش مهمی در بهبود فرایند تصمیم‌گیری و ارزیابی راهبردهای مختلف ایفا می‌کند. یکی از چالش‌های مهم در این زمینه، نحوه تصمیم‌گیری و افزایش اثربخشی تیمی است که از نظر توان رزمی و منابع در شرایط ضعیف‌تری نسبت به حریف قرار دارد. در این مقاله، با هدف بهبود فرایند فرماندهی و تصمیم‌گیری تیم ضعیف‌تر، یک مدل ترکیبی مبتنی بر مدل‌سازی ریاضی نظریه بازی‌ها و یادگیری تقویتی ارائه شده است. در مدل پیشنهادی، تعاملات رقابتی دو تیم و تصمیمات حمله و دفاع با استفاده از یک مدل ریاضی مبتنی بر نظریه بازی‌ها شبیه‌سازی شده و الگوریتم‌های Q-Learning و Deep Q-Network (DQN) برای آموزش یک مربی هوشمند به‌منظور انتخاب اقدامات مناسب برای تیم ضعیف‌تر به کار گرفته شده‌اند. عملکرد مدل پیشنهادی در قالب یک سناریوی نبرد متعارف و طی ۱۰۰۰ اپیزود شبیه‌سازی و بر اساس مجموع پاداش کسب‌شده و میزان سلامت باقی‌مانده اعضای تیم‌ها ارزیابی شده است. نتایج شبیه‌سازی نشان می‌دهد که از حدود اپیزود ۴۰۰، روند افزایشی و مثبت در مجموع پاداش تیم ضعیف‌تر مشاهده شده است؛ این روند با کاهش نرخ اکتشاف (ε) به کمتر از ۰.۲ و افزایش بهره‌گیری از سیاست آموخته‌شده همراه بوده و بیانگر بهبود و تثبیت تدریجی فرایند تصمیم‌گیری مربی هوشمند است.

واژگان کلیدی: مدلسازی ریاضی، یادگیری تقویتی، صحنه نبرد، فرماندهی و کنترل.

[۱]* نویسنده مسئول، پژوهشگر مجتمع دانشگاهی برق و کامپیوتر، دانشگاه صنعتی مالک اشتر m.shekarbaigi@gmail.com

[۲] پژوهشگر مجتمع دانشگاهی برق و کامپیوتر، دانشگاه صنعتی مالک اشتر s.orafa@chmail.ir

[Downloaded from ic4i-journal.ir on 2026-10-19]

۱. مقدمه و کلیات

شبیه سازی های صحنه نبرد به عنوان یک ابزار مؤثر، در تحقیقات علمی و نظامی برای تحلیل و بهینه‌سازی استراتژی‌ها مورد استفاده قرار می‌گیرند. در این شبیه‌سازی‌ها، تیم‌های مختلف با توانایی‌ها و منابع متفاوت در یک محیط پویا به رقابت می‌پردازند. یکی از مسائل چالش‌برانگیز در این زمینه، ارائه راهکاری برای تیم‌های ضعیف‌تر به منظور افزایش توان رقابتی با تیم‌های قوی‌تر است.

یادگیری تقویتی یکی از روش‌های نوین در حوزه یادگیری ماشین است که از طریق تعامل عامل با محیط و دریافت پاداش، بهبود عملکرد و تصمیم‌گیری بهینه را ممکن می‌سازد. این تکنیک‌ها به دلیل قابلیت یادگیری تدریجی و تعمیم‌دهی به شرایط جدید، ابزار مناسبی برای شبیه‌سازی و بهینه‌سازی استراتژی‌ها در بازی‌های جنگ هستند.

هدف اصلی این مقاله، توسعه یک مربی هوشمند است که با استفاده از یادگیری تقویتی، توانایی تحلیل وضعیت، پیش‌بینی اقدامات حریف و ارائه بهترین تصمیم‌ها را برای تیم ضعیف‌تر فراهم کند. برای این منظور، الگوریتم Q-Learning را به عنوان ابزار اصلی مورد استفاده قرار گرفته است. همچنین، یک مدل ریاضی برای توصیف دینامیک بازی و قوانین آن ارائه شده است تا بتوان عملکرد الگوریتم‌ها را ارزیابی کرد.

یکی از چالش‌های اساسی در طراحی مربی هوشمند برای تیم‌های ضعیف‌تر، نیاز به اطلاعات دقیق درباره مدل‌های فکری، الگوهای تصمیم‌گیری، ساختار فرماندهی و تاکتیک‌های تیم قوی‌تر یا دشمن است. در واقع، یادگیری تقویتی به‌صورت مستقیم قادر به پیش‌بینی اقدامات حریف نیست، مگر آن‌که به‌صورت مکرر با رفتار واقعی دشمن تعامل داشته باشد یا مدل دقیقی از تفکر و تصمیم‌گیری او در اختیار داشته باشد. در حوزه نظامی، دسترسی به چنین اطلاعات پایه‌ای درباره دشمن معمولاً امکان‌پذیر نیست یا بسیار محدود است. بنابراین، هرچند طراحی مربی هوشمند مبتنی بر یادگیری تقویتی می‌تواند به بهبود تصمیم‌گیری در برابر سناریوهای فرضی و مدل‌های تخمینی دشمن کمک کند، اما نمی‌توان انتظار داشت بدون داده‌های دقیق واقعی از حریف، به سطحی از پیش‌بینی کامل و تصمیم‌گیری بهینه در محیط‌های واقعی دست یابد. این محدودیت یکی از چالش‌های اصلی پیاده‌سازی الگوریتم‌های هوش مصنوعی در فرماندهی و کنترل نظامی است. لازم به ذکر است که در این تحقیق، به دلیل عدم دسترسی به اطلاعات دقیق دشمن، مدل تیم قوی‌تر به‌صورت ساده‌شده و با توزیع‌های احتمالی ثابت مدل‌سازی شده است. در صورتی که داده‌های رفتاری واقعی یا مدل‌های شناختی دشمن موجود باشد، می‌توان الگوریتم‌های یادگیری تقویتی مبتنی بر opponent modeling یا adversarial RL را برای بهبود دقت تصمیم‌گیری مربی تیم ضعیف‌تر به کار برد.

ساختار مقاله به صورت زیر است: ابتدا الگوریتم‌های پایه مورد بررسی قرار می‌گیرند. سپس مدل ریاضی صحنه نبرد ساخته شده و ترکیب آن با یادگیری تقویتی بیان می شود. در ادامه، نتایج شبیه‌سازی‌ها و تحلیل آن‌ها ارائه می‌گردد و در نهایت پیشنهاداتی برای تحقیقات آینده ارائه خواهد شد.

لازم به ذکر است که در این تحقیق، محیط بازی به‌عنوان یک محیط زمینی[1] مدل‌سازی شده است و سناریوهای هوایی، دریایی یا ترکیبی در دامنه این مدل قرار ندارند. هدف از این مدل‌سازی اولیه، ساده‌سازی تحلیل و تمرکز بر اصول یادگیری تقویتی و مدلسازی ریاضی برای تیم‌های پیاده‌نظام یا زمینی است.

۱.۱. مرور کارهای انجام شده و هدف

در مقالات و مطالعات اخیر در زمینه ی یادگیری تقویتی و کاربرد آن در بازی‌های جنگ نشان داده شده که این روش‌ها می‌توانند عملکرد بازیکنان یا تیم‌ها را بهبود بخشند. برای مثال ها[2] و همکاران در تحقیق خود به بررسی یادگیری تقویتی چندعامله و کاربرد آن در محیط‌های رقابتی پرداخته‌اند [۱]. آن‌ها نشان دادند که یادگیری چندعامله می‌تواند باعث بهبود همکاری

[1] Land-based Battlefield

[2] Huh

فصلنامهٔ علمی - پژوهشی فرماندهی و کنترل، سال هشتم، شمارهٔ چهار، زمستان ۱۴۰۳

[Downloaded from ic4i-journal.ir on 2026-10-19]

میان عوامل و ایجاد استراتژی‌های پیچیده شود. به‌عنوان مثال، در شبیه‌سازی‌های آن‌ها، تیم‌ها توانستند با ترکیب تصمیمات فردی، تاکتیک‌های گروهی مؤثری ایجاد کنند.

وانگ[1] و همکاران [۲] با استفاده از الگوریتم‌های تکاملی و یادگیری تقویتی، استراتژی‌های هوشمند برای نبردهای چندپهپادی توسعه داده‌اند. در این تحقیق، الگوریتم‌های تکاملی برای بهبود همگرایی و کاهش نوسانات در آموزش استفاده شدند. به‌عنوان نمونه، پهپادها توانستند با بهینه‌سازی تصمیمات پرواز و حمله، مأموریت‌های پیچیده‌ای را به‌طور هماهنگ انجام دهند. چنگ‌وانگ[2] و همکاران [۳] کاربرد یادگیری تقویتی در کشف کانال‌های فرماندهی و کنترل را بررسی کرده‌اند. این مطالعه نشان داد که الگوریتم‌های تقویتی می‌توانند در محیط‌های نامطمئن و پیچیده به صورت پویا عمل کنند. در مثال‌های این تحقیق، عوامل یادگیرنده موفق به شناسایی کانال‌های مخفی ارتباطی در شرایطی با حداقل اطلاعات اولیه شدند. در مطالعات اخیر، کاربردهای متنوع یادگیری تقویتی در حوزه فرماندهی و کنترل بررسی شده است. به عنوان مثال، در مرجع [٤] به بررسی آسیب‌پذیری عوامل RL در برابر حملات خصمانه پرداخته‌اند . مرجع [٥] چارچوب‌های سازماندهی اطلاعات برای طراحی سامانه‌های تصمیم‌یار را ارائه داده‌اند. همچنین[٦] و [۷] و [۸] کاربرد RL و DQN را در آموزش عوامل رزمی بررسی کرده‌اند . مرجع [۹] به معماری سلسله‌مراتبی برای هماهنگی عوامل یادگیرنده اشاره کرده و [۱۰] ارزیابی الگوریتم‌های مختلف RL را در محیط‌های نظامی انجام داده‌اند. این مطالعات نشان‌دهنده اهمیت یادگیری تقویتی در توسعه سامانه‌های هوشمند فرماندهی هستند. در مرجع [۱۱] تاکید کرده‌اند که بازی‌ها و شبیه‌سازها بسترهای مناسبی برای توسعه الگوریتم‌های هوش مصنوعی فرماندهی و کنترل هستند، زیرا امکان ایجاد سناریوهای متنوع، جمع‌آوری داده‌های آموزشی و ارزیابی عملکرد الگوریتم‌ها را فراهم می‌کنند.

اگرچه مطالعات فوق در بهبود یادگیری تقویتی در محیط‌های مختلف موفق بوده‌اند، اما محدودیت‌هایی نیز داشته‌اند:

۱. فقدان مدل‌سازی دقیق صحنه نبرد: بیشتر تحقیقات به شبیه‌سازی‌های ساده پرداخته‌اند و مدل ریاضی پیچیدگی کافی برای تحلیل دینامیک نبردهای واقعی را نداشته است.
۲. تمرکز کم بر تیم‌های ضعیف‌تر: تحقیقات قبلی بیشتر به بهبود عملکرد کلی عوامل پرداخته‌اند و به طور خاص بر تقویت عملکرد تیم‌های ضعیف‌تر تمرکز نکرده‌اند.

در مقابل، مقاله ی ما برتری های زیر را دارا می باشد:

۱. مدل‌سازی ریاضی پیشرفته: از روابط ریاضی دقیق برای توصیف قوانین صحنه نبرد و تعاملات بازیکنان استفاده می‌کند.
۲. تمرکز بر تیم‌های ضعیف‌تر: مربی هوشمند طراحی‌شده به طور خاص برای کمک به تیم‌های ضعیف‌تر در مواجهه با تیم‌های قوی‌تر بهینه شده است.
۳. استفاده از الگوریتم‌های یادگیری تقویتی: ترکیب الگوریتم‌های یادگیری تقویتی با مدل ریاضی به صورت کارآمد برای افزایش قابلیت تعمیم‌دهی و پایداری استفاده شده است.
٤. کاربرد عملی در سناریوهای پیچیده: برخلاف مطالعات پیشین، این روش بر روی محیط‌های پیچیده و پویا آزمایش شده است که نشان‌دهنده کارایی و قابلیت تعمیم‌دهی بالاتر آن است.

۲.۱. روش تحقیق یا اصول و تئوری مقاله

مدل‌سازی ریاضی این تحقیق مبتنی بر نظریه بازی‌های رقابتی دوطرفه است که با درنظر گرفتن سلامت، قدرت آتش و احتمال برخورد تیر تیم‌ها، تصمیمات بهینه را برای تیم ضعیف‌تر استخراج می‌کند.

در این بخش ابتدا الگوریتم های یادگیری تقویتی استفاده شده در این مقاله با پارامترهای متناسب برای بازی جنگ را شرح می

[1] Wang

[2] Cheng Wang

[Downloaded from ic4i-journal.ir on 2026-10-19]

دهیم و در ادامه یک مدل ریاضی پویا طراحی شده که در کنار این الگوریتم ها، تصمیم گیری مربی را در بازی بهبود می بخشند.

دلیل علمی انتخاب الگوریتم‌ها Q-Learning و DQN همانطور که در مرجع [۱۲] ذکر شده است به این دلیل است که الگوریتم Q-Learning به دلیل سادگی مفهومی، عدم نیاز به مدل محیط[۱] و اثبات همگرایی آن در مسائل گسسته به سیاست بهینه به‌عنوان پایه‌ای‌ترین روش یادگیری تقویتی انتخاب شد. این الگوریتم توانایی یادگیری سیاست بهینه فقط از طریق تجربه و بدون نیاز به مدل محیط را دارد، که در مسائل فرماندهی و کنترل نظامی با عدم قطعیت‌های بالا، یک مزیت کلیدی محسوب می‌شود. همچنین دلیل انتخاب الگوریتم DQN همانطور که در مرجع [۱۳] ذکر شده است به این دلیل است که با افزایش ابعاد فضای حالت، Q-Table سنتی ناکارآمد می‌شود الگوریتم DQN با استفاده از شبکه‌های عصبی عمیق برای تقریب مقدار تابع Q-Value ، امکان استفاده از Q-Learning در مسائل با فضای حالت پیوسته یا بسیار بزرگ را فراهم می‌سازد. برتری این الگوریتم‌ها نسبت به سایر روش‌هاعبارت است از:

- Q-Learning و DQN به‌عنوان الگوریتم های استاندارد یادگیری تقویتی شناخته می‌شوند که در اغلب مقالات مرجع حوزه RL نظامی برای شروع مدل‌سازی استفاده می‌شوند.
- الگوریتم‌های Actor-Critic و Policy Gradient با وجود مزیت در محیط‌های پیوسته، پیچیدگی محاسباتی و نیاز به تنظیمات دقیق‌تری دارند.[۸]
- روش‌های Evolutionary RL مانند CMA-ES نیاز به تعداد نمونه‌های بسیار بیشتری دارند و برای مسائل real-time کمتر مناسب هستند.[۵]

بعلاوه سایر الگوریتم‌ها کاستی‌هایی در کاربرد برای این تحقیق دارند، مثلا الگوریتم‌های مبتنی بر Policy Gradient در مسائل گسسته کوچک عملکرد مشابه Q-Learning دارند اما پیچیدگی پیاده‌سازی و حساسیت بالاتری به پارامترهای یادگیری دارند [۸]. همچنین الگوریتم‌های Model-based RL به دلیل نیاز به مدل محیط، برای سناریوهای با اطلاعات ناقص فعلی این تحقیق قابل استفاده نیستند.

در ادامه با توجه به نیاز به توابع و روابط این الگوریتم ها در مدلسازی، به صورت کاربردی، الگوریتم های استفاده شده را بیان می کنیم:

یکی از این الگوریتم ها، الگوریتم Q-Learning می باشد که یک الگوریتم پایه در یادگیری تقویتی است که هدف آن یافتن یک سیاست بهینه π^* است که مقدار پاداش کلی R را بیشینه کند. مفاهیم اصلی این الگوریتم به شرح زیر است:

۱. تابع پاداش

تابع $Q(s,a)$ مقدار پاداش مورد انتظار را از انجام اقدام $\boldsymbol{a}$ در حالت $\boldsymbol{s}$ نشان می‌دهد.

فرمول به‌روزرسانی $\boldsymbol{Q}$ به صورت زیر است:

$$Q^{new}(S_t,A_t) \leftarrow (1-\alpha)\cdot Q\,(S_t,A_t) + \alpha \cdot \left(R_{t+1} + \gamma \cdot \max_a Q\,(S_{t+1},a)\right) \quad (1)$$

که در آن

- α نرخ یادگیری
- γ ضریب تخفیف[۲] برای اهمیت پاداش‌های آینده.
- R_{t+1} پاداش لحظه ای مربوط به رفتن از حالت S_t به حالت S_{t+1}
- $\max_a Q\,(S_{t+1},a)$ ماکزیمم پاداشی که می تواند در حالت $\boldsymbol{S_{t+1}}$ به دست بیاید.

۲. الگوریتم:

[۱] model-free

[۲] Discount Factor

[Downloaded from ic4i-journal.ir on 2026-10-19]

- محیط شبیه‌سازی می‌شود.
- عامل به‌صورت آزمون و خطا اقدام می‌کند.
- بر اساس پاداش دریافتی، مقدار **Q** برای حالت و اقدام به‌روز می‌شود.

۳. مزایا:

- پیاده‌سازی ساده.
- مناسب برای محیط‌های گسسته.

٤. معایب:

- نیاز به جدول **Q** بزرگ برای محیط‌های پیچیده.
- کارایی پایین در محیط‌های پیوسته یا با فضای حالت بزرگ.

برای حل محدودیت Q-Learning در محیط‌های پیچیده با فضای حالت بزرگ، الگوریتم Deep Q-Network (DQN) معرفی شد که از شبکه‌های عصبی برای تقریب تابع **Q** استفاده می‌کند. در DQN، یک شبکه عصبی عمیق با ورودی $\boldsymbol{s}$ (حالت) و خروجی $Q(s,a)$ برای هر اقدام a طراحی می‌شود:

- ورودی: بردار ویژگی‌ها که نمایانگر حالت s در محیط است.
- خروجی: مقدار Q مربوط به تمام اقدامات ممکن $\boldsymbol{a}$ در آن حالت می باشد.

معمولاً از چند لایه مخفی[۱] با توابع فعال‌سازی مانند $ReLU(x) = max(۰, x)$ استفاده می شود.

۱. تابع هدف[۲]

شبکه عصبی باید طوری آموزش داده شود که مقادیر $Q(s,a)$ پیش‌بینی‌شده نزدیک به مقادیر واقعی باشد. مقدار واقعی با استفاده از معادله Bellman محاسبه می‌شود:

$$y = r + \gamma \max_{a'} Q(s', a';\ \theta^-) \qquad (۲)$$

که در آن:

- r پاداش لحظه‌ای.
- γ ضریب تخفیف برای اهمیت پاداش‌های آینده.
- θ^- وزن‌های شبکه هدف[۳] که به‌صورت دوره‌ای به‌روزرسانی می‌شوند.

ضریب تخفیف γ نقش کلیدی در یادگیری تقویتی دارد. این پارامتر در بازه (۰,۱) تعریف می‌شود و اهمیت پاداش‌های آینده را نسبت به پاداش‌های فعلی تعیین می‌کند. مقدار γ نزدیک به ۰ به این معنی است که عامل فقط پاداش‌های لحظه‌ای را مهم می‌داند، در حالی‌که γ نزدیک به ۱ باعث می‌شود پاداش‌های آینده ارزش بیشتری داشته باشند.

تابع هدف برای مینیمم کردن اختلاف بین $\boldsymbol{y}$ (هدف) و خروجی شبکه $Q(s,a;\theta)$ به صورت زیر تعریف می‌شود:

$$L(\theta) = E\left[\left(y - Q(s,a;\theta)\right)^۲\right] \qquad (۳)$$

که در عمل، این امید ریاضی با استفاده از داده‌های نمونه‌گیری شده از Replay Buffer (تجربه قبلی) تقریب زده می‌شود.

۲. گرادیان کاهشی[٤]

برای بهینه‌سازی وزن‌های شبکه یعنی θ، از الگوریتم گرادیان کاهشی استفاده می‌شود. مشتق تابع هدف نسبت به وزن‌ها θ به صورت زیر محاسبه می‌شود:

$$\nabla_\theta L(\theta) = E\left[\nabla_\theta Q(s,a;\theta)\left(y - Q(s,a;\theta)\right)\right] \qquad (٤)$$

که در آن:

- $\nabla_\theta Q(s,a;\theta)$ گرادیان خروجی شبکه نسبت به وزن‌ها که از طریق روش backpropagation محاسبه می‌شود.
- وزن‌ها با استفاده از رابطه زیر به‌روز می‌شوند:

$$\theta \leftarrow \theta - \alpha \nabla_\theta L(\theta)$$

و $\boldsymbol{\alpha}$ نرخ یادگیری می باشد.

[۱] Hidden Layers
[۲] Loss Function
[۳] Target Network
[۴] Gradient Descent

[Downloaded from ic4i-journal.ir on 2026-10-19]

۳. Replay Buffer و بهبود پایداری

- برای کاهش همبستگی بین نمونه‌ها، می توان تجربه‌های عامل[1] شامل (s,r,a,s') در حافظه Replay Buffer ذخیره کرد.
- به‌جای استفاده از نمونه‌های متوالی، از نمونه‌برداری تصادفی[2] برای انتخاب داده‌های آموزش استفاده می‌شود.

٤. شبکه هدف[3]

- برای جلوگیری از نوسانات و همگرایی بهتر، DQN از یک شبکه هدف جداگانه استفاده می‌کند. وزن‌های این شبکه θ^- به‌صورت دوره‌ای با وزن‌های شبکه اصلی $\boldsymbol{\theta}$ همگام‌سازی می‌شوند:

$$\theta \leftarrow \theta^-$$

٥. مزایا و ریاضیات پیشرفته‌تر

- با استفاده از توابع فعال‌سازی غیرخطی (مانند ReLU)، شبکه عصبی می‌تواند روابط پیچیده بین حالات و اقدامات را مدل‌سازی کند.
- استفاده از ماتریس‌های ژاکوبی یا هسی برای بررسی دقیق‌تر تأثیر وزن‌ها روی خروجی در برخی نسخه‌های پیشرفته‌تر DQN (مانند Double DQN) مطرح است.

در جدول ۱، مقایسه الگوریتم‌های Q-Learning و DQN آورده شده است:

جدول ۱: مقایسه الگوریتم ها.

ویژگی	Q-Learning	DQN
نوع فضای حالت	گسسته	گسسته یا پیوسته
ذخیره Q	جدول	شبکه عصبی
مقیاس پذیری	محدود	بالا
نیاز به حافظه	کم	زیاد

۲. مدلسازی پیشرفته ریاضی و ترکیب آن با یادگیری تقویتی

برای مدل‌سازی ریاضی مسئله اصلی، هدف ما ترکیب یادگیری تقویتی با مدل‌های صحنه نبرد به‌گونه‌ای است که یک طرف ضعیف‌تر بتواند راهکارهای خود را با کمک مربی هوشمند بهینه کرده و در نهایت پیروز شود. این مدل‌سازی شامل موارد زیر است:

- در اینجا، حالت‌ها شامل موقعیت مکانی بازیکنان، میزان تیرها، و سلامت بازیکنان هستند.
- اقدامات شامل تیراندازی به یک هدف مشخص یا دفاع است.
- شبکه عصبی به‌طور خودکار بهترین اقدام‌ها را در هر حالت با استفاده از این ریاضیات یاد می‌گیرد:

۱.۲. تعریف متغیرها و پارامترها

- **تیم‌ها:**
  - $\boldsymbol{T_1}$ تیم قوی‌تر (دارای منابع بیشتر، دقت تیراندازی بالاتر).
  - $\boldsymbol{T_2}$ تیم ضعیف‌تر (دارای مربی یادگیری تقویتی).
- **پارامترهای بازیکنان:**
  - $\boldsymbol{S_i}$ میزان سلامت بازیکن $\boldsymbol{i}$
  - $\boldsymbol{B_i}$ تعداد گلوله‌های باقی‌مانده بازیکن $\boldsymbol{i}$
  - $\boldsymbol{A_i}$ دقت تیراندازی بازیکن $\boldsymbol{i}$ به صورت توزیع نرمال

$$\boldsymbol{A_i \sim N(\mu_i, \sigma_i^2)}$$

- **موقعیت‌ها و موانع:**
  - $\boldsymbol{P_i}$ موقعیت بازیکن $\boldsymbol{i}$ در صفحه شبیه‌سازی.
  - $\boldsymbol{O}$ مجموعه‌ای از موانع (مانند سنگرها) که احتمال برخورد تیر را تغییر می‌دهند.
- **استراتژی مربی:**

[1] Agent

[2] Random Sampling

[3] Target Network

[Downloaded from ic4i-journal.ir on 2026-10-19]

- $\pi(s,a)$ استراتژی بهینه مربی که بر اساس حالت s و اقدام a عمل می‌کند.
- r_t پاداش در گام زمانی t

۲.۲. تعریف مدل پویا

ابتدا تابع توزیع تجمعی نرمال $\Phi(x)$ را تعریف می کنیم که در حل مسائل صحنه نبرد و دقت تیراندازی بسیار به ما کمک می کند. تابع توزیع تجمعی نرمال $\Phi(x)$ به صورت ریاضی احتمال اینکه یک متغیر تصادفی نرمال مقدار کمتری از x داشته باشد، را نشان می‌دهد. به عبارت دیگر، $\Phi(x)$ مقدار تجمعی تابع چگالی احتمال توزیع نرمال استاندارد از منفی بی‌نهایت تا x است. که رابطه ی آن به صورت زیر می باشد:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{t^2}{2}}dt \qquad (5)$$

که در آن

- x مقدار ورودی
- $\frac{1}{\sqrt{2\pi}}$ ضریب نرمال‌سازی برای توزیع نرمال استاندارد
- $e^{-\frac{t^2}{2}}$ تابع چگالی احتمال توزیع نرمال استاندارد که دارای ویژگی های زیر است:

۱. محدوده مقادیر $\Phi(x)$ همیشه مقداری بین ۰ و ۱ است.

- برای $x \to -\infty$ داریم $\Phi(x) \to 0$
- برای $x \to +\infty$ داریم $\Phi(x) \to 1$

۲. تقارن: تابع تجمعی برای توزیع نرمال استاندارد متقارن است، یعنی

$$\Phi(-x) = 1 - \Phi(x)$$

۳. رابطه با توزیع نرمال کلی: اگر متغیر تصادفی $X \sim N(\mu, \sigma^2)$ باشد، تابع تجمعی آن به صورت زیر است:

$$F_X(x) = \Phi\left(\frac{x-\mu}{\sigma}\right)$$

به دلیل اینکه $\Phi(x)$ به صورت انتگرال تعریف شده است، محاسبه دقیق آن نیاز به استفاده از الگوریتم‌های عددی یا تقریبی مانند بسط سری تیلور برای تقریب، دارد. تابع توزیع تجمعی نرمال در زمینه تیراندازی به احتمال خطا یا دقت مرتبط می‌شود، زیرا دقت تیراندازی معمولاً به عوامل تصادفی بستگی دارد و این عوامل اغلب به صورت نرمال توزیع شده‌اند. در ادامه مرحله ی دوم مدلسازی، روابط زیر را داریم:

۱. قوانین تیراندازی: اگر تیرانداز بخواهد هدف را در محدوده مشخصی بزند (برای مثال، دایره‌ای با شعاع مشخص r این احتمال را می‌توان با تابع تجمعی توزیع نرمال محاسبه کرد:

$$P_{hit} = \Phi(\frac{r-\mu}{\sigma})$$

که در آن

- r شعاع هدف یا حداکثر خطای مجاز.
- μ فاصله میانگین از مرکز هدف.
- σ پراکندگی خطا.

به عنوان مثال برای محاسبه احتمال برخورد برای یک تیر در تیم قوی تر داریم:

$$P_{hit} = \Phi\left(\frac{r-\mu}{\sigma}\right) = \Phi\left(\frac{۰-۱}{۰.۵}\right) = \Phi(۲)$$

از جداول توزیع نرمال استاندارد داریم:

$$\Phi(۲) \approx ۰.۹۷۷۲$$

پس احتمال برخورد برای یک تیر حدود ۹۷.۷۲٪ است. محاسبه احتمال برخورد حداقل یک تیر از بین ۱۰ تیر را نیز می توان به صورت زیر محاسبه کرد:

$$P(Hit\ at\ least\ one) = ۱ - (۱ - P_{hit})^{۱۰}$$
$$= ۱ - (۱ - ۰.۹۷۷۲)^{۱۰} \approx ۱ - ٦.٧٩ \times ۱۰^{-۲۰} \approx ۱$$

این نشان می‌دهد که تقریباً قطعاً یکی از این ۱۰ تیر به هدف برخورد خواهد کرد. همین روند را برای تیم ضعیف تر نیز می توان داشت: انحراف معیار σ برای تیم ضعیف‌تر به‌طور فرضی انتخاب شده است تا نشان‌دهنده دقت کمتر آن‌ها در تیراندازی باشد. در مدل‌های واقعی، این مقدار را می‌توان از روابط ریاضی و آماری محاسبه کرد.

$$P_{hit} = \Phi\left(\frac{r-\mu}{\sigma}\right) = \Phi\left(\frac{۰-۱}{۱.۵}\right) = \Phi(۰.٦٧)$$

از جداول توزیع نرمال استاندارد داریم:

$$\Phi(۰.٦٧) \approx ۰.٧٤٨٦$$

[Downloaded from ic4i-journal.ir on 2026-10-19]

پس احتمال برخورد برای یک تیر در تیم ضعیف تر حدود ٪۷٤.۸٦ است.

توجه: این مقادیر فرضی بوده و در دنیای واقعی P_{hit} به پارامترهای بسیار زیادی بستگی دارد و محاسبات پیچیده ای نیز نیاز می باشد تا آن را حساب کرد.

۲. پویایی سلامت: تغییر سلامت بازیکن S_i در اثر برخورد یک تیر با قدرت k به صورت زیر فرموله می شود:

$$S_i^{(t+1)} = S_i^{(t)} - k \cdot P_{hit} \cdot \Delta S$$

۳. پاداش مربی: پاداش بر اساس تعداد بازیکنان باقی‌مانده، تعداد تیرهای شلیک شده و اهداف برخورد کرده محاسبه می‌شود:

$$r_t = \alpha \cdot survival\,rate + \beta \cdot accuracy - \gamma \cdot resource\,usage$$

٤. تابع ارزش: استراتژی مربی بر اساس مقدار بهینه تابع ارزش $V(s)$

برای این قسمت از معادله بهینه‌سازی بلمن[1] که در یادگیری تقویتی استفاده می‌شود و اساس بسیاری از الگوریتم‌های یادگیری تقویتی است استفاده می کنیم:

$$V(s) = \max_a \left[\gamma \cdot E_{s'}\big(V(s')\big) + r(s,a)\right]$$

که در آن:

- $V(s)$ تابع ارزش[2]: مقدار مورد انتظار پاداش کلی که عامل[3] می‌تواند از حالت s دریافت کند، در صورتی که بهترین اقدامات ممکن را انتخاب کند.
- $r(s,a)$ یا پاداش لحظه‌ای: مقدار پاداشی که عامل با انجام اقدام a در حالت s دریافت می‌کند.
- γ یا ضریب تخفیف[4]: اهمیت پاداش‌های آینده نسبت به پاداش فعلی. مقدار آن بین ۰ و ۱ است:
  - $\gamma = ۰$ تنها پاداش لحظه‌ای مهم است.
  - $\gamma \to ۱$ پاداش‌های آینده بسیار اهمیت دارند.
- $E_{s'}\big(V(s')\big)$ امید ریاضی مقدار ارزش $V(s')$ برای حالت بعدی s' که با اجرای a حاصل می‌شود. این مقدار متوسط پاداش مورد انتظار در آینده را توصیف می‌کند. این معادله می‌گوید ارزش یک حالت s برابر است با حداکثر مقدار پاداشی که می‌توان از این حالت با انتخاب بهترین اقدام و ادامه اجرای بهترین سیاست‌ها در آینده به دست آورد. عامل ابتدا پاداش لحظه‌ای $r(s,a)$ را می‌گیرد و سپس ارزش حالت‌های آینده $V(s')$ را با ضریب تخفیف γ به پاداش فعلی اضافه می‌کند.

نکته۱: بهترین عمل تیم ضعیف‌تر و تحلیل آنچه یاد می‌گیرد:

تیم ضعیف‌تر از طریق یادگیری تقویتی در حال یادگیری یک استراتژی بهینه برای زنده ماندن و افزایش احتمال پیروزی است. این یادگیری بر اساس دریافت پاداش‌ها و به‌روزرسانی تابع ارزش[5] و تابع Q انجام می‌شود.

در مدل ما، پاداش‌ها برای اقدامات مختلف به تیم ضعیف‌تر به شرح زیر داده می‌شوند:

- پاداش مثبت برای ضربه زدن به دشمن
- اگر تیم ضعیف‌تر به یکی از بازیکنان تیم قوی‌تر شلیک کند و موفق شود، پاداش مثبت دریافت می‌کند. (بازیکن قوی تر: در این شبیه‌سازی، تیم قوی‌تر بازیکنانی هستند که دقت تیراندازی بالاتر، سلامت اولیه بیشتر و احتمال برخورد بیشتری دارند. این تیم معمولاً در ابتدای بازی برتری دارد.)
- در صورتی که تا پایان اپیزود بازیکنی از تیم ضعیف‌تر زنده بماند، پاداش نهایی ۱۰+ دریافت می‌شود.

نکته۲: تیم ضعیف‌تر چه چیزی را یاد می‌گیرد؟

[1] Bellman Optimality Equation

[2] Value Function

[3] Agent

[4] Discount Factor

[5] Value Function

[Downloaded from ic4i-journal.ir on 2026-10-19]

بر اساس دریافت پاداش‌ها، تیم ضعیف‌تر یاد می‌گیرد که:

- انتخاب هدف بهینه:
  - بهتر است بازیکنی از تیم قوی‌تر را هدف بگیرد که کمترین سلامت را دارد، زیرا شانس حذف او بیشتر است.
  - تیم ضعیف‌تر یاد می‌گیرد که به بازیکنانی شلیک کند که احتمال برخورد بالاتر دارند.
- مدیریت مهمات:
  - یاد می‌گیرد که بی‌هدف شلیک نکند و در زمان‌های حساس‌تر از گلوله‌هایش استفاده کند.

۳.۲. مدل یادگیری تقویتی برای مربی

مربی از طریق آزمون و خطا استراتژی خود را بهینه می‌کند و با هر بازی، عملکرد تیم ضعیف‌تر بهتر می‌شود. برای این منظور از ترکیب الگوریتم های یادگیری تقویتی *Q-Learning*و *Deep Q-Network (DQN)* استفاده می کنیم.

تفسیر عملیاتی پاداش برای تیم ضعیف‌تر را می توان به صورت زیر بررسی کرد:

(الف) پاداش برای آسیب زدن به دشمن

- این پاداش نشان‌دهنده اثربخشی حملات تیم ۲ است.
- در عملیات نظامی، این می‌تواند معادل با موارد زیر باشد:
  - کاهش توان رزمی دشمن مثلاً نابود کردن تجهیزات یا کاهش توان نیروی انسانی دشمن
  - برتری تاکتیکی موقت مانند مجبور کردن دشمن به عقب‌نشینی
  - افزایش روحیه و انگیزه نیروهای تیم ۲ زیرا می‌بینند که حملاتشان اثرگذار است

(ب) پاداش بزرگ در صورت پیروزی نهایی

- این پاداش ۱۰ امتیازی در صورت پایان بازی، نشانه‌ای از دستاورد استراتژیک تیم ۲ است.
- در میدان نبرد، این می‌تواند موارد زیر را نشان دهد:
  - دستیابی به برتری عملیاتی مثلاً کنترل یک منطقه کلیدی
  - زنده ماندن و بقای نیروها یعنی کاهش تلفات نیروهای تیم ۲
  - کسب تجربه و یادگیری تاکتیکی چراکه توانسته‌اند در مقابل نیروی قوی‌تر مقاومت کنند و پیروز شوند

۴.۲. هدف و شرایط خاتمه

- هدف تیم T_2 کاهش تعداد بازیکنان تیم T_1 و حفظ حداکثر بازیکنان خود.
- شرط خاتمه: زمانی که یکی از تیم‌ها تمام بازیکنان خود را از دست بدهد یا تعداد دورهای از پیش تعیین‌شده به پایان برسد.

۳. پیاده سازی و نتایج

ساختار کلی پیاده سازی از بخش های زیر تشکیل شده است:

۱. **تعریف محیط:**

– محیط شامل موقعیت بازیکنان، مقاومت سنگرها، تعداد گلوله‌ها و اقدامات ممکن است.

– حالات (s) و اقدامات (a) مشخص می‌شوند.

۲. **یادگیری مربی:**

استفاده از الگوریتم یادگیری تقویتی برای به‌روزرسانی ارزش اقدامات $Q(s,a)$

۳. **بازی:**

بازی از چند مرحله تشکیل شده و مربی تیم ضعیف‌تر یاد می‌گیرد چگونه استراتژی‌ها را بهبود دهد.

جدول ۲ به تفصیل پارامترها و مراحل دقیق‌تر را با آستانه‌هایی برای تیم ضعیف مشخص می‌کند تا روند کد

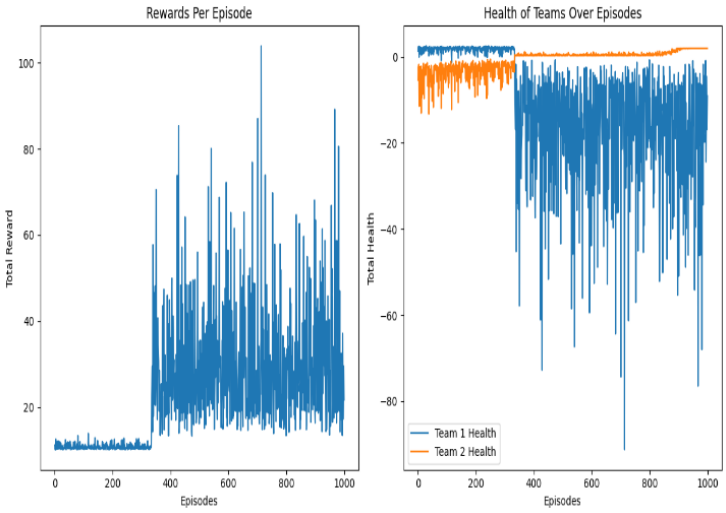
[Downloaded from ic4i-journal.ir on 2026-10-19]

شفاف‌تر شود. این آستانه‌ها می‌توانند برای تحلیل بهتر بازی و کنترل رفتار تیم ضعیف مفید باشند.

جدول ۲. پارامترها.

گام	عملیات	پارامترهای مرتبط	توضیحات آستانه‌ها
۱. مقداردهی اولیه	تعریف تیم‌ها و پارامتر های محیط	team1_health, team2_health, team2_bullets	تیم ۱ قوی‌تر؛ سلامت 1.0 تیم ۲ ضعیف‌تر؛ سلامت 0.5.
۲. مقداردهی اولیه	تعریف تیم‌ها و پارامتر های محیط	تعداد نیروها	$N_1 = 10$ $N_2 = 10$
۳. تعریف محیط بازی	ایجاد کلاس WarGameEnvironment و تعریف توابع reset و step	damage_to_team1, damage_to_team2, healing_rate	ضریب آسیب تیم ۲ مقداری بین [0.2,0.5] ضریب آسیب تیم ۱ مقداری بین [1.0,2.5]
۴. تعریف مربی	ایجاد کلاس QLearningAgent، مقداردهی اولیه Q-Table, و ε	learning_rate, discount_factor, exploration_rate	نرخ یادگیری $\alpha = 0.1$ نرخ اکتشاف ۰.۹۹ تا حداقل ۰.۱
۵. شروع هر اپیزود	محیط بازی ریست شده و متغیرهای بازی مقداردهی می‌شوند	n_episodes, episode	تعداد اپیزودها برابر 1000 متغیرهای سلامت تیم‌ها و گلوله‌ها تنظیم می‌شوند.
۶. حلقه بازی	اجرای بازی تا زمانی که یکی از تیم‌ها شکست بخورد.	state, action, reward, done	اجرای حلقه تا پایان تیرها یا اتمام سلامت نیروها (می توان آستانه تعریف کرد)
۶/۱. انتخاب اقدام	مربی تیم ۲ (ضعیف‌تر) اقدام مناسب را بر اساس سیاست اکتشاف یا بهره‌برداری انتخاب می‌کند.	state, exploration_rate, n_actions	انتخاب تصادفی (اکتشاف). در غیر این صورت استفاده از مقادیر Q بهره‌برداری
۶/۲. اجرای اقدام	اقدام انتخابی تیم ۲ در محیط اجرا شده و وضعیت بازی به‌روزرسانی می‌شود.	damage_to_team1, target_team1, target_team2, healing_rate	تیم ۲ به تیم ۱ شلیک می‌کند (تصادفی، قدرت محدود). تیم ۱ پاسخ می‌دهد (قدرت بالاتر)
۶/۳. بهبود تیم ضعیف	تیم ۲ در اواخر بازی سلامت خود را بازیابی می‌کند.	healing_rate, healing_threshold	سلامت تیم ۲ با نرخ ثابتی بهبود می یابد
۶/۴. به‌روزرسانی مربی	مربی تیم ضعیف Q-Table را بر اساس پاداش دریافتی و وضعیت‌های جدید به‌روزرسانی می‌کند.	q_values, learning_rate, discount_factor	بروزرسانی مقدار Q
۷. ذخیره اطلاعات آماری	ثبت مجموع پاداش‌ها و سلامت تیم‌ها برای هر اپیزود	total_reward, health_team1, health_team2	مجموع پاداش‌ها و سلامت تیم‌ها ذخیره می‌شود تا روند پیشرفت تیم‌ها بررسی شود.
۸. کاهش اکتشاف	کاهش نرخ اکتشاف ε برای تمرکز بیشتر روی بهره‌برداری.	exploration_rate	بهبود مقدار ε
۹. رسم نمودارها	نمایش تغییرات پاداش‌ها و سلامت تیم‌ها در طول اپیزودها	rewards_per_episode, health_team1, health_team2	رسم نمودار و تحلیل تغییر سلامت تیم ها

نتایج خروجی این مدل پیاده سازی شده در ادامه در شکل ۱ آورده شده است. تمامی شبیه‌سازی‌ها در محیط Python ۳،۱۰ با استفاده از کتابخانه‌های Numpy و Matplotlib انجام شده است. کدنویسی در محیط Jupyter Notebook روی سیستم عامل Windows ۱۰ با پردازنده Core i۷ و ۱۶ GB RAM پیاده‌سازی شده است. کدها برای تکرارپذیری کامل در دسترس هستند برای دستیابی به یافته‌ها، الگوریتم Q-Learning طی ۱۰۰۰ اپیزود شبیه‌سازی شد و مجموع پاداش‌های دریافتی تیم ضعیف‌تر در هر اپیزود به عنوان معیار عملکرد مربی در نظر گرفته شد. همچنین مجموع سلامت تیم قوی‌تر و ضعیف‌تر در پایان هر اپیزود به‌عنوان شاخص قدرت نسبی تیم‌ها ثبت گردید . روند تغییرات این شاخص‌ها از طریق رسم نمودار پاداش (Total Reward) و سلامت تیم‌ها (Team Health) بر حسب اپیزود تحلیل شد. همان‌گونه که در شکل ۱ نتایج مشاهده می‌شود، حدود اپیزود ۴۰۰ یک تغییر روند مثبت در مجموع پاداش‌ها رخ می‌دهد .دلیل این تغییر، کاهش تدریجی نرخ اکتشاف (exploration rate, ε) به زیر مقدار ۰.۲ در آن مقطع است که باعث شده مربی بیشتر به بهره‌برداری از Q-Table آموزش‌دیده بپردازد و اقدامات بهینه‌تری انتخاب کند. این تغییر نشان‌دهنده آغاز تثبیت یادگیری مربی هوشمند و بهبود سیاست تصمیم‌گیری تیم ضعیف‌تر است. برای اطمینان از تکرارپذیری نتایج، آزمایش‌ها با خوراک اولیه[1] ثابت در توابع رندوم انجام شده است. همچنین تعداد اپیزودها، اندازه تیم‌ها، و پارامترهای الگوریتم در کد به‌صورت قابل تنظیم طراحی شده‌اند.



شکل ۱. نتایج خروجی مدل.

٤. پیشنهادات

۱. رفع مشکل عدم وجود همکاری بین بازیکنان تیم:

- هر بازیکن تیم ۱ به صورت مستقل عمل می‌کند و هیچ استراتژی گروهی یا هماهنگی وجود ندارد.

۲. هدف‌گیری تصادفی:

- تیم ۱ بدون توجه به وضعیت سلامت یا استراتژی تیم ۲، اهداف را تصادفی انتخاب می‌کند. این

[1] seed

[Downloaded from ic4i-journal.ir on 2026-10-19]

منجر به رفتار غیرهوشمندانه می‌شود که برای اصلاح آن می توان بازیکنان تیم ۱ اهداف مشترکی را دنبال کنند و تلاش کنند یک بازیکن خاص از تیم ۲ را از بازی خارج کنند.

۳. نداشتن استراتژی دفاعی یا تاکتیکی:

- تیم ۱ فقط حمله می‌کند و هیچ رویکرد تدافعی یا تطبیقی ندارد که برای بهبود آن می توان تیم ۱ از استراتژی‌های دفاعی استفاده کند، مثلاً:
- بازیکنانی که سلامت کمتری دارند، از شلیک اجتناب کنند و به حالت دفاعی بروند.
- تیم ۱ می‌تواند به جای حمله تصادفی، بر بازیکنان خاص تیم ۲ تمرکز کند که بیشترین تهدید را ایجاد می‌کنند.

٤. عدم تغییر رفتار با شرایط بازی:

- تیم ۱ به شرایط بازی (مثل کاهش سلامت خود یا افزایش قدرت تیم ۲) واکنش نشان نمی‌دهد که برای بهبود آن می توان تیم ۱ رفتار خود را بر اساس وضعیت تیم ۲ به صورت زیر تنظیم کند:
- اگر سلامت کلی تیم ۲ کمتر از یک مقدار آستانه شد، شدت حملات تیم ۱ افزایش یابد.
- اگر تعداد گلوله‌های تیم ۲ کاهش یابد، تیم ۱ حملات خود را بیشتر کند.

٥. اضافه کردن پیچیدگی به محیط بازی:

- اضافه کردن موانع محیطی
- مدل را می‌توان با اضافه کردن موانع فیزیکی مانند دیوارها یا سنگرها پیچیده‌تر کرد. این موانع می‌توانند بر مسیر شلیک‌ها و احتمال موفقیت تأثیر بگذارند.
- مدل‌سازی ریاضی: احتمال برخورد به موانع را می‌توان با استفاده از هندسه و نظریه ی احتمالاتی پیشرفته محاسبه کرد و بررسی تعاملات بازیکنان را در تحلیل دقیق‌تر از تعاملات بین آن ها و تأثیر تصمیمات آن‌ها در نظر گرفت.
- شبیه‌سازی در کد: استفاده از یک شبکه مشبک (Grid) برای تعریف مکان بازیکنان و موانع.
- افزودن زمان واکنش:
- بازیکنان می‌توانند یک تأخیر زمانی در شلیک یا دفاع داشته باشند که به واقع‌گرایی مدل کمک می‌کند.

٦. یادگیری تقویتی چند عامله[1]:

- مدلی که در آن هر بازیکن (تیم یا فرد) یک عامل مستقل است و با دیگر عوامل تعامل دارد. به عنوان مثال: تیم ۱ و تیم ۲ به‌صورت جداگانه استراتژی یاد می‌گیرند.

۷. تلفیق مدل پیشنهادی با الگوریتم های نظریه ی بازی ها جهت مدلسازی و شبیه سازی طیف وسیع تری از بازی ها در صحنه نبرد.

۸. در مدل فعلی، پاداش فقط به وارد کردن آسیب به تیم‌وابسته است، اما در دنیای واقعی، نیروهای ضعیف‌تر (مانند چریک‌ها یا نیروهای نامنظم) معمولاً برای بقا، استفاده بهینه از منابع، و ایجاد بیشترین آسیب با کمترین هزینه تلاش می‌کنند. با اصلاح سیستم پاداش به روش‌هایی مانند:

- دادن امتیاز برای بقا و مدیریت منابع
- تشویق تاکتیک‌های ضربه و فرار [2]
- افزایش پاداش درصورت جلوگیری از آسیب دیدن.

۵. مراجع

[1] Multi-Agent Reinforcement Learning (MARL)

[2] Hit and Run

[Downloaded from ic4i-journal.ir on 2026-10-19]

[۱] D. Huh and P. Mohapatra (۲۰۲٤), Multi-agent Reinforcement Learning: A Comprehensive Survey, arXiv:۲۳۱۲,۱۰۲٥٦v۲ [cs.MA] ۳ Jul.

[۲] B. Wang, X. Gao, T. Xie (۲۰۲٤), An evolutionary multi-agent reinforcement learning algorithm for multi-UAV air combat, Knowledge-Based Systems ۲۹۹,۱۱۲۰۰۰.

[۳] C. Wang and others (۲۰۲٤), Discovering Command and Control Channels Using Reinforcement Learning, conference.

[٤] A. Dabholkar and others (۲۰۲٤), Adversarial Attacks on Reinforcement Learning Agents for Command and Control, arXiv:۲٤۰٥,۰۱٦۹۳v۲ [cs.CR] ۱ Jul.

[٥] S. A. Sarcia, G. Colo, (۲۰۲۳), Organizing Structures and Information for Developing AI-enabled Military Decision-Making Systems, EEE International Workshop on Technologies for Defense and Security (TechDefense).

[٦] J. Boron, C. Darken (۲۰۲۰), Developing Combat Behavior through Reinforcement Learning in Wargames and Simulations, IEEE Conference on Games (CoG).

[۷] C. Rinaudo and others (۲۰۲٤), Applying Deep Reinforcement Learning to Train AI Agents in a Wargaming Framework, Published in SoutheastCon ,Computer Science.

[۸] J. Zhang, Q. Xue (۲۰۲۲), Actor–critic-based decision-making method for the artificial intelligence commander in tactical wargames, Journal of Defense Modeling and Simulation: Applications, Methodology, Technology.

[۹] Bin Li (۲۰۲۲), Hierarchical Architecture for Multi-Agent Reinforcement Learning in Intelligent Game, Published in IEEE International Joint Conference on Neural Networks (IJCNN).

[۱۰] J. Chao, J. Sato, C. Lucero, D. S. Lange (۲۰۲۱), Evaluating Reinforcement Learning Algorithms For Evolving Military Games, Computer Science.

[۱۱] V .G. Goecks and others (۲۰۲۱), On games and simulators as a platform for development of artificial intelligence for command and control, The Journal of Defense Modeling and Simulation.

[۱۲] C.J.C.H. Watkins and P. Dayan (۱۹۹۲), Machine learning, Springer.

[۱۳] V. Mnih and others (۲۰۱٥), Human-level control through deep reinforcement learning, Nature.

[Downloaded from ic4i-journal.ir on 2026-10-19]