IDMC'07
20-21 Nov.2007
Amir Kabir University

بهينه‌سازي زمان در گريدهاي محاسباتي اقتصادي با استفاده از اتوماتاهاي يادگير

ياسر مهدوي‌فر* محمدرضا ميبدي†

چكيده

كاربران در گريدهاي محاسباتي اقتصادي بايستي براي اجراي برنامه‌هاي كاربردي خود هزينه پرداخت كنند. هر كاربر، مهلت زماني و بودجه مورد نظر خود را تعيين كرده و مي‌تواند بهينه‌سازي هزينه يا زمان را درخواست كند. يک الگوريتم زمانبندي با هدف بهينه‌سازي زمان، بايد با توجه به قيمت و توانمندي منابع گريد، عمل تخصيص آنها به كارهاي ناهمگون كاربر را طوري انجام دهد كه اجراي كارها با صرف بودجه تعيين شده و در حداقل زمان پايان يابد. در اين مقاله، يک الگوريتم مبتني بر اتوماتاهاي يادگير، براي اين منظور پيشنهاد شده است. با استفاده از شبيه‌سازي نشان داده شده است كه الگوريتم پيشنهادي در مقايسه با الگوريتم‌هاي مكاشفه‌اي گزارش شده، از كارايي بالاتري برخوردار بوده و درخواست‌هاي كاربر را در زمان كمتري انجام مي‌دهد.

كلمات كليدي: گريد محاسباتي، زمانبندي اقتصادي، بهينه‌سازي زمان، اتوماتاي يادگير

Time Optimization in Economic Computational Grids Using Learning Automata

Yasser Mahdavifar Mohammad Reza Meybodi

Soft Computing Laboratory
Computer Engineering and Information Technology Department
Amirkabir University of Technology, Tehran, Iran

Abstract

In economic computational grids, resources have price and the users must pay for executing their applications. The user determines his deadline and budget and then requests cost or time optimization. A scheduling algorithm that adopts time optimization strategy, should allocate heterogeneous grid resources to heterogeneous user jobs so that their execution finishes with specified budget and in minimum time. In this paper, a new algorithm is introduced for this purpose that uses learning automata. It is shown by using simulation that suggested algorithm has higher performance and performs users' requests in less time with respect to the reported heuristics.

Keywords: Computational Grid, Economic Scheduling, Time Optimization, Learning Automata

۱- مقدمه

گريد محاسباتي[۱] يک زيربناي سخت‌افزاري و نرم‌افزاري مي‌باشد كه دسترسي قابل‌اعتماد، پايدار، فراگير و ارزان را به توانايي‌هاي محاسباتي ديگران فراهم مي‌كند [6-3]. يک گريد محاسباتي با مجموعه‌اي از منابع ناهمگون[۲] (كامپيوترهاي شخصي، ايستگاه‌هاي كاري[۳]، كلاسترها و ابركامپيوترها)، در مقياس وسيع در ارتباط است. گريدهاي محاسباتي به تدريج به سوي تجاري شدن پيش مي‌روند و دارندگان منابع با انگيزه‌هاي مالي، منابع خود را در اختيار ديگران قرار مي‌دهند. مشتريان گريد نيز با پرداخت هزينه درخواست خود، مي‌توانند از اين منابع استفاده كنند. صاحبان و

* دانشجوي كارشناسي ارشد، دانشگاه صنعتي اميركبير، دانشكده مهندسي كامپيوتر و فناوري اطلاعات، mahdavifar@ce.aut.ac.ir

† عضو هيأت علمي دانشگاه، دانشگاه صنعتي اميركبير، دانشكده مهندسي كامپيوتر و فناوري اطلاعات، meybodi@ce.aut.ac.ir

استفاده‌کنندگان منابع داراي اهداف، استراتژي‌ها و الگوهاي عرضه و تقاضاي متفاوتي هستند. در چنين شرايطي نمي‌توان از راهکارهاي متداول براي مديريت منابع که سعي مي‌کنند ميزان کارايي کل سيستم را بهبود دهند، استفاده کرد. براي اين منظور، در سال‌هاي اخير از رويکردهاي اقتصادي براي مديريت تخصيص منابع در گريد استفاده شده است [7-10]. يکي از مدل‌هاي اقتصادي که بدين منظور استفاده شده است، مدل بازار کالا[4] مي‌باشد. در اين مدل، هر منبع داراي قيمت مشخصي است که بر اساس عرضه، تقاضا و ارزش در سيستم اقتصادي تعيين شده است.

منابع در گريد به صورت ناهمگون و توزيع شده هستند و به طور مشترک مورد استفاده قرار مي‌گيرند. از طرف ديگر، کاربران، محدوديت‌هاي مهلت[5] (زمان اتمام اجراي برنامه) و بودجه (هزينه محاسبات) را براي گريد تعيين مي‌کنند. اين مسائل، باعث پيچيدگي عمل زمانبندي برنامه کاربر مي‌گردد. با توجه به نياز کاربر، يکي از استراتژي‌هايي که يک الگوريتم زمانبندي مي‌تواند اتخاذ کند، مينيمم کردن زمان در محدوده بودجه (بهينه‌سازي زمان) مي‌باشد. تاکنون چندين الگوريتم مکاشفه‌اي[6] به منظور بهينه‌سازي زمان گزارش شده است. الگوريتم BTO توسط بويا در [11,12] پيشنهاد شده است. همچنين، براي برطرف کردن ضعف‌هاي اين الگوريتم، الگوريتم‌هاي مکاشفه‌اي ABTO، EBTO و AEBTO در [۲] ارائه و ارزيابي شده‌اند.

در اين مقاله با به‌کارگيري مجموعه‌اي از اتوماتاهاي يادگير، يک الگوريتم جديد به نام LATO[7] براي زمانبندي کارهاي مستقل از هم و با هدف بهينه‌سازي زمان در گريدهاي محاسباتي اقتصادي با مدل بازار کالا پيشنهاد مي‌شود. الگوريتم پيشنهادي، با استفاده از جعبه‌ابزار GridSim [13] شبيه‌سازي شده و کارايي آن مورد بررسي قرار گرفته است. همچنين، نتايج بدست آمده با نتايج الگوريتم‌هاي گزارش شده، مقايسه شده است. الگوريتم پيشنهادي بدليل انجام فرايند يادگيري، در مقايسه با الگوريتم‌هاي مکاشفه‌اي به زمان بيشتري نياز دارد. شبيه‌سازي‌ها نشان داده است که زمان لازم براي زمانبندي ۲۰۰ کار تقريباً ۱ ثانيه مي‌باشد. اين مقدار زمان در مقابل کاهشي که در زمان اجراي کارها بدست مي‌آيد قابل ملاحظه نمي‌باشد. همچنين الگوريتم‌هاي مکاشفه‌اي، در شرايط خاص مانند ناهمگوني بالاي کارها، داراي عملکرد مناسبي نمي‌باشند. به همين دليل استفاده از الگوريتم يادگير پيشنهادي به منظور زمانبندي کارها در گريد، داراي توجيه اقتصادي بوده و با جلب مشتريان بيشتر، موجب استفاده بهتر از منابع گريد و سودآوري بالاتر مي‌شود. ادامه مقاله بدين صورت سازماندهي شده است. در بخش ۲ اتوماتاهاي يادگير به اختصار معرفي مي‌گردد. در بخش ۳ زمانبندي اقتصادي در گريد، شرح داده مي‌شود و در بخش ٤ الگوريتم‌هاي زمانبندي پيشنهادي ارائه مي‌گردد. بخش ٥ اختصاص به نتايج شبيه‌سازي‌ها دارد و بخش ٦ نتيجه‌گيري مي‌باشد.

۲- اتوماتاهاي يادگير

يک اتوماتاي يادگير يك ماشين با حالات محدود[8] است که مي‌تواند تعداد محدودي عمل را انجام دهد. هر عمل انتخاب شده توسط محيطي احتمالي ارزيابي مي‌گردد و پاسخي به اتوماتاي يادگير داده مي‌شود. اتوماتاي يادگير از اين پاسخ استفاده نموده و عمل بعدي خود را انتخاب مي‌کند. در طي اين فرايند، اتوماتاي يادگير ياد مي‌گيرد که چگونه بهترين عمل را از بين اعمال مجاز خود انتخاب نمايد. شکل (۱) ارتباط بين اتوماتاي يادگير و محيط را نشان مي دهد. محيط را مي‌توان توسط سه تايي $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha = \{\alpha_1, ..., \alpha_r\}$ مجموعه ورودي‌ها، $\beta = \{\beta_1, ..., \beta_m\}$ مجموعه خروجي‌ها و $c = \{c_1, ..., c_r\}$ مجموعه احتمال‌هاي جريمه مي‌باشد. هرگاه β مجموعه دو عضوي باشد، محيط از نوع P مي‌باشد. در چنين محيطي $\beta_1 = 1$ به عنوان جريمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته مي‌شود. در محيط Q، $\beta(n)$ مي‌تواند به طور گسسته يک مقدار از مقادير محدود در فاصله [0,1] و در محيط از نوع S، هر مقدار در فاصله [0,1] را اختيار کند. c_i احتمال اينکه عمل α_i نتيجه نامطلوب داشته باشد، مي‌باشد. در محيط ايستا مقادير c_i در طي زمان بدون تغيير مي‌مانند؛ حال آنکه در محيط غير ايستا اين مقادير در طي زمان تغيير مي‌کنند. اتوماتاي يادگير به دو گروه اتوماتاهاي يادگير با ساختار ثابت و اتوماتاهاي يادگير با ساختار متغير تقسيم مي‌شود. در اين مقاله از اتوماتاهاي يادگير با ساختار متغير استفاده شده است که در ادامه اين بخش به شرح مختصري از آن مي‌پردازيم.

شکل (۱): رابطه بين اتوماتاي يادگير و محيط

اتوماتاهاي يادگير با ساختار متغير[9]: اتوماتاي يادگير با ساختار متغير توسط ۴تايي $\{\alpha\ ,\ \beta\ ,\ p\ ,T\ \}$ نشان داده مي‌شود كه در آن $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ مجموعهٔ عملهاي اتوماتا، $\beta = \{\beta_1, \beta_2, \ldots , \beta_m\}$ مجموعهٔ وروديهاي اتوماتا، $p = \{p_1, p_2, \ldots , p_m\}$ بردار احتمال انتخاب هريک از عملها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوريتم يادگيري مي‌باشد. در اين نوع از اتوماتاهاي يادگير، اگر عمل α_i در مرحلهٔ nام انجام شود و پاسخ مطلوب از محيط دريافت نمايد، احتمال $p_i(n)$ كاهش يافته و ساير احتمالها افزايش مي‌يابند. در هر حال، تغييرات به گونه‌اي صورت مي‌گيرد تا حاصل جمع $p_i(n)$ها همواره ثابت و مساوي يک باقي بماند. الگوريتم زير يک نمونه از الگوريتم‌هاي يادگير خطي است.

الف- پاسخ مطلوب

$$
\begin{aligned}
p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\
p_j(n+1) &= (1-a)p_j(n) \qquad \forall j \quad j \neq i
\end{aligned}
$$

ب- پاسخ نامطلوب

$$
\begin{aligned}
p_i(n+1) &= (1-b)p_i(n) \\
p_j(n+1) &= \frac{b}{r-1} + (1-b)p_j(n) \quad \forall j \quad j \neq i
\end{aligned}
$$

در روابط فوق، a پارامتر پاداش و b پارامتر جريمه است. با توجه به مقادير a و b سه حالت را مي‌توان در نظر گرفت. زمانيكه a و b با هم برابر باشند، الگوريتم را L_{RP}[10] مي‌ناميم. زمانيكه b از a خيلي كوچكتر باشد، الگوريتم را $L_{R\varepsilon P}$[11] مي‌ناميم. زمانيكه b مساوي صفر باشد، الگوريتم را L_{RI}[12] مي‌ناميم. براي اطلاعات بيشتر درباره اتوماتاهاي يادگير و كاربردهاي آن مي‌توان به [14,15] مراجعه نمود.

۳- زمانبندي اقتصادي در گريد

طول يک كار، تعداد دستورالعمل‌هاي آن مي‌باشد و بر حسب واحد MI[13] (ميليون دستورالعمل) اندازه‌گيري مي‌شود. اگر كارهاي يک برنامه كاربردي، كاملاً همگون[14] باشند (طول يكسان داشته باشند)، مي‌توان الگوريتم‌هاي زمانبندي با پيچيدگي چندجمله‌اي براي مينيمم كردن زمان در محدوده بودجه، طراحي كرد؛ ولي هنگامي كه كارها داراي ناهمگوني باشند (طول‌هاي متفاوت داشته باشند)، مسأله زمانبندي با هدف بهينه‌سازي زمان به يک مسأله NP-Complete تبديل مي‌شود.

قيمت اعلام شده براي يک منبع، مقدار هزينه استفاده از آن در واحد زمان مي‌باشد و قيمت مفيد يک منبع به مقدار هزينه استفاده از آن براي اجراي هر MI گفته مي‌شود. در اين مقاله، واژه‌هاي قيمت، ارزاني و گراني با توجه به قيمت مفيد منبع به كار مي‌روند.

كاربر گريد پس از اينكه محدوديت‌هاي مهلت زماني و بودجه خود را مشخص كرد، ممكن است بهينه‌سازي هزينه و يا زمان را درخواست كند. در صورتي كه كاربر خواستار بهينه‌سازي زمان باشد، الگوريتم زمانبندي بايد با صرف بودجه‌اي كه كاربر تعيين كرده است، كمترين زمان را براي اجراي كارها بدست آورد و در صورتي كه كاربر خواستار بهينه‌سازي هزينه باشد، الگوريتم زمانبندي بايد در محدوده مهلتي كه كاربر تعيين كرده است، كمترين هزينه را براي اجراي كارها بدست آورد.

الگوريتم‌هاي زمانبندي با هدف بهينه‌سازي هزينه و يا زمان از دو نوع مي‌باشند: زمانبندي مرحله‌اي و زمانبندي يكباره. الگوريتم‌هايي كه از روش زمانبندي مرحله‌اي استفاده مي‌كنند، در طول اجراي خود به تدريج كارها را به منابع موجود در گريد واگذار مي‌كنند. در مقابل، الگوريتم‌هايي كه رويكرد زمانبندي يكباره را اتخاذ مي‌كنند، يک نگاشت از كارهاي برنامه كاربر به منابع موجود را توليد مي‌كنند كه براي بررسي به بخش كنترل پذيرش داده مي‌شود. پس از بررسي، اگر نگاشت بدست آمده، نيازمندي‌هاي كاربر را برآورده كند، كارها طبق اين نگاشت به منابع براي اجرا واگذار مي‌گردد. براي توضيحات بيشتر مي‌توان به [۱] مراجعه كرد.

الگوريتم‌هاي بهينه‌سازي هزينه و يا زمان، داراي چند مرحله مشترک ابتدايي مي‌باشند. اين مراحل عبارتند از يافتن، تجارت و مرتب‌سازي منابع. الگوريتم‌هايي كه از روش زمانبندي يكباره استفاده مي‌كنند، در پايان نيز داراي دو مرحله مشترک كنترل پذيرش و توزيع هستند. در ادامه توضيحاتي درباره اين مراحل داده مي‌شود [۱].

- يافتن منابع[15]: شناسايي منابعي كه مي‌توانند در اجراي كارها مورد استفاده قرار بگيرند و همچنين بدست آوردن توانمندي‌ها و ويژگي‌هاي آنها كه از طريق سرويس اطلاعاتي گريد انجام مي‌گيرد.
- تجارت منابع[16]: شناسايي هزينه هر يک از منابع در واحد زمان (G$/sec) ، شناسايي ميزان توانمندي منبع در واحد زمان (نرخ اجراي ميليون دستورالعمل در ثانيه، MI/sec) و محاسبه قيمت واقعي و مفيد منبع كه همان هزينه اجراي يک ميليون دستورالعمل در ثانيه مي‌باشد (G$/MI).

- مرتب‌سازي منابع: يک الگوريتم بهينه‌سازي، حتماً به يک ترتيب خاص از منابع بر حسب قيمت آنها نياز دارد. در اين مرحله، منابع به ترتيب صعودي از نظر قيمت مفيد مرتب‌سازي مي‌شوند. در واقع، اولويت واگذاري کار با منابع ارزان‌تر مي‌باشد. اگر دو منبع داراي قيمت مفيد يکسان باشند، منبعي که توانمندي بيشتري دارد در اولويت قرار مي‌گيرد.
- کنترل پذيرش[۱۷]: هزينه کل اجراي کارها بر روي منابع بر طبق زمانبندي مشخص و با بودجه تعيين شده توسط کاربر مقايسه مي‌گردد. همچنين زمان تقريبي اجراي برنامه بر طبق زمانبندي محاسبه و سپس با مهلت تعيين شده توسط کاربر مقايسه مي‌شود. در صورتي که هزينه‌ها در محدوده بودجه باشد و اجراي برنامه قبل از مهلت زماني پايان مي‌يابد، کاربر پذيرفته مي‌شود.
- توزيع[۱۸]: در اين مرحله در صورت پذيرش کاربر، کارها بر طبق نگاشت تعيين شده، به منابع واگذار مي‌شود.

بدليل اينکه منابع در گريد، کارها را به طور موازي اجرا مي‌کنند، زماني که براي اجراي تمامي کارها صرف مي‌شود، مدت زماني است که پرکارترين منبع (منبعي که بيشترين زمان را براي اجراي کارهاي خود نياز دارد) براي اجراي کارهاي واگذار شده به او نياز دارد. در واقع، پس از اينکه پرکارترين منبع، پردازش کارهاي خود را به پايان رساند، گريد مي‌تواند نتيجه اجراي برنامه کاربردي را به کاربر تحويل دهد. به اين ترتيب، منظور از اينکه يک الگوريتم بهينه‌سازي زمان بايد زمان اجراي کارها را تا حد ممکن کاهش دهد، اين است که الگوريتم بايد زمان اجراي کارها را در پرکارترين منبع کاهش دهد. اين عمل با انتقال کارهاي نگاشت شده به اين منبع به منابعي که بتوانند زودتر کار را اجرا کنند، انجام مي‌گيرد. البته انجام اين عمل تا زماني ممکن است که بودجه کافي براي انتقال کارها به منابع سريع‌تر موجود باشد.

الگوريتم پيشنهادي در اين مقاله، از روش زمانبندي يکباره استفاده مي‌کند و بنابراين امکان تضمين محدوديت‌هاي بودجه و مهلت براي چندين کاربر به طور همزمان وجود خواهد داشت. براي توضيحات بيشتر مي‌توان به [۱] مراجعه کرد.

۴- الگوريتم پيشنهادي (LATO)

الگوريتم LATO الگوريتمي است مبتني بر اتوماتاهاي يادگير که عمل پيچيده بهينه‌سازي زمان را با کارايي بالاتري در مقايسه با الگوريتم‌هاي مکاشفه‌اي موجود انجام مي‌دهد. اين الگوريتم براي زمانبندي کارهاي برنامه کاربردي متعلق به کاربر، هر کدام از کارها را به يک اتوماتاي يادگير، مجهز مي‌کند. هر يک از اين اتوماتاهاي يادگير، داراي ساختار متغير بوده و داراي تعداد اعمال برابر با تعداد منابع در گريد مي‌باشد. عمل iام يک اتوماتاي يادگير، انتخاب منبع iام در گريد مي‌باشد.

يک تکرار از الگوريتم LATO بدين قرار است. در ابتدا هر کدام از اتوماتاهاي يادگير، يکي از اعمال خود را که يکي از منابع موجود در گريد مي‌باشد، انتخاب مي‌کند. سپس کار متناظر با اين اتوماتاي يادگير، در صف موقت منبع به نام صف انتساب که در ابتداي هر تکرار خالي مي‌شود، قرار داده مي‌شود. از اين طريق، يک نگاشت از کارها به منابع در گريد ايجاد مي‌شود. سپس هزينه نگاشت ايجاد شده، محاسبه مي‌گردد. فرايند دادن پاداش و يا جريمه به اعمال انتخابي اتوماتاهاي يادگير با توجه به هزينه محاسبه شده و نکات زير که بيان کننده شرايطي است که بايستي در يک زمانبندي مورد توجه قرار گيرد، انجام مي‌گيرد. شرايطي که بايستي در يک زمانبندي مورد توجه قرار گيرد عبارتند از: ۱- مجموع هزينه اجراي کارها نبايد از بودجه تعيين شده توسط کاربر بيشتر باشد، ۲- زمان اجراي کارهاي واگذار شده به پرکارترين منبع نبايد با زمان اجرا در منابع ديگر اختلاف زيادي داشته باشد و ۳- هر کار بايد به منبعي واگذار شود که بتواند اجراي آن کار را زودتر از بقيه منابع به اتمام برساند. در هر مرحله از الگوريتم، فرايند دادن پاداش و يا جريمه به شرح زير انجام مي‌گيرد.

اگر مجموع هزينه از مقدار بودجه تعيين شده، فراتر رود، محيط، اعمال انتخابي اتوماتاهاي يادگير کارهايي را که هزين هزيادي براي اجرا لازم دارند، بر طبق الگوريتم يادگيري جريمه مي‌کند. عمل جريمه کردن، از انتهاي صف کارهاي گران‌ترين منبع استفاده شده (گران‌ترين منبعي که حداقل يک کار به آن منتسب شده است) شروع مي‌شود. بعد از جريمه کردن هر کار و حذف آن از صف انتساب منبع انتخابي، تفاوت هزينه اجراي کار بر روي منبع انتخابي و هزينه اجراي آن بر روي ارزان‌ترين منبع موجود، از مقدار کل هزينه‌هاي اجرا کسر مي‌شود. عمل جريمه دادن تا زماني ادامه مي‌يابد که مقدار کل هزينه از مقدار بودجه تعيين شده، بيشتر شود. در واقع با اين جريمه، کارها به سمت استفاده از منابع ارزانتر هدايت مي‌شوند.

در هر تکرار همچنين اگر يکي از منابع استفاده شده در زمانبندي (منابعي که حداقل يک کار به آنها منتسب شده دارند)، به غير از آخرين (گران‌ترين) منبع استفاده شده، بتواند يک کار را زودتر از پرکارترين منبع، پردازش کند، عمل انتخاب شده اتوماتاي يادگير متناظر با اين کار بر طبق الگوريتم يادگيري جريمه مي‌شود. اين عمل به توزيع متعادل کارها بين منابع کمک مي‌کند. در آخرين مرحله در يک تکرار، کارهايي که توانسته‌اند زمان اجراي کمتر يا برابر با زمان اجراي تکرار قبلي براي خود بدست بياورند، پاداش مي‌گيرند؛ البته به شرطي که در همين تکرار، جريمه نشده باشند. از اين طريق، يک کار سعي مي‌کند منبعي را براي اجراي خود انتخاب کند که در کمترين زمان، اجرايش را به اتمام برساند.

تعداد تکرارهاي الگوريتم بايستي با توجه به پارامترهاي گريد تعيين شود. در شبيه‌سازي‌هاي اين مقاله، اين تعداد ۱۰۰۰۰ در نظر گرفته شده است. الگوريتم LATO در شکل (۲) آورده شده است.

۱. مرتب‌سازي: کارها را از نظر طول به صورت نزولي مرتب کن.

۲. يادگيري: به تعداد ۱۰۰۰۰ بار تکرار کن:

a. صف‌هاي انتساب منابع را خالي کن.

b. انتخاب: براي هر کار به ترتيب انجام بده:

i. با استفاده از اتوماتاي يادگير مربوطه، منبعي را براي واگذاري، انتخاب کن.

ii. کار را در صف انتساب منبع انتخابي قرار بده.

c. جريمه (۱): تا هنگامي که مجموع هزينه اجراي کارها بر روي منابع انتخاب شده از بودجه تعيين شده بيشتر است، تکرار کن:

i. از انتهاي صف انتساب در گران‌ترين منبع استفاده شده، يک کار را انتخاب کن.

ii. تفاوت هزينه اجراي کار بر روي اين منبع و هزينه اجرا بر روي ارزان‌ترين منبع موجود را از مجموع هزينه اجراي کارها کم کن.

iii. عمل انتخابي اتوماتاي يادگير اين کار را جريمه کن.

d. جريمه (۲): آخرين کار منتسب شده به پرکارترين منبع را در نظر بگير. اگر يکي از منابع استفاده شده به جز منبع آخر (گران‌ترين منبع استفاده شده) مي‌تواند اجراي اين کار را زودتر به اتمام برساند، اين کار را جريمه کن.

e. پاداش: براي هر کار انجام بده:

اگر عمل انتخابي اتوماتاي يادگير اين کار جريمه نشده است و اگر زمان اجراي اين کار بر روي منبع انتخابي کمتر و يا برابر با زمان اجراي آن در تکرار قبلي مي‌باشد، عمل انتخابي اتوماتاي يادگير اين کار را پاداش بده.

۳. زمانبندي: براي هر کار به ترتيب نزولي طول، با استفاده از اتوماتاي يادگير منتسب به آن، منبعي را انتخاب کرده و کار را به آن منبع واگذار کن.

شکل (۲): الگوريتم **LATO**

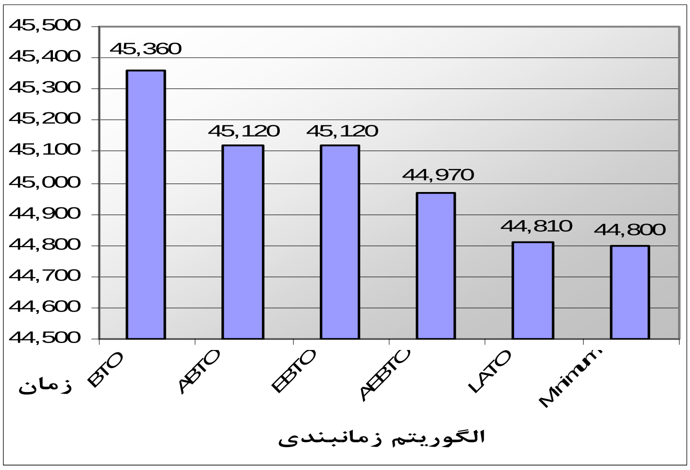
۵- نتايج شبيه‌سازي‌ها

الگوريتم پيشنهادي با استفاده از جعبه‌ابزار GridSim [13] شبيه‌سازي شده و نتايج آن با نتايج بدست آمده از الگوريتم‌هاي مکاشفه‌اي مقايسه شده است. نتايج گزارش شده، ميانگين ۲۰ بار شبيه‌سازي مي‌باشد. محيط شبيه‌سازي شده براي گريد، شامل تعدادي منبع و يک کاربر است. کليه منابع محاسباتي داراي يک پردازنده هستند که مشخصات آنها در جدول (۱) داده شده است. همانطور که مشاهده مي‌شود، ناهمگوني بالايي براي منابع در نظر گرفته شده است. برنامه کاربر از ۲۰۰ کار مستقل از هم تشکيل شده است که طول هر کار به صورت تصادفي از محدوده (۲۰۰۰۰...۱۰۰۰۰) انتخاب مي‌شود. گستردگي اين محدوده، منجر به ناهمگوني کارها مي‌گردد. کاربر مقدار ۱۲۰۰۰۰ را براي بودجه تعيين کرده و بهينه‌سازي زمان را درخواست مي‌کند. براي هر کار، اتوماتاي يادگير از نوع L_{RP} با مقادير $a = b = 0.01$ مي‌باشد.

جدول (۱): پيکربندي منابع

نام منبع	نرخ اجرا (MI/sec)	قيمت (G$/sec)	قيمت مفيد (G$/1000MI)
R1	۱۰۰	۰,۵	۵
R2	۱۸۰	۱,۰	۵,۵۵
R3	۲۴۰	۱,۵	۶,۲۵
R4	۲۸۰	۲	۷,۱۴
R5	۳۰۰	۲,۵	۸,۳۳
R6	۴۰۰	۴	۱۰
R7	۵۰۰	۶	۱۲
R8	۶۰۰	۹	۱۵

در شکل (۳)، الگوريتم‌هاي مکاشفه‌اي BTO، ABTO، EBTO و AEBTO [۲] و الگوريتم پيشنهادي LATO با يکديگر مقايسه شده‌اند. همانطور که مشاهده مي‌شود، الگوريتم LATO کمترين زمان را براي اجراي کارها بدست آورده و در مقايسه با بهترين الگوريتم مکاشفه‌اي گزارش شده (AEBTO)، بيش از ۱۵۰ واحد (۰,۳ درصد) کاهش زمان را نشان مي‌دهد. اختلاف زمان بدست آمده توسط الگوريتم LATO با حداقل زمان مورد نياز براي اجراي کارها بسيار کم است. براي بدست آوردن حداقل زمان مورد نياز براي اجراي کارها، از روش ارائه شده در [۲] استفاده شده است.



شکل (۳): نتايج الگوريتم‌هاي زمانبندي با هدف بهينه‌سازي زمان

الگوريتم پيشنهادي بدليل انجام فرايند يادگيري، در مقايسه با الگوريتم‌هاي مکاشفه‌اي به زمان بيشتري نياز دارد. شبيه‌سازي‌ها نشان داده است که زمان لازم براي زمانبندي ۲۰۰ کار تقريباً ۱ ثانيه مي‌باشد. اين مقدار زمان در مقابل کاهش زمان اجراي کارها که در صورت استفاده از اين الگوريتم‌ها بدست مي‌آيد، قابل ملاحظه نمي‌باشد. همچنين الگوريتم‌هاي مکاشفه‌اي، در شرايط خاص مانند ناهمگوني بالاي کارها، داراي عملکرد مناسبي نمي‌باشند. به همين دليل، استفاده از الگوريتم پيشنهادي به منظور زمانبندي کارها در گريد، داراي توجيه اقتصادي مي‌باشد؛ زيرا که حتي يک کاهش پايين در زمان اجراي کارها منجر به جلب مشتريان بيشتر، استفاده بهتر از منابع در گريد و سودآوري بالاتر براي گريد مي‌شود.

۶- نتيجه‌گيري

در اين مقاله، الگوريتمی مبتني بر اتوماتاهاي يادگير براي زمانبندي در گريدهاي محاسباتي اقتصادي با مدل بازار کالا به منظور مينيمم کردن زمان در محدوده بودجه تعيين شده، پيشنهاد گرديد. اين الگوريتم با استفاده از جعبه‌ابزار GridSim شبيه‌سازي شده و کارايي آن در محيط گريد با منابع و کارهاي ناهمگون، مورد بررسي قرار گرفت. نشان داده شد که الگوريتم پيشنهادی (LATO) در مقايسه با الگوريتم‌هاي مکاشفه‌اي گزارش شده، از کارايي بالاتري برخوردار بوده و درخواست‌هاي کاربر را در زمان کمتري اجرا مي‌کند.

مراجع

[۱] ياسر مهدوي‌فر و محمدرضا ميبدي، الگوريتم‌هاي زمانبندي براي بهينه‌سازي هزينه در گريدهاي محاسباتي اقتصادي، گزارش فني، آزمايشگاه سيستم‌هاي نرم‌افزاري، دانشکده مهندسي کامپيوتر و فناوري اطلاعات، دانشگاه صنعتي اميرکبير، تهران، دي ۱۳۸۴.

[۲] ياسر مهدوي‌فر و محمدرضا ميبدي، الگوريتم‌هاي زمانبندي براي بهينه‌سازي زمان در گريدهاي محاسباتي اقتصادي، گزارش فني، آزمايشگاه سيستم‌هاي نرم‌افزاري، دانشکده مهندسي کامپيوتر و فناوري اطلاعات، دانشگاه صنعتي اميرکبير، تهران، ارديبهشت ۱۳۸۵.

[3] I. Foster and C. Kesselman, *The Grid: Blueprint for a Future Computing Infrastructure*, Morgan Kaufmann, San Francisco, 1999.

[4] I. Foster, C. Kesselman and S. Tuecke, *"The anatomy of the Grid: Enabling scalable virtual organizations"*, International Journal of Supercomputer Applications, 2001.

[5] M. Baker, R. Buyya and Domenico Laforenza, *"Grids and Grid technologies for wide-area distributed computing"*, The Journal of Concurrency and Computation: Practice and Experience, Vol. 14, Issue 13-15, Nov. 2002.

[6] V. Berstis, *Fundamentals of Grid Computing*, IBM Redbooks, November 2002.

[7] 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, April 2001.

[8] R. Buyya, D. Abramson, and J. Giddy, *"An Economy Driven Resource Management Architecture for Global Computational Power Grids"*, Proceedings of the 2000 International Conference on Parallel and Distributed Processing Techniques and Applications, June 2000.

[9] R. Buyya, D. Abramson, and J. Giddy, *"Nimrod-G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid"*, The 4th International Conference on High Performance Computing in Asia-Pacific Region, May 2000.

[10] 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, May 2002.

[11] 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 2nd International Workshop on Active Middleware Services, August 2000.

[12] 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, April 2002.

[13] R. Buyya and M. Murshed, *"GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing"*, Journal of Concurrency and Computation: Practice and Experience, pp. 1-32, May 2002.

[14] K. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*, Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[15] K. Najim and A. S. Poznyak, *Learning Automata: Theory and Application*, Elsevier Science Ltd., Tarrytown, NY, 1994.

زیرنویس‌ها

[1] Computational Grids
[2] Heterogeneous
[3] Workstations
[4] Commodity Market Model
[5] Deadline
[6] Heuristic
[7] Learning Automata Time Optimization
[8] Finite State Machine
[9] Variable structure
[10] Linear reward penalty
[11] Linear reward epsilon penalty
[12] Linear reward inaction
[13] Million Instruction
[14] Homogeneous
[15] Resource Discovery
[16] Resource Trading
[17] Admission Control
[18] Dispatching