



دانشگاه کاشان
University of Kashan

مجله محاسبات نرم

SOFT COMPUTING JOURNAL

تارنمای مجله: scj.kashanu.ac.ir



روشی جدید جهت تخصیص منابع به جریان‌های کاری بی‌درنگ در محیط‌های مه‌مبتنی بر شبکه‌های نرم‌افزار محور*

محمد صادق زاده^۱، کارشناسی ارشد، رضا محمدی^{۱*}، استادیار، محمد نصیری^۱، دانشیار، سکینه سهرابی^۱، دانشجوی دکتری^۱ گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه بوعلی سینا، همدان، ایران.

چکیده

اطلاعات مقاله

تاریخچه مقاله:

دریافت ۲ دی ماه ۱۴۰۱

پذیرش ۲ مرداد ماه ۱۴۰۲

کلمات کلیدی:

اینترنت اشیا

محاسبات ابر

الگوریتم تخصیص منابع

الگوریتم ژنتیک

الگوریتم گرگ خاکستری

محاسبات مه به طور فزاینده‌ای به عنوان یک بستر برای پردازش برنامه‌های اینترنت اشیا استفاده می‌شود. در نتیجه، این معماری خدمات رایانش ابری را به لبه شبکه می‌آورد، جایی که پردازش ممکن است ارزان‌تر و سریع‌تر انجام شود. یکی از چالش‌های اساسی در زمینه تامین کیفیت خدمات سطح سرویس مانند تاخیر و مصرف انرژی در محیط مه توجه به محدودیت انرژی و ظرفیت کم سرورهای مه می‌باشد که تصمیم‌گیری اختصاص وظایف به گره‌های مه را دشوار می‌کند. این مقاله ابتدا یک مدل ریاضی برای تخصیص منابع ارائه می‌دهد که هدف آن به حداقل رساندن تاخیر و انرژی در شرایطی است که معیارهای کیفیت خدمات سطح سرویس در نظر گرفته می‌شود. سپس به ارائه الگوریتم ترکیبی ژنتیک و گرگ خاکستری جهت حل مدل ریاضی پرداخته می‌شود. باید توجه داشت که ترکیب این دو الگوریتم باعث ارائه جواب‌های متنوع و در نهایت بهینه می‌شود. لازم به ذکر است که اجرای الگوریتم‌های مذکور دارای هزینه پردازشی و تاخیر محاسباتی می‌باشد ولی با توجه به بهبود معیارهای کیفیت سطح سرویس می‌توان از این هزینه چشم‌پوشی کرد. نتایج حاصل شده حاکی از آن است که ترکیب و استفاده همزمان از نقاط مثبت دو الگوریتم، معیارهای زمان اجرا و زمان اتمام آخرین کار و همچنین مصرف انرژی را به ترتیب به میزان ۱۸/۳۰٪ و ۱۵/۱۴٪ و ۱۰/۲۱٪ به نسبت روش نیمه حریصانه بهبود می‌دهد.

© ۱۴۰۲ نویسندگان. مقاله با دسترسی آزاد تحت مجوز CC-BY

۱. مقدمه

توسعه و گسترش دستگاه‌های اینترنت اشیا (IoT) باعث پدیدار شدن بسیاری از برنامه‌های اینترنت اشیا شده است [۱]. اینترنت

اشیا شبکه‌ای با طیف گسترده از وسایل هوشمند است که در آن اشیای فیزیکی قادر به تبادل اطلاعات با سایر اشیای هستند. سرویس‌های متنوع اینترنت اشیا منجر شده است که زندگی روزمره بشری به فعالیت‌های مطمئن و قابل دسترس این شبکه وابسته شود [۲]. از سوی دیگر امروزه تعداد برنامه‌های کاربردی که به زمان پاسخ‌دهی کمی نیاز دارند، در حال افزایش است [۳]. بالا رفتن حجم تبادلات دستگاه‌های اینترنت اشیا باعث مصرف بالای پهنای باند، تاخیر زیاد، عدم زمان‌بندی مناسب و کاهش کارایی سیستم می‌شود. با توجه به رشد چشمگیر برنامه‌های

* نوع مقاله: پژوهشی

* نویسنده مسئول

پست(های) الکترونیک:

mohammad.sadeghzadeh@tejaratbank.ir (صادق زاده)

r.mohammadi@basu.ac.ir (محمدی)

m.nassiri@basu.ac.ir (نصیری)

ssohrabi69@yahoo.com (سهرابی)

نحوه ارجاع به مقاله: محمد صادق زاده، رضا محمدی، محمد نصیری، سکینه سهرابی، «روشی جدید جهت تخصیص منابع به جریان‌های کاری بی‌درنگ در محیط‌های مه‌مبتنی بر شبکه‌های نرم‌افزار محور»، مجله محاسبات نرم، جلد ۱۱، شماره ۲، صفحات ۹۰-۱۰۵، پاییز و زمستان ۱۴۰۱.

توجه به اطلاعات موجودیت‌های مختلف درگیر در محاسبه مه ارائه شده است. از آنجایی که مساله زمان‌بندی وظیفه، یک مساله NP سخت است، برای یافتن راه‌حل‌های بهینه به روش‌های ابتکاری و فراابتکاری نیاز دارد.

الگوریتم‌های بهینه‌سازی به دو دسته الگوریتم‌های دقیق و الگوریتم‌های تقریبی تقسیم‌بندی می‌شوند. الگوریتم‌های دقیق قادر به یافتن جواب بهینه به صورت دقیق هستند، اما کارایی کافی را در رابطه با مسائل بهینه‌سازی سخت ندارند. برخلاف الگوریتم‌های دقیق، الگوریتم‌های تقریبی قادر به یافتن جواب‌های خوب در زمان کوتاه برای مسائل بهینه‌سازی سخت هستند. الگوریتم‌های تقریبی نیز به سه دسته الگوریتم‌های ابتکاری، فراابتکاری و فوق‌ابتکاری بخش‌بندی می‌شوند. می‌توان گفت که هدف الگوریتم‌های ابتکاری یا هیوریستیک این است که در زمان معقول بتوانند جوابی منطقی تولید کنند. باید توجه داشت که این جواب لزوماً جواب بهینه مساله نیست، اما نزدیک به آن است. در واقع این مساله که بتوان جوابی نزدیک به بهینه را در مدت زمان کمتر پیدا کرد، نسبت به پیدا کردن جواب بهینه در مدت زمان بسیار طولانی، ترجیح داده شده است [۵]. دو مشکل اصلی الگوریتم‌های ابتکاری، همگرایی زودرس به نقاط بهینه محلی و گیر افتادن آنها در این نقاط است. الگوریتم‌های فراابتکاری برای حل این مشکلات الگوریتم‌های ابتکاری، ارائه شده‌اند. در واقع الگوریتم‌های فراابتکاری، یکی از انواع الگوریتم‌های بهینه‌سازی تقریبی هستند که دارای راهکارهای برون‌رفت از نقاط بهینه محلی هستند و در این مقاله مورد استفاده قرار گرفته‌اند.

از سوی دیگر، هر یک از دستگاه‌های اینترنت اشیا دارای توان محاسباتی و ذخیره‌سازی و شبکه‌ای محدودی هستند و افزایش درخواست‌های کاربر منجر به ازدحام شبکه، سربار محاسباتی، افزایش ارتباطات و مدت زمان طولانی ارائه خدمات می‌شود [۶]. از این‌رو ایده پیشنهادی در این پژوهش برای برنامه‌هایی که حساس به زمان هستند و کیفیت خدمات نیاز اکید آنها هست، مناسب است. نمونه‌ای از این کاربردها مراقبت‌های بهداشتی، اورژانس، بازی برخط و نظارت می‌باشد. عمده معیارهایی که

دستگاه‌های هوشمند و بالا رفتن حجم تبادلات دستگاه‌های اینترنت اشیا، دیگر معماری سنتی و متمرکز پاسخگوی تقاضای زیاد کاربران و اجرای مناسب برنامه‌های بی‌درنگ و حساس به تاخیر نمی‌باشد [۴]. برای رسیدگی به این مساله، محاسبات مه^۱ که در میان دستگاه‌های ابر و اینترنت اشیا قرار دارد، پیشنهاد شد. محاسبات مه با ویژگی‌هایی که دارد، شکل دیگری از خدمات را به کاربران ارائه کرده است و بخش عظیمی از صنعت فناوری اطلاعات را دگرگون کرده است. باید توجه داشت که در محیط محاسبات مه، دستگاه‌های اینترنت اشیا به دستگاه‌های مه متصل می‌شوند و به دلیل اینکه این دستگاه‌ها در نزدیکی کاربران قرار دارند منجر به کاهش تاخیر و در نتیجه بهبود کارایی سیستم می‌شوند. محاسبات مه یک مدل محاسباتی غیرمتمرکز است که می‌تواند به رایانش ابری برای ارائه خدمات با کیفیت بالا کمک کند.

دستگاه‌های مه در نزدیکی شبکه با محدودیت منابع مواجه هستند، بنابراین زمان‌بندی کار در یک گره مه^۲ بسیار حائز اهمیت می‌باشد. در واقع اگرچه محاسبات مه برای پردازش برنامه‌های اینترنت اشیا مناسب است، اما با توجه به محدودیت‌های گره‌های مه باید الگوریتم تخصیص وظایف با در نظر گرفتن معیارهای کیفیت خدمات سطح سرویس معین نماید که به هر گره چه میزان حجم کار تخصیص داده شود که بتوان میزان تاخیر و مصرف انرژی را کاهش داد. از سوی دیگر به دلیل آنکه منابع موجود در مه، در حال تغییر هستند، مساله تخصیص منابع، امر مهمی است و تاثیر زیادی در عملکرد محیط محاسبات مه دارد. الگوریتم تخصیص منابع، روشی است که به وسیله آن، وظایف به منابع موجود در سیستم تخصیص داده می‌شوند. باید توجه داشت که زمان‌بندی کار در یک گره مه بسیار حائز اهمیت است و با توجه به ماهیت پویا، ناهمگنی منابع و محدودیت منابع مساله زمان‌بندی وظایف در محیط به نحوی که نیازهای کیفیت خدمات را برآورده کنند، یک مساله NP سخت می‌باشد. برای غلبه بر این مسائل، استراتژی‌های مختلفی برای برنامه‌ریزی با

¹ Fog Computing

² Fog Node

الگوریتم متمرکز وجود یک گره مرکزی است که وظیفه بررسی وضعیت لحظه‌ای منابع گره‌های مه، اجرای الگوریتم مورد نظر و تخصیص وظایف را دارد. الگوریتم‌های متمرکز را به طور کلی به دو دسته الگوریتم‌های متمرکز و غیرهوشمند و الگوریتم‌های متمرکز و هوشمند طبقه‌بندی می‌کنند. لازم به ذکر است که در ادامه به بررسی برخی از روش‌های مرتبط در هر دسته پرداخته می‌شود.

۱.۲. الگوریتم‌های متمرکز و هوشمند

در این بخش به توضیح برخی از الگوریتم‌های متمرکز و هوشمند ارائه شده می‌پردازیم.

در مقاله [۶]، به پیاده‌سازی الگوریتم تخصیص منابع متمرکز به نام PSG پرداخته شده است. در این مقاله از الگوریتم‌های MIN-VIOL و SEMI-GREEDY استفاده شده است که تصمیم‌گیری بر اساس زمان پاسخ انجام می‌شود. هدف از این مقاله کاهش انرژی سیستم مه با توجه به مهلت انجام وظایف می‌باشد. در روش مذکور هنگامی که وظیفه مورد نظر زمان کافی برای پردازش را داشته باشد، الگوریتم SEMI-GREEDY اجرا می‌شود و هنگامی که زمان انجام کار از مهلت تعیین شده بیشتر می‌شود، الگوریتم MIN-VIOL اجرا شده و سعی در کاهش زمان تخطی می‌شود.

مقاله [۷] به اختصاص کارآمد وظایف برای برنامه‌های مبتنی بر اینترنت اشیا در محیط محاسبات مه پرداخته است. در این مقاله، از دو الگوریتم فراابتکاری بهینه‌سازی ازدحام ذرات و بهینه‌سازی کلونی مورچگان، استفاده شده است. در مقاله مذکور تمرکز اصلی بر روی تعادل وظایف بارگذاری شده اینترنت اشیا بر روی گره‌های مه می‌باشد و این امر با هدف رعایت معیارهای کیفیت سرویس انجام می‌شود. ارزیابی‌ها بیانگر آن است که زمان‌بندی پیشنهادی مبتنی بر کلونی مورچگان باعث بهبودی در زمان پاسخ برنامه‌های اینترنت اشیا در مقایسه با الگوریتم زمان‌بندی نوبت گردشی و الگوریتم مبتنی بر بهینه‌سازی ازدحام ذرات می‌شود. از سوی دیگر، با افزایش تعداد گره‌های دستگاه‌های اینترنت اشیا عملکرد الگوریتم کلونی مورچگان

برای مقایسه و بررسی عملکرد الگوریتم‌های زمان‌بندی مورد استفاده قرار می‌گیرند و در بررسی عملکرد یک سیستم اهمیت دارند، عبارتند از هزینه اجرا، زمان اجرا، مصرف انرژی، نرخ درخواست‌های رد شده، مدت زمان انجام کار^۱، تعادل بار^۲ در سیستم، سطح رضایت کاربر و سطح کیفیت خدمت^۳، که در این پژوهش سعی در بهبود معیارهای زمان اجرا، زمان اتمام آخرین کار و همچنین مصرف انرژی شده است.

در این مقاله با توجه به محدودیت ظرفیت کم سرورهای مه و انرژی و اهمیت تصمیم‌گیری اختصاص وظایف به گره‌های مه، یک مدل ریاضی برای تخصیص منابع با هدف کاهش انرژی و مدت زمان تاخیر ارائه شده است. در ادامه به ارائه الگوریتم ترکیبی ژنتیک و گرگ خاکستری جهت حل مدل ریاضی پرداخته شده است. ترکیب این دو الگوریتم منجر به تولید جواب‌های متنوع و در نهایت بهینه می‌شود. لازم به ذکر است که اجرای الگوریتم‌های پیشنهادی دارای تاخیر محاسباتی و هزینه پردازشی می‌باشد ولی با توجه به بهبود معیارهای کیفیت سطح سرویس می‌توان از این هزینه چشم‌پوشی کرد. نتایج به دست آمده بیانگر آن است که استفاده همزمان از نقاط مثبت دو الگوریتم و ترکیب آنها، معیارهای زمان اتمام آخرین کار و زمان اجرا و همچنین مصرف انرژی را بهبود می‌بخشد.

ادامه مطالب این مقاله به شرح زیر سازماندهی شده است. بخش ۲ مطالعات مربوط به تخصیص منابع در محاسبات مه مرور شده است. بخش ۳ مدل سیستم شامل معماری و فرمول مساله شرح داده شده است. الگوریتم‌های پیشنهادی در بخش ۴ بیان و تحلیل شده است. در نهایت، در بخش ۵ ارزیابی و نتایج تجربی ارائه شده مورد بحث و نتیجه‌گیری قرار گرفته است.

۲. پیشینه تحقیق

در سال‌های اخیر تحقیقات گسترده‌ای در زمینه زمان‌بندی وظایف اینترنت اشیا در محیط مه و لبه انجام شده است. تمرکز این مقاله بر روی الگوریتم‌های متمرکز می‌باشد. منظور از

^۱ Makespan

^۲ Load Balance

^۳ Quality of Service

نسبت به الگوریتم ازدحام ذرات در زمان پاسخگویی وظایف بارگذاری شده تا حد زیادی بهبود داشته است و این روش به طور موثر باعث تعادل بارکاری بر روی گره‌های مه می‌شود.

در مقاله [۸]، الگوریتم MBFD ارائه شده است. این الگوریتم یک روش بهینه مبتنی بر روش بهینه‌سازی با شبکه عصبی مصنوعی (ANN) و الگوریتم کلونی زنبورهای مصنوعی (ABC) می‌باشد. این تحقیق به منظور حل چالش‌های پیشروی محققان در زمینه زمان‌بندی کار در محاسبات مه ارائه شده است. نتایج آزمایش‌هایی که برای ارزیابی اثربخشی الگوریتم مذکور انجام شده است بیانگر این مطلب است که این الگوریتم با بهره‌گیری از الگوریتم‌های ANN و ABC در مقایسه با سایر روش‌های موجود بر اساس معیارهای مصرف انرژی و زمان اتمام کار بهتر عمل می‌کند. لازم به ذکر است که در این روش به معیارهایی مانند تعادل بار و اولویت پرداخته نشده است.

در مقاله [۹] مبحث ارائه خدمات در لایه مه با در نظر گرفتن چالش‌های عملیاتی موجود و توجه به معیارهای کیفیت خدمات، مورد بررسی قرار گرفته است. این روش به طور خاص، یک استراتژی مطلوب را برای تخصیص مناسب‌ترین منبع مه ارائه می‌دهد و خدمات مختلف درخواست‌های اینترنت اشیا را تجزیه و تحلیل می‌کند.

۲.۲. الگوریتم‌های متمرکز و غیرهوشمند

در این بخش به توضیح برخی از الگوریتم‌های متمرکز و غیرهوشمند ارائه شده می‌پردازیم.

در مقاله مسیریابی چندلایه‌ای در محیط‌های مه‌آلود با زیرساخت شبکه‌های نرم‌افزار محور^۱ [۱۰]، با در نظر گرفتن ناهمگنی گره‌ها از نظر ویژگی‌های نرم‌افزاری، سخت‌افزاری و خدمات برای برنامه‌های حساس به تاخیر مسیریابی انجام می‌شود و گره‌ها به صورت پویا به شبکه وارد و یا از شبکه خارج می‌شوند. مدل چندلایه‌ای شبکه‌ای، یک راه‌حل چندلایه مسیریابی که مبتنی بر شبکه‌های نرم‌افزار محور است را پیشنهاد می‌کند. هر دستگاهی که قابلیت محاسبه، ارتباط شبکه‌ای و ذخیره‌سازی را داشته باشند

می‌تواند به عنوان گره مه در نظر گرفته شود [۱۱].

در مقاله [۱۲]، الگوریتم MDAF با توجه به معیار کیفیت سرویس برای سیستم‌های محاسبات ابر-مه ارائه شده است که یک سازوکار زمان‌بندی و اختصاص خدمات می‌باشد. در روش مذکور تا حد ممکن در لایه‌ای که به کاربر نزدیک‌تر است وظایفی که به زمان حساس‌تر هستند، سرویس داده می‌شوند. نتایج شبیه‌سازی بیانگر آن است که الگوریتم فوق از نظر هزینه اجرا و تاخیر بهبود قابل توجهی داشته است، اما در این مقاله به معیار منابع و بهره‌وری انرژی پرداخته نشده است و از منابع به طور موثر استفاده نمی‌شود.

در مقاله [۱۳]، استراتژی تخصیص منابع پویا توضیح داده شده است. روش مذکور مبتنی بر شبکه‌های پتری دوره‌ای قیمت‌گذاری شده برای محاسبات مه است. ایده اصلی این کار به این صورت است که کاربر می‌تواند منابع رضایت‌بخش را به طور خودکار از یک گروه منابع از پیش تخصیص یافته، انتخاب کند. استراتژی پیشنهادی آنها زیرساخت‌های مه و اعتبار کاربران را برای انجام وظایف در نظر گرفته است. در این مقاله کاربری که دارای محدودیت اعتباری بالایی باشد، قادر است که منابع بسیار قابل اعتمادی را برای انجام وظایف خود بکار گیرد. باید توجه داشت که در این مقاله به معیارهایی همچون اولویت و تعادل بار پرداخته نشده است.

در مقاله [۱۴]، یک چارچوب کلی برای برنامه‌های کاربردی اینترنت اشیا و مه معرفی شده است و یک سیاست به منظور به حداقل رساندن تاخیر برای دستگاه‌های مه ارائه شده است. هدف روش مذکور کاهش تاخیر سرویس برای برنامه‌های اینترنت اشیا است. آنها یک مدل تحلیلی برای ارزیابی سیاست تخصیص و یک سازوکار برای کاهش تاخیر ارائه داده‌اند.

در مقاله [۱۵]، الگوریتم JointPT ارائه شده است. روش مذکور یک سیاست اختصاص خدمات و زمان‌بندی می‌باشد که با توجه به معیارهای توان، اولویت و ترافیک برای حل موثر مساله اختصاص ماشین مجازی‌های در یک مرکز داده ابر ارائه شده است. در روش مذکور از توپولوژی Fat-Tree استفاده می‌شود و بررسی نتایج بیانگر آن است که روش پیشنهادی می‌تواند

¹ Software-Defined Wide Area Networking

هدررفت منابع و مصرف انرژی را کاهش دهد. در این مقاله به معیارهایی همچون هزینه و تعادل بار پرداخته نشده است. در مقاله [۱۶]، الگوریتم LBP-ACS توضیح داده شده است. روش پیشنهادی ترکیبی از الگوریتم‌های ACS و LBPA است و به منظور حل مشکل زمان‌بندی کار در محیط محاسبات مه ارائه شده است. در الگوریتم مذکور برای زمان‌بندی کارها مهلت پایان یافتن و اولویت کارها در نظر گرفته شده است و به منظور رسیدگی به حساسیت تأخیر کار، برای زمان‌بندی و ساخت یک دنباله کار با اولویت معقول از الگوریتم Laxity-based priority استفاده شده است. همچنین از الگوریتم بهینه‌سازی مبتنی بر الگوریتم کلونی مورچگان به منظور به حداقل رساندن مصرف انرژی، بهره گرفته شده است. باید توجه داشت که نتایج حاصل از زمان‌بندی با استفاده از روش مذکور برای کارهای وابسته به

هم دارای جواب بهینه نیست. در جدول (۱) روش مورد استفاده، ابزار شبیه‌سازی، معیارهای کارایی، مجموعه داده و ویژگی‌های مربوط به الگوریتم‌های بررسی شده در این پژوهش به اختصار آورده شده است.

۳. مدل‌سازی و فرمول‌سازی مساله

این بخش شامل دو زیربخش است که به صورت کلی به توضیح مدل سیستم می‌پردازد. در بخش نخست، معماری کلی اینترنت اشیا-مه-ابر ارائه شده است و در بخش دوم، مساله زمان‌بندی کار به صورت ریاضی فرموله شده است. مدل پیشنهادی این مقاله ترکیبی از مدل‌های ارائه شده در مقالات [۶]، [۷]، [۱۷] و [۱۶] می‌باشد.

جدول (۱): ویژگی‌های مربوط به الگوریتم‌های بررسی شده

مقاله	هوشمندی	مزایا	معایب	روش مورد استفاده	ابزار شبیه‌سازی	مجموعه داده
[۶]	هوشمند	کاهش مصرف انرژی و به حداقل رساندن زمان نقض مهلت انجام کار	عدم جستجوی کامل فضای راه‌حل‌های مساله و قرار گرفتن در بهینه‌های محلی	SEMI-GREEDY (PSG Algorithm)	C++ programming language	فاقد
[۷]	هوشمند	زمان پاسخ و تعادل بار	در نظر نگرفتن معیارهای انرژی و زمان اتمام آخرین وظیفه	ACO and PSO Algorithm	Matlab	فاقد
[۹]	هوشمند	معیارهای زمان اتمام کار و مصرف انرژی	در نظر نگرفتن معیارهایی همچون اولویت و تعادل بار	ABC and ANN Algorithm	Matlab	فاقد
[۱۰]	هوشمند	تأخیر سرویس و پایداری انجام کار به طور مؤثر	عدم بررسی کامل معیارهای کیفیت خدمات سطح سرویس	NSGA-II Algorithm	Matlab	WordCount data set
[۱۱]	هوشمند	تعادل بار، تأخیر و مدیریت ترافیک	سربار پردازشی شبکه‌های نرم‌افزار محور	WOA algorithm	iFogSim	فاقد
[۱۲]	غیر هوشمند	تأخیر و هزینه اجرا	عدم توجه به معیارهای بهره‌وری انرژی و منابع	MDAF algorithm	iFogSim	فاقد
[۱۳]	غیر هوشمند	تعادل بار و تأخیر	در نظر نگرفتن معیار انرژی	Petri nets	MANET	فاقد
[۱۴]	غیر هوشمند	کمینه کردن تأخیر	گرسنگی وظایف با حجم بالا	linear programming	Matlab	فاقد
[۱۵]	غیر هوشمند	اولویت، توان و ترافیک	در نظر نگرفتن معیارهایی تعادل بار و هزینه	JointPT Algorithm	Java programming language	فاقد
[۱۶]	غیر هوشمند	اولویت و کاهش مهلت پایان یافتن وظیفه	عدم کارایی الگوریتم در وظایف وابسته به هم	laxity and ant colony system algorithm (LBP-ACS)	Cloudsim	فاقد

جدول (۲): معرفی متغیرهای پرکاربرد در روابط مدل سازی مساله

معنی	نماد
دستگاه اینترنت اشیا	s_i
زمان تخطی از مهلت زمان مقرر شده برای انجام وظیفه t_i	v_i
سرورهای مه	f_j
مدت زمان اجرای وظایف در گره مه j	A_j
کارگزار مه	f_b
مدت زمان اجرای روشن بودن دستگاه اینترنت اشیا i	A_i
سرورهای ابر	C_k
پهنای باند پیوند ارتباطی دستگاه اینترنت اشیا تا کارگزار مه	BW_s
مجموعه وظایف	t_i
پهنای باند پیوند ارتباطی کارگزار مه تا کارگزار ابر	BW_c
کارگزار ابر	c_b
میزان نیاز پردازشی وظیفه i	t_i^m
میزان مصرف انرژی در حالت فعال	F_j^{act}
میزان زمان مقرر شده برای انجام وظیفه i	t_i^d
مصرف برق در حالت بیکار	F_j^{idl}
اندازه وظیفه i	t_i^s
نرخ وظایف ارسالی کارگزار مه	λ_{f_b}
مجموع انرژی مصرفی	J_E
نرخ سرویس دهی گره مه	μ_f
تاخیر کل سیستم	J_D
نرخ وظایف دریافتی گره مه	$\lambda_{s_{total}}$
مصرف انرژی زمان بارگذاری وظایف از مه به ابر	$E_i^{offloading}$

در مدل پیشنهادی تحقیق جاری، مجموعه وظایف مستقل به صورت $T = \{t_1, t_2, \dots, t_n\}$ در نظر گرفته می شود که این وظایف از دستگاه های اینترنت اشیا $S = \{s_1, s_2, \dots, s_w\}$ به مجموعه ای از گره های مه $F = \{f_1, f_2, \dots, f_m\}$ تخصیص داده می شوند. این تخصیص توسط f_b انجام می پذیرد. همچنین گره های ابر به صورت $C = \{c_1, c_2, \dots, c_n\}$ نمایش داده می شوند.

مساله اصلی تخصیص این وظایف به گره های مه می باشد و نگاشت $\Psi: T \rightarrow F$ باید طوری تعریف شود که مصرف انرژی و تاخیر کل سیستم کمینه شود. بنابراین این مساله با استفاده از رابطه (۱) مدل می شود.

$$\min \quad \alpha \cdot J_{E+(1-\alpha)} \cdot J_D \quad (1)$$

۱.۳. مدل سیستم

معماری سیستم از سه بخش دستگاه های اینترنت اشیا، محیط مه و محیط ابری تشکیل شده است که در ادامه شرح داده خواهند شد.

دستگاه های اینترنت اشیا: این بخش شامل دستگاه های اینترنت اشیا می باشد که با نماد s (که در واقع بیانگر تعداد حسگر می باشد)، نشان داده می شود. لازم به ذکر است که در شبیه سازی انجام شده از حسگر دما استفاده شده است.

محیط مه: یک محیط مه شامل یک کارگزار مه f_b و تعدادی سرور مه f_j است که با کارگزار مه ارتباط دارند. کارگزار مه جزء مرکزی یک ارائه دهنده خدمات مه است که وظیفه مدیریت سرورهای مه را برعهده دارد و وظایف را بر اساس الگوریتم تخصیص منابع که در بخش بعد شرح داده خواهد شد، تخصیص می دهد.

محیط ابری: در این بخش، گره ابر به صورت C_k نشان داده می شود که شامل سرورهایی است که از مجموعه ای از ماشین های مجازی^۱ با قدرت محاسباتی و ظرفیت ذخیره سازی بالا تشکیل شده است.

۲.۳. فرموله سازی مساله

در این بخش، یک مدل رسمی برای مساله زمان بندی کار در یک سیستم محاسباتی مه ارائه شده است. ابتدا عناصر اساسی و متغیرهای تصمیم گیری مساله بهینه سازی معرفی می شوند. سپس، مدل های زمان پاسخ و انرژی تشریح می شوند. در نهایت، بررسی اجمالی مشکل ارائه شده است. در جدول (۲) متغیرها و نمادهای پرکاربرد در روابط مدل سازی مساله آورده شده است.

۱.۲.۳. متغیرهای اساسی و مدل بار کاری

درستی شکل های گره های مه و وظایف تولیدی توسط اینترنت اشیا از متغیرهای اساسی به شمار می روند که در ادامه به شرح نحوه فرموله کردن آنها پرداخته می شود.

¹ Virtual Machine

در رابطه (۳) برای هر وظیفه t_i ، متغیر دودویی x_{ij} تعریف شده و با استفاده از رابطه (۴) مقدار این متغیر مشخص می‌شود. با استفاده از متغیر e_{ij} انتخاب یا عدم انتخاب پیوند برای مسیریابی وظیفه t_i مشخص می‌شود.

$$y_{ij} = \begin{cases} 1 & \text{if } e_{ij} \text{ assigned to } t_i \\ 0 & \text{Otherwise} \end{cases} \quad (4)$$

در رابطه (۴)، متغیر e_{ij} بیانگر آن است که از کدام مسیر خاص برای رسیدن وظیفه به کارگزار مه و سپس کارگزار ابر استفاده شده است، زیرا مسیرهای مختلفی بین کارگزار ابر و کارگزار مه وجود دارد. در پایان از یک متغیر دودویی به نام s_{t_i} استفاده می‌شود که مهلت انجام وظیفه t_i را مشخص می‌کند و با استفاده از رابطه (۵) مقداردهی می‌شود.

$$s_{t_i} = \begin{cases} 1 & \text{if } J_D \leq \tau_i^d \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

در رابطه (۵)، متغیر s_{t_i} بیانگر آن است که وظیفه t_i در مهلت زمان مقرر شده (J_D) پردازش شده است یا خیر که اگر میزان زمان پردازشی و تاخیر شبکه‌ای کمتر از مهلت زمان مقرر شده برای انجام یک وظیفه باشد مقدار آن برابر یک و در غیر این صورت مقدار آن برابر صفر می‌باشد.

۳.۲.۳. مدل تاخیر

در این بخش، ابتدا تاخیر دستگاه اینترنت اشیا تا کارگزار مه مورد بررسی قرار گرفته و سپس برای وظایفی که به کارگزار مه منتقل می‌شوند، تاخیر انتقال آنها از دستگاه اینترنت اشیا به کارگزار مه محاسبه می‌شود. متغیر $L_{iot-fb(i)}$ بیانگر این مقدار می‌باشد و با استفاده از رابطه زیر محاسبه می‌شود.

$$L_{iot-fb(i)} = Prp_{Delay}^{iot-fb} + \frac{t_i^s}{BW_s} \quad (6)$$

در این رابطه، متغیر Prp_{Delay}^{iot-fb} بیانگر تاخیر انتشار پیوند ارتباطی بین دستگاه اینترنت اشیا و کارگزار مه می‌باشد. تاخیر پردازشی وظیفه نام در گره مه با توجه به فرکانس پردازشی و به صورت $\delta_i^{fog} = \frac{t_i^m}{f_j^{mips}}$ محاسبه می‌شود.

باید توجه داشت که در هنگامی که پس از اتمام وظیفه کارگزار

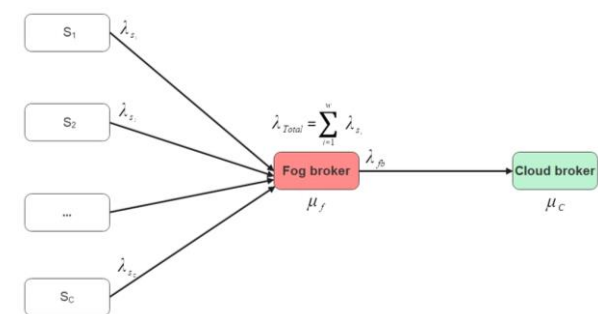
در این سناریو وظایف اینترنت اشیا مستقل از هم و مقدار متغیرهای مورد استفاده برای کارگزار مه (f_b) قابل دسترسی فرض شده است. لازم به ذکر است که در این پژوهش از تاخیر انتشار چشم پوشی شده است.

همان‌طور که گفته شد در این پژوهش T بیانگر مجموعه n وظیفه مستقلی است که از دستگاه‌های اینترنت اشیا در یک دوره زمانی خاص به کارگزار مه بارگذاری می‌شود. هر وظیفه $t_i \in T$ توسط سه مولفه آورده شده در رابطه (۲) تعریف می‌شود.

$$t_i = \langle t_i^m, t_i^d, t_i^s \rangle \quad (2)$$

در این رابطه، متغیرهای t_i^m ، t_i^d و t_i^s به ترتیب نشان‌دهنده تعداد دستورالعمل، مهلت زمان انجام یک وظیفه (ثانیه) و اندازه وظیفه (کیلوبایت) می‌باشند [۷].

مدل بار کاری سیستم در شکل (۱) نشان داده شده است که در ادامه به تشریح آن پرداخته می‌شود. با توجه به این شکل هر دستگاه اینترنت اشیا (s_i) با نرخ λ_{s_i} وظایف را تولید می‌کند و در کارگزار مه نرخ ورود بار کاری، مجموع تمامی نرخ‌های دستگاه‌های اینترنت اشیا می‌باشد. سپس کارگزار مه وظایف را پس از پیش‌پردازش با نرخ (λ_{fb}) به سمت کارگزار ابر ارسال می‌کند.



شکل (۱): فرآیند اجرای وظیفه در روش پیشنهادی

۲.۲.۳. متغیرهای تصمیم [۶]

در این مقاله، ماتریس X به عنوان ماتریس تخصیص وظایف در نظر گرفته شده است و درایه‌های ماتریس مذکور با استفاده از رابطه (۳) که در ادامه آورده شده است، تعریف می‌شوند.

$$x_{ij} = \begin{cases} 1 & \text{if } t_i \text{ assigned to } f_j \\ 0 & \text{Otherwise} \end{cases} \quad \forall i \in T, j \in F \quad (3)$$

در رابطه (۱۰)، متغیر δ_i^{fog} بیانگر زمان پردازش وظیفه t_i در گره f_j می باشد و متغیر x_{ik} بیانگر متغیر تصمیم است که مقدار آن صفر یا یک است. مقدار این متغیر در زمانی که وظیفه در گره مه بارگذاری می شود، برابر یک است و در غیر این صورت صفر می باشد. برای به دست آوردن زمان ایستا هر f_j ، ابتدا نیاز داریم که زمان اتمام آخرین وظیفه^۲ یک f_j را در بین همه f_j ها به دست آوریم که ابتدا طبق رابطه (۱۱) زمان اتمام آخرین وظیفه را محاسبه می کنیم [۱۸].

$$M_{j(t)} = \max_{j \in F} A_j \quad (11)$$

در رابطه (۱۱)، متغیر $M_{j(t)}$ بیانگر زمان اتمام آخرین وظیفه i در گره f_j در بازه زمانی t می باشد. برای محاسبه زمان در حالت ایستا گره f_j از رابطه (۱۲) استفاده می شود.

$$l_{j(t)} = (M_{j(t-1)} - A_j) \quad \forall j \in F \quad (12)$$

بنابراین مصرف انرژی کلی f_j با استفاده از رابطه (۱۳) محاسبه می شود [۶].

$$E_j = \sum_{j=0}^m A_j \times F_j^{act} + l_{j(t)} \times F_j^{idl} \quad (13)$$

در رابطه (۱۳)، متغیر F_j^{act} و F_j^{idl} به ترتیب بیانگر توان مصرفی (W) در حالت پویا و میزان توان مصرفی (W) در حالت ایستا می باشد. انرژی مصرفی در اینترنت اشیا با توجه به اینکه داده ها پیوسته در حال تولید می باشند، با استفاده از رابطه (۱۴) محاسبه می شود.

$$E_i = \sum_{i=0}^n A_i \times S_i^{act} \quad \forall i \in S \quad (14)$$

با توجه به رابطه (۱۴)، متغیر A_i بیانگر مدت زمان روشن بودن دستگاه اینترنت اشیا و متغیر S_i^{act} بیانگر توان مصرفی می باشد. زمانی که کارگزار گره مه تصمیم به بارگذاری وظیفه به کارگزار ابر می گیرد، انرژی مصرف شده برای این کار با استفاده از رابطه (۱۵) محاسبه می شود.

$$E_j^{offloading} = \sum_{i=0}^T P_{fb-cb} \times \frac{t_i^s}{BW_c} \quad (15)$$

مه تصمیم به بارگیری وظایف به سمت ابر می گیرد، میزان تاخیر با کمک رابطه (۷) محاسبه می شود.

$$L_{fb-cb(i)} = Prp_{Delay}^{fb-cb} + \frac{t_i^s}{BW_c} \quad (7)$$

لازم به ذکر است که متغیرهای Prp_{Delay}^{fb-cb} و t_i^s ، BW_c به ترتیب بیانگر پهنای باند پیوند ارتباطی بین کارگزار مه و کارگزار ابر، اندازه وظیفه و تاخیر انتشار پیوند بین کارگزار ابر و کارگزار مه می باشد. از سوی دیگر برای محاسبه تاخیر پردازشی گره ابر از رابطه $\delta_i^{cloud} = \frac{t_i^m}{c_k^{mips}}$ استفاده می شود.

در این پژوهش از تاخیر انتشار چشم پوشی شده است. در روش پیشنهادی تاخیر به صورت رابطه (۸) قابل محاسبه است.

$$J_D = \sum_{i=1}^n L_{iot-fb(i)} + \delta_i^{fog} + L_{fb-cb(i)} + \delta_i^{cloud} \quad (8)$$

برای هر وظیفه t_i میزان زمان نقض مهلت انجام^۱ آن را می توان با استفاده از رابطه (۹) محاسبه کرد.

$$v_i = \max(0, J_D - t_i^d) \quad \forall i \in T \quad (9)$$

۴.۲.۳. مدل مصرف انرژی

در این قسمت، مدلی برای انرژی مصرف شده برای سرورهای پردازشی ارائه می شود. لازم به ذکر است که در این پژوهش انرژی مصرفی هر گره مه در دو حالت ایستا و پویا بررسی شده است. حالت ایستا زمانی است که گره در حالت idle بوده و وظیفه ای به آن تخصیص نیافته است. حالت پویا زمانی است که وظیفه به گره تخصیص داده شده است و گره در حال پردازش وظیفه مورد نظر می باشد.

انرژی مصرف شده توسط یک گره $f_j \in F$ از مجموع انرژی مصرف شده در حالت ایستا و حالت پویا به دست می آید. حالت پویا f_j برابر است با زمان مورد نیاز برای پردازش تمام وظایف محول شده به گره مه که با رابطه (۱۰) قابل محاسبه است.

$$A_j = \sum_{i \in T} \delta_i^{fog} \times x_{ij} \quad \forall j \in F \quad (10)$$

² Makepan

¹ Violation time

$$\sum_{i=1}^n t_i^m \leq \sum_{k=1}^h c_j^{mips} \quad (20)$$

$$\sum_{i=1}^n B_i \leq BW_s \quad (21)$$

$$\sum_{i=1}^n B_i \leq BW_c \quad (22)$$

روابط (۱۹) و (۲۰) به ترتیب محدودیت‌های توانایی پردازشی گره‌های مه و گره‌های ابری را مشخص می‌کنند [۶]. همچنین روابط (۲۱) و (۲۲) بیانگر محدودیت مربوط به شبکه هستند و بیان می‌کند که نیازمندی‌های شبکه‌ای وظیفه t_i نمی‌تواند از پهنای باند پیوند ارتباطی مه و ابر بیشتر باشد.

۴. الگوریتم پیشنهادی

در این بخش، الگوریتم ترکیبی فراابتکاری پیشنهادی که جهت تخصیص منابع به جریان‌های کاری بی‌درنگ در محیط‌های مه مبتنی بر شبکه‌های نرم‌افزار محور می‌باشد، ارائه شده است.

با مطرح شدن مفهوم شبکه نرم‌افزار محور (SDN)، در صنعت شبکه رویکرد جدیدی ارائه شد که به کمک آن نیازهای نسل جدید سیستم‌های مبتنی بر شبکه نیز پوشش داده می‌شود. باید توجه داشت که سطح کنترل شبکه که پیش از این روی هر دستگاه شبکه بود، در رویکرد SDN جدا شده و کلیه تنظیمات به صورت متمرکز پس از پیکربندی، در شبکه انجام می‌شود. در واقع این رویکرد مدیران شبکه را قادر می‌نماید تا خدمات شبکه از طریق انتزاع سطح بالاتر را مدیریت کنند.

الگوریتم پیشنهادی وظیفه تخصیص وظایف به سلول‌های مه را به عهده دارد به نحوی که شاخص‌های عملکردی شبکه در این مقاله بهبود پیدا کرده و در وضعیت بهتری قرار گیرند. بر همین اساس با در نظر گرفتن مطالب ذکر شده، در روش پیشنهادی تمرکز بر استفاده از الگوریتم‌های فرااکتشافی ترکیبی گرگ خاکستری و ژنتیک [۱۹] می‌باشد که برای تخصیص بهینه منابع استفاده می‌شود. الگوریتم ژنتیک دارای قابلیت جستجوی چند جانبه و تولید راه‌حل‌های جدید با استفاده از فرآیند تکامل است و به دلیل سادگی بسیار موثر و سریع عمل می‌کند که در روش

با توجه به رابطه (۱۵)، متغیر Pfb^{cb} نشان‌دهنده توان مصرفی کارگزار مه در بارگذاری وظیفه i به سمت کارگزار ابر و متغیر $\frac{t_i^s}{BW_c}$ بیانگر مدت زمان لازم جهت بارگذاری وظیفه از کارگزار مه به کارگزار ابر می‌باشد. تابع هزینه برای مصرف انرژی با استفاده از رابطه (۱۶) محاسبه می‌شود.

$$J_E = E_j^{offloading} + E_i + E_j + E_h \quad (16)$$

در این رابطه، متغیر $E_j^{offloading}$ بیانگر انرژی مصرف شده برای بارگذاری وظایف از کارگزار مه به سمت کارگزار ابر است. متغیرهای E_i و E_j به ترتیب بیانگر انرژی مصرفی دستگاه اینترنت اشیا و انرژی مصرفی گره مه می‌باشند. همچنین مقدار متغیر E_h برابر میزان انرژی مصرفی سرور ابر می‌باشد که با استفاده از رابطه (۱۴) به دست می‌آید.

با توجه به تعریف و فرموله‌سازی شاخص‌های استفاده شده در تابع برازندگی، انتظار بر این است که در روش پیشنهادی زمان پاسخ و انرژی مصرفی بهینه گردد.

۵.۲.۳. محدودیت‌ها

در مبحث تخصیص منابع با توجه به ماهیت محدود ظرفیت گره‌ها و پیوندهای ارتباطی محدودیت‌ها با استفاده از روابط (۱۷) و (۱۸) تعریف می‌شود.

$$\sum_{i=0}^n \sum_{j=0}^m X_{ij} = 1 \quad (17)$$

$$\sum_{i=0}^n \sum_{k=0}^h X_{ik} = 1 \quad (18)$$

رابطه (۱۷) بیانگر محدودیت مربوط به هر وظیفه است که بیان می‌کند هر وظیفه فقط باید به یک گره مه اختصاص داده شود. همچنین رابطه (۱۸) بیانگر محدودیت دیگری است که بیان می‌کند که هر وظیفه می‌تواند دقیقاً به یک گره ابر اختصاص داده شود. در بحث‌های پردازشی و پهنای باند نیز روابط (۱۹) تا (۲۲) تعریف می‌شود [۶].

$$\sum_{i=1}^n t_i^m \leq \sum_{j=1}^m f_j^{mips} \quad (19)$$

در فضای جستجو می باشد. خود کروموزوم ها از تعداد ثابتی ژن تشکیل می شوند، از این رو می توان هر کروموزوم را به صورت یک آرایه که در شکل (۲) نشان داده شده است، مدل کرد.

f_1	f_2	f_3	...	f_j
-------	-------	-------	-----	-------

شکل (۲): مدل سازی کروموزوم در روش پیشنهادی

همان طور که در شکل (۲) آورده شده است، باید توجه داشت که اندیس های یک کروموزوم بیانگر شماره وظایف وارد شده به سیستم می باشد و متغیرهای f_1 تا f_j بیانگر مجموعه گره های مه می باشند که با استفاده از روش پیشنهادی وظایف به گره های مه نگاشت می شوند. جمعیت اولیه در روش پیشنهادی مجموعه ای از راه حل های تصادفی (کروموزوم ها) است که برای تخصیص بهینه منابع تولید شده اند و در ابتدا به صورت تصادفی مقداری می شوند و سپس در مرحله های الگوریتم به سمت راه حل بهینه همگرا می شوند.

به منظور اینکه راه حل ها در فرآیند فراابتکاری شرکت داده شوند و در نتیجه به سمت جواب بهینه که همان راه حل مطلوب برای تخصیص وظایف می باشد؛ همگرا شوند، بایستی برازندگی این راه حل ها محاسبه گردد. در الگوریتم های فرااکتشافی به ازای هر یک از اعضای جمعیت اولیه تابع برازندگی محاسبه و ذخیره می شود. در روش پیشنهادی نیز به ازای هر کروموزوم تابع برازندگی محاسبه خواهد شد. بر همین اساس، تابع پیشنهادی برای محاسبه برازندگی کروموزوم ها در روش پیشنهادی ترکیبی از شاخص های یاد شده خواهد بود. این تابع طبق رابطه (۲۳) مدل سازی می گردد.

$$fitness(chromosome) = \alpha . J_E + (1 - \alpha) . J_D \quad (23)$$

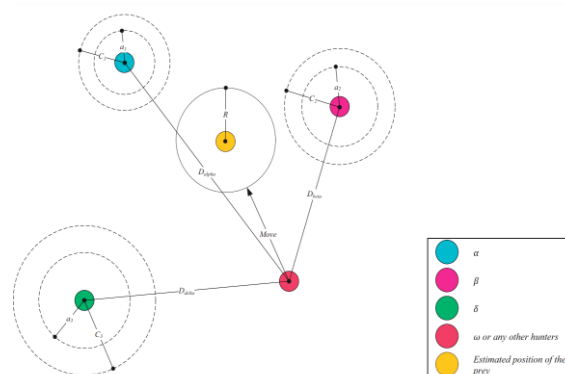
در گام بعد، عملگر ترکیب الگوریتم بهینه سازی ژنتیک اجرا می شود. از فرآیند ترکیب کروموزوم ها برای تولید راه حل های جدید استفاده می شود و برای این منظور از عملگر به روزرسانی الگوریتم بهینه سازی گرگ خاکستری بهره گرفته شده است. این کار منجر به افزایش کارایی الگوریتم بهینه سازی ژنتیک در تولید راه حل های جدید (کروموزوم ها) برای تخصیص بهینه وظایف و همگرایی مطلوبی راه حل ها به راه حل بهینه می شود. شکل (۳)

پیشنهادی از این الگوریتم بهره گرفته شده است. از سوی دیگر الگوریتم بهینه سازی گرگ خاکستری GWO [۲۰]، یک الگوریتم فراابتکاری است که از رفتار اجتماعی گرگ های خاکستری در هنگام شکار کردن الهام گرفته است و دارای ساختار سلسله مراتبی می باشد. این الگوریتم مبتنی بر جمعیت بوده، به سادگی قابلیت تعمیم به مسائل با ابعاد بزرگ را دارد و دارای فرآیند ساده ای است. در الگوریتم گرگ خاکستری دو فرآیند استخراج و اکتشاف بسیار مهم می باشد که برقراری تعادل میان این دو فاکتور دارای اهمیت می باشد. منظور از اکتشاف سراسری بودن جستجو است و این بدین معنا است که نیازمند جستجوی بیشتر می باشد. منظور از استخراج یا بهره برداری به دست آوردن پاسخ های بهتر حول یک پاسخ می باشد. اکتشاف پاسخ های جدیدی را ارائه می دهد زیرا در اکتشاف به دنبال جستجوی سراسری هستیم. از سوی دیگر فرآیند استخراج به دنبال تغییرات کم و مهم است که روی پاسخ فعلی تغییراتی را اعمال می کند. در این روش یک راه حل در نظر گرفته می شود و در طول اجرا تغییرات روی همان راه حل اعمال می شود. باید توجه داشت که فرآیند اکتشاف منجر به تولید راه حل جایگزین می شود تا قادر به واکنش انعطاف پذیرتر نسبت به تغییرات مورد نیاز محیط باشد. لازم به ذکر است که در الگوریتم گرگ خاکستری زمانی که $|A| \geq 1$ است الگوریتم در مرحله اکتشاف می باشد و زمانی که $|A| \leq 1$ است الگوریتم وارد مرحله استخراج می شود.

در این مقاله، پیشنهاد می شود که از عملگر به روزرسانی الگوریتم بهینه سازی گرگ خاکستری به جای عملگر ترکیب پیش فرض در الگوریتم بهینه سازی ژنتیک استفاده شود. بر اساس این عملگر، راه حل های جدید برای مساله به تبعیت از سه کروموزوم برتر که بالاترین میزان برازندگی را دارا می باشند و به راه حل بهینه نزدیک هستند؛ ایجاد می شوند. در ادامه به تشریح گام های مختلف الگوریتم پیشنهادی پرداخته می شود.

در روش پیشنهادی هر کروموزوم به عنوان یک راه حل بالقوه برای تخصیص وظایف به سلول های مه در نظر گرفته می شود؛ در الگوریتم ترکیبی گرگ خاکستری و ژنتیک، هر کروموزوم نشان دهنده یک راه حل ممکن برای مساله مورد نظر و یک نقطه

فرآیند به‌روزرسانی الگوریتم بهینه‌سازی گرگ خاکستری را نشان می‌دهد.



شکل (۳): به‌روزرسانی موقعیت‌ها در الگوریتم گرگ خاکستری

در روش پیشنهادی برای تولید فرزندان جدید از الگوریتم گرگ خاکستری استفاده خواهد شد. به این منظور پس از آنکه جمعیت مورد نظر برای مرحله ترکیب انتخاب شد، ابتدا سه کروموزومی که بهترین میزان برازندگی را دارند، از بین جمعیت اولیه الگوریتم بهینه‌سازی ژنتیک انتخاب و به عنوان گرگ‌های آلفا، بتا و دلتا ذخیره می‌شوند. سپس مابقی کروموزوم‌ها به استخر جفت منتقل می‌شوند. در ادامه عملگر ترکیب بر روی کروموزوم‌هایی که به عنوان گرگ امگا در نظر گرفته شده‌اند، اعمال می‌شود. در این مرحله برای تولید فرزندان سه کروموزومی که در جمعیت اولیه الگوریتم بهینه‌سازی ژنتیک بهترین جواب را دارند، با راه‌حل‌های موجود در استخر جفت‌گیری، به‌روزرسانی شده و بدین صورت فرزندان جدید طبق الگوریتم گرگ خاکستری ایجاد می‌شوند. رابطه (۲۴) این فرآیند را مدل‌سازی کرده است.

$$\begin{aligned} \vec{D}_\alpha &= \text{random}[1, \text{hd}(X_\alpha, X)], \\ \vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha) \\ \vec{D}_\beta &= \text{random}[1, \text{hd}(X_\beta, X)], \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta) \\ \vec{D}_\delta &= \text{random}[1, \text{hd}(X_\delta, X)], \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta) \end{aligned} \quad (24)$$

برای محاسبه موقعیت جدید از رابطه (۲۵) استفاده می‌شود. در این رابطه، متغیر $\vec{X}(t+1)$ بیانگر راه‌حل جدید و در واقع موقعیت جدیدی است که با استفاده از موقعیت گرگ‌های آلفا و بتا و دلتا به دست می‌آیند. بردارهای ضرایب الگوریتم فراابتکاری

در رابطه (۲۶) با استفاده از متغیرهای A و C نشان داده شده‌اند. دقت داشته باشید که مقدار A به صورت خطی از ۲ تا ۰ کاهش می‌یابد. همچنین متغیرهای r_1 و r_2 نشان‌دهنده یک بردار تصادفی با مقداری بین صفر و یک می‌باشند [۲۰].

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (25)$$

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r}_1 - \vec{a} \quad (26)$$

$$\vec{C} = 2 \cdot \vec{r}_2$$

با توجه به ماهیت گسسته مساله، متغیر $\vec{D}_\alpha$ یک مقدار تصادفی می‌باشد که مقدار آن بین یک تا فاصله همینگ هر یک از کروموزوم‌های گرگ‌های برتر و کروموزوم‌های استخر جفت‌گیری است. پس از آنکه ترکیبی از عملگر گرگ خاکستری و عملگر جهش الگوریتم ژنتیک روی جمعیت اولیه کروموزوم‌ها اعمال گردید، تعدادی کروموزوم جدید به جمعیت اولیه که توسط الگوریتم گرگ خاکستری ایجاد شده است، اضافه می‌گردد.

در الگوریتم (۱) شبه کد روش پیشنهادی آورده شده است که نحوه ترکیب الگوریتم گرگ خاکستری و ژنتیک را شرح می‌دهد. در ابتدا جمعیت اولیه به صورت تصادفی ایجاد می‌شود و سپس میزان برازندگی هر راه‌حل (که به عنوان یک گرگ برای الگوریتم گرگ خاکستری در نظر گرفته می‌شود) مشخص می‌شود. در ادامه سه گرگ (راه‌حل) برتر به عنوان گرگ‌های آلفا، بتا و دلتا انتخاب شده و مابقی گرگ‌ها (راه‌حل‌ها) به عنوان گرگ امگا در نظر گرفته می‌شوند. در ادامه به کمک الگوریتم ژنتیک سعی در بهبود گرگ‌های امگا می‌شود. به این صورت که به ازای هر یک از گرگ‌های امگا (از چهارمین بهترین گرگ) در یک حلقه، ابتدا هر گرگ به عنوان یک کروموزوم برای الگوریتم ژنتیک در نظر گرفته شده و سپس با استفاده از روابط (۲۴) و (۲۵) و عملگر جهش، کروموزوم جدید (راه‌حل جدید) ساخته می‌شود و تابع برازندگی کروموزوم حاصل با تابع برازندگی کروموزوم انتخاب شده، مقایسه می‌شود. سپس اگر از لحاظ برازندگی کروموزوم جدید بهتر باشد، با کروموزوم انتخاب شده جایگزین می‌شود. در واقع به کمک الگوریتم ژنتیک موقعیت گرگ در الگوریتم

لازم به ذکر است که در روش پیشنهادی برای تولید رشته‌ها یا کروموزوم‌های بهتر از عملگر ترکیب یا آمیزش برای بازترکیب دو کروموزوم استفاده می‌شود. در الگوریتم ژنتیک، برای تولید کروموزوم‌های جدیدی در نسل‌های فعلی از ترکیب کردن دو کروموزوم در جمعیت نسل قبل، استفاده می‌شود. به عبارت دیگر، فرآیند بازترکیب، ژن‌های موجود در دو کروموزوم را ترکیب و در نتیجه، کروموزوم‌های جدیدی در جمعیت فعلی تولید می‌کند. این فرآیند به صورت تکراری و در تمامی نسل‌های یک الگوریتم ژنتیک انجام خواهد شد. تا زمانی که در آخرین نسل به راه‌حل بهینه دست پیدا کنیم، این روند ادامه دارد. لازم به ذکر است که پیچیدگی محاسباتی روش پیشنهادی از نظر زمانی $O(N^2)$ می‌باشد که در مقایسه با سایر روش‌های مورد بررسی دارای سربار محاسباتی قابل ملاحظه‌ای نیست. در شکل (۴) مراحل ایجاد راه‌حل‌های جدید (به صورت کروموزوم) در روش پیشنهادی نمایش داده شده است.

۵. ارزیابی کارایی

در این بخش به مقایسه روش پیشنهادی با سایر الگوریتم‌های مورد بررسی پرداخته شده است. سناریوهای مختلفی را برای شبکه می‌توان در نظر گرفت. در این مقاله، از سناریو مبتنی بر افزایش تعداد گره‌های اینترنت اشیا، کمک گرفته شده است. برای اینکه با قطعیت بیشتری در مورد نتایج اظهار نظر شود و تفسیر نتایج دقیق‌تر باشد، سطح اطمینان ۹۹ درصد در نظر گرفته می‌شود. به همین منظور، ابتدا ده بار روش مورد نظر اجرا شده و سپس میانگین آنها ملاک ارزیابی قرار می‌گیرد. لازم به ذکر است که معیارهای در نظر گرفته شده در این بخش شامل زمان اجرا، زمان اتمام آخرین کار و مصرف انرژی است. میانگین زمان اجرا وظایف شاخصی برای سنجش بهینه بودن میزان زمان پردازش وظایف الگوریتم پیشنهادی در لایه مه است. برای به دست آوردن این معیار، تمامی تاخیرهای شبکه‌ای و پردازشی برای یک وظیفه محاسبه و به عنوان زمان اجرا آن وظیفه در نظر گرفته می‌شود. در ادامه میانگین زمان اجرا برحسب ثانیه، برای تمامی وظایف محاسبه می‌شود.

گرگ خاکستری که یک راه‌حل برای مساله پیش‌رو است، تعیین می‌شود. پس از اعمال الگوریتم ژنتیک بر روی تمامی گرگ‌های امگا، تمامی گرگ‌ها بر اساس تابع ارزیابی مرتب می‌شوند تا سه گرگ برتر به عنوان گرگ‌های آلفا، بتا و دلتا انتخاب شده و همانند قبل مابقی گرگ‌ها به عنوان گرگ امگا در نظر گرفته شوند. این رویه تا رسیدن تعداد تکرار اجرای الگوریتم به یک تعداد معین ادامه می‌یابد.

الگوریتم (۱): شبه کد الگوریتم پیشنهادی

Algorithm.1: GaGwo Algorithm

Input: Tasks, Vms, Max-Iterations, population size (Npop)

Output: The Optimal (Sub-Optimal) Tasks Assignment

```
//X_pop is population
1: Randomly initialize the population (X_pop)
2: Calculate the fitness of each member of population
3: Sort the X_pop based on the fitness function
   /*Xα, Xβ و Xδ are the best three Top from X_pop in terms
   of fitness value*/
4: Select the best three Top from X_pop as: Xα, Xβ و Xδ
5: Xα= first best member
6: Xβ= second best member
7: Xδ= third best member
Begin:
8: For t=1 to Max-Iterations
9:   For i=4 to Npop
10:    /*x_new is a chromosome in the Genetic
    algorithm*/
    Select the i-th best fitness function from X_pop
    and use relations 24 and 25 to Generate a new
    solution (x_new)

    /*Mutation operator in genetic algorithm that is
    applied to the element of a matrix that include
    fog nodes*/
11:    Mutation(x_new)

    /*Variables A and C are vectors of meta-
    heuristic algorithm coefficients */
12:    Update A and C //(using 26 relation)
13:    Calculate fitness function for x_new

    /*Each solution is a Wolf in the Grey wolf
    algorithm that has a position and an fitness
    value*/
14:    if (x_new.fitness < X_popi.fitness) then
15:      | Update X_popi.position
16:    End if
17:    Sort the X_pop based on the fitness value and select
    the best three Top as: Xα, Xβ و Xδ
18:  End for
19:  t ← t + 1
20: End while
21: Return Xα // Best solution
```

X_α	9	5	4	7	8	3	1	2
X_β	5	2	8	6	3	1	9	7
X_δ	7	9	8	5	3	2	7	6
X	3	7	5	4	6	9	2	1

 $\Rightarrow$

X_α	9	5	4	7	8	3	1	2
X_β	5	2	8	6	3	1	9	7
X_δ	7	9	8	5	3	2	7	6
X	3	7	5	4	6	9	2	1

New chromosome accordint to GaGWO

X_Ω	3	3	5	2	2	1	4	3
------------	---	---	---	---	---	---	---	---

$$\vec{D}_\alpha = random[1,8] = 2$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha = ((9 - 2 \times 2), (5 - 1 \times 2), (4 - 1 \times 2), (7 - 2 \times 2), (8 - 2 \times 2), (3 - 1 \times 2), (1 - 2 \times 2), (2 - 2 \times 2))$$

$$= (5, 3, 2, 3, 4, 4, 3, 2)$$

$$\vec{D}_\beta = random[1,8] = 1$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta = ((5 - 2 \times 1), (2 - 1 \times 1), (8 - 1 \times 1), (6 - 2 \times 1), (3 - 2 \times 1), (1 - 1 \times 1), (9 - 2 \times 1), (7 - 2 \times 1))$$

$$= (3, 1, 7, 4, 2, 0, 7, 5)$$

$$\vec{D}_\delta = random[1,hd(1,8)] = 2$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta = ((7 - 2 \times 2), (9 - 1 \times 2), (8 - 1 \times 2), (5 - 2 \times 2), (3 - 2 \times 2), (2 - 1 \times 2), (7 - 2 \times 2), (6 - 2 \times 2))$$

$$= (3, 7, 6, 1, 1, 0, 3, 2)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} = \frac{(5, 3, 2, 3, 4, 4, 3, 2) + (3, 1, 7, 4, 2, 0, 7, 5) + (3, 7, 6, 1, 1, 0, 3, 2)}{3} = (3, 3, 5, 2, 2, 1, 4, 3)$$

شکل (۴): مراحل ایجاد راه‌حل‌های جدید (به صورت کروموزوم) در روش پیشنهادی

روش پیشنهادی از شبیه‌ساز IoTSim-Osmosis استفاده شده است [۲۱]. این شبیه‌ساز یک چارچوب برای شبیه‌سازی محیط‌های اینترنت اشیا، ابر و مه است که برنامه‌های محاسبات مه را ارزیابی می‌کند. در ضمن این شبیه‌ساز قابلیت پشتیبانی از شبکه گسترده نرم‌افزار محور را نیز دارد. تمامی آزمایش‌ها بر روی یک سیستم با پردازنده core i7-7700 HQ 2.8 Ghz با حافظه ۸ گیگابایت و سیستم عامل ویندوز ۱۰ انجام شده است. پیوند دسترسی به کد مقاله^۱ در وبگاه Github قرار داده شده است. در این سناریو سازوکار به این صورت است که تعداد سرورهای مه و ابر و همچنین نرخ وظایف ارسالی توسط هر دستگاه اینترنت اشیا ثابت در نظر گرفته شده است و با افزایش تعداد دستگاه‌های اینترنت اشیا الگوریتم پیشنهادی با معیارهای مختلف ارزیابی می‌شوند. در جدول (۳) متغیرهای شبیه‌سازی آورد شده است. در ادامه به بررسی نتایج حاصل از شبیه‌سازی با توجه به معیارهای مد نظر در این پژوهش پرداخته می‌شود. لازم به ذکر است که برای اثبات درستی نتایج شبیه‌سازی، هر سناریو به تعداد ۱۰ بار اجرا شده است و از آزمون آماری توزیع

مصرف انرژی شاخصی برای بهینه بودن میزان مصرف انرژی الگوریتم پیشنهادی می‌باشد و لازم به ذکر است که در ارزیابی میزان مصرف انرژی گره‌های مه ارزیابی می‌شوند. زمان اتمام آخرین وظیفه (makespan) که با استفاده از رابطه (۲۷) محاسبه می‌شود، بیانگر زمان پایان یافتن آخرین وظیفه در مه می‌باشد که به عنوان معیار کیفیت سرویس در نظر گرفته می‌شود.

$$makespan = \max_{t \in tasks} \{Finish Time_i\} \quad (25)$$

به منظور بررسی عملکرد، روش GAGWO با دو مقاله با روش‌هایی نزدیک به پژوهش حاضر و الگوریتم ژنتیک مقایسه می‌شود. برای شبیه‌سازی‌های انجام شده در این کار، یک محیط مه متشکل از یک FN با توپولوژی شبکه تصادفی و مجموعه‌ای از وظایف مختلف بارگذاری شده از دستگاه‌های اینترنت اشیا در نظر گرفته شده تا بر روی FN ها برنامه‌ریزی شوند. تاخیر انتشار در کلیه پیوندهای ارتباطی صفر در نظر گرفته می‌شود، در حالی که پهنای باند پیوندهای ارتباطی روی ۱۰۰۰ مگابیت بر ثانیه تنظیم شده است. لازم به ذکر است که برای پیاده‌سازی

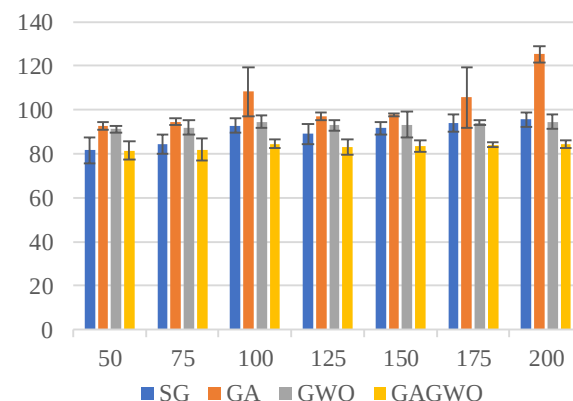
^۱ <https://github.com/msadeghy69/gagwoalgorithm>

T-student در قالب مقادیر T-Value استفاده شده است که نتایج آن در شکل های (۵) تا (۷) آورده شده است.

جدول (۳): معرفی متغیرهای شبیه ساز

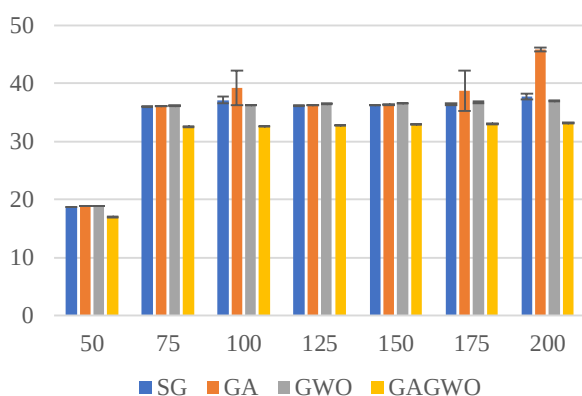
متغیر	مقدار
تعداد گره مه	۵
تعداد گره ابر	۱۶
مدت شبیه سازی	۱۰۰ ثانیه
نرخ تولید وظیفه	۱۰ وظیفه در هر ثانیه
اندازه وظیفه	۵ مگابایت
تعداد دستگاه های اینترنت اشیا	۵۰، ۷۵، ۱۰۰، ۱۲۵، ۱۵۰، ۱۷۵ و ۲۰۰
BW_c	50 Mbps
BW_c	10 Gbps

میانگین زمان اجرا: در شکل (۵) میانگین زمان اجرا برای روش پیشنهادی در مقایسه با روش مقاله نیمه حریصانه ژنتیک و گرگ خاکستری آورده شده است و همان طور که مشاهده می شود این مقدار در روش پیشنهادی در مقایسه با سایر روش های ذکر شده کمتر است. زمان همگرایی کمتر الگوریتم پیشنهادی نسبت به سایر روش ها سبب می شود در تعداد تکرارهای یکسان، الگوریتم پیشنهادی جواب های بهتری از لحاظ تاخیر به نسبت الگوریتم های دیگر به دست آورد. با توجه به شکل (۵)، هر چه تعداد دستگاه های اینترنت اشیا افزایش می یابد به تبع وظایف ارسالی بیشتر شده و میانگین زمان اجرا افزایش پیدا می کند ولی در روش پیشنهادی، تنوع جواب ها در مقایسه با الگوریتم گرگ خاکستری باعث کاهش زمان اجرا به نسبت سایر الگوریتم ها می شود.



شکل (۵): میانگین زمان اجرا با تعداد دستگاه های اینترنت اشیا مختلف

مصرف انرژی: مصرف انرژی سرورهای مه به ازای تعداد گره های مختلف اینترنت اشیا در شبکه، در شکل (۶) آورده شده است. نتایج بیانگر آن است که در روش GAGWO مصرف انرژی سرورهای مه نسبت به روش های پیشین کاهش پیدا کرده است، زیرا در روش GAGWO برخلاف روش های ژنتیک و گرگ خاکستری جواب ها گوناگونی بهتری دارند، در حالی که در الگوریتم های پیشین تنوع جواب کمتری نسبت به GAGWO وجود دارد و باعث کاهش ارائه راه حل های بهینه در زمینه انرژی می شود. همچنین الگوریتم های پیشین در بهینه های محلی گرفتار شده و بهینه ترین جواب که شامل کمترین مصرف انرژی است را ارائه نمی دهند.



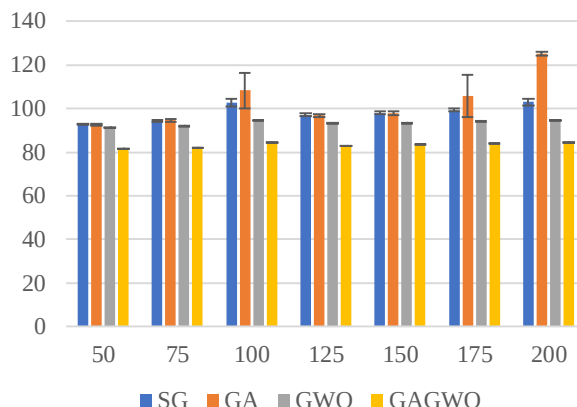
شکل (۶): مصرف انرژی با تعداد دستگاه های اینترنت اشیا مختلف

زمان اتمام آخرین وظیفه: شکل (۷) بیانگر زمان اتمام آخرین وظیفه به ازای تعداد گره های مختلف اینترنت اشیا در شبکه می باشد. همان طور که مشاهده می شود با افزایش تعداد گره های اینترنت اشیا، در روش GAGWO زمان اتمام آخرین وظیفه کاهش می یابد. دلیل این امر آن است که در روش پیشنهادی از سه کروموزوم برتر برای عمل ترکیب استفاده می شود. این کار باعث می شود راه حل های بهتری ارائه شود زیرا تنوع جواب ها افزایش پیدا می کند.

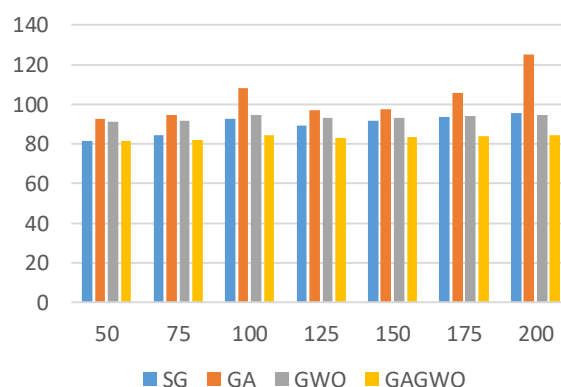
نرخ همگرایی: شکل (۸) مدت زمان اجرای الگوریتم پیشنهادی در مقایسه با سایر روش های مورد بررسی را به ازای تعداد گره های اینترنت اشیا مختلف در شبکه نشان می دهد. نتایج به دست آمده بیانگر آن است که روش پیشنهادی سرعت همگرایی بیشتری نسبت به دیگر الگوریتم های مورد بررسی دارد.

تخصیص منابع در محیط‌های محاسباتی ابر و مه پیشنهاد شده است. برای این منظور، فرمول ریاضی برای تخصیص منابع با هدف کاهش تاخیر و انرژی، در حالی که معیارهای کیفیت خدمات سطح سرویس رعایت شود، ارائه شده است. در گام بعد به ارائه الگوریتم ترکیبی گرگ خاکستری و ژنتیک پرداخته شده است. ترکیب این دو الگوریتم منجر به ارائه جواب‌های متنوع می‌شود. آزمایش‌های گسترده جهت ارزیابی روش پیشنهادی انجام شده است که اثربخشی الگوریتم پیشنهادی را تایید می‌کند. نتایج حاصله حاکی از آن است که ترکیب و استفاده همزمان از نقاط مثبت دو الگوریتم و همچنین مدل ارائه شده، معیارهای زمان اجرا و زمان اتمام آخرین کار و همچنین مصرف انرژی را به ترتیب به میزان ۱۸/۳۰ و ۱۵/۱۴ و ۱۰/۲۱ درصد بهبود می‌دهد. به عنوان کار آینده، ما قصد داریم الگوریتم‌های پیشنهادی را برای برنامه‌ریزی وظایف وابسته به یکدیگر بهبود دهیم. همچنین هدف ما ارزیابی عملکرد الگوریتم پیشنهادی با مجموعه داده‌های واقعی می‌باشد.

تعارض منافع: نویسندگان اعلام می‌کنند که هیچ تعارض منافی ندارند.



شکل (۷): زمان اتمام آخرین وظیفه با تعداد دستگاه‌های اینترنت اشیا مختلف



شکل (۸): سرعت همگرایی با تعداد دستگاه‌های اینترنت اشیا مختلف

۶. نتیجه‌گیری

در این مقاله، یک الگوریتم اکتشافی کارآمد برای حل مشکل

مراجع

- [1] Z. J. K. Abadi, N. Mansouri, and M. Khalouie, "Task scheduling in fog environment - Challenges, tools & methodologies: A review," *Comput. Sci. Rev.*, vol. 48, p. 100550, 2023, doi: 10.1016/j.cosrev.2023.100550.
- [2] M. Khorasani, M. Ramezani, and R. Khorsand, "Energy Efficient Multi Path Routing Protocol in Internet of Things," *Soft Comput. J.*, vol. 7, no. 1, pp. 34-49, 2018, doi: 10.22052/7.1.34 [In Persian].
- [3] S.M. Jameii, "Dynamic Service Provisioning in Fog Environment based on Learning Automata and Multi-objective Genetic Algorithm," *Soft Comput. J.*, vol. 11, no. 2, pp. 72-87, 2023, doi: 10.22052/SCJ.2023.248643.1115 [In Persian].
- [4] M. Nickray and E. Hosseini, "A Mobile and Fog-based Computing Method to Execute Smart Device Applications in a Secure Environment," *Soft Comput. J.*, vol. 8, no. 1, pp. 43-57, 2019, doi: 10.22052/8.1.43 [In Persian]
- [5] M.S.U. Islam, A. Kumar, and Y.-C. Hu, "Context-aware scheduling in Fog computing: A survey, taxonomy, challenges and future directions," *J. Netw. Comput. Appl.*, vol. 180, p. 103008, 2021, doi: 10.1016/j.jnca.2021.103008.
- [6] S. Azizi, M. Shojafar, J.H. Abawajy, and R. Buyya, "Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach," *J. Netw. Comput. Appl.*, vol. 201, p. 103333, 2022, doi: 10.1016/j.jnca.2022.103333.
- [7] 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, doi: 10.1109/ACCESS.2020.2975741.
- [8] H. Arri and R. Singh, “Energy Optimization-based Optimal Trade-off Scheme for Job Scheduling in Fog Computing,” in 8th Int. Conf. Comput. Sustainable Global Dev. (INDIACom), New Delhi, India, 2021, pp. 551-558.
- [9] Y. Sun, F. Lin, and H. Xu, “Multi-objective Optimization of Resource Scheduling in Fog Computing Using an Improved NSGA-II,” Wirel. Pers. Commun., vol. 102, no. 2, pp. 1369-1385, 2018, doi: 10.1007/s11277-017-5200-5.
- [10] K.S. Sahoo and B. Sahoo, “SDN Architecture on Fog Devices for Realtime Traffic Management: A Case Study,” in Proc. Int. Conf. Signal Netw. Comput. Syst., Springer, New Delhi, 2017, doi: 10.1007/978-81-322-3592-7_33.
- [11] S. Wang, T. Zhao, and S. Pang, “Task Scheduling Algorithm Based on Improved Firework Algorithm in Fog Computing,” IEEE Access, vol. 8, pp. 32385-32394, 2020, doi: 10.1109/ACCESS.2020.2973758.
- [12] S. Azizi, F. Khosroabadi, and M. Shojafar, “A priority-based service placement policy for fog-cloud computing systems,” Comput. Methods Differ. Eq., vol. 7, no. 4, pp. 521-534, 2019 [In Persian].
- [13] F. Bonomi, R.A. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in Proc. first ed. MCC Worksh. Mobile Cloud Comput., Helsinki, Finland, 2012, pp. 13-16, doi: 10.1145/2342509.2342513.
- [14] A. Yousefpour, G. Ishigaki, and J.P. Jue, “Fog Computing: Towards Minimizing Delay in the Internet of Things,” in IEEE Int. Conf. Edge Comput. (EDGE), Honolulu, HI, USA, 2017, pp. 17-24, doi: 10.1109/IEEE.EDGE.2017.12.
- [15] S. Omer, S. Azizi, M. Shojafar, and R. Tafazolli, “A priority, power and traffic-aware virtual machine placement of IoT applications in cloud data centers,” J. Syst. Archit., vol. 115, p. 101996, 2021, doi: 10.1016/j.sysarc.2021.101996.
- [16] J. Xu, Z. Hao, R. Zhang, and X. Sun, “A Method Based on the Combination of Laxity and Ant Colony System for Cloud-Fog Task Scheduling,” IEEE Access, vol. 7, pp. 116218-116226, 2019, doi: 10.1109/ACCESS.2019.2936116.
- [17] S. Misra and N. Saha, “Detour: Dynamic Task Offloading in Software-Defined Fog for IoT Applications,” IEEE J. Sel. Areas Commun., vol. 37, no. 5, pp. 1159-1166, 2019, doi: 10.1109/JSAC.2019.2906793.
- [18] S. Ghanavati, J.H. Abawajy, and D. Izadi, “An Energy Aware Task Scheduling Model Using Ant-Mating Optimization in Fog Computing Environment,” IEEE Trans. Serv. Comput., vol. 15, no. 4, pp. 2007-2017, 2022, doi: 10.1109/TSC.2020.3028575.
- [19] M. Yang, H. Ma, S. Wei, Y. Zeng, Y. Chen, and Y. Hu, “A Multi-Objective Task Scheduling Method for Fog Computing in Cyber-Physical-Social Services,” IEEE Access, vol. 8, pp. 65085-65095, 2020, doi: 10.1109/ACCESS.2020.2983742.
- [20] S. Mirjalili, S.M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” Adv. Eng. Softw., vol. 69, pp. 46-61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.
- [21] K. Alwasel, D.N. Jha, F. Habeeb, U. Demirbaga, O.F. Rana, T. Baker, S. Dustdar, M. Villari, P. James, E. Solaiman, and R. Ranjan, “IoTSim-Osmosis: A framework for modeling and simulating IoT applications over an edge-cloud continuum,” J. Syst. Archit., vol. 116, p. 101956, 2021, doi: 10.1016/j.sysarc.2020.101956.