

ICBR-LA: الگوریتمی ترکیبی برای تخصیص وظایف در سیستمهای محاسباتی ناهمگن

محمد رضا میبیدی
دانشگاه صنعتی امیرکبیر
mmeybodi@aut.ac.ir

بهروز معصومی
دانشگاه آزاد اسلامی قزوین
bmasoumi@Qazviniau.ac.ir

سعید صبا منیری
دانشگاه آزاد اسلامی واحد صوفیان
Saba_Moniry@Hotmail.com

چکیده: در این مقاله، روشی بهبود یافته بر پایه مدل ارائه شده $CBR-LA$ برای انتساب ایستای وظایف در سیستمهای محاسباتی ناهمگن، ارائه گردیده است که در آن، ترکیبی از تکنیکهای استنتاج مبتنی بر مورد (CBR) و مدل اتوماتای یادگیر (LA) استفاده شده است. در این مدل، اتوماتای یادگیر بعنوان مکانیزم تطبیق بکار گرفته می شود که موردهای تجربه شده گذشته را به مساله ای که باید حل شود، تطبیق می دهد. هدف این مقاله ارائه تغییراتی جهت کاهش مقادیر معیار کارایی مورد نظر ($Makespan$) برای یافتن یک راه حل تقریباً بهینه می باشد. در نهایت با استفاده از شبیه سازی کامپیوتری و مقایسه ی نتایج بدست آمده از اجرای این الگوریتم با الگوریتم قبلی نشان داده شده که الگوریتم ارائه شده کارایی بهتری را در زمینه ی تولید پاسخها دارا است.

واژه های کلیدی: استنتاج مبتنی بر مورد، اتوماتای یادگیر، زمانبندی وظایف، سیستمهای محاسباتی ناهمگن.

۱- مقدمه

$min[14]$ ، $GA[15]$ ، $A^*[4]$ ، و اتوماتای یادگیر $[6]$ از جمله این روش ها هستند. در تمام این مدلها تخصیص وظیفه، بدون استفاده از تجربیات گذشته، راه حل را می سازند. در $[7]$ یک مدل ترکیبی از تکنیکهای اتوماتای یادگیر و استنتاج مبتنی بر مورد به نام $CBR-LA$ ارائه شده است که هدف عمده و ویژگی کلیدی آن کاهش تعداد تکرار و دستیابی نسبتاً سریع به پاسخ بر اساس تجارب گذشته می باشد. مدل $CBR-LA$ برای تخصیص وظایف بر روی مجموعه ای از پردازنده ها در یک محیط محاسباتی ناهمگن استفاده شده است.

هدف اصلی این مقاله ارائه الگوریتمی بهبود یافته برای تخصیص ایستای وظایف با توجه به روش ارائه شده در $[7]$ است. این روش با استفاده از ترکیب روش یادگیری اتوماتای یادگیر و استنتاج مبتنی بر مورد عمل می کند که در روش پیشنهادی با انجام تغییراتی در مدل LA و مدل پاسخ محیط، عملکرد الگوریتم بهبود داده شده است. جهت نمایش میزان این بهبود و نحوه استفاده صحیح الگوریتم پیشنهادی از تجارب خود، آزمایشهایی انجام گرفته اند که نشان می دهند الگوریتم جدید ($ICBR-LA$)، پاسخهای تولیدی الگوریتم قبلی را تا حدود ۴۰ درصد بهبود می دهد. در ادامه مقاله، در بخش ۲

محیطهای محاسباتی ناهمگن (HC) ماشین-مربک، دنباله ای توزیع شده از ماشینهای با کارایی بالا را بکار می گیرند تا بتوانند برنامه های کاربردی را که از نظر محاسباتی نیازمندیهای متفاوتی دارند اجرا نمایند. عمل انتساب مجموعه ای از وظایف (متاوظیفه) به ماشینها و زمانبندی ترتیب اجرای آنها را نگاشت می نامند. مساله ی عمومی نگاشت بهینه وظایف به ماشینها در یک دنباله HC بعنوان یک مساله $NP-complete$ شناخته شده است $[1]$. متاوظیفه بعنوان مجموعه ای از وظایف مستقل تعریف می شود که هیچ وابستگی داده ای بین وظایف وجود ندارد $[2]$. مساله نگاشت می تواند بطور ایستا یا پویا انجام شود. در نگاشت ایستا فرض می شود که هر ماشین در هر لحظه یک وظیفه واحد را به ترتیبی که وظایف تخصیص داده می شوند، اجرا می کند و هدف از آن، مینیمم کردن زمان کل اجرای متاوظیفه است. اندازه ی متاوظیفه، و تعداد ماشینهای موجود در دنباله HC ، ایستادگی از قبل شناخته شده است. روشهای مختلفی برای نگاشت ایستای وظایف به کار رفته اند، الگوریتمهایی که بر پایه تکنیکهای تئوری گراف عمل می کنند، $simulated$ - Max ، Min ، $Min-min$ ، MCT ، MET ، $annealing[3]$.

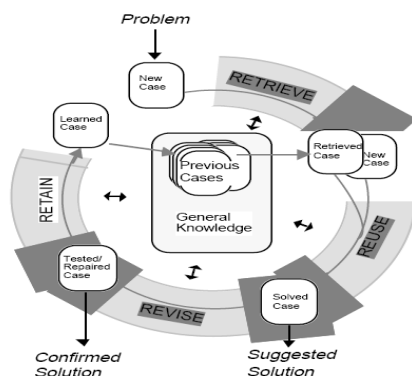
مقدمه‌ای در مورد CBR ارائه شده است. در بخش ۳ به معرفی مدل‌های HC، LA و CBR پیشنهادی پرداخته شده است. در بخش ۴ اتوماتای یادگیر و مدل پیشنهادی ارائه گردیده و در بخش ۵ محیط شبیه‌سازی و نتایج آزمایشهای انجام شده در جهت نشان دادن کاهش معیار کارایی Makespan، ارائه می‌گردد.

۲- استنتاج مبتنی بر مورد و کاربردهای آن

تکنیک استنتاج مبتنی بر مورد (CBR)^۱ یک روش حل مساله مبتنی بر دانش است که بر پایه استفاده مجدد از تجربیات پیشین عمل می‌کند و از تحقیقات علوم شناختی^۲ پدیدار شده است [۹ و ۸]. در این روش فرض بر این است مسائل مشابه می‌توانند راه‌حلهای مشابهی داشته باشند. بنابراین، ممکن است مسائل جدید با روشهای تجربه شده در مسائل قبلی قابل حل باشند. برخلاف تکنیکهای سنتی مبتنی بر دانش، CBR بر روی تجربه حل مساله خاصی تمرکز می‌کند که برگرفته از موارد جمع‌آوری شده در پایگاه مورد^۳ است. این موارد، تجربه خاصی را در یک دامنه حل مساله نشان می‌دهند. فعالیتهای اصلی در حین حل مسائل با این روش، در چرخه CBR، تشریح می‌شود که در شکل ۱ نشان داده شده است [۹ و ۲۰ و ۸]. باید توجه داشت که CBR یک راه-حل قطعی را پیشنهاد نمی‌کند بلکه فرضیات و نظراتی را برای عبور از فضای راه‌حل، ارائه می‌کند. برای مطالعه بیشتر می‌توان به منابع [۲۰ و ۸] مراجعه نمود.

تکنیک CBR در کاربردهایی مفید است که درک کاملی از فضای پاسخ مساله وجود نداشته باشد و نیز امکان تکراری بودن مسائل و مشابه بودن آنها وجود داشته باشد، و از طرفی نیز از لحاظ برخی معیارها همانند زمان، حل یک مساله اهمیت داشته باشد [۱۰ و ۸]. در سالهای اخیر، CBR قابلیت‌های بالایی را در حوزه‌های مختلفی نظیر تصمیم‌گیری، پیشگویی، تشخیص عیب، برنامه‌ریزی، کنترل کیفیت/فرایند، پشتیبان تصمیم، محاسبات توزیع-شده [۱۱] و بازیابی اطلاعات نشان داده است. با توجه به ویژگیهای مذکور مساله‌ای تخصیص وظایف می‌تواند بعنوان یکی از کاربردهایی مطرح باشد که امکان سود جستن از CBR برای آن امکانپذیر می‌باشد. در [۷] آقای قنبری و همکارانش این تکنیک را با ترکیب LA برای

حل مسائل در این کاربرد، مورد استفاده قرار داده و پس از انجام آزمایشهای مختلف و مقایسه با روش LA چنین نتیجه‌گیری کرده‌اند که بکار بردن CBR و استفاده از تجارب قبلی، در بدترین حالت تا حدود ۱۰ درصد تعداد تکرارهای مورد نیاز الگوریتم را کاهش داده و از لحاظ زمانی مقرون به صرفه است. در ادامه مدل سیستم HC را معرفی کرده و سپس به معرفی مدل اتوماتای یادگیر پیشنهادی و نیز مدل استنتاج مبتنی بر مورد بکار رفته در الگوریتم CBR-LA پرداخته می‌شود.



شکل ۱: چرخه استنتاج مبتنی بر مورد [۸ و ۲۰]

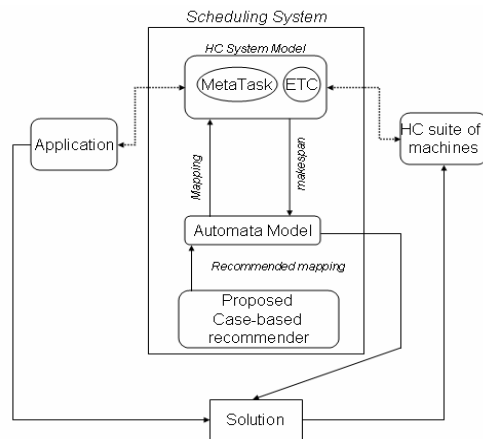
۳- مدل سیستم HC

این بخش، یک مدل کلی از چارچوب ارائه شده برای تخصیص وظیفه در سیستم HC را ارائه می‌دهد. شکل ۲ نمایشی کلی از این چارچوب را ترسیم می‌کند. سیستم زمانبندی و اجزای مشخص شده آن در شکل ۲ برای تخصیص وظایف به ماشینهای مختلف بکار می‌روند [۷]. در این مدل فرض شده است که وظایف متاوظیفه مستقل بوده و به تبادل داده و برقراری ارتباط نیازی ندارند. درضمن، سیستم HC شامل مجموعه‌ای از ماشینهای ناهمگن است که بوسیله یک زیرساخت شبکه با هم ارتباط دارند. در دنباله HC، زمان اجرای مورد انتظار وظایف مربوط به مساله جاری، در هر ماشین مشخص است.

۳-۱- ویژگیهای مسائل نگاشت ورودی

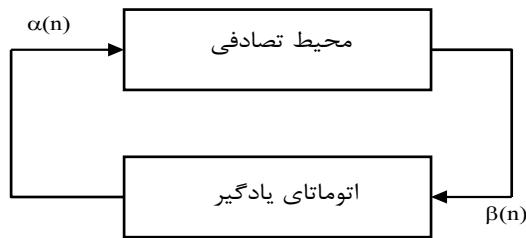
مساله تخصیص وظیفه‌رانی توان بصورت نگاشت $\pi: S \rightarrow M$ از مجموعه S (وظایف) به مجموعه M (ماشینها) در نظر گرفت، بطوریکه شاخص هزینه‌ی c بهبود یابد. c

نشاندهنده شاخص هزینه می‌باشد و منظور از آن، مدت زمانی است که برای تکمیل اجرای یک متاوظیفه مورد نیاز است و این شاخص Makespan نامیده می‌شود. در ضمن، در هر تکرار الگوریتم، یک بردار به طول تعداد وظایف، نمونه‌ای از نگاشت π را نشان می‌دهد. هدف سیستم زمانبندی، یافتن نگاشتی از وظایف موجود در متاوظیفه به پردازنده‌های موجود در HC است، بطوریکه شاخص هزینه تعریف شده بهینه شود.



شکل ۲: مدل چارچوب پیشنهادی [۷]

کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل ۳ مشاهده می‌شود. محیط را می‌توان توسط سه-تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن مجموعه ورودی‌ها، β مجموعه خروجی‌ها و c مجموعه احتمال-های جریمه می‌باشد. هرگاه β مجموعه‌ای دو عضوی $\{0, 1\}$ باشد، محیط از نوع P است. در محیط از نوع Q ، $\beta(n)$ می‌تواند به طور گسسته مقداری از مقادیر محدود در فاصله $[0, 1]$ را اختیار کند و در محیط از نوع S ، $\beta(n)$ متغیر تصادفی در فاصله $[0, 1]$ است. اتوماتاهای یادگیر به دو دسته با ساختار ثابت و با ساختار متغیر (VSLA)^۵ دسته بندی می‌شوند.



شکل ۳: ارتباط بین اتوماتای یادگیر و محیط

۴-۱- اتوماتای یادگیر با ساختار متغیر

اتوماتای یادگیر با ساختار متغیر، با ۴ تایی $\{\alpha, \beta, p, T\}$ نشان داده می‌شود که در آن مجموعه اقدامهای اتوماتا، β مجموعه ورودی‌های اتوماتا، p بردار احتمال مربوط به اقدامهای اتوماتا و T الگوریتم یادگیری می‌باشد.

در این نوع اتوماتا، اگر اقدام α_i در مرحله n -ام انتخاب شود و پاسخ مطلوبی از محیط دریافت شود، احتمال $p_i(n)$ افزایش یافته و احتمال مربوط به سایر اقدامها کاهش می‌یابد و به ازای دریافت پاسخ نامطلوب، احتمال $p_i(n)$ کاهش یافته و احتمال مربوط به سایر اقدامها افزایش می‌یابد. در هر حال، تغییرات به گونه‌ای صورت می‌گیرند که حاصل جمع مقادیر $p_i(n)$ ها همواره ثابت و مساوی یک باقی بماند. الگوریتم یادگیری که در این مقاله استفاده شده است، هنگام دریافت پاسخ مطلوب، تغییر احتمال اقدامها با توجه به روابط زیر انجام می‌شود.

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \quad (1)$$

$$j \neq i \quad \forall j \quad p_j(n+1) = (1-a)p_j(n) \quad (2)$$

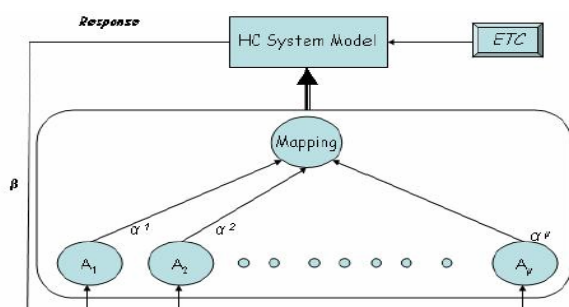
و نیز هنگام دریافت پاسخ نامطلوب، تغییرات احتمال اقدامها، با توجه به روابط زیر انجام می‌شود.

مسائل جدید ورودی به سیستم زمانبند، شامل متاوظیفه و زمانهای تخمینی اجرای آنها بر روی هر یک از ماشینهای محیط HC می‌باشند. این زمانهای تخمینی ممکن است با استفاده از تکنیکهای محک‌زنی بدست آمده باشند [۱۶]. تخمین زمان اجرای مورد انتظار روی هر ماشین به ازای هر وظیفه، قبل از اجرا مشخص بوده و در ماتریسی به ابعاد $T \times M$ بنام ETC^۶ نگهداری می‌شود. ممکن است چنین فرض شود که مدخل $ETC(t_i, m_j)$ ، زمان مورد نیاز جهت انتقال فایل‌های قابل اجرا و داده مربوط به وظیفه t_i از ماشین مبدا m_j را در بر دارد [۷].

۴-۲ اتوماتای یادگیر و مدل پیشنهادی

اتوماتای یادگیر یکی از مدل‌های یادگیری تقویتی است که در آن یک اتوماتا عملی بهینه را با توجه به اقدام انجام شده و بازخورد محیط فرا می‌گیرد. هدف نهایی اتوماتا، یادگیری انتخاب بهترین عمل از بین اعمال خود می‌باشد. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداکثر برساند.

انجام این تغییر در الگوریتم یادگیری، تا حد قابل توجهی در پاسخهای تولیدی آنها از نظر معیار Makespan بهبود حاصل می‌گردد.



شکل ۴: مدل اتوماتای یادگیر [۷]

برای تولید پاسخ محیط، ابتدا درصد اختلاف پاسخ تکرار جاری با بهترین پاسخ تولید شده تا آن لحظه، محاسبه می‌شود بطوریکه حاصل عددی بین صفر و ۱۰۰ می‌باشد. سپس برای تولید اعداد پیوسته بین صفر و یک (S-Model)، درصد اختلاف بدست آمده را به عدد ۱۰۰ تقسیم می‌کنیم. بدین ترتیب پاسخ محیط همواره عددی در بازه‌ی پیوسته‌ی صفر و یک خواهد بود. لازم به تذکر است که هرچه عدد نهایی به صفر نزدیکتر باشد پاسخ محیط مطلوبتر بوده و هرچه این عدد به یک نزدیکتر باشد، نشاندهنده‌ی نامطلوبتر بودن پاسخ محیط به ازای اقدام انجام شده می‌باشد. الگوریتم پیشنهادی برای یادگیری اتوماتا را ICBR-LA می‌نامیم و شکل ۵ روش پیشنهادی و رویه کلی برای ایجاد پاسخ محیط را نشان می‌دهد.

۳-۴- مدل CBR مورد استفاده در روش ICBR-LA

در مدل CBR پیشنهاد شده در الگوریتم CBR-LA، مرحله تطبیق که بوسیله اتوماتای یادگیر انجام می‌شود، یک روش تطبیق ترکیبی از روشهای تبدیلی و ترکیبی است. در این مدل فرض می‌شود هر یک از موارد پایگاه مورد دارای سه بخش می‌باشند که عبارتند از ETC، نگاشت، و Makespan (کارایی بدست آمده از تجربه‌ی قبل) که با یک سه‌تایی (C, π , ETC) نشان داده می‌شود. برای بازیابی موارد مشابه، معیار تشابه بصورت تشابه بین دو ماتریس تعریف می‌شود که برای این منظور از رابطه‌ی

$$p_i(n+1) = (1-b)p_i(n) \quad (3)$$

$$p_j(n+1) = \frac{b}{r-1} + (1-b)p_j(n) \quad (4)$$

$$\forall j \quad j \neq i$$

لازم به ذکر است که در روابط فوق، a پارامتر پاداش و b پارامتر جریمه و r نشان دهنده‌ی تعداد اقدامهای ممکن برای اتوماتای یادگیر است. اتوماتاها بر حسب مقادیر پارامترهای پاداش و جریمه به چند نوع تقسیم می‌شوند. زمانی که $\alpha=b$ باشد، اتوماتا بنام L_{RP} ، اگر $b=0$ باشد، اتوماتا بنام L_{RI} و اگر $0 < b < \alpha < 1$ ، اتوماتا L_{REP} نامیده می‌شود. برای مطالعه بیشتر در مورد اتوماتای یادگیر، به منابع [۱۷ و ۱۸] رجوع شود.

۲-۴- مدل اتوماتای یادگیر پیشنهادی برای فاز تطبیق

اتوماتای یادگیر در مساله تخصیص وظیفه، بعنوان مکانیزم تطبیق مورد بکار می‌رود. شکل ۴ طرح کلی مدل اتوماتای یادگیر را در این کاربرد نشان می‌دهد. این مدل با مربوط ساختن هر وظیفه s_i در متاوظیفه، با یک اتوماتای یادگیر با ساختار متغیر ساخته می‌شود. از آنجائیکه وظایف می-

توانند به هر یک از ماشینهای محیط HC تخصیص داده

شوند، مجموعه اقدام تمام اتوماتاها، یکسان و شامل انتخاب ماشینهای محیط است ($r=\mu$). در [۷] فرض شده است که پاسخهای محیط از نوع P-Model می‌باشد. در این مقاله پاسخ محیط را از نوع S-Model فرض کرده و نشان داده شده است که با در نظر گرفتن پاسخ محیط از نوع S-Model، در محیطهای سازگار نتایج بهتری بدست می‌آید. الگوریتم یادگیری بکار گرفته شده در CBR-LA بدین صورت عمل می‌کند: اگر چنانچه مقدار Makespan در تکرار n ام ($c(n)$) کمتر از مقدار Makespan در تکرار $n-1$ ام ($c(n-1)$) باشد، آنگاه خروجی مطلوب فرض شده و پاسخ محیط در ازای اقدام انجام شده (P-Model) برابر با صفر منظور می‌شود؛ در غیر اینصورت، خروجی محیط نامطلوب فرض می‌شود (یک).

در روش پیشنهادی، زمانی که الگوریتم، پاسخ هر مرحله از عملکرد خود را با بهترین پاسخ تولیدی‌اش تا آن لحظه، مقایسه می‌کند، نتایج بهتری را تولید می‌کند. بنابراین با

فاصله‌ی اقلیدسی وزن دار بصورت زیر استفاده می‌شود:

$$ETC_i \in E \quad r = \sqrt{\sum_{i=1}^{|E|} (ETC[1][i] - ETC_i[1][i])^2}$$

بنابراین حقیقت که وظایف موجود در یک متاوظیفه دلخواه هستند، فرض میشود که ستون اول ماتریس ETC بصورت نزولی مرتب شده است تا فاصله اقلیدسی بعنوان یک معیار تشابه معنی‌دار باشد.

در غیر اینصورت اگر ماشین انتخابی تجربه‌ی قبلی در مساله‌ی جاری وجود نداشته باشد (ابعاد مساله‌ی جدید متفاوت باشد) آنگاه بردار احتمال اتوماتا بدون تغییر و دست نخورده باقی می‌ماند (احتمال انجام تمام اقدامها یکسان می‌باشد). در الگوریتم CBR-LA ابعاد مسائل ورودی و مسائل تجربه شده‌ی قبلی یکسان در نظر گرفته شده بود و لذا در انجام آزمایشهای مذکور تمام اتوماتاها حالت بایاس شده داشته و نتایج بدست آمده بدون مد نظر قرار دادن وضعیت بایاس نشدن اتوماتاها حاصل شده بودند. لذا در آزمایشهای انجام شده در این مقاله ابعاد مسائل را متنوع در نظر گرفته‌ایم.

در حالت بایاس شدن اتوماتا، برای اینکه در صورت متمایل کردن اتوماتا به سمت پاسخ نادرست (با دستکاری بردار احتمال)، اتوماتا قدرت اصلاح سریع عملکرد خود را داشته باشد، نرخ جریمه بسیار بزرگتر از نرخ پاداش در نظر گرفته می‌شود. پس از اتمام مرحله‌ی تطبیق راه‌حل، اتوماتای یادگیر به روشی که در بخش ۴-۲ شرح داده شد، تا زمان برآورده شدن یکی از شرایط خاتمه، شروع به تکرار الگوریتم یادگیری می‌کند.

طی مرحله‌ی تطبیق برای موارد بازیابی شده، اتوماتاهای یادگیر طی چالشی اقدام به یافتن راه‌حلی نسبتاً بهینه می‌کنند. در انتهای این مرحله به تعداد موارد بازیابی شده، راه‌حل‌های مختلفی (نگاشتهای مختلف) برای مساله-ی جدید وجود دارد که از بین آنها راه‌حلی که دارای معیار کارایی مناسبی (Makespan کمتر) باشد، بعنوان راه‌حل نهایی انتخاب می‌شود. اگر راه‌حل نهایی با مورد اصلی‌اش تفاوت عمده‌ای داشته باشد، بعنوان مورد جدید در پایگاه موارد ذخیره می‌شود.

۵- محیط شبیه‌سازی و نتایج آزمایشها

در این بخش، مجموعه‌ای از متاوظیفه‌ها در مقابل روش CBR-LA و تاثیر تغییر الگوریتم یادگیری اتوماتاها و نیز نوع پاسخ محیط در کاهش مقدار Makespan، نشان داده شده است. لازم به ذکر است که برای ایجاد ماتریس ETC از روش Baseline مورد استفاده در [۷] استفاده شده است. برای ایجاد ناهمگنی بالای وظایف، حد بالای دامنه مقادیر ممکن در بردار baseline با مقدار ۳۰۰۰ و برای

While (true)
Begin

LA_i selects its action for all $1 \leq i \leq \tau$;

Evaluate the makespan ;
If Current-Makespan < The-Best-Makespan **Then**

Set Reward for all LA_i that $1 \leq i \leq \tau$;

Calculate current-makespan's difference percentage with best- makespan;

If difference <= 0 **Then** $\beta = 0$;

Else

Set $\beta = \text{difference percentage} / 100$;

EndIf

جدید وارد سیستم می‌شود. سپس در مرحله‌ی بازیابی، موردهای مشابه به مساله‌ی جدید از بین موارد ذخیره شده در پایگاه موارد انتخاب و بازیابی می‌شوند. سپس فاز تطبیق آغاز می‌گردد. تطبیق، فرایند تبدیل نگاشت موارد بازیابی شده به منظور یافتن نگاشتی برای متاوظیفه جدید دریافت شده می‌باشد. این فرایند توسط مدل اتوماتای یادگیر انجام می‌شود. فاز تطبیق برای هر یک از موارد بازیابی شده انجام می‌شود.

در مرحله تطبیق مدل اتوماتای بیان شده در بخش ۴-۲ شروع به کار می‌کند. در صورتیکه در بخش راه‌حل مورد بازیابی شده، این وظیفه به ماشینی تخصیص داده شده باشد، احتمال انتخاب آن اقدام در بردار احتمال اتوماتا با مقداری نزدیک به یک مقداردهی اولیه می‌شود که اصطلاحاً در این حالت می‌گوئیم اتوماتا بایاس می‌شود.

ایجاد ناهمگنی پائین وظایف این مقدار را برابر با ۱۰۰ در نظر گرفته‌ایم. و نیز، ناهمگنی بالای ماشین با استفاده از مقدار ۱۰۰۰ برای ضرب کننده و ناهمگنی پائین ماشین با استفاده از مقدار ۱۰ برای این پارامتر تولید می‌شوند. همچنین، هشت حالت مختلف برای ویژگیهای ماتریس ETC در این مطالعه بکار می‌رود: ناهمگنی وظیفه بالا یا پائین، ناهمگنی ماشین بالا یا پائین، و یک نوع از سازگاری (سازگار، ناسازگار).

در این مطالعه ماتریسهای ETC را با ابعاد گوناگون در نظر می‌گیریم بدین معنی که تعداد وظایف و تعداد ماشینهای مسائل ورودی را متغیر فرض می‌کنیم. در تولید ماتریسهای ETC، تعداد وظایف بین ۱۰۰ تا ۱۵۰ و تعداد ماشینها بین ۴ تا ۸ ماشین متغیر در نظر گرفته شده است تا بدین طریق تنوع بیشتری را در بین مسائل ایجاد کنیم. در ضمن قبل از آغاز آزمایشها، پایگاه موارد با ۱۰۰ مورد تصادفی پر شده است و نیز تعداد موارد مشابه انتخابی از پایگاه موارد، ۳ مورد بوده و مدل LA به ازای اتوماتاهای بایاس نشده، از الگوریتم تقویتی L_{REP} با مقادیر $\alpha=0.5$ و $\beta=0.1$ برای اتوماتاهای بایاس شده، از مقادیر $\alpha=0.01$ و $\beta=0.1$ استفاده می‌کند. درضمن نتیجه هر آزمایش به ازای هر نوع ناهمگنی، میانگین ۲۰ بار اجرا می‌باشد.

۵-۱- آزمایش سری اول

در این آزمایش با مقایسه‌ی پاسخهای تولیدی دو الگوریتم CBR-LA و ICBR-LA برای مسائل یکسان، تاثیر الگوریتم یادگیری پیشنهاد شده بر کاهش مقدار کارایی Makespan را مورد بررسی قرار می‌دهیم. نتایج این آزمایش در جدول ۱ فهرست شده است. همانطوریکه در جدول ۱ مشخص است، در بدترین حالت یعنی محیط با ناهمگنی ماشین و وظیفه‌ی پائین، حداقل ۲۴/۱۱ درصد کاهش در مقدار Makespan بدست آمده است و در محیط با ناهمگنی بالای ماشین و وظیفه، این مقدار به ۳۶/۸۹ درصد افزایش یافته است که این موضوع بهبود قابل توجهی را نشان می‌دهد.

۵-۲- آزمایش سری دوم

در این آزمایش به بررسی و مقایسه‌ی تاثیر تغییر در نوع

پاسخ محیط می‌پردازیم. در این آزمایش نشان داده می‌شود که اگر پاسخهای محیط بصورت S-Model باشند، نسبت به حالتی که محیط بصورت P-Model عمل می‌کند، تا حدی در پاسخهای تولیدی بهبود حاصل می‌گردد. نتایج بدست آمده از این آزمایش به شرح جدول ۲ می‌باشد. با بررسی میانگین پاسخهای بدست آمده و مقایسه‌ی آنها با پاسخهای حالت P-Model از الگوریتم پیشنهادی، چنین برمی‌آید که حالت S-Model در محیطهای سازگار، موجب بهبود کارایی الگوریتم حداکثر تا ۴/۸۳ درصد می‌شود اما در مورد محیطهای ناسازگار تاثیر اندکی را موجب می‌شود. با اعمال هر دو تغییر پیشنهادی در مدل اتوماتای یادگیر الگوریتم CBR-LA (جدول ۳)، مشاهده می‌شود که پاسخهای بدست آمده حداقل تا حدود ۲۵/۱۹ درصد و حداکثر تا حدود ۳۹/۹۵ درصد، از لحاظ معیار کارایی Makespan بهبود پیدا می‌کنند.

جدول ۱: نتایج بدست آمده در نتیجه‌ی تغییر الگوریتم بکار رفته در مدل اتوماتای یادگیر

Consistency	Task Heterogeneity	Machine Heterogeneity	CBR-LA	ICBR-LA	Reduction in Makespan values
Incons.	High	High	19316288	13951096	27.77
	High	Low	211800.6	157053.3	25.85
	Low	High	662140.3	481184.3	27.33
	Low	Low	7723.55	5861.6	24.11
Cons.	High	High	26888098	16967742	36.89
	High	Low	249500.8	166106.2	33.42
	Low	High	753277.2	515709.2	31.54
	Low	Low	8452.1	5725.35	32.26

جدول ۲: مقادیر پاسخ تولیدی به ازای تغییر الگوریتم یادگیری و پاسخ محیط

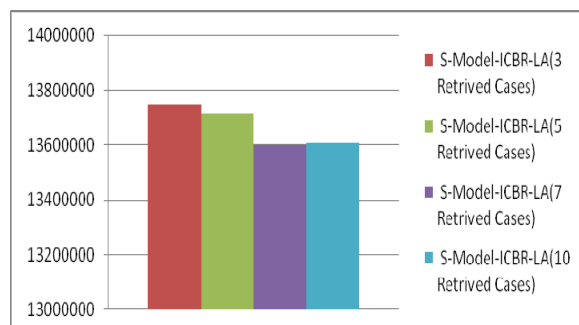
Consistency	Task Heterogeneity	Machine Heterogeneity	ICBR-LA (P-Model)	ICBR-LA (S-Model)	Reduction in Makespan values
Incons.	High	High	13951096	13747204.5	1.46
	High	Low	157053.3	154768.9	1.45
	Low	High	481184.3	473516.9	1.59
	Low	Low	5861.6	5777.35	1.44
Cons.	High	High	16967742	16147229	4.83
	High	Low	166106.2	159174.4	4.17
	Low	High	515709.2	510468.3	1.01
	Low	Low	5725.35	5492.9	4.06

۵-۳- آزمایش سری سوم

در این آزمایش به بررسی استفاده صحیح الگوریتم-ICBR LA از تجارب گذشته‌ی خود و تاثیر افزایش تعداد موارد انتخابی از پایگاه موارد در تولید پاسخهای نزدیک به بهینه می‌پردازیم. در این آزمایش نشان داده می‌شود که با افزایش دادن موارد مشابه انتخابی الگوریتم از ۳ مورد به ۵، ۷ و ۱۰ مورد، مقدار معیار کارایی نیز بیشتر کاهش یافته و پاسخهای بهتری تولید می‌شود. این آزمایش را برای محیط گاملا ناهمگن (محیط ناسازگار با ناهمگنی بالای ماشینها و وظایف) انجام دادیم و نتایج بدست آمده در شکل ۶ ارائه شده‌اند. در حالتی که الگوریتم ICBR-LA از تعداد ۱۰ تجربه خود به ازای هر یک از مسائل ورودی استفاده می‌کند، نسبت به حالتی که تنها ۳ تجربه بکار می‌برد، ۹۰/۹ درصد مقدار Makespan کاهش می‌یابد. بنابراین می‌توان نتیجه گرفت که استفاده صحیح و بیشتر الگوریتم از تجارب قبلی‌اش در تولید پاسخ مساله، موجب بهبود عملکرد آن می‌شود.

جدول ۳: مقادیر پاسخ تولیدی به ازای انجام هر دو تغییر پیشنهادی در الگوریتم CBR-LA

Consistency	Task Heterogeneity	Machine Heterogeneity	CBR-LA	ICBR-LA (S-Model)	Reduction in Makespan values
Incons.	High	High	19316288	13747204.5	28.83
	High	Low	211800.6	154768.9	26.92
	Low	High	662140.3	473516.9	28.49
Cons.	Low	Low	7723.55	5777.35	25.20
	High	High	26888098	16147229	39.95
	High	Low	249500.8	159174.4	36.20
	Low	High	753277.2	510468.3	32.23
	Low	Low	8452.1	5492.9	35.01



شکل ۶: مقایسه تاثیر افزایش تعداد موارد تجربه‌شده در تولید پاسخ

۶- نتیجه‌گیری

در این مقاله عملکرد مدل ترکیبی LA و CBR ارائه شده در منبع [۷] برای تخصیص وظایف در سیستم محاسباتی ناهمگن، تا حد زیادی بهبود داده شد. ایده اصلی، تغییر شرط بررسی عملکرد اتوماتاها می‌باشد که در هر مرحله، پاسخ تولیدی اتوماتا با بهترین پاسخی که تا آن لحظه تولید شده است، مقایسه می‌شود و در صورت بهبود یافتن پاسخ، اقدام اتوماتاها مورد پاداش و در غیراینصورت مورد جریمه قرار می‌گیرند. با انجام برخی آزمایشها نشان داده شد که تغییرات پیشنهادی برای الگوریتم یادگیری روش CBR-LA، کارایی بالاتری را در تولید پاسخها نسبت به روش اصلی ایجاد می‌کند. ایده‌ی دوم تغییر نوع پاسخهای محیط از P-Model به S-Model بود که با انجام آن، طبق آزمایشهای انجام شده نشان داده شد که الگوریتم اصلی تا حدی در تولید پاسخها بهبود می‌یابد. با بکار بردن الگوریتم جدید، آزمایشها کاهش ۳۹/۹۵ درصدی را در مقادیر Makespan نشان دادند که این موضوع، اثربخشی قابل توجهی در تولید پاسخهای نزدیک به بهینه است. در ضمن برای نشان دادن استفاده صحیح الگوریتم از تجارب گذشته، آزمایشهایی با تعداد تجارب بازایی شده متفاوت انجام شد که صحت این موضوع را نشان می‌دهد.

مراجع

- [1] D. Fernandez-Baca, "Allocating Modules to Processors in a Distributed System," *IEEE Transaction on Software Engineering*, Vol.15, pp.1427-1436, 1989.
- [2] T. D. Braun, H. J. Siegel, N. Beck, "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems," *Journal of Parallel and Distributed Computing*, Vol.61, pp. 810-837, 2001.
- [3] M. Coli, P. Palazzari, "Real Time Pipelined System Design through Simulated Annealing," *Journal of Systems Architecture*, Vol.42, pp.465-475, 1996.
- [4] K. Chow, B. Liu, "On Mapping Signal Processing Algorithms to a Heterogeneous Multiprocessor System," *International Conference on Acoustics, Speech, and Signal Processing*, Vol.3, pp.1585-1588, 1991.
- [5] H. Singh, A. Youssef, "Mapping and Scheduling Heterogeneous Task Graphs Using Genetic Algorithms," *5th IEEE Heterogeneous Computing Workshop*, pp. 86-97, 1996.
- [6] R. D. Venkataramana, N. Ranganathan, "Multiple Cost Optimization for Task Assignment in Heterogeneous Computing Systems Using Learning Automata," *IEEE 8th Heterogeneous Computing Workshop*, pp.137, 1999.
- [7] S. Ghanbari, M. R. Meybodi, K. Badie, "A Case-Based Recommender for Task Assignment in Heterogeneous Computing Systems", *Proceedings of the Fourth IEEE International Conference on Hybrid Intelligent Systems*, pp. 110-115, 2004.

-
- ¹ Case based Reasoning
 - ² Cognitive Science
 - ³ CaseBase
 - ⁴ Expected Time to Compute
 - ⁵ Variable Structure Learning Automata

- [8] S. K. Pal, S. C. K. Shiu, *Fundations of Soft Case-Based Reasoning*, Wiley Series on Intelligent Systems, A John Wiley & Sons INC. Publication, 2004.
- [9] R. Bergman, "Engineering Applications of Case-Based Reasoning," *Journal of Engineering Applications of Artificial Intelligence*, Vol. 12, pp.805, 1999.
- [10] J. L. Kolodner, D. B. Leake, "A Tutorial Introduction to Case-Base Reasoning, in Case-Base Reasoning Experiences" *Lessons and Future Directions*, MIT Press, pp. 31-66, 1996.
- [11] R. S. Freeman, R. DiGiorgio, "Assessing Alternative Technologies for the Cost-Effective Computation of Derivatives," *Applied Artificial Intelligence*, pp.491-503, 1997.
- [12] T. Cavasant, J. Kuh, "A Taxonomy of Scheduling in General-Purpose Distributed Computing System," *IEEE Transactions on software Engineering*, Vol.12, pp.662-675, 1996.
- [13] D. Gupta, P. Bepari, "Load Sharing in Distributed Systems," National Workshop on Distributed Computing, 1999.
- [14] R. Armstrong, D. Hensgen, T. Kidd, "The Relative Performance of Various Mapping Algorithms is independent of sizable variances in run-time predictions," *7th IEEE Heterogeneous Computing Workshop*, pp. 79-87, 1998.
- [15] L. Wang, H. J. Siegel, V. P. Roychowdhury, A. A. Maciejewski, "Task Matching and Scheduling in Heterogeneous Computing Environments Using a Genetic-Algorithm-based Approach," *Journal of Parallel Distributed Computing*, Vol. 47, pp.1-15, 1997.
- [16] A. A. Khokhar, V. K. Prasanna, M. E. Shaaban, C. L. Wang, "Heterogeneous Computing: Challenges and Opportunities," *IEEE Computing*, Vol.26, pp.18-27, 1993.
- [17] K. Narendra, M. A. L. Thathachar, *Learning Automata: An Introduction*, Prentice Hall, Englewood Cliffs, New Jersey, 1989.
- [18] K. Najim, A. S. Poznyak, *Learning Automata: Theory and Application*, Tarrytown, NY: Elsevier Science Ltd., Pergamon Press, Cambridge, England, 1994.
- [19] R. D. Venkataramana, N. Ranganathan, "New Cost Metrics for Iterative Task Assignment Algorithms in Heterogeneous Systems", *IEEE 9th Heterogeneous Computing Workshop*, Cancun, Mexico, pp. 160, 2000.
- [20] A. Aamodt, E. Plaza, "Case-Based Reasoning: Foundational Issues," *Methodological Variations and System Approaches AI Communications*, IOS Press, Vol. 7: 1, p. 39-59, 1994.
- [21] M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen, R. F. Freund, "Dynamic mapping of a class of independent tasks onto heterogeneous computing systems," *Journal of Parallel Distributed Computing*, Vol.59, pp.107-121, 1999.