DOI: 10.22034/tjee.2023.56169.4613

A Mobility Aware Task Offloading Scheme Based On Ant Colony Optimization Algorithm In Software-Defined Fog Computing

Elham Barkhordari, Seyed Akbar Mostafavi*

Department of Computer Engineering, Faculty of Engineering, Yazd University, Yazd, Iran
E-mails: elham.barkhordari@stu.yazd.ac.ir; a.mostafavi@yazd.ac.ir

Short Abstract
Fog computing is a new paradigm which enables offloading IOT data (tasks) to the network devices. The aim of this approach is reducing the response time for delay-sensitive applications and improving the quality of service for users. This paper presents a task offloading scheme with taking advantages of the Software-Defind Networks. In this research a mixed integer linear programming (MILP) optimization model is presented with the aim of minimizing delay and mobility cost of things, which considers local computing, fog nodes participation, applications distribution and resource limitations. Whereas the presented mathematical model is Np-hard, a meta-heuristic algorithm based on the ant colony optimization is proposed by considering constraints of the mathematical model. The results obtained from evaluation of the proposed method is compared with the optimal value obtained from the mathematical model, random method and a heuristic algorithm presented in related works. The results of evaluation show that the delay and the total offloading cost in the proposed method are 22% and 28.75% higher than the optimal values, respectively. Also, the proposed method is capable to reduce the delay by 20% and reduce the migration cost of computing results by 40% compared to the heuristic method in state-of-the-art.

Keywords
Internet of Things, fog computing, task offloading, mobility of things, ant colony optimization, software-defined networking.

1- Short Introduction (4-5 lines)

Things in IOT have limited resources, and sending IOT's tasks to the cloud computing servers harms the quality of service. Therefore, tasks are offloaded to fog computing servers in order to improve the quality of service. Resource limitations of fog nodes is an important challenge of fog computing that has been studied in many articles such as [9],[10],[11],[12]. Also applications distributions and mobility of things are another significant challenges that need study, but a few articles have considered their such as [16],[17],[18]. This paper presents a offloading scheme to reduce delay and cost caused by mobility of things, which considers resource limitations of fog nodes, it's corporation and applications distribution compared to previous works.

2- Proposed Work and Methodology (including comprision, simulation/experimental results and discusion)

We formulated the task offloading problem as a MILP model and defined the objective function as minimizing the total computational delay and mobility cost for all things. Therefore, we calculated the total computational delay for local computing as processing delay and for fog computing as the sum of transmission delay, propagation delay, and processing delay. Then migration cost of results was defined as mobility cost, which was considered as the relative sum of the probability of moving the thing from cover of near fog node before receiving the results and expected value of distance of the thing from the processor fog node when receiving the results. Constraints of MILP model are bandwidth and flow rules capacity of fog nodes, the maximum tolerable delay for things and their maximum residual energy and computing the things' tasks by their Adjacent fog node and its neighbors that provide required application. The presented model is solvable with limited number of variables, so we proposed a meta-heuristic algorithm based on ant colony optimization which manages constraints by penalty functions and repairing unfeasibility. We use random selections in this algorithm inspired by the Simulated Annealing, and use global best pheromone concept to convergence management. Furthermore, we proposed a new function for global best pheromone injection which control convergence in local optimal. For evaluating the proposed algorithm, we considered two scenarios as evaluation with and without resource limitations, and we plotted delay and migration cost graphs and compared the proposed algorithm, optimal value, random offloading and soft-VAN method. Results show the proposed method provides comparable delay and migration cost with optimal value and is better than two other compared methods.

3- Conclusion (4-5 lines)

In this article, A MILP model was presented with the objective of minimizing delay and cost of mobility with constraints such as bandwidth and flow rules capacity of fog nodes, the maximum tolerable delay for things and their maximum residual energy, which considers local computing by things, computing by their Adjacent fog node and its neighbors that provide required application. A meta-heuristic algorithm based on ant colony optimization was proposed, which manages constraints by penalty functions and repairing unfeasibility. Results of simulations and evaluations show that the proposed method provides good delay and migration cost compared to the optimal values and related methods.

4- References (2-3 references)

[16] Z. L. X. W. and Y. L. D. Wang, "Mobility-Aware Task Offloading and Migration Schemes in Fog Computing Networks," IEEE Access, vol. 7, pp. 43356–43368, 2019.
[21] S. Misra and S. Bera, "Soft-VAN: Mobility-Aware Task Offloading in Software-Defined Vehicular Network," IEEE Trans Veh Technol, vol. 69, no. 2, pp. 2071–2078, 2020, doi: 10.1109/TVT.2019.2958740.

یک طرح برون‌سپاری وظایف آگاه از جابه‌جایی اشیاء مبتنی بر الگوریتم بهینه‌سازی کلونی مورچه در پردازش مه نرم‌افزار-محور

الهام برخورداری
دانشجوی دکترا، دانشکده مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران

سید اکبر مصطفوی
استادیار، دانشکده مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران

چکیده
محاسبات مه رویکرد نوینی است که با هدف کاهش زمان پاسخ کاربردهای حساس به تأخیر و بهبود ارائه خدمات به کاربران، امکان برون‌سپاری وظایف (داده‌ها) اینترنت اشیاء را به تجهیزات شبکه فراهم می‌کند. این مقاله یک طرح برون‌سپاری وظیفه به محیط مه با بهره‌مندی از مزایای شبکه‌های نرم‌افزار-محور ارائه می دهد. در این پژوهش یک مدل ریاضی بهینه‌سازی برنامه‌ریزی خطی عدد صحیح مختلط (MILP) با هدف کمینه‌سازی تأخیر و هزینه ناشی از جابه‌جایی اشیاء و با در نظر گرفتن پردازش محلی، مشارکت گره‌های مه، توزیع برنامه‌های کاربردی و محدودیت منابع گره‌های مه ارائه شده است. با توجه به اینکه مدل ریاضی ارائه شده در این مسئله ان‌پی-سخت است، یک الگوریتم فرا-ابتکاری مبتنی بر بهینه‌سازی کلونی مورچه و با در نظر گرفتن محدودیت‌های مدل ریاضی ارائه شده است. مقادیر حاصل از ارزیابی روش پیشنهادی با مقدار بهینه حاصل از مدل ریاضی، روش تصادفی و یک الگوریتم ابتکاری ارائه شده در کارهای مرتبط مقایسه شده است. نتایج حاصل نشان می‌دهد تأخیر و هزینه کل برون‌سپاری در روش پیشنهادی به ترتیب ۲۲٪ و ۲۸,۷۵٪ از مقادیر بهینه بیشتر است و روش پیشنهادی قادر به کاهش تأخیر به میزان ۲۰٪ و کاهش هزینه مهاجرت نتایج به میزان ۴۰٪ نسبت به روش ابتکاری مقایسه شده است.

کلمات کلیدی
اینترنت اشیاء، پردازش مه، برون‌سپاری وظایف، جابه‌جایی اشیاء، بهینه‌سازی کلونی مورچه، شبکه نرم‌افزارمحور.

نام نویسنده مسئول: دکتر سید اکبر مصطفوی
ایمیل نویسنده مسئول: a.mostafavi@yazd.ac.ir

تاریخ ارسال مقاله: ۱۴۰۲/۰۳/۰۸
تاریخ(های) اصلاح مقاله: ۱۴۰۲/۰۵/۲۰
تاریخ پذیرش مقاله: ۱۴۰۲/۰۶/۲۳

۱- مقدمه

اینترنت اشیاء مجموعه‌ای از تجهیزات ناهمگنی است که به شبکه اینترنت متصل هستند و بدون دخالت انسان به جمع‌آوری و ارسال داده می‌پردازند [۱]. تجهیزات اینترنت اشیاء شامل حسگرها، پردازنده‌ها، محرک‌ها و سخت‌افزارهای ارتباطی هستند و از آن‌ها برای جمع‌آوری، انتقال و پردازش اطلاعات محیطی استفاده می‌کنند [۲]. تجهیزات اینترنت اشیاء معمولاً کوچک و از نظر منابع انرژی و قابلیت‌های محاسباتی محدود هستند. این محدودیت‌ها برآورده نمودن نیازمندی‌ها و الزامات متنوع ترافیک اینترنت اشیاء را دشوار می‌کند. بنابراین نیاز است داده‌ها در بستر شبکه اینترنت به سرورهای پردازنده قوی‌تری جهت پردازش ارسال شوند [۳]. هدف پردازش ابری ارائه خدمات میزبانی شده از طریق اینترنت است. پردازش ابری توسط سرورهای قدرتمند مراکز داده مستقر در نقاط مختلف جهان انجام می‌شود[۴]. فاصله طولانی بین سرورهای ابری و تجهیزات اینترنت اشیاء باعث ایجاد تأخیر غیر قابل تحمل برای برخی از کاربرد های حساس به تأخیر اینترنت اشیاء و افت کیفیت سرویس کاربران می‌شود [۵]. پردازش مه و پردازش لبه رویکردهای توزیع شده‌ای هستند که محاسبات را جهت ارائه سریع‌تر خدمات به کاربران به لبه شبکه گسترش می‌دهند. محاسبات لبه و مه از منظر محل انجام محاسبات متفاوت هستند. پردازش مه مابین پردازش ابری و پردازش لبه قرار می‌گیرد. پردازش مه توسط تجهیزات شبکه از ابر تا شبکه محلی انجام می‌شود، درحالی‌که محاسبات لبه روی دستگاه‌های هوشمند متصل به حسگرها انجام می‌شود [۶]. پردازش داده‌های اینترنت اشیاء توسط تجهیزات مه و لبه که منابع محدودی دارند، عملکرد شبکه را از نظر تأخیر، متعادل‌سازی بار شبکه و استفاده از منابع بهبود می‌بخشد. تکنیک ارسال داده‌های جمع‌آوری شده توسط اشیاء به یک پردازشگر قوی راه دور در ادبیات، برون‌سپاری وظیفه نامیده می‌شود [۷]. در پردازش مه، تخصیص منابع به وظایف مختلف جهت اطمینان از کارایی مناسب به دلیل محدودیت منابع یک چالش اساسی است. بنابراین به روش‌های بهینه‌سازی نیاز است تا کیفیت خدمات مورد نیاز میلیون‌ها وظیفه آماده برون‌سپاری تضمین شود [۲].

در تحقیقات اخیر یک طرح برون‌سپاری وظایف بر اساس تأخیر کل مبتنی بر الگوریتم بهینه‌سازی کلونی مورچه در سال ۲۰۲۱ در مقاله [۸] ارائه شده است. در این پژوهش، وظایف متناسب با تأخیر ارتباط و پردازش به یکی از گره‌های مه در دسترس یا فضای ابر ارسال می‌شود. طرح برون‌سپاری دیگری مبتنی بر الگوریتم فرا-ابتکاری بهینه‌سازی ازدحام ذرات با در نظر گرفتن نهان سازی سرویس‌ها در پردازش مه در سال ۲۰۲۲ در مقاله [۹] ارائه شده است. یک مدل ریاضی غیرخطی در مقاله [۱۰] برای برون‌سپاری وظایف جهت کمینه

نمودن میانگین تأخیر پردازش و انرژی مصرفی اشیاء با در نظر گرفتن محدودیت‌های منابع پردازشی ارائه شده است. با توجه به محدب نبودن مدل ریاضی ارائه شده از الگوریتم ژنتیک و الگوریتم زنبورها برای تقریب مدل ریاضی استفاده شده است. در سال ۲۰۲۱ در پژوهش [۱۱] یک طرح برون‌سپاری با در نظر گرفتن اولویت وظایف و با هدف کمینه نمودن تأخیر کل ارائه شده است. در این مقاله محدودیت‌های نرخ انتقال، منابع پردازشی و بیشینه تأخیر قابل تحمل در نظر گرفته‌شده است. در سال ۲۰۲۰ یک مدل ریاضی برای برون‌سپاری وظایف با هدف کاهش مصرف انرژی اشیاء و با در نظر گرفتن محدودیت‌های تأخیر و منابع محاسباتی گره‌های مه ارائه شده است که عدم قطعیت خطای کانال ارتباطی اشیاء و گره‌های مه را نیز در نظر می‌گیرد [۱۲].

رشد درخواست‌های ارسال شده به گره‌های مه و محدودیت منابع آن‌ها باعث افت کارایی گره‌های مه در سرویس‌دهی به کاربران و افزایش تأخیر کل پردازش وظیفه می‌شود. این موضوع می‌تواند در شرایطی باشد که سایر گره‌های مه منابع پردازشی آزاد مناسبی برای پردازش درخواست‌های جدید داشته باشند. بنابراین به‌جای ارسال وظیفه به فضای ابر و افزایش تأخیر دریافت نتایج می‌توان وظیفه را به یک گره مه همکار ارسال نمود [۱۳]. مقاله [۱۴] برون‌سپاری وظایف را به‌صورت مشارکت پردازش محلی، ابر و مه در نظر گرفته است. در این پژوهش امکان ارسال وظایف از یک گره مه به یک گره مه مشخص دیگر متناسب با توان پردازشی گره‌های مه در نظر گرفته‌شده است. طرح مشارکتی دیگری در سال ۲۰۱۹ در مقاله [۱۵] ارائه شده است که پس از بارگیری وظیفه توسط گره مه، متناسب با بار پردازشی موجود امکان ارسال همه یا بخشی از وظیفه به سایر گره‌های مه وجود دارد.

درصورتی‌که یک شئ وظیفه‌ای را به یک گره مه برای پردازش ارسال کند و پیش از دریافت نتایج از محدوده گره مه خارج شود، نتایج پردازش جهت تداوم ارائه خدمات باید به محل جدید شئ ارسال شود. ارسال نتایج پردازش به محل جدید کاربر منجر به افزایش زمان پاسخ، بار شبکه و هزینه‌های مسیریابی در شبکه می‌شود. بنابراین تعیین نحوه پردازش وظایف اشیاء و گره مه پردازشگر متناسب با کاهش هزینه مهاجرت نتایج از موضوعات مهمی است که باید در برون‌سپاری وظایف در نظر گرفت که تعداد مقالات کمی به این موضوع پرداخته‌اند [۱۶]. یک طرح زمان‌بندی جهت پردازش وظایف و تخصیص منابع با در نظر گرفتن جابه‌جایی در فضای مه در سال ۲۰۲۱ در مقاله [۱۷] ارائه شده است. مدل‌سازی جابه‌جایی در این پژوهش بر اساس حرکت عابران پیاده است. در این مقاله یک الگوریتم ابتکاری جهت برون‌سپاری وظایف با در نظر گرفتن تغییر گره‌های مه در دسترس و میزان منابع محاسباتی آن‌ها در هر شکاف زمانی و حرکت کاربران ارائه شده است. طرح برون‌سپاری وظایف دیگری با در نظر گرفتن هزینه مهاجرت نتایج ناشی از جابه‌جایی کاربران در مقاله [۱۶] ارائه شده است. هزینه مهاجرت نتایج در این پژوهش با محاسبه احتمال جابه‌جایی شئ از محدوده گره مه سرویس‌دهنده پیش از دریافت نتیجه و متناسب با اندازه وظیفه برون‌سپاری شده در نظر گرفته شده است. در سال ۲۰۱۸ طرح برون‌سپاری وظایفی با در نظر گرفتن ناهمگونی نیازمندی‌های تأخیر و کاهش پراکندگی تأخیر پردازش وظایف با استفاده از روش عدالت بیشینه-کمینه در پژوهش [۱۸] ارائه شده است که به افزایش هزینه‌های شبکه در اثر جابه‌جایی و نیاز به مهاجرت نتایج پردازش به گره‌های مه دیگر نیز اشاره دارد.

به‌منظور تخصیص بهینه منابع شبکه و ارائه خدمات مناسب به اشیاء نیاز به اطلاعات جامعی از شبکه است [۱۹]. بنابراین یک کنترل‌کننده مرکزی می‌تواند با در نظر گرفتن اطلاعات جامع از وضعیت انرژی باقی‌مانده اشیاء، درخواست سایر اشیاء متقاضی برون‌سپاری و وضعیت منابع گره‌های مه یک طرح مناسب برای برون‌سپاری وظایف ارائه کند. شبکه نرم‌افزار-محور پارادایمی است که امکان استفاده بهینه از منابع محدود شبکه و مدیریت انعطاف‌پذیر شبکه را با جداسازی لایه کنترل از لایه داده و برنامه‌ریزی پویای نحوه ارسال داده‌ها فراهم آورده است [۲۰]. در سال ۲۰۲۰ در مقاله [۲۱] یک طرح برون‌سپاری وظایف با بهره‌گیری از شبکه‌های نرم‌افزار-محور و با در نظر گرفتن جابه‌جایی اشیاء ارائه شده است. در این پژوهش ابتدا مسئله به‌صورت یک مدل ریاضی برنامه‌ریزی غیرخطی جهت کمینه نمودن تأخیر کل و انرژی مصرفی اشیاء با در نظر گرفتن محدودیت‌های انرژی، تأخیر کل و فضای قواعد جریان گره‌های مه فرمول‌بندی شده است. سپس یک الگوریتم ابتکاری حریصانه جهت تعیین نحوه پردازش، گره مه پردازنده، مسیر ارسال وظایف و مسیر ارسال نتیجه پردازش پس از پیش‌بینی مکان جدید شئ ارائه شده است.

به علت محدودیت منابع گره‌های مه و تنوع بالای برنامه‌های کاربردی امکان ارائه تمامی برنامه‌های اینترنت اشیاء توسط تمامی گره‌های مه نیست. بنابراین هر گره مه، گروهی از سرویس‌های کاربردی را متناسب با سیاست‌هایی چون میزان منابع، تکرار درخواست کاربران و موقعیت جغرافیایی ارائه می‌دهد. جابه‌جایی اشیاء و انتقال یک شئ از محدوده یک گره مه به گره مه دیگر نیز باعث افزایش احتمال فقدان سرویس مورد نیاز اشیاء توسط برخی از گره‌های مه می‌شود [۲۰]. نحوه توزیع برنامه‌های کاربردی روی گره‌های مه، از دیگر مواردی است که باید در پردازش وظایف در فضای مه در نظر گرفته شود.

در این مقاله یک طرح برون‌سپاری وظایف ارائه شده است که امکان پردازش وظایف به‌صورت محلی توسط اشیاء و پردازش مه با مشارکت گره‌های مه را در نظر گرفته است و از قابلیت دید جامع شبکه‌های نرم‌افزار-محور استفاده می‌کند. طرح پیشنهادی به‌صورت یک مدل ریاضی بهینه‌سازی برنامه‌ریزی خطی عدد صحیح مختلط[۱] (MILP) مدل‌سازی شده است. هدف مدل ارائه شده کمینه‌سازی تأخیر کل و هزینه ناشی از جابه‌جایی اشیاء است. در این مدل‌سازی محدودیت‌های پردازش وظایف توسط گره‌های مه ارائه دهنده سرویس، همکاری گره‌های مه همسایه، میزان منابع ارتباطی در دسترس، حداکثر تأخیر قابل تحمل، سطح انرژی اشیاء و میزان فضای قواعد جریان در دسترس گره‌های مه در نظر گرفته شده است. با توجه به اینکه مدل‌سازی این مسئله غیر محدب و ان‌پی-سخت (NP-hard) است، متناسب با تابع هدف و محدودیت‌های مسئله یک الگوریتم فرا-ابتکاری مبتنی بر بهینه‌سازی کلونی مورچه (ACO) ارائه شده است. دستاوردهای اصلی این مقاله به شرح زیر می‌باشد:

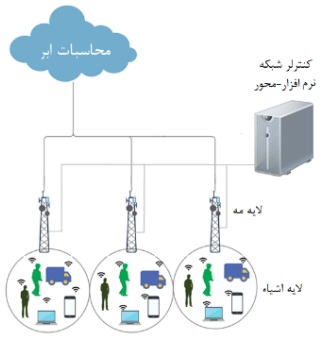
- ✓ مدل‌سازی پارامترهای مسئله ازجمله تأخیر کل و انرژی مصرفی اشیاء در هر دو حالت پردازش محلی و پردازش مه متناسب با معماری مسئله
- ✓ مدل‌سازی هزینه ناشی از جابه‌جایی اشیاء متناسب با هزینه مهاجرت نتایج پردازش و به‌صورت مجموع نسبی امید ریاضی فاصله شئ از گره مه پردازنده در هنگام دریافت نتایج و احتمال جابه‌جایی شئ از محدوده گره مه مجاور پیش از دریافت نتیجه
- ✓ ارائه یک مدل ریاضی MILP برای طرح پیشنهادی با هدف کمینه نمودن تأخیر کل و هزینه مهاجرت نتایج و با در نظر گرفتن محدودیت‌های پهنای باند و فضای قواعد جریان گره‌های مه، حداکثر تأخیر قابل تحمل، پردازش وظایف توسط گره‌های مه دارای برنامه کاربردی درخواست شده، مشارکت همسایگان گره مه مجاور و بیشینه انرژی باقی مانده اشیاء
- ✓ ارائه یک الگوریتم فرا-ابتکاری مبتنی بر بهینه‌سازی کلونی مورچه با مدیریت محدودیت‌های مسئله از طریق اعمال جریمه در صورت بروز تخطی و تصحیح حالات دارای تخطی

[۱] Mixed Integer Linear Programming

در ادامه این مقاله در بخش دوم معماری و مدل سیستم تشریح شده است. فرمول‌بندی مسئله در بخش سوم بیان شده است و در بخش چهارم به تشریح الگوریتم فرا-ابتکاری ارائه شده پرداخته شده است. در بخش پنجم کارایی الگوریتم پیشنهادی در مقایسه با پاسخ بهینه و روش‌های مرتبط بررسی شده و نهایتاً در بخش ششم جمع بندی و نتیجه‌گیری ارائه شده است.

۲- معماری و مدل سیستم

معماری در نظر گرفته شده مطابق شکل ۱ شامل گره‌های اینترنت اشیاء، گره‌های مه و ابر است. هر شئ پس از تولید وظیفه، باید یک بسته شامل مشخصات وظیفه و سطح انرژی باقی‌مانده خود را به گره مه‌ای که در محدوده پوشش آن قرار دارد ارسال کند. سپس شئ مطابق نتیجه دریافتی از گره مه، وظیفه را به‌صورت محلی پردازش و یا به گره مه مجاورش ارسال می‌کند. هر شئ در مجاورت و ارتباط با یک گره مه است.



شکل ۱- معماری شبکه متشکل از اشیاء، گره‌های مه، محاسبات ابر و کنترلر شبکه نرم‌افزار-محور

وظایف بارگیری شده توسط یک گره می‌تواند توسط همان گره یا یکی از همسایگان آن یا فضای ابری پردازش شود. کنترل‌کننده شبکه نرم‌افزار-محور تصمیم می‌گیرد که نحوه پردازش وظیفه چگونه باشد و همواره منابع گره‌های مه و همبندی شبکه را پایش می‌کند. درصورتی‌که پیچیدگی محاسباتی وظایف و حداکثر تأخیر قابل تحمل آن‌ها از حد آستانه Π_{cloud} بیشتر باشد، وظایف به ابر ارسال می‌شوند. در این پژوهش تمرکز بر برون‌سپاری وظایفی است که پیچیدگی محاسباتی و حداکثر تأخیر قابل تحمل آن‌ها از حد آستانه Π_{cloud} کمتر است و صرفاً به ایجاد تعادل بین پردازش محلی و پردازش مه و تعیین گره مه پردازنده پرداخته شده است. همانند پژوهش [۲۲] جهت دسترسی چندگانه به کانال گره‌های مه روش دسترسی چندگانه با تقسیم فرکانس متعامد[۲] (OFDMA) در نظر گرفته شده است. گره‌های مه دارای ماژول اوپن‌فلو هستند و صفحه کنترل در قالب یک کنترل‌کننده متمرکز قرار گرفته است.

مجموعه اشیاء به‌صورت $I = \{1,2,3,\dots,|I|\}$ در نظر گرفته شده است. یک شئ در هر بازه زمانی یک وظیفه جهت پردازش دارد. وظیفه مربوط به شئ i با نماد θ_i نمایش داده می‌شود. مجموعه $\theta_i = \{s_i, c_i, a_i, b_i^{min}, d_i^{max}\}, \forall i \in I$ مشخصات وظیفه θ_i را نمایش می‌دهد. نماد s_i معادل اندازه وظیفه، c_i نمایانگر تعداد سیکل مورد نیاز در پردازنده جهت پردازش وظیفه، a_i شناسه برنامه کاربردی مورد نیاز جهت پردازش وظیفه، b_i^{min} حداقل پهنای باند مورد نیاز جهت برون‌سپاری وظیفه و نماد d_i^{max} بیانگر بیشینه تأخیر قابل تحمل برای وظیفه است. مجموعه سرویس‌های کاربردی به‌صورت $A = \{1,2,3,\dots,|A|\}$ است که متناسب با سیاست‌های کنترلر زیرمجموعه‌ای از آن‌ها در هر گره مه در دسترس می‌باشد. مجموعه گره‌های مه به‌صورت $F = \{1,2,3,\dots,|F|\}$ تعریف شده است. مجموعه $U_{f_j} = \{1,2,\dots,|U_{f_j}|\}, \forall f_j \in F, U_{f_j} \subseteq A$ سرویس‌های کاربردی گره مه f_j را نشان می‌دهد. نماد f_i^{near} نشان دهنده گره مه مجاور شئ i است. مجموعه $N_{f_j} = \{1,2,\dots,|N_{f_j}|\}, \forall f_j \in S_F$ گره‌های مه همسایه گره مه f_j را نشان می‌دهد.

۲-۱- تأخیر

تأخیر در پردازش محلی معادل زمان پردازش وظیفه توسط پردازنده شئ در نظر گرفته شده است که به‌صورت رابطه (۱) محاسبه می‌شود [۲۳]. نماد c_i تعداد سیکل‌های مورد نیاز در پردازنده جهت پردازش θ_i و v_i فرکانس پردازشگر شئ i است.

$$D_i^{local} = \frac{c_i}{v_i}, \forall i \in I \quad (۱)$$

جهت پردازش وظیفه در فضای مه ابتدا باید وظیفه از شئ i به گره مه f_i^{near} ارسال شود. تأخیر انتقال وظیفه θ_i از شئ i به گره مه مجاور آن به‌صورت معادله (۲) محاسبه می‌شود.

$$D_{i,f_i^{near}}^{tr} = \frac{s_i}{r_{i,f_i^{near}}}, \forall i \in I, f_i^{near} \in F \quad (۲)$$

در معادله فوق $r_{i,f_i^{near}}$ نرخ انتقال کانال بین شئ i و گره مجاور آن است. محاسبه نرخ انتقال از طریق معادله (۳) انجام می‌شود. نماد $SNR_{i,f_i^{near}}$ نشان دهنده قدرت سیگنال به نویز کانال شئ و گره مه مجاور است که مطابق قانون شانون با معادله (۴) محاسبه می‌شود [۲۴].

$$r_{i,f_i^{near}} = b_i^{min} \times SNR_{i,f_i^{near}}, \forall i \in I, f_i^{near} \in F \quad (۳)$$

$$SNR_{i,f_i^{near}} = log_2 1 + \left(\frac{p_i \times \varsigma_{i,f_i^{near}} \times \xi_{i,f_i^{near}}}{I_{i,f_i^{near}} + b_i^{min} \times N_0}\right), \forall i \in I, f_i^{near} \in F \quad (۴)$$

p_i نماد قدرت انتقال[۳] شئ i، $\varsigma_{i,f_i^{near}}$ فاکتور افت مسیر[۴] و $\xi_{i,f_i^{near}}$ عامل سایه[۵] مربوط به لینک بی‌سیم می‌باشد. نماد N_0 و $I_{i,f_i^{near}}$ نشان‌دهنده چگالی طیفی قدرت نویز و قدرت تداخل است. درصورتی‌که وظیفه θ_i توسط یکی از همسایگان f_i^{near} پردازش شود، تأخیر انتقال بین دو گره مه مطابق رابطه (۵) محاسبه می‌شود. $r_{f_i^{near},f_j}$ نماد نرخ انتقال لینک سیمی بین دو گره مه است که به‌صورت تساوی (۶) محاسبه می‌شود.

$$D_{f_i^{near},f_j}^{tr} = \frac{s_i}{r_{f_i^{near},f_j}}, \forall i \in I, f_i^{near} \in F, f_j \in N_{f_i^{near}} \quad (۵)$$

$$r_{f_i^{near},f_j} = b_i^{min} \times SNR_{f_i^{near},f_j}, \forall i \in I, f_i^{near} \in F, f_j \in N_{f_i^{near}} \quad (۶)$$

تأخیر انتشار بین دو گره مه و گره مه و شئ مطابق تساوی (۷) محاسبه می‌شود. مطابق این تساوی تأخیر انتشار حاصل تقسیم فاصله بین دو گره و سرعت انتشار داده است [۱۵].

$$D^{prop} = \frac{Distance}{PropagationSpeed} \quad (۷)$$

صف گره‌های مه به‌صورت M/M/1 در نظر گرفته شده است. نرخ ورود به هر گره مه به‌صورت معادله (۸) محاسبه می‌شود. برای هر گره مه f_j پارامتر μ_{f_j} نرخ سرویس و پارامتر λ_{f_j} نرخ ورود به آن است. پارامتر T زمان کل و پارامتر x_{i,f_j} یک پارامتر دودویی است که مشخص می‌کند که وظیفه θ_i به گره مه f_j

[۲] Orthogonal Frequency-Division Multiple Access

[۳] Transmission Power

[۴] Path Loss Factor

[۵] Shadowing Factor

برون‌سپاری شده یا نشده است [۲۱]. تأخیر صف برای وظیفه θ_i در گره مه f_j به‌صورت معادله (۹) محاسبه می‌شود [۲۵].

$$\lambda_{f_j} = \frac{1}{T}\sum_{i\in I}^{i}\sum_{f_j\in F}^{j} x_{i,f_j} \tag{۸}$$

$$D_{f_j}^{que} = \frac{\lambda_{f_j}}{\mu_{f_j}-\lambda_{f_j}}, \forall f_j \in N_{f_i^{near}} \cup f_i^{near}, f_i^{near} \in F \tag{۹}$$

محاسبه تأخیر پردازش وظیفه θ_i در گره مه f_j به‌صورت رابطه (۱۰) است. v_{f_j} نشان‌دهنده‌ی فرکانس پردازنده گره مه می‌باشد [۲۴].

$$D_{i,f_j}^{exe} = \frac{c_i}{v_{f_j}}, \forall i \in I, f_j \in N_{f_i^{near}} \cup f_i^{near}, f_i^{near} \in F \tag{۱۰}$$

حجم نتیجه پردازش به نسبت وظیفه ورودی بسیار کمتر است و پهنای باند بارگذاری داده در گره‌های مه بسیار بالا است. بنابراین تأخیر انتقال مربوط به بارگیری نتایج در نظر گرفته نشده است [۱۹، ۲۴]. تأخیر کل پردازش وظیفه θ_i در گره مه f_j با معادله (۱۱) محاسبه می‌شود. پارامتر باینری ρ_{i,f_j} که در رابطه (۱۱) آورده شده است، مطابق تساوی (۱۲) مشخص می‌کند که گره مه پردازنده وظیفه θ_i گره مجاور شئ i می‌باشد یا خیر.

$$D_{i,f_j}^{fog} = D_{i,f_i^{near}}^{tr} + D_{i,f_i^{near}}^{prop} + \left(D_{f_i^{near},f_j}^{tr} + D_{f_i^{near},f_j}^{prop}\right) \times \rho_{i,f_j} + D_{i,f_j}^{exe} + D_{f_j}^{que}, \forall i \in I, f_j \in N_{f_i^{near}} \cup f_i^{near}, f_i^{near} \in F \tag{۱۱}$$

$$\rho_{i,f_j} = \begin{cases} 1 & if\ f_j \neq f_i^{near} \\ 0 & otherwise \end{cases} \tag{۱۲}$$

۲-۲- انرژی مصرفی شئ

معادله (۱۳) میزان انرژی مصرفی شئ i در پردازش وظیفه θ_i را نشان می‌دهد که در آن e_i میزان مصرف انرژی در هر سیکل پردازنده است [۲۴]. انرژی مصرفی شئ در پردازش مه مربوط به ارسال وظیفه به گره مه مجاور است. معادله (۱۴) بیانگر انرژی مصرفی شئ i در پردازش مه است [۲۶].

$$E_i^{local} = e_i \times c_i, \forall i \in I \tag{۱۳}$$

$$E_{i,f_i^{near}}^{fog} = D_{i,f_i^{near}}^{tr} \times p_i, \forall i \in I, f_i^{near} \in F \tag{۱۴}$$

۲-۳- هزینه ناشی از جابه‌جایی اشیاء

اشیاء متحرک می‌توانند از محدوده پوشش یک گره مه خارج و به محدوده گره مه دیگری وارد شوند. بنابراین پس از اتمام ارسال وظیفه به گره مه مجاور احتمال این وجود دارد که شئ پیش از دریافت نتیجه از محدوده پوشش گره مه جاری خارج شود. فرض بر این است که شئ قبل از ارسال کامل وظیفه به گره مه مجاور از محدوده آن خارج نمی‌شود. خروج شئ از محدوده گره مجاورش در چنین شرایطی باعث افزایش تأخیر انتشار، هزینه و تأخیر مسیریابی در ارسال نتیجه پردازش به محل جدید شئ می‌شود. در این پژوهش حرکت شئ و انتقال از یک سلول به سلول دیگر تصادفی و به‌صورت یک فرآیند پواسون در نظر گرفته شده است. زمان انتظار برای وقوع رخداد در فرآیند پواسون دارای توزیع نمایی است [۲۷]. مشابه کارهای [۱۶، ۲۸] با بهره‌گیری از تابع چگالی احتمال توزیع نمایی، تابع چگالی احتمال زمان اقامت شئ i در محدوده پوشش گره مه f_i^{near} مطابق معادله (۱۵) ارائه شده است. نماد $\Delta_{i,f_i^{near}}$ بیانگر میانگین زمان اقامت شئ i در محدوده گره مه f_i^{near} می‌باشد. کنترلر با ردیابی حرکت کاربران، میانگین زمان اقامت شئ در محدوده هر گره مه را محاسبه می‌کند. درصورتی‌که وظیفه شئ i برای پردازش به گره مه f_j برون‌سپاری شده باشد، احتمال خروج شئ i از محدوده گره مه مجاور خود قبل از دریافت نتیجه (احتمال نیاز به مهاجرت نتیجه) به‌صورت عبارت (۱۶) محاسبه می‌شود.

$$P_{i,f_i^{near}}(t) = \frac{1}{\Delta_{i,f_i^{near}}} \times e^{-\frac{t}{\Delta_{i,f_i^{near}}}}, \forall i \in I, f_i^{near} \in F \tag{۱۵}$$

$$P_i\left(\Delta_{i,f_i^{near}} < D_{i,f_j}^{fog}\right) = \int_0^{D_{i,f_j}^{fog}} \frac{1}{\Delta_{i,f_i^{near}}} \times e^{-\frac{t}{\Delta_{i,f_i^{near}}}} \times dt = 1 - e^{-\frac{D_{i,f_j}^{fog}}{\Delta_{i,f_i^{near}}}}, \forall i \in I, f_i^{near} \in F, f_j \in N_{f_i^{near}} \cup f_i^{near} \tag{۱۶}$$

درصورتی‌که وظیفه θ_i به گره مه f_j برون‌سپاری شود و شئ i در زمان دریافت نتیجه در محدوده پوشش گره مه $f_k \in S_F$ باشد، امید ریاضی فاصله شیء i از گره مه f_j در زمان دریافت نتیجه پردازش به‌صورت تساوی (۱۷) محاسبه می‌شود. ماتریس H کوتاه‌ترین فاصله هر گره مه با گره‌های مه دیگر را برحسب تعداد گام نشان می‌دهد. احتمال اینکه شیء i در هنگام دریافت نتیجه پردازش در محدوده هر یک از گره‌های مه باشد را با ماتریس TR_i نشان می دهیم. این ماتریس به‌عنوان ورودی در نظر گرفته شده است که کنترلر با ردیابی حرکت کاربران می‌تواند آن را متناسب با سرعت، میانگین زمان اقامت و جهت حرکت کاربران تولید و ارائه کند.

$$E\left(d_{i,f_j}\right) = \sum_{f_k \in F} TR_i[f_k] \times H[f_k][f_j], \forall i \in I, f_j \in N_{f_i^{near}} \cup f_i^{near} \tag{۱۷}$$

هزینه ناشی از جابه‌جایی برای شئ i در صورت برون‌سپاری وظیفه θ_i در گره مه f_j به‌صورت معادله (۱۸) است. پارامتر HC هزینه مربوط به ارسال نتیجه پردازش وظایف به ازای هر گام می‌باشد. HC متناسب با تأخیر انتشار و تأخیر و هزینه مسیریابی در هر گام است. مطابق رابطه (۱۸) هرچه احتمال جابه‌جایی شئ از محدوده گره جاری پیش از دریافت نتیجه کمتر باشد و گره پردازنده وظیفه در صورت جابه‌جایی شئ و انتقال به محدوده پوشش گره مه دیگر فاصله کمتری با آن داشته باشد، هزینه ناشی از جابه‌جایی اشیاء و مهاجرت نتایج پردازش کمتر خواهد شد. پارامتر α برابر ۱۰ و β برابر ۱ در نظر گرفته شده است.

$$MobilityCost_{i,f_j} = \alpha \times P_i\left(\Delta_{i,f_i^{near}} < D_{i,f_j}^{fog}\right) + \beta \times E\left(d_{i,f_j}\right) \times HC, \forall i \in I, f_j \in N_{f_i^{near}} \cup f_i^{near}, f_i^{near} \in F \tag{۱۸}$$

۳- فرمول‌بندی مسئله

تصمیم‌گیری برای نحوه پردازش وظایف بر اساس هزینه پردازش وظایف در فضای مه و پردازش محلی است. رابطه (۱۹) نشان می‌دهد که هزینه پردازش محلی وظیفه شئ i معادل تأخیر پردازش محلی است. متغیر دودویی σ_i نشان می‌دهد که وظیفه θ_i به‌صورت محلی پردازش می‌شود یا خیر. اگر وظیفه به‌صورت محلی پردازش شود، $\sigma_i = 1$ می‌باشد. در غیر این صورت $\sigma_i = 0$ است.

$$LocalCost_i = D_i^{local} \times \sigma_i, \forall i \in I \tag{۱۹}$$

هزینه پردازش مه به‌صورت رابطه (۲۰) است. تأخیر کل و هزینه مهاجرت نتایج دو پارامتر مؤثر بر این هزینه می‌باشند. درصورتی‌که وظیفه θ_i توسط گره مه f_j پردازش شود، متغیر $x_{i,f_j} = 1$ است. در غیر این صورت $x_{i,f_j} = 0$ است. دو پارامتر A و B برابر ۰/۵ در نظر گرفته شده است.

$$FogCost_{i,f_j} = \left(A \times D_{i,f_j}^{fog} + B \times MobilityCost_{i,f_j}\right) \times x_{i,f_j}, \forall i \in I, f_j \in N_{f_i^{near}} \cup f_i^{near}, f_i^{near} \in F \tag{۲۰}$$

جهت تعیین پاسخ بهینه برای نحوه پردازش وظایف، مدل ریاضی P_1 تعریف شده است.

$$P_1: \tag{۲۱}$$

$$Minimize_{\sigma_i, x_{i,f_j}} \sum_{i\in I}(LocalCost_i + \sum_{f_j\in F} FogCost_{i,f_j})$$

$$s.t.$$

$$\sigma_i + \sum_{f_j\in F} x_{i,f_j} = 1, \forall i \in I \quad (۲۲)$$

$$\sum_{f_j\in F} s_{i,f_j} \times x_{i,f_j} = 0, \forall i \in I \quad (۲۳)$$

$$D_i^{local} \times \sigma_i + (\sum_{f_j\in F} D_{i,f_j}^{fog} \times x_{i,f_j}) \le d_i^{max}, \forall i \in I \quad (۲۴)$$

$$E_i^{local} \times \sigma_i + (\sum_{f_j\in F} E_{i,f_i^{near}}^{fog} \times x_{i,f_j}) \le e_i^{max}, \forall i \in I \quad (۲۵)$$

$$\sum_{i\in I} b_i^{min} \times (1-\sigma_i) \le B_{f_i^{near}}^{max}, \forall f_i^{near} \in F \quad (۲۶)$$

$$\sum_{i\in I} b_i^{min} \times x_{i,f_j} \le B_{f_j}^{max}, \forall f_j \in F \quad (۲۷)$$

$$\sum_{i\in I} R^i \times (1-\sigma_i) \le R_{f_i^{near}}^{max}, \forall f_i^{near} \in F \quad (۲۸)$$

$$\sum_{i\in I} R^i \times x_{i,f_j} \le R_{f_j}^{max}, \forall f_j \in F \quad (۲۹)$$

$$\sigma_i, x_{i,f_j} = \{0,1\}, \forall i \in I, f_j \in F \quad (۳۰)$$

رابطه (۲۱) یک تابع خطی با هدف کمینه‌سازی هزینه کل پردازش وظایف است. تساوی (۲۲) نشان می‌دهد که هر وظیفه تنها در یکی از حالت‌های پردازش محلی و پردازش مه قابل پردازش است. همچنین در صورت پردازش وظیفه در مه صرفاً یک گره مه می‌تواند وظیفه را پردازش کند [۲۹-۳۱]. رابطه (۲۳) نشان می‌دهد که در صورت انتخاب پردازش مه، وظیفه باید صرفاً توسط گره مه‌ای پردازش شود که گره مجاور شئ یا همسایگانش است و برنامه کاربردی مورد نیاز وظیفه را ارائه می‌دهد. پارامتر s_{i,f_j} در رابطه (۲۳) یک پارامتر دودویی است که مطابق تساوی (۳۱) تعریف شده است.

$$s_{i,f_j} = \begin{cases} 0 & if\ f_j \in N_{f_i^{near}} \cup f_i^{near}, a_i \in U_{f_j} \\ 1 & otherwise \end{cases} \quad (۳۱)$$

رابطه (۲۴) بیانگر این محدودیت است که تأخیر کل نباید از بیشینه تأخیر قابل تحمل برای وظیفه بیشتر باشد. نامساوی (۲۵) مشخص می‌کند که انرژی مصرفی شئ در پردازش وظیفه نباید از سطح انرژی باقی‌مانده شئ بیشتر باشد. نماد e_i^{max} بیشینه انرژی باقی‌مانده شئ i است. برای پردازش وظیفه θ_i توسط گره‌های مه، وظیفه باید به گره مه مجاور ارسال گردد. مطابق رابطه (۲۶) مجموع حداقل پهنای باند مورد نیاز وظایفی که از طریق گره مه f_i^{near} برون‌سپاری می شوند نباید از بیشینه پهنای باند در دسترس آن بیشتر شود. نامساوی (۲۷) نشان می‌دهد که پهنای باند مورد نیاز برای برون‌سپاری وظایف به گره مه پردازنده (گره مه مجاور یا یکی از همسایگانش) نباید از بیشینه پهنای باند آن گره بیشتر شود. نماد $B_{f_i^{near}}^{max}$ نشان‌دهنده بیشینه پهنای باند در دسترس گره مه مجاور شئ i و $B_{f_j}^{max}$ بیشینه پهنای باند در دسترس گره مه پردازنده است. روابط (۲۸) و (۲۹) نیز محدودیت فضای قواعد جریان گره‌های مه برای برون‌سپاری وظایف را نشان می‌دهد. نماد R^i تعداد قواعد مورد نیاز در گره‌های مه برای برون‌سپاری وظیفه θ_i و $R_{f_j}^{max}$ و $R_{f_i^{near}}^{max}$ به ترتیب حداکثر فضای قواعد جریان در دسترس گره مه مجاور و گره مه پردازنده است. محدودیت (۳۰) نیز نشان می‌دهد که متغیرهای مسئله به‌صورت دودویی هستند. مدل ریاضی ارائه شده یک مدل برنامه‌ریزی خطی عدد صحیح مختلط است. تابع هدف و کلیه محدودیت‌ها به‌صورت خطی است و متغیرهای مسئله از نوع عدد صحیح و دودویی می‌باشد [۳۲]. نشان داده می‌شود که مدل برنامه‌ریزی خطی عدد صحیح مختلط برای برون‌سپاری وظایف به سیستم‌های ناهمگون و با در نظر گرفتن محدودیت منابع یک مسئله ان‌پی-سخت است [۳۳، ۳۴].

۴- طرح برون‌سپاری مبتنی بر الگوریتم بهینه‌سازی کلونی مورچه

مدل ریاضی ارائه شده با تعداد محدودی از متغیرهای دودویی توسط پکیج‌های نرم‌افزاری مانند Cplex قابل‌حل است. با توجه به پیچیدگی محاسباتی نمایی در یافتن پاسخ بهینه مدل ریاضی، یک الگوریتم مبتنی بر بهینه‌سازی کلونی مورچه (ACO) ارائه شده است.

پیچیدگی محاسباتی و زمانی روش‌های بهینه‌سازی، منابع محدود گره‌های مه و محدودیت زمانی کاربردهای اینترنت اشیاء از چالش‌های مهم اجرای روش‌های بهینه‌سازی در پردازش مه می‌باشند. اجرای روش‌های بهینه‌سازی با پیچیدگی محاسباتی بالای روی منابع محدود گره‌های مه به‌دلیل زمان اجرای بالا و اشغال بخشی از منابع محدود ناکارآمد است. در روش پیشنهادی، کنترلر شبکه نرم‌افزار-محور مسئول اجرای روش بهینه‌سازی پیشنهادی است. الگوریتم بهینه‌سازی کلونی مورچه یک الگوریتم با پیچیدگی محاسباتی کم می‌باشد که دارای قابلیت اجرای موازی در راستای کاهش زمان اجرا، یافتن سریع پاسخ بهینه و تطبیق‌پذیری با شرایط پیچیده است [۲]. بدین ترتیب زمان اجرای الگوریتم برای وظایف حساس به تأخیر برون‌سپاری شده مناسب است و الگوریتم دارای قابلیت یافتن پاسخ‌های نزدیک بهینه مناسبی با توجه به پیچیدگی ناشی از محدودیت منابع و جابه‌جایی اشیاء در این مسئله است.

مورچه‌ها در مسیر حرکتشان برای هر شئ یکی از گره‌های مه امکان‌پذیر یا پردازش محلی را انتخاب می‌کنند. گره‌های مه امکان‌پذیر برای هر شئ گره مه مجاور آن و همسایگانش است که سرویس کاربردی مورد نیاز شئ را ارائه می‌دهد. مجموعه $O_i = \{l\} \cup \{f_j | f_j \in N_{f_i^{near}} \cup f_i^{near}, a_i \in U_{f_j}\}, \forall i \in I$ گزینه‌های قابل انتخاب برای پردازش وظیفه θ_i را نشان می‌دهد. نماد l نشان‌دهنده پردازش محلی است. رابطه (۳۲) احتمال انتخاب هر یک از گزینه‌های محتمل برای پردازش وظیفه θ_i را نشان می‌دهد. نماد $\tau_{i,c}$ میزان فرومون انتخاب c برای شئ i و $\eta_{i,c}$ یک تابع ابتکاری است که کیفیت این انتخاب را نشان می‌دهد. α وزن تأثیر فرومون و β وزن تأثیر تابع ابتکاری است [۳۵]. تابع $\eta_{i,c}$ به‌صورت تساوی (۳۳) تعریف شده است که پارامتر Q در آن یک مقدار ثابت است. تابع $Cost(i,c)$ مطابق رابطه (۳۴) تعریف می‌شود و هزینه پردازش وظیفه θ_i در انتخاب c را نشان می‌دهد. تأخیر پردازش محلی به‌عنوان هزینه پردازش محلی و مجموع تأخیر و هزینه مهاجرت نتایج در انتخاب c برای شئ i به عنوان هزینه پردازش مه در نظر گرفته شده است.

$$P_{i,c}^k = \frac{(\tau_{i,c})^\alpha \times (\eta_{i,c})^\beta}{\sum_{o\in O_i}((\tau_{i,o})^\alpha \times (\eta_{i,o})^\beta)}, \forall i \in I, \forall c \in O_i \quad (۳۲)$$

$$\eta_{i,c} = \frac{Q}{Cost(i,c)}, \forall i \in I, \forall c \in O_i \quad (۳۳)$$

$$Cost(i,c) = \begin{cases} D_i^{local} & if\ c = l \\ D_{i,f_j}^{fog} + MobilityCost_{i,f_j} & if\ c = f_j \in N_{f_i^{near}} \cup f_i^{near} \end{cases}, \quad \forall i \in I, \forall c \in O_i, \forall f_i^{near} \in F \quad (۳۴)$$

پس از تعیین احتمالات از روش چرخ رولت جهت انتخاب گزینه‌ها و ایجاد مسیر استفاده می‌شود. مقدار برازندگی برای مسیر به دست آمده از مورچه k متناسب با معکوس هزینه کل مسیر آن است و به‌صورت رابطه (۳۵) محاسبه می‌شود. المان i از ماتریس $selected^k$ انتخاب مورچه kام را برای شئ i نشان می‌دهد. هزینه کل مسیر مطابق معادله (۳۶) برابر مجموع هزینه هر یک از انتخاب‌های مورچه k است.

$$Fitness^k = \frac{1}{TotalCost^k} \quad (۳۵)$$

$$TotalCost^k = \sum_{i \in S_I} Cost(i,c)\,, c = selected^k[i] \quad (۳۶)$$

با توجه به محدودیت‌های بیشینه تأخیر ، انرژی باقی‌مانده شئ، فضای قواعد جریان و پهنای باند گره‌های مه صرفاً برازندگی تعریف شده معیار مناسبی برای ارزیابی مسیر نیست. تابع v_{Total}^k در معادله (۳۷) تخطی کل برای مورچه kام را نشان می‌دهد. $v_{i,c}^{B^k}$ تخطی پهنای باند، $v_{i,c}^{R^k}$ تخطی فضای قواعد جریان گره‌های مه، $v_{i,c}^{E^k}$ تخطی انرژی و $v_{i,c}^{D^k}$ تخطی تأخیر انتخاب مورچه kام برای شئ i است.

$$v_{Total}^k = \sum_{i \in S_I} v_{i,c}^{B^k} + v_{i,c}^{R^k} + v_{i,c}^{E^k} + v_{i,c}^{D^k}\,, c = selected^k[i] \quad (۳۷)$$

تابع تخطی پهنای باند به‌صورت تساوی (۳۸) است. نماد $[x]^+$ معادل بیشینه ۰ و x است. اگر پردازش به‌صورت محلی انجام شود، به دلیل عدم مصرف پهنای باند تخطی صفر است.

$$v_{i,c}^{B^k} = \begin{cases} 0 & if\ c = l \\ \left[\frac{b_i^{min}}{B_{f_i^{near}}^{max}} - 1\right]^+ & if\ c = f_i^{near} \\ \left[\frac{b_i^{min}}{B_{f_i^{near}}^{max}} - 1\right]^+ + \left[\frac{b_i^{min}}{B_{f_j}^{max}} - 1\right]^+ & if\ c = f_j \in N_{f_i^{near}} \end{cases}, \quad (۳۸)$$
$$\forall i \in I, \forall f_i^{near} \in F, c = selected^k[i]$$

تابع تخطی تأخیر مطابق معادله (۳۹) و تابع تخطی انرژی مطابق معادله (۴۰) است.

$$v_{i,c}^{D^k} = \begin{cases} \left[\frac{D_i^{local}}{d_i^{max}} - 1\right] & if\ c = l \\ \left[\frac{D_{i,f_j}^{fog}}{d_i^{max}} - 1\right]^+ & otherwise \end{cases}, \forall i \in I, \forall f_i^{near} \in F, \quad (۳۹)$$
$$c = selected^k[i]$$

$$v_{i,c}^{E^k} = \begin{cases} \left[\frac{E_i^{local}}{e_i^{min}} - 1\right] & if\ c = l \\ \left[\frac{E_{i,f_i^{near}}^{fog}}{e_i^{min}} - 1\right]^+ & otherwise \end{cases}, \forall i \in I, \forall f_i^{near} \in F, \quad (۴۰)$$
$$c = selected^k[i]$$

تابع تخطی فضای قواعد جریان به‌صورت تساوی (۴۱) است. اگر پردازش به‌صورت محلی انجام شود، تخطی صفر است.

$$v_{i,c}^{R^k} = \begin{cases} 0 & if\ c = l \\ \left[\frac{R_i}{R_{f_i^{near}}^{max}} - 1\right]^+ & if\ c = f_i^{near} \\ \left[\frac{R_i}{R_{f_i^{near}}^{max}} - 1\right]^+ + \left[\frac{R_i}{R_{f_j}^{max}} - 1\right]^+ & if\ c = f_j \in N_{f_i^{near}} \end{cases}, \quad (۴۱)$$
$$\forall i \in I, \forall f_i^{near} \in F, c = selected^k[i]$$

در هر تکرار از الگوریتم فرومون مربوط به هر انتخاب مطابق رابطه (۴۲) به روزرسانی می‌شود. نماد ρ نرخ تبخیر فرومون و $\Delta\tau^k$ مقدار فرومون آزاد شده توسط مورچه k است [۳۶]. مقدار فرومون آزاد شده توسط مورچه k به‌صورت تساوی (۴۳) تعریف شده است. جهت هدایت پاسخ‌های آتی به سمت محدوده پاسخ‌های امکان‌پذیر درصورتی‌که تخطی کل پاسخ یک مورچه مخالف صفر باشد، مقدار فرومون ترشح‌شده مربوط به آن مورچه صفر خواهد بود. Q برابر یک مقدار ثابت است.

$$\tau_{i,c}^k = (1-\rho) \times \tau_{i,c}^k + \Delta\tau^k\,, \forall i \in I, c = selected^k[i] \quad (۴۲)$$

$$\Delta\tau^k = \begin{cases} 0 & if\ v_{Total}^k \neq 0 \\ \frac{Q}{TotalCost^K} & otherwise \end{cases} \quad (۴۳)$$

در پایان هر تکرار از الگوریتم، بهترین پاسخ به دست آمده از بهترین مورچه به‌عنوان بهینه عمومی در نظر گرفته می‌شود و فرومون بهترین مورچه عمومی مطابق رابطه (۴۴) مجدداً به مسیر انتخابی آن تزریق می‌شود. فرومون تزریق شده متناسب با اختلاف هزینه حاصل از مورچه بهینه عمومی تکرار قبلی و هزینه حاصل از مورچه بهینه عمومی تکرار جاری است. بدین ترتیب هرچه بهبود به نسبت بهینه قبلی بیشتر باشد، فرومون مسیر بیشتر خواهد بود و از دستیابی به پاسخ‌های بهینه محلی ناشی از تجمع فرومون در بهینه‌های محلی جلوگیری می‌شود. نماد $TotalCost^{GBAnt}$ هزینه کل مسیر طی شده توسط مورچه بهینه عمومی تکرار جاری و $TotalCost^{LGBAnt}$ هزینه کل مسیر طی شده توسط مورچه بهینه عمومی تکرار قبلی است که مطابق معادله (۳۶) محاسبه می‌شود. ρ_G نرخ تبخیر در به‌روزرسانی فرومون بهینه عمومی است.

$$\tau_{i,c}{}^{GBAnt} = (1-\rho_G)\tau_{i,c}{}^{GBAnt} + \frac{TotalCost^{LGBAnt} - TotalCost^{GBAnt}}{TotalCost^{GBAnt}} \quad (۴۴)$$

الگوریتم ۱ شبه کد الگوریتم فرا-ابتکاری مبتنی بر بهینه‌سازی کلونی مورچه است.

Algorithm 1 The proposed ACO task offloading algorithm

Input: α, β, $RandFactor$, $RandFactor_{min}$, N_{Ants}, ρ, ρ_G, Q, N_{it}
Output: $BestSelections$

```
Initialize the pheromone matrix
while θ ≤ N_it do
  for k = 1 to N_Ants do
    for each i ∈ I do
      if a random number between 0 and 1> RandFactor then
        selected^k[i]= a random selection from O_i
      else
        selected^k[i]=choose a member of O_i by P^k_{i,c} using the
        roulette wheel method
      end if
    end for
  end for
  shuffle things ID in I
  for each i ∈ I do
    if v^{B^k}_{i,c} > 0 or v^{R^k}_{i,c} > 0 then
      if D^local_i < d^max_i and E^local_i < e^min_i then
        selected^k[i] = l
        v^{B^k}_{i,c}, v^{R^k}_{i,c} =0
      end if
    end if
    Update bandwidth and flow rule capactity
  end for
  if fitness of ant k > BestFit then
    BestFit= Fitness^k
    BestSelections= selected^k
  end if
  Update pheromones using Eq(42)
  if BestFit > GBest and v^{BestAnt}_{Total} = 0 then
    GBest = BestFit
    Update pheromones using Eq(44)
  end if
  θ = θ+1
  if RandFactor > RandFactor_min then
    Reduce the RandFactor by multiplying by 0/99
  end if
end while
```

جهت تقویت احتمال جست‌وجو در حالت‌هایی که قبلاً انتخاب نشده است، انتخاب‌هایی به‌صورت تصادفی متناسب با پارامتر $RandFactor$ در نظر گرفته شده است. جهت کنترل سرعت همگرایی و تعادل بین اکتشاف و بهره‌برداری با الهام از الگوریتم شبیه‌سازی تبرید در ابتدا پارامتر $RandFactor$ زیاد و در هر مرحله تا رسیدن به حد آستانه مقدار آن با ضرب شدن در مقدار ۰/۹۹ کم می‌شود. مشابه مقاله [۳۷] در صورت بروز تخطی سعی در رفع آن و انتقال انتخاب به فضای امکان‌پذیر مسئله شده است. درصورتی‌که تخطی مربوط به پهنای باند و فضای قواعد جریان باشد، یک انتخاب قطعاً امکان‌پذیر پردازش محلی وظیفه می‌باشد. بنابراین اگر در پردازش محلی وظیفه شئ تخطی تأخیر و انرژی ایجاد نشود، پردازش محلی برای شئ انتخاب می‌شود. هدایت مسئله به فضای امکان‌پذیر مانع از همگرایی پیش از یافتن پاسخ امکان‌پذیر برای مسئله می‌شود. پیچیدگی محاسباتی الگوریتم فرا-ابتکاری ارائه شده در بدترین حالت به‌صورت $O(N_{iteration} \times |N_{Ants}| \times |I| \times |F|)$ است.

۵- ارزیابی کارایی روش پیشنهادی

الگوریتم فرا-ابتکاری پیشنهادی با زبان برنامه‌نویسی جاوا پیاده‌سازی و مدل ریاضی خطی عدد صحیح مختلط با استفاده از پکیج نرم‌افزاری Cplex حل شده است. ارزیابی روش پیشنهادی با استفاده از یک لپ‌تاپ با پردازنده اینتل پنج هسته‌ای با فرکانس ۱/۸ گیگاهرتز و با حافظه رم به فضای ۶ گیگابایت انجام شده است.

همبندی شبکه به‌صورت مش جزئی و یک گراف بدون جهت همبند است. توزیع گره‌های مه و همسایگان آن‌ها و اشیاء به‌صورت تصادفی است. ۲۰۰ گره مه در فضای مسئله با حداکثر ۲۰ همسایگی در نظر گرفته شده است. فرکانس پردازنده گره‌های مه بین (۲/۹-۴/۲) گیگاهرتز و شعاع هر گره مه ۱۰۰ متر است. جهت سادگی مانند روش ارائه شده در مقاله [۲۶] به ازای هر وظیفه صرفاً یک قاعده در فضای قواعد جریان ثبت می‌شود. ظرفیت باقی‌مانده فضای قواعد جریان گره‌های مه در بازه (۵۰۰-۱۰۰) و پهنای باند در دسترس آن‌ها ۲۰ مگاهرتز است. قدرت نویز ۱۰۰- دسی‌بل-میلی‌وات، قدرت انتقال کانال بین گره های مه در بازه (۲۰-۸) وات و قدرت انتقال کانال گره مه و شئ در بازه (۰۸/۱-۰/۰) وات است. زمان اقامت اشیاء، یک مقدار تصادفی با توزیع گوسی و با میانگین ۳۰ ثانیه و انحراف معیار۱۰ ثانیه در نظر گرفته شده است [۱۶]. فرکانس پردازنده اشیاء در بازه (۳۰-۱۰) مگاهرتز، بیشینه تأخیر قابل تحمل آن‌ها ۵۰ ثانیه و حداقل پهنای باند مورد نیازشان بین (۱-۰/۵) مگاهرتز است. انرژی باقی‌مانده باتری اشیاء ۲۰۰ ژول و مصرف انرژی برای هر سیکل پردازنده آن‌ها 5×10^{-4} ژول است. مجموعه برنامه‌های کاربردی دارای ۱۰ عضو است که هر گره مه به‌صورت تصادفی بین ۸ تا ۱۰ عدد از این سرویس‌های کاربردی را ارائه می‌دهد. حجم وظایف بین ۱۰ تا ۱۵ مگابایت و تعداد سیکل مورد نیاز آن‌ها جهت پردازش در بازه (۳۰۰-۲۰۰) مگاسیکل بر مگابایت است. هزینه مهاجرت به ازای هر گام ۲ و تأخیر انتشار و مسیریابی در هر گام ۲ ثانیه در نظر گرفته شده است.

جهت ارزیابی روش پیشنهادی پارامتر N_{it}برابر ۴۰۰، α و β به ترتیب برابر ۱ و ۵، $RandFactor$ و $RandFactor_{min}$به ترتیب ۰/۵ و ۰/۰۵ و N_{Ants} برابر ۲۰ تنظیم شده است. پارامترهای ρ_G و ρ به ترتیب برابر ۰/۱ و ۰/۷ قرار داده شده است.

روش پیشنهادی با مقدار بهینه حاصل از حل مدل ریاضی، روش تخصیص وظایف به‌صورت تصادفی و روش سافت-ون (soft-VAN) ارائه شده در مقاله [۲۱] مقایسه شده است. در روش تصادفی با احتمال مساوی پردازش محلی یا مه انتخاب می‌شود. سپس در صورت انتخاب پردازش مه، یک گره مه به‌صورت تصادفی با توزیع یکنواخت برای پردازش وظیفه تعیین می‌گردد. در صورت نقض محدودیت‌ها، یک انتخاب تصادفی دیگر صورت می‌گیرد. در روش سافت-ون هر خودرو (شئ) تحت پوشش یک واحد کنار جاده‌ای[۶] است. تمامی واحدهای کنار جاده‌ای قابلیت پردازش مه را ندارند و وظایف باید به‌صورت چندگامه به گره مه مناسب ارسال شوند. معیار انتخاب گره مه در این روش کاهش تأخیر پردازش و صف است. روش سافت-ون تأثیر جابه‌جایی را در انتخاب گره‌های مه در نظر نمی‌گیرد و صرفاً مسیر مناسب برای ارسال نتایج به شئ پس از جابه‌جایی را تعیین می‌کند. این درحالی است که در بسیاری از موارد با توجه به رفتار اشیاء در جابه‌جایی می‌توان وظایف را به گره مه‌ای تخصیص داد که هزینه مهاجرت نتایج و تأخیر دریافت نتایج کاهش یابد. در روش پیشنهادی کلیه ایستگاه‌های پایه دارای قابلیت پردازش مه هستند و هر وظیفه توسط گره مه مجاور یا همسایگانش پردازش می‌شود. معیار انتخاب گره مه در روش پیشنهادی کاهش تأخیر کل و هزینه مهاجرت نتایج ناشی از جابه‌جایی اشیاء است. با توجه به هدف پردازش مه مبنی بر نزدیک نمودن محل پردازش به اشیاء، جهت کاهش تأخیر و جلوگیری از افزایش فاصله محل پردازش از اشیاء محدودیت پردازش صرفاً توسط گره مه مجاور و همسایگانش در نظر گرفته شده است. روش پیشنهادی برخلاف سافت-ون توزیع برنامه‌های کاربردی در گره‌های مه را در نظر می‌گیرد. روش ابتکاری پیشنهاد شده در سافت-ون محدودیت منابع را در نظر نمی‌گیرد. اما روش فرا-ابتکاری پیشنهادی به مدیریت محدودیت منابع در انتخاب گره‌های مه می‌پردازد. بدین ترتیب جهت مقایسه روش پیشنهادی با روش سافت-ون علاوه بر سناریوی ارزیابی روش پیشنهادی با در نظر گرفتن محدودیت منابع، سناریوی عدم وجود محدودیت منابع نیز در نظر گرفته شده است.

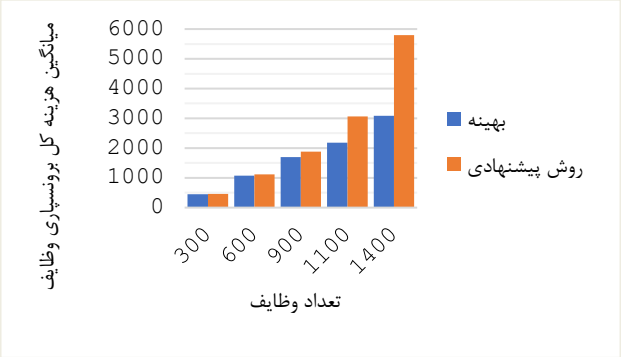
۵-۱- ارزیابی روش پیشنهادی با در نظر گرفتن محدودیت منابع

مجموع هزینه برون‌سپاری وظایف اشیاء به‌عنوان هزینه برون‌سپاری کل در نظر گرفته شده است. میانگین هزینه کل برون‌سپاری وظایف از ۳۰۰ وظیفه تا ۱۴۰۰ وظیفه در شکل ۲ نشان داده شده است.هزینه برون‌سپاری وظایف در روش تصادفی از ۳۰۰ تا ۱۴۰۰ وظیفه به ترتیب برابر ۲۷۸۹,۳۴، ۵۵۰۸,۲۱، ۸۲۵۸,۰۴، ۱۰۰۳۸,۶۴ و ۱۲۷۳۸,۶۴ است. الگوریتم پیشنهادی به نسبت الگوریتم تصادفی نتایج بهتری را ارائه نموده است. نمودار ارائه شده نشان می‌دهد، اختلاف روش فرا-ابتکاری با مقدار بهینه با افزایش تعداد وظایف بیشتر می‌شود. این موضوع ناشی از رشد حالت‌های احتمالی فضای مسئله و محدود شدن حالت‌های امکان‌پذیر به علت محدودیت‌های مسئله است. نتایج روش پیشنهادی در حالت ۳۰۰ و ۶۰۰ به مقدار بهینه بسیار نزدیک است. زیرا در برون‌سپاری این تعداد وظیفه محدودیت منابع تأثیر کمی دارد. با افزایش تعداد وظایف از ۹۰۰ تا ۱۴۰۰ به علت افزایش تقاضا برای مصرف منابع و تشدید محدودیت منابع، اختلاف روش پیشنهادی و روش بهینه افزایش می‌یابد. میانگین هزینه برون‌سپاری وظایف در روش پیشنهادی از روش بهینه ۲۸,۷۵ درصد بیشتر است.

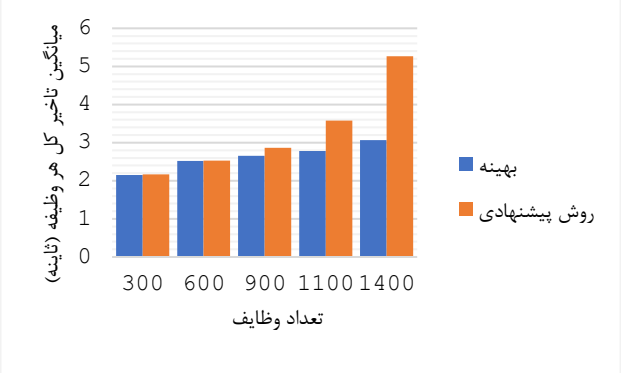
شکل ۳ نشان‌دهنده تأثیر افزایش تعداد وظایف اشیاء بر میانگین تأخیر کل هر وظیفه است. میانگین تأخیر کل هر وظیفه در روش تصادفی از ۳۰۰ تا ۱۴۰۰ وظیفه به ترتیب برابر ۹,۳۴، ۹,۲۳، ۹,۲۳، ۹,۱۹۴ و ۹,۱۸۲ است. با توجه به اینکه احتمال انتخاب پردازش محلی و پردازش مه در روش تصادفی برابر ۰,۵ است، افزایش تعداد وظایف تأثیر به سزایی در میانگین تأخیر وظایف ندارد. در صورت انتخاب پردازش مه، احتمال انتخاب تمامی گره‌های مه برای هر وظیفه در صورت وجود منابع کافی برابر است. روال نزولی جزئی روش تصادفی ناشی

[۶] Road Side Unit

از این است که با افزایش تعداد و تنوع وظایف احتمال تخصیص وظایف با حجم و پیچیدگی محاسباتی بیشتر به گره‌های مه با منابع بیشتر افزایش می‌یابد. نتایج نشان‌دهنده کارایی بهتر روش پیشنهادی به نسبت روش تخصیص تصادفی و روند صعودی تأخیر در روش بهینه و روش پیشنهادی با افزایش تعداد وظایف است. محدودیت پهنای باند گره‌های مه و محدودیت فضای قواعد آن‌ها مانع از برون‌سپاری تمامی وظایف با افزایش تعداد وظایف می‌شود. وظایفی که امکان برون‌سپاری آن‌ها وجود ندارد، در صورت عدم نقض محدودیت تأخیر و انرژی باقی‌مانده شئ، به‌صورت محلی پردازش می‌شوند. محدودیت منابع پردازشی اشیاء باعث می‌شود که زمان اجرای وظایف در حالت پردازش محلی به نسبت پردازش مه بیشتر باشد. افزایش تأخیر صف در گره‌های مه با افزایش تعداد وظایف متقاضی دریافت خدمات از دیگر دلایل رشد صعودی تأخیر با افزایش تعداد وظایف است. روش پیشنهادی در مقایسه با مقدار بهینه ۲۲ درصد میانگین تأخیر کل بیشتری ایجاد می‌کند.



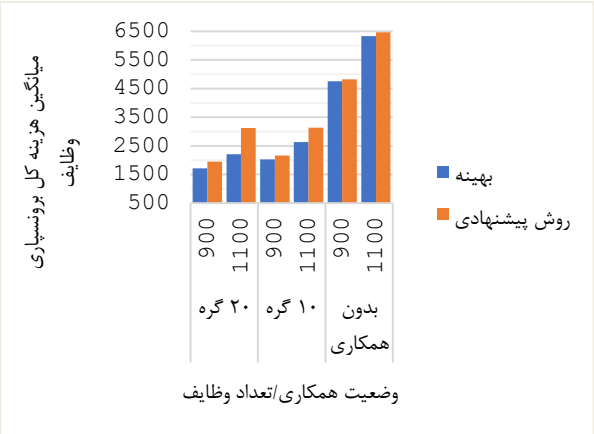
شکل۲- تأثیر افزایش تعداد وظایف بر متوسط هزینه کل برون‌سپاری وظایف



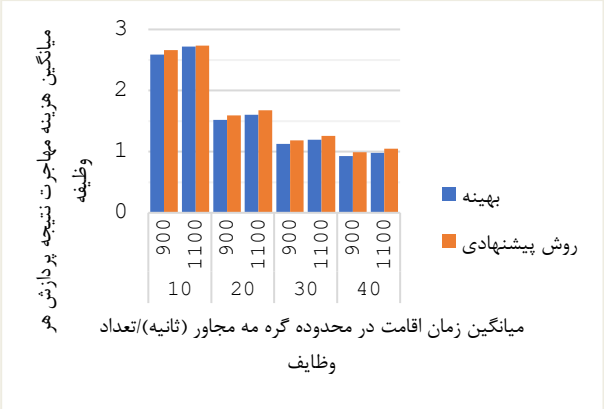
شکل۳- تأثیر افزایش تعداد وظایف بر متوسط تأخیر کل هر وظیفه

جهت بررسی تأثیر همکاری گره‌های مه بر کارایی برون‌سپاری وظایف سه حالت عدم همکاری (برون‌سپاری وظیفه فقط به گره مه مجاور)، همکاری ۱۰ گره مه و همکاری ۲۰ گره مه در نظر گرفته شده است. در هر سه حالت دو وضعیت برون‌سپاری ۹۰۰ و ۱۱۰۰وظیفه به ۲۰۰ گره مه در نظر گرفته شده است. شکل ۴ نتایج بررسی وضعیت همکاری گره‌های مه را نشان می‌دهد. نتایج حاصل شده حاکی از آن است که با افزایش تعداد گره‌های مه همکار هزینه کل برون‌سپاری کاهش می‌یابد. درصورتی‌که گره مه مجاور منابع پردازشی قوی‌ای نداشته باشد یا بار پردازشی زیادی در آن وجود داشته باشد، زمان دریافت نتایج افزایش می‌یابد. این موضوع باعث افزایش تأخیر و احتمال مهاجرت نتایج و به‌تبع آن افزایش هزینه برون‌سپاری وظایف می‌شود. اختلاف روش پیشنهادی و روش بهینه با کاهش تعداد گره‌های مه همکار کاهش می‌یابد، این موضوع ناشی از کاهش حالت‌های امکان‌پذیر مسئله و کاهش گزینه‌های قابل انتخاب برای وظایف می‌باشد. متوسط هزینه برون‌سپاری وظایف در روش پیشنهادی از مقدار بهینه در شرایط عدم همکاری، ۱۰ گره مه همکار و ۲۰ گره مه همکار به ترتیب ۱٫۶۷ درصد، ۱۲٫۸۲ درصد و ۲۱ درصد بیشتر است.

میانگین زمان اقامت اشیاء در محدوده گره مه مجاور، به‌عنوان پارامتر ارزیابی هزینه مهاجرت نتایج در نظر گرفته‌شده است. شکل ۵ نمودار ارتباط هزینه مهاجرت با میانگین زمان اقامت اشیاء را نشان می‌دهد. در این ارزیابی میانگین هزینه مهاجرت نتیجه پردازش وظایف برای برون‌سپاری ۹۰۰ و ۱۱۰۰ وظیفه به ۲۰۰ گره مه در میانگین زمان‌های اقامت ۱۰،۲۰،۳۰ و ۴۰ ثانیه سنجیده شده است. مطابق این شکل با افزایش میانگین زمان اقامت اشیاء هزینه مهاجرت کاهش می‌یابد. زیرا افزایش میانگین زمان اقامت اشیاء باعث کاهش احتمال مهاجرت نتایج می‌شود. نتایج حاکی از آن است که با افزایش میانگین زمان اقامت اشیاء، اخلاف هزینه مهاجرت نتایج در روش پیشنهادی و مقدار بهینه افزایش می‌یابد. اخلاف هزینه مهاجرت نتایج در روش پیشنهادی و مقدار بهینه با میانگین زمان‌های اقامت ۱۰ثانیه، ۲۰ ثانیه، ۳۰ ثانیه و ۴۰ ثانیه به ترتیب برابر ۱٫۶۵ درصد، ۴٫۶ درصد، ۵٫۲ درصد و ۶٫۹ درصد است. این موضوع ناشی از آن است که با افزایش زمان اقامت شئ تعداد بیشتری از گره‌های مه کاندید برای پردازش وظیفه برون‌سپاری شده هستند. علاوه‌براین با افزایش تعداد وظایف به علت محدودیت منابع و عدم امکان برون‌سپاری به گره‌های مه مناسب از نظر هزینه جابه‌جایی، هزینه مهاجرت نتایج افزایش می‌یابد.

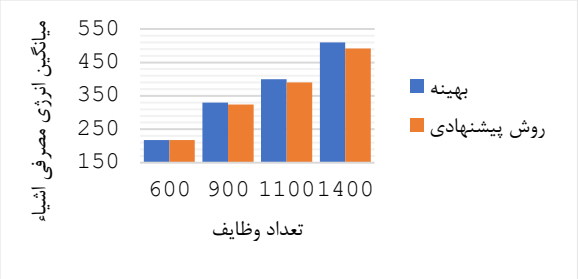


شکل۴- بررسی تأثیر همکاری گره‌های مه بر هزینه کل برون‌سپاری



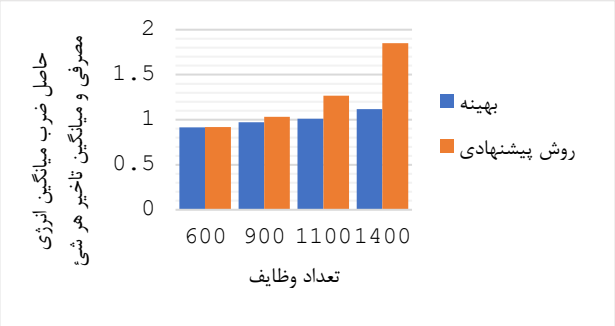
شکل۵- تأثیر افزایش میانگین زمان اقامت اشیاء بر متوسط هزینه مهاجرت نتایج پردازش وظایف

تأثیر افزایش تعداد وظایف بر مصرف انرژی اشیاء و مقایسه انرژی مصرفی اشیاء در روش پیشنهادی و مقدار بهینه در شکل ۶ نشان داده شده است. با افزایش تعداد وظایف از ۶۰۰ تا ۱۴۰۰ وظیفه اختلاف انرژی مصرفی اشیاء در روش پیشنهادی و مقدار بهینه افزایش می‌یابد. نتایج نشان می‌دهد که روش پیشنهادی در مقایسه با مقدار بهینه دارای مصرف انرژی کمتری می‌باشد. انرژی مصرفی شئ در برون‌سپاری وظایف ناشی از ارسال وظیفه به گره مه مجاور و در پردازش محلی ناشی از پردازش وظیفه است. بنابراین انتخاب گره‌های مه متفاوت به‌عنوان گره مه پردازنده در مصرف انرژی شئ تأثیری ندارد. با توجه به استفاده از تصحیح تخطی در روش پیشنهادی با تشدید محدودیت منابع، احتمال پردازش محلی وظایف توسط اشیاء افزایش می‌یابد. مصرف انرژی اشیاء در پردازش محلی نسبت به برون‌سپاری وظایف کمتر در نظر گرفته شده است.



شکل ۶- تأثیر افزایش تعداد وظایف بر میانگین انرژی مصرفی اشیاء

معیار ارزیابی حاصل ضرب تأخیر و انرژی مصرفی اشیاء[۷] (EDP) به‌منظور بررسی کارایی روش پیشنهادی در مقایسه با روش بهینه برای برون‌سپاری ۶۰۰ تا ۱۴۰۰ وظیفه در شکل ۷ ارائه شده است. اختلاف روش پیشنهادی و مقدار بهینه مطابق معیار کارایی EDP معادل ۲۴ درصد است که نشان می‌دهد روش پیشنهادی از منظر تأخیر و انرژی مصرفی کارایی مناسبی در مقایسه با روش بهینه دارد.



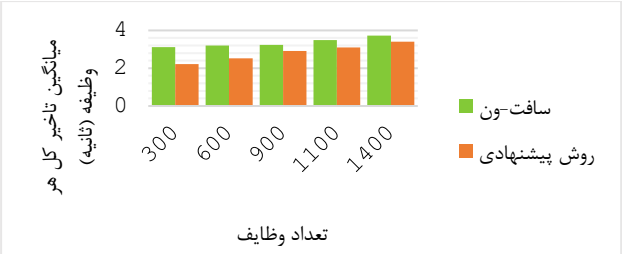
شکل ۷- تأثیر افزایش تعداد وظایف بر میانگین انرژی مصرفی اشیاء

۵-۲- ارزیابی روش پیشنهادی بدون نظر گرفتن محدودیت منابع

جهت ارزیابی کارایی روش‌های پیشنهادی در مقایسه با روش سافت-ون شرایط مسئله را کاملاً بدون محدودیت در نظر می‌گیریم. محدودیت‌های فضای مسئله مانع از انتخاب کم‌هزینه‌ترین پردازش برای وظایف می‌شوند. بنابراین در شرایطی که محدودیت وجود دارد، ممکن است سافت-ون کم‌هزینه‌ترین گزینه را بدون در نظر گرفتن محدودیت‌ها انتخاب کند و حتی از حالت بهینه در مسئله نیز هزینه کمتری را فراهم کند. به‌منظور سازگاری بیشتر سافت-ون و روش پیشنهادی، برای این روش نیز همکاری بین گره‌های مه و عدم امکان برون‌سپاری وظایف به گره‌های مه‌ای که سرویس کاربردی مورد نیاز را ندارند، در نظر گرفته شده است.

شکل ۸ نشان‌دهنده میانگین تأخیر کل هر وظیفه، متناسب با افزایش تعداد اشیاء در شرایط بدون محدودیت است. میانگین تأخیر وظایف در روش سافت-ون از روش پیشنهادی بیشتر است. یک علت این موضوع این است که سافت-ون برای انتخاب گره مه تأخیر انتشار و تأخیر انتقال مربوط به برون‌سپاری وظایف و تعداد گام‌های مهاجرت نتایج و احتمال آن را در نظر نمی‌گیرد. بنابراین تأخیر انتشار و انتقال نامناسب در برون‌سپاری وظایف و تأخیر انتشار و مسیریابی زیاد برای نتایج پردازش باعث بیشتر بودن تأخیر کل سافت-ون می‌شود. یکی از دلایل دیگر ماهیت حریصانه سافت-ون و در نظر نگرفتن نیازمندی سایر وظایف در هنگام تعیین گره مه برای آن‌ها است. روش پیشنهادی جهت تعیین نحوه پردازش و محل برون‌سپاری، هزینه برون‌سپاری تمامی وظایف را باهم در نظر می‌گیرد. علاوه بر این ماهیت تصادفی روش پیشنهادی به یافتن حالت‌هایی با تأخیر کل کمتر کمک می‌کند. روش پیشنهادی به نسبت روش سافت-ون ۲۰ درصد میانگین تأخیر وظایف را کاهش می‌دهد. روند صعودی تأخیر در هر دو روش ناشی از افزایش تأخیر صف گره‌های مه است. با توجه به اینکه با افزایش تعداد وظایف تأخیر صف بر سایر تأخیرها غلبه می‌کند و روش سافت-ون گره‌های مه را بر اساس تأخیر صف و پردازش تخصیص می دهد، اختلاف روش پیشنهادی و روش سافت-ون با افزایش تعداد وظایف کاهش می‌یابد. با مقایسه شکل ۳ و ۶ می‌توان استنتاج نمود که تأخیر کل در روش پیشنهادی برای برون‌سپاری ۳۰۰ و ۶۰۰ وظیفه که محدودیت منابع تأثیر ناچیزی دارد، تقریباً برابر با شرایط بدون محدودیت است. اختلاف تأخیر در شرایط وجود محدودیت منابع و عدم محدودیت جهت برون‌سپاری ۹۰۰ تا ۱۴۰۰ وظیفه به علت تشدید محدودیت‌های منابع بیشتر است. نتایج نشان می‌دهد تأخیر کل حاصل از روش پیشنهادی در صورت وجود محدودیت منابع از شرایط عدم وجود محدودیت منابع بیشتر است.

روش سافت-ون در تخصیص وظایف به گره‌های مه هزینه مهاجرت نتایج پردازش را در نظر نمی‌گیرد. شکل ۹ جهت مقایسه هزینه کل مهاجرت نتایج در روش پیشنهادی و روش سافت-ون ارائه شده است. روش سافت-ون جابه‌جایی را صرفاً در انتخاب مسیر کم هزینه‌تر برای بارگیری نتیجه در نظر می‌گیرد. در روش پیشنهادی در بسیاری از موارد پیش از جابه‌جایی شئ نتیجه به آن تحویل داده می‌شود. از طرف دیگر انتخاب گره مه‌ای که با مکان بعدی شئ فاصله کمتری داشته باشد، باعث کاهش هزینه‌های ناشی از جابه‌جایی شئ می‌شود. روش پیشنهادی به میزان ۴۰ درصد هزینه مهاجرت نتایج را به نسبت روش سافت-ون کاهش می‌دهد.



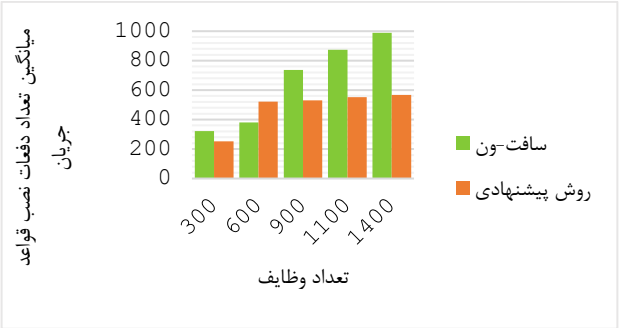
شکل۸-نمودار میانگین تأخیر هر وظیفه جهت مقایسه روش پیشنهادی و روش سافت-ون

شکل ۱۰ نشان‌دهنده نحوه تغییر تعداد متوسط نصب قواعد جریان متناسب با افزایش تعداد اشیاء است. سوییچ‌های اپن‌فلو با دریافت یک بسته از یک جریان

[۷] Energy-Delay-Product

داده، جدول جریان خود را برای یافتن قاعده مرتبط با آن جریان جست‌وجو می‌کنند. درصورتی‌که قاعده مربوط به جریان مرتبط با بسته در سوییچ وجود نداشته باشد، درخواستی جهت نصب قاعده مرتبط به کنترلر ارسال می‌شود. بنابراین برای مسیریابی یک جریان داده به ازای هر گام باید یک بار نصب قواعد جریان صورت پذیرد. ارسال پیام‌های مربوط به نصب قواعد جریان باعث ایجاد تأخیر در مسیریابی و بار اضافی برای شبکه می‌شود. نتایج نشان می‌دهد روش سافت-ون به علت در نظر نگرفتن تعداد گام‌های مورد نیاز جهت ارسال نتیجه پردازش وظایف، میزان نصب قواعد جریان بیشتری نسبت به روش پیشنهادی ارائه می‌دهد.

شکل ۹- نمودار میانگین هزینه مهاجرت نتایج در روش پیشنهادی و روش سافت-ون



شکل ۱۰- نمودار میانگین نصب قواعد جریان ناشی از ارسال نتیجه پردازش وظایف

۶- نتیجه گیری

در این پژوهش یک طرح برون‌سپاری وظایف با در نظر گرفتن جابه‌جایی اشیاء، نیازمندی‌های ترافیک اینترنت اشیاء و مزایای شبکه‌های نرم‌افزار-محور ارائه شده است. همچنین با هدف استفاده حداکثری از فضای مه، همکاری گره‌های مه در پردازش وظایف اشیاء در نظر گرفته شده است. یک مدل ریاضی بهینه‌سازی برنامه‌ریزی خطی عدد صحیح مختلط با هدف کاهش تأخیر و هزینه مهاجرت نتایج پردازش وظایف ارائه گردیده است و جهت تقریب مدل ریاضی یک الگوریتم فرا-ابتکاری مبتنی بر بهینه‌سازی کلونی مورچه با در نظر گرفتن محدودیت منابع ارائه شده است. نتایج حاصل روشنگر این موضوع می‌باشند که الگوریتم پیشنهادی در کاهش هزینه مهاجرت نتایج پردازش و تأخیر نتایج قابل مقایسه‌ای با حالت بهینه دارد و در مقایسه با روش سافت-ون و روش تصادفی تأخیر و هزینه جابه‌جایی کمتری فراهم می‌کند. طرح جاری صرفاً برون‌سپاری وظایف در پردازش مه را در نظر گرفته است. در ادامه این کار پژوهشی می‌توان مباحث زمان‌بندی پردازش وظایف و نحوه قرارگیری سرویس‌های کاربردی را در کنار برون‌سپاری وظایف در گره‌های مه در نظر گرفت.

مراجع

[۱] بهشید شایسته، وصال حکمی، سید اکبر مصطفوی، احمد اکبری، «ارائه روشی نوین برای محاسبه اعتماد در کاربردهای اینترنت اشیاء»، مجله مهندسی برق دانشگاه تبریز، جلد ۵۰، شماره ۲، صفحات ۷۵۵-۷۴۳، ۱۳۹۹

[2] S. O. Ogundoyin and I. A. Kamil, "Optimization techniques and applications in fog computing: An exhaustive survey," Swarm Evol Comput, vol. 66, p. 100937, Oct. 2021, doi: 10.1016/J.SWEVO.2021.100937.

[3] V. K. M. D. O. S. H. Pedram, "Energy and task completion time trade-off for task offloading in fog-enabled IoT networks," Pervasive Mob Comput, vol. 74, 2021.

[۴] شهرام جمالی، سمیرا حورعلی، «موازنه گر نامتمرکز بار در محیط ابر با بهره گیری از سیاست تصمیم گیری چندشاخصه»، مجله مهندسی برق دانشگاه تبریز، جلد ۴۶، شماره ۳، صفحات ۱۰۶-۹۶، ۱۳۹۵.

[5] Y. Chen, F. Zhao, Y. Lu, and X. Chen, "Dynamic Task Offloading for Mobile Edge Computing with Hybrid Energy Supply," Tsinghua Sci Technol, vol. 28, no. 3, pp. 421–432, 2023, doi: 10.26599/TST.2021.9010050.

[6] A. M. Alwakeel, "An Overview of Fog Computing and Edge Computing Security and Privacy Issues," Sensors, vol. 21, no. 24, Dec. 2021.

[7] M. H. A. A. W. A. H. A. a. M. A. B. Alouffi, "A Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies," IEEE Access, pp. 1–1, 2021.

[8] A. Kishor and C. Chakarbarty, "Task Offloading in Fog Computing for Using Smart Ant Colony Optimization," Wirel Pers Commun, 2021, doi: 10.1007/s11277-021-08714-7.

[9] T. Gao, Q. Tang, J. Li, Y. Zhang, Y. Li, and J. Zhang, "A Particle Swarm Optimization With Lévy Flight for Service Caching and Task Offloading in Edge-Cloud Computing," IEEE Access, vol. 10, pp. 76636–76647, 2022, doi: 10.1109/ACCESS.2022.3192846.

[10] M. Keshavarznejad, M. H. Rezvani, and S. Adabi, "Delay-aware optimization of energy consumption for task offloading in fog environments using metaheuristic algorithms," Cluster Comput, vol. 24, no. 3, pp. 1825–1853, 2021, doi: 10.1007/s10586-020-03230-y.

[11] I. Sarkar, M. Adhikari, N. Kumar, and S. Kumar, "Dynamic Task Placement for Deadline-Aware IoT Applications in Federated Fog Networks," IEEE Internet Things J, vol. 9, no. 2, pp. 1469–1478, 2022, doi: 10.1109/JIOT.2021.3088227.

[12] Z. Wu, B. Li, Z. Fei, Z. Zheng, B. Li, and Z. Han, "Energy-Efficient Robust Computation Offloading for Fog-IoT Systems," IEEE Trans Veh Technol, vol. 69, no. 4, pp. 4417–4425, 2020, doi: 10.1109/TVT.2020.2975056.

[13] L.-A. Phan, D.-T. Nguyen, M. Lee, D.-H. Park, and T. Kim, "Dynamic fog-to-fog offloading in SDN-based fog computing systems," Future Generation Computer Systems, vol. 117, pp. 486–497, 2021, doi: https://doi.org/10.1016/j.future.2020.12.021.

[14] C. Kai, H. Zhou, Y. Yi, and W. Huang,"Collaborative Cloud-Edge-End Task Offloading in Mobile-Edge Computing Networks With Limited Communication Capability," IEEE Trans Cogn Commun Netw, vol. 7, no. 2, pp. 624–634, 2021, doi: 10.1109/TCCN.2020.3018159.

[15] M. Al-khafajiy, T. Baker, H. Al-Libawy, Z. Maamar, M. Aloqaily, and Y. Jararweh, "Improving fog computing performance via Fog-2-Fog collaboration," Future Generation Computer Systems, vol. 100, pp. 266–280, 2019, doi: https://doi.org/10.1016/j.future.2019.05.015.

[16] Z. L. X. W. and Y. L. D. Wang, "Mobility-Aware Task Offloading and Migration Schemes in Fog Computing Networks," IEEE Access, vol. 7, pp. 43356–43368, 2019.

[17] U. Saleem, Y. Liu, S. Jangsher, Y. Li, and T. Jiang, "Mobility-Aware Joint Task Scheduling and Resource Allocation for Cooperative Mobile Edge Computing," IEEE Trans Wirel Commun, vol. 20, no. 1, pp. 360–374, 2021, doi: 10.1109/TWC.2020.3024538.

[18] Y. Jiang and D. H. K. Tsang, "Delay-Aware Task Offloading in Shared Fog Networks," IEEE Internet Things J, vol. 5, no. 6, pp. 4945–4956, 2018, doi: 10.1109/JIOT.2018.2880250.

[19] M. Chen and Y. Hao, "Task Offloading for Mobile Edge Computing in Software Defined Ultra-Dense Network," IEEE Journal on Selected Areas in Communications, vol. 36, no. 3, pp. 587–597, 2018, doi: 10.1109/JSAC.2018.2815360.

[20] I. E. A. C. and A. K. O. Salman, "IoT survey: An SDN and fog computing perspective," Comput. Netw, vol. 143, pp. 221–246, 2018.

[21] S. Misra and S. Bera, "Soft-VAN: Mobility-Aware Task Offloading in Software-Defined Vehicular Network," IEEE Trans Veh Technol, vol. 69, no. 2, pp. 2071–2078, 2020, doi: 10.1109/TVT.2019.2958740.

[22] C. Yang, Y. Liu, X. Chen, W. Zhong, and S. Xie, "Efficient Mobility-Aware Task Offloading for Vehicular Edge Computing Networks," IEEE Access, vol. 7, pp. 26652–26664, 2019, doi: 10.1109/ACCESS.2019.2900530.

[23] A. Bozorgchenani, D. Tarchi, and G. E. Corazza, "Mobile Edge Computing Partial Offloading Techniques for Mobile Urban Scenarios," in 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1–6. doi: 10.1109/GLOCOM.2018.8647240.

[24] G. Zhang, F. Shen, Z. Liu, Y. Yang, K. Wang, and M.-T. Zhou, "FEMTO: Fair and Energy-Minimized Task Offloading for Fog-Enabled IoT Networks," IEEE Internet Things J, vol. 6, no. 3, pp. 4388–4400, 2019, doi: 10.1109/JIOT.2018.2887229.

[25] Q. C. and J. Z. X. H. X. Yang, "Task Offloading Optimization for UAV-assisted Fog-enabled Internet of Things Networks," IEEE Internet Things J, 2021.

[26] S. Misra and N. Saha, "Detour: Dynamic Task Offloading in Software-Defined Fog for IoT Applications," IEEE Journal on Selected Areas in Communications, vol. 37, no. 5, pp. 1159–1166, 2019, doi: 10.1109/JSAC.2019.2906793.

[27] Sheldon M Ross, Introduction to Probability Models, 11th ed. Boston: USA: Academic Press, 2010.

[28] X. Liu, J. Zhang, X. Zhang, and W. Wang, "Mobility-Aware Coded Probabilistic Caching Scheme for MEC-Enabled Small Cell Networks," IEEE Access, vol. 5, pp. 17824–17833, 2017, doi: 10.1109/ACCESS.2017.2742555.

[29] A. Asensio et al., "Designing an efficient clustering strategy for combined Fog-to-Cloud scenarios," Future Generation Computer Systems, vol. 109, pp. 392–406, 2020, doi: https://doi.org/10.1016/j.future.2020.03.056.

[30] H. O. Hassan, S. Azizi, and M. Shojafar, "Priority, network and energy-aware placement of IoT-based application services in fog-cloud environments," IET Communications, vol. 14, no. 13, pp. 2117–2129, 2020, doi: https://doi.org/10.1049/iet-com.2020.0007.

[31] I. Lera, C. Guerrero, and C. Juiz, "Availability-Aware Service Placement Policy in Fog Computing Based on Graph Partitions," IEEE Internet Things J, vol. 6, no. 2, pp. 3641–3651, 2019, doi: 10.1109/JIOT.2018.2889511.

[32] A. Younis, B. Qiu, and D. Pompili, "Latency-aware Hybrid Edge Cloud Framework for Mobile Augmented Reality Applications," in 2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), 2020, pp. 1–9. doi: 10.1109/SECON48991.2020.9158429.

[33] Z. Ning, P. Dong, X. Kong, and F. Xia, "A Cooperative Partial Computation Offloading Scheme for Mobile Edge Computing Enabled Internet of Things," IEEE Internet Things J, vol. 6, no. 3, pp. 4804–4814, 2019, doi: 10.1109/JIOT.2018.2868616.

[34] H. Ma, P. Huang, Z. Zhou, X. Zhang, and X. Chen, "GreenEdge: Joint Green Energy Scheduling and Dynamic Task Offloading in Multi-Tier Edge Computing Systems," IEEE Trans Veh Technol, vol. 71, no. 4, pp. 4322–4335, 2022, doi: 10.1109/TVT.2022.3147027.

[35] 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.

[36] F. Dahan, "An Effective Multi-Agent Ant Colony Optimization Algorithm for QoS-Aware Cloud Service Composition," IEEE Access, vol. 9, pp. 17196–17207, 2021, doi: 10.1109/ACCESS.2021.3052907.

[37] A. Mseddi, W. Jaafar, H. Elbiaze, and W. Ajib, "Joint Container Placement and Task Provisioning in Dynamic Fog Computing," IEEE Internet Things J, vol. 6, no. 6, pp. 10028–10040, 2019, doi: 10.1109/JIOT.2019.2935056.