

# زمان بندی مبتنی بر اولویت وظایف با استفاده از سیستم فازی در محاسبات لبه سیار

انتصار حسینی، محسن نیکرآی و شمس اله قنبری

همراه به نام نسل پنجم (5G) با هدف تسریع در توسعه برنامه‌های کاربردی دستگاه‌های سیار به وجود آمدند. شبکه‌های 5G بر استفاده از تکنولوژی‌های نوین از جمله محاسبات ابری سیار (MCC) و محاسبات لبه سیار (MEC) برای دستیابی به انعطاف پذیری بیشتر و قابلیت پردازش بی‌درنگ تأکید می‌کنند [۳] و [۴]. برخی از برنامه‌های کاربردی دستگاه‌های سیار مانند واقعیت افزوده، بازی و تشخیص چهره، نیاز به توان محاسباتی بالایی دارند و به این دلیل در هنگام پردازش، انرژی زیادی مصرف می‌کنند [۵]. از طرفی، تقسیم منابع در محاسبات ابری به دلیل دسترسی دور کاربر باعث تأخیر در زمان پاسخ می‌شود و به این دلیل نیاز به وجود یک فناوری جدید برای سرعت بخشیدن و افزایش کارایی در برنامه‌های حساس به تأخیر و بی‌درنگ احساس می‌شود. از سال ۲۰۱۶، مؤسسه استانداردهای ارتباطات اروپا (ETSI) یک چارچوب و معماری مرجع MEC تولید کرد. این معماری از خدماتی مانند اجرای برنامه‌های پویا و انتشار اطلاعات شبکه رادیویی به صورت بلادرنگ پشتیبانی می‌کند. [۶]. محاسبات لبه بالقوه برای رفع محدودیت‌های زمان پاسخ، محدودیت عمر باتری، صرفه جویی در هزینه پهنای باند و همچنین ایمنی داده‌ها و حفظ حریم خصوصی است که در نزدیکی کاربر، خدمات خود را ارائه می‌دهد [۷]. با توجه به منابع محدود دستگاه‌های هوشمند، صرفه جویی در منابع و انرژی موضوع مهمی است. محاسبات لبه به کاربران تلفن همراه اجازه می‌دهد که وظایف خود را در لبه به صورت پویا انجام دهند و بنابراین از این طریق، هزینه استفاده از منابع دستگاه کاهش می‌یابد. هدف MEC، ارتباط میلیون‌ها برنامه بی‌درنگ و حساس به تأخیر در لبه شبکه است، به طوری که به دنبال این کار، هزینه مصرف انرژی در دستگاه سیار کمتر شود، وظایف بی‌درنگ با سرعت بیشتری اجرا شوند و فضای ذخیره‌سازی داده و پردازش برنامه‌ها افزایش یابد. استفاده از این بستر علاوه بر مزیت‌هایی که به همراه دارد با چالش‌هایی نیز روبه‌رو است. انتخاب وظایف مناسب برای ارسال به MEC نیاز به دقت بالایی دارد زیرا در بعضی از موارد، ارسال مجموعه‌ای از وظایف در شبکه، هزینه بیشتری را از لحاظ انرژی مصرفی در باتری یا پهنای باند شبکه به دستگاه سیار اعمال می‌کند. چالش مطرح دیگری در این زمینه در سمت MEC وجود دارد و آن این است که وظایف رسیده، بدون دانش قبلی وارد این لایه می‌شوند و یا حداقل، یافتن اطلاعات کافی درباره معیارهای اولویت بندی کار زمان‌بری است که در زمان بندی وظایف اهمیت زیادی دارد. اولویت بندی پویای وظایف با هزینه مناسب بر اساس دانش فعلی

چکیده: محاسبات لبه سیار، تکنولوژی نوینی برای بهبود مشکل تأخیر، ظرفیت و منابع موجود در محیط محاسبات ابری سیار است. هدف اصلی در محاسبات لبه سیار، زمان بندی پویا و بارگذاری بهینه با کمترین هزینه در استفاده از منابع است. ما در این مقاله، از یک مدل سیستم سه سطحی دستگاه‌های سیار، لبه و ابر استاندارد، استفاده و دو الگوریتم بارگذاری و زمان بندی را پیشنهاد می‌کنیم. یک الگوریتم تصمیم‌گیری برای بارگذاری وظایف مبتنی بر الگوریتم کوله‌پشتی حریصانه در سمت دستگاه سیار است که وظایف با انرژی مصرفی بالا را برای بارگذاری انتخاب می‌کند و باعث صرفه جویی در انرژی مصرفی دستگاه می‌شود. همچنین در سمت MEC، یک الگوریتم زمان بندی پویا را با اولویت بندی وظایف مبتنی بر فازی جهت اولویت بندی و زمان بندی وظایف بر اساس دو معیار ارائه می‌کنیم. نتایج عددی نشان می‌دهند که کار ارائه شده در مقایسه با سایر روش‌ها باعث کاهش زمان انتظار وظایف برای اجرا، تأخیر و بار سیستم می‌شود و تعادل سیستم با کمترین تعداد منابع تأمین می‌گردد و سیستم ارائه شده، مصرف باتری را در دستگاه هوشمند تا حدود ۹۰٪ کاهش می‌دهد. نتایج نشان می‌دهند که بیش از ۹۲٪ وظایف با موفقیت در محیط لبه اجرا می‌شوند.

کلیدواژه: محاسبات لبه سیار، زمان بندی، حریصانه، فازی، انرژی مصرفی، زمان انتظار.

## ۱- مقدمه

امروزه دستگاه‌های تلفن همراه، نقش بسزایی در زندگی مردم ایفا می‌کنند. با توسعه شبکه تلفن همراه و اینترنت اشیا (IoT)، محاسبات، ذخیره سازی و مدیریت شبکه با چالش‌های بسیاری مواجه شده‌اند. این چالش‌ها شامل تأخیرهای زیاد، محدودیت فضای ذخیره سازی، دستگاه‌ها با منابع محدود، لزوم سرویس‌دهی متناوب بدون وقفه و امنیت بیشتر می‌باشند که با معماری مبتنی بر ابر به طور کامل حل نمی‌شوند [۱] و [۲]. دستگاه‌های سیار مانند تلفن‌های هوشمند، تبلت و دستگاه‌های پوشیدنی، از جمله دستگاه‌های فعال در IoT هستند که در آن بسیاری از حسگرها به سایر دستگاه‌ها در خانه‌ها، اداره‌ها و جاده‌ها تحت شبکه‌های پیشرفته امروزی متصل می‌شوند. نسل جدید شبکه سیستم‌های تلفن

این مقاله در تاریخ ۲۱ اردیبهشت ماه ۱۳۹۸ دریافت و در تاریخ ۱۸ خرداد ماه ۱۴۰۱ بازنگری شد.

انتصار حسینی، دانشکده فنی و مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه قم، قم، ایران، (email: e.hoesini@stu.qom.ac.ir).

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

شمس اله قنبری، دانشکده فنی و مهندسی کامپیوتر، واحد آشتیان، دانشگاه آزاد اسلامی، آشتیان، ایران، (email: myrshg@gmail.com).

1. Internet of Things

2. Fifth Generation

3. Mobile Cloud Computing

4. Mobile Edge Computing

5. European Telecommunications Standards Institute

عملکرد روش پیشنهادی نسبت به سایر روش‌ها مشخص می‌شود. قسمت‌های بعدی این مقاله به شرح زیر است: در بخش ۲، پیشینه تحقیق را بررسی و سپس در بخش ۳، مدل سیستم و روابط را مفصلاً بیان می‌کنیم. در بخش ۴، روش پیشنهادی برای بارگذاری و زمان‌بندی وظایف ارائه شده و نتایج شبیه‌سازی در بخش ۵ آمده است. نهایتاً در بخش ۶ خلاصه و نتیجه‌گیری مقاله ارائه می‌شود.

## ۲- کارهای پیشین

در این بخش، کارهای مربوط در زمینه بارگذاری و زمان‌بندی در محیط محاسبات سیار بررسی می‌گردد. کارهای مورد نظر در ۳ محیط محاسبات سیار تقسیم می‌شوند: MEC، ابر لبه سیار و MCC. اهداف کارهای ارائه‌شده در این ۳ محیط در ۳ زیربخش زیر آمده است.

### ۲-۱ محاسبات سیار ابری

تعدادی از محققان در [۱۱]، تعدادی از کلون‌ها را برای چندین دستگاه هوشمند در محیط ابر ایجاد کردند و وظایف از دستگاه سیار به ابر منتقل شد. در این صورت تعدادی کلون از یک دستگاه وجود دارد که در صورت آسیب‌دیدن یا گم‌شدن دستگاه، جایگزین برنامه‌های آن خواهد شد. مزیت دیگر این کار، مقابله با محدودیت سخت‌افزاری دستگاه هوشمند است. تجزیه و تحلیل داده‌های بزرگ با ترکیب MCC و Hadoop برای رفع محدودیت‌های ترافیکی و تأخیر نیز در [۱۲] ارائه شد. نویسندگان در [۱۳]، مسأله پارتیشن‌بندی محاسبات چندکاربره (MCCP) را برای به حداقل رساندن میانگین زمان‌بندی و زمان تأخیر مطالعه کردند. آنها از یک الگوریتم اکتشافی آفلاین یعنی جستجو در مدل (PRL) برای حل مشکل در برنامه‌های حساس به تأخیر استفاده نمودند. نویسندگان در [۱۴]، یک چهارچوب در زمینه برنامه‌های سلامت به نام Health Edge را برای زمان‌بندی وظایف پیشنهاد دادند. محققان، اولویت‌های مختلف پردازش را برای وظایف مختلف بر اساس داده‌های سلامت جمع‌آوری شده تنظیم کردند. Health Edge کل زمان پردازش را تا آنجا که ممکن است کاهش می‌دهد. نویسندگان در [۱۵]، الگوریتم RR را در محیط محاسبات ابری برای زمان‌بندی وظایف پیشنهاد دادند. نتایج الگوریتم RR نشان داد که زمان پاسخ و توازن بار در مقایسه با الگوریتم‌های دیگر بهینه‌تر است. یک کار دیگر، زمان‌بندی در ابر را با متوسط تأخیر در جریان ترافیکی و میانگین زمان انتظار در LTE به حداقل رساند [۱۶]. نویسندگان در [۱۷]، پیشنهاد یک برنامه زمان‌بندی و تصمیم‌گیری بارگذاری را بر اساس بهینه‌سازی Lyapunov برای صرفه‌جویی در زمان متوسط اجرا و متوسط مصرف انرژی ارائه دادند. در [۱۸]، نویسندگان یک مدل تصمیم‌گیری بارگذاری محاسباتی را به نام MCOB برای به حداقل رساندن مصرف انرژی و زمان اجرای آن پیشنهاد کردند. محققان در [۱۹]، یک مدل سلسله‌مراتبی جدید را در محاسبات ابر ارائه دادند که از لایه کاربر و لایه کنترل ابر با یک الگوریتم یادگیری ماشین برای زمان‌بندی استفاده می‌کند. نویسندگان در [۲۰]، یک الگوریتم ژنتیک سازگار مبتنی بر رمزگذاری صحیح را برای تصمیم‌گیری در مورد بهبود بارگذاری تک‌وظیفه‌ای، چندوظیفه‌ای و چندین بخش پیشنهاد کردند. در [۲۱]، روش تصمیم‌گیری مبتنی بر الگوریتم ژنتیک برای حل مسئله بارگذاری

موجود در شبکه از مسائل مهم در زمینه زمان‌بندی وظایف است [۸]. در این مقاله از لایه لبه میان ابر و برنامه‌های کاربردی دستگاه‌های سیار و در سمت MEC از تعدادی ماشین مجازی<sup>۱</sup> با تعدادی پردازنده استفاده می‌کنیم. وظایف می‌توانند به صورت محلی در تلفن همراه، اجرا یا برای اجرا به MEC ارسال شوند [۶]. وظایف دارای معیارهایی هستند که در این کار بر دو معیار انرژی مصرفی و زمان اجرا تأکید می‌کنیم [۳]. در این مقاله، داشتن دانش در رابطه با این دو معیار یکی از گزینه‌های مهم در انتخاب وظایف برای ارسال و همین‌طور زمان‌بندی آنها است. انتخاب بهترین مجموعه از وظایف در سمت تلفن همراه برای ارسال به سمت MEC با استفاده از الگوریتم کوله‌پشتی حریمانه (GKOA) انجام شده که هدف آن، کاهش انرژی مصرفی در سمت دستگاه است. زمان‌بندی پویای وظایف در سمت MEC بر اساس روش فازی انجام می‌شود. در این الگوریتم، انرژی مصرفی و زمان اجرا دو ورودی روش فازی هستند که اولویت وظایف برای اجرا، خروجی این روش است. زمان‌بندی با اولویت‌بندی وظایف مبتنی بر فازی (FPTS) از منطق فازی برای تبدیل داده‌های ورودی به متغیرهای استنتاجی استفاده می‌کند. منطق فازی، یک روش مناسب برای اولویت‌بندی وظایف نسبت به روش‌های کلاسیک مانند خروج به ترتیب ورود (FIFO)، نوبت گردشی (RR) و ابتدا کوتاه‌ترین کار (SJF) است [۹] و [۱۰]. ما روش پیشنهادی خود را با این روش برای پارامترهای مختلف از جمله زمان انتظار، سطح سرویس‌دهی و درصد تأخیر مقایسه می‌کنیم. در سمت لبه از یک سیاست تخصیص منابع بر اساس منابع فعال موجود در شبکه استفاده می‌کنیم که در صورت نبود منبع کافی مورد نیاز یکی از وظایف در شبکه، آن وظیفه برای اجرا به ابر ارسال می‌شود. به طور کلی هدف ما در این مقاله، ارسال مجموعه مناسب از وظایف از سمت دستگاه به لبه در جهت کاهش مصرف انرژی در دستگاه می‌باشد. همچنین هدف ما در سمت MEC، زمان‌بندی پویای وظایف با تخصیص کمترین منابع در جهت تحقق کاهش هزینه‌هاست. چنانچه یکی از کارها در لایه MEC اجرا نشد، برای اجرا به لایه بالاتر یعنی ابر ارسال می‌شود. اهداف اصلی این مقاله را می‌توان به صورت زیر خلاصه کرد:

- ارائه GKOA به عنوان یک الگوریتم بارگذاری مطلوب در مصرف انرژی هر وظیفه و افزایش سرعت آن در سمت دستگاه سیار به منظور تقسیم‌بندی وظایف در جهت بهبود مصرف منابع دستگاه، هدف اولیه این کار است.
- ارائه FPTS به عنوان یک روش بارگذاری پویای مبتنی بر دو معیار مصرف انرژی و زمان اجرا و همچنین اعمال یک سیاست تخصیص بهینه منابع با حداقل تعداد پردازنده‌ها جهت اولویت‌بندی وظایف و افزایش سرعت اجرای وظایف در سمت لبه، بخش مهم زمان‌بندی در این کار است.
- "تجزیه و تحلیل عددی نتایج ارزیابی برای پارامترهای زمان انتظار، احتمال تأخیر، سطح خدمات و سربار سیستم با حداقل تعداد پردازنده ها، در روش زمان‌بندی پیشنهادی و روشهای خروج به ترتیب ورود، نوبت گردشی و ابتدا کوتاهترین کار مقایسه می‌شود و میزان برتری

1. Virtual Machine
2. Greedy Knapsack Offloading Algorithm
3. Fuzzy-Based Priority Task Scheduling
4. First in-First out
5. Round Robin
6. Shortest Job First

7. Multi-User Computation Partitioning Problem
8. Performance-Resource-Load
9. Multi-Criteria Offloading Decision

[۳۵]، پیشنهاد یک الگوریتم تخلیه پویای محاسباتی (DCOA) را به عنوان یک مشکل بهینه سازی برای به حداقل رساندن هزینه بارگذاری پیشنهاد کردند. در [۳۶]، برنامه ریزی خطی صحیح تلفیقی (MILP) برای حل مشکل تخلیه محاسباتی چندکاربره پیشنهاد گردید. برای حل پیچیدگی محاسباتی فرموله شده، یک الگوریتم اختصاص منابع اکتشافی مکرر (IHRA) پیشنهاد شد تا تصمیمات تخلیه به صورت پویا با کمترین زمان اجرا و راندمان بالا انجام شوند. نویسندگان در [۳۷]، پیشنهاد یک الگوریتم بهبود سربار زمان بندی چندمنظوره (OOMSG) را برای به حداقل رساندن سربار دستگاه و تکمیل زمان بندی کار ارائه کردند. نویسندگان در [۳۸]، معماری محاسباتی لبه مبتنی بر شبکه تعریف نرم افزاری (SDN) را که زیرساختار نرم افزار (SDI) نامیده می شود، پیشنهاد دادند. آنها از OpenFlow و OpenStack برای مجازی سازی خدمات برای ساخت برنامه های هوشمند استفاده کردند. نویسندگان در [۳۹]، مکانیسم تخلیه انرژی محاسباتی (EECO) را برای به حداقل رساندن مصرف انرژی پیشنهاد دادند.

نویسندگان در [۴۰]، یک چارچوب نظارت زیستی را برای شناسایی و بومی سازی تهدیدات بیولوژیکی با پشتیبانی از محاسبات مه و لبه موبایل ارائه می کنند. گره های مه، داده های نظارتی را در سرورهای خود جمع می کنند و در افزایش سرعت اجرا و ذخیره سازی داده ها تأثیرگذار هستند. نویسندگان در [۴۱]، ساختاری را برای یک محیط لبه ارائه کردند که شامل ۳ لایه بود: لایه همکاری مهاجرت، لایه اشتراک محاسباتی و لایه محاسبات از راه دور. آنها الگوریتم های مسیریابی بهینه را برای توزیع وظایف (ORTD) و زمان بندی اولیه پیشنهاد کردند که مسیریابی بهینه را برای هر وظیفه و تخصیص منابع افزایش می دهد. همچنین ORTD، ترافیک کار و واریانس زمانی را برای هر کار و طول کارها کاهش داد. در [۴۲]، یک برنامه نظارت هوشمند برای تخلیه وظایف در لبه یا ابر یا اجرای محلی ارائه گردید. از روش های یادگیری تقویتی عمیق (DRL) و مقیاس گذاری خودکار حریصانه و بارگذاری مبتنی بر یادگیری تقویت عمیق (GASDEO)، برای انتخاب بهترین گره و تخلیه آنها استفاده شد. نتیجه این بود که مصرف انرژی، هزینه کل اجرا، استفاده از شبکه و تأخیر به ترتیب ۲، ۴، ۱ و ۰/۰۷ درصد نسبت به اجرای محلی، بهینه سازی شد. نویسندگان در [۴۳]، دو روش زمان بندی را بر اساس برنامه ریزی خطی اعداد صحیح ارائه کردند که وظایف را در ابر و لبه به صورت پویا زمان بندی می کرد. نتایج آنها نشان داد که الگوریتم آنها بهتر از روش های سنتی مانند RR کار می کند.

### ۳-۲ ابر لبه سیار

در [۴۴]، مدل تصمیم گیری مارکوف برای به دست آوردن یک سیاست مطلوب برای به حداقل رساندن هزینه های محاسبات و تخلیه پیشنهاد شد.

وظایف در محیط های چندسایتی پیشنهاد شد که باعث بهبود زمان اجرا و مصرف انرژی در MCC گردید. در [۲۲]، یک مدل بهینه سازی چندهدفه مبتنی بر الگوریتم مرتب سازی ژنتیک برای به حداقل رساندن مصرف انرژی در محیط ابری برای مسئله تخصیص وظایف چندهدفه ارائه شد که در مقایسه با سایر الگوریتم های سنتی، زمان تکمیل و مصرف انرژی را به طور قابل توجهی کاهش داده است.

### ۲-۲ محاسبات لبه سیار

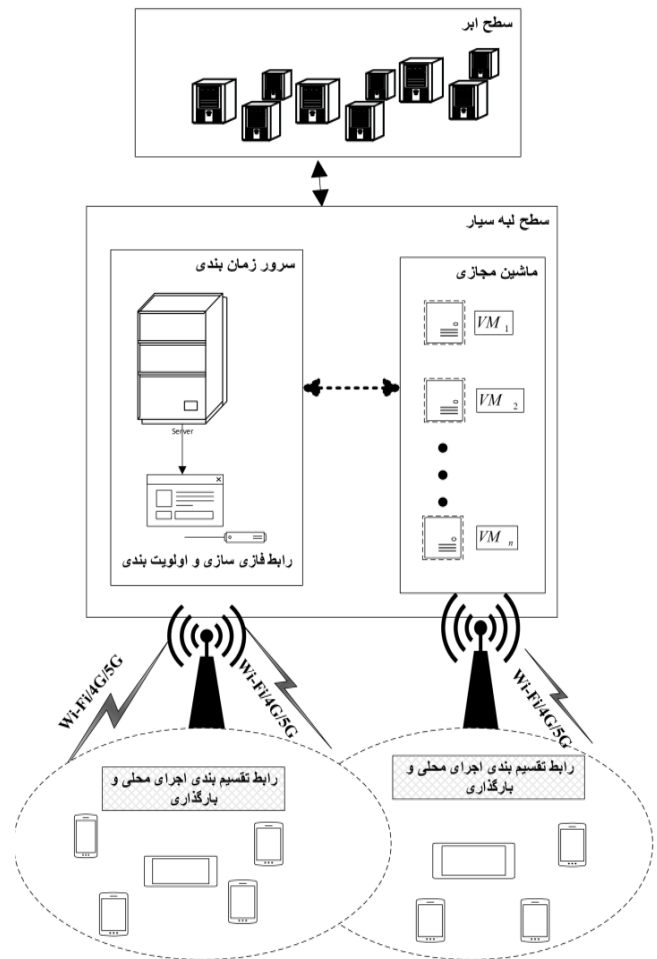
در [۲۳]، گره های لبه به عنوان واحدهای جاده ای در شبکه های حمل و نقل برای ارائه خدمات IoT به وسیله OpenM2M به منظور بهبود تأخیر و ترافیک در شبکه توسعه داده شد. نویسندگان در [۲۴]، یک الگوریتم بارگذاری را با استفاده از تکنیک های بهینه سازی تصادفی و بهینه سازی Lyapunov در لبه برای یک دستگاه پیشنهاد کردند. نتایج آنها نشان می دهد هزاران دستگاه IoT برای بارگذاری، انتخاب و باعث صرفه جویی در مصرف انرژی در دستگاه شدند. کار [۲۵]، یک مثال از زمان بندی در لبه است که حداقل، تأخیر متوسط ترافیک را در پایین صفحه LTE کاهش می دهد. نویسندگان در [۲۶]، مدل زمان بندی در ایستگاه پایه را با استفاده از تئوری شبکه صف مشکل از چندین نوع سرور ارائه دادند. کاربران در این مدل بر اساس اولویت، زمان بندی می شوند که این کار سرعت را در بارگذاری فایل های چندرسانه ای بهبود می بخشد. در [۲۷]، یک سیستم چندسروری بر اساس تکنیک مارکوف پیشنهاد شده و یک دستگاه، وظایف محاسباتی را به سرورهای متعدد برای به حداقل رساندن زمان اجرا و مصرف انرژی دستگاه ارسال می کند. در [۲۸]، یک مدل سرویس محاسباتی لبه بر اساس دروازه<sup>۱</sup> پیشنهاد شد که از طریق مجازی سازی باعث کاهش تأخیر، پهنای باند شبکه و افزایش سرعت انتقال در ابر شد. این مدل، عملکرد عملیاتی گره های لبه را بهبود بخشید. در [۲۹]، یک مشکل بهینه سازی بر اساس اتخاذ یک احتمال ارتقا (PBP) پیشنهاد شد تا هزینه انرژی و تراکم بسته ها را بهبود بخشد. آنها از یک الگوریتم بهینه سازی اکتشافی انبوه کریل برای به حداقل رساندن تراکم وظایف در صف استفاده کردند. محققان در [۳۰]، پیشنهاد دسترسی چندگانه غیر متعامد (NOMA) را برای بهینه سازی کارایی انتقال رادیویی با دسترسی چندگانه پیشنهاد کردند. در این کار، یک ساختار لایه ای با استفاده از یک الگوریتم لایه ای کارآمد استفاده شد تا زمان تأخیر را برای تک کاربر دستگاه کاهش دهد. نویسندگان در [۳۱]، یک مبادله بر اساس الگوریتم جستجوی تکرار بین مصرف انرژی و زمان تأخیر در شبکه MEC تک و چندسولوی برای کاهش این دو پارامتر پیشنهاد دادند. در [۳۲]، چارچوب محاسبات لبه سیار برای برنامه های حساس به تأخیر مورد استفاده قرار گرفت. آنها یک مدل صف بندی، یعنی تخلیه محاسباتی چندکاربره (MUMCO) را برای بهبود هزینه و کنترل زمان بندی انتقال اعمال کردند. در [۳۳]، یک الگوریتم مبتنی بر بازی ائتلافی برای بهبود تأخیر، قابلیت محاسبه و ذخیره سازی خدمات ناهمگن پیشنهاد شد. این میزان، تأخیر کاربران را به طور متوسط با مقدار ۲۷/۸٪ و ۸۲/۱٪ کاهش می دهد. نویسندگان در [۳۴]، یک الگوریتم بارگذاری محاسباتی توزیع شده را بر اساس نظریه بازی برای بهبود عملکرد بارگذاری محاسبات برای اندازه کاربری بالا طراحی کردند. نویسندگان در

1. Gateway
2. Promoted by Probability
3. Non-Orthogonal Multiple Access
4. Multi-User Mobile Computation Offloading

5. Dynamic Computation Offloading Algorithm
6. Mixed Integer Linear Programming
7. Iterative Heuristic MEC Resource Allocation
8. Overhead-Optimizing Multi-Device Scheduling Game
9. Software-Defined Network
1. Software-Defined Infrastructure 0
1. Energy-Efficient Computation Offloading
1. Optimal Routing for Task Dispatching
1. Deep Reinforcement Learning 3
1. Greedy Auto-Scaling Deep Reinforcement Learning-Based Offloading

جدول ۱: نمادها.

نماد	تعریف
$d(t)$	اندازه داده‌های ورودی پردازشی برنامه (به عنوان مثال، برنامه و داده‌ها)
$s(t)$	اندازه پردازش‌ها (به عنوان مثال، تعداد چرخه پردازنده)
$C_n^I$	ظرفیت پردازنده در اجرای محلی
$C_n^M$	ظرفیت پردازنده در اجرا در لبه
$e_{idle}$	انرژی مصرف‌شده دستگاه تلفن همراه
$e_{iran}$	انرژی مصرف‌شده برای انتقال
$e_{active}$	انرژی در دسترس دستگاه تلفن همراه
$C_{vm}(t)$	توانایی محاسبه پردازنده توسط $n$ امین ماشین مجازی
$h_n$	وضعیت تصمیم‌گیری برای بارگذاری
$H$	مجموعه وظایف برای بارگذاری
$I_n$	مجموعه وظایف اجراشده
$u_n^I$	توانایی اجرای محلی یک وظیفه
$u_n^M$	توانایی اجرای یک وظیفه در لبه
$m$	تعداد ماشین‌های مجازی
$n$	تعداد وظایف موجود در $i$ امین ماشین مجازی
$cap$	ظرفیت یک ماشین مجازی
$t_s$	شکاف زمانی
$\mu$	نرخ سرویس‌دهی
$\lambda$	نرخ ورود به صف
$\phi$	تابع هزینه
$\beta$	پارامتر وزن در تابع هزینه
$W$	ظرفیت کوله‌پشتی
$P$	مقدار ارزش هر وظیفه
$T_a$	زمان آغاز برای انتخاب یک پردازنده برای اجرا
$T_c$	زمان انتها برای اجرای یک وظیفه در یک پردازنده



شکل ۱: مدل سیستم پیشنهادی.

سطح ارسال می‌شوند. تصویر مدل سیستم پیشنهادی در شکل ۱ نشان داده شده است.

### ۳-۲ فرمول‌بندی مسئله

این بخش، نحوه ارسال و اجرای وظایف به لایه لبه را نشان می‌دهد. برای سهولت فهم، تعریف نمادهای مهم استفاده شده روابط در جدول ۱ ذکر گردیده است.

### ۳-۳ بارگذاری بهینه

به فرض، زمان به برش‌های زمانی  $t$  تقسیم می‌شود که به صورت  $M \in \{0, 1, \dots, n\}$  هستند. مجموعه اولیه وظایف،  $t \in \{1, \dots, n\}$  است و جایی که  $M = 0$  باشد، یعنی هیچ وظیفه‌ای در مجموعه وجود ندارد. مجموعه  $H \in \{h_i\}$ ، مجموعه‌ای از وظایف برای اجرا در سمت تلفن همراه می‌باشد. ما  $k(t) \in \{0, 1\}$  را به عنوان پارامتر تصمیم‌گیری نشان می‌دهیم. اگر  $k(t) = 0$  باشد، یعنی وظایف به لبه ارسال می‌شوند و اگر  $k(t) = 1$  باشد، یعنی وظایف به صورت محلی در دستگاه تلفن همراه اجرا می‌گردند. مجموعه پاسخ بهینه به صورت  $X$  نشان داده می‌شود و  $ew[i]$  وزن وظیفه  $i$ ام را نشان می‌دهد. همچنین  $e[i]$  سودمندی وظیفه  $i$ ام را نشان می‌دهد. در الگوریتم ما مصرف انرژی مطابق با وزن وظایف آنها و سود وظایف، مقداری ثابت است. رابط GKOA مجموعه‌ای با بالاترین وزن را برای بارگذاری به لایه ۲ انتخاب می‌کند.

نویسندگان در [۴۵] از تئوری صف‌بندی با مدل  $M/M/m$  برای حرکت گره‌های لبه در شبکه وسیله نقلیه استفاده کردند که میانگین پاسخ در زمان بارگذاری را به حداقل می‌رساند. نویسندگان در [۴۶]، یک الگوریتم زمان‌بندی را برای وظایف تصادفی در جهت بهبود زمان‌بندی و عملکرد با داشتن کامل در شیوه سلسله‌مراتبی پیشنهاد کردند. در [۴۷]، سیستم جمع‌آوری دستگاه به دستگاه (D2D) در محاسبات ابر لبه‌های سیار پیشنهاد شد که مصرف انرژی و تخلیه موازی را بهبود می‌بخشد.

### ۳- روش پیشنهادی

در این بخش، مدل بارگذاری بهینه و زمان‌بندی پویا برای به حداقل رساندن استفاده از منابع محدود در سمت تلفن همراه و لبه و همچنین افزایش سرعت و کارایی اجرای وظایف، معرفی و سپس مدل‌سازی مسئله در سیستم ارائه شده است.

#### ۳-۱ مدل سیستم

سیستم ما از ۳ سطح تشکیل شده است. در پایین‌ترین سطح، چندین دستگاه تلفن همراه وجود دارند و رابط GKOA در این سطح، مسئول انتخاب بهترین مجموعه از وظایف برای ارسال به لبه می‌باشد. در سطح دوم، رابط FPTS مسئولیت تصمیم‌گیری و اولویت‌بندی وظایف را برای اجرا دارد و تخصیص منابع نیز در این سطح انجام می‌شود. اگر منبع فعالی جهت تخصیص در این سطح وجود نداشت، وظایف به ابر در بالاترین

جدول ۲: قوانین فازی بر اساس زمان اجرا و مصرف انرژی.

انرژی مصرفی		زمان اجرا			
بسیار بالا	بالا	متوسط	پایین	بسیار پایین	
بسیار بالا	بالا	متوسط	پایین	بسیار پایین	بسیار بالا
بالا	بالا	متوسط	پایین	بسیار پایین	بالا
بالا	متوسط	متوسط	پایین	بسیار پایین	متوسط
بالا	متوسط	پایین	بسیار پایین	بسیار پایین	پایین
متوسط	پایین	پایین	بسیار پایین	بسیار پایین	بسیار پایین

در (۴)، مقادیر  $c_i$  و  $\delta_i$  مرکزیت تابع و  $x$  مجموعه‌ای از دو ورودی انرژی مصرفی و زمان اجرا است که عرض  $i$  امین مجموعه فازی  $A^i$  می‌باشد. خروجی تابع عضویت  $\mu_{A^i}(x) \in (0, 1]$  است. توابع عضویت فازی گاوسی در ادبیات منطق فازی رایج هستند، زیرا آنها برای ارتباط بین سیستم‌های فازی و شبکه‌های عصبی با استفاده از تابع پایه شعاعی (RBF) مورد استفاده قرار می‌گیرند. قوانین استفاده شده در سیستم ما در جدول ۲ نشان داده شده است. خروجی سیستم تعیین اولویت بر اساس ورودی‌های زمان اجرا و مصرف انرژی به عنوان خروجی سیستم فازی است. عناصر مقیاس در مجموعه  $A$ ، {بسیار زیاد، زیاد، متوسط، پایین، بسیار پایین} هستند و خروجی توسط ضرب مجموعه‌های فازی ارائه شده است. فرض می‌کنیم که دو مجموعه  $A$  و  $B$  را به عنوان ورودی داریم. ضرب مجموعه‌های فازی با  $A \cap B$  نشان داده شده که به صورت  $\mu_{A \cap B}(X) = \min(\mu_A(X), \mu_B(X))$  می‌باشد. ماشین مجازی، اطلاعات و برنامه‌های کاربردی یک دستگاه هوشمند را زمانی که یک درخواست از طرف کاربر وارد می‌شود، کلون و در هر  $t$  برنامه کنترل، بهترین ماشین را برای سرویس‌دهی انتخاب می‌کند. این برنامه با اندازه محاسبات و داده بالا بهترین گزینه برای بارگذاری است.  $d(t)$  برای توصیف اندازه داده ورودی برنامه استفاده می‌شود و  $S(t)$  اندازه محاسبات را توصیف می‌کند. هر دو بردار به صورت مستقل و یکسان در هر شکاف زمانی  $t$  توزیع شده‌اند. زمان اجرا در لبه از روابط زیر محاسبه می‌شود

$$T_{MEC}(t) = T_r + T_h + T_{tr} \quad (5)$$

$$T(t) = \sum_{n \in \mathcal{G}} k(t) T_{MEC} + [1 + \sum_{n \in \mathcal{G}} k(t) T_{local}] \quad (6)$$

که در آن، مقدار  $T_h(t) = s(t)/c_{vm}(t)$  مدت زمانی است که طول می‌کشد تا یک درخواست در ماشین مجازی  $n$  ام بررسی شود و  $T_r(t)$  زمان پاسخ به اضافه زمان بارگذاری درخواست از ماشین مجازی است. مقدار  $T_{tr}(t) = d(t)/c_{vm}(t)$ ، زمان انتقال وظیفه برای بارگذاری در ماشین مجازی در شکاف زمانی  $t$  و مقدار  $T_{local}(t) = d(t)/c_{local}(t)$  در (۶)، زمان اجرای وظیفه به صورت محلی در دستگاه است [۴۴]. ما فرض می‌کنیم که  $k(t) \in \{0, 1\}$ ، احتمال اجرای یک وظیفه به صورت محلی یا بارگذاری در لبه را نشان دهد. اگر مقدار  $k(t) = 0$  باشد، وظیفه به صورت محلی در دستگاه تلفن همراه اجرا می‌شود و در غیر این صورت، وظیفه با بارگذاری به لبه اجرا می‌شود. برای محاسبه انرژی مصرفی، مقدار  $E_{MEC}(t) = E_{tr}(t) + E_{vm}(t)$  را تعریف می‌کنیم که انرژی مصرفی هنگام انتقال وظیفه و اجرا در ماشین مجازی را نشان می‌دهد [۵۰]. در این رابطه  $E_{tr} = T_{tr} \times e_{tran}$  و  $E_{vm} = T_h \times e_{idle}$  کل انرژی مصرفی از رابطه زیر به دست می‌آید

مجموع وزن وظایف، کمتر یا مساوی ظرفیت کوله‌پشتی الگوریتم ما است که با  $W$  نشان داده می‌شود. مقدار  $0 \leq x_i \leq 1$  برای هر وظیفه تعریف می‌شود به طوری که  $x_i \times ew[i]$ ، وزن اشغال شده از هر وظیفه در کوله‌پشتی و  $x_i \times p_i$  نیز ارزش هر وظیفه را نشان می‌دهد [۴۸]. هدف الگوریتم GKO در (۱) و (۲) نشان داده شده است

$$\max \sum_{i=1}^n (x_i \times p_i) \quad (1)$$

$$\sum_{i=1}^n x_i \times ew[i] \leq W \quad (2)$$

$$p_i = \frac{e[i]}{ew[i]} \quad (3)$$

در (۳)،  $p_i$  ارزش وظیفه  $i$  ام، آرایه  $p \in [1, 2, \dots, n]$ ، مجموعه ارزش تمام وظایف انتخاب شده برای بارگذاری و  $e[i]$ ، سودمندی وظیفه  $i$  ام را نشان می‌دهد.

### ۳-۴ زمان بندی پویا

برای سودمندی بهتر، زمان بندی بر اساس صف اولویت انجام می‌گردد که اولویت این وظایف از خروجی الگوریتم FPTS گرفته می‌شود. ما قصد داریم که درباره پارامترهای انرژی مصرفی و زمان اجرا برای افزایش قابلیت اطمینان برای محیط لبه مؤثر بحث کنیم. سیستم فازی ما از سه سطح تشکیل گردیده است: سطح فازی سازی، سطح استنتاج و سطح تخریب پذیری فازی. هر وظیفه با دو پارامتر به صورت وظیفه (پارامتر انرژی مصرفی و زمان اجرا)، ورودی سطح اول است. خروجی سطح فازی، ورودی سطح استنتاج است و نهایتاً خروجی سطح استنتاج، ورودی سطح سوم است. مقادیر ورودی در سطح اول تبدیل به داده‌های فازی می‌شوند. قوانین فازی در سطح دوم تعریف می‌شوند و نهایتاً سطح تخریب پذیری فازی، مجموعه‌های فازی را در فرم زبانی برای ارائه خروجی خالص تبدیل می‌کند. ورودی سیستم به صورت  $M(ct \times et)$  نشان داده می‌شود که در واقع مجموعه وظایف با دو پارامتر زمان اجرا و انرژی مصرفی هستند و به ترتیب به صورت  $ct$  و  $et$  می‌آیند. خروجی سیستم به فرم آرایه  $P[1, 2, \dots, n]$  نشان داده می‌شود، به طوری که  $task_i^p$  نشان دهنده آن است که این وظیفه، اولویت  $p$  دارد. به منظور تبدیل پارامترها به پارامترهای قابل استنتاج در سیستم فازی، قوانینی را در سطح استنتاج تعیین کرده که مجموعه‌ای از قوانین if...else هستند [۴۹]. فازی کردن ورودی‌ها با استفاده از تابع عضویت گاوسی تعیین می‌گردد که در (۴) نشان داده شده است

$$\mu_{(A^i)}(x) = \exp \frac{(c_i - x)^2}{2\sigma_i^2} \quad (4)$$

- الگوریتم بهینه بارگذاری GKOA در سمت دستگاه
- الگوریتم زمان‌بندی پویای FPTS در سمت لبه
- الگوریتم تخصیص منابع در سمت لبه

جزئیات الگوریتم‌ها به تفصیل در زیربخش‌های زیر آمده است. شکل ۲ روند اجرای کار و تقسیم‌بندی وظایف را در روندنمایی ساده نشان می‌دهد.

#### ۴-۱ الگوریتم بارگذاری به لبه GKOA

جزئیات مراحل الگوریتم GKOA در شکل ۳ آمده است. ما مجموعه وظایف  $n$  تایی از  $\{1, \dots, n\}$  را داریم. ابتدا وظایف را بر اساس مقدار ارزش و سود مرتب می‌کنیم؛ به طوری که  $p[1] \geq p[2] \geq \dots \geq p[n]$  و مقدار  $p_i$  برای  $i = 1, 2, \dots, n$  از رابطه  $e[i]/ew[i]$  محاسبه می‌شود. در واقع بر اساس وزن نیز مقدار  $p[i]/e[i]$  به صورت نزولی مرتب می‌شود که در نتیجه  $p[i+1]/e[i+1] \leq p[i]/e[i]$  است. با توجه به این توضیحات، وظایف با درصد انرژی مصرفی بیشتر برای بارگذاری انتخاب می‌شوند. در هر تکرار یک وظیفه از میان وظایف انتخاب می‌گردد، در صورتی که مقدار  $ew[i] \leq W$ ، آن گاه  $x[i] = 1$  و  $W = W - ew[i]$  می‌شود. مجموعه  $X$  آرایه جواب می‌باشد که شامل وظایف با بیشترین ارزش‌ها است. با توجه به این که وظایف در ابتدا مرتب شدند، در نتیجه یک حلقه for الگوریتم با پیچیدگی  $O(n)$  اجرا می‌شود و همراه با مراحل دیگر الگوریتم پیچیدگی  $O(n \log n)$  دارد.

#### ۴-۲ الگوریتم زمان‌بندی FPTS

این بخش بر مراحل انجام الگوریتم FPTS تمرکز دارد. شکل ۴، مراحل FPTS را نشان می‌دهد. ما مجموعه‌ای از قوانین را تعریف کرده و سپس اولویت‌بندی وظایف بر اساس استنتاج فازی تعریف می‌شود [۴۹]. برای هر وظیفه با انرژی مصرفی و زمان اجرای بسیار بالا، اولویت، بالا و با انرژی مصرفی و زمان اجرای بسیار پایین، اولویت، بسیار پایین است. اولویت وظایف از ضرب اولویت دو مقدار انرژی مصرفی و زمان اجرا به دست می‌آید که به طور نمونه، بسیار پایین  $\times$  بسیار بالا برابر با اولویت پایین می‌شود. به این معنی که وظایفی که انرژی مصرفی بسیار پایین و زمان اجرای بسیار بالا دارند، دارای اولویت پایین هستند. این قوانین در جدول ۲ ارائه شدند. برای تخصیص منابع، هر درخواستی که از سمت یک وظیفه می‌آید شامل سه پارامتر حافظه، پردازنده و فضای داده است که این مقادیر مورد نیاز با مقادیر موجود مقایسه می‌شوند. سه مجموعه از ماشین‌های مجازی  $V_1\{1, 2, \dots, n\}$ ،  $V_2\{1, 2, \dots, n\}$  و  $V_3\{1, 2, \dots, n\}$  به ترتیب شامل مجموعه ماشین‌های مجازی دارای حافظه، پردازنده و فضای داده کافی هستند. اشتراک این سه مجموعه شامل مناسب‌ترین ماشین‌ها برای انتخاب است. تصمیم‌گیری برای انتخاب ماشین در دسترس تحت ۲ سیاست زیر انجام می‌شود:

(۱) همه ماشین‌های مجازی مشغول هستند و  $V_1 \cap V_2 \cap V_3 \neq \emptyset$  که

در این صورت وظیفه به ابر مهاجرت می‌کند.

(۲) بیش از یک ماشین مجازی فعال وجود دارد و  $V_1 \cap V_2 \cap V_3 = \emptyset$

که بر اساس این سه پارامتر تخصیص انجام می‌شود.

مجموعه  $X$ ، ورودی این الگوریتم است که از خروجی الگوریتم GKOA می‌آید. همه وظایف به رابط FPTS، وارد و وظایف به ورودی‌های فازی بر اساس زمان اجرا و انرژی مصرفی تبدیل می‌شوند. پس از اولویت‌بندی وظایف، تخصیص منابع انجام می‌گردد که برای هر وظیفه، حداقل مقدار منابع لازم برای اجرا، نسبت به منابع موجود باید ۱ یا بیشتر از ۱ باشد. یعنی این روابط باید برقرار باشند:  $c_{vm}^k/s_i \geq 1$ ،  $cap_{vm}^k/d_i \geq 1$  و

$$E(t) = \sum_{n \in \mathcal{G}} k(t)E_{MEC} + [1 + \sum_{n \in \mathcal{G}} k(t)E_{local}] \quad (7)$$

در (۷)،  $E_{local} = T_{local} \times e_{active}$  مقدار انرژی مصرفی در دستگاه هوشمند است. وظایف با بیشترین زمان اجرا و مصرف انرژی بالا به لبه، منتقل می‌شوند و بهترین ماشین مجازی برای اجرای آنها انتخاب می‌گردد. ما چند پارامتر برای ارزیابی FPTS و مقایسه آن با سه روش زمان‌بندی دیگر در نظر می‌گیریم. نتیجه این مقادیر به عنوان پارامترهای ارزیابی در بخش ۵ ارائه می‌شوند. در سمت لبه، هدف این است که بهترین عملکرد را با حداقل تعداد پردازنده‌ها داشته باشیم. زمان انتظار، زمان متوسط انتظار یک وظیفه در صف و احتمال تأخیر برابر با نسبت وظایفی است که به صف انتظار می‌پیوندند. ما یک مقدار آستانه برای سطح سرویس را در نظر می‌گیریم. حداقل تعداد پردازنده‌های مورد نیاز، مقداری است که به سطح سرویس برابر یا بالاتر از هدف آستانه برسیم. هنگامی که وظایف به لبه می‌رسند تا زمانی که اجرا شوند، زمان انتظاری را طی می‌کنند که از رابطه زیر محاسبه می‌شود

$$W_c = T_c - T_a \quad (8)$$

که در (۸)، مقادیر  $T_c$  و  $T_a$  به ترتیب زمان رسیدن یک وظیفه و زمان تخصیص یک پردازنده برای اجرا است. تعداد درخواست‌های رسیده، نشان‌دهنده تعداد درخواست‌های فرستاده‌شده از هر وظیفه است که این اندازه‌گیری می‌تواند میزان ترافیک شبکه را نشان دهد. نرخ درخواست پایین به این معنی است که ترافیک کمتری در شبکه وجود دارد و نرخ درخواست بالاتر به معنای ترافیک بیشتر در شبکه است [۵۱]. نرخ درخواست ورود با  $\lambda$  نشان داده می‌شود. زمانی که ترافیک در شبکه متناسب باشد، سیستم در حالت تعادل قرار دارد که فاکتور تعادل سیستم از رابطه زیر به دست می‌آید

$$L = \frac{n_i}{cap_i} \quad (9)$$

در (۹)،  $n_i$  تعداد وظایف موجود در  $i$  امین ماشین مجازی و مقدار  $cap_i$  ظرفیت آن ماشین مجازی است. هنگامی که سیستم از حالت تعادل خارج می‌شود، بار سیستم افزایش می‌یابد. مقدار بار سیستم از رابطه زیر به دست می‌آید [۵۲]

$$d = \sqrt{\frac{\sum_{i=1}^m (L_i - L_{avg})^2}{m}} \quad (10)$$

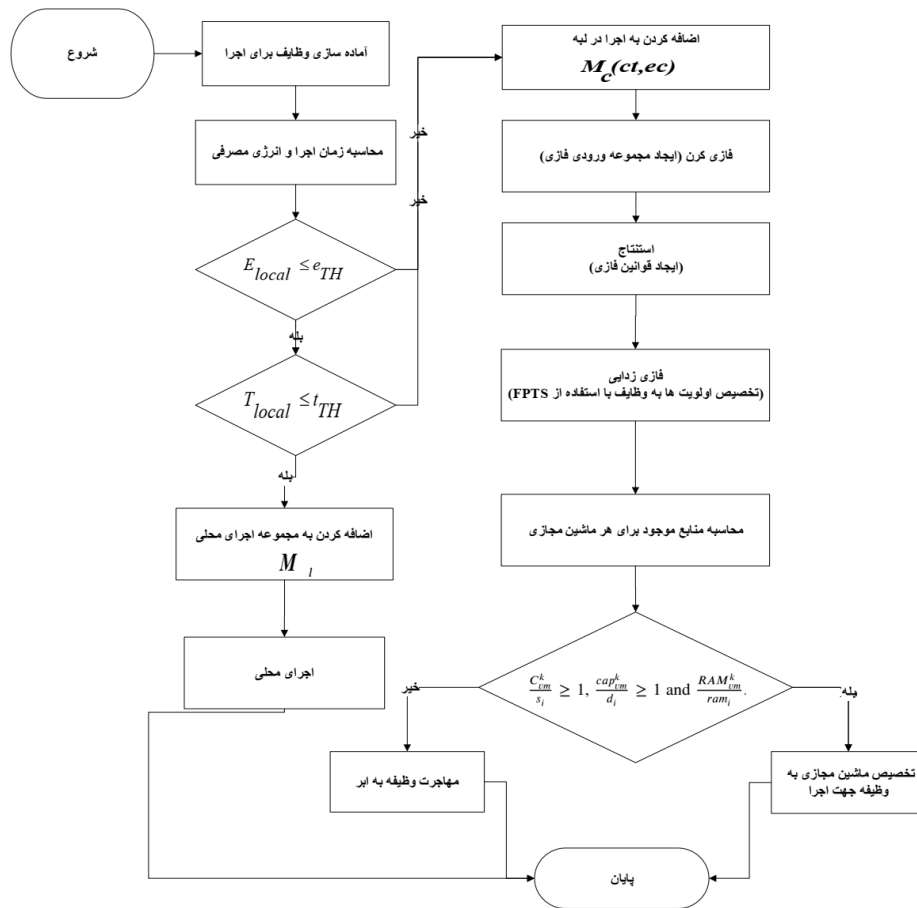
که  $m$  تعداد ماشین‌های مجازی و  $L_{avg}$  میانگین بار روی ماشین مجازی را نشان می‌دهد.

برای محاسبه تابع هزینه، یک مقدار را به عنوان پارامتر وزن در نظر می‌گیریم. پارامتر  $\beta$  به عنوان پارامتر وزن که در محدوده  $[0, 1]$  قرار دارد تعریف می‌شود. اگر  $\beta = 1$  باشد، مصرف انرژی در تصمیم‌گیری برای بارگذاری به لبه تأثیر ندارد. تابع هزینه برای مصرف انرژی و زمان اجرا به شرح زیر است

$$\phi(t) = \beta \frac{T(t)}{T_{local}(t)} + \beta \frac{E(t)}{E_{local}(t)} \quad (11)$$

#### ۴- روش پیشنهادی

الگوریتم بارگذاری و زمان‌بندی بهینه در دو سمت دستگاه و لبه در این بخش ارائه می‌شوند. در سمت لبه، تخصیص منابع نیز انجام می‌شود. در نتیجه مدل ارائه‌شده ما با سه الگوریتم در سه مرحله ارائه می‌گردد:



شکل ۲: روندنمای تقسیم بندی وظایف.

**Require:** Work flow consisting offloading task as  $M_c\{0,1,\dots,M_c\}$

and set of VM as  $V \in \{0,1,\dots,m\}$

**Ensure:** Priority set of tasks  $P\{1,\dots,p\}$  and  $V_1 \cap V_2 \cap V_3$

Ready task

**for**  $i=0$  to  $M_c$  **do**

Fuzzification  $task_i$  according to completion time and energy consumption with gaussian membership function.

Determine prioritize  $task_i$  based on fuzzy inference Rules

Defuzzification the output.

**for**  $k=0$  to  $m$  **do**

**if**  $\frac{C_{vm}^k}{s_i} \geq 1$  **then**

$V_1 \rightarrow vm^k$

**if**  $\frac{RAM_{vm}^k}{ram_i} \geq 1$  **then**

$V_2 \rightarrow vm^k$

**if**  $\frac{cap_{vm}^k}{d_i} \geq 1$  **then**

$V_3 \rightarrow vm^k$

**if**  $(V_1 \cap V_2 \cap V_3 \neq \emptyset)$  **then**

Select  $(V_1 \cap V_2 \cap V_3)$  and assign to  $task_i$

**if**  $(V_1 \cap V_2 \cap V_3 = \emptyset)$  **then**

Task migrates to Cloud

Analyzes all task in work flow has been executed

**if** All work flow did not execute in MEC **then**

Task migrates to Cloud

**end for**

**end for**

**Input:**  $n\{1,2,\dots,n\}, ew\{1,\dots,n\}, e\{1,\dots,n\}, W$

**Output:**  $X$  and  $W$

$x[i] \leftarrow 0$

$ew[i] \leftarrow 0$

Calculate  $p_i$  according to "(3)"

Sort the tasks by decreasing order according to  $p_i$

**for**  $i=n; i \geq 0; i--$  **do**

**for**  $k=0; k \leq i; k++$  **do**

**if**  $e[k-1] > e[k]$  **then**

Swap  $x[i]=temp$

Swap  $(ew[k-1], ew[k])$

Swap  $(p_{k-1}, p_k)$

**end if**

**end for**

**end for**

**for**  $i=1$  to  $n$  **do**

**if**  $ew[i] < W$  **then**

$temp = ew[i]$

$max\ value = max\ value + e[i]$

$x[i] = temp$

**else**

$temp = W$

$x[i++] = temp$

$W = W - temp$

**end if**

**end for**

**return**  $X$

شکل ۳: الگوریتم بارگذاری GKOA.

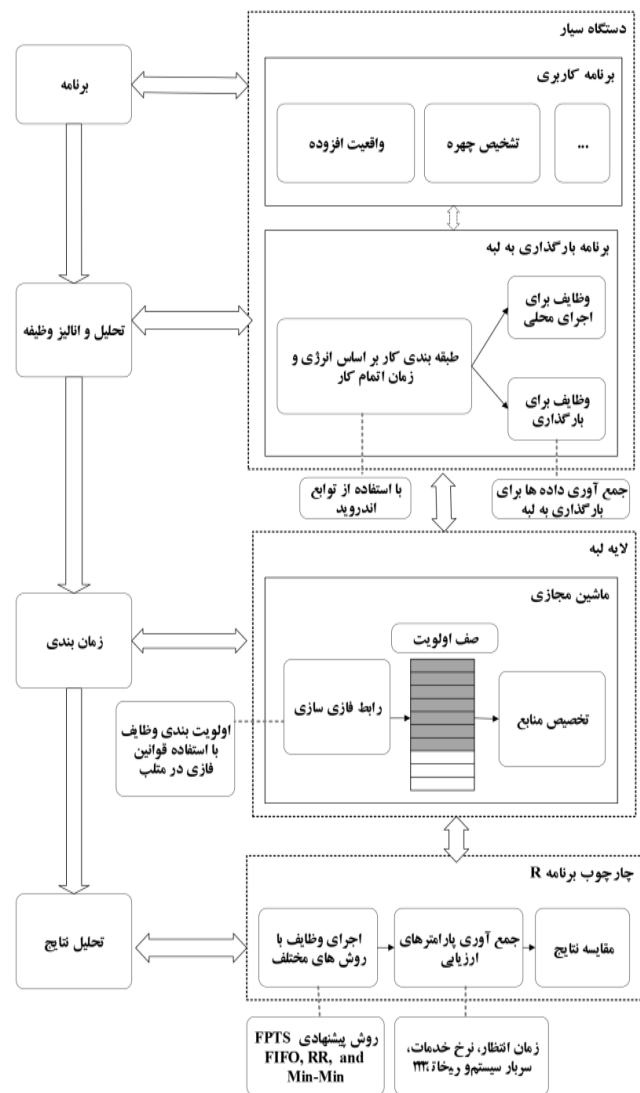
شکل ۴: الگوریتم زمان بندی FPTs.

$RAM_{vm}^k / ram_i \geq 1$  هر ماشین مجازی که یکی از این شرایط را داشته باشد، انتخاب و نهایتاً مجموعه اشتراک این ماشین ها به عنوان منابع

است که در ابتدا مقدار باتری ۹۰٪ بود. قدرت باتری غیر فعال، فعال و برای انتقال به ترتیب ۷۹، ۱/۵ و ۲/۲ درصد است. ما از طریق شبکه Wi-Fi ارتباط میان لبه و دستگاه را ایجاد می‌کنیم. دو نوع برنامه کاربردی در دستگاه هوشمند اجرا می‌شوند. یک برنامه کاربردی وجود دارد که GKOA را اجرا می‌کند. کارها از برنامه‌های واقعیت افزوده و تشخیص چهره که هر کدام دارای دو معیار میزان انرژی مصرفی و زمان اجرا هستند، گرفته می‌شود. این معیارها از جمله انرژی مصرفی از توابع اندرویدی در کد اندروید برنامه محاسبه شده‌اند. برنامه کاربردی داخل دستگاه سیار، الگوریتم بارگذاری GKOA را اجرا و کارها را جهت اجرای محلی یا بارگذاری، دسته‌بندی می‌کند. کارها سپس به ماشین‌های مجازی که به عنوان سطح لبه در معماری ما در نظر گرفته شده‌اند، ارسال شده و به محض رسیدن به لبه، توسط برنامه‌ای که الگوریتم زمان‌بندی FPTS را اجرا می‌کند، اولویت‌بندی و به ترتیب اولویت بالا به پایین اجرا می‌شوند. پارامترهای ارزیابی از جمله زمان انتظار، تأخیر و نرخ سرویس‌دهی توسط دستگاه و ماشین مجازی مانیتورینگ می‌شوند و پس از جمع‌آوری، آنها را در نرم‌افزار R Studio با زبان کدنویسی C تجزیه و تحلیل می‌کنیم. بخش فازی این سیستم نیز در نرم‌افزار MATLAB انجام می‌شود. ما مقایسه نتایج تحقیقات خود را در نرم‌افزار R انجام دادیم. همچنین پارامترهای ارزیابی یک کار زمان‌بندی و اولویت‌بندی را با استفاده از کارهای تحقیقاتی پیشین استخراج کردیم و برای مقایسه با نتایج کار خود از جدول و نمودار در این نرم‌افزار کمک گرفتیم. رشد صعودی و نزولی نتایج تحقیقات ما با مقایسه عددی نیز در نمودارها مشخص است. در اکثر قسمت‌ها برای نتایج به دست آمده از جدول عددی نیز برای نشان دادن جزئی کار استفاده کردیم. لازم به ذکر است که از بخش‌هایی کار که میزان استفاده از منابع دستگاه سیار را نشان می‌دهد، با استفاده از فراخوانی توابع اندرویدی، در کد برنامه اندروید، هنگام بارگذاری وظایف استفاده کردیم. در محاسبه تابع هزینه، مقدار  $\beta = 0.5$  را تنظیم می‌کنیم. مقدار ظرفیت در الگوریتم GKOA با ظرفیت ۲۰۰ و ۵۰۰ تنظیم شده است. مراحل ساختار کلی کار ارائه شده برای روشی و وضوح کار به طور کامل در شکل ۵ آمده‌اند.

## ۲-۵ تجزیه و تحلیل نتایج

در این بخش، ابتدا مسئله بارگذاری را با توجه به الگوریتم شکل ۳ ارزیابی می‌کنیم. وضعیت الگوریتم پیشنهادی GKOA نسبت به چند وضعیت دیگر بررسی می‌شود. همچنین این الگوریتم را برای موردی که در آن وظایف به صورت تصادفی انتخاب می‌شوند و مرتب نمی‌شوند، پیاده‌سازی می‌کنیم. سپس الگوریتم زمان‌بندی را در سمت لبه با توجه به الگوریتم شکل ۴ تجزیه و تحلیل می‌کنیم و مدل پیشنهادی را با چندین الگوریتم معیار مقایسه می‌نماییم. FPTS با سه الگوریتم RR، FCFS و SJF مقایسه می‌شود. در SJF وظایف به ترتیب ورود در صف، اجرا می‌شوند. در RR هر وظیفه در یک برش زمانی اختصاص داده شده، اجرا می‌شود. SJF یک صف زمان‌بندی است که وظایف با کمترین زمان اجرا را برای اجرای بعدی انتخاب می‌کند [۹] و [۱۰]. توابع عضویت الگوریتم FPTS با توجه به مقادیر ورودی زمان اجرا و انرژی مصرفی در شکل ۶ آمده‌اند. نهایتاً قسمت تخصیص منابع را بررسی کرده و تعیین می‌نماییم که چند درصد از وظایف به سمت ابر منتقل می‌شوند. الگوریتم GKOA پیشنهاد شده را در دو حالت ظرفیت ۲۰۰ و ۵۰۰ با حالت اجرای تمام وظایف به صورت محلی، بارگذاری تمام وظایف به لبه و روش PBP [۲۹] جایی که  $\gamma = 0.2$  است، مقایسه می‌کنیم. شکل ۷-الف مصرف انرژی را



شکل ۵: سیستم و ساختار مراحل کار پیشنهادی.

مناسبی برای تخصیص انتخاب می‌شود. در صورتی که هیچ منبع فعالی وجود نداشته باشد، وظایف به ابر ارسال می‌گردند.

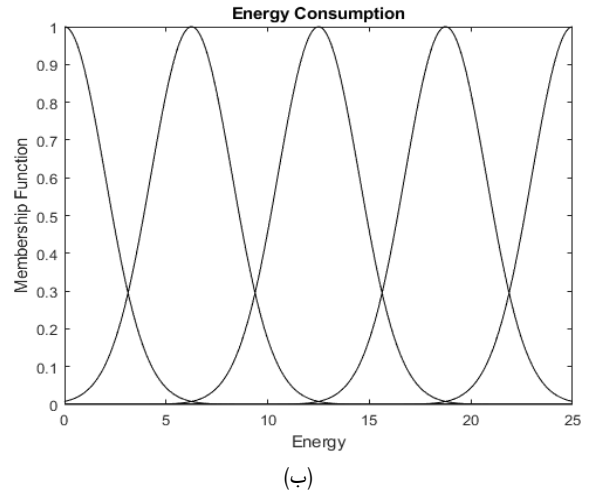
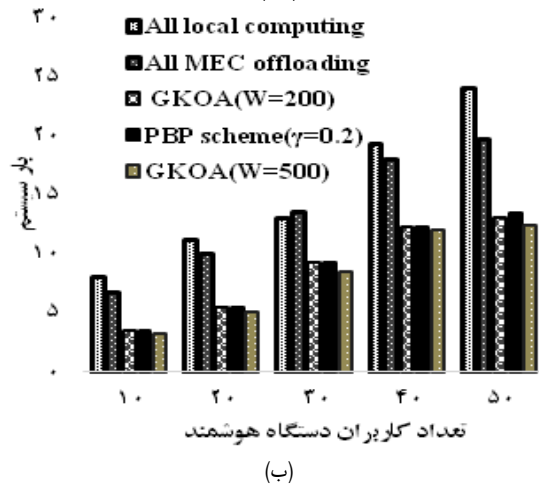
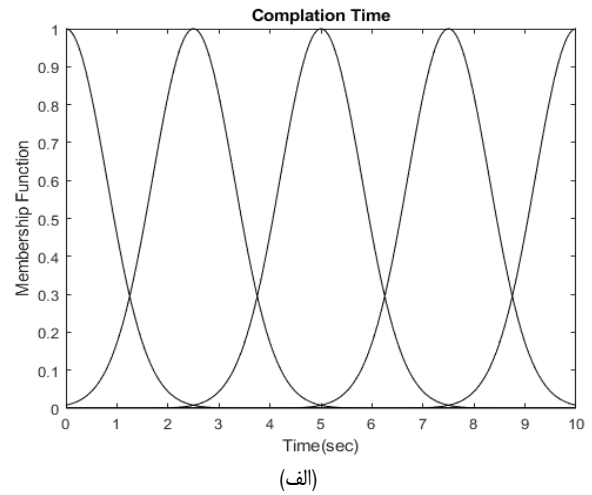
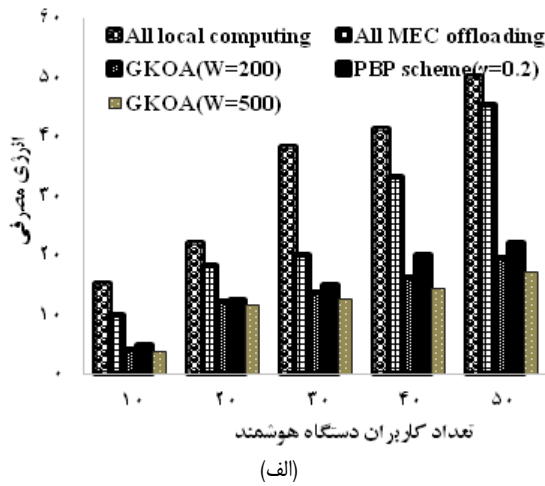
## ۵- ارزیابی

در این بخش، جزئیات مدل سیستم، ارائه و نتایج عددی الگوریتم‌های استفاده شده، بررسی و تجزیه و تحلیل گردیده و در ۲ بخش محیط ارزیابی و تجزیه و تحلیل نتایج آمده‌اند.

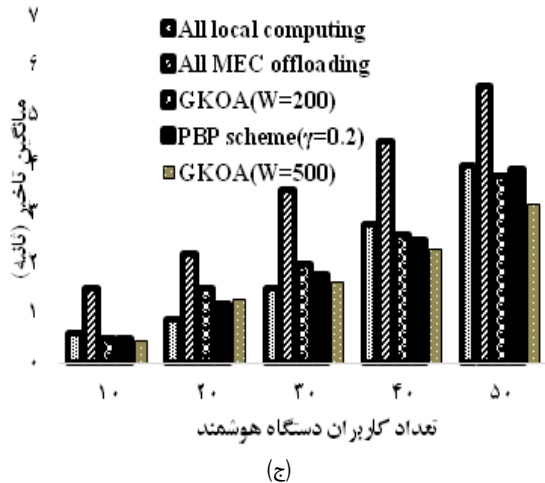
### ۱- محیط ارزیابی

در این بخش، ارزیابی روش‌های بارگذاری، زمان‌بندی و تخصیص منابع در سیستم پیشنهاد شده انجام می‌شود. ما عملکرد مدل پیشنهادی را با استفاده از یک کامپیوتر ۱۰ هسته اینتل R در فرکانس ۳٫۶ گیگاهرتز و RAM ۸ گیگابایت ارزیابی می‌کنیم. سیستم عامل ویندوز ۱۰ از نوع ۶۴ بیت است. ما از دو ماشین مجازی به عنوان گره‌های لبه در لبه و یک دستگاه تلفن هوشمند استفاده می‌کنیم. هر ماشین با حافظه ۲ مگابایتی و پردازنده ۵ هسته کار می‌کند. کل هارددیسک ۱ ترابایت است که هر ماشین مجازی با فضای ۲۰۰ گیگابایت کار می‌کند. مدل دستگاه هوشمند سامسونگ F۷۲۰ همراه با پردازنده ۸ هسته با قدرت ۲ گیگاهرتز و RAM ۲۸۱۵ مگابایت و رابط شبکه ۴G و Wi-Fi است. سیستم عامل تلفن هوشمند از نوع نسخه اندروید ۸ و ظرفیت باتری ۳۶۰۰ مگاهرتز





شکل ۶: تابع عضویت فازی مجموعه‌های ورودی.

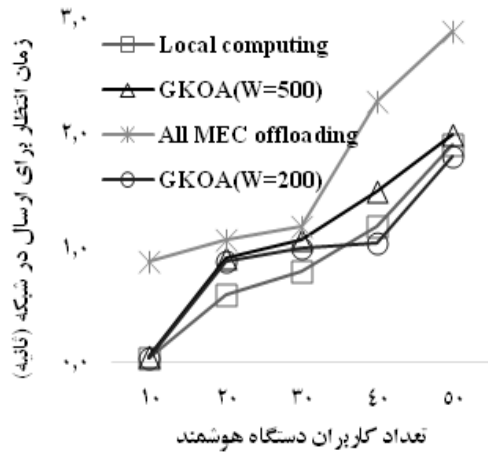


شکل ۷: مقایسه تعدادی از پارامترها برای تعداد مختلف کاربران در ۵ حالت روش پیشنهادی با ظرفیت ۵۰۰ و ۲۰۰، روش PBP، اجرای محلی و بارگذاری تمام وظایف به لبه، (الف) تأثیر تعداد کاربران بر انرژی مصرفی، (ب) تأثیر تعداد کاربران بر هزینه سیستم و (ج) تأثیر تعداد کاربران بر تاخیر.

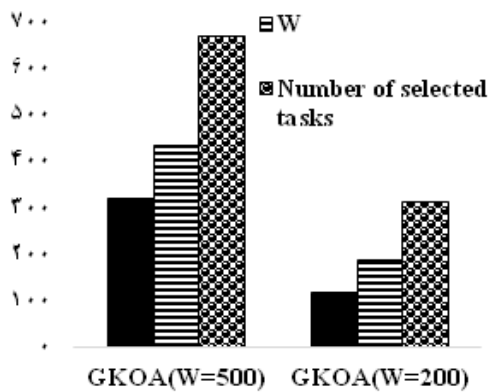
در ابر است. زمان انتظار برای انتقال در اجرای محلی برابر ۰ است زیرا انتقالی صورت نمی‌گیرد و در الگوریتم بارگذاری پیشنهادی با افزایش تقاضا برای انتقال مقدار مطلوب دارد. وقتی مقدار  $W$  برابر با ۵۰۰ شود، استفاده از الگوریتم GKO A برای پارامتر زمان اجرا بهترین مقدار را دارد، زیرا مقدار زمان انتظار در سمت دستگاه نسبت به اجرای محلی برای تعداد زیادی از کاربران کاهش می‌یابد. مقایسه این وضعیت در شکل ۸-ب نشان داده شده است. شکل ۸-ج نشان می‌دهد که میزان مصرف شبکه در حالت ارسال همه وظایف بیشترین مقدار را دارد و پس از آن، زمانی که در الگوریتم پیشنهادی ما مقدار  $W$  برابر با

برای این چهار حالت و تعداد مختلف کاربران نشان می‌دهد. مصرف انرژی در GKO A کمتر از PBP برای تعداد زیادی از کاربران است. همچنین مصرف انرژی برای بارگذاری تمام وظایف به لبه نسبت به اجرای تمام وظایف به صورت محلی پایین‌تر است. روش پیشنهادی با ظرفیت ۵۰۰ برای حالتی که تعداد کاربران بالاتر از ۲۰ است، بهترین مقدار را دارد. استفاده از GKO A کارآمدتر از بارگیری همه وظایف به لبه است و برای کاربران با تعداد زیاد بهتر عمل می‌کند. همان‌طور که در شکل ۷-ب نشان داده شده است، در طرح PBP که در کار [۲۹] ارائه گردید، میزان بار سیستم در هنگام افزایش تعداد کاربران افزایش می‌یابد. GKO A عملکرد بهتری را در سربار سیستم نسبت به سایر حالت‌ها ارائه می‌کند که بالاترین سربار سیستم برای اجرای محلی است. همان‌طور که در شکل ۷-ج نشان داده شده است، میزان تأخیر در اجرای محلی از همه حالت‌ها کمتر می‌باشد. نتیجه این ارزیابی و برتری روش پیشنهادی با ظرفیت ۵۰۰ نسبت به بقیه حالت‌ها در جدول ۳ آمده است.

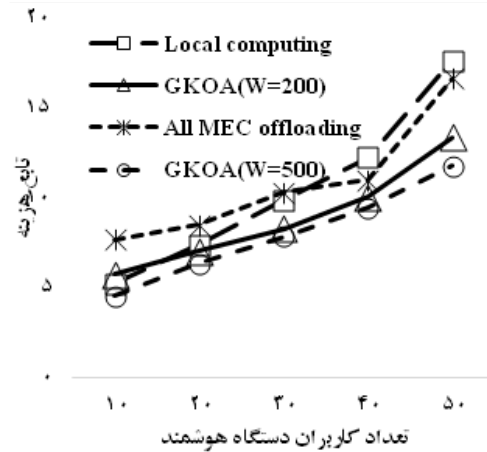
شکل ۸-الف مقایسه عملکرد روش پیشنهادی با ظرفیت ۲۰۰ و ۵۰۰، اجرای محلی و بارگذاری تمام وظایف به لبه را نشان می‌دهد. برای  $GKO A(W=200)$ ، تابع هزینه حداقل است که تا حدودی کمتر از عملکرد هزینه در اجرای تمام وظایف به صورت محلی و بارگذاری تمام وظایف به لبه است. هنگامی که  $W$  در GKO A به ۵۰۰ افزایش می‌یابد، تابع هزینه کاهش می‌یابد، زیرا امکان بارگذاری وظایف بیشتر به لبه وجود دارد. زمان اجرا و اتمام یک وظیفه کامل شامل جزئیاتی از زمان‌ها از جمله زمان اجرا در دستگاه تلفن همراه، زمان انتظار برای اجرای محلی، زمان انتظار برای انتقال، زمان انتقال، زمان انتظار برای اجرا در لبه و زمان اجرا



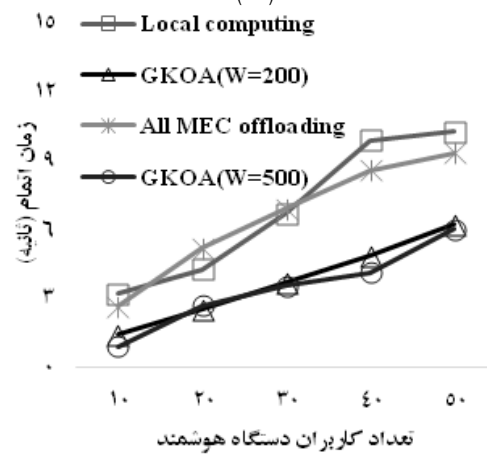
(ج) Profit



(د)



(ب)



(ا)

شکل ۸: مقایسه چند پارامتر برای تعداد مختلف کاربران تلفن در چهار طرح  $GKO(A(W=200))$  و  $GKO(A(W=500))$ ، اجرای محلی و بارگذاری تمام وظایف به لبه و همچنین جزئیات الگوریتم  $GKO(A)$ ، (الف) تأثیر تعداد کاربران بر تابع هزینه، (ب) تأثیر تعداد کاربران بر زمان اتمام وظایف، (ج) تأثیر تعداد کاربران بر زمان انتظار برای ارسال در شبکه و (د) جزئیات الگوریتم  $GKO(A)$ .

جدول ۳: مزیت  $GKO(A(W=500))$  نسبت به سایر روش‌ها با توجه به شکل ۵.

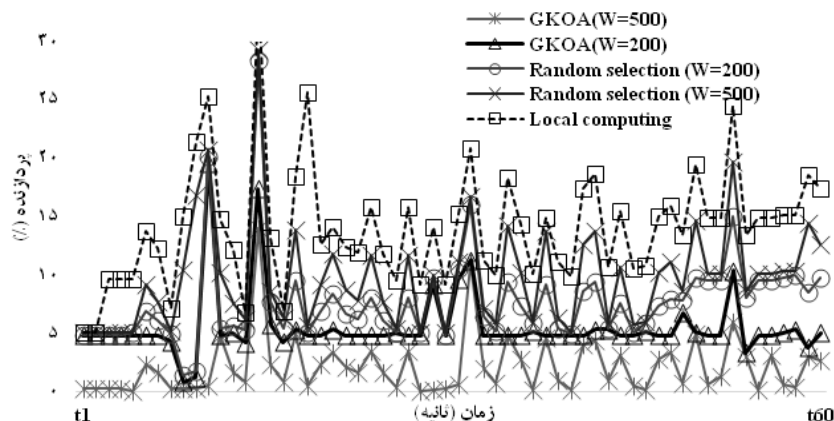
پارامتر	اجرای محلی	اجرای همه وظایف در لبه	$GKO(A(W=200))$	PBP
انرژی مصرفی	۲۸.۲	۲۱.۳۵	۱۰.۱	۱۲.۵۴
بار سیستم	۱۸.۰۵	۱۶.۲۴	۱۰.۵	۱۰.۷
میانگین تأخیر (ثانیه)	۱.۱	۱۹.۶۳	۱۱.۶	۰.۹۵

جدول ۴: جزئیات الگوریتم  $GKO(A)$ .

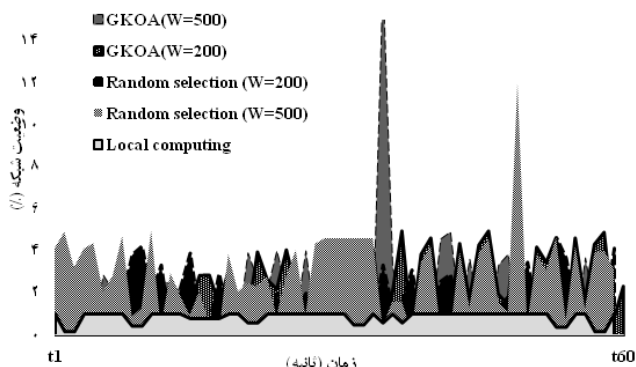
الگوریتم	مجموع ارزش	وزن کل	تعداد وظایف انتخاب شده برای بارگذاری
$GKO(A(W=500))$	۳۲۰.۹	۴۳۵	۶۶۸
$GKO(A(W=200))$	۱۱۷.۶	۱۸۹.۲	۳۱۴

مصرف باتری در اجرای محلی بیشتر از حالت‌های دیگر است، زیرا تمام وظایف در دستگاه اجرا می‌شوند. مقدار بهینه مصرف باتری در حالت  $GKO(A(W=500))$  است. بیشترین مقدار استفاده از باتری در این حالت به مقدار ۷٪ در  $t=60$  می‌باشد که در حالت انتخاب تصادفی  $GKO(A(W=500))$  برابر با ۱۹٪ است. وضعیت مصرف باتری در شکل ۱۰ نشان داده شده است. با توجه به شکل ۱۱ و ۱۲، میزان استفاده از RAM در اجرای محلی، بالا و وضعیت استفاده از شبکه در اجرای محلی کمترین مقدار را دارد، زیرا هیچ کدام از وظایف بارگذاری نمی‌شوند. در حالت انتخاب تصادفی  $GKO(A(W=200))$  و  $GKO(A(W=500))$ ، وضعیت استفاده از RAM بهترین مقدار را دارد. وضعیت انتقال در شبکه در بازه زمانی تعریف شده منتهای زمان‌های  $t_{off}$  و  $t_{on}$  برای چهار حالت

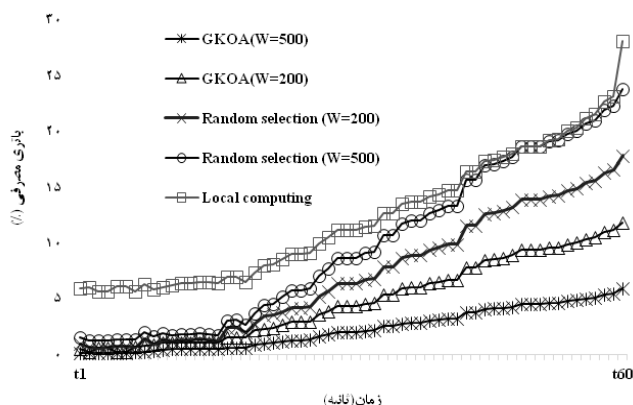
$500$  است، بیشترین استفاده از شبکه را داریم. جزئیات الگوریتم پیشنهادی در شکل ۲ حالت در شکل ۸-د نشان داده شده است. در  $GKO(A(W=200))$  مقدار وزن پر شده برابر با  $188.2/200$  به همراه ۳۱۴ وظیفه انتخابی و مقدار ارزش برابر با  $P=117$  است. در  $GKO(A(W=500))$ ، مقدار وزن پر شده برابر با  $335/500$  به همراه ۶۶۸ وظیفه انتخابی و مقدار ارزش برابر با  $P=320.9$  است. این جزئیات در جدول ۴ نشان داده شده‌اند. ما در این قسمت، وضعیت پردازنده، RAM و شبکه را در چند حالت روش پیشنهادی با ظرفیت ۲۰۰ و ۵۰۰، انتخاب تصادفی وظایف ( $W=500$ )، انتخاب تصادفی وظایف  $GKO(A(W=200))$  و اجرای محلی همه وظایف در زمان  $1 \leq t \leq 60$  می‌سنجیم. در شکل ۹، وضعیت پردازنده در اجرای محلی تمام وظایف در بالاترین ظرفیت استفاده است.



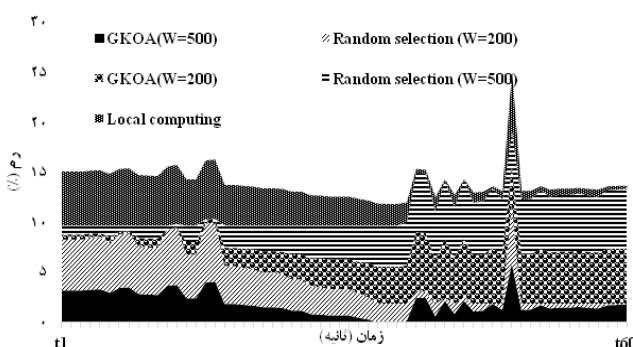
شکل ۹: وضعیت پردازنده.



شکل ۱۰: وضعیت پهنای باند شبکه.



شکل ۱۱: وضعیت باتری مصرفی.



شکل ۱۲: وضعیت RAM.

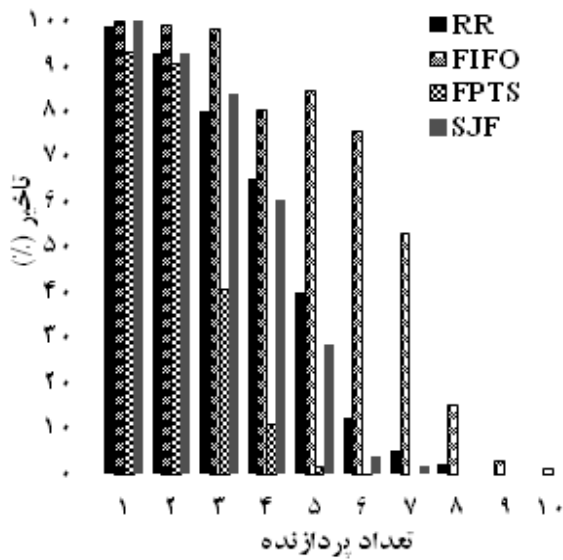
درد اما برای تعداد زیاد وظایف، پاسخی بهینه دارد. استفاده از این الگوریتم زمان انتظار، احتمال تأخیر و سربار سیستم را تا حد قابل قبولی کاهش می‌دهد و از سوی دیگر، سطح سرویس به میزان قابل توجهی افزایش می‌یابد. این الگوریتم با تعداد کم پردازنده‌ها کارایی بهتری دارد. جدول ۵ وضعیت چهار الگوریتم را مطابق شکل ۱۳ نشان می‌دهد. با توجه به مقادیر ارائه شده در اکثر حالت‌ها، الگوریتم پیشنهادی ما یک مقدار بهینه برای همه پارامترها ارائه می‌دهد. سیستم پیشنهادی با استفاده از FPTS، دارای حالت پایدار با کمترین تعداد پردازنده‌ها در مقایسه با سایر الگوریتم‌ها است.

مسئله بعدی که مورد ارزیابی قرار می‌گیرد درباره تخصیص منابع است. اکثر وظایف در سمت لبه اجرا می‌شوند، اما در صورت نبود منابع کافی و انتظار، به ابر منتقل می‌گردند. این فرایند برای روش پیشنهادی و روش معیار مورد بررسی قرار گرفت و در شکل ۱۴ نشان داده شده است. این جریان برای داده‌ها با نرخ ورود ۲۵۰ و ۵۰۰ آزمایش شد. در الگوریتم پیشنهادی و بعد از آن SJF درصد بیشتری از وظایف در سمت لبه اجرا می‌شوند که به ترتیب ۹۲/۱۱٪ در جایی که نرخ ورود داده برابر با ۲۵۰ است و ۹۰/۸۸٪ برای جایی که نرخ ورود داده برابر با ۵۰۰ است، می‌باشد. این مقادیر برای الگوریتم SJF با کارایی بهتر در سطح دوم برابر با ۸۰/۶۸٪ برای نرخ ورود ۲۵۰ و ۷۱/۳۸٪ برای نرخ ورود ۵۰۰ هستند.

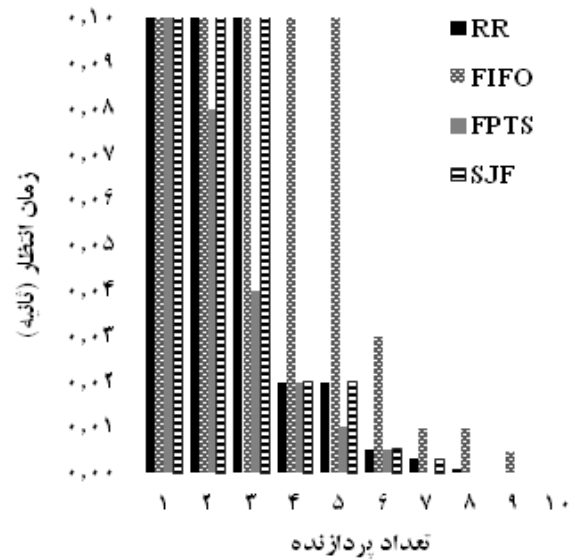
### ۶- نتیجه‌گیری

در این مقاله مدل بارگذاری و زمان بندی در سمت دستگاه و لبه بررسی شده است. بر اساس مدل پیشنهادی، بهینه‌سازی مصرف باتری در سمت دستگاه، بهینه‌سازی تأخیر و زمان انتظار سطح سرویس‌دهی در طرف لبه تحلیل شد. ما یک سیاست بارگذاری بهینه را ارائه کردیم که از الگوریتم کوله‌پشتی حریصانه، استفاده و بهترین مجموعه‌ای از کار را برای تخلیه در

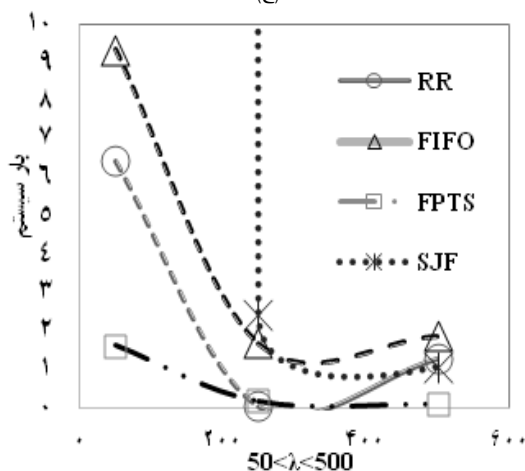
دیگر تفاوت زیادی ندارد. بیشترین میزان استفاده از شبکه برای هر دو حالتی است که مقدار  $W = 500$  است، زیرا وظایف بیشتری برای ارسال انتخاب می‌شوند. یکی از اهداف ما کاهش استفاده از منابع است، یعنی با کمترین تعداد پردازنده، بهترین بازدهی را داشته باشیم. مقایسه پارامترهای زمان انتظار، تأخیر، سربار سیستم و سطح سرویس برای الگوریتم پیشنهادی و سه الگوریتم RR، SJF و FIFO در شکل ۱۳ آمده است. در FPTS بیشترین مقدار زمان انتظار با یک پردازنده برابر با مقدار ۵ ثانیه است و با افزایش مقدار پردازنده‌ها به ۳، زمان انتظار برابر ۰/۵ می‌شود. در حالی که این مقدار برای الگوریتم‌های RR، SJF و FIFO به ترتیب برابر با ۰/۱۵، ۰/۲ و ۰/۲ با تعداد ۴، ۴ و ۴ پردازنده است. از طرفی در الگوریتم FPTS، احتمال کمترین تأخیر و بالاترین سطح سرویس با تعداد ۴ پردازنده حاصل می‌شود. احتمال تأخیر و سطح خدمات با پردازنده‌های ۸، ۷ و ۷-۸ برای الگوریتم‌های RR و FIFO بهبود می‌یابد. سربار سیستم نیز در FPTS مقدار بهینه دارد و پس از الگوریتم پیشنهادی SJF مقدار بهینه‌ای دارد. به طور کلی الگوریتم ارائه شده با روش SJF تفاوت کمی



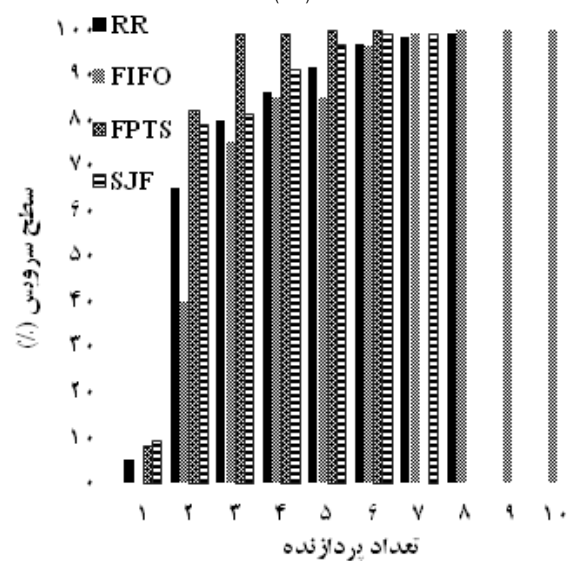
(ج)



(الف)



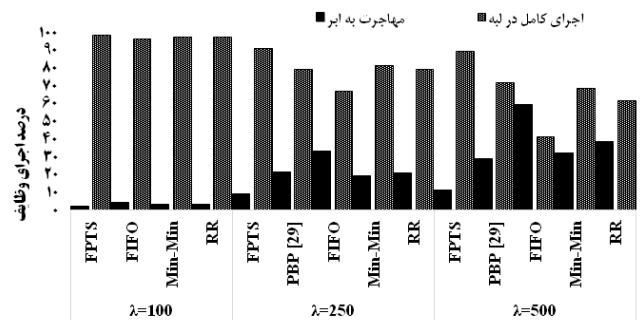
(د)



(ب)

شکل ۱۳: مقایسه پارامترهای ارزیابی با کمترین تعداد پردازنده‌ها برای الگوریتم پیشنهادی و الگوریتم‌های معیار، (الف) میانگین زمان انتظار، (ب) سطح سرویس، (ج) تأخیر و (د) بار سیستم.

انرژی نشان داد. نتایج عددی نشان دادند که الگوریتم GKOA در مصرف انرژی نسبت به اجرای تمام وظایف به صورت محلی ۲۸/۲٪، نسبت به اجرای تمام وظایف در سمت لبه به مقدار ۲۱/۳۵٪ و نسبت به روش PBP [۲۹] در حدود ۱۲/۵۴٪ بهبود یافت. این الگوریتم با افزایش تعداد کاربران، عملکرد بهتری را نسبت به بقیه حالت‌های ارزیابی شده دارد. الگوریتم FPTS، اولویت‌بندی مناسبی را از وظایف بر اساس دو معیار زمان اجرا و انرژی مصرفی ارائه می‌کند که در سرعت اجرای وظایف در محیط لبه نسبت به اجرای محلی در دستگاه مؤثر است. به طوری که اجرای مجموعه وظایف اولویت‌بندی شده در این زمان‌بند نسبت به روش‌های دیگر، باعث بهبود زمان انتظار، تأخیر و سطح سرویس با حداکثر ۵ پردازنده است، در حالی که این مقدار برای الگوریتم‌های FIFO، RR و SJF به ترتیب ۷، ۸ و ۷ است. در سیستم ارائه شده، میزان وظایف اجرا شده در بستر لبه به ترتیب ۹۲/۱۱٪ در جایی که نرخ ورود داده برابر با ۲۵۰ است و ۹۰/۸۸٪ برای جایی که نرخ ورود داده برابر با ۵۰۰ است، که تنها در حدود ۱۰٪ از وظایف برای اجرا به ابر منتقل می‌شوند. این کار یکی از محدودیت‌های مهم محیط محاسبات ابری یعنی دوری از کاربران و ترافیک بالا را با انتقال وظایف به لایه میانی لبه رفع می‌کند. همچنین



روش زمان‌بندی با نرخ ورود متفاوت

شکل ۱۴: درصد اجرای وظایف در لبه و مهاجرت به ابر.

لبه انتخاب می‌کند. با تقسیم‌بندی این الگوریتم، وظایف به دو دسته جهت اجرای محلی و بارگذاری در لبه تقسیم می‌شوند. این الگوریتم وظیفه دارد که بهترین انتخاب از وظایف را برای بارگذاری به لبه داشته باشد، به طوری که با بارگذاری آنها هزینه اجرای وظایف در لبه نسبت به اجرای آنها به صورت محلی بیشتر نشود و بتوان در مصرف منابع صرفه‌جویی کرد. GKOA بهترین نتایج بهینه را برای به حداقل رساندن مصرف

جدول ۵: مقایسه پارامترهای ارزیابی با توجه به شکل ۱۳.

الگوریتم	بار سیستم	تاخیر (%)	سروریسی (%)	زمان انتظار (ثانیه)	تعداد پردازنده
PPTS	۱۶	۹۳	۸	۵	۱
	۰٫۴۳۱	۹۰٫۳	۸۲٫۰۸	۰٫۰۸	۲
	۰٫۱۵۶	۴۶٫۴۳	۹۸٫۸۵	۰٫۰۴	۳
	۰٫۰۳۹	۱۰٫۸	۹۹٫۲	۰٫۰۲	۴
	۰٫۰۱۰	۱٫۴۲	۹۹٫۷	۰٫۰۱	۵
	۰٫۰۰۲	۰٫۳۵	۹۹٫۹	۰٫۰۰۵	۶
FHO	۱	۱۰۰	۱	۱۰	۱
	۴۰۰	۹۹	۴۰	۵	۲
	۸۶٫۰۳۶	۹۸	۷۵٫۲۱	۳٫۹	۳
	۷۲٫۲۵۰	۸۰	۸۵	۱	۴
	۱٫۶۳۶	۸۴٫۳	۸۵٫۲۶	۰٫۱۵	۵
	۰٫۰۸۴	۷۵٫۴۲	۹۶٫۵۶	۰٫۰۳	۶
	۰٫۰۱۰	۵۳٫۰۲	۹۹٫۲۶	۰٫۰۱	۷
	۰٫۰۱۰	۱۴٫۸۳	۹۹٫۸۵	۰٫۰۱	۸
	۰٫۰۰۲	۲٫۳۶	۹۹٫۹۷	۰٫۰۰۵	۹
	۰	۱	۹۹٫۹	۰٫۰۰۱	۱۰
SJF	۷٫۶۱۸	۹۹٫۹	۹٫۲	۳	۱
	۶٫۴۱۰	۹۳	۷۹	۱	۲
	۲٫۳۸۴	۸۳٫۸۷	۸۱٫۲۷	۰٫۱۹	۳
	۰٫۰۳۳	۶۰٫۲۵	۹۱٫۰۲	۰٫۰۲	۴
	۰٫۰۳۷	۲۸٫۵۲	۹۶٫۸۲	۰٫۰۲	۵
	۰٫۰۰۲	۳٫۶۹	۹۸٫۹۸	۰٫۰۰۵	۶
	۰٫۰۰۱	۱٫۴	۹۹	۰٫۰۰۳	۷
	۲٫۲۵	۹۹	۵	۳	۱
RR	۴۲٫۲۵	۹۳	۶۵	۱	۲
	۲٫۳۱۰	۷۹٫۹	۸۰	۰٫۱۹	۳
	۰٫۰۳۰	۶۵	۸۶٫۰۲	۰٫۰۲	۴
	۰٫۰۳۴	۴۰	۹۱٫۸۲	۰٫۰۲	۵
	۰٫۰۰۲	۱۲٫۳	۹۶٫۶	۰٫۰۰۵	۶
	۰٫۰۰۱	۵	۹۸٫۲	۰٫۰۰۳	۷
	۰٫۰۰۱	۲٫۱	۹۹	۰٫۰۰۱	۸

## مراجع

- [1] J. Huang, C. C. Xing, and C. Wang, "Simultaneous wireless information and power transfer: technologies, applications, and research challenges," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 26-32, Nov. 2017.
- [2] K. Zhang, et al., "Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896-5907, 2016.
- [3] Y. Sun, J. Li, X. Fu, H. Wang, and H. Li, "Application research based on improved genetic algorithm in cloud task scheduling," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 1, pp. 239-246, Jan. 2020.
- [4] G. Li, J. Wang, J. Wu, and J. Song, "Data processing delay optimization in mobile edge computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID: 6897523, 9 pp., 2018.
- [5] W. Zhimin, Z. Qinglin, X. Fangxin, D. Hongning, and Z. Yujun, "Detection performance of packet arrival under downclocking for mobile edge computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID: 9641712, 2018.
- [6] W. Yu, et al., "A survey on the edge computing for the Internet of things," *IEEE Access*, vol. 6, pp. 6900-6919, 2018.
- [7] Y. Guo, et al., "Distributed machine learning for multiuser mobile edge computing systems," *IEEE J. of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 460-473, Apr. 2022.
- [8] Z. Jiao, et al., "Energy-latency trade-off for energy-aware offloading in mobile edge computing networks," *IEEE Internet of Things J.*, vol. 5, no. 4, pp. 2633-2645, Aug. 2018.
- [9] A. Selvaraj and S. Sundararajan, "Evidence-based trust evaluation system for cloud services using fuzzy logic," *International J. of Fuzzy Systems*, vol. 19, no. 2, pp. 329-37, Apr. 2017.
- [10] M. Fonseca, U. H. Bezerra, J. D. Brito, J. C. Leite, and M. H. Nascimento, "Pre-dispatch of load in thermoelectric power plants considering maintenance management using fuzzy logic," *IEEE Access*, vol. 6, pp. 41379-41390, 2018.
- [11] K. Wang, K. Yang, and C. Magurawalage, "Joint energy minimization and resource allocation in C-RAN with mobile cloud," *IEEE Trans. Cloud Comput.*, vol. 6, no. 3, pp. 331-346, Jul./Sept. 2017.
- [12] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590-3605, Dec. 2016.
- [13] H. Kchaou, Z. Kechaou, and A. M. Alimi, "Towards an offloading framework based on big data analytics in mobile cloud computing environments," *Procedia Comput. Sci.*, vol. 53, pp. 292-297, 2015.
- [14] S. Misra and S. Sarkar, "Theoretical modeling of fog computing: a green computing paradigm to support IoT applications," *IET Networks*, vol. 5, no. 2, pp. 23-29, Mar. 2016.
- [15] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Trans. Computers*, vol. 64, no. 8, pp. 2253-2266, Aug. 2015.
- [16] G. Shuaishuai, W. Dalei, Z. Haixia, and Y. Dongfeng, "Resource modeling and scheduling for mobile edge computing: a service provider's perspective," *IEEE Access*, vol. 6, pp. 35611-35623, 2018.
- [17] Z. Wenchen, et al., "Markov approximation for task offloading and computation scaling in mobile edge computing," *Mobile Information Systems*, vol. 2019, Article ID: 8172698, 2019.
- [18] T. Chia Wei, T. Fan Hsun, Y. Yao Tsung, L. Chien Chang, and C. LiDer, "Task scheduling for edge computing with agile VNFs on-demand service model toward 5G and beyond," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID: 7802797, 2018.
- [19] Y. Yibo, M. Yongkui, X. Wei, G. Xuemai, and Z. Honglin, "Joint optimization of energy consumption and packet scheduling for

محدودیت روش های سنتی زمان بندی را در اولویت بندی وظایف، تنها بر اساس یک معیار از جمله زمان اجرا نیز با اولویت بندی دومعیاره مبتنی بر فازی که اهمیت هر دو معیار را در نظر می گیرد، مرتفع می کند. انتقال وظایف به لبه در جهت افزایش سرعت اجرای وظایف نسبت به اجرای محلی و صرفه جویی در منابع دستگاه است. نحوه انتخاب دسته مناسبی از وظایف، نحوه انتقال آنها بر اساس خطوط ارتباطی موجود و نحوه اجرای آنها در محیط لبه، یکی از محدودیت های کار است و در صورتی که هر کدام از این موارد رعایت نشوند، اهداف گفته شده یعنی اجرای سریع و صرفه جویی در منابع تأمین نمی گردد.

با توجه به این که وظایف در هنگام انتقال به محیط محاسبات سیار، حامل معیارهای بسیاری از جمله حجم داده، پهنای باند مصرفی و ... هستند، محاسبه این معیارها و یافتن اهمیت آنها جهت به کارگیری در مسئله اولویت بندی و زمان بندی یکی از مسائل مهم است. در آینده، لازم است که روش پیشنهادی را در تعداد زیادی ماشین مجازی توزیع شده اجرا

- IEEE Wireless Communications*, vol. 24, no. 4, pp. 64-71, Aug. 2017.
- [39] A. Salehan, H. Deldari, and S. Abrishami, "An online context-aware mechanism for computation offloading in ubiquitous and mobile cloud environments," *The J. of Supercomputing*, vol. 75, no. 7, pp. 1-41, Ju., 2019.
- [40] M. Al-Zinati, R. Alrashdan, B. Al-Duwairi, and M. Aloqaily, "A reorganizing biosurveillance framework based on fog and mobile edge computing," *Multimedia Tools and Applications*, vol. 80, no. 11, pp. 16805-16825, 2021.
- [41] S. Deng, *et al.*, "Burst load evacuation based on dispatching and scheduling in distributed edge networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 32, no. 8, pp. 1918-1932, Aug. 2021.
- [42] F. Jazayeri, A. Shahidinejad, and M. Ghobaei-Arani, "Autonomous computation offloading and auto-scaling the in the mobile fog computing: a deep reinforcement learning-based approach," *J. of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 8265-8284, 2021.
- [43] J. C. Guevara and N. L. da Fonseca, "Task scheduling in cloud-fog computing systems," *Peer-to-Peer Networking and Applications*, vol. 14, no. 2, pp. 962-977, 2021.
- [44] T. D. Lee, B. M. Lee, and W. Noh, "Hierarchical cloud computing architecture for contextaware IoT services," *IEEE Trans. Consume Electron*, vol. 64, no. 2, pp. 222-230, May 2018.
- [45] Y. Shi, S. Chen, and X. Xu, "MAGA: a mobility-aware computation offloading decision for distributed mobile cloud computing," *IEEE Internet of Things*, vol. 5, no. 1, pp. 164-174, Feb. 2018.
- [46] M. Goudarzi, M. Zamani, and A. Toroghi Haghighat, "A genetic-based decision algorithm for multisite computation offloading in mobile cloud computing," *Int J. Commun Syst*, vol. 30, no. 10, Article ID: e3241, Jul. 2017.
- [47] A. Raneetha, H. Moshe, O. Binyamin, and Z. Ilze, "Strategic bidding in a discrete accumulating priority queue," *Operations Research Letters*, vol. 47, no. 3, pp. 162-167, May 2019.
- [48] J. Yuan and Y. Li, "Solving binary multi-objective knapsack problems with novel greedy strategy," *Memetic Comp.*, vol. 13, pp. 447-458, 2021.
- [49] V. Kumar and K. Dinesh, "Transitive blocks and their applications in fuzzy interconnection networks," *Fuzzy Sets and Systems*, vol. 352, pp. 142-160, Dec. 2018.
- [50] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *Proc. IEEE Global Communication Conf., GLOBECOM'16*, 6 pp., Washington, DC, USA, 4-8 Dec. 2016.
- [51] R. Gopi, S. T. Suganthi, R. Rajadevi, *et al.*, "An enhanced green cloud based queue management (GCQM) system to optimize energy consumption in mobile edge computing," *Wireless Pers Commun*, vol. 117, pp. 3397-3419, 2021.
- [52] P. Zou, O. Ozel, and S. Subramaniam, "Optimizing information freshness through computation-transmission tradeoff and queue management in edge computing," *IEEE/ACM Trans. on Networking*, vol. 29, no. 2, pp. 949-963, Apr. 2021.
- mobile edge computing in cyber-physical networks," *IEEE Access*, vol. 6, pp. 15576-15586, 2018.
- [20] W. Yuan, *et al.*, "NOMA-Assisted multi-access mobile edge computing: a joint optimization of computation offloading and time allocation," *IEEE Trans. on Vehicular Technology*, vol. 67, no. 12, pp. 12244-12258, Dec. 2018.
- [21] A. Aghababaeipour and S. Ghanbari, "A new adaptive energy-aware job scheduling in cloud computing," in *Proc. Int. Conf. on Soft Computing and Data Mining*, pp. 308-317, Senai, Malaysia, 5-7 Feb. 2018.
- [22] S. Ghanbari and M. Othman, "Time cheating in divisible load scheduling: sensitivity analysis, results and open problems," *Procedia Computer Science*, pp. 935-943, vol. 1, no. 125, Jan. 2018.
- [23] Y. Changyan, C. Jun, and S. Zhou, "A multi-user mobile computation offloading and transmission scheduling mechanism for delay sensitive application," *IEEE Trans. on Mobile Computing*, vol. 19, no. 2, pp. 99-110, Jan. 2018.
- [24] D. Yueyue, X. Du, M. Sabita, and Y. Zhang, "Joint computation offloading and user association in multi-task mobile edge computing," *IEEE Trans. on Vehicular Technology*, vol. 67, no. 12, pp. 12313-12325, Dec. 2018.
- [25] X. Chen, J. Lei, L. Wenzhong, and F. Xiaoming, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. on Networking*, vol. 24, no. 5, pp. 2795-2808, Oct. 2016.
- [26] C. Ying, Z. Ning, Z. Yongchao, and C. Xin, "Dynamic computation offloading in edge computing for the Internet of Things," *IEEE Internet of Things J.*, vol. 6, no. 3, pp. 239-251, Jun. 2019.
- [27] N. Zhaolong, D. Peiran, K. Xiangjie, and X. Feng, "A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things," *IEEE Internet of Things J.*, vol. 6, no. 3, pp. 4804-4814, Jun. 2019.
- [28] Y. Zhang, D. Niyato, and P. Wang, "Offloading in mobile cloudlet systems with intermittent connectivity," *IEEE Trans. Mobile Computing*, vol. 14, no. 12, pp. 2516-2529, Dec. 2015.
- [29] L. Tianze, W. Muqing, Z. Min, and L. Wenxing, "An overhead optimizing task scheduling strategy for ad hoc based mobile edge computing," *IEEE Access*, vol. 5, pp. 5609-5622, 2017.
- [30] R. Morabito and N. Bejar, "Enabling data processing at the network edge through lightweight visualization technologies," in *Proc. of the IEEE Int. Conf. on Sensing, Communication, and Networking, SECON Workshops*, 6 pp., London, UK, Jun. 2016.
- [31] H. Wang, J. Gong, Y. Zhuang, H. Shen, and J. Lach, "Health edge: task scheduling for edge computing with health emergency and human behavior consideration in smart homes," in *Proc. of the IEEE Int. Conf. on Big Data*, pp. 1213-1222, Shenzhen, China, 7-9 Aug. 2017.
- [32] P. Samal and P. Mishra, "Analysis of variants in round robin algorithms for load balancing in cloud computing," *International J. of Computer Science and Information Technologies*, vol. 4, no. 3, pp. 416-419, 2013.
- [33] C. You, Y. Mao, J. Zhang, and K. Huang, "Energy-efficient offloading for mobile edge computing," In *Wiley 5G Ref: The Essential 5G Reference Online*, 2016.
- [34] J. Maozhu, W. Hua, S. Lijun, L. Yuxue, and Z. Yucheng, "Man-machine dialog system optimization based on cloud computing," *Personal and Ubiquitous Computing*, vol. 22, no. 5-6, pp. 937-942, Oct. 2018.
- [35] J. Wang, J. Peng, Y. Wei, D. Liu, and F. Jielin, "Adaptive application offloading decision and transmission scheduling for mobile cloud computing," in *Proc. IEEE Int. Conf. on Communications, ICC'26*, 7 pp., Kuala Lumpur, Malaysia, 22-27 May 2016.
- [36] F. Scott, K. Carlenri, Y. Di, B. George, and K. Mellow, "An analysis of vehicular wireless channel communication via queueing theory model," in *Proc. IEEE Int. Conf. on Communications*, pp. 1736-1741, Sydney, Australia, 10-14 Jun. 2014.
- [37] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proc. 35th Annual IEEE Int. Conf. on Computer Communications, IEEE INFOCOM'16*, 9 pp., San Francisco, CA, USA, 10-14 Apr. 2016.
- [38] X. Chen, L. Pu, L. Gao, W. Wu, and D. Wu., "Exploiting massive D2D collaboration for energy-efficient mobile edge computing,"
- انتصار حسینی** مدرک دکتری را در رشته امنیت اطلاعات در سال ۱۴۰۰ از دانشگاه قم دریافت نمود. برخی از زمینه های تحقیقی ایشان عبارتند از: رایانش ابری، امنیت در محیطهای محاسباتی رایانش ابری و محاسبات سیار.
- محسن نیک رای** مدرک دکتری خود را در سال ۱۳۹۱ در رشته معماری کامپیوتر از دانشگاه تهران اخذ نمود. در حال حاضر ایشان استادیار گروه کامپیوتر دانشگاه قم می باشد. برخی از زمینه های تحقیقاتی ایشان عبارت است از: معماری کامپیوتر، زمان بندی، رایانش ابری، محاسبات سیار.
- شمس اله قنبری** مدرک دکتری را در سال ۱۳۹۴ از دانشگاه UPM مالزی در زمینه سیستمهای توزیع شده دریافت نمود. ایشان در حال حاضر عضو هیأت علمی گروه کامپیوتر دانشگاه آزاد استان مرکزی واحداشتیان می باشند. برخی از زمینههای تحقیقاتی مورد علاقه ایشان عبارت است از: زمان بندی در محیطهای سیستمهای موازی و توزیع شده، رایانش ابری، پایگاههای داده، پیچیدگی محاسباتی.