

# بهینه‌سازی استقرار مطمئن و انرژی کارای کاربردهای اینترنت اشیا در زیرساخت ابر و مه با استفاده از الگوریتم جستجوی فاخته

یاسر رمضانپور فومشی و میرسعید حسینی شیروانی

(پردازش، حافظه، ذخیره‌سازی و ...) مجزا است. بارهای کاری<sup>۸</sup> روی هر یک از گره‌های شبکه پویا می‌باشد و هر کاربرد، نیازمندی‌های خاص خود را از قبیل منابع محاسباتی، حساسیت به تأخیر و حفظ حریم خصوصی دارد. بنابراین برای استقرار مؤلفه‌های<sup>۹</sup> کاربرد در گره‌های مه باید نیازمندی‌های کاربرد و امکانات نرم‌افزاری، سخت‌افزاری، پهنای باند و تأخیر بین گره‌های شبکه مه در نظر گرفته شود [۱].

استقرار مؤلفه‌های یک کاربرد روی حداقل گره‌ها در شبکه مه منجر به کاهش مصرف انرژی و استفاده بهینه از منابع رایانشی و همچنین کاهش تأخیر بین مؤلفه‌های کاربرد می‌شود، اما این طرح استقرار منجر به تقویت پدیده نقطه تکی شکست<sup>۱۰</sup> می‌گردد. هنگامی که یک گره مه حاوی کلیه مؤلفه‌های یک کاربرد دچار آسیب شود، عملکرد کاربرد مختل شده و از کار می‌افتد. لذا پدیده نقطه تکی شکست روی قابلیت اطمینان<sup>۱۱</sup> کاربرد مشتری تأثیر منفی می‌گذارد و در نتیجه باید راهکاری برای استقرار مناسب مؤلفه‌ها برای ارائه خدمات مطمئن اتخاذ نمود.

در استقرار یک کاربرد با تعداد مؤلفه‌های اندک در شبکه مه، حالت‌های متعددی وجود دارد. با افزایش تعداد مؤلفه‌ها و گره‌های مه با توجه به ماهیت ناهمگن آنها، کشف راه حل بهینه استقرار دشوارتر شده و استخراج آن برای کاربر انسانی غیر ممکن می‌شود. موضوع استقرار مؤلفه‌ها، یک مسأله ان پی سخت<sup>۱۲</sup> است که در آن راه حل دقیق در زمان چندجمله‌ای وجود ندارد [۲] و [۳]. در سال‌های اخیر، تحقیقاتی در رابطه با توزیع مؤلفه‌ها در زیرساخت ابر و مه انجام شده است. هونگ و همکاران، یک بستر<sup>۱۳</sup> رایانش مه یکپارچه برای استقرار پویای مؤلفه‌ها روی گره‌های مه پیشنهاد داده‌اند، به طوری که در این روش برای مقابله با نقطه تکی شکست، مؤلفه‌ها روی بیش از یک گره توزیع می‌شوند [۴]. یک رویکرد توزیع مؤلفه‌های کاربردهای اینترنت اشیا، با توجه به حساسیت به تأخیر و استفاده کارآمد از منابع شبکه توسط تانجا و همکاران ارائه شده است [۵]. روشی برای حل مسأله جایابی خدمات مه شامل نگاشت بهینه بین کاربردهای اینترنت اشیا و منابع محاسباتی با توجه به تأمین نیازمندی‌های کیفیت خدمات کاربرد توسط محققین ارائه گردیده است. در این روش، نیازمندی کاربرد به حسگرها مورد توجه قرار نگرفته است [۶]. بروگی و همکاران، یک مدل عمومی و قابل توسعه برای توصیف استقرار آگاه از

چکیده: استقرار کاربردهای اینترنت اشیا در زیرساخت مه به عنوان مکمل ابر به طور مؤثری باعث صرفه‌جویی در استفاده از منابع محاسباتی در زیرساخت ابر می‌شود. تلاش‌های تحقیقاتی اخیر در حال بررسی چگونگی بهره‌برداری بهتر از قابلیت‌های مه برای اجرا و پشتیبانی از کاربردهای اینترنت اشیا است. استقرار ناکارآمد مؤلفه‌های کاربردها در مه منجر به اتلاف منابع، پهنای باند و افزایش مصرف انرژی می‌شود. همچنین توزیع مؤلفه‌های یک کاربرد روی تعداد حداقل ممکن از گره‌های مه به منظور کاهش مصرف انرژی منجر به کاهش سطح قابلیت اطمینان خدمات می‌شود. در این مقاله یک الگوریتم فراابتکاری ترکیبی بر مبنای الگوریتم جستجوی فاخته برای استقرار ایستای مؤلفه‌های کاربرد روی زیرساخت مه با هدف مصالحه بین مصرف بهینه انرژی و کاهش اثر نقطه تکی شکست و تقویت قابلیت اطمینان کاربرد در برابر خرابی ارائه می‌شود. نتایج شبیه‌سازی نشان می‌دهد که روش ارائه‌شده در این مقاله، مصرف انرژی در شبکه مه را کاهش داده و نیازمندی‌های کیفیت خدمات کاربرد اینترنت اشیا را با قابلیت اطمینان بالا تأمین می‌کند.

کلیدواژه: استقرار مطمئن، انرژی کارا، رایانش مه و ابر، کاربردهای اینترنت اشیا، کاربردهای توزیع شده.

## ۱- مقدمه

رایانش مه<sup>۱</sup> با پشتیبانی از استفاده یکپارچه منابع لبه و ابر<sup>۲</sup> محل قرارگیری کاربردهای<sup>۳</sup> اینترنت اشیا<sup>۴</sup> (IoT) را در نزدیکی منبع داده تسهیل می‌کند. استقرار<sup>۵</sup> کاربردها در لبه، منجر به کاهش بار شبکه و تضمین تحویل به موقع خدمات می‌شود. استقرار، مدیریت و به روز رسانی کاربردها در چنین محیط لایه‌ای چالش‌های جدیدی ایجاد می‌کند. شبکه مه در مقیاس وسیع، شامل تعداد انبوهی گره ناهمگن<sup>۶</sup> با منابع رایانشی<sup>۷</sup>

این مقاله در تاریخ ۲۹ آبان ماه ۱۳۹۹ دریافت و در تاریخ ۱۳ مهر ماه ۱۴۰۰ بازنگری شد.

یاسر رمضانپور فومشی، دانشکده مهندسی کامپیوتر، واحد بابل، دانشگاه آزاد اسلامی، بابل، ایران، (email: yaser.ramzanpoor@gmail.com).

میرسعید حسینی شیروانی (نویسنده مسئول)، دانشکده مهندسی کامپیوتر، واحد ساری، دانشگاه آزاد اسلامی، ساری، ایران، (email: mirsacid\_hosseini@iausari.ac.ir)

8. Workload
9. Component
10. Single Point of Failure
11. Reliability
12. NP-Hard
13. Platform

1. Fog Computing
2. Cloud
3. Application
4. Internet of Things
5. Deployment
6. Heterogeneous
7. Computing Resource

الگوریتم ابتکاری برای استخراج طرح استقرار مؤلفه‌ها اجرا و طرح به دست آمده برای توزیع مؤلفه‌ها به بستر مه ارسال می‌شود. هدف اصلی، دستیابی حداکثری به تولید طرح‌های استقرار موفق برای کاربردهای کاربران می‌باشد [۴].

یک الگوریتم آگاه از شبکه<sup>۸</sup> برای استفاده بهینه از منابع توسط تانجا و همکاران پیشنهاد شده است. این الگوریتم، گره‌های مه را بر اساس ظرفیت و مؤلفه‌های کاربرد را بر اساس نیازهای موجود مرتب می‌کند، سپس در صورت تأمین شرطها، نگاشت مؤلفه‌های کاربرد به گره‌های مه انجام می‌شود. بدین ترتیب مؤلفه‌ها بر اساس انتظار منبع اولویت‌بندی می‌شوند. روش پیشنهادی نشان می‌دهد که استفاده ترکیبی از ابر و مه در مقایسه با استفاده تنها از ابر برای ارائه خدمات به مؤلفه‌ها، باعث کمینه‌سازی تأخیر انتهابه‌انتهای<sup>۹</sup> می‌شود [۵].

ونتیسینکو و همکاران، روشی را برای پشتیبانی از توسعه‌دهنده برای حل مسأله جایابی خدمات مه ارائه داده‌اند. در این روش، نگاشت بهینه بین کاربردهای اینترنت اشیا و منابع محاسباتی با توجه به نیازمندی‌های کیفیت خدمات کاربرد استخراج می‌شود. در روش پیشنهادی، بهترین پیکربندی استقرار در زمینه انرژی هوشمند ارائه می‌گردد. برای پاسخ‌گویی به نیازمندی‌های کیفیت خدمات کاربرد، مهلت‌های زمانی و توان عملیاتی بایستی حفظ شود [۶].

در دسترس بودن مدل‌های مناسب زیرساخت‌ها و کاربردهای مه، برای موفقیت استقرارهای خودکار آگاه از کیفیت خدمات بر روی زنجیره ابر تا اشیا بسیار مهم است. اکثر ابزارهای پیشرفته برای استقرار خودکار نرم‌افزار توزیع شده<sup>۱۰</sup>، در استخراج طرح‌های استقرار مطلوب از ویژگی‌های غیر عملیاتی پشتیبانی نمی‌کنند. بدین ترتیب یک مدل ساده و کلی برای پشتیبانی از استقرار آگاه از کیفیت خدمات کاربردهای چندمؤلفه‌ای اینترنت اشیا در زیرساخت‌های مه توسط بروگی و همکاران ارائه شد. این مدل، کیفیت سیستم‌های عملیاتی زیرساخت‌های موجود (تأخیر و پهنای باند)، تعامل بین مؤلفه‌های نرم‌افزار، اشیا و سیاست‌های تجاری را توصیف می‌کند. الگوریتم‌هایی نیز برای استقرار مطلوب مؤلفه‌های کاربرد در زیرساخت مه ارائه شده است [۷].

محمود و همکاران، مدیریت آگاه از تأخیر<sup>۱۱</sup> مؤلفه‌های کاربرد را در رایانش مه پیشنهاد داده‌اند. در روش پیشنهادی، تأخیر در دسترسی به خدمات، زمان تحویل خدمات و تأخیر در ارتباطات داخلی در نظر گرفته می‌شود. هدف از کار، تضمین مهلت تحویل خدمات و استفاده بهینه از منابع در مه می‌باشد. برای بهینه‌سازی تعداد گره‌های مه میزبان مؤلفه‌ها، از روش هدایت و تخصیص مجدد مؤلفه‌های کاربرد استفاده می‌شود. در این مقاله، استفاده از سازمان‌دهی غیر متمرکز برای غلبه بر محدودیت‌هایی مثل سربرار مدیریت کاربرد، نقطه تکی شکست، ارتباطات افزونه و تأخیر در تصمیم‌گیری برای جایابی و هدایت مؤلفه‌ها پیشنهاد می‌شود [۸].

یک چارچوب برای تولید توپولوژی استقرار بهینه کاربردهای ابری اینترنت اشیا، متناسب با زیرساخت‌های فیزیکی موجود، توسط وگلر و همکاران پیشنهاد گردیده است. برای انعطاف‌پذیری کاربردهایی که توپولوژی استقرار آنها با گذشت زمان تکامل می‌یابد، جدایی مؤلفه‌ها لازم

کیفیت خدمات<sup>۱</sup> کاربردهای اینترنت اشیا در شبکه مه پیشنهاد داده‌اند [۷]. [۷]. پژوهشگران برای جایابی و هدایت مؤلفه‌ها، سازمان‌دهی غیر متمرکز را پیشنهاد کرده‌اند که این روش بر چالش‌های نظارت متمرکز مانند سربرار مدیریت کاربرد، نقطه تکی شکست، ارتباطات افزونه و تأخیر در تصمیم‌گیری غلبه می‌کند [۸]. در روش پیشنهادی این مقاله، یک الگوریتم انرژی کارا<sup>۲</sup> برای استقرار مطمئن مؤلفه‌های کاربرد در شبکه مه و ابر ارائه می‌شود. این روش در سناریوهایی مانند سیستم هشدار سرعت، سیستم اطفای حریق، مراقبت از سالمندان و غیره کاربرد دارد. در این سناریوها، قابلیت اطمینان و اتکا به سیستم برای مشتری از اهمیت زیادی برخوردار است. از دید مهندسی نرم‌افزار، ایجاد پروژه دارای دو فاز طراحی و زمان اجراست. تمرکز مقاله حاضر بر روی فاز طراحی در راستای ایجاد طرح استقرار ایستای مؤلفه‌های کاربرد روی زیرساخت مه بر اساس پروفایل کیفیت خدمات درخواستی کاربران است. بنابراین سهم اصلی این مقاله به شرح زیر خلاصه می‌شود:

۱) برای مصرف بهینه انرژی، زیرشبکه‌های مش‌کامل<sup>۳</sup> از شبکه مه استخراج می‌شود. از بین آنها مناسب‌ترین زیرشبکه انتخاب گردیده و مؤلفه‌های کاربرد روی گره‌های زیرشبکه انتخابی با توجه به منابع گره‌ها و ترافیک ارتباطی بین مؤلفه‌ها توزیع می‌شود که برای این منظور دو الگوریتم ابتکاری<sup>۴</sup> ارائه می‌گردد.

۲) برای کاهش اثر نقطه تکی شکست در استقرار مؤلفه‌های یک کاربرد و تأمین قابلیت اطمینان و اتکای مشتری به کاربرد، حداقل تعداد گره‌های مه مورد نیاز برای استقرار مؤلفه‌های یک کاربرد به تعداد گره‌های زیرشبکه مش‌کامل انتخابی محدود می‌شود.

۳) یک الگوریتم استقرار مؤلفه‌ها بر اساس الگوریتم فراابتکاری ترکیبی<sup>۵</sup> ترکیبی<sup>۵</sup> با رویکرد بهینه‌سازی مصرف انرژی و تأمین قابلیت اطمینان کاربرد کاربر طراحی می‌شود.

روش پیشنهادی در سناریوهای مختلف ارزیابی شده و نتایج به دست آمده از شبیه‌سازی، نشان‌دهنده برتری قابل توجه روش پیشنهادی در برابر سایر رویکردها است.

در ادامه، مقاله به شرح زیر سازمان‌دهی شده است: بخش ۲ به مروری بر کارهای مرتبط اختصاص دارد. مدل‌ها و چارچوب‌های پیشنهادی سیستم در بخش ۳ قرار گرفته است. بیان مسئله در بخش ۴ تشریح و الگوریتم پیشنهادی در بخش ۵ ارائه می‌شود. کار پیشنهادی در بخش ۶ شبیه‌سازی شده و مورد ارزیابی قرار می‌گیرد. در آخر، بخش ۷ به نتیجه‌گیری و تحقیقات آینده اشاره دارد.

## ۲- مروری بر کارهای گذشته

در این بخش، کارهای انجام‌شده در سال‌های اخیر در زمینه استقرار کاربردها در زیرساخت مه و ابر مرور می‌شود.

هونگ و همکاران، یک بستر برای توزیع پویای مؤلفه‌ها در زیرشبکه مه پیشنهاد داده‌اند. در روش پیشنهادی، تقاضاها به یک سرویس‌دهنده ارسال و سپس در پایگاه داده ذخیره می‌شود. هر تقاضا به چندین مؤلفه تقسیم شده و با استفاده از داکر<sup>۶</sup> یا کانتینر<sup>۷</sup> بسته‌بندی می‌گردد. یک

1. QoS-Aware
2. Energy Efficient
3. Full Mesh
4. Heuristic Algorithm
5. Meta-Heuristic Algorithm
6. Docker

7. Container
8. Network-Aware
9. End to End Delay
10. Distributed
11. Latency-Aware

جدول ۱: خلاصه مروری بر کارهای گذشته.

مقاله‌ها	اهداف استقرار			
	توزیع شده	قابلیت اطمینان	آگاه از منابع	آگاه از ترافیک
[۴]	✓	x	✓	x
[۵]	✓	x	✓	x
[۶]	✓	x	✓	x
[۸]	✓	x	✓	x
[۹]	✓	x	✓	x
[۱۰]	✓	x	x	x
[۱۱]	✓	x	✓	x
[۱۲]	✓	✓	✓	✓
مقاله پیشنهادی	✓	✓	✓	✓



شکل ۱: معماری مبتنی بر مش.

مقاله، قابلیت اطمینان کاربر به کاربرد در نظر گرفته نمی‌شود. همچنین رویکرد ارائه‌شده برای توزیع مؤلفه‌های کاربرد در لبه شبکه قابل استفاده نیست [۱۲]. در جدول ۱ مقایسه تحقیقات پیشین برای استقرار کاربردهای اینترنت اشیا در زیرساخت مه و ابر ارائه گردیده است.

در مقالات بررسی‌شده برای توزیع مؤلفه‌ها، معیارهای نقطه تکی شکست و همچنین مصرف منابع شبکه مانند پهنای در نظر گرفته نمی‌شود. لازم به ذکر است که در نظر گرفتن پارامترهای مذکور، در فرایند توزیع مناسب مؤلفه‌ها تأثیر بسزایی دارد. جنبه منحصر به فرد کار ما در مقایسه با کارهای موجود، تقویت قابلیت اطمینان کاربرد کاربر از جنبه تحمل‌پذیری در برابر خطا و استقرار آگاه از ترافیک<sup>۵</sup> کاربردها روی گره‌های مه می‌باشد.

### ۳- مدل سیستم

در این بخش مدلی مبتنی بر مه به منظور استقرار مؤلفه‌های کاربرد ارائه می‌شود و همچنین یک چارچوب<sup>۶</sup> برای مدیریت مؤلفه‌ها پیشنهاد می‌گردد. در آخر، مسأله استقرار مؤلفه‌های کاربرد روی گره مه تشریح می‌شود. متغیرها و پارامترهای ورودی استفاده‌شده در مدل و مفاهیم آنها در جدول ۲ ارائه گردیده است.

#### ۳-۱ چارچوب سیستم

مدل سیستم هدف در شکل ۱ ارائه گردیده است و با توجه به شکل، از یک سازمان‌دهنده<sup>۷</sup> بر بالای مه استفاده می‌شود. یکی از مسؤلیت‌های سازمان‌دهنده، استخراج زیرشبکه‌های مش کامل از گره‌های مه می‌باشد. معماری ارتباطی گره‌ها در هر زیرشبکه، شبیه معماری شبکه مش بی‌سیم است [۱۳]. الگوی رایانشی در هر شبکه مش کامل با شبکه مش سنتی متفاوت است. از شبکه مش گره‌های مه (سوئیچ‌ها و سرویس‌دهنده‌های مه) برای عملیات توزیع‌شده در داخل شبکه استفاده می‌گردد. پس از استخراج مش‌های کامل، از بین آنها زیرشبکه مناسب انتخاب شده و سازمان‌دهنده با توجه به ویژگی مؤلفه‌های کاربرد در مورد استقرار آنها در

است. تغییرات در توپولوژی‌های استقرار کاربرد می‌تواند به خاطر نیاز استقرار کاربرد جدید، تغییر در زیرساخت‌های فیزیکی در لبه شبکه (به عنوان مثال، حذف/افزافه‌کردن حسگرها یا دروازه‌ها)، تغییرات محیطی مانند تغییر الگوهای درخواست مشتری یا تغییرات تکاملی در منطق تجارت کاربرد در طول چرخه عمر آن نیز باشد. در تولید توپولوژی استقرار، پارامترهایی نظیر زمان مورد نیاز برای استقرار، زمان و پهنای باند مورد نیاز برای اجرای کاربرد و بهره‌برداری از دستگاه‌های لبه<sup>۹</sup> ارزیابی شده است [۹].

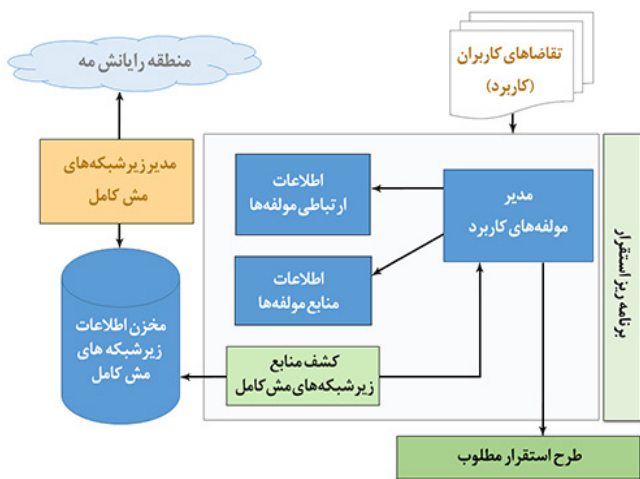
سوارز و همکاران، یک زیرساخت برنامه‌نویسی توزیع‌شده برای زنجیره محاسباتی گره‌های مه و ابر تحت عنوان فاگلت<sup>۲</sup> ارائه داده‌اند. فاگلت به طور خودکار منابع محاسباتی مه را در سلسله‌مراتب شبکه کشف می‌کند و مؤلفه‌های کاربرد را بر روی گره‌های مه متناسب با نیازمندی تأخیر هر مؤلفه مستقر می‌نماید. همچنین از چند کاربرد کنار هم روی هر گره مه پشتیبانی کرده و ای‌پی‌ای‌های (API) ارتباطی را برای مؤلفه‌های توزیع‌شده کاربرد در گره‌های مختلف در سلسله‌مراتب شبکه فراهم می‌کند. به کمک این ای‌پی‌ای‌ها، مؤلفه‌ها شرایط و وضعیت کاربرد را با هم تبادل می‌کنند. زیرساخت پیشنهادی به منظور انتزاع داده‌های وابسته به مکان و زمان و برای ذخیره و بازیابی داده‌های تولیدشده در گره‌های محلی نیز ای‌پی‌ای‌هایی را فراهم می‌کند. در این مقاله برای اجرا، مدیریت و مهاجرت<sup>۳</sup> مؤلفه‌ها بین گره‌های مه و نیازهای محاسباتی کاربرد الگوریتم‌هایی ارائه شده است [۱۰].

ارورا و همکاران یک الگوریتم برای جایابی مؤلفه‌های ناهمگن کاربردها در زیرساخت مه ارائه کرده‌اند. در این الگوریتم ابتدا مؤلفه‌ها بر اساس اندازه مرتب شده و سپس مؤلفه‌های کوچک‌تر در یک توپولوژی سلسله‌مراتبی در گره‌های میزبان جایابی می‌شوند. در فرایند جایابی، وابستگی بین مؤلفه‌ها، پارامتر قابلیت اطمینان کاربر به کاربرد، مصرف انرژی و تأخیر کل کاربرد در نظر گرفته نمی‌شود [۱۱].

یک رویکرد آگاه از ترافیک و انرژی مبتنی بر اولویت برای جایابی ماشین‌های مجازی کاربردهای اینترنت اشیا در زیرساخت ابری توسط اومر و همکاران ارائه شد. در این روش، ماشین‌های مجازی کاربردها در یک توپولوژی Fat Tree در گره‌های میزبان در ابر جایابی می‌شوند. در این

5. Traffic Aware  
6. Framework  
7. Organizer

1. Edge Device  
2. Foglet  
3. Application Programming Interface  
4. Migration



شکل ۲: چارچوب مدیریت مؤلفه‌های کاربرد.

شده است، قسمت برنامه‌ریزی شامل مدیر مؤلفه‌های کاربرد و همچنین ماژول‌هایی<sup>۲</sup> است که در مدیریت مؤلفه‌ها کمک می‌کنند. در کنار برنامه‌ریز استقرار، ماژول‌هایی برای ذخیره و بازیابی اطلاعات شبکه و سایر منابع زیرشبکه‌های مش کامل مه قرار دارد. اطلاعات گردآوری شده، توسط ماژول برنامه‌ریز استقرار برای مدیریت مؤلفه‌های کاربرد و ارائه یک طرح استقرار مطلوب استفاده می‌شود. در ادامه جزئیات مربوط به این مؤلفه‌ها در چارچوب پیشنهادی تشریح می‌گردند:

۱) مدیر مؤلفه‌های کاربرد: ماژول اصلی است که از سایر ماژول‌های موجود در چارچوب برای تصمیم‌گیری در مورد نحوه استقرار مؤلفه‌های کاربرد در گره‌های مه یا ابر استفاده می‌کند. در یک کاربرد چندمؤلفه‌ای، به خاطر وابستگی مؤلفه‌ها به هم، تصمیم‌گیری برای استقرار بر اساس عوامل متعددی مانند قابلیت دسترسی به منابع، ساختار شبکه، نیازمندی‌های کیفیت خدمات برای کاربرد، توزیع بار و غیره صورت می‌گیرد. استقرار مؤلفه‌ها می‌تواند بر اساس اهدافی مثل کاهش مصرف انرژی، کمینه‌سازی ارتباطات شبکه و همچنین کاهش تأخیر کلی در کاربرد انجام شود.

۲) اطلاعات منابع مؤلفه‌ها: نیازهای پردازشی و حافظه مورد نیاز مؤلفه‌های کاربرد را از تقاضای ارسالی توسط کاربران استخراج می‌کند و سپس برای اتخاذ تصمیم‌های استقرار در اختیار مدیر مؤلفه‌های کاربرد قرار می‌دهد.

۳) اطلاعات ارتباطی مؤلفه‌ها: ارتباطات سهم عمده‌ای در مصرف منابع گره‌های مه مورد استفاده در اینترنت اشیا دارند. مدیریت مؤلفه‌های کاربرد در گره‌های مه شامل بهینه‌سازی توأم منابع محاسباتی، حافظه و ارتباطات می‌باشد. این ماژول، اطلاعات ارتباطی مؤلفه‌های کاربرد را از تقاضای ارسالی توسط کاربران استخراج می‌کند و در اختیار مدیر مؤلفه‌های کاربرد قرار می‌دهد.

۴) کشف منابع زیرشبکه‌های مش کامل: بر مبنای اطلاعات دریافتی از مدیر مؤلفه‌های کاربرد، مخزن اطلاعات زیرشبکه‌های مش کامل را پایش می‌کند و اطلاعات مش‌های کامل مطلوب برای استقرار مؤلفه‌ها را به مدیر مؤلفه‌های کاربرد ارسال می‌کند.

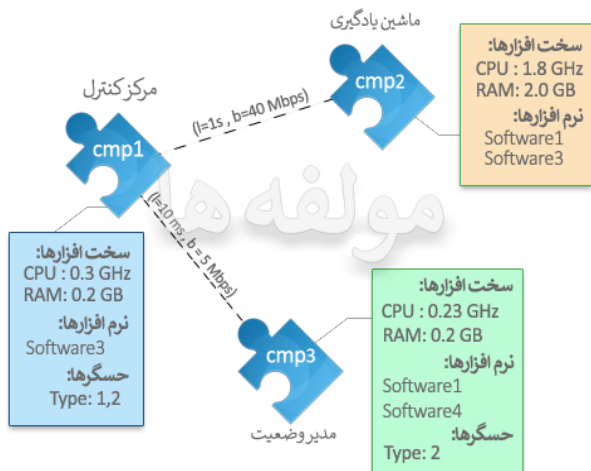
۵) مدیر زیرشبکه‌های مش کامل: با توجه به اطلاعات دریافتی از مه، زیرشبکه‌های مش کامل از گره‌های مه را استخراج کرده و اطلاعات این زیرشبکه‌ها را در مخزن اطلاعات ذخیره می‌کند. همچنین با

جدول ۲: علائم مورد استفاده در مدل سیستم.

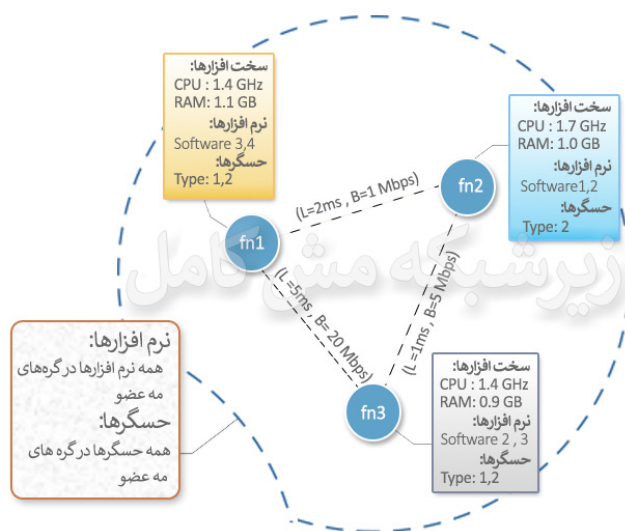
علامت	تعریف
$F$	شبکه مه
$f_{mesh}$	یک زیرشبکه با مش کامل از گره‌های مه
$N$	تعداد گره‌های مه در یک $f_{mesh}$
$fn_i$	گره مه $i = 1, 2, \dots, N$
$Id$	شناسه گره مه
$H$	مشخصه ساخت‌افزایی گره مه
$S$	مشخصه نرم‌افزاری گره مه
$HR_{fn}$	ظرفیت محاسباتی، حافظه و ذخیره‌سازی گره مه
$SR_{f_{mesh}}$	ظرفیت نرم‌افزاری $f_{mesh}$
$S_{f_{mesh}}$	ظرفیت حسگر $f_{mesh}$
$fn_{f_{mesh}}$	گره عضو $f_{mesh}$
$sensorlist$	حسگرهای گره مه
$B$	پهنای باند پیوند
$L$	تأخیر پیوند
$B_{mn}$	پهنای باند بین گره $n$ و $m$
$L_{mn}$	تأخیر پیوند ارتباطی بین گره $n$ و $m$
$d_{mn}$	فاصله گره مه $n$ و $m$
$UApp$	کاربرد کاربر
$n_h$	حداقل گره‌های مورد نیاز برای میزبانی مؤلفه‌های یک کاربرد
$M_i$	تعداد مؤلفه‌های کاربرد $i$
$cmplist_i$	لیست مؤلفه‌های کاربرد $i$
$cmp_k$	مشخصات مؤلفه $k$ از کاربرد $i$
$h$	سخت‌افزار مورد نیاز مؤلفه $k$
$s$	نرم‌افزار مورد نیاز مؤلفه $k$
$m_i$	مؤلفه‌های کاربرد $i$ ، مستقر در گره $L$
$b_{ij}$	پهنای باند مطلوب بین مؤلفه $i$ و $j$
$l_{ij}$	تأخیر مطلوب بین مؤلفه $i$ و $j$
$hr_{cmp}$	منابع سخت‌افزاری درخواستی مؤلفه‌های کاربرد
$sr_{cmp}$	منابع نرم‌افزاری درخواستی مؤلفه‌های کاربرد
$s_{cmp}$	منابع حسگر درخواستی مؤلفه‌های کاربرد
$t_{ij}$	ترافیک بین مؤلفه $i$ و $j$
$x_{cmp,fn}$	اگر مؤلفه روی گره مه باشد برابر با ۱ و در غیر این صورت ۰ است.
$y_{fn}$	اگر گره مه فعال باشد برابر با ۱ و در غیر این صورت ۰ است.

مش کامل انتخابی تصمیم‌گیری می‌کند. این سازمان‌دهنده به صورت مفهومی متمرکز بوده و می‌توان به منظور تحمل‌پذیری در برابر خطا آن را به صورت توزیع‌شده پیاده‌سازی کرد تا از نقطه تکی شکست جلوگیری شود. اولویت اصلی در چارچوب پیشنهادی، استخراج طرح استقرار با توجه به مش‌های کامل انتخابی و توزیع مؤلفه‌ها بر مبنای طرح می‌باشد. در توزیع مؤلفه‌ها، تنها آنهایی برای استقرار به زیرساخت ابر منتقل می‌شوند که حساس به تأخیر نبوده و به صورت دوره‌ای برای پردازش اطلاعات با آنها ارتباط برقرار می‌شود.

به منظور مدیریت و استقرار مناسب مؤلفه‌های کاربرد در گره‌های مه با توجه به کارایی سیستم، از یک چارچوب برنامه‌ریز استقرار<sup>۱</sup> در سازمان‌دهنده مه استفاده می‌شود. همان طور که در شکل ۲ نشان داده



شکل ۴: مشخصات مؤلفه‌های کاربرد.



شکل ۳: مشخصات یک زیرشبکه مش کامل و گره‌های عضو.

### ۳-۳ مدل کاربرد

در سال‌های اخیر با توجه به تغییر ماهیت خواسته‌های کاربران و انتظارات جدید از خدمات مبتنی بر اینترنت، نوع طراحی کاربردهایی که داده‌های کاربر را در مقیاس گسترده پردازش می‌کنند نیز متناسب با تقاضا تغییر کرده و برای تأمین نیازهای کاربران از ساختار چندمؤلفه‌ای برخوردار است [۱۴]. این مؤلفه‌ها به هم وابسته بوده و برای تأمین خواسته مشتری با هم همکاری می‌کنند. برای مثال یک کاربرد اینترنت اشیای ساده مراقبت از سالمندان را در نظر بگیرید که توسط یک شرکت ارائه خدمات سلامت هوشمند به مشتریان خود ارائه شده است. این کاربرد از سه مؤلفه مرکز کنترل (cmp1) برای تفسیر داده جمع‌آوری شده و کنترل دستی سیستم، مدیر وضعیت (cmp3) برای نظارت بر وضعیت افراد سالخورده و ناتوان و اطلاع‌رسانی فوری به نزدیک‌ترین مرکز درمانی در صورت بروز مشکلات جسمی و روحی و ماشین یادگیری (cmp2) برای ثبت تاریخچه اطلاعات افراد و تخمین وضعیت سلامتی آینده آنها تشکیل شده است. مؤلفه cmp2 حساس به تأخیر نبوده و می‌توان آن را در ابر یا مراکز داده در زیرساخت مه مستقر کرد.

منابع سخت‌افزاری و قابلیت‌های نرم‌افزاری مورد نیاز هر مؤلفه در شکل ۴ آمده و ارتباط بین مؤلفه‌ها از طریق پیوندهایی به تصویر کشیده شده است. برای مدیریت به موقع وضعیت سالمندان، مؤلفه cmp3 باید به حسگرهای مورد نیاز (حسگرهای کنترل وضعیت جسمانی) و یک محرک فعال‌کننده مکانیزم‌های عملیات مقدماتی و اطلاع‌رسانی به مراکز درمانی، دسترسی پیدا کند و این باید در عرض ۱۰ میلی‌ثانیه از محل استقرار cmp3 تا محل نصب حسگر و محرک اتفاق بیفتد. توجه داشته باشید که انتظار می‌رود گره‌های مه یا ابر بتوانند از طریق ای‌پی‌ای‌های ارائه‌شده توسط لایه میان‌افزار مه، از راه دور به اشیای موجود در گره‌های همسایه دسترسی داشته باشند [۱۵].

مسئله‌ای که برای استقرار مؤلفه‌های کاربرد فوق بایستی حل شود این است که چگونه می‌توان این مؤلفه را به گونه‌ای مستقر کرد که منابع مورد نیاز آنها برآورده شود. حتی برای این مثال ساده، باید طرح‌های متعدد استقرار برای استخراج یک نگاشت مطلوب برای مؤلفه‌های این کاربرد، ارزیابی شود. زیرا بر اساس منابع موجود، بیش از یک مؤلفه می‌تواند در یک گره مستقر شود و تعیین استقرارهای مطلوب با رشد زیرساخت‌ها و تعداد مؤلفه‌های کاربرد، از حد توان کاربر انسانی خارج شده و لازم است از الگوریتم‌های جستجوی هوشمند استفاده گردد.

نظارت دوره‌ای بر زیرساخت مه، وضعیت مش‌های کامل موجود در مخزن را اعتبارسنجی می‌نماید.

### ۳-۲ مدل مه

در این مقاله فرض می‌شود که شبکه‌ای متشکل از  $N$  گره مه با توان پردازشی و انرژی ناممکن داریم که قابلیت ذخیره‌سازی و اجرای مؤلفه‌های کاربرد را دارند. گره‌های مه در شبکه مفروض، عضو یک یا چند زیرشبکه مش کامل می‌باشند. هر یک از گره‌های مه به صورت مستقیم یا غیر مستقیم، از طریق اتصالات بی‌سیم یا باسیم به انواعی از گره‌های حسگر دسترسی دارند. یک گره مه  $fn \in F$  با استفاده از یک بردار  $\{id, mid, H, S, sensorlist\}$  نشان داده می‌شود که در این بردار  $id$  شناسه گره در شبکه مه،  $mid$  شناسه در مش کامل،  $H$  سخت‌افزار،  $S$  نرم‌افزار و  $sensorlist$  لیست حسگرهای در دسترس گره مه می‌باشند. مؤلفه‌هایی که در گره‌های یک مش کامل توزیع می‌شوند، به قابلیت‌های نرم‌افزاری و حسگرهای سایر گره‌های مه در همان مش کامل دسترسی دارند. لینک ارتباطی بین گره‌های مه را می‌توان با استفاده از بردار  $\{L, B\}$  نشان داد. در این بردار،  $L$  میزان تأخیر و  $B$  پهنای باند پیوند می‌باشد. شکل ۳ جزئیات یک مش کامل را نشان می‌دهد.

مدل شبکه ارتباطی: شبکه ارتباطی در یک زیرشبکه مش کامل با استفاده از یک گراف  $G = (FN, D)$  مدل می‌شود. در این گراف  $FN = \{fn_1, fn_2, \dots, fn_N\}$  یک مجموعه از گره‌های مه و  $D = d_{ij}$  فاصله بین  $fn_i$  و  $fn_j$  می‌باشد. در هر زیرشبکه مش کامل اگر  $fn_i = fn_j$  باشد، آن گاه  $d_{ij} = 0$  است و در غیر این صورت  $d_{ij} = 1$  خواهد بود. در (۱) شبکه ارتباطی و ماتریس فاصله گره‌های مه نشان داده شده است

$$\begin{matrix}
 & fn_1 & fn_2 & fn_3 & \dots & fn_n \\
 fn_1 & \begin{bmatrix} d_{11} & d_{12} & d_{13} & \dots & d_{1n} \end{bmatrix} \\
 fn_2 & \begin{bmatrix} d_{21} & d_{22} & d_{23} & \dots & d_{2n} \end{bmatrix} \\
 fn_3 & \begin{bmatrix} d_{31} & d_{32} & d_{33} & \dots & d_{3n} \end{bmatrix} \\
 \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
 fn_n & \begin{bmatrix} d_{n1} & d_{n2} & d_{n3} & \dots & d_{nn} \end{bmatrix}
 \end{matrix} \quad (1)$$

در عبارت فوق،  $b_{ij}$  و  $l_{ij}$  به ترتیب پهنای باند و تأخیر مطلوب بین مؤلفه‌های  $i$  و  $j$  و  $B_{mn}$  و  $L_{mn}$  به ترتیب پهنای باند و تأخیر پیوند ارتباطی بین گره‌های  $m$  و  $n$  است. یک متغیر دودویی  $x_{cmp,fn}$  برای تعیین وضعیت استقرار مؤلفه‌های کاربرد  $cmp \in UApp$  روی گره  $fn$  در  $F$  تعریف می‌کنیم، اگر مؤلفه  $cmp$  روی گره  $fn$  قرار گیرد،  $x_{cmp,fn}$  برابر با ۱ و در غیر این صورت ۰ خواهد شد. یک مؤلفه زمانی می‌تواند روی یک گره مه واقع شود که گره مه فعال باشد، یعنی  $y_{fn} = 1$  و در نتیجه داریم

$$x_{cmp,fn} \leq y_{fn}, \forall cmp \in UApp, fn \in F \quad (۴)$$

همچنین باید توجه داشت که کل منابع سخت‌افزاری مورد نیاز مؤلفه‌های مستقر در گره مه  $fn$  فراتر از ظرفیت آن گره نباشد و بنابراین داریم

$$\sum_{cmp \in UApp} x_{cmp,fn} \cdot hr_{cmp} \leq HR_{fn}, \forall fn \in F \quad (۵)$$

در عبارت فوق  $HR_{fn}$  برابر با ظرفیت محاسباتی، حافظه و ذخیره‌سازی گره مه و  $sr_{cmp}$  برابر با منابع محاسباتی، حافظه و ذخیره‌سازی درخواستی مؤلفه‌های کاربرد می‌باشد. از آنجایی که فرض بر این است که منابع نرم‌افزاری در گره‌های مه یک زیرشبکه مش‌کامل، در دسترس همه گره‌های زیرمجموعه آن قرار دارد، داریم

$$\sum_{cmp \in UApp} x_{cmp,fn} \cdot sr_{cmp} \leq SR_{fmesh}, \forall fmesh \in F \quad (۶)$$

در عبارت فوق  $SR_{fmesh}$  برابر با ظرفیت منابع نرم‌افزاری زیرشبکه مش‌کامل و  $sr_{cmp}$  برابر با منابع نرم‌افزاری درخواستی مؤلفه‌های کاربرد می‌باشد. همچنین داریم

$$\sum_{cmp \in UApp} x_{cmp,fn} \cdot s_{cmp} \leq S_{fmesh}, \forall fmesh \in F \quad (۷)$$

در عبارت فوق  $S_{fmesh}$  برابر با حسگرهای در دسترس از طریق زیرشبکه مش‌کامل انتخابی و  $s_{cmp}$  برابر با منابع حسگر درخواستی مؤلفه‌های کاربرد می‌باشد. هر یک از مؤلفه‌ها فقط روی یکی از گره‌های مه اجرا می‌شوند، لذا داریم

$$\sum_{fn \in F} x_{cmp,fn} = 1, \forall cmp \in UApp \quad (۸)$$

#### ۴- بیان مسأله

به طور کلی، چندین عامل بر روی کل مصرف انرژی سیستم مورد مطالعه تأثیر می‌گذارد، مانند بارهای محاسباتی، فناوری ارتباطات، میزان ترافیک تبادل‌شده، فاصله بین هر جفت گره مه و غیره. برای محاسبه مصرف انرژی هر گره مه، مصرف انرژی مربوط به اجرای هر یک از مؤلفه‌های روی گره مه و توان مصرفی برای تبادل اطلاعات بین گره‌های مه در نظر گرفته شده است. مصرف انرژی هر گره مه به طور مستقیم به استفاده از منابع آن بستگی دارد و بنابراین میانگین استفاده از منابع نرمال‌شده از یک گره مه با استفاده از (۹) محاسبه می‌شود

$$U_{fn}^{res} = \frac{w_1 \sum_j \frac{r_{cmp_j}^{CPU}}{R_{fn_j}^{CPU}} + w_2 \sum_j \frac{r_{cmp_j}^{RAM}}{R_{fn_j}^{RAM}}}{2} \quad (۹)$$

در معادله فوق، ضرایب  $w_1$  و  $w_2$  دارای مقادیر حقیقی هستند، به طوری که  $0 \leq w_1 \leq 1$ ،  $0 \leq w_2 \leq 1$  و  $w_1 + w_2 = 1$  است. از این ضرایب برای تعیین اهمیت منابع ترکیب‌شده در کل مصرف انرژی استفاده می‌شود. از

در این مقاله فرض می‌شود که تعداد  $R$  کاربرد اینترنت اشیا داریم و هر کاربرد  $r \in R$  با استفاده از یک بردار  $\{M, cmlist\}$  نشان داده می‌شود. هر کاربرد چندین مؤلفه دارد که  $M$  و  $cmlist$  به ترتیب تعداد و لیست مؤلفه‌های کاربرد می‌باشد و هر مؤلفه با استفاده از یک بردار  $\{k, h, s, sensorlist\}$  نشان داده می‌شود (شکل ۴). در بردار هر مؤلفه،  $k$  شناسه مؤلفه،  $h$  سخت‌افزار مورد نیاز،  $s$  نرم‌افزار مورد نیاز و  $sensorlist$  لیست حسگرهای مورد نیاز است که توسط گره مه میزبان مؤلفه بایستی تأمین گردد.

کاربرد کاربر با استفاده از یک گراف  $G = (cmlist, T)$  مدل می‌شود که در این گراف،  $cmlist = \{cmp_1, cmp_2, \dots, cmp_M\}$  مؤلفه‌های کاربرد و  $T = t_{ij}$  ترافیک ارسالی بین  $cmp_i$  و  $cmp_j$  می‌باشد. در (۲) ماتریس ترافیک بین مؤلفه‌های کاربرد  $i$  نشان داده شده است

$$(۲) \quad \begin{matrix} & cmp_1 & cmp_2 & \dots & cmp_m \\ \begin{matrix} cmp_1 \\ cmp_2 \\ \vdots \\ cmp_m \end{matrix} & \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1m} \\ t_{21} & t_{22} & \dots & t_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1} & t_{m2} & \dots & t_{mm} \end{bmatrix} \end{matrix} \quad \text{نمایش درایه ۳ و ۲ ماتریس تغییر کرد}$$

#### ۳-۴ مدل قابلیت اطمینان

استقرار مؤلفه‌های یک کاربرد روی حداقل تعداد گره‌ها در مه، منجر به دستیابی حداکثری به اهدافی مانند کاهش مصرف انرژی و استفاده بهینه از منابع رایانشی می‌شود، اما یکی از چالش‌های پیش رو تقویت پدیده نقطه تکی شکست برای کاربردهای کاربر می‌باشد. لذا برای این که هم اهداف مالکان رایانش مه در دستیابی به اهداف بهینه تأمین شود و همچنین درجه آسیب‌پذیری کاربردها در توزیع متمرکز در زیرساخت مه نیز کمتر گردد، یک حد آستانه برای تعداد گره‌های مه برای توزیع مؤلفه‌های کاربرد در نظر گرفته می‌شود. بدین منظور، حداقل تعداد گره‌ها برای توزیع مؤلفه‌ها برابر با تعداد گره‌ها در زیرشبکه مش‌کامل انتخابی است. در صورت عدم تحقق با کمترین فاصله نسبت به تعداد گره‌های زیرشبکه مش‌کامل، فرایند توزیع مؤلفه‌ها انجام می‌شود.

#### ۳-۵ مدل استقرار

برای استقرار مؤلفه‌ها از مش‌های کامل استخراج‌شده، متناسب با نیازهای کاربرد یک مش‌کامل انتخاب می‌شود. گره‌های مه در یک مش‌کامل، منابع مورد نیاز مؤلفه‌ها (تأخیر، پهنای باند و حسگرها) را تأمین می‌کنند. فرض ما در مدل پیشنهادی بر این است که حسگرها یا نرم‌افزارهای مورد نیاز مؤلفه‌های کاربرد به صورت اشتراکی توسط گره‌های مه در یک زیرشبکه مش‌کامل در دسترس خواهند بود.

در فرایند توزیع مؤلفه‌های کاربرد  $i$  روی گره‌های مه بایستی محاسباتی، فاصله گره‌های مه و پارامترهای کیفیت خدمات مورد نیاز مؤلفه‌های کاربرد در نظر گرفته شود. به منظور کاهش بار ترافیک بایستی ماتریس فاصله بین گره‌های مه در گراف شبکه و همچنین ماتریس ترافیک بین مؤلفه‌های یک کاربرد را نیز محاسبه کرد. باید توجه داشت که پیوند ارتباطی بین گره‌های مه  $m$  و  $n$  به لحاظ تأخیر و پهنای باند، ظرفیت ثابتی دارد و بنابراین نرخ ترافیک مؤلفه‌های کاربردها با ظرفیت پیوند بین گره‌های مه محدود می‌شود، لذا داریم

$$\sum_{cmp_i \in f_{fn_m}} \sum_{cmp_j \in f_{fn_n}} b_{ij} \times l_{ij} < B_{mn} \times L_{mn} \quad (۳)$$

محدودیت‌ها

$$n_h > \sum fn_{fmesh} \quad (14)$$

$$\sum_{cmp_i \in fn_m} \sum_{cmp_j \in fn_n} b_{ij} \times l_{ij} < B_{mn} \times L_{mn} \quad (15)$$

$$\sum_{cmp \in UApp} x_{cmp,fn} \cdot hr_{cmp} \leq HR_{fn}, \forall fn \in F \quad (16)$$

$$\sum_{cmp \in UApp} x_{cmp,fn} \cdot sr_{cmp} \leq SR_{fmesh}, \forall fmesh \in F \quad (17)$$

$$\sum_{cmp \in UApp} x_{cmp,fn} \cdot s_{cmp} \leq S_{fmesh}, \forall fmesh \in F \quad (18)$$

$$x_{cmp,fn} \leq y_{fn}, \forall cmp \in UApp, fn \in F \quad (19)$$

$$\sum_{fn \in F} x_{cmp,fn} = 1, \forall cmp \in UApp \quad (20)$$

$$x_{cmp,fn} \in \{0, 1\}, y_{fn} \in \{0, 1\} \quad (21)$$

### ۵- الگوریتم پیشنهادی

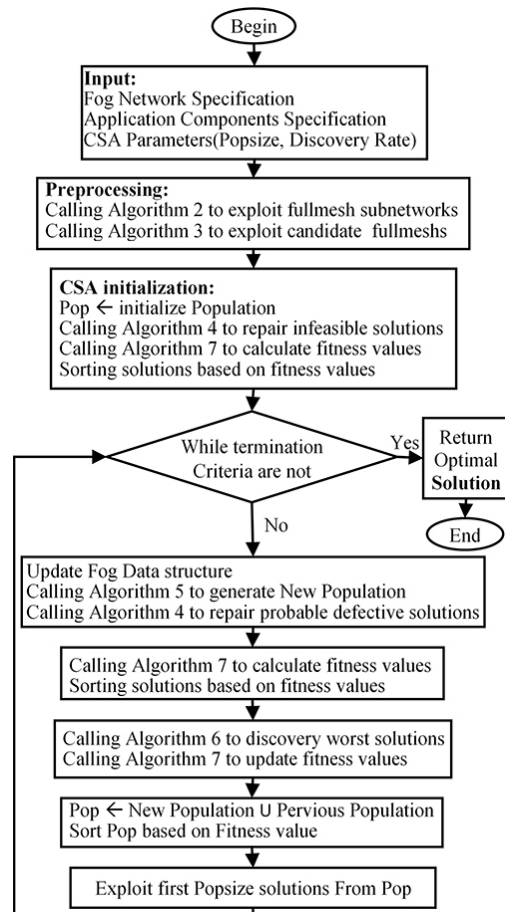
جستجوی فاخته<sup>۱</sup> یک الگوریتم هوش ازدحامی<sup>۲</sup> است که می‌توان در مسایل بهینه‌سازی دنیای واقعی از آن استفاده کرد. با استفاده از این الگوریتم می‌توان فضای جستجوی مسأله را کاهش داد و به راه حل بهینه تقریبی در زمان کوتاه‌تر دست یافت. در این مقاله، ما از جستجوی فاخته برای حل مسأله استقرار مؤلفه‌های کاربرد استفاده می‌کنیم. تأثیر و کارآمدی این الگوریتم در [۱۷] نشان داده شده است. در جستجوی فاخته برای بهینه‌سازی از سه قانون زیر استفاده می‌شود:

- هر فاخته هر بار یک تخم می‌گذارد و آن را در لانه‌ای که به طور تصادفی انتخاب شده است قرار می‌دهد.

- بهترین لانه‌ها با کیفیت بالای تخم‌ها (راه حل‌ها) به نسل‌های بعدی منتقل می‌شوند.

- تعداد لانه‌های میزبان موجود ثابت است و یک میزبان می‌تواند تخم بیگانه‌ای را با احتمال  $p_a \in [0, 1]$  کشف کند. در این حالت، پرند میزبان می‌تواند تخم را دور بیندازد یا لانه را رها کند تا لانه‌ای کاملاً جدید در مکان جدید ایجاد کند.

از نظر ریاضی، اولین قانون برای تولید یک راه حل جدید می‌تواند به طور تصادفی یا از طریق گام تصادفی<sup>۳</sup> یا پرواز لوی<sup>۴</sup> انجام شود. هم‌زمان، یک جایگزشت تصادفی محلی مانند تقاطع<sup>۵</sup> روی راه حل‌ها انجام می‌شود. قانون دوم مبتنی بر نخبه‌گرایی<sup>۶</sup> است تا بهترین راه حل‌ها به نسل بعدی منتقل شود و انتخاب اینچنین بهترین‌ها به اطمینان از همگرایی مناسب الگوریتم کمک می‌کند. قاعده سوم را می‌توان جهش<sup>۷</sup> (عملگر برداری از طریق ترکیب پرواز لوی و کیفیت تفاضل راه حل‌ها) در نظر گرفت تا بدترین راه حل‌ها با یک احتمال حذف و راه حل‌های جدیدی با توجه به شباهت راه حل‌ها با راه حل دیگر تولید شود. شکل ۵ بلاک دیاگرام الگوریتم پیشنهادی را نشان می‌دهد. این الگوریتم، داده‌های مسئله استقرار



شکل ۵: بلاک دیاگرام الگوریتم پیشنهادی.

آنجایی که بخش عمده‌ای از مصرف انرژی مربوط به واحدهای پردازشی است، در این مقاله،  $w_1 = 0.9$  و  $w_2 = 0.1$  در نظر گرفته شده است [۱۶]. بنابراین مصرف انرژی مربوط به هر گره میزبان مؤلفه‌های مختلف با استفاده از (۱۰) محاسبه می‌شود

$$P_{fn}^{res} = y_{fn} \times (P_{max} - P_{min}) \times U_{fn}^{res} + P_{min} \quad (10)$$

در (۱۰)، از پارامترهای  $P_{max}$  و  $P_{min}$  برای تعیین حداقل و حداکثر مصرف انرژی هر گره پردازشی به ترتیب در کمترین و بالاترین شرایط بهره‌برداری و از متغیر دودویی  $y_{fn}$  برای اشاره به فعال یا غیر فعال بودن گره پردازش استفاده می‌شود. همچنین مصرف انرژی حاصل از انتقال داده از طریق پیوندهای ارتباطی با استفاده از (۱۱) انجام می‌گردد

$$P_{fn}^{tr} = \sum_{fn_i \neq fn_j} t_{cmp_i, cmp_j} \times P_{tr} \quad (11)$$

در عبارت فوق  $P_{tr}$  انرژی انتقال داده در گره مه و  $t_{cmp_i, cmp_j}$  ترافیک مؤلفه‌های وابسته می‌باشد. با توجه به رابطه فوق، انرژی انتقال زمانی در محاسبات استفاده می‌شود که مؤلفه‌های وابسته  $i$  و  $j$  در دو گره متفاوت مه مستقر باشند. در نهایت انرژی مصرفی کل در یک گره مه که حاصل جمع انرژی انتقال و انرژی مصرفی منابع محاسباتی است، از رابطه زیر به دست می‌آید

$$P_{fn} = P_{fn}^{res} + P_{fn}^{tr} \quad (12)$$

مدل پیشنهادی بهینه‌سازی برای کمینه‌سازی مصرف انرژی شبکه مه در استقرار مؤلفه‌ها به صورت زیر فرموله می‌شود

$$\min TPC = \min \sum_{fn \in F} P_{fn} \quad (13)$$

1. Cuckoo Search
2. Swarm Intelligence
3. Random Walk
4. Levy Flight
5. Crossover
6. Elitism
7. Mutation

**Algorithm 2.** Fullmesh Subnetworks Exploitation**Input Parameters:**

Fog Network Communication Matrix ( $i \times j$ ) As Fog\_Net,  
where  $i$  and  $j = 1, \dots, n$   
and  $n$  is number of fog nodes

**Output:**

Full\_Meshs is vertical array of fullmesh subnetworks

**Initialization:**

define Full\_Meshs = {};

define Fog\_Net = upper triangular (Fog\_Net);

```

1: While the termination criteria not met do
2:   If Full_Meshs is empty then
3:     k = 0;
4:     For Each Fognodei in Fog_Net do
5:       For Each Fognodej = Fognodei+1 in Fog_Net do
6:         If Fognodei connected to Fognodej in Fog_Net then
7:           k = k+1;
8:           Full_Meshs [row k] = [Fognodei, Fognodej];
9:         End If
10:      End For Each
11:    End For Each
12:  Else
13:    For Each Fognodei do
14:      For Each k = all row of Full_Meshs do
15:        Current_row = Full_Meshs [row k];
16:        If Fognodei Not in Current_row and Fognodei
17:          connected to all node in Current_row in Fog_Net
18:          then
19:            Full_Meshs [row k] = [Full_Meshs [row k],
20:              Fognodei];
21:          End If
22:        End For Each
23:      End For Each
24:    End If
25:    remove duplicate rows in Full_Meshs;
26:  End While
27: Return Full_Meshs

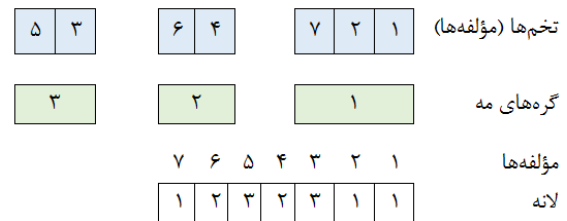
```

شکل ۸: الگوریتم استخراج زیرشبکه‌های مش کامل.

هستند. مقدار تخصیص داده شده به هر تخم، یک عدد صحیح بین ۱ و  $|N|$  می‌باشد که گره مه محل استقرار مؤلفه‌ها را نشان می‌دهد. شکل ۶ مثالی از الگوی کدگذاری مؤلفه‌های کاربرد در گره‌های مه و زیستگاه مربوط را نشان می‌دهد.

**۲-۵ الگوریتم جستجوی فاخته سفارشی شده**

شبه‌کد کلی الگوریتم جستجوی فاخته برای حل مسأله استقرار مؤلفه‌ها در زیرشبکه مه در شکل ۷ ارائه شده است. الگوریتم ۱ در شکل ۷ مشخصات مسأله را به عنوان ورودی دریافت کرده و راه حل بهینه را بر طبق تابع هدف مسأله ارائه می‌دهد. این الگوریتم با تعداد اجرای  $MaxIteration$  خاتمه می‌یابد. قبل از اجرای حلقه اصلی برنامه که بین خطوط ۱۰ تا ۲۳ می‌باشد، مراحل پیش‌پردازش انجام می‌شود. الگوریتم‌های ۲ و ۳ در شکل‌های ۸ و ۹ که در بخش ۵-۳ بیان شده است، برای استخراج زیرشبکه‌های مش کامل اجرا می‌شوند. سپس در حلقه اصلی، دو بخش اجرا می‌گردد. راه حل‌های جدید در الگوریتم ۵ بر اساس توزیع پرواز لوی انجام می‌شود. الگوریتم ۴ در شکل ۱۰ راه حل‌های غیر ممکن را اصلاح می‌کند. سپس برای به روز رسانی درصد از راه حل‌های کم‌ارزش‌تر بر مبنای تابع هدف، الگوریتم ۶ در شکل ۱۳ فراخوانی می‌شود. برای انجام این کار، ما  $P_a$  درصد را بر اساس بدترین رتبه راه حل‌ها انتخاب می‌کنیم. سپس الگوریتم ۶ در شکل ۱۳ آنها را با اجرای رویه گام تصادفی به روز رسانی می‌کند. مقادیر شایستگی با استفاده از الگوریتم ۷ در شکل ۱۴ برای به روز رسانی راه حل‌ها محاسبه می‌گردد. بعد از اجرای آخرین دور حلقه تکرار، راه حل بهینه برگردانده می‌شود.



شکل ۶: مثالی از الگوی کدگذاری مؤلفه‌های کاربرد.

**Algorithm 1.** The Proposed CSA**Input Parameters:**

Parameters for Application And Fog network:  
componentsDataset, Fog Network Dataset, Fog nodes and component delay and bandwidth matrix;  
Parameters for CS; PopSize: number of solution in Population, MaxIteration,  $P_a$ : Discovery Rate;  
 $N$ : Number of Fog Nodes;  
 $M$ : Number of Components

**Output:****Optimal Solution**

```

1: i = 1;
2: Fog_DS = Initial Fog Data Structure using Datasets;
3: Full_Meshs = call Algorithm2 for Exploit FullMeshs;
4: Candidate_Full_Meshes = call Algorithm3 for Find desired Full Meshs;
5: Solutionsi = initial Random Population (PopSize, componentsDataset, Candidate_Full_Meshes);
6: Update Fog_DS using Solutionsi information;
7: call Algorithm4 for Repair infeasible Solutions in Solutionsi;
8: call Algorithm7 for Calculate Solutions cost in Solutionsi;
9: Sorting Solutions in Solutionsi Based upon Costs;
10: While i ≤ Max Iteration do
11:   Fog_DS = Initial Fog Data Structure using Datasets;
12:   Solutionsnew = call Algorithm5;
13:   call Algorithm4 for Repair infeasible Solutions in Solutionsnew;
14:   call Algorithm7 for Calculate Solutions cost in Solutionsnew;
15:   Sorting Solutions in Solutionsnew Based upon Costs;
16:   call Algorithm6 for discovery worse Solutions in Solutionsnew with probability  $P_a$ ;
17:   call Algorithm7 for Update Solutions cost in Solutionsnew;
18:   Update Fog_DS using Solutionsnew information;
19:   Solutionstemp = merge Solutionsnew and Solutionsi;
20:   Sorting Solutionstemp Based upon Costs;
21:   Solutionsi+1 = first Population Size in Solutionstemp;
22:   i = i+1;
23: End While
Return Optimal Solution;

```

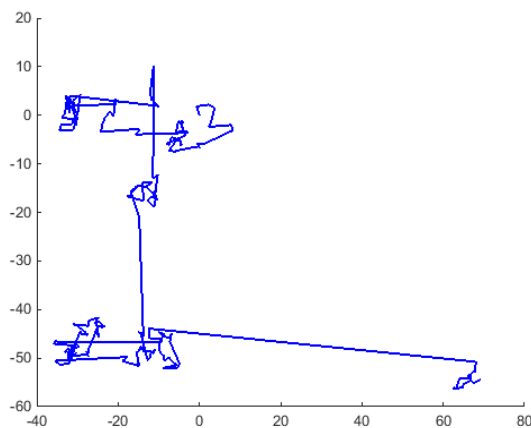
شکل ۷: الگوریتم جستجوی فاخته سفارشی شده.

مؤلفه‌های کاربرد شامل اطلاعات منابع درخواستی مؤلفه‌های کاربرد، ارتباطات مؤلفه‌ها، پیکربندی زیرساخت شبکه مه، تعداد اعضای جمعیت و حداکثر تعداد تکرار را به عنوان پارامترهای ورودی دریافت می‌کند و سپس یک طرح استقرار بهینه را به عنوان خروجی ارائه می‌دهد. در ادامه این بخش الگوی کدگذاری، جستجوی فاخته سفارشی شده، پیش‌پردازش، عملگرها و تابع شایستگی<sup>۱</sup> تشریح می‌گردد.

**۱-۵ الگوی کدگذاری**

یکی از مهم‌ترین مؤلفه‌های الگوریتم فاخته لانه می‌باشد و کدگذاری لانه روی کارایی الگوریتم تأثیر می‌گذارد. الگوهای کدگذاری متفاوتی برای مسأله‌های مختلف استفاده شده است. هر لانه یک راه حل ممکن برای مسأله ما در استقرار مؤلفه‌های کاربرد در گره‌های مه می‌باشد. یک لانه شامل  $|M|$  تخم فاخته است که هر کدام بیانگر یک مؤلفه کاربرد





شکل ۱۱: مثالی از مسیر حرکت پرواز لوی.

**Algorithm 5. Elitism****Input Parameters:**

define **Solutions** is current Population of Solutions;  
Define **best** is best solution in current population;

**Output:**

**Solutions<sub>new</sub>** is Next generation of population;

- 1: **For Each** solution in **Solutions** as **s do**
  - 2:  $S_{new} =$  combined **s** and **best** using random walk or levy flight;
  - 3: **Boundary\_update** ( $S_{new}$ );
  - 4: **Solutions<sub>new</sub>** = [**Solutions<sub>new</sub>**,  $S_{new}$ ];
  - 5: **End For Each**
- Return Solutions<sub>new</sub>**;

شکل ۱۲: الگوریتم نخبه‌گرایی.

**Algorithm 6. Discovery Inferior Solutions and Update****Input Parameters:**

define **Solutions<sub>new</sub>** is new Population of Solutions;  
define  $P_a$  is Discovery Rate

**Output:**

updated **Solutions<sub>new</sub>**;  
/\* A solution is an interchange with a nest in cuckoo search context \*/

- 1: **Worse\_solutions** = a fraction of **Solutions** based upon  $P_a$  Probability;
  - 2: **For Each** **Worse\_solutions** in **Solutions<sub>new</sub>** as **ws do**
  - 3:  $ws_{new} =$  modify **ws** using random walk ();
  - 4: **Boundary\_update** ( $ws_{new}$ );
  - 5: Replace **ws** in **Solutions<sub>new</sub>** with  $ws_{new}$
  - 6: **End For Each**
- Return updated Solutions<sub>new</sub>**;

شکل ۱۳: الگوریتم کشف و اصلاح راه حل‌های بدتر.

**Algorithm 7. Fitness Function****Input Parameters:**

define **Solutions** is a Population;

**Output:**

Updated each solution in **Solutions** with solution cost

- 1: **For Each** solution in **Solutions** as solution<sub>i</sub> **do**
  - 2: **Energy** = calculate computing and network resource energy consumption of all fog node in solution<sub>i</sub>;
  - 3: solution<sub>i</sub> **Cost** in **Solutions** = **Energy**;
  - 4: **End For Each**
- Return Updated Solutions**;

شکل ۱۴: الگوریتم تابع شایستگی.

فضای جستجو برای استخراج طرح بهینه استقرار است و مزیت دوم تأمین اشتراکی حسگرها و نرم‌افزارهای مورد نیاز مؤلفه‌ها توسط زیرشبکه‌های مش کامل می‌باشد که هم‌ارز مسئله Clique در تئوری گراف است.

در الگوریتم ۲ در شکل ۸، در ابتدای ورود به شرط *while* در خطوط ۳ تا ۱۱ ابتدا همه گره‌هایی که به هم متصل هستند استخراج شده و هر دوتایی در یک سطر از آرایه *Full\_Meshs* ذخیره می‌شود. در خطوط ۱۳ تا ۲۰ در حلقه *for* هر گره *i* با سطرهایی از *Full\_Meshs*

**Algorithm 3. Find Candidate Fullmesh Subnetworks****Input Parameters:**

define  $fn = (1, \dots, n)$  is fog node list;  
define  $Appcmp = (1, \dots, m)$  is Application components list;  
define  $FN\_BW =$  array ( $n \times n$ ) is bandwidth between fog nodes;  
define  $FN\_Latency =$  array ( $n \times n$ ) is Latency between fog nodes;  
define  $Appcmp\_BW =$  array ( $m \times m$ ) is bandwidth between components;  
define  $Appcmp\_Latency =$  array ( $m \times m$ ) is Latency between components;  
define **AppcmpDataset** is Application components resource requirements;  
define **FNDataset** is Fog nodes resources;  
define **Full\_Meshs** is Full Meshs list (**Algorithm2 output**);

**Output:**

Candidate Full Meshs

- 1: **For each** **Full\_Meshs** as row<sub>i</sub> **do**
  - 2:  $Latency\_BW\_status =$  Check  $Latency\_BW$  ( $Appcmp\_Latency$ ,  $Appcmp\_BW$ , row<sub>i</sub>, **FNDataset**, **FN\_BW**, **FN\_Latency**);  
//constrain 15
  - 3:  $HR\_status =$   $HR\_chk$  ( $AppcmpDataset$ , **FNDataset**, row<sub>i</sub>);  
//constrain 16
  - 4:  $SR\_status =$   $SR\_chk$  ( $AppcmpDataset$ , **FNDataset**, row<sub>i</sub>);  
//constrain 17
  - 5:  $S\_status =$   $S\_chk$  ( $AppcmpDataset$ , **FNDataset**, row<sub>i</sub>);  
//constrain 18
  - 6: **If**  $Latency\_BW\_status$  &  $HR\_status$  &  $SR\_status$  &  $S\_status$  are true **then**
  - 7: **Candidate\_Full\_Meshs** = [**Candidate\_Full\_Meshs**; row<sub>i</sub>];
  - 8: **End if**
  - 9: **End For**
- Return Candidate\_Full\_Meshs**;

شکل ۹: الگوریتم استخراج زیرشبکه‌های مش کامل کاندیدا.

**Algorithm 4. Repair Infeasible Solutions****Input Parameters:**

define **Fog\_Data\_Structure** is Fog nodes status in each solution;  
define **Full\_Mesh** is Candidate Nodes taken from **Algorithm 3**;  
define  $Appcmp = (1, \dots, m)$  is Application component list;  
define **AppcmpDataset** is Application component Specification;  
define **Solutions** is all solution in a population;

**Output:**

Feasible Solutions and **Fog\_Data\_Structure**

- 1: **For Each** **Solutions** As solution **do**
  - 2: **For Each** Fog node in solution as sourceFN **do**
  - 3: **If** sourceFN Resource overload is true **then**
  - 4: violation = true;
  - 5: **While** violation is true **do**
  - 6: find **Appcmp** with minimum dependency to other component on sourceFN and migrate it to other fog node in **Full\_Mesh** with enough resource. Candidate **Appcmp** has maximum dependency to components on destination Fog node;
  - 7: Update solution;
  - 8: Update **Fog\_Data\_Structure**;
  - 9: **If** sourceFN resource overload is false **then**
  - 10: violation = false;
  - 11: **End If**
  - 12: **End While**
  - 13: **End If**
  - 14: **End For Each**
  - 15: **End For Each**
- Return Solutions** and **Fog\_Data\_Structure**;

شکل ۱۰: الگوریتم ترمیم راه حل‌های غیر ممکن.

**۳-۵ پیش‌پردازش**

در مرحله پیش‌پردازش با استفاده از الگوریتم ۲ در شکل ۸ مجموعه‌ای از زیرشبکه‌های مش کامل در شبکه مه برای استقرار مؤلفه‌ها انتخاب می‌شوند. استخراج این زیرشبکه‌ها چندین مزیت دارد: مزیت اول کاهش

$$y = (1-u)^{\frac{1}{\alpha}} \quad (22)$$

که متغیر  $u$  یک متغیر یکنواخت در بازه  $[0, 1]$  است و پارامتر  $\alpha$  با استفاده از (۲۳) به دست می‌آید

$$\alpha = G^{\frac{1}{\epsilon}} \quad (23)$$

در عبارت فوق پارامتر  $G$  شماره نسل است [۱۸]. بعد از آن، خط ۳ راه حل‌های به دست آمده را بر طبق کرانه‌های دامنه مسأله به روز رسانی می‌کند. به عبارت دیگر، در ابتدا از مفهوم پرواز لوی پیوسته جهت پیمایش فضای جستجو استفاده می‌شود و سپس مقادیر پیوسته محاسبه‌گردیده به نزدیک‌ترین گره مه معتبر گسسته‌سازی می‌شوند. در خط ۴، راه حل جدید به دست آمده به لیست راه حل‌های جدید اضافه می‌شود و بدین ترتیب راه حل‌های نسل بعدی تولید می‌گردد. پیچیدگی الگوریتم ۵ در شکل ۱۲ به خاطر تنها حلقه  $for$  آن از مرتبه  $O(PopSize)$  است.

### ۵-۶ کشف راه حل‌های بدتر

در این فرایند، کسری از راه حل‌های بدتر کشف و اصلاح می‌شوند که این عمل مشابه جهش نسل می‌باشد [۱۷] تا [۲۰]. در خط ۱ از الگوریتم ۶ در شکل ۱۳، تعدادی از راه حل‌ها با احتمال  $P_a$  از جمعیت جدید استخراج می‌شوند. در خط ۳ از حلقه  $for$  با استفاده از گام تصادفی، راه حل‌های استخراج‌شده اصلاح می‌شوند، در خط ۴ مقادیر نامعتبر در راه حل به دست آمده، به روز رسانی می‌شوند و در نهایت در خط ۵، راه حل‌های اصلاح‌شده در  $Solutions_{new}$  جایگزین می‌گردند.

پیچیدگی زمانی الگوریتم ۶ در شکل ۱۳  $\theta(P_a.PopSize)$  است، بنابراین از مرتبه  $O(PopSize)$  می‌باشد زیرا تنها یک حلقه  $for$  در الگوریتم وجود دارد و همچنین  $P_a < 1$  است.

### ۵-۷ تابع شایستگی

به طور کلی، یکی از موضوعات مهم، نحوه ارزیابی عملکرد هر راه حل در جمعیت است. از آنجا که هر راه حل متناسب با یک طرح استقرار مؤلفه است، تابع هدف (۱۳)، یعنی میزان مصرف انرژی در طرح استقرار مؤلفه‌ها بر اساس اطلاعات دریافتی از راه حل با استفاده از تابع پیشنهادی در الگوریتم ۷ در شکل ۱۴ محاسبه می‌شود.

واضح است که پیچیدگی زمانی الگوریتم ۷ در شکل ۱۴ به دلیل استفاده از تنها حلقه تکرار از مرتبه  $O(PopSize)$  است.

### ۶- شبیه‌سازی و ارزیابی

برای بررسی کارایی الگوریتم جستجوی فاخته پیشنهادی در استخراج طرح استقرار بهینه مؤلفه‌های کاربرد در زیرساخت مه، یک سری آزمایش‌ها طراحی و انجام می‌شود. ما از معیار ارزیابی مصرف انرژی برای ارزیابی رویکرد پیشنهادی استفاده کردیم. به طور کلی عوامل مختلفی روی مصرف انرژی گره‌های مه تأثیر می‌گذارند. این عوامل شامل حجم محاسبات، تکنولوژی ارتباطی، میزان ترافیک تبادل، فاصله گره‌ها از هم و غیره می‌باشد. برای محاسبه انرژی مصرفی یک گره مه، انرژی مصرفی حاصل از پردازش مؤلفه‌های کاربرد روی گره و همچنین انرژی مصرفی به خاطر انتقال اطلاعات بین گره‌های مه در نظر گرفته می‌شود. ما نتایج به دست آمده از طریق الگوریتم پیشنهادی را با الگوریتم‌های فراابتکاری گرگ خاکستری [۲۱]، ژنتیک [۲۲] و ازدحام ذرات [۲۳] مقایسه می‌کنیم.

که در آن وجود ندارد، مقایسه می‌شود و اگر به همه گره‌های موجود در آن متصل است به لیست گره سطر جاری اضافه می‌شود. در خط ۲۲ سطرهای تکراری از  $Full\_Meshs$  حذف می‌شوند. این روند تا پایان شرط  $while$  ادامه یافته و در نهایت آرایه  $Full\_Meshs$  که شامل مجموعه‌ای از زیرشبکه‌های مش کامل است، برگردانده می‌شود. شرط خاتمه الگوریتم ۲ در شکل ۸، تعداد کلیک‌های مطلوب با اندازه  $k$  است و به عبارت دیگر، حلقه اصلی  $k$  بار تکرار می‌شود. از آنجایی که دستورات مؤثر الگوریتم ۲ در شکل ۸ در حلقه  $while$  قرار دارند، پیچیدگی زمانی از مرتبه  $O(K.N^2)$  می‌باشد که  $K < N$  است. پس از استخراج مش‌های کامل، با توجه به شرط‌های (۱۴) تا (۱۸) در بخش ۴ با استفاده از الگوریتم ۳ در شکل ۹، تعدادی از مش‌های کامل برای استقرار مؤلفه‌ها انتخاب می‌شوند.

در خط ۲ از الگوریتم ۳ در شکل ۹ اگر منابع مورد نیاز در پهنای باند و تأخیر برای همه مؤلفه‌های کاربرد توسط زیرشبکه مش کامل جاری تأمین شود،  $Latency\_BW\_status$  برابر با  $true$  می‌شود. همچنین در خطوط ۳ تا ۵ اگر منابع سخت‌افزار، نرم‌افزار و حسگر توسط مش کامل جاری تأمین شود، مقادیر  $HR\_Status$ ،  $SR\_status$  و  $S\_status$  برابر با  $true$  خواهند شد. در خط ۶ و ۷، اگر همه منابع توسط مش کامل جاری تأمین شود، زیرشبکه مش کامل مورد نظر در لیست مش‌های کامل انتخابی قرار خواهد گرفت. پیچیدگی زمانی الگوریتم ۳ در شکل ۹ از مرتبه  $O(N+M)$  است، به این دلیل که کار اصلی در حلقه  $for$  بین خط‌های ۱ تا ۹ انجام می‌شود.

### ۵-۸ مرحله آغازین

در تولید نسل اول الگوریتم پیشنهادی، جمعیت اولیه به طور تصادفی تولید می‌شود. به منظور کاهش زمان محاسبه اجرای الگوریتم فاخته، می‌توان دامنه مقادیر هر تخم را بر اساس طرح کدگذاری توصیف‌شده در بخش ۵-۱ از پیش تعیین کرد. توجه داشته باشید که در تولید راه حل‌ها ممکن است بخشی از راه حل‌ها محدودیت‌های ذکرشده را نقض کنند و به راه حل غیر ممکن تبدیل شوند. در الگوریتم پیشنهادی برای بهره‌برداری حداکثری از جمعیت تولیدشده، راه حل‌های غیر ممکن را به کمک الگوریتم ۴ در شکل ۱۰ ترمیم کرده و از آنها در یافتن راه حل بهینه نهایی استفاده می‌کنیم.

پیچیدگی الگوریتم ۴ در شکل ۱۰ از مرتبه  $O(N.PopSize)$  است، زیرا حلقه‌های  $for$  تودرتو مؤثرترین دستورات در این الگوریتم می‌باشند.

### ۵-۵ نسخه‌گرایی

برای تولید نسل بعدی راه حل‌ها از مکانیزم نسخه‌گرایی استفاده می‌شود تا بهترین راه حل‌ها به نسل بعدی منتقل شود و چنین انتخابی از بهترین‌ها به اطمینان از همگرایی صحیح الگوریتم کمک می‌کند. در خط ۲ الگوریتم ۵ در شکل ۱۲، از ترکیب هر یک از راه حل‌ها ( $s$ ) و بهترین راه حل از جمعیت فعلی با استفاده از پرواز لوی [۱۷]، راه حل جدید به دست می‌آید. شکل ۱۱ نمونه‌ای از مسیر حرکت پرواز لوی با توزیع یکنواخت را نشان می‌دهد.

در خط ۲، الگوریتم ۵ در شکل ۱۲ با استفاده از توزیع لوی، عدد  $\gamma$  را به عنوان شماره تصادفی یک لانه بر اساس (۲۲) تولید می‌کند

جدول ۳: سناریوهای شبیه‌سازی.

سناریوها	۱	۲	۳	۴
گره‌های مه	۱۰	۱۵	۲۰	۲۵
مؤلفه‌ها	۲۰	۲۵	۳۰	۴۰

جدول ۴: منابع گره‌های مه.

گره‌های مه	۱	۲	۳	۴	۵
CPU (GHz)	۱٫۰۲	۱٫۱۵	۱٫۳۸	۱٫۴۶	۱٫۰۶
اندازه حافظه (GB)	۱٫۳	۱٫۶	۱٫۲	۱٫۴	۱٫۳
آستانه مصرف CPU	۰٫۹۸	۰٫۹۳	۰٫۹۶	۰٫۹۴	۰٫۹۲
آستانه مصرف حافظه	۱٫۰۰	۰٫۹۹	۰٫۹۱	۰٫۹۲	۰٫۹۸
حداقل مصرف انرژی	۹۴	۸۲	۹۹	۸۱	۹۱
حداکثر مصرف انرژی	۱۳۳	۱۳۲	۱۳۳	۱۴۷	۱۴۲
حسگرها	۱	۱٫۲	۱٫۲	۲	۰
نرم‌افزارها	۰	۱٫۲	۲	۰	۱٫۲
انرژی انتقال	۰٫۲	۰٫۲	۰٫۲	۰٫۱	۰٫۱

جدول ۵: پهنای باند بین گره‌های مه.

گره‌های مه	۱	۲	۳	۴	۵
۱	۱	۰٫۹۸	۰٫۸۰	۰٫۸۹	۰٫۹۷
۲	۰٫۸۴	۱	۰٫۸۲	۰٫۹۴	۰٫۹۲
۳	۰٫۹۳	۰٫۹۴	۱	۰٫۹۷	۰
۴	۰٫۸۸	۰٫۹۲	۰٫۹۱	۱	۱
۵	۰٫۹۲	۰٫۹۹	۰	۰٫۹۰	۱

جدول ۶: تأخیر بین گره‌های مه.

گره‌های مه	۱	۲	۳	۴	۵
۱	۰	۰٫۱۷	۰٫۱۹	۰٫۱۰	۰٫۱۰
۲	۰٫۱۲	۰	۰٫۱۰	۰٫۱۱	۰٫۱۸
۳	۰٫۱۸	۰٫۱۴	۰	۰٫۱۲	۱
۴	۰٫۱۶	۰٫۱۹	۰٫۱۴	۰	۰٫۱۴
۵	۰٫۱۱	۰٫۱۴	۱	۰٫۱۶	۰

جدول ۷: منابع مورد نیاز مؤلفه‌های کاربرد.

مؤلفه‌ها	۱	۲	۳	۴	۵
CPU	۰٫۱۵	۰٫۱۹	۰٫۲۴	۰٫۲۶	۰٫۳۹
حافظه	۰٫۲	۰٫۲	۰٫۱	۰٫۲	۰٫۱
حسگرها	۱	۰	۱	۰	۲
نرم‌افزارها	۰	۱	۰	۱٫۲	۱

جدول ۸: پهنای باند مورد نیاز مؤلفه‌های کاربرد.

مؤلفه‌ها	۱	۲	۳	۴	۵
۱	۱	۰	۰	۰	۰
۲	۰	۱	۰٫۳۳	۰	۰٫۳۲
۳	۰	۰	۱	۰٫۳۱	۰٫۲۰
۴	۰	۰	۰	۱	۰٫۳۹
۵	۰	۰	۰	۰	۱

جدول ۹: تأخیر مورد نیاز مؤلفه‌های کاربرد.

مؤلفه‌ها	۱	۲	۳	۴	۵
۱	۰	۱	۱	۱	۱
۲	۰	۰	۰٫۲۰	۱	۰٫۲۸
۳	۰	۰	۰	۰	۰٫۲۱
۴	۰	۰	۰	۰	۰٫۲۰
۵	۰	۰	۰	۰	۰

جدول ۱۰: پارامترهای فاخته، گرگ خاکستری، ژنتیک و ازدحام ذرات.

الگوریتم‌ها	پارامترها	اندازه جمعیت	حداکثر تکرار
جستجوی فاخته	$P_g$	۰٫۲۵	
گرگ خاکستری	برگرفته از [۲۱]		
ژنتیک	$CP$	۰٫۷	
	$MP$	۰٫۴	۱۰۰
	$W$	۱	
ازدحام ذرات	$C_1$	۱٫۵	
	$C_2$	۲	

## ۶-۱ تنظیمات شبیه‌سازی

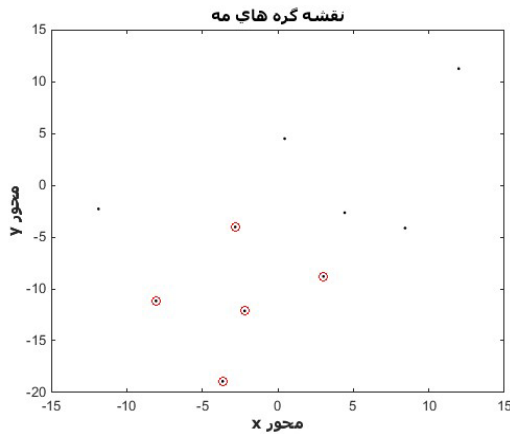
برای ارزیابی چارچوب پیشنهادی، ۴ سناریو اجرا شد. در هر سناریو تعداد مؤلفه‌ها و تعداد گره‌های مه تغییر می‌یابد. جزئیات سناریوها در جدول ۳ نشان داده شده است. ما همه آزمایش‌ها را روی کامپیوتری با پردازنده اینتل Core i۳ ۳۸۰M با دو هسته فیزیکی، نرخ کلاک ۲٫۵۳ گیگاهرتز و چهار پردازنده منطقی و ۸ گیگابایت حافظه انجام دادیم.

به منظور ارزیابی قابلیت الگوریتم پیشنهادی در ارائه طرح بهینه استقرار مؤلفه‌های کاربرد، فرض بر این است که گره‌های مه از نظر قابلیت پردازش، ظرفیت ذخیره‌سازی، پهنای باند و تأخیر ناهمگن هستند. برای نمایش بهتر، نمونه‌ای از اطلاعات مجموعه داده<sup>۱</sup> شبکه مه با ۵ گره در جدول ۴ تا ۶ نشان داده شده است. همان طور که در جدول ۴ مشاهده می‌شود، منابع محاسباتی، آستانه مصرف CPU و حافظه و همچنین حداقل و حداکثر مصرف انرژی، نوع حسگرها و نرم‌افزارهای پشتیبانی شده و انرژی انتقال برای هر گره، متفاوت در نظر گرفته شده است. مقدار ۰ در

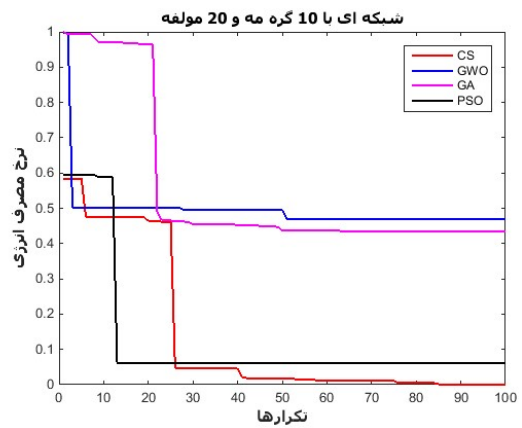
حسگرها و نرم‌افزارهای جدول ۴ به معنای عدم پشتیبانی از نرم‌افزار و حسگرها می‌باشد و مقادیر ۱ و ۲، نوع حسگرها و نرم‌افزارهای قابل پشتیبانی را نشان می‌دهد. در جدول ۵ و ۶ پهنای باند و تأخیر در ارتباط مستقیم بین گره‌ها آمده و مقادیر این جداول، اعدادی نرمال بین ۰ و ۱ در نظر گرفته شده است. پهنای باند ۰ به معنای عدم ارتباط بین گره  $i$  و  $j$  و پهنای باند ۱ به معنای این است که گره  $i$  همان گره  $j$  می‌باشد. در جدول ۶ تأخیر ۱ بین دو گره به معنای عدم ارتباط بین دو گره و تأخیر ۰ به معنای یکسان بودن دو گره است.

نمونه‌ای از مشخصات مؤلفه‌های کاربرد شامل CPU، حافظه، نوع حسگر و نرم‌افزار مورد نیاز، همچنین ارتباطات و تأخیر بین آنها در جداول ۷، ۸ و ۹ (مشابه جداول ۴، ۵ و ۶) نشان داده شده است.

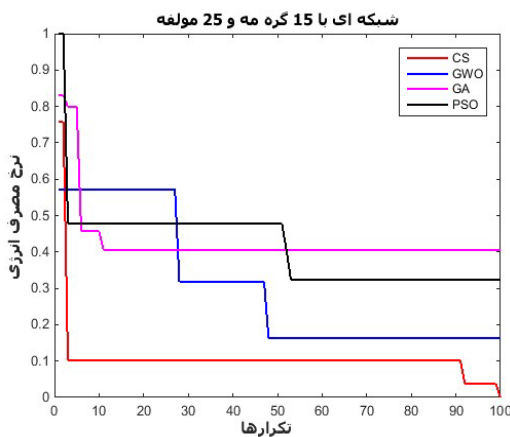
لازم به ذکر است با توجه به این که بنچمارکی<sup>۲</sup> در ادبیات در این حوزه تحقیق منتشر نشده است، مجموعه داده با محدوده مقادیر مشابه جداول ۴ تا ۹ را تولید کردیم. این مقادیر، توان پردازش و ظرفیت ذخیره‌سازی گره‌ها مه بر اساس معماری‌ها و سیستم‌های محاسباتی گره‌های مه



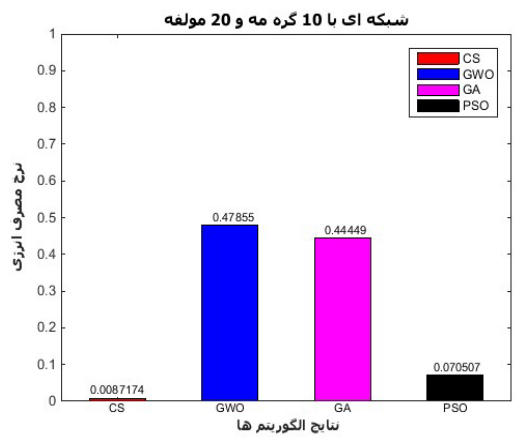
شکل ۱۷: نقشه گره های مه.



شکل ۱۵: عملکرد الگوریتم‌ها در تابع هدف TPC با افزایش تکرار.



شکل ۱۸: عملکرد الگوریتم‌ها در تابع هدف TPC با افزایش تکرار.



شکل ۱۶: نرخ بهینه مصرف انرژی در پایان شبیه‌سازی.

در پایان شبیه‌سازی، راه حل بهینه به دست آمده با توجه به تابع هدف TPC (معادله (۱۳)) توسط الگوریتم جستجوی فاخته و ازدحام ذرات در مقایسه با ژنتیک و گرگ خاکستری، نتایج بهتری را نشان می‌دهد (شکل ۱۶).

زیر شبکه مش کامل انتخابی توسط الگوریتم پیشنهادی برای توزیع مؤلفه‌ها شامل گره‌های ۳، ۴، ۵، ۸ و ۹ می‌باشد. در شکل ۱۷ موقعیت این گره‌ها در نقشه شبکه مه نشان داده شده است. زمان اجرای شبیه‌سازی در جدول ۱۱ آمده و در مقایسه با زمان اجرای سایر الگوریتم‌ها، روش پیشنهادی از زمان سپری‌شده مطلوبی برخوردار است.

### سناریوی دوم: شبکه‌ای با ۱۵ گره مه و ۲۵ مؤلفه

در این سناریو راه حل‌های به دست آمده توسط جستجوی فاخته در مراحل اولیه اجرا با توجه به (۱۳) از مقدار مطلوبی برخوردار است و با توجه به محدودیت فضای جستجو به زیر شبکه مش کامل، تا پایان شبیه‌سازی تغییر زیادی در نتیجه حاصل شده رخ نمی‌دهد (شکل ۱۸).

در پایان شبیه‌سازی، راه حل بهینه به دست آمده توسط الگوریتم گرگ خاکستری، ازدحام ذرات و ژنتیک در تابع هدف (۱۳) نتایج نزدیک به هم را نشان می‌دهد. با این حال با توجه به شکل ۱۹، برتری روش پیشنهادی در ارائه راه حلی با تابع هدف کمینه، نسبت به سایر راه حل‌ها مشاهده می‌شود.

در شکل ۲۰ موقعیت گره‌های مش کامل انتخابی در نقشه شبکه نشان داده شده است. با توجه به نقشه و فاصله توزیع گره‌های مه، چندین زمان اجرای شبیه‌سازی در جدول ۱۲ آمده است. در این سناریو نیز در مقایسه با زمان اجرای سایر الگوریتم‌ها، روش پیشنهادی از زمان سپری‌شده مطلوبی برخوردار است.

جدول ۱۱: زمان اجرای شبیه‌سازی.

جستجوی فاخته	۱۸۷٫۹۲ s	ژنتیک	۲۱۳٫۱۲ s
گرگ خاکستری	۱۸۶٫۳۴ s	ازدحام ذرات	۱۸۶٫۹۳ s

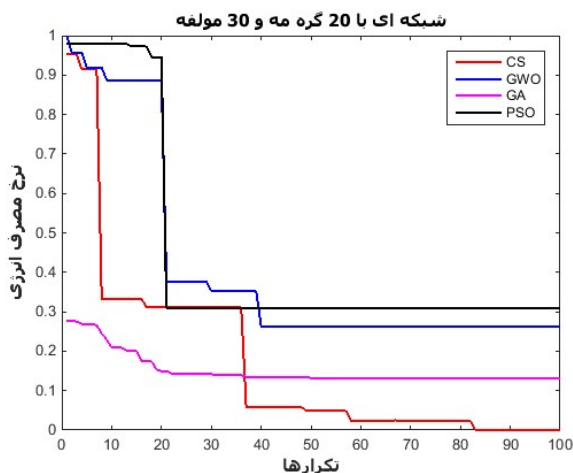
فعلی مانند دستیارهای دیجیتال شخصی، تلفن‌های هوشمند، رایانه‌های داخلی و غیره انتخاب شده‌اند. پارامترهای الگوریتم‌های جستجوی فاخته، گرگ خاکستری، ژنتیک و ازدحام ذرات در جدول ۱۰ ارائه شده است. در این جدول،  $P_g$  نرخ اکتشاف راه حل‌ها در جستجوی فاخته،  $CP$  و  $MP$  نرخ تقاطع و جهش در الگوریتم ژنتیک و پارامترهای  $W$ ،  $C_1$  و  $C_2$  به ترتیب ضرایب اینرسی، یادگیری محلی و یادگیری سراسری می‌باشند.

### ۶-۲ نتایج شبیه‌سازی

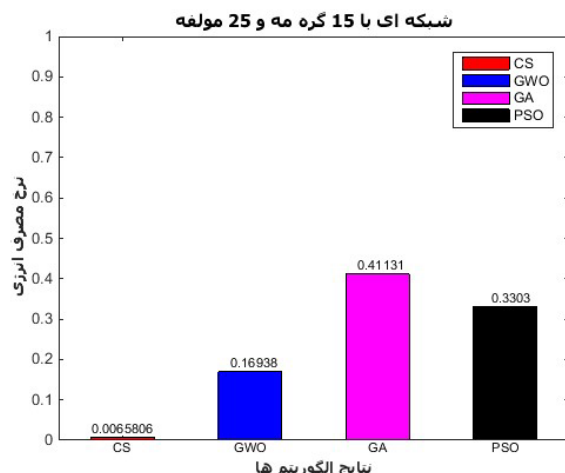
برای مقایسه عملکرد الگوریتم‌ها از نمودارهای وضعیت تابع هدف در تکرار الگوریتم‌ها، نرخ کمینه مصرف انرژی و همچنین زمان سپری‌شده استفاده گردیده است. گره‌ها در شبکه مه به صورت تصادفی و با توزیع نرمال پراکنده شده‌اند. نقشه توزیع گره‌های شبکه مه در هر سناریو شبیه‌سازی نمایش داده می‌شود و همچنین در نقشه شبکه مه گره‌های میزبان مؤلفه‌های کاربرد با رنگ قرمز نشان داده می‌شوند.

### سناریوی اول: شبکه‌ای با ۱۰ گره مه و ۲۰ مؤلفه

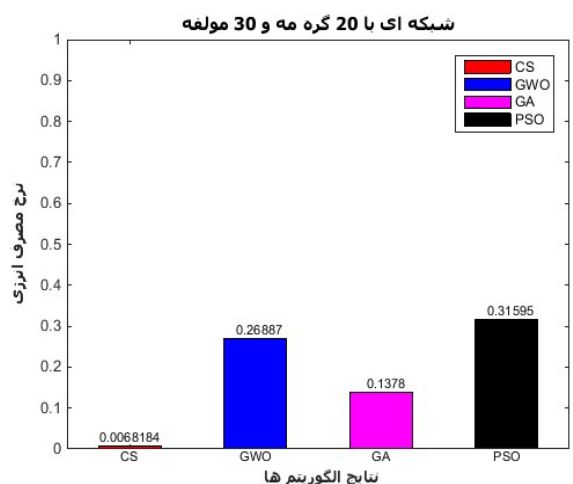
با توجه به شکل ۱۵ در مراحل اولیه اجرای شبیه‌سازی، راه حل‌های به دست آمده توسط الگوریتم‌های گرگ خاکستری و ازدحام ذرات در رقابت با روش پیشنهادی می‌باشند، اما با افزایش مراحل تکرار راه حل‌های به دست آمده توسط الگوریتم جستجوی فاخته بر نتایج سایر الگوریتم‌ها غلبه می‌کند.



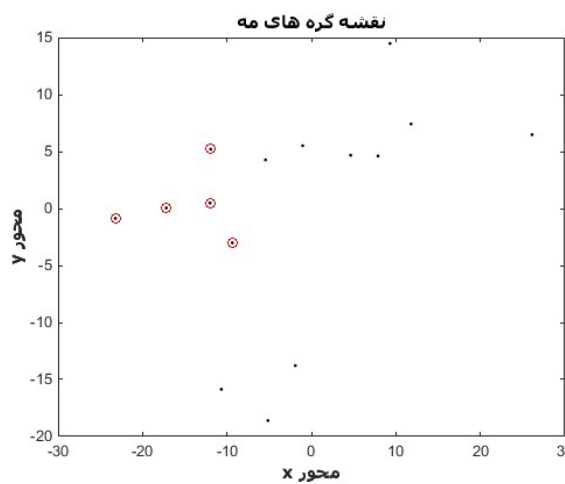
شکل ۱۹: نرخ بهینه مصرف انرژی در پایان شبیه‌سازی.



شکل ۲۰: نقشه گره‌های مه.



شکل ۲۱: عملکرد الگوریتم‌ها در تابع هدف TPC با افزایش تکرار.



شکل ۲۲: نرخ بهینه مصرف انرژی در پایان شبیه‌سازی.

جدول ۱۲: زمان اجرای شبیه‌سازی.

جستجوی فاخته	۲۶۹٫۶۵ s	ژنتیک	۴۵۷٫۱۳ s
گرگ خاکستری	۴۱۸٫۵۰ s	ازدحام ذرات	۳۹۲٫۵۴ s

جدول ۱۳: زمان اجرای شبیه‌سازی.

جستجوی فاخته	۲۱۲٫۶۸ s	ژنتیک	۲۶۳٫۴۵ s
گرگ خاکستری	۲۱۶٫۹۶ s	ازدحام ذرات	۲۱۳٫۸۷ s

جدول ۱۴: زمان اجرای شبیه‌سازی.

جستجوی فاخته	۲۶۹٫۹۴ s	ژنتیک	۲۱۹٫۵۱ s
گرگ خاکستری	۲۷۷٫۰۹ s	ازدحام ذرات	۲۷۶٫۷۷ s

شده است. زمان اجرای شبیه‌سازی در جدول ۱۳ آمده و در مقایسه با زمان اجرای سایر الگوریتم‌ها، روش پیشنهادی از زمان سپری‌شده مطلوبی برخوردار است.

### سناریوی چهارم: شبکه‌ای با ۲۵ گره مه و ۴۰ مؤلفه

در شکل ۲۴ رقابت نزدیک الگوریتم‌ها در استخراج راه حل‌هایی با مقدار کمینه در تابع هدف TPC مشاهده می‌شود، اما در نهایت در مراحل پایانی، تکرار روش پیشنهادی بر سایر الگوریتم‌ها با اختلاف کم غلبه می‌کند.

راه حل بهینه به دست آمده با توجه به تابع هدف (۱۳)، اختلاف پایین روش پیشنهادی و الگوریتم‌های ازدحام ذرات و گرگ خاکستری را در پایان شبیه‌سازی نشان می‌دهد (شکل ۲۵).

زیرشبکه مش‌کامل انتخابی توسط الگوریتم پیشنهادی برای توزیع مؤلفه‌ها شامل گره‌های ۲، ۴، ۹، ۱۰، ۱۱، ۱۳، ۱۴، ۱۷، ۲۰ و ۲۲ می‌باشد. در شکل ۲۶ موقعیت گره‌های مش‌کامل انتخابی در نقشه شبکه نشان داده شده است.

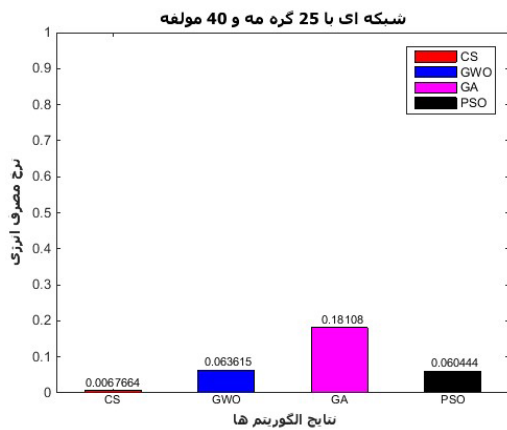
زمان اجرای شبیه‌سازی در جدول ۱۴ نشان داده شده که در مقایسه با زمان اجرای سایر روش‌ها، الگوریتم پیشنهادی از زمان سپری‌شده مطلوبی برخوردار است.

### سناریوی سوم: شبکه‌ای با ۲۰ گره مه و ۳۰ مؤلفه

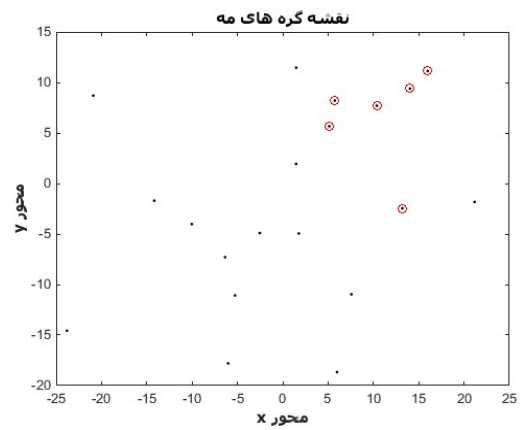
در نمودار شکل ۲۱ برتری الگوریتم ژنتیک نسبت به روش پیشنهادی در بخشی از مراحل تکرار مشاهده می‌شود اما در نهایت، روش پیشنهادی در مقایسه با ژنتیک و سایر الگوریتم‌ها، هدف ما را در (۱۳) تأمین می‌کند.

در پایان شبیه‌سازی، راه حل بهینه به دست آمده با توجه به تابع هدف (۱۳) توسط الگوریتم جستجوی فاخته پیشنهادی و ژنتیک در مقایسه با ازدحام ذرات و گرگ خاکستری، نتایج بهتری را نشان می‌دهد (شکل ۲۲).

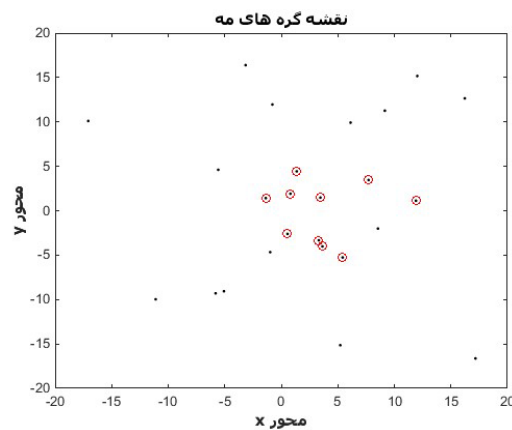
با افزایش تعداد گره‌ها در مه، تعداد مش‌های کامل قابل استخراج برای توزیع مؤلفه‌ها افزایش می‌یابد، اما با توجه به ماهیت ناهمگن گره‌ها و مؤلفه‌های کاربرد، مش‌های کامل کاندیدا برای توزیع مؤلفه‌ها محدود می‌باشد. زیرشبکه مش‌کامل انتخابی توسط الگوریتم پیشنهادی برای توزیع مؤلفه‌ها شامل گره‌های ۴، ۶، ۱۳، ۱۴، ۱۷ و ۱۹ می‌باشد. در شکل ۲۳ موقعیت گره‌های مش‌کامل انتخابی در نقشه شبکه نشان داده



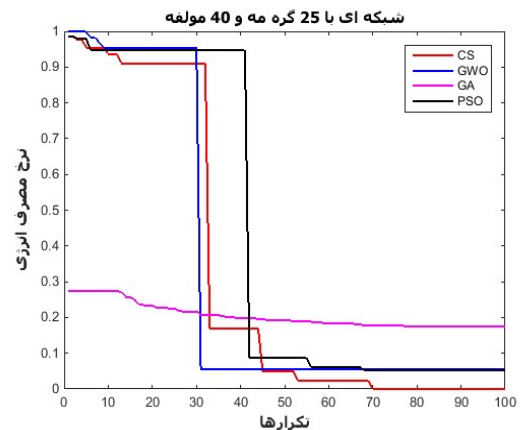
شکل ۲۵: نرخ بهینه مصرف انرژی در پایان شبیه‌سازی.



شکل ۲۳: نقشه گره‌های مه.



شکل ۲۶: نقشه گره‌های مه.



شکل ۲۴: عملکرد الگوریتم‌ها در تابع هدف TPC با افزایش تکرار.

برای رفع چالش استقرار مؤلفه‌های کاربرد، یک الگوریتم فراابتکاری سفارشی مبتنی بر فاخته ارائه داده‌ایم. مجموعه‌ای از آزمون‌های شبیه‌سازی را انجام داده و نتایج به دست آمده را با گرگ خاکستری، ژنتیک و ازدحام ذرات مقایسه کردیم. با بررسی کلی آزمایش‌ها مشخص شد که روش پیشنهادی بر سایر الگوریتم‌ها برتری دارد، زیرا در تمام آزمایش‌ها، راه حل‌های به دست آمده توسط الگوریتم پیشنهادی از نظر مصرف انرژی بر راه حل سایر الگوریتم‌ها غلبه می‌کند. در آینده، ما رویکرد استقرار پویای مؤلفه‌های کاربرد را بررسی خواهیم کرد. علاوه بر این در استقرار کاربردهای کاربر، پهنای باند مصرفی، تأخیر کلی کاربرد و هزینه پرداختی برای تأمین منابع مه و نوع کاربرد کاربر نیز در نظر گرفته خواهد شد.

## مراجع

- [1] OpenFog Consortium Architecture Working Group, *An OpenFog Architecture Overview*, Available: [https://www.iiconsortium.org/pdf/OpenFog\\_Reference\\_Architecture\\_2\\_09\\_17.pdf](https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf), 2017.
- [2] S. Azimi, C. Pahl, and M. Hosseini Shirvani, "Particle swarm optimization for performance management in multi-cluster IoT edge architectures," in *Proc. 10th Int. Conf. on Cloud Computing and Services Science*, pp. 328-337, Prague, Czech Republic, 7-9 May 2020.
- [3] M. Hosseini Shirvani, "A hybrid meta-heuristic algorithm for scientific workflow scheduling in heterogeneous distributed computing systems," *Engineering Applications of Artificial Intelligence*, vol. 90, Article ID: 103501, Apr. 2019.
- [4] H. Hong, P. Tsai, and C. Hsu, "Dynamic module deployment in a fog computing platform," in *Proc 18th Asia-Pacific Network Operations and Management Symp.*, 6 pp., Kanazawa, Japan, 5-7 Oct. 2016.

جدول ۱۴: زمان اجرای شبیه‌سازی.

جستجوی فاخته	۳۶۹٫۶۵ s	ژنتیک	۴۵۷٫۱۳ s
گرگ خاکستری	۴۱۸٫۵۰ s	ازدحام ذرات	۳۹۲٫۵۴ s

در آخر، روش پیشنهادی در میانگین مقادیر بهینه به دست آمده برای تابع هدف (۱۳) یعنی TPC، در کل سناریوهای شبیه‌سازی شده ۲۳٪ بهتر از گرگ خاکستری، ۲۸٪ بهتر از ژنتیک و ۱۸٪ بهتر از ازدحام ذرات عملکرد داشته است.

## ۳-۶ پیچیدگی زمانی

با محاسبه پیچیدگی زمانی همه الگوریتم‌های فرعی، اکنون پیچیدگی زمان الگوریتم ۱ به راحتی قابل محاسبه است. مرحله پیش‌پردازش به میزان  $O(M + K.N^2 + M + N)$  طول می‌کشد که از مرتبه  $O(M + K.N^2)$  است. همچنین حلقه اصلی  $MaxIteration$  بار تکرار می‌شود. برای حلقه اصلی،  $MaxIteration \times (N.PopSize + PopSize^2)$ ، اگر  $N$  را کوچک‌تر از  $PopSize$  در نظر بگیریم، پیچیدگی زمانی الگوریتم ۱ از مرتبه  $O(M + K.N^2 + MaxIteration.PopSize^2)$  است که نسبتاً قابل قبول می‌باشد.

## ۷- نتیجه‌گیری و کار آینده

در این مقاله، ما بر روی مسئله استقرار مطمئن مؤلفه‌های کاربرد در محیط مه و ابر تمرکز کردیم. برای حل مسأله استقرار در محاسبات مه، یک مدل ارائه دادیم و از معیار ارزیابی مصرف انرژی در این مطالعه استفاده نمودیم.

- [19] M. Tavana, S. Shahdi-Pashaki, E. Teymourian, F. J. Santos-Arteaga, and M. Komaki, "A discrete cuckoo optimization algorithm for consolidation in cloud computing," *Computers & Industrial Engineering*, vol. 115, pp. 495-511, Jan. 2017.
- [20] M. Hosseini Shirvani and S. Farzai, "Optimal deployment of IoT application components on hybrid fog2cloud infrastructure for reduction of power consumption toward green computing by cuckoo search algorithm," in *Proc. of the 1st National Conf. of New Development in Green Studies, Computations, Applications, and Challenges*, 10 pp. Noor, Iran, 10-11 Sept.. 2020.
- [21] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69?, pp. 46-61, Mar. 2014.
- [22] C. G. Martínez, F. J. Rodriguez, and M. Lozano, "Genetic Algorithms," in *Handbook of Heuristics*, Springer, pp. 431-464, 2018.
- [23] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of the IEEE Int. Conf. on Neural Networks*, pp. 1942-1948, Perth, Australia, 27 Nov.-1 Dec. 1995.
- [5] M. Taneja and A. Davy, "Resource-aware placement of IoT application modules in fog-cloud computing paradigm," in *Proc IFIP/IEEE Symp. on Integrated Network and Service Management*, pp. 1222-1228, Lisbon, Portugal, 8-12 May 2017.
- [6] S. Venticinque and A. Amato, "A methodology for deployment of IoT application in fog," *J. of Ambient Intelligence and Humanized Computing*, vol. 10, no. 2, pp. 1955-1976, May 2019.
- [7] A. Brogi and A. Forti, "QoS-aware deployment of IoT applications through the fog," *IEEE Internet of Things J.*, vol. 4, no. 5, pp. 1185-1192, Oct. 2017.
- [8] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Latency-aware application module management for fog computing environments," *ACM Trans. on Internet Technology*, vol. 19, no. 1, Article No.: 9, pp. 1-21, Feb. 2018.
- [9] M. Vogler, J. M. Schleicher, C. Inzinger, and S. Dustdar, "DIANE-dynamic IoT application deployment," in *Proc IEEE Int. Conf. on Mobile Services*, pp. 298-305, New York, NY, USA, 27 Jun.-2 Jul. 2015.
- [10] E. Saurez, K. Hong, D. Lillethun, U. Ramachandran, and B. Ottenwalder, "Incremental deployment and migration of geo-distributed situation awareness applications in the fog," in *Proc 10th ACM Int. Conf. on Distributed and Event-based Systems*, pp. 258-269, Irvine, CA, USA, 20-24 Jun. 2016.
- [11] U. Arora and N. Singh, "IoT application modules placement in heterogeneous fog-cloud," *International J. of Information Technology*, vol. 13, no. 9, pp. 1975-1982, May 2021.
- [12] S. Omer, S. Azizi, M. Shojafar, and R. Tafazoli, "A priority, power and traffic-aware virtual machine placement of IoT applications in cloud data centers," *J. of Systems Architecture*, vol. 15, Article ID: 101996, May 2021.
- [13] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445-487, 15 Mar. 2005.
- [14] J. P. Arcangeli, R. Boujbel, and S. Leriche, "Automatic deployment of distributed software systems: definitions and state of the art," *The J. of Systems and Software*, vol. 3pp. 198-218, May 2015.
- [15] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: a platform for Internet of Things and analytics," In: N. Bessis and C. Dobre (eds), *Big Data and Internet of Things: A Roadmap for Smart Environments. Studies in Computational Intelligence*, vol 546. Springer, Cham, pp. 169-186, 2014.
- [16] S. Farzai, M. Hosseini Shirvani, and M. Rabbani, "Multi-objective communication-aware optimization for virtual machine placement in cloud datacenters," *Sustainable Computing: Informatics and Systems*, vol. 28, Article ID: 100374, Dec. 2020.
- [17] X. S. Yang and S. Deb, "Cuckoo search via Le'vy flights," in *Proc. of World Congress on Nature & Biologically Inspired Computing*, pp. 210-214, Coimbatore, India, 9-11 Dec. 2009.
- [18] S. M. Sait, A. Bala, and A. H. El-Maleh, "Cuckoo search based resource optimization of datacenters," *Applied Intelligence*, vol. 44, no. 3, pp. 489-506, Apr. 2016.

**یاسر رمضانپور** در سال ۱۳۸۳ مدرک کارشناسی مهندسی کامپیوتر گرایش نرم‌افزار خود را از دانشگاه آزاد اسلامی واحد ساری و در سال ۱۳۸۷ مدرک کارشناسی ارشد مهندسی فناوری اطلاعات گرایش شبکه‌های کامپیوتری خود را از دانشگاه آزاد اسلامی واحد قزوین دریافت نمود. پس از آن در سال ۱۳۹۴ به دوره دکترای مهندسی کامپیوتر سیستم‌های نرم‌افزاری در دانشگاه آزاد اسلامی واحد بابل وارد گردید و در سال ۱۴۰۰ موفق به اخذ درجه دکترا در مهندسی کامپیوتر از دانشگاه مذکور گردید. ایشان از سال ۱۳۸۹ عضو هیأت علمی دانشکده فنی و مهندسی دانشگاه آزاد اسلامی واحد قائم‌شهر می‌باشد. زمینه‌های علمی مورد علاقه نام‌برده متنوع بوده و شامل موضوعاتی مانند رایانش ابری، اینترنت اشیا، مدیریت شبکه‌های کامپیوتری، امنیت شبکه و مهندسی نرم‌افزار می‌باشد.

**میرسعید حسینی شیروانی** تحصیلات خود را در مقاطع کارشناسی (مهندسی کامپیوتر گرایش نرم‌افزار)، کارشناسی ارشد (مهندسی کامپیوتر گرایش نرم‌افزار) و دکتری (مهندسی کامپیوتر با گرایش سیستم‌های نرم‌افزاری) به‌ترتیب در سال‌های ۱۳۷۹، ۱۳۸۳ و ۱۳۹۶ از دانشگاه‌های تهران به پایان رسانده است و هم‌اکنون استادیار دانشکده مهندسی کامپیوتر دانشگاه آزاد اسلامی واحد ساری می‌باشد. نام‌برده به عنوان عضو هیأت علمی تمام وقت در دانشگاه آزاد اسلامی واحد ساری مشغول به فعالیت‌های آموزشی و پژوهشی می‌باشد. در ضمن، ایشان دارای سابقه‌ی تدریس در سایر موسسات دولتی و آزاد نظیر دانشگاه صنعتی نوشیروانی بابل، دانشگاه مازندران بابل، دانشگاه‌های آزاد واحدهای بابل، آمل و قائم‌شهر می‌باشند. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: رایانش ابری، رایانش مه، اینترنت اشیا، شبکه‌های حسگر بی‌سیم، سیستم‌های توزیع شده، الگوریتم‌های موازی، محاسبات نرم هوش مصنوعی و یادگیری ماشین.