

## A Multi-objective Model for Task Scheduling Optimization in Fog-Cloud Computing Environments

Sadoon Azizi<sup>1\*</sup>

<sup>1\*</sup>- Department of Computer Engineering and IT, University of Kurdistan, Sanandaj, Iran.

<sup>1\*</sup>s.azizi@uok.ac.ir

Corresponding author's address: Sadoon Azizi, Sanandaj, Pasdaran St, University of Kurdistan , Department of Computer Engineering and IT.

**Abstract-** With the advent of Internet of Things (IoT) applications, the number of processing requests has dramatically increased. In order to response to these requests, the Fog-Cloud environment has recently been introduced as a hybrid computing system. Although, the Fog-Cloud is a very promising environment for processing IoT requests, it faces many challenges. In this regard, task scheduling problem is one of the key challenges which has a significant impact on the efficiency and overall system cost. Motivated by this, in this paper, first a multi-objective optimization model including makespan, energy consumption and processing cost for scheduling tasks in an integrated Fog-Cloud environment is presented. Then, a heuristic algorithm to efficiently solve the model is proposed. Simulation results demonstrate that the proposed algorithm significantly reduces all the aforementioned metrics and can achieve a good tradeoff between them. Specifically, the proposed algorithm improves the objective function around 98%, 43% and 32% in comparison with the random, genetic and the power of two choices algorithms, respectively.

**Keywords-** Internet of Things (IoT), Cloud Computing, Fog Computing, Task Scheduling Problem, Multi-objective Optimization, Heuristic Algorithm.

## یک مدل چندهدفه برای بهینه‌سازی زمان‌بندی وظیفه‌ها در محیط‌های رایانشی

### مه-ابر

سعدون عزیزی<sup>۱\*</sup>

\*۱- گروه مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه کردستان، سنندج، ایران.

<sup>۱\*</sup>s.azizi@uok.ac.ir

\* نشانی نویسنده مسئول: سعدون عزیزی، سنندج، بلوار پاسدران، دانشگاه کردستان، گروه مهندسی کامپیوتر و فناوری اطلاعات.

چکیده- با ظهور برنامه‌های کاربردی مبتنی بر اینترنت اشیاء، تعداد درخواست‌های پردازشی به شدت افزایش یافته است. به منظور پاسخگویی به این درخواست‌ها، اخیراً محیط مه-ابر به عنوان یک سیستم رایانشی ترکیبی ارائه شده است. اگرچه مه-ابر یک محیط بسیار امیدبخش برای پردازش درخواست‌های اینترنت اشیاء است، اما با چالش‌های متعددی مواجه است. یکی از چالش‌های کلیدی، مسئله زمان‌بندی وظیفه‌ها است که تأثیر به‌سزایی روی کارایی و هزینه کلی سیستم دارد. با این انگیزش، در این مقاله ابتدا یک مدل بهینه‌سازی چندهدفه شامل زمان خاتمه آخرین وظیفه، مصرف انرژی و هزینه پردازش برای مسئله زمان‌بندی وظیفه‌ها در محیط یکپارچه مه-ابر ارائه می‌شود. سپس یک الگوریتم ابتکاری کارآمد برای حل آن پیشنهاد می‌گردد. نتایج شبیه‌سازی نشان می‌دهد که الگوریتم پیشنهادی به طور چشمگیری هر سه معیار را کاهش می‌دهد و به خوبی می‌تواند بین آنها تعادل برقرار نماید. به طور مشخص، از نظر مقدار تابع هدف، الگوریتم پیشنهادی به طور متوسط ۹۸٪ بهتر از روش تصادفی، ۴۳٪ بهتر از الگوریتم ژنتیک و ۳۲٪ بهتر از روش قدرت دو انتخاب عمل می‌کند.

واژه‌های کلیدی: اینترنت اشیاء، رایانش ابری، رایانش مه، مسئله زمان‌بندی وظیفه‌ها، بهینه‌سازی چند هدفه، الگوریتم ابتکاری.

### ۱- مقدمه

پارادایم رایانش مه پیشنهاد شد [۳]. رایانش مه یک معماری توزیع شده است که قابلیت‌های ابر را به لبه شبکه گسترش داده و منابع محاسباتی، ذخیره‌سازی و شبکه‌ای را به نزدیکی کاربران اینترنت اشیاء می‌آورد. در واقع، رایانش مه مکملی برای رایانش ابر است که با یکپارچه کردن آنها یک محیط رایانشی امیدبخش به نام مه-ابر تشکیل می‌شود.

همکاری مه و ابر امکان این را می‌دهد که نسل جدیدی از برنامه‌های کاربردی و سرویس‌های جریانی و بی‌درنگ ظهور کنند. از طرف این برنامه‌های کاربردی تعداد زیادی درخواست تولید می‌شود که بایستی توسط منابع موجود در محیط مه و ابر رسیدگی شود. به طور واضح‌تر، هر درخواست شامل مجموعه‌ای از وظیفه‌ها است که باید توسط دستگاه‌های مه و سرورهای ابر پردازش شود.

با فراگیر شدن اینترنت اشیاء، تعداد برنامه‌های کاربردی و درخواست‌های پردازشی به سرعت در حال افزایش است. به عنوان مثال، بر اساس پیش‌بینی سیسکو، تا سال ۲۰۲۵ تعداد دستگاه‌های اینترنت اشیاء به ۷۵ میلیارد خواهد رسید [۱]. برای حمایت از نیازمندی‌های این دستگاه‌های با منابع محدود، یکپارچه‌سازی اینترنت اشیاء و رایانش ابر از اهمیت به‌سزایی برخوردار است [۲]. اگرچه تامین‌کنندگان سرویس‌های ابری روز به روز در حال استقرار زیرساخت‌های توزیع شده در نزدیکی شبکه‌های دسترسی کاربران هستند، اما به دلیل تأخیر ارتباطی بالا و مصرف زیاد پهنای باند شبکه، ابر به تنهایی قادر به پاسخگویی برنامه‌های کاربردی حساس به تأخیر نیست. به منظور رفع محدودیت‌های ابر،

کنند که زمان اجرا و حافظه اختصاص داده شده کمینه شود. در راستای بهبود این مقاله، در [۷] یک الگوریتم مبتنی بر ژنتیک برای حل مسئله زمان‌بندی وظیفه‌ها در محیط یکپارچه مه-ابر پیشنهاد شده است که هدف‌های اصلی آن کاهش زمان خاتمه آخرین وظیفه و هزینه پولی سیستم است. به منظور بهبود زمان پاسخ وظیفه‌ها، در [۱۲] نویسندگان یک مدل بهینه‌سازی تک هدفه ارائه داده‌اند که تابع هدف آن روی بهینه‌سازی تأخیر ارتباطی کل تمرکز دارد. برای حل مدل، آنها دو الگوریتم ابتکاری طراحی کرده‌اند که هر دوی آنها از پیچیدگی زمانی خوبی برخوردار هستند. به طور مشابه، در [۱۳] از الگوریتم ژنتیک و در [۱۴] از الگوریتم ترکیبی بهینه‌سازی ازدحام ذرات و منطق فازی برای کمینه‌سازی زمان پاسخ وظیفه‌ها استفاده شده است. در همین راستا، نویسندگان در [۱۵] یک تکنیک ابتکاری بر مبنای ترکیب استراتژی اولویت‌بندی و حریم‌بندی برای بهبود میانگین زمان پاسخ وظیفه‌ها پیشنهاد داده‌اند. علاوه بر این، به منظور کاهش زمان تکمیل وظیفه‌ها، در [۱۶] الگوریتم قدرت d-انتخاب<sup>۷</sup> برای حل مسئله زمانبندی وظیفه‌ها در محیط‌های ناهمگن ارائه شده است. اخیراً در [۱۷]، نویسندگان دو الگوریتم ابتکاری کارآمد برای کاهش هزینه محاسباتی، هزینه پهنای باند و هزینه نقض مهلت در محیط‌های رایانشی مه-ابر پیشنهاد داده‌اند. نتایج شبیه‌سازی نشان می‌دهد که الگوریتم‌های پیشنهادی آنها به طور چشمگیری هزینه کلی سیستم را کاهش می‌دهد. اگرچه همه کارهای ذکر شده سهم قابل توجهی در راستای حل مسئله زمان‌بندی وظیفه‌ها دارند، اما هیچ کدام از آنها مصرف انرژی را در مدل بهینه‌سازی خود لحاظ نکرده‌اند.

در چند سال اخیر کارهای زیادی در راستای بهینه‌سازی مصرف انرژی در سیستم‌های رایانشی مه-ابر انجام شده است. به عنوان مثال در [۱۸]، نویسندگان مسئله زمان‌بندی وظیفه‌ها در محیط مه-ابر را به سه زیرمسئله مستقل تبدیل کرده و با استفاده از تکنیک‌های بهینه‌سازی موجود، مصرف توان و تأخیر را به طور توأم کاهش داده‌اند. نتایج شبیه‌سازی نشان می‌دهد که رویکرد ارائه شده در این کار می‌تواند به تعادل خوبی بین مصرف توان و تأخیر برسد. نویسندگان در [۶] زمان‌بندی وظیفه‌ها را به عنوان یک مسئله بهینه‌سازی دو هدفه جهت کاهش زمان خاتمه آخرین وظیفه و مصرف انرژی فرموله کرده و سه الگوریتم فراابتکاری<sup>۸</sup> بهینه‌سازی ازدحام ذرات، بهینه‌سازی ازدحام ذرات دودویی و خفاش<sup>۹</sup> برای حل آن ارائه داده‌اند. نتایج شبیه‌سازی آنها بیانگر این است که عملکرد الگوریتم خفاش در مقایسه با دو الگوریتم دیگر بهتر است. اخیراً در [۱۰] یک رویکرد جدید بهینه‌سازی الهام

مسئله زمان‌بندی وظیفه‌ها به این صورت تعریف می‌شود که چگونه وظیفه‌ها به دستگاه‌های محاسباتی نگاشت شوند به گونه‌ای که یک یا چند هدف مشخص بهینه شوند [۴، ۵]. زمان خاتمه آخرین وظیفه<sup>۱</sup>، مصرف انرژی و هزینه پولی<sup>۲</sup> از مهمترین اهداف بهینه‌سازی به شمار می‌روند که در بسیاری از کارهای گذشته مورد توجه قرار گرفته‌اند [۱۰-۶]. اما هیچ کدام از آنها این سه هدف اساسی را به طور همزمان در نظر نگرفته‌اند. علاوه بر این، زمان اجرای بسیاری از الگوریتم‌های ارائه شده تاکنون یک چالش اساسی قلمداد می‌شود و در نتیجه مانع از آن می‌گردد که بتوان آنها را برای محیط‌های بی‌درنگ و با مقیاس بزرگ به کار گرفت.

در این مقاله، ابتدا یک مدل چندهدفه برای مسئله زمان‌بندی وظیفه‌ها در محیط‌های رایانشی مه-ابر ارائه می‌شود که در آن زمان خاتمه آخرین وظیفه، مصرف انرژی و هزینه پولی به طور توأم بهینه می‌شوند. سپس، یک الگوریتم ابتکاری<sup>۳</sup> کارآمد برای حل این مدل پیشنهاد می‌گردد که با تنظیم مناسب پارامترهای آن می‌توان به کارایی بسیار خوبی دست یافت. برای این منظور، در این کار دو معیار مؤثر جدید تحت عنوان نرخ پردازش-توان<sup>۴</sup> و نرخ پردازش-هزینه<sup>۵</sup> تعریف می‌شود. این دو معیار به ما کمک می‌کنند تا مدیریت منابع را به گونه‌ای انجام دهیم که مصرف انرژی و هزینه سیستم به طور چشمگیری کاهش پیدا کند. برخلاف بیشتر کارهای گذشته، الگوریتم پیشنهادی از زمان اجرای بسیار پایینی برخوردار است؛ بنابراین، به راحتی می‌توان از آن در محیط‌های بی‌درنگ و با مقیاس بزرگ نیز بهره‌مند شد.

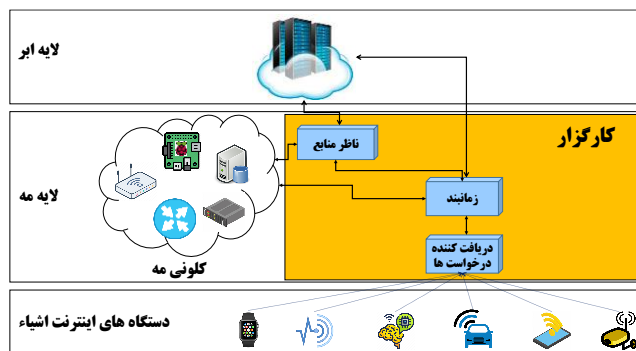
در ادامه، ساماندهی مقاله به این صورت است که در بخش ۲ به مرور کارهای گذشته پرداخته می‌شود. مدل‌سازی سیستم شامل معماری سیستم و فرموله‌سازی مسئله را در بخش ۳ ارائه می‌گردد. الگوریتم ابتکاری پیشنهادی در بخش ۴ توصیف می‌شود. بخش ۵ به ارزیابی کارایی الگوریتم پیشنهادی اختصاص داده شده است. در نهایت، در بخش ۶ مقاله با نتیجه‌گیری و بیان چند جهت‌دهی تحقیقاتی برای آینده به پایان می‌رسد.

## ۲- کارهای گذشته

مسئله زمان‌بندی وظیفه‌ها در محیط ترکیبی مه-ابر یک موضوع تحقیقاتی داغ است که در مقالات مروری [۲، ۴] پیش زمینه‌ها و رویکردهای موجود برای حل آن ارائه شده است. در ادامه بعضی از مهم‌ترین کارهای انجام شده در این راستا بررسی می‌شود.

نویسندگان در [۱۱] با استفاده از الگوریتم زنبور عسل<sup>۶</sup> تلاش کرده‌اند که وظیفه‌ها را در محیط رایانش مه به گونه‌ای زمان‌بندی

لایه مه شامل یک کلونی مه و یک کارگزار است. در کلونی مه مجموعه‌ای از دستگاه‌های محاسباتی مانند مسیریاب‌ها، سوئیچ‌ها، سرورهای کوچک، بردهای زبری پای و نقاط دسترسی قرار دارند که تحت عنوان گره‌های مه شناخته می‌شوند. هر گره مه شامل یک یا چند ماشین مجازی است که هر یک از آنها قابلیت پردازش وظیفه‌ها را دارد. بخش مهم لایه مه کارگزار آن است که از سه مؤلفه‌ی دریافت‌کننده درخواست‌ها، زمانبند و ناظر منابع تشکیل شده است. درخواست کاربران نهایی به صورت مستقیم یا غیرمستقیم ابتدا به مؤلفه‌ی دریافت‌کننده درخواست‌ها می‌رسد. این مؤلفه اطلاعات لازم در مورد هر درخواست مانند تعداد وظیفه‌های هر کار و اندازه هر وظیفه را جمع‌آوری می‌کند و آنها را تحویل مؤلفه زمانبند می‌دهد. زمانبند اطلاعات مربوط به منابع پردازشی مانند در دسترس بودن، قدرت محاسباتی، هزینه پردازش و پارامترهای مصرف توان ماشین‌های مجازی را توسط ناظر منابع دریافت می‌کند. سپس بر اساس اطلاعاتی که در اختیار دارد، الگوریتم زمان‌بندی را اجرا می‌کند و وظیفه‌ها را بین ماشین‌های مجازی توزیع می‌کند. در نهایت خروجی حاصل از اجرای وظیفه‌ها را از طریق دریافت‌کننده درخواست‌ها تحویل کاربران نهایی می‌دهد. لازم به ذکر است که لایه مه نیز شامل مجموعه‌ای از ماشین‌های مجازی است که معمولاً قدرت محاسباتی آنها از ماشین‌های مجازی لایه مه بالاتر است.



شکل ۱. معماری سیستم در نظر گرفته شده

### ۳-۲- فرموله‌سازی مسئله

در ابتدا عناصر مسئله بهینه‌سازی زمان‌بندی وظیفه‌ها در سیستم مه-ابر معرفی می‌شود. فرض کنید تعداد کل ماشین‌های مجازی در دسترس برابر  $M$  باشد که ما آنها را توسط مجموعه  $V = \{v_1, v_2, \dots, v_M\}$  نشان می‌دهیم. لازم به ذکر است که همانند [۶, ۸] برای سهولت فرض می‌شود که هر گره محاسباتی شامل یک ماشین مجازی است. هر ماشین مجازی دارای یک سری ویژگی است.  $v_i^{cpu}$  بیانگر سرعت پردازش ماشین مجازی  $v_i$  است که واحد

گرفته از زیست به نام بهینه‌سازی جفت‌گیری مورچه<sup>۱۰</sup> برای حل مسئله زمان‌بندی وظیفه‌ها در محیط رایانش مه پیشنهاد شده است. ارزیابی کارایی مقاله نشان می‌دهد که رویکرد پیشنهادی از نظر زمان خاتمه آخرین وظیفه و مصرف انرژی در مقایسه با الگوریتم‌های زنبور عسل، بهینه‌سازی ازدحام ذرات و ژنتیک عملکرد بهتری دارد. به طور مشابه، نویسندگان در [۸] الگوریتم بهینه‌سازی شاهین‌های هریس<sup>۱۱</sup> مبتنی بر استراتژی جستجوی محلی را برای کاهش زمان خاتمه آخرین وظیفه و مصرف انرژی ارائه داده‌اند. اگرچه این کارها مصرف انرژی را به عنوان یکی از اهداف بهینه‌سازی در نظر گرفته اند، اما هنوز دو چالش اساسی وجود دارد. (۱) مدل بهینه‌سازی‌شان دو هدفه بوده و هزینه پولی در آنها در نظر گرفته نشده است. (۲) مهمتر این که زمان اجرای آنها برای محیط‌های بی‌درنگ و با مقیاس بزرگ یک معضل محسوب می‌شود.

به منظور پر کردن این خلاء تحقیقاتی، در این مقاله یک مدل بهینه‌سازی چند هدفه با در نظر گرفتن زمان خاتمه آخرین وظیفه، مصرف انرژی و هزینه پولی ارائه می‌گردد. همچنین، یک الگوریتم ابتکاری کارآمد برای حل این مدل پیشنهاد می‌شود که می‌تواند به خوبی در محیط‌های بی‌درنگ و با مقیاس بزرگ مورد استفاده قرار بگیرد.

### ۳-۲- مدل‌سازی سیستم

در این بخش، ابتدا معماری سیستم در نظر گرفته شده توصیف می‌شود. پس از آن، فرموله‌سازی مسئله در قالب برنامه‌ریزی خطی عدد صحیح مختلط<sup>۱۲</sup> ارائه می‌گردد.

### ۳-۱- معماری سیستم

شکل ۱ معماری سیستم در نظر گرفته شده را نشان می‌دهد. این سیستم از سه لایه تشکیل شده است که شامل مجموعه‌ای از دستگاه‌های اینترنت اشیا به عنوان کاربر نهایی در پایین‌ترین لایه، مجموعه‌ای از گره‌های ناهمگن مه و یک کارگزار در لایه مه و یک مرکز داده ابری در لایه ابر است.

دستگاه‌های اینترنت اشیا درخواست‌های خود را جهت پردازش به لایه بالاتر (لایه مه) ارسال می‌کنند. هر درخواست در قالب یک کار دسته‌ای<sup>۱۳</sup> است که معمولاً به مجموعه‌ای از چندین وظیفه تقسیم می‌شود. این وظیفه‌ها می‌توانند مستقل یا بهم وابسته باشند [۱۹]. به منظور ایجاد قابلیت موازی‌سازی و کاهش زمان پاسخ کارها در سیستم‌های توزیع‌شده و موازی، معمولاً بیشتر کارها در قالب وظیفه‌های مستقل از هم تعریف می‌شوند [۷, ۱۰, ۲۰].

$$\mathcal{M} = \max_{1 \leq i \leq M} [E_i] \quad (3)$$

### ۳-۲-۲- مصرف انرژی

مصرف انرژی کل یک سیستم محاسباتی مه-ابر شامل مجموع مصرف انرژی همه ماشین‌های مجازی آن است. برای هر ماشین مجازی مانند  $v_i$ ، مصرف انرژی‌اش با توجه به حالت آن و مدت زمانی که در آن حالت قرار دارد، قابل محاسبه است. به عبارت دیگر:

$$\xi_i = E_i \times v_i^{active} + (\mathcal{M} - E_i) \times v_i^{idle} \quad \forall i \in V \quad (4)$$

که واحد آن وات ثانیه (یا ژول) می‌باشد<sup>۱۹</sup>.

بر اساس رابطه بالا می‌توان مصرف انرژی کل سیستم جهت اجرای همه وظیفه‌ها را به صورت زیر بیان کرد:

$$\mathcal{E} = \sum_{i=1}^M \xi_i \quad (5)$$

### ۳-۲-۳- هزینه پردازش

هزینه‌های عملیاتی یک سیستم رایانشی می‌تواند شامل موارد متعددی همچون هزینه‌های پردازش، حافظه، پهنای باند شبکه، ذخیره‌سازی، خنک‌سازی و به روز رسانی نرم‌افزارها باشد [۱۹]. با توجه به اینکه تمرکز مقاله حاضر روی پردازش وظیفه‌ها است، بنابراین ما در مدل‌سازی مسئله فقط هزینه پردازش را در نظر می‌گیریم.

وقتی که ماشین مجازی  $v_i$  وظیفه  $\tau_j$  را اجرا می‌کند، هزینه پردازشی آن برابر خواهد بود با:

$$C_{i,j} = E_{i,j} \times v_i^{cost} \quad \forall j \in T, \forall i \in V \quad (6)$$

بنابراین، هزینه پردازش کل سیستم را می‌توان با استفاده از رابطه زیر محاسبه کرد:

$$\mathcal{C} = \sum_{i=1}^M \sum_{j=1}^N X_{i,j} \times C_{i,j} \quad (7)$$

### ۳-۲-۴- تابع هدف

هدف مسئله بهینه‌سازی ما شامل کمینه‌سازی زمان خاتمه آخرین وظیفه، مصرف انرژی و هزینه پردازشی کل سیستم است. با استفاده از رویکرد مجموع وزنی می‌توان یک مسئله بهینه‌سازی چندهدفه را به یک مسئله بهینه‌سازی تک هدفه تبدیل کرد [۲۲].

آن میلیون دستورالعمل به ازای هر ثانیه (MIPS<sup>۱۴</sup>) می‌باشد. برای نشان دادن هزینه پردازشی آن از نماد  $v_i^{cost}$  استفاده می‌کنیم که واحد آن گرید دلار به ازای هر ثانیه (G\$PS<sup>۱۵</sup>) است. همچنین از  $v_i^{active}$  و  $v_i^{idle}$  به ترتیب برای مصرف توان ماشین مجازی  $v_i$  در حالت فعال و حالت بیکاری استفاده می‌نمائیم که واحد آن به وات (W<sup>۱۶</sup>) است.

مشابه [۱۰، ۲۱]، فرض می‌شود که هر کاربر یک کار برای اجرا دارد که شامل مجموعه‌ای از وظیفه‌های سنگین محاسباتی است. این وظیفه‌ها توسط ماشین‌های مجازی پردازش می‌شوند و نتایج آنها به کاربر برگشت داده می‌شود. در این کار، مشابه بسیاری از کارهای گذشته [۷، ۸، ۱۰، ۱۱]، وظیفه‌ها از یکدیگر مستقل در نظر گرفته می‌شوند و می‌توانند به طور موازی اجرا شوند. فرض کنید در طول یک دوره زمانی مشخص  $N$  وظیفه وارد کارگزار شده‌اند که ما آنها را با استفاده از مجموعه  $T = \{\tau_1, \tau_2, \dots, \tau_N\}$  نمایش می‌دهیم. به ازای هر وظیفه  $\tau_j \in T$ ، ما از نماد  $\tau_j^{size}$  برای مشخص کردن اندازه آن وظیفه استفاده می‌کنیم جایی که واحد آن میلیون دستورالعمل (MI<sup>۱۷</sup>) است.

### ۳-۲-۱- زمان خاتمه آخرین وظیفه

زمان تکمیل کلی<sup>۱۸</sup> برای اجرای تمام  $N$  وظیفه معادل زمان خاتمه آخرین وظیفه است. این زمان هر چقدر کمتر باشد بیانگر کارایی بالای سیستم است؛ در نتیجه کیفیت سرویس بالا و رضایت کاربران نهایی را به همراه خواهد داشت.

ما از  $X$  به عنوان ماتریس نگاشت استفاده می‌کنیم.  $X_{i,j} = 1$  به این معنی است که وظیفه  $\tau_j$  به ماشین مجازی  $v_i$  نگاشت شده است،  $\forall j \in T, \forall i \in V$ ؛ در غیر این صورت  $X_{i,j} = 0$ . زمان پردازش وظیفه  $\tau_j$  روی ماشین مجازی  $v_i$  به صورت زیر محاسبه می‌شود:

$$E_{i,j} = \frac{\tau_j^{size}}{v_i^{cpu}} \quad \forall j \in T, \forall i \in V \quad (1)$$

زمان اجرای کل ماشین مجازی  $v_i$  از حاصل جمع زمان اجرای تمام وظیفه‌های اختصاص داده شده به آن ماشین بدست می‌آید. به عبارت ریاضی:

$$E_i = \sum_{j=1}^N X_{i,j} \times E_{i,j} \quad \forall i \in V \quad (2)$$

با در اختیار داشتن رابطه (۲)، می‌توان زمان خاتمه آخرین وظیفه را توسط رابطه زیر بدست آورد:

با رعایت قیدهای (۹)، (۱۰) و (۱۱).

توجه داشته باشید که پس از انجام نرمال‌سازی، تابع هدف ما به بیشینه‌سازی تبدیل می‌شود. مشابه این در [۷] نیز انجام شده است.

#### ۴- الگوریتم پیشنهادی

در این بخش به توصیف الگوریتم پیشنهادی برای حل مسئله بهینه‌سازی زمان‌بندی وظیفه‌ها در محیط رایانشی مه-ابر پرداخته می‌شود. در الگوریتم پیشنهادی سعی می‌شود که عمل نگاشت وظیفه‌ها به ماشین‌های مجازی به گونه‌ای انجام شود که در نهایت هر سه هدف در نظر گرفته شده به صورت وزنی بهینه‌سازی شوند. برای کاهش زمان خاتمه آخرین وظیفه، بهتر است که بین زمان اجرای کل ماشین‌های مجازی، رابطه (۲)، تعادل بار برقرار شود. به منظور کمینه‌سازی مصرف انرژی بهتر است وظیفه‌ها روی ماشین‌هایی اجرا شوند که از لحاظ توان مصرفی کارآمد باشند. در همین راستا، ما معیاری تحت عنوان نرخ پردازش-توان برای هر ماشین مجازی  $v_i$  به صورت زیر تعریف می‌کنیم:

$$v_i^{cpr} = \frac{v_i^{cpu}}{v_i^{active}} \quad \forall i \in V \quad (16)$$

در واقع، این معیار به ماشین‌هایی که از قدرت پردازش بالاتر و توان مصرفی کمتر برخوردارند، اولویت می‌دهد. به طور مشابه، ما معیار دیگری تحت عنوان نرخ پردازش-هزینه به صورت زیر تعریف می‌کنیم:

$$v_i^{ccr} = \frac{v_i^{cpu}}{v_i^{cost}} \quad \forall i \in V \quad (17)$$

که بر اساس آن ماشین‌هایی که قدرت پردازش بالاتری در ازای هزینه پایین‌تر ارائه می‌دهند، مطلوب‌تر تلقی می‌شوند.

شکل ۲ شبه‌کد الگوریتم پیشنهادی را نشان می‌دهد. جزئیات این الگوریتم به شرح زیر است. در خط‌های ۱ تا ۵، ابتدا زمان دسترسی هر ماشین مجازی ( $v_i^a$ ) روی صفر تنظیم می‌شود (خط ۲). در خط ۳ و ۴ به ترتیب نرخ پردازش-توان و پردازش-هزینه ماشین‌های مجازی محاسبه می‌شود. خط ۶ و ۷ به ترتیب ماشین مجازی با بیشترین مقدار نرخ پردازش-توان ( $v_{max}^{cpr}$ ) و پردازش-هزینه ( $v_{max}^{ccr}$ ) را پیدا می‌کند. با استفاده از خط‌های ۸ تا ۱۱،  $v_i^{ncpr}$  (نرخ پردازش-توان نرمال‌سازی شده) و  $v_i^{nccr}$  (نرخ پردازش-هزینه نرمال‌سازی شده) را برای هر ماشین مجازی  $v_i$  بدست می‌آوریم. با توجه به نحوه محاسبه، واضح است که مقادیر

پس می‌توان تابع هدف مسئله بهینه‌سازی را به صورت زیر بیان نمود:

**Minimize:**

$$F = w_1 \times M + w_2 \times E + w_3 \times C \quad (8)$$

با رعایت قیدهای زیر:

$$\sum_{i=1}^M x_{ij} = 1 \quad \forall j \in T \quad (9)$$

$$0 \leq w_k \leq 1 \quad \forall k \in \{1, 2, 3\} \quad (10)$$

$$w_1 + w_2 + w_3 = 1 \quad (11)$$

قید (۹) بیان می‌کند که هر وظیفه می‌تواند فقط روی یک ماشین مجازی قرار بگیرد. قیدهای (۱۰) و (۱۱) به ترتیب بیانگر این هستند که ضریب هر یک از اهداف باید بین صفر و یک و مجموع آنها برابر یک باشد.

با توجه به اینکه مقادیر اهداف از یک جنس نیستند، لازم است که مقادیر آنها نرمال‌سازی شود. برای انجام این کار، ما یک کران پایین برای هر یک از اهداف بهینه‌سازی ارائه می‌دهیم. برای زمان خاتمه آخرین وظیفه، می‌توان از رابطه زیر به عنوان کران پایین استفاده کرد [۷]:

$$M^{min} = \frac{\sum_{j=1}^N \tau_j^{size}}{\sum_{i=1}^M v_i^{cpu}} \quad (12)$$

کران پایین مصرف انرژی زمانی بدست می‌آید که هر وظیفه روی ماشین مجازی اجرا شود که کمترین مصرف انرژی را به همراه داشته باشد. برای این منظور، می‌توان از رابطه زیر بهره گرفت:

$$E^{min} = \sum_{j=1}^N \min_{1 \leq i \leq M} (E_{i,j} \times v_i^{active}) \quad (13)$$

به طور مشابه، اگر همه وظیفه‌ها به ارزان‌ترین ماشین مجازی تخصیص داده شوند آنگاه کران پایینی برای هزینه پردازش بدست می‌آید. پس داریم:

$$C^{min} = \sum_{j=1}^N \min_{1 \leq i \leq M} (E_{i,j} \times v_i^{cost}) \quad (14)$$

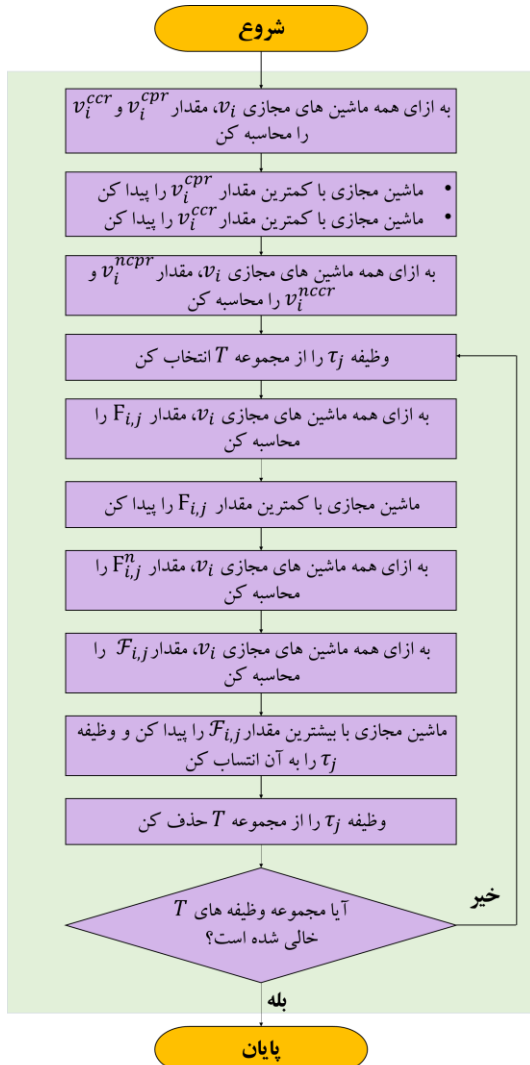
در نهایت، تابع هدف رابطه (۸) را می‌توان به صورت زیر بازنویسی کرد:

**Maximize:**

$$F = w_1 \times \frac{M^{min}}{M} + w_2 \times \frac{E^{min}}{E} + w_3 \times \frac{C^{min}}{C} \quad (15)$$

۱۸). در نهایت، مقدار تابع هدف (خط ۲۱) برای ماشین  $v_i$  ( $F_{i,j}$ ) در مقایسه با سایر ماشین‌ها کمتر خواهد شد. بنابراین، چنین ماشینی توسط الگوریتم انتخاب نخواهد شد.

برای اینکه روش پیشنهادی به راحتی قابل درک باشد، مراحل آن در قالب یک فلوجارت نیز به نمایش گذاشته شده است (شکل ۳ را ببینید).



شکل ۳. فلوجارت الگوریتم پیشنهادی

## ۵- ارزیابی کارایی

در این بخش، ما الگوریتم پیشنهادی را ارزیابی و با الگوریتم‌های تصادفی (به عنوان الگوریتم پایه)، الگوریتم قدرت d-انتخاب [۱۶] و الگوریتم ژنتیک [۷] مقایسه می‌کنیم. اگرچه در [۱۶] نویسندگان الگوریتم قدرت d-انتخاب را برای محیط ابر ارائه داده‌اند، اما کارآمد بودن آن در محیط مه نیز تایید شده است [۹، ۲۳، ۲۴]. هدف اصلی این الگوریتم دستیابی به تعادل بار در سیستم است. بنابراین، در آن وظیفه‌ها به گونه‌ای زمانبندی می‌شوند که

آنها بین ۰ و ۱ خواهد بود. برای یک ماشین مجازی، هر اندازه این مقدار به ۱ نزدیک‌تر باشد بیانگر مطلوبیت آن ماشین است.

**Input:**  $V, T, v_i^{cpu}, v_i^{cost}, v_i^{active}, v_i^{idle}, \tau_j^{size}, w_1, w_2, w_3$

**Output:** Mapping  $T \rightarrow V$

1. **for all**  $v_i \in V$  **do**
2.     set  $v_i^a = 0$ ;
3.     calculate  $v_i^{cpr}$  using eq. (۱۶);
4.     calculate  $v_i^{ccr}$  using eq. (۱۷);
5. **end for**
6. find  $v_i$  with the maximum  $v_i^{cpr}$ ; // say  $v_{max}^{cpr}$
7. find  $v_i$  with the maximum  $v_i^{ccr}$ ; // say  $v_{max}^{ccr}$
8. **for all**  $v_i \in V$  **do**
9.      $v_i^{ncpr} = v_i^{cpr} / v_{max}^{cpr}$ ;
10.     $v_i^{ncr} = v_i^{ccr} / v_{max}^{ccr}$ ;
11. **end for**
12. **for all**  $\tau_j \in T$  **do**
13.     **for all**  $v_i \in V$  **do**
14.          $F_{i,j} = v_i^a + E_{i,j}$ ;
15.     **end for**
16.     find  $v_i$  with the minimum  $F_{i,j}$ ; // say  $F_j^{min}$
17.     **for all**  $v_i \in V$  **do**
18.          $F_{i,j}^n = F_j^{min} / F_{i,j}$ ;
19.     **end for**
20.     **for all**  $v_i \in V$  **do**
21.          $\mathcal{F}_{i,j} = w_1 \times F_{i,j}^n + w_2 \times v_i^{ncpr} + w_3 \times v_i^{ncr}$ ;
22.     **end for**
23.     find  $v_i$  with the maximum  $\mathcal{F}_{i,j}$ ; // say  $v_{index}$
24.      $\tau_j \rightarrow v_{index}$ ;
25.      $v_{index}^a = v_{index}^a + E_{index,j}$ ;
26. **end for**

شکل ۲. شبه‌کد الگوریتم پیشنهادی

پس از تنظیمات و محاسبات اولیه، زمانبندی وظیفه‌ها توسط خط‌های ۱۲ تا ۲۶ انجام می‌گیرد. برای این منظور، به ازای هر وظیفه  $\tau_j$  (خط ۱۲) ابتدا زمان خاتمه آن روی هر ماشین مجازی  $v_i$  ( $F_{i,j}$ ) محاسبه می‌شود (خط ۱۳ تا ۱۵). سپس کمترین مقدار زمان خاتمه برای وظیفه  $\tau_j$  ( $F_j^{min}$ ) را توسط خط ۱۶ پیدا می‌کنیم. در خط‌های ۱۷ تا ۱۹، زمان خاتمه نرمال‌سازی شده برای وظیفه  $\tau_j$  روی تمام ماشین‌های مجازی ( $F_{i,j}^n$ ) محاسبه می‌گردد. پس از انجام این محاسبات، اکنون مقدار برازندگی هر ماشین مجازی به ازای وظیفه  $\tau_j$  ( $\mathcal{F}_{i,j}$ ) را می‌توانیم داشته باشیم (خط-های ۲۰ تا ۲۲). در نهایت، ماشین مجازی با بیشترین مقدار برازندگی، به عبارت دیگر مناسب‌ترین ماشین، را پیدا کرده (خط ۲۳) و وظیفه  $\tau_j$  را به آن اختصاص می‌دهیم (خط ۲۴). پس از تخصیص، زمان دسترسی آن ماشین را با استفاده از خط ۲۵ به روز می‌کنیم. لازم به ذکر است چنانچه ماشین مجازی  $v_i$  تعداد وظیفه زیادی در صف داشته باشد در حالی که ماشین‌های دیگر خلوت‌تر باشند، الگوریتم پیشنهادی با احتمال بسیار پایینی وظیفه  $\tau_j$  را به آن تخصیص می‌دهد. زیرا اگر وظیفه  $\tau_j$  به آن اختصاص داده شود، مقدار زمان خاتمه آن وظیفه ( $\mathcal{F}_{i,j}$ ) بسیار بالا خواهد بود (خط ۱۴)، در نتیجه مقدار  $F_{i,j}^n$  به شدت کاهش می‌یابد (خط

## ۵-۲-۱- تحلیل حساسیت پارامترهای الگوریتم پیشنهادی

الگوریتم پیشنهادی ما به سه پارامتر  $w_1$ ،  $w_2$  و  $w_3$  وابسته است که به ترتیب اولویت بین زمان خاتمه آخرین وظیفه، نرخ پردازش-توان و نرخ پردازش-هزینه را مشخص می‌کنند (خط ۲۱ الگوریتم پیشنهادی). تعیین مقدار بهینه برای آنها یک امر چالش‌برانگیز است. به منظور تنظیم دقیق این پارامترها، لازم است که اهمیت نیازمندی‌های مختلف کاربران و تأمین‌کنندگان سرویس‌های مه-ابر شامل کارایی بالا و کارآمدی هزینه مشخص گردد. الگوریتم پیشنهادی از انعطاف‌پذیری بسیار خوبی جهت انتخاب این نیازمندی‌ها برخوردار است.

جدول ۲. تحلیل حساسیت پارامترهای الگوریتم پیشنهادی.

$M$ : زمان خاتمه آخرین وظیفه؛  $E$ : انرژی؛  $C$ : هزینه؛  $F$ : تابع هدف.

$w_1 = 0.25$	$w_1 = 0.25$	$w_1 = 0.5$	$w_1 = 0.34$	
$w_2 = 0.25$	$w_2 = 0.5$	$w_2 = 0.25$	$w_2 = 0.33$	
$w_3 = 0.5$	$w_3 = 0.25$	$w_3 = 0.25$	$w_3 = 0.33$	
۸۶/۰۸	۴۹/۳۲	۲۰/۱۶	۵۰/۹۲	$M$
۱۵۷۸۵۷/۲۰	۱۰۱۲۸۲/۳۹	۴۵۷۲۲/۰۵	۹۷۵۵۷/۸۰	$E$
۷۳/۲۹	۸۱/۸۷	۹۵/۹۰	۸۶/۵۰	$C$
۰/۳۹	۰/۲۹	۰/۵۰	۰/۲۷	$F$

به منظور بررسی تأثیر این ضریب‌ها، ما چهار سناریوی مختلف را در نظر می‌گیریم. در سناریو اول، اهمیت یکسانی به هر سه معیار می‌دهیم. در حالی که برای سناریوی دوم اهمیت بیشتری به زمان خاتمه آخرین وظیفه، برای سناریوی سوم اهمیت بیشتری به انرژی و در سناریوی چهارم اهمیت بیشتری به هزینه می‌دهیم. سپس الگوریتم پیشنهادی را برای مجموعه‌ای شامل ۱۰۰ وظیفه اجرا می‌کنیم. نتایج بدست آمده در جدول ۲ گزارش شده است. همان‌طور که نتایج نشان می‌دهد در حالتی که به زمان خاتمه آخرین وظیفه نسبت به نرخ پردازش-توان و نرخ پردازش-هزینه اهمیت بیشتری داده شده است (یعنی سناریوی دوم)، مقادیر زمان خاتمه آخرین وظیفه و مصرف انرژی در مقایسه با بقیه حالت‌ها به طرز چشمگیری کمتر است. همچنین، هزینه پردازش آن نیز در مقایسه با سه حالت دیگر قابل قبول است. در نتیجه، برای این حالت تابع هدف بیشترین مقدار را در اختیار ما قرار می‌دهد. برای توجیه این رفتار لازم است به این نکته اشاره کنیم که کاهش زمان خاتمه آخرین وظیفه تأثیر مستقیم روی کاهش زمان بیکاری ماشین‌های مجازی دارد. بنابراین، با توجه به اینکه ماشین‌های

بین‌گره‌های محاسباتی سیستم تعادل برقرار شود.

## ۵-۱- تنظیمات شبیه‌سازی

برای محیط رایانشی مه-ابر، ما ۲۵ ماشین مجازی ناهمگن در نظر می‌گیریم که ۲۰ تای آنها در محیط مه و ۵ تای دیگر در محیط ابر قرار دارند. برای بررسی تأثیر تعداد وظیفه‌ها، ما تعداد آنها را از ۵۰ تا ۲۵۰ تغییر می‌دهیم (با گام‌های ۵۰ تایی). جدول ۱ مشخصات محیط رایانشی را نشان می‌دهد. مشابه [۷]، سرعت پردازش ماشین‌های مجازی مستقر در محیط ابر بیشتر از ماشین‌های مجازی محیط مه فرض شده است. در مقابل، هزینه استفاده از منابع ابر گران‌تر از منابع مه است. همچنین، ماشین‌های محیط ابر مصرف توان بالاتری نیز دارند. برای هر ماشین مجازی، مصرف توان آن در حالت بیکاری معادل ۶۰٪ مصرف توان آن در حالت فعال است [۲۵]. اندازه هر وظیفه را نیز به طور تصادفی از بازه [۱۰ - ۱۰۰۰۰] انتخاب می‌کنیم. با توجه به اینکه هنوز مجموعه داده واقعی برای مسئله زمان‌بندی وظیفه‌ها در محیط مه-ابر در دسترس نیست [۷]، ما از مجموعه داده تصادفی برای انجام شبیه‌سازی استفاده می‌کنیم که به ما کمک می‌کند تا بتوانیم سناریوهای متنوعی را پوشش دهیم.

جدول ۱. مشخصات محیط رایانشی مه-ابر

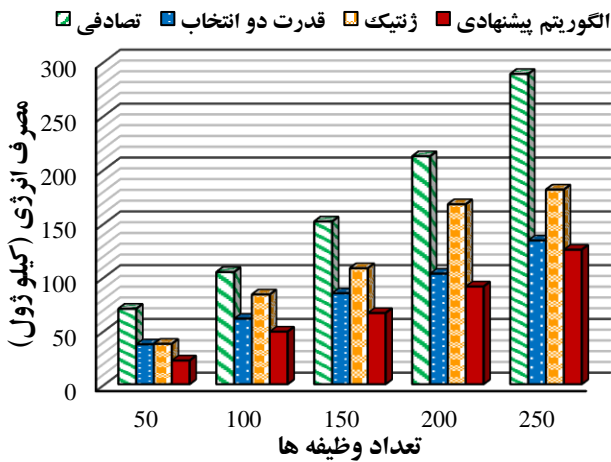
پارامترها	محیط مه	محیط ابر
سرعت پردازش	[۵۰۰ - ۱۵۰۰]	[۳۰۰۰ - ۵۰۰۰]
هزینه پردازش	[۰/۱ - ۰/۴]	[۰/۷ - ۱]
مصرف توان در حالت فعال	[۴۰ - ۱۰۰]	[۲۰۰ - ۴۰۰]
مصرف توان در حالت بیکاری	۶۰٪ حالت فعال	۶۰٪ حالت فعال

آزمایش‌ها با استفاده از زبان برنامه‌نویسی ++C در محیط Dev- Cpp 5.11 انجام شده است. به منظور افزایش قابلیت اطمینان نتایج، هر آزمایش 100 بار تکرار شده و میانگین آنها گزارش گردیده است. تمامی اجراها روی یک کامپیوتر شخصی Intel® Core i7-6600U CPU @ 2.6 GHz (چهار هسته)، 16.00 GB RAM و سیستم عامل Windows 10 گرفته شده است. سورس‌کد مقاله از طریق [۲۶] قابل دسترس است.

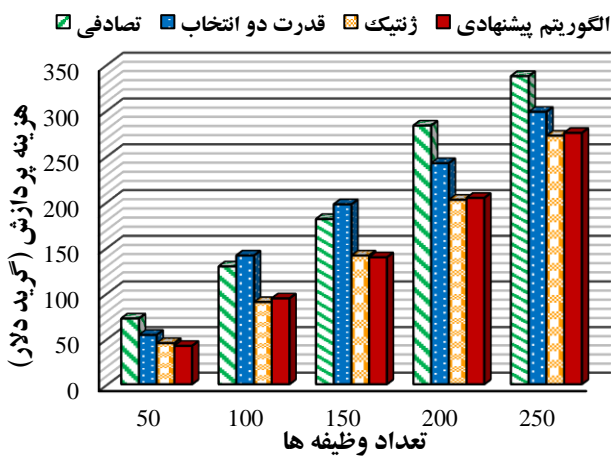
## ۵-۲- نتایج

در این زیربخش، نتایج حاصل از پیاده‌سازی گزارش خواهد شد. برای این منظور، ابتدا نتایج مربوط به تحلیل حساسیت پارامترهای دخیل در الگوریتم پیشنهادی را ارائه می‌دهیم. سپس نتایج مربوط به مقایسه الگوریتم پیشنهادی با سایر الگوریتم‌ها را بررسی می‌کنیم.





شکل ۵. مقایسه مصرف انرژی



شکل ۶. مقایسه هزینه پردازش

شکل ۵ و شکل ۶ به ترتیب نتایج مربوط به مصرف انرژی و هزینه پردازش سیستم رایانشی در نظر گرفته شده را نشان می‌دهند. از شکل‌ها به طور واضح می‌توان به این نکته پی برد که به تناسب افزایش تعداد وظیفه‌ها، مصرف انرژی و هزینه پردازش کل سیستم نیز برای همه الگوریتم‌ها افزایش پیدا می‌کند. از نظر مصرف انرژی (شکل ۵)، الگوریتم پیشنهادی به مراتب از بقیه بهتر عمل می‌کند. این برتری از آنجا ناشی می‌شود که هنگام نگاشت وظیفه‌ها به ماشین‌های مجازی، هیچ کدام از الگوریتم‌های مورد مقایسه پارامتر انرژی را در نظر نمی‌گیرند. در حالی که الگوریتم پیشنهادی این پارامتر مهم را نیز مدنظر قرار می‌دهد و سعی می‌کند به ماشین‌های مجازی که از لحاظ انرژی کارآمدتر هستند، اولویت بیشتری بدهد. در اینجا میزان برتری الگوریتم پیشنهادی در مقایسه با الگوریتم‌های تصادفی، ژنتیک و قدرت دو انتخاب به ترتیب تا ۶۸٪، ۴۶٪ و ۴۱٪ است.

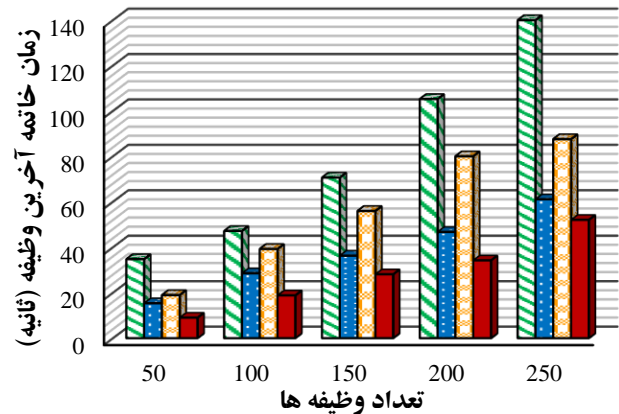
از نظر هزینه پردازش (شکل ۶)، عملکرد الگوریتم پیشنهادی قابلیت رقابت با الگوریتم ژنتیک را دارد اما از برتری چشمگیری

مجازی در حالت بیکاری نیز حدود ۶۰٪ حالت فعال انرژی مصرف می‌کند [۲۵]. کم کردن زمان بیکاری ماشین‌های مجازی با کاهش مصرف انرژی سیستم رایانشی همراه خواهد بود.

#### ۲-۲-۵- مقایسه الگوریتم پیشنهادی با سایر الگوریتم‌ها

برای الگوریتم پیشنهادی، پارامترها را به صورت  $w_1 = 0.5$ ,  $w_2 = 0.25$ ,  $w_3 = 0.25$  تنظیم کرده‌ایم. همچنین برای الگوریتم قدرت d-انتخاب [۱۶] مقدار  $d=2$  در نظر گرفته‌ایم و در نمودارها از آن به عنوان قدرت دو انتخاب یاد می‌کنیم. برای الگوریتم ژنتیک نیز از تنظیمات خود مقاله مرجع استفاده شده است [۷].

شکل ۴. مقایسه زمان خاتمه آخرین وظیفه



شکل ۴. مقایسه زمان خاتمه آخرین وظیفه

شکل ۴ نتایج مربوط به زمان خاتمه آخرین وظیفه چهار الگوریتم را نشان می‌دهد. با افزایش تعداد وظیفه‌ها بار بیشتری به سیستم تحمیل می‌شود. در نتیجه، زمان خاتمه آخرین وظیفه برای همه الگوریتم‌ها تقریباً به صورت خطی افزایش می‌یابد. اما، الگوریتم پیشنهادی بهترین عملکرد را در مقایسه با سایر الگوریتم‌ها از خود نشان می‌دهد. در مقایسه با روش تصادفی، الگوریتم پیشنهادی مقدار زمان خاتمه آخرین وظیفه را بین ۶۰٪ تا ۷۴٪ بهبود می‌دهد و در مقایسه با روش قدرت دو انتخاب و الگوریتم ژنتیک، مقدار بهبود به ترتیب بین ۱۵٪ تا ۴۱٪ و ۴۰٪ تا ۵۷٪ است. دلیل اصلی این برتری به خاطر در نظر گرفتن اولویت بیشتر برای زمان خاتمه آخرین وظیفه توسط الگوریتم پیشنهادی است. بایستی یادآوری کنیم که بر اساس آزمایش‌های مربوط به تحلیل حساسیت پارامترها (زیربخش ۰)، ضریب زمان خاتمه آخرین وظیفه در الگوریتم پیشنهادی برابر ۰/۵ در نظر گرفته شده است که این مقدار دو برابر ضریب انرژی و هزینه است.

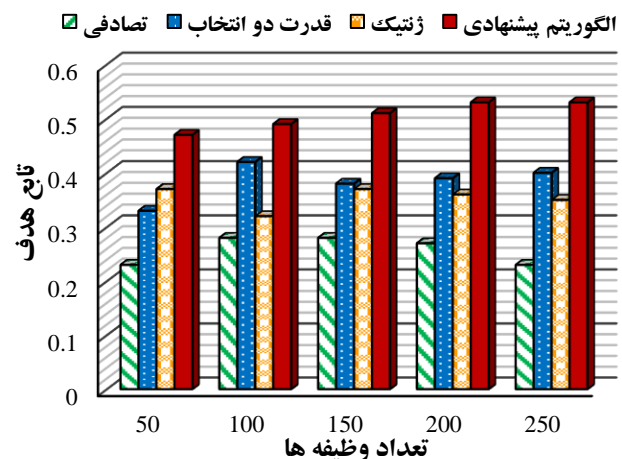
## ۶- نتیجه‌گیری و کارهای آینده

در این مقاله، ما مسئله زمان‌بندی وظیفه‌ها در محیط یکپارچه مه-ابر را مطالعه کردیم. ما یک مدل چندهدفه برای بهینه‌سازی توأم زمان خاتمه آخرین وظیفه، مصرف انرژی و هزینه پردازش وظیفه-ها ارائه دادیم. به منظور حل مدل، ما یک الگوریتم ابتکاری کارآمد پیشنهاد دادیم و آن را با الگوریتم‌های تصادفی، ژنتیک و قدرت دو انتخاب از جنبه‌های مختلف مقایسه نمودیم. نتایج حاصل از پیاده‌سازی نشان داد که الگوریتم پیشنهادی ما به طور چشمگیری نسبت به رقبانش بهتر عمل می‌کند. علاوه بر این، الگوریتم پیشنهادی از انعطاف‌پذیری بسیار خوبی جهت دستیابی به نیازمندی‌های مختلف تأمین‌کنندگان سرویس‌های مه-ابر و کاربران اینترنت اشیا برخوردار است. به عنوان کار آینده ما قصد داریم که محیط مه را به صورت یک شبکه مدل کنیم و پارامترهای آن مانند تأخیر انتقال و محدودیت پهنای باند را نیز در مدل‌سازی مسئله مدنظر قرار دهیم. به منظور ارزیابی بیشتر کارآمدی الگوریتم پیشنهادی، ما علاقمند هستیم که کارایی الگوریتم ارائه شده را تحت مجموعه داده‌های واقعی و با استفاده از چارچوب‌های شناخته شده محیط مه مانند iFogSim [۲۷] ارزیابی کنیم.

## مراجع

- [1] Cisco. "The future of IoT miniguide: The burgeoning IoT market continues." <https://www.cisco.com/c/en/us/solutions/internet-of-things/future-of-iot.html> (accessed December 2020).
- [2] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future generation computer systems*, vol. 56, pp. 684-700, 2016.
- [3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13-16.
- [4] P. Hosseinioun, M. Kheirabadi, S. R. Kamel Tabbakh, and R. Ghaemi, "aTask scheduling approaches in fog computing: A survey," *Transactions on Emerging Telecommunications Technologies*, p. e3792, 2020.
- [5] M. R. Alizadeh, V. Khajehvand, A. M. Rahmani, and E. Akbari, "Task scheduling approaches in fog computing: A systematic review," *International Journal of Communication Systems*, vol. 33, no. 16, p. e4583, 2020.
- [6] S. K. Mishra, D. Puthal, J. J. Rodrigues, B. Sahoo, and E. Dutkiewicz, "Sustainable service allocation using a metaheuristic technique in a fog server for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4497-4506, 2018.
- [7] B. M. Nguyen, H. Thi Thanh Binh, and B. Do Son, "Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud-fog computing environment," *Applied Sciences*, vol. 9, no. 9, p. 1730, 2019.
- [8] M. Abdel-Basset, D. El-shahat, M. Elhoseny, and H. Song, "Energy-Aware Metaheuristic algorithm for Industrial Internet of

نسبت به الگوریتم‌های تصادفی و قدرت دو انتخاب برخوردار است. برای توجیه این رفتار بایستی به این نکته مهم اشاره شود که هنگام نگاشت وظیفه‌ها به ماشین‌های مجازی، الگوریتم پیشنهادی به ماشین‌هایی که از لحاظ هزینه پردازش به صرفه‌تر هستند نسبت به بقیه اولویت بیشتری می‌دهد. در حالی که الگوریتم تصادفی و الگوریتم قدرت دو انتخاب این معیار را نادیده می‌گیرند که در نهایت منجر به افزایش هزینه پردازش می‌گردد. شایان ذکر است که یکی از اهداف اصلی الگوریتم ژنتیک ارائه شده در [۷] کاهش هزینه پردازش است.



شکل ۷. مقایسه تابع هدف

شکل ۷ توازن بین زمان خاتمه آخرین وظیفه، مصرف انرژی و هزینه پردازش الگوریتم‌ها را به نمایش گذاشته است. همان گونه که می‌توان مشاهده کرد الگوریتم پیشنهادی از این نظر نیز عملکرد بسیار مطلوبی از خود نشان می‌دهد و برتری چشمگیری نسبت به بقیه دارد. دلیل اصلی این برتری به خاطر این است که الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها تنها روشی است که تلاش می‌کند زمان خاتمه آخرین وظیفه، مصرف انرژی و هزینه پردازش سیستم را به طور توأم بهینه کند. مقدار تابع هدف برای الگوریتم پیشنهادی بین ۰/۴۷ تا ۰/۵۳ است در حالی که برای الگوریتم قدرت دو انتخاب مقدار آن بین ۰/۳۳ تا ۰/۴۲، برای الگوریتم ژنتیک بین ۰/۳۲ تا ۰/۳۷ و برای الگوریتم تصادفی بین ۰/۲۳ تا ۰/۲۸ است. به عبارت دیگر، الگوریتم پیشنهادی به طور میانگین ۳۲٪ بهتر از روش قدرت دو انتخاب، ۴۳٪ بهتر از الگوریتم ژنتیک و ۹۸٪ بهتر از روش تصادفی عمل می‌کند. در نتیجه، می‌توان گفت که الگوریتم پیشنهادی به خوبی می‌تواند بین هدف‌های زمان خاتمه آخرین وظیفه، مصرف انرژی و هزینه پردازش سیستم تعادل برقرار کند.

- [26] S. Azizi. "A Multi-objective Model for Task Scheduling Optimization in Fog-Cloud Computing Environments." <https://github.com/S-Azizi/Sourcecodes/blob/main/Makespan-Energy-Cost.rar>.
- [27] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275-1296, 2017.
- Things task scheduling problems in fog computing applications," *IEEE Internet of Things Journal*, 2020.
- [9] F. Hoseiny, S. Azizi, and S. Dabiri, "Using the Power of Two Choices for Real-Time Task Scheduling in Fog-Cloud Computing," in *2020 4th International Conference on Smart City, Internet of Things and Applications (SCIOT)*, 2020: IEEE, pp. 18-23.
- [10] S. Ghanavati, J. H. Abawajy, and D. Izadi, "An Energy Aware Task Scheduling Model Using Ant-Mating Optimization in Fog Computing Environment," *IEEE Transactions on Services Computing*, 2020.
- [11] S. Bitam, S. Zeadally, and A. Mellouk, "Fog computing job scheduling optimization based on bees swarm," *Enterprise Information Systems*, vol. 12, no. 4, pp. 373-397, 2018.
- [12] N. Auluck, A. Azim, and K. Fizza, "Improving the schedulability of real-time tasks using fog computing," *IEEE Transactions on Services Computing*, 2019.
- [13] R. O. Aburukba, M. AliKarrar, T. Landolsi, and K. El-Fakih, "Scheduling Internet of Things requests to minimize latency in hybrid Fog-Cloud computing," *Future Generation Computer Systems*, vol. 111, pp. 539-551, 2020.
- [14] S. Javanmardi, M. Shojafar, V. Persico, and A. Pescape, "FPFTS: A Joint Fuzzy PSO Mobility-aware Approach to Fog Task Scheduling Algorithm for IoT Devices," *Software Practice and Experience*, 2020.
- [15] Z. Zhou, H. Xie, and F. Li, "A novel task scheduling algorithm integrated with priority and greedy strategy in cloud computing," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 4, pp. 4647-4655, 2019.
- [16] A. Moaddeli, I. N. Ahmadi, and N. Abhar, "The Power of d Choices in Scheduling for Data Centers with Heterogeneous Servers," *arXiv preprint arXiv:1904.00447*, 2019.
- [17] F. Hoseiny, S. Azizi, M. Shojafar, and R. Tafazolli, "Joint QoS-aware and Cost-efficient Task Scheduling for Fog-Cloud Resources in a Volunteer Computing System," *ACM Transaction on Internet Technology*, vol. 21, no. 4, pp. 1-21, 2021.
- [18] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE internet of things journal*, vol. 3, no. 6, pp. 1171-1181, 2016.
- [19] B. Wang, C. Wang, Y. Song, J. Cao, X. Cui, and L. Zhang, "A survey and taxonomy on workload scheduling and resource provisioning in hybrid clouds," *Cluster Computing*, pp. 1-26, 2020.
- [20] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.
- [21] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397-1411, 2016.
- [22] Y. Donoso and R. Fabregat, *Multi-objective optimization in computer networks using metaheuristics*. CRC Press, 2016.
- [23] R. Beraldi, H. Alnuweiri, and A. Mtibaa, "A power-of-two choices based algorithm for fog computing," *IEEE Transactions on Cloud Computing*, vol. 8, no. 3, pp. 698-709, 2018.
- [24] R. Beraldi and G. P. Mattia, "Power of random choices made efficient for fog computing," *IEEE Transactions on Cloud Computing*, 2020.
- [25] A. M. Sampaio, J. G. Barbosa, and R. Prodan, "PIASA: A power and interference aware resource management strategy for heterogeneous workloads in cloud data centers," *Simulation Modelling Practice and Theory*, vol. 57, pp. 142-160, 2015.

### پاورقی‌ها:

---

Makespan	<sup>۱</sup>
Monetary Cost	<sup>۲</sup>
Heuristic	<sup>۳</sup>
Computation-Power Ratio (CPR)	<sup>۴</sup>
Computation-Cost Ratio (CCR)	<sup>۵</sup>
Bee Life Algorithm (BLA)	<sup>۶</sup>
The Power of <i>d</i> Choices	<sup>۷</sup>
Metaheuristic	<sup>۸</sup>
BAT	<sup>۹</sup>
Ant-Mating Optimization (AMO)	<sup>۱۰</sup>
Harris Hawks Optimization (HHO)	<sup>۱۱</sup>
Mixed Integer Linear Programming (MILP)	<sup>۱۲</sup>
Batch Job	<sup>۱۳</sup>
Million Instructions Per Second	<sup>۱۴</sup>
Grid Dollar Per Second	<sup>۱۵</sup>
Watt	<sup>۱۶</sup>
Million Instructions	<sup>۱۷</sup>
Overall Completion Time	<sup>۱۸</sup>
هر وات ثانیه معادل یک ژول است.	<sup>۱۹</sup>