مقاله پژوهشی

برون‌سپاری محاسبات غیرمتمرکز مبتنی بر یادگیری تقویتی عمیق چندعامله در رایانش لبه همراه

آتوسا دقایقی و محسن نیک‌رأی

چکیده: پشتیبانی از برنامه‌های کاربردی حساس به تأخیر و نیازمند محاسبات سنگین برای دستگاه‌های همراه با ظرفیت باتری محدود و منابع محاسباتی کم به‌سختی امکان‌پذیر است. توسعه فناوری‌های رایانش لبه همراه و انتقال توان بی‌سیم به دستگاه‌های همراه امکان می‌دهند تا وظایف محاسباتی خود را به سرورهای لبه برون‌سپاری کنند و انرژی را برای افزایش طول عمر باتری خود برداشت کنند. با این حال برون‌سپاری محاسبات با چالش‌هایی مانند منابع محاسباتی محدود سرور لبه، کیفیت کانال ارتباطی موجود و زمان محدود برای برداشت انرژی مواجه است. ما در این مقاله مسئله مشترک برون‌سپاری محاسبات و تخصیص منابع غیرمتمرکز را در محیط پویای رایانش لبه همراه مطالعه می‌کنیم. برای این منظور یک طرح برون‌سپاری مبتنی بر یادگیری تقویتی عمیق چندعامله را پیشنهاد می‌دهیم که همکاری بین دستگاه‌های همراه را برای تنظیم استراتژی‌هایشان در نظر می‌گیرد. به طور خاص، ما یک نسخه بهبودیافته الگوریتم گرادیان سیاست قطعی عمیق چندعامله را با به‌کارگیری ویژگی‌های clipped double Q-learning، به‌روزرسانی با تأخیر سیاست، هموارسازی سیاست هدف و بازپخش تجربه اولویت‌بندی‌شده پیشنهاد می‌دهیم. نتایج شبیه‌سازی نشان می‌دهند طرح برون‌سپاری پیشنهادی، عملکرد همگرایی بهتری نسبت به سایر روش‌ها دارد و همچنین میانگین مصرف انرژی، میانگین تأخیر پردازش و نرخ شکست وظیفه را کاهش می‌دهد.

کلیدواژه: **برون‌سپاری محاسبات، تخصیص منابع، رایانش لبه همراه، یادگیری تقویتی عمیق چندعامله، برداشت انرژی.**

۱- مقدمه

نیازمندی‌های کاربران در مورد نرخ داده و کیفیت خدمات (QoS)[1] به طور نمایی در حال افزایش است [۱]. علاوه بر این، پیشرفت فناوری در گوشی‌های هوشمند، لپ‌تاپ‌ها و تبلت‌ها امکان ظهور برنامه‌های کاربردی جدید را فراهم می‌کند. اگرچه دستگاه‌های همراه[2] جدید از لحاظ قابلیت‌های پردازشی روزبه‌روز قدرتمندتر می‌شوند، حتی این دستگاه‌ها نیز ممکن است نتوانند در مدت زمان کوتاهی برنامه‌هایی را که به پردازش عظیم نیاز دارند اداره کنند [۲]. علاوه بر این، مصرف بالای باتری همچنان مانع قابل توجهی است که کاربران را برای لذت‌بردن کامل از برنامه‌های کاربردی در دستگاه‌های خود محدود می‌کند [۳]. برون‌سپاری محاسبات[3] بخشی از وظایف برنامه‌های کاربردی از دستگاه‌های همراه به مراکز داده ابری، راه‌حل امیدوارکننده‌ای برای این مسئله است [۴]. با این حال از آنجا که مراکز داده ابری معمولاً از نظر فیزیکی در مناطق دور قرار دارند، پردازش این وظایف در ابر با تأخیر زیادی همراه خواهد بود که برآورده‌کردن نیازمندی‌های برنامه‌های کاربردی حساس به تأخیر را به چالش می‌کشد. بنابراین استفاده از مدل رایانش لبه همراه (MEC)[4] که دستگاه‌های همراه با ظرفیت باتری و منابع محاسباتی محدود را قادر می‌سازد تا برنامه‌های کاربردی نیازمند محاسبات سنگین و حساس به تأخیر را در لبه شبکه‌ها اجرا کنند، ضروری است [۵] و [۶].

اگرچه برون‌سپاری محاسبات، مصرف انرژی دستگاه‌های همراه را تا حدی کاهش می‌دهد، دستگاه‌های همراه با ظرفیت باتری محدود برای پشتیبانی از عملکرد برنامه‌های کاربردی مختلف مناسب نیستند که موضوع مهمی برای توسعه سیستم MEC است. این امر به این دلیل است که نرخ داده‌های بالا، ارتباطات با تأخیر بسیار کم و توانایی‌های پردازش داده قوی برای شبکه‌های ۵G مورد نیاز است که منجر به مصرف انرژی بالای دستگاه‌های همراه می‌شود. با این حال، افزایش ظرفیت باتری دستگاه‌های همراه تا بی‌نهایت عملی نیست. علاوه بر این در مناطق دورافتاده و شرایط اضطراری، شارژ مجدد دستگاه‌های همراه با برق شبکه غیرممکن است [۷]. با خالی‌شدن باتری، دستگاه‌های همراه قادر به انجام عملیات خود شامل پردازش وظایف و انتقال آنها به سرور MEC نخواهند بود. چالش فوق را می‌توان با توسعه اخیر فناوری‌های انتقال توان بی‌سیم (WPT)[5] کاست. WPT، شارژ بی‌سیم دستگاه‌های همراه مجهز به ماژول‌های برداشت انرژی (EH)[6] را با استفاده از نقاط دسترسی (AP)[7] بی‌سیم که سیگنال‌های فرکانس رادیویی (RF)[8] را ارسال می‌کنند، تحقق می‌بخشد؛ بنابراین باعث افزایش عمر باتری و خودپایداری[9] دستگاه‌های همراه می‌شود و دستیابی به محاسبات سبز را ممکن می‌سازد [۶] تا [۸]. با ترکیب سیستم MEC با دستگاه‌های مجهز به EH می‌توان از مزایای دو فناوری فوق برخوردار شد؛ یعنی افزایش قابلیت‌های محاسباتی دستگاه‌های همراه و در عین حال کاهش کمبود انرژی. علی‌رغم مزایای بالقوه ادغام EH در MEC، رقابت برای منابع

این مقاله در تاریخ ۱۲ مهر ماه ۱۴۰۲ دریافت و در تاریخ ۲۱ بهمن ماه ۱۴۰۲ بازنگری شد.

آتوسا دقایقی، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه قم، قم، ایران، (email: atousa.daghayeghi@stu.qom.ac.ir).

محسن نیک‌رأی (نویسنده مسئول)، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه قم، قم، ایران، (email: m.nickray@qom.ac.ir).

1. Quality of Service
2. Mobile Devices
3. Computation Offloading
4. Mobile Edge Computing
5. Wireless Power Transfer
6. Energy Harvesting
7. Access Point
8. Radio Frequency
9. Self-Sustainability

محاسباتی محدود سرور MEC در بین دستگاه‌های همراه، منابع ارتباطی محدود، محدودبودن انرژی برداشت‌شده در یک زمان محدود، فرایند برون‌سپاری محاسبات را با چالش مواجه می‌کنند. علاوه بر این، اگرچه برون‌سپاری وظایف به سرور MEC، بار محاسباتی و مصرف انرژی را برای دستگاه‌های همراه کاهش می‌دهد، با این حال از آنجا که اجرای وظیفه به‌صورت محلی انجام نمی‌شود منجر به تأخیر انتقال بیشتر هنگام برون‌سپاری داده‌های وظیفه می‌شود. این عوامل متضاد باید به‌دقت مورد بررسی قرار گیرند تا به راه‌حلی مناسب دست یابیم. از طرفی تخصیص کارآمد منابع برای پشتیبانی از برون‌سپاری محاسبات در سیستم‌های MEC ضروری است. بدون هماهنگی کارآمد تخصیص منابع، سیستم ممکن است به تأخیر و مصرف انرژی بالا منجر شود که به نوبه خود بر عملکرد کلی برون‌سپاری تأثیر می‌گذارد. بنابراین برای به حداکثر رساندن کارایی سیستم، مطالعه مسئله بهینه‌سازی مشترک برون‌سپاری محاسبات و تخصیص منابع برای بهره‌گیری از مزایای کامل MEC با دستگاه‌های همراه مجهز به قابلیت EH بسیار مهم است؛ زیرا این دو موضوع ارتباط نزدیکی با هم دارند.

به طور کلی، مسئله برون‌سپاری محاسبات را می‌توان در دو روش متمرکز[1] و غیرمتمرکز[2] انجام داد [۹]. در تکنیک متمرکز، یک کنترل‌کننده مرکزی اطلاعات محیط ازجمله قابلیت‌های محاسباتی محلی، اطلاعات وضعیت کانال (CSI)[3] و نیازهای محاسباتی را از دستگاه‌های همراه جمع‌آوری می‌کند و تصمیمات برون‌سپاری را اتخاذ و دستگاه‌های همراه را در مورد تصمیمات آگاه می‌نماید [۶]. بنابراین رویکردهای برون‌سپاری متمرکز زمانی که برای سیستم‌های MEC در مقیاس بزرگ اعمال می‌شوند، به حجم عظیمی از اطلاعات و محاسبات نیاز دارند که به‌طور اجتناب‌ناپذیر تأخیر اجرایی قابل توجهی را متحمل می‌شود و همچنین هزینه ارتباطات را افزایش می‌دهد [۱۰]. چالش دیگر در طرح برون‌سپاری متمرکز، نقطه واحد شکست (SPoF)[4] است که در صورت بروز مشکل برای سرور متمرکز که وظیفه تصمیم‌گیری برون‌سپاری را به عهده دارد، عملکرد کل شبکه مختل می‌گردد [۱۱]. بنابراین روش‌های متمرکز برای به‌کارگیری در شبکه دنیای واقعی مناسب نیستند. در مقابل، رویکردهای غیرمتمرکز از سربار ارتباطات که از هماهنگ‌سازی اطلاعات همه دستگاه‌ها تحمیل می‌شود، جلوگیری می‌کند. در سیستم‌های MEC، سربار ارتباطات برای جمع‌آوری اطلاعات کاربران معمولاً بزرگ‌تر از سربار محاسبه است و نادیده‌گرفتن آن برای سیستم‌های واقعی عملی نیست.

هنگام طراحی طرح برون‌سپاری غیرمتمرکز در یک سیستم MEC، باید مسائل مربوط به همکاری و رقابت بین دستگاه‌های همراه برای منابع سرور MEC و همچنین یک مکانیزم عملی که ویژگی‌های تحرک، تصادفی‌بودن و ناهمگونی را در نظر می‌گیرد، مورد توجه قرار گیرد [۱۲]. یادگیری تقویتی عمیق چندعامله (MADRL)[5] می‌تواند به عنوان یک رویکرد مؤثر برای یادگیری یک سیاست برون‌سپاری غیرمتمرکز برای دستگاه همراه بدون دانش قبلی از محیط استفاده شود و مسائل مربوط به ویژگی‌های تصادفی و نامطمئن در سیستم‌های دینامیک را در نظر بگیرد. یک رویکرد برای پرداختن به محیط MADRL رویکرد یادگیرنده مستقل (IL)[6] است که در آن عامل‌ها به طور مستقل سیاست‌های خود را تنها با دسترسی به اطلاعات محلی خود برای حداکثرسازی بازده[7] خود بهینه می‌کنند [۱۳]. با این حال، رقابت یا همکاری بین عامل‌ها را نمی‌توان با این رویکرد مدل کرد. علاوه بر این از آنجا که عامل‌ها به طور همزمان سیاست‌های خود را بهبود می‌دهند، اقدامات یک عامل بر پاداش‌های سایر عامل‌ها و همچنین انتقال حالت[8] محیط تأثیر می‌گذارد. در نتیجه محیط از دیدگاه هر عامل به دلیل نقض خاصیت مارکوف غیرایستا[9] می‌شود و همگرایی الگوریتم را نمی‌توان تضمین کرد [۱۴] تا [۱۶]. رویکرد دیگری که به مسائل فوق می‌پردازد، استفاده از چارچوب آموزش متمرکز و اجرای غیرمتمرکز (CTDE)[10] است که در آن در طول آموزش، یک کنترل‌کننده مرکزی برای جمع‌آوری اطلاعات اضافی درباره عامل‌ها ازجمله مشاهدات و اقدامات مشترک در نظر گرفته می‌شود. با این حال، سیاست‌های آموخته‌شده غیرمتمرکز هستند و اقدامات تنها با استفاده از مشاهدات محلی عامل تصمیم‌گیری می‌شوند [۱۳]. بنابراین در چارچوب CTDE، عامل قادر به یادگیری یک سیاست غیرمتمرکز از طریق بهینه‌سازی جهانی است [۱۴]. از شناخته‌شده‌ترین الگوریتم‌های مبتنی بر CTDE، گرادیان سیاست قطعی عمیق چندعامله (MADDPG)[11] است [۱۷]. این الگوریتم، توسعه الگوریتم گرادیان سیاست قطعی عمیق (DDPG)[12] [۱۸] در محیط‌های چندعامله است که عملکرد خوبی برای مسائل با فضای اقدام پیوسته ارائه می‌دهد. MADDPG [۱۷] بر اساس روش بازیگر- منتقد[13] طراحی شده که در آن هر عامل شامل یک شبکه بازیگر غیرمتمرکز و یک شبکه منتقد متمرکز است. در MADDPG، شبکه بازیگر سیاست خود را در جهتی به‌روز می‌کند که ارزش را بهبود می‌بخشد و یاد می‌گیرد که اقدامی را با بالاترین ارزش تخمینی بر اساس شبکه منتقد انتخاب کند؛ بنابراین محدودیت الگوریتم‌های بازیگر- منتقد، بهره‌برداری بازیگر از خطای تخمین ارزش منتقد است. طبق [۱۹] بایاس تخمین بیش از حد[14] می‌تواند در روش‌های بازیگر- منتقد به دلیل خطای تقریب تابع رخ دهد و بنابراین تخمین دقیق ارزش توسط منتقد در روش‌های بازیگر- منتقد به بازیگر امکان یادگیری سیاست بهتری را می‌دهد. علاوه بر این در MADDPG، نمونه‌های تجربه به طور یکنواخت از بافر بازپخش[15] برای آموزش انتخاب می‌شوند [۱۷]. با این حال، تجربیات مختلف ممکن است اهمیت متفاوتی داشته باشند و عامل از برخی تجربیات بیشتر از بقیه یاد بگیرد. بنابراین اولویت‌بندی تجربیات با توجه به اهمیت آنها باعث می‌شود تا بازپخش تجربه مؤثرتر باشد و به فرایند یادگیری الگوریتم سرعت می‌بخشد.

با انگیزه مسائل ذکرشده در بالا، این مقاله مسئله مشترک برون‌سپاری محاسبات و تخصیص منابع غیرمتمرکز را در محیط پویای MEC با هدف به حداقل رساندن هزینه محاسباتی بلندمدت از لحاظ مصرف انرژی و تأخیر پردازش وظیفه مورد مطالعه قرار می‌دهد. برای هماهنگ‌کردن کاربرانی که برای دسترسی به منابع محدود سرور MEC رقابت می‌کنند،

1. Centralized
2. Decentralized
3. Channel State Information
4. Single Point of Failure
5. Multi-Agent Deep Reinforcement Learning
6. Independent Learner
7. Return
8. State Transition
9. Non-Stationary
10. Centralized Training Decentralized Execution
11. Multi-Agent Deep Deterministic Policy Gradient
12. Deep Deterministic Policy Gradient
13. Actor-Critic
14. Overestimation Bias
15. Replay Buffer

چارچوب مبتنی بر CTDE را در نظر می‌گیریم و یک الگوریتم MADDPG بهبودیافته را پیشنهاد می‌کنیم. به طور خاص، نوآوری‌های اصلی ما به شرح زیر خلاصه می‌شود:

- ما مسئله مشترک برون‌سپاری محاسبات بخشی[1] و تخصیص منابع را در شبکه MEC با دستگاه‌های همراه مجهز به قابلیت EH به‌صورت یک مسئله بهینه‌سازی برنامه‌ریزی غیرخطی (NLP)[2] فرموله کرده‌ایم. هدف مسئله پیشنهادی حداقل‌سازی مصرف انرژی و تأخیر پردازش وظیفه تحت محدودیت‌های حداکثر تأخیر قابل تحمل وظیفه و ظرفیت باتری محدود دستگاه‌های همراه است.
- برای حل مسئله و پرداختن به چالش محیط پویای MEC، ما MADRL را به کار می‌گیریم و یک طرح برون‌سپاری مبتنی بر چارچوب CTDE پیشنهاد می‌کنیم. بنابراین هر دستگاه همراه به عنوان عاملی در نظر گرفته می‌شود که در حین اجرا به طور مستقل با توجه به مشاهدات محلی خود از محیط در مورد اقدام تصمیم می‌گیرد؛ اما در طول آموزش، اطلاعات اضافی شامل مشاهدات و سیاست‌های خود را با سایر دستگاه‌ها به اشتراک می‌گذارد.
- به طور خاص، ما الگوریتم MADDPG را بهبود بخشیده‌ایم. برای بهبود عملکرد و کاهش مسئله بایاس تخمین بیش از حد در MADDPG، ما از clipped double Q-learning، به‌روزرسانی با تأخیر سیاست و هموارسازی سیاست هدف[3] استفاده کرده‌ایم. علاوه بر این از بازپخش تجربه اولویت‌بندی‌شده (PER)[4] برای بهبود کارایی داده‌ها[5] و تسریع فرایند آموزش استفاده کرده‌ایم.
- نتایج شبیه‌سازی عملکرد برتر و سرعت همگرایی بالاتر الگوریتم پیشنهادی را در مقایسه با سایر الگوریتم‌ها نشان می‌دهد. تحلیل عددی نشان می‌دهد که الگوریتم پیشنهادی ما هزینه محاسباتی سیستم را از نظر تأخیر پردازش وظیفه و مصرف انرژی کاهش می‌دهد و نیز منجر به کاهش نرخ شکست وظیفه[6] می‌شود.

سازماندهی مقاله به این صورت است که در بخش ۲ پژوهش‌های مرتبط بررسی می‌شوند. بخش ۳ به تشریح مدل سیستم و فرموله‌سازی مسئله پیشنهادی و بخش ۴ به بیان جزئیات الگوریتم پیشنهادی اختصاص یافته است. در بخش ۵ به ارزیابی عملکرد رویکرد پیشنهادی می‌پردازیم و در بخش ۶ نتیجه‌گیری و کارهای آتی را خواهیم داشت.

۲- پژوهش‌های مرتبط

در این بخش، کارهای انجام‌شده در حوزه برون‌سپاری محاسبات را به دو دسته طرح‌های برون‌سپاری مبتنی بر روش‌های بهینه‌سازی سنتی و طرح‌های برون‌سپاری مبتنی بر روش‌های یادگیری ماشین (ML)[7] دسته‌بندی کرده‌ایم و به بررسی این رویکردها پرداخته‌ایم.

۲-۱ برون‌سپاری محاسبات مبتنی بر روش‌های سنتی

الگوریتم‌های فرااکتشافی[8] و رویکردهای مبتنی بر بهینه‌سازی محدب زمانی که محیط شبکه ایستا است، به عنوان رویکرد مناسبی شناخته می‌شوند؛ بنابراین برخی پژوهشگران از این الگوریتم‌ها برای حل مسئله برون‌سپاری استفاده کرده‌اند. به عنوان مثال پژوهشگران در [۲۰] یک طرح برون‌سپاری وظیفه در شبکه IoT توانمند به مه[9] با در نظر گرفتن محدودیت‌های باتری باقیمانده دستگاه IoT و مهلت زمانی وظایف پیشنهاد داده‌اند و مسئله را با هدف حداقل‌سازی تأخیر تکمیل وظیفه و مصرف انرژی دستگاه‌های IoT فرموله کرده‌اند و برای حل آن، یک الگوریتم هیبریدی با ترکیب الگوریتم ژنتیک (GA)[10] و بهینه‌سازی ازدحام ذرات (PSO)[11] طراحی کرده‌اند. در [۲۱]، یک طرح برون‌سپاری بخشی برای به حداقل رساندن انرژی کل مصرف‌شده توسط دستگاه‌های همراه هوشمند و سرورهای لبه با بهینه‌سازی مشترک نسبت برون‌سپاری وظیفه، پهنای باند تخصیص‌یافته، سرعت CPU و توان انتقال هر دستگاه همراه پیشنهاد شده و یک الگوریتم هیبریدی با ترکیب الگوریتم‌های PSO، GA و شبیه‌سازی تبرید[12] طراحی شده است. در [۲۲] یک شبکه MEC با کاربران مختلف با هدف حداقل‌سازی مصرف انرژی بررسی شده که حالت دسترسی چندگانه تقسیم زمانی (TDMA)[13] را اتخاذ می‌کند. با توجه به غیرمحدب‌بودن[14] مسئله فرموله‌شده، نویسندگان بر اساس چارچوب تقریب محدب متوالی (SCA)[15] یک رویکرد تکراری برای حل توسعه داده‌اند. با این حال از معایب الگوریتم‌های فرااکتشافی می‌توان به افزایش زمان اجرا با افزایش تعداد وظایف در برنامه و گیرافتادن در بهینه محلی اشاره کرد [۱۲]. از سوی دیگر از آنجا که الگوریتم‌های فرااکتشافی و رویکردهای مبتنی بر بهینه‌سازی محدب برای دستیابی به سیاست بهینه یا نزدیک به بهینه نیاز به تعداد زیادی تکرار دارند، برای تصمیم‌گیری برون‌سپاری بلادرنگ در محیط متغیر MEC غیرعملی هستند.

علاوه بر این، برخی کارها مسئله برون‌سپاری مبتنی بر فرایند تصمیم‌گیری مارکوف (MDP)[16] را مطالعه کرده‌اند. پژوهشگران در [۲۳]، استراتژی انتخاب گره برون‌سپاری بهینه را به‌صورت یک مدل MDP با در نظر گرفتن پهنای باند شبکه در دسترس سرورهای لبه و موقعیت دستگاه‌های همراه در سیستم MEC فرموله می‌کنند و الگوریتم تکرار ارزش (VIA)[17] را برای حل MDP و دستیابی به زمان برون‌سپاری بهینه به کار می‌گیرند. با این حال برای حل MDP، مدل سیستم مانند احتمال انتقال[18] باید کاملاً شناخته‌شده باشد. در حالی که در یک شبکه واقعی، دستیابی به این اطلاعات بسیار دشوار است و تخمین احتمال انتقال به داده‌های انبوهی نیاز دارد که کاربرد آن در MEC را چالش‌برانگیز می‌کند. همچنین در مسائل MDP، پیچیدگی محاسباتی به‌صورت نمایی با تعداد حالت‌ها افزایش می‌یابد که منجر به مسئله نفرین ابعاد[19] می‌شود [۲۴].

به منظور پرداختن به مسئله برون‌سپاری محاسبات چندکاربره در یک محیط ایستا، برخی از پژوهشگران این مسئله را از طریق نظریه بازی مدل‌سازی کرده‌اند. در [۲۵] نویسندگان مسئله برون‌سپاری بخشی را

1. Partial
2. Non-Linear Programming
3. Target Policy Smoothing
4. Prioritized Experience Replay
5. Data Efficiency
6. Task Failure Rate
7. Machine Learning
8. Meta-Heuristics
9. Fog-Enabled IoT
10. Genetic Algorithm
11. Particle Swarm Optimization
12. Simulated Annealing
13. Time Division Multiplexing Access
14. Non-Convex
15. Successive Convex Approximation
16. Markov Decision Process
17. Value Iteration Algorithm
18. Transition Probability
19. Curse of Dimensionality

به‌صورت یک مسئله تعادل نش [1](NE) تعمیم‌یافته با هدف کاهش تأخیر فرموله کرده‌اند. وجود NE با استفاده از نظریه نقطه ثابت [2] اثبات شده و یک الگوریتم برون‌سپاری وظیفه توزیعی که بر اساس اطلاعات محلی و روش Gauss-Seidel-type است، توسعه یافته است. پژوهشگران در [۲۶]، مسئله برون‌سپاری محاسبات توزیعی در یک سیستم MEC سبز با دستگاه‌های مجهز به EH را به‌صورت یک بازی مدل کرده‌اند که تصمیمات برون‌سپاری و برداشت انرژی و همچنین نرخ محاسبه محلی و توان انتقال را برای حداقل‌کردن تأخیر و طولانی‌کردن عمر باتری دستگاه‌های EH با به‌کارگیری الگوریتم‌های برداشت انرژی مبتنی بر Lyapunov-drift و الگوریتم بهترین پاسخ ارائه می‌دهد. مسئله برون‌سپاری بخشی در شبکه MEC با هدف حداقل‌سازی زمان پاسخ میانگین به‌صورت یک بازی برون‌سپاری غیرمشارکتی برای نشان‌دادن رقابت بین کاربران برای دستیابی به منابع سرورها با در نظر گرفتن تصادفی‌بودن تولید وظیفه، ورود انرژی برداشت‌شده، حالت کانال بی‌سیم و تداخل بین دستگاه‌های همراه در [۲۷] فرموله شده است. برای حداکثرسازی مزیت کل سیستم از لحاظ درآمد سرور MEC، [۲۸] طرح برون‌سپاری سمت سرور را به‌صورت یک بازی غیرمشارکتی فرموله کرده است. برای همگراشدن به NE، پژوهشگران در این مقاله یک الگوریتم به‌روزرسانی پاسخ بهتر اکتشافی حریصانه [3](GH-BRU) را پیشنهاد کرده‌اند. با این حال در یک بازی برون‌سپاری، افزایش تعداد کاربران منجر به افزایش نمایی در پیچیدگی محاسباتی می‌شود؛ بنابراین دستیابی به NE را نمی‌توان تضمین کرد.

۲-۲ برون‌سپاری محاسبات مبتنی بر روش‌های ML

روش‌های مبتنی بر یادگیری عمیق به دلیل عملکرد مناسب پیش‌بینی و استدلال، در طرح‌های برون‌سپاری محاسبات و تخصیص منابع استفاده شده‌اند. در [۲۹] نویسندگان یک سناریوی برون‌سپاری را با در نظر گرفتن ناهمگنی سرورهای لبه و ابر مرکزی در انتخاب مقصد برون‌سپاری مطالعه می‌کنند و یک طرح برون‌سپاری مبتنی بر یادگیری عمیق توزیع‌شده [4](DDTO) را برای به حداکثر رساندن مزیت سیستم با حداقل‌سازی مصرف انرژی و تأخیر پیشنهاد می‌کنند. با این حال، آموزش مدل‌های مبتنی بر یادگیری عمیق به مجموعه داده‌های برچسب‌گذاری‌شده عظیمی نیاز دارد. تولید و برچسب‌گذاری داده‌ها در سیستم MEC بسیار چالش‌برانگیز است. علاوه بر این، مدل‌های آموزش داده‌شده ممکن است در صورت تغییر در مجموعه داده‌ها نیاز به آموزش مجدد داشته باشند.

در مقابل در یادگیری تقویتی عمیق [5](DRL) که شبکه‌های عصبی را در یادگیری تقویتی ادغام می‌کند، به داده‌های برچسب‌گذاری‌شده برای آموزش نیاز نیست و سیاست بهینه توسط عامل از طریق تعامل با محیط آموخته می‌شود. بنابراین برخی پژوهشگران روش‌های DRL مبتنی بر ارزش [6] را برای حل مسائل برون‌سپاری با فضای اقدام پیوسته به کار گرفته‌اند. در [۳۰]، برون‌سپاری محاسبات در سیستم دینامیک MEC با وظایف محاسباتی با نیازمندی‌های مختلف با هدف حداکثرکردن تعداد وظایف تکمیل‌شده قبل از مهلت زمانی‌شان و حداقل‌سازی مصرف انرژی بررسی شده است. برای بهینه‌سازی مسئله و کاهش نوسانات مدل هنگام آموزش، یک الگوریتم Q-Network عمیق [7](DQN) با به‌کارگیری روش روش تابع جانشین بریده‌شده [8] پیشنهاد گردیده است. نویسندگان در [۳۱] مسئله برون‌سپاری بخشی در شبکه MEC با کاربران و APهای مختلف با را در نظر گرفتن مشخصات وظیفه و قابلیت‌های محاسباتی ناهمگن APها مطالعه کرده‌اند و یک الگوریتم DQN را با هدف بهینه‌سازی مصرف انرژی و تأخیر توسعه داده‌اند. در [۳۲]، مسئله برون‌سپاری بخشی و تخصیص منبع محاسباتی در سیستم MEC متشکل از دستگاه‌های همراه با قابلیت EH و یک مدل تحرک یکنواخت با هدف حداقل‌سازی هزینه تأخیر و مصرف انرژی مدل شده و الگوریتم DQN برای حل پیشنهاد گردیده است. نویسندگان در [۳۳] یک شبکه MEC با به‌کارگیری فناوری دسترسی چندگانه غیرمتعامد [9](NOMA) و قابلیت WPT را مطالعه کرده‌اند و یک چارچوب بهبود نمونه آنلاین مبتنی بر DRL [10](DROS) برای دستیابی به حداکثر نرخ محاسباتی پیشنهاد داده‌اند. با این حال روش‌های مبتنی بر ارزش برای مسائل با فضای اقدام پیوسته مناسب نیستند. یک راه‌حل برای تطبیق روش‌های مبتنی بر ارزش با فضای اقدام پیوسته، گسسته‌سازی فضای اقدام است. با این حال گسسته‌سازی دقیق منجر به نفرین ابعاد می‌شود و اکتشاف این فضا برای یافتن سیاست بهینه بسیار چالش‌برانگیز می‌گردد؛ در حالی که گسسته‌سازی ساده منجر به از دست رفتن اطلاعات ساختاری فضای اقدام می‌شود.

برای مقابله با مسائل مربوط به روش‌های DRL مبتنی بر ارزش در فضای اقدام پیوسته، برخی دیگر از پژوهشگران از روش‌های DRL مبتنی بر بازیگر- منتقد برای مسئله برون‌سپاری متمرکز استفاده کرده‌اند. در [۳۴] برای بهبود تجربه کاربران، یک مدل بهینه‌سازی در محیط IoT توانمند به لبه با در نظر گرفتن تأخیر سرویس، مصرف انرژی و نرخ موفقیت وظیفه پیشنهاد شده و الگوریتمی با نام D^3PG [11] که روش‌های Double Q-learning و Dueling network را در الگوریتم DDPG ترکیب می‌کند، پیشنهاد شده است. نویسندگان در [۳۵] بر کاهش مجموع وزن‌دار مصرف انرژی و تأخیر در سیستم MEC دینامیک با شرایط کانال متغیر و پروتکل‌های دسترسی چندگانه هیبریدی دسترسی چندگانه متعامد [12](OMA) و NOMA تمرکز کرده‌اند و با به‌کارگیری مزایای روش‌های DQN و بازیگر- منتقد، الگوریتمی با نام ACDQN برای بهینه‌سازی برون‌سپاری بخشی و تخصیص کانال تحت محدودیت حداکثر توان را توسعه داده‌اند. در [۳۶]، مسئله برون‌سپاری و تخصیص منبع در MEC با هدف کاهش تأخیر و مصرف انرژی همه وظایف فرموله شده است. مسئله برای کاهش پیچیدگی بهینه‌سازی به دو زیرمسئله تجزیه و با به‌کارگیری الگوریتم گرادیان سیاست قطعی عمیق دوقلو با تأخیر [13](TD3) و روش جهت متناوب ضرب‌کننده‌ها [14](ADMM) حل شده است. در [۳۷] یک عامل DDPG بهبودیافته با نام GCN-DDPG پیشنهاد گردیده که از توانایی شبکه‌های کانولوشنی گراف برای بهینه‌سازی تصمیمات نسبت

1. Nash Equilibrium
2. Fixed-Point
3. Greedy Heuristic Better Response Update
4. Distributed Deep Learning-Based Task Offloading
5. Deep Reinfircement Learning
6. Value-Based
7. Deep Q-Network
8. Clipped Surrogate Function
9. Non-Orthogonal Multiple Access
10. Deep Reinforcement Learning-Based Online Sample-Improving
11. Double Dueling Deterministic Policy Gradient
12. Orthogonal Multiple Access
13. Twin-Delayed Deep Deterministic Policy Gradient
14. The Alternating Direction Method of Multipliers

برون‌سپاری، تعیین ظرفیت محاسباتی محلی و توان انتقال در شبکه MEC با دستگاه‌های EH استفاده می‌کند. برای دستیابی به QoE بهتر با در نظر گرفتن مصرف انرژی تحت محدودیت تأخیر سرویس، نویسندگان در [۳۸] یک طرح برون‌سپاری مبتنی بر الگوریتم DDPG پیشرفته را با نام PS-DDPG با به‌کارگیری مکانیسم‌های PER و میانگین‌گیری وزن تصادفی (SWA)[1] در اینترنت وسایل نقلیه (IoV)[2] توانمند به لبه ارائه داده‌اند. اگرچه این روش‌ها می‌توانند محیط در حال تغییر و فضاهای اقدام پیوسته را مدیریت کنند، این کارها یک طرح برون‌سپاری متمرکز را در نظر می‌گیرند که در آن یک سرور متمرکز برای جمع‌آوری تمام اطلاعات ضروری و تصمیم‌گیری برای همه دستگاه‌های همراه مورد نیاز است. بنابراین این مدل‌سازی مقیاس‌پذیری کافی ندارد.

از آنجا که برون‌سپاری متمرکز با افزایش مقیاس شبکه چالش‌برانگیز می‌شود، نویسندگان طرح‌های برون‌سپاری غیرمتمرکز را پیشنهاد کرده‌اند. یک طرح برون‌سپاری بخشی غیرمتمرکز مبتنی بر الگوریتم DDPG برای حداقل‌کردن تأخیر بافر و مصرف توان در محیط MEC در [۳۹] ارائه شده که ورود تصادفی وظایف و شرایط کانال ارتباطات متغیر را در نظر می‌گیرد. پژوهشگران در [۴۰] الگوریتمی بر اساس DDPG با نام گرادیان سیاست قطعی توجه زمانی (TADPG)[3] برای بهینه‌سازی برون‌سپاری و تخصیص منبع با هدف حداقل‌سازی میانگین مصرف انرژی و زمان تکمیل وظیفه توسعه داده‌اند که از دو ویژگی شبکه استخراج ویژگی زمانی (TFEN)[4] و PER مبتنی بر رتبه (rPER)[5] برای همگرایی سریع‌تر و ثبات بهتر الگوریتم استفاده می‌کند. با این حال، این روش‌ها از چارچوب IL برای پیاده‌سازی طرح برون‌سپاری غیرمتمرکز استفاده می‌کنند که در آن همه عامل‌ها به طور همزمان سیاست خود را یاد می‌گیرند؛ بنابراین محیط از نظر هر عامل غیرثابت است و با افزایش تعداد عامل‌ها، عملکرد سیستم کاهش می‌یابد و همگرایی آموزش چالش‌برانگیز می‌شود.

نویسندگان در [۴۱] تا [۴۵]، طرح‌های برون‌سپاری مبتنی بر چارچوب CTDE را پیشنهاد کرده‌اند. در [۴۱]، مسئله کنترل توان و تقسیم وظیفه برای حداکثرسازی مزیت بلندمدت سیستم شامل مصرف انرژی و تأخیر اجرا در پارادایم رایانش مه به‌صورت یک بازی تصادفی چندعامله مدل شده و یک الگوریتم برون‌سپاری وظیفه مبتنی بر MADDPG ارائه گردیده است. با تمرکز بر چارچوب MADRL مشارکتی، پژوهشگران در [۴۲] یک طرح برون‌سپاری بخشی غیرمتمرکز در محیط MEC مبتنی بر NOMA برای هماهنگ‌کردن تداخل بین کاربران مختلف را در نظر گرفته‌اند و برای دستیابی به استراتژی تعیین توان اجرای محلی و توان برون‌سپاری با هدف حداقل‌کردن مصرف توان و تأخیر بافر، الگوریتم PSMADDPG را پیشنهاد داده‌اند که از یک تکنیک به اشتراک‌گذاری پارامتر[6] برای کاهش پیچیدگی آموزش استفاده می‌کند. در [۴۳] بر روی طراحی مشترک برون‌سپاری و هماهنگی تداخل با هدف حداقل‌سازی مصرف انرژی و در عین حال برآورده‌کردن نیازمندی‌های تأخیر تمرکز گردیده و از الگوریتم MADDPG برای حل مسئله استفاده شده است. در [۴۴]، یک طرح برون‌سپاری وظیفه با هدف حداقل‌سازی هزینه کل سیستم از لحاظ تأخیر اجرایی و مصرف انرژی مبتنی بر الگوریتم TD3 چندعامله در محیط MEC مبتنی بر پهپاد (UAV)[7] پیشنهاد شده است. در [۴۵] یک طرح برون‌سپاری و تخصیص منبع با هدف حداقل‌سازی تأخیر کل وظایف دستگاه‌های همراه مبتنی بر MADDPG در MEC توسعه داده شده است. اگرچه این رویکردها می‌توانند مسئله عدم ایستایی در طرح‌های برون‌سپاری مبتنی بر IL را حل کنند، در الگوریتم‌های مبتنی بر بازیگر- منتقد ازجمله MADDPG، عملکرد شبکه بازیگر به‌شدت به ارزش تخمین‌زده‌شده توسط شبکه منتقد بستگی دارد. از آنجایی که MADDPG از مسئله تخمین بیش از حد توسط منتقد رنج می‌برد، ممکن است منجر به سیاست‌های غیربهینه شود.

با در نظر گرفتن معایب کارهای مرتبط، هدف ما ارائه یک مدل بهینه‌سازی برای مسئله مشترک برون‌سپاری محاسبات و تخصیص منابع در سیستم MEC با دستگاه‌های همراه با قابلیت EH و حل آن با استفاده از MADRL است. با توجه به مطالعات انجام‌شده در زمینه پژوهشی، هیچ کار مبتنی بر MADRL وجود ندارد که مسئله بهینه‌سازی مشترک برون‌سپاری بخشی و تخصیص منابع غیرمتمرکز را در یک شبکه MEC با دستگاه‌های مجهز به قابلیت EH مطالعه کرده باشد. برای حل مسئله، یک نسخه بهبودیافته الگوریتم MADDPG با بهبود تخمین ارزش شبکه منتقد و به‌کارگیری تکنیک بازپخش تجربه کارآمدتر را پیشنهاد می‌دهیم.

۳- مدل سیستم و فرموله‌سازی مسئله

در این بخش ابتدا مدل سیستم و پس از آن، مدل EH، مدل ارتباطات و مدل محاسبات ارائه می‌شوند. در نهایت، مسئله بهینه‌سازی مشترک برون‌سپاری محاسبات و تخصیص منابع را با اهداف مورد نظر فرموله می‌کنیم. نمادهای استفاده‌شده این مقاله در جدول ۱ ارائه شده است.

۳-۱ مدل سیستم

یک سیستم MEC تک‌سلولی[8] با چندین دستگاه همراه را در نظر می‌گیریم. همان طور که در شکل ۱ نشان داده شده است، سیستم از یک AP با دو آنتن با یک سرور MEC متصل و K دستگاه همراه تک‌آنتن با منابع محاسباتی و باتری محدود تشکیل شده است. هر دستگاه همراه دارای وظایف حساس به تأخیر و نیازمند محاسبات سنگین است که باید قبل از مهلت زمانی‌شان برای برآورده‌کردن نیازمندی‌های QoS کاربران تکمیل شوند. فرض بر این است که سرور MEC، منابع محاسباتی بیشتری در مقایسه با دستگاه‌های همراه دارد. همچنین فرض می‌کنیم که AP به یک منبع برق پایدار متصل است و مجهز به فرستنده انرژی RF است که می‌تواند قابلیت WPT را انجام دهد و توان را به دستگاه‌های همراه از طریق RF پخش[9] کند. کل زمان سیستم به برش‌های زمانی[10] مساوی با طول υ تقسیم شده و به‌صورت $t \in \{1,\dots,T\}$ نشان داده می‌شود. مشابه [۴۱] و [۴۶]، سناریوی شبه‌ایستا[11] اتخاذ گردیده که در آن محیط شبکه در هر برش زمانی ثابت می‌باشد؛ در حالی که در برش‌های زمانی مختلف متفاوت است. مجموعه دستگاه‌های همراه با $K = \{1,2,\dots,K\}$ نشان داده می‌شود. فرض می‌کنیم که در برش زمانی t، هر دستگاه همراه دارای یک وظیفه قابل تقسیم است که باید با توجه به محدودیت‌های سطح باتری دستگاه و حداکثر تأخیر قابل

1. Stochastic Weight Averaging
2. Internet of Vehicles
3. Temporal Attentional Deterministic Policy Gradient
4. Temporal Feature Extraction Network
5. Rank-Based Priority Experience Replay
6. Parameter Sharing
7. Unmanned Aerial Vehicle
8. Single-Cell
9. Broadcast
10. Time Slot
11. Quasi-Static

جدول ۱: خلاصه نمادهای اصلی مورد استفاده.

نماد	تعریف
K	تعداد دستگاه‌های همراه
$E_k(t)$	باتری باقیمانده دستگاه همراه k در برش زمانی t
$J_{k,j}(t)$	وظیفه j دستگاه همراه k
υ	مدت زمان یک برش زمانی
$\tau_{k,j}^{\max}$	حداکثر تأخیر قابل تحمل وظیفه $J_{k,j}(t)$
$b_{k,j}$	اندازه داده ورودی $J_{k,j}(t)$
$c_{k,j}$	تعداد چرخه‌های مورد نیاز برای پردازش هر بیت وظیفه $J_{k,j}(t)$
$\alpha(t)$	نسبت برش زمانی اختصاص‌یافته به EH
$d_k(t)$	فاصله دستگاه همراه k از AP در برش زمانی t
$H_k(t)$	انرژی برداشت‌شده توسط دستگاه k در برش زمانی t
A_g	بهره آنتن
$g_k(t)$	بهره کانال بین AP و دستگاه همراه k در برش زمانی t
$R_{k,j}(t)$	نرخ انتقال دستگاه همراه k در برش زمانی t
B	پهنای باند کانال
$P_k^{\max}$	حداکثر توان انتقال دستگاه همراه k
$p_{k,j}^{tran}(t)$	توان انتقال تعیین‌شده در برش زمانی t توسط دستگاه همراه k برای ارسال $J_{k,j}(t)$ به AP
f_k^l	قابلیت محاسباتی دستگاه همراه k
$f^s(t)$	منابع محاسباتی در دسترس سرور MEC در برش زمانی t
$\lambda_{k,j}(t)$	نسبت منابع اختصاص‌یافته سرور MEC به $J_{k,j}(t)$
$\tau_{k,j}^l(t)$	تأخیر اجرایی محلی دستگاه k برای پردازش $J_{k,j}(t)$
$e_{k,j}^l(t)$	انرژی مصرف‌شده محلی برای اجرای $J_{k,j}(t)$
$\tau_{k,j}^{stran}(t)$	تأخیر انتقال برای ارسال زیروظیفه $J_{k,j}(t)$ به AP
$e_{k,j}^{stran}(t)$	مصرف انرژی انتقال دستگاه همراه k برای ارسال زیروظیفه $J_{k,j}(t)$
$\tau_{k,j}^{sexe}(t)$	زمان پردازش زیروظیفه $J_{k,j}(t)$ بر روی سرور MEC
$\tau_{k,j}^{so}(t)$	کل تأخیر برون‌سپاری زیروظیفه دستگاه k به سرور MEC
$y_{k,j}(t)$	نسبت برون‌سپاری $J_{k,j}(t)$
$\tau_{k,j}^{tot}(t)$	کل تأخیر پردازش برای $J_{k,j}(t)$
$e_{k,j}^{tot}(t)$	مصرف انرژی کل دستگاه همراه k در برش زمانی t برای تکمیل $J_{k,j}(t)$
$c_{k,j}(t)$	هزینه محاسباتی هر دستگاه همراه

تحمل وظیفه پردازش شود. بنابراین همانند [۳۴] و [۳۵] برون‌سپاری بخشی را اتخاذ می‌کنیم که در آن وظیفه به طور دلخواه به دو زیروظیفه[1] تقسیم می‌گردد و به‌طور مشترک در سرور MEC و دستگاه همراه پردازش می‌شود. در برش زمانی t، وظیفه j دستگاه همراه k با تاپل $J_{k,j}(t)=\langle b_{k,j},\tau_{k,j}^{\max},c_{k,j}\rangle$ نشان داده می‌شود که در آن $b_{k,j}$، اندازه داده ورودی j امین وظیفه k امین دستگاه همراه را نشان می‌دهد و اندازه وظیفه بر اساس توزیع پواسون است. $\tau_{k,j}^{\max}$ و $c_{k,j}$ به ترتیب حداکثر تأخیر قابل تحمل $J_{k,j}(t)$ و تعداد چرخه‌های CPU مورد نیاز برای پردازش هر بیت از $J_{k,j}(t)$ را نشان می‌دهند. علاوه بر این، دستگاه‌های همراه به یک باتری قابل شارژ مجهز هستند. در برش زمانی t، سطح باتری دستگاه همراه k به‌صورت $E_k(t)\in[\underline{E_k},\overline{E_k}]$ مشخص می‌شود که در آن $\overline{E_k}$ و $\underline{E_k}$ به ترتیب حداکثر و حداقل سطح باتری دستگاه هستند. بنابراین برای جلوگیری از اتمام باتری، فرض بر این است که باتری دستگاه همراه با استفاده از فناوری EH در برش‌های زمانی مختلف شارژ می‌شود. ما با توجه به [۴۷] فرض می‌کنیم که محاسبات محلی و EH می‌توانند به طور همزمان انجام شوند. همچنین از آنجایی که EH و برون‌سپاری در باند فرکانسی یکسان انجام می‌شوند، طبق [۴۸] و [۴۹] هر دستگاه از یک مدار تقسیم زمان- مولتی‌پلکس (TDD)[2] برای جداسازی EH و برون‌سپاری و جلوگیری از تداخل استفاده می‌کند؛ پس برون‌سپاری و EH از قانون برداشت و سپس برون‌سپاری[3] تبعیت می‌کند.

به طور خاص در طرح برون‌سپاری پیشنهادی، دستگاه‌های همراه برای دسترسی به منابع محاسباتی سرور MEC با یکدیگر رقابت می‌کنند و هر دستگاه همراه سیاست برون‌سپاری را برای هر وظیفه به طور مستقل یاد می‌گیرد. در هر برش زمانی، هر دستگاه همراه بر اساس مشاهدات محلی، سه تصمیم می‌گیرد: ۱) نسبت برون‌سپاری: نسبتی از وظیفه که به سرور MEC بارگذاری می‌شود، ۲) مدت زمان برداشت انرژی: نسبتی از برش زمانی که دستگاه همراه به برداشت انرژی اختصاص می‌دهد و ۳) توان انتقال: مقدار توان انتقال دستگاه همراه برای برون‌سپاری وظیفه که با توجه به سطح باتری و حداکثر تأخیر قابل تحمل وظیفه تعیین می‌شود.

۳-۲ مدل برداشت انرژی

همان طور که قبلاً ذکر شد، دستگاه‌های همراه به قابلیت EH مجهز هستند و می‌توانند سیگنال‌های RF ارسال‌شده توسط AP را جمع‌آوری کنند. سپس انرژی ذخیره‌شده در باتری دستگاه‌ها برای انجام پردازش محلی و برون‌سپاری وظایف به سرور MEC استفاده می‌شود. در طرح برون‌سپاری پیشنهادی در ابتدای برش زمانی t، دستگاه همراه k برای $\alpha(t)\upsilon$ شارژ می‌شود که υ نشان‌دهنده مدت زمان هر برش زمانی و $\alpha(t)$ نسبت برش زمانی اختصاص‌داده‌شده به EH است. بنابراین مشابه [۳۶]، انرژی برداشت‌شده توسط دستگاه همراه k در برش زمانی t، $H_k(t)$، با استفاده از معادله زیر محاسبه می‌شود [۳۶]

$$H_k(t)=\mu P g_k(t)\alpha(t)\upsilon \qquad (۱)$$

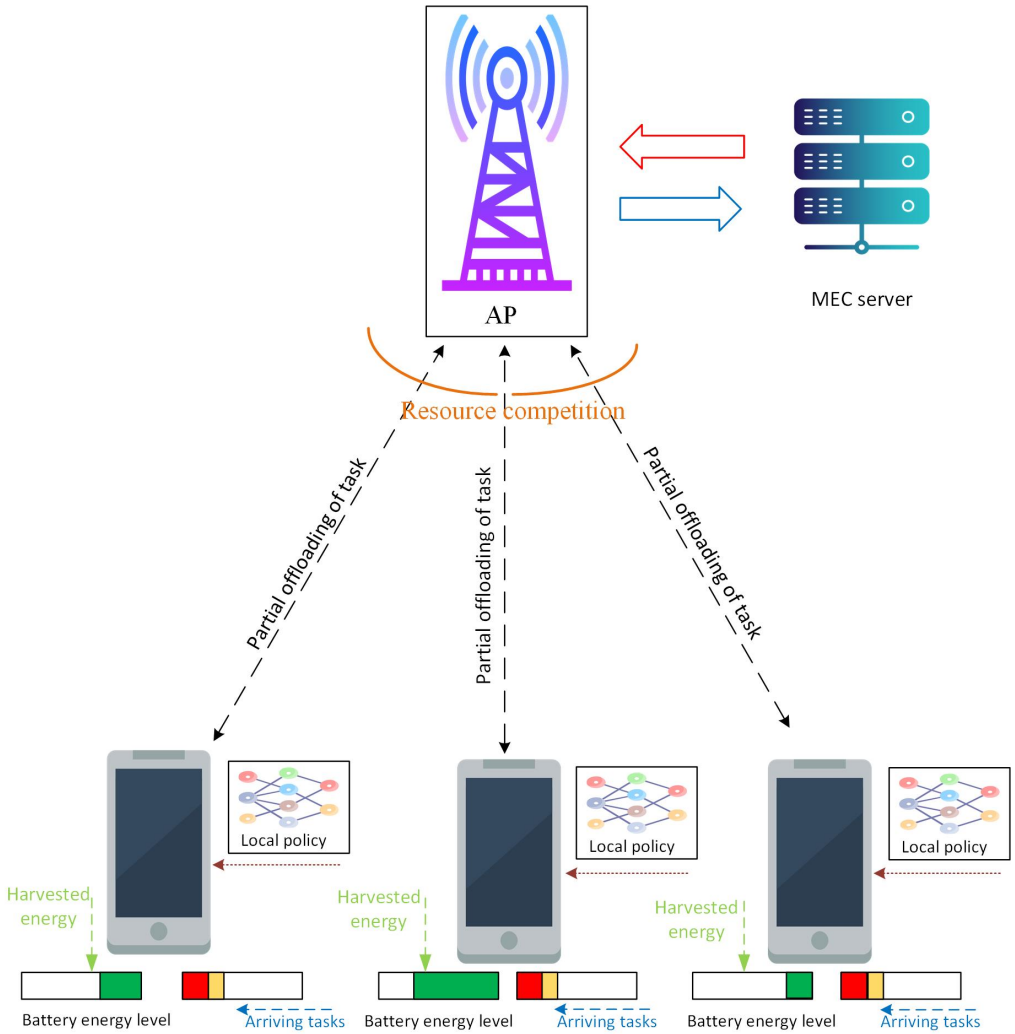
در (۱)، $\mu\in(۰,۱)$ و P به ترتیب کارایی EH و توان انتقال AP هستند. $g_k(t)$ بهره کانال بین دستگاه همراه k و AP در برش زمانی t است و بر اساس مدل کانال محوشدگی رایلی[4] [۴۸] است که با استفاده از $g_k(t)=\varepsilon(t)\overline{g_k(t)}$ محاسبه می‌شود که در آن $\varepsilon(t)$ یک متغیر تصادفی نمایی مستقل با میانگین واحد را نشان می‌دهد و میانگین بهره کانال $\overline{g_k(t)}$ [۴۹] با استفاده از معادله زیر به‌دست می‌آید

$$\overline{g_k(t)}=A_g\left(\frac{۳\times۱۰^{۸}}{۴\pi f_c d_k(t)}\right)^{l_e} \qquad (۲)$$

در (۲)، f_c، A_g و l_e به ترتیب فرکانس حامل، بهره آنتن و توان افت مسیر[5] را نشان می‌دهند. $d_k(t)$ فاصله دستگاه همراه k تا AP در برش برش زمانی t است که همانند [۵۰] با استفاده از زنجیره مارکوف مدل شده است. باتری باقیمانده دستگاه همراه k در پایان برش زمانی t با استفاده از (۳) محاسبه می‌شود

$$E_k(t+۱)=\min\{\max\{E_k(t)+H_k(t)-e_{k,j}^{tot}(t),\underline{E_k}\},\overline{E_k}\} \qquad (۳)$$

1. Subtask
2. Time-Division-Multiplexing
3. Harvest-then-Offload Rule
4. Rayleigh Fading Channel Model
5. Path Loss Exponent



شکل ۱: شماتیک سیستم MEC با دستگاه‌های همراه مختلف مجهز به قابلیت EH.

که در آن $H_k(t)$ مقدار انرژی برداشت‌شده در برش زمانی t و $E_k(t)$ مقدار باتری دستگاه همراه k در ابتدای برش زمانی t را نشان می‌دهند. $e^{tot}_{k,j}(t)$ مقدار کل انرژی مصرف‌شده در برش زمانی t توسط دستگاه همراه k برای پردازش وظایف به‌صورت محلی یا انتقال آنها به سرور MEC است و با استفاده از (۱۳) محاسبه می‌شود. اگر باتری دستگاه همراه کافی نباشد، یعنی $E_k(t+1) \le \underline{E_k}$، وظیفه فعلی حذف خواهد شد.

۳-۳ مدل ارتباطات

در این کار، مشابه با [۲۵] و [۴۱] فرض می‌کنیم AP منابع ارتباطات و منابع محاسباتی سرور MEC را به هر دستگاه همراه متناسب با بار کاری وظیفه آن اختصاص می‌دهد. از این رو نسبت منابع ارتباطات و محاسبات تخصیص‌یافته به زیروظیفه $J_{k,j}(t)$ به‌صورت زیر محاسبه می‌شود

$$\lambda_{k,j}(t) = \frac{y_{k,j}(t) b_{k,j}}{\sum_{k=1}^{K} y_{k,j}(t) b_{k,j}} \qquad (۴)$$

که در آن $\sum_{k=1}^{K} y_{k,j}(t) b_{k,j}$ کل اندازه داده ورودی ارسال‌شده (بارکاری) به AP را نشان می‌دهد.

در صورتی که $p^{tran}_{k,j}(t)$ و $P^{\max}_k$ توان انتقال تعیین‌شده توسط دستگاه همراه k برای ارسال وظیفه $J_{k,j}(t)$ به AP و حداکثر توان انتقال آن را نشان دهند، نرخ انتقال uplink دستگاه همراه k در برش زمانی t به‌صورت زیر محاسبه می‌شود [۴۳]

$$R_{k,j}(t) = \lambda_{k,j}(t) B \log_2\left(1 + \frac{p^{tran}_{k,j}(t) g_k(t)}{\sigma^2}\right) \qquad (۵)$$

که در آن B منابع ارتباطات در دسترس یعنی پهنای باند AP در برش زمانی t و σ^2 توان نویز است.

۴-۳ مدل محاسبات

در این بخش، تأخیر و مصرف انرژی محاسبات محلی و برون‌سپاری محاسبات مورد بحث قرار گرفته است.

۱) محاسبه محلی: همان طور که قبلاً اشاره کردیم، وظایف را می‌توان به طور دلخواه به دو زیروظیفه برای اجرای محلی و اجرای راه دور تقسیم کرد. در برش زمانی t، دستگاه همراه k نسبتی از وظیفه، $y_{k,j}(t)$، را تعیین می‌کند تا به سرور MEC برون‌سپاری شود که در آن $0 \le y_{k,j}(t) \le 1$ و نسبت باقیمانده از وظیفه یعنی $1 - y_{k,j}(t)$ به‌صورت محلی پردازش می‌شود. تأخیر اجرای محلی در دستگاه همراه k به‌صورت زیر محاسبه می‌شود

$$\tau^l_{k,j}(t) = \frac{(1 - y_{k,j}(t)) b_{k,j} c_{k,j}}{f^l_k} \qquad (۶)$$

در (۶) f^l_k قابلیت محاسباتی دستگاه همراه k را برای پردازش وظایف نشان می‌دهد. مطابق با [۴۱] مصرف انرژی پردازش محلی برای دستگاه همراه k با استفاده از (۷) مشخص می‌شود

$$e^{l}_{k,j}(t) = \kappa (f^{l}_{k})^{۲} (۱ - y_{k,j}(t)) b_{k,j} c_{k,j} \quad (۷)$$

۲) برون‌سپاری محاسبات: زمان لازم برای تکمیل پردازش وظیفه در مورد برون‌سپاری شامل سه بخش است: ۱) تأخیر انتقال uplink، ۲) تأخیر اجرای زیروظیفه در سرور MEC و ۳) دانلود نتیجه محاسباتی وظیفه. مطابق با [۴۰] و [۴۱] از آنجا که مقدار نتیجه محاسبات بسیار کوچک‌تر از داده‌های وظیفه است، تأخیر انتقال و مصرف انرژی ناشی از دانلود نتیجه نادیده گرفته می‌شود. شایان ذکر است که ما فرض کرده‌ایم سرور MEC از طریق سیم مسی یا فیبر نوری به AP متصل است؛ بنابراین از تأخیر انتقال بین سرور MEC و AP صرف نظر می‌شود. تأخیر انتقال برای ارسال زیروظیفه دستگاه همراه k به AP، $\tau^{stran}_{k,j}(t)$، با استفاده از معادله زیر محاسبه می‌شود

$$\tau^{stran}_{k,j}(t) = \frac{y_{k,j} b_{k,j}}{R_{k,j}(t)} \quad (۸)$$

علاوه بر این، انرژی مصرف‌شده دستگاه همراه k برای برون‌سپاری زیروظیفه‌اش به AP در برش زمانی t، $e^{stran}_{k,j}(t)$، برابر است با

$$e^{stran}_{k,j}(t) = p^{tran}_{k,j}(t) \tau^{stran}_{k,j}(t) \quad (۹)$$

ما $f^{s}(t)$ را به‌صورت منابع محاسباتی در دسترس سرور MEC در برش زمانی t نشان می‌دهیم. بنابراین زمان لازم برای پردازش زیروظیفه وظیفه $J_{k,j}(t)$ در سرور MEC، $\tau^{sexe}_{k,j}(t)$، به‌صورت زیر مشخص می‌شود

$$\tau^{sexe}_{k,j}(t) = \frac{y_{k,j}(t) b_{k,j} c_{k,j}}{\lambda_{k,j}(t) f^{s}(t)} \quad (۱۰)$$

که در آن $\lambda_{k,j}(t)$ با استفاده از (۴) به‌دست می‌آید و $\lambda_{k,j}(t) f^{s}(t)$ منبع محاسباتی تخصیص‌یافته توسط سرور MEC به دستگاه همراه k را نشان می‌دهد. نهایتاً تأخیر تجربه‌شده کل برون‌سپاری زیروظیفه وظیفه $J_{k,j}(t)$ به سرور MEC با استفاده از معادله زیر محاسبه می‌شود

$$\tau^{so}_{k,j}(t) = \tau^{stran}_{k,j}(t) + \tau^{sexe}_{k,j}(t) \quad (۱۱)$$

۳) هزینه کل سیستم: در برش زمانی t، دستگاه همراه k وظیفه محاسباتی را به دو زیروظیفه تقسیم می‌کند و زیروظایف را می‌توان به‌صورت موازی در سرور MEC و دستگاه همراه پردازش کرد. همان طور که ذکر شد، در این مقاله مشابه [۳۶] و [۴۸]، فرض شده که دستگاه‌های همراه می‌توانند اجرای محلی و EH را به طور همزمان انجام دهند؛ در حالی که عملیات برون‌سپاری پس از تکمیل فرایند EH انجام می‌شود. بنابراین کل تأخیر پردازش اعمال‌شده برای وظیفه $J_{k,j}(t)$ با استفاده از (۱۲) تعریف می‌شود

$$\tau^{tot}_{k,j}(t) = \max(\tau^{l}_{k,j}(t), \alpha_k(t)\upsilon + \tau^{so}_{k,j}(t)) \quad (۱۲)$$

انرژی کل مصرف‌شده توسط دستگاه همراه k برابر است با مجموع انرژی مصرف‌شده در اجرای محلی و انرژی مصرف‌شده برای انتقال زیروظیفه وظیفه $J_{k,j}(t)$ به سرور MEC که مطابق زیر به دست می‌آید

$$e^{tot}_{k,j}(t) = e^{l}_{k,j}(t) + e^{stran}_{k,j}(t) \quad (۱۳)$$

برای ارزیابی سیاست برون‌سپاری دستگاه همراه k، هزینه محاسباتی هر دستگاه در برش زمانی t از لحاظ مجموع وزن‌دار مصرف انرژی و تأخیر کل پردازش وظیفه با استفاده از (۱۴) محاسبه می‌شود

$$c_{k,j}(t) = w \cdot \tau^{tot}_{k,j}(t) + (۱ - w) \cdot e^{tot}_{k,j}(t) \quad (۱۴)$$

که در آن $w \in [۰,۱]$ پارامتر وزن است و با توجه به اولویت‌های دستگاه همراه در مورد تأخیر پردازش و مصرف انرژی تعیین می‌شود.

٤) جریمه شکست وظیفه: زمانی که تأخیر پردازش وظیفه از حداکثر تأخیر قابل تحمل آن بیشتر شود یا زمانی که سطح باتری دستگاه همراه برای تکمیل وظیفه کافی نیست، جریمه‌ای متناسب با میزان تخطی، تعیین می‌شود. مقدار جریمه شکست اجرای وظیفه به دلیل ناکافی‌بودن سطح باتری دستگاه همراه و برآورده‌نشدن حداکثر تأخیر قابل تحمل وظیفه به ترتیب با استفاده از (۱۵) و (۱۶) بیان می‌شود

$$Pen^{e}_{k,j}(t) = \max(\frac{e^{tot}_{k,j}(t)}{E_k(t)} - ۱, ۰) \quad (۱۵)$$

$$Pen^{l}_{k,j}(t) = \max(\frac{\tau^{tot}_{k,j}(t)}{\tau^{\max}_{k,j}} - ۱, ۰) \quad (۱۶)$$

۳-۵ فرمولاسیون مسئله

آخرین برش زمانی سپری‌شده در سیستم MEC با T و تعداد وظایف پردازش‌شده تا زمان T با $Q(T)$ نشان داده می‌شود. در این کار، هدف ما کاهش هزینه محاسباتی بلندمدت تمام وظایف با بهینه‌سازی مشترک سیاست برون‌سپاری محاسبات و تخصیص منابع برای هر دستگاه همراه است که در آن، دستگاه همراه استراتژی نسبت برون‌سپاری، $y_{k,j}(t)$، استراتژی مدت زمان EH، $\alpha_k(t)$، و استراتژی توان انتقال، $P^{tran}_{k,j}(t)$ را تعیین می‌کند. نهایتاً مسئله به‌صورت زیر فرموله می‌شود

$$\min_{y_{k,j}(t), \alpha_k(t), p^{tran}_{k,j}(t)} (\lim_{T \to \infty} \sum_{k=۱}^{K} \frac{۱}{Q(T)} \sum_{j=۱}^{Q(T)} c_{k,j}(t)) \quad (۱۷)$$

$$C۱ : y_{k,j}(t) \in [۰,۱] \ , \ k \in K \ , \ j \in Q(T) \quad (۱۸)$$

$$C۲ : \alpha_k(t) \in [۰,۱] \ , \ k \in K \quad (۱۹)$$

$$C۳ : p^{tran}_{k,j}(t) \in [۰, P^{\max}_{k}] \ , \ k \in K \ , \ j \in Q(T) \quad (۲۰)$$

$$C۴ : \sum_{k \in K} \lambda_{k,j}(t) = ۱ \ , \ k \in K \ , \ j \in Q(T) \quad (۲۱)$$

$$C۵ : \tau^{tot}_{k,j}(t) \le \tau^{\max}_{k,j} \ , \ k \in K \ , \ j \in Q(T) \quad (۲۲)$$

$$C۶ : e^{tot}_{k,j}(t) \le E_k(t) \ , \ k \in K \ , \ j \in Q(T) \quad (۲۳)$$

محدودیت‌های (۱۸) و (۱۹) مشخص می‌کنند که نسبت برون‌سپاری و مدت زمان برداشت انرژی بین صفر و یک متغیر است. محدودیت (۲۰) بیان می‌کند که توان انتقال دستگاه همراه نمی‌تواند از حداکثر توان آن بیشتر باشد. محدودیت (۲۱) تضمین می‌کند که کل منابع تخصیص‌یافته به دستگاه‌های همراه نمی‌تواند بیشتر از منابع موجود سرور MEC باشد. محدودیت (۲۲) الزام می‌کند که محدودیت مربوط به حداکثر تأخیر قابل تحمل وظیفه نقض نشود. محدودیت (۲۳) تضمین می‌کند که مصرف انرژی دستگاه همراه k برای انجام وظیفه کمتر از باتری باقیمانده آن باشد. مسئله بهینه‌سازی فرموله‌شده در (۱۷) یک مسئله NLP با ماهیت NP-hard است؛ بنابراین پیچیدگی زمانی با افزایش دستگاه‌های همراه افزایش می‌یابد. علاوه بر این، محیط بسیار پویای سیستم MEC استفاده از رویکردهای بهینه‌سازی سنتی را برای یافتن راه‌حل‌های بهینه بلادرنگ چالش‌برانگیز می‌کند. از سوی دیگر، استفاده از تکنیک‌های متمرکز برای حل مسئله فوق به دلیل نیاز به سرور مرکزی برای جمع‌آوری تمام اطلاعات محیط MEC و سپس توزیع تصمیمات به دستگاه‌های همراه منجر به افزایش سربار سیستم و تأخیر اضافی می‌شود که ممکن است برای برنامه‌های حساس به تأخیر مناسب نباشد. بنابراین در بخش بعدی روشی مبتنی بر MADRL برای حل مسئله (۱۷) پیشنهاد می‌کنیم.

٤- طرح برون‌سپاری محاسبات غیرمتمرکز مبتنی بر MADRL

برای یادگیری سیاست‌های برون‌سپاری و تخصیص منبع غیرمتمرکز در محیط پویای MEC با هدف حداقل‌سازی هزینه محاسباتی بلندمدت سیستم، الگوریتم MADDPG که مبتنی بر چارچوب CTDE است به کار گرفته شده است. ما به طور خاص برای بهبود عملکرد سیاست آموخته‌شده، یک ورژن بهبودیافته الگوریتم MADDPG را پیشنهاد می‌دهیم. در ادامه، عناصر کلیدی DRL شرح داده خواهند شد.

٤-١ عناصر کلیدی DRL

هر دستگاه همراه در طرح برون‌سپاری پیشنهادی به عنوان یک عامل در نظر گرفته می‌شود. عامل k در هر برش زمانی پس از تعامل با محیط سیستم MEC، اطلاعات محلی را مشاهده می‌کند و پس از آن اقدام $a_k(t)$ را بر اساس مشاهدات محلی $o_k(t)$ از طریق سیاست آموخته‌شده انجام می‌دهد. پس از انجام اقدام $a_k(t)$، عامل پاداش فوری $r_k(o_k(t),a_k(t))$ را از محیط دریافت می‌نماید. هدف عامل یافتن یک سیاست بهینه است که پاداش بلندمدت $R_k(T)$ را در یک اپیزود تصمیم‌گیری به حداکثر می‌رساند. در ادامه، فضاهای حالت[1] و اقدام[2] و همچنین تابع پاداش[3] تعریف می‌شوند.

فضای حالت: در برش زمانی t، فضای حالت عامل k شامل میزان باتری دستگاه همراه $E_k(t)$ است که با استفاده از (۳) به دست می‌آید. سایر پارامترهای فضای حالت عبارت هستند از منابع محاسباتی در دسترس سرور MEC، $f^s(t)$، بهره کانال بین دستگاه همراه و AP، $g_k(t)$، نیازمندی‌های وظیفه شامل حداکثر تأخیر قابل تحمل وظیفه، $\tau_{k,j}^{\max}$، اندازه داده وظیفه، $b_{k,j}$، و تعداد چرخه‌های CPU مورد نیاز برای پردازش هر بیت وظیفه، $c_{k,j}$. بنابراین فضای حالت عامل k به‌صورت $o_k(t)=\{E_k(t),f^s(t),g_k(t),\tau_{k,j}^{\max},b_{k,j},c_{k,j}\}$ نشان داده می‌شود. در نتیجه در برش زمانی t، حالت مشترک همه عامل‌ها به‌صورت $S(t)=\{o_1(t),o_2(t),\dots,o_k(t)\}$ تعریف می‌شود.

فضای اقدام: در برش زمانی t، عامل k استراتژی نسبت برون‌سپاری وظیفه $J_{k,j}(t)$، استراتژی مدت زمان EH، $\alpha_k(t)$، و استراتژی توان انتقال برای ارسال‌کردن وظیفه $J_{k,j}(t)$ به AP، $p_{k,j}^{tran}(t)$، را تعیین می‌کند و در نتیجه اقدام عامل k به‌صورت $a_k(t)=\{y_{k,j}(t),\alpha_k(t),p_{k,j}^{tran}(t)\}$ بیان می‌گردد. اقدام مشترک همه عامل‌ها به‌صورت $A(t)=\{a_1(t),a_2(t),\dots,a_k(t)\}$ نشان داده می‌شود.

پاداش: هدف عامل k یافتن سیاست بهینه برون‌سپاری محاسبات و تخصیص منبع غیرمتمرکز با بالاترین پاداش بلندمدت $R_k(T)$ است که به‌صورت زیر تعریف می‌شود

$$R_k(T)=\sum_{t=1}^{T}\gamma^t\cdot r_k(o_k(t),a_k(t)) \qquad (۲۴)$$

که $r_k(o_k(t),a_k(t))$ پاداش فوری عامل k است و اهمیت هر پاداش فوری بر اساس فاکتور تخفیف[4] $\gamma\in[0,1]$ تعیین می‌شود. پاداش فوری عامل در برش زمانی t، مطابق با معکوس هزینه محاسباتی و جریمه‌های دریافتی در نتیجه نقض محدودیت‌های انرژی و تأخیر به دست می‌آید

$$r_k(o_k(t),a_k(t))=-(c_{k,j}(t)+Pen_{k,j}^{e}(t)+Pen_{k,j}^{l}(t)) \qquad (۲۵)$$

در (۲۵)، $c_{k,j}(t)$ هزینه محاسباتی عامل از لحاظ مصرف انرژی و تأخیر پردازش وظیفه است و با استفاده از (۱۴) به دست می‌آید. $Pen_{k,j}^{e}(t)$ و $Pen_{k,j}^{l}(t)$ مقادیر جریمه‌ای هستند که عامل در صورت نقض محدودیت‌های انرژی و حداکثر تأخیر قابل تحمل وظیفه دریافت می‌کند و به ترتیب با استفاده از (۱۵) و (۱۶) محاسبه می‌شود.

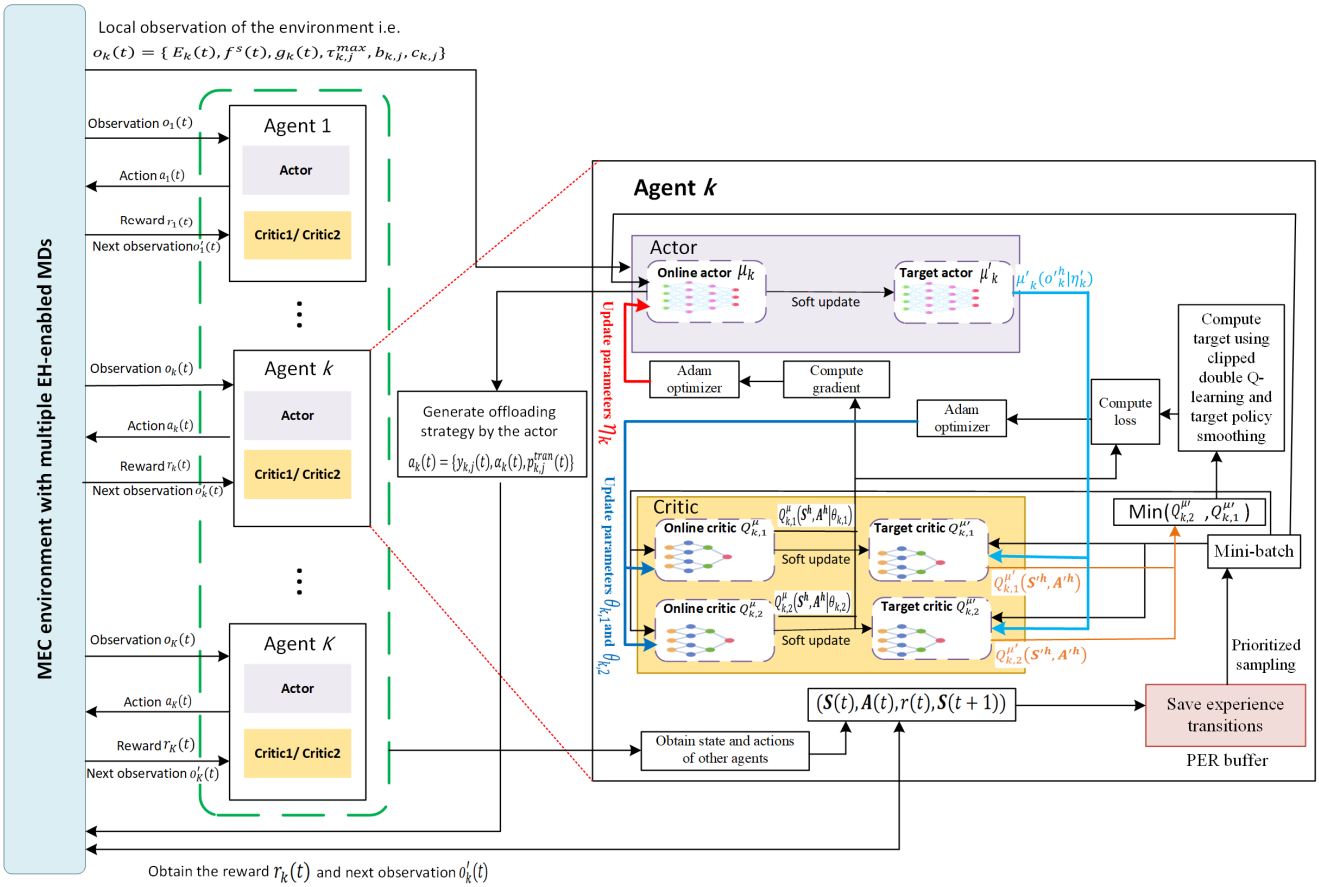
٤-٢ الگوریتم پیشنهادی

در این کار، شبکه MEC به عنوان یک محیط RL در نظر گرفته شده است. هر دستگاه همراه به عنوان عاملی در نظر گرفته می‌شود که الگوریتم پیشنهادی را برای تصمیم‌گیری مشترک برون‌سپاری محاسبات و تخصیص منبع غیرمتمرکز اجرا می‌کند؛ بنابراین محیط به عنوان یک سیستم چندعامله در نظر گرفته می‌شود. یک رویکرد برای پیاده‌سازی سیستم‌های MADRL استفاده از چارچوب IL است که در آن هر عامل به طور مستقل سیاست مشترک برون‌سپاری و تخصیص منبع غیرمتمرکز را یاد می‌گیرد. با این حال، این چارچوب با مسائلی مانند عدم ثبات، قابلیت مشاهده جزئی و واریانس بالا روبه‌رو است. بنابراین در این چارچوب با افزایش تعداد عامل‌ها، عملکرد سیستم کاهش یافته و همگرایی الگوریتم را نمی‌توان تضمین کرد [۱۴] و [۱۷]. برای غلبه بر معایب فوق، رویکرد دیگر استفاده از چارچوب CTDE است. MADDPG [۱۷] توسعه الگوریتم DDPG بر اساس چارچوب CTDE است. DDPG [۱۸] شامل دو شبکه بازیگر و منتقد است که در آن شبکه بازیگر از روش گرادیان سیاست قطعی (DPG)[5] [۵۱] برای تعیین اقدام مناسب بر اساس حالت مشاهده‌شده استفاده می‌کند؛ در حالی که شبکه منتقد با استفاده از روش‌های مبتنی بر ارزش، ارزش سیاست آموخته‌شده بازیگر را تقریب می‌زند. هم بازیگر و هم منتقد دارای دو زیرشبکه با ساختار یکسان هستند: یعنی بازیگر آنلاین و بازیگر هدف و منتقد آنلاین و منتقد هدف. تفاوت اصلی MADDPG و DDPG ورودی شبکه منتقد است. MADDPG از یک شبکه منتقد متمرکز استفاده می‌کند که در طول آموزش، اطلاعات بیشتری در مورد سیاست‌های سایر عامل‌ها دریافت می‌کند. با این حال در زمان اجرا، بازیگر تنها بر اساس مشاهدات محلی تصمیم می‌گیرد. این تغییر، محیط را از دیدگاه هر عامل در سیستم‌های چندعاملی ثابت می‌کند.

در روش‌های مبتنی بر بازیگر- منتقد، منتقد کیفیت سیاست آموخته‌شده بازیگر را با تخمین ارزش آن ارزیابی می‌کند و بازیگر، سیاست را بر اساس تخمین ارزش منتقد به‌روزرسانی می‌نماید. بنابراین در این روش‌ها برای این که بازیگر بتواند یک سیاست بهتر را یاد بگیرد، تخمین تابع ارزش در روشی مناسب ضروری است. با این حال، مسئله تخمین بیش از حد ارزش می‌تواند در روش‌های بازیگر- منتقد به دلیل خطاهای تقریب تابع رخ دهد که منجر به به‌روزرسانی‌های غیربهینه سیاست و رفتار واگرا می‌شود [۱۹] و [۵۲]. برای پرداختن به مسئله بایاس تخمین بیش از حد و اصلاح تخمین ارزش شبکه منتقد، عملکرد همگرایی الگوریتم MADDPG را با به‌کارگیری clipped double Q-learning، به‌روزرسانی‌های با تأخیر سیاست و هموارسازی سیاست هدف بهبود می‌دهیم. در ادامه جزئیات الگوریتم پیشنهادی توضیح داده خواهد شد.

ساختار طرح برون‌سپاری غیرمتمرکز مبتنی بر الگوریتم پیشنهادی در شکل ۲ نشان داده شده است. در الگوریتم پیشنهادی ما هر عامل دارای

1. State Space
2. Action Space
3. Reward Function
4. Discount Factor
5. Deterministic Policy Gradient



شکل ۲: ساختار طرح برون‌سپاری مبتنی بر الگوریتم پیشنهادی.

یک تابع بازیگر (که به عنوان سیاست نیز شناخته می‌شود) به‌صورت $\mu_k(o_k(t)|\eta_k)$ است. بنابراین $\mu = \{\mu_1, \mu_2, \ldots, \mu_K\}$ مجموعه سیاست همه عامل‌ها و $\eta = \{\eta_1, \eta_2, \ldots, \eta_K\}$ مجموعه پارامترهای مربوطه سیاست‌ها را نشان می‌دهد. همچنین هر عامل دو تابع منتقد متمرکز به‌صورت $Q^{\mu}_{k,j}(S(t), A(t)|\theta_{k,j})$ دارد که در آن $j \in \{1,2\}$ است. هر عامل یک کپی از شبکه‌های بازیگر و منتقد متمرکز را برای بهبود پایداری یادگیری نگه می‌دارد؛ یعنی شبکه‌های بازیگر هدف و منتقد متمرکز هدف که به ترتیب به‌صورت $\mu'_k(o_k(t)|\eta'_k)$ و $Q^{\mu'}_{k,j}(S(t), A(t)|\theta'_{k,j})$ نشان داده می‌شوند. در طول فرایند آموزش، mini-batch ی از تجربیات با اندازه H با استفاده از نمونه‌برداری اولویت‌بندی‌شده که در بخش ۴-۳ شرح داده شده است، انتخاب می‌گردد که برای به‌روزرسانی پارامترهای شبکه‌های منتقد متمرکز عامل k با حداقل‌سازی loss استفاده می‌شود

$$L(\theta_{k,j}) = \frac{1}{H}\sum_{h=1}^{H}(y^h - Q^{\mu}_{k,j}(S^h, A^h|\theta_{k,j}))^2 \quad (۲۶)$$

که در آن y^h ارزش هدف[1] را نشان می‌دهد. ما به‌منظور کاهش مسئله تخمین بایاس بیش از حد از clipped double Q-learning [۱۹] برای محاسبه ارزش هدف استفاده می‌کنیم که در آن، حداقل ارزش بین دو شبکه منتقد در نظر گرفته می‌شود. از طرفی در صورتی که شبکه منتقد متمرکز، ارزش برخی اقدامات را به نادرستی بیش از حد تخمین بزند، شبکه بازیگر به‌سرعت از آن بهره‌برداری می‌کند و سپس رفتاری نادرست خواهد داشت. ما برای حل این مسئله از هموارسازی سیاست هدف [۱۹] استفاده می‌کنیم که به‌عنوان یک تنظیم‌کننده[2] برای الگوریتم عمل می‌کند و با اضافه‌کردن نویز clipشده به اقدامات تعیین‌شده توسط سیاست هدف، بهره‌برداری از خطاهای تابع Q را برای سیاست سخت‌تر می‌کند. بنابراین برای محاسبه ارزش هدف از معادله زیر استفاده می‌شود

$$y^h = r^h_k + \gamma \min_{j=1,2} Q^{\mu'}_{k,j}(S'^h, A'^h)\Big|_{A'^h = \{clip((\mu'_n(o'^h_n|\eta'_n)+\zeta))^K_{n=1}\}} \quad (۲۷)$$

$$\zeta \sim clip(N(0,\sigma), -c, c) \quad (۲۸)$$

که ζ نویز clipشده است. در MADDPG نویز به اقدام خروجی نهایی هنگام تعامل با محیط برای اکتشاف اضافه می‌گردد؛ در حالی که در الگوریتم پیشنهادی، نویز به اقدام ارائه‌شده توسط بازیگر هنگام محاسبه هدف اضافه می‌شود، برای آن که ارزش هدف دقیق‌تر شود.

هدف عامل k یافتن سیاست بهینه μ با حداکثر پاداش مورد انتظار درازمدت یعنی $J(\eta_k) = \mathbb{E}[R_k]$ است. هر عامل می‌تواند به طور مستقل با نگاشت قطعی مشاهدات محلی خود به اقدامات تصمیم بگیرد؛ یعنی $a_k = \mu_k(o_k(t))$. برای کاهش هزینه‌های محاسباتی، شبکه بازیگر با استفاده از یکی از شبکه‌های منتقد متمرکز یعنی $Q^{\mu}_{k,1}$ بهینه می‌شود. ممکن است در هر مرحله به‌روزرسانی، خطاهای کوچکی ایجاد گردد و زمانی که شبکه بارها به‌روزرسانی می‌شود، انباشتگی آن می‌تواند منجر به عملکرد ضعیف شود. بنابراین برای کاهش خطا در هر به‌روزرسانی و بهبود عملکرد، شبکه‌های سیاست و هدف با فرکانس کمتری نسبت به شبکه منتقد به‌روز می‌شوند و آنها را پس از تعداد ثابتی به‌روزرسانی d برای منتقد به‌روز می‌کنیم. به‌روزرسانی‌های سیاست که با فرکانس کمتری رخ می‌دهند، از یک تخمین ارزش با واریانس کمتر استفاده می‌کنند و در اصل، منجر به به‌روزرسانی سیاست با کیفیت بالاتر می‌شوند. بنابراین سیاست μ_k هر عامل k با استفاده از گرادیان تابع هدف نسبت به پارامترهای آن η_k با اعمال قانون زنجیره‌ای به‌صورت زیر به‌روز می‌شود

1. Target Value
2. Regularizer

$$\nabla_{\eta_k} J(\eta_k) \approx \frac{1}{H}\sum_{h=1}^{H} \nabla_{\eta_k} \times \mu_k(o_k^h|\eta_k)\nabla_{a_k} Q_{k,1}^{\mu}(S^h, a_1^h, a_2^h, \ldots, a_K^h|\theta_{k,1})\Big|_{a_k^h=\mu_k(o_k^h)} \quad (۲۹)$$

در (۲۹) $Q_{k,1}^{\mu}(S^h, a_1^h, a_2^h, \ldots, a_K^h|\theta_{k,1})$ تابع منتقد متمرکز است که حالت و اقدام مشترک همه عامل‌ها را به عنوان ورودی دریافت می‌کند و ارزش اقدام سیاست عامل k را تقریب می‌زند.

۴-۳ بازپخش تجربه اولویت‌بندی‌شده

در نسخه استاندارد MADDPG [۱۷]، مکانیسم بازپخش تجربه یکنواخت استفاده شده است. بازپخش تجربه مکانیزمی است که امکان ذخیره تجربیات گذشته در یک بافر بازپخش و استفاده مجدد از آنها را در طول آموزش می‌دهد. بازپخش تصادفی تجارب از بافر تجربه، همبستگی‌های زمانی بین تجربه‌های متوالی را کاهش می‌دهد و نمونه‌های مستقل با توزیع یکسان (i.i.d.)[1] را که برای آموزش DNN مورد نیاز است، فراهم می‌کند [۵۳]. با این حال در بازپخش تجربه یکنواخت، تجربیات با اهمیت یکسان در نظر گرفته می‌شوند؛ در حالی که عامل می‌تواند از برخی تجربیات نسبت به بقیه بیشتر بیاموزد. به عنوان مثال، تجارب مربوط به تلاش‌های موفق و تجربیات مربوط به عملکرد نادرست عامل، ارزش بیشتری نسبت به سایر تجربیات دارند. تجارب موفق عامل را وادار می‌کند تا در موقعیت‌های مشابه، اقدامات مشابهی انجام دهد. از سوی دیگر، تجربیات ناموفق به عامل امکان می‌دهد تا به سرعت عواقب منفی رفتارهای اشتباه در موقعیت‌های مربوط را درک و از انجام مجدد اقدامات اشتباه در این شرایط اجتناب کند [۵۴]. از این رو ما مکانیسم PER [۵۵] را در MADDPG به کار می‌گیریم که در آن، تجربیات باارزش‌تر شانس بیشتری برای بازپخش برای آموزش عامل دارند. این امر می‌تواند به فرایند آموزش عامل سرعت بخشد و منجر به پایداری بیشتر آن شود. به طور خاص، تجربیات به‌دست‌آمده از تعامل عامل با محیط به‌صورت تاپل $(S(t), A(t), r(t), S(t+1))$ در بافر بازپخش ذخیره می‌گردند و یک مقدار اولویت به آنها اختصاص داده می‌شود. مقدار اولویت هر تجربه متناسب با خطای TD آن در نظر گرفته می‌شود؛ زیرا تجارب با خطای TD منفی بزرگ مربوط به عملکرد ناموفق عامل و تجارب با خطای TD مثبت بزرگ مربوط به عملکرد موفق هستند [۵۴]. در طول آموزش، یک mini-batch از تجارب از بافر بازپخش نمونه‌برداری می‌شود. بدین منظور از اولویت‌بندی تصادفی استفاده شده که به تجربیاتی با مقدار اولویت کم نیز امکان می‌دهد تا شانس انتخاب‌شدن داشته باشند [۵۵]. به این ترتیب، تنوع در تجارب بازپخش‌شده افزایش می‌یابد و از بیش‌برازش[2] عامل به نمونه‌های دارای مقدار اولویت بالا جلوگیری می‌شود. احتمال بازپخش هر تجربه در mini-batch، $P(i)$، متناسب با مقدار اولویت آن محاسبه می‌شود

$$P(i) = \frac{(pri_i)^l}{\sum_h (pri_h)^l} \quad (۳۰)$$

در (۳۰) پارامتر l میزان اولویت‌بندی را کنترل می‌کند. pri_i مقدار اولویت تجربه i را مشخص می‌کند و به‌صورت $pri_i = |\delta_i| + \varepsilon$ محاسبه می‌شود. $|\delta_i|$ نشان‌دهنده قدرمطلق خطای TD تجربه i است و ε ثابتی با مقدار مثبت و کوچک است که از صفرشدن مقدار اولویت هر تجربه جلوگیری می‌کند.

۴-۴ فرایند آموزش

همان طور که گفته شد، هر دستگاه همراه به عنوان یک عامل در نظر گرفته می‌شود؛ بنابراین K عامل در شبکه MEC وجود دارد و هر عامل به طور مستقل تصمیمات مشترک برون‌سپاری و تخصیص منابع را با استفاده از سیاست آموخته‌شده اتخاذ می‌کند. شکل ۳ جزئیات فرایند آموزش الگوریتم پیشنهادی را ارائه می‌دهد. در ابتدا شبکه بازیگر μ_k و شبکه‌های منتقد متمرکز $Q_{k,j}^{\mu}$ به ترتیب با پارامترهای تصادفی η_k و $\theta_{k,j}$ مقداردهی اولیه می‌شوند. سپس شبکه بازیگر هدف μ_k' و شبکه‌های منتقد هدف $Q_{k,j}^{\mu'}$ با استفاده از پارامترهای شبکه آنلاین متناظر خود مقداردهی اولیه می‌شوند. همچنین بافر PER، D، با مقدار تهی مقداردهی اولیه می‌شود. فرایند آموزش تا تعداد اپیزودهای از پیش تعیین شده، $EP_{\max}$ ادامه می‌یابد و هر اپیزود زمانی خاتمه می‌یابد که گام زمانی بزرگ‌تر از آستانه از پیش تعیین‌شده، T باشد. در هر گام زمانی، عامل k، مشاهدات محلی خود $o_k(t)$ را با تعامل با محیط به دست می‌آورد و با واردکردن مشاهدات محلی به شبکه بازیگر، اقدام $a_k(t)$ به دست می‌آید. برای بهبود اکتشاف الگوریتم از یک فرایند OU[3] [۵۶] استفاده می‌شود که یک نویز همبسته زمانی[4]، ΔN، به سیاست بازیگر اضافه می‌کند

$$a_k(t) = \mu_k(o_k(t)|\eta_k) + \Delta N \quad (۳۱)$$

پس از آن تمامی عامل‌ها اقدامات تعیین‌شده را اجرا می‌نمایند. به این ترتیب دستگاه‌های همراه برای نسبت تعیین‌شده برش زمانی با استفاده از مدل EH شارژ می‌شوند، نسبت وظیفه تعیین‌شده با توان انتقال مشخص‌شده به سرور MEC ارسال می‌گردد و قسمت باقیمانده از وظیفه به‌صورت محلی پردازش می‌شود. پس از انجام‌دادن اقدام، عامل یک پاداش فوری $r_k(o_k(t), a_k(t))$ را بر اساس (۲۵) دریافت می‌کند و حالت مشترک بعدی محیط یعنی $S(t+1)$ مشاهده می‌شود. پس از آن عامل‌ها تجربه حاصل از تعامل با محیط $(S(t), A(t), r(t), S(t+1))$ را در بافر PER ذخیره می‌کنند. در هر گام زمانی t، عامل یک mini-batch از داده‌ها را با اندازه H با استفاده از نمونه‌برداری اولویت‌بندی‌شده از بافر PER انتخاب می‌کند و شبکه‌های منتقد آنلاین را آموزش می‌دهد. پس از به‌روزرسانی شبکه‌های منتقد به تعداد مشخص d، پارامترهای شبکه بازیگر به‌روز می‌شوند. نهایتاً پارامترهای بازیگر هدف و منتقدان هدف با استفاده از به‌روزرسانی نرم به‌روز می‌شوند.

۵- ارزیابی عملکرد

در این بخش، عملکرد طرح برون‌سپاری پیشنهادی ارزیابی می‌شود. ما محیط شبکه را شبیه‌سازی کردیم و الگوریتم پیشنهادی را با استفاده از کتابخانه TensorFlow ۲ توسعه دادیم. در ادامه، ابتدا تنظیمات شبیه‌سازی توصیف می‌شوند و سپس عملکرد با طرح‌های برون‌سپاری مختلف از نظر هزینه محاسباتی بلندمدت، مصرف انرژی، تأخیر پردازش و نرخ شکست وظیفه مقایسه می‌گردد.

1. Independent and Identically Distributed
2. Overfitting
3. Ornstein-Uhlenbeck
4. Temporally-Correlated

۱: مقداردهی اولیه حداکثر گام زمانی در هر اپیزود T، حداکثر اپیزود $EP_{\max}$، فاکتور تخفیف γ، ضریب به‌روزرسانی نرم τ، اندازه mini-batch H، فرکانس به‌روزرسانی سیاست d.

۲: مقداردهی اولیه شبکه بازیگر آنلاین $\mu_k(o_k,\eta_k)$ و شبکه‌های منتقد متمرکز آنلاین $Q^{\mu}_{k,j}(S,A|\theta_{k,j})$ با استفاده از پارامترهای تصادفی η_k و $\theta_{k,j}$ برای $k\in K$ و $j\in\{1,2\}$.

۳: مقداردهی اولیه شبکه بازیگر هدف $\mu'_k(o_k|\eta'_k)$ و شبکه‌های منتقد متمرکز هدف $Q^{\mu'}_{k,j}(S,A|\theta'_{k,j})$ به ترتیب با $\eta'_k\leftarrow\eta_k$ و $\theta'_{k,j}\leftarrow\theta_{k,j}$.

۴: مقداردهی بافر PER، $D=\{\}$.

۵: **آغاز حلقه اول:** برای $episode=1$ تا $EP_{\max}$ انجام دهید:

۶: مشاهده حالت اولیه $o_k(1)$ توسط هر دستگاه همراه و دستیابی به حالت مشترک اولیه همه عامل‌ها $S(1)=\{o_1(1),\dots,o_K(1)\}$.

۷: **آغاز حلقه دوم:** برای هر گام زمانی $t=1$ تا T انجام دهید:

۸: انتخاب اقدام $a_k(t)=\mu_k(o_k(t)|\eta_k)+\Delta N$ بر اساس بازیگر آنلاین برای هر عامل k و افزودن نویز اکتشاف OU به آن.

۹: اجرای اقدامات همه عامل‌ها $A(t)=\{a_1(t),a_2(t),\dots,a_K(t)\}$.

۱۰: دریافت پاداش فوری $r_k(o_k(t),a_k(t))$ و حالت بعدی $S(t+1)=\{o_1(t+1),o_2(t+1),\dots,o_K(t+1)\}$ بر اساس (۲۵).

۱۱: ذخیره تاپل تجربه $(S(t),A(t),r(t),S(t+1))$ در D با بالاترین اولویت.

۱۲: $S(t)\leftarrow S(t+1)$.

۱۳: **آغاز حلقه سوم:** برای عامل $k=1$ تا K انجام دهید:

۱۴: **آغاز حلقه چهارم:** برای $h=1$ تا H انجام دهید:

۱۵: انتخاب تجربه h بر اساس احتمال انتخاب آن در (۳۰).

۱۶: محاسبه خطای TD، δ_h و به‌روزرسانی اولویت تجربه $pri_h\leftarrow|\delta_h|+\varepsilon$.

۱۷: محاسبه ارزش هدف y^h با استفاده از (۲۷).

۱۸: **پایان حلقه چهارم.**

۱۹: به‌روزرسانی شبکه‌های منتقد $Q^{\mu}_{k,j}(S,A|\theta_{k,j})$ با حداقل‌سازی تابع loss در (۲۶).

۲۰: **آغاز شرط اول:** اگر $t \bmod d$ است:

۲۱: به‌روزرسانی شبکه بازیگر $\mu_k(o_k|\eta_k)$ از طریق گرادیان سیاست قطعی با استفاده از (۲۹).

۲۲: به‌روزرسانی پارامترهای شبکه بازیگر و شبکه‌های منتقد با استفاده از به‌روزرسانی نرم: $\eta'_k\leftarrow\tau\eta_k+(1-\tau)\eta'_k$ و $\theta'_{k,j}\leftarrow\tau\theta_{k,j}+(1-\tau)\theta'_{k,j}$.

۲۳: **پایان شرط اول.**

۲۴: **پایان حلقه سوم.**

۲۵: **پایان حلقه دوم.**

۲۶: **پایان حلقه اول.**

شکل ۳: فرایند آموزش الگوریتم برون‌سپاری پیشنهادی.

جدول ۲: پارامترهای شبیه‌سازی.

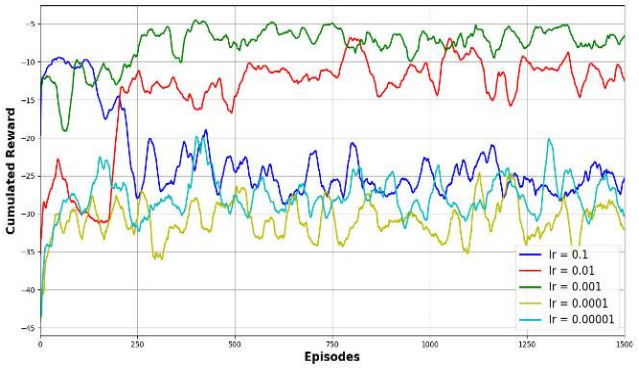
پارامتر	مقدار
اندازه داده ورودی وظیفه، $b_{k,j}$	توزیع پواسون، [۱-۷] Mbps [۴۰]
تعداد مورد نیاز چرخه‌های CPU برای	توزیع یکنواخت،
پردازش هر بیت وظیفه، $c_{k,j}$	[۱۶۸، ۳۶۰] cycles/bit [۳۵]
حداکثر تأخیر قابل تحمل وظیفه، $\tau^{\max}_{k,j}$	توزیع یکنواخت، [۵ -۱۵] ms
حداکثر ظرفیت باتری دستگاه همراه، $\overline{E_k}$	۵۰۰ mJ
حداکثر توان انتقال هر دستگاه همراه، $P^{\max}_k$	۲۰ dBm [۴۰]
پهنای باند کانال، B و بهره آنتن، A_g	۱ MHz و ۴/۱۱ [۳۵]
توان نویز، σ^2	-۱۷۵ dBm/Hz [۳۵]
فرکانس حامل، f_c و توان افت مسیر، l_e	۹۱۵ و ۲/۸ [۳۶]
فاکتور وزن در تابع هدف w	۰/۵

۵-۱ تنظیمات شبیه‌سازی

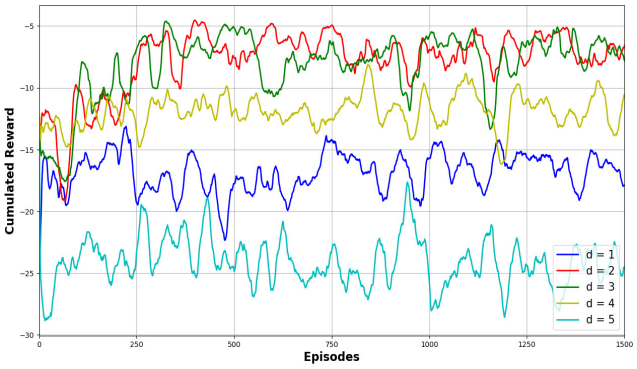
ما یک سناریوی زمان گسسته را در نظر خواهیم گرفت که در آن زمان به بخش‌های مساوی تقسیم گردیده و مدت زمان هر برش زمانی $\upsilon=10\,\text{ms}$ است. به طور پیش‌فرض، محیط شبکه MEC از یک AP مجهز به سرور MEC با $K=3$ دستگاه همراه تشکیل شده است. پارامترهای AP و گیرنده انرژی در هر دستگاه همراه به ترتیب مشابه Powercast TX91501-3W و Powerharvester P2110 [۵۷] در نظر گرفته شده‌اند که در آن، توان انتقال برای فرستنده انرژی در AP، $P=3$ وات و کارایی EH، $\mu=0/51$ است. موقعیت دستگاه‌های همراه به‌صورت پویا در برش‌های زمانی مختلف متفاوت است و فاصله هر دستگاه همراه از AP، $d_k(t)$، مشابه [۵۰] به‌صورت زنجیره مارکوف با $\Pr(d_k(t+1)=z|d_k(t)=z')=d_{zz'},\forall z,z'\in\{3,5,7,9,11\}\,m$ مدل‌سازی می‌شود. فرض می‌کنیم دستگاه‌های همراه از نظر قابلیت‌های محاسباتی، سطح باتری باقیمانده، اندازه داده‌های وظیفه، پیچیدگی محاسباتی وظیفه و حداکثر تأخیر قابل تحمل ناهمگن هستند. ظرفیت محاسباتی محلی هر دستگاه همراه، f^l_k، به طور تصادفی از مجموعه $\{0/8,0/9,1/0,1/1,1/2\}$GHz انتخاب شده است [۴۱]. علاوه بر این، منابع محاسباتی در دسترس سرور MEC در هر برش زمان، $f^s(t)$ به‌صورت زنجیره مارکوف با $\Pr(f^s(t+1)=z\,|f^s(t)=z')=f^s_{zz'}$ مدل شده [۴۱] که $\forall z,z'\in\{8,10,12,14,16\}$GHz می‌باشد. سایر پارامترهای شبیه‌سازی در جدول ۲ آمده‌اند.

تنظیمات فرایند آموزش: برای توسعه الگوریتم پیشنهادی، شبکه‌های بازیگر و منتقد متمرکز در هر عامل دستگاه همراه k یک DNN با ۴ لایه کاملاً متصل (FC)[1] یک لایه ورودی، دو لایه پنهان[2] و یک لایه خروجی هستند که در آن لایه‌های پنهان به ترتیب شامل ۲۵۶ و ۱۲۸

1. Fully Connected
2. Hidden Layer



شکل ۴: پاداش تجمیعی به‌دست‌آمده نسبت به نرخ یادگیری منتقد.



شکل ۵: پاداش تجمیعی نسبت به فرکانس به‌روزرسانی سیاست.

جدول ۳: خلاصه‌ای از پارامترهای آموزش.

پارامتر	مقدار
تعداد گام زمانی در هر اپیزود، T	۱۰۰
تعداد اپیزودهای آموزش، $EP_{\max}$	۱۵۰۰
نرخ به‌روزرسانی هدف، τ	۰٫۰۰۱
نرخ یادگیری شبکه بازیگر، α_a	۰٫۰۰۰۱
نرخ یادگیری شبکه‌های منتقد، α_c	۰٫۰۰۱
اندازه mini-batch، H	۶۴
اندازه بافر PER، N	۲۵۰۰۰۰
فاکتور تخفیف، γ	۰٫۹۹
فرکانس به‌روزرسانی بازیگر، d	۲

نورون هستند. خروجی یک بردار با اندازه ۳ است که آیتم‌های آن به ترتیب نسبت برون‌سپاری وظیفه $y_{k,j}(t)$، توان انتقال $p_{k,j}^{tran}(t)$ و نسبت برش زمانی برای برداشت انرژی $\alpha(t)$ را مشخص می‌کنند. هر لایه پنهان تابع فعال‌سازی ReLU[1] را اعمال می‌کند. علاوه بر این برای محدودکردن اقدامات در محدوده مورد نظر، تابع فعال‌سازی سیگموئید به لایه خروجی شبکه بازیگر اعمال می‌شود. بهینه‌ساز Adam[2] [۵۸] برای بهینه‌سازی تابع loss استفاده گردیده که در آن نرخ یادگیری برای شبکه‌های منتقد و بازیگر به ترتیب به ۰٫۰۰۱ و ۰٫۰۰۰۱ مقداردهی شده است. سایر پارامترهای فرایند آموزش در جدول ۳ لیست شده‌اند.

برای ارزیابی عملکرد الگوریتم پیشنهادی، آن را با سه طرح برون‌سپاری موجود مقایسه می‌کنیم. این طرح‌ها عبارتند از

- طرح برون‌سپاری مبتنی بر DQN [۳۱]: عامل DQN برای حل مسئله برون‌سپاری استفاده شده است. برای مدیریت فضای اقدام پیوسته، فضای اقدام گسسته[3] می‌شود. سطح گسسته‌سازی هر متغیر تصمیم‌گیری ۸ تنظیم شده است. از آنجا که فضای اقدام شامل ۳ متغیر تصمیم‌گیری است، 8^3 راه‌حل در فضای اقدام وجود دارد که عامل دستگاه همراه k می‌تواند از بین آنها انتخاب کند.
- طرح برون‌سپاری مبتنی بر DDPG [۳۹]: در این مقاله، طرح برون‌سپاری مبتنی بر چارچوب IL پیشنهاد شده که در آن هر دستگاه همراه تصمیمات را به طور مستقل با استفاده از یک عامل DDPG اتخاذ می‌کند.
- طرح برون‌سپاری مبتنی بر MADDPG [۴۱]: در این مقاله، طرح برون‌سپاری مبتنی بر چارچوب CTDE پیشنهاد شده که در آن هر دستگاه همراه از یک عامل MADDPG برای تصمیم‌گیری برون‌سپاری استفاده می‌کند.

برای مقایسه منصفانه، ساختار شبکه‌های عصبی در طرح‌های فوق مشابه ساختار شبکه‌های عصبی در الگوریتم پیشنهادی در نظر گرفته شده است. علاوه بر این، هایپرپارامترهای مشترک در این طرح‌ها به طور یکسان تنظیم شده است.

۵-۲ عملکرد همگرایی فرایند آموزش

۵-۲-۱ تنظیم هایپرپارامترهای الگوریتم پیشنهادی

در این بخش، تأثیر هایپرپارامترهای مختلف بر عملکرد همگرایی الگوریتم پیشنهادی بررسی شده و بهترین مقدار نرخ یادگیری برای شبکه منتقد و فرکانس به‌روزرسانی شبکه بازیگر را از طریق آزمایش به دست خواهیم آورد. برای نمایش بهتر، منحنی‌ها با استفاده از استراتژی هموارسازی[4] با پنجره کشویی[5] ۳۰ مطابق با [۴۸] رسم شده‌اند.

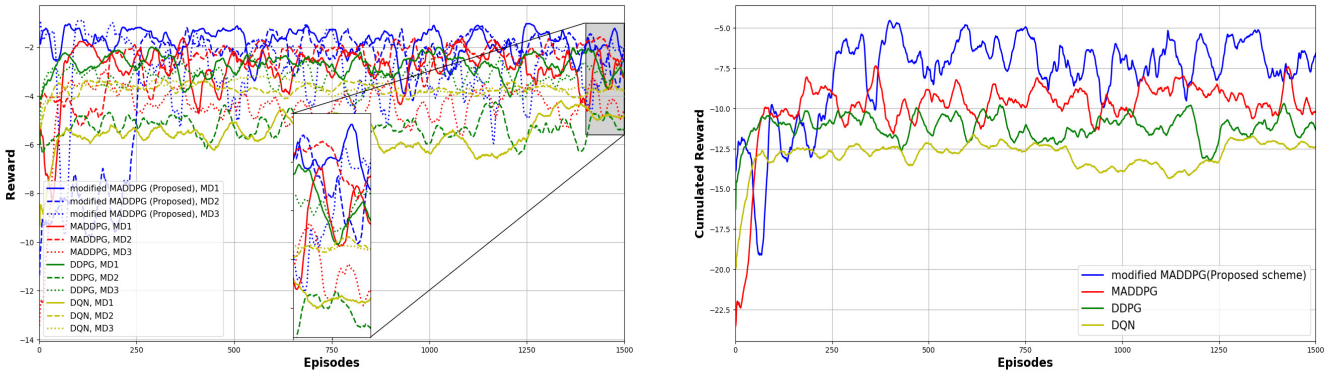
عملکرد همگرایی طرح برون‌سپاری پیشنهادی تحت نرخ‌های یادگیری مختلف شبکه منتقد در شکل ۴ بررسی شده است. همان طور که مشاهده می‌شود، یک نرخ یادگیری کوچک منجر به یادگیری آهسته عامل می‌شود؛ در حالی که نرخ یادگیری بزرگ باعث واگرایی عامل و گیرافتادن در بهینه محلی می‌گردد و از این رو نرخ یادگیری شبکه منتقد ۰٫۰۰۱ تنظیم شده است.

شکل ۵ تأثیر فرکانس به‌روزرسانی سیاست d را بر عملکرد همگرایی الگوریتم پیشنهادی بررسی می‌کند. هنگامی که $d=1$ است الگوریتم پیشنهادی در فرایند آموزش شیکه‌های منتقد و بازیگر مشابه با الگوریتم MADDPG عمل می‌کند که در آن فرکانس به‌روزرسانی شبکه‌های منتقد و بازیگر یکسان در نظر گرفته می‌شود. به این ترتیب، به ازای هر به‌روزرسانی برای شبکه منتقد، پارامتر شبکه بازیگر نیز به‌روز می‌شود. اگر چه d بزرگ‌تر منجر به منفعت بیشتر در رابطه با انباشته‌شدن خطاها می‌شود، با این حال مقدار بزرگ این هایپرپارامتر باعث می‌گردد که شبکه بازیگر در تکرارهای کمتر آموزش داده شود و از یادگیری مناسب سیاست جلوگیری می‌کند. مطابق با نتایج نشان‌داده‌شده در شکل ۵، بهترین عملکرد همگرایی با $d=2$ به‌دست آمده است.

۵-۲-۲ ارزیابی عملکرد همگرایی طرح‌های برون‌سپاری مختلف

عملکرد همگرایی طرح‌های برون‌سپاری مختلف در شکل ۶ نشان داده شده است. شکل ۶- الف منحنی‌های پاداش تجمیعی همه کاربران مربوط به عملکرد همگرایی فرایند آموزش طرح‌های مختلف برون‌سپاری را نشان

1. Activation Function
2. Adaptive Moment Estimation
3. Discrete
4. Smoothing Strategy
5. Sliding Window



(الف) (ب)

شکل ۶ عملکرد همگرایی طرح‌های برون‌سپاری مختلف، (الف) پاداش تجمیعی همه دستگاه‌های همراه و (ب) پاداش هر دستگاه همراه.

می‌دهد. همان طور که مشاهده می‌شود برای طرح‌های برون‌سپاری مختلف، پاداش تجمیعی هر اپیزود با تعامل عامل هر دستگاه همراه با محیط مربوطه‌اش افزایش می‌یابد و نهایتاً همه منحنی‌ها به مقادیر پاداش متفاوتی همگرا می‌شوند. با این حال واضح است که طرح برون‌سپاری پیشنهادی ما به همگرایی بهتری نسبت به سایر طرح‌ها از نظر پاداش تجمیعی همه دستگاه‌های همراه منجر می‌شود. در طرح برون‌سپاری الگوریتم پیشنهادی از ویژگی‌های clipped double Q-learning، به‌روزرسانی‌های با تأخیر سیاست و هموارسازی سیاست هدف برای اصلاح تخمین ارزش شبکه منتقد استفاده می‌کنیم. از آنجایی که شبکه بازیگر سیاست‌های خود را تحت هدایت شبکه منتقد یاد می‌گیرد، تخمین ارزش دقیق‌تر توسط منتقد منجر به یادگیری سیاست‌های بهتر و دستیابی به پاداش بالاتر توسط شبکه بازیگر می‌شود. علاوه بر این بر اساس نتایج به‌دست‌آمده، مشاهده می‌شود از آنجا که طرح‌های برون‌سپاری غیرمتمرکز مبتنی بر چارچوب CTDE مکانیسم همکاری را بین دستگاه‌های همراه مختلف در نظر می‌گیرند، بهتر از طرح‌های برون‌سپاری غیرمتمرکز مبتنی بر چارچوب IL عمل می‌کنند.

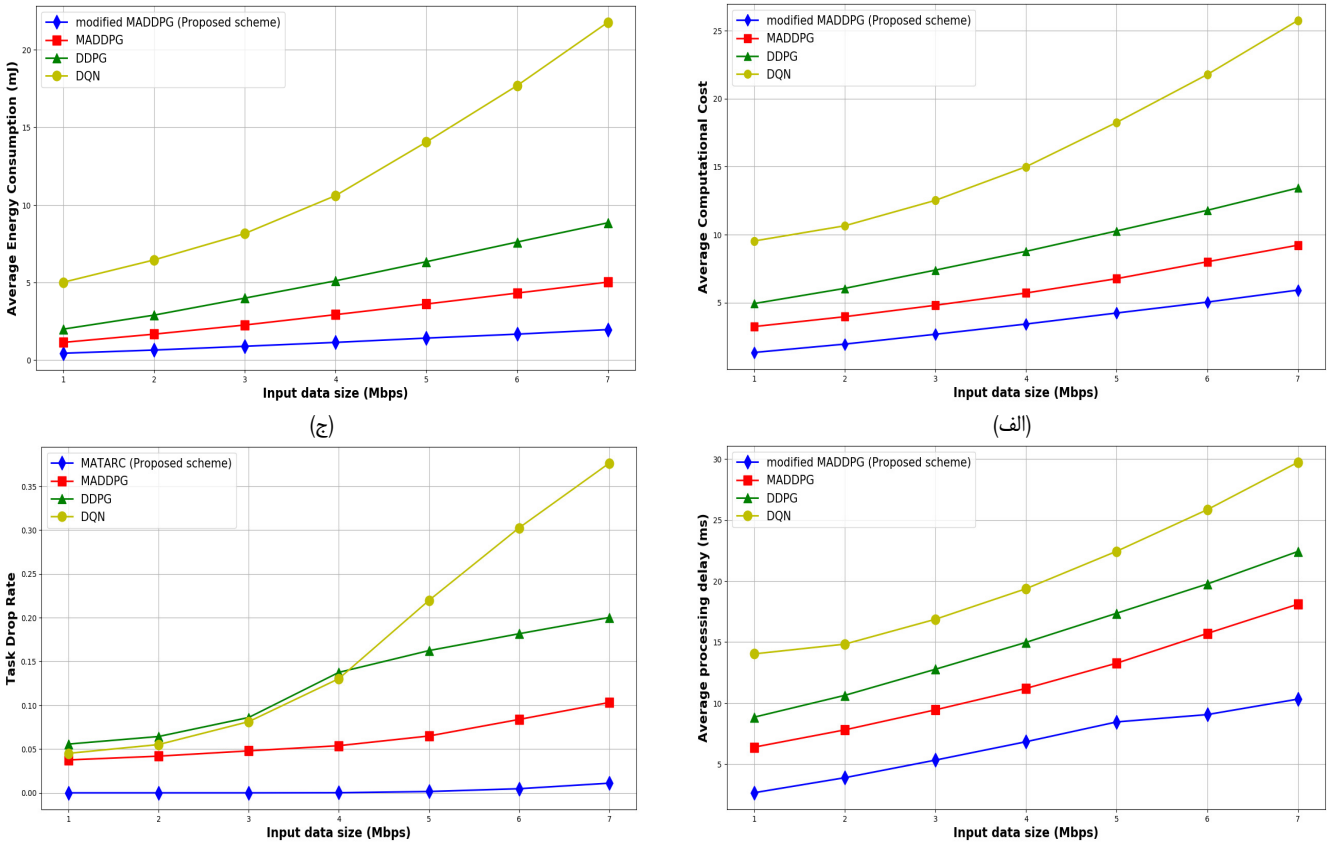
پاداش فردی هر دستگاه همراه برای طرح‌های برون‌سپاری مختلف در شکل ۶- ب نشان داده شده است. همان طور که مشاهده می‌شود، پاداش هر دستگاه در نهایت به یک مقدار پایدار همگرا می‌شود. با این حال طرح‌های برون‌سپاری مبتنی بر چارچوب CTDE از طرح‌های مبتنی بر چارچوب IL بهتر عمل می‌کنند؛ زیرا پاداش‌های دریافتی توسط دستگاه‌های مختلف در این طرح‌ها نزدیک‌تر است. این نشان می‌دهد که دستگاه‌های همراه در نهایت به یک نقطه تعادل می‌رسند و عدالت بین آنها تضمین می‌شود. با این حال همان طور که در شکل مشخص است، دستگاه همراه ۲ در طرح مبتنی بر DDPG و دستگاه همراه ۱ در طرح مبتنی بر DQN عملکرد بدتری نسبت به سایر دستگاه‌ها دارند که نتیجه عدم وجود مکانیسم همکاری بین دستگاه‌های همراه در این طرح‌ها است.

۵-۳ ارزیابی عملکرد کلی

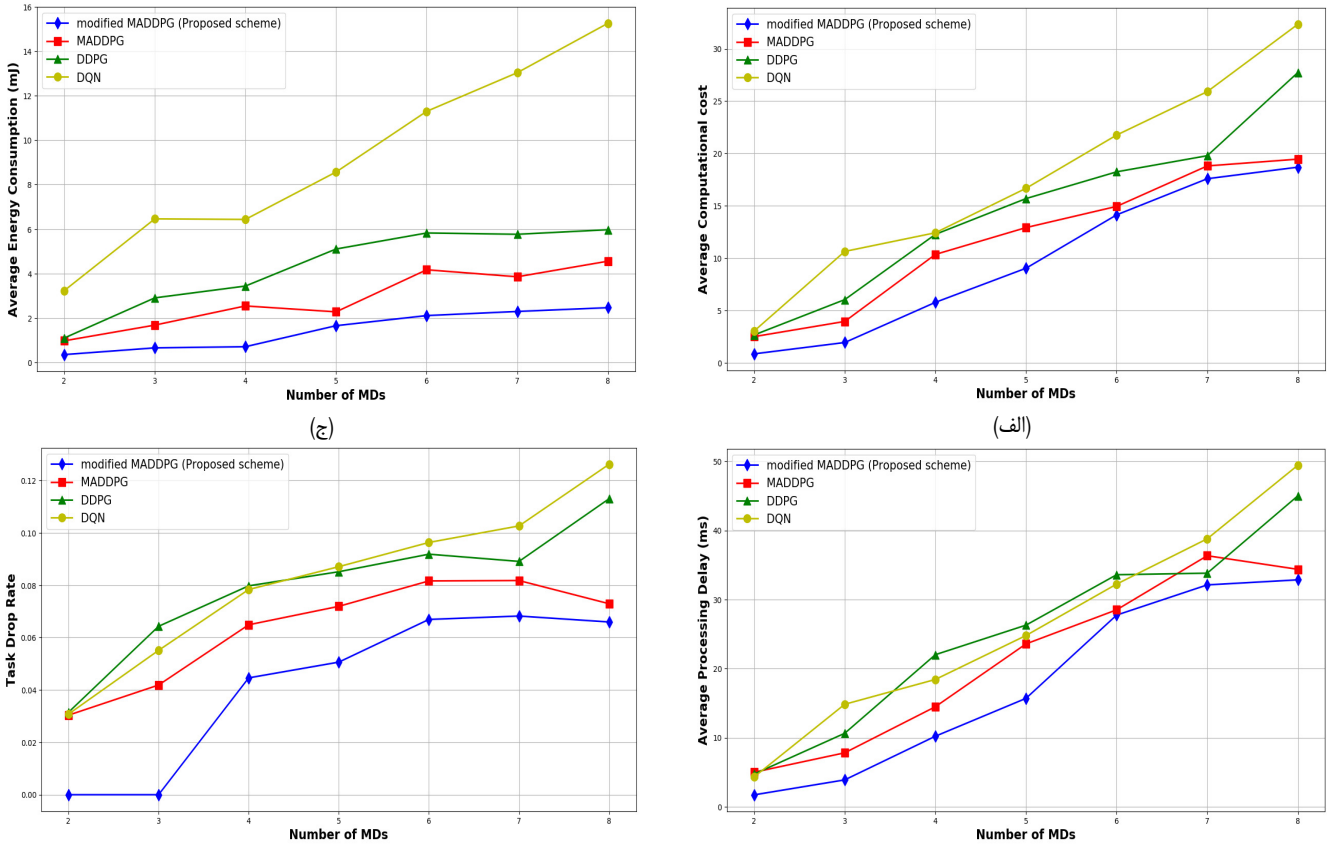
شایان ذکر است که پس از ۱۵۰۰ اپیزود آموزشی، پارامترهای آموخته‌شده ذخیره می‌شوند. در هنگام تست، سیاست‌های آموخته‌شده در عامل هر دستگاه همراه بارگذاری می‌شود. برای ارزیابی عملکرد طرح پیشنهادی، نتایج نشان‌داده‌شده در منحنی‌ها از میانگین روی ۱۰۰ اپیزود متوالی به دست آمده است.

ارزیابی عملکرد تحت اندازه مختلف وظیفه: میانگین عملکرد سیستم بر روی ۱۰۰ اپیزود با سه دستگاه همراه با اندازه وظیفه متغیر از ۱ مگابیت در ثانیه تا ۷ مگابیت در ثانیه برای طرح‌های برون‌سپاری مختلف در شکل ۷ نشان داده شده است. همان طور که مشاهده می‌شود با افزایش اندازه وظیفه، میانگین هزینه محاسباتی، میانگین مصرف انرژی و میانگین تأخیر پردازش همه طرح‌های برون‌سپاری افزایش می‌یابد. همان طور که قبلاً ذکر شد، هزینه محاسباتی از دو بخش تشکیل شده است: مصرف انرژی دستگاه همراه و تأخیر پردازش وظیفه. از آنجا که وظایف با اندازه داده‌های بزرگ‌تر به زمان پردازش و مصرف انرژی بیشتر نیاز دارند، این امر منجر به هزینه محاسباتی بالاتر می‌شود. همچنین افزایش مصرف انرژی دستگاه همراه و تأخیر پردازش وظیفه از برآورده‌شدن محدودیت‌های مربوط به انرژی و حداکثر تأخیر قابل تحمل وظیفه جلوگیری می‌کند که منجر به افزایش نرخ شکست وظیفه می‌شود. با این حال طرح برون‌سپاری مبتنی بر الگوریتم پیشنهادی بهتر از سایر طرح‌ها با کمترین هزینه محاسباتی، تأخیر پردازش وظیفه، مصرف انرژی و نرخ شکست وظیفه عمل می‌کند. از آنجا که ما برای بهبود عملکرد شبکه منتقد در الگوریتم پیشنهادی، clipped double q-learning، به‌روزرسانی با تأخیر سیاست و هموارسازی سیاست هدف را به کار گرفته‌ایم، این امر، بازیگر را به دستیابی به سیاست بهتر هدایت می‌کند. همچنین ما از روش PER برای انتخاب تجارب از بافر بازپخش در حین یادگیری استفاده می‌کنیم که باعث استفاده مؤثرتر از تجربیات ارزشمند می‌شود و فرایند یادگیری را کوتاه‌تر و پایدارتر می‌کند. طرح مبتنی بر MADDPG دومین طرح برتر است؛ زیرا از چارچوب CTDE برای یادگیری سیاست‌های برون‌سپاری غیرمتمرکز استفاده می‌کند که مشارکت دستگاه‌های همراه را در طول یادگیری سیاست‌ها در نظر می‌گیرد. واضح است که طرح مبتنی بر DQN دارای بدترین عملکرد است. این امر نشان می‌دهد طرح‌های برون‌سپاری مبتنی بر DQN در مسائل با فضای اقدام پیوسته نمی‌توانند به سیاست‌های مطلوب دست یابند.

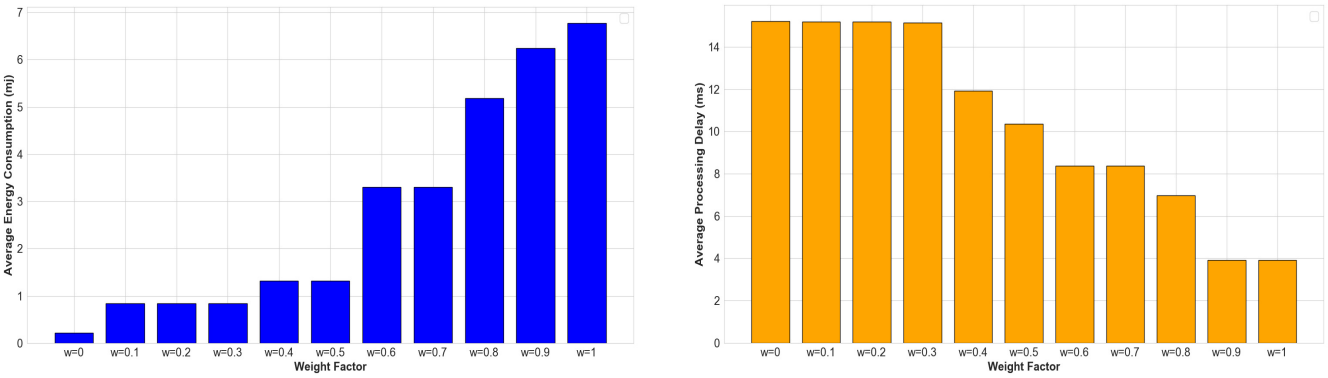
ارزیابی عملکرد تحت تعداد مختلف دستگاه‌های همراه: برای بررسی مقیاس‌پذیری سیستم، تأثیر افزایش تعداد دستگاه‌های همراه را بر روی طرح‌های مختلف ارزیابی می‌کنیم. ابتدا شبکه‌های عصبی طرح‌های مختلف برای تعداد معین k دستگاه همراه آموزش داده می‌شوند و سپس سیاست‌های آموخته‌شده ارزیابی می‌گردند. شکل ۸ میانگین هزینه محاسباتی، میانگین تأخیر پردازش وظیفه، میانگین مصرف انرژی و نرخ شکست وظیفه هر طرح را نشان می‌دهد؛ در حالی که تعداد دستگاه‌های همراه از ۲ تا ۸ متغیر است و اندازه وظیفه ۲ مگابیت بر ثانیه تنظیم شده است. می‌توان مشاهده کرد که با افزایش تعداد دستگاه‌ها، میانگین هزینه محاسباتی، میانگین تأخیر پردازش وظیفه، میانگین مصرف انرژی و نرخ شکست وظیفه همه طرح‌ها افزایش می‌یابد؛ اما با افزایش تعداد دستگاه‌ها شکاف عملکرد بین طرح‌های مبتنی بر چارچوب‌های CTDE و IL بیشتر



شکل ۷: عملکرد طرح‌های برون‌سپاری مختلف نسبت به اندازه داده وظیفه، (الف) میانگین هزینه محاسباتی نسبت به اندازه داده وظیفه، (ب) میانگین تأخیر پردازش نسبت به اندازه داده وظیفه، (ج) میانگین مصرف انرژی نسبت به اندازه داده وظیفه و (د) میانگین نرخ شکست وظیفه نسبت به اندازه داده وظیفه.



شکل ۸: عملکرد طرح‌های برون‌سپاری مختلف نسبت به تعداد دستگاه‌های همراه، (الف) میانگین هزینه محاسباتی نسبت به تعداد دستگاه‌های همراه، (ب) میانگین تأخیر پردازش نسبت به تعداد دستگاه‌های همراه، (ج) میانگین مصرف انرژی نسبت به تعداد دستگاه‌های همراه و (د) میانگین نرخ شکست وظیفه نسبت به تعداد دستگاه‌های همراه.



(الف) (ب)

شکل ۹: تأثیر فاکتور وزن بر میانگین تأخیر پردازش و میانگین مصرف انرژی، (الف) تأثیر فاکتور وزن بر میانگین تأخیر پردازش و (ب) تأثیر فاکتور وزن بر میانگین مصرف انرژی.

می‌شود. عامل‌های مبتنی بر چارچوب CTDE با به‌کارگیری یک شبکه منتقد متمرکز که از اطلاعات اضافی در مورد سیاست‌های سایر دستگاه‌ها در طول آموزش استفاده می‌کنند، می‌توانند رفتارهای مشارکتی را بیاموزند که باعث بهبود عملکرد سیستم می‌شود. علاوه بر این، طرح برون‌سپاری مبتنی بر الگوریتم پیشنهادی، عملکرد بهتری نسبت به طرح مبتنی بر MADDPG دارد و به این دلیل است که عملکرد الگوریتم MADDPG را با استفاده از ویژگی‌های clipped double Q-learning، هموارسازی سیاست هدف و PER بهبود بخشیده‌ایم که به بهبود تخمین تابع ارزش کمک می‌کند و منجر به یافتن سیاست‌های مطلوب‌تر می‌شود. همچنین استفاده از به‌روزرسانی با تأخیر شبکه بازیگر منجر به یادگیری یک سیاست باثبات‌تر و برتر می‌شود.

ما در شکل ۹ تأثیر فاکتور وزن را بر مصرف انرژی و تأخیر پردازش وظیفه بررسی می‌کنیم. مقادیر کوچک‌تر w باعث می‌شود که تمرکز عامل بیشتر بر کاهش مصرف انرژی باشد. همان طور که در شکل ۹ مشاهده می‌شود با کاهش مقادیر w سیاست آموخته‌شده در هر کاربر برای صرفه‌جویی در مصرف انرژی تلاش می‌کند که باعث تأخیر پردازش وظیفه طولانی‌تر می‌شود. بنابراین مطابق با نتایج به‌دست‌آمده با تخصیص مقادیر مختلف w می‌توان به‌راحتی موازنه بین تأخیر پردازش و مصرف انرژی را تنظیم کرد و به این معناست که طرح برون‌سپاری پیشنهادی با وظایف حساس به تأخیر و حساس به انرژی سازگار است.

٦- نتیجه‌گیری و کارهای آتی

در این مقاله به بهینه‌سازی مشترک مسئله برون‌سپاری محاسبات و تخصیص منابع غیرمتمرکز در MEC با دستگاه‌های همراه مجهز به قابلیت EH پرداخته شد. ما به‌طور خاص بر روی برون‌سپاری بخشی تمرکز کردیم که در آن هر وظیفه را می‌توان به دو زیروظیفه مستقل تقسیم کرد و به‌طور موازی توسط دستگاه همراه و سرور MEC پردازش کرد. ما برای پرداختن به این مسئله، یک الگوریتم برون‌سپاری مبتنی بر چارچوب CTDE برای کاهش هزینه محاسباتی بلندمدت سیستم از لحاظ تأخیر پردازش و مصرف انرژی دستگاه‌های همراه پیشنهاد دادیم. به‌طور خاص، الگوریتم پیشنهادی، نسخه‌ای بهبودیافته از الگوریتم MADDPG است که از clipped double Q-learning، به‌روزرسانی با تأخیر شبکه بازیگر و هموارسازی سیاست هدف برای حل مسئله بایاس تخمین بیش از حد در MADDPG و تخمین ارزش دقیق‌تر منتقد متمرکز استفاده می‌کند. همچنین تکنیک PER برای بهبود کارایی نمونه و سرعت یادگیری الگوریتم پیشنهادی اتخاذ شد.

نتایج شبیه‌سازی همگرایی سریع‌تر الگوریتم پیشنهادی را در مقایسه با MADDPG، DDPG و DQN نشان داد. علاوه بر این، نتایج آزمایش‌ها نشان داد که الگوریتم پیشنهادی نسبت به سایر طرح‌ها در سناریوهای مختلف با توجه به میانگین هزینه محاسباتی بلندمدت، میانگین مصرف انرژی، میانگین تأخیر پردازش و نرخ شکست وظیفه، عملکرد بهتری را به دست می‌آورد. به طور خاص، زمانی که اندازه داده وظیفه ۳ مگابیت در ثانیه تنظیم شده است، طرح پیشنهادی در مقایسه با MADDPG از لحاظ معیار هزینه محاسباتی ٪۴۴٫۴۶، از لحاظ معیار تأخیر پردازش ٪۴۳٫۶۴ و از لحاظ معیار مصرف انرژی ٪۶۰٫۱۷ عملکرد را بهبود می‌دهد.

ما در این کار، یک سیستم MEC تک‌سلول را فرض کردیم. در پژوهش‌های آتی، قصد داریم یک سناریوی چندسلولی با چندین سرور MEC را در نظر بگیریم که در آن دستگاه‌های همراه از یک سلول به سلول دیگر مهاجرت می‌کنند. علاوه بر این، یکی دیگر از جهت‌گیری‌های پژوهشی، تسریع فرایند یادگیری عامل‌ها با به‌کارگیری تکنیک یادگیری انتقالی خواهد بود.

مراجع

[1] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: a survey," *IEEE Internet of Things J.*, vol. 5, no. 1, pp. 450-465, Feb. 2018.

[2] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, and P. Mohapatra, "Edge cloud offloading algorithms: issues, methods, and perspectives," *ACM Computing Surveys*, vol. 52, no. 1, pp. 1-23, Feb. 2019.

[3] Q. H. Nguyen and F. Dressler, "A smartphone perspective on computation offloading-a survey," *Computer Communications*, vol. 159, pp. 133-154, Jun. 2020.

[4] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," *J. of Network and Computer Applications*, vol. 169, Article ID: 102781, Nov. 2020.

[5] P. Mach and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628-1656, Mar. 2017.

[6] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322-2358, Aug. 2017.

[7] X. Wang, *et al.*, "Wireless powered mobile edge computing networks: a survey," *ACM Computing Surveys*, vol. 55, no. 13s, Article ID: 263, 37 pp., Dec. 2023.

[8] U. M. Malik, M. A. Javed, S. Zeadally, and S. ul Islam, "Energy-efficient fog computing for 6G-enabled massive IoT: recent trends and future opportunities," *IEEE Internet of Things J.*, vol. 9, no. 16, pp. 14572-14594, Aug. 2022.

[9] Q. Luo, S. Hu, C. Li, G. Li, and W. Shi, "Resource scheduling in edge computing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2131-2165, Aug. 2021.

[10] Y. Fan, J. Ge, S. Zhang, J. Wu, and B. Luo, "Decentralized scheduling for concurrent tasks in mobile edge computing via deep

[31] C. Li, J. Xia, F. Liu, D. Li, L. Fan, G. K. Karagiannidis, and A. Nallanathan, "Dynamic offloading for multiuser muti-CAP MEC networks: a deep reinforcement learning approach," *IEEE Trans. on Vehicular Technology*, vol. 70, no. 3, pp. 2922-2927, Feb. 2021.

[32] L. Wang and G. Zhang, "Deep reinforcement learning based joint partial computation offloading and resource allocation in mobility-aware MEC system," *China Communications*, vol. 19, no. 8, pp. 85-99, Aug. 2022.

[33] J. Niu, S. Zhang, K. Chi, G. Shen, and W. Gao, "Deep learning for online computation offloading and resource allocation in NOMA," *Computer Networks*, vol. 216, Article ID: 109238, Oct. 2022.

[34] H. Lu, X. He, M. Du, X. Ruan, Y. Sun, and K. Wang, "Edge QoE: computation offloading with deep reinforcement learning for Internet of Things," *IEEE Internet of Things J.*, vol. 7, no. 10, pp. 9255-9265, Mar. 2020.

[35] V. D. Tuong, T. P. Truong, T. V. Nguyen, W. Noh, and S. Cho, "Partial computation offloading in NOMA-assisted mobile-edge computing systems using deep reinforcement learning," *IEEE Internet of Things J.*, vol. 8, no. 17, pp. 13196-13208, Mar. 2021.

[36] Z. Hu, J. Niu, T. Ren, B. Dai, Q. Li, M. Xu, and S. K. Das, "An efficient online computation offloading approach for large-scale mobile edge computing via deep reinforcement learning," *IEEE Trans. on Services Computing*, vol. 15, no. 2, pp. 669-683, Sept. 2021.

[37] J. Chen and Z. Wu, "Dynamic computation offloading with energy harvesting devices: a graph-based deep reinforcement learning approach," *IEEE Communications Letters*, vol. 25, no. 9, pp. 2968-2972, Jul. 2021.

[38] X. He, H. Lu, M. Du, Y. Mao, and K. Wang, "QoE-based task offloading with deep reinforcement learning in edge-enabled Internet of Vehicles," *IEEE Trans. on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2252-2261, Aug. 2020,

[39] Z. Chen and X. Wang, "Decentralized computation offloading for multi-user mobile edge computing: a deep reinforcement learning approach," *EURASIP J. on Wireless Communications and Networking*, vol. 2020, Article ID: 188, 21 pp., 2020.

[40] J. Chen, H. Xing, Z. Xiao, L. Xu, and T. Tao, "A DRL agent for jointly optimizing computation offloading and resource allocation in MEC," *IEEE Internet of Things J.*, vol. 8, no. 24, pp. 17508-17524, May 2021.

[41] Z. Cheng, M. Min, M. Liwang, L. Huang, and Z. Gao, "Multiagent DDPG-based joint task partitioning and power control in fog computing networks," *IEEE Internet of Things J.*, vol. 9, no. 1, pp. 104-116, Jun. 2021.

[42] Z. Chen, L. Zhang, Y. Pei, C. Jiang, and L. Yin, "NOMA-based multi-user mobile edge computation offloading via cooperative multi-agent deep reinforcement learning," *IEEE Trans. on Cognitive Communications and Networking*, vol. 8, no. 1, pp. 350-364, Jun. 2021.

[43] X. Huang, S. Leng, S. Maharjan, and Y. Zhang, "Multi-agent deep reinforcement learning for computation offloading and interference coordination in small cell networks," *IEEE Trans. on Vehicular Technology*, vol. 70, no. 9, pp. 9282-9293, Jul. 2021.

[44] N. Zhao, Z. Ye, Y. Pei, Y. C. Liang, and D. Niyato, "Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing," *IEEE Trans. on Wireless Communications*, vol. 21, no. 9, pp. 6949-6960, Mar. 2022.

[45] M. Chen, A. Guo, and C. Song, "Multi-agent deep reinforcement learning for collaborative task offloading in mobile edge computing networks," *Digital Signal Processing*, vol. 140, Article ID: 104127, Aug. 2023.

[46] Q. Tang, R. Xie, F. R. Yu, T. Huang, and Y. Liu, "Decentralized computation offloading in IoT fog computing system with energy harvesting: a Dec-POMDP approach," *IEEE Internet of Things J.*, vol. 7, no. 6, pp. 4898-4911, Feb. 2020.

[47] S. Zeng, X. Huang, and D. Li, "Joint communication and computation cooperation in wireless-powered mobile-edge computing networks with NOMA," *IEEE Internet of Things J.*, vol. 10, no. 11, pp. 9849-9862, Jan. 2023.

[48] L. Huang, S. Bi, and Y. J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. on Mobile Computing*, vol. 19, no. 11, pp. 2581-2593, Jul. 2019.

[49] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. on Wireless Communications*, vol. 17, no. 6, pp. 4177-4190, Apr. 2018.

[50] M. Min, *et al.*, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. on Vehicular Technology*, vol. 68, no. 2, pp. 1930-1941, Jan. 2019.

reinforcement learning," *IEEE Trans. on Mobile Computing*, vol. 23, no. 4, pp. 2765-2779, Apr. 2023.

[11] P. Gazori, D. Rahbari, and M. Nickray, "Saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach," *Future Generation Computer Systems*, vol. 110, pp. 1098-1115, Sept. 2020.

[12] H. Djigal, J. Xu, L. Liu, and Y. Zhang, "Machine and deep learning for resource allocation in multi-access edge computing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2449-2494, Aug. 2022.

[13] A. Feriani and E. Hossain, "Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: a tutorial," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1226-1252, Mar. 2021.

[14] T. Li, K. Zhu, N. C. Luong, D. Niyato, Q. Wu, Y. Zhang, and B. Chen, "Applications of multi-agent reinforcement learning in future internet: a comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1240-1279, Mar. 2022.

[15] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: a review of challenges, solutions, and applications," *IEEE Trans. on Cybernetics*, vol. 50, no. 9, pp. 3826-3839, Sept. 2020.

[16] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: a selective overview of theories and algorithms," In: Vamvoudakis, K.G., Wan, Y., Lewis, F.L., Cansever, D. (eds) *Handbook of Reinforcement Learning and Control. Studies in Systems, Decision and Control,* vol. 325, pp. 321-384, 2021.

[17] R. Lowe, *et al.*, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. 31st Conf. on Neural Information Processing Systems, NIPS'17,* 12 pp., Long Beach, CA, USA, 4-9 Dec. 2017.

[18] T. P. Lillicrap, *et al.*, *Continuous Control with Deep Reinforcement Learning*, arXiv preprint arXiv: 1509.02971, 2015.

[19] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. of the 35th Int. Conf. on Machine Learning, PMLR'80*, pp. 1587-1596, Stockholm Sweden, 10-15 Jul. 2018.

[20] O. K. Shahryari, H. Pedram, V. Khajehvand, and M. D. TakhtFooladi, "Energy and task completion time trade-off for task offloading in fog-enabled IoT networks," *Pervasive and Mobile Computing*, vol. 74, Article ID: 101395, Jul. 2021.

[21] J. Bi, H. Yuan, S. Duanmu, M. Zhou, and A. Abusorrah, "Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization," *IEEE Internet of Things J.*, vol. 8, no. 5, pp. 3774-3785, Sept. 2020.

[22] S. Fu, F. Zhou, and R. Q. Hu, "Resource allocation in a relay-aided mobile edge computing system," *IEEE Internet of Things J.*, vol. 9, no. 23, pp. 23659-23669, Jul. 2022.

[23] G. Yang, L. Hou, X. He, D. He, S. Chan, and M. Guizani, "Offloading time optimization via markov decision process in mobile-edge computing," *IEEE Internet of Things J.*, vol. 8, no. 4, pp. 2483-2493, Oct. 2020.

[24] B. Cao, L. Zhang, Y. Li, D. Feng, and W. Cao, "Intelligent offloading in multi-access edge computing: a state-of-the-art review and framework," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 56-62, Mar. 2019.

[25] Z. Liu, Y. Yang, K. Wang, Z. Shao, and J. Zhang, "POST: parallel offloading of splittable tasks in heterogeneous fog networks," *IEEE Internet of Things J.*, vol. 7, no. 4, pp. 3170-3183, Jan. 2020.

[26] M. Guo, Q. Li, Z. Peng, X. Liu, and D. Cui, "Energy harvesting computation offloading game towards minimizing delay for mobile edge computing," *Computer Networks*, vol. 204, Article ID: 108678, Feb. 2022.

[27] T. Zhang and W. Chen, "Computation offloading in heterogeneous mobile edge computing with energy harvesting," *IEEE Trans. on Green Communications and Networking*, vol. 5, no. 1, pp. 552-565, Jan. 2021.

[28] H. Teng, Z. Li, K. Cao, S. Long, S. Guo, and A. Liu, "Game theoretical task offloading for profit maximization in mobile edge computing," *IEEE Trans. on Mobile Computing*, vol. 22, no. 9, pp. 5313-5329, May 2022.

[29] H. Wu, Z. Zhang, C. Guan, K. Wolter, and M. Xu, "Collaborate edge and cloud computing with distributed deep learning for smart city Internet of Things," *IEEE Internet of Things J.*, vol. 7, no. 9, pp. 8099-8110, May 2020.

[30] L. Ale, N. Zhang, X. Fang, X. Chen, S. Wu, and L. Li, "Delay-aware and energy-efficient computation offloading in mobile-edge computing using deep reinforcement learning," *IEEE Trans. on Cognitive Communications and Networking*, vol. 7, no. 3, pp. 881-892, Mar. 2021.

[51] D. Silver *et al.*, "Deterministic policy gradient algorithms," in *Proc. of the 31st Int. Conf. on Machine Learning, PMLR'32*, pp. 387-395, Beijing, China, 22-24 Jun. 2014.
[52] F. Zhang, J. Li, and Z. Li, "A TD3-based multi-agent deep reinforcement learning method in mixed cooperation-competition environment," *Neurocomputing*, vol. 411, pp. 206-215, Oct. 2020.
[53] P. Sun, W. Zhou, and H. Li, "Attentive experience replay," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 34, no. 04, pp. 5900-5907, Apr. 2020.
[54] Y. Hou, L. Liu, Q. Wei, X. Xu, and C. Chen, "A novel DDPG method with prioritized experience replay," in *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics, SMC'17,* pp. 316-321, Banff, Canada, 5-8 Oct. 2017.
[55] T. Schaul, J. Quan, I. Antonoglou, and D. Silver*, Prioritized Experience Replay*, arXiv preprint arXiv:1511.05952, 2015.
[56] P. Cheridito, H. Kawaguchi, and M. Maejima, "Fractional ornstein-uhlenbeck processes," *Electron. J. Probab*, vol. 8, Article ID: 3, 14 pp., 2003.
[57] http://www.powercastco.com
[58] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv preprint arXiv:1412.6980, 2014.

آتوسا دقایقی در سال ۱۳۸۹ مدرک کارشناسی مهندسی فناوری اطلاعات خود را از دانشگاه آزاد اسلامی واحد تهران‌جنوب و در سال ۱۳۹۶ مدرک کارشناسی ارشد مهندسی فناوری اطلاعات گرایش تجارت الکترونیک خود را از دانشگاه آزاد اسلامی واحد تهران‌مرکز دریافت نموده است. وی در سال ۱۳۹۸ به دوره دکتری مهندسی فناوری اطلاعات در دانشگاه قم وارد گردید و هم‌اکنون به صورت تمام‌وقت مشغول به تحصیل است. زمینه‌های علمی مورد علاقه او عبارتند از سیستم‌های توزیعی، تخلیه بار محاسباتی و مدیریت منابع در محاسبات لبه موبایل، بهینه‌سازی و یادگیری تقویتی عمیق.

محسن نیک‌رأی در سال ۱۳۸۱ مدرک کارشناسی مهندسی کامپیوتر خود را از دانشگاه علم و صنعت ایران و مدرک کارشناسی ارشد و دکتری مهندسی کامپیوتر خود را در سال‌های ۱۳۸۵ و ۱۳۹۲ از دانشگاه تهران دریافت نموده است. دکتر نیک‌رأی از سال ۱۳۹۵ در گروه مهندسی کامپیوتر و فناوری اطلاعات دانشگاه قم به عنوان هیأت علمی مشغول به فعالیت است. زمینه‌های علمی مورد علاقه وی عبارتند از زمان‌بندی و مدیریت منابع در محیط ابر و مه.