Journal of Soft Computing and Information Technology (JSCIT)
Babol Noshirvani University of Technology, Babol, Iran
Journal Homepage: jscit.nit.ac.ir
DOI: 10.22034/jscit.2026.560929.2194
Volume 15, Number 2, Summer 2026, pp. 48-63
Received: 11/22/2025, Revised: 05/13/2026, Accepted: 05/18/2026

A Multi-Stage Hybrid Optimization Model Based on IJFSA for Load Balancing and Performance Enhancement in Fog–Cloud Environments

Sakine Sohrabi[1], Reza Mohammadi [2*], Mehdi Sakhaei-nia [3], Mohammad Nassiri [4]
1,2*,3,4- Computer Engineering Department, Bu-Ali Sina University, Hamedan, Iran.

[1]s.sohrabi@eng.basu.ac.ir, [2*] r.mohammadi@basu.ac.ir,[3] sakhaei@basu.ac.ir, and [4] m.nassiri@basu.ac.ir

Corresponding author's address: Reza Mohammadi, Faculty of Engineering, Bu Ali Sina University, Hamedan, Iran.

Abstract- In recent years, emerging technologies have transformed human interaction with digital systems. The integration of the Internet of Things (IoT) with cloud infrastructures has enabled intelligent and automated services, particularly in healthcare. The growing number of networked sensors generates massive volumes of real-time data, increasing the demand for fast processing. Reliance on remote data centers, however, can lead to latency and reduced efficiency in time-sensitive systems. In this context, fog computing, as an intermediate layer between edge devices and the cloud, provides an effective solution and highlights the need for intelligent optimization algorithms. In this study, the IJFSA-ODA algorithm is introduced as an improved version of the Jellyfish algorithm that enhances search performance through the purposeful integration of several metaheuristic operators. Specifically, an operator inspired by Differential Evolution is employed to increase diversity, a mechanism inspired by the Artificial Bee Colony algorithm is used to strengthen exploitation, and an operator inspired by Simulated Annealing is applied to prevent premature convergence. Simulation results demonstrate that the proposed algorithm exhibits competitive performance compared with IJFSA, IJFA, PSO-SA, and PSO, and achieves acceptable improvements in key performance metrics under certain scenarios. For a scenario involving 25 virtual machines and up to 2000 tasks, the results indicate a relative reduction of approximately 5–10% in total processing time and 5–12% in energy consumption compared to PSO-based methods. From the perspective of load balancing, the results of Jain's Fairness Index indicate that although the index does not reach its ideal value (close to 1) for any of the evaluated algorithms, the proposed algorithm achieves higher fairness values than the reference methods in most scenarios, reflecting a more balanced workload distribution among virtual machines. Furthermore, the results show that the performance of the IJFSA-ODA algorithm remains stable as the number of tasks and nodes increases, exhibiting lower fluctuations in the deadline-meet ratio compared to some baseline algorithms. In addition, the results obtained from 10 independent runs confirm the acceptable statistical stability and reliability of the proposed algorithm.

Keywords- Load Balancing, Task Scheduling, IJFSA, Artificial Bee Colony, Evolutionary Algorithm, Energy Consumption

مجله علمی رایانش نرم و فناوری اطلاعات
دانشگاه صنعتی نوشیروانی بابل
صفحه مجله: jscit.nit.ac.ir
شناسه دیجیتال: 10.22034/jscit.2026.560929.2194
جلد ۱۵، شماره ۲، تابستان ۱۴۰۵، صفحه ۴۸-۶۳
دریافت: ۱۴۰۴/۰۹/۱۰، بازنگری: ۱۴۰۵/۰۲/۲۳، پذیرش: ۱۴۰۵/۰۲/۲۸

مدل ترکیبی بهینه‌سازی چندمرحله‌ای مبتنی بر IJFSA برای تعادل بار و بهبود کارایی در محیط‌های مه–ابر

سکینه سهرابی[۱]، رضا محمدی *[۲]، مهدی سخایی نیا [۳]، محمد نصیری [۴]
۱،۲*،۳،۴ - دانشکده فنی و مهندسی، دانشگاه بوعلی سینا، همدان، ایران.
[1] s.sohrabi@eng.basu.ac.ir, [2*] r.mohammadi@basu.ac.ir, [3] sakhaei@basu.ac.ir, [4] m.nassiri@basu.ac.ir

* نشانی نویسنده مسئول: رضا محمدی، همدان، دانشگاه بوعلی سینا، دانشکده فنی و مهندسی.

چکیده- در سال‌های اخیر، فناوری‌های نوین شیوه تعامل انسان با سامانه‌های دیجیتال را دگرگون کرده‌اند. ترکیب اینترنت اشیا با زیرساخت‌های ابری، بستر ارائه خدمات هوشمند و خودکار را، به‌ویژه در حوزه سلامت، فراهم کرده است. با افزایش حسگرهای متصل به شبکه، حجم زیادی از داده‌های بلادرنگ تولید می‌شود که نیازمند پردازش سریع و کارآمد است. از سوی دیگر، اتکا به مراکز داده دوردست می‌تواند موجب افزایش تأخیر و کاهش کارایی در سامانه‌های حساس به زمان شود. در این راستا، محاسبات مه به‌عنوان لایه‌ای واسط میان دستگاه‌های لبه‌ای و ابر، راهکاری مؤثر برای کاهش تأخیر و بهبود کیفیت خدمات ارائه می‌دهد. در این پژوهش، الگوریتم IJFSA-ODA به‌عنوان نسخه‌ای بهبودیافته از الگوریتم عروس دریایی معرفی شده است که با بهره‌گیری از عملگرهای تکاملی و سازوکارهای مکمل، توان جست‌وجو و تعادل میان اکتشاف و بهره‌برداری را ارتقا می‌دهد. نتایج شبیه‌سازی نشان می‌دهد که الگوریتم پیشنهادی در مقایسه با الگوریتم‌های IJFSA، IJFA، PSO-SA و PSO عملکردی رقابتی داشته و در برخی سناریوها بهبود قابل‌توجهی در شاخص‌های اصلی ارائه می‌دهد. در سناریوی شامل ۲۵ گره پردازشی مه و ۲۰۰۰ وظیفه، نتایج بیانگر کاهش نسبی حدود ۵ تا ۱۰ درصد در زمان پردازش کل و حدود ۵ تا ۱۲ درصد در مصرف انرژی نسبت به روش‌های مبتنی بر PSO است. همچنین، از منظر عدالت توزیع بار، مقادیر شاخص عدالت جین نشان می‌دهد که هرچند هیچ‌یک از الگوریتم‌ها به مقدار ایده‌آل نزدیک به ۱ نرسیده‌اند، الگوریتم پیشنهادی در اغلب سناریوها عملکرد بهتری نسبت به روش‌های مرجع داشته و توزیع متوازن‌تری از بار را میان ماشین‌های مجازی فراهم کرده است. افزون بر این، نتایج نشان می‌دهد که عملکرد IJFSA-ODA با افزایش تعداد وظایف و گره‌ها از پایداری مناسبی برخوردار بوده و در شاخص درصد وظایف انجام‌شده در مهلت مقرر، نوسان کمتری نسبت به برخی روش‌های مرجع دارد. نتایج حاصل از ۱۰ تکرار مستقل نیز پایداری آماری قابل‌قبول الگوریتم پیشنهادی را تأیید می‌کند.

واژه‌های کلیدی: تعادل بار، زمان‌بندی وظایف، الگوریتم IJFSA، کلونی زنبور عسل مصنوعی، الگوریتم تکاملی، مصرف انرژی

۱- مقدمه

اینترنت اشیا[۱] به‌عنوان یکی از فناوری‌های کلیدی و تحول‌آفرین، نقشی بنیادین در گسترش زیرساخت‌های ارتباطی ایفا کرده است. توسعه تجهیزات مقرون‌به‌صرفه و هوشمند موجب شده است میلیاردها دستگاه در سراسر جهان به یکدیگر متصل شوند و طیف وسیعی از خدمات شامل کنترل هوشمند ترافیک و پایش سلامت فراهم گردد. این روند علاوه بر ارتقای کیفیت زندگی، سهم قابل‌توجهی در افزایش سطح دسترسی به اینترنت و خدمات مبتنی بر آن داشته است. با وجود مزایا، دستگاه‌های اینترنت اشیا حجم عظیمی از داده تولید می‌کنند که پردازش مؤثر آن‌ها برای بهره‌برداری عملیاتی ضروری است. اشتراک‌گذاری منابع توسط

تعداد زیاد دستگاه‌ها محدودیت پهنای باند ایجاد کرده و تأخیر ارتباطی را افزایش می‌دهد؛ بنابراین، استفاده از راهکارهای بهینه برای مدیریت منابع و اولویت‌بندی وظایف اجتناب‌ناپذیر است. محاسبات ابری با ظرفیت بالای محاسباتی و ذخیره‌سازی، محدودیت پردازشی دستگاه‌ها را کاهش می‌دهد، اما انتقال داده‌ها به مراکز دوردست تأخیر و ترافیک شبکه ایجاد می‌کند. در پاسخ، محاسبات مه[۲] امکان پردازش داده‌ها نزدیک به منبع را فراهم کرده و تأخیر را کاهش، مصرف انرژی را بهبود و امنیت و مدیریت منابع را ارتقا می‌دهد [۱]. ترکیب مه و ابر، پردازش سریع‌تر و دسترسی کارآمدتر به داده‌ها را ممکن می‌سازد، اما ماهیت پویا و ناهمگن منابع، تخصیص و زمان‌بندی را به مسئله‌ای پیچیده و NP سخت تبدیل کرده که بر هزینه، انرژی و کیفیت خدمات تأثیر مستقیم دارد [۲-۴]. زمان‌بندی وظایف در محیط مه–ابر علاوه بر افزایش بهره‌وری، نقش مهمی در جلوگیری از گره‌های گلوگاهی و حفظ تعادل بار دارد. درواقع زمان‌بندی وظایف با توزیع یکنواخت بار و مدیریت مؤثر گلوگاه‌ها، امکان افزایش بهره‌وری و کاهش احتمالی تأخیر در سیستم را فراهم می‌کند. تعادل بار موجب توزیع یکنواخت وظایف، کاهش زمان پاسخ، صرفه‌جویی در انرژی و کاهش هزینه‌های عملیاتی می‌شود و کیفیت خدمات[۳] را ارتقا می‌بخشد. روش‌های موجود عمدتاً بر یک معیار عملکرد تمرکز دارند و توانایی مقابله همزمان با شرایط پویا و بارکاری نامتوازن را ندارند [۵-۷].

در سال‌های اخیر، رویکردهای متعددی در حوزه زمان‌بندی وظایف در محیط مه–ابر مورد بررسی قرار گرفته‌اند که شامل الگوریتم‌های ابتکاری و فراابتکاری هستند. این رویکردها عمدتاً با هدف ایجاد توازن میان شاخص‌هایی چون زمان اجرا، مصرف انرژی و درصد وظایف انجام‌شده در مهلت مقرر[۴] طراحی شده‌اند. در این میان، ترکیب الگوریتم‌های الهام‌گرفته از طبیعت، به‌عنوان روش‌هایی نوین برای ارتقای کارایی سیستم مطرح شده‌اند. آنها با در نظر گرفتن اهمیت وظایف بر اساس معیارهایی همچون مصرف انرژی و زمان اجرا، امکان زمان‌بندی بهینه و تخصیص مؤثرتر منابع را فراهم می‌سازند [۸-۹]. باید توجه داشت که تعادل بار در زمان‌بندی وظایف نقش کلیدی در جلوگیری از ایجاد گره‌های گلوگاهی ایفا می‌کند. این امر باعث کاهش زمان پاسخ، کاهش تأخیر و افزایش پایداری شبکه می‌شود. تعادل بار مناسب موجب می‌گردد وظایف به‌صورت یکنواخت در گره‌های پردازشی مه (ماشین مجازی) توزیع شوند، بار اضافی از روی گره‌های اشباع‌شده برداشته شده و به گره‌های پایدارتر منتقل گردد، که این خود تحمل‌پذیری خطا را افزایش می‌دهد. چنین سازوکاری به کاهش مصرف انرژی، افزایش طول عمر دستگاه‌ها و کاهش هزینه‌های عملیاتی نیز کمک می‌کند.

با وجود پیشرفت‌های انجام‌شده در زمان‌بندی وظایف و توزیع بار، بسیاری از الگوریتم‌های تکاملی و هوش جمعی همچنان در بهینه‌های محلی گرفتار شده و قادر به ایجاد تعادل مناسب میان جستجوی سراسری و بهره‌برداری محلی نیستند. این مسئله موجب کاهش کارایی سیستم می‌شود. در این راستا، پژوهش حاضر با معرفی الگوریتم IJFSA-ODA می‌کوشد این خلا پژوهشی را برطرف کرده و با بهره‌گیری از مکانیزم‌های فعال و غیرفعال، امکان بهینه‌سازی همزمان چندین معیار کلیدی و دستیابی به عملکردی پایدار در محیط‌های دارای بارکاری نامتوازن را فراهم آورد. در این پژوهش، الگوریتم پیشنهادی تحت عنوان IJFSA-ODA برای زمان‌بندی وظایف و توزیع بار در محیط‌های مه–ابر ارائه شده است. این الگوریتم با الهام از الگوریتم‌های تکاملی و هوش جمعی شامل DE[۵] و ABC[۶] برای بهبود الگوریتم جستجوی عروس دریایی بهبودیافته[۷]، طراحی شده تا توانایی فرار از بهینه‌های محلی را افزایش دهد و در عین حال جستجوی سراسری مؤثر و بهره‌برداری محلی کارآمد را فراهم کند. مکانیزم‌های فعال و غیرفعال عروس دریایی به الگوریتم اجازه می‌دهند در شرایط پویا و بارکاری نامتوازن عملکرد پایدار و قابل اعتمادی داشته باشد. نتایج شبیه‌سازی کاهش زمان پردازش کل، بهبود تعادل بار میان ماشین‌های مجازی مستقر در لایه مه و کاهش مصرف انرژی را در مقایسه با روش‌های مرجع نشان می‌دهد و همزمان معیارهای کلیدی کیفیت خدمات و عدالت توزیع بار بهینه شده‌اند. مزیت اصلی IJFSA-ODA توانایی همزمان بهینه‌سازی چند معیار کلیدی از جمله تعادل بار و کیفیت خدمات است. این ویژگی موجب عملکرد پایدار و قابل اعتماد الگوریتم در شرایط بارکاری نامتوازن و پویا می‌شود و نقطه قوت آن نسبت به بسیاری از روش‌های تک‌معیاره به شمار می‌آید.

هدف اصلی این پژوهش، ارائه یک راهکار زمان‌بندی و تخصیص منابع کم‌مصرف و منصفانه برای کاربردهای اینترنت اشیا است. در این راستا، یک الگوریتم مبتنی بر بهینه‌سازی ازدحام عروس دریایی و تخصیص بهینه وظایف ارائه می‌شود که با بهره‌گیری گزینشی از عملگرهای تکاملی، تلاش می‌کند نرخ انجام وظایف در مهلت مقرر را افزایش داده و هم‌زمان مصرف انرژی و عدم تعادل بار را کاهش دهد. نوآوری اصلی این پژوهش در طراحی یک چارچوب چندمرحله‌ای سبک‌وزن نهفته است که بدون افزایش پیچیدگی محاسباتی، عملکرد سیستم را در معیارهای کلیدی بهبود می‌بخشد. ادامه مقاله به این صورت سازمان‌دهی شده است. در بخش دوم، ضمن مرور کارهای پیشین و جایگاه پژوهش حاضر در میان مطالعات انجام‌شده، معماری سیستم و فرموله‌سازی مسئله شامل متغیرهای تصمیم، معیارهای عملکرد و محدودیت‌های سیستم تشریح می‌شود.

در بخش سوم، جزئیات الگوریتم پیشنهادی IJFSA-ODA، از جمله نحوه تولید جمعیت اولیه، سازوکارهای جست‌وجو، مراحل به‌روزرسانی و تابع هدف، ارائه می‌شود. در بخش چهارم، نتایج شبیه‌سازی، ارزیابی عملکرد، مقایسه با روش‌های مرجع و تحلیل پایداری آماری الگوریتم مورد بررسی قرار می‌گیرد. سرانجام، بخش پنجم به جمع‌بندی یافته‌های اصلی پژوهش و ارائه پیشنهادهایی برای تحقیقات آینده اختصاص دارد.

۲- پژوهش‌های پیشین

در این مقاله، مطالعات مرتبط با زمان‌بندی وظایف در محیط‌های مه–ابر به‌صورت هدفمند در چهار دسته کلی سازمان‌دهی شده‌اند: روش‌های مبتنی بر الگوریتم‌های فراابتکاری با تمرکز بر کاهش زمان تکمیل، رویکردهای ترکیبی چندالگوریتمی با اهداف چندمعیاره نظیر تعادل بار و مصرف انرژی، روش‌های مبتنی‌بر فناوری بلاک‌چین که عمدتاً با هدف افزایش امنیت و قابلیت اعتماد طراحی شده‌اند و چارچوب‌های هوشمند و پیش‌بین مبتنی‌بر مدل‌های یادگیری برای تصمیم‌گیری زمان‌بندی. تمرکز اصلی این مطالعه بر دسته‌های اول، دوم و چهارم قرار دارد و رویکردهای مبتنی بر بلاک‌چین صرفاً به‌منظور تکمیل تصویر کلی حوزه و تفکیک دامنه مسئله مورد اشاره قرار گرفته‌اند. در ادامه، هر یک از این دسته‌ها متناسب با میزان ارتباط با مسئله پژوهش حاضر به‌صورت هدفمند مرور می‌شوند.

فرآیند زمان‌بندی وظایف به‌عنوان یکی از ارکان اساسی، نقش تعیین‌کننده‌ای در بهبود عملکرد برنامه‌های مبتنی بر اینترنت اشیا دارد که متکی بر زیرساخت‌های ابری و مه‌ای هستند. پیچیدگی و تنوع وظایف در این محیط‌ها سبب شده است که در سال‌های اخیر رویکردهای متعددی برای پاسخ‌گویی به چالش‌های موجود مطرح شوند. در این بخش، روش‌های زمان‌بندی وظایف در محیط‌های صرفاً ابری، صرفاً مه‌ای و همچنین محیط‌های یکپارچه مه–ابر مرور و تحلیل می‌گردند. در محیط مه، پژوهش‌های متنوعی با هدف ارتقای کارایی و کاهش تأخیر انجام شده است. برای نمونه،Zhang و همکاران الگوریتم کلونی مورچگان[۸] (ACO) را برای کاهش زمان تکمیل و بهبود تعادل بار معرفی کردند، هرچند این الگوریتم در مقیاس‌های بزرگ کارایی مطلوبی نداشت [۱۰]. در ادامه، Kumar و همکاران با ترکیب الگوریتم ژنتیک[۹] (GA) و ازدحام ذرات[۱۰] (PSO) رویکردی نوین برای زمان‌بندی ارائه دادند، اما ضعف اساسی در اولویت‌بندی وظایف همچنان باقی ماند [۱۱]. از سوی دیگر، Chen و همکاران با ترکیب فناوری بلاک‌چین و الگوریتم ACO بر کاهش زمان اجرای کل تمرکز داشتند، اما جنبه‌هایی نظیر هزینه و تأخیر مورد توجه قرار نگرفت [۱۲]. لازم به ذکر است که اگرچه استفاده از بلاک‌چین می‌تواند مزایایی نظیر افزایش امنیت و قابلیت اعتماد را فراهم آورد، اما این دسته از مطالعات معمولاً با افزایش پیچیدگی محاسباتی و سربار سیستمی همراه هستند. از آنجا که تمرکز پژوهش حاضر بر بهینه‌سازی زمان‌بندی و کارایی منابع است و ملاحظات امنیتی در دامنه مسئله تعریف نشده‌اند، این رویکردها به‌عنوان راهکارهای مرجع غیرهم‌راستا در نظر گرفته می‌شوند.

در پژوهشی دیگر، میرحسینی همکاران یک راهبرد ترکیبی ارائه دادند که در آن GA با الگوریتم ازدحام ذرات فازی[۱۱] (FPSO) ادغام شده بود. هدف این روش، توزیع بهینه وظایف میان ماشین‌های مجازی بود و نتایج شبیه‌سازی با CloudSim نشان داد که ترکیب FPSO–GA به دلیل بهره‌گیری از منطق فازی عملکرد بهتری در ایجاد تعادل بار دارد [۱۳]. Malik و Suman نیز با ترکیب PSO و روش مبتنی بر گرگ خاکستری، الگوریتم LW–PSO را طراحی کردند که بهبود چشمگیری در زمان پاسخ و تکمیل وظایف به همراه داشت [۱۴]. همچنین Singh و همکاران با ترکیب PSO و ACO به بهبود تعادل بار دست یافتند، هرچند مسئله مصرف انرژی همچنان حل‌نشده باقی ماند [۱۵]. از منظر رویکردهای ترکیبی، Baniata و همکاران الگوریتمی تلفیقی مبتنی بر ACO و فناوری بلاک‌چین[۱۲] را در محیط ابری–مه پیشنهاد دادند. در این مدل، ماینرهای بلاک‌چین وظیفه ایجاد زمان‌بندی وظایف را برعهده دارند و سیستم پاداش‌دهی نیز به بهبود کارایی کمک می‌کند. نتایج شبیه‌سازی بیانگر بهبود حریم خصوصی، کاهش ترافیک شبکه و تسریع اجرای وظایف بود، هرچند به دلیل فرض ورود هم‌زمان وظایف، این الگوریتم در دسته روش‌های زمان‌بندی ایستا قرار می‌گیرد [۱۶]. در راستای توزیع متوازن بار Aldia و همکاران دو الگوریتم ارسال ترتیبی[۱۳] و ارسال تطبیقی [۱۴]را ارائه کردند که قابلیت پایداری در شرایط متنوع شبکه مه با بارهای کاری متغیر و تأخیرهای بالا را تضمین می‌کردند [۱۷].

در حوزه بهینه‌سازی چندمعیاره، Sumathi و همکاران الگوریتمی ترکیبی از شاهین‌های هریس[۱۵] (HHO) و ACO ارائه کردند که تحت عنوان HLB شناخته می‌شود. نتایج نشان داد که HLB در معیارهایی نظیر زمان پاسخ متوسط، زمان اجرا، توان عملیاتی و تعادل بار عملکرد برتری نسبت به روش‌های منفرد دارد [۱۸]. همچنین Gabhane و همکاران روشی نوآورانه با ادغام ACO و جستجوی ممنوعه[۱۶] (TS) پیشنهاد کردند که در مقایسه با روش‌های مرسوم همچون GA، PSO، ACO و TS در کاهش زمان تکمیل و هزینه کل نتایج بهتری کسب نمود [۱۹].

Salimi و همکاران [۲۰] با ترکیب یک روش ژنتیکی و یک سازوکار

یادگیری مبتنی بر مقایسه متضاد، مدلی ارائه کرده‌اند که در آن تصمیم‌گیری زمان‌بندی نه‌تنها به وضعیت منابع، بلکه به سطح اهمیت وظایف نیز وابسته است. ارزیابی‌های تجربی نشان می‌دهد این رویکرد در سناریوهای شبیه‌سازی‌شده قادر است شاخص‌های کارایی سیستم را نسبت به راهکارهای متداول بهبود دهد. با این حال، افزایش سربار محاسباتی و وابستگی عملکرد به تنظیمات اولیه، استفاده گسترده از این روش را با محدودیت مواجه می‌سازد. در مقابل، گروهی از مطالعات با تمرکز بر ساده‌سازی ساختار الگوریتم و بهبود بهره‌وری منابع شبکه به سراغ اصلاح الگوریتم‌های جمعیتی رفته‌اند. Alsadie و Alsulami [۲۱] با اعمال تغییراتی در منطق جستجوی الگوریتم گرگ خاکستری، راهکاری را برای مدیریت بار کاری گره‌های fog ارائه کرده‌اند که هدف آن کاهش مصرف انرژی و جلوگیری از تمرکز بار روی تعداد محدودی از گره‌هاست. نتایج حاصل از شبیه‌سازی بیانگر بهبود عملکرد این روش در مقایسه با الگوریتم‌های مرجع است؛ با این وجود، در محیط‌های بزرگ و ناهمگون، مسائلی مانند افت مقیاس‌پذیری و همگرایی زودهنگام همچنان مشاهده می‌شود.

رویکردهای پیچیده‌تر معمولا به سراغ ترکیب چند الگوریتم فراابتکاری می‌روند تا بتوانند هم‌زمان چند معیار متضاد را پوشش دهند. در همین چارچوب، Rateb و همکاران [۲۲] مدلی چندلایه برای زمان‌بندی جریان‌های کاری پیشنهاد کرده‌اند که از تلفیق الگوریتم عقاب با یک روش جستجوی تطبیقی الهام‌گرفته از رفتار دارکوب استفاده می‌کند. این ساختار ترکیبی با درنظرگرفتن محدودیت‌های زمانی، موفق به کاهش زمان تکمیل و مصرف انرژی شده است. با این حال، پیچیدگی ذاتی مدل و دشواری پیاده‌سازی آن در شرایط عملی، از چالش‌های اصلی این رویکرد به شمار می‌رود. در دسته‌ای دیگر از تحقیقات، به‌جای تکیه صرف بر بهینه‌سازی واکنشی، از مدل‌های پیش‌بینی برای تصمیم‌گیری استفاده شده است Nazeri .و همکاران [۲۳] چارچوبی هوشمند را معرفی کرده‌اند که با بهره‌گیری از سیستم استنتاج عصبی–فازی تطبیقی و چرخه کنترلی MAPE-K، رفتار آینده بار کاری را تخمین زده و تخصیص منابع را به‌صورت پیش‌نگر انجام می‌دهد. این رویکرد به‌ویژه در محیط‌های پویا عملکرد قابل قبولی از خود نشان می‌دهد و احتمال نقض مهلت‌های اجرایی را کاهش می‌دهد؛ هرچند نیاز به داده‌های آموزشی و حساسیت به دقت پیش‌بینی، از محدودیت‌های اجتناب‌ناپذیر آن محسوب می‌شود.

مطالعات اخیر (جدول ۱) نشان می‌دهند که رویکردهای زمان‌بندی وظایف به مجموعه‌ای از اهداف از جمله کاهش مصرف انرژی، بهبود بهره‌وری منابع، کاهش هزینه‌های عملیاتی، کوتاه‌سازی زمان پاسخ و زمان تکمیل، و ایجاد توزیع متوازن بار توجه داشته‌اند. تصمیم‌گیری‌های ضعیف در زمینه زمان‌بندی می‌تواند تأثیر مستقیم و منفی بر کیفیت خدمات در محیط‌های مه–ابر داشته باشد.

برای پاسخ به چالش‌های موجود، این پژوهش یک چارچوب چندهدفه مبتنی بر الگوریتم IJFSA ارائه می‌کند که با بهره‌گیری از راهبردهای الهام‌گرفته از DE و ABC، تعادل مؤثری میان اکتشاف سراسری و بهره‌برداری محلی برقرار می‌سازد و احتمال گرفتار شدن در بهینه‌های محلی را کاهش می‌دهد. استفاده از مکانیزم‌های فعال و غیرفعال عروس دریایی، امکان لحاظ هم‌زمان اولویت وظایف، بهبود تعادل بار و رعایت معیارهای کلیدی توافق‌نامه سطح خدمات از جمله مهلت اجرا، محدودیت تأخیر و کارایی مصرف منابع را فراهم می‌کند. این مطالعه در ادامه کار پیشین ما بر روی IJFSA انجام شده و نسخه بهبودیافته آن با نام IJFSA-ODA، زمینه زمان‌بندی تطبیقی و توزیع بار آگاه از انرژی را فراهم کرده و در نهایت به افزایش مقیاس‌پذیری، کاهش زمان اجرا و پایداری عملکرد در محیط‌های پویا و پیچیده مه–ابر منجر می‌شود.

۳- مدل سیستم

در این پژوهش، ساختار سیستم و تعریف مسئله با استفاده از یک چارچوب سفارشی که بر پایه‌ی مفاهیم مطرح‌شده در مطالعات پیشین [۲۴] شکل گرفته، مورد بررسی قرار می‌گیرد. توضیحات این بخش در دو قسمت اصلی ارائه شده است که بخش نخست به معرفی کلی معماری محیط اینترنت اشیا–محاسبات مه–محاسبات ابری اختصاص دارد. در این پژوهش، به‌منظور جلوگیری از ابهام مفهومی، کلیه منابع پردازشی موجود در لایه مه تحت عنوان «گره‌های پردازشی مه» در نظر گرفته می‌شوند. هر گره پردازشی مه به‌صورت منطقی در قالب یک ماشین مجازی مدل‌سازی شده است. بنابراین، در ادامه مقاله، اصطلاحات «گره پردازشی مه» و «ماشین مجازی» به‌صورت معادل و با اشاره به یک مفهوم واحد به‌کار می‌روند.

۳-۲- معماری سیستم

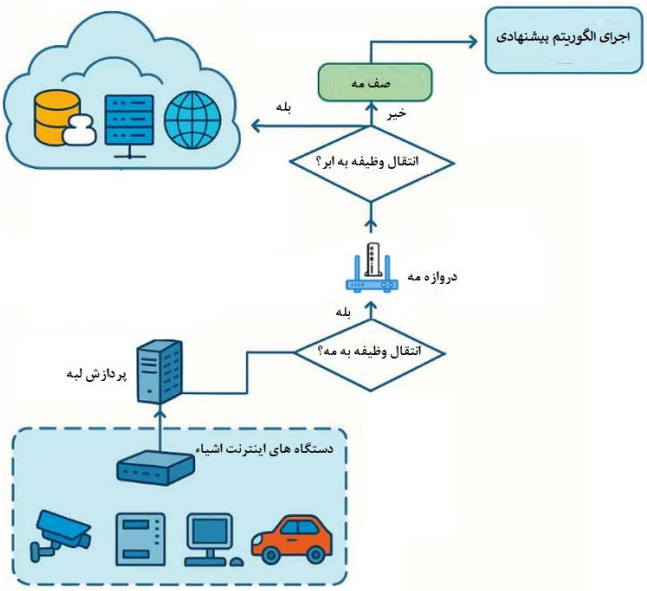
شکل ۱ نمایی کلی از معماری پیشنهادی را نمایش می‌دهد. این چارچوب بر سه مؤلفه‌ی اصلی شامل تجهیزات نهایی اینترنت اشیا، منابع پردازشی در لایه مه و خدمات ابری استوار است. در پیکربندی پیشنهادی، دستگاه‌های اینترنت اشیا به‌عنوان گره‌های انتهایی در نظر گرفته شده‌اند که قادر به انجام پردازش محلی اولیه بر روی وظایف خود هستند. با این حال، به‌دلیل محدودیت منابع محاسباتی و به‌منظور بهبود شاخص‌های عملکردی، بخشی یا کل وظایف می‌توانند به لایه مه offload شوند. کلیه پردازش‌های

جدول ۱: مروری مقایسه‌ای بر روش‌های مختلف زمان‌بندی وظایف

مرجع	روش مورد استفاده	کاستی‌ها	محیط آزمایش	اهداف اصلی	سال انتشار
[۱۰]	ACO	کاهش کارایی در مقیاس بزرگ	MATLAB	کاهش زمان تکمیل، تعادل بار	۲۰۲۲
[۱۱]	GA-PSO	عدم توجه به اولویت وظایف	CloudSim	بهبود زمان‌بندی و هزینه	۲۰۲۲
[۱۳]	Blockchain-ACO	عدم بررسی هزینه و تأخیر	MATLAB	کاهش زمان اجرا و بار شبکه	۲۰۲۱
[۱۴]	ترکیب FPSO وGA	در زمینه‌ی اولویت‌دهی، موازنه بار و کاهش تأخیر، نتایج مطلوبی ارائه نمی‌دهد.	CloudSim	کاهش زمان اجرا (Makespan) و هزینه	۲۰۲۱
[۱۵]	LW-PSO	در این روش، اولویت وظایف در نظر گرفته نشده است.	CloudSim	استفاده بهینه از پردازنده، میانگین زمان چرخش، زمان اجرا، مصرف حافظه، و میانگین زمان پاسخ	۲۰۲۲
[۱۶]	الگوریتم ACO با پشتیبانی بلاک‌چین	هزینه، اولویت وظایف و تأخیرها لحاظ نشده‌اند.	Python 3.8	کاهش زمان کلی اجرا و بار شبکه	۲۰۲۱
[۱۷]	روش‌های انتقال ترتیبی و انتقال تطبیقی	هزینه محاسباتی در مدل لحاظ نشده است.	Omnet++	کاهش زمان پاسخ، کاهش نرخ خطا، در نظر گرفتن شرایط بلادرنگ	۲۰۲۰
[۱۸]	موازنه بار ترکیبی	در مقیاس بزرگ و شبکه‌های SDN با کنترل‌کننده‌های متعدد، با چالش روبه‌روست.	CloudSim	عملکرد بهتر نسبت به الگوریتم‌های HHO و ACO منفرد	۲۰۲۳
[۱۹]	کلونی مورچگان همراه با جست‌وجوی ممنوع(Tabu)	با بارهای کاری پویا یا ترافیک غیرمنتظره تطبیق‌پذیری ندارد.	شبیه‌ساز اختصاصی	کاهش زمان اجرا، افزایش توان عملیاتی و اجرای مقرون‌به‌صرفه وظایف	۲۰۲۳
[۲۰]	الگوریتم ژنتیک توسعه‌یافته با یادگیری تقابلی	سربار محاسباتی قابل توجه و وابستگی عملکرد به تنظیم دقیق پارامترها	شبیه‌سازی	بهینه‌سازی هم‌زمان مدت اجرای وظایف و مصرف انرژی	۲۰۲۵
[۲۱]	نسخه بهبودیافته بهینه‌سازی گرگ خاکستری(GWO)	افت کارایی در مقیاس‌های بزرگ و احتمال توقف در جواب‌های غیربهینه	شبیه‌سازی	کاهش مصرف انرژی همراه با توزیع یکنواخت بار	۲۰۲۵
[۲۲]	روش ترکیبی ASSA با استفاده از الگوریتم های Aquila Optimizer و Salp Swarm	ساختار پیچیده و محدودیت در تعمیم‌پذیری به شرایط عملی	شبیه‌سازی	تضمین مهلت اجرا در کنار کاهش انرژی و زمان تکمیل	۲۰۲۵
[۲۳]	چارچوب هوشمند مبتنی بر ANFIS و MAPE-K	نیازمند داده‌های پیشین و حساس به دقت پیش‌بینی	شبیه‌سازی همراه با مدل پیش‌بینی	زمان‌بندی پویا و پیشگیری از نقض مهلت اجرا	۲۰۲۴

برون‌سپاری‌شده در گره‌های پردازشی مه انجام می‌گیرد و این گره‌ها مسئول اجرای بخش اصلی محاسبات هستند. جریان کاری تعریف‌شده به این صورت است که وظایف ابتدا در سطح دستگاه اینترنت اشیا ایجاد شده و در صورت نیاز پس از پردازش محلی اولیه، به لایه مه منتقل می‌شوند تا پردازش تکمیلی بر روی آن‌ها انجام شود. در برخی شرایط خاص، نظیر بار کاری بالا یا نیاز به منابع محاسباتی گسترده‌تر، امکان انتقال وظایف از لایه مه به منابع ابری نیز در نظر گرفته شده است.

در مطالعه‌ی حاضر، تمرکز اصلی بر اجرای وظایف در لایه مه قرار دارد و تصمیم‌گیری در خصوص تخصیص وظایف به مه یا ابر توسط مؤلفه‌ی کارگزار انجام می‌شود. فرایند زمان‌بندی وظایف بر اساس فرمول‌بندی‌های ریاضی تعریف شده و همان‌گونه که در شکل ۱ نشان داده شده است، رویکرد پیشنهادی در قالب سه گام کلی ارائه می‌شود.



شکل ۱: معماری سیستم در رویکرد پیشنهادی

۱. برآورد تأخیرها، تعیین اولویت‌ها و سازمان‌دهی صف وظایف بر اساس سطح اهمیت آن‌ها.
۲. تخمین زمان تحویل و ارزیابی تأخیرها و سپس بازتنظیم اولویت‌ها به منظور بهبود سازمان‌دهی صف وظایف.
۳. افزایش اهمیت وظایف بر اساس مهلت اجرا تکمیل آن‌ها به‌گونه‌ای که وظایف حساس‌تر در صف پردازش در موقعیت بالاتری قرار گیرند.

این بخش یک چارچوب زمان‌بندی ویژه برای سامانه‌های محاسبات مه–ابر معرفی می‌کند که با استفاده از یک الگوریتم نوآورانه، کارایی در مدیریت وظایف را ارتقا می‌دهد. هدف اصلی، ارتقای کارایی سیستم و بهبود بهره‌وری شبکه از طریق توزیع متعادل بار محاسباتی است. در ابتدای این بخش، متغیرهای تصمیم‌گیری و مؤلفه‌های اصلی مسئله بهینه‌سازی معرفی می‌شوند. سپس مدل‌های پیشنهادی به‌صورت دقیق تشریح شده و در ادامه تابع هدف بر اساس شاخص‌های ارزیابی انتخاب‌شده تدوین می‌گردد. در پایان، محدودیت‌های سیستم از جمله ظرفیت پردازشی گره‌ها و قیود مربوط به پهنای باند بررسی می‌شوند. در این مدل، درخواست‌های کاربران به یک کارگزار ارسال می‌شود که وظیفه هماهنگی و مدیریت منابع موجود در لایه مه را بر عهده دارد.

۳-۳- فرموله سازی مسئله

در این بخش، یک مدل جامع برای زمان‌بندی وظایف با هدف کاهش تأخیر انتها به انتها، مصرف انرژی و ایجاد تعادل بار میان گره‌های پردازشی ارائه می‌شود. زمان‌بندی پیشنهادی مبتنی بر یک رویکرد چندمرحله‌ای است که در آن الگوریتم IJFSA به‌عنوان هسته اصلی جستجوی سراسری به‌کار گرفته شده و با مکانیزم‌های الهام‌گرفته از الگوریتم‌های DE و ABC بهبود یافته است. در این چارچوب، IJFSA مسئول اکتشاف فضای جستجو بوده، سازوکار مبتنی بر DE برای اصلاح موقعیت‌ها و جلوگیری از همگرایی زودرس استفاده می‌شود و مکانیزم الهام‌گرفته از ABC تقویت جستجوی محلی را بر عهده دارد. این ترکیب با حفظ تنوع و ایجاد تعادل میان اکتشاف و بهره‌برداری، منجر به یک زمان‌بند هوشمند و سازگار با ماهیت پویا و نامطمئن محیط‌های مه–ابر می‌شود.

۳-۳-۱- متغیرهای تصمیم

تحلیل شبکه مورد نظر، یک معماری سه‌لایه‌ای سلسله‌مراتبی شامل اینترنت اشیا، مه و ابر در نظر گرفته شده است. لایه لبه[۱۷] شامل حسگرها و دستگاه‌های اینترنت اشیا است که به‌صورت تصادفی در محیط استقرار یافته‌اند.

در پایین‌ترین سطح، مجموعه‌ای از دستگاه‌های اینترنت اشیا قرار دارند که در نقاط مختلف محیط توزیع شده‌اند. هر دستگاه اینترنت اشیا با نرخ مشخصی وظایف تولید می‌کند. این وظایف ممکن است مستقیماً در همان دستگاه پردازش شوند یا برای انجام عملیات محاسباتی به لایه‌های بالاتر ارسال شوند. مجموعه دستگاه‌های اینترنت اشیا به صورت رابطه (۱) تعریف می‌شود:

$$I = \{d_1, d_2, \dots, d_N\} \quad (۱)$$

هر دستگاه d_i دارای سه ویژگی اصلی است که به صورت $d_i = (f_i^{cpu}, f_i^{tx}, \lambda_i)$ نشان داده می‌شود که به ترتیب عبارت‌اند از: فرکانس پردازنده، توان ارسال و نرخ تولید وظیفه. لایه مه شامل مجموعه‌ای از گره‌های پردازشی است که به صورت نیمه‌متمرکز مدیریت می‌شوند. مجموعه گره‌های مه به شکل رابطه (۲) نمایش داده می‌شود:

$$\mathcal{F} = \{g_1, g_2, \dots, g_M\} \quad (۲)$$

هر گره مه با یک پنج‌تایی جدید به شکل $g_m = (C_m^{cpu}, C_m^{mem}, B_m^{up}, B_m^{down}, P_m)$ مدل می‌شود. باید توجه داشت که متغیرهای C_m^{cpu}، C_m^{mem}، B_m^{up}, B_m^{down} و P_m به‌ترتیب نشان‌دهنده توان پردازشی، ظرفیت حافظه، پهنای باند آپلود، پهنای باند دانلود و توان مصرفی پردازنده گره مه هستند. علاوه بر این، یک کارگزار مه مرکزی وظیفه مدیریت فرآیند زمان‌بندی و هماهنگ‌سازی بین گره‌های مه را بر عهده دارد. لایه سوم شامل مراکز داده ابری با توان پردازشی بسیار بالا است. این لایه وظایفی را دریافت می‌کند که پردازش آن‌ها فراتر از ظرفیت گره‌های مه باشد. هر وظیفه τ_kمطابق رابطه (۳) مدل‌سازی می‌شود.

$$\tau_k = (s_k, \gamma_k, pr_k, dl_k) \quad (۳)$$

که در آن متغیرهای s_k، γ_k، pr_k و dl_k به ترتیب بیانگر اندازه داده، چگالی محاسباتی (تعداد دستور به ازای بیت)، سطح اولویت و مهلت اجرا است که با استفاده از رابطه (۴) محاسبه می‌شود.

$$dl_k = t_k^{arrive} + \Delta_k^{net} \quad (۴)$$

زمان انتقال اولیه با $\Delta_k^{net} = \frac{s_k}{\bar{B}}$ محاسبه می‌شود که $\bar{B}$ پهنای‌باند مؤثر مسیر است.

هر دستگاه اینترنت اشیا ابتدا بررسی می‌کند که آیا می‌تواند وظیفه را بدون ارسال به مه اجرا کند. شرط اجرای محلی به شکل $f_i^{cpu} \geq \gamma_k \cdot s_k$ بیان شده است. در صورت برقرار بودن شرط، زمان محاسبه محلی برابر با $T_{i,k}^{loc} = \frac{\gamma_k s_k}{f_i^{cpu}}$ می‌باشد.

برای محاسبه انرژی مصرفی از رابطه (۵) استفاده می‌شود:

$$E_{i,k}^{loc} = P_i^{cpu} \cdot T_{i,k}^{loc} \quad (۵)$$

که P_i^{cpu}توان پردازشی دستگاه اینترنت اشیا است. در صورتی که قابلیت اجرای وظیفه در دستگاه اینترنت اشیا وجود نداشته باشد، به این معنا که توان پردازشی محلی برای پردازش بار موردنظر کافی نباشد، وظیفه به نزدیک‌ترین گره مه ارسال می‌شود تا تحت مجموعه‌ای از محاسبات شبکه‌ای و پردازشی اجرا گردد. در ادامه، تمامی معیارهای عملکردی مربوط به اجرای وظایف در لایه مه ارائه شده و برای هر رابطه یک توضیح جامع و دقیق بیان می‌شود.

۱- تأخیر انتقال

برای محاسبه تاخیر انتقال از رابطه (۶) استفاده شده است.

$$T_{k,m}^{up} = \frac{s_k}{B_m^{up}} \quad (۶)$$

در این رابطه، مقدار $T_{k,m}^{up}$ مدت‌زمان لازم برای انتقال داده‌های ورودی وظیفه τ_kاز دستگاه مبدأ به گره مه شماره m را بیان می‌کند. پارامتر s_k نشان‌دهنده حجم داده‌های ورودی وظیفه بر حسب بایت یا کیلوبایت است و B_m^{up} پهنای‌باند رو به بالای در دسترس گره مه می‌باشد. نسبت این دو مقدار مشخص می‌کند که ارسال داده با چه سرعتی انجام می‌شود و چه مقدار زمان برای انتقال اولیه لازم است. پهنای‌باند کم یا داده‌های حجیم به‌طور مستقیم موجب افزایش این نوع تأخیر می‌شود.

۲- نرخ انتقال داده بر اساس شانون

به کمک رابطه (۷) R_m^{up} و R_m^{down} به ترتیب نرخ انتقال داده در مسیرهای ارسال و دریافت نشان داده می‌شود. این روابط برگرفته از ظرفیت کانال شانون هستند که بیان می‌کنند نرخ اطلاعات قابل انتقال تابعی از پهنای‌باند کانال و نسبت سیگنال به نویز است. در این روابط متغیر P_i^{tx} بیانگر توان ارسال دستگاه اینترنت اشیا و P_m بیانگر توان ارسال گره مه می‌باشد. متغیرهای η_m و ξ_m نشان دهنده‌ی شدت نویز کانال در مسیرهای آپلود و دانلود هستند. افزایش نسبت سیگنال به نویز، کیفیت کانال را بهبود داده و نرخ انتقال را افزایش می‌دهد. این معیار تأثیر مستقیم بر زمان لازم برای تبادل داده بین لایه‌ها دارد.

$$\begin{aligned} R_m^{up} &= B_m^{up} \log_2\left(1 + \frac{P_i^{tx}}{\eta_m}\right) \\ R_m^{down} &= B_m^{down} \log_2\left(1 + \frac{P_m}{\xi_m}\right) \end{aligned} \quad (۷)$$

۳- زمان پردازش

$$T_{k,m}^{exe} = \frac{\gamma_k s_k}{C_m^{cpu}} \quad (۸)$$

در رابطه (۸)، $T_{k,m}^{exe}$ زمان لازم برای انجام عملیات پردازشی وظیفه τ_k در گره مه m را اندازه‌گیری می‌کند. حاصل‌ضرب γ_k در s_k نشان‌دهنده حجم کل عملیات محاسباتی مورد نیاز (تعداد دستورالعمل‌ها) است، زیرا γ_k مقدار دستورالعمل به ازای هر بیت داده را بیان می‌کند. تقسیم این مقدار بر C_m^{cpu} که توان پردازشی گره مه برحسب MIPS است، زمان لازم برای اجرا را مشخص می‌سازد. هرچه توان پردازنده بیشتر باشد، مدت‌زمان اجرا کوتاه‌تر خواهد بود.

۴- زمان دانلود نتایج

زمان بازگشت نتیجه پردازش به دستگاه اینترنت اشیا به‌وسیله رابطه (۹) اندازه‌گیری می‌شود. پارامتر δ_k ضریب مقیاس‌دهی داده خروجی است و مشخص می‌کند که خروجی تولیدشده نسبت به ورودی چقدر کاهش یا افزایش دارد. اگر داده خروجی کوچکتر از ورودی باشد ($\delta_k < 1$) تأخیر دانلود به همان نسبت کاهش خواهد یافت. نرخ دانلود R_m^{down} نیز طبق فرم شانون محاسبه شده و تعیین‌کننده ظرفیت کانال بازگشتی است.

$$T_{k,m}^{down} = \frac{\delta_k s_k}{R_m^{down}} \quad (۹)$$

۵- انرژی مصرفی

با استفاده از رابطه (۱۰) سه مؤلفه انرژی محاسبه می‌شود: انرژی پردازش، انرژی ارسال و انرژی دریافت. انرژی اجرا $E_{k,m}^{exe}$ از حاصل‌ضرب توان مصرفی پردازنده گره مه در مدت‌زمان پردازش به دست می‌آید. انرژی انتقال آپلود متناسب با توان ارسال دستگاه اینترنت اشیا و زمان انتقال داده است، در حالی‌که انرژی دانلود تابع توان ارتباطی گره مه و زمان دانلود است. این سه مؤلفه مجموعاً مصرف انرژی لازم برای اجرای یک وظیفه را تشکیل می‌دهند. لازم به ذکر است انرژی مصرفی در زمان بیکاری[۱۸] در نظر گرفته نشده است و به همین دلیل، نتایج بهبود مصرف انرژی صرفاً مربوط به حالت فعال سیستم می‌باشد.

$$\begin{aligned} E_{k,m}^{exe} &= P_m \cdot T_{k,m}^{exe} \\ E_{k,m}^{up} &= P_i^{tx} \cdot T_{k,m}^{up} \\ E_{k,m}^{down} &= P_m \cdot T_{k,m}^{down} \end{aligned} \quad (۱۰)$$

۶- مجموع تأخیر

به کمک رابطه (۱۱)، کل زمان لازم برای پردازش یک وظیفه در لایه مه با جمع سه جزء اصلی زمان ارسال داده به مه، زمان پردازش و زمان دانلود نتایج تعیین می‌شود. این معیار تأثیر مستقیم بر کیفیت سرویس دارد و مبنای زمان‌بندی وظایف حساس به محدودیت تأخیر است.

$$T_{k,m}^{fog} = T_{k,m}^{up} + T_{k,m}^{exe} + T_{k,m}^{down} \quad (۱۱)$$

۷- مجموع انرژی

مجموع انرژی صرف‌شده برای پردازش یک وظیفه در لایه مه با جمع انرژی انتقال داده (آپلود و دانلود) و انرژی پردازشی به دست می‌آید. این معیار برای سیستم‌هایی که منابع انرژی محدود دارند بسیار حیاتی است و در طراحی الگوریتم‌های سبز (Energy-aware) به‌طور جدی لحاظ می‌شود.

$$E_{k,m}^{fog} = E_{k,m}^{up} + E_{k,m}^{exe} + E_{k,m}^{down} \quad (۱۲)$$

۸- بار گره مه

بار گره مه شاخصی از میزان ترافیک پردازشی وارد شده به هر گره به کمک رابطه (۱۳) تعیین می‌شود. صورت کسر مجموع عملیات موردنیاز تمام وظایفی است که به گره موردنظر تخصیص یافته‌اند. مخرج نیز توان پردازشی گره را نشان می‌دهد. اگر مقدار بار بزرگ باشد، گره دچار ازدحام شده و تأخیر پردازش افزایش می‌یابد؛ بنابراین این معیار برای تصمیم‌گیری در زمان‌بندی و offloading نقش کلیدی دارد.

$$L_m = \frac{\sum_{\tau_k \to g_m} (\gamma_k s_k)}{C_m^{cpu}} \quad (۱۳)$$

یکی از معیارهای شناخته‌شده و پرکاربرد در زمینه زمان‌بندی کارها، معیار تعادل بارکاری ماشین‌های مجازی است و به منظور ارزیابی آن می توان از شاخص عدالت جین[۱۹] استفاده کرد که مقدار آن بین ۰ و ۱ قرار دارد و هرچه این عدد به ۱ نزدیک‌تر باشد، تعادل بار بیشتری میان ماشین‌های مجازی برقرار است.

لازم به ذکر است که در این گام به‌منظور ارزیابی عدالت در توزیع بار بین n ماشین مجازی، شاخص جین از طریق فرمول (۱۴) محاسبه می‌گردد:

$$\text{jain} = \frac{(\sum_{i \in m} \text{LB}(VM_i))^2}{m^2 \times \sum_{i \in m} \text{LB}(VM_i)} \quad (۱۴)$$

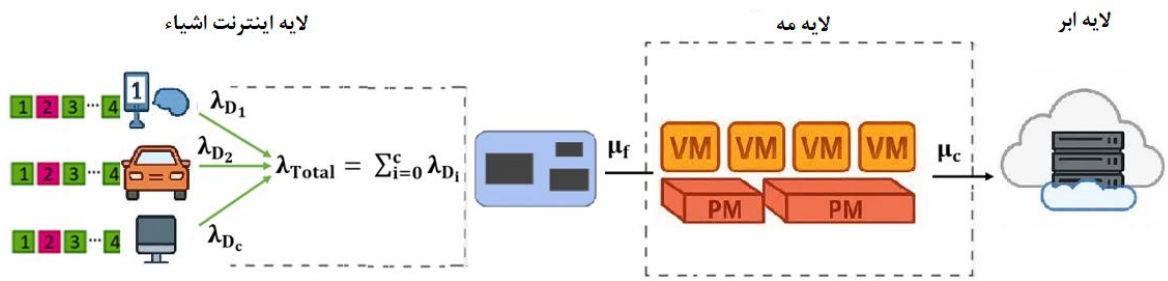
در این رابطه m بیانگر تعداد ماشین مجازی و $\sum_{i \in m} \text{LB}(VM_i)$ بیانگر مجموع مقدار تعادل بارکاری در ماشین‌های مجازی است. باید توجه داشت که مقدار $jain$ در بازه $[0,1]$ قرار دارد. مقدار $J = 1$ بیانگر تعادل کامل بار میان تمام ماشین‌های مجازی است، در حالی‌که نزدیک‌شدن آن به صفر نشان‌دهنده وجود ناهماهنگی شدید در تخصیص بارها می‌باشد. در جدول ۲ فهرست نمادها و پارامترهای مورد استفاده و در شکل ۲ فرآیند اجرای وظیفه در مدل پیشنهادی آورده شده است.

جدول ۲: فهرست نمادها و پارامترهای مورد استفاده در مدل پیشنهادی

نماد	توضیح	واحد
	پهنای باند تخصیص‌یافته به گره مه (k)	Mbps
	ضریب مقیاس‌دهی داده‌ها در انتقال میان لایه‌ها	-
	چگالی محاسباتی (تعداد دستورالعمل به ازای بیت داده ورودی)	Mbps/bit
	متغیر دودویی تخصیص وظیفه (k) به گره (m) (۱ در صورت تخصیص، ۰ در غیر این صورت)	{۱و۰}
	اندازه داده ورودی وظیفه (k)	MB
	اندازه داده خروجی وظیفه (k)	وظیفه بر ثانیه
	نرخ تولید وظایف توسط دستگاه اینترنت اشیا (i)	Watt

۴- الگوریتم پیشنهادی

در این مطالعه، الگوریتم پیشنهادی IJFSA-ODA با هدف بهینه‌سازی زمان‌بندی وظایف و توزیع‌بار ارائه شده است. نوآوری اصلی این رویکرد با الهام گرفتن از دو تکنیک تکاملی و مبتنی بر هوش جمعی شامل الگوریتم‌های DE و ABC و توسعه و بهبود



شکل ۲: فرآیند اجرای وظیفه طبق روش معرفی شده

الگوریتم IJFSA نهفته است. برخلاف ترکیب‌های مرسوم مانند PSO–ACO یا GA–PSO که در مطالعات قبلی استفاده شده و اغلب با مشکل گرفتار شدن در بهینه‌های محلی و عدم تعادل بار مواجه می‌شوند، IJFSA-ODA با مکانیزم حرکت فعال و غیرفعال عروس دریایی و استفاده از عملگر پیشنهادی SA[۲۰] ضمن حفظ تنوع جمعیت، توانایی فرار از بهینه‌های محلی را دارد و زمان اجرا و عدم تعادل‌بار را کاهش می‌دهد. این الگوریتم همچنین تخصیص تطبیقی وظایف و توزیع بار آگاه از انرژی را فراهم می‌کند و پایبندی به محدودیت‌های سخت‌افزاری سیستم را تضمین می‌کند.

در مجموع، IJFSA-ODA یک چارچوب چندهدفه و نوآورانه ارائه می‌دهد که قابلیت انطباق بالا و عملکرد پایدار در مواجهه با شرایط پویا و بارکاری نامتوازن را تضمین می‌کند. برای افزایش کیفیت جستجو در سه فاز اصلی این الگوریتم، عملگرهای تکاملی و فراابتکاری به کار گرفته شده‌اند: در فاز گروهی اولیه به جای مکانیزم معمولی IJFSA از عملگر الهام گرفته از الگوریتم DE استفاده می‌شود تا تنوع جمعیت اولیه تضمین شده و فضای جستجو به صورت گسترده‌تر پوشش داده شود. عملگر پیشنهادی با استفاده از اختلاف بین دو زمان‌بندی متفاوت، یک اختلال کنترل‌شده در

زمان‌بندی فعلی ایجاد می‌کند تا از تکرار الگوهای مشابه جلوگیری شود. این مکانیزم باعث تغییر هم‌زمان جایگاه چند وظیفه در زمان‌بندی می‌شود و در نتیجه، برنامه‌های زمان‌بندی متنوع‌تری تولید می‌کند که منجر به توزیع بار متعادل‌تر در نودهای مه می‌گردد. در فاز حرکت غیرفعال، از یک سازوکار الهام‌گرفته از ABC استفاده می‌شود تا فرآیند بهره‌برداری با دقت بیشتری هدایت شده و راه‌حل‌ها به‌صورت تدریجی به سمت نواحی امیدبخش فضای جستجو حرکت کنند. در فاز حرکت فعال، عملگر الهام گرفته شده از الگوریتم SA به‌کار گرفته می‌شود تا با به‌روزرسانی تدریجی و پذیرش احتمالی راه‌حل‌های ضعیف، از گرفتار شدن در بهینه‌های محلی جلوگیری گردد. تمرکز روش پیشنهادی بر تعادل بار کاری ماشین‌های مجازی و بهینه‌سازی عملکرد سیستم می‌باشد.

۴-۱- تولید جمعیت اولیه

در آغاز، پس از تعیین پارامترها، جمعیت اولیه با استفاده از نگاشت لجستیک مبتنی بر نظریه آشوب (۱۴) تولید می‌شود. این رویکرد به‌جای روش‌های سنتی، از یک مکانیزم تصادفی کنترل‌شده بهره می‌گیرد. سپس وظایف بر اساس سطح اولویت دستگاه‌ها سازمان‌دهی می‌شوند.

$$x_{k+1} = a x_k (1 - x_k) \qquad (۱۴)$$

در این روش، طول آرایه برابر با تعداد کل وظایف است و هر عروس دریایی نمایانگر یک راه‌حل بالقوه برای مسئله محسوب می‌شود. توجه شود که انجام وظایف بر اساس ترتیب مشخص‌شده در آرایه انجام می‌گیرد. به عبارت دیگر، شاخص آرایه نشان‌دهنده اولویت اجرای وظایف و تعداد کل آن‌هاست. از آنجا که هر عروس دریایی یک مختصات در فضای جستجو را نمایش می‌دهد که بیانگر یک تخصیص ممکن از وظایف میان سلول‌های مه است. موقعیت آن را می‌توان به شکل یک آرایه نشان داد (شکل ۳).

x_1	x_2	x_3	...	x_{n-1}	x_n

شکل ۳: موقعیت هر عروس دریایی در فضای پاسخ

شکل ۳ موقعیت هر عروس دریایی در فضای پاسخ را نشان می‌دهد که متناظر با یک پیکربندی احتمالی تخصیص وظایف است. عناصر آرایه x_1 تا x_n نمایانگر شناسه‌های گره‌های مه هستند و وظایف بر اساس مقادیر موجود در آرایه به این گره‌ها اختصاص داده می‌شوند.

۴-۲- مکانیزم کنترل زمان و جریان‌های دریایی

عروس‌های دریایی از مکانیزم «کنترل زمان» بهره می‌برند که به آن‌ها امکان می‌دهد بین انواع حرکت جابجا شوند. در مراحل اولیه، تجمع در نواحی غنی از منابع غذایی شکل می‌گیرد و با تغییر شرایط محیطی، خوشه‌های جدید ایجاد می‌شوند. حرکت آن‌ها به دو حالت نوع A (غیرفعال) و نوع B (فعال) تقسیم می‌شود که با گذشت زمان فعالیت نوع B غالب می‌شود. این مکانیزم با تولید مقادیر تصادفی بین ۰ و ۱، الگویی برای شبیه‌سازی رفتار در الگوریتم‌های IJFSA، DE و ABC فراهم می‌کند.

۴-۲-۱- حرکت گروهی

در این قسمت از عملگر الهام گرفته از الگوریتم DE استفاده می‌شود تا پراکندگی راه‌حل‌ها در فضای جستجو بیشتر شود. عملگر جهش DE به صورت زیر تعریف می‌شود:

$$X_{mutation} = X_{r1} + f.(X_{r2} - X_{r3}) \qquad (۱۵)$$

که در آن X_{r1}، X_{r2} و X_{r3} سه راه‌حل تصادفی و مستقل از جمعیت هستند و F ضریب تقویت جهش است. این مرحله باعث می‌شود تنوع بالا برود و الگوریتم از همان ابتدا توان اکتشاف بیشتری داشته باشد.

۴-۲-۲- حرکت فعال

در این حالت، موقعیت عروس‌های دریایی در مدل حرکتی به‌صورت آرایه‌ای به‌روزرسانی می‌شود و از تکنیک SA برای حرکت تدریجی از یک راه‌حل اولیه به راه‌حل‌های مجاور استفاده می‌شود. در هر گام، راه‌حل بهتر مجاور به‌عنوان راه‌حل فعلی پذیرفته می‌شود و در صورت بدتر بودن، با احتمال exp(-ΔE/T) قابل قبول است. مکانیزم کنترل زمان در الگوریتم عروس دریایی این حرکت را مدیریت می‌کند؛ در ابتدا یک راه‌حل مجاور بهبود یافته انتخاب می‌شود و با گذشت زمان احتمال پذیرش کاهش می‌یابد. نیمی از شاخص‌های آرایه به‌صورت تصادفی انتخاب شده، ماشین‌های مجازی مرتبط مشخص و آرایه با بهترین راه‌حل تصادفی به‌روزرسانی می‌شود.

۴-۲-۳- حرکت غیرفعال

در فاز حرکت غیرفعال، الگوریتم بر بهره‌برداری از نواحی امیدبخش فضای جستجو تمرکز دارد. در روش پیشنهادی، به‌جای اجرای کامل الگوریتم ABC، از یک مکانیزم جستجوی محلی الهام‌گرفته از رفتار زنبورهای کارگر در ABC استفاده شده است. در این مکانیزم، هر راه‌حل مشابه یک منبع غذایی در نظر گرفته شده و با ایجاد تغییرات کنترل‌شده در اطراف آن، کیفیت راه‌حل به‌صورت تدریجی بهبود می‌یابد .به‌طور مشخص، برای هر راه‌حل یک همسایه محلی با اعمال یک اغتشاش کوچک و جهت‌دار بر اساس تفاوت آن با سایر راه‌حل‌های موجود در جمعیت ایجاد شده و در صورت بهبود مقدار تابع هدف، جایگزینی انجام می‌گیرد .این فرآیند صرفاً نقش هدایت بهره‌برداری را ایفا کرده و بدون پیاده‌سازی فازهای ناظر و پیشاهنگ ABC، موجب کاهش پیچیدگی محاسباتی الگوریتم می‌شود.

۴-۳- تابع هدف و انتخاب راه‌حل

در مسائل بهینه سازی چندهدفه، تضاد بین توابع هدف مختلف که به‌طور همزمان باید حداکثر یا حداقل شـوند، چالش ایجاد می‌کند. پیشـرفت در یک هدف معمولاً باعث کاهش عملکرد در هدف دیگر می‌شـود و به همین دلیل مجموعه‌ای از راه‌حل‌های بهینه، موسـوم به «جبهه پارتو»، شکل می‌گیرد. این جبهه شامل راه‌حل‌هایی است که در فضای مسئله توزیع شده و توسط هیچ راه‌حل دیگری پشت سـر گذاشـته نمی‌شـوند. نقاط مرجع به‌صـورت دقیق روی یک ابرصـفحه‌ی نرمال‌شـده با m-1 بعد قرار می‌گیرند و به p بخش مساوی تقسیم می‌شوند. در این پژوهش، نرمال‌سازی و وزن‌دهی به هر هدف برای کمک به انتخاب مناسـب‌ترین راه‌حل از جبهه پارتو انجام شـده، با این توجه که فرآیند انتخاب با تولید جبهه متفاوت است.

$$\min(\alpha * Normalize(ResponseTime) + \beta * Normalize(Energy)) \quad (۱۶)$$

در این مدل، α و β به‌عنوان پارامترهای اهمیت هر هدف در نظر گرفته می‌شوند. با انجام مرحله نرمال‌سازی، داده‌ها بازمقیاس می‌شوند تا اطمینان حاصل شود که هر هدف تأثیر مشابهی بر فرآیند بهینه‌سازی دارد.

$$Normalize(x, x_i) = abs(mean(x) - x_i)/std(x) \quad (۱۷)$$

۴-۴- آزمون همگرایی

الگوریتم زمانی خاتمه می‌یابد که تفاوت بین بهترین راه‌حل‌ها در تکرارهای متوالی کمتر از یک آستانه مشخص باشد یا حد مجاز تعداد تکرارها رسیدگی شود. روش پیشنهادی زمان‌بندی وظایف، ترکیبی از الگوریتم IJFSA با الگوریتم‌های ABC و DE است که با عنوان IJFSA-ODA شناخته می‌شود. در چارچوب پیشنهادی، برای بهبود کیفیت جستجو در فازهای مختلف الگوریتم، از ترکیب عملگرهای تکاملی و فراابتکاری استفاده شده است. این رویکرد با ایجاد تعادل میان جستجوی سراسری و بهره‌برداری محلی، موجب بهبود توزیع بار، تخصیص کارآمد منابع و کاهش زمان تکمیل وظایف در ماشین‌های مجازی می‌شود. در این فرآیند، راه‌حل‌های کاندیدا با استفاده از تابع هدف ارزیابی شده و به‌صورت تدریجی به سمت پیکربندی‌های کاراتر و مناسب‌تر هدایت می‌شوند.

در این چارچوب، برای افزایش کیفیت جستجو در سه فاز اصلی این الگوریتم، عملگرهای تکاملی و فراابتکاری به کار گرفته شده‌اند. این ترکیب سه‌گانه موجب می‌شود الگوریتم پیشنهادی هم از توانایی جستجوی سراسری قوی بهره ببرد و هم از قابلیت‌های بهره‌برداری محلی مؤثر، که نهایتاً منجر به تعادل بار کاری بهتر در ماشین‌های مجازی و کاهش چشمگیر زمان تکمیل وظایف می‌گردد. در این روش هر ذره با دو بردار تصادفی موقعیت و سرعت خود را تعیین می‌کند. راه‌حل‌های پیشنهادی که نحوه تخصیص منابع به درخواست‌های کاربران را نمایش می‌دهند، با استفاده از تابع هدف ارزیابی می‌شوند. سپس مقدار تناسب کلی و موقعیت ذره متناظر به‌روزرسانی می‌گردد. این فرآیند موجب بهبود توزیع بار در سیستم شده و تعادل مناسبی میان گره‌ها ایجاد می‌کند. با ورود هر وظیفه، میزان بار مطلوب هر گره با هدف کاهش زمان پردازش و مصرف انرژی محاسبه می‌شود.

لازم به ذکر است که در مدل پیشنهادی IJFSA-ODA، بخش‌های تکمیلی شامل الگوریتم‌های DE و ABC به‌صورت موضعی و در فازهای مشخصی از فرآیند اجرا مورد استفاده قرار گرفته‌اند، نه در

تمام تکرارها یا کل جمعیت. درواقع، الگوریتم ABC تنها در فاز حرکت غیرفعال و صرفاً برای زیرمجموعه‌ای کوچک از راه‌حل‌ها (به‌صورت نمونه‌برداری محدود) اعمال می‌شود تا بهره‌برداری از نواحی امیدبخش با دقت بیشتری انجام گیرد. این طراحی هدفمند باعث می‌شود محاسبات اضافی به حداقل برسد و بخش عمده بار محاسباتی بر دوش الگوریتم پایه (IJFSA) باقی بماند. از نظر زمانی، اجرای موضعی ABC تأثیر قابل‌توجهی بر پیچیدگی کلی ندارد، زیرا فراخوانی آن در هر چرخه به‌صورت تطبیقی و بر اساس آستانه‌ی بهبود تعریف‌شده انجام می‌شود. باید توجه داشت که از دیدگاه نظری، پیچیدگی محاسباتی کل الگوریتم IJFSA-ODA تقریباً در حد $O(N \times T)$ باقی می‌ماند که N اندازه جمعیت و T تعداد تکرارهاست؛ اضافه‌شدن عملگرهای DE و ABC صرفاً منجر به افزایش ضرایب ثابت در این مرتبه شده و تغییری در رشد کلی پیچیدگی ایجاد نمی‌کند. از آنجا که این عملگرها به‌صورت انتخابی و بدون ساختارهای بازگشتی پیچیده اجرا می‌شوند، از افزایش غیرضروری بار محاسباتی جلوگیری شده است. در نتیجه، الگوریتم پیشنهادی ضمن حفظ کیفیت همگرایی و بهبود تعادل بار، کارایی مناسبی را در محیط‌های پویا و توزیع‌شده ارائه می‌دهد.

۵- تحلیل عددی

کارایی روش پیشنهادی زمان‌بندی وظایف با الگوریتم‌های IJFA [۲۵]، IJFSA، PSO-SA [۲۶] و PSO [۲۷] مقایسه شد. معیارهای ارزیابی شامل زمان اجرا، مصرف انرژی و تأخیر بود تا بهبود عملکرد روش پیشنهادی مشخص شود.

۵-۱- تنظیمات پارامترها

برای ارزیابی روش پیشنهادی در محیط اینترنت اشیا–مه–ابر از پلتفرم IoTSim-Osmosis استفاده شد. ساختار شبیه‌سازی شامل سه لایه‌ی اینترنت اشیا، مه و ابر است. در لایه اینترنت اشیا، سنسورهای متنوعی برای جمع‌آوری داده‌ها نصب شده‌اند. داده‌ها ابتدا به لایه مه منتقل و پردازش می‌شوند و سپس به لایه ابر ارسال می‌شوند که به‌صورت یک مرکز داده متصل به اینترنت شبیه‌سازی شده است. هر گره مه مجهز به ماشین‌های مجازی برای اجرای وظایف است. در این سیستم هوشمند سلامت، دستگاه‌های متصل به اینترنت امکان نظارت از راه دور بیماران را فراهم می‌کنند. این دستگاه‌ها شامل پوشیدنی‌ها و ابزارهای ثبت علائم حیاتی هستند که اطلاعات را به متخصصان پزشکی منتقل می‌کنند. اکوسیستم درنظر گرفته شده، شامل فروشنده خدمات، ارائه‌دهنده شبکه، تأمین‌کننده تجهیزات و پزشکان است که سلامت بیماران را پایش می‌کنند. اینترنت اشیا نقش اصلی در تولید و تحلیل داده‌ها دارد و با توجه به محدودیت‌های سیستم‌های سنتی در پشتیبانی از قابلیت‌های پیشرفته اینترنت اشیا، این مطالعه یک سیستم سلامت هوشمند مبتنی بر اینترنت اشیا را معرفی می‌کند که با استفاده از چارچوب زمان‌بندی وظایف پیشرفته امکان رصد علائم حیاتی بیماران را فراهم می‌آورد.وظایف این سیستم دارای زمان‌بندی مشخص، داده‌های مرتبط و نیاز پردازشی CPU هستند و شامل سه نوع وظیفه اصلی می‌شوند، همان‌طور که در شکل ۴ نشان داده شده است.

الگوریتم IJFSA-ODA

ورودی: وظایف، ماشین‌های مجازی، حداکثر تکرار، اندازه جمعیت (**Npop**)، **خروجی:** تخصیص بهینه وظایف

مراحل:

۱. استفاده از نقشه لجستیک برای مقداردهی اولیه جمعیت X_pop // (X_pop) جمعیت اولیه است

۲. محاسبه تابع برازندگی هر راه‌حل

۳. اجرای جبهه پارتو (یافتن راه‌حل‌های غیرغالب)

۴. برای t=1 تا حداکثر_تعداد_تکرارها مراحل ۵ تا ۱۶ را انجام بده کن

۵. برای i=1 تا Npop مراحل ۵ تا ۱۵ را انجام بده کن // (Npop) اندازه جمعیت است

۶. محاسبه c_t= (1-(itrer/max_itrer))*(2*Math.random()-1); // مکانیزم کنترل زمان اجرا

۷. اگر (c_t >= 0.5)

۸. اجرای عملگر الهام گرفته ازالگوریتم DE //جریان اقیانوس

۹. وگرنه اگر (Math.random() > (1- c_t))

۱۰. اصلاح راه‌حل‌های کاندید با استفاده از مکانیزم الهام گرفته از الگوریتم ABC //حرکت غیرفعال

۱۱. وگرنه کاوش راه‌حل‌ها با استفاده از اعمال عملگر الهام گرفته از الگوریتم SA //حرکت فعال

۱۲. محاسبه تابع برازندگی هر راه‌حل

۱۳. اجرای جبهه پارتو (یافتن راه‌حل‌های غیرغالب)

۱۴. بازگشت X_α // بهترین راه‌حل

شکل ۴: الگوریتم پیشنهادی IJFSA-ODA برای زمان‌بندی وظایف و توزیع بار

در این پژوهش، وظایفی که محدودیت تأخیر سخت‌گیرانه دارند، به گره‌های لبه‌ای اختصاص می‌یابند، به شرطی که مجموع بارکاری از ظرفیت پردازشی آن‌ها فراتر نرود. روش زمان‌بندی به‌صورت گام‌به‌گام وظایف را میان گره‌های پردازشی موجود تقسیم می‌کند تا استفاده متعادل و کارآمد از منابع حفظ شود. برای ارزیابی تجربی، از چندین مجموعه داده از جمله Cloud–Fog Computing Dataset، یک مجموعه داده بزرگ شامل حدود ۵۰ هزار درخواست و تقریباً ۱٫۳ میلیون وظیفه استفاده شده است.

داده‌های تولیدشده توسط هر سنسور بسته به نوع وظیفه بین ۲۵۰ تا ۱۰۲۴ کیلوبایت متغیر است و حداکثر تأخیر مجاز برابر با ۱۰۰ میلی‌ثانیه تعیین شده است. لایه مه شامل ۸ تا ۲۰ گره مه است، که هر کدام دارای حافظه، توان پردازشی و پهنای باند مستقل می‌باشند. هر وظیفه با مشخصاتی مانند حجم داده‌های ورودی و خروجی، نیاز حافظه و تعداد کل دستورالعمل‌های لازم تعریف می‌شود. جدول ۳ مشخصات گره‌های مه شامل عملکرد CPU برحسب MIPS، هزینه حافظه، مصرف CPU و هزینه پهنای باند را ارائه می‌دهد

در شکل ۵ جریان کاری وظایف سیستم سلامت هوشمند آورده شده است که ترتیب اجرای وظایف را به صورت شماتیک نشان می‌دهد و وابستگی‌های DAG در مدل ما صدق نمی‌کند .ترتیب‌ها صرفاً برای نمایش جریان کلی کار است. شبیه‌سازی‌ها روی سیستمی با پردازنده Intel Core i8 با فرکانس ۲٫۷۰ گیگاهرتز و حافظه ۸ گیگابایت انجام شد. تنظیمات پارامترها در جدول ۴ ارائه شده و بر اساس آزمایش‌های اولیه تعیین گردید. در تمامی سناریوها، اندازه جمعیت و تعداد تکرارها به ترتیب برابر با ۱۰۰ و ۱۰ در نظر گرفته شد.

۵-۲- نتایج و یافته‌های آزمایش

در این بخش به‌منظور حفظ انسجام مفهومی، پردازش صرفاً در لایه مه انجام شده و منابع پردازشی این لایه به‌صورت ماشین‌های مجازی مدل‌سازی شده‌اند. شکل‌های ۶ تا ۱۰ نتایج حاصل از ده بار اجرای شبیه‌سازی را برای چهار الگوریتم مختلف، در شرایط گوناگون را نمایش می‌دهند. شاخص‌های ارزیابی شامل زمان اجرا، مصرف انرژی و درصد وظایف انجام‌شده در مهلت مقرر می‌باشند. نتایج نشان می‌دهد که الگوریتم پیشنهادی IJFSA-ODA در تمامی سناریوها عملکرد بهتری نسبت به الگوریتم‌های مقایسه‌ای شامل IJFSA، IJFA، PSO-SA و PSO دارد. در بخش بعدی، تحلیل دقیق‌تری از تغییرات عملکرد بر اساس حجم وظایف ارائه می‌شود و برای اطمینان از پایداری و کارایی روش پیشنهادی، شبیه‌سازی‌هایی با نرخ‌های ورودی متفاوت نیز انجام شده است.

جدول ۳: مشخصات گره‌های مه

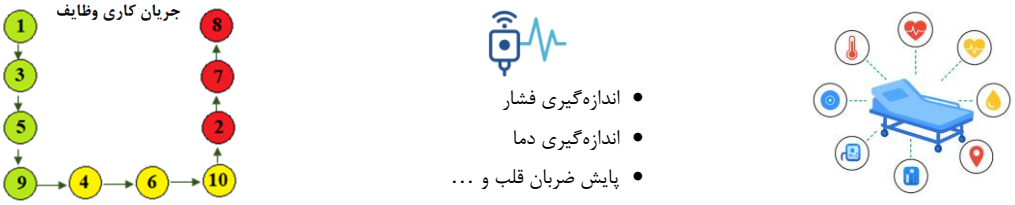
پارامتر	نماد	مقدار
فضای دیسک CPU	CPU_{DS}	۱۰۰۰ – ۵۰۰
حافظه CPU	CPU_{mem}	۱۰۰۰ – ۵۰۰
هزینه پهنای باند	BW_{Cost}	۰/۰۷۹۹ – ۰/۰۱۰۵
هزینه حافظه	$Memory_{Cost}$	۰/۰۴۷۷۴ – ۰/۰۱۰۷۸
هزینه مصرف CPU	CPU_{Cost}	۰/۹۸۴۹ – ۰/۱۰۰۶
سرعت CPU (MIPS)	CR_{MIPS}	۴۹۵۲ – ۶۵۴

جدول ۴: پارامترهای آزمایش‌های انجام شده

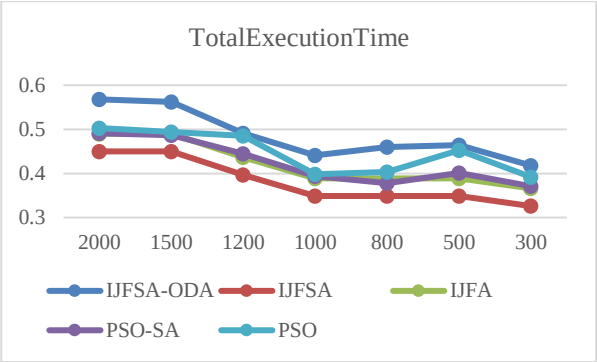
کاربرد	مقدار	نماد
حجم خروجی دستگاه اینترنت اشیا (MB)	۲۰-۱۰۰۰	$t_i^{IoTDevice_s}$
تعداد گره‌ها	۲۵-۱۰۰	F
اولویت برنامه	۱-۳	$A_i^{Priority}$
حافظه مورد نیاز (MB)	۵۰-۲۰۰	t_i^{memory}
تعداد دستورات	۱۰۰-۱۰۰۰	$t_i^{Num_i}$
مهلت انجام (s)	۱۰۰-۵۰۰	$t_i^{deadline}$
توقف تولید داده (s)	۸-۲۰	$Stop_{Di}$

۵-۲-۱- مدت زمان اجرا

همان‌طور که در شکل ۶ دیده می‌شود، با افزایش تعداد وظایف از ۳۰۰ تا ۲۰۰۰، زمان اجرای تمام الگوریتم‌ها افزایشی است. برای مثال، زمان اجرای IJFSA-ODA از ۰٫۴۱۸ ثانیه به ۰٫۵۶۸ ثانیه می‌رسد، در حالی که IJFSA و IJFA به‌ترتیب از ۰٫۳۲۶ به ۰٫۴۵۰ و از ۰٫۳۶۶ به ۰٫۴۹۰ ثانیه افزایش می‌یابند. الگوریتم پیشنهادی به‌طور کلی زمان اجرای رقابتی و قابل قبول را حفظ می‌کند و معمولاً



شکل ۵: جریان کاری وظایف سیستم سلامت هوشمند



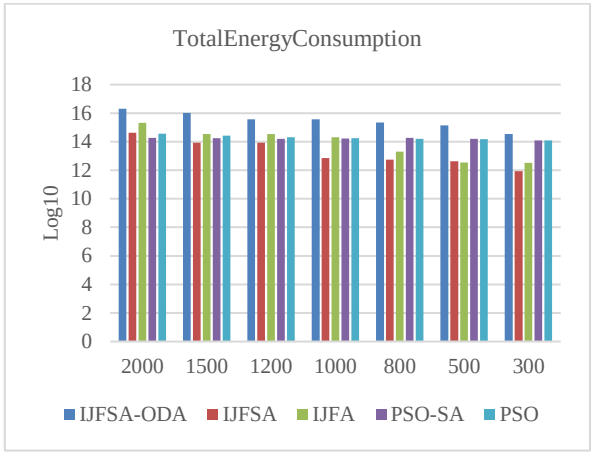
شکل ۶: زمان اجرای کل با افزایش تعداد وظایف (VM=۲۵)

سریع‌تر از IJFA و PSO است، اگرچه برای برخی اندازه‌ها مانند ۱۲۰۰ و ۱۵۰۰ وظیفه، زمان آن کمی بیشتر است که ناشی از محاسبات اضافی توازن بار و بهینه‌سازی با آگاهی از انرژی است.

به‌طور کلی، IJFSA-ODA توزیع مؤثرتر وظایف و بهره‌وری انرژی بالاتر را بدون افزایش چشمگیر زمان اجرا فراهم می‌کند.

۵-۲-۲- میزان مصرف انرژی

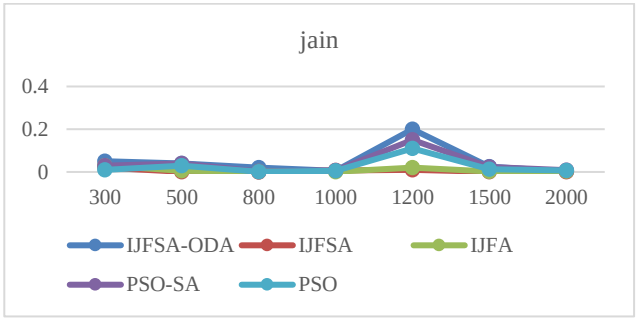
در این بخش، مصرف انرژی الگوریتم‌های IJFSA-ODA، IJFSA، IJFA، PSO-SA و PSO تحت بارهای کاری مختلف و تعداد متفاوت گره‌ها بررسی شده است. همان‌طور که در شکل ۷ مشاهده می‌شود، مصرف انرژی IJFSA-ODA در مقایسه با PSO و PSO-SA برای برخی اندازه‌های وظایف مانند ۳۰۰، ۵۰۰ و ۸۰۰ وظیفه، کمی بالاتر است. نرخ افزایش مصرف انرژی IJFSA-ODA با افزایش تعداد وظایف نسبت به دیگر الگوریتم‌ها کندتر است که نشان‌دهنده مقیاس‌پذیری بهتر انرژی و مدیریت تخصیص وظایف می‌باشد. این موازنه باعث می‌شود که الگوریتم گاهی پیک‌های گذرا در مصرف انرژی داشته باشد، اما در مجموع بهره‌وری انرژی برای بارهای کاری بزرگ بهبود یافته است و اثربخشی استراتژی بهینه‌سازی هیبریدی پیشنهادی را تأیید می‌کند.



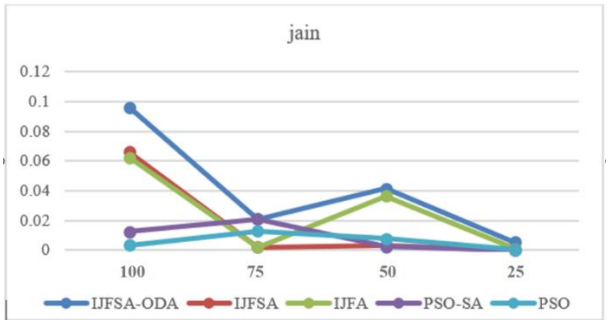
شکل ۷: میزان مصرف انرژی با افزایش تعداد وظایف (VM= ۲۵)

۵-۲-۳- تعادل بارکاری ماشین‌های مجازی

این بخش به عملکرد توازن بار الگوریتم پیشنهادی با استفاده از شاخص عدالت جین پرداخته است، که میزان یکنواختی توزیع بار میان ماشین‌های مجازی را نشان می‌دهد. نتایج در شکل‌های ۸ و ۹ آورده شده است. در سناریو با ۲۵ ماشین مجازی و تعداد وظایف متغیر از ۳۰۰ تا ۲۰۰۰، الگوریتم IJFSA-ODA به‌طور مداوم بالاترین مقادیر عدالت را کسب می‌کند، که نشان‌دهنده توزیع مؤثر وظایف حتی با افزایش بار کاری است. در سناریوی دیگری که تعداد وظایف برابر ۲۰۰۰ و تعداد ماشین‌های مجازی از ۲۵ تا ۱۰۰ تغییر می‌کند، افزایش تعدادVMها تقریبا در همه الگوریتم‌ها موجب بهبود توزیع بار می‌شود. با این حال، IJFSA-ODA همچنان بالاترین میزان عدالت را در تمامی تعداد VMها حفظ می‌کند. برتری IJFSA-ODA ناشی از ادغام بهینه‌سازی پویا است که هم پایداری و هم عدالت توزیع بار را بهبود می‌بخشد.



شکل ۸: نتایج شاخص عدالت جین برای ۲۵ ماشین‌مجازی وتعداد وظایف متعدد



شکل ۹: نتایج شاخص عدالت جین برای ۲۰۰۰ وظایف با تعداد ماشین مجازی متفاوت

۵-۲-۴- درصد وظایف انجام‌شده در مهلت مقرر

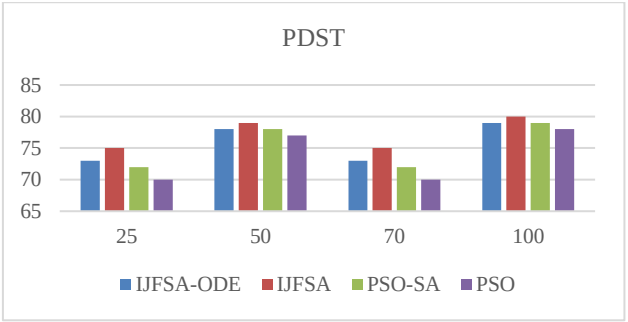
در این بخش، درصد وظایف انجام‌شده در مهلت مقرر برای پنج الگوریتم بررسی شد. همان‌طور که در شکل ۱۰ آورده شده است، نتایج نشان می‌دهد که IJFSA-ODA عملکرد پایدار و قابل اطمینانی در رعایت مهلت‌ها دارد و درصد وظایف انجام‌شده در مهلت مقرر آن کمی کمتر از IJFSA است که ناشی از توزیع یکنواخت‌تر بار بین گره‌ها و تأخیر مختصر برخی وظایف است. در سناریوی با ۲۵ VM و ۳۰۰ تا ۲۰۰۰ وظیفه، IJFSA-ODA رعایت مهلت‌ها را پایدار حفظ و توزیع بارکاری را بهبود بخشید. در سناریوی دوم با ۲۰۰۰ وظیفه و تعداد VMهای متغیر، این الگوریتم بالاترین درصد وظایف انجام‌شده در مهلت مقرر را حفظ کرد و مقیاس‌پذیری و مقاومت آن را نشان داد. بنابراین، استراتژی توزیع وظایف پیشنهادی تعادلی میان رعایت مهلت‌ها و استفاده از منابع ایجاد می‌کند و یک trade-offهوشمندانه بین عملکرد زمانی و بهره‌وری انرژی ارائه می‌دهد.

لازم به ذکر است که هر سناریو ده بار شبیه‌سازی و نتایج تحت سطح اطمینان ۹۵٪ در جدول ۵ آورده شده است. به‌منظور بررسی معناداری آماری تفاوت عملکرد الگوریتم‌ها، آزمون فردمن[۲۱] به‌صورت جداگانه برای هر معیار ارزیابی و بر روی نتایج حاصل از ۱۰ اجرای مستقل اعمال شد. بر اساس آزمون فردمن، تفاوت‌های مشاهده‌شده در معیارهای مصرف انرژی، تعادل بار، زمان اجرا و درصد وظایف انجام‌شده در مهلت مقرر از نظر آماری معنادار بوده‌اند (p-value < 0.05)، با این حال رتبه‌بندی الگوریتم‌ها در این آزمون لزوماً منطبق با مقادیر میانگین گزارش‌شده در جدول ۵ نیست. نتایج جدول ۵ نشان می‌دهد که هیچ الگوریتمی در تمامی معیارها به‌طور هم‌زمان برتری مطلق ندارد و عملکرد الگوریتم پیشنهادی پیشنهادی بیانگر وجود یک trade-off منطقی میان زمان اجرا، بهره‌وری انرژی و کیفیت توازن بار می‌باشد.

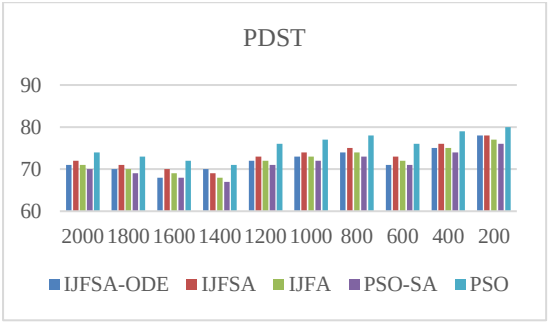
۶- نتیجه‌گیری و کارهای آتی

الگوریتم IJFSA-ODA با بهره‌گیری از مکانیزم‌های فعال و غیرفعال و الهام از الگوریتم‌های تکاملی و هوش جمعی، تلاش می‌کند تا زمان پردازش، مصرف انرژی و تعادل بار را بهبود دهد. استفاده از عملگرهی الهام گرفته شده به الگوریتم این امکان را می‌دهد که از بهینه‌های محلی اجتناب کند و تعادل مناسبی بین جستجوی سراسری و بهره‌برداری محلی برقرار کند. نتایج شبیه‌سازی نشان می‌دهد که IJFSA-ODA به‌طور مؤثری توازن مناسبی میان این شاخص‌های عملکردی برقرار می‌کند. اگرچه زمان اجرای کل نسبت به برخی الگوریتم‌های پایه کمی بیشتر است، که ناشی از سربار ناشی از مکانیزم‌های متعادل‌سازی بار و مدیریت انرژی است، الگوریتم به‌طور مداوم توزیع عادلانه بار کاری، مقیاس‌پذیری انرژی بهبود یافته و عملکرد پایدار در رعایت مهلت‌های وظایف را تضمین می‌کند. با وجود بهبودهای مشاهده‌شده، الگوریتم IJFSA-ODA محدودیت‌هایی دارد.

عملکرد آن عمدتاً برای مجموعه‌ای مشخص از وظایف و منابع ارزیابی شده و برای سیستم‌های کاملاً پویا که وظایف به‌صورت پیوسته وارد می‌شوند نیازمند سازوکارهای به‌روزرسانی پویا است و در محیط‌های بزرگ ممکن است زمان اجرا چالش‌برانگیز شود. با این حال، با افزودن مکانیزم‌های بروزرسانی جمعیت و تخصیص پویا، الگوریتم پتانسیل استفاده در محیط‌های دنیای واقعی را دارد.



الف. درصد وظایف انجام‌شده در مهلت مقرر با افزایش تعداد گره (Nodes = ۲۰۰۰)



ب. درصد وظایف انجام‌شده در مهلت مقرر با افزایش تعداد وظایف (۲۵ VMs =)

شکل ۱۰: درصد وظایف انجام‌شده در مهلت مقرر

جدول ۵. نتایج شبیه‌سازی اضافی برای ۱۰ تکرار، تعداد وظایف متغیر و ۲۵ ماشین مجازی

		PSO	PSO-SA	IJFA	IJFSA	IJFSA-ODA
sum processing time(ms)	میانگین	۹۴۲۷/۶۳	۹۴۲۶/۲۰	۹۳۰۹/۵۵	۹۴۵۱/۲۵	۹۳۵۷/۷۷
	انحراف معیار	۳۲/۸۸	۳۹/۲۹	۴۳/۶۱	۲۱/۱۸	۲۱/۲۵
	فاصله اطمینان [حد پایین]	۹۳۸۷/۴۱	۹۳۴۸/۸۱	۹۲۲۳/۴۷	۹۴۰۹/۳۲	۹۳۱۵/۴۴
	فاصله اطمینان [حد بالا]	۹۵۳۶/۵۴	۹۵۰۴/۴۴	۹۳۹۵/۶۳	۹۴۹۳/۱۷	۹۳۹۸/۷۷
TotalExecution Time(s)	میانگین	۳/۴۶E+۱۳	۲/۳۷E+۱۳	۴/۳۵E+۱۵	۲/۸۷E+۱۳	۳/۸۵E+۱۳
	انحراف معیار	۴/۵۳E+۱۱	۲/۶۵E+۱۲	۱/۰۲E+۱۴	۱/۰۷E+۱۲	۲/۳۰E+۱۱
	فاصله اطمینان [حد پایین]	۱/۸۵E+۱۳	۴/۴۲E+۱۱	۴/۱۵E+۱۵	۲/۱۲E+۱۳	۲/۸۳E+۱۳
	فاصله اطمینان [حد بالا]	۲/۱۱E+۱۳	۵/۴۶E+۱۱	۴/۵۵E+۱۵	۲/۵۴E+۱۳	۲/۹۲E+۱۳
درصد وظایف انجام‌شده در مهلت مقرر	میانگین	۷۶/۴۵	۶۷/۹۳	۷۴	۷۶	۷۶/۲۸
	انحراف معیار	۱/۱۴	۰/۶۹	۰/۴۶	۰/۵۰	۰/۶۳
	فاصله اطمینان [حد پایین]	۶۴/۶۹	۶۶/۹۲	۷۳/۵۰	۷۴/۹۲	۷۴/۸۱
	فاصله اطمینان [حد بالا]	۶۷/۰۹	۶۹/۶۳	۷۵/۳۰	۷۶/۸۸	۷۷/۲۷
TotalEnergy Consumption (j)	میانگین	۴/۵۳E+۱۳	۱/۳۷E+۱۳	۴/۳۵E+۱۵	۲/۸۷E+۱۳	۳/۳۷E+۱۳
	انحراف معیار	۶/۶۳E+۱۱	۳/۰۶E+۱۱	۱/۰۲E+۱۴	۷/۱۴E+۱۱	۲/۳E+۱۱
	فاصله اطمینان [حد پایین]	۲/۲۵E+۱۳	۲/۳۱E+۱۳	۴/۱۵E+۱۵	۱/۹۳E+۱۳	۲/۸۳E+۱۳
	فاصله اطمینان [حد بالا]	۲/۵۱E+۱۳	۲/۴۳E+۱۳	۴/۵۵E+۱۵	۲/۲۱E+۱۳	۲/۹۲E+۱۳
Loadbalancing(ratio)	میانگین	۷۲/۴۲	۸۳/۷۴	۸۳	۸۷	۹۰/۶۶
	انحراف معیار	۱/۱۲	۰/۳۰	۱/۰۲	۰/۵۱	۰/۷۱
	فاصله اطمینان [حد پایین]	۷۸/۳۷	۸۳/۹۷	۸۱	۸۶	۸۹/۳۴
	فاصله اطمینان [حد بالا]	۸۲/۶۶	۸۵/۱۴	۸۵	۸۸	۹۲/۱۴

مراجع

[1] Boubaker, N. E. H., Zarour, K., Guermouche, N., & Benmerzoug, D. (2025). A comprehensive survey on resource management for IoT applications in edge–fog–cloud environments. IEEE Access.

[2] Sohrabi, S., Sakhaei-nia, M., Nassiri, M., & Mohammadi, R. (2025). IJFSA-OPG: A hybrid energy efficient load-balancing mechanism for fog-based IoT healthcare. Manuscript submitted for publication, Cloud Computing Journal.

[3] Ghorbian, Mohsen, Mostafa Ghobaei-Arani, and Leila Esmaeili. "A survey on the scheduling mechanisms in serverless computing: a taxonomy, challenges, and trends." Cluster Computing (2024): 1-40.

[4] Mokni, I., & Yassa, S. (2024). A multi-objective approach for optimizing IoT applications offloading in fog–cloud environments with NSGA-II. The Journal of Supercomputing, 1-39.

[5] Aghazadeh, Rafat, Ali Shahidinejad, and Mostafa Ghobaei-Arani. "Proactive content caching in edge computing environment: A review." Software: Practice and Experience 53.3 (2023): 811-855.

[6] Bezdan, T.; Zivkovic, M.; Bacanin, N.; Strumberger, I.; Tuba, E.; Tuba, M. Multi-objective task scheduling in a cloud computing environment by hybridized bat algorithm. J. Intell. Fuzzy Syst. 2022, 42, 411–423

[7] Available online: https://eucloudedgeiot.eu/ (accessed on 14 February 2023).

[8] Ghorbian, Mohsen, Mostafa Ghobaei-Arani, and Rohollah Asadolahpour-Karimi. "Function Placement Approaches in Serverless Computing: A Survey." Journal of Systems Architecture (2024): 103291.

[9] Abohamama, A.S.; El-Ghamry, A.; Hamouda, E. Real-time task scheduling algorithm for IoT-based applications in the cloud–fog environment. J. Netw. Syst. Manag. 2022, 30, 1–35.

[10] Zhang, Y., et al. (2022). "Task Scheduling in Cloud Computing Environment Using Advanced Phasmatodea Population Evolution Algorithm." *MDPI Electronics*, 11(9), 1451. DOI: 10.3390/electronics11091451MDPI

[11] Kumar, R., et al. (2022). "An Efficient Task Scheduling in a Cloud Computing Environment Using Hybrid Genetic Algorithm - Particle Swarm Optimization (GA-PSO) Algorithm." *World Scientific Journal*, 28(3), 1850024. DOI: 10.1142/S0219622018500244ResearchGate+1

[12] Chen, W., et al. (2021). "Blockchain-Enhanced Fair Task Scheduling for Cloud-Fog-Edge Systems." *IEEE Transactions on Industrial Informatics*, 17(4), 2567-2576. DOI: 10.1109/TII.2020.3023184buyya.com

[13] Baniata, H., Anaqreh, A., Kertesz, A.: PF-BTS: a privacy-aware fog-enhanced blockchain-assisted task scheduling. Inf. Process. Manage. (2021). https:// doi. org/ 10. 1016/j. ipm. 2020. 102393.

[14] Mirmohseni, S.M.; Tang, C.; Javadpour, A. FPSO-GA: A Fuzzy Metaheuristic Load Balancing Algorithm to Reduce Energy Consumption in Cloud Networks. Wirel. Pers. Commun. 2022, 127, 2799–2821.

[15] Malik, M.; Suman. Lateral Wolf Based Particle Swarm Optimization (LW-PSO) for Load Balancing on Cloud Computing. Wirel. Pers. Commun. 2022, 1, 1–20.

[16] Baniata, H., Anaqreh, A., Kertesz, A.: PF-BTS: a privacy-aware fog-enhanced blockchain-assisted task scheduling. Inf. Process. Manage. (2021). https:// doi. org/ 10. 1016/j. ipm. 2020. 102393.

[17] Beraldi R, Canali C, Lancellotti R, Mattia G. Distributed load balancing for heterogeneous fog computing infrastructures in smart cities. Pervasive Mob Comput 2020;67. https://doi.org/10.1016/j.pmcj.2020.101221.

[18] Sumathi M, Vijayaraj N, Raja SP, Rajkamal M. HHO-ACO hybridized load balancing technique in cloud computing. Int J Inf Technol 2023;15(3):1357–65.

[19] Gabhane JP, Pathak S, Thakare NM. A novel hybrid multi-resource load balancing approach using ant colony optimization with Tabu search for cloud computing. Innov Syst Softw Eng 2023;19(1):81–90. https://doi.org/10.1007/s11334-022-00508- 9.

[20] Salimi, R., Azizi, S., & Dogani, J. (2025). A hybrid priority-aware genetic algorithm and opposition-based learning for scheduling IoT tasks in green fog computing. *Computer Networks*, 267, Article 111349.

[21] Alsadie, D., & Alsulami, M. (2025). Modified grey wolf optimization for energy-efficient Internet of Things task scheduling in fog computing. *Scientific Reports*, 15, Article 14730.

[22] Rateb, R., Hadi, A. A., Tamanampudi, V. M., Abualigah, L., Ezugwu, A. E., et al. (2025). An optimal workflow scheduling in IoT–fog–cloud systems for minimizing time and energy. *Scientific Reports*, 15, Article 3607.

[23] Nazeri, M., Soltanaghaei, M., & Khorsand, R. (2024). A predictive energy-aware scheduling framework for scientific workflows in fog computing using MAPE-K and ANFIS. *Expert Systems with Applications*, 247, Article 123192.

[24] S. Sohrabi, M. Sakhaei-nia, M. Nassiri, and R. Mohammadi, "Fog-based architecture and efficient task offloading methodology in IoT-based applications for smart irrigation system," Computing, vol. 107, no. 3, p. 76, 2025.

[25] N. Jangu and Z. Raza, "Improved Jellyfish Algorithm-based multi-aspect task scheduling model for IoT tasks over fog integrated cloud environment," Journal of Cloud Computing, vol. 11, no. 1, p. 98, 2022.

[26] Khaledian, Navid, et al. "An energy-efficient and deadline-aware workflow scheduling algorithm in the fog and cloud environment." Computing 106.1 (2024): 109-137.

[27] Liu, W., Li, C., Zheng, A., Zheng, Z., Zhang, Z., & Xiao, Y. (2023). Fog computing resource-scheduling strategy in IoT based on artificial bee colony algorithm. Electronics, 12(7), 1511.

پاورقی‌ها:

[1] Fog Computing
[2] Fog Computing
[3] Quality of Service
[4] Percentage of Deadlines Satisfied Tasks
[5] Differential Evolution
[6] Artificial Bee Colony
[7] Improved Jellyfish Algorithm using Simulated Annealing
[8] Ant Colony Optimization
[9] Genetic Algorithm
[10] Particle Swarm Optimization
[11] Fuzzy Particle Swarm Optimization
[12] Blockchain
[13] Sequential Scheduling
[14] Adaptive Scheduling
[15] Harris Hawks Optimization
[16] Tabu Search
[17] Edge Layer
[18] Idle Power
[19] Jain's Fairness Index
[20] Simulated Annealing
[21] Friedman