

بهینه‌سازی زمان در زمانبندی گریدهای محاسباتی اقتصادی با استفاده از

اتوماتای سلولی

اسکندر خواجه‌وند^۱؛ محمدرضا میبیدی^۲

چکیده

زمانبندی کار یکی از مسائل مهم در تحقیقات گریدهای محاسباتی می‌باشد. گریدهای محاسباتی زیرساختی برای اشتراک منابع توزیعی و محاسبات علمی است. در چنین محیطی جهت استفاده موثر از منابع، الگوریتم زمانبندی کارا به منظور انتساب کارها به منابع ضروری است. در گریدهای محاسباتی اقتصادی، کاربران مهلت زمانی و بودجه مورد نظر خود را تعیین کرده و بهینه‌سازی زمان یا هزینه را درخواست می‌کنند. الگوریتم زمانبندی با هدف بهینه‌سازی زمان، باید با توجه به قیمت و توانمندی منابع گریدها، تخصیص منابع به کارهای ناهمگون کاربران را با حداقل زمان در محدوده بودجه معین انجام دهد. در این مقاله الگوریتم جدید مبتنی بر اتوماتای سلولی بنام CATO، به منظور بهینه‌سازی زمان پیشنهاد شده است. سپس الگوریتم پیشنهادی در محیط GridSim، شبیه‌سازی شده و نتایج آزمایشها نشان می‌دهد که الگوریتم CATO، در زمانبندی کارها، کارایی بالاتری داشته و درخواست کاربران را در زمان کمتری انجام می‌دهد.

کلمات کلیدی

اتوماتای سلولی، انتساب کار، بهینه‌سازی هزینه، تجارت منابع، زمانبندی اقتصادی، گریدهای محاسباتی.

Time optimization for task scheduling in economical grid computing by using cellular automata

ABSTRACT

Scheduling is one of the important issues in grid computing researches. Grid computing is an infrastructure to share distributing data resources and scientific computations. To use the data resources efficiently in such environments, scheduling algorithms are of utmost importance in order to assign tasks to resources. In economical grid computing, users define their deadline and budgets, and request the optimization time or cost. To achieve the most optimal time, the timing algorithm must assign the optimal resources to the users' heterogeneous tasks in the shortest time and based on the allocated budget. The present article proposes a new algorithm based on cellular automata called "CATO" in order to optimize the time. Afterwards, the proposed algorithm is simulated in GridSim environment. The obtained results show that the CATO algorithm is highly efficient in scheduling, and meets the users' requirements in a shorter time.

KEYWORDS

Cellular Automata, Task Assignment, Cost Optimization, Resource Trading, Economical Scheduling, Grid Computing.

۱- مقدمه

امروزه مسائل علمی، خیلی پیچیده بوده و نیاز به قدرت محاسباتی بالایی دارند. تکنیکهای قدیمی همچون محاسبات توزیعی و موازی جوابگوی اینگونه مسائل نمی‌باشد و مستلزم صرف زمان زیاد است. گریدهای محاسباتی [۱]، برای حل اینگونه مسائل مناسب است. از آنجائیکه در گریدها، مجموعه‌ای از منابع ناهمگون و خود مختار و توزیع شده برای انجام محاسبات همکاری می‌کنند، باید در هر لحظه وضعیت شبکه، قابلیت و کارایی منابع مورد بررسی قرار گیرد. بنابراین در چنین محیطی یک الگوریتم زمانبندی موثر برای کارها حائز

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

^۲. محمدرضا میبیدی، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران - ایران، mmeybodi@aut.ac.ir

اهمیت است. تعدادی الگوریتم زمانبندی کار در [۲-۴] پیشنهاد شده است. همچنین یک مطالعه حرفه ای روی الگوریتم های زمانبندی کار و دسته بندی آنها و معرفی مسائل باز، در [۵]، ارائه شده است.

گرید محاسباتی^۱، زیر بنای سخت افزاری و نرم افزاری می باشد که دسترسی قابل اعتماد، پایدار، فراگیر و ارزان را فراهم می کند [۸-۱۱]. یک گرید محاسباتی با مجموعه ای از منابع ناهمگون^۲ (کامپیوترهای شخصی، ایستگاه های کاری^۳، کلاسترها و ابرکامپیوترها) در مقیاس وسیع در ارتباط است. گریدهای محاسباتی به تدریج به سوی تجاری شدن پیش می روند و دارندگان منابع با انگیزه های مالی، منابع خود را در اختیار دیگران قرار می دهند. مشتریان گرید نیز با پرداخت هزینه درخواست خود، می توانند از این منابع مشترک استفاده کنند. از آنجائیکه صاحبان و استفاده کنندگان منابع دارای اهداف، استراتژی ها و الگوهای عرضه و تقاضای متفاوتی هستند. در چنین شرایطی نمی توان از راهکارهای متداول برای منابعی که سعی می کنند میزان کارایی کل سیستم را بهبود دهند، استفاده کرد. برای این منظور، در سال های اخیر از رویکردهای اقتصادی برای مدیریت تخصیص منابع در گرید استفاده شده است [۱۲-۱۵]. یکی از مدل های اقتصادی که بدین منظور استفاده می شود مدل بازار کالا^۴ می باشد. در این مدل هر منبع دارای قیمت مشخصی است که براساس عرضه و تقاضا، ارزش آن در سیستم اقتصادی تعیین می شود.

منابع گرید به صورت ناهمگون و توزیع شده هستند و به طور مشترک مورد استفاده قرار می گیرند. از طرفی کاربران، محدودیت های مهلت^۵ (زمان اتمام اجرای برنامه) و بودجه (هزینه محاسبات) را برای گرید تعیین می کنند. این مسائل، باعث پیچیدگی عمل زمانبندی و تبدیل مسئله به $NP_Complete$ می گردد. یکی از استراتژی های الگوریتم زمانبندی، مینیم سازی زمان محاسبات در محدوده بودجه (بهینه سازی زمان) می باشد، که تا کنون چندین الگوریتم مکاشفه ای^۶ بدین منظور گزارش شده است. الگوریتم BTO در [۱۶-۱۷] برای مینیم سازی زمان در محدوده بودجه معین، جهت زمانبندی کارهای همگون و مستقل روی منابع ناهمگون پیشنهاد شده و در [۱۸] مورد ارزیابی قرار گرفته است. این الگوریتم برای کارهای ناهمگون جواب قابل قبولی را ارائه نمی کند. همچنین برای برطرف کردن ضعف های این الگوریتم، الگوریتم های $ABTO$ ، $EBTO$ و $AEBTO$ در [۶]، جهت زمانبندی کارهای مستقل و ناهمگون با هدف مینیم سازی زمان ارائه و ارزیابی شده است. همچنین الگوریتم های مبتنی بر اتوماتای یادگیر به نام های $LATO$ و $ALATO$ در [۷] ارائه و ارزیابی شده است. این الگوریتم ها هر کار را با یک اتوماتای یادگیر تجهیز می کند، که هر اتوماتا دارای ساختار متغیر می باشد، تعداد عملیات هر اتوماتا برابر تعداد منابع موجود در سیستم است که باید اتوماتا تصمیم بگیرد هر کار را به کدام منبع تخصیص بدهد. البته تصمیم بر اساس احتمال انتخاب منبع از طرف اتوماتا است و مجموع احتمال انتخاب باید برابر عدد یک باشد. پس از انتخاب یک منبع و ارزیابی نتیجه احتمالات انتخاب منابع تغییر می کند. در این روش با افزایش منابع، تعداد عمل اتوماتا افزایش یافته و تصمیم گیری سخت می شود در نتیجه این دو الگوریتم مقیاس پذیر نمی باشد.

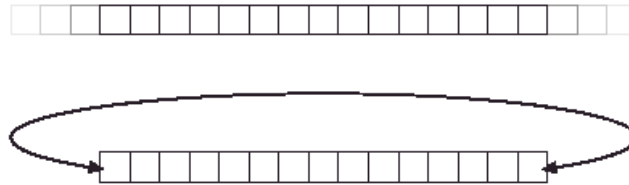
در این مقاله با بکارگیری اتوماتای سلولی، الگوریتمی جدید به نام $CATO$ ^۷، برای زمانبندی کارهای مستقل با هدف بهینه سازی زمان در محدوده بودجه معین در محیط گریدهای محاسباتی اقتصادی بر اساس مدل بازار کالا پیشنهاد شده است. در این روش ابتدا لیست منابع موجود را از سرویس اطلاعاتی گرید دریافت می کنیم و منابع مجاور در لیست را بعنوان همسایه یکدیگر در نظر می گیریم. بنابراین هر منبع یکی از سلولهای اتوماتا است و با منابع چپ و راست لیست همسایه می باشد. در ابتدا مجموعه کارها را بین منابع یا سلولها بصورت تصادفی توزیع می کنیم، در هر مرحله تکرار الگوریتم یک کار را از صف هر سلول انتخاب کرده، در صورتیکه کار مورد نظر روی سلول های همسایه با زمان کمتر و در محدوده بودجه انجام گیرد، کار مذکور را به صف سلول همسایه انتقال می دهیم. قابل ذکر است که این تصمیم گیری توسط هر سلول بطور مستقل و موازی انجام می شود. الگوریتم پیشنهادی در محیط شبیه سازی $GridSim$ [۱۸]، مورد ارزیابی قرار گرفته و در مقایسه با الگوریتم های مطرح شده زمان اجرا را به مقدار قابل توجهی کاهش می دهد. همچنین در شرایط خاص مانند ناهمگونی بسیار زیاد کارها، دارای عملکرد مناسب است، به همین دلیل استفاده از الگوریتم پیشنهادی $CATO$ به منظور زمانبندی کارها در گرید برای توجیه اقتصادی بوده و با جذب مشتریان بیشتر، موجب استفاده بهتر از منابع و سودآوری بالاتر می شود.

ادامه مقاله بدین صورت سازماندهی شده است: در بخش ۲ اتوماتای سلولی را معرفی می کنیم و سپس در بخش ۳ زمانبندی اقتصادی گرید شرح داده می شود و در بخش ۴ الگوریتم زمانبندی پیشنهادی ارائه می گردد. بخش ۵ و ۶ به ترتیب نتایج شبیه سازی و ارزیابی الگوریتم پیشنهادی و مقایسه آن با الگوریتم های دیگر را در بر می گیرد.

۲- اتوماتای سلولی

اتوماتای سلول [۱۹-۲۰]، مدل ریاضی برای بررسی ساختار سیستم های پویا و پیچیده است که در آنها چندین مؤلفه ساده برای الگوهای پیچیده با هم همکاری می کنند. اتوماتای سلولی از یک شبکه منظم سلولها تشکیل شده است که هر سلول می تواند ($K > 1$) مقدار مختلف به خود بگیرد. سلولهای اتوماتای سلولی در زمانهای گسسته بطور همزمان و برطبق یک قانون محلی بنام Φ بهنگام می شوند، که در آن مقدار هر سلول براساس مقادیر سلولهای همسایه تعیین می گردد. اتوماتای سلولی براساس معیارهای مورد بررسی به دسته های مختلف تقسیم می گردد.

بعنوان نمونه براساس معیار بعد شبکه، اتوماتای سلولی به اتوماتای سلولی یک بعدی، دوبعدی و غیره تقسیم می گردد و براساس مقدار k به اتوماتای سلولی دودویی ($k=2$) و اتوماتای سلولی چند مقداره ($k>2$) تقسیم می شود. همچنین اتوماتای سلولی را براساس شبکه همسایه ها می-توان به دودسته اتوماتای سلولی با مرز پرریودیک و اتوماتای سلولی با مرز غیر پرریودیک تقسیم نمود. بعنوان مثال در اتوماتای سلولی یک بعدی با n سلول که در شکل (۱) نشان داده شده است، در صورتی که سلولهای شماره ۱ و n را با هم همسایه درنظر بگیریم اتوماتای سلولی با مرز پرریودیک و در صورتی که برای سلولهای ۱ و n دو سلول مجازی با مقدار صفر بعنوان همسایه در نظر بگیریم آن را اتوماتای سلولی با مرز غیر پرریودیک می نامند [۲۰-۲۱].



شکل ۱: اتوماتای سلولی یک بعدی با مرز غیر پرریودیک و اتوماتای سلولی یک بعدی با مرز پرریودیک

در اتوماتای سلولی یک بعدی مقدار سلول i (برای $1 \leq i \leq n$) در زمان t که با $a_i(t)$ نشان داده میشود با فرمول (۱) محاسبه میگردد.

$$a_i(t+1) = \Phi[a_{i-1}(t), a_i(t), a_{i+1}(t)] \quad (1)$$

در رابطه بالا، اگر قانون Φ فقط به مقدار همسایه ها بستگی داشته باشد آنرا قانون *general* مینامند و اگر قانون Φ تابعی از مجموع مقادیر سلولهای همسایه و سلول مرکزی باشد آنرا قانون *totalistic* میگویند و با فرمول (۲) بیان میشود:

$$a_i(t+1) = \Phi[a_{i-1}(t) + a_i(t) + a_{i+1}(t)] \quad (2)$$

در صورتی که قانون تابعی از مجموع مقادیر سلولهای همسایه و مقدار سلول مرکزی باشد آنرا قانون *Outer totalistic* می گویند و با فرمول (۳) نشان داده می شود:

$$a_i(t+1) = \Phi[a_i(t), a_{i-1}(t) + a_{i+1}(t)] \quad (3)$$

همچنین قانون Φ میتواند بصورت قطعی یا احتمالی باشد. در صورتیکه Φ یک تابع تصادفی باشد آنرا قانون احتمالی و در غیر اینصورت آنرا قانون قطعی میگویند.

از زمانی که مفاهیم اولیه اتوماتای سلولی [۲۲] پیشنهاد شد، تا کتاب اخیر از ولفرام بنام "یک علم جدید" [۲۳]، ساختار ساده اتوماتای سلولی برای محققان زیادی جذاب است و تعداد مقاله های زیادی از محققان هر ساله منتشر می شود. دلیل عمومیت یافتن اتوماتای سلولی سادگی و توانایی مدل سازی سیستم های پیچیده می باشد. اتوماتای سلولی همچنین در مدل سازی و طراحی الگوریتم بسیاری از مسائل بکار می رود. از جمله کاربردهای آن عبارت است از: بازی ها، مدلسازی پدیده های فیزیکی و شیمیایی، رمزنگاری، علوم طبیعی و اجتماعی، پیاده سازی مدارات مجتمع، بازار سهام، پردازش موازی، تشخیص و دسته بندی الگو، پردازش تصویر و غیره. برای مطالعه بیشتر به [۲۴] مراجعه کنید. در این مقاله جهت حل مسئله بهینه سازی زمان زمانبندی کارها در محیط گرید از یک اتوماتای سلولی یک بعدی با مرز پرریودیک و سلول های چند مقداری و قوانین *Outer totalistic* استفاده شده است.

۳- زمانبندی اقتصادی گرید

رشد و عمومیت یافتن اینترنت و در دسترس بودن کامپیوترها و شبکه های پر سرعت بعنوان اجزا آماده و کم هزینه، طریقه انجام محاسبات و استفاده کامپیوترها را تغییر داده است. گرید محاسباتی، منابع نا همگون توزیع شده را به منظور حل مسائل محاسباتی پیچیده بکار می برد. مدیریت منابع در محیط گرید، به علت توزیع جغرافیایی، ناهمگونی، خود مختار بودن و پویا بودن بار کاری منابع، مسئله پیچیده ای است. این مقدمه باعث مطرح شدن گرید اقتصادی [۱۳] می شود. یکی از مدل های اقتصادی که بدین منظور استفاده شده است مدل بازار کالا می باشد. در این مدل، هر منبع دارای قیمت مشخصی است که براساس عرضه، تقاضا و ارزش درسیستم اقتصادی تعیین شده است.

طول یک کار، تعداد دستورالعمل های آن می باشد و برحسب واحد MT^h (میلیون دستورالعمل) اندازه گیری می شود. اگر کارهای یک برنامه کاربردی، کاملاً همگون^۱ باشند (طول یکسان داشته باشند)، می توان الگوریتم زمانبندی با پیچیدگی چند جمله ای برای مینیمم سازی هزینه در محدوده مهلت اجرا طراحی کرد، ولی هنگامی که کارها دارای نا همگونی باشند (طول های متفاوت داشته باشند)، مساله زمانبندی با هدف بهینه سازی هزینه به یک مساله *NP-Complete* تبدیل می شود.

قیمت اعلام شده برای یک منبع، مقدار هزینه استفاده از آن در واحد زمان می باشد و قیمت مفید یک منبع به مقدار هزینه استفاده از آن برای اجرای هر میلیون دستورالعمل (MI) گفته می شود. در این مقاله، واژه های قیمت، ارزانی و گرانی با توجه به قیمت مفید منبع به کار می روند.

کاربر گرید پس از اینکه محدودیت های مهلت زمانی و بودجه خود را مشخص کرد، ممکن است بهینه سازی هزینه و یا زمان را در خواست کند. در صورتی که کاربر خواستار بهینه سازی زمان باشد، الگوریتم زمانبندی باید با صرف بودجه ای که کاربر تعیین کرده است، کمترین زمان را برای اجرای کارها بدست آورد و در صورتی که کاربر خواستار بهینه سازی هزینه باشد، الگوریتم زمانبندی باید در محدوده مهلتی که کاربر تعیین کرده است، کمترین هزینه را برای اجرای کارها بدست آورد.

الگوریتم زمانبندی با هدف بهینه سازی هزینه و یا زمان از دو نوع می باشند: زمانبندی مرحله ای^{۱۰} و زمانبندی یکباره^{۱۱}. الگوریتم هایی که از روش زمانبندی مرحله ای استفاده می کنند، در طول اجرای خود به تدریج کارها را به منابع موجود در گرید واگذار می کنند. در مقابل، الگوریتم هایی که رویکرد زمانبندی یکباره را اتخاذ می کنند، یک نداشت از کارهای برنامه کاربر به منابع موجود را تولید می کنند که برای بررسی به بخش کنترل پذیرش داده می شود. پس از بررسی، اگر نداشت بدست آمده، نیازمندی های کاربر را برآورده کند کارها طبق این نداشت به منابع برای اجرا واگذار می گردد. برای توضیحات بیشتر می توان به [۶] مراجعه کرد.

الگوریتم های بهینه سازی هزینه و یا زمان، دارای چند مرحله مشترک ابتدایی می باشند. این مراحل عبارتند از یافتن، تجارت و مرتب سازی منابع. الگوریتم هایی که از روش زمانبندی یکباره استفاده می کنند، در پایان نیز دارای دو مرحله مشترک کنترل پذیرش و توزیع هستند. در ادامه توضیحاتی درباره این مراحل داده می شود.

- **یافتن منابع**^{۱۲}: شناسایی منابعی که می توانند در اجرای کارها مورد استفاده قرار گیرند و همچنین بدست آوردن توانمندی ها و ویژگی های آنها که از طریق سرویس اطلاعاتی گرید انجام می گیرد.
- **تجارت منابع**^{۱۳}: شناسایی هزینه هر یک از منابع در واحد زمان ($G\$/Sec$)، شناسایی میزان توانمندی منبع در واحد زمان (نرخ اجرای میلیون دستورالعمل در ثانیه، MI/Sec) و محاسبه قیمت واقعی و مفید منبع که همان هزینه اجرای یک میلیون دستورالعمل در ثانیه می باشد ($G\$/Sec$).

- **مرتب سازی منابع**: یک الگوریتم بهینه سازی، حتما به یک ترتیب خاص از منابع بر حسب قیمت آنها نیاز دارد. در واقع، اولویت واگذاری کار با منابع ارزان تر می باشد. اگر دو منبع دارای قیمت مفید یکسان باشند، منبعی که توانمندی بیشتری دارد در اولویت قرار می گیرد.
- **کنترل پذیرش**^{۱۴}: هزینه کل اجرای کارها بر روی منابع بر طبق زمانبندی مشخص و با بودجه تعیین شده توسط کاربر مقایسه می گردد. همچنین زمان تقریبی اجرای برنامه بر طبق زمانبندی محاسبه و سپس با مهلت تعیین شده توسط کاربر مقایسه می شود. در صورتی که هزینه ها در محدوده بودجه باشد و اجرای برنامه قبل از مهلت زمانی پایان یابد، قابل قبول است.
- **توزیع**^{۱۵}: در این مرحله در صورت پذیرش کاربر، کارها طبق نداشت تعیین شده، به منابع واگذار می شود.

الگوریتم پیشنهادی در این مقاله از زمانبندی یکباره استفاده می کند بنابراین امکان تضمین محدودیت های بودجه و مهلت برای چندین کاربر به طور هم زمان وجود خواهد داشت.

۴- الگوریتم پیشنهادی

الگوریتم پیشنهادی $CACO$ عمل زمانبندی کارها را با هدف مینیمم سازی هزینه محاسبات در محدوده مهلت تعیین شده با بکارگیری اتوماتای سلولی انجام می دهد. برای حل مسئله از اتوماتای سلولی یک بعدی با مرز پریودیک استفاده می کنیم. بطوری که هر سلول نماینده یک منبع گرید است. وقتی از سرویس دهنده اطلاعات گرید ویژگی های منابع را درخواست می کنیم لیستی از مشخصات منابع در دسترس در اختیار ما قرار می گیرد. منابعی که در این لیست در کنار یکدیگر قرار دارند را بعنوان همسایه یکدیگر فرض می کنیم. بطوری که هر منبع در مکان i با منبع در مکان $i+1$ و $i-1$ همسایه بوده و همچنین منبع ابتدای لیست با منبع آخر لیست نیز همسایه است. شکل (۱) این همسایگی را نشان می دهد. می توان تصور کرد که منابع با توپولوژی $Ring$ به یکدیگر متصل شده اند و هر منبع با منبع بعدی و قبلی در حلقه با یکدیگر همسایه است. برای هر منبع یک لیست در نظر می گیریم، که در ابتدا کارها بصورت تصادفی با توزیع یکنواخت در درون لیست منابع ذخیره می گردد. الگوریتم طی دو فاز انجام می گیرد که در زیر هر دو فاز را با جزئیات بیشتر و الگوریتم مربوطه توضیح می دهیم.

۴-۱- فاز اول

ابتدا کارها را بطور تصادفی با توزیع یکنواخت در لیست منابع توزیع می کنیم و زمان پاسخ هر منبع را با توجه به لیست کارهای انتساب شده محاسبه می کنیم. حالا الگوریتم را بصورت مرحله ای اجرا می کنیم تا اینکه ماکزیمم زمان پاسخ منابع به حداقل ممکن تقلیل یابد. بدین صورت که

در هر مرحله هر منبع کوچکترین کار لیست خود را پیدا کرده و در صورتیکه آن کار توسط منابع همسایه در زمان کمتر اجرا شود، کار را به لیست منبع همسایه انتقال می‌دهد. سپس زمان پاسخ هر منبع را با توجه به لیست کارهای جدید انتقال یافته تا زمانیکه واریانس زمان پاسخ منابع در مرحله جدید با واریانس زمان پاسخ منابع در مرحله قدیم متفاوت است، محاسبه می‌کنیم. در غیر این صورت فاز اول خاتمه یافته و به فاز دوم می‌رویم. قطعاً باید در هر مرحله زمان پاسخ هر منبع کمتر و یا مساوی مرحله قبل گردد. بدلیل اینکه هر منبع کوچکترین کار خود را انتخاب کرده و در صورتیکه یکی از همسایگان آن منبع کار را سریعتر انجام دهد آن کار را به همسایه منتقل می‌کند. الگوریتم فاز اول CATO در شکل (۲) نمایش داده شده است.

۱. کارها بصورت تصادفی با توزیع یکنواخت در لیست منابع توزیع می‌شوند.
۲. زمان پاسخ هر منبع محاسبه می‌گردد.
۳. واریانس زمان پاسخ کل منابع را محاسبه کن.
۴. تا زمانی که واریانس زمان پاسخ کل منابع مرحله جدید، مخالف واریانس زمان پاسخ کل منابع مرحله قبل است:
 - کوچکترین کار را برای هر منبع بطور موازی پیدا کن.
 - در صورتیکه همسایه چپ یا راست هر منبع کار مورد نظر را در زمان کمتر انجام می‌دهد. کار را به لیست کارهای آن همسایه منتقل کن.
 - واریانس زمان پاسخ کل منابع را محاسبه کن.

شکل ۲: الگوریتم فاز اول (CATO)

۴-۲- فاز دوم

در پایان فاز اول در لیست هر منبع تعدادی کار وجود دارد که ماگزیمم زمان پاسخ در کل منابع به حداقل ممکن کاهش یافته است. حالا باید ماگزیمم زمان پاسخ بدست آمده را بعنوان مهلت زمانی در نظر بگیریم و سعی کنیم که هزینه کل اجرا را تا حد بودجه تعیین شده کاهش دهیم. الگوریتم بصورت مرحله‌ای اجرا می‌شود، در هر مرحله هر منبع از داخل لیست کارهای مربوطه کوچکترین کار را انتخاب کرده و در صورتیکه یکی از همسایگان، کار مورد نظر را در مهلت زمانی تعیین شده با هزینه کمتر می‌تواند انجام دهد آن کار را به لیست کارهای آن همسایه منتقل می‌کنیم. در پایان این مرحله هزینه کل را محاسبه می‌کنیم و در صورتیکه هزینه کل از بودجه تعیین شده کمتر باشد الگوریتم به پایان رسیده و ماگزیمم زمان پاسخ که بعنوان مهلت در نظر گرفته شده بود، زمان پاسخ بهینه را نشان می‌دهد اما در صورتیکه هزینه کل مرحله جاری بیشتر از بودجه تعیین شده باشد باید یک مرحله دیگر الگوریتم را اجرا کنیم. همچنین در صورتیکه هزینه کل مرحله جاری بیشتر از بودجه تعیین شده باشد و با هزینه کل مرحله قبل برابر باشد باید به اندازه یک مقدار کوچکی مهلت زمانی را افزایش دهیم و یکبار دیگر الگوریتم را با شرایط جدید اجرا کنیم. البته این مقدار کوچک به اندازه زمان اجرای کوچکترین کار می‌باشد. الگوریتم فاز دوم CATO در شکل (۳) نمایش داده شده است.

۱. ماگزیمم زمان پاسخ بدست آمده از فاز ۱ را بعنوان مهلت زمانی در نظر بگیر.
۲. هزینه اجرای کل منابع را محاسبه کن.
۳. تا زمانی که هزینه اجرای کل منابع مرحله جدید بیشتر از بودجه تعیین شده است:
 - کوچکترین کار در لیست هر منبع را پیدا کن.
 - از بین منابع جاری و دو همسایه چپ و راست منبعی که کار مورد نظر را با در نظر گرفتن مهلت زمانی با هزینه کمتر اجرا می‌کند را پیدا کن و کار را به آن منبع انتقال بده.
 - هزینه اجرای کل منابع را در مرحله جدید محاسبه کن.
 - اگر هزینه اجرای کل منابع مرحله جدید و قدیم مساوی است و این هزینه از بودجه تعیین شده بیشتر است:
 - مهلت زمانی را به اندازه زمان مورد نیاز کوچکترین کار افزایش بده.

شکل ۳: الگوریتم فاز دوم (CATO)

پس از انجام این دو فاز حالا در لیست هر منبع تعدادی کار موجود است که باید به منبع مربوطه انتساب شود.

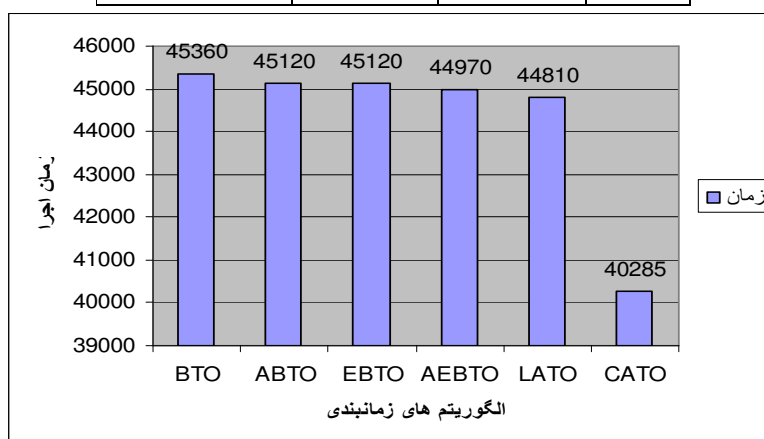
۵- نتایج شبیه سازی

الگوریتم پیشنهادی در محیط شبیه سازی *GridSim* [۱۸]، شبیه سازی شده و نتیجه آن با نتایج بدست آمده از الگوریتم های مکاشفهای و الگوریتم های مبتنی بر اتوماتای یادگیر مقایسه شده است. نتایج گزارش شده، میانگین ۲۰ بار شبیه سازی می باشد. محیط شبیه سازی برای گرید، شامل هشت منبع و یک کاربر است، کلیه منابع محاسباتی دارای یک پردازنده هستند که مشخصات آنها در جدول (۱) داده شده است. همانطور که مشاهده می شود، منابع کاملاً ناهمگون در نظر گرفته شده است. همچنین برنامه کاربر از ۲۰۰ کار مستقل تشکیل شده است، که طول هر کار به صورت تصادفی از محدوده [۱۰۰۰۰، ۲۰۰۰۰] انتخاب می شود. گستردگی این محدوده، منجر به ناهمگونی کارها می گردد. کاربر مقدار ۱۲۰۰۰۰ را برای بودجه تعیین کرده و بهینه سازی زمان را درخواست می کند.

در شکل (۴) الگوریتم های مکاشفهای *BTO*، *ABTO*، *EBTO*، *AEBTO* [۱۶-۱۷] و الگوریتم مبتنی بر اتوماتای یادگیر *LATO* [۷]، همچنین الگوریتم پیشنهادی *CATO* با یکدیگر مقایسه شده اند. همانطور که مشاهده می شود، الگوریتم پیشنهادی کمترین زمان را برای اجرای کارها بدست آورده و در مقایسه با بهترین الگوریتم مبتنی بر اتوماتای یادگیر گزارش شده *LATO*، در حدود ۴۵۰۰ واحد (۱ درصد) کاهش زمان را نشان می دهد. کاهش زمان اجرای کارها در گرید منجر به جذب مشتریان بیشتر و استفاده بهتر از منابع و سود آوری بالاتر می شود.

جدول ۱: پیکر بندی منابع [۷]

نام منبع	نرخ اجرا (MI/sec)	قیمت (G\$/sec)	قیمت مفید (G\$/1000MI)
R _۱	۱۰۰	۰.۵	۵
R _۲	۱۸۰	۱	۵.۵۵
R _۳	۲۴۰	۱.۵	۶.۲۵
R _۴	۲۸۰	۲	۷.۱۴
R _۵	۳۰۰	۲.۵	۸.۳۳
R _۶	۴۰۰	۴	۱۰
R _۷	۵۰۰	۶	۱۲
R _۸	۶۰۰	۹	۱۵



شکل ۴: مقایسه الگوریتم های بهینه سازی زمان

۶- نتیجه گیری

در این مقاله، الگوریتم مبتنی بر اتوماتای سلولی برای زمانبندی در گریدهای محاسباتی اقتصادی *CATO*، بر اساس مدل بازار کالا به منظور مینیمم سازی زمان در محدوده بودجه تعیین شده، پیشنهاد گردید. الگوریتم پیشنهادی بصورت غیر متمرکز و با شروع از یک وضعیت کاملاً تصادفی قوانین محلی به یک هدف سراسری می رسد. از آنجائیکه هر منبع تنها با دو همسایه خود ارتباط دارد، هر چه تعداد منابع و تعداد کارهای مستقل افزایش یابد باز هم هر منبع با دو همسایه خود ارتباط داشته، در نتیجه الگوریتم مقیاس پذیر است. الگوریتم با استفاده از جعبه ابزار *GridSim* شبیه سازی شده و کارایی آن در محیط گرید با منابع و کارهای ناهمگون مورد بررسی قرار گرفت. نتایج شبیه سازی نشان می دهد که الگوریتم پیشنهاد *CATO* در مقایسه با الگوریتم مبتنی بر اتوماتای یادگیر گزارش شده، از کارایی بالاتری برخوردار بوده و درخواست های کاربر را در زمان کمتری اجرا می کند. کاهش زمان اجرای کارها باعث افزایش توان عملیاتی در گرید می شود.

- [1] D. A. Reed, "*Grids: The Teragrid and Beyond*," IEEE Computer, Vol. 36, No. 1, PP. 62-68, 2003.
- [2] R. S. Chang, J. S. Chang, and S. Y. Lin, "*Job Scheduling and Data Replication on Data Grids*," Future Generation Computer Systems, Vol. 23, No. 7, PP. 846-860, 2007.
- [3] Y. Gao, H. Rong, and J. Z. Huang, "*Adaptive Grid Job Scheduling with Genetic Algorithms*," Future Generation Computer Systems, Vol. 21, No. 1, PP. 151-161, 2005.
- [4] E. Byun, S. Choi, M. Baik, J. Gil, C. Park, and C. Hwang, "*Markov Job Scheduler Based on Availability in Desktop Grid Computing Environment*," Future Generation Computer Systems, Vol. 23, No. 4, PP. 16-22, 2007.
- [5] F. dong, and S. K. Akl, "*Scheduling Algorithms for Grid Computing: State of the Art and Open Problems*," School of Computing, Queen's university, Kingston, Ontario, Canada, Tech. Rep. TR-2006-504, 2006.
- [6] Y. MahdaviFar, and M. Meybodi, "*Scheduling Algorithms for Time Optimization in Economic Computational Grid*," CEE, 2007.
- [7] Y. MahdaviFar, and M. Meybodi, "*Time Optimization in Economic Computational Grid using Learning Automata*," IDMC' 07, Amir kabir university, 2007.
- [8] I. Foster, and C. Kesselman, *The Grid: Blueprint for New Computing Infrastructure*, Elsevier Inc, 2004.
- [9] I. Foster, C. Kesselman, and S. Tuecke, "*The Anatomy of the Grid: Enabling Scalable Virtual Organization*," International Journal of Supercomputer Applications, 2001.
- [10] M. Baker, R. Buyya, and D. Laforenza, "*Grids and Grid Technologies for Wide-Area Distributed Computing*," The Journal of Concurrency and Computation, Practice and Experience, Vol. 14, 2002.
- [11] V. Berstis, *Fundamentals of Grid Computing*, IBM Redbooks, 2005.
- [12] R. Buyya, D. Abramson, and J. Giddy, "*A Case for Economy Grid Architecture for Service-Oriented Grid Computing*," Proceedings of the 10th IEEE International Heterogeneous Computing Workshop, 2001.
- [13] R. Buyya, D. Abramson, and J. Giddy, "*An Economy Driven Resource Management Architecture for Global Computational Power Grids*," Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, 2000.
- [14] R. Buyya, D. Abramson, and J. Giddy, "*Nimrod-G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid*," The 9th International Conference on High Performance Computing in Asia-Pacific Region, 2000.
- [15] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "*Economic Models for Resource Management and Scheduling in Grid Computing*," The Journal of Concurrency and Computation: Practice and Experience, 2002.
- [16] R. Buyya, J. Giddy, and D. Abramson, "*An Evaluation of Economy-Based Resource Trading and Scheduling on Computational Power Grids for Parameter Sweep Applications*," Proceedings of the 3rd International Workshop on Active Middleware Services, 2000.
- [17] R. Buyya, "*Economic-Based Distributed Resource Management and Scheduling for Grid Computing*," Ph.D Thesis, School of Computer Science and Software Engineering, Monash University, Melbourne, Australia, 2002.
- [18] R. Buyya, and M. Murshed, "*GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing*," Journal of Computation: Practice and Experience, 2002.
- [19] M. Sipper, "*Evolution of Parallel Cellular Machines, The Cellular Programming Approach*," LNCS 1194, 1997.
- [20] S. Wolfram, *Cellular Automata and Complexity: Collected Papers*, Addison-Wesley, 1994.
- [21] P. P. Chaudhuri, D. R. Chowdhury, S. Nandi, and S. Chattopadhyay, "*Additive Cellular Automata Theory and Applications*," IEEE Computer Society Press, Vol. 1, 1997.
- [22] J. V. Neumann, and A. W. Burks, "*The Theory of Self-Reproducing Automata*," University of Illinois Press, London, 1966.
- [23] S. Wolfram, *A New Kind of Science*, Wolfram Media Inc, 2002.
- [24] N. Ganguly, B. K. Sikdar, A. Deutsch, G. Canright, and P. P. Chaudhuri, "*A Survey on Cellular Automata*," Future and Emerging Technologies Unit of the European Commission, 2003.

¹ Computational Grids

² Heterogeneous

³ Workstations

⁴ Commodity Market Model

⁵ Deadline

⁶ Heuristic

⁷ Cellular Automata Cost Optimization (CACO)

⁸ Million Instruction

⁹ Homogeneous

¹⁰ Time shared

¹¹ Space shared

¹² Resource Discovery

¹³ Resource Trading

¹⁴ Admission Control

¹⁵ Dispatching