بررسی و مطالعه روش‌های زمانبندی درخواست‌ها در رایانش ابری

صبا سرحدی [*1]، شیوا رزاق‌زاده [2]، مهدی عفت‌پرور [3]

1- دانشجوی کارشناسی ارشد کامپیوتر-نرم‌افزار دانشگاه آزاد اسلامی واحد اردبیل
2- گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد اردبیل، اردبیل، ایران
3- گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد اردبیل، اردبیل، ایران

خلاصه

در سیستم رایانش ابری، الگوریتم‌های زمانبندی مختلفی برای تخصیص منابع به کارهای درخواستی کاربران ارائه شده است. زمانی که در یک سیستم محاسباتی با تعداد زیادی از پردازش‌ها مواجه است، این پردازش برای به دست آوردن منابع جهت اجرا، با یکدیگر در حال رقابت هستند. واضح است که در این حال سیستم نیازمند یک بخش مدیریتی اختصاص پردازش‌ها به منابع می‌باشد. در سیستم‌های محاسباتی راه‌حل‌های که برای حل این مسئله بیان شده است الگوریتم‌های زمانبندی نام دارد. زمانبندی وظایف در رایانش ابری باید به گونه‌ای انجام گیرد تا کیفیت خدمات تأمین کننده ابر را به بالاترین حد خود برساند و مشتری را از سرویس‌های تأمین کننده راضی نگه دارد. در همین راستا، بسیاری از تحقیقات گذشته، بر روی یک هدف تمرکز کرده‌اند و آن هدف "بهترین زمان اجرا" یا به عبارتی کمترین زمان اجرای وظایف است، در صورتی که کیفیت خدمات هنگامی حاصل می‌شود که تمامی اهداف در نظر گرفته شود. از آنجا که زمانبندی وظایف در رایانش ابری دارای پیچیدگی محاسباتی زیادی می‌باشد و در مجموعه مسائل NP-Hard قرار می‌گیرد بنابراین استفاده از رویکردهای فراابتکاری می‌تواند جواب بهینه‌ای را در مدت زمان معقول به دست آورد. درواقع در زمانبندی وظایف و منابع، هدف ایجاد نوعی الگوریتم برای زمانبندی است به طوریکه مجموعه‌ای از کارهای موجود در تعداد منابع محدود در کمترین زمان اجرا به درستی اجراشده و خاتمه یابند؛ بنابراین انجام پژوهش در این زمینه می‌تواند ارزشمند باشد. با در نظر گرفتن زمان پردازش کارها، استفاده از منابع بر اساس استفاده از حافظه و بازده، محیط پردازش ابری برای کنترل تمام درخواست‌های مشتری‌ها، می‌تواند حداکثر سرویس را برای تمام مشتریان فراهم نماید. با توجه به چالش ذکر شده در این پژوهش، به مطالعه و ارزیابی روش‌های ارائه شده برای بهبود زمانبندی در سیستم‌های رایانش ابری پرداخته شده است.

کلمات کلیدی: تعادل بار، رایانش ابری، زمانبدی درخواست، الگوریتم فراابتکاری

1. مقدمه

مسئله زمانبندی کارها در محیط رایانش ابری، چالش مهمی است که تأثیر مستقیمی بر روی کلیات سرویس ارائه شده در آن دارد. تخصیص منابع عبارت است از تصمیم‌گیری درباره اینکه چگونه، چه مقدار، کجا و چه زمان منابع در دسترس برای کاربر ایجاد شود. به طور نمونه، کاربران درباره نوع و میزان منابع درخواستی تصمیم می‌گیرند، سپس فراهم‌کنندگان، منابع درخواستی را بر روی گره‌هایی در مراکز داده قرار می‌دهند. برای اجرای برنامه‌های کاربردی به طور کارا، نوع منابع باید با ویژگی‌های حجم بار مطاقبت داشته باشد و مقدار آن نیز باید محدودیت‌ها را (مانند مهلت زمانی تکمیل کار) به طور شایسته ارضا نماید. در یک محیط الاستیک مانند ابر که کاربران می‌توانند منابع را به طور پویا درخواست نمایند یا بازگردانند، در نظر گرفتن اینگونه تنظیمات دارای اهمیت زیادی است. پس از تخصیص منابع محاسباتی نیاز به اتخاذ تصمیماتی درباره زمانبندی کارها می‌باشد. زمانبندی کار فرآیند نگاشت کارها به منابع در دسترس بر پایه نیازمندی‌ها و ویژگی کارها است. مسئله زمانبندی کار در رایانش ابری، یک مسئله بسیار مهم و از رده مسائل NP محسوب می‌شود که سعی دارد یک زمانبندی بهینه برای اجرای وظایف و تخصیص بهینه منابع مشخص نماید به شکلی که در زمان کمتر کارهای بیشتری را بتوان پردازش کرد. زمانیکه منابع به کاربر داده می‌شود، برنامه کاربردی یک تصمیم زمانبندی اتخاذ می‌نماید. در بسیاری موارد، برنامه کاربردی شامل کارهای چندگانه است که منابع تخصیص داده شده به آن‌ها داده می‌شود[1]. برنامه کاربردی همچنین باید مطمئن شود که به هر کار مقدار منابع کافی و مناسب داده می‌شود، یا اشتراک آن‌ها منصفانه خواهد بود در مسئله پیشرو، به طور کلی، m ماشین مجازی و n کار موجود است. باید توجه داشت که این m ماشین در c سلول (سرور) مستقر بوده و از هر کدام، یک یا بیشتر موجود است. هر کار دارای یک فرآیند خاص است و توالی انجام عملیات مختلف به منظور تکمیل آن کار مشخص است. همچنین هر یک از کارها دارای تعدادی کار فرعی یا وظیفه می‌باشد. انجام هر وظیفه از هر کار بر روی هر ماشین، دارای زمان پردازش خاص خود است. ظرفیت سخت‌افزاری و نرم‌افزاری سرورها و ماشین‌های مجازی مشخص بوده و هر ماشین تنها یک عملیات یا وظیفه را در هر لحظه می‌تواند انجام دهد.از یک منظر دیگر، زمانبندی وظایف موضوع مهمی است که عملکرد محیط رایانش ابری را به شدت تحت تأثیر قرار می‌دهد. زمانبندی فرآیند نگاشت وظایف به منابع موجود بر اساس ویژگی‌ها و الزامات وظایف است. زمانبندی کارآمد یک نیاز ضروری برای ابر است، زیرا می‌تواند استفاده از منابع ابری را برای ارائه خدمات بسیار کارآمد با کیفیت بالا بهینه کند. همانطور که کاربران بیشتری شروع به استفاده از ابرها برای استقرار برنامه‌های کاربردی پیچیده و ذخیره داده‌های راه دور می‌کنند، نیاز مبرمی به داشتن الگوریتم‌های زمان‌بندی قوی وجود دارد که بتواند وظایف را به مراکز داده اختصاص دهد. الگوریتم‌های زمانبندی گردش کار زیادی برای محیط‌های محاسباتی ناهمگن وجود دارد، این الگوریتم‌ها مصالحه مناسب را پیاده‌سازی می‌کنند یا ترکیبی از الگوریتم‌های زمانبندی را با توجه به نیازهای برنامه‌های کاربردی مختلف اعمال می‌کنند. با این حال، این الگوریتم‌ها در اعمال مستقیم در محیط‌های ابری با مشکل مواجه هستند[2].

در این مقاله، به مطالعه و ارزیابی روش‌های ارائه شده جهت زمانبندی درخواست در منابع ابری تمرکز شده است. مزایا و معایب هر یک از روش‌ها به صورت جداگانه مورد بررسی قرار گرفته است. هدف نهایی در این گزارش این است که مطابق با نتایج حاصل از بررسی روش‌ها و تکنیک‌های پیشین راه کاری بهینه برای بهبود زمانبندی درخواست برای کارهای آتی ارائه شود. در ادامه ساختار مقاله بدین شرح است : در بخش دوم (2) مفاهیم بنیادی، در بخش سوم (3) مطالعه و بررسی روش‌های زمانبندی درخواست‌ها در رایانش ابری، در بخش چهارم (4) به مقایسه و ارزیابی روش‌های مورد بررسی پرداخته شده است .در نهایت بخش پنجم (5) نتایج اصلی و مهم تحقیق، با ارائه یک مسیر روشن نشان داده شده است.

2. مفاهیم بنیادی

در این بخش برای آشنایی با محوریت موضوع و اصطلاحات مهم در این مقاله، مفاهیم به صورت کوتاه و مفید بیان شده است.

الف) رایانش ابری: توسعه و به‌کارگیری فناوری کامپیوتر بر مبنای اینترنت است. این فناوری شیوه‌ای از محاسبات کامپیوتری در فضایی است که قابلیت‌های مرتبط با فناوری اطلاعات به عنوان سرویس یا خدمات برای کاربر عرضه می‌شود و به او امکان می‌دهد به سرویس‌های مبتنی بر فناوری در اینترنت دسترسی داشته باشد، بدون آنکه اطلاعات تخصصی در مورد این فناوری‌ها داشته باشد یا بخواهد زیرساخت را در دست گیرد. محاسبات ابری ساختاری شبیه یکه توده ابر دارد که به واسطه آن کاربران می‌توانند به برنامه‌های کاربردی از هر جایی از دنیا دسترسی داشته باشند [1].

ب) زمانبندی وظایف: یک فرآیند کلیدی در زیرساخت به عنوان سرویس است که هدف آن، اجرای درخواست‌های وارد شده به سیستم بر روی منابع، به شیوه‌ای کارآمد با درنظر گرفتن سایر خصوصیات ابر می‌باشد. زمانبندی وظایف، ماشین‌های مجازی را به عنوان واحدهای زمانبندی، جهت تخصیص منابع فیزیکی ناهمگون برای اجرای وظایف در نظر می‌گیرد. به دلیل خصوصیات پویای ابر و ناهمگون بودن آن بحث زمانبندی می‌بایست به صورت اتوماتیک و بسیار سریع انجام گیرد. در ابر یک کاربر وظایف را از طریق ترمینال‌ها می‌فرستد. زمانبند به مخزن اطلاعاتی متصل می‌شود و سرویس اطلاعاتی ابر را فراخوانی می‌کند. این سرویس، همه اطلاعات را درباره منابع ابر داراست و شامل تعدادی شبکه خصوصی برای دستیابی به پایگاه داده‌ها است [2].

ج) تخصیص منابع: یک موضوع است که در بسیاری از حوزه‌های محاسبات، مانند سیستم‌عامل، محاسبات شبکه و مدیریت مرکز داده قرار گرفته است. سیستم تخصیص منابع در رایانش ابری می‌تواند به عنوان هر گونه سازوکاری که هدف آن تضمین کننده برنامه‌های کاربردی مورد نیاز به شکل صحیح توسط زیرساخت‌های ارائه دهنده وجود دارد دیده شود. همراه با این تضمین، توسعه‌دهنده سازوکارهای تخصیص منابع، نیز باید به منظور اعمال الگوریتم‌ها جهت تخصیص بهتر منابع فیزیکی یا مجازی به برنامه‌های کاربردی، وضعیت فعلی هر یک از منابع در محیط ابر را در نظر بگیرد، در نتیجه هزینه‌های عملیاتی محیط ابری نیز کاهش می‌یابد. تأمین منابع وظیفه نگاشت منابع به اشخاص مختلف ابر بر اساس تقاضا است[3].

د) الگوریتم‌های فراابتکاری: مجموعه‌ای از روش‌های محاسباتی هستند که برای حل مسائل پیچیده و بهینه‌سازی مسائل بزرگ و دشوار استفاده می‌شوند. این الگوریتم‌ها، برخلاف الگوریتم‌های سنتی به جای جستجوی دقیق و کامل در فضای مسئله، از جستجوی هوشمندانه و تقریبی استفاده می‌کنند. هدف اصلی آن‌ها، یافتن راه حل‌های خوب و کارآمد در زمان قابل قبول است، حتی اگر بهترین راه حل ممکن را پیدا نکنند. بهینه‌سازی مسائل در هر یک از الگوریتم‌های فراابتکاری، با استفاده از اصول و الهام‌گیری از پدیده‌های طبیعی و بیولوژیکی انجام می‌شود. از الگوریتم‌های فراابتکاری به عنوان روش‌های هوشمندانه برای مدیریت چالش‌های دشوار، بهینه‌سازی مسائل و پیشبرد مشکلات فنی و علمی استفاده می‌شود. برخلاف روش‌های سنتی، الگوریتم فراابتکاری با استفاده از جستجوی تقریبی و الهام گرفته از طبیعت، به یافتن راه حل‌های نزدیک به بهینه در زمان کوتاه‌تر می‌پردازند[4].

3. مطالعه تحقیقات انجام شده

در این بخش به مطالعه و بررسی روش‌ها و تحقیقات انجام شده در حوزه زمانبندی درخواست در بستر ابر پرداخته شده است. هر یک از تحقیقات مورد بررسی در نهایت از دید مزایا، معایب و اهداف مقایسه شده‌اند.

در سال 2019 در [5]، الگوریتم ژنتیکی برای حل زمان‌بندی وظایف وابسته ارائه کرد که در آن به دو پارامتر مهم کیفیت سرویس که زمان و هزینه می‌باشند توجه شده است. در این الگوریتم به جای تولید جمعیت اولیه به صورت تصادفی از متغیرهای آشفته استفاده شده است. ترکیب امتیازات الگوریتم ژنتیک با متغیرهای آشفته باعث شده که راه‌حل‌های تولید شده توسط این الگوریتم در تمام فضای جستجو پخش شوند و از همگرایی زودرس الگوریتم جلوگیری شود و راه‌حل‌ها و تولیدات بهتر در زمان کوتاه‌تری بدست آیند و سرعت همگرایی الگوریتم افزایش یابد.

در [6] الگوریتم زمانبندی جدیدی را بر مبنای دو الگوریتم پایه‌ای Min-Min و Max-Min طراحی کرده است. به طوری که از مزایای این دوالگوریتم استفاده کرده و در عین حال معایب پوشانده شده است. معیار انتخاب الگوریتم پیشنهادی، الگوریتم انتخابی بین دو الگوریتم مذکور، انحراف معیار استاندارد مدت زمان پایان کار وظایف روی منابع است. الگوریتم Min-Min شامل دو مرحله است که در مرحله اول زمان انتظار برای هر زیر وظیفه معین می‌شود و در مرحله بعد وظایف بر اساس زمان اتمام کار به صورت نزولی مرتب و علامت گذاری می‌شوند. وظایف بر اساس اولویت منابع به منابع متناظرشان اختصاص داده می‌شوند و این روند ادامه دارد تا زمانی که همه وظایف موجود در MT پردازش شوند. الگوریتم Max-Min مشابه الگوریتم Min –Min است با این تفاوت که در مرحله اول وظایف و کارها براساس زمان پایان کار به صورت صعودی مرتب و علامت‌گذاری می‌شوند.

در [7]، الگوریتم Genetic- Annealing را که یک الگوریتم تکامل ترکیبی است، برای حل مساله زمانبندی وظایف مستقل در گرید،در سال 2010 ارائه کرده است. هدف اصلی در این الگوریتم، یافتن راه حلی می باشد که زمان کلی اجرا در آن مینیمم باشد.با توجه به اینکه الگوریتم ژنتیک به صورت سراسری ، فضای مسئله را جستجو می‌کند و در جستجوی محلی، ضعیف عمل می کند،با ترکیب آن با شبیه سازی حرارتی که یک الگوریتم جستجوی محلی است، سعی شده که این عیب را برطرف کرده و بدین ترتیب از ترکیب امتیازات این دو الگوریتم استفاده شده است. نحوه نمایش کروموزوم ها به صورت الگوریتم RFOH می‌باشد. الگوریتم ژنتیک شامل یک تولید کننده تصادفی جمعیت، عملگر انتخاب نخبه‌گرا، ترکیب و جهش تکراری به کمک شبیه ساز حرارتی می‌باشد. بر اساس تابع برازش،عملگر انتخاب، نصف بهترین کروموزوم‌ها را از جمعیت انتخاب می‌کند. عملگر ترکیب اجرا می‌شود و بدین صورت فرزندان جدید برای نسل بعد تولید می‌شوند. بکاربردن یک عملگر ترکیب، باعث می‌شود که فضای راه‌حل به خوبی جستجو شود. عملگر تکراری که به عنوان جهش استفاده شده است، باعث بهینه شد افراد می‌شود و سبب می‌شود که فرد بهتر با استفاده از یک جستجوی محلی تکراری، توسط شبیه‌سازی حرارتی پیدا شود. این پروسه برای هر نسل از جمعیت، تکرار می‌شود. لازم به ذکر است که تکنیک شبیه‌سازی حرارتی بر روی هر فرد یک جمعیت که جهش داده شده است، اجرا می‌شود.

در سال 2024 در [8]، الگوریتم ترکیبیHPSO که در واقع ترکیب PSO و شبیه‌سازی حرارتی می‌باشد را ارائه کردند. به کار گرفتن شبیه‌سازی حرارتی برای دوری گزیدن از افتادن در یک بهینه‌ی محلی می‌باشد.نحوه نگاشت مسئله زمانبندی به یک راه حل در این مقاله، به این صورت است که فضای جستجو را به صورت N×M بعد در نظر گرفته است که N به تعداد زیر وظایف و M به تعداد منابع اشاره دارد.هر ذره شامل M قسمت است و هر بخش تعداد N وظیفه مستقل دارد که ترکیب N وظیفه رویM ماشین را نشان می‌دهد. در هر بار تکرار و ایجاد بردارهای موقعیت جدید، با توجه به اینکه احتمال این که اعداد اعشاری در هرموقعیت بدست آید که برای شماره وظایف نامعتبر است. در الگوریتم ارائه شده، مقادیر اعشاری بدست آمده به نزدیکترین عدد صحیح گرد می‌شوند.این کار سبب می‌شود که یک مسئله بهینه‌سازی پیوسته به یک مسئله بهینه سازی گسسته تبدیل شود. تابع برازش نیز، میزان زمان اجرا در نظر گرفته شده است. از آنجایی که احتمال گیر افتادن الگوریتم پرندگان در بهینه محلی وجود دارد،در هر بار تکرار الگوریتم،بردار جی بست تولید شده به الگوریتم شبیه سازی حرارتی داده می‌شود تا جستجو بصورت محلی برای یافتن راه حل مناسب صورت گیرد و از افتادن در یک بهینه محلی

جلوگیری شود. این زمانبندی‌ها در راستای رسیدن به هدفی بوده و اکثرا سعی در افزایش پارامترهای کیفیت سرویس دارند.

در سال 2022، الگوریتم GELS برای اولین بار در[9]، برای حل مسئله رزرو منابع به کار گرفته شده است. تضمین کیفیت سرویس برای کاربران می تواند با رزرو منابع فراهم شود. رزرواسیون در واقع مکانیزمی است که می تواند این قابلیت را برای کاربران فراهم کند که منابع تخصیص داده شده به آن ها،بر اساس توافق روی کیفیت سرویس مورد نیازشان و افزایش تعداد درخواست های پذیرفته شده ی کاربران در شبکه بندی باشد.با توجه به اینکه الگوریتم GELS یک الگوریتم جستجوی محلی می باشد،فضای مسئله را به خوبی جستجو کرده و از افتادن در مینیمم محلی، دوری می کند.در این الگوریتم، ابعاد مسئله برابر تعداد وظایف ورودی در نظر گرفته شده است،ب طوریکه با تغییر هر یک از این ابعاد می‌توان یک راه حل همسایه برای راه حل جاری ایجاد کرد.راه حل همسایه در واقع راه حلی است که در آن،منبع تخصیص داده شده به آن بعد، متفاوت باشد، به نحوی که اجرای وظیفه را در آینده تضمین کند. به عبارت دیگر،هر وظیفه ای که نتواند توسط منبع تخصیص داده شده به آن انجام شود، منبع مورد نظر برای آن وظیفه تغییر می‌کند. الگوریتم جاذبه گرانشی برعکس الگوریتم‌های جستجوی دیگر، راه حل همسایه را به صورت تصادفی بدست نمی‌آورد، بلکه هر راه حل جاری، همسایه‌های مختلفی دارد که هر یک از آن‌ها بر اساس یک تغییر بخصوصی که مسیر حرکت نامیده می‌شود، بدست می‌آیند. نتایج شبیه‌سازی این الگوریتم نشان داد که GELS نسبت به الگوریتم ژنتیک نتایج بهتری بدست آورده و میزان زمان اجرا کلی را کاهش داده است. مراحل الگوریتم بصورت زیر است:

الف- در الگوریتم شبیه سازی شده، ابتدا تعداد وظایف و منابع مشخص می‌شوند.

ب- با یک تابع تصادفی، زمان آزاد هر منبع و زمان اجرای هر وظیفه، مشخص می‌شود.

ج- مقادیر بردار سرعت بصورت تصادفی از مقدار یک تا تعداد کل وظایف انتخاب می‌شود.

د- در هر مرحله، بزرگترین اندیس بردار سرعت انتخاب شده و منابع تخصیص داده شده به آن بعد تغییر می‌کند. سپس مقدار برازش راه حل جاری محاسبه می‌گردد. اگر مقدار برازش راه حل جاری نسبت به بهترین راه حل بدست آمده بزرگتر باشد، به عنوان بهترین راه حل این مرحله انتخاب می‌شود.

ه- نیروی گرانشی بین راه حل جاری و کاندید محاسبه می‌شود.

و- نیروی گراشی بدست آمده در مرحله 5 به بردار سرعت اولیه اندیس مورد نظر اضافه می‌شود.

ز- مرحله 1 تا 6 ادامه می یابد تا وقتی که یکی از دو شرط زیر رخ دهد: همه عناصر بردار سرعت اولیه، صفر شود یا تعداد تکرار الگوریتم، به ماکزیمم تعدادش برسد.

گریدهای محاسباتی با توجه به کاربرد و بستر محاسباتی آن دارای مدل‌ها و پیش فرض‌های متفاوتی می‌باشد. با توجه به مدل‌ها و فرضیات مختلف الگوریتم‌های زمانبندی بسیاری برای مسئله زمانبندی ارائه شده است. در این بخش الگوریتم‌های زمانبندی سیستم‌گرا و کاربرگرا به منظور افزایش پارامترهای کارایی در محیط گرید صورت گرفته ارائه شده است. همان گونه که مشاهده می‌گردد تاکنون الگوریتمی زمانبندی که پارامترهای کارایی را هم از دید کاربر و هم از دید سیستم مورد بررسی قرار دهد ارائه نشده است.

در [10] 2021 مدلسازی توزیع وظایف و محاسبه‌ی قابلیت اطمینان در شبکه‌های گرید با نوپولوژی ستاره را در سال ارائه دادند.زمانبندی وظایف برای رسیدن به سطح کیفیت مطلوب، از جمله زمینه‌های مهم و مطرح در شبکه‌های گرید است. در این تحقیق قابلیت اطمینان در سرویس های گرید بررسی‌شده و با استفاده از شبکه‌های پتری رنگی، مدلی برای محاسبه ارائه گردیده است. در این تحقیق، محیط گرید دارای توپولوژی ستاره بوده و در نتیجه RMS با همه منابع در گرید ارتباط دارد. همانطوری که مشخص شده است وظیفه RMS دریافت وظایفی تشکیل شده از یک سری زیر وظایف از کاربر و سپس توزیع آن زیر وظایف متناظر بر روی منابع موجود بر روی گرید است. فرض می‌شود که در گرید برای افزایش قابلیت اعتماد

از تکنیک افزونگی استفاده شده است به طوریکه RMS یک زیر وظیفه را به بیش از یک منبع ارسال می‌کند تا هر کدام از منابع متناظر تخصیص یافته که زودتر محاسبات را اتمام کرد، از نتایج آن استفاده گردد. RMS پس از دریافت وظیفه آنرا به زیر وظایف تقسیم می‌کند. فرض می‌شود که تعداد زیر وظایف n و تعداد منابع موجود r است. طبق تکنیک افزونگی بایستی تعداد n از r کمتر باشد،پس از تقسیم، RMS هر زیر وظیفه از یک وظیفه را به تعدادی منبع اختصاص می‌دهد. از فرضیات مهم دیگر این است که یک وظیفه به طوری به زیر وظایفی تقسیم شده است که آن زیر وظایف کاملاً مستقل از همدیگر هستند و نیازی به نتایج یکدیگر برای شروع کار ندارند. C حجم کل محاسبات لازم برای اجرای وظیفه‌ی محوله به RMS در نظر گرفته می‌شود و تعداد کل منابع گرید را r فرض می‌شود. RMS وظیفه دریافتی S را به زیر وظایف $S_1, S_2, \dots, S_n$ تقسیم می‌کند که هر کدام از زیر وظایف دارای پیچیدگی محاسباتی c_j $1; \leq j \leq n$ و داده مورد نیاز برای شروع محاسبات d_j $1; \leq j \leq n$ است.داریم:

$$\sum_{j=1}^{n} c_j = C \qquad (1)$$

پس از تقسیم وظایف به زیر وظایف هر کدام از وظایف به مجموعه‌ای از منابع تخصیص داده می‌شود،بطوریکه S_1 به مجموعه $\{R_1^1, R_2^1, \dots, R_{r_1}^1\}$ و S_2 به مجموعه $\{R_1^2, R_2^2, \dots, R_{r_2}^2\}$ از منابع و ... تخصیص داده می‌شوند و داریم:

$$\sum_{j=1}^{n} r_j = r \qquad (2)$$

که r تعداد منابع موجود در گرید است. همانطوریکه ذکر شد برای شروع محاسبه هر زیر وظیفه در یک منبعی از گرید، نیازمند انتقال مقدار داده مورد نیاز برای آن زیر وظیفه از RMS به منبع متناظر هستیم.فرض می‌کنیم که سرعت انتقال اطلاعات از RMS به منبع k برابر sk باشد،آنگاه زمان لازم برای انتقال داده مورد نیاز زیر وظیفه s_j از طریق رابطه زیر محاسبه می‌شود:

$$\tau_{kj} = \frac{a_j}{sk} \qquad (3)$$

پس از منتقل شدن داده موردنیاز برای زیروظیفه s_j از RMS به منبع k،نوبت پردازش زیر وظیفه روی منبع می‌رسد،با فرض اینکه قدرت پردازش منبع k برابر xk باشد،زمان مورد نیاز برای پردازش زیر وظیفه s_j بروی منبع k عبارتست از:

$$T_{kj} = \frac{c_j}{xk} \qquad (4)$$

پس از پردازش وظیفه s_j، چون از تکنیک افزونگی برای افزایش قابلیت اعتماد استفاده شده بود در نتیجه چند کپی از وظیفه s_j به منابع ارسال شده است تا محاسبات را به صورت همزمان بر روی آن انجام دهند و چون قدرت محاسباتی و همچنین سرعت انتقال داده منابع می‌تواند متفاوت باشد پس وظیفهs_j بر روی منابع مختلف دارای زمان انتقال و پردازش مختلف خواهد بود، سریعترین آنها را انتخاب می‌کنیم. یعنی زمان اتمام زیر وظیفه s_j عبارتست از:

$$T_{s_j} = \mathrm{Min}(\tau_{kj}, T_{kj}); k \in \{R_1^k, R_2^k, \dots, R_{r_k}^k\} \qquad (5)$$

از آنجایی که برای اتمام شدن هر وظیفه ارسال شده توسط کاربر نیاز به اتمام تمامی زیر وظایفی دارد که آن وظیفه را تشکیل می‌دهند، پس در این مرحله باید منتظر تمامی زیروظایف متناظر یک وظیفه باشیم تا آنها نیز محاسبات خود را به اتمام برسانند پس در این مرحله برای بدست آوردن زمان اتمام وظیفه S یک ماکزیمم گیری از زمان اتمام زیر وظایف آن انجام می‌دهیم:

$$T_S = \mathrm{Max}\left(T_{s_j}\right); s_j \in \{S_1, S_2, \dots, S_n\} \qquad (6)$$

و در نهایت زمان مورد نیاز برای اتمام وظیفه بدست می‌آید.

در [11]، الگوریتم RFOH را برای زمان‌بندی وظیفه با تحمل‌پذیر خطا در گرید محاسباتی ارائه کردند. این روش تاریخچه‌ی رخداد خطا در منابع را در یک جدولی به نام جدول تاریخچه‌ی رخداد خطا در سرور اطلاعاتی گرید نگهداری می‌کند. هر سطر جدول FOHT برای هر منبع شامل دو ستون می‌باشد. یک ستون تاریخچه‌ی رخداد خرابی را در آن منبع

نشان می‌دهد و دیگری تعداد وظایف اجرا شده توسط آن منبع را مشخص می‌کند.

در [12]، الگوریتم IGA را برای حل مسئله زمان‌بندی وظایف وابسته ارائه دادند که در آن به 3 پارامتر کیفیت سرویس به طور هم زمان توجه شده است. این سه پارامتر عبارتند از زمان،هزینه و قابلیت اطمینان. چون این پارامترها با هم در تضاد هستند و نمی‌توانند هم زمان با هم بهبود یابند و بهبود یکی باعث کاهش کارآیی دیگری می‌شود به هر یک از پارامترها وزن داده می‌شود، به طوریکه وزن‌دهی یا توسط کاربر انجام می‌شود بدین صورت که هر کدام از پارامترها که برای کاربر ارزش بیشتری داشت وزن بیشتر و دیگری وزن کمتری می‌گیرد و یا اینکه وزن‌دهی به صورت تصادفی انجام می‌شود.

سانجایا کومار پاندا، پابیر تاموهان خیلار و دورگاپرا سادموها پاترا درسال 2015 [13]، الگوریتم FTMXT را برای زمانبندی وظایف در گرید ارائه دادند که در آن قابلیت تحمل پذیری خطا نیز در نظر گرفته شده است.از دو الگوریتم سنتی MCT و MET الهام گرفته شده است.در الگوریتم MCT براساس بهترین زمان تکمیل کارها را بین ماشین ها تقسیم می‌شود و اما در الگوریتم MET براساس بهترین زمان اجرا کارها را بین ماشین‌ها تقسیم می‌شود اما در این دو الگوریتم مسئله تحمل پذیری خطا در نظر گرفته نشده است لذا در الگوریتم FTMXT ،کومار پاندا و دوستان این مسئله را مدنظر گرفته‌اند. این الگوریتم هنگامی که کارها را بین منابع یا ماشین ها تقسیم می‌کند یا بر حسب زمان اجرا یا بر حسب زمان تکمیل کارها این عمل انجام می‌شود بگونه‌ای که در صورت خرابی ماشینی که کار به آن واگذار شده است دومین ماشینی که بهترین زمان اجرا یا بهترین زمان تکمیل را داراست انجام کار را بر عهده می‌گیرد و این عمل در صورت خرابی ماشین دوم هم تا ماشین بعدی ادامه پیدا می‌کند یعنی بر اساس تعداد ماشین ها این عمل ادامه پیدا می کند تا وظایف موجود در سیستم به نحوی به اتمام برسند که باعث افزایش قابلیت اطمینان شده است .

در مقاله [14] دو الگوریتم برای محیط محاسبات ابری ارائه کردند و آن را با سیاست پیش فرض ابزار cloudsim که پیچیدگی محاسباتی هر وظیفه را در نظر می‌گرفت، مقایسه کردند. الگوریتم Min-Min با مجموعه‌ای از وظایف زمانبندی نشده U شروع می‌شود. سپس وظیفه با حداقل زمان محاسباتی کلی انتخاب می‌شود و به منبع مربوطه تخصیص می‌یابد. سرانجام جدیدترین وظیفه زمانبندی شده از U حذف می‌شود و این فرایند تا زمانی که همه وظایف زمانبندی شوند، تکرار می‌شود. هر دو الگوریتم Min-Min و Max-Min یک تخصیص فرضی از وظایف به منابع درنظر می‌گیرند، زمانی که منابع براساس تخصیص فرضی بیکار شوند نگاشت انجام می‌شود.

در مقاله [15] یک الگوریتم بهینه‌سازی تعادل بار مبتنی بر کلونی مورچه، برای حل مسئله تعادل بار ماشین مجازی در فرآیند زمانبندی وظیفه ارائه شده است. این الگوریتم می‌تواند در محیط ابر پویا سازگار باشد و نه تنها زمان اجرای زمانبندی وظایف را کاهش می‌دهد بلکه تعادل بار ماشین‌های مجازی در مراکز داده را نیز فراهم می‌کند.

در مقاله [16] یک الگوریتم بهینه‌سازی اجتماع ذرات، برای زمانبندی وظایف در محیط ابر پیشنهاد شده که زمان پردازش را حداقل می‌کند. این الگوریتم با مجموعه وظایف بزرگتر همگرایی سریعتری دارد. فقدان تنوع PSO در همگرایی نابجای آن نقش دارد؛ بنابراین از PSO ترکیبی با عملیات متفاوت برای جلوگیری از همگرایی نابجا در مشکل زمانبندی وظایف استفاده میشود. تمرکز این مقاله، بر حداقل‌سازی زمان تکمیل کارها و همچنین حداکثر استفاده منابع است.

در مقاله [17] یک الگوریتم زمانبندی منابع برای ماشین مجازی ارائه شده که ظرفیت محاسباتی اجزای پردازش و پیچیدگی محاسباتی، در نظر گرفته شده است. در این مقاله از الگوریتم اجتماع ذرات بهبودیافته برای حل مشکل زمانبندی و بر طبق نیاز ابر، با محدود کردن سرعت ذره استفاده کرده است. نتایج نشان می‌دهد که این الگوریتم کارایی بهتری نسبت به نسخه اصلی داراست.

درمقاله [18] الگوریتم بهبودیافته مبتنی بر هزینه برای زمانبندی وظیفه ارائه شده است. این الگوریتم به منظور نگاشت کارآمد وظایف به منابع موجود در ابر مطرح شده است. دو فاز اصلی در این الگوریتم شامل استفاده از الگوریتم لانه زنبور

بهبودیافته* برای اختصاص اولویت به وظایف و سپس از الگوریتمی به منظور گروه‌بندی وظایف بر اساس اولویت آنها می‌باشد. این الگوریتم زمانبندی، هزینه صرف شده بابت در اختیار گرفتن منابع و کارایی محاسبات انجام شده برای اتمام وظایف جریان کار را محاسبه می‌نماید. در این الگوریتم نسبت هزینه صرف شده برای در اختیار گرفتن منابع به هزینه ارتباط کارآمد برای انجام وظایف جریان کار، بهبود قابل توجهی داشته است.

در مقاله [19] یک الگوریتم اکتشافی زمانبندی جریان کار، مبتنی بر بهینه‌سازی گروهی وظایف ارائه شده است. الگوریتم گروهی وظایف، از رفتار گروهی حیوانات، مانند حرکت دسته جمعی پرندگان و ماهی‌ها الهام گرفته شده است. به این جهت که این الگوریتم نیز با یک ماتریس جمعیت تصادفی اولیه، شروع می‌شود، عملکرد آن مشابه بسیاری دیگر از الگوریتم‌های تکاملی همچون الگوریتم ژنتیک پیوسته و الگوریتم‌های مبتنی بر رقابت می‌باشد اما برخلاف الگوریتم ژنتیک، الگوریتم گروهی وظایف هیچ عملگر تکاملی همانند جهش و تزویج ندارد. الگوریتم اکتشافی زمانبندی جریان کار، مبتنی بر بهینه‌سازی گروهی وظایف به منظور بهینه‌سازی گروهی وظایف، بر اساس رویکرد اکتشافی برنامه‌های کاربردی و با توجه به منابع موجود در ابر و با هدف کاهش زمان انجام محاسبات و کاهش زمان انتقال داده‌ها طراحی شده است. این الگوریتم از دو جزء اصلی که شامل استفاده از الگوریتم اکتشافی به منظور کشف صحیح منابع و سپس استفاده از الگوریتم بهینه‌سازی گروهی وظایف به منظور نگاشت صحیح این منابع به وظایف می‌باشد، تشکیل شده است. نتایج تجربی نشان می‌دهد که استفاده از این الگوریتم صرفه‌جویی در هزینه‌ها و توزیع متعادل حجم کار بر روی منابع را در برداشته است.

در مقاله [20] یک الگوریتم بهینه زمانبندی جریان کار ارائه شده است. این الگوریتم، به منظور زمانبندی جریان‌های کار در محیط ابر مطرح شده و ساختار کلی آن بر اساس یک درخت بیان می‌شود. این الگوریتم متشکل از دو فاز اصلی است که در فاز اول آن با استفاده از الگوریتم کشف منابع، تمامی منابع موجود برچسب خورده و سپس هر کدام از منابع، اطلاعات خود را به گره پدر خود که مستقیماً به آن متصل است ارسال می‌کند، بدین ترتیب کنترل منابع توسط هر گره پدر آسانتر خواهد شد، سپس در فاز دوم با توجه به فاکتور کیفیت سرویس QOS که توسط کاربر تعیین می‌شود، اختصاص منابع برای انجام وظایف جریان کار صورت می‌گیرد. به طور کلی این الگوریتم، راه‌حلی را به منظور زمانبندی جریان‌های کاری، با توجه به فاکتورهای QOS درخواستی کاربر ارائه می‌دهد. با استفاده از این الگوریتم، بهبود قابل توجهی در استفاده از CPU مشاهده شده است.

در مقاله [21] الگوریتم HPC برای برنامه‌ریزی وظیفه و تخصیص داده‌ها ارائه شده است که به نقش مهم داده‌ها و دسترسی به آنها در زمانبندی سیستم‌های توزیع شده اشاره کرده است. آنها تعداد وظایف هر برنامه را که نشان دهنده ارتفاع اجرایی پروسه مورد نظر می‌باشد را به عنوان یک عامل تأثیرگذار در زمانبندی درنظر گرفتند. بدین صورت که چه تعداد از پروسه‌های قبلی مربوط به این برنامه کاربردی تاکنون اجرا شده‌اند. این نشان می‌دهد که هر چقدر ارتفاع اجرایی پروسه مورد نظر بیشتر باشد امکان وجود دیتای مورد نیاز این پروسه نیز بیشتر خواهد بود و همچنین امکان به اتمام رسیدن کار این برنامه نیز بیشتر خواهد بود که می‌تواند آمار کیفی سیستم را بهبود ببخشد لذا این برنامه اولویت بیشتری برای اجرا خواهد داشت.

4. مقایسه و ارزیابی روش‌های مورد بررسی

در این بخش به مقایسه‌ای کوتاه و مفید از روش‌ها و تکنیک‌های مشابه با موضوع مورد مطالعه که پیش‌تر تشریح شده است پرداخته شده است. در جدول (1) مقایسه انجام شده از الگوریتم‌های یاد شده نشان داده شده است.

* Bee Colony Optimization (BCO)

جدول 1. مقایسه الگوریتم‌های مورد مطالعه با دیگر روش‌های مشابه

موضوع	هدف	مزایا	معایب
الگوریتم ژنتیکی برای زمانبندی با رویکرد کاهش زمان و هزینه[5]	زمانبندی با رویکرد کاهش زمان و هزینه	افزایش کارایی، کاهش مصرف انرژی	زمان انجام کار ضعیف، هزینه و استفاده از منابع
الگوریتمی بر مبنای دو الگوریتم پایه ای Min-Min و Max-Min با رویکرد کاهش زمان اجرا[6]	استفاده حداکثری از منابع در عین بهینه‌سازی ابزار (سود) با استفاده از پروتکل چانه‌زنی و توزیع کارها به VMهای معتبر	استفاده بهینه از منابع، توازن بار، کاهش Make Span، همگرایی مطلوب	وابستگی به پارامترهای اولیه، پیچیدگی در طراحی و فرموله‌سازی مساله
الگوریتمی بر مبنای تئوری صف برای زمانبندی و کاهش هزینه[7]	زمانبندی و کاهش هزینه در رایانش ابری به منظور توزیع متعادل بار موجود در بین مراکز داده	سرعت تخصیص و کیفیت تخصیص، کاهش مصرف انرژی	نیاز به حافظه بزرگ امکان کند بودن سرعت، پیچیدگی پیاده‌سازی
الگوریتم Genetic- Annealing با رویکر کاهش زمان کلی اجرا[8]	مدیریت سرویس‌دهنده با توجه به سطح فعالیت، توان مالی، سود، هزینه	بهبود زمان انتظار، زمان اجرا، سرعت و کارایی، استفاده حداکثر از منابع	میزان مصرف برق در دیتاسنتر و عدم پوشش مهاجرت زنده VM
الگوریتمی جهت زمانبندی گرید با رویکرد دوری گزیدن از افتادن در بهینه محلی[9]	بهبود تخصیص منبع در محیط ابرهای انجمنی	حداکثر استفاده از منابع، بهبود عملکرد VMها، ارزیابی کارایی مدل	پیچیدگی مدیریت منابع، فرایند طولانی برای رسیدن به پاسخ با هزینه و زمان
الگوریتمی برای رزرو منابع در جهت بهبود کیفیت سرویس[10]	رزرو منابع در جهت بهبود کیفیت سرویس	انجام پیش‌پردازش و شناخت منابع برای تخصیص، تحمل‌پذیری خطا، بهینه‌سازی توازن بار	عدم معماری مدون برای رسیدگی و سرویس‌دهی به درخواست‌های بیشمار، پیاده-سازی سخت
الگوریتمی جهت زمانبندی گرید با رویکرد بهبود قابلیت اطمینان[11]	پیدا کردن ترکیبی بهینه از سرویس‌ها برای وظایف ورودی	کاهش مصرف انرژی مرکز داده، قابلیت اطمینان سرویس با توازن توان	غیر متمرکز بودن بار مناسب شبکه های بزرگ
الگوریتم RFOH برای زمانبندی وظایف با تحمل پذیری خطا[12]	زمانبندی وظایف با تحمل پذیری خطا	همگرایی سریع، بهبود زمان پاسخ‌دهی، انعطاف‌پذیری بالا، بهبود مصرف انرژی	هزینه بالای پیاده‌سازی با توجه به پیچیدگی و ترکیب تکنیک‌های مختلف،،متکی بر انتخاب درست جمعیت اولیه
الگوریتم IGA برای زمانبندی وظایف و بهبود سه پارامترکیفیت به طور همزمان[13]	برای زمانبندی وظایف و بهبود سه پارامترکیفیت به طور همزمان	سادگی، حداکثر استفاده از منابع، توازن بار	نتساب وظایف به پردازندهای که زمان اجرای نامناسب، زمان تکمیل کار بالا

4. نتیجه‌گیری و جمع‌بندی

مدیریت منابع و تخصیص وظایف در کنار امنیت، قابلیت اطمینان و حفظ اعتماد مشترکین یکی از چالش‌های مهم در زمینه رایانش ابری می‌باشد که می‌تواند روی سایر مسائل نیز تاثیرگذار باشد. واضح است که تعداد کارها و تعداد منابع در محیط ابر می تواند بسیار گسترده باشد و به همین علت ترتیب اجرای کارها و نحوه تخصیص منابع تاثیر مهمی برکارآیی

سرور ابر دارد. از آنجائیکه مزیت های سرویس‌های ابری بسیار می‌باشند، تمایل به استفاده از محیط رایانش ابری به سرعت در حال افزایش می‌باشد و به نوبه خود رقابت بین ذینفعان به منظور دستیابی به این منابع را نیز افزایش می‌دهد. البته باید توجه نمود که در محیط ابر مواردی همچون نا همگونی، عدم اطمینان و پراکندگی منابع به عنوان جزئی از دشواری‌های اختصاص منابع ایجاد می‌گردند که ذیل مدیریت منابع، نیازمند بررسی و بازبینی اختصاصی دارند. مدیریت منابع در محیط ابری از یک فرآیند چند مرحله‌ای تشکیل شده که از ارسال وظایف تا اتمام اجرای آنها در منابع را در بر می‌گیرد. گردآوری منابع به عنوان مرحله ای به منظور شناسایی منابع مکفی بابت یک میزان کار مشخص و بر اساس الزامات کیفیت سرویس توسط بهره‌برداران ابر، تعریف می‌گردد. اما زمانبندی منابع، قرار دادن و اجرای وظایف بهره برداران ابر بر پایه منابع انتخاب شده و از مسیر تهیه منابع می باشد. منظور زمانبندی وظایف مشخص نمودن روش تخصیص وظایف کاربران به ماشین‌های مجازی بوده و وظیفه اصلی در ایجاد اثر بخشی و بهره‌وری در سیستم را با هدف کامل نمودن کار در کم‌ترین زمان ممکن و با بهره‌برداری بهینه از بیش‌ترین ابزار را دارد لذا استفاده از الگوریتم‌های برنامه‌ریزی هوشمند که قابلیت چیره شدن بر مشکلات و کاستی‌های ارائه خدمات در شبکه طراحی ابر را داشته باشند، ضروری می‌باشد. در فضای خدمات زیر ساخت رایانش ابری، زمانبندی وظایف به فرآیند اختصاص بهینه فعالیت‌های محاسباتی و منابع محاسباتی درخور ذیل تنگناهای خاص اشاره می‌نماید. می‌توان این مراحل را به عنوان نگاشت، مابین فعالیت‌های محاسباتی و منابع محاسباتی محیط ابری در نظر گرفت که این موضوع با فرض برآورده شدن از مقاصد بهینه‌سازی منحصر به فرد لحاظ می‌گردد. همچنین عامل مصرف انرژی که تاثیر مستقیم بر حداکثر بهره‌برداری از منابع و کاهش وضعیت‌های بیکار منابع دارد، حلقه دیگری از پیچیدگی در زمان‌بندی وظایف را به دنبال خواهد داشت. فرآیند زمان‌بندی است که در چنین ساختارهای پیچیده‌ای، وظایف کاربر را به منظور کم کردن زمان به پایان رساندن کار با لحاظ نمودن مدیریت مصرف انرژی به ماشین‌های مجازی اختصاص می‌دهد. مقاصد زمان‌بندی احتمالا با توجه به ناهمگونی منابع، احتیاج به کاستن مخارج کاربران، بهره‌برداری حداکثری از منابع به صورت بهینه، کاستن مدت زمان پایان یافتن اجرای وظایف و مدیریت مصرف انرژی در مراکز داده با یکدیگر در تداخل می‌باشند. در حقیقت، در محیط ابر درهر لحظه بصورت همزمان تعدادی درخواست داریم که برای پاسخ به هریک از آنها باید یک یا چند کار انجام دهیم و هریک از این کارها به منابع خاصی نیاز دارند، که در این رابطه روش‌های مختلفی ارائه شده است که زمانبندی نامیده می‌شود. زمانبندی، به معنی واگذاری کارآمد و مناسب منابع به کارها است. هدف اصلی آن، کوتاه کردن زمان تکمیل کار، بالا بردن توان عملیاتی سیستم و ایجاد تعادل بار روی منابع است. این مسئله در ابر به دلیل مقیاس بزرگ منابع، پیچیده‌تر هم می‌شود؛ بنابراین شناخت بهتر الگوریتم‌ها می‌تواند در انتخاب الگوریتم مناسب کمک زیادی کند. در این مقاله با آشنایی با روش‌های و تکنیک‌های مختلف زمانبندی در ابر توانستیم به نقاط قوت و چالش‌های موجود دست بیابیم تا برای ارائه روش‌ها و رویکردهای جدید، افق‌های روشنی در پیش ذهن داشته باشیم. نتایج بدست آمده در این تحقیق را می‌توان چنین ذکر کرد:

1) شناخت مفاهیم مرتبط با زمانبندی درخواست مبتنی بر بستر ابر
2) بررسی روش‌ها و تکنیک‌های مختلف در جهت بهبود زمانبندی درخواست و بهبود آن
3) شناخت چالش‌های زمانبندی درخواست و رویکردهای کاهش مصرف انرژی و بهبود کیفیت سرویس
4) کنکاش و بهبود زمان اجرا و دیگر پارامترهایی که در هزینه‌های زیرساخت ابری تأثیرگذار هستند

با توجه به بررسی‌ها و تحقیقات انجام شده در این حوزه همچنان نیازهای تعریف شده مرتفع نشده‌اند، بعبارتی نیازها منعطف و حجیم هستند و روش‌های مورد استفاده نیز تا حد ممکن پوشش‌دهی این نیازمندی‌ها را انجام می‌دهند. با توجه به این که در این مقاله نقاط قوت و ضعف مسئله مورد نظر شناسایی شد برای تحقیقات آینده استفاده از تکنیک‌های یادگیری عمیق برای زمانبندی درخواست‌ها پینهاد می‌شود. همچنین استفاده از تکنیک‌های فراابتکاری می‌تواند کمک زیادی در

راستای بهینه‌سازی به ما نماید. تعدادی از فعالیت‌ها که می‌تواند باعث بهبود تحقیقات انجام شده گردد و در واقع نوعی محدودیت برای روش‌های مورد بررسی در این مقاله محسوب می‌شود به شرح زیر است.

1) استفاده از الگوریتم‌های فرااکتشافی جهت بهبود فرآیند زمانبندی
2) اعمال سیاست زمانبندی غیرانحصاری جهت اعمال گزینه مدت زمان باقی‌مانده از اجرا در انتخاب برنامه و بررسی معیارهای کیفیت خدمات
3) بررسی نتایج روش‌های زمانبندی درخواست به لحاظ کاهش مصرف انرژی، بهبود کیفیت سرویس
4) بررسی بهبود زمان اجرا و دیگر پارامترهای مرتبط با هزینه‌های زیرساخت ابری تأثیرگذار

مراجع

1. Ali, I. M., Sallam, K. M., Moustafa, N., Chakraborty, R., Ryan, M., & Choo, K. K. R. (2020). An automated task scheduling model using non-dominated sorting genetic algorithm II for fog-cloud systems. IEEE Transactions on Cloud Computing, 10(4), 2294-2308.
2. Houssein, E. H., Gad, A. G., Wazery, Y. M., & Suganthan, P. N. (2021). Task scheduling in cloud computing based on meta-heuristics: review, taxonomy, open challenges, and future trends. Swarm and Evolutionary Computation, 62, 100841.
3. Sathishkumar, P., Kumar, N., Raju, S. H., & Victoria, D. R. S. (2024). An intelligent task scheduling approach for the enhancement of collaborative learning in cloud computing. Sustainable Computing: Informatics and Systems, 43, 101024.
4. Abualigah, L., Hussein, A. M., Almomani, M. H., Zitar, R. A., Migdady, H., Alzahrani, A. I., & Alwadain, A. (2024). Improved Jaya Synergistic Swarm Optimization Algorithm to Optimize Task Scheduling Problems in Cloud Computing. Sustainable Computing: Informatics and Systems, 101012.
5. Ibrahim, M., Nabi, S., Baz, A., Alhakami, H., Raza, M. S., Hussain, A., ... & Djemame, K. (2020)., Improving safety and availability of complex systems using a risk-based failure assessment approach, IEEE Access, 8, 128282-128294.
6. Benjamin Khoo, B.T., Veeravalli, B., Hung, T., Simon See, C.W., (2018), A multi-dimensional scheduling scheme in a Grid computing environment, Journal of Parallel and Distributed Computing 67, pp. 659-673.
7. Entezari-Maleki, R. and Movaghar, A., (2021), A Genetic Algorithm to Increase the Throughput of the Computational Grids, InternationalJournal of Grid and Distributed Computing, 4(2), pp. 11-24.
8. Wang, Y., Zheng, Q., Guo, M., Xiao, H., & Bai, Y. (2024). An optimal grid scheduling method considering coupling degree of nodes and duality of reciprocal inverse mapping. Results in Engineering, 102813.
9. Abdulal, W., Jabas, A., Ramachandram, S., Al Jadaan, O.,(2022),” Task Scheduling in Grid Environment Using Simulated Annealing and Genetic Algorithm. Book: Grid Computing-Technology and Applications”, Widespread Coverage and New Horizons, pp. 89-110.
10. Khanli, L. M., Far, M. E., & Ghaffari, A. (2021). Reliable job scheduler using RFOH in grid computing. Journal of Emerging Trends in Computing and Information Sciences, 1(1), 43-47.

11. Cheung, R., Roshandel, N., Medvidovic, (2020). Early Prediction of Software Component Reliability, ICSE' 08, Leipzig, Germany, pp. 111-120.
12. Panda, S. K., Khilar, P. M., & Mohapatra, D. P. (2019). FTMXT: Fault-Tolerant Immediate Mode Heuristics in Computational Grid. In Informatics and Communication Technologies for Societal Development: Proceedings of ICICTS 2014 (pp. 103-113). Springer India.
13. Ye, L., Xia, Y., Yang, L., & Zhan, Y., (2015), Efficient task scheduling algorithms for cloud computing environment. . IEEE Transactions on Automation Science and Engineering, 20(4), 2594-2606.
14. Sun, J., Gu, Q., Zheng, T., Dong, P., Valera, A., & Qin, Y. (2020). Joint optimization of computation offloading and task scheduling in vehicular edge computing networks. Ieee Access, 8, 10466-10477.
15. Lee, S., Lee, S., & Lee, S. S. (2021). Deadline-aware task scheduling for IoT applications in collaborative edge computing. IEEE Wireless Communications Letters, 10(10), 2175-2179.
16. Alahmad, Y., Daradkeh, T., & Agarwal, A. (2021). Proactive failure-aware task scheduling framework for cloud computing. IEEE Access, 9, 106152-106168.
17. Tang, J., Jalalzai, M. M., Feng, C., Xiong, Z., & Zhang, Y. (2022). Latency-aware task scheduling in software-defined edge and cloud computing with erasure-coded storage systems. IEEE Transactions on Cloud Computing, 11(2), 1575-1590.
18. Kruekaew, B., & Kimpan, W. (2022). Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning. IEEE Access, 10, 17803-17818.
19. Lou, J., Tang, Z., Zhang, S., Jia, W., Zhao, W., & Li, J. (2022). Cost-effective scheduling for dependent tasks with tight deadline constraints in mobile edge computing. IEEE Transactions on Mobile Computing, 22(10), 5829-5845.
20. Lipsa, S., Dash, R. K., Ivković, N., & Cengiz, K. (2023). Task scheduling in cloud computing: A priority-based heuristic approach. IEEE access, 11, 27111-27126.
21. Mangalampalli, S., Karri, G. R., Mohanty, S. N., Ali, S., Almari, A. M., & Alqahtani, S. A. (2024). Efficient Hybrid DDPG task scheduler for HPC and HTC in cloud environment. IEEE Access.