

# ارائه روشی جدید برای تخمین احتمال وقوع قواعد در پایگاه داده پویا به منظور بهبود زمانبندی اجرای قواعد در آن

احمد عبدالله زاده آزمایشگاه سیستم‌های هوشمند <a href="http://ce.aut.ac.ir/islab">http://ce.aut.ac.ir/islab</a> دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر <a href="mailto:ahmad@ce.aut.ac.ir">ahmad@ce.aut.ac.ir</a>	روح‌الله آل‌شیخ آزمایشگاه سیستم‌های هوشمند <a href="http://ce.aut.ac.ir/islab">http://ce.aut.ac.ir/islab</a> دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر <a href="mailto:alesheykh@ce.aut.ac.ir">alesheykh@ce.aut.ac.ir</a>	عباس رسول‌زادگان آزمایشگاه سیستم‌های هوشمند <a href="http://ce.aut.ac.ir/islab">http://ce.aut.ac.ir/islab</a> دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر <a href="mailto:rasoolzade@ce.aut.ac.ir">rasoolzade@ce.aut.ac.ir</a>
--	--	---

**واژه‌های کلیدی:** پایگاه داده پویا، قاعده پویا، زمانبندی قواعد، انتخاب قاعده، تخمین احتمال وقوع، تکنیک‌های یادگیری.

## ۱- مقدمه

پایگاه‌های داده‌ی متداول دارای ماهیت ایستا هستند، یعنی اعمال پرس و جو، بهنگام سازی، درج، حذف، گزارش‌گیری و غیره هر زمان که توسط کاربر درخواست شوند، انجام می‌پذیرند و سیستم مدیریت پایگاه داده هیچ ابتکار عملی در هنگام رخ دادن شرایط خاص در سیستم ندارد. بسیاری از برنامه‌های کاربردی مانند برنامه‌های انبارداری، اتوماسیون کارخانه‌جات و سیستم‌های دارای محاسبات پیچیده‌ی مالی (مانند بازار سهام) نیاز به نظارت خودکار دارند تا در صورت وقوع رویداد خاصی، سیستم مدیریت پایگاه داده واکنش مقتضی را انجام دهد [۱]. برای این منظور باید سیستم پایگاه داده‌ای طراحی شود که امکان تعریف رویدادهای مورد نظر و واکنش‌های متناظر آن‌ها در آن گنجانده شود، به چنین سیستمی، سیستم پایگاه داده‌ی پویا گفته می‌شود. کار بر روی سیستم‌های پایگاه داده‌ی پویا از اوایل دهه‌ی ۸۰ شروع شده است ولی در سال ۱۹۹۶ مشخصه‌ها و ویژگی‌های آن استاندارد شد [۱،۲]. تاکنون چندین نمونه سیستم پایگاه داده‌ی پویا به حالت عملیاتی درآمده است [۳]. برای اطلاعات کامل در زمینه سیستم‌های پایگاه داده پویا به [۳، ۴، ۵] مراجعه نمایید.

رفتار واکنشی سیستم پایگاه داده‌ی پویا بوسیله‌ی مجموعه‌ای از قواعد پویا سازماندهی می‌شود. این قواعد، وضعیت‌های مختلف پایگاه داده را که باید تحت کنترل قرار گرفته و واکنش لازم نسبت به آنها انجام شود، تعیین می‌کنند. در حالت کلی، این قواعد شامل سه بخش به نام‌های رویداد، شرط و عمل هستند. به همین دلیل به آنها قواعد رویداد-شرط-عمل یا به اختصار ECA<sup>۹</sup> گفته می‌شود [۲،۴]. قواعد پویا<sup>۱۰</sup> بر تغییراتی

**چکیده:** سیستم مدیریت پایگاه داده‌ی پویا<sup>۱</sup>، سیستمی است که امکان تعریف مجموعه‌ای از رویدادها و واکنش‌های متناظر آن‌ها را میسر می‌سازد تا در صورت وقوع رویداد خاصی، بتواند بطور خودکار، واکنش مقتضی را انجام دهد. رفتار واکنشی سیستم پایگاه داده‌ی پویا بوسیله‌ی مجموعه‌ای از قواعد پویا<sup>۲</sup> سازماندهی می‌شود. روش زمانبندی اجرای قواعد<sup>۳</sup>، تأثیر بسزایی در کارایی سیستم پایگاه داده پویا دارد. تاکنون روش‌های متعددی برای زمانبندی اجرای قواعد در سیستم‌های پایگاه داده پویا ارائه شده است. در این مقاله، هدف بهبود زمانبندی اجرای قواعد در پایگاه داده پویا می‌باشد. بدین منظور، ابتدا چارچوبی برای مقایسه و ارزیابی روش‌های زمانبندی موجود ارائه می‌شود. در این چارچوب پنج پارامتر ارزیابی میانگین زمان پاسخگویی<sup>۴</sup>، انحراف معیار زمان پاسخگویی<sup>۵</sup>، توان عملیاتی<sup>۶</sup>، میزان سربار محاسباتی<sup>۷</sup> و بهره‌پذیری<sup>۸</sup> در نظر گرفته شده است. براساس این چارچوب روش‌های موجود با یکدیگر مقایسه و با تحلیل نقاط قوت و ضعف هر یک، روش کارتر معرفی می‌گردد. سپس با هدف بهبود فرآیند انتخاب قواعد در روش منتخب، به ارائه روشی برای تخمین احتمال وقوع قواعد می‌پردازیم. آزمایشات انجام شده نشان می‌دهد که تکنیک جدید ارائه شده برای انتخاب قواعد باعث بهبود روش زمانبندی منتخب از نظر میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی و توان عملیاتی می‌شود.

<sup>1</sup> Active Database Management System

<sup>2</sup> Active Rule

<sup>3</sup> Rule Scheduling Approach

<sup>4</sup> Average Response Time

<sup>5</sup> Response Time Variance

<sup>6</sup> Throughput

<sup>7</sup> Time Overhead per Transaction

<sup>8</sup> CPU Utilization

<sup>9</sup> Event-Condition-Action

<sup>10</sup> Active Rule

که روی جداول پایگاه داده انجام می‌شود نظارت می‌کنند. در صورت رخ دادن رویداد خاصی، قاعده‌ای (هایی) که حساس به آن رویداد است (هستند) فعال می‌شود (می‌شوند) اگر شرطی که در قاعده پویا از قبل تعبیه شده است دارای ارزش درست باشد، عمل درج شده در قاعده انجام خواهد شد. در این قسمت مثالی از یک سیستم خرید و فروش سهام را که به نوسانات بازار سهام حساس است، نشان می‌دهیم. قاعده‌ی پویای ارایه شده در زیر براساس استاندارد زبان توصیف قاعده شی‌گرای SAMOS است [۶]:

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، هنگامی که قیمت یک سهم بهنگام می‌شود (Stock.UpdatePrice)، در صورتیکه، حداقل به مقدار  $100 * (1 - e)$  کاهش یافته و خط مشی استفاده شده Low\_risk باشد، سیستم مدیریت پایگاه داده پویا این سهم را خریداری کند.

برای پشتیبانی از قواعد پویا، یک سیستم مدیریت پایگاه داده پویا علاوه بر بخش‌هایی که یک سیستم مدیریت پایگاه داده ایستا دارد، حداقل باید شامل دو بخش (۱) تولید و مدیریت قواعد و (۲) پردازش قواعد باشد. فرآیند تولید قواعد خود، شامل پنج مرحله‌ی (۱) استخراج قواعد، (۲) توصیف قواعد، (۳) پیاده‌سازی قواعد، (۴) بررسی و اعتبارسنجی قواعد و (۵) نگهداری قواعد می‌باشد. فرآیند پردازش قواعد نیز شامل پنج مرحله‌ی (۱) تشخیص رویداد، (۲) فعال‌سازی قواعد، (۳) انتخاب قاعده، (۴) کنترل شرط و (۵) اجرا می‌باشد [۷، ۵]. روش مورد استفاده برای انتخاب یا 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.

این مقاله در پنج بخش تنظیم شده است، در بخش دوم به تحلیل روش‌های زمان‌بندی اجرای قواعد در پایگاه داده پویا می‌پردازیم. در بخش سوم چارچوبی برای مقایسه و ارزیابی روش‌های زمان‌بندی موجود ارائه می‌نماییم. در این چارچوب پارامترهای ارزیابی متعددی نظیر میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی، توان عملیاتی، میزان سربار محاسباتی و بهره پردازشگر در نظر گرفته شده است. براساس این چارچوب روش‌های موجود با یکدیگر مقایسه و با تحلیل نقاط قوت و ضعف هریک، روش کارا تر معرفی می‌گردد. سپس در بخش چهارم، با هدف بهینه کردن فرآیند انتخاب قواعد در روش منتخب، به ارائه روشی برای تخمین احتمال وقوع قواعد می‌پردازیم. با افزودن پیمانه‌ی تخمین به روش منتخب نسخه‌ی جدیدی از آن را با نام

## ۲- تحلیل روش‌های زمان‌بندی اجرای قواعد در پایگاه داده پویا

در یک سیستم پایگاه داده‌ی پویا، در ابتدای امر، برنامه‌ی کاربردی در حال اجراست. تا زمانی که هیچ رویداد از پیش تعریف شده‌ای در سیستم رخ نداده است اجرای برنامه کاربردی ادامه پیدا می‌کند. به محض اینکه رویدادی در سیستم رخ داد، مجموعه‌ای از قواعد که با این رویداد مرتبط هستند، فعال شده و وارد لیست قواعد فعال می‌شوند. حال با توجه به مکانیزم انتخاب و روش زمان‌بندی، یکی از قواعد فعال موجود در لیست، انتخاب شده و بخش شرط آن، ارزیابی می‌شود. در صورت درست بودن شرط، بخش عمل قاعده‌ی مربوطه اجرا می‌شود. اگر اجرای این عمل موجب رخ دادن رویدادهای دیگری در سیستم شود، این قواعد جدید نیز به لیست قواعد فعال سیستم اضافه می‌شوند. ارزیابی و اجرای قواعد تا زمانی که لیست قواعد فعال خالی شود، ادامه پیدا می‌کند. پس از آن کنترل اجرا به برنامه‌ی کاربردی باز می‌گردد.

در یک سیستم مدیریت پایگاه داده‌ی پویا ایجاد ترتیب بین قواعد فعال در سیستم و ایجاد تقدم و تاخر در زمان اجرا بین آنها برای فرستادن به مرحله بعد در فرآیند پردازش قاعده را زمان‌بندی اجرای قواعد می‌نامیم. در ادامه، روش‌های ارایه شده برای زمان‌بندی اجرای قواعد در سیستم‌های پایگاه داده پویا را معرفی می‌نماییم.

### ۱-۲ روش اتفاقی

روش اتفاقی<sup>۱۱</sup> یکی از ساده‌ترین روش‌هایی است که تاکنون برای زمان‌بندی اجرای قواعد ارایه شده است [۵]. این روش اولین بار در سیستم‌های پایگاه داده پویای RPL و Ode [۵] پیاده سازی شده است. در روش زمان‌بندی اتفاقی هرگاه که در چرخه‌ی پردازش قواعد نیاز به انتخاب قاعده جدیدی برای اجرا باشد، از بین قواعد فعال در سیستم به صورت اتفاقی یکی برای اجرا انتخاب می‌شود. ویژگی مهم و ممتاز روش زمان‌بندی اتفاقی، سادگی پیاده‌سازی آن است. اما از نظر کارایی در مقایسه با سایر روش‌ها در پایین‌ترین جایگاه قرار می‌گیرد.

### ۲-۲ روش اولویت ایستا

استفاده از اولویت ایستا<sup>۱۲</sup> روش دیگری است که برای انتخاب قواعد و زمان‌بندی آنها استفاده می‌شود. در این روش به هر یک از قواعد یک

<sup>11</sup> Random Scheduling Approach

<sup>12</sup> Total Order (Static Priority)

عدد صحیح به عنوان اولویت نسبت داده می‌شود. به عنوان مثال در سیستم‌های [A] Ariel و [9] POSTGRES به هر یک از قواعد یک عدد در بازه‌ی [۱۰۰۰+ و ۱۰۰۰-] نسبت داده می‌شود. سپس در هنگام انتخاب قواعد برای اجرا، قاعده‌ای که در بین قواعد فعال موجود دارای کوچکترین اولویت باشد، برای اجرا انتخاب می‌شود. نسخه‌ی دیگر روش اولویت ایستا استفاده از گروه‌های اولویتی [5] است. در روش گروه‌های اولویتی برای سیستم،  $n$  گروه اولویتی تعریف می‌شود که با توجه به ساختار سیستم، هر یک از قواعد در یکی از گروه‌ها یا کلاس‌های اولویتی قرار می‌گیرد. به صورت کلی روش اولویتی ایستا بیشتر در سیستم‌هایی استفاده می‌شود که برخی قواعد هنگام فعال شدن در مقابل قواعد دیگر برای اجرا دارای اولویت‌های خاصی هستند و تقدم و تاخر اجرای آنها برای سیستم حیاتی است. از نظر کارایی روش اولویت ایستا فرق زیادی با روش زمانبندی اتفاقی ندارد.

### ۳-۲ روش مبتنی بر برچسب زمانی

روش زمانبندی مبتنی بر برچسب زمانی<sup>۱۳</sup> از جمله روش‌های کلاسیک زمانبندی است که برای زمانبندی اجرای قواعد در سیستم‌های پایگاه داده پویا نیز استفاده می‌شود [5]. این روش زمانبندی در سیستم SAMOS [۱۰] پیاده‌سازی شده و مورد استفاده قرار گرفته است. در این روش هر قاعده که در سیستم در حالت فعال قرار گرفت یک برچسب زمانی که مشخص کننده زمان فعال شدن آن قاعده است، همراه آن قرار می‌گیرد. در هنگام انتخاب و زمانبندی اجرای قواعد، قاعده‌ای که دارای کوچکترین برچسب زمانی باشد، برای اجرا انتخاب خواهد شد. این روش در مقایسه با دو روش قبلی دارای میانگین زمان پاسخگویی و زمان بازگشت کمتری است.

### ۴-۲ روش مبتنی بر الگوریتم نزدیکترین ضرب العجل

یکی از کامل‌ترین روش‌هایی که تاکنون برای زمانبندی اجرای قواعد در سیستم‌های پایگاه داده پویا ارائه شده، روش مبتنی بر الگوریتم EDF<sup>۱۴</sup> [۱۱] است. در این روش، هنگام انتخاب و زمانبندی اجرای قواعد، قاعده‌ای که دارای نزدیکترین ضرب العجل باشد، برای اجرا انتخاب می‌شود. براساس نحوه‌ی محاسبه‌ی اولویت قواعد (که مبتنی بر ضرب العجل قواعد است)، سه نسخه مختلف از روش مبتنی بر EDF طراحی شده است. اولی در حقیقت همان الگوریتم EDF صرف است. در دومی یک تغییر کوچک در الگوریتم EDF داده شده است به این ترتیب که اولویت قاعده‌ی پدر وقتی یک قاعده‌ی فرزند تولید می‌شود، تغییر می‌کند. در سومی اولویت قاعده‌ی پدر در هر لحظه براساس سه پارامتر تخمین زمان اجرای قاعده، تعداد قواعد فرزند فعلی و تخمین تعداد قواعد فرزند که در آینده تولید خواهد شد، محاسبه می‌شود.

<sup>۱۳</sup> First Come First Serve

<sup>۱۴</sup> Earliest Deadline First

بدیهی است این کار در صورتی عملی است که اطلاعات مورد نیاز آن موجود باشد. در صورت وجود چنین اطلاعاتی، کارایی این نسخه نسبت به دو نسخه دیگر بیشتر خواهد بود. این سه نسخه به ترتیب EDF<sub>PD</sub>، EDF<sub>DIV</sub> و EDF<sub>SL</sub> نامیده می‌شوند.

روش مذکور برای سیستم‌های پایگاه داده پویای بلادرنگ ارائه شده است که نتایج حاصل از شبیه‌سازی آن نشان می‌دهد کارایی قابل توجهی دارد [۱۲]. ضعف عمده روش EDF این است که این روش (۱) برای حالت خاص مجموعه قواعد دو سطحی [۱۲] ارائه شده است و برای قواعد با سطوح بالاتر هیچ راه کار خاصی ارائه ننموده است و (۲) مخصوص استفاده در سیستم‌های پایگاه داده پویای بلادرنگ است و در سایر سیستم‌های پایگاه داده پویا کاربردی ندارد.

### ۵-۲ روش اجرای موازی

یکی از جدیدترین روش‌هایی که در زمینه‌ی زمانبندی اجرای قواعد در سیستم‌های پایگاه داده پویا ارائه شده است، روش موازی اجرای قواعد و استفاده از چند ریسمان اجرایی<sup>۱۵</sup> است [۴،۵]. در این روش چند ریسمان اجرایی وجود دارد که قواعد به آنها نسبت داده می‌شود و به صورت موازی اجرا می‌گردند. روش اجرای موازی در سیستم پایگاه داده پویای FAR [۱۳] پیاده‌سازی شده است و بیشتر برای سیستم‌های مدیریت جریان کار<sup>۱۶</sup> مورد استفاده قرار می‌گیرد. از جمله نقطه ضعف‌های عمده روش اجرای موازی این است که (۱) به دلیل استفاده از ویژگی ریسمان‌های اجرایی، این روش در همه بستری‌های سخت‌افزاری و نرم‌افزاری قابل پیاده‌سازی نیست. (۲) این روش فقط تراکنش‌های نوع مستقل<sup>۱۷</sup> [۳] را پشتیبانی می‌کند و از تراکنش‌های فوری<sup>۱۸</sup> و تعویقی<sup>۱۹</sup> [۳] پشتیبانی نمی‌کند [۱۳]. بدلیل آنکه اصولاً ماهیت اجرای قواعد در حالت موازی با اجرای آنها در حالت سریال متفاوت است، این روش را نمی‌توان با سایر روش‌ها مورد مقایسه کمی قرار داد و ما در اینجا به ذکر نام و معرفی جزئی این روش بسنده می‌کنیم.

### ۶-۲ روش $E_x - SJF$

این روش مبتنی بر روش زمانبندی SJF<sup>۲۰</sup> است که یکی از روش‌های کلاسیک زمانبندی است که بر اصل ساده‌ی "هر تراکنشی که دارای زمان اجرای کوتاه‌تری است، زودتر باید اجرا شود"، استوار است. استفاده

<sup>۱۵</sup> Multi Threading

<sup>۱۶</sup> Work Flow Management

<sup>۱۷</sup> تراکنش مستقل (Independent) پس از کامل شدن تراکنش جاری و بصورت مستقل و مجزا اجرا خواهد شد.

<sup>۱۸</sup> پس از تولید تراکنش فوری (Immediate)، تراکنش جاری متوقف می‌شود و تراکنش فوری تولید شده، اجرا خواهد شد.

<sup>۱۹</sup> تراکنش تعویقی (Deferred) پس از تولید تا زمان کامل شدن تراکنش جاری منتظر خواهد بود و سپس به اجرا در خواهد آمد.

<sup>۲۰</sup> Shortest Job First

جدول ۱. تعریف پارامترهای ارزیابی

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

برای پیاده‌سازی روش‌های زمانبندی اجرای قواعد نیاز به محیطی است که بتواند عملکرد یک سیستم پایگاه داده پویا را شبیه‌سازی کند. در صورت وجود چنین محیط آزمایشی می‌توان روش‌های مختلف زمانبندی را در آن پیاده‌سازی و عملکرد هر یک از روش‌ها را در آن مشاهده نمود. برای این منظور در آزمایشگاه سیستم‌های هوشمند، یک محیط آزمایشگاهی به نام شبیه‌ساز سیستم پایگاه داده پویا طراحی و پیاده‌سازی شده است [۷]. این محیط شامل سه بخش اصلی است: واحد مدیریت اشیا داده، واحد مدیریت قواعد و واحد مدیریت تراکنش‌ها. واحد مدیریت اشیا داده که در حقیقت بخش پایگاه داده یک سیستم پایگاه داده پویا را شبیه‌سازی می‌کند، مدیریت کلیه امور مربوط به اقلام داده را برعهده دارد. به عبارت دیگر واحد مدیریت اشیا داده بخش داده‌های ایستا در محیط شبیه‌ساز را مدل می‌نماید. 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 مسوول نگهداری، مدیریت، زمانبندی و اجرای تراکنش‌های تولید شده توسط واحد مدیریت قواعد و کاربر می‌باشد. آزمایشات در سه حالت مختلف انجام شده است: حالت تراکنش‌های تعویقی، حالت تراکنش‌های فوری و حالت ترکیبی. در حالت تراکنش‌های تعویقی از پایگاه قاعده‌ای استفاده شده است که تراکنش‌های تولیدی در آن فقط از نوع تعویقی هستند. در حالت تراکنش‌های فوری، پایگاه قاعده استفاده شده فقط تراکنش‌های نوع فوری را پشتیبانی می‌کند و در حالت ترکیبی هم تراکنش‌های فوری و هم تراکنش‌های تعویقی پشتیبانی می‌شود. نتایج حاصل از شبیه‌سازی روش‌های زمانبندی اجرای قواعد، در حالت‌های تعویقی، فوری و ترکیبی به ترتیب در جداول ۲ و ۳ و ۴ نشان داده شده است. عددی که مقابل هر روش زمانبندی و پارامتر ارزیابی در هر خانه

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 پویا هستند، (مانند سیستم‌های پایگاه داده پویا) نیست و استفاده از آن موجب بهبود زمانبندی نخواهد شد.

در روش  $E_x - SJF$  [۷] هرگاه که در سیستم پایگاه داده پویا نیاز به انتخاب یکی از تراکنش‌های فعال در سیستم باشد، از بین آنها تراکنشی که دارای کمترین مقدار زمان اجراست انتخاب می‌شود. تفاوت بین روش SJF و روش  $E_x - SJF$  در نحوه محاسبه زمان اجرای تراکنش‌ها است. به دلیل ماهیت پویای تراکنش‌ها، در روش  $E_x - SJF$  نمی‌توان فاکتور انتخاب تراکنش‌ها را فقط زمان اجرای خود تراکنش قرار داد. زیرا هر تراکنشی که در یک سیستم پایگاه داده پویا اجرا می‌شود به صورت بالقوه مجموعه‌ای از قواعد پویا را در سیستم فعال می‌کند. تراکنش‌های حاصل از این قواعد نیز در صورت اجراء، مجموعه‌ای دیگر از قواعد را به صورت بالقوه فعال خواهند کرد و این عمل تا چند سطح ممکن است انجام پذیرد. لذا زمان اجرای واقعی یک تراکنش، علاوه بر زمان اجرای خودش، شامل زمان اجرای تراکنش‌های فوری و تعویقی که به صورت بالقوه توسط این قاعده تولید خواهند شد، نیز می‌باشد.

براساس نحوه محاسبه زمان اجرای واقعی هر قاعده، دو نسخه مختلف از روش  $E_x - SJF$  طراحی شده است. اولی فقط براساس زمان اجرای واقعی قواعد و دومی براساس زمان اجرای واقعی قواعد و فاکتور احتمال وقوع آنها کار می‌کند و به ترتیب  $E_x - SJF_{EXA}$  و  $E_x - SJF_{PRO}$  نامیده می‌شوند.

این روش در مجموع نسبت به روش‌های قبلی از کارایی بهتری برخوردار است. از جمله نقطه ضعف‌های این روش آن است که (۱) برای دو دسته سیستم‌های پایگاه داده پویای بلادرنگ و سیستم‌های دارای ویژگی اجرای موزای کاربردی ندارد و (۲) زمان اجرای واقعی قواعد را به دقت محاسبه نمی‌کند.

### ۳- چارچوب پیشنهادی برای ارزیابی و مقایسه‌ی روش‌های زمانبندی اجرای قواعد

در این بخش چارچوبی به منظور مقایسه‌ی روش‌های زمانبندی اجرای قواعد ارائه می‌دهیم. این چارچوب شامل پنج پارامتر ارزیابی میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی (هرچه انحراف معیار زمان پاسخگویی کمتر باشد تجمع زمان پاسخگویی به تراکنش‌ها حول نقطه میانگین بیشتر است و روش مورد استفاده کارایی بیشتری دارد)، توان عملیاتی، میزان زمان سربار محاسباتی به ازای هر تراکنش و بهره پردازشگر [۷] می‌باشد. تعریف پارامترهای ارزیابی فوق در جدول ۱ نشان داده است.

نوشته شده است، نشان دهنده رتبه آن روش زمانبندی در بین کلیه روش‌های زمانبندی موجود از نقطه نظر آن پارامتر ارزیابی است.

#### جدول ۲. نتایج حاصل از شبیه‌سازی روش‌های موجود برای زمانبندی

##### اجرای قواعد در حالت تعویقی

پارامترهای ارزیابی روش‌ها	میانگین زمان پاسخگویی	انحراف معیار زمان پاسخگویی	توان عملیاتی	سربار به ازای هر تراکنش	بهره پردازشگر
اتفاقی	۲	۳	۱	۱	۳
اولویت ایستا	۲	۳	۱	۱	۳
برچسب زمانی	۲	۲	۱	۱	۳
EDF <sub>PD</sub>	۲	۵	۲	۱	۳
EDF <sub>DIV</sub>	۲	۲	۲	۱	۳
EDF <sub>SL</sub>	۲	۴	۳	۱	۱
E <sub>x</sub> -SJF <sub>EXA</sub>	۱	۴	۲	۱	۲
E <sub>x</sub> -SJF <sub>PRO</sub>	۱	۱	۲	۱	۲

#### جدول ۳. نتایج حاصل از شبیه‌سازی روش‌های موجود برای زمانبندی

##### اجرای قواعد در حالت فوری

پارامترهای ارزیابی روش‌ها	میانگین زمان پاسخگویی	انحراف معیار زمان پاسخگویی	توان عملیاتی	سربار به ازای هر تراکنش	بهره پردازشگر
اتفاقی	۳	۳	۲	۳	۳
اولویت ایستا	۳	۳	۲	۳	۳
برچسب زمانی	۳	۳	۲	۱	۱
EDF <sub>PD</sub>	۲	۲	۱	۱	۲
EDF <sub>DIV</sub>	۲	۲	۱	۱	۲
EDF <sub>SL</sub>	۱	۱	۳	۲	۲
E <sub>x</sub> -SJF <sub>EXA</sub>	۲	۲	۳	۳	۲
E <sub>x</sub> -SJF <sub>PRO</sub>	۱	۱	۴	۲	۲

#### جدول ۴. نتایج حاصل از شبیه‌سازی روش‌های موجود برای زمانبندی

##### اجرای قواعد در حالت ترکیبی

پارامترهای ارزیابی روش‌ها	میانگین زمان پاسخگویی	انحراف معیار زمان پاسخگویی	توان عملیاتی	سربار به ازای هر تراکنش	بهره پردازشگر
اتفاقی	۴	۳	۱	۱	۱
اولویت ایستا	۳	۲	۳	۲	۱
برچسب زمانی	۳	۲	۳	۲	۱
EDF <sub>PD</sub>	۲	۱	۴	۲	۱
EDF <sub>DIV</sub>	۲	۱	۴	۲	۱
EDF <sub>SL</sub>	۲	۴	۵	۲	۱
E <sub>x</sub> -SJF <sub>EXA</sub>	۱	۱	۲	۲	۱
E <sub>x</sub> -SJF <sub>PRO</sub>	۱	۱	۲	۲	۱

زمان اجرای قواعد را نمی‌تواند به دقت محاسبه کند. در بخش بعدی راه‌حلی به منظور برطرف نمودن مشکل دوم ارائه می‌شود.

#### ۴- روش پیشنهادی برای تخمین احتمال وقوع قواعد

در بخش ۳ به این نتیجه رسیدیم که روش  $E_x - SJF_{PRO}$  براساس پنج پارامتر ارزیابی ارائه شده، مجموعاً دارای بهترین عملکرد در هر سه حالت تعویقی، فوری و ترکیبی می‌باشد. همچنین ملاحظه کردیم که یکی از معایب این روش، عدم محاسبه‌ی دقیق زمان اجرای قواعد می‌باشد. پیش از ارائه‌ی نسخه‌ی جدید روش  $E_x - SJF_{PRO}$  که مبتنی بر تخمین احتمال وقوع قواعد است، نحوه محاسبه زمان اجرای قواعد در روش  $E_x - SJF_{PRO}$  بررسی می‌شود [۷].

همانطور که در بخش ۲-۶ بیان شد، زمان اجرای واقعی یک قاعده، علاوه بر زمان اجرای خودش، شامل زمان اجرای قواعد فوری و تعویقی که به صورت بالقوه توسط این قاعده تولید خواهند شد، نیز می‌باشد. لذا بدیهی است برای محاسبه‌ی زمان اجرای واقعی قواعد، ابتدا باید تمام روابط موجود بین قواعد را بدست آورد. برای این منظور، در روش  $E_x - SJF_{PRO}$  ابتدا درخت اجرای قواعد تشکیل می‌شود. به کمک این درخت می‌توان پیش‌بینی کرد که در حین اجرای یک قاعده، چه قواعد دیگری فعال خواهند شد، اما همانطور که در بخش ۱ اشاره شد، فعال شدن یک قاعده، شرط لازم برای اجرای آن قاعده است، نه شرط کافی. شرط لازم و کافی برای اجرای یک قاعده برقراری هر دو شرط (۱) فعال شدن آن قاعده توسط یک قاعده‌ی دیگر (۲) درست بودن بخش شرط آن قاعده می‌باشد. لذا محاسبه‌ی زمان اجرای واقعی یک قاعده (پدر) دو پیشنهاد دارد: (۱) ساخت درخت اجرای آن قاعده (۲) محاسبه احتمال وقوع هر یک از قواعد فرزند (احتمال درستی شرط هریک از قواعد فرزند).

بخش شرط موجود در بدنه قواعد از عبارات شرطی، عبارات پرس و جوی پایگاه داده، فراخوانی رویه‌ها و توابع و ترکیب منطقی آنها تشکیل شده است. فرض کنید قاعده R وجود دارد به طوریکه بخش شرط آن به صورت  $[(A \cap B) \cup (C \cap D)]$  بیان شده است. A، B، C و D عبارات منطقی هستند. با فرض اینکه A، B، C و D عبارات منطقی مستقل از هم باشند داریم:

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

چون عبارات A، B، C و D مستقل از هم هستند، در نتیجه:

$$P[(A \cap B) \cup (C \cap D)] = P(A) * P(B) + P(C) * P(D) - P(A) * P(B) * P(C) * P(D) \quad (۱)$$

اگر احتمال درست بودن عبارات شرطی A، B، C و D به صورت قطعی وجود داشته باشد، می‌توان با جانشین کردن مقادیر آنها در رابطه (۱)، احتمال درست بودن شرط قاعده R را به دقت محاسبه نمود. ولی با توجه به اینکه احتمال درستی یک عبارت شرطی غالباً قبل از اجرای آن به صورت قطعی وجود ندارد، در روش  $E_x - SJF_{PRO}$  احتمال درستی هر

نتایج آزمایشات حاکی از آن است که در مجموع روش  $E_x - SJF_{PRO}$  از سایر روش‌ها کاراتر است و برای استفاده در طیف گسترده‌ای از سیستم‌های پایگاه داده پویا مناسب می‌باشد. البته این روش معایبی نیز دارد، از جمله این‌که (۱) برای دو دسته سیستم‌های پایگاه داده پویای بلادرنگ و سیستم‌های دارای ویژگی اجرای موازی کاربردی ندارد و (۲)

مرحله ی د، زمان اجرای واقعی قواعد است. همانطور که ملاحظه می شود، زمان اجرای واقعی قواعد  $R_2, R_1$  به ترتیب  $13/2$  و  $5/9$  است. همانطور که بیان شد، در روش  $E_x - SJF_{PRO}$  احتمال درست بودن هر یک از عبارات منطقی موجود در بخش شرط یک قاعده،  $1/2$  در نظر گرفته شده است که با واقعیت فاصله دارد، لذا بدیهی است که اگر ما بتوانیم به نحوی احتمال درست بودن عبارات منطقی یک شرط را با دقت بیشتری محاسبه کنیم، زمان اجرای واقعی قواعد با دقت بیشتری محاسبه می شود و در نهایت به افزایش کارایی روش زمانبندی منجر می گردد.

جدول ۵. قاعده ی  $R_2$  و فرزندان آن

<p>DEFINE R2 ON Stock.Updateprice IF (Stock.Policy=Low_risk) and (Stock.Price&lt; Stock.Initprice*E) DO Stock.Buy</p>	<p>DEFINE R7 ON Stock.Buy IF Stock.Price&lt; Bank_Table.Money) DO Stocks_InfoTable.Update And Bank_Table.Update</p>
<p>DEFINE R12 ON Bank_Table.Update IF Bank_Table.Money&lt; 1000000\$ DO Send.Message ("Warning", "Admin" ,"Not Enough money") Else DO E.Increase</p>	<p>DEFINE R16 ON E.Increase IF E&lt;1 DO E=E+.01*E  Hypothesis : 0&lt;E&lt;1</p>
<p>DEFINE R15 ON Send.message IF message.Type="Warning" and Message.Owner="Admin" DO Resend.message until receive answer</p>	

راه حل پیشنهادی، اضافه کردن پیمانه ی تخمین به زمانبند فعلی ( $E_x - SJF_{PRO}$ ) است که آن را به عنوان نسخه ی جدید روش  $E_x - SJF_{PRO}$  با نام  $E_x - SJF_{PRO} - V.1.8$  معرفی می نماییم. بدین ترتیب که در ابتدا زمان اجرای واقعی هر یک از قواعد، به روش  $E_x - SJF_{PRO}$  محاسبه می شود. سپس هر بار که یک قاعده ی فعال ( $R$ ) با توجه به مکانیزم زمانبندی مورد استفاده، از لیست قواعد فعال انتخاب می شود تا بخش شرط آن 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 رابطه ۳ محاسبه و ذخیره می شود.

$$P(R, LS_i) = \frac{\text{تعداد دفعاتی که } LS_i \text{ تاکنون ارزیابی شده و درست بوده است}}{\text{تعداد دفعاتی که } R \text{ تاکنون فعال شده است}}$$

این کار برای هر عبارت منطقی در بخش شرط یک قاعده تا زمانی تکرار می شود که آهنگ تغییرات احتمال درستی آن عبارت منطقی به حد مطلوب برسد (مثلاً  $0.000001$  و در حالت کلی  $\epsilon$ )، در این هنگام مقدار جدید، جایگزین مقدار پیش فرض اولیه ( $1/2$ ) می شود. بدیهی است هرچه مقدار  $\epsilon$  کوچک تر در نظر گرفته شود،  $P(R, LS_i)$  دقیق تر محاسبه می گردد. زمانیکه احتمال درستی همه ی عبارات منطقی یک شرط بهنگام شد، احتمال درستی آن شرط نیز بهنگام می شود. و در نهایت زمان اجرای واقعی قاعده ی  $R$  زمانی بهنگام می شود که احتمال

یک از عبارات شرطی به صورت یکسان  $1/2$  در نظر گرفته شده است. بنابراین احتمال درست بودن شرط قاعده  $R$  طبق رابطه ۱ خواهد شد:

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

این کار برای هر عبارت منطقی در بخش شرط یک قاعده تا زمانی تکرار می شود که آهنگ تغییرات احتمال درستی آن بدیهی است که در این روش هر چه عبارات شرطی به کار رفته در بدنه شرط، کمتر و ساده تر باشد، احتمال وقوع آن شرط بیشتر بوده و تأثیر زمان اجرای قاعده متبوع آن شرط بر زمان اجرای قاعده پدرش، بیشتر خواهد بود. با توجه به مطالب فوق، زمان اجرای واقعی هر قاعده در روش  $E_x - SJF_{PRO}$  از رابطه ۲ قابل محاسبه است.

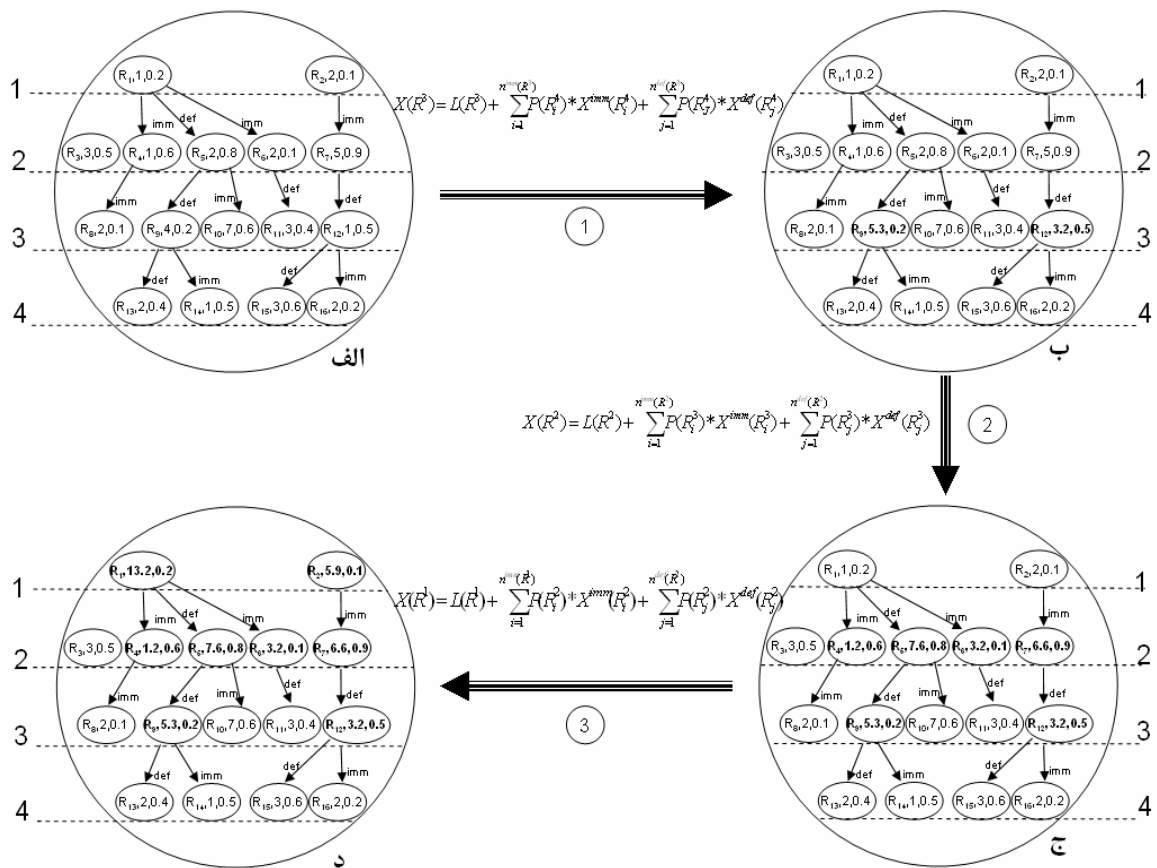
$$X(R^l) = L(R^l) + \sum_{i=1}^{n^{imm}(R^l)} P(R_i^{l+1}) * X^{imm}(R_i^{l+1}) + \sum_{j=1}^{n^{def}(R^l)} P(R_j^{l+1}) * X^{def}(R_j^{l+1}) \quad (2)$$

$$I = \{n-1, n-2, \dots, 1\}$$

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

در رابطه ۲ با فرض اینکه اجرای هر دستور در بخش عمل قواعد، یک واحد زمانی به طول بیانجامد، زمان اجرای هر قاعده به تنهایی،  $L(R)$  خواهد شد و چنانچه رابطه ۲ را از سمت برگ ها به سمت بالا در درخت اجرای قواعد اعمال نماییم، زمان اجرای هر قاعده محاسبه خواهد شد. حال نحوه محاسبه ی زمان اجرای واقعی قواعد در روش  $E_x - SJF_{PRO}$  با ذکر یک مثال بیان می شود. فرض کنید دو قاعده ی  $R_1$  و  $R_2$  در سیستم فعال شده اند (جدول ۵ قاعده ی  $R_2$  و فرزندانش را نشان می دهد). شکل ۱ - الف، درخت اجرای این دو قاعده را نشان می دهد. درخت اجرای قواعد، یک بار و در همان ابتدای شروع کار سیستم، بصورت ایستا برای تمامی قواعد ساخته می شود. این درخت، برای هر قاعده شامل تمام تراکنش های فوری و تعویقی است که در حین اجرای آن قاعده، فعال می شوند.

هر قاعده به صورت سه تایی  $(R_i, X_i, P_i)$  نشان داده شده است که  $R_i$  نام قاعده،  $X_i$  زمان اجرای قاعده و  $P_i$  احتمال درست بودن شرط  $R_i$  می باشد. مقدار  $X_i$  در مرحله الف زمان اجرای اولیه هر قاعده است، اما با سه مرحله اجرای رابطه ۲ بر روی درخت اجرای قواعد در شکل ۱، زمان اجرای واقعی هر یک از قواعد محاسبه خواهد شد. مقدار  $X_i$  در



شکل ۱. مراحل محاسبه زمان اجرای واقعی قواعد در روش  $E_x - SJF_{PRO}$

جدول ۶، نتایج آزمایش در نسخه جدید را در مقایسه با  $E_x - SJF_{PRO}$  نشان می‌دهد. همانطور که ملاحظه می‌شود، افزوده شدن پیمانه‌ی تخمین احتمال وقوع قواعد به  $E_x - SJF_{PRO}$ ، به ترتیب موجب کاهش و افزایش زمان محاسبه شده برای اجرای قواعد  $R_1$  و  $R_2$  شده است. بدیهی است زمان محاسبه شده‌ی اخیر به زمان واقعی اجرای قواعد مذکور، نزدیک‌تر است.

در ادامه روش  $E_x - SJF_{PRO} - V.1.8$  را با  $E_x - SJF_{PRO}$  در سه حالت تراکنش‌های تعویقی، تراکنش‌های فوری و ترکیبی مقایسه کردیم.

جدول ۷. درصد کارا تر بودن  $E_x - SJF_{PRO} - V.1.8$  نسبت به  $E_x - SJF_{PRO}$  در حالت‌های مختلف

توان	انحراف معیار	میانگین	پارامتر ارزیابی
عملیاتی <td>زمان پاسخگویی <td>زمان پاسخگویی <td>حالات مقایسه</td> </td></td>	زمان پاسخگویی <td>زمان پاسخگویی <td>حالات مقایسه</td> </td>	زمان پاسخگویی <td>حالات مقایسه</td>	حالات مقایسه
٪۹	٪۱۲/۶	٪۸	فوری
٪۱۴/۳	٪۳۳/۶	٪۱۸/۸	تعویقی
٪۱۲	٪۲۱/۴	٪۱۴/۲	حالت ترکیبی

جدول ۷ درصد بهبود زمانبندی در نسخه جدید را نسبت به  $E_x - SJF_{PRO}$  در هر یک از سه حالت آزمایش شده بر اساس سه پارامتر ارزیابی میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی و توان عملیاتی نشان می‌دهد. نتایج آزمایشات حاکی از آن است که با اضافه شدن پیمانه‌ی تخمین، زمان اجرای واقعی قواعد با دقت بیشتری (میزان این دقت متناسب با مقدار  $\mathcal{E}$  می‌باشد) نسبت به نسخه  $E_x - SJF_{PRO}$

درستی شرط قاعده‌ی  $R$  و زمان اجرای واقعی همه فرزندان بهنگام 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 بسیار ناچیز است)، زمان اجرای واقعی همه قواعد، با دقت قابل قبولی محاسبه می‌شود (که البته میزان این دقت، با تعیین مقدار  $\mathcal{E}$  قابل کنترل و تنظیم است). یکی از ویژگی‌های مهم شبیه‌ساز سیستم پایگاه داده پویا [۷] که در بخش ۳ معرفی شد، 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. لذا روش زمانبندی  $E_x - SJF_{PRO} - V.1.8$  را پیاده‌سازی و در شبیه‌ساز سیستم پایگاه داده پویا قرار دادیم. زمان اجرای واقعی قواعد  $R_1$  و  $R_2$  (شکل ۱) در روش  $E_x - SJF_{PRO} - V.1.8$  و با فرض  $\mathcal{E} = 0.001$  محاسبه شد.

جدول ۶. مقایسه‌ی زمان اجرای قواعد  $R_1$  و  $R_2$  در روش‌های

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

$R_2$	$R_1$	قاعده
		روش زمانبندی
۵/۹	۱۳/۲	$E_x - SJF_{PRO}$
۶/۸	۱۲/۱	$E_x - SJF_{PRO} - V.1.8$

## سپاس‌گزاری

بدینوسیله نویسندگان از پیشنهادات مفید و کمک‌های بی‌دریغ آقای محمدرضا آیت‌الله‌زاده شیرازی در تهیه این مقاله تشکر و قدردانی می‌نمایند.

## مراجع

- [1] A. Kotz Dittrich and E. Simon, "Active Database Systems: Expectations, Commercial Experience and Beyond", Active Rules in Database Systems, pp. 367-404, Springer-Verlag, Berlin, 1999.
- [2] K. Dittrich, S. Gatzju, A. Geppert, eds. (The ACT-NET Consortium), "The active database management system manifesto: a rule base of ADBMS features", Sigmod Record, 1996.
- [3] A. Vadua. "Rule Development for active database", PhD Thesis, CS Department, University of Zurich, 1999.
- [4] H. Theodore. "A survey of Active Database Systems", available in <http://www.doc.ic.ac.uk/~twh1/academic/papers/active.ps>, April, 1997.
- [5] E. N. Hanson and J. Widom. "An Overview of Production Rules in Database Systems". In the Knowledge Engineering Review, Vol.8, No.2, pp.121-143, 1993.
- [6] S. Gatzju, "Events in an Active Object-Oriented Database System", PhD thesis, University of Zurich, 1994.
- [7] R. Alesheykh, A. Abdollahzadeh, "Evaluation of Shortest Job First Approach for Rule Scheduling in Active Database Systems", in Proceedings of the 9th IASTED International Conference on Artificial Intelligence & Soft Computing (ASC'05), Benidorm, Spain, September 2005.
- [8] S. Potaminsto and M. Stonebraker. "The POSTGRES Rule System", in Active Database Systems: Triggers and Rules for Advanced Systems, Morgann Kaufmann Publishers, Sanfrancisco, CA, 1996.
- [9] A. Geppert, S. Gatzju, K. R. Dittrich, H. Fritschi, and A. Vadua. "Architecture and implementation of the active object-oriented database management system SAMOS", Technical Report 95.29, CS Department, University of Zurich, 1995.
- [10] J. Stankovic, M. Spuri, M.D. Natale and G.C. Buttazo. "Implications of Classical Scheduling Results for Real-Time Systems", IEEE Computer Society Press, Los Alamitos, CA, 1995.
- [11] R. M. Sivasankaran, J. A. Stankovic, D. Towsley, B. Purimetla and K. Ramamritham. "Priority Assignment in Real-Time Active Databases", The International Journal on Very Large Data Bases, Vol. 5, No. 1, January 1996.
- [12] S. Ceri, C. Gennaro, S. Paraboschi and G. Serazzi, "Effective Scheduling of Detached Rules in Active Database", IEEE Transaction on Knowledge and Data Engineering, Vol. 15, No.1, January/February 2003.
- [13] E. Hanson. "The design and implementation of the Ariel active database rule system", Technical Report UF-CIS-018-92, CIS Department, University of Florida, Gainesville, FL 32611, 1992.

محاسبه می‌شود که این امر منجر به بهبود روش زمانبندی  $E_x - SJF_{PRO}$  از نظر میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی و توان عملیاتی می‌شود. فرآیند تخمین احتمال وقوع قواعد در  $E_x - SJF_{PRO} - V.1.8$  به گونه‌ای است که سربار محاسباتی بر سیستم تحمیل نمی‌کند. لذا نسخه  $E_x - SJF_{PRO} - V.1.8$  در مقایسه با نسخه  $E_x - SJF_{PRO}$  از نظر میزان سربار محاسباتی و بهره‌پذیری پردازشگر برابری می‌کند.

## ۵- جمع‌بندی و کارهای آینده

در این مقاله، ابتدا چارچوبی برای مقایسه و ارزیابی روش‌های زمانبندی موجود ارائه شد. در این چارچوب پارامترهای ارزیابی متعددی نظیر میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی، توان عملیاتی، میزان سربار محاسباتی و بهره‌پذیری پردازشگر در نظر گرفته شده بود. براساس این چارچوب روش‌های موجود با یکدیگر مقایسه و با تحلیل نقاط قوت و ضعف هر یک، روش  $E_x - SJF_{PRO}$  بعنوان روش کارا تر معرفی گردید. روش  $E_x - SJF_{PRO}$  برای استفاده در طیف گسترده‌ای از سیستم‌های پایگاه داده مناسب است و فقط برای دو دسته سیستم‌های پایگاه داده‌ی پویای بلادرنگ و سیستم‌های دارای ویژگی اجرای موازی کاربرد ندارد. یکی از ویژگی‌های مهم این روش این است که هیچ محدودیتی بر روی تعداد سطوح پایگاه قاعده ندارد و از پایگاه‌های قواعد چند سطحی نیز پشتیبانی می‌کند. اما یکی از معایب این روش، نحوه برآورد احتمال وقوع قواعد فرزند در حین اجرای قاعده پدر می‌باشد که تقریبی است و نه دقیق. بدیهی است هرچه این احتمال با دقت بیشتری محاسبه شود، زمان واقعی اجرای تراکنش پدر با دقت بیشتری محاسبه می‌شود و در نهایت دقت روش زمانبندی برای انتخاب، بیشتر خواهد شد. سپس با هدف بهبود فرآیند انتخاب قواعد در روش  $E_x - SJF_{PRO}$ ، نسخه‌ی جدید آنرا که مبتنی بر تخمین احتمال وقوع قواعد می‌باشد، با نام  $E_x - SJF_{PRO} - V.1.8$  معرفی نمودیم. آزمایشات انجام شده، بهبود فرآیند انتخاب قواعد را در نسخه  $E_x - SJF_{PRO} - V.1.8$  از نقطه نظر میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی و توان عملیاتی نشان داد.

قصد داریم در آینده با افزودن پیمانه‌ی یادگیری (با استفاده از روش «آتوماتای یادگیر» و روش یادگیری Reinforcement) به روش  $E_x - SJF_{PRO} - V.1.8$ ، کارایی آن را افزایش دهیم، بدین ترتیب که مقدار  $\epsilon$  بطور خودکار توسط سیستم تعیین شود. سیستم مقدار  $\epsilon$  را با توجه به بازخوردی که از عملکرد سیستم دریافت می‌کند، تنظیم می‌نماید. عملکرد سیستم با توجه به وضعیت سیستم براساس پارامترهای ارزیابی تعیین می‌شود.