

تکنیک هوشمند مبتنی بر الگوریتم چتر دریایی برای زمان بندی وظایف بر اساس اولویت در شبکه های IoT/Fog

سکینه سهرابی، مهدی سخایی‌نیا، محمد نصیری و رضا محمدی

و بروز ازدحام در شبکه گردد؛ به‌ویژه زمانی که چندین دستگاه به‌طور هم‌زمان اقدام به ارسال داده می‌کنند [۱]. IoT تأثیر چشمگیری بر فناوری‌های ارتباطی و اطلاعاتی داشته و موجب ظهور طیف وسیعی از کاربردها مانند کنترل هوشمند ترافیک، پایش سلامت، لجستیک، خرده‌فروشی هوشمند و شبکه‌های وسایل نقلیه شده است [۲] تا [۵]. دستگاه‌های IoT با اتصال ابزارهای مختلف خانگی و صنعتی به اینترنت، داده‌های حجیمی تولید می‌کنند که تحلیل آن‌ها با استفاده از کلان‌داده‌ها ضروری است. رایانش ابری برای پردازش این داده‌ها به کار می‌رود، اما می‌تواند باعث افزایش تأخیر و ازدحام در شبکه شود [۶] تا [۱۰]. برای حل این مشکلات، منابع رایانشی به لبه شبکه انتقال یافتند تا کیفیت خدمات (QoS) ارتقا یابد [۱۱]. محاسبات مه، که نخستین بار در سال ۲۰۱۲ توسط شرکت سیسکو مطرح شد، با هدف پردازش سریع‌تر داده‌ها در نزدیکی منابع IoT طراحی شد [۱۲] و [۱۳]. این فناوری به‌عنوان مکملی برای رایانش ابری، به کاهش تأخیر و مدیریت مؤثر داده‌های IoT کمک می‌کند [۱۴] و [۱۵]. محاسبات مه را می‌توان مرحله‌ای از تکامل رایانش ابری دانست که منابع را از مرکز به لبه گسترش می‌دهد و قابلیت‌هایی مشابه رایانش ابری در سطح محلی فراهم می‌آورد [۱۶] و [۱۷]. گره‌های مه، مانند ماشین‌های مجازی (VM)، امکان پردازش و ذخیره‌سازی را در نزدیکی دستگاه‌های IoT فراهم می‌کنند و در حفظ امنیت، حریم خصوصی و بهره‌وری انرژی نقش دارند [۱۸] و [۱۹]. این معماری شامل اجزایی چون دروازه‌ها، سرورها، کنترل‌کننده‌ها و دوربین‌ها است و به دلیل تنوع نیازهای دنیای واقعی، سازماندهی برنامه‌های IoT را به چالشی جدی تبدیل کرده است [۲۰] و [۲۱]. از سوی دیگر، به دلیل محدودیت‌های منابع، ماهیت پویا، ناهمگنی منابع، تقاضای منابع و دیگر محدودیت‌های موجود نیاز به تخصیص منابع به شدت احساس می‌شود و زمان‌بندی کار در یک گره مه بسیار حائز اهمیت است زیرا زمان‌بندی وظایف به نحوی که نیازهای کیفیت خدمات (QoS) را برآورده کنند یک مسئله NP کامل/سخت می‌باشد [۲۲]. برخلاف مدل محاسبات ابری، ظرفیت پردازش منابع در محاسبات مه اغلب محدود است. با وجود پیشرفت‌های چشمگیر در کاهش تأخیر و افزایش بهره‌وری سیستم‌های رایانش مه، مدیریت بهینه منابع همچنان یکی از چالش‌های اصلی در این حوزه باقی مانده است. نتایج مطالعات مختلف نشان می‌دهد که استفاده از رویکردهای نوین می‌تواند عملکرد این سامانه‌ها را در زمینه‌هایی چون مصرف انرژی، هزینه اجرا، بهره‌برداری از شبکه و کاهش تأخیر به‌طور قابل توجهی بهبود بخشد [۲۳] تا [۲۵]. در نتیجه ارایه یک روش مؤثر برای تخصیص منابع و زمان‌بندی وظایف در این شرایط لازم است [۲۶] و [۲۷]. انتخاب یک روش زمان‌بندی کارآمد به یکی از مشکلات محققان تبدیل شده است. مطالعه و ارزیابی جامع روش‌های زمان‌بندی وظایف در

چکیده: با گسترش روزافزون اینترنت اشیا (IoT)، حجم بالای داده‌های تولیدی توسط دستگاه‌ها منجر به افزایش تأخیر و مصرف پهنای باند در محیط‌های ابری شده است. محاسبات مه به‌عنوان راه‌حلی برای کاهش این چالش‌ها مطرح شده است، اما تنوع منابع و وظایف در این محیط، زمان‌بندی کارآمد وظایف را با پیچیدگی مواجه کرده است. در این پژوهش، یک الگوریتم زمان‌بندی ترکیبی نوآورانه با بهره‌گیری از الگوریتم چتر دریایی و الگوریتم ژنتیک ارائه شده است که با هدف کاهش مصرف انرژی و زمان اجرا، وظایف را بر اساس اولویت و ویژگی‌های آن‌ها بهینه‌سازی می‌کند. الگوریتم چتر دریایی به دلیل ویژگی‌هایی چون ساده بودن ساختار، سرعت همگرایی بالا، و قابلیت انطباق پویا با تغییرات در فضای جستجو، گزینه‌ای مناسب برای مسائل زمان‌بندی پویا در محیط‌های مه-ابر است. در روش پیشنهاد شده، این الگوریتم با یک الگوریتم ژنتیک ترکیب شده تا هم تنوع جمعیت راه‌حل‌ها حفظ شود و هم از همگرایی زودهنگام جلوگیری شود. به‌واسطه این ترکیب، مدل پیشنهادی می‌تواند فضای جستجو را به‌صورت گسترده و متوازن پوشش دهد و به‌جای گیر افتادن در کمینه‌های محلی، به سمت جواب‌های نزدیک به بهینه جهانی حرکت کند. نوآوری اصلی این روش در ترکیب سازگار دو الگوریتم الهام‌گرفته از طبیعت برای بهینه‌سازی چندهدفه و ایجاد توازن میان بهره‌وری انرژی و کارایی پردازش در محیط‌های مه-ابر است. نتایج شبیه‌سازی و مقایسه با الگوریتم‌های پیشرفته‌ای مانند JIFA، GA-PSO، MFO-PSO-GA و PWAO نشان می‌دهد که روش پیشنهادی عملکرد بهتری در کاهش زمان اجرای کل، مصرف انرژی و افزایش شاخص PDST دارد. این یافته‌ها نشان‌دهنده اثربخشی و برتری مدل پیشنهادی در بهبود کیفیت خدمات در سیستم‌های محاسباتی توزیع شده هستند.

کلیدواژه: اینترنت اشیا، الگوریتم تخصیص منابع، الگوریتم چتر دریایی، الگوریتم ژنتیک.

۱- مقدمه

با گسترش روزافزون فناوری‌های هوشمند، به‌ویژه اینترنت اشیا (IoT)، حجم قابل توجهی از داده‌ها توسط دستگاه‌های مختلف تولید می‌شود. این حجم بالا از داده‌ها می‌تواند منجر به مصرف زیاد پهنای باند

این مقاله در تاریخ ۳۰ آذرماه ۱۴۰۳ دریافت و در تاریخ ۱۱ تیر ماه ۱۴۰۴ بازنگری شد.

سکینه سهرابی، گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بوعلی سینا، همدان، ایران، (email: s.sohrabi@eng.basu.ac.ir).

مهدی سخایی‌نیا (نویسنده مسئول)، گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بوعلی سینا، همدان، ایران، (email: sakhaei@basu.ac.ir).

محمد نصیری، گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بوعلی سینا، همدان، ایران، (email: m.nassiri@basu.ac.ir).

رضا محمدی، گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بوعلی سینا، همدان، ایران، (email: r.mohammadi@basu.ac.ir).

الگوریتم بهینه‌سازی کرم شب تاب پیشنهاد کرده‌اند. حجم کار برای این رویکرد توسط الگوریتم‌های بلادرنگ HPC2N و NASA در نظر گرفته شده است و از مجموعه داده‌های ساخته شده با توزیع‌های مختلف استفاده می‌شود. برای ارزیابی رویکرد پیشنهادی، از شبیه‌ساز Cloudsim استفاده شد. استفاده از بهینه‌سازی کرم شب‌تاب به این دلیل بوده است که الگوریتم هدف ذکر شده می‌تواند فضای مساله را با کمک کرم شب‌تاب با دقت طی کند و به تعداد تکرار کمتری نیاز دارد. بنابراین، حل مسئله NP-hard مورد توجه محققین است. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی تأثیر قابل‌توجهی بر معیارهای مدنظر در مقاله مذکور دارد.

مقاله [۳۱] به معرفی روشی جهت اختصاص کارآمد وظایف در محیط محاسباتی مه پرداخته است که تمرکز اصلی بر روی برنامه‌های مبتنی بر IoT بوده و سعی در بهبود تعادل وظایف بارگذاری شده بر روی گره‌های مه کرده است. در مقاله مذکور از توانایی‌های دو الگوریتم فراابتکاری بهینه‌سازی کلونی مورچگان و بهینه‌سازی ازدحام ذرات، بهره گرفته شده است. بررسی نتایج حاصل از ارزیابی‌ها بیانگر آن است که استفاده از روش زمان‌بندی پیشنهادی منجر به بهبودی در زمان پاسخ برنامه‌های IoT در مقایسه با الگوریتم مبتنی بر بهینه‌سازی ازدحام ذرات و الگوریتم زمان‌بندی نوبت گردشی می‌شود. به صورت کلی می‌توان گفت که این روش به طور مؤثر باعث تعادل بارکاری بر روی گره‌های مه می‌شود. زمان‌بندی مقرون‌به‌صرفه آگاه از منابع (RACE) توسط آرشد و همکاران [۳۲] پیشنهاد شده است که استفاده از منابع را در لایه مه افزایش می‌دهد و هزینه را در فضای ابری با کمترین میزان استفاده از پهنای باند کاهش می‌دهد. روش پیشنهادی این کار را به کمک ماژول‌های کاربردی تخصیص وظایف به گره‌های مه انجام می‌دهد. در روش پیشنهادی RACE از دو الگوریتم استفاده شده است. ماژول زمان‌بندی در RACE بر اساس پهنای باند و نیازهای محاسباتی وظایف، آن‌ها را طبقه‌بندی می‌کند.

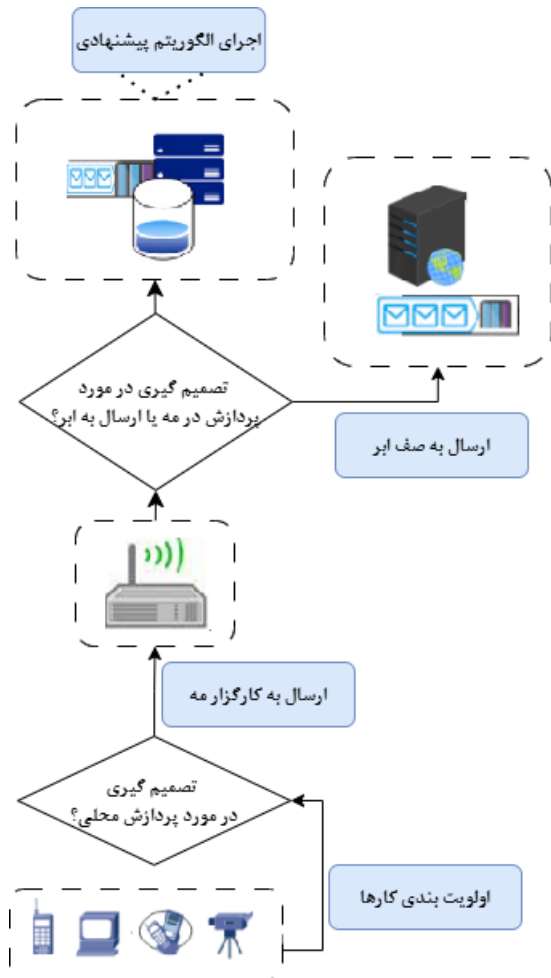
در مقاله [۳۳]، الگوریتم MBFD ارائه شده است که به منظور حل چالش‌های پیشروی محققان ارائه شده است. الگوریتم ارایه شده در این مقاله یک روش بهینه مبتنی بر الگوریتم کلونی زنبورهای مصنوعی (ABC) و تکنیک بهینه‌سازی با شبکه عصبی مصنوعی (ANN) است که یک تکنیک مؤثر جهت زمان‌بندی کار در محاسبات مه می‌باشد. به منظور ارزیابی اثربخشی الگوریتم مذکور، نتایج آزمایش‌های انجام شده است مورد بررسی قرار گرفته است که بیانگر بهبود معیارهای مصرف انرژی و زمان اتمام کار بهتر در مقایسه با سایر روش‌های مورد بررسی می‌باشد. باید توجه داشت که در روش مذکور به معیارهایی همچون اولویت و تعادل بار پرداخته نشده است. پیروزمند و همکاران [۳۴] یک مکانیسم زمان‌بندی کار طراحی کرده‌اند که یک رویکرد ترکیبی است و در آن از روش GSA برای جستجوی محلی و از الگوریتم GA به منظور جستجوی سراسری استفاده می‌شود. رویکرد پیشنهادی برای دستیابی به عملکرد بالا ارائه شده است. روش پیشنهادی GSAGA در Cloudsim پیاده‌سازی شده است و حجم کاری ثبت‌شده در این رویکرد به یک مجموعه داده کار بلادرنگ با ماهیت ناهمگن اشاره دارد. روش پیشنهادی با الگوریتم‌های IWC و Profit-MTC مقایسه شده است و بررسی نتایج نشان می‌دهد که GSAGA تأثیر زیادی بر روی کاهش مصرف انرژی داشته و ظرفیت پردازش را به حداکثر می‌رساند. آرشد و همکاران [۳۵] یک روش زمان‌بندی مؤثر در یک چارچوب مه-ابر پیشنهاد کرده‌اند. رویکرد پیشنهادی یک تکنیک زمان‌بندی مبتنی بر الگوریتم ژنتیک است.

محیط مه برای دستیابی به هدف فوق ضروری است. محققان مختلف روش‌های مختلف زمان‌بندی وظایف در محیط مه را با توجه به ویژگی‌های آنها بررسی می‌کنند. استفاده بهینه از منابع مه-ابر برای افزایش پارامترهای کیفی مختلف مانند هزینه عملیات، مصرف انرژی و زمان اجرای کار یک موضوع مهم است. زمان‌بندی کار مناسب در محیط مه، زمان اجرای کار، مصرف انرژی و درصد وظایفی که مهلت اجرای آن‌ها رعایت شده است (PDST) را کاهش می‌دهد. در این مقاله تکنیک زمان‌بندی کار و تخصیص منابع بر اساس الگوریتم اکتشافی و فراابتکاری پیشنهاد شده است. در این رویکرد، از مزایای چتر دریایی و الگوریتم ژنتیک برای اجرای زمان‌بندی کارآمد استفاده می‌شود. در این مطالعه، ما اهمیت وظایف را بر اساس مصرف انرژی، PDST و زمان اجرای کل در هنگام برنامه‌ریزی آنها در محاسبات مه-ابر در نظر می‌گیریم. هدف اصلی این مقاله ارائه روشی مؤثر برای زمان‌بندی کار و تخصیص منابع در مه است. به گونه‌ای که محدودیت‌های در نظر گرفته شده در این مقاله را برآورده سازد. ادامه مقاله به شرح زیر سازماندهی شده است. در بخش ۲، مطالعه بر روی تکنیک‌های زمان‌بندی کار موجود پوشش داده شده است. مدل سیستم، فرمول بندی مسئله و تابع هدف در بخش ۳ به تفصیل آمده است. مفروضات و محدودیت‌ها در بخش ۴ بیان شده است. در بخش ۵ الگوریتم پیشنهادی که یک روش پیشرفته از الگوریتم چتر دریایی و الگوریتم ژنتیک می‌باشد (EMJFGA) توضیح داده شده است. تجزیه و تحلیل عددی و اثربخشی تکنیک EMJFGA در بخش ۶ نشان داده شده است. نتیجه‌گیری و تمرکز آینده کار در بخش ۷ ارائه شده است.

۲- پیشینه تحقیق

در سال‌های اخیر تحقیقات گسترده‌ای در زمینه زمان‌بندی وظایف IoT در محیط مه و لبه انجام شده است که در ادامه به بررسی برخی از روش‌های مرتبط در هر دسته می‌پردازیم.

در مقاله [۲۸] الگوریتمی جهت تخصیص منابع به نام PSG پیاده‌سازی شده است. در روش مذکور از الگوریتم‌های MIN-VIOL و SEMI-GREEDY بهره گرفته شده است که تصمیم‌گیری براساس زمان پاسخ انجام می‌شود. هدف از این مقاله کاهش انرژی سیستم مه با توجه به مهلت انجام وظایف می‌باشد. الگوریتم SEMI-GREEDY هنگامی که وظیفه مورد نظر زمان کافی برای پردازش را داشته باشد اجرا می‌شود و هنگامی که زمان انجام کار از مهلت تعیین شده بیشتر می‌شود، سعی در کاهش زمان تخطی شده و الگوریتم MIN-VIOL اجرا می‌شود. برای متعادل کردن فضای جستجوی جهانی و محلی با تغییر وزن اینرسی ذرات از رویکرد مذکور استفاده می‌شود. عمده‌تأ با پارامترهای توان عملیاتی، متوسط استفاده از منابع و طول عمر سروکار دارد. یکی از پارامترها و جنبه‌های مهم در محیط محاسبات ابری را می‌توان مصرف انرژی نامید. غفاری و منصوری [۲۹] با استفاده از الگوریتم بهینه‌سازی گرگ خاکستری، یک تکنیک زمان‌بندی کار را برای سیستم ابری توسعه داده‌اند. در این تکنیک، حجم کاری از منابع موجود ترسیم می‌شود. در این مقاله کاهش هزینه‌های اجرا، مصرف انرژی و زمان راه‌اندازی را می‌توان به عنوان یکی از اهداف اصلی در نظر گرفت. وظایف ورودی توسط کنترل‌کننده VM دریافت می‌شوند. سپس وضعیت منابع توسط تیم مدیریت VM ارزیابی می‌شود. در صورت امکان، وظایف وارد شده به سیستم به ماشین‌های مجازی موجود واگذار می‌شود. در غیر این صورت، ماشین‌های مجازی جدید برای پردازش وظایف ایجاد می‌شوند. مگالامپاتی و همکاران [۳۰]، یک رویکرد زمان‌بندی کار را بر اساس



شکل ۱: معماری سیستم در روش پیشنهادی.

و اجرای آن‌ها در این لایه تکمیل می‌شود. این نشان می‌دهد که کارگزار^۲ در سناریوی مشخص شده قادر به ارسال مستقیم وظایف به لایه ابری نخواهد بود. شکل ۱ رویکرد پیشنهادی از سه مرحله موثر به شرح زیر تشکیل شده است:

۱. برآورد تاخیر و تعیین اولویت و تشکیل صف برای انجام وظایف،
۲. زمان‌بندی با کمک الگوریتم بهینه‌سازی ترکیبی با استفاده از الگوریتم چتر دریایی و الگوریتم بازبخت شبیه‌سازی شده،
۳. بهبود اولویت کارها با کمک مهلت اجرای آنها (پس از هر بار اجرای الگوریتم پیشنهادی).

۲-۳ مبانی نظری

این بخش مدلی را که برای مسئله زمان‌بندی کار در سیستم محاسباتی ابر-مه در نظر گرفته شده است، توضیح می‌دهد. الگوریتم پیشنهادی مدیریت زمان‌بندی وظایف در محیط مه-ابر را بر عهده دارد. تاکید اصلی رویکرد پیشنهادی بر استفاده از الگوریتم‌های فراابتکاری پیشرفته چتر دریایی و الگوریتم ژنتیک شده است. هدف این مقاله ارتقای کیفیت سرویس با بهبود شاخص‌های عملکرد شبکه با استفاده از روش توصیف شده، می‌باشد. برای کمک به درک مدل سیستم، در ابتدا متغیرهای تصمیم و مؤلفه‌های اساسی مسئله بهینه‌سازی معرفی می‌شوند. بخش‌های بعدی به توضیح مدل‌های تحت بررسی برای تعریف تابع هدف بر اساس معیارهای مربوطه می‌پردازند. در نهایت، محدودیت‌های شناسایی شده به

جدول ۱: ویژگی‌های مربوط به الگوریتم‌های بررسی‌شده.

مرجع	مزایا	روش مورد استفاده	سال
[۲۸]	کاهش مصرف انرژی و به حداقل رساندن زمان نقض مهلت انجام کار	SEMI-GREEDY (PSG Algorithm)	۲۰۲۲
[۲۹]	زمان پاسخ، هزینه و مصرف انرژی	GWOTS	۲۰۲۲
[۳۰]	معیارهای زمان اتمام کار	TAFFA	۲۰۲۲
[۳۱]	زمان پاسخ و تعادل بار	ACO and PSO Algorithm	۲۰۲۳
[۳۲]	زمان پاسخ و پهنای باند شبکه	RACE	۲۰۲۰
[۳۳]	معیارهای زمان اتمام کار و مصرف انرژی	ABC and ANN Algorithm	۲۰۲۲۱
[۳۴]	ظرفیت پردازش و مصرف انرژی	GSAGA	۲۰۲۱
[۳۵]	معیارهای زمان اتمام کار	GA-IRACE	۲۰۲۲
[۳۶]	معیارهای زمان اتمام کار و بارکاری شبکه	ACO with the help of blockchain	۲۰۲۱

در ابتدا، جمعیت اولیه (کروموزوم‌ها) ایجاد شده، سپس به منظور تکامل راه‌حل‌ها، انواع تغییرات پارامترها توسط عملگرهای جهش و ترکیب بر اساس الگوریتم ژنتیک مورد استفاده قرار می‌گیرد. از نظر هزینه، زمان اجرا و پهنای باند تکنیک پیشنهادی در مقایسه با تکنیک‌های موجود ارزیابی شده است که بیانگر بهبود معیارهای مذکور به کمک تکنیک پیشنهادی می‌باشد. بانیاتا و همکاران [۳۶] یک الگوریتم زمان‌بندی کار مبتنی بر الگوریتم بهینه‌سازی کلونی مورچه‌ها (ACO) با کمک زنجیره بلوکی ارائه کرده‌اند. از ماینرهای بلاک‌چین برای ایجاد زمان‌بندی وظایف کارآمد در روش پیشنهادی استفاده شده است. همچنین برای تولید بهترین زمان‌بندی از اعطای پاداش به گره‌های ماینر استفاده می‌کند. بررسی نتایج حاصل از شبیه‌سازی نشان می‌دهد که الگوریتم پیشنهادی منجر به بهبود بار شبکه و مدت زمان اجرای کل می‌شود. لازم به ذکر است که در الگوریتم پیشنهادی، فرض بر این است که تمامی کارها زمان رسیدن یکسانی داشته‌اند.

در جدول ۱ تکنیک مورد استفاده، ابزار شبیه‌سازی، معیارهای کارایی، دیتاست و ویژگی‌های مربوط به الگوریتم‌های بررسی‌شده در این پژوهش به اختصار آورده شده است.

۳- مدل سیستم

این بخش به بررسی مدل سیستم در این تحقیق می‌پردازد و مدلی که پیشنهاد می‌شود ترکیبی از مدل‌هایی است که در مقالات [۳۷] و [۳۸] ارائه شده است. در این تحقیق سه لایه معماری سیستم مورد بررسی قرار گرفته است: لایه ابر، لایه مه و لایه IoT. لازم به تاکید است که در این مدل لایه محاسبات لبه^۱ و لایه IoT یکسان در نظر گرفته شده است. این بخش شامل دو بخش فرعی است که با هم مدل سیستم را تعریف می‌کنند. در قسمت اول زیر، معماری کلی IoT-مه-ابر ارائه شده است. در بخش دوم، مسئله زمان‌بندی کار به صورت ریاضی فرموله شده است.

۱-۳ معماری

این مقاله به بررسی طرح معماری نمایش داده شده در شکل ۱ می‌پردازد. در این معماری، وظایف ابتدا برای پردازش قبل از هدایت به لایه ابری به مه فرستاده می‌شوند. در سناریوی در نظر گرفته شده در این مقاله فرض شده است که وظایف برای اجرا به لایه مه ارسال می‌شوند

چهار مشخصه اصلی برای هر گره f_i در نظر گرفته شده است که شامل تاخیر ارتباطی از کارگزار (f_j^{del}) ، حالت بیکار (f_j^{idl}) ، میزان مصرف انرژی در حالت فعال (f_j^{act}) و نرخ پردازش پردازنده (f_j^{cpu}) می‌شود. همچنین فرض بر این است که هر گره ابری نیز دارای چهار ویژگی فوق می‌باشد. برای طبقه‌بندی وظایف بر اساس اولویت و مهلت، چهار کلاس اصلی در نظر گرفته شده است که شامل کلاس‌های به شدت شدید، شدید، متوسط و کم می‌شود. سپس وظایف طبقه‌بندی شده

برای برآورده کردن QoS با کمک مدل پیشنهادی زمان‌بندی می‌شوند. در مدل پیشنهادی در پایین‌ترین لایه که شامل دستگاه‌های اینترنت اشیا است، تعداد سنسورهایی که توسط لایه IoT نشان داده می‌شود. محیط مه شامل چندین سرور مه f_j و یک کارگزار مه f_b است که همگی به سرور مه متصل هستند. در واقع در معماری شبکه، لایه محاسبات مه به عنوان لایه دوم در نظر گرفته می‌شود و کارگزار مه عنصر اصلی یک ارائه دهنده خدمات مه است که وظیفه تعیین اولویت وظایف و نظارت بر سرورهای مه را بر عهده دارد. در مدل پیشنهادی، گره ابری با حرف C_k نشان داده شده است که نشان‌دهنده لایه محاسبات ابری است و بالاترین لایه در سیستم است. لازم به ذکر است که در این تحقیق $X_{m \times n}$ به عنوان ماتریس انتساب وظیفه در نظر گرفته شده و هر درایه این ماتریس توسط (۴) نشان داده شده است.

$$x_{ik} = \begin{cases} 1, & \text{if task } t \text{ assigned to } F_i \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

در مدل پیشنهادی وظایف مستقل از هم در نظر گرفته شده است و برای اجرا به گروهی از گره‌های مه اختصاص داده می‌شود. در این مقاله، تاخیر انتشار نیز نادیده گرفته شده است. همچنین فرض می‌شود که مقدار پارامترهای مورد استفاده برای کارگزار مه (f_b) قابل دسترسی است. لازم به ذکر است که همانطور که قبلاً ذکر شد برای بررسی دقیق تر الگوریتم تخصیص در لایه محاسبه مه، این تخصیص توسط عامل مه f_b بر اساس الگوریتم تخصیص در نظر گرفته شده، انجام می‌شود. همچنین به دلیل محدودیت منابع در لایه مه ممکن است وظایف خاصی برای اجرا به لایه ابر ارسال شود. هر (D_i) IoT وظایفی را با نرخ λ_{D_i} تولید و ارسال می‌کند. نرخ ورود بار در سرور مه با کل نرخ‌های IoT برابر است. در این تحقیق IoT که درخواست را به سیستم ارسال می‌کنند انواع مختلفی دارند.

پس از دریافت درخواست، در صورتی که IoT (فرستنده درخواست) قدرت پردازش داشته باشد، بخشی از درخواست (در این تحقیق درصدی از حجم داده ارسالی در نظر گرفته شده است) به صورت محلی بر روی دستگاه ارسال کننده پردازش می‌شود. در غیر این صورت، کار برای پردازش توسط گره مه به کارگزار مه ارسال می‌شود. اگر مه نتواند کار را پردازش کند، کار به سرور ابری ارسال می‌شود و سرور ابری بر اساس الگوریتم نوبت چرخشی^۱ تصمیم می‌گیرد که کار را به کدام گره ابری اختصاص دهد. در این تحقیق فرض بر این است که پردازش کار در لایه مه انجام می‌شود و از ارسال کار به لایه ابر صرف نظر می‌شود و این قابلیت در مطالعه بعدی اضافه خواهد شد. در نهایت در صورت انجام حجم کار، پاسخ به IoT ارسال می‌شود و میزان انرژی و تاخیر در ارسال نتیجه نیز باید محاسبه شود. در ادامه، نحوه محاسبه انرژی، زمان پاسخ و تاخیر در مدل پیشنهادی را مورد بحث قرار خواهیم داد.

دلیل ظرفیت‌های محدود گره‌ها و پیوندهای ارتباطی در سیستم نشان‌دهنده شده، روشن می‌شوند. باید توجه داشت که در ابتدا، تمام درخواست‌های کاربر به همراه مقادیر پارامتر مربوطه به یک ارسال می‌شوند. این کارگزار بر تخصیص منابع با توجه به خواسته‌های کاربر نظارت می‌کند و به عنوان یک مرجع مرکزی برای تصمیم‌گیرانجام وظیفه می‌کند.

۳-۲-۱ متغیرهای تصمیم

برای بررسی شبکه، فرض می‌شود که یک شبکه مه-ابر از تعداد محدودی دستگاه مه (m) تشکیل شده است که عبارتند از: $F = \{f_1, f_2, \dots, f_m\} \forall f \in F$ و همچنین در این شبکه تعداد N سرور ابری موجود است که به شکل $C = \{c_1, c_2, \dots, c_m\} \forall c \in C$ نمایش داده شده است. نماد A بیانگر مجموعه‌ای از IoTها است، $A = \{1, 2, \dots, d\} \forall d \in A$. در این شبکه G بیانگر تعداد کارگزارهای موجود در سیستم می‌باشد که امکان تصمیم‌گیری برای پردازش وظایف به صورت محلی یا در یک دستگاه مه یا سرور ابری را دارند که به شکل $G = \{1, 2, \dots, g\} \forall g \in G$ نمایش داده شده است. هر IoT ظرفیت تولید k وظایف را دارد که $K = \{1, 2, \dots, t\} \forall t \in T$. اندازه داده‌های ورودی و خروجی برای هر وظیفه تولید شده توسط دستگاه‌ها می‌تواند به ترتیب K_k^{in} و K_k^{out} بیت باشد. این وظایف بر اساس الزامات QoS اجرا می‌شوند. مانند بسیاری از روش‌های زمان‌بندی، به هر وظیفه یک گره مه یا ابر مناسب اختصاص داده می‌شود. لازم به ذکر است که t_i نشان‌دهنده i مین وظیفه در دنباله وظایف است و TK بیانگر تعداد وظایف ارسال شده توسط دستگاه‌های اینترنت اشیا است. هر وظیفه t_i دارای ویژگی‌هایی است که می‌توان آنها را به صورت زیر فرموله کرد:

$$t_i = \langle t_i^{ID}, t_i^{TL}, t_i^p, t_i^{dl} \rangle \quad (1)$$

که در آن t_i^p سطح اولویت، t_i^{TL} طول کار (واحد: میلیون دستورالعمل)، t_i^{dl} مهلت و t_i^{ID} شناسه وظیفه را نشان می‌دهد. وظایفی که از دستگاه‌های اینترنت اشیا سرچشمه می‌گیرند توسط کارگزار جمع‌آوری می‌شوند که مسئول طبقه‌بندی و زمان بندی وظایف است. در این سناریو ابتدا کارها بر اساس مهلت مقرر مرتب می‌شوند سپس به ترتیب زیر به هر کار اولویت داده می‌شود. مقادیر آستانه برای اولویت یک کار را می‌توان به صورت (۲) محاسبه کرد که در آن $H(T_i^{dl})$ و $H(T_i^p)$ به ترتیب آستانه‌های مربوط به مهلت و اولویت اجرای کار هستند. علاوه بر این، t_i^{dl} مقدار مهلت اجرا و t_i^p مقدار اولویت کار t_i با مقدار آستانه از ۰ تا ۱ است.

$$H(T_i^{dl}) = \sum_{i=1}^n T_i^{dl} \text{Log} T_i^{dl} \quad (2)$$

$$H(T_i^p) = \sum_{i=1}^n T_i^p \text{Log} T_i^p$$

اولویت و مهلت به دو کلاس کم و زیاد طبقه‌بندی می‌شوند و می‌توانند فرمول‌بندی شوند. همان‌طور که در (۳) آورده شده است.

$$H(T_i^{dl}) = \begin{cases} low, & \text{if } 0 \leq H(T_i^{dl}) \leq 0.5 \\ high, & \text{if } 0.5 < H(T_i^{dl}) \leq 1 \end{cases} \quad (3)$$

$$H(T_i^p) = \begin{cases} low, & \text{if } 0 \leq H(T_i^p) \leq 0.5 \\ high, & \text{if } 0.5 < H(T_i^p) \leq 1 \end{cases}$$

۳-۲-۲ پردازش محلی

مجازی موجود در لایه مه اشاره دارد که به صورت (۱۰) نشان داده شده است.

$$ECT_{ir} = \frac{TL(t_i)}{N_r^{mips}}, i = 1, 2, 3, \dots, k, r = 1, 2, 3, \dots, m \quad (10)$$

همچنین E_m^{cpu} نشان دهنده ظرفیت محاسباتی دستگاه مه است. با کمک (۱۱)، می توان نرخ کلی مصرف انرژی در دستگاه مه را محاسبه کرد

$$E_{am}^{Process} = ECT_{ir} \times E_m^{cpu} \quad (11)$$

به طور مشابه، می توان برای محاسبه نرخ انتقال پاسخ به IoT از (۱۲) استفاده کرد. در (۱۲) متغیر B_{ma}^{down} نشان دهنده پهنای باند انتقال موجود است.

$$R_{ma}^{down} = B_{ma}^{down} \log_2(1 + [\rho_m^{down} H_m / \xi_a^\gamma]) \quad (12)$$

متغیر T_{am}^{down} بیانگر مدت زمان دانلود و متغیر E_{ma}^{down} بیانگر میزان مصرف انرژی در دستگاه مه هستند که می توان به صورت (۱۳) بیان کرد

$$T_{am}^{down} = \frac{x(m, k) \times \phi K_k^{out}}{R_{ma}^{down}} \quad (13)$$

$$E_{ma}^{down} = T_{ma}^{down} \times \rho_m^{down}$$

بنابراین، میزان کل مدت زمان تأخیر و میزان انرژی مصرف شده توسط کار $(k \in K)$ در هنگام پردازش در دستگاه مه به کمک (۱۴) محاسبه می شود

$$T_{am}^{Total} = T_{am}^{up} + T_{am}^{Process} + T_{am}^{down} \quad (14)$$

$$E_{ma}^{Total} = E_{ma}^{up} + E_{ma}^{Process} + E_{ma}^{down}$$

بنابراین، با توجه به روابط داده شده در بالا، میزان مدت زمان تأخیر و مصرف انرژی با استفاده از (۱۵) محاسبه می شود. توجه داشته باشید که:

- تمام پردازش یک کار را می توان در دستگاه های IoT انجام داد.
- اگر دستگاه IoT باتری کافی نداشته باشد، وظیفه برای اجرا به گره مه فرستاده می شود و پردازش توسط گره مه انجام می گردد.
- بخشی از کار توسط IoT پردازش شده و سپس به گره مه ارسال می شود و پردازش کار در گره مه کامل می شود.

لازم به ذکر است که مدت زمانی که یک کار در صف پردازش دستگاه مه قرار می گیرد در نظر گرفته می شود. توجه داشته باشید که متغیر isProcessing یک متغیر باینری است که تعیین می کند آیا IoT وظایف را به صورت محلی اجرا می کنند یا خیر، اگر مقدار این متغیر درست باشد، بخشی از کار به صورت محلی پردازش شده است.

در ابتدا، دستگاه های IoT بررسی می کنند که آیا فرکانس پردازنده در دسترس است یا خیر. به عبارت دیگر، اگر فرکانس پردازنده یک (Γ_k^{CPU}) بیشتر از نیاز وظیفه وارد شده به سیستم باشد (K_k^{CPU}) ، یعنی $\Gamma_k^{CPU} > K_k^{CPU}$ سپس دستگاه وظایف را به صورت محلی انجام می دهد. برای کار K ام، اجازه دهید ϕ نشان دهنده چگالی پردازش باشد. در نتیجه، $T_{ka}^{Process}$ نشان دهنده زمان پردازش کار است. معادله زیر فرمول محاسبه مدت زمان پردازش کار به صورت محلی و در (۵) آورده شده است

$$T_{ka}^{Process} = \frac{x(k, a) \times \phi K_k^{in}}{\Gamma_a^{CPU}} \quad (5)$$

همچنین، مصرف انرژی $E_{ka}^{Process}$ فرایند پردازش یک کار در IoT به صورت (۶) محاسبه می شود که در آن میزان مصرف انرژی Z امین گره مه در حالت فعال با $\rho_{ka}^{Process}$ نشان داده می شود، در حالی که مصرف انرژی در حالت بیکار با ρ_a^{idl} نشان داده می شود.

$$E_{ka}^{Process} = T_{ka}^{Process} \times \rho_{ka}^{Process} + T_{ka}^{idl} \times \rho_a^{idl} \quad (6)$$

زمان صرف شده در حالت های فعال و بیکار با $T_{ka}^{Process}$ و T_{ka}^{idl} ارائه می شود که به ترتیب با (۵) و (۷) تعیین می شوند.

$$T_{ka}^{idl} = TotalTime - T_{ka}^{Process} \quad (7)$$

۳-۲-۳ پردازش در لایه مه

IoT می تواند بخش کوچکی از وظایف را به صورت محلی پردازش کنند. با این حال، به دلیل فرکانس محدود پردازنده در دستگاه های IoT، وظایف به دستگاه های محاسباتی مناسبی که باید حداقل تأخیر و نیازهای منابع موجود را برآورده کنند، هدایت می شوند. ρ_a^{up} و H_a به ترتیب بیانگر توان انتقال و توان کانال در دستگاه IoT می باشد. طبق فرمول ظرفیت شانون [۲۳]، R_{am}^{up} نرخ انتقال داده های بارگذاری را نشان می دهد و استفاده از (۸) محاسبه می شود.

$$R_{am}^{up} = B_{am}^{up} \log_2(1 + [\rho_a^{up} H_a / \xi_m^\gamma]) \quad (8)$$

با توجه به (۸) متغیر B_{am}^{up} نشان دهنده پهنای باند انتقال تخصیص یافته بین دستگاه IoT و M امین گره مه است. بنابراین، میزان مصرف انرژی برای انتقال (E_{am}^{up}) و مدت زمان انتقال داده (T_{am}^{up}) تا یک دستگاه مه را می توان به صورت (۹) نشان داد.

$$T_{am}^{up} = \frac{x(k, m) \times \phi K_k^{in}}{R_{am}^{up}}, E_{am}^{Process} = T_{am}^{up} \times \rho_a^{up} \quad (9)$$

ECT_{ir} به مدت زمان لازم برای اجرای کار i ام در Z امین ماشین

$$T_{kl}^{Total} = \begin{cases} T_{kd}^{Process} + T_{kd}^{idl}, & \text{if } l \in A \\ T_{dm}^{up} + T_m^{queue} + T_{dm}^{Process} + T_{dm}^{down} + T_{kd}^{idl}, & \text{if } l \in m \text{ and is Processing} == \text{false} \\ T_{kd}^{Process} + T_{dm}^{up} + T_m^{queue} + T_{dm}^{Process} + T_{dm}^{down} + T_{kd}^{idl}, & \text{if } l \in m \text{ and is Processing} == \text{true} \end{cases} \quad (15)$$

$$E_{kl}^{Total} = \begin{cases} E_{kd}^{Process}, & \text{if } l \in A \\ E_{dm}^{up} + E_{dm}^{Process} + E_{dm}^{down}, & \text{if } l \in m \text{ and is Processing} == \text{false} \\ E_{kd}^{Process} + E_{dm}^{up} + E_{dm}^{Process} + E_{dm}^{down}, & \text{if } l \in m \text{ and is Processing} == \text{true} \end{cases}$$

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

شکل ۲: مدل سازی موقعیت چتر دریایی در روش پیشنهادی.

این پارامتر در الگوریتم GA استفاده کرد. مهمترین مزیت این الگوریتم اجتناب از بهینه محلی است.

هر ذره تولیدشده در جمعیت اولیه به عنوان راه‌حلی برای مسأله در نظر گرفته می‌شود. هر ذره دارای دو بخش است: قسمت متغیر نشان‌دهنده انتساب وظیفه برای گره مه است و مقدار تابع هدف نشان‌دهنده کارایی سیستم برای راه‌حل ارائه شده توسط متغیرها است. چهار مرحله‌ای که الگوریتم پیشنهادی را تشکیل می‌دهند در زیر به اختصار توضیح داده شده است. لازم به ذکر است که علاوه بر عوامل ذکر شده قبلی، نیازهای سخت‌افزاری درخواست‌ها نیز در زمان‌بندی و تعیین وظایف با استفاده از تکنیک پیشنهادی در نظر گرفته شده است. الگوریتم پیشنهادی از چهار گام اصلی تشکیل شده است که در ادامه به توضیح هر یک می‌پردازیم.

۱-۵ رمزنگاری و تولید جمعیت اولیه

اولین مرحله در الگوریتم پیشنهادی، تولید جمعیت اولیه پس از تنظیم پارامترهای لازم است. در الگوریتم پیشنهادی استفاده از نظریه آشوب برای تولید جمعیت جایگزین‌رئوس تولید تصادفی شده است. در (۲۰) نگاشت لجستیک^۱ توضیح داده شده است.

$$x_{k+1} = \alpha x_k (1 - x_k) \quad (20)$$

طول آرایه در روش پیشنهادی برابر با تعداد وظایف است و هر چتر دریایی یک راه‌حل بالقوه برای مسأله مورد نظر را نشان می‌دهد. توجه داشته باشید که کارها باید به ترتیب قرارگیری در آرایه جهت پردازش ارسال شوند. با بیان متفاوت، این شاخص ترتیب اهمیت تکمیل کار و تعداد کل وظایف را نشان می‌دهد. مکان هر چتر دریایی را می‌توان به عنوان یک آرایه نشان داد، همانطور که در شکل ۲ مشاهده می‌شود. موقعیت یک چتر دریایی در مسأله نشان‌دهنده راه‌حل ممکن است و مقدار داخل هر درایه نشان‌دهنده شناسه گره مه است.

۲-۵ مکانیزم کنترل زمان

یک "مکانیسم کنترل زمان" وجود دارد که به چتر دریایی اجازه می‌دهد تا به طور متناوب بین حرکت در گروه‌ها یا دنبال کردن جریان های اقیانوسی حرکت کند. در این سیستم کنترل زمان، چتر دریایی در ابتدا به دلیل فراوانی مواد غذایی غنی از مواد مغذی در جریان اقیانوس به آنها نزدیک می‌شود. با گذشت زمان، چتر دریایی بیشتر به سمت آنها کشیده می‌شود و باعث ازدحام می‌شود. از سوی دیگر، چتر دریایی زمانی که دما یا باد جریان‌های اقیانوس را تغییر می‌دهد، به جریان دیگری در اقیانوس مهاجرت می‌کند و یک گروه چتر دریایی جدید ایجاد می‌کند. در یک دسته از چتر دریایی، دو نوع حرکت وجود دارد: نوع A که شامل حرکت غیرفعال است و نوع B که شامل حرکت فعال است. حرکت نوع A در ابتدا مورد علاقه است، اما با گذشت زمان به حرکت نوع B کشش پیدا می‌کند. در زیر توضیح مختصری در مورد هر یک از این حرکات ارائه شده است.

۴- مفروضات و محدودیت‌ها

۱. در مبحث تخصیص منابع، با توجه به محدود بودن ظرفیت گره‌ها و لینک‌های ارتباطی، محدودیت‌ها با استفاده از (۱۶) و (۱۷) تعریف می‌شوند.

این روابط بیان می‌کنند که اگر پیوند x_{ij} برای برقراری ارتباط بین درخواست t_i و گره مه f_j استفاده شود، مقدار $x_{ij} = 1$ است. همچنین با کمک متغیر باینری y_j^i می‌توان اطمینان حاصل کرد که درخواست t_i تنها توسط یک گره مه f_j پردازش می‌شود.

$$\sum_{i=1}^T \sum_{j=1}^m x_{ij} = 1 \quad (16)$$

$$\sum_{j \in F} y_j^i = 1 \quad (17)$$

۲. هر IoT نباید بیش از ظرفیت باتری خود انرژی مصرف کند زیرا دارای باتری محدود (d_k) است.

۳. با استفاده از (۱۸) می‌توان حدود توانایی پردازش مه را مشخص کرد.

$$\sum_{i=1}^n t_i^m \leq \sum_{j=1}^m f_j^{mips} \quad (18)$$

همچنین هر لینک l_{ij} دارای پهنای باند مشخصی است و برای کاهش ازدحام و در نتیجه کاهش تعداد ارسال مجدد، بهتر است اطلاعاتی بیشتر از ظرفیت لینک ارسال نشود. (۱۹) محدودیت مربوط به این مسأله در شبکه را بیان می‌کند.

$$\sum_{i=1}^n B_i \leq BW_f \quad (19)$$

۵- الگوریتم پیشنهادی

در این بخش به توضیح الگوریتم پیشنهادی پرداخته شده است. استفاده از الگوریتم فراابتکاری چتر دریایی هدف اولیه روش پیشنهادی تحقیق است. به منظور ارتقای وضعیت کلی شاخص‌های مورد بررسی، در این مقاله تلاش شده است تا با استفاده از روش ارائه شده، شاخص‌های عملکرد شبکه ارتقا یابد. در این بخش، الگوریتم بهینه‌سازی ترکیبی با استفاده از الگوریتم چتر دریایی و الگوریتم ژنتیک (GA) برای زمان‌بندی وظایف در محیط محاسبات مه تشریح می‌شود. الگوریتم پیشنهادی با استفاده از یک الگوریتم چند هدفه به معیارهای زمان اجرا، درصد وظایفی که مهلت اجرای آنها رعایت شده است و میزان مصرف انرژی پرداخته است. شبیه‌سازی حرکت چتر دریایی در جستجوی غذا شامل سه حرکت است که در زیر به طور خلاصه بیان می‌شود. از سوی دیگر، در روش پیشنهادی، علاوه بر الگوریتم چتر دریایی، از عملگرهای الگوریتم GA نیز در تولید راه‌حل‌های جدید استفاده می‌شود. در این زمینه، T نشان‌دهنده پارامتر دما است، که در ابتدا در دمای بسیار بالا تنظیم می‌شود تا امکان پذیرش راه‌حل‌های بدتر را فراهم کند. با کاهش تدریجی دما، در مراحل نهایی احتمال پذیرش راه‌حل‌های بدتر کمتر می‌شود و در نتیجه الگوریتم به سمت یک جواب خوب همگرا می‌شود. با توجه به ماهیت الگوریتم چتر دریایی و مکانیسم زمان، مکانیسم کنترل زمان را می‌توان به عنوان

S	۲	۱	۲	۳	۴	۷	۶	۷	۸	۹	۱۰	۱
S_{best1}	۶	۲	۱	۲	۶	۷	۷	۸	۹	۱۰	۱	۸
S_{best2}	۴	۷	۶	۷	۲	۱	۶	۷	۴	۷	۶	۷
S_{best3}	۷	۳	۴	۷	۶	۲	۶	۷	۸	۹	۱۰	۱
S_{best4}	۶	۷	۲	۱	۲	۱	۴	۷	۶	۷	۴	۲
S_{best4}	۱۰	۷	۷	۸	۹	۱۰	۱	۲	۵	۱	۲	۲
S_{best5}	۷	۳	۴	۷	۶	۷	۲	۱	۲	۲	۶	۷
S_{new}	۶	۷	۲	۷	۴	۷	۶	۷	۸	۹	۱۰	۲

شکل ۵: موقعیت ذره پس از حرکت غیرفعال.

کروموزوم (راه‌حل‌ها) عمل می‌کند. به این صورت که به تعداد $m/2$ ، در واقع به تعداد نصف خانه‌های آرایه (راه‌حل مورد بررسی) از بین درایه‌های آرایه انتخاب کرده و برای تعیین مقدار درایه، یکی از بهترین کروموزوم‌ها را انتخاب کرده و محتوای ژن مدنظر در کروموزوم جدید را برابر محتوای ژن متناظر در کروموزوم انتخابی قرار می‌دهد.

۵-۲-۴ تابع هدف و انتخاب راه‌حل

در مسائل چند هدفه در این گام با مشکل مواجه خواهیم شد زیرا در مسائل بهینه‌سازی چندهدفه، همواره چند تابع هدف مختلف وجود دارند که تمایل به یافتن بیشینه یا کمینه‌ی آن‌ها به‌طور هم‌زمان وجود دارد. باید توجه داشت که اکثر این توابع هدف در نقطه‌ی مقابل یک‌دیگر قرار دارند، به طوری که بهبود یکی از آن‌ها، منجر به بدتر شدن دیگری می‌شود. بنابراین در این گونه مسائل مجموعه‌ای از پاسخ‌های بهینه به عنوان جواب به دست می‌آیند که «جبهه پارتو» خوانده می‌شوند. مجموعه راه‌حل‌های انتخابی در جبهه پارتو، راه‌حلی هستند که تحت سلطه هیچ راه‌حل دیگری نیستند و توسط مرجع در فضای مسئله پراکنده شده‌اند. تعداد نقاط مرجع توسط کارشناسان و یا با صورت سیستماتیک تعیین می‌شود. در یک مسئله با m تابع هدف، نقاط مرجع روی یک ابرصفحه نرمال‌سازی شده با تعداد $m-1$ بعد واقع شده است. به نحوی که ابر صفحه در هر بعد به p قسمت مساوی تقسیم می‌شود.

تعداد کل نقاط مرجع با استفاده از روابط زیر محاسبه می‌شود. همچنین برای محاسبه‌ی تابع هدف از (۲۱) کمک گرفته شده است که در آن α و β ضرایب اهمیت هر یک از اهداف را نشان می‌دهند. در (۲۲) با کمک تابع $Normalize$ داده‌ها نرمال‌سازی می‌گردند تا تاثیر هر یک از اهداف به یک میزان مساوی در نظر گرفته شود.

$$\min(\alpha \times Normalize(ResponseTime) + \beta \times Normalize(Energy)) \quad (21)$$

$$Normalize(x, x_i) = \frac{abs[\text{mean}(x) - x_i]}{\text{std}(x)} \quad (22)$$

۵-۳ آزمون همگرایی

برای به پایان رسیدن الگوریتم یکی از موارد زیر باید اتفاق بیفتد: الف) اختلاف بین بهترین راه‌حلی که در دو تکرار متوالی پیدا شده‌اند، باید کمتر از مقدار آستانه تعیین شده باشد. ب) تعداد تکرار الگوریتم به یک آستانه از پیش تعیین شده برسد. شکل ۶ شبه‌کد الگوریتم پیشنهادی آورده شده است که نحوه ترکیب الگوریتم فراابتکاری چتر دریایی و ژنتیک را شرح می‌دهد. در ابتدا جمعیت

S	۲	۱	۲	۳	۴	۷	۶	۷	۸	۹	۱۰	۱
S_{new}	۲	۱	۱	۳	۵	۷	۶	۱	۲	۹	۱۰	۱۰

شکل ۳: موقعیت ذره پس از حرکت در جریان اقیانوس.

S	۲	۱	۲	۳	۴	۷	۶	۷	۸	۹	۱۰	۱
S_{new}	۲	۱	۶	۳	۴	۸	۶	۷	۸	۹	۱۰	۵

شکل ۴: موقعیت ذره پس از حرکت فعال.

۵-۲-۱ حرکت همسو با جریان اقیانوس

در حرکت همسو با جریان اقیانوس^۱ حرکت موقعیت چتر دریایی که به شکل آرایه مدل شده است به کمک اعداد تصادفی در فضای مسئله به‌روزرسانی می‌شود. به صورت تصادفی به تعداد نیمی از درایه‌های آرایه انتخاب شده و سپس به همان تعداد برای هر یک از درایه‌های انتخاب شده، به صورت تصادفی یک ماشین مجازی به صورت تصادفی انتخاب شده و جایگزین مقدار آن درایه (ماشین مجازی فعلی) می‌شود. به عنوان مثال با فرض اینکه S ذره مورد بررسی می‌باشد و اندیس‌های ۲، ۴، ۵، ۷، ۸ و ۱۱ انتخاب شده باشد همچنین ماشین‌های مجازی انتخاب شده (با فرض اینکه ۱۰ ماشین مجازی موجود باشد) به ترتیب ۵، ۱، ۷، ۱، ۲ و ۱۰ باشد، ذره جدید q_{new} به صورت شکل ۳ خواهد بود.

۵-۲-۲ حرکت غیرفعال

در حرکت غیرفعال^۲ موقعیت چتر دریایی به کمک عملگر پیشنهادی به روزرسانی می‌شود. عملگر پیشنهادی به این صورت عمل می‌کند که به صورت تصادفی به تعداد نیمی از درایه‌های آرایه (راه‌حل مورد بررسی) انتخاب شده و سپس به همان تعداد برای هر یک از درایه‌های انتخاب شده، به صورت تصادفی یک ماشین مجازی به صورت تصادفی انتخاب شده و جایگزین مقدار درایه انتخاب شده، می‌گردد. برای مدیریت بین این نوع حرکت و پذیرش یا عدم پذیرش راه‌حل تولید شده از مکانیسم کنترل زمان در الگوریتم چتر دریایی کمک گرفته شده است. به این صورت است که در ابتدا اگر جواب همسایه بهتر از جواب فعلی باشد، الگوریتم آن را به عنوان جواب فعلی قرار می‌دهد اما با گذشت زمان، احتمال پذیرش کمتر می‌شود. به عنوان مثال با فرض اینکه s ذره مورد بررسی می‌باشد و اگر اندیس‌های ۲، ۴، ۵، ۷، ۸ و ۱۱ به صورت تصادفی انتخاب شده باشد و از بین ماشین‌های مجازی موجود، ماشین‌های مجازی شماره ۶، ۴، ۸، ۷، ۸ و ۵ به صورت تصادفی انتخاب شده باشد ذره جدید q_{new} به صورت شکل ۴ خواهد بود.

۵-۲-۳ حرکت فعال

در حرکت فعال^۳ موقعیت چتر دریایی به کمک عملگر پیشنهادی به‌روزرسانی می‌شود. عملگر پیشنهادی در این بخش از عملگر آمیزش الگوریتم ژنتیک ایده گرفته است. الگوریتم ژنتیک به عنوان یکی از مهمترین الگوریتم‌های فراابتکاری مبتنی بر جمعیت شناخته می‌شود. در این الگوریتم اطلاعات گذشته در روند جستجو مورد استفاده قرار می‌گیرند. می‌توان الگوریتم ژنتیک را به عنوان یک رویکرد مناسب برای حل مسئله زمان‌بندی کارها ساخت. این عملگر بر روی بهترین کروموزوم

1. Ocean Current
2. Passive Movements
3. Active Movements

EMJFGA Algorithm

Input: $Tasks, Vms, Max_Iterations, population\ size(Npop)$ Output: The Optimal (*Sub-Optimal*) Tasks Assignment

Steps:

1. Using Logistic Map for initialize the population $X_{pop} // X_{pop}$ is population
2. Calculate the fitness of each member of population
3. Run pareto front (find non-dominate solution)
4. For $t = 1$ to $Max_Iterations$
5. For $i = 1$ to $N_{pop} // N_{pop}$ is population size
6. $c_t = (1 - (iter / max_iter)) * (2 * Math.random() - 1); /* Run Time control mechanism*/$
7. if $c_t \geq 0.5$ {Ocean_current(X_{pop}, Vms); }
8. Elseif ($Math.random() > 1 - c_t$)
9. Jellyfish_swarm_typeA(X_{pop}, Vms); //type A: passive motions
10. Else
11. Jellyfish_swarm_typeB(X_{pop}, Vms); //type B: Active motions that using GA Algorithm
12. Calculate the fitness of each member of population
13. Run pareto front (find non-dominate solution)
14. End For
15. End For
16. Return $x_{\alpha} // Best\ solution$

شکل ۶: شبه‌کد الگوریتم پیشنهادی.

در نظر گرفته شده است. این مقاله به بررسی سیستم بهداشت و درمان و مراقبت‌های پزشکی می‌پردازد که با توضیح مختصری در مورد سناریوی مورد استفاده ادامه می‌یابد. زندگی روزمره افراد به طور کامل توسط اینترنت اشیا دگرگون شده است. در حوزه سلامت و درمان نیز کادر پزشکی و متخصصان می‌توانند از اینترنت اشیا بهره‌فرآوانی را ببرند.

نمایش گذاشته می‌شوند. الگوریتم زمان‌بندی کار پیشنهادی با استفاده از شبیه‌سازی IoT-Sim-Osmosis مورد ارزیابی قرار گرفته است که امکان دستکاری پارامترهای محیطی و تکرار آزمایش با سناریوها و محدودیت‌های متفاوت را فراهم می‌کند. برای آزمایش اثربخشی الگوریتم پیشنهادی، از سناریوهای متفاوت استفاده شده است. سناریوی مورد استفاده از سه لایه تشکیل گردیده است (IoT-fog-cloud). اولین لایه از تعداد زیادی سنسور تشکیل شده است. لایه دوم شامل گره‌های مه است که برای جمع‌آوری داده‌ها و پردازش داده‌های تولید شده توسط لایه اول استفاده می‌شود. فرض بر این است که لایه مه شامل یک مرکز داده است که از طریق اینترنت به لایه ابری متصل است. همچنین در هر گره مه، تعدادی ماشین مجازی برای پردازش وظایف در نظر گرفته شده است. این مقاله به بررسی سیستم بهداشت و درمان و مراقبت‌های پزشکی می‌پردازد که با توضیح مختصری در مورد سناریوی مورد استفاده ادامه می‌یابد. زندگی روزمره افراد به طور کامل توسط اینترنت اشیا دگرگون شده است. در حوزه سلامت و درمان نیز کادر پزشکی و متخصصان می‌توانند از اینترنت اشیا بهره‌فرآوانی را ببرند.

استفاده از محاسبات مه در صنایعی که نیازمند زمان پاسخ سریع هستند، توصیه شده است. از این رو محیط‌های IoT-fog-cloud برای ذخیره داده‌ها و انجام محاسبات به کار می‌روند. در این مقاله به بررسی تکنیک‌های توسعه یک سیستم بهداشت و درمان و مراقبت‌های پزشکی می‌پردازیم. ما پیش‌بینی می‌کنیم که نتایج این ارزیابی به محققان در توسعه یک سیستم کارآمد برای این حوزه کمک کند. پایش لحظه‌ای وضعیت جسمانی فرد با استفاده از تجهیزات مختلف مانند دستبند یا ساعت هوشمند، می‌تواند جان فرد را در شرایط اضطراری مانند دیابت، حمله قلبی نجات دهد. داده‌های مختلف مانند اکسیژن

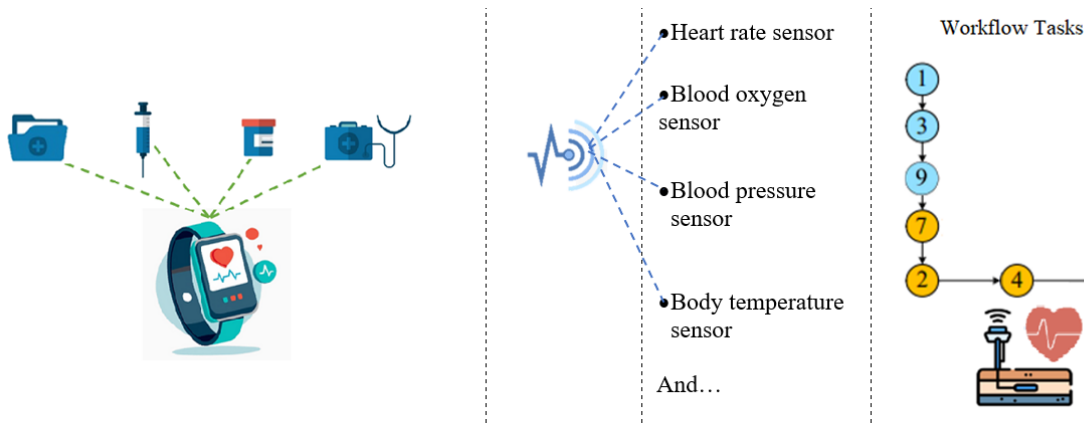
اولیه با استفاده از نظریه آشوب ایجاد می‌شود و سپس میزان برانزندی هر راه‌حل (که به عنوان یک چتر دریایی در نظر گرفته می‌شود) مشخص می‌شود. در ادامه به کمک مکانیسم کنترل زمان حرکت چترهای دریایی کنترل می‌شود که برای حرکت چترهایی دریایی از عملگرهای الگوریتم ژنتیک نیز بهره‌گرفته شده است. در واقع به کمک الگوریتم ژنتیک موقعیت چتر دریایی در فضای مسئله که یک راه‌حل برای مسئله پیش‌رو می‌باشد، تعیین می‌شود. سپس تابع محاسبه برانزندی فراخوانی می‌شود و برای پیدا کردن راه‌حل بهینه برای مسئله‌ی پیش‌رو که چند هدفه می‌باشد از جبهه پرتو استفاده شده است. این رویه تا رسیدن تعداد تکرار اجرای الگوریتم به یک تعداد معین ادامه می‌یابد.

۶- تحلیل عددی

در این بخش به بررسی عملکرد الگوریتم پیشنهادی با دیگر الگوریتم‌های زمان‌بندی مانند IJFA [۳۹]، GA-PSO [۴۰]، PWAO [۴۱] و MFO-PSO-GA [۴۲] می‌پردازیم. به همین جهت به بررسی (۱) زمان اجرا، (۲) تاخیر و (۳) مصرف انرژی به منظور نشان دادن بهبود عملکرد سیستم پرداخته شده است.

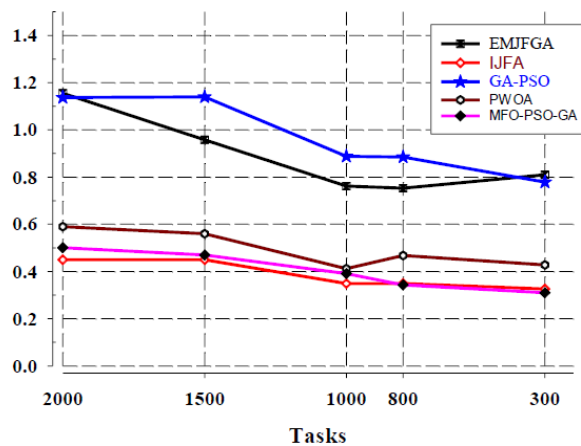
۶-۱ پارامترهای تنظیم

در این بخش نتایج آزمایش‌ها برای ارزیابی الگوریتم پیشنهادی به نمایش گذاشته می‌شوند. الگوریتم زمان‌بندی کار پیشنهادی با استفاده از شبیه‌سازی IoT-Sim-Osmosis مورد ارزیابی قرار گرفته است که امکان دستکاری پارامترهای محیطی و تکرار آزمایش با سناریوها و محدودیت‌های متفاوت را فراهم می‌کند. برای آزمایش اثربخشی الگوریتم پیشنهادی، از سناریوهای متفاوت استفاده شده است. سناریوی مورد استفاده از سه لایه تشکیل شده است (IoT-fog-cloud). اولین لایه از تعداد زیادی سنسور تشکیل شده است. لایه دوم شامل گره‌های مه است که برای جمع‌آوری داده‌ها و پردازش داده‌های تولید شده توسط لایه اول استفاده می‌شود. فرض بر این است که لایه مه شامل یک مرکز داده است که از طریق اینترنت به لایه ابری متصل است. همچنین در هر گره مه، تعدادی ماشین مجازی برای پردازش وظایف



شکل ۷: سیستم بهداشت و درمان و مراقبت های پزشکی.

جدول ۲: مقدار پارامترها در شبیه سازی های انجام شده.



شکل ۸: نمودار تغییر مدت زمان اجرای کارها با افزایش تعداد درخواستها (تعداد ماشین مجازی = ۲۵).

مانند اندازه ورودی، اندازه خروجی، میزان مصرف حافظه و تعداد دستورالعملها می باشد. این شبیه سازی بر روی رایانه شخصی با پردازنده Core i۸ اینتل ۲٫۷۰ گیگاهرتز و حافظه ۸ گیگابایت اجرا شده است. جدول ۲ تنظیمات انجام شده بر روی پارامترهای به کار رفته در آزمایشها را فهرست می کند. چندین آزمایش اولیه برای تنظیمات پارامترها انجام شده است. اندازه جمعیت و تعداد تکرارها در هر سناریو به ترتیب ۱۰۰ و ۱۰ در نظر گرفته شده است. تمام عوامل داخلی باقیمانده در این موارد در جدول ۳ به تفصیل آمده است.

۲-۶ بررسی نتایج حاصل از شبیه سازی

در این قسمت نتایج اجرای شبیه سازی برای اینج الگوریتم از نظر زمان اجرا، درصد وظایفی که مهلت اجرای آنها رعایت شده است (PDST) و میزان مصرف انرژی تحت سناریوهای مختلف نمایش داده می شود. نتایج هر سناریو برای تعداد کارهای مختلف در بخش بعدی مقاله آورده شده است. به منظور بررسی کارایی الگوریتم ارائه شده، شبیه سازی برای کارهایی با نرخ ورودی و داده های مختلف با اندازه های مختلف تکرار شده است.

۱-۲-۶ زمان اجرای کل

زمان اجرای کل چندین الگوریتم از جمله EMJFGA، IJFA،

نماد	معنی	مقدار	نماد	معنی	مقدار
D_i	اندازه خروجی وظیفه i	۱۰-۱۰۰	$t_{IoTDevice_i}^s$	میزان حافظه مورد نیاز وظیفه i	۲۰-۱۰۰۰
m	تعداد گره های مهلت اجرای وظیفه i	۲۵-۱۰۰	t_i^{memory}	میزان داده ی تولید شده توسط وظیفه i	۵۰-۲۰۰
t_i^d	نرخ وظایف	۱۰۰-۵۰۰	t_i^s	اولویت دستگاه	۲۵۰-۱۰۲۴
λ_{Di}	ارسالی کارگزار	۲۰۰-۲۰۰۰	App_{Di}		۱-۳

خون، سطح خواب، ضربان قلب با اتصال دستگاه های مختلف مبتنی بر اینترنت اشیا جمع آوری شده و سپس تجزیه و تحلیل شده و پیغام مناسب به بیمار نشان می دهد. حسگرها باید بتوانند اطلاعات مورد نیاز را با عملکرد دقیق جمع آوری کرده و برای انجام فرآیند درمان به کنترل کننده مرکزی ارسال کنند. سیستم پیشنهادی این پتانسیل را دارد که نقش مهمی در پردازش حجم عظیمی از داده های مراقبت های بهداشتی تولید شده توسط میلیاردها دستگاه هوشمند داشته باشد. دسترسی به اطلاعات پزشکی در هر نقطه مستلزم یک فرآیند تصمیم گیری به موقع و هوشمند است. در سیستم مدنظر هر وظیفه دارای داده ها، مهلت اجرا و میزان پردازنده مورد نیاز خود در هر اجرا در طول فرآیند تصمیم هستند در شکل ۷ دارای سه نوع وظیفه هستند.

در این مقاله از قاعده توالی برای زمان بندی وظایف استفاده شده است. وظایف حساس به تاخیر به گره لبه اختصاص داده می شوند. لازم به ذکر است که حجم کاری درخواستی باید در حین پردازش کمتر از منابع آنها باشد. الگوریتم پیشنهادی به طور مکرر تمام وظایف را به گره های محاسباتی ناهمگن اختصاص می دهد. لازم به ذکر است برای شبیه سازی از اطلاعات مربوط به مجموعه داده محاسبات ابری، گره های اینترنت اشیا نزدیک به ۵۰ هزار نمونه وظیفه استفاده شده است. هر سنسور بسته به وظیفه سنسور ۲۵۰-۱۰۲۴ کیلوبایت داده تولید می کند. میزان تاخیر برابر ۱۰۰ میلی ثانیه در نظر گرفته شده است. لایه دوم از ۲۰-۸ گره مه تشکیل شده است. در این مقاله هر گره مه دارای ویژگی های حافظه، نرخ پردازنده و پهنای باند خاص خود است. همچنین هر درخواست وارد شده به سیستم دارای ویژگی هایی

جدول ۳: مقادیر پارامترهای داخلی برای الگوریتم‌های فراابتکاری استفاده شده در شبیه‌سازی‌ها.

مقدار	معنی	نماد	الگوریتم	سال
۳	توزیع فضایی	β	IJFA	۲۰۲۲
۰/۱	ضرایب حرکت	γ		
۲	ضریب‌گیری شخصی و سراسری	C_1 و C_2		
۰/۱	حفظ حرکت ذرات در جهت و سرعت ثابت	I_w		
[۰,۱]	متغیرهای تصادفی	r_1 و r_2	PWOA	۲۰۲۳
۱	متغیر ثابت برای تعریف شکل ماریچ لگاریتمی	b		
[۰,۱]	نسبت میرایی وزن اینرسی	w_{damp}		
۰/۰۰۱	متغیرهای تصادفی	r_1 و r_2	GA-PSO	۲۰۲۳
۲	ضریب‌گیری شخصی	C_1		
۲	ضریب‌گیری سراسری	C_2		
[۰,۱]	متغیرهای تصادفی	r_1 و r_2	MFO-PSO-GA	۲۰۲۵

جدول ۵: مدت زمان اجرا در شبیه‌سازی‌های انجام شده (تعداد ماشین مجازی=۲۵ و تعداد درخواست‌های وارد شده به سیستم $=\{2000, 1500, 1000, 800, 300\}$.)

تعداد وظیفه الگوریتم	۳۰۰	۸۰۰	۱۰۰۰	۱۵۰۰	۲۰۰۰
EMJFGA	$2,83E+13$	$4,37E+15$	$7,03E+15$	$8,61E+14$	$7,03E+15$
IJFA	$1,93E+13$	$2,02E+15$	$4,33E+15$	$3,39E+14$	$4,33E+15$
GA-PSO	$1,93E+13$	$2,01E+15$	$4,59E+15$	$3,91E+14$	$4,59E+15$
PWAO	$1,50E+13$	$1,17E+15$	$1,52E+15$	$3,10E+14$	$1,52E+15$
MFO-PSO-GA	$2,53E+13$	$2,01E+15$	$4,33E+15$	$3,39E+14$	$4,33E+15$

جدول ۴: مدت زمان اجرا در شبیه‌سازی‌های انجام شده (تعداد ماشین مجازی=۲۵ و تعداد درخواست‌های وارد شده به سیستم $=\{2000, 1500, 1000, 800, 300\}$.)

تعداد وظیفه الگوریتم	۳۰۰	۸۰۰	۱۰۰۰	۱۵۰۰	۲۰۰۰
EMJFGA	۰/۳۲۶	۰/۳۴۹	۰/۳۴۹	۰/۴۵۰	۰/۴۶۰
IJFA	۰/۸۱۰	۰/۷۵۳	۰/۷۶۲	۰/۹۵۸	۱/۱۵۵
GA-PSO	۰/۴۲۸	۰/۴۶۸	۰/۴۱۲	۰/۵۶۰	۰/۵۹۰
PWAO	۰/۷۷۹	۰/۸۸۵	۰/۸۸۸	۱/۱۴۰	۱/۱۲۷
MFO-PSO-GA	۰/۴۱۱	۰/۳۷۹	۰/۴۰۱	۰/۴۹۰	۰/۵۰۱

انرژی این الگوریتم با کاهش تعداد وظایف مشاهده می‌شود. در مقایسه با الگوریتم‌های دیگر، به‌ویژه GA-PSO و IJFA که در اکثر موارد مصرف انرژی بیشتری دارند، الگوریتم EMJFGA موفق به کاهش چشمگیر در مصرف انرژی شده است. اگرچه الگوریتم PWOA نیز عملکرد نسبتاً مناسبی دارد، اما در بارهای کاری بالا از نظر پایداری و بهینگی در مصرف انرژی، نسبت به الگوریتم پیشنهادی ضعیف‌تر عمل کرده است. این نتایج نشان‌دهنده‌ی کارایی انرژی بالای الگوریتم EMJFGA و قابلیت آن در بهینه‌سازی مصرف منابع انرژی در سیستم‌های محاسباتی می‌باشد.

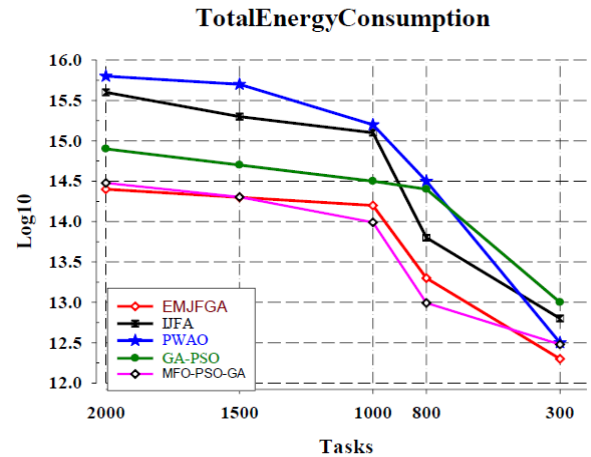
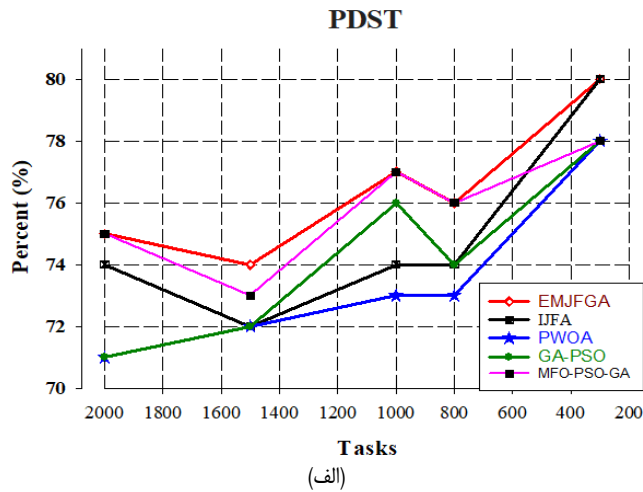
۶-۲-۳ زمان اجرای کل درصد وظایفی که مهلت اجرای آن‌ها رعایت شده است (PDST)

در این بخش، به بررسی PDST برای الگوریتم‌های مورد بررسی، پرداخته شده است. پس از ارزیابی الگوریتم پیشنهادی، نتایج بیانگر آن است که هر الگوریتم EMJFGA با افزایش تعداد کارهای وارد شده به سیستم، تعداد کارهایی که مهلت اجرای آن‌ها رعایت شده است در مقایسه با سایر الگوریتم‌های مورد بررسی بیشتر می‌باشد. برای مثال در حالتی که تعداد کارهای وارد شده به سیستم برابر ۱۰۰۰ می‌باشد عدد PDST یعنی درصد وظایفی که مهلت اجرای آن‌ها رعایت شده است، برای الگوریتم‌های EMJFGA، IJFA، GA-PSO، MFO-PSO-GA و PWOA به ترتیب برابر ۷۷، ۷۴، ۷۶، ۷۷ و ۷۳ می‌باشد که با توجه به

MFO-PSO-GA، PWAO و GA-PSO در جدول ۴ نشان داده شده است. همانطور که در شکل ۸ نشان داده شده است، عملکرد الگوریتم پیشنهادی EMJFGA از نظر مدت زمان اجرای کل در مقایسه با چهار الگوریتم دیگر شامل IJFA، GA-PSO، MFO-PSO-GA و PWOA مورد ارزیابی قرار گرفته است. نتایج به‌دست‌آمده از جدول و نمودار نشان می‌دهند که الگوریتم EMJFGA در اغلب سناریوها با تعداد وظایف مختلف، زمان اجرای کمتری نسبت به دیگر الگوریتم‌ها داشته و از پایداری بالاتری برخوردار است. اگرچه الگوریتم ترکیبی MFO-PSO-GA در برخی سناریوها، به‌ویژه در وظایف با تعداد پایین‌تر (مانند ۳۰۰ وظیفه)، زمان اجرای کمتری نسبت به EMJFGA دارد، اما با افزایش بار وظیفه، الگوریتم پیشنهادی عملکرد پایدارتر و کارآمدتری را نشان می‌دهد.

۶-۲-۲ میزان مصرف انرژی

در این بخش مقدار انرژی مصرف‌شده توسط الگوریتم‌های زمان‌بندی EMJFGA، IJFA، PWAO، MFO-PSO-GA و GA-PSO در طیف وسیعی از میزان درخواست‌های وارد شده به سیستم و طیف وسیعی از حاصل از مقایسه میزان مصرف انرژی الگوریتم‌ها در شکل ۹ و جدول ۵ ارائه شده است. بررسی نتایج بیانگر آن است که الگوریتم پیشنهادی EMJFGA در اغلب سناریوها دارای کمترین مصرف انرژی نسبت به سایر الگوریتم‌ها می‌باشد. روند نزولی واضحی در مصرف



شکل ۹: نمودار تغییر میزان مصرف انرژی با افزایش تعداد درخواست‌ها (تعداد ماشین مجازی = ۲۵).

عدد PDST تعداد کارهایی که مهلت اجرای آن‌ها رعایت شده است به ترتیب برابر ۷۷۰، ۷۴۰، ۷۶۰، ۷۷۰ و ۷۳۰ خواهد بود (شکل ۱۰-الف). همچنین می‌توان نتیجه گرفت که افزایش تعداد گره‌ها باعث افزایش عدد PDST و در نتیجه تعداد کارهایی می‌شود که مهلت اجرای آن‌ها رعایت شده است (شکل ۱۰-ب).

برای اطمینان بیشتر و توصیف دقیق نتایج، فاصله اطمینان ۹۵ درصد در نظر گرفته شده است. برای دستیابی به این هدف، سناریو ۱۰ بار تکرار شد، نتایج شبیه‌سازی چندین الگوریتم از جمله EMJFGA، IJFA، PWAO، GA-PSO و MFO-PSO-GA در جدول ۶ نمایش داده شده است.

۷- نتیجه‌گیری و کارهای آینده

ماهیت متفاوت و اولویت متنوع وظایف وارد شده به لایه مه منجر به زمان اجرای کل طولانی‌تر، مصرف انرژی بیشتر و تاخیر در تکمیل کار می‌شود و در نهایت منجر به نرخ بالاتری از شکست محاسباتی کار می‌شود. این مطالعه تکنیکی را با استفاده از الگوریتم چتر دریایی و الگوریتم ژنتیک برای افزایش عملکرد و کاهش مصرف انرژی معرفی کرده است. این روش فرض می‌کند که داده‌ها از گره‌های IoT به گره‌های مه برای پردازش اولیه ارسال می‌شوند. در این پژوهش مدل EMJFGA مورد ارزیابی قرار گرفته است و یافته‌ها نشان می‌دهد که این تکنیک به طور قابل توجهی QoS و SLA را افزایش می‌دهد. توزیع کارآمد منابع براساس ویژگی‌های درخواست وارد شده به سیستم با روش پیشنهادی می‌تواند مصرف انرژی و زمان اجرای کل را کاهش دهد و در عین حال PDST را نیز افزایش دهد. در مرحله بعد، مدل پیشنهادی با استفاده از الگوریتم‌هایی مانند IJFA، GA-PSO، PWAO و MFO-PSO-GA به کار گرفته شد و یافته‌ها در موقعیت‌های مختلف مورد بررسی و مقایسه قرار گرفتند و نتایج حاصل از شبیه‌سازی بیانگر آن است که روش پیشنهادی نسبت به سایر روش‌های مورد بررسی نتایج بهتری داشته و منجر به بهبود کارایی سیستم می‌شود. بر اساس یافته‌های این مطالعه، تحقیقات آینده ما بر بررسی راه‌هایی برای افزایش توزیع بارکاری سیستم و ارزیابی عملکرد سیستم در زمان ازدحام متمرکز خواهد بود.

مراجع

- [2] T. Salehnia, et al., "An optimal task scheduling method in IoT-Fog-cloud network using multi-objective moth-flame algorithm," *Multimedia Tools and Applications*, vol. 83, pp. 34351-34372, 2024.
- [3] T. Bezdán, et al., "Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm," *Journal of Intelligent & Fuzzy Systems*, vol. 42, no. 1, pp. 411-423, Dec. 2021.
- [4] D. A. Amer, G. Attiya, I. Zeidan, and A. A. Nasr, "Elite learning Harris hawks optimizer for multi-objective task scheduling in cloud computing," *The Journal of Supercomputing*, vol. 78, no. 2, pp. 2793-2818, Feb. 2022.
- [5] G. Fortino, A. Guerrieri, P. Pace, C. Savaglio, and G. Spezzano, "IoT platforms and security: An analysis of the leading industrial/commercial solutions," *Sensors*, vol. 22, no. 6, Article ID: 2196, Mar.-2022.
- [6] I. Ullah, et al., "Integration of data science with the intelligent IoT (IIoT): current challenges and future perspectives," *Digital Communications and Networks*, vol. 11, no. 2, pp. 280-298, Apr. 2024.
- [7] M. Laroui, et al., "Edge and fog computing for IoT: A survey on current research activities & future directions," *Computer Communications*, vol. 180, pp. 210-231, Dec. 2021.
- [8] D. Milošević, "The edge-to-cloud continuum," *Computer*, vol. 53, no. 11, pp. 16-25, Nov. 2020.
- [9] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. of MCC Workshop on Mobile Cloud Computing*, pp. 13-16, Low Wood Bay, UK, 17-17 Aug. 2012.
- [10] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: fundamental, network applications, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826-1857, Thirdquarter 2018.
- [11] I. Attiya, et al., "An improved hybrid swarm intelligence for scheduling IoT application tasks in the cloud," *IEEE Trans. on Industrial Informatics*, vol. 18, no. 9, pp. 6264-6272, Sept. 2022.
- [12] J. Lim, "Latency-aware task scheduling for IoT applications based on artificial intelligence with partitioning in small-scale fog computing environments," *Sensors*, vol. 22, no. 19, Article ID: 7326, Oct.-12022.
- [1] M. Ghorbian, M. Ghobaei-Arani, and L. Esmaeili, "A survey on the scheduling mechanisms in serverless computing: a taxonomy, challenges, and trends," *Cluster Computing*, vol. 27, no. 5, pp. 5571-5610, 2024.

جدول ۶: نتایج حاصل از شبیه‌سازی با تعداد تکرار برابر ۱۰ (تعداد ماشین مجازی=۲۵، تعداد درخواست‌های وارد شده به سیستم = ۲۰۰۰)

	EMJFGA	IJFA	GA-PSO	PWAO	MFO-PSO-GA	
میانگین	۱۳,۴۵۷	۱۵,۶۳۴	۱۴,۹۴۰	۱۵,۸۵۴	۱۳,۹۴۰	Total Energy Consumption (log ₁₀)
انحراف معیار	۰,۱۷	۰,۱۴۳	۰,۲۶۶	۰,۲۲۵	۰,۲۸۰	
فاصله اطمینان [کران پایین]	۱۳,۴۶۲	۱۵,۷۱۸	۱۴,۹۱۹	۱۵,۶۴۵	۱۳,۵۵	
فاصله اطمینان [کران بالا]	۱۳,۴۷۳	۱۵,۹۵۸	۱۴,۹۵۹۹	۱۵,۹۶۴	۱۳,۹۵	
میانگین	۰,۴۹	۱,۱۳	۰,۶۲	۱,۱۳	۰,۶۲	Total Execution Time
انحراف معیار	۰,۰۲۷	۰,۰۱۸	۰,۰۲۵	۰,۰۱۵	۰,۰۲۱	
فاصله اطمینان [کران پایین]	۰,۴۳	۱,۰۹	۰,۵۷	۱,۱۱	۰,۵۵	
فاصله اطمینان [کران بالا]	۰,۵۵	۱,۱۶	۰,۶۶	۱,۱۶	۰,۶۴	
میانگین	۷۶	۷۴	۷۰	۷۱	۷۴	PDST
انحراف معیار	۰,۰۸۷	۰,۷۵۵	۱,۰۴۸	۱,۰۴۷	۰,۶۵۲	
فاصله اطمینان [کران پایین]	۷۴,۹۲	۷۳,۵۰	۶۹,۲۲	۶۹,۵۸	۷۳,۹۲	
فاصله اطمینان [کران بالا]	۷۶,۸۸	۷۵,۳۰	۷۱,۵۸	۷۲,۶۲	۷۵,۹۸	

- [28] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, "Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach," *Journal of Network and Computer Applications*, vol. 201, Article ID: 103333, May2022.
- [29] J. U. Arshed and M. Ahmed, "Race: resource aware cost-efficient scheduler for cloud fog environment," *IEEE Access*, vol. 9, pp. 65688-65701, 2021.
- [30] T. S. Nikoui, A. Balador, A. M. Rahmani, and Z. Bakhshi, "Cost-aware task scheduling in fog-cloud environment," in *Proc. CSI/CPSSI Int. Symp. on Real-Time and Embedded Systems and Technologies*, 8 pp., Tehran, Iran, 9-11Jun. 2020.
- [31] M. K. Hussein and M. H. Mousa, "Efficient task offloading for IoT-based applications in fog computing using ant colony optimization," *IEEE Access*, vol. 8, pp. 37191-37201, 2020.
- [32] J. U. Arshed, et al., "GA-IRACE: Genetic algorithm-based improved resource aware cost-efficient scheduler for cloud fog computing environment," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID: 6355192, 2022.
- [33] Y. Sun, F. Lin, and H. Xu, "Multi-objective optimization of resource scheduling in fog computing using an improved NSGA-II," *Wireless Personal Communications*, vol. 102, pp. 1369-1385, 2018.
- [34] E. Elsedimy and F. Algarni, "MOTS-ACO: An improved ant colony optimiser for multi-objective task scheduling optimisation problem in cloud data centres," *IET Networks*, vol. 11, no. 2, pp. 43-57, 2022.
- [35] S. M. Mirmohseni, C. Tang, and A. Javadpour, "FPSO-GA: a fuzzy metaheuristic load balancing algorithm to reduce energy consumption in cloud networks," *Wireless Personal Communications*, vol. 127, no. 4, pp. 2799-2821, 2022.
- [36] S. Banihashemi, et al., "A blockchain-assisted ant colony optimization algorithm for task scheduling in cloud computing," *Journal of Cloud Computing*, vol. 10, Article ID: 34, March 2021.
- [37] S. D. Vispute and P. Vashisht, "Energy-efficient task scheduling in fog computing based on particle swarm optimization," *SN Computer Science*, vol. 4, no. 4, Article ID: 391, Jul. 2023.
- [38] S. Bansal and H. Aggarwal, "A hybrid particle whale optimization algorithm with application to workflow scheduling in cloud-fog environment," *Decision Analytics Journal*, vol. 9, Article ID: 100361, Dec. 2023.
- [39] 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, ArticleID: 98, 2022.
- [40] M. Guo, H. Yang, Y. Zhang, and X. Luo, "A novel GA-PSO-based optimization strategy for fractional grey forecasting models," *Sensors*, vol. 23, no. 2, Article ID: 636, Jan. 2023.
- [41] M. K. Mafarja, et al., "Binary whale optimization algorithm approaches for feature selection," *Applied Soft Computing*, vol. 120, Article ID: 108593, May2023.
- [42] M. Saad, R. N. Enam, and R. Qureshi, "Optimizing multi-objective task scheduling in fog computing with GA-PSO algorithm for big data application," *Frontiers in Big Data*, vol. 7, Article ID: 1358486, 2024.
- [13] S. M. Hussain and G. R. Begh, "Hybrid heuristic algorithm for cost-efficient QoS aware task scheduling in fog-cloud environment," *Journal of Computational Science*, vol. 64, Article ID: 101828, Oct. 2022.
- [14] N. Ghani, "A comprehensive review of fog computing: Concepts, architectures, parameters, and challenges," *Int. J. Informatics Visualization*, vol. 9, no. 2, pp. 121-129, May 2025.
- [15] Z. R. Ahmed, S. Askar, D. H. Hussein, and M. A. Ibrahim, "Fog computing challenges and opportunities in IoT networks: A review," *Procedia Computer Science*, vol. 259, pp. 1749-1764, 2025.
- [16] I. Mokni and S. Yassa, "A multi-objective approach for optimizing IoT applications offloading in fog-cloud environments with NSGA-II," *The Journal of Supercomputing*, vol. 80, pp. 27034-27072, 2024.
- [17] I. Attiya et al., "An intelligent chimp optimizer for scheduling of IoT application tasks in fog computing," *Mathematics*, vol. 10, no. 7, Article ID: 1100, Apr.-12022.
- [18] L. Liu, D. Qi, N. Zhou, and Y. Wu, "A task scheduling algorithm based on classification mining in fog computing environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID: 2102348, 2018.
- [19] A. H. Attar and A. Sutagundar, "A survey on resource management for fog-enhanced services and applications," *International Journal of Scientific Research*, vol. 17, no. 2, Article ID: 138, 2018.
- [20] W. Jing, et al., "QoS-DPSO: QoS-aware task scheduling for cloud computing system," *Journal of Network and Systems Management*, vol. 29, no. 1, Article ID:5, Jan. 2021.
- [21] A. S. Abohamama, A. El-Ghamry, and 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, Article ID:54, 2022.
- [22] A. M. Yadav, K. N. Tripathi, and S. C. Sharma, "An enhanced multi-objective fireworks algorithm for task scheduling in fog computing environment," *Cluster Computing*, Article ID: 5, 16 pp., 2022.
- [23] M. Ghorbian, M. Ghobaei-Arani, and L. Esmaeili, "A survey on the scheduling mechanisms in serverless computing: A taxonomy, challenges, and trends," *Cluster Computing*, vol. 27, no. 5, pp. 5571-5610, Aug.2024.
- [24] R. Aghazadeh, A. Shahidinejad, and M. Ghobaei-Arani, "Proactive content caching in edge computing environment: A review," *Software: Practice and Experience*, vol. 53, no. 3, pp. 811-855, 2023.
- [25] M. Tari, et al., "Auto-scaling mechanisms in serverless computing: A comprehensive review," *Computer Science Review*, vol. 53, Article ID:100650, Aug. 2024.
- [26] R. Ghafari and N. Mansouri, "Cost-aware and energy-efficient task scheduling based on grey wolf optimizer," *Journal of Mahani Mathematical Research*, vol. 12, no. 1, pp. 257-288, 2023.
- [27] S. Mangalampalli, G. R. Karri, and A. A. Elngar, "An efficient trust-aware task scheduling algorithm in cloud computing using firefly optimization," *Sensors*, vol. 23, no. 3, Article ID: 1384, Feb.-12023.

محمد نصیری دانشیار مهندسی کامپیوتر دانشگاه بوعلی سینا، پس از اخذ مدرک کارشناسی ارشد از دانشگاه صنعتی شریف (ایران)، مدرک دکتری خود را در رشته مهندسی کامپیوتر در سال ۲۰۰۸ از Grenoble INP (فرانسه) دریافت نمودند. ایشان در تابستان ۲۰۱۹ به عنوان پژوهشگر مهمان در دانشگاه Grenoble Alpes (فرانسه) فعالیت داشتند. علایق پژوهشی ایشان عمدتاً بر بهبود عملکرد فناوری‌های مختلف بی‌سیم، از جمله شبکه‌های محلی بی‌سیم، شبکه‌های حسگر بی‌سیم، شبکه‌های زیر آب، و فناوری‌های اتصال متقابل برای اینترنت اشیا متمرکز است.

رضا محمدی دانشیار مهندسی کامپیوتر دانشگاه بوعلی سینا از سال ۲۰۱۸، مدرک کارشناسی ارشد و دکتری خود را در رشته شبکه‌های کامپیوتری به ترتیب در سال‌های ۲۰۱۳ و ۲۰۱۷ از دانشگاه صنعتی شیراز دریافت کرده‌اند. زمینه‌های اصلی مورد علاقه ایشان شامل شبکه‌های نرم‌افزارمحور، الگوریتم‌های ابتکاری، امنیت SDN، شبکه‌های حسگر بی‌سیم زیر آب، رایانش مه، اینترنت اشیا، اینترنت اشیا صنعتی و اینترنت اشیا شهری است.

سکینه سهرابی دانشجوی مقطع دکتری مهندسی نرم‌افزار در دانشگاه بوعلی سینا از سال ۱۳۹۸، مدرک کارشناسی ارشد خود را در رشته مهندسی کامپیوتر از دانشگاه تربیت دبیر شهید رجایی در سال ۱۳۹۴ دریافت کرده‌اند. زمینه تحقیقاتی اصلی ایشان در مقطع دکتری، زمان‌بندی وظایف در محیط‌های محاسبات ابری با استفاده از الگوریتم‌های فراابتکاری است. زمینه‌های مورد علاقه پژوهشی ایشان شامل الگوریتم‌های فراابتکاری، رایانش مهو زمان‌بندی وظایف در محیط‌های محاسبات مه-ابر می‌باشد.

مهدی سخائی‌نیا استادیار گروه مهندسی کامپیوتر دانشگاه بوعلی سینا، مدرک کارشناسی خود را در رشته مهندسی کامپیوتر گرایش نرم‌افزار در سال ۱۳۷۵ از دانشگاه علم و صنعت ایران اخذ نمودند. ایشان تحصیلات تکمیلی خود را با دریافت مدرک کارشناسی ارشد و دکتری در رشته مهندسی کامپیوتر گرایش نرم‌افزار از دانشگاه علم و صنعت ایران، به ترتیب در سال‌های ۱۳۸۵ و ۱۳۹۴، به پایان رساندند. زمینه‌های تخصصی پژوهشی ایشان شامل تحلیل زمانی و ایستا برنامه‌ها، محاسبات توزیع‌شده، پردازش ابری و سیستم‌های بی‌درنگ است.