



استفاده از الگوریتم‌های ژنتیک و زمانبندی نوبت گردشی به منظور بهبود توازن بار در محاسبات ابری

زهرا عبادی

گروه کامپیوتر - دانشگاه آزاد اسلامی واحد بوشهر
Zahra.ebadi1398@yahoo.com

خلاصه: محاسبات ابری بر پایه شبکه‌های رایانه‌ای مانند اینترنت است که الگویی تازه برای عرضه، مصرف و تحویل خدمات محاسباتی شامل زیرساخت، نرم‌افزار، بستر و سایر منابع محاسباتی با به‌کارگیری شبکه ارائه می‌کند. هدف اصلی محاسبات ابری کاهش هزینه استفاده از منابع محاسباتی است چرا که در این روش کاربران دیگر نیاز به صرف هزینه گزاف جهت خرید و نگهداری منابع محاسباتی ندارند. با توجه به محیط پویا و انعطاف‌پذیر محاسبات ابری، روش‌های زمانبندی وظایف جهت مدیریت تخصیص سرویس و توازن بار بر روی ماشین‌های مجازی و گره‌های فیزیکی اهمیت بالایی دارد. این مسئله اغلب به عنوان یک مسئله بهینه‌سازی در نظر گرفته شده و در آن تضمین بهره‌وری و تخصیص کارآمد منابع و همچنین رضایت کاربر مورد توجه می‌باشد. در این مقاله روشی برای یافتن بهترین توالی اجرای وظایف بر مبنای حداقل‌سازی گسترده پیکربندی مجموعه وظایف ارائه شده است. برای اینکار از ترکیب الگوریتم‌های ژنتیک و زمانبندی نوبت گردشی استفاده شده است. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی بهبود توازن بار بهتری را در محیط ابر نسبت به روش‌های مشابه ارائه می‌دهد.

کلمات کلیدی: محاسبات ابری، الگوریتم ژنتیک، زمانبندی نوبت گردشی، توازن بار.

۱ - مقدمه

الگوریتم به منظور از بین بردن محدودیت‌های سیستم تعادل بار در محیط ابر به صورت پویا استفاده می‌شود. این الگوریتم ابتدا از طریق برنامه‌ریزی در ابر منابع مورد نیاز را انتخاب کرده و سپس آنها را دسته‌بندی می‌کند. در مرحله دوم، این الگوریتم منابعی را انتخاب می‌کند که به طور تصادفی طبقه بندی شده اند؛ سپس اطلاعات پویای مربوطه را برای ایجاد فیلتر بار و مرتب سازی صعودی به دست می‌آورد. این الگوریتم مشخصه‌های منابع را به دو دسته پویا و استاتیک تقسیم می‌کند، سپس آنها را مورد ارزیابی قرار می‌دهد [۱].

با توجه به افزایش روز افزون شرکت‌ها و برنامه‌های کاربردی که از محیط ابری استفاده می‌کنند، تقاضاهایی مبنی بر توسعه تکنیک‌های متعادل کننده بار جهت بهره‌مندی از منابع و کاهش تلفات منابع در محیط‌های ابری شده است. یکی از راه‌های توازن بار، برنامه‌ریزی وظایف در ابر است. به طور کلی برنامه‌ریزی وظایف بخش مهمی در سیستم‌های توزیع شده است [۲]. برنامه‌ریزی وظایف را به منابع

محاسبات ابری بر پایه شبکه‌های رایانه‌ای مانند اینترنت است که الگویی تازه برای عرضه، مصرف و تحویل خدمات محاسباتی شامل زیرساخت، نرم‌افزار، بستر و سایر منابع محاسباتی با به‌کارگیری شبکه ارائه می‌کند. محاسبات ابری از ترکیب دو کلمه محاسبات و ابر ایجاد شده است. ابر در اینجا استعاره از شبکه یا شبکه‌ای از شبکه‌های وسیع مانند اینترنت است که کاربر معمولی از پشت صحنه و آنچه در پی آن اتفاق می‌افتد اطلاع دقیقی ندارد در نمودارهای شبکه‌های رایانه‌ای نیز از شکل ابر برای نشان دادن شبکه اینترنت استفاده می‌شود. مکانیزم توازن بار در محاسبات ابری به دنبال یافتن دو پردازنده که یکی سنگین بار و دیگری سبک‌بار باشد نیست بلکه به دنبال این است که بهترین مجموعه تخصیص کارها را از میان راه‌حل‌های موجود پیدا کند.

الگوریتم زمانبندی یک بحث داغ در محاسبه به روش ابری است. از این

الگوریتم جستجو محلی سفارشی شده استفاده می‌شود. علاوه بر این روابط بین وظایف در این روش روی گراف بدون دور مستقیم (DAG) و با قابلیت اجرای هر وظیفه فقط روی یک منبع اجرا می‌شود [۹]. روش ACHBDF: این الگوریتم یک الگوریتم ترکیبی موازی کردن بار در ابر به منظور بهینه‌سازی استفاده از منابع در محاسبات ابری است. این الگوریتم ترکیبی از الگوریتم‌های مورچه کلونی و زنبور عسل با بازخورد پویا است، که کارهای ناکارآمد را زمانبندی می‌کند. نتایج شبیه سازی این الگوریتم نشان داده که این الگوریتم در زمینه زمانبندی کارها و موازی کردن بار در ابر بهتر از الگوریتم‌های مورچه کلونی و زنبور عسل کار می‌کند [۱۰].

۳ - روش تحقیق

یک سیستم زمانبندی رایانش ابری مسئول تخصیص وظایف درخواستی کاربران به منابع مورد نیاز می‌باشد. این منابع اغلب محاسباتی هستند و بر روی ماشین‌های مجازی (VM) آنلاین قرار دارند. استراتژی‌های مختلفی برای زمانبندی وظایف کاربران با اهدافی نظیر توازن بار، حداقل‌سازی زمان اتمام وظایف و غیره وجود دارد. از جمله این استراتژی‌ها می‌توان به زمانبندی اجرا به ترتیب ورود (FCFS)، زمانبندی گردشی وزن‌دار (WRR)، زمانبندی کوتاه‌ترین کار بعدی (SJN)، زمانبندی اولویت (Priority)، زمانبندی کوتاه‌ترین زمان باقیمانده (SRT)، زمانبندی اولین ورودی اولین خروجی (FIFO)، زمانبندی صف‌های چند سطحی (MLQ)، زمانبندی نوبت گردشی (RR) اشاره کرد. هر یک از این استراتژی‌های زمانبندی وظایف می‌تواند وضعیت‌های متفاوتی از توازن بار را روی ماشین‌های مجازی ایجاد کند که در نهایت ممکن است باعث ایجاد مشکلاتی در سیستم زمانبندی ابر شود. به همین دلیل در این مقاله از یک الگوریتم ژنتیک برای بهبود این استراتژی‌ها استفاده می‌شود. به طور خاص در اینجا بهبود استراتژی زمانبندی نوبت گردشی مورد توجه می‌باشد.

در این تحقیق M گره فیزیکی به صورت $PN = \{pn_1, pn_2, \dots, pn_M\}$ در محیط ابر وجود دارد، بطوریکه در هر گره نظیر pn_i تعداد w_i ماشین مجازی به صورت $VM_i = \{vm_1, vm_2, \dots, vm_{w_i}\}$ در دستری است. هدف یک سیستم زمانبندی در رایانش ابری در اینجا تخصیص N وظیفه درخواستی کاربران به صورت $T = \{t_1, t_2, \dots, t_N\}$ به هر گره فیزیکی می‌باشد به طوریکه حداکثر بهره‌وری حاصل شود. ورودی سیستم زمانبندی پیشنهادی شامل پارامترهای مربوط به محیط ابر و ویژگی‌های درخواست‌های کاربران می‌باشد. در خواست‌های کاربران، وظایفی هستند که باید توسط سیستم زمانبندی ابر به یک منبع از گره‌های فیزیکی تخصیص داده شود. در این مقاله بهترین توالی اجرای وظایف را روی گره‌های فیزیکی براساس الگوریتم ژنتیک بهینه‌سازی می‌کنیم. در اینجا برای اجرای وظایف در سیستم ابر از زمانبندی نوبت گردشی بهره می‌گیریم.

مناسب جهت اجرا اختصاص می‌دهد. مهمترین قسمت محیط محاسبات ابری، تعادل بار است. طرح متعادل کننده بار کارآمد در محیط ابری با استفاده از منابع کاربر بر اساس تقاضای پرداخت هزینه به کاربر می‌گوید که استفاده از کدام یک از منابع مؤثرتر است. تعادل بار حتی می‌تواند با تعیین معیارهای برنامه ریزی مناسب، اولویت‌بندی کاربران را پشتیبانی کند [۳]. رویکردهای فراشناختی برای استفاده از راه حل‌های بهینه یا نزدیک به مطلوب برای مسئله برنامه‌ریزی، به طور گسترده ای مورد استفاده قرار می‌گیرند.

افزایش تقاضاها برای استفاده از منابع رایانشی موجب پیدایش انواع سیستم‌های موازی و توزیع شده گردیده است. استفاده صحیح از این منابع، باعث بهبود معیارهایی در سیستم نظیر توان عملیاتی، زمان اجرا، بهره‌وری و غیره می‌گردد [۴]. توازن بار که یکی از مهمترین چالش‌ها در رایانش ابری است. در این مسئله تلاش می‌شود تا با متعادل‌سازی بارهای کاری بر روی واحدهای پردازشی، عملکرد کلی سیستم بهبود یابد. هر اندازه یک الگوریتم توازن بار قابلیت توزیع مناسبتری از منابع را روی واحدهای پردازشی داشته باشد، در نهایت سیستم رایانش ابری عملکرد بهتری خواهد داشت [۵]. این مسئله را میتوان به صورت یک مسئله بهینه‌سازی در نظر گرفت که در این تحقیق نیز از یک استراتژی ترکیبی بر پایه الگوریتم ژنتیک و زمانبندی نوبت گردشی برای بهینه‌سازی توازن بار در رایانش ابری استفاده می‌شود.

۲ - پیشینه تحقیق

تحقیقات زیادی پیرامون مسئله توازن بار در محاسبات ابری انجام شده است که اغلب مبتنی بر الگوریتم‌های بهینه‌سازی هستند. در این بخش تعدادی از این روش‌ها را مورد بررسی قرار می‌دهیم.

روش PSO: در این روش از الگوریتم PSO برای توازن بار و بهبود هزینه زمانبندی در IaaS استفاده می‌شود. محدودیت‌های در نظر گرفته شده در این تحقیق شامل اجرای جریان وظایف در یک زمان مشخص و ریسک زمانبندی در بهینه‌سازی است [۶].

روش HEFT-B: در این روش از الگوریتم ژنتیک برای زمانبندی وظایف بر روی سیستم‌های ابر محاسباتی ناهمگن استفاده شده که دارای یکسری صف اولویت‌دار است. الگوریتم ژنتیک چند بعدی (MPQGA) ارائه شده قابلیت تعیین اولویت هر وظیفه را براساس رویکرد اولین زمان پایان (EFT) دارد [۷].

روش ACO-PSO: در این روش از ترکیب دو PSO و ACO بهینه‌سازی زمانبندی وظایف و همچنین توازن بار استفاده می‌شود. این روش زمانبندی را برای جریان کار علمی در ابر IaaS انجام می‌دهد و همزمان از تکنیک‌های محلی و سراسری برای جستجوی بهترین توالی اجرای کارها بهره می‌گیرد [۸].

روش PSO-Hill: در این روش از ترکیب بهینه‌سازی PSO و الگوریتم تپه‌نوردی استفاده شده تا بهترین زمانبندی وظایف را در محیط ابر فراهم کند. در این روش همچنین برای افزایش سرعت همگرایی از یک

به طور کلی، اگر $TT(pn_i)$ مجموع زمان اجرای وظایف تخصیص داده شده به گره فیزیکی i -ام باشد، میانگین کل زمان اجرا وظایف در همه گره‌های فیزیکی به صورت رابطه (۳) محاسبه می‌شود.

$$f_3 = TT(PN) = \frac{1}{M} \sum_{i=1}^M TT(pn_i) \quad (3)$$

زمان تکمیل یک وظیفه با توجه به مجموع زمان درخواست و مدت زمان پردازش محاسبه می‌شود. زمان اتمام وظایف با توجه به زمان اتمام آخرین (بزرگترین) وظیفه در نظر گرفته می‌شود که به عنوان $MakeSpan$ (زمان تکمیل کلی وظایف) تعریف می‌شود. رابطه (۴) این معیار را نشان می‌دهد.

$$f_4 = MakeSpan = \max\{CT_{ij} | i \in PN, j \in T\} \quad (4)$$

بنابراین، f_1 توازن بار، f_2 نرخ مهاجرت، f_3 متوسط زمان اجرای وظایف و f_4 متوسط زمان اتمام وظایف می‌باشد.

۳-۲- شبیه‌سازی اجرای وظایف بر مبنای زمانبندی نوبت گردشی

وظایف براساس توالی ارائه شده در هر راه‌حل بر مبنای زمانبندی نوبت گردشی روی گره‌های فیزیکی محیط ابر اجرا شده و توابع هدف براساس این اجرا محاسبه می‌گردد. اینکار به صورت شبیه‌سازی انجام می‌شود و فقط به منظور محاسبه زمان اجرا و توازن بار گره‌های فیزیکی است. الگوریتم زمانبندی نوبت گردشی به هر پردازش زمان ثابتی برای اجرا داده می‌شود که کوانتوم نامیده می‌شود. زمانی که یک پردازش برای دوره زمانی معینی اجرا شد، معلق می‌شود و پردازش دیگری برای آن دوره زمانی خاص اجرا می‌شود. در اینجا از سوئیچ زمینه برای ذخیره حالت‌های پردازش‌های معلق شده استفاده می‌شود. در این الگوریتم مدیریت تمام پردازش‌ها بدون اولویت انجام می‌پذیرد، به همین دلیل به آن اجرای نوبت چرخشی نیز می‌گویند. در ادامه مثالی از اجرای وظایف توسط زمانبندی نوبت گردشی ارائه می‌دهیم. فرض کنید مشخصات درخواست‌های مربوط به چهار وظیفه از طرف کاربران به صورت جدول (۱) باشد.

جدول (۱) مشخصات درخواست‌های کاربران

وظایف	زمان درخواست	مدت زمان اجرا	اولویت	زمان سرویس
t_1	۰	۵	-	۹
t_2	۱	۳	-	۶
t_3	۲	۸	-	۱۴
t_4	۳	۶	-	۰

بر مبنای این اطلاعات وظایف هیچ اولویت اجرایی ندارند و زمان سرویس نیز با توجه به قدرت ماشین مجازی اجرای می‌شود. از اینرو در صورت در نظر گرفتن مدت زمان ۳ ثانیه برای کوانتوم، روش اجرای وظایف با نوبت گردشی به صورت شکل (۱) است.

در الگوریتم ژنتیک ساختار راه‌حل‌ها بر مبنای یک بردار به طول تعداد وظایف است که هر عنصر آن معرف گره فیزیکی منتسب شده به آن وظیفه است. با تعیین ساختار راه‌حل‌ها جمعیت اولیه به صورت تصادفی ایجاد شده و توابع هدف برای هر راه‌حل محاسبه می‌شود. در اینجا از چهار تابع هدف «توازن بار»، «نرخ مهاجرت»، «متوسط زمان اجرای وظایف» و «متوسط زمان اتمام وظایف» استفاده می‌شود. بنابراین مسئله به یک مسئله چند هدفه کدگذاری شده است و برای تعیین جواب‌های برتر از آرشیو پارتو استفاده می‌شود. علاوه بر این، شبیه‌سازی اجرای وظایف بر مبنای زمانبندی نوبت گردشی انجام می‌شود. این روش یک روش زمانبندی است که هر پردازش را بر اساس یک زمان ثابت به نام کوانتوم اجرا می‌کند. در مرحله بعد عملگرهای ژنتیکی اعمال شده و جمعیت نسل بعد ایجاد می‌شود. الگوریتم ژنتیک با توجه به تکرار ثابتی اعمال شده تا راه‌حل‌ها بهینه‌سازی شوند و در نهایت خروجی الگوریتم، معیارهای متفاوتی برای ارزیابی سیستم زمانبندی پیشنهادی می‌باشد. در ادامه جزئیات روش پیشنهادی ارائه می‌گردد.

۳-۱- محاسبه توابع هدف

در این مرحله شایستگی هر راه‌حل محاسبه می‌شود. این شایستگی بر اساس یک تابع چندهدفه محاسبه می‌شود. در این مقاله چهار هدف متفاوت توازن بار، نرخ مهاجرت، متوسط زمان اجرای وظایف و متوسط زمان اتمام وظایف در نظر گرفته شده که ضرورت استفاده از لیست جواب‌های پارتو را نشان می‌دهد. این اهداف به صورت زیر می‌باشند. به منظور بیان توازن بار گره‌های فیزیکی، از واریانس بار استفاده می‌کنیم. رابطه (۱) توازن بار گره‌های فیزیکی را تعریف می‌کند.

$$f_1 = \sigma(TS) = \sqrt{\frac{1}{M-1} \sum_{i=1}^M (V(pn_i, TS) - \bar{V}(TS))^2} \quad (1)$$

در این رابطه، $V(pn_i, TS)$ بار گره فیزیکی i -ام در کل دوره زمانبندی است. TS کل دوره زمانبندی در K دوره به صورت $(t_1 - t_0), (t_2 - t_1), \dots, (t_K - t_{K-1})$ است و $t_1, \dots, (t_K - t_{K-1})$ دوره زمانی k -ام می‌باشد. هر چه مقدار واریانس بار کمتر باشد نشان دهنده توازن بار کل سیستم زمانبندی است. در غیر این صورت سیستم در حالت نامتعادل است.

در بحث زمانبندی وظایف در رایانش ابری، نرخ مهاجرت به صورت نسبت تعداد وظایف منتقل شده به کل وظایف در یک دوره زمانی محاسبه می‌شود. $\rho(k)$ نرخ مهاجرت در دوره زمانی k -ام است و به صورت رابطه (۲) تعریف می‌شود.

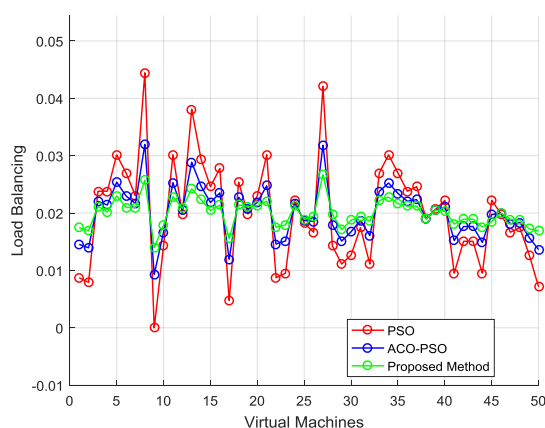
$$f_2 = \rho(k) = \frac{N'(k)}{N}, \quad k = 1, 2, \dots, K \quad (2)$$

در این رابطه $N'(k)$ تعداد وظایف منتقل شده در دوره زمانی k -ام و N تعداد کل وظایف است. $\rho(TS)$ نرخ مهاجرت در کل دوره‌های زمانی است و به صورت متوسط در K دوره در نظر گرفته می‌شود.

۴ - یافته ها

در این بخش آزمایش‌های گسترده‌ای را به منظور اثبات برتری روش پیشنهادی در توازن بار گره‌های فیزیکی یک محیط ابر شبیه‌سازی شده ارائه می‌دهیم. همه آزمایش‌ها با استفاده از نرم افزار متلب ورژن ۲۰۱۷ انجام شده است. در این تحقیق آزمایش‌ها بر روی یک سیستم ابر مجازی انجام شده است. این سیستم ابر بر اساس چندین پارامتر به طور مجازی شبیه‌سازی می‌شود. این پارامترها اغلب مطابق تحقیقات [۱۱] و [۶] هستند. از جمله این پارامترها می‌توان به تعداد گره‌های فیزیکی، قدرت هر گره برای اجرای وظایف، تعداد منابع موجود در هر گره، نوع منابع و غیره هستند.

در آزمایش نخست، بار موجود روی هر گره فیزیکی را پس از پایان شبیه‌سازی محاسبه و برای هر گره فیزیکی گزارش می‌کنیم. با هر گره فیزیکی براساس حجم وظایف تخصیص داده شده به آن گره در نظر گرفته می‌شود. نتایج این مقایسه بین روش پیشنهادی و دو روش PSO، ACO-PSO در شکل (۲) گزارش شده است.



شکل (۲) مقایسه روش‌های مختلف از لحاظ بار روی گره‌های فیزیکی

آزمایش دوم به بررسی زمان اجرای کل وظایف متمرکز شده است. در این آزمایش فرایند همگرایی و کاهش زمان اجرای وظایف نسبت به تکرار الگوریتم ژنتیک برای روش پیشنهادی و دو روش PSO و ACO-PSO مقایسه شده است. نتایج این مقایسه در شکل (۳) برای ۵۰ گره فیزیکی، ۱۰۰ کار و ۲۰۰ تکرار الگوریتم ژنتیک گزارش شده است. روش پیشنهادی در این آزمایش نیز نشان می‌دهد که همگرایی بهتری در کاهش زمان اجرای کل وظایف نسبت به سایر روش‌های مورد مقایسه دارد.

t_1	t_2	t_3	t_4	t_1	t_3	t_4	t_3	
0	3	6	9	12	14	17	20	22

شکل (۱) روش اجرای وظایف با نوبت گردشی با کوانتوم برابر ۳ ثانیه

با توجه به این شکل و فرایند اجرای وظایف در روش زمانبندی نوبت گردشی، زمان انتظار هر وظیفه در جدول (۲) نشان داده شده است. زمان انتظار با توجه به تفاضل بین زمان سرویس و زمان درخواست محاسبه می‌شود.

جدول (۲) زمان انتظار وظایف در زمانبندی نوبت گردشی

وظایف	زمان انتظار
t_1	$(0 - 0) + (12 - 3) = 9$
t_2	$(3 - 1) = 2$
t_3	$(6 - 2) + (14 - 9) + (20 - 17) = 12$
t_4	$(9 - 3) + (17 - 12) = 11$
میانگین زمان انتظار	$(9 + 2 + 12 + 11)/4 = 8.5$

۳-۳- پیکربندی الگوریتم ژنتیک

با توجه به نظریه داروین و برای تولید نسل جدید از جمعیت فعلی، باید راه‌حل‌هایی از این جمعیت برای ادغام و تکثیر انتخاب شوند. راه‌حل‌های انتخابی باید از شایستگی بیشتری برخوردار باشند. در این مقاله از عملگر چرخ رولت برای انتخاب راه‌حل‌ها استفاده می‌شود. با توجه به چند هدفه بودن مسئله، راه‌حل‌ها از آرشو جواب‌های پارتو و بر مبنای امتیاز (استعداد) بیشتر انتخاب می‌شوند.

عملگر ترکیب مهمترین عملگر الگوریتم ژنتیک است و به احتمال C_R روی دو جفت والد (راه‌حل‌های انتخاب شده) به منظور تولید راه‌حل‌های جدید اعمال می‌گردد. در اینجا از عملگر برش تک نقطه‌ای برای ایجاد راه‌حل‌های جدید (فرزندان) استفاده می‌شود. این عملگر بصورت تصادفی دو راه‌حل والد را از یک نقطه یکسان برش داده و بخش راست آنها را بدون تغییر با هم تعویض می‌کند. بدین ترتیب تمام ژن‌های سمت راست راه‌حل والد اول با همان ترتیب و بدون تغییر با ژن‌های سمت راست راه‌حل دوم جابجا می‌شود.

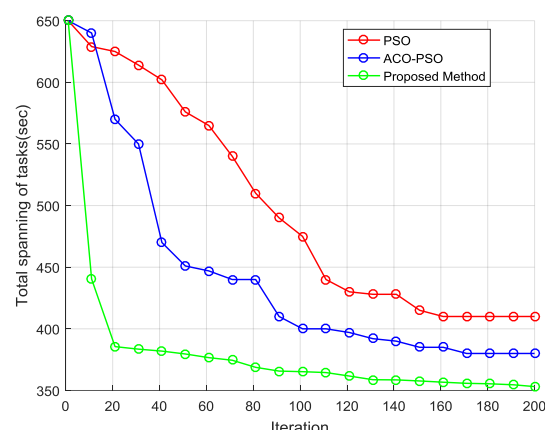
عملگر جهش اغلب برای حفظ تنوع در جمعیت اعمال می‌شود و دارای نرخ احتمال M_R به ازای هر ژن است. در این مقاله از عملگر جهش تغییر بیت استفاده شده که به ازای هر ژن و احتمال M_R یک گره فیزیکی تصادفی به وظایف نگاشت داده می‌شود.

با توجه به تعداد کل اعضای جمعیت، تولید و مثل با عملگرهای ژنتیکی انجام می‌شود و جمعیتی برابر تعداد کل اعضای جمعیت نسل قبل ایجاد می‌شود. در اینجا جمعیت نسل بعد بر مبنای بهترین اعضا هر دو جمعیت نسل قبل و جاری انتخاب می‌شود.

نهایت باعث کاهش زمان اجرای کل وظایف شده است. با توجه به چند هدف بودن مسئله در این آزمایش میانگین اهداف مربوط به بهترین عضو لیست پارتو گزارش شده است.

مراجع

- [1] Bhimani, J., Yang, Z., Leaser, M., & Mi, N. (2017, September). Accelerating big data applications using lightweight virtualization framework on enterprise cloud. In 2017 IEEE High Performance Extreme Computing Conference (HPEC) (pp. 1-7). IEEE.
- [2] Radchenko, G. I., Alaasam, A. B., & Tchernykh, A. N. (2019). Comparative Analysis of Virtualization Methods in Big Data Processing. *Supercomputing Frontiers and Innovations*, Vol. 6, No. 1, 48-79.
- [3] Buyya, R., Ramamohanarao, K., Leckie, C., Calheiros, R. N., Dastjerdi, A. V., & Versteeg, S. (2015, December). Big data analytics-enhanced cloud computing: Challenges, architectural elements, and future directions. In 2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS) (pp. 75-84). IEEE.
- [4] Lenk, A., Klems, M., Nimis, J., Tai, S., & Sandholm, T. (2009, May). What's inside the Cloud? An architectural map of the Cloud landscape. In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing (pp. 23-31). IEEE Computer Society.
- [5] Ryan, M. D. (2016). Cloud computing security: The scientific challenge, and a survey of solutions. *Journal of Systems and Software*, Vol. 86, No. 9, 2263-2268.
- [6] Netjinda, N., Sirinaovakul, B., & Achalakul, T. (2014). Cost optimal scheduling in IaaS for dependent workload with particle swarm optimization. *The Journal of Supercomputing*, Vol. 68, No. 3, 1579-1603.
- [7] Xu, Y., Li, K., Hu, J., & Li, K. (2014). A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues. *Information Sciences*, Vol. 270, 255-287.
- [8] Goyal, M., & Aggarwal, M. (2017). Optimize workflow scheduling using hybrid ant colony optimization (ACO) & particle swarm optimization (PSO) algorithm in cloud environment. *Int. J. Adv. Res. Ideas Innov. Technol*, Vol. 3, No. 2.
- [9] Dordaie, N., & Navimipour, N. J. (2017). A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments. *ICT Express*.
- [10] Pawar, N., Lilhore, U. K., Agrawal, N. (2017), A Hybrid ACHBDF Load Balancing Method for Optimum Resource Utilization In Cloud Computing. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 367-373.
- [11] Kumar, D. P., Sreenivasula T., Yugandhar A. (2014), Finding Best Time Quantum for Round Robin Scheduling Algorithm to avoid Frequent Context Switch, 6750- 6754.



شکل (۳) مقایسه روش‌های مختلف از لحاظ زمان اجرای کل وظایف

۵ - بحث و نتیجه‌گیری

نتایج آزمایش‌ها اثبات می‌کنند روش پیشنهادی در مقایسه با الگوریتم‌های «PSO» و «ACO-PSO» عملکرد و بهره‌وری مناسب‌تری را فراهم می‌کند. نتایج شکل (۲) به وضوح نشان می‌دهد که بار روی گره‌های فیزیکی در روش پیشنهادی تقریباً برای همه گره‌ها یکسان است. این آزمایش برای ۵۰ گره فیزیکی و ۱۰۰ وظیفه بررسی شده است. عملکرد ضعیف روش‌های PSO و ACO-PSO در ایجاد توازن بار نامناسب نسبت به روش پیشنهادی به دلیل تخصیص وظایف تنها براساس وضعیت فعلی سیستم است. روش ACO-PSO نشان می‌دهد که با در نظر گرفتن رویکرد حریصانه توانسته نتایجی بهتر از روش PSO کلاسیک ارائه دهد، هرچند این نتایج از روش پیشنهادی ضعیف‌تر است و نوسانات بیشتری را ایجاد می‌کند. به طور کلی در روش پیشنهادی بار موجود روی گره‌های فیزیکی در بازه $[0.017 - 0.025]$ متغیر است و این بار برای روش‌های PSO و ACO-PSO در محدوده $[0.0 - 0.043]$ و $[0.01 - 0.031]$ است. بنابراین با توجه به این نتایج، محدوده تغییر بار گره‌های فیزیکی را برای روش پیشنهادی به طور میانگین کمتر گزارش شده است. همچنین با توجه به این مهم که در دوره‌های زمانی تعادل بار گره‌های فیزیکی محاسبه می‌شود، لذا در این آزمایش تعادل بار هر گره فیزیکی به صورت متوسط کل دوره‌های زمانی گزارش شده است.

نتایج شبیه‌سازی شکل (۳) نیز نشان می‌دهد که روش پیشنهادی عملکرد و بهره‌وری بهتری نسبت به روش‌های PSO و ACO-PSO دارد. روش پیشنهادی از ۲۰۰ تکرار الگوریتم ژنتیک، در تکرار حدود ۲۰ همگرا شده و نتایج معقولی بدست آورده است. در اینجا الگوریتم به زمان اجرای حدود ۳۷۵ ثانیه رسیده است. در ادامه تکرارهای الگوریتم ژنتیک در نهایت زمان اجرا به حدود ۳۵۵ ثانیه کاهش یافته است. این در حالیست که روش PSO در تکرار ۱۶۰ به زمان اجرای ۴۱۰ و روش ACO-PSO در تکرار ۱۰۰ به زمان اجرای ۴۰۰ ثانیه همگرا شده است. به طور کلی روش پیشنهادی با ایجاد توزیع مناسب وظایف بین گره‌های فیزیکی و تخصیص آنها به منابع موجود جهت اجرا، پاسخی تقریبی و نزدیک به بهینه حاصل شده است. اینکار در