

Multi-objective IoT Application Provisioning with Optimized Delay and Cost in Fog Computing

Farid Moradi Dalvand¹, Kamran Zamanifar^{2*}

1- Pervasive Computing Lab, Department of Computer Engineering, University of Isfahan, Azadi square, Isfahan, Iran.

2*- Pervasive Computing Lab, Department of Computer Engineering, University of Isfahan, Azadi square, Isfahan, Iran.

¹f.moradi@eng.ui.ac.ir, ^{2*}zamanifar@eng.ui.ac.ir

Corresponding author address: Kamran Zamanifar, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran

Abstract- Fog computing brought a new collaborative computing model to make the growing of IoT possible. It provides the possibility to satisfy IoT applications needs by utilizing computational resources at the edge of the network. Most of the works have been done in researches, are trying to maximize or minimize one objective in different resource management problems. This approach could have some unwanted influence on other decision making aspects. This paper presents a multi-objective framework to find eligible fog nodes to dynamically deploy the IoT applications on them. The proposed framework could be employed to achieve a trade-off between the cost of resources and average service delay. The multi-objective dynamic service provisioning (MDSP) problem is formulated as an integer linear programming (ILP) model and the weighted goal programming is applied to solve the multi-objective problem. In addition, the Pareto front has been discussed and different simulations have been proposed in order to show the importance of using multi-objective solutions for service provisioning in fog computing.

Keywords- Fog computing, Internet of Things (IoT), Service provisioning, Multi-objective Optimization, Weighted goal programming, Quality of Service (QoS), Resource management.

فراهم‌سازی چندهدفه‌ی برنامه‌های کاربردی اینترنت اشیا با تأخیر و هزینه‌ی کمینه در رایانش مهی

فرید مرادی دالوند^۱، کامران زمانی فر^{۲*}

۱- آزمایشگاه رایانش فراگیر، دانشکده مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان، ایران.

۲* - آزمایشگاه رایانش فراگیر، دانشکده مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان، ایران.

^۱f.moradi@eng.ui.ac.ir, ^{۲*}zamanifar@eng.ui.ac.ir

* نشانی نویسنده مسئول: کامران زمانی فر، اصفهان، میدان آزادی، دانشگاه اصفهان، دانشکده مهندسی کامپیوتر

چکیده- اینترنت اشیا موقعیت‌های فراوانی برای برنامه‌های کاربردی تازه ظهور یافته مانند ماشین‌های خودکار و برنامه‌های کاربردی شهر هوشمند فراهم ساخته است. با ظهور برنامه‌های کاربردی اینترنت اشیا که در مقایسه با برنامه‌های کاربردی موجود نیازهای متفاوتی دارند، رایانش ابری توانایی برآورده ساختن نیازهای این برنامه‌های کاربردی نوظهور را نخواهد داشت. رایانش مهی به‌عنوان یک مدل محاسباتی به‌منظور برآورده ساختن نیازهای برنامه‌های کاربردی اینترنت اشیا معرفی شد. بی‌شتر تلاش‌های انجام شده در تحقیق‌ها، کمینه یا بیشینه کردن یک هدف در مسائل مختلف مدیریت منابع در رایانش مهی است که می‌تواند به دی‌گر خروجی‌های مهم تصمیم‌گیری آسیب بزند. در این مقاله یک چهارچوب چندهدفه برای یافتن گره‌های مناسب به‌منظور قرار دادن برنامه‌های کاربردی اینترنت اشیا پیشنهاد شده است. مسئله‌ی فراهم‌سازی پویای سرویس‌ها در رایانش مهی به‌عنوان یک مدل برنامه‌ریزی خطی عدد صحیح فرمول‌بندی شده است و روش برنامه‌ریزی وزن‌دار آرمانی برای حل این مسئله‌ی چندهدفه به‌منظور برقراری مصالحه‌های گوناگون میان هزینه‌ی منابع و تأخیر برنامه‌های کاربردی به کار گرفته شده است. علاوه بر این، با استفاده از نمودار جبهه‌ی پرتو و شبیه‌سازی‌های مختلف، اهمیت استفاده از روش‌های چندهدفه برای فراهم‌سازی سرویس‌ها در رایانش مهی نشان داده شده است.

واژه‌های کلیدی: رایانش مهی، اینترنت اشیا، فراهم‌سازی سرویس، بهینه‌سازی چندهدفه، برنامه‌ریزی وزن‌دار آرمانی، کیفیت سرویس، مدیریت منابع.

۱- مقدمه

مقیاس‌پذیری الاستیک، فناوری مجازی‌سازی و پرداخت به‌اندازه‌ی استفاده، نیازمندی‌های اولیه برنامه‌های کاربردی اینترنت اشیا را فراهم می‌کند [۱]. با این وجود، از نظر جغرافیایی متمرکز بودن مراکز داده‌ی ابر منجر به عدم توانایی آن برای برطرف ساختن نیازمندی‌های اساسی برخی برنامه‌های کاربردی اینترنت اشیا می‌شود [۲]. فاصله ارتباطی میان مکانی که داده تولید می‌شود و ابر و مقدار بسیار زیاد داده‌های تولیدشده به‌وسیله‌ی دستگاه‌های اینترنت اشیا منجر به زمان پاسخ بالا و ازدحام شبکه می‌شود.

اینترنت اشیا یک شبکه از اشیا است که توانایی پردازش، ارتباط و انجام عمل خاصی را دارند. از کاربردهای آن می‌توان به خانه‌ی هوشمند، شهر هوشمند و ماشین‌های خودکار اشاره کرد. رایانش ابری یکی از عوامل اصلی برای تحقق اینترنت اشیا بوده است. به دلیل اینکه اشیا داده‌های بسیار زیادی تولید می‌کنند، مراکز داده‌ی ابر بهترین مکان برای ذخیره‌سازی این داده‌ها بوده و همچنان هست. ابر به دلیل منابع از نظر تئوری نامحدود،

مدیریت منابع در رایانش مهی ارائه شده است. در بیشتر راه‌حل‌های ارائه‌شده، هدف، بهینه‌سازی یک تابع هدف مانند مصرف انرژی، زمان پاسخ، هزینه و استفاده از منابع لبه‌ی شبکه است. با در نظر گرفتن تنها یک هدف و اجرای راه‌حل به‌دست‌آمده با این سیاست، امکان فاصله گرفتن از جواب بهینه برای دیگر اهداف به‌طور جدی وجود دارد. برای مثال، می‌توان بدون در نظر گرفتن هزینه به‌عنوان یک تابع هدف، زمان پاسخ را در مسئله‌ی فراهم‌سازی سرویس‌ها کمینه کرد. این سیاست منجر به تحمیل هزینه‌ی بسیار زیادی به فراهم‌کننده‌ی مه می‌شود. حتی با قرار دادن هزینه به عنوان یک قید در مسئله‌ی بهینه‌سازی، نمی‌توان به راه‌حلی که هر دو عنصر تصمیم‌گیری زمان پاسخ و هزینه را کمینه کند دست یافت. در نتیجه، نیاز به راه‌حل‌های چندهدفه برای فعالیت‌های مختلف در فرآیند مدیریت منابع در رایانش مهی وجود دارد.

در این مقاله، یک مدل چندهدفه برای مسئله‌ی فراهم‌سازی سرویس‌ها در مه با هدف برقراری مصالحه میان هزینه‌ی منابع و تأخیر برنامه‌های کاربردی اینترنت اشیا پیشنهاد شده است. با استفاده از مدل پیشنهادی، تصمیم‌گیرنده برای فراهم‌سازی سرویس‌ها در مه امکان مصالحه میان هدف‌ها را با توجه به اهمیت هر کدام دارد. در نتیجه، جواب‌ها برای قرار دادن سرویس‌ها صرفاً محدود به هزینه‌ی کمینه یا تأخیر کمینه نیست.

دسته‌بندی باقیمانده‌ی مقاله به این شرح است: در بخش دو، آخرین تحقیق‌های انجام‌شده در مه بررسی می‌شوند. در بخش سه، مدل سیستم به‌صورت یک مسئله فراهم‌سازی پویای چندهدفه‌ی برنامه‌های کاربردی فرمول‌بندی می‌شود. سپس، خطی‌سازی مدل برنامه‌ریزی غیرخطی عدد صحیح^۵، به‌صورت یک مدل معادل برنامه‌ریزی عدد صحیح^۵ ارائه خواهد شد. در بخش چهارم، راه‌کار چندهدفه‌ی پیشنهادی در مقابل دو روش تک‌هدفه‌ی موجود برای کمینه کردن هزینه‌ی منابع و کمینه کردن تأخیر برنامه‌های کاربردی مورد ارزیابی قرار می‌گیرد. در بخش پنجم، نتیجه‌گیری مقاله و کارهای آینده مطرح می‌شوند.

۲- کارهای پیشین

در این بخش، راه‌کارهای مدیریت منابع در رایانش مهی و رایانش ابری شامل راه‌کارهای تک‌هدفه و چندهدفه مورد بررسی قرار می‌گیرند. هدف از مرور این تحقیق‌ها، نشان دادن اهمیت استفاده از روش‌های چندهدفه در مقابل روش‌های تک‌هدفه برای حل مسئله‌های مدیریت منابع است. اهمیت مدیریت بهینه‌ی منابع در کل و فراهم‌سازی سرویس به‌صورت خاص در رایانش ابری و مهی

تعداد بسیار زیاد دستگاه‌های اینترنت اشیا و داده‌های بسیار زیادی که توسط آن‌ها تولید می‌شود و در نتیجه بالا رفتن زمان پاسخ و استفاده‌ی بسیار زیاد از پهنای باند، نیاز به یک مدل رایانش جدید را در جامعه‌ی تحقیقاتی و صنعت به‌وجود آورد.

رایانش مهی به‌منظور غلبه بر محدودیت‌های استفاده از مراکز داده‌ی ابر، به‌عنوان یک مدل رایانش جدید برای میزبانی برنامه‌های کاربردی اینترنت اشیا معرفی شد. رایانش مهی به‌صورت توزیع‌شده سرویس‌های ابر را در لبه‌ی شبکه فراهم می‌کند. در این مدل رایانش، منابع مه و ابر برای فراهم‌سازی سرویس‌ها با یکدیگر همکاری می‌کنند [۳]. از مزایای قابل‌توجه استفاده از رایانش مهی توانایی آن برای برآورده ساختن زمان پاسخ موردنیاز برنامه‌های کاربردی اینترنت اشیا حساس به تأخیر است.

یکی از مهم‌ترین روندها در تحقیق‌های رایانش مهی، ارائه‌ی روش‌های گوناگون برای مدیریت منابع در مدل محاسباتی مبتنی بر رایانش مهی است. مسائل موردبررسی برای مدیریت منابع بسیار گوناگون هستند. برای نمونه، می‌توان به مسئله‌های پیش‌بینی منابع، فراهم‌سازی منابع^۱، فراهم‌سازی سرویس‌ها، زمان‌بندی، توزیع بار^۲ و مهاجرت سرویس^۳ها اشاره کرد.

یکی از مهم‌ترین فعالیت‌ها در فرآیند مدیریت منابع در ابر و مه، تصمیم‌گیری درباره‌ی فراهم‌سازی سرویس‌ها یا قرار دادن سرویس‌ها است. برنامه‌های کاربردی اینترنت اشیا در یک محیط بسیار پویا اجرا می‌شوند و برای مدیریت منابع به شکلی که سرویس‌ها قادر به پاسخ‌گویی به درخواست‌های کاربران باشند، استفاده از روش‌های فراهم‌سازی سرویس‌ها به شکلی که با این پویایی سیستم سازگار باشد، بسیار ضروری است. خروجی یک مسئله فراهم‌سازی سرویس‌ها تصمیم درباره‌ی چگونگی قرار دادن توزیع‌شده‌ی اجزای برنامه‌های کاربردی اینترنت اشیا در منابع ابر و مه است. در نتیجه‌ی این تصمیم‌گیری، سرویسی که درخواست بیشتری را از دستگاه‌های اینترنت اشیا تجربه می‌کند ممکن است در چندین گره‌ی مه قرار داده شود. از طرف دیگر، اگر درخواست برای یک سرویس ناچیز باشد، ممکن است برای کاهش هزینه‌ها آن سرویس از منابع مه حذف شود [۴]. از مزایای فراهم‌سازی پویای سرویس‌ها در مه می‌توان به کاهش استفاده از پهنای باند، مدیریت بهینه‌ی منابع ابر و مه، کاهش هزینه‌ی اجاره‌ی منابع، برآوردن زمان پاسخ موردنیاز برای درخواست‌های برنامه‌های کاربردی، دسترس‌پذیری، قابلیت اطمینان سرویس‌های ارائه‌شده به کاربران برنامه‌های کاربردی، کاهش انرژی و غیره اشاره کرد.

در تحقیقات، راه‌حل‌های گوناگونی برای مسئله‌های مختلف

بیشتر از آستانه نشود. این رویکرد باعث می‌شود که در بسیاری از موارد، جواب ممکن برای مسئله وجود نداشته باشد. همچنین، هزینه که مؤلفه‌ی بسیار اساسی برای تصمیم‌گیری درباره‌ی فراهم‌سازی سرویس‌ها در مه است، در این تحقیق در نظر گرفته نشده است.

تحقیق‌های دیگر مانند [۹] و [۱۰] نیز، به ترتیب زمان پاسخ و هزینه‌ی منابع را در فراهم‌سازی سرویس‌ها کمینه می‌کنند. تمامی

جدول ۱: هدف‌ها و مسئله‌های موردبررسی در تحقیق‌ها در ابر و مه

مقالات	مسئله‌ی موردبررسی														
	زمان‌بندی	معماری	انتخاب کار	فراهم‌سازی	مهاجرت	برون‌سپاری	درخواست‌ها	کارایی	تکمیل کارها	تضمین کیفیت	قابلیت اطمینان	مصرف انرژی	میزان استفاده	هزینه	تأخیر
[11]				✓											
[12]				✓				✓							
[13]				✓					✓						
[14]	✓			✓											✓
[15]				✓				✓							✓
[16]				✓				✓							
[17]	✓	✓		✓											
[18]	✓	✓					✓								
[19]				✓				✓							
[20]				✓											
[21]	✓	✓		✓			✓								
[22]	✓	✓						✓							✓
[23]															✓
[7]				✓											✓
[6]	✓							✓							
[24]	✓														✓
[4]	✓	✓													✓
[8]				✓											✓
[9]	✓														✓
[10]		✓													✓
[25]	✓														✓
[26]		✓													✓
[27]	✓														✓
[28]		✓													✓
[29]				✓				✓							✓
[30]	✓							✓							✓
[31]		✓	✓							✓					✓

تحقیق‌های ذکرشده برای بهبود یک ویژگی از کیفیت سرویس^۷ برنامه‌های کاربردی تلاش می‌کنند. در مسائلی مانند فراهم‌سازی سرویس‌ها در مه، تصمیم‌گیرنده معمولاً علاقه‌مند به بهینه‌سازی چند هدف است. این هدف‌ها در بیشتر سیستم‌ها با یکدیگر در تضاد هستند. در نتیجه، برای داشتن راه‌حل‌های مفیدتر برای مسئله‌ی فراهم‌سازی پویای سرویس، لازم است راه‌حلهایی که شامل چند تابع هدف باشند را به‌کار گرفت. مدل‌های چندهدفه در رایانش ابری [۱۱، ۱۲] و [۱۷-۱۴]، بیشتر اهدافی مانند بهینه‌سازی مصرف انرژی، کارایی، هزینه، قابلیت اطمینان و زمان پاسخ را در نظر گرفته‌اند. همچنین، در تحقیق‌های رایانش ابر متحرک نیز مدل‌های چندهدفه برای بهینه‌سازی مصرف انرژی و هزینه پیشنهاد شده است [۱۳، ۲۸]. [۲۳] یک معماری جامع برای رایانش مهی پیشنهاد داده است. [۱۸، ۳۲] برای مسئله‌ی فراهم‌سازی منابع راه‌کارهای چندهدفه پیشنهاد داده‌اند. [۲۴] مهاجرت در مه را با اهداف گوناگون بررسی کرده است. [۲۱، ۲۲] به ترتیب راه‌کارهای چندهدفه برای مسئله‌های واگذاری بار و

توسط جامعه‌ی تحقیق درک شده است. برخلاف ابر، تحقیقات در مه همچنان نابالغ است [۵]. از طرف دیگر، ساختار، ویژگی‌ها و سرویس‌های این دو مدل محاسباتی متفاوت هستند. در نتیجه، امکان استفاده از راه‌حل‌های موجود در رایانش ابری، برای مدیریت منابع در رایانش مهی وجود ندارد.

تاکنون بیشتر توجه محققان مدیریت منابع در رایانش مهی، کاهش هزینه یا تأخیر برنامه‌های کاربردی بوده است. علاوه بر این، اهدافی مانند کمینه کردن مصرف انرژی، بیشینه استفاده از منابع لبه‌ی شبکه و کمترین انتقال به ابر توسط محققان انتخاب شده‌اند.

یوسف پور و همکاران [۶] یک چهارچوب برای برنامه‌های کاربردی اینترنت اشیاء، ابر و مه و یک سیاست برون‌سپاری بار باهدف کاهش تأخیر برنامه‌های کاربردی معرفی کرده‌اند. هدف اصلی این تحقیق کاهش تأخیر پاسخ درخواست‌ها با استفاده از برون‌سپاری بار است.

اسکارات و همکاران [۷] با معرفی معماری کلونی مه، برای فراهم‌سازی سرویس‌ها نیز چند الگوریتم پیشنهاد داده‌اند که برای استفاده در این معماری است. آن‌ها کار خود را با مقایسه‌ی راه‌حل‌های بهینه‌سازی دقیق، الگوریتم ابتکاری first-fit و الگوریتم ژنتیک ارزیابی کرده‌اند. در این تحقیق، تابع هدف، بیشینه کردن استفاده از منابع لبه‌ی شبکه یا همان گره‌های مه است.

یوسف پور و همکاران [۴] یک مدل جامع از برنامه‌های کاربردی اینترنت اشیاء و منابع در رایانش مهی و ابری را پیشنهاد داده‌اند. آن‌ها یک مدل غیرخطی عدد صحیح برای حل مسئله‌ی فراهم‌سازی سرویس و دو الگوریتم ابتکاری برای برطرف کردن پیچیدگی حل این مسئله‌ی غیرخطی ارائه داده‌اند. اگرچه مدل پیشنهادی آن‌ها جامع است و هر دو الگوریتم، راه‌حل‌های کاربردی برای دو هدف متفاوت ارائه می‌دهند، زمانی که هم تأخیر سرویس‌ها و هم هزینه‌ی منابع برای تصمیم‌گیرنده اهمیت دارند، این دو الگوریتم جواب بهینه را تولید نمی‌کنند. هدف الگوریتم‌های ابتکاری تولید جواب‌های نزدیک به جواب بهینه بر اساس یک هدف است و ایده‌ی اصلی استفاده از آن‌ها حذف قیدها یا اهداف دیگر برای کاهش پیچیدگی است. در نتیجه، با توجه به شرایط سیستم، تصمیم‌گیرنده باید به نفع پیچیدگی یا جواب بهینه مصالحه‌ای برقرار کند.

در تحقیقی دیگر، مینه و همکاران [۸] راه‌کاری برای مسئله‌ی فراهم‌سازی سرویس در مه با هدف بیشینه کردن استفاده از منابع ارائه داده‌اند. تأخیر برنامه‌های کاربردی تنها به‌صورت یک قید در مسئله تعریف شده است و باعث می‌شود که زمان پاسخ هیچ‌گاه

بدون مدل‌سازی رها شده و نتایج این نوع تصمیم‌گیری‌ها ارائه نمی‌شود. این موضوع باعث می‌شود راه‌کارهای پیشنهادی برای استفاده در دنیای واقعی به‌طور مؤثر کارا نباشند. به دلیل اینکه بسیاری از عوامل مهم تصمیم‌گیری برای فراهم‌سازی سرویس‌ها در رایانش مهی با یکدیگر در تضاد هستند، راه‌کار پیشنهادی باید همه‌ی این عوامل مهم در تصمیم‌گیری را در نظر داشته باشد. برای مثال، راهکاری که تنها هزینه را کمینه کند، ممکن است باعث تأخیر بسیار زیاد برنامه‌های کاربردی شود که برای اینترنت اشیا مفید نخواهد بود. از طرف دیگر، اگر یک روش تنها تأخیر برنامه‌های کاربردی را کمینه کند، این موضوع می‌تواند منجر به سربرار هزینه‌ای بسیار بالایی شود که مورد قبول فراهم‌کننده‌ی مه نخواهد بود. در نتیجه، راهکاری که بتواند به‌صورت چندهدفه میان این اهداف متضاد مصالحه‌ای برقرار کند مورد توجه ذی‌نفعان است. با استفاده از این روش‌ها، فراهم‌کننده‌ی مه این امکان را خواهد داشت که در حالت‌های مختلف سیستم و با توجه به اهمیت هر هدف در این شرایط، مصالحه‌ی موردنظر خود را برقرار کند.

تنها روش ارائه‌شده برای مسئله‌ی فراهم‌سازی چندهدفه‌ی سرویس‌ها در مه [۳۱]، اهداف تضمین کیفیت سرویس، منابع استفاده‌شده لبه‌ی شبکه و هزینه‌ی استقرار برنامه‌های کاربردی را در نظر گرفته است. با این وجود، در مقایسه با منابع مصرفی و پیوند ارتباطی که در نظر گرفته شده است، زمان پاسخ و دسترس‌پذیری^۱ و هزینه در یک محیط رایانش مهی از اهمیت بسیار بالاتری برخوردار هستند [۵]. از آنجاکه مهم‌ترین عوامل تصمیم‌گیری در مدیریت منابع در رایانش مهی، فراهم کردن زمان پاسخ بهینه و انجام این کار با کمترین هزینه‌ی ممکن است، در این تحقیق تأخیر برنامه‌های کاربردی و هزینه‌ی منابع به‌عنوان اهداف مسئله‌ی بهینه‌سازی چندهدفه انتخاب شده‌اند.

دستگاه‌های اینترنت اشیا در یک منطقه، درخواست‌های خود را به گره‌های مه موجود در آن منطقه که سرویس موردنظر در آن گره‌ها قرار داده شده می‌فرستند. برای هر منطقه یک سرور ابر نیز در نظر گرفته شده که در صورتی که گره‌های مه نتوانند به درخواست‌ها سرویس ارائه کنند، ابر این سرویس را برای آن‌ها فراهم کند. طول جغرافیایی منطقه و تعداد گره‌های مه و فاصله‌ی آن‌ها با دستگاه‌های اینترنت اشیا در مرحله‌ی برنامه‌ریزی منابع تعیین می‌شود. این اطلاعات برای مسئله‌ی پویای فراهم‌سازی چندهدفه‌ی سرویس‌ها در رایانش مهی به‌عنوان چند ورودی هستند. راه‌حل‌های زمان اجرا برای مسئله‌های مختلف مدیریت منابع در یک واحد منطقی به نام کنترل‌کننده‌ی لبه اجرا می‌شوند

زمانبندی پیشنهاد داده‌اند. [۱۹] یک معماری بر اساس containerها پیشنهاد داده است و مسئله‌ی انتخاب کارها و مهاجرت به صورت چندهدفه در این تحقیق پاسخ داده شده است. همچنین، [۲۷-۲۵] برای مسئله‌ی فراهم‌سازی سرویس‌ها مدل‌های تک‌هدفه پیشنهاد داده‌اند. جدول ۱ خلاصه‌ای از تحقیق‌ها در رایانش ابری، رایانش مهی و رایانش‌های شبیه به رایانش مهی را نشان می‌دهد. فراهم‌سازی چندهدفه‌ی سرویس‌ها در مه توجه لازم را در جامعه‌ی تحقیقاتی کسب نکرده است. بیشتر مدل‌های چندهدفه مانند مدل ارائه‌شده در [۲۹] روی مسئله‌ی زمان‌بندی کارها با اهداف کمینه کردن زمان کل و مصرف انرژی تمرکز کرده‌اند و مانند [۳۰] برای حل مسئله‌ی مهاجرت ماشین مجازی با اهداف بیشینه کردن درخواست‌های پذیرفته‌شده و کمینه کردن تأخیر احساس شده برای کاربران راه‌حلی پیشنهاد داده‌اند.

تا جایی که نویسنده آگاه است، کار جدید ارائه‌شده در [۳۱] تنها تحقیقی است که برای حل مسئله‌ی فراهم‌سازی چندهدفه‌ی سرویس‌ها راهکاری ارائه داده است. نویسندگان به مصالحه‌ی میان تضمین کیفیت سرویس، منابع استفاده‌شده‌ی مه و هزینه‌ی استقرار برنامه‌های کاربردی دست یافته‌اند. با این وجود، در مقایسه با منابع مصرفی و پیوند ارتباطی که در نظر گرفته شده است، زمان پاسخ و دسترس‌پذیری و هزینه در یک محیط رایانش مهی از اهمیت بسیار بالاتری برخوردار هستند [۵].

۳- فراهم‌سازی چندهدفه‌ی برنامه‌های کاربردی اینترنت اشیا

در رایانش ابری یکی از مهم‌ترین توابع هدف استفاده‌شده هزینه‌ی منابع است. به دلیل بالغ بودن تحقیق‌ها در رایانش ابری، راه‌کارهای چندهدفه و کاربردی بسیاری برای مسئله‌ی فراهم‌سازی سرویس‌ها پیشنهاد شده است. با این وجود، تابع هدفی مانند تأخیر برنامه‌های کاربردی، تنها در چند مورد خاص که برای برنامه‌های کاربردی خاص پیشنهاد شده‌اند استفاده شده است. با توجه به نیازهای برنامه‌های کاربردی اینترنت اشیا و مهم‌ترین هدف رایانش مهی برای محقق شدن آن‌ها، تأخیر برنامه‌های کاربردی مهم‌ترین هدف برای بهینه شدن است.

راه‌کارهای مدیریت منابع ارائه‌شده در رایانش مهی تلاش بر کاهش تأخیر درخواست‌ها برای برنامه‌های کاربردی اینترنت اشیا کرده‌اند. زمانی که هدف نهایی، کمینه کردن تأخیر برنامه‌های کاربردی باشد، دیگر خروجی‌های مسئله‌ی فراهم‌سازی سرویس‌ها

$$C^C = \sum_{i \in A} \sum_{j \in F} L(R, j) \delta_{aj} (\alpha_i + \beta_i) \tau, \quad (4)$$

$$C^S = \sum_{i \in A} \sum_{j \in F} \theta_i S^F x_i^j \tau, \quad (5)$$

جدول ۲: نشانه‌گذاری

مجموعه‌ی سرویس‌ها	A	برنامه‌ی کاربردی
مجموعه‌ی دستگاه‌های درخواست‌کننده‌ی سرویس i	E_i	
کیفیت سرویس i بیان‌شده در SLA	Δ_i	
فرجه‌ی سرویس i	D_i	
واحد هزینه‌ی جریمه برای راضی نکردن Δ_i	C_i	
نیازمندی حافظه‌ی سرویس i	M_i	
نیازمندی حافظه‌ی کمکی سرویس i	θ_i	
نیازمندی پردازنده‌ی سرویس i	U_i	
میانگین اندازه‌ی درخواست برای سرویس i	α_i	
میانگین اندازه‌ی پاسخ برای سرویس i	β_i	
درصد تخطی از آستانه برای سرویس i	V_i	
زمان فعلی	t	زمان
زمان میان دو اجرای MDSP	τ	
واحد هزینه ارتباطی بر واحد هزینه‌ی پهنای باند از منبع i به منبع j	L(i,j)	زیرساخت
پهنای باند ارتباط i به j	B(i,j)	
تأخیر انتشار ارتباط i به j	G(i,j)	
گره‌ی ابر	R	منابع
واحد هزینه‌ی پردازشی در ابر	P^R	
واحد هزینه‌ی حافظه‌ی کمکی در ابر	S^R	
نرخ درخواست‌های سرویس i به ابر	λ_i^R	
تأخیر صف به‌اضافه‌ی تأخیر اجرای سرویس i در ابر	W_i^R	
ظرفیت پردازنده در ابر	μ^R	
ظرفیت حافظه‌ی کمکی در ابر	θ^R	
ظرفیت حافظه در ابر	M^R	
نرخ ورود به ابر	Π_R	
مجموعه‌ی گره‌های مه	F	
کنترل‌کننده‌ی لبه	Ψ	
تأخیر صف و اجرای سرویس i در مه j	W_{ij}^F	
واحد هزینه‌ی پردازشی در گره‌ی مه j	P^F	
واحد هزینه‌ی حافظه‌ی کمکی در گره‌ی مه j	S^F	
نرخ درخواست‌های انتقال داده شده برای سرویس i از گره‌ی مه j به ابر	δ_{ij}	
نرخ درخواست سرویس i به گره‌ی مه j	λ_{ij}^F	
ظرفیت حافظه در گره‌ی مه j	M_j^F	
ظرفیت حافظه‌ی کمکی در گره‌ی مه j	θ_j^F	
نرخ ورود گره‌ی مه j	Π_j	
ظرفیت پردازنده در گره‌ی مه j	μ_j^F	

C^D ، هزینه‌ی استقرار container است. برای استقرار containerها، ابتدا باید تصویر container را از پایگاه داده‌ای که همه‌ی آن‌ها را ذخیره می‌کند بارگذاری کرد و سپس روی هر گره‌ی مه، آن‌ها را آماده‌ی اجرا کرد. x_{ij}^F یک متغیر وابسته است که نشان می‌دهد آیا

و تصمیم‌گیری در این واحد منطقی اتخاذ می‌شود. گره‌های مه به‌صورت دوره‌ای اطلاعات و حالت خود را به این واحد منطقی ارسال می‌کنند تا این اطلاعات برای تصمیم‌گیری‌های مختلف مانند فراهم‌سازی سرویس‌ها مورد استفاده قرار گیرد. برای مدل‌سازی پایه‌ی این مقاله، از اصلاح‌شده‌ی مدل غیرخطی ارائه‌شده در [۴] استفاده شده است.

در جدول ۲ نشانه‌گذاری برنامه‌های کاربردی اینترنت اشیاء، منابع موجود ابر و مه و اطلاعات مربوط به حل مسئله‌ی فراهم‌سازی سرویس‌ها که در این مقاله استفاده شده، نشان داده شده است. x_i^j یک متغیر تصمیم دودویی است که نشان‌دهنده‌ی قرار گرفتن سرویس i روی گره‌ی مه j است. اگر مقدار این متغیر صفر باشد به این معناست که سرویس i نباید روی گره‌ی مه j قرار گیرد و اگر این مقدار یک باشد یعنی این سرویس باید روی گره‌ی مه j استقرار یابد. این متغیر تصمیم، متغیر تصمیم استقرار نام دارد.

۳-۱- تابع هدف

رابطه‌ی (۱) تأخیر برنامه‌های کاربردی را نشان می‌دهد [۴]. تأخیر برنامه‌های کاربردی از سه قسمت جدا تشکیل شده است. تأخیر انتشار، تأخیر ارسال و میانگین زمان انتظار این سه جزء از تأخیر را برای هر سرویس روی هر گره‌ی مه تعیین می‌کنند.

$$T_i^j = (2G(E_i, j) + \frac{\alpha_i + \beta_i}{B(E_i, j)} + W_{ij}^F) x_i^j + (2(G(E_i, j) + G(j, R)) + W_i^R) + (\frac{\alpha_i + \beta_i}{B(E_i, j)} + \frac{\alpha_i + \beta_i}{B(j, R)})(1 - x_i^j), \quad \forall i, \forall j, \quad (1)$$

رابطه‌ی (۲) مجموع تأخیرهای برنامه‌های کاربردی قرار داده شده در گره‌های مه و ابر را محاسبه می‌کند که به‌عنوان یکی از توابع هدف در نظر گرفته می‌شود. برای مدل‌سازی هزینه، چندین زیر هزینه تعریف شده است. هزینه‌ی کل در رابطه‌ی (۳) از هزینه‌های مختلفی مانند هزینه‌ی ارتباطات، هزینه‌ی ذخیره‌سازی، هزینه‌ی استقرار container، هزینه‌ی تخطی^۱ و هزینه‌ی پردازش تشکیل شده است. رابطه‌های (۴) تا (۸)، به ترتیب هر کدام از این هزینه‌ها را محاسبه می‌کنند. هزینه‌های ذخیره‌سازی، استقرار و تخطی در رابطه‌های (۵) تا (۷) نشان داده شده‌اند [۴]. هزینه‌ی کل، به‌عنوان هدفی دیگر در کنار مجموع تأخیر برنامه‌های کاربردی در نظر گرفته شده است.

$$totalDelay = \sum_{i \in A} \sum_{j \in F} T_i^j, \quad (2)$$

$$totalCost = C^C + C^S + C^D + C^V + C^P, \quad (3)$$

کاربردی در یک گرهی مه از آستانه بیشتر باشد، تخطی اتفاق افتاده و متغیر تخطی باید مقدار یک داشته باشد [۴].

$$v_i^j = \begin{cases} 1, & \text{if } T_i^j \geq D_i \\ 0, & \text{otherwise} \end{cases}, \quad \forall i, \forall j, \quad (14)$$

پس از محاسبه‌ی تمامی v_i^j ها، می‌توان میزان تخطی از آستانه را برای هر سرویس به صورت جداگانه به صورت درصد نشان داد. رابطه‌ی (۱۵) نسبت تعداد درخواست‌های دستگاه‌های اینترنت اشیا برای یک سرویس که زمان پاسخ آن‌ها از میزان آستانه بیشتر است، به تعداد کل درخواست‌های آن سرویس را نشان می‌دهد [۴]. با استفاده از رابطه‌ی (۱۵)، می‌توان هزینه‌ی تخطی از آستانه در رابطه‌ی (۷) را محاسبه کرد.

$$V_i = \frac{\sum_{j \in F} \lambda_{ij}^F v_i^j}{\sum_{j \in F} \lambda_{ij}^F}, \quad \forall i, \quad (15)$$

۳-۳- منابع استفاده شده در مه

رابطه‌ی (۱۶) نیازمندی‌های حافظه‌ای و ذخیره‌سازی سرویس‌های قرار داده شده در هر گرهی مه را با ظرفیت حافظه با دستیابی تصادفی و حافظه‌ی کمکی آن گره مقایسه می‌کند.

$$\sum_{i \in A} x_i^j N_i \leq N_j^F, \quad N \in \{\theta, M\}, \forall j, \quad (16)$$

۳-۴- نرخ ورود و ظرفیت پردازشی گره‌های مه و ابر

رابطه‌ی (۱۷) تعیین می‌کند که در صورت انتخاب جواب بهینه برای متغیرهای تصمیم قرار دادن سرویس‌ها در مه، نرخ ورود درخواست‌ها برای هر گرهی مه چقدر است [۴].

$$\Pi_j = \sum_{i \in A} U_i \lambda_{ij}^F x_i^j, \quad \forall j, \quad (17)$$

پس از محاسبه‌ی نرخ ورود برای هر گرهی مه، می‌توان با توجه به قرار گرفتن سرویس‌ها در مه یا قرار نگرفتن آن‌ها، نرخ خروج درخواست‌ها برای هر سرویس از هر گرهی مه موجود در لبه‌ی شبکه را با استفاده از رابطه‌ی (۱۸) محاسبه کرد [۴].

$$\delta_{ij} = \lambda_{ij}^F (1 - x_i^j), \quad \forall i, \forall j, \quad (18)$$

با استفاده از رابطه‌ی (۱۹)، نرخ ترافیک ورودی درخواست‌های هر سرویس به ابر نیز محاسبه می‌شود. نرخ ورود درخواست‌ها برای یک سرویس به ابر در رابطه‌ی (۲۰) محاسبه می‌شود.

$$\lambda_i^R = \sum_{j \in F} \delta_{ij}, \quad \forall i, \quad (19)$$

سرویس i با توجه به تصمیم‌گیری در دوره‌ی قبل، روی گرهی مه j قرار داده شده است یا خیر.

$$C^D = \sum_{i \in A} \sum_{j \in F} \theta_i L(\psi, j) (1 - x_{ij}^i) x_{ij}^j, \quad (6)$$

در صورتی که تخطی از ویژگی‌های بیان‌شده در کیفیت سرویس بیشتر باشد، فراهم‌کننده‌ی مه باید هزینه‌ای را به صورت جریمه پرداخت کند.

$$C^V = \sum_{i \in A} \max(0, V_i - (1 - \Delta_i)) C_i \tau, \quad (7)$$

از آنجاکه تمام سرویس‌ها در ابر قرار گرفته‌اند، هزینه‌ی پردازش برای همه‌ی آن‌ها محاسبه می‌شود. درحالی‌که، برای محاسبه‌ی هزینه‌ی پردازش در مه، تنها گره‌های قرار داده شده در نظر گرفته می‌شوند.

$$C^P = \sum_{i \in A} (\sum_{j \in F} p^F U_i \lambda_{ij}^F x_i^j \tau + p^R U_i \lambda_i^R \tau). \quad (8)$$

با ارائه‌ی مدل مجموع تأخیر برنامه‌های کاربردی و هزینه‌ی کل که در رابطه‌های (۱) تا (۸) نمایش داده شده‌اند، می‌توان تابع هدف فراهم‌سازی پویای چندهدفه‌ی سرویس‌ها در مه را با استفاده از روش برنامه‌ریزی وزن‌دار آرمانی^{۱۱} [۳۳] محاسبه کرد. با کمینه کردن p_{cost} و p_{delay} می‌توان تأخیر برنامه‌های کاربردی و هزینه‌ی منابع را تا حد امکان به مقادیر آرمانی هرکدام از هدف‌ها در رابطه‌های ۱۰ تا ۱۳ نزدیک کرد. همچنین، با قرار دادن وزن‌های مختلف برای نشان دادن اهمیت هر تابع هدف در رابطه‌ی (۹)، مصالحه‌های گوناگون فراهم می‌شود.

تابع‌های هدف دیگر را می‌توان به مدل ارائه‌شده اضافه کرد. با تعریف یک وزن جدید برای هدف مورد نظر و اضافه کردن قیدهای مربوطه، می‌توان مدل پیشنهادی را به شکلی گسترش داد که شامل دیگر اهداف نیز باشد. این موضوع تا زمانی که مجموع وزن‌های هدف‌ها برابر با یک باشد صادق است.

$$\text{Minimize } \left(\frac{w_{delay} p_{delay}}{k_{delay}} + \frac{w_{cost} p_{cost}}{k_{cost}} \right) \quad (9)$$

Subject to

$$f_{delay} - p_{delay} = b_{delay}, \quad (10)$$

$$f_{cost} - p_{cost} = b_{cost}, \quad (11)$$

$$f_{cost} = totalCost, \quad (12)$$

$$f_{delay} = totalDelay. \quad (13)$$

۳-۲- محاسبه‌ی تخطی از آستانه

v_i^j یک متغیر دودویی وابسته است که متغیر تخطی نام دارد. رابطه‌ی (۱۴) نشان می‌دهد در صورتی که زمان تأخیر یک برنامه‌ی

غیرخطی عدد صحیح در بخش قبل است و بدون حذف قید حاصل شود.

۳-۶-۱- خطی سازی تابع بیشینه

تابع بیشینه میان دو مقدار، یک تابع غیرخطی است. در رابطه‌ی (۷)، در صورتی که نسبت تخطی محاسبه‌شده در رابطه‌ی (۱۵) از میزان اجازه داده‌شده برای تخطی طبق کیفیت سرویس بیشتر باشد، این مقدار در رابطه‌ی (۷) قرار می‌گیرد و در غیر این صورت عدد صفر در مقابل واحد هزینه قرار می‌گیرد. برای خطی سازی این رابطه، بجای کل تابع بیشینه در رابطه‌ی (۷)، یک متغیر جدید با نام M_i قرار می‌گیرد. شکل بازنویسی شده‌ی رابطه‌ی (۷) در رابطه‌ی (۲۵) نشان داده شده است. با اضافه کردن دو قید (۲۶) و (۲۷) به مدل ارائه‌شده می‌توان فرآیندی را به وجود آورد که مقدار تابع بیشینه با مقدار M_i برابر باشد. اگر نسبت تخطی V_i از مقدار تخطی مجاز بر اساس کیفیت سرویس $1 - \Delta_i$ کمتر باشد، M_i بزرگ‌تر یا مساوی یک عدد منفی می‌شود و در رابطه‌ی (۲۷) نیز بزرگ‌تر یا مساوی صفر می‌شود. هزینه‌ی کل دارای پنج زیر هزینه است و هرکدام از آن‌ها عدد مثبت هستند و در تابع هدف، مقدار هزینه‌ی کل باید کمینه شود. در نتیجه، M_i برابر با کمترین مقدار ممکن یعنی صفر می‌شود. این عملکرد برای حالتی که نسبت تخطی V_i از مقدار تخطی مجاز بر اساس کیفیت سرویس $1 - \Delta_i$ بیشتر باشد هم برقرار است.

$$C^V = \sum_{i \in A} M_i C_i \tau, \quad (25)$$

$$M_i \geq V_i - (1 - \Delta_i), \quad \forall i, \quad (26)$$

$$M_i \geq 0, \quad \forall i. \quad (27)$$

۳-۶-۲- خطی سازی تابع انتخاب دودویی

در رابطه‌ی (۱۴) برای تعیین تخطی از آستانه برای هر سرویس روی هر گره‌ی مه از متغیر وابسته‌ی دودویی v_i^j استفاده شده است. این نوع مقایسه برای انتخاب یک متغیر دودویی یک رابطه‌ی غیرخطی است. به این دلیل که می‌توان آن را به شکل یک رابطه با عنصر قدر مطلق نمایش داد که یک عملگر غیرخطی است. با استفاده از دو رابطه‌ی خطی (۲۸) و (۲۹) و جایگزینی آن‌ها با رابطه‌ی (۱۴) می‌توان به معادل خطی این رابطه دست یافت. اگر تأخیر درخواست برای برنامه‌ی کاربردی i که در گره‌ی مه z استقرار یافته از آستانه‌ی برنامه‌ی کاربردی i بیشتر باشد، v_i^j در رابطه‌ی (۲۸) کوچک‌تر یا مساوی با عددی بزرگ‌تر از یک می‌شود. همچنین، v_i^j در رابطه‌ی (۲۹) نیز بزرگ‌تر یا مساوی با عددی بزرگ‌تر از صفر می‌شود. از آنجاکه v_i^j یک متغیر دودویی است و

$$\Pi_R = \sum_{i \in A} U_i \lambda_i^R, \quad (20)$$

در رابطه‌های (۲۱) و (۲۲)، نرخ ورود درخواست‌ها برای هر گره‌ی مه یا ابر نباید از ظرفیت پردازشی این منابع محاسباتی بیشتر باشد. [۴]

$$\Pi_j \leq \mu_j^F, \quad \forall j, \quad (21)$$

$$\Pi_R \leq \mu^R, \quad (22)$$

۳-۵- نظریه‌ی صف و میانگین زمان اجرا در ابر و مه

ویژگی‌های سیستم‌های گوناگون مانند تعداد افراد موجود در سیستم، میانگین زمان انتظار، میانگین زمان بودن در صف، میانگین زمان پردازش درخواست‌ها و پایداری سیستم که برای تصمیم‌گیری بسیار اساسی هستند، با استفاده از رابطه‌ها و مدل‌های گوناگون نظریه‌ی صف، قابل‌محاسبه هستند [۳۴]. مدل صف M/M/1 به‌عنوان ساده‌ترین و معروف‌ترین مدل صف در تحقیق‌های گوناگون مورد استفاده قرار گرفته است. با استفاده از رابطه‌های این مدل صف، می‌توان ویژگی‌های بسیار مهمی مانند توزیع احتمال زمان انتظار که برابر با زمان صف به‌اضافه‌ی زمان پردازش است را محاسبه کرد. رابطه‌های (۲۳) و (۲۴)، به ترتیب میانگین زمان انتظار برای گره‌های مه و ابر را محاسبه می‌کنند. [۴]

$$W_{i,j}^F = \frac{1}{\mu_j^F - \Pi_j}, \quad \forall i, \forall j, \quad (23)$$

$$W_i^R = \frac{1}{\mu^R - \Pi_R}, \quad \forall i. \quad (24)$$

نرخ سرویس باید همیشه بزرگ‌تر از نرخ درخواست باشد تا طول صف بی‌نهایت نشود. رابطه‌های (۲۱) و (۲۲) این ویژگی که پایداری صف است را تضمین می‌کنند.

۳-۶-۳- مدل برنامه‌ریزی خطی عدد صحیح برای مسئله‌ی

فراهم‌سازی سرویس در مه

در میان رابطه‌های بخش قبل، رابطه‌های (۱)، (۷)، (۱۴)، (۲۳) و (۲۴) باعث می‌شوند که این مدل غیرخطی باشد. مدل‌های غیرخطی، دارای پیچیدگی محاسباتی بسیار بالایی هستند و از طرف دیگر، حل‌کننده‌های مسائل بهینه‌سازی برای حل این‌گونه مسائل دارای محدودیت‌های بیشتری نسبت به مدل‌های خطی هستند. در این بخش، با استفاده از روش‌های مختلف، رابطه‌های غیرخطی موجود در بخش قبل، خطی می‌شوند تا یک مدل برنامه‌ریزی خطی عدد صحیح که معادل مدل برنامه‌ریزی

در رابطه‌های (۳۲) و (۳۳)، ε و ε' به ترتیب برابر با کمترین و بیشترین مقادیری هستند که W_{ij}^F می‌تواند داشته باشد. اگر x_i^j برابر با صفر باشد، در رابطه‌ی (۳۲)، z_i^j برابر با صفر می‌شود. در رابطه‌ی (۳۳)، z_i^j بین یک عدد منفی و یک عدد مثبت قرار می‌گیرد که در نتیجه‌ی این دو قید، مقدار نهایی z_i^j برابر با صفر است. در صورتی که x_i^j برابر با یک باشد، در رابطه‌ی (۳۲)، z_i^j میان کمترین و بیشترین مقدار W_{ij}^F قرار می‌گیرد. در رابطه‌ی (۳۳) با توجه به یک بودن x_i^j ، مقدار z_i^j کوچک‌تر مساوی و بزرگ‌تر مساوی W_{ij}^F می‌شود و در نتیجه این متغیر کمکی برابر با W_{ij}^F می‌شود. وجود این دو قید نیز، در صورتی که x_i^j برابر با یک باشد، باعث می‌شود که z_i^j معادل $x_i^j W_{ij}^F$ باشد.

مراحل خطی سازی رابطه‌ی (۲۴) که در رابطه‌های (۳۴) تا (۳۷) نشان داده شده، مانند مراحل خطی سازی رابطه‌ی (۲۳) است با این تفاوت که باید بجای x_i^j از $(1 - x_i^j)$ استفاده کرد.

$$W_{ij}^R \mu^R - U_i \lambda_{ij}^F W_{ij}^R (1 - x_i^j) = 1, \quad \forall i, \forall j. \quad (34)$$

$$W_{ij}^R = \frac{1 + U_i \lambda_{ij}^F z_{ij}^R}{\mu^R}, \quad \forall i, \forall j. \quad (35)$$

$$(1 - x_i^j) \pi \leq z_{ij}^R \leq (1 - x_i^j) \pi', \quad \forall i, \forall j, \quad (36)$$

$$W_{ij}^R - \pi' x_i^j \leq z_{ij}^R \leq W_{ij}^R - \pi x_i^j, \quad \forall i, \forall j, \quad (37)$$

در رابطه‌های (۳۶) و (۳۷)، π و π' به ترتیب برابر با کمترین و بیشترین مقادیری هستند که W_{ij}^R می‌تواند داشته باشد. با جایگذاری z_i^j به ترتیب بجای $x_i^j W_{ij}^F$ و z_{ij}^R به $W_{ij}^R (1 - x_i^j)$ در رابطه‌ی (۱) نیز، رابطه‌ی خطی معادل آن در رابطه‌ی (۳۸) به دست می‌آید.

با توجه به رابطه‌های خطی بیان شده در این بخش، رابطه‌ی (۳۹) شکل نهایی مسئله‌ی فراهم‌سازی پویای چندهدفه‌ی سرویس در مه است.

$$T_i^j = (2G(E_i, j) + \frac{\alpha_i + \beta_i}{B(E_i, j)}) x_i^j + z_i^j + (2(G(E_i, j) + G(j, R))) \quad (38)$$

$$+ (\frac{\alpha_i + \beta_i}{B(E_i, j)} + \frac{\alpha_i + \beta_i}{B(j, R)})(1 - x_i^j) + z_{ij}^R, \quad \forall i, \forall j,$$

$$\text{Minimize } (\frac{W_{delay} P_{delay}}{k_{delay}} + \frac{W_{cost} P_{cost}}{k_{cost}}) \quad (39)$$

Subject to
(2) to (6), (8), (10) to (13), (15) to (22), (25) to (29), (31) to (33) and (35) to (37).

۴- ارزیابی

در این بخش، محیط شبیه‌سازی و ارزیابی راه‌کار پیشنهادی در مقابل دو راه‌کار تک‌هدفه برای فراهم‌سازی سرویس‌ها بررسی می‌شوند.

تنها می‌تواند مقادیر صفر یا یک داشته باشد، تنها مقدار مجاز برای این متغیر یک است. H باید عدد به‌اندازه‌ی کافی بزرگی باشد که باعث شود مقدار به‌دست‌آمده برای $T_i^j - D_i$ همواره با استفاده از H به بازه‌ی منفی یک تا یک نگاشت شود. مقادیری که می‌توان برای H تعریف کرد $H \geq \text{Max}(T_i^j) + \text{Max}(D_i)$ است.

$$v_i^j \leq \frac{T_i^j}{D_i}, \quad \forall i, \forall j, \quad (28)$$

$$v_i^j \geq \frac{T_i^j - D_i}{H}, \quad \forall i, \forall j. \quad (29)$$

۳-۶-۳- خطی سازی زمان پاسخ و زمان انتظار

در رابطه‌ی (۱) برای به دست آوردن تأخیر برنامه‌های کاربردی در ابر و گره‌های مه، دو متغیر x_i^j و W_{ij}^F و دو متغیر x_i^j و W_{ij}^R در هم ضرب می‌شوند که باعث می‌شود این رابطه غیرخطی باشد. چراکه در مرحله‌های محاسبه‌ی W_{ij}^F و W_{ij}^R از x_i^j استفاده می‌شود. از طرفی، به خاطر وجود نرخ ورود درخواست‌ها به ابر و گره‌های مه در رابطه‌های (۲۳) و (۲۴) در مخرج کسر، هر دو رابطه نیز غیرخطی هستند. برای تبدیل رابطه‌های (۱)، (۲۳) و (۲۴) به معادل خطی آن‌ها، نیاز به تعریف دو متغیر کمکی z_{ij}^R و z_i^j است که به ترتیب حاصل ضرب متغیرهای x_i^j و W_{ij}^F و $(1 - x_i^j)$ و W_{ij}^R هستند. در اولین مرحله از فرایند خطی سازی برای رابطه‌ی (۲۳)، سمت چپ تساوی با مخرج کسر در سمت راست تساوی ضرب می‌شود. سپس، رابطه‌ی (۱۷) بجای Π_j در رابطه‌ی (۲۳) قرار می‌گیرد تا رابطه‌ی (۳۰) حاصل شود. در مرحله‌ی بعد، بجای $x_i^j W_{ij}^F$ متغیر کمکی z_i^j قرار داده می‌شود. شکل خطی رابطه‌ی (۲۳) در رابطه‌ی (۳۱) نشان داده شده است.

$$W_{ij}^F \mu_j^F - U_i \lambda_{ij}^F x_i^j W_{ij}^F = 1, \quad \forall i, \forall j. \quad (30)$$

$$W_{ij}^F = \frac{1 + U_i \lambda_{ij}^F z_i^j}{\mu_j^F}, \quad \forall i, \forall j. \quad (31)$$

همچنین، بجای $x_i^j W_{ij}^F$ در رابطه‌ی (۱)، z_i^j قرار داده می‌شود. برای اینکه متغیر پیوسته‌ی کمکی جدید که حاصل ضرب یک متغیر دودویی و یک متغیر پیوسته است، معادل $x_i^j W_{ij}^F$ باشد، باید دو رابطه‌ی جدید (۳۲) و (۳۳) را به مدل اضافه کرد. یک متغیر دودویی و یک متغیر پیوسته است. در صورتی که مقدار x_i^j برابر با صفر باشد، z_i^j برابر با صفر می‌شود و غیر این صورت، متغیر وابسته‌ی z_i^j برابر با مقدار متغیر پیوسته‌ی W_{ij}^F می‌شود. دو رابطه‌ی (۳۲) و (۳۳) به شکلی عمل می‌کنند که z_i^j و $x_i^j W_{ij}^F$ معادل هم باشند.

$$x_i^j \varepsilon \leq z_i^j \leq x_i^j \varepsilon', \quad \forall i, \forall j, \quad (32)$$

$$W_{ij}^F - \varepsilon'(1 - x_i^j) \leq z_i^j \leq W_{ij}^F - \varepsilon(1 - x_i^j), \quad \forall i, \forall j, \quad (33)$$

۴-۱- محیط شبیه‌سازی

به‌منظور ارزیابی راه‌حل پیشنهادی، یک محیط همکاری ابر و مه برای یک برنامه‌ی اینترنت اشیاء نمونه در نظر گرفته شده است. در این محیط، منطقه‌هایی شامل چند گره‌ی مه و یک ابر برای پوشش تمام منطقه انتخاب شده است. پارامترهای شبیه‌سازی در نرم‌افزار GAMS [۳۵] پیاده‌سازی شده‌اند و راه‌کار فراهم‌سازی پویای چندهدفه‌ی سرویس‌ها و دو راه‌کار تک‌هدفه‌ی مرتبط، با استفاده از این نرم‌افزار پیاده‌سازی شده‌اند. از حل‌کننده‌ی مسائل بهینه‌سازی IBM ILOG CPLEX [۳۶] در ارتباط با نرم‌افزار GAMS برای حل راه‌کارهای گوناگون استفاده شده است. شبیه‌سازی در سیستم‌عامل ویندوز و با استفاده از پردازنده‌ی Intel i5-2420M 2.3GHz انجام شده است.

در شبکه‌ی موردبررسی، در یک منطقه تعدادی دستگاه اینترنت اشیاء به گره‌های مه موجود در منطقه، درخواست برای سرویس‌دهی می‌فرستند. این سرویس‌ها در صورتی که به‌عنوان خروجی مسئله‌ی فراهم‌سازی سرویس‌ها، روی گره‌های مه قرار گرفته باشند، به درخواست‌ها پاسخ می‌دهند. در غیر این صورت، درخواست‌ها برای پاسخ‌دهی، به ابر که برای منطقه‌ی موردنظر پیش‌بینی شده است فرستاده می‌شوند. در این تحقیق، تمامی سرویس‌ها برای بالا بردن قابلیت اطمینان به‌صورت پیش‌فرض در ابر قرار داده شده‌اند. برای ارزیابی راه‌حل ارائه‌شده در این مقاله، ۱۰ گره‌ی مه در یک منطقه با ابر در نظر گرفته شده در آن منطقه ارتباط برقرار می‌کنند. مسئله‌ی فراهم‌سازی سرویس‌ها در گره‌ی منطقی کنترل‌کننده‌ی لبه اجرا می‌شود و راه‌حل‌های فراهم‌سازی پویای چندهدفه‌ی سرویس در مه، کمینه کردن تأخیر برنامه‌های کاربردی در مه و کمینه کردن هزینه‌ی منابع در این گره برای مقایسه اجرا می‌شوند. راه‌کارها از این‌پس به‌صورت خلاصه و به ترتیب MDSP، MinDelay و MinCost نامیده می‌شوند. در این مقاله، از پارامترهایی به‌عنوان ورودی مسئله‌ی فراهم‌سازی سرویس‌ها در مه استفاده شده است که در تحقیق‌های پیشین معرفی شده‌اند. تأخیر انتشار برای ارتباط میان دستگاه‌های اینترنت اشیاء و گره‌های مه یک عدد تصادفی با توزیع احتمال یکنواخت بین ۱ و ۲ میلی‌ثانیه و برای ارتباط میان گره‌های مه و ابر یک عدد تصادفی با توزیع احتمال یکنواخت بین ۱۵ و ۳۵ میلی‌ثانیه در نظر گرفته شده است [۳۵]. رسانه‌ی ارتباطی میان دستگاه‌های اینترنت اشیاء در یک منطقه و گره‌های مه مرتبط با آن‌ها، WiFi و لینک اتر نت یک گیگا بیت بر ثانیه است. همچنین، گره‌های مه با ابر با استفاده از رسانه‌های ۱۰۰ گیگا بیت بر ثانیه با فاصله‌ی ۲ تا ۶ گام در ارتباط هستند [۴]. سطح کیفیت سرویس

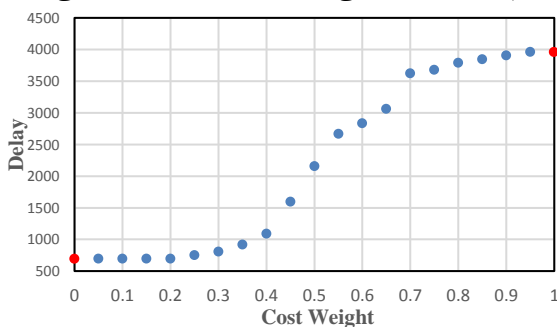
برای برنامه‌های کاربردی در این تنظیمات برابر با یک عدد تصادفی با توزیع احتمال یکنواخت بین ۹۰ و ۹۹ درصد است. همچنین، به دلیل انتخاب برنامه‌های کاربردی واقعیت افزوده که مثال خوبی برای برنامه‌های کاربردی اینترنت اشیاء هستند، آستانه‌ی زمان پاسخ برنامه‌های کاربردی در نظر گرفته شده در این ارزیابی، ۱۰ میلی‌ثانیه است [۲۳].

راه‌کار MDSP به‌صورت دوره‌ای اجرا می‌شود. این راه‌حل یک راه‌حل نگاه به جلو است و برای پیش‌بینی درخواست‌های ورودی در بازه‌ی پیش رو، از پنجره‌ی تاریخیچه‌ی اجرای برنامه‌ی کاربردی استفاده می‌کند. پنجره‌ی تاریخیچه یک‌ساعته انتخاب شده و نرخ درخواست‌های ورودی برای گره‌های مه یک عدد تصادفی با توزیع احتمال پواسن میان ۳۰ و ۶۰ درخواست بر ثانیه است. در این شبیه‌سازی، ظرفیت پردازشی هر گره‌ی مه یک عدد تصادفی با توزیع احتمال یکنواخت بین ۸۰۰ و ۱۳۰۰ میلیون دستور بر ثانیه است. همچنین، ظرفیت پردازشی ابر نیز یک عدد تصادفی با توزیع احتمال یکنواخت بین ۸۰۰۰ و ۱۳۰۰۰ میلیون دستور بر ثانیه در نظر گرفته شده است [۳۶]. ظرفیت ذخیره‌سازی در نظر گرفته شده برای گره‌های مه ۲۵ گیگابایت است. مقدار حافظه‌ی کمکی برای هر سرویس یک عدد تصادفی با توزیع احتمال یکنواخت بین ۵۰ تا ۵۰۰ مگابایت است که حجم Linux containerهای شناخته شده است. ظرفیت ذخیره‌سازی برای ابر نیز ۲۵۰ گیگابایت است. حافظه‌ی گره‌های مه و ابر در این شبیه‌سازی به ترتیب ۸ گیگابایت و ۳۲ گیگابایت در نظر گرفته شده است [۴].

میانگین اندازه‌ی درخواست و پاسخ برای برنامه‌ی کاربردی به ترتیب عددهای تصادفی با توزیع احتمال یکنواخت بین ۱۰ و ۲۶ کیلوبایت و بین ۱۰ و ۲۰ بایت است [۳۲]. همچنین، مقدار مورد نیاز پردازش برای این برنامه‌های کاربردی یک عدد تصادفی با توزیع احتمال یکنواخت بین ۵۰ تا ۲۰۰ میلیون دستور بر ثانیه است [۳۷]. واحد هزینه‌ی ارتباطی میان گره‌های مختلف ۰.۲ بر گیگا بیت است. واحد هزینه‌ی ارتباطی میان منبع تصاویر containerها و گره‌های مه که در رابطه‌ی (۶) برای محاسبه‌ی هزینه‌ی قرار دهی سرویس‌ها در مه استفاده شده است نیز ۰.۲ بر گیگا بیت در نظر گرفته شده است. واحد هزینه‌ی تخطی که در رابطه‌ی (۷) برای محاسبه‌ی هزینه‌ی تخطی کل استفاده شده است، برابر با یک عدد تصادفی با توزیع احتمال یکنواخت بین ۱ و ۲ بر درصد بر ثانیه مشخص شده است. واحد هزینه‌ی پردازش در مه برابر با ۰.۰۲ بر میلیون دستور است و واحد هزینه‌ی پردازش در ابر ۰.۰۰۲ بر میلیون دستور در نظر گرفته شده است. برای ذخیره‌سازی نیز، برای هر گیگا بیت بر ثانیه، ۰.۰۴ واحد هزینه در

را نشان می‌دهند. برای نشان دادن تأثیر وزن‌های مختلف روی مقادیر به‌دست‌آمده برای هزینه‌ی منابع و تأخیر برنامه‌های کاربردی، شکل‌های ۱ تا ۳ در یک شبیه‌سازی به دست آمده‌اند. شکل‌های ۱ و ۲ به ترتیب، تأثیر وزن‌های مختلف روی مقادیر تأخیر برنامه‌های کاربردی و هزینه‌ی منابع را نشان می‌دهند. همچنین، شکل ۳ با کنار هم قرار دادن مقادیر تأخیر درخواست‌های برنامه‌های کاربردی و هزینه‌ی منابع برای وزن‌های مختلف، برای مقایسه‌ی مقادیر این دو در کنار هم، رسم شده است.

از آنجاکه مجموع مقادیر وزن‌ها برای هر دو هدف برابر با یک است و اینکه از وزن یک هدف می‌توان وزن هدف دیگر را به دست آورد، برای یکپارچگی تنها وزن هزینه‌ی منابع در شکل‌های مختلف رسم شده است. در شکل‌های ۱ تا ۳، نتایج اجراهای مختلف راه‌کار MDSP برای وزن‌های مختلف با اخلاف ۰.۰۵ نشان داده شده است. نتایج به دست آمده از اجرای MDSP با وزن‌های صفر و یک برای هزینه‌ی منابع، به ترتیب برابر است با نتایج به‌دست‌آمده از دو روش MinDelay و MinCost. جواب‌های به دست آمده از روش‌های تک‌هدفه در شکل ۱ و ۲ با رنگ قرمز مشخص شده‌اند. زمانی که وزن هزینه‌ی منابع صفر در نظر گرفته می‌شود، هدف هزینه‌ی منابع برای تصمیم‌گیرنده اهمیتی ندارد و تنها تأخیر برنامه‌های کاربردی کمینه می‌شود، این نمونه از اجرای MDSP با MinDelay که یک روش تک‌هدفه برای کمینه کردن تأخیر برنامه‌های کاربردی در فراهم‌سازی سرویس در مه است، نتایج یکسانی دارند. همان‌طور که در شکل ۱ مشخص است، هر چه وزن تأخیر درخواست‌ها از صفر بیشتر می‌شود، مقدار تابع هدف تأخیر برنامه‌های کاربردی کمتر می‌شود و به سمت مقدار بهینه‌ی خود حرکت می‌کند. زمانی که مقدار وزن هزینه‌ی منابع ۱ است، برای تصمیم‌گیرنده تنها هزینه‌ی منابع اهمیت دارد و مقدار تأخیر به‌دست‌آمده از اجرای MDSP، بیشترین مقدار ممکن میان تمامی وزن‌ها را دارد. این مقدار برابر با مقدار تأخیر نتیجه شده از اجرای MinCost است که در این روش نیز، قیدهای تأخیر در تصمیم‌گیری شرکت نمی‌کنند و فقط هزینه کمینه می‌شود و با



شکل ۱: تأثیر وزن‌های مختلف بر تأخیر برنامه‌های کاربردی

مه و ۰.۰۰۴ واحد هزینه در ابر باید پرداخت شود. در واقع، در سلسله‌مراتب منابع از لبه‌ی شبکه تا ابر، با حرکت از لبه‌ی شبکه به سمت ابر هزینه‌ی منابع کمتر و ظرفیت آن‌ها نیز افزایش می‌یابد.

۴-۲- ارزیابی نتایج

در این بخش نشان داده خواهد شد که در صورتی که تصمیم‌گیرنده برای قرار دادن سرویس‌ها از روش‌های تک‌هدفه استفاده کند، ویژگی‌های دیگر تصمیم چه مقادیری خواهند داشت. برای مثال، اگر تنها تأخیر برنامه‌های کاربردی به‌عنوان تابع هدف مسئله‌ی فراهم‌سازی سرویس انتخاب شود، مقادیر هزینه و منابع استفاده‌شده در لبه‌ی شبکه مقادیر کاملاً متفاوتی نسبت به زمانی که هزینه به‌عنوان تابع هدف انتخاب شود دارند. همچنین، نشان داده خواهد شد که با استفاده از روش‌های چندهدفه، امکان دستیابی به یک مصالحه میان اهداف مهم در تصمیم‌گیری برای فراهم‌سازی سرویس‌ها در مه فراهم می‌شود.

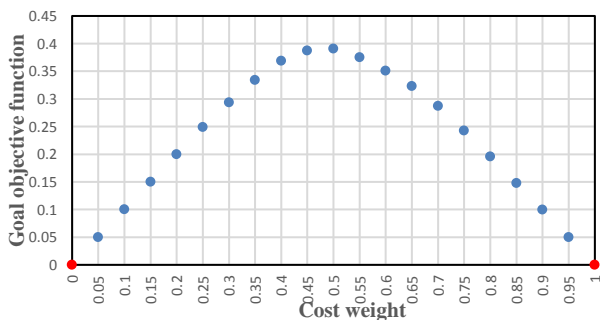
برای نشان دادن تأثیر روش‌های چندهدفه نسبت به تک‌هدفه برای فراهم‌سازی سرویس در مه، از شبیه‌سازی‌های گوناگونی استفاده شده است. از آنجاکه در این تحقیق هزینه‌ی منابع و تأخیر درخواست‌های برنامه‌های کاربردی اینترن‌ت اشیاء به‌عنوان تابع‌های هدف انتخاب شده‌اند، مقادیر به‌دست‌آمده برای این هدف‌ها با اجرای راه‌کارهای MDSP، MinDelay و MinCost با هم مقایسه می‌شوند. در شبیه‌سازی‌های مختلف، مجموع وزن تابع هدف هزینه‌ی منابع و وزن تابع هدف تأخیر برنامه‌های کاربردی برابر با یک است. در این مقایسه، مقادیر تابع‌های هدف، میزان استفاده از منابع لبه، جبهه‌ی پرتو، اختلاف مقادیر هزینه و تأخیر و پیچیدگی محاسباتی بررسی می‌شوند. در مدل ارائه‌شده در بخش سه برای فراهم‌سازی چندهدفه‌ی سرویس در مه، تنها قیدهایی که تابع‌های هدف را به شکلی با یکدیگر مرتبط می‌سازد، رابطه‌های (۷) و (۱۴) هستند. این رابطه‌ها برای تعیین هزینه‌ی تخطی استفاده می‌شوند. این ارتباط میان هدف‌ها تأثیر ویژه‌ای در نتایج به‌دست‌آمده دارد. برای بررسی این تأثیر، میزان استفاده از منابع لبه و نمودار جبهه‌ی پرتو برای حالتی که هزینه‌ی تخطی از مدل حذف شود نیز بررسی می‌شوند.

۴-۲-۱- مقادیر تابع‌های هدف

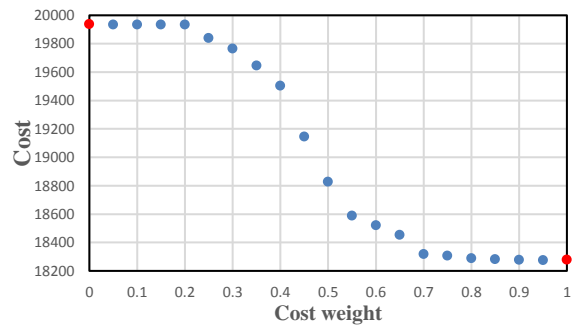
در این شبیه‌سازی، با مقایسه‌ی نتایج مختلف ارائه‌شده توسط راه‌کار MDSP با دو روش MinCost و MinDelay، مصالحه‌های مختلف میان هزینه‌ی منابع و تأخیر برنامه‌های کاربردی نشان داده می‌شود. وزن‌ها در روش برنامه‌ریزی آرمانی اهمیت هر تابع هدف

اجرای MDSP با وزن صفر برای هزینه‌ی منابع و وزن یک برای تأخیر برنامه‌های کاربردی، کمترین تأخیر و بیشترین هزینه‌ی منابع نسبت به دیگر اجراهای MDSP به دست می‌آید. این مقادیر دقیقاً برابر با خروجی روش MinDelay است و نشان‌دهنده‌ی بیشترین اهمیت برای کمینه کردن تأخیر است. با افزایش وزن هزینه‌ی منابع از مقدار صفر و اجراهای MDSP با استفاده از این وزن‌ها، رفته‌رفته هزینه کم می‌شود و به تأخیر درخواست‌های برنامه‌های کاربردی اضافه می‌شود. اگر تنها روش‌های MinCost یا MinDelay توسط کنترل‌کننده‌ی لبه اجرا شود، یا هزینه‌ی منابع کمینه می‌شود و یا تأخیر درخواست‌های برنامه‌های کاربردی. با کمینه کردن هر هدف، دیگر مؤلفه‌های مهم سیستم که با هدف کمینه شده در تضاد هستند، مقادیر غیر بهینه‌ای تولید می‌کنند. این در حالی است که تصمیم‌گیرنده راه‌حلی برای مصالحه میان این اهداف متضاد ندارد. در صورتی که از روش فراهم‌سازی پویای چندهدفه‌ی سرویس در مه استفاده شود، این امکان برای تصمیم‌گیرنده وجود دارد که با استفاده از وزن‌های مختلف برای اهداف بهینه‌سازی، به مصالحه‌های مختلف برای فراهم‌سازی سرویس دست یابد. ایجاد مصالحه میان هدف‌های متضاد می‌تواند به تصمیم‌گیرنده کمک کند تا سیاست‌های پویاتر و کاراتری نسبت به روش‌های تک‌هدفه برای فراهم‌سازی سرویس‌ها در رایانش مهی اتخاذ کند. در نتیجه، این امکان وجود دارد که مقادیر تأخیر برنامه‌های کاربردی و هزینه‌ی منابع توسط تصمیم‌گیرنده کنترل شوند.

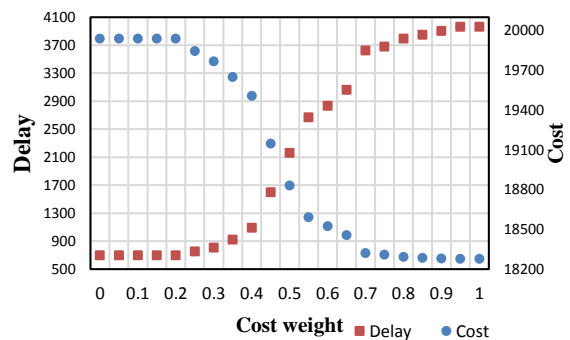
با استفاده از وزن‌های مختلف برای تابع‌های هدف هزینه‌ی منابع و تأخیر درخواست‌های برنامه‌های کاربردی، مقادیر متفاوتی برای مقدار تابع هدف راه‌کار ارائه شده با استفاده از برنامه‌ریزی وزن‌دار آرمانی در رابطه‌ی (۹) به دست می‌آید. این مقادیر برای وزن‌های مختلف در شکل ۴ نشان داده شده‌اند. با مقایسه‌ی شکل‌های ۳ و ۴ می‌توان مشاهده کرد که اختلاف مقادیر به‌دست‌آمده برای هزینه‌ی منابع و تأخیر برنامه‌های کاربردی با استفاده از اجراهای مختلف با وزن‌های مختلف، رابطه‌ای با مقدار تابع هدف روش برنامه‌ریزی وزن‌دار آرمانی دارد.



شکل ۴: مقدار تابع هدف روش برنامه‌ریزی وزن‌دار آرمانی

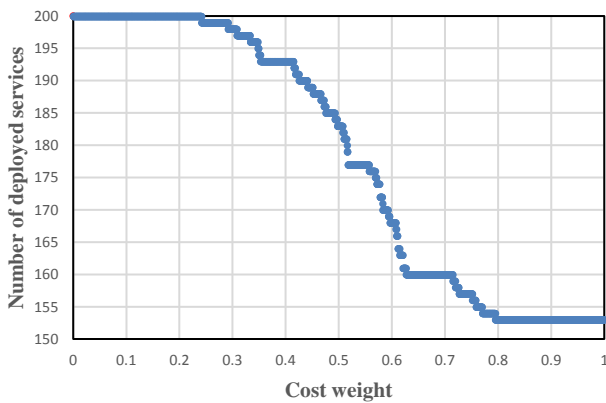


شکل ۲: تأثیر وزن‌های مختلف بر هزینه‌ی منابع



شکل ۳: مقایسه‌ی تابع‌های هدف برای وزن‌های مختلف

توجه به اینکه اهداف هزینه‌ی منابع و تأخیر درخواست‌های برنامه‌های کاربردی دو هدف متضاد و در مقابل هم هستند، تأخیر در این روش بیشترین مقدار خود را دارد. تمامی مقادیر به‌دست‌آمده برای تأخیر با استفاده از روش MDSP با وزن‌های مختلف، بین تأخیرهای حاصل از MinCost و MinDelay هستند. شکل ۲ هزینه‌ی به‌دست‌آمده از روش MDSP با وزن‌های مختلف را نشان می‌دهد. مانند تأخیر که در شکل ۱ نشان داده شد، هزینه‌ی منابع به‌دست‌آمده برای وزن‌های مختلف هزینه‌ی منابع ۰، ۰.۵، ۰.۱، ۰.۱۵ و ۰.۲ با یکدیگر برابرند. همچنین، وزن‌های ۱ و ۰.۹۵ برای هزینه‌ی منابع نیز مقادیر یکسانی به‌عنوان خروجی مسئله تولید می‌کنند. با این وجود، مقادیر هزینه‌ی منابع برای اجراهای MDSP با وزن‌های ۰.۲۵ تا ۰.۹ کاملاً متفاوت است و هر چه وزن هزینه‌ی منابع در رابطه‌ی (۹) که تابع هدف در روش برنامه‌ریزی آرمانی برای فراهم‌سازی چندهدفه‌ی سرویس‌ها است بیشتر شود، هزینه‌ی منابع کمتر شده و در نهایت به مقدار بهینه‌ی خود در اجراهای MDSP با وزن‌های ۱ و ۰.۹۵ می‌رسد. این نتایج نیز به دلیل وجود قیدهای مختلف و متغیر تصمیم دودویی قرار دادن سرویس‌ها در مه، به‌صورت خطی نیست و هزینه‌های منابع در اجراهای مختلف MDSP با تغییر وزن‌ها به همان نسبت تغییر نمی‌کند. شکل ۳، نتایج به‌دست‌آمده برای تأخیر و هزینه‌ی منابع در شکل‌های ۱ و ۲ را در کنار هم نشان می‌دهد. مهم‌ترین نتیجه‌ای که از این شکل می‌توان گرفت، الگوی تغییر مقادیر تأخیر و هزینه‌ی منابع در اجراهای MDSP با وزن‌های مختلف است. در



شکل ۵: تعداد سرویس‌های قرار داده شده در لبه‌ی شبکه با در نظر گرفتن هزینه‌ی تخطی

گیرنده بتواند تعداد سرویس‌های موردنظر خود را در منابع مه قرار دهد. این مصالحه با اجرای MDSP در کنترل‌کننده‌ی لبه ممکن می‌شود.

با افزایش وزن هزینه‌ی منابع از صفر و کاهش وزن تأخیر درخواست‌های برنامه‌های کاربردی از یک، رفته‌رفته تعداد سرویس‌های قرار داده شده در مه کمتر می‌شود. تمامی وزن‌های استفاده شده در ۱۰۰۱ اجرای MDSP می‌توانند منجر به قرار دادن ۱۵۳ تا ۲۰۰ سرویس روی منابع مه به‌جز چهار حالت شوند. از آنجاکه دو روش MinCost و MinDelay مانند حد پایین و بالای MDSP عمل می‌کنند، تعداد سرویس‌هایی که با استفاده از روش MDSP می‌توان در منابع مه قرار داد، حداکثر برابر با تعداد سرویس‌هایی است که با اجرای MinDelay در مه قرار داده می‌شود و حداقل برابر با تعداد سرویس‌هایی که با استفاده از اجرای MinCost در مه قرار داده می‌شود. از میان ۴۸ سرویسی که می‌توانند با استفاده از روش MDSP با وزن‌های مختلف به حداقل تعداد سرویس‌های قرار داده شده توسط MinCost اضافه شوند، تنها ۴ سرویس امکان قرار داده شدن در منابع مه را ندارند. به‌بیان‌دیگر، در اجرای MDSP هیچ وزنی برای هزینه‌ی منابع و تأخیر درخواست‌های برنامه‌های کاربردی وجود ندارد که بتواند تعداد ۱۷۸، ۱۷۳، ۱۶۵ و ۱۶۲ سرویس را روی منابع مه قرار دهد. این در صورتی است که برای تمامی تعداد سرویس‌ها از ۱۵۳ تا ۲۰۰، وزن‌های متفاوتی وجود دارند که در صورت اجرای MDSP با این وزن‌ها تعداد سرویس‌های موردنظر به‌جز ۴ تعداد سرویس ذکرشده، در منابع مه قرار داده شوند.

۴-۲-۳- نمودار جبهه‌ی پرتو

به مجموعه‌ای از تمامی جواب‌های بهینه پرتو برای یک مسئله‌ی تصمیم‌گیری، جبهه‌ی پرتو می‌گویند. منظور از جواب بهینه پرتو، راه‌حلی برای یک تابع هدف است که در صورت بهبود آن هدف،

درواقع، هر چه اختلاف این تابع‌های هدف بیشتر باشد، مقدار تابع هدف روش برنامه‌ریزی وزن دار آرمانی کمتر است. همچنین، در شکل ۴، نقطه‌ی بیشینه مربوط به اجرای MDSP با وزن ۰.۵ برای هر دو تابع هدف است. این نقطه‌ی بیشینه به معنای کمترین اختلاف میان تابع‌های هدف است. در اجراهای MDSP با وزن‌های صفر و یک برای هزینه‌ی منابع، بیشترین اختلاف میان مقادیر هزینه‌ی منابع و تأخیر برنامه‌های کاربردی محاسبه می‌شود. در شکل ۴، با استفاده از این وزن‌ها، مقدار تابع هدف روش برنامه‌ریزی آرمانی صفر و در حالت کمینه است.

۴-۲-۲- میزان استفاده از منابع لبه‌ی شبکه

میزان استفاده از منابع را می‌توان با تعداد سرویس‌های قرار داده شده در منابع مه بیان کرد. از آنجاکه در این تحقیق، هدف کمینه کردن هزینه‌ی منابع و تأخیر درخواست‌های برنامه‌های کاربردی است، در این بخش تأثیر سیاست اتخاذشده برای قرار دادن سرویس‌ها در راه‌کار فراهم‌سازی پویای چندهدفه‌ی سرویس‌ها در مه با استفاده از وزن‌های مختلف برای تابع‌های هدف، روی تعداد سرویس‌هایی که در مه قرار داده می‌شوند مورد بررسی قرار می‌گیرد. تعداد سرویس‌های قرار داده شده در مه به‌عنوان یک متغیر وابسته‌ی مسئله‌ی فراهم‌سازی برای هر سه روش MDSP، MinCost و MinDelay تعریف شده است. با اجرای MDSP با وزن‌هایی برای هزینه‌ی منابع بین صفر و یک و بافاصله‌ی ۰.۰۰۱، ۱۰۰۱ اجرای MDSP حاصل می‌شود. تعداد سرویس‌های قرار داده شده در لبه‌ی شبکه به‌عنوان یک متغیر وابسته در سه راه‌کار MDSP، MinCost و MinDelay، در شکل ۵ نشان داده شده است. زمانی که وزن هزینه‌ی منابع یک باشد نتایج حاصل از اجرای MDSP برابر است با روش تک‌هدفه‌ی MinCost. با استفاده از مجموعه داده‌ی موردبررسی، در صورتی که روش MinDelay در کنترل‌کننده‌ی لبه اجرا شود، تمامی سرویس‌ها روی تمامی منابع مه قرار داده می‌شوند. در صورتی که راه‌کار MinCost در کنترل‌کننده‌ی لبه اجرا شود، ۱۵۳ سرویس و در صورتی که MinDelay اجرا شود ۲۰۰ سرویس روی منابع مه قرار داده می‌شوند. نتایج حاصل از این دو روش با توجه به هزینه‌ی بیشتر قرار دادن سرویس‌ها روی منابع مه نسبت به منابع ابر کاملاً توجیه‌پذیر است.

استفاده از دو روش MinCost و MinDelay هیچ انتخابی را برای تصمیم‌گیرنده درباره‌ی تعداد سرویس‌های قرار داده شده در مه فراهم نمی‌کند. در نتیجه، یک مصالحه میان هزینه‌ی منابع و تأخیر درخواست‌های برنامه‌های کاربردی منجر می‌شود که تصمیم

را کاهش دهد، این کار با آزاد ساختن یک سرویس از منابع لبه‌ی شبکه و ارسال درخواست‌های مربوط به آن سرویس به ابر امکان پذیر است. در صورت انتقال درخواست‌های یک سرویس از لبه‌ی شبکه به ابر، به دلیل فاصله‌ی زیاد لبه‌ی شبکه تا ابر، تأخیر درخواست‌ها زیادتر می‌شود. نمودار جبهه‌ی پرتو برای نمونه‌ای از راه‌کار فراهم‌سازی پویای چندهدفه‌ی سرویس در مه که هزینه‌ی تخطی از آن حذف شده است در شکل ۷ رسم شده است.

تفاوت نمودار جبهه‌ی پرتو در شرایطی که هزینه‌ی تخطی از مدل حذف شده باشد با شکل ۶، مقیاس گسترده‌تر هزینه‌ی منابع و تأخیر است. این مقیاس از حالتی که هیچ سرویسی در مه قرار داده نشده است شروع می‌شود و تا جایی که تمامی سرویس‌ها در مه قرار داده شوند ادامه می‌یابد. مانند شکل ۶، در صورتی که مقدار یک تابع هدف کم‌تر از مقداری که در نمودار مشخص است شود، مقدار تابع هدف دیگر افزایش خواهد یافت. در نتیجه، تمامی نقاطی که بتوان با توجه به اهمیت هر تابع هدف، جواب بهینه را به دست آورد در شکل ۷ نمایش داده شده است.

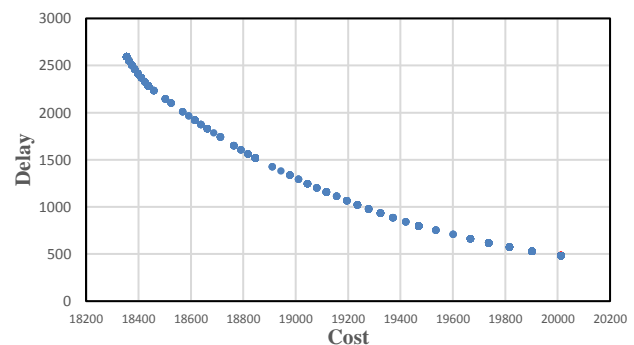
۴-۲-۴- اختلاف مقادیر هزینه و تأخیر

با توجه به نمودار جبهه‌ی پرتو می‌توان مشاهده کرد زمانی که تأخیر درخواست‌های برنامه‌های کاربردی کمترین مقدار و هزینه‌ی منابع بیشترین مقدار را دارند، نقاط به اندازه‌ی حالتی که هزینه‌ی منابع کمترین مقدار را دارد، به هم نزدیک نیستند. به دلیل وجود متغیر تصمیم دودویی، رفتار خطی مورد انتظار از مقادیر هزینه‌ی منابع و تأخیر برنامه‌های کاربردی در شکل‌های ۶ و ۷ مشاهده نمی‌شود. همچنین، وجود قیدهای پیچیده در مدل مسئله باعث می‌شود که برای درک این‌گونه رفتار راه‌کار MDSP به عوامل موجود در قیدها مراجعه شود. یکی از راه‌هایی که می‌توان پخش نبودن یکسان داده‌ها در نمودار جبهه‌ی پرتو را توضیح داد، بررسی اختلاف مقادیر نمونه‌های مختلف اجرای MDSP است.

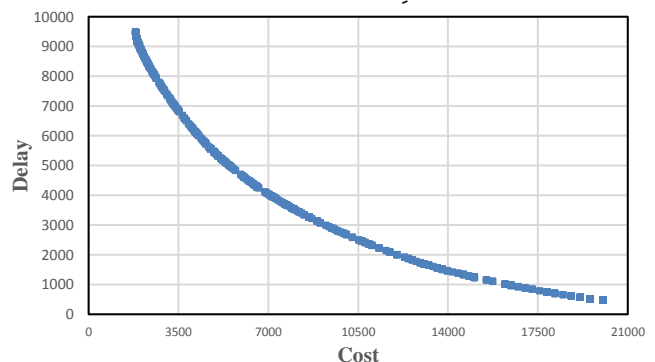
در شکل‌های ۸ و ۹، بر اساس تعداد سرویس‌هایی که از منابع مه آزاد می‌شوند، اختلاف اعداد به‌دست‌آمده برای اجراهای MDSP با وزن‌های مختلف نمایش داده شده است. برای مثال، در صورتی که یکی از منابع قرار داده شده در مه از آن آزاد شود، مقداری به تأخیر اضافه شده و مقداری از هزینه‌ی منابع کاهش می‌یابد. در صورتی که در هر گام یک سرویس از منابع مه آزاد شود، میزان تأخیر اضافه‌شده بر تأخیر فعلی در شکل ۸ نشان داده شده است. با توجه به رابطه‌ی ۱ برای تعیین تأخیر برنامه‌های کاربردی، در صورت آزاد شدن یک سرویس از مه، متغیر تصمیم دودویی قرار دادن سرویس در مه آن سرویس صفر شده و قسمت دوم رابطه که

دیگر هدف‌ها جواب‌های بدتری را فراهم کنند [۳۳]. هر نقطه از نمودار جبهه‌ی پرتو نشان دهنده‌ی یک اجرا از MDSP با وزن‌های مشخص است. با ۱۰۰۱ اجرای MDSP و رشد وزن هزینه‌ی منابع از صفر تا یک و با فاصله‌ی ۰.۰۰۱، مقادیر به‌دست‌آمده برای هزینه‌ی منابع به‌عنوان محور افقی و تأخیر درخواست‌های برنامه‌های کاربردی به‌عنوان محور عمودی در شکل ۶ رسم شده است. چهار خروجی برای تعداد سرویس‌های قرار داده شده در مه توسط وزن‌های مختلف تولید نمی‌شود. این خروجی در بخش قبل مورد بررسی قرار گرفت. این موضوع به دلیل ظرفیت منابع مه و نیازمندی‌های سرویس‌های مختلف اتفاق می‌افتد.

درواقع، هیچ جواب ممکن‌ی برای قرار دادن ۱۷۸، ۱۷۳، ۱۶۵ و ۱۶۲ سرویس در مه وجود ندارد. در بررسی هر نقطه از شکل ۶، با تلاش برای بهبود یک هدف، هدف دیگر افزایش پیدا کرده و از نقطه‌ی بهینه‌ی خود دور می‌شود. برای مثال، در نقطه‌ی (۸۰۰.۱۳۴، ۱۹۴۶۹.۰۰۷) در صورتی که بخواهیم هزینه‌ی منابع را کاهش دهیم به نقطه‌ی (۸۴۵.۰۳۳، ۱۹۴۱۹.۳۴۱) می‌رسیم. در این نقطه، هزینه‌ی منابع کاهش یافته است ولی تأخیر درخواست‌های برنامه‌های کاربردی افزایش یافته و از نقطه‌ی بهینه‌ی خود دور شده است.

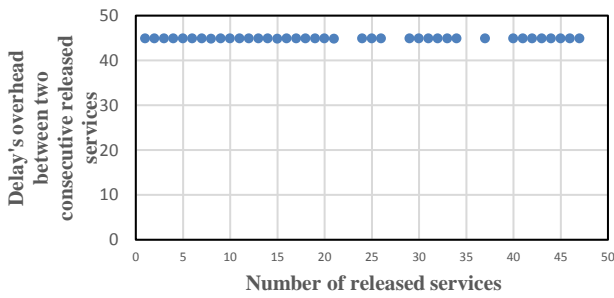


شکل ۶: نمودار جبهه‌ی پرتو با در نظر گرفتن هزینه‌ی تخطی

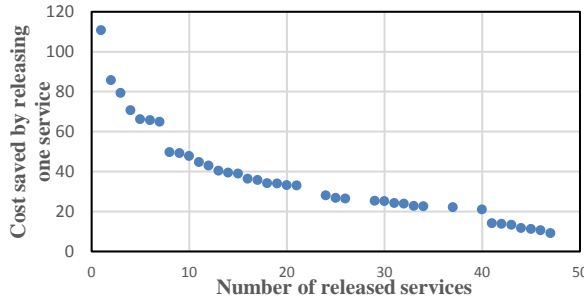


شکل ۷: نمودار جبهه‌ی پرتو بدون در نظر گرفتن هزینه‌ی تخطی

نتایج نشان داده شده در شکل ۶، با منطق یک محیط همکاری ابر و مه برای فراهم‌سازی چندهدفه‌ی سرویس‌های اینترنت اشیاء نیز همخوانی دارد. در صورتی که تصمیم‌گیرنده بخواهد هزینه‌ی منابع



شکل ۸: اختلاف مقادیر تأخیر برای اجراهای MDSP



شکل ۹: اختلاف مقادیر هزینه‌ی منابع برای اجراهای MDSP

۴-۲-۵- پیچیدگی محاسباتی

مدل ارائه شده در فصل سه، یک مدل برنامه‌ریزی خطی عدد صحیح است و از آنجا که پیچیدگی محاسباتی این نوع مدل‌های بهینه‌سازی را نمی‌توان اثبات کرد، در این بخش یک شبیه‌سازی برای بررسی پیچیدگی محاسباتی به صورت عددی انجام شده است. در این شبیه‌سازی افزایش زمان اجرای راه‌کار فراهم‌سازی پویای چندهدفه‌ی سرویس در مه، با افزایش تعداد گره‌های مه مورد بررسی قرار می‌گیرد.

زمان اجرای کل برای ۱۰، ۲۵، ۵۰، ۷۵، ۱۰۰، ۲۰۰، ۳۰۰، ۴۰۰ و ۵۰۰ گره‌ی مه در شکل ۱۱ نشان داده شده است. این زمان اجراها با استفاده از مفهوم اختلاف نسبی محاسبه شده است. این مفهوم نشان می‌دهد که پاسخ به دست آمده حداکثر چند درصد پاسخ بهینه با آن فاصله دارد. در نرم‌افزار GAMS، برای حل مسئله‌های فراهم‌سازی برنامه‌های کاربردی در مه از اختلاف نسبی ۰.۰۱ استفاده شده است. برای هر تعداد گره‌ی مه، ۵۰ اجرا صورت گرفته است و کمینه، میانگین و بیشینه‌ی زمان‌های اجرا در شکل ۱۰ برای هر تعداد گره‌ی مه رسم شده است. در تنظیمات اولیه‌ی شبیه‌سازی، تعداد گره‌های مه تنها ۱۰ عدد انتخاب شده بود. دلیل این انتخاب، وجود مفهوم منطقه برای منابع مه و ابر و دستگاه‌های اینترنت اشیا است. به بیان دیگر، در صورتی که منطقه در مرحله‌ی برنامه‌ریزی برای تعیین مقیاس و پیش‌بینی تعداد گره‌های مه در هر منطقه به درستی و دقت انتخاب شود، رشد بسیار زیاد برای تعداد گره‌های مه به وجود نخواهد آمد. نتیجه‌ی به دست آمده نشان می‌دهد که راه‌کار ارائه شده حتی برای تعداد ۵۰۰ گره‌ی مه نیز

برای محاسبه‌ی تأخیر درخواست‌ها برای سرویس‌های ابر است به تأخیر فعلی اضافه می‌شود. این موضوع باعث می‌شود که در صورت آزاد سازی سرویس‌ها از منابع مه، در هر گام به طور تقریبی مقدار ثابتی به تأخیر درخواست‌های برنامه‌های کاربردی اضافه شود. در مجموعه داده‌ی استفاده شده در این شبیه‌سازی، این مقدار در حدود ۴۵ میلی‌ثانیه است. اختلاف مقادیر تأخیر برای وزن‌های مختلف تقریباً به صورت خطی رفتار می‌کند و تأثیری در سؤال اول بخش درباره‌ی چگونگی پخش نتایج در نمودار جبهه‌ی پرتو ندارد. برخلاف تأخیر برنامه‌های کاربردی، اختلاف هزینه‌ی منابع برای هر آزاد سازی سرویس از مه به صورت تقریباً خطی عمل نمی‌کند. همان‌طور که در شکل ۹ مشخص است، زمانی که اولین سرویس از مه حذف می‌شود، اختلاف هزینه‌ی به دست آمده از اجرای این نمونه از MDSP با هزینه‌ی منابع زمانی که MDSP با وزن یک برای تأخیر اجرا می‌شود، در حدود ۱۱۰ و بیشترین اختلاف را میان دیگر گام‌ها دارد. این در حالی است که هر چه از منابع مه بیشتر کم شود، اختلاف مقادیر هزینه‌ی منابع برای اجراهای مختلف کمتر می‌شود. گسستگی میان نتایج به دست آمده در نمودار جبهه‌ی پرتو زمانی که وزن هزینه‌ی منابع نزدیک به صفر است، به دلیل همین غیرخطی بودن اختلاف مقادیر هزینه در گام‌های مختلف است. برای مثال، زمانی که وزن هزینه‌ی منابع نزدیک به یک است، نتایج MDSP با وزن‌های مختلف به هم نزدیک هستند. در شکل ۹ نیز، زمانی که وزن هزینه‌ی منابع نزدیک به یک است، اختلاف هزینه‌ی گام‌های مختلف بسیار به هم نزدیک هستند. هزینه‌ی کل از عوامل گوناگونی تشکیل می‌شود. با بررسی مقادیر مختلف زیر هزینه‌ها، دلیل اصلی این رشد اختلاف میان هزینه‌های منابع در اجراهای مختلف MDSP رابطه‌ی (۸) که هزینه‌ی پردازش در مه است تشخیص داده شد.

زمانی که راه‌کار MinCost در کنترل‌کننده‌ی لبه اجرا شود، ۱۵۳ سرویس در منابع مه قرار داده می‌شود. در صورتی که ضریبی برای تأخیر درخواست‌های برنامه‌های کاربردی در نظر گرفته شود، یک سرویس به مجموعه سرویس‌های قرار گرفته در مه اضافه می‌شود. هر چه تعداد سرویس‌های بیشتری در مه قرار داده شوند، هزینه‌ی پردازش در مه افزایش یافته و به دنبال آن هزینه‌ی کل نیز نسبت به حالت قبل افزایش بیشتری می‌یابد. در نتیجه، در نقطه‌هایی که نزدیک به وزن صفر برای هزینه‌ی منابع هستند، تعداد نقاطی که بتوان متغیرهای تصمیم دودویی را به شکلی انتخاب کرد که جواب ممکن برای مسئله باشند، نسبت به تعداد جواب‌هایی از MDSP که در نزدیکی وزن یک برای هزینه‌ی منابع هستند، کمتر خواهند بود.

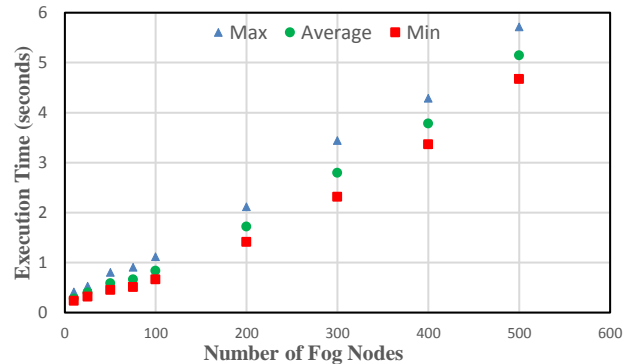
جواب‌های آن‌ها با جواب‌های بهینه‌ی ارائه‌شده در این تحقیق، می‌تواند بسیار سودمند باشد.

مراجع

- [1] K. Bachmann, "Design and implementation of a fog computing framework," Master's thesis, Vienna University of Technology (TU Wien), Vienna, Austria, 2017.
- [2] R. Mahmud, S. N. Srirama, K. Ramamohanarao, and R. Buyya, "Quality of Experience (QoE)-aware placement of applications in Fog computing environments," *Journal of Parallel and Distributed Computing*, 2018.
- [3] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, pp. 112-116, 2016.
- [4] A. Yousefpour, A. Patil, G. Ishigaki, I. Kim, X. Wang, H. C. Cankaya, et al., "FogPlan: A Lightweight QoS-aware Dynamic Fog Service Provisioning Framework," *IEEE Internet of Things Journal*, 2019.
- [5] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, et al., "Fog Computing: survey of trends, architectures, requirements, and research directions," *IEEE access*, vol. 6, pp. 47980-48009, 2018.
- [6] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing iot service delay via fog offloading," *IEEE Internet of Things Journal*, 2018.
- [7] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner, "Optimized IoT service placement in the fog," *Service Oriented Computing and Applications*, vol. 11, pp. 427-443, 2017.
- [8] Q. T. Minh, D. T. Nguyen, A. Van Le, H. D. Nguyen, and A. Truong, "Toward service placement on fog computing landscape," in *2017 4th NAFOSTED conference on information and computer science*, 2017, pp. 291-296.
- [9] V. B. C. Souza, W. Ramirez, X. Masip-Bruin, E. Marín-Tordera, G. Ren, and G. Tashakor, "Handling service allocation in combined fog-cloud scenarios," in *2016 IEEE international conference on communications (ICC)*, 2016, pp. 1-5.
- [10] S. Wang, R. Urgaonkar, T. He, K. Chan, M. Zafer, and K. K. Leung, "Dynamic service placement for mobile micro-clouds with predicted future costs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, pp. 1002-1016, 2016.
- [11] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing," *Journal of Computer and System Sciences*, vol. 79, pp. 1230-1242, 2013.
- [12] M.-H. Malekloo, N. Kara, and M. El Barachi, "An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments," *Sustainable Computing: Informatics and Systems*, vol. 17, pp. 9-24, 2018.
- [13] S. Ghasemi-Falavarjani, M. Nematbakhsh, and B. S. Ghahfarokhi, "Context-aware multi-objective resource allocation in mobile cloud," *Computers & Electrical Engineering*, vol. 44, pp. 218-240, 2015.
- [14] C. Guerrero, I. Lera, and C. Juiz, "Genetic algorithm for multi-objective optimization of container allocation in cloud architecture," *Journal of Grid Computing*, vol. 16, pp. 113-135, 2018.
- [15] R. Jena, "Multi objective task scheduling in cloud environment using nested PSO framework," *Procedia Computer Science*, vol. 57, pp. 1219-1227, 2015.
- [16] B. Shrimali and H. Patel, "Multi-objective optimization oriented policy for performance and energy efficient resource allocation in Cloud environment," *Journal of King Saud University-Computer and Information Sciences*, 2017.
- [17] C. Guerrero, I. Lera, and C. Juiz, "Resource optimization of container orchestration: a case study in multi-cloud microservices-based applications," *The Journal of Supercomputing*, vol. 74, pp. 2956-2983, 2018.
- [18] Y. Nan, W. Li, W. Bao, F. C. Delicato, P. F. Pires, and A. Y. Zomaya, "A dynamic tradeoff data processing framework for delay-sensitive applications in Cloud of Things systems,"

به‌خوبی عمل می‌کند و برای یک دوره‌ی اجرای راه‌حل، که در این شبیه‌سازی ۱۸۰ ثانیه در نظر گرفته شده، در حدود ۵ ثانیه تصمیم‌گیری می‌کند.

علاوه بر این، این ارزیابی برای پیچیدگی محاسباتی راه‌کار ارائه‌شده می‌تواند توسط تصمیم‌گیرنده در مرحله‌ی برنامه‌ریزی منابع مه و تعیین منطقه به کار گرفته شود.



شکل ۱۰: زمان حل MDSP

۵- نتیجه‌گیری و کارهای آینده

برنامه‌های کاربردی اینترنت اشیا نیاز به دریافت پاسخ‌های بسیار سریع برای درخواست‌هایشان دارند. در ابتدا و بعد از مطرح شدن مفهوم اینترنت اشیا، رایانش ابری به‌عنوان رایانشی برای میزبانی برنامه‌های کاربردی اینترنت اشیا در نظر گرفته شد. با ظهور برنامه‌های کاربردی جدید با نیازمندی‌های متفاوت از برنامه‌های کاربردی موجود، به دلیل فاصله‌ی زیاد مراکز داده‌ی ابر از لبه‌ی شبکه و جایی که داده تولید می‌شود، نیازمندی‌های برنامه‌های کاربردی تنها با رایانش ابری برآورده نمی‌شد. به‌منظور برطرف کردن این چالش، رایانش مهی برای قرار دادن برنامه‌های کاربردی نزدیک به لبه‌ی شبکه معرفی شد. در این مقاله، یک محیط همکاری ابر و مه به‌منظور قرار دادن برنامه‌های کاربردی اینترنت اشیا معرفی شد. مسئله‌ی فراهم‌سازی سرویس‌ها در مه برای مصالحه میان هزینه و تأخیر با استفاده از برنامه‌ریزی آرمانی به‌عنوان مسئله‌ی موردبررسی در این مقاله معرفی شد.

در این مقاله برای بالا بردن قابلیت اطمینان، تمامی سرویس‌ها در ابر قرار داده شده‌اند. ارائه‌ی مدلی برای بهبود قابلیت اطمینان در یک محیط همکاری ابر و مه می‌تواند امکان قرار دادن سرویس‌ها به‌صورت پویا را در گره‌های ابر فراهم کند. هر کدام از نقاط پرتو ارائه شده در این مقاله می‌توانند مورد استفاده قرار گیرند تا میان اهداف متضاد مصالحه ایجاد شود. یافتن یک سیاست فراهم‌سازی سرویس از میان جواب‌های ممکن که در این مقاله ارائه شده‌اند به کارهای آینده موکول شده است. استفاده از روش‌های فرا ابتکاری برای حل مسئله‌ی فراهم‌سازی سرویس‌ها در مه و مقایسه‌ی

international conference on Fog and Edge Computing (ICFEC), 2017, pp. 89-96.

زیر نویس ها:

- ¹ Resource provisioning
- ² Dispatching
- ³ Service migration
- ⁴ Integer non-Linear Programming (INLP)
- ⁵ Integer Linear Programming (ILP)
- ⁶ Trade-off
- ⁷ Quality of Service (QoS)
- ⁸ Availability
- ⁹ Violation
- ¹⁰ Weighted goal programming

- Journal of Parallel and Distributed Computing*, vol. 112, pp. 53-66, 2018.
- [19] K. Kaur, T. Dhand, N. Kumar, and S. Zeadally, "Container-as-a-service at the edge: Trade-off between energy efficiency and service availability at fog nano data centers," *IEEE wireless communications*, vol. 24, pp. 48-56, 2017.
- [20] A. Majd, G. Sahebi, M. Daneshalab, J. Plosila, and H. Tenhunen, "Hierarchical placement of smart mobile access points in wireless sensor networks using fog computing," in *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, 2017, pp. 176-180.
- [21] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet of Things Journal*, vol. 5, pp. 283-294, 2017.
- [22] Y. Sun, F. Lin, and H. Xu, "Multi-objective optimization of resource scheduling in Fog computing using an improved NSGA-II," *Wireless Personal Communications*, vol. 102, pp. 1369-1385, 2018.
- [23] C. C. Byers, "Architectural imperatives for fog computing: Use cases, requirements, and architectural techniques for fog-enabled IoT networks," *IEEE Communications Magazine*, vol. 55, pp. 14-20, 2017.
- [24] E. Saurez, K. Hong, D. Lillethun, U. Ramachandran, and B. Ottenwalder, "Incremental deployment and migration of geodistributed situation awareness applications in the fog," in *Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems*, 2016, pp. 258-269.
- [25] G. Lee, W. Saad, and M. Bennis, "An online optimization framework for distributed fog network formation with minimal latency," *IEEE Transactions on Wireless Communications*, vol. 18, pp. 2244-2258, 2019.
- [26] H. R. Arkian, 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.
- [27] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative Cloud and Edge Computing for Latency Minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, pp. 5031-5044, 2019.
- [28] L. Yang, J. Cao, G. Liang, and X. Han, "Cost aware service placement and load dispatching in mobile cloud systems," *IEEE Transactions on Computers*, vol. 65, pp. 1440-1452, 2016.
- [29] S. Sobhanayak, A. K. Turuk, and B. Sahoo, "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm," *Future Computing and Informatics Journal*, 2018.
- [30] D. Gonalves, K. Velasquez, M. Curado, L. Bittencourt, and E. Madeira, "Proactive virtual machine migration in fog environments," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018, pp. 00742-00745.
- [31] A. Brogi, S. Forti, and A. Ibrahim, "Optimising QoS-assurance, Resource Usage and Cost of Fog Application Deployments," presented at the Communications in Computer and Information Science, 2018.
- [32] K. Ha, P. Pillai, G. Lewis, S. Simanta, S. Clinch, N. Davies, et al., "The impact of mobile multimedia applications on data center consolidation," in *2013 IEEE international conference on cloud engineering (IC2E)*, 2013, pp. 166-176.
- [33] D. Jones and M. Tamiz, *Practical goal programming* vol. 141: Springer, 2010.
- [34] F. S. Hillier, *Introduction to operations research*: Tata McGraw-Hill Education, 2012.
- [35] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: Towards minimizing delay in the internet of things," in *2017 IEEE international conference on edge computing (EDGE)*, 2017, pp. 17-24.
- [36] A. Kapsalis, P. Kasnesis, I. S. Venieris, D. I. Kaklamani, and C. Z. Patrikakis, "A cooperative fog approach for effective workload balancing," *IEEE Cloud Computing*, vol. 4, pp. 36-45, 2017.
- [37] O. Skarlat, M. Nardelli, S. Schulte, and S. Dustdar, "Towards qos-aware fog service placement," in *2017 IEEE Ist*