JIPET

Journal of Intelligent Procedures in Electrical Technology
Vol. 16/ No. 61/ Spring 2025
P-ISSN: 2322-3871, E-ISSN: 2345-5594, http://jipet.iaun.ac.ir/

https://dorl.net/dor/20.1001.....
Research Article

Increase the Efficiency of the Offloading Algorithm in Fog Computing by Particle Swarm Optimization Algorithm

Seyed Ebrahim Dashti[1], Hossin Zare[2], *M.Sc.*

[1]Department of Computer Engineering- Jahrom Branch, Islamic Azad University, Jahrom, Iran
[2]Department of Computer Engineering- Zahedshar, Branch, Islamic Azad University, Zahedshar, Iran
seyedebrahim.dashti@iau.ac.ir, hossin4604@gmail.com

Abstract
Edge computing is a computing paradigm that extends cloud services to devices at the edge. This processing model refers to technologies that allow computing and storage to be performed on devices at the edge of the network. In this architecture, computing and storage operations take place close to objects and data sources. In order to reduce latency and network traffic between end devices and cloud centers, groups at the edge have processing capabilities, perform a large number of processing and computing tasks, including data processing, temporary storage, device management, decision making, and privacy protection. Since the number of edge devices is large, there must be a mechanism to select these tasks and offload them to the cloud. The problem to be decided is that which one of the available edge devices should be selected for unloading and then unloaded. This problem is classified as one of the hard non-polynomial problems and by using deterministic algorithms simply and in polynomial time, it is not possible to find a suitable and efficient solution for it found.

key words: cloud processing, fog computing, offloading, evolutionary algorithm, particle swarm optimization algorithm

Received: 31 July 2022
Revised: 14 September 2022
Accepted: 22 November 2022

Corresponding Author: Dr. Seyed-Ebrahim Dashti

Citation: S.E. Dashti, H. Zare, "Increase the efficiency of the offloading algorithm in fog computing by particle swarm optimization algorithm ", Journal of Intelligent Procedures in Electrical Technology, vol. 16, no. 61, pp. 79-96, June 2025 (in Persian).

https://dorl.net/dor/20.1001.....
مقاله پژوهشی

افزایش کارایی الگوریتم تخلیه در محاسبات مه با کمک الگوریتم بهینه‌سازی ازدحام ذرات

سید ابراهیم دشتی[۱]، حسین زارع[۲]

۱- دانشکده برق و کامپیوتر- واحد جهرم، دانشگاه آزاد اسلامی، جهرم، ایران
۲- دانشکده برق و کامپیوتر- واحد زاهدشهر، دانشگاه آزاد اسلامی، زاهدشهر، ایران
seyedebrahim.dashti@iau.ac.ir, hossin4604@gmail.com

چکیده: با توجه به کاربرد روزافزون محاسبات مه لزوم ارائه راه حل‌هایی جهت افزایش کارآیی آنها به شدت احساس می‌شود. بر اساس این مسئله که تعداد دستگاه‌های لبه‌ای زیاد است، باید ساز و کاری برای انتخاب این وظایف و تخلیه آن‌ها به ابر وجود داشته باشد. مسئله مورد نظر برای تصمیم‌گیری این است که از بین آن دستگاه لبه‌های موجود برای تخلیه کدام یک از آن‌ها انتخاب و سپس تخلیه گردد، که این مسئله در زمره مسائل غیر چندجمله‌ای سخت قرار گرفته و با استفاده از الگوریتم‌های قطعی به سادگی و در زمان چندجمله‌ای نمی‌توان راه‌حلی مناسب و کارآمد برای آن یافت نمود. در این مقاله برای حل این مساله از الگوریتم بهینه‌سازی ازدحام ذرات (PSO) استفاده شده است. روش پیشنهادی با تعریف تابع هدف مناسب برحسب تخلیه مناسب بار و توزیع عادلانه وظایف بر روی منابع محاسباتی کارآیی را نسبت به روش‌های مشابه بهبود داده است. در مقایسه با روش‌های دیگر، مانند روش بدون بارگیری، بارگیری کامل به روش ابر و بارگذاری کامل به روش ابر، آزمایش‌ها و شبیه‌سازی‌های گسترده نشان داده‌اند که روش پیشنهادی مؤثر است و می‌تواند استراتژی بارگذاری بهینه را برای کاربران سیار فراهم کند. برای ارزیابی این روش از داده‌های واقعی پلنت‌لب استفاده شده و نتایج بیانگر این است که روش پیشنهادی مصرف انرژی را بین ۳ تا ۱۰ درصد و زمان اجرای کل نیز بین ۵ تا ۸ درصد در مقایسه با روش‌های دیگر کاهش یافته است.

کلمات کلیدی: الگوریتم بهینه‌سازی ازدحام ذرات، الگوریتم تکاملی، پردازش ابری، محاسبات مه، مسئله تخلیه بار

تاریخ ارسال مقاله: ۱۴۰۱/۵/۹
تاریخ بازنگری مقاله: ۱۴۰۱/۶/۲۳
تاریخ پذیرش مقاله: ۱۴۰۱/۹/۱

نام نویسنده‌ی مسئول: دکتر سیدابراهیم دشتی
نشانی نویسنده‌ی مسئول: شیراز- دانشگاه آزاد اسلامی واحد جهرم – دانشکده مهندسی برق و کامپیوتر - گروه کامپیوتر

۱- مقدمه

محاسبات سبز و صرفه‌جویی در مصرف انرژی و محدودیت منابع و افزایش قابل‌توجه کاربران موبایل و علایق تجاری قوی در زمینه رایانش و سرویس‌های ابری فرصت‌ها و نیازهایی در خصوص رایانش ابری مبتنی بر لبه ایجاد کرده است. رایانش ابری[۱] به دلیل قابلیت‌های بالای پردازشی و ذخیره‌سازی که دارد یک الگوی پردازشی متمرکز برای پردازش داده‌ها است، می‌توان بیان کرد که تقریباً تمامی وظایف پردازشی باید اصطلاحاً در یک ابر انجام شود [۱]. به عبارت دیگر تمامی تقاضاها باید به یک ابر متمرکز ارسال شود که همین مسئله خود یکی از مشکلات الگوی پردازش ابری است، چرا که سرعت توسعه منابع پردازشی با به معنای باند شبکه متناسب نبوده و همین مسئله به پهنای باند خود یک گلوگاه اصلی برای این الگوی پردازشی محسوب شده و باعث تأخیر در ارتباطات می‌شود [۲].

در برخی از کاربردهای رایانش ابری از جمله سیستم‌های کنترل ترافیک، سیستم‌های مرتبط با ابر سلامت و شبکه‌های هوشمند و سیستم‌های حساس به تأخیر، دیگر نیاز به تأخیر خیلی کم و همچنین تحقق‌پذیری وجود دارد [۳]. به همین خاطر تأخیری که در اثر تبادل داده‌ها با یک ابر متمرکز به وجود می‌آید، برای سیستم قابل قبول نخواهد بود. با توجه به‌تمامی این موانع در راه به‌کارگیری رایانش ابری نیاز به یک الگوی پردازشی جدید که مشکلات مورد نظر را تا حدودی برطرف کند، کاملاً حس می‌شود. از این‌رو الگوی پردازش جدیدی به نام محاسبات لبه[۲]، معرفی شده است که یک الگوی پردازشی است که سرویس‌های ابری را به دستگاه‌های موجود در لبه بسط می‌دهد. این الگوی پردازشی به فناوری‌هایی برمی‌گردد که اجازه می دهد تا محاسبات و ذخیره‌سازی روی دستگاه‌های موجود در لبه شبکه انجام شود. در این معماری، عملیات محاسباتی و ذخیره‌سازی نزدیک به اشیا و منابع داده‌ای صورت می‌گیرد [۴]. به‌منظور کاهش تأخیر و ترافیک شبکه بین دستگاه‌های انتهایی و مراکز ابر، گروه‌های موجود در لبه قابلیت پردازش دارند، تعداد زیادی از وظایف پردازشی و محاسباتی را، از جمله پردازش داده‌های ذخیره‌سازی موقت مدیریت دستگاه‌ها تصمیم‌گیری و حفاظت از حریم خصوصی انجام می‌دهند. علاوه بر آن دستگاه‌های لبه از طریق هسته شبکه به مراکز داده‌های موجود در ابر متصل می‌شوند [۵] با استفاده از تخلیه بار محاسباتی، می‌توان مشکل محدودیت منابع دستگاه‌های موجود در لبه را تا حد زیادی برطرف کرد. نوآوری اصلی به کار رفته در این مقاله را می‌توان به شرح زیر خلاصه کرد:

الف- بارگذاری محاسباتی برای برنامه کاربردی جریان کاری در رایانش لبه به‌خوبی بررسی شده است. هم مصرف انرژی و هم مصرف زمان و همچنین هزینه کاربر سیار با استفاده از ابر، از اهداف بهینه‌سازی محسوب می‌شوند. همچنین بر اساس تجزیه و تحلیل نظری، یک مدل بهینه‌سازی چند هدفی ایجاد می‌شود.

ب- روشی به نام روش بارگذاری محاسباتی چندهدفه برای برنامه‌های کاربردی جریان کاری بر اساس الگوریتم بهینه‌سازی ازدحام ذرات برای دستیابی به راه‌حل بهینه پیشنهاد شده است. برخی از پارامترهای موجود در مرحله الگوریتم با توجه به نیازهای این مسئله بهبود یافته‌اند.

ج- در مقایسه با روش‌های دیگر، مانند روش بدون بارگیری، بارگیری کامل به روش ابر و بارگذاری کامل به روش ابر، آزمایش‌ها و شبیه‌سازی‌های گسترده نشان داده‌اند که روش پیشنهادی مؤثر است و می‌تواند استراتژی بارگذاری بهینه را برای کاربران سیار فراهم کند.

سازماندهی پژوهش به این صورت است که در بخش دوم ادبیات و پیشینه تحقیق مطرح شده است و در بخش سوم روش پیشنهادی و در بخش چهارم نتایج روند و ارزیابی آن صورت گرفته است و در نهایت در بخش آخر نتیجه‌گیری بیان شده است.

۲- ادبیات و پیشینه تحقیق

امروزه رشد چشمگیر فناوری‌هایی نظیر اینترنت اشیا[۳] و اینترنت همراه باعث شده است تا اشیا و افراد زیادی بتوانند در هر زمان و هر مکان به اینترنت متصل شوند. به‌طوری‌که پیش‌بینی می‌شود تا سال ۲۰۳۰ میلادی تعداد دستگاه‌های متصل به اینترنت بیش از ۵۰ میلیارد دستگاه شود که قطعا منجر به تولید حجم انبوهی از داده با تنوع زیاد خواهد شد به‌طوری‌که تخمین زده می‌شود حجم این داده‌ها از مرتبه‌ی زتابایت[۴] و در حدود ۵۰۰ زتابایت شود [۶]. با توجه به این حجم زیاد از

داده‌ها، تجهیزات پردازشی و ذخیره‌سازی فعلی نمی‌تواند پاسخگوی بسیاری از این تقاضاها باشد و مدیریت آن‌ها با فناوری‌های کنونی از جمله سیستم‌های توزیع شده و یا رایانش ابری دشوار خواهد بود. رایانش ابری به دلیل قابلیت‌های بالای پردازشی و ذخیره‌سازی که دارد گزینه مناسبی برای پردازش داده‌ها است. اما از آنجایی‌که این الگوی پردازشی یک الگوی پردازشی متمرکز است، می‌توان گفت که تقریباً تمامی وظایف پردازشی باید اصطلاحاً در یک ابر انجام شود که همین مسئله خود یکی از مشکلات الگوی پردازشی ابری است. چرا که سرعت توسعه منابع پردازشی با پهنای باند شبکه متناسب نبوده است و همین مسئله پهنای باند خود یک گلوگاه اصلی برای این الگوی پردازشی محسوب شده و باعث تأخیر در ارتباطات می‌شود [۷].

۲-۱- رایانش لبه

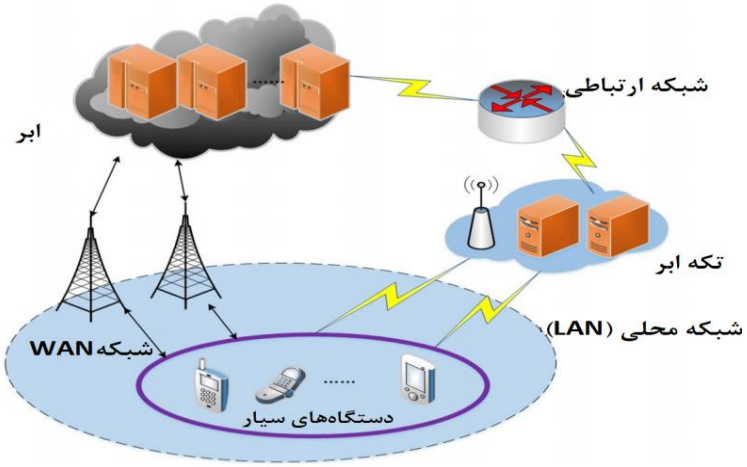
پردازش مبتنی بر لبه یک الگوی پردازشی جدید است که سعی بر این دارد تا قابلیت‌های رایانش ابری را در لبه‌های شبکه را توسعه دهد. این فناوری پردازش، ارتباطات، فرامین کنترلی و قابلیت‌های ذخیره‌سازی را در لبه شبکه فراهم می‌کند. پردازش مبتنی بر لبه تفاوت‌هایی با پلتفرم‌های غیرمتمرکز[۵] از لحاظ معماری دارد. مدل مرجع معماری پردازش مبتنی بر لبه یکی از موضوعات پژوهشی مهم است. طی سال‌های اخیر معماری‌های زیادی برای این الگوی پردازشی پیشنهاد شده است. این مدل‌ها اکثراً از ساختار پایه سه لایه‌ای مشتق شده‌اند. این الگوی پردازشی قابلیت‌های رایانش ابری را در لبه‌های شبکه گسترش می‌دهد و به این ترتیب یک لایه مه[۶] را بین دستگاه‌های انتهایی و مراکز ابر اضافه می‌کند [۸].

۲-۲- تخلیه بار در محاسبات لبه

هدف اصلی از ساز و کار مدیریت تأخیر[۷] در پردازش مبتنی بر مه، محدود کردن زمان پاسخ سرویس ها در یک حد قابل قبول است. این آستانه عبارت است از حداکثر تأخیر قابل تحمل برای یک سرویس یا حداقل نیازمندی یک کاربرد برای کیفیت سرویس. تأخیر کلی محاسبات و ارسال و دریافت دادهی تمامی درخواست ها می‌تواند به وسیله توزیع وظایف پردازشی و تعدیل کارها روی گره‌های مختلف موجود در لبه به حداقل رسد [۹]. با استفاده از ساز و کارهای تخلیه بار محاسباتی می‌توان بر مشکل محدودیت منابع دستگاه‌های موجود در لبه غلبه کرد، بخصوص برای وظایفی که از لحاظ محاسباتی سنگین هستند. این مکانیزم به بهبود کارایی و کاهش مصرف باتری نیز کمک می‌کند. در این مقاله یک مدل توزیع شده برای تخلیه بار محاسباتی ارائه شده است. مدل مذکور بر اساس رویکرد مبتنی بر تکامل الگوریتم بهینه‌سازی ازدحام ذرات پیشنهاد شده است، به‌طوری‌که مسئله تصمیم‌گیری تخلیه بار به عنوان یک روند تکامل مدل شده است. زمانی که چند دستگاه، همزمان با هم سعی می کنند تا وظایف پردازشی خود را در ابر تخلیه کنند و همگی در حال استفاده از یک کانال بی‌سیم مشترک هستند، تنها یک وظیفه برای ارسال به ابر انتخاب می‌شود و آن وظیفه‌ای است که زمان مورد نیاز برای انجام آن و همچنین منابع قابل مصرف آن بعد از تخلیه افزایش می‌یابد [۹].

۲-۳- صرفه‌جویی در انرژی محاسبات ابری از طریق تخلیه بار

محققان در مرجع [۱۰] محاسبات سبز را به‌عنوان مطالعه و شیوه طراحی، تولید، استفاده و مرتب‌کردن موثر و کارایی کامپیوترها، سرورها و زیرسیستم‌های مربوطه مانند مانیتورها، چاپگرها، دستگاه‌های ذخیره‌سازی و سیستم‌های ارتباطی و شبکه‌سازی با حداقل تاثیر بر محیط تعریف می‌کند. یکی از اهداف اولیه، کاهش مصرف انرژی و افزایش کارآیی انرژی در دستگاه‌های موبایل، برنامه‌های کاربردی و همچنین اجرای آن‌ها است [۱۱]. از لحاظ شهودی، مزایای اصلی آن عبارتند از: کاهش هزینه‌ها، حفظ انرژی و حفاظت از محیط زیست. در شکل (۱) معماری محاسبات لبه‌ای مشخص شده است. در این معماری به‌منظور کاهش تأخیر و ترافیک شبکه بین دستگاه‌های انتهایی و مراکز ابر، گره‌های موجود در لبه که قابلیت‌های پردازشی دارند، تعداد زیادی از وظایف پردازشی و محاسباتی را، از جمله پردازش داده‌ها، ذخیره‌سازی موقت، مدیریت دستگاه‌ها، تصمیم‌گیری و حفاظت از حریم خصوصی، انجام می‌دهند. در محاسبات لبه‌ای، دستگاه‌های موجود در لبه می‌توانند از طیف تقریباً گسترده‌ای از دستگاه‌ها باشند.



شکل (۱): معماری رایانش لبه
Figure (1): Edge computing architecture

به‌طور مثال حسگرهای هوشمند، تلفن‌های همراه هوشمند، وسایل نقلیه هوشمند و حتی یک سرویس‌دهنده لبه‌ای باشد. این دستگاه‌ها می‌توانند در یک شبکه محلی با یکدیگر ارتباط برقرار کرده و یک شبکه لبه‌ای را تشکیل دهند. علاوه بر آن دستگاه‌های لبه از طریق هسته شبکه به مراکز داده موجود در ابر متصل می‌شوند. ایده اصلی، آفلودکردن وظایف نیازمند محاسبات فشرده از دستگاه‌های لبه به سرورهای ابری در ابر موبایل جهت حفظ انرژی روی دستگاه‌های لبه و بهبود تجربه کاربر است [۱۲]. این شیوه با دو چالش روبرو است که عبارتند از: الف- شرایط بهینه هنگام آفلود محاسبات از کاربران لبه به ابر چیست؟ و ب- عواملی که باید هنگام آفلود محاسبات به ابر مد نظر قرار گیرند، کدامند؟ روش ارائه شده در مرجع [۱۲] این مسائل را بر اساس آنالیز مصرف انرژی برای وظیفه محاسبه روی دستگاه‌های موبایل و سرور در ابر خطاب قرار می‌دهد. از رابطه (۱) برای برقراری ارتباط بین آفلود محاسبه از دستگاه‌های لبه با ابر استفاده کردند. با این رابطه مقدار انرژی ذخیره شده در طول آفلود را تعیین می‌شود.

$$E_S = P_C \times \frac{C}{M} - P_i \times \frac{C}{S} - P_{tr} \times \frac{D}{B} \qquad (۱)$$

که در آن C تعداد دستورالعمل‌های محاسباتی آفلود شده، M سرعت دستگاه موبایل (دستورالعمل/ ثانیه)، S سرعت سرور ابری (دستورالعمل/ ثانیه)، P_C توان مصرفی دستگاه لبه (وات)، P_i توان مصرفی بی‌کار دستگاه موبایل (وات)، P_{tr} توان مصرفی انتقال دستگاه لبه (وات)، D بایت داده مبادله شده و B پهنای باند شبکه را نشان می‌دهد [۱۲]. در صورت ثابت بودن P_{tr}، P_i و P_C اگر رابطه (۱)، عدد مثبتی حاصل کند، آنگاه عملیات آفلود، مصرف انرژی را کاهش می‌دهد، در غیر این‌صورت این اتفاق نمی‌افتد. همچنین عملیات آفلود در صورتی می‌تواند در انرژی بیشتری صرفه‌جویی کند که داده‌های مورد نیاز روی ابر ذخیره شده باشند، زیرا روی شبکه بی‌سیم، نیازی به انتقال داده‌ها از دستگاه‌های موبایل به ابر نیست. با این‌حال، اگر نیاز به انتقال داده‌ها روی شبکه‌های بی‌سیم با نیازهای امنیتی باشد، به عبارتی پردازش رمزگذاری، آنگاه مصرف انرژی کل، زمانی که سرور F بار سریعتر عمل می‌کند را می‌توان با رابطه (۲) به‌دست آورد:

$$E_C = \frac{C}{M} \times (P_C - \frac{P_i}{F}) - P_{tr} \times \frac{D}{B} - P_C \times \frac{C_p}{M} \qquad (۲)$$

دراین رابطه C_P تعداد دستورالعمل‌های اضافی مورد نیاز برای محاسبه اضافه شده جهت امنیت (به عبارتی رمزگذاری) است. از مصرف انرژی اضافی زمانی که C_P صفر می‌شود، در صورتی می‌توان اجتناب نمود که عملیات رمزگذاری روی دستگاه موبایل انجام‌شده باشد [۱۲]. بدیهی است، آفلود محاسبه منجر به مسائل دیگری می‌گردد. زمانی که عملیات آفلود شامل انتقال داده‌ها روی شبکه‌ها (شبکه‌های بی‌سیم یا اینترنت بی‌سیم) برای سرور ابری می‌شود، آنگاه با ارتباطات بی‌سیم و سروها چگونه می‌توان مصرف توان را به حداقل رساند؟ عملیات آفلود نیازمند اسکن رابط‌های بی‌سیم موجود توسط دستگاه‌های لبه است زمانی که

انتقال داده‌ها لازم و ضروری است. این مسئله حاکی از آن است که آفلودهای بسیار زیاد نیز مصرف توان دستگاه‌های موبایل برای کشف شبکه را افزایش می‌دهد. بنابراین، مدیریت تعداد آفلودها به یک نگرانی تبدیل می‌شود.

۴-۲- شاخص‌های تخلیه بار بهینه در رایانش لبه

این بخش به بررسی مناسبت شاخص‌های مختلف مهاجرت در رایانش لبه می‌پردازد. بررسی این شاخص‌ها در زمینه اجرای برنامه بهینه، یک بعد تحقیقاتی مهم است. این شاخص‌ها مربوط به ۵ زمینه شامل دستگاه لبه، شبکه، نوع برنامه، ترجیح کاربر و هزینه است [۱۳]. یک چالش بزرگ برای دستگاه‌های تلفن همراه با منابع محدود، اجرای برنامه‌های مختلف همراه با پیچیدگی و مصرف انرژی بالا است که در این راستا، محاسبه لبه به‌عنوان یک الگوی محاسباتی جدید می‌تواند منابع محاسباتی فراوانی را برای انجام همه یا بخش‌هایی از وظایف دستگاه‌های سیار فراهم کند و از این طریق می‌تواند انرژی دستگاه سیار را به‌شدت کاهش داده و کیفیت سرویس برنامه‌ها را بهبود ببخشد. با این‌حال، بارگذاری وظایف گردش کار به سرورهای پردازشی ابری در معرض تهدیدات امنیتی خارجی (به عنوان مثال تعویق افتادن، تغییر کردن) است. در روش ارائه شده در مرجع [۱۴]، یک استراتژی تخلیه بارگذاری محاسبات امنیتی و انرژی کارآمد[۹] برای گردش کار در محیط محاسبات لبه پیشنهاد شده که هدف از آن بهینه‌سازی مصرف انرژی تحت احتمال خطر و محدودیت‌های مهلت است. اولاً، برای اندازه‌گیری زمان اجرای خدمات امنیتی، یک مدل سربار امنیتی ایجاد نموده و سپس با اضافه‌کردن امنیت، مصرف انرژی و زمان اجرای برنامه گردش کار، مشکل بارگذاری محاسبه را شکل می‌دهد. سرانجام، بر اساس الگوریتم بهینه‌سازی ازدحام ذرات، استراتژی‌های کدگذاری مربوط به انرژی کارآمد با در نظر گرفتن دستور اجرای وظایف و انتخاب مکان و خدمات امنیتی تدوین می‌شود. آزمایش‌های گسترده با انواع پارامترهای گردش کار نشان می‌دهد که استراتژی انرژی کارآمد می‌تواند امنیت و بهره‌وری انرژی را برای برنامه‌های تلفن همراه به‌دست آورد. با الهام از این ایده که کاربر سیار باید مبلغی را که از آن‌ها در ابر استفاده شده بپردازد، در روش ارائه شده در مرجع [۱۴]، مصرف انرژی، مصرف زمان و بهینه‌سازی قیمت را برای برنامه‌های کاربردی جریان کاری در رایانش لبه در حال اتمام در نظر می‌گیرد. زمان برنامه کاربردی جریان کاری به‌عنوان شرط محدودیت در نظر گرفته می‌شود. در روش ارائه شده در مرجع [۱۴] با تمرکز بر روی موضوع فرآیند پیچیده و زمان پاسخگویی طولانی تخلیه وظایف در چند ابر، یک الگوریتم بهینه‌سازی ازدحام ذرات با انطباق وزنی[۹] بر اساس همکاری چند ابر ارائه شده است. در مرحله اول، فرایند اجرای کار از راه دور ابر تلفن همراه ترمینال تلفن همراه مدل شده است. ثانیاً، با توجه به رقابت منابع محاسباتی توسط چندین کاربر، مدل بارگذاری کار بر اساس همکاری چند تکه ابر [۱۰] ساخته می‌شود. از آنجاکه پیچیدگی حل طرح بارگذاری مطلوب بیش از حد زیاد است، انطباق وزنی برای حل مسئله تخلیه بار پیشنهاد می‌شود. نتایج شبیه‌سازی نشان می‌دهد که در مقایسه با الگوریتم بهینه‌سازی ذرات استاندارد و الگوریتم بهینه‌سازی ازدحام ذرات با کاهش وزن اینرسی بر اساس عملکرد گاوسی، الگوریتم انطباق وزنی می‌تواند وزن اینرسی را مطابق با تعداد تکرار تکاملی و فرد تنظیم کند. زمان یافتن یک طرح بارگیری مطلوب کوتاه‌ترین زمان است. این آزمایش همچنین نشان می‌دهد که طرح بارگذاری وظیفه مشترک بر اساس الگوریتم انطباق وزنی می‌تواند زمان بارگیری کل را حداقل ۲۰ درصد در مقایسه با برنامه غیر همکارانه در سیستم چند ابر کاهش دهد.

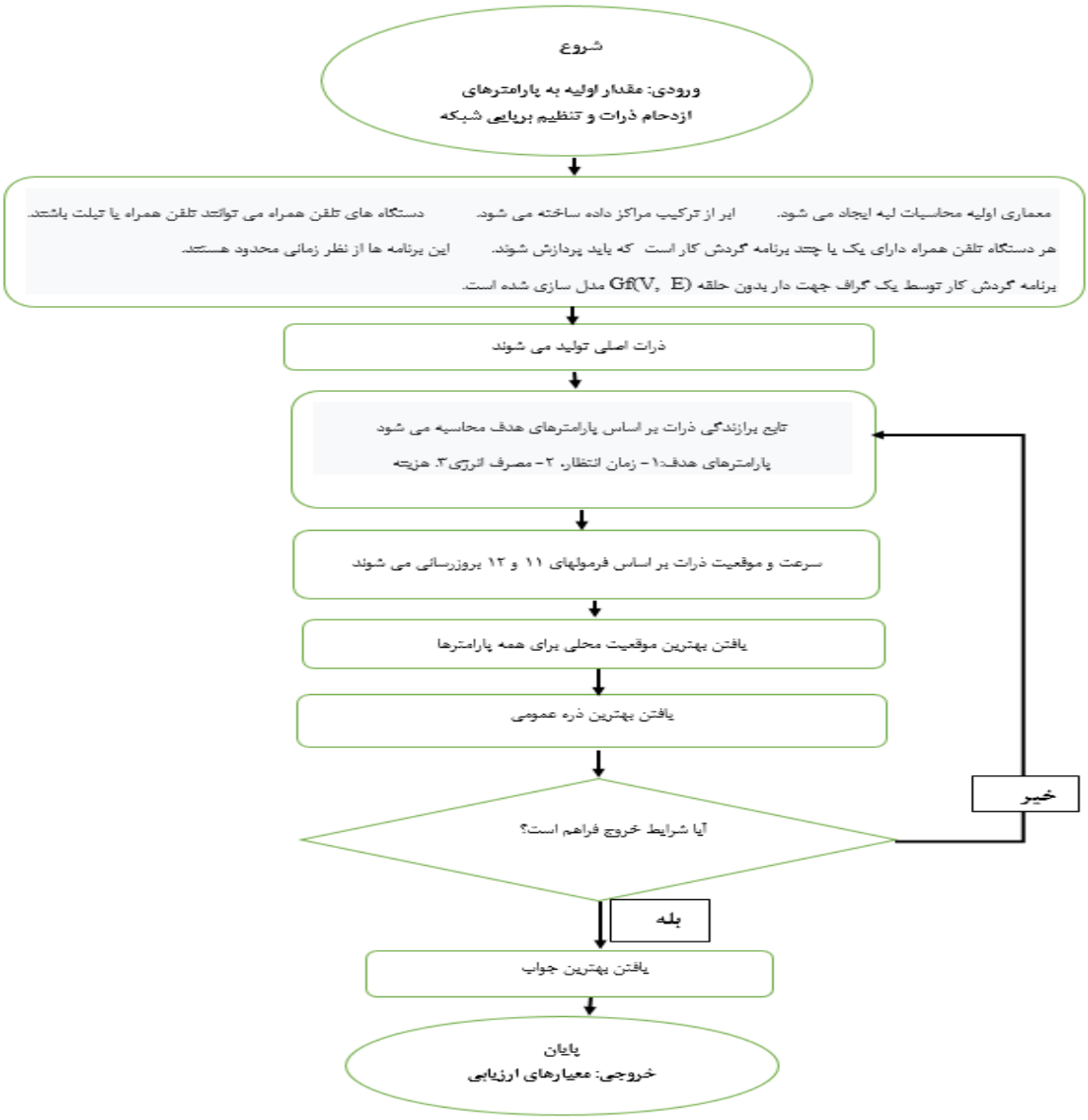
در این مقاله بررسی مشکل بارگیری کار محاسباتی در چند ابر اشاره و مدل بارگیری کار محاسباتی با استفاده از همکاری چندپارچه ای ایجاد شده و و انطباق وزنی به دلیل پیچیدگی در حل مسئله پیشنهاد می‌شود که زمان کل مسئله بارگیری کار را به شیوه‌ای مؤثر کوتاه می‌کند. نتایج آزمایش‌های شبیه‌سازی نشان می‌دهد که طرح بارگذاری کار مشترک بر اساس الگوریتم انطباق وزنی قادر به محدودیت در محاسبات منابع محصوره تنها است و اطمینان حاصل می‌کند که کارهای محاسباتی می‌توانند منابع ابری کافی را به دست آورند. کل زمان اتمام بارگذاری وظایف بر اساس الگوریتم انطباق وزنی در سناریوهای مختلف نسبت به سایر طرح‌های بارگیری کم است.

۳- روش پیشنهادی

با توسعه شبکه رایانه‌ای، رایانش ابری و همچنین شبکه حسگر بی‌سیم، دستگاه‌های تلفن همراه به بخشی ضروری از زندگی روزمره مردم تبدیل شده‌اند. همچنین توسعه سیستم‌های سایبر فیزیکی-اجتماعی و داده‌های بزرگ بیشتر بر زندگی افراد تأثیر گذاشته است. با این‌حال، در مقایسه با یک وسیله سنتی مانند کامپیوتر شخصی، دستگاه‌های سیار دارای محدودیت‌های خاصی در توان محاسباتی، ظرفیت ذخیره‌سازی، به‌ویژه در ظرفیت باتری است. رایانش ابری موبایل خدمات و امکانات جدیدی را برای کاربران تلفن همراه به ارمغان می‌آورد تا از رایانش ابری کاملاً استفاده کنند. با این‌حال، ابر از راه دور معمولاً بسیار دور از کاربران قرار دارد، که ممکن است منجر به تأخیر زیاد شبکه در روند انتقال داده شود. این امر ناچار کیفیت خدمات کاربر را کاهش می‌دهد، به‌ویژه برای برخی از برنامه‌ها، از جمله برنامه‌های جریان کار که به‌طور کلی دارای مهلت‌های سخت اجرا هستند، اگر تأخیر انتقال خیلی زیاد باشد، ممکن است کار به پایان نرسد. برای حل مسئله تأخیر شبکه، پارادایم جدیدی به نام رایانش لبه‌ای ارائه شده است. رایانش لبه‌ای سیار به یک فناوری مهم برای تحقق اینترنت اشیا و 5G تبدیل شده است. رایانش لبه سیار را می‌توان نمونه خاصی از رایانش لبه‌ای در نظر گرفت. ابر نوعی سرور مرزی است که خدمات مختلفی به کاربران در مجاورت دستگاه‌های سیار ارائه می‌دهد. این بدان معناست که می‌تواند با بارگذاری برنامه‌های کاربردی جریان کاری به ابر، تأخیر و مصرف انرژی را کاهش دهد. فرض بر این است که منابع ابر در رایانش لبه نامحدود است. اگر چندین کاربر سیار درخواست هم‌زمان از خدمات ابر را داشته باشند، یک تأخیر در صف رخ می‌دهد. هنگامی‌که یک کاربر سیار خدماتی را درخواست می‌کند که از توانایی ابر فراتر رود، سرویس ابر را نمی‌توان به دست آورد. برای اطمینان از اجرای موفقیت‌آمیز برنامه کاربردی جریان کاری، برای اجرای کار به‌صورت محلی یا بارگیری آن‌ها به ابر باید ملاحظات بیشتری در نظر گرفته شود. علاوه بر این، اجرای برنامه کاربردی جریان کاری توسط ابر باید محدودیت مهلت کار را برآورده کند، که این امر سختی محاسبه بارگیری را بیشتر می‌کند. اگرچه مسئله بارگذاری محاسبات در محاسبات لبه به‌خوبی مورد بررسی قرار گرفته است، اما دلیل اصلی مطالعه رایانش لبه این است که رایانش لبه و محاسبات لبه معماری‌های کاملاً متفاوتی دارند. برای برنامه کاربردی جریان کاری، توجه به محدودیت‌های زمانی بسیار مهم است. با الهام از این ایده که کاربر سیار باید مبلغی را که از آن‌ها در ابر استفاده شده است بپردازد، در این مطالعه، ما مصرف انرژی، مصرف زمان و بهینه‌سازی قیمت را برای برنامه‌های کاربردی جریان کاری در رایانش لبه در حال اتمام در نظر می‌گیریم. زمان برنامه کاربردی جریان کاری به‌عنوان شرط محدودیت در نظر گرفته می‌شود. شبه کد روند پیشنهادی در شکل (۲) نشان داده شده است. همان‌طور که در این شبه کد مشخص است در ابتدا ذرات اولیه ایجاد خواهد شد و در ادامه پردازش‌ها بر اساس تابع هدف که در ادامه توضیح داده خواهد شد صورت می‌گیرد.

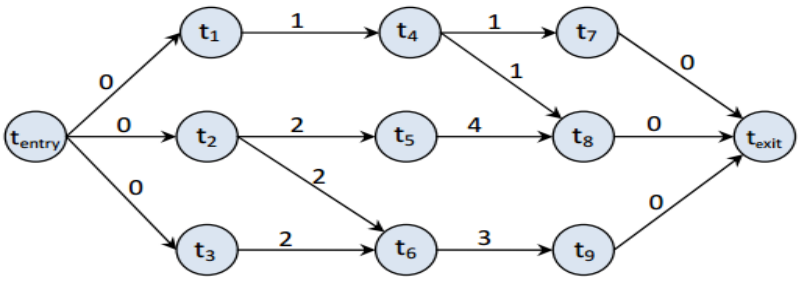
۱-۳- مدل سیستم و فرموله‌سازی مسئله

در اینجا مدل سیستم و فرمول‌بندی مسئله ارائه شده است. در ابتدا معماری پایه رایانش لبه شرح داده شده و سپس حالت پایه معرفی می‌شود. علاوه بر این، حالت مصرف زمان، حالت مصرف انرژی و حالت هزینه توصیف شده است. ابر ترکیبی از مراکز داده است. دستگاه‌های سیار می‌تواند تلفن همراه یا تبلت باشد. هر دستگاه‌های سیار دارای یک یا چند برنامه کاربردی جریان کاری است که باید پردازش شوند. به‌طور کلی، این برنامه‌ها محدود به زمان هستند. این برنامه‌ها می‌توانند به‌صورت مستقیم به‌صورت محلی اجرا شوند و کاربران می‌توانند بخشی از برنامه یا کل برنامه را از طریق شبکه محلی یا ابر از طریق شبکه گسترده با توجه به نیاز خود به ابر منتقل کنند تا زمان یا انرژی مصرفی کاربر یا هر دو کاهش یابد. سازمان‌دهی کامل رایانش لبه در شکل (۱) نشان داده شده است. همان‌طور که در شکل (۳) نشان داده شده است، برنامه کاربردی جریان کاری توسط یک گراف جهت‌دار بدون حلقه Gf (V,E)، مدل‌سازی می‌شود، که f نمایانگر fامین برنامه کاربردی جریان کاری ($1\leq f \leq F$) است و F تعداد کل برنامه‌های کاربردی جریان کاری را نشان می‌دهد. هر برنامه شامل چندین وظیفه است و هر گره در شکل (۳) را می‌توان به‌عنوان یک وظیفه مشاهده کرد. $V=\{v1,f , v2,f , . . . , vN,f \}$ مجموعه‌ای از وظایف را نشان می‌دهد و یال در E مجموعه‌ای از وابستگی بین هر دو وظیفه را نشان می‌دهد. هر یال با یک وزن، dij نشان‌دهنده اندازه انتقال داده از وظیفه vi,f به vj,f، f است. ابر به‌عنوان ماشین چندگانه مجازی برای پردازش هم‌زمان برنامه‌های کاربردی جریان کاری پیکربندی شده است، که توسط یک رکورد ۳تایی مدل‌سازی شده و به‌عنوان CIT=(M, fcl, LLAN) مشخص می‌شود.



شکل (۲): شبه کد روند پیشنهادی

Figure (2): simcode of proposed method.



شکل (۳): نمونه‌ای از یک برنامه جریان کاری

Figure (3): An example of a workflow program

فرض بر این است که ظرفیت ابر برابر با تعداد ماشین‌های مجازی موجود در ابر است، بنابراین M تعداد ماشین‌های مجازی در ابر است، fcl ظرفیت پردازش ابر و LLAN تأخیر انتقال در LAN است.

هر وظیفه vi,f، در V به‌صورت 2-tuple vi,f =(wi,f,si,f) مدل‌سازی می‌شود که wi,f، میانگین تعداد دستورالعمل‌های وظایف vi,f و si,f راهبرد آفلودینگ برای وظیفه vi,f است که می‌تواند به‌عنوان یک بردار یک‌بعدی S به‌صورت زیر بیان شود:

$$S = \{si,f \mid i = 1, 2, \ldots, Nf , f = 1, 2, \ldots, F\} \quad (۳)$$

که در آن Nf تعداد وظایف را درfامین برنامه کاربردی جریان کاری و si,f = 0 نشان می‌دهد که vi,f به‌صورت محلی پردازش می‌شود، si,f=1 نشان می‌دهد vi,f به ابر بارگذاری می‌شود. به‌طور مشابه، vi,f=2 نشان می‌دهد، vi,f به ابر بارگذاری می‌شود.

۲-۳- اهداف کارایی

در زیر به ۳ هدف کارایی مختلف که در این مقاله استفاده می‌شود، اشاره می‌گردد:

الف- زمان انتظار: مصرف زمان کلی عمدتاً شامل سه جنبه است یعنی زمان انتظار، زمان پردازش و زمان انتقال که در اینجا زمان انتظار مدنظر قرار گرفته است. فرض بر این است که فاصله زمان رسیدن وظیفه از توزیع نمایی پیروی می‌کند و زمان سرویس ابر وابسته به وظایف قبلی و طول صف است. به‌طورکلی میانگین زمان انتظار برای وظایف در ابر با متغیر Wait-time بیان می‌گردد.

ب- مدل مصرف انرژی: کل انرژی مصرفی برنامه کاربردی جریان کاری شامل مصرف انرژی پردازش و انتقال است. Epro(vi,f) میزان مصرف انرژی حاصل از پردازش وظیفه vi,f را نشان می‌دهد، در حالی‌که Etrans(vi,f , vj,f) نمایانگر مصرف انرژی تولید شده توسط انتقال داده از وظیفه vi,f به vj,f بر روی دستگاه‌های سیار است. کل انرژی مصرفی fامین برنامه کاربردی جریان کاری به‌صورت زیر است:

$$E_{waf}(S)= \sum_{vi,f \hat{I} v} E_{pro}(v_{i,f}) + \sum_{i=1}^{j-1} \sum_{j}^{n} E_{trans}(v_{i,f}, v_{j,f}) \quad (۴)$$

ج- حالت هزینه: کاربر سیار مجبور است مبلغی مرتبط به استفاده‌ای که از ابر دارد را بپردازد. فرض بر این است که قیمت تمام‌شده برای ابر برابر با a و ابر از راه دور 2G است. این عبارت به معنای این است که اگر کار کاربر به‌صورت محلی پردازش شود، هزینه آن صفر است. اگر وظیفه برای پردازش به ابر بارگذاری شود، هزینه آن a خواهد بود. به همین ترتیب، اگر کار به ابر بارگیری شود، هزینه 2G خواهد بود. میانگین هزینه به‌صورت رابطه ۵ و ۶ نشان داده می‌شود که در آن N تعداد گره‌ها یا وظایف یک برنامه کاربردی جریان کاری است.

$$E_{waf}(S)= \frac{1}{N} \sum_{vi,f \hat{I} v} C \quad (۵)$$

$$C= \begin{cases} 0 \ldots\ldots\ldots\ldots s_{i,f}=0 \\ a \ldots\ldots\ldots\ldots s_{i,f}=1 \\ 2a \ldots\ldots\ldots\ldots s_{i,f}=2 \end{cases} \quad (۶)$$

هدف در این مطالعه بهینه‌سازی میزان تأخیر زمانی، انرژی و همچنین هزینه تمام برنامه‌های کاربردی جریان کاری در حین برطرف کردن محدودیت مهلت داده شده توسط برنامه کاربردی جریان کاری است.

ما برای حل این مسائل یک بهینه‌سازی چندمنظوره با الگوریتم بهینه‌سازی ازدحام ذرات را برای مسئله بارگذاری محاسبات وظایف جریان کاری ابری از طریق تخلیه بار بین دستگاه سیار، تکه ابر و پردازش ابری ارائه داده‌ایم. هدف اصلی بهینه‌سازی چندمنظوره پیدا کردن مجموعه‌ای از راه‌حل‌های بهینه قابل قبول است. این راه‌حل‌های تولیدی توسط روش پیشنهادی با الگوریتم‌های تکاملی که قابل قبول بوده به کاربر توانایی بیشتر به‌منظور اتخاذ یک تصمیم آگاهانه با مدنظر قراردادن دامنه وسیعی از راه‌حل نزدیک به بهینه ارائه داده که از یک چشم‌اندازِ «کلی» دارای حالت نزدیک به بهینه هستند. یکی از اصلی‌ترین چالش‌ها برای به‌کارگیری الگوریتم بهینه‌سازی ازدحام ذرات در مسئله بارگذاری محاسبات وظایف جریان کاری ابری از طریق تخلیه بار این است که چگونه به‌عنوان یک راه‌حل جستجو وارد یک زمان‌بند گردیده، نقشه‌های مناسب را میان راه‌حل‌های مسئله پیدا کرده و اینکه الگوریتم بهینه‌سازی ازدحام ذرات چگونه با راه‌حل‌های اولیه تولید شده، عمل نماید.

۳-۳- نگاشت مسئله بارگذاری محاسبات به الگوریتم بهینه‌سازی ازدحام ذرات

به دلیل اهمیت مسئله زمان‌بندی موازی وظایف در مراکز داده‌ای ابری، در سال‌های اخیر مطالعات زیادی در این زمینه صورت گرفته است و توجه پژوهشگران به‌خوبی به این زمینه تحقیقاتی معطوف گردیده است. در برخی مطالعات الگوریتم‌هایی تکاملی برای این مسئله استفاده شده است. در این مقاله، الگوریتم بهینه‌سازی ازدحام ذرات را روی مسئله بارگذاری محاسبات رایانش لبه از طریق تخلیه بار ارائه داده‌ایم. در الگوریتم بهینه‌سازی ازدحام ذرات به مجموعه‌ای از مقادیر که یک راه‌حل برای حل یک مسئله است، ذره می‌گویند. در فرآیند تکامل، تغییر ماهیت ذرات نسل‌های جدید بر اساس ترکیب بهترین ذرات نسل‌های قبل اتفاق می‌افتد تا بهترین راه‌حل‌ها را باهدف دستیابی به راه‌حل بهینه ترکیب می‌کند. سپس راه‌حل‌های جدید تشکیل‌شده دچار تغییر می‌شود. تغییر به این معناست که مقادیر کمی تغییر پیدا می‌کنند و این تغییرات اغلب نتیجه نسخه‌برداری غلط از حالت‌های قبلی ذرات است. در این مقاله از یک مدل ذره خاص استفاده شده است که یک راهبرد زمان‌بندی موازی وظایف بر روی مراکز داده‌ای ابری که برآورده‌کننده محدودیت‌های مختلف اجرایی بوده، توسط ذره ارائه شده است. در طی هر نسل، ذرات بر اساس سازگاری افراد مختلف در دامنه مشکل خاصی انتخاب می‌شوند. پس افراد توسط اپراتورهای بهینه‌سازی ازدحام ذرات حرکت می‌کنند و سپس جمعیت جدید که ارائه دهنده مجموعه راه‌حلی جدید است تولید می‌شود. بهترین راه‌حل انتخاب خواهد شد و راه‌حل‌های ضعیف بعد از چندین نسل تکامل حذف خواهند شد. بر اساس وضعیت واقعی محاسبات ابری، این مقاله استراتژی مسئله بارگذاری محاسبات وظایف جریان کاری ابری از طریق تخلیه بار بین دستگاه سیار، تکه ابر و پردازش ابری را از طریق الگوریتم بهینه‌سازی ازدحام ذرات پیشنهاد می‌کند. در ادامه به بیان جزییات نگاشت الگوریتم بهینه‌سازی ازدحام ذرات به مسئله پیش روی این مقاله می‌پردازیم.

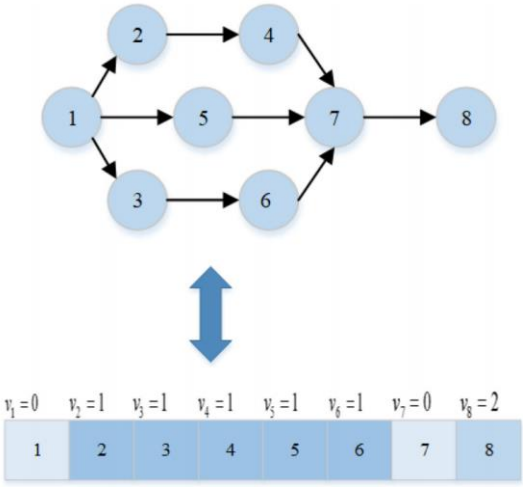
۱-۳-۳- نگاشت ذره به فضای مسئله

برنامه کاربردی جریان کاری با استفاده از نتایج توپولوژیک با یک عدد صحیح شماره‌گذاری می‌شود و از {۰،۱،....} شروع می‌شود. ذره مقدار هر متغیر تصمیم را بیان می‌کند و همچنین استراتژی بارگذاری هر وظیفه از برنامه کاربردی جریان کاری را نشان می‌دهد. مجموعه ذره‌های مرتبط یک پاسخ کامل را تشکیل می‌دهند و یک راه‌حل برای مسئله بهینه‌سازی را نشان می‌دهد. تعداد پاسخ ها جمعیتی را تشکیل می‌دهند که این نشانگر تنوع راه‌حل است. هر پاسخ یک استراتژی بارگذاری محاسباتی را برای برنامه‌های کاربردی جریان کاری نشان می‌دهد. از روش کدگذاری عدد صحیح استفاده می‌شود، یعنی هر استراتژی بارگذاری به‌صورت {۰،۱،۲}رمزگذاری می‌شود. عدد صفر نشان می‌دهد که هر وظیفه برنامه کاربردی جریان کاری توسط خود دستگاه‌های سیار پردازش می‌شود و شماره ۱ نشان می‌دهد وظیفه برنامه کاربردی جریان کاری به لبه بارگذاری شده است. به همین ترتیب، عدد ۲ نشانگر وظیفه برنامه کاربردی جریان کاری بر اساس استراتژی‌های بارگذاری به ابر بارگیری می‌شود. بر اساس اینکه هر وظیفه در دستگاه سیار یا لبه اجرا گردد، یا اینکه به محاسبات ابری برای اجرا تخلیه گردد، با محاسبه تابع هدف به سمت جواب بهینه همگرا می‌گردد و بهترین راه‌حل برای تخلیه محاسبات جستجو می‌گردد.

همان‌طور که در شکل (۴) نشان داده شده است، نمونه‌ای از رمزگذاری هر ذره در مسئله پیش رو ارائه شده است. هر جعبه روی ذره پایین نشان‌دهنده یک موقعیت ذره است و همچنین یک وظیفه برنامه کاربردی جریان کاری را نشان می‌دهد. مقدار احتمالی هر موقعیت ذره {۰،۱،۲} است و به‌عنوان ovi برابر ۲، vi برابر ۱ و vi برابر صفر بیان می‌شود که در بالای آن موقعیت ذره نشان داده شده است. به‌علاوه موقعیت ذره با همان رنگ به معنای داشتن استراتژی بارگذاری یکسان است.

۲-۳-۳- تابع برازش الگوریتم بهینه‌سازی ازدحام ذرات

تابع برازش معیار ارزیابی کیفیت فردی است که بعد از نرمال سازی به‌وسیله رابطه (۷) توسط رابطه (۸) ارائه می‌شود. سه جزء تابع برازش مدل بارگذاری محاسباتی به ترتیب میزان زمان تأخیر، مصرف انرژی و هزینه را نشان می‌دهد. ضروری است مصالحه میان چندین تابع هدف انجام شود.



شکل (۴): مثالی از نمایش ذره در روش پیشنهادی

Figure (4): An example of particle representation in the proposed method

به عبارت دیگر، ما باید بهترین استراتژی بارگذاری را به دست آوریم تا سه جزء این تابع برازش به‌خوبی انجام شود. در اینجا به چندین ویژگی و تابع هدف و معیار کارایی توجه کرده‌ایم. این معیارها بر روی اهداف زمان تأخیر، مصرف انرژی و هزینه توجه نموده‌اند. به طور معمول کاربران این فاکتورهای کیفیتی را بیشتر از سایر کاربران مدنظر قرار می‌دهند و در پی یافتن خدمتی ترکیبی با ترکیب مناسبی از آن‌ها هستند. در بسیاری از کاربردها اهمیت هر یک از این اهداف می‌توانند متفاوت از یکدیگر بوده و کاربر ممکن است حتی به برخی از آن‌ها هیچ اهمیتی ندهد. برای روبرو شدن با این مسئله در این مقاله به کاربر این آزادی عمل داده می‌شود تا میزان اهمیت هر یک از این فاکتورهای کیفیتی را تعیین کند. در اینجا سه هدف کارایی زمان تأخیر، مصرف انرژی و هزینه نشان داده می‌شوند و سه ضریب وزنی برای آن‌ها در نظر گرفته می‌شود که به ترتیب با α، β و γ نشان داده می‌شود. تحقیقات موجود غالباً همه معیارها را در درون یک هدف قرار می‌دهند. این کار از چشم‌اندازه یک بهینه‌سازی چندمنظوره دارای تناسب بوده چرا که بجای مجموعه‌ای از راه‌حل‌ها که در یک زمینه چندمنظوره موردنیاز بوده به راه‌حلی تکی منجر می‌شوند و کیفیت با یک عدد واحد سنجیده می‌شود. البته باید به این نکته توجه داشت که چون محدوده هر یک از اهداف کارآیی با یکدیگر متفاوت است، باید ابتدا عملیات نرمال‌سازی را بر روی این اهداف انجام داده و در نهایت آن‌ها را برحسب یک تابع هدف واحد قرار داد. نرمال‌سازی هر یک از اهداف کارایی از طریق رابطه (۷) صورت می‌پذیرد. برای نرمال کردن پارامترهای اهداف کارایی معیارهای زیر به کار گرفته شده‌اند.

$$NewValue=\frac{Value-Xmin}{Xmax-Xmin} \quad (۷)$$

پس از نرمال‌کردن پارامترهای کیفیت خدمات، مرحله بعدی تعیین کردن تابع هدف یا تابع تناسب است. ما اینجا با سه هدف زمان تأخیر، مصرف انرژی و هزینه مواجه هستیم که هر سه فاکتور زمان تأخیر، مصرف انرژی و هزینه باید کاهش داده شود. بنابراین می‌توان از جمع این اهداف استفاده نمود. برای این کار از رابطه زیر استفاده می‌گردد.

$$Fitness=\alpha*Wait_time+\beta*E_{waf}(S)+\gamma*E_{waf} \quad (۸)$$

به‌طوری‌که:

$$\alpha+\beta+\gamma=1 \quad (۹)$$

این تابع تناسب و برازش، بهترین حالت خود را زمانی می‌یابد که میزان به‌دست‌آمده از محاسبه این کسر کوچک‌ترین مقدار ممکن باشد. این تابع به‌عنوان معیار ارزیابی هر یک از راه‌حل‌های ایجادشده توسط ذرات در الگوریتم بهینه‌سازی ازدحام ذرات در فرآیند تکامل است. بعد از ایجاد راه‌حل‌ها توسط ذرات که بیانگر ترکیب‌های متفاوت از تخلیه بار رایانش لبه است، معیار وزن دهی شده بیان‌شده را به‌عنوان متریک‌های اندازه‌گیری مدنظر قرار داده که کیفیت خدمات را یکپارچه می‌سازد. این مقادیر

وزنی به صورت سعی خطا بر اساس خروجی های مسئله وزن دهی خواهند شد و بر این اساس خود الگوریتم را بهینه خواهد کرد.

۳-۳-۳- مقداردهی اولیه‌ی جمعیت

الگوریتم‌های بهینه‌سازی ازدحام ذرات ساده به‌صورت تصادفی جمعیت اولیه را ایجاد می‌کنند و افراد عملکرد پایینی دارند. در این مقاله، الگوریتم مبتنی بر ساختار اولویت تعیین‌شده استفاده می‌شود. ما جمعیت اولیه را از طریق انجام انتخاب بهینه بر اساس وضعیت فعلی ایجاد می‌کنیم.

۴-۳-۳- عملیات تغییر موقعیت ذرات

الگوریتم PSO یک روش سراسری کمینه‌سازی است که با استفاده از آن می‌توان با مسائلی که جواب آن‌ها یک نقطه یا سطح در فضای D بعدی است، برخورد نمود. در این‌چنین فضایی، فرضیاتی مطرح می‌شود و یک سرعت ابتدایی به ذرات اختصاص داده می‌شود. همچنین کانال‌های ارتباطی بین ذرات در نظر گرفته می‌شود. سپس این ذرات در فضای پاسخ حرکت می‌کنند و نتایج حاصله بر مبنای یک «ملاک شایستگی» پس از هر بازه زمانی محاسبه می‌شود. با گذشت زمان، ذرات به سمت ذراتی که دارای ملاک شایستگی بالاتری هستند و در گروه ارتباطی یکسانی قرار دارند، شتاب می‌گیرند. علی‌رغم اینکه هر روش در محدوده‌ای از مسائل به‌خوبی کار می‌کند، این روش در حل مسائل بهینه‌سازی پیوسته موفقیت بسیاری از خود نشان داده است. منبع الهام الگوریتم PSO، رفتار اجتماعی حیوانات، همانند حرکت دسته جمعی پرندگان و ماهی‌ها بود. از این جهت که PSO نیز با یک ماتریس جمعیت تصادفی اولیه، شروع می‌شود، شبیه بسیاری دیگر از الگوریتم‌های تکاملی همچون الگوریتم ژنتیک پیوسته و الگوریتم رقابت استعماری است. برخلاف الگوریتم ژنتیک، PSO هیچ عملگر تکاملی همانند جهش و تزویج ندارد. از این جهت می‌شود گفت که الگوریتم PSO شباهت بیشتری به الگوریتم رقابت استعماری دارد تا به GA. هر عنصر جمعیت، یک ذره نامیده می‌شود که معادل کروموزوم در GA و یا کشور در الگوریتم رقابت استعماری است. در واقع الگوریتم PSO از تعداد مشخصی از ذرات تشکیل می‌شود که به‌طور تصادفی، مقدار اولیه می‌گیرند. برای هر ذره دو مقدار وضعیت و سرعت تعریف می‌شود که به ترتیب با یک بردار مکان و یک بردار سرعت، مدل می‌شوند. این ذرات، به‌صورت تکرار شونده‌ای در فضای D بعدی مسئله حرکت می‌کنند تا با محاسبه مقدار بهینگی به‌عنوان یک ملاک سنجش، گزینه‌های ممکن جدید را جستجو کنند. بعد فضای مسئله، برابر تعداد پارامترهای موجود در تابع موردنظر برای بهینه‌سازی می‌باشد. یک حافظه به ذخیره بهترین موقعیت هر ذره در گذشته و یک حافظه به ذخیره بهترین موقعیت پیش آمده در میان همه ذرات، اختصاص می‌یابد. با تجربه حاصل از این حافظه‌ها ذرات تصمیم می‌گیرند که در نوبت بعدی، چگونه حرکت کنند. در هر بار تکرار، همه ذرات در فضای D بعدی مسئله حرکت می‌کنند تا بالاخره نقطه بهینه عام، پیدا شود. ذرات، سرعت‌ها و موقعیت‌شان را برحسب بهترین جواب‌های مطلق و محلی به‌روز می‌کنند [۱۵،۱۶]. پیش از بیان دقیق روش کار PSO، ابتدا چهار پارامتر اصلی این الگوریتم تعریف می‌شود: xid معرف موقعیت مکانی بعد d ذره iام است، vid معرف موقع سرعت بعد d ذره i ام است، P_best معرف بهترین تجربه کسب شده توسط ذره iام از ابتدا تاکنون است و G_best معرف بهترین تجربه کسب شده توسط تمام ذرات از ابتدا تاکنون است. هر ذره در هر دور از اجرای الگوریتم، بر اساس رابطه (۱۰) سرعت خود را اصلاح می‌نماید. بر اساس این رابطه سرعت جدید برآیندی از سرعت فعلی، تفاوت مکانی ذره با بهترین جواب خود و تفاوت مکانی ذره با بهترین جواب سراسری است.

$$v_{id}(t+1)=w.v_{id}(t)+c_1.rand(p_best_{id}(t)-x_{id}(t))+c_2.Rand(g_best_{d}(t)-x_{id}(t)) \quad (۱۰)$$

که در آن t شمارنده دور الگوریتم است که در بازه [1,N]، w ضریب اینرسی در بازه [0,1]، c1 و c2 ضرایب یادگیری یا شتاب است که در بازه [0,2] انتخاب می‌شود و در بیشتر موارد برابر ۲ است. همچنین rand و Rand اعداد تصادفی در بازه [0,1] است و V سرعت ذره که برای جلوگیری از واگرایی الگوریتم است. مقدار نهایی سرعت هر ذره در بازه رابطه (۱۱) محدود می‌شود.

$$v_{id} \hat{I} [-v_{max}, v_{max}] \quad (۱۱)$$

پس از محاسبه سرعت جدید ذره، از طریق رابطه (۱۲) که از قوانین فیزیک حرکت گرفته شده است، مکان جدید ذره به‌دست آید:

$$x_{id}(t+1)=x_{id}(t)+v_{id}(t+1) \quad i=1,2,3,...,N \quad d=1,2,3,...,D \qquad (۱۲)$$

همچنین در هر دور اجرای الگوریتم برای ارزیابی پاسخ هر ذره نیازمند یک تابع برازش یا برازندگی هستیم تا نشان دهد که موقعیت فعلی ذره چقدر به جواب مطلوب نزدیک است. پس از تعیین مقدار برازندگی تمام ذرات، در پایان هر دور و پیش از شروع دور جدید الگوریتم، مقادیر P_best و G_best محاسبه می‌شوند. در این الگوریتم P_best در ابتدا مقداردهی می‌شود و در ادامه بر اساس حرکت ذرات به سمت بهینه های هر مرحله این مقدار بر اساس مقادیر جدید بروزرسانی صورت می‌گیرد و مقادیر جدید بر اساس مقادیر موقعیت ذراتی که تابع هدف بهینه‌تری دارند بروز می‌شوند. پارامتر G_best نیز بر اساس مقدار برتر P_best بروزرسانی می‌شود. نتایج آزمایشی بر این دلالت داشته که این الگوریتم پیشنهادی در مقایسه با دیگر الگوریتم‌های بهینه‌سازی نشأت گرفته از طبیعت، بهتر عمل می‌کند.

۴- ارزیابی روش پیشنهادی

در رایانش لبه، کارایی انرژی به چالشی برای قابلیت فناوری تبدیل شده است که با صرفه‌جویی در انرژی باطری دستگاه‌های لبه‌ای، مزایای متعددی برای وسایل سیار با نیروی باطری عرضه می‌کند. در حال حاضر، مشخصه سیار در مورد آینده فناوری ابری تصمیم‌گیری کرده و برای وسایل سیار، مدیریت انرژی برای طول عمر آن‌ها اهمیت زیادی دارد. پس کارایی انرژی وسایل سیار و لبه‌ای مهم است چرا که در حال حاضر، رایانش لبه اجازه اجرای برنامه‌های موجود با انرژی کم را نمی‌دهند. بنابراین یکی از چالش‌های پیش روی رایانش لبه تأمین انرژی و صرفه‌جویی در مصرف آن است. انرژی در رایانش لبه یا در بخش ارتباطات یا در بخش محاسبات مصرف می‌گردد. راه‌حل‌هایی برای بهینه‌سازی انرژی وجود دارند که می‌توانند در کاهش توان مصرفی تا حدود زیادی مؤثر واقع شوند. با توجه به NP هارد بودن مسئله تخصیص منابع، رویکردهای مبتنی بر روش‌های قطعی در این زمینه کارا نخواهند بود. لذا در این مقاله پس از بررسی نقاط ضعف و قوت روش‌های پیشین، با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات، یک روش نوین به‌منظور تخصیص منابع محاسباتی در شبکه‌های توزیع شده محاسبات لبه ارائه گردید، که مسئله زمان‌بندی کارای وظایف ورودی را مدنظر قرار داده است. روش پیشنهادی دارای نتایج جدید و متفاوتی نسبت به سایر روش‌های پیشین بود. در این بخش نتایج شبیه‌سازی به همراه پارامترها و جزییات آن بیان می‌گردد و سپس نتایج به‌دست آمده همراه با جزییات تحلیل می‌گردد. همچنین برای اثبات کارآیی روش پیشنهادی، مقایسه نیز با سایر روش‌های موجود در این حوزه انجام شده است.

۴-۱- ویژگی‌های محیط پیاده‌سازی

برای پیاده‌سازی روش پیشنهادی از زبان برنامه‌نویسی پایتون در نرم‌افزار آناکوندا استفاده شده است. این شبیه‌سازی بر روی دستگاهی کامپیوتری با پردازنده کور i7 با میزان حافظ اصلی ۸ گیگابایت انجام شده است.

۴-۲- مدل شبیه‌سازی

در این بخش، شبیه‌سازی و آزمایش جامع برای ارزیابی عملکرد روش پیشنهادی الگوریتم پیشنهادی انجام شده است. به‌طور خاص، تنظیم شبیه‌سازی در مرحله اول معرفی می‌شود از جمله تنظیمات پارامتر آزمایش و سایر روش‌های مقایسه‌ای. سپس تأثیر مقیاس‌های مختلف برنامه کاربردی جریان کاری بر عملکرد مصرف انرژی، عملکرد مصرف زمان و عملکرد هزینه روش‌های مقایسه شده و الگوریتم پیشنهادی ارزیابی می‌شود. در ادامه ما پنج آزمایش که در هرکدام تعداد متفاوتی از ماشین‌های مجازی، میزبان‌ها وظایف استفاده شده است را ترتیب دادیم و در آزمایش‌ها از الگوریتم تکاملی پیشنهادی و نیز دو مورد از مطالعات پیشین، برای استقرار اولیه ماشین‌های مجازی استفاده کردیم. برای این‌که نتایج شبیه‌سازی، همانند اجرا بر روی یک محیط واقعی باشد، از یک حجم کاری که از یک سیستم واقعی به‌دست آمده است، استفاده کرده‌ایم. برای ارزیابی

روش پیشنهادی از شبیه‌سازی قابل اجرا و مبتنی بر محیط واقعی استفاده می‌گردد که این امر در افزایش کیفیت روش پیشنهادی بسیار مهم است. ما برای آزمایش‌های داده‌های ارائه‌شده از پروژه CoMon و مربوط به زیرساخت‌های مانیتورینگ در پلنت‌لب[11] که از آدرس http://comon.cs.princeton.edu قابل دسترس است، استفاده کرده‌ایم. این داده‌ها مربوط به اطلاعات استفاده از پردازنده‌های بیش از یک هزار سرور واقع در بیش از ۵۰۰ مکان مختلف در سراسر جهان است. داده‌ها بر اساس بازه‌های هر پنج دقیقه یک‌بار در طول دوره‌ای از ۱۰ تا ۱۹ مه سال ۲۰۱۰ جمع‌آوری شده است. وظایف نیز به‌خوبی تعیین گردیده‌اند و روش پیشنهادی بر اساس ورودی‌های مسئله، عملیات مکان‌یابی را انجام داده و در نهایت بر اساس معیارهای کیفی بیان‌شده، مورد تحلیل و ارزیابی قرار بگیرد. ما از بخشی از پروژه CoMon را برای ارزیابی‌های خود استفاده کرده‌ایم که در واقع زیرساخت نظارتی بر پلنت‌لب است. در این میان فاصله‌های اندازه‌گیری بهینگی منابع، ۵ دقیقه است. برای هر آزمایش تعداد میزبان مورد استفاده را محاسبه کردیم. در جدول شماره (۱) مشخصات آزمایش‌ها ذکر شده است. تمامی کارها یکسان بوده و هرکدام شامل بارکاری استاندارد پلنت‌لب هستند، اما ماشین‌های مجازی دارای ۳ نوع پیکربندی متفاوت هستند که در جدول (۲) آمده است. و همچنین میزبان‌ها دارای توان پردازشی MIPS (اجرای میلیون دستورالعمل در ثانیه) برابر با ۴۰۰۰ و حافظه RAM برابر با ۲ گیگابایت و پهنای باند bit/sec10000 هستند. لازم به ذکر است که ۳۰ درصد از ماشین‌های مجازی دارای پیکربندی شماره ۱ و ۳۰ درصد هم دارای پیکربندی شماره ۲ و مابقی دارای پیکربندی شماره ۳ هستند. برای ارزیابی روش پیشنهادی از مقایسه با دو مطالعه پیشین که از الگوریتم‌های تکاملی استفاده می‌کنند، بهره می‌بریم. برای این کار از روش‌های ارائه شده در مرجع‌های [۱۷] و [۱۸] با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات بهره برده می‌شود.

۳-۴- پارامترهای الگوریتم‌های تکاملی مورد استفاده

به‌منظور ارزیابی عملکرد و نشان‌دادن کارآمدی رویکرد بهینه‌سازی با الگوریتم بهینه‌سازی ازدحام ذرات، رویه الگوریتم در نرم‌افزار متلب کدگذاری شده و اجرا شده است. در این آزمایش‌ها، پارامترهای دو الگوریتم قرار داده شده‌اند: اندازه جمعیت ۲۰۰ بوده و تعداد تکرارها، متفاوت است. هدف بهینه‌سازی عبارت از به‌دست آوردن یک توالی عملیاتی بهینه بوده که بر طبق رابطه‌های ارائه شده در قسمت‌های قبل منجر به حداقل هزینه گردد.

۴-۴- ارزیابی مصرف انرژی بر اساس زمان اجرا

Table (1): Configuration of experiments

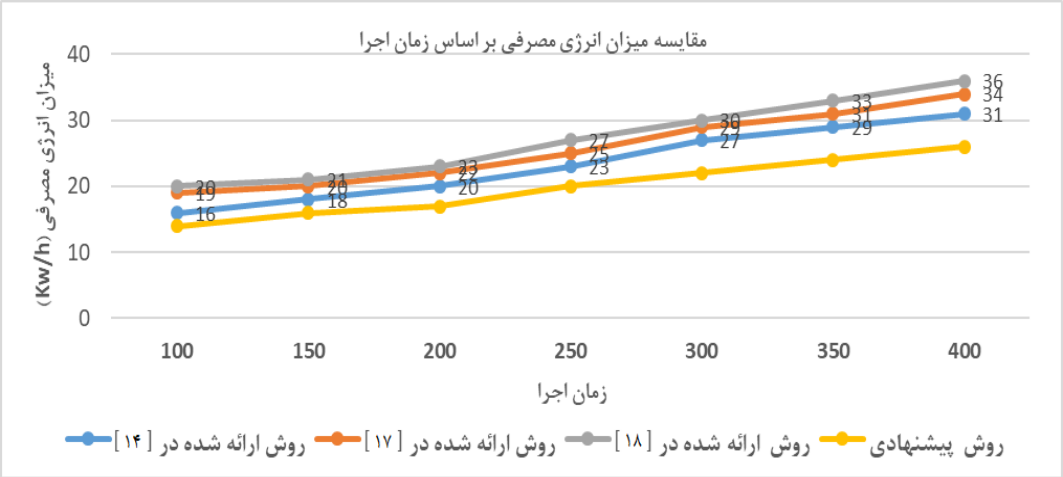
جدول (۱): پیکربندی آزمایش‌ها

آزمایش	تعداد وظایف هر ماشین مجازی	تعداد میزبان	تعداد ماشین مجازی
۱	۵	۵۰	۵۰۰
۲	۱۰	۱۲۰	۱۰۰۰
۳	۱۵	۱۷۵	۱۵۰۰
۴	۲۰	۲۳۰	۲۰۰۰
۵	۲۵	۳۵۰	۲۵۰۰

Table (2): Configuration of virtual machines

جدول (۲): پیکربندی ماشین‌های مجازی

پیکربندی	میزان توان پردازشی برحسب میلیون دستورالعمل در ثانیه	میزان حافظه اصلی موردنیاز برحسب مگابایت	پهنای باند موردنیاز برحسب بیت بر ثانیه
۱	۲۵۰	۲۵۰	۱۰۰۰
۲	۵۰۰	۵۰۰	۱۵۰۰
۳	۱۰۰۰	۱۰۰۰	۲۰۰۰



شکل (۵): نمودار مصرف انرژی بر اساس زمان اجرا

Figure (5): Graph of energy consumption based on execution time

در ادامه مقایسه انجام شده، بر روی زمان اجرای واقعی تمرکز کرده‌ایم. در این بخش برای ارزیابی روش پیشنهادی از مقایسه روش پیشنهادی با دو الگوریتم تکاملی مشابه یعنی الگوریتم تفاضل تکاملی و الگوریتم بهینه‌سازی ازدحام ذرات مقایسه می‌گردد. بر اساس نتایج به‌دست آمده، این مسئله مشخص است که الگوریتم پیشنهادی بر اساس الگوریتم بهینه‌سازی ازدحام ذرات برای کاهش مصرف انرژی در تخصیص منابع محاسباتی رایانش لبه، منجر به یک همگرایی سریع‌تر برای یک راه‌حل بهتر در مقایسه با الگوریتم‌های پیشین برای هر تعداد از نسل‌ها می‌گردد. در این بخش مقایسه هایی بر اساس زمان اجرا با دو الگوریتم تکاملی مشابه صورت گرفته است. نتایج در شکل (۵) نشان داده شده که در آنجا، محور افقی نشان دهنده زمان اجرا است.

همان‌طور که مشاهده می‌شود الگوریتم بهینه‌سازی ازدحام ذرات پیشنهادی از دو الگوریتم تکامل تفاضلی و الگوریتم بهینه‌سازی ازدحام ذرات بهتر عمل کرده و توانسته است با بهره‌گیری از مزایای هر یک، معایب آن‌ها را نیز پوشش دهد. نتیجه نهایی این الگوریتم نشان دهنده پیشرفت محسوسی نسبت به سایر الگوریتم‌های پایه داشته باشد. در این پژوهش از سه هدف زمان تأخیر، مصرف انرژی و هزینه استفاده شده است، بنابراین به دلیل استفاده از تابع هدف انرژی الگوریتم پیشنهادی نتایج مناسب‌تری را داشته است و همان‌طور که در این شکل نشان داده شده نتایج بهینه شده است چون در هر دور این پارامتر بر روی فضای مسئله تاثیر مثبتی داشته است.

۵-۴- ارزیابی زمان اجرای الگوریتم‌های تکاملی بر اساس تعداد نسل متفاوت

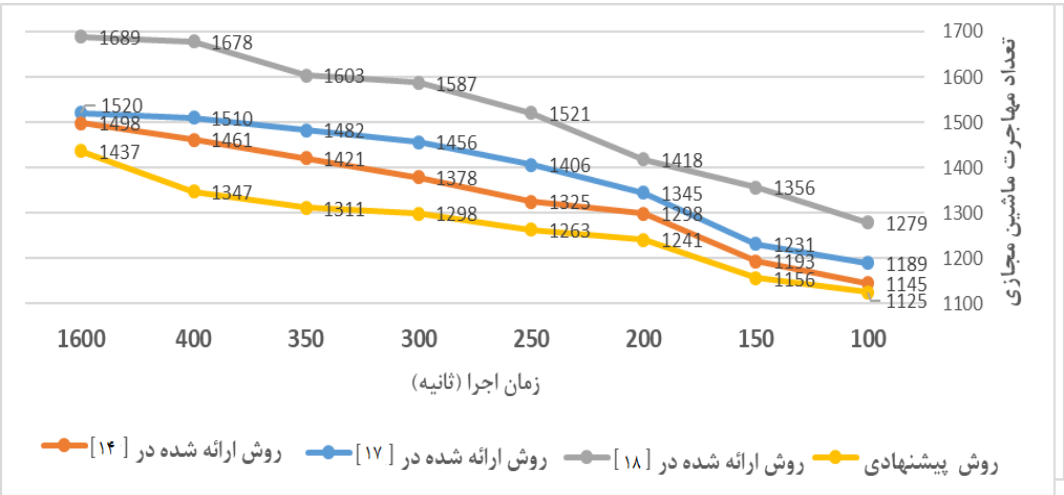
رویکردهای موجود در زمینه تخصیص منابع و تخلیه بار پویا در رایانش ابری لبه به دو بخش ایستا و پویا تقسیم‌بندی می‌شوند. در رویکردهای ایستا، شرایط و محیط برنامه و متغیرهای احتمالی موجود در برنامه از قبل تعیین شده و ثابت هستند و در یک شرایط پایدار و تغییرناپذیر برنامه اجرا و نتایج بررسی و ارزیابی می‌گردند و این نوع رویکردها نسبت به رویکردهای پویا از انعطاف‌پذیری کم‌تری برخوردار هستند و رنج کم‌تری از احتمالات را در خود جای می‌دهند. در حالی‌که در رویکردهای پویا، شرایط و محیط عملیاتی متغیر و پویا هستند و در یک شرایط ناپایدار و تغییرپذیر برنامه‌های کاربردی اجرا می‌شوند و نتایج مورد بررسی و ارزیابی قرار می‌گیرند و همچنین نتایج به‌دست آمده ازاین رویکردها قابلیت اعتماد بیشتری نسبت به نتایج به‌دست آمده از رویکردهای ایستا دارند به این دلیل که فضای عملیاتی و متغیرهای قابل‌تغییر این رویکردها بیشتر و رنج بیشتری از احتمالات و متغیرها را در خود جای می‌دهند. در شرایط تخصیص منابع پویا، استفاده از یک الگوریتم کارآمد جهت تخصیص وظایف بسیار مهم بوده و عملیات تخلیه کارآمد بر روی زمان انتقال نیز بسیار تأثیرگذار است. پیچیدگی این الگوریتم مطابق رابطه (۱۳) است. در این روند n نشان دهنده تعداد ذرات موجود در مسئله است که در شرایط اجرایی الگوریتم تکاملی یک تابع توانی است.

$$C= O(_2\sqrt{n}\ ) \quad (۱۳)$$

روش پیشنهادی برای تخصیص منابع در محاسبات لبه با دو چالش روبه‌رو است: ۱- انتخاب بهترین ماشین فیزیکی برای قراردادن وظایف ماشین مجازی بر روی آن به‌گونه‌ای که کمترین مصرف انرژی را داشته باشیم. ۲- اگر عمل نگاشت به‌درستی انجام شود منجر به کاهش زمان اجرای کل در محاسبات ابری خواهد شد که کاهش مصرف انرژی را به دنبال دارد چرا که در عمل تخصیص منابع، پردازنده و پهنای باند درگیر انتقال صفحات حافظه از گره مبدأ به گره مقصد خواهند شد. بنابراین انتخاب بهترین ماشین فیزیکی برای تخصیص منابع موجود به‌منظور کاهش مصرف انرژی امری ضروری و هدف تحقیق است. بنابراین روش پیشنهادی مصرف انرژی را به‌خوبی کم نموده است و در این بخش نیز نتایج بیانگر کمتر بودن زمان اجرای کل در مقایسه با روش‌های پایه است. این امر به این دلیل است که روش پیشنهادی با تعریف تابع هدف مناسب برحسب تخلیه مناسب بار و توزیع عادلانه وظایف بر روی منابع محاسباتی ابری بوده و با در نظر گرفتن انرژی مصرفی به دنبال استفاده از مزایای جستجوی تصادفی و تکاملی الگوریتم بهینه‌سازی ازدحام ذرات بوده و دارای کارآیی بهتری نسبت به روش‌های مشابه است.

۶-۴- مقایسه تعداد مهاجرت‌ها در مطالعات مختلف بر اساس پیشرفت زمان اجرا

تعداد مهاجرت فاکتور بسیار مهمی در ارزیابی روش‌های تخصیص منابع محاسباتی است. حجم کاری زیاد ناشی از انتقال ماشین مجازی از یک تکه ابر، باعث کاهش کیفیت سرویس‌های ارائه شده توسط سرور مجازی در حال انتقال در رایانش لبه می‌گردد. در شرایطی که مشتریان بر اساس سطح سرویس مشخصی قرارداد بسته‌اند، سرور قادر به سرویس‌دهی بر اساس مفاد قرارداد نشده و از مفاد آن تخطی خواهد نمود. بنابراین ارائه یک روش بهینه با سربار کم جهت تخلیه بار در رایانش لبه ضروری است. در این مقاله الگوریتمی جهت تخلیه بار در رایانش لبه ارائه گردید که در بخش ۳ با جزییات تشریح گردید و در این بخش کارایی این روش از نظر تعداد مهاجرت (انتقال از پردازش لبه به رایانش ابری) در مقایسه با سایر روش‌های پایه می‌گردد. نتایج ارائه شده در شکل (۶) با هم مقایسه شده‌اند. از نتایج شکل (۶) به‌روشنی مشخص است که استفاده از الگوریتم ترکیبی پیشنهادی جهت انتقال وظایف باعث کاهش تعداد صفحات ارسال شده، مدت مهاجرت و زمان قطع اجرای ماشین مجازی می‌شود و تعداد مهاجرت‌ها را در مقایسه با روش‌های ارائه شده پیشین به‌خوبی کاهش می‌دهد که این امر بیانگر و اثبات‌کننده کارایی روش پیشنهادی است. این امر به این دلیل است که روش ترکیبی با تعریف تابع هدف مناسب برحسب جایابی بهینه وظایف و با در نظر گرفتن انرژی مصرفی به دنبال استفاده از مزایای هر دو الگوریتم به‌صورت توأمان بوده و نقاط ضعف هر یک را پوشش داده است و دارای کارایی بهتری نسبت به دو الگوریتم تفاضل تکاملی و بهینه‌سازی ازدحام ذرات است.



شکل (۶): مقایسه تعداد مهاجرت‌ها بر اساس زمان اجرا در روش‌های مختلف

Figure (6): Comparison of the number of migrations based on execution time in different methods

در این پژوهش از سه هدف زمان تأخیر، مصرف انرژی و هزینه استفاده شده که هر سه فاکتور زمان تأخیر، مصرف انرژی و هزینه باید کاهش داده شده است، بنابراین همان‌طور که در شکل (۵) نشان داده شده است تعداد مهاجرت‌ها بر اساس زمان اجرا در روش‌های مختلف بهبود پیدا کرده است و دلیل این بهبود استفاده همزمان از این سه تابع است که مصرف انرژی و هزینه تاثیری بیشتری بر روی این پارامتر دارد.

۵- نتیجه‌گیری و پیشنهادات

در این مقاله سعی شد برای حل مسئله تخلیه بار در محاسبات لبه یک بهینه‌سازی چندمنظوره با الگوریتم‌های تکاملی را برای انتخاب بهینه منابع محاسباتی برای تخصیص وظایف ابری ارائه شود. هدف اصلی بهینه‌سازی چندمنظوره پیداکردن مجموعه‌ای از راه‌حل‌های بهینه قابل قبول است. راه‌حل‌های تولیدی توسط روش پیشنهادی با الگوریتم‌های تکاملی که قابل قبول بوده به کاربر توانایی بیشتر به‌منظور اتخاذ یک تصمیم آگاهانه با مدنظر قراردادن دامنه وسیعی از راه‌حل ارائه داده که از یک چشم‌اندازه «کلی» دارای حالت نزدیک به بهینه هستند.

یکی از اصلی‌ترین چالش‌ها برای به‌کارگیری الگوریتم بهینه‌سازی ازدحام ذرات در مسئله تخلیه بار در محاسبات لبه و تخصیص وظایف به ماشین فیزیکی این است که چگونه به‌عنوان یک راه‌حل جستجو، وارد یک زمان‌بند گردیده، نقشه‌های مناسب را میان راه‌حل‌های مسئله پیدا کرده و اینکه الگوریتم بهینه‌سازی ازدحام ذرات چگونه با راه‌حل‌های بیان شده می‌تواند مسئله تخصیص منابع محاسباتی را حل نماید. روش پیشنهادی مصرف انرژی را به‌خوبی کم نموده است و نتایج نیز بیانگر تعداد کمتر مهاجران در مقایسه با روش‌های پایه است. این امر به این دلیل است که روش ترکیبی با تعریف تابع هدف مناسب برحسب تخصیص بهینه منابع به وظایف و با در نظر گرفتن انرژی مصرفی به دنبال استفاده از مزایای هر دو الگوریتم به‌صورت توأمان بوده و نقاط ضعف هر یک را پوشش داده است و دارای کارآیی بهتری نسبت به هر یک از روش‌های تکی است. همچنین روش پیشنهادی به‌خوبی نقطه ضعف روش‌هایی که از الگوریتم تفاضل تکاملی و بهینه‌سازی ازدحام ذرات استفاده می‌کنند را پوشش داده و در بهینه محلی گیر نمی‌افتد و با افزایش تعداد نسل‌ها همواره با بهبود در نتایج مواجه گشته است. همچنین با توجه به استفاده از الگوریتم بهینه‌سازی ازدحام ذرات، کارایی بهتری نسبت به الگوریتم بهینه‌سازی ازدحام ذرات و بهینه‌سازی ازدحام ذرات نیز حاصل گردیده است. نتایج بیانگر این است که روش پیشنهادی راه‌حلی بهتر را در انتهای تکرارها به‌دست آورده و در مقایسه با الگوریتم‌های پیشین به پیشرفتی مناسب برحسب مصرف انرژی دست پیدا کرده است. در مطالعات آینده به دنبال ارائه یک الگوریتم جدید برای حل مشکل تخصیص منابع در یک محیط چند توزیع شده محاسبات لبه خواهیم بود و در این راستا برای مسئله تخصیص منابع در یک محیط چند توزیع شده محاسبات لبه از روش‌های تکاملی دیگر نظیر الگوریتم بهینه‌سازی کرم شب‌تاب[۱۲] [۱۹]، الگوریتم شیرمورچه[۱۳] [۲۰]، الگوریتم بازار بورس[۱۴] [۲۱] و غیره استفاده خواهد شد که هدف آن کاهش مصرف انرژی برای انتخاب بهینه ماشین‌های موجود در ابر و کاهش مصرف انرژی در مراکز داده‌ای است به‌طوری‌که با در نظر گرفتن مهلت انجام وظایف بتوان در مصرف انرژی صرفه‌جویی کرد.

سپاسگزاری

این مقاله مستخرج از پایان‌نامه دوره کارشناسی‌ارشد در دانشگاه آزاد اسلامی است. نویسندگان بر خود لازم می‌دانند مراتب تشکر صمیمانه خود را از همکاران حوزه پژوهشی دانشگاه آزاد اسلامی که ما را در انجام و ارتقای کیفی این مقاله یاری نموده‌اند، اعلام نمایند.

References

مراجع

[1] M. Keshavarznejad, M.H. Rezvani, S. Adabi, "Delay-aware optimization of energy consumption for task offloading in fog environments using metaheuristic algorithms", Cluster Computing, vol. 24, no. 3, pp. 1825-1853, Sept. 2021 (doi: 10.1007/s10586-020-03230-y).

[2] A. Shakarami, M. Ghobaei-Arani, M. Masdari, M. Hosseinzadeh, "A survey on the computation offloading approaches in mobile edge/cloud computing environment: A stochastic-based perspective", Journal of Grid Computing, vol. 18, no. 4, pp. 639–671, Dec. 2020 (doi: 10.1007/s10723-020-09530-2).
[3] A.M.A. Hamdi, F.K. Hussain, O.K. Hussain. "Task offloading in vehicular fog computing: State-of-the-art and open issues", Future Generation Computer Systems, vol. 133, pp. 201-212, Aug. 2022 (doi: 10.1016/j.future.2022.03.019).
[4] A. Banerjee, U.C. Gupta. "Reducing congestion in bulk-service finite-buffer queueing system using batch-size-dependent service", Performance Evaluation, vol. 69, no. 1, pp. 53-70, Jan. 2012 (doi: 10.1016/j.peva.2011.09.002).
[5] A. Shakarami, M. Ghobaei-Arani, M. Masdari, M. Hosseinzadeh, "A survey on the computation offloading approaches in mobile edge/cloud computing environment: a stochastic-based perspective", Journal of Grid Computing, vol. 18, no. 4, pp. 639-671, Aug. 2020 (doi: 10.1007/s10723-020-09530-2).
[6] S. AlShathri, S.A. Chelloug, D.S.M. Hassan, "Parallel meta-heuristics for solving dynamic offloading in fog computing", Mathematics, vol. 10, no. 8, Article Number: 1258, April 2022 (doi: 10.3390/math10081258).
[7] M. Zhang, L. Liu, S. Liu. "Genetic algorithm based QoS-aware service composition in multi-cloud", Proceeding of the IEEE/CIC, Hangzhou, China, Oct. 2015 (doi: 10.1109/CIC.2015.23).
[8] M. Chen, M. Dong, B. Liang, "Resource sharing of a computing access point for multi-user mobile cloud offloading with delay constraints", IEEE Trans. on Mobile Computing, vol. 17, no. 12, pp. 2868-2881, March 2018 (doi: 10.1109/TMC.2018.2815533).
[9] H. Boostanimehr, K. Vijay, "Joint downlink and uplink aware cell association in HetNets with QoS provisioning", IEEE Trans. on Wireless Communications, vol. 14, no. 10, pp. 5388-5401, Oct. 2015 (doi: 10.48550/arXiv.2110.11121).
[10] Q. Yao, T. QS Quek, A. Huang, H. Shan, "Joint downlink and uplink energy minimization in WET-enabled networks", IEEE Trans. on Wireless Communications, vol. 16, no. 10, pp. 6751-6765, Oct. 2017 (doi: 10.1155/2018/7906957).
[11] C. Wang, F. Richard Yu, C. Liang, Q. Chen, L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing", IEEE Trans. on Vehicular Technology, vol. 66, no. 8, pp. 7432-7445, Aug. 2017 (doi: 10.1109/GLOCOM.2018.8647593).
[12] Z. Dong, Y. Liu, H. Zhou, X. Xiao, Y. Gu, L. Zhang, C. Liu, "An energy-efficient offloading framework with predictable temporal correctness", Proceedings of the Second ACM/IEEE Symposium on Edge Computing, pp. 1-12, 2017 (doi: 10.1145/3132211.3134448).
[13] G. Klas, "Edge computing and the role of cellular networks", Computer, vol. 50, no. 10, pp. 40-49, Oct. 2017 (doi: 10.1109/MC.2017.3641649).
[14] B. Huang, Z. Li, P. Tang, S. Wang, J. Zhao, H. Hu, W. Li, V. Chang. "Security modeling and efficient computation offloading for service workflow in mobile edge computing", Future Generation Computer Systems, vol. 97, pp. 755-774, Aug. 2019 (doi: 10.1016/j.future.2019.03.011).
[15] S.R. Nabavi, N. Osati-Eraghi, J. Akbari-Torkestani, "Wireless sensor networks routing using clust-eri-ng based on multi-objective particle swarm optimization algorithm", Journal of Intelligent Procedures in Electrical Technology, vol. 12, no. 47, pp. 29-47, Dec. 2021 (in Persian) (dor: 20.1001.1.23223871.1400.12.3.3.3).
[16] M. Momeny, S. Gharravi, F. Hourali, "Reducing the impact of SYN flood attacks by improving the accuracy of the PSO algorithm by adaptive effective filters", Journal of Intelligent Procedures in Electrical Technology, vol. 10, no. 37, pp. 51-57, May 2019 (in Persian) (dor: 20.1001.1.23223871.1398.10.37.6.0).
[17] Q. Wang, Y. Mao, Y. Wang, L. Wang. "Computation tasks offloading scheme based on multi-cloudlet collaboration for edge computing", Proceeding of the IEEE/CBD, pp. 339-344, Suzhou, China, Sept. 2019 (doi: 10.1109/CBD.2019.00067).
[18] D. Jun, E. Gelenbe, C. Jiang, H. Zhang, Y. Ren. "Contract design for traffic offloading and resource allocation in heterogeneous ultra-dense networks", IEEE Journal on Selected Areas in Communications, vol. 35, no. 11, pp. 2457-2467, Nov. 2017 (doi: 10.1109/JSAC.2017.2760459).
[19] G. Mardanian, N. Behzadfar, "A new method for detection of breast cancer in mammography images using a firefly algorithm", Journal of Intelligent Procedures in Electrical Technology, vol. 10, no. 40, pp. 23-32, March 2020 (in Persian) (dor: 20.1001.1.23223871.1398.10.40.3.3).
[20] L. Abualigah, M. Shehab, M. Alshinwan, S.A. Mirjalili, M.A. Elaziz, "Ant lion optimizer: A comprehensive survey of its variants and applications", Archives of Computational Methods in Engineering, vol. 28, pp. 1397–1416, May 2021 (doi: 10.1007/s11831-020-09420-6).
[21] Z. Song, C. Yu, "Investor sentiment indices based on k-step PLS algorithm: A group of powerful predictors of stock market returns", International Review of Financial Analysis, vol. 83, Article Number: 102321, Oct. 2022 (doi: 10.1016/j.irfa.2022.102321).

زیرنویس‌ها

1. Cloud computing
2. Edge computing
3. Internet of things (IOT)
4. Zettabyte (ZB)
5. Decentralized platform
6. Fog layer
7. Latency management
8. Security and energy efficient computation offloading (SEECO)
9. Weighted self-adaptive inertia weight particle swarm optimization algorithm (WAIW-PSO)
10. cloudlet
11. Planet-Lab
12. Firefly algorithm
13. Ant Lion algorithm
14. Stock market algorithm