

زمان‌بندی قواعد در پایگاه داده پویا به کمک اتوماتاهای یادگیر

محمد رضا میبدی

عباس رسولزادگان

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی امیرکبیر

دانشگاه صنعتی امیرکبیر

mmeybodi@aut.ac.ir

rasoolzadegan@yahoo.com

چکیده- سیستم مدیریت پایگاه داده پویا، امکان انجام واکنش‌های خودکار و مقتضی در هنگام وقوع رویدادهای مختلف را از طریق پردازش قواعد پویا فراهم می‌نماید. یکی از قسمت‌های مهم سیستم پایگاه داده پویا که تأثیر بسزایی در کارایی آن دارد، زمان‌بند قواعد است که در هر لحظه از بین قواعد آماده اجرا یکی را برای اجرا انتخاب می‌نماید. در این مقاله، یک روش مبتنی بر اتوماتاهای یادگیر به منظور بهبود زمان‌بندی قواعد بر حسب میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی و توان عملیاتی پیشنهاد می‌گردد. از اتوماتاهای یادگیر برای بهبود تخمین احتمال اجرای قواعد استفاده می‌گردد. روش پیشنهادی با بهترین روش زمان‌بند قواعد گزارش شده مقایسه شده است. نتایج آزمایشها نشان از برتری روش پیشنهادی دارد.

کلمات کلیدی- سیستم مدیریت پایگاه داده پویا، زمان‌بندی قواعد، تخمین احتمال اجرای قواعد، اتوماتاهای یادگیر

داده هیچ ابتکار عملی در هنگام رخدادن شرایط خاص در سیستم ندارد. بسیاری از برنامه‌های کاربردی مانند برنامه‌های انبارداری، اتوماسیون کارخانه‌جات و سیستم‌های دارای محاسبات پیچیده مالی (مانند بازار سهام) نیاز به نظارت خودکار دارند تا در صورت وقوع رویداد خاصی، واکنش مقتضی انجام شود. برای این منظور باید سیستم پایگاه داده‌ای

۱- مقدمه

پایگاه‌های داده متداول دارای ماهیت ایستا هستند، یعنی اعمال پرس و جو، بهنگام سازی، درج، حذف، گزارش‌گیری و غیره فقط زمانی که توسط کاربر درخواست شوند، انجام می‌پذیرند و سیستم مدیریت پایگاه

```

DEFINE LowRisk
ON Stock.UpdatePrice
IF (Stock.policy = Low_risk) and
    (Stock.price < Stock.initprice * e)
DO Stock.Buy

```

با فرض $e < 1$ ، قاعده‌ی پویای بالا تضمین می‌کند که هرگاه قیمت یک سهم بهنگام شود (بخش رویداد)، درصورتیکه، ارزش آن حداقل $100 \cdot (1 - e)$ درصد کاهش یافته و خط مشی استفاده شده Low_risk باشد (بخش شرط)، سیستم مدیریت پایگاه داده پویا این سهم را خریداری کند (بخش عمل).

در سیستم پایگاه داده پویا، در ابتدای امر، برنامه کاربردی در حال اجراست. تا زمانیکه هیچ رویداد از پیش تعریف شده‌ای در سیستم رخ نداده است اجرای برنامه کاربردی ادامه پیدا می‌کند. به محض اینکه رویدادی در سیستم رخ دهد، مجموعه‌ای از قواعد فعال است که با این رویداد، مرتبط هستند فعال شده و وارد لیست قواعد فعال می‌شوند. در ادامه با توجه به روش زمان‌بندی مورد استفاده، یکی از قواعد فعال انتخاب و بخش شرط آن ارزیابی می‌شود و در صورت درست بودن، به لیست قواعد آماده اجرا اضافه می‌شود. از بین مجموعه قواعد آماده اجرا نیز با توجه به روش زمان‌بندی مورد استفاده، یکی انتخاب و دستورات بخش عمل آن اجرا می‌شود. اگر اجرای این دستورات موجب رخ دادن رویدادهای دیگری در سیستم شود، قواعد مرتبط (قواعد فرزند) نیز به لیست قواعد فعال اضافه می‌شوند. ارزیابی و اجرای قواعد تا زمانی که لیست قواعد فعال و آماده اجرا خالی شوند، ادامه پیدا می‌کند. پس از آن کنترل اجرا به برنامه کاربردی باز می‌گردد. به مجموعه عملیات فوق، چرخه‌ی پردازش قواعد گویند [۵].

دو فاکتور بسیار مؤثر در نحوه اجرای چرخه‌ی پردازش قواعد، پیوستگی رویداد-شرط^۲ و پیوستگی شرط-عمل^۳ قواعد است. این دو پیوستگی دارای سه حالت مختلفاند: (۱) فوری، (۲) تعویقی و (۳) مستقل. اگر پیوستگی رویداد-شرط یک قاعده از نوع فوری باشد، شرط آن قاعده بلافصله پس از رخ دادن رویداد متناظر شرط و فعل شدن قاعده ارزیابی می‌شود. به همین ترتیب اگر پیوستگی شرط-عمل یک قاعده از نوع فوری باشد، پس از فعل شدن آن قاعده و ارزیابی شرط مربوطه، در صورت ثابت بودن نتیجه‌ی ارزیابی، دستورات بخش عمل آن قاعده فوراً اجرا می‌شوند. اگر پیوستگی رویداد-شرط (شرط-عمل) یک قاعده از نوع تعویقی باشد، ارزیابی شرط (اجرا عمل) آن قاعده تا زمان اتمام تراکنش جاری به تأخیر می‌افتد. در برخی موارد ارزیابی شرط (اجرا عمل) تا رخ دادن رویداد خاصی به تأخیر خواهد افتاد. در حالت مستقل، ارزیابی شرط (اجرا عمل) قاعده در قالب یک تراکنش مستقل و پس از اتمام تراکنش جاری انجام می‌شود [۵].

در سیستم مدیریت پایگاه داده پویا به فرایند تخصیص اولویت به قواعد فعل جهت ارزیابی بخش شرط و نیز به قواعد آماده اجرا جهت اجرای دستورات بخش عمل زمان‌بندی قواعد گویند. همانطور که پیش از این نیز بیان شد، قواعدی آماده اجرا هستند که اولاً اثر وقوع رویداد متناظرشان در سیستم، فعل شده باشند و ثانیاً بخش شرط‌شان در لحظه‌ی ارزیابی درست باشد. در این قسمت روش‌های مختلف زمان‌بندی قواعد را به طور اجمالی معرفی می‌نماییم [۳]، [۷]، [۸]. در شکل (۱) توصیف فرمال زمان‌بندی قواعد آماده اجرا در حالت کلی نشان داده

طراحی شود که در آن امکان تعریف رویدادهای مورد نظر و واکنش‌های متناظر آنها گنجانده شود، به چنین سیستمی، سیستم پایگاه داده پویا گویند [۵]، [۶]، [۷]، [۸]، [۱۱].

رفتار واکنشی سیستم پایگاه داده پویا از طریق پردازش قواعد پویا در قالب چرخه پردازش قواعد، سازمان‌دهی می‌شود. این چرخه شامل پنج مرحله است که عبارتند از: (۱) تشخیص رویدادها، (۲) فعال‌سازی قواعد مرتبط با رویدادهای رخ داده، (۳) زمان‌بندی قواعد فعل جهت ارزیابی بخش شرط آنها، (۴) زمان‌بندی قواعد آماده اجرا و (۵) اجرای دستورات بخش عمل قاعده‌ای که در مرحله‌ی پیشین برای اجرا انتخاب شده است. شایان ذکر است که مجموعه دستورات بخش عمل یک قاعده در قالب یک تراکنش اجرا می‌شوند [۷]، [۸]، [۱۱]، [۱۸].

روش مورد استفاده برای زمان‌بندی قواعد به طور مستقیم در فاکتورهایی مانند زمان پاسخگویی سیستم به تراکنش‌ها، زمان بازگشت تراکنش‌ها، توان عملیاتی سیستم و بطور کلی در کارایی سیستم پایگاه داده پویا، بسیار مؤثر است. بهبود زمان‌بندی قواعد در سیستم مدیریت پایگاه داده پویا یکی از موضوعات تحقیقاتی مهم است [۷]، [۱۱]، [۱۲]، [۱۳]، [۱۴]، [۱۵]، [۱۶]، [۱۷]، [۱۸]، [۲]، [۶]، [۱۱] که در بخش دوم مورد بررسی قرار می‌گیرند. پیش از این روش‌های زمان‌بندی موجود را در یک محیط آزمایشگاهی مبتنی بر معماری سه بخشی (شبیه‌ساز پایگاه داده پویا) براساس تعدادی پارامتر ارزیابی با یکدیگر مقایسه نموده و روش کاراتر کردن اوتوماتی یادگیر به روش Ex-SJF_{PRO}-V.2.8 را معرفی کردند [۱]، [۲]. در این مقاله با اضافه تخمین احتمال اجرای قواعد و متعاقباً ارائه یک روش زمان‌بندی کاراتر از بعد پارامترهای ارزیابی تعریف شده، برآمدۀایم، همچنین با افزودن ویژگی‌هایی به شبیه‌ساز پایگاه داده پویا، سعی کرده‌ایم تا حد امکان به شبیه‌سازی و ارزیابی دقیق‌تر رفتار پایگاه داده پویا و عملکرد روش‌های زمان‌بندی بپردازیم.

این مقاله در پنج بخش تنظیم شده است، در بخش دوم به معرفی روش‌های موجود برای زمان‌بندی قواعد در پایگاه داده پویا می‌پردازیم، در بخش سوم با هدف بهبود زمان‌بندی قواعد بر مبنای بهبود فرایند تخمین احتمال اجرای قواعد به کمک اوتوماتی یادگیر، روشی جدید معرفی می‌نماییم. در بخش چهارم به بررسی عملکرد روش پیشنهادی جدید در مقایسه با روش‌های زمان‌بندی موجود بر اساس تعدادی پارامتر ارزیابی بر مبنای نتایج حاصل از انجام آزمایشات متعدد و متنوع در یک محیط آزمایشگاهی مبتنی بر معماری سه بخشی می‌پردازیم، سرانجام در بخش پنجم مطالب مطرح شده در مقاله را جمع بندی می‌نماییم.

۲- معرفی روش‌های مختلف زمان‌بندی قواعد

در حالت کلی، قواعد پویا شامل سه بخش به نامهای رویداد، شرط و عمل هستند. به همین دلیل به آنها قواعد رویداد-شرط-عمل یا به اختصار ECA^۴ گفته می‌شود [۵]، [۸]. در این قسمت، به عنوان مثال، قاعده پویایی را که برای یک سیستم خرید و فروش سهام تعریف شده است، نشان می‌دهیم. این قاعده براساس استاندارد زبان توصیف قاعده SAMOS^۵ است:

۴-۲- روش مبتنی بر الگوریتم نزدیکترین ضرب العجل^۸

این روش یکی از کامل‌ترین روش‌هایی است که تاکنون برای زمان-بندی قواعد در سیستم‌های پایگاه داده پویای بلادرنگ^۹ ارایه شده است و به اختصار EDF نامیده می‌شود. در این روش، هنگام انتخاب، قاعده‌ای که دارای نزدیکترین ضرب‌العجل باشد، برای اجرا انتخاب می‌شود. براساس نحوه محاسبه اولویت قواعد که مبتنی بر ضرب‌العجل قواعد است، سه نسخه مختلف از روش EDF طراحی شده است که عبارتند از: EDF_{EDF_{SL}}، EDF_{EDF_{DIV}} و EDF_{PD}. اولی در حقیقت همان الگوریتم صرف است. در دومی یک تغییر کوچک در الگوریتم EDF داده شده است به این ترتیب که اولویت قاعده‌ی پدر و قوتی یک قاعده‌ی فرزند تولید می‌شود، تغییر می‌کند. در سومی اولویت قاعده‌ی پدر در هر لحظه براساس سه پارامتر تاخین زمان اجرای قاعده، تعداد قواعد فرزند فعلی و تخمین تعداد قواعد فرزند که در آینده تولید خواهد شد، محاسبه می‌شود [۵، ۸]. با جایگزین کردن «شبه کد» زیر بجای سطر ۵ شکل (۱) توصیف فرمال روش زمانبندی مبتنی بر الگوریتم نزدیکترین ضرب‌العجل، در حالت کلی بدست می‌آید.

```
5') Activate(R) ≡ {Create Instance of R and Set_DeadLine(R)}
Rule_Scheduling (ReadyToExecute_Rule_Base) ≡
  ∀ Ri ∈ ReadyToExecute_Rule_Base, [(Ri, Di) ... (Rn, Dn)],
  Ri Selected | {Ri ∈ ReadyToExecute_Rule_Base | 1 ≤ i ≤ n}
  / ∀ Rj ∈ ReadyToExecute_Rule_Base | 1 ≤ j ≤ n, Di ≤ Dj}}}
```

شده است. مطالبی که در ادامه در مورد زمانبندی قواعد آمده‌ی اجرا بیان می‌شود، در مورد زمانبندی قواعد فعال نیز صادق است و بالعکس.

```
1) Rule_Base ≡ {Set of ECA Rules}
2) Active_Rule_Base ≡ {Sub Set of Rule_Base}
3) ReadyToExecute_Rule_Base ≡ {Sub Set of Active_Rule_Base}
4) n = {i ∈ N | i = Number of ECA Rules in the
ReadyToExecute_Rule_Base}
5) Rule_Scheduling (ReadyToExecute_Rule_Base)
```

شکل (۱): توصیف فرمال زمانبندی اجرای قواعد در حالت کلی

۴-۲- روش اتفاقی^۵

در این روش هرگاه که در چرخه‌ی پردازش قواعد نیاز به انتخاب قاعده جدیدی برای اجرا باشد، از بین قواعد آمده‌ی اجرا به صورت اتفاقی یکی برای اجرا انتخاب می‌شود [۳، ۷، ۸]. با جایگزین کردن «شبه کد» زیر بجای سطر ۵ شکل (۱) توصیف فرمال روش زمانبندی اتفاقی قواعد بدست می‌آید.

```
5' Random_Generator (n) ≡ {i ∈
  | 1 ≤ i ≤ n and i selected randomly}
Rule_Scheduling (ReadyToExecute_Rule_Base) ≡
  { ∀ Ri ∈ ReadyToExecute_Rule_Base, (Ri, ..., Rn)
  | R = ReadyToExecute_Rule_Base (RRandom_Generator(n))}
```

۶ -۲-۲

۴-۳- روش مبتنی بر الگوریتم ابتدا کوچکترین کار^{۱۰}

این روش که مبتنی بر الگوریتم ابتدا کوچکترین کار (SJF) است، به اختصار E_X-SJF نامیده می‌شود. SJF که یکی از روش‌های کلاسیک زمان‌بندی است بر اصل ساده‌ی "هر تراکنشی که دارای زمان اجرای کوتاهتری است، زودتر باید اجرا شود"، استوار است. استفاده از روش زمان‌بندی SJF در سیستم‌های متداول پردازش تراکنش موجب کاهش میانگین زمان پاسخگویی و کاهش میانگین زمان بازگشت خواهد شد. اما این روش کلاسیک به همین صورت قابل استفاده در سیستم‌هایی که دارای خاصیت تولید تراکنش بیوای استند، (مانند سیستم‌های پایگاه داده بیوای) نیست و استفاده از آن موجب بهبود زمان‌بندی خواهد شد [۸، ۹]. با توجه به اینکه روش جدیدی که در بخش سوم ارائه خواهد شد، E_X-SJF را نسبت به سایر روش‌های موجود با تفصیل بیشتری شرح می-دهیم. با جایگزینی «شبه کد» زیر بجای سطر ۵ شکل (۱) توصیف فرمال روش زمانبندی E_X-SJF در حالت کلی بدست می‌آید.

```
5') Rule_Scheduling (ReadyToExecute_Rule_Base) ≡
  { ∀ Ri ∈ ReadyToExecute_Rule_Base,
  (Ri, ExecTi) ... (Rn, ExecTn), Ri Selected |
  {Ri ∈ ReadyToExecute_Rule_Base | 1 ≤ i ≤ n} { ∀ Rj ∈
  ReadyToExecute_Rule_Base | 1 ≤ j ≤ n, ExecTi ≤ ExecTj}}}
```

در روش E_X-SJF هرگاه که نیاز به انتخاب یکی از تراکنش‌های

در این روش به هر یک از قواعد یک عدد صحیح به عنوان اولویت نسبت داده می‌شود. سپس در هنگام انتخاب قواعد برای اجرا، قاعده‌ای که در بین قواعد آمده‌ی اجرا کوچکترین اولویت را دارد، انتخاب می‌شود [۳، ۷، ۸]. با جایگزین کردن «شبه کد» زیر بجای سطر ۵ شکل (۱) توصیف فرمال روش زمانبندی اولویت ایستا بدست می‌آید.

```
5' Rule_Scheduling (ReadyToExecute_Rule_Base) ≡
  { ∀ Ri ∈ ReadyToExecute_Rule_Base, [(Ri, Pi), ..., (Rn, Pn)],
  Ri Selected | {Ri ∈ ReadyToExecute_Rule_Base | 1 ≤ i ≤ n}
  / ∀ Rj ∈ ReadyToExecute_Rule_Base | 1 ≤ j ≤ n, Pi ≤ Pj}}}
```

۴-۳- روش مبتنی بر برچسب زمانی^۷

در این روش به هر قاعده که در حالت فعال قرار بگیرد یک برچسب زمانی که مشخص کننده زمان فعال شدن آن قاعده است، تخصیص می-یابد. در هنگام انتخاب، قاعده‌ای که دارای کوچکترین برچسب زمانی باشد، برای اجرا انتخاب خواهد شد [۳، ۷، ۸]. با جایگزین کردن «شبه کد» زیر بجای سطر ۵ شکل (۱) توصیف فرمال روش زمانبندی مبتنی بر برچسب زمانی بدست می‌آید.

```
5' Set_TimeStamp(R) ≡ {R's TimeStamp = Current Time}
Activate(R) ≡ {Create Instance of R and Set_TimeStamp(R)}
Rule_Scheduling (ReadyToExecute_Rule_Base) ≡ { ∀ Ri ∈
ReadyToExecute_Rule_Base, [(Ri, Ti) ... (Rn, Tn)], Ri Selected
{Ri ∈ ReadyToExecute_Rule_Base | 1 ≤ i ≤ n}
/ ∀ Rj ∈ ReadyToExecute_Rule_Base | 1 ≤ j ≤ n, Ti ≤ Tj}}
```

$X^{imm}(R_i)$	زمان اجرای زامین تراکنش فوری حاصل از
$X^{def}(R_j)$	زمان اجرای زامین تراکنش تعویقی حاصل از
$L(R)$	تعداد دستورات بخش عمل R
$n^{def}(R)$	تعداد قواعد تعویقی تولیدی توسط قاعده‌ی R
$n^{imm}(R)$	تعداد قواعد فوری تولیدی توسط قاعده‌ی R
$X(R)$	زمان اجرای بخش عمل قاعده‌ی R
$P(R_i)$	احتمال درستی شرط زامین قاعده فوری حاصل از
$P(R_j)$	احتمال درستی شرط زامین قاعده تعویقی حاصل از

در (۲) فرض شده است که اجرای هر دستور در بخش عمل قواعد، واحد زمانی به طول می‌انجامد. لذا زمان اجرای هر قاعده به تنها یک واحد شد.

$$L(R)$$

روش زمان‌بندی $\text{Ex-SJF}_{\text{PRO}}$

در این روش احتمال درستی هر یک از عبارات شرطی موجود در بخش شرط یک قاعده به صورت یکسان $\frac{1}{2}$ در نظر گرفته شده است. بنابراین احتمال درستی شرط قاعده R که در بخش پیشین بیان شد، طبق (۱) خواهد شد:

$$P(R) = P[(A \cap B) \cup (C \cap D)] = \frac{1}{2} * \frac{1}{2} + \frac{1}{2} * \frac{1}{2} - \frac{1}{2} * \frac{1}{2} * \frac{1}{2} = \frac{7}{16}$$

با توجه به مطالع فوق، زمان اجرای هر قاعده در این روش نیز از (۲) قابل محاسبه است [۴، [۶، [۸].

روش زمان‌بندی $\text{Ex-SJF}_{\text{PRO}}\text{-V.1.8}$

در این روش که پیش از این توسط خودمان ارائه شده است، یک پیمانه تخمین به روش $\text{Ex-SJF}_{\text{PRO}}$ اضافه کرده‌ایم [۴، [۵]. بدین ترتیب که در ابتدا زمان اجرای هر قاعده، همانند روش $\text{Ex-SJF}_{\text{PRO}}$ محاسبه می‌شود. سپس با شروع کار سیستم هر بار که بخش شرط یکی از قواعد فعل مورد ارزیابی قرار گیرد، احتمال درستی هر یک از عبارات منطقی (LS_i) موجود در بخش شرط آن قاعده یعنی $P(R, LS_i)$ طبق (۳) محاسبه و ذخیره می‌شود.

$$P(R, LS_i) = \frac{\text{تعداد دفعاتی که بخش شرط } R \text{ تاکنون ارزیابی شده است}}{\text{تعداد دفعاتی که شرط } LS_i \text{ درست بوده است}} \quad (3)$$

این کار برای هر عبارت منطقی در بخش شرط یک قاعده تا زمانی تکرار می‌شود که آهنگ تغییرات احتمال درستی آن عبارت منطقی به حد مطلوب برسد (مثلاً 0.0001 و در حالت کلی)، در این هنگام مقدار جدید، جایگزین مقدار پیش فرض اولیه ($\frac{1}{2}$) می‌شود. زمانیکه احتمال درستی همه عبارات منطقی یک شرط بهنگام شد، احتمال درستی آن شرط (احتمال اجرای قاعده‌ی صاحب آن شرط) نیز بهنگام می‌شود. و در نهایت زمان اجرای قاعده R زمانی طبق (۲) بهنگام می‌شود که زمان اجرای همه عبارات منطقی شده باشند. بدین ترتیب بعد از مدتی که از اجرای سیستم بگذرد (که البته این مدت نسبت به کل زمانی که قرار است سیستم کار کند ناچیز است)، زمان اجرای قواعد با دقت بیشتری نسبت به زمان شروع اجرای سیستم، محاسبه می‌شود [۵]. با جایگزینی «شبیه کد» شکل (۲) در سطر ۵ شکل (۱) توصیف فرمال روش زمان‌بندی $\text{Ex-SJF}_{\text{PRO}}\text{-V.1.8}$ به دست می‌آید.

فعال در سیستم باشد، تراکنشی که دارای کمترین زمان اجراست، انتخاب می‌شود. تفاوت دو روش Ex-SJF و SJF در نحوه محاسبه زمان اجرای تراکنش‌ها است. در روش Ex-SJF به دلیل تولید پویای تراکنش‌ها، نمی‌توان فاکتور انتخاب آنها را فقط زمان اجرای خود تراکنش‌ها قرار داد. زیرا هر تراکنشی (پدر) که در یک سیستم پایگاه داده پویا اجرای می‌شود به صورت بالقوه مجموعه‌ای از قواعد پویا (فرزنده) را در سیستم فعال می‌کند. تراکنش‌های حاصل از قواعد نیز در صورت اجرا، مجموعه‌ای دیگر از قواعد را به صورت بالقوه فعال خواهند کرد و این عمل تا چند سطح ممکن است انجام پذیرد. لذا زمان واقعی اجرای یک تراکنش، علاوه بر زمان اجرای خودش، شامل زمان اجرای تراکنش‌های فوری و تعویقی که به صورت بالقوه در اثر اجرای آن تراکنش تولید و اجرا خواهند شد، نیز می‌باشد. محاسبه زمان واقعی اجرای قواعد در زمان اجرا ممکن‌پذیر است. اما انجام آن به دلیل در پی داشتن سربار زمانی زیاد، منجر به یک روش زمان‌بندی ناکارا خواهد شد. لذا تمام نسخه‌های روش Ex-SJF زمان اجرای قواعد را پیش از اجرا محاسبه می‌نمایند. اما محاسبه زمان واقعی اجرای یک قاعده پیش از زمان اجرا مستلزم تخمین احتمال اجرای قواعدی است که در دل آن قاعده در زمان اجرا به صورت پویا فعال می‌شوند. هرچه این تخمین دقیق‌تر باشد، مقدار محاسبه شده برای زمان اجرای قاعده به واقعیت نزدیکتر است [۳، [۵، [۸].

همانطور که پیش از این بیان شد، اجرای یک قاعده‌ی فعال منوط به درستی بخش شرط آن در لحظه‌ی ارزیابی است. به عبارت دیگر احتمال اجرای یک قاعده با احتمال درستی بخش شرط آن یکسان است. بخش شرط موجود در بدنه قواعد از عبارات شرطی، عبارات پرس و جوی پایگاه داده، فرخوانی رویه‌ها و توابع و ترکیب منطقی آنها تشکیل شده است. فرض کنید قاعده R وجود دارد به طوریکه بخش شرط آن به صورت $P[(A \cap B) \cup (C \cap D)]$ بیان شده است. A و C و B و D عبارات منطقی هستند. در نتیجه احتمال درستی بخش شرط قاعده R به صورت زیر محاسبه می‌شود:

$$P[(A \cap B) \cup (C \cap D)] = P(A \cap B) + P(C \cap D) - P(A \cap B \cap C \cap D)$$

با فرض مستقل بودن عبارات A و C و B و D از هم داریم:

$$P[(A \cap B) \cup (C \cap D)] = P(A) * P(B) + P(C) * P(D) - P(A) * P(B) * P(C) * P(D) \quad (1)$$

اگر احتمال درست بودن عبارات شرطی A و C و B و D به صورت قطعی وجود داشته باشد، می‌توان با جانشینی کردن مقادیر آنها در (۱)، احتمال درست بودن شرط قاعده R را به دقت محاسبه نمود. ولی احتمال درستی یک عبارت شرطی غالباً قبل از اجرای آن به صورت قطعی وجود ندارد، هر چهار نسخه $\text{Ex-SJF}_{\text{EXA}}$ $\text{Ex-SJF}_{\text{PRO}}$ $\text{Ex-SJF}_{\text{PRO-V.2.8}}$ $\text{Ex-SJF}_{\text{PRO-V.1.8}}$ فرزند یک قاعده به تشکیل درخت اجرای قواعد می‌پردازند [۵]. اما وجه تمایز این نسخه‌ها در نحوه تخمین احتمال اجرای قواعد فرزند می‌باشد.

روش زمان‌بندی $\text{Ex-SJF}_{\text{EXA}}$

در این روش احتمال درستی بخش شرط قواعد (احتمال اجرای قواعد)، ۱ در نظر گرفته می‌شود [۸]. در نتیجه زمان اجرای قواعد با فرض $P(R) = 1$ طبق (۲) محاسبه می‌شود.

$$X(R) = L(R) + \sum_{i=1}^{n^{imm}(R)} P(R_i) * X^{imm}(R_i) + \sum_{j=1}^{n^{def}(R)} P(R_j) * X^{def}(R_j) \quad (2)$$

جدول (۱): بخش شرط چند قاعدهی نمونه

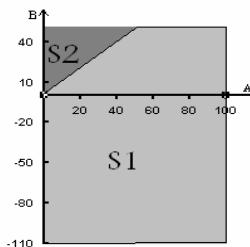
نام	بخش شرط قاعده
R_1	$DI_A > DI_B$
R_2	$DI_C = \{ \text{فولاد, مس, سیمان, برنج, طلا, برنز, فرش, نقره, پرتال, کفش} \}$ OR مس
R_3	$DI_B = 20 \text{ AND } DI_D \leq DI_A$
R_4	$DI_A < 20 \text{ AND } DI_C = \{ \text{برنج} \}$

جدول (۲): دامنه متغیرهای شرطی قواعد جدول (۱)

$0 < DI_A < 100$	$-110 < DI_B < 50$	$-2000 < DI_D < 50$
$DI_C = \{ \text{فولاد, مس, سیمان, برنج, طلا, برنز, فرش, نقره, پرتال, کفش} \}$		

با توجه به اینکه پیش از زمان اجرا هم بخش شرط کلیه قواعد مشخص است و هم دامنه مقادیر مجاز برای کلیه متغیرهای شرطی موجود در بخش شرط قواعد و با فرض یکنواختی توزیع احتمال متغیرهای شرطی و مستقل بودن عبارات منطقی از هم، می‌توان احتمال درستی هر یک از عبارات منطقی را پیش از زمان اجرا، به صورت ریاضی محاسبه نمود. فرض یکنواختی توزیع احتمال یک متغیر شرطی به این معناست که احتمال وقوع تمامی مقادیر موجود در دامنه آن متغیر یکسان است.

در اینجا قصد داریم به عنوان نمونه احتمال درستی شرط قاعده R_1 را با فرضیات و معلومات فوق محاسبه نماییم. همانطور که در شکل (۳) نشان داده شده است، در هر نقطه داخل ذوزنقه $S1$ قلم داده A از قلم B بزرگتر است و در هر نقطه داخل مثلث $S2$ قلم داده A از قلم داده B کوچکتر است. لذا احتمال اینکه شرط $DI_A > DI_B$ درست باشد، با فرض اینکه مساحت ذوزنقه $S1$ برابر S_{S1} و مساحت مثلث $S2$ برابر S_{S2} باشد، طبق (۴) محاسبه می‌شود:



$$P(R_1) = \frac{S_{S1}}{S_{S1} + S_{S2}} \quad (4)$$

که با مقدارگذاری در (۴) خواهیم داشت:

$$P(R_1) = \frac{(110 * 100) + ((100 + 50) * 25)}{100 * 160} = \frac{14750}{16000} = 0.92$$

شکل (۳): محاسبه احتمال درستی شرط قاعده R_1

به همین ترتیب احتمال درستی شرط تمامی قواعد (احتمال اجرای بخش عمل تمامی قواعد) پیش از زمان اجرا قابل محاسبه است. پس از محاسبه احتمال اجرای بخش عمل قواعد، زمان اجرای قواعد را طبق (۲) محاسبه می‌نماییم. حال با در اختیار داشتن زمان اجرای قواعد، سیستم

When Active Rule R is selected among Active Rules for condition evaluation by Scheduling Approach, LS_Probability() is called for all conditional Statement of condition part of Rule R

```

LS_Probability(LS)
{ Counter1 = Counter1+1
IF Check_Correctness (LS) Then
    Counter2 = Counter2+1
LS → New_Probability = counter2/counter1
IF -E <= LS → New_Probability - LS → Old_Probability <= E
    Then LS → fixed=True
    Condition_Probability(R → C)
LS → Old_Probability = LS → New_Probability

```

```

Condition_Probability(R → C)
{ For Each LS of R → C That LS → fixed=False Do
    LS_Probability(R → C → LS)
    IF R → C → LS → Fixed=False Then
        R → C → Fixed=False
        Return(R → C → Fixed)
    R → C → Fixed=True
    Calc_Execution_Time(R → parents)
    Return(Calculate_Probability(R → C)) }

```

"At first R → visited = False for all Rules"

```

Calc_Execution_Time(R)
{ IF R = NULL Then Return (0)
For Each Child of R DO
    IF R → Child → C → Fixed=True
        Then R → Fixed=True
    IF (R → Fixed=True) or (R → visited = False) Then
        {I = 0
        For Each Child of R DO
            I = I + Calc_Execution_Time (R → Child)
        R → visited = True
        R → ExecutionTime =
        R → ExecutionTime * Condition_Probability(R → C) + I;
        Return (R → ExecutionTime); }

```

```

Rule_Scheduling (ReadyToExecute_Rule_Base) ≡
{ ∀ R ∈ ReadyToExecute_Rule_Base,
  {(R, ExecT1)... (R, ExecTn)}, R, Selected | /R ∈ ReadyToExecute_Rule_Base | 1 ≤ i ≤ n
  ∀ Rj ∈ ReadyToExecute_Rule_Base | 1 ≤ j ≤ n,
  Ri → ExecTime ≤ Rj → ExecTime.}

```

شکل (۲): شبکه محاسبه زمان اجرای واقعی قواعد و نحوه انتخاب آنها

$$E_x - SJF_{PRO} - V.1.8$$

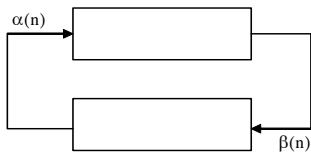
روش زمان‌بندی EX-SJF_{PRO}-V.2.8

این روش نیز مانند روش EX-SJF_{PRO}-V.1.8 بر مبنای تخمین احتمال اجرای قواعد طراحی و پیاده‌سازی شده است [۱، [۲، [۳، [۴، [۵، [۶]. تفاوت این دو روش در نحوه تخمین احتمال اجرای قواعد می‌باشد. در این قسمت نحوه عملکرد روش جدید را با ذکر یک مثال شرح می‌دهیم. همانطور که پیش از این بیان شد، بخش شرط هر قاعده ترکیبی از عبارات منطقی است. جدول (۱) بخش شرط چند قاعده و جدول (۲) دامنه متغیرهای شرطی بخش شرط این قواعد را نشان می‌دهد. به عنوان مثال بخش شرط قاعده R_1 در صورتی درست است که قلم داده (DI_A) از قلم داده (DI_B) بزرگتر باشد.

در روش پیشنهادی جدید مکانیزم زمانبندی قواعد فعال و قواعد آمده‌ی اجرا با هم متفاوتند. بدین ترتیب که برای زمانبندی قواعد آمده‌ی اجرا همانند روش $E_{X-SJF_{PRO}-V.2.8}$ عمل می‌نماید. اما برای زمانبندی قواعد فعال، دارای مکانیزم زمانبندی متفاوتی بر مبنای اتوماتای یادگیر است. عبارت دیگر در روش جدید، ملاک زمانبندی قواعد فعال، بر خلاف روش $E_{X-SJF_{PRO}-V.2.8}$ ، زمان اجرای قواعد نیست. بلکه احتمال درستی بخش شرط آنهاست و اتوماتای یادگیر نیز در این راستا به تخمین احتمال درستی بخش شرط قواعد، در حین اجرای سیستم می‌پردازد. اما در ادامه پیش از تشریح جایگاه و عملکرد اتوماتای یادگیر در زمانبندی قواعد فعال، به تعریف اتوماتای یادگیر و مفاهیم مرتبط می‌پردازیم.

۱-۳- اتوماتاهای یادگیر

اتوماتای یادگیر [۱۰]، مانندی است که می‌تواند تعدادی متناهی عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می‌شود و نتیجه ارزیابی در قالب پاسخی مثبت یا منفی به اتوماتا داده می‌شود و اتوماتا از این پاسخ در انتخاب عمل بعدی تأثیر می‌گیرد. هدف نهایی این است که اتوماتا یاد بگیرد تا از بین اعمال خود



شکل (۴): ارتباط بین محیط و اتوماتای یادگیر

بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداقل برساند. کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل (۴) مشاهده می‌شود.

محیط را می‌توان توسط سه‌تایی $E \equiv \{v, \beta, c\}$ نشان داد که در آن $\{\alpha_r, \alpha_1, \alpha_2, \dots, \alpha_m\} \equiv v$ مجموعه ورودی‌ها، $\{\beta_1, \beta_2, \dots, \beta_m\} \equiv \beta$ مجموعه احتمال‌های جریمه مجموعه خروجی‌ها و $\{c_1, c_2, \dots, c_m\} \equiv c$ مجموعه درصد افزایش کاربردی. 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 به طور گستته یک مقدار از مقادیر محدود در فاصله $[0, 1]$ و در محیط از نوع K ، $\beta(n)$ مقدار از مقادیر محدود در فاصله $[0, 1]$ است. c_i احتمال این است که عمل α_i متغیر تصادفی در فاصله $[0, 1]$ باشد. احتمال این است که عمل α_i نتیجه نامطلوب U^* داشته باشد. در محیط ایستا^{۱۵} مقادیر c_i بدون تغییر می‌مانند، حال آنکه در محیط غیرایستا^{۱۶} این مقادیر در طی زمان تغییر می‌کنند.

اتوماتای یادگیر با ساختار ثابت توسط پنج‌تایی $\{v, \beta, F, G, \phi\}$ نشان داده می‌شود که در آن $v \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عمل‌های اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودی‌های اتوماتا، $\phi \equiv \{\phi_1, \phi_2, \dots, \phi_s\}$ مجموعه وضعیت‌های داخلی اتوماتا، $F: \phi \times \beta \rightarrow \phi$ تابع تولید وضعیت جدید اتوماتا و $G: \phi \rightarrow \phi$ تابع خروجی می‌باشد که وضعیت کنونی اتوماتا را به خروجی بعدی می‌نگارد.

می‌تواند کار خود را آغاز کند و زمان‌بند نیز در هر لحظه از بین قواعد فعالی که نتیجه‌ی ارزیابی بخش شرط‌شان مثبت بوده، قاعده‌های را که زمان اجرای کمتری دارد، برای اجرای بخش دستورات انتخاب می‌نماید. اما همانطور که پیش از این بیان کردیم، در محاسبه زمان اجرای قواعد به روش فوق، توزیع احتمال متغیرهای شرطی به صورت یکنواخت فرض شده است. این در حالی است که در سیستم‌های واقعی ممکن است توزیع احتمال متغیری یکنواخت نباشد. لذا برای حذف فرض فوق و محاسبه هرچه دقیق‌تر زمان اجرای قواعد به روش زیر عمل می‌نماییم. پس از شروع کار سیستم، در بازه‌های زمانی مشخص (δt)، برای هر یک از متغیرهای شرطی بخش شرط قواعد، تمامی مقادیری که اخذ شده، به همراه مدت زمان اخذ آنها ثبت می‌گردد. در پایان هر δt بر اساس اطلاعات بدست آمده توزیع احتمال متغیرها اصلاح و در نتیجه احتمال و زمان اجرای قواعد مجدد محاسبه و بهنگام می‌شوند. بدین ترتیب که مقادیر فعلی احتمال وقوع مقادیر دامنه‌ی متغیرها را براساس مقادیری که در δt گذشته توسط آن متغیرها اخذ شده و همچنین با توجه به مدت زمان اخذ هر مقدار توسط هر متغیر، بهنگام و سپس زمان اجرای قواعد را براساس احتمالات بهنگام شده، مجدد، طبق (۲) محاسبه می‌نماییم. جدول (۳) مطلب فوق را در رابطه با قلمداده DIA نشان می‌دهد. محاسبات و بهنگام سازی‌ها را در زمان بیکاری سیستم انجام می‌دهیم. به عبارت دیگر اندازه‌ی δt را طوری تعیین می‌کنیم که پایان آن با زمان بیکاری سیستم مصادف شود. بدیهی است هرچه از شروع کار سیستم بیشتر بگذرد، توزیع‌های احتمالی که برای متغیرها بدست می‌آیند به واقعیت نزدیکتر شده و درنتیجه زمان اجرای قواعد دقیق‌تر محاسبه می‌شوند. اصلاح توزیع‌های احتمال متغیرها را تا زمانی تکرار می‌کنیم، که آنچه تغییرات زمان اجرای قواعد به حد مطلوب برسد (۴). بدیهی است که تعیین اندازه‌ی δt به میزان اهمیت دقت زمان اجرای قواعد در سیستم بستگی دارد.

۳- معرفی روش جدید زمانبندی $E_{X-SJF_{ESTLA}}$

این روش نیز مانند روش‌های $E_{X-SJF_{PRO}}$ و $E_{X-SJF_{PRO}-V.1.8}$ بر مبنای تخمین احتمال اجرای قواعد طراحی شده است. عبارت دیگر در روش جدید با اضافه کردن یک اتوماتای یادگیر به کاربردی روش موجود [۱۱]، درصد افزایش کارایی فرایند زمانبندی قواعد از بُعد پارامترهای ارزیابی تعریف شده، برآمده‌ایم. همانطور که پیش از این نیز بیان شد، فرایند پردازش قواعد پویا، مشتمل بر دو زمانبندی است که عبارتند از: ۱- زمانبندی قواعد فعال جهت انتخاب یکی از قواعد فعال به منظور ارزیابی بخش شرط آنها و ۲- زمانبندی قواعد آمده‌ی اجرا جهت انتخاب یکی از قواعد آمده‌ی اجرا به منظور اجرای دستورات بخش عمل آنها. روش‌هایی که تاکنون ارائه شده‌اند، همگی دارای مکانیزم یکسانی برای هر دو زمانبندی مذکور می‌باشند. بنابراین مثال در روش $E_{X-SJF_{PRO}-V.2.8}$ ، هر دو زمانبند قواعد فعال و آمده‌ی اجرا بر مبنای زمان اجرای قواعد عمل می‌کنند. بدین ترتیب که ملاک انتخاب یک قاعده فعال یا آمده‌ی اجرا در هر لحظه، زمان اجرای آن قاعده است و لذا قاعده‌ای انتخاب می‌شود که کمترین زمان اجرای داشته باشد.

جدول (۳): نحوه‌ی ثبت مقادیر اخذ شده توسط قلم داده‌ی DI_A به همراه مدت زمان اخذ آنها در حین اجرای سیستم

δt_n (۱۰۰۰ دقیقه)		...	δt_2 (۱۰۰۰ دقیقه)		δt_1 (۱۰۰۰ دقیقه)		بازه‌های زمانی اقلام داده
مدت زمان اخذ (برحسب دقیقه)	مدار اخذ شده	...	مدت زمان اخذ (برحسب دقیقه)	مدار اخذ شده	مدت زمان اخذ (برحسب دقیقه)	مدار اخذ شده	
۱۰۲	۲۰	...	۸	۲۳	۱۰	۲	DI_A
۵۰	۷۱	...	۲۸	۴۷	۵	۶۷	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
۳۵۲	۱۱	...	۹۵	۸۶	۳۹	۲۲	
۵۲	۱	...	۵	۴	۱۵	۱۴	
...	

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

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

که r تعداد اقدامات اتومات، a پارامتر پاداش و b پارامتر جرمیه می‌باشند. لذا شکل عمومی الگوریتم یادگیری بصورت زیر است. اگر در گام n ام اقدام α_i انتخاب شده باشد، سپس در گام $n+1$ خواهیم داشت:

الف- پاسخ مطلوب از محیط

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

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

ب- پاسخ نامطلوب از محیط

$$p_i(n+1) = (1-b)p_i(n)$$

$$p_j(n+1) = \frac{b}{r-1} + (1-b)p_j(n) \quad \forall j, j \neq i$$

با توجه به مقادیر a و b در روابط فوق، سه حالت را می‌توان در نظر گرفت. اگر مقادیر a و b برابر باشند، اتومات اتومات یادگیر L_{RP} نامیده می‌شود. زمانیکه b مساوی با صفر باشد اتومات یادگیر L_{RI} ^{۱۷} نامیده می‌شود. اگر $a < b$ باشد، اتومات یادگیر L_{REP} ^{۱۸} نامیده می‌شود.

۲-۳- نحوه‌ی بکارگیری اتومات اتومات یادگیر در فرایند تخمین احتمال اجرای قواعد

در روش جدید به کمک یک اتومات اتومات یادگیر تصادفی^{۱۹} با ساختار متغیر و الگوریتم یادگیری L_{RI} به زمانبندی قواعد فعال بر مبنای احتمال درستی شرط آنها می‌پردازیم. وظیفه این اتومات اتومات یادگیر این است که با توجه به پاسخ‌های دریافتی از محیط، استراتژی زمانبند قواعد را برای انتخاب اقدام مناسب بهنگام نماید تا نهایتاً به افزایش کارایی فرایند زمانبندی قواعد پویا از بعد پارامترهای ارزیابی تعریف شده، منجر گردد. جایگاه اتومات اتومات یادگیر مذکور در معماری سیستم پایگاه داده پویا در شکل (۶) نشان داده شده است.

محیطی که اتومات اتومات یادگیر مذکور در آن واقع است، از نوع مدل P است، یعنی پاسخی که از طرف محیط به اقدام LA داده می‌شود، دودویی است (در صورتیکه شرط قاعده‌ی فعل انتخاب شده، درست باشد، پاسخ محیط ۱ و در صورتیکه نادرست باشد، پاسخ محیط ۰ خواهد بود). اقدام اتوماتی مذکور در هر لحظه، انتخاب یکی از قواعد فعل برای ارزیابی بخش شرط آن است. با توجه به متغیر بودن تعداد قواعد فعل در هر

اتوماتی یادگیر با ساختار متغیر را می‌توان توسط چهارتایی $\{v, \beta, p, T\}$ نشان داد که $v = \{\alpha_1, \dots, \alpha_r\}$ مجموعه عملهای اتومات، $\beta = \{\beta_1, \dots, \beta_m\}$ مجموعه ورودی‌های اتومات، $p = \{p_1, \dots, p_r\}$ احتمال انتخاب هریک از عملها و $p(n+1) = T[v(n), \beta(n), p(n)]$ احتمال انتخاب هریک از عملها در لحظه n بصورت زیر است:

$$P(n) \equiv \{p_1(n), p_2(n), \dots, p_r(n)\}$$

بطوریکه

$$\sum_{i=1}^r p_i(n) = 1, \quad \forall n, \quad p_i(n) = \text{Prob}[\alpha(n) = \alpha_i] \quad (5)$$

در آغاز فعالیت اتومات، احتمال اقدامات اتومات آن با هم برابر و مساوی $\frac{1}{r}$ می‌باشند. (که r تعداد اقدامات اتومات می‌باشد) اگر اتوماتی یادگیر در تکرار n ام، یک اقدام خود مانند α_i را انتخاب کند و یک پاسخ مطلوب از محیط دریافت نماید، ($p_i(n)$) احتمال اقدام (α_i) افزایش و احتمال سایر اقدامات کاهش می‌یابد. بالعکس در صورت نامطلوب بودن پاسخ دریافتی از محیط، احتمال اقدام α_i کاهش و احتمال سایر اقدامات افزایش می‌یابد. در هر حال، تغییرات به گونه‌ای صورت می‌گیرد تا حاصل جمع ($p_i(n)$ ها) همواره ثابت و مساوی یک باقی بماند. تغییر احتمال اقدامها بصورت زیر می‌باشد.

الف- پاسخ مطلوب از محیط

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

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

ب- پاسخ نامطلوب از محیط

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

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

توابع f_j و g_j دو تابع غیر منفی هستند که بترتیب توابع باداش و جرمیه نامیده می‌شوند. در یک الگوریتم یادگیری تقویتی خطی (در اتوماتیکی با چند اقدام) توابع f_j و g_j بصورت زیر تعریف شده‌اند

$$\begin{aligned} p_q(k+1) &= p_q(k) \\ p_j(k+1) &= p_j(k) \quad \forall j, j \neq q \\ \alpha_i &\notin v(k) \end{aligned} \quad (8)$$

ج- برای هر $\alpha_i \notin v(k)$

$$p_i(k+1) = p_i(k)$$

شکل (۵) چرخه‌ی بهنگام سازی احتمال درستی شرط قواعد توسط اتوماتای یادگیر را نشان میدهد. شایان ذکر است که با وجود حداقل دو قاعده فعال، چرخه‌ی مذکور فعالیت‌های خود را انجام می‌دهد. در صورت وجود فقط یک قاعده‌ی فعال، همان قاعده جهت ارزیابی شرط انتخاب می‌گردد، بدون تغییر احتمال انتخاب قواعد.

پس از شروع کار سیستم، در بازه‌های زمانی مشخص (σt_i) احتمال درستی بخش شرط (احتمال اجرای) قواعد، براساس اطلاعات بدست آمده اخیر به همان روش بیان شده در بخش ۴-۵-۲، بهنگام می‌شوند. بهنگام سازی احتمال درستی شرط هر قاعده تا زمانی تکرار می‌شود که آنگ تغییرات آن کمتر از ۶ گردد. متعاقباً زمان اجرای یک قاعده زمانی بهنگام می‌شود که احتمال درستی شرط آن، نهایی و نیز زمان اجرای کلیه فرزندان آن بهنگام شده باشند. در آغاز هر بازه‌ی زمانی (σt_{i+1}) از اجرای سیستم، احتمال انتخاب اقدامات اتوماتای یادگیر متناسب با احتمال بهنگام شده‌ی درستی شرط قواعد و احتمال انتخاب اقدامات اتوماتای یادگیر در انتهای بازه‌ی زمانی قبلی (σt_i)، بهنگام می‌شوند.

طبق (۹) بهنگام می‌شوند.

$$p_{R_i,1}(\sigma_k)$$

احتمال انتخاب اقدام نام (قاعده R_i) محاسبه شده به روش-Ex SJF_{PRO}-V.2.8 در پایان σt_k در مقیاس ۱

$$p_{R_i,2}(\sigma_k)$$

احتمال انتخاب اقدام نام (قاعده R_i) محاسبه شده توسط اتوماتای یادگیر در پایان σt_k

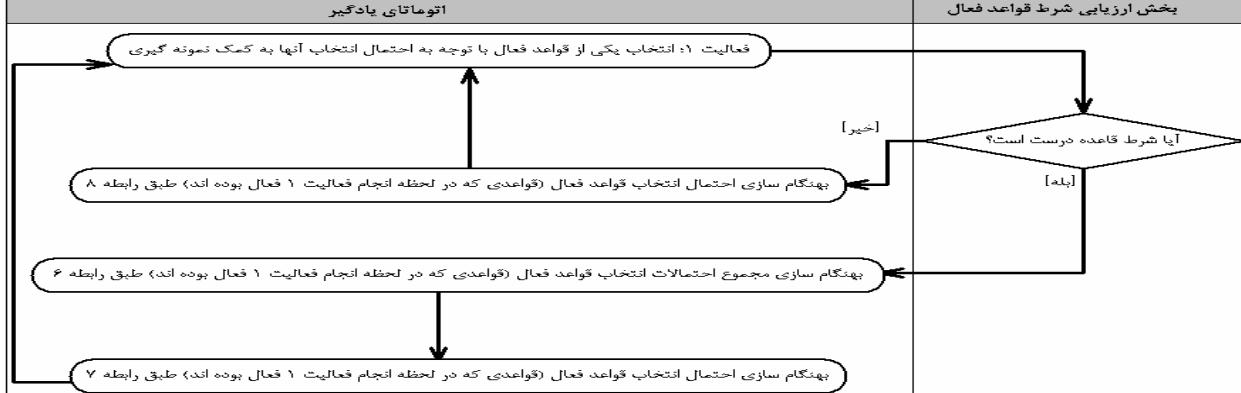
$$\phi(t) = \text{Min}(k/k', 1) \quad \exists k' \geq 25$$

احتمال انتخاب اقدام نام (قاعده R_i) محاسبه شده توسط اتوماتای یادگیر در پایان σt_k

$$p_{R_i}(\sigma_{k+1}) = \frac{\phi(t) * p_{R_i,2}(\sigma_k) + p_{R_i,1}(\sigma_k)}{\phi(t) + 1}$$

احتمال انتخاب اقدام نام (قاعده R_i) در پایان σt_k توسط اتوماتای یادگیر و در آغاز σt_{k+1} بر

(۹)



شکل (۵): چرخه‌ی بهنگام سازی احتمال درستی شرط قواعد توسط اتوماتای یادگیر

لحظه، تعداد اقدامات اتوماتا متغیر خواهد بود [۲۰]. لذا لازم است در هر بار تغییر تعداد قواعد فعال، پردازش‌هایی در رابطه با احتمال انتخاب اقدام‌های اتوماتا صورت پذیرد که در ادامه به شرح آن خواهیم پرداخت. اگر تعداً قواعد، n باشد، مجموعه اقدام‌های اتوماتا در حالت کلی، $\{\alpha_1, \dots, \alpha_n\}$ خواهد بود. یعنی بازای هر قاعده، یک اقدام تعریف می‌شود. لذا با توجه به مطالب پیش‌گفته، بدیهی است که در هر لحظه مجموعه اقدام‌های اتوماتا زیر مجموعه‌ای از مجموعه‌ای فوق خواهد بود. V_i نشان‌دهنده نامین زیرمجموعه اقدام‌های بطوریکه $1 \leq i \leq 2^n - 1$. $V(k)$ نشان‌دهنده زیرمجموعه‌ای از V است که در لحظه K فعال است.

شایان ذکر است که احتمال انتخاب هریک از اقدامها در لحظه k مناسب است با احتمال درستی بخش شرط قاعده‌ی متناظر آن اقدام در *Correctness probability R_i 's condition at instant k*)

$$p_i(k) = \text{prob}[\alpha_i(k) = \alpha_i] \approx \text{of rule}$$

همانطور که پیش از این بیان شد، در این روش احتمال درستی شرط قواعد در آغاز، بر مبنای تکمیک ارائه شده در روش Ex-SJF_{PRO}-V.2.8 محاسبه می‌شود. لذا با فرض اینکه $K(0)$ نشان‌دهنده مجموع احتمالات درستی شرط قواعد در آغاز باشد. با توجه به رابطه‌ی (۵) احتمال انتخاب هر اقدام در آغاز در مقیاس ۱ طبق رابطه‌ی زیر محاسبه می‌گردد:

$$p_i(0) = (\text{Correctness probability of rule } R_i \text{'s condition at instant } 0) / k(0)$$

در لحظه k با فرض اینکه اقدامات متعلق به $V(k)$ ، فعال باشند، احتمال انتخاب اقدامها به روش زیر بهنگام می‌شوند:

$$K(k) = \sum_{\substack{q \\ \alpha_q \in v(k)}} p_q(k) \quad (8)$$

الف- پاسخ مطلوب از محیط در جواب اعمال اقدام ($\alpha_q \in v(k)$) بر محیط

$$(k+1) = p_q(k) - ap_q(k) + ak(k) \quad \alpha_q \in v(k) \\ (k+1) = p_j(k) - ap_j(k) \quad \forall j, j \neq q, \quad \alpha_j \in v(k) \quad (7)$$

ب- پاسخ نامطلوب از محیط در جواب اعمال اقدام ($\alpha_q \in v(k)$) بر محیط

شکل (۵): چرخه‌ی بهنگام سازی احتمال درستی شرط قواعد توسط اتوماتای یادگیر

۴- مقایسه و ارزیابی روش‌های زمان‌بندی

قواعد

داده پویا را شبیه‌سازی می‌کند، مدیریت کلیه‌ی امور مربوط به اقلام داده را بر عهده دارد. واحد مدیریت قواعد که خود شامل چند بخش کوچکتر دیگر است کلیه امور مربوط به حفظ، نگهداری، بهنگام‌سازی، فعال سازی قواعد و تولید تراکنش براساس بخش عمل آنها را بر عهده دارد. واحد مدیریت تراکنش‌ها مسئول نگهداری، مدیریت، زمان‌بندی و اجرای تراکنش‌های تولید شده توسط واحد مدیریت قواعد و کاربر می‌باشد. شکل (۶) معماری ADSS را نشان می‌دهد (بخش اتوماتی یادگیر مختص روش EX-SJF_{ESTLA} است).

در بخش طراحی و پیاده‌سازی شبیه‌ساز سیستم پایگاه داده پویا، سعی شد با افزودن ویژگی‌هایی به ADSS موجود، رفتار سیستم مدیریت پایگاه داده پویا را به طور کاملتری شبیه‌سازی و متعاقباً عملکرد روش-

های زمان‌بندی موجود را با دقت بیشتری مقایسه و ارزیابی نماییم. از جمله‌ی این ویژگیها عبارتند از افزودن کامپایلرهای شرط و دستور به سیستم موجود، بقسمی که به کمک آنها می‌توانیم هنگام تولید قواعد پویا از عبارات شرطی و دستورات واقعی در بخش شرط و عمل قواعد استفاده کنیم و در زمان اجرا نیز با کامپایل عبارات شرطی بخش شرط قواعد فعال و دستورات بخش عمل قواعد در حال اجرا، محیط شبیه‌سازی را تا حد امکان به محیط‌های واقعی نزدیک‌تر نماییم. این در حالی است که در سیستم قبلی بدلیل عدم وجود کامپایلرهای مذکور، به گونه‌ای مجازی به تولید شرط و دستور برای قواعد پویا می‌پرداختیم [۸].

در تولید تصادفی قواعد پویا، پارامترهای مورد نیاز پوشش کلیه‌ی حالات و ویژگی‌های قواعد پویا در سیستم‌های واقعی در نظر گرفته شده‌اند.

در مرجع [۸] چارچوبی به منظور مقایسه روش‌های زمان‌بندی قواعد ارائه شده است. این چارچوب شامل پنج پارامتر ارزیابی شامل میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی، توان عملیاتی، میزان زمان سریار محاسباتی به ازای هر تراکنش و بهره پردازشگر می‌باشد. اولویت این پارامترها (W_i در رابطه (۱۰)) به ترتیب معرفی آنها در عبارت پیشین می‌باشد. جدول (۴) تعریف پارامترهای ارزیابی فوق را به صورت فرمال نشان می‌دهد.

در این چارچوب به منظور شبیه‌سازی رفتار سیستم پایگاه داده پویا یک محیط آزمایشگاهی به نام شبیه‌ساز سیستم پایگاه داده پویا (ADSS) طراحی و پیاده‌سازی شده است که با پیاده‌سازی روش‌های زمان‌بندی مختلف در آن می‌توان عملکرد آنها را مورد ارزیابی قرار داد. معناری، نحوه‌ی طراحی و پیاده‌سازی ADSS و روش‌های مختلف زمان‌بندی قواعد به طور مفصل در مرجع [۸] شرح داده شده است. یکی از ویژگی‌های مهم ADSS انعطاف‌پذیر بودن آن است. به این معنی که هر روش زمان‌بندی را می‌توان در آن پیاده‌سازی کرد، بدون آنکه نیاز به تغییر سایر بخش‌ها باشد.

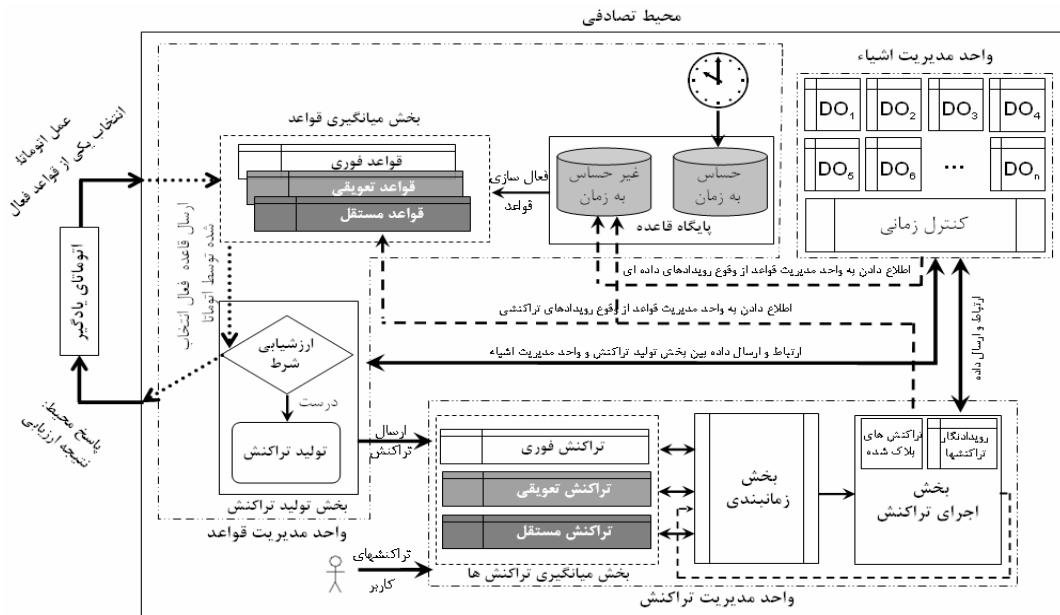
ADSS دارای معماری سه بخشی است: (۱) واحد مدیریت اشیاء داده، (۲) واحد مدیریت قواعد و (۳) واحد مدیریت تراکنش‌ها. واحد مدیریت اشیاء داده که در حقیقت بخش پایگاه داده یک سیستم پایگاه

جدول (۴): تعریف پارامترهای ارزیابی [۸]

$N = \text{Number of Executed Rules}$	$T_2^i = \text{Start of Execution Time of } i^{\text{th}} \text{ Rule}$		
$ART = \text{Average Response Time}$	$T = (T_2^N + \text{ExecutionTime of } N^{\text{th}} \text{ Rule}) - T_1$		
$RTSV = \text{Response Time Standard Variance}$	$T^* = \sum_{i=1}^N \text{Real Execution Time of } i^{\text{th}} \text{ Rule}$		
$U_{CPU} = \text{CPU Utilization}$			
$TOPT = \text{Time Overhead Per Transaction}$			
$U_{CPU} = \frac{T^*}{T} * 100$	$RTSV = \sqrt{\frac{\sum_{i=1}^N ((T_2^i - T_1^i) - ART)^2}{N}}$	$ART = \frac{\sum_{i=1}^N (T_2^i - T_1^i)}{N}$	$\text{Throughput} = \frac{N}{T^*}$
			$TOPT = \frac{T - T^*}{N}$

قاعده‌ای استفاده شده است که در آن پیوستگی رویداد-شرط و شرط-عمل قواعد از هر سه نوع فوری، تعویقی و مستقل می‌تواند باشد [۷، ۸]. نتایج حاصل از مقایسه‌ی نسخه‌های مختلف روش EX-SJF جداول (۵)، (۶) و (۷) نشان داده شده است.

آزمایشات در سه حالت مختلف انجام شده است: حالت فوری، حالت تعویقی و حالت ترکیبی. در حالت فوری از پایگاه قاعده‌ای استفاده شده است که در آن پیوستگی رویداد-شرط و شرط-عمل فقط از نوع فوری است. در حالت تعویقی نیز پیوستگی رویداد-شرط و شرط-عمل قواعد فقط از نوع تعویقی می‌باشد و سرانجام در حالت ترکیبی از پایگاه



شکل (۶): معماری سیستم شبیه‌ساز پایگاه داده پویا (ADSS)

جدول (۷): نتایج ارزیابی روش‌های زمان بندی در حالت ترکیبی

روش‌ها							پارامترهای ارزیابی
روش	نیازمندی	زمان بندی	محدودیت	میزان	میزان	میزان	ارزیابی
Random	۱	۱	۳	۵	۷		
Static Priority	۱	۲	۴	۴	۶		
FCFS	۱	۲	۴	۴	۶		
EDF _{PD}	۱	۲	۵	۳	۵		
EDF _{DIV}	۱	۲	۵	۳	۵		
EDF _{SL}	۱	۲	۶	۶	۵		
E _x -SJF _{EXA}	۱	۲	۳	۳	۴		
E _x -SJF _{PRO}	۱	۲	۳	۳	۴		
E _x -SJF _{PRO-V.1.8}	۱	۲	۲	۲	۳		
E _x -SJF _{PRO-V.2.8}	۱	۲	۱	۱	۲		
E _x -SJF _{ESTLA}	۱	۲	۱	۱	۱		

عددی که مقابله هر روش زمان بندی و پارامتر ارزیابی در هر خانه نوشته شده است، نشان دهنده رتبه آن روش زمان بندی در بین کلیه روش‌های مورد ارزیابی از نقطه نظر آن پارامتر ارزیابی است. رتبه بندی نهایی روش‌ها بر طبق (۱۰) محاسبه می‌شود. شایان ذکر است که رتبه نهایی روشی بهتر خواهد بود، که مقدار $S(k)$ متناظرش کمتر باشد.

شکل‌های (۷) الی (۱۰) چند نمونه از نمودارهای ارزیابی روش‌های زمان‌بندی اجرای قواعد در حالات مختلف، براساس پارامترهای ارزیابی پیش-گفته را نشان می‌دهند. بدین ترتیب که در شکل (۷) روش‌های مختلف، براساس میانگین زمان پاسخگویی در حالت تعویقی با یکدیگر مقایسه شده‌اند، شکل (۸) به ارزیابی این روش‌ها از نقطه نظر توان عملیاتی در حالت فوری می‌پردازد و شکل‌های (۹) و (۱۰) ارزیابی روش‌های مختلف را براساس میانگین زمان سربار محاسباتی به ازای هر تراکنش و انحراف معیار زمان پاسخگویی در حالت ترکیبی نشان می‌دهند. همانگونه که در شکل (۷) ملاحظه می‌شود، دو روش E_x-SJF_{ESTLA} دارای کمترین میانگین زمان پاسخگویی بین کلیه

جدول (۵): نتایج ارزیابی روش‌های زمان بندی در حالت تعویقی

روش‌ها							پارامترهای ارزیابی
روش	نیازمندی	زمان بندی	محدودیت	میزان	میزان	میزان	ارزیابی
Random	۳	۱	۳	۶	۵		
Static Priority	۳	۱	۳	۶	۵		
FCFS	۳	۱	۳	۵	۵		
EDF _{PD}	۳	۱	۳	۵	۵		
EDF _{DIV}	۳	۱	۴	۸	۵		
EDF _{SL}	۱	۱	۵	۷	۵		
E _x -SJF _{EXA}	۲	۱	۴	۷	۴		
E _x -SJF _{PRO}	۲	۱	۴	۴	۴		
E _x -SJF _{PRO-V.1.8}	۲	۱	۳	۳	۳		
E _x -SJF _{PRO-V.2.8}	۲	۱	۲	۲	۲		
E _x -SJF _{ESTLA}	۲	۱	۱	۱	۱		

جدول (۶): نتایج ارزیابی روش‌های زمان بندی در حالت فوری

روش‌ها							پارامترهای ارزیابی
روش	نیازمندی	زمان بندی	محدودیت	میزان	میزان	میزان	ارزیابی
Random	۳	۳	۲	۵	۵		
Static Priority	۳	۳	۲	۵	۵		
FCFS	۱	۱	۲	۵	۵		
EDF _{PD}	۲	۱	۱	۴	۴		
EDF _{DIV}	۲	۱	۱	۴	۴		
EDF _{SL}	۲	۲	۴	۳	۳		
E _x -SJF _{EXA}	۲	۳	۵	۴	۴		
E _x -SJF _{PRO}	۲	۲	۴	۳	۳		
E _x -SJF _{PRO-V.1.8}	۲	۳	۳	۲	۲		
E _x -SJF _{PRO-V.2.8}	۲	۲	۳	۱	۲		
E _x -SJF _{ESTLA}	۲	۲	۲	۱	۱		

$k \in \text{Rule Scheduling Approaches Set}$

W_i is the weight of i th evaluation metric

$j=1, 2$ and 3 indicates deferred, immediate and composite state, respectively

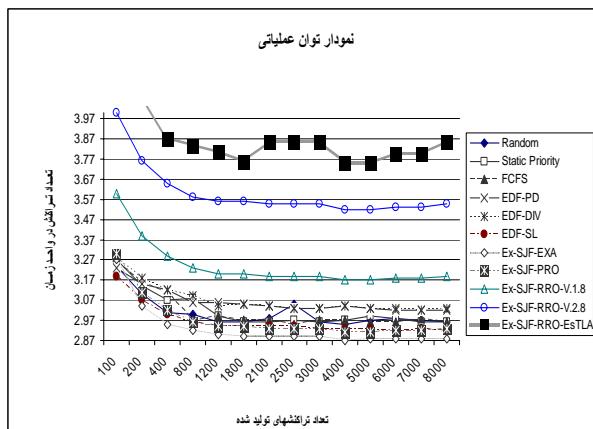
$Rnk_{i,j}(k)$ is the rank of approach k in j th state based on i th evaluation metric

$S(k)$ is the total score of approach k among all approaches

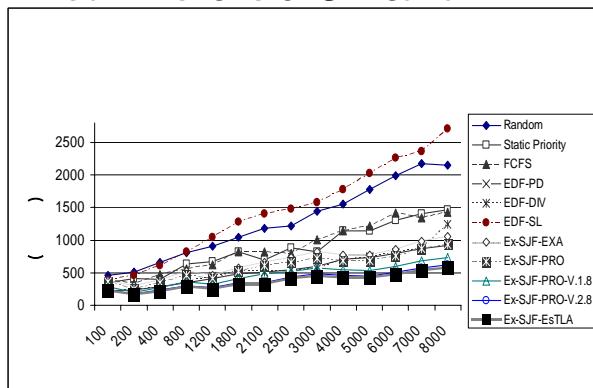
تولید شده در سیستم افزایش یابد میزان افزایش میانگین زمان پاسخگویی در این چهار روش از سایر روش‌ها کمتر خواهد بود.

شکل (۸) رفتار روش‌های زمانبندی در حالت فوری از نقطه نظر توان عملیاتی را نشان می‌دهد. روش $E_x\text{-SJF}_{\text{EXA}}$ دارای کمترین توان عملیاتی است. دو روش $E_x\text{-SJF}_{\text{PRO}}$ و EDF_{SL} دارای توان عملیاتی بیشتری است. دو روش $E_x\text{-SJF}_{\text{PRO-V.2.8}}$ و $E_x\text{-SJF}_{\text{PRO-V.1.8}}$ هستند. روش‌های $E_x\text{-SJF}_{\text{EXA}}$ از این حیث در رتبه‌های بهتری نسبت به سه روش پیش قرار دارند به طوریکه روش $E_x\text{-SJF}_{\text{PRO-V.1.8}}$ موجب افزایش ۹ درصدی توان عملیاتی روش $E_x\text{-SJF}_{\text{PRO}}$ می‌شود و روش $E_x\text{-SJF}_{\text{PRO-V.2.8}}$ نیز موجب افزایش ۱۱ درصدی توان عملیاتی روش $E_x\text{-SJF}_{\text{PRO-V.1.8}}$ می‌شود. اما این روش‌ها نیز خود از نظر این معیار ارزیابی بعد از چهار روش $E_x\text{-SJF}_{\text{ESTLA}}$, «اولویت ایستا», «FCFS» و «انتخاب اتفاقی» قرار می‌گیرند. به طوریکه توان عملیاتی روش $E_x\text{-SJF}_{\text{ESTLA}}$ ۷ درصد بهتر از توان عملیاتی روش $E_x\text{-SJF}_{\text{PRO-V.2.8}}$ است. همانگونه که در شکل (۸) نشان داده شده است دو روش EDF_{PD} و EDF_{DIV} دارای بیشترین توان عملیاتی در میان روش‌های مختلف زمانبندی در این حالت هستند.

همانگونه که در شکل (۹) ملاحظه می‌شود روش «انتخاب اتفاقی» دارای کمترین «سریار محاسباتی» به ازای هر تراکنش در حالت ترکیبی است. در حالیکه رفتار بقیه روش‌ها در رابطه با این معیار تقریباً یکسان است. شکل (۱۰) نشان می‌دهد که دو روش $E_x\text{-SJF}_{\text{ESTLA}}$ و $E_x\text{-SJF}_{\text{PRO}}$ در حالت فوری از این نظر رشد میانگین زمان پاسخگویی روش‌های زمانبندی در حالت تعویقی دارند.

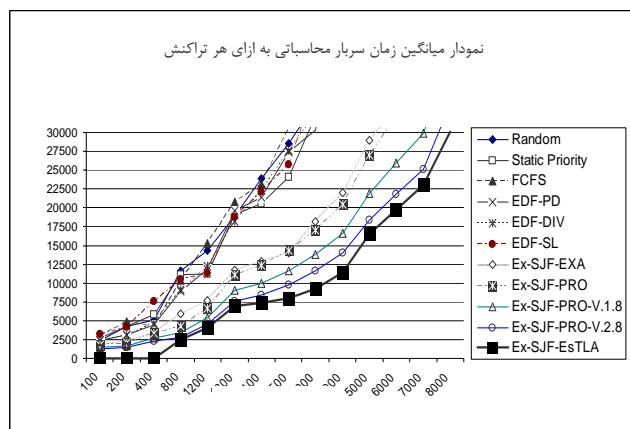


شکل (۸): نمودار توان عملیاتی روش‌های زمانبندی در حالت فوری

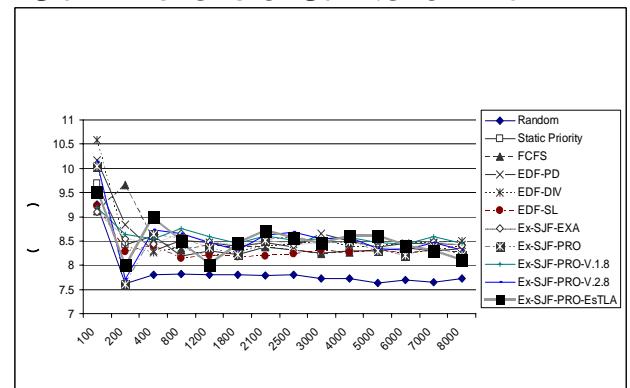


شکل (۱۰): نمودار انحراف معیار زمان پاسخگویی روش‌های زمانبندی در حالت ترکیبی

روش‌ها در حالت تعویقی هستند. روش‌های $E_x\text{-SJF}_{\text{PRO-V.1.8}}$, $E_x\text{-SJF}_{\text{EXA}}$ و $E_x\text{-SJF}_{\text{PRO}}$ از نظر میانگین زمان پاسخگویی در رتبه‌های بعدی قرار دارند. سایر روش‌ها از این حیث دارای رفتار مشابهی هستند. با استفاده از نمودار شکل (۷) و پس از محاسبه میزان کاهش میانگین زمان پاسخگویی هر یک از نسخه‌های روش $E_x\text{-SJF}_{\text{ESTLA}}$ نسبت به سایر روش‌ها به این نتیجه می‌رسیم که استفاده از روش $E_x\text{-SJF}_{\text{ESTLA}}$ موجب کاهش $9/8$ درصدی میانگین زمان پاسخگویی نسبت به روش $E_x\text{-SJF}_{\text{PRO-V.2.8}}$ و استفاده از روش $E_x\text{-SJF}_{\text{PRO-V.2.8}}$ موجب کاهش $15/8$ درصدی میانگین زمان پاسخگویی نسبت به روش $E_x\text{-SJF}_{\text{PRO-V.1.8}}$ و استفاده از روش $E_x\text{-SJF}_{\text{PRO-V.1.8}}$ موجب کاهش $18/8$ درصدی میانگین زمان پاسخگویی نسبت به روش $E_x\text{-SJF}_{\text{PRO}}$ و $E_x\text{-SJF}_{\text{PRO-V.1.8}}$. در حالت تعویقی بسیار نزدیک به هم بوده ولی در حالیست که دو روش اخیر ($E_x\text{-SJF}_{\text{EXA}}$ و $E_x\text{-SJF}_{\text{PRO}}$) نیز به نوبه‌ی خود موجب کاهش $43/43$ درصدی میانگین زمان پاسخگویی نسبت به سایر روش‌ها در حالت تعویقی خواهند شد. نکته‌ی دیگری که از این شکل مشخص است، شبیه نمودار است که برای روش‌های $E_x\text{-SJF}_{\text{PRO-V.1.8}}$, $E_x\text{-SJF}_{\text{PRO-V.2.8}}$, $E_x\text{-SJF}_{\text{ESTLA}}$ از بقیه روش‌ها کمتر بوده که نشان می‌دهد سرعت رشد میانگین زمان پاسخگویی در این سه روش از بقیه روش‌ها بسیار کمتر است، به عبارت دیگر چنانچه تعداد تراکنش‌های



شکل (۷): نمودار میانگین زمان سربار محاسباتی به ازای هر تراکنش



شکل (۹): نمودار میانگین زمان سربار محاسباتی به ازای هر تراکنش روش‌های زمانبندی در حالت ترکیبی

کارایی فرایند زمانبندی قواعد از نظر میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی و توان عملیاتی افزایش یافته است. از نظر منطقی نیز انتظار می‌رود که چنین بهبودی حاصل شود. همانطور که پیش از این نیز بیان شد، از بین قواعد فعل فقط آنهای اجرا می‌شوند که شرط‌شان در لحظه‌ی ارزیابی درست باشد و سایر قواعد فعل از گردونه‌ی پردازش خارج می‌شوند. اما زمانی که صرف انتخاب این دسته از قواعد فعل و نیز ارزیابی آنها می‌شود از جمله زمان‌های غیر مفید کارکرد سیستم محسوب می‌شود که مستقیماً در توان عملیاتی (تعداد قواعد اجرا شده در واحد زمان)، میانگین زمان پاسخگویی (میانگین زمان انتظار قواعد برای اجرا) و به طور غیر مستقیم در انحراف معیار زمان پاسخگویی (میانگین فاصله‌ی زمان پاسخگویی هر قاعده از میانگین زمان پاسخگویی) تأثیرگذار است.

بدیهی است هرچه اولویت پردازش قواعد فعلی که بدون اجرا شدن از گردونه پردازش، خارج می‌شوند کمتر باشد، کارایی سیستم از بُعد پارامترهای ارزیابی پیش‌گفته بهتر می‌شود. لذا اگر فرایند زمانبندی قواعد فعل، مبتنی بر احتمال درستی شرط قواعد باشد، دستیابی به این هدف را ممکن می‌سازد. علاوه بر آن اگر مکانیزمی وجود داشته باشد که به تخمین دقیقتر احتمال درستی شرط قواعد در حین اجرای سیستم پردازش، امکان نیل به این هدف را بیشتر می‌سازد. کلی که در روش فرایند زمانبندی قواعد در روش اخیر نسبت به روش $E_x-SJF_{PRO}-V.2.8$ فقط در حد بهنگام سازی احتمال درستی شرط برخی از قواعد فعل، طبق (۷) است که بسیار ناچیز و قابل اغماض می‌باشد.

۵- نتیجه‌گیری و کارهای آینده

در این مقاله ابتدا پایگاه داده پویا و چرخه‌ی پردازش قواعد را تعریف کردیم. سپس جایگاه و اهمیت فرایند زمانبندی قواعد را در چرخه‌ی پردازش قواعد بیان نمودیم. در ادامه به معرفی روش‌های مختلف زمان‌بندی قواعد پرداختیم. سپس به منظور بهبود کارایی فرایند زمانبندی موجود ($E_x-SJF_{PRO}-V.2.8$) درصد بهبود فرایند زمانبندی قواعد فعل به کمک اتماتایی یادگیر برآمدیم. بر همین اساس روش جدیدی ارائه کردیم و آن را E_x-SJF_{ESTLA} نامیدیم. در نهایت به بررسی عملکرد روش پیشنهادی جدید در مقایسه با روش‌های زمانبندی موجود بر اساس تعدادی پارامتر ارزیابی بر مبنای نتایج حاصل از انجام آزمایشات متعدد و متنوع در یک محیط آزمایشگاهی مبتنی بر معماری سه بخشی (ADSS) پرداختیم. نتایج آزمایشات بهبود فرایند زمانبندی قواعد را در نسخه‌ی E_x-SJF_{ESTLA} از نظر میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی و توان عملیاتی نشان می‌دهد.

از جمله فعالیت‌های تحقیقاتی که در راستای تکمیل پژوهش جاری می‌توان انجام داد، ارائه یک بستر استاندارد به منظور تست و ارزیابی نحوه عملکرد روش‌های زمانبندی قواعد است. همچنین، تلاش در جهت افزایش هرچه بیشتر کارایی فرایند زمانبندی قواعد پویا از طریق توسعه روش‌های زمانبندی موجود با ارائه روش‌های جدید، از دیگر فعالیت‌های پژوهشی مرتبه پیشنهادی برای آینده است.

V.2.8 دارای کمترین انحراف معیار زمان پاسخگویی هستند. روش E_x-SJF_{ESTLA} موجب کاهش $6/9$ درصدی انحراف معیار زمان پاسخگویی روش $E_x-SJF_{PRO}-V.2.8$ شده است. روش $E_x-SJF_{PRO}-V.1.8$ نیز به نوبه‌ی خود موجب کاهش $16/4$ درصدی انحراف معیار زمان پاسخگویی روش $E_x-SJF_{PRO}-V.1.8$ شده است. بعد از این سه روش، نسخه‌های E_x-SJF_{PRO} روش E_x-SJF_{EXA} و E_x-SJF_{PRO} به همراه دو روش EDF_{DIV} و EDF_{PD} به ترتیب دارای کمترین انحراف معیار زمان پاسخگویی هستند. روش $E_x-SJF_{PRO}-V.1.8$ نیز موجب کاهش $21/4$ درصدی انحراف معیار زمان پاسخگویی روش E_x-SJF_{PRO} شده است و پس از آنها دو روش «اولویت ایستا» و $FCFS$ دارای کمترین انحراف معیار زمان پاسخگویی هستند. نمودار شکل (۱۰) همچنین نشان می‌دهد دو روش EDF_{SL} و «انتخاب اتفاقی» به ترتیب دارای بیشترین انحراف معیار زمان پاسخگویی هستند. نکته‌ی دیگری که از این شکل مشخص می‌شود شبیه شدن نمودار برای روش‌های EDF_{SL} و «انتخاب اتفاقی» است که نشان می‌دهد در این روشها در صورت افزایش تعداد تراکنش‌های تولید شده، تجمع زمان‌های پاسخگویی تراکنش‌ها حول نقطه‌ی میانگین به شدت کاهش خواهد یافت. شبیه نمودار برای روش‌های E_x-SJF_{ESTLA} و $E_x-SJF_{PRO}-V.2.8$ از همه کمتر بوده و نشان می‌دهد سرعت رشد انحراف معیار زمان پاسخگویی در این چهار روش از بقیه روش‌ها بسیار کمتر است.

نتایج آزمایشات و تحلیلهای مربوطه حاکی از آن است که در مجموع روش E_x-SJF_{ESTLA} از سایر روش‌های مورد ارزیابی کارتر است. همانطور که ملاحظه می‌شود زمانبندی قواعد فعل بر مبنای احتمال درستی شرط آنها و نه بر مبنای زمان احرای آنها به کمک اتماتایی یادگیر، منجر به بهبود روش زمانبندی E_x-SJF از نظر میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی و توان عملیاتی می‌شود. فرایند تخمین احتمال وقوع قواعد در E_x-SJF_{ESTLA} به گونه‌ای است که سربار محاسباتی بر سیستم تحمیل نمی‌کند. لذا نسخه‌ی SJF از نظر میزان سربار محاسباتی به ازای هر تراکنش و بهره پردازشگر برابر می‌کند.

جدول (۸) درصد کارتر بودن روش E_x-SJF_{ESTLA} نسبت به روش $E_x-SJF_{PRO}-V.2.8$ (کارترین روش زمانبندی موجود) در سه حالت تعویقی، فوری و ترکیبی براساس سه پارامتر ارزیابی میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی و توان عملیاتی نشان می‌دهد.

جدول (۸) درصد کارتر بودن روش E_x-SJF_{ESTLA} نسبت به روش $E_x-SJF_{PRO}-V.2.8$

پارامترهای ارزیابی	حالات مقایسه	انحراف معیار زمان پاسخگویی	میانگین زمان پاسخگویی	توان عملیاتی
فوري		٪۶/۴	٪۸/۶	٪۷/۷
تعويقي		٪۱۱/۹	٪۹/۸	٪۶/۷
ترکيبي		٪۶/۹	٪۷/۸	٪۱۲/۳

همانطور که مشاهده می‌شود با بهبود فرایند زمانبندی قواعد فعل در نسخه جدید (E_x-SJF_{ESTLA}) نسبت به نسخه‌های ارائه شده پیشین،

مراجع

- [13] R.M. Sivasankaran, J.A. Stankovic, D. Towsley, B. Purimeta, K. Ramamritham, "Priority Assignment in Real-Time Active Databases", The International Journal on Very Large Data Bases, Vol. 5, No. 1, January 1996.
- [14] S. Potaminsto, M. Stonebraker, "The POSTGRES Rule System", in Active Database Systems: Triggers and Rules for Advanced Systems, Morgan Kaufmann Publishers, Sanfrancisco, CA, 1996.
- [15] J. Stankovic, S.M.D. Natale, G.C. Buttazzo, "Implications of Classical Scheduling Results for Real-Time Systems", IEEE Computer Society Press, Los Alamitos, CA, 1995.
- [16] A. Geppert, S. Gatziu, K.R. Dittrich, H. Fritsch, Vaduva, A.; "Architecture and implementation of the active object-oriented database management system SAMOS", Technical Report 95.29, CS Department, University of Zurich, 1995.
- [17] S. Gatziu, Events in an Active Object-Oriented Database System, PhD thesis, University of Zurich, 1994.
- [18] E.N. Hanson, J. Widom, "An Overview of Production Rules in Database Systems". In the Knowledge Engineering Review, Vol.8, No.2, pp.121-143, 1993.
- [19] K.S. Narendra, L. Thathachar, "Learning Automata: An Introduction", Prentice Hall, Inc., Upper Saddle River, NJ, 1989.
- [20] M.A.L. Thathachar, B.R. Harita, "Learning Automata with Changing Number of Actions", IEEE Transaction on Systems, Man and Cybernetics, Vol. SMC-17, NO. 6, pp. 1095-1100, 1987.
- [1] ع. رسولزادگان، ا. عبداللهزاده، "روش جدید زمانبندی قواعد بر مبنای تخمین احتمال اجرای قواعد در پایگاه داده پویا"، در مجموعه مقالات پانزدهمین کنفرانس مهندسی برق ایران، ۲۷-۲۵ اردیبهشت ۱۳۸۶، مرکز تحقیقات مخابرات ایران، تهران، ایران.
- [2] ع. رسولزادگان، ا. عبداللهزاده، "ADSS شبیه‌ساز پایگاه داده پویا برای مقایسه تطبیقی روش‌های زمانبندی قواعد"، در مجموعه مقالات پانزدهمین کنفرانس مهندسی برق ایران، ۲۷-۲۵ اردیبهشت ۱۳۸۶، مرکز تحقیقات مخابرات ایران، تهران، ایران.
- [3] ع. رسولزادگان، ا. عبداللهزاده، "اولویت‌گذاری روشنی جدید برای تخمین احتمال وقوع قواعد در پایگاه داده پویا به منظور بهبود زمانبندی اجرای قواعد در آن"، در مجموعه مقالات یازدهمین کنفرانس سالانه انجمن کامپیوتر ایران، ۱۳۸۴، مرکز همایش‌های بین‌المللی صدا و سیما، تهران، ایران.
- [4] A. Rasoolzadegan, A. Abdollahzadeh, "A New Rule Scheduling Approach based on Estimation of Rule Execution Probability in Active Database", Journal of Convergence Information Technology (JCIT), Vol. 3, No. 3, pp. 6-13, September 2008.
- [5] A. Rasoolzadegan, "A New Rule Scheduling Approach based on Estimation of Rule Execution Probability in Active Database System", MSc Thesis, Amirkabir University of Technology (Tehran PolyTechnic), 2007.
- [6] A. Rasoolzadegan, R. Alesheykh, A. Abdollahzadeh, "A New Approach for Event Triggering Probability Estimation in Active Database Systems to Rule Scheduling Improvement", 2nd IEEE International Conference on Information & Communication Technologies: From Theory To Applications, Damascus, Syria , pp. 1069-1071, April 24 - 28, 2006.
- [7] A. Rasoolzadegan, R. Alesheykh, A. Abdollahzadeh, "Measuring Evaluation Parameters in Benchmarking Rule Scheduling Methods in Active Database Systems", The IEEE International Conference on Computer and Communication Engineering, Kuala Lumpur, Malaysia, pp. 225-230, 2006.
- [8] R. Alesheykh, "An Effective Rule Selection Approach in Active Database Systems", MSc Thesis, Amirkabir University of Technology (Tehran PolyTechnic), 2005.
- [9] S. Ceri, C. Gennaro, S. Paraboschi, G. Serazzi, "Effective Scheduling of Detached Rules in Active Database", IEEE Transaction on Knowledge and Data Engineering, Vol. 15, No.1, 2003.
- [10] M.A.L. Thathachar, P.S. Sastry, "Varieties of Learning Automata: An Overview", IEEE Transaction on Systems, Man and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, pp. 711-722, 2002.
- [11] A. Vadua, Rule Development for active database, PhD Thesis, CS Department, University of Zurich, 1999.
- [12] S.G. Vaduva, K.R. Dittrich, "Investigating Termination in Active Database Systems with Expressive Rule Languages", In Proceedings of the 3rd International Workshop on Rules In Database Systems, pp. 149-164, Skovde (Sweden), 1997.

عباس رسولزادگان: دانشجوی دکترای مهندسی نرم‌افزار در دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر، لیست فعالیت‌های تحقیقاتی، پژوهشی و سایر اطلاعات مرتبط در مورد علاقه‌مندی‌ها و زمینه‌های تحقیقاتی نامبرده در آدرس اینترنتی زیر موجود است:



<http://ceit.aut.ac.ir/~86131901/index.htm>

محمد رضا میبدی: استاد دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر، الگوریتم‌های موازی، نظریه زبان‌ها از جمل دروسی است که توسط نامبرده تدریس می‌شود. نتایج فعالیت‌های تحقیقاتی و پژوهشی نامبرده در مجموعه مقالات و مجلات معتبر داخلی و خارجی به چاپ رسیده است. لیست فعالیت‌های تحقیقاتی نامبرده از طریق آدرس اینترنتی زیر قابل دسترسی است:



<http://ceit.aut.ac.ir/softlab/>

واژه نامه

¹ Active Database System Simulator

² Event - Condition - Action

³ Event - Condition Coupling

⁴ Condition - Action Coupling

⁵ Random

⁶ Total Order (Static Priority)

⁷ First Come First Serve

⁸ Earliest Deadline First

⁹ Real-time systems

¹⁰ Extended-Shortest Job First

¹¹ Extended-Shortest Job First-Exactly

¹² Extended-Shortest Job First-Probabilistic

¹³ Extended-Shortest Job First-(based on)-Estimation-(using)-Learning Automata

¹⁴ Unfavorable

¹⁵ Stationary

¹⁶ Non-Stationary

¹⁷ Linear Reward Inaction

¹⁸ Linear Reward Epsilon Penalty

¹⁹ Stochastic