

# الگوریتم توزیع شده زمانبندی کارهای محاسبات توری مبتنی بر اتوماتای یادگیر

آسیه ایرانپور مبارکه

دانشجوی کارشناسی ارشد گروه مهندسی کامپیوتر، واحد بین الملل قشم، دانشگاه آزاد اسلامی، قشم، ایران  
پست الکترونیکی: [asiye.iranpoor@gmail.com](mailto:asiye.iranpoor@gmail.com)

علی اصغر صفائی

استادیار گروه انفورماتیک پزشکی، دانشکده علوم پزشکی، دانشگاه تربیت مدرس، تهران، ایران  
پست الکترونیکی: [aa.safaei@modares.ac.ir](mailto:aa.safaei@modares.ac.ir)

امیر صحافی

استادیار گروه مهندسی کامپیوتر، واحد تهران جنوب، دانشگاه آزاد اسلامی، تهران، ایران  
پست الکترونیکی: [sahafi@iau.ac.ir](mailto:sahafi@iau.ac.ir)

## چکیده

محاسبات توری یک زیرساخت سخت افزاری و نرم افزاری می باشد که دسترسی به منابع مختلف را به شکلی قابل اعتماد، پایدار، فراگیر و ارزان فراهم می کند. منابع توزیع شده، فاصله های جغرافیایی زیاد، معماری ها و سیستم عامل های ناهمگون ماشین های مختلف، تفاوت ها در روش های دستیابی به هر یک از منابع، متغیر بودن محیط به لحاظ منابع در دسترس و در نهایت وجود سیاست های مختلف در دسترسی به منابع، همگی جزء مسائل پیچیده ای هستند که در یک محیط توری وجود دارند. به منظور حصول کارآمدی مناسب در چنین محیطی، زمانبندی کارها یکی از موضوعات مهم در طراحی محیط های توری می باشد. عملکرد سیستم تور به روش زمانبندی کار در آن وابسته است و در صورت زمانبندی نامناسب محیط تور، ناکارآمد خواهد بود. از آنجا که زمانبندی کار جزء مسائل پیچیده به حساب می آید، الگوریتم های قطعی کارایی لازم را برای حل این مسئله نخواهند داشت. در این مقاله، یک الگوریتم زمانبندی کارها در محیط توری بر پایه اتوماتای یادگیر ارائه شده است. در الگوریتم پیشنهادی، ابتدا یک گره به طور تصادفی انتخاب، و زمانبندی کارها در این گره با استفاده از اتوماتای یادگیر انجام می گیرد. با توجه به احتمال هر اقدام در اتوماتای یادگیر، کار مورد نظر انتخاب و به تور ارسال می شود. پاداش و جریمه اتوماتا براساس وجود یا عدم وجود کار در زمانبندی می باشد. به منظور بررسی کارایی الگوریتم پیشنهادی آزمایش های مختلفی در محیط های تور با مقیاس های کوچک، متوسط و بزرگ صورت گرفته است. نتایج ارزیابی نشان می دهد که وضعیت پارامترهای مهم یک محیط تور شامل زمان اتمام کار، زمان اجرای کارها و تعادل بارگذاری در الگوریتم پیشنهادی به مراتب بهتر از سایر الگوریتم های مشابه می باشد.

**واژگان کلیدی:** الگوریتم زمانبندی، تعادل بارگذاری، زمان اتمام کار، زمان اجرای کار، محاسبات توری

## 1- مقدمه

امروزه فناوری جدیدی به نام تور به عرصه ارتباطات قدم نهاده است که بر اساس آن می توان به کامپیوترهای شخصی خود اجازه داد که وقتی از آن استفاده نمی کنیم به شبکه جهانی متصل شوند و به سیستم های بزرگ تحقیقاتی اجازه دهند از منابع آزاد و بلااستفاده سیستم ما استفاده نماید. محاسبات توری در واقع به ما اجازه می دهد با استفاده از منابع سیستم های متصل به شبکه و ایجاد یک منبع

بزرگ از سرویس‌ها و امکانات این سیستم‌ها، منبعی بزرگ و قدرتمند به وجود آوریم که توانایی انجام دادن عملیات بسیار پیچیده مانند پردازش‌های موازی و سنگین یا محاسبات شبکه‌ای که یک سیستم به تنهایی نمی‌تواند انجام دهد را داشته باشد. با استفاده از این رویکرد می‌توان منابع کامپیوترهای ناهمگون را به اشتراک گذاشت. به عبارت دیگر، تور عبارت است از شبکه گسترده‌ای با توان محاسباتی بالا که امکان اتصال به اینترنت را هم دارد. یک محیط تور از کامپیوترهای همگن اختصاصی تشکیل نمی‌شود، بلکه مجموعه‌ای است از کامپیوترهای توزیع شده در سطح اینترنت و یا اینترانت‌های متعدد که به صورت غیراختصاصی و تحت یک پروتکل ارتباطی، از طریق یک سیستم مدیریت تور با یکدیگر در ارتباط می‌باشند. هدف اصلی تور استفاده از این منابع مشترک مانند قدرت پردازنده، پهنای باند، پایگاه‌های اطلاعاتی و در اختیار گذارتن آن‌ها برای سایر کامپیوترهای متصل به این تور می‌باشد.

با استفاده از تور در یک سازمان یا شرکت بزرگ می‌توان از منابع بلا استفاده کامپیوترهای سازمان کمال استفاده را برد و سرعت پردازش اطلاعات در سیستم‌هایی که با کمبود حافظه مواجه هستند را جبران نمود. از طرف دیگر، سرعت نرم‌افزارهایی که از این منبع بزرگ سخت‌افزاری استفاده می‌کنند، بسیار بالاتر خواهد بود و در نتیجه می‌توانیم به فکر درست کردن نرم‌افزارهایی با قابلیت‌های بالاتر باشیم و منابع بیشتری را در اختیار استفاده‌کنندگان قرار دهیم.

تور می‌تواند مزایای زیادی برای مدیران و برنامه‌نویسان داشته باشد. در مقایسه با دیگر مدل‌های محاسبات از قبیل رایانه‌های بزرگ، کارخواه/کارساز<sup>1</sup> یا چند لایه‌ای<sup>2</sup>، هدف سیستم‌های طراحی شده و پیاده‌سازی شده در روش محاسبات توری در حوزه فناوری اطلاعات، کیفیت بالای سرویس‌ها، هزینه کمتر و انعطاف‌پذیری بیشتر است. مثلاً با آن می‌توان برنامه‌هایی را که نیاز به حافظه زیادی دارند اجرا نمود و به اطلاعات، دسترسی آسان‌تری پیدا کرد. فناوری‌های تور در واقع می‌توانند از منابع و سیستم‌های غیرمتمرکز پشتیبانی کنند و امکان ارتباط سیستم‌ها را با هم فراهم سازند. وقتی برای اولین بار فناوری تور ابداع شد، هدف آن تنها به اشتراک گذاشتن منابع سیستم و در اختیار داشتن سیستمی قدرتمند بود و به طور کلی بیشتر در اختیار مؤسسات تحقیقاتی قرار داشت. اما امروزه از تور توقع بیشتری می‌رود و اهمیت بیشتری پیدا کرده است؛ به‌ویژه در تجارت الکترونیکی و سیستم‌های تجاری غیرمتمرکز و توزیع‌یافته.

اگر تصور می‌کنید سیستم‌های تور پیچیده‌اند و ممکن است کار با آن‌ها مشکل باشد، کاملاً درست فکر می‌کنید. مثلاً سیستم‌های تور باید به سرعت قادر باشند منابع سیستم‌های متصل به آن‌ها را شناسایی کنند و در عین حال نباید از سرعت و کارایی این سیستم‌ها بکاهند. وقتی یک نرم‌افزار یا مجموعه‌ای از نرم‌افزارها بخواهند از تور استفاده کنند، مسئله به اشتراک گذاشتن اطلاعات، تکه تکه کردن این اطلاعات، انتقال آن یا امنیت کامل و مدیریت این اطلاعات کار دشواری است و کاربر تور فقط اطلاعاتی را وارد می‌کند و نمی‌داند در داخل سیستم چه می‌گذرد. در نتیجه این سیستم‌ها باید طوری این کار را انجام دهند که کاربران تور بدون نگرانی از انتشار اطلاعات سری خود یا بروز اشتباه در محاسبات، به تور اطمینان کنند.

## 2- پیشینه پژوهش

برمان و همکاران در سال 2003 در مقاله‌ای به بیان بحث تور که تقریباً از سال 1990 شروع شده، پرداختند که در واقع بستری است برای بهبود کارها و سرعت پردازش و انجام پردازش‌های سنگین و موازی با کمترین هزینه. تور تنها یک مجموعه از منابع نمی‌باشد بلکه سیستمی با یکسری ویژگی‌های پویاست [9].

فوستر و همکاران در سال 2004 در مقاله‌ای به بیان این‌که محبوبیت اینترنت و در دسترس بودن کامپیوترهای قوی و شبکه‌های پرسرعت به‌عنوان قطعات ارزان باعث تغییر در روش استفاده از کامپیوترهای امروزی شده است، پرداختند. این مقوله، امکان استفاده از منابع چند کاربره و توزیع شده جغرافیایی جهت حل مسائل بزرگ در صنعت و تجارت را فراهم آورده است. تحقیقات اخیر در این زمینه بحث جدیدی به نام محاسبات توری را مطرح کرده است [7].

بويا و همکاران در سال 2012 در مقاله‌ای به بیان این پرداخته‌اند که در محیط تور رفتار و ویژگی‌های اشیا و موجودیت‌ها بیشتر شبیه سیستم‌های تصادفی می‌باشد، فعالیت‌ها و واکنش‌های موجودیت‌ها در تور پیرو یکسری قواعد مشخص می‌باشد. اگر مطالعه

<sup>1</sup> client/server

<sup>2</sup> Multitier

دقیقی روی این فعالیت‌ها انجام دهیم قادر خواهیم بود یکسری واکنش‌ها، رفتارها و فعالیت‌های بعدی موجودیت‌ها را تخمین بزنیم [1].

کلوریس و همکاران در سال 2011 در مقاله‌ای بیان کردند که بدون شک آینده سیستم‌های اطلاعاتی و پایگاه داده بر این اساس است که می‌توان برخی عملیات سنگین مثل داده کاوی، پردازش تصویر و غیره را با سرعت بسیار زیاد در محیط‌های توری اجرا کرد [2].

متیاس و همکاران در سال 2010 در مقاله‌ای به بیان محاسبات توزیع شده روی یک سیستم کامپیوتری پرداختند که در آن یک تعداد عناصر پردازشی مستقل از طریق یک شبکه کامپیوتری به هم متصل شده و وظایف محاسباتی را به صورت مشترک انجام می‌دهند. منظور از مولفه یا عنصر پردازشی همان سیستم‌های محاسباتی (مثل کامپیوترهای موجود در شبکه) است که می‌تواند به خودی خود برنامه را اجرا کند [3].

رانتاز و همکاران در سال 2008 در مقاله‌ای به ارائه میان‌افزاری در یک سیستم محاسباتی توزیع شده پرداختند که به عنوان لایه نرم‌افزاری بین سیستم عامل و برنامه‌ها قرار می‌گیرد و اجرای چند فرآیند بر روی یک یا چند ماشین در شبکه را امکان‌پذیر می‌سازد. میان‌افزار برای انتقال برنامه‌های بزرگ رایانه‌ها به برنامه‌های کارخواه/کارساز ضروری است. این فناوری در سال‌های 1990 تکامل یافته و با رشد برنامه‌های مبتنی بر شبکه اهمیت بیشتری پیدا کرد [4].

روینی در سال 2006 به بیان این موضوع پرداخت که جهت دستیابی به پتانسیل عظیم موجود در منابع توزیع شده، الگوریتم‌های زمانبندی کارا و موثر از اهمیت خاصی برخوردار هستند. متاسفانه الگوریتم‌های زمانبندی موجود در سیستم‌های توزیع شده و سیستم‌های موازی قدیمی که معمولاً روی منابع اختصاصی و همگن (مانند خوشه‌های کامپیوتری) اعمال می‌شوند را نمی‌توان روی تور به کار گرفت [5].

بویا و همکاران در سال 2005 در [6] نشان دادند که ناهمگونی و استقلال کاربران تور نیز در پارامترهایی مانند انواع مختلف کارها، نیازمندی‌های منابع، مدل‌های کارایی و اهداف بهینه‌سازی نشان داده می‌شود. این ناهمگونی‌ها باعث شدند ایده‌های جدیدی مانند زمانبندی سطح کاربرد و مدل اقتصادی تور مطرح شده و برای زمانبندی تور به کار گرفته شوند. زوهو و همکاران در سال 2003 نشان دادند که مدل‌های زمانبندی قدیمی در عمل برای سیستم‌های تور مناسب نیستند. دلیل این امر فرضیات پایه‌ای موجود در سیستم‌های قدیمی است که متاسفانه در شرایط تور این فرضیات وجود ندارند. سیستم تور خصوصیات منحصر به فردی دارد که باعث پیچیدگی طراحی الگوریتم‌های زمانبندی می‌شود [8].

در سال 2002 فریا و همکاران به بیان این موضوع پرداختند که ابزارهای نظارت و حسابرسی در تور ممکن است اطلاعات مربوط به وضعیت‌ها و فعالیت‌های قبلی اشیای درون تور را ثبت کنند. این مسئله منجر به ایجاد حجم بالایی از داده‌های مربوط به بهره‌وری سرویس‌های تور و اطلاعات مربوط به وضعیت سیستم تور می‌شود و لازم است این داده‌ها در سطوح مختلف دسته‌بندی، تجزیه و تحلیل شوند و سپس از تکنیک‌های داده کاوی برای استخراج اطلاعات مهم از درون این داده‌های حجیم استفاده شود [10].

### 3- پس زمینه

امنیت سیستم‌های تور نیز بسیار حائز اهمیت است. کاربران این سیستم‌ها باید از دسترسی به منابعشان در سیستم اطلاع حاصل کنند و بدانند کدام کاربر به اطلاعات آن‌ها دسترسی دارد. علاوه بر این قابلیت اطمینان و سرعت این سیستم‌ها بسیار حائز اهمیت است که اگر سیستم‌های تور از سرعت کافی برخوردار نباشند، کاربران تور از استفاده از این سیستم‌ها دلسرد می‌شوند [11].

سیستم مدیریت منابع تور<sup>3</sup>، مسئول کنترل منابع تور و هدف آن بالا بردن کارایی سیستم تور است. سرویس اطلاعات تور یک بخش از سیستم مدیریت منابع تور است که اطلاعات ایستا و پویای منابع تور را مهیا می‌کند. زمانبندی تور یک بخش دیگر از سیستم‌های مدیریت منابع تور است. زمانبندی با استفاده از اطلاعات منابع که توسط سرویس اطلاعات تور مهیا می‌شوند و اطلاعات کارها و وظایف آن‌ها، یک انتساب از وظایف به منابع ایجاد می‌کند. به این انتساب زمانبندی گفته می‌شود. یک زمانبندی کارا باعث کاهش زمان اتمام و کاهش هزینه اجرای کارها می‌شود. فرآیند زمانبندی تور را می‌توان در سه مرحله زیر مطرح کرد [8]:

<sup>3</sup> Grid Resource management system

### 1- کشف و پالایش منابع

### 2- انتخاب و زمانبندی منابع براساس اهداف مشخص شده

### 3- انتساب وظایف به منابع

زمانبندی‌های تور برخلاف زمانبندی‌های سیستم‌های موازی و توزیع شده معمولاً قادر به کنترل مستقیم منابع نیستند بلکه همانند واسطه‌ها یا عامل‌ها کار می‌کنند [6]، یا گاهی اوقات همراه با کار مطرح می‌شوند. مانند زمانبندی سطح برنامه که در [12] پیشنهاد شده است.

امروزه افزایش کارایی تور یک مسئله مهم است. جهت افزایش کارایی تور به یک زمانبندی درست و کارآمد احتیاج است. متأسفانه طبیعت پویای منابع تور و همچنین تقاضاهای مختلف کاربران، باعث پیچیدگی مسئله تور شده است. پویایی کارایی منابع ناشی از ناهمگونی، خودمختاری و اشتراکی بودن منابع تور است. هدف مسئله زمانبندی تور، انتساب بهینه کارها به منابع می‌باشد. مطالعه زمانبندی تور از این جهت حائز اهمیت است که سیستم‌های دنیای واقعی معمولاً به طور فیزیکی یا عملیاتی توزیع شده و ناهمگون می‌باشند. سیستم‌هایی مانند کنترل ترافیک هوایی، نجوم، پزشکی، زیست‌شناسی، نظامی و... از این دسته هستند.

**اتوماتای یادگیر** فرایندی است که طی آن موجودات زنده مطالب مختلف را یاد می‌گیرند. تحقیقاتی که در این زمینه انجام شده است، در دو شاخه کلی متمرکز می‌باشد:

1. درک فرآیندی که منجر به یادگیری موجودات زنده می‌گردد در طی آن، اقدام به یادگیری می‌نمایند.

2. به دست آوردن روش‌هایی که با استفاده از آن‌ها بتوان این قابلیت یادگیری را در ماشین‌ها ایجاد نمود.

یادگیری به روش‌های مختلفی تعریف شده است که از جمله می‌توان به اعمال یکسری تغییرات در کارایی سیستم براساس تجارب قبلی اشاره کرد. مهم‌ترین مشخصه این سیستم یادگیر، قابلیت آن در افزایش کارایی در طول زمان می‌باشد. اگر به صورت ریاضی این تعریف را بیان نماییم می‌گوییم هدف یک سیستم یادگیر، بهینه‌سازی یک عملکرد می‌باشد که امکان شناسایی آن به‌طور کامل وجود ندارد.

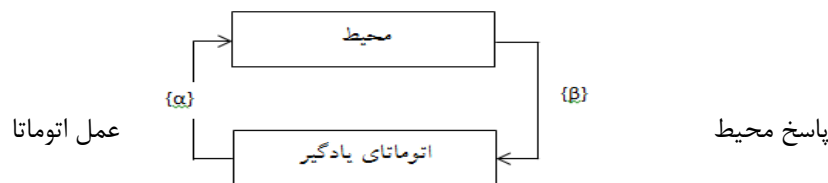
اتوماتای یادگیر عاملی است که برای قرار گرفتن در محیطی احتمالی و غیرقطعی طراحی شده است. این ماشین تعدادی عمل متناهی می‌تواند انجام دهد. هر اتوماتای یادگیر دارای برداری از احتمال‌ها می‌باشد. این بردار نشان می‌دهد هر عمل با چه احتمالی انجام می‌شود. جمع درایه‌های این بردار برابر می‌باشد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می‌شود و نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به اتوماتا داده می‌شود و اتوماتا از این پاسخ در انتخاب عمل بعدی تاثیر می‌گیرد. هدف این است که اتوماتا بتواند بهترین عمل را از بین اعمال خود انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداکثر برساند. کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل 1 مشاهده می‌شود که محیط را می‌توان توسط سه تایی  $E$  نشان داد که در آن  $\alpha$  مجموعه ورودی‌ها،  $\beta$  مجموعه خروجی‌ها و  $c$  مجموعه احتمال‌های جریمه می‌باشد.

$$E = \{ \alpha, \beta, c \}$$

$$\alpha = \{ \alpha_1, \alpha_2, \dots, \alpha_r \}$$

$$\beta = \{ \beta_1, \beta_2, \dots, \beta_r \}$$

$$c = \{ c_1, c_2, \dots, c_r \}$$



شکل 1: ارتباط بین اتوماتای یادگیر و محیط

هرگاه  $\beta$  مجموعه دو عضوی باشد، محیط از نوع  $P$  می‌باشد. در چنین محیطی  $\beta_1 = 1$  به عنوان جریمه و  $\beta_2 = 0$  به عنوان پاداش در نظر گرفته می‌شود. در محیط از نوع  $Q, \beta(n)$  می‌تواند به‌طور گسسته یک مقدار از مقادیر محدود در فاصله  $[0, 1]$  و در محیط از

نوع S،  $\beta(n)$  متغیر تصادفی در فاصله [0,1] است.  $c_1$  احتمال این که عمل  $\alpha_1$  نتیجه نامطلوب داشته باشد می باشد. در محیط ایستا مقادیر  $c_1$  بدون تغییر می ماند، حال آن که در محیط غیر ایستا این مقادیر در طی زمان تغییر می کنند. هر اتوماتا را می توان یک ماشین حالت متناهی در نظر گرفت که به وسیله پنج تایی زیر قابل نمایش است:

$$SA = \{ \alpha, \beta, F, G, \phi \}$$

پارامترهای موجود در این پنج تایی عبارتند از:

- مجموعه عمل های اتوماتا =  $(\alpha_1, \alpha_2, \dots, \alpha_r)$
  - مجموعه ورودی های اتوماتا =  $\beta = \{ \beta_1, \beta_2, \dots, \beta_r \}$
  - تابع یکه ورودی و حالت جاری را به حالت بعدی نگاشت می کند =  $F = \phi \times \beta \rightarrow \phi$
  - تابع خروجی که حالت جاری را به خروجی بعدی نگاشت می کند =  $G = \phi \rightarrow \alpha$
  - مجموعه حالت های داخلی اتوماتا در لحظه n ام =  $\phi(n) = \{ \phi_1, \phi_2, \dots, \phi_r \}$
- مجموعه  $\alpha$  شامل اعمال اتوماتا می باشد که اتوماتا در هر بار تکرار، یکی از آن ها را انتخاب می کند. مجموعه ورودی  $\beta$ ، ورودی اتوماتا را تعریف می کند.
- الگوریتم زیر یک نمونه از الگوریتم های یادگیری خطی است. فرض کنید عمل  $\alpha_i$  در مرحله n ام انتخاب شود.

- پاسخ مطلوب  
رابطه (1)

$$p_i(n+1) = p_i(n) + \sum_{j=1}^r f_j [p_j(n)]$$

$$p_j(n+1) = p_j(n) - f_j [p_j(n)]; \forall j; j \neq i$$

- پاسخ نامطلوب  
رابطه (2)

$$p_j(n+1) = p_j(n) - g_j [p_j(n)]; \forall j; j \neq i$$

$$p_i(n+1) = p_i(n) + \sum_{j=r}^r g_j [p_j(n)]$$

در روابط (1) و (2) توابع  $f_i$  و  $g_j$  توابع غیرمنفی می باشند که به ترتیب توابع پاداش و جریمه نامیده می شود. در این نوع از الگوریتم ها، توابع پاداش و جریمه به صورت زیر تعریف می شوند [13]:

رابطه (3)

$$f_j [p_j(n)] = ap_j(n) \quad 0 < a < 1$$

رابطه (4)

$$g_j [p_j(n)] = \frac{1}{r-1} - bp_j(n) \quad 0 < b \leq 1$$

در روابط فوق داریم، r تعداد اعمال یک اتوماتا، a پارامتر پاداش و b پارامتر جریمه.

با توجه به مقادیر  $\alpha$  و b سه حالت زیر را می توان در نظر گرفت. زمانی که  $\alpha$  و b با هم برابر باشند، الگوریتم را  $LRP^4$  می نامیم، زمانی که b از  $\alpha$  خیلی کوچک تر باشد، الگوریتم را  $LRE^5$  می نامیم. و زمانی که b مساوی صفر باشد الگوریتم را  $LRI^6$  می نامیم. این به روز شدن احتمالات به شکلی است که جمع احتمالات برابر با یک می ماند.

<sup>4</sup> Linear Reward Penalty Scheme

<sup>5</sup> Linear Reward Penalty Epsilon Penalty Scheme

<sup>6</sup> Linear Reward Inaction Scheme

نحوه کار یک اتوماتای یادگیر بدین صورت است که یک عمل از مجموعه عمل‌های خود را انتخاب کرده و آن را به محیط اعمال می‌کند. سپس عمل مذکور توسط یک محیط تصادفی ارزیابی می‌شود و اتوماتا از پاسخ محیط برای انتخاب عمل بعدی خود استفاده می‌کند. در طی این فرآیند، اتوماتا یاد می‌گیرد که چگونه عمل بهینه را انتخاب نماید. نحوه استفاده از پاسخ محیط به عمل انتخابی اتوماتا که در جهت انتخاب عمل بعدی اتوماتا استفاده می‌شود، توسط الگوریتم یادگیری اتوماتا مشخص می‌گردد. در بخش بعد جزئیات قسمت‌های یک اتوماتای با ساختار متغیر<sup>7</sup> معرفی می‌شود.

الگوریتم‌های یادگیری همواره سعی دارند با شناسایی مشخصات محیطی که با آن در ارتباط هستند، عملکرد و خصوصیات خود را در جهت رسیدن به یک هدف خاص بهینه نمایند. اتوماتای یادگیر نیز نوعی از این الگوریتم‌ها است که سعی دارد بر حسب پاسخی که از محیط دریافت می‌کند احتمال مربوط به حالات خود را تغییر دهد و در هر حالت نیز عملکرد خاص خود را اجرا نماید. اتوماتاهای یادگیر بر حسب ساختارشان به دو دسته عمده تقسیم می‌شوند. در دسته اول، ساختار اتوماتا ثابت می‌باشد و بر حسب یک توالی خاص از پیش تعریف شده، انتقال حالات صورت می‌گیرد. این نوع از اتوماتاها را اتوماتای ساختار ثابت می‌نامند. در نوعی دیگر از اتوماتاها، هنگامی که اتوماتا در حال فعالیت است، احتمالات مربوط به هر یک از حالات دائماً در حال تغییر می‌باشد؛ لذا این نوع از اتوماتا را اتوماتای ساختار متغیر می‌نامند.

زمانبندی کار جزء مسائل پیچیده به حساب می‌آید و الگوریتم‌های قطعی کارایی لازم را برای حل این مسئله نخواهند داشت.

الگوریتم غیرقطعی به الگوریتمی گفته می‌شود که رفتار غیرقطعی دارد. برای نمونه در یکی از مراحل چنین الگوریتمی مقدار صفر یا یک به صورت غیرقطعی (تصادفی) به یک متغیر نسبت داده می‌شود، رفتار الگوریتم از این نقطه به بعد دو مسیر متفاوت خواهد داشت. این الگوریتم‌ها در مرحله تصمیم‌گیری خود یک حالت را از میان تعدادی متناهی از حالت‌های ممکن برمی‌گزینند. اگر دست کم یکی از حالت‌های ممکن بتواند خروجی مورد انتظار را برای یک ورودی مشخص به بار آورد، می‌گویند الگوریتم آن خروجی را می‌پذیرد و در غیر این صورت خروجی را رد می‌کند (یعنی خروجی مورد انتظار را در هیچ مسیری تولید نمی‌کند یا این‌که الگوریتم به صورت نامتناهی ادامه پیدا می‌کند).

تفاوت الگوریتم‌های غیرقطعی با الگوریتم‌های قطعی در وجود عناصری از نوع حدس‌زدن در برخی مراحل آن‌ها است. در حالی که الگوریتم‌های قطعی برای هر ورودی مشخص یک خروجی مشخص تولید می‌کنند، الگوریتم‌های غیرقطعی با توجه به مقادیری که در گام‌های غیرقطعی خود محاسبه می‌کنند ممکن است خروجی‌های متفاوتی داشته باشند.

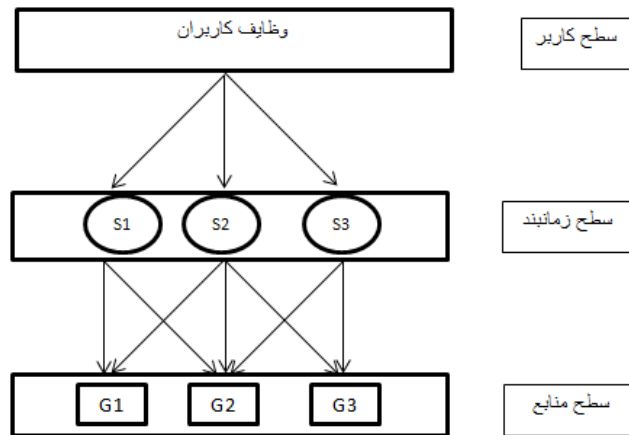
#### 4- الگوریتم زمانبندی پیشنهادی

در این مقاله روشی مبتنی بر اتوماتای یادگیر برای زمانبندی کارهای محیط تور پیشنهاد شده است. در روش پیشنهادی منبع تور به طور کاملاً تصادفی انتخاب می‌شود و این گره تور از زمانبند درخواست کار می‌کند. در صورت وجود کار در زمانبند، زمانبند انتخاب شده پاداش می‌گیرد و کار را به گره تور تحویل می‌دهد. در غیر این صورت زمانبند جریمه می‌شود و احتمال انتخاب آن در مراحل بعدی کاهش می‌یابد. بر این اساس زمانبندی‌هایی که کاری برای ارسال دارند فضای محیط را کمتر اشغال می‌کنند. انتخاب گره تور به صورت شبه تصادفی می‌باشد. در واقع اتوماتای یادگیر در یک محیط تصادفی جهت رسیدن به هدفشان که همان بروزرسانی می‌باشد به طور پیش فرض از دو تابع پاداش و جریمه استفاده می‌کنند تا زمانی که سیستم به حالت پایدار برسد. الگوریتم پیشنهادی به صورت چند پرده‌ای می‌باشد و با دیگر الگوریتم‌ها بر اساس زمان اتمام کار، زمان اجرای کارها و تعادل بارگذاری مقایسه می‌شود و عملکرد بسیار مناسب‌تری نسبت به دیگر روش‌ها مخصوصاً در محیط تور مقیاس بزرگ از خود نشان می‌دهد. اصولاً کاربران مختلف بارکاری متفاوتی داشته و بنابراین نیاز به ظرفیت محاسباتی متفاوتی دارند. تخصیص ظرفیت مشابه منابع تور به کاربران مختلف نامطلوب می‌باشد. علاوه بر این منابع تور، ناهمگون، مستقل و پویا می‌باشند. بنابراین خصوصیات منابع موقتی هستند و با زمان تغییر می‌کنند. عملکرد موفق و ناموفق تور به روش زمانبندی بستگی دارد.

در ابتدا فرض می‌شود که محیط تور شامل  $m$  زمانبند  $s = \{s_1, s_2, \dots, s_m\}$  و  $n$  گره ناهمگون  $G = \{G_1, G_2, \dots, G_n\}$  می‌باشد. هر زمانبند  $s_i$  با مجموعه‌ای از یک یا چند گره  $G_i \subseteq G$  مرتبط است. از طرف دیگر گره  $G_j$  با یک یا بیش از یک زمانبند  $S_j \subseteq s$  مرتبط

<sup>7</sup> Variable Structure Learning Automata

است. اعضای هر زمانبند مجموعه  $S_j$  را زمانبندهای همجوار می‌نامیم. بنابراین ارتباط چند به چند بین گره‌های تور و زمانبندها وجود دارد. در اینجا برای هر گره تور  $G_j$  زمانبند  $s_i$  به اتوماتای یادگیر  $A_i^j$  مجهز می‌شود. در شکل 2 نمونه‌ای از ارتباط چند به چند بین گره‌ها و زمانبندها در سیستم تور نشان داده شده است.



شکل 2: ارتباط چند به چند بین گره تور و زمانبندها در سیستم تور

اگر  $A_i^* = \{ A_i^j \mid \forall G_j \in G_i \}$  مجموعه‌ای از اتوماتای یادگیر در زمانبند  $s_i$  و  $A_i^j = \{ A_i^j \mid \forall s_i \in S_j \}$  مجموعه‌ای از اتوماتای یادگیر در گره  $G_j$  باشد. مقدار کاردینال  $G_i$  نشان دهنده تعداد اتوماتای یادگیر که زمانبند  $s_i$  با آن مجهز شده است یا تعداد گره متصل شده به زمانبند  $s_i$  می‌باشد.  $|s_i|$  نشان دهنده تعداد زمانبندهای متناظر با گره  $G_j$  می‌باشد. اگر  $\alpha_i^j = \{ \alpha_i^j(k) \mid \forall s_j \in S_j \}$  مجموعه اتوماتای یادگیر  $A_i^j \in A_i^*$  باشد، بنابراین مجموعه اقدام‌های  $\alpha_i^j$  برابر با  $|S_j|$  می‌باشد. تمام اقدام‌های اتوماتای یادگیر  $A_i^j \in A_i^*$  احتمال انتخاب اولیه مشابه و برابر با  $\frac{1}{|S_j|}$  دارند. انتخاب اقدام  $\alpha_i^j(k)$  بدین معنی است که زمانبند  $s_k$  کارش را برای  $G_j$  ارسال می‌کند. در الگوریتم پیشنهاد شده از اتوماتای یادگیر  $L_{R-P}$  استفاده شده است یعنی مقدار  $a=b$  می‌باشد. هر اتوماتای یادگیر براساس احتمال هر اقدام انتخاب می‌شود. در روش پیشنهادی از تنها اتوماتای یادگیر متعلق به مجموعه  $A_i^*$  استفاده شد.

#### 1-4- شبه کد الگوریتم پیشنهاد شده

در اینجا گره به صورت تصادفی انتخاب می‌شود و هر گره خود تقاضای کار می‌کند. در این الگوریتم وقتی یک گره انتخاب می‌شود، پیام درخواست کار به تمام زمانبندهای همجوارش (متعلق به مجموعه  $S_j$ ) ارسال می‌کند و منتظر دریافت کار جدید یا عدم وجود کار می‌شود. در ادامه با ارسال پیام درخواست کار، تمام اتوماتای یادگیر متعلق به مجموعه  $A_i^* = \{ A_i^j \mid \forall s_i \in S_j \}$  یکی از اقدام‌هایشان را با توجه به احتمالشان انتخاب می‌کنند. هر اقدام به زمانبند  $S_j$  وابسته می‌باشد. زمانبند متناظر با عمل انتخاب شده به وسیله اتوماتای یادگیر  $A_i^j \in A_i^*$ ، می‌تواند کارش را برای گره  $G_j$  ارسال می‌کند. بعد از انتخاب اقدام، هر اتوماتای یادگیر  $A_i^j \in A_i^*$  زمانبند متناظرش را بررسی می‌کند تا ببیند که آیا با اقدام انتخاب شده برابر است یا نه، اگر برابر بود زمانبند انتخاب شده ( $S_i$ ) اجازه ارسال کار برای گره  $G_j$  را می‌دهد. در غیر این صورت زمانبند متوقف می‌شود تا پیام درخور را دریافت کند. در اینجا تمام کارها به صورت تصادفی به زمانبندها تخصیص داده می‌شوند و اولویت قرارگیری آن‌ها در صف بر اساس زمان تولید آن‌ها می‌باشد. بدین صورت که ابتدا کار یا کارهایی که ابتدا زودتر تولید شده‌اند انتخاب شده و سپس بر اساس تعداد کارها به تصادف زمانبند انتخاب می‌شود و کار یا کارها نیز به زمانبندها اختصاص داده می‌شوند و در صف قرار می‌گیرند.

- ✓ محیط تور شامل  $m$  زمانبند  $S = \{s_1, s_2, \dots, s_m\}$
- ✓  $n$  گره تور ناهمگون  $G = \{G_1, G_2, \dots, G_n\}$  می باشد.
- ✓ هر زمانبند  $s_i$  با مجموعه‌ای از یک یا چند گره تور  $G_r \subseteq G$  در ارتباط است
- ✓ از طرف دیگر گره تور  $G_r$  با یک یا بیش از یک زمانبند  $s_j \subseteq S$  مرتبط است
- ✓ برای هر گره تور  $G_r$  زمانبند  $s_i$  به اتوماتای یادگیر  $A_r^j$  مجهز می شود

**Algorithm:** //the proposed scheduling algorithm

- 1:  $K \leftarrow$  Number Step and At the beginning equal to one
- 2:  $S_j \leftarrow$  Set of Schedule neighboring connected to the grid node  $G_j$
- 3:  $A_i^j \leftarrow$  Learning Automata Scheduler  $S_i$  dependent on  $G_j$  With initial value  $1/s_j$
- 4: **For all**  $K$  is loop index **do**
- 5:     **Input**  $G_j, A$  //  $G_j = "x"$  in random select
- 6:     **If** ( $S_j$  is correct){
- 7:         **For** (each element in  $S_j$  set ) **do**
- 8:             **If** ( $j=i$ )
- 9:                  $P_j(k+1)=p_j(k)+a[1-p_j(k)]$
- 10:             **Elseif**( $j \neq i$ )
- 11:                  $P_j(k+1)=(1-a)p_j(k)$
- 12:             **End if**
- 13:         **End For**
- 14:     **Else**
- 15:         **For** (each element in  $S_j$  set ) **do**
- 16:             **If**( $j=i$ )
- 17:                  $P_j(k+1)=i(1-b)p_j(k)$
- 18:             **Else if** ( $j \neq i$ )
- 19:                  $p_j(k+1) = \left(\frac{1}{r-2}\right) + (1-b)p_j(k)$
- 20:             **Endif**
- 21:         **End For**
- 22:     **End if**
- 23:      $K=k+1$
- 24:     **If** ( $S_j$  is correct **goto** line 4)
- 25:     **Output**  $p_j(k+1)$
- 26: **End**

## 5- ارزیابی الگوریتم زمانبندی پیشنهادی

برای انجام ارزیابی عملی الگوریتم زمانبندی پیشنهادی از سیستمی با مشخصات زیر استفاده شد.



مشخصات رایانه				
مدل سیستم	حافظه	پردازنده	سیستم عامل	نرم افزار
System type: 64bit Operating system	Installed memory(RAM): 4.00GB	Intel Pentium(R) cpu GHZ 2.60,620G GHZ2.60	Windows 8 Enterprise	MATLAB a(7.140)64-bit 2012 R

الگوریتم پیشنهاد شده 50 بار اجرا شد و میانگین این پارامتر محاسبه شد. اگر T نشان دهنده مجموعه n کار مستقل که به m منبع در محیط تور تخصیص داده شده باشند. در زمان t=0 همه m منبع و n کار مستقل جهت پردازش در دسترس می باشند. ماتریس زمان مورد انتظار برای محاسبات (ETC) با اندازه m\*n برای ذخیره زمان پردازش n کار در m منبع استفاده می شود. بنابراین ETC(j,i) زمان مورد انتظار برای محاسبه کار  $T_j$  در منبع  $M_i$  را نشان می دهد. زمان اتمام ( $c_j$ ) کار  $T_j$  برابر است با:

$$c_j = w_{i,j} + ETC(j,i)$$

در اینجا  $w_{i,j}$  بار قرار گرفته شده در منبع  $M_i$  قبل از پردازش کار  $T_j$  می باشد.

عملکرد روش پیشنهادی از طریق آزمایش های شبیه سازی مختلف در محیط تور با سه اندازه مختلف شامل: سیستم تور مقیاس کوچک<sup>8</sup> شامل 64 گره و 512 عنصر پردازش، سیستم تور مقیاس متوسط<sup>9</sup> شامل 128 گره و 1024 عنصر پردازش و سرانجام سیستم تور مقیاس بزرگ<sup>10</sup> شامل 512 گره و 4096 عنصر پردازش، اجرا شد. برای اجرای آزمایش های فرض شد که همه کارها تنها به منابع محاسباتی نیاز دارند. برای هر محیط تور فرض شد که هر گره عناصر پردازشی با تعداد مشابهی دارد. ظرفیت پردازشی عناصر پردازنده با استفاده از تابع توزیع احتمال گاوس با میانگین 1000 میلیون دستوالعمل در ثانیه و واریانس 150 تولید شد. پهنای باند اسمی شبکه بین هر دو گره 100 mbps فرض شد.

تعداد کاربران ثابت و برابر 50 و 100 و 200 برای هر یک از محیط های تور در نظر گرفته شد. تعداد کل کارهای ثبت شده برای کل کاربران در سه محیط تور به ترتیب 1000، 4000، 32000 می باشد. زمان اجرای کارهای ایجاد شده به وسیله هر کاربر از طریق توزیع نرمال با میانگین 500 میلیون دستوالعمل و واریانس 100 بدست آمد. سرعت تولید کار جدید هر کاربر با توزیع پواسون با میانگین {20 و 15 و 10 و 5} در نظر گرفته شد. این سرعت به صورت تصادفی از مجموعه ذکر شده در هر آزمایش انتخاب شد. برنامه 50 بار اجرا شد و میانگین نتایج محاسبه شد.

### 1-5- پارامتر مورد ارزیابی

برای مقایسه روش پیشنهادی با دیگر روش ها از سه پارامتر شامل زمان اتمام کار<sup>11</sup>، زمان اجرای کارها<sup>12</sup> و تعادل بار<sup>13</sup> استفاده شده است.

- **زمان اتمام کار** متداولترین پارامتر اندازه گیری روش بهینه سازی می باشد. این پارامتر به عنوان بیشترین زمان اجرای همه کارهای ثبت شده تعریف می شود. به عبارت دیگر، نشان دهنده زمان کامل شدن آخرین کار می باشد. کمینه سازی این پارامتر نشان می دهد که کارها در زمان طولانی انجام نمی شوند. این پارامتر بر حسب میلی ثانیه اندازه گیری می شود و بر اساس رابطه روبرو تعریف می شود:

$$c_{max} = \max \{c_j, j = 1, \dots, n\}$$

<sup>8</sup> Small scale

<sup>9</sup> Medium scale

<sup>10</sup> Large scale

<sup>11</sup> Make span

<sup>12</sup> Flow Time

<sup>13</sup> Load balancing

- **زمان اجرای کار** این پارامتر به عنوان مجموعه زمان کامل شدن همه کارها تعریف می‌شود و زمان کلی را که پردازشگرها به وسیله کارها اشغال شده‌اند نشان می‌دهد. این پارامتر کیفیت سرویس‌دهی سیستم تور را نشان می‌دهد و بر حسب میلی ثانیه اندازه‌گیری می‌شود و بر اساس  $f = \sum_{j=1}^n c_j$  تعریف می‌شود.

- **تعداد بار** نشان دهنده توزیع بار تخصیص داده شده به گره‌های تور می‌باشد. توزیع حجم کار یکنواخت نشان دهنده تعادل بار گذاری در هنگامی که تمام گره تورها حجم پردازشی مشابهی دارند، می‌باشد. به هر حال در حالتی که ظرفیت‌های پردازشی متفاوتی وجود دارد، حجم کار ثبت شده برای هر گره تور بایستی متناسب با ظرفیت آن باشد. در این مورد، انحراف معیار زمان کامل شدن (یعنی زمانی که یک گره تور اجرا یا آخرین کارش را کامل می‌کند) گره‌های تور برای تعادل بار گذاری منتظر می‌ماند. این پارامتر نشان دهنده اختلاف میانگین بین زمان تکمیل تورهای مختلف می‌باشد. دو پارامتر زمان اتمام کار و زمان اجرای کار در صورتی که حجم کارهای تورها متعادل باشد کمینه می‌شود. اگر  $M$  و  $S_d$  به ترتیب میانگین و انحراف معیار زمان تکمیل تمام گره‌ها باشد، در این صورت تعادل بارگذاری به صورت  $\frac{M - S_d}{M} \times 100$  تعریف می‌شود.

از آنجا که در نرم‌افزار متلب تمام این اطلاعات به صورت ماتریس باید ذخیره شود، بنابراین در ابتدا ماتریس زمانبند تولید می‌شود و صف کارها در هر زمانبند مشخص می‌شود. حال بر اساس این ماتریس، پردازش شروع می‌شود. یک گره تور به تصادف انتخاب می‌شود و درخواست کار جدید می‌کند. این درخواست به تمام زمانبندهای همجوارش ارسال می‌شود اگر در زمانبند انتخاب شده کاری برای ارسال وجود داشته باشد آن کار را به گره تور اختصاص می‌دهد و پاداش می‌گیرد، پاداش در اینجا در واقع به معنی افزایش احتمال انتخاب می‌باشد. اما در صورتی که کاری در زمانبند برای ارسال وجود نداشته باشد پیغام عدم ارسال کار فرستاده می‌شود و زمانبند جریمه می‌شود، جریمه به معنی کاهش احتمال انتخاب می‌باشد.

## 2-5- الگوریتم‌های مورد مقایسه

برای ارزیابی کارایی الگوریتم زمانبندی پیشنهادی، الگوریتم‌های مشابه به شرح زیر مورد ارزیابی و مقایسه قرار گرفته‌اند:

- الگوریتم GRZ [14]: این مقاله در سال 2005 توسط ژائو و همکاران ارائه شد. در این تحقیق دو مدل برای پیش بینی زمان کار در سیستم‌های تور ارائه شد. در این الگوریتم‌ها از الگوریتم ژنتیک برای کمینه کردن زمان پایان کارها، از طریق بهینه سازی تخصیص کار به هر گره تور استفاده شد.

- الگوریتم LAH [15]: این الگوریتم توسط سای هولینگ در سال 2011 چاپ شد. در این مقاله یک منطق نادقیق ترکیبی بر پایه بهینه سازی ازدحام ذرات با روش عملکرد جهش-مقاطع برای کمینه‌سازی زمان اتمام کار در مسائل زمانبندی، ارائه شده است. در روش پیشنهاد شده، ازدحام ذرات ترکیبی، سیستم استنباط نادقیق به منظور تعیین وزن اینرسی PSO و کنترل پارامتر عملکرد توسط دانسته‌های انسانی استفاده می‌شود. با استفاده از سیستم نادقیق، وزن اینرسی تطبیق داده می‌شود. عملکرد ذکر شده نیروی موثر اعمال می‌کند، تا بهینه محلی حاصل شود. از سازوکار چرخ رولت برای تبدیل مقادیر مکان ذرات استفاده شد.

- الگوریتم CCL [16]: این الگوریتم توسط شیانگ چانگ در سال 2007 براساس الگوریتم مورچه پیشنهاد شد. در محیط طبیعی، مورچگان توانایی عجیبی در حرکت گروهی به سمت غذا براساس مسیری بهینه دارند. الگوریتم مورچگان، رفتار مورچه‌ها را شبیه سازی می‌کند. در اینجا الگوریتم بهینه سازی متعادل کلونی مورچگان برای زمانبندی کارها و تخصیص به منابع تور پیشنهاد شده است.

- الگوریتم پیشنهادی (LAJS): نتایج یک بار اجرای این الگوریتم برای سناریوهای مختلف سیستم‌های تور مقیاس کوچک، متوسط و بزرگ اندازه‌گیری و ارائه شده است.

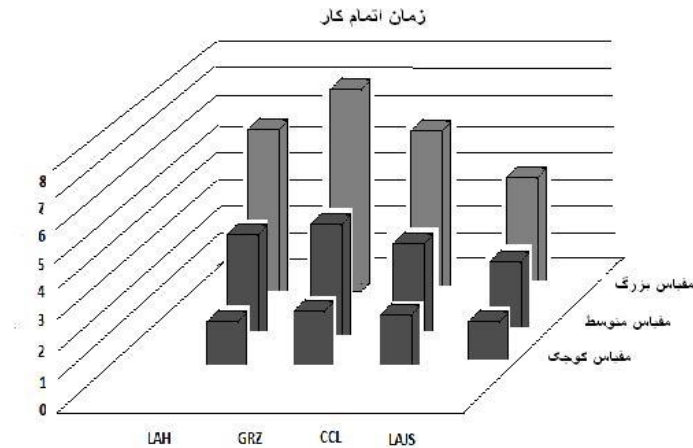
وضعیت و عملکرد الگوریتم پیشنهادی در مقایسه با هریک از این الگوریتم‌های مشابه در بخش نتایج ارزیابی ارائه و تحلیل خواهد شد.

## 3-5- نتایج ارزیابی

شکل 3 زمان اتمام کار را در سیستم تور با اندازه‌های مختلف نشان می‌دهد. در هر مقیاس از تور عملکرد چهار الگوریتم ذکر شده با یکدیگر مقایسه شده است، به طوری که مستطیل اول نشان دهنده الگوریتم LAH، سپس مستطیل دوم نشان دهنده الگوریتم GRZ

وهمینطور CCL و مستطیل آخر نشانگر الگوریتم پیشنهاد شده (LAJS) می‌باشد. زمان اتمام کار در الگوریتم LAJS به‌طور چشمگیری از سایر الگوریتم‌ها کمتر است.

با توجه به نتایج به‌دست آمده در بالا، زمان اتمام کار در الگوریتم پیشنهادی از نصف زمان اتمام کار در الگوریتم‌های دیگر نیز کمتر است و این موضوع در محیط تور با مقیاس بزرگ به‌خوبی نشان داده شده است.

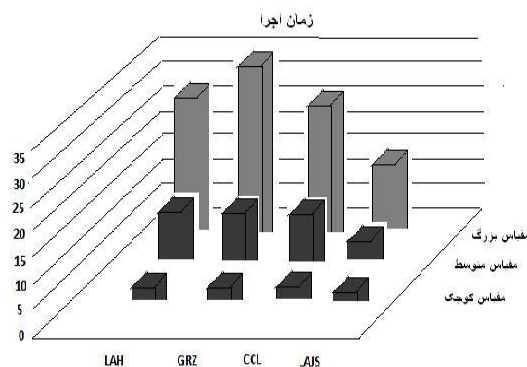


شکل 3: میانگین زمان اتمام کار در محیط‌های تور با اندازه‌های مختلف

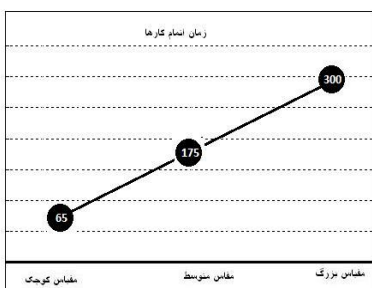
شکل 3 الگوریتم پیشنهاد شده را با دیگر الگوریتم‌ها مقایسه می‌کند و نشان می‌دهد که زمان اجرای اتمام کار نیز در الگوریتم LAJS به شدت از دیگر الگوریتم‌ها کمتر می‌باشد. الگوریتم‌های LAH و GRZ و CCL در محیط مقیاس کوچک و متوسط تقریباً شبیه به هم عمل می‌کنند یعنی نتایج خروجی آن‌ها با هم نسبتاً برابر است اما با افزایش مقیاس محیط، نتایج آن‌ها کاملاً با یکدیگر متفاوت می‌باشد، به‌طوری که الگوریتم CCL به نسبت بهتر از دو الگوریتم دیگر عمل می‌کند.

این موضوع نشان می‌دهد که الگوریتم‌های LAH و GRZ به مقیاس محیط تور حساس می‌باشند و با افزایش حجم کارها کارایی آن‌ها کاهش می‌یابد.

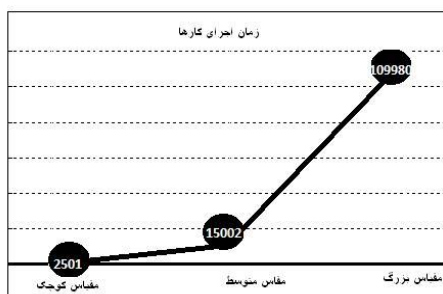
در شکل‌های 5 و 6 نمودار تغییرات زمان اتمام کار و زمان اجرای کارها نشان داده شده است. تغییرات زمان اتمام کار در شکل 5 تقریباً خطی می‌باشد که نشان می‌دهد با افزایش مقیاس محیط تور زمان اتمام کار تقریباً خطی تغییر پیدا می‌کند. شکل 6 نشان می‌دهد که مقدار زمان اجرای کارها با افزایش مقیاس محیط تور به‌طور نمایی افزایش پیدا می‌کند، که این موضوع نشان‌دهنده واگرایی سیستم می‌باشد. اما این موضوع در الگوریتم پیشنهاد شده به‌طور چشم‌گیری از الگوریتم‌های دیگر کمتر می‌باشد که در شکل 4 نشان داده شده است.



شکل 4: زمان اجرای تمام کارها در سیستم‌های تور

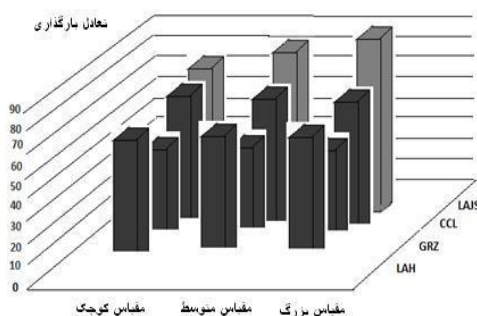


شکل 5: زمان اتمام کار در محیط‌های تور



شکل 6: زمان اجرای کارها در محیط‌های تور

نکته قابل توجه در اینجا، افزایش تعداد بارگذاری با افزایش مقیاس سیستم تور می‌باشد، در حالی که در دیگر الگوریتم‌ها با افزایش مقیاس تور این مقدار کاهش می‌یابد. این موضوع نشان‌دهنده سازگاری روش پیشنهادی با مقیاس‌های بزرگ‌تر می‌باشد. از طرف دیگر نیز انحراف معیار کاهش می‌یابد که نشان‌دهنده افزایش توزیع یکنواخت تکرار آزمایش‌ها می‌باشد که در شکل 7 نمایش داده شده است.



شکل 7: تعادل بارگذاری در سیستم‌های تور

## 6- نتیجه گیری

محاسبات توری به‌عنوان یک رویکرد جدید برای حل مسائل در مقیاس بزرگ در زمینه‌های علمی، مهندسی و تجاری به‌وجود آمده‌اند. آن‌ها این قابلیت را به‌وجود آوردند که بتوان بنگاه‌های مجازی را به منظور اشتراک و یکپارچه‌سازی میلیون‌ها منبع از نظر جغرافیایی در سطح سازمان‌ها و حوزه‌های مدیریتی پراکنده‌اند، ایجاد کرد. ایده اصلی تور این است که منابع ماشین‌های شرکت‌کننده، از طریق یک لایه نرم‌افزاری به صورت شفاف و مطمئن مورد استفاده قرار بگیرد. محاسبات توری، محاسبات توزیع شده‌ای است که به سطح تکامل بالاتری رسیده است. اهداف مورد نظر در طراحی تور همچنین کاربردهایی که برای آن در نظر گرفته شده است، از عوامل تعیین‌کننده در نوع کاربری تور می‌باشد. در این مقاله الگوریتمی برای زمانبندی کارهای محیط تور پیشنهاد شده است. الگوریتم پیشنهادی با دیگر الگوریتم‌ها بر اساس پارامتر زمان اتمام کار، زمان اجرای کارها و تعادل بارگذاری مقایسه شد و عملکرد بسیار مناسب‌تری نسبت به دیگر روش‌ها مخصوصاً در محیط تور مقیاس بزرگ از خود نشان داد. نکته قابل توجه در نتایج تعادل بارگذاری

می‌باشد. همان‌طور که در نمونه‌های ارائه‌شده مشخص است، با افزایش مقیاس محیط تور تعادل بارگذاری در محیط‌های مختلف کاهش یافته است، در صورتی که در الگوریتم پیشنهادی علی‌رغم این که میزان تعادل بارگذاری از دیگر الگوریتم‌ها بهتر است، با افزایش مقیاس محیط تور، تعادل بارگذاری نیز افزایش پیدا کرده است و انحراف معیار آن که نشان‌دهنده پراکندگی داده‌ها است کاهش پیدا کرده است.

در ادامه کارهای انجام‌گرفته در این مقاله، پیشنهادهایی به شرح زیر برای کارهای آتی ارائه می‌شود:

- انتخاب منبع تور بر اساس حجم کار و میزان بار موجود در تور صورت گیرد و تصادفی نباشد.
- از روش‌های مختلف به صورت ترکیبی استفاده شود، به عنوان مثال زمانبندی با الگوریتم ژنتیک انجام شود، تخصیص با اتوماتای یادگیر و انتخاب گره تور با تجمع ذرات.
- الگوریتم جهش قورباغه یکی از الگوریتم‌های پرکاربرد می‌باشد از آن نیز می‌توان در زمانبندی، تخصیص کار و کشف منبع استفاده کرد.

## منابع

- [1] Buyya, Rajkumar. Economic-based Distributed Resource Management and Scheduling for Grid Computing. Diss. Monash University, Melbourne, Australia, 2012.
- [2] Coulouris, George F., Jean Dollimore, and Tim Kindberg. Distributed systems: concepts and design. pearson education, 2011.
- [3] Matthias Jarke, Mantred A. Jeusteld, Christoph Quix, and Panos Vassiliadis, Architecture and quality in datawarehouses: An extended approach, information Systems, 2010.
- [4] Rantzaou, Ralf, and Holger Schwarz. "A Multi-Tier Architecture for High-Performance Data Mining." ipvr, 2008
- [5] El-Rewini, Hesham, Theodore G. Lewis, and Hesham H. Ali. "Task scheduling in parallel and distributed systems." (2006).
- [6] Buyya, Rajkumar, David Abramson, and Srikumar Venugopal. "The grid economy." Proceedings of the IEEE 93.3 (2005): 698-714.
- [7] Foster, Ian, and Carl Kesselman. "The Grid 2: Blueprint for a future computing infrastructure." Waltham: Morgan Kaufmann Publishers (2004).
- [8] Zhu, Yanmin. "A survey on grid scheduling systems." Department of Computer Science, Hong Kong University of science and Technology (2003): 32.
- [9] Berman, Fran, Geoffrey C. Fox, and Anthony JG Hey. "The Grid: past, present, future." (2003): 9-50.
- [10] Berstis, Viktors, et al. Introduction to grid computing with globus. IBM Corporation, International Technical Support Organization, 2003.
- [11] Foster, Ian, et al. "Grid services for distributed system integration." Computer 35.6 (2012): 37-46.
- [12] Sun, Xian-He, and Ming Wu. "Grid Harvest Service: a system for long-term, application-level task scheduling." Parallel and Distributed Processing Symposium, 2003. Proceedings. International. IEEE, 2013.
- [13] K. S. Narendra, M. A. L. Thathachar, Learning automata: An introduction, Prentice Hall, [1989].
- [14] Gao, Yang, Hongqiang Rong, and Joshua Zhexue Huang. "Adaptive grid job scheduling with genetic algorithms." Future Generation Computer Systems 21.1 (2005): 151-161.
- [15] LING, SAI HO, et al. "Hybrid fuzzy logic-based particle swarm optimization for flow shop scheduling problem." International Journal of Computational Intelligence and Applications 10.03 (2011): 335-356.
- [16] Chang, Ruay-Shiung, Jih-Sheng Chang, and Po-Sheng Lin. "An ant algorithm for balanced job scheduling in grids." Future Generation Computer Systems 25.1 (2009): 20-27.