دانشکده مهندسی برق

یادگیری تقویتی در کنترل

گزارش پروژه اول (اصلی)

Multi-Agent Reinforcement Learning via Adaptive Kalman Temporal Difference and Successor Representation

استاد: دکتر سعید شمقدری

دانشجویان: مبینا لشگری، سیده ستاره خسروی

زمستان ۱۴۰۳

چکیده

توسعه الگوریتم‌های یادگیری تقویتی چندعاملی توزیع‌شده (MARL) اخیراً مورد توجه بسیاری قرار گرفته است. به طور کلی، الگوریتم‌های سنتی یادگیری تقویتی مبتنی بر مدل (MB) یا بدون مدل (MF) به دلیل استفاده از یک مدل ثابت پاداش برای یادگیری تابع ارزش، به طور مستقیم برای مسائل MARL قابل استفاده نیستند. اگرچه راهکارهای مبتنی بر شبکه‌های عصبی عمیق (DNN) عملکرد مناسبی دارند، اما همچنان در معرض مشکلاتی همچون بیش‌برازش، حساسیت بالا به انتخاب پارامترها و ناکارآمدی نمونه‌ها هستند.

در این مقاله، یک چارچوب مبتنی بر فیلتر کالمن تطبیقی (KF) به عنوان جایگزینی کارآمد برای حل مشکلات فوق معرفی شده است. این چارچوب با بهره‌گیری از ویژگی‌های منحصربه‌فرد فیلتر کالمن، مانند مدل‌سازی عدم قطعیت و یادگیری مرتبه دوم آنلاین، عمل می‌کند. به طور خاص، مقاله حاضر چارچوب یادگیری تقویتی چندعاملی با تفاوت زمانی کالمن تطبیقی (MAK-TD) و نسخه مبتنی بر نمایش جانشین آن، با نامMAK-SR ، را پیشنهاد می‌کند.

این چارچوب‌ها طبیعت پیوسته فضای عمل را که در محیط‌های چندعاملی با ابعاد بالا مرتبط است، در نظر گرفته و از تفاوت زمانی کالمن (KTD) برای مقابله با عدم قطعیت پارامترها بهره می‌برند. این روش‌ها از طریق مجموعه‌ای از آزمایش‌ها، که با استفاده از معیارهای OpenAI Gym برای MARL پیاده‌سازی شده‌اند، ارزیابی شده‌اند. نتایج تجربی نشان‌دهنده عملکرد برتر چارچوب‌های پیشنهادی MAK-TD/SR در مقایسه با پیشرفته‌ترین روش‌های موجود است.

واژه‌های کلیدی: یادگیری تقویتی، یادگیری تقویتی چند عامله، فیلتر کالمن

فهرست مطالب

عنوان صفحه

فهرست تصاویر و نمودارها

عنوان صفحه

فصل ۱: گزارش پروژه

۱.۱ مقدمه

یادگیری تقویتی (RL) به عنوان یکی از تکنیک‌های یادگیری ماشین (ML) هدف دارد تا با ایجاد یک سیاست کنترلی بهینه، رفتار تطبیقی در سطح انسانی را ارائه دهد. به طور کلی، هدف اصلی این تکنیک یادگیری از طریق آزمون و خطا با استفاده از تعاملات قبلی یک عامل خودکار با محیط پیرامون آن است. سیاست کنترلی بهینه (عملکرد بهینه) را می‌توان از طریق الگوریتم‌های RL و با استفاده از بازخوردهایی که محیط پس از هر عمل عامل ارائه می‌دهد، به دست آورد. دستیابی به بهینگی سیاست از طریق این روش با هدف افزایش پاداش در طول زمان صورت می‌گیرد.

در بسیاری از کاربردهای موفق RL، مانند بازی‌های Go و Poker، رباتیک، و رانندگی خودکار، معمولاً چندین عامل خودکار درگیر هستند. این امر به طور طبیعی در حوزه یادگیری تقویتی چندعاملی (MARL)قرار می‌گیرد، که اگرچه حوزه‌ای نسبتاً قدیمی است، اما اخیراً به دلیل پیشرفت‌های صورت‌گرفته در رویکردهای RL تک‌عاملی، دوباره مورد توجه قرار گرفته است.

در حوزه MARL، که تمرکز اصلی این مقاله است، چندین عامل تصمیم‌گیرنده در یک محیط مشترک تعامل دارند (همکاری یا رقابت می‌کنند) تا به یک هدف مشترک یا متضاد دست یابند.

۱.۲ شرح چکیده مقاله

در این مقاله، دو چارچوب جدید به نام‌های MAK-TD و MAK-SR معرفی شده‌اند که از ویژگی‌های فیلتر کالمن برای بهبود الگوریتم‌های تقویت یادگیری چندعامله استفاده می‌کنند.

دلیل اصلی طراحی این چارچوب‌ها حل مشکلات رایج در یادگیری تقویتی چندعامله (MARL) است. مشکلات کلیدی که این مقاله به آنها پرداخته است شامل موارد زیر است:

۱. **بیش‌برازش(Overfitting)** : روش‌های مبتنی بر شبکه‌های عصبی عمیق (DNN) معمولاً در مواجهه با داده‌های محدود دچار بیش‌برازش می‌شوند. استفاده از فیلتر کالمن در این چارچوب‌ها باعث مدلسازی دقیق‌تر عدم قطعیت و جلوگیری از این مشکل شده است.

۲. **حساسیت بالا به انتخاب پارامترها**: الگوریتم‌های مرسوم مانند DDPG و DQN نسبت به انتخاب پارامترها بسیار حساس هستند، که این امر می‌تواند باعث ناپایداری در عملکرد شود. چارچوب‌های پیشنهادی با استفاده از مکانیزم‌های تطبیقی، حساسیت به پارامترها را کاهش داده و پایداری بیشتری ارائه می‌دهند.

۳. **ناکارآمدی نمونه‌ها(Sample Inefficiency)** : یکی از چالش‌های اصلی MARL نیاز به تعداد زیادی نمونه برای یادگیری است. چارچوب‌های MAK-TD و MAK-SR با بهره‌گیری از یادگیری مرتبه دوم و بهینه‌سازی بهره‌وری نمونه، این مشکل را کاهش می‌دهند.

٤. **مدیریت فضاهای پیوسته و پیچیده**: محیط‌های چندعامله اغلب شامل فضاهای حالت و عمل بسیار پیچیده و پیوسته هستند. این چارچوب‌ها با استفاده از نمایه‌های شعاعی پایه (RBF) و مدلسازی مناسب، به بهبود یادگیری در چنین محیط‌هایی کمک می‌کنند.

این چارچوب‌ها مزایایی همچون مدلسازی عدم قطعیت و یادگیری آنلاین مرتبه دوم را ارائه می‌دهند.

۱. **MAK-TD**:

- این روش، مسائل مرتبط با پارامترهای نامشخص را از طریق ترکیب روش یادگیری اختلاف زمانی (TD) و فیلتر کالمن حل می‌کند.
- برای افزایش بهره‌وری از نمونه‌ها، از روش یادگیری خارج از سیاست (off-policy) استفاده شده است.

۲. **MAK-SR**:

- این روش بر مبنای نماینده جانشین (SR) طراحی شده که به یادگیری سریع‌تر سیاست‌ها در محیط‌های متغیر کمک می‌کند.
- استفاده از فیلتر کالمن در این چارچوب به کاهش نیاز به حافظه و زمان یادگیری کمک می‌کند.

این چارچوب‌ها از نمایه‌های شعاعی پایه (RBF) برای کاهش پیچیدگی محاسباتی و افزایش دقت بهره می‌برند. نتایج آزمایش‌ها نشان می‌دهد که این روش‌ها عملکرد بهتری نسبت به روش‌های مرسوم مبتنی بر شبکه عصبی دارند، به خصوص در شرایطی که منابع آموزشی محدود هستند.

کاربردهای چارچوب‌های پیشنهادی:

۱. **محیط‌های متغیر**: روش MAK-SR به دلیل انعطاف‌پذیری خود در تغییرات محیطی می‌تواند در مسائلی که مدل پاداش تغییر می‌کند بسیار مفید باشد.

۲. **تعادل بین کاوش و بهره‌برداری(Exploration vs. Exploitation)** : این چارچوب‌ها از یک مکانیزم یادگیری فعال برای بهره‌گیری از عدم قطعیت در انتخاب عمل استفاده می‌کنند، که بهبود عملکرد و افزایش پاداش تجمعی را به دنبال دارد.

۳. **کاربردهای بلادرنگ**: به دلیل نیاز کمتر به حافظه و محاسبات بهینه‌تر، این چارچوب‌ها برای سیستم‌هایی که نیاز به تصمیم‌گیری سریع دارند مناسب هستند.

۱.۳ پرسش‌های پژوهشی و چالش‌ها

در این مقاله، ما به دنبال پاسخ به پرسش‌های پژوهشی زیر هستیم:

- چگونه می‌توان با مشکلاتی همچون بیش‌برازش، حساسیت بالا به انتخاب پارامترها، و ناکارآمدی نمونه‌ها در MARL که معمولاً با راه‌حل‌های مبتنی بر DNN همراه هستند، مقابله کرد؟
- چگونه می‌توان تغییر در مدل پاداش را برای یادگیری تابع ارزش زیربنایی به درستی مدیریت کرد و عدم قطعیت در نمایش جانشین (SR) را ثبت کرد؟

- چگونه می‌توان از یادگیری تقویتی چندعاملی مبتنی بر تفاوت زمانی کالمن (KTD) در قالب SRبهره‌برداری کرد؟
- چگونه می‌توان تعادل میان اکتشاف و بهره‌برداری در MARL را یافت؟

برای پاسخ به سؤالات فوق، با چالش‌های زیر روبه‌رو شدیم:

۱. یادگیری توابع پاداش محلی و مدیریت کمبود دانش پیشین درباره نویز مشاهدات و توابع نگاشت آن.

۲. انتخاب پارامترهای KF برای یادگیری تابع پاداش که عملکرد آن به شدت به این مقادیر وابسته است.

۳. کدگذاری حالت‌های پیوسته به بردارهای ویژگی و نمایش تابع پاداش به عنوان یک تابع خطی از ویژگی‌های استخراج‌شده.

۴. ترکیب روش یادگیری اختلاف زمانی کالمن با یادگیری SR

۵. ثبت عدم قطعیت مرتبط با SR و محاسبه تابع ارزش بر اساس مقادیر SR و تابع پاداش یادگرفته‌شده.

۶. مدیریت توازن میان کاوش و بهره‌برداری.

۱.۴ بررسی کارهای پیشین، پیشینه پژوهش و نوآوری‌ها

به طور سنتی، الگوریتم‌های یادگیری تقویتی به دو دسته تقسیم می‌شوند:

۱. **رویکردهای بدون مدل (Model-Free یا MF)**: در این روش‌ها از مسیرهای نمونه برای یادگیری تابع ارزش استفاده می‌شود.

۲. **رویکردهای مبتنی بر مدل (Model-Based یا MB)**: در این روش‌ها، توابع پاداش با استفاده از درخت‌های جستجو یا برنامه‌ریزی پویا تخمین زده می‌شوند.

روش‌های MF معمولاً در تطبیق با تغییرات محلی در توابع پاداش ناکارآمد هستند، در حالی که روش‌های MB با وجود تطبیق سریع با تغییرات محیط، هزینه محاسباتی بالایی دارند. برای حل این مشکلات،

روش‌های مبتنی بر نماینده جانشین (SR) به عنوان یک دسته‌بندی جایگزین مطرح شده‌اند. این روش‌ها انعطاف‌پذیری روش‌های MB را با کارایی محاسباتی روش‌های MF ترکیب می‌کنند.

در روش‌های SR، هم پاداش فوری مورد انتظار و هم اشغال مورد انتظار آتی حالت‌ها (که به عنوان SR شناخته می‌شود) یاد گرفته می‌شوند. این رویکرد با تغییر شرایط پاداش، تنها نیاز به یادگیری تابع پاداش دارد و ارزیابی سیاست‌ها را تسریع می‌کند. با این حال، در محیط‌های چندعامله که تعداد زیادی حالت پیوسته وجود دارد، محاسبه دقیق تابع ارزش امکان‌پذیر نیست و روش‌های سنتی مانند یادگیری Q تک‌عامله یا گرادیان سیاست قابل استفاده نیستند.

برای استفاده از روش‌های مبتنی بر SR در MARL، تقریب تابع ارزش ضروری است. تخمین‌های خطی و غیرخطی از این تابع به عنوان رویکردهای ممکن معرفی شده‌اند. تخمین‌های غیرخطی مانند شبکه‌های عصبی عمیق (DNN) به حل مسائل پیچیده کمک کرده‌اند، اما مشکلاتی مانند بیش‌برازش، حساسیت بالا به پارامترها، ناکارآمدی نمونه‌ها، و نیاز به داده‌های زیاد را به همراه دارند.

مشکلات اصلی روش‌های مبتنی بر شبکه عصبی عمیق (DNN) شامل **بیش‌برازش، حساسیت زیاد به انتخاب پارامترها، ناکافی بودن نمونه‌ها** و **نیاز به تعداد زیادی اپیزود برای آموزش مدل‌ها** هستند. در مقابل، تخمین‌گرهای خطی با تبدیل مسئله تخمین به یک مسئله محاسبه وزن، ابزارهای مناسبی برای ترکیب تخمین‌های محلی ارائه می‌دهند. از آنجا که تخمین‌گرهای خطی نسبت به همتایان غیرخطی خود بهتر درک شده‌اند، بررسی همگرایی آن‌ها ساده‌تر است. نمونه‌هایی از تخمین‌گرهای خطی شامل **کنترل‌کننده‌های مدل مفصلی مخچه‌ای (CMAC)** و **توابع پایه شعاعی (RBF)** هستند.

با این حال، فرآیند تخمین توابع می‌تواند به شیوه‌ای تدریجی و پیوسته بهتر نمایش داده شود. پارامترهای توابع پایه شعاعی (RBF) معمولاً بر اساس دانش پیشین از مسئله مشخص می‌شوند، اما این پارامترها را می‌توان با استفاده از انتقال‌های مشاهده‌شده برای افزایش خودمختاری روش تطبیق داد. در این زمینه، روش‌های **آنتروپی متقاطع** و **گرادیان نزولی** می‌توانند برای تطبیق پارامترها به کار گرفته شوند. ثبات روش‌های مبتنی بر گرادیان نزولی با استفاده از یک روش محدودکننده بهبود یافته است.

پس از تأیید ساختار تابع ارزش، می‌توان از روش‌های زیر برای آموزش تخمین‌گر تابع ارزش استفاده کرد:

۱. **روش‌های بوت‌استرپینگ** مانند **فیلتر کالمن نقطه ثابت (FPKF)**

۲. **تکنیک‌های باقی‌مانده** مانند **اختلاف زمانی کالمن (KTD)** و **اختلاف زمانی فرآیند گوسی (GPTD).**

۳. **روش‌های نقطه ثابت پیش‌بینی‌شده** مانند **اختلاف زمانی حداقل مربعات.(LSTD)**

روش KTD به دلیل ارائه تخمین مینیموم خطای میانگین مربعات (MMSE) و مدلسازی عدم قطعیت، یک تکنیک برجسته است. این روش از مزایای اطلاعات عدم قطعیت برای افزایش بهره‌وری نمونه‌ها استفاده می‌کند. با این حال، KTD به دانش پیشین درباره پارامترهای فیلتر (مانند کوواریانس نویز فرآیند و مدل اندازه‌گیری) نیاز دارد که در شرایط واقعی به راحتی در دسترس نیست. تخمین پارامتر در زمینه فیلتر کالمن یک مسئله به خوبی مطالعه‌شده است که طرح‌های تطبیقی متعددی برای آن توسعه یافته‌اند.

روش‌های موجود چندمدله (MMAE) و روش‌های تطبیقی مبتنی بر نوآوری برای تغییر حالت سیستم‌ها توسعه یافته‌اند. کارایی این روش‌ها نشان داده شده است. با این حال، تعمیم این روش‌ها برای حل مسائل MARLآسان نیست.

در روش‌های پیشنهادی که یادگیری اختلاف زمانی کلاسیک را با DNNها ترکیب می‌کنند، عدم قطعیت تابع ارزش و بازنمایی جانشین مورد مطالعه قرار نگرفته است. برای مدیریت این عدم قطعیت، ترکیبی مناسب از بهره‌برداری و کاوش باید استفاده شود. کاوش می‌تواند از اطلاعات عدم قطعیت به دو روش بهره ببرد: افزودن تصادفی‌سازی به تابع ارزش و تغییر به سمت انتخاب اعمال نامطمئن.

پیشینه پژوهش: تحقیقات ما در زمینه راه‌حل‌های یادگیری تقویتی مبتنی بر پردازش سیگنال با معرفی روشMM-KTD آغاز شد. این روش یک رویکرد **اختلاف زمانی کالمن چندمدله (MM-KTD)** برای محیط‌های تک‌عامله با فضای حالت پیوسته است. سپس رویکرد AKF-SR به‌عنوان یک روش بازنمایی جانشین مبتنی بر فیلتر کالمن تطبیقی برای سناریوهای تک‌عامله توسعه یافت. این مقاله بر گسترش این رویکردها به سناریوهای چندعامله با فضای حالت ناهمگن و پیوسته تمرکز دارد.

مشارکت‌های مقاله:

- **چارچوب MAK-TD :** یک چارچوب یادگیری اختلاف زمانی کالمن تطبیقی چندعامله.

- **چارچوب MAK-SR :** نسخه مبتنی بر بازنمایی جانشین از چارچوب.MAK-TD

این چارچوب‌ها فضای عمل پیوسته مرتبط با محیط‌های چندعامله با ابعاد بالا را در نظر گرفته و از KTD برای مدیریت عدم قطعیت پارامترها بهره می‌برند. فرآیند یادگیری بازنمایی جانشین (SR) در این چارچوب‌ها به‌عنوان یک مسئله فیلترینگ مدل‌سازی شده است. هدف این است که از مزایای ذاتی فیلتر کالمن شامل یادگیری آنلاین مرتبه دوم، تخمین عدم قطعیت، و مدیریت محیط‌های غیرایستا بهره‌برداری شود.

روش‌ها:

۱. **MAK-TD:** این چارچوب برای جبران اطلاعات ناکافی درباره پارامترهای کلیدی فیلتر، مانند کوواریانس نویز اندازه‌گیری، طراحی شده است. از روش Q-learning خارج از سیاست برای یادگیری سیاست بهینه و بهبود بهره‌وری نمونه استفاده می‌شود.

۲. **MAK-SR:** فرآیند یادگیری بازنمایی جانشین در قالب فیلترینگ با استفاده از KTD گسترش یافته است. این روش حافظه و زمان مورد نیاز برای یادگیری بازنمایی جانشین را کاهش می‌دهد و حساسیت به پارامترها را نسبت به روش‌های مبتنی بر شبکه عصبی کاهش می‌دهد.

۳. **استفاده از MMAE و گرادیان نزولی:** این ترکیب برای تخمین توابع پاداش محلی استفاده می‌شود و حساسیت به دانش پیشین درباره پارامترهای کلیدی فیلتر را کاهش می‌دهد.

۴. **مکانیزم یادگیری فعال:** این مکانیزم برای ایجاد تعادل بین کاوش و بهره‌برداری طراحی شده است. اطلاعات عدم قطعیت از تابع ارزش حاصل از یادگیری بازنمایی جانشین برای بهبود عملکرد استفاده می‌شود.

نوآوری:

- ادغام یادگیری اختلاف زمانی کالمن، تخمین تطبیقی چندمدله، و بازنمایی جانشین برای حل مسائل یادگیری تقویتی چندعامله.
- کاهش بیش‌برازش و حساسیت به انتخاب پارامترها.
- استفاده از مکانیزم یادگیری فعال برای بهره‌برداری از اطلاعات عدم قطعیت و بهبود عملکرد.

نتایج:

برای ارزیابی چارچوب‌های MAK-TD وMAK-SR ، از یک نسخه چندعامله OpenAI Gym برای شبیه‌سازی سناریوهای همکاری، رقابت و تعامل مختلط استفاده شده است. آزمایش‌ها نشان دادند که این چارچوب‌ها عملکرد بهتری نسبت به روش‌های موجود دارند.

۱.۵ فرموله سازی مسئله: مروری بر یادگیری تقویتی

برای ارائه پیش‌زمینه‌ای که برای توسعه چارچوب‌های پیشنهادی MAK-TD/SR مورد نیاز است، در این بخش مروری بر تکنیک‌های یادگیری تقویتی تک‌عاملی و چندعاملی ارائه می‌شود.

۱. یادگیری تقویتی تک‌عاملی(Single-Agent Reinforcement Learning)

در سناریوهای متعارف RL، معمولاً یک عامل در یک محیط ناشناخته قرار می‌گیرد و اقدامات خودکار انجام می‌دهد تا پاداش تجمعی خود را به حداکثر برساند. در این سناریوها، عامل تعاملات خود را با محیط از یک حالت اولیه s_0 آغاز می‌کند و این تعاملات تا رسیدن به یک حالت پایانی تعریف‌شده s_T ادامه می‌یابد.

مجموعه‌ای از اقدامات ممکن A تعریف شده است که عامل می‌تواند با دنبال کردن یک سیاست بهینه، اقدامات بالقوه خود را انتخاب کند. به عبارت دیگر، عامل با توجه به حالت فعلی خود $s_k \in S$ ، سیاستی را دنبال می‌کند که با π_k نشان داده می‌شود و اقدام $a_k \in A$ را در زمان k انجام می‌دهد. پس از انجام اقدام، عامل بر اساس احتمال انتقال $P(s_{k+1}|s_k,a_k)$ به یک حالت جدید $s_{k+1} \in S$ منتقل شده و پاداشی به مقدار $r_k \in R$ دریافت می‌کند.

یک عامل کاهش‌دهنده $\gamma \in (0,1)$ برای ترکیب پاداش‌های آینده استفاده می‌شود تا تعادلی میان پاداش‌های آنی و آینده برقرار کند. به طور خلاصه، یک فرآیند تصمیم‌گیری مارکوفی (MDP) که با مجموعه پنج‌تایی $\{S,A,P_a,R,\gamma\}$ نمایش داده می‌شود، به طور معمول به عنوان مدل ریاضی زیربنایی برای فرآیند RL استفاده می‌شود.

هدف اصلی یادگیری یک سیاست بهینه π^* است که حالت‌ها را به اقدامات نگاشت می‌کند و با حداکثرسازی مجموع انتظاری پاداش‌های تخفیف‌یافته تعریف می‌شود. این هدف با استفاده از تابع ارزش حالت-اقدام به صورت زیر حاصل می‌شود:

$$Q_\pi(s,a) = \mathbb{E}\left\{ \sum_{k=0}^{T} \gamma^k r_k | s_0 = s, a_0 = a, a_k = \pi(s_k) \right\}$$

در اینجا، $\mathrm{E}\{\cdot\}$ نشان‌دهنده عملگر امید ریاضی است. در مرحله یادگیری، سیاست فعلی برای انجام اقدام استفاده می‌شود. پس از رسیدن به همگرایی، عامل می‌تواند از سیاست بهینه برای انجام اقدامات مورد نیاز استفاده کند:

$$a_k = \arg\max_{a \in \mathcal{A}} Q_{\pi^*}(s_k, a)$$

این توضیحات مروری بر RL را کامل می‌کند. در ادامه، یادگیری تفاوت زمانی (TD) به عنوان یکی از اجزای سازنده چارچوب‌های MAK-TD/SR بررسی خواهد شد.

۲. یادگیری تفاوت زمانی خارج از سیاست(Off-Policy Temporal Difference Learning)

با انجام یک اقدام و انتقال از یک حالت به حالت دیگر، بر اساس معادله و روش به‌روزرسانی بلمن، تابع ارزش به‌تدریج با استفاده از انتقالات نمونه‌ای به‌روزرسانی می‌شود. این فرآیند به‌عنوان به‌روزرسانی تفاوت زمانی (TD) شناخته می‌شود.

دو رویکرد برای به‌روزرسانی سیاست وجود دارد: **یادگیری مبتنی بر سیاست جاری** و **یادگیری خارج از سیاست جاری**. در روش اول، سیاست جاری برای انتخاب اقدام استفاده می‌شود. به عنوان مثال، الگوریتم SARSA یک روش مبتنی بر سیاست جاری است که شبکه را به صورت زیر بهینه می‌کند:

$$Q_\pi(s_k, a_k) = Q_\pi(s_k, a_k) + \alpha \left(r_k + \gamma\, Q_\pi(s_{k+1}, a_{k+1}) - Q_\pi(s_k, a_k) \right)$$

که در آن، α نرخ یادگیری و $Q_\pi(s_k,a_k)$ تابع ارزش حالت-اقدام است.

روش‌های مبتنی بر سیاست جاری کارآمدی کمتری در انتخاب نمونه دارند، زیرا به‌روزرسانی تابع ارزش از طریق سیاست جاری به جای استفاده از سیاست بهینه انجام می‌شود. در مقابل، **راه‌حل‌های خارج از سیاست**

جاری، مانند یادگیری Q ، اطلاعات دریافت‌شده از سیاست‌های قبلی را برای به‌روزرسانی سیاست و دستیابی به سیاست جدید (بهره‌برداری) به کار می‌گیرند.

فرآیند یادگیری Q به شکل زیر بر اساس معادله بهینه بلمن تعریف می‌شود:

$$Q_{\pi^*}(s_k, a_k) = Q_{\pi^*}(s_k, a_k) + \alpha\left(r_k + \gamma \max_{a \in \mathcal{A}} Q_{\pi^*}(s_{k+1}, a) - Q_{\pi^*}(s_k, a_k)\right)$$

سیاست بهینه π^* از طریق رویکرد حریصانه زیر قابل دستیابی است:

$$V_{\pi^*}(s) = \max_{a \in \mathcal{A}} Q_{\pi^*}(s_k, a)$$

پس از همگرایی، اقدامات می‌توانند بر اساس سیاست بهینه انتخاب شوند:

$$a_k = \arg\max_{a \in \mathcal{A}} Q_{\pi^*}(s_k, a)$$

۳. تنظیمات چندعاملی(Multi-Agent Setting)

در زمینه یادگیری تقویتی چندعاملی(MARL) ، یک سناریو با N عامل در نظر گرفته می‌شود که هر کدام دارای مشاهدات، اقدامات و حالت‌های محلی خود هستند. به عبارت دیگر، عامل i (برای $1\le i\le N$) از سیاست $\pi^{(i)}$ استفاده می‌کند که تابعی از حاصل‌ضرب کارتزین مجموعه اقدام‌های محلی $A^{(i)}$ و مجموعه مشاهده‌های محلی $Z^{(i)}$ به یک عدد حقیقی در بازه $[0,1]$ است.

مجموعه بزرگتر $S=\{S^{(1)},\dots,S^{(N)}\}$ تمام حالت‌های محلی $S^{(i)}$ برای $1\le i\le N$ را به صورت جمعی نمایش می‌دهد. به طور مشابه، مجموعه‌های $A=\{A^{(1)},\dots,A^{(N)}\}$ و $Z=\{Z^{(1)},\dots,Z^{(N)}\}$ به ترتیب تمامی اقدامات و مشاهدات محلی را به صورت یکپارچه نمایش می‌دهند.

هر عامل تصمیمات خود را به صورت محلی اتخاذ می‌کند و از تابع انتقال $T : S\times A^{(1)}\times\cdots\times A^{(N)}\to S^2$ پیروی می‌کند. بنابراین، هر اقدام به صورت محلی انجام می‌شود و منجر به اندازه‌گیری جدید و دریافت پاداش محلی $r^{(i)}:S\times A^{(i)}\to R$ می‌شود. هدف اصلی هر عامل، حداکثرسازی بازده مورد انتظار محلی $R^{(i)}=\sum\gamma^t(r^{(i)})^t$ در یک بازه پایانی T با استفاده از یک عامل کاهش γ از پیش تعیین‌شده است.

چالش‌های مدل‌های سنتی

مدل‌های سنتی مانند گرادیان سیاست یا یادگیری Q برای سناریوهای MARL مناسب نیستند، زیرا سیاست یک عامل در طول فرآیند آموزش تغییر می‌کند و محیط از دیدگاه عامل خاص، غیرایستا می‌شود.

به همین دلیل، پلتفرم‌های اخیر برای سناریوهای چندعاملی از استراتژی‌های دیگری استفاده می‌کنند، جایی که مشاهدات خود عامل (که در زمان اجرا به عنوان اطلاعات محلی شناخته می‌شود) برای یادگیری سیاست‌های محلی بهینه به کار گرفته می‌شود.

معمولاً این روش‌ها هیچ الگوی ارتباطی خاصی بین عوامل یا هیچ مدل مشتقی از دینامیک محیط را در نظر نمی‌گیرند. علاوه بر این، این مدل‌ها از تعاملات مختلف میان عوامل، از همکاری تا رقابت یا ترکیبی از هر دو پشتیبانی می‌کنند. در این زمینه، یک تطبیق بین اجرای غیرمتمرکز و آموزش متمرکز ایجاد شده است تا گام‌های آموزش سیاست با داده‌های بیشتری تغذیه شوند و فرآیند یافتن سیاست بهینه تسریع یابد.

۴. نمایش جانشین در تنظیمات چندعاملی(Multi-Agent Successor Representation)

به جای یادگیری مستقیم تابع ارزش Q(s,a) بازنمایی جایگزین M_π یاد گرفته می‌شود. این بازنمایی نشان‌دهنده انتظار اشغال وضعیت‌های آینده (future state occupancy) با توجه به سیاست جاری π است.

$$\boldsymbol{M}_{\pi^{(i)}}(\boldsymbol{s}^{(i)}, \boldsymbol{s'}^{(i)}, a^{(i)}) = \mathbb{E}\left[\sum_{k=0}^{T} \gamma^k \mathbb{1}[\boldsymbol{s}_k^{(i)} = \boldsymbol{s'}^{(i)}] | \boldsymbol{s}_0^{(i)} = \boldsymbol{s}^{(i)}, a_0^{(i)} = a^{(i)}\right]$$

که در آن $\mathbb{1}[\cdot]$ برابر با ۱ است اگر $s_k^{(i)}=s'^{(i)}$، وگرنه مقدار آن صفر است.

SR را می‌توان به صورت یک ماتریس $N_{s(i)} \times N_{s(i)}$ نمایش داد زمانی که فضای حالت گسسته باشد. روش بازگشتی مورد استفاده در معادله‌ی TD می‌تواند برای به‌روزرسانی SR به کار گرفته شود:

$$\boldsymbol{M}^{\text{new}}_{\pi^{(i)}}(\boldsymbol{s}_k^{(i)}, \boldsymbol{s'}^{(i)}, a_k^{(i)}) = \boldsymbol{M}^{\text{old}}_{\pi^{(i)}}(\boldsymbol{s}_k^{(i)}, \boldsymbol{s'}^{(i)}, a_k^{(i)}) + \alpha\Big(\mathbb{1}[\boldsymbol{s}_k^{(i)} = \boldsymbol{s'}^{(i)}] + \gamma \boldsymbol{M}_{\pi^{(i)}}(\boldsymbol{s}_{k+1}^{(i)}, \boldsymbol{s'}^{(i)}, a_{k+1}^{(i)}) - \boldsymbol{M}^{\text{old}}_{\pi^{(i)}}(\boldsymbol{s}_k^{(i)}, \boldsymbol{s'}^{(i)}, a_k^{(i)})\Big)$$

۱.۶ فرموله سازی مسئله: فیلترهای کالمن

در این بخش به صورت خلاصه به بیان فیلترهای کالمن پرداخته می‌شود.

با فرض دینامیک گسسته زیر برای یک سیستم:

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1}$$

$$y_k = H_k x_k + v_k$$

و با توجه به اینکه نویز مشاهده از نویز فرایند مستقل است:

$$E(w_k) = E(v_k) = 0 \quad E(w_k w_j^T) = Q_k \delta_{k-j} \quad E(v_k v_j^T) = R_k \delta_{k-j} \quad E(v_k w_j^T) = 0$$

بر اساس آنچه که در تئوری تخمین و فیلترهای بیزین مطالعه می‌گردد، همچنین براساس انتشار کواریانس و میانگین، گام پیش‌بینی در فیلتر کالمن به صورت زیر بدست می‌آید:

$$\bar{x}_k = F_{k-1}\bar{x}_{k-1} + G_{k-1}u_{k-1}$$

$$P_k = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$$

با الگو گیری از فیلترهای LMMSE، گام بروز رسانی نیز به صورت زیر بدست خواهد آمد:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad = P_k^+ H_k^T R_k^{-1}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-)$$

$$P_k^+ = (I - K_k H_k)P_k^- = (I - K_k H_k)P_k^-(I - K_k H_k)^T + K_k R_k K_k^T$$

در گام بروزرسانی K از فرم اول آن، و در بروز رسانی P از فرم دوم آن در این مقاله استفاده می‌گردد. فرم دوم P، علیرغم محاسبات بیشتر تضمین می‌دهد که P به صورت مثبت معین بدست می‌آید.

۱.۷ چارچوب MAK-TD

همان‌طور که قبلاً ذکر شد، چارچوب MAK-TD یک راه‌حل یادگیری خارج از سیاست مبتنی بر فیلتر کالمن برای شبکه‌های چندعاملی است. به طور خاص، با بهره‌گیری از روش TD که در معادله زیر نمایش داده شده است، تابع ارزش بهینه مربوط به عامل i (برای $1\leq i\leq N$) را می‌توان از طریق برآورد یک‌مرحله‌ای آن تقریب زد:

$$Q_{\pi^{(i)*}}(s_k^{(i)}, a_k^{(i)}) \approx r_k^{(i)} + \gamma \max_{a^{(i)}\in\mathcal{A}} Q_{\pi^{(i)*}}(s_{k+1}^{(i)}, a^{(i)})$$

با تغییر ترتیب متغیرها، پاداش در هر لحظه را می‌توان به‌عنوان یک مشاهده نویزدار مدل‌سازی کرد:

$$r_k^{(i)} = Q_{\pi^{(i)*}}(s_k^{(i)}, a_k^{(i)}) - \gamma \max_{a^{(i)}\in\mathcal{A}} Q_{\pi^{*}}(s_{k+1}^{(i)}, a^{(i)}) + v_k^{(i)}$$

که در آن $v_k^{(i)}$ به‌عنوان یک توزیع نرمال با میانگین صفر و واریانس $R^{(i)}$ مدل‌سازی می‌شود.

تقریب تابع ارزش محلی

برای در نظر گرفتن فضای حالت محلی هر عامل، از توابع پایه محلی برای تقریب تابع ارزش هر عامل استفاده می‌کنیم. بنابراین، تابع ارزش زیر برای عامل i (برای $1\leq i\leq N$) تشکیل می‌شود:

$$Q_{\pi^{(i)}}(s_k^{(i)}, a_k^{(i)}) = \boldsymbol{\phi}(s_k^{(i)}, a_k^{(i)})^T \boldsymbol{\theta}_k^{(i)}$$

که در آن $\phi(s^{(i)},a^{(i)})$ یک بردار از توابع پایه را نمایش می‌دهد، $\pi(i)$ سیاست مربوط به عامل i است و در نهایت $\theta_k^{(i)}$ بردار وزن‌ها را نشان می‌دهد.

با جایگذاری معادله فوق در معادله پاداش نویزدار، مدل مشاهده خطی زیر به دست می‌آید:

$$r_k^{(i)} = [\boldsymbol{h}_k^{(i)}]^T \boldsymbol{\theta}_k^{(i)} + v_k^{(i)}$$

که در آن:

$$\boldsymbol{h}_k^{(i)} = \boldsymbol{\phi}(s_k^{(i)}, a_k^{(i)}) - \gamma \max_{a^{(i)}\in\mathcal{A}} \boldsymbol{\phi}(s_{k+1}^{(i)}, a^{(i)})$$

تخمین وزن‌ها با فیلتر کالمن

اگر معادلات بالا برای r را با دینامیک خروجی سیستم گسسته معادل سازی کنیم، می‌توانیم برای فاز پیش‌بینی و بروز رسانی از معادلات کالمن استفاده کنیم.

برای تقریب وزن محلی $\theta_k^{(i)}$:

۱. از پاداش مشاهده‌شده استفاده می‌کنیم، که از انتقال از حالت $S_k^{(i)}$ به $S_{k+1}^{(i)}$ به دست آمده است.

۲. با توجه به اینکه واریانس نویز اندازه‌گیری از پیش مشخص نیست، از روش تطبیقی MMAE استفاده می‌کنیم و آن را با M مقدار مختلف $(R_j^{(i)})$ نمایش می‌دهیم، که $1 \leq j \leq M$ است.

در نتیجه، ترکیبی از M فیلتر کالمن برای تخمین $\hat{\theta}_k^{(i)}$ بر اساس هر مقدار پیشنهادی آن استفاده می‌شود:

$$K_k^{j\,(i)} = P_{(\theta,k|k-1)}^{(i)} h_k^{(i)} \left(h_k^{T(i)} P_{(\theta,k|k-1)}^{(i)} h_k^{(i)} + R^{j(i)} \right)^{-1}$$

بردار وزن به‌روز شده به صورت زیر تعریف می‌شود:

$$\hat{\theta}_k^{j\,(i)} = \hat{\theta}_{(k|k-1)}^{(i)} + K_k^{j\,(i)} \left(r_k^{(i)} - h_k^{T(i)} \hat{\theta}_{(k|k-1)}^{(i)} \right)$$

و واریانس خطای تخمین به صورت زیر به‌روزرسانی می‌شود:

$$P_{\theta,k}^{j\,(i)} = \left(I - K_k^{j\,(i)} h_k^{T(i)} \right) P_{(\theta,k|k-1)}^{T\,(i)} \left(I - K_k^{j\,(i)} h_k^{T(i)} \right) + K_k^{j\,(i)} R^{j(i)} K_k^{j\,T(i)}$$

تخمین ترکیبی

برای ادغام تخمین‌های مختلف حاصل از M فیلتر کالمن، توزیع پسین کلی به صورت زیر محاسبه می‌شود:

$$P^{(i)}(\theta_k | Y_k) = \sum_{j=1}^{M} \omega^{j(i)} P^{(i)}(\theta_k^{(i)} | Y_k^{(i)}, R^{j(i)})$$

که در آن $\omega^{j(i)}$ وزن مربوط به هر فیلتر است و بر اساس احتمال مشاهده محاسبه می‌شود.

بردار توابع پایه‌ای نیز به صورت زیر تعریف می‌شود:

$$\phi(s_k^{(i)}) = \left[\phi_1(s_k^{(i)}), \phi_2(s_k^{(i)}), \ldots, \phi_{N_b-1}(s_k^{(i)}), \phi_{N_b}(s_k^{(i)}) \right]^T$$

که هر یک از المان‌های آن به صورت زیر بدست می‌آید:

$$\phi_n(s_k^{(i)}) = \exp\{\frac{-1}{2}(s_k^{(i)} - \mu_n^{(i)})^T {\Sigma_n^{(i)}}^{-1}(s_k^{(i)} - \mu_n^{(i)})\}$$

where $\mu_n^{(i)}$ and $\Sigma_n^{(i)}$ are the mean and covariance of $\phi_n(s_k^{(i)})$, for $(1 \leq n \leq N_b)$

اما لازم است صرفا ویژگی‌های مربوط به یک عمل خاص یعنی a، درون بردار حضور داشته باشد، بدین ترتیب، سایر المان‌ها با صفر جایگزین می‌شوند.

$$\boldsymbol{\phi}(s_k^{(i)}, a_k^{(i)}) = [\phi_{1,a_1}(s_k^{(i)}), \ldots \phi_{N_b,a_1}(s_k^{(i)}), \phi_{1,a_2}(s_k^{(i)}), \ldots \phi_{N_b,a_{D^{(i)}}}(s_k^{(i)})]^T$$

$$\boldsymbol{\phi}(s_k^{(i)}, a_k^{(i)}) = [0, \ldots 0, \phi_1(s_k^{(i)}), \ldots, \phi_N(s_k^{(i)}), 0, \ldots 0,]^T$$

حال لازم است loss تعریف شود:

$$L_k^{(i)} = (\boldsymbol{\phi}^T(s_k^{(i)}, a_k)\,\boldsymbol{\theta}_k^{(i)} - r_k^{(i)})^2$$

$$\begin{aligned} \Delta\boldsymbol{\mu}^{(i)} &= -\frac{\partial L_k^{(i)}}{\partial \boldsymbol{\mu}^{(i)}} = -\frac{\partial L_k^{(i)}}{\partial Q_{\pi^{*(i)}}} \frac{\partial Q_{\pi^{*(i)}}}{\partial \boldsymbol{\phi}^{(i)}} \frac{\partial \boldsymbol{\phi}^{(i)}}{\partial \boldsymbol{\mu}^{(i)}} \\ \text{and } \Delta\boldsymbol{\Sigma}^{(i)} &= -\frac{\partial \boldsymbol{\Sigma}_k^{(i)}}{\partial \boldsymbol{\mu}^{(i)}} = -\frac{\partial L_k^{(i)}}{\partial Q_{\pi^{*(i)}}} \frac{\partial Q_{\pi^{*(i)}}}{\partial \boldsymbol{\phi}^{(i)}} \frac{\partial \boldsymbol{\phi}^{(i)}}{\partial \boldsymbol{\Sigma}^{(i)}} \end{aligned}$$

براساس loss تعریف شده، المان‌های بردار توابع پایه‌ای نیز بروزرسانی می‌گردد.

و درنهایت با الگوی زیر عمل انتخاب می‌شود.

$$\begin{aligned} a_k^{(i)} &= \arg\max_a \left(\boldsymbol{h}_k^{(i)}(s_k^{(i)}, a^{(i)}) R^{-1^{(i)}} \boldsymbol{h}_k^{T^{(i)}}(s_k^{(i)}, a^{(i)})\right) \\ &= \arg\max_a \left(\boldsymbol{h}_k^{(i)}(s_k^{(i)}, a^{(i)}) \boldsymbol{h}_k^{T^{(i)}}(s_k^{(i)}, a^{(i)})\right). \end{aligned}$$

مجموع توضیحات بالا در قالب الگوریتم زیر خلاصه می‌شود.

Algorithm 1 THE PROPOSED MAK-TD FRAMEWORK

Learning Phase:
Set $\boldsymbol{\theta}_0, \boldsymbol{P}_{\boldsymbol{\theta},0}, \boldsymbol{F}, \boldsymbol{\mu}_{n,i_d}, \boldsymbol{\Sigma}_{n,i_d}$ for $n = 1, 2, \ldots, N$ and $i_d = 1, 2, \ldots, D$
Repeat (for each episode):
  Initialize $\boldsymbol{s}_k$
  Repeat (for each agent i):
    While $\boldsymbol{s}_k^{(i)} \neq \boldsymbol{s}_T$ **do**:
      $a_k^{(i)} = \arg\max_a \left(\boldsymbol{h}_k^{(i)}(\boldsymbol{s}_k^{(i)}, a^{(i)}) \boldsymbol{h}_k^{T^{(i)}}(\boldsymbol{s}_k^{(i)}, a^{(i)}) \right)$
      Take action $a_k^{(i)}$, observe $\boldsymbol{s}_{k+1}^{(i)}, r_k^{(i)}$
      Calculate $\boldsymbol{\phi}^{(i)}(\boldsymbol{s}^{(i)}, a^{(i)})$ via Equations (22) and (23)
      $\boldsymbol{h}_k^{(i)}(\boldsymbol{s}_k^{(i)}, a_k^{(i)}) = \boldsymbol{\phi}^{(i)}(\boldsymbol{s}_k^{(i)}, a_k^{(i)}) - \gamma \arg\max_a \boldsymbol{\phi}^{(i)}(\boldsymbol{s}_{k+1}^{(i)}, a^{(i)})$
      $\hat{\boldsymbol{\theta}}_{(k|k-1)}^{(i)} = \boldsymbol{F}^{(i)} \hat{\boldsymbol{\theta}}_k^{(i)}$
      $\boldsymbol{P}_{(\boldsymbol{\theta},k|k-1)}^{(i)} = \boldsymbol{F}^{(i)} \boldsymbol{P}_{\boldsymbol{\theta},k-1}^{(i)} \boldsymbol{F}^{T^{(i)}} + \boldsymbol{Q}^{(i)}$
      for $j = 1 : M$ **do**:
        $\boldsymbol{k}_k^{j\,(i)} = \boldsymbol{P}_{(\boldsymbol{\theta},k|k-1)}^{(i)} \boldsymbol{h}_k^{(i)} (\boldsymbol{h}_k^{T^{(i)}} \boldsymbol{P}_{(\boldsymbol{\theta},k|k-1)}^{(i)} \boldsymbol{h}_k^{(i)} + R^{j^{(i)}})^{-1}$
        $\hat{\boldsymbol{\theta}}_k^{j\,(i)} = \hat{\boldsymbol{\theta}}_{(\boldsymbol{\theta},k|k-1)}^{(i)} + \boldsymbol{k}_k^{j\,(i)} (r_k^j - \boldsymbol{h}_k^{T^{(i)}} \hat{\boldsymbol{\theta}}_{(k|k-1)}^{(i)})$
        $\boldsymbol{P}_{\boldsymbol{\theta},k}^{(i)} = (\boldsymbol{I} - \boldsymbol{K}_k^{j\,(i)} \boldsymbol{h}_k^{T^{(i)}}) \boldsymbol{P}_{(\boldsymbol{\theta},k|k-1)}^{(i)} (\boldsymbol{I} - \boldsymbol{K}_k^{j\,(i)} \boldsymbol{h}_k^{T^{(i)}})^T + \boldsymbol{K}_k^{j\,(i)} R^j \boldsymbol{K}_k^{j\,T^{(i)}}$
      end for
      Compute the value of c and $w^{j^{(i)}}$ by using $\sum_{j=1}^{M} w^{j^{(i)}} = 1$ and Equation (19)
      $\hat{\boldsymbol{\theta}}_k^{(i)} = \sum_{j=1}^{M} w^{j^{(i)}} \hat{\boldsymbol{\theta}}_k^{j\,(i)}$
      $\boldsymbol{P}_{\boldsymbol{\theta}_k}^{(i)} = \sum_{j=1}^{M} \omega^{j^{(i)}} \left(\boldsymbol{P}_{\boldsymbol{\theta},k}^{j\,(i)} + (\hat{\boldsymbol{\theta}}^{j^{(i)}} - \hat{\boldsymbol{\theta}}^{(i)})(\hat{\boldsymbol{\theta}}^{j^{(i)}} - \hat{\boldsymbol{\theta}}^{(i)})^T \right)$
      RBFs Parameters Update:
      $L_k^{(i)} = (\boldsymbol{\phi}^T(\boldsymbol{s}_k^{(i)}, a_k)\, \boldsymbol{\theta}_k^{(i)} - r_k^{(i)})^2$
      if $L_k^{(i)\,\frac{1}{2}} (\boldsymbol{\theta}_k^{(i)\,T} \boldsymbol{\phi}(\cdot)) > 0$ **then**:
        Update $\boldsymbol{\Sigma}_{n,a_d}$ via Equation (29)
      else:
        Update $\boldsymbol{\mu}_{n,a_d}$ via Equation (30)
      end if
    end while
Testing Phase:
Repeat (for each trial episode):
  While $\boldsymbol{s}_k \neq \boldsymbol{s}_T$ **do**:
    Repeat (for each agent):
      $a_k = \arg\max_a \boldsymbol{\phi}(\boldsymbol{s}_k, a)^T \boldsymbol{\theta}_k$
      Take action a_k, and observe $\boldsymbol{s}_{k+1}, r_k$
      Calculate Loss S_k for all agents
  End While

۱.۸ چارچوب MAK-SR

این چارچوب بر مبنای بازنمایی جانشین (Successor Representation) طراحی شده است که هدف آن تخمین احتمال اشغال آینده وضعیت‌ها است. با این روش، تابع ارزش Q(s,a) به صورت ضرب داخلی بازنمایی جانشین و تابع پاداش تخمین زده می‌شود.

مدل‌سازی ریاضی

۱. **تعریف بازنمایی جانشین**: بازنمایی جانشین $M_\pi(s,a)$ به صورت زیر تعریف می‌شود:

$$\boldsymbol{M}_{\pi^{(i)}}(\boldsymbol{s}^{(i)},:,a^{(i)}) = \mathbb{E}\left[\sum_{k=0}^{T}\gamma^k \boldsymbol{\phi}(\boldsymbol{s}_k^{(i)},a_k^{(i)})|\boldsymbol{s}_0^{(i)}=\boldsymbol{s}^{(i)},a_0^{(i)}=a^{(i)}\right]$$

۲. **تقریب خطی بازنمایی جانشین**: ماتریس بازنمایی جانشین به صورت یک تابع خطی از ویژگی‌ها تقریب زده می‌شود:

$$\boldsymbol{M}_{\pi^{(i)}}(\boldsymbol{s}_k^{(i)},:,a_k^{(i)}) \approx \boldsymbol{M}_k\,\boldsymbol{\phi}(\boldsymbol{s}_k^{(i)},a_k^{(i)})$$

که M_k ماتریس وزن بازنمایی جانشین است.

۳. **به‌روزرسانی TD برای M_k**: بازنمایی جانشین با استفاده از TD به‌روزرسانی می‌شود:

$$\boldsymbol{M}^{\text{new}}_{\pi^{(i)}}(\boldsymbol{s}_k^{(i)},:,a_k^{(i)}) = \boldsymbol{M}^{\text{old}}_{\pi^{(i)}}(\boldsymbol{s}_k^{(i)},:,a_k^{(i)}) + \alpha\left(\boldsymbol{\phi}^{(i)}(\boldsymbol{s}_k^{(i)},a_k^{(i)}) + \gamma \boldsymbol{M}_{\pi^{(i)}}(\boldsymbol{s}_{k+1}^{(i)},:,a_{k+1}^{(i)}) - \boldsymbol{M}^{\text{old}}_{\pi^{(i)}}(\boldsymbol{s}_k^{(i)},:,a_k^{(i)})\right)$$

٤. **محاسبه تابع ارزش**: پس از تخمین M_k و بردار وزن پاداش θ ، تابع ارزش به صورت زیر محاسبه می‌شود:

$$Q(\boldsymbol{s}_k^{(i)},a_k^{(i)}) = \boldsymbol{\theta}_k^{(i)^T}\boldsymbol{M}(\boldsymbol{s}_k^{(i)},:,a_k^{(i)})$$

۵. **به‌روزرسانی با فیلتر کالمن**: مشابه چارچوب MAK-TD، فیلتر کالمن برای تخمین وزن‌های M_k استفاده می‌شود. مدل مشاهده و مدل حالت به صورت زیر تعریف می‌شوند:

$$\hat{\boldsymbol{\phi}}(\boldsymbol{s}_k^{(i)},a_k^{(i)}) = \boldsymbol{M}_k\,\boldsymbol{\phi}(\boldsymbol{s}_k^{(i)},a_k^{(i)}) + \boldsymbol{n}_k^{(i)}$$

$$\boldsymbol{m}_{k+1}^{(i)} = \boldsymbol{m}_k^{(i)} + \boldsymbol{\mu}_k^{(i)}$$

که m_k بردار وزن و μ_k نویز فرآیند است.

۶. **استفاده از RBF برای ویژگی‌ها**: ویژگی‌ها با استفاده از توابع پایه شعاعی (RBF) به صورت زیر مدل می‌شوند:

$$\phi_n(s_k^{(i)}) = \exp\{\frac{-1}{2}(s_k^{(i)} - \mu_n^{(i)})^T \Sigma_n^{(i)^{-1}} (s_k^{(i)} - \mu_n^{(i)})\}$$

که μ_n میانگین و Σ_n کوواریانس توابع پایه هستند.

الگوریتم آن نیز مشابه چارچوب قبلی است، فقط در برخی بخش‌ها تغییرات مشروح در بالا اعمال شده که الگوریتم آن را به فرم زیر تبدیل می‌کند.

Algorithm 2 THE PROPOSED MAK-SR FRAMEWORK

Learning Phase:
Initialize: $\theta_0, P_{\theta,0}, m_0, P_{M,0}, \mu_n$, and Σ_n for $n = 1, 2, \ldots, N$
Parameters: $Q_\theta, Q_M, \lambda_\mu, \lambda_\Sigma$, and $\{R_\theta^j, R_M^j\}$ for $j = 1, 2, \ldots, M$
Repeat (for each episode):
  Initialize s_k
  Repeat (for each agent i):
    While $s_k^{(i)} \neq s_T$ **do**:
      Reshape m_k into $L \times L$ to construct 2-D matrix M_k.
      $a_k^{(i)} = \arg\max_a \left(g_k^{(i)}(s_k^{(i)}, a) g_k^{(i)^T}(s_k^{(i)}, a^{(i)}) \right)$
      Take action $a_k^{(i)}$, observe $s_{k+1}^{(i)}$ and $r_k^{(i)}$.
      Calculate $\phi(s_k^{(i)}, a_k^{(i)})$ via Equations (23) and (25).
      Update reward weights vector: Perform MMAE to update $\theta_k^{(i)}$.
      Update SR weights vector: Perform KF on Equations (40) and (41) to update $m_k^{(i)}$.
      Update RBFs parameters: Perform RGD on the loss function L_k to update Σ_n and μ_n.
    end while

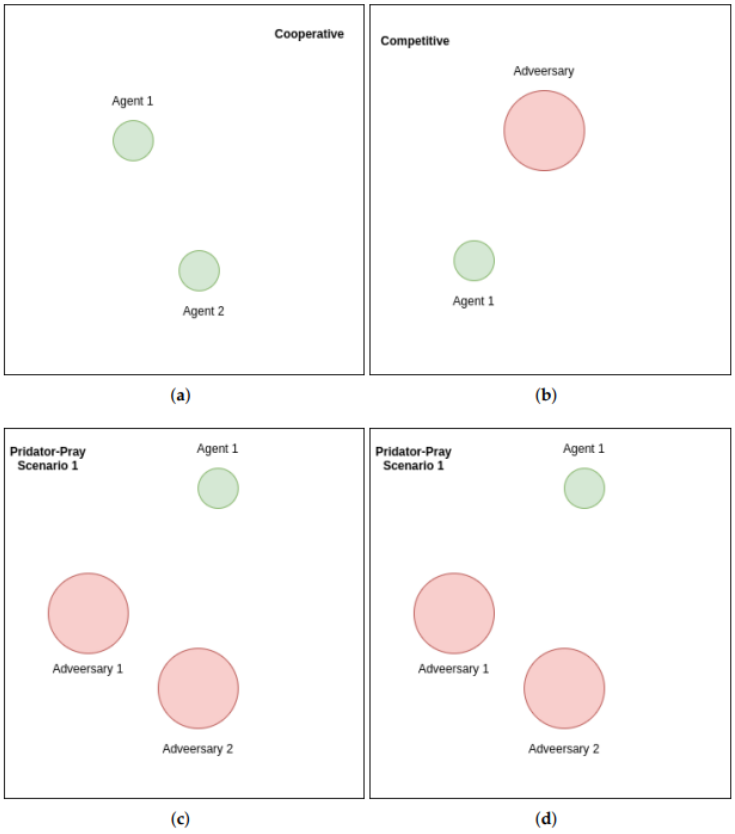
۱.۹ نتایج آزمایش

برای ارزیابی چارچوب‌های پیشنهادی MAK-TD وMAK-SR ، مجموعه‌ای از آزمایش‌ها در محیط‌های شبیه‌سازی‌شده انجام شده است. در این آزمایش‌ها، سناریوهای مختلف چندعاملی شامل همکاری، رقابت و ترکیب همکاری-رقابت بررسی شده است.

۱. محیط شبیه‌سازی(Simulation Environment)

برای ارزیابی عملکرد، از یک گسترش چندعاملی از OpenAI Gym استفاده شده است. این محیط شامل:

- **فضای دو بعدی با حالت‌های پیوسته**: که در آن عوامل می‌توانند اقدامات خود را در فضای پیوسته انجام دهند.
- **تعاملات چندعاملی مختلف**: شامل سناریوهای کاملاً همکاری، کاملاً رقابت و ترکیب همکاری-رقابت.



شکل ۱: تصویر محیط‌های شبیه سازی

۲. معیارهای ارزیابی(Evaluation Metrics)

عملکرد چارچوب‌های پیشنهادی با استفاده از معیارهای زیر ارزیابی شد:

- **پاداش تجمعی**: میزان پاداش کلی که عوامل در طول زمان دریافت می‌کنند.
- **کارایی نمونه**: تعداد نمونه‌های مورد نیاز برای رسیدن به یک سیاست بهینه.
- **حساسیت به انتخاب پارامترها**: میزان تأثیر تغییرات پارامترها بر عملکرد.

۳. سناریوهای آزمایشی(Experimental Scenarios)

۱. سناریوی همکاری(Cooperative Scenario)

در این سناریو، تمامی عوامل برای رسیدن به یک هدف مشترک تلاش می‌کنند. نتایج نشان داد:

- چارچوب‌های MAK-TD و MAK-SR عملکرد بهتری نسبت به روش‌های مبتنی بر شبکه‌های عصبی عمیق (DNN) داشتند.
- روش MAK-SR با استفاده از تخمین عدم قطعیت توانست سیاست بهینه را سریع‌تر بیاموزد.

۲. سناریوی رقابت(Competitive Scenario)

در این سناریو، عوامل اهداف متضادی را دنبال می‌کنند. نتایج حاکی از آن بود که:

- چارچوب‌های پیشنهادی توانستند سیاست‌هایی را بیاموزند که تعادل بهتری میان اکتشاف و بهره‌برداری ایجاد کند.
- در مقایسه با روش‌های مرسوم، چارچوب MAK-SR توانست عملکرد پایدارتری ارائه دهد.

۳. سناریوی ترکیبی(Mixed Cooperative-Competitive Scenario)

در این سناریو، برخی عوامل با یکدیگر همکاری می‌کنند، در حالی که سایر عوامل رقابت می‌کنند. نتایج نشان داد:

- چارچوب MAK-SR با استفاده از یادگیری فعال و تخمین SR توانست پاداش تجمعی بیشتری کسب کند.

- توانایی مدیریت تغییرات در مدل پاداش از نقاط قوت کلیدی MAK-SR در این سناریو بود.

۴. مقایسه با روش‌های دیگر(Comparison with State-of-the-Art Methods)

۱. شبکه‌های عصبی عمیق(DNNs)

روش‌های مبتنی بر DNN معمولاً از مشکل بیش‌برازش و ناکارآمدی نمونه‌ها رنج می‌برند. چارچوب-MAK TDو MAK-SR این مشکلات را با استفاده از فیلتر کالمن و مدل‌سازی عدم قطعیت کاهش دادند.

۲. روش‌های یادگیری Q و سیاست گرادیان

روش‌های سنتی مانند Q-Learning و گرادیان سیاست توانایی مدیریت تعاملات پیچیده چندعاملی را ندارند. چارچوب‌های پیشنهادی با استفاده از توابع پایه و تخمین SR به طور قابل‌توجهی عملکرد بهتری داشتند.

براساس معیار loss عملکرد چارچوب‌های پیشنهادی مطابق جدول زیر است که به وضوح چارچوب‌های پیشنهادی بهتر عمل کرده‌اند.

جدول ۱-۱: جدول عملکرد براساس loss

Table 1. Total loss averaged across all the episodes and for all the four implemented scenarios.

Environment	MAK-SR	MAK-TD	MADDPG	DDPG	DQN
Cooperation	8.93	2.4088	9649.84	10,561.16	10.93
Competition	0.43	4.9301	10,158.18	10,710.37	107.39
Predator–Prey 1v2	0.005	1.9374	6816.34	6884.33	8.21
Predator–Prey 2v1	8.87	1.2421	7390.18	6882.2	10.24

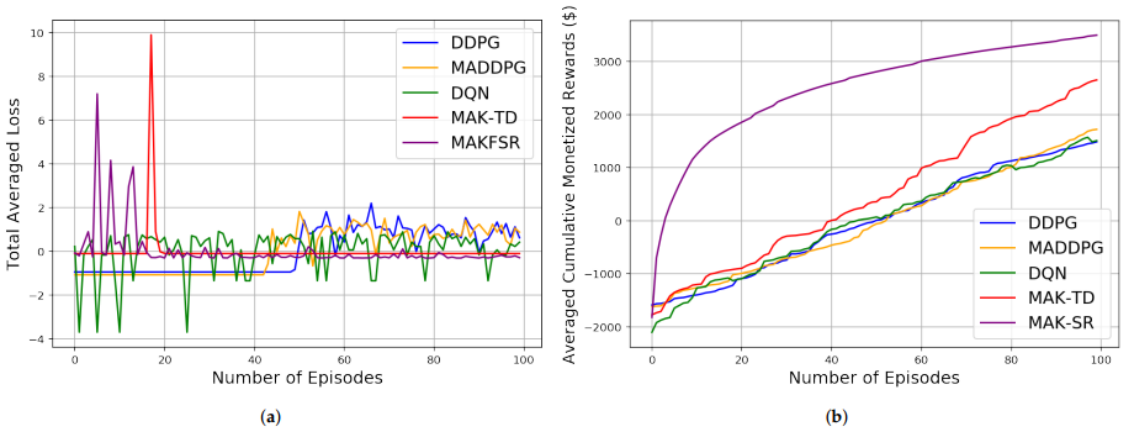
براساس پاداش دریافتی نیز عملکرد چارچوب‌های پیشنهادی بهتر از سایر روش‌ها بوده است.

جدول ۲-۱: جدول عملکرد براساس کل پاداش دریافتی عامل‌ها

Table 2. Total received reward by the agents averaged for all the four implemented scenarios.

Environment	MAK-SR	MAK-TD	MADDPG	DDPG	DQN
Cooperation	−16.0113	−23.0113	−69.28	−66.29	−39.96
Competition	−0.778	−13.358	−63.30	−61.34	−14.49
Predator–Prey 1v2	−0.0916	−13.432	−46.17	−20.53	−23.451
Predator–Prey 2v1	−0.081	−17.0058	−55.69	−49.41	−44.32

به صورت گرافیکی در نمودارهای زیر مقایسه‌ی روش‌ها قابل مشاهده است که مطابق با جداول بالاست.



شکل ۲: نمودارهای میانگین loss و میانگین پاداش‌های دریافتی

۵. تحلیل حساسیت(Sensitivity Analysis)

یکی از ویژگی‌های برجسته چارچوب‌های پیشنهادی، کاهش حساسیت به انتخاب پارامترها بود. در حالی که روش‌های دیگر به تنظیم دقیق پارامترها نیاز داشتند، چارچوب MAK-SR با استفاده از MMAE توانست پارامترهای کلیدی را به طور تطبیقی تنظیم کند.

۱.۱۰ گزارش کد، نتایج شبیه سازی

در این شبیه‌سازی، هدف مقایسه عملکرد سه الگوریتم یادگیری تقویتی است:

MAK-TD (Multi-agent Kalman Temporal Difference)

MAK-SR (Multi-agent Kalman Successor Representation)

DQN (Deep Q-Network)

این الگوریتم‌ها به‌طور گسترده در یادگیری تقویتی مورد استفاده قرار می‌گیرند و هر یک از آن‌ها روش‌های خاص خود را برای یادگیری و به‌روزرسانی مدل‌ها دارند. این شبیه‌سازی، چگونگی آموزش و مقایسه این الگوریتم‌ها را در یک محیط فرضی با استفاده از اپیزودها و مراحل مختلف شبیه‌سازی می‌کند.

اجزای کد

۱. کلاس KalmanFilter

کلاس KalmanFilter پیاده‌سازی فیلتر کالمن است که برای به‌روزرسانی وزن‌ها با استفاده از مشاهدات و خطای تفاوت زمانی (TD) در الگوریتم MAK-TD به کار می‌رود. فیلتر کالمن روشی برای پیش‌بینی و به‌روزرسانی است که به‌ویژه در مسائل یادگیری ماشین برای فیلتر کردن نویز و پیش‌بینی دقیق‌تر استفاده می‌شود.

- **ویژگی‌ها**:
  - Theta: بردار وزن‌ها است که به‌صورت ستونی ذخیره می‌شود. این بردار از وزن‌هایی تشکیل شده که مدل برای پیش‌بینی مقادیر Q استفاده می‌کند.
  - P: ماتریس کوواریانس که برای محاسبه ضریب کالمن به‌کار می‌رود.
- **متدها:**
  - update(h, r, R): این متد فیلتر کالمن را به‌روزرسانی می‌کند. ابتدا ضریب کالمن (Kalman Gain) محاسبه می‌شود که برای به‌روزرسانی وزن‌ها استفاده می‌شود. سپس وزن‌ها با استفاده از ضریب کالمن به‌روزرسانی می‌شوند. این به‌روزرسانی بر اساس خطای تفاوت زمانی (TD) و پاداش دریافتی صورت می‌گیرد.

ورودی‌ها:

- h: ویژگی‌ها (Feature Vector).
- r: پاداش دریافتی.
- R: پارامتر ثابت در فیلتر کالمن.

خروجی:

- Theta: بردار وزن‌های به‌روز شده.

۲. کلاس MAKTD (Multi-agent Kalman Temporal Difference)

کلاس MAKTD پیاده‌سازی الگوریتم **MAK-TD** است که ترکیبی از فیلتر کالمن و یادگیری تفاوت زمانی است. این الگوریتم برای پیش‌بینی پاداش‌های آینده با استفاده از ویژگی‌های فیلتر کالمن و الگوریتم یادگیری تقویتیTD به‌کار می‌رود.

- **ویژگی‌ها:**

- kf: نمونه‌ای از کلاس KalmanFilter است که برای به‌روزرسانی وزن‌ها و محاسبه ضریب کالمن استفاده می‌شود.
- gamma: عامل تخفیف که در یادگیری تفاوت زمانی برای تخمین پاداش‌های آینده به کار می‌رود.
- R: پارامتر ثابت در فیلتر کالمن.

- **متدها:**

- step(state, action, reward, next_state, phi): این متد در هر مرحله از یادگیری به‌روزرسانی وزن‌ها را انجام می‌دهد. ابتدا ویژگی‌ها از حالت-اقدام جاری استخراج می‌شود و سپس مقادیر Q برای تمامی اقدامات محاسبه شده و بهترین اقدام انتخاب می‌شود. در نهایت، وزن‌ها به‌روزرسانی می‌شوند.

ورودی‌ها:

- state: حالت جاری.
- action: اقدام انجام شده.
- reward: پاداش دریافتی.
- next_state: حالت بعدی.
- phi: تابع استخراج ویژگی که ویژگی‌ها را از حالت-اقدام استخراج می‌کند.

خروجی:

- Theta: بردار وزن‌های به‌روز شده.

۳. کلاس MAKSR (Multi-agent Kalman Successor Representation)

کلاس MAKSR پیاده‌سازی الگوریتم **MAK-SR** است که از نمای جانشینی (Successor Representation) برای پیش‌بینی پاداش‌های آینده استفاده می‌کند. نمای جانشینی یک ماتریس است که پیش‌بینی می‌کند وضعیت‌ها در آینده چگونه تغییر خواهند کرد.

- **ویژگی‌ها**:
  - gamma: عامل تخفیف که در محاسبه پاداش‌های آینده استفاده می‌شود.
  - alpha: نرخ یادگیری برای به‌روزرسانی ماتریس نمای جانشینی.
  - M: ماتریس نمای جانشینی که در آن هر عنصر نمایانگر پیش‌بینی تکامل وضعیت‌ها است.
- **متدها**:
  - update_sr(state, next_state, action): این متد ماتریس نمای جانشینی M را به‌روزرسانی می‌کند. این به‌روزرسانی بر اساس معادله بلمن انجام می‌شود که نشان‌دهنده نحوه تکامل وضعیت‌ها بر اساس اقدامات است.

- compute_q(reward, state, action): این متد مقدار Q برای یک جفت حالت-اقدام خاص را با استفاده از ماتریس نمای جانشینی محاسبه می‌کند.

ورودی‌ها:

- state: حالت جاری.
- next_state: حالت بعدی.
- action: اقدام انجام شده.
- reward: پاداش دریافتی.

خروجی:

- M: ماتریس نمای جانشینی به‌روز شده.

٤. کلاس DQN (Deep Q-Network)

کلاس DQN پیاده‌سازی **Deep Q-Network** است که از شبکه‌های عصبی برای یادگیری مقادیر Q استفاده می‌کند. این الگوریتم یک روش مدل‌دار نیست و از طریق به‌روزرسانی مقادیر Q از تجربیات گذشته به‌طور مداوم خود را بهبود می‌بخشد.

- **ویژگی‌ها**:
  - state_dim: تعداد ویژگی‌ها در فضای حالت.
  - action_dim: تعداد اقدامات ممکن.
  - gamma: عامل تخفیف که در محاسبه پاداش‌های آینده استفاده می‌شود.
  - lr: نرخ یادگیری برای به‌روزرسانی شبکه عصبی.
  - memory: دک از تجربیات (حالت، اقدام، پاداش، حالت بعدی، پایان).
  - Model: مدل شبکه عصبی برای پیش‌بینی مقادیر Q.

- **متدها**:

- build_model(): مدل شبکه عصبی را می‌سازد که برای پیش‌بینی مقادیر Q از آن استفاده می‌شود.
- remember(state, action, reward, next_state, done): تجربه‌ای را به حافظه اضافه می‌کند.
- act(state): با استفاده از سیاست epsilon-greedy یک اقدام انتخاب می‌کند. اگر با احتمال epsilon ، یک اقدام تصادفی (اکتشاف) انتخاب شود و در غیر این صورت، اقدامی با بالاترین مقدار Q انتخاب می‌شود.
- replay(batch_size=32): یک دسته از تجربیات از حافظه نمونه‌برداری می‌کند و مدل را به‌روزرسانی می‌کند.

ورودی‌ها:

- state: حالت جاری.
- action: اقدام انجام شده.
- Reward: پاداش دریافتی.
- next_state: حالت بعدی.

خروجی:

- model: شبکه عصبی به‌روز شده.

۵. تابع simulate_and_compare

این تابع اصلی‌ترین بخش کد است که در آن شبیه‌سازی انجام می‌شود و الگوریتم‌ها با یکدیگر مقایسه می‌شوند.

- **ورودی‌ها**:
  - state_dim: تعداد ویژگی‌ها در فضای حالت.
  - action_dim: تعداد اقدامات ممکن.

- episodes: تعداد اپیزودهایی که شبیه‌سازی اجرا می‌شود.
- steps: تعداد مراحل در هر اپیزود.

- **روش‌شناسی**:

۱. **مقداردهی اولیه**: سه الگوریتم MAK-TD، MAK-SR و DQN با پارامترهای مشخص ایجاد می‌شوند.

۲. **حلقه شبیه‌سازی:** برای هر اپیزود و هر مرحله زمانی، شبیه‌سازی انجام می‌شود. در هر مرحله، الگوریتم‌ها اقداماتی را انتخاب کرده، پاداش‌هایی دریافت کرده و مدل‌های داخلی خود را به‌روزرسانی می‌کنند.

۳. **محاسبه پاداش‌ها**: پاداش‌های تجمعی برای هر الگوریتم محاسبه می‌شود.

٤. **بازپخش برایDQN** : پس از هر اپیزود، DQN برای آموزش مدل خود از تجربیات گذشته بازپخش می‌کند.

٥. **نمایش نتایج**: نتایج مقایسه‌ای از پاداش‌های تجمعی برای هر الگوریتم در یک نمودار گرافیکی نمایش داده می‌شود.

نتیجه‌گیری

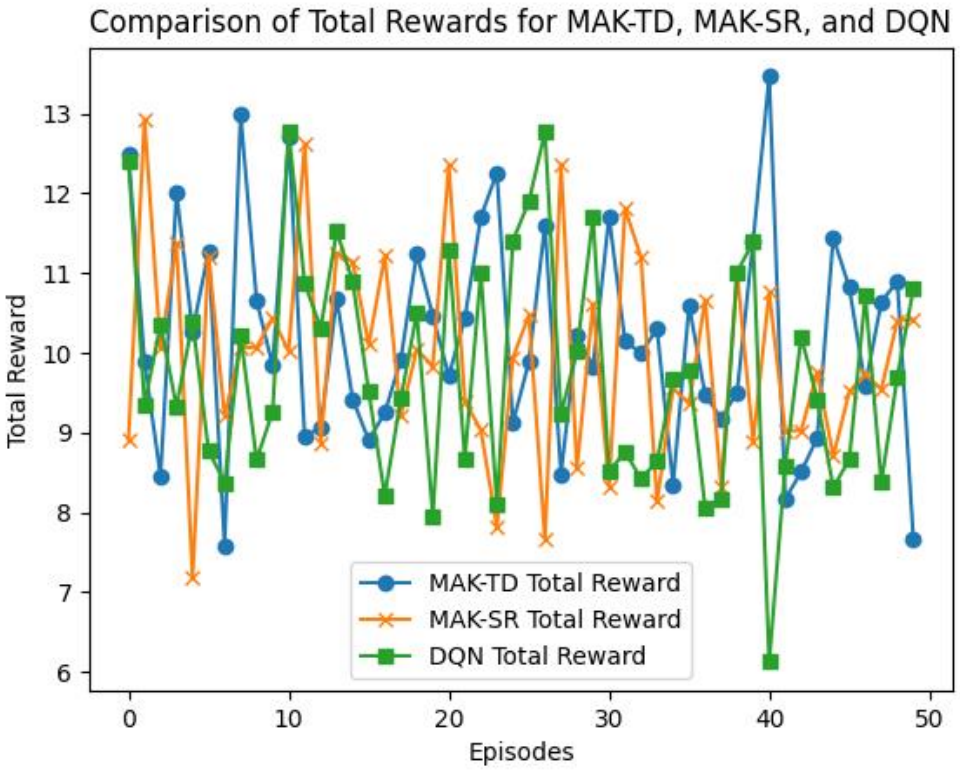
شبیه‌سازی انجام‌شده مقایسه‌ای از عملکرد سه الگوریتم یادگیری تقویتی **MAK-TD**، **MAK-SR** و **DQN** را در ۱۰۰ اپیزود به نمایش می‌گذارد. هر یک از این الگوریتم‌ها ویژگی‌های منحصر به‌فردی دارند:

۱. **MAK-TD** از فیلتر کالمن برای به‌روزرسانی وزن‌ها و یادگیری تفاوت زمانی برای بهبود پیش‌بینی‌ها استفاده می‌کند.

۲. **MAK-SR** از نمای جانشینی برای پیش‌بینی پاداش‌های آینده و مدل‌سازی تکامل حالات استفاده می‌کند.

۳. **DQN** از شبکه عصبی برای تقریب مقادیر Q استفاده می‌کند و قادر است به‌طور خودکار سیاست‌های بهینه را یاد بگیرد.

در نهایت نتایج شبیه سازی صورت گرفته نیز به صورت زیر است:

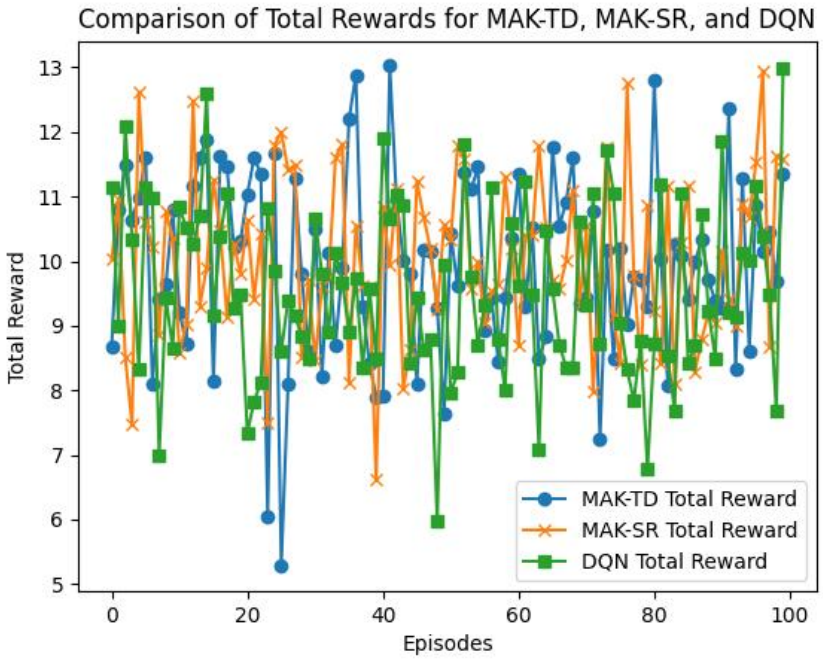


شکل ۳: کل پاداش دریافتی در طول ۵۰ اپیزود توسط الگوریتم‌های مختلف

به طور دقیق‌تر خواهیم داشت:

```
Final Total Rewards after 50 Episodes:
MAK-TD Total Reward: 510.10918848452957
MAK-SR Total Reward: 498.1273978229948
DQN Total Reward: 488.53212117234517
```

که نتایج فوق نشان دهنده عملکرد بهتر چارچوب‌های پیشنهادی نسبت به الگوریتم DQN است. این شبیه سازی در ۱۰۰ اپیزود تکرار گردید، که نتایج آن در شکل ۴ آورده شده است.



شکل ۴: کل پاداش دریافتی در طول ۱۰۰ اپیزود توسط الگوریتم‌های مختلف

پاداش‌های دریافتی در این ۱۰۰ اپیزود به صورت زیر است:

```
MAK-TD Total Reward: 996.1441199350996
MAK-SR Total Reward: 1003.7453232643612
DQN Total Reward: 955.8240584992651
```

نتایج فوق همچنان برتری چارچوب‌های پیشنهادی نسبت به DQN را نشان می‌دهد.

۱.۱۱ جمع‌بندی نهایی

در این مقاله، دو چارچوب جدید به نام‌های **MAK-TD** و **MAK-SR** برای یادگیری تقویتی چندعاملی (MARL) معرفی شدند. این چارچوب‌ها با بهره‌گیری از ویژگی‌های فیلتر کالمن تطبیقی، عدم قطعیت را مدل‌سازی کرده و کارایی یادگیری در محیط‌های چندعاملی پیچیده را بهبود می‌بخشند.

۱. دستاوردهای اصلی

- **MAK-TD**: این چارچوب به عنوان یک راهکار مبتنی بر تفاوت زمانی (TD) توسعه داده شد که از فیلتر کالمن برای تخمین تابع ارزش و کاهش حساسیت به نویز استفاده می‌کند.
- **MAK-SR**: این چارچوب با ترکیب یادگیری SR و مدل‌سازی عدم قطعیت، توانست عملکرد برتری در مقایسه با روش‌های پیشرفته دیگر ارائه دهد.
- هر دو چارچوب توانستند:
  - مشکل بیش‌برازش و ناکارآمدی نمونه‌ها را که معمولاً در روش‌های مبتنی بر شبکه‌های عصبی عمیق (DNN) وجود دارد، کاهش دهند.
  - در محیط‌های چندعاملی با تعاملات پیچیده (همکاری، رقابت، و ترکیب آن‌ها) به نتایج قابل توجهی دست یابند.

۲. نوآوری‌ها

- استفاده از **تخمین تطبیقی مدل (MMAE)** برای تنظیم خودکار پارامترهای کلیدی، که باعث کاهش حساسیت به پارامترها شد.
- به‌کارگیری **مکانیسم یادگیری فعال** برای یافتن تعادل میان اکتشاف و بهره‌برداری، که باعث افزایش پاداش تجمعی عوامل شد.
- **مدل‌سازی SR با فیلتر کالمن** که نیاز به زمان و حافظه کمتر را فراهم کرد و عملکرد پایدارتر و قابل اعتمادتری ارائه داد.

۳. محدودیت‌ها و پیشنهادها برای کارهای آینده

- این چارچوب‌ها در محیط‌های شبیه‌سازی‌شده آزمایش شده‌اند. برای ارزیابی بیشتر، می‌توان از کاربردهای دنیای واقعی مانند رباتیک یا وسایل نقلیه خودران استفاده کرد.
- بررسی عملکرد در محیط‌هایی با عوامل بیشتری و پیچیدگی بالاتر، می‌تواند راه را برای توسعه بیشتر این چارچوب‌ها هموار کند.
- بهبود روش‌های ترکیب داده‌های محلی و جهانی در آموزش سیاست‌ها می‌تواند تأثیر قابل توجهی در محیط‌های پیچیده‌تر داشته باشد.

نتیجه نهایی

چارچوب‌های پیشنهادی MAK-TD و MAK-SR راه‌حلی قدرتمند و کارآمد برای یادگیری تقویتی چندعاملی ارائه می‌دهند. با استفاده از این چارچوب‌ها، امکان یادگیری سریع‌تر و پایدارتر در محیط‌های چندعاملی فراهم می‌شود، که آن‌ها را به گزینه‌ای ایده‌آل برای مسائل واقعی و چالش‌برانگیز تبدیل می‌کند.