ITRC
ICT Research Institute
(Iran Telecommunication Research Center)

Journal of Information and Communication Technology

Iranian Society of Information and Communication Technology

Volume 16, Issue 59-60, Spring and Summer 2024, pp. 108-124

Improving Resource Allocation in Mobile Edge Computing Using Particle Swarm and Gray Wolf Optimization Algorithms

Seyed Ebrahim Dashti[1*], Saeed Shabboyi[2]

[1] Electrical and Computer Faculty, Jahrom Branch, Islamic Azad University, Jahrom, Iran
[2] Electrical and Computer Faculty, Zahidshahr Branch, Islamic Azad University, Zahidshahr, Iran

Received: 19 October 2022, Revised: 19 November 2022, Accepted: 10 April 2023
Paper type: Research

Abstract

Mobile edge computing improves the experience of end users to achieve appropriate services and service quality. In this paper, the problem of improving resource allocation, when offloading tasks, based on mobile devices to edge servers in computing systems is investigated. Some tasks are uploaded and processed locally and some to edge servers. The main issue is that the offloaded tasks for virtual machines in computing networks are properly scheduled to minimize computing time, service cost, computing network waste, and the maximum connection of a task with the network. In this paper, a multi-objective hybrid algorithm of particle swarm and gray wolf was introduced to manage resource allocation and task scheduling to achieve an optimal result in edge computing networks. Local search in the particle swarm algorithm has good results in the problem, but it will cause the loss of global optima, so in this problem, in order to improve the model, the gray wolf algorithm was used as the main basis of the proposed algorithm, in the wolf algorithm Gray, due to the graphical approach to the problem, the set of global searches will reach the optimal solution, so by combining these functions, we tried to improve the operational conditions of the two algorithms for the desired goals of the problem. In order to create a network in this research, the network creation parameters in the basic article were used and the LCG data set was used in the simulation. The simulation environment in this research is the sim cloud environment. The comparison results show the improvement of waiting time and cost in the proposed approach. The results show that, on average, the proposed model has performed better by reducing the work time by 10% and increasing the use of resources by 16%.

Keywords: Resource Allocation Improvement, Mobile Edge Computing, Particle Swarm Optimization Algorithm, Gray Wolf Algorithm.

[*] Corresponding Author's email: sayed.dashty@gmail.com

دوفصلنامه
فناوری اطلاعات و ارتباطات ایران

سال شانزدهم، شماره های ۵۹ و ۶۰، بهار و تابستان ۱۴۰۳، صفحه ۱۰۸ الی ۱۲۴

بهبود تخصیص منابع در محاسبات لبه موبایل با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات و گرگ خاکستری

سید ابراهیم دشتی[1*]، سعید شب بویی[2]

[1] دانشکده برق و کامپیوتر ، واحد جهرم، دانشگاه آزاد اسلامی، جهرم، ایران

[2] دانشکده برق و کامپیوتر، واحد زاهدشهر، دانشگاه آزاد اسلامی، زاهدشهر، ایران

تاریخ دریافت: ۱۴۰۱/۰۷/۲۷ تاریخ بازبینی: ۱۴۰۱//۰۸/۲۸ تاریخ پذیرش: ۱۴۰۲/۰۱/۲۱

نوع مقاله: پژوهشی

چکیده

محاسبات لبه موبایل تجربه کاربران نهایی را برای دستیابی به خدمات مناسب و کیفیت خدمات بهبود می‌دهد. در این مقاله مسئله بهبود تخصیص منابع، هنگام بارگیری وظایف، براساس دستگاه‌های تلفن همراه به سرورهای لبه در سیستم‌های محاسباتی بررسی می‌شود. برخی وظایف به صورت محلی و برخی به سرورهای لبه بارگذاری و پردازش می‌شوند. مسئله اصلی این است که وظایف تخلیه‌شده برای ماشین‌های مجازی در شبکه‌های محاسباتی بصورت مناسب زمانبندی شوند تا زمان محاسبات، هزینه خدمات، اتلاف شبکه‌های محاسباتی و حداکثر ارتباط یک کار با شبکه به حداقل برسد. در این مقاله الگوریتم ترکیبی چند هدفه ازدحام ذرات و گرگ خاکستری برای مدیریت تخصیص منابع و زمان‌بندی وظایف برای دستیابی به یک نتیجه بهینه در شبکه‌های محاسبات لبه معرفی شد. جست و جوی محلی در الگوریتم ازدحام ذرات نتایج مناسبی را در مسئله دارد اما باعث از بین رفتن بهینه‌های سراسری خواهد شد، بنابراین در این مسئله به منظور بهبود مدل، از الگوریتم گرگ خاکستری به عنوان پایه اصلی الگوریتم پیشنهادی استفاده شد، در الگوریتم گرگ خاکستری به دلیل رویکرد گرافی مسئله، مجموعه جست و جوهای سراسری به جواب بهینه‌ای خواهد رسید، بنابراین با ترکیب این توابع سعی در بهبود شرایط عملیاتی دو الگوریتم برای اهداف مورد نظر مساله شد. به منظور ایجاد شبکه در این پژوهش از پارامترهای ایجاد شبکه در مقاله پایه استفاده شده است و در شبیه‌سازی از مجموعه داده LCG استفاده شد. محیط شبیه‌سازی در این پژوهش محیط شبیه‌ساز کلود سیم است. نتایج مقایسه نشان‌دهنده بهبود زمان انتظار و هزینه در رویکرد پیشنهادی است. نتایج نشان می‌دهد که به طور میانگین مدل پیشنهادی با کاهش ۱۰ درصدی زمان انجام کار و افزایش استفاده از منابع به میزان ۱۶ درصد بهتر عمل کرده است.

کلیدواژگان: بهبود تخصیص منابع، محاسبات لبه موبایل، الگوریتم بهینه‌سازی ازدحام ذرات، الگوریتم گرگ خاکستری.

* رایانامهٔ نویسنده مسؤول: sayed.dashty@gmail.com

۱- مقدمه

رشد روزافزون دستگاه‌های هوشمند منجر به تغییر چشمگیری در جامعه شده است که اکنون به شدت به فناوری‌های سلولی[1] متکی است [۱]. اینترنت اشیا در حال تبدیل شدن به یک ضرورت مهم برای بسیاری از کاربردهای صنعتی و فناوری ارتباطات است. بهبود و توسعه بسیاری در پیاده‌سازی اینترنت اشیا وجود داشته است، این در حالی است که تعداد انبوه ۵۰ میلیارد دستگاه متصل به اینترنت تا سال ۲۰۲۵ وجود دارد. علاوه بر این، برنامه‌های اینترت اشیا که برای کمک به افراد ناتوان یا سالخورده طراحی شده‌اند، سهولت و تحرک را با درجات مختلف غیرمتعارف با قیمتی معقول فراهم می‌کنند. علاوه بر این، اینترنت اشیاء در زمینه‌های متعددی مورد استفاده قرار می‌گیرد، به عنوان مثال، تجارت کشاورزی، آب و هوا، مناطق بالینی، بخش آموزشی، حمل و نقل و امور مالی. نتایج آماری نشان دهنده تعداد دستگاه‌های اینترنت اشیا به میلیاردها دستگاه فعال در سراسر جهان از سال ۲۰۱۸ تا ۲۰۳۰ نشان می‌دهد.

سازمان‌ها و انجمن‌های علمی تحقیقاتی متعددی روی جنبه‌های مختلف اینترنت اشیا کار می‌کنند. آنها یک طرح کلی کاربردی برای تأثیرات اینترنت اشیا بر اقتصاد و اکثریت قریب به اتفاق سایر زمینه‌های موجود در ۱۰ سال آینده ارائه کرده‌اند. سیسکو اولین سازمانی است که تعهدات متعدد اینترنت اشیا را ارائه می‌دهد که شامل ۲۴ میلیارد شی هوشمند در سال ۲۰۱۹ بوده است و همچنین انتظار می‌رود که شرکت هواوی تا سال ۲۰۲۵، ۱۰۰ میلیارد دستگاه اینترنت اشیا را سراسر دنیا داشته باشد. در هر ثانیه در جهان، ۱۲۷ دستگاه به اینترنت متصل می‌شوند. تا سال ۲۰۲۰، ۶۳ درصد از کل استفاده از دستگاه‌های الکترونیکی از فناوری اینترنت اشیا بوده است. از کل پروژه‌های تجاری عظیم و هوشمند شهر، ۲۳ درصد شامل پیاده‌سازی اینترنت اشیا می‌شود، در حالی که تا پایان سال ۲۰۲۰، ۴۰ درصد از تمام سازمان‌های بهداشتی و درمانی از اینترنت اشیا استفاده می‌کردند[۲]. افزایش داده‌های حاصل، بار سنگینی را برای ارائه‌دهندگان خدمات تلفن همراه ایجاد کرده است. بدون اقدامات مناسب برای ذخیره و پردازش چنین بارهای کاری، شبکه‌ها شلوغ‌تر می‌شوند و در نتیجه کیفیت و سرعت دانلود روند کاهشی خواهد داشت. از این رو، منابع محاسباتی اضافی برای دستگاه‌های تلفن همراه ضروری است. علاوه بر این، تکامل بلندمدت فناوری 5G الهام‌بخش طیف گسترده‌ای از خدمات است که به وظایف محاسباتی بالایی نیاز دارد، که دستگاه‌های تعیین‌شده برای انجام آنها مجهز نیستند[۲].

محاسبات لبه موبایل[2] (MEC) یک راه‌حل امیدوارکننده است که از سرورهای ابری مستقر در پشتیبانی از دستگاه‌های تلفن همراه برای کاهش بارهای کاری محاسباتی از طریق تخلیه فرآیند استفاده می‌کند. در سال ۲۰۰۹، اولین مفهوم محاسبات لبه (به عنوان مثال، ابر) ارائه شد. محاسبات ابری به کاربران تلفن همراه اجازه می‌دهند تا از خدمات ابری استفاده کنند [۳].

وظایف را نیز می‌توان به صورت محلی در دستگاه‌های تلفن همراه با استفاده از مفهوم ابر موقت اجرا کرد[۴]. چندین دستگاه کاربر را قادر می‌سازد تا منابع محاسباتی خود را برای پردازش وظایف ترکیب کنند. قابل ذکر است که بارگذاری در یک سرور لبه مستقیماً کیفیت تجربه و طول عمر باتری را بهبود می‌بخشد[۵]. در سال ۲۰۱۲، سیسکو مفهوم بارگذاری محاسباتی را پیشنهاد کرد[۶]. از این رو، هر دستگاه تلفن همراهی که منابع محدودی داشته باشد، می‌تواند وظایف پردازشی را به صورت بی‌سیم به دستگاه‌های دیگری که منابع کافی دارند، منتقل کند. سپس آن دستگاه‌های دیگر وظایف را تکمیل کرده و نتایج را به دستگاه‌های تلفن همراه ارسال می‌کنند. متأسفانه این روش به دلیل ویژگی‌های شبکه‌های بی‌سیم در مناطق روستایی و کوهستانی، همچنان کمتر از حد انتظار است. علاوه بر این، موارد واکنش اضطراری همیشه باید اولویت داشته باشند. بنابراین، حتی در ایده‌آل‌ترین محیط‌ها، حفظ کیفیت تجربه و بهره‌وری انرژی در عین اجتناب از تاخیرهای ارتباطی دشوار است بنابراین می‌توان این گونه بیان کرد که با ظهور شبکه 5G، سرویس‌های مختلف هوشمند دائماً در حال ظهور هستند. میلیاردها دستگاه ترمینال هوشمند برای انجام تعداد زیادی از کارها مورد نیاز است، اما تعداد محدود دستگاه‌های ترمینال توانایی‌های انجام آنها را محدود می‌کند[۱]. محاسبات لبه دستگاه‌های ترمینال را قادر می‌سازد نه تنها به شبکه بی‌سیم دسترسی داشته باشند بلکه برخی از کارهای محاسباتی را به سرورهای محاسبات لبه بارگیری کنند تا فاصله بین سرورها و سرورهای لبه کاهش یابد و زمان اتمام وظایف نیز کوتاه شود[۲]. در محاسبات لبه، استراتژی‌های تخلیه و تخصیص منابع به طور مستقیم بر عملکرد سیستم تأثیر می‌گذارد، بنابراین اخیراً به یک چالش تحقیقاتی تبدیل شده است[۳]. اهداف پژوهش شامل موارد زیر است:

- به حداقل رساندن زمان اتمام کلی وظایف در محاسبات لبه با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات و گرگ خاکستری

[1] Cellular Technologies

[2] Mobile edge computing

- ارائه الگوریتم بهینه‌سازی برای تخلیه و تخصیص منابع بر اساس الگوریتم ازدحام ذرات و گرگ خاکستری

بنابراین در این پژوهش سعی در ارائه روندی بهینه به منظور ارائه استراتژی تخلیه مناسب پرداخته شده است. سازماندهی این مقاله به این صورت است که در بخش بعد به بیان پیشینه پژوهش پرداخته شده است در بخش سوم روش پیشنهادی و در نهایت در بخش آخر به بررسی نتایج پژوهش پرداخته شد.

۲- پیشینه تحقیق

در این بخش به بررسی پیشینه پژوهش پرداخته خواهد شد در ابتدا به بررسی مفاهیم مربوط به مبانی پرداخته شد. طبق مؤسسه استانداردهای مخابرات اروپا، محاسبات لبه سیار به این صورت تعریف می‌شود [۷]:

«محاسبات لبه موبایل یک محیط خدمات فناوری اطلاعات و قابلیت‌های محاسبات ابری را در لبه شبکه تلفن همراه، در شبکه دسترسی رادیویی[1] (RAN) و در نزدیکی مشترکین تلفن همراه فراهم می‌کند.»

طبق تعریف بالا می‌توان این نتیجه را گرفت که محاسبات لبه موبایل قابلیت‌های رایانش ابری را در شبکه دسترسی رادیویی ارائه می‌دهد. در عوض، محاسبات لبه موبایل کاربر را مستقیماً به نزدیکترین شبکه لبه فعال با سرویس ابری متصل می‌کند. استقرار محاسبات لبه موبایل در ایستگاه پایه محاسبات را بهبود می‌دهد [۸] [۹].

طبق مرجع [۱۰]، مزایای لایه محاسبات لبه را می‌توان این گونه مشخص کرد:

۱. محلی[2]: محاسبات لبه موبایل به صورت تفکیک شده انجام می‌شود که عملکرد آن را در محیط محاسباتی بهبود می‌دهد. با وجود محاسبات لبه موبایل و جدا بودن آن از شبکه‌های دیگر، این لایه را کمتر آسیب‌پذیر می‌کند.
۲. نزدیکی: محاسبات لبه تلفن همراه با استقرار در نزدیکترین مکان، مزیتی برای تجزیه و تحلیل و تحقق داده‌های بزرگ دارد. همچنین برای دستگاه‌های محاسباتی مانند تجزیه و تحلیل ویدیویی و غیره مفید است.
۳. تأخیر کمتر: خدمات محاسباتی لبه تلفن همراه در نزدیکترین مکان به دستگاه‌های کاربر مستقر می‌شوند که حرکت داده‌های شبکه را از شبکه اصلی جدا می‌کند. از این رو، تجربه کاربری با کیفیت بالا با تاخیر بسیار کم و پهنای باند بالا به حساب می‌آید.
۴. آگاهی از موقعیت: دستگاه‌های توزیع شده لبه از سیگنالینگ سطح پایین برای به اشتراک گذاری اطلاعات استفاده می‌کنند. محاسبات لبه موبایل اطلاعاتی را از دستگاه‌های لبه در شبکه دسترسی محلی برای کشف مکان دستگاه دریافت می‌کند.
۵. اطلاعات شبکه: برنامه‌های کاربردی ارائه اطلاعات شبکه و خدمات داده‌های شبکه بلادرنگ می‌توانند با پیاده‌سازی محاسبات لبه موبایل در مدل کسب و کار خود به کسب و کارها و رویدادها کمک کنند. بر اساس اطلاعات بلادرنگ شبکه دسترسی رادیویی، این برنامه‌ها می‌توانند در مورد ازدحام سلول‌های رادیویی و پهنای باند شبکه قضاوت کنند که در آینده به آنها کمک می‌کند تا تصمیم‌گیری هوشمندانه برای تحویل بهتر مشتری بگیرند.

معماری مرجع برای محاسبات لبه یک ساختار شبکه فدرال است که قادر است خدمات ابری را تا لبه شبکه از طریق معرفی دستگاه‌های لبه بین دستگاه‌های ترمینال و رایانش ابری گسترش دهد. چنین معماری به طور کلی به موارد زیر تقسیم می‌شود:

- لایه ترمینال
- لایه لبه
- لایه محاسباتی

لایه ترمینال از تمام دستگاه‌هایی که به شبکه لبه متصل هستند، از جمله پایانه‌های تلفن همراه و دستگاه‌های اینترنت اشیا مانند حسگرها، گوشی‌های هوشمند، خودروهای هوشمند و دوربین‌ها تشکیل شده است. این دستگاه‌ها ممکن است نقش مصرف‌کنندگان داده و ارائه‌دهندگان داده را بازی کنند.

لایه مرزی یا لایه لبه در لبه شبکه قرار دارد و از گره‌های لبه‌ای تشکیل شده است که بین دستگاه‌های ترمینال و ابرها شامل ایستگاه‌های پایه، نقاط دسترسی، روترها، سوئیچ‌ها و دروازه‌ها توزیع شده‌اند.

در نهایت، لایه رایانش ابری قدرتمندترین مرکز پردازش داده را در معماری مورد نظر، متشکل از سرورها و دستگاه‌های ذخیره‌سازی با کارایی بالا، با قابلیت‌های محاسباتی و ذخیره‌سازی پیشرفته، تعبیه می‌کند: اینها وظیفه تجزیه و تحلیل داده‌ها، نگهداری منظم و پشتیبانی تصمیم را بر عهده دارند.

محاسبات لبه از نظر پردازش و تجزیه و تحلیل سریع داده‌ها در زمان

[1] Radio Access

[2] On-Premises

واقعی مزایایی نسبت به رایانش ابری دارد: در عوض رشد سریع حجم داده و فشار پهنای باند شبکه از معایب اصلی محاسبات ابری است [۸]. از این نظر، محاسبات لبه نزدیکتر به منبع داده، ذخیره‌سازی داده‌ها و وظایف محاسباتی را می‌توان در گره محاسباتی لبه انجام داد، که فرآیند انتقال داده میانی را کاهش می‌دهد. محاسبات لبه خدمات هوشمند با کارایی بالا را در اختیار کاربران قرار می‌دهد، بنابراین عملکرد انتقال داده‌ها را بهبود می‌بخشد، پردازش بلادرنگ را تضمین می‌کند و زمان تاخیر را کاهش می‌دهد.

بنابراین تفاوت اصلی در نتیجه این امر در این است که محاسبات لبه برای تجزیه و تحلیل هوشمند مقیاس کوچک در زمان واقعی و با فشار پهنای باند کم شبکه مفید است. در مقابل، محاسبات ابری با وجود نمایش فشار پهنای باند شبکه بسیار بالاتر، پردازش متمرکز در مقیاس بزرگ را هدف قرار داده است.

در سال‌های اخیر بسیاری از شرکت‌ها اعلام کرده‌اند که استفاده از فناوری جدید را در کسب‌وکار خود آغاز کرده‌اند یا به‌دلیل مزایای آن در افزایش عملکرد کسب‌وکار و برآورده کردن نیازهای مشتریان، قصد دارند به زودی از آن استفاده کنند. از سوی دیگر، بسیاری از مقالات، مقالات و بررسی‌های دانشگاهی برای پیشنهاد زیرساخت‌های جدید با استفاده از محاسبات لبه برای سیستم‌های موجود یا برای اثبات امکان‌سنجی، بهره‌وری و عملکرد بالای برخی از آزمایش‌های عملی مبتنی بر چنین فناوری منتشر شده‌اند. علاوه بر این، پلتفرم‌های لبه‌های متعدد در حال حاضر در بازار استفاده می‌شوند و بسیاری از شرکت‌ها با استفاده از سرورهای قدرتمندتر، دروازه‌ها و راه‌حل‌های دیگر شروع به تغییر کسب‌وکار خود به لبه کرده‌اند.

محاسبه لبه کاهش مربوطه را در تأخیر بین پردازش داده‌ها در مراکز ابری و بهره برداری از آنها در مکان‌های کاربر نهایی، متعادل کردن ترافیک شبکه، اجتناب از گلوگاه شبکه و کاهش زمان پاسخ برای برنامه‌های حساس به زمان و واقعی را امکان‌پذیر می‌کند.

محاسبات لبه موبایل یک معماری نوظهور است که در آن خدمات رایانش ابری تا لبه شبکه‌ها به ایستگاه‌های پایه موبایل گسترش می‌یابد. فناوری لبه می‌تواند برای سناریوهای تلفن همراه، بی‌سیم و با سیم، با استفاده از پلت‌فرم‌های نرم‌افزاری و سخت‌افزاری که در لبه شبکه در مجاورت کاربران نهایی قرار دارند، اعمال کرد. محاسبات لبه موبایل یکپارچه‌سازی ارائه‌دهندگان و فروشندگان خدمات برنامه‌های کاربردی متعدد را برای مشترکین تلفن همراه، شرکت‌ها و سایر بخش‌های عمودی فراهم می‌کند. این یک جزء مهم در معماری 5G پیشنهادی است که از انواع برنامه‌ها و خدمات نوآورانه در مواردی که تأخیر بسیار کم مورد نیاز است، پشتیبانی می‌کند. با این حال، برخی از چالش‌ها در سیستم محاسبات لبه موبایل وجود دارد. برای رسیدگی به این چالش‌ها، ابتدا باید زیرساخت شبکه محاسبات لبه موبایل ، ابر و شبکه سلولی را درک کنید. اصطلاح «محاسبات لبه موبایل» برای اولین بار در سال ۲۰۱۳ معرفی شد، زمانی که شبکه‌های نوکیا زیمنس[1] و IBM پلت فرم محاسبات لبه موبایل را توسعه دادند که برنامه‌ها را قادر می‌سازد مستقیماً اجرا شوند. این پلت فرم تنها محدوده محلی را تسریع می‌کند که از مهاجرت برنامه، قابلیت همکاری و غیره پشتیبانی نمی‌کند [۹]. بعداً، در سال ۲۰۱۴، محاسبات لبه موبایل توسط موسسه استانداردهای مخابراتی اروپا[2] (ETSI) گروه مشخصات صنعتی[3] (ISG) استاندارد شد، این گروه شامل شبکه‌های نوکیا، اینتل، ووداfون، آی بی ام، هواوی و ان‌تی‌تی دوکومو[4] است. محاسبات لبه موبایل همچنین توسط زیرساخت 5G مشارکت عمومی‌خصوصی[5] به عنوان یک فناوری نوظهور اصلی برای شبکه‌های 5G تأیید شده است [۱۰].

در حال حاضر از پلتفرم‌های لبه‌های متعدد در بازار استفاده می‌شود و بسیاری از شرکت‌ها با استفاده از سرورهای قدرتمندتر، دروازه‌ها و راه حل‌های دیگر شروع به تغییر تجارت خود به لبه کرده‌اند. نمونه‌ای از این موارد در ادامه مطرح شده است[۱۱]:

در [۱۲] مسئله قرارگیری سرورهای لبه مطالعه شد و الگوریتمی‌پیشرفته پیشنهاد کردند که در آن مکان‌های سرورهای لبه به گونه‌ای بهینه می‌شوند که علاوه بر کاهش تعداد، استفاده از سرورهای نصب‌شده را بهبود بخشد. برای حل این مسئله، نویسندگان فرض می‌کنند که شبکه لبه‌های تلفن همراه به‌عنوان یک گراف بدون جهت رفتار می‌کند.

در [۱۴،۱۳] نویسندگان الگوریتم بهینه‌سازی ازدحام ذرات (PSO) را برای شبکه لبه تلفن همراه استفاده کردند. در این پژوهش هر نود لبه باید یک سرور لبه متناظر داشته باشد و به دنبال راه حلی بهینه برای افزایش کاربرد سرورها و در عین حال کاهش تعداد کل آنها هستند. راه حل این مقاله یک رابطه انتساب یک به چند را بین سرور نصب شده در شبکه از یک سو و گره‌های لبه از سوی دیگر

[1] Siemens Nokia
[2] European Telecommunications Standards Institute
[3] Industry Standards Group
[4] NTT DOCOMO
[5] G5 Infrastructure Public Private Partnership

تعریف می‌کند و بهترین راه حل موقعیت یابی را پیدا می‌کند که مصرف انرژی CPU سرورها را کاهش می‌دهد

در [۱۵]، یک الگوریتم اکتشافی برای مکان استراتژیک سرورهای لبه بر اساس پیش‌بینی نیاز منابع استفاده کردند. نویسندگان با پیش‌بینی مقصد بعدی این منبع داده، علاوه بر استفاده از مکانیزم نام‌گذاری داده‌ها، تعداد مناسبی از مکان‌های سرور ممکن را برای یک منبع داده خاص (موبایل، ایستگاه اتوبوس، رایانه شخصی، ایستگاه‌های بازی و غیره) تعیین می‌کنند.

در [۱۶]، یک مدل تحلیلی کارآمد برای مهاجرت سرویس خود تطبیقی در شبکه‌های محاسباتی لبه پیشنهاد شده است. هدف نویسندگان این است که مکان سرویس را که توسط گره‌های لبه ارائه می‌شود، تا حد امکان نزدیک به مکان کاربر نهایی از طریق سازگاری پویا با هدف به حداکثر رساندن کارایی عملکرد شبکه نگه دارند. میانگین درصد مطلق خطا به عنوان یک شاخص عملکرد کلیدی در نظر گرفته شده است.

یک پلتفرم محاسباتی نوآورانه به نام DNR[1] در [۱۷] به عنوان توسعه ابزار منبع باز معروف پیشنهاد شده است که می‌تواند جریان داده بصری را برای برنامه‌های اینترنت اشیا فراهم کند. سه نسخه از این پلتفرم قبلاً آزمایش شده است که هر کدام برای حل یک چالش خاص استفاده می‌شود. آزمایش‌های DNR راه‌حل‌های جدیدی برای داشتن یک پلتفرم برون‌زا ابداع کردند تا بازده محاسباتی را از طریق شبکه‌های لبه افزایش دهند.

در [۱۸]، یک مدل نظری با هدف تصمیم‌گیری در مورد زمان استفاده از ابر یا لبه و یا فقط لبه یا ترکیبی و همچنین زمانی که بهتر است بر معماری ابر سنتی تکیه کرد پیشنهاد شده است. مدل پیشنهادی عمدتاً مبتنی بر پارامترهای زیر است: نسبت گزینش، نسبت محاسبات-ارتباطات و سرعت پردازش ابری. این مقاله ثابت می‌کند که وقتی پارامتر افزایش سرعت ابر کم است و نسبت محاسبات به ارتباط کوچک است، سیستم‌های فقط لبه سریع‌تر از سیستم‌های فقط ابری هستند. هنگام مقایسه محاسبات ترکیبی ابر-لبه با سیستم‌های فقط ابری، مشخص می‌شود که راه‌اندازی ترکیبی سریع‌تر زمانی است که پارامتر افزایش سرعت ابر-لبه هیبریدی مربوطه بیشتر از یک باشد. نویسندگان عملکرد سه معماری سیستم قبلی را روی یک چارچوب خاص تجزیه و تحلیل می‌کنند که در آن دو تابع MapReduce بر روی یک معماری سیستم ترکیبی اجرا می‌شوند، اولی در لبه و دو می‌در ابر. سپس عملکرد بر روی هفت برنامه واقعی آزمایش می‌شود که سه مورد از آنها اجرای ترکیبی را پشتیبانی ن می‌کنند. این روش به عنوان فناوری محاسبات ابری مربوطه برای این آزمایش انتخاب شده است، در حالی که از دو دستگاه کم مصرف برای نمایش اجزای لبه استفاده می‌شود. زمانی که پهنای باند در محاسبات ابری در مقایسه با سیستم‌های فقط لبه‌ای کم باشد، تنها یک برنامه زمان اجرای سریع‌تری را نشان می‌دهد، در حالی که برای پهنای باند بزرگ، همه برنامه‌های کاربردی دیگر در سیستم‌های فقط لبه در مقایسه با سیستم‌های فقط ابری زمان پردازش سریع‌تری دارند. هنگا می‌که برنامه دارای جابجایی داده کم است، راه‌اندازی فقط ابری همیشه سریع‌تر است، در حالی که سیستم لبه-ابر هیبریدی برای ورودی‌های بزرگ و اندازه داده‌های متوسط کوچک همیشه سریع‌تر است. در مجموع، مدل پیشنهادی چارچوبی مفید برای تصمیم‌گیری برای اجرای وظایف با استفاده از سناریوی سنتی رایانش ابری یا انتقال حجم کار به لبه-ابر فقط لبه یا ترکیبی ارائه می‌کند.

در [۱۹]، چالش‌های اصلی از نظر توسعه برنامه با تکیه بر فناوری محاسبات لبه بررسی شده و یک راه‌حل کارآمد با استفاده از فناوری پیشنهاد شده است. پیاده‌سازی یک برنامه کاربردی در شبکه‌های لبه نیازمند برآورده کردن برخی از الزامات سخت‌افزاری خاص از نظر اتصال به شبکه، قدرت CPU و ظرفیت است که به طور قابل‌توجهی بیشتر از برنامه‌هایی است که روی رایانه‌های رومیزی معمولی اجرا می‌شوند. به ویژه، وابستگی برنامه‌های محاسباتی لبه به دست‌آوری داده‌های بلادرنگ از برنامه‌های دیگر فروشندگان، چالش مصرف زمان جدیدی را از نظر تامین‌کنندگان داده در زمانی که دومی‌ها در منابع داده خود مشکل دارند، ایجاد می‌کند. علاوه بر این، پس از هر تغییری در برنامه، فرآیند تکراری ظاهر می‌شود که شامل شروع مجدد چرخه عمر فرآیند کانتینری‌سازی از ابتدا و ارسال نسخه جدید کد به دستگاه‌های لبه است. نویسندگان یک روش اشکال زدایی از راه دور را پیشنهاد می‌کنند که با تمام کتابخانه‌ها و بسته‌های برنامه نویسی مورد نیاز در گره لبه راه دور ایجاد می‌کند.

علاوه بر این، نسبت به حوزه خدمات اورژانس پزشکی، یک کاربرد تشخیص ضربان قلب مبتنی بر ویدیو بر روی یک پلت فرم محاسباتی لبه‌ای در [۲۰] پیشنهاد شده است که از چهار موجودیت تشکیل شده است: تلفن هوشمند (Huawei Honor 8)، ایستگاه پایه (مطابق با پروژه منبع باز سیستم سلولی 5G SoftRAN)، یک سرور لبه و یک سرور ابری. آزمایش برنامه از ارسال فیلم‌های چهره از تلفن

[1] Distributed Node Red

هوشمند کاربر به سرور لبه‌ای که در آن پیش‌پردازش داده‌ها انجام می‌شود، شروع می‌شود، در حالی که سایر ویژگی‌ها برای تجزیه و تحلیل پیچیده‌تر به ابر ارسال می‌شوند، جایی که سرورهای ابری این مسئولیت را بر عهده می‌گیرند. در نهایت، نتیجه ترکیبی یک اندازه‌گیری قابل اعتماد ضربان قلب را به گوشی هوشمند باز می‌گرداند. نتایج تجربی تاکید کردند که زمان پاسخ این معماری ترکیبی لبه-ابر به ترتیب ۲۰ و ۴۰ درصد کمتر از استفاده از فناوری لبه و تنها فناوری ابر است. این نشانه‌ای است که علیرغم بسیاری از انتظارات فعلی مبنی بر اینکه محاسبات لبه جایگزین فناوری ابری خواهد شد، آزمایش‌های عملی به طور مداوم نشان می‌دهد که بهترین عملکرد را می‌توان از طریق ترکیب آنها به دست آورد.

در [۲۱] یک شبکه عصبی را بر روی یک پلتفرم محاسبات لبه با پشتیبانی از VPU (واحد پردازش بینایی) اجرا می‌کند. در این پژوهش نشان داده شد که چگونه هنگام استفاده از دستگاه‌هایی با سیستم عامل‌های مختلف و با قدرت‌های مختلف کار می‌کند. یک پردازنده چهار هسته‌ای به عنوان یک دستگاه لبه استفاده شد. این پردازنده چهار هسته‌ای ۱,۲ گیگاهرتز ۶۴ بیتی، چهار پورت USB 2.0 از طریق هاب USB پنج پورت داخلی و ۱ گیگابایت RAM کم مصرف دارد و حداکثر ۶,۷ وات مصرف می‌کند. در اوج بار با این حال، یک نورال محاسبات استیک به عنوان دستگاه یادگیری عمیق آزمایش استفاده شد که توسط همان ویژگی‌های قدرتمند و عملکردی تامین می‌شود.

خلاصه‌ای از این روندهای بررسی شده در ادامه در قالب جدول ۱ نمایش داده شده است.

۳- روش پیشنهادی

در این پژوهش یک سیستم محاسبات لبه موبایل با دو ناحیه تعریف می‌کنیم که شامل چندین دستگاه تلفن همراه و سرورهای لبه تعبیه شده در ایستگاه پایه مخابراتی است. همچنین، فرض می‌کنیم که وظایف بارگذاری شده‌اند و قبلاً به سرور لبه رسیده‌اند. بر این اساس، این بخش فرآیند در مدل محاسبات لبه شامل محاسبات، وظایف بارگذاری شده، مدیریت زمان‌بندی وظایف و مدیریت منابع لبه را شامل شده است که به دلیل هدر رفتن قابلیت سرور لبه ایجاد می‌شود و مسئله زمان‌بندی کار را برای کاهش تأخیر محاسبات و به حداقل رساندن هزینه سرویس حل می‌کنیم.

اولین ماشین مجازی برای برآوردن نیازهای پردازشی، وظیفه بارگذاری شده ایجاد شده است. به نوبه خود، واحد کنترل ماشین‌های مجازی یک نمونه از ماشین‌های مجازی اصلی ایجاد می‌کند تا ماشین‌های مجازی دیگر را تولید کند، در نتیجه قابلیت‌های باقی مانده شبکه با مشخصات مشابه ماشین‌های مجازی اول را اشغال می‌کند.

ما فرض می‌کنیم که یک سرور لبه دارای منابع محدود با قدرت محاسباتی متفاوتی است. ما همچنین فرض می‌کنیم که هر سرور لبه می‌تواند چندین نمونه ماشین مجازی را اجرا کند و به آنها اجازه می‌دهد وظایف بارگذاری شده ارسال شده توسط چندین دستگاه موبایل را محاسبه کنند. الگوریتم مدیریت تخصیص منابع در سرور لبه در فواصل زمانی منظم و برنامه‌ریزی شده اجرا می‌شود. سرور لبه دارای چهار واحد اصلی است: واحد شبیه‌سازی برنامه، واحد MARC، واحد تحرک و ارتباطات، و واحد برنامه‌ریزی و تعادل.

کنترل کننده محاسبات ابری لبه کنترل، مسئول پیچیده‌ترین محاسبات است: یعنی یافتن نگاشت بین سرور لبه‌هایی که نزدیک به یکدیگر هستند و همچنین نگاشت بهینه چندین دستگاه موبایل به سرور لبه‌هایی که به عنوان انجمنی شناخته می‌شود. علاوه بر این، نگاشت وظایف بارگذاری شده به چندین دستگاه موبایل باید انجام شود زیرا هر وظیفه حاوی ((XId,Data,Did) است، که در آن XId شناسه وظیفه و Did شناسه دستگاه موبایل است که برنامه از آنجا ارسال می‌شود. هنگامی‌که دستگاه موبایل حرکت می‌کند محاسبات ابری لبه کنترل نتایج را از طریق نزدیکترین سرور لبه به دستگاه موبایل می‌دهد؛ این روش با تعیین مکان دستگاه موبایل مقصد و اتصال آن به نزدیکترین سرور لبه به دست می‌آید.

واحد تحرک و ارتباطات حاوی اطلاعاتی در مورد تحرک گذشته هر دستگاه موبایل است که به سرور لبه متصل است و همچنین مکان‌های جغرافیایی (Xe,Ye). علاوه بر این، واحد تحرک الگوهای تحرک دستگاه را برای پیش‌بینی سرعت و جهت در مراحل زمانی آینده بر اساس اطلاعات تاریخی دستگاه موبایل تجزیه و تحلیل می‌کند. واحد زمان‌بندی و متعادل‌سازی درخواست، تمام وظایف بارگذاری‌شده را دریافت کرده و آنها را پس از یک بازه زمانی مشخص ε زمان‌بندی می‌کند. بنابراین چالش‌های بیشتر برای حل مشکل تخصیص منابع از زمان‌های پاسخ متفاوت به کار tp کار و همچنین حرکت دستگاه موبایل ناشی می‌شود.

چندین ماشین مجازی با منابع محدود در سرور لبه ایجاد می‌شوند تا به دستگاه موبایل‌ها سرویس دهند. هر دستگاه می‌تواند به طور مستقل وظایف خود را از طریق سرور لبه ارسال کند، در حالی که هر سرور لبه که برنامه‌ها را مستقر می‌کند می‌تواند وظایف را همزمان پردازش کند.

جدول ۱. خلاصه‌ای از این روندهای بررسی شده

مجموعه داده و شبیه‌ساز	نتایج و بحث	نقاط قوت	نقاط ضعف	نام نویسنده و سال	مرجع
در این پژوهش روی دو مجموعه داده آزمایشی با تعداد پردازنده‌ها و گره‌های لبه و ابری آزمایش صورت گرفته است.	الگوریتم ابتکاری انتخاب سرور مبتنی بر رویکردی جالب برای انتخاب بهترین ارائه دهنده خدمات پیش بینی نیاز منابع	استفاده از یک الگوریتم ابتکاری	با پیش‌بینی مقصد بعدی در مسئله باعث یک بعدی شدن مسئله و کاهش عملکرد می‌شود.	ویرا و همکارانش[1] (۲۰۱۹)	[15]
مجموعه داده مورد استفاده در این پژوهش بر اساس تعداد معرفی شده در مقاله متفاوت است.	مهاجرت خود سرویس تطبیقی در شبکه لبه منجر به ترافیک داده کمتر می‌شود			یانگ و همکارانش[2] (۲۰۱۸)	[16]
مجموعه داده مورد استفاده بر اساس تعداد پردازنده‌ها از ۴، ۸، ۱۲، ۱۶، ۲۰ و ۲۴ با ۲۰، ۴۰، ۸۰، ۱۶۰، ۳۲۰ و ۴۰۰ کار متفاوت است.	این افزونه ابزار منبع باز Node Red است Node Redرا قادر می‌سازد تا در شبکه لبه توزیع شده اعمال شود و گره‌ها را به صورت موازی اجرا کند. برنامه‌های کاربردی پیچیده‌تر و با توزیع جغرافیایی بالاتر و راندمان محاسباتی بهتر دارد.	استفاده از رویکرد Node Red قابلیت الگوریتم را در این مسئله بالا میبرد.	راه‌حل پیشنهادی پیچیدگی محاسباتی بالایی دارد و با افزایش تعداد منابع با مشکل روبه رو خواهد شد.	الرووالی[3] و همکارانش(۲۰۱۸)	[17]
مجموعه داده معرفی شده در این روند با کمک یک ابزار شبیه‌سازی مبتنی بر FogSim تولید شده است.	راه‌حل‌های بهینه‌سازی شده برای کاهش تخصیص سرور پویا ارائه می‌کند، تعداد عملکرد سرورها را به جای راه‌حل‌های استاتیک فعلی بهبود می‌بخشد.	استفاده از یک گراف بدون جهت برای حل مسئله	راه‌حل پیشنهادی پیچیدگی محاسباتی بالایی دارد.	لی و همکارانش[4] (۲۰۱۸)	[18]
در شبیه‌سازی از مجموعه داده LCG استفاده شده است.	راه‌حلی با استفاده از فناوری اشکال زدایی	رویکرد خودآموزنده پیشنهادی دقت مسئله را در پارامترها افزایش می‌دهد.	در نظر گرفتن مولفه فاصله می‌تواند نتایج این روند را بهبود دهد.	ژانگ و همکارانش[5] (۲۰۱۹)	[19]
در شبیه‌سازی از مجموعه داده LCG استفاده شده است.	آثاری که زمان پاسخ را هنگام بارگذاری فقط در فضای ابری، فقط با لبه و هر دوی آنها مقایسه می‌کنند نتایج آزمایش نشان می‌دهد که بهترین عملکرد را می‌توان با ترکیب بین فناوری‌های لبه و ابری، بدون استفاده از یکی از آنها به دست آورد	در نظر گرفتن فضای ابری، فقط با لبه و هر دوی آنها	نیاز به قدرت تصمیم و تحلیل خاصی برای مسئله دارد.	اسچوارچ و همکارانش[6] (۲۰۱۸)	[20]
مجموعه داده بر اساس تعداد کارهای متفاوت و منابع متفاوت ایجاد شده است.	در این پژوهش مشخص می‌شود که هنگام استفاده از دستگاه‌هایی از سیستم عامل‌های مختلف و داشتن قدرت‌های مختلف چگونه کار می‌کند.	در نظر گرفتن مسئله حجم داده‌ها و اطلاعات	ارائه مسائل به صورت کلی	ژانگ و همکارانش[7] (۲۰۲۰)	[21]

[1] Vieira et al. , 2019
[2] Yang
[3] Alrowaily
[4] Li et al.
[5] Xing
[6] Schwartz
[7] Zhang

دستگاه موبایل یک وظیفه X را به سرور لبه بارگذاری کرده است، پس از آن، این کار با استفاده از واحد حل‌کننده برای انجام همزمان وظایف با قرار دادن آن در ماشین‌های مجازی بدون اتلاف اجرا می‌کند. بر این اساس، ماشین‌های مجازی همانطور که در بالا ذکر شد پر می‌شوند تا وظایف فرعی روی ماشین‌های مجازی را یک به یک پردازش کنند (یعنی در نتیجه هزینه سرویس و زمان سرویس از طریق پردازش همزمان کاهش می‌یابد). بنابراین، X,xυ,2,...,xv,j داریم که xυ بردار نشانگر وظیفه υth VM و xv,j نشانگر زیرکار j است. اگر ماشین مجازی (υ) وظایف فرعی j را انجام دهد، mvj = 1 در غیر این صورت، mvj = 0. جزئیات کار برای همه ماشین‌های مجازی را می‌توان با استفاده از یک ماتریس نشانگر وظیفه X = {X1,X2,....,Xn} بیان کرد، که در آن υامین ردیف بردار نشانگر وظیفه سرور لبه است. υ بر این اساس، با توجه به رابطه عرضه و تقاضا بین وظایف بارگذاری شده و ماشین‌های مجازی روی سرور لبه، هر وظیفه برای دریافت خدمات از منابع هر VM پیشنهاد می‌کند. ماشین‌های مجازی با ظرفیت محدود به طور همزمان درخواست‌هایی را برای منابع خود ارسال می‌کنند.

زمان‌بندی منابع برای به حداقل رساندن تأخیر محاسباتی برای وظایف بارگذاری شده به سرور لبه و اطمینان از اینکه منابع فعلی می‌توانند بیشترین تعداد کارها را با کمترین هزینه انجام دهند، مورد نیاز است. کنترل مدیریت تخصیص منابع با تخصیص وظایف ارسالی از واحد برنامه‌ریزی منابع به ماشین‌های مجازی مناسب مرتبط است، کنترل مدیریت تخصیص منابع با توجه به عملکرد CPU انتخاب می‌شود. این قابلیت‌های باقی‌مانده ماشین مجازی را اشغال می‌کند، که تضمین می‌کند که منابع هدر نمی‌روند، که به آن اتلاف بیش از توانایی سرور لبه می‌گویند. هر کار از یک ماشین مجازی برای انجام پردازش موازی برای کاهش زمان پردازش استفاده می‌کند، اما نتیجه آن اتلاف بیش از توانایی سرور لبه است. این هزینه خدمات را افزایش می‌دهد و وظایف در صف انتظار می‌رود تا ماشین‌های مجازی موجود برای دریافت وظایف جدید منتظر بمانند و همچنین تاخیر محاسباتی را تشدید می‌کند. کنترل مدیریت تخصیص منابع همچنین مسئول انتقال وظایف بین ماشین‌های مجازی است. هر زمان که وظیفه‌ای از سرور لبه درخواست منابع می‌کند، کنترل مدیریت تخصیص منابع با VM دیگری ارتباط برقرار می‌کند و منابع VM مناسب را برای انجام وظایف اختصاص می‌دهد. این به عنوان زمان بندی منابع در بین ماشین‌های مجازی شناخته می‌شود.

زمان‌بندی منابع در بین ماشین‌های مجازی از دو اصل پیروی می‌کند: اول، در دسترس بودن و استفاده از منابع ماشین‌های مجازی، و دوم، قابلیت ماشین‌های مجازی برای انجام وظایف ارسالی. اگرچه حفظ منابع منجر به انجام بیشترین تعداد کارها با کمترین هزینه می‌شود، اما در صورت استفاده بیش از حد از قابلیت VM تلف شده، کاربر هزینه خدمات اضافی را متحمل نخواهد شد. به دلیل اینکه پردازش همزمان اتفاق می‌افتد، زمان سرویس نیز کاهش می‌یابد. در نتیجه، هنگام زمان‌بندی منابع در بین ماشین‌های مجازی، لازم است استفاده از منابع سرور لبه نیز در نظر گرفته شود.

ما فرض می‌کنیم دستگاه تلفن همراه که درخواست بارگذاری به سرور لبه را دارد، برنامه فشرده در آن وجود دارد. این مبنایی برای طراحی برنامه‌ای است که در آن یک بخش شامل پردازش محلی است و بخشی دیگر نیاز به تخلیه محاسباتی به یک سرور لبه نزدیک دارد. هنگامی که وظایف به سرور لبه می‌رسند، در یک صف برنامه‌ریزی شده برای شروع پردازش قرار می‌گیرند. هر کار به وظایف فرعی تقسیم می‌شود و ماشین‌های مجازی در داخل سرور لبه ایجاد می‌شوند تا وظایف را به صورت موازی اجرا کنند. با توجه به تحرک دستگاه تلفن همراه، فرض می‌کنیم که پس از بارگذاری کار به سرور لبه، که در ناحیه پوشش سرور لبه قرار دارد، دستگاه تلفن همراه حرکت می‌کند و از منطقه پوشش اولیه خارج می‌شود. دستگاه تلفن همراه نتایج کار را از سرور لبه فعلی دریافت نمی‌کند. بنابراین، چالش اصلی این است که چگونه نتایج را به دستگاه تلفن همراه برگردانیم و نیازهای زمانی کار را برآورده کنیم.

ما فرض می‌کنیم که یک کنترل شبکه وجود دارد که با ردیابی دستگاه تلفن همراه که وظایف را تخلیه کرده است کار می‌کند. همچنین می‌تواند مدت زمانی که دستگاه تلفن همراه در ناحیه تحت پوشش سرور لبه باقی می‌ماند و سرویس در سرور لبه را پیش‌بینی کند، به‌علاوه جهت بعدی دستگاه تلفن همراه را برای دریافت نتیجه پردازش پیش‌بینی می‌کند بنابراین دستگاه تلفن همراه را قادر به دریافت خروجی از طریق نزدیک‌ترین سرور لبه که وارد محدوده خدمات آن می‌شود می‌کند.

ما فرض می‌کنیم که برنامه‌های کاربردی فشرده قبلاً بر روی سرور لبه، یکپارچه شده با یک ایستگاه پایه ارتباطی بارگذاری شده‌اند. این برنامه‌ها به عنوان مجموعه‌ای از وظایف فرعی X = {xi} نشان داده می‌شوند. مشکلات و چالش‌های اصلی را می‌توان به شرح زیر خلاصه کرد:

- وظایف در یک صف انتظار برای پردازش هر کار به طور جداگانه قرار می‌گیرند که منجر به افزایش زمان سرویس Te و تاخیر در پاسخگویی می‌شود. بنابراین، هدف ما رسیدن به

Minimize (Te) است.

- چالش‌های تحرک از نحوه نگهداری خدمات محاسباتی زمانی که دستگاه تلفن همراه از منطقه تحت پوشش اولیه سرور لبه خارج می‌شود، ناشی می‌شود. بنابراین، ما باید مدت زمان اتصال دستگاه تلفن همراه به سرور لبه و حداکثر ارتباط (AE,X) یک کار با سرور لبه را اندازه‌گیری کنیم.

۳-۱- روش حل مسئله

• هنگام ایجاد VMs V = {v1, v2, v3,...,vn}، هر زیرکار به یک ماشین مجازی برای پردازش موازی اختصاص داده می‌شود و در نتیجه زمان پردازش کاهش می‌یابد. با این حال، این منجر به اتلاف بیش از توانایی ES، همراه با افزایش هزینه خدمات (Ci) می‌شود. وظایف در صف انباشته می‌شوند تا منتظر ماشین‌های مجازی موجود برای دریافت وظایف جدید بمانند و تاخیر محاسباتی را افزایش دهند. بنابراین، هدف ما کاهش هزینه با به حداقل رساندن تعداد VMها، یعنی Minimize(ηj) است.

در این پژوهش از معیارهای قابلیت فرآیند (Cp) و عملکرد فرآیند (Pp) برای نظارت و تأیید توانایی سرور لبه برای ارائه وظایف بارگذاری شده استفاده می‌شود. Cp و Pp برای مقایسه الزامات فرآیند برای اجرای وظایف با عملکرد سرور لبه کار می‌کنند. اگر سرور لبه قادر به محاسبه وظایف بارگذاری شده باشد، سرورهای مجازی (یعنی ماشین‌های مجازی) ایجاد می‌شوند و وظایف به آنها اختصاص داده می‌شود. به نوبه خود، مقدار هر VM (مقدار VM، VMV) بر اساس سرعت CPU و اندازه حافظه اختصاص داده شده به آن تعیین می‌شود. اگر سرور مجازی (VS) بیش از حد کار کند، قادر به انجام وظایف فرعی نخواهد بود. در این پژوهش سعی می‌شود بر اساس بهبود رویکرد ازدحام ذرات و گرگ خاکستری مسئله زمانبندی حل شود.

شبه کد روند در شکل ۱ نمایش داده شده است:

در این پژوهش از یک روش ترکیبی بهینه‌سازی گرگ خاکستری و بهینه‌سازی ازدحام ذرات (GWO-PSO) برای دستیابی به راه‌حل بهینه استفاده می‌شود. PSO یک الگوریتم بهینه‌سازی فراابتکاری مبتنی بر ازدحام است که هدف آن جستجوی بهترین راه‌حل برای یک مسئله با حرکت ذرات در یک میدان اکتشافی خاص است. از سوی دیگر، GWO یک تکنیک بهینه‌سازی فراابتکاری است که از گرگ‌های خاکستری الهام گرفته شده است.

```
1:  input: Set input data, A(t) := Ø; t := 0
2:  output: Optimal solution

3:  Initialize
4:  while (not termination condition) do
5:     create P(t) by observing the state of Q(t)
6:     determine new positions and velocities of particles followed by create B(t)
7:  find Fonseca-Fleming ranks for an extended archive C(t) = A(t)∪B(t)∪P(t)
8:     calculate crowding distances, fitness and then sort particles in C(t)
9:     form A(t) of Pareto-optimal solutions from the sorted set C(t)
10: a tournament selection of an angle rotation matrix M_ϑ based on rating R(M_ϑ)
11: mutate the selected matrix M_ϑ with the rate p_m
12:    modify Q(t) using the rotation gates
13:    t := t + 1
14: end while
```

شکل ۱. شبه کد روند

PSO به عنوان یک تکنیک بهینه سازی فراابتکاری مبتنی بر ازدحام طبقه بندی می‌شود. در PSO، استراتژی اکتشاف برای یافتن بهینه‌سازی سراسری، یک استراتژی مبتنی بر جمعیت است. انگیزه آن ناشی از رفتار پرندگان است. ذرات ایجاد می‌شوند، سپس از طریق میدان اکتشاف به منظور تعیین جمعیت بهینه که بهترین راه‌حل مسئله را برآورده می‌کند، به اطراف منتقل می‌شوند. ذرات در یک میدان اکتشافی چند بعدی تولید می‌شوند و هر ذره از تجربه خود و همچنین تجربه ذرات مجاور خود برای تنظیم موقعیت خود استفاده می‌کند. همچنین، ذرات با موقعیت بهینه‌ای که خود و ذرات مجاورشان به آن می‌رسند هدایت می‌شوند. نقاط قوت PSO این است که اجرای آن آسان است و نیازی به تنظیمات بسیاری از پارامترها نیست.

GWO الهام گرفته از رفتار و استراتژی شکار گرگ‌های خاکستری در طبیعت است. در یک گروه، گرگ‌های خاکستری از سلسله مراتب رهبری قوی پیروی می‌کنند. رهبران این گروه گرگ آلفا (الف) نامیده می‌شوند. دسته بعدی گرگ‌های خاکستری جزء گرگ‌های ثانویه محسوب می‌شوند. آنها به آلفا کمک می‌کنند. آنها به عنوان گرگ بتا (ب) شناخته می‌شوند. علاوه بر این، گرگ‌های دلتا (d) دارای اولویت کمتری نسبت به گرگ‌های آلفا و بتا هستند. ماموریت آنها تسلیم شدن در برابر گرگ‌های آلفا و بتا است، اما کنترل گرگ‌های امگا. گرگ‌هایی که کمترین اولویت را دارند، امگاها (x) هستند که باید از گرگ‌های خاکستری پیشرو پیروی کنند. تکنیک GWO در نمایش‌های ریاضی به این صورت است:

در نمایش ریاضی سلسله مراتب گرگ‌های خاکستری، گرگ آلفا به

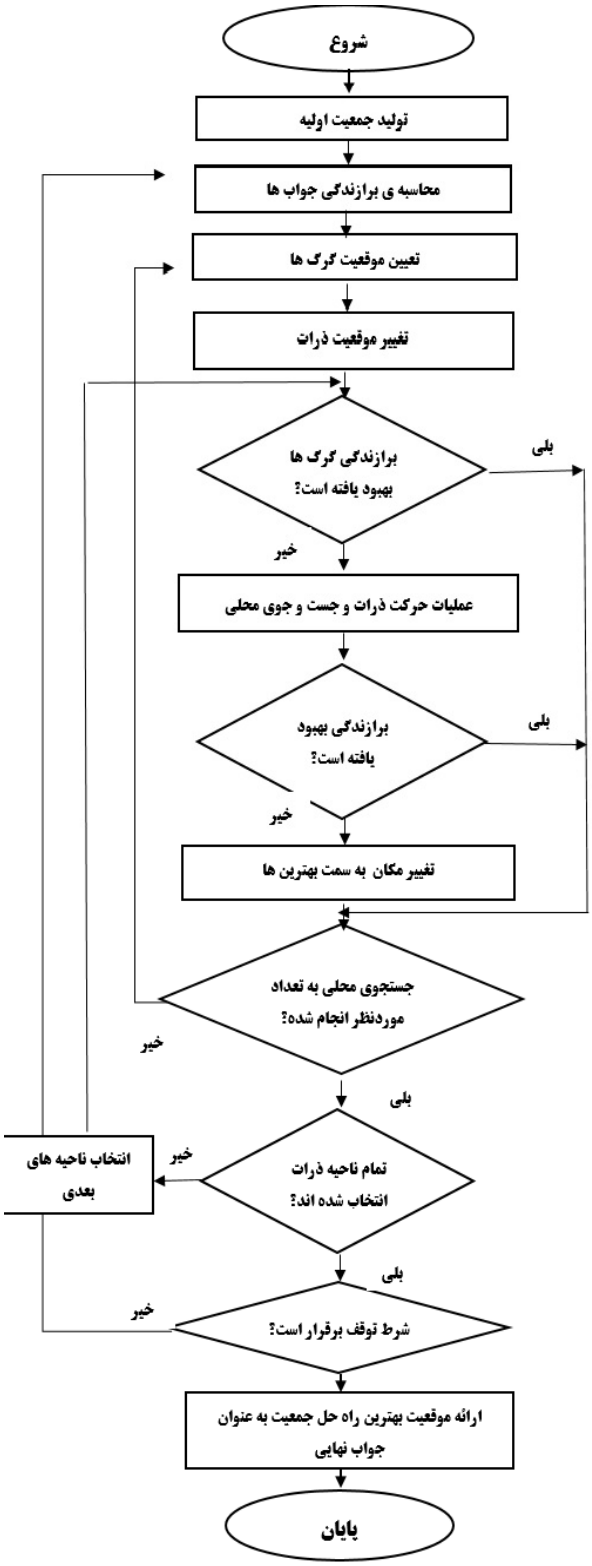
عنوان مناسب‌ترین راه‌حل شناخته می‌شود. در نتیجه بتا گرگ به عنوان دومین راه‌حل مناسب شناخته می‌شود و سپس راه‌حل مناسب بعدی دلتا نامیده می‌شود. سایر جمعیت‌هایی که دورترین راه‌حل‌ها را نشان می‌دهند، امگا (x) هستند. از طریق تکنیک GWO، فرآیند شکار توسط آلفا، بتا و دلتا هدایت می‌شود. امگاها فقط باید همین گام‌ها را بردارند و از گرگ‌های اولویت‌های بالاتر اطاعت کنند. گرگ‌های خاکستری هنگام شکار قربانی را محاصره می‌کنند. فرآیند شکار اغلب توسط آلفاها، بتاها و دلتاها هدایت می‌شود زیرا آنها تجربه بیشتری در تعیین موقعیت قربانی به دست می‌آورند. عوامل اکتشافی باقی مانده باید مکان عامل بهینه را دنبال کنند و مکان‌های خود را مطابق با آن تغییر دهند. بنابراین عوامل جستجو و تعریف منطقه راه‌حل در ادامه به صورت موردی بیان شده است:

- تکنیک GWO اجرا می‌شود.
- تولید کمترین مقادیر برای همه عوامل.
- این عوامل به عنوان نقاط اولیه به تکنیک PSO منتقل می‌شود.
- موقعیت‌های اصلاح شده به GWO برگردانده می‌شود.
- این مراحل تا رسیدن به معیارهای توقف تکرار می‌شود.

فلوچارت روند در شکل ۲ نمایش داده شده است:

الگوریتم PSO یکی از قدیمی‌ترین الگوریتم‌های بهینه‌سازی طبیعی است. در سال ۱۹۹۵ با ایده اولیه الهام گرفته از رفتار پرندگان، یعنی از تعامل اجتماعی بین اعضای، پیشنهاد شد. GWO یک الگوریتم نسبتاً جدید است که در سال ۲۰۱۴ پیشنهاد شد که عمدتاً از رفتار سلسله مراتبی حیوان گرگ خاکستری و استراتژی آن در شکار الهام گرفته شده است.

PSO با یک توزیع تصادفی شروع می‌شود، جایی که هر ذره، که نماینده یک عضو گروه است، در فضای جستجو حرکت می‌کند و کیفیت موقعیت فعلی خود را ارزیابی می‌کند. ایده اصلی این الگوریتم این است که در هر تکرار، ذرات بهترین موقعیتی را که در گذشته داشته‌اند به خاطر می‌آورند و در صورت یافتن موقعیت بهتر، جایگزین آن می‌شوند. این بهترین امتیازها بهترین‌های شخصی (pi) و بهترین امتیاز مطلق در بین آنها بهترین سراسری (g) نامیده می‌شود. این نکته عنصر حیاتی همکاری بین ذرات است. زیرا به تعیین جهت حرکت بعدی ذرات کمک می‌کند. این مورد در معادله ۱ توضیح داده شده است، که در آن شاخص itr برای تکرارها و i برای نمایه‌سازی ذرات است.



شکل ۲. فلوچارت روند پیشنهادی

$$x_i^{itr+1} = x_i^{itr} + wv_i^{itr} + c_1 r_1 \left(p_i - x_i^{itr}\right) + c_2 r_2 (\mathrm{g} - x_i^{itr}) \quad (۱)$$

در این معادله، موقعیت فعلی و سرعت جریان هر ذره به عنوان

پایه‌ای برای موقعیت استفاده می‌شود. دو عبارت آخر این معادله ذرات را هدایت می‌کند که همزمان به سمت بهترین موقعیت شخصی خود (pi) و به سمت بهترین سراسری (g) بروند. فاصله این دو نقطه از موقعیت فعلی تعیین می‌شود. c1 و c2 پارامترهای الگوریتم هستند و r1 و r2 مقادیر تصادفی بین ۰ و ۱ هستند. این جنبه تغییر بین ذرات را ایجاد می‌کند که به کاوش بهتر هر فضا کمک می‌کند و در نهایت، w عاملی است که با ادامه تکرارها کاهش می‌یابد. این نقش اساسی در کاهش اندازه سرعت در مراحل بعدی جستجو دارد و باعث می‌شود اعضای ذرات با گام‌های کوچک‌تر حرکت کنند، بنابراین جستجو را در یک منطقه کوچک متمرکز می‌کند.

الگوریتم GWO بر اساس رتبه بندی ذرات بر اساس کیفیت تابع هدف آنها است که چهار سطح α، β، δ و ω را پیشنهاد می‌کند. در جایی که ذره مربوط به بهترین تابع هدف در رتبه α قرار می‌گیرد، β به دومین راه‌حل برتر اختصاص داده می‌شود و δ به سومین راه‌حل برتر اختصاص داده می‌شود. بقیه راه‌حل‌های کاندید با امگا ω نشان داده می‌شوند. در این الگوریتم، بهینه‌سازی فرآیند شکار را با هدایت ذرات رتبه بندی شده، β و δ شبیه‌سازی می‌کند. بقیه این سه گرگ را دنبال می‌کنند و جستجو به دو مرحله تقسیم می‌شود. شکار و سپس حمله به طور مشابه، برای PSO، این الگوریتم با یک توزیع تصادفی شروع می‌شود، سپس از موقعیت فعلی و اطلاعات نتایج جستجو برای تعیین موقعیت‌های تکرار بعدی، که در (معادله ۲) محاسبه می‌شود، استفاده می‌کند.

$$x_w^{itr+1} = x_p^{itr} - A * D$$
$$D = \left|CR * x_p^{itr} - x_{wo}^{itr}\right| \quad (۲)$$

که در آن wo یک شاخص برای ذرات گرگ است و itr شاخص تکرار است. p برای موقعیت طعمه است. A و CR بردارهایی حاوی ضرایب هستند و با استفاده از معادله ۳ و ۴ محاسبه می‌شوند:

$$A = 2ar_1 - a, \quad (۳)$$
$$CR = 2r_2 \quad (۴)$$

که در آن r1 و r2 مقادیر تصادفی بین ۰ و ۱ هستند و a مقداری است که با ۲ شروع می‌شود و در طول تکرارها به صورت خطی به ۰ کاهش می‌یابد. این یک استراتژی برای تمرکز جستجو، در تکرار دیرهنگام، بر روی بهینه بالقوه است. موقعیت طعمه، ارائه شده در معادله۲ فرض می‌شود که در اطراف رهبران جمعیت، یعنی ذرات گرگ بالاترین رتبه باشد و D برای هر رتبه به صورت زیر محاسبه می‌شود:

$$D_a = |CR_1 * x_\alpha^{itr} - x_{wo}^{itr}|,$$
$$D_\beta = \left|CR_2 * x_\beta^{itr} - x_{wo}^{itr}\right|,$$
$$D_\delta = |CR_3 * x_\delta^{itr} - x_{wo}^{itr}|. \quad (۵)$$

که در آن x_α^{itr}، x_β^{itr} و x_δ^{itr} موقعیت‌های رهبران هستند. و CR_1، CR_2 و CR_3 بردارهای ضریب هستند که با مقادیر تصادفی متفاوت محاسبه می‌شوند. سه امتیاز جدید بر اساس نتایج معادله فوق محاسبه می‌شود. آنها x_1، x_2 و x_3 هستند.

$$x_1 = \left|x_\alpha^{itr} - a_1 D_\alpha\right|,$$
$$x_2 = \left|x_\beta^{itr} - a_2 D_\beta\right|,$$
$$x_3 = \left|x_\delta^{itr} - a_3 D_\delta\right|. \quad (۶)$$

موقعیت طعمه برای تکرار بعدی به عنوان میانگین سه نقطه محاسبه شده در بالا محاسبه می‌شود.

$$x_\rho^{itr+1} = \frac{x_1 + x_2 + x_3}{3} \quad (۷)$$

آخرین استراتژی این الگوریتم حمله نام دارد که هنگام محاسبه ضریب A مشخص می‌شود (معادله ۳). زیرا مقدار A از [a, 2a۲ -] متفاوت است و مقدار a از ۲ به ۰ کاهش می‌یابد. ایده این است که وقتی مقدار |A| بزرگتر یا مساوی ۱ است. فرض بر این است که ذرات گرگ به اندازه کافی به طعمه نزدیک هستند، یعنی بهینه سراسری. ذرات گرگ مجبور به حمله می‌شوند.

الگوریتم ترکیبی پیشنهادی، ویژگی‌های این دو الگوریتم را به شیوه‌ای تکاملی ادغام می‌کند و ایده این است که حیاتی‌ترین نقاط هر الگوریتم، یعنی توانایی‌های اکتشافی GWO و استراتژی جست و جوی PSO را در کنار هم قرار دهیم. در این الگوریتم ترکیبی، ثابت اینرسی به صورت زیر به معادلات حاکم معرفی می‌شود:

$$D_\alpha = \left|CR_1 * x_\alpha^{itr} - w x_{wo}^{itr}\right|,$$
$$D_\beta = \left|CR_2 * x_\beta^{itr} - w x_{wo}^{itr}\right|,$$
$$D_\delta = \left|CR_3 * x_\delta^{itr} - w x_{wo}^{itr}\right|. \quad (۸)$$

که در آن X1، X2 و X3 با معادله ۹ محاسبه می‌شوند. معادله زیر ترکیب PSO و GWO را به روشی مشابه نشان می‌دهد:

$$x_i^{itr+1} = x_i^{itr} + w v_i^{itr} + C_1 r_1 (X_\alpha - x_i^{itr}) + C_2 r_2 (X_\beta - x_i^{itr}) + C_3 r_3 (X_\delta - x_i^{itr}) \quad (۹)$$

که C_1، C_2 و C_3 ثابت هستند. هدف مقاله جاری کاهش فرآیند ایجاد ماشین‌های مجازی و همچنین تعداد ماشین‌های مجازی مورد استفاده برای انجام یک کار تخلیه‌شده است، تا با حفظ منابع، کاهش هزینه خدمات را کاهش دهیم. تابع هدف اولیه با قرار دادن بهترین مجموعه وظایف که باید روی منابع اجرا شوند، سعی در کاهش زمان و هزینه دارد.

$$ETC = \frac{TLI}{MIPS_{\,k}} \quad K = ۱.۲.۳ \ldots . \; i = ۱.۲.۳ \qquad (۱۰)$$

که در آن ETC به زمان اجرای مورد نیاز iامین کار در VM اشاره دارد که در آن N تعداد VM ها و N تعداد وظایف است. TLI طول کار i است. $MIPS_{\,k}$ سرعت پردازش اطلاعات ماشین‌های مجازی است.

تابع هدف به صورت زیر تعریف می‌شود:

$$FIT=MAX\{ETCLK\}*P \qquad (۱۱)$$

به عنوان مثال $\{V_۱، V_۲، V_۳، V_۴\}$ ماشین‌های مجازی ابری هستند و $\{T_۱، T_۲، T_۳، و\ T_۴\}$ وظایف فعال هستند. جمعیت ایجاد شده همه وظایف فعال در آرایه [j][j] است، که i تعداد وظایف و j تعداد ماشین‌های مجازی است. سپس تابع هدف راه‌حل هر ستون در آرایه محاسبه می‌شود. از این رو، ماشین مجازی بهینه برای هر کار بر اساس سرعت انتقال تخمین زده می‌شود. به عنوان مثال، $V_۲/T_۱$، $V_۳/T_۲$، $V_۴/T_۳$، و $V_۲/T_۴$ راه‌حل‌های بهینه برای هر وظیفه در آرایه هستند. بر اساس جمعیت الگوریتم پیشنهادی برای بهینه‌سازی زمان‌بندی وظایف بر اساس منابع ابری موجود با توجه به وظایف انتقال اعمال خواهد شد. P توان عملیاتی می‌باشد که در ادامه این رابطه مشخص شده است:

$$P=Speed_of_CPU*CPU_percentage \qquad (۱۲)$$

در این روند از جبهه پارتو برای ترکیب این توابع استفاده شده است.

۴- ارزیابی و مقایسه

به منظور ایجاد شبکه در این پژوهش از پارامترهای ایجاد شبکه در مقاله پایه استفاده شده است در شبیه‌سازی از مجموعه داده LCG استفاده شد. محیط شبیه‌سازی در این پژوهش محیط شبیه‌ساز کلود سیم است. در این بخش به بررسی نتایج روند بر اساس شبیه‌سازی صورت گرفته پرداخته خواهد شد، شرایط ماشین‌های مجازی استفاده شده در جدول ۲ و بار این ماشین‌های مجازی در جدول ۳ نمایش داده شده است.

فرآیند تکاملی را می‌توان برای مسائلی که راه‌حل‌های اکتشافی در دسترس نیستند یا به طور کلی به نتایج رضایت بخشی منجر می‌شوند، اعمال کرد. در نتیجه، الگوریتم‌های تکاملی اخیراً مورد توجه بیشتری قرار گرفته‌اند، به‌ویژه با توجه به روشی که ممکن است برای حل مسئله عملی به کار گرفته شوند.

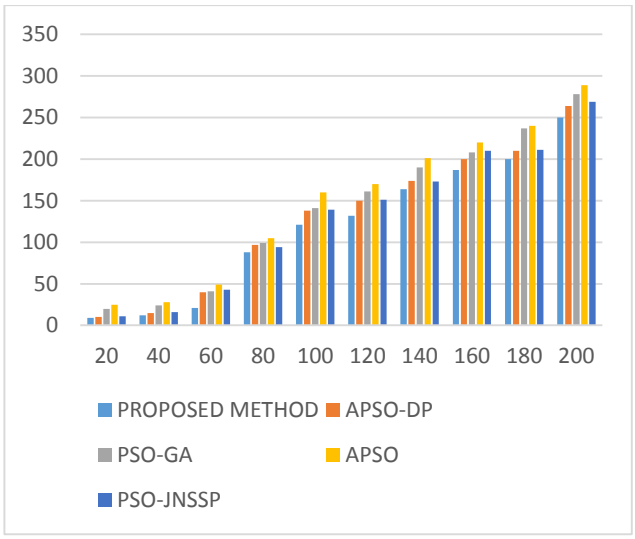
جدول ۲. مقداردهی پارامترهای مسئله

Parameter	Value
CPU MIPS	2400
Storage size	40000 MB
RAM size	1024
No. VMs	42
VM Price	20 cents
Population pso	10
Population GWO	15

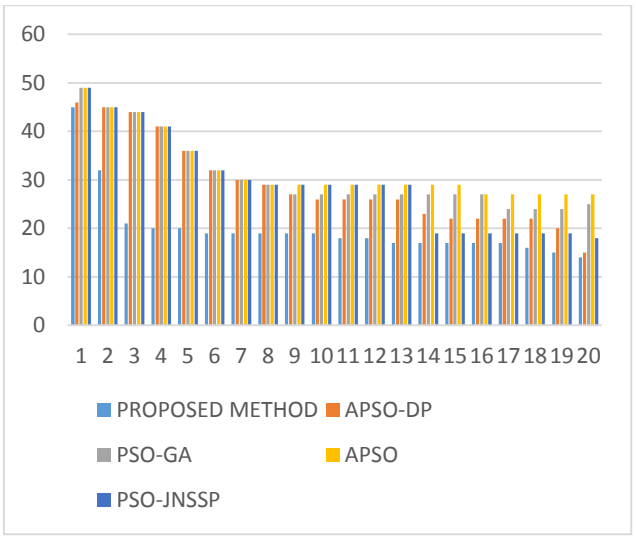
جدول ۳. مشخصات وظایف در این ماشین‌ها طبق مقاله پایه[۲۲]

Parameter	Value
Task number	1000-200000
Length(MB)	100-3000
MPIS	500-5000

محاسبات تکاملی، اصطلاحی که اکنون برای توصیف حوزه تحقیقاتی که مربوط به الگوریتم‌های تکاملی است، استفاده می‌شود، مزایای عملی را برای محققی که با مسائل دشوار بهینه‌سازی مواجه است، ارائه می‌کند. این مزایا چندگانه هستند، از جمله سادگی رویکرد، پاسخ قوی آن به شرایط در حال تغییر، انعطاف‌پذیری آن و بسیاری از جنبه‌های دیگر. بنابراین در رویکرد پیشنهادی نیز از این روند استفاده شده است. میانگین زمان انتظار با محاسبه تفاوت بین زمان بارگذاری یک کار به ES و زمان شروع اجرای کار اندازه‌گیری شد. در پژوهش مقاله پایه[۲۲] از الگوریتم ازدحام ذرات با توابع هدف خاص استفاده شده است در این پژوهش به منظور بهبود از توابع هدف جدیدی در کنار توابع دیگر استفاده شده است و به منظور بهبود جست‌وجوی محلی الگوریتم از الگوریتم گرگ خاکستری نیز استفاده شده است، الگوریتم‌های مقاله پایه شامل MOAPSO-DP که از سه الگوریتم زمان‌بندی پیشگیرانه با اولویت، APSO و برنامه‌نویسی (DP) استفاده می‌کند. الگوریتم APSO یکی از الگوریتم‌های بهینه برای مدیریت تخصیص منابع است که از PSO تکامل یافته است. در الگوریتم APSO استفاده از بهترین راه‌حل‌های فردی ضروری نیست مگر اینکه مسئله بهینه‌سازی مورد نظر غیرخطی و چندوجهی باشد و DP با متعادل کردن منابع بین ماشین‌های مجازی، هزینه‌های خدمات را کاهش می‌دهد. الگوریتم ژنتیک (GA) و الگوریتم‌های PSO نیز یک رویکرد ترکیبی هستند.الگوریتم PSO-JNSSP روند بهبود یافته حرکات ذرات در الگوریتم است[۲۳]. شکل ۳ نشان می‌دهد که زمان انتظار الگوریتم پیشنهادی کمتر از بقیه الگوریتم‌ها بوده است. تعداد وظایف در ۵۰ تعیین شد، اما تعداد VMها در هر تکرار بر اساس روش متعادل‌سازی منابع تغییر می‌کرد.



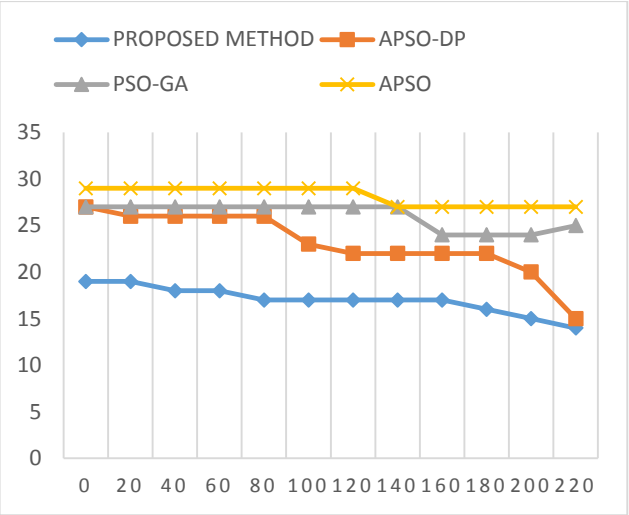
شکل ۳. میانگین زمان انتظار برای تعداد کارها



شکل ۴. میانگین هزینه بر اساس زمان در تکرار الگوریتم

شکل۳ میانگین زمان انتظار برای تعداد کارها را نشان می‌دهد در این روند نتایج الگوریتم پیشنهادی به ازای ۲۰ کار برابر با ۹ می‌باشد این در حالی است که این نتیجه برای الگوریتم پیشنهادی برابر با ۱۰ می‌باشد بهبود روند پیشنهادی نسبت به مقاله پایه به دلیل بهبود رویکرد جست و جو در الگوریتم ازدحام ذرات و درنظر گرفتن زمان انتظار برای انجام کارها به عنوان یک پارامتر و تاثیر آن در توابع هدف روش پیشنهادی است. در طول سالیان متمادی بهینه‌سازی در حوزه‌های مختلف سیستم قدرت به شدت مورد توجه مهندسین و محققین قدرت قرار گرفته است. محاسبات تکاملی در حل مشکلات دشوار و دنیای واقعی در صنعت، پزشکی رایج می‌شود. مزایای خاص روش پیشنهادی بر پایه الگوریتم تکاملی شامل انعطاف‌پذیری رویه‌ها و همچنین توانایی خودسازگاری جستجو برای راه‌حل‌های بهینه است. مقایسه هزینه سرویس بین الگوریتم‌های مقاله پایه و الگوریتم پیشنهادی در شکل ۴ نشان داده شده است. زمان‌بندی وظایف مبتنی بر روش پیشنهادی هزینه‌های کمتری نسبت به الگوریتم‌های دیگر دارد. این را می‌توان به توانایی الگوریتم برای متعادل کردن بار نسبت داد که باعث کاهش تعداد ماشین‌های مجازی مورد استفاده شده است این مسئله به طور مستقیم هزینه‌ها را کاهش می‌دهد. شکل ۴ هزینه سرویس را برای هر الگوریتم با تعداد ماشین‌های مجازی متفاوت نشان می‌دهد، که در آن تعداد وظایف ۵۰ عدد ثابت شده بود. وظایف به تعداد کافی VM توزیع شد تا زمان پردازش به حداقل برسد. هزینه بر اساس زمان در ادامه در شکل ۴ نمایش داده شده است:

شکل۴ میانگین هزینه بر اساس زمان در تکرار الگوریتم را نشان می‌دهد که همانطور که از این نتایج مشخص است طی دورهای متفاوت میانگین هزینه کاهش پیدا کرده است و بر اساس استفاده از تابع هزینه بهبودی در این مولفه داشته است. شاید بزرگترین مزیت الگوریتم‌های تکاملی از توانایی رسیدگی به مسائلی باشد که هیچ متخصص انسانی برای آنها وجود ندارد. اگرچه تخصص انسانی باید زمانی که در دسترس است مورد استفاده قرار گیرد، اما اغلب برای خودکارسازی روال‌های حل مسئله کافی نیست. مشکلات چنین سیستم‌های خبره‌ای به خوبی شناخته شده است: کارشناسان ممکن است موافق نباشند، ممکن است خودسازگار نباشند، ممکن است واجد شرایط نباشند، یا ممکن است به سادگی در اشتباه باشند. تحقیقات در زمینه هوش مصنوعی به مجموعه‌ای از روش‌ها و ترفندها برای حل مشکلات خاص در حوزه‌های محدود مورد علاقه تقسیم شده است. مطمئناً، این روش‌ها با موفقیت برای مشکلات خاص اعمال شده است. اما بیشتر این برنامه‌ها به تخصص انسانی نیاز دارند. آنها ممکن است به طرز چشمگیری در مسائل دشواری که به سرعت محاسباتی زیادی نیاز دارند به کار گرفته شوند، اما عموماً درک ما از هوش را ارتقاء می‌دهند. «آنها مشکلات را حل می‌کنند، اما مشکل چگونگی حل مشکلات را حل نمی‌کنند». در مقابل، تکامل روشی را برای حل مسئله چگونگی حل مسائل ارائه می‌دهد. همانطور که در شکل ۵ نشان داده شده است، زمانی که تعداد وظایف تغییر می‌کند، هزینه به تدریج با تعداد وظایف افزایش میابد. وظایف به تعداد کافی VM توزیع شد تا زمان پردازش به حداقل برسد.



شکل ۵. هزینه به ازای تعداد کارها

شکل ۵ هزینه به ازای تعداد کارها را نشان می‌دهد همانطور که ازنتایج این نمودار مشخص است ئر این روند بر اساس تعداد کارهای متفاوت هزینه انجام آنها بهبود یافته است با در نظر گرفتن تابع هزینه و قدرت پردازنده‌ها در این مسئله این پارامتر بهبود داشته است. الگوریتم‌های تکاملی را می‌توان تقریباً برای هر مسئله‌ای که می‌تواند به عنوان یک کار بهینه‌سازی تابع فرموله شود، اعمال کرد. به یک ساختار داده برای نشان دادن راه‌حل‌ها، یک شاخص عملکرد برای ارزیابی راه‌حل‌ها و عملگرهای تغییر برای تولید راه‌حل‌های جدید از راه‌حل‌های قدیمی نیاز دارد (انتخاب نیز مورد نیاز است اما کمتر به ترجیحات انسانی وابسته است). فضای حالت راه‌حل‌های ممکن می‌تواند از هم گسسته باشد و می‌تواند ناحیه‌های غیرممکن را در بر بگیرد و شاخص عملکرد می‌تواند با زمان متغیر باشد یا حتی تابعی از راه‌حل‌های موجود در جمعیت باشد. طراح انسان می‌تواند نمایشی را انتخاب کند که از شهود پیروی کند. از این نظر، این روش مستقل از نمایش است، برخلاف سایر تکنیک‌های عددی که ممکن است فقط برای مقادیر پیوسته یا سایر مجموعه‌های محدود قابل اجرا باشند. بازنمایی باید به عملگرهای تغییری اجازه دهد که پیوند رفتاری بین والدین و فرزندان را حفظ کنند. تغییرات کوچک در ساختار والدین باید منجر به تغییرات کوچک در فرزندان حاصل شود و به همین ترتیب تغییرات بزرگ باید تغییرات فاحشی را ایجاد کند. پیوستاری از تغییرات احتمالی باید به گونه‌ای مجاز باشد که «اندازه گام» مؤثر الگوریتم را بتوان تنظیم کرد، این انعطاف‌پذیری اجازه می‌دهد تا اساساً همان رویه را برای مسائل ترکیبی گسسته، مسائل بهینه‌سازی پارامترهای با ارزش پیوسته، مسائل اعداد صحیح مختلط و غیره اعمال کرد.

۵- بحث

اگرچه تاریخچه محاسبات تکاملی به دهه‌های ۱۹۵۰ و ۱۹۶۰ برمی‌گردد، تنها در دهه گذشته الگوریتم‌های تکاملی برای حل مسائل دنیای واقعی در رایانه‌ها عملی شده‌اند. از آنجایی که رایانه‌ها به ارائه عملکرد سریع ادامه می‌دهند، استفاده از این روش‌ها متداولتر خواهد شد. انعطاف‌پذیری الگوریتم‌های تکاملی برای پرداختن به مسائل بهینه‌سازی عمومی با استفاده از تقریباً هر نمایش معقول و شاخص عملکرد، با عملگرهای تغییر که می‌توانند برای مسئله در دست طراحی شوند و مکانیسم‌های انتخابی که برای سطح مناسبی از سخت‌گیری تنظیم شده‌اند، به این تکنیک‌ها برتری نسبت به کلاسیک می‌دهد. روش‌های بهینه‌سازی عددی علاوه بر این، روش دو مرحله‌ای برای تطبیق خود پارامترهایی که جستجوی تکاملی را کنترل می‌کنند، اپراتور انسانی را از نیاز به ساخت راه‌حل‌های دستی رها می‌کند، که اغلب زمان‌بر یا به سادگی غیرممکن است. الگوریتم‌های تکاملی مجموعه‌ای از رویه‌ها را ارائه می‌دهند که ممکن است به طور مفیدی برای مسائلی که با روش‌های رایج در برابر راه‌حل مقاومت کرده‌اند، اعمال شوند و زمانی که چنین ترکیب‌هایی سودمند به نظر می‌رسند، می‌توانند با چنین تکنیک‌هایی ترکیب شوند. مقایسه تجربی مدل پیشنهادی با MOAPSO-DP نشان داد که مدل پیشنهادی با کاهش ۱۰ درصدی زمان انجام کار و افزایش استفاده از منابع به میزان ۱۶,۳ درصد از آن بهتر عمل کرده است. علاوه بر این، روش پیشنهادی هزینه سرویس، زمان انتظار را در مقایسه با الگوریتم‌های دیگر کاهش داده است.

۶- نتیجه‌گیری

روش‌های سنتی بهینه‌سازی نسبت به تغییرات پویا در محیط مقاوم نیستند و اغلب به یک راه‌اندازی مجدد کامل برای ارائه راه‌حل نیاز دارند (مثلاً برنامه‌نویسی پویا). در مقابل، الگوریتم‌های تکاملی می‌توانند برای تطبیق راه‌حل‌ها با شرایط متغیر استفاده شوند. جمعیت موجود راه‌حل‌های تکامل‌یافته، مبنایی را برای بهبود بیشتر فراهم می‌کند و در بیشتر موارد نیازی نیست و نه مطلوب است که جمعیت را به‌طور تصادفی مجدداً شروع کنیم. در واقع، این روش سازگاری در مواجهه با یک محیط پویا را می‌توان به نفع خود استفاده کرد. توانایی انطباق در رسیدن به راه‌حل با شرایط در حال تغییر برای حل عملی مسئله اهمیت حیاتی دارد. به عنوان مثال، فرض کنید که یک شبیه‌سازی خاص کامل را به یک محیط تولید صنعتی ارائه می‌دهد. تمام ایستگاه‌های کاری و فرآیندها دقیقاً مدل‌سازی می‌شوند و از یک الگوریتم برای یافتن یک زمان‌بندی

«عالی» برای به حداکثر رساندن تولید استفاده می‌شود. با این حال، این برنامه زمانی کامل هرگز در عمل اجرا نخواهد شد، زیرا زمانی که برای بررسی ارائه شود، کارخانه تغییر کرده است: ممکن است ماشین آلات خراب شده باشند، ممکن است پرسنل به کار خود گزارش نداده باشند یا سوابق کافی را در مورد قبلی نگهداری نکرده باشند. کار در حال انجام، سایر تعهدات ممکن است مستلزم تغییر جهت استفاده از تجهیزات و غیره باشد. طرح «کامل» قبل از اینکه اجرا شود منسوخ شده است. به جای صرف تلاش محاسباتی قابل توجه برای یافتن چنین طرح‌هایی بی‌نقص، تجویز بهتر این است که تلاش محاسباتی کمتری برای کشف طرح‌های مناسبی که در برابر ناهنجاری‌های مورد انتظار مقاوم هستند و می‌توانند در زمان وقوع رویدادهای غیرمنتظره در رسیدن تکامل یابند، صرف کنیم. بنابراین در این پژوهش نیز از همین دیدگاه استفاده شده و از یک رویکرد تکاملی استفاده شده است.

سیستم‌های محاسبات لبه موبایل (MEC) خدمات کاربران نهایی را برای دستیابی به QoS و کیفیت QoE ارائه می‌کنند. در این پژوهش به یک مسئله بهینه‌سازی تخصیص منابع هنگام بارگیری وظایف براساس دستگاه‌های تلفن همراه (MD) به سرورهای لبه (ES) در سیستم‌های MEC بررسی شد و در این مقاله سعی شد زمان و هزینه خدمات به حداقل برسد. در این مورد، وظایف را می‌توان به صورت محلی پردازش و به سرور MEC بارگذاری کرد. مسئله اصلی این بود که وظایف تخلیه‌شده برای ماشین‌های مجازی در ES زمانبندی شوند تا زمان محاسبات، هزینه خدمات، اتلاف ES، و حداکثر ارتباط (AE) یک کار با یک ES به حداقل برسد. در این مقاله با استفاده از الگوریتم ترکیبی ازدحام ذرات و گرگ خاکستری یک روش جدید برای مدیریت تخصیص منابع و زمان‌بندی وظایف برای دستیابی به یک نتیجه بهینه در MEC معرفی شد. نتایج این پژوهش با الگوریتم‌ها مقایسه شد و نتایج نشان دهنده برتری رویکرد پیشنهادی است. نتایج نشان می‌دهد که مدل پیشنهادی با کاهش ۱۰ درصدی زمان انجام کار و افزایش استفاده از منابع به میزان ۱۶٫۳ درصد بهتر از دیگر الگوریتم‌ها عمل کرد.

مراجع

[1] Huda, S. A., & Moh, S. (2022). Survey on computation offloading in UAV-Enabled mobile edge computing. Journal of Network and Computer Applications, 103341.

[2] Li, X., Lan, X., Mirzaei, A., & Bonab, M. J. A. (2022). Reliability and robust resource allocation for Cache-enabled HetNets: QoS-aware mobile edge computing. Reliability Engineering & System Safety, 220, 108272.

[3] Sulieman, N. A., Ricciardi Celsi, L., Li, W., Zomaya, A., & Villari, M. (2022). Edge-Oriented Computing: A Survey on Research and Use Cases. Energies, 15(2), 452.

[4] Kumar, D., Baranwal, G., & Vidyarthi, D. P. (2022). A Survey on Auction based Approaches for Resource Allocation and Pricing in Emerging Edge Technologies. Journal of Grid Computing, 20(1), 1-52.

[5] Wang, Z., Lv, T., & Chang, Z. (2022). Computation offloading and resource allocation based on distributed deep learning and software defined mobile edge computing. Computer Networks, 108732.

[6] Qiu, H., Zhu, K., Luong, N. C., Yi, C., Niyato, D., & Kim, D. I. (2022). Applications of auction and mechanism design in edge computing: A survey. IEEE Transactions on Cognitive Communications and Networking.

[7] Qiu, H., Zhu, K., Luong, N. C., Yi, C., Niyato, D., & Kim, D. I. (2022). Applications of auction and mechanism design in edge computing: A survey. *IEEE Transactions on Cognitive Communications and Networking*.

[8] Li, X., Lan, X., Mirzaei, A., & Bonab, M. J. A. (2022). Reliability and robust resource allocation for Cache-enabled HetNets: QoS-aware mobile edge computing. Reliability Engineering & System Safety, 220, 108272.

[9] Elgendy, I. A., Zhang, W., Tian, Y. C., & Li, K. (2019). Resource allocation and computation offloading with data security for mobile edge computing. Future Generation Computer Systems, 100, 531-541.

[10] Abbas, N.; Zhang, Y.; Taherkordi, A.; Skeie, T. Mobile Edge Computing: A Survey. *IEEE Internet Things J.* **2017**, *5*, 1–12.

[11] Wang, S.; Zhang, X.; Zhang, Y.; Wang, L.; Yang, J.; Wang, W. A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications. *IEEE Access* **2017**, *5*, 6757–6779.

[12] Sulieman, N. A., Ricciardi Celsi, L., Li, W., Zomaya, A., & Villari, M. (2022). Edge-Oriented Computing: A Survey on Research and Use Cases. *Energies*, *15*(2), 452.

[13] Li, Y.; Wang, S. An energy-aware edge server placement algorithm in mobile edge computing. In Proceedings of the 2018 IEEE International Conference on Edge Computing (EDGE), San Francisco, CA, USA, 2–7 July 2018; pp. 66–73.

[14] Abbas, N.; Zhang, Y.; Taherkordi, A.; Skeie, T. Mobile edge computing: A survey. *IEEE Internet Things J.* **2017**, *5*, 450–465. [CrossRef]

[15] Maia, A.M.; Ghamri-Doudane, Y.; Vieira, D.; de Castro, M.F. Optimized placement of scalable iot services in edge computing. In Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Washington, DC, USA, 8–12 April 2019; pp. 189–197.

[16] Xiao, K.; Gao, Z.; Wang, Q.; Yang, Y. A heuristic algorithm based on resource requirements forecasting for server placement in edge computing. In Proceedings of the 2018 IEEE/ACM Symposium on Edge Computing (SEC), Bellevue, WA, USA, 25–27 October 2018; pp. 354–355.

[17] Alrowaily, M.; Lu, Z. Secure edge computing in iot systems: Review and case studies. In Proceedings of the 2018 IEEE/ACM Symposium on Edge Computing (SEC), Bellevue, WA, USA, 25–27 October 2018; pp. 440–444.

[18] Li, X.; Ding, R.; Liu, X.; Yan, W.; Xu, J.; Gao, H.; Zheng, X. Comec: Computation offloading for video-based heart rate detection app in mobile edge computing. In Proceedings of the 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), Melbourne, Australia, 11–13 December 2018 ; pp. 1038–1039.

[19] Xing, H.; Liu, L.; Xu, J.; Nallanathan, A. Joint Task Assignment and Resource Allocation for D2D-Enabled Mobile-Edge Computing. *IEEE Trans. Commun.* **2019**, *67*, 4193–4207.

[20] Nowak, D.; Mahn, T.; Al-Shatri, H.; Schwartz, A.; Klein, A. A Generalized Nash Game for Mobile Edge Computation Offloading. In Proceedings of the 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (Mobile Cloud), Bamberg, Germany, 26–29 March 2018.

[21] Zhang, D., Piao, M., Zhang, T., Chen, C., & Zhu, H. (2020). New algorithm of multi-strategy channel allocation for edge computing. AEU-International Journal of Electronics and Communications, 126, 153372.

[22] Alfakih, T., Hassan, M. M., & Al-Razgan, M. (2021). Multi-objective accelerated particle swarm optimization with dynamic programing technique for resource allocation in mobile edge computing. IEEE Access, 9, 167503-167520..

[23] Ma, S., Song, S., Zhao, J., Zhai, L., & Yang, F. (2020). Joint network selection and service placement based on particle swarm optimization for multi-access edge computing. IEEE Access, 8, 160871-160881.