

- 
- <sup>۷</sup> First Come First Serve  
<sup>۸</sup> Earliest Deadline First  
<sup>۹</sup> Real-time systems  
<sup>۱۰</sup> Extended-Shortest Job First  
<sup>۱۱</sup> Extended-Shortest Job First-Exactly  
<sup>۱۲</sup> Extended-Shortest Job First-Probabilistic  
<sup>۱۳</sup> Extended-Shortest Job First-(based on)-Estimation-(using)-Learning Automata  
<sup>۱۴</sup> Unfavorable  
<sup>۱۵</sup> Stationary  
<sup>۱۶</sup> Non-Stationary  
<sup>۱۷</sup> Linear Reward Inaction  
<sup>۱۸</sup> Linear Reward Epsilon Penalty  
<sup>۱۹</sup> Stochastic



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. Buttazo, "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.

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://eeit.aut.ac.ir/~86131901/index.htm>

محمد رضا میدی: 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

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

<sup>2</sup> Event - Condition - Action

<sup>3</sup> Event - Condition Coupling

<sup>4</sup> Condition - Action Coupling

<sup>5</sup> Random

<sup>6</sup> Total Order (Static Priority)

## مراجع

- [۱] ع. رسولزادگان، ا. عبداللهزاده، "روش **Ex-SJF<sub>PRO</sub>-V.2.8**: جدید زمانبندی قواعد بر مبنای تخمین احتمال اجرایی قواعد در پایگاه داده پویا"، در مجموعه مقالات پانزدهمین کنفرانس مهندسی برق ایران، ۲۷-۲۵ اردیبهشت ۱۳۸۶، مرکز تحقیقات مخابرات ایران، تهران، ایران.
- [۲] 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.
- [۳] ع. رسولزادگان، ا. عبداللهزاده، "ارائه روشنی جدید برای تخمین احتمال وقوع قواعد در پایگاه داده پویا به منظور بهبود زمانبندی اجرایی قواعد در آن"، در مجموعه مقالات پانزدهمین کنفرانس سالانه انجمن کامپیوترا ایران، ۱۳۸۴، مرکز همایش‌های بین‌المللی صدا و سیما، تهران، ایران.
- [۴] 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.
- [۵] 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.
- [۶] 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.
- [۷] 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.
- [۸] R. Alesheykh, "An Effective Rule Selection Approach in Active Database Systems", MSc Thesis, Amirkabir University of Technology (Tehran PolyTechnic), 2005.
- [۹] 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.
- [۱۰] 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.
- [۱۱] A. Vadua, **Rule Development for active database**, PhD Thesis, CS Department, University of Zurich, 1999.
- [۱۲] 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.
- [۱۳] R.M. Sivasankaran, J.A. Stankovic, D. Towsley, B. Purimetla, K. Ramamritham, "Priority Assignment in Real-Time Active Databases", The International



نسخه جدید ( $Ex\text{-}SJF_{ESTLA}$ ) نسبت به نسخه‌های ارائه شده پیشین، 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 آنهاستی اجرا می‌شوند که شرط‌شان در لحظه 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 قواعد 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 بعد 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. کاری که در روش فرایند زمان‌بندی قواعد در روش اخیر نسبت به روش  $Ex\text{-}SJF_{PRO-V.2.8}$  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 روش زمان‌بندی موجود ( $Ex\text{-}SJF_{PRO-V.2.8}$ ) در صدد بهبود فرایند زمان‌بندی قواعد فعل به کمک اتماتیابی یادگیر امکان‌پذیر شده است. سربار محاسباتی فرایند زمان‌بندی قواعد در روش اخیر نسبت به روش  $Ex\text{-}SJF_{PRO-V.2.8}$  فقط در حد بهنگام سازی احتمال درستی شرط برخی از قواعد فعل، طبق (۷) است که بسیار ناچیز و قابل اغماض می‌باشد.

از جمله فعالیت‌های تحقیقاتی که در راستای تکمیل پژوهش جاری می‌توان انجام داد، ارائه یک پست‌استاندارد 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 یا ارائه روش‌های جدید، از دیگر فعالیت‌های پژوهشی مرتبط پیشنهادی برای آینده است.

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

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

جدول (۸) درصد کارآثر بودن روش  $E_x\text{-}SJF_{ESTLA}$  را نسبت به روش  $E_x\text{-}SJF_{PRO-V.2.8}$  (کارآثرین 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 کارآثر بودن روش  $E_x\text{-}SJF_{ESTLA}$  نسبت به روش  $V.2.8$

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

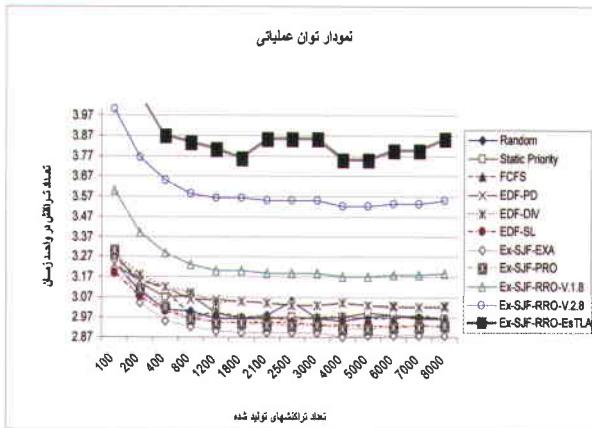
همانطور که مشاهده می‌شود با بهبود فرایند زمان‌بندی قواعد فعل در



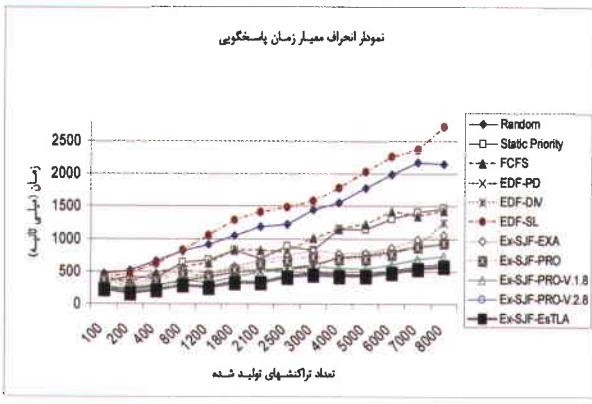
$$\begin{aligned}
 S(k) &= \sum_{j=1}^3 \sum_{i=1}^5 W_i * Rnk_{i,j}(k) \\
 &\quad k \in \text{Rule Scheduling Approaches Set} \\
 &\quad W_i \text{ is the weight of } i\text{th evaluation metric} \\
 &\quad j=1, 2 \text{ and } 3 \text{ indicates deferred, immediate and composite state, respectively} \\
 &\quad Rnk_{i,j}(k) \text{ is the rank of approach } k \text{ in } j\text{th state based on } i\text{th evaluation metric} \\
 &\quad S(k) \text{ is the total score of approach } k \text{ among all approaches}
 \end{aligned} \tag{10}$$

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  $E_x\text{-SJF}_{EXA}$  دارای کمترین توان عملیاتی است. دو روش  $EDF_{SL}$  و  $EDF_{DIV}$  دارای توان عملیاتی بیشتری است. دو روش  $E_x\text{-SJF}_{PRO}$  و  $E_x\text{-SJF}_{PRO-V.2.8}$  هستند. روش‌های  $E_x\text{-SJF}_{EXA}$  از این حیث در رتبه‌های بهتری نسبت به سه روش پیش قرار دارند به طوریکه روش  $E_x\text{-SJF}_{PRO-V.1.8}$  موجب افزایش ۹ درصدی توان عملیاتی روش  $E_x\text{-SJF}_{PRO}$  می‌شود و روش  $E_x\text{-SJF}_{PRO-V.2.8}$  نیز موجب افزایش ۱۱ درصدی توان عملیاتی روش  $E_x\text{-SJF}_{PRO-V.1.8}$  می‌شود. اما این روش‌ها نیز خود از نظر این معیار ارزیابی بعد از چهار روش  $E_x\text{-SJF}_{ESTLA}$ , «اولویت ایستا», FCFS و «انتخاب اتفاقی» قرار می‌گیرند. به طوریکه توان عملیاتی روش  $E_x\text{-SJF}_{ESTLA}$ , ۷ درصد بهتر از توان عملیاتی روش  $E_x\text{-SJF}_{PRO-V.2.8}$  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  $E_x\text{-SJF}_{ESTLA}$  و

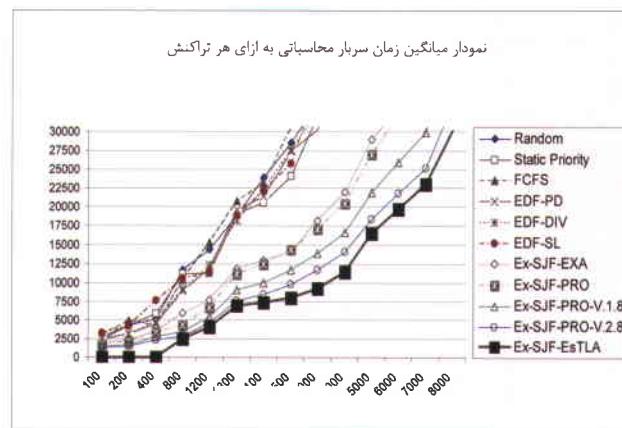


شکل (۸): نمودار توان عملیاتی روش‌های زمانبندی در حالت فوری

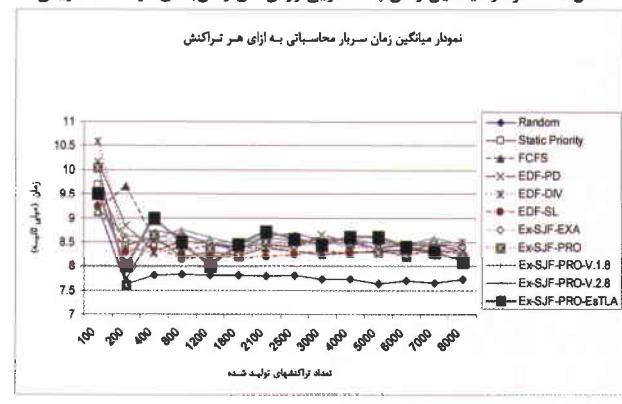


شکل (۱۰): نمودار انحراف معیار زمان پاسخگویی روش‌های زمانبندی در حالت ترکیبی

کلیه روش‌ها در حالت تعویقی هستند. روش‌های  $E_x\text{-SJF}_{PRO-V.1.8}$ ,  $E_x\text{-SJF}_{EXA}$  و  $E_x\text{-SJF}_{PRO}$  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  $E_x\text{-SJF}$  نسبت به سایر روش‌ها به این نتیجه می‌رسیم که استفاده از روش  $E_x\text{-SJF}_{ESTLA}$  موجب کاهش ۹/۸ درصدی میانگین زمان پاسخگویی نسبت به روش  $E_x\text{-SJF}_{PRO-V.2.8}$  و استفاده از روش  $E_x\text{-SJF}_{PRO-V.2.8}$  موجب کاهش ۱۵/۸ درصدی میانگین زمان پاسخگویی نسبت به روش  $E_x\text{-SJF}_{PRO-V.1.8}$  و  $V.1.8$  و استفاده از روش  $E_x\text{-SJF}_{PRO-V.1.8}$  موجب کاهش ۱۸/۸ درصدی میانگین زمان پاسخگویی نسبت به روش  $E_x\text{-SJF}_{PRO}$  و  $E_x\text{-SJF}_{ESTLA}$  در زمینه SJF<sub>EXA</sub> می‌شود (عملکرد دو نسخه روش در زمینه SJF<sub>EXA</sub> میانگین زمان پاسخگویی در حالت تعویقی بسیار نزدیک به هم بوده ولی روش  $E_x\text{-SJF}_{PRO}$  اندکی بهتر از روش  $E_x\text{-SJF}_{EXA}$  و  $E_x\text{-SJF}_{PRO}$  در حالیست که دو روش اخیر ( $E_x\text{-SJF}_{EXA}$  و  $E_x\text{-SJF}_{PRO}$ ) نیز به نوبه‌ی خود موجب کاهش ۴۳ 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  $E_x\text{-SJF}_{ESTLA}$ ,  $E_x\text{-SJF}_{PRO-V.2.8}$ ,  $SJF_{PRO-V.2.8}$ ,  $E_x\text{-SJF}_{PRO-V.1.8}$  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

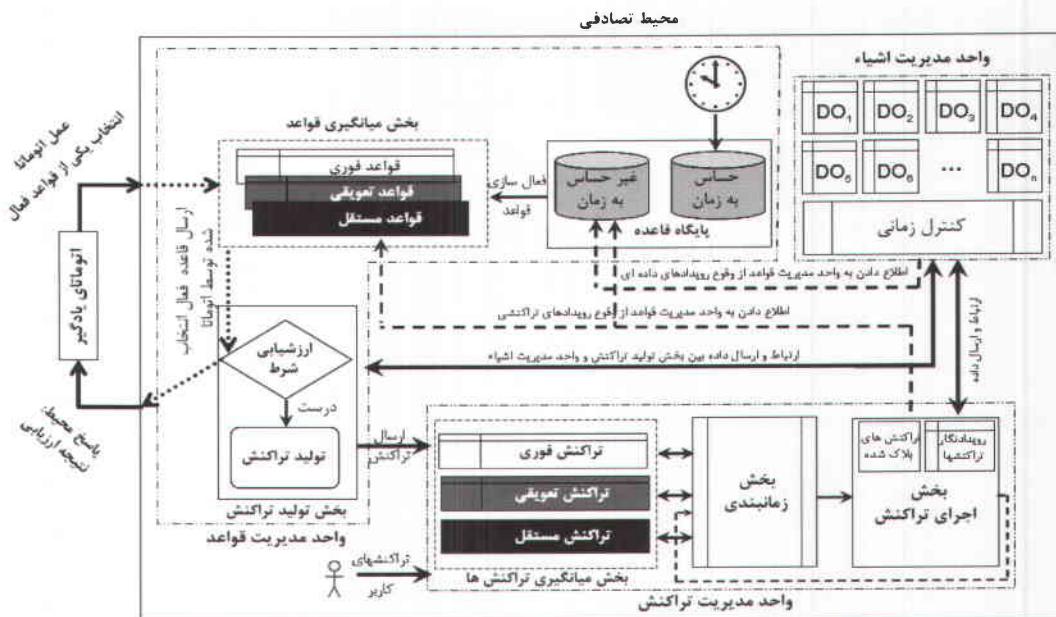


شکل (۷): نمودار میانگین زمان سریار محاسباتی روش‌های زمانبندی در حالت تعویقی



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





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

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

ردیف	نام روش	پارامترهای ارزیابی				
		کمترین زمان	متوسط زمان	میانگین زمان	متوسط میانگین زمان	زمان میانگین
۱	Random	۱	۳	۵	۷	۹
۱	Static Priority	۱	۲	۴	۶	۸
۱	FCFS	۱	۲	۴	۶	۸
۱	EDF <sub>PD</sub>	۱	۲	۵	۵	۷
۱	EDF <sub>DIV</sub>	۱	۲	۵	۵	۷
۱	EDF <sub>SL</sub>	۱	۲	۶	۵	۷
۱	E <sub>x</sub> -SJF <sub>EXA</sub>	۱	۲	۳	۴	۵
۱	E <sub>x</sub> -SJF <sub>PRO</sub>	۱	۲	۳	۴	۵
۱	E <sub>x</sub> -SJF <sub>V.1.8</sub>	۱	۲	۲	۳	۴
۱	E <sub>x</sub> -SJF <sub>V.2.8</sub>	۱	۲	۱	۲	۳
۱	E <sub>x</sub> -SJF <sub>ESTLA</sub>	۱	۲	۱	۱	۲

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

ردیف	نام روش	پارامترهای ارزیابی				
		کمترین زمان	متوسط زمان	میانگین زمان	متوسط میانگین زمان	زمان میانگین
۳	Random	۱	۳	۶	۵	۹
۳	Static Priority	۱	۳	۶	۵	۸
۳	FCFS	۱	۳	۵	۵	۷
۳	EDF <sub>PD</sub>	۱	۳	۵	۵	۷
۳	EDF <sub>DIV</sub>	۱	۴	۸	۵	۱۰
۱	EDF <sub>SL</sub>	۱	۵	۷	۵	۱۰
۲	E <sub>x</sub> -SJF <sub>EXA</sub>	۱	۴	۷	۴	۱۰
۲	E <sub>x</sub> -SJF <sub>PRO</sub>	۱	۴	۷	۴	۱۰
۲	E <sub>x</sub> -SJF <sub>V.1.8</sub>	۱	۳	۳	۳	۱۰
۲	E <sub>x</sub> -SJF <sub>V.2.8</sub>	۱	۲	۲	۲	۱۰
۲	E <sub>x</sub> -SJF <sub>ESTLA</sub>	۱	۱	۱	۱	۱۰

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

ردیف	نام روش	پارامترهای ارزیابی				
		کمترین زمان	متوسط زمان	میانگین زمان	متوسط میانگین زمان	زمان میانگین
۳	Random	۲	۵	۵	۵	۹
۳	Static Priority	۲	۵	۵	۵	۸
۱	FCFS	۱	۲	۵	۵	۷
۲	EDF <sub>PD</sub>	۱	۱	۴	۴	۷
۲	EDF <sub>DIV</sub>	۱	۱	۴	۴	۷
۲	EDF <sub>SL</sub>	۲	۴	۳	۳	۷
۲	E <sub>x</sub> -SJF <sub>EXA</sub>	۵	۴	۴	۴	۷
۲	E <sub>x</sub> -SJF <sub>PRO</sub>	۲	۴	۳	۳	۷
۲	E <sub>x</sub> -SJF <sub>V.1.8</sub>	۳	۲	۲	۲	۷
۲	E <sub>x</sub> -SJF <sub>V.2.8</sub>	۲	۲	۱	۲	۷
۲	E <sub>x</sub> -SJF <sub>ESTLA</sub>	۲	۲	۱	۱	۷

عددی که مقابل 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. رتبه‌بندی نهایی روش‌ها بر طبق (۱۰) محاسبه می‌شود. شایان ذکر است که رتبه نهایی روشی بهتر خواهد بود، که مقدار  $S(k)$  متناظر باشد.

شکل‌های (۷) الی (۱۰) چند نمونه از نمودارهای ارزیابی روش‌های زمان بندی اجرای قواعد در حالات مختلف، براساس پارامترهای ارزیابی پیش‌گفته را نشان می‌دهند. شکل (۷) روش‌های مختلف، براساس میانگین زمان پاسخگویی در حالت تعویقی در حالات فعالیتی در حالت فوری می‌بردارد 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<sub>x</sub>-SJF<sub>PRO</sub> و E<sub>x</sub>-SJF<sub>ESTLA</sub> V.2.8 دارای کمترین میانگین زمان پاسخگویی بین

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 مسؤول نگهداری، مدیریت، زمان‌بندی و اجرای تراکنش‌های تولید شده توسط واحد مدیریت قواعد و کاربر می‌باشد. شکل (۶) معماری ADSS را نشان می‌دهد (بخش اتماناتی یادگیر مختص روش  $E_{x-SJF_{estLA}}$  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 روش‌های 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 فعل 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 جهت پوشش کلیه‌ی حالات و ویژگی‌های قواعد پویا در سیستم‌های واقعی در نظر گرفته شده‌اند.

## ۴- مقایسه و ارزیابی روش‌های زمان‌بندی قواعد

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. معماري، 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]

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

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  $E_{x-SJF}$ ، در حالت‌های تعویقی، فوری و ترکیبی به ترتیب در 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 قواعد 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



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

ج- برای هر

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

شكل (۵) چرخه‌ی بهنگام سازی احتمال درستی شرط قواعد توسط اتوماتای یادگیر را نشان میدهد. شایان ذکر است که با وجود حداقل دو قاعده فعال، چرخه 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 ( $\sigma_{t_i}$ ) 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 آهنگ تغییرات آن کمتر از ۴ گردد. متعاقباً زمان اجرای یک قاعده زمانی بهنگام می‌شود که احتمال درستی شرط آن، نهایی و نیز زمان اجرای کلیه‌ی فرزندان آن بهنگام شده باشد. در آغاز هر بازه‌ی زمانی ( $\sigma_{t_{i+1}}$ ) 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 ( $\sigma_{t_i}$ )، 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  $\sigma_k$  در مقیاس ۱

$$p_{R_{i,2}}(\sigma_k)$$

احتمال انتخاب اقدام نام (قاعده  $R_i$ ) محاسبه شده توسط اتوماتای یادگیر در پایان  $\sigma_k$

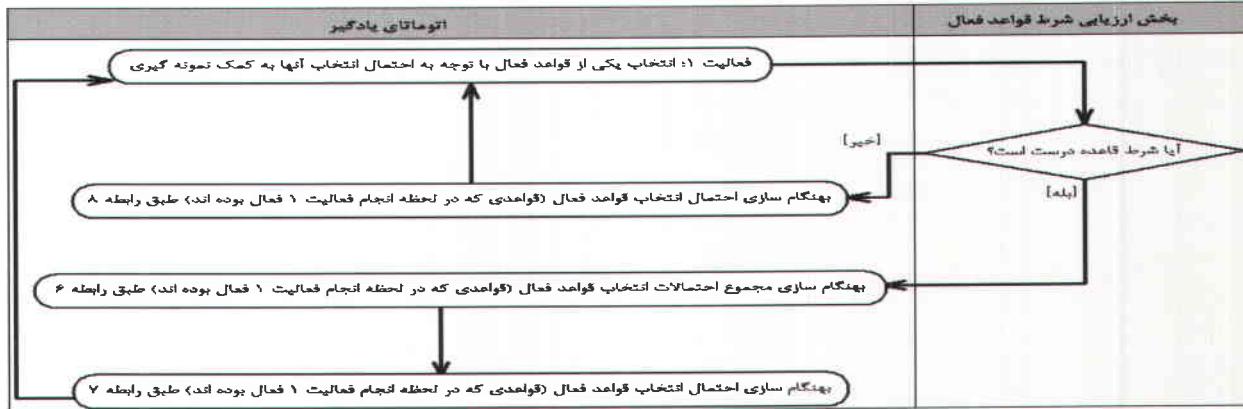
$$\phi(t) = \min(k/k', 1) \quad \exists k' \geq 25$$

احتمال انتخاب اقدام نام (قاعده  $R_i$ ) محاسبه شده توسط اتوماتای یادگیر در پایان  $\sigma_k$

$$\sigma_k$$

$$p_{R_i}(\sigma_{k+1}) = \frac{\phi(t) * p_{R_{i,2}}(\sigma_k) + p_{R_{i,1}}(\sigma_k)}{\phi(t) + 1} \quad (9)$$

احتمال انتخاب اقدام نام (قاعده  $R_i$ ) در پایان  $\sigma_k$  توسط اتوماتای یادگیر و در آغاز  $\sigma_{k+1}$



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 اقدامات اتوماتا صورت پذیرد که در ادامه به شرح آن خواهیم پرداخت. اگر تعداً قواعد،  $n$  باشد، مجموعه اقدامات اتوماتا در حالت کلی،  $v = \{\alpha_1, \dots, \alpha_n\}$  خواهد بود. یعنی بازی از قاعده، یک اقدام تعریف می‌شود. لذا با توجه به مطلب پیش‌گفته، بدیهی است که در هر لحظه مجموعه اقدامات اتوماتا زیر مجموعه‌ای از مجموعه فوق خواهد بود.  $V_i$  نشان‌دهنده نامین زیرمجموعه اقدامهای 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 تکیک ارائه شده در روش  $Ex-SJF_{PRO-V.2.8}$  محاسبه می‌شود، لذا با فرض اینکه  $R_i(0)$  نشان‌دهنده مجموع احتمالات درستی شرط قواعد در آغاز باشد. با توجه به رابطه (۵) احتمال انتخاب هر اقدام در آغاز در مقیاس ۱ طبق رابطه 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  $(V, F)$ ، فعال باشند، احتمال انتخاب اقدامها به روش زیر بهنگام می‌شوند:

$$K(k) = \sum_{\substack{q \\ \alpha_q \in v(k)}} p_q(k) \quad (6)$$

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

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

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



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

$\alpha_1$ ۱۰۰۰ (دقیقه)		$\alpha_2$ ۱۰۰۰ (دقیقه)		$\alpha_3$ ۱۰۰۰ (دقیقه)		بازه‌های زمانی اقلام داده
مدت زمان اخذ (بر حسب دقیقه)	مقدار اخذ شده	مدت زمان اخذ (بر حسب دقیقه)	مقدار اخذ شده	مدت زمان اخذ (بر حسب دقیقه)	مقدار اخذ شده	
۱۰۲	۲۰	...	۸	۲۲	۱۰	$DI_A$
۵۰	۷۱	...	۲۸	۴۷	۵	
⋮	⋮	⋮	⋮	⋮	⋮	
۳۵۲	۱۱	...	۹۵	۸۶	۳۹	
۵۲	۱	...	۵	۴	۱۵	
⋮	⋮	⋮	⋮	⋮	⋮	

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

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

که  $r$  تعداد اقدامهای اتوماتا،  $a$  پارامتر پاداش و  $b$  پارامتر جریمه می‌باشدند. لذا شکل عمومی الگوریتم یادگیری بصورت زیر است. اگر در گام  $n$  اقدام  $\alpha_i$  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، اتوماتای یادگیر  $L_{RP}$  نامیده می‌شود. زمانیکه  $b$  مساوی با صفر باشد اتوماتای یادگیر  $L_{RI}$ <sup>۱۷</sup> نامیده می‌شود. اگر  $a < b$  باشد، اتوماتای یادگیر  $L_{REP}$ <sup>۱۸</sup> نامیده می‌شود.

### ۲-۳- نحوه بکارگیری اتوماتای یادگیر در فرایند تخمین احتمال اجرای قواعد

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

محیطی که اتوماتای 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 فعال انتخاب شده، درست باشد، پاسخ محیط ۱ و در صورتیکه نادرست باشد، پاسخ محیط ۰ خواهد بود). اقدام اتوماتای 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

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

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

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 اتوماتا، احتمال اقدامهای آن با هم برابر و مساوی  $\frac{1}{r}$  می‌باشدند. (که  $r$  تعداد اقدامهای اتوماتا می‌باشد)

اگر اتوماتای یادگیر در تکرار  $n$ ام، یک اقدام خود مانند  $\alpha_i$  را انتخاب کند و یک پاسخ مطلوب از محیط دریافت نماید،  $p_i(n)$  افزایش و احتمال سایر اقدامها کاهش می‌یابد. بالعکس در صورت نامطلوب بودن پاسخ دریافتی از محیط، احتمال اقدام  $\alpha_i$  کاهش و احتمال سایر اقدامهای اتوماتا 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

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

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

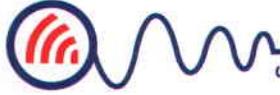
ب- پاسخ نامطلوب از محیط

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

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

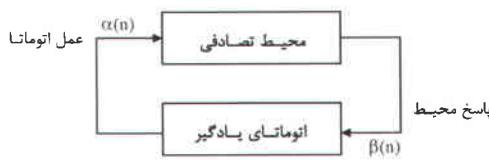
تابع  $f_j$  و  $g_j$  دوتابع غیر منفی هستند که بترتیب توابع پاداش و جریمه نامیده می‌شوند. در یک الگوریتم یادگیری تقویتی خطی (در اتوماتایی با چند اقدام) توابع  $f_j$  و  $g_j$  بصورت زیر تعريف شده‌اند

[۱۰]



فعالی که نتیجه‌ی ارزیابی بخش شرط‌شان مثبت بوده، قاعده‌ای 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 (۱)، 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 ۲۰ 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 ۲۰ 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 قalemدادی *D<sub>I</sub>* نشان می‌دهیم. محاسبات و بهنگام سازی‌ها را در زمان بیکاری سیستم انجام می‌دهد. محاسبات و بهنگام سازی‌ها 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 آهنگ تغییرات زمان اجرای قواعد به حد مطلوب برسد (۴). بدیهی است که تعیین اندازه‌ی *E* به میزان اهمیت دقت زمان اجرای قواعد در سیستم بستگی دارد.



شکل (۴): ارتباط بین محیط و آتماتای یادگیر

محیط را می‌توان توسط ستایی  $\{v, \beta, c\}$  نشان داد که در آن  $\{v, \beta, c\} = \{\alpha_1, \alpha_2, \dots, \alpha_r, \beta_1, \beta_2, \dots, \beta_m, P\}$  مجموعه خروجی‌ها و  $P$  مجموعه احتمال‌های جرمیه می‌باشد. هرگاه  $\beta$  مجموعه دو عضوی باشد، محیط از نوع  $P$  می‌باشد. در چنین محیطی  $\beta_1 = 1$  به عنوان جرمیه 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  $S$ ،  $\beta(n)$  متغیر تصادفی در فاصله  $[0, 1]$  است. احتمال این است که عمل  $\alpha_i$  نتیجه نامطلوب  $\beta$  داشته باشد. در محیط ایستا<sup>۱۵</sup> مقادیر  $c$  بدون تغییر می‌مانند، حال آنکه در محیط غیرایستا<sup>۱۶</sup> این مقادیر در طی زمان تغییر می‌کنند.

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

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  $Ex-SJF_{PRO-V.2.8}$  عمل می‌نماید. اما برای زمانبندی قواعد فعال، دارای مکانیزم زمانبندی متفاوتی بر مبنای آtomاتای یادگیر است. بعبارت دیگر در روش جدید، ملاک زمانبندی قواعد فعال، بر خلاف روش  $Ex-SJF_{PRO-V.2.8}$ ، زمان اجرای قواعد نیست. بلکه احتمال درستی بخش شرط آنهاست و آtomاتای 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 آtomاتای یادگیر در زمانبندی قواعد فعال، به تعریف آtomاتای یادگیر و مفاهیم مرتبط می‌پردازیم.

### ۱-۳ آtomاتاهای یادگیر

آtomاتای یادگیر [۱۰، ۱۹]، ماشینی 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 آtomاتا داده می‌شود و آtomاتا از این پاسخ در انتخاب عمل بعدی تأثیر می‌گیرد. هدف نهایی این است که آtomاتا یادگیرد تا از بین اعمال خود بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداقل برساند. کارکرد آtomاتای یادگیر در تعامل با محیط، در شکل (۴) مشاهده می‌شود.

### ۳-۱<sup>۱۳</sup> Ex-SJF<sub>ESTLA</sub> معرفی روش جدید زمانبندی

این روش نیز مانند روش‌های  $Ex-SJF_{PRO-V.1.8}$  و  $Ex-SJF_{PRO-V.2.8}$  بر مبنای تخمین احتمال اجرای قواعد طراحی شده است. بعبارت دیگر در روش جدید با اضافه کردن یک آtomاتای یادگیر به کارترین روش موجود [۱]،  $Ex-SJF_{PRO-V.2.8}$ ، درصد افزایش کارایی فرایند زمانبندی قواعد از بُعد پارامترهای ارزیابی تعريف شده، برآمدایم. 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 بکسانی برای هر دو زمانبندی مذکور می‌باشند. بعنوان مثال در روش  $Ex-SJF_{PRO-V.2.8}$ ، 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 اجرا را داشته باشد.



جدول (۱): بخش شرط چند قاعده نمونه

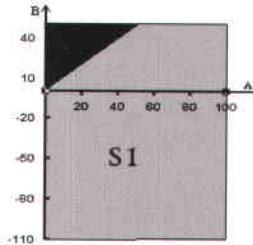
نام	بخش شرط قاعده
$R_1$	$DI_A > DI_B$
$R_2$	$DI_C = \{ \text{فولاد} \text{ OR } \text{سیمان} \}$
$R_3$	$DI_B = 20 \text{ AND } DI_D \leq DI_A$
$R_4$	$DI_A < 20 \text{ AND } DI_C = \{ \text{برنج} \}$

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

$0 < DI_A < 100$	$-110 < DI_B < 50$	$-2000 < DI_D < 50$
$DI_C = \{ \text{فولاد}, \text{سیمان}, \text{برنج}, \text{طلاء}, \text{برنز}, \text{فرش}, \text{نقره}, \text{پرتفال}, \text{کفشن} \}$		

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 تامی 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 داده  $B$  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:

شکل (۳): محاسبه احتمال درستی شرط قاعده  $R_1$ 

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

که با مقدار گذاری در (۴) خواهیم داشت:

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

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 می‌تواند کار خود را آغاز کند و زمان‌بند نیز در هر لحظه از بین قواعد بزرگتر از قلم داده  $A$  ( $DI_A$ ) باشد.

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}
```

$Condition\_Probability(R \rightarrow C)$

```
{ For Each LS of R → C That LS → fixed=False Do
```

$LS\_Probability(R \rightarrow C \rightarrow LS)$

$IF R \rightarrow C \rightarrow LS \rightarrow Fixed=False Then$

$R \rightarrow C \rightarrow Fixed=False$

$Return(R \rightarrow C \rightarrow Fixed)$

$R \rightarrow C \rightarrow Fixed=True$

$Calc\_Execution\_Time(R \rightarrow parents)$

$Return(Calculate_Probability(R \rightarrow C)) \}$

"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) \ }

$Rule\_Scheduling (ReadyToExecute\_Rule\_Base) =$

{  $\forall R_i \in ReadyToExecute\_Rule\_Base,$

$\{ (R_i, ExecT_i), \dots, (R_n, ExecT_n) \}, R_i \text{ Selected } | \{ R_i \in$

$ReadyToExecute\_Rule\_Base | 1 \leq i \leq n,$

$\forall R_i \in ReadyToExecute\_Rule\_Base | 1 \leq i \leq n,$

$R_i \rightarrow ExecutionTime \leq R_j \rightarrow ExecutionTime. \}$

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

در روش  $Ex-SJF_{PRO}-V.1.8$

#### ۴-۵-۲ روش زمان‌بندی Ex-SJF<sub>PRO</sub>-V.2.8

این روش نیز مانند روش Ex-SJF<sub>PRO</sub>-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 انتخیل 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  $A$  ( $DI_A$ ) بزرگتر از قلم داده  $B$  ( $DI_B$ ) باشد.



$X^{imm} (R_i)$	زمان اجرای آمین تراکنش فوری حاصل از $R$
$X^{def} (R_j)$	زمان اجرای زامین تراکنش تعویقی حاصل از $R$
$L(R)$	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 آمین قاعده فوری حاصل از $R$
$P(R_j)$	احتمال درستی شرط زامین قاعده تعویقی حاصل از $R$

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 خواهد شد.

## ۲-۵-۲ روش زمان‌بندی Ex-SJF<sub>PRO</sub>

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 شده، طبق (۱) خواهد شد:

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

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

## ۳-۵-۲ روش زمان‌بندی Ex-SJF<sub>PRO-V.1.8</sub>

در این روش که پیش از این توسط خودمان ارائه شده است، یک پیمانه تخمین به روش Ex-SJF<sub>PRO</sub> اضافه کرده‌ایم [۴، [۵]. بدین ترتیب که در ابتدا زمان اجرای هر قاعده، همانند روش Ex-SJF<sub>PRO</sub> 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.

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

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 (متلاً  $0.001$  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 فرزندانش 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 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  $E_x-SJF_{PRO-V.1.8}$  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 اجرا می‌شود به صورت بالقوه مجموعه‌ای از قواعد پویا (فرزنده) را در سیستم فعال می‌کند. تراکنش‌های حاصل از این قواعد نیز در صورت اجرا، مجموعه‌ای دیگر از قواعد را به صورت بالقوه فعال خواهند کرد و این عمل تا جند 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 ممکن‌بزیر است. اما انجام آن به دلیل در بی داشتن سربار زمانی زیاد، منجر به یک Ex-SJF روش زمان‌بندی ناکارا خواهد شد. لذا تمام نسخه‌های روش 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، [۴، [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، فرآخوانی 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  $B$  و  $C$  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  $R$  و  $C$  و  $B$  و  $D$  به صورت قطعی وجود داشته باشد، می‌توان با جانشین 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 Ex-SJF<sub>EXA</sub> یعنی Ex-SJF<sub>PRO</sub><sup>12</sup>، Ex-SJF<sub>EXA</sub><sup>11</sup>، Ex-SJF<sub>EXA</sub><sup>10</sup>، Ex-SJF<sub>PRO-V.2.8</sub>، Ex-SJF<sub>PRO-V.1.8</sub> به منظور پیش‌بینی قواعد فرزند یک قاعده به تشکیل درخت اجرای قواعد می‌پردازند [۵]. اما وجه تمایز این نسخه‌ها در نحوه تخمین احتمال اجرای قواعد فرزند می‌باشد.

## ۱-۵-۲ روش زمان‌بندی Ex-SJF<sub>EXA</sub>

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.

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



#### ۴-۲- روش مبتنی بر الگوریتم نزدیکترین ضرب العجل<sup>۴</sup>

این روش یکی از کامل‌ترین روش‌هایی است که تاکنون برای زمان-بندي قواعد در سیستم‌های پایگاه داده پویای بلدرنگ ارایه شده است و به اختصار EDF نامیده می‌شود. در این روش، هنگام انتخاب، قاعده‌ای که دارای نزدیکترین ضرب العجل باشد، برای اجرا انتخاب می‌شود. براساس نحوی محاسبه اولویت قواعد که مبتنی بر ضرب العجل قواعد است، سه نسخه‌ی مختلف از روش EDF طراحی شده است که عبارتند از: EDF<sub>EDF</sub>، EDF<sub>SL</sub> و EDF<sub>DIV</sub>. اولی در حقیقت همان الگوریتم EDF است. در دومی یک تغییر کوچک در الگوریتم EDF داده شده است به این ترتیب که اولویت قاعده‌ی پدر و قنی 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 [۵]. [۶] با جایگزین کردن «شبه کد» زیر بجای سطر ۵ شکل (۱) توصیف فرمال روش زمانبندی مبتنی بر الگوریتم نزدیکترین ضرب العجل، در حالت کلی بدست می‌آید.

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)  $n \equiv \{i \in N \mid i = \text{Number of ECA Rules in the ReadyToExecute_Rule_Base}\}$
- 5) Rule\_Scheduling (ReadyToExecute\_Rule\_Base)

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

#### ۴-۱- روش اتفاقی<sup>۵</sup>

در این روش هرگاه که در چرخه‌ی پردازش قواعد نیاز به انتخاب قاعده‌ی 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 ∈ N | 1 ≤ i ≤ n and i selected randomly}
- Rule\_Scheduling (ReadyToExecute\_Rule\_Base) ≡ { ∀ R\_i ∈ ReadyToExecute\_Rule\_Base, (R\_1, ..., R\_n), R\_i Selected | {R\_i ∈ ReadyToExecute\_Rule\_Base | 1 ≤ i ≤ n} }
- | R = ReadyToExecute\_Rule\_Base (R<sub>Random\_Generator(n)</sub>) }

#### ۴-۲- روش اولویت ایستا<sup>۶</sup>

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 [۷]، [۱۳]. 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) ≡ { ∀ R\_i ∈ ReadyToExecute\_Rule\_Base, [(R\_1, P\_1), ..., (R\_n, P\_n)], R\_i Selected | {R\_i ∈ ReadyToExecute\_Rule\_Base | 1 ≤ i ≤ n} }
- { ∀ R\_j ∈ ReadyToExecute\_Rule\_Base | 1 ≤ j ≤ n, P\_i ≤ P\_j }

#### ۴-۳- روش مبتنی بر برچسب زمانی<sup>۷</sup>

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 می‌یابد. 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) ≡ { ∀ R\_i ∈ ReadyToExecute\_Rule\_Base, [(R\_1, T\_1), ..., (R\_n, T\_n)], R\_i Selected }
- | {R\_i ∈ ReadyToExecute\_Rule\_Base | 1 ≤ i ≤ n}
- { ∀ R\_j ∈ ReadyToExecute\_Rule\_Base | 1 ≤ j ≤ n, T\_i ≤ T\_j }

#### ۵-۱- روش مبتنی بر الگوریتم ابتدا کوچکترین کار<sup>۸</sup>

این روش که مبتنی بر الگوریتم ابتدا کوچکترین کار (SJF) است، به اختصار Ex-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 پویا هستند، (مانند سیستم‌های پایگاه داده پویا) نیست و استفاده از آن موجب بهبود زمانبندی نخواهد شد [۸]. [۹]. با توجه به اینکه روش جدیدی که در بخش سوم ارائه خواهد شد، بهبودی بر روش Ex-SJF می‌باشد، در این قسمت نحوهی عملکرد روش Ex-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 Ex-SJF، در حالت کلی بدست می‌آید

- 5' Rule\_Scheduling (ReadyToExecute\_Rule\_Base) ≡ { ∀ R\_i ∈ ReadyToExecute\_Rule\_Base, (R\_1, ExecT\_1), ..., (R\_n, ExecT\_n), R\_i Selected }
- | {R\_i ∈ ReadyToExecute\_Rule\_Base | 1 ≤ i ≤ n} { ∀ R\_j ∈ ReadyToExecute\_Rule\_Base | 1 ≤ j ≤ n, ExecT\_i ≤ ExecT\_j }

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



```

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)، در صورتیکه، ارزش آن حداقل  $100 \cdot e$  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 اجرایت. 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 که با این رویداد، مرتبط هستند فعال شده و وارد لیست قواعد فعال می‌شوند. در ادامه با توجه به روش زمان‌بندی موجود 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 مجموعه‌ی قواعد آماده اجرا نیز با توجه به روش زمان‌بندی موجود 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، چرخهٔ پردازش قواعد گویند [۵].

دو فاکتور بسیار مؤثر در نحوه اجرای چرخهٔ پردازش قواعد، پیوستگی رویداد-شرط<sup>۲</sup> و پیوستگی شرط-عمل<sup>۳</sup> 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 بلافاصله پس از رخ دادن رویداد متناظرش 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 مثبت بودن نتیجهٔ ارزیابی، دستورات بخش عمل آن قاعده فوراً اجرا می‌شوند. اگر پیوستگی رویداد-شرط (شرط-عمل) یک قاعده از نوع تعویقی باشد، ارزیابی شرط (اجرای عمل) آن قاعده تا زمان اتمام تراکنش جاری به تأخیر می‌افتد. در برخی موارد ارزیابی شرط (اجرای عمل) تا رخ دادن رویداد خاصی به تأخیر خواهد افتاد. در حالت مستقل، ارزیابی شرط (اجرای 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 فال 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 در 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

پنج مرحله است که عبارتند از: (۱) تشخیص رویدادها، (۲) فعل سازی 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]. تاکنون چندین روش برای زمان‌بندی قواعد ارائه شده است [۱، 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 (Ex-SJF<sub>PRO</sub>-V.2.8) را معرفی کرده‌ایم [۱، ۲]. در این مقاله با اضافه کردن اتوماتای یادگیر به روش Ex-SJF<sub>PRO</sub>-V.2.8، درصد بهبود فرایند تخمین احتمال اجرای قواعد و متعاقباً ارائه یک روش زمان‌بندی کاراتر از بعد پارامترهای ارزیابی تعریف شده، برآمده‌ایم. همچنین با افزودن ویژگی‌هایی به شبیه‌ساز پایگاه داده پویا، سعی کرده‌ایم تا حد امکان به شبیه‌سازی و ارزیابی دقیقت رفتار پایگاه داده پویا و عملکرد روش‌های زمان‌بندی ببرداریم.

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 اتوماتای 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 جمع‌بندی 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<sup>۴</sup> 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 شی‌گرای SAMOS است:



## زمانبندی قواعد در پایگاه داده پویا به کمک اتماتاهای یادگیر

محمد رضا میبدی

عباس رسولزادگان

دانشگاه صنعتی امیرکبیر

دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

[mmeybodi@aut.ac.ir](mailto:mmeybodi@aut.ac.ir)

[rasoolzadegan@yahoo.com](mailto:rasoolzadegan@yahoo.com)

تاریخ دریافت: ۱۳۸۷/۱۲/۲۰ - تاریخ پذیرش: ۱۳۸۸/۵/۲۰

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 اتماتاهای یادگیر به منظور بهبود زمان‌بندی قواعد بر حسب میانگین زمان پاسخگویی، انحراف معیار زمان پاسخگویی و توان عملیاتی پیشنهاد می‌گردد. از اتماتاهای 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، اتماتاهای یادگیر

سهام) نیاز به نظارت خودکار دارند تا در صورت وقوع رویداد خاصی، واکنش مقتضی انجام شود. برای این منظور باید سیستم پایگاه داده‌ای 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، [۹].

رفتار واکنشی سیستم پایگاه داده پویا از طریق پردازش قواعد پویا در قالب چرخه پردازش قواعد، سازماندهی می‌شود. این چرخه شامل

### ۱- مقدمه

پایگاه‌های داده متداول دارای ماهیت ایستاده 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 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 پیچیده مالی (مانند بازار