International Conference on
RESEARCH IN ELECTRICAL AND COMPUTER ENGINEERING
کنفرانس بین‌المللی پژوهش در مهندسی برق و کامپیوتر
15 December 2016 Singapore www.receconf.com سنگاپور ۲۵ آذر ۱۳۹۵

زمان‌بندی وظایف توسط الگوریتم‌های اکتشافی مبتنی بر تعادل بار در محیط محاسبات ابری

سحر اورک پور
گروه کامپیوتر، واحد علوم و تحقیقات خوزستان، دانشگاه آزاد اسلامی، اهواز، ایران
گروه کامپیوتر، واحد اهواز، دانشگاه آزاد اسلامی، اهواز، ایران
Sahar.orakpour@yahoo.com

ابراهیم بهروزیان نژاد
گروه کامپیوتر، واحد شوشتر، دانشگاه آزاد اسلامی، شوشتر، ایران
E.Behrozian@iau-shoushtar.ac.ir

چکیده

الگوریتم‌های زمان‌بندی وظایف،نقش اساسی را در سیستم محاسبات ابری ایفا می‌کند.بارشد محاسبات ابری،سرویس‌گیرندگان،تقاضای سرویس‌های بیشتر با نتایج مطلوب‌تری رادارند.تعادل بار برای سیستم‌های ابری به‌عنوان یک زمینه‌ی جذاب و مهم تحقیقاتی شده است. الگوریتم‌های متنوعی جهت ایجاد مکانیسم‌های موثربرای تخصیص درخواست‌های کلاینت‌ها به گره‌های ابری، پیشنهادشده است. اغلب این الگوریتم‌ها معیارهایی همچون، نرخ بهره‌وری استفاده از منابع، توازن بار و لزوم پاسخ‌دهی سریع به درخواست‌ها رادرنظر نمی‌گیرند. در این تحقیق روشی جدید برای تعادل بار، با استفاده از الگوریتم‌های اکتشافی ترکیبی ژنتیک و زنبورعسل مصنوعی ، در محاسبات ابری ارائه داده‌ایم که باعث کاهش زمان اجرای وظایف و افزایش درصد بهره‌وری منابع می‌شود. به‌منظور متعادل کردن بار در محاسبات ابری با استفاده از این تکنیک، از سرعت پردازنده و میزان بار تخصیص داده‌شده به ماشین مجازی به‌عنوان پارامتر استفاده‌شده است. دریک سیستم توزیع‌شده مانند ابر،توزیع وظایف روی ماشین‌های مجازی بطورتصادفی است.این وضعیت باعث می‌شود که بار کل سیستم غیرمتعادل شودودر نتیجه کارایی کل سیستم افت پیدا کند.الگوریتم متعادل‌سازی بار،تلاش می‌کند تا زمان پاسخ کلیه‌ی وظایف راباستفاده از توزیع مناسب کارها بر روی پردازنده‌ها به حداقل برساند.

واژگان کلیدی: محاسبات ابری، تعادل بار، الگوریتم ژنتیک، الگوریتم زنبورعسل،زمان اجرا

۱- مقدمه

سیستم‌های محاسبات ابری به‌عنوان یک مدل محاسباتی که پس از ظهور محاسبات شبکه‌ای[۱] پدیدار شده است، در دنیای فناوری اطلاعات و اینترنت، بیش‌ازپیش مطرح می‌شود. این مدل که پس از Web 2.0 مطرح‌شده است، به‌منظور پردازش حجم نامحدودی از داده‌ها و اطلاعات، در سراسر اینترنت، با سرعت‌بالا و هزینه‌ی کم، ایجادشده است. قدرت عظیم پردازشی در این مدل، به دلیل وجود کامپیوترهای خوشه‌بندی‌شده، سرورهای مجازی شده، سیستم‌های توزیع‌شده و مجموعه‌ای قدرتمند از سخت‌افزارها و نرم‌افزارهای شبکه‌ای است. با تکنولوژی مجازی‌سازی، پلتفرم‌های ابری به شرکت‌ها و سازمان‌ها این امکان را می‌دهد که توان محاسباتی رادرقالب ماشین‌های مجازی به کاربران اجاره دهند. یکی از مسائل مهم مرتبط با این زمینه، تعادل بار پویا یا به‌عبارتی‌دیگر زمان‌بندی وظـــــایف است. زمان‌بندی کارها یک فرآیند کلیدی در IaaS[۲] است که هدف آن، اجرای درخواست‌های واردشده به سیستم بر روی منابع، به شیوه‌ای کارآمدبادرنظرگرفتن سایر خصوصیات محیط ابر می‌باشد؛ بنابراین برای زمان‌بندی کارهادرمحیط ابرنیازبه یک الگوریتم زمان‌بند کارا داریم. یک زمان‌بند کار مناسب، می‌بایست استراتژی زمان‌بندی‌اش راباتغییرات محیط و انواع کارها تطبیق دهد. دلایل متعددی باعث شده که این موضوع به‌عنوان یک مسئله‌ی NP- Complete نمود پیدا کند، ازجمله، ناهمگون بودن و پویایی خصوصیات منابع و درخواست هادرمحیط محاسبات ابری. در چنین سیستمی پروسه‌ی زمان‌بندی می‌بایست به‌صورت اتوماتیک وبسیارسریع انجام گیرد.(Zhao and Zhang, 2009)

در این پژوهش به‌منظور کاهش زمان تکمیل وظایف و بهبود محیط ابری از الگوریتم‌های ترکیبی ژنتیک و زنبورعسل مصنوعی با اعمال مکانیسم‌های مؤثر تعادل بار ،(LBGA-ABC[۳])استفاده‌شده است.در این روش مقادیر سرریز و زیر ریز منابع برحسب توان پردازشی هر ماشین مجازی محاسبه‌شده و در یک ماتریس نگهداری می‌شود،سپس مقادیر کم‌بار شناسایی‌شده و وظایف اضافی از ماشین‌های دیگر به ماشین کم‌بار تخصیص داده می‌شوند.برای شبیه‌سازی نتایج تحقیق از نرم‌افزار متلب استفاده‌شده است.نتایج ارزیابی نشان داد که این الگوریتم در مقایسه با الگوریتم‌های LBGA،GA وGA-ABCعملکرد بهتری را از خود نشان می‌دهد و نرخ بهره‌وری منابع[۴] را نیز تا حد مطلوبی افزایش می‌دهد.

۲- رده‌بندی الگوریتم‌های مبتنی بر تعادل بار

توازن بارعبارتند از توزیع مناسب و بهینه‌ی درخواست‌ها بر روی انواع منابع که جهت انجام عملیات کارادرمحیط‌های توزیع‌شده ضروری است.

الگوریتم‌های تعادل بار به دو رده‌ی ایستا و پویا طبقه‌بندی می‌شوند. الگوریتم‌های ایستا اغلب جهت محیط‌های همگن و پایدار مناسب هستندومی‌توانند نتایج قابل قبولی در این محیط‌ها تولید کنند. اگرچه آن‌ها معمولاانعطاف‌پذیر نیستند و نمی‌توانند با تغییرات پویای مشخصه‌های موجود در طول زمان اجرا منطبق شوند. الگوریتم‌های پویا انعطاف‌پذیرترند و انواع مختلفی از ویژگی‌های زمان اجرا را در نظر می‌گیرند.

---

[۱] Grid Computing

[۲] Infrastructure as a Service

[۳] Load Balanced Genetic Algorithm Artificial bee Colony

[۴] Resource Utilization Ratio

این الگوریتم‌ها می‌توانند با تغییرات سازگار شوند ودرمحیط‌های ناهمگن و پویا نتایج بهتری حاصل می‌شود؛ اما برخی مواقع این الگوریتم‌ها به دلیل ایجاد سربار سیستم منجر به عدم کارایی و تنزل سرا سری کارایی سرویس‌ها می‌شوند(Al Nuaimi et al,2012) .

همچنین به‌کارگیری تکنیک‌های تعادل بار بر روی کاهش زمان تکمیل آخرین کار[۵] نیز اثرگذاراست. برای حل مسائل زمان‌بندی کارها، اغلب از الگوریتم‌های تکاملی و مکاشفه‌ای گوناگونی در محیط‌های ابری استفاده‌شده است. در این پژوهش با توجه به اینکه الگوریتم ژنتیک بر اساس تجربه، علاوه بر سادگی محاسبه، دقت بیشتری در مسائل جستجو داشته می‌تواند به‌طور موازی بکار گرفته شود، اما برای تسریع یافتن مسیر پاسخ بهینه، می‌توان از الگوریتم‌های بهینه‌سازی محلی نیز استفاده کرد، ازاین‌رو در این پژوهش سعی خواهد شد یک الگوی مناسب زمان‌بندی ترکیبی از الگوریتم ژنتیک و الگوریتم کلونی زنبور مصنوعی، با به‌کارگیری مکانیسم‌های تعادل بارمیان منابع حین تخصیص وظایف، ارائه شود که به دلیل وجود تراکم بار منابع در هرلحظه در سیستم، زمان اجرای مجموعه کارهای نگاشت شده، را تا حد امکان به حداقل برساند.

۲-۱- چالش‌های موجود درزمینه ی تعادل بار

توزیع خاص گره‌های ابر: برخی الگوریتم‌های طراحی‌شده فقط برای اینترانت یا گره‌های مرتبط به هم که تا خیرات ارتباطی قابل‌چشم‌پوشی است، کارایی مؤثرتری دارند. البته طراحی یک الگوریتم تعادل بار برای این محیط‌ها خود یک چالش محسوب می‌شود زیرا باید مواردی همچون سرعت اتصالات شبکه‌ای بین گره‌ها، فاصله میان کلاینت و گره‌های پردازش کاروگره‌های تأمین‌کننده‌ی سرویس درنظرگرفته شود(Ghutke and Shrawankar,2014) .

تکثیروذخیره‌سازی: الگوریتم‌های تکثیر کامل، بهره‌وری موثرذخیره‌سازی را درنظرنمی‌گیرند. به دلیل اینکه داده‌های مشابه در همه‌ی گره‌های تکثیرشده ذخیره خواهد شد. این الگوریتم‌ها به دلیل نیاز به رسانه‌های ذخیره‌سازی، هزینه‌ی بیشتری را تحمیل می‌کنند.

اگرچه الگوریتم‌های تکثیرجزیی، می‌توانند بخشی از بانک داده‌هادرهرگره (با سطحی خاص از همپوشانی) را بر اساس قابلیت‌های هر گره مانند توان و ظرفیت پردازشی، ذخیره کنند(Ghutke and Shrawankar,2014).

پیچیدگی الگوریتم: الگوریتم‌های تعادل بار که برحسب پیاده‌سازی و عملیات، پیچیدگی کمتری دارنددراولویت هستند. پیچیدگی پیاده‌سازی بیشترمنجربه فرآیندهای پیچیده‌تری می‌شود که باعث کارایی منفی در برخی از مواردمی‌شود؛ بنابراین، بنابراین هنگامی‌که الگوریتم‌ها جهت نظارت و کنترل، نیازمند اطلاعات بیشترواررتباطات فراتری هستند، تأخیرهای ارتباطی می‌تواند منجر به افت کارایی شود؛ بنابراین این الگوریتم‌ها بایددرساده ترین شکل ممکن طراحی شوند(Ghutke and Shrawankar,2014).

نقطه‌ی شکست: کنترل تعادل بارو جمع‌آوری داده هادرباره‌ی گره‌ها، می‌بایست طوری طراحی شودکه ازبوجودآمدن نقطه‌ی شکست در الگوریتم اجتناب شود. برخی الگوریتم‌ها (الگوریتم‌های متمرکز) می‌توانند کارایی ومکانیزم های مؤثری را برای حل تعادل بار تحت الگوی مشخص، تأمین کنند. در چنین مواردی اگر گره‌ی کنترل‌کننده‌ی مرکزی با مشکل مواجه شد، کل سیستم از کار می‌افتد. هر الگوریتم تعادل بار باید بر این‌چنین چالش‌هایی غلبه کند. الگوریتم‌های تعادل بار توزیع‌شده معمولاً روش مناسب‌تری به در نظر می‌رسد، اگرچه پیچیدگی بیشتری دارند و برای عملکرد صحیح باید کنترل شوند(Ranjan et al,2010).

[۵] Make Span

۳- کارهای انجام‌شده

در (Fang et al,2010) از یک مکانیسم دوسطحی در رده‌ی ماشین مجازی و ماشین میزبان بر اساس تعادل بار پویا استفاده شد.بدین منظور از تکنیک مهاجرت ماشین مجازی استفاده‌شده است.

الگوریتم زمان‌بندی ترکیبی برمبنای دو روش اکتشافی ژنتیک و شبیه‌سازی تبریدی در(Guo-ning et al,2010) ارائه‌شده است.سپس برای یافتن راه‌حل بهینه،در ابتدا از الگوریتم ژنتیک جهت جستجوی سراسری و برای جستجوی محلی و افزایش سرعت همگرایی دررسیدن به پاسخ مطلوب،پس ازعملگرجهش،شبیه‌سازی تبریدی فراخوانی می‌شود.نتایج آزمایش‌ها نشان داد که کارایی تکمیل کارها افزایش‌یافته است.

(Li et al,2011) الگوریتم زمان‌بندی کارهای ابری را بر اساس بهبود تعادل باربر پایه‌ی الگوریتم کلونی مورچه‌ها،(LBACO) را پیشنهاد دادند.بخش اصلی این پژوهش متعادل کردن بار کل سیستم می‌باشد.درحالی‌که سعی می‌کند زمان اجرای کل مجموعه کارها رانیزبه حداقل برساند.در مقایسه با الگوریتم‌هایFCFS وACO،میانگین زمان اجرای کارها،کاهش‌یافته است و همچنین درجه‌ی عدم تعادل بار[۶] نیز کاهش‌یافته است.

در(Radojevic and Zagar,2011)مدل تصمیم‌گیری تعادل بار مرکزی را برای محیط ابری ارائه داده‌شده است. CLBDM بهبودیافته‌ی الگوریتم نوبت گردشی[۷] می‌باشد که بر اساس تعویض متن نشست هادرلایه ی کاربرد عمل می‌کند. CLBDMدر اینجا به‌عنوان ناظر عمل می‌کند.

(Nishant et al,2012)مدل زمان‌بندی کارهابرمبنای تعادل بارباستفاده از الگوریتم مورچگان را طراحی کردند.در این الگوریتم از اضافه کردن خاصیتی به نام عقب‌نشینی به مورچه‌ها[۸]،استفاده‌شده است.در این روش مورچه‌ها در طول مسیر گره‌های دارای اضافه‌بار را کشف می‌کنند.

در(Antony et al,2012)یک الگوریتم زمان‌بندی باقابلیت تحمل خرابی ارائه داده‌شده است.با توجه به اینکه پهنای باند درابر محدود است،برای صرفه‌جویی در آن از الگوریتمBAR استفاده می‌شود.که یک الگوریتم اکتشافی مبنی بر محلی بودن داده‌هاست.الگوریتم بهبودیافته‌یBAR برای اداره‌ی خطا وبروزخرابی در ماشین می‌باشد.در مقایسه با الگوریتم BARموجود،این روش پیشنهادی زمان تکمیل کارهارادرمواجه با خرابی به طرزموثری کاهش داد.

الگوریتم زمان‌بندیMin-Min آگاه از اولویت کاربر برای تعادل باردرمحاسبات ابری را ارائه نمودند(Chen et al,2013).سپس با توجه به سرویس انتخاب‌شده توسط کاربر،هزینه‌ی هرواحدمنبع،مشخص می‌شود.اولویت کاربران در الگوریتم پیشنهادی-PA LBIMMدرنظرگرفته شد.نتایج شبیه‌سازی نشان داد که این الگوریتم منجر به کارایی بالاوکاهش زمان تکمیل کل کارها می‌شود.

الگوریتم PACO (Sun et al,2013) از الگوریتم بهبود کلونی مورچه‌ها[۹] توسط استراتژی اولین دوره‌ی زمان‌بندی پیشنهادی و استراتژی بهبودبروزرسانی مقدارفرمون،در محاسبات ابری استفاده می‌کند.در مقایسه با الگوریتم Min-Min کارایی بیشتری راازخودنشان داده است.

[۶] Degree Imbalanced

[۷] Round-Robin

[۸] suicide

[۹] Ant Colony Optimization

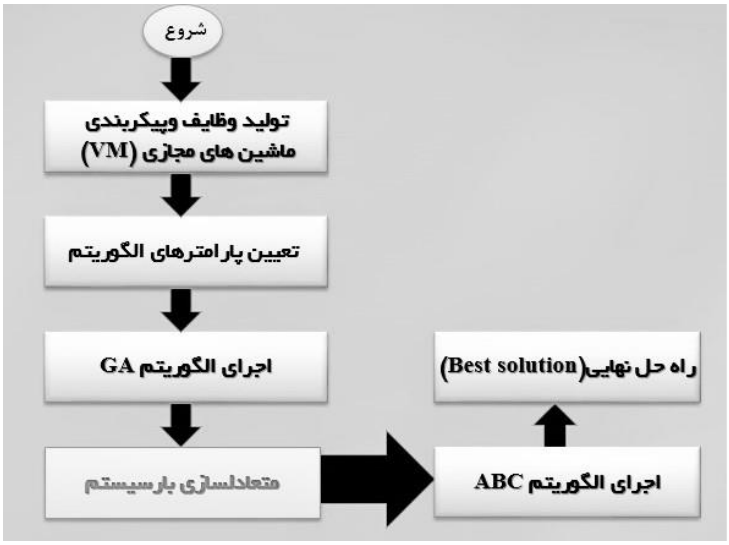
در الگوریتمECMM (Li et al,2014) از ویژگی‌های ابر انعطاف‌پذیر استفاده‌شده است.ایده‌ی اصلی شامل استفاده ازمولفه های جدول وضعیت وظیفه و جدول وضعیت ماشین مجازی است.در مقایسه باRR وMax-Min ،ECMMکمترین زمان پاسخ و Max-Min بالاترین زمان پاسخ رابدست آورد.

در الگوریتم TBSLB-PSO (Ramezani et al,2014) جهت برقراری تعادل بار،بجای مهاجرت کل ماشین مجازی با بار اضافه،فقط کارهای اضافی از آن ماشین انتقال داده می‌شود.مهاجرت این‌کارهای اضافی توسط الگوریتم بهبود ازدحام ذرات[۱۰] انجام می‌شود.آزمایش‌ها نشان داد زمان صرف شده برای فرایند تعادل باردرمقایسه با روش‌های سنتی به طرز قابل‌توجهی کاهش‌یافته است.

در این استراتژی(Haidri et al,2014) ،به‌محض پذیرش درخواست،متعادل‌کننده‌ی بار،آن را به ماشین مجازی قوی‌تر(n_k) اعزام می کندو مقدارn_k به‌روزرسانی می‌شود.هدف از ین به‌روزرسانی استفاده‌ی کامل از ظرفیت‌های ماشین مجازی می‌باشد.در مقایسه با الگوریتم‌هایRR،Min-Min،Max-Min ،LJFR-SJFR ،RR بدترین عملکرد و الگوریتم موردبحث بهترین عملکرد را دارد.

۴- الگوریتم پیشنهادی LBGA-ABC

الگوریتم پیشنهادی مبتنی بر جمعیت بوده و در یک فرآیند تکراری به راه‌حل بهینه (یا راه‌حل نزدیک به بهینه) دست می‌یابد. در این الگوریتم فرآیند جستجو با تولید یک جمعیت اولیه تصادفی آغاز می‌گردد. هر عضو از جمعیت بیان‌گر یک راه‌حل ممکن است که به‌صورت شکل۲ نشان داده می‌شود. در هر تکرار از الگوریتمGA، سه مرحله کلی وجود دارد: یکی ارزیابی شایستگی راه‌حل‌های تولیدشده، و دیگری متعادل‌سازی بار وظایف میان ماشین‌های مجازی درنهایت بروز رسانی جمعیت (تولید جمعیت جدید). این سه مرحله، پی‌درپی و به‌صورت تکراری اجرا می‌شوند، تا زمانی که شرط خاتمه ارضا شود. شرط خاتمه درروش پیشنهادی به اتمام رسیدن تعداد تکرارهای الگوریتم تعیین‌شده است. پس‌ازآن الگوریتم زنبورعسل مصنوعی برای رسیدن سریع‌تر به جواب بهینه اعمال می‌شود.روال کلی الگوریتم مطابق با شکل ۱ نشان داده می‌شود.



شکل ۱: روال الگوریتم LBGA-ABC

[۱۰] Particle Swarm Optimization

٤-١- نمایش راه‌حل‌ها(رمزگذاری)

با توجه به این‌که کارایی الگوریتم ژنتیک وابستگی زیادی به نحوه نمایش کروموزوم‌های آن دارد، در الگوریتم زمان‌بندی پیشنهادی، از اعداد طبیعی برای رمز کردن کروموزوم‌ها استفاده‌شده است. در الگوریتم پیشنهادی یک راه‌حل ممکن[11] برای مسئله با یک کروموزوم بیان می‌گردد. هر کروموزوم به‌صورت یک‌رشته از اعداد طبیعی به طول N در نظر گرفته می‌شود، که N تعداد کل وظایف مستقل موجود در مسئله است؛ به‌طوری‌که مقادیر درون ژن‌ها اعداد تصادفی بین ۱ تا M می‌باشند که M تعداد کل ماشین‌های مجازی (منابع محاسباتی)است.

جدول ۱: یک راه‌حل ممکن برای مسئله در الگوریتم پیشنهادی LBGA-ABC

A Solution:	T_1	T_2	T_3	T_4	T_5	T_6	T_7	. . .	T_N
	R_4	R_1	R_3	R_1	R_2	R_4	R_3	. . .	R_1

٤-٢- تولید جمعیت اولیه

تولید راه‌حل‌های اولیه مناسب نقش بسزایی در سرعت همگرایی و دقت الگوریتم دارد. در الگوریتم پیشنهادی یک عدد تصادفی بین ۱ تا M که نشان‌دهنده شماره ماشین مجازی است تولید می‌شود تا وظیفه موردنظر بر روی آن اجرا شود. به‌عبارت‌دیگر هر وظیفه به‌طور کاملاً تصادفی به یکی از منابع محاسباتی اختصاص می‌یابد. در هر تکرار پس از به‌روزرسانی جمعیت، راه‌حل‌های تولیدشده مورد ارزیابی قرار می‌گیرند. این ارزیـــابی بر اساس بخش ۴-۳ انجام می‌گیرد .

برای ارزیابی یک عضو از جمعیت (یک راه‌حل برای مسئله)، ابتدا وظایف اختصاص‌یافته به هر ماشین مجازی در نظر گرفته می‌شود؛ سپس زمان اجرای هر وظیفه بر روی ماشین مربوطه محاسبه می‌گردد.

٤-٣- تابع ارزیابی زمان پاسخ

اولین هدف الگوریتم ما، به حداقل رساندن زمان تکمیل وظایف یا طولانی‌ترین زمان اتمام وظایف در میان همه پردازنده‌ها (منابع) در سیستم است. توجه داشته باشید که این زمان همیشه باید کوچک‌تر یا مساوی ماکزیمم مهلت زمانی(MD[12]) در میان همه‌ی وظایف باشد .فرض کنید L_i و PS_j به ترتیب نشان‌دهنده اندازه وظیفه i(طول دستورالعمل) و سرعت پردازش ماشین j باشند .پس زمان اجرای وظیفه i بر روی منبع j را می‌توان از طریق رابطه (۱) به دست آورد:

$$T_{exe}(i,j)=\frac{Li}{PS_j} \qquad (۱)$$

مهم‌ترین هدف زمان‌بندی وظیفه این است که بتوان زمان تکمیل وظایف را مینیمم کرد. Completion Task Timeبرابر است بازمان کلی که موردنیاز است تا همه وظایف ورودی، اجر ای‌شان را به اتمام برسانند. درروش پیشنهادی برای حل مسئله‌ی زمان‌بندی وظیفه، کروموزومی مناسب‌تر می‌باشد که Make span آن مینیمم باشد.

$$Cost=Min\{Makespan=Max_{i\in Tasks}RuningTime_i\} \qquad (۲)$$

بنابراین Make span سیستم را می‌توان با استفاده از معادله (۳) محاسبه نمود .که در آن Aj مجموعه‌ای از وظایف هستند که به منبع j اختصاص داده‌شده‌اند.

$$T_{Complete}(j)=\frac{\sum_{k\in A}Task\ lenght_k}{processor\ Speed_j} \qquad 1\le j\le m \qquad (۳)$$

$$Makespan(\alpha)=Max\{T_{Complete}(j)\} \qquad 1\le j\le m \qquad (۴)$$

[11] Feasible Solution

[12] Maximum Deadline

Processor Speed به‌صورت یک ماتریس M×۱ در نظر گرفته می‌شود که M تعداد ماشین‌های مجازی می‌باشد. مقدار درایه j این ماتریس نشان‌دهنده‌ی سرعت پردازشگر منبع jاُم است.

در هر تکرار پس از به‌روزرسانی جمعیت، راه‌حل‌های تولیدشده مورد ارزیابی قرار می‌گیرند. این ارزیابی بر اساس تابع ارزیابی ارائه‌شده در رابطه (۲) انجام می‌گیرد.

٤-٤- تابع ارزیابی هزینه

هدف دوم الگوریتم ما، قیمت کل است که می‌بایست به حداقل رسانده شود. فرض کنید P_j بیانگر قیمت واحد برای اجرای وظیفه‌ی iاُم روی ماشین jاُم می‌باشد. بنابراین هزینه اجرای وظیفه jاُم با استفاده از معادله (۳) محاسبه نمود:

$$Cost\ (i, j) = T_{exe}(i,j) \times P_j \qquad (۵)$$

بدین ترتیب هزینه کلی یک‌راه حل، که بیانگر هزینه ناشی از یک کروموزوم در جمعیت است را می‌توان با استفاده از معادله (۵)بیان نمود .که در آن $Cost(j)$ هزینه ماشین j و $Cost(\alpha)$ هزینه یک راه‌حل است.

$$Cost(j)=T_{Complete}(j)\times p_j \qquad (۶)$$

$$Cost(\alpha)= \sum_{j=1}^{m} cost(j) \qquad 1\leq j\leq m \qquad (۷)$$

٤-٥- ساختار تابع برازش[۱۳]

درروش پیشنهادی، کاربر هنگام معرفی کارهای خود می‌تواند وزن‌های زمان اتمام و هزینه‌ی اجرای کارها (W_t و W_c) را مشخص کند .این دو وزن در محدوده [0,1] بوده و مجموع آن‌ها 1 است .مثلاً اگر برای یک کار W_c برابر ۰,۶ باشد به این معناست که کاربر ۶۰ درصد نگران هزینه مالی اجرا و ۴۰ درصد نگران زمان اتمام آن کار است .ضرایب وزن، توسط کاربر در حین کار مشخص می‌شوند W_t و W_c باعث می‌شوند که کاربر آزادی بیشتری در معرفی کارهای خود به منابع محاسباتی داشته باشد .تابع برازش یک کروموزوم را می‌توان از طریق معادله(۶) محاسبه نمود:

$$\text{Fitness Function}=w_c \times \frac{cost(\alpha)}{Budget} + w_t \times \frac{Makespan(\alpha)}{MD} \qquad (۸)$$

که در آن $cost(\alpha)$ و Budget به ترتیب بیانگر هزینه کلی راه‌حل و هزینه پیشنهادشده توسط کاربر در سیستم زمان‌بند ابری برای پردازش کل وظایف است و $Makespan(\alpha)$ نشان‌دهنده طولانی‌ترین زمان اتمام وظایف در میان همه منابع سیستم در یک کروموزوم و MD بیانگر ماکزیمم مهلت زمانی اتمام وظایف است .

٤-٦- ساختار عملگر انتخاب، ترکیب و جهش درروش پیشنهادی LBGA-ABC

- **عملگر انتخاب**

در الگوریتم پیشنهادی از عملگر انتخاب مسابقه‌ای[۱۴] استفاده‌شده است. این روش یکی از کاراترین روش‌های انتخاب والدین[۱۵] است و نسبت به سایر روش‌های مشابه مانند انتخاب متناسب با برازش[۱۶] انتخاب مبتنی بر پاداش[۱۷] فواید بیشتری دارد،ازجمله:کد نویسی مؤثر،قابلیت به‌کارگیری بر روی کامپیوترهای دارای معماری موازی و همچنین قابلیت تطبیق فشار انتخاب[۱۸]

[۱۳] Fitness Function
[۱۴] Tournament Selection
[۱۵] Parent
[۱۶] Fitness Proportionate Selection
[۱۷] Reward-based Selection
[۱۸] Selection Pressure

- **عملگر ترکیب**

برای عملیات تقاطع از عملگر ترکیب دونقطه‌ای استفاده‌شده است.

- **عملگر جهش**

در مرحله جهش، پس از انتخاب یک کروموزوم در مرحله قبل، یک ژن به‌صورت تصادفی انتخاب و مقدار فیلد منبع آن با یک عدد تصادفی بین ۱ تا m تغییر می‌کند. هدف اصلی از اعمال این نوع جهش تعویض پارامتر منبع یک وظیفه است تا در این صورت یک وظیفه بتواند در منابعی که بهتر هستند مورد پردازش قرار بگیرد. شکل ۳ اعمال عملگر جهش بر روی یک کروموزوم را نشان می‌دهد.

جدول ۲- مثالی از اعمال عملگر جهش در الگوریتم پیشنهادی

کروموزوم اولیه:

T1	T2	T3	T4	T5	T6
VM_2	VM_3	VM_4	VM_1	VM_2	VM_4

کروموزوم جهش‌یافته:

T1	T2	T3	T4	T5	T6
VM_2	VM_3	VM_2	VM_1	VM_2	VM_4

٤-٧- ارزیابی به‌کارگیری تکنیک تعادل بار در مسئله

در این روش برای ارزیابی تعادل بار از محاسبه‌ی ظرفیت پردازشی ماشین‌های مجازی استفاده‌شده است.ابتدا با توجه به‌سرعت پردازش هر یک از منابع تعداد کارهایی را که باید به آن‌ها اختصاص دهیم را مشخص می‌نماییم.این مقادیر دریک ماتریس 1*M ذخیره می‌شود که M در اینجا تعداد ماشین‌های مجازی را نشان می‌دهد.نحوه‌ی محاسبه در رابطه‌ی (۹) و(۱۰) نشان داده‌شده است :

$$num_{Capacity(i)} = \frac{processor\ Speed(i)}{\sum_{j=1}^{num\_resources} processor\ Speed(j)} * num\_jobs \qquad (۹)$$

$$num\_Capacity\ (i) = fix\ (num\_Capacity\ ) \qquad (۱۰)$$

ظرفیت پردازشی منابع در مسئله به‌صورت عدد صحیح می‌باشد.

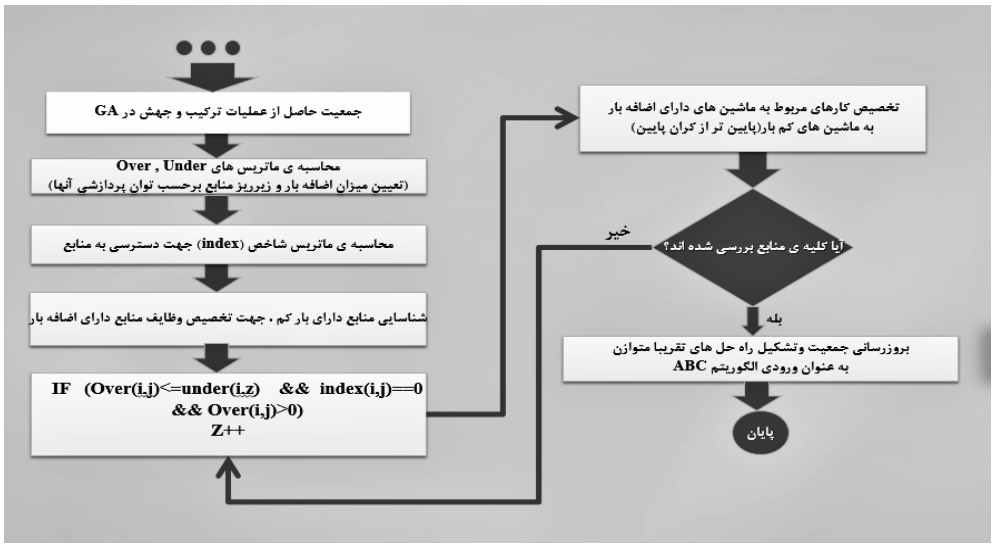
برای هرکدام از منابع یک کران بالا کران پایین تعداد کارهای اختصاص‌یافته را تعریف می‌نماییم،که در روابط (۱۱)و (۱۲) نشان داده‌شده است.

$$max\_num = num\_Capacity + 5 \qquad (۱۱)$$

$$min\_num = num\_Capacity - 5 \qquad (۱۲)$$

در حلقه مربوط به الگوریتم ژنتیک به صورتی کارها را به ماشین‌ها اختصاص می‌دهیم تا دریک محدوده‌ی خاص قرار بگیرند،مطابق با روابط (۱۱) و (۱۲).

در غیراینصورت تعداد وظایف سرریز محاسبه‌شده و دریک ماتریس با اندازه‌ی n pop*num_Resourec به نام Over ذخیره می‌شودوتعداد ظرفیت‌های موجود برای منابع دارای بار پایین‌تر از حد مجاز دریک ماتریس با اندازه‌ی n pop*num_Resource به نام under ذخیره و درنهایت کارهای مربوط به ماشین‌هایی که دارای تعداد کار اضافی هستند را به ماشین‌هایی که دارای کار پایین‌تر از کران پایینشان هستند اختصاص می‌دهیم.روال الگوریتم اعمال تعادل بار در شکل ۲ نشان داده‌شده است.



شکل ۲- عملکرد تکنیک تعادل بار در الگوریتم پیشنهادی

٤-۸- پارامترهای الگوریتم زنبورعسل مصنوعی

در الگوریتم پیشنهادی از پاسخ‌های بهینه با اعمال تکنیک تعادل بار در حلقه‌ی الگوریتم ژنتیک برای بهبود الگوریتم زنبورعسل مصنوعی استفاده‌شده است. الگوریتم زنبورعسل در جستجوی محلی عملکرد خوبی دارد؛ اما در جستجوی سراسری ضعیف عمل می‌کند. درعین‌حال الگوریتم ژنتیک در جستجوی سراسری عملکرد خوبی دارد. استفاده از خروجی الگوریتم ژنتیک در الگوریتم زنبورعسل به‌عنوان پاسخ‌های اولیه، موجب می‌شود تا ضعف عملکرد الگوریتم زنبورعسل در جستجوی سراسری برطرف شود و سرعت همگرایی و یافتن جواب بهینه در آن بهبود یابد. به کار بُردن تکنیک تعادل بار باعث می‌شود که فاز جستجوی کل فضای راه‌حل[۱۹] سریع‌تر انجام گیرد، و سبب می‌شود که راه‌حل بهینه با استفاده از یک جستجوی محلی تکراری توسط الگوریتم زنبورعسل پیدا شود. این پروسه برای هر نسل از جمعیت تکرار می‌شود.

در الگوریتم زنبورعسل، موقعیت هر زنبور در فضای N- بعدی بیان‌گر یک راه‌حل ممکن برای مسئله بهینه‌سازی است، که N تعداد متغیرهای بهینه‌سازی (تعداد کل وظایف) را بیان می‌کند.

- **جمعیت اولیه:** در الگوریتم‌های ابتدایی زنبورعسل مصنوعی،راه‌حل‌های اولیه توسط زنبورهای کارگر[۲۰]،شناسایی و ارزیابی می‌شوند.در این پژوهش ،به دلیل اینکه الگوریتم ژنتیک در جستجوهای محلی کند عمل می‌کند،راه‌حل‌هایی که ازنظر طول کلی زمان‌بندی مشابه خروجی الگوریتم LBGAهستند، به الگوریتم ABC داده می‌شوند تا یک‌راه حل همسایه برای آن‌ها تولید شود.

[۱۹] Exploration
[۲۰] Employed bees

- **فاز تولید راه‌حل جدید توسط زنبورهای کارگروارزیابی آن‌ها**:در الگوریتم ABC می‌توان به تعداد اعضای جمعیت (npop)،راه‌حل‌های متفاوت داشته باشیم.این راه‌حل هابه اندازه‌ی همان کلونی‌های موجود در الگوریتم(Colony Size) می‌باشد که یک زنبور برای پیدا کردن بهترین کلونی و ارزیابی آن، آن‌ها را وارسی می‌کند. کیفیت هر یک از راه‌حل‌ها با استفاده از رابطه (۴-۵) مورد ارزیابی قرار می‌گردد که بر مبنای زمان پایان کار طراحی گردیده است. ابتدا اولین جمعیت(راه‌حل) به‌دست‌آمده از مرحله‌ی قبل انتخاب می‌شود،سپس Kامین عضو غیرتکراری از جمعیت توسط زنبور کارگر انتخاب‌شده و برای ترکیب طبق رابطه‌ی (۱۳)و(۱۴) راه‌حل همسایه‌ی مناسب با آن به دست می‌آید و توسط فرمول(۴-۵) ارزیابی می‌شود.

$$Newbee.newsolution_{i,j}=Newbee_{i,j}+\emptyset_{i,j}\left(Newbee_{i,j}-Newbee_{k,j}\right) \quad (۱۳)$$

$$\emptyset_{i,j} \sim u(0,+1) \quad (۱۴)$$

اگر راه‌حل موردنظر باعث بهبود درمساله نشود،آن کلونی(عضو iام جمعیت)جریمه می‌شود.برای جریمه‌ی کلونی موردنظر،پارامتر جریمه (L)یک واحد افزایش یابد.در صورت رسیدن به یک حد آستانه،آن کلونی پاک می‌شود.

- **فاز زنبورهای جستجوگر**:در این مرحله ، هر زنبورعسل جستجوگر غذا، راه‌حل موردنظرش را با رقص ارتعاشی نشان می‌دهد و با این روش برخی از زنبورهای دیگر را به استفاده از راه‌حلی که پیداکرده ترغیب می‌نماید.به‌طورکلی عملیات محاسبه‌ی احتمالات(p_i)،انتخاب کلونی، تولید پاسخ جدیدومحاسبه ی آن در این مرحله صورت می‌گیرد.در این روش احتمال جذب هر زنبورعسل کارگر توسط زنبورهای جستجوگر با استفاده از رابطه‌ی بولتزمان[۲۱] طبق (۱۵) به دست می‌آید.

$$p_i=\exp\left(\frac{Cost_i}{\frac{\sum_{i=1}^{n} Cost}{n}}\right) \quad (۱۵)$$

سپس از قاعده‌ی انتخاب چرخ رولت،برای شناسایی تصادفی و هدفمند بهترین راه‌حل ممکن استفاده می‌شود.استفاده از تکنیک انتخاب چرخ رولت(RWS)،بجای انتخاب تصادفی،باعث همگرایی سریع‌تر در جستجوی فضای حل مسئله می‌شود.خروجی روال چرخ رولت به‌عنوان راه‌حلی قابل‌بهبود توسط زنبورهای جستجوگر شـــــــــناسایی می‌شود.ارزیابی‌ها و جایگزینی راه‌حل‌های مناســـــب پس‌ازآن طبق روابط (۱۳)و(۱۴) انجام می‌شود.کلونی‌ها(راه‌حل)هایی که پارامتر حد(L) برای آن‌ها به صفر رسیده است،توسط زنبورهای پیشاهنگ، با راه‌حل‌های تصادفی جدید،جایگزین می‌شوند.در هر مرحله از ارزیابی‌ها،زنبورعسل بهترین راه‌حل را به‌عنوان بهینه‌ترین راه‌حل ممکن به‌روز می‌کند.عملکرد زنبورهای پیشاهنگ مطابق با شبه کد شکل ۳ می‌باشد.

```
 for i=1:nPop
        if C(i)>=L

pop(i).Position=unifrnd(VarMin,VarMax,VarSize);

pop(i).Cost=CostFunction(pop(i).Position);
            C(i)=0;
        end
    end
```

شکل ۳- شبه کد فاز زنبورهای پیشاهنگ

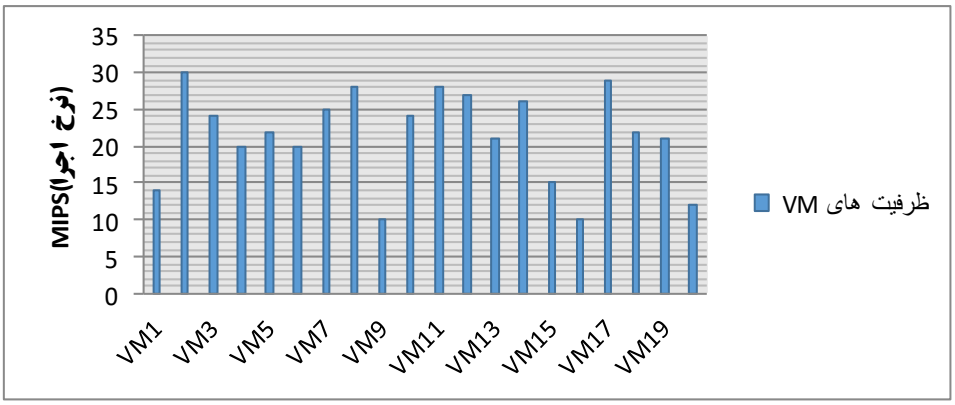
[۲۱] Boltzmann

۵- نتایج شبیه‌سازی

یک الگوریتم خوب و قابل‌قبول باید بتواند نتایـــج قابل قبولی را در شرایط گوناگون مانند ناهمگونی وظایف و منابع به کاربر ارائه کند. برای ارزیابی کارایی و بهره‌وری منابع، الگوریتم پیشنهادی،با الگوریتم‌های LBGA،GA و GA-ABCدر دو وضعیت تعداد وظایف متغیر و تعداد ماشین‌های مجازی متغیر با شرایط یکسان محیط پیاده‌سازی مورد مقایسه قرارگرفته است.نتایج آزمایش‌ها نشان می‌دهد که الگوریتم پیشنهادی LBGA-ABC معیار بیشترین زمان اجرای کلیه‌ی وظایف را به طرز قابل‌توجهی کاهش می‌دهد و شاخص تعادل بار که با اندازه‌گیری درصد بهره‌وری منابع مشخص می‌شود نیز افزایش‌یافته است. کلیه آزمایش‌ها بر روی سیستمی با مشخصات پردازنده‌ی ۲,۶۷ گیگاهرتز، حافظه‌ی RAM ۶ گیگابایت و ویندوز هفت انجام‌گرفته است. برای شبیه‌سازی از محیط و زبان برنامه‌نویسی MATLAB استفاده‌شده است.

۵-۱- پیکربندی ماشین‌های مجازی

در اینجا از یک پیکربندی نمونه‌ی ماشین‌های مجازی بانام VM، برای منابع محیط ابری، در آزمایش‌ها استفاده‌شده است. خصوصیات منابع در VM،طوری است که بتوان زمان‌بندی‌های مختلف را به‌خوبی مقایسه نمود. همه‌ی منابع دارای یک پردازنده هستند.نرخ اجرای ماشین‌های مجازی برحسب میلیون دستورالعمل در ثانیه(MIPS[۲۲]) می‌باشد. این پیکربندی را می‌توان در شکل۴ مشاهده کرد.



شکل۴-توانمندی ماشین‌های مجازی(VM)

۵-۲- بهینه‌سازی زمان با تعداد وظایف متغیر

الگوریتم پیشنهادی با چندین الگوریتم زمان‌بندی دیگر، با توجه به شرایط جدول ۳، مورد آزمایش قرارگرفته و برای بررسی مناسب طول وظایف، در آزمودن، تمامی مدل‌ها برابر در نظر گرفته‌شده است.

جدول۳- شرایط آزمایش بهینه‌سازی زمان با تعداد وظایف متغیر

پارامتر	الگوریتم	تعداد کاربر	تعداد وظایف	محدوده طول وظایف	پیکربندی منابع	تعداد تکرار
مقدار	متغیر	۱	متغیر	[۱۰-۳۰]	VM	۱۰۰۰

حال شرایط بالا برای ارزیابی الگوریتم‌های مختلف مورداستفاده قرار خواهد گرفت.

[۲۲] Million Instruction Per Second

الگوریتم پیشنهادی تحت شرایط جدول ۳ با الگوریتم‌های(GA)، (LBGA) و(GA-ABC) مورد ارزیابی قرار گرفت. قبل از بررسی نتایج لازم است که مقادیر اولیه پارامترهای به کار گرفته‌شده در الگوریتم پیشنهادیLBGA-ABC مشخص شوند. این مقادیر در جدول‌های ۴ و ۵ آورده شده است.

جدول۴- مقادیر اولیه‌ی پارامترها در الگوریتم ژنتیک

پارامتر	نرخ جهش	نرخ ترکیب
مقدار	۰/۰۰۰۴	۰/۵۰

جدول۵- مقادیر اولیه‌ی پارامترها در الگوریتم زنبورعسل

پارامتر	سایز کلونی (n_{pop})	تعداد زنبورهای جستجوگر (β)	پارامتر جریمه(L)	حد بالای ضریب تسریع(a)
مقدار	۵۰	۵۰	۲۰	۱

شکل۵ نموداری را نشان می‌دهد که در آن تعداد وظایف بین ۱۰۰۰ و ۵۰۰۰ بر روی مدل منبع VMبا استفاده از الگوریتم‌های موردنظر زمان‌بندی‌شده‌اند.

شکل۵- میزان Make span در درخواست‌های متفاوت

همان‌طور که مشاهده می‌شود، هر چه تعداد وظایف ورودی بیشتر می‌شود، Make span نیز افزایش می‌یابد. در بین الگوریتم‌های زمان‌بندی، مشاهده می‌شود که الگوریتم پیشنهادی نسبت به بقیه Make span کمتری را تولید می‌کند. در این

آزمایش مشخص می‌گردد که هر چه تعداد وظایف ورودی بیشتر باشد،با اعمال مکانیسم‌های تعادل بار، اختلاف‌زمان Makespan بین الگوریتم پیشنهادی و دیگر الگوریتم‌ها بیشتر می‌شود.

۵-۳- بهینه‌سازی با تعداد ماشین مجازی متغیر

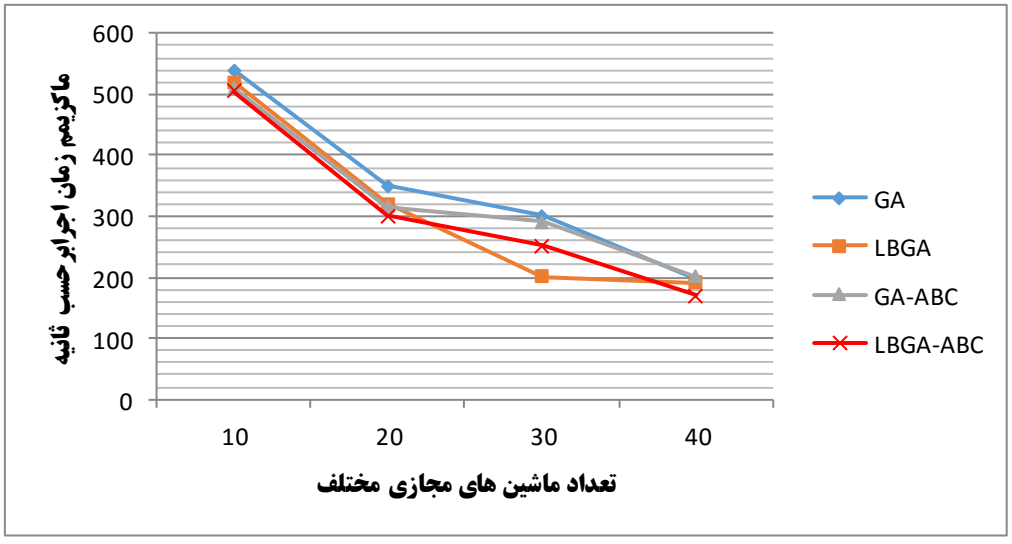
الگوریتم پیشنهادی تحت شرایط جدول ۶ با الگوریتم‌های ژنتیک (GA)،LBGA وGA-ABC مورد ارزیابی قرار گرفت و برای بررسی مناسب تعداد منابع، در تست، تمامی مدل‌ها برابر در نظر گرفته‌شده است.

پارامتر تعداد منابع، نشان‌دهنده‌ی تعداد ماشین‌های مجازی می‌باشد. هرکدام از این ماشین‌های مجازی دارای یک واحد پردازشگر می‌باشند که بر اساس استراتژی‌های موجود در الگوریتم‌های زمان‌بندی به وظایف ورودی در سیستم تخصیص داده می‌شوند. پارامتر تعداد تکرارها نشان می‌دهد که برای به دست آوردن زمان اجرای برنامه با استفاده از الگوریتم‌های موجود، تعداد ۱۰۰۰ تکرار انجام‌شده است و سپس میانگین مقادیر برای بررسی انتخاب می‌شود.

جدول۶- شرایط آزمایش بهینه‌سازی زمان با تعداد منابع متفاوت

پارامتر	الگوریتم	تعداد کاربر	تعداد وظایف	محدوده‌ی طول وظایف	تعداد منابع	پیکربندی منابع	تعداد تکرار
مقدار	متغیر	۱	۵۰۰۰	[۱۰-۳۰]	متغیر	VM	۱۰۰۰

مقادیر اولیه پارامترهای به کار گرفته‌شده در الگوریتم پیشنهادی LBGA-ABC طبق مقادیر درج‌شده در جدول‌های ۴ و ۵ می‌باشند.شکل۶ نموداری را نشان می‌دهد که در آن تعداد منابع بین ۱۰ و ۴۰ بر روی مدل منبع VMبا استفاده از الگوریتم‌های موردنظر زمان‌بندی‌شده‌اند.



شکل۶- میزان Makespan با تعداد ماشین مجازی متفاوت

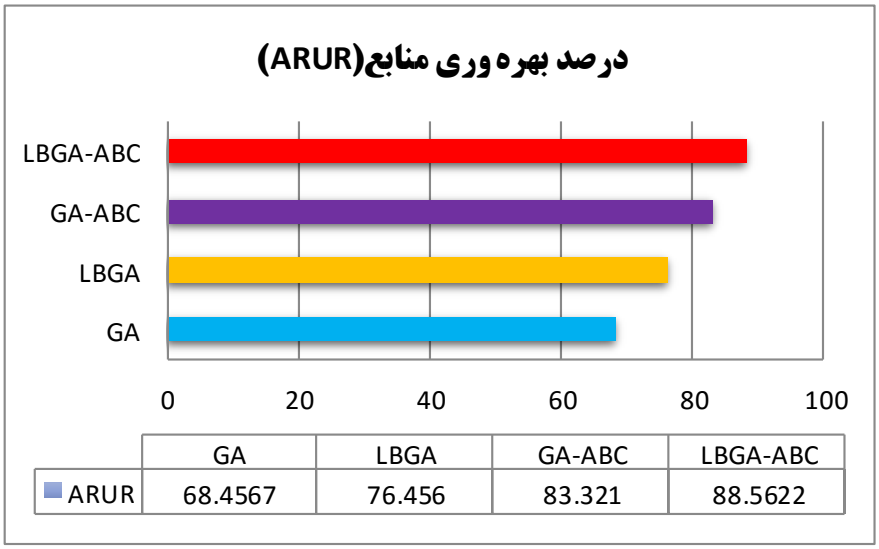
همان‌طور که در نمودار مشاهده می‌شود، تغییرات زمان در اثر افزایش تعداد منابع در هرکدام از الگوریتم‌ها روند متفاوتی دارد و با افزایش تعداد منابع، زمان اجرای الگوریتم‌ها کاهش می‌یابد. همان‌طور که از روی شکل مشخص است، الگوریتم پیشنهادی نسبت به دیگر الگوریتم‌ها کارایی بهتری را به دست آورده است.

۵-٤- ارزیابی بهبود تعادل باردرالگوریتم پیشنهادی

برای محاسبه‌ی بهره‌وری ماشین‌های مجازی از رابطه‌ی (۱۶) استفاده‌شده است که نشان‌دهنده‌ی نسبت میانگین بهره‌وری کل منابع [۲۳]می‌باشد(Chen et al,2013).

$$ARUR=\frac{\sum_{j=1}^{m} rt_j / m}{Makespan} * 100\% \qquad (۱۶)$$

که در این رابطه rt_j زمان حضور هر منبع[۲۴] بعدازفرایند زمان‌بندی وm نشان‌دهنده‌ی تعداد منابع سیستم می‌باشد. مقدار ARUR در محدوده‌ی ۰ و ۱ می‌باشد. اگر این مقدار ۱ باشد، بهترین و کاراترین سطح تعادل بار به دست می‌آید. بنابراین هرچه مقدار ARUR به ۱ نزدیک‌تر باشد، الگوریتم زمان‌بندی بهترین عملکرد را خواهد داشت.



شکل ۷- میزان درصد بهره‌وری منابع در الگوریتم‌های مختلف

در شکل ۷ نسبت بهره‌وری منابع در الگوریتم‌های مختلف مورد آزمایش نسبت به الگوریتم پیشنهادی LBGA-ABC مورد مقایسه قرارگرفته است. همان‌طور که مشخص است، الگوریتم پیشنهادی در شرایط یکسان آزمایش، با میزان ۸۸،۵۶۲۲ درصد، نسبت به سایر روش‌ها، عملکرد مطلوب‌تری داشته است.

[۲۳] ARUR(Average Resource Utilization Ratio)

[۲۴] Ready time

۶- بحث و نتیجه‌گیری

در این تحقیق روش تعادل بار مبتنی بر الگوریتم‌های اکتشافی در محیط محاسبات ابری ارائه‌شده است. درروش پیشنهادی، در حلقه‌ی الگوریتم ژنتیک ما تکنیک تعادل بار را برمبنای محاسبه‌ی ظرفیت پردازشی منابع و تخصیص وظایف به منابع دارای بار کم، اعمال می‌شود. بر اساس این تحقیق ما به این نتیجه رسیدیم که با انتخاب ماشین فیزیکی مؤثر، تمام عملکرد محیط آزمایش تحت تأثیر قرار می‌گیرد.

تاکنون روش‌های بسیاری برای تعادل بار در محاسبات ابری پیشنهادشده است. تعداد زیادی از این روش‌ها سعی بر این داشتند که با کمترین زمان پاسخ، منبع موردنظر را کشف کنند. روش‌های پیشین تعادل بار، زمان پاسخ بهینه و هزینه پایینی، در اختیار کاربران قرار نداده‌اند. الگوریتمی که در این پایان‌نامه ارائه‌شده است به‌منظور کاهش زمان پاسخ و توازن بازطراحی شده است.ما کارایی روش خود را با لحاظ کردن پارامترهای متعدد مورد آزمایش قراردادیم و برای اینکه کارایی روش پیشنهادی مشخص شود، نتایج به‌دست‌آمده را با سایر الگوریتم‌های زمان‌بندی، مانند GA-ABC،LBGA،GA، مقایسه کردیم. نتایج زیر از آزمایش به دست آمدند:

۱) درروش پیشنهادی توزیع بار بین ماشین‌های فیزیکی به‌صورت تقریباً یکنواخت ایجادشده است و تعادل بار به‌صورت سراسری اعمال‌شده است.
۲) بیشترین زمان پاسخ در الگوریتم LBGA-ABC نسبت به دو روش دیگر کاهش چشمگیری پیداکرده است.
۳) درصد بهره‌وری منابع در الگوریتم پیشنهادی نسبت به سایر الگوریتم‌های مورد مقایسه به میزان ۸۸,۵۶۲۲ درصد افزایش پیداکرده است.

در آینده ما می‌توانیم از جهات بسیاری الگوریتم پیشنهادی خود را بهبود دهیم.ازجمله:لحاظ کردن مسئله زمان انتقال و فاصله درروش پیشنهادی و بررسی اینکه آیا الگوریتم تعادل بار تصمیمات بهتر و بهینه‌تری را در انتخاب و کشف منابع برای کاربران اتخاذ می‌کند یا خیر.در نظر گرفتن بار درخواستی کاربر و لحاظ کردن آن در تصمیمات مربوط به ارسال درخواست،لحاظ کردن سطح اجرای وظایف خاص و معمولی بر اساس تصمیمات کاربر.به‌کارگیری تکنیک پیشنهادی در سایر الگوریتم‌های اکتشافی مانند PSO، FA و...

برای عملی کردن روش پیشنهادی، می‌توان آن را در محیط ماشین مجازی واقعی مثل Xen Hypervisor اجرا کرد. استفاده از فن‌های فازی برای تعریف سطح بار قابل‌قبول در سیستم می‌تواند برای پژوهش‌های آتی مورداستفاده قرار گیرد.

۷- منابع

zhao.Chenhong, zhang,Shanshan, Liu .Qingfeng and Xie.Jian.(2009) .Independent Task Scheduling based On Genetic Algorithm in Cloud Computing. 5th IEEE International Confference on Wireless Communication, Networking and mobile Computing,1-4

Al Nuaimi. Klaithem, Mohamed.Nader, Al Nuaimi.Mariam and Al-Jaroodi.Jameela.(2012). A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms. IEEE Second Symposium on Network Cloud Computing and Applications (NCCA). 137-142

Ghutke. Bhushan and Shrawankar.Urmila.(2014). Pros and cons of load balancing algorithms for cloud computing, IEEE Internatinal Conference Information Systems and Computer Networks (ISCON).123-127

Ranjan.Rajiv, zhao.Liang, Wu.Xiaomin, liu.Anna, Quiroz.Andres and Parashar.Manish.(2010). Peer-to-Peer Cloud Provisioning: Service discovery and Load balancing.Principles ,System and Applications.195-217

Fang.Yiqiu, Wang.Fie and Ge.Junwie,(2010) .A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing. Springer US.Web Information Systems and Mining. 271-277

Guo-ning.Gan, Ting-lie.Huang and Shuai.Shuai.(2010). Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment.IEEE, Intelligent Computing and Integrated Systems (ICISS).60-63

Li.Kun,Xu.Gaochao,Zhao.Guangyu,Dong.Yushuang and Wang.Dan.(2011). Cloud Task scheduling based on Load Balancing Ant Colony Optimization,IEEE,Sixth Annual ChinaGrid Conference.3-9

Radojevic.Branko and Zagar.Mario.(2011). Analysis of issues with load balancing algorithms in hosted (cloud) environments. IEEE Proceedings of the 34th International Convention on MIPRO.416-420

Nishant.Kumar , Sharma.Pratik , Krishna.Vishal ,Gupta.Chhavi ,Singh.Kuwar Pratap , Nitin and Rastogi .Ravi.(2012). Load Balancing of Nodes in Cloud Using Ant Colony Optimization. IEEE In proc 14th International Conference on Computer Modelling and Simulation (UKSim). 3-8

Antony.Simy,Antony.Soniya,Beegom A S and Rajasree M S.(2012). Task Scheduling Algorithm with Fault Tolerance for Cloud.IEEE, International Conference on Computing Sciences.180-182

Chen.Huankai, Wang.Frank, Helian.Na, Akanmu.Gbola.(2013). User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing. IEEE National Conference on Parallel Computing Technologies.1-8

Sun.Weifeng, Zhang.Ning, Wang.Haotian and Yin.Wenjuan.(2013).PACO: A Period ACO Based Scheduling Algorithm in Cloud Computing.IEEE International Conference on Cloud Computing and Big Data.482 – 486

Li.Xiaofang, Mao. Yingchi, Xiao. Xianjian, and Zhuang. Yanbin.(2014). An Improved Max-Min Task-Scheduling Algorithm for Elastic Cloud.IEEE International Symposium on Computer, Consumer and Control.340-343

Ramezani.Fahimeh, Lu.Jie, Khadeer Hussain.Farookh.(2014). Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization.Springer US. Volume.42. Issue .5. 739-754

Abbas Haidrir.Reza, C.P.Katti and P.C.Saxena.(2014).A load balancing strategy for Cloud Computing environment.IEEE International Conference on Signal Propagation and Computer Technology (ICSPCT). 636 – 641.