ITRC
ICT Research Institute
(Iran Telecommunication Research Center)

**Journal of Information and Communication Technology**

Iranian Society of Information and Communication Technology

Volume 16, Issue 59-60, Spring and Summer 2024, pp. 17-32

# Community-Based Multi-Criteria Placement of Applications in the Fog Environment

**Masomeh Azimzadeh[1], Ali Rezaee[1*], Somayyeh Jafarali Jassbi[1], Mehdi Esnaashari [2]**

[1] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[2] Faculty of Computer Engineering, K. N. Toosi University of Technology University, Tehran, Iran

Received: 27 November 2022, Revised: 04 January 2023, Accepted: 31 January 2023
Paper type: Research

**Abstract**

Fog computing technology has emerged to respond to the need for modern IoT applications for low latency, high security, etc. On the other hand, the limitations of fog computing such as heterogeneity, distribution, and resource constraints make service management in this environment challenging. Intelligent service placement means placing application services on fog nodes to ensure their QoS and effective use of resources. Using communities to organize nodes for service placement is one of the approaches in this area, where communities are mainly created based on the connection density of nodes, and applications are placed based on a single-criteria prioritization approach. This leads to the creation of unbalanced communities and inefficient placement of applications. This paper presents a priority-based method for deploying applications in the fog environment. To this end, balanced communities are created and applications are placed in balanced communities based on a multi-criteria prioritization approach. This leads to optimal use of network capacities and increases in QoS. The simulation results show that the proposed method improves deadline by up to 22%, increases availability by about 12%, and increases resource utilization by up to 10%.

**Keywords**: Application Placement, Internet of Things, Fog Computing.

[*] Corresponding Author's email: alirezaee@srbiau.ac.ir

دوفصلنامه
فناوری اطلاعات و ارتباطات ایران

سال شانزدهم، شماره های ۵۹ و ۶۰، بهار و تابستان ۱۴۰۳، صفحه ۱۷ الی ۳۲

# جایگذاری چندمعیاره مبتنی بر جوامع برنامه‌های کاربردی در محیط مه

**معصومه عظیم‌زاده[۱]، علی رضایی[۲*]، سمیه جعفرعلی‌جاسبی[۲]، مهدی اثنی عشری[۳]**

[۱] دانش آموخته دکتری دانشکده مهندسی کامپیوتر، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران

[۲] استادیار دانشکده مهندسی کامپیوتر، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران

[۳] استادیار دانشکده مهندسی کامپیوتر، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران، ایران

تاریخ دریافت: ۱۴۰۱/۰۹/۰۶ تاریخ بازبینی: ۱۴۰۱/۱۰/۱۴ تاریخ پذیرش: ۱۴۰۱/۱۱/۱۱

نوع مقاله: پژوهشی

## چکیده

فناوری محاسبات مه برای پاسخ به نیاز برنامه‌های کاربردی اینترنت اشیاء نظیر تاخیر کم، امنیت بالا و غیره ظهور پیدا کرد. از سویی محدودیت‌های محاسبات مه، نظیر ناهمگونی، توزیع شدگی و محدودیت منابع، مدیریت و استقرار یا جایگذاری برنامه‌ها در این محیط را دچار چالش می‌کند. جایگذاری هوشمند سرویس در محیط مه، باید منجر به تامین کیفیت سرویس و استفاده موثر از منابع گردد. یکی از رویکردهای جایگذاری برنامه‌ها، ایجاد جوامعی از گره‌های مه بر اساس چگالی اتصال آنها است که منجر به ایجاد جوامع نامتوازن شده و از سوی دیگر استفاده از روش تک معیاره برای اولویت‌بندی استقرار برنامه‌ها منجر به عدم جایگذاری موثر آنها می‌شود. در این مقاله روشی برای جایگذاری مبتنی بر اولویت برنامه‌های کاربردی در محیط مه ارائه شده است. روش پیشنهادی، با رویکردی مبتنی بر اولویت‌بندی چندمعیاره، برنامه‌ها را در جوامعی متوازن جایگذاری می‌کند. ایجاد جوامع متوازن منجر جایگذاری بهتر برنامه‌ها و استفاده هر چه بهتر از ظرفیت‌های شبکه می‌شود. همچنین جایگذاری مبتنی بر اولویت‌بندی چندمعیاره برنامه‌های کاربردی منجر به افزایش کیفیت برنامه‌ها و استفاده موثرتر از منابع موجود می‌گردد. نتایج شبیه‌سازی نشان‌دهنده افزایش ۲۲ درصدی تامین موعدزمانی، افزایش ۱۲ درصدی دسترس‌پذیری برنامه‌های کاربردی و همچنین افزایش ۱۰ درصدی میزان استفاده از منابع است.

**کلیدواژگان**: جایگذاری برنامه‌های کاربردی، اینترنت اشیاء، محاسبات مه.

* رایانامهٔ نویسنده مسؤول: alirezaee@srbiau.ac.ir

## ۱- مقدمه

محاسبات مه[1] در لبه شبکه برای پاسخ به نیاز کاربر انتهایی ظهور پیدا کرد. هدف اصلی این فناوری، پشتیبانی از نیازهای خاص برنامه‌های کاربردی حساس به تاخیر مانند واقعیت افزوده و اینترنت اشیاء[2] است که تولید کننده حجم زیادی از داده هستند[۱]. با توجه به ویژگی‌های محیط مه مانند محدودیت منابع، ناهمگونی و پویایی محیط از یک سو و ماهیت پیچیده و چندمولفه‌ای اینترنت اشیاء از سوی دیگر مدیریت سرویس در این محیط از اهمیت زیادی برخوردار است. در این رابطه جایگذاری سرویس[3] یکی از مباحث مهم در حوزه مدیریت سرویس[4] است. جایگذاری سرویس به معنای استقرار مناسب سرویس‌ها روی گره‌های مه است[۲]- [۳]. این جایگذاری باید به نحوی انجام شود که ضمن تامین نیازهای کیفیت سرویس برنامه‌های کاربردی اینترنت اشیا، از منابع مه نیز به خوبی استفاده شود [۵]-[۶].

یکی از رویکردهایی که در این حوزه برای مواجهه با چالش‌های مدیریت و جایگذاری سرویس وجود دارد، گروه‌بندی گره‌ها در قالب جوامع[5] است. ایده ایجاد جوامع در شبکه با ارائه نظریه گیروان نیومن[6] مبنی بر ماژولار بودن سیستم‌های پیچیده به یکی از ستون‌های اصلی تحقیقات این حوزه تبدیل شده است. این ادعا که ما می‌توانیم با تشخیص الگوهای ساختاری مهم در یک شبکه به درک عمیق‌تری از یک سیستم دست یابیم، تعداد زیادی از مطالعات در این زمینه را برانگیخته است[۴]، [۷] و[۱۰]- [۱۳]. روش‌های جایگذاری سرویسی که در حوزه مه مبتنی بر ایجاد جوامع پیشنهاد شده، عموما چگالی اتصال[7] گره‌ها و توپولوژی[8] را به عنوان معیار اصلی تشکیل جوامع قرار داده اند [۱۴]-[۱۸]. توجه صرف به میزان متصل بودن گره‌ها و توپولوژی منجر به ایجاد جوامع نامتوازن می‌شود. به دلیل اینکه کاربردهای اینترنت اشیا مدرن ساختار پیچیده و چند مولفه‌ای دارند، توزیع آنها در جوامع نامتوازن، منجر به پیچیدگی و تاخیر در تصمیم‌گیری نحوه جایگذاری شده و همچنین به دلیل پراکنده شدن سرویس‌های آنها در جوامع مختلف، با مشکلاتی نظیر افزایش تاخیر ارتباطی و کاهش دسترس‌پذیری مواجهه می‌شوند. در مقابل ترکیب ویژگی‌های گره‌ها و نودها منجر به کسب دانش بیشتر از محیط و ایجاد جوامع با کیفیت بالاتر می‌شود [۱۹]-[۲۰].

روش ارائه شده توسط لرا و همکاران [۲۷] تحت عنوان Partition، از جمله رویکردهایی است که از مفهوم جوامع برای جایگذاری برنامه‌های کاربردی در دستگاه‌های مه استفاده می‌کند. به این منظور از الگوریتم پارتیشن‌بندی گراف مبتنی بر نظریه Girvan-Newman استفاده می‌کند. سیاست آنها افزایش دسترس‌پذیری و کیفیت سرویس است که از طریق نگاشت برنامه‌های کاربردی به جوامع متشکل از دستگاه‌های مه و نگاشت سرویس‌های یک برنامه کاربردی به آن جوامع دنبال می‌شود. متکی بودن صرف به چگالی اتصال گره‌ها برای ایجاد جوامع، منجر به ایجاد جوامع ناهمگن می‌شود. بنابراین ضمن اینکه بخشی از ظرفیت گره‌ها غیرقابل استفاده شده یا کمتر مورد استفاده قرار میگیرد منجر به پراکندگی سرویس‌های مربوط به یک برنامه کاربردی نیز می‌شود. ضمن اینکه سرعت تصمیم‌گیری برای جایگذاری برنامه‌های کاربردی را کاهش می‌دهد. به همین منظور در این مقاله روش موثری برای جایگذاری برنامه‌های کاربردی اینترنت اشیاء تحت عنوان PDAF[9] ارائه شده است که در آن از هدف ایجاد جوامع متوازن به عنوان زیرساخت جایگذاری و در نهایت جایگذاری مبتنی بر اولویت برنامه‌های کاربردی است. روش PDAF برای توزیع متوازن ظرفیت‌های موجود در شبکه بین جوامع، از الگوریتم ژنتیک استفاده می‌کند. به این منظور ظرفیت‌های شبکه در سه دسته، ۱) میزان منابع گره‌های مه ۲) تعداد گره‌های موجود در محیط مه و ۳) ویژگی‌های اتصالی و ارتباطی گراف مه در نظر گرفته شده است. ایجاد جوامع متوازن منجر به تسریع تصمیم‌گیری در خصوص نحوه جایگذاری برنامه‌های کاربردی چندمولفه‌ای مرتبط با درخواست‌های متعدد موجود در محیط می‌شود. همچنین با جایگذاری هر برنامه کاربردی چندمولفه‌ای در یک جامعه واحد، تاخیر بین سرویس‌های یک برنامه کاربردی کاهش یافته و دسترس‌پذیری آنها افزایش می‌یابد.

جنبه دیگری که روش پیشنهادی PDAF توسعه داده است، استفاده از روش چندمعیاره برای اولویت‌بندی برنامه‌های کاربردی به منظور جایگذاری در جوامع است. در روش Partition و تعداد قابل توجهی از فعالیت‌های مرتبط صرفا از معیار موعد زمانی[10] برای اولویت‌بندی برنامه‌های کاربردی استفاده می‌کند [۱۷]، [۲۲]، [۲۴]، [۲۷] و[۳۰]. با توجه به ماهیت پراکنده درخواست‌های متعدد برنامه‌های کاربردی در محیط، اتکاء صرف به موعد زمانی منجر به عدم استفاده کارا از ظرفیت‌های شبکه برای جایگذاری برنامه‌ها می‌شود. روش

[1] Fog Computing
[2] Internet of Things
[3] Service Placement
[4] Service Management
[5] Communities
[6] Girvan-Newman's theory
[7] Connection Density
[8] Topology
[9] Priority based Deployment of IoT Applications in Fog
[10] Deadline

پیشنهادی PDAF، علاوه بر موعد زمانی، به نحوه پراکندگی برنامه‌های کاربردی نیز توجه دارد. در این خصوص معیار متوسط فاصله درخواست‌های هر برنامه کاربردی را نیز در اولویت‌بندی آنها در نظر گرفته است. به این ترتیب امکان برآورد نیاز تعداد بیشتری از برنامه‌های کاربردی با یک ویژگی که در نقاط مختلف پراکنده هستند فراهم شده است.

به طور کلی نوآوری‌های روش پیشنهادی شامل موارد زیر می‌شود:

۱. ایجاد زیرساخت جایگذاری مبتنی بر جوامع: ایجاد جوامع مبتنی بر روش ژنتیک با در نظر گرفتن پارامترهای مختلف موثر بر توزیع متوازن ظرفیت‌های شبکه منجر به استفاده بهینه از منابع موجود و افزایش کیفیت جوامع شده است.
۲. رتبه‌بندی چندمعیاره برنامه‌های کاربردی برای جایگذاری برنامه‌های کاربردی: استفاده از رتبه‌بندی چند معیاره برای جایگذاری برنامه‌های کاربردی منجر به جایگذاری تعداد بیشتری از برنامه‌های کاربردی در محیط مه شده ضمن اینکه نیازهای کیفیت سرویس برنامه‌های کاربردی را نیز لحاظ کرده است.

این پژوهش به دنبال ارائه راهکاری موثر برای بهره برداری موثرتر از زیرساخت مه برای پاسخ به نیازهای کیفیت سرویس برنامه‌های کاربردی اینترنت اشیاء می‌باشد. بنابراین به پرسش‌هایی نظیر نحوه ایجاد زیرساخت جایگذاری به منظور غلبه بر پیچیدگی این محیط و سیاست‌های لازم برای بهره برداری موثر از زیرساخت جایگذاری برای تامین نیازهای کیفی برنامه‌های کاربردی و استفاده بهینه از منابع موجود می‌پردازد. بنابراین به طور خلاصه باید در این تحقیق به سوالات زیر پاسخ داد:

- ایجاد جوامع به کمک ترکیب اطلاعات ساختاری و ویژگی موجودیت‌های آن شامل گره‌ها و نودها چه تاثیری بر جایگذاری برنامه‌های کاربردی خواهد داشت؟
- سیاست استفاده از جوامع ایجاد شده در مرحله جایگذاری سرویس‌ها به چه نحو باشد که از جوامع به نحو موثرتری بهره برداری شود؟
- معیار اولویت‌بندی جوامع برای جایگذاری سرویس‌ها به چه صورت تعریف شود؟
- استفاده از ترکیبی از چند فاکتور برای اولویت‌بندی برنامه‌های کاربردی چه تاثیری بر کارایی روش جایگذاری خواهد داشت؟

فرضیات این پژوهش عبارتند از:

- هر برنامه کاربردی توسط چندین کاربر یا درخواست‌دهنده فراخوانی می‌شوند.
- درخواستهای هر برنامه کاربردی در مکانهای جغرافیایی مختلف توزیع شده‌اند.
- امکان خرابی گره‌ها و افزایش بار کاری در نظر گرفته شده است.
- در صورت محدودیت منابع محیط مه برای جایگذاری برنامه‌های کاربردی، امکان انتقال آنها به ابر وجود دارد.
- کمیت‌های همه منابع و اجزای محیط در هر اجرای برنامه، به صورت تصادفی و در بازه مقادیر تعریف شده تعیین می‌شوند.

در ادامه این مقاله ابتدا در بخش ۲ به مرور اجمالی کارهای مرتبط پرداخته شده است. سپس در بخش ۳ فضای مساله تعریف گردیده است. بخش ۴ اختصاص به روش پیشنهادی دارد که در دو مرحله ایجاد جوامع برای سازماندهی زیرساخت جایگذاری و جایگذاری سرویس مبتنی بر اولویت برنامه‌های کاربردی ارائه شده است. در بخش ۵ روش پیشنهادی مورد آزمون و ارزیابی قرار گرفته است. در بخش ۶ به تحلیل نتایج بدست آمده بخش ۵ پرداخته شده است. بخش ۷ به محاسبه پیچیدگی زمانی روش پیشنهادی اختصاص دارد. در انتها جمع‌بندی روش پیشنهادی و ارائه فعالیت‌های آتی در قالب بخش ۸ انجام شده است.

## ۲- کارهای مرتبط

فعالیت‌های مختلفی در زمینه جایگذاری سرویس در محیط مه صورت گرفته است. دسته‌بندی‌های مختلفی در این زمینه قابل ارائه است. با توجه به تمرکز این مقاله بر ایجاد جوامع، کارهای مرتبط از منظر نحوه پرداختن به زیرساخت جایگذاری مورد بررسی قرار گرفته‌اند. از این منظر روش‌های مطرح در حوزه جایگذاری سرویس، عمدتا شامل دو دسته هستند. دسته اول صرفا به سیاست جایگذاری و نحوه استقرار برنامه‌های کاربردی روی گره‌های مه می‌پردازند[۲۱]- [۲۴]. دسته دوم در گام اول زیرساخت جایگذاری را فراهم کرده و در گام دوم برنامه‌های کاربردی را روی آنها مستقر می‌کنند [۲۵]- [۲۹] و [۱۴]- [۱۸].

از جمله فعالیت‌هایی که تنها متکی به سیاست جایگذاری هستند می‌توان به موارد زیر اشاره نمود. عباسی و همکاران یک الگوریتم ژنتیک برای تخصیص بار کاری در محیط مه پیشنهاد داده‌اند[۲۱]. آنها سعی در بهبود مصرف انرژی و کاهش تاخیر دارند. ردی و همکاران با چرخه‌های خواب و بیداری هوشمند گره‌های مه، به حداقل رساندن انرژی در لایه مه دنبال می‌کنند[۲۲]. بنابراین آنها با حداقل تعداد گره‌های مه فعال به درخواست‌ها پاسخ می‌دهند.

ناتاشا و گودتی یک راه حل بهینه‌سازی چندهدفه برای قرار دادن خدمات در محیط مه ارائه می‌کنند[۲۳]. هدف آنها به حداقل رساندن تاخیر خدمات، هزینه و مصرف انرژی است. التراونه از یک الگوریتم ژنتیک برای قرار دادن ماژول‌های برنامه بر روی دستگاه‌های مه استفاده می‌کند[۲۴]. به این منظور سطوح بحرانی برنامه‌ها و الزامات امنیتی در نظر گرفته می‌شود.

ویجویه و همکاران [۳۸] به مساله جایگذاری سرویس با یک رویکرد برنامه ریزی خطی عدد صحیح (ILP) برای استقرار برنامه‌های کاربردی در شبکه زیرساخت و هدایت ترافیک از دستگاه‌های پایانی به برنامه‌های کاربردی مستقر پرداخته است. اهداف اصلی آنها تامین نیازهای مختلف کاربران و به حداکثر رساندن سود ارائه‌دهنده زیرساخت خدمات بود.

سریراگاوندرا و همکاران [۳۹] روشی برای استفاده از منابع مه با رعایت موعد زمانی برنامه‌های کاربردی پیشنهاد داده‌اند. آنها از الگوریتم ژنتیک برای جایگذاری سرویس در محیط مه استفاده می‌کنند. این پژوهش زمان جایگذاری سرویس در لایه‌های مختلف ابر و مه را تحلیل می‌کند و در مورد جایگذاری سرویس برنامه‌های کاربردی اینترنت اشیا در لایه‌های مختلف تصمیم‌گیری می‌کند.

فعالیت‌هایی که به صورت دو مرحله‌ای به جایگذاری سرویس‌های اینترنت اشیا می‌پردازند، برای سازماندهی گره‌ها دو رویکرد کلی را در پیش می‌گیرند. رویکرد اول گروه‌بندی گره‌ها بر اساس شاخص‌هایی مانند فاصله از یک گره مرکزی[1] یا قرار گرفتن در یک دامنه سرویس[2] و غیره انجام می‌شود [۱۴]-[۱۸]، [۲۶] و [۲۸]. در رویکرد دوم که عمدتا در قالب شبکه‌های پیچیده پویا[3] دنبال می‌شود، گره‌ها بر اساس چگالی اتصال خوشه‌بندی شده و در قالب جوامع سازماندهی می‌شوند [۲۵]، [۲۷]، [۲۹] و [۲۸].

یوسف پور و ایشیگاکی، گره‌های مه را بر اساس حوزه عملیاتی یا نیازهای خاص برنامه‌ها گروه‌بندی می‌کنند [۱۷]. ایده آنها پاسخ به نیاز درخواست‌های سرویس با تأخیرکم در محیط مه و پاسخ به سایر درخواست‌ها در ابر است. کیموفسکی و همکاران از نظریه گراف برای قرار دادن خدمات استفاده کرده‌اند[۲۶]. آنها برای افزایش سرعت تصمیم‌گیری و ارائه روش مدیریت منابع تطبیقی، به تعریف یک معماری مه پرداخته‌اند. لرا و همکاران محیط را از طریق شاخص‌های مرکزیت مدل‌سازی می‌کند تا دستگاه‌های مه نزدیک به حسگرها[4] را برای بهبود استفاده از منابع تعیین کنند[۲۶]. اسکارلات و همکاران در چند پژوهش از کولونی‌های مه برای تخصیص خدمات اینترنت اشیا به منابع مه استفاده کرده‌اند[۱۶] [۳۰]. آنها کولونی‌ها را به عنوان مراکز داده میکرو[5] می‌دانند که از تعداد دلخواه سلول مه تشکیل شده است. این مستعمرات مراکز میکرو داده ای هستند که از تعداد دلخواه سلول مه تشکیل شده اند. هدف به حداکثر رساندن کارایی منابع در مه و در عین حال برآورده کردن مهلت هر برنامه است. اسکارلات و همکاران تمایز بین مهلت درخواست و زمان استقرار آن را به عنوان یک معیار اولویت‌بندی اعمال کرد[۱۶]. زمان استقرار به زمان انتظار برنامه قبل از تخصیص صحیح به منابع محاسباتی اشاره دارد. اسکارلات و همکاران در مطالعه دیگری، سرویس‌های اینترنت اشیا را روی منابع مه با در نظر گرفتن محدودیت‌های QoS[6] مستقر می‌کنند[۳۰]. الخطیب و همکاران همچنین از قابلیت‌های محاسباتی میکروکلاد برای ارائه خدمات مه به منظور کاهش تأخیر استفاده می‌کنند [۱۵]. بارانوال و ویدیارتی [۴۰] یک روش جایگذاری سرویس را بر اساس انتخاب برخی از گره‌های ارکستراتور مه پیشنهاد می‌کنند. آنها از گره‌های ارکستراتور مه توزیع شده برای تخصیص منصفانه منابع محاسباتی بین گره‌های مه برای بهبود کیفیت خدمات (QoS) استفاده می‌کنند.

فیلیپوسکا و همکاران اولین روش استفاده از جوامع دستگاه‌های مه برای مدیریت منابع را پیشنهاد داده‌اند[۲۹]. آنها از معیار تعداد گام بین ماشین‌های مجازی برای تعریف جوامع استفاده می‌کنند. نتیجه این موضوع افزایش کارایی ارتباطی و کاهش مصرف انرژی است. لرا و همکاران در مطالعه دیگری از ویژگی شبکه پیچیده برای سازماندهی جوامعی از گره‌های مه برای قرار دادن خدمات استفاده می‌کنند[۲۷]. آنها از معیار مرکزیت بینابینی برای ایجاد مجموعه‌ای از دستگاه‌های به خوبی متصل، برای بهبود در دسترس بودن خدمات استفاده می‌کنند. ولاسکوئز و همکاران بر اساس امکان اشتراک بار دروازه بین گره‌ها، آنها را گروه‌بندی می‌کنند [۲۵]. به این منظور، آنها گره‌ها را برای ایجاد جوامع رتبه‌بندی می‌کنند تا جوامعی را شکل دهند که دارای گره‌هایی با بالاترین احتمال انتقال باشند.

روش‌های دسته اول فاز ایجاد زیرساخت جایگذاری را نادیده می‌گیرند، که این امر منجر به عدم استفاده بهینه از گره‌های مه و تامین نیازهای کیفیت سرویس برنامه‌های کاربردی می‌شود. در حالی که یکی از فازهای اصلی روش پیشنهادی PDAF ایجاد زیرساخت جایگذاری است. روش‌های دو مرحله‌ای که دارای دو فاز ایجاد زیرساخت جایگذاری و جایگذاری سرویس هستند، برای ایجاد

[1] Central Node
[2] Service Domain
[3] Dynamic Complex Networks
[4] Sensors
[5] Micro Datacenters
[6] Quality of Service

جوامع تنها به تحلیل ساختار یا استفاده از ویژگی‌های ساختاری محدود می‌شوند. در حالی که روش پیشنهادی PDAF از تحلیل ساختار در کنار تحلیل ویژگی‌های گره‌ها و لینک‌ها برای ایجاد جوامع با کیفیت استفاده می‌کنند.

ضمن اینکه روش پیشنهادی PDAF برخلاف اغلب روش‌های ذکر شده که مبتنی بر روش تک معیاره برای الویت‌بندی جایگذاری برنامه‌های کاربردی هستند، از ترکیب چندین معیار استفاده می‌کند.

## ۳- تعریف فضای مساله

محیط مه به صورت گراف نشان داده شده است که در آن دستگاه‌های مه گره‌های گراف محسوب شده با نماد F نشان داده می‌شوند و لینک بین دستگاه‌ها به عنوان لبه‌های گراف با نماد E نشان داده شده است. این موضوع در معادله ۱ نشان داده شده است.

$$G=(F, E) \quad (۱)$$

هر گره مه موجود در محیط با نماد $f_i$ نشان داده شده است. در نهایت مجموعه گره‌ها موجود در محیط طبق معادله ۲ تعریف شده است.

$$F = \{f_1,f_2,\ldots,f_m\} \quad (۲)$$

یال‌های گراف اتصالی نیز به عنوان پیوندهای موجود بین گره‌های مه در نظر گرفته شده است که معادله ۳ نشان‌دهنده این رابطه است.

$$E = \{ e_{ij} \mid i,j\epsilon[ 1,m] , i\neq j , e_{ij} = <f_i,f_j> \} \quad (۳)$$

طبق تعریف جوامع به صورت زیرگراف‌هایی از گراف اصلی (G) در نظر گرفته شده است که اجتماع آنها شکل‌دهنده گراف اصلی است. معادله ۴ نشان‌دهنده k جامعه موجود در محیط است.

$$F = \{f_1,f_2,\ldots,f_m\} \quad (۴)$$

در معادله ۵ هر زیرگراف شامل مجموعه ای از گره‌ها ($CF_i$) و مجموعه ای از یال‌ها ($CE_i$) است که هر کدام از آنها خود زیر مجموعه گراف اصلی فرض شده در محیط مه هستند.

$$C_i = (CF_i, CE_i) \quad (۵)$$

معادله ۶ نشان‌دهنده گره‌های زیرگراف فرضی $CF_i$ است. با فرض اینکه زیرگراف مذکور دارای s گره باشد.

$$CF_i = \{cf_{i1}, cf_{i2}, \ldots, cf_{is}\} \quad (۶)$$

همچنین معادله ۷ نشان‌دهنده پیوندهای بین گره‌های هر جامعه است.

$$CE_i = \{ ce_{ij} \mid i,j\epsilon[ 1,s] , i\neq j , ce_{ij} = <cf_i,cf_j>j\} \quad (۷)$$

در فضای مساله علاوه بر گره‌های مه و گراف ارتباطی آن مجموعه برنامه‌های کاربردی و درخواست‌های متناظر با آنها نیز وجود دارد. در واقع هر برنامه کاربردی ممکن است توسط چندین کاربر یا مصرف کننده درخواست شده باشد. این کاربران ممکن است در نقاط مختلف محیط پراکنده شده باشند. علاوه بر تعریف گراف مه و جوامع دو مفهوم مهم دیگر که از عناصر مهم محیط هستند برنامه‌های کاربردی و درخواست‌های متناظر آنها هستند. معادله ۸ نشان‌دهنده مجموعه‌ای از برنامه‌های کاربردی است. همچنین در معادله ۹ مجموعه درخواست‌ها نشان داده شده است. طبق این معادله هر درخواست وارد شده به محیط، یک برنامه کاربردی را فراخوانی می‌کند.

$$A = \{a_1, a_2, a_3, \ldots, a_p \} \quad (۸)$$

$$R=\{r_j^{a_i} \mid a_i \text{ is Requested by } r_j, i \epsilon [0, p], j \epsilon [0, r]\} \quad (۹)$$

## ۴- روش پیشنهادی جایگذاری چند معیاره مبتنی بر جوامع برنامه‌های کاربردی در محیط مه

در این بخش به معرفی روش پیشنهادی PDAF می‌پردازیم. روش پیشنهادی در دو مرحله پیاده‌سازی شده است در مرحله اول زیرساخت جایگذاری در قالب جوامع سازماندهی می‌شود و بعد از آن برنامه‌های کاربردی اینترنت اشیاء بعد از رتبه‌بندی در این جوامع جایگذاری می‌شوند.

در مرحله ایجاد زیرساخت جایگذاری، جوامع مبتنی بر روش ژنتیک با ترکیبی از اطلاعات ساختاری و ویژگی گره‌ها و لینک‌ها سازماندهی می‌شوند تا جوامع متوازنی ایجاد گردد. به این منظور بعد از مقداردهی اولیه به کروموزوم‌ها به عنوان جوامع بذر این کروموزوم‌ها وارد چرخه ژنتیک می‌شوند و بعد از طی مراحل تعیین شده در چرخه ژنتیک، بهترین کروموزوم، متشکل از جوامع متوازن و باکیفیت، به عنوان ورودی به مرحله جایگذاری سرویس داده می‌شود. در مرحله جایگذاری سرویس نیز ابتدا برنامه‌های کاربردی مبتنی بر روش چند معیاره اولویت‌بندی شده و سپس به عنوان ورودی به الگوریتم جایگذاری سرویس داده می‌شوند. در الگوریتم جایگذاری سرویس برنامه‌های الویت‌بندی شده به ترتیب مورد بررسی قرار می‌گیرند. به ازای هر درخواست برنامه کاربردی، بهترین و نزدیکترین جامعه برای جایگذاری شناسایی شده و در صورت عدم وجود برنامه کاربردی در آن جامعه، نسخه جدیدی از آن در جامعه مورد نظر جایگذاری می‌شود و به درخواست کاربر پاسخ داده می‌شود.

در بخش ۴-۱ مرحله ایجاد زیرساخت جایگذاری تشریح گردیده است و بخش ۴-۲ به مرحله جایگذاری سرویس مبتنی بر اولویت اختصاص دارد.

## ۴-۱- ایجاد جوامع به عنوان زیرساخت جایگذاری

در این مقاله ایجاد جوامع مبتنی بر روش ژنتیک انجام شده است. به این منظور هر راه‌حل مساله که شامل تعریف جوامع متوازن است در قالب یک کروموزوم دیده می‌شود. بنابراین هر جمعیت در این روش مجموعه‌ای از راه‌حل‌های شکل‌گیری جوامع است. برای مقداردهی اولیه به کروموزومها، به هر ژن کروموزوم شناسه یک گره همسایه به صورت تصادفی اختصاص می‌یابد. سپس گره‌های همسایه پرتکرار به عنوان سرخوشه انتخاب می‌شوند و سایر گره‌ها حول آنها جوامع اولیه را تشکیل می‌دهند. بعد از آن مجموعه کروموزوم‌های اولیه وارد سیکل ژنتیک شده و بهترین آنها مبتنی بر نمره کسب شده از تابع برازش به عنوان کروموزوم‌های پدر انتخاب می‌شوند. کروموزوم‌های پدر تحت عملگرهای تقاطع و جهش قرار گرفته تا کروموزوم‌های جدیدی به عنوان فرزندان تولید شوند. بعد از آن برای سیکل بعدی ژنتیک، مجموعه‌ای از بهترین کروموزوم‌های پدر و فرزند به عنوان کروموزوم‌های سیکل بعدی الگوریتم مورد استفاده قرار می‌گیرند. روند ذکر شده تا رسیدن به بهترین کروموزوم بر اساس امتیاز تخصیص داده شده توسط تابع برازش، که نماینده بهترین مجموعه جوامع ایجاد شده است ادامه می‌یابد.

- عملگر تقاطع:

عملگر تقاطع دو نقطه‌ای به عنوان اولین فرآیند برای ایجاد فرزندان استفاده می‌شود. بدین منظور جمعیت به دو قسمت مساوی تقسیم شده و در هر مرحله از هر قسمت از جمعیت یک کروموزوم به طور تصادفی انتخاب می‌شود. سپس دو نقطه تقاطع به صورت تصادفی روی هر کروموزوم انتخاب می‌شود. به این ترتیب هر کروموزوم را می‌توان به سه بخش تقسیم کرد و محتویات کروموزوم‌ها شامل شناسه جوامع تخصیص داده شده در این سه بخش انتخاب شده جابجا می‌شوند.

- عملگر جهش:

عملگر جهش عملگر اصلی الگوریتم ژنتیک است که به طور تصادفی ژن‌های کروموزوم را تغییر می‌دهد. این عملگر منجر به ایجاد تنوع بیشتر در کروموزوم‌ها و همگرایی سریعتر به راه‌حل مسئله می‌شود. برای عملیات جهش، یک نقطه به صورت تصادفی روی هر کروموزوم انتخاب شده و شماره جامعه آن با یک شماره جامعه که به صورت تصادفی انتخاب شده، جایگزین می‌شود.

- تابع برازش:

در مرحله ایجاد جوامع همانطور که پیشتر اشاره شد، هدف بهره برداری هر چه بهتر از ظرفیت‌های شبکه است. به این منظور ظرفیت شبکه در قالب سه پارامتر زیر در تعریف جوامع در نظر گرفته شده است:

۱. متوسط میزان منابع تخصیصی به هر جامعه ($R_C$)
۲. متوسط تعداد گره‌های توزیع شده بین جوامع ($N_C$)
۳. میزان متصل بودن جوامع ($C_C$)

به منظور توزیع متوازن ظرفیت شبکه بین جوامع این پارامترها در تابع برازش مورد استفاده قرار گرفته اند که در ادامه به تشریح آن می‌پردازیم.

متوسط منابع تخصیص داده شده به ازای هر جامعه، در معادله ۱۰ نشان داده شده است. در این معادله x متغیر است که متناسب با اینکه با چه منبعی روبرو هستیم، می‌تواند با RAM، IPT و TB جایگزین شود که به ترتیب نشان‌دهنده متوسط RAM، CPU و فضای ذخیره‌سازی تخصیصی به هر جامعه است.

$$C_i^{\,x} = \frac{\sum_{j=0}^{|CF_i|} cf_{ij}^{\,x}}{\sum_{m=0}^{|FN|} fn_m^{\,x}} \mid x = RAM, TB, IPT \qquad (۱۰)$$

در نهایت اولین معیار که متوسط منابع استفاده شده توسط هر جامعه است، توسط معادله ۱۱ محاسبه می‌شود:

$$R_C = \frac{\sum_{i=0}^{|C|} (\alpha_* C_i^{RAM} + \beta_* C_i^{TB} + \gamma_* C_i^{IPT})}{|C|} \qquad (۱۱)$$

به منظور محاسبه متوسط تعداد گره‌های توزیع شده بین جوامع ابتدا معیار توازن ($\rho$) محاسبه شده است. به این منظور مطابق با معادله ۱۲ تعداد کل گره‌ها بر تعداد جوامع تقسیم شده است. بعد نسبت تعداد گره‌های هر جامعه را به معیار توزان برای دستیابی به هدف دومین معیار که توزیع متوازن محاسبه می‌شود (معادله ۱۳ ).

$$\rho = \frac{|FN|}{|C|} \qquad (۱۲)$$

$$N_C = \frac{\sum_{i=0}^{|C|} \frac{C_k}{\rho}}{|C|} \qquad (۱۳)$$

به منظور بررسی میزان متصل بودن یک جامعه، وجود بزرگترین مولفه متصل به عنوان مولفه متصل شاخص و میزان مشارکت گره‌های تخصیص داده شده در مولفه متصل بررسی می‌شود. در صورتیکه گره‌های تخصیص داده شده به یک جامعه، مولفه متصلی ایجاد نکنند، به عنوان جامعه درنظر گرفته نمی‌شوند. همچنین هر

چه نسبت تعداد گره‌های مشارکت کننده در مولفه متصل شاخص بیشتر باشد، آن جامعه امتیاز بیشتری کسب می‌کند. بنابراین در معادله ۱۴ نسبت میزان متصل بودن جوامع به کل جوامع حساب می‌شود. حالت ایده‌آل آن است که این مقدار برابر ۱ باشد.

$$C_C = \frac{\sum_{i=0}^{|Com|} \frac{|Largest_Component(C_k)|}{C_k}}{|C|} \qquad (۱۴)$$

در نهایت فرمول برازش مطابق با معادله ۱۵ به عنوان ترکیب وزنی سه معیار متوسط منابع استفاده شده، متوسط گره‌های توزیع شده و میزان متصل بودن جوامع محاسبه می‌شود:

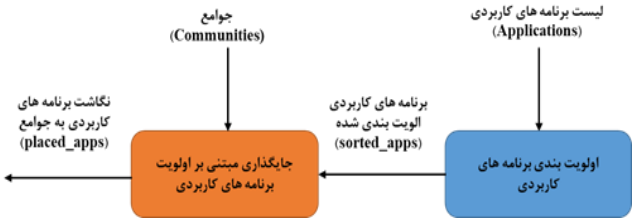
$$f(x) = \lambda_1 * R_C + \lambda_2 * N_C + \lambda_3 * C_C \qquad (۱۵)$$

لازم به ذکر است که مقادیر پارامترهای وزنی در جدول ۱ ارائه گردیده است.

## ۴-۲- جایگذاری سرویس مبتنی بر اولویت

در این مرحله برای جایگذاری برنامه کاربردی، نزدیکترین جامعه که دارای ظرفیت کافی برای جایگذاری سرویس باشد انتخاب می‌شود. برای جایگذاری سرویس ابتدا برنامه‌های کاربردی را برای جایگذاری اولویت‌بندی می‌کنیم. این کار به کمک الگوریتم ۱ انجام می‌شود. مطابق شکل ۱، بعد از اولویت‌بندی برنامه‌های کاربردی، لیست برنامه‌های کاربردی اولویت‌بندی شده به عنوان ورودی به الگوریتم ۲ داده می‌شود. الگوریتم ۲ وظیفه جایگذاری برنامه‌های کاربردی در جوامع شکل گرفته در بخش ۴-۱ را بعهده دارد. بنابراین ورودی دیگری که الگوریتم ۲ دریافت می‌کند، لیست جوامعی است که در بخش ۴-۱ ایجاد شده‌اند. الگوریتم ۲ به کمک این دو ورودی به جایگذاری برنامه‌های کاربردی مبتنی بر اولویت در جوامع می‌پردازد.

به این منظور ترکیب معیار موعد زمانی ($dl_{app}$) و متوسط فاصله درخواست‌ها ($ad_{app}$) در نظر گرفته شده است. بنابراین طبق الگوریتم، ابتدا برنامه‌های کاربردی را مطابق خط ۲ بر اساس معیار متوسط فاصله درخواست‌ها به صورت نزولی مرتب کرده، سپس در هر مرحله از الگوریتم، طبق خط ۶ برنامه کاربردی با حداقل موعد زمانی ($app_j$) را در لیست پیدا کرده و نسبت موعد زمانی برنامه کاربردی موجود در سر لیست مرتب شده ($app_i$) را نسبت به برنامه کاربردی با حداقل موعد زمانی می‌سنجیم (خط ۸). اگر این نسبت از آستانه‌ای کمتر باشد، برنامه کاربردی با متوسط فاصله درخواست بیشتر اولویت داده شده و به لیست اولویت‌بندی نهایی (sorted-apps) اضافه می‌شود و از لیست مرتب شده کنونی حذف می‌شود.




**شکل ۱. مراحل جایگذاری برنامه‌های کاربردی در جوامع**

این مطالب در خطوط ۱۱ و ۱۲ الگوریتم نشان داده شده است. در غیراینصورت مطابق با خطوط ۱۴ و ۱۵، برنامه کاربردی با حداقل مهلت زمانی به لیست نهایی اضافه شده و از لیست مرتب شده فعلی حذف می‌گردد. در نهایت لیست نهایی ایجاد شده از برنامه‌های کاربردی، برای جایگذاری در نزدیکترین جامعه دارای فضای کافی مورد استفاده قرار می‌گیرد.

لازم به ذکر است که نسخه یک روش برنامه‌های کاربردی بر اساس متوسط فاصله درخواست‌ها به صورت نزولی مرتب شده‌اند و در نسخه دو روش به صورت صعودی. بنابراین تاثیر هر دو حالت زیاد بودن متوسط فاصله درخواست‌ها و کم بودن متوسط فاصله درخواست‌ها در این روش ارزیابی شده است.

**جدول ۱. مقداردهی به پارامترهای محیطی و روش**

| عنوان | پارامتر | مقدار | توصیف |
|---|---|---|---|
| لینک | BW | $6*10^6$-$6*10^7$ | پهنای باند(bit/s) |
| | PD | 3-5 | زمان انتشار(ms) |
| گره | Fog | 100 | تعداد |
| | RAM | 10-25 | حافظه (MB) |
| | IPT | 100-1000 | سرعت (Instr/ms) |
| | TB | (0.2-100) | فضای ذخیره‌سازی (Terabyte) |
| برنامه کاربردی | Deadline (ms) | 2600-6600 | موعد زمانی(ms) |
| | Service(number) | 2-10 | Service (number) |
| | Resource | 1-6 | res. Units |
| | Packet | 1,500,000–4,500,000 | size (bytes) |
| جایگذاری برنامه کاربردی | $threshold_{pr}$ | 0.9 | اولویت‌بندی برنامه-های کاربردی |
| | $threshold_{pl}$ | 0.6 | انتخاب جامعه |
| ایجاد جوامع | $\lambda_1$ | 0.4 | تخصیص منابع |
| | $\lambda_2$ | 0.25 | توزیع گره‌ها |
| | $\lambda_3$ | 0.35 | میزان متصل بودن جوامع |

**Algorithm 1: Application Prioritization**
**Input: Applications**
**Output: Sorted Applications (sorted_apps)**

```
# Sort applications based on ad_app
sapps = Sort (applications, ad_app, descending)
# Create final prioritization list
for each app_i in sapps do:
    # Find the application with minimum deadline (app_j)
    app_j = Find (applications, min (dl_app))
    # The deadline ratio of two applications
    ratiodl = dl_app_j / dl_app_i
    # Create final list of prioritized applications
    if ratiodl <= threshold_pr then:
      sorted_apps.append (appi)
      sapps.remove (appi)
    else:
      sorted_apps. append (appj)
      sapps.remove (appj)
Return (sorted_apps)
```

**Algorithm 2: application_placement**
**Input: Communities, sorted_apps**
**Output: placed_apps**

```
for each app in sorted_apps do:
  # Find list of the application requests
  reqs = find_requests (app)
  # Place app in community per request
  for each req in reqs do:
    # find near communities to req
    comms = distance (communities, req)
    # sort communities based on ad_app
    Sort (comms, distance, ascending);
    Place = False;
    for each com in comms do:
      # check exitance of app in community
      if (app is in com) then:
        Place = True;
        Break;
      if required_resources(app) <= resources (com)
        and distance (com)<= threshold_pl then:
        place_apps (app, com);
        Update (com, resources)
        Place = True;
        Break;
```

مطابق الگوریتم ۲، برای جایگذاری برنامه‌های کاربردی دو ورودی از مراحل قبلی دریافت می‌شود که یکی لیست جوامع ایجاد شده در فاز ایجاد زیرساخت جایگذاری است. همچنین لیست برنامه‌های کاربردی مرتب‌سازی شده از الگوریتم ۱ دریافت می‌شود. سپس برنامه‌های کاربردی برای جایگذاری به ترتیب از سرلیست انتخاب می‌گردند. به ازای هر برنامه کاربردی درخواست‌های متناظر آن مطابق خط ۳ شناسایی شده و در لیست جداگانه نگهداری می‌شود. درخواست‌های مربوط به برنامه کاربردی در گام بعدی یکی یکی بررسی می‌شوند. به ازای هر درخواست نزدیکترین جوامع به آن شناسایی می‌شوند (خط ۷). نزدیکی جوامع بر اساس تعداد هاب‌های بین درخواست‌دهنده و آن جامعه محاسبه می‌شود. مطابق خط ۹ جوامع نزدیک به درخواست‌دهنده بر اساس فاصله به صورت صعودی در لیستی مرتب می‌شود. بنابراین جوامع نزدیکتر در اولویت بالاتری قرار می‌گیرند.

در مرحله بعد جوامع برای جایگذاری برنامه کاربردی، به ترتیب اولویت از دو منظر مورد بررسی قرار می‌گیرند. یکی از منظر وجود یا عدم وجود نسخه‌ای از برنامه در آن و جنبه دوم وجود منابع کافی در آن جامعه. در وحله اول وجود یا عدم وجود نسخه‌ای از برنامه در آن جامعه مورد بررسی قرار می‌گیرد. در صورتی که نسخه‌ای از برنامه در جامعه موجود بود به درخواست اختصاص یافته و فرآیند جایگذاری به آن درخواست مطابق خطوط ۱۳ تا ۱۵ پایان می‌یابد. در غیر اینصورت وجود منابع کافی در آن جامعه بررسی شده و در صورت احراز شرایط، برنامه کاربردی در آن جامعه جایگذاری شده و منابع مربوطه بروزرسانی می‌شوند (خطوط ۱۶ تا ۲۰).

اگر برنامه کاربردی تا این مرحله هنوز به نسخه مد نظر دسترسی نداشته باشد، سایر جوامع نزدیک مورد بررسی قرار می‌گیرند. لازم به ذکر است که انتخاب جوامع نزدیک تا آستانه مشخصی ادامه می‌یابد و در نهایت برنامه کاربردی در ابر جایگذاری می‌شود.

## ۵- ارزیابی تجربی

### ۵-۱- محیط شبیه‌سازی

برای ارزیابی روش پیشنهادی از نرم‌افزار شبیه‌سازی YAFS استفاده شده است. این نرم‌افزار شبیه‌ساز محیط مه بوده و در کارهای مرتبط مورد استفاده قرار گرفته است. YAFS همچنین ویژگی‌های مبتنی بر گراف را دارد و از ویژگی‌های حیاتی محیط مه پشتیبانی می‌کند. همچنین متن باز بوده و کد منبع آن قابل دسترس می‌باشد [۳۲]. در جدول ۱ مقداردهی به ویژگی‌های محیط نظیر ویژگی لینک‌ها، گره‌ها و برنامه‌های کاربردی و همچنین پارامترهای روش ارائه شده است.

لازم به ذکر است که در جدول ۱، بازه مقادیر مربوط به ویژگی‌های محیطی شامل لینک، گره و برنامه کاربردی طبق مقاله پایه مورد مقایسه [۲۷] مشخص شده است. همچنین مقادیر مربوط به آستانه‌های جایگذاری برنامه کاربردی و وزندهی به مقادیر در بخش ایجاد جوامع بر اساس آزمایش‌های متعدد و به صورت تجربی تعیین گردیده است.

## ۵-۲- معیارهای ارزیابی

برای ارزیابی روش پیشنهادی معیارهای مختلفی در نظر گرفته شده است. معیارهای درنظر گرفته شده در جهت ارزیابی سنجش کیفیت سرویس و میزان منابع استفاده شده تعریف شده است. از جمله معیارهای سنجش تامین کیفیت سرویس، می‌توان به میزان تامین موعد زمانی برنامه‌های کاربردی، تاخیر پاسخ به آنها و دسترس‌پذیری آنها برای درخواست‌دهنده‌ها اشاره نمود. همچنین در راستای سنجش میزان استفاده از منابع نیز معیارهای مربوطه تعریف شده و مورد استفاده قرار گرفته است.

- **میزان رعایت موعد زمانی:**

با استفاده از این معیار، نسبت برنامه‌های کاربردی که پیش از سررسید موعد زمانی پاسخ خود را دریافت کرده‌اند به کل برنامه‌های کاربردی محاسبه می‌شود. این محاسبه در معادله ۱۶ نمایش داده شده است.

$$\mu = \frac{\sum_{i=0}^{|A|}( x_i=1|if(responsetime(a_i)\leq deadline(a_i)))}{|A|} \quad (۱۶)$$

- **تاخیر:**

همانگونه که معادله ۱۷ نشان می‌دهد تاخیر بر اساس دو معیار ۱) تاخیر بین درخواست‌دهنده تا محل استقرار برنامه کاربردی ($D_r$) و ۲) تاخیر بین سرویس‌های برنامه کاربردی ($D_s$) محاسبه می‌شود.

$$\vartheta = \text{avg}(D_r) + \text{avg}(D_s) \quad (۱۷)$$

مبنای محاسبه تاخیر در دو معیار $D_r$ و $D_s$، تاخیر بین دستگاه‌های موجود در کوتاهترین مسیر بین مبدا و مقصد است که طبق معادله ۱۸ محاسبه می‌شود.

$$\rho(f_i, f_j) = {e_{ij}}^{PD} + \frac{Packet_{size}}{{e_{ij}}^{BW}} \quad (۱۸)$$

- **میزان استفاده از منابع:**

برای محاسبه میزان استفاده از منابع، میزان استفاده جوامع از منابع مختلف را محاسبه کرده و سپس میانگین میزان استفاده جوامع مختلف از منابع را مطابق معادله ۱۹ محاسبه می‌کنیم. پارامترهای استفاده شده در معادله ۱۹ متوسط میزان حافظه، متوسط میزان فضای ذخیره‌سازی و متوسط میزان سرعت پردازشی آن جامعه را نشان می‌دهد.

$$\partial = \text{avg}\,(R_{RAM}, R_{TB}, R_{IPT}) \quad (۱۹)$$

معادله ۲۰، محاسبه پارمترهای مختلف ذکر شده به ازای منابع مختلف را نشان می‌دهد.

$$R_x = \frac{(\sum_{i=1}^{|F|} f_i{}^{x})}{|F|} \mid x = RAM, TB, IPT \quad (۲۰)$$

- **دسترس پذیری:**

این معیار مطابق معادله ۲۱ میزان در دسترس بودن برنامه کاربردی برای درخواست‌های متناظر را محاسبه می‌کند.

$$\tau = \frac{(\sum_{i=0}^{|p|} \sum_{j=0}^{|r|} r_j^{a_i})}{r} \quad (۲۱)$$

معادله ۲۲ نشان می‌دهد آیا همه سرویس‌های برنامه کاربردی برای درخواست‌دهنده دسترس پذیر بوده یا خیر. در صورتیکه همه سرویس‌ها در دسترس باشد در معادله ۲۱ امتیازی بابت دسترس‌پذیری دریافت می‌کند.

$$r_j^{a_i} = \begin{cases} 1\ if\ all\ services\ of\ a_i\ are\ accessible\ for\ r_j \\ 0\ if\ all\ services\ of\ a_i\ are\ not\ accessible\ for\ r_j \end{cases} \quad (۲۲)$$

## ۵-۳- نتایج تجربی

همانگونه که در بخش مقدمه اشاره شد، با توجه به محدودیت‌های گره‌های مه و نیاز برنامه‌های کاربردی اینترنت اشیاء لازم است که راهکارهای بهینه‌ای برای جایگذاری موثر برنامه‌های کاربردی اینترنت اشیاء در محیط مه ارائه گردد. در این راستا در این مقاله روش پیشنهادی PDAF با دو رویکرد استفاده بهینه از منابع مه و همچنین افزایش کیفیت سرویس برنامه‌های کاربردی اینترنت اشیاء ارائه گردید. به همین منظور روش PDAF با معیارها و سناریوهای ارزیابی متعددی در این بخش مورد ارزیابی قرار گرفته است. آزمایش‌های شبیه‌سازی متعدد مبتنی بر ویژگی‌های تصادفی انجام شده است. همچنین پارامترهای اجزاء موجود در دامنه مسئله با توزیع یکنواخت مدل‌سازی گردیده است. بازه مقادیر مورد استفاده برای ایجاد و مقداردهی تصادفی در جدول ۱ مشخص شده است. به عنوان نمونه توپولوژی شبکه بر اساس مدل باراباسی آلبرت و به صورت تصادفی ایجاد می‌گردد. تعداد گره‌های مه در نظر گرفته شده در هر آزمون ۱۰۰ گره است. ۵ درصد از گره‌ها با بیشترین مقدار مرکزیت بینابینی به عنوان دروازه‌های ارتباطی با گره‌های ابر و ۲۵ درصد از گره‌ها با کمترین مقدار مرکزیت بینابینی به عنوان دروازه‌های بین لایه اینترنت اشیاء و لایه مه در نظر گرفته می‌شود. تعداد و نوع برنامه‌های درخواستی از دستگاه‌های اینترنت اشیا متصل به دروازه‌ها با استفاده از توزیع یکنواخت تعیین شده است. برنامه‌های کاربردی متشکل از چندین سرویس نیز به صورت تصادفی ایجاد می‌شوند. در واقع تعداد سرویس‌های آنها و توپولوژی ارتباطی آنها نیز به صورت کاملا تصادفی ایجاد می‌شوند.

دو سناریو برای مقایسه روش‌ها استفاده شده است. در سناریوی اول

کارایی روش‌ها با افزایش بار کاری انجام می‌شود و در سناریوی دوم میزان خرابی گره‌ها تغییر داده می‌شود تا عملکرد روش‌های مورد مقایسه در شرایط خرابی و محدودیت منابع نیز ارزیابی شود. تمرکز در سناریوی دوم بر ارزیابی میزان دسترس‌پذیری برنامه‌های کاربردی در شرایط ایجاد خرابی است.

تعداد برنامه‌های کاربردی در سناریوهای مختلف بین ۱۰ تا ۸۰ برنامه کاربردی تغییر داده می‌شود تا تاثیر افزایش بار بر عملکرد روش پیشنهادی ارزیابی گردد. در آزمایش‌هایی که تاثیر خرابی گره‌ها بر کارایی روش پیشنهادی بررسی می‌گردد، نرخ خرابی گره‌ها بین ۱۰ تا ۸۰ درصد تغییر داده شده است.

به منظور بررسی نتایج روش پیشنهادی PDAF ، این روش با سایر روش‌ها مانند Partition به عنوان نماینده دسته روش‌های مبتنی بر جامعه و روش ILP به عنوان یکی از پرکاربردترین روش‌های بهینه‌سازی و نماینده روش‌های مبتنی بر سیاست جایگذاری انتخاب شده است[۳۳]-[۴۲]. همچنین در این ارزیابی دو نسخه روش شامل نسخه یک (PDAF.v1) و نسخه دو (PDAF.v2) با سایر روش‌ها مورد مقایسه قرار گرفته است.

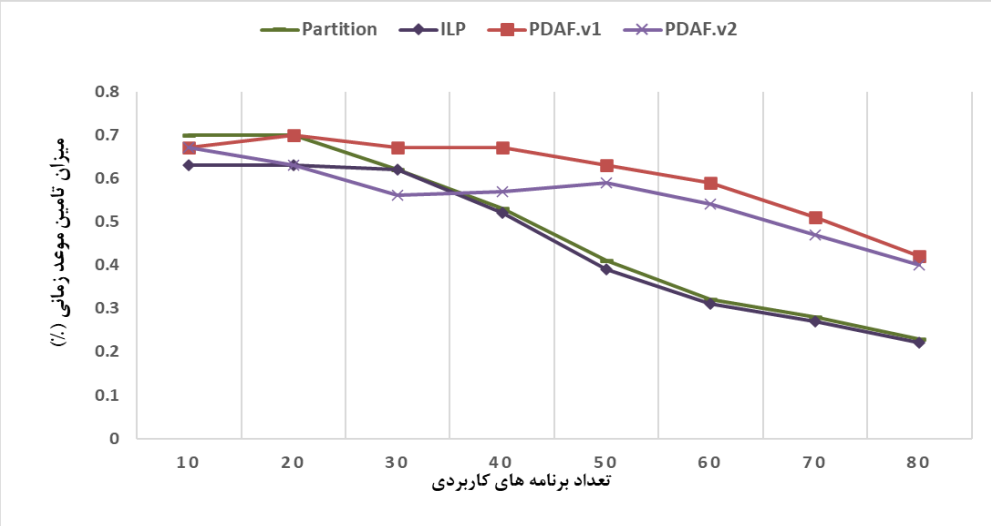
- **میزان تامین موعد زمانی:**

مطابق با شکل ۲، از بین دو نسخه روش پیشنهادی، نسخه یک (PDAF.v1) که در آن به برنامه‌های کاربردی با متوسط فاصله درخواست بیشتر اولویت می‌دهد، در معیار تامین موعد زمانی عملکرد بهتری داشته است. دلیل این امر را می‌توان به ماهیت توزیع شده درخواست‌ها در محیط نسب داد. بنابراین اگر درخواست‌های یک برنامه با متوسط فاصله بیشتر به تاخیر بیافتند، ممکن است به دلیل اشباع منابع، با محدودیت منابع بیشتری برای پاسخ به موقع به درخواست‌های خود مواجه شوند. از طرفی درخواست‌هایی که به نوعی محلیت دارند، با روش‌هایی نظیر استفاده از نسخه‌های موجود نزدیک می‌توانند در زمانی دیرتر پاسخ خود را به موقع دریافت می‌کنند. همین امر منجر به برتری نسخه یک روش پیشنهادی در مقایسه با نسخه دو همین روش شده است. ضمن اینکه با ترکیب معیار فاصله درخواست‌ها با موعد زمانی و بالا بودن وزن موعد زمانی در تعیین اولویت‌ها همچنین نسبت به سایر روش‌های متکی به تک معیار موعد زمانی نیز عملکرد بهتری داشته است. مطابق شکل ۳، این روش در معیار تاخیر در مجموع از دو روش دیگر بهتر عمل کرده است. ضمن اینکه نسخه ۱ روش پیشنهادی (PDAF.v1)، با اولویت دادن به درخواست‌های با متوسط فاصله بیشتر و موعدزمانی کوتاهتر توانسته است به طور متوسط به درخواست‌ها با تاخیر کمتری پاسخ دهد. این موضوع خصوصا با افزایش برنامه‌های کاربردی از ۳۰ و بیشتر به صورت بسیار محسوس‌تری قابل مشاهده است. به این ترتیب قبل از اینکه جوامع به اشباع برسند توانسته به درخواست‌ها در جامعه‌ای نزدیکتر به آنها پاسخ دهد.

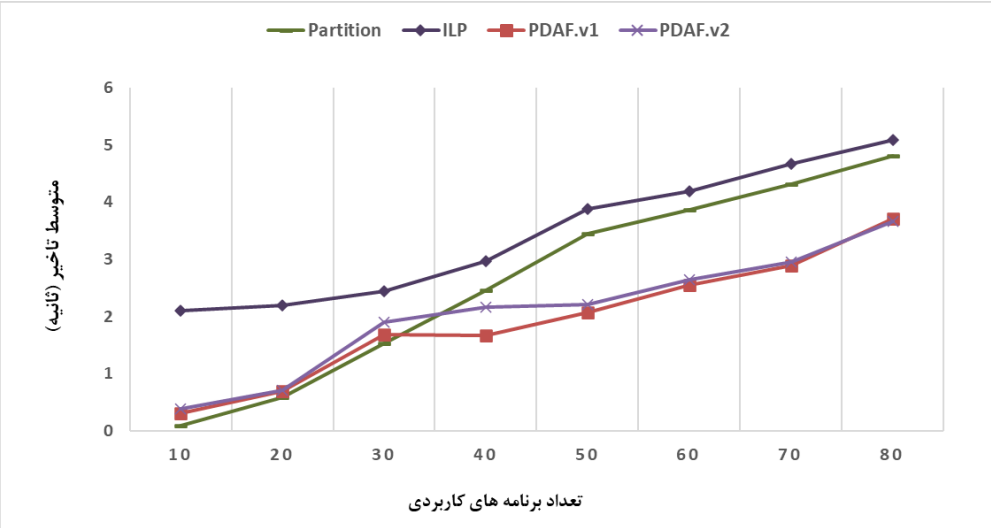
- **میزان استفاده از منابع:**

شکل ۴ روند استفاده از منابع شبکه با افزایش برنامه‌های کاربردی را نشان می‌دهد. مطابق این شکل، روش پیشنهادی PDAF استفاده موثرتری از منابع گره‌های مه داشته است.

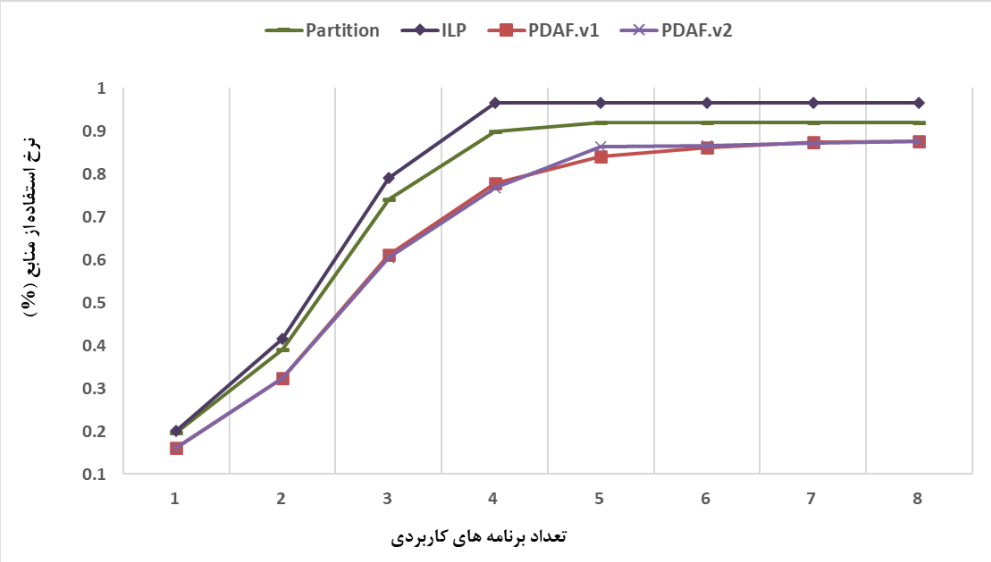
بعد از آن روش Partition توانسته میزان منابع مورد استفاده را کاهش دهد. همچنین روش ILP تقریبا از همه منابع موجود استفاده کرده است و بنابراین بهینه‌سازی به این منظور نداشته است.




**شکل ۲. تامین موعد زمانی با افزایش تعداد برنامه‌های کاربردی**




**شکل ۳. تاخیر پاسخگویی با افزایش تعداد برنامه‌های کاربردی**




**شکل ۴. استفاده از منابع با افزایش تعداد برنامه‌های کاربردی**

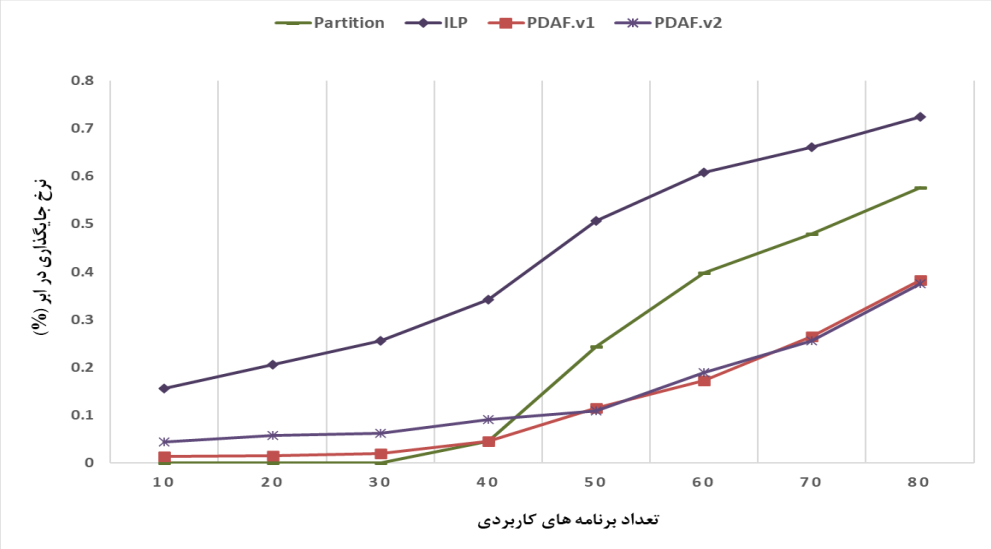
یکی از دلایل برتری روش‌های Partition و PDAF نسبت به ILP، توجه دو روش اول مبنی بر ایجاد و سازماندهی زیرساخت جایگذاری و استفاده از نسخه‌های موجود و استقرار داده شده از قبل برای پاسخ به درخواست‌ها تا حد امکان است. همچنین برتری روش پیشنهادی نسبت به روش Partition به دلیل ایجاد جوامع متوازن با کیفیت و استفاده از معیارهای ترکیبی برای اولویت‌بندی جایگذاری برنامه‌های کاربردی است.

- **میزان جایگذاری در ابر:**

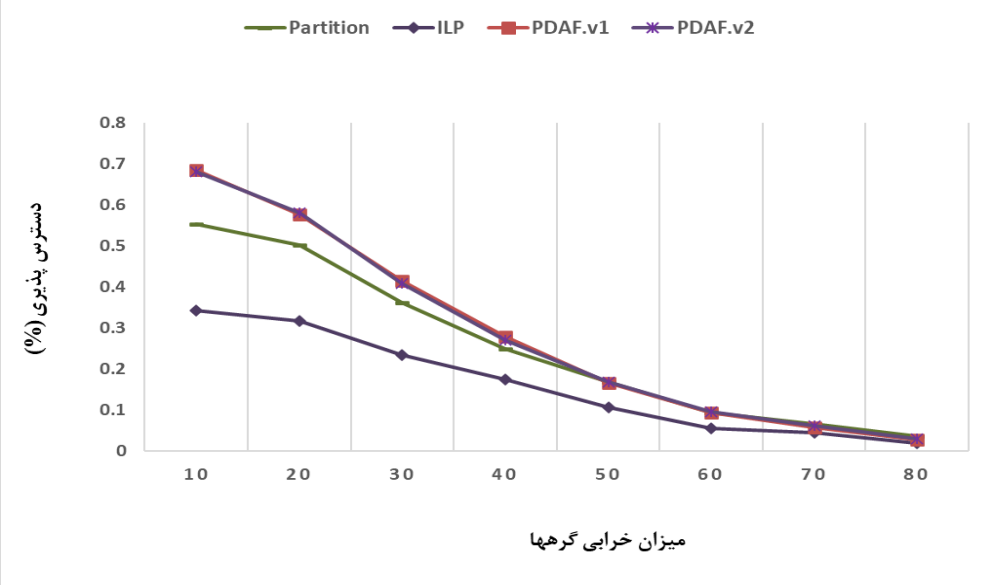
شکل ۵ نشان‌دهنده روند نرخ جایگذاری در ابر با افزایش تعداد برنامه‌های کاربردی است. مطابق این شکل روش پیشنهادی PDAF، توانسته خصوصا با افزایش تعداد برنامه‌های کاربردی تعداد جایگذاری‌های کمتری در محیط ابری داشته باشد. بنابراین موید این است که استفاده موثرتری از منابع موجود داشته و به درخواست‌های بیشتری در محیط مه پاسخ داده است. دلیل این امر را می‌توان به استفاده از جوامع متوازن و اولویت دهی به درخواست‌های متعدد پراکنده در محیط نسبت داد.

- **دسترس پذیری:**

شکل ۶ نرخ دسترس‌پذیری برنامه‌های کاربردی توسط درخواست‌دهنده‌ها را نشان می‌دهد. مطابق با این شکل، روش پیشنهادی بر روش‌های ILP و Partition برتری دارد.




**شکل ۵. نرخ جایگذاری در ابر با افزایش تعداد برنامه‌های کاربردی**




**شکل ۶. میزان دسترس پذیری با افزایش نرخ خرابی گره‌ها**

با افزایش خرابی برنامه‌های کاربردی به بیش از ۵۰ درصد به دلیل محدود شدن شدید منابع و عدم امکان برنامه ریزی توسط روش‌های مختلف، رفتار روش‌ها به یکدیگر نزدیک شده است. با توجه به اینکه روش ILP برای جایگذاری برنامه‌های کاربردی، توجهی به ساختار توزیع گره‌ها ندارد، در این معیار نیز بسیار ضعیف عمل کرده است.

روش Partition تا ۵۰ درصد خرابی گره‌ها عملکرد ضعیف‌تری از روش پیشنهادی داشته است اما بعد از آن به دلیل خرابی زیاد گره‌ها رفتار دو روش بسیار نزدیک شده است.

با افزایش خرابی برنامه‌های کاربردی به بیش از ۵۰ درصد به دلیل محدود شدن شدید منابع و عدم امکان برنامه ریزی توسط روش‌های مختلف، رفتار روش‌ها به یکدیگر نزدیک شده است. با توجه به اینکه روش ILP برای جایگذاری برنامه‌های کاربردی، توجهی به ساختار توزیع گره‌ها ندارد، در این معیار نیز بسیار ضعیف عمل کرده است.

روش Partition تا ۵۰ درصد خرابی گره‌ها عملکرد ضعیف‌تری از روش پیشنهادی داشته است اما بعد از آن به دلیل خرابی زیاد گره‌ها رفتار دو روش بسیار نزدیک شده است.

## ۶- تحلیل نتایج

تمرکز این مقاله بر ارائه روشی بهینه برای جایگذاری چند معیاره برنامه‌های کاربردی مبتنی جوامع ایجاد شده در محیط مه بود. همانگونه که در بخش ۵-۲ اشاره شد، روش پیشنهادی دو هدف عمده بهبود میزان استفاده از منابع و همچنین افزایش کیفیت

برنامه‌های کاربردی را دنبال می‌کند. به همین منظور تنوعی از معیارهای ارزیابی مورد استفاده در مراجع معتبر[۱۴] و [۲۷] و [۴۱] و [۴۳]، برای سنجش عملکرد روش از این دوجنبه مورد استفاده قرار گرفته است. از دیگر نکات مهم در نحوه ارزیابی روش، بازه تغییرات گسترده بارکاری و نرخ خرابی گره‌ها در سناریوهای ارزیابی مختلف است. به نحوی که این بازه از ۱۰ تا ۸۰ برنامه کاربردی در نظر گرفته شده است در حالیکه روش Partition تنها به ارزیابی کارایی روش خود در مواجهه با ۲۰ برنامه کاربردی پرداخته است. همچنین بازه تغییرات نرخ خرابی از ۱۰ تا ۸۰ درصد گره‌ها تغییر داده شده تا عملکرد روش‌ها در مقایسه با یکدیگر به صورت دقیقتری بررسی شود.

نتایج ارائه شده در بخش نتایج تجربی (۳-۵) نیز موید آن است که روش پیشنهادی با افزایش بارکاری برتری خود را نسبت به سایر روش‌ها حفظ کرده است. همچنین با افزایش نرخ خرابی دچار افت کیفیت عملکرد نشده است.

نتایج ارائه شده در شکل‌های ۴ و ۵ نشان‌دهنده بهره‌برداری موثرتر روش PDAF از منابع موجود است. شکل ۴ نشان‌دهنده میزان استفاده از منابع است. طبق این شکل با زیاد شدن برنامه‌های کاربردی یا بار کاری، با توجه به استفاده بیشتر روش‌ها از منابع برای پاسخگویی، میزان استفاده از منابع به حداکثر مقدار خود می‌رسد. با این حال روش پیشنهادی PDAF با شیب ملایمتری به این نقطه حداکثری خود رسیده است و در نهایت نیز در شرایط بارکاری زیاد استفاده بهتری از منابع داشته است. نتایج نشان‌دهنده کاهش ۱۰ درصدی استفاده از منابع توسط روش پیشنهادی است.

شکل ۵ میزان جایگذاری در ابر را نشان می‌دهد. روش پیشنهادی به میزان قابل توجهی نرخ جایگذاری در ابر را کاهش داده است. خصوصا با افزایش بارکاری، مشابه نکته‌ای که در نمودار میزان استفاده از منابع مشاهده شد، با شیب ملایمی میزان جایگذاری در ابر افزایش می‌یابد در حالیکه روش Partition با افزایش برنامه‌های کاربردی به میزان ۴۰ برنامه و بیشتر، جایگذاری برنامه‌های کاربردی در ابر را با جهش قابل توجه و شیب تندی افزایش می‌دهد. به نحوی که نتایج به دست آمده نشان‌دهنده کاهش ۴۰ درصدی جایگذاری برنامه‌های کاربردی در ابر توسط روش PDAF است. این موضوع نشان‌دهنده مدیریت بهتر منابع و جایگذاری بیشتر برنامه‌های کاربردی در محیط مه توسط روش پیشنهادی است.

معیار مهم دیگری که نشان‌دهنده تاثیر روش بر افزایش کیفیت برنامه‌های کاربردی و رضایت کاربر است، معیار میزان تامین موعد زمانی است. مطابق با شکل ۲ روش پیشنهادی در معیار تامین موعد زمانی نسبت به روش‌های Partition و ILP موفق‌تر عمل کرده است و توانسته با اختلاف قابل توجهی موعد زمانی برنامه‌های کاربردی بیشتری را تامین کند. همچنین با افزایش بارکاری دچار افت محسوسی نشده است در حالیکه دو روش دیگر با افزایش بارکاری دچار افت قابل توجهی شده‌اند. نتایج بدست آمده نشان می‌دهد روش PDAF میزان تامین موعد زمانی برنامه‌های کاربردی را تا ۲۲ درصد افزایش داده است.

در مورد معیار تاخیر نیز روش پیشنهادی با افزایش تعداد برنامه‌های کاربردی در مقایسه با دو روش دیگر با شیب ملایمتری دچار افزایش تاخیر می‌شود(شکل ۳). نتایج نشان می‌دهد که روش پیشنهادی به طور متوسط تاخیر را تا ۲۶ درصد کاهش داده است..

عملکرد بهتر روش پیشنهادی نسبت به روش Partition [۲۷] و ILP [۳۳]-[۴۲] در معیارهای مختلف به چند دلیل است که در ادامه به آنها می‌پردازیم:

- ایجاد جوامع متوازن، متصل و با کیفیت و جایگذاری برنامه کاربردی در یک جامعه واحد
- استفاده از معیارهای ترکیبی برای اولویت‌بندی جایگذاری برنامه‌های کاربردی
- جایگذاری برنامه کاربردی در نزدیکترین جامعه نسبت به کاربر یا درخواست‌دهنده
- استفاده از نسخه موجود برنامه کاربردی جایگذاری شده تا حد امکان

## ۷- پیچیدگی زمانی روش پیشنهادی

با توجه به اینکه بیشترین پیچیدگی زمانی و هزینه روش پیشنهادی به الگوریتم ژنتیک برمی‌گردد، پیچیدگی زمانی این الگوریتم در ادامه محاسبه شده است.

شرط توقف الگوریتم ژنتیک، رسیدن به یک امتیاز کمی یا تکرار الگوریتم به تعداد مشخصی است که حداکثر به تعداد MAX_ITER است. بنابراین در فرمول ۲۳، به عنوان اولین فاکتور در محاسبه پیچیدگی زمانی در نظر گرفته می‌شود. همچنین در هر تکرار الگوریتم ژنتیک جمعیتی به اندازه Pop_Size مقداردهی اولیه شده و مورد پردازش قرار می‌گیرد. بنابراین اندازه جمعیت به عنوان فاکتور دوم پیچیدگی زمانی در نظر گرفته می‌شود، در هر جمعیت کروموزوم‌ها مقداردهی اولیه می‌شوند. از طرفی طول هر کروموزوم به اندازه تعداد گره‌های مه است بنابراین به عنوان فاکتور دیگری در قالب |F| در پیچیدگی نشان داده شده است. در مرحله بعد جوامع موجود در هر کروموزوم مقداردهی شده اولیه، در قالب جوامع

شناسایی می‌شوند. بنابراین به تعداد جوامع و به طول کروموزوم پردازش مورد نیاز است که این دو فاکتور با مقادیر |C| و |F| در پیچیدگی الگوریتم تاثیر می‌گذارند.

$$O\ (MAX_ITER * Pop_Size * |C| * |F|^{2}) \quad (۲۳)$$

## ۸- جمع‌بندی ارزیابی

در این مقاله روشی تحت عنوان PDAF به منظور استقرار برنامه‌های کاربردی اینترنت اشیاء در محیط مه ارائه شد. روش پیشنهادی یک روش دو مرحله‌ای است. روش پیشنهادی در مرحله اول زیرساخت جایگذاری را مبتنی بر مفهوم جوامع و با هدف توزیع متوازن منابع ایجاد می‌کند. در مرحله دوم ابتدا برنامه‌های کاربردی را مبتنی بر روش چند معیاره با ترکیب معیارهای موعد زمانی و متوسط فاصله درخواست‌ها اولویت‌بندی نموده و سپس آنها را در نزدیکترین جامعه از جوامع شکل گرفته که دارای منابع کافی باشد جایگذاری می‌کند.

در فاز ایجاد زیرساخت جایگذاری ظرفیت شبکه را بر اساس سه معیار میزان منابع گره‌های مه، نحوه توزیع گره‌های مه و ویژگی‌های اتصالی و ارتباطی گره‌ها به صورت متوزان بین جوامع تقسیم می‌کند. در پژوهش‌های پیشین در حوزه تعریف جوامع تنها چگالی اتصال گره‌ها را برای ایجاد جوامع مدنظر قرار داده بودند که منجر به ایجاد جوامع نامتوازن و عدم امکان بهره‌برداری کامل از ظرفیت‌های شبکه می‌شد.

در مرحله جایگذاری سرویس نیز بر خلاف روش‌های قبلی که عمدتا برای اولویت‌بندی برنامه‌ها، متکی به موعد زمانی یا میزان تاخیر بودند پارامترهای بیشتری در نظر گرفته است. در واقع در این روش ترکیبی از پارامترهای موعد زمانی و متوسط فاصله درخواست‌های هر برنامه کاربردی برای اولویت‌بندی آنها در نظر گرفته شده است. چرا که متناظر با هر برنامه کاربردی درخواست‌های متعددی وجود دارد که در محیط پراکنده هستند. بنابراین در نظر گرفتن این دو معیار در کنار هم کمک می‌کند که برنامه ریزی لازم برای برنامه‌های کاربردی با نیازهای مشابه به صورت همزمان انجام شود. ضمن اینکه موعد زمانی نیز با اهمیت بالایی در تصمیم‌گیری اولویت‌بندی برنامه‌های کاربردی نقش بازی می‌کند.

نتایج نشان می‌دهد که روش پیشنهادی PDAF امکان مدیریت بهتر منابع و افزایش کیفیت سرویس برنامه‌های کاربردی اینترنت اشیاء را فراهم ساخته است. در این خصوص نتایج نشان می‌دهد که تامین موعدزمانی برنامه‌های کاربردی ۲۲ درصد افزایش یافته است. ضمن اینکه تاخیر ۲۶ درصد کاهش یافته است. همچنین شاهد کاهش ۴۰ درصدی جایگذاری در ابر هستیم. ضمن اینکه استفاده از منابع ۱۰ درصد کاهش داشته است. همچنین روش پیشنهادی منجر به این شده که دسترس‌پذیری هم ۱۲ درصد افزایش یابد.

با توجه به گستردگی بکارگیری خدمات مبتنی بر اینترنت اشیاء و نقش و اهمیت محاسبات مه در ارائه بهینه این خدمات، پرداختن به موضوع مدیریت و جایگذاری سرویس در این محیط از اهمیت زیادی برخوردار است. بنابراین این موضوع هم توسط صنایع و مدیران در تسهیل ارائه و افزایش کیفیت خدمات نوین و هم از طرف پژوهشگران به عنوان یک روند پژوهشی اثرگذار در حال و آینده باید مورد توجه قرار گیرد. ارائه راه‌حل بهینه مدیریت و جایگذاری سرویس با توجه به نیاز برنامه‌های کاربردی و محدودیت‌های محیط مه با چالش‌های مختلفی مواجه است. از طرفی استفاده و بکارگیری فناوری ابری برای برنامه‌های اینترنت اشیاء هزینه‌های عملیاتی، محاسباتی و امنیتی متعددی ایجاد می‌کند. بنابراین توصیه می‌شود ضمن حمایت جدی از پژوهش‌های انجام شده در این حوزه، برنامه‌ریزی و سیاست‌گذاری لازم برای استفاده موثر از تحقیقات انجام شده در این حوزه در صنعت نیز صورت پذیرد، تا بتوان پژوهش‌های انجام شده در این حوزه را به نحو موثری در صنایع و کاربردهای واقعی استفاده کرد.

یکی از موضوعات جذاب در ادامه این پژوهش، پرداختن به موضوع اشتراک منابع بین جوامع است. با توجه به اینکه برخی سرویس‌ها و منابع به صورت مشترک بین برنامه‌های کاربردی مورد استفاده قرار می‌گیرد، پرداختن به این موضوع کمک قابل توجهی در استفاده بهینه از منابع و همچنین سرعت پاسخگویی به درخواست‌ها خواهد داشت. موضوع مهم دیگری که باید به آن پرداخت موضوع پویایی گره‌های مه است که لزوم ایجاد پویایی بیشتر در ایجاد و بروزرسانی جوامع را ایجاب می‌کند. همچنین بهینه‌سازی روش پیشنهادی از نظر هزینه و زمان نیز موضوعات مهمی هستند که می‌توان در ادامه به آنها پرداخت.

## مراجع


[1] Das R, Inuwa MM., "A review on fog computing: Issues, characteristics, challenges, and potential applications", Telematics and Informatics Reports, 48, pp. 100049, 2023.

[2] Srirama SN. , "A decade of research in fog computing: Relevance, challenges, and future directions", Software: Practice and Experience,. vol. 54, no. 1, pp.3-23, 2024.

[3] Apat HK, Nayak R, Sahoo B. A, "comprehensive review on Internet of Things application placement in Fog computing environment", Internet of Things, pp.100866, 2023.

[4] Verma NK, Naik KJ., "Optimized fog community framework with advanced genetic algorithm for enhanced performance dynamics", The Journal of Supercomputing. vol. 80, no. 6, pp.8202-35, 2024.

[5] Ayoubi, M., Ramezanpour, M., and Khorsand, R., "An autonomous IoT service placement methodology in fog computing.", Software: Practice and Experience, vol. 51, no. 5, pp.1097-1120, 2021.

[6] Shooshtarian, L., Lan, D., and Taherkordi, A. "A clustering-based approach to efficient resource allocation in fog computing", In International Symposium on Pervasive Systems, Algorithms and Networks, pp. 207-224. Springer, Cham, 2019.

[7] Schaub, M.T., Delvenne, J.C., Rosvall, M. and Lambiotte, R., "The many facets of community detection in complex networks", Applied network science, vol. 2, no. 1, pp.1-13, 2017.

[8] Ahuja, M., R. Kaur, and D. Kumar, "Trend towards the use of complex networks in cloud computing environment", Int J Hybrid Inf Technol, vol. 8, no. 3, p. 297-306, 2015.

[9] Cazabet, R. and G. Rossetti, "Challenges in community discovery on temporal networks", in Temporal Network Theory, Springer. p. 181-197, 2019.

[10] Lei, Y. and S.Y. Philip, "Cloud service community detection for real-world service networks based on parallel graph computing", IEEE Access, p. 131355-131362, 2019.

[11] Chandusha, K., Chintalapudi, S.R. and Krishna Prasad, M.H.M., "An empirical study on community detection algorithms", In Smart Intelligent Computing and Applications, Springer, Singapore, 2019, pp. 35-44.

[12] Wang, W., Liu, D., Liu, X. and Pan, L., "Fuzzy overlapping community detection based on local random walk and multidimensional scaling", Physica A: Statistical Mechanics and its Applications, vol. 392, no. 24, pp.6578-6586,2013.

[13] Xie, J., Kelley, S. and Szymanski, B.K., "Overlapping community detection in networks: The state-of-the-art and comparative study", Acm computing surveys (csur), vol. 45, no. 4, pp.1-35, 2013.

[14] Skarlat, O., S. Schulte, M. Borkowski and P. Leitner, "Resource provisioning for IoT services in the fog", in 2016 IEEE 9th international conference on service-oriented computing and applications (SOCA), 2016. IEEE.

[15] Elkhatib, Y., et al., "On using micro-clouds to deliver the fog", IEEE Internet Computing, vol. 21, no. 2, pp. 8-15, 2017.

[16] Skarlat, O., M. Nardelli, S. Schulte, M. Borkowski and P. Leitner, "Optimized IoT service placement in the fog", Service Oriented Computing and Applications, vol. 11, no. 4, pp. 427-443, 2017.

[17] Yousefpour, A., G. Ishigaki, R. Gour, and J. P. Jue, "On reducing IoT service delay via fog offloading", IEEE Internet of things Journal, vol. 5, no. 2, pp. 998-1010, 2018.

[18] Guerrero, C., I. Lera, and C. Juiz, "On the influence of fog colonies partitioning in fog application makespan", in 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), 2018, IEEE.

[19] Chunaev, P., "Community detection in node-attributed social networks: a survey", Computer Science Review, vol. 37, pp. 100286, 2020.

[20] Interdonato, R., et al., "Feature-rich networks: going beyond complex network topologies", Applied Network Science, vol. 4, no. 1, pp. 1-13, 2019.

[21] Abbasi, M., E.M. Pasand, and M.R. Khosravi, "Workload allocation in iot-fog-cloud architecture using a multi-objective genetic algorithm", Journal of Grid Computing, vol. 18, no. 1, pp. 1-14, 2020.

[22] Reddy, K., AK Luhach , B. Pradhan, JK Dash and DS Roy, "A genetic algorithm for energy efficient fog layer resource management in context-aware smart cities", Sustainable Cities and Society, vol. 63, pp. 102428, 2020.

[23] Natesha, B. and R.M.R. Guddeti, "Adopting elitism-based Genetic Algorithm for minimizing multi-objective problems of IoT service placement in fog computing environment", Journal of Network and Computer Applications, vol. 178, pp. 102972, 2021.

[24] Al-Tarawneh, M.A., "Bi-objective optimization of application placement in fog computing environments", Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 2, pp. 1-24, 2021.

[25] Velasquez, K., DP Abreu, L. Paquete, M. Curado, and E. Monteiro, "A rank-based mechanism for service placement in the fog", in 2020 IFIP Networking Conference (Networking). 2020, IEEE.

[26] Kimovski, D., et al., "Adaptive nature-inspired fog architecture", in IEEE 2nd International Conference on Fog and Edge Computing (ICFEC), 2018, IEEE.

[27] Lera, I., C. Guerrero, and C. Juiz, "Availability-aware service placement policy in fog computing based on graph partitions", IEEE Internet of Things Journal, vol. 6, no. 2, pp. 3641-3651, 2018.

[28] Lera, I., C. Guerrero, and C. Juiz, "Comparing centrality indices for network usage optimization of data placement policies in fog devices", in Third International Conference on Fog and Mobile Edge Computing (FMEC), 2018, IEEE.

[29] Filiposka, S., A. Mishev, and C. Juiz, "Community-based VM placement framework", The Journal of Supercomputing, vol. 71, no. 12, pp. 4504-4528, 2015.

[30] Skarlat, O., M. Nardelli, S. Schulte, and S. Dustdar. "Towards qos-aware fog service placement", in 1st international conference on Fog and Edge Computing (ICFEC), 2017, IEEE.

[31] Nayeri, Z.M., Ghafarian, T. and Javadi, B., "Application placement in Fog computing with AI approach: Taxonomy and a state of the art survey", Journal of Network and Computer Applications, vol. 185, pp.103078, 2021.

[32] Lera, I.a.C.G., YAFS, "Yet Another Fog Simulator", https://yafs.readthedocs.io/en/latest/.

[33] Velasquez, K., DP Abreu, M. Curado and E. Monteiro, "Service placement for latency reduction in the internet of things", Annals of Telecommunications, vol. 72, no. 1-2, pp. 105-115, 2017.

[34] Salaht, F., F. Desprez, A. Lebre, C. Prud'Homme, and M. Abderrahim, "Service placement in fog computing using constraint programming", in International Conference on Services Computing (SCC), 2019, IEEE.

[35] Baranwal, G. and D.P. Vidyarthi, "FONS: a fog orchestrator node selection model to improve application placement in fog computing", The Journal of Supercomputing, pp. 1-28, 2021.

[36] Arkian, H.R., A. Diyanat, and A. Pourkhalili, "MIST: Fog-based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications", Journal of Network and Computer Applications, vol. 82, pp. 152-165, 2017.

[37] Yang, L., J. Cao, G. Liang, and X. Han, "Cost aware service placement and load dispatching in mobile cloud systems", IEEE Transactions on Computers, vol. 65, no. 5, pp. 1440-1452, 2015.

[38] Vijouyeh, L. N., Sabaei, M., Santos, J., Wauters, T., Volckaert, B., & De Turck, F., "Efficient application deployment in fog-enabled infrastructures", In 16th International Conference on Network and Service Management (CNSM), 2020, pp. 1-9. IEEE.

[39] Sriraghavendra, M., Chawla, P., Wu, H., Gill, S.S. and Buyya, R., "DoSP: A Deadline-Aware Dynamic Service Placement Algorithm for Workflow-Oriented IoT Applications in Fog-Cloud Computing Environments", In Energy Conservation Solutions for Fog-Edge Computing Paradigms, , Springer, Singapore, 2022, p. 21-47.

[40] Baranwal, G. and D.P. Vidyarthi, "FONS: a fog orchestrator node selection model to improve application placement in fog computing", The Journal of Supercomputing, pp. 1-28, 2021.

[41] Gasmi, K., Dilek, S., Tosun, S. and Ozdemir, S., "A survey on computation offloading and service placement in fog computing-based IoT", the Journal of Supercomputing, vol. 78, no. 2, pp.1983-2014, 2022.

[42] Heng L, Yin G, Zhao X., "Energy aware cloud-edge service placement approaches in the Internet of Things communications", International Journal of Communication Systems, vol. 35, no. 1, pp. e4899, 2022.

[43] Smolka, S. and Mann, Z.Á., "Evaluation of fog application placement algorithms: A survey", Computing, pp.1-27, 2022.