

ترکیب بازی اقلیت و یادگیری موردی و بررسی نوع پایگاه مورد در تخصیص منابع سیستم‌های ناهمگن

زهره سلیمانی جلودار^{*}، بهروز معصومی[†]، محمدرضا میبدی[‡]

چکیده

یادگیری مبتنی بر مورد یکی از الگوریتم‌های یادگیری ماشین است که برای حل مسأله جدید، از تجربه‌های گذشته و روش‌های حلی که مشابه با مسأله جاری باشد، استفاده می‌کند. بازی اقلیت نیز، از حوزه مسائل نظریه بازی‌ها است که در سیستم‌های چندعامله بکار می‌رود، در این بازی تعداد فردی عامل با هم رقابت کرده و دسته ای برنده و دسته ای بازنده می‌شوند. این عامل‌ها در اثر تعامل با محیط به مرور زمان یاد می‌گیرند که چگونه با هم رقابت یا همکاری کنند به طوریکه بیشترین سود را دریافت کنند. از آنجائیکه زمان حل مسأله اهمیت زیادی دارد، در یادگیری مبتنی بر مورد از اتوماتای یادگیر و بازی اقلیت استفاده شد تا زمان یا هزینه تخصیص منابع را کاهش دهد. در این مقاله الگوریتم MG-ICBR-LA ارائه شده است. با استفاده از این الگوریتم در مسأله تخصیص منابع سیستم‌های ناهمگن، خاصیت رقابت اقلیتی اعمال شد. آزمایش‌ها با دو نوع پایگاه مورد مستقل و مشترک انجام شده است. نتایج آزمایش‌های صورت گرفته نشان می‌دهند که هزینه تخصیص منابع در روش پیشنهادی کاهش یافته است.

کلمات کلیدی

استنتاج مبتنی بر مورد، اتوماتای یادگیر، بازی اقلیت، تخصیص منابع، پایگاه مورد، سیستم‌های چند عامله.

^{*} دانشکده برق، رایانه و فناوری اطلاعات، دانشگاه آزاد اسلامی قزوین، ایران. Z.soleimani@qiau.ac.ir

[†] عضو هیات علمی دانشکده برق، رایانه و فناوری اطلاعات، دانشگاه آزاد اسلامی قزوین، ایران. masoumi@qiau.ac.ir

[‡] عضو هیات علمی دانشکده برق، رایانه و فناوری اطلاعات، دانشگاه امیرکبیر، تهران، ایران. mmeybodi@aut.ac.ir

1- مقدمه

یکی از کارهای مهم برای رسیدن به بهترین کارایی در محیط‌های محاسباتی ناهمگن، تخصیص دادن برنامه‌های کاربردی به ماشین‌ها و زمانبندی آن برنامه‌ها بر روی هر ماشین است [3]. تخصیص و زمانبندی برنامه‌های کاربردی بعنوان نگاشت تعریف می‌شود. مسأله‌ی نگاشت، مسأله‌ی مدیریت منابع یا تخصیص منابع نیز نامیده می‌شود [5]. روش‌های مختلفی برای حل مسأله نگاشت معرفی شده است، مثلاً الگوریتم Simulated-Annealing، ژنتیک و اتوماتای یادگیر [7]. همه روش‌های فوق، بدون استفاده از تجربیات گذشته، به تخصیص وظیفه می‌پردازند. همچنین الگوریتم Min-Min و الگوریتم Max-Min نیز در این زمینه بکار رفته‌اند و نتایج خوبی هم داشته‌اند. در مقاله [4] آقای قنبری روشی را مبتنی بر CBR^1 و اتوماتای یادگیر ارائه داده‌اند که از تجربیات گذشته برای حل این مسأله استفاده می‌کند و نام الگوریتم خود را $CBR-LA$ نهاده‌اند. در مقاله [1]، آقای منیری و همکارانش با بهبود الگوریتم یادشده، الگوریتم $ICBR-LA^2$ را ارائه کردند و عامل‌ها با استفاده از سیاست همکاری در جهت تخصیص وظایف در سیستم محاسباتی ناهمگن با یکدیگر همکاری کرده و راه‌حلی نزدیک به بهینه را برای مسائل ورودی (با ابعاد متفاوت) ارائه کردند. در کارهای انجام شده، همه عامل‌ها همگن بوده و از یک نوع الگوریتم استفاده می‌کردند، همچنین این عامل‌ها از پایگاه مورد مستقل استفاده می‌کردند.

در این مقاله با ایجاد رقابت بین وظایف ورودی، سعی شده هزینه تخصیص وظایف به ماشین‌ها کاهش یافته و در مجموع کارایی سیستم افزایش یابد. برای ایجاد این رقابت از بازی اقلیت استاندارد استفاده شده است؛ همچنین در این رقابت از دو نوع پایگاه مورد مشترک و پایگاه مورد مستقل استفاده می‌شود.

در ادامه مقاله در بخش 2، استنتاج CBR و $ICBR$ معرفی شده است. در بخش 3، تعریفی در مورد بازی اقلیت ارائه شده است. در بخش 4، روش پیشنهادی $MG-ICBR-LA^3$ همراه با دو نوع پایگاه مورد مستقل و مشترک معرفی می‌شود سپس محیط شبیه‌سازی، نتایج آزمایش‌های انجام شده و مقایسه آنها نشان داده می‌شوند. در پایان در بخش 5، نتیجه‌گیری نهایی و کارهای آینده مورد بررسی قرار می‌گیرد.

2- استنتاج مبتنی بر مورد CBR و کاربردهای آن

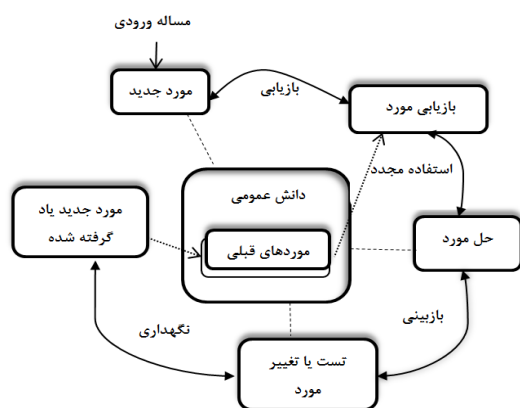
استنتاج بر مبنای مورد یکی از تکنیک‌های تصمیم‌گیری در خصوص راه حل مشکلات می‌باشد. این سیستم‌ها با استفاده از راه حل مشکلات قبلی پیش آمده در یک دامنه خاص سعی می‌کند تا برای مشکل موجود راه حل ارائه کند. ایده اصلی آن مبتنی بر این است که مشکلات مشابه راه حل‌های مشابه نیز خواهند داشت. بر خلاف سایر تکنیک‌های مبتنی بر دانش، CBR بر روی تجربه حل مسأله خاصی تمرکز می‌کند که برگرفته از موارد جمع‌آوری شده در پایگاه موارد است، همچنین CBR روشی است برای یادگیری مستمر افزایشی، زیرا

یک تجربه جدید در هر بار حل شدن یک مسأله نگهداری می‌شود و آن راه حل برای مسائل بعدی فوراً قابل دسترسی است. این قابلیت CBR منجر به یادگیری می‌شود. یادگیری در CBR پس از حل کردن یک مسأله رخ می‌دهد، بنابراین وقتی که یک مسأله با موفقیت حل شد تجربه آن نگهداری می‌شود تا در آینده برای حل یک مسأله مشابه استفاده شود [4].

2-1- چرخه CBR

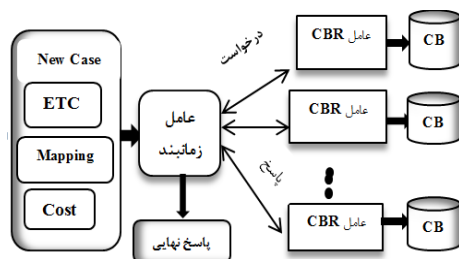
چرخه CBR در شکل (1) نشان داده شده است، این چرخه فعالیت‌های اصلی CBR در حین حل مسائل را نشان می‌دهد، شامل [1, 4, 6]:

- بازیافت: بازیابی موردهای مشابه قبلی
- استفاده مجدد: استفاده مجدد از مورد برای حل مشکل جدید
- اصلاح کردن: بازیابی و اصلاح راه حل پیشنهاد شده
- نگهداری: حفظ راه حل ارائه شده به عنوان یک مورد جدید



شکل (1): چرخه CBR

گام آخر همان فاز یادگیری در CBR است. برای تطبیق راه حل روش‌های متنوعی وجود دارد که در الگوریتم $CBR-LA$ از یک مدل اتوماتای یادگیر برای تطبیق راه حل مورد تجربه شده به مسأله جدید استفاده می‌کند. توابع پاداش و جریمه اتوماتا در مقاله‌های [1, 4, 6] آمده است. عملکرد این الگوریتم بدین صورت است که اگر چنانچه مقدار $Cost$ در تکرار n کمتر از مقدار $Cost$ در تکرار $n-1$ باشد، آنگاه خروجی مطلوب فرض شده و پاسخ محیط در ازای اقدام انجام شده با توجه به P مدل بودن محیط برابر با صفر می‌شود؛ یعنی در این الگوریتم هر یک از اتوماتاهای مربوط به وظایف، پس از انجام اقدام مورد نظرشان طبق الگوریتم فوق پاداش یا جریمه دریافت می‌کنند. طرز کار این الگوریتم در شکل (2) درج شده است.



شکل (2): مدل سیستم CBR

با توجه به شکل (2)، هر مورد سه بخش دارد که عبارتند از ماتریس ETC، نگاشت و Cost که با یک سه‌تایی (ETC, π , C) نشان داده می‌شود. ماتریس ETC در واقع دارای سطرها و ستون‌های به ترتیب متاوظیفه و ماشین است. برای بازیابی موارد مشابه، معیار تشابه بصورت تشابه بین دو ماتریس تعریف می‌شود؛ یعنی ماتریس ETC متاوظیفه جدید دریافت شده و ماتریس ETC موارد ذخیره شده. تابع تشخیص تشابه H بصورت $r \in R \rightarrow H:(ETC_{new}, ETC_{old})$ است که این تابع فاصله اقلیدسی دو ماتریس ETC را برمی‌گرداند که نشان‌دهنده مقدار تشابه بین دو ماتریس است. در کل، به دلیل پیچیدگی زمانی بالا، تنها فاصله اقلیدسی ستون اول دو ماتریس را بعنوان یک فاصله تخمینی محاسبه می‌کنیم. بدین ترتیب، هزینه محاسباتی $O(|S| \times |E|)$ خواهد شد. هزینه هم همان مدت زمانی است که طول می‌کشد تا یک متاوظیفه اجرا شود. در هر تکرار الگوریتم، یک بردار به طول تعداد وظیفه، نگاشت π را نشان می‌دهد. هدف عامل زمانبند هم یافتن نگاشتی از وظایف موجود در متاوظیفه نسبت به پردازنده‌های موجود در سیستم ناهمگن است.

میزان تنوع در زمان‌های اجرای وظایف یک متاوظیفه برای یک ماشین، بعنوان ناهمگنی وظیفه تعریف می‌شود. ناهمگنی بالای وظیفه با $w_b=3000$ و ناهمگنی پائین وظیفه با $w_b=100$ نشان داده می‌شود. ناهمگنی ماشین، نشان‌دهنده تنوع ممکن در زمان‌های اجرای یک وظیفه بر روی تمام ماشین‌های محیط است. مقادیر ناهمگنی بالای ماشین با استفاده از $w_r=1000$ تولید می‌شوند درحالی‌که مقادیر ناهمگنی پائین ماشین با استفاده از $w_r=10$ تولید می‌شوند [64,1]. برای ایجاد تنوع بیشتر در ویژگی‌های ماتریس‌های ETC، سازگاری-های مختلفی برای ماتریس ETC اعمال شده است. به یک ماتریس ETC سازگار گفته می‌شود اگر هر زمان که یک ماشین m_j ، هر وظیفه t_i را سریعتر از ماشین m_k اجرا کند، آنگاه ماشین m_j تمام وظایف را سریعتر از ماشین m_k اجرا می‌کند [6]. ماتریس‌های سازگار با مرتب کردن مستقل هر سطر ماتریس ETC تولید می‌شوند، که ماشین m_0 همواره سریعترین بوده و ماشین $m_{\mu-1}$ کندترین است. در مقابل، ماتریس‌های ناسازگار وضعیتی را مشخص می‌کنند که ماشین m_j ممکن است به ازای برخی از وظایف، سریعتر از ماشین m_k و به ازای برخی دیگر از وظایف کندتر باشد. این ماتریس‌ها در همان حالت تصادفی و نامرتب که تولید می‌شوند، رها می‌شوند (یعنی، هیچ سازگاری به آنها اعمال نمی‌شود).

2-2- استنتاج مبتنی بر مورد بهبود یافته ICBR-LA

در الگوریتم CBR، مقدار Cost با دو مقدار قبلی مقایسه می‌شود. در اینجا یک مشکلی وجود داشت، ممکن بود در سه تکرار اخیر پاسخ‌های تولی‌دی بصورت رابطه (1) باشند [1]:

$$T(n-1) << T(n) >> T(n+1) \quad (1)$$

یعنی پاسخ تولیدی در مرحله $n+1$ نسبت به پاسخ مرحله $n-1$ نامطلوب‌تر باشد اما بدلیل اینکه الگوریتم فوق همواره دو پاسخ اخیر را در نظر می‌گیرد، بنابراین در دادن پاداش و جریمه به صورت محلی تصمیم می‌گیرد. به طور مثال ممکن است در مرحله $n-1$ پاسخ تولیدی به ازای اقدام اتوماتاها 100 باشد و این پاسخ در تکرار m 2000 یعنی 20 برابر پاسخ قبلی باشد و در نتیجه اتوماتا باید جریمه شود. حال فرض که در ادامه تکرار $n+1$ ام بازی پاسخ 1500 تولید شود، این پاسخ نسبت به اقدام قبلی بهتر است و طبق الگوریتم فوق باید پاداش بگیرد حال آنکه با توجه به پاسخ 100 مرحله $n-1$ ، پاسخ 1500 خیلی بد است.

به منظور حل این مشکل با تغییر در الگوریتم CBR، الگوریتم ICBR معرفی شد [6,1]. در الگوریتم ICBR هر یک از اتوماتاهای یادگیر پاسخ هر مرحله از عملکرد خود را با بهترین پاسخ تولیدی تا آن لحظه، مقایسه می‌کند. در اینصورت زمانی که اتوماتاها در فضای پاسخ یک پاسخ تقریباً بهینه پیدا کردند، این پاسخ را در تکرار بعدی از دست نمی‌دهند و بدین ترتیب رفتار تصادفی اتوماتاها در حین جستجوی در فضای پاسخ کنترل می‌شود.

نتایج شبیه‌سازی در [1] نشان می‌دهد که این الگوریتم بهتر از CBR-LA عمل کرده است، اما از دو الگوریتم استاندارد MaxMin و MinMin بدتر بوده است. شکل (3) کد الگوریتم ICBR-LA را نشان می‌دهد.

```
% % Start Learning Phase for Automata
While (true)
Begin
LAi select its action for all 1 ≤ i ≤ t;
Evaluate Cost;
If {current Cost < Best Cost Then set Reward for all automata else
set punish for all automata}
If no change occurs in Cost for 150 iterations or 10000 iteration is over,
Exit (); Then
End
*****
% % Start ICBR-LA
Begin
Find n closest cases to the new-case & put them in
For each case in Θ Do
Set bias LA to the solution of the case;
Resolve solution;
End For
Return the solution with minimum Cost;
End
```

شکل (3): الگوریتم ICBR-LA [1]

3- بازی اقلیت

بازی اقلیت⁴ اولین بار توسط رایان آرتور [8] در سال 1994 با ارائه مسأله رستوران ال فارول مطرح شده است. صورت این مسأله بدین صورت است که N فرد (عامل) به طور مستقل تصمیم می‌گیرند که برای تفریح در شب خاصی به یک رستوران بروند یا نروند. فضای آن جا محدود است و زمانی به آنها خوش می‌گذرد که رستوران خیلی

چنانچه $A(t)$ کمتر از $(N-1)/2$ باشد یعنی گروه 1 در سمت اقلیت بوده و بازی را برده است و اگر بیشتر از $(N-1)/2$ باشد گروه 0 بازی را برده و پاداش می گیرد، در برخی نسخه های بازی اقلیت، گروه بازنده جریمه می شود [9,10]. در معادله (2)، N تعداد عامل ها است.

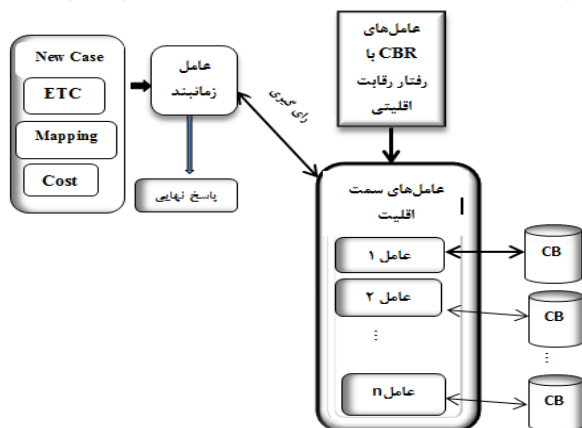
$$A(t) = \sum_{i=1}^N a_i(t) \quad (2)$$

3-1- بازی تخصیص منبع

تخصیص منابع از مسأله ال فارول الهام گرفته شده است، در این مسأله، رستوران یک منبع مشترک و محدود بود. در واقع اهمیت مسأله در این است که عامل ها چه منابعی را در چه زمانی انتخاب نمایند که در این بین هم کیفیت سرویس افت نکند و هم تمامی منابع به خوبی مورد استفاده قرار گیرند. از بازی اقلیت به عنوان بازی تخصیص منبع یاد شده است [11]. آقای Galstyan برای اولین بار مسأله تخصیص منابع با استفاده از بازی اقلیت را مطرح کرد [12]. همچنین از بازی اقلیت ژنتیکی در تخصیص منابع شبکه های حسگر استفاده شده است [13].

4- الگوریتم MG-ICBR-LA (روش پیشنهادی)

از آنجائیکه نتایج شبیه سازی در مرجع [1] نسبت به الگوریتم استاندارد MaxMin خوب نبود، با اعمال تغییراتی در الگوریتم ICBR-LA، الگوریتم MG-ICBR-LA را همراه با دو نوع پایگاه مورد معرفی می کنیم. در الگوریتم پیشنهادی هر وظیفه را یک عامل فرض شد و سپس بین این وظایف ورودی رقابت اقلیتی ایجاد شد بطوریکه وظایف سمت برنده که همان سمت اقلیت است به اتوماتا وصل شده و اتوماتای تولید شده به ازای وظایف در فضای پاسخ جستجو کرده و در نهایت این وظایف برنده را به ماشین مربوطه نگاشت می دهد. در این مقاله از بازی اقلیت نوع استاندارد استفاده شد؛ همچنین دو نوع پایگاه مورد استفاده شده در شکل (5) و شکل (6) نشان داده شده است. شکل (5) استفاده از پایگاه مورد مستقل⁵ برای هر عامل را نشان می دهد، یعنی هر عامل تجربه منحصر بفرد خود را در حل مسأله ورودی دارد. در شکل (5)، عامل n ، آخرین عامل در سمت اقلیت است. در کل تعداد عامل های سمت اقلیت $(N-1)/2$ است که N تعداد کل عامل ها است.



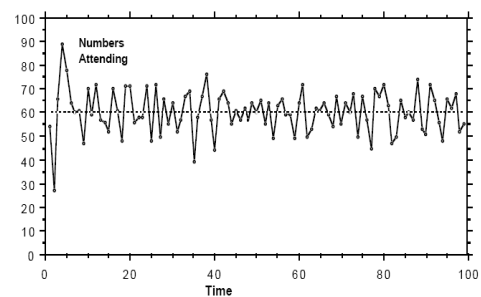
شکل (5): استفاده از پایگاه مورد مستقل در الگوریتم MG-ICBR-LA

شلوغ نباشد. یعنی اگر کمتر از $a.N$ ($a < 1$) نفر در رستوران باشند رستوران خلوت تلقی می شود و مشتریان آن سرویس مناسبی را دریافت کرده و به آن ها خوش می گذرد. با توجه به ظرفیت محدود رستوران (به طور مثال 60 نفر)، عامل ها دو گزینه برای انتخاب دارند:

1. به رستوران بروند. عامل ها این گزینه را در شرایطی انتخاب می کنند که پیش بینی کنند جمعیت حاضر کمتر از $a.N$ نفر می باشد.

2. به رستوران نروند و در خانه بمانند. عامل ها این گزینه را در شرایطی انتخاب می کنند که پیش بینی کنند جمعیت حاضر در رستوران از حد آستانه بیشتر می باشد.

در شکل (4) نوسانات حضور 100 عامل مشتری در رستوران ال فارول نشان داده شده است. هر چه ارتفاع نوسانات کمتر باشد، یعنی عامل های بیشتری توانسته اند پیش بینی درست انجام دهند.



شکل (4): بازی اقلیت برای 100 تکرار با 100 عامل مشتری رستوران [8]

سپس آقایان Challet و Zhang برای اولین بار در دانشگاه فرایبورگ از روی مسأله ال فارول، بازی اقلیت را فرمول بندی ریاضی کردند [9]. در این بازی، تعدادی بازیکن (عامل) در هر مرحله از بازی یکی از دو دسته 1 یا 0 را انتخاب می کنند. بازیکنانی که در دسته اقلیت قرار بگیرند امتیاز دریافت می کنند و بقیه امتیازی دریافت نمی کنند. در این مدل بازیکنان در هر مرحله براساس یک استراتژی تصمیم می گیرند. هر بازیکن تعدادی استراتژی دارد و از بین آنها بهترین استراتژی در مراحل قبلی را برای مرحله بعد انتخاب می کند و براساس آن بازی می کند. استراتژی که دسته اقلیت را به درستی پیش بینی می کند استراتژی مناسب تر تلقی می شود و در مراحل بعدی شانس بیشتری دارد. به استراتژی برنده پاداش تعلق می گیرد و استراتژی بازنده نیز تنبیه می شود. هر استراتژی 2^P است و هر $P=2^m$ که m همان حافظه و P تاریخچه مراحل قبلی بازی با توجه به تعداد خانه های حافظه است، بنابراین حافظه نقش مهمی در تعیین استراتژی دارد [2]. البته در نسخه بازی اقلیت بدون حافظه ثابت شده که حافظه نقشی در نتیجه پایانی نخواهد داشت، در واقع نتایج بازی هیچ ارتباطی به یک بازخورد صحیح و واقعی از مرحله قبلی بازی اقلیت به عنوان تاریخچه ندارد و اگر این تاریخچه به طور تصادفی ایجاد شود باز هم نتایج حاصله با نتایج اولیه مطابق و یکسان است [10].

Action یا همان عمل عامل ها به صورت $\{0, 1\} \in a(t)$ است، سپس در هر تکرار از بازی Action ها طبق معادله (1) جمع می شود.

جدید می‌گردد؛ در نهایت پاسخ‌ها به زمانبند ارسال می‌شود و عامل زمانبند بهترین پاسخ را از میان آنها انتخاب می‌کند.

4-1- شبیه‌سازی روش پیشنهادی

در پیاده‌سازی‌ها، ماتریس‌های ETC را با ابعاد گوناگون در نظر می‌گیریم بدین معنی که تعداد وظایف و تعداد ماشین‌های مسائل ورودی را ثابت و یا متغیر فرض می‌کنیم؛ به همین دلیل در حالت ثابت اتوماتای مربوط به وظایف به صورت بایاس شده و در حالت متغیر، اتوماتا به صورت بایاس نشده شروع به اجرا خواهد کرد. برای ایجاد تنوع در مسائل، در تولید ماتریس‌های ETC، تعداد وظایف بین 101 تا 151 و تعداد ماشین‌ها بین 4 تا 8 ماشین در نظر گرفته شده است. در ضمن تعداد موارد مشابه انتخابی از پایگاه مورد، 3 مورد می‌باشد. میزان پاداش در اتوماتای یادگیر برای هر کدام از عامل‌ها نیز طبق جدول (1) در نظر گرفته شده است؛ طبق این جدول اتوماتا برای متد اول از نوع RεP و برای متد دوم از نوع RεR است. میزان حافظه برای عامل‌ها 1 و تعداد استراتژی هم 4 است. همچنین هشت حالت مختلف برای ویژگی‌های ماتریس ETC در این قسمت بررسی می‌شود یعنی ناهمگنی وظیفه⁷ پائین یا بالا، ناهمگنی ماشین⁸ پائین یا بالا و یک نوع از سازگاری (سازگار⁹، ناسازگار¹⁰). همچنین هر مسئله 5 بار اجرا شده است. نتیجه نهایی برای هر کدام از حالت‌ها، میانگین پاسخ اجراهای متفاوت بر روی یک مسئله ورودی می‌باشد.

در جدول (2) خروجی حاصل از شبیه‌سازی روش پیشنهادی در حالی که از پایگاه مورد مستقل و هر دو متد جدول (1) استفاده کرده است، نشان داده شد و همچنین در جدول (3) خروجی حاصل از شبیه‌سازی روش پیشنهادی در حالی که از پایگاه مورد مشترک و هر دو متد جدول (1) استفاده کرده است، نشان داده شد. در پایان در جدول (4) و جدول (5) دو الگوریتم پیشنهادی با دو الگوریتم استاندارد MaxMin و MinMin مقایسه شده است. نقاط پرنرگ در جدول‌ها نشان‌دهنده نقاط بهبود یافته الگوریتم پیشنهادی است.

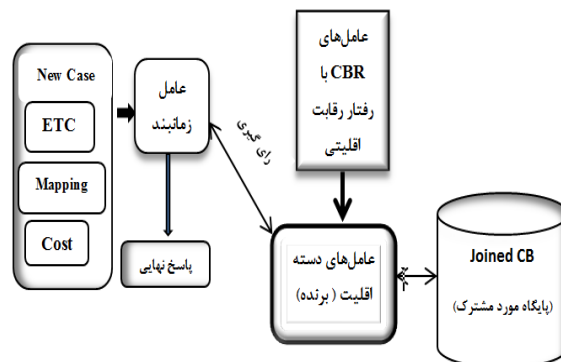
جدول(1): مقادیر پاداش و جریمه اعمال شده در روش MG-ICBR-LA

متد	اتوماتای بایاس شده		اتوماتای بایاس نشده	
	پاداش	جریمه	پاداش	جریمه
متد 1	0.0005	0.0001	0.0005	0.0001
متد 2	0.0005	0.001	0.0005	0.0001

جدول(2): مقایسه نتایج حاصل از اجرای الگوریتم MG-ICBR-LA با ICBR-LA به همراه پایگاه مورد مستقل

Consistency	ناهمگنی ماشین	ناهمگنی وظیفه	ICBR-LA	MG-ICBR-LA+ method one	MG-ICBR-LA+ method two
ناسازگاری ماتریس	High	High	13951096	4750139	5369592
	High	Low	157053.3	66677	67205.5
	Low	High	481184.3	281062	155309
	Low	Low	5861.6	2421	2124.5
سازگاری ماتریس	High	High	16967742	7033509	5647497
	High	Low	166106.2	75836.7	77490
	Low	High	515709.2	216972	149197
	Low	Low	5725.35	2277	2307

در شکل (6)، عامل‌ها از پایگاه موارد مشترک⁶ استفاده کردند، یعنی همه عامل‌ها از تجربه یکسانی برای حل مساله ورودی استفاده می‌کنند و در پایان تجربه‌های خود را در یک پایگاه مورد مشترک ذخیره می‌کنند تا برای مرحله بعدی استفاده شود.



شکل (6): استفاده از پایگاه مورد مشترک در الگوریتم MG-ICBR-LA

همانطور که در شکل (6) نشان داده شده است، تنها یک پایگاه مورد CB برای همه عامل‌ها وجود دارد. این عامل‌ها با الگوریتم MG-ICBR-LA به حل مسئله جاری می‌پردازند. روش کار این مدل پیشنهادی به این صورت است که ابتدا مسئله به صورت یک مورد جدید وارد می‌شود که دارای ETC و Cost می‌باشد، زمانبند پس از دریافت مسئله جدید، آن را به تمامی عامل‌ها ارسال می‌کند.

فرض می‌کنیم هر عامل می‌تواند تا 3 مورد مشابه مسئله ورودی را از پایگاه استخراج کند بنابراین همه عامل‌ها با متد خود بر روی پایگاه مورد مشترک کار می‌کنند و سه مورد مشابه به مسئله جاری را انتخاب می‌کنند و به ازای هر یک از این سه مورد مشابه، فاز تطبیق را انجام می‌دهند و سپس با ایجاد رقابت بازی اقلیت، به ازای وظایف برنده اتوماتا تولید می‌شود. اتوماتا اقدام به جستجوی فضای پاسخ می‌کند و در نهایت از بین پاسخ‌هایی که برای این سه مورد مشابه بدست آمده است، هر عامل بهترین جوابش به عامل زمانبند ارسال می‌کند. عامل زمانبند هم تا رسیدن پاسخ بقیه عامل‌ها منتظر می‌ماند و پس از دریافت نظر بقیه عامل‌ها، راه حل نهایی را ارائه می‌کند. در آخر نیز جواب نهایی به عنوان یک مورد جدید در پایگاه مورد اشتراکی ذخیره می‌شود؛ اما در مدل شکل(5) که پایگاه موردها مستقل از هم هستند هر عامل در پایگاه مورد خود به دنبال پاسخ مناسب برای حل مسئله

جدول(3): مقایسه نتایج حاصل از اجرای الگوریتم MG-ICBR-LA با الگوریتم ICBR-LA به همراه پایگاه مورد مشترک

Consistency	نا همگنی وظیفه	نا همگنی ماشین	ICBR-LA	MG-ICBR-LA+ method one	MG-ICBR-LA+ method two
ناسازگاری ماتریس	High	High	13951096	4761040	4896305
	High	Low	157053.3	53335	56284
	Low	High	481184.3	131840	130340
	Low	Low	5861.6	1699	2196
سازگاری ماتریس	High	High	16967742	6816468.5	7109495.6
	High	Low	166106.2	99320	29673
	Low	High	515709.2	274897	118606
	Low	Low	5725.35	2232.5	784

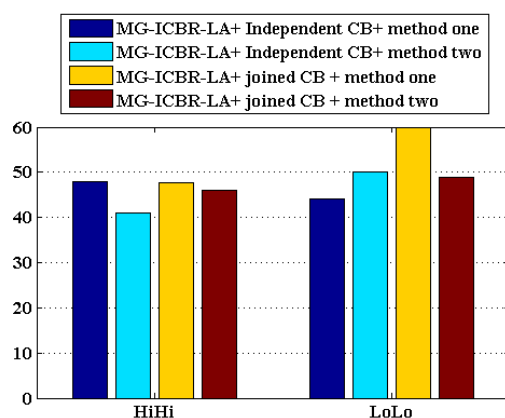
جدول(4): مقایسه نتایج حاصل از اجرای الگوریتم MG-ICBR-LA با الگوریتم‌های استاندارد MaxMin و MinMin با پایگاه مورد مستقل

Consistency	نا همگنی وظیفه	نا همگنی ماشین	MaxMin	MinMin	MG-ICBR-LA+ Independent CB+ method one	MG-ICBR-LA+ Independent CB+ method two
ناسازگاری ماتریس	High	High	9135809	5460156	4750139	5369592
	High	Low	109420.9	63562.5	66677	67205.5
	Low	High	299151	169390	281062	155309
	Low	Low	4335.154	2494.5	2421	2124.5
سازگاری ماتریس	High	High	9078309	5967740	7033509	5647497
	High	Low	127206.1	88076	75836.7	77490
	Low	High	295614.4	191751	216972	149197
	Low	Low	4576.5	3035.6	2277	2307

جدول(5): مقایسه نتایج حاصل از اجرای الگوریتم MG-ICBR-LA با الگوریتم‌های استاندارد MaxMin و MinMin با پایگاه مورد مشترک

Consistency	نا همگنی وظیفه	نا همگنی ماشین	MaxMin	MinMin	MG-ICBR-LA+ joined CB + method one	MG-ICBR-LA+ joined CB + method two
ناسازگاری ماتریس	High	High	9135809	5460156	4761040	4896305
	High	Low	109420.9	63562.5	53335	56284
	Low	High	299151	169390	131840	130340
	Low	Low	4335.154	2494.5	1699	2196
سازگاری ماتریس	High	High	9078309	5967740	6816468.5	7109495.6
	High	Low	127206.1	88076	99320	29673
	Low	High	295614.4	191751	274897	118606
	Low	Low	4576.5	3035.6	2232.5	784

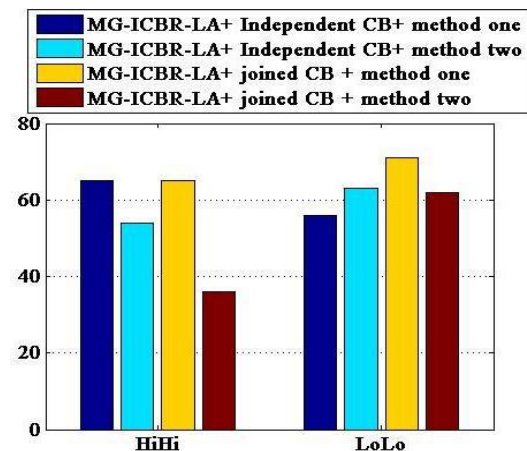
در شکل (8) نمودارهای کاهش هزینه روش پیشنهادی در مقابل الگوریتم استاندارد MaxMin در دو حالت ناهمگنی بالای ماشین و وظیفه و ناهمگنی پایین ماشین و وظیفه رسم شده است.



شکل(8): میزان کاهش هزینه روش‌های پیشنهادی در مقابل MaxMin

در شکل (9)، نمودارهای کاهش هزینه روش پیشنهادی در مقابل الگوریتم استاندارد MinMin در دو حالت ناهمگنی بالای ماشین و وظیفه و ناهمگنی پایین ماشین و وظیفه رسم شده است.

برای مقایسه خروجی‌ها، دو حالت ناهمگنی بالای ماشین و وظیفه (HiHi)¹¹ و ناهمگنی پایین ماشین و وظیفه (LoLo)¹²، به تصادف انتخاب شده است. در شکل (7)، نمودارهای کاهش هزینه¹³ نشان داده شده در جدول (2) و جدول (3) در مقابل روش-ICBR LA رسم شده است؛ در ضمن این خروجی‌ها به درصد هستند.

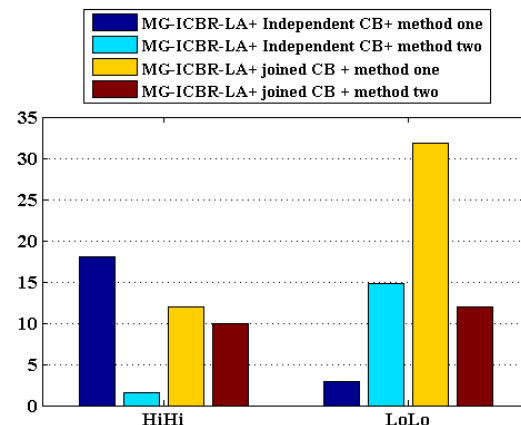


شکل(7): میزان کاهش هزینه روش‌های پیشنهادی در مقابل ICBR-LA

[2] زهرا سلیمانی، بهروز معصومی، "بررسی نقش پارامترهای حافظه و استراتژی عامل‌ها در تعادل بازار با استفاده از بازی اقلیت ژنتیکی"، دومین کنفرانس ملی دانشگاه آزاد لاهیجان، آبان 1391.

- [3] S.Ali, T.Braun, H. Siegel, A. Maciejewski, "Characterizing Resource Allocation Heuristics for Heterogeneous Computing Systems", *Advances In Computers*, Vol. 63, DOI 10.1016/S0065-2458(04)63003-8, Elsevier Inc, 2005.
- [4] 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.
- [5] L. N. Nassif, M. Ahmed, J. M. S.Nogueira, R.Impey, "Negotiation Process for Resource Allocation in Grid Using a Multi-agent System", In *Proceedings of MATA*, pp. 158-167, 2004.
- [6] S .SabaMoniri, B.Masoumi and M. R.Meybodi, "ICBR-LA: A Hybrid Algorithm for Task Allocation for Homogenous Computing System", *Proceedings of The Third Joint Congress on Fuzzy and Intelligent Systems*, University of Yazd, Yazd, Iran, 15-17 July 2009.
- [7] R.Venkataramana, N .Ranganathan, "Multiple Cost Optimization for Task Assignment in Heterogeneous Computing Systems Using Learning Automata" *IEEE 8th Heterogeneous Computing Workshop*, pp.137, 1999.
- [8] B.Arthur, "Inductive Reasoning And Bounded Rationality (The El Farol Problem)", *American Economic Review*, Vol.84, Pp.406-411, 1994.
- [9] D.Challet, Y.Zhang, "Emergence of Cooperation and Organization in an Evolutionary Game", *Physica Elsevier Physica A* 246, Pp. 407-418, 1997.
- [10] A.Cavagna, "Irrelevance of Memory in the Minority Game", *Arxiv, Physical Review E, Journals*, Vol.59, Issue 4, 1999.
- [11] Y.Chevaletre, P. E. Dunne, U.Endriss, "Issues in Multiagent Resource Allocation", *Slovenian Society Informatika*, ISSN: 0350-5596, 2006.
- [12] A.Galstyan, S.Kolar And K.Lerman "Resource Allocation Games With Changing Resource Capacities", *Proceedings Of The Second International Joint Conference On Autonomous Agents And Multi-Agent Systems, AAMAS '03*, 2003.
- [13] S.Seyednejad, M. Meybodi, "Evolutionary minority game based on genetic algorithm", *Seventh International Conference on Natural Computation, IEEE*, 2011.

1. Case Base Reasoning
2. Improved CBR-Learning Automata
3. Minority Game- Improved CBR-LA
4. Minority Game
5. Independent Case Base
6. Joined Case Base
7. Task Heterogeneity
8. Machine Heterogeneity
9. Consistency
10. Inconsistency
11. High-High
12. Low-Low
13. Reduction in Cost



شکل (9): میزان کاهش هزینه روش‌های پیشنهادی در مقابل MinMin

5- نتیجه‌گیری

شبیه‌سازی‌های انجام شده نشان می‌دهد که ایجاد رقابت اقلیتی بین وظایف سبب می‌شود تنها وظایف دسته اقلیت به اتوماتا مرتبط شود، در نتیجه اتوماتا مجبور به جستجوی همه فضای پاسخ نیست و فضایی را می‌گردد که احتمال پاسخ در آن بیشتر است، بدین ترتیب هزینه کمتری نسبت به سایر روش‌ها دارد. با توجه به جدول (2) و (3)، که کاهش هزینه الگوریتم MG-ICBR-LA را با پایگاه مورد مشترک و مستقل نشان می‌دهد، مشاهده می‌کنیم که هزینه (Cost) روش پیشنهادی در همه حالت‌ها نسبت به روش ICBR-LA کاهش یافته است، این روند کاهش هزینه در شکل (7) نشان داده شده است. همچنین در جدول‌های (4) و (5)، روش پیشنهادی با الگوریتم‌های استاندارد مقایسه شد. روند کاهش هزینه الگوریتم پیشنهادی در مقابل الگوریتم MaxMin در شکل (8) و نیز در مقابل الگوریتم MinMin در شکل (9) نشان داده شده است. با توجه به شکل (8)، الگوریتم پیشنهادی در همه حالت‌ها با هر دو متد، جواب قابل قبولی نسبت به روش MaxMin داشته است. در شکل (9) نیز، روش پیشنهادی جواب مناسبی در مقابل روش MinMin داشته است. در حالت کلی می‌توان گفت که روش پیشنهادی MG-ICBR-LA زمانی که از پایگاه مورد مشترک استفاده می‌کند نسبت به پایگاه مورد مستقل، در همه حالت‌ها جواب بهتری داشته است. این امر نشان می‌دهد که عامل‌های CBR زمانی که تجربه مشابهی در حل مساله داشته باشند، وقتی در دسته اقلیت نیز قرار گیرند، در زمان کمتری می‌توانند جواب مساله را پیدا کنند. در کارهای آینده قصد داریم با اعمال تغییرات بیشتر بر اتوماتاها و نوع نگاشت وظایف به منابع و همچنین ایجاد رقابت اقلیتی به همراه الگوریتم‌های تکاملی، هزینه تخصیص منابع را بیشتر کاهش دهیم.

مراجع

- [1] سعید صابمیری، بهروز معصومی، محمدرضا میبیدی، "بکارگیری سیاست کمیسیون چندعامله در تخصیص ایستای منابع گرید با استفاده از عامل‌های CBR"، سومین کنفرانس داده کاوی، آذر 1390.