

توزیع منابع در رایانش ابری با رویکرد فراتکاملی

امید حاجبی سرگروئی

کارشناسی ارشد مهندسی کامپیوتر گرایش نرم‌افزار دانشگاه آزاد اسلامی واحد کرمان

چکیده

رایانش ابری مدلی است برای فراهم کردن دسترسی آسان بر اساس تقاضای کاربر از طریق شبکه به مجموعه ای از منابع رایانشی قابل تغییر و پیکربندی (مثل: شبکه‌ها، سرورها، فضای ذخیره‌سازی، برنامه‌های کاربردی و سرویس‌ها) که این دسترسی بتواند با کمترین نیاز به مدیریت منابع یا نیاز به دخالت مستقیم فراهم کننده سرویس به سرعت فراهم شده یا آزاد (رها) گردد. مدل رایانشی بر پایه شبکه‌های رایانه‌ای مانند اینترنت است که الگویی تازه برای عرضه، مصرف و تحویل خدمات رایانشی (شامل زیرساخت، نرم‌افزار، بستر، و سایر منابع رایانشی) با به‌کارگیری شبکه ارائه می‌کند. «رایانش ابری» از ترکیب دو کلمه رایانش و ابر ایجاد شده است. هدف اصلی در استفاده از رایانش ابری، کاهش هزینه‌های استفاده از منابع است. منابع محاسباتی در این سیستم‌ها، به‌عنوان ماشین‌های مجازی ارائه می‌شوند و هر درخواستی که از طرف کاربران به ارائه‌دهندگان خدمات فرستاده شود بخشی از منابع آن‌ها را به خود اختصاص می‌دهد. اگرچه تلاش‌هایی برای توزیع منابع به کارهای مشتری در محیط رایانش ابری با الگوریتم‌های مختلف و در نظر گرفتن پارامترهای موجود در موافقت‌نامه سطح سرویس‌ها انجام شده است، اما بسیاری از آن‌ها از سرعت پایین در زمان‌بندی برخوردارند. در این پژوهش، یک الگوریتم فراتکاملی بهبودیافته با الهام از الگوریتم فاخته چندهدفه را معرفی می‌کنیم و سعی در بهبود پارامترهای زمان پاسخ، زمان انتظار، هزینه و انرژی داریم. در نهایت الگوریتم پیشنهادی را با الگوریتم‌های زمان‌بندی بر اساس پارامترهای زمان پاسخ، زمان انتظار، هزینه و انرژی را در محیط شبیه‌سازی کلود سیم باهم مقایسه و بررسی می‌کنیم. روش پیشنهادی از کارایی بهتری نسبت به سایر الگوریتم‌های چندهدفه که مورد بررسی و مقایسه قرار گرفته بودند برخوردار است. روش پیشنهادی بر این اساس عمل می‌کند که جایگشتی از ماشین‌های مجازی به‌عنوان ماشین‌های انتسابی به کارها به همراه خودکارها به‌عنوان ورودی در نظر گرفته می‌شوند.

کلمات کلیدی: رایانش ابری، فراتکاملی، الگوریتم

مقدمه

ارائه‌دهندگان سرویس‌های ابر ابتدا یک سیستم محاسباتی به نام ابر را ایجاد می‌کنند که در آن چندین ماشین مجازی را از طریق این بستر ایجاد می‌کنند، که وظیفه نهایی آن‌ها پردازش کار کاربران است. در محیط رایانش ابری چندین ماشین مجازی می‌توانند منابع فیزیکی (CPU، حافظه و پهنای باند) را در یک میزبان فیزیکی مشترک به اشتراک بگذارند و ماشین‌های مجازی متعدد می‌توانند با استفاده از مجازی‌سازی شبکه، پهنای باند یک مرکز داده را به اشتراک بگذارند (Rehan, H). (۲۰۲۴).

از آنجایی که بسیاری از کاربران و برنامه‌های کاربردی نیاز به اشتراک منابع سیستم دارند، ارائه یک برنامه زمان‌بندی مناسب برای استفاده منابع و عملکرد سیستم بسیار دشوار است. بسیاری از پارامترهای سیستم، مانند قدرت پردازنده، فضای حافظه و پهنای باند شبکه، بر کارایی الگوریتم‌های زمان‌بندی منابع تأثیر می‌گذارند (Putzier, et al) (۲۰۲۴).

توزیع منابع^۱ در رایانش ابری به دو صورت ثابت و پویا انجام می‌شود. در توزیع منابع به صورت ثابت منابع به صورت ثابت به درخواست‌ها توزیع داده می‌شود و در ادامه نیز بهینگی منابع، زمان پاسخ و پردازش، مصرف انرژی و ... بررسی نمی‌گردد. مزیت این روش‌ها کم بودن سربار سیستم و پایین بودن هزینه‌ی مدیریت منابع است.

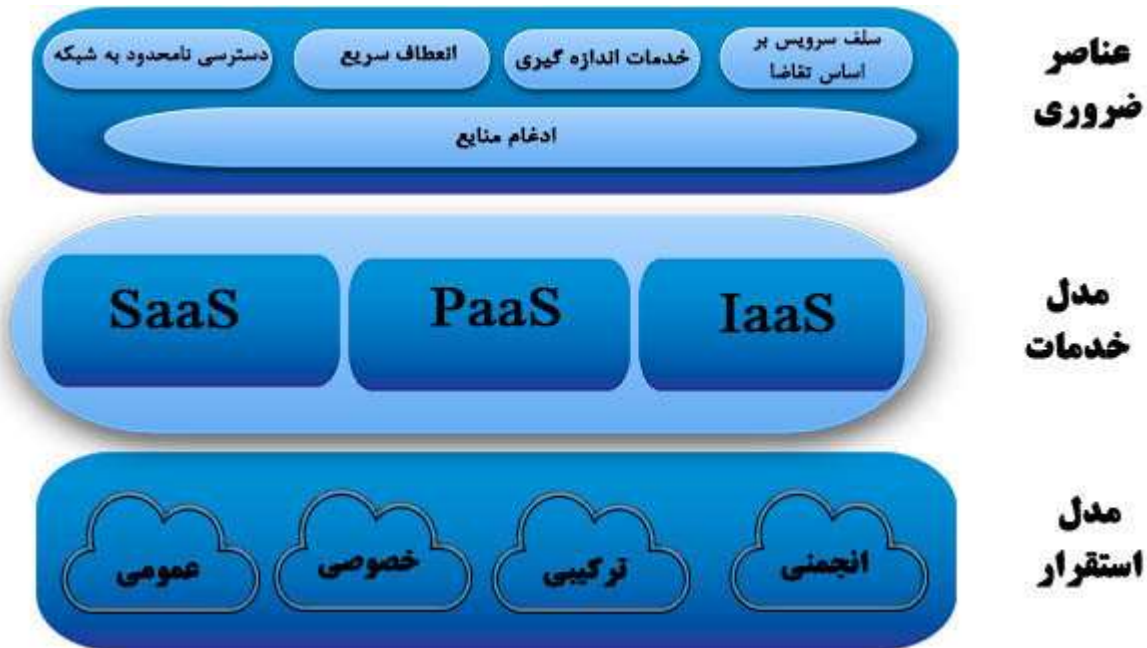
در توزیع منابع به صورت پویا، هدف اصلی این است که از اتلاف منابع به دلیل کم بودن بهینگی آن‌ها و همچنین بالا رفتن زمان پاسخ درخواست‌های مشتری به دلیل استفاده‌ی بیش‌ازحد از منابع جلوگیری گردد (Ra'afat, 2013). حمایت از توسعه فناوری یکی از مهم‌ترین وظایف و دغدغه‌های دولت‌هاست؛ که در خصوص رایانش ابری شاید دولت‌ها باید این نقش را ایفا کنند. چراکه دولت‌ها به‌طور بالقوه یکی از بزرگ‌ترین ذینفعان رایانش ابری خواهند بود. امکانات رایانش ابری (به‌ویژه در بخش زیرساخت به‌عنوان خدمت) می‌تواند در توسعه دولت الکترونیکی مستقل و یکپارچه نقش اساسی ایفا کند. خبرگان برخی دولت‌ها مانند دولت هند توسعه رایانش ابری را راهی برای پر کردن شکاف دیجیتالی و در پی آن کاهش تبعیض و نابرابری اجتماعی در جامعه خود یافته‌اند. کشورها همچنین می‌توانند به رایانش ابری به مثابه یکی از ارکان توسعه صادرات دانش مبنا بنگرند. سیاستگذاران کشورهای توسعه یافته توانایی خود در حفظ سلطه را در توفیق در این عرصه جستجو می‌کنند. پروژه‌های نظامی و دولتی بسیاری در کشورهای پیشرو در جریان است که در رأس آن‌ها پروژه‌های دولت ایالات متحده آمریکا قرار دارد (MT, I) (۲۰۲۴). دولت‌ها در کشورهای مختلف در زمینه تشویق به پذیرش رایانش ابری و تسهیل استفاده از خدمات مبتنی بر ابر توسط محققان و بخش کسب‌وکار و بخش‌های مختلف دولت نقش مهمی ایفا می‌کنند. برای نمونه، در ایالات متحده مؤسسه علوم ملی طبق برنامه‌ای که در سال ۲۰۰۸ آغاز شد، استفاده از خدمات ابر توسط محققان را ترویج می‌کند. دولت‌ها در انگلستان، ژاپن و سوئد نیز استفاده از ابر توسط محققان و کسب‌وکار را تسهیل و تشویق می‌کنند (Abd Al Ghaffar, 2024).

در طول دهه گذشته، پیشرفت‌های محاسباتی و فن‌آوری‌های مجازی منجر به تحقق مقرون‌به‌صرفه مراکز داده در مقیاس بزرگ شده است که بخش بزرگی از برنامه‌های کاربردی و پردازشی امروزی را اداره می‌نماید. این امکان به زیرساخت مرکز داده‌ها اجازه داد تا به اشخاص ثالث به‌طور سودآور آن‌ها را اجرا داده شوند. بنابراین با ظهور رایانش ابری استخری از منابع محاسباتی بین برنامه‌های کاربردی که از طریق اینترنت به دست می‌آیند، به اشتراک گذاشته می‌شوند. محاسبه ابری، نه تنها توسط جامعه فن‌آوری بلکه عموم مردم تبدیل به یک اصطلاح گسترده و محبوب شده است. در رایانش ابری تعداد زیادی منابع مانند واحدهای محاسباتی، فضای ذخیره‌سازی و نرم‌افزار باقیمت نسبتاً پایین را در اختیار مشتری قرار می‌دهد. با افزایش تعداد مشتریان، برآورده کردن نیازهای آن‌ها و توزیع منبع موضوع مهمی است که نیاز به بررسی کاملی دارد (Jennings, 2015) و (Tembhurne, P. S) (۲۰۲۴).

تعریف رایانش ابری

موسسه ملی فناوری و استانداردها (NIST)^۲ رایانش ابری را این‌گونه تعریف می‌کند:

"رایانش ابری مدلی است برای داشتن دسترسی فراگیر، آسان و بنا به سفارش شبکه به مجموعه‌ای از منابع رایانشی پیکربندی پذیر (مثل: شبکه‌ها، سرورها، فضای ذخیره‌سازی، برنامه‌های کاربردی و سرویس‌ها) که بتوانند با کمترین کار و زحمت یا نیاز به دخالت فراهم‌کننده سرویس به سرعت فراهم‌شده یا آزاد (رها) گردند". بنابراین رایانش ابری از دید زیرساخت، به‌گونه‌ای سیستم‌های توزیع‌شده و موازی اطلاق می‌گردد که مجموعه‌ای از رایانه‌های مجازی را که به یکدیگر متصل هستند شامل می‌شود. این رایانه‌ها به‌طور پویا عرضه‌شده و به‌عنوان یک یا چند منبع محاسباتی یکپارچه بر اساس توافقات سطح سرویس ارائه می‌شوند. این توافقات در طول انجام مذاکرات سرویس‌دهندگان و مصرف‌کنندگان برقرار می‌گردند. رایانش ابری سعی دارد نسل جدیدی از مراکز داده‌های راه، با ارائه کردن سرویس‌ها و خدمات در ماشین‌های مجازی شبکه شده به‌صورت پویا، به‌گونه‌ای ممکن سازد که ارائه‌دهندگان خدمات کاربردی بتوانند سرویس‌ها و برنامه‌های کاربردی را با انعطاف‌پذیری و سهولت بیشتری ارائه کنند و کاربران نیز بتوانند از هرجایی از دنیا به برنامه‌های کاربردی دسترسی داشته باشند (NIST, 2009).



شکل ۱. تعریف NIST از محاسبات ابری

مدل سرویس ابری

ابری خدمات خود را در قالب سه نوع ارائه می‌دهد:

نرم‌افزار به‌عنوان سرویس^۳ (SaaS)

ارائه‌دهنده ابری SaaS را از طریق اینترنت برای زمانی که کاربر می‌خواهد بدون نیاز به نصب آن بر روی کامپیوتر خود اجرا کند ارائه می‌دهند. این هزینه به‌روزرسانی و تعمیر و نگهداری را از کاربر نهایی حذف، و باعث کاهش قیمت خرید نرم‌افزار و حذف نیاز به مجوز نرم‌افزارهای ناخوشایند می‌شود.

ارائه‌کننده و کاربر با استفاده از مدل استفاده درازای پرداخت برای دسترسی از هر ترمینال با یک سیستم‌عامل و مرورگر وب بدون محدودیت فیزیکی به‌صورت خودکار از آن استفاده خواهند کرد.

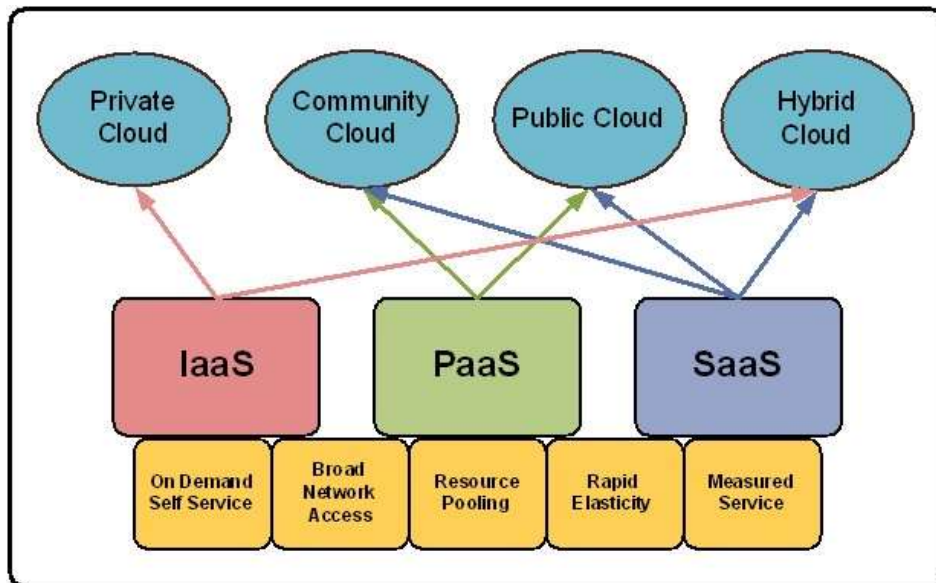
³ Software-as-a services

پلتفرم به‌عنوان سرویس (PaaS)^۴

با PaaS، ارائه‌دهندگان ابر به مشتریان خود یک محیط پیچیده فناوری اطلاعات برای توسعه برنامه‌های کاربردی که مصرف‌کننده بدون استفاده از کنترل شبکه یا سرور از آن استفاده می‌کند، فراهم می‌کند. این خدمات شرکت را با یک پلت فرم توسعه برای ایجاد برنامه‌های خود به‌عنوان مثال، (Google App Engine) ارائه می‌دهد.

زیرساخت به‌عنوان سرویس (IaaS)^۵

با IaaS، ارائه‌دهندگان ابر، کاربر نهایی را با حافظه ذخیره‌سازی، سرور و شبکه بر پایه پرداخت به ازای هر استفاده، که هزینه را کاهش می‌دهد و برای کسب‌وکار سودمند است فراهم می‌کند. این سرویس‌ها این شرکت را قادر به اضافه کردن یا حذف خدمات به راحتی می‌کند، مانند Amazon (Zhan, 2015).



شکل ۲. مدل استقرار ابر

منابع ذخیره‌سازی^۶

ارائه‌دهندگان عمومی ابر، مانند آمازون، خدمات ذخیره‌سازی مداوم از انواع مختلفی را، اعم از دیسک‌های مجازی و خدمات پایگاه داده برای انبارهای کالا ارائه می‌دهند، که هر یک از خدمات سطوح مختلف تضمین انجام داده‌ها و قابلیت اطمینان را دارا می‌باشند (Abdulwahab, et al ۲۰۲۴).

یک مسئله دشوار برای خدمات ذخیره‌سازی این است که چگونه به انعطاف‌پذیری برسیم، به طوری که یک سرویس به صورت پویا با تعداد بیشتری از کاربران، بار، و یا حجم اطلاعات موردسجش قرار گرفته و، به طور مشابه کاهش یابد، اگر این معیارهای به جهت مخالف تمایل پیدا کنند. دستیابی به مقیاس در سیستم‌های پایگاه داده‌های سنتی دشوار است، پایگاه‌هایی ثبات داده‌های قوی و ظرفیت اتمی، قوام، جداسازی و دوام (ACID)^۷ ویژگی‌های کاربردی را ارائه می‌دهند. خوشبختانه، بسیاری از برنامه‌های کاربردی وب میزبان ابر، مانند وبلاگ‌ها، ضعیفی سطح سازگاری، به‌عنوان مثال، سازگاری احتمالی را تحمل می‌کنند. این اجازه می‌دهد تا طراحان در چنین سیستم‌هایی به بهره‌برداری از تجارت بین عملکرد (زمان پاسخ، در دسترس بودن) و ثبات بپردازند. این امر منجر به توسعه طیف وسیعی از فن‌آوری ذخیره‌سازی به اصطلاح 'اطلاعات' No Sql " بهینه‌سازی شده برای شرایط مختلف عملیاتی و کاربردی می‌گردد (Atadoga, et al ۲۰۲۴).

⁴ Platform-as-a services

⁵ Infrastructure-as-a-services

⁶ Storage Resource

⁷ Atomicity, Consistency, Isolation and Durability

منابع قدرت^۸

مراکز داده‌ها برای بخش قابل توجهی از مصرف انرژی در سراسر جهان به حساب آمده و هزینه‌های انرژی، هر دو هزینه‌های مستقیم آب و برق و توزیع قدرت مربوطه و هزینه‌های خنک‌کننده، بخش قابل توجهی از هزینه‌های کلی برای اپراتور مرکز داده‌ها محسوب می‌گردند. در یک مرکز داده، قدرت توسط خود سرورها، و همچنین توسط تجهیزات شبکه، تجهیزات توزیع برق، زیرساخت‌های خنک‌کننده و زیرساخت‌های حمایتی (روشنایی و غیره) مصرف می‌شود. معمولاً مراکز داده منبع قدرت از یک یا چند ارائه‌دهنده انرژی تغذیه می‌کند، با یک رویکرد اخیر به سمت تولید / ذخیره‌سازی قدرت‌های محلی، به‌ویژه از منابع انرژی تجدید پذیر از جمله باد و خورشید. با توجه به هزینه‌های مرتبط با تأمین انرژی مراکز داده، تمرکز عمده روی به حداقل رساندن مصرف انرژی است. چهار روش اصلی وجود دارد: توسعه اجزای کم قدرت برای بهبود بهره‌وری انرژی ساخت‌افزار؛ تکنیک‌های توسعه برای مدیریت منابع انرژی آگاه؛ اجرای برنامه‌های کاربردی به شیوه‌ای کارآمد؛ و توسعه سیستم‌های خنک‌کننده کارآمدتر و محل مراکز داده در مناطق جغرافیایی با شرایط آب و هوایی سودمند (Jennings, 2015).

مدل توزیع منابع پویا مبتنی بر توافق^۹

SLA توافقی است که QoS را بین سرویس‌دهنده و سرویس‌دهنده مشخص می‌کند و شامل قیمت خدمات با سطح QoS باقیمت سرویس می‌شود. بیشتر مدل‌های توزیع منابع در محیط محاسبات ابری بر برآورده کردن خدمات مشخص شده SLA برای کاربر ابر تمرکز می‌کنند. برخی از استراتژی‌های مدل‌های دیگر در توزیع منابع بر اهداف ارائه‌دهنده ابر تأکید دارند، که می‌تواند بر برخی از الزامات کاربران و سطح QoS ارائه‌شده تأثیر منفی داشته باشد (Son, 2013).

مدل توزیع منابع پویا مبتنی بر بازار^{۱۰}

استفاده از اقتصاد بازار برای مدیریت توزیع منابع به‌طور گسترده در گذشته مورد مطالعه قرار گرفته است، و چندین محقق، جنبه‌های اقتصادی محاسبات ابر را از دیدگاه‌های مختلف بررسی کرده‌اند. برای مقابله با تقاضای منابع دائمی، نوسانات بازار، توزیع منابع پیشنهاد شده است، و توسط بسیاری از ارائه‌دهندگان عمومی IaaS از جمله آمازون اجرا شده است. در این محیط، ارائه‌دهنده ابر می‌تواند از رویکرد بازار کالا یا مکانیزم‌های مبتنی بر حراج استفاده کند، باهدف اصلی دستیابی به حداکثر درآمد درحالی که حداقل هزینه را دنبال می‌کند (Amazon, 2009).

مدل توزیع منابع پویا مبتنی بر استفاده^{۱۱}

به‌منظور غلبه بر کمبود منابع که از توزیع منابع ثابت به برنامه‌ها و خدمات منجر شده است، رویکرد اصلی روش‌هایی که در این دسته قرار می‌گیرند این است که به‌صورت پویا VM ها را مدیریت کنند تا حداکثر بهره‌برداری از منابع را در پی داشته باشد و هزینه را به حداقل برساند. یک مدل که VM را با توجه به نیازهای واقعی برنامه اجرا می‌کند، توسط Lin و همکارانش ساخته شده است و بر اساس آستانه است. الگوریتم پیشنهادی از نظارت و پیش‌بینی نیازهای برنامه‌های ابر استفاده می‌کند که منجر به افزایش استفاده از منابع و کاهش هزینه‌ها می‌شود (Lin, 2011).

الگوریتم‌های فرا تکاملی

یک الگوریتم فرا تکاملی، یک چهارچوب است که می‌تواند با تغییرات کم، برای مسائل بهینه‌سازی مختلف به کار رود (اصغرپور، ۱۳۹۴). استفاده از الگوریتم‌های فرا تکاملی، به‌طور قابل‌ملاحظه‌ای توانایی یافتن جواب‌های باکیفیت بالا را برای مسائل بهینه‌سازی ترکیبی افزایش می‌دهد. به عبارت دیگر یک الگوریتم فرا تکاملی یک روش ابتکاری است که قادر به جستجوی

⁸ Power Resource

⁹ SLA-Based Dynamic RA Models

¹⁰ Market-Based Dynamic RA Models

¹¹ Utilization-Based RA Models

فضای جواب، برای یافتن جواب‌های باکیفیت بالاست. هدف مشترک تمامی الگوریتم‌های فرا تکاملی، حل مسائل بهینه‌سازی سخت مشهور هستند (اقبال نیا و دلیران، ۱۳۹۸).

روش‌های فرا تکاملی، دارای ویژگی‌های زیر هستند:

این روش‌ها، همگی تا حدودی احتمالی هستند، چنین رویکردی، این امکان را می‌دهد که از قرار گرفتن الگوریتم در دام بهینه‌ی محلی جلوگیری شود و معمولاً برای حل مسائل گسسته ارائه شده است اما امکان استفاده در مسائل پیوسته را نیز دارد و معمولاً از مفاهیم زیست‌شناسی، رفتارشناسی جانوری و فیزیک الهام گرفته‌اند.

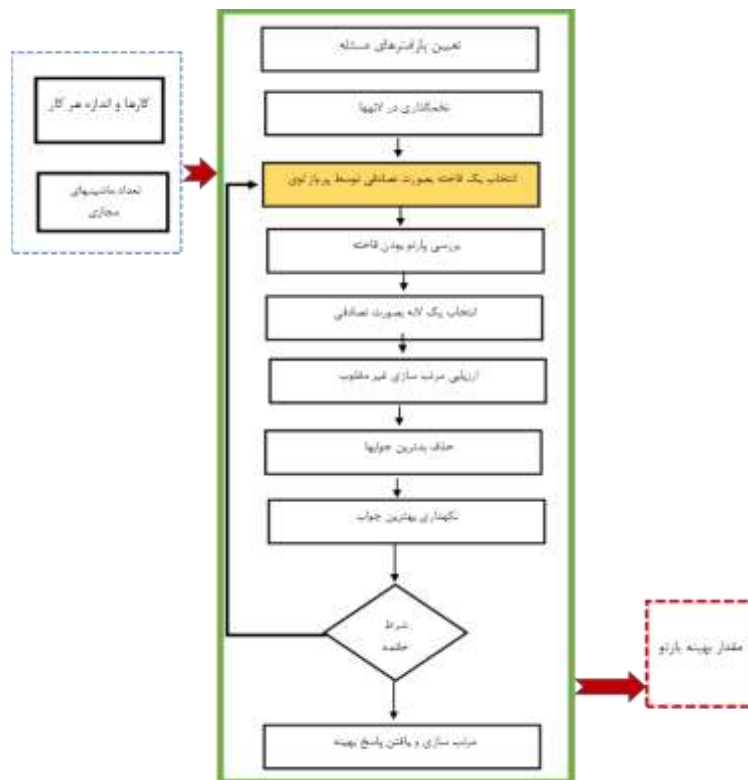
یکی از عیب‌های مشترک در این روش‌ها، دشوار بودن تنظیم و تطبیق پارامترهاست.

همان‌طور که گفته شد، الگوریتم‌های فرا تکاملی، به دو گروه کلی، الگوریتم‌های مبتنی بر یک جواب و الگوریتم‌های مبتنی بر جمعیت تقسیم می‌شوند، بدین‌صورت که الگوریتم‌های مبتنی بر یک جواب در حین فرایند جستجو یک جواب را تغییر می‌دهند، درحالی‌که در الگوریتم‌های مبتنی بر جمعیت در حین جستجو، یک جمعیت از جواب‌ها در نظر گرفته می‌شوند (تهرانی و همکاران، ۱۳۹۷).

فرایند طراحی و پیاده‌سازی الگوریتم‌های فراابتکاری دارای سه مرحله متوالی است که هر کدام از آن‌ها دارای گام‌های مختلفی هستند. در هر گام فعالیت‌هایی باید انجام شود تا آن گام کامل شود. مرحله ۱ آماده‌سازی است که در آن باید شناخت دقیقی از مسئله‌ای که می‌خواهیم حل کنیم بدست آوریم، و اهداف طراحی الگوریتم فراابتکاری برای آن باید با توجه به روش‌های حل موجود برای این مسئله به‌طور واضح و شفاف مشخص شود. مرحله ۲ بعدی، ساخت نام دارد. مهم‌ترین اهداف این مرحله انتخاب استراتژی حل، تعریف معیارهای اندازه‌گیری عملکرد، و طراحی الگوریتم برای استراتژی حل انتخابی می‌باشد. آخرین مرحله پیاده‌سازی است که در آن پیاده‌سازی الگوریتم طراحی شده در مرحله ۲ قبل، شامل تنظیم پارامترها، تحلیل عملکرد، و در نهایت تدوین و تهیه گزارش نتایج باید انجام شود (ایمانی کله سر و هوشیار و همکاران (۱۴۰۲)).

روش پیشنهادی

گام‌های الگوریتم بهینه‌سازی فاخته



شکل ۳. مراحل الگوریتم فاخته

گام ۱: مکان‌های سکونت فعلی فاخته‌ها را به‌صورت تصادفی مشخص می‌کنیم.
 گام ۲: تعدادی تخم به هر فاخته اختصاص می‌دهیم.
 گام ۳: شعاع تخم‌گذاری هر فاخته را تعیین می‌نماییم.
 گام ۴: فاخته‌ها در لانه‌های میزبانانی که در شعاع تخم‌گذاری آن‌ها قرار دارند، تخم‌گذاری می‌کنند.
 گام ۵: تخم‌هایی که توسط پرندگان میزبان شناسایی می‌شوند از بین می‌روند.
 گام ۶: تخم فاخته‌هایی که شناسایی نشده‌اند پرورش می‌یابند.
 گام ۷: محل سکونت فاخته‌های جدید را ارزیابی می‌کنیم.
 گام ۸: ماکزیمم تعداد فاخته‌هایی که در هر مکان امکان زندگی دارند را مشخص می‌نماییم و آن‌هایی را که در مکان‌های نامناسب هستند از بین می‌بریم.
 گام ۹: فاخته‌ها را با استفاده از روش K-means خوشه‌بندی و بهترین گروه فاخته را به‌عنوان مکان سکونت هدف مشخص می‌کنیم.

گام ۱۰: جمعیت جدید فاخته‌ها به سمت مکان هدف حرکت می‌کنند.
 گام ۱۱: اگر شرط توقف برقرار گردیده توقف، در غیر این صورت به گام ۲ می‌رویم.
 با تکرار مراحل فوق به تدریج لانه‌ها به سمت نقاط بهینه حرکت می‌کنند و در پایان اجرای الگوریتم تمامی N لانه در اطراف نقاط بهینه جمع می‌شوند. بنابراین اگر تعداد لانه‌ها به اندازه‌ی کافی از تعداد نقاط بهینه بیشتر باشد الگوریتم جستجوی فاخته همگرایی به سمت جواب بهینه را تضمین می‌کند. (Rajabioun, 2011)

سیر تصادفی

سیر تصادفی یک فرآیند تصادفی، شامل یک سری از گام‌های تصادفی متوالی است. به زبان ریاضی، اگر X_N مجموع گام‌های تصادفی متوالی X_i باشد آنگاه S_N یک سیر تصادفی را تشکیل می‌دهد:

$$S_N = \sum_{i=1}^N X_i = X_1 + \dots + X_N \quad (1)$$

که در آن یک گام تصادفی است که از یک توزیع تصادفی تبعیت می‌کند. رابطه‌ی فوق را می‌توان به‌صورت بازگشتی نیز نوشت:

$$S_N = \sum_{i=1}^{N-1} X_i = X_{N-1} + X_N \quad (2)$$

از رابطه‌ی فوق مشاهده می‌شود که S_N و انتقال X_N از وضعیت جاری به وضعیت بعدی وابسته است. این ویژگی خاصیت اصلی زنجیره‌ی مارکوف ۱۲ است. سیر تصادفی کاربردهای زیادی در فیزیک، اقتصاد، آمار، علوم کامپیوتر، محیط‌زیست و مهندسی دارد (Branch, 2015).

توزیع لوی

در طبیعت، حیوانات برای جستجوی غذا معمولاً از روش تصادفی یا شبه تصادفی استفاده می‌کنند. به‌طور کلی مسیر جستجوی غذای یک حیوان، یک مسیر پیاده‌روی تصادفی مؤثر است چراکه حرکت بعدی بر پایه محل یا وضعیت فعلی و احتمال حرکت به مکان بعدی استوار است. که این مسیر انتخاب‌شده می‌تواند بورت ضمنی وابسته به یک احتمال باشد که توسط روابط ریاضی قابل مدل‌سازی باشد. مطالعات اخیر بر روی زنبورها و مگس‌ها نشان می‌دهد که محیط پیرامونی خود را با مسیرهای مستقیم و چرخش‌های ۹۰ درجه ناگهانی کاوش می‌کنند. امری که منجر به ارائه سبکی از الگوی پرواز لوی متناوب در مقیاس منعطف شد. اگرچه مبحث نور هم می‌تواند با پرواز لوی مرتبط باشد. پس از آن موجب شد این الگو برای بهینه‌سازی و جستجوی بهینه مورد استفاده قرار بگیرد و به نتایج امیدوارکننده‌ای دست‌یافته شود (Csanád, M., & Kincses, D, 2024).

¹² Markof chain

توزیع لوی یک توزیع پیوسته برای متغیرهای تصادفی غیر منفی می‌باشد. این توزیع که به نام ریاضیدان فرانسوی، پاول لوی ۱۳ نام‌گذاری شده است، یکی از معدود توزیع‌های پایدار است. این توزیع در مهندسی مالی در مدل کردن تغییرات قیمت بازارهای مالی بسیار کاربرد دارد. تابع چگالی احتمال توزیع لوی به صورت تحلیلی قابل بیان می‌باشد. تابع چگالی توزیع لوی به صورت زیر است:

$$L(s, \gamma, \mu) = \sqrt{\frac{\gamma}{2\pi}} \frac{1}{(s-\mu)^{\frac{3}{2}}} \exp\left(-\frac{\gamma}{2(s-\mu)}\right) \quad 0 < \mu < s < \infty \quad (3)$$

که در آن μ مینیمم گام‌ها و γ پارامتر اندازه است. اگر $s \rightarrow \infty$ آنگاه:

$$L(s, \gamma, \mu) = \sqrt{\frac{\gamma}{2\pi}} \frac{1}{(s-\mu)^{\frac{3}{2}}} \quad s \rightarrow \infty \quad (4)$$

می‌توان برای تولید گام تصادفی s از الگوریتم مانتگنا ۱۴ استفاده کرد. با این الگوریتم طول گام s به صورت زیر خواهد بود:

$$s = \frac{u}{|v|^{1/\beta}} \quad (5)$$

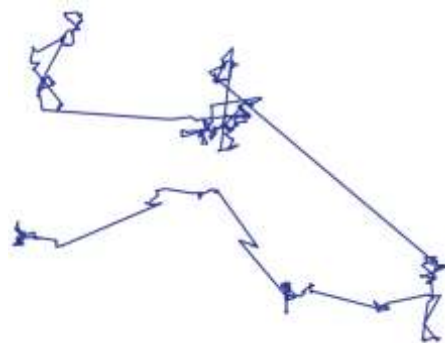
که u و v متغیرهای با توزیع نرمال است.

$$u \sim N(0, \sigma_u^2) \quad , \quad v \sim N(0, \sigma_v^2) \quad (6)$$

که در آن:

$$\sigma = \left\{ \frac{\Gamma(1+\beta) \sin(\pi\beta/2)}{\Gamma(1+\beta)/2 | \beta 2^{(\beta-1)/2} } \right\}^{1/\beta} \quad (7)$$

می‌باشد. Γ در رابطه فوق، تابع گاما می‌باشد. توزیعی که از رابطه‌ی (۳-۵) برای s به دست می‌آید یک توزیع لوی خواهد بود برای $|s| \geq |s_0|$ که s_0 کوچک‌ترین گام است. (Govindan, 2011) در تصویر زیر نمونه‌ای از پرواز لوی در ۱۰۰ گام نمایش داده شده است.



شکل ۴. نمونه‌ای از پرواز لوی در ۱۰۰ گام

ویژگی‌های جستجوی فاخته

سادگی، فهم آسان، سهولت پیاده‌سازی، قابلیت استفاده در مسائل چند-معیاره، جستجوی جامع فضای جستجو، برداشتن مناسب‌ترین گام‌ها به سوی هدف به کمک پرواز لوی و عملکرد بسیار خوب در بهینه‌سازی توابع نامعین ۱۵ از جمله مزایای جستجوی فاخته است. علاوه بر این، روش جستجوی فاخته مکانیزمی ساده و درعین حال کارا برای نخبه‌گزینی و تولید نسل

¹³ Pule Pierre Levy

¹⁴ Mantegna's Algorithm

¹⁵ Stochastic

آینده دارد که باعث می‌شود سرعت اجرا و همگرایی الگوریتم بالا رود. این ویژگی‌ها باعث شده است جستجوی فاخته در بسیاری از کاربردها از سایر الگوریتم‌های فرا تکاملی عملکرد بهتری از خود نشان دهد (Shadkam, 2015).

مسئله زمان‌بندی

به صورت کلی می‌توان یک مسئله زمان‌بندی در فضای ابر را در قالب چهار متغیر $M = (S, V, F, \theta)$ تعریف کرد که برای n کار و m منبع مجازی به کار برده می‌شود. S مجموعه‌ای از n کار است. V مجموعه‌ای از m منبع مجازی است. F تابع هدف زمان‌بندی منابع است. و θ الگوریتم بهینه زمان‌بندی منابع است. در این پژوهش ما از فرض‌های زیر برای مسئله زمان‌بندی استفاده نموده‌ایم.

۱. عملکرد هر ماشین مجازی باید متناسب نیازمندی هر کار باشد.
 ۲. هر کار تنها می‌تواند بر روی یک ماشین مجازی اجرا شود.
 ۳. تأثیر زمان انتقال کارها بر روی مدل بهینه‌سازی در نظر گرفته نشده است.
- در این پژوهش فرض شده است تمامی کارها توسط یک کاربر ایجاد می‌شود. و مدل زمان‌بندی ما شامل مشخصه‌های زیر است.

۱. کارهای $S = \{s_1, s_2, L, s_n\}$ به وسیله کاربر ارسال می‌شوند و هر کار از کار دیگر مخفی و مستقل است. SL_i طول کار i است که همیشه توسط کاربر تعیین می‌شود.
۲. منابع مجازی $V = \{v_1, v_2, L, v_m\}$ است، که هر ماشین مجازی شامل مشخصه‌های از قبیل توان مصرفی، حافظه و پهنای باند مشخص است. در این پژوهش توان مصرفی ماشین‌های مجازی $V = \{vc_1, vc_2, L, vc_i, L, vc_m\}$ در نظر گرفته شده است. که vc_i توان مصرفی ماشین i می‌باشد.
۳. برای زمان اجرایی کارها از ماتریس زمانی

$$Time = \{time_{ij}\} = (i = 1, 2, L, n; j = 1, 2, L, m) \quad (8)$$

استفاده می‌شود که $time_{ij}$ بیانگر زمان اجرای کار i بر روی ماشین j است. زمان اجرای $time_{ij}$ توسط فرمول زیر محاسبه می‌شود.

$$time_{ij} = \frac{SL_i}{VC_j} \quad (9)$$

ماتریس توزیع به صورت زیر است.

$$|X| = \begin{vmatrix} x_{11} & x_{12} & L & x_{1m} \\ x_{21} & x_{22} & L & x_{2m} \\ M & M & M & M \\ x_{n1} & x_{n2} & L & x_{nm} \end{vmatrix} \rightarrow x_{ij} = \begin{cases} 1, \\ 0, \end{cases} \quad (10)$$

از ماتریس برای نمایش این موضوع استفاده شده تا درک درست‌تری از مسئله به دست آید برای نمونه بر اساس ماتریس مشخص می‌شود کار i به ماشین j توزیع داده شده یا خیر.

برای بهبود QoS در مسئله زمان‌بندی ما سه عامل رو در نظر گرفته‌ایم و سعی نموده‌ایم بر اساس این سه عامل که عبارت‌اند از زمان تکمیل، هزینه و انرژی مصرفی تابع هدف ساخته شود و این مقدار به مقدار بهینه متمایل شود.

عامل زمان: منظور از زمان، زمان تکمیل شدن یک کار است. در محیط ابر RET_j بیانگر زمان اجرای منابع مجازی j است که توسط تابع $RET_j = \sum_{i=1}^n x_{ij} * time_{ij}$ محاسبه می‌گردد. با در نظر گرفتن اینکه ماشین‌های مجازی به صورت موازی کار می‌کنند زمان کامل شدن تمامی کارها بر اساس فرمول زیر به دست می‌آید.

$$TIME = \max_{j=1}^m RET_j \quad (11)$$

C به معنای هزینه در زمان‌بندی کارها است. با در نظر گرفتن این نکته که تمامی کارها در تقابل باهم مستقل می‌باشند. ماتریس $RC\{j\} = (j = 1, 2, L, m)$ برای توصیف هزینه منبع مجازی j در هر واحد زمانی است. هزینه تمامی اجراهای کارها در ابر می‌تواند توسط فرمول محاسبه شود.

$$C = \sum_{j=1}^m RET_j * RC(j) \quad (12)$$

برای توصیف مصرف انرژی از ماتریس $E = \{e_{ij}\} = \{i = 1, 2, L, n; j = 1, 2, L, m\}$ استفاده می‌شود. که e_{ij} بیانگر میزان انرژی مصرف شده کار i بر روی ماشین j است. انرژی تمام شده کل کارها بر اساس فرمول زیر محاسبه می‌شود.

$$EC = \sum_{i=1}^n \sum_{j=1}^m x_{ij} * e_{ij} \quad (13)$$

با در نظر گرفتن عامل‌های که در بالا ذکر شد. نیاز به یک شمای قابل درک برای مسئله زمانمندی می‌باشد. هدف اصلی این پژوهش کاهش زمان اجراء هزینه و انرژی مصرفی است. با در نظر گرفتن راه حل مسئله به صورت چند هدف تابع راه حل پیشنهادی به صورت زیر فرموله می‌شود.

$$\text{Minimize } F(x) = \omega_1 * EC + \omega_2 * C \quad (14)$$

MiniMize Total Time

ω_1 و ω_2 به ترتیب ضرایب وزنی زمان و انرژی و هزینه می‌باشند. وزن این ضرایب بزرگتر یا مساوی صفر و مجموع این اوزان برابر یک است. برای حل مسئله ما در این پژوهش از الگوریتم چندهدفه جستجوی فاخته استفاده کردیم که فرآیند حل مسئله و روش پیشنهادی در شکل زیر نمایش داده شده است.

بهینه‌سازی چندهدفه

در بهینه‌سازی چندهدفه ما با چندین تابع هدف مواجه هستیم که می‌بایست بهینه شوند. چندین راه وجود دارد که با یکدیگر قابل مقایسه نیستند، که معمولاً به عنوان بهینه پارتو شناخته می‌شوند. یک مسئله چند هدف می‌تواند توسط n متغیر با m هدف بدون از دست دادن عمومیت مسئله فرمول بندی شود.

$$\min y = f(\bar{x}) = \min(f_1(\bar{x}), f_2(\bar{x}), \dots, f_m(\bar{x})) \quad (15)$$

که

$$\bar{x} = (x_1, x_2, \dots, x_n) \text{ و } y = (y_1, y_2, \dots, y_m) \quad (15)$$

در اغلب موارد توابع هدف با یکدیگر در تضادند. بنابراین کاهش یکی از توابع بدون افزایش حداقل یکی از توابع امکان پذیر نیست. چنین امری بیانگر مفهوم بهینه پارتو است. بنابراین با توجه به طبیعت چندهدفه بودن مسئله زمانمندی کارها، در این پژوهش از الگوریتم چندهدفه جستجو فاخته به عنوان راه‌حلی برای این مسئله استفاده شده است.

الگوریتم فاخته چندهدفه

الگوریتم فاخته اصلی که در تک هدفه بودن از سه قانون کلی پیروی می‌کند (Nazemi, S., & Khorsand, R). (۲۰۲۴).

- هر فاخته در هر لحظه تنها یک تخم می‌گذارد و تنها یک خانه را برای تخم‌گذاری به صورت تصادفی انتخاب می‌کند.
- بهترین لانه همراه با کیفیت ترین تخم‌ها (جواب مسئله) برای تولید نسل بعدی انتقال داده می‌شود.
- تعداد لانه‌های میزبان در دسترس ثابت است. و میزبان می‌تواند اطراف تخم‌ها را با احتمال $p_a \in (0, 1)$ کاوش کند. در این حالت میزبان می‌تواند تخم را به بیرون پرتاب کند. یا لانه برای ساختن لانه جدید کاملاً رها سازد.
- برای مسئله بهینه‌سازی چندهدفه با k هدف قانون اول و سوم نیاز به اصلاح دارند.
- هر فاخته می‌تواند در هر بازه زمانی k تخم بگذارد و آن‌ها را در درون یک لانه تصادفی انتخاب شده بگذارد. در اینجا k متناظر با تعداد تابع هدف‌های مسئله است.
- هر لانه با احتمال p_a ممکن است رها شود و لانه جدیدی برای k تخم ایجاد شود. با در نظر گرفتن شباهت‌ها و تفاوت‌ها می‌توان در تنوع تولید نسل‌ها از این پتانسیل استفاده نمود.

برای ساده نمودن می توان آخرین قانون را بدین صورت تعریف نمود که تعداد لانه‌ها به صورت تقریب با کسری از p_a با لانه‌های جدید جایگزین می‌شوند (جواب تصادفی جدید در محل جدید). برای به حداکثر رساندن هدف‌ها می توان برآزش و کیفیت را متناسب با هر تابع هدف در نظر گرفت و به دنبال آن جواب غیر غالب^{۱۶} را جستجو نمود.

برای اینکه بتوان تعریف ریاضی به مسئله داد، می توان قانون اولیه را به پروسه تصادفی تبدیل نمود. که جواب‌های جدید می‌توانند توسط پیاده‌روی تصادفی یا پرواز لوی تولید شوند. در یک زمان جایگشت تصادفی محلی می‌تواند بر روی جواب‌ها به‌مثابه یک عمل متقاطع انجام شود. به ازای هر لانه می‌تواند k جواب بر اساس رابطه ایجاد شود. اما قانون دوم در واقع این قانون نخبه‌گرایی است و هدف آن استفاده از بهترین جواب‌ها در تولید نسل بعدی است این خاصیت این امکان را به ما می‌دهد که اطمینان یابیم که الگوریتم به درستی در مسیر صحیح در حال همگرا شدن است. قانون سوم را هم می‌توان قانون جهش نام نهاد که در آن جواب‌های نامناسب توسط احتمالی از جواب‌های تولیدشده جدید حذف و جایگزین می‌شوند. این بدین معنی است جهش یک اپراتور برداری است که با پرواز لوی و دیفرانسیل کیفیت جواب‌ها ترکیب شده است. این ویژگی منحصر به فرد از ترکیب‌ها کارها می‌تواند کارایی الگوریتم را تضمین کند (Nebagiri, M. H., & Hnumanthappa, L. P). (۲۰۲۴).

بر اساس این سه قانون می‌توان مراحل پایه الگوریتم چندهدفه فاخته را در قالب شبه کد زیر توصیف نمود. وقتی که نسل جدید $x^{(t+1)}$ ساخته می‌شود در اصطلاح می‌گوییم که فاخته یک پرواز لوی انجام داده

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus \text{Lévy}(\beta) \quad (16)$$

که $\alpha > 0$ اندازه گام‌هاست و باید متناسب با خصوصیات مسئله تعیین شود. در اغلب موارد می‌توان از $\alpha = O(1)$ استفاده نمود. به عبارت دیگر برای جبران تفاوت کیفیت جواب‌های مسئله می‌توان از رابطه زیر استفاده کرد.

$$\alpha = \alpha_0 (x_j^{(t)} - x_i^{(t)}) \quad (17)$$

که α_0 یک ثابت می‌باشد و عبارات درون پرانتز تفاوت دو جواب تصادفی است. و این در واقع دست‌کاری بدین منظور است که تخم‌مرغ‌های مشابه با احتمال کمتری در تولید نسل شرکت داده شوند بلکه نسلی ایجاد شود که از تفاوت بیشتری باهم برخوردار باشند. علامت \oplus به معنای ضرب و رودی است. پرواز لوی اساساً یک پیاده‌روی تصادفی استوار بر توزیع لوی برای گام‌های بزرگ است.

$$\text{Lévy} \sim u = t^{-1-\beta}, (0 < \beta \leq 2) \quad (18)$$

که دارای واریانس بی‌نهایت با میانگین بی‌نهایت است. در اینجا گام‌ها/ پرش‌های یک فاخته اساساً ناشی از یک فرآیند تصادفی است که تابع قانون توان توزیع طول گام با یک دنباله سنگین است. علاوه بر این کسری از p_a از لانه‌های نامناسب تخلیه و با ترکیبی از پیاده‌روی تصادفی در ساختن لانه‌های جدید در محل‌های جدید مورد استفاده قرار بگیرند. البته بدیهی است نسلی با اندازه S که توسط پرواز لوی ایجاد شده‌اند بی‌اهمیت و غیر مؤثر نیست. رابطه مذکور به صورت زیر توصیف شود.

$$s = \alpha_0 (x_j^{(t)} - x_i^{(t)}) \oplus \text{Lévy}(\beta) \sim 0.01 \frac{u}{|v|^{\frac{1}{\beta}}} (x_j^{(t)} - x_i^{(t)}) \quad (19)$$

که u و v توسط توزیع نرمال به دست می‌آید.

$$u \sim N(0, \sigma_u^2) \quad v \sim N(0, \sigma_v^2) \quad (20)$$

Initialize objective functions $f_1(x), \dots, f_k(x)$ $x = (x_1, \dots, x_d)^T$
 Generate an initial population of n host nests x_i and each with K eggs
While ($t < \text{MaxGeneration}$) or (stop criterion)
 Get a cuckoo (say i) randomly by Lévy flights
 Evaluate and check if it is Pareto optimal
 Choose a nest among n (say j) randomly

¹⁶ non-dominated

Evaluate K solution for nest j
If new solutions of nest j dominate those of nest I,
 Replace nest I by new solution set of nest j
 end
 Abandon a fraction (p_a) of worse nests
 Keep the best solutions (or nest with non-dominated sets)
 Sort and find the current Pareto optimal solutions
 End
 Post Process results and visualization

که

$$\sigma_u = \left\{ \frac{\Gamma(1 + \beta) \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1 + \beta}{2}) \beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \sigma_v = 1 \quad (21)$$

نکته آخر که می‌بایست بدان اشاره کرد ویژگی‌های منحصر به فرد الگوریتم چندهدفه جستجو فاخته است که آن را با سایر الگوریتم‌ها متفاوت می‌سازد و این ویژگی‌ها عبارت‌اند از اکتشاف توسط پرواز لوی، عمل جهش توسط پرواز لوی و دیفرانسیل برداری جواب، عمل تقاطع توسط جایگشت‌های تصادفی و نخبه‌گرایی. نتیجه ترکیب این مؤلفه‌ها موجب شده تا الگوریتم جستجوی فاخته در عمل الگوریتم قوی و کارآمدی ظاهر شود.

اصل پارتو

می‌گوییم بردار جواب، $u = (u_1, \dots, u_n)^T \in \mathcal{F}$ بر بردار $v = (v_1, \dots, v_n)^T$ غلبه می‌کند اگر و فقط اگر $u_i \leq v_i$ به ازای $\forall i \in (1, \dots, n)$ و $\exists i \in [1, \dots, n] : u_i < v_i$ به عبارت دیگر هیچ مؤلفه از u بزرگ‌تر از مؤلفه‌های v نباشد و حداقل دارای یک مؤلفه کوچک‌تر باشد. می‌توان رابطه غلبه را به صورت دیگری نیز نوشت:

$$u \preceq v \Leftrightarrow u < v \vee u = v \quad (22)$$

قسمت پارتو در یک مسئله چندهدفه می‌تواند توسط مجموعه‌ای از راه‌حل‌های بدون غلبه باشند.

$$PF^* = \{s \in S | \exists s' \in S : s' < s\} \quad (23)$$

و یا مجموعه بهینه پارتو در فضای جستجوی مسئله به صورت زیر تعریف می‌شود.

$$PF^* = \{x \in F | \exists x' \in \mathcal{F} : f(x') < f(x)\} \quad (24)$$

که $f = (f_1, \dots, f_k)^T$ است. برای رسیدن به پارتو بهینه باید از تکنیک‌های مختلفی جهت ایجاد تنوع در نسل‌ها ایجاد نمود که پرواز لوی نمونه مناسبی از این تکنیک‌ها است.

نگاشت الگوریتم به عنوان راه‌حل پیشنهادی

مسئله اصلی در واقع چگونگی توزیع مجموعه‌ای از کارها دریافت شده توسط برآکر به لیستی از ماشین‌های مجازی است. بنابراین اگر زمان اجرا کاهش داده شود خودبه‌خود موجب کمتر شدن زمان بهینه خواهد شد که بهبود مناسبی تلقی خواهد شد. در الگوریتم‌های تک هدفه نیز این مسئله وجود دارد. در زمان‌بندی کارها مبتنی بر اولویت کارهای با اولویت بالا همیشه از شانس بیشتری برای اجرا برخوردارند. بنابراین کارهای با اولویت کمتر زمان طولانی‌تری منتظر اجرا خواهند بود (Mishra, A., & Goel, L). (۲۰۲۴).

با توجه به شرح الگوریتم فاخته و نوع چندهدفه آن لازم است تا الگوریتم بر اساس نیاز مسئله تعریف و پیاده‌سازی شود. اصولاً برای حل مسائل زمان‌بندی کارها در فضای ابر دو روش کلی وجود دارد و آن نیز به ترتیب ماشین‌های مجازی و کارها برمی‌گردد بدین صورت می‌توان ترتیب کارها را ثابت فرض کرد و ماشین‌های مجازی برای انجام کارها را به‌عنوان برداری از شناسه ماشین‌ها و کارهای متناظر با آن در نظر گرفت. برای نمونه فضای مسئله را بدین صورت در نظر می‌گیریم 10 کار با 5 ماشین مجازی موجود است بنابراین بردار جواب‌ها به صورت زیر است.

جدول ۱. توزیع تصادفی کارها به ماشین‌های مجازی

Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10
v_0	v_3	v_1	v_2	v_1	v_3	v_2	v_0	v_1	v_4
v_1	v_0	v_4	v_1	v_3	v_0	v_2	v_2	v_1	v_0
v_0	v_1	v_3	v_3	v_0	v_2	v_3	v_0	v_0	v_1
v_2	v_4	v_0	v_1	v_2	v_1	v_0	v_3	v_3	v_0

هر بردار آرایه‌ای به طول تعداد کارها است و مقادیر هر خانه از این بردار عددی در بازه تعداد ماشین‌های مجازی می‌باشد بدین معنی اگر تعداد ماشین مجازی داریم مقدار هر خانه با در نظر گرفتن شروع آدرس ماشین‌ها از صفر عددی در بازه صفر تا چهار خواهد بود. هر جواب مسئله دارای دو تابع هدف خواهد بود که در قسمت زمان‌بندی بدان اشاره شد. که نهایتاً بهترین جواب که توالی از ماشین‌های مجازی اختصاص‌یافته است به‌عنوان جواب بهینه انتخاب می‌شود.

روش پیشنهادی ما در اصل بر پایه الگوریتم چندهدفه فاخته است. در اینجا به‌جای اینکه ما با یک هدف روبرو باشیم از چند هدف استفاده کرده‌ایم. بدین صورت که ابتدا تعدادی جواب تصادفی به‌عنوان جواب‌های اولیه انتخاب می‌شوند و بر اساس الگوریتم فاخته چندهدفه فرآیند برازندگی آن‌ها تعیین می‌گردد و بعد از هر تکرار مقدار بهینه پارتو به‌عنوان نتیجه بهینه‌تر به‌عنوان خروجی انتخاب می‌شود. روش پیشنهادی به‌صورت زیر است.

۱. ساخت مجموعه‌ای از کارها
۲. ایجاد لیستی از ماشین‌های مجازی
۳. توزیع کارها به ماشین‌های مجازی به‌صورت تصادفی
۴. ارزیابی اصل پارتو و مرتب‌سازی غیر مغلوب جواب‌ها بر اساس توابع هدف.
۵. افزودن جواب بهینه به مجموعه جواب‌ها

همان‌گونه که قبلاً نیز اشاره شده از اصل مرتب‌سازی غیر مغلوب برای حل مسائل چندهدفه استفاده می‌شود و روش کارآمدی نیز می‌باشد. با توجه به استفاده از این اصل در الگوریتم پیشنهادی نحوه استفاده از این روش به شکل شبکه کد زیر است.

-
- 1: Create List of Task
 - 2: Create List of VMs
 - 3: Non-dominate Sorting (List of VMs index that assign to tasks)
 - 4: Create empty Non-Dominating List
- $i \leftarrow 0$

```

in Non-Dominating List initially put  $Solution_i$ 
For  $i \leftarrow 1$  size of  $Solution$  list do
For  $j \leftarrow 0$  size of Non-Dominating List do
  If  $Solution_j$  dominate  $Solution_i$  then
    put  $Solution_j$  in Non-Dominating List
  else
    if
       $Solution_i$  dominate  $Solution_j$  then
        put  $Solution_i$  in Non-Dominating List
      else
        put  $Solution_i$  and  $Solution_j$  in Non-Dominating List
ENDFOR
ENDFOR
5: Sort List According Non-Dominating List

```

تابع هزینه :

برای تابع هدف نیز در هر اجرای الگوریتم لیستی از شناسه ماشین‌های مجازی که معرف توزیع به کار مشخصی است به تابع هدف ارسال می‌شود و در بدنه تابع هدف ماشین‌ها توزیع داده می‌شوند و بعد از مرحله شبیه‌سازی مقادیر توابع هدف محاسبه و به هر جواب یا فاصله توزیع داده می‌شود. در اینجا کد تابع هزینه استفاده شده در الگوریتم پیشنهادی آمده است :

```

Size ← Number of Tasks
Foreach task in tasks
  Assign  $vm(i)$  to task (i)
Simulate()
for (int  $i = 0$ ;  $i < size$ ;  $i++$ )
  calculate  $c = \omega_1 * EC + \omega_2 * C$  for task(i)
  calculate  $fin = FinishTime(task(i))$ 
   $F += c$ 
   $Total\ Time += fin$ 
Return  $F, Total\ Time$ 

```

شرط توقف الگوریتم

روش‌های فرا تکاملی در دسته روش‌های تکراری قرار می‌گیرند. زیرا در این روش‌ها با انتخاب یک جمعیت تصادفی از جواب‌های اولیه تلاش می‌شود تا در تکرارهای متعدد حرکت به سمت یافتن جواب‌های بهتر ادامه پیدا کند. بنابراین، یکی از بخش‌های اصلی این الگوریتم‌ها شرط توقف و خاتمه تکرار است. یکی از شرط‌های توقف در این روش‌ها، بررسی رسیدن به بهترین پاسخ است که برای مسائلی مناسب است که مقدار برازندگی بهترین پاسخ در آن‌ها مشخص است. در خصوص مسئله انتخاب زیرمجموعه بهینه از گره‌های گراف با هزینه محدود این شرطی امکان‌پذیر نیست.

حالت دوم برای توقف الگوریتم را می‌توان با راکد شدن جمعیت مشخص کرد. در این حالت، جمعیت در نسل‌های متوالی تغییر نمی‌کند و می‌توان با شمارش تعداد نسل‌هایی که در آن‌ها بهترین پاسخ تغییر نکرده است، توقف الگوریتم را مشخص کرد.

اما، حالت سومی که در این پژوهش از آن و حالت دوم برای شرط توقف استفاده شده است، محدود کردن تعداد نسل است. در این حالت، به‌عنوان پارامتر ورودی مسئله، تعداد تکرار الگوریتم که معمولاً با MaxIt نشان داده می‌شود، مشخص می‌شود و الگوریتم به این تعداد تکرار می‌شود.

نتایج و آزمایش

برای ارزیابی نتایج و کارایی الگوریتم پیشنهادی ما از سیستم و نرم‌افزارهای مشخصی استفاده نمودیم. برای شبیه‌سازی از محیط شبیه‌ساز cloudsim 3 که در محیط NetBeans IDE 7.2.1 پیکربندی شده بود استفاده شد و از سیستمی با 8 Gb Ram , cpu : core i7 2.7 GHz , windows 10 64bit استفاده شد. در شبیه‌سازی سناریوها ما از سناریوهای مختلفی برای ارزیابی روش پیشنهادی در مقایسه با سایر الگوریتم‌ها بهره بردیم.

کلودسیم

کلودسیم چارچوبی برای مدل‌سازی و پیاده‌سازی زیرساخت و سرویس‌های ابر است که در آزمایشگاه ابر دانشگاه ملبورن ارائه شده است. کلودسیم مجموعه‌ای از کتابخانه‌ها و کلاس‌های پیاده‌سازی شده به زبان جاوا است که امکان مدل‌سازی رفتاری و سیستمی اجزای ابری مثل مراکز داده، ماشین‌های مجازی و سیاست‌های تأمین منابع را فراهم می‌کند (Cavdar, et al (۲۰۲۴)).

کتابخانه کلودسیم کلاس‌های اصلی برای تعریف مراکز داده، ماشین‌های مجازی، کاربردها، کاربرها، منابع محاسباتی و سیاست‌هایی برای مدیریت بخش‌های متنوع یک سیستم (برای مثال زمان‌بندی) را فراهم می‌کند. این اجزا می‌توانند توسط کاربران برای ارزیابی استراتژی‌های جدید در بکارگیری ابرها (سیاست‌ها، الگوریتم‌ها زمان‌بندی، سیاست‌های توزیع بار و نگاشت، و غیره) کنار هم‌دیگر قرار بگیرند.

ارزیابی

برای بررسی و ارزیابی روش پیشنهادی ما چند الگوریتم را به‌عنوان الگوریتم‌های پایه در نظر گرفتیم که این الگوریتم‌ها عبارت بودند از NSGAI, MOPSO و مشخصات کارها و ماشین مجازی را در مقیاس‌های مختلفی مورد آزمایش قرار دادیم تا کارایی روش پیشنهادی به اثبات برسد. مشخصات و خصوصیات داده‌ها بر اساس محیط شبیه‌سازی کلودسیم ذکر گردیده است.

دیتاسنتر:

درواقع هر دیتاسنتر شامل چندین میزبان است که هر کدام از این میزبان‌ها می‌تواند شامل چندین ماشین مجازی باشد. که این میزبان‌ها متناظر با یک سرور فیزیکی در فضای ابر است. و هر کدام از این میزبان‌ها به‌نوبه خود می‌توانند شامل مشخصات سخت‌افزاری و نرم‌افزاری مخصوص به خود باشند (Chauhan, et al (۲۰۲۴)).

ماشین مجازی:

هر ماشین مجازی می‌تواند انواع مختلفی از معماری مانند (Intel, Mac, IBM, ...) داشته باشد. هر پردازنده درون ماشین مجازی نیز به‌نوبه خود می‌تواند ویژگی‌های خاص خود را داشته باشد. مانند قدرت پردازنده، هزینه انجام کاربر حسب ثانیه، تعداد هسته، و پهنای باند.

کار (Cloudlet):

کلادلت مدلی برای سرویس‌های کاربردی مبتنی برابری مانند تحویل محتوا، شبکه‌های اجتماعی و گردش کار تجاری است. کلادلت درخواستی است که از سوی کاربر برای اجرا به واسطه ابر ارسال می‌شود. درواقع کلادلت همان کارها است که توسط ماشین مجازی می‌بایست اجرا شوند.

برای ارزیابی مدل پیشنهادی ما تعدادی آزمایش برای بررسی الگوریتم انجام دادیم در مرحله اول ما توابع سیستم را به وسیله مدیریت کارهای درون ماشین‌های مجازی و نگاشت آن‌ها به ماشین‌های فیزیکی درون دیتاسنتر آزمایش نمودیم. در مرحله دوم آزمایش ما اقدام به مقایسه زمان اجرا و مجموع وزن‌ها در مدل پیشنهادی با سایر الگوریتم‌ها نمودیم. برای جزئیات بیشتر ما از چهار نوع ماشین مجازی برای بررسی آزمایش‌ها استفاده نمودیم. که این ماشین‌های مجازی بر اساس نوع پردازنده، حافظه و پهنای باند است که در ذیل مشخصات آن ذکر شده است.

جدول ۲. مشخصات ماشین‌های مجازی بر اساس کانفیگ تعریف شده

No of Data Centres	۲
No of Cloudlets	{20,200}
No of Hosts in Data Centre	40
No of VMs	{5,10}

جدول ۳. مشخصات پیکربندی ماشین‌های مجازی

CPU	{ ۵۰۰, ۱۰۰۰, ۲۰۰۰, ۲۵۰۰ }
RAM Size	{ ۶۱۳, ۱۷۴۰, ۱۷۴۰, ۸۷۰ }
Bandwidth	Mbit/s ۱۰۰ // ۱۰۰۰۰
Storage Size	۱۰۰۰۰
VM Cost	{ ۰,۰۲۰, ۰,۰۴۷, ۰,۱۴۸, ۰,۲۰۰ }
Number of Cores in each VM	۱

جدول ۴. مشخصات کارها

length	(500,3000)
long fileSize	(300,2500)
outputSize	(300,500)
pesNumber	1

بعد از اجرای الگوریتم پیشنهادی و دو الگوریتم مورد مقایسه NSGAI و MOPSO در محیط برنامه نویسی نت بینز و ایکلیپس و شبیه سازی در محیط کلودسیم، ما معیارهایی را برای کارایی سیستم اندازه‌گیری نمودیم که این معیارها عبارت بودند از:

متوسط زمان انتظار:

این معیار در واقع نرخ مجموع زمان انتظار تمام کارهای است که منتظر اجرا هستند. متوسط زمان انتظار در واقع معیاری است که توسط اختلاف بین ارسال کار به سیستم و زمان شروع کار برای اجرا محاسبه می‌شود. این اختلاف توسط فرمول زیر محاسبه می‌شود.

$$WT = \sum_{i=0}^n \text{Start time}(t) - \text{submission time}(t) / m$$

در این فرمول m تعداد کارهای در حال اجرا درون سیستم است.

متوسط زمان اجرا:

این معیار نرخ مجموع زمان اجرای تمامی کارها به نسبت تعداد کل کارهای اجرا شده است. متوسط زمان اجرای یک کار توسط اختلاف بین زمان شروع و پایان اجرای هر کار محاسبه می‌شود. فرمول این محاسبه در فرمول زیر نشان داده شده است.

$$ET = \sum_{i=0}^m \text{end execution time}(t) - \text{start execution time}(t) \quad \blacklozenge$$

در این فرمول m تعداد کارهای در حال اجرا درون سیستم است.

بازدهی:

معیار بازدهی مربوط به کارایی کلی الگوریتم است. در واقع این معیار نشان‌دهنده تعداد تمامی کارهای است که سیستم می‌تواند در یک زمان مشخص انجام دهد.

۱. Error! \blacklozenge

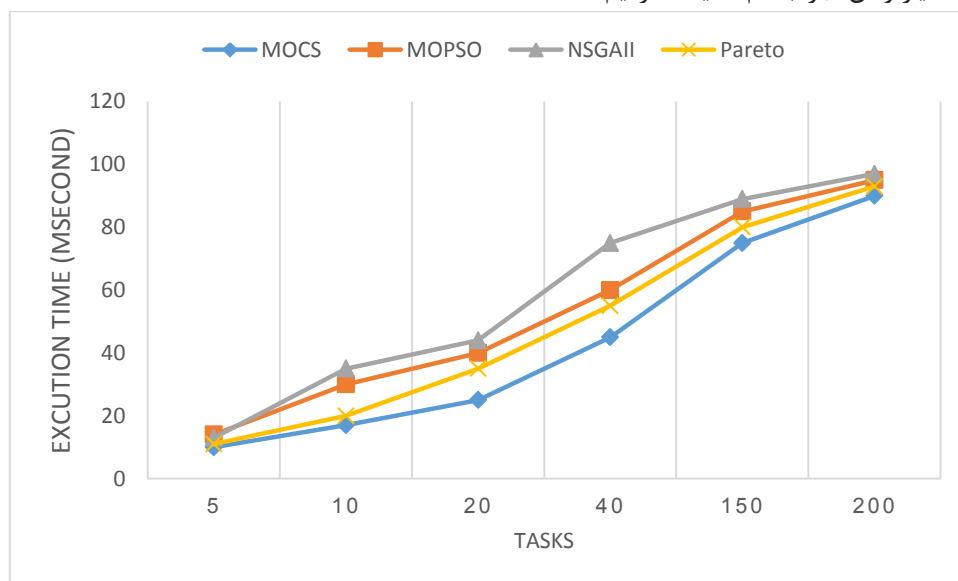
$$\text{Throughput} = \sum_{i=0}^m \text{completed tasks} / T \quad \blacklozenge$$

No text of specified style in document. (۲۵)

در این فرمول T بیانگر زبان شبیه‌سازی و m تعداد کارهای تمام‌شده است.

مقایسه الگوریتم فاخته با چند الگوریتم هم‌سطح بر اساس عامل زمان اجرا

در اینجا الگوریتم فاخته چند هدفه پیشنهادی را با الگوریتم NSGAI و MOPSO با مقادیر ورودی یکسان و شرایط محیط شبیه‌سازی برابر، اجرا کرده و نتایج بدست آمده را در ازای تعداد مختلف کارهای توزیع داده شده در بازه ی صفر تا ۲۰۰ بر اساس معیار زمان اجرا با هم مقایسه کردیم.

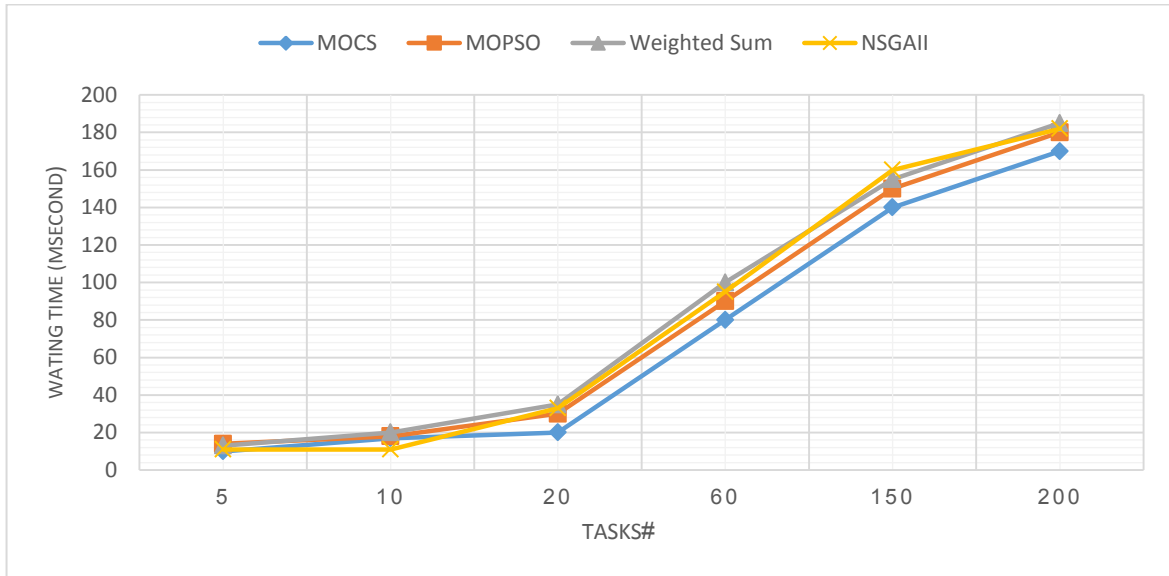


شکل ۵. مقایسه الگوریتم فاخته با چند الگوریتم هم‌سطح بر اساس زمان اجرا

همانگونه که در نمودار نیز قابل مشاهده می‌باشد روش پیشنهادی از سایر الگوریتم‌ها کارایی بهتری از خود نشان داده و به نسبت مقادیر NSGAI از بهبود ۱۰٪ برخوردار می‌باشد طبیعتاً هر چه مقدار متوسط زمان اجرای کارها کمتر باشد ناشی از عملکرد بهینه‌تر راهکار پیشنهادی خواهد بود.

مقایسه الگوریتم پیشنهادی با سایر الگوریتم‌های هم‌سطح بر اساس زمان انتظار

در این قسمت الگوریتم پیشنهادی در شرایط شبیه‌سازی شده برابر مورد سنجش قرار گرفت و در بازه ی تعداد کار توزیع داده شده برابر در هر زمان، از نظر زمان انتظار توزیع کارها با دو الگوریتم مورد مقایسه دیگر نتیجه قابل قبولی از خود نشان داد. این معیار معرف این می‌باشد هر کار متوسط چه مقدار می‌بایست منتظر بماند تا به مرحله اجرا برسد.



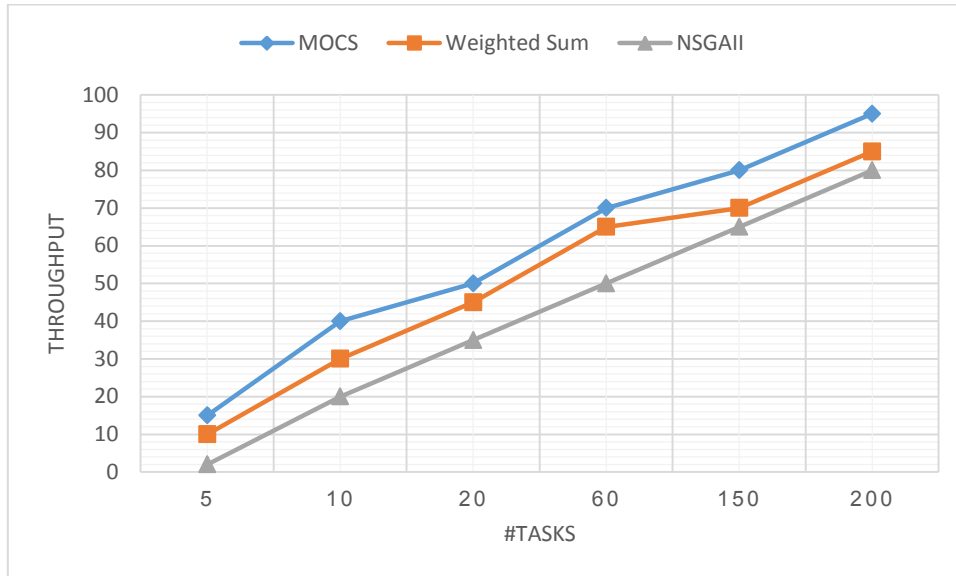
شکل ۶. مقایسه الگوریتم پیشنهادی با سایر الگوریتم‌های هم‌سطح بر اساس زمان انتظار

با در نظر گرفتن زمان انتظار همانگونه که در نمودار نیز قابل مشاهده می‌باشد روش پیشنهادی در قیاس با روش متوسط وزن زمانی از کاهش ۳۰٪ برخوردار بوده و بیانگر بهبود روش پیشنهادی می‌باشد. دلیل این امر نیز زمان انتظار درون سیستم به حداقل ترین مقدار ممکن رسیده است. باید توجه داشت اعدادی که در نمودار نشان داده شده است زمان کلی انتظار کارها می‌باشد. بنابراین اگر زمان هر یک از بازه‌ها به تعداد کارها تقسیم شود به متوسط زمان انتظار هر کار خواهیم رسید برای نمونه اگر زمان انتظار ۵ کار ۲۰ میلی ثانیه باشد، متوسط زمان انتظار هر کار ۴ میلی ثانیه خواهد بود. این امر در مورد متوسط زمان اجرای کارها نیز صادق می‌باشد.

مقایسه بازدهی الگوریتم پیشنهادی به نسبت الگوریتم NSGAI

پس از اجرای هر سه الگوریتم در محیط شبیه سازی شده برابر و داده های مشابه و محاسبه میزان بازدهی بر اساس فرمول تعریف شده، نتیجه مقایسه برابر با نمودار زیر بدست آمده است:

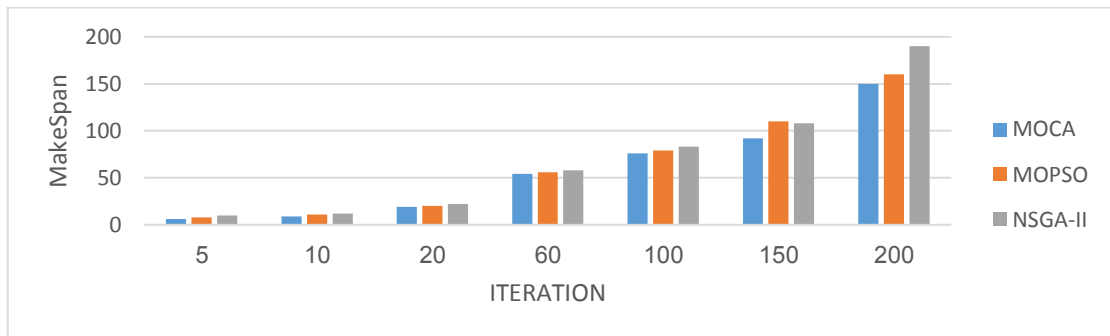
برای بازدهی سیستم هم به همان شیوه و با مشخصاتی که در بالا ذکر شد معیار مورد نظر مورد ارزیابی قرار گرفت. روش کار در این معیار بدین صورت است که شما بازه زمانی را در نظر می‌گیرید و آزمایش می‌کنید که چه تعداد کار می‌تواند در آن بازه زمانی اجرا شوند. اگر به نمودار آزمایش مورد نظر بنگریم متوجه خواهیم شد که بازدهی روش پیشنهادی به نسبت مقدار مجموع وزنی ۱۰٪ و به نسبت میزان الگوریتم NSGAI از بهبود نسبی ۲۰٪ برخوردار است.



شکل ۷. مقایسه بازدهی الگوریتم پیشنهادی به نسبت الگوریتم NSGAI

مقایسه الگوریتم پیشنهادی با سایر الگوریتم‌های هم‌سطح بر اساس Makespan

در این مسئله n شغل J_1, J_2, \dots, J_n با اندازه‌های متفاوت که باید روی m ماشین یکسان زمان‌بندی شوند در تلاشند تا زمان کل (make span) را به حداقل برسانند. زمان خالی مجموع زمان لازم برای انجام همه شغل‌هاست (که همه شغل‌ها تمام شده‌اند). روش پیشنهادی همراه با سایر الگوریتم‌ها مورد ارزیابی قرار گرفت و نتایج آن در نمودار زیر نمایش داده شده است.

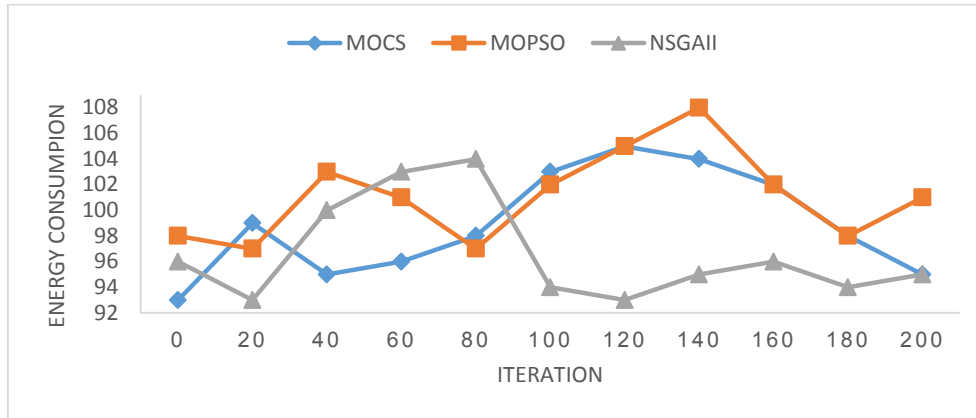


شکل ۸. مقایسه الگوریتم پیشنهادی با سایر الگوریتم‌های هم‌سطح بر اساس متوسط عامل Makespan

مشخص است که هدف هر الگوریتم زمان‌بندی به حداقل رساندن زمان اجرای کارها می‌باشد. بنابراین روش پیشنهادی نیز از این قاعده مستثنی نمی‌باشد. و همانگونه که در نمودار مشخص است در این زمینه به نسبت سایر الگوریتم‌ها موفق‌تر عمل کرده است.

مقایسه میزان مصرف انرژی الگوریتم پیشنهادی با سایر الگوریتم‌ها

سناریو دیگری که ما برای کارایی الگوریتم پیشنهادی در نظر گرفتیم میزان انرژی مصرفی الگوریتم پیشنهادی در برابر سایر الگوریتم‌ها بود. برای این منظور میزان انرژی مصرفی بهینه‌ترین جواب‌ها در هر تکرارهای مختلف اندازه‌گیری شد و نتایج آن در نمودار زیر نمایش داده شده است.

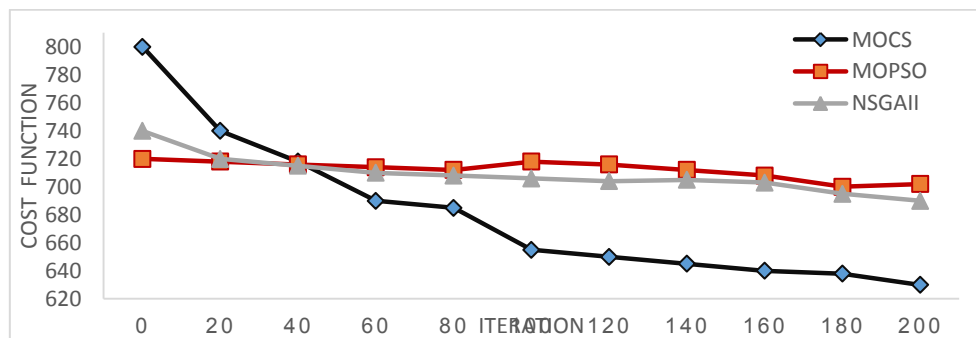


شکل ۹. مقایسه میزان مصرف انرژی الگوریتم پیشنهادی با سایر الگوریتم‌ها

با توجه به نمودار می‌توان دریافت که الگوریتم پیشنهادی در رابطه با مصرف انرژی به نسبت الگوریتم MOPSO بهتر و به نسبت NSGAI بدتر عمل کرده. نهایتاً بررسی که صورت گرفت آزمایش همگرایی الگوریتم پیشنهادی است که بر اساس تابع هزینه کلی است که متشکل از دو هدف اصلی مسئله زمان بندی است.

مقایسه الگوریتم پیشنهادی بر اساس تابع هزینه در مقایسه با سایر الگوریتم‌ها

در نهایت بعد از مقایسه الگوریتم‌های اشاره شده تا کنون ثابت کردیم بدلیل همگرایی بالای الگوریتم پیشنهادی، عملکرد این الگوریتم بر اساس تابع هزینه از سایر الگوریتم‌ها از تعداد مشخصی از تکرار الگوریتم به بعد بهبود قابل قبولی را از خود نشان می‌دهد.



شکل ۱۰. مقایسه الگوریتم پیشنهادی بر اساس تابع هزینه در مقایسه با سایر الگوریتم‌ها

از نمودار بالا استنباط می‌شود که روش پیشنهادی به نسبت سایر الگوریتم‌ها از لحاظ بهینگی بهتر عمل کرده و در تکرارهای بالاتر به سمت کمینه کردن تابع هدف یا همان بهینه‌سازی جواب بهتر عمل کرده در صورتی که دو الگوریتم مورد مقایسه قرار گرفته‌شده در قیاس با این الگوریتم به روندی ثابت تمایل پیدا می‌کنند. چرایی این مسئله را نیز باید در ماهیت الگوریتم فاخته جستجو نمود چرا که در هر تکرار فاخته‌های که از نظر تابع هدف یا هزینه بد تشخیص داده می‌شوند حذف می‌شوند. البته نباید تأثیر پرواز لوی را دستکم گرفت.

نتیجه‌گیری

مدل رایانشی بر پایه شبکه‌های رایانه‌ای مانند اینترنت است که الگویی تازه برای عرضه، مصرف و تحویل خدمات رایانشی (شامل زیرساخت، نرم‌افزار، بستر، و سایر منابع رایانشی) با به‌کارگیری شبکه ارائه می‌کند. «رایانش ابری» از ترکیب دو کلمه رایانش و ابر ایجاد شده‌است. ابر در اینجا استعاره از شبکه یا شبکه‌های وسیع مانند اینترنت است که کاربر معمولی از پشت صحنه و آنچه در پی آن اتفاق می‌افتد اطلاع دقیقی ندارد (مانند داخل ابر) در نمودارهای شبکه‌های رایانه‌ای نیز از شکل ابر برای نشان دادن شبکه اینترنت استفاده می‌شود.



شکل ۱۱. نمودار مفهومی رایانش ابری

محاسبات ابری یکی از مدل‌ها برای میزبانی و سرویس‌دهی بر بستر اینترنت است. این مدل نقش مهمی را در سرویس‌دهی ایفا می‌کند چراکه می‌تواند به هر کس و هر سازمانی بنا به سطح نیاز از پایین‌ترین تا سطح تا امکانات و منابع گسترده و زیاد خدمات‌دهی کند. تکنولوژی‌های جدید برای ارائه خدمات باکیفیت‌تر برای کاربران به وجود آمده. سیستم‌های مبتنی بر محاسبات ابری با حجم بسیار زیاد از منابع و داده‌ها درگیر می‌باشند که مستلزم مصرف انرژی فراوانی است. از طرف دیگر زمان پاسخگویی به درخواست‌های کاربر از مهم‌ترین مسئله‌ها و چالش‌ها در حوزه محاسبات ابری است. بنابراین مسئله زمان‌بندی به‌عنوان یکی از چالش‌ها مطرح می‌شود. در حوزه‌های مختلفی سعی شده است، کارایی بهینه زمان‌بندی کارها که ضروری نیز است بهبود داده شود. از منظرهای متفاوتی می‌توان به مسئله زمان‌بندی در محیط ابر نگریسته شود. چراکه مسئله زمان‌بندی تأثیر مستقیمی بر کیفیت سرویس در محیط ابر دارد و خود این مسئله نیز از پارامترهای مختلفی تشکیل شده تا بر این اساس کلیت کیفیت سرویس را تشکیل دهد. روش‌ها مختلفی برای بهبود مسئله زمان‌بندی ارائه شده است. از روش‌های ابتکاری گرفته تا روش‌های فرا تکاملی. در این پژوهش سعی شد که به مسئله زمان‌بندی به‌عنوان یک مسئله با یک پارامتر و یا یک تابع نگریسته نشود. بلکه در فاز نخست سعی شد مسئله به‌عنوان یک مسئله چندهدفه مطرح و نگاه‌شده شود. بنابراین سعی شد تابع هدف یا هزینه روش پیشنهادی دارای یک هدف نباشد بلکه سعی شود بر اساس پارامترهای مختلف که می‌تواند در یک راستا و یا اینکه متضاد باشند روش پیشنهادی هزینه زمان‌بندی را کاهش داده و موجب ارتقاء کیفیت سرویس شود. روش پیشنهادی بر این اساس عمل می‌کند که جایگشتی از ماشین‌های مجازی به‌عنوان ماشین‌های انتسابی به کارها به همراه خودکارها به‌عنوان ورودی در نظر گرفته می‌شوند. در هر بار تکرار الگوریتم جستجوی فاخته چندهدفه جایگشت تولیدی توسط فاخته‌ها به‌عنوان جوابی از مسئله توسط تابع هدف ارزیابی می‌شوند و هر کدام از جواب‌ها که شرایط موردنظر را دارا باشند به‌عنوان نقاط پارتو در نظر گرفته می‌شوند و نهایتاً جواب بهینه در بهینه‌های مرتب‌شده به‌عنوان جواب بهینه در نظر گرفته خواهد شد. همان‌گونه که شرح داده شد روش پیشنهادی از کارایی بهتری به نسبت سایر الگوریتم‌های چندهدفه که موردبررسی و مقایسه قرار گرفته بودن از خود نشان داده بود. از نمودار مصرف انرژی استنباط می‌شود که تنها در این مورد به نسبت الگوریتم NSGA II خوب عمل نکرده است.

در این مقاله به مفاهیم اولیه رایانش ابری، مدل‌ها و عناصر زیربنایی بیان شده و روش‌های توزیع منابع، اهداف و چالش‌ها، الگوریتم‌های موجود مطرح شده است. در ادامه الگوریتم‌های ابتکاری و فرا تکاملی و انواع آن‌ها به اختصار توضیح داده شده است. سپس الگوریتم فاخته، بهینه‌سازی چندهدفه و روش پیشنهادی شامل نگاهت الگوریتم به راه حل پیشنهادی، توابع ارزیابی شامل زمان پاسخ، زمان انتظار، هزینه و انرژی مصرفی بیان شده است. نحوه اجرا و تست الگوریتم پیشنهادی با استفاده از زبان برنامه‌نویسی جاوا و نرم‌افزارهای نت بینز، ایکلیپس و کلودسیم پیاده‌سازی و نتایج مورد نظر و بهینه‌ی الگوریتم

بدست آمده و نتایج شبیه‌سازی نشان می‌دهد که الگوریتم MOCOا، از الگوریتم‌های NSGAI و MOPSO بهتر عمل می‌کند.

کارهای آتی

هر الگوریتم بهینه‌سازی چندهدفه جدید می‌تواند برای زمان‌بندی توزیع منابع ابر به کارهای مشتری برای رسیدن به کمترین زمان پاسخ، زمان انتظار و هزینه و انرژی مصرفی کارها اعمال گردد. الگوریتم‌های فرا تکاملی چندهدفه با دقت و عملکرد بهتر و ترکیب اهداف مختلف می‌توانند نتایج بهتری را حاصل کنند.

در زمان‌بندی، مسئله هزینه اعم از هزینه زمان اجرا، هزینه مهاجرت، هزینه ارتباطی و ... اهمیت زیادی دارد. انتخاب الگوریتم مناسب که بتواند باعث کاهش این هزینه‌ها شود از جمله مواردی است که به آن توجه زیادی شده است. در تحقیقات آتی، الگوریتم MOCOا می‌تواند با بهبود عملکرد بین وظایف و هزینه‌های منابع و با تلفیق با سایر الگوریتم‌های فرا تکاملی موجود توسعه یابد.

یکی از این الگوریتم‌ها، PSO است و مشاهده شده که هزینه اجرای وظایف کم و وظایف برای اجرا به حداقل تعداد منابع نیاز دارند بنابراین در تلفیق با خصوصیات منحصر بفرد الگوریتم فاخته می‌تواند بهره‌وری توزیع منابع را نیز بهبود ببخشد. پیشنهاد بعدی استفاده تلفیقی با الگوریتم GA است که سرعت نسبتاً مناسبی دارد و می‌تواند در بالا بردن عملکرد الگوریتم بسیار موثر باشد، بنابراین استفاده ترکیبی از این الگوریتم‌ها می‌تواند در طراحی یک الگوریتم موثر و کارآمد به منظور کاهش هزینه و قابلیت اطمینان در ابر مناسب باشد.

منابع

۱. اصغرپور حسین، رضازاده علی. تعیین سبد بهینه سهام با استفاده از روش ارزش در معرض خطر. فصلنامه نظریه‌های کاربردی اقتصاد. ۱۳۹۴. شماره ۴: ۹۳-۱۱۸.
۲. اقبال نیا محمد. دلیران سید مازیار. بهینه‌سازی سبد سهام به شیوه فازی و با استفاده از الگوریتم فراابتکاری جستجوی ناخودآگاه. فصلنامه دانش سرمایه‌گذاری. ۱۳۹۸. شماره ۳۲: ۲۶۹-۲۵۱.
۳. ایمانی کله سر، هوشیار، اسمعیلی اقدم، امیررضا. (۱۴۰۲). بهینه‌سازی اندازه سازه‌های خرابی با استفاده از الگوریتم بازار بورس. مهندسی سازه و ساخت. doi: 10.22065/jsce.2024.405938.3168
۴. تهرانی رضا، فلاح تفتی سیما، آصفی سپهر. بهینه‌سازی سبد سهام به کمک الگوریتم فراابتکاری دسته‌های میگو با استفاده از معیارهای مختلف ریسک در بورس اوراق بهادار تهران. فصلنامه تحقیقات مالی. ۱۳۹۷. شماره ۴: ۴۲۶-۴۰۹.

5. Abd Al Ghaffar, Hedaia-t-Allah Nabil. "Government cloud computing and national security." *Review of Economics and Political Science* 9.2 (2024): 116-133.
6. Abdulkader, S. J. and A. M. Abualkishik (2013). "Cloud Computing and E-commerce in Small and Medium Enterprises (SME's): the Benefits, Challenges." *International Journal of Science and Research (IJSR)* 2(12): 285-288.
7. Abdulwahab, H. M., Ajitha, S., Saif, M. A. N., Murshed, B. A. H., & Ghanem, F. A. (2024). MOBCSA: Multi-Objective Binary Cuckoo Search Algorithm for Features Selection in Bioinformatics. *IEEE Access*.
8. Abts, D. and B. Felderman (2012). "A guided tour of data-center networking." *Communications of the ACM* 55(6): 44-51.
9. Ahn, J., et al. (2012). Dynamic Virtual Machine Scheduling in Clouds for Architectural Shared Resources. *HotCloud*.
10. Alnajdi, S., et al. "A Survey ON RESOURCE ALLOCATION IN CLOUD COMPUTING".

11. Amazon, E. (2008). Amazon elastic compute cloud (amazon ec2) 2010.
12. Amazon, E. (2009). Amazon EC2 service level agreement.
13. Amazon, E. (2012). Instance Types: <http://aws.amazon.com/ec2/instance-types>, Accessed.
14. Armbrust, M., et al. (2010). "A view of cloud computing." *Communications of the ACM* 53(4): 50-58.
15. Atadoga, A., Umoga, U. J., Lottu, O. A., & Sodiya, E. O. (2024). Evaluating the impact of cloud computing on accounting firms: A review of efficiency, scalability, and data security. *Global Journal of Engineering and Technology Advances*, 18(2), 065-074.
16. Babu, K. D. and D. G. Kumar "Allocation Strategies Of Virtual Resources In Cloud Computing Networks." *Journal Of Engineering Research And Applications* 201: 51-55.
17. Bari, M. F., et al. (2013). "Data center network virtualization: A survey." *IEEE Communications Surveys & Tutorials* 15(2): 909-928.
18. Blau, B., et al. (2008). Planning and pricing of service mashups. *E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 2008 10th IEEE Conference on*, IEEE.
19. Branch, K. (2015). "A novel task scheduling method in cloud environment using cuckoo optimization algorithm." *International Journal of Cloud-Computing and Super-Computing* 2(2): 7-20.
20. Buyya, R., et al. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*, Ieee.
21. Çavdar, M. C., Korpeoglu, I., & Ulusoy, Ö. (2024). A Utilization Based Genetic Algorithm for virtual machine placement in cloud systems. *Computer Communications*, 214, 136-148.
22. Chauhan, N., Kaur, N., Saini, K. S., Verma, S., Alabdulatif, A., Khurma, R. A., ... & Castillo, P. A. (2024). A Systematic Literature Review on Task Allocation and Performance Management Techniques in Cloud Data Center. *arXiv preprint arXiv:2402.13135*.
23. Csanád, M., & Kincses, D. (2024). Femtoscopy with Lévy sources from SPS through RHIC to LHC. *Universe*, 10(2), 54.
24. Dréo, J., et al. (2006). *Metaheuristics for hard optimization: methods and case studies*, Springer Science & Business Media.
25. Farokhi, S. (2014). Towards an SLA-based service allocation in multi-cloud environments. *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*, IEEE.
26. Fujiwara, I., et al. (2009). "Market-based resource allocation for distributed computing." *IPSJ SIG Technical Report*: 835-848.
27. Govindan, S., et al. (2011). Cuanta: quantifying effects of shared on-chip resource interference for consolidated virtual machines. *Proceedings of the 2nd ACM Symposium on Cloud Computing*, ACM.
28. Herbst, N. R., et al. (2013). *Elasticity in Cloud Computing: What It Is, and What It Is Not*. ICAC.

29. Jennings, B. and R. Stadler (2015). "Resource management in clouds: Survey and research challenges." *Journal of Network and Systems Management* 23(3): 567-619.
30. Jiao, J. R., et al. (2006). "An agent-based framework for collaborative negotiation in the global manufacturing supply chain network." *Robotics and Computer-Integrated Manufacturing* 22(3): 239-255.
31. Kalra, M. and S. Singh (2015). "A review of metaheuristic scheduling techniques in cloud computing." *Egyptian informatics journal* 16(3): 275-295.
32. Karaboga, D. and B. Basturk (2007). "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm." *Journal of global optimization* 39(3): 459-471.
33. Kumar, A., et al. (2013). An efficient framework for resource allocation in cloud computing. *Computing, Communications and Networking Technologies (ICCCNT)*, 2013 Fourth International Conference on, IEEE.
34. Kyriazis, D. (2013). "Cloud computing service level agreements—exploitation of research results." *European Commission Directorate General Communications Networks Content and Technology Unit*, Tech. Rep 5: 29.
35. Lee, G., et al. (2010). Topology-aware resource allocation for data-intensive workloads. *Proceedings of the first ACM asia-pacific workshop on Workshop on systems*, ACM.
36. Li, J., et al. (2012). "Online optimization for scheduling preemptable tasks on IaaS cloud systems." *Journal of Parallel and Distributed Computing* 72(5): 666-677.
37. Lin, W., et al. (2011). "A threshold-based dynamic resource allocation scheme for cloud computing." *Procedia Engineering* 23: 695-703.
38. Lin, W.-Y., et al. (2010). Dynamic auction mechanism for cloud resource allocation. *Cluster, Cloud and Grid Computing (CCGrid)*, 2010 10th IEEE/ACM International Conference on, IEEE.
39. Liu, J., et al. (2013). "Job scheduling model for cloud computing based on multi-objective genetic algorithm." *IJCSI International Journal of Computer Science Issues* 10(1): 134-139.
40. Mell, P. and T. Grance (2011). "The NIST definition of cloud computing".
41. Mishra, A., & Goel, L. (2024). Metaheuristic algorithms in smart farming: An analytical survey. *IETE Technical Review*, 41(1), 46-65.
42. MT, I. (2024). Differential evolution ensemble designer.
43. Nazemi, S., & Khorsand, R. (2024). C-KHCS: Multi-Objective Workflow Scheduling using Chaotic Krill Herd Optimization and Improved Cuckoo Search in Fog Computing. *IEEE Transactions on Services Computing*.
44. Nebagiri, M. H., & Hnumanthappa, L. P. (2024). Multi-Objective of Load Balancing in Cloud Computing using Cuckoo Search Optimization based Simulation Annealing. *International Journal of Intelligent Systems and Applications in Engineering*, 12(9s), 466-474.
45. NIST, N. (2009). "Definition of cloud computing v15." NIST, Editor.
46. Papazoglou, M. P. and W.-J. Heuvel (2007). "Service oriented architectures: approaches, technologies and research issues." *The VLDB Journal—The International Journal on Very Large Data Bases* 16(3): 389-415.

47. Popovici, F. I. and J. Wilkes (2005). Profitable services in an uncertain world. Proceedings of the 2005 ACM/IEEE conference on Supercomputing, IEEE Computer Society.
48. Putzier, M., Khakzad, T., Dreischarf, M., Thun, S., Trautwein, F., & Taheri, N. (2024). Implementation of cloud computing in the German healthcare system. *NPJ Digital Medicine*, 7(1), 12.
49. Rajabioun, R. (2011). "Cuckoo optimization algorithm." *Applied soft computing* 11(8): 5508-5518.
50. Rehan, H. (2024). Revolutionizing America's Cloud Computing the Pivotal Role of AI in Driving Innovation and Security. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, 2(1), 189-208.
51. Salot, P. (2013). "A survey of various scheduling algorithm in cloud computing environment." *International Journal of Research in Engineering and Technology* 2(2): 131-135.
52. Shadkam, E. and N. Jahani (2015). "A hybrid COA ϵ -constraint method for solving multi-objective problems." *arXiv preprint arXiv:1509.08302*.
53. Son, S., et al. (2013). "An SLA-based cloud computing that facilitates resource allocation in the distributed data centers of a cloud provider." *The Journal of Supercomputing* 64(2): 606-637.
54. Srinivas, N. and K. Deb (1994). "Multiobjective optimization using nondominated sorting in genetic algorithms." *Evolutionary computation* 2(3): 221-248.
55. Tembhrne, P. S. (2024). EVALUATING THE USE OF CLOUD-COMPUTING CONCEPT EMERGED IN THE HUMAN RESOURCE MANAGEMENT FIELD TO INCREASE THE PRODUCTIVITY IN THE ORGANIZATION. *Vegueta. Anuario de la Facultad de Geografía e Historia*, 24, 1.
56. Wu, L., et al. (2011). Sla-based resource allocation for software as a service provider (saas) in cloud computing environments. *Cluster, Cloud and Grid Computing (CCGrid)*, 2011 11th IEEE/ACM International Symposium on, IEEE.
57. You, X., et al. (2009). RAS-M: resource allocation strategy based on market mechanism in cloud computing. *ChinaGrid Annual Conference, 2009. ChinaGrid'09. Fourth*, IEEE.
58. Zaman, S. and D. Grosu (2013). "A combinatorial auction-based mechanism for dynamic VM provisioning and allocation in clouds." *IEEE Transactions on Cloud Computing* 1(2): 129-141.
59. Zhan, S. and H. Huo (2012). "Improved PSO-based task scheduling algorithm in cloud computing." *Journal of Information & Computational Science* 9(13): 3821-3829.
60. Zhan, Z.-H., et al. (2015). "Cloud computing resource scheduling and a survey of its evolutionary approaches." *ACM Computing Surveys (CSUR)* 47(4): 63.
61. Zhang, Q., et al. (2011). Dynamic resource allocation for spot markets in cloud computing environments. *Utility and Cloud Computing (UCC)*, 2011 Fourth IEEE International Conference on, IEEE.