

روش جدید جهت تخصیص منابع به جریان‌های کاری بی‌درنگ در محیط

های مه‌مبتنی بر شبکه‌های نرم‌افزار محور

محمد صادق زاده^۱، رضا محمدی^{۲*}، محمد نصیری^۳، سکینه سهرابی^۴

^۱ کارشناسی ارشد گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه بوعلی سینا، همدان، ایران

mohammad.sadeghzadeh@tejaratbank.ir

^۲ استادیار گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه بوعلی سینا، همدان، ایران

r.mohammadi@basu.ac.ir

^۳ دانشیار گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه بوعلی سینا، همدان، ایران

m.nassiri@basu.ac.ir

^۴ دانشجوی دکتری گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه بوعلی سینا، همدان، ایران

ssohrabi69@yahoo.com

چکیده

محاسبات مه به طور فزاینده‌ای به عنوان یک پلت فرم 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 تأمین کیفیت خدمات سطح سرویس مانند تأخیر 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 تأخیر 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 تأخیر 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 $18,30\%$ و $15,14\%$ و $10,21\%$ درصد به نسبت روش نیمه حریصانه بهبود می‌دهد.

کلمات کلیدی: اینترنت اشیا، محاسبات ابر، الگوریتم تخصیص منابع، الگوریتم ژنتیک، الگوریتم گرگ خاکستری

*نویسنده مسئول، ایمیل نویسنده مسئول: r.mohammadi@basu.ac.ir، رضا محمدی

A Novel Resource Allocation Approach for Real Time Workloads in SDN Based Fog Environments

Mohamad sadegh zade¹, Reza Mohammadi^{2*}, Mohamad Nassiri³, Sakine Sohrabi⁴

¹ Computer Engineering Department, Faculty of Engineering, Bu Ali Sina University
mohammad.sadeghzadeh@tejaratbank.ir

² Computer Engineering Department, Faculty of Engineering, Bu Ali Sina University
r.mohammadi@basu.ac.ir

³ Computer Engineering Department, Faculty of Engineering, Bu Ali Sina University
m.nassiri@basu.ac.ir

⁴ Computer Engineering Department, PhD Student, Faculty of Computer Engineering, Bu Ali Sina University
ssohrabi69@yahoo.com

Abstract:

Recently, fog computing has become a suitable platform for processing IoT applications. Thus, this architecture extends cloud computing services to the edge of the network, and processes are performed at the edge of the network with less delay and cost. Due to the energy limitation and low computational capacity of fog nodes, deciding to assign tasks to fog nodes is a critical issue. In this paper, a mathematical model for resource allocation with the aim of reducing delay and energy while considering Quality of Service (QoS) is proposed, and in the next step, a combined genetic and gray wolf algorithm is introduced to solve the model. The combination of these two algorithms provides helps to find an optimal solution in an efficient manner. It should be noted that the implementation of the mentioned algorithms has a processing cost and a computational delay, but due to the improvement of QoS, this cost can be ignored. Simulation results indicate that the combination and simultaneous use of the positive points of the two algorithms, the criteria of execution time and makespan of the tasks, as well as energy consumption by 18.30%, 15.14% and 10.21%, respectively, compared to the semi-greedy method improves

Keywords: Internet of Thing, Cloud Computing, Resource Allocation Algorithm, Genetic Algorithm, Gray Wolf Algorithm

*corresponding author, Email of the corresponding author: Reza Mohammadi, R.mohammadi@basu.ac.ir

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 شده که بتوان میزان تاخیر و مصرف انرژی را کاهش داد.

از سوی دیگر به دلیل آنکه منابع موجود در مه، در حال تغییر هستند، مسئله تخصیص منابع، امر مهمی است و تأثیر 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^۲ (FN) بسیار حائز اهمیت است و با توجه به ماهیت پویا، ناهمگنی منابع و محدودیت منابع مسئله‌ی زمان‌بندی وظایف در محیط مه نحوی که نیازهای کیفیت خدمات (QoS) را برآورده کنند یک مسئله NP کامل / سخت می‌باشد. برای غلبه بر این مسائل، استراتژی‌های مختلفی برای برنامه‌ریزی با توجه به اطلاعات موجودیت‌های مختلف درگیر در محاسبه مه ارائه شده است. از آنجایی که مسئله‌ی زمان‌بندی وظیفه یک مسئله NP کامل 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 دسته‌ی الگوریتم‌های ابتکاری، فراابتکاری و فوق ابتکاری بخش‌بندی می‌شوند. می‌توان گفت که هدف الگوریتم‌های ابتکاری یا هیوریستیک این

توسعه و گسترش دستگاه‌های اینترنت اشیا (IoT) باعث پدیدار شدن بسیاری از برنامه‌های اینترنت اشیا شده است [1]. اینترنت اشیا شبکه‌ای با طیف گسترده از وسایل هوشمند و اشیاء می‌باشد و اشیای فیزیکی در این شبکه قادر به تبادل اطلاعات با سایر اشیای فیزیکی هستند. از آنجایی که اینترنت اشیا سرویس‌های متنوعی را معرفی کرده است، منجر شده است که زندگی روزمره بشری به فعالیت‌های مطمئن و قابل دسترس این شبکه وابسته شود [2]. از سوی دیگر امروزه تعداد برنامه‌های کاربردی که به زمان پاسخ‌دهی کمی نیاز دارند، در حال افزایش می‌باشد [3].

بالا رفتن حجم تبادلات دستگاه‌های اینترنت اشیا باعث مصرف بالای پهنای باند، تأخیر زیاد، عدم زمان‌بندی مناسب و کاهش کارایی سیستم می‌شود. با توجه به رشد چشمگیر برنامه‌های 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 تأخیر نمی‌باشد [4]. برای رسیدگی به این مسئله، 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 تأخیر و در نتیجه بهبود کارایی سیستم می‌شود. محاسبات مه یک مدل محاسباتی غیرمتمرکز است که می‌تواند به رایانش ابری برای ارائه خدمات با کیفیت بالا (QoS) کمک کند.

دستگاه‌های مه در نزدیکی شبکه با محدودیت منابع مواجه هستند، بنابراین، زمان‌بندی کار در یک گره مه (Fog node)

است که در زمان معقول بتواند جوابی منطقی تولید کند. باید توجه داشت که این جواب لزوماً جواب بهینه مسئله نیست اما نزدیک به آن است. در واقع این مسئله 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 [5]. 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 در حوزه اینترنت اشیا، هر یک از دستگاه‌های اینترنت اشیا دارای توان محاسباتی و ذخیره‌سازی و شبکه‌ای محدودی هستند افزایش درخواست‌های کاربر منجر به ازدحام شبکه، سربار محاسباتی، افزایش ارتباطات و مدت زمان طولانی ارائه خدمات می‌شود [6]. از این رو ایده پیشنهادی در این پژوهش برای برنامه‌هایی که حساس به زمان هستند و QoS نیاز اکید آنها هست، مناسب است. نمونه‌ای از این کاربردها مراقبت‌های بهداشتی، اورژانس، بازی آنلاین و نظارت می‌باشد. عمده معیارهایی که برای مقایسه و بررسی عملکرد الگوریتم‌های زمان‌بندی مورد استفاده قرار می‌گیرند و در بررسی عملکرد یک سیستم اهمیت دارند، عبارت‌اند از: هزینه اجرا، زمان اجرا، مصرف انرژی، نرخ درخواست‌های رد شده، مدت زمان انجام کار^۳، تعادل بار^۴ در سیستم، سطح رضایت کاربر و سطح کیفیت خدمت^۵ است 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 تأخیر 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 می‌شود. بخش ۳ مدل سیستم شامل معماری و فرمول مسئله شرح داده می‌شود. الگوریتم‌های پیشنهادی ما در بخش ۴ شرح و تحلیل شده است. در نهایت در بخش ۵ ارزیابی، نتایج تجربی ارائه شده و مقاله را 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 ما در این مقاله بر روی الگوریتم‌های متمرکز می‌باشد. منظور از الگوریتم متمرکز وجود یک گره مرکزی می‌باشد که وظیفه‌ی 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 می‌پردازیم.

3 Makespan

4 Load Balance

5 Quality Of Service

۱.۲ الگوریتم‌های متمرکز و هوشمند

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

در مقاله [6] 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 به توجه به مهلت انجام وظایف می‌باشد. در روش مذکور هنگامی که وظیفه موردنظر زمان کافی برای پردازش را داشته باشد از 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 می‌کند.

مقاله [7] 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 می‌شود. 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 نوبت‌گردشی و الگوریتم مبتنی بر بهینه‌سازی ازدحام ذرات می‌شود. از سوی دیگر با افزایش تعداد گره‌های دستگاه‌های اینترنت اشیا عملکرد الگوریتم کلونی مورچگان نسبت به الگوریتم ازدحام ذرات در مبحث زمان پاسخگویی وظایف بارگذاری شده تا حد زیادی بهبود داشته است و این روش به طور مؤثر باعث تعادل بارکاری بر روی گره‌های مه می‌شود.

در مقاله [8]، الگوریتم MBFD ارائه شده است. الگوریتم مذکور یک روش بهینه‌سازی مبتنی بر تکنیک 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 الگوریتم مذکور 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 همچون تعادل بار و اولویت پرداخته نشده است. در مقاله [9] 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 غیر هوشمند ارائه شده می‌پردازیم.

در مقاله مسیریابی چندلایه‌ای در محیط‌های مه‌آلود با زیرساخت شبکه‌های نرم‌افزار محور^۶ [10] با در نظر گرفتن ناهمگنی گره‌ها از نظر ویژگی‌های نرم‌افزاری و سخت‌افزاری و خدمات و برنامه‌های حساس به تأخیر مسیریابی انجام می‌شود و گره‌ها به صورت پویا به شبکه وارد و یا از شبکه خارج می‌شوند در 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 به‌عنوان گره مه در نظر گرفته می‌شود [11].

در مقاله [12]، الگوریتم MDAF با توجه به معیار کیفیت سرویس برای سیستم‌های محاسبات ابر-مه است، ارائه شده است که یک ساز و کار زمان‌بندی و اختصاص خدمات می‌باشد. در روش مذکور تا حد ممکن در لایه‌ای که به کاربر نزدیک‌تر

⁶ Software-Defined Wide Area Networking

- مه - ابر استفاده کرد. در این مقاله به معیارهایی همچون هزینه و تعادل بار پرداخته نشده است.

در مقاله [16]، الگوریتم LBP-ACS توضیح داده شده است. روش پیشنهادی ترکیبی از الگوریتم‌های ACS و LBPA می‌باشد و به منظور حل مشکل زمان‌بندی کار در محیط محاسبات مه ارائه شده است. در الگوریتم مذکور برای زمان‌بندی کارها مهلت پایان‌یافتن و اولویت کارها در نظر گرفته شده است. در روش مذکور به منظور رسیدگی به حساسیت تأخیر کار، برای زمان‌بندی و ساخت یک دنباله کار با اولویت معقول از الگوریتم Laxity-based priority استفاده می‌شود. همچنین از الگوریتم بهینه‌سازی مبتنی بر الگوریتم سیستم 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) تکنیک مورد استفاده، ابزار شبیه‌سازی، معیارهای کارایی، دیتاست و ویژگی‌های مربوط به الگوریتم‌های بررسی شده در این پژوهش به اختصار آورده شده است.

می‌باشد وظایفی که به زمان حساس‌تر هستند، سرویس داده می‌شوند. نتایج شبیه‌سازی بیانگر آن است که الگوریتم فوق از نظر هزینه اجرا و تأخیر بهبود قابل توجهی داشته است، اما در این مقاله به معیار منابع و بهره‌وری انرژی پرداخته نشده است و از منابع به‌طور مؤثرتر استفاده نمی‌شود.

در مقاله [13]، 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 به‌طور خودکار از یک گروه منابع از پیش تخصیص‌یافته 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 [14]، یک چارچوب کلی برای برنامه‌های کاربردی اینترنت اشیا و مه معرفی شده است و یک سیاست به منظور به حداقل رساندن تأخیر برای دستگاه‌های دارای مه ارائه می‌شود. هدف روش مذکور کاهش تأخیر سرویس برای برنامه‌های اینترنت اشیا است. در ادامه یک مدل تحلیلی برای ارزیابی سیاست تخصیص و ارائه یک ساز و کار پیشنهادی برای کاهش تأخیر ارائه داده شده است.

در مقاله [15]، 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 مؤثر مسئله 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 هدرفت منابع و مصرف انرژی را کاهش دهد. امروزه محققان در تلاش هستند تا بتوانند روش‌های پیشنهادی را گسترش دهند طوری که بتوانند از آن‌ها در محیط ترکیبی دستگاه اینترنت اشیا

جدول ۱: ویژگی‌های مربوط به الگوریتم‌های بررسی شده

مقاله	هوشمندی	مزایا	معایب	تکنیک مورد استفاده	ابزار شبیه سازی	دیتاست
[۶]	هوشمند	کاهش مصرف انرژی و به حداقل رساندن زمان نقض مهلت انجام کار	عدم جستجوی کامل فضای راه‌حل‌های مسئله و قرار گرفتن در بهینه‌های محلی	SEMI-GREEDY (PSG Algorithm)	C++ programming language	فاقد دیتاست
[۷]	هوشمند	زمان پاسخ و تعادل بار	در نظر نگرفتن معیارهای انرژی و زمان اتمام آخرین وظیفه	ACO and PSO Algorithm	Matlab	فاقد دیتاست
[۹]	هوشمند	معیارهای زمان اتمام کار و مصرف انرژی	عدم در نظر گرفتن معیارهایی همچون اولویت و تعادل بار	ABC and ANN Algorithm	Matlab	فاقد دیتاست
[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 نرم افزار محور	WOA algorithm	iFogSim	فاقد دیتاست
[۱۲]	غیر هوشمند	تأخیر و هزینه اجرا	عدم توجه به معیارهای بهره‌وری انرژی و منابع	MDAF algorithm	iFogSim	فاقد دیتاست
[۱۳]	غیر هوشمند	تعادل بار و تاخیر	در نظر نگرفتن معیار انرژی	Petri nets	MANET	فاقد دیتاست
[۱۴]	غیر هوشمند	به حداقل رساندن تأخیر	گرسنگی وظایف با حجم بالا	linear programming	Matlab	فاقد دیتاست
[۱۵]	غیر هوشمند	اولویت، توان و ترافیک	عدم در نظر گرفتن معیارهایی تعادل بار و هزینه	JointPT Algorithm	Java programming language	فاقد دیتاست
[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 دیتاست

۳ مدل‌سازی و فرمول‌سازی مسئله

این بخش شامل دو زیربخش است که به صورت کلی به توضیح مدل سیستم پرداخته است. در بخش اول، معماری کلی اینترنت اشیا-مه-ابر ارائه شده است. در بخش دوم، مسئله زمان‌بندی کار به صورت ریاضی فرموله شده است. مدل پیشنهادی ترکیبی از مدل‌های ارائه شده در مقالات [6]، [7]، [17] و [18] می‌باشد.

۱۳، مدل سیستم

معماری سیستم از سه بخش دستگاه‌های اینترنت اشیا، محیط مه و محیط ابری تشکیل شده است.

الف) دستگاه‌های اینترنت اشیا: این بخش شامل دستگاه‌های اینترنت اشیا می‌باشد که با نماد S (که در واقع بیانگر تعداد حسگر می‌باشد) نشان داده می‌شود. لازم به ذکر است که در شبیه‌سازی انجام شده از سنسور دما استفاده شده است.

ب) 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 دارد و وظایف را بر اساس الگوریتم تخصیص منابع که در بخش بعد تشریح می‌شود تخصیص می‌دهد.

ج) محیط ابری: در این بخش گره ابر به صورت C_k نشان داده می‌شود که شامل سرورهایی است که از مجموعه‌ای از ماشین‌های مجازی^۷ با قدرت محاسباتی و ظرفیت ذخیره‌سازی بالا تشکیل شده است.

۲۳، فرموله سازی مسئله

در این بخش، یک مدل رسمی برای مسئله زمان‌بندی کار در یک سیستم محاسباتی مه ارائه شده است. ابتدا عناصر اساسی و متغیرهای تصمیم‌گیری مسئله بهینه‌سازی معرفی می‌شوند. سپس، مدل‌های زمان پاسخ و انرژی تشریح می‌شوند. در نهایت، بررسی اجمالی مشکل ارائه شده است. جدول (۲) پارامترها و نمادهای پرکاربرد در روابط مدل‌سازی مسئله آورده شده است.

۱.۲۳، متغیرهای اساسی

درستی شکل‌ها گره‌های مه و وظایف تولیدی توسط اینترنت اشیا از متغیرهای اساسی به شمار می‌روند که به شرح زیر فرموله می‌شوند. در مدل پیشنهادی مجموعه وظایف مستقل به صورت $T = \{t_1, t_2, \dots, t_n\}$ در نظر گرفته می‌شود که این وظایف از دستگاه‌های اینترنت اشیا $S = \{s_1, s_2, \dots, s_w\}$ به مجموعه‌ای از گره‌های مه $F = \{f_1, f_2, \dots, f_m\}$ تخصیص داده می‌شود. این تخصیص توسط fb انجام می‌پذیرد. همچنین گره‌های ابر به صورت $C = \{c_1, c_2, \dots, c_n\}$ نمایش داده می‌شود.

مسئله اصلی تخصیص این وظایف به گره‌های مه می‌باشد و نگاهت $\Psi: T \rightarrow F$ باید طوری تعریف شود که مصرف انرژی و تأخیر کل سیستم کمینه شود. بنابراین ما این مسئله را با استفاده از معادله (۱) مدل می‌کنیم.

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

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

⁷ Virtual Machine

جدول ۲: معرفی پارامترهای پرکاربرد در روابط مدل‌سازی مسئله

معنی	نماد	معنی	نماد
دستگاه اینترنت اشیا	s_i	میزان 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	fb	مدت زمان اجرای روشن بودن دستگاه اینترنت اشیا i	A_i
سرورهای ابر	C_k	پهنای باند لینک ارتباطی دستگاه اینترنت اشیا تا کارگزار مه	BW_s
مجموعه وظایف	t_i	پهنای باند لینک ارتباطی کارگزار مه تا کارگزار ابر	BW_c
کارگزار ابر	cb	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_{fb}	مجموع انرژی مصرفی	J_E
نرخ سرویس دهی گره مه	μ_f	تأخیر کل سیستم	J_D
نرخ وظایف دریافتی گره مه	$\lambda_{s_{total}}$		
مصرف انرژی زمان بارگذاری وظایف از مه به ابر	$\epsilon_i^{offloading}$		

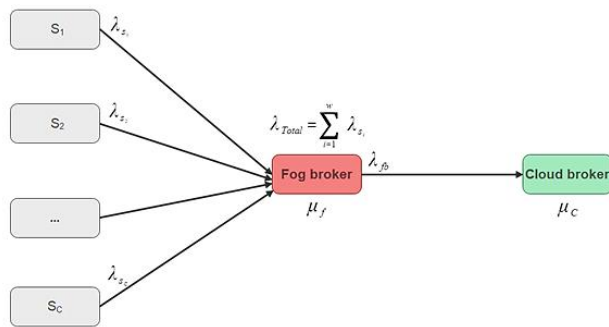
۱.۱.۲.۳ مدل بار کاری

همان‌طور که گفته شد در این پژوهش $T = \{t_1, t_2, \dots, t_n\}$ بیانگر مجموعه n تا وظایف 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 مؤلفه آورده شده در معادله (۲) تعریف می‌شود.

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

با توجه به معادله (۲) متغیرهای t_i^m ، t_i^d و t_i^s به ترتیب بیانگر تعداد دستورالعمل، مهلت زمان انجام یک وظیفه (ثانیه) و اندازه وظیفه (کیلوبایت) می‌باشد [7].

مدل بار کاری سیستم در شکل (۱) آورده شده است که در ادامه به تشریح این مدل می‌پردازیم. با توجه به شکل (۱) هر دستگاه اینترنت اشیا (S_i) با نرخ λ_{S_i} وظایف را تولید می‌کند و در کارگزار مه نرخ ورود بارکاری، مجموع تمامی نرخ‌های دستگاه‌های اینترنت اشیا می‌باشد و سپس کارگزار مه وظایف را پس از پیش پردازش با نرخ (λ_{fb}) به سمت کارگزار ابر ارسال می‌کند.



شکل (۱) فرایند اجرای وظیفه در روش پیشنهادی

۲.۲۳. متغیرهای تصمیم [6]:

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.

$$\forall i \in T, \forall j \in F \quad (3)$$

$$X_{ij} = \begin{cases} 1 & \text{if } t_i \text{ assigned to } f_j \\ 0 & \text{Otherwise} \end{cases}$$

در فرمول (۳) برای هر وظیفه t_i ، متغیر دودویی X_{ij} را تعریف کرده و با استفاده از فرمول (۴) مقدار این متغیر مشخص می‌شود. با استفاده از متغیر e_{ij} انتخاب یا عدم انتخاب لینک 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 فرمول (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 فرمول (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 معادله (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 تأخیر 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 تأخیر:

در این بخش ابتدا تأخیر دستگاه اینترنت اشیا تا کارگزار مه مورد بررسی قرار گرفته و سپس وظایفی که به کارگزار مه منتقل می‌شوند تأخیر انتقال آنها از دستگاه اینترنت اشیا به کارگزار مه

محاسبه می‌شود. متغیر $L_{iot-fb(i)}$ بیانگر این مقدار می‌باشد و با استفاده از رابطه‌ی $L_{iot-fb(i)} = Prp_{Delay}^{iot-fb} + \frac{t_i^s}{BW_s}$ محاسبه می‌شود. متغیر Prp_{Delay}^{iot-fb} بیانگر تأخیر انتشار لینک ارتباطی بین دستگاه اینترنت اشیا و کارگزار مه می‌باشد. تأخیر پردازشی وظیفه t_i در گره مه با توجه به فرکانس پردازشی و بصورت $\delta_i^{fog} = \frac{t_i^m}{f_j^{mips}}$ محاسبه می‌شود.

باید توجه داشت که در هنگامی که پس از اتمام وظیفه کارگزار مه تصمیم به بارگیری وظایف به سمت ابر می‌گیرد میزان تأخیر با کمک رابطه‌ی $L_{fb-cb(i)} = Prp_{Delay}^{fb-cb} + \frac{t_i^s}{BW_c}$ محاسبه می‌شود. لازم به ذکر است که متغیرهای BW_c ، t_i^s و Prp_{Delay}^{fb-cb} به ترتیب بیانگر پهنای باند لینک ارتباطی بین کارگزار مه و کارگزار ابر، اندازه وظیفه و تأخیر انتشار لینک ما بین کارگزار ابر و کارگزار مه می‌باشد.

از سوی دیگر برای محاسبه‌ی تأخیر پردازشی گره ابر از رابطه‌ی $\delta_i^{cloud} = \frac{t_i^m}{c_k^{mips}}$ استفاده می‌شود. در این پژوهش از تأخیر انتشار چشم پوشی شده است. در روش پیشنهادی تأخیر بصورت رابطه (۶) قابل محاسبه است.

$$J_D = \sum_{i=1}^n L_{iot-fb(i)} + \delta_i^{fog} + L_{fb-cb(i)} + \delta_i^{cloud} \quad (6)$$

برای هر وظیفه t_i میزان زمان نقض مهلت انجام آن را می‌توان با استفاده از رابطه (۷) محاسبه کرد.

$$v_i = \max(0, J_D - t_i^d) \quad \forall i \in T \quad (7)$$

۴.۲۳، مدل مصرف انرژی:

در فرمول (۹) متغیر $M_j(t)$ بیانگر زمان اتمام آخرین وظیفه i در گره f_j در بازه زمانی t می‌باشد. برای محاسبه‌ی زمان در حالت ایستا گره f_j از رابطه (10) استفاده می‌شود.

$$l_j(t) = (M_j(t-1) - A_j) \quad \forall j \in F \quad (10)$$

بنابراین مصرف انرژی کلی f_j با استفاده از رابطه (11) محاسبه می‌شود [6].

$$\epsilon_j = \sum_{j=0}^m A_j * F_j^{act} + l_j(t) * F_j^{idl} \quad (11)$$

در فرمول (۱۱) 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 می‌باشد. انرژی مصرفی در اینترنت اشیا با توجه به اینکه داده‌ها پیوسته در حال تولید می‌باشند با استفاده از رابطه (12) محاسبه می‌شود.

$$\epsilon_i = \sum_{i=0}^n A_i * S_i^{act} \quad \forall i \in S \quad (12)$$

با توجه به رابطه (۱۲) متغیر A_i بیانگر مدت زمان روشن بودن دستگاه اینترنت اشیا و متغیر S_i^{act} بیانگر توان مصرفی می‌باشد. زمانی که کارگزار گره مه تصمیم به بارگذاری وظیفه به کارگزار ابر می‌گیرد انرژی مصرف شده برای این کار با استفاده از رابطه (13) محاسبه می‌شود.

$$\epsilon_j^{offloading} = \sum_{i=0}^T p_{fb-cb} * \frac{t_i^s}{BW_c} \quad (13)$$

با توجه به فرمول (۱۳) متغیر p_{fb-cb} نشان‌دهنده‌ی 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 فرمول (14) محاسبه می‌شود.

$$J_E = \epsilon_j^{offloading} + \epsilon_i + \epsilon_j + \epsilon_h \quad (14)$$

در رابطه (۱۴) متغیر $\epsilon_j^{offloading}$ بیانگر انرژی مصرف شده برای بارگذاری وظایف از کارگزار مه به سمت کارگزار ابر می‌باشد. متغیرهای ϵ_i و ϵ_j به ترتیب بیانگر انرژی مصرفی دستگاه

در این قسمت مدلی برای انرژی مصرف شده برای سرورهای پردازشی ارائه می‌شود. 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 وظیفه‌ای 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 می‌باشد.

انرژی مصرف شده توسط یک گره $f_j \in F$ از مجموع انرژی مصرف شده در حالت ایستا و حالت پویا به دست می‌آید.

انرژی مصرف شده توسط یک گره $f_j \in F$ از مجموع انرژی مصرف شده در حالت ایستا و حالت پویا به دست می‌آید. حالت پویا f_j برابر است با زمان مورد نیاز برای پردازش تمام وظایف محول شده به گره مه که با رابطه (8) قابل محاسبه است.

$$A_j = \sum_{i \in T} \delta_i^{fog} * x_{ij} \quad \forall j \in F \quad (8)$$

در فرمول (۸) متغیر δ_i^{fog} بیانگر زمان پردازش وظیفه t_i در گره f_j می‌باشد 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 می‌شود برابر یک است و در غیر این صورت صفر می‌باشد. برای به دست آوردن زمان ایستا هر f_j ، ابتدا نیاز داریم که زمان اتمام آخرین وظیفه^۹ یک f_j را در بین همه f_j ها به دست آوریم که ابتدا طبق رابطه (9) زمان اتمام آخرین وظیفه را محاسبه می‌کنیم [18].

$$M_j(t) = \max_{j \in F} A_j \quad (9)$$

⁹ Makepan

بیان می‌کند که نیازمندی‌های شبکه‌ای وظیفه T_i نمی‌تواند از پهنای باند لینک 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 قبلاً 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 به مدیریت خدمات شبکه از طریق انتزاع سطح بالاتر می‌نماید.

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 [19] می‌باشد که برای تخصیص بهینه منابع استفاده می‌شود. الگوریتم ژنتیک دارای قابلیت جستجوی چند جانبه، تولید راه‌حل‌های جدید با استفاده از فرآیند تکامل است و به دلیل سادگی بسیار موثر و سریع عمل می‌کند که در روش پیشنهادی از این الگوریتم بهره‌گرفته شده است. از سوی دیگر الگوریتم بهینه‌سازی گرگ خاکستری GWO [20]، یک الگوریتم فراابتکاری است که از رفتار اجتماعی گرگ‌های خاکستری در هنگام شکار کردن الهام گرفته است و دارای ساختار سلسله‌مراتبی می‌باشد. این الگوریتم مبتنی بر جمعیت

اینترنت اشیا می‌باشد و انرژی مصرفی گره مه می‌باشد. همچنین مقدار متغیر E_n برابر میزان انرژی مصرفی سرور ابر می‌باشد که با استفاده از رابطه (12) به دست می‌آید.

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

5.23, محدودیت‌ها:

در مبحث تخصیص منابع با توجه به ماهیت محدود ظرفیت گره‌ها و لینک‌های ارتباطی محدودیت‌ها با استفاده از روابط (۱۵) و (۱۶) تعریف می‌شود.

$$\sum_{i=0}^n \sum_{j=0}^m X_{ij} = 1 \quad (15)$$

$$\sum_{i=0}^n \sum_{k=0}^h X_{ik} = 1 \quad (16)$$

رابطه (۱۵) بیانگر محدودیت مربوط به هر وظیفه است که بیان می‌کند هر وظیفه فقط و فقط باید به یک گره مه اختصاص داده شود. همچنین رابطه (۱۶) بیانگر محدودیت دیگری است که بیان می‌کند که هر وظیفه فقط می‌تواند دقیقاً به یک گره ابر اختصاص داده شود. در بحث‌های پردازشی و پهنای باند نیز روابط (۱۷) تا (۲۰) تعریف می‌شود [6].

$$\sum_{i=1}^n t_i^m \leq \sum_{j=1}^m f_j^{mips} \quad (17)$$

$$\sum_{i=1}^n t_i^m \leq \sum_{k=1}^h c_k^{mips} \quad (18)$$

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

$$\sum_{i=1}^n B_i \leq BW_c \quad (20)$$

روابط (۱۷) و (۱۸) به ترتیب محدودیت‌های توانایی پردازشی گره‌های مه و گره‌های ابری را مشخص می‌کنند [6]. همچنین روابط (۱۹) و (۲۰) بیانگر محدودیت مربوط به شبکه است و

تشکیل می‌شوند از این رو می‌توان 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) مدل‌سازی کروموزوم در روش پیشنهادی

همان‌طور که در شکل (۲) آورده شده است، باید توجه داشت که اندیس‌های یک کروموزوم (آرایه آورده شده در شکل (۲)) بیانگر شماره وظایف وارد شده به سیستم می‌باشد و متغیرهای f_1 تا f_j بیانگر مجموعه گره‌های مه می‌باشد که با استفاده روش پیشنهادی وظایف به گره‌های مه نگاشت می‌شوند.

جمعیت اولیه در روش پیشنهادی مجموعه‌ای از راه‌حل‌های تصادفی (کروموزوم‌ها) می‌باشد که برای تخصیص بهینه منابع تولید شده‌اند که در ابتدا به صورت تصادفی مقداردهی می‌شوند و سپس در فازهای الگوریتم به سمت راه‌حل بهینه همگرا می‌شود.

به منظور اینکه راه‌حل‌ها در فرایند فراابتکاری شرکت داده شوند و در نتیجه به سمت جواب بهینه که همان راه‌حل مطلوب برای تخصیص وظایف می‌باشد؛ همگرا شوند. بایستی برانزنگی این راه‌حل‌ها محاسبه گردد. در الگوریتم‌های فرا اکتشافی به ازای تک تک اعضای جمعیت اولیه تابع برانزنگی محاسبه شده و به ازای هر عضو ذخیره می‌شود. در روش پیشنهادی نیز به ازای هر کروموزوم تابع برانزنگی محاسبه خواهد شد. بر همین اساس تابع پیشنهادی برای محاسبه برانزنگی کروموزوم‌ها در روش پیشنهادی ترکیبی از شاخص‌های یاد شده خواهد بود. این تابع طبق رابطه (۲۱) مدل‌سازی می‌گردد.

$$fitness(chromozome) = \alpha \cdot J_E + (1 - \alpha) \cdot J_D \quad (15)$$

در گام بعد از روش پیشنهادی، عملگر ترکیب الگوریتم بهینه سازی ژنتیک اجرا می‌شود. از فرایند ترکیب کروموزوم‌ها در جهت تولید راه‌حل‌های جدید استفاده می‌شود و برای این منظور از عملگر بروز رسانی الگوریتم بهینه‌سازی گرگ خاکستری بهره

برده، به سادگی قابلیت تعمیم به مسائل با ابعاد بزرگ را دارد و فرایند ساده‌ای دارد. در الگوریتم گرگ خاکستری دو فرآیند استخراج و اکتشاف بسیار مهم می‌باشد که برقراری تعادل میان این دو فاکتور دارای اهمیت می‌باشد. منظور از اکتشاف سراسری بودن جستجو است و این بدین معنا است که نیازمند جستجوی بیشتر می‌باشد. منظور از استخراج یا بهره‌وری به دست آوردن پاسخ‌های بهتر حول یک پاسخ می‌باشد. اکتشاف پاسخ‌های جدیدی را ارائه می‌دهد 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 می‌کند. در این روش یک راه‌حل در نظر گرفته می‌شود و در طول اجرا تغییرات روی همان راه‌حل اعمال می‌شود. باید توجه داشت که فرآیند اکتشاف منجر به تولید راه‌حل جایگزین می‌شود تا قادر به واکنش انعطاف پذیرتر نسبت به تغییرات مورد نیاز محیط باشد. لازم به ذکر است که در الگوریتم گرگ خاکستری زمانی که $|A| \geq 1$ الگوریتم در فاز اکتشاف می‌باشد و زمانی که $|A| \leq 1$ که الگوریتم وارد فاز استخراج می‌شود.

در این مقاله برای این منظور پیشنهاد می‌شود که از عملگر به روزرسانی الگوریتم بهینه‌سازی گرگ خاکستری به جای عملگر ترکیب پیش فرض در الگوریتم بهینه‌سازی ژنتیک استفاده شود. بر اساس این عملگر، راه‌حل‌های جدید برای مسئله به تبعیت از سه کروموزوم برتر که بالاترین میزان برانزنگی را دارا می‌باشند و به راه‌حل بهینه نزدیک هستند؛ ایجاد می‌شوند. در این بخش به تشریح گام‌های مختلف الگوریتم پیشنهادی می‌پردازیم:

در روش پیشنهادی هر کروموزوم به عنوان یک راه‌حل بالقوه برای تخصیص وظایف به سلول‌های مه در نظر گرفته می‌شود؛ در الگوریتم ترکیبی گرگ خاکستری و ژنتیک، هر کروموزوم نشان دهنده یک راه‌حل ممکن برای مسئله مورد نظر و یک نقطه در فضای جستجو می‌باشد. خود کروموزوم‌ها از تعداد ثابتی ژن

$$\vec{D}_\beta = \text{random}[1, \text{hd}(X_\beta, X)],$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta)$$

$$\vec{D}_\delta = \text{random}[1, \text{hd}(X_\delta, X)],$$

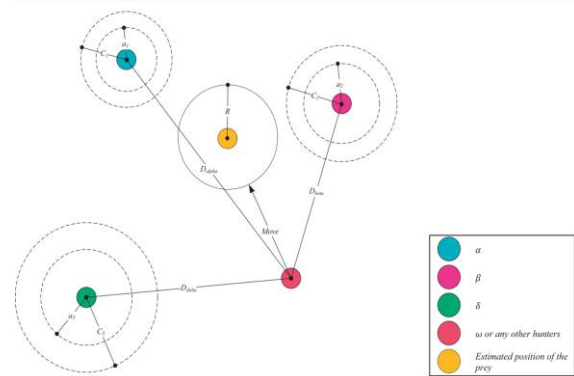
$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (23)$$

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r}_1 - \vec{a} \quad \vec{C} = 2 \cdot \vec{r}_2 \quad (24)$$

با توجه به معادله‌ی (۲۳) متغیر $\vec{X}(t+1)$ بیانگر راه حل جدید و در واقع موقعیت جدیدی است که با استفاده از موقعیت گرگ‌های آلفا و بتا و دلتا به دست آمده است. در معادله (۲۴) متغیرهای A و C که با استفاده از رابطه (۲۴) به دست می‌آیند، بیانگر بردارهای ضرایب الگوریتم فراابتکاری هستند. مقدار A به صورت خطی از ۲ تا ۰ کاهش می‌یابد. متغیرهای r_1 و r_2 نشان‌دهنده‌ی یک بردار تصادفی با مقداری بین صفر و یک می‌باشند [20]. با توجه به ماهیت گسسته مسئله متغیر \vec{D}_α 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 بر روی جمعیت اولیه کروموزوم‌ها اعمال گردید. تعدادی کروموزوم جدید به جمعیت اولیه که توسط الگوریتم گرگ خاکستری ایجاد شده است، اضافه می‌گردد. در شکل (۴) شبه کد مربوط به ترکیب الگوریتم گرگ خاکستری و ژنتیک آورده شده است.

گرفته شده است. این کار منجر به افزایش کارایی الگوریتم بهینه سازی ژنتیک در تولید راه‌حل‌های جدید (کروموزوم‌ها) برای تخصیص بهینه وظایف و همگرایی مطلوبی راه‌حل‌ها به راه‌حل بهینه می‌شود.



شکل (۳) به‌روزرسانی موقعیت‌ها در الگوریتم گرگ خاکستری

شکل (۳) بیانگر فرایندی است که در به‌روزرسانی الگوریتم بهینه سازی گرگ خاکستری انجام می‌شود؛ در روش پیشنهادی برای تولید فرزندان جدید از الگوریتم گرگ خاکستری استفاده خواهد شد. به این منظور پس از آنکه جمعیت موردنظر برای فاز ترکیب انتخاب شد. نخست سه کروموزومی که بهترین میزان برازندگی را دارند؛ از بین جمعیت اولیه الگوریتم بهینه‌سازی ژنتیک انتخاب و به‌عنوان گرگ‌های آلفا، بتا و دلتا ذخیره می‌شوند. سپس مابقی کروموزوم‌ها به استخر جفت منتقل می‌شوند. درآمد عملگر ترکیب بر روی کروموزوم‌هایی که به‌عنوان 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 را با راه‌حل‌های موجود در استخر جفت‌گیری، به‌روزرسانی می‌کنند و بدین صورت فرزندان جدید طبق الگوریتم گرگ خاکستری ایجاد می‌شوند. رابطه (۲۲) این فرایند را مدل‌سازی کرده است.

$$\vec{D}_\alpha = \text{random}[1, \text{hd}(X_\alpha, X)],$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha) \quad (22)$$

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 می‌شوند. سپس مابقی گرگ‌ها (راه‌حل‌ها) به عنوان گرگ امگا انتخاب در نظر گرفته می‌شوند.

در ادامه به کمک الگوریتم ژنتیک سعی در بهبود گرگ‌های امگا می‌شود. به این صورت که به ازای تک تک گرگ‌های امگا (از چهارمین بهترین گرگ) در یک حلقه ابتدا هر گرگ را به عنوان یک کروموزوم برای الگوریتم ژنتیک در نظر می‌گیریم و سپس با استفاده از روابط ۲۲ تا ۲۴ 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 گرگ خاکستری که یک راه‌حل برای مسئله پیش‌رو می‌باشد، تعیین می‌شود. پس از اعمال الگوریتم ژنتیک بر روی تمامی گرگ‌های امگا، تمامی گرگ‌ها بر اساس تابع ارزیاب مرتب می‌شوند تا سه گرگ برتر به عنوان گرگ‌های آلفا، بتا و دلتا انتخاب شوند و همانند قبل مابقی گرگ‌ها به عنوان گرگ امگا در نظر گرفته می‌شوند. این رویه تا رسیدن تعداد تکرار اجرای الگوریتم به یک تعداد معین ادامه می‌یابد.

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، فرایند بازترکیب، ژن‌های موجود در دو کروموزوم را ترکیب و در نتیجه، کروموزوم‌های جدیدی در جمعیت فعلی تولید می‌کند.

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
   /*X_alpha, X_beta and X_delta are the best three Top from X_pop in terms of
   fitness value*/
4: Select the best three Top from X_pop as: X_alpha, X_beta and X_delta
5: X_alpha = first best member
6: X_beta = second best member
7: X_delta = third best member
Begin:
8: For t=1 to max_number_of_iterations
   // Npop is population size
9:   For i=4 to Npop
     /*x_new is a chromosome in the Genetic
     algorithm*/
     Select the i-th best fitness function from X_pop
     and use relations 22 to 24 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*/
10:    Mutation(x_new)

     /*Variables A and C are vectors of meta-heuristic
     algorithm coefficients */
11:    Update A,C //(using 22 relation)

12:    Calculate fitness function for x_new

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

```

شکل (۴) شبه کد الگوریتم پیشنهادی

شکل (۴) شبه کد الگوریتم پیشنهادی آورده شده است که نحوه ترکیب الگوریتم گرگ خاکستری و ژنتیک را شرح می‌دهد. در

بهینه دست پیدا کنیم، این روند ادامه دارد.

این فرایند به صورت تکراری و در تمامی نسل‌های یک الگوریتم ژنتیک انجام خواهد شد. تا زمانی که در آخرین نسل به راه‌حل

شکل (۵) مراحل ایجاد راه‌حل‌های جدید (به صورت کروموزوم) در روش پیشنهادی نمایش داده شده است.

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 نمی‌باشد. در

Alpha	9	5	4	7	8	3	1	2
Beta	5	2	8	6	3	1	9	7
Delta	7	9	8	5	3	2	7	6
X	3	7	5	4	6	9	2	1

Use relations 22 to 24 to generate a new solution

Alpha	9	5	4	7	8	3	1	2
Beta	5	2	8	6	3	1	9	7
Delta	7	9	8	5	3	2	7	6
X	3	7	5	4	6	9	2	1
omega	3	3	5	2	2	1	4	3

} new chromosome according to GAGWO

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

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

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

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

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

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

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

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

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

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

$$= \frac{\lfloor (5, 3, 2, 3, 4, 4, 3, 2) + (3, 1, 7, 4, 2, 0, 7, 5) + (3, 7, 6, 1, 1, 0, 3, 2) \rfloor}{3}$$

$$= 3, 3, 5, 2, 2, 1, 4, 3$$

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r}_1 - \vec{a} \quad \vec{C} = 2 \cdot \vec{r}_2$$

شکل (۵) مراحل ایجاد راه‌حل‌های جدید (به صورت کروموزوم) در روش پیشنهادی

پذیرفته شده در مجله محاسبات نرم

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 به منظور بررسی روش GAGWO و به دست آوردن نتایج، 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 می‌شود 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 همچنین مصرف انرژی می‌باشد.

الف) میانگین زمان اجرا: میانگین زمان اجرا وظایف شاخصی برای سنجش بهینه بودن میزان زمان پردازش وظایف الگوریتم پیشنهادی در لایه مه می‌باشد. برای به دست آوردن این معیار، تمامی تأخیرهای شبکه‌ای و پردازشی برای یک وظیفه به دست آمده و به عنوان زمان اجرا یک وظیفه در نظر گرفته می‌شود. در ادامه میانگین زمان اجرا که واحد این شاخص ثانیه است، برای تمامی وظایف محاسبه می‌شود.

ب) 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.

ج) زمان اتمام آخرین وظیفه: معیار makespan که با استفاده از رابطه (۲۵) محاسبه می‌شود، بیانگر زمان پایان یافتن آخرین وظیفه در مه می‌باشد که به عنوان معیار کیفیت سرویس در نظر گرفته می‌شود.

$$\text{Makespan} = \text{MAX}_{i \in \text{tasks}} \{ \text{Finish Time}_i \} \quad (25)$$

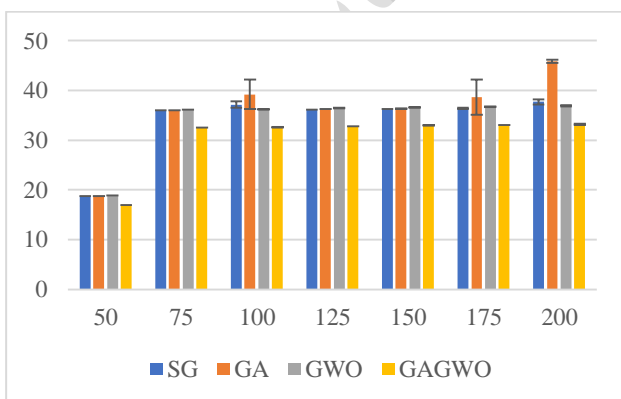
به منظور بررسی عملکرد، روش GAGWO با دو مقاله‌ای که نزدیک به پژوهش انجام شده است و الگوریتم ژنتیک مقایسه می‌شود.

برای آزمایش‌های شبیه‌سازی انجام شده در این کار، ما یک محیط مه متشکل از یک FN با توپولوژی شبکه تصادفی و مجموعه‌ای از وظایف مختلف بارگذاری شده از دستگاه‌های IoT را در نظر می‌گیریم تا بر روی FNها برنامه‌ریزی شوند. تأخیر انتشار در کلیه لینک‌های 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 روش پیشنهادی از شبیه‌ساز IoTSim-Osmosis استفاده شده است [21]. IoTSim-Osmosis یک چارچوب شبیه‌سازی برای شبیه‌سازی محیط‌های اینترنت اشیا و ابر و مه است که برنامه‌های محاسبات مه در این شبیه‌سازی ارزیابی می‌شوند این شبیه‌ساز قابلیت پشتیبانی از شبکه گسترده نرم‌افزار محور را نیز دارد. تمامی آزمایش‌ها بر روی یک لپ‌تاپ با پردازنده -core i7 7700 HQ 2.8 Ghz و رم ۸ گیگ و سیستم عامل ویندوز ۱۰ انجام شده است. لینک دسترسی به کد مقاله^{۱۰} در سایت 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) پارامترهای شبیه‌سازی آورده شده است. در ادامه به بررسی نتایج حاصل از شبیه‌سازی با توجه به معیارهای مدنظر در این پژوهش می‌پردازیم. لازم به ذکر است که برای اثبات درستی نتایج شبیه‌سازی، هر سناریو به تعداد ۱۰ بار اجرا شده است و از تست آماری توزیع T-student در قالب مقادیر T-Value استفاده شده است که نتایج آن در شکل‌های ۶ تا ۸ آورده شده است.

¹⁰ <https://github.com/msadeghy69/gagwoalgorithm>

تأخیر به نسبت الگوریتم‌های دیگر به دست آورد. با توجه به شکل (۶) هر چه تعداد دستگاه‌های اینترنت اشیا افزایش می‌یابد به تبع وظایف ارسالی بیشتر شده و میانگین زمان اجرا با افزایش دستگاه‌های اینترنت اشیا افزایش پیدا می‌کند ولی در روش پیشنهادی همان‌طور که مشاهده می‌شود دلیل تنوع جواب‌ها در مقایسه با الگوریتم گرگ خاکستری باعث کاهش زمان اجرا به نسبت سایر الگوریتم‌ها می‌شود.

مصرف انرژی: مصرف انرژی سرورهای مه به ازای تعداد گره‌های اینترنت اشیا مختلف در شبکه در شکل (۷) آورده شده است. نتایج آورده شده در شکل (۷) بیانگر آن است که در روش **GAGWO** مصرف انرژی سرورهای مه نسبت به روش‌های پیشنهادی پیشین کاهش پیدا کرده است، زیرا در روش **GAGWO** بر خلاف روش‌های ژنتیک و گرگ خاکستری جواب‌ها گوناگونی بهتری دارند و الگوریتم‌های پیشنهادی تنوع جواب کمتری نسبت به الگوریتم **GAGWO** وجود دارد و باعث کاهش ارائه راه حل‌های بهینه در زمینه انرژی می‌شود. همچنین الگوریتم‌های پیشنهادی باعث گرفتار شدن در بهینه‌های محلی می‌شوند و بهینه‌ترین جواب که شامل کمترین مصرف انرژی می‌باشند را ارائه نمی‌دهند.

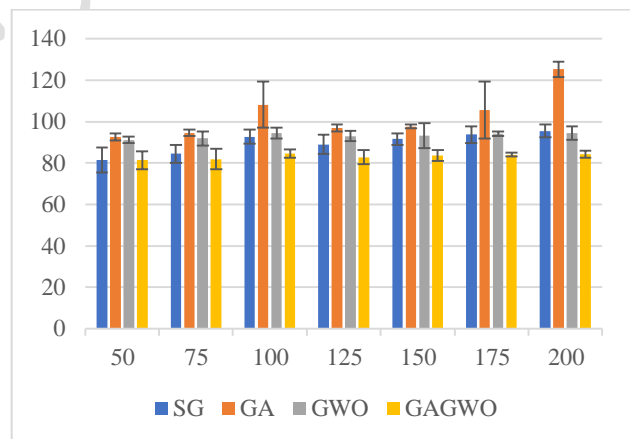


شکل (۷) مقایسه مصرف انرژی با تعداد دستگاه‌های اینترنت اشیا مختلف **زمان اتمام آخرین وظیفه:** شکل (۸) بیانگر زمان اتمام تمام آخرین وظیفه به ازای تعداد گره‌های مختلف اینترنت اشیا در شبکه می‌باشد. همان‌طور که مشاهده می‌شود با افزایش تعداد گره‌های

مقدار	پارامتر
۵	تعداد گره مه
۱۶	تعداد گره ابر
۱۰۰ ثانیه	مدت شبیه‌سازی
۱۰ وظیفه در هر ثانیه	نرخ تولید وظیفه
۵ مگابایت	اندازه وظیفه
50,75,100,125,150,175,200	تعداد دستگاه‌های اینترنت اشیا
50 Mbps	BW_C
10 Gbps	BW_C

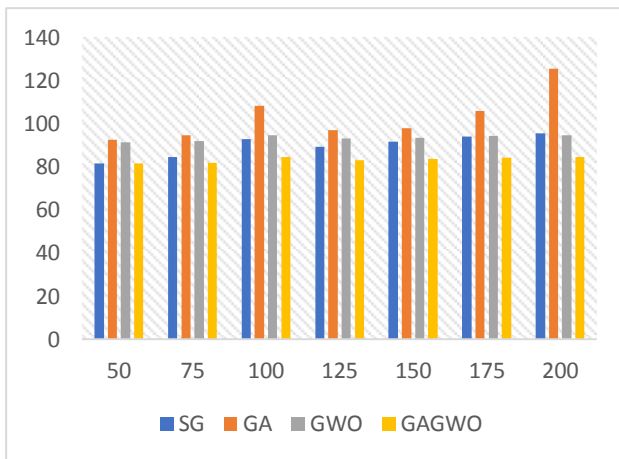
جدول ۳: معرفی پارامترهای شبیه‌ساز

میانگین زمان اجرا: در شکل (۶) میانگین زمان اجرا برای روش پیشنهادی و دیگر روش‌های مورد بررسی در این مقاله آورده شده است.



شکل (۶) مقایسه میانگین زمان اجرا با تعداد دستگاه‌های اینترنت اشیا مختلف

در شکل (۶) میانگین زمان اجرا برای روش پیشنهادی در مقایسه با روش مقاله نیمه حریصانه ژنتیک و گرگ خاکستری آورده شده است و همان‌طور که مشاهده می‌شود این مقدار در روش پیشنهادی در مقایسه با سایر روش‌های آورده شده کمتر است. علت آن است با توجه به اینکه زمان همگرایی در الگوریتم پیشنهادی کمتر از روش‌های دیگر است باعث می‌شود در تکرارهای یکسان الگوریتم پیشنهادی جواب‌های بهتری از لحاظ

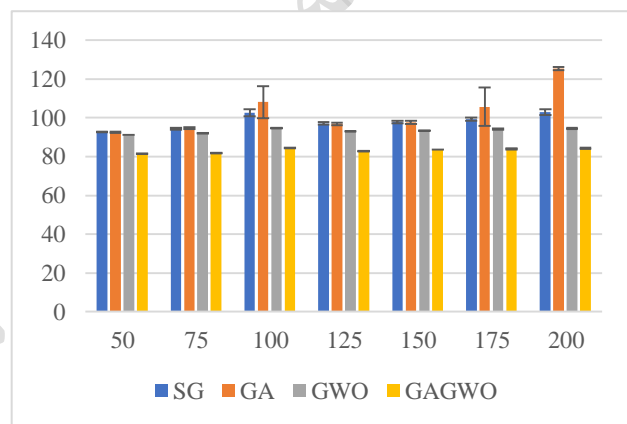


شکل (۹) مقایسه سرعت همگرایی با تعداد دستگاه‌های اینترنت اشیا مختلف

۷ نتیجه گیری

در این مقاله، یک الگوریتم اکتشافی کارآمد را برای حل مشکل تخصیص منابع در محیط‌های محاسباتی ابر و مه پیشنهاد شده است. ارائه فرمول ریاضی برای تخصیص منابع با هدف کاهش تأخیر و انرژی در حالی که معیارهای کیفیت خدمات سطح سرویس رعایت شود، انجام 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 حاصل شده 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 18.30% و 15.14% و 10.21% درصد بهبود می‌دهد. به عنوان کار آینده، ما قصد داریم الگوریتم‌های پیشنهادی را برای برنامه ریزی وظایف وابسته به یکدیگر بهبود دهیم. همچنین هدف ما ارزیابی عملکرد الگوریتم پیشنهادی با مجموعه داده‌های واقعی می‌باشد.

اینترنت اشیا، در روش GAGWO زمان اتمام آخرین وظیفه کاهش می‌یابد. کاهش اتمام آخرین وظیفه در الگوریتم پیشنهادی به دلیل آن است که در روش پیشنهادی از سه کروموزوم برتر برای عمل ترکیب استفاده می‌شود. این کار باعث می‌شود راه حل‌های بهتری ارائه شود زیرا تنوع جواب‌ها افزایش پیدا می‌کند. همان‌طور که در شکل (۸) مشاهده می‌شود زمانی که دستگاه‌های اینترنت اشیا افزایش پیدا می‌کند میزان زمان اتمام آخرین وظیفه در الگوریتم گرگ خاکستری به نسبت الگوریتم‌های ژنتیک و الگوریتم مقاله پایه کمتر می‌باشد.



شکل (۸) مقایسه زمان اتمام آخرین وظیفه با تعداد دستگاه‌های اینترنت اشیا مختلف

نرخ همگرایی مصرف انرژی: شکل (۹) نمودار مدت زمان اجرای الگوریتم پیشنهادی در مقایسه با سایر روش‌های مورد بررسی را به ازای تعداد گره‌های اینترنت اشیا مختلف در شبکه نشان می‌دهد. نتایج آورده شد در شکل (۹) بیانگر آن است که روش پیشنهادی سرعت همگرایی بیشتری نسبت به دیگر الگوریتم‌های مورد بررسی دارد.

- Computing," in *8th International Conference on Computing for Sustainable Global Development (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," in *Wireless Pers Commun* 102, 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 *Proceedings of the International Conference on Signal, Networks, Computing, and Systems*, vol 395. Springer, New Delhi ,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," in *Computational Methods for Differential Equations* 7.4 (Special Issue) (2019): 521-534. [In Persian]
- [13] F.Bonomi , R.Milito , J.Zhu and S.Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* .2012,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 International Conference on Edge Computing (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," in *Journal of systems architecture* 115
- [1] Z.Jalali , N.Mansouri and M.Khalouie, " Task scheduling in fog environment Challenges, tools & methodologies: A review," in *Computer Science Review* Rev. 48, C May2023 ,doi:10.1016/j.cosrev.2023.100550.
- [2] M.Khorasani , M.Ramezanzpour and R.Khorsand, "Energy Efficient Multi Path Routing Protocol in Internet of Things," in *Soft Computing Journal* 7.1 (2021):34-49, doi:10.22052/7.1.34 [In Persian].
- [3] M.Jameii , "Dynamic Service Provisioning in Fog Environment based on Learning Automata and Multi-objective Genetic Algorithm," in *Soft Computing Journal* (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," in *Soft Computing Journal* 8.1 (2021):43-57, doi:10.22052/8.1.43 [In Persian]
- [5] M.Islam , A.Kumar and Y.Hu, "Context-aware scheduling in Fog computing: A survey, taxonomy, challenges and future directions," in *Journal of Network and Computer Applications* 180 (2021) ,doi:10.1016/j.jnca.2021.103008.
- [6] 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," in *Journal of network and computer applications* 201(2022) ,doi:10.1016/j.jnca.2022.103333
- [7] M.Hussein and M.Mousa, "Efficient Task Offloading for IoT-Based Applications in Fog Computing Using Ant Colony Optimization," in *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

- (2021): 101996, 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," in *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," in *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1159-1166, May 2019, doi: 10.1109/JSAC.2019.2906793.
- [18] S. Ghanavati , J. Abawajy and D. Izadi, "An Energy Aware Task Scheduling Model Using Ant-Mating Optimization in Fog Computing Environment," in *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 2007-2017, 1 July-Aug. 2022, doi: 10.1109/TSC.2020.3028575.
- [19] M. Yang , et al, "A Multi-Objective Task Scheduling Method for Fog Computing in Cyber-Physical-Social Services." in *IEEE Access*, vol. 8, pp. 65085-65095, 2020, doi: 10.1109/ACCESS.2020.2983742.
- [20] A.Mirjalili , M.Mirjalili and A.Lewis, "Grey wolf optimizer." in *Advances in engineering software* 69 (2014) , doi:10.1016/j.advengsoft.2013.12.007.
- [21] K.Alwasel, et al, "IoTSim-Osmosis: A framework for modeling and simulating IoT applications over an edge-cloud continuum," in *Journal of Systems Architecture* 116 (2021): 101956, doi: 10.1016/j.sysarc.2020.101956