

زمانبندی وظایف در سیستم‌های توزیع شده لایه مه و ابر محاسباتی با استفاده از الگوریتم بهینه‌سازی سوسک سرگین

محسن اقبالی^{1*}، رضا عزیزی²

¹ دانشجوی دکتری مهندسی کامپیوتر، گروه مهندسی کامپیوتر، واحد میبد، دانشگاه آزاد اسلامی، میبد، ایران،

m.eghbali@maybodiau.ac.ir

² استادیار گروه مهندسی کامپیوتر، واحد میبد، دانشگاه آزاد اسلامی، میبد، ایران، *azizi.reza@maybodiau.ac.ir*

چکیده

در چند سال گذشته اینترنت اشیا رشد قابل توجهی داشته‌است و تعداد زیادی شیء هوشمند به آن متصل شده‌است. رایانش ابری به عنوان یک سیستم پردازش داده‌ها در اینترنت اشیا است با این حال، سرورها در الگوی محاسبات ابری معمولاً در یک فاصله فیزیکی طولانی از دستگاه‌های اینترنت اشیا قرار دارند و تأخیر زیاد ناشی از فواصل طولانی نمی‌تواند به طور مؤثر برنامه‌های اینترنت اشیا بلادرنگ را برآورده کند. به دلیل این مسائل، محاسبات لبه و مه به عنوان فناوری محاسباتی محبوب در زمینه اینترنت اشیا ظاهر شده است. یکی از چالش‌های مهم اینترنت اشیا، مسئله زمان‌بندی وظایف در لایه مه و ابر است. در روش پیشنهادی برای تخصیص منابع آزاد از شبکه عصبی LSTM استفاده می‌شود و برای زمانبندی بهینه وظایف در لایه ابر و مه از الگوریتم بهینه‌سازی سوسک سرگین استفاده می‌شود. آزمایشات نشان می‌دهد که در مجموعه داده HPC2N دقت، حساسیت و صحت روش پیشنهادی برای پیش‌بینی وضعیت منابع به ترتیب برابر 94/72 درصد، 93/21 درصد و 91/64 درصد است. در مجموعه داده NASA دقت، حساسیت و صحت پیش‌بینی روش پیشنهادی در تخصیص منابع به ترتیب برابر 95/68 درصد، 94/61 درصد و 92/37 درصد است. روش پیشنهادی نسبت به روش‌های RNN، MLP، 1DCNN دقت بیشتری در تخصیص منابع برای زمان‌بندی دارد. شاخص Makespan روش پیشنهادی نسبت به روش‌های AO_AVOA، AVOA، PSO، FA و HHO مقدار کمتری و بهتری را در زمان‌بندی وظایف نشان می‌دهد.

کلیدواژه- اینترنت اشیا، الگوریتم بهینه‌سازی سوسک سرگین، زمانبندی وظایف، لایه ابر، لایه مه

1-مقدمه

با تکامل فناوری‌های دیجیتال، حجم عظیمی از داده‌ها از منابع متعدد تولید می‌شود. چنین داده‌هایی را می‌توان با استفاده از راه‌حل‌های رایانش ابری ذخیره و پردازش کرد. با این حال، رایانش ابری نمی‌تواند تحرک و امنیت اینترنت اشیا¹ را پشتیبانی کند [1]. اینترنت اشیا از گره‌های توزیع شده عظیمی تشکیل شده است که در آنها هر گره متشکل از حسگرهای مختلفی است. هر حسگر در داخل یک گره متوجه تفاوت‌هایی می‌شود که در ناحیه محصور در پهنای باند مجاز خود رخ داده است. متعاقباً پس از خوشه‌بندی و ارزیابی داده‌های جمع‌آوری شده، تصمیمات سنتی و اقدامات ترتیب داده شده اتخاذ می‌شود. اینترنت اشیا سیستمی است که در کاربردهای مختلف مانند تولید، مراقبت‌های بهداشتی، کشاورزی، مناطق شهری و غیره کشف شده است [2].

رایانش ابری² یک مرحله دسترسی به داده‌ها برای پشتیبان‌گیری و پردازش داده‌ها است زیرا دارای ذخیره‌سازی و پردازش بالایی است. داده‌های گسترده تولید شده توسط دستگاه‌های اینترنت اشیا برای پردازش (تخلیه کار) به محاسبات ابری ارائه می‌شود. با این حال، انتقال مستقیم داده‌ها به ابر ممکن است باعث هزینه‌های شبکه یا به عبارتی سربار پهنای باند شود. ابر معمولاً تأخیر پردازش داده‌های آپلود شده را به

¹Internet of Things (IoT)

²Cloud computing(CC)

دلیل زمان پاسخ طولانی به تاخیر می‌اندازد [3]. در نتیجه، یک جایگزین جدید، یعنی محاسبات مه³، برای رسیدگی به توانایی‌های دستگاه‌های اینترنت اشیا معرفی شد. بنابراین، محاسبات مه توسط سیسکو در سال 2012 برای حل این چالش‌ها ارائه شد. محاسبات مه، خدمات ابری را تا لبه شبکه گسترش می‌دهد. با انجام عملیات مربوطه با استفاده از منابع محلی که در نزدیکی دستگاه‌های لبه اینترنت اشیا وجود دارد، زمان و حجم انتقال داده را کاهش داد. بنابراین استفاده از منابع محلی باعث کاهش هزینه‌ها، کاهش تاخیر، افزایش سطح محرمانگی و امنیت و کاهش بار ترافیک شبکه می‌شود. وقتی منابع موثری در محاسبات مه وجود نداشته باشد، از منابع ابری با هزینه‌های بالاتر استفاده می‌شود. محاسبات مه و محاسبات لبه اغلب به جای یکدیگر در مطالعات برای اشاره به همان مفهوم توزیع منابع محاسباتی نزدیک‌تر به کاربران نهایی استفاده می‌شوند [4].

با اینترنت اشیا، اتصال به اینترنت فراتر از دستگاه‌های هوشمند سنتی مانند گوشی‌های هوشمند و تبلت‌ها به طیف متنوعی از دستگاه‌ها و موارد دیگر (حسگرها، ماشین‌ها، وسایل نقلیه و غیره) گسترش می‌یابد. اینترنت اشیا حجم عظیمی از داده‌ها را تولید می‌کند که نیاز به ذخیره، پردازش و تجزیه و تحلیل برای به دست آوردن اطلاعات ارزشمند به منظور برآوردن اهداف و نیازهای کاربر دارند. علاوه بر این، تعداد و مقیاس برنامه‌ها و خدمات نیز به سرعت در حال افزایش است بنابراین، نیازمند قابلیت پردازشی است تا آنجا که حتی قدرتمندترین دستگاه‌های هوشمند در حال حاضر نمی‌توانند آن را برآورده کنند [5]. محیط ابری که به عنوان یک مرکز منبع بزرگ شناخته می‌شود و امکان دسترسی همه‌جانبه به اشتراک‌گذاری و ارائه منابع به کاربران را به صورت انعطاف‌پذیر از طریق مکانیزم مجازی‌سازی فراهم می‌کند، می‌تواند یک پلتفرم بالقوه برای پشتیبانی از پیشرفت‌های اینترنت اشیا باشد. محدودیت‌های دستگاه‌های هوشمند موجود (عمر باتری، قدرت پردازش، ظرفیت ذخیره‌سازی و منابع شبکه) را می‌توان با انجام کارهای وقت‌گیر و پرمحصول به یک پلتفرم محاسباتی قدرتمند مانند محاسبات ابری به حداقل رساند و در همان حال وظایف ساده را برای دستگاه‌های هوشمند واگذار کرد [6]. مدل محاسباتی مه- ابر دارای مزایای متعددی از جمله کاهش تأخیر، کاهش ترافیک شبکه و افزایش کارایی انرژی است، با این حال، این مدل جدید با مجموعه‌ای از چالش‌ها نیز همراه است [7].

یکی از چالش‌های مطرح در اینترنت اشیا، تخصیص منابع [8]⁴ و زمان بندی وظایف [9]⁵ است. یک سیستم بسیار توزیع شده مانند محاسبات مه- ابر یک پلت فرم ایده آل برای استقرار برنامه‌های کاربردی است؛ آن دسته از برنامه‌های موازی که وظایف آنها مستقل از یکدیگر است. برنامه‌های کاربردی در بسیاری از سناریوها ظاهر می‌شوند، از جمله داده‌کاوی، جست‌وجوهای گسترده (مانند شکستن کلید)، محاسبات فراکتال، زیست‌شناسی محاسباتی، تصویربرداری کامپیوتری، رمزگذاری/رمزگشایی ویدیو و برنامه‌های کاربردی مختلف اینترنت اشیا [9]. چالش اصلی، زمان بندی وظایف در مجموعه گره‌های پردازشی از جمله گره‌های ابری (مانند سرورها یا ماشین‌های مجازی) و گره‌های مه است. هدف از زمان بندی وظایف در سیستم مه- ابر به نفع کاربران یا ارائه‌دهندگان خدمات است. از طرف کاربران، آنها نگران برخی از معیارهای ساخت، بودجه، مهلت، امنیت و هزینه هستند. از سوی دیگر، هدف ارائه‌دهندگان خدمات، متعادل کردن بار، استفاده از منابع و بهره‌وری انرژی است. برای تضمین QoS، زمان پاسخگویی نقش مهمی در زمانی که مستقیماً بر تجربه کاربر تأثیر می‌گذارد، ایفا می‌کند [10]. علاوه بر این، هزینه پیاده‌سازی نیز جنبه‌ای است که بسیار مورد علاقه و اقبال کاربران بوده است. یک برنامه زمان بندی کار که زمان تکمیل را به حداقل می‌رساند و در هزینه‌های پولی صرفه‌جویی می‌کند، توافق نامه سطح خدمات⁶ امضا شده با کاربران را برآورده می‌کند [11]. مدیریت وظایف⁷ در معماری‌های ابر- مه یک موضوع مهم است. استفاده بهینه از منابع ابر مه برای افزایش پارامترهای کیفی مختلف مانند زمان اجرای کار، هزینه عملیات و مصرف انرژی موضوع مهمی است. زمان بندی کار مناسب در محیط مه، هزینه‌ها و تاخیرهای پردازش/ارتباطات را کاهش می‌دهد. یکی از مشکلات محققان، انتخاب یک روش زمان بندی کارآمد است [12].

مسئله زمان بندی کار، دشواری چند جمله‌ای غیر قطعی⁸ و بهینه‌سازی چالش برانگیز است. امروز، الگوریتم فراابتکاری کاربردی می‌تواند برای حل مسئله زمان بندی وظایف، مورد استفاده قرار گیرد. از جمله روش‌های فراابتکاری که برای زمان بندی وظایف مورد استفاده قرار گرفته است، می‌توان به الگوریتم ژنتیک⁹ [13]، الگوریتم بهینه‌سازی ذرات¹⁰ [14]، الگوریتم بهینه‌سازی وال [15]¹¹ و الگوریتم بهینه‌سازی شاهین¹² [16]

³ Fog Computing (FC)

⁴ Resource allocation

⁵ Tasks scheduling

⁶ Service Level Agreement (SLA)

⁷ Task management

⁸ NP-hard

⁹ Genetic algorithms(GA)

¹⁰ Particle swarm optimization (PSO) algorithm

¹¹ Whale optimization algorithm(WOA)

¹² Harris hawks optimization(HHO) algorithm

اشاره کرد. هدف از این مقاله، ارائه یک رویکرد جدید برای در نظر گرفتن منابع ابر و مه برای زمانبندی کارها و وظایف در اینترنت اشیا است. ارائه یک تابع هدف جدید در زمانبندی برای کاهش تاخیر اجرای وظایف از دیگر اهداف تحقیق به شمار می‌رود. در روش پیشنهادی برای زمانبندی وظایف از الگوریتم بهینه‌سازی سوسک سرگین استفاده می‌شود تا تاخیر اجرای وظایف کاهش داده شود. ارائه یک نسخه گسسته از الگوریتم سوسک سرگین¹³ [17] برای زمانبندی وظایف در لایه مه و ابر، نخستین نوآوری مقاله است. نوآوری دیگر مقاله ارائه یک تابع هدف کارآمد برای زمانبندی وظایف است. به کارگیری منابع در لایه مه و ابر برای زمانبندی بهینه، دیگر نوآوری این مقاله به شمار می‌رود. پیش‌بینی بار محاسباتی منابع با استفاده از شبکه عصبی LSTM برای اجرای وظایف روی منابع آزاد از نوآوری‌های دیگر مقاله است. سهم نویسندگان در ارائه یک الگوریتم زمانبندی کارآمد در اینترنت اشیا در موارد ذیل خلاصه می‌شود:

- پیش‌بینی منابع آزاد با شبکه عصبی LSTM و زمانبندی وظایف طبق منابع آزاد
- ارائه یک نسخه گسسته از الگوریتم بهینه‌سازی سوسک سرگین برای زمانبندی
- زمانبندی دو سطحی در لایه مه و ابر
- تلفیق هوش گروهمی و یادگیری عمیق در زمانبندی بهینه
- ارائه یک تابع هدف جدید برای زمانبندی وظایف در لایه مه و ابر

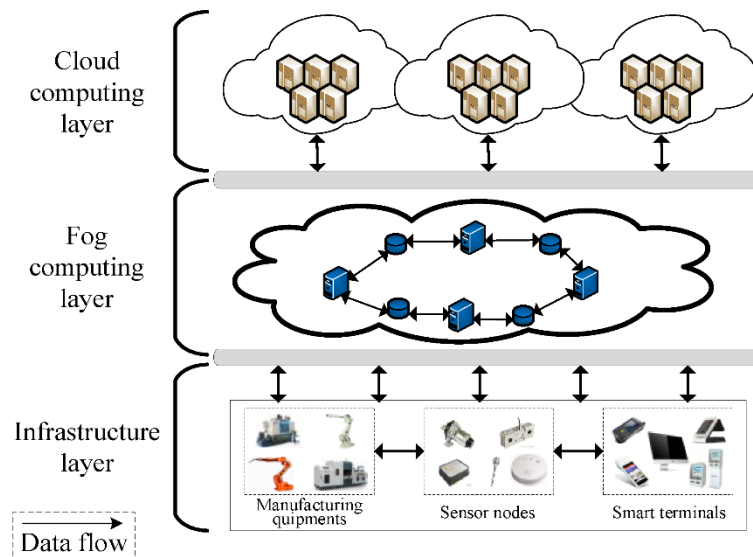
این مقاله یک روش زمانبندی دو سطحی در لایه مه و ابر و دارای 5 بخش است. در بخش II کارهای مرتبط در زمینه زمانبندی کارها ارائه می‌شود. در بخش III، سیستم زمانبندی دو سطحی بر اساس یادگیری عمیق و الگوریتم بهینه‌سازی سوسک سرگین توسعه داده شده است. در بخش IV، روش پیشنهادی پیاده‌سازی و با روش‌های مشابه مورد مقایسه قرار می‌گیرد. در بخش V نتایج تحقیق و یافته‌های تحقیق به همراه پیشنهادها آتی ارائه می‌شود.

2- کارهای مرتبط

رایانش ابری با استفاده از تکنیک‌های محاسبات موازی و توزیع شده، منابع رایانه مشترک را از طریق اینترنت در دسترس مشتریان قرار می‌دهد. مدل کسب‌وکار «پرداخت به حساب» رایانش ابری را تقریباً دموکراتیک کرده است. ارائه دهندگان ابر، ارائه دهندگان خدمات و کاربران نهایی در این مرحله از استقرار نرم افزار شرکت می‌کنند. ارائه دهندگان خدمات ابری قابلیت‌های محاسباتی را از طریق رایانه‌های مجازی به مشتریان خود ارائه می‌دهند. ارائه‌دهندگان خدمات از این ماشین‌های مجازی برای خدمات مشتری در سطح برنامه استفاده می‌کنند. ارائه‌دهندگان خدمات، الگوریتم‌های زمان‌بندی کار را برای پخش مشاغل مشتری در سراسر ماشین‌های مجازی، کاهش زمان پاسخگویی، اطمینان از کیفیت بالای خدمات و به حداکثر رساندن منابع پیاده‌سازی می‌کنند. به همین دلیل، الگوریتم زمان‌بندی کار، بخش مهمی از هر معماری ابری است. رایانش ابری نیاز به تنظیماتی برای چندین تکنیک زمانبندی مورد استفاده در محیط‌های مختلف رایانه دارد. ممکن است یک روش زمانبندی بهینه شده برای یک خوشه در فضای ابری ضعیف عمل کند. قبل از اینکه الگوریتم بتواند با ساختار محیط ابری مقابله کند، بخش‌های روش باید به فضای مشکل منتقل شوند. هرچه تنوع ماشین‌های مجازی و اندازه بارهای کاری که مدیریت می‌شوند بیشتر باشد، تعداد پیکربندی‌های کار موجود بیشتر است. یافتن کوتاه‌ترین مسیر در میان همه جایگشت‌های بالقوه یکی از چالش‌برانگیزترین مشکلات در علم کامپیوتر برای زمانبندی است. معماری اینترنت اشیا به طور معمول مانند شکل 1، از سه لایه شامل لایه زیرساخت، لایه محاسبات مه و لایه محاسبات ابری تشکیل شده است [18]:

- لایه زیرساخت: این لایه از دستگاه‌های پایانه با عملکردهای مختلف مانند سنسور (حسگر)های مختلف، دستگاه‌های پردازشی و پایانه‌های مختلف هوشمند تشکیل شده است. پایانه‌های هوشمند، وظایف ساده را به صورت محلی انجام می‌دهند اما قادر به انجام کارهای پیچیده در زمان واقعی نیستند.
- لایه محاسبات مه: این لایه عمدتاً از گره‌های مه تشکیل شده است. اینها سرورهایی با قابلیت‌های محاسباتی، ارتباطی و ذخیره‌سازی معین در خطوط تولید هوشمند مانند حسگرهای هوشمند، دستگاه‌های پردازش هوشمند و دستگاه‌های چندرسانه‌ای هوشمند

- هستند. این لایه می‌تواند درخواست‌های پایانه‌های خط تولید هوشمند را حس کرده و خدمات مختلفی را بلادرنگ ارائه دهد که می‌تواند تاخیر پردازش وظایف را تا حد زیادی کاهش دهد و کیفیت سرویس اپلیکیشن‌های بلادرنگ را تضمین کند.
- لایه محاسبات ابری: لایه رایانش ابری شامل خوشه‌هایی با ظرفیت محاسبات و ذخیره سازی عظیم است که خدمات از راه دور را برای خطوط تولید هوشمند به‌منظور انجام وظایف محاسباتی پیچیده ارائه می‌دهد.

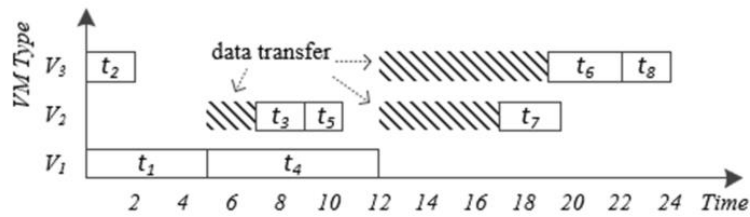


شکل 1: معماری سه لایه اینترنت اشیا [18]

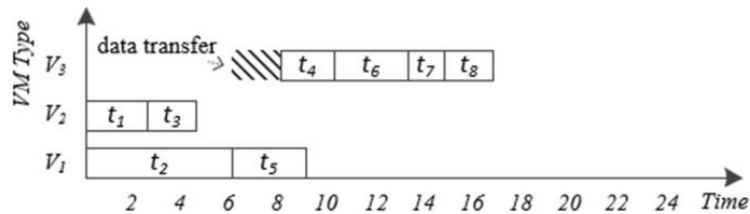
Figure 1. The three-layer architecture of the Internet of Things [18]

تعادل بار یکی از مهم‌ترین مشکلات برای اجرای وظایف زمان‌بندی چند منبعی در یک محیط محاسباتی ناهمگن است. اگر رویکرد متعادل کننده بار در برنامه کمک شود، می‌تواند بر مصرف انرژی تأثیر بگذارد. استفاده از CPU نیز برای هر دو قرارداد سطح سرویس و مصرف انرژی اجباری است که مستقیماً بر زمان‌بندی وظایف متعادل‌سازی بار تأثیر می‌گذارد. استفاده از CPU بر پردازش کار تأثیر می‌گذارد و هر گره CPU خود را برای پردازش وظایف دارد. بنابراین، روش پیشنهادی با هدف افزایش زمان‌بندی کار برای بهبود پردازش کار انجام می‌شود. ایجاد تعادل نگرانی بین منابع و وظایف، پاسخ پردازش وظایف و مدیریت منابع را به طور قابل توجهی بهبود می‌بخشد. متعادل‌سازی بار یک تکنیک ضروری در مدیریت منابع است که با تعیین عوامل مدیریت وظیفه می‌تواند به یک سیستم قوی تبدیل شود. زمان‌بندی شامل یافتن یک نقشه‌برداری بهینه برای تخصیص n وظیفه به پردازنده‌های متعدد است و علاوه بر این، چندین متغیر حیاتی برای افزایش زمان‌بندی در رایانش ابری وجود دارد [19]. زمان‌بندی را می‌توان یک نگاهت از اجرای وظایف روی منابع آزاد در لایه مه یا ابر در نظر گرفت. در شکل 2، یک فرآیند زمان‌بندی در لایه ابر و روی ماشین‌های مجازی نمایش داده شده است. با توجه به شکل 2، وظایف t_1 تا t_8 را می‌توان روی منابع مختلف V_1 ، V_2 و V_3 اجرا کرد. در حالت اول وظیفه اول و دوم در ماشین مجازی اول و سوم اجرا شده اما در سناریوی دوم وظیفه یک و دوم روی ماشین مجازی اول و دوم اجرا شده است. اجرای ترتیبی کارها و وظایف روی ماشین‌های مجازی باعث می‌شود که زمان‌بندی‌های مختلفی برای اجرای وظایف در لایه مه و ابر در نظر گرفته شود و از این رو یافتن اجرای وظایف بهینه از اهمیت بالایی برخوردار است [20]. برای انجام زمان‌بندی، تاکنون روش‌های مختلفی ارائه شده که یک دسته‌بندی از آن در شکل 3، قابل مشاهده است. با توجه به شکل مورد نظر برای زمان‌بندی وظایف روش‌های مبتنی بر لیست، اکتشافی، فراابتکاری، یادگیری ماشین و ترکیبی ارائه شده است [21].

Schedule Plan 1

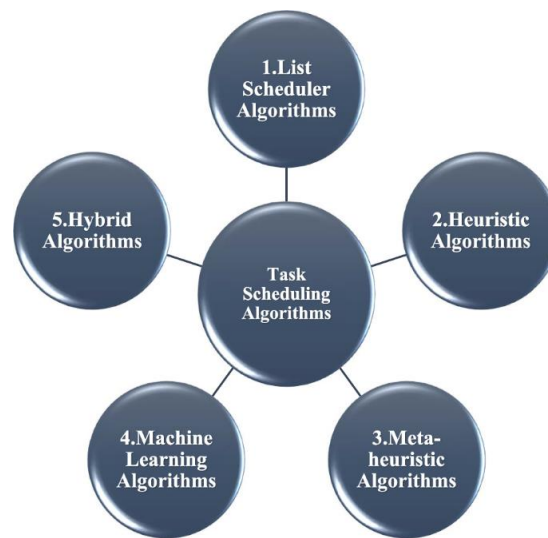


Schedule Plan 2



شکل 2: اجرای چند وظیفه در منابع اینترنت اشیا [20]

Figure 2. Multitasking in Internet of Things resources [20]

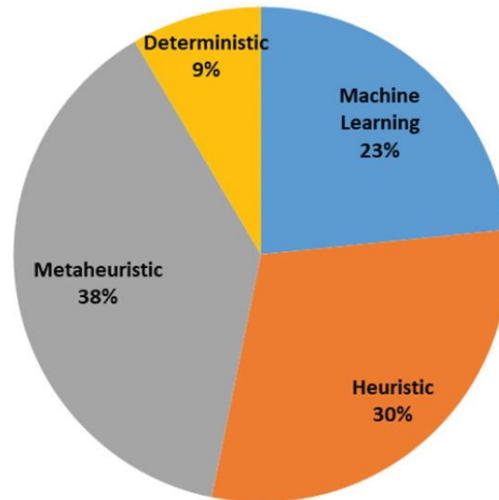


شکل 3: روشهای مختلف زمانبندی در اینترنت اشیا [21]

Figure 3. Different scheduling methods in the Internet of Things [21]

مکانیسم‌های یادگیری ماشین را می‌توان در زمان‌بندی وظایف مه برای بهبود کارایی و دقت فرآیند زمان‌بندی استفاده کرد. با تجزیه و تحلیل داده‌های تاریخی و شناسایی الگوها، الگوریتم‌های یادگیری ماشین می‌توانند بارهای ترافیکی آینده را پیش‌بینی و بر این اساس تخصیص وظایف را بهینه کنند. این می‌تواند منجر به کاهش تاخیر و بهبود عملکرد سیستم و همچنین استفاده بهتر از منابع شود. علاوه بر این، یادگیری ماشین می‌تواند برای شناسایی ناهنجاری‌ها یا خرابی‌های احتمالی در سیستم و اتخاذ اقدامات پیشگیرانه برای جلوگیری از آنها استفاده شود. به طور کلی، ترکیب مکانیسم‌های یادگیری ماشین در زمان‌بندی وظایف مه می‌تواند اثربخشی فرآیند را افزایش داده و عملکرد کلی سیستم‌های محاسباتی مه را بهبود بخشد [22]. مکانیسم‌های مبتنی بر اکتشاف برای زمان‌بندی وظایف مه روش‌هایی هستند که از قوانین سرانگشتی یا بهترین شیوه‌ها برای تخصیص وظایف به دستگاه‌ها در یک محیط محاسباتی مه استفاده می‌کنند. این مکانیسم‌ها بر تکنیک‌های یادگیری ماشین تکیه نمی‌کنند، بلکه بر قوانین تصمیم‌گیری ساده مبتنی بر تجربه یا شهود هستند. مکانیسم‌های مبتنی بر اکتشاف می‌توانند ساده و آسان برای پیاده‌سازی باشند، اما ممکن است همیشه به تخصیص بهینه کار منجر نشوند. آنها همچنین مانند رویکردهای مبتنی بر یادگیری ماشینی، با شرایط در حال تغییر سازگار نیستند یا از تجربیات گذشته درس نمی‌آموزند. بخش زیر چندین

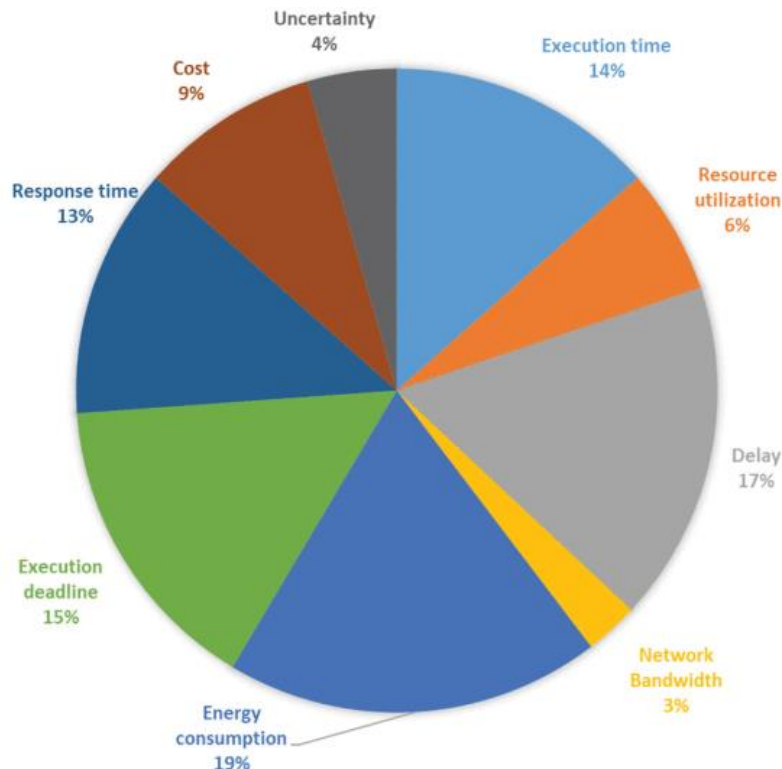
روش زمان‌بندی مه مبتنی بر اکتشافی را بررسی می‌کند که از عوامل مختلفی برای دستیابی به نتایج زمان‌بندی بهبودیافته استفاده می‌کند [22]. در الگوریتم‌های فراابتکاری، یک فضای حل تصادفی برای زمان‌بندی کار استفاده می‌شود. با تغییرات اندکی در الگوریتم‌های فراابتکاری، می‌توان از آنها برای حل مسائل مختلف بهینه‌سازی استفاده کرد. الگوریتم‌های فراابتکاری، مستقل از مسئله هستند. به‌کارگیری محاسبات مه با استفاده از الگوریتم‌های فراابتکاری در بهبود استفاده از منابع و کاهش تأخیر در محیط‌های محاسباتی مه توزیع شده امیدوارکننده است [22]. مکانیسم‌های قطعی همیشه خروجی یکسانی را برای یک ورودی معین تولید می‌کنند. مکانیسم‌های قطعی برای زمان‌بندی وظایف مه شامل استفاده از مجموعه‌ای از قوانین و الگوریتم‌های از پیش تعریف شده برای تخصیص وظایف به گره‌های مه است. جست‌وجوی جامع نمونه‌ای از رویکرد قطعی به مسئله زمان‌بندی کار است. الگوریتم‌های جامع کل فضای جست‌وجو را برای یافتن طرح بهینه بر اساس مدل هزینه داده شده برمی‌شمارند. این مکانیسم‌ها شامل هیچ روش تصادفی یا احتمالی نیستند. مکانیسم‌های قطعی برای زمان‌بندی وظایف مه ساده و آسان برای پیاده‌سازی هستند. با این حال، ممکن است همیشه منجر به استفاده بهینه از منابع یا کاهش تأخیر نشوند [22]. شکل 4، سهم انواع روش‌های زمان‌بندی و بخصوص الگوریتم‌های فراابتکاری و ابتکاری را نشان می‌دهد [22].



شکل 4: سهم روش‌های مختلف زمان‌بندی در اینترنت اشیا [22]
Figure 4. Contribution of different scheduling methods in Internet of Things [22]

با توجه به نمودار فوق، الگوریتم‌های زمان‌بندی فراابتکاری و ابتکاری در زمان‌بندی کارها و وظایف دارای سهمی به ترتیب و در حدود 38 درصد و 30 درصد است.

در زمان‌بندی وظایف شاخص‌های مختلفی به کار گرفته می‌شود که تعدادی از آنها در شکل 5، نمایش داده شده است. با توجه به شکل مورد نظر می‌توان گفت که مصرف انرژی، مهلت زمانی در اجرای وظایف، تأخیر اجرا و زمان اجرا از مهمترین شاخص‌های ارزیابی در زمان‌بندی است [22].



شکل 5: شاخص های اصلی در زمانبندی [22]
Figure 5. Main indicators in timing [22]

برای زمانبندی وظایف در لایه مه یا ابر تاکنون چندین مطالعه انجام شده است که بیشتر آنها بر محور روش های بهینه سازی و فراابتکاری است که در ادامه تعدادی از کارهای مرتبط در این زمینه مرور می شود.

در [23]، سال 2023، یک روش زمان بندی وظایف و متعادل سازی منابع محاسبات مه در کارخانه هوشمند ارائه شده است. در این پژوهش یک الگوریتم زمان بندی ژنتیکی تقسیم بازه های بهبود یافته یا الگوریتم زمان بندی تقسیم فاصله زمانی بر اساس الگوریتم ژنتیک برای برنامه ریزی و تخصیص وظایف در کارخانه هوشمند پیشنهاد شده است. نتایج شبیه سازی نشان می دهد که در مقایسه با الگوریتم زمان بندی پیش فرض IDGSA، Kubernetes می تواند زمان پردازش داده ها را تا 50 درصد کاهش دهد و استفاده از منابع محاسباتی مه را تا 60 درصد بهبود بخشد. در مقایسه با الگوریتم ژنتیک سنتی، با تکرارهای کمتر، روش آنها می تواند زمان پردازش داده ها را تا 7 درصد کاهش دهد و استفاده از منابع محاسباتی مه را تا 9 درصد بهبود بخشد و در مقایسه با روش معمولی Joines&Houck، الگوریتم پیشنهادی می تواند بسیار سریع تر همگرا شود و نتایج بهینه سازی بهتری را ارائه کند. علاوه بر این، شبیه سازی نشان می دهد که روش آنها در محاسبات مشترک ابری و مه می تواند کل تاخیر کار را به ترتیب 18 و 7 درصد در مقایسه با محاسبات فقط ابری و فقط مه کاهش دهد.

در [24]، سال 2023، مروری بر تکنیک های زمان بندی وظایف در محاسبات ابری و مه انجام شده است. زمان بندی کارآمد بر روی ابر برای استفاده بهینه از منابع در مراکز داده حیاتی است. با ظهور برنامه های نسل پنجم و اینترنت اشیا که تعداد زیادی کار با الزامات تأخیر دقیق ایجاد می کنند، چالش برانگیزتر شده است. این امر محاسبات مه / لبه را به وجود می آورد که یک لایه مکمل برای ابر محاسباتی است. تأخیر وظایف در محاسبات مه را می توان کاهش داد زیرا پردازش در شبکه به دستگاه های نهایی و کاربران نزدیک تر انجام می شود، اما به دلیل در دسترس بودن منابع محدود، نمی توان هر کار را در مه برنامه ریزی کرد. الگوریتم های زمان بندی مرسوم اغلب در بهره برداری از منابع ناهمگن شکست می خورند. بنابراین، الگوریتم های زمان بندی به خوبی طراحی شده و به خوبی تنظیم شده برای دستیابی به کیفیت بهتر خدمات مورد نیاز هستند. در این مطالعه، الگوریتم های زمان بندی کار پیشرفته در محیط های ابر و مه در مجموعه ای از ابعاد مختلف مورد بررسی قرار می گیرند. مطالعات نشان داد 97 درصد از مطالعات بر اهداف چندگانه تمرکز می کنند و 68 درصد از تکنیک ها غیر قطعی هستند. علاوه بر این، در مجموع بیست هدف زمان بندی مختلف با زمان ساخت، استفاده از منابع، تاخیر، متعادل سازی بار و مصرف انرژی به عنوان مهم ترین

معیارها شناسایی شده‌اند. روش‌های ارزیابی شامل شبیه‌سازی (51 درصد)، آزمایش‌های واقعی (4 درصد)، معادلات تحلیلی (2 درصد) و مجموعه داده‌ها (43 درصد) و غیره بررسی می‌شوند. در پایان، مسائل باز، چالش‌ها و جهت‌گیری‌های آتی مورد بحث قرار می‌گیرد. در [25]، سال 2023، یک روش زمان‌بندی کار بهینه در شبکه مه و ابر با استفاده از الگوریتم پروانه-شعله چندهدفه ارائه دادند. آزمایشات آنها نشان داد روش پیشنهادی از الگوریتم بهینه‌سازی ازدحام ذرات، الگوریتم کرم شب تاب، الگوریتم‌های ازدحام سالپ، بهینه ساز شاهین هریس و کلونی زنبورهای مصنوعی در زمان‌بندی تأخیر کمتری ارائه می‌کند. بر اساس آزمایش‌ها، راه‌حل پیشنهادی زمان تکمیل وظایف اینترنت اشیا و زمان خروجی را کاهش داده‌است، بنابراین تأخیر ناشی از پردازش وظایف، مصرف انرژی و انتشار CO2 و افزایش نرخ عملکرد سیستم را کاهش داده است.

در [26]، سال 2023، برای زمان‌بندی کار در معماری لبه-مه-ابر یک رویکرد متعادل کننده بار چند هدفه با استفاده از الگوریتم یادگیری تقویتی ارائه دادند. در این پژوهش، یک الگوریتم زمان‌بندی مه یادگیری تقویتی برای رسیدگی زمان‌بندی وظایف پیشنهاد شده‌است. نتایج تجربی نشان می‌دهد که الگوریتم پیشنهادی تعادل بار را افزایش داده و زمان پاسخ را در مقایسه با الگوریتم‌های زمان‌بندی موجود کاهش می‌دهد. علاوه بر این، الگوریتم پیشنهادی از نظر تعداد دستگاه‌های مورد استفاده از سایر رویکردها بهتر عمل می‌کند.

در [27]، سال 2023، یک الگوریتم جدید الهام گرفته از طبیعت برای زمان‌بندی وظایف بهینه در سیستم مه-ابر ارائه دادند. در سال‌های اخیر، به دلیل رشد تصاعدی داده‌های تولید شده توسط دستگاه‌های هوشمند متصل به هم، استفاده از برنامه‌های کاربردی اینترنت اشیا مبتنی بر ابر مه به طور پیوسته در حال افزایش بوده‌است. با این حال، ارائه‌دهندگان ابری که مسئول این برنامه‌های کاربردی اینترنت اشیا هستند با دو مشکل اساسی نحوه محافظت از سیستم در برابر کاربران غیرقابل اعتماد و نحوه تخصیص واحدهای پردازش برای برآورده کردن خواسته‌ها با هزینه‌های قابل قبول روبرو هستند. در این پژوهش، یک نسخه بهبود یافته از الگوریتم بهینه‌سازی مبتنی بر انتخاب زندگی¹⁴ برای حل زمان‌بندی کار ارائه شده‌است. زمان‌بندی وظایف یکی از برجسته‌ترین مشکلات در تخصیص منابع است. الگوریتم پیشنهادی آنها نه تنها سرعت همگرایی را افزایش می‌دهد، بلکه تنوع را بهتر حفظ می‌کند و قدرت، تأخیر و هزینه را بهینه می‌کند. آزمایشات نشان می‌دهد الگوریتم بهینه‌سازی مبتنی بر انتخاب زندگی با دستیابی به نتایج بهتر برای تأخیر و مصرف انرژی از روش‌های پیشرفته مشابه بهتر عمل می‌کند.

در [28]، سال 2023، یک الگوریتم جست‌وجوی گرانشی چند هدفه برای زمان‌بندی وظایف در سیستم محاسباتی ابری مبتنی بر اینترنت اشیا ارائه شده است. در این پژوهش، یک تابع چند هدفه جدید برای به حداقل رساندن هزینه ساخت، انرژی و هزینه پولی برای زمان‌بندی کار در سیستم مه-ابر پیشنهاد شده است. در این پژوهش، عملکرد الگوریتم پیشنهادی در دو سناریو مورد ارزیابی قرار گرفته‌است. ابتدا، عملکرد روش پیشنهادی با برخی از روش‌های بهینه‌سازی چندهدفه محبوب در برخی از توابع تست استاندارد مقایسه می‌شود. نتایج نشان می‌دهد که الگوریتم پیشنهادی نسبت به سایر روش‌ها عملکرد بهتری دارد. سپس از این الگوریتم برای یافتن راه حل مناسب برای مسئله زمان‌بندی کار استفاده می‌شود. رویکرد پیشنهادی مصرف انرژی را تا 22 درصد و هزینه پردازش را تا 40 درصد بهبود بخشیده است. تجزیه و تحلیل آماری نشان می‌دهد که در [29]، سال 2023، یک استراتژی زمان‌بندی کار مبتنی بر الگوریتم بهینه‌سازی کرکس‌های آفریقای برای محاسبات مه-ابر ارائه دادند. این پژوهش یک استراتژی برنامه‌ریزی وظایف مبتنی بر الگوریتم بهینه‌سازی کرکس‌های آفریقای پیشرفته برای محاسبات مه-ابر پیشنهاد می‌کند. کمیته‌سازی زمان ساخت، هزینه و مصرف انرژی در الگوریتم پیشنهادی به عنوان تابع هدف در نظر گرفته می‌شود. برای اولویت بندی وظایف، بهترین و بدترین روش به‌منظور رسیدگی به حساسیت تأخیرهای کار استفاده می‌شود. وظایف حساس به تأخیر، به محیط مه فرستاده می‌شوند، در حالی که وظایف تحمل تأخیر به ابر ارسال می‌شوند. نتایج شبیه‌سازی نشان می‌دهد روش آنها شاخص makepan را تا 53 درصد و مصرف انرژی را تا 44 درصد بهبود می‌دهد.

در [30]، سال 2023، یک روش زمان‌بندی بهینه در شبکه اینترنت اشیا-مه-ابر با استفاده از ترکیب الگوریتم بهینه‌ساز عقاب طلایی و الگوریتم بهینه‌سازی کرکس‌های آفریقای ارائه دادند. در این پژوهش مشکل زمان‌بندی وظایف در محیط مه-ابر برای اجرای درخواست‌های وظایف اینترنت اشیا به عنوان یک مشکل بهینه‌سازی با توجه به نیازهای بیشتر QoS در نظر گرفته شده است. ترکیبی از الگوریتم بهینه‌ساز عقاب طلایی و الگوریتم بهینه‌سازی کرکس‌های آفریقای برای حل مشکل زمان بندی کار در محیط مه-ابر استفاده شده است. روش پیشنهادی آنها در زمان‌بندی برای بهبود فرآیند اکتشاف الگوریتم کرکس استفاده شد و مرحله اکتشاف الگوریتم عقاب طلایی با فاز بهره برداری کرکس ترکیب شد. کمیته‌سازی تابع زمان انجام برای ارزیابی الگوریتم بهینه‌ساز برای بهینه‌سازی مسئله زمان‌بندی و یافتن بهترین ماشین‌های مجازی در محیط ابر-مه به کار گرفته شده است. روش پیشنهادی آنها در زمان‌بندی برای دو مجموعه داده با استفاده از معیارهای زمان ساخت، مقدار تابع

¹⁴ Life-choice-based optimization algorithm (ILCO)

تناسب، PIR و زمان توان اعمال شد. آزمایشات نشان داد روش پیشنهادی آنها در زمانبندی در مقایسه با الگوریتم‌های PSO، FA، HHO، AVOA و AO از نظر معیارهای ارزیابی مانند تأخیر بهتر عمل می‌کند.

در [31]، سال 2024، یک روش زمانبندی مبتنی بر یادگیری تقویتی عمیق برای بهینه‌سازی بار سیستم و زمان پاسخگویی در محیط‌های محاسباتی لبه و مه ارائه شده‌است. از آنجایی که تعداد زیادی از برنامه‌های اینترنت اشیاء نیاز به اجرا در منابع لبه/مه دارند، ممکن است سرورها بیش از حد بارگذاری شوند. از این رو، ممکن است سرورهای لبه/مه را مختل کند و همچنین بر زمان پاسخگویی برنامه‌های اینترنت اشیاء تأثیر منفی بگذارد. علاوه بر این، بسیاری از برنامه‌های کاربردی اینترنت اشیاء از اجزای وابسته تشکیل شده‌اند که محدودیت‌های اضافی را برای اجرای خود متحمل می‌شوند. علاوه بر این، محیط‌های محاسباتی لبه/مه و برنامه‌های اینترنت اشیاء ذاتاً پویا و تصادفی هستند. بنابراین، زمان‌بندی کارآمد و سازگار برنامه‌های اینترنت اشیاء در محیط‌های محاسباتی لبه/مه ناهمگن، از اهمیت بالایی برخوردار است. با این حال، منابع محاسباتی محدود در سرورهای لبه/مه بار اضافی را برای اعمال تکنیک‌های بهینه اما محاسباتی نیازمند تحمیل می‌کند. برای غلبه بر این چالش‌ها، آنها یک الگوریتم برنامه‌ریزی و زمانبندی مبتنی بر یادگیری تقویتی را پیشنهاد دادند تا زمان پاسخ‌دهی برنامه‌های ناهمگن اینترنت اشیاء را به طور سازگار و کارآمد بهینه کند و بار سرورهای لبه/مه را متعادل کند. نتایج به‌دست‌آمده نشان می‌دهد که روش آنها در مقایسه با الگوریتم‌های فراابتکاری و سایر تقویت‌کننده‌ها، به ترتیب تا 55، 37 درصد و 50 درصد هزینه اجرای برنامه‌های اینترنت اشیاء را از نظر تعادل بار، زمان پاسخ و هزینه وزنی کاهش می‌دهد.

در [32]، سال 2024، رویکرد مبتنی بر الگوریتم بهینه‌سازی هریس هاکس و سیستم فازی برای بهبود زمان‌بندی وظایف مبتنی بر ابر را ارائه دادند. زمان‌بندی وظایف مستلزم تخصیص وظایف مختلف به ماشین‌های مجازی است. در نتیجه، الگوریتم‌های زمان‌بندی برای دستیابی به مجموعه‌ای از اهداف، از جمله کاهش زمان ساخت، به حداقل رساندن مصرف انرژی، افزایش بهره‌وری منابع، دستیابی به متعادل‌سازی بار و بهینه‌سازی هزینه‌ها، با دقت ساخته می‌شوند. با توجه به اهمیت عمیق این اهداف، الگوریتم‌هایی که برای چنین سناریوهایی طراحی شده‌اند، همواره اهداف متعددی را در بر می‌گیرند. این پژوهش یک الگوریتم زمان‌بندی کار چندهدفه ابتکاری را برای محاسبات ابری معرفی می‌کند که به طور یکپارچه الگوریتم بهینه‌سازی هریس هاکس را ادغام می‌کند و قدرت منطق فازی را در خود جای می‌دهد. این روش، از الگوریتم شاهین برای کشف فضای راه حل گسترده استفاده می‌کند در حالی که راه حل‌های تولید شده را از طریق منطق فازی در معرض ارزیابی دقیق قرار می‌دهد. آزمایشات نشان داد روش آنها در حدود 73 درصد باعث کاهش در مصرف انرژی و 19 درصد کاهش چشمگیر هزینه می‌شود.

3- روش پیشنهادی

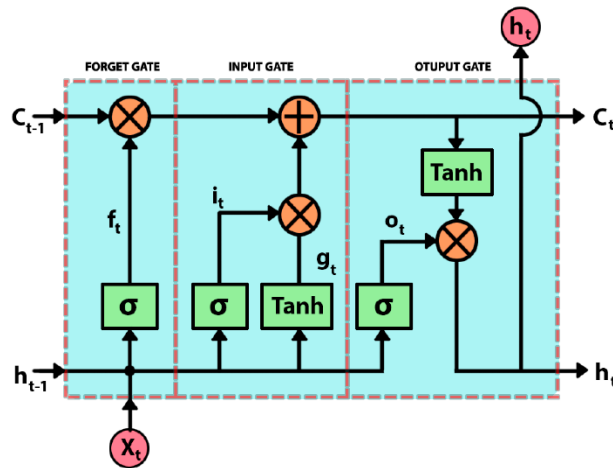
روش پیشنهادی در این بخش برای زمانبندی وظایف، ارائه و معرفی می‌شود. در روش پیشنهادی در فاز اول با استفاده از شبکه عصبی LSTM میزان بار هر منبع پیش‌بینی می‌شود و در ادامه هر منبع که دارای بار محاسباتی کمتری باشد برای به‌کارگیری زمانبندی تخصیص داده می‌شود. برای تخصیص وظایف به منابع که می‌تواند ماشین‌های فیزیکی و مجازی باشد از الگوریتم بهینه‌سازی سوسک سرگین استفاده می‌شود. روش پیشنهادی برای تخصیص منابع از الگوریتم LSTM و برای زمانبندی از الگوریتم سوسک سرگین استفاده می‌شود. مراحل روش پیشنهادی برای تخصیص منابع و زمانبندی در شکل 6، نمایش داده شده است.



شکل 6: چارچوب سیستم پیشنهادی برای تخصیص منابع و زمانبندی وظایف
Figure 6. A proposed system framework for resource allocation and task scheduling

3-1- پیش بینی منابع در دسترس با LSTM

یکی از چالش‌های مهم انواع روش‌های زمانبندی بی‌توجهی به تخصیص منابع است. در بیشتر مطالعات مرتبط با زمانبندی، فرض بر آن است که همه منابع در دسترس است اما در عمل، برخی از منابع که می‌توانند ماشین‌های فیزیکی یا مجازی باشند دارای بار محاسباتی زیادی می‌شوند. استفاده از منابعی که مرتبط اشغال می‌شوند باعث می‌شود تا وظایف به کندی روی آنها اجرا شود. یک رویکرد مناسب آن است که وظایف روی منابعی اجرا شود که بار محاسباتی آنها زیاد نشود. به عبارت بهتر یک مکانیزم پیش‌بینی برای اشغال یا آزاد بودن منابع می‌تواند به زمانبندی بهینه کمک نماید. برای پیش‌بینی وضعیت منابع در روش پیشنهادی از شبکه عصبی مصنوعی LSTM استفاده می‌شود زیرا این شبکه، توانایی بالایی برای پیش‌بینی رویدادها دارد. واحدهای LSTM توسعه شبکه‌های تکراری¹⁵ هستند که به آنها اجازه می‌دهد تا ورودی‌های خود را برای مدت طولانی‌تری مانند حافظه رایانه به خاطر بسپارند. می‌تواند اطلاعات را از سلول خود بخواند، بنویسد و حذف کند، اطلاعات را ذخیره کند و از ناپدید شدن آنها در طول زمان جلوگیری کند. سلول بر اساس درجه اهمیتی که به آن اختصاص داده شده است تصمیم می‌گیرد اطلاعاتی را که دریافت می‌کند ذخیره یا حذف کند. این اهمیت از طریق وزن اتصالات تعیین می‌شود، بنابراین LSTM در طول زمان یاد می‌گیرد که تشخیص دهد چه بخشی از اطلاعات مهم است و چه چیزی مهم نیست. اجرای یک حالت داخلی سلول که حافظه بلندمدت ایجاد می‌کند. همانطور که در شکل 7، نشان داده شده است، یک نورون LSTM از سه گیت: ورودی، فراموشی و خروجی تشکیل شده است که تعیین می‌کند آیا ورودی جدید مجاز است، کدام اطلاعات حذف می‌شود یا اینکه اجازه دارد در زمان فعلی بر خروجی تأثیر بگذارد [33].



شکل 7: ساختار یک نورون عصبی در شبکه LSTM
Figure 6. The structure of a neuron in a network

ساختار نورون LSTM (دروازه فراموشی، دروازه ورودی، دروازه خروجی) در شکل فوق نشان داده شده است. σ مخفف تابع سیگموئید، x_t داده ورودی و h_t خروجی شبکه جاری است. اولین دروازه‌ای که از چپ به راست در شکل فوق ظاهر می‌شود، دروازه فراموشی است که تصمیم می‌گیرد چه اطلاعاتی از حالت سلول (C_t) از طریق یک تابع سیگموئید (σ) کنار گذاشته شود و مطابق معادله 1، فرموله می‌شود:

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \quad (1)$$

که در آن (h_{t-1}) خروجی قبلی واحد، (x_t) ورودی، (W_f) ماتریس وزن دروازه فراموشی و (b_f) بایاس گیت است. خروجی بین مقداری از 0 تا 1 خواهد بود که 0 نشان دهنده حذف کل و 1 کل اطلاعات را حفظ می‌کند. پس از این مرحله، دروازه ورودی ادامه می‌یابد که نشان می‌دهد چه بخشی از اطلاعات جدید در حالت سلول ذخیره می‌شود. این دروازه از دو تابع یکی سیگموئیدی و دیگری مماس هذلولی تشکیل شده است که میزان به‌روز رسانی حالت را تعیین می‌کند که در معادلات 2 و 3 فرموله شده‌اند.

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \quad (2)$$

$$g_t = \tanh(W_g * [h_{t-1}, x_t] + b_g) \quad (3)$$

ماتریس وزن مطابق معادله 4، با حذف اطلاعات غیر ضروری و به‌روز رسانی آن با اطلاعات جدید به دست آمده، وضعیت جدید واحد به دست می‌آید.

$$C_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \quad (4)$$

در نهایت خروجی واحد را تعیین کنید (h_t)، که نسخه‌ای از حالت سلولی است که توسط ورودی فعلی و خروجی قبلی فیلتر شده است. جایی که W_o و b_o ماتریس وزن و بایاس برای دروازه خروجی هستند که در معادلات 5 و 6 فرموله شده است:

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

در روش پیشنهادی شبکه عصبی LSTM با اجزای قبلی از سیستم آموزش داده می‌شود. ورودی شبکه LSTM منابع و خروجی آن پیش‌بینی اشغال یا آزاد بودن منبع است و هدف از به‌کارگیری LSTM در روش پیشنهادی، اجرای وظایف در منابع آزاد است.

3-2- تابع هدف زمانبندی

هنگامی که درخواست‌های لایه حسگر به لایه مه در قالب وظایف ارسال می‌شوند، به وظایف کوچک و مستقل تجزیه می‌شوند تا در زیرساخت محاسباتی مه و ابر پردازش شوند. هر کار دارای ویژگی‌هایی نظیر تعداد دستورالعمل‌ها، حافظه مورد نیاز، اندازه فایل‌های ورودی و خروجی است [34]. با فرض اینکه T_k نشان دهنده k امین وظیفه باشد، در هر بار مجموعه‌ای از n وظیفه مستقل به صورت معادله 7، به سیستم ارسال می‌شود:

$$T = \{T_1, T_2, T_3, \dots, T_n\} \quad (7)$$

زیرساخت محاسباتی مه و ابر، از پردازنده‌ها، یعنی گره‌های ابری و گره‌های مه که ویژگی‌های یکسانی مانند نرخ CPU، هزینه استفاده از CPU، هزینه استفاده از حافظه و هزینه استفاده از پهنای باند را دارند، تشکیل شده است. با این حال، گره‌های ابری معمولاً قدرتمندتر از گره‌های مه هستند اما هزینه استفاده از آنها بیشتر است. در روش پیشنهادی فرض می‌شود که m پردازنده در گره‌های لایه مه و ابر قرار دارد و تعداد گره‌های لایه مه و ابر با f و c نمایش داده می‌شود ($N = N_{\text{cloud}} \cup N_{\text{Fog}}$) و به صورت معادله 8 بیان می‌شود:

$$N = \{N_1, N_2, N_3, \dots, N_m\} \quad (8)$$

که در آن N_i گره پردازش شماره i را نشان می‌دهد. هر کار T_k به پردازنده N_i اختصاص داده می‌شود که به صورت T_{ik} نمایش داده می‌شود. یک پردازنده را می‌توان برای پردازش مجموعه‌ای از یک یا چند کار اختصاص داد که در معادله 9، ارایه شده است:

$$N_i \text{Tasks} = \{T_x^i, T_y^i, \dots, T_z^i\} \quad (9)$$

مسئله زمانبندی کار در محیط محاسباتی مه و ابر را می‌توان به صورت جست‌وجوی مجموعه‌ای معادله 10، فرموله کرد:

$$\text{NodeTasks} = \{T_1^a, T_2^b, T_3^c, \dots, T_n^p\} \quad (10)$$

برای مجموعه‌ای از وظایف $N_i \text{Tasks}$ ، زمان اجرا (EXT) آن گره نیاز به تکمیل تمام وظایف محول شده به شرح رابطه 11 محاسبه می‌شود:

$$\text{EXT}(N_i) = \sum_{T_k^i \in N_i \text{Tasks}} \text{Exetime}(T_k^i) = \frac{\sum_{T_k^i \in N_i \text{Tasks}} \text{length}(T_k^i)}{\text{CPUrate}(N_i)} \quad (11)$$

که در آن $\text{ExeTime}(T_k^i)$ زمان اجرای T_k پردازش شده در گره N_i است که به صورت معادله 12، محاسبه می‌شود:

$$\text{ExeTime}(T_k^i) = \frac{\text{length}(T_k^i)}{\text{CPUrate}(N_i)} \quad (12)$$

با $\text{length}(T_k)$ طول زمانی اجرای وظیفه T_k و $\text{CPUrate}(N_i)$ نرخ پردازش در پردازنده گره i ام است. برای تکمیل تمام وظایف، از زمانی که درخواست دریافت می‌شود تا زمانی که آخرین کار تکمیل یا زمانی که آخرین ماشین تمام می‌شود، تعریف شده است که با Makespan نمایش داده شده و با فرمول معادله 13، تعیین می‌شود:

$$\text{Makespan} = \text{Max}_{1 \leq i \leq m} \{\text{EXT}(N_i)\} \quad (13)$$

فرض کنید MinMakespan حد پایینی Makespan باشد، یعنی سیستم به کوتاه‌ترین زمان نیاز دارد تا تمام آن کار را کامل کند. در حالت ایده‌آل، زمانی که همه گره‌ها همه وظایف محول شده را همزمان به پایان برسانند با MinMakespan نشان داده شده و با معادله 14، فرموله می‌شود و می‌توان آن را مانند معادله 15، فرموله کرد:

$$\text{MinMakespan} = \text{EXT}(N_1) = \dots = \text{EXT}(N_m) \quad (14)$$

$$\text{MinMakespan} = \frac{\sum_{1 \leq k \leq n} \text{length}(T_k)}{\sum_{1 \leq i \leq n} \text{CPUrate}(N_i)} \quad (15)$$

MinTotalCost کمترین هزینه‌ای است که برای انجام مجموعه‌ای از وظایف T در سیستم مه و ابر لازم است و زمانی به دست می‌آید که هر وظیفه به ارزان‌ترین گره اختصاص داده شود. با ارائه اطلاعات هر گره، به راحتی می‌توان تعیین کرد که کدام گره وظیفه T_k را با کمترین هزینه پردازش می‌کند که با $\text{MinCost}(T_k)$ نمایش داده می‌شود، بنابراین، MinTotalCost یک مجموعه از وظایف T خاص است و می‌تواند به صورت معادله 16 تعریف شود:

$$\text{MinTotalCost} = \sum_{T_k \in T} \text{Min cost}(T_k) = \sum_{T_k \in T} \text{Min}_{1 \leq i \leq m} (\text{Cost}(T_k^i)) \quad (16)$$

با توجه به مطالب ارایه شده یک تابع هدف مناسب می‌تواند به صورت معادله 17، در نظر گرفته شود:

$$\min(F) = \alpha * \frac{\text{MinMakespan}}{\text{Makespan}} + (1 - \alpha) * \frac{\text{MinTotalCost}}{\text{TotalCost}} \quad (17)$$

اگر Alpha برابر 0/5 باشد؛ یعنی زمان و هزینه در بهینه‌سازی T اولویت یکسانی دارند. زمانی که Alpha از 0/5 بیشتر باشد نشان دهنده آن است که حداقل رساندن زمان ساخت با اولویت بالاتر از هزینه کل تمرکز می‌کند، این مورد، زمانی است که کاربر مایل است پول بیشتری برای دستیابی به عملکرد بهتر بپردازد. برعکس، زمانی که Alpha از 0/5 کمتر است، هزینه بیشتر از زمان، اولویت بندی می‌شود یعنی کاربر، بودجه محدودی دارد. هدف، بهینه سازی زمان اجرا و هزینه پردازش است، یعنی یافتن راه حل به گونه ای که TotalCost حداقل و نزدیک به MinTotalCost باشد.

3-3- زمانبندی با الگوریتم سوسک سرگین

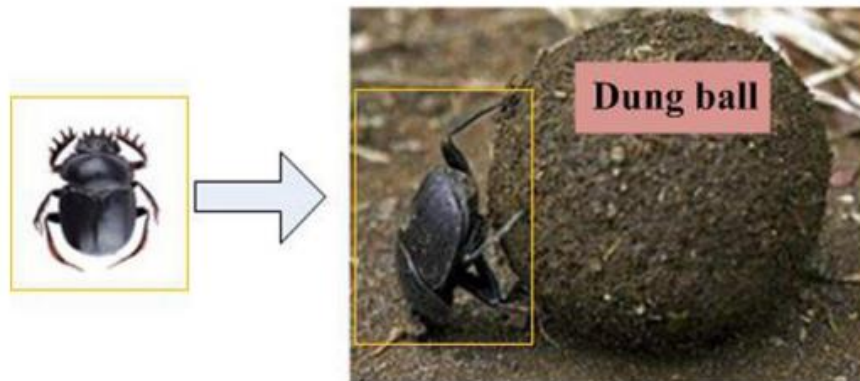
در روش پیشنهادی برای بهینه‌سازی نگاشت وظایف به منابع آزاد در لایه مه و ابر از الگوریتم بهینه‌سازی سوسک سرگین استفاده می‌شود. در روش پیشنهادی هر نگاشت از مجموعه‌ای از وظایف روی تعدادی منبع به عنوان یک راه‌حل در الگوریتم بهینه‌سازی سوسک سرگین در نظر گرفته می‌شود. در شکل 7، نگاشت مجموعه‌ای از وظایف به منابع در لایه مه و ابر نمایش داده شده است که این ماتریس را می‌توان یک سوسک سرگین در نظر گرفت. در اینجا فرض می‌شود که 10 وظیفه وجود دارد که روی سه منبع آزاد زمانبندی می‌شوند. صفر نشان دهنده آن است که وظیفه در منبع اجراء نمی‌شود و یک نشان دهنده اجرای وظیفه در منبع است [34].

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
N_1	0	1	0	0	1	0	0	0	1	0
N_2	0	0	0	1	0	1	1	0	0	0
N_3	1	0	1	0	0	0	0	1	0	1

شکل 7: ساختار یک راه حل در روش پیشنهادی برای زمانبندی وظایف

Figure 7. Structure of a solution in the proposed method for task scheduling

الگوریتم بهینه‌سازی سوسک سرگین از رفتار سوسک‌های سرگین مدلسازی و فرموله شده است. گونه‌های مختلفی برای سوسک‌های سرگین وجود دارد، مانند *Copris ochus* Motschulsky، *Onthophagus gibbulus*، *Onthophagus gibbulus*، *Caccobius jessoensis* Harold و غیره. معروف است که سوسک سرگین به عنوان یک حشره معمولی در طبیعت از سرگین حیوانات تغذیه می‌کند. سوسک‌های سرگین در اکثر نقاط جهان یافت می‌شوند و به عنوان تجزیه کننده در طبیعت عمل می‌کنند، به این معنا که آنها در اکوسیستم، اهمیت حیاتی دارند. تحقیقات نشان داده است که سوسک‌های سرگین عادت جالبی دارند؛ سرگین را به شکل توپ در می‌آورند سپس آن را حرکت در می‌آورند، همانطور که در شکل 8، نشان داده شده است. شایان ذکر است هدف سوسک‌های سرگین این است که گوی سرگین خود را به سرعت و کارآمدتر حرکت دهند که این کار می‌تواند از تکمیل شدن آنها توسط سایر سوسک‌های سرگین جلوگیری کند [17].

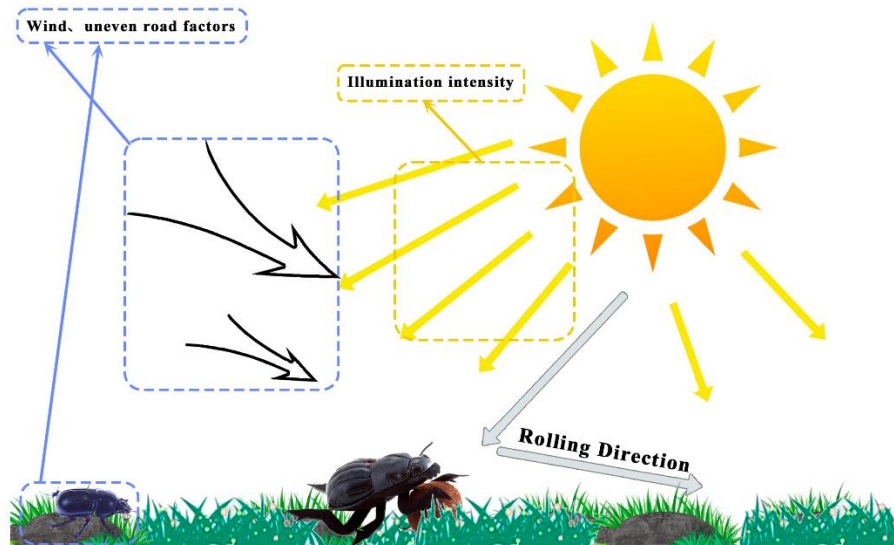


شکل 8: حرکت دادن سرگین توسط سوسک سرگین

Figure 8. Dung moving by dung beetles

الگوریتم بهینه‌سازی سوسک سرگین یک الگوریتم جدید بهینه‌سازی هوش ازدحامی است که از رفتار اجتماعی جمعیت سوسک‌های سرگین الهام گرفته شده است. این سوسک‌ها سرگین را قبل از اینکه به مکانی امن بغلتانند، به صورت توپ، فشرده می‌کنند. آنها می‌توانند توپ‌های سرگین بسیار بزرگ‌تری را بغلتانند و از نشانه‌های آسمانی برای چرخاندن آنها در یک خط مستقیم در صورت وجود منبع نور استفاده کنند. با این حال، در غیاب منبع نور، مسیرهای آنها منحنی و مستعد اختلالات طبیعی می‌شود. بقای سوسک‌های سرگین به طور پیچیده‌ای با به دست آوردن گلوله‌های سرگین مرتبط است، جایی که برخی برای تولید مثل و پرورش فرزندان‌شان استفاده می‌شوند در حالی که بقیه به عنوان غذا مورد استفاده قرار می‌گیرند. الگوریتم بهینه‌سازی سوسک سرگین پنج رفتار کلیدی را که توسط سوسک‌های سرگین به کار گرفته می‌شود، شبیه‌سازی می‌کند و این رفتارها توپ غلتاندن، رقصیدن، جستجوی غذا، دزدی و تولید مثل است. جمعیت سوسک‌های سرگین به چهار زیر گروه تقسیم می‌شوند که عبارتند از: غلتک‌ها، بازتولیدکنندگان، خردسالان و دزدها و هر نوع سوسک استراتژی‌های جستجوی متفاوتی برای خود در نظر می‌گیرند [17]. در طول فرآیند غلتیدن، سوسک‌های سرگین باید با استفاده از نشانه‌های آسمانی، به‌ویژه خورشید و ماه حرکت

کنند تا مسیر حرکت مستقیم توپ سرگین را حفظ کنند. در شکل 9، مدل مسیر سوسک سرگین نمایش داده شده است. می توان مشاهده کرد که سوسک های سرگین از خورشید برای جهت یابی استفاده می کنند، با فلش نشان دهنده، جهت چرخش آنها مشخص شده است. [35].



شکل 9: حرکت دادن سرگین توسط سوسک سرگین

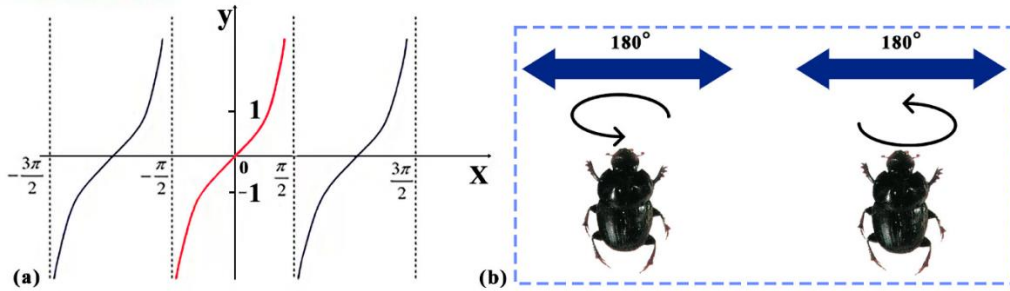
Figure 9. Dung moving by dung beetles

با فرض اینکه شدت منبع نور نیز بر مسیر غلتهای سوسک های سرگین تأثیر می گذارد، موقعیت سوسک به روز می شود و می توان آن را به صورت معادله 18، نشان داد:

$$X_i(t+1) = X_i(t) + \alpha \times k \times X_i(t+1) + b \times \Delta x \quad (18)$$

$$\Delta x = |X_i(t) - X^w| \quad (19)$$

در این معادلات t نشان دهنده تعداد تکرار فعلی است. $X_i(t)$ نشان دهنده اطلاعات موقعیت سوسک i ام در تکرار t است. پارامتر k یک عدد ثابت بین 0 و 0/2 است که ضریب انحراف را نشان می دهد. b یک مقدار ثابت متعلق به بازه صفر و یک است. α یک ضریب طبیعی است که به صورت 1 یا -1 اختصاص داده می شود. X^w نشان دهنده بدترین موقعیت است و Δx تغییر شدت نور را شبیه سازی می کند. پارامتر a عوامل طبیعی (مانند باد و زمین ناهموار) را شبیه سازی می کند که می تواند باعث انحراف سوسک سرگین از جهت اصلی خود شود. به طور خاص، $a = 1$ نشان دهنده عدم انحراف است، در حالی که $a = -1$ نشان دهنده انحراف از جهت اصلی است. Δx بزرگتر به معنی منبع نور ضعیف تر است که دو مزیت را به همراه دارد که عبارتند از: کاوش کامل کل فضای مشکل در طول فرآیند بهینه سازی و افزایش قابلیت های جستجو و کاهش احتمال به دام افتادن در بهینه محلی. عوامل طبیعی مختلف، مانند باد و زمین ناهموار، می توانند تأثیری قابل توجه بر مسیر حرکت سوسک های سرگین داشته باشند. در چنین شرایطی، سوسک های سرگین معمولاً به روی گوی سرگین می روند و رفتار رقصانی را انجام می دهند که شامل یک سری چرخش و مکث است. آنها از طریق این رفتار رقصی، جهت حرکت خود را با تغییر جهت تعیین می کنند و در نتیجه مسیر جدیدی را به دست می آورند. برای تقلید از این رفتار رقصیدن، یک تابع مماس برای به دست آوردن جهت چرخش جدید استفاده می شود. شکل 10، مدل تابع مماس و مدل رقص سوسک های سرگین را نشان می دهد. توجه به این نکته مهم است که فقط مقادیر تعریف شده در بازه 0 و 1 تابع مماس باید در نظر گرفته شوند [35]:



شکل 10: تابع مماس و مدل رقص سوسک های سرگین. (الف) تابع مماس در جهت نور. (ب) مدل رقص سوسک های سرگین.
Figure 10. Tangent function and dance model of dung beetles. (a) Tangent function in light direction. (b) Dance model of dung beetles

هنگامی که جهت صحیح با موفقیت مشخص شد، سوسک غلتکی باید به چرخاندن توپ به جلو ادامه دهد. در این مرحله، به روزرسانی موقعیت سوسک غلتکی به شرح معادله 20، است:

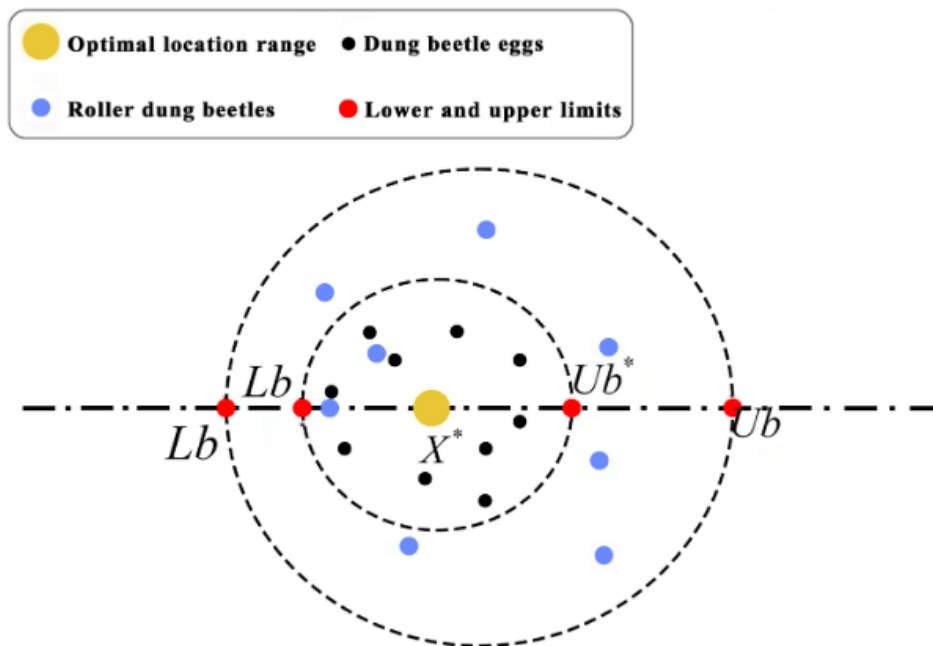
$$X_i(t+1) = X_i(t) + \tan \beta |X_i(t) - X_i(t-1)| \quad (20)$$

که در آن زاویه انحراف $\beta \in [0, \pi]$ است و در معادله، t تعداد تکرارهای جاری را نشان می دهد. $X_i(t)$ اطلاعات موقعیت سوسک غلتکی i در تکرار t را نشان می دهد. $|X_i(t) - X_i(t-1)|$ تفاوت مطلق بین موقعیت i ام سوسک در تکرار t و موقعیت آن در تکرار قبلی $(t-1)$ است. می توان مشاهده کرد که به روز رسانی موقعیت سوسک غلتکی ارتباط نزدیکی با اطلاعات موقعیت فعلی و سابق آن دارد. توجه به این نکته ضروری است که اگر زاویه انحراف برابر با 0 یا $\pi/2$ یا π باشد، موقعیت سوسک به روز نمی شود. برای ایجاد یک محیط امن برای فرزندان خود، انتخاب محل تخمگذاری مناسب برای سوسک های سرگین بسیار مهم است. با الهام از بحث فوق الذکر، یک استراتژی انتخاب مرز برای شبیه سازی منطقه ای که سوسک های ماده سرگین تخم می گذارند پیشنهاد شده است و به صورت زیر معادله 21 و 22 فرموله می شود:

$$\min(X^* * (1-R), Uh) \quad (21)$$

$$\min(X^* * (1-R), Lh) \quad (22)$$

که در آن X^* نشان دهنده بهترین موقعیت محلی فعلی است. Lh^* و Uh^* به ترتیب مرزهای بالایی و پایینی ناحیه تخمگذاری را نشان می دهند. $R = 1 - t/T_{max}$ حداکثر تعداد تکرار را نشان می دهد. Lh و Uh به ترتیب کران های بالایی و پایینی فضای جستجو هستند، همانطور که در شکل 11، نشان داده شده است [35].



شکل 11: مدل مفهومی استراتژی انتخاب مرز

Figure 11. Conceptual model of boundary selection strategy

شایان ذکر است که هر گلوله سرگین حاوی یک تخم یک سوسک سرگین است. علاوه بر این، نقاط قرمز مرزهای بالایی و پایینی را نشان می‌دهد. در الگوریتم سوسک سرگین فرض بر این است که هر سوسک سرگین ماده در هر تکرار فقط یک تخم می‌گذارد. با توجه به تغییرات دینامیکی در محدوده مرزی در طول تکرار، این به جلوگیری از به دام افتادن الگوریتم در بهینه محلی کمک می‌کند که عمدتاً توسط مقدار وزن اینرسی R تعیین می‌شود. بنابراین، موقعیت توپ‌های تخم‌مرغ نیز در طول فرآیند تکرار پویا است و با معادله 23 می‌شود:

$$Y_i(t+1) = X^* + b_1(Y_i(t) - Lh^*) + b_2(Y_i(t) - Uh^*) \quad (23)$$

که در آن $Y_i(t)$ اطلاعات موقعیت i امین توپ تخم مرغ را در تکرار t -امین نشان می‌دهد، X^* نشان دهنده بهترین موقعیت محلی است و از طرفی Lh^* و Uh^* نشان دهنده مرزهای بالایی و پایینی ناحیه تخم‌گذاری است. از طرفی $(Y_i(t) - Lh^*)$ ، b_2 یک بردار تصادفی در محدوده 0 و 1 است و D بعد مسئله بهینه سازی را نشان می‌دهد. برخی از سوسک‌های سرگین بالغ در جستجوی غذا در زمین فرو می‌روند و این نوع سوسک‌ها به عنوان سوسک‌های سرگین کوچک شناخته می‌شوند. برای شبیه‌سازی فرآیند جستجوی علوفه سوسک‌های سرگین کوچک، تعیین منطقه بهینه علوفه ضروری است. معادله شبیه‌سازی برای این ناحیه به صورت زیر معادله 24 و 25 مشخص می‌شود:

$$Lm = \max(X^h * (1 - R), Lh) \quad (24)$$

$$Lm = \min(X^h * (1 - R), Lh) \quad (25)$$

به گونه‌ای که Xh نشان دهنده بهترین موقعیت سراسری است. Lm و Um به ترتیب مرزهای پایین و بالایی منطقه جستجوی بهینه هستند. Lh و Uh مرزهای پایین و بالایی فضای جستجو هستند. بنابراین، به روز رسانی موقعیت برای سوسک‌های سرگین کوچک به شرح 26، است:

$$X_i(t+1) = X_i(t) + C_1 * (X_i(t) - Lm) + C_2 * (X_i(t) - U_m) \quad (26)$$

که در آن $X_i(t)$ نشان دهنده موقعیت i -امین سوسک سرگین در تکرار t -ام است. $C_1 - (t)$ عددی است که به طور تصادفی به دنبال توزیع نرمال ایجاد می‌شود و C_2 یک بردار تصادفی در محدوده 0 و 1 است. با توجه به اینکه برخی از سوسک‌های سرگین که به آنها دزد می‌گویند، توپ‌های سرگین را از سایر سوسک‌ها می‌دزدند، اطلاعات موقعیت سارق در طول فرآیند تکرار به صورت زیر به روز رسانی می‌شود. بر اساس رابطه 26، می‌توان مشاهده کرد که Xh منبع غذایی بهینه را نشان می‌دهد. با فرض اینکه منطقه اطراف Xh نشان دهنده مکان اصلی برای غذای رقابتی است، به روز رسانی موقعیت برای دزد مطابق معادله (27)، ارائه می‌شود:

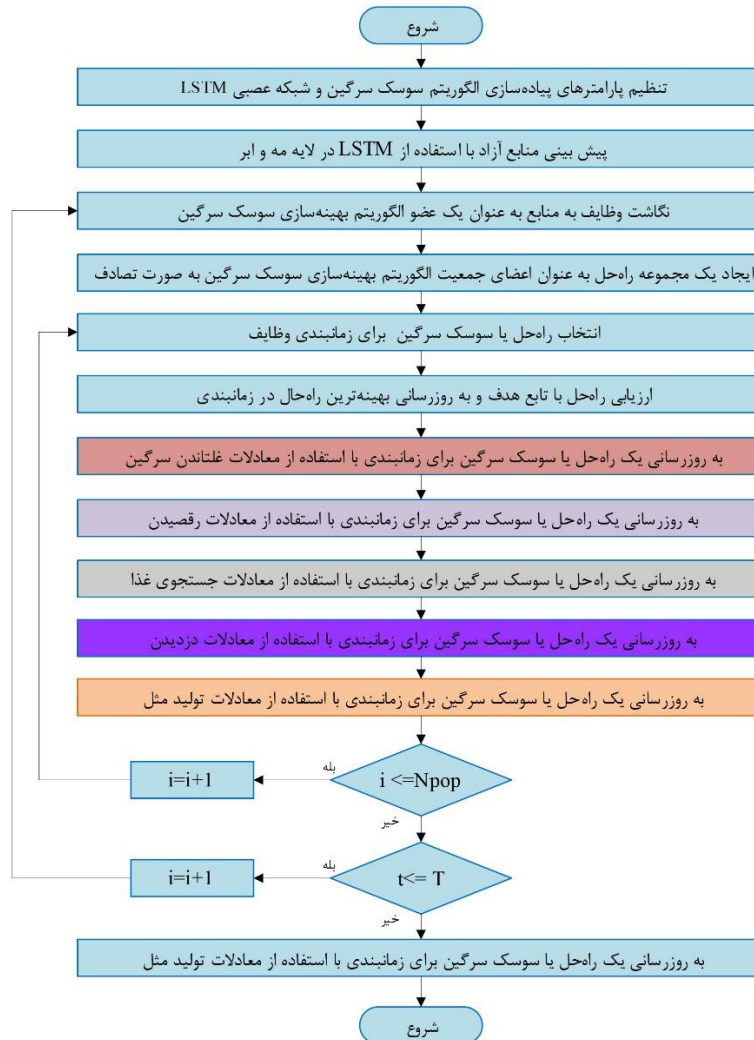
$$X_i(t+1) = X^h + W * g * |X_i(t) - X^*| + |X_i(t) - X^h| \quad (27)$$

که در آن $X_i(t)$ اطلاعات موقعیت i -امین سارق را در تکرار t -امین نشان می‌دهد، Xh نشان دهنده بهترین موقعیت سراسری، X^* نشان دهنده بهترین موقعیت محلی فعلی، $|X_i(t) - X^*|$ و $|X_i(t) - Xh|$ به ترتیب نشان دهنده تفاوت‌های مطلق بین موقعیت فعلی و بهترین موقعیت‌های محلی و جهانی است. g یک بردار تصادفی با اندازه 1 در D و یک دنباله توزیع نرمال است و W یک ثابت است. اگرچه الگوریتم سوسک سرگین دارای قابلیت‌های بهینه‌سازی قوی و سرعت هم‌گرایی سریع است، اما از عدم تعادل بین اکتشاف جهانی و توانایی‌های بهره‌برداری محلی رنج می‌برد و آن را مستعد گیر کردن در بهینه محلی و نشان دادن قابلیت‌های ضعیف اکتشاف جهانی می‌کند. در شکل 11، فلوچارت روش پیشنهادی برای تخصیص منابع و زمانبندی وظایف و کارها ارائه شده است. در روش پیشنهادی دو فاز اصلی ذیل وجود دارد:

▪ پیش‌بینی منابع آزاد با استفاده از شبکه عصبی LSTM

▪ زمانبندی بهینه وظایف با استفاده از الگوریتم بهینه‌سازی سوسک سرگین

در روش پیشنهادی هر نگاشت وظایف به منابع به عنوان یک راه‌حل یا یک سوسک سرگین در نظر گرفته می‌شود و توسط این الگوریتم تلاش می‌شود تا بهینه‌ترین سوسک یا زمانبندی وظایف محاسبه شود.



شکل 12: فلوچارت پیشنهادی برای زمانبندی وظایف
Figure 12. Suggested flowchart for task scheduling

4- نتایج تجربی

در این بخش روش پیشنهادی برای زمانبندی کارها و وظایف مورد ارزیابی قرار گرفته می‌شود. در بخش اول شاخص‌های ارزیابی برای پیش‌بینی وضعیت اشغال یا آزاد بودن منابع معرفی شده و سپس شاخص‌های ارزیابی برای زمانبندی نیز ارائه می‌شود سپس پارامترهای به کار رفته برای پیاده‌سازی‌ها نیز معرفی می‌شود. در ادامه نیز با آزمایشات روش پیشنهادی در Malab پیاده‌سازی و با روشهای مشابه مقایسه می‌شود.

4-1- شاخص‌های ارزیابی

برای ارزیابی روش پیشنهادی در فاز پیش‌بینی وضعیت منابع از شاخص‌های دقت، حساسیت و صحت استفاده می‌شود که ضابطه آنها به ترتیب در رابطه 28، 29 و 30 فرموله شده است.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (28)$$

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \quad (29)$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (30)$$

مفهوم هر کدام از شاخص‌های ارزیابی از جمله TP، FN و FP برای زمانبندی به شرح ذیل است:

- نمونه‌های صحیح مثبت (TP): منبع به کار رفته دارای بار عملیاتی زیادی است و روش پیشنهادی نیز به درستی پیش‌بینی کرده این منبع دارای بار عملیاتی زیاد است.
- نمونه‌های غلط مثبت (FP): منبع به کار رفته دارای بار عملیاتی کم است و روش پیشنهادی نیز به اشتباه پیش‌بینی کرده این منبع دارای بار زیاد است.
- نمونه‌های صحیح منفی (TN): منبع به کار رفته دارای بار عملیاتی اندک است و روش پیشنهادی نیز به درستی پیش‌بینی کرده این منبع دارای بار عملیاتی اندک است.
- نمونه‌های غلط منفی (FN): منبع به کار رفته دارای بار عملیاتی زیادی است و روش پیشنهادی نیز به اشتباه پیش‌بینی کرده این منبع دارای بار عملیاتی اندک است.

علاوه بر این شاخص‌ها، برای زمانبندی وظایف می‌توان به شاخص‌های مانند تأخیر زمانبندی و makespan اشاره کرد. کمینه نودن تین مقادیر نشان دهنده آن است که الگوریتم زمانبندی به کار رفته دارای کارایی بالایی برای زمانبندی وظایف در لایه مه و ابر محاسباتی است.

2-4- پارامترها

در جدول (1)، مجموعه‌ای از پارامترهای به کار رفته برای شبیه‌سازی و پیاده‌سازی در نرم‌افزار Matlab نشان داده شده است. در این جدول، اطلاعات مرتبط با ماشین‌های فیزیکی و مجازی به عنوان منابع به کار رفته نمایش داده شده است.

جدول 1: پارامترهای مرتبط با پیاده‌سازی مرتبط با منابع

Table 1. Implementation-related parameters related to resources

	Specification	Amount
Client	Clients Count	[60, 120]
	Hosts Count	6
Physical Machine	CPU capacity	[100, 5000]
	Storage	1 TB
	Network Bandwidth	10 Gb/s
	RAM size	6 GB
	VMs count	25
Virtual Machine	CPU capacity	[100, 5000]
	Storage	20 GB
	RAM size	1 GB
	Network Bandwidth	1 Gb/s
	Processor	Xen
	Processors' count	1

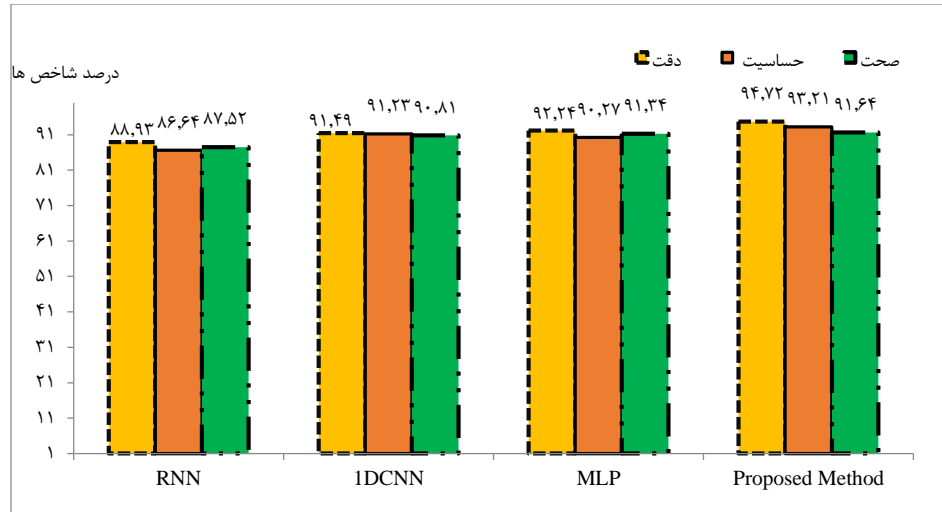
علاوه بر این پارامترها، برخی از پارامترها نیز مرتبط با الگوریتم بهینه‌سازی سوسک سرگین است و به عنوان مثال اندازه جمعیت اولیه برابر 15 و تعداد تکرار الگوریتم برابر 50 و هر آزمایش 25 مرتبه تکرار می‌شود تا متوسط شاخص‌ها محاسبه شود.

3-4- مجموعه داده

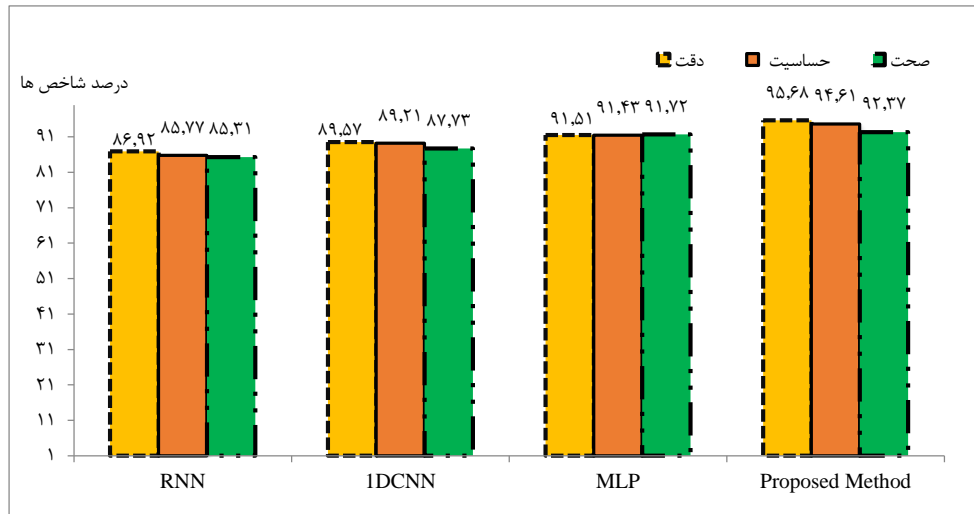
برای پیاده‌سازی‌ها دو مجموعه داده مختلف، HPC2N و NASA استفاده شده است که هر کدام دارای 500، 1000، 1500 و 2000 وظیفه هستند. مجموعه داده iPSC ناسا مربوط به 257 کاربر است، به 240 پردازنده یا CPU نیاز است. میانگین تعداد وظایف در این مجموعه داده 202/871 است. در مجموع 128 پردازنده یا CPU برای مجموعه داده HPC2N مورد نیاز است که مربوط به 69 کاربر است. میانگین تعداد وظایف در این مجموعه داده 18/239 است [30].

4-4- پیش‌بینی در تخصیص منابع

در این بخش شبکه عصبی LSTM روی مجموعه‌ای از ویژگی‌ها مرتبط با مجموعه داده HPC2N و NASA آموزش داده می‌شود. در نمودار شکل 13 و 14 به ترتیب شاخص دقت، حساسیت و صحت روش پیشنهادی در این دو مجموعه داده با شبکه عصبی RNN، 1DCNN، MLP مورد مقایسه قرار گرفته است.

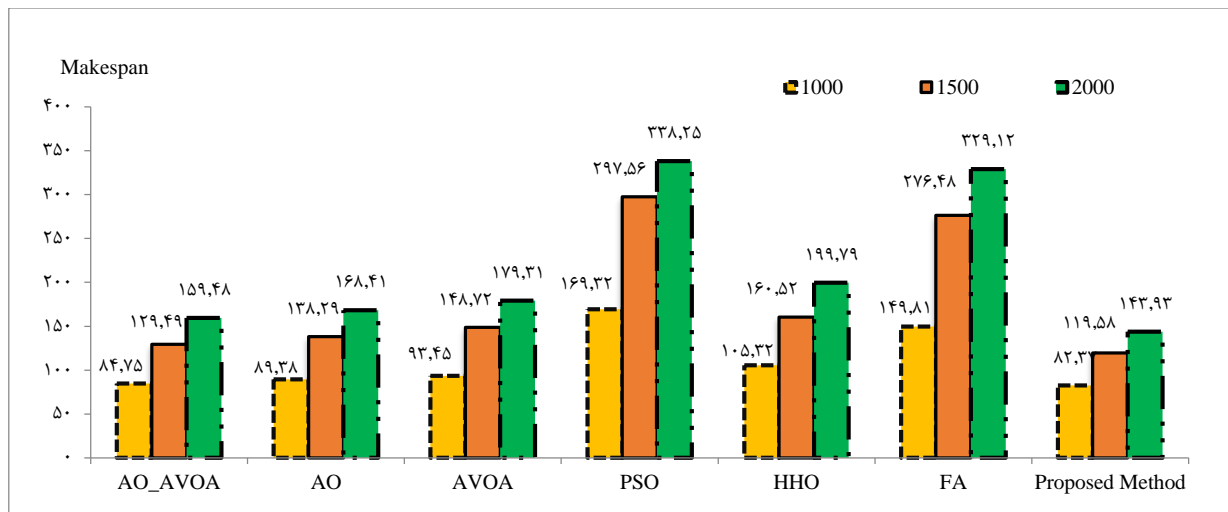


شکل 13: مقایسه دقت، حساسیت و صحت در پیش بینی وضعیت منابع در مجموعه داده HPC2N
Figure 13. Comparison of accuracy, sensitivity and accuracy in predicting resource status in HPC2N dataset

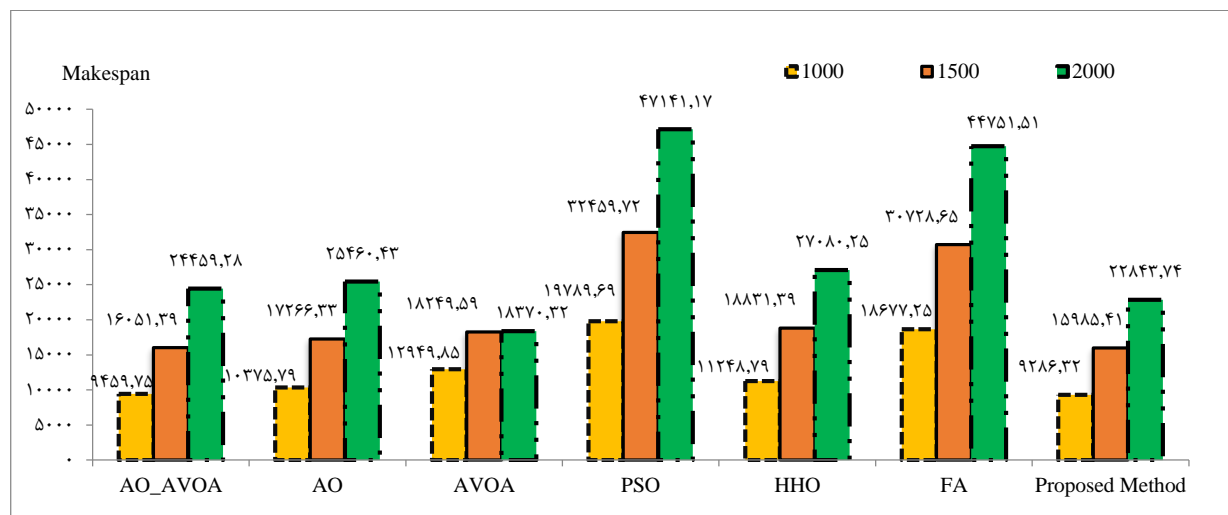


شکل 14: مقایسه دقت، حساسیت و صحت در پیش بینی وضعیت منابع در مجموعه داده NASA
Figure 14. Comparison of accuracy, sensitivity and accuracy in predicting resource status in NASA dataset

آزمایشات در مجموعه داده HPC2N نشان می‌دهد که دقت، حساسیت و صحت روش پیشنهادی برای پیش بینی وضعیت منابع (آزاد یا مشغول) به ترتیب برابر 94/72 درصد، 93/21 درصد و 91/64 درصد است. روش پیشنهادی در مجموعه داده HPC2N نسبت به روش‌های RNN، IDCNN، MLP، دقت بیشتری در پیش بینی وضعیت منابع برای زمانبندی دارد. در مجموعه داده NASA، حساسیت و صحت پیش بینی روش پیشنهادی به ترتیب برابر 95/68 درصد، 94/61 درصد و 92/37 درصد است. با توجه به آزمایشات در مجموعه داده NASA، دقت روش پیشنهادی و سایر روش‌ها نسبت به مجموعه داده HPC2N بیشتر است که دلیل آن می‌تواند دقت گردآوری در مجموعه داده NASA باشد. در هر دو مجموعه داده روش پیشنهادی نسبت به سه روش پیش‌بینی کننده دیگر موفق‌تر بوده است و بدترین عملکرد در پیش‌بینی منابع و تخصیص آنها مرتبط با RNN است. برای فاز زمانبندی روش پیشنهادی در دو مجموعه داده HPC2N و NASA در شاخصی نظیر Makespan با روش‌های فراابتکاری مقایسه شده است و برای مقایسه از نتایج به‌دست آمده در پژوهش [30]، در سال 2023 استفاده می‌شود که در نمودار شکل 15 و 16 مقایسه شده است.

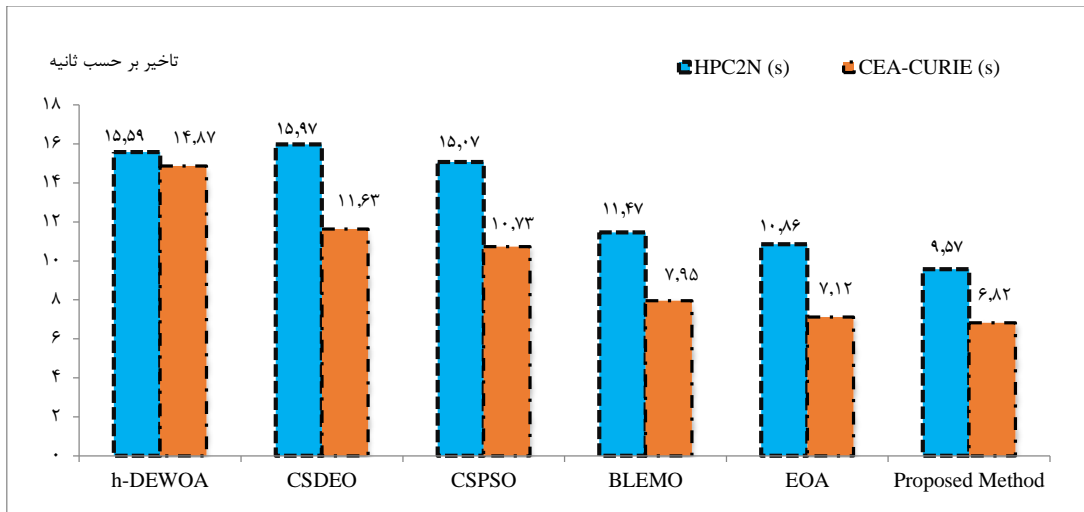


شکل 15: مقایسه شاخص Makespan در مجموعه داده NASA به ازای 1000، 1500 و 2000 وظیفه
Figure 15. Comparison of Makespan index in NASA dataset for 1000, 1500 and 2000 tasks



شکل 16: مقایسه شاخص Makespan در مجموعه داده HPC2N به ازای 1000، 1500 و 2000 وظیفه
Figure 16. Comparison of Makespan index in HPC2N dataset for 1000, 1500 and 2000 tasks

آزمایشات نشان می‌دهد در دو مجموعه HPC2N و NASA شاخص Makespan روش پیشنهادی نسبت به روش‌های AO_AVOA، AVOA، PSO، HHO و FA دارای مقادیر کمتری است و تین موضوع نشان می‌دهد روش پیشنهادی نسبت به این روش‌ها سریع‌تر وظایف را در لایه مه و ابر اجرا می‌کند. در نمودار شکل 17، تاخیر و زمان اجرای الگوریتم پیشنهادی در دو مجموعه داده HPC2N و CEA-CURIE با روش‌های h-DEWOA، CSDEO، CSPSO، BLEMO و EOA مطابق پژوهش [36]، که در سال 2023 انجام شده است، نمایش داده می‌شود.



شکل 17: مقایسه تاخیر در اجرای زمانبندی وظایف

Figure 17. Comparison of delays in the implementation of task scheduling

آزمایشات نشان داد تاخیر روش پیشنهادی در دو مجموعه داده HPC2N و CEA-CURIE به ترتیب برابر 6/82 و 9/57 ثانیه است و نسبت به روش‌های h-DEWOA، CSDEO، CSPSO، BLEMO و EOA دارای تاخیر کمتری در زمانبندی وظایف در لایه مه و ابر است.

5- نتیجه گیری

پتانسیل برجسته پلتفرم‌های ابری، پردازش و ذخیره‌سازی کلان داده‌های جمع‌آوری شده از تجهیزات اینترنت اشیا را تسریع می‌کند. طرح‌های پردازش مبتنی بر مه می‌توانند کیفیت خدمات را برای برنامه‌های اینترنت اشیا بهبود بخشند و تاخیرهای بیش از حد و چالش‌های امنیتی را کاهش دهند. همچنین، از آنجایی که مصرف انرژی می‌تواند مستقیماً باعث انتشار CO2 از مه و گره‌های ابری شود، یک روش زمان‌بندی کار کارآمد، مصرف انرژی را کاهش می‌دهد. در این راستا، نیاز روزافزون به یک مکانیسم زمان‌بندی وظایف کارآمد با در نظر گرفتن مدیریت بهینه منابع اینترنت اشیا به طور فزاینده‌ای احساس می‌شود. زمان‌بندی وظایف اینترنت اشیا بر اساس محاسبات مه-ابر نقش مهمی در پاسخ به درخواست‌های کاربران دارد. برنامه‌ریزی بهینه وظایف می‌تواند عملکرد سیستم را بهبود بخشد. بنابراین، این مطالعه از یک روش زمان‌بندی درخواست وظایف اینترنت اشیا بر روی منابع توسط الگوریتم بهینه‌سازی سوسک سرگین استفاده می‌کند. کیفیت خدمات اینترنت اشیا مبتنی بر محاسبات مه-ابر را افزایش می‌دهد تا زمان تکمیل درخواست‌های کار و زمان عملیات سیستم و مصرف انرژی را کاهش دهد. اگر مصرف انرژی کاهش یابد، درصد انتشار CO2 نیز کاهش می‌شود. سپس روش زمان‌بندی پیشنهادی برای حل مسئله زمان‌بندی کار با استفاده از مجموعه داده‌ها ارزیابی می‌شود. در این مقاله یک نسخه زمان‌بندی برای وظایف در لایه مه و ابر ارائه شده است. در روش پیشنهادی با استفاده از شبکه عصبی LSTM منابع آزاد پیش‌بینی و تخصیص داده می‌شود و سپس با استفاده از الگوریتم بهینه‌سازی سوسک سرگین تلاش می‌شود تا زمان‌بندی بهینه برای وظایف و کارها محاسبه شود. آزمایشات نشان داد دقت تخصیص منابع در لایه مه و ابر در روش پیشنهادی بیشتر از شبکه عصبی MLP، IDCNN، RNN است. آزمایشات نشان داد شاخص Makespan روش پیشنهادی در زمان‌بندی کارها در لایه مه و ابر نسبت به روش‌های AO_AVOA، AVOA، PSO، HHO و FA کمتر است. در کارهای آتی تلاش می‌شود برای پیش‌بینی منابع زمان‌بندی در لایه مه و لایه ابر از ترکیب شبکه CNN-LSTM استفاده شود. یکی از پیشنهادها آتی ما بهبود الگوریتم بهینه‌سازی سوسک سرگین با توابع آشوبناک در زمان‌بندی وظایف در لایه مه و ابر است.

مراجع

- [1] A. S. Abohamama, A. El-Ghamry & E. Hamouda, "Real-time task scheduling algorithm for IoT-based applications in the cloud-fog environment," *Journal of Network and Systems Management*, vol. 30, no.4, 1-35, 54, 27 May 2022, doi: 10.1007/s10922-022-09664-6.
- [2] D. R. Prapti, A. R. Mohamed Shariff, H. Che Man, N. M. Ramli, T. Perumal, & M. Shariff, "Internet of Things (IoT)-based aquaculture: An overview of IoT application on water quality monitoring," *Reviews in Aquaculture*, vol. 14, no. 2, pp. 979-992, 19 November 2021, doi: 10.1111/raq.12637.

- [3] I. Attiya, M. Abd Elaziz, L. Abualigah, T. N. Nguyen, & A. A. Abd El-Latif, "An improved hybrid swarm intelligence for scheduling iot application tasks in the cloud," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6264-6272, 04 February 2022, doi: 10.1109/TII.2022.3148288
- [4] A. Rahimikhanghah, M. Tajkey, B. Rezazadeh, & A. M. Rahmani, "Resource scheduling methods in cloud and fog computing environments: a systematic literature review," *Cluster Computing*, vol. 25, pp. 911-945, 1-35. April 2022, doi: 10.1007/s10586-021-03467-1.
- [5] M. T. Zhou, T. F. Ren, Z. M. Dai, & X. Y. Feng, "Task scheduling and resource balancing of fog computing in smart factory," *Mobile Networks and Applications*, vol. 28, no. 1, pp. 19-30. February 2023, doi: 10.1007/s11036-022-01992-w.
- [6] S. Subbaraj, R. Thiyagarajan, & M. Rengaraj, "A smart fog computing based real-time secure resource allocation and scheduling strategy using multi-objective crow search algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 2, pp. 1003-1015. February 2023, doi: 10.1007/s12652-021-03354-y.
- [7] M. R. Raju, & S. K. Mothku, "Delay and energy aware task scheduling mechanism for fog-enabled IoT applications: A reinforcement learning approach," *Computer Networks*, vol. 224, 109603, 8 February 2023, doi: 10.1016/j.comnet.
- [8] H. Wadhwa, & R. Aron, "Optimized task scheduling and preemption for distributed resource management in fog-assisted IoT environment," *The Journal of Supercomputing*, vol. 79, no. 2, pp. 2212-2250, February 2023, doi : 10.1007/s11227-022-04747-2.
- [9] T. K. Vashishth, , V. Sharma, K. K. Sharma, B. Kumar, S. Chaudhary, & R. Panwar, (2024). "Intelligent Resource Allocation and Optimization for Industrial Robotics Using AI and Blockchain," In *AI and Blockchain Applications in Industrial Robotics IGI Global*, pp. 82-110, December 2023, doi : 10.4018/979-8-3693-0659-8.ch004.
- [10] Saifeng, Z. (2024). "AQINM: an adaptive QoS management framework based on intelligent negotiation and monitoring in cloud," *International Journal of Information Technology and Management*, vol. 23, no. 1, pp. 33-47, 22 January 2024, doi: 10.1504/IJITM.2024.136183.
- [11] E. Khezri, R. O. Yahya, H. Hassanzadeh, M. Mohaidat, S. Ahmadi, & M. Trik, "DLJSF: Data-Locality Aware Job Scheduling IoT tasks in fog-cloud computing environments," *Results in Engineering*, vol. 21, 101780. 24 January 2024, doi: 10.1016/j.rineng.2024.101780.
- [12] Y. Lin, Y. Xu, J. Zhu, X. Wang, L. Wang, & G. Hu, "MLATSO: A method for task scheduling optimization in multi-load AGVs-based systems," *Robotics and Computer-Integrated Manufacturing*, vol. 79, 102397, February 2023, doi: 10.1016/j.rcim.2022.102397.
- [13] Y. Shen, & H. Li, "A multi-strategy genetic algorithm for solving multi-point dynamic aggregation problems with priority relationships of tasks," *Electronic Research Archive*, vol.32, no. 1, pp. 445-472, 2024, doi: 10.3934/era.2024022.
- [14] X. Fu, Y. Sun, H. Wang, & H. Li, "Task scheduling of cloud computing based on hybrid particle swarm algorithm and genetic algorithm," *Cluster Computing*, vol. 26, no. 5, pp. 2479-2488, October 2023, doi: 10.1007/s10586-020-03221-z.
- [15] S. Mangalampalli, S. K. Swain, G. R. Karri, & S. Mishra, "SLA Aware Task-Scheduling Algorithm in Cloud Computing Using Whale Optimization Algorithm," *Scientific Programming*, vol. 2023, 20 Apr 2023, doi: 10.1155/2023/8830895.
- [16] S. Mangalampalli, G. R. Karri, S. N. Mohanty, S. Ali, M. I. Khan, D. Abduvalieva, F. A. Awwad & E. A.

- Ismail, "Fault tolerant trust based task scheduler using Harris Hawks optimization and deep reinforcement learning in multi cloud environment," *Scientific Reports*, vol. 13, no. 1, 19179. 06 November 2023, doi.org/10.1038/s41598-023-46284-9.
- [17] J. Xue, & B. Shen, "Dung beetle optimizer: A new meta-heuristic algorithm for global optimization," *The Journal of Supercomputing*, vol. 79, no. 7, pp. 7305-7336, May 2023, doi.org/10.1007/s11227-022-04959-6.
- [18] Z. Yin, F. Xu, Y. Li, C. Fan, F. Zhang, G.Han, & Y. Bi, "A multi-objective task scheduling strategy for intelligent production line based on cloud-fog computing," *Sensors*, vo. 22, no. 4, 1555, 15 February 2022, doi: 10.3390/s22041555.
- [19] A. A. Mutlag, M. Khanapi Abd Ghani, M. A. Mohammed, M. S. Maashi, O. Mohd, S. A. Mostafa, , k. h. Abdulkareem, G. Marques, & I. de la Torre Díez, (2020). "MAFC: Multi-agent fog computing model for healthcare critical tasks management," *Sensors*, vol. 20, no. 7, 1853, 25 March 2020, doi.org: 10.3390/s20071853.
- [20] X. Ma, H. Gao, H. Xu, & M. Bian, "An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no, 249, pp. 1-19, 08 November 2019, doi: 10.1186/s13638-019-1557-3.
- [21] M. Hosseini Shirvani, "A survey study on task scheduling schemes for workflow executions in cloud computing environment: classification and challenges," *The Journal of Supercomputing*, pp. 1-54, 07 December 2023 , doi: 10.1007/s11227-023-05806-y.
- [22] M. Hosseinzadeh, E. Azhir, J. Lansky, S. Mildeova, O. H. Ahmed, M. H. Malik, & F. Khan, "Task Scheduling Mechanisms for Fog Computing: A Systematic Survey," *IEEE Access*, vol. 11, pp. 50994–51017, 18 May 2023, doi: 10.1109/ACCESS.2023.3277826.
- [23] Zhou, M. T., Ren, T. F., Dai, Z. M., & Feng, X. Y. (2023). "Task scheduling and resource balancing of fog computing in smart factory," *Mobile Networks and Applications*, vol. 28, no. 1, pp. 19-30, February 2023, doi: 10.1007/s11036-022-01992-w.
- [24] Z. A. Khan, I. A. Aziz, & N. A. B. Osman, "A Review on Task Scheduling Techniques in Cloud and Fog Computing: Taxonomy, Tools, Open Issues, Challenges, and Future Directions," *IEEE Access*, vol. 11, pp. 143417 – 143445, 18 December 2023, doi: 10.1109/ACCESS.2023.3343877.
- [25] T. Salehnia, A. Seyfollahi, , S. Raziani, A. Noori, A. Ghaffari, A. R. Alsoud, & L. Abualigah, "An optimal task scheduling method in IoT-Fog-Cloud network using multi-objective moth-flame algorithm," *Multimedia Tools and Applications*, pp. 1-22, 11 September 2023, doi: 10.1007/s11042-023-16971-w
- [26] F. Ramezani Shahidani, A. Ghasemi, A. Toroghi Haghighat, & A. Keshavarzi, "Task scheduling in edge-fog-cloud architecture: a multi-objective load balancing approach using reinforcement learning algorithm," *Computing*, vol. 105, no. 6, pp. 1337-1359, June 2023, doi: 10.1007/s00607-022-01147-5.
- [27] B. M. Nguyen, T. Nguyen, Q. H. Vu, H. H. Tran, H. Vo, H. T. T. Binh, S. Yu , Z. Wu, "A novel nature-inspired algorithm for optimal task scheduling in fog-cloud blockchain system," *IEEE Internet of Things Journal*, vol. 11, 06 July 2023, doi: 10.1109/JIOT.2023.3292872.
- [28] J. Z. Ahmadabadi, S. E. Mood, & A. Souri, "Star-quake: A new operator in multi-objective gravitational search algorithm for task scheduling in IoT based cloud-fog computing system," *IEEE Transactions on Consumer Electronics*, January 2023, doi: 10.1109/TCE.2023.3321708.
- [29] R. Ghafari, & N. Mansouri, "E-AVOA-TS: Enhanced African vultures optimization algorithm-based task scheduling strategy for fog-cloud computing," *Sustainable Computing: Informatics and Systems*, vol. 40,

100918, December 2023, doi: 10.1016/j.suscom.2023.100918.

- [30] Q. Liu, H. Kosarirad, S. Meisami, K. A. Alnowibet, & A. N. Hoshyar, “An Optimal Scheduling Method in IoT-Fog-Cloud Network Using Combination of Aquila Optimizer and African Vultures Optimization,” *Processes*, vol. 11, no. 4, 1162, 10 April 2023, doi: 10.3390/pr11041162.
- [31] Z. Wang, M. Goudarzi, M. Gong, & R. Buyya, “Deep Reinforcement Learning-based scheduling for optimizing system load and response time in edge and fog computing environments,” *Future Generation Computer Systems*, vol. 152, pp. 55-69, Mch 2024, doi: /10.1016/j.future.2023.10.012
- [32] M. Osmanpoor, A. Shameli-Sendi, & F. Faraji Daneshgar, “Convergence of the Harris hawks optimization algorithm and fuzzy system for cloud-based task scheduling enhancement,” *Cluster Computing*, pp. 1-15, 09 January 2024, doi.org/10.1007/s10586-023-04225-1.
- [33] D. Sanchez Narvaez, C. Villaseñor, C. Lopez-Franco, & N. Arana-Daniel, “Order-Based Schedule of Dynamic Topology for Recurrent Neural Network,” *Algorithms*, vol. 16, no. 5, 231. 28 April 2023, doi: 10.3390/a16050231.
- [34] B. M. Nguyen, , H. Thi Thanh Binh, T. The Anh, & D. Bao Son, “Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud-fog computing environment,” *Applied Sciences*, vol. 9, no. 9, 1730, 26 April 2019, doi: 10.3390/app9091730.
- [35] L. Li, L. Liu, Y. Shao, X. Zhang, Y. Chen, C. Guo, & H. Nian, “Enhancing Swarm Intelligence for Obstacle Avoidance with Multi-Strategy and Improved Dung Beetle Optimization Algorithm in Mobile Robot Navigation,” *Electronics*, vol. 12, no. 21, 4462, 30 October 2023 , doi: 10.3390/electronics12214462.
- [36] M. S. Kumar, & G. R. Karri, “Eeo: cost and energy efficient task scheduling in a cloud-fog framework,” *Sensors*, vol. 23, no. 5, 2445, 22 February 2023, doi: 10.3390/s23052445.

Tasks scheduling in distributed fog layer and cloud computing systems using dung beetle optimization algorithm

Reza Aziz, Assistant Professor ¹, Mohsen Eghbali, PhD Student ²

Abstract

The Internet of Things has grown significantly in the past few years, and many intelligent objects have been connected to it. Cloud computing is a data processing system in the Internet of Things. However, the servers in the cloud computing paradigm are usually located at a long physical distance from the Internet of Things devices. The high latency caused by long distances cannot effectively implement real-time Internet of Things applications. Edge and fog computing has emerged as a popular computing technology in the field of the Internet of Things. One of the critical challenges of the Internet of Things is the problem of scheduling tasks in the fog and cloud layer. In the proposed method, the LSTM neural network allocates free resources, and the dung beetle optimization algorithm is used to schedule tasks optimally in the cloud and fog layer. Experiments show that in the HPC2N data set, the accuracy, sensitivity, and precision of the proposed method for predicting the state of resources are equal to 94.72%, 93.21%, and 91.64%, respectively. In the NASA data set, the proposed method's accuracy, sensitivity, and precision in resource allocation are 95.68%, 94.61%, and 92.37%, respectively. The proposed method is more accurate in allocating resources for scheduling than the RNN, 1DCNN, and MLP methods. The Makespan index of the proposed method shows a lower and better value in task scheduling than the AO_AVOA, AVOA, PSO, HHO, and FA methods.

Keywords: Cloud layer, Dung beetle optimization algorithm, Fog layer, Task scheduling, Internet of Things.