

# بهینه سازی هزینه زمانبندی در گرید های محاسباتی اقتصادی با استفاده از اتوماتای سلولی

محمد رضا میبدی

دانشکده مهندسی کامپیوتر و فناوری اطلاعات  
دانشگاه صنعتی امیرکبیر  
تهران - ایران  
[mmeybodi@aut.ac.ir](mailto:mmeybodi@aut.ac.ir)

اسکندر خواجه وند

دانشکده مهندسی برق، کامپیوتر و فناوری اطلاعات  
دانشگاه آزاد اسلامی  
قزوین - ایران  
[khajevand@qazviniau.ac.ir](mailto:khajevand@qazviniau.ac.ir)

چکیده: گرید محاسباتی ، منابع ناهمگون توزیع شده در سطح جغرافیایی را به منظور انجام محاسبات پیچیده ارائه می کند. در گریدهای محاسباتی اقتصادی جهت استفاده از منابع اشتراکی برای اجرای کار باید هزینه پرداخت شود . زمانبندی کار در گرید محاسباتی اقتصادی یکی از مسائل مهم است . برای استفاده موثر از منابع گرید، یک الگوریتم زمانبندی کارا برای انتساب کارها به منابع نیاز است. در گرید محاسباتی اقتصادی هر کاربر ، مهلت زمانی و بودجه مورد نظر خود را تعیین کرده و بهینه سازی هزینه یا زمان را در خواست می کند. یک الگوریتم زمانبندی با هدف بهینه سازی هزینه ، باید با توجه به قیمت و توانمندی منابع گرید ، عمل تخصیص آنها به کارهای نا همگون کاربر را طوری انجام دهد که اجرای کارها با حداقل هزینه و قبل از مهلت تعیین شده ، پایان باید. در این مقاله ، الگوریتم مبتنی بر اتماتای سلولی ، برای این منظور پیشنهاد شده است. نتایج شبیه سازی نشان می دهد که الگوریتم پیشنهادی در مقایسه با الگوریتم های مکاشفه ای و الگوریتم های مبتنی بر اتماتای یادگیری ، از کارایی بالاتری برخوردار بوده و در خواست های کاربر را بهینه کمتر انجام می دهد.

**کلمات کلیدی :** گرید محاسباتی ، تجارت منابع ، انتساب کار ، زمانبندی اقتصادی ، بهینه سازی هزینه ، اتماتای سلولی

## ۱ - مقدمه

امروزه مسائل علمی، خیلی پیچیده بوده و نیاز به قدرت محاسباتی و فضای ذخیره سازی بالایی دارند. تکنیکهای قدیمی همچون محاسبات توزیعی و موازی برای اینگونه مسائل مناسب نیستند. پردازش و ذخیره سازی حجم زیادی از داده ها ، زمان زیادی را صرف می کند. گرید محاسباتی [1] برای حل اینگونه مسائل مناسب است. در گرید شرایطی همچون وضعیت شبکه و وضعیت منابع باید مورد بررسی قرار گیرد. اگر شبکه و منابع نا پایدار باشند اجرای کارها در حین اجرا با خطای مواجه شده و زمان اجرای کارها افزایش می یابد. بنابر این یک الگوریتم زمانبندی موثر برای کارها در گرید محاسباتی سیار حائز اهمیت است به همین منظور تعدادی الگوریتم زمانبندی کار برای گرید در [2-4] پیشنهاد شده است. همچنین در [5] یک مطالعه حرفه ای روی الگوریتم های زمانبندی کار و دسته بندی آنها و معروفی مسائل باز انجام شده است.

گرید محاسباتی<sup>1</sup> ، یک زیر بنای سخت افزاری و نرم افزاری می باشد که دسترسی قابل اعتماد ، پایدار، فرآگیر و ارزان را فراهم می کند [8-11]. یک گرید محاسباتی با مجموعه ای از منابع ناهمگون<sup>2</sup> (کامپیوتر های شخصی ، ایستگاه های کاری<sup>3</sup> ، کلاستر ها و ابر کامپیوتر ها) در مقیاس وسیع در ارتباط است. گرید های 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 در خواست خود ، می توانند از این منابع استفاده کنند.

<sup>1</sup> Computational Grids

<sup>2</sup> Heterogeneous

<sup>3</sup> Workstations

صاحبان و استفاده کنند گان منابع دارای اهداف ، استراتژی ها و الگوهای عرضه و تقاضای متفاوتی هستند . در چنین شرایطی نمی توان از راهکارهای متداول برای منابع 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 گردید استفاده شده است [15-12]. یکی از مدل های اقتصادی که بدین منظور استفاده می شود مدل بازار کالا<sup>4</sup> 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 گردید به صورت ناهمگون و توزیع شده هستند و به طور مشترک مورد استفاده قرار می گیرند. از طرفی کاربران ، محدودیت های مهلت<sup>5</sup> (زمان اتمام اجرای برنامه) و بودجه (هزینه محاسبات) را برای گردید تعیین می کنند. این مسائل ، باعث پیچیدگی عمل زمانبندی و تبدیل مسئله به NP\_Complete می گردد. با توجه به نیاز کاربر، یکی از استراتژی های الگوریتم زمانبندی ، مینیمم کردن هزینه در محدوده مهلت (بهینه سازی هزینه) می باشد.

تا کنون چندین الگوریتم مکافهه ای<sup>6</sup> بدین منظور گزارش شده است. الگوریتم<sup>7</sup> BCO در [16,17] برای مینیمم کردن هزینه در محدوده مهلت معین ، جهت زمانبندی کارهای همگون و مستقل روی منابع ناهمگون پیشنهاد شده است. بنابراین برای کارهای ناهمگون جواب قابل قبولی را ارائه نمی کند ، این الگوریتم در [18] مورد ارزیابی قرار گرفته است. همچنین برای برطرف کردن ضعف های این الگوریتم ، الگوریتم های مکافهه ای<sup>8</sup> ABCO و AEBCO<sup>10</sup> در [6] جهت زمانبندی کارهای مستقل و ناهمگون با هدف بهینه سازی هزینه ارائه و ارزیابی شده است. همچنین دو الگوریتم مبتنی بر اتوماتای یادگیر به نام های LACO<sup>11</sup> و ALACO<sup>12</sup> در [7] ارائه و ارزیابی شده اند. این الگوریتمها هر کدام از کارها را به یک اتوماتای یادگیر مجهز می کنند ، هر اتوماتا دارای ساختار متغیر بوده و به تعداد منابع گرید، عمل دارد که هر عمل معادل انتخاب یک منبع است. بنابراین با افزایش منابع، عمل اتوماتاها افزایش یافته در نتیجه این دو الگوریتم مقیاس پذیر نمی باشد.

در این مقاله با بکارگیری اتوماتای سلولی ، الگوریتم جدید به نام CACO<sup>13</sup> برای زمانبندی کارهای مستقل از هم با هدف بهینه سازی هزینه در گردید های محاسباتی اقتصادی با مدل بازار کالا پیشنهاد می شود. الگوریتم پیشنهادی بصورت غیرمت مرکز و مقیاس پذیر بوده و با شروع از یک وضعیت کاملاً تصادفی و قوانین محلی به یک هدف سراسری ، بهینه سازی هزینه می رسد. الگوریتم LACO و ALACO طی ۱۰۰۰۰ مرحله فرایند یادگیری به جواب می رسد در حالی که در الگوریتم پیشنهادی در کمتر از ۱۰۰ مرحله فرایند جابجایی کار با منابع همسایه ، به جواب قابل توجهی می رسیم. الگوریتم پیشنهادی با استفاده از جعبه ابزار GridSim [18] شبیه سازی شده و کارایی آن مورد بررسی قرار گرفته که نسبت به الگوریتم های قبلی به مقدار قابل توجهی هزینه را کاهش می دهد . همچنین در شرایط خاص مانند ناهمگونی بالای کارها ، دارای عملکرد مناسب نسبت به الگوریتم های قبلی است . به همین دلیل استفاده از الگوریتم پیشنهادی مبتنی بر اتوماتای سلولی به منظور زمانبندی کارها در گردید دارای توجیه اقتصادی بوده و با جلب مشتریان بیشتر، موجب استفاده بهتر از منابع و سود آوری بالاتر می شود.

<sup>4</sup> Commodity Market Model

<sup>5</sup> Deadline

<sup>6</sup> Heuristic

<sup>7</sup> Buyya cost optimization(BCO)

<sup>8</sup> Advanced Buyya Cost Optimization(ABCO)

<sup>9</sup> Extended Buyya Cost Optimization(EBCO)

<sup>10</sup> Advanced Extended Buyya Cost Optimization(AEBCO)

<sup>11</sup> Learning Automata Cost Optimization(LACO)

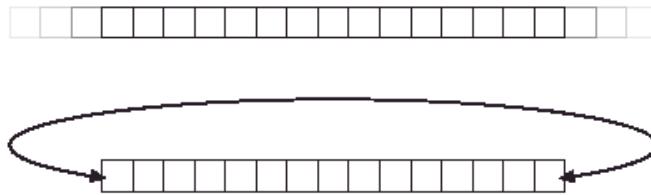
<sup>12</sup> Advanced Learning Automata Cost Optimization(ALACO)

<sup>13</sup> Cellular Automata Cost Optimization (CACO)

ادامه مقاله بدین صورت سازماندهی شده است : در بخش ۲ اتماتای 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- اتماتای سلولی

اتماتای سلولی [19,20]، 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. اتماتای سلولی از یک شبکه منظم سلولها تشکیل شده است که هر سلول می تواند  $k > 1$  مقدار مختلف به خود بگیرد. سلولهای اتماتای سلولی در زمانهای گسسته بطور همزمان و برطبق یک قانون محلی بنام  $\Phi$  بهنگام می شوند که در آن مقدار هر سلول براساس مقادیر سلولهای همسایه تعیین می گردد. اتماتای سلولی براساس معیارهای مورد بررسی به دسته های مختلف تقسیم می گردد. عنوان نمونه براساس معیار بعد شبکه، اتماتای سلولی به اتماتای سلولی یک بعدی، دو بعدی و غیره تقسیم می گردد و براساس مقدار  $k$  به اتماتای سلولی دودوئی ( $k=2$ ) و اتماتای سلولی چند مقداره ( $k > 2$ ) تقسیم می شود. همچنین اتماتای سلولی را براساس شبکه همسایه ها می توان به دو دسته اتماتای سلولی با مرز پریودیک و اتماتای سلولی با مرز غیر پریودیک تقسیم نمود. عنوان مثال در اتماتای سلولی یک بعدی با  $n$  سلول که در شکل ۱ نشان داده شده است در صورتی که سلولهای شماره ۱ و  $n$  را به هم همسایه درنظر بگیریم اتماتای سلولی با مرز پریودیک و در صورتی که برای سلولهای ۱ و  $n$  دو سلول مجازی با مقدار صفر عنوان همسایه در نظر بگیریم آن را اتماتای سلولی با مرز غیر پریودیک می نامند. [20,21]



شکل ۱: اتماتای سلولی یک بعدی با مرز غیر پریودیک و اتماتای سلولی یک بعدی با مرز پریودیک

در اتماتای سلولی یک بعدی مقدار سلول آن برای  $i \leq i \leq n$  در زمان  $t$  که با  $a_i(t)$  نشان داده می شود بصورت زیر محاسبه می گردد.

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

در رابطه بالا، اگر قانون  $\Phi$  فقط به مقدار همسایه ها بستگی داشته باشد آنرا قانون general مینامند و اگر قانون  $\Phi$  تابعی از مجموع مقادیر سلولهای همسایه و سلول مرکزی باشد آنرا قانون totalistic می گویند و بصورت زیر بیان می شود

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

در صورتی که قانون تابعی از مجموع مقادیر سلولهای همسایه و مقدار سلول مرکزی باشد آنرا قانون Outer totalistic می گویند و بصورت زیر نشان داده می شود.

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

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

از روزهای اول که مفاهیم اولیه اتوماتای سلولی [22] پیشنهاد شد تا کتاب اخیر از ولفرام بنام " یک علم جدید " [23] ساختار ساده اتوماتای سلولی برای محققان زیادی جذاب بوده است. تعداد زیادی از مقاله های محققان هر ساله منتشر می شود. دلیل عمومیت یافتن اتوماتای سلولی سادگی آن و توانایی مدل سازی سیستم های پیچیده می باشد. اتوماتای سلولی همچنین در مدل سازی و طراحی الگوریتم ها برای مسائل زیادی بکار می رود. از جمله کاربردهای آن عبارت است از : بازی ها ، مدلسازی پدیده های فیزیکی و شیمیایی ، رمزگاری ، علوم طبیعی و اجتماعی ، پیاده سازی مدارات مجتمع ، بازار سهام ، پردازش موازی ، تشخیص و دسته بندی الگو ، پردازش تصویر و غیره که می توانید برای مطالعه بیشتر به [24] مراجعه کنید .

در این مقاله جهت حل مسئله بهینه سازی هزینه زمانبندی کارها در محیط گرید از یک اتوماتای سلولی یک بعدی با مرز پریودیک 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 ، نا همگونی ، خود مختار بودن و پویا بودن بار کاری منابع ، مسئله پیچیده ای است . این مقدمه باعث مطرح شدن گرید اقتصادی [13] 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 بر حسب واحد  $MI^{14}$  (میلیون دستورالعمل) اندازه گیری می شود. اگر کارهای یک برنامه کاربردی ، کاملا همگون<sup>15</sup> باشند (طول یکسان داشته باشند) ، می توان الگوریتم زمانبندی با پیچیدگی چند جمله ای برای مینیمم کردن 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<sup>16</sup> و زمانبندی یکباره<sup>17</sup> . 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 کند . در

<sup>14</sup> Million Instruction

<sup>15</sup> Homogeneous

<sup>16</sup> Time shared

<sup>17</sup> Space shared

مقابل ، الگوریتم هایی که رویکرد زمانبندی یکباره را اتخاذ می کنند، یک نگاشت 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 نگاشت بدست آمده ، نیازمندی های کاربر را برآورده کند کارها طبق این نگاشت به منابع برای اجرا و اگذار می گردد. برای توضیحات بیشتر می توان به [6] 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<sup>۱۸</sup>**: 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 گردید انجام می گیرد.
- **تجارت منابع<sup>۱۹</sup>**: شناسایی هزینه هر یک از منابع در واحد زمان (G\$/sec) ، شناسایی میزان توانمندی منبع در واحد زمان (نحو 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 ، حتماً 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<sup>۲۰</sup>**: هزینه کل اجرای کارها بر روی منابع بر طبق زمانبندی مشخص وبا 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 تعیین شده توسط کاربر مقایسه می شود. در صورتی که هزینه ها در محدوده بودجه باشند اگر دو منبع دارای قیمت مفید یکسان نباشند، منبعی که همان هزینه اجرای یک میلیون دستورالعمل در ثانیه می باشد (G\$/sec).
- **توزیع<sup>۲۱</sup>** : در این مرحله در صورت پذیرش کاربر، کارها طبق نگاشت تعیین شده ، به منابع واگذار می شود. الگوریتم پیشنهادی در این مقاله از زمانبندی یکباره استفاده می کنند 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 کردن هزینه محاسبات در محدوده مهلت تعیین شده با بکارگیری اتوماتای سلولی انجام می دهد . برای حل مسئله بر روی اتوماتای سلولی یک بعدی با مرز پریویویک متصرکز می شویم . بطوری که هر سلول نماینده یک منبع گردید است . وقتی از سرویس دهنده اطلاعات گردید 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 ۱ با منبع در مکان  $i+1$  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 تپولوژی Ring به یکدیگر متصل شده اند و هر منبع با منبع بعدی و قبلی در حلقه با یکدیگر همسایه است. برای هر منبعی یک لیست پیوندی در نظر می گیریم که در ابتدا کارها بصورت تصادفی با توزیع یکنواخت بین منابع توزیع شده و مشخصات آنها در داخل این لیست های پیوندی ذخیره می گردد. الگوریتم طی دو فاز انجام می گیرد . بطور کلی در فاز اول می خواهیم زمان پاسخ هر منبع را در محدوده مهلت زمانی معین شده قرار دهیم . و در فاز دوم با در نظر گرفتن محدوده مهلت زمانی، هزینه اجرا را کاهش می دهیم . در زیر هر دو فاز را با جزئیات بیشتر و الگوریتم مربوطه توضیح می دهیم.

<sup>18</sup> Resource Discovery

<sup>19</sup> Resource Trading

<sup>20</sup> Admission Control

<sup>21</sup> Dispatching

**فاز اول :** ابتدا کارها را بطور تصادفی با توزیع یکنواخت بین منابع توزیع کرده و هر کار را در داخل لیست مربوط به منبع قرار می دهیم 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 از مهلت زمانی معین شده کمتر با مساوی شود. در هر مرحله هر منبع کوچکترین کار لیست خود را پیدا کرده و در صورتیکه آن کار توسط منبع همسایه در زمان کمتر اجرا شود، آن را به لیست منبع همسایه انتقال می دهد . سپس زمان پاسخ هر منبع را با توجه به لیست کارهای جدید انتقال یافته محاسبه می کنیم. در صورتیکه زمان پاسخ ماکزیمم در بین منابع از محدوده مهلت زمانی معین شده کمتر باشد فاز اول پایان یافته در غیر این صورت مرحله بعدی همین فاز را ادامه می دهیم. 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 همسایه آن منبع کار را سریعتر انجام دهد آن کار را به همسایه منتقل می کند. الگوریتم فاز اول CACO در زیر نمایش داده شده است .

۱. کارها بصورت تصادفی با توزیع یکنواخت بین منابع توزیع می شوند.
۲. زمان پاسخ هر منبع محاسبه می گردد.
۳. تا زمانی که ماکزیمم زمان پاسخ مرحله جدید کمتر یا مساوی مهلت زمانی معین شده نیست :
  - 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 می دهد. کار را به لیست کارهای آن همسایه منتقل کن.
  - ماکزیمم زمان پاسخ بین منابع را پیدا کن.

**فاز دوم :** حالا هر منبعی تعدادی کار در داخل لیست خودش دارد که زمان پاسخ هر منبع از مهلت زمانی معین شده کمتر و یا مساوی است . در این فاز باید هزینه اجرا را با در نظر گرفتن مهلت زمانی کاهش دهیم. الگوریتم این فاز نیز بصورت مرحله ای اجرا می شود ، به طوری که در هر مرحله هر منبع کوچکترین کار لیست خود را پیدا کرده و اگر هریک از منابع همسایه کار مورد نظر را با هزینه کمتر در محدوده مهلت زمانی انجام دهد ، آن کار را به لیست آن همسایه منتقل می کند . الگوریتم فاز دوم CACO در زیر نمایش داده شده است.

۱. هزینه اجرای کل منابع را محاسبه کن.
۲. تا زمانی که هزینه اجرای کل منابع مرحله قدیم بیشتر از هزینه اجرای کل منابع مرحله جدید است :
  - کوچکترین کار را برای هر منبع بطور موازی پیدا کن .
  - در صورتیکه با در نظر گرفتن محدوده مهلت زمانی ، همسایه چپ یا راست هر منبع کار مورد نظر را با هزینه کمتر انجام دهد ، کار را به لیست کارهای آن همسایه منتقل کن.
  - هزینه اجرای کل منابع را در مرحله جدید محاسبه کن.

پس از انجام این دو فاز حالا در لیست هر منبع تعدادی کار موجود است که باید هر کار را به منبع مورد نظر انتساب کنیم.

## ۵- نتایج شبیه سازی

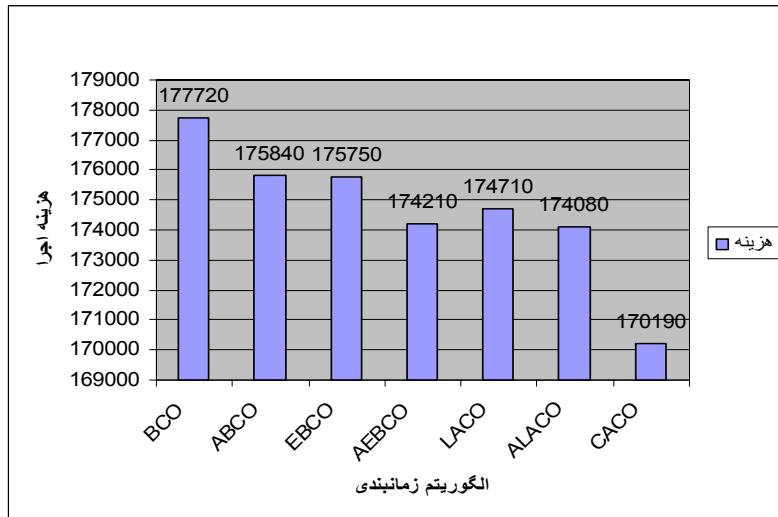
الگوریتم پیشنهادی با استفاده از جعبه ابزار GridSim [18] شبیه سازی شده و نتیجه آن با نتایج بدست آمده از الگوریتم های مکاشفه ای و الگوریتم های مبتنی بر اتوماتی یادگیر مقایسه شده است. نتایج گزارش شده ، میانگین ۲۰ بار شبیه سازی می باشد. محیط شبیه سازی شده برای گردید، شامل تعدادی منبع و یک کاربر است. کلیه منابع محاسباتی دارای یک پردازنده هستند که مشخصات آنها در جدول (۱) داده شده است. همانطور که مشاهده می شود ، منابع کاملا نا همگون در نظر گرفته شده است. برنامه کاربر از ۲۰۰ کار مستقل از هم تشکیل

شده است که طول هر کار به صورت تصادفی از محدوده (۰...۲۰۰۰۰۰)۱۰۰۰۰۰) انتخاب می شود. گستردگی این محدوده، منجر به ناهمگونی کارها می گردد. کاربر مقدار ۱۲۰۰۰ را برای مهلت زمانی تعیین کرده و بهینه سازی هزینه را در خواست می کند.

در شکل (۲) الگوریتم های مکاشفه ای AEBCO , EBCO , ABCO , BCO [6,17] و الگوریتم های مبتنی بر یادگیری ALACO , LACO [7] ، همچنین الگوریتم پیشنهادی CACO با یکدیگر مقایسه شده اند. همانطور که مشاهده می شود، الگوریتم پیشنهادی کمترین هزینه را برای اجرای کارها بدست آورده و در مقایسه با بهترین الگوریتم مبتنی بر اتوماتای یادگیر گزارش شده ALACO در حدود ۴۰۰۰ واحد (۲,۲ درصد) کاهش هزینه اجرای کارها در گرید منجر به جلب مشتریان بیشتر و استفاده بهتر از منابع و سود آوری بالاتر می شود.

نام منبع	نرخ اجرا (Ml/sec)	قیمت (G\$/sec)	قیمت مفید (G\$/1000MI)
R1	100	0.5	5
R2	180	1	5.55
R3	240	1.5	6.25
R4	280	2	7.14
R5	300	2.5	8.33
R6	400	4	10
R7	500	6	12
R8	600	9	15

جدول(۱): پیکربندی منابع



شکل(۲) مقایسه الگوریتم های بهینه سازی هزینه

#### ۶- نتیجه گیری

در این مقاله ، الگوریتم مبتنی بر اتوماتای سلولی برای زمانبندی در گریدهای محاسباتی اقتصادی CACO ، با مدل بازار کالا به منظور مینیمم کردن هزینه در محدوده مهلت تعیین شده، پیشنهاد گردید. الگوریتم پیشنهادی بصورت غیر مت مرکز و با شروع از یک وضعیت کاملاً تصادفی و قوانین محلی به یک هدف سراسری می رسد. چون 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. این 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 پیشنهادی CACO در مقایسه با الگوریتم های مبتنی بر اتوماتای یادگیر گزارش شده ، از کارایی بالاتری برخوردار بوده و در خواست های کاربر را با هزینه کمتری اجرا می کند.

## مراجع

- [1] D.A.Reed , " Grids : The teragrid and beyond " , IEEE Computer , 36 , 2003 .
- [2] R.S.Chang , J.S.Chang , S.Y.Lin , " Job scheduling and data replication on data grids " , Future Generation Computer Systems , 23 , 2007 .
- [3] Y.Gao , H.Rong , J.Z.Huang , " Adaptive grid job scheduling with genetic algorithms " , Future Generation Computer Systems , 21 , 2005 .
- [4] E.Byun , S.Chi , M.Baik , J.Gil , C.Park , C.Hwang , " Markov job scheduler based on availability in desktop grid computing environment " , Future Generation Computer Systems , 23 , 2007 .
- [5] F.dong , S.K.Akl , " Scheduling algorithms for grid computing : state of the art and open problems " , Technical Report 2006-504 , School of computing , Queen's university , Kingston , Ontario , Canada , 2006 .
- [6] Y.Mahdavifar , M.Meybodi , " New scheduling algorithms for cost optimization .in economic computational Grid " , CSICC , 2006 .
- [7] Y.Mahdavifar , M.Meybodi , " Cost optimization in economic computational grid using learning automata" , IDMC' 07 , Amir Kabir University , 2007 .
- [8] I.Foster , C.Kesselman , " The Grid : Blueprint for new computing infrastructure " , Elsevier Inc , 2004 .
- [9] I.Foster , C.Kesselman , S.Tuecke , " The anatomy of the Grid : Enabling scalable virtual organization " , International Journal of Supercomputer Applications , 2001 .
- [10] M.Baker , R.Buyya , D.Laforenza , " Grids and grid technologies for wide-area distributed computing " , The Journal of Concurrency and Computation , Practice and Experience , 14 , 2002 .
- [11] V.Berstis , " Fundamentals of grid computing " , IBM Redbooks , 2005 .
- [12] R.Buyya , D.Abramson , J.Giddy , " A case for economy grid architecture for Service-oriented grid computing " , proceedings of the 10<sup>th</sup> IEEE International Heterogeneous Computing Workshop , 2001 .
- [13] R.Buyya , D.Abramson , 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 , J.Giddy , " Nimrod-G : An architecture for a resource management and scheduling system in a global computational grid " , The 4<sup>th</sup> International Conference on High Performance Computing in Asia-Pacific Region , 2000 .
- [15] R.Buyya , D.Abramson , J.Giddy , H.Stockinger , " Economic models for resource management and scheduling in grid computing " , The journal of Concurrency and Computation , 2002 .
- [16] R.Buyya , J.Giddy , D.Abramson , " An evaluation of economy-based resource trading and scheduling on computational power grids for parameter sweep applications " , Proceedings of the 2<sup>nd</sup> 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 , 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.Pal.Chaudhuri , D.Roy.Chowdhury , S.Nandi , S.Chatopadhyay , “ Additive cellular automata theory and applications ” , 1 , IEEE Computer Society Press , 1997 .
- [22] J.V.Neumann , A. W. Burks , " The theory of self-reproducing automata " , Univ. 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 , P.P.Chaudhuri , " A survey on cellular automata " , Future and Emerging Technologies unit of the European Commission , 2003