

بهینه‌سازی زمانبندی وظایف در محیط ابر با استفاده از ویرایش فازی الگوریتم بهینه‌سازی اجتماع ذرات*

شبنم قرائیان^۱

خسرو امیری زاده^۲

چکیده

امروزه با توجه به استفاده روزافزون از محیط ابر توسط شرکت‌ها و سازمان‌ها، زمان‌بندی وظایف در این نوع محیط‌ها از اهمیت ویژه‌ای برخوردار است. الگوریتم‌های متنوعی برای تخصیص وظایف به منابع در محیط ابر پیشنهاد شده است که اغلب آن‌ها معیارهایی همچون توزیع بار متعادل، تخصیص بهینه منابع و کاهش زمان اجرای وظایف را در نظر نمی‌گیرند. در این کار با استفاده از الگوریتم فرااکتشافی ازدحام ذرات و منطق فازی، زمان اجرای وظایف کاهش و در نتیجه بهره‌وری منابع افزایش می‌یابد. به‌طور کلی در یک سیستم توزیع شده همچون محیط ابر، توزیع وظایف بر روی ماشین‌های مجازی به صورت تصادفی انجام می‌شود. در نتیجه، بار کل بر روی محیط ابر غیرمتعادل می‌شود که این امر از بهره‌وری منابع می‌کاهد. در این تحقیق، الگوریتم ازدحام ذرات^۳ و منطق فازی برای زمانبندی وظایف استفاده می‌شود. علاوه بر این، استفاده از الگوریتم آنیلینگ شبیه‌سازی شده^۴ برای بهبود شرایط اولیه داده‌های تصادفی پیشنهاد می‌شود. نتایج نشان می‌دهد که روش بهینه‌سازی پیشنهادی در این کار، معیارهای عملکرد الگوریتم زمانبند مانند زمان خاتمه کار^۵ را در مقایسه با الگوریتم‌های فاقد بهینه‌سازی (مانند راندرابین^۶) و حتی در مقایسه با الگوریتم‌های بهینه‌سازی دیگر همچون الگوریتم ژنتیک^۷ به طور موثری بهبود می‌بخشد.

کلمات کلیدی: رایانش ابری؛ زمانبندی وظایف؛ الگوریتم ازدحام ذرات؛ منطق فازی؛ الگوریتم آنیلینگ شبیه‌سازی شده

* تاریخ دریافت: ۹۷/۶/۱۰؛ تاریخ پذیرش: ۹۷/۱۰/۲۱.

^۱ کارشناس ارشد مهندسی کامپیوتر، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه آزاد اسلامی، واحد گرمسار، گرمسار، ایران
shabnam.gh7@gmail.com

^۲ استادیار گروه مهندسی کامپیوتر، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه آزاد اسلامی، واحد گرمسار، گرمسار، ایران (نویسنده مسئول)
khosrowamirizadeh@gmail.com

^۳ Particle Swarm Optimization (PSO)

^۴ Simulated Annealing

^۵ Make Span

^۶ Round-Robin

^۷ Genetic Algorithm

مقدمه

در حال حاضر، محاسبات ابری (پوتینی و محمد، ۲۰۱۳؛ گارگ و همکاران، ۲۰۱۲) از ارزشمندترین و نویدبخش‌ترین مسیرهای پژوهشی پس از محاسبات توزیع شده است (نان و همکاران، ۲۰۱۴). این روش محاسبه، خدمات زیرساختی و نرم‌افزاری برای کاربران فراهم می‌کند و به ارائه سرویس‌های درخواستی کاربران از طریق اینترنت می‌پردازد (یو و همکاران، ۲۰۱۳). با توجه به رشد چشمگیر محاسبات ابری، تعداد کلاینت‌ها و تعداد تقاضاها به سرعت در حال افزایش است. این امر حجم کاری سنگینی را روی سرورها و منابع محاسباتی موجب می‌شود. این شرایط طبیعتاً نیازمند یک الگوریتم زمانبندی برای استفاده بهینه از منابع است.

زمان بندی کارها روشی است که وظایف کاربران را به منظور اجرا به ماشین‌های مجازی تخصیص می‌دهد. از دید مشتری، الگوریتم زمان بندی مناسب باید بتواند وظایف خواسته شده را در کمترین زمان اجرا کند. از سوی دیگر، شرکت ارائه‌دهنده سرویس نیز نیازمند نوعی زمان بندی است که بتواند در عین جلب رضایت مشتری، از حداکثر امکانات منابع استفاده کند. این مسئله نیازمندی فراهم کننده سرویس به انتخاب روش بهینه برای زمان بندی درخواست‌ها را بیشتر می‌کند.

در این مقاله یک الگوریتم زمان بندی جدید با استفاده از الگوریتم اکتشافی بهینه سازی ازدحام ذرات فازی (لیو و همکاران، ۲۰۱۰) با این هدف ارائه می‌شود که روش پیشنهادی بتواند نیازمندی‌های کیفیت سرویس را از منظر فراهم کننده سرویس و مشتری برآورده کند. هدف از الگوریتم ارائه شده استفاده از بهینه ساز ذرات برای تأمین نیازمندی‌های زیر است:

- ۱: کاهش زمان تکمیل جریان کاری
- ۲: کاهش زمان پاسخ به درخواست‌ها
- ۳: افزایش میزان تعادل بار

برای ارزیابی روش پیشنهادی از ابزار منبع باز CloudSim (یو و همکاران، ۲۰۱۳) استفاده شده است. ما هسته اولیه این ابزار را برای ارائه الگوریتم مورد نظر توسعه داده‌ایم و سپس روش پیشنهادی خود را با الگوریتم‌های مشهور PSO برهارت و کندی (۱۹۹۵) و ژنتیک گلدبرگ و هالند (۱۹۸۸) مقایسه کرده‌ایم. ادامه مقاله از بخش‌های زیر تشکیل شده است: در بخش دوم کارهای مرتبط با زمانبندی بار کاری بیان شده است، سپس در بخش سوم جزئیات روش پیشنهادی به تفصیل ارائه شده است، در بخش چهارم شرایط ارزیابی روش پیشنهادی و نتایج الگوریتم ارائه شده است و در نهایت در بخش آخر نتیجه‌گیری و کارهای آینده ارائه می‌شوند.

یکی از مهم‌ترین مسئله‌ها در بستر ابر، زمانبندی کارها می‌باشد؛ زمانبندی مکانیزمی است که وظایف کاربران را به منابع مناسب جهت اجرا اختصاص می‌دهد. به طور کلی، کارایی روش‌های زمانبندی به طور مستقیم بر روی کارایی کل سیستم محاسبات ابری تأثیر می‌گذارد. در سال‌های گذشته، محققان متعددی برای انجام زمانبندی بهینه در رایانش ابری به رهیافت‌های مبتنی بر هوش مصنوعی روی آورده‌اند. از آنجا که روش پیشنهادی ما نیز در حوزه هوش مصنوعی می‌باشد، در ادامه به بررسی روش‌هایی می‌پردازیم که در این حوزه استفاده شده‌اند.

زمانبندی وظایف در محیط ابر نقش عمده‌ای در قابلیت اطمینان و کارایی آن دارد. با افزایش تعداد درخواست‌ها و تنوع ماشین‌های مجازی و همچنین افزایش اهداف کیفیت سرویس، پیدا کردن راه‌حل نزدیک به بهینه، پیچیده‌تر و چالش برانگیزتر می‌شود و در نتیجه مسئله زمانبندی به عنوان یکی از مهمترین مسائل np-hard در محیط‌های ابر مطرح می‌شود. محققان برای حل این مسئله، با توجه به کارایی الگوریتم‌های فرااکتشافی مانند ژنتیک، بهینه‌سازی ازدحام ذرات، کلونی مورچگان و غیره، آنها را برای پیدا کردن یک جایگشت مناسب از درخواست‌ها بر روی ماشین‌های مجازی پیشنهاد داده‌اند. هر کدام از این الگوریتم‌ها روی یک یا چند هدف مانند تعادل بار، بهره‌وری منابع، حداقل زمان پاسخ

و کاهش اتلاف حافظه تمرکز دارند. در ادامه برخی از مهم‌ترین کارها در حوزه زمانبندی بیان می‌شود.

جینگ و همکاران (۲۰۱۳) الگوریتم MO-GA را برای زمانبندی وظایف پیشنهاد دادند. هدف آنها، حداقل کردن مصرف انرژی و حداکثر استفاده بهینه از سرویس‌های محاسبات ابری، با توجه به ضرب‌الاجل هر کدام از وظایف است. برای رسیدن به اهداف مذکور، ابتدا معماری زمانبندی وظایف در محیط محاسبات ابری تعریف شده است. سپس، الگوریتم MO-GA برای زمانبندی وظایف در شبیه‌ساز Cloudsim پیشنهاد شده است (لیو و همکاران، ۲۰۱۰). جوانمردی و همکارانش در سال ۲۰۱۴، یک الگوریتم ترکیبی جهت توازن بار، کاهش زمان اجرا و هزینه‌ها، از نوع فرااکتشافی پیشنهاد داده‌اند. در رهیافت پیشنهادی ایشان از ترکیب الگوریتم ژنتیک و تئوری فازی استفاده شده است. در واقع الگوریتم جدید کارها را براساس طول کار و ظرفیت منابع، به منابع اختصاص می‌دهد (جوانمردی و همکاران، ۲۰۱۳).

ژانگو و همکارانش، یک مدل تعادل بار را بر مبنای کلونی بهینه‌سازی شده مورچگان ارائه داده‌اند که روی فدراسیون محاسبات باز ابرها انجام شده است. این فدراسیون شامل بسیاری از سرویس‌دهندگان سرویس محاسبات ابری می‌باشد و هدف این است که بار را به طور پویا در سراسر فدراسیون ابر تقسیم کند. این الگوریتم میانگین نسبی درصد تعادل بار را به حداقل می‌رساند و با القای قابلیت کلونی مورچه‌ها، جواب بهینه برای نحوه زمانبندی وظایف پیدا می‌کند. همچنین در الگوریتم ذکر شده، ویژگی‌های کوچک و مقیاس آزاد شبکه‌های پیچیده نیز در نظر گرفته شده است (عبدالحمید و همکاران، ۲۰۱۵).

در مقاله رمضانی و همکاران (۲۰۱۴) برای رسیدن به تعادل بار، نیاز بود به جای این که کل وظیفه منتقل شود، وظایف اضافی از روی ماشین مجازی پربار منقل شود. الگوریتم TBSLB-PSO برای ایجاد تعادل بار در محیط ابری پیشنهاد شده است. در روش پیشنهادی، مدل بهینه‌سازی مهاجرت وظیفه پیشنهاد شده است که مدت زمان اجرای وظایف و زمان انتقال وظیفه را حداقل می‌کند.

در مقاله پارساراتی و ونداتسواران (۲۰۱۲) الگوریتم اصلاح شده ازدحام ذرات برای زمانبندی در محیط‌های ابری پیشنهاد شده است. هدف اصلی از ارائه این راهکار، زمان بندی بهینه وظایف در محیط‌های ابری به گونه‌ای است که زمان اجرا و در ادامه آن زمان deadline کاهش پیدا کند. سه الگوریتم اکتشافی برای زمانبندی وظایف در محیط محاسبات ابری با یکدیگر مقایسه شده‌اند. این سه الگوریتم اکتشافی، الگوریتم PSO، الگوریتم ژنتیک و الگوریتم PSO اصلاح شده است. با نتایج به دست آمده از شبیه‌سازی، هنگامی که داده‌ها بیش‌تر می‌شوند، الگوریتم PSO بهتر از الگوریتم ژنتیک عمل می‌کند. در مقاله ژانگ و ژانگ (۲۰۱۰) از منطق فازی برای ایجاد تعادل بار در محیط محاسبات ابری استفاده شده است. با استفاده از منطق فازی، سرعت پردازنده و تخصیص بار در ماشین‌های مجازی با هدف تعادل بار انجام می‌شود. در این مقاله برای ایجاد تعادل بار از منطق فازی براساس نوبت گردش و به منظور بهره‌وری منابع و دسترسی به منابع در محیط ابری استفاده شده است. دو پارامتر سرعت پردازنده و بار ماشین‌های مجازی برای ارزیابی تعادل بار با استفاده از منطق فازی استفاده شده‌اند.

در مقاله سوشی (۲۰۱۲) الگوریتم زمان بندی محیط‌های ابری با استفاده از شبکه عصبی فازی پیشنهاد شده است. ابتدا وظایف کار براساس کیفیت سرویس از جمله حافظه و اندازه و... طبقه‌بندی می‌شوند، سپس وظایف کلاسه‌بندی شده به یک سرویس فازی ارائه می‌گردند، بطوری که مقادیر ورودی به بازه‌های بین صفر و یک تبدیل می‌شوند. شبکه‌های عصبی که از سه لایه ورودی، مخفی و خروجی تشکیل شده‌اند برای تعدیل وزن وظایف کاربر و هماهنگی آن با منابع در دسترس سیستم مورد استفاده قرار می‌گیرند. نتایج حاصل از این الگوریتم نشان داده است که زمان برگشت کلی کاهش و بهره‌وری سیستم افزایش یافته است.

روش‌شناسی پژوهش

در این بخش ابتدا الگوریتم بهینه‌سازی ازدحام ذرات و منطق فازی بیان شده است. سپس در بخش بعد نحوه استفاده از ترکیب این دو روش در مسئله زمانبندی ارائه می‌شود.

الگوریتم بهینه‌سازی ازدحام ذرات پایه

الگوریتم بهینه‌سازی ازدحام ذرات از دسته الگوریتم‌های بهینه‌سازی است که بر مبنای تولید تصادفی جمعیت اولیه عمل می‌کنند. این الگوریتم با الگوگیری از رفتار پرواز دسته جمعی (گروهی) پرندگان یا حرکت دسته جمعی (گروهی) ماهی‌ها بنا نهاده شده است. هر عضو در این گروه توسط بردار سرعت و بردار موقعیت در فضای جستجو تعریف می‌گردد. در هر تکرار، موقعیت جدید ذرات با توجه به بردار سرعت و بردار موقعیت در فضای جستجو با استفاده از رابطه (۱) تعریف می‌گردد. در هر تکرار زمانی، سرعت جدید ذرات با توجه به بردار سرعت فعلی، بهترین موقعیت یافت شده توسط آن ذره و بهترین موقعیت یافت شده توسط بهترین ذره موجود در گروه (رابطه ۲)، بروزرسانی می‌شود.

$$x_j^i[t+1] = x_j^i[t] + v_j^i[t+1] \quad \text{رابطه ۱}$$

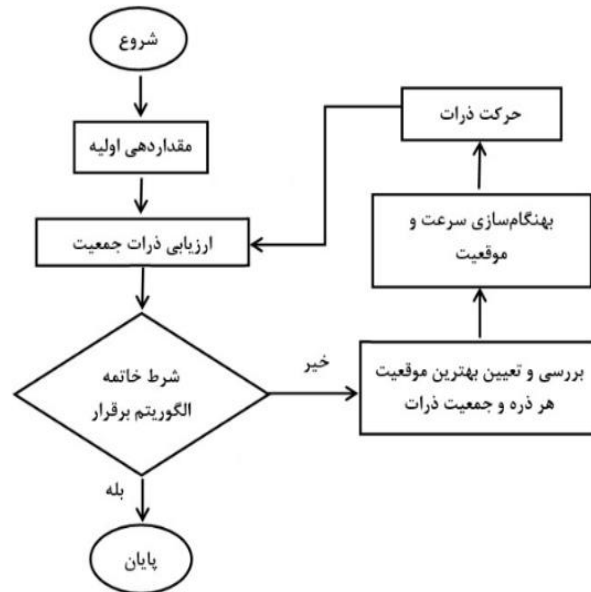
رابطه $x_j^i[t+1]$ مکان ذره در لحظه $t+1$ و $x_j^i[t]$ مکان ذره در لحظه t و $v_j^i[t+1]$ سرعت ذره در لحظه $t+1$ می‌باشد.

$$v_j^i[t+1] = wv_j^i[t] + c_1r_1(x_j^{ibest}[t] - x_j^i[t]) + c_2r_2(x_j^{gbest}[t] - x_j^i[t]) \quad \text{رابطه ۲}$$

به طوری که در رابطه بالا، w فاکتور سکون^۱، و r_1 و r_2 اعداد تصادفی هستند که برای حفظ گوناگونی جمعیت مورد استفاده قرار می‌گیرند و به صورت یکنواخت در بازه $[0,1]$ برای i امین بعد ذره t ام توزیع می‌گردد. طبق رابطه (۲) ذره قادر خواهد بود که حرکت بعدی خود را با توجه به تجربه حاصل از بهترین موقعیت گذشته خود و تجربه ذرات موفق، تصمیم‌گیری نماید. شکل ۱ فلوجارت الگوریتم ازدحام ذرات را نشان می‌دهد. $C1$ و $C2$ به ترتیب تعیین کننده نقش بهترین مکان در تاریخچه خود ذره، و بهترین مکان تا کنون بدست آمده در بین کل ذرات، در تعیین سرعت جدید ذره هستند. لازم به

¹ Inertial Factor

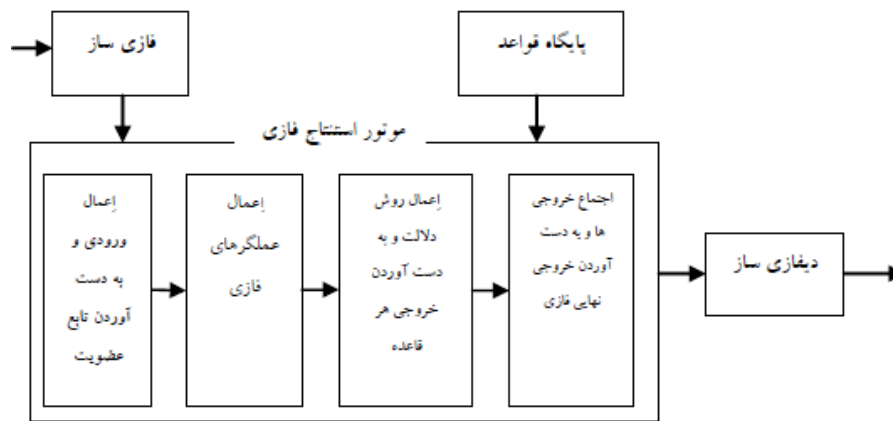
ذکر است که نتایجی که در مقاله خواهند آمد، پس از انجام اجراهای متعدد و یافتن مقدار بهینه پارامترهای آزاد، از جمله C1 و C2، بدست آمده‌اند.



شکل ۱. شمای کلی از الگوریتم PSO

منطق فازی

در تئوری کلاسیک مجموعه‌ها، یک عنصر دو حالت دارد، یا عضو مجموعه هست یا نیست. به عبارتی دیگر، عضویت عناصر از یک الگوی صفر و یک پیروی می‌کند. اما تئوری مجموعه‌های فازی این مفهوم را توسعه داده است. در این منطق، عضویت درجه‌بندی می‌شود. بنابراین، یک عنصر می‌تواند تا درجاتی و نه کاملاً عضو یک مجموعه باشد. شکل ۲ نشان‌دهنده سیستم استنتاج فازی در روش پیشنهادی ما است.



شکل ۲. سیستم استنتاج فازی

استفاده از الگوریتم بهینه‌سازی ازدحام ذرات به همراه منطق فازی در مسئله زمانبندی

در این بخش، ما یک طرح فازی مبتنی بر بهینه‌سازی ازدحام ذرات گسسته برای حل مسئله زمانبندی وظایف در شبکه‌های محاسبات ابری طراحی خواهیم کرد. بردارهای ماتریس فازی برای نشان دادن موقعیت و سرعت ذرات برای زمانبندی وظایف در این شبکه‌ها محاسبه می‌شوند. با این رویکرد، جواب‌های پیدا شده توسط الگوریتم ازدحام ذرات به جواب‌های واقعی و بهینه نزدیک خواهد شد و این انتظار می‌رود که زمانبندی وظایف در محیط ابری با کاهش پیچیدگی و کاهش زمان تکمیل کار انجام گیرد.

مدل ریاضی الگوریتم

فرض کنید $VM = \{VM_1, VM_2, \dots, VM_m\}$ مجموعه ماشین‌های مجازی باشند که برای میزبانی درخواست‌های کاربران استفاده می‌شوند و همچنین $Task = \{T_1, T_2, \dots, T_n\}$ مجموعه‌ای از وظایفی باشند که قصد اجرای آنها بر روی ماشین‌های مجازی را داریم. هدف از الگوریتم، کاهش میزان زمان انجام طولانی‌ترین کار و افزایش درجه تعادل بار کرشنا (۲۰۱۳) است. زمان انجام طولانی‌ترین کار را به صورت رابطه (۳) تعریف می‌کنیم:

$$Makespan = \max\{CT_{ij} | i \in T, i = 1, 2, \dots, n \text{ and } j \in VM, j = 1, 2, \dots, m\} \quad \text{رابطه ۳}$$

بهبودسازی زمانبندی وظایف در محیط ابر با استفاده از ویرایش فازی الگوریتم // ۲۰۷

ما از معیار درجه عدم تعادل^۱ برای اندازه گیری تناسب توزیع بار استفاده کرده ایم. این معیار طبق رابطه (۴) محاسبه می شود.

$$DI = \frac{T_{\max} - T_{\min}}{T_{\text{avg}}} \quad (\text{رابطه ۴})$$

در رابطه (۴) T_{\max} و T_{\min} بیشترین و کمترین زمان انجام کارها در بین ماشین های مجازی هستند و T_{avg} متوسط زمان اجرای کارها در بین تمام ماشین های مجازی است. در نتیجه تابع برازندگی با هدف کاهش زمان تکمیل کار به صورت رابطه (۵) تعریف می شود:

$$\text{Fitness Function} = \text{Min (Makespan)} \quad (\text{رابطه ۵})$$

با توجه به رابطه (۵) می توان نتیجه گرفت که هر چقدر مقدار تابع برازندگی کمتر باشد میزان مطلوبیت ذره از جهت کاهش زمان انجام طولانی ترین کار و افزایش درجه تعادل بار بیشتر است. در ادامه جزئیات الگوریتم پیشنهادی را به تفصیل بیان می کنیم.

جزئیات روش پیشنهادی

انتخاب ذرات اولیه و سرعت اولیه

برای ایجاد مکان اولیه برای ذرات، ابتدا برای هر ذره یک لیست به اندازه تعداد وظایف ایجاد کرده و سپس برای هر وظیفه یک آرایه به تعداد ماشین های مجازی ایجاد می کنیم. برای هر ماشین مجازی عدد تصادفی بین صفر و یک را انتخاب کرده و آن ماشین به میزان آن عدد شانس انتخاب شدن برای اجرای آن وظیفه را دارد. نمایی از مکان راه حل های اولیه در زیر نشان داده شده است.

$$\begin{pmatrix} x_{11} & \dots & x_{1n} \\ x_{21} & \dots & x_{2n} \\ \vdots & \dots & \vdots \\ x_{m1} & \dots & x_{mn} \end{pmatrix}$$

استفاده از الگوریتم آنیلینگ شبیه‌سازی شده برای بهبود مکان ذرات

در این مرحله، از الگوریتم سرد و گرم فلزات یا همان آنیلینگ شبیه‌سازی شده برای بهبود مکان اولیه استفاده می‌شود. الگوریتم به این ترتیب عمل می‌کند که در ابتدا جوابی تصادفی (s) را انتخاب می‌کند، سپس وارد حلقه تکرار می‌شود؛ در هر تکرار، جواب تصادفی جدیدی (s') متناسب با جواب فعلی انتخاب می‌شود. در اینجا تابع احتمالی محاسبه می‌شود و متناسب با آن جواب جدید پذیرفته و جایگزین جواب پیشین می‌شود یا این که رد می‌شود. برای هر یک از جواب‌ها یک تابع تناسب^۱ f(s) و f(s') محاسبه می‌شود که در کار فعلی، این تابع، زمان خاتمه کار یا همان makespan خواهد بود. تابع احتمال به صورت رابطه (۶) محاسبه می‌شود:

$$P = \exp((f(s) - f(s')) / T) \quad \text{رابطه ۶}$$

در رابطه فوق، T دما می‌باشد که به تدریج با افزایش شمارش تکرارها کاهش می‌یابد. لازم به ذکر است که در برنامه نوشته شده، در صورتی که f(s') بهتر از f(s) باشد (کوچکتر باشد)، جواب جدید به طور مستقیم پذیرفته می‌شود و از تابع احتمال تنها برای (عدم) پذیرش جواب‌های نامطلوب استفاده می‌شود. در ضمن کاهش دما با پیشرفت فرایند حل به این معنا خواهد بود که احتمال پذیرش جواب‌های نامطلوب با حرکت به سمت تکرارهای پایانی کم و کمتر می‌شود. دلیل این امر افزایش احتمال گشتن تمام فضای حل در عین همگرایی به یک جواب معقول در انتها تکرار می‌باشد. در برنامه نوشته شده، دما به شکل خطی با افزایش شمارش تکرار از یک به سمت صفر حرکت می‌کند.

نرمال سازی مقادیر مکان ذرات

در این بخش برای هر یک از اعضای ذرات نرمال‌سازی استفاده می‌شود. نرمال‌سازی به منظور قرار دادن داده‌ها در بازه [0-1] صورت می‌گیرد. برای نرمال کردن هر ستون، مقدار داده هر عضو بر مجموع داده‌های ستون ماتریس تقسیم می‌شود. ماتریس نرمال شده در رابطه (۷) نمایش داده شده است:

^۱ Fitness

$$\left(\begin{array}{ccc} \frac{x_{11}}{\sum_{i=1}^m x_{i1}} & \dots & \frac{x_{in}}{\sum_{i=1}^m x_{in}} \\ \frac{x_{21}}{\sum_{i=1}^m x_{i1}} & \dots & \frac{x_{2n}}{\sum_{i=1}^m x_{in}} \\ \frac{x_{m1}}{\sum_{i=1}^m x_{i1}} & \dots & \frac{x_{mn}}{\sum_{i=1}^m x_{in}} \end{array} \right) \quad \text{رابطه (۷)}$$

محاسبه تابع برازندگی و انتخاب مقادیر Gbest و Pbest

در این مرحله مقدار تابع برازندگی را برای هر یک از ذرات طبق رابطه (۵) محاسبه می‌کنیم. مقدار بهترین موقعیت محلی را مقدار اولیه هر ذره قرار می‌دهیم و از بین ذرات، ذره‌ای که دارای بهترین مقدار تابع برازندگی است را به عنوان بهترین موقعیت سراسری انتخاب می‌کنیم.

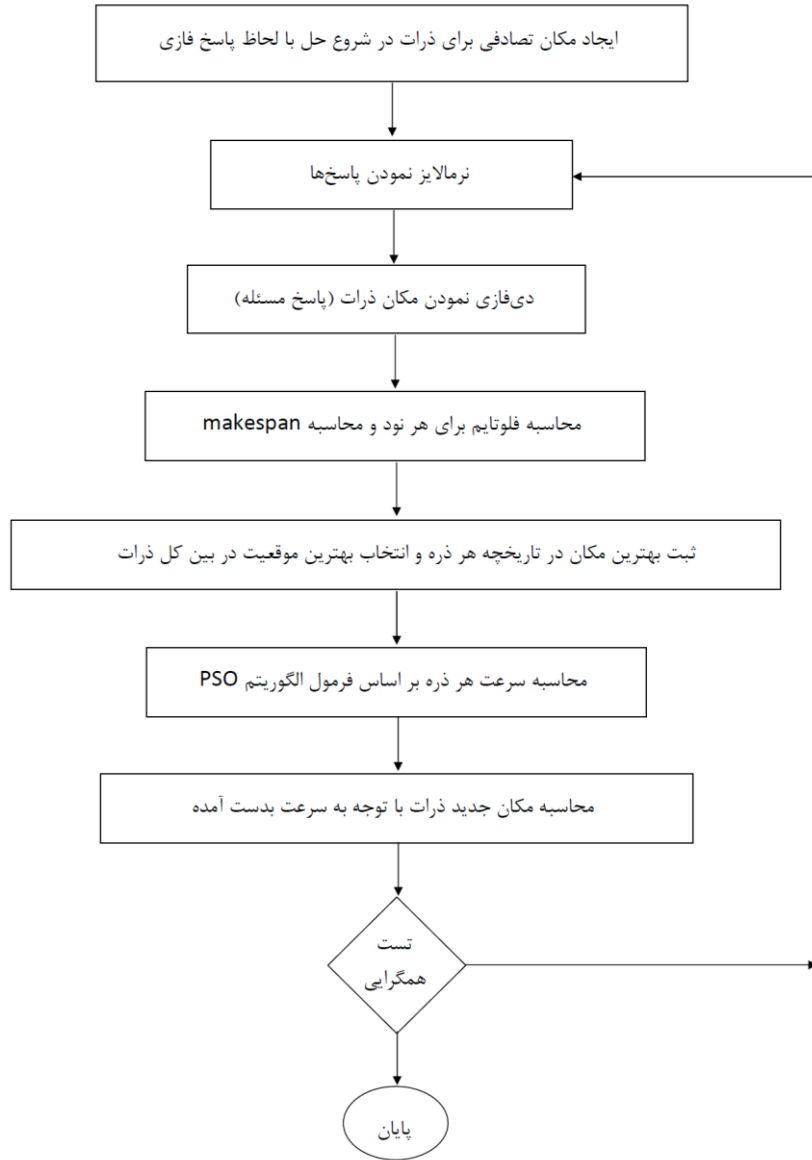
به روز رسانی مکان ذرات و سرعت آنها

در این مرحله، مطابق رابطه (۸) ابتدا میزان سرعت اولیه بر اساس اطلاعاتی که از سرعت مرحله قبل و بهترین مکان فعلی ذره و بهترین مکان در کل ذرات داریم، به روز می‌شود، سپس مکان ذره بر اساس سرعت به روز شده طبق رابطه (۹) تعیین می‌شود. سپس تابع برازندگی برای موقعیت جدید ذره محاسبه شده و در صورتی که مقدار آن از موقعیت قبلی ذره کمتر بود، مکان بهترین موقعیت محلی ذره به مکان جدید انتقال می‌یابد. در بین مکان جدید ذرات، اگر مقدار تابع برازندگی ذره‌ای از مقدار بهترین موقعیت سراسری قبلی کمتر باشد، مقدار بهترین موقعیت سراسری نیز بروزرسانی می‌شود.

$$X[t+1] = X[t] \oplus V[t+1] \quad \text{رابطه (۸)}$$

$$V[t+1] = w \otimes V[t] \oplus (c_1 * r_1) \otimes (X^{ibest}[t] \ominus X[t]) \oplus (c_2 * r_2) \otimes (X^{gbest}[t] \ominus X[t]) \quad \text{رابطه (۹)}$$

به تعداد مراحل الگوریتم، موقعیت هر ذره بروزرسانی شده و همچنین بر اساس موقعیت هر ذره و گروه ذرات، مقادیر بهترین موقعیت محلی و بهترین موقعیت سراسری محاسبه می‌شوند. در نهایت ذره‌ای که کمترین مقدار بهترین موقعیت سراسری را دارد به عنوان پاسخ مساله در نظر گرفته می‌شود. در شکل (۳) فلوجارت الگوریتم پیشنهادی قرار گرفته شده است.



شکل ۳. شمای کلی روش پیشنهادی

یافته‌های پژوهش

در این بخش، رویکرد پیشنهادی مقاله شامل زمانبندی وظایف با استفاده از الگوریتم ازدحام ذرات و منطق فازی در رایانش ابری با شبیه‌ساز کلودسیم کالهریوس و همکاران (۲۰۱۱) آزمایش خواهد شد. این پژوهش قصد دارد با بکارگیری الگوریتم ازدحام ذرات و منطق فازی به یک رویکرد بهینه در زمانبندی وظایف در محیط‌های ابری برسد تا زمان اجرای وظایف کاهش یابد و توزیع وظایف به شکل همگن‌تری صورت گیرد.

برای ارزیابی الگوریتم ارائه شده، یک مرکز داده رایانش ابری را شبیه‌سازی می‌نماییم که از ۲ میزبان تشکیل شده که هر کدام قابلیت مجازی‌سازی^۱ دارند. در واقع فرض شده است که بر روی آن‌ها مجازی‌سازهایی مثل Xen نصب شده است، که می‌تواند منابع را به اشتراک بگذارد. جدول (۱) مشخصات ماشین‌های فیزیکی را نشان می‌دهد.

جدول ۱. مشخصات ماشین‌های فیزیکی مورد استفاده برای ارزیابی

CPU MIPS	Core	RAM (GB)	Storage (GB)	Band Width (GBPS)
۱۰۰۰۰۰۰	۴	۲۰ GB	۱ GB	۱۰
۱۰۰۰۰۰۰	۲	۲۰ GB	۱ GB	۱۰

همچنین از ۱۰ ماشین مجازی با مشخصات مختلف برای ارزیابی روش ارائه شده استفاده می‌شود. جدول (۲) مشخصات ماشین مجازی مورد استفاده را نشان می‌دهد.

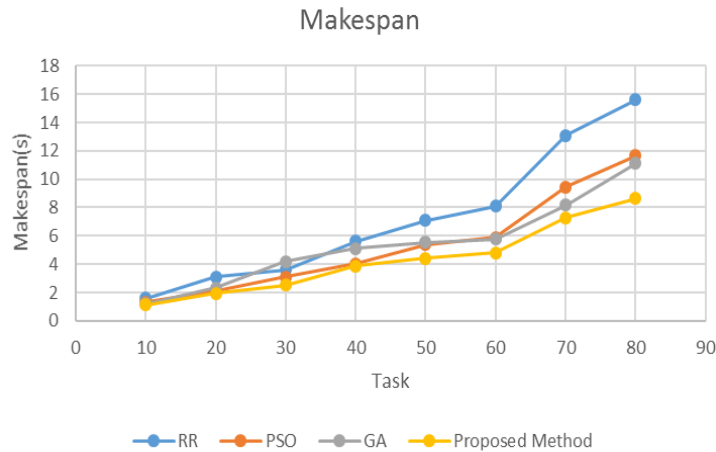
جدول ۲. مشخصات ماشین‌های مجازی مورد استفاده برای ارزیابی

CPU MIPS	Core	RAM (GB)	Storage (GB)	Band Width (GBPS)
۱۰۰۰-۱۰۰۰۰	۱	۰/۵ GB	۱۰ GB	۱

جدول ۳. مشخصات الگوریتم‌های مورد استفاده برای ارزیابی

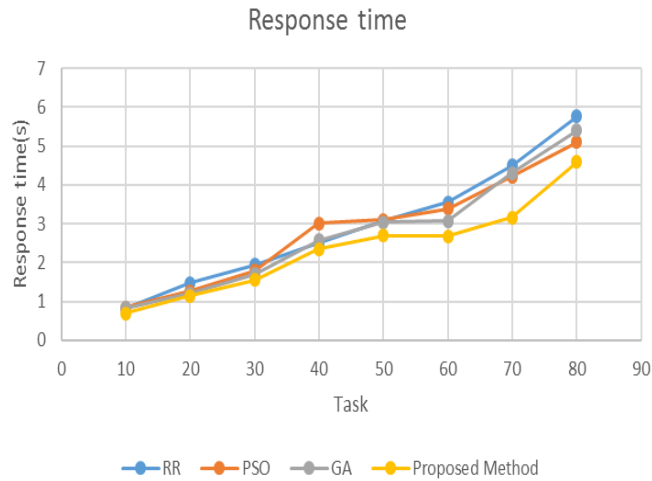
Algorithm	Parameter	Value
PSO	Particle Size	۱۰
	Self-Recognition coefficient C1	۲
	Social Effect C2	۲
	Static Inertia Weight	۰/۹
	Variable Inertia Weight	۰/۰-۴/۹
	Number of Iterations	۱۰۰
GA	Population Size	۱۰
	Number of Iterations	۱۰۰

درخواست‌ها (وظایف) به صورت نمونه ایجاد می‌شوند. طول درخواست‌ها از ۱۰۰ تا ۱۰۰۰۰ متغیر است و همه درخواست‌ها برای هر یک از روش‌ها یکسان است. شکل ۴ مقادیر makespan را نشان می‌دهد.



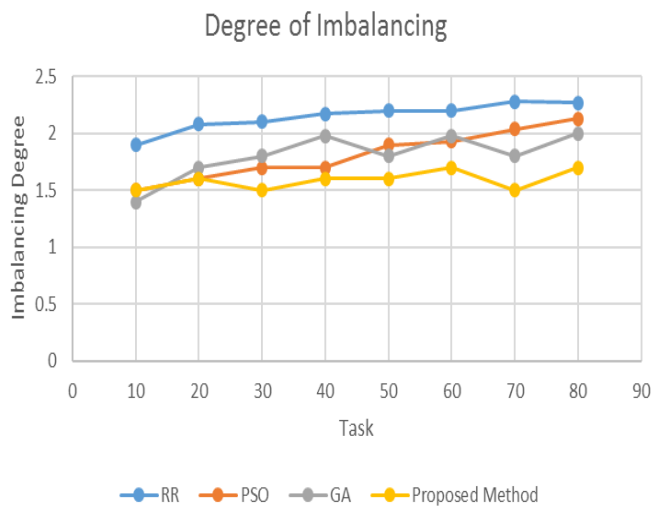
شکل ۴. مقادیر makespan برای درخواست‌های نمونه

مطابق شکل ۴، با افزایش درخواست‌ها، میزان makespan در روش پیشنهادی نسبت به الگوریتم‌های ژنتیک و بهینه‌سازی ازدحام بهبود بیشتری دارد. شکل ۵ مقادیر زمان پاسخ را برای هر یک از توزیع‌ها نشان می‌دهد.



شکل ۵. مقادیر response time برای درخواستهای نمونه

همانطور که دیده می‌شود، استفاده از روش ارائه شده سبب کاهش میزان زمان پاسخ در زمانبندی وظایف می‌شود (شکل ۵). همچنین چون در روش پیشنهادی میزان Makespan کاهش می‌یابد، اختلاف کمترین زمان اتمام کار و بیشترین زمان تکمیل کار و در نتیجه میزان درجه عدم تعادل بار نیز کاهش می‌یابد (شکل ۶).



شکل ۶. مقادیر درجه تعادل بار برای توزیع نمونه در سناریو دوم

نتیجه‌گیری و پیشنهادها

با توجه به اهمیت و نو بودن موضوع محاسبات ابری، در این مقاله روشی جهت اختصاص بهینه منابع محاسباتی به وظایف در این گونه محیط‌ها پیشنهاد شد. این روش، ترکیبی از الگوریتم ازدحام ذرات و منطق فازی برای دستیابی به یک روش بهینه زمان‌بندی است. از رویکرد فازی برای افزایش دقت زمان‌بندی و اختصاص منابع استفاده شده است. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی به طور متوسط ۲۱/۲ درصد بهبود در فاکتور makespan ۱۳/۴۵ درصد بهبود در فاکتور زمان پاسخ، و ۱۶/۴ درصد بهبود در درجه تعادل بار برای همه سناریوها ایجاد می‌کند. در آینده می‌توان روش پیشنهادی را برای محاسبات مه به کار گرفت. همچنین می‌توان از این روش برای درخواست‌های پویا در محیط ابر استفاده کرد.

منابع

- Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R. CloudSim. (2011). A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw Pract Exp.* 41, 23-50.
- Eberhart RC, Kennedy J. (1995). A new optimizer-using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 26, 39-43.
- E. Goldberg and J. Holland. (1988). Genetic algorithms and machine learning”, *Mach Learn.* 3(2), 95-99.
- F. Ramezani, J. Lu, and F. K. Hussain. (2014). Task-based system load balancing in cloud computing using particle swarm optimization. *International journal of parallel programming*, 42, 739-754.
- G. Nan, Z. Mao, M. Li, Y. Zhang, S. Gjessing, H. Wang, and M. Guizani. (2014). Distributed resource allocation in cloud-based wireless multimedia social networks. *IEEE Netw. Mag.*, 28(4), 74-80.
- H. Liu, A. Abraham, and A. E. Hassanien. (2010). Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm. *Future Generation Computer Systems*, 26(8), 1336-43.
- J. Liu, X. G. Luo, X. M. Zhang, and F. Zhang. (2013). Job scheduling algorithm for cloud computing based on particle swarm optimization. in *Advanced Materials Research*, 957-960.
- Krishna PV. (2013). Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Appl Soft Comput.* 13, 2292-303.
- R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang (2013) Toward cloud based vehicular networks with efficient resource management. *IEEE Netw. Mag.*, 27(5), 48-55.
- S. Javanmardi, M. Shojafar, D. Amendola, N. Cordeschi, H. Liu, and A. Abraham. (2014). Hybrid job scheduling algorithm for cloud computing environment. in *Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014*, 43-52.
- S. K. Garg, S. Versteeg, and R. Buyya. (2012). A framework for ranking of cloud computing services. *Future Generation Comput. Syst.*, 29(4), 1012-1023.
- S. M. Abdulhamid, M. S. A. Latiff, and I. Idris. (2015). Tasks scheduling technique using league championship algorithm for makespan minimization in IAAS cloud. *arXiv preprint arXiv: 1510.03173*.
- S. Parthasarathy and C. J. Venkateswaran. (2012). Scheduling jobs using oppositional-GSO algorithm in cloud computing environment,” *Wireless Networks*, 1-11.
- S. Sethi, A. Sahu, and S. K. Jena. (2012). Efficient load balancing in cloud computing using fuzzy logic. *IOSR Journal of Engineering*, 2, 65-71.
- T. Erl, R. Puttini, and Z. Mahmood. (2013). *Cloud Computing: Concepts, Technology & Architecture*. 1st ed. Englewood Cliffs, NJ, USA:Prentice-Hall.
- Z. Zhang and X. Zhang. (2010). A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation. in *Industrial Mechatronics and Automation (ICIMA), 2010 2nd International Conference on*, 240-243.

استناد به این مقاله:

شناسه دیجیتال (DOI): 10.22091/jemsc.2018.1271

قرائیان، شبنم، امیری زاده، خسرو. (۱۳۹۸). «بهبودسازی زمانبندی وظایف در محیط ابر با استفاده از ویرایش فازی الگوریتم بهبودسازی اجتماع ذرات». *مدیریت مهندسی و رایانش نرم*، ۶ (۲)، ۱۹۹-۲۱۵.