

الگوریتم کلونی مورچه برای زمانبندی کار متوازن و پویا در محیط گرید محاسباتی

اسکندر خواجه وند^۱؛ فریبرز محمودی^۲

چکیده

گرید محاسباتی با سرعت در سیستمهای ناهمگون توزیعی برای استفاده و اشتراک منابع برای حل مسائل علمی پیچیده توسعه یافته است. برای رسیدن به کارایی بالا در محیط گرید زمانبندی موثر ضروری است. در این مقاله با استفاده از اطلاعات پویای منابع و مبتنی بر الگوریتم بهینه سازی کلونی مورچه الگوریتمی بنام *DBACA* جهت زمانبندی مجموعه کارهای مستقل ارائه می-کنیم که علاوه بر بهبود زمان کل اجرا، بار محاسباتی را بطور متوازن بین منابع ناهمگون توزیع کرده و کارایی سیستم را افزایش می-دهد. در پایان الگوریتم پیشنهادی در محیط *GridSim* بطور موفقیت آمیزی شبیه سازی شده و نتایج آزمایشها نشان می دهد که الگوریتم *DBACA* در زمانبندی کارها، کارایی بالاتری داشته و زمان کل اجرا را کاهش می دهد.

کلمات کلیدی

گرید محاسباتی، الگوریتم کلونی مورچه، زمانبندی کار

An ant algorithm for balanced and dynamic job scheduling in grids

ABSTRACT

Grid computing has developed very quickly in heterogeneous distributing systems for sharing and using the data resources in order to solve complicated scientific problems. Efficient scheduling is essential to achieve such high-performance in grid environments. This article presents an algorithm called "DBACA", by applying dynamic resource data and based on ant colony optimization algorithm, for scheduling of independent jobs. The proposed algorithm not only enhances the processing time, but also distributes the balanced computational load amongst the heterogeneous resources and, as a result, leads to high performance of the system. In the end, the proposed algorithm is simulated successfully in GridSim environment. The obtained results show that the DBACA algorithm enhances the performance, has a high efficiency and reduces the total processing time remarkably.

KEYWORDS

Computational grids, Ant Colony Algorithm, Task Scheduling

۱- مقدمه

تکنولوژی گرید محاسباتی در اواسط دهه ۱۹۹۰ برای ارائه یک زیر ساخت محاسبات توزیعی در علوم پیشرفته و پروژه های مهندسی، پیشنهاد شده است [۱]. تکنولوژی گرید توسعه تکنولوژی اطلاعات است و هدف اصلی آن ایجاد یک محیط محاسباتی ناهمگون، توزیعی و پویا جهت همکاری منابع اشتراکی برای حل مسائل در سازمانهای مجازی و پویا است [۲، ۳].

^۱ اسکندر خواجه وند، دانشکده مهندسی برق، کامپیوتر و فناوری اطلاعات، دانشگاه آزاد اسلامی، قزوین - ایران، khajevand@qazviniau.ac.ir

^۲ فریبرز محمودی، دانشکده مهندسی برق، کامپیوتر و فناوری اطلاعات، دانشگاه آزاد اسلامی، قزوین - ایران، mahmoudi@qazviniau.ac.ir

اشتراک منابع ویژگی مهم گرید است. همچنین گرید محاسباتی زیربنای سخت‌افزاری و نرم‌افزاری می‌باشد که دسترسی قابل‌اعتماد، پایدار، فراگیر و ارزان را فراهم می‌کند [۲-۴]. بنابراین یک گرید محاسباتی با مجموعه‌ای از منابع ناهمگون^۱ (کامپیوترهای شخصی، ایستگاه‌های کاری^۲، کلاسترها و ابرکامپیوترها) در مقیاس وسیع در ارتباط است که هر یک از این منابع قابلیت محاسبات و پیکربندی متفاوتی دارند و توسط حوزه‌های مدیریتی و سیاست متفاوت سازماندهی می‌شوند.

بنابراین در چنین محیط پیچیده‌ای زمانبندی و مدیریت منابع چالش بزرگی را ایجاد می‌کند و ارائه یک الگوریتم زمانبندی مناسب کارها به منابع بسیار مهم و مشکل بوده و مسئله را تبدیل به یک مسئله NP کامل می‌کند که جهت بدست آوردن کارایی لازم بسیار حائز اهمیت است. یک الگوریتم زمانبندی مناسب باید استراتژی زمانبندی را مطابق با تغییر وضعیت محیط گرید تنظیم کند که بکارگیری الگوریتم کلونی مورچه در چنین محیطی میتواند مناسب باشد [۵]. الگوریتم کلونی مورچه یک الگوریتم اکتشافی با جستجوی محلی موثر برای مسائل ترکیبی [۵، ۶]، بدون هماهنگی و کنترل مرکزی می‌باشد که رفتار جستجوی غذای مورچه را شبیه‌سازی می‌کند. با در نظر گرفتن اصول تشویق و تنبیه فرومون برای مسیرهای انتخاب شده توسط مورچه‌ها، مسیری که فرومون بیشتری می‌توانیم مسیری را که فرومون بیشتری به آن تخصیص یافته و در مراحل بعدی با احتمال بیشتری انتخاب شده و مرحله به مرحله به سمت جواب بهتر همگرا می‌شود.

در این مقاله یک الگوریتم کلونی مورچه بنام $DBACA$ ، برای مسئله زمانبندی کار در محیط گرید محاسباتی ارائه و ارزیابی شده است. الگوریتم پیشنهادی شرایط پویای منابع و توان پردازش اصلی و جاری هر منبع را در تصمیم‌گیری شرکت می‌دهد. همچنین الگوریتم پیشنهادی بار محاسباتی را بطور متوازن بین منابع توزیع می‌کند که موجب کارایی بالاتر سیستم گرید می‌شود. در الگوریتم پیشنهادی ابتدا لیست کارهایی که باید اجرا شود از طریق کاربرد به سیستم گرید ارسال می‌شود. سپس زمانبند کار، اطلاعات منابع ثبت شده را از سیستم گرید دریافت کرده و بر اساس توان پردازشی و پهنای باند هر منبع، برای منابع فرومون را محاسبه می‌کند. پس از این مرحله تعدادی مورچه ایجاد کرده که هر مورچه بطور مستقل مجموعه کار را به منابع انتساب می‌کند. زمانیکه کلیه کارها از طریق هر مورچه به منابع تخصیص داده شد معیار زمان کل را برای زمانبندی هر مورچه محاسبه می‌کنیم و در صورتیکه بهترین زمانبندی این مرحله، زمان کل را نسبت به مرحله قبل کاهش دهد، طبق زمانبندی مورچه مورد نظر فرومون منابع تغییر می‌کند. الگوریتم پیشنهادی در محیط $GridSim$ [۷] شبیه‌سازی شده و نتایج، علاوه بر توازن بار بین منابع، کاهش بسیار خوب زمان کل را نسبت به الگوریتم مبتنی بر کلونی مورچه ACA [۱] نشان می‌دهد.

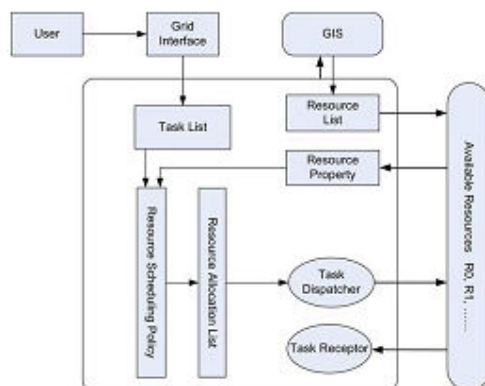
در سالهای اخیر، محققان الگوریتم‌های زمانبندی متفاوتی را برای سیستم‌های موازی معرفی نموده‌اند [۸-۱۱]، که هر کدام از آنها مزایا و معایبی دارد و هیچکدام الگوریتم زمانبندی کاملی نیست و همچنان این مسئله تعداد زیادی از محققان را جذب خود نموده‌است [۵، ۱۱، ۱۲]. در حال حاضر هدف زمانبندی کار در گرید علاوه بر کشف منبعی که کارایی سیستم را بهبود دهد، بهره‌برداری موثر از منابع موجود نیز می‌باشد. الگوریتم $FIFO$ یکی از استراتژیهای مهم در زمانبندی کارها در سیستم‌عامل است. اکثر الگوریتم‌های زمانبندی منابع بر پایه استراتژی $FIFO$ یا $FIFO$ بهبود یافته است، به‌عنوان مثال در [۱۳] یک الگوریتم زمانبندی منابع بر اساس خصوصیت منبع که کار را به منبعی با کمترین بار محاسباتی انتساب می‌کند، ارائه شده است که کارها را با ترتیب $FIFO$ ارسال می‌کند. همچنین یازده الگوریتم اکتشافی از جمله Min_Min و Max_Min و غیره در [۱۴]، الگوریتم ژنتیک [۱۵]، مدل اقتصادی محاسباتی [۷، ۱۶-۱۸] و الگوریتم‌های کلونی مورچه [۱، ۱۹، ۲۰] معرفی شده است.

محیط گرید یک محیط پویا و با مدیریت توزیعی منابع است، بنابر این الگوریتم کلونی مورچه مناسب حل مسائل چنین محیطی می‌باشد. همچنین الگوریتم‌های مطالعه شده، پویایی و توازن بار منابع را بطور همزمان بررسی نکرده‌اند. از این رو در این مقاله قدرت محاسباتی و پویایی بار محاسباتی منابع در طول زمان اجرا در نظر گرفته شده و یک الگوریتم کارا، متوازن و پویا مبتنی بر الگوریتم کلونی مورچه بنام $DBACA$ ارائه و ارزیابی شده است.

ادامه این مقاله بدین صورت سازماندهی شده است: بخش ۲ یک مطالعه مروری روی کارهای انجام شده در این زمینه است. در بخش ۳ فرایند زمانبندی منابع در گرید بررسی شده و در بخش ۴ الگوریتم زمانبندی پیشنهادی $DBACA$ شرح داده شده است. سپس در بخش‌های ۵ و ۶ به ترتیب نتایج شبیه‌سازی، مقایسه ارزیابی آزمایش‌ها و نتیجه‌گیری مقاله ارائه شده است.

۲- فرایند زمانبندی منابع در محیط گرید

وقتی سیستم گرید شروع به کار می‌کند، منابع مشخصات خودشان را در سیستم اطلاعات گرید^۳ ثبت می‌کنند. سپس کاربران لیست کارها را به سیستم گرید ارسال می‌کنند و دلال منابع^۴، کارهای ارسال شده را با توجه به الگوریتم زمانبندی به منابع انتساب می‌کند[۷، ۲۱]. فرایند زمانبندی در شکل (۱) نشان داده شده است.



شکل ۱: دیاگرام روال زمانندی منابع [۱]

لیست کارها، شامل جزئیات کار، مثل توصیف برنامه کاربردی و نیاز کاربر می‌باشد که از طریق واسطه گزید، به دلال منابع ارسال می‌شود [۱۴].

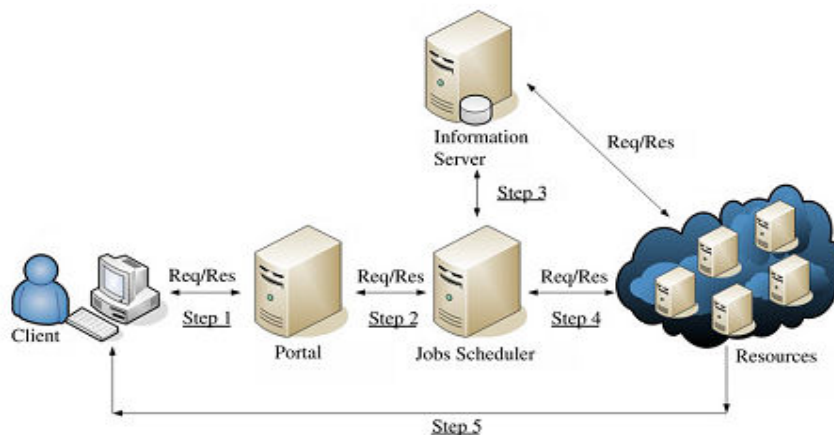
همینکه لیست کارها در دسترس قرار گرفت، دلال منابع با سیستم اطلاعات گزید، جهت کسب اطلاعات پویای منابع (از جمله در دسترس بودن تعداد پردازنده‌های هر منبع، نوع پردازش و غیره) محاوره می‌کند [۲۲، ۲۳]. مطابق با نیاز کاربران و با توجه به سیاست زمانبندی، دلال منابع کارها را زمانبندی می‌کند. سپس توزیع کننده کار با توجه به زمانبندی انجام شده کارها را به منابع مربوطه توزیع می‌کند و در پایان اجرا وقتی کارها کامل می‌شوند نتایج آن بوسیله گیرنده کار جمع آوری می‌شود. البته در صورتی که در حین اجرای کاری خطایی وجود آید در نتیجه اجرای کار بطور کامل به پایان نرسیده و باید دوباره مراحل زمانبندی و توزیع را سیری کند.

۳- الگوریتم پیشنهادی DBACA مبتنی بر الگوریتم کلونی مورچه

الگوریتم *DBACA* مطابق با قدرت محاسباتی اولیه منابع، تعداد و حجم کار پردازش شده توسط هر منبع و بار محاسباتی پویای منابع در طول زمان اجرا، چگالی فرومون را تغییر می‌دهد. هدف این الگوریتم کمینه‌سازی زمان کل اجرای کارها با در نظر گرفتن وضعیت پویا و توازن بار منابع است.

۳-۱ معماری سیستم

معماری سیستم گزید در شکل (۲) نشان داده شده است [۲۴]. چهار قسمت اصلی این معماری عبارتند از: واسطه کاربر^۵، سیستم اطلاعات گزید^۶، زمانبندکار^۷ و منابع گزید. کاربران از طریق واسطه گزید کارهای خود را به سیستم گزید ارسال می‌کنند باتوجه به مشخصات منابع گزیدکه در سیستم اطلاعات گزید ثبت شده است، زمانبندکار، کارهای تحویل شده از طریق کار بر را زمانبندی می‌کند [۷، ۲۱]. البته فرایند زمانبندی در شکل (۱) نشان داده شده است. در بخش ۴-۲ الگوریتم مورد نظر شرح داده شده است.



شکل ۲: معماری سیستم گرد [۲۴]

۳-۲ الگوریتم پیشنهادی DBACA

الگوریتم کلونی مورچه در دنیای واقعی که فرومون روی مسیر توزیع شده و مورچه بدنبال جستجوی غذا است، جهت استفاده در زمانبندی منابع در محیط گرید باید تغییر کند. در یک سیستم گرید واقعی چندین فاکتور روی کارایی سیستم تاثیر می‌گذارد، برای کاهش پیچیدگی مسئله می‌توان بعضی از فاکتورها را در نظر نگرفت. در زمانبندی منابع گرید، پردازش کار روی منابع بخش مهمی را در کارایی سیستم دارد، بنابراین پردازش کار بعنوان خصوصیت اصلی کارها و منابع تعریف می‌شود. در الگوریتم پیشنهادی بجای اینکه فرومون روی مسیر توزیع شود، به منابع اختصاص می‌یابد و در طول زمان اجرا با توجه به قدرت پردازشی هر منبع مقدار فرومون آن کاهش یا افزایش می‌یابد. هدف این روش کمینه‌سازی زمان کل اجرا است. از آنجائیکه در محیط گرید اجرای کارها بطور موازی روی منابع مختلف بطور همزمان انجام می‌شود در نتیجه معیار کارایی معادل زمان کل اجرا است که از زمان شروع اولین کار تا زمان پایان آخرین کار می‌باشد. مراحل الگوریتم پیشنهادی DBACA در زیر شرح داده شده است:

مرحله ۱: همینکه یک لیست کار از طرف کاربر به سیستم گرید ارسال شد، سیستم اطلاعات گرید وضعیت منابع را از جمله تعداد پردازنده‌های هر منبع، قدرت محاسباتی هر منبع پهنای باند انتقال داده‌های هر منبع و غیره را بررسی می‌کند. فرومون اولیه منبع j بوسیله رابطه (۱) بدست می‌آید.

$$\tau_j(\cdot) = m * p + Rate_j \quad (1)$$

که m تعداد پردازنده‌های منبع j ، p توان محاسباتی منبع j و $Rate_j$ پهنای باند انتقال داده‌ها توسط منبع j می‌باشد. مقدار فرومون اولیه هر منبع توانایی محاسبات اولیه هر منبع و توانایی برقراری ارتباط منبع را منعکس می‌کند. در ابتدا فرومون اولیه و فرومون فعلی یکسان است.

مرحله ۲: تعدادی مورچه ایجاد می‌کنیم و یک کپی از لیست کارها، فرومون اولیه و فرومون فعلی را به هر مورچه می‌دهیم تا هر مورچه بطور کامل لیست کارهایش را زمانبندی کند. هر مورچه در این روش اندازه محاسبه و انتقال مورد نیاز کاری که می‌خواهد زمانبندی شود را حمل می‌کند. در ابتدا مورچه‌ها بطور تصادفی روی منابع مختلف قرار داده می‌شوند. زمانیکه یک مورچه یک مرحله انتقال می‌یابد بدین معنی است که یک کار به یک منبع تخصیص می‌یابد.

مرحله ۳: احتمال اینکه منبع j به کار بعدی انتساب یابد بوسیله چگالی فرومون منبع j و رابطه (۲) مشخص می‌شود. به هر منبع به تعداد پردازنده‌های بیکار منبع، توسط هر مورچه بطور مستقل کار انتساب می‌شود. همچنین در صورتی که منبعی، پردازنده بیکار نداشته باشد باید آن منبع را در لیست منابع ممنوعه قرار دهیم و تا زمانیکه حداقل یکی از پردازنده‌های آن منبع بیکار نشده، کار جدیدی از طرف مورچه مربوطه به آن انتساب نشود.

$$p_j^k(t) = \begin{cases} \frac{[\tau_j(t)]^\alpha * [\eta_j]^\beta * [\mu_j(t)]^\gamma}{\sum_{u \in \text{online resource left}} [\tau_u(t)]^\alpha * [\eta_u]^\beta * [\mu_u(t)]^\gamma} & j, u \in \text{online resource left} \\ 0 & \text{others} \end{cases} \quad (2)$$

جائیکه $\tau_j(t)$ فرومون منبع j در زمان t است، η_j توانایی اولیه منبع j است که معادل $\tau_j(\cdot)$ است، $\mu_j(t)$ معکوس بار پویای منبع j در زمان t است. α , β , γ بترتیب درجه اهمیت سه فاکتور توانایی فعلی منبع، توانایی اصلی منبع و بار پویای منبع را مشخص می‌کند که فاکتور سوم یعنی بار پویای منبع توازن بار بین منابع را تنظیم می‌کند. مطابق اینکه اجرای چه تعدادی کار در زمانبندی هر مورچه به پایان رسیده، فرومون منابع مورچه مربوطه تغییر می‌کند و تغییر فرومون باعث تغییر احتمالات انتخاب منابع توسط مورچه می‌شود. وقتی منبعی یک کار را از یک مورچه می‌پذیرد $\tau_j(t)$ بدین صورت تغییر می‌کند $\tau_j(t) = \rho * \tau_j(t-1) + \Delta \tau_j$ جائیکه ρ ضریب تبخیر فرومون است و $\Delta \tau_j$ با روش زیر تعیین می‌شود:

- وقتی کار جدید بطور نرمال به یک منبع انتساب می‌شود $\Delta \tau_j = C_p * (-k)$ می‌شود که k پیچیدگی محاسبات کار و C_p ضریب تنبیه می‌باشد.

- وقتی اجرای کار با موفقیت به پایان می‌رسد، $\Delta \tau_j = C_e * k$ می‌شود که C_e ضریب تشویق می‌باشد.

مرحله ۴: اگر همه کارهای یک مورچه زمانبندی شده است، به مرحله ۵ می‌رویم در غیر این صورت باید مرحله ۳ بازگردد.

مرحله ۵: تابع هدف یعنی زمان کل اجرای هر مورچه را بدست آورده و از بین آنها مورچه ای که توانسته است کمترین زمان اجرا را بدست آورد را شناسایی کرده و در صورتیکه زمان کل اجرای آن بهتر از زمان کل اجرای مرحله قبلی است فرومون تغییر یافته توسط مورچه به عنوان فرومون پایه برای تصمیم‌گیری مرحله بعد تغییر می‌یابد. لازم به ذکر است که تغییرات فرومون در مراحل قبلی به ازای هر مورچه بطور مستقل انجام می‌شود و در صورتیکه بهترین جواب آنها از بهترین جواب قبلی بهتر باشد فرومون پایه برای مرحله بعدی تغییر می‌یابد.

مرحله ۶: در صورتیکه بهترین جواب بین مورچه‌ها در یک مرحله بهتر از مرحله قبلی نباشد مسئله حل شده است در غیر این صورت مراحل ۲ تا ۵ را تکرار می‌کنیم.

از آنجائیکه بار منابع در محیط گرید بطور پویا تغییر می‌کند بنابراین می‌توانیم با توجه به اهمیت هر یک از فاکتورها، پارامترهای α , β , γ را تنظیم کنیم. در این مقاله ما هر سه پارامتر را از لحاظ درجه اهمیت یکسان در نظر گرفته‌ایم.

۴- نتایج شبیه‌سازی

الگوریتم پیشنهادی *DBACA* در محیط شبیه‌سازی *GridSim* [۷] شبیه‌سازی شده و نتایج آن با الگوریتم مبتنی بر *ACA* مقایسه شده است. در این شبیه‌سازی ۵ منبع داریم که مشخصات منابع از جمله تعداد پردازنده‌ها، قدرت محاسباتی و استراتژی استفاده شده در زمانبندی در جدول (۱) معرفی شده است که این مشخصات در آزمایش الگوریتم‌های قبلی [۱۳، ۱] نیز استفاده شده است. همچنین در این آزمایش یک کاربر جهت ارسال کار در نظر گرفته شده و پهنای باند انتقال داده بین کاربر، منابع و زمانبند $1000 MB/S$ است. پارامترهای استفاده شده در انجام آزمایشها در جدول (۲) معرفی شده است.

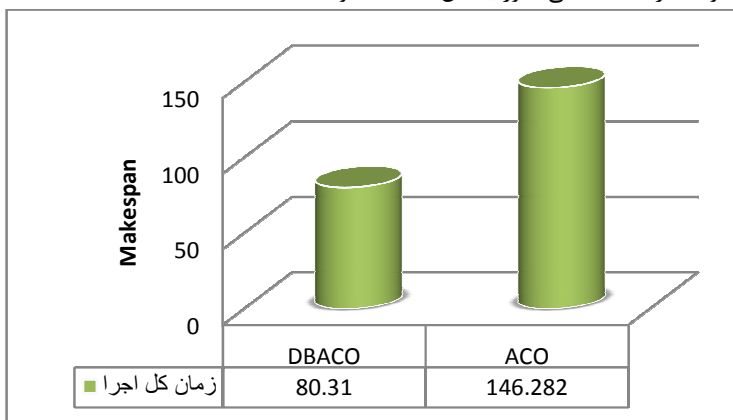
جدول ۱: مشخصات منابع گرید [۱]

منبع	تعداد پردازنده	قدرت محاسبات	نوع مدیریت
R ₀	۸	۱۰۰	Time-share
R _۱	۴	۱۳۳	Time-share
R _۲	۸	۵۰	Space-share
R _۳	۸	۸۰	Time-share
R _۴	۱۶	۱۰۰	Time-share

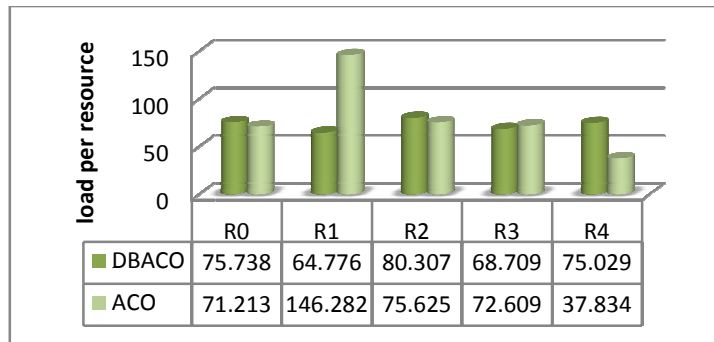
جدول ۲: مقادیر پارامترها در انجام آزمایش

پارامتر	توضیح	مقدار
α	اهمیت توانایی فعلی	۱
β	اهمیت توانایی اصلی	۱
γ	اهمیت بار پویای منبع	۱
$1-p$	ضریب تبخیر	۰.۷۵
Ce	ضریب تشویق	۱.۱
Cp	ضریب تنبیه	۰.۸
N	تعداد مورچه	۱۰۰

آزمایش اول: در آزمایش اول کاربر ۱۰۰ کار را به زمانبند ارسال می‌کند که هر کار با تابع توزیع یکنواخت در محدوده اندازه [۵۰۰، ۱۰۰۰] انتخاب شده است. نتایج بدست آمده میانگین ۲۰ بار آزمایش می‌باشد. شکل (۳) زمان کل اجرا را برای این آزمایش بین الگوریتم مبتنی بر *ACA* و الگوریتم پیشنهادی *DBACA* نشان می‌دهد، همانطور که می‌بینید حدود ۴۵ درصد زمان کل اجرا کاهش یافته است. همچنین شکل (۴) مقدار استفاده هر منبع را در طول کل آزمایش برای هر دو الگوریتم نشان می‌دهد که الگوریتم پیشنهادی تقریباً بطور متوازن بار را بین منابع تقسیم کرده و از همه منابع با توجه به تفاوت قدرت محاسباتی بطور یکسان استفاده نموده است.

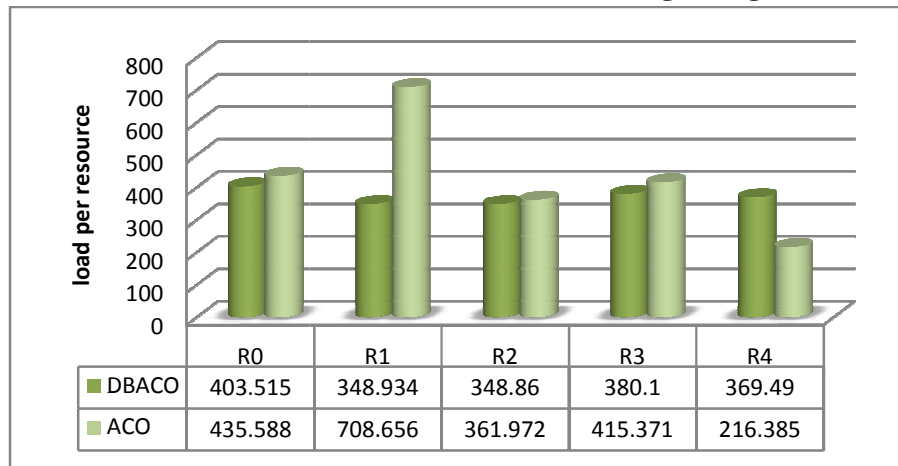


شکل ۳: زمان کل اجرا بین الگوریتم مبتنی بر *ACA* و *DBACA*



شکل ۴: مدت زمان استفاده از هر منبع بین الگوریتم مبتنی بر *DBACO* و *ACA*

آزمایش دوم: در آزمایش دوم کاربر ۱۰۰ کار را به زمانبند ارسال می‌کند که هر کار با تابع توزیع یکنواخت در محدوده اندازه [۱۰۰۰۰۰ و ۱۰۰۰۰۰۰] انتخاب شده است. شکل (۵) نتایج میانگین ۲۰ بار آزمایش بین الگوریتم مبتنی بر *ACA* و الگوریتم پیشنهادی *DBACO* را نشان می‌دهد. هدف از انجام این آزمایش این است که نشان دهیم الگوریتم پیشنهادی با توجه به ناهمگونی خیلی زیاد کارها نیز جواب بسیار مناسبی را ارائه می‌کند. همانطور که در شکل (۵) نشان داده شده الگوریتم پیشنهادی در زمانبندی کارهای ناهمگون، توازن بار محاسباتی بین منابع ناهمگون را به خوبی اعمال می‌کند.



شکل ۵: مدت زمان استفاده از هر منبع بین الگوریتم مبتنی بر *DBACO* و *ACA* در ناهمگونی زیاد کارها

۵- نتیجه‌گیری

از الگوریتم کلونی مورچه، برای حل بسیاری از مسائل از جمله فروشنده دوره‌گرد استفاده شده است. در این مقاله الگوریتم کلونی مورچه جهت زمانبندی کار در محیط گرید بکار رفته و الگوریتمی با استفاده از اطلاعات پویای منابع بنام *DBACO* پیشنهاد شده است. آزمایشهای شبیه-سازی شده نشان می‌دهند که الگوریتم پیشنهادی با کاهش زمان کل اجرا و برقراری توازن بار بین منابع، کارایی سیستم را بهبود می‌بخشد.

- [١] B. Tang, Y. Yin, Q. Liu, and Z. Zhou, "Research on the Application of Ant Colony Algorithm in Grid Resource Scheduling," in *Wireless Communications, Networking and Mobile Computing*, ٢٠٠٨. WiCOM '٠٨. ٤th International Conference Dalian, ٢٠٠٨, pp. ١-٤.
- [٢] I. Foster and C. Kesselman, *The grid: blueprint for a new computing infrastructure*: Morgan Kaufmann, ٢٠٠٤.
- [٣] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations " *International Journal of High Performance Computing Applications*, vol. ١٥, pp. ٢٠٠-٢٢٢, ٢٠٠١.
- [٤] V. Berstis, "Fundamentals of grid computing," IBM Redbooks, ٢٠٠٥.
- [٥] M. Dorigo and L. M. Gambardella, "Ant colonies for the traveling salesman problem," *BioSystems*, vol. ٤٣, pp. ٧٣-٨١, ١٩٩٧.
- [٦] W. yue and T. Hongjiu, "Application in TSP Based on Ant Colony Optimization," *Journal of Wuhan University of Technology (Information and Managing Engineering Edition)*, vol. ٢٨, pp. ٢٤-٢٦, ٢٠٠٦.
- [٧] R. Buyya and M. Murshed, "GridSim: a toolkit for the modelling and simulation of distributed resource management and scheduling for Grid computing," *Concurrency and Computation: Practice and Experience*, vol. ١٤, pp. ١١٧٥-١٢٢٠, ٢٠٠٢.
- [٨] D. Kondo, D. P. Anderson, and J. M. VII, "Performance Evaluation of Scheduling Policies for Volunteer Computing," in *IEEE International Conference on e-Science and Grid Computing*, ٢٠٠٧, pp. ٤١٥-٤٢٢.
- [٩] R.-S. Chang, J.-S. Chang, and S.-Y. Lin, "Job scheduling and data replication on data grids " *Future Generation Computer Systems*, vol. ٢٣, pp. ٨٤٦-٨٦٠, August ٢٠٠٧.
- [١٠] Y. Gao, H. Rong, and J. Z. Huang, "Adaptive grid job scheduling with genetic algorithms," *Future Generation Computer Systems*, vol. ٢١, pp. ١٥١-١٦١, January ٢٠٠٥.
- [١١] F. Dong and S. G. Akl, "Scheduling Algorithms for Grid Computing: State of the Art and Open Problems," Kingston, Ontario, Canada: School of Computing, Queen's University ٢٠٠٦.
- [١٢] S. E and E. K, "An ACO algorithm for graph coloring problem," in *Computational Intelligence Methods and Applications*, ٢٠٠٥ ICSC Congress ٢٠٠٥.
- [١٣] R. Mah, *Chemical process structures and information flows*: Butterworth-Heinemann, ١٩٩٠.
- [١٤] Z. HuiFu, Z. Zude, and L. Fangmin, "Research on Interface Model of Manufacturing Resource Sharing Grid," *China Mechanical Engineering*, vol. ١٦, pp. ٤٢٤-٤٢٧, ٢٠٠٥.
- [١٥] V. D. Martino and M. Mililotti, "Scheduling in a Grid Computing Environment Using Genetic Algorithms," ٢٠٠٢, pp. ٢٣٥-٢٣٩.
- [١٦] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economic Models for Resource Management and Scheduling in Grid Computing," *Concurrency and Computation: Practice and Experience*, vol. ١٤, pp. ١٥٠٧-١٥٤٢, ٢٠٠٢.
- [١٧] R. Buyya, D. Abramson, and S. Venugopal, "The Grid Economy," in *Proceedings of the IEEE*, ٢٠٠٥, pp. ٦٩٨-٧١٤.
- [١٨] O. O. Sonmez and A. Gursoy, "A Novel Economic-Based Scheduling Heuristic for Computational Grids " *International Journal of High Performance Computing Applications*, vol. ٢١, pp. ٢١-٢٩, ٢٠٠٧.
- [١٩] Z. Xu, X. Hou, and J. Sun, "Ant algorithm-based task scheduling in grid computing," in *Electrical and Computer Engineering*, Canadian, ٢٠٠٣, pp. ١١٠٧-١١١٠.
- [٢٠] X. Zhihong and G. Junhua, "Research on Ant Algorithm Based Classified Task Scheduling in Grid Computing," in *Journal of Hebei University of Technology*, ٢٠٠٦, pp. ٧١-٧٨.
- [٢١] L. Hao, "Implement of Computational Grid Application Scheduling Simulation with GridSim Toolkit," *Journal of Jilin Normal University (Nature Science Edition)*, vol. ٣, pp. ٦٣-٦٤, ٢٠٠٣.
- [٢٢] N. Muthuvelu, J. Liu, and N. L. Soe, "A dynamic job grouping-based scheduling for deploying applications with fine-grained tasks on global grids," in *Proceedings of the ٣rd Australasian Workshop on Grid Computing and e-Research (AusGrid ٢٠٠٥)*, Newcastle, New South Wales, Australia, ٢٠٠٥, pp. ٤١-٤٨.
- [٢٣] C. Wei, Y. Shoubao, and S. Kai, "Grouped task scheduling design for coarse-grained grid application," *Journal of Huazhong University of Science & Technology (Nature Science Edition)*, vol. ٣٤, pp. ١٤٨-١٥١, ٢٠٠٦.
- [٢٤] R.-S. Chang, J.-S. Chang, and P.-S. Lin, "An ant algorithm for balanced job scheduling in grids," *Future Generation Computer Systems*, vol. ٢٥, pp. ٢٠-٢٧, ٢٩ June ٢٠٠٩.

^١ Heterogeneous

^٢ Workstations

^٣ Grid Information System(GIS)

^٤ Resource Broker(RB)

^٥ portal

^٦ Information Server

^٧ Jobs Scheduler