

زمان‌بندی وظایف در ابرهای چندمکانه با استفاده از الگوریتم‌های تکاملی

مژده ایوانی^{۱*}، حسنا قهرمانی^۲

۱) - گروه کامپیوتر، دانشکده فنی مهندسی دانشگاه آزاد اسلامی واحد کرمانشاه

* mozhdeh.eivani@gmail.com

ارسال: اردیبهشت ماه ۹۸ پذیرش: خرداد ماه ۹۸

چکیده

رایانش ابری، مدل شبکه‌ای جدیدی است که با استفاده از پردازشگرهای متصل به هم امکان انجام دادن عملیات حجیم محاسباتی را میسر می‌سازد. به خاطر سرویس‌هایی که توسط محیط ابری ارائه می‌شود در بین کاربران از محبوبیت بالایی برخوردار است. با توجه به فراوانی کاربران، سرویس‌دهنده نیاز به مدیریت بار کاری و بهره‌گیری بهینه از منابع دارد. یکی از بهترین وظایف در ابر نگاشت صحیح وظایف به منابع می‌باشد، زمان‌بندی یک وظیفه فرایندی مهم در سیستم‌های توزیع شده مثل ابر است و تاثیر زیادی روی کارایی سیستم دارد که این امر توسط الگوریتم‌های زمان‌بند انجام می‌شود لذا ارائه و استفاده از الگوریتم زمان‌بندی کارآمد در محیط ابر حائز اهمیت است. راه حل برای اینگونه مسائل با پیچیدگی‌های زیاد، استفاده از روش‌های فراابتکاری و الگوریتم‌های بهینه‌سازی تکاملی می‌باشد. بدین منظور، در این مقاله جهت به حداقل رساندن زمان انجام وظایف از الگوریتم بهینه‌سازی گرگر خاکستری (GWO) به صورت چند هدفه استفاده شده است. علت این ارتقاء عملکرد، چند هدفه بودن طرح پیشنهادی نسبت به تک هدفه بودن تابع برازش طرح‌های پایه (تنها بر مبنای کمینه کردن زمان تکمیل وظایف) می‌باشد. برای این منظور، جهت شبیه‌سازی طرح پیشنهادی از محیط کلودسیم و نصب افزونه workflowsim استفاده شده است.

کلمات کلیدی: رایانش ابری، زمان‌بندی، GWO، الگوریتم‌های بهینه‌سازی تکاملی، روش‌های فراابتکاری.

۱- مقدمه

با رشد روزافزون اینترنت، نیاز به سرویس‌های متعدد مبتنی بر اینترنت نیز رو به افزایش است. یکی از سرویس‌های محبوبی که اخیراً توسط کاربران و جامعه علمی بسیار مورد استفاده قرار گرفته است، سرویس رایانش ابری است. بنا به تعریف موسسه ملی و فناوری استاندارد، رایانش ابری به صورت زیر تعریف می‌شود: «رایانش ابری مدلی است برای فراهم‌آوری دسترسی آسان بر اساس تقاضای کاربر، از طریق شبکه به مجموعه‌ای از منابع محاسباتی قابل تغییر و پیکربندی (مانند شبکه‌ها، سرورها، فضای ذخیره‌سازی، برنامه‌های کاربردی و سرویس‌ها) که این دسترسی بتواند با کمترین نیاز به مدیریت منابع و نیاز به دخالت مستقیم فراهم‌کننده سرویس، به سرعت فراهم شده یا آزاد گردد [۲].»

رایانش ابری به عنوان یک الگوی جدید برای تأمین خدمات پویای محاسباتی، بسیار مورد توجه است. با استفاده از رایانش ابری، پردازش های بسیار بزرگ می توانند به راحتی اجرا شود. ایده پردازش ابری، همان تکامل یافته پردازش موازی، پردازش توزیع شده و پردازش شبکه ای است. همه خدمات ارائه شده در رایانش ابری مبتنی بر وب است. خدمات مبتنی بر وب به این معناست که یک سیستم نرم افزاری برای پشتیبانی از تعامل بین ماشین ها در سراسر شبکه طراحی شود [۳].

ابری تصویری انتزاعی از شبکه ای عظیم و توده ای است که حجم آن مشخص نیست؛ مکان قرارگیری سخت افزارها و نرم افزارها، ابعاد زمانی و مکانی مشخص نیست. تنها چیزی که در دسترس قرار می گیرد قابل شناسایی است. در واقع این شبکه تمام جزئیات فنی از دید کاربران خود پنهان کرده است. شما به جای حفظ اطلاعات روی رایانه شخصی آن را روی سروری در اینترنت نگهداری می کنید و در پردازش ابری تنها وظیفه ای که بر عهده رایانه شخصی است ارتباط برقرار کردن با ابر است که به سادگی اتصال به یک سرور اینترنت است و از آنجا به بعد تمام کار توسط ابر رایانه ای پردازش می شود [۳].

یک سرویس، نقطه پایانی از یک اتصال است و دارای بعضی از انواع سیستم های کامپیوتری زیربنایی است که ارائه این اتصال را حمایت می کند. خدمات داخلی و خارجی از یک سازمان، معماری سرویس گرا را تشکیل می دهند. ارائه دهندگان سرویس ابری و مصرف کنندگان دو جهت سرویس های رایانش ابری را تشکیل می دهند. خدمات متمایز رایانش ابری از جمله نرم افزار به عنوان سرویس، بستر به عنوان سرویس و زیرساخت به عنوان سرویس و حتی اگر بخواهیم به صورت جزئی تر بیان کنیم، اطلاعات به عنوان سرویس، ذخیره سازی به عنوان سرویس، پهنای باند به عنوان سرویس، امنیت به عنوان سرویس، طراحی به عنوان یک سرویس، محصول به عنوان یک سرویس، ارتباط به عنوان سرویس، سرگرمی به عنوان سرویس، محافظت به عنوان سرویس، بهره وری به عنوان سرویس، مدیریت زمان به عنوان سرویس، پاداش به عنوان سرویس، رویداد به عنوان سرویس و سرویس به عنوان سرویس می تواند ارائه شود [۴].

این فناوری جدید به دلیل ویژگی هایش به سرعت محبوب شده است، زیرا در رایانش ابری همه نوع امکاناتی به کاربران به عنوان یک سرویس ارائه شده است. طبیعتاً هر تغییر و مفهوم جدیدی در دنیای فناوری، مشکلات و پیچیدگی های خاص خود را دارد. بهره گیری از رایانش ابری نیز از این قاعده مستثنا نبوده و چالش های فراوانی را پیش روی صاحب نظران این حوزه قرار داده است که از آن جمله می توان به مواردی نظیر زمان بندی، توازن بار، امنیت، قابلیت اطمینان، مالکیت، پشتیبان گیری از داده ها و قابلیت حمل داده ها اشاره کرد [۵].

محاسبات ابری نسبت به روش های سنتی توزیع و به کارگیری نرم افزار مزایای فراوانی دارد. محاسبات ابری سازمان ها را قادر می سازد امکانات خود را به سرعت گسترش داده و در هزینه ها صرفه جویی نمایند و برنامه های سازمانی خود را ایجاد و توسعه دهند. مهم ترین مشکل در پذیرش این فناوری، نگرانی های مربوط به حفاظت از اطلاعات محرمانه است. همچنین بزرگ ترین مزیت آن عدم نیاز سازمان ها در سرمایه گذاری عظیم در حوزه فناوری اطلاعات است [۶].

در کنار مزیت های بی شمار رایانش ابری، چالش های بسیاری نیز با آن همراه بوده است که یکی از مهم ترین چالش های موجود در رایانش ابری مساله زمان بندی کارها (Task) یا اختصاص منابع به درخواست های کاربران است، دلایل متعددی باعث شده است که این موضوع به عنوان یک مساله NP-Complete نمود پیدا کند. روش های بهینه سازی تکاملی، روش هایی هستند که می توانند جوابی خوب (نزدیک به بهینه) در زمانی محدود برای یک مساله ارائه کنند. این روش ها از نظر حوزه کاربرد، کاملاً عمومی هستند و می توانند مسائل خیلی پیچیده را حل کنند. عمده این روش ها تصادفی بوده و از طبیعت الهام گرفته شده اند. در واقع روش پیشنهادی ما در این پایان نامه یک روش زمان بندی وظایف در ابرهای چندمکانه باهدف کاهش زمان اجرا و کاهش هزینه های مرکز داده است که با زمان بندی صحیح وظایف و تخصیص درست منابع به آن ها صورت می گیرد. به عبارت دیگر، در این مطالعه به بررسی و تحقیق مسئله زمان بندی کار مبتنی بر الگوریتم های بهینه سازی تکاملی در ابرهای چندمکانه می پردازیم. جهت به حداقل رسانی زمان اتمام (Makespan) و زمان اتمام کارها می باشد.

۲- مروری بر پیشینه تحقیق

تاکنون مقالات متعددی در زمینه در زمینه مسئله‌ی زمان بندی وظایف در حوزه رایانش ابری به چاپ رسیده است. شرح تعدادی از مهمترین این مقالات در این بخش ارائه شده است.

جیانگ^۱ و همکارانش روشی برای زمان بندی داده آگاه در ابر پیشنهاد دادند. الگوریتم آن‌ها که یک زمان بندی دسته‌ای به حساب می‌آید، با پروفایل کردن اطلاعات گره‌ها و کارها، زمان بندی را انجام می‌داد. پس از جمع‌آوری اطلاعات، الگوریتم تصمیم می‌گرفت که بهترین گره برای اجرای کار کدام خواهد بود. آن‌ها روش خود را پیاده‌سازی کردند و نتایج نشان داد که الگوریتم آن‌ها بهبودی خوبی را ایجاد کرده است [۷].

سائو^۲ و همکارانش به حل مسئله‌ی پردازش LIGO^۳ در ابر پرداختند. آن‌ها یک تئوری بهبود برای زمان بندی را ارائه کردند. آن‌ها این کار را با استفاده از یک الگوریتم تطابق با محیط اجرا کردند و نتایج خود را در محیط کلودسیم پیاده‌سازی کردند [۸].

مهدی و همکارانش (۱۳۹۵) به مسئله‌ی زمان بندی وظایف در حوزه‌ی رایانش ابری به‌عنوان یک مسئله‌ی حداقل سازی با استفاده از روش‌های تکاملی الگوریتم ژنتیک و الگوریتم بهینه‌سازی زنبور عسل تحلیل پرداختند. در این مقاله قصد آن‌ها روش backfill را به‌عنوان یک راهکار سنتی برای زمان بندی معرفی کرده و سپس با کمک الگوریتم‌های تکاملی مورد بحث در این مقاله، روش backfill را بهینه‌سازی نمودند. نتایج حاصل از این پژوهش توسط شبیه‌ساز کلودسیم شبیه‌سازی شد. نتایج این پژوهش حاکی از آنست که، حل مسئله زمان بندی وظایف و تقسیم منابع با استفاده از روش‌های تکاملی بسیار مقرون به‌صرفه تر از سایر روش‌های سنتی برای حل این مسئله است [۱].

لی^۴ و همکاران به مدل سازی زمان بندی در زمان بندی ابر پویا از طریق جابجایی ماشین‌های مجازی پرداختند. در مقاله مذکور، آن‌ها به ارزیابی زمان بندی‌های پویای کنونی پرداختند و یک مدل برنامه‌نویسی خطی برای زمان بندی پویا در ابر ارائه کردند. آن‌ها در نهایت به ارزیابی زمان بندی خود با استفاده از کلودسیم پرداختند. نتایج عملی آن‌ها نشان داد که روش آن‌ها می‌تواند در ابرهای عمومی و محیط همگن بهبودی ایجاد کند [۹].

ونگ^۵ و همکاران الگوریتمی جدید بانام JAWS را ارائه کردند. تمرکز آن‌ها بر روی بهبود زمان پاسخ‌دهی در پایگاه داده‌های عظیم بود. دیتاست مورد بررسی، دارای حجمی با مقیاس پتابایت بود. آن‌ها پرس‌وجوها را به زیر پرسش‌های مربوط به ورودی/خروجی تقسیم کردند، سپس نیازمندی داده‌هایی که همپوشانی^۶ دارند را درون بارکاری شناسایی کرده و بر اساس آن زمان بندی را انجام دادند. در نهایت، آن‌ها الگوریتم خود را با سایر الگوریتم‌ها از نظر میزان کارایی و زمان پاسخ مقایسه کردند که کارایی مناسبی را، بخصوص در زمان افزایش میزان کارها، از خود به نمایش گذاشت [۱۰].

ژنگ^۷ و همکارانش به بررسی احتمال قرارگیری ماشین‌های مجازی در یک سازمان‌دهی منسجم برای بهبود سرعت یافتن بهترین تخصیص و در عین حال افزایش حداکثر میزان استفاده از منابع پرداختند. سیاست زمان بندی آن‌ها با استفاده از یک الگوریتم ژنتیک موازی بود که بسیار سریع تر از الگوریتم ژنتیک سنتی بود. نتایج آن‌ها نشان داد که روش پیشنهادی آن‌ها در زمینه‌های اختصاص منابع و استفاده از منابع سیستم، بهبودی قابل توجهی را ایجاد کرده است [۱۱].

¹ Jiang

² Cao

⁴ Li

⁵ Wang

⁶ Overlap

⁷ Zheng

^۲ داده‌های به‌دست‌آمده از محاسبه گرانش زمین

پاندی^۸ و همکارانش یک زمان بندی مبتنی بر بهینه سازی ذرات را برای محیط های محاسبات ابری ارائه کردند. آن ها با استفاده از PSO یا بهینه سازی حرکت ذرات، کارها را در منابع ابری زمان بندی کردند که هم هزینه های محاسباتی و هم هزینه های انتقال را در نظر داشت. آن ها هزینه های محاسباتی و ارتباطی چند کار را با یکدیگر مقایسه کردند و در نهایت، میزان هزینه های صرفه جویی شده با استفاده از الگوریتم انتخاب بهترین منبع، محاسبه کردند. نتایج آن ها نشان داد که PSO می تواند تا سه برابر به میزان صرفه جویی بیشتری نسبت به BSR دستیابی پیدا کند و دارای توزیع خوب کارها در منابع است [۱۲].

فیکو^۹ و همکارانش میان افزاری برای اجرای شبیه سازی های ترکیبی سیستم های بحرانی بزرگ مقیاس استفاده کردند. در روش پیشنهادی آن ها، میان افزار اجازه ی تجمیع زیر سیستم های تقلید شده را می داد که باعث واقعی شدن هر چه بیشتر شرایط آزمایش می شد؛ بنابراین روش آن ها به صورت بهینه سازی تخصیص منابع در محیط ابری خصوصی شد [۱۳].

توسی^{۱۰} و همکارانش به بررسی فاکتورهای تأثیر گذار بر روی کیفیت سرویس و کاهش هزینه ها در ابرهای ترکیبی پرداختند. در این مطالعه آن ها به بررسی سیاست ها که به تصمیم گیری بر توزیع کارهای تولید شده توسط کاربران برای افزایش میزان استفاده از منابع و کاهش هزینه ها کمک می کند، بررسی کردند. نتایج شبیه سازی نشان داد که سیاست های ارائه شده باعث بهبود استفاده، کاهش هزینه ها و افزایش کیفیت سرویس در یک محیط ابر ترکیبی شده است [۱۴].

ون دن بوچه^{۱۱} و همکاران بر روی تصمیم برای ارسال کارها به یکی از ابزارهای عمومی در ابرهای ترکیبی تمرکز کردند تا با در نظر گرفتن کیفیت سرویس، هزینه های کاربران را نیز کاهش دهند. در این مطالعه، آن ها مجموعه ای از هیروستیک ها^{۱۲} را برای تصمیم گیری در مورد ابر عمومی یا خصوصی مناسب برای اجرای کاری ارائه کردند. این هیروستیک شامل هزینه های انتقال داده، هزینه های محاسباتی و زمان تخمین زده شده برای انتقال نیز بود. نتایج مطالعه آن ها نشان داد که تمامی این فاکتورها در اجرای دقیق و کاهش هزینه های کاربران موثر است [۱۵].

روبن ون دن بوچه و همکارانش به ارائه روشی برای قطعه بندی کردن کارها و تخصیص هر قطعه به یکی از ابرهای در دسترس در ابرهای ترکیبی پرداختند. آن ها با این روش مهلت های زمانی مورد نظر برای اجرای کارها را به راحتی بر آورده کردند. اجرای سریع برنامه ها نیز منجر به رضایت کاربران و کاهش هزینه ی آن ها خواهد گردید. به این منظور آن ها الگوریتم هایی را ارائه دادند تا بتوانند تمامی این فاکتورها را اجرایی کنند. آن ها پس از ارزیابی الگوریتم خود با داده های واقعی دریافتند که الگوریتم آن ها کارایی بسیار مناسبی را ارائه می کند [۱۶].

مالاوسکی^{۱۳} و همکارانش مدلی برای کاهش هزینه ها برای زمان بندی کارها بر روی یک IaaS ارائه کرد. آن ها مجموعه ای از چند IaaS را در نظر گرفتند که در هر یک تعداد محدودی از نمونه ها دربر وجود داشت که دارای هزینه های ساعتی مختلفی بودند. آن ها پس از دسته بندی کارهای خود به چند کلاس با سطوح مختلف شباهت، مدل خود را با استفاده از AMPL و CMPL مدل سازی کردند. سپس نتایج کار خود به همراه پنج مارک های^{۱۴} انجام شده بر روی آن را در دامنه های مختلفی ارائه کردند [۱۷].

در مرجع [۱۸] مسئله ی زمان بندی وظایف و تشخیص منابع در محاسبات ابری الگوریتمی ارائه شده است. هدف آن ها ارائه مدلی برای تشخیص منابع بر مبنای وظایف در یک محیط محاسبات ابری بود. روش آن ها بر مبنای تحلیل ریاضیاتی وظایف بود. در مدل پیشنهادی آن ها، منابع محاسباتی می توانند بر اساس رتبه های در نظر گرفته برای منابع تخصیص پیدا می کردند. آن ها برای اعتبارسنجی روش خود دو مثال را مورد بررسی قرار دادند که در آن روش پیشنهادی آن ها پاسخ مناسبی برای این مسئله داشت [۱۹].

⁸ Pandy

⁹ Ficco

¹⁰ Toosi

¹¹ Van den Bossche

¹² Heuristic

¹³ Malawski

¹⁴ Benchmark

در مرجع [۲۰] روشی برای تخصیص پویای منابع در پردازش ابری ارائه کردند. در مدل آن‌ها، تخصیص منابع در محیط‌های ابری در نظر گرفتند. از آنجایی که درخواست منابع و تامین منابع می‌تواند به صورت پویا و به صورت غیرقطعی هستند، آن‌ها مکانیسم مذاکره‌ای توزیعی ارائه کردند که عوامل در آن‌ها هم در مورد هزینه‌ها و هم در مورد جریمه‌ها مذاکره می‌کردند. آن‌ها در نهایت روش خود را به صورت علمی با ارائه سناریوها و کارهایی پیاده‌سازی کردند تا کارایی مدل خود را در محیط پردازش ابری انعطاف‌پذیر^{۱۵} آمازون مورد بررسی قرار دهد و در نتیجه مدل آن‌ها نشان داد که از لحاظ کاهش هزینه‌ها کارایی بهتری را خواهد داشت.

روشی برای تخصیص تطابقی منابع برای کارهای قابل پیش‌تخلیه^{۱۶} در سیستم‌های ابری در مرجع [۲۰] ارائه گردیده است. آن‌ها با اعمال پردازش موازی در محاسبات ابری کارایی را بهبود دادند. براساس مطالعه‌ی آن‌ها، پیاده‌سازی مکانیسمی برای تخصیص منابع و زمان‌بندی اجرای وظایف امری ضروری است. در این مقاله آن‌ها مکانیسم تخصیص منبع را با قابلیت پیش‌تخلیه کارها ارائه کردند. آن‌ها یک الگوریتم تطابقی تخصیص منابع برای سیستم‌های پردازش ابری ارائه کردند. الگوریتم آن‌ها براساس کارهای رسیده، منابع به صورت تطبیقی اختصاص می‌داد. نتایج ارزیابی آن‌ها نشان داد که الگوریتم پیشنهادی آن‌ها پاسخ بسیار مناسبی را در مقایسه با روش‌های سنتی داشته است [۲۰].

در مرجع [۲۱] یک الگوریتم جدید دو لایه ناهمگن DE در محیط توزیع‌شده محاسبات ابری ارائه شده است. لایه اول شامل، جمعیت‌های مختلف با پارامترهای مختلف یا اپراتورهای که به طور همزمان اجرا می‌شوند. لایه دوم مجموعه‌ای از ماشین‌های مجازی ابری که به صورت موازی اجرا می‌شوند، برای ارزیابی سازگاری جمعیت مربوطه که منجر به کاهش هزینه محاسباتی که توسط ابر ارائه شده است. نتایج آزمایش بر روی مجموعه‌ای از مشکلات نشان داده است که الگوریتم پیشنهادی منجر به بهبود عملکرد و کاهش زمان محاسبات شده است. افزایش سرعت به خصوص در مسائل گران قابل توجه است [۲۱].

در مرجع [۲۲] تمرکز بر روی یک زمان‌بندی برای الگوریتم‌های ابرهیریدی که سعی در بهینه‌سازی زمان اجرا و هزینه را دارد. برای پیدا کردن زمان‌بندی مطلوب از الگوریتم تکاملی چندهدفه استفاده می‌شود. زمان‌بندی که به طور گسترده در پیاده‌سازی کردن زمان-اجرا و یا هزینه دارد، اما نه هر دو به طور همزمان. روش ارائه شده در مقایسه با روش‌های بهینه‌سازی زمان‌بندی کارایی بهتری دارد [۲۲]. مرجع [۲۳] نشان می‌دهد که زمان‌بندی گردش علمی^{۱۷} می‌تواند مشکلات زمان‌بندی وظایف و تخصیص فایل داده‌ها را بهبود دهد. بنابراین یک گردش کار^{۱۸} جدید را که در آن گره گردش کار نشان‌دهنده وظایف یا فایل‌های داده و تعریف زمان‌بندی وظیفه می‌باشد را نشان می‌دهد. مسئله زمان‌بندی وظایف و تخصیص داده (TaSDAP) با توجه به این مدل جدید است. این مسئله را با یک فرمول برنامه‌نویسی انتگرال مورد بررسی قرار می‌دهد. علاوه بر این، یک الگوریتم تکاملی ترکیبی برای حل آن، به نام HES-TaSDAP نیز معرفی می‌کند. در تمام آزمایشات HES-TaSDAP عملکرد بهتری نسبت به روش‌های کلاسیک دیگر دارد [۲۳].

در مرجع [۲۴] مشکل مدل زمان‌بندی گردش کار که بهینه‌سازی makespan و هزینه می‌باشد را به عنوان یک مسئله بهینه‌سازی چندهدفه (MOP)^{۱۹} برای محیط‌های ابر مورد بررسی قرار داده است. یک بهینه‌سازی تکاملی (EMO) مبتنی بر الگوریتم برای حل مشکل زمان‌بندی گردش کار بر روی یک زیرساخت به عنوان سرویس (IaaS) پیشنهاد کرده است. آزمایش‌ها نشان می‌دهد که برنامه‌های تولید شده توسط الگوریتم تکاملی ثبات و پایداری بیشتری دارد و از نظر منابع و هزینه‌ها مطلوب‌تر می‌باشد [۲۴].

در مرجع [۲۵] به انواع دیگری از الگوریتم‌های ژنتیک جهت تخصیص منابع به وظایف اشاره می‌کند. در نهایت به نتیجه رسیده که اولویت‌های متعدد الگوریتم صف ژنتیک یک روش کارآمد برای زمان‌بندی وظایف است. maskspan و به طور کلی هزینه‌ها را

¹⁵ Elastic

¹⁶ Pre-emptible

¹⁷ scientific workflows

¹⁸ workflow

¹⁹ Multi-objective Optimization Problem

کاهش می‌دهد. این الگوریتم زمان‌بندی قادر به اختصاص منابع براساس اولویت و تجزیه آن به کارهای دیگر برای به حداقل رساندن maskspan می‌باشد [۲۵].

۳- روش پیشنهادی

زمان‌بندی وظایف، یک مساله NP-Complete است زیرا یافتن پاسخ این دسته مسائل دارای پیچیدگی نمایی است. یک راه برای یافتن پاسخ‌های این مسائل، استفاده از روش‌های فراابتکاری و الگوریتم‌های بهینه‌سازی تکاملی می‌باشد. روش‌های بهینه‌سازی تکاملی، روش‌هایی هستند که می‌توانند جوابی بهینه یا نزدیک به بهینه در یک زمان معقول برای یک مساله ارائه کنند. یک زمان‌بند وظیفه مناسب، باید استراتژی زمان‌بندی خود را بر اساس تغییرات محیطی و نوع وظایف، اصلاح و سازگار نماید. هدف اصلی این مقاله ارائه مدلی چندهدفه برای زمان‌بندی جریان کاری در محیط ابر است که هم بتواند حداقل کیفیت خدمت مورد نظر کاربران را بهینه کند و هم سودمندی عرضه‌کنندگان خدمات ابری را افزایش دهد. همچنین، به دلیل چندهدفه بودن ماهیت مسئله و وجود تعارض بین اهداف، یک رویکرد بهینه‌سازی چندهدفه با استفاده از الگوریتم گرگ خاکستری پیشنهاد شده است که نسبت به الگوریتم‌های بهینه‌سازی مرسوم مبتنی بر روش‌های فراابتکاری عملکرد بهتری داشته باشد. این الگوریتم چندهدفه پیشنهادی قصد دارد تا مصالحه بهینه‌ای را در بین سه هدف متعارض کمینه کردن زمان اجرا جریان کاری، کمینه کردن هزینه اجرا و بیشینه کردن بازدهی منابع بدست می‌آورد.

۴- الگوریتم بهینه‌سازی گرگ خاکستری

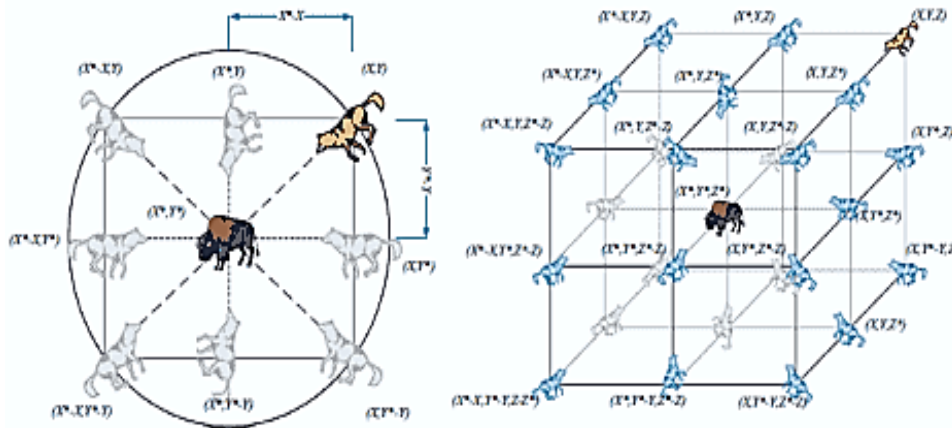
الگوریتم گرگ خاکستری (GWO) ^{۲۰} یک الگوریتم فراابتکاری الهام گرفته از طبیعت است که رفتار گرگ‌های خاکستری و سلسله مراتب رهبری و روش شکار آنها را تقلید می‌کند. در این الگوریتم، ۴ نوع گرگ خاکستری بنام گرگ‌های آلفا، بتا، دلتا و امگا برای شبیه‌سازی رفتار سلسله‌مراتبی رهبری در گله گرگ‌های خاکستری استفاده شده‌است. علاوه بر این الگوریتم شامل سه گام اصلی شکار گرگ‌های خاکستری می‌باشد که عبارتند از جستجو برای طعمه، محاصره طعمه و درنهایت حمله به طعمه.

گرگ‌های آلفا رهبران گروه می‌باشند که می‌تواند از نوع مونث یا مذکر باشد و غالباً تصمیمات کلی همانند شکار کردن، زمان و مکان خواب و بیدار شدن گله را رهبر گروه انجام می‌دهد و سایر اعضای گروه تصمیمات گرگ‌های آلفا را انجام می‌دهند. در سطح دوم رهبری و بعد از گرگ‌های آلفا، گرگ‌های بتا قرار می‌گیرند که به گرگ‌های آلفا در تصمیم‌گیری و سایر فعالیت‌ها کمک می‌کنند. گرگ بتا بهترین گزینه برای جایگزینی گرگ آلفا می‌باشد. گرگ بتا بعنوان مشاور گرگ آلفا عمل می‌کند. گرگ‌های دلتا بعد از گرگ‌های بتا قرار می‌گیرند و بالاتر از گرگ‌های امگا می‌باشند و وظایفی مانند پیشاهنگی (مراقبت از مرزهای قلمرو)، تضمین ایمنی گروه و شکار و .. را بر عهده دارند. گرگ‌های دلتا به گرگ‌های آلفا و بتا کمک می‌کنند و در نهایت مسئول نگهداری و مراقبت از گرگ‌های ضعیف و بیمار در گله می‌باشند. گرگ‌های امگا پایین‌ترین نقش را در دسته گرگ‌ها بازی می‌کنند و در تصمیم‌گیری‌ها مشارکتی ندارند و نقش قربانی را بازی می‌کنند. گرگ‌های امگا بعد از همه گرگ‌ها مجاز به غذا خوردن می‌باشند و کارهایی همانند نگهداری بچه و .. را انجام می‌دهند. یکی از رفتارهای اجتماعی جالب توجه گرگ‌های خاکستری رفتار شکار می‌باشد.

ساختار سلسله‌مراتبی و رفتار اجتماعی گرگ‌ها در حین فرایند شکار به صورت ریاضی مدل شده و برای طراحی الگوریتمی برای بهینه‌سازی مورد استفاده قرار گرفته است. گرگ‌های خاکستری توانایی تخمین موقعیت شکار را دارند. در جستجوی اولیه هیچ ایده‌ای در مورد موقعیت شکار نداریم. فرض می‌شود گرگ‌های alpha, beta, delta دانش اول بهتری در خصوص موقعیت

²⁰ Grey Wolf Optimizer

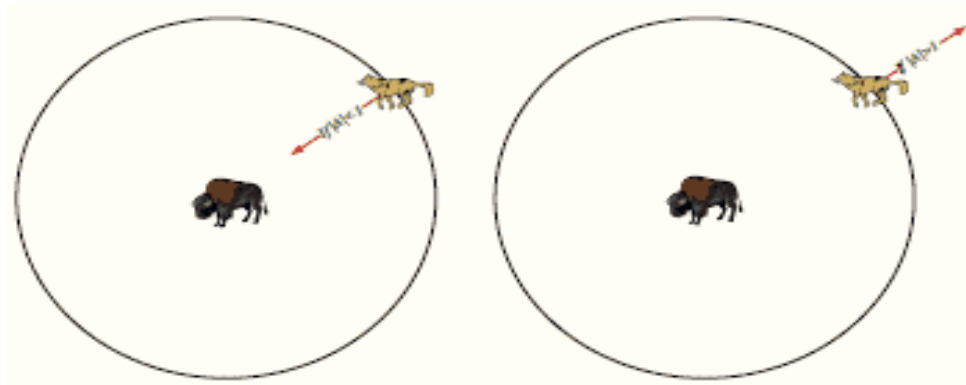
شکار (نقطه بهینه جواب) دارند. در واقع گرگ‌های alpha, beta, delta موقعیت شکار را تخمین می‌زنند اما بقیه گرگ‌ها موقعیت خود را بطور تصادفی حول شکار برورسانی می‌کنند. در شکل زیر نمایش دو بعدی و سه بعدی بردارهای موقعیت گرگ‌ها و موقعیت‌های ممکن برای حرکت بعدی نشان داده شده است.



شکل ۱- بردارهای موقعیت گرگ‌ها و موقعیت‌های ممکن برای حرکت بعدی

وقتی شکار توسط گرگ‌ها احاطه شده و از حرکت بایستند حمله به رهبری گرگ alpha شروع می‌شود. مدل کردن این فرایند با استفاده از کاهش بردار a انجام می‌شود. از آنجا که A برداری تصادفی است، با کاهش a ، بردار ضرایب A هم کاهش می‌یابد. اگر $|A| < 1$ باشد، گرگ alpha به شکار (و بقیه گرگ‌ها) نزدیک می‌شود و اگر $|A| > 1$ گرگ از شکار (و بقیه گرگ‌ها) دور خواهد شد. الگوریتم GWO الزام دارد که تمام گرگ‌ها موقعیت خود را برحسب موقعیت گرگ‌های alpha, beta, delta برورسانی کنند [۲۶].

فرایند جستجو پروسه‌ای دقیقاً عکس فرایند حمله دارد: در هنگام جستجو گرگ‌ها از یکدیگر دور می‌شوند تا شکار را ردیابی کنند ($|A| > 1$) در حالی که پس از ردیابی شکار، گرگ‌ها در فاز حمله به یکدیگر نزدیک می‌شوند ($|A| < 1$). به این پروسه واگرایی در جستجو - همگرایی در حمله می‌گویند [۲۶].



Exploration: $|A| > 1$
Exploitation: $|A| < 1$

نقش بردار C : بردار C به عنوان موانع موجود در طبیعت که نزدیک شدن گرگ‌ها به شکار را کند می‌کنند در نظر گرفته می‌شود. بردار C به شکار وزن داده و آنرا برای گرگ‌ها غیر قابل دستیابی تر می‌کند. پس به طور خلاصه می‌توان الگوریتم را به ترتیب زیر جمع بندی نمود:

۱. برازندگی کلیه جواب‌ها محاسبه شده و سه جواب برتر به عنوان alpha, beta, delta تا پایان الگوریتم انتخاب می‌شوند.
۲. در هر تکرار سه جواب برتر (گرگ‌های alpha, beta, delta) قابلیت تخمین موقعیت شکار را داشته دارند.

۳. در هر تکرار بعد از تعیین موقعیت گرگ‌های alpha, beta, delta، بروزرسانی موقعیت بقیه جواب‌ها با تبعیت از آن‌ها انجام می‌شود.

۴. در هر تکرار بردار a و به تبع آن A و C بر اساس فرمول‌های ۱ و ۲ بروزرسانی می‌شوند.

۵. در پایان تکرارها موقعیت گرگ alpha به عنوان نقطه بهینه معرفی می‌شود.

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (1)$$

$$\vec{C} = 2 \vec{r}_2 \quad (2)$$

$$\vec{D}_a = |\vec{C}_1 \cdot \vec{X}_a \cdot \vec{X}| \quad \vec{D}_b = |\vec{C}_2 \cdot \vec{X}_b \cdot \vec{X}|, \quad \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta \cdot \vec{X}|, \quad (3)$$

$$\vec{X}_2 = \vec{X}_b \cdot \vec{A}_2 \cdot (\vec{D}_b), \quad \vec{X}_3 = \vec{X}_\delta \cdot \vec{A}_3 \cdot (\vec{D}_\delta), \quad \vec{X}_1 = \vec{X}_a \cdot \vec{A}_1 \cdot (\vec{D}_a) \quad (4)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (5)$$

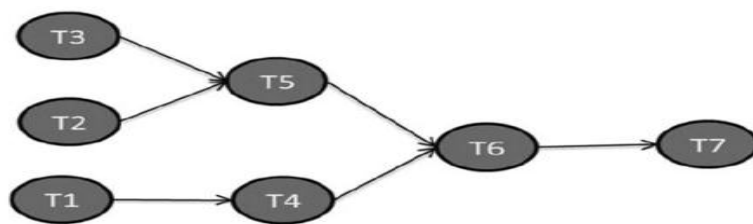
۴-۱- مدل سیستم

در مدل ارائه شده تمام وظایف رابطه‌های وابستگی و محدود کننده دارند. هر وظیفه، وظیفه والد خود را دارد. به عبارت دیگر بصورت تصادفی برخی از وظایف بصورت والد هستند و برخی دیگر زیر وظیفه. ماتریس وابستگی زیر برای نمایش رابطه بین وظایف استفاده می‌شود. مولفه $A_{ij}=0$ به این معنی است که وظایف T_i و T_j رابطه متقابلی ندارند و $A_{ij}=1$ رابطه متقابل وظایف T_i و T_j را نشان می‌دهد.

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(۶) ماتریس وابستگی

بر اساس ماتریس وابستگی، مدل جریان کاری مانند شکل ۲ ترسیم می‌گردد. هر وظیفه یک درجه ورودی و یک درجه خروجی دارد. بعد از اجرای هر وظیفه، درجه ورودی یک زیروظیفه T_i یک واحد کسر می‌شود. وظیفه T_j زمانی اجرا می‌شود که درجه ورودی اش صفر شود.

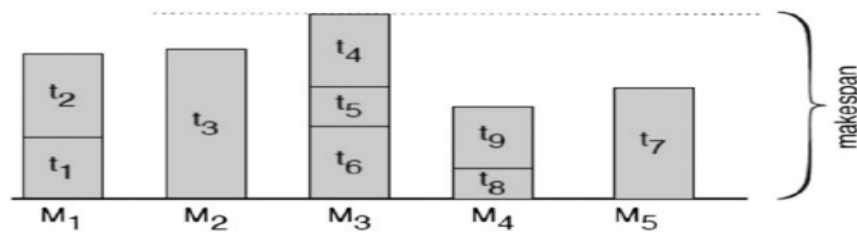


شکل ۲- مدل جریان کاری [۲۶]

اگر $P_{i,j}$ زمان اجرای وظیفه i باشد، آنگاه $l_i = \sum P_{i,j}$ زمان تکمیل وظایف در ماشین مجازی i خواهد بود و در نتیجه ما $l_{max} = \max L_i$ را به صورت زمان تکمیل اجرا در تمام ماشین‌های مجازی می‌نامیم.

در شکل ۳ یک زمان‌بندی نمونه برای ۹ کار و ۵ ماشین مجازی ارائه شده است. همان‌گونه که در شکل نشان داده شده است، برخی از وظایف موازی با سایر وظایف بر روی چند ماشین مجازی اجرا می‌گردد و برخی دیگر به صورت ترتیبی انجام می‌شود که بصورت

ترتیبی می باشد [۲۶].
 $t_7 || (t_9; t_8) || (t_4; t_5; t_6) || t_3 || (t_2; t_1)$ نشان داده شده است. علامت || نشان دهنده اجرای موازی و کاما نشان دهنده اجرای



شکل ۳- یک نمونه زمانبندی جریان کاری با استفاده از ۵ ماشین مجازی [۲۶]

زمان اجرای یک وظیفه t بزرگترین زمان طی شده برای دریافت داده ورودی (نشان داده شده با T_{wait-t}) به علاوه زمان پردازش وظیفه ای T_{exe-t} می باشد که در فرمول های زیر نشان داده شده است. $maxinput(t)$ بزرگترین داده ای است که وظیفه t از وظیفه فرزند خود دریافت می کند و BW ^{۲۱} نشانگر میانگین پهنای باند ماشین مجازی بر حسب مگابایت بر ثانیه می باشد. MI اندازه وظیفه بر حسب میلیون دستورالعمل و $MIPS$ ^{۲۲} معادل میلیون دستورالعمل در ثانیه می باشد. زمان تکمیل وظیفه t به صورت $T_{wait-t} + T_{exe-t}$ می باشد و زمان تکمیل اجرا در ماشین مجازی VM_i به صورت فرمول (۱۰) و با اجتماع زمان اجراهای همه وظایف نشان داده می شود. بزرگترین زمان تکمیل وظایف ماشین های مجازی به عنوان زمان $Makespan$ می باشد.

$$T_{wait-t} = \frac{maxinput(t)}{BW}, \quad (7) \text{ زمان دریافت داده ورودی}$$

$$T_{exe-t} = \frac{MOI(t)}{MIPS(VM_i)}, \quad (8) \text{ زمان پردازش وظیفه}$$

$$T_{total-it} = \sum_{t \in T} (T_{exe-t} + T_{wait-t}), \quad (9) \text{ زمان تکمیل یک فرایند}$$

$$makespan = \max(T_{total-1t}, T_{total-2t}, \dots, T_{total-mt}). \quad (10) \text{ زمان اجرای کل وظایف}$$

در رایانش ابری کاربران باید براساس مدل پرداخت به ازای استفاده^{۲۳}، هزینه سرویس را پردازند. بدین معنی که مشتریان بر حسب مدت زمانی که اجرای سرویس درخواستی آنها طول بکشد هزینه پرداخت می کنند. هزینه یک وظیفه t شامل هزینه پردازش $C_{Processing-t}$ ، هزینه ذخیره سازی $C_{Storage-t}$ و هزینه انتقال داده $C_{Transfer-t}$ می باشد متغیرهای $C_{Processing-t}$ ، $C_{Storage-t}$ و $C_{Transfer-t}$ به ترتیب نشانگر هزینه به ازای اجرای یک میلیون دستورالعمل برای پردازش، هزینه ذخیره سازی به ازای هر ثانیه و هزینه انتقال داده بین ماشین های مجازی بر حسب مگابایت بر ثانیه می باشد مجموع هزینه هر ماشین مجازی شامل مجموع هزینه تمام وظایف اجرا شده بر روی ماشین مجازی می باشد و هزینه کل شامل هزینه تمام ماشین های مجازی است که بصورت زیر محاسبه می گردد [۲۷].

$$C_{Processing-t} = T_{exe-t} * CostPerProcess, \quad (11) \text{ هزینه پردازش}$$

$$C_{Storage-t} = (T_{exe-t} + T_{wait-t}) * CostPerStorage \quad (12) \text{ هزینه ذخیره سازی}$$

$$C_{Transfer-t} = \frac{\sum t. Output}{BM} * CostPerTransfer, \quad (13) \text{ هزینه انتقال داده}$$

$$C_{total} = \sum C_{total-i}. \quad (14) \text{ هزینه کل جریان وظایف}$$

21 - bandwidth

22 - million instruction per second

23- pay-per-use

نرخ گذردهی یک ماشین مجازی برابر تعداد دستورالعمل‌های پردازش شده توسط ماشین مجازی می‌باشد بنابراین مجموع گذردهی، متوسط گذردهی همه ماشین‌های مجازی می‌باشد که M تعداد ماشین‌های مجازی می‌باشد [۲۶].

$$\text{ThroughPut}_{it} = \frac{\sum_{t \in T} MI(t)}{T_{total-i}} \quad (15) \text{ گذردهی یک ماشین مجازی}$$

$$\text{ThroughPut}_{total} = \frac{\sum_{i=1}^m \text{ThroughPut}_i}{m} \quad (16) \text{ متوسط گذردهی ماشین‌های مجازی}$$

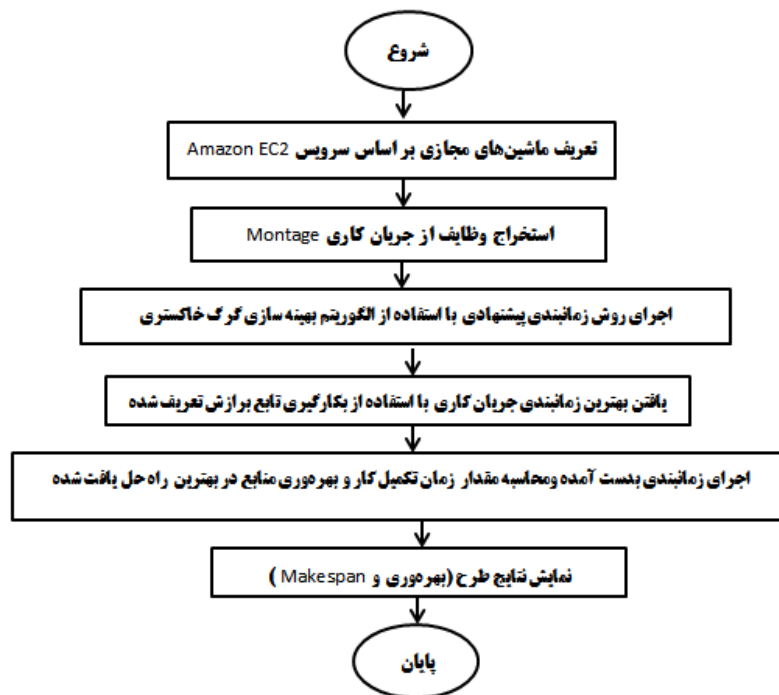
برای لحاظ نمودن بهره‌وری منابع، از مدت زمان بیکاری ماشین‌های مجازی نیز استفاده شده است. با توجه مطالب گفته شده بالا تابع برازش چند هدفه پیشنهادی بصورت زیر خواهد بود:

$$f(i) = \frac{1}{T_{total}(i) + T_{trans}(i) + T_{idle}(i)} \quad (17) \text{ تابع برازش پیشنهادی}$$

T_{total} برابر مجموع زمان پردازش و انتظار برای انجام یک وظیفه، T_{trans} برابر زمان انتقال داده بین سایت‌های مختلف و T_{idle} مجموع زمان بیکاری ماشین‌های مجازی می‌باشد.

۲-۴- فلوچارت روش پیشنهادی

در شکل ۴ فلوچارت مراحل اجرای زمان‌بندی بر اساس طرح پیشنهادی برای جریان کاری نشان داده شده است.



شکل ۴- فلوچارت روش زمان‌بندی جریان کار در طرح پیشنهادی

۵- شبیه سازی و ارائه نتایج

برای شبیه سازی طرح پیشنهادی از محیط کلودسیم و نصب افزونه workflowsim استفاده شده است.

شبیه سازی شامل تعداد ۱۰۰ وظیفه است که در دفعات تکرار آزمایش باید بر روی ماشین های مجازی مختلف زمانبندی شوند بطوری که زمان تکمیل کار و هزینه سرویس بهبود یابد. مشخصه های سیستم برای پیاده سازی الگوریتم پیشنهادی در جدول ۱ و پارامترهای شبیه سازی در جدول ۲ بیان شده است.

جدول ۱- مشخصات سیستم جهت پیاده سازی الگوریتم پیشنهادی

مقدار	مشخصات
Core i7	CPU
8 GB	RAM
Windows 7	OS

جدول ۲- پارامترهای شبیه سازی

مقدار	پارامتر شبیه سازی
۲۵،۵۰،۱۰۰	تعداد وظایف
۵	تعداد ماشین های مجازی
طبق جریان کاری Montage	مقادیر زمان محاسباتی وظایف
طبق مشخصات سرویس Amazon Ec2	هزینه اجرا در واحد زمان

جدول ۳- پارامترهای زمانبندی با روش GWO

مقدار	پارامتر شبیه سازی
MaxIt=۱۰۰	حداکثر دفعات تکرار
۲۰	تعداد گرگ ها (اندازه جمعیت)
به اندازه تعداد وظایف	طول یک ذره

برای بررسی بهینگی هزینه و زمان تکمیل Makespan در طرح پیشنهادی، عملکرد خود را ابتدا، از نظر زمان تکمیل وظایف در زمانبندی انجام شده با طرح های پایه مقایسه نمودیم. برای این کار شبیه سازی را در تعداد دفعات بالا (۳۰۰ بار) اجرا کردیم که مقادیر حداکثر، حداقل، بیشترین تکرار و میانگین برای هر اجرا با تعداد وظایف مختلف به صورت جدول ۴ می باشد. نتایج تخصیص مرحله به مرحله ماشین های مجازی در یکی از این اجراها بصورت جدول ۵ می باشد. در جدول ۴ مشاهده می شود که بهترین مقدار زمان تکمیل کار برای ۱۰۰ وظیفه برابر ۲۰۷.۷۹ و برای تعداد ۵۰ و ۲۵ وظیفه، به ترتیب برابر ۱۶۲/۳۴ و ۱۰۷/۵۵ می باشد.

جدول ۴- مقادیر Makespan در ۳۰۰ بار اجرای طرح با روش GWO برای تعداد وظایف مختلف

تعداد وظایف	MAX	MIN	MOD	AVERAGE
۲۵	115.85	107.55	108.62	109.23
۵۰	171.91	162.34	162.34	164.12
۱۰۰	221.32	207.79	207.79	209.35

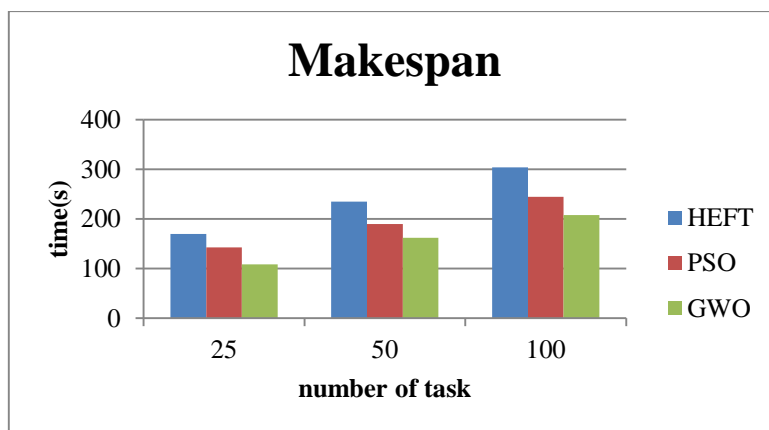
جدول ۵- نتیجه اجرای زمانبندی با روش GWO برای تعداد ۱۰۰ وظیفه

Task id	Task number	state	Data center	VM	Task time	Start time	Finish time	Depth in DAG
100	Stage-in	SUCCESS	2	0	0.19	0.1	0.29	0
15	16	SUCCESS	2	0	1.55	0.29	1.84	1
1	2	SUCCESS	2	1	10.49	0.29	10.78	1
4	5	SUCCESS	2	2	10.6	0.29	10.89	1
14	15	SUCCESS	2	3	10.6	0.29	10.89	1
6	7	SUCCESS	2	4	10.71	0.29	11	1
Task id	Task number	state	Data center	VM	Task time	Start time	Finish time	Depth in DAG

3	4	SUCCESS	2	0	10.64	1.84	12.48	1
28	29	SUCCESS	2	2	8.46	10.89	19.34	2
77	78	SUCCESS	2	4	8.52	11	19.52	2
40	41	SUCCESS	2	0	8.41	12.48	20.89	2
13	14	SUCCESS	2	1	10.69	10.78	21.46	1
2	3	SUCCESS	2	3	10.74	10.89	21.62	1
74	75	SUCCESS	2	2	8.39	19.34	27.73	2
34	35	SUCCESS	2	4	8.42	19.52	27.94	2
22	23	SUCCESS	2	0	8.46	20.89	29.35	2
39	40	SUCCESS	2	1	8.46	21.46	29.93	2
37	38	SUCCESS	2	3	8.43	21.62	30.05	2
35	36	SUCCESS	2	2	8.42	27.73	36.16	2
33	34	SUCCESS	2	4	8.49	27.94	36.43	2
68	69	SUCCESS	2	0	8.49	29.35	37.84	2
21	22	SUCCESS	2	1	8.51	29.93	38.44	2
73	74	SUCCESS	2	3	8.52	30.05	38.57	2
27	28	SUCCESS	2	2	8.54	36.16	44.69	2
29	30	SUCCESS	2	4	8.59	36.43	45.02	2
72	73	SUCCESS	2	0	8.63	37.84	46.47	2
30	31	SUCCESS	2	1	8.73	38.44	47.17	2
10	11	SUCCESS	2	3	10.84	38.57	49.41	1
5	6	SUCCESS	2	2	10.86	44.69	55.55	1
12	13	SUCCESS	2	4	10.9	45.02	55.92	1
8	9	SUCCESS	2	0	10.9	46.47	57.38	1
7	8	SUCCESS	2	1	10.92	47.17	58.09	1
9	10	SUCCESS	2	3	10.96	49.41	60.37	1
43	44	SUCCESS	2	2	8.32	55.55	63.87	2
38	39	SUCCESS	2	4	8.33	55.92	64.24	2
71	72	SUCCESS	2	0	8.34	57.38	65.72	2
55	56	SUCCESS	2	1	8.38	58.09	66.46	2
56	57	SUCCESS	2	3	8.38	60.37	68.76	2
42	43	SUCCESS	2	2	8.37	63.87	72.24	2
60	61	SUCCESS	2	4	8.39	64.24	72.64	2
32	33	SUCCESS	2	0	8.39	65.72	74.11	2
57	58	SUCCESS	2	1	8.41	66.46	74.87	2
44	45	SUCCESS	2	3	8.43	68.76	77.19	2
51	52	SUCCESS	2	2	8.44	72.24	80.68	2
31	32	SUCCESS	2	4	8.44	72.64	81.08	2
54	55	SUCCESS	2	0	8.45	74.11	82.56	2
67	68	SUCCESS	2	1	8.46	74.87	83.33	2
25	26	SUCCESS	2	3	8.46	77.19	85.65	2
65	66	SUCCESS	2	2	8.47	80.68	89.15	2
47	48	SUCCESS	2	4	8.47	81.08	89.55	2
Task id	Task number	state	Data center	VM	Task time	Start time	Finish time	Depth in DAG
58	59	SUCCESS	2	0	8.47	82.56	91.03	2
24	25	SUCCESS	2	1	8.49	83.33	91.81	2
75	76	SUCCESS	2	3	8.49	85.65	94.14	2
46	47	SUCCESS	2	2	8.5	89.15	97.64	2
61	62	SUCCESS	2	4	8.51	89.55	98.06	2
50	51	SUCCESS	2	0	8.52	91.03	99.55	2
69	70	SUCCESS	2	1	8.53	91.81	100.34	2
52	53	SUCCESS	2	3	8.53	94.14	102.67	2
26	27	SUCCESS	2	2	8.54	97.64	106.18	2
36	37	SUCCESS	2	4	8.54	98.06	106.59	2
49	50	SUCCESS	2	0	8.55	99.55	108.1	2
59	60	SUCCESS	2	1	8.58	100.34	108.92	2
66	67	SUCCESS	2	3	8.58	102.67	111.24	2
23	24	SUCCESS	2	2	8.59	106.18	114.77	2
41	42	SUCCESS	2	4	8.62	106.59	115.21	2
70	71	SUCCESS	2	0	8.62	108.1	116.72	2
11	12	SUCCESS	2	1	11.05	108.92	119.97	1
0	1	SUCCESS	2	3	11.08	111.24	122.32	1
63	64	SUCCESS	2	0	8.28	119.97	128.25	2
76	77	SUCCESS	2	1	8.39	119.97	128.36	2
48	49	SUCCESS	2	2	8.39	119.97	128.36	2

53	54	SUCCESS	2	4	8.5	119.97	128.47	2
64	65	SUCCESS	2	3	8.51	122.32	130.84	2
20	21	SUCCESS	2	0	8.36	128.25	136.61	2
62	63	SUCCESS	2	1	8.37	128.36	136.72	2
45	46	SUCCESS	2	2	8.48	128.36	136.83	2
16	17	SUCCESS	2	4	8.48	128.47	136.94	2
18	19	SUCCESS	2	3	8.46	130.84	139.3	2
17	18	SUCCESS	2	0	8.48	136.61	145.09	2
19	20	SUCCESS	2	1	8.51	136.72	145.24	2
78	79	SUCCESS	2	0	3.97	145.24	149.2	3
79	80	SUCCESS	2	0	4.27	149.2	153.48	4
85	86	SUCCESS	2	0	8.39	153.48	161.87	5
82	83	SUCCESS	2	1	8.5	153.48	161.98	5
84	85	SUCCESS	2	2	8.61	153.48	162.09	5
91	92	SUCCESS	2	3	8.61	153.48	162.09	5
93	94	SUCCESS	2	4	8.61	153.48	162.09	5
88	89	SUCCESS	2	0	8.58	161.87	170.44	5
83	84	SUCCESS	2	1	8.58	161.98	170.56	5
94	95	SUCCESS	2	2	8.58	162.09	170.67	5
90	91	SUCCESS	2	3	8.65	162.09	170.73	5
92	93	SUCCESS	2	4	8.65	162.09	170.73	5
87	88	SUCCESS	2	0	8.6	170.44	179.04	5
Task id	Task number	state	Data center	VM	Task time	Start time	Finish time	Depth in DAG
95	96	SUCCESS	2	1	8.62	170.56	179.18	5
89	90	SUCCESS	2	2	8.69	170.67	179.36	5
80	81	SUCCESS	2	3	8.74	170.73	179.47	5
86	87	SUCCESS	2	4	8.74	170.73	179.47	5
81	82	SUCCESS	2	0	8.77	179.04	187.81	5
96	97	SUCCESS	2	0	6.1	187.81	193.91	6
97	98	SUCCESS	2	0	7.68	193.91	201.59	7
98	99	SUCCESS	2	0	5.53	201.59	207.12	8
99	100	SUCCESS	2	0	0.66	207.12	207.79	9

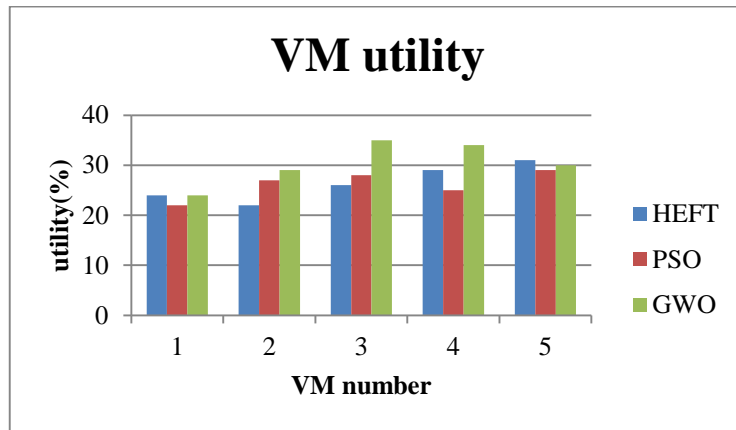
نتایج طرح پیشنهادی با دو روش زمانبندی مبتنی بر PSO [۲۵] و روش HEFT^{۲۴} [۲۶] بر حسب مقادیر میانگین زمان تکمیل کار مقایسه شده است. می‌بینیم که طرح پیشنهادی نسبت به طرح‌های پایه در زمان کمتری کلیه وظایف را بر روی منابع نگاشت و اجرا می‌کند. بنابراین روش پیشنهادی توانسته Makespan کوچکتری را نسبت به سایر طرح‌های زمانبندی ارائه دهد.



شکل ۵- عملکرد طرح پیشنهادی از نظر زمان تکمیل جریان وظایف

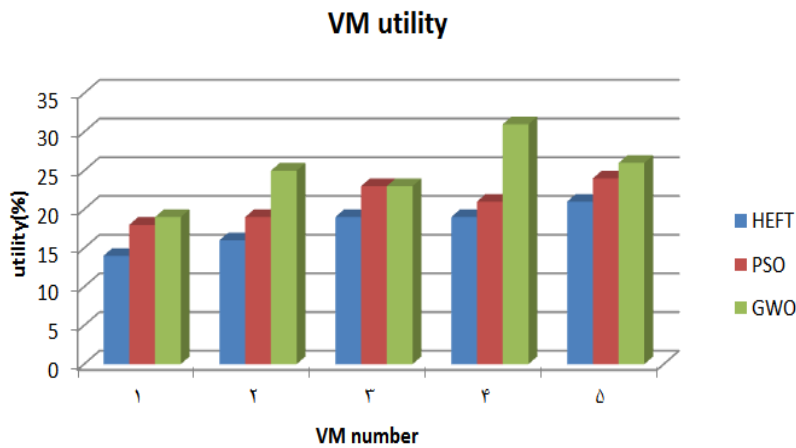
در گام بعد میزان بهره‌وری منابع را در هر طرح به تصویر کشیده شده است (تعداد ۱۰۰ وظیفه). برای محاسبه میزان بهره‌وری منابع، مجموع زمان تکمیل کار در هر منبع را بر بزرگترین زمان انجام کار در کل منابع تقسیم نمودیم که حاصل بصورت درصدی در شکل ۵ بیان شده است همانگونه که در شکل مشاهده می‌شود، بهره‌وری منابع در مقایسه با طرح‌های دیگر بهبود یافته است. علت

این ارتقاء عملکرد، چند هدفه بودن طرح پیشنهادی نسبت به تک هدفه بودن تابع برازش طرح‌های پایه (تنها بر مبنای کمینه کردن زمان تکمیل وظایف) می‌باشد.

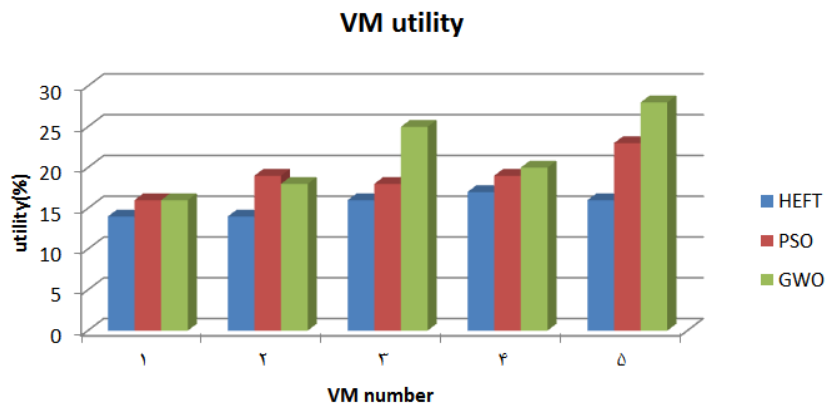


شکل ۶- مقایسه بهره‌وری منابع در هر طرح (با تعداد ۱۰۰ وظیفه)

در ادامه، همین آزمایش را با تعداد ۵۰ و ۲۵ وظیفه انجام دادیم و میزان بهره‌وری منابع را بررسی نمودیم. همانگونه که در شکل‌های ۶ و ۷ می‌بینیم طرح پیشنهادی در این حالت نیز نسبت به سایر طرح‌ها، بهره‌وری بیشتری در تخصیص منابع دارد.



شکل ۷- مقایسه بهره‌وری منابع در هر طرح (با تعداد ۵۰ وظیفه)



شکل ۸- مقایسه بهره‌وری منابع در هر طرح (با تعداد ۲۵ وظیفه)

۶- نتیجه گیری

در این مقاله، روش زمانبندی مبتنی بر الگوریتم بهینه سازی GWO را برای رسیدن به بهبود بیشتر روی زمان تکمیل کار و بهره‌وری منابع استفاده شده که در تابع برازش آن این معیارها لحاظ گردیده است. همچنین برای شبیه سازی طرح پیشنهادی از محیط کلودسیم و نصب افزونه workflowsim استفاده شده است. با توجه به نتایج بدست آمده می توان بیان نمود که، طرح پیشنهادی با توجه به بهره گیری از الگوریتم بهینه سازی در زمان کمتری کلیه وظایف بر روی منابع نگاشت و اجرا نمود. این امر به دلیل خصوصیات الگوریتم بهینه سازی در بدست آوردن بهینه ترین حالت سیستم می باشد. به عبارت دیگر Makespan بهبود یافته است. نتایج شبیه سازی نشان از آن دارد که بهره‌وری منابع در مقایسه با طرح مبتنی بر PSO و HEFT بهبود یافته است.

۷- مراجع

۱. مهدی، سارا؛ الهام‌زارع و حمیدرضاکیخا، ۱۳۹۵، حل مسئله زمان‌بندی وظایف رایانش ابری با استفاده از الگوریتم ژنتیک و الگوریتم-بهینه‌سازی زنبورعسل، سومین کنفرانس بین‌المللی علوم و مهندسی، استانبول- کشور ترکیه، موسسه‌مدیران ایده‌پرداز پایتخت‌ویرا.
2. Armbrust, Michael, et al. "A view of cloud computing." *Communications of the ACM* 53.4 (2010): 50-58.
3. B. Mondal, K. Dasgupta, and P. Dutta, "Load balancing in cloud computing using stochastic hill climbing-a soft computing approach," *Procedia Technology*, vol. 4, pp. 783-789, 2012.
4. C. Banerjee, A. Kundu, and R. Dattagupta, "Customized 3-tier Service Suite Conceptualization in Cloud Computing," *Procedia Technology*, vol. 4, pp. 561-565, 2012.
5. J. W. Rittinghouse, and J. F. Ransome, *Cloud computing: implementation, management, and security*: CRC press, 2016.
6. F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, "NIST Cloud Computing Reference Architecture". National Institute of Standards and Technology of the US Department of Commerce, 'Special Publication 500-292', 2014.
7. A. Sinha, "Cloud Computing in Libraries: Opportunities and Challenges," *Pearl: A Journal of Library and Information Science*, vol. 10, no. 2, pp. 113-118, 2016.
8. Jiang, Jianhua, Gaochao Xu, and Xiaohui Wei. "An enhanced data-aware scheduling algorithm for batch-mode dataintensive jobs on data grid." *Hybrid Information Technology*, 2006. ICHIT'06. International Conference on. Vol. 1. IEEE, 2006.
9. Cao, Junwei, et al. "Gridflow: Workflow management for grid computing." *Cluster Computing and the Grid*, 2003. Proceedings. CCGrid 2003. 3rd IEEE/ACM International Symposium on. IEEE, 2003.
10. Li, Wubin, Johan Tordsson, and Erik Elmroth. "Modeling for dynamic cloud scheduling via migration of virtual machines." *Cloud Computing Technology and Science (CloudCom)*, 2011 IEEE Third International Conference on. IEEE, 2011.
11. Wang, Xiaodan, Eric Perlman, Randal Burns, Tanu Malik, Tamas Budavári, Charles Meneveau, and Alexander Szalay. "Jaws: Job-aware workload scheduling for the exploration of turbulence simulations." In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1-11. IEEE Computer Society, 2010.
12. Zheng, Zhongni, et al. "An approach for cloud resource scheduling based on Parallel Genetic Algorithm." *Computer Research and Development (ICCRD)*, 2011 3rd International Conference on. Vol. 2. IEEE, 2015.
13. Pandey, Suraj, et al. "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments." *Advanced Information Networking and Applications (AINA)*, 2010 24th IEEE International Conference on. IEEE, 2010.
14. M. Ficco, B. Di Martino, R. Pietrantuono, and S. Russo, "Optimized task allocation on private cloud for hybrid simulation of large-scale critical systems," *Future Generation Computer Systems*, 2016.
15. A. N. Toosi, R. N. Calheiros, R. K. Thulasiram, and R. Buyya, "Resource provisioning policies to increase iaaS provider's profit in a federated cloud environment." pp. 279-287.
16. R. Van den Bossche, K. Vanmechelen, and J. Broeckhove, "Cost-efficient scheduling heuristics for deadline constrained workloads on hybrid clouds." pp. 320-327.
17. R. Van den Bossche, K. Vanmechelen, and J. Broeckhove, "Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds," *Future Generation Computer Systems*, vol. 29, no. 4, pp. 973-985, 2013.

18. M. Malawski, K. Figiela, M. Bubak, E. Deelman, and J. Nabrzyski, "Scheduling multilevel deadline-constrained scientific workflows on clouds based on cost optimization," *Scientific Programming*, vol. 2015, pp. 5, 2015.
19. D. Ergu, G. Kou, Y. Peng, Y. Shi, and Y. Shi, "The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment," *The Journal of Supercomputing*, pp. 1-14, 2013.
20. B. An, V. Lesser, D. Irwin, and M. Zink, "Automated negotiation with decommitment for dynamic resource allocation in cloud computing." pp. 981-988.
21. W.-Y. Lin, G.-Y. Lin, and H.-Y. Wei, "Dynamic auction mechanism for cloud resource allocation." pp. 591-592.
22. Z. H. Zhan, X. F. Liu, H. Zhang, Z. Yu, J. Weng, T. Gu, J. Zhang, "Cloudde: A Heterogeneous Differential Evolution Algorithm and Its Cloud Version", *IEEE Transactions on Parallel and Distributed Systems* (Volume: 28, Issue: 3, March 1 2017).
23. V. A. Leena, A. S. Ajeena Beegom, M. S. Rajasree, "Genetic Algorithm Based Bi-Objective Task Scheduling in Hybrid Cloud Platform", *International Journal of Computer Theory and Engineering*, Vol. 8, No. 1, February 2016.
24. L. Teyloa, U. Paulab, Y. Frotaa, D. Oliveiraa, L. M. A. Drummond, "A Hybrid Evolutionary Algorithm for Task Scheduling and Data Assignment of Data-intensive Scientific Workflows on Clouds", *Future Generation Computer Systems* 00 (2017) 1–27.
25. Z. Zhu, G. Zhang, M. Li, X. Liu, "Evolutionary Multi-Objective Workflow Scheduling in Cloud", *IEEE Transactions on Parallel and Distributed Systems* (Volume: 27, Issue: 5, May 1 2016).
26. B. M. Varghese, Dr. R. J. Samuel Raj, "A Survey on Variants of Genetic Algorithm for Scheduling Workflow of Tasks", *Science Technology Engineering and Management (ICONSTEM)*, Second International Conference on 30-31 March 2016.
27. Xue. Shengjun, et al. "A Heuristic Scheduling Algorithm based on PSO in the Cloud Computing Environment." *International Journal of u and e Service, Science and Technology*, Vol.9, No. 1, pp.349-362, 2016.