

زمان بندی کاربردهای جریان کاری علمی در محیط چندابری با استفاده از الگوریتم جستجوی فاخته

سمیه محمدی، لطیف پور کریمی و سمیه عبدی

قابلیت کشسانی منابع می شود. تا کنون اتحادیه ابر^۲ و چندابری^۳ به عنوان دو مدل اصلی میان- ابر مطرح شده اند.

دانشمندان روزانه در طی تحقیقات علمی خود با برنامه ها و داده هایی سروکار دارند که بزرگ تر از آن هستند که خود بتوانند از عهده مدیریتشان برآیند [۱]. کاربردهای جریان کاری علمی^۴ در جهت کمک به دانشمندان برای مدیریت تمام مراحل مربوط به آزمایش های بسیار بزرگ، مورد توجه قرار گرفته اند [۲] و [۳]. مسایل مختلف علمی مانند ستاره شناسی، زلزله نگاری، ژنتیک، بیولوژی و فیزیک گرانشی به صورت جریان کاری مدل شده اند. اکثر این کاربردهای جریان کاری علمی مقیاس- وسیع هستند و به محیط های محاسباتی با قدرت بالا و تکنیک های موازی برای اجرا شدن در یک زمان معقول نیاز دارند [۲] تا [۴]. با ظهور تکنولوژی محاسبات ابری، در چند سال اخیر دانشمندان با اجرای پردازش های مربوط به آزمایش هایشان در این محیط توانسته اند بر مسایل زیرساختی غلبه کنند؛ جایی که منابع محاسباتی از طریق اینترنت، بر حسب تقاضا و پرداخت به اندازه استفاده برای کاربران ارائه می شود [۵] و [۶].

در محیط میان- ابر فراهم کنندگان مختلف انواع بسیار زیادی از نمونه های ماشین مجازی را ارائه می دهند که از لحاظ سیاست های قیمت گذاری، فاصله زمانی پرداخت، قیمت انواع نمونه های مختلف و غیره با هم تفاوت دارند که این گوناگونی منابع و تعداد بیشتر آنها نه تنها قدرت محاسباتی بیشتری نسبت به یک فراهم کننده تنها را در اختیار کاربر قرار می دهد بلکه می تواند باعث کاهش هزینه ها و اجرای سریع تر برنامه ها نیز گردد. اگرچه این تنوع و گوناگونی منابع باعث می شود برای کاربر نهایی انتخاب نمونه های مناسب برای اجرای کاربردهای مورد نظر بسیار مشکل گردد. بنابراین لازم است برای زمان بندی کاربردهای جریان کاری در محیط میان- ابر از یک الگوریتم کارآمد استفاده شود.

حل مسایل بهینه سازی از طریق جستجوی تمام جواب های امکان پذیر (فضای جستجو) و محاسبه توابع هدف بسیار زمان بر است. اخیراً تمرکز بیشتری بر روش های فراابتکاری صورت گرفته است که در آنها به جای بررسی تمام جواب ها بخشی از آنها را در نظر می گیرند. روش های جستجوی فراابتکاری روش هایی هستند که می توانند جواب خوبی (نزدیک به بهینه) در زمانی معقول برای مسأله ارائه کنند. زمان بندی کاربردهای جریان کاری یک شکل کلی از مسأله بهینه سازی زمان بندی وظایف است که از نظر پیچیدگی زمانی جزء مسایل NP-Hard است [۷]. در این تحقیق با به کارگیری یکی از جدیدترین روش های جستجوی فراابتکاری

چکیده: محیط های چندابری شامل منابع متنوع قابل ملاحظه ای هستند که هزینه های زمان بندی کاربردهای جریان کاری در چنین محیط هایی می تواند به طور چشم گیری کاهش یابد و همچنین محدودیت ارائه منابع توسط فراهم کنندگان تجاری ابر رفع شود. بر این اساس، این تحقیق به مسأله زمان بندی کاربردهای جریان کاری علمی در محیط چندابری تحت قید مهلت زمانی با هدف کمینه سازی هزینه می پردازد. در این مقاله با به کارگیری الگوریتم جستجوی فاخته که یکی از مشهورترین روش های جستجوی فراابتکاری می باشد، الگوریتمی برای مسأله زمان بندی کاربردهای جریان کاری در محیط چندابری ارائه شده است. الگوریتم فراابتکاری جستجوی فاخته قادر است در مدت زمانی کوتاه فضای جواب را جستجو نموده و جواب هایی را در همسایگی جواب بهینه سراسری بیابد که به آن نزدیک می باشد. نتایج به دست آمده نشان می دهند که راهکار پیشنهادی این تحقیق در مقایسه با دیگر راهکارهای فراابتکاری در موارد کاهش هزینه کارایی بهتری داشته و همچنین جواب های به دست آمده از الگوریتم فراابتکاری پیشنهادی، در حد مطلوبی نزدیک به جواب های بهینه سراسری به دست آمده از مدل ریاضی است.

کلیدواژه: چندابری، زمان بندی، جریان کاری علمی، بهینه سازی هزینه، الگوریتم جستجوی فاخته.

۱- مقدمه

در محاسبات ابری معمولاً کاربران از منابع و سرویس های یک فراهم کننده استفاده می کنند. از آنجایی که هر فراهم کننده منابع فیزیکی محدودی دارد، ممکن است در ساعات اوج بار کاری به خاطر کمبود منابع درخواست کاربران جدید را رد کند و یا کیفیت سرویس های در حال ارائه کاهش یابد. همچنین ممکن است منابع مورد نیاز کاربر بیش از آنچه باشد که یک فراهم کننده به خاطر محدودیت های واگذاری می تواند ارائه دهد. بعد از ظهور محاسبات ابری، ایده اتصال ابرها که میان- ابر^۱ نامیده می شود برای افزایش کیفیت ارائه سرویس ها مطرح شد. میان- ابر ایده تجمیع چندین فراهم کننده مختلف است که باعث افزایش قابلیت اعتماد و

این مقاله در تاریخ ۱۹ مهر ماه ۱۳۹۸ دریافت و در تاریخ ۲۸ آبان ماه ۱۳۹۹ بازنگری شد.

سمیه محمدی (نویسنده مسئول)، گروه مهندسی کامپیوتر، واحد کرمانشاه، دانشگاه آزاد اسلامی، کرمانشاه، ایران، (email: s.mohamadi@srbiau.ac.ir).
لطیف پور کریمی، گروه ریاضی، دانشکده علوم پایه، دانشگاه رازی، کرمانشاه، ایران، (email: l.pourkarimi@razi.ac.ir).

سمیه عبدی، گروه مهندسی کامپیوتر، واحد اسلام آباد غرب، دانشگاه آزاد اسلامی، اسلام آباد غرب، ایران، (email: somayeh.abdi@gmail.com).

1. Inter-Cloud

2. Cloud Federation

3. Multi-Cloud

4. Scientific Workflows

فراهم‌کنندگان برای ارائه منابع به هر مشتری محدودیت اعمال می‌کنند. در [۱۲] مؤلفان یک مکانیسم قابل اعتماد برای زمان‌بندی جریان‌های کاری داده- حجیم^۶ ارائه داده‌اند. آنها دو هدف هزینه مالی و زمان اتمام کارها را در یک محیط چندابری تجاری در نظر گرفته و رهیافت خود را طوری طراحی کرده‌اند که برای زمان‌بندی کاربردهای جریان کاری به صورت پویا مناسب باشد. کار آنها از مزیت کشسانی منابع ابری بهره برده است اما طبیعت ناهمگن بودن ابر را نادیده گرفته و تنها یک نمونه منبع در یک ابر معین را در نظر گرفته است. به علاوه، هدف آنها بهینه‌سازی هزینه و زمان اتمام بوده در حالی که روی هزینه اجرای جریان کاری با برآورده شدن مهلت‌های زمانی معین تمرکز کرده‌اند.

در [۱۳] ما یک مدل ریاضی خطی صحیح به نام ILP-DC برای مسأله زمان‌بندی کاربردهای جریان کاری علمی با هدف بهینه‌سازی هزینه در محیط چندابری ارائه کرده‌ایم که قادر به یافتن جواب‌های بهینه سراسری است. این کار محدودیت ارائه منابع در محیط ابر را در نظر گرفته است.

بسیاری از پژوهش‌های اخیر برای حل مسأله زمان‌بندی کاربردهای جریان کاری حجیم- داده از روش‌های فراابتکاری استفاده کرده‌اند. نویسندگان [۱۴] از روش فراابتکاری ازدحام ذرات بهره برده‌اند در حالی که در [۱۵] از روش فراابتکاری ژنتیک استفاده شده است. در [۱۶] نیز از راهکاری مبتنی بر یک روش ابتکاری که نویسندگان ارائه داده‌اند استفاده شده است.

۳- مدل کاربرد جریان کاری

خصوصیات و ساختار برنامه کاربردی بر زمان‌بندی آن تأثیرگذار است. در این تحقیق کاربردهایی با مدل جریان کاری مورد نظر است. هر کاربرد جریان کاری با یک گراف بدون دور جهت‌دار^۷ مدل می‌شود. در این گراف، گره‌ها نشان‌دهنده وظایف کاربرد هستند در حالی که یال‌های جهت‌دار محدودیت تقدم و وابستگی داده‌ای بین وظایف را نشان می‌دهند.

به طور خاص در این تحقیق کاربردهای جریان کاری علمی شناخته شده Montage [۱۷]، CyberShake [۱۸]، Epigenomics [۱۹] و LIGO [۲۰] که به ترتیب در علم نجوم، زلزله‌نگاری، ژنتیک و فیزیک گرانشی کاربرد دارند در نظر گرفته می‌شوند. شکل ۱ ساختار کلی این چهار کاربرد جریان کاری را نشان می‌دهد به طوری که در آنها وظایف هر سطح رنگ یکسان دارند و از یک نوع می‌باشند.

به طور ایده‌آل هدف زمان‌بندی در محیط محاسبات ابری تعیین یک تخصیص بهینه برای وظایف مختلف کاربردها به نمونه ماشین‌های مجازی متفاوت از ابرهاست. از طرف دیگر، در مسایل واقعی مثل زمان‌بندی کاربردهای جریان کاری در محیط چندابری، به دلیل زیادبودن تعداد وظایف کاربرد و تنوع ماشین‌های مجازی، یافتن چنین تخصیصی بسیار زمان‌بر یا غیر ممکن است. این چالش می‌تواند با بخش‌بندی وظایف کاربرد و زمان‌بندی ترتیبی بخش‌ها مرتفع شود. با استفاده از این ایده، وظایف هر سطح از گراف را در یک بخش دسته‌بندی می‌کنیم. از آنجایی که در کاربردهای جریان کاری مورد نظر این تحقیق همه وظایف یک بخش از یک نوع یکسان هستند، هر بخش را یک دسته وظایف می‌نامیم.

که الگوریتم جستجوی فاخته^۱ نام دارد، الگوریتمی برای یافتن جواب بهینه مسأله زمان‌بندی کاربردهای جریان کاری در محیط چندابری ارائه شده است. الگوریتم فراابتکاری جستجوی فاخته قادر است در مدت زمانی کوتاه فضای جواب را جستجو نموده و جواب‌هایی را در همسایگی جواب دقیق بیابد که به جواب دقیق نزدیک می‌باشد.

ساختار ادامه مقاله به این صورت می‌باشد: در بخش ۲ تعدادی از کارهای مطرح که در این زمینه انجام شده است مورد بررسی قرار می‌گیرد. در بخش ۳ مدل برنامه کاربردی توصیف می‌شود. در بخش ۴ روش پیشنهادی برای مسأله مورد نظر به همراه جزئیات آن ارائه می‌شود. شبیه‌سازی و نتایج به دست آمده در بخش ۵ بیان می‌شوند و در نهایت بخش ۶ به نتیجه‌گیری و کارهای آتی می‌پردازد.

۲- کارهای مرتبط

زمان‌بندی یک کاربرد جریان کاری می‌تواند به صورت تک‌هدفی یا چندهدفی باشد. از میان کارهای انجام شده با یک هدف مربوط به زمان‌بندی کاربردهای جریان کاری تحت قید کیفیت سرویس در سیستم‌های ابری، بهینه‌سازی هزینه تحت قید مهلت زمانی^۲ و بهینه‌سازی زمان اتمام تحت قید بودجه^۳ دو تا از مهم‌ترین گروه‌ها به حساب می‌آیند [۷].

در [۸] مؤلفان یک روش بخش‌بندی وظیفه بر اساس مسیر بحرانی جزئی^۴ (PCP) ارائه کرده‌اند که توسعه کار قبلی‌شان [۹] برای محیط گرید است. آنها دو الگوریتم به نام‌های IC-PCP و IC-PCPD₂ ارائه داده‌اند. IC-PCPD₂ مثل PCP شامل دو فاز است که ابتدا مهلت زمانی را روی وظایف توزیع می‌کند و سپس هر وظیفه را روی ارزان‌ترین منبع زمان‌بندی می‌کند به طوری که بتواند قبل از زیرمهلت زمانی^۵ اجرا شود. IC-PCP یک فاز دارد و از یک سیاست مشابه PCP برای توزیع مهلت زمانی استفاده می‌کند با این تفاوت که به جای این که به هر وظیفه یک زیرمهلت زمانی نسبت دهد آن را واقعاً زمان‌بندی می‌کند. فرض نویسندگان بر این بوده که منابع ارائه شده نامحدود هستند، در حالی که در دنیای واقعی هر فراهم‌کننده یک محدودیت برای ارائه تعداد نمونه‌های ماشین مجازی اعمال می‌کند. نتایج نشان داده‌اند که هر دو الگوریتم کارایی قابل قبولی داشته‌اند در حالی که کارایی IC-PCP بهتر بوده است.

در [۱۰] مؤلفان با توسعه الگوریتم PCP توانسته‌اند راهکاری به نام MCPCP برای زمان‌بندی جریان کاری علمی در محیط چندابری ارائه دهند به طوری که سعی در بهینه‌سازی هزینه با رعایت قید مهلت زمانی دارد. آنها هزینه‌ها و زمان انتقال داده بین ماشین‌های مجازی را نیز در نظر گرفته‌اند. همین نویسندگان در [۱۱] الگوریتم MCPCPP را ارائه کرده‌اند که بهبود یافته کار قبلی‌شان است، به این ترتیب که در ساختار گرافی کاربرد، گره‌هایی که تنها یک والد و یک فرزند دارند با گره والد ادغام می‌شوند. آنها هیچ محدودیتی برای به کارگیری سرویس‌ها در نظر نگرفته‌اند، یعنی کاربر می‌تواند هر تعداد نمونه ماشین مجازی را از هر فراهم‌کننده‌ای برای هر زمانی اجاره کند، در حالی که در واقعیت

1. Cuckoo Search Algorithm
2. Optimizing Cost under Deadline Constrain
3. Optimizing Makespan under Budget Constrain
4. Partial Critical Path
5. Sub-Deadline

6. Big Data

7. Directed Acyclic Graph

	1	2	3	...	Nbag*nvm*Nk
1					
2					
3					
⋮					
m					

شکل ۲: نمایش یک جواب به صورت آرایه.

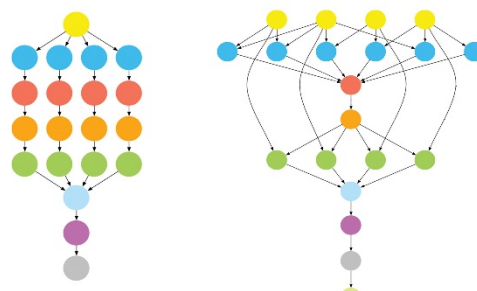
کاربرد و هزینه‌های انتقال داده می‌باشد. هزینه اجاره نمونه‌های ماشین مجازی توسط عبارت $\sum_{k=1}^m \sum_{j \in J_k} P_{kj} \cdot \sum_{i=1}^n \sum_{l \in I_j} T_{ikjl}$ محاسبه می‌شود که در آن پارامترهای P_{kj} و T_{ikjl} در جدول ۱ تعریف شده‌اند. برای محاسبه هزینه انتقال داده باید توجه داشت که اولاً از لحاظ انتقال داده دسته وظایف جریان کاری به صورت متوالی به هم وابسته هستند، یعنی B_{i+1} بعد از B_i قرار دارد اگر B_{i+1} از داده‌های تولیدشده از اجرای B_i استفاده کند. ثانیاً هزینه انتقال داده بین دو دسته وظایف پشت سر هم صفر است اگر به ماشین‌های مجازی یک ابر یکسان تخصیص یابد و در غیر این صورت این هزینه باید محاسبه شود. بنابراین هزینه انتقال داده توسط عبارت $\sum_{i=1}^n \sum_{k=1}^m w_i \cdot DS_i \cdot CD_k$ محاسبه می‌شود که در آن پارامترهای w_i ، DS_i و CD_k در جدول ۱ تعریف شده‌اند. از آنجایی که مطلوبیت ما کاهش مقدار این هزینه است، این مسأله یک مسأله کمینه‌سازی به حساب می‌آید

$$Totalcost =$$

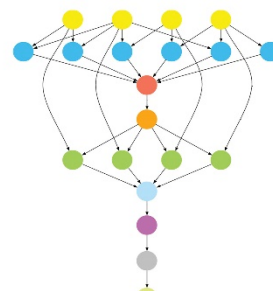
$$\min \sum_{k=1}^m \sum_{j \in J_k} P_{kj} \cdot \sum_{i=1}^n \sum_{l \in I_j} T_{ikjl} + \sum_{i=1}^n \sum_{k=1}^m w_i \cdot DS_i \cdot CD_k \quad (۱)$$

- محل سکونت: برای حل یک مسأله بهینه‌سازی لازم است تا یک جواب^۲ از مسأله به فرم یک آرایه شکل گیرد که در الگوریتم COA به این آرایه محل سکونت یا فاخته یا لانه یا تخم گوئیم. در این تحقیق ساختار آرایه محل سکونت برای مسأله زمان بندی کاربردهای جریان کاری در شکل ۲ نمایش داده شده است. تعداد سطرهاى این آرایه برابر تعداد ابرهای موجود در محیط چندابری (m) می‌باشد و تعداد ستون‌های آن برابر حاصل تعداد دسته وظایف جریان کاری یعنی همان تعداد سطح‌ها (n_{bag}) ضربدر تعداد انواع نمونه ماشین‌های مجازی موجود در هر ابر (n_{vm}) ضربدر تعداد موجود از هر کدام از نمونه‌های ماشین مجازی (N_k) است. هر کدام از درایه‌های این آرایه تعداد وظایفی از سطح (دسته وظایف) i ام که به ماشین نوع z ام و شماره l ام از ابر k ام تخصیص یافته را نشان می‌دهد یعنی همان مقدار متغیر N_{ikjl} در مدل ILP-DC. برای هر کدام از جواب‌ها در جمعیت تولیدشده دو نوع حرکت وجود دارد، مهاجرت^۳ به سمت بهترین جواب و حرکت تصادفی با الگوی خاص که آن را تخم‌گذاری^۴ می‌نامند.

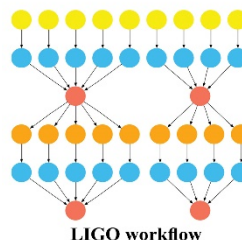
- مهاجرت: در الگوریتم COA، جستجوی سراسری یا همان مهاجرت به کمک پرواز لوی طبق (۲) صورت می‌گیرد. رابطه بیانگر این است که محل سکونت جدید یک فاخته برابر است با محل



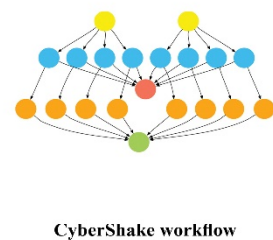
Enigenome workflow



Montage workflow



LIGO workflow



CyberShake workflow

شکل ۱: کاربردهای جریان کاری علمی.

داده‌های ورودی مسأله شامل اطلاعاتی درباره ابرهای محیط چندابری و کاربرد جریان کاری ارسال شده به محیط چندابری است. اطلاعات و نمادهای به کار رفته در این تحقیق در جدول ۱ نشان داده شده است.

۴- بیان مسأله

در این مقاله مسأله زمان بندی کاربردهای جریان کاری علمی در محیط چندابری با هدف بهینه‌سازی هزینه تحت قید مهلت زمانی مورد بررسی قرار می‌گیرد. برای حل این مسأله از روش جستجوی فاخته استفاده می‌کنیم که یکی از مشهورترین الگوریتم‌های فراابتکاری به حساب می‌آید.

۴-۱ روش پیشنهادی

همان طور که در بخش ۱ اشاره شد، مسأله زمان بندی وظایف در محیط محاسبات ابری جزء مسایل NP-Hard بوده و زمان اجرای بسیار زیادی نیاز دارد. برای غلبه بر این دشواری ما با استفاده از روش بهینه‌سازی فاخته سعی داریم این مسأله را در زمان بسیار کوتاه‌تری حل کنیم.

۴-۲ جستجوی فاخته

الگوریتم فاخته با الهام از روش زندگی پرنده‌ای به همین نام توسعه یافته است [۲۱]. این الگوریتم همانند سایر الگوریتم‌های تکاملی با یک جمعیت اولیه (جمعیتی متشکل از فاخته‌ها) کار خود را شروع می‌کند. این جمعیت از فاخته‌ها تعدادی تخم دارند که آنها را در لانه تعدادی پرنده میزبان خواهند گذاشت. موقعیتی که در آن بیشترین تعداد تخم‌ها نجات یابند پارامتری خواهد بود که COA قصد بهینه‌سازی آن را دارد. در الگوریتم فاخته برای هر دو نوع جستجوی سراسری و محلی، از پرواز لوی استفاده می‌شود.

- تابع برازش^۱: تابع برازش در الگوریتم COA پیشنهادی برای مسأله مسأله زمان بندی کاربردهای جریان کاری در (۱) آمده است. این تابع هزینه‌های اجرای کاربرد جریان کاری را محاسبه می‌کند که شامل هزینه‌های اجاره نمونه‌های ماشین مجازی برای اجرای

2. Solution
3. Immigration
4. Egg Laying

1. Fitness Function

جدول ۱: توصیف نمادهای استفاده شده در این تحقیق.

توصیف نماد	نماد
اطلاعات کاربرد جریان کاری	
مجموعه دسته‌های کاربرد جریان کاری در چند سطح	$A = \{B_1, B_2, \dots, B_n\}$
دسته i ام از کاربرد جریان کاری در سطح i ام	B_i
مهلت زمانی کاربرد بر حسب ساعت	DI
اندازه داده لازم برای دسته i ام	DS_i
تعداد دسته وظایف (سطوح) جریان کاری	n
اطلاعات ابرهای محیط چندابری	
تعداد ابرهای محیط چندابری	m
ابرهای فراهم کننده زیرساخت به عنوان سرویس در محیط چندابری	$CP = \{AmazonEC2, \dots\}$
ابر k ام در محیط چندابری	CP_k
نمونه ماشین مجازی شماره l ام از نوع j ام روی ابر k ام	VM_{klj}
مجموعه نمونه ماشین مجازی موجود روی ابر k ام	$J_k = \{e, xlarge, \dots\}$
قیمت اجرای یک ماشین مجازی از نمونه VM_{klj} در واحد ساعت	P_{klj}
تعداد ماشین‌های مجازی هم‌زمان که کاربر می‌تواند در ابر k ام اجرا کند	N_k
هزینه انتقال داده به ابر k ام برای یک گیگابایت داده	CD_k
مجموعه اندیس‌های همه ماشین‌های نوع J ام	I_J

$$P_{i,j} = \begin{cases} 1 & \text{if } rand < pa \\ 0 & \text{if } rand \geq pa \end{cases} \quad (9)$$

الگوریتم شکل ۳ الگوریتم COA را نشان می‌دهد که برای مسأله مورد نظر این تحقیق طراحی و پیاده‌سازی شده است. پارامترهای α ، β ، pa و σ ورودی‌های این الگوریتم هستند که مقادیر آنها همان مقادیری است که در الگوریتم پیاده‌سازی شده توسط مؤلف [۲۱] پیشنهاد شده است. همان طور که در این الگوریتم مشاهده می‌شود در هر تکرار برای هر کدام از جواب‌ها ابتدا عملیات مهاجرت و سپس حرکت تصادفی انجام می‌شود و در صورتی که جواب تولیدشده جدید بهتر باشد جواب قبلی و یا بهترین جواب به روز رسانی می‌شوند.

الگوریتم‌های شکل ۴ و ۵ به ترتیب عملیات مهاجرت و تخم‌گذاری برای یک جواب را نشان می‌دهند. قابل توجه است که بعد از تولید یک جواب جدید باید امکان‌پذیری آن ارزیابی شود. در الگوریتم‌های مهاجرت و حرکت تصادفی به ترتیب در خط ۵ و ۱۴ امکان‌پذیری جواب جدید با استفاده از الگوریتم Feasible-solution ارزیابی می‌شود و در صورتی که جواب امکان‌پذیر نباشد مقدار برازش آن به بی‌نهایت مقداردهی می‌شود. الگوریتم شکل ۶ نیز امکان‌پذیری یک جواب را ارزیابی می‌کند. یک جواب در صورتی امکان‌پذیر خواهد بود که اولاً زمان اتمام اجرای آن از مهلت زمانی تعیین شده بزرگ‌تر نباشد و ثانیاً وظایف هر کدام از دسته‌ها تنها به ماشین‌های یک ابر اختصاص یافته باشند و در غیر این صورت جواب مورد نظر امکان‌پذیر نیست.

۵- شبیه‌سازی و نتایج

الگوریتم فراابتکاری در محیط Matlab و C++ پیاده‌سازی شده است. مشخصات سیستمی که نرم‌افزارها روی آن اجرا شده‌اند عبارت است از: RAM ۸ GB، حافظه Intel®core™ i۷-۵۵۰۰U CPU@۲٫۴۰GHz و BlueLock، GoGorid، Amazon EC2 و Storm cloud به عنوان فراهم‌کنندگان ابر موجود در یک محیط چندابری در نظر گرفته شده‌اند. در همه ابرها ظرفیت یکسانی برای تأمین

سکونت قبلی آن به علاوه ضریبی از فاصله این جواب تا بهترین جواب موجود در کل جمعیت فاخته‌ها. ضریب α را اندازه‌گام‌های فاخته گویند، طبق (۳) ضریب r یک عدد نرمال با میانگین صفر و واریانس σ^2 می‌باشد. ضریب S از (۴) محاسبه می‌شود که در آن u و v متغیرهای با توزیع نرمال می‌باشند. σ_u و σ_v نیز از (۶) محاسبه می‌شوند

$$nest_i^{(t+1)} = nest_i^{(t)} + a.s.(nest_i^{(t)} - nest_{best}^{(t)}) \cdot r \quad (2)$$

$$r \sim N(0, \sigma^{-2}) \quad (3)$$

$$S = \frac{u}{|V|^{\frac{1}{\beta}}} \quad (4)$$

$$u \sim N(0, \sigma_u^2) \quad (5)$$

$$v \sim N(0, \sigma_v^2) \quad (5)$$

$$\sigma_u = \left\{ \frac{r(1+\beta) \sin \frac{\pi\beta}{2}}{r \left| \frac{1+\beta}{2} \right|^{\frac{\beta-1}{2}} \beta^{\frac{\beta-1}{2}}} \right\}^{\frac{1}{\beta}}, \quad \sigma_v = 1 \quad (6)$$

- تخم‌گذاری: در الگوریتم COA، جستجوی محلی به کمک پرواز لوی طبق (۷) صورت می‌گیرد. رابطه بیانگر این است که موقعیت جدید یک فاخته برابر است با موقعیت قدیم آن به علاوه یک مقدار که از ضرب ماتریس S در P حاصل می‌گردد. ماتریس S طبق (۸) به دست می‌آید و درایه‌های ماتریس P از (۹) محاسبه می‌شوند

$$nest^{t+1} = nest^t + S * P \quad (7)$$

$$S = rand(nests(randperm(n),:)) - nests(randperm(n),:) \quad (8)$$

```

1: Input: Solution sol, Pop, Pa;
2: Output: Solution solnew;
3: Select two random solutions sol1 and sol2 in Pop;
4: S = rand(sol1 - sol2);
5: P = rand(size(sol));
6: for all element Pij in P matrix do
7:   if Pij < Pa then
8:     Pij = 1;
9:   else
10:    Pij = 0;
11:   end if
12: end for
13: solnew = sol + S.P;
14: if Feasible-solution(solnew)=0 then
15:   fitness(solnew)=∞;
16: end if
17: return solnew

```

شکل ۶: الگوریتم امکان پذیری جواب.

جدول ۲: مشخصات نمونه‌های عرضه شده توسط ابرهای موجود در محیط چندابری.

فراهم کننده ابری	نوع نمونه	قیمت (دلار در ساعت)	CCU
Amazon EC2	<i>m۲.xlarge</i>	۲,۶۸	۱۲,۴۶
	<i>m۲.۲xlarge</i>	۱,۳۴	۹,۶۲
	<i>m۱.xlarge</i>	۰,۳۸	۳,۹۲
GoGrid	<i>gg-۸gb</i>	۱,۵۲	۲۳,۲
	<i>gg-۴gb</i>	۰,۷۶	۹,۲۸
	<i>gg-۱gb</i>	۰,۱۹	۴,۴۲
BlueLock	<i>b۱-۸gb-۴cpu</i>	۰,۶۶۱	۲۶,۲۶
	<i>b۱-۴gb-۲cpu</i>	۰,۳۰۸	۹,۰۶
	<i>b۱-۲gb</i>	۰,۱۳۴	۶,۱۴
StormCloud	<i>cloud: ۸gb</i>	۰,۲۷	۲۱,۴۱
	<i>cloud: ۴gb</i>	۰,۱۴	۹,۳۳
	<i>cloud: ۱۶gb</i>	۰,۳۴	۴,۷۳۱

شبیه‌سازی‌ها برای کاربردهای جریان کاری Epigenome، LIGO، Montage و CyberShoke اجرا شده‌اند که خصوصیات آنها را از Pegasus workflow Generator گرفته‌ایم. هزینه و زمان انتقال داده بین دو دسته وظایف در دو سطح متوالی از یک کاربرد جریان کاری در صورتی که وظایف هر دو دسته روی یک ابر یکسان تخصیص داده شوند صفر در نظر گرفته شده و در غیر این صورت این مقادیر محاسبه می‌شوند.

۵-۱ مقایسه راهکار پیشنهادی با راهکارهای مرتبط

در این بخش راهکار پیشنهادی این تحقیق (COA) را با راهکار پیشنهادی در [۱۱] به نام MCPCPP و همچنین با راهکار ارائه شده در [۱۳] به نام ILP-DC مقایسه می‌کنیم. دلایل انتخاب این راهکارها برای مقایسه به شرح زیر می‌باشد:

- مشابه راهکار پیشنهادی این تحقیق، هدف دیگر راهکارهای مورد مقایسه، بهینه‌سازی هزینه تحت یک مهلت زمانی در محیط چندابری برای زمان بندی جریان کاری است.
- راهکار MCPCPP بر اساس استراتژی مسیر بحرانی جزئی (PCP) کار می‌کند که یکی از مشهورترین راهکارهای ابتکاری برای مسأله زمان بندی جریان کاری می‌باشد. بنابراین انتخاب مناسبی برای ارزیابی و مقایسه بهینگی جواب‌های به دست آمده است.
- راهکار ILP-DC مسئله مورد نظر را به صورت ریاضی مدل سازی کرده و جواب‌های به دست آمده از آن بهینه سراسری می‌باشند. از آنجا که الگوریتم‌های فراابتکاری جواب‌های نزدیک به بهینه را پیدا

```

1: Input:  $\alpha, \beta, pa, \sigma$ ;
2: Output: Solution sol;
3: Generate Pop randomly which they are feasible
4: i = 0;
5: solbest = the solution with best fitness in Pop;
6: while i <= Maxiter do
7:   for all individual Popi in Pop do
8:     (newPop) = Immigration(Popi, solbest);
9:     if fitness(newPop) < fitness(Popi) then
10:       Popi = newPop;
11:     end if
12:     if fitness(newPop) < fitness(solbest) then
13:       solbest = newPop;
14:     end if
15:     (newPop) = Egg laying(Popi, Pop, Pa);
16:     if fitness(newPop) < fitness(Popi) then
17:       Popi = newPop;
18:     end if
19:     if fitness(newPop) < fitness(solbest) then
20:       solbest = newPop;
21:     end if
22:   end for
23:   i + +;
24: end while
25: return solbest;

```

شکل ۳: الگوریتم جستجوی فاخته.

```

1: Input: Solution sol, Solution solbest;
2: Output: Solution solnew;
3: Calculate r, S values;
4: solnew = sol +  $\alpha.S.(sol - sol_{best}).r$ ;
5: if Feasible-solution(solnew)=0 then
6:   fitness(solnew)=∞;
7: end if
8: return solnew

```

شکل ۴: الگوریتم مهاجرت فاخته.

```

1: Input: Solution sol;
2: Output: flagfs;
3: flagfs = 1;
4: if makespan of sol > deadline then
5:   flagfs = 0;
6: else
7:   for all CPi in Cloud Provider Set do
8:     if sum(ith row elements of sol) > NK then
9:       flagfs = 0;
10:    end if
11:   end for
12: end if
13: return flagfs

```

شکل ۵: الگوریتم تخم گذاری.

منابع در نظر گرفته‌ایم به طوری که هر ابر سه نوع نمونه ماشین مجازی عرضه می‌کند. همچنین محدودیت منابع اعمال شده توسط ابرها هم به طور یکسان برای همه فراهم کنندگان ۲۰ ماشین برای هر فراهم کننده در نظر گرفته شده است ($N_k = 20$). فراهم کنندگان در نظر گرفته شده در محیط چندابری نمونه‌های ماشین مجازی را با کارایی و قیمت متنوع ارائه می‌دهند. کارایی و قیمت نمونه‌ها از داده‌های موجود در پایگاه داده CloudHarmony^۱ گرفته شده است. معیار کارایی بر حسب ccu است که توسط CloudHarmony برای سنجش کارایی فراهم کنندگان زیرساخت مختلف ارائه شده است. قیمت نمونه‌های ماشین مجازی بر حسب ساعت محاسبه می‌شود. مشخصات نمونه‌های ارائه شده توسط ابرهای موجود در محیط چندابری در جدول ۲ نمایش داده شده است.

1. <http://blog.cloudharmony.com/2010/09/benchmarking-of-ec2s-new-cluster.html>

جدول ۵: هزینه کل راهکارهای ILP-DC، MCPCPP و COA برای جریان کاری MONTAGE با ۲۰۰۰ وظیفه بر حسب دلار.

مهلت زمانی	ILP-DC	MCPCPP	COA
۱	۱,۲۱۸	۹,۴۵	۳,۱۵
۲	۱,۲۰۶	۷,۱۹	۳,۰۷
۳	۱,۲۰۶	۷,۱۹	۱,۲۰۶
۴	۱,۲۰۶	۷,۱۹	۱,۲۰۶
۵	۱,۲۰۶	۷,۱۹	۱,۲۰۶
۶	۱,۲۰۶	۷,۱۹	۱,۲۰۶
۷	۱,۲۰۶	۷,۱۹	۱,۲۰۶
۸	۱,۲۰۶	۷,۱۹	۱,۲۰۶
۹	۱,۲۰۶	۷,۱۹	۱,۲۰۶
۱۰	۱,۲۰۶	۷,۱۹	۱,۲۰۶

جدول ۳: هزینه کل راهکارهای ILP-DC، MCPCPP و COA برای جریان کاری EPIGENOME با ۲۰۰۰ وظیفه بر حسب دلار.

مهلت زمانی	ILP-DC	MCPCPP	COA
۱	بدون جواب	۷۱,۷۹	۵۰,۳۶
۲	بدون جواب	۶۸,۶۵	۴۷,۱
۳	۳۵,۷۴	۶۶,۳۳	۴۶,۴۵
۴	۱۷,۸۴	۶۶,۳۳	۴۵,۶۸
۵	۱۷,۸۴	۶۶,۳۳	۲۵,۱۴
۶	۱۷,۷۱	۶۶,۳۳	۲۱,۹۴
۷	۱۷,۵۷	۶۶,۳۳	۱۹,۸۷
۸	۱۷,۵۷	۶۶,۳۳	۱۹,۲۰
۹	۱۷,۵۷	۶۶,۳۳	۱۸,۵۷
۱۰	۱۷,۵۷	۶۶,۳۳	۱۸,۵۷

جدول ۶: هزینه کل راهکارهای ILP-DC، MCPCPP و COA برای جریان کاری CYBERSHAKE با ۲۰۰۰ وظیفه بر حسب دلار.

مهلت زمانی	ILP-DC	MCPCPP	COA
۱	۰,۹۴۲	۸,۵۴	۱,۱۳
۲	۰,۸۱۲	۸,۱۰	۰,۸۱۲
۳	۰,۸۱۲	۷,۸	۰,۸۱۲
۴	۰,۸۱۲	۷,۸	۰,۸۱۲
۵	۰,۸۱۲	۷,۸	۰,۸۱۲
۶	۰,۸۱۲	۷,۸	۰,۸۱۲
۷	۰,۸۱۲	۷,۸	۰,۸۱۲
۸	۰,۸۱۲	۷,۸	۰,۸۱۲
۹	۰,۸۱۲	۷,۸	۰,۸۱۲
۱۰	۰,۸۱۲	۷,۸	۰,۸۱۲

جدول ۴: هزینه کل راهکارهای ILP-DC، MCPCPP و COA برای جریان کاری LIGO با ۲۰۰۰ وظیفه بر حسب دلار.

مهلت زمانی	ILP-DC	MCPCPP	COA
۱	۵,۵۵	۸۰,۴	۱۰,۴۵
۲	۲,۹۸	۷۰,۱۸۱	۷,۲۴
۳	۲,۷۲	۷۰,۱۸۱	۶,۹۸
۴	۲,۷۲	۷۰,۱۸۱	۵,۱۲
۵	۲,۷۲	۷۰,۱۸۱	۳,۶۱
۶	۲,۷۲	۷۰,۱۸۱	۲,۷۲
۷	۲,۷۲	۷۰,۱۸۱	۲,۷۲
۸	۲,۷۲	۷۰,۱۸۱	۲,۷۲
۹	۲,۷۲	۷۰,۱۸۱	۲,۷۲
۱۰	۲,۷۲	۷۰,۱۸۱	۲,۷۲

طوری که در مهلت‌های زمانی طولانی توانسته جوابی بهینه برابر جواب بهینه سراسری بیابد.

چنان که در جدول‌ها مشاهده می‌شود، هزینه کل به دست آمده توسط MCPCPP به طور قابل ملاحظه‌ای بیشتر از مقادیر به دست آمده توسط ILP-DC و COA است. MCPCPP یک راهکار ابتکاری است و بنابراین جواب‌های زیربینه را پیدا می‌کند. از طرف دیگر این افزایش قابل ملاحظه در هزینه کل برای MCPCPP می‌تواند در اثر روش بخش‌بندی آن باشد. راهکار MCPCPP، جریان کاری را به مسیرهای بحرانی جزئی بخش‌بندی می‌کند که این مسیرهای بحرانی جزئی را از طریق پیمایش گراف DAG مربوط به جریان کاری در عمق (به صورت عمودی) بین گره‌های ابتدا و انتها تعیین می‌کند در حالی که راهکار پیشنهادی این تحقیق (COA) و همچنین ILP-DC جریان کاری را به صورت افقی^۳ (سطح به سطح) بخش‌بندی می‌کنند.

در تمامی آزمایش‌ها با افزایش مهلت زمانی، هزینه کاهش می‌یابد. هنگامی که مهلت زمانی کوتاهی برای اجرای کاربرد تعیین شود زمان‌بند مجبور است برای رعایت آن، نمونه ماشین‌های سریع‌تری را انتخاب کند و همان طور که می‌دانیم هر چه سرعت پردازش نمونه ماشین مجازی بیشتر باشد هزینه آن نیز گران‌تر است و برعکس. بنابراین با افزایش مهلت زمانی، زمان‌بند می‌تواند

می‌کنند و یافتن جواب بهینه را تضمین نمی‌کنند، این مقایسه می‌تواند نشان دهد که جواب‌های به دست آمده از الگوریتم فراابتکاری پیشنهادی تا چه اندازه نزدیک به جواب‌های بهینه است. جدول‌های ۳ تا ۶ مقایسه هزینه کل به دست آمده از راهکارهای ILP-DC، MCPCPP و COA را به ترتیب برای کاربردهای جریان کاری Epigenome، LIGO، Montage و CyberShake با اندازه ۲۰۰۰ وظیفه نشان می‌دهند. آستانه شدنی مهلت زمانی^۱ را به صورت کوتاه‌ترین مهلت زمانی که در آن راهکار مورد نظر بتواند جواب پیدا کند تعریف می‌کنیم. در آزمایش‌های این بخش مهلت زمانی از یک تا ده ساعت تغییر می‌کند. از آنجایی که راهکار ما (COA) از نوع فراابتکاری بوده و جواب‌های زیربینه را پیدا می‌کند، بنابراین در هر بار اجرا ممکن است جوابی متفاوت را پیدا کند، پس جواب‌های ذکرشده در جدول‌ها میانگین جواب‌های به دست آمده از ۲۰ اجرای این راهکار است. از این جدول‌ها نتایج زیر مشهود می‌باشد:

در تمامی جدول‌ها هزینه به دست آمده از راهکار ILP-DC کمترین است که این امری بدیهی است زیرا این راهکار از نوع مدل‌سازی ریاضی بوده و بهینه سراسری را می‌یابد. قابل توجه است که مقدار هزینه حاصل از جواب‌های الگوریتم فراابتکاری پیشنهادی COA بسیار کمتر از جواب‌های راهکار MCPCPP و بسیار نزدیک به جواب بهینه به دست آمده از مدل ریاضی ILP-DC است، به

2. Vertically
3. Horizontally

1. Feasible Deadline Threshold

دست آمده از راهکار پیشنهادی به طور قابل ملاحظه‌ای از راهکار MCPCPP کمتر بوده و در حد مطلوبی نزدیک به مدل ریاضی ILP-DC می‌باشد. به علاوه این که زمان اجرای راهکار پیشنهادی نیز معقول می‌باشد. در حالت کلی روش پیشنهادی در مقایسه با روش MCPCPP دارای مزیت کاهش هزینه و ضعف افزایش زمان اجرا است. ولی باید توجه داشت که مزیتش بسیار چشم‌گیر است و با توجه به این که زمان اجرا در مقیاس ثانیه و مهلت زمانی در مقیاس ساعت است، این ضعف قابل چشم‌پوشی است.

مراجع

- [1] M. Mattoso, C. Werner, G. H. Travassos, V. Braganholo, E. Ogasawara, D. Oliveira, S. Cruz, W. Martinho, and L. Murta, "Towards supporting the life cycle of large scale scientific experiments," *Int. J. Bus. Process Integr. Manag.*, vol. 5, no. 1, pp. 79-92, May 2010.
- [2] T. Hey, S. Tansley, and K. M. Tolle, *The Fourth Paradigm: Data-Intensive Scientific Discovery*, vol. 1. Microsoft Research Redmond, WA, 2009.
- [3] I. J. Taylor, E. Deelman, D. B. Gannon, and M. Shields, *Workflows for E-Science: Scientific Workflows for Grids*, Springer Publishing Company, Incorporated, 2014.
- [4] L. Grandinetti, *High Performance Computing and Grids in Action*, vol. 16, IOS Press, 2008.
- [5] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50-55, Dec. 2008.
- [6] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Proc. Grid Computing Environments Workshop*, 10 pp., Austin, TX, USA, 12-16 Nov. 2008.
- [7] F. Wu, Q. Wu, and Y. Tan, "Workflow scheduling in cloud: a survey," *J. Supercomput.*, vol. 71, no. 9, pp. 3373-3418, May 2015.
- [8] S. Abrishami, M. Naghibzadeh, and D. H. J. Epema, "Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds," *Futur. Gener. Comput. Syst.*, vol. 29, no. 1, pp. 158-169, Jun. 2013.
- [9] S. Abrishami, M. Naghibzadeh, and D. H. J. Epema, "Cost-driven scheduling of grid workflows using partial critical paths," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 8, pp. 1400-1414, Aug. 2012.
- [10] B. Lin, W. Guo, G. Chen, N. Xiong, and R. Li, "Cost-driven scheduling for deadline-constrained workflow on multi-clouds," in *Proc. IEEE Int. Parallel and Distributed Processing Symp. Workshop. IPDPSW'15*, pp. 1191-1198, Hyderabad, India, 25-29 May 2015.
- [11] B. Lin, et al., "A pretreatment workflow scheduling approach for big data applications in multi cloud environments," *IEEE Trans. on Network and Service Management*, vol. 13, no. 3, pp. 581-594, Sept. 2016.
- [12] H. M. Fard, R. Prodan, and T. Fahringer, "A truthful dynamic workflow scheduling mechanism for commercial multicloud environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1203-1212, Jun. 2013.
- [13] S. Mohammadi, H. Pedram, and L. PourKarimi, "Integer linear programming-based cost optimization for scheduling scientific workflows in multi-cloud environments," *the J. of Supercomputing*, vol. 74, no. 9, pp. 4717-4745, Sep. 2018.
- [14] M. S. Hosseini Shirvani, "A hybrid meta-heuristic algorithm for scientific workflow scheduling in heterogeneous distributed computing systems," *Engineering Applications of Artificial Intelligence*, vol. 90, Article No.: 103501, 20 pp., Apr. 2020.
- [15] K. R. Escott, H. Ma, and G. Chen, "Genetic programming based hyper heuristic approach for dynamic workflow scheduling in the cloud," in *Proc. Int. Conf. on Database and Expert Systems Applications*, pp. 3141-3148, Canberra, ACT, Australia, 1-4 Dec. 2020.
- [16] N. Rizvi and D. Ramesh, "HBDCWS: heuristic-based budget and deadline constrained workflow scheduling approach for heterogeneous clouds," *Soft Computing*, vol. 24, pp. 18971-18990, Jul. 2020.
- [17] G. B. Berriman, et al., "Montage: a grid-enabled engine for delivering custom science-grade mosaics on demand," *SPIE Astronomical Telescopes+ Instrumentation*, pp. 221-232, Sept. 2004.

جدول ۷: زمان اجرای سه راهکار ILP-DC، COA و MCPCPP برای جریان کاری‌های Epigenome و LIGO بر حسب ثانیه.

تعداد وظایف	راهکار	Epigenome	LIGO
	COA	۲۹	۳۹
۲۰۰۰	MCPCPP	<۱	<۱
	ILP-DC	۲۲۹	۱۳۲۰
	COA	۷۶	۱۲۳
۴۰۰۰	MCPCPP	<۱	<۱
	ILP-DC	۷۵۴	۳۰۹۴

ماشین‌هایی را انتخاب کند که قدرت پردازش کمتری داشته و ارزان‌تر هستند، به همین دلیل با افزایش مهلت زمانی هزینه بهینه کاهش می‌یابد.

- در همه جدول‌ها می‌توان دید که بعد از یک مقدار مشخص از مهلت زمانی، هزینه بهینه ثابت می‌ماند. مهلت زمانی یک قید محدودکننده^۱ در این مسأله زمان‌بندی است اما هنگامی که به مقدار مشخصی برسد دیگر افزایش آن باعث کاهش هزینه نمی‌شود زیرا ارزان‌ترین نمونه ماشین‌ها برای زمان‌بندی کاربرد مورد نظر انتخاب شده‌اند و دیگر امکان کاهش هزینه وجود ندارد. برای مثال در جدول ۳ می‌بینیم که این مقدار مشخص از مهلت زمانی برای کاربرد Epigenome در راهکار ILP-DC با اندازه بزرگ برابر ۷ ساعت است.

۲-۵ زمان بهینه‌سازی

جدول ۷ زمان اجرای راهکارهای ILP-DC، COA و MCPCPP را برای جریان کاری‌های Epigenome و LIGO نشان می‌دهد. تعداد وظایف ۲۰۰۰ و ۴۰۰۰ در نظر گرفته شده در حالی که دیگر پارامترها ثابت هستند (مهلت زمانی برابر یک ساعت است). همان‌طور که مشاهده می‌شود، زمان اجرای راهکارهای ILP-DC و COA با افزایش اندازه جریان کاری، افزایش می‌یابد و این افزایش در راهکار ILP-DC بسیار بیشتر از COA است و این در حالی است که زمان اجرای راهکار MCPCPP در همه موارد کمتر از یک ثانیه است. باید توجه داشت که زمان اجرای راهکار COA در مقایسه با زمان اجرای کاربرد جریان کاری قابل چشم‌پوشی است. بنابراین در عمل سرعت نسبتاً کم راهکار COA در مقایسه با MCPCPP محدودکننده نیست. به خصوص این که کیفیت جواب‌های حاصل از آن بسیار بالاتر از جواب‌های MCPCPP است.

۶- نتیجه‌گیری

حل مسایل بهینه‌سازی از طریق جستجوی تمامی جواب‌های امکان‌پذیر (فضای جستجو) و محاسبه توابع هدف بسیار زمان‌بر است. اخیراً تمرکز بیشتری بر روش‌های فراابتکاری صورت گرفته است. روش‌های جستجوی فراابتکاری روش‌هایی هستند که می‌توانند جواب خوبی (نزدیک به بهینه) در زمانی معقول برای مسأله ارائه کنند. در این تحقیق با به کارگیری یکی از جدیدترین روش‌های جستجوی فراابتکاری به نام الگوریتم جستجوی فاخته، یک راهکار فراابتکاری برای مسأله زمان‌بندی کاربردهای جریان کاری با هدف کمینه‌سازی هزینه در محیط چندابری ارائه شده است. نتایج به دست آمده نشان می‌دهند که هزینه به

لطیف پورکریمی تحصیلات خود را در مقاطع کارشناسی در سال ۱۳۸۲ از دانشگاه شهرکرد، کارشناسی ارشد در سال ۱۳۸۴ از دانشگاه تربیت معلم تهران (خوارزمی) و دکتری در سال ۱۳۸۹ از دانشگاه شهید باهنر کرمان به پایان رسانده است و هم‌اکنون دانشیار گروه ریاضی دانشگاه رازی می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: برنامه‌ریزی خطی و غیرخطی، بهینه‌سازی چندهدفه و تحلیل پوششی داده‌ها.

سمیه عبدی در سال ۱۳۸۴ مدرک کارشناسی مهندسی کامپیوتر خود را از دانشگاه رازی کرمانشاه به پایان رسانده است. نام‌برده در سال ۱۳۸۹ مدرک کارشناسی ارشد مهندسی کامپیوتر خود را از دانشگاه علوم و تحقیقات تهران دریافت نمود و در سال ۱۳۹۵ موفق به اخذ درجه دکترا در مهندسی کامپیوتر از دانشگاه مذکور گردید. دکتر عبدی از سال ۱۳۸۹ در دانشکده مهندسی کامپیوتر دانشگاه آزاد اسلامی واحد اسلام آباد غرب مشغول به فعالیت گردید و اینک نیز عضو هیأت علمی این دانشکده می‌باشد. زمینه‌های علمی مورد علاقه نام‌برده متنوع بوده و شامل موضوعاتی مانند سیستم‌های توزیع شده، زمانبندی جریان کاری، برنامه‌ریزی ریاضی، معماری نرم‌افزار و یادگیری عمیق می‌باشد.

- [18] P. Maechling, *et al.*, "SCEC cybershake workflows-automating probabilistic seismic hazard analysis calculations," in *Workflows for E-Science: Scientific Workflows for Grids*, I. J. Taylor, E. Deelman, , D. B. Gannon, and M. Shields, (Eds.), pp. 143-163, Springer, 2007.
- [19] -, *USC Epigenome Center*, [Online]. Available: <http://epigenome.usc.edu>.
- [20] "LIGO project, LIGO-laser interferometer gravitational wave observatory," [Online]. Available: <http://www.ligo.caltech.edu/>.
- [21] X. S. Yang and S. Deb, "Cuckoo search via Levy flights," in *Proc. World Congress on Nature & Biologically Inspired Computing, NaBIC'09*, pp. 210-214, Coimbatore, India, 91-11 Dec. 2009.

سمیه محمدی تحصیلات خود را در مقطع کارشناسی مهندسی کامپیوتر در سال ۱۳۸۴ از دانشگاه رازی کرمانشاه، کارشناسی ارشد و دکتری را به ترتیب در سال‌های ۱۳۸۹ و ۱۳۹۷ از دانشگاه علوم تحقیقات تهران به پایان رسانیده و هم‌اکنون استادیار دانشگاه آزاد اسلامی واحد کرمانشاه-مرکز قصرشیرین می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: برنامه‌ریزی ریاضی، یادگیری ماشین، جریان کاری‌های علمی.