Information Technology Innovations and Applied Communication
Vol. 4, No. 2, Spring, 2026, Serial No. 9,pp.83-94

Improving Multi-Layer Decision Making in Simultaneous Strategy Games with Improved Command and Optimization Algorithm Method and Tactical Adaptive Control

Sina samadi gharehveran*[1]

* Assistant Professor, University of Tabriz, Tabriz, Iran.

(**Received:** 27 March 2026**, Revised:** 14 May 2026, **Accepted:** 22 May 2026, **Published:** 03 June 2026)

Abstract

Real-time strategy (RTS) games, with their complex dynamics and requirement for simultaneous decisions, provide a rich testbed for artificial intelligence research. This paper proposes a hybrid model to enhance intelligent agent performance in RTS environments, combining a hierarchical structure for strategic decision-making with adaptive algorithms for tactical control. At the strategic level, a hierarchical task network evaluates possible paths for base and army development, considering environmental state, resources, and objectives. At the tactical level, battle management and unit deployment leverage Puppet Search and Monte Carlo Tree Search algorithms with adaptive control. An improved command and optimization method, extending the BOSS algorithm, integrates strategic hierarchies, tactical adaptive algorithms, and reactive control (ABCD and UCT-CD) to support multi-layer decision-making. Low-level control employs ABCD and adaptive Monte Carlo algorithms to manage unit movements precisely. The proposed model was evaluated in StarCraft scenarios and compared with BWSAL, BOSS, and micro-based models. Results indicate superior performance in execution speed, accuracy, and tactical coordination, demonstrating suitability for real-time applications. Beyond gaming, these findings offer insights for designing multi-agent decision-support systems in military simulations, educational platforms, and intelligent gaming environments.

Keywords: Multi-layer decision making, simultaneous strategy games, tactical adaptive control, hierarchy

* Corresponding Author E-mail: s.samadi@tabrizu.ac.ir

«نوآوری‌های فناوری اطلاعات و ارتباطات کاربردی»
سال چهارم، شماره ۲، فصل بهار ۱۴۰۵؛ ص ۹۴-۸۳.

بهبود تصمیم‌گیری چندلایه در بازی‌های راهبُردی هم‌زمان با روش الگوریتم بهبودیافته فرماندهی و بهینه‌سازی و کنترل تطبیقی تاکتیکی

سینا صمدی قره‌ورن*[۱]

۱- استادیار، دانشگاه تبریز، تبریز، ایران.

(دریافت: ۰۷ فروردین ۱۴۰۵ بازنگری: ۲۴ اردیبهشت ۱۴۰۵ پذیرش: ۰۱ خرداد ۱۴۰۵ انتشار: ۱۳ خرداد ۱۴۰۵)

چکیده

بازی‌های ویدئویی راهبُرد هم‌زمان، با ساختاری پیچیده و نیاز به تصمیم‌گیری هم‌زمان، به بستری ایدئال برای پژوهش در زمینه هوش مصنوعی تبدیل شده‌اند. در این مقاله، مدلی ترکیبی برای بهبود عملکرد عامل هوشمند در این بازی‌ها ارائه شده که شامل ساختاری سلسله‌مراتبی برای تصمیم‌گیری راهبردی و الگوریتم‌های تطبیقی برای کنترل تاکتیکی است. در سطح راهبردی از ساختار شبکه وظیفه‌ای سلسله‌مراتبی بهره گرفته شده که با درنظرگرفتن وضعیت محیط، منابع و هدف بازی، مسیرهای ممکن برای توسعه پایگاه و ارتش را تحلیل می‌کند. در سطح تاکتیکی، با استفاده از الگوریتم جستجوی عروسکی و الگوریتم درخت جستجوی مونت‌کارلو با کنترل تطبیقی، تصمیم‌گیری برای مدیریت نبردها و چینش نیروها صورت می‌گیرد. در این مقاله، روش پیشنهادی الگوریتم بهبودیافته فرماندهی و بهینه‌سازی ارائه شده است که نسخه‌ای بهبودیافته از الگوریتم BOSS برای تصمیم‌گیری چندلایه در محیط‌های راهبُرد زمان واقعی است. این روش با ترکیب ساختار سلسله‌مراتبی راهبردی، الگوریتم‌های تطبیقی تاکتیکی (Puppet Search)، و کنترل واکنشی مبتنی بر ABCD و UCT-CD طراحی شده است. همچنین، در سطح کنترل پایین، از الگوریتم‌هایی نظیر ABCD و الگوریتم درخت جستجوی مونت‌کارلو با کنترل تطبیقی D جهت مدیریت دقیق حرکات واحدها بهره گرفته شده است. برای ارزیابی مدل پیشنهادی، مجموعه‌ای از سناریوهای عملیاتی در محیط بازی StarCraft پیاده‌سازی و با سه الگوریتم مرجع (BWSAL ،BOSS و مدل‌های میکرو پایه) مقایسه شد. نتایج تجربی نشان می‌دهد که عامل پیشنهادی در مقایسه با مدل‌های موجود، عملکرد بهتری از نظر زمان، دقت و هماهنگی تاکتیکی داشته و درعین‌حال، با سرعت اجرای مناسب برای محیط‌های زمان واقعی همراه بوده است. یافته‌های این پژوهش می‌تواند در طراحی سامانه‌های تصمیم یار چندعامله در حوزه‌های نظامی، شبیه‌سازی آموزشی و بازی‌های هوشمند نیز به کار رود.

کلیدواژه‌ها: تصمیم‌گیری چندلایه، بازی‌های راهبُرد هم‌زمان، کنترل تطبیقی تاکتیکی، سلسله‌مراتبی

* رایانامه نویسنده مسئول: E-mail: s.samadi@tabrizu.ac.ir

۱. مقدمه

بازی‌های ویدئویی راهبُرد هم‌زمان[1] مانند StarCraft، به‌عنوان محیط‌هایی پیچیده و چندبعدی، بستری ایدئال برای توسعه و ارزیابی سامانه‌های هوش مصنوعی پیشرفته فراهم کرده‌اند. این بازی‌ها، برخلاف محیط‌های نوبتی مانند شطرنج، شامل تصمیم‌گیری‌های بلادرنگ، اطلاعات ناقص، چندعاملی بودن، و فضای حالت‌های بسیار بزرگ هستند [۱]. در چنین محیط‌های پیچیده و پویایی، عامل هوشمند باید قادر باشد وضعیت فعلی را به‌دقت تحلیل کرده، واکنش‌های احتمالی دشمن را پیش‌بینی نماید، منابع موجود را بهینه تخصیص دهد، و کنترل هم‌زمان صدها واحد را به‌صورت هماهنگ و یکپارچه انجام دهد [۲].

باوجود رشد چشمگیر الگوریتم‌های کنترل محلی مانند Alpha-Beta و الگوریتم درخت جستجوی مونت‌کارلو با کنترل تطبیقی در مدیریت درگیری‌ها و تصمیمات تاکتیکی، یکی از چالش‌های اساسی، عدم انسجام بین سطوح تصمیم‌گیری مختلف (راهبردی، تاکتیکی و عملیاتی) است [۳]. اکثر پژوهش‌های گذشته، تنها بر یکی از این سطوح تمرکز داشته‌اند و تعامل بین لایه‌ها را نادیده گرفته‌اند. برای مثال، مدل BOSS در بهینه‌سازی ساخت واحدها[2] موفق عمل کرده، اما فاقد قابلیت‌های تطبیق تاکتیکی در مواجهه با حملات ناگهانی است [۴]. همچنین الگوریتم BWSAL ، علی‌رغم قابلیت تولید مسیرهای ساخت اولیه، به دلیل ساختار خطی و عدم انطباق با وضعیت‌های پویا، کارایی کمتری در شرایط واقعی دارد [۵].

ازسوی‌دیگر، در حوزه کنترل تاکتیکی، الگوریتم‌هایی مانند الگوریتم درخت جستجوی مونت‌کارلو با کنترل تطبیقی D و جستجوی عروسکی توانسته‌اند راهکارهایی برای انتخاب واکنش مناسب در لحظه ارائه دهند، اما چون فاقد اتصال با اهداف کلان و برنامه‌ریزی راهبُردی هستند، تصمیماتشان گاهی در تعارض باهدف اصلی بازی قرار می‌گیرد [۶]. پژوهش‌های اخیر تلاش کرده‌اند تا با استفاده از شبکه‌های وظیفه‌ای سلسله‌مراتبی[3] ساختار بهتری برای برنامه‌ریزی سطح بالا ایجاد کنند، ولی این روش‌ها نیز عمدتاً در مرحله راهبردی متوقف مانده‌اند [۷].

در این مقاله مدلی سه‌لایه ارائه شده که هر یک از لایه‌های راهبردی، تاکتیکی و واکنشی را با ابزارهای مناسب پوشش داده و از طریق یکپارچه‌سازی تصمیم‌گیری بین لایه‌ها، انسجام رفتاری عامل را افزایش داده است. نوآوری اصلی این مدل در ترکیب ساختار سلسله‌مراتبی راهبردی با الگوریتم‌های تطبیقی تاکتیکی مانند جستجوی عروسکی و همچنین طراحی یک زیرسیستم واکنشی بر مبنای الگوریتم درخت جستجوی مونت‌کارلو با کنترل تطبیقی D و ABCD است. در این ساختار، لایه راهبردی وظیفه تعیین اهداف کلان (مانند توسعه یا حمله) را دارد، لایه تاکتیکی نحوه تحقق آن هدف را باتوجه‌به موقعیت میدان تعیین می‌کند، و لایه‌ی میکرو جزئیات اجرای آن را کنترل می‌کند [۸].

افزون بر آن، برای ارزیابی کیفیت این انسجام رفتاری، یک شاخص جدید با عنوان ثبات تصمیم‌گیری در این پژوهش معرفی شده است که عملکرد عامل را در طول زمان و در فازهای مختلف بازی (اوایل، میانه، و انتها) می‌سنجد. این معیار که در منابع گذشته سابقه‌ای ندارد، توانسته تفاوت عملکرد عامل پیشنهادی را به‌وضوح نسبت به مدل‌های مرجع نشان دهد. طبق نتایج، مدل پیشنهادی علاوه بر برتری عددی در زمان ساخت، درصد برد در نبردها، و سرعت اجرا، در حفظ انسجام رفتاری نیز عملکردی ممتاز از خود نشان داده است [۹].

در مجموع، مقاله حاضر تلاشی است در جهت پوشش هم‌زمان سه‌لایه کلیدی تصمیم‌گیری در بازی‌های راهبُرد هم‌زمان، با تأکید بر تعاملی‌سازی الگوریتم‌های کلاسیک و افزودن شاخص‌های نوین برای ارزیابی کیفیت عامل هوشمند. رویکرد پیشنهادی می‌تواند نه‌تنها در حوزه بازی، بلکه در محیط‌های نظامی شبیه‌سازی‌شده، آموزش چندعامله، و مدیریت سامانه‌های پیچیده بلادرنگ نیز مورداستفاده قرار گیرد. هدف اصلی این پژوهش، ارائه‌ی الگوریتم بهبودیافته فرماندهی و بهینه‌سازی به‌عنوان نسخه‌ی توسعه‌یافته‌ای از روش BOSS است که از طریق یکپارچه‌سازی تصمیم‌گیری در سطوح راهبردی، تاکتیکی و واکنشی، به بهبود انسجام رفتاری عامل و افزایش نرخ موفقیت در محیط‌های RTS منجر می‌شود.

این مقاله مروری به طور خاص بر تقاطع سه حوزه پیشرفته و حیاتی در مدیریت انرژی سامانه‌های چند – میکروگرید (MMG EMS) تمرکز دارد: ۱)مدل‌سازی پیشرفته عدم قطعیت، ۲) بهینه‌سازی چند - هدفه (Multi-Objective Optimization)، و ۳) رویکردهای دورگه‌ای و داده‌محور (Hybrid/Data-Driven). نوآوری اصلی و مشارکت‌های متمایز این پژوهش مروری نسبت به مطالعات پیشین (مانند مراجع [X, Y, Z] که مرورهای عمومی‌تری بوده‌اند)، در موارد زیر خلاصه می‌شود:

۱. چارچوب مفهومی یکپارچه و به‌هم‌پیوسته: ما یک چارچوب تحلیلی نوآورانه را ارائه می‌دهیم که برای اولین‌بار، نقش متقابل مدل‌سازی دقیق عدم قطعیت (به‌ویژه در سناریوهای تصادفی و توزیعی - مقاوم) و ضرورت استفاده از بهینه‌سازی چند - هدفه برای دستیابی به سازش‌های پارتو (Pareto Trade-offs) بین اهداف متعارض (هزینه، انتشار، قابلیت اطمینان) را به‌صورت هم‌زمان تحلیل می‌کند.

[1] Real-Time Strategy
[2] Build-Order
[3] Hierarchical Task Networks

۲. به‌روزرسانی متمرکز بر روندهای نوین: با جمع‌آوری و تحلیل انتقادی ۵۰ مقاله علمی جدید (۲۰۲۴-۲۰۲۰)، این مرور بر روی روندهای تحقیقاتی آتی، به‌ویژه ظهور مدل‌های EMS دورگه‌ای مبتنی بر یادگیری ماشین، تمرکز دارد. این مدل‌ها به طور مؤثر عدم قطعیت را مدیریت کرده و درعین‌حال، چالش‌های سرعت و دقت محاسباتی را برطرف می‌سازند.

۳. تحلیل مقایسه‌ای عملکردی (Functional Comparative Synthesis): مهم‌تر از همه، این پژوهش یک مقایسه کیفی و کمی (Quantitative) از عملکرد مدل‌های نوظهور در MMG EMS ارائه می‌دهد. این مقایسه با استفاده از معیارهای کلیدی گزارش‌شده (مانند میزان کاهش هزینه عملیاتی، بهبود ضریب پایداری/قابلیت اطمینان شبکه و درصد کاهش آلاینده‌ها) انجام می‌شود. این رویکرد تحلیلی، خواننده را قادر می‌سازد تا اثربخشی واقعی و مزیت رقابتی روش‌های جدید را در مقایسه با الگوریتم‌ها و نسخه‌های پیشین EMS به‌وضوح درک نماید.

۲. روش تحقیق

در این پژوهش، یک عامل هوشمند سه‌لایه طراحی و پیاده‌سازی شده است که توانایی تصمیم‌گیری در سطوح مختلف از کلان‌ترین راهبردها تا دقیق‌ترین اقدامات یگان‌ها را داراست.

یکی از رویکردهای کلیدی در برنامه‌ریزی و تصمیم‌گیری چندلایه، شبکه‌ی وظیفه‌ای سلسله‌مراتبی است. در این روش، وظایف پیچیده به مجموعه‌ای از زیر وظایف ساده‌تر تقسیم می‌شوند تا امکان تحلیل، زمان‌بندی و کنترل مؤثرتر فراهم شود. HTN به‌ویژه در حوزه‌ی بازی‌های راهبُرد زمان واقعی و سامانه‌های چندعاملی، به دلیل توانایی در سازمان‌دهی اهداف در سطوح راهبردی، تاکتیکی و اجرایی، کاربرد گسترده‌ای دارد. با استفاده از این چارچوب، می‌توان رفتار عامل را از سطح تصمیم‌گیری کلان تا سطح کنترل خرد به‌صورت منسجم و قابل‌تطبیق مدل‌سازی کرد. به همین دلیل، در مدل پیشنهادی این پژوهش نیز ساختار سلسله‌مراتبی مبتنی بر HTN به‌عنوان مبنای طراحی لایه‌های راهبردی و تاکتیکی در نظر گرفته شده است.

طراحی این ساختار، بر مبنای ایده سلسله‌مراتبی تصمیم‌گیری در سامانه‌های نظامی، الگوریتم‌های یادگیری تطبیقی و تحلیل رفتار انسانی در بازی‌های راهبُرد هم‌زمان صورت‌گرفته است. عامل پیشنهادی شامل سه‌لایه عملکردی اصلی است:

۲-۱. کارهای پیشین مرتبط و معرفی روش BOSS

در زمینه تصمیم‌گیری چندلایه در سامانه‌های راهبُردی، الگوریتم‌های سلسله‌مراتبی نقش مهمی ایفا می‌کنند. یکی از روش‌های پایه و شناخته‌شده در این حوزه، الگوریتم BOSS (الگوریتم پایه فرماندهی و بهینه‌سازی) است که به دلیل ساختار سه‌لایه‌ای خود قادر به مدیریت تصمیمات راهبُردی، تاکتیکی و واکنشی به‌صورت هم‌زمان است. لایه برنامه‌ریزی کلان در BOSS وظیفه انتخاب مسیرها و تصمیم‌گیری‌های راهبردی را بر عهده دارد، لایه تاکتیکی مدیریت منابع و تخصیص تصمیمات میان واحدها را انجام می‌دهد و لایه کنترل واکنشی پاسخ سریع به تغییرات محیط و شرایط پیش‌بینی نشده را فراهم می‌کند. این ساختار سلسله‌مراتبی باعث می‌شود الگوریتم BOSS بتواند تعامل میان تصمیمات مختلف را به‌صورت یکپارچه مدل‌سازی کند، بااین‌حال، محدودیت‌هایی در انعطاف‌پذیری کنترل واکنشی و توانایی بررسی هم‌زمان چندین گزینه دارد.

در زمینه تصمیم‌گیری چندلایه در سامانه‌های راهبُردی، الگوریتم‌های سلسله‌مراتبی نقش مهمی ایفا می‌کنند. یکی از این الگوریتم‌ها، BOSS است که به‌ویژه در مسائل ساختار ائتلافی بهینه مورداستفاده قرار می‌گیرد. این الگوریتم با استفاده از جستجوی دوطرفه، ساختار ائتلافی بهینه را در مسائل چندعاملی شبیه‌سازی می‌کند و عملکرد بهتری نسبت به روش‌های موجود ارائه می‌دهد [۱۰].

باوجود مزایای الگوریتم BOSS، محدودیت‌هایی نیز در آن مشاهده می‌شود. به‌عنوان‌مثال، در مسائل تصمیم‌گیری چندلایه با ابعاد بزرگ، پیچیدگی محاسباتی الگوریتم افزایش می‌یابد و ممکن است کارایی لازم را نداشته باشد. همچنین، در برخی از کاربردها، نیاز به بهبود در سرعت و دقت الگوریتم احساس می‌شود. در اغلب پژوهش‌های پیشین، ارزیابی عملکرد الگوریتم‌ها بر اساس معیارهای استانداردی مانند دقت[1]، فراخوانی[2]، درستی[3] و میانگین دقت[4] انجام شده است.

در پاسخ به این محدودیت‌ها، پژوهش‌های متعددی به بهبود الگوریتم BOSS پرداخته‌اند. به‌عنوان نمونه، در زمینه بهینه‌سازی پرتفوی، الگوریتم‌هایی با استفاده از جستجوی پرتفو سلسله‌مراتبی ارائه شده‌اند که می‌توانند تصمیمات بهینه‌تری در مسائل پیچیده اتخاذ کنند [۱۱].

باتوجه‌به این پیشینه، در این مقاله نسخه‌ای بهبودیافته از الگوریتم BOSS با نام BOSS-بهبودیافته ارائه شده است که با تمرکز بر بهبود کنترل واکنشی، جستجوی پرتفو سلسله‌مراتبی و کنترل تطبیقی تاکتیکی، توانایی تصمیم‌گیری دقیق‌تر و پاسخ سریع‌تر به تغییرات محیط را دارا است.

باتوجه‌به این محدودیت‌ها، بسیاری از پژوهش‌های اخیر تلاش کرده‌اند تا الگوریتم‌های BOSS را بهبود دهند یا با

[1] Precision
[2] Recall
[3] Accuracy
[4] Mean Precision

رویکردهای مکمل ترکیب کنند تا قابلیت انعطاف و دقت[1] تصمیم‌گیری چندلایه افزایش یابد. در این مطالعه، با الهام از ساختار BOSS و با تمرکز بر بهبود کنترل واکنشی، جستجوی پرتفو سلسله‌مراتبی و کنترل تطبیقی تاکتیکی، نسخه‌ای بهبودیافته با نام BOSS-بهبودیافته (نسخه‌ی بهبودیافته الگوریتم پایه فرماندهی و بهینه‌سازی) ارائه شده است که توانایی تصمیم‌گیری دقیق‌تر و پاسخ سریع‌تر به تغییرات محیط را دارا است.

الگوریتم BOSS (Best Option Simulation Strategy) یکی از روش‌های مطرح در حوزه‌ی برنامه‌ریزی و تصمیم‌گیری در بازی‌های راهبُرد زمان واقعی است. در این الگوریتم، عامل تصمیم‌گیرنده مجموعه‌ای از گزینه‌های ممکن را شبیه‌سازی کرده و بهترین گزینه را بر اساس ارزیابی نتایج شبیه‌سازی انتخاب می‌کند. مزیت اصلی BOSS در کاهش فضای جستجو و توانایی تصمیم‌گیری سریع در محیط‌های پویا است. بااین‌حال، این روش در مواجهه با سناریوهای چندلایه و تصمیم‌گیری تاکتیکی تطبیقی، محدودیت‌هایی مانند ناپایداری در رفتار عامل و وابستگی بالا به عامل‌های ثابت دارد.

در روش پیشنهادی این پژوهش، با عنوان الگوریتم بهبودیافته فرماندهی و بهینه‌سازی، ساختار BOSS با ترکیب الگوریتم‌های جستجوی پرتفو[2] و کنترل واکنشی مبتنی بر ABCD و UCT-CD توسعه‌یافته است تا ضمن حفظ سرعت تصمیم‌گیری، پویایی و ثبات رفتاری عامل نیز بهبود یابد. در لایه‌ی کنترل واکنشی، از الگوریتم‌های ABCD و UCT-CD برای مدیریت هم‌زمان واحدها و انتخاب کنش‌های سریع استفاده شده است. الگوریتم ABCD نسخه‌ای توسعه‌یافته از روش کلاسیک آلفا - بتا است که در آن مدت‌زمان انجام هر کنش[3] در فرایند ارزیابی لحاظ می‌شود. این ویژگی باعث می‌شود عامل هوشمند بتواند در محیط‌های با محدودیت زمانی واقعی، تصمیم‌های دقیق‌تر و هماهنگ‌تری اتخاذ کند.

الگوریتم پایه BOSS الگوریتم پایه فرماندهی و بهینه‌سازی یک روش بهینه‌سازی سلسله‌مراتبی است که برای تصمیم‌گیری چندلایه در سامانه‌های راهبُردی طراحی شده است. این الگوریتم ساختاری سه‌لایه دارد که شامل لایه برنامه‌ریزی کلان، لایه تاکتیکی و لایه کنترل واکنشی است. لایه برنامه‌ریزی کلان وظیفه تصمیم‌گیری راهبردی و تعیین مسیرهای احتمالی را بر عهده دارد، لایه تاکتیکی مدیریت منابع و تعاملات میان واحدها را انجام می‌دهد و لایه کنترل واکنشی پاسخ سریع به تغییرات محیط و شرایط پیش‌بینی نشده را فراهم می‌کند. الگوریتم BOSS به دلیل ساختار سلسله‌مراتبی و سه‌لایه‌ای خود قادر است تصمیمات راهبُردی و تاکتیکی را به‌صورت هم‌زمان مدل‌سازی کند، اما محدودیت‌هایی در انعطاف‌پذیری کنترل واکنشی و توانایی بررسی هم‌زمان چندین گزینه دارد.

برای غلبه بر محدودیت‌های الگوریتم پایه، در این مقاله نسخه بهبودیافته آن با نام -BOSSبهبودیافته (نسخه‌ی بهبودیافته الگوریتم پایه فرماندهی و بهینه‌سازی) ارائه شده است. در این نسخه، بهبودها شامل سه محور اصلی هستند: ۱) کنترل واکنشی ارتقایافته که با الگوریتم ABCD بهبودیافته است تا پاسخ سامانه به تغییرات محیط سریع‌تر و دقیق‌تر باشد، ۲) جستجوی پرتفو سلسله‌مراتبی که امکان بررسی و انتخاب چندین گزینه به‌صورت هم‌زمان و سلسله‌مراتبی را فراهم می‌کند و تصمیم‌گیری چندلایه را منعطف‌تر می‌سازد، و ۳) کنترل تطبیقی تاکتیکی که تخصیص منابع و تصمیمات در لایه تاکتیکی را به‌صورت تطبیقی بهینه می‌کند تا عملکرد سامانه در شرایط متغیر بهبود یابد. این ساختار باعث می‌شود روش پیشنهادی قادر باشد تصمیمات راهبُردی و تاکتیکی را بادقت بالاتر و پاسخ سریع‌تر به تغییرات محیط ارائه دهد.

۲-۲. لایه راهبردی

لایه راهبردی[4] به‌عنوان بالاترین سطح تصمیم‌گیری در عامل پیشنهادی، مسئول تعیین اهداف کلان بازی و هدایت جهت‌گیری کلی رفتار عامل در مواجهه با وضعیت‌های مختلف محیطی است. این سطح وظیفه دارد با تحلیل داده‌های جاری بازی، نوع راهبُرد مناسب - اعم از تهاجمی، تدافعی، گسترش پایگاه یا تمرکز بر توسعه فناوری - را انتخاب و به لایه‌های پایین‌تر ابلاغ کند. به‌منظور پیاده‌سازی این سطح، از ساختار شبکه وظیفه‌ای سلسله‌مراتبی استفاده شده که قابلیت تجزیه یک هدف کلی به مجموعه‌ای از وظایف میانی و اقدامات عملیاتی را دارد [۱۰]. این ساختار با بهره‌گیری از برنامه‌ریزی بالابه‌پایین، تصمیمات راهبردی را به شکل درختی سازمان‌دهی می‌کند، به‌گونه‌ای که هر گره بیانگر یک وظیفه و هر شاخه نشان‌دهنده تبدیل آن وظیفه به مراحل اجرایی‌تر است.

برای مثال، در شرایطی که عامل متوجه شود دشمن به طور فعال درحال‌توسعه اقتصادی است (مثلاً در ثانیه ۶۰ بازی تعداد زیادی کارگر جدید تولید شده یا ساختمان‌های معدنی گسترش یافته‌اند)، ساختار شبکه وظیفه‌ای سلسله‌مراتبی می‌تواند تصمیم حمله زودهنگام[5] را اتخاذ کند. این تصمیم به طور خودکار باعث فعال‌سازی یک زیرشبکه تاکتیکی برای تولید سریع واحدهای

[1] precision
[2] Portfolio Search
[3] Duration
[4] Strategic Layer
[5] Strategic Rush

تهاجمی مانند Zealot و حمله به خطوط تأمین منابع دشمن خواهد شد. این تصمیم نه‌تنها متناسب با وضعیت فعلی بازی اتخاذ می‌شود، بلکه به‌گونه‌ای طراحی شده که بتواند در سناریوهای مختلف نیز اعمال شود. [۱۱].

از مزایای استفاده از این ساختار می‌توان به قابلیت تعمیم به شرایط جدید، سازگاری با داده‌های پویا و غیرقطعی محیط، و امکان پشتیبانی از سناریوهای چندهدفه به طور هم‌زمان اشاره کرد. برخلاف روش‌های سنتی تصمیم‌گیری ایستا، شبکه وظیفه‌ای سلسله‌مراتبی این امکان را فراهم می‌سازد که عامل بتواند به‌صورت مستمر اهداف خود را بازتنظیم کرده و تصمیمات بلندمدت را به تصمیمات تاکتیکی و عملیاتی قابل‌اجرا ترجمه کند.

۲-۳. لایه راهبردی

لایه تاکتیکی[1] در ساختار عامل پیشنهادی، به‌عنوان واسطه‌ای بین اهداف کلان‌راهبردی و اقدامات دقیق واکنشی عمل می‌کند. نقش اصلی این لایه، تعیین چگونگی اجرای یک هدف راهبردی در سطح میدان نبرد است. به عبارتی، این لایه مسئول طراحی و مدیریت مانورها، انتخاب محل حمله یا دفاع، زمان‌بندی درگیری‌ها، و سازماندهی حرکت و ترکیب یگان‌ها به‌صورت گروهی است. لایه تاکتیکی باید بتواند در زمان محدود و با اطلاعات ناقص، مناسب‌ترین مسیر عملیاتی را برای تحقق هدف انتخاب‌شده از سوی لایه راهبردی تعیین کند.

برای دستیابی به این هدف، در این پژوهش ترکیبی از دو الگوریتم مکمل استفاده شده است. نخست، الگوریتم جستجوی عروسکی به‌کارگرفته‌شده تا از میان مجموعه‌ای از رفتارهای ازپیش‌طراحی‌شده (اسکریپت شده) که دارای نقاط تصمیم‌گیری داخلی هستند، گزینه‌های تاکتیکی اولیه استخراج شود (باریگا، ۲۰۱۵). این ساختار امکان بهره‌برداری از تجارب قبلی را با سرعت بالا فراهم می‌سازد. دوم، به‌منظور تحلیل و انتخاب بین این گزینه‌ها، از جستجوی درخت مونت‌کارلو باسیاست الگوریتم درخت جستجوی مونت‌کارلو با کنترل تطبیقی بهره گرفته شده است که با استفاده از شبیه‌سازی‌های تصادفی و ارزیابی چندباره، توالی‌هایی از اقدامات را که بیشترین احتمال موفقیت دارند، انتخاب می‌کند. این ترکیب موجب شده است که عامل بتواند ضمن بهره‌گیری از تجربه، در هر لحظه نیز متناسب با وضعیت میدان نبرد، رفتار خود را بهینه‌سازی کند [۱۲].

برای مثال، اگر لایه راهبردی تصمیم به اجرای حمله بدهد، لایه تاکتیکی با ارزیابی شرایط ممکن تعیین می‌کند که آیا حمله باید از جبهه مستقیم باشد یا به‌صورت حمله انحرافی[2] از پهلو صورت گیرد. همچنین در این مرحله، زمان آغاز عملیات، چیدمان اولیه نیروها، و مسیرهای حرکتی نیز برنامه‌ریزی می‌شود. آنچه این طراحی را از روش‌های مرسوم متمایز می‌سازد، استفاده از مدل وزن‌دهی دینامیک به سناریوها و به‌کارگیری الگوریتم انتخاب حریصانه پرتفو[3] است که در کنار افزایش سرعت تصمیم‌گیری، دقت انتخاب گزینه‌های تاکتیکی را نیز بالا برده است. عامل در هر لحظه، چند اسکریپت تاکتیکی را به‌صورت تطبیقی ارزیابی کرده و باتوجه‌به شرایط لحظه‌ای بازی، مناسب‌ترین آن‌ها را برای اجرا برمی‌گزیند.

در مدل پیشنهادی، وزن‌های دینامیک برای هر لایه الگوریتم به‌منظور انعطاف‌پذیری و پاسخ سریع به تغییرات محیط طراحی شده‌اند. این وزن‌ها نشان‌دهنده اهمیت نسبی هر لایه در اتخاذ تصمیم نهایی هستند و به‌صورت مرحله‌ای و بر اساس عملکرد الگوریتم و شرایط محیطی به‌روزرسانی می‌شوند. به طور مشخص، اگر یک‌لایه عملکرد بهتری نسبت به پیش‌بینی‌های خود داشته باشد، وزن آن افزایش می‌یابد تا نقش آن در تصمیم نهایی بیشتر شود، و بالعکس اگر عملکرد لایه کاهش یابد، وزن آن کاهش‌یافته و تأثیر آن کمتر می‌شود. این مکانیزم باعث می‌شود که سامانه بتواند به‌صورت خودتنظیم و تطبیقی، اهمیت هر بخش از الگوریتم را بر اساس شرایط فعلی محیط بهینه کند و تصمیمات نهایی دقیق‌تر و کارآمدتری اتخاذ شود.

در مجموع، طراحی لایه تاکتیکی به‌گونه‌ای انجام شده که نه‌تنها به تصمیمات راهبردی پاسخ دقیق می‌دهد، بلکه از نظر عملکردی نیز توانایی تطابق با سناریوهای غیرمنتظره را دارد. این امر، انسجام رفتاری عامل را در تعامل با سایر لایه‌ها تضمین می‌کند و موجب می‌شود که مسیر اجرای اهداف کلان، با واقعیت میدان نبرد هم‌راستا باقی بماند. شکل (۱) آرایش اولیه نیروها در سناریوی حمله انحرافی که توسط عامل هوشمند پیشنهادی اتخاذ شده است را نشان می‌دهد.



شکل (۱). آرایش اولیه نیروها در سناریوی حمله انحرافی. ناحیه هدف با دایره قرمز مشخص شده است.

[1] Tactical Layer
[2] Flanking
[3] Greedy Portfolio Selection

۲-۴. لایه کنترل واکنشی

لایه کنترل واکنشی[1]، پایین‌ترین سطح در ساختار عامل پیشنهادی را تشکیل می‌دهد و وظیفهٔ اجرای مستقیم و دقیق دستورات صادره از سطوح تاکتیکی و راهبردی را در میدان نبرد بر عهده دارد. این لایه به‌گونه‌ای طراحی شده که بتواند در بازه‌های زمانی بسیار کوتاه (در حد چند میلی‌ثانیه) نسبت به تغییرات ناگهانی در محیط بازی واکنش نشان دهد و تصمیماتی نظیر حرکت، هدف‌گیری، درگیری یا عقب‌نشینی یگان‌ها را به‌صورت انفرادی یا گروهی مدیریت کند. برخلاف لایه‌های بالا که بر تصمیمات کلان و نیمه کلان تمرکز دارند، در این سطح تمرکز بر بهینه‌سازی رفتارهای جزئی و آنی واحدها در لحظه است.

برای پیاده‌سازی تصمیم‌گیری در این لایه، از دو الگوریتم بهینه بهره گرفته شده است. نخست، الگوریتم ABCD که نسخه‌ای پیشرفته از الگوریتم کلاسیک Alpha-Beta است و با درنظرگرفتن مدت‌زمان اجرای اقدامات، درخت تصمیم‌گیری را به‌صورت واقع‌گرایانه‌تری هدایت می‌کند [۱۲]. این الگوریتم به‌ویژه در سناریوهایی که واحدهای بازی دارای سرعت‌های مختلف یا زمان‌های بارگذاری[2] متفاوت هستند، عملکرد مؤثری دارد. دوم، الگوریتم درخت جستجوی مونت‌کارلو با کنترل تطبیقی D که ترکیبی از Monte Carlo Tree Search و درک زمانی از اجرای اقدامات[3] است، امکان اتخاذ تصمیماتی را فراهم می‌سازد که هم تطبیق‌پذیر با محیط هستند و هم محدودیت‌های سخت‌گیرانهٔ زمان واقعی بازی را لحاظ می‌کنند [۱۳].

برای مثال، در شرایطی که دشمن از واحدهای دورزن مانند Dragoon برای تهاجم استفاده می‌کند، لایه واکنشی به‌صورت مستقل تشخیص می‌دهد که واحدهای نزدیک زن Melee مانند Zealot باید از تاکتیک‌های مانوری مانند جاخالی‌دادن یا حرکت مارپیچی Kiting برای نزدیک‌شدن بدون آسیب جدی استفاده کنند. همچنین در صورت احساس تهدید فوری، این لایه قادر است بدون نیاز به صدور فرمان جدید از سطوح بالا، تصمیم به عقب‌نشینی یا تغییر آرایش اتخاذ کرده و واحدهای موجود را حفظ نماید.

از مزایای این لایه می‌توان به توانایی پاسخ سریع در شرایط بحرانی، کاهش تلفات از طریق مدیریت مستقل تک‌تک یگان‌ها و هماهنگی مؤثر در موقعیت‌هایی با اقدامات هم‌زمان[4] اشاره کرد. طراحی این لایه به‌گونه‌ای انجام شده که نه‌تنها با سرعت اجرای بازی هم‌خوان باشد، بلکه خروجی آن کاملاً با تصمیمات لایه تاکتیکی هم‌راستا باقی بماند. در نتیجه، عامل پیشنهادی در این پژوهش نه‌فقط در سطوح برنامه‌ریزی، بلکه در سطح اجرا نیز رفتاری هوشمند، واکنش محور و منطبق با واقعیت‌های میدانی از خود بروز می‌دهد. شکل (۲) واکنش بلادرنگ عامل پیشنهادی به حمله دشمن در فاز میانی بازی را نشان می‌دهد که در این سناریو، واحدهای Melee نوع Zealot است.



شکل (۲). واکنش بلادرنگ عامل پیشنهادی به حمله دشمن در فاز میانی بازی

موتور برنامه‌ریز ساخت در این مدل بر پایهٔ الگوریتم BOSS طراحی شده، اما چند ویژگی کلیدی به آن اضافه شده تا قابلیت اجرا در شرایط متغیر و در زمان واقعی را داشته باشد (چورچیل، ۲۰۱۳). ویژگی‌های مدل پیشنهادی شامل موارد زیر است:

۱. برنامه‌ریزی تطبیقی: هدف ساخت (مثلاً «۲ Dragoon و ۴ Zealot») بر اساس تحلیل میدان نبرد به‌روزرسانی می‌شود.
۲. درنظرگرفتن ریسک: اگر دشمن حمله قریب‌الوقوع داشته باشد، برنامه‌ریز از ادامه ساخت واحدهای تهاجمی صرف‌نظر کرده و به‌سرعت ساخت یگان‌های دفاعی را جایگزین می‌کند.
۳. مدیریت منابع بلادرنگ: محاسبه نرخ ورود و مصرف منابع در هر فریم و تطابق آن با برنامه ساخت.
۴. شبیه‌سازی سریع آینده بازی برای پیش‌بینی اینکه ساخت فعلی در زمان X به چه ترکیب نیروهایی منتهی می‌شود.

در نهایت، به‌منظور ارزیابی مدل، این ساختار در محیط واقعی StarCraft: Brood War پیاده‌سازی شده است. اتصال عامل با موتور بازی از طریق کتابخانه BWAPI انجام شده و الگوریتم‌ها در زبان C++ توسعه داده شده‌اند. تحلیل داده‌های عملکرد، مانند نرخ پیروزی، زمان ساخت، و سرعت اجرا، با استفاده از پایتون و کتابخانه‌هایی نظیر Pandas و Matplotlib صورت‌گرفته است. کل آزمایش‌ها بر روی یک سامانه با پردازنده Core i7 و رم ۱۶ گیگابایت انجام شده تا شرایط عملیاتی نیز لحاظ گردد.

[1] Micro / Reactive Control
[2] Cooldown
[3] Command Durations
[4] Simultaneous Actions

۲-۴. پیاده‌سازی الگوریتم‌های ترکیبی

الگوریتم‌های ترکیبی استفاده شده در این مطالعه شامل UCT-CD و ABCD هستند که به‌منظور تصمیم‌گیری چندلایه در بازی‌های راهبُردی هم‌زمان با عملکرد بهینه طراحی شده‌اند. فرایند پیاده‌سازی به‌صورت سلسله‌مراتبی و گام‌به‌گام انجام شده است تا هر لایه از الگوریتم بتواند نقش مشخص خود را در تصمیم‌گیری ایفا کند.

۱. UCT-CD: این الگوریتم بر پایه درخت جستجوی متمرکز بر ارزش (UCT) طراحی شده و برای پیش‌بینی نتایج بلندمدت تصمیمات، امکان کاوش تصادفی اما هدایت‌شده در فضای تصمیمات را فراهم می‌کند. UCT-CD با استفاده از معیارهای ارزش و هزینه، مسیرهای بالقوه را ارزیابی کرده و بهترین مسیر تاکتیکی را انتخاب می‌کند. در پیاده‌سازی، تعداد شبیه‌سازی‌ها و عمق درخت جستجو بر اساس پیچیدگی مسئله تنظیم شده است.

۲. ABCD: این الگوریتم به‌عنوان لایه کنترل واکنشی طراحی شده و وظیفه پاسخ سریع به تغییرات محیط را بر عهده دارد. ABCD با تحلیل شرایط فعلی و پیش‌بینی کوتاه‌مدت، تصمیمات اصلاحی فوری اتخاذ می‌کند. در پیاده‌سازی، این الگوریتم به‌گونه‌ای تنظیم شده است که ورودی‌ها از لایه تاکتیکی دریافت شوند و خروجی‌های اصلاحی به‌صورت مستقیم بر اقدامات واحدها اثرگذار باشد.

۳. ادغام الگوریتم‌ها: در سطح بالاتر، نتایج حاصل از UCT-CD و ABCD با یکدیگر تلفیق می‌شوند تا تصمیم نهایی در هر مرحله اتخاذ گردد. ترکیب این دو الگوریتم به‌گونه‌ای طراحی شده که لایه تاکتیکی توسط UCT-CD هدایت شود و لایه واکنشی ABCD تغییرات ناگهانی محیط را مدیریت کند. این ترکیب باعث می‌شود سامانه هم‌زمان توانایی برنامه‌ریزی بلندمدت و پاسخ سریع به شرایط غیرمنتظره را داشته باشد.

در این مطالعه، الگوریتم‌های مرجع کلاسیک به‌عنوان خطوط پایه انتخاب شده‌اند؛ زیرا این الگوریتم‌ها در مطالعات پیشین در محیط‌های بازی‌های راهبُرد هم‌زمان به طور گسترده استفاده شده و عملکرد آن‌ها به‌خوبی مستند شده است. انتخاب این خطوط پایه امکان مقایسه مستقیم و عینی عملکرد مدل پیشنهادی با روش‌های مرسوم را فراهم می‌کند. علاوه بر این، به طور خلاصه، مدل پیشنهادی با روش‌های مبتنی بر یادگیری تقویتی عمیق (DRL) مقایسه شده است؛ درحالی‌که DRL توانایی یادگیری خودکار و استخراج سیاست‌های پیچیده را دارد، اغلب به منابع محاسباتی بالاتر و زمان آموزش طولانی‌تر نیاز دارد، درحالی‌که مدل پیشنهادی با ساختار سه‌لایه و وزن‌های دینامیک، تعادلی میان دقت تصمیم‌گیری، زمان پاسخ‌دهی بلادرنگ و قابلیت توسعه ماژولار ایجاد می‌کند. این مقایسه نشان می‌دهد که الگوریتم پیشنهادی ضمن حفظ سادگی و سرعت، قابلیت رقابت با روش‌های پیشرفته DRL را دارد.

۳. نتایج و تحلیل تجربی

به‌منظور ارزیابی مدل پیشنهادی، مجموعه‌ای از آزمایش‌های تجربی در محیط بازی StarCraft: Brood War انجام شد. سناریوها باهدف بررسی عملکرد عامل پیشنهادی در مقایسه با روش‌های مرجع طراحی شده‌اند. نتایج در سه محور اصلی تحلیل شده‌اند:

۱. بهینه‌سازی ترتیب ساخت
۲. کنترل تاکتیکی در درگیری‌ها[1]
۳. سرعت اجرا و قابلیت عملکرد در زمان واقعی[2]

جدول (۱). عامل‌های اصلی الگوریتم‌های UCT-CD و ABCD و مقادیر پیشنهادی آنها

الگوریتم	عامل	مقدار	توضیح
UCT-CD	تعداد شبیه‌سازی‌ها (Simulations)	۱۰۰۰	تعداد دفعات اجرای شبیه‌سازی برای ارزیابی هر شاخه
	عمق درخت جستجو (Search Depth)	۱۰	تعداد گام‌هایی که درخت جستجو در هر شاخه پیش می‌رود
	ضریب اکتشاف/اکسپلویت (Exploration Parameter, C)	۱/۴	وزن‌دهی بین کاوش و بهره‌برداری
	معیار ارزش/هزینه (Value/Cost Function)	میانگین پاداش‌های احتمالی در طول مسیر	تابع ارزیابی مسیرها برای انتخاب بهترین تصمیم
	نرخ پاسخ واکنشی (Reaction Rate)	۰/۲	سرعت واکنش الگوریتم به تغییرات محیط
	بازه زمانی به‌روزرسانی (Update Interval)	یک گام زمانی	فاصله زمانی بین دو تصمیم واکنشی
ABCD	ورودی‌های تاکتیکی (Input Parameters)	موقعیت فعلی، وضعیت منابع، اهداف کوتاه‌مدت	داده‌های دریافتی از لایه تاکتیکی برای اصلاح تصمیم
	محدوده عامل‌های اصلاحی (Parameter Bounds)	[-1,1] برای تغییرات نرمالیزه شده	حداقل و حداکثر مقادیر قابل‌اعمال در تصمیمات اصلاحی

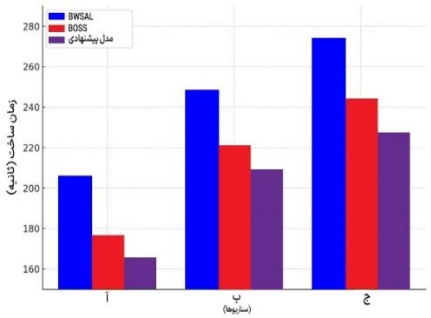
[1] Tactical Combat
[2] Real-Time Feasibility

۳-۱. مقایسه الگوریتم‌های ساخت واحدها

در این بخش، عامل پیشنهادی[1] با الگوریتم‌های مرجع BOSS و BWSAL مقایسه شده است. هدف، بررسی زمان رسیدن به ترکیب مشخصی از یگان‌ها در سناریوهای مختلف است.

جدول (۱). زمان رسیدن به ترکیب هدف (ثانیه)

سناریو	ترکیب هدف	BWSAL	BOSS	الگوریتم بهبودیافته فرماندهی و بهینه‌سازی پیشنهادی
آ	6 Zealot	۲۰۵	۱۷۶	۱۶۵
ب	2 Dragoon + 3 Zealot	۲۴۸	۲۲۰	۲۰۸
ج	4 Zealot + 1 Reaver	۲۷۴	۲۴۳	۲۲۷



شکل (۳). مقایسه زمان ساخت

مدل پیشنهادی میانگین بهبود حدود ۷٪ نسبت به BOSS و ۱۹٪ نسبت به BWSAL دارد. شکل (۳) زمان دستیابی به ترکیب موردنظر را نشان می‌دهد.

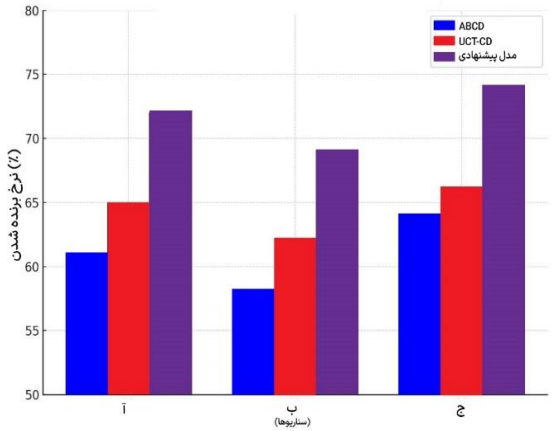
۳-۲. ارزیابی کیفیت تصمیم‌گیری تاکتیکی

معیار «ثبات تصمیم‌گیری» نشان‌دهنده میزان پایداری و تداوم تصمیمات اتخاذشده توسط الگوریتم در طول زمان است. این معیار نوآورانه، رفتار الگوریتم را در مواجهه با تغییرات محیط و نوسانات داده‌ها ارزیابی می‌کند و مشخص می‌کند که الگوریتم تا چه اندازه تصمیمات خود را بدون تغییرات ناگهانی حفظ می‌کند. برای محاسبه این معیار، رفتار الگوریتم در چندین مرحله متوالی بررسی می‌شود و اختلافات بین تصمیمات متوالی اندازه‌گیری می‌گردد. هرچه این اختلافات کمتر باشد، ثبات تصمیم‌گیری بالاتر است و نشان می‌دهد الگوریتم در شرایط مشابه، تصمیمات هماهنگ و پایدار اتخاذ می‌کند.

در این مرحله، عامل پیشنهادی با مدل‌های ABCD و الگوریتم درخت جستجوی مونت‌کارلو با کنترل تطبیقی D در سه سناریوی درگیری برابر آزمایش شد. معیار ارزیابی: درصد برد در ۳۰ اجرای متوالی. مدل پیشنهادی در تمام سناریوها بالاترین نرخ موفقیت را دارد شکل (۴) درصد برد تاکتیکی را نشان می‌دهد.

جدول (۲). درصد برد عامل‌ها در سناریوهای نبرد

سناریو	ترکیب نیروها	ABCD	الگوریتم درخت جستجوی مونت‌کارلو با کنترل تطبیقی D	مدل پیشنهادی
آ	8 vs 8 Dragoon	٪۶۱	٪۶۵	٪۷۲
ب	5 Zealot vs 4 Dragoon	٪۵۸	٪۶۲	٪۶۹
ج	3 Zealot + 3 Dragoon vs 6 Zealot	٪۶۴	٪۶۶	٪۷۴



شکل (۴). درصد برد تاکتیکی

۳-۳. تحلیل سرعت و زمان اجرای الگوریتم‌ها

توانایی اجرای عامل در محیط واقعی بسیار مهم است. میانگین زمان تصمیم‌گیری (در هر فریم) برای هر الگوریتم در جدول (۳) آمده است:

جدول (۴). میانگین زمان تصمیم‌گیری

الگوریتم	میانگین زمان تصمیم	قابلیت اجرای بلادرنگ
ABCD	۱۲/۵ میلی‌ثانیه	بله
الگوریتم درخت جستجوی مونت‌کارلو با کنترل تطبیقی D	۱۶/۳ میلی‌ثانیه	بله
مدل پیشنهادی	۱۰/۲ میلی‌ثانیه	بله

باتوجه‌به محدودیت ۱۶/۷ میلی‌ثانیه در هر فریم بازی، تمام الگوریتم‌ها قابل‌اجرا هستند؛ اما مدل پیشنهادی سریع‌ترین زمان پاسخ را دارد.

[1] Enhanced-BOSS

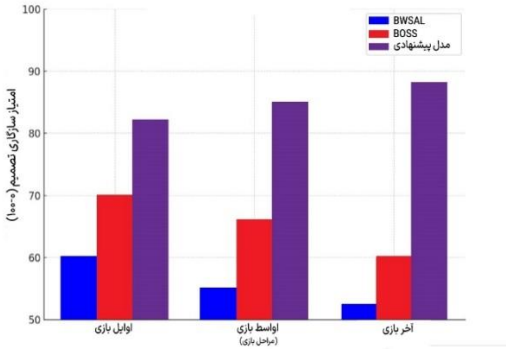
جدول (۴). خلاصه نتایج

شاخص	مدل پیشنهادی	BOSS	BWSAL	الگوریتم درخت جستجوی مونت‌کارلو با کنترل تطبیقی D	ABCD
زمان ساخت	بهتر	متوسط	ضعیف	-	-
درصد برد	بالاتر	-	-	خوب	متوسط
سرعت اجرا	سریع	متوسط	-	کند	سریع

یکی از چالش‌های مهم در طراحی عامل‌های هوشمند برای محیط‌های پیچیده و پویا مانند بازی‌های ویدئویی راهبُرد هم‌زمان، ثبات در فرایند تصمیم‌گیری در فازهای مختلف بازی است. عامل‌هایی که در اثر تغییر شرایط محیطی، پی‌درپی تصمیمات خود را تغییر می‌دهند، معمولاً دچار ناهماهنگی بین لایه‌های تصمیم‌گیری می‌شوند و این امر باعث کاهش عملکرد کلی آن‌ها می‌شود. به‌منظور بررسی این موضوع، در این پژوهش یک شاخص نوین با عنوان "ثبات تصمیم‌گیری[1]" تعریف و معرفی شده است که در منابع پیشین موردتوجه قرار نگرفته بود. این شاخص، میزان پایداری تصمیمات عامل را در طول زمان و در سه‌فاز Early Game، Mid Game و Late Game اندازه‌گیری می‌کند. ثبات تصمیم‌گیری نشان می‌دهد که عامل در شرایط پیچیده و متغیر بازی تا چه میزان تصمیمات پایدار، منطقی اتخاذ می‌کند. این معیار به‌صورت ترکیبی از:

- نوسان در انتخاب اهداف بین فریم‌ها
- تغییر ناگهانی در رفتار تاکتیکی
- میزان هماهنگی بین لایه‌های تصمیم‌گیری

محاسبه می‌شود و عددی بین ۰ تا ۱۰۰ به هر فاز بازی اختصاص داده می‌شود.



شکل (۵). نمودار ثبات تصمیم در فازهای بازی

شکل (۵) نمودار ثبات تصمیم در فازهای بازی را نشان می‌دهد که نتایج به‌دست‌آمده از مقایسه مدل پیشنهادی با روش‌های مرجع BWSAL و BOSS نشان می‌دهد که عامل پیشنهادی در هر سه‌فاز بازی، ثبات بیشتری در تصمیم‌گیری از خود نشان داده است. این پایداری به‌ویژه در فاز پایانی بازی مشهودتر است که ناشی از طراحی تعاملی و هماهنگ بین لایه‌های راهبردی، تاکتیکی و واکنشی در مدل پیشنهادی است. در مقابل، مدل‌های کلاسیک به دلیل ساختار خطی یا گسسته، در مدیریت وضعیت‌های ترکیبی و پرتنش با نوسان در رفتار مواجه می‌شوند. ازاین‌رو، می‌توان شاخص ثبات تصمیم‌گیری را به‌عنوان ابزاری نوین برای ارزیابی کیفیت عامل‌های چندلایه در محیط‌های پیچیده معرفی نمود.

باوجود نتایج مثبت و عملکرد برتر مدل پیشنهادی، برخی محدودیت‌ها نیز وجود دارند که باید در نظر گرفته شوند. اولاً، مدل فعلی هنوز به طور کامل در سناریوهای بزرگ‌تر و محیط‌های پیچیده‌تر با تعداد واحدهای بیشتر ارزیابی نشده و مقیاس‌پذیری آن نیازمند بررسی‌های بیشتر است. ثانیاً، عملکرد الگوریتم به مقادیر عامل‌های UCT-CD و ABCD و وزن‌های دینامیک وابسته است و برای دستیابی به نتایج بهینه نیازمند تنظیم دقیق این عامل‌ها است. ثالثاً، ساختار ارائه شده تاکنون تنها در محیط‌های مشخص بازی‌های راهبُرد هم‌زمان آزمایش شده و تعمیم آن به سایر سبک‌های بازی، مانند Total War یا Company of Heroes، نیازمند اصلاحات و آزمایش‌های جدید است. در نهایت، ماژول تحلیل حریف به‌صورت کامل پیاده‌سازی نشده و تعامل با سبک بازی دشمن محدود است که می‌تواند در بهبود تصمیم‌گیری تاکتیکی و واکنشی عامل اثرگذار باشد.

۴. نتیجه‌گیری

در این مقاله، مدلی سه‌لایه برای تصمیم‌گیری هوشمند در محیط‌های بازی‌های ویدئویی راهبُرد هم‌زمان طراحی شد که بر اساس ساختار سلسله‌مراتبی راهبردی، الگوریتم‌های تطبیقی تاکتیکی (جستجوی عروسکی + الگوریتم درخت جستجوی مونت‌کارلو با کنترل تطبیقی D)، و کنترل دقیق واکنشی ABCD عمل می‌کند. نوآوری اصلی پژوهش در این است که برخلاف مطالعات گذشته که هر لایه تصمیم‌گیری را به‌صورت مستقل بررسی کرده‌اند، اینجا برای نخستین‌بار یکپارچگی هدفمند بین این لایه‌ها برقرار شده و تعامل آن‌ها با یکدیگر در قالب یک ساختار عملیاتی پیاده‌سازی شده است.

[1] Decision Consistency

علاوه‌برآن، برای ارزیابی رفتار بلندمدت عامل، یک شاخص جدید به نام ثبات تصمیم‌گیری معرفی شد که عملکرد پایدار عامل را در فازهای مختلف بازی اندازه‌گیری می‌کند - موضوعی که در پژوهش‌های پیشین مطرح نبوده و نشان‌دهنده زاویه دید نوین این پژوهش به ارزیابی کیفی عامل‌های چندلایه است.

نتایج تجربی نیز بیانگر برتری مدل پیشنهادی نسبت به الگوریتم‌های مرجع از نظر زمان ساخت، نرخ برد در نبرد، و زمان پاسخ‌دهی بلادرنگ بوده است؛ درحالی‌که هم‌زمان ساختار عامل ساده، ماژولار و قابل‌توسعه باقی‌مانده است.

باتوجه‌به ظرفیت بالای ساختار پیشنهادی، چند مسیر برای توسعه آتی پیشنهاد می‌شود:

- تعمیم ساختار عامل به سایر سبک‌های بازی‌های ویدئویی راهبُرد هم‌زمان با ویژگی‌های متفاوت مانند Total War یا Company of Heroes
- استفاده از الگوریتم‌های یادگیری عمیق برای بهبود سیاست‌های تاکتیکی و میکرو
- افزودن ماژول تحلیل حریف برای انطباق راهبُرد با سبک بازی دشمن
- طراحی سیستم‌عامل چند عامل با هماهنگی بین چندین عامل هوشمند هم‌زمان

کارهای آینده

باتوجه‌به ظرفیت بالای ساختار پیشنهادی و محدودیت‌هایی که در این مطالعه وجود داشت، چند مسیر برای توسعه آتی پیشنهاد می‌شود:

- تعمیم ساختار عامل به سایر سبک‌های بازی‌های ویدئویی راهبُرد هم‌زمان با ویژگی‌های متفاوت، مانند Total War یا Company of Heroes.
- استفاده از الگوریتم‌های یادگیری عمیق برای بهبود سیاست‌های تاکتیکی و تصمیم‌گیری‌های میکرو در سطح لایه تاکتیکی.
- افزودن ماژول تحلیل حریف برای انطباق راهبُرد با سبک بازی دشمن و بهبود تصمیمات واکنشی.
- طراحی سیستم‌عامل چند عامل با هماهنگی بین چندین عامل هوشمند هم‌زمان برای شبیه‌سازی محیط‌های پیچیده‌تر و واقعی‌تر.
- ارزیابی مقیاس‌پذیری مدل در سناریوهای بزرگ‌تر و تحلیل عملکرد الگوریتم‌ها در شرایط پیچیده‌تر که در نسخه فعلی انجام نشده و به‌عنوان موضوع مطالعات آتی پیشنهاد شده است.

۵. مراجع

[1] G. C. Barros e Sá and C. A. G. Madeira, “Deep reinforcement learning in real-time strategy games: a systematic literature review,” Applied Intelligence, vol. 55, p. 243, 2025, doi: 10.1007/s10489-024-06220-4.

[2] S. Ontañón, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss, “RTS AI Problems and Techniques,” in Encyclopedia of Computer Graphics and Games, N. Lee, Ed. Cham: Springer, 2024, doi: 10.1007/978-3-031-23161-2_17.

[3] N. A. Barriga, M. Stanescu, and M. Buro, “Building placement optimization in real-time strategy games,” in Proc. 10th Artificial Intelligence and Interactive Digital Entertainment Conf. (AIIDE), Raleigh, NC, USA: AAAI Press, 2014, doi: 10.1609/aiide. v10i2.12735.

[4] C. Madeira, V. Corruble, and G. Ramalho, “Designing a reinforcement learning-based adaptive AI for large-scale strategy games,” in Proc. AI and Interactive Digital Entertainment Conf. (AIIDE), Boston, MA, USA: AAAI Press, 2006, doi: 10.1609/aiide.v2i1.18759.

[5] W. J. Chang, F. Pittaluga, M. Tomizuka, W. Zhan, and M. Chandraker, “SAFE-SIM: Safety-Critical Closed-Loop Traffic Simulation with Diffusion-Controllable Adversaries,” in Computer Vision, 2025, doi: 10.1007/978-3-031-72664-4_14.

[6] . A. Mousavi, M. Nasiri, and M. J. Nateghi, “Target Detection in Ground Based Images with Limited Dataset Based on Deep Learning,” Journal of Advanced Defense Science & Technology, vol. 15, no. 1, pp. 27-35, 2024. (In Persian).

[7] M. Zahiri, K. Shirini, and S. Samadi Gharehveran, “Network traffic analysis with machine learning for faster detection of distributed denial of service attack,” Journal of Advanced Defense Science & Technology, vol. 14, no. 4, pp. 273-282, 2024. (In Persian). DOR: 20.1001.1.26762935.1402.14.4.6.2

[8] X. Ma, W. Abdelfattah, D. Luo, et al., “non-cooperative game theory with generative adversarial network for effective decision-making in military cyber warfare”, Annals of Operations Research, 2024, doi: 10.1007/s10479-024-06406-6.

[9] A. Cullen, T. Alpcan, and A. Kalloniatis, “Game-Theoretic Analysis of Adversarial Decision Making in a Complex Socio-Physical System,” Dynamic Games and Applications, vol. 15, pp. 709–728, 2025, doi: 10.1007/s13235-024-00593-4.

[10] Ontañón, S., Synnaeve, G., Uriarte, A., Richoux, F., Churchill, D., & Preuss, M. (2013). A survey of real-time strategy game AI research and competition in StarCraft. IEEE Transactions on Computational Intelligence and AI in Games, 5(4), 293–311.

[11] F. Sailer, M. Buro, and M. Lanctot, “Adversarial planning through strategy simulation,” in 2007 IEEE Symposium on Computational Intelligence and Games (CIG), 2007, pp. 80–87, doi: 10.1109/CIG.2007.368106.

[12] Changder, N., Aknine, S., Ramchurn, S. D., & Dutta, A. (2021). BOSS: A bi-directional search technique for optimal coalition structure generation with minimal overlapping. Proceedings of the AAAI Conference on Artificial Intelligence, 35, 15765–15766.

[13] Dockhorn, A., Hurtado-Grueso, J., Jeurissen, D., Xu, L., & Perez-Liebana, D. (2021). Portfolio search and optimization for general strategy game-playing. Proceedings of the IEEE Congress on Evolutionary Computation (CEC), 1–8.

[14] S. Shao, Y. Tian, Y. Zhang, S. Yang, P. Zhang, C. He,... and Y. Jin, “Evolutionary Computation for Sparse Multi-Objective Optimization: A Survey,” ACM Computing Surveys, 2025. https://doi.org/10.1145/3734865

[15] A. Soufi Nezhad and S. H. Rafieiazar, “Channel Estimation in SC-FDE Using FZC and Performance Comparison with OFDM in PAPR Reduction for High Order Modulation,” 2021. (In Persian).

[16] A. AsadiGhanbari, S. A. Sadatinejad, M. Mohammadnia, and H. Alaei, “Dynamic Battle Management Using Meta-Heuristic Algorithms, Fuzzy Inference Systems and the Decision Tree,” Journal of Advanced Defense Science & Technology, vol. 13, no. 1, pp. 1-11, 2022. (In Persian). DOR: 20.1001.1.26762935.1401.13.1.7.0

[17] Justesen, N., Bontrager, P., Togelius, J., & Risi, S. (2019). Deep learning for video game playing. IEEE Transactions on Games, 11(4), 347–355.

[18] C. Sun, Q. Shao, Z. Zhou, and J. Zhang, “An enhanced FCM clustering method based on multi-strategy tuna swarm optimization,” Mathematics, vol. 12, no. 3, p. 453, 2024. https://doi.org/10.3390/math12030453.

[19] A. Tolui Far, A., A. Karimi, and F. Karimi, “Code Smells detection using ensemble feature selection methods and wrapper techniques based on neural network,” Journal of Advanced Defense Science & Technology, vol. 15, no. 3, pp. 189-207, 2024.(In Persian).

DOR: 20.1001.1.26762935.1403.15.3.2.3