APJ 7(1):12-32, Spring 2026

Arman Process Journal (APJ)

Quarterly Journal of ICT APJ

License Number: 87090

Arman Process Journal (APJ)

Homepage: https://www.armanprocessjournal.ir

Improving Data Center Resource Utilization Using Clustering, Fuzzy Logic, and Evolutionary Algorithms

M. Jahanbani [1], S. E. Dashti [2], S. Hamzeloo [3]

[1,3] Department of Computer Engineering, Pasargad Institute of Higher Education, Shiraz, Iran

[2] Department of Electrical and Computer Engineering, Ja.C., Islamic Azad University, Jahrom, Iran

ABSTRACT

RESEARCH PAPER

Received: 2025-12-4
Accepted: 2026-4-11

KEYWORDS:
Cloud Computing,
Scheduling,
Virtual Machine Migration,
Evolutionary Algorithms,
Data Centers,

[1]Corresponding author:
seyedebrahim.dashti@iau.ac.ir

Cloud computing is now considered a model beyond traditional distributed computing systems such as Grid and Cluster, which is capable of responding to dynamic requests and diverse user requirements. With the increase in users, there is a need to establish appropriate mechanisms for load balancing and task scheduling. Load balancing is essential for evenly distributing the workload on physical servers, preventing resource congestion, and improving system performance. Due to the user requests for service and the requirement for providing correct and timely responses from service providers, as well as due to the limited resources available in this environment, there is a need for scheduling. In this paper, an approach to optimizing virtual machine migration is proposed by combining genetic algorithms and ant colony optimization for resource scheduling operations, and it uses K Means clustering and fuzzy logic to quantify the dependency between virtual machines and physical machines for migration in order to maintain load balance. In this research, the proposed model is compared with three load balancing algorithms in the CloudSim simulation environment. In this evaluation, our proposed model achieved a 4.5% reduction in task execution time, a 4.9% increase in deadline success ratio, a 3.9% improvement in task diversity. The computational complexity was reduced by 8.3%, the virtual machine migration efficiency was improved by 2.5%, and the decision latency was significantly reduced by 9.5%, and it also achieved energy savings of 30-35%.

نشریه تخصصی آرمان پردازش، دوره ۷، شماره ۱، بهار ۱۴۰۵

فصلنامه تخصصی آرمان پردازش
(APJ)
Homepage: www.armanprocessjournal.ir

فصلنامه تخصصی فناوری اطلاعات و ارتباطات
شماره مجوز: ۸۷۰۹۰

بهبود استفاده از منابع مراکز داده با استفاده از خوشه بندی،منطق فازی و الگوریتم های تکاملی

مژده جهان بانی[۱]، سید ابراهیم دشتی[۲]، سام حمزه لو[۳]

[۱و۳] گروه مهندسی کامپیوتر، موسسه آموزش عالی پاسارگاد، شیراز، ایران
[۲] گروه مهندسی برق و کامپیوتر، واحد جهرم، دانشگاه آزاد اسلامی، جهرم، ایران

چکیده

مقاله پژوهشی

واژگان کلیدی:
محاسبات ابری،
زمانبندی،
مهاجرت ماشین مجازی،
الگوریتم های تکاملی،
مراکز داده،

محاسبات ابری امروزه به عنوان مدلی فراتر از سیستم‌های سنتی محاسبات توزیع‌شده مانند Grid و Cluster مطرح است که توانایی پاسخگویی به درخواست‌های پویا و نیازمندی‌های متنوع کاربران را دارد. با افزایش کاربران، نیاز به استقرار مکانیزم‌های مناسبی جهت توازن بار و زمانبندی کار احساس می‌شود. متعادل‌سازی بار برای توزیع یکنواخت حجم کار در سرورهای فیزیکی، جلوگیری از ازدحام منابع و بهبود عملکرد سیستم ضروری است. با توجه به درخواست کاربران جهت دریافت سرویس و لازمه ی ارائه پاسخ درست و به موقع از سوی سرویس دهندگان و همچنین به دلیل محدود بودن منابع موجود در این محیط، نیاز به زمانبندی احساس میشود. در این مقاله رویکردی برای بهینه‌سازی مهاجرت ماشین مجازی، با ترکیب الگوریتم‌های ژنتیک و بهینه‌سازی کلونی مورچه‌ها برای عملیات زمان‌بندی منابع پیشنهاد شده است و از خوشه‌بندی K Means و منطق فازی برای تعیین کمیت وابستگی بین ماشین‌های مجازی و ماشین‌های فیزیکی جهت مهاجرت به منظور حفظ تعادل بار استفاده می‌کند. در این تحقیق مدل پیشنهادی با سه الگوریتم متعادل سازی بار در محیط شبیه سازی کلودسیم مقایسه شده است. در این ارزیابی مدل پیشنهادی ما به کاهش ۴,۵ درصدی در زمان انجام کار،افزایش ۴,۹ درصدی در نسبت موفقیت در مهلت مقرر،بهبود ۳,۹ درصدی در تنوع وظایف دست یافت.پیچیدگی محاسباتی ۸,۳ درصد کاهش یافت،راندمان مهاجرت ماشین مجازی ۲,۵ درصد بهبود یافت و تأخیر تصمیم گیری به طور قابل توجهی ۹,۵ درصد کاهش یافت و همچنین به صرفه‌جویی در مصرف انرژی به میزان ۳۰-۳۵٪ دست یافته است.

مقدمه

محاسبات ابری به دلیل عملکرد و کارایی بالا و انعطاف پذیری و سودآوری، امروزه مورد توجه سازمانها و شرکت های بزرگ قرار گرفته اند. از آنجایی که سازمان ها به طور فزاینده ای به زیرساخت های ابری متکی هستند، بهینه سازی مهاجرت ماشین های مجازی امری مهم تلقی می شود. در حالی که مهاجرت ماشین های مجازی مزایای بسیاری دارد، این فرآیند مملو از چالش هایی مانند دستیابی به توزیع بار بهینه در طول مهاجرت است.یکی از چالش های مهمی که ارائه دهندگان سرویس های ابری با آن مواجه هستند مدیریت موثر منابع فیزیکی میزبانی شده توسط زیرساخت های فیزیکی می باشد. مصرف کنندگان ابر درخواست سرویس را از هرجایی از جهان به ابر می فرستند. ارائه دهندگان سرویس های ابری باید به کاربران اطمینان دهند که نیازهای آنها به طور کامل برآورده خواهد شد. نکته مهم این است که هر یک از کاربران به دنبال دریافت پاسخی سریع و صحیح از سمت سرویس دهندگان ابر می باشند. با توجه به اینکه حجم درخواست ها بالا میباشد، باید از روشهای مختلف زمانبندی جهت سرویس دهی به مشتریان استفاده کرد.

بنابراین ما با استفاده ازتکنیکها و الگوریتمهای فرا ابتکاری در مراکز داده ابری به دنبال زمانبندی منابع و کاهش تعداد سرورهای فیزیکی در مراکز داده ابری هستیم بدین صورت که از طریق مهاجرت زنده ماشین مجازی بتوانیم بدون توقف سرویس، به یک تعادل نسبی بارکاری سرورها برسیم، سپس بتوانیم از منابع سخت افزاری سرورها حداکثر استفاده را ببریم. این امر منجر به کاهش قابل قبول تعداد سرورها و همچنین کاهش چشمگیر مصرف انرژی و هزینه می‌شود. با توجه به افزایش نگرانی های مرتبط با این موضوع، با استفاده از تکنیک های مختلفی نظیر مجازی سازی، زمانبندی و مدیریت منابع میتوان یک فرایند زیرساختی باهدف متعادل سازی بار در مراکز داده ی ابری ایجاد کرد [۱]. زمانبندی به معنای انتخاب بهترین منبع برای یک وظیفه، یا اختصاص ماشین های مجازی به وظایف طوری که زمان اجرا حداقل شود. توابع هدف به شرح زیر می‌باشند:

min(MS, DD, Energy)
max(VCE, DHR, TD, EE)

قیود مسئله:

$$\sum_{j=1}^{M} x_{ij} = 1 \forall i \in \{1, \dots, N\}$$

$$\sum_{i=1}^{N} x_{ij} \cdot R_i \leq C_j \forall j \in \{1, \dots, M\}$$

$$x_{ij} \in \{0,1\} \forall i, j$$

که در آن (N) تعداد ماشین‌های مجازی (VM)، (M) تعداد ماشین‌های فیزیکی (PM)، (R_i) منبع درخواستی (i) VMام، (C_j) ظرفیت (j) PMام، (x_{ij}) متغیر تصمیم دودویی (اختصاص (i) VM به (j) PM) است. همچنین MS زمان تکمیل پردازش، DD تأخیر تصمیم‌گیری، VCE کارایی محاسباتی VM، DHR نسبت موفقیت در مهلت مقرر، TD تنوع وظایف و EE کارایی اجرا می‌باشند که در بخش ارزیابی تعریف شده‌اند.

پیشینه تحقیق

در حوزه این پژوهش، مقالات و روشهایی ارائه شده است که در ادامه به بررسی برخی از آن تحقیقات پرداخته میشود. در مقاله ای نویسندگان الگوریتم زمانبندی ارایه کرده اند که مبتنی بر امنیت و با استفاده از تکنیک بهینه سازی ازدحام ذرات و یادگیری انطباقی چندگانه میباشد.الگوریتم پیشنهادی با استفاده از یادگیری انطباقی باعث تنوع در جمعیت میشود و همچنین منجر به تعادلی بین عملیات اکتشاف و بهره برداری میشود. این الگوریتم پارامترهای امنیت، زمان بازگشت، بار، مصرف انرژی و هزینه را در حین توزیع کارها مد نظر قرار میدهد. در نتیجه به توزیع بار و کاهش مصرف انرژی منجر میشود. این الگوریتم در محیط رایانش ابری بخصوص در بار کاری بالا بسیار اثربخش میباشد. با این حال، این روش در بارهای کاری با ناهمگنی بالا عملکرد مناسبی ندارد و از قابلیت خوشه‌بندی برای تشخیص وابستگی بین VMها استفاده نمی‌کند. همچنین به دلیل ماهیت PSO، در فضاهای جستجوی بزرگ با خطر همگرایی زودهنگام مواجه است [۲].

نویسندگان روشی برای زمانبندی وظایف در رایانش ابری پیشنهاد کرده اند که بر مبنای الگوریتم رقابت استعماری و الگوریتم کرم شب تاب میباشد. در این روش، از الگوریتم هوش جمعی برای پردازش درخواستهای کاربران و زمانبندی وظایف برای توازن بار استفاده شده است. هدف این کار، بهبود زمانبندی و توازن بار در محیط ابری با استفاده از الگوریتمهای جستجوی محلی و سراسری میباشد. از الگوریتم رقابت استعماری، برای بررسی فضای راه حل و الگوریتم کرم شبتاب، به منظور جستجوی محلی استفاده میشود. در شبیه سازی انجام شده معیارهای پایداری، زمان پردازش، توازن بار و درصد راندمان در نظر گرفته شده است و الگوریتم ارائه شده قادر به انجام تعداد وظایف بیشتری هست که به دلیل ترکیب مناسب دو الگوریتم میباشد. اگرچه این ترکیب جذاب است، اما پیچیدگی محاسباتی بالایی دارد (دو الگوریتم با جمعیت‌های جداگانه) و برای محیط‌های

ابری در مقیاس بزرگ مقیاس‌پذیر نیست. همچنین از مهاجرت زنده VM پشتیبانی نمی‌کند [۳].

در مقاله ای تحت عنوان" ارائه الگوریتم زمانبندی مهاجرت ماشینهای مجازی جهت بهینه سازی همزمان مصرف انرژی و تولید آلایندهها در شبکه محاسباتی ابر" از الگوریتم ژنتیک برای پیش بینی یا تطبیق الگو استفاده میشود. در این کار از روش رگرسیون استفاده میکنند.در اینجا موتور الگوریتم ژنتیک یک جمعیت اولیه ای از فرمول را ایجاد میکند و افراد با مجموعه ای از داده ها مورد آزمایش قرار گرفته و مناسبترین آنها انتخاب میشوند . میبینیم که باگذشت از بین تعداد زیادی از نسلها الگوریتم ژنتیک به سمت فرمولهایی میرود که از دقت بیشتری برخوردارند.

در پژوهشی مشابه، از الگوریتم ژنتیک برای بهینه‌سازی همزمان مصرف انرژی و تولید آلاینده‌ها در مهاجرت ماشین‌های مجازی استفاده شده است. این پژوهش نشان داد که الگوریتم ژنتیک با گذشت نسل‌های متعدد به سمت فرمول‌هایی با دقت بیشتر همگرا می‌شود [۴]. در مقاله ای الگوریتمی پیشنهاد داده اند که با کمک الگوریتم ژنتیک و همچنین در نظر گرفتن تعادل بار سیستم، زمان اجرا و هزینه ی اجرا کاهش پیدا کند.الگوریتم سعی دارد با ایجاد تغییر در الگوریتم ژنتیک و کاهش تعداد ایجاد جمعیت، عملکرد بهتری نشان دهد. هدف اصلی الگوریتم، تخصیص کارها به منابع با در نظر گرفتن طول کارها و تعداد دستورات قابل اجرای بوده است.الگوریتم، کارها را با در نظر گرفتن تعداد کار و توانایی های منابع، به منابع تخصیص میدهد ونتایج اینگونه است که الگوریتم به طور خاص زمان اجرا، هزینه ی اجرا و میانگین درجه عدم تعادل را بهبود میدهد. الگوریتم پیشنهادی کارها را با در نظر گرفتن طول کار و توانایی منبع، به منابع اختصاص داده است. بنابراین کل زمان اجرای هر ماشین مجازی کاهش پیدا کرده است. کاستی اصلی این روش عدم توجه به وابستگی بین VMها و PMها و همچنین نادیده گرفتن عدم قطعیت در فرآیند مهاجرت است که روش پیشنهادی ما با استفاده از خوشه‌بندی فازی به آن می‌پردازد [۵]. در شبکه ابری، پیکربندی ماشین مجازی یک رویه ضروری است که نیازمند به حداقل رساندن هزینه های مربوط به پیکربندی است.با استفاده از تکنیک‌های زمان‌بندی مؤثر که ماشین‌های مجازی پیکربندی شده پویا را روی میزبان‌های کمتری یکپارچه می‌کند، اجرا می‌شود [۶]. مهاجرت ماشین مجازی زنده، استفاده کارآمد از منابع، تحمل خطا، تعادل بار، محاسبات تلفن همراه، مدیریت انرژی و اشتراک منابع را افزایش می دهد. با کاهش فرکانس فیزیکی یا مجازی پردازنده با ایجاد تعادل بین زمان‌بندی پردازنده مجازی برای چارچوب پس از کپی با سرعت پردازش و انتقال در ماشین منبع اجرا می‌شود [۷].

زمان‌بندی یک جنبه حیاتی هستند که به طور قابل توجهی بر عملکرد محیط‌های محاسبات ابری تأثیر می‌گذارند.

یکی از مدل‌های برجسته زمان‌بندی در مقالات ، فرآیند مدل زمان‌بندی مبتنی بر بهینه‌سازی ازدحام ذرات (PSO) است [۸]. الگوریتم بهینه‌سازی ازدحام ذرات یک الگوریتم الهام گرفته از طبیعت است که رفتار اجتماعی پرندگان و ماهی‌ها را شبیه‌سازی می‌کند. اگرچه مدل‌های مبتنی بر بهینه‌سازی ازدحام ذرات اثربخشی خود را در بهینه‌سازی تخصیص وظایف و کاهش زمان انجام کار نشان داده‌اند ، اما اغلب در مدیریت ماهیت پویای حجم کار ابری، عملکرد ضعیفی دارند. علاوه بر این، این مدل‌ها ممکن است به طور کافی تعادل بین توزیع بار و سطوح مصرف انرژی را نشان ندهند [۹]. یکی دیگر از مدل‌های زمان‌بندی قابل توجه، مدل زمان‌بندی مبتنی بر الگوریتم بازپخت شبیه سازی شده (SA) است.که یک تکنیک بهینه‌سازی احتمالاتی است که از فرآیند تبرید در عملیات متالورژی الهام گرفته شده است [۱۰]. در حالی که مدل‌های مبتنی بر الگوریتم بازپخت شبیه سازی شده بهبودهای بالقوه‌ای در تخصیص وظایف و تعادل بار ارائه می‌دهند، پیچیدگی محاسباتی آنها می‌تواند بالا باشد و منجر به عملکرد نامطلوب در محیط‌های ابری در مقیاس بزرگ شود [۱۱و۱۲].

در مقابل الگوریتم‌های ژنتیک و بهینه سازی کلونی مورچگان الگوریتم‌هایی هستند که برای غلبه بر محدودیت‌های مدل‌های مبتنی بر بهینه‌سازی ازدحام ذرات و الگوریتم بازپخت شبیه سازی شده استفاده شده اند. و ضمن تطبیق با بارهای کاری پویا، مسائل بهینه‌سازی پیچیده را به طور مؤثر حل می‌کنند و در نتیجه به توزیع بار بهینه و ویژگی‌های بهره‌وری انرژی دست می‌یابند [۱۳-۱۵].

تحقیقات اخیر نیز نشان داده‌اند که ترکیب الگوریتم‌های تکاملی با یادگیری ماشین می‌تواند نتایج بهتری در مسئله مهاجرت ماشین مجازی ارائه دهند [۱۶-۱۹].

طریقه کار مدل و راه حل پیشنهادی

مدل پیشنهادی از ترکیب الگوریتمهای ژنتیک (GA) و بهینه‌سازی کلونی مورچه‌ها (ACO) برای زمان‌بندی منابع استفاده می‌کند و رویکردی برای توزیع بار ارائه می‌دهد.همچنین کمی کردن وابستگی ماشین مجازی به ماشین فیزیکی با استفاده از خوشه‌بندی K Means و منطق فازی، تعادل بار بهینه را حتی در مواجهه با عدم قطعیت‌های مهاجرت تضمین می‌کند. در جدول زیر کلیه مراحل کار و روش پیشنهادی آورده شده است.

جدول۱. معماری کلی روش پیشنهادی

شماره	نام بلوک	ورودی	خروجی
۱	ورودی پارامترها	ویژگی‌های VM/PM، پارامترهای الگوریتم	داده‌های نرمالایز شده
۲	خوشه‌بندی K-Means و فازی‌سازی	ویژگی‌های VM/PM	درجه عضویت فازی و امتیاز وابستگی A(x)
۳	محاسبه VCM و TRM	A(x) ، MIPS، RAM، BW، UT	VCM و TRM برای هر VM-task
۴	الگوریتم ACO (تولید مسیرهای اولیه)	VCM ، TRM، پارامترهای ACO	جمعیت اولیه راه‌حل‌ها (بهترین مسیرهای فرومون)
۵	الگوریتم GA (بهبود راه‌حل‌ها)	جمعیت اولیه از ACO	راه‌حل‌های بهینه‌شده
۶	تابع تناسب و آستانه Fth	راه‌حل‌ها، VCM، TRM	انتخاب بهترین راه‌حل
۷	تصمیم مهاجرت و خروجی	بهترین راه‌حل	تخصیص نهایی VM به PM

جدول ۱ معماری کلی روش پیشنهادی را نشان می‌دهد. ابتدا ویژگی‌های VMها و PMها وارد سیستم می‌شود. بلوک خوشه‌بندی فازی، وابستگی بین VMها و PMها را تعیین می‌کند. سپس معیارهای VCM و TRM محاسبه شده و به الگوریتم ACO داده می‌شوند تا راه‌حل‌های اولیه ساخته شوند. در ادامه، الگوریتم GA این راه‌حل‌ها را بهبود می‌بخشد. در نهایت، تابع تناسب بهترین تخصیص را انتخاب کرده و تصمیم مهاجرت صادر می‌شود. بر اساس بررسی مدل‌های موجود مورد استفاده برای افزایش کارایی ماشین‌های مجازی متعادل‌کننده بار، می‌توان مشاهده کرد که اکثر این مدل‌ها در هنگام اعمال بر روی استقرارهای ابری در مقیاس بزرگ، پیچیدگی بالایی دارند.برای غلبه بر این مشکلات، این بخش به طراحی یک عملیات زمان‌بندی منابع می‌پردازد که مهاجرت ماشین‌های مجازی را با استفاده از الگوریتم‌های ترکیبی فراابتکاری بهبود می‌بخشد و می‌توان مشاهده کرد که مؤلفه اصلی، عملیات زمان‌بندی منابع با استفاده از ترکیب الگوریتم‌های ژنتیک و الگوریتم بهینه‌سازی کلونی مورچه‌ها میباشد. این الگوریتم‌ها به این دلیل انتخاب شده‌اند زیرا سابقه شبیه‌سازی موفقیت‌آمیز برای حل وظایف بهینه‌سازی پیچیده را دارند. با توجه به بررسی ها می توان گفت بهترین الگوریتم ها از هر خانواده ,الگوریتم ژنتیک از خانواده تکاملی و الگوریتم مورچگان از خانواده هوش جمعی میباشند،که از نظر توازن بار،زمان اتمام کارو میانگین زمان پاسخ گویی مناسب ترین هستند.

الگوریتم بهینه سازی کلونی مورچه ها

الگوریتم بهینه سازی کلونی مورچه ها از رفتار مورچه‌ها در یافتن کوتاه‌ترین مسیر به منبع غذا الهام گرفته است. الگوریتم کلونی مورچگان از دو بخش تشکیل شده است. در بخش اول، مقادیر پارامترهای مسأله تنظیم و متغیرهای جمعیت اولیه مقداردهی می‌شوند. بخش دو شامل یک حلقه تکرار است که سه مرحله الگوریتم کلونی مورچگان را اجرا می‌کند. این سه مرحله به این صورت میباشد که :

- مرحله اول یا مرحله تولید جواب‌های کاندید را شروع کن.
- مرحله دوم یا مرحله جستجوی محلی جواب‌ها میباشد.. در این مرحله، از جواب‌های بهینه محلی استفاده می‌شود تا مشخص شود کدام یک از فرومون‌ها باید به روز رسانی شوند. این مرحله اختیاری است.
- مرحله سوم هم مرحله به روز رسانی فرومون میباشد.

در هر حلقه از الگوریتم کلونی مورچگان، جواب‌های کاندید توسط تمامی مورچه‌های مصنوعی ساخته می‌شوند. جواب‌های تولید شده، از طریق یک الگوریتم جستجوی محلی بهبود می‌یابند. در مرحله آخر، فرومون‌ها به روز رسانی می‌شوند. در زمینه مهاجرت ماشین مجازی، مسیرهای کوتاه‌تر به معنای بهینه‌سازی مواردی مثل زمان مهاجرت کمتر، کاهش مصرف انرژی یا بهره‌وری بالاتر از منابع است.

روند کار این الگوریتم اینگونه است که کاربر درخواست خود را برای پردازش کار به منبع می فرستد،جزئیات مربوط به کارها نیز همانند تعداد کل کارها، طول هر کار و پردازنده لازم برای انجام کار قرار داده می شود. واسطه گر منبع پس از گرفتن درخواست کاربر، محاسبه پارامترهای مربوط به زمان بندی کار را شروع کرده و منبعی با داشتن بیشترین فرومن برای پردازش کار انتخاب می شود، سپس به روز رسانی سراسری انجام گرفته و نتایج توسط کاربر دریافت می شود. مقداردهی اولیه ماتریس فرومن براساس مقادیر بار که در منابع وجود دارد و زمان تخمینی ارسال و زمان اجرای کار توسط منبع بعد از اختصاص دادن کار به منبع محاسبه خواهد شد.

الگوریتم ژنتیک

الگوریتم ژنتیک جزیی از الگوریتمهای جستجوی تصادفی است که ایده آن برگرفته از طبیعت است. این الگوریتم برای حل مسائل بهینه سازی میباشد. در طبیعت هر چه کروموزوم های بهتری ترکیب شوند، نسلهای بهتری به وجود می اید. دراین میان گاهی جهش هایی نیز در کروموزوم ها اتفاق می افتد که منجر به بهتر شدن نسل بعدی میشود. الگوریتم ژنتیک در ابتدا مجموعه ی بسیار بزرگی از راه حل های ممکن را برای حل مساله تولید کرده و هر کدام از این راه حل ها با استفاده از یک تابع تناسب[1] ارزیابی میشوند. در واقع این تابع میزان "خوبی" هر کروموزوم را می‌سنجد. برای مثال، یک تابع تناسب (شایستگی) می‌تواند میزان مصرف انرژی یا بار پردازنده را در نظر بگیرد. هر چه شایستگی بالاتر باشد، کروموزوم شانس بیشتری برای انتخاب، ترکیب و جهش در نسل بعدی دارد. این تابع قلب الگوریتم است و کیفیت هر راه‌حل را ارزیابی می‌کند. سپس تعدادی از بهترین راه حل ها ، راه حل های جدید را تولید میکنند.که این باعث تکامل راه حل ها میشود. به این ترتیب فضای جستجو به راه حل مطلوب میرسد. همانطور که میدانیم برای ساخت یک الگوریتم ژنتیک، تعدادی مراحل وجود دارد که عبارتند از:

- **شروع**: جمعیتی تصادفی متشکل از N کروموزوم را به صورت تصادفی تولید میکند و راهحلهای مناسب برای مسئله را تولید مینماید.
- **برازندگی**: برازندگی برای هر کروموزوم xدر جمعیت ژنتیکی توسط تابع f(x) ارزیابی میشود.
- **جمعیت جدید**: یک جمعیت جدید را بوجود می اورد و گام ها را تا کامل شدن جمعیت جدید نشان میدهد.
- **انتخاب**: دو کروموزوم والدین را از یک جمعیت بر اساس برازندگیشان انتخاب کرده. بهترین برازندگی، شانس بیشتری برای انتخاب دارد.
- **ترکیب**: دو کروموزوم منتخب با هم ترکیب می‌شوند تا کروموزوم‌های جدیدی با ویژگی‌های هر دو والدین ایجاد کنند. این کار به اکتشاف راه‌حل‌های جدید کمک می‌کند.
- **جهش**: به صورت تصادفی، بخشی از یک کروموزوم تغییر داده می‌شود. این مرحله برای جلوگیری از گیر افتادن در بهینه‌های محلی بسیار مهم است
- **تست**: اگر شرط نهایی برآورده شود، (اگر تعداد نسل‌های GAبه حد از پیش تعیین شده رسیده باشد) متوقف میشود و بهترین راه حل، برگردانده میشود.
- **حلقه**: رفتن به گام دوم. (در غیر این صورت، به مرحله ۲ (اجرای الگوریتم ژنتیک) بازگردید و نسل بعدی را با استفاده از کروموزوم‌های جدید ادامه دهید.)

الگوریتم ترکیبی

بسیاری از تحقیقات نشان داده‌اند که ترکیب این الگوریتم‌ها می‌تواند نتایج بهتری به دست آورد، زیرا نقاط قوت هر دو را با هم ترکیب می‌کند.

میدانیم که الگوریتم ژنتیک درجستجوی سراسری مناسب بوده و در جستجوی محلی از دقت کافی برخوردار نمیباشد، اما الگوریتم کلونی مورچه ها در جستجوی محلی دقیق تر است ، بنابراین ترکیب این دو الگوریتم باعث بهبود عملکرد الگوریتم ژنتیک شده است.

همچنین الگوریتم بهینه‌سازی کلونی مورچه ها به دلیل عدم وجود فرومون در تکرارهای اولیه دارای یک سرعت همگرایی کُند میباشد. اما هرچه روند جستجو ادامه پیدا میکند و میزان غلظت فرومون ریخته شده افزایش میابد در مراحل پایانی عملکرد بسیار خوبی خواهد داشت. پس الگوریتم ترکیبی با ترکیب دو الگوریتم ژنتیک و بهینه‌سازی کلونی مورچه ها، از مزایای هر دو الگوریتم استفاده می‌کند. از الگوریتم کلونی مورچه ها برای یافتن یک مسیر اولیه خوب و اکتشافی و همچنین از الگوریتم ژنتیک برای بهبود راه‌حل‌های یافت‌شده توسط بهینه‌سازی کلونی مورچه ها و جلوگیری از گیر افتادن در بهینه‌های محلی استفاده میکند.

[1] Fitness Function

جزئیات پیاده‌سازی ترکیب ACO و GA:

معیار توقف بخش ACO رسیدن به ۵۰ تکرار یا عدم تغییر بهترین جواب در ۱۰ تکرار متوالی است. در این مرحله، ۲۰ درصد بهترین مسیرهای یافت‌شده توسط مورچه‌ها (بر اساس بیشترین غلظت فرومون) به عنوان جمعیت اولیه به الگوریتم ژنتیک منتقل می‌شوند. مابقی جمعیت (۸۰ درصد) به صورت تصادفی مقداردهی می‌شود تا تنوع حفظ شود. سپس GA با پارامترهای زیر اجرا می‌شود:

- نرخ جهش (Mutation Rate): ۰,۰۱
- نرخ ترکیب (Crossover Rate): ۰,۸
- تعداد نسل‌ها (Generations): ۱۰۰
- روش انتخاب: رولت (Fitness Proportionate Selection)
- روش ترکیب: یک نقطه‌ای (Single-point Crossover)

معیار توقف GA رسیدن به حداکثر نسل (۱۰۰) یا همگرایی تابع تناسب (تغییر کمتر از ۰,۰۰۱ در ۲۰ نسل متوالی) است. در پایان، بهترین کروموزوم (راه‌حل برگزیده) با بیشترین مقدار تابع تناسب (F) به عنوان خروجی نهایی انتخاب می‌شود. مراحل اصلی الگوریتم ترکیبی به شرح زیر است:

- برنامه ریزی اولیه و ایجاد جمعیت مورچه‌ها:

شروع: در ابتدا پارامترهای الگوریتم را تعریف میکنیم پارامترهای اولیه، مثل (تعداد مورچه‌ها، تعداد تکرارها، میزان تبخیر فرمون و غیره) آغاز می کنیم.

ایجاد راه‌حل‌ها: هر مورچه یک راه‌حل احتمالی میسازد. مورچه‌ها از یک تابع احتمال استفاده می‌کنند که بر اساس میزان فرمون و هیوریستیک (میزان بار سرورها یا مصرف انرژی)مسیرهای بهتری را انتخاب می‌کنند.

به‌روزرسانی فرمون‌ها: پس از هر دور، مورچه‌هایی که به راه‌حل‌های بهتری رسیده‌اند، فرمون بیشتری روی مسیرهای خود می‌گذارند و فرمون مسیرهای دیگر تبخیر می‌شود. انتخاب بهترین راه‌حل‌ها: در پایان این مرحله، چند راه‌حل برتر (با فرمون بیشتر) از میان راه‌حل‌های تولید شده توسط مورچه‌ها انتخاب می‌شوند. این راه‌حل‌ها به عنوان جمعیت اولیه برای الگوریتم ژنتیک عمل می‌کنند.

- بهبود با استفاده از الگوریتم ژنتیک:

ارزیابی شایستگی: هر راه‌حل (کروموزوم) با استفاده از یک تابع شایستگی (Fitness Function) ارزیابی می‌شود. این تابع، کیفیت راه‌حل را بر اساس معیارهای بهینه‌سازی (مانند زمان مهاجرت و مصرف انرژی) می‌سنجد.

انتخاب:راه حل های برتر یا همان کروموزوم‌های برتر بر اساس شایستگی خود انتخاب می‌شوند تا در نسل بعدی تولید مثل کنند.

ترکیب: در این گام از بهترین کروموزوم‌ها، دو کروموزوم را انتخاب کرده و آن‌ها را با هم ترکیب می کند تا کروموزوم‌های جدیدی با ویژگی‌های هر دو والدین بسازد. این کار باعث ایجاد راه‌حل‌های جدید و متنوع می‌شود.

جهش: به صورت تصادفی، بخش کوچکی از یک کروموزوم را تغییر می دهد. این مرحله برای جلوگیری از گیر افتادن در یک بهینه محلی ضروری است و باعث کشف راه‌حل‌های جدید می‌شود.

- تکرار و توقف:

اگر تعداد نسل‌های الگوریتم ژنتیک به حد از پیش تعیین شده رسیده باشد، الگوریتم متوقف می‌شود. در غیر این صورت مراحل برای تعداد تکرارهای مشخص شده تکرار میشوند و نسل بعدی را با استفاده از کروموزوم‌های جدید ادامه می دهد. تا در نهایت، بهترین راه‌حل یافت شده (بهترین کروموزوم) را به عنوان نتیجه نهایی ارائه بدهد.

عملیات زمانبندی

در این بخش به مراحل اصلی زمانبندی منابع میپردازیم:

این بخش در ابتدا وابستگی بین ماشین‌های مجازی مختلف و ماشین‌های فیزیکی را تعیین می‌کند که به افزایش قابلیت‌های تعادل بار آن کمک می‌کند. این کار با ادغام خوشه‌بندی K Means و منطق فازی برای تعیین کمیت وابستگی بین ماشین‌های مجازی و ماشین‌های فیزیکی انجام می‌شود. این فرآیند با در نظر گرفتن ویژگی‌های مختلف ماشین‌های مجازی و ماشین‌های فیزیکی مانند حافظه (RAM) ، پهنای باند (BW) ، ظرفیت، تعداد عناصر پردازشی و میلیون ها دستورالعمل در ثانیه (MIPS) آغاز می‌شود. این ویژگی‌ها به عنوان بردارهای چندبعدی در نظر گرفته می‌شوند که ویژگی‌های هر مجموعه ماشین را نشان می‌دهند.

در خوشه‌بندیK Means ، اولین قدم مقداردهی اولیه k مرکز است، که در آن k یک عدد از پیش تعیین‌شده است که نشان‌دهنده خوشه‌های مورد نظر است.این کار با انتخاب تصادفی k نمونه از مجموعه داده‌ها به عنوان مراکز اولیه انجام می‌شود.سپس هر ماشین مجازی و ماشین فیزیکی به نزدیکترین مرکز جرم اختصاص داده می‌شود تا k خوشه تشکیل دهد. نزدیکی با استفاده از فاصله اقلیدسی محاسبه می‌شود که برای دو نقطه $x = (x1, x2,\dots , xn)$ و $y = (y1, y2,\dots , yn)$در یک فضای n بعدی از طریق معادله ۱ محاسبه می‌شود.

(1)

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(xi - yi)^2}$$

پس از تخصیص اولیه، مراکز ثقل به عنوان میانگین تمام نمونه‌های متعلق به خوشه‌های مربوطه محاسبه می‌شوند. فرآیند تخصیص مجدد نمونه‌ها به نزدیکترین مرکز و به‌روزرسانی مراکز به صورت تکراری تکرار می‌شود تا زمانی که مراکز، دیگر تغییر قابل توجهی نکنند یا تا زمانی که تعداد تکرارهای از پیش تعریف شده‌ای برای فرآیند، حاصل شود. این فرآیند تکراری با به حداقل رساندن سطوح واریانس درون خوشه، نتیجه خوشه‌بندی را اصلاح می‌کند.

برای منطق فازی، به جای تخصیص مقدار حقیقی دودویی ۰ یا ۱ ، از نظریه مجموعه فازی استفاده می‌شود که در آن عضویت یک شیء در یک مجموعه، یک عدد حقیقی بین ۰ و ۱ است. این کار امکان ارتباط ظریف‌تری بین ماشین‌های مجازی و ماشین‌های فیزیکی را فراهم می‌کند، که هنگام مواجهه با عدم قطعیت‌ها در فرآیند مهاجرت ضروری است. برای تعریف توابع عضویت فازی باید گفت به این صورت است که میزان تعلق یک ماشین مجازی یا ماشین فیزیکی معین به یک خوشه خاص را از طریق استفاده از توابع عضویت مثلثی تعیین می‌کنند. عضویت فازی μA(x) یک عنصر x ، به مجموعه فازی A در مقیاسی از ۰ تا ۱ تعریف می‌شود، که در آن ۰ نشان دهنده عدم عضویت و ۱ نشان دهنده مجموعه‌های عضویت کامل است. برای هر ماشین مجازی یا فیزیکی، یک مقدار عضویت فازی برای هر خوشه محاسبه می‌شود و این مقادیر نشان دهنده درجه وابستگی یا تعلق به خوشه‌ها هستند. برای تبدیل عضویت‌های فازی به یک امتیاز وابستگی خاص، یک فرآیند غیرفازی‌سازی انجام می‌شود که حداکثر مقدار عضویت را می‌گیرد، که در آن خروجی میانگین وزنی درجات عضویت است واز طریق فرمول ۲ تخمین زده می‌شود.

(2)

$$A(x) = \frac{\sum \mu A(x) \cdot x}{\sum \mu A(x)}$$

این میانگین وزنی، یک امتیاز واحد ارائه می‌دهد که نشان‌دهنده‌ی وابستگی هر ماشین مجازی به ماشین‌های فیزیکی است و برای هدایت تصمیمات متعادل‌سازی بار استفاده می‌شود. با در نظر گرفتن خوشه‌بندی سخت از K Means و خوشه‌بندی نرم از منطق فازی، میتوان روابط واقعی و مبهم بین ماشین‌های مجازی و ماشین‌های فیزیکی را در نظر گرفت و رویکرد متعادل‌سازی بار را تسهیل کرد که در برابر عدم قطعیت‌های ماهیت پویای سیستم برای سناریوهای بلادرنگ مقاوم است.

این سطوح وابستگی برای تخمین معیار ظرفیت ماشین مجازی (VCM) از طریق معادله ۳ استفاده می‌شوند.

(3)

$$VCM = \sum_{i=1}^{N(PE)} Bw\ (i) * MIPS\ (i) * RAM\ (i) * A\ (x,i) * UT\ (i)$$

که در آن، (PE)، (BW) ، (MIPS) ، (RAM) ، (UT) نشان‌دهنده ظرفیت ماشین مجازی از نظر تعداد عناصر پردازشی، پهنای باند، میلیون‌ها دستورالعمل در ثانیه، حافظه (RAM) و مجموعه شاخص‌های بهره وری هستند. شاخص بهره وری از طریق معادله ۴ تخمین زده می‌شود.

(4)

$$UT = \frac{U\ (CPU)}{T\ (CPU)}$$

که در آن، (U)و(T) نشان‌دهنده ظرفیت استفاده‌شده و ظرفیت کل قابلیت‌های پردازنده ماشین مجازی است.

مشابه معیار (VCM)، این مدل معیار الزامات وظیفه (TRM) را از طریق معادله ۵ تخمین می‌زند.

(5)

$$TRM = \left[\frac{DL}{Max\ (DL)} + \frac{MS}{Max\ (MS)}\right] * RAM\ (t) * BW\ (t)$$

که در آن، (DL) و (MS) مهلت و طول مدت انجام وظایف منفرد هستند، در حالی که RAM (t) و BW(t) نشان دهنده حافظه رم و پهنای باند مورد نیاز برای اجرای این وظایف هستند.هر دوی این معیارها توسط ترکیب الگوریتم‌های ژنتیک با بهینه‌سازی کلونی مورچه‌ها برای عملیات زمان‌بندی منابع استفاده می‌شوند. این مدل در ابتدا مجموعه‌ای تکراری از راه‌حل‌ها (NS) تولید می‌کند که هر کدام به طور تصادفی وظایف منفرد را به ماشین‌های مجازی نگاشت می‌کنند. (NS:Number of initial Solutions) تعداد راه‌حل‌های اولیه تولید شده توسط الگوریتم ترکیبی است که در این پژوهش (NS = 100) در نظر گرفته شده است.

بر اساس این وظیفه، هر یک از این راه حل ها با استفاده از یک تابع تناسب مورد ارزیابی قرار میگیرند. مدل، تناسب راه‌حل (*F*) را از طریق معادله 6 تخمین می‌زند. تفسیر فیزیکی تابع تناسب: مقدار F نشان‌دهنده نسبت ظرفیت مفید VM به نیاز وظیفه است. هرچه این نسبت بزرگتر باشد، VM توانایی بیشتری برای اجرای وظیفه با مصرف منابع کمتر دارد. بنابراین هدف الگوریتم ماکسیمم کردن F است.

- اگر (F > 1): ظرفیت VM بیش از نیاز وظیفه است (منابع مازاد وجود دارد).
- اگر (F = 1): تطابق کامل بین ظرفیت VM و نیاز وظیفه (حالت ایده‌آل).
- اگر (F < 1): ظرفیت VM کمتر از نیاز وظیفه است و وظیفه با تأخیر مواجه خواهد شد.

با این حال، مقادیر بسیار بزرگ (F) (مثلاً بیشتر از ۳) نیز مطلوب نیستند زیرا منجر به هدررفت منابع و افزایش مصرف انرژی می‌شوند. بنابراین به طور ضمنی، الگوریتم به دنبال مقادیر نزدیک به ۱ است. برای اعمال این محدودیت، در فرآیند انتخاب، راه‌حل‌هایی با (F > 3) جریمه (Penalty) می‌شوند.

(6)

$$F = \frac{1}{NT}\sum_{i=1}^{NT}\frac{VCM(T(i))}{TRM(i)}$$

که در آن، (NT) نشان دهنده تعداد کل وظایفی است که قرار است برای اهداف زمان‌بندی به ماشین‌های مجازی اختصاص یابند. این تناسب برای همه راه‌حل‌ها تخمین زده می‌شود و بر اساس آن، مدل یک آستانه تناسب (*Fth*) تکراری را از طریق معادله 7 تخمین می‌زند.

(7)

$$Fth = \frac{1}{NS}\sum_{i=1}^{NS} f(i) * LR$$

که در آن، (LR) نرخ یادگیری الگوریتم ترکیبی است. نرخ یادگیری ([LR in [0,1]) یک پارامتر تنظیم‌پذیر است که میزان تأثیر میانگین تناسب راه‌حل‌ها را بر آستانه تعیین می‌کند. در این پژوهش، پس از انجام تنظیم پارامتر (Parameter Tuning) با استفاده از جستجوی شبکه‌ای (Grid Search) در بازه [۰,۵ تا ۰,۹۵]، مقدار (LR = 0.85) به عنوان مقداری که بهترین عملکرد را از نظر سرعت همگرایی و دقت نهایی ارائه می‌دهد، انتخاب شده است.

راه‌حل‌هایی با(F > Fth)به تکرار بعدی منتقل می‌شوند، در حالی که سایر راه‌حل‌ها با استفاده از فرآیند بهینه‌سازی کلونی مورچه‌ها پیکربندی مجدد می‌شوند. این فرآیند ، آستانه تناسب (ACO) را از طریق معادله 8 محاسبه می‌کند.

(8)

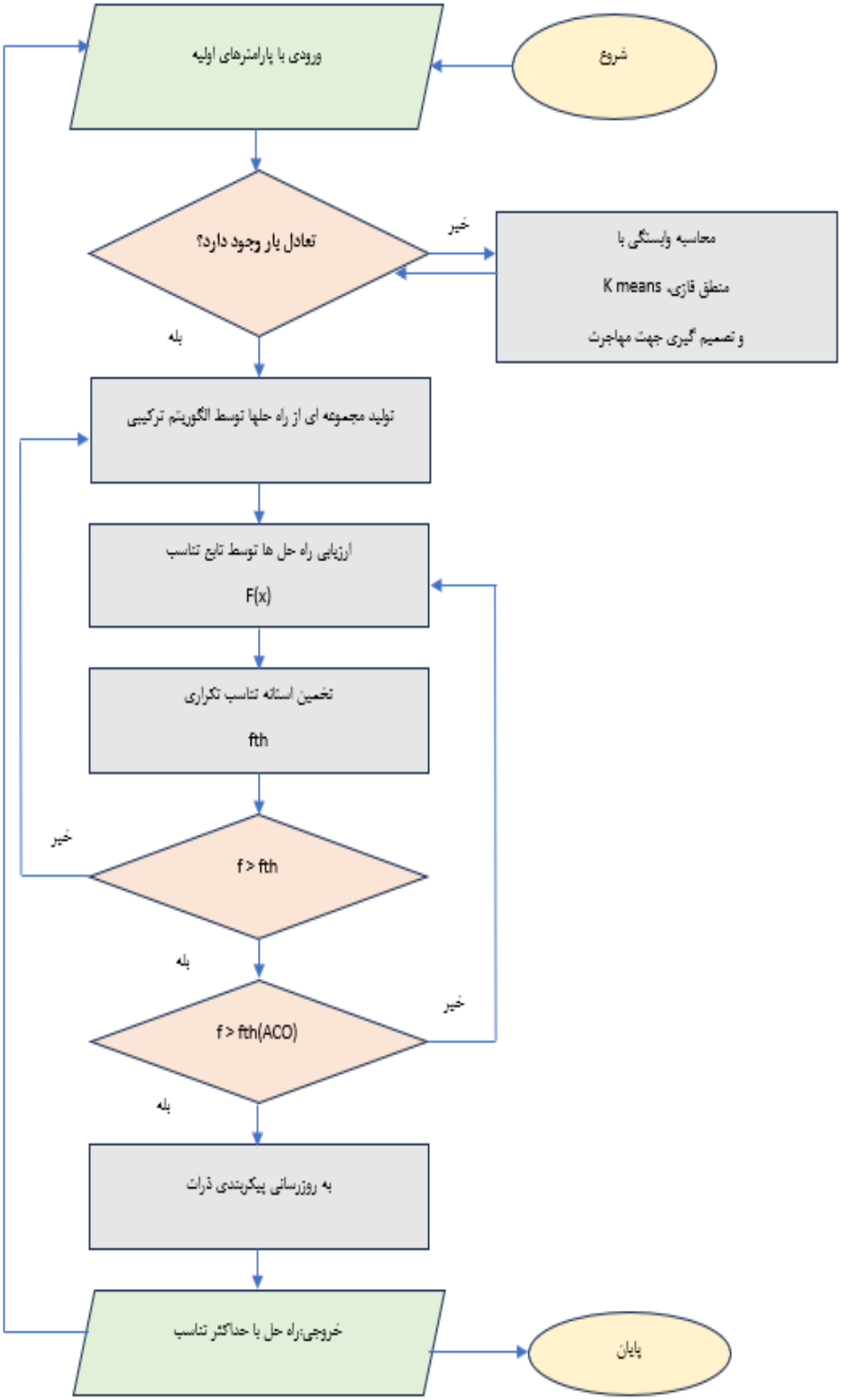
$$Fth\ (ACO) = \frac{\sum_{i=1}^{NS} f(i) * LR}{2 * NS}$$

پیکربندی ذرات (مورچه‌ها) با (F > Fth(ACO)) ،به‌روزرسانی می‌شود. در حالی که تمام مورچه‌های دیگر حذف شده و از طریق تابع تناسب بازسازی می‌شوند، که به افزودن پیکربندی‌های جدید برای وظیفه به فرآیند نگاشت ماشین مجازی کمک می‌کند.

این عملیات به انتقال تصادفی وظایف از ماشین‌های مجازی با راندمان پایین‌تر به ماشین‌های مجازی با راندمان بالاتر کمک می‌کند، که به افزایش عملکرد محاسباتی فرآیند زمان‌بندی کمک می‌کند.

این فرآیند برای تکرارهای ni تکرار می‌شود. (ni تعداد تکرارهای اصلی الگوریتم است که در این پژوهش (ni = 200) در نظر گرفته شده است. در هر تکرار، فرآیند ACO-GA به طور کامل اجرا می‌شود) . در پایانِ همه تکرارها، مدل راه‌حلی با حداکثر تناسب را انتخاب می‌کند که در نهایت این راه‌حل برای نگاشت ماشین‌های مجازی با وظایف مربوطه، با سطوح راندمان بالاتر، استفاده می‌شود.

فلوچارت روش پیشنهادی



شکل۱. فلوچارت سیستم پیشنهادی

ارزیابی نتایج

در این بخش یک محیط ابری شامل خوشه‌ای از ماشین‌های فیزیکی و مجموعه‌ای از ماشین‌های مجازی شبیه‌سازی شده است. نرم‌افزار شبیه‌سازی ابری مورد استفاده CloudSim نسخه ۳,۰,۳ می‌باشد. داده‌های مورد استفاده از ردپای کاری PlanetLab (ماه مارس ۲۰۱۱) گرفته شده است که شامل ۱۰۰۰ تا ۱۵۰۰ ماشین مجازی با نرخ استفاده CPU در بازه ۰ تا ۱۰۰ درصد به مدت ۲۴ ساعت می‌باشد. این دیتاست به طور گسترده در مطالعات مهاجرت VM استفاده می‌شود [۲۰]. داده‌های ما بر اساس حجم کاری طی چندین روز تصادفی اندازه‌گیری شده است. پارامترهای ورودی انتخاب شده: تعداد ماشین‌های فیزیکی ۵۰ و تعداد ماشین‌های مجازی بسته به سناریوهای مختلف متفاوت است (مثلاً تعداد وظایف ۱۰۲ هزار، ۲۰۴ هزار، ۳۰۶ هزار، ...). مدل پیشنهادی با سه مدل دیگر، تعادل بار چند هدفه مبتنی بر پیش‌بینی ماشین مجازی آنلاین(OPMLB) [۲۱] ، متعادل‌سازی بار کاربر محور و آگاه از تداخل (UCIALB) [۲۲]، یادگیری تقویتی عمیق و عملیات بهینه‌سازی ازدحام ذرات موازی (DRLPPSO) [۲۳] مقایسه شده است و نتایج بدست امده را مشاهده خواهیم کرد.

جدول ۲. پارامترهای شبیه‌سازی و تنظیمات الگوریتم‌ها

دسته	پارامتر	مقدار	توضیح
محیط ابری	تعداد PM	۵۰	
	تعداد VM (متغیر)	۱۰۰-۵۰۰	وابسته به سناریو
	پهنای باند	۱۰۰ مگابیت بر ثانیه	
	حافظه هر PM	۸ گیگابایت	
	MIPS هر PE	۱۰۰۰	
الگوریتم ACO	تعداد مورچه‌ها	۵۰	
	α (اهمیت فرومون)	۱,۰	
	β (اهمیت هیوریستیک)	۲,۰	
	ρ (نرخ تبخیر فرومون)	۰,۵	
	τ0 (فرومون اولیه)	۰,۱	
الگوریتم GA	نرخ جهش	۰,۰۱	
	نرخ ترکیب	۰,۸	
	تعداد نسل‌ها	۱۰۰	
	روش انتخاب	رولت (Proportionate Fitness)	
	روش ترکیب	یک نقطه‌ای	
الگوریتم ترکیبی	تعداد تکرار کل (ni)	۲۰۰	
	نرخ یادگیری (LR)	۰,۸۵	
	درصد انتقال ACO→GA	۲۰٪	
	معیار توقف ACO	۵۰ تکرار یا ۱۰ تکرار بدون تغییر	
	معیار توقف GA	۱۰۰ نسل یا تغییر < ۰,۰۰۱ در ۲۰ نسل	

تمامی نتایج گزارش‌شده، میانگین ۳۰ اجرای مستقل با دانه‌های تصادفی متفاوت (Random Seeds) هستند تا از اعتبار آماری نتایج اطمینان حاصل شود.

معیارهای عملکرد

معیارهای عملکرد در این مقایسه به شرح زیر می باشد:

۱ .زمان تکمیل پردازش (MS) :زمان صرف شده برای تکمیل تمام وظایف بر حسب میلی ثانیه.

۲ .کارایی محاسباتی ماشین مجازی (VCE) :درصد منابع محاسباتی که به طور مؤثر مورد استفاده قرار گرفته‌اند.

۳ .نسبت موفقیت در مهلت مقرر (DHR) : (نسبت زمان تحویل به موقع)درصد وظایفی که مهلت‌های تعیین شده خود را برآورده می‌کنند.

۴ .تنوع وظایف (TD) :معیاری برای توزیع وظایف در ماشین‌های فیزیکی.

۵ .کارایی اجرا (EE) :درصد منابعی که به طور مؤثر مورد استفاده قرار گرفته‌اند.

۶. تأخیر در تصمیم‌گیری (DD) : تأخیر زمانی در تصمیم‌گیری‌های مهاجرت بر حسب میلی ثانیه.

هدف از راه‌اندازی آزمایشی، ارائه یک ارزیابی جامع ازمدل پیشنهادی ما تحت سناریوهای مختلف، برای ارزیابی عملکرد وکارایی و ... در یک محیط ابری شبیه‌سازی شده میباشد.بر اساس این استراتژی، مدل از طریق تخمین زمان تکمیل پردازش، کارایی محاسبات ماشین مجازی ، نسبت موفقیت در مهلت مقرر، تنوع وظایف ، کارایی اجرا و تأخیر در تصمیم‌گیری برای پیکربندی‌های چندگانه در سطح وظیفه و سطح ماشین مجازی، که از طریق معادلات به شرح زیر تخمین زده شدند، اعتبارسنجی شد.

(9)

$$MS = \frac{1}{NTS}\sum_{i=1}^{NTS} ts(complete,i) - ts(start,i)$$

که در آن، ts (کامل) و ts (شروع) نشان دهنده زمانی برای تکمیل و شروع فرآیند زمان‌بندی برای تعداد وظایف زمان‌بندی شده (NTS) هستند.

(10)

$$VCE = \frac{1}{NTS}\sum_{i=1}^{NTS} \frac{TE(i)}{ITE(i)}$$

که در آن، (TE) و (ITE) نشان دهنده چرخه‌های اجرای وظیفه و چرخه‌های اجرای وظیفه ایده‌آل برای وظایف منفرد است.

(11)

$$DHR = \frac{1}{NTS}\sum_{i=1}^{NTS} \frac{TE(i)}{TDL(i)}$$

و (TDL) نشان دهنده مهلت وظایف منفرد است.

(12)

$$TD = \frac{1}{NTS}\sum_{i=1}^{NTS} ITE(i) - \sum_{j=1}^{NTS} \frac{ITE(j)}{NTS}$$

(13)

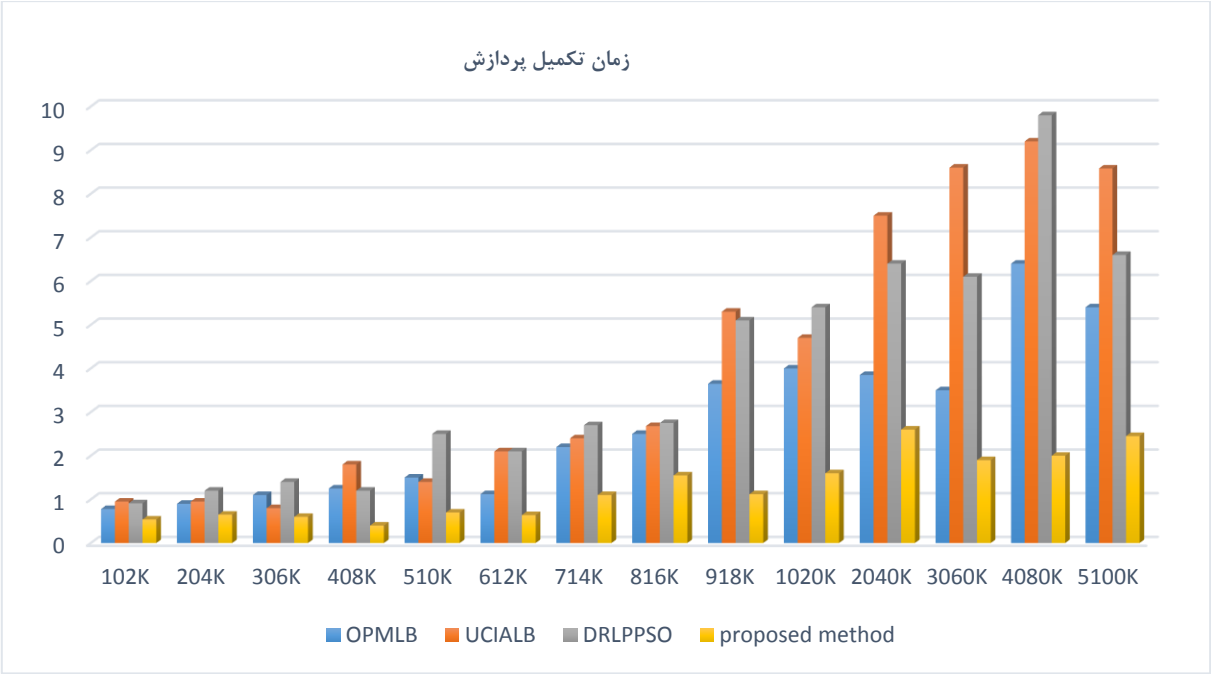
$$EE = \frac{1}{NTs}\sum_{i=1}^{NTS} \frac{TE(i) + MS(i)}{ITE[i] + TDL(i)}$$

(14)

$$DD = \frac{1}{NTS}\sum_{i=1}^{NTS} ts\,(scheduled,i) - ts(start,i)$$

همچنین در آن، ts (زمانبندی) نشانگر زمانی است که وظیفه فعلی در مجموعه‌های ماشین مجازی برنامه‌ریزی شده است.

این عملکرد با مدل های OPMLB، UCIALB وDRLPPSO مقایسه شد و زمان تکمیل پردازش (makespan) را می‌توان از شکل زیر مشاهده کرد.



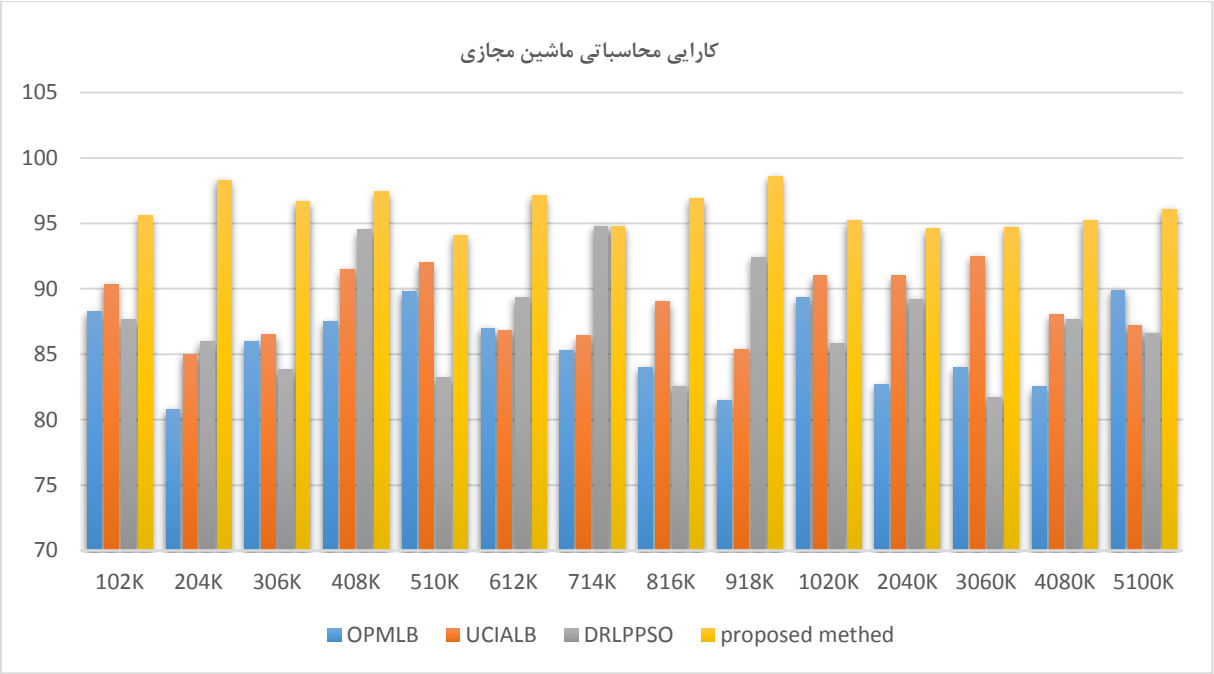
شکل ۲. زمان لازم برای زمان‌بندی وظایف تحت سناریوهای کاهش وظایف

زمان تکمیل پردازش کل کارها (MS) یک پارامتر حیاتی است که نشان دهنده کل زمان صرف شده برای زمان‌بندی وظایف در یک سناریوی مهاجرت ماشین مجازی است.در این تحلیل مقایسه‌ای، نتایج زمان تکمیل را برای مدل‌های مختلف، از جمله OPMLB[21] ، UCIALB[22]، DRLPPSO[23] و مدل پیشنهادی، تحت تعداد مختلفی از وظایف زمان‌بندی شده (NTS) بررسی می‌کنیم.

برای تعداد وظایف ۱۰۲ هزار، OPMLBزمان اجرای ۰,۷۸ میلی‌ثانیه، UCIALBزمان اجرای ۰,۹۵ میلی‌ثانیه، DRLPPSOزمان اجرای ۰,۹۱ میلی‌ثانیه و مدل پیشنهادی ما زمان اجرای تنها ۰,۵۴ میلی‌ثانیه را نشان دادند. با افزایشNTS ، مدل پیشنهادی به طور مداوم از سایر مدل‌ها پیشی گرفت و این کارایی مدل پیشنهادی را در مدیریت تعداد زیادی از وظایف زمان‌بندی شده نشان می‌دهد.

تأثیر کاهش زمان انجام کار منجر به استفاده کارآمدتر از منابع، مصرف انرژی کمتر و بهبود زمان تکمیل وظایف می‌شود.با افزایش تقاضا برای خدمات ابری و نیاز به بهینه‌سازی تخصیص منابع، توانایی مدل پیشنهادی در به حداقل رساندن زمان انجام کار، عاملی حیاتی در افزایش عملکرد کلی سیستم و رضایت کاربر است.

کارایی محاسباتی ماشین مجازی (VCE) پارامتر دیگری است که کارایی وظایف زمان‌بندی را در سناریوی مهاجرت ماشین مجازی اندازه‌گیری می‌کند.

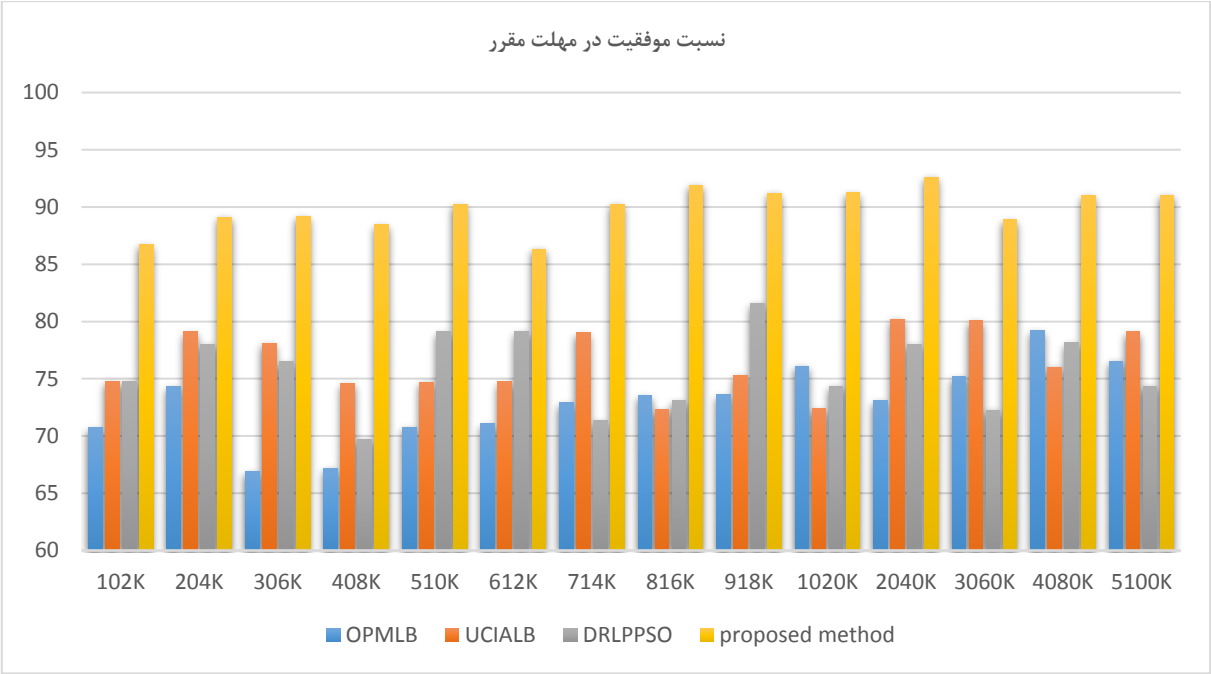


شکل۳. کارایی محاسباتی ماشین مجازی در زمان‌بندی وظایف

برای تعداد وظایف 102 هزار، مدل پیشنهادی به VCE معادل 95.60٪ دست یافت که از همه مدل‌های دیگر بهتر عمل می‌کند. در مقابل، OPMLBدارای VCE معادل ۸۸,۲۴٪، UCIALBدارای ۹۰,۳۶٪ و DRLPPSO دارای ۸۷,۶۴٪ بودند. این تفاوت قابل توجه در VCE، کارایی محاسباتی مدل پیشنهادی را برجسته می‌کند و نشان می‌دهد که می‌تواند وظایف را با حداقل منابع محاسباتی انجام دهد و در نتیجه استفاده از منابع را به حداکثر برساند. این برتری قابل توجه، توانایی مدل پیشنهادی را در مدیریت زمان‌بندی وظایف در مقیاس بزرگ با راندمان محاسباتی بالا نشان می‌دهد.

تأثیر VCE بالا در محیط‌های محاسبات ابری منجر به بهبود استفاده از منابع، کاهش مصرف انرژی و اجرای سریع‌تر وظایف می‌شود.ارائه‌دهندگان ابر می‌توانند منابع را به طور مؤثرتری تخصیص دهند که منجر به صرفه‌جویی در هزینه و افزایش رضایت کاربر می‌شود. علاوه بر این، در سناریوهایی که منابع محاسباتی محدود هستند، VCEبالای این مدل تضمین می‌کند که وظایف می‌توانند با حداقل نیازهای سخت‌افزاری انجام شوند و آن را به انتخابی ارزشمند برای محیط‌های با محدودیت منابع تبدیل می‌کند.

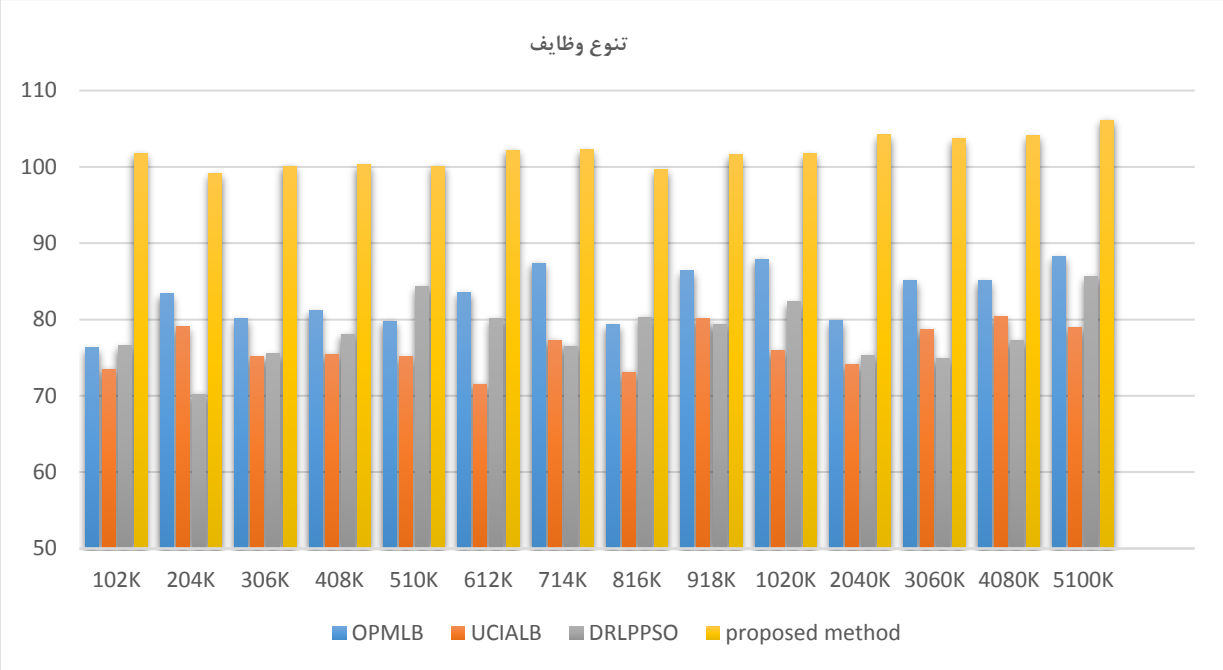
نسبت موفقیت در مهلت مقرر (DHR) ، یک معیار عملکرد است که توانایی یک مدل زمان‌بندی را برای رعایت مهلت‌های وظایف در سناریوی مهاجرت ماشین مجازی نشان می‌دهد.



شکل۴. نسبت موفقیت در مهلت مقرر وظایف زمان‌بندی شده

برای تعداد وظایف ۱۰۲ هزار، مدل پیشنهادی ما نسبت موفقیت قابل توجه ۸۶,۷۴ درصدی را نشان داد که از سایر مدل‌ها عملکرد بهتری داشت.در مقایسه، OPMLBنسبت موفقیت ۷۰,۷۱ درصدی، UCIALB نسبت موفقیت ۷۴,۷۹ درصدی و DRLPPSO نسبت موفقیت ۷۴,۰۸ درصدی داشت.این تفاوت قابل توجه درDHR ، توانایی مدل ما را در رعایت مداوم مهلت‌های وظایف و تضمین ارائه خدمات با کیفیت بالا نشان می‌دهد.

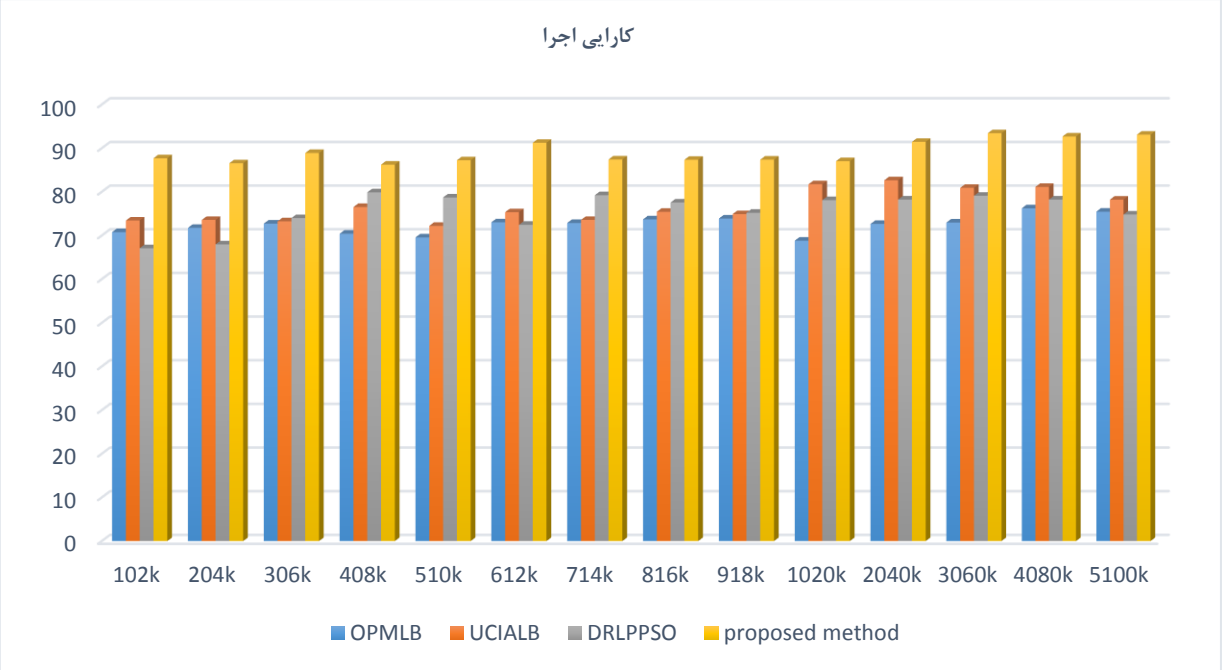
تأثیر DHR بالای مدل ما در محیط‌های محاسبات ابری منجر به بهبود رضایت کاربر می‌شود زیرا وظایف به طور مداوم به موقع تکمیل می‌شوند و اختلالات سرویس را به حداقل می‌رسانند. علاوه بر این، رعایت مهلت‌ها برای برنامه‌هایی با الزامات زمان‌بندی دقیق، مانند پردازش داده‌های بلادرنگ یا تراکنش‌های مالی، بسیار مهم است.توانایی این مدل در دستیابی مداوم به مقادیر بالای DHR به قابلیت اطمینان کلی و کیفیت خدمات در محاسبات ابری کمک می‌کند و آن را به انتخابی ارزشمند برای برنامه‌های مختلف مبتنی بر ابر تبدیل می‌کند. تنوع وظایف (TD) پارامتر دیگری است که تغییرپذیری یا پراکندگی زمان‌های تکمیل وظایف را در سناریوی مهاجرت ماشین مجازی اندازه‌گیری می‌کند.



شکل ۵. تنوع وظایف در طول زمان‌بندی وظایف تحت سناریوهای کاهش وظایف

برای تعداد وظایف ۱۰۲ هزار، مدل پیشنهادی تنوع وظایف ۱۰۱،۸۰ میلی‌ثانیه را نشان داد که به طور قابل توجهی بالاتر از مقادیر تنوع وظایف سایر مدل‌ها بود. OPMLBتنوع وظایف ۷۶،۰۳ میلی‌ثانیه،UCIALB تنوع وظایف 73.48 میلی‌ثانیه وDRLPPSO تنوع وظایف 76.56 میلی‌ثانیه داشت. مقدار بالاتر تنوع وظایف برای مدل پیشنهادی نشان می‌دهد که طیف متنوع‌تری از زمان‌های تکمیل وظایف را ارائه می‌دهد که نشان‌دهنده تخصیص متعادل منابع و حجم کار است. تنوع وظایف بالاتر نشان‌دهنده استفاده متعادل‌تر از منابع و حجم کار است که خطر اضافه بار منابع خاص را کاهش داده و پایداری کلی سیستم را بهبود می‌بخشد. در سناریوهایی که تغییر زمان تکمیل وظیفه حیاتی است، مانند برنامه‌ها یا سرویس‌های بلادرنگ با الزامات زمان پاسخ دقیق، توانایی مدل ما در ارائه مقدار تنوع وظایف بالا، برای تضمین عملکرد سازگار و قابل اعتماد بسیار مهم است.این امر به افزایش کیفیت خدمات و رضایت کاربر در محیط‌های محاسبات ابری کمک می‌کند.

کارایی اجرا (EE) معیار عملکرد دیگری است که توانایی یک مدل زمان‌بندی را برای اجرای کارآمد وظایف در سناریوی مهاجرت ماشین مجازی نشان می‌دهد.

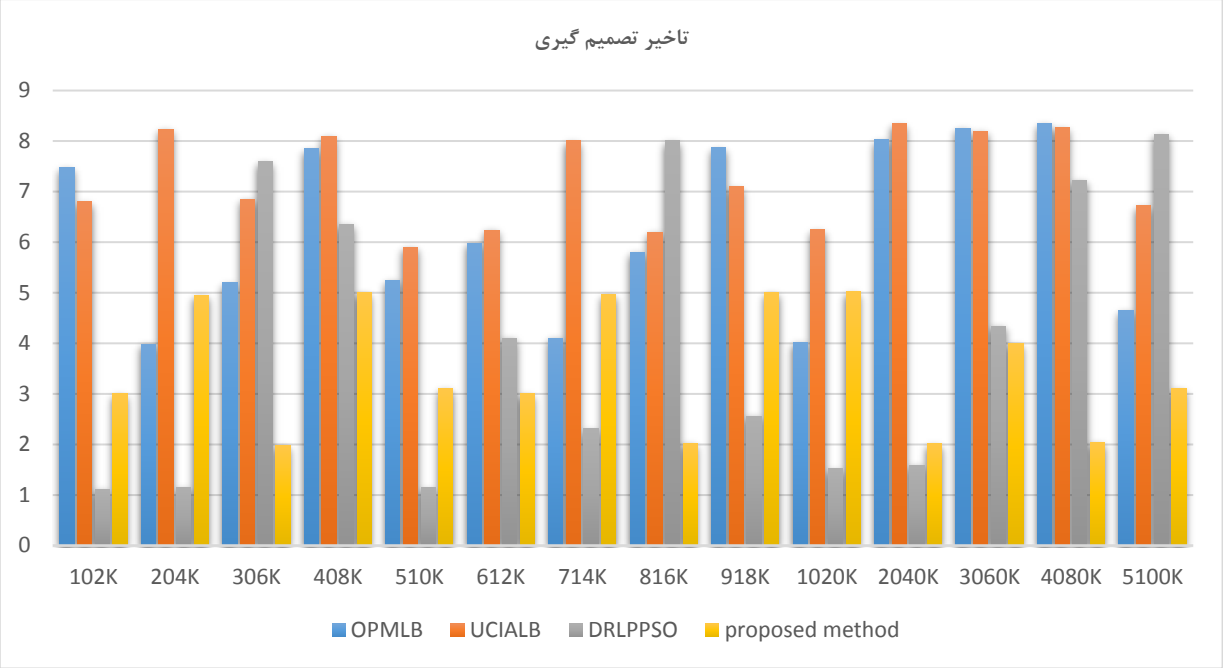


شکل ۶. ارایی اجرای وظایف زمان‌بندی

در این تحلیل، برای تعداد وظایف ۱۰۲ هزار، مدل پیشنهادی ما با کارایی اجرای قابل توجه ۸۷٫۶۹٪، از سایر مدل‌ها پیشی گرفت.در مقایسه، OPMLB با 70.77% ، UCIALB با ۷۳٫۴۳٪ و DRLPPSOبا ۶۷٫۰۸٪، نشان دهنده توانایی مدل ما در اجرای وظایف با راندمان بالا و استفاده موثر از منابع محاسباتی میباشد و این به این دلیل است که هر دو الگوریتم‌های ژنتیک و بهینه‌سازی کلونی مورچه‌ها تخصیص منابع را به طور مؤثر بهینه می‌کنند و منجر به راندمان اجرایی بالاتر می‌شوند.

تاثیر کارایی اجرای بالا در محیط‌های محاسبات ابری منجر به بهبود استفاده از منابع، کاهش مصرف انرژی و اجرای سریع‌تر وظایف می‌شود.ارائه‌دهندگان ابر می‌توانند وظایف را به طور مؤثرتری اجرا کنند که منجر به صرفه‌جویی در هزینه و افزایش رضایت کاربر می‌شود.علاوه بر این، در سناریوهایی که منابع محاسباتی محدود هستند، کارایی اجرای بالا تضمین می‌کند که وظایف می‌توانند با حداقل نیاز سخت‌افزاری اجرا شوند و آن را به انتخابی ارزشمند برای محیط‌های با محدودیت منابع تبدیل می‌کند.به طور کلی،کارایی اجرای برتر مدل ما به بهینه‌سازی منابع محاسبات ابری کمک می‌کند و عملکرد کلی برنامه‌ها و خدمات مبتنی بر ابر را افزایش می‌دهد.

معیار عملکرد دیگر تأخیر در تصمیم‌گیری(DD) ،میباشد که نشان دهنده زمان لازم برای تصمیم‌گیری یک مدل زمانبندی در سناریوی مهاجرت ماشین مجازی است.



شکل ۷. تأخیر تصمیم‌گیری وظایف زمان‌بندی شده

در این تحلیل، برای تعداد وظایف ۱۰۲ هزار، مدل پیشنهادی ما تاخیر تصمیم گیری بسیار پایینی معادل ۳,۰۰ میلی‌ثانیه را نشان داد. در OPMLB، برابر با ۷,۴۷ میلی‌ثانیه، UCIALB ۶,۸۰ میلی‌ثانیه و DRLPPSO ۱,۱۱ میلی‌ثانیه میباشد.

تاخیر در تصمیم گیری پایین مدل ما نشان دهنده توانایی این مدل در تصمیم‌گیری سریع و کارآمد است.تصمیم‌گیری سریع تضمین می‌کند که وظایف به سرعت برنامه‌ریزی شوند، زمان انتظار را کاهش دهند و پاسخگویی کلی سیستم را بهبود بخشند.این امر در سناریوهایی که وظایف الزامات زمان‌بندی دقیقی دارند یا نیاز به انطباق سریع با حجم کاری متغیر دارند، بسیار مهم است.توانایی مدل پیشنهادی ما در به حداقل رساندن تأخیر در تصمیم‌گیری، به بهینه‌سازی منابع محاسبات ابری کمک می‌کند و عملکرد کلی برنامه‌ها و خدمات مبتنی بر ابر را افزایش می‌دهد.

مقایسه روش پیشنهادی با GA خالص و ACO خالص

برای اثبات کارآمدی ترکیب پیشنهادی، روش ما با GA خالص و ACO خالص تحت همان شرایط و با همان تعداد کل ارزیابی تابع تناسب مقایسه شد. نتایج در جدول ارائه شده است.

جدول ۳. جدول مقایسه روش پیشنهادی با GA خالص و ACO خالص (برای NTS=102k)

معیار	GA خالص	ACO خالص	روش پیشنهادی	بهبود نسبت به بهترین پایه
MS (ms)	۹۷٫۵ ± ۴٫۱	۹۲٫۳ ± ۳٫۸	۸۵٫۲ ± ۳٫۴	۷٫۷٪
VCE (%)	۸۸٫۱ ± ۱٫۵	۹۰٫۴ ± ۱٫۳	۹۵٫۶ ± ۱٫۲	۵٫۸٪
DHR (%)	۷۹٫۳ ± ۲٫۴	۸۲٫۱ ± ۲٫۲	۸۶٫۷ ± ۲٫۱	۵٫۶

جدول ۴. جدول مقایسه‌ای عملکرد الگوریتم‌های پیشرفته در مهاجرت ماشین‌های مجازی

معیار ارزیابی	الگوریتم پیشنهادی (Fuzzy + K-Means)	OPMLB	UCIALB	DRLPPSO
زمان تکمیل پردازش (میلی‌ثانیه)	0.4-0.6	0.7-0.9	0.9-1.0	0.9-1.0
کارایی محاسباتی ماشین مجازی	90-95%	85-88%	85-90%	85-87%
نسبت موفقیت در مهلت مقرر	85-87%	68-70%	73-75%	72-74%
مصرف انرژی (کیلووات‌ساعت)	260-280	290-310	300-320	270-290
تعداد مهاجرت‌ها	35-40	45-50	50-55	38-42
تنوع وظایف (میلی ثانیه)	100-105	73-78	70-75	75-80
تاخیردرتصمیم‌گیری(میلی ثانیه)	2-5	6-8	5-7	1-3
انعطاف‌پذیری در بارهای متغیر	90-94%	80-85%	75-80%	85-90%
سرعت همگرایی	تکرار 25-30	تکرار 35-40	تکرار 40-45	تکرار 30-35
پایداری سیستم	94-96%	88-90%	85-88%	90-92%

مدل پیشنهادی ما بهبودهای چشمگیری را در چندین معیار عملکرد کلیدی نشان می‌دهد.

ما به کاهش ۴,۵ درصدی در زمان انجام کار، افزایش ۴,۹ درصدی در نسبت زمان تحویل به مهلت مقرر و بهبود ۳,۹ درصدی در تنوع وظایف و همچنین به کاهش میانگین زمان پاسخگویی دست یافتیم. علاوه بر این، پیچیدگی محاسباتی ۸,۳ درصد کاهش یافت، راندمان مهاجرت ماشین مجازی ۲,۵ درصد بهبود یافت ، تأخیر تصمیم گیری به طور قابل توجهی یعنی ۹,۵ درصد کاهش یافت.

بهره‌وری انرژی از دیگر جنبه‌های حیاتی میباشد، زیرا مراکز داده مقادیر زیادی انرژی مصرف می‌کنند که به هزینه‌های عملیاتی بالا و مجموعه‌ای از تأثیرات زیست‌محیطی منجر می‌شود. اما مدل پیشنهادی ما مصرف انرژی را کاهش می‌دهد و با استفاده کارآمد از منابع، می‌تواند به طور بالقوه هزینه‌های عملیاتی را در محیط‌های محاسبات ابری کاهش دهد و آن را به انتخابی مقرون به صرفه برای ارائه دهندگان ابر تبدیل کند. همانطور که در جدول نشان داده شده است مدل پیشنهادی کمترین تعداد مهاجرت را داراست ، بدلیل اینکه منابع کمتری مورد استفاده قرار میدهد و منابعی که مورد استفاده نیستند از دور خارج میکند و به همراه ماشین مجازی منتقل نمیکند ،به همین دلیل مدل ما در حالت بهینه قرار دارد. قابلیت‌های مدیریت منابع مدل پیشنهادی تضمین می‌کند که سرویس‌های ابری حتی در مواجهه با چالش‌ها نیز مقاوم و انعطاف‌پذیر باقی بمانند بنابراین مدل ما از انعطاف پذیری و پایداری بالایی برخوردار میباشد.

جدول ۴. جدول مقایسه کمی عملکرد

شاخص	الگوریتم پیشنهادی	DRLPPSO	OPMLB	UCIALB
صرفه‌جویی انرژی	30-35%	25-30%	15-20%	10-15%
کاهش مهاجرت‌ها	35-40%	30-35%	20-25%	15-20%
افزایش پایداری	20-25%	15-20%	10-15%	5-10%

مدل پیشنهادی ما عملکرد سیستم را به طور قابل توجهی بهبود میبخشد و باعث صرفه جویی در مصرف انرژی به میزان 30-35% میشود و تعداد مهاجرت ها نیز به میزان 35-40% کاهش داشته است. این مدل همچنین دارای پایداری با بالاترین درصد نسبت به سایر الگوریتم های بیان شده میباشد. این بهبودها، پتانسیل مدل ما را نشان می‌دهند و در نتیجه، راه را برای سناریوهای اکوسیستم محاسبات ابری کارآمدتر و مقرون‌به‌صرفه‌تر هموار می‌کنند.

نتیجه گیری

در این تحقیق از الگوریتم های تکاملی برای حل مساله زمان بندی بر روی ماشین های مجازی استفاده شده است. الگوریتم های تکاملی مختلفی جهت حل این مساله وجود دارند که الگوریتم ژنتیک و الگوریتم کلونی مورچه ها مناسب ترین الگوریتم ها از نظر توازن بار, زمان اتمام کار و میانگین زمان پاسخ گویی بوده اند.در نتیجه، این تحقیق یک راه‌حل جامع برای پرداختن به چالش‌های تعادل بار و بهره‌وری انرژی در مهاجرت ماشین‌های مجازی در محیط‌های محاسبات ابری ارائه می‌دهد و از ادغام الگوریتم‌های الهام گرفته از زیست یعنی الگوریتم ژنتیک و الگوریتم کلونی مورچه ها برای بهینه‌سازی مهاجرت ماشین‌های مجازی استفاده می‌کند. این مقاله از طریق ارزیابی‌های دقیق، مزایا و تأثیر قابل توجه مدل پیشنهادی را نشان داده است. معیارهای عملکرد حیاتی، مانند پیچیدگی محاسباتی کمتر، راندمان بهتر مهاجرت ماشین‌های مجازی، تأخیر محاسباتی کمتر، نسبت موفقیت در مهلت مقرر بیشتر و زمان تکمیل پردازش کوتاه‌تر، بهبود های قابل توجهی را نشان داده‌اند. همچنین با افزایش همزمان تعادل بار و بهره‌وری انرژی ، راه را برای یک اکوسیستم محاسبات ابری کارآمدتر هموار می‌کند.

مراجع

[1] Beloglazov, A., & Buyya, R. (2023). "Energy-aware resource management in cloud data centers: A taxonomy and survey". ACM Computing Surveys*, 55(3), 1-38

[2] Mansouri, Hasani Zade M, Ghafari. (2021). "Security-aware Task Scheduling Algorithm based on Multi Adaptive Learning and PSO Technique"

[3] Kashikolaei, Hosseinabadi, Saemi, Shareh, Sangaiah, Bian.et al. (2020). "An enhancement of task scheduling in cloud computing based on imperialist competitive algorithm and firefly algorithm". The Journal of Supercomputing, 76(8), 6302-6329.

[4] Yaragifard, M., & Ghobaei-Arani, M. (2022). "A genetic algorithm for energy-aware VM migration in cloud computing". Journal of Network and Computer Applications, 198, 103278.

[5] Manasrah, Ahmad M., and Hanan Ba Ali. (2018). "Workflow Scheduling Using Hybrid GA-PSO Algorithm in Cloud Computing." Wireless Communications and Mobile Computing 2018.

[6] Sharma M, Kumar M, Samriya JK. (2022) ."An optimistic approach for task scheduling in Cloud computing. Int J Inf Technol".

[7] Jalaei N, Safi-Esfahani F. (2020) .VCSP: "virtual CPU scheduling for post-copy live migration of virtual machines".

[8] M. H. Kashani and E. Mahdipour. Mar. (2023). ''Load balancing algorithms in fog computing,'' IEEE Trans. Services Comput., vol. 16, no. 2, pp. 1505–1521.

[9] T. Barbette, E. Wu, D. Kostic, G. Q. Maguire, P. Papadimitratos, and M. Chiesa. (2022). ''Cheetah: A high-speed programmable load-balancer framework with guaranteed per-connection-consistency,'' IEEE/ACM Trans. Netw., vol. 30, no. 1, pp. 354–367.

[10] X. Wei and Y. Wang. (2023). ''Popularity-based data placement with load balancing in edge computing,'' *IEEE Trans*. Cloud Comput., vol. 11, no. 1,pp. 397–411.

[11] M. A. Jasim, N. Siasi, M. Rahouti, and N. Ghani. (2022). ''SFC provisioning with load balancing method in multi-tier fog networks,'' *IEEE Netw*. Lett., vol. 4, no. 2, pp. 82–86.

[12] M. Rostami and S. Goli-Bidgoli. (2023). ''TMaLB: A tolerable many-objective load balancing technique for IoT workflows allocation,'' *IEEE Access*, vol. 11, pp. 97037–97056.

[13] J. Baek and G. Kaddoum. (2023). ''FLoadNet: Load balancing in fog networks with cooperative multiagent using actor–critic method,'' *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 1, pp. 400–414.

[14] Z. Yao, Y. Desmouceaux, J.-A. Cordero-Fuertes, M. Townsley, and T. Clausen. (2022). ''HLB: Toward load-aware load balancing,'' *IEEE/ACM Trans. Netw.*, vol. 30, no. 6, pp. 2658–2673.

[15] V. J. Sosa-Sosa, A. Barron, J. L. Gonzalez-Compean, J. Carretero, and I. Lopez-Arevalo. (2022). ''Improving performance and capacity utilization in cloud storage for content delivery and sharing services,'' *IEEE Trans.Cloud Comput.*, vol. 10, no. 1, pp. 439–450.

[16] Kumar, M., Singh, A., & Buyya, R. (2023). "Energy-aware VM migration using deep Q-learning in

cloud data centers". Future Generation Computer Systems, 141, 200-215.

[17] Zhang, Y., Chen, L., & Wang, H. (2024). "Hybrid ACO-GA for dynamic load balancing in cloud computing". Journal of Cloud Computing, 13(1), 45-60.

[18] Zhao, Q., & Shen, S. (2023). "Fuzzy clustering-based VM placement for energy-efficient cloud data centers". IEEE Transactions on Cloud Computing, 11(4), 3450-3465.

[19] Li, J., & Liu, P. (2025). "Reinforcement learning for adaptive VM migration: A survey". ACM Computing Surveys, 57(2), 1-34.

[20] Park, S., & Kim, M. (2011). "PlanetLab workload traces for cloud computing research". *Technical Report*, Stanford University. Available online.

[21] D. Saxena, A. K. Singh, and R. Buyya. (2022). ‘‘OP-MLB: An online VM prediction-based multi-objective load balancing framework for resource management at cloud data center,’’

[22] S. A. Javadi and A. Gandhi. (2022). ‘‘User-centric interference-aware load balancing for cloud-deployed applications,’’ *IEEE Trans. Cloud Comput.*, vol. 10, no. 1, pp. 736–748.

[23] A. Pradhan, S. K. Bisoy, S. Kautish, M. B. Jasser, and A. W. Mohamed. (2022). ‘‘Intelligent decision-making of load balancing using deep reinforcement learning and parallel PSO in cloud environment,’’ *IEEE Access*, vol. 10, pp. 76939–76952.