Journal of Soft Computing and Information Technology (JSCIT)
Babol Noshirvani University of Technology, Babol, Iran
Journal Homepage: jscit.nit.ac.ir
Volume 13, Number 2, Summer 2024, pp. 55-68
Received: 03/02/2023, Revised: 09/03/2024, Accepted: 11/16/2024

# Performance-aware Partial Task Offloading in Multi-access Edge Computing

Mohsen Darchini-Tabrizi[1] and Reza Entezari-Maleki[2*]
1- School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.
2*- School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.
[1]mohsen_darchini@comp.iust.ac.ir, and [2*]entezari@iust.ac.ir

Corresponding author's address: Reza Entezari-Maleki, School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

**Abstract-** The proliferation of new applications has led to new challenges in energy consumption, task processing, and data storage. Multi-access Edge Computing (MEC) is a new computational paradigm that can transfer workloads from users' devices to powerful servers in the same location with the least possible time and energy overhead to improve the QoS and performance measures. Since joint task offloading and resource allocation in MEC is one of the main concerns of performance-aware applications, this paper explores a fine-grained view to this problem under the dynamic and time-varying conditions of the entire system. The main goal of this paper is to reduce the normalized cost of the system, which is the weighted sum of the completion time and the consumed energy, by formulating the problem and proposing a new algorithm based on the reinforcement learning approach. The results of the simulations performed under different scenarios corresponding to the real-world systems show the improvement in the completion time and energy consumption of the tasks compared to other existing methods and leads to an average reduction of 22 and 24 percent in the scenarios related to the evaluation of the normalized cost of the system.

**Keywords**- Task Offloading, Multi-access Edge Computing, Resource Allocation, Performance, Reinforcement Learning.

مجله علمی رایانش نرم و فناوری اطلاعات
دانشگاه صنعتی نوشیروانی بابل
صفحه مجله: jscit.nit.ac.ir
جلد ۱۳، شماره ۲، تابستان ۱۴۰۳، صفحه ۵۵-۶۸
دریافت: ۱۴۰۱/۱۲/۱۱، بازنگری: ۱۴۰۳/۰۶/۱۳، پذیرش: ۱۴۰۳/۰۸/۲۶

# ارائه‌ی یک روش بارسپاری جزئی آگاه از کارایی در محاسبات لبه‌ای دسترسی چندگانه

محسن دارچینی تبریزی[۱] و رضا انتظاری ملکی*[۲]
۱- دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران.
*۲- دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران.
[1]mohsen_darchini@comp.iust.ac.ir, [2*]entezari@iust.ac.ir

* نشانی نویسنده مسئول: رضا انتظاری ملکی، تهران، رسالت، خیابان هنگام، خیابان دانشگاه، دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر.

**چکیده- فراگیری کاربردهای نوظهور چالش‌های جدیدی را در زمینه‌های مصرف انرژی، پردازش محاسبات و ذخیره‌سازی داده‌ها ایجاد کرده است. محاسبات لبه‌ای دسترسی چندگانه به عنوان یک الگوی محاسباتی نوین که می‌تواند بارکاری دستگاه‌های کاربران را به سرورهای قدرتمند موجود در محل آنها با صرف کمترین زمان و انرژی ممکن منتقل کرده و از این طریق موجب بهبود کیفیت ارائه خدمات و کارایی اجرای وظایف شود، امیدواری‌هایی را ایجاد نموده است. با توجه به اینکه در نظر گرفتن امکان بارسپاری وظایف و تخصیص منابع به صورت توأمان در این الگوی محاسباتی یکی از اصلی‌ترین محورهای توجه برای کاربردهای آگاه از کارایی می‌باشد، در این مقاله به بررسی ریزدانه‌ی این مسئله در شرایط پویا و متغیر با زمان پرداخته شده است. هدف ایده‌ی پیشنهادی در این مقاله، کاهش هزینه‌ی نرمال‌شده‌ی سیستم ناشی از جمع وزن‌دار زمان تکمیل وظایف و میزان انرژی مصرفی آنها می‌باشد که این امر با ارائه‌ی راه‌حلی بر اساس یادگیری تقویتی بعد از فرموله‌سازی و مدل کردن مسئله صورت گرفته است. نتایج شبیه‌سازی‌های انجام شده تحت سناریوهای مختلف، بیانگر بهبود معیارهای زمان تکمیل وظایف و انرژی مصرفی آنها نسبت به سایر روش‌های موجود بوده که نهایتاً، به طور متوسط، منجر به کاهش ۲۲ و ۲۴ درصدی در سناریوهای ارزیابی هزینه‌ی نرمال‌شده‌ی سیستم شده است.**

**واژه‌های کلیدی:** بارسپاری وظایف، محاسبات لبه‌ای دسترسی چندگانه، تخصیص منابع، کارایی، یادگیری تقویتی.

## ۱- مقدمه

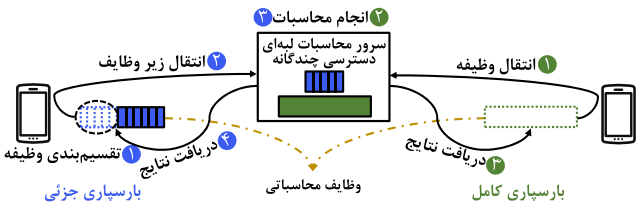
در سال‌های اخیر شاهد ظهور فناوری‌های نیازمند نرخ داده بالا همچون 5G و رشد انفجاری انواع تجهیزات سیار کاربران بوده‌ایم؛ به‌طوریکه بر اساس گزارش شرکت سیسکو[۱] تعداد تجهیزات سیار کاربران در سال ۲۰۲۳ به ۱۳.۱ میلیارد در سراسر جهان خواهد رسید [۱]. با توجه به این گزارش و مشاهده‌ی روند رو به رشد استفاده از برخی کاربردهای نوظهور مانند تشخیص چهره، تشخیص گفتار، واقعیت مجازی/افزوده، بازی‌های برخط تعاملی، خودروهای خودران و اینترنت اشیاء، امکان استفاده از الگوهای محاسباتی قدیمی‌تری نظیر محاسبات ابری و مِه به دلیل تأخیر پردازشی زیاد، عدم آگاهی کامل از تحرک کاربران، فاصله نسبتاً زیاد تا محل ایجاد درخواست‌های محاسبات پردازشی/ذخیره‌سازی و پهنای باند محدود به جهت پاسخگویی با کیفیت مناسب وجود ندارد [۲-۴]. شایان ذکر است که حتی اگر امکان انجام محاسبات مورد نظر نیز فراهم شود، هزینه‌های مالی و میزان منابع مصرفی بسیار زیاد محدودیت‌های دیگری را به وجود می‌آورد. از طرفی دیگر، پیشرفت‌های اخیر در حوزه‌ی طراحی و ساخت قطعات مدرن، راه را برای توسعه‌ی تعداد بسیار زیادی از برنامه‌های کاربردی در حوزه‌های مختلفی نظیر گردشگری، مراقبت‌های بهداشتی و سلامتی، ارائه‌ی خدمات عمومی و همچنین ایمنی

خیابان‌ها و ارتقای کیفیت رانندگی هموار نموده است [۵]. بهره‌گیری از ربات‌های هوشمند سخنگو برای تعامل با گردشگران و ارائه‌ی انواع خدمات لازم به آنها، استفاده از پهپادهای بدون سرنشین[۲] دارای دوربین برای پایش شرایط ترافیکی خیابان‌ها و میزان آلودگی هوای مناطق گوناگون و تعبیه انواع سنسورهای هوشمند در دستگاه‌های پوشیدنی یا تجهیزات محیطی نظیر ساعت‌ها و عینک‌های هوشمند یا کنتورهای مصرف برق، گاز و آب از جمله نمونه‌های شاخص آن می‌باشند [۷،۶].

با وجود پیشرفت‌های حاصل شده و تکامل روزافزون بخش‌های مختلف دنیای فناوری اما همچنان ممکن است در برخی کاربردهای حساس، امکان انجام محاسبات پردازشی/ذخیره‌سازی لازم خصوصاً در یک بازه‌ی زمانی بسیار محدود وجود نداشته باشد که این امر حتی با ازدیاد تعداد هسته‌های پردازشی، افزایش پهنای باند مصرفی، بهبود توان سرورهای محاسباتی یا افزایش ظرفیت حافظه‌های ذخیره‌سازی مورد استفاده آنها نیز قابل حل نباشد [۹،۸]. از این رو در پاسخ به چنین نیازمندی‌های جدیدی می‌توانیم از بارسپاری وظایف[۳] در محاسبات لبه‌ای دسترسی چندگانه (MEC[۴]) به عنوان راهکاری که می‌تواند به شیوه‌ای کارآمد این مسئله را حل نماید استفاده کنیم؛ چرا که با انتقال منابع پردازشی/ذخیره‌سازی به لبه‌ی شبکه و انجام وظایف بارسپاری‌شده در آن، هم سربار محاسباتی ناشی از تأخیر ارتباطی و انرژی مصرفی بالا در مقایسه با دیگر الگوهای محاسباتی نظیر محاسبات ابری و مِه وجود نخواهد داشت و هم احتیاج گسترده‌ای به پهنای باند بی‌سیم برای انتقال داده‌های لازم بین دستگاه‌های کاربران و سرورها نخواهد بود [۱۲-۱۰]. با توجه به اینکه حجم داده‌های نیازمند پردازش تجهیزات کاربران امروزی بسیار زیاد بوده و تعبیه تمامی توان محاسباتی مورد نیاز آنها در خود تجهیزات تقریباً غیر ممکن و در صورت انجام، بسیار پر هزینه خواهد بود، در نتیجه توسعه الگوی محاسبات لبه‌ای دسترسی چندگانه و سوق دادن پژوهشگران و فعالین صنعتی به استفاده از آن برای کاربردهای فوق‌الذکر با سرعت بیشتری در حال انجام می‌باشد. از این رو، بهره‌گیری از سرورهای راه‌دور در قالب روشی با عنوان بارسپاری وظایف برای انجام محاسبات پیچیده و حساس به تأخیر[۵] راهکار جذابی بوده که در سال‌های گذشته، کارآمدی‌اش در زمینه‌ی کاهش زمان تکمیل وظایف مختلف و مصرف انرژی تجهیزات کاربران در سیستم‌های مبتنی بر محاسبات لبه‌ای دسترسی چندگانه به اثبات رسیده است. افزایش کارایی‌پذیری و نرخ تکمیل وظایف به‌طوریکه در سناریوهای مختلف منجر به کاهش تأخیر، مصرف منابع یا هزینه‌های مالی شود از جمله دلایل اهمیت استفاده از این روش در محیط MEC می‌باشد [۱۴،۱۳].

در نظر گرفتن جمع وزن‌داری از چندین مورد مؤثر در قالب هزینه کلی سیستم می‌تواند علاوه بر جامعیت ارزیابی‌های انجام شده، باعث نزدیک‌سازی هرچه بیشتر به شرایط دنیای واقعی شود که با توجه به نوع کاربرد مورد استفاده و ساختار شبکه‌ی در دسترس می‌توان تعریف متفاوتی از هزینه‌ی کلی سیستم داشت. برای مثال با وجود اینکه اگر برنامه‌ای حساس به تأخیر باشد، بحث کاهش تأخیر در پاسخگویی و اگر ظرفیت باتری دستگاه‌های کاربران برای اجرای تمامی وظایف دارای محدودیت باشد، بحث افزایش طول عمر آنها یا کمینه‌سازی هرچه بیشتر انرژی مصرفی کاربران، مسئله‌ی اصلی می‌باشد اما باید توجه داشت که بهینه‌سازی جداگانه هر معیار، اثر نامناسبی روی سایر معیارهای ارزیابی سیستم نداشته باشد. تمامی موارد مؤثر در زمان تکمیل وظایف و میزان مصرف انرژی آنها در سه مرحله‌ی انتقال داده‌های مربوطه، انجام محاسبات لازم و بازگشت نتایج قابل دسته‌بندی می‌باشند که وضعیت کانال‌های ارتباطی مورد استفاده، امکان وجود تداخل، نویز و محوشدگی[۶] در محیط، توانایی محاسباتی سرورها، مقدار پهنای باند در دسترس و فاصله‌ی توپولوژیکی کاربران با سرور مد نظر، در بازده فرایند بارسپاری وظایف تأثیر بسزایی می‌گذارد [۱۶،۱۵].

مطابق با شکل ۱ برای انجام بارسپاری وظایف دو حالت کلی بارسپاری کامل[۷] و بارسپاری جزئی[۸] را می‌توان متصور بود. در بارسپاری کامل، کل یک وظیفه‌ای که تصمیم بر بارسپاری آن گرفته شده است را به سرور MEC و در بارسپاری جزئی فقط بخشی از آن وظیفه که نیازمندی بیشتری به منابع پردازشی/ذخیره‌سازی دارد را منتقل می‌نمائیم؛ در نتیجه، امکان کاهش تأخیر انتقال داده‌ها در ازای گسترده‌تر شدن فضای تصمیم‌گیری حاصل می‌شود [۱۷]. پس از ایجاد درخواست بارسپاری توسط دستگاه‌های کاربران، اقدام به ثبت مشخصات وظایف بارسپاری‌شونده، بررسی شرایط کانال ارتباطی و مقدار منابع لازم برای اجرای آنها در زمان واقعی می‌شود. در بارسپاری جزئی پس از این مرحله بر اساس معیارهای مختلف سیستم به تقسیم‌بندی یک وظیفه به دو بخش بارسپاری‌شونده و غیر آن پرداخته می‌شود تا در خصوص زمان و مکان مناسب بارسپاری هر




**شکل ۱ - انواع مختلف انجام بارسپاری وظایف**

کدام به صورت ریزدانه تصمیم‌گیری شود. پس از اتخاذ تصمیم بارسپاری برای هر قسمت از وظیفه، باید تخصیص مناسبی از منابع برای انجام آنها در حوزه‌های مربوط به انتقال و پردازش نیز صورت پذیرد تا هم انجام وظایف دچار کمبود منابع و کاهش نرخ اجرا نشود و هم هزینه‌ی اضافی ناشی از تأمین منابع بدون استفاده به وجود نیاید. با وجود اینکه بارسپاری وظایف می‌تواند به‌طور بالقوه باعث بهبودهایی در زمان تکمیل وظایف و میزان انرژی مصرفی آنها شود اما رسیدن بدین مقصود و به دنبال آن افزایش کیفیت خدمات[۹] ارائه‌شده سیستم، خصوصاً با گسترش مقیاس آن، علاوه بر نوع پیاده‌سازی خود فرایند بارسپاری وظایف، در گرو عوامل دیگری نظیر سازوکار تخصیص منبع[۱۰]، پشتیبانی از امکان تقسیم‌بندی وظایف و شرایط کانال ارتباطی می‌باشد [۲۱-۱۸].

شیوع استفاده از هوش مصنوعی در ابعاد مختلف زندگی بشر و بکار گیری روش‌های مختلف یادگیری ماشین برای دستیابی به راه‌حل‌های بهینه بدون داشتن هیچگونه دانش قبلی نسبت به پویایی‌های ناشناخته محیط‌ها از طریق تعامل مکرر با آنها می‌تواند منجر به بهبود کارایی سیستم شود. باید توجه داشت که بهره‌گیری از این روش‌ها با افزایش مقیاس سیستم مورد استفاده و گسترش حجم داده‌های آنها، همچنان کار پیچیده‌ای می‌باشد؛ از این رو برای بهره‌مندی کامل از قابلیت‌های این روش‌ها ابتدا باید به تعریف مسئله‌ی اصلی و جمع‌آوری داده‌های لازم پرداخته و سپس با فرموله‌سازی مسئله، تعیین متناسب پارامترهای مختلف و اعمال پردازش‌هایی بر روی داده‌های خام جمع‌آوری شده، اقدام به استخراج ویژگی‌های مد نظر از فضای راه‌حل شود [۲۴-۲۲].

در این مقاله برخلاف بسیاری از کارهای پیشین این حوزه، بر روی بهینه‌سازی مسئله‌ی بارسپاری جزئی وظایف و تخصیص توأمان منابع در یک سیستم MEC چند کاربره و متغیر با زمان تمرکز شده و تلاش‌هایی در راستای کمینه‌سازی جمع وزن‌دار نرمالی از دو پارامتر مدت زمان تکمیل وظایف ناهمگن کاربران سیستم و میزان انرژی مصرفی آنها صورت گرفته است. برای حل مسئله مذکور نیز پس از فرموله‌سازی و مدل‌سازی آن به ارائه‌ی راه‌حلی مبتنی بر یادگیری تقویتی[۱۱] اقدام شده است. اصلی‌ترین نوآوری‌های این مقاله به شرح زیر قابل بیان می‌باشد:

- ارائه‌ی مدلی از یک سیستم MEC چندکاربره و متغیر با زمان به منظور کمینه‌سازی جمع وزن‌دار نرمالی از زمان تکمیل وظایف کاربران و میزان انرژی مصرفی آنها.
- بهبود کارایی فرایند بارسپاری جزئی وظایف با اتخاذ تصمیم‌گیری‌های مناسب در خصوص تقسیم‌بندی وظایف و تخصیص توأمان منابع به آنها.
- حل مسئله‌ی تشریح‌شده با استفاده از رویکرد یادگیری تقویتی و ارائه‌ی الگوریتمی مبتنی بر روش Q-Learning.
- انجام شبیه‌سازی‌های گسترده و ارائه‌ی نمودارهایی از نتایج آن برای نمایش کارایی الگوریتم ارائه‌شده در مقایسه با سه روش پایه و یک کار پیشین پژوهشی انجام شده در این حوزه.

ساختار این مقاله بدین صورت است که پس از بیان مقدمه، به معرفی انواع روش‌های مورد استفاده در کارهای پیشین مرتبط پرداخته و بعد از انجام مروری جامع، اقدام به انجام مقایسه میان کارهای پیشین و پژوهش انجام شده در این مقاله می‌شود. در ادامه جزئیات مربوط به روش پیشنهادی در قالب بخش‌های مدل‌سازی و فرموله‌سازی مسئله تشریح شده و الگوریتم پیشنهادی برای حل آن نیز پس از توصیف، مورد واکاوی قرار گرفته است. پس از آن، نتایج به دست آمده از شبیه‌سازی‌های انجام شده بر اساس معیارهای معرفی‌شده مورد ارزیابی قرار گرفته و مقایسه‌های لازم انجام شده است. در نهایت مطالب بیان شده جمع‌بندی شده و چشم‌انداز آتی مسیر تحقیقاتی مربوطه ترسیم گردیده است.

## ۲- پیشینه پژوهش

پژوهش‌های پیشین حوزه‌ی بارسپاری وظایف، طیف گسترده‌ای از تحقیقات دانشگاهی و صنعتی را شامل می‌شود که بسته به محل و هدف مورد استفاده، از روش‌های مختلفی برای پیاده‌سازی آن استفاده شده است. در این بخش ابتدا به شرح مختصری از این روش‌ها پرداخته و سپس تعدادی از کارهای قبلی انجام شده که با کار پژوهشی این مقاله مشابهت ساختاری یا محتوایی دارند را مورد واکاوی قرار می‌دهیم. ضمناً در **جدول ۱** نیز مقایسه‌ای میان تعدادی از کارهای پیشین موجود در این حوزه و پژوهش صورت گرفته در این مقاله انجام شده است.

### ۲-۱- دسته‌بندی کارهای مرتبط

با توجه به روند دنیای فناوری و بر اساس بررسی کارهای پیشین این حوزه می‌توان تمامی آنها را به دو بخش کلی غیر مبتنی بر هوش مصنوعی و مبتنی بر هوش مصنوعی دسته‌بندی کرد. از جمله روش‌های مربوط به دسته‌بندی اول می‌توان به نظریه بازی[۱۲]، بهینه‌سازی لیاپانوف[۱۳] و انواع رویکردهای مبتنی بر برنامه‌ریزی[۱۴] اشاره کرد. از طرف دیگر کارهای مربوط به دسته‌بندی دوم می‌توانند از انواع روش‌های مربوط به یادگیری ماشین نظیر یادگیری عمیق، یادگیری تقویتی و سایر روش‌های ترکیبی استفاده نمایند.

روش‌های مبتنی بر نظریه بازی با تجزیه و تحلیل رفتار کاربران

سعی در ارائه‌ی راه‌حل‌هایی دارند که همه‌ی کاربران از آن راضی باشند که اصطلاحاً به این حالت، تعادل نَش[15] گفته می‌شود. استفاده از چنین راهکارهایی برای مدل‌سازی تعاملات میان کاربران مختلف که ممکن است اهداف متضادی داشته باشند، عموماً به منظور اتخاذ تصمیمات توزیع‌شده در محیط می‌باشد. در روش‌های مبتنی بر بهینه‌سازی لیاپانوف از طریق کنترل انواع صف‌های موجود در سیستم، سعی در فراهم‌سازی شرایطی برای پایداری سیستم‌های بهره‌مند از آن می‌شود. بر این اساس اقداماتی در جهت کمینه‌سازی همزمان مقادیر توابع جریمه[16] و انحراف[17] که تابع جریمه آن عمدتاً شامل مقدار تأخیر یا انرژی مصرفی بوده و تابع انحراف عمدتاً برای صفوف اجرایی وظایف یا برداشت انرژی لازم آنها می‌باشد، انجام می‌شود. انواع روش‌های مربوط به برنامه‌ریزی نظیر برنامه‌ریزی خطی/غیرخطی، محدب، معین/نیمه معین و پویا در جهت یافتن پاسخ‌هایی بهینه برای مسائلی که تعداد محدودیت‌ها و متغیرهای تصمیم‌گیری زیادی دارند، مورد استفاده قرار گرفته و نتیجه‌ی آن فراهم‌سازی شرایطی برای دستیابی به بالاترین کارایی ممکن با کمترین هزینه ممکن می‌باشد. اساس چنین روش‌هایی، مدل‌سازی بر مبنای مفاهیم ریاضیاتی و یافتن مطلوب‌ترین مقادیر از بین جواب‌های امکان‌پذیر آن می‌باشد [۲۵].

هدف اصلی روش‌های مبتنی بر یادگیری ماشین، توسعه‌ی الگوریتم‌هایی است که بتوانند از داده‌های پیشین یک محیط و تجربه برآمده از آن در جهت بهبود کارایی سیستم استفاده نمایند. از این رو بازتولید هوش انسانی از طریق بهره‌گیری از یادگیری، استدلال و خود اصلاحی الگوریتم‌ها می‌تواند شانس دستیابی به هدف مذکور را افزایش دهد [۲۷،۲۶]. در روش‌های مبتنی بر یادگیری عمیق به مدل‌هایی که از لایه‌های پردازشی متعددی تشکیل شده‌اند، این اجازه داده می‌شود که با نمایش داده‌ها در سطوح انتزاعی متعدد، استخراج دانش مد نظر از داده‌های پیچیده به سر انجام برسد. با اینکه استفاده از این فرایند باعث ساده‌تر شدن فرایند یادگیری می‌شود اما عملکرد آن وابستگی زیادی به منابع در دسترس و محدودیت‌های محاسباتی کاربران آن دارد [۲۹،۲۸]. روش‌های مبتنی بر شبکه‌های عصبی با استفاده از تعدادی لایه پنهان، یک لایه ورودی و یک لایه خروجی اقدام به استخراج مرزهای مربوط به تصمیم‌گیری‌های پیچیده و پیشبینی خروجی‌های با ارزش‌تر برای آنها می‌کنند. ضمناً در روش‌های مبتنی بر یادگیری تقویتی، سعی در بیشینه‌سازی پاداش کاربران از طریق تعامل با تمامی عوامل مؤثر در محیط می‌شود. در اینگونه روش‌ها هر عامل موجود در محیط با توجه به سیاست یادگیری و پایگاه دانش خود، اقدام به انجام بهترین کنش ممکن در جهت افزایش کارایی سیستم می‌نماید که نمونه‌هایی از آن را در بخش بعدی مورد بررسی قرار می‌دهیم [۳۰،۲۹].

### ۲-۲- مقایسه کارهای مرتبط

پیرو توضیحات مطرح‌شده‌ی قبلی در ادامه‌ی این بخش به واکاوی تعدادی از کارهای پیشین مشابه حوزه این مقاله پرداخته می‌شود. در [۳۱] به ارائه‌ی مدلی از یک سیستم MEC چندکاربره پرداخته شده که با در نظر گرفتن مهلت زمانی انجام وظایف و محدودیت‌های مربوط به مصرف منابع، بهینه‌سازی‌هایی در خصوص مسئله‌ی بارسپاری کامل وظایف و تخصیص توأمان منابع انجام شده تا انرژی مصرفی تمامی کاربران موجود در سیستم از طریق حل یک مسئله برنامه‌ریزی غیرخطی صحیح مختلط و ارائه‌ی یک الگوریتم یادگیری تقویتی مبتنی بر Q-Learning کمینه‌سازی شود. در [۳۲] بر اساس یک مجموعه‌ی داده واقعی به بررسی وضعیت پویای مکان کاربران سیار و کانال ارتباطی اقدام می‌شود؛ سپس از طریق حل یک مسئله‌ی برنامه‌ریزی غیرخطی صحیح مختلط برای مسئله‌ی بارسپاری وظایف همراه با تخصیص توأمان منابع، به ارائه‌ی یک الگوریتم توزیع‌شده دارای پیچیدگی چندجمله‌ای مبتنی بر بهینه‌سازی مارکوف پرداخته می‌شود. در [۳۳] علاوه بر تحرک کاربران، حداکثر تأخیر قابل قبول برای اجرای وظایف که متشکل از جمع زمان‌های مربوط به تصمیم‌گیری، انتقال، تخصیص منابع و انجام محاسبات می‌باشد، راه‌حلی در راستای ایجاد مصالحه بین هزینه‌های سیستم ناشی از تخصیص منابع و تأمین الزامات لازم برای انجام کارآمد محاسبات ارائه شده و نهایتاً یک روش مشارکتی مبتنی بر مکان کاربران در خصوص بارسپاری کامل وظایف تک کاربره در این نوع شبکه‌ها ارائه می‌گردد. در [۳۴] به ارائه‌ی یک الگوریتم برخط و مبتنی بر یادگیری تقویتی عمیق[18] برای بارسپاری وظایف پرداخته شده که با توجه به وضعیت کانال ارتباطی و میزان پیچیدگی محاسباتی وظایف، اقدام به تخصیص منابع موجود در سیستم با استفاده از یک شبکه عصبی عمیق می‌نماید. در [۳۵] به بررسی چالش‌های بارسپاری وظایف و تخصیص منابع در یک شبکه‌ی اینترنت اشیای چند سروره که از تحرک و برداشت انرژی پشتیبانی می‌کند، پرداخته شده و در راستای کمینه‌سازی میانگین هزینه بلندمدت سیستم برای ارائه‌ی خدمات به دستگاه‌های سیار اینترنت اشیاء، به یافتن بهترین مکان برای بارسپاری وظایف چند کاربره با هدف بیشینه‌سازی نرخ اجرای وظایف و کمینه‌سازی میزان مصرف انرژی آنها اقدام می‌شود. در [۳۷] با هدف ایجاد مصالحه بین تأخیر اجرا و میزان انرژی مصرفی وظایف و همچنین اتخاذ تصمیم‌گیری‌های

بهینه در مورد فرایندهای مختلف انجام بارسپاری وظایف، همزمان با تخصیص توأمان منابع در یک سیستم MEC، به ارائه‌ی یک روش مبتنی بر الگوریتم A3C پرداخته شده که با در نظر گرفتن تأثیر متغیر بودن وضعیت سیستم با گذر زمان، توان محاسباتی سرورهای موجود در محیط و امکان وقوع پدیده‌ی محو شدگی در کانال بی‌سیم، تلاشی در جهت نزدیک‌سازی هر چه بیشتر مدل ارائه‌شده به دنیای واقعی صورت گرفته است؛ ضمناً با بهره‌گیری از ساختار شبکه نقاد کم‌پیچیدگی[۱۹] و ذخیره‌ی همه‌ی حالت-کنش‌ها در یک حافظه بافر بازپخش[۲۰]، فضای راه‌حل نیز در راستای بهبود کنش‌های در دسترس هر حالت محدود شده‌اند. در [۳۸] با

**جدول ۱ مقایسه‌ای میان کارهای پیشین انجام شده و پژوهش این مقاله**

| مرجع | نوع بارسپاری | | تعداد کاربران | | مبتنی بر شکاف زمانی | تخصیص توأمان منابع | پویایی ناشی از تغییر زمان | هدف بهینه‌سازی | قیود در نظر گرفته شده | اساس راه‌حل ارائه‌شده |
|---|---|---|---|---|---|---|---|---|---|---|
| | کامل | جزئی | تک کاربره | چند کاربره | | | | | | |
| این مقاله | | ✓ | | ✓ | ✓ | ✓ | ✓ | هزینه‌ی کلی ناشی از جمع وزن‌دار نرمالی از انرژی مصرفی و زمان انجام وظایف | مهلت زمانی اجرای وظایف<br>میزان انرژی مصرفی<br>مصرف بهینه منابع<br>تأخیر اجرای وظایف<br>الزامات تقسیم‌بندی وظایف | یک روش مبتنی بر Q-Learning |
| [۳۱] | ✓ | | | ✓ | ✓ | ✓ | ✓ | بهره‌وری مصرف انرژی در انجام فرایند بارسپاری | مهلت زمانی اجرای وظایف<br>میزان انرژی مصرفی<br>مصرف بهینه منابع | یک روش مبتنی بر Q-Learning |
| [۳۲] | | ✓ | | ✓ | | ✓ | | هزینه کلی ناشی از مصرف انرژی و تأخیر انجام محاسبات وظایف | مهلت زمانی اجرای وظایف<br>میزان انرژی مصرفی<br>مصرف بهینه منابع<br>الزامات تقسیم‌بندی وظایف | یک روش بارسپاری وظایف مبتنی بر برنامه‌ریزی غیرخطی صحیح مختلط |
| [۳۳] | ✓ | | ✓ | | ✓ | ✓ | | هزینه ایجاد مصالحه بین منابع مصرفی و تأخیر اجرای وظایف | مهلت زمانی اجرای وظایف<br>مصرف بهینه منابع | یک روش بارسپاری کامل و مشارکتی مبتنی بر مکان |
| [۳۴] | ✓ | | | ✓ | | ✓ | ✓ | کاهش پیچیدگی و افزایش توأمان نرخ اجرا محاسبات | محدودیت انرژی کاربران<br>مصرف بهینه منابع | یک روش بارسپاری مبتنی بر یادگیری تقویتی عمیق |
| [۳۵] | | ✓ | | ✓ | | ✓ | | میانگین وزن‌داری از هزینه بلندمدت ایجاد مصالحه بین کیفیت خدمات و مصرف انرژی | مهلت زمانی اجرای وظایف<br>مصرف بهینه منابع<br>میزان انرژی مصرفی<br>الزامات تقسیم‌بندی وظایف | یک روش مبتنی بر بهینه‌سازی لیاپانوف و برنامه‌ریزی نیمه معین |
| [۳۶] | ✓ | | | ✓ | ✓ | ✓ | ✓ | نرخ تکمیل وظایف با در نظر گرفتن انرژی مصرفی و مهلت زمانی اجرای آنها | میزان انرژی مصرفی<br>مهلت زمانی اجرای وظایف<br>مصرف بهینه منابع | یک روش بارسپاری مبتنی بر یادگیری تقویتی عمیق |
| [۳۷] | ✓ | | ✓ | | | ✓ | ✓ | هزینه ایجاد مصالحه بین انرژی مصرف و تأخیر اجرای وظایف | مهلت زمانی اجرای وظایف<br>مصرف بهینه منابع | یک رویکرد مبتنی بر یادگیری تقویتی عمیق و الگوریتم A3C |
| [۳۸] | | ✓ | | ✓ | ✓ | ✓ | | کیفیت خدمات و تأخیر اجرای وظایف | تداخل میان کاربران<br>مصرف بهینه منابع<br>الزامات تقسیم‌بندی وظایف | یک روش بارسپاری وظایف مبتنی بر تکرار |
| [۳۹] | ✓ | | | ✓ | ✓ | | | بیشینه سازی نرخ تکمیل وظایف | زمان تکمیل نهایی وظایف<br>تأخیر اجرای وظایف | یک الگوریتم مربوط به تصمیم‌گیری بارسپاری مبتنی بر زمانبندی اولویت‌دار |
| [۴۰] | ✓ | | ✓ | | ✓ | ✓ | | تأخیر اجرای وظایف و میزان انرژی مصرفی | مهلت زمانی اجرای وظایف<br>ظرفیت باتری دستگاه‌ها<br>مصرف بهینه منابع | روش‌های ابتکاری و مبتنی بر الگوریتم ژنتیک برای بارسپاری دستگاه به دستگاه |
| [۴۱] | ✓ | | | ✓ | ✓ | | | مجموع پاداش‌های بلندمدت حاصل از اجرای بارسپاری‌شده‌ی وظایف در حوزه‌ی برداشت انرژی | میزان انرژی مصرفی<br>تأخیر اجرای وظایف | یک الگوریتم بارسپاری وظایف برخط مبتنی بر بهینه‌سازی لیاپانوف |
| [۴۲] | ✓ | | | ✓ | ✓ | ✓ | ✓ | بهره‌وری مصرف انرژی در انجام فرایند بارسپاری | ظرفیت باتری دستگاه‌ها<br>مهلت زمانی اجرای وظایف<br>مصرف بهینه منابع | یک روش بارسپاری مبتنی بر یادگیری تقویتی عمیق |
| [۴۳] | ✓ | | | ✓ | ✓ | | ✓ | بهره‌وری مصرف انرژی در انجام فرایند بارسپاری | میزان انرژی مصرفی<br>مهلت زمانی اجرای وظایف | یک روش بارسپاری وظایف مبتنی بر برنامه‌ریزی صحیح خطی |
| [۴۴] | | ✓ | ✓ | | | | ✓ | مجموع وزن‌داری از انرژی مصرفی و تأخیر اجرای وظایف | تأخیر اجرای وظایف<br>میزان انرژی مصرفی<br>الزامات تقسیم‌بندی وظایف | یک روش مبتنی بر یادگیری عمیق |
| [۴۵] | ✓ | | | ✓ | ✓ | ✓ | ✓ | سربار تصمیم‌گیری‌های ناشی از انجام بارسپاری | میزان انرژی مصرفی<br>مهلت زمانی اجرای وظایف<br>وضعیت کانال ارتباطی | یک روش بارسپاری وظایف برخط مبتنی بر استفاده از یک شبکه عصبی عمیق مقیاس‌پذیر |

استفاده از الگوریتم شاخه و حد[۲۱] و فرموله‌سازی مسئله بارسپاری وظایف در قالب یک مسئله‌ی برنامه‌ریزی خطی صحیح مختلط، اقدام به اعمال بهینه‌سازی‌هایی در زمینه‌های تأخیر اجرایی وظایف و کیفیت ارائه‌ی خدمات به کاربران می‌شود تا شرایط مناسبی برای اجرای با کیفیت برنامه‌های حساس به تأخیر بر روی تجهیزات سیار کاربران از طریق بهره‌گیری از یک الگوریتم تکرارشونده که قادر به اتخاذ تصمیم‌گیری‌های پویا در زمان واقعی باشد، ایجاد گردد. در [۳۹] با استفاده از تکنیک‌های شکل‌دهی پرتو همدوسی[۲۲] و لغو تداخل متوالی[۲۳] به ارائه‌ی الگوریتمی برای تصمیم‌گیری در خصوص انتخاب محل مناسب در بارسپاری کامل وظایف چند کاربره پرداخته شده و از طریق بهره‌گیری توأمان از یک الگوریتم زمانبندی وظایف اولویت‌دار سعی در بیشینه‌سازی نرخ تکمیل وظایف و کاهش میزان تأخیر آنها شده است. در [۴۰] با در نظر گرفتن محدودیت‌هایی بر روی مقدار منابع محاسباتی لازم، نوع وظایف، میزان انرژی مصرفی، مهلت زمانی وظایف، الگوی تحرک کاربران سیار و امکان بارسپاری دستگاه به دستگاه[۲۴]، تلاش‌هایی در جهت فراهم‌سازی شرایطی برای تسریع اجرای وظایف و امکانی برای اتخاذ تصمیم‌گیری‌های پویا، آگاه از تحرک و توزیع‌شده از طریق ارائه‌ی روش‌هایی ابتکاری و مبتنی بر الگوریتم ژنتیک برای بهبود تأخیر اجرایی وظایف و میزان انرژی مصرفی آنها شده است. در [۴۱] با هدف بیشینه‌سازی پاداش تجمعی حاصل از انجام وظایف و بهبود کارایی فرایند بارسپاری کامل از طریق استفاده از یک ایستگاه پایه به عنوان توزیع‌کننده[۲۵] بارکاری برای دریافت و پخش متوازن وظایف بین سرورهای موجود در یک سیستم $MEC$، الگوریتمی برخط و مبتنی بر بهینه‌سازی لیاپانوف ارائه می‌شود تا با توجه به محدودیت‌های در نظر گرفته شده از قبیل ستاره‌ای بودن توپولوژی شبکه و فاصله‌ی فقط یک گام هر دستگاه سیار با توزیع‌کننده، بهینه‌سازی مد نظر نویسندگان صورت پذیرد. در [۴۲] به ارائه‌ی یک الگوریتم برخط، توزیع‌شده و متغیر با زمان مبتنی بر $DRL$ که از تکنیک دسترسی چندگانه غیر متعامد[۲۶] برای فراهم‌سازی امکانی برای همزمانی درخواست‌ها، و از ساختار بهینه‌سازی غیر محدب برای فرموله‌سازی مسئله‌ی بارسپاری وظایف چند کاربره همراه با تخصیص توأمان منابع در سیستم‌های $MEC$ استفاده کرده است، پرداخته می‌شود. در [۴۳] الگوریتمی مبتنی بر روش برنامه‌ریزی صحیح خطی برای بهینه‌سازی چالش بارسپاری پویا و چند کاربره‌ی وظایف به چند سرور $MEC$ ارائه شده که برای تصمیم‌گیری در خصوص اجرای محلی یا بارسپاری‌شده‌ی وظایف، یک تابع هزینه‌ای با تمرکز بر روی بهبود کیفیت تجربه کاربران سیار توسط نویسندگان آن تعریف می‌شود. در [۴۴] یک الگوریتم بارسپاری وظایف چند کاربره مبتنی بر یادگیری عمیق معرفی شده که به کمینه‌سازی میزان انرژی مصرفی و تأخیر اجرایی آنها می‌پردازد؛ این الگوریتم با استفاده از یک مجموعه داده آموزشی واقعی برای اعمال بهینه‌سازی در تابع هزینه‌ی متشکل از میزان مصرف منابع، کیفیت ارائه‌ی خدمات و تأخیرهای ناشی از انتقال، تقسیم‌بندی و اجرای وظایف، سعی در افزایش سرعت تصمیم‌گیری فرایندهای مربوط به آنها را دارند. در [۴۵] به اتخاذ تصمیمات زمان واقعی همراه با در نظر گرفتن شرایط متغیر کانال ارتباطی در خصوص نحوه‌ی اجرای بارسپاری‌شده‌ی وظایف بر اساس خروجی یک الگوریتم یادگیری برخط پرداخته شده و به منظور دستیابی به یک راه‌حل مقیاس‌پذیر برای مسئله، چارچوبی مبتنی بر یادگیری تقویتی عمیق که در پیاده‌سازی آن از یک شبکه عصبی عمیق و دانش تجمعی برآمده از آن بهره برده شده، ارائه شده است.

## ۳- روش پیشنهادی

به منظور توضیح روش پیشنهادی، ابتدا مدل‌سازی بخش‌های مختلف مؤثر در محیط انجام شده و سپس فرموله‌سازی آن صورت گرفته است. در نهایت، الگوریتم پیشنهادی برای حل مسئله مذکور ارائه شده است.

### ۳-۱- مدل‌سازی بخش‌های مختلف مسئله

از جمله بخش‌های مؤثر در محیط مسئله می‌توان به شبکه، ارتباطات و محاسبات اشاره نمود. لازم به ذکر است که کلیات مدل‌سازی‌های صورت گرفته بر مبنای مرجع [۳۱] بوده و مقداردهی پارامترها منطبق بر دیگر کارهای پیشین بررسی شده در این مقاله می‌باشد. ضمناً در پیاده‌سازی بخش‌های مختلف، نگاه جامع‌تری به جزئیات فرایند بارسپاری وظایف و تخصیص منابع به آنها وجود دارد که در ادامه به توضیح آنها خواهیم پرداخت.

#### ۳-۱-۱- مدل شبکه

برای مدل‌سازی ساختار شبکه $MEC$ در روش پیشنهادی، یک سرور و $n$ کاربر لبه‌ای دسترسی چندگانه ناهمگن ($n = \{1,2, \dots, N\}$) که از نظر توان پردازش محلی و نوع وظایف ایجاد شونده توسط آنها کاملاً تصادفی و به صورت متغیر با گذر زمان در سیستم حضور دارند را مطابق با شکل ۲ در نظر می‌گیریم. ضمناً به منظور بررسی ریزدانه‌ی مسئله، امکان تقسیم‌بندی وظایف به $m$ زیر وظیفه می‌باشد ($m = \{1,2, \dots, M\}$) که فرایند تقسیم‌بندی آن با استفاده از یک شیوه‌ی پویای زمان واقعی برای هر وظیفه‌ی ایجادشده در سیستم صورت

می‌پذیرد. لازم به ذکر است که علت در نظر گرفتن یک سرور، مطالعه و بررسی تأثیر استفاده از بارسپاری جزئی در بهبود کارایی اجرای وظایف مختلف کاربران سیستم می‌باشد؛ چرا که در غیر اینصورت امکان انجام پردازش‌های مربوط به بخش‌های مختلف یک وظیفه (زیروظایف بارسپاری‌شده) توسط چندین سرور ناهمگن از نظر توان پردازشی/ذخیره‌سازی به وجود می‌آمد که باعث دخیل شدن تعدادی از سایر عوامل اثرگذار مانند مباحث مربوط به مهاجرت وظایف، زمانبندی وظایف و غیره در ارزیابی کارایی سیستم می‌شد.




**شکل ۲ - مدل ساختار شبکه MEC چند کاربره**

### ۳-۱-۲- مدل ارتباطات

در صورتی که امکان اجرای محلی با کیفیت یک زیروظیفه بر اساس محدودیت‌های زمانی و یا مصرف انرژی وجود نداشته باشد، نیاز به ارسال آن زیروظیفه به سرور MEC موجود در سیستم ایجاد خواهد شد. با توجه به وجود یک سرور MEC، امکان وقوع همپوشانی در محدوده‌ی سرورها وجود نداشته ولی امکان تداخل میان چند کاربر به علت همزمانی ارسال درخواست‌هایشان به سرور، با استفاده از یک مکانیزم پویا برای دسترسی به منابع مرتفع می‌شود. نرخ انتقال داده قابل دستیابی برای هر کاربر از طریق رابطه (۱) که مبتنی بر رابطه‌ی شانون [۴۷،۴۶] می‌باشد در سیستم اعمال می‌شود. فهرستی از نمادهای به کار رفته در این مقاله و توضیحی از مفاهیم آنها در جدول ۲ گردآوری شده است.

$$r_n^t = \frac{W}{K}\ log_2(1+\frac{P_n g_n}{\frac{W}{K}N_0}) \qquad (۱)$$

### ۳-۱-۳- مدل محاسبات

همانطور که در بخش‌های قبلی اشاره شد تمامی زیروظایف($\varphi_{nm}^t$) که با چهار پارامتر اصلی ($K_{nm}^t, S_{nm}^t, \tau_{nm}^t, \rho_{nm}^t$) تعریف می‌شوند، یا به صورت محلی یا به صورت بارسپاری‌شده اجرا می‌شوند ($\delta_{nm}^t \in \{0, 1\}$). در این میان، نقش میزان محاسبات لازم برای

**جدول ۲ نمادهای به کار رفته در این مقاله و مفاهیم آنها**

| نماد | مفهوم | نماد | مفهوم |
|---|---|---|---|
| $K$ | تعداد کاربران نیازمند بارسپاری | $T_{nm,tr}^t$ | زمان انتقال داده‌های هر $\varphi_{nm}^t$ در اجرای بارسپاری‌شده |
| $P_n$ | توان مصرفی کاربر $n$ برای انتقال داده | $T_{nm,pr}^t$ | زمان پردازش داده‌های هر $\varphi_{nm}^t$ در اجرای بارسپاری‌شده |
| $r_n^t$ | نرخ انتقال داده قابل دستیابی برای کاربر $n$ | $f_{nm}^t$ | ظرفیت محاسباتی تخصیص‌یافته از سرور MEC برای اجرای بارسپاری‌شده $\varphi_{nm}^t$ |
| $\varphi_{nm}^t$ | زیروظیفه $m$ از وظیفه کاربر $n$ | $K_{dl}^t$ | مقدار داده‌های مربوط به نتایج اجرای هر $\varphi_{nm}^t$ |
| $K_{nm}^t$ | مقدار محاسبات لازم برای هر $\varphi_{nm}^t$ | $T_{nm,dl}^t$ | زمان دریافت نتایج حاصل از داده‌های پردازش شده هر $\varphi_{nm}^t$ در اجرای بارسپاری‌شده |
| $S_{nm}^t$ | تعداد چرخه‌های پردازشی لازم برای تکمیل هر $\varphi_{nm}^t$ | $E_{n,off}^t$ | انرژی مصرفی کلی در اجرای بارسپاری‌شده هر وظیفه کاربر $n$ |
| $\tau_{nm}^t$ | بیشینه زمان اجرای قابل تحمل برای اجرای $\varphi_{nm}^t$ | $E_{nm,tr}^t$ | انرژی مصرفی برای انتقال داده‌های هر $\varphi_{nm}^t$ در کاربر $n$ در اجرای بارسپاری‌شده |
| $\rho_{nm}^t$ | بیشینه انرژی مصرفی قابل تحمل برای اجرای $\varphi_{nm}^t$ | $E_{nm,MEC}^t$ | انرژی مصرفی برای پردازش داده‌های هر $\varphi_{nm}^t$ در اجرای بارسپاری‌شده |
| $\delta_{nm}^t$ | تصمیم بارسپاری هر $\varphi_{nm}^t$ | $P_n^{MEC}$ | توان مصرفی لازم برای تکمیل زیروظایف در سرور MEC |
| $T_{n,loc}^t$ | زمان اجرای محلی زیروظایف در کاربر $n$ | $E_{n,idle}^t$ | انرژی مصرفی کاربر $n$ در حالت آماده به کار حین اجرای بارسپاری‌شده زیروظایف |
| $f_n^{loc}$ | ظرفیت محاسباتی محلی هر کاربر $n$ | $P_n^w$ | توان مصرفی کاربر $n$ در حالت آماده به کار |
| $E_{n,loc}^t$ | کل انرژی مصرفی برای اجرای محلی زیروظایف در کاربر $n$ | $C_{total}, ct$ | هزینه‌ی کلی سیستم |
| $\xi_n^{loc}$ | مقدار انرژی مصرفی به ازای اجرای هر CPU cycle برای تکمیل هر زیروظیفه در کاربر $n$ | $ar$ | مقدار منابع در دسترس سرور MEC در هر حالت سیستم |
| $C_{n,loc}^t$ | هزینه‌ی کلی سیستم در اجرای محلی | $D_o(t)$ | وکتور مربوط به تمامی تصمیم‌های بارسپاری |
| $C_{n,off}^t$ | هزینه‌ی کلی سیستم در اجرای بارسپاری‌شده | $R_a(t)$ | وکتور مربوط به تمامی منابع موجود در سرور MEC |
| $T_{n,off}^t$ | زمان کلی اجرای بارسپاری‌شده برای هر وظیفه کاربر $n$ | $L$ | فاصله بین کاربر و سرور MEC |

اجرای زیروظایف، میزان توان پردازشی محلی و محدودیت‌های اجرایی وظایف در حوزه‌های زمان و انرژی بسیار پر اهمیت می‌باشد.

با توجه به توضیحات قبلی، اگر برای اجرای یک وظیفه تصمیم بر اجرای محلی باشد، برای محاسبه زمان تکمیل اجرای محلی یک وظیفه از کاربر $n$ و میزان مصرف انرژی آن خواهیم داشت:

$$T_{n,loc}^t = \sum_{m=1}^{M} \frac{S_{nm}^t}{f_n^{loc}} \qquad (۲)$$

$$E_{n,loc}^t = \sum_{m=1}^{M} \xi_n^{loc} S_{nm}^t \qquad (۳)$$

که مقدار $\xi_n^{loc}$ با توجه به آزمایش‌های عملی انجام‌شده توسط کارهای پژوهشی قبلی برابر با $10^{-27}(f_n^{loc})^2$ در نظر گرفته شده [۴۸] و مقدار $f_n^{loc}$ نیز با توجه به توانایی پردازشی محلی هر کاربر متفاوت است. با دو مقدار $T_{n,loc}^t$ و $E_{n,loc}^t$ می‌توان هزینه‌ی کلی سیستم برای اجرای محلی را به صورت زیر محاسبه نمود:

$$C_{n,loc}^{t} = I_n^t T_{n,loc}^{t} + I_n^e E_{n,loc}^{t} \quad (0 \le I_n^t, I_n^e \le 1 \text{ و } I_n^t + I_n^e = 1) \qquad (۴)$$

حال اگر برای اجرای یک وظیفه تصمیم بر انجام بارسپاری باشد، با توجه به اینکه اجرای بارسپاری‌شده نیازمند طی سه مرحله انتقال داده‌های مربوطه، انجام محاسبات لازم و بازگشت نتایج می‌باشد، نیاز به فرموله کردن هر کدام از این سه بخش وجود خواهد داشت. البته بر خلاف کارهای قبلی انجام شده در این حوزه که بخش مربوط به انرژی مصرفی در حالت آماده به کار بودن دستگاه‌های کاربران و زمان لازم برای بازگشت نتایج به آنها را به دلیل کوچک بودن مقادیر مربوط به آنها در نظر نگرفتند، در این مقاله دو بخش مذکور نیز در نظر گرفته شده است. در نتیجه برای اجرای بارسپاری‌شده خواهیم داشت:

$$\begin{aligned} T_{n,off}^{t} &= \sum_{m=1}^{M} [(T_{nm,tr}^{t}) + (T_{nm,pr}^{t}) + (T_{nm,dl}^{t})] \\ &= \sum_{m=1}^{M} [(\frac{K_{nm}^{t}}{r_n^t}) + (\frac{S_{nm}^{t}}{f_{nm}^t}) + (\frac{K_{dl}^{t}}{r_n^t})] \end{aligned} \qquad (۵)$$

$$\begin{aligned} E_{n,off}^{t} &= \sum_{m=1}^{M} [(E_{nm,tr}^{t}) + (E_{nm,MEC}^{t}) + (E_{n,idle}^{t})] \\ &= \sum_{m=1}^{M} \left[ \left( P_n T_{nm,tr}^{t} = \frac{P_n K_{nm}^{t}}{r_n^t} \right) \right. \\ &\quad + \left( P_n^{MEC} T_{nm,pr}^{t} = \frac{P_n^{MEC} S_{nm}^{t}}{f_{nm}^t} \right) \\ &\quad \left. + \left( P_n^{w} T_{nm,pr}^{t} = \frac{P_n^{w} S_{nm}^{t}}{f_{nm}^t} \right) \right] \end{aligned} \qquad (۶)$$

در رابطه‌ی (۵) $T_{nm,tr}^{t}$ و $T_{nm,dl}^{t}$ به ترتیب بیانگر زمان انتقال داده‌های هر زیروظیفه و دریافت نتایج حاصل از پردازش آنها در اجرای بارسپاری‌شده بوده که مقادیرشان از تقسیم مقدار محاسبات لازم برای اجرای زیروظیفه و داده‌های نتایج، به نرخ انتقال داده قابل دستیابی برای کاربر ایجاد کننده‌ی آن زیروظیفه به دست می‌آید. مقدار $T_{nm,pr}^{t}$ که بیانگر زمان پردازش داده‌های هر زیروظیفه در اجرای بارسپاری‌شده‌ی آن می‌باشد از طریق تقسیم تعداد چرخه‌های پردازشی لازم برای تکمیل آنها به ظرفیت محاسباتی تخصیص‌یافته از سرور MEC قابل محاسبه می‌باشد. توضیحات بیان شده برای رابطه‌ی (۶) نیز صادق بوده با این تفاوت که در هر بخش، مقادیر به دست آمده باید در توان مصرفی مربوطه ضرب شده تا میزان انرژی مصرفی آن بخش محاسبه شود. در نتیجه مشابه با رابطه‌ی (۴) و مقادیر محاسبه شده برای $T_{n,off}^{t}$ و $E_{n,off}^{t}$ که از روابط (۵) و (۶) به دست آمدند، می‌توانیم به محاسبه‌ی هزینه‌ی کلی سیستم برای اجرای بارسپاری‌شده‌ی وظایف بپردازیم:

$$C_{n,off}^{t} = I_n^t T_{n,off}^{t} + I_n^e E_{n,off}^{t} \quad (0 \le I_n^t, I_n^e \le 1 \text{ و } I_n^t + I_n^e = 1) \qquad (۷)$$

حال برای محاسبه‌ی هزینه‌ی کلی سیستم که متشکل از جمع وزن‌دار زمان تکمیل وظایف و میزان انرژی مصرفی آنها در حالت‌های اجرای محلی و بارسپاری‌شده می‌باشد می‌توان از نتایج دو رابطه‌ی (۴) و (۷) استفاده کرد. ضمناً مقدار $\delta_{nm}^{t}$ برای اجرای محلی برابر با ۰ و برای اجرای بارسپاری‌شده برابر با ۱ در نظر گرفته می‌شود که در این صورت خواهیم داشت:

$$C_{total} = \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} (1 - \delta_{nm}^{t}) C_{n,loc}^{t} + \delta_{nm}^{t} C_{n,off}^{t} \qquad (۸)$$

با توجه به روابط و ساختار مدل‌سازی انجام شده می‌توان به شیوه‌ای مشابه با رابطه‌ی (۸)، ارزیابی مجزایی از زمان تکمیل وظایف و انرژی مصرفی آنها از طریق رابطه‌های (۹) و (۱۰) داشت:

$$T_{total} = \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} (1 - \delta_{nm}^{t}) T_{n,loc}^{t} + \delta_{nm}^{t} T_{n,off}^{t} \qquad (۹)$$

$$E_{total} = \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} (1 - \delta_{nm}^{t}) E_{n,loc}^{t} + \delta_{nm}^{t} E_{n,off}^{t} \qquad (۱۰)$$

### ۳-۲- فرموله‌سازی مسئله

در این بخش به توضیح نحوه‌ی فرموله کردن مسئله و معرفی هدف بهینه‌سازی می‌پردازیم. بر این اساس رابطه نهایی به دست آمده در بخش قبل را به عنوان فرمول بهینه‌سازی اصلی مسئله معرفی کرده و محدودیت‌های زیر را برای آن تعریف می‌نمائیم.

$$\min_{D_o(t), R_a(t)} \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} [(1 - \delta_{nm}^{t}) C_{n,loc}^{t} + \delta_{nm}^{t} C_{n,off}^{t}] \qquad (۱۱)$$

C1: $\delta_{nm}^{t} \in \{0,1\},\ \forall n, m, t \in N, M, T$

C2: $(1 - \delta_{nm}^{t}) T_{n,loc}^{t} + \delta_{nm}^{t} T_{n,off}^{t} \le \min_{T_{total}} (\tau_{nm}^{t}, T_{n,loc}^{t}),\ \forall n, m, t \in N, M, T$

C3: $(1 - \delta_{nm}^{t}) E_{n,loc}^{t} + \delta_{nm}^{t} E_{n,off}^{t} \le \min_{E_{total}} (\rho_{nm}^{t}, E_{n,loc}^{t}),\ \forall n, m, t \in N, M, T$

C4: $0 \le f_{nm}^{t} \le F,\ \forall n, m, t \in N, M, T$

C5: $\sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} \delta_{nm}^{t} f_{nm}^{t} \le F,\ \forall n, m, t \in N, M, T$

C6: $f_{nm}^{t} \ge \frac{S_{nm}^{t}}{\tau_{nm}^{t} - \frac{K_{nm}^{t}}{r_n^t}},\ \forall n, m, t \in N, M, T$

محدودیت C1 که مربوط به یک متغیر دودویی می‌باشد، بیانگر آن است که تمامی زیروظایف یا به صورت محلی یا به صورت بارسپاری‌شده اجرا می‌شوند. محدودیت‌های C2 و C3 به ترتیب بیانگر آن هستند که کل زمان اجرا و انرژی مصرف شده برای اجرای یک وظیفه نباید از حد قابل قبول آنها بیشتر شود که در

کارهای پژوهشی مختلف، حد قابل قبول زمان اجرا و انرژی مصرفی را عموماً با مقادیر آنها در اجرای محلی می‌سنجند. محدودیت C4 بیانگر آن است که مجموع منابع تخصیص‌یافته محلی در تمامی شکاف‌های زمانی نمی‌تواند از کل منابع موجود در سرور MEC بیشتر باشد. محدودیت C5 بیانگر آن است که مجموع منابع تخصیص‌یافته برای اجرای بارسپاری‌شده‌ی زیروظایف مختلف در تمامی شکاف‌های زمانی نمی‌تواند از مجموع منابع موجود در سرور MEC بیشتر شود. محدودیت C6 که در راستای محدودسازی فضای راه‌حل در نظر گرفته شده است، بیانگر آن است که ظرفیت محاسباتی تخصیص‌یافته از سرور MEC برای اجرای بارسپاری‌شده‌ی زیروظایف مختلف در یک شکاف زمانی باید همواره بیشتر از حاصل تقسیم تعداد چرخه‌های پردازشی لازم برای تکمیل زیروظایف به تفاضل بیشینه زمان اجرای قابل تحمل برای اجرا و انتقال داده‌های آنها باشد. شایان ذکر است که با ارضای محدودیت C6 می‌توان کارایی فرایند تصمیم‌گیری در خصوص انجام یا عدم انجام بارسپاری را نسبت به حالتی که چنین محدودیتی در نظر گرفته نشود، بهبود بخشید.

**الگوریتم ۱ – روش Q-Learning ارائه‌شده**

**Input:** state space S, action space A, learning rate $\boldsymbol{\psi}$, discount factor $\boldsymbol{\lambda}$
**Output:** The Q-values for each state-action pair

**Initialize** $Q(s, a)$ $\forall s \in StateSpace$, $a \in ActionSpace$
  **for** each episode **do**
    Choose a random state for initializing $s_t$
    **for** each step of episode, **do**
      Choose an action from some possible actions of current state $s_t$
      Execute chosen $a_t$, observe the reward $R_t$ and next state $s_{t+1}$
      $$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \psi\left[R(s_t, a_t) + \lambda \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right]$$
      Let $s_t \leftarrow s_{t+1}$
      **Terminate** when expected state $s_{terminal}$ reaches
    **end for**
  **end for**

### ۳-۳- الگوریتم حل مسئله

در این بخش ابتدا به تعریف $State$، $Action$ و $Reward$ می‌پردازیم و سپس بر مبنای آنها الگوریتم‌های مربوط به روش‌های پیشنهادی را ارائه می‌کنیم. لازم به ذکر است که دستگاه‌های کاربران، نقش عامل‌ها را در سیستم ایفا می‌نمایند.

$State$: حالت‌های سیستم با استفاده از یک دوتایی $(ct, ar)$ تعریف شده که $ct$ بیانگر هزینه‌ی کلی سیستم (محاسبه شده در بخش قبلی) و $ar$ بیانگر ظرفیت محاسباتی در دسترس سرور $MEC$ موجود در محیط بوده که از طریق رابطه‌ی $F - \sum_{n=0}^{N}\sum_{m=0}^{M}\sum_{t=0}^{T} f_{nm}^{t}$ قابل محاسبه می‌باشد.

$Action$: برای تعریف کنش‌های موجود در سیستم، ترکیبی از مقادیر ممکن برای تصمیم‌گیری بارسپاری $(D_o(t))$ و تخصیص منابع $(R_a(t))$ به صورت توأمان در نظر گرفته می‌شود.

$Reward$: در هر حالت $s_t$ و با انجام کنش $a_t$ بهترین مقدار ممکن $R(s_t, a_t)$ در هر مرحله منظور شده که این مقدار با استفاده از رابطه نرمال‌شده‌ی $\frac{ct_{loc}^{t} - ct_{(s,a)}^{t}}{ct_{loc}^{t}}$ محاسبه می‌شود.

مطابق با الگوریتم ۱ ابتدا مقداردهی اولیه Q-Table با مقدار صفر انجام شده و پس از آن به انتخاب یک کنش موجود با استفاده از سیاست حریصانه مبتنی بر نرخ یادگیری از محیط پرداخته شده و پاداش برآمده از اجرای آن محاسبه می‌شود. این عمل تا زمانی که فرایند یادگیری به صورت اجباری یا از طریق دستیابی به بهترین حالت ممکن متوقف شود، ادامه پیدا می‌کند تا در هر مرحله باعث بروزرسانی مقادیر مربوط به Q-Table شود. لازم به ذکر است که نرخ یادگیری از محیط در ابتدا بالا بوده و در طول انجام فرایند یادگیری به مرور کاهش پیدا کرده تا استفاده بیشتری از مقادیر ثبت شده در Q-Table رخ بدهد. می‌توان نتیجه گرفت که هر عامل به‌طور جداگانه از طریق تعامل مکرر با محیط و در نظر گرفتن پاداش تجمعی به دست آمده از اجرای کنش‌های موجود در هر حالت، اقدام به اتخاذ تصمیم‌گیری در خصوص انتخاب کنش‌های بعدی می‌نماید.

## ۴- نتایج عددی

در این بخش به منظور بررسی کارایی روش ارائه‌شده، نتایج حاصل از شبیه‌سازی‌های صورت گرفته ارائه می‌گردد.

### ۴-۱- تنظیمات شبیه‌سازی

برای انجام شبیه‌سازی‌های مد نظر به منظور ارزیابی کارایی روش پیشنهادی از زبان پایتون ۳.۷ و یک کامپیوتر شخصی با مشخصات ۸ گیگابایت رَم، ۲ ترابایت حافظه داخلی و پردازنده‌ی Intel® Core™ i7-6700HQ @ 2.6GHz استفاده شده است. وزن زمان و انرژی مصرفی در محاسبه هزینه‌ی کلی سیستم یکسان و برابر با ۰.۵ و ظرفیت محاسباتی محلی هر کاربر از طریق توزیع یکنواخت تصادفی بین ۰.۳ تا ۱.۸ گیگاهرتز بر ثانیه در نظر گرفته شده است. ضمناً پهنای باند کانال مورد استفاده و توان محاسباتی سرور MEC نیز به ترتیب برابر با ۱۰ مگاهرتز و ۸ گیگاهرتز بر ثانیه می‌باشد. سایر پارامترهای مختلف تعیین شده برای انجام شبیه‌سازی بر اساس کارهای مشابه پیشین تعیین و طبق جدول ۳ مقداردهی شده است.

**جدول ۳ مقادیر پارامترهای شبیه‌سازی و مفاهیم آنها**

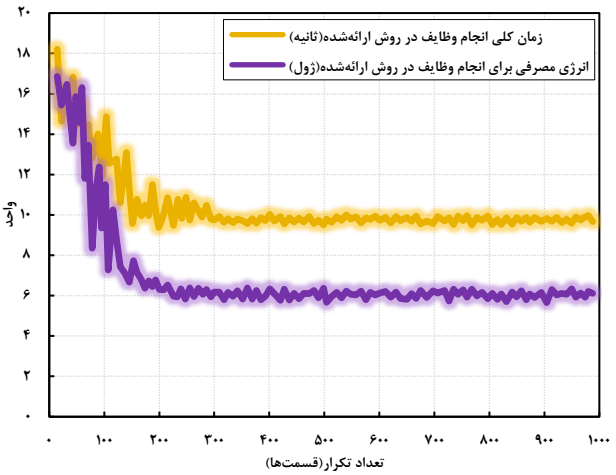
| نماد | مفهوم | مقدار |
|---|---|---|
| $N_0$ | نویز گاوسی کانال | -158 [dBm] |
| $g_n$ | بهره کانال برای کاربر $n$ | $127+30\log(L)$ [۴۹] |
| $W$ | پهنای باند کانال | 10 [MHz] |
| $F$ | کل ظرفیت محاسباتی سرور MEC | 8 [GHz/s] |
| $f_n^{loc}$ | ظرفیت محاسباتی محلی هر کاربر $n$ | تصادفی بر اساس توزیع یکنواخت بین [0.3,1.8] [GHz/s] |
| $P_n$ | توان انتقال داده توسط کاربر $n$ | 0.5 [W] |
| $P_n^w$ | توان مصرفی کاربر $n$ در حالت آماده به کار | 0.1 [W] |
| $m$ | تعداد مولفه‌های هر وظیفه | تصادفی بر اساس توزیع یکنواخت بین [3,10] |
| $P_n^{MEC}$ | توان مصرفی لازم برای تکمیل زیروظایف در سرور MEC | 5 [W] |
| $K_{nm}^t$ | مقدار محاسبات لازم برای هر $\varphi_{nm}^t$ | تصادفی بر اساس توزیع یکنواخت بین [16,80] [Mbit] |
| $S_{nm}^t$ | تعداد چرخه‌های پردازشی لازم برای تکمیل هر $\varphi_{nm}^t$ | تصادفی بر اساس توزیع یکنواخت بین [1,3.5] [Gigacycle] |
| $I_n^t$ | وزن زمان در هزینه‌ی کلی سیستم | 0.5 |
| $I_n^e$ | وزن انرژی در هزینه‌ی کلی سیستم | 0.5 |
| $\tau_{nm}^t$ | بیشینه زمان اجرای قابل تحمل برای اجرای هر $\varphi_{nm}^t$ | 1 [s] |
| $\rho_{nm}^t$ | بیشینه انرژی مصرفی قابل تحمل برای اجرای هر $\varphi_{nm}^t$ | 1 [J] |

### ۴-۲- مقایسه و ارزیابی کارایی

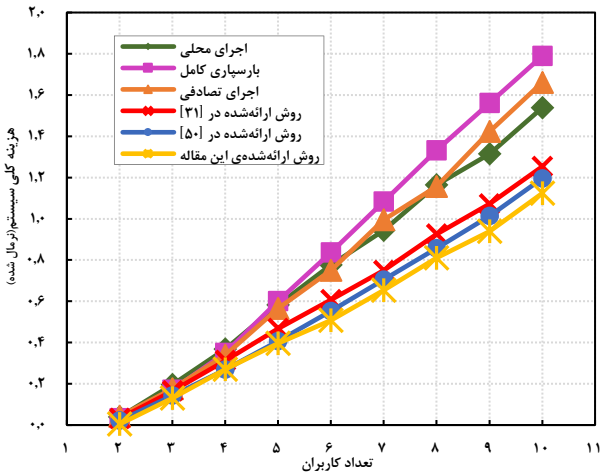
در این قسمت ابتدا به نمایش همگرایی[۲۷] و سپس مقایسه نتایج به دست آمده با سه روش پایه و دو کار انجام‌شده در مراجع [۳۱] و [۵۰] پرداخته می‌شود. اجرای محلی و بارسپاری کامل به ترتیب بیانگر پردازش تمامی زیروظایف در دستگاه‌های کاربران و سرور MEC موجود در محیط با در نظر گرفتن حداکثر تأخیر و مصرف انرژی قابل تحمل در هر دوی آنها می‌باشد. همچنین اجرای تصادفی از طریق طی یک فرایند تصادفی، یکی از دو روش قبل را برای هر زیروظیفه انتخاب می‌کند.

شکل ۳ بیانگر همگرایی الگوریتم پیشنهادی از نظر زمان و انرژی مصرفی تکمیل وظایف هنگام حضور تعداد پنج کاربر در سیستم می‌باشد. با توجه به این شکل می‌توان دریافت که مقدار این دو پارامتر با افزایش تعداد تکرارها (قسمت‌ها) دچار کاهش نسبتاً سریعی شده و پس از همگرایی به یک محدوده‌ی خاص، دیگر با افزایش تعداد تکرارها تغییر چندانی نمی‌کند که این رخداد نشان‌دهنده‌ی موفقیت الگوریتم پیشنهادی در تعامل با عناصر مؤثر محیط مدل‌شده و یادگیری مناسب از آن می‌باشد.

با توجه به اینکه هزینه‌ی کلی سیستم شامل جمع وزن‌دار نرمالی از زمان تکمیل وظایف و میزان انرژی مصرفی آنان است، شکل ۴ تغییرات هزینه‌ی سیستم را در برابر تغییر تعداد کاربران نشان




**شکل ۳ – همگرایی روش ارائه‌شده در زمان و انرژی مصرفی کلی وظایف**

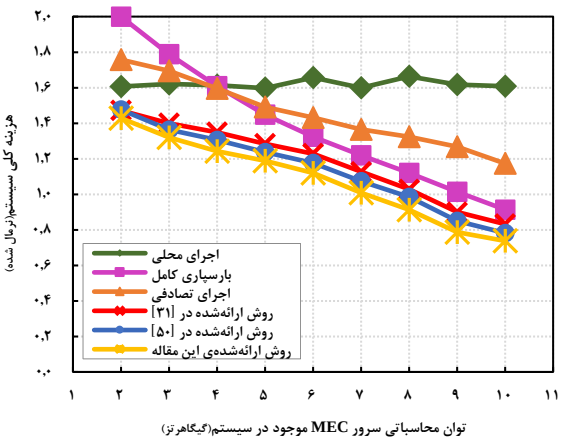



**شکل ۴ – هزینه‌ی کلی سیستم در برابر تغییر تعداد کاربران سیستم**

می‌دهد که نشان‌دهنده‌ی افزایش هزینه‌ی کلی سیستم با افزایش تعداد کاربران است. نکته‌ی قابل توجه این است که با افزایش تعداد کاربران سیستم (بیشتر از ۵ کاربر)، عملکرد روش بارسپاری کامل با سرعت بسیار زیادی افت می‌کند و نتایج ضعیفی به همراه دارد. همچنین، در مقایسه با سایر مورد مقایسه، روش پیشنهادی ما عملکرد بهتری از خود نشان داده است.

شکل ۵ نشان‌دهنده‌ی تغییرات هزینه‌ی کلی نرمال‌شده‌ی سیستم در برابر تغییر توان محاسباتی سرور است که کاهش هزینه‌ی کلی سیستم با افزایش این توان محاسباتی را نشان می‌دهد. در مقایسه با روش‌های پایه و دو روش ارائه‌شده در مقالات [۳۱] و [۵۰]، روش پیشنهادی ما عملکرد بهتری در کاهش هزینه‌ی کلی سیستم نشان داده است. لازم به ذکر است که با تأمین سرور پرقدرت‌تر، عملکرد بارسپاری کامل به سرعت بهبود می‌یابد و در توان محاسباتی بالاتر از ۴ گیگاهرتز حتی از اجرای محلی نیز

پیشی می‌گیرد. تغییر توان محاسباتی سرور لبه‌ای دسترسی چندگانه نباید تغییر چشم‌گیری در کارایی روش اجرای محلی ایجاد کند، چرا که این روش ارتباطی با سرور MEC موجود ندارد. با این حال، ظرفیت محاسباتی محدود محلی هر کاربر، میزان محاسبات لازم برای وظایف مختلف، تعداد مؤلفه‌های هر وظیفه و تعداد چرخه‌های پردازشی لازم برای اجرای هر زیروظیفه مربوط به آن می‌باشد.

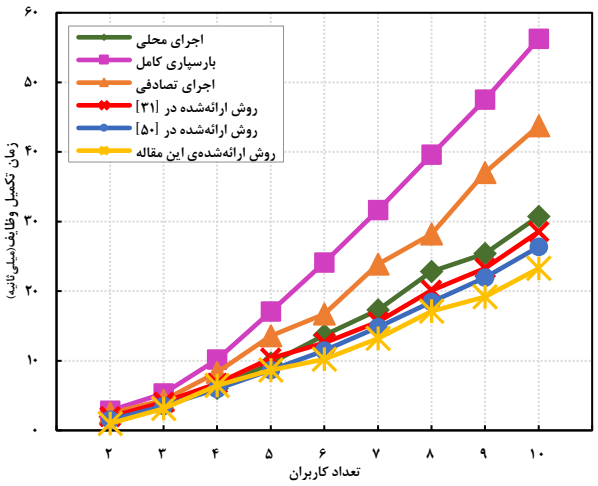



**شکل ۵ – هزینه‌ی کلی سیستم در برابر تغییر ظرفیت محاسباتی سرور**

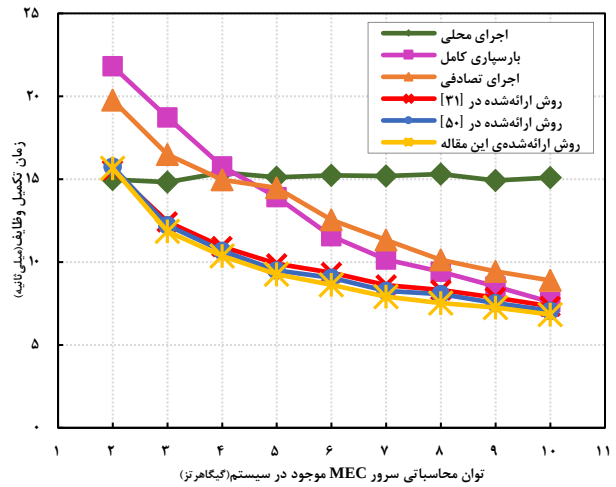
شکل ۶ نشان‌دهنده‌ی تغییرات زمان تکمیل وظایف در سیستم با تغییر تعداد کاربران است که افزایش این معیار ارزیابی سیستم را با افزایش تعداد کاربران نشان می‌دهد. در این میان، مقایسه‌ی نتایج روش پیشنهادی ما با روش‌های مورد مقایسه، برتری آن را در کاهش زمان تکمیل وظایف نشان می‌دهد. نکته‌ی قابل توجه دیگر در این شکل، آهنگ تغییر زمان تکمیل وظایف در روش بارسپاری کامل نسبت به سایر روش‌هاست. از این مسئله می‌توان دریافت که استفاده توأمان از تقسیم‌بندی وظایف محاسباتی و بهره‌گیری صرف از تکنیک بارسپاری وظایف نه‌تنها ارزش افزوده‌ای برای کارایی سیستم ایجاد نمی‌کند، بلکه به علت افزایش ابعاد تصمیم‌گیری و متغیرهای اثرگذار، باعث افت سریع‌تر عملکرد سیستم می‌شود. با توجه به پیشرفت‌های فناوری و استفاده بیشتر از سخت‌افزارهای محاسباتی قدرتمند در دستگاه‌های کاربران، محدودیت اصلی ما در انجام وظایف به نقل و انتقال داده‌ها مربوط می‌شود. از این رو، همان‌طور که نتایج نشان می‌دهد، زمان تکمیل وظایف با اجرای محلی نسبت به بارسپاری کامل کمتر است.

شکل ۷ نشان‌دهنده‌ی تغییرات زمان تکمیل وظایف در سیستم در برابر تغییر توان محاسباتی سرور لبه‌ای دسترسی چندگانه است که کاهش کلی این معیار ارزیابی سیستم را با افزایش توان محاسباتی نشان می‌دهد. مقایسه‌ی این نتایج با روش‌های پایه و دو روش ارائه‌شده در مقالات [۳۱] و [۵۰] نشان می‌دهد که روش پیشنهادی ما عملکرد بهتری داشته و زمان تکمیل وظایف را به شکل موثرتری کاهش می‌دهد. نکته‌ی دیگر، آهنگ تغییر زمان تکمیل وظایف در روش بارسپاری کامل نسبت به سایر روش‌هاست که از آن می‌توان نتیجه گرفت وابستگی بالایی بین عملکرد سیستم و توان پردازشی سرور MEC وجود دارد.

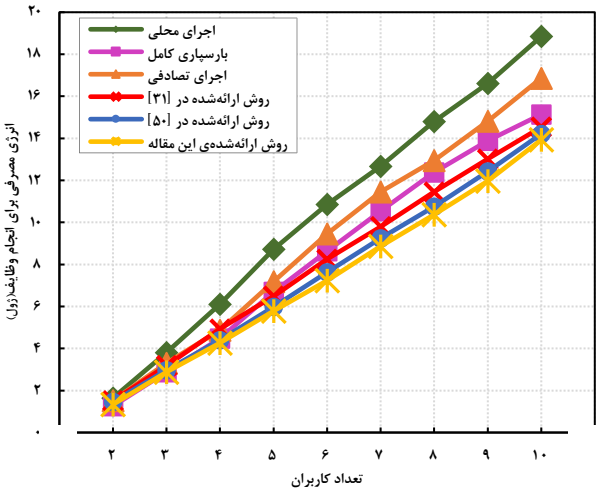
شکل ۸ نشان‌دهنده‌ی تغییرات انرژی مصرفی وظایف در سیستم با تغییر تعداد کاربران است که افزایش این معیار ارزیابی سیستم را با افزایش تعداد کاربران نشان می‌دهد. در مقایسه با روش‌های پایه و دو روش ارائه‌شده در مقالات [۳۱] و [۵۰]، روش پیشنهادی ما به‌طور قابل‌توجهی در کاهش انرژی مصرفی عملکرد بهتری داشته است. نکته‌ی مهم این شکل مربوط به محدودیت اصلی اجرای محلی وظایف در حوزه‌ی انرژی مصرفی و میزان ظرفیت باتری دستگاه‌های مختلف کاربران است که این محدودیت با بالاتر بودن نمودار مربوط به اجرای محلی نسبت به تمامی روش‌های دیگر مشخص شده است.




**شکل ۶ – زمان تکمیل وظایف در برابر تغییر تعداد کاربران سیستم**

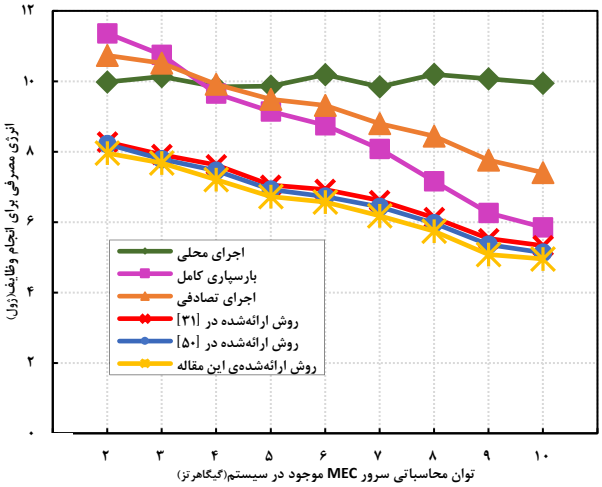



**شکل ۷ – زمان تکمیل وظایف در برابر تغییر ظرفیت محاسباتی سرور**




**شکل ۸ – انرژی مصرفی انجام وظایف در برابر تغییر تعداد کاربران سیستم**

شکل ۹ نشان‌دهنده‌ی تغییرات انرژی مصرفی کلی وظایف در سیستم در برابر تغییر توان محاسباتی سرور MEC است که کاهش انرژی مصرفی کل سیستم را با افزایش توان محاسباتی نشان می‌دهد. مقایسه‌ی این نتایج با روش‌های مورد مقایسه برتری روش پیشنهادی ما را در کاهش انرژی مصرفی تأیید می‌کند. با افزایش توان محاسباتی سرور MEC، انرژی مصرفی برای پردازش محاسبات کاهش یافته و این موضوع به‌طور مستقیم بر انرژی مصرفی کلی وظایف تأثیر می‌گذارد. همچنین، با تغییر توان محاسباتی سرور MEC و عدم بهره‌گیری از آن در روش اجرای محلی، نباید تغییر چشم‌گیری در میزان انرژی مصرفی مشاهده شود. علاوه بر این، با افزایش توان محاسباتی سرور لبه‌ای دسترسی چندگانه، تعداد دفعات انتقال داده‌ها کاهش یافته و این امر باعث بهبود عملکرد روش بارسپاری کامل نسبت به اجرای محلی در ظرفیت‌های بالاتر از ۴ گیگاهرتز می‌شود.




**شکل ۹ – انرژی مصرفی انجام وظایف در برابر تغییر ظرفیت محاسباتی سرور**

## ۵- نتیجه‌گیری و کارهای آینده

در این مقاله به بررسی مسئله بارسپاری جزئی وظایف ناهمگن و تخصیص توأمان منابع در یک محیط MEC چندکاربره و متغیر با زمان پرداختیم. محدودیت‌هایی از قبیل مهلت زمانی قابل تحمل و تأخیر اجرای وظایف، میزان انرژی مصرفی دستگاه‌های کاربران و بهینگی تخصیص منابع را در هدف نهایی کاهش هزینه‌ی کلی ناشی از جمع وزن‌دار نرمالی از انرژی مصرفی و زمان تکمیل وظایف در نظر گرفته و برای دستیابی به آن یک الگوریتم مبتنی بر یادگیری تقویتی بر اساس روش Q-Learning ارائه نمودیم. ضمناً به منظور جلوگیری از گستردگی بیش از اندازه در فضای حالت و کنش‌های در دسترس به تنظیم محدودیتی ابتکاری در حوزه‌ی ظرفیت محاسباتی تخصیص‌یافته از سرور MEC برای اجرای بارسپاری‌شده‌ی زیروظایف مختلف در یک شکاف زمانی اقدام کردیم. با استناد به نتایج عددی به دست آمده حاصل از انجام شبیه‌سازی تحت سناریوهای مختلف و مقایسه آنها با سه روش پایه و دو کار انجام شده در مراجع [۳۱] و [۵۰]، بهبود مؤثر پارامترهای مذکور و کارایی روش ارائه‌شده قابل ملاحظه می‌باشد.

در ادامه می‌توان با در نظر گرفتن چندین سرور و امکان مهاجرت برخی از زیروظایف میان آنها با بهره‌گیری از انواع روش‌های مبتنی بر یادگیری عمیق به منظور مقابله با پیچیدگی‌های زیاد عوامل اثرگذار به عمق ایده‌ی پیشنهادی در این مقاله افزود. همچنین در نظر گرفتن امکان وقوع چندمسیری[۲۸] در صورت استفاده از بارسپاری دستگاه به دستگاه، تعبیه سرورهای MEC در وسائل نقلیه و زمانبندی اجرای مناسب عملیات واگذاری[۲۹] محاسبات و ترکیب شبکه‌های نرم‌افزار محور[۳۰] با محاسبات لبه‌ای دسترسی چندگانه برای انجام بارسپاری وظایف از دیگر مواردی هستند که در آینده می‌توان به آنها پرداخت.

## مراجع


[1] "Cisco Annual Internet Report (2018–2023)", White Paper.

[2] W. Yang, N. Wang, Z. Guan, L. Wu, X. Du and M. Guizani, "A Practical Cross-Device Federated Learning Framework over 5G Networks", *IEEE Wireless Communications*, pp. 1-8, 2022.

[3] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward Computation Offloading in Edge Computing: A Survey", *IEEE Access*, vol. 7, pp. 131543–131558, 2019.

[۴] آتوسا دقایقی و محسن نیک‌رای، "ارائه‌ی یک مدل زمان‌بندی وظایف مبتنی بر الگوریتم ژنتیک چند هدفه با مرتب‌سازی نامغلوب برای برنامه‌های کاربردی شهر هوشمند در محیط ابر-مه"، *مجله علمی رایانش نرم و فناوری اطلاعات*، دوره ۱۱، شماره ۳، صفحات ۶۴-۸۲، ۱۴۰۱.

[5] L. Lei, H. Xu, X. Xiong, K. Zheng and W. Xiang, "Joint Computation Offloading and Multiuser Scheduling Using Approximate Dynamic Programming in NB-IoT Edge Computing

[24] L. Liu, C. Chen, Q. Pei, S. Maharjan and Y. Zhang, "Vehicular Edge Computing and Networking: A Survey", *Mobile Networks and Applications*, vol. 26, pp. 1145–1168, 2021.

[۲۵] م. دارچینی‌تبریزی و ر. انتظاری‌ملکی، "مروری بر بارسپاری آگاه از کارایی وظایف در محاسبات لبه‌ای سیار"، *بیست و هفتمین کنفرانس بین‌المللی کامپیوتر انجمن کامپیوتر ایران*، ۱۴۰۰.

[26] G. Carvalho, B. Cabral, V. Pereira, and J. Bernardino, "Computation Offloading in Edge Computing Environments Using Artificial Intelligence Techniques", *Engineering Applications of Artificial Intelligence*, vol.95, no.103840, pp. 1-19, 2020.

[27] S. Wu, W. Xia, W. Cui, Q. Chao, Z. Lan, F. Yan and L. Shen, "An Efficient Offloading Algorithm Based on Support Vector Machine for Mobile Edge Computing in Vehicular Networks", *10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1-6, 2018.

[28] I. Goodfellow, Y. Bengio, and A. C. Courville, "Deep Learning", *Nature*, vol. 521, pp. 436–444, 2015.

[29] S. Teerapittayanon, B. McDanel and H. T. Kung, "Distributed Deep Neural Networks Over the Cloud, the Edge and End Devices", *IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 328-339, 2017.

[30] C. Zhang, P. Patras and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey", *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224-2287, 2019.

[31] K. Jiang, H. Zhou, D. Li, X. Liu and S. Xu, "A Q-learning Based Method for Energy-Efficient Computation Offloading in Mobile Edge Computing", *29th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1-7, 2020.

[32] H. Chen, S. Deng, H. Zhu, H. Zhao, R. Jiang, S. Dustdar and A. Y. Zomaya, "Mobility-Aware Offloading and Resource Allocation for Distributed Services Collaboration", *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 10, pp. 2428-2443, 2022.

[33] C. Yang, Y. Liu, X. Chen, W. Zhong and S. Xie, "Efficient Mobility-Aware Task Offloading for Vehicular Edge Computing Networks", *IEEE Access*, vol. 7, pp. 26652-26664, 2019.

[34] L. Huang, S. Bi and Y. -J. A. Zhang, "Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks", *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2581-2593, 2020.

[35] H. Hu, Q. Wang, R. Q. Hu and H. Zhu, "Mobility-Aware Offloading and Resource Allocation in a MEC-Enabled IoT Network With Energy Harvesting", *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17541-17556, 2021.

[36] K. Li, "Heuristic Computation Offloading Algorithm for Mobile Users in Fog Computing", *ACM Transactions on Embedded Computing Systems*, vol. 20, no. 11, pp. 1-28, 2021.

[37] J. Yan, S. Bi and Y. J. A. Zhang, "Offloading and Resource Allocation With General Task Graph in Mobile Edge Computing: A Deep Reinforcement Learning Approach", *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5404-5419, 2020.

[38] Z. Ning, P. Dong, X. Kong and F. Xia, "A Cooperative Partial Computation Offloading Scheme for Mobile Edge Computing Enabled Internet of Things", *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4804-4814, 2019.

[39] Z. Li, L. Shi, Y. Shi, Z. Wei and Y. Lu, "Task Offloading Strategy to Maximize Task Completion Rate in Heterogeneous Edge Computing Environment", *Computer Networks*, vol. 210, no. 108937, pp. 1-14, 2022.

[40] U. Saleem, Y. Liu, S. Jangsher, Y. Li and T. Jiang, "Mobility-Aware Joint Task Scheduling and Resource Allocation for Cooperative Mobile Edge Computing", *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 360-374, 2021.

[41] D. Zhang, L. Tan, J. Ren, M. K. Awad, S. Zhang, Y. Zhang and P. J. Wan, "Near-Optimal and Truthful Online Auction for Computation Offloading in Green Edge-Computing Systems", *IEEE Transactions on Mobile Computing*, vol. 19, no. 4, pp. 880-893, 2020.

[42] L. Qian, Y. Wu, F. Jiang, N. Yu, W. Lu and B. Lin, "NOMA Assisted Multi-Task Multi-Access Mobile Edge Computing via Deep Reinforcement Learning for Industrial Internet of Things", System", *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5345-5362, 2019.

[6] F. Zantalis, G. Koulouras, S. Karabetsos, and D. Kandris, "A Review of Machine Learning and IoT in Smart Transportation", *Future Internet*, vol. 11, no. 4, pp. 1-23, 2019.

[7] F. Al-Turjman, A. Malekloo, "Smart Parking in IoT-enabled Cities: A Survey", *Sustainable Cities and Society*, vol 49, pp. 1-20, 2019.

[8] H. Yang, Z. Wei, Z. Feng, X. Chen, Y. Li and P. Zhang, "Intelligent Computation Offloading for MEC-Based Cooperative Vehicle Infrastructure System: A Deep Reinforcement Learning Approach", *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7665-7679, 2022.

[9] F. Lu, H. Zhao, X. Zhao, X. Wang, A. Saleem, and G. Zheng, "Investigation of Near-Field Source Localization Using Uniform Rectangular Array", *Electronics*, vol. 11, no. 12, pp. 1-15, 2022.

[10] S.K.u. Zaman, A.I. Jehangiri, T. Maqsood, Z. Ahmad, A.I. Umar, J. Shuja, E. Alanazi, W. Alasmary, "Mobility-aware Computational Offloading in Mobile Edge Networks: a Survey", *Cluster Computing*, vol. 24, pp. 2735–2756, 2021.

[11] J. Plachy, Z. Becvar, E. C. Strinati and N. d. Pietro, "Dynamic Allocation of Computing and Communication Resources in Multi-Access Edge Computing for Mobile Users", *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 2089-2106, 2021.

[12] J. Shuja, K. Bilal, W. Alasmary, H. Sinky, E. Alanazi, "Applying Machine Learning Techniques for Caching in Next-generation Edge Networks: A Comprehensive Survey", *Journal of Network and Computer Applications*, vol. 181, pp. 1-24, 2021.

[13] N. Abbas, Y. Zhang, A. Taherkordi and T. Skeie, "Mobile Edge Computing: A Survey", *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450-465, 2018.

[14] Y. Liao, L. Shou, Q. Yu, Q. Ai, and Q. Liu, "Joint Offloading Decision and Resource Allocation for Mobile Edge Computing Enabled Networks," *Computer Communication*, vol. 154, pp. 361–369, 2020.

[15] F. Saeik, M. Avgeris, D. Spatharakis, N. Santi, D. Dechouniotis, J. Violos, A. Leivadeas, N. Athanasopoulos, N. Mitton, and S. Papavassiliou, "Task Offloading in Edge and Cloud Computing: A Survey on Mathematical, Artificial Intelligence and Control Theory Solutions", *Computer Networks*, vol.195, pp. 1-26, 2021.

[۱۶] سعدون عزیزی، "یک مدل چندهدفه برای بهینه‌سازی زمان‌بندی وظیفه‌ها در محیط‌های رایانشی مه-ابر"، *مجله علمی رایانش نرم و فناوری اطلاعات*، دوره ۱۰، شماره ۴، صفحات ۴۳-۵۲، ۱۴۰۰.

[17] M. Deng, H. Tian and B. Fan, "Fine-granularity Based Application Offloading Policy in Cloud-enhanced Small Cell Networks", *IEEE International Conference on Communications Workshops (ICC)*, pp. 638-643, 2016.

[18] F. Farahbakhsh, A. Shahidinejad, and M. Ghobaei-Arani, "Multiuser Context-aware Computation Offloading in Mobile Edge Computing Based on Bayesian Learning Automata", *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, pp. 1–26, 2021.

[19] F. Sufyan and A. Banerjee, "Computation Offloading for Distributed Mobile Edge Computing Network: A Multi-objective Approach", *IEEE Access*, vol. 8, pp. 149915–149930, 2020.

[20] P. Match and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading", *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1628-1656, 2017.

[21] A. Baktayan and I. Al-Balta, "Intelligent Computation Offloading and Pricing Strategies in UAV-enabled MEC Network for Utility Maximization: A Survey", *International Journal of Computing and Digital Systems*, pp. 1–21, 2021.

[22] A. Shakarami, M. Ghobaei-Arani, A. Shahidinejad, "A Survey on the Computation Offloading Approaches in Mobile Edge Computing: A Machine Learning-based Perspective", *Computer Networks*, vol. 182, pp. 1-24, 2020.

[23] A. Haydari and Y. Yılmaz, "Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 11-32, 2022.

*IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5688-5698, 2021.

[43] Z. Kuang, Z. Ma, Z. Li, X. Deng, "Cooperative Computation Offloading and Resource Allocation for Delay Minimization in Mobile Edge Computing", *Journal of Systems Architecture*, vol. 118, pp. 1-9, 2021.

[44] Z. Ali, Z. H. Abbas, G. Abbas, A. Numani, and M. Bilal, "Smart Computational Offloading for Mobile Edge Computing in Next-generation Internet of Things Networks," *Computer Networks*, vol.198, no.108356, pp. 1-9, 2021.

[45] 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 Journal*, vol. 7, no. 10, pp. 9255-9265, 2020.

[46] J. Li, H. Gao, T. Lv and Y. Lu, "Deep Reinforcement Learning Based Computation Offloading and Resource Allocation for MEC", *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1-6, 2018.

[47] P. Zhao, H. Tian, C. Qin and G. Nie, "Energy-Saving Offloading by Jointly Allocating Radio and Computational Resources for Mobile Edge Computing", *IEEE Access*, vol. 5, pp. 11255-11268, 2017.

[48] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic Computation Offloading for Mobile-edge Computing With Energy Harvesting Devices", *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[49] M. Chen, B. Liang and M. Dong, "Joint Offloading and Resource Allocation for Computation and Communication in Mobile Cloud With Computing Access Point", *IEEE Conference on Computer Communications*, pp. 1-9, 2017.

[50] Z. Wang, B. Lin, L. Sun and Y. Wang, "Intelligent Task Offloading for 6G-Enabled Maritime IoT Based on Reinforcement Learning", International Conference on Security, Pattern Analysis, and Cybernetics（SPAC), pp. 566-570, 2021.

## پاورقی‌ها:

[1] Cisco
[2] Unmanned Aerial Vehicles
[3] Tasks Offloading
[4] Multi-access Edge Computing
[5] Delay Sensitive
[6] Fading
[7] Fully Offloading
[8] Partial Offloading
[9] QoS
[10] Resource Allocation
[11] Reinforcement Learning
[12] Game Theory
[13] Lyapunov Optimization
[14] Programming
[15] Nash Equilibrium
[16] Penalty
[17] Drift
[18] Deep Reinforcement Learning
[19] Low-complexity Critic Network
[20] Replay Buffer Memory
[21] Branch and Bound Algorithm
[22] Coherent Beamforming
[23] Successive Interference Cancellation
[24] D2D
[25] Dispatcher
[26] Non-orthogonal Multiple Access
[27] QoE
[28] Multipath
[29] Handover
[30] SDN