مجله محاسبات نرم، جلد ۱۰، شماره ۲ (پاییز و زمستان ۱۴۰۰)، صفحه ۳۲ـ۴۳
https://doi.org/10.22052/scj.2022.243181.1002

مجله محاسبات نرم
SOFT COMPUTING JOURNAL
تارنمای مجله: scj.kashanu.ac.ir

دانشگاه کاشان
University of Kashan

تعیین برش زمانی پویا در الگوریتم زمانبندی نوبت گردشی با استفاده از یادگیری ماشین✧

سحر نجفی[۱]، دانشجوی کارشناسی ارشد، سمیرا نوفرستی[۱*]، استادیار
[۱] دانشکده مهندسی برق و کامپیوتر، دانشگاه سیستان و بلوچستان، زاهدان، ایران.

اطلاعات مقاله

تاریخچه مقاله:
دریافت ۱۸ مهر ماه ۱۴۰۰
پذیرش ۲۰ اردیبهشت ماه ۱۴۰۱

کلمات کلیدی:
زمانبندی پردازنده
نوبت گردشی
برش زمانی
تعیین برش زمانی پویا
یادگیری ماشین
تعویض‌متن

چکیده

یکی از رایج‌ترین الگوریتم‌های زمانبندی پردازنده در سیستم‌های اشتراک زمانی، الگوریتم نوبت گردشی است. در این الگوریتم یک برش زمانی در نظر گرفته می‌شود که بیانگر حداکثر زمانی است که فرآیند می‌تواند پردازنده را در اختیار داشته باشد و پردازنده به اندازه برش زمانی به صورت چرخشی به فرآیندهای صف آماده تخصیص می‌یابد. اندازه برش زمانی تاثیر بسزایی بر کارایی الگوریتم نوبت گردشی دارد به گونه‌ای که اگر برش زمانی خیلی کوتاه باشد، به دلیل افزایش تعداد تعویض‌متن و سربار ناشی از آن، بهره‌وری پردازنده کاهش می‌یابد. در مقابل اگر برش زمانی خیلی بزرگ باشد، میانگین زمان پاسخ فرآیندها افزایش می‌یابد که استفاده از الگوریتم نوبت گردشی در کاربردهای تعاملی را ناکارآمد می‌سازد. هدف این مقاله ارائه روشی کارا برای تعیین برش زمانی به صورت پویا با استفاده از یادگیری ماشین است. برای این منظور، ابتدا یک مجموعه آموزش شامل ویژگی‌های تعداد فرآیندها و بیشینه، کمینه، میانگین و میانه زمان انفجار فرآیندها و ویژگی هدف برش زمانی بهینه ساخته شد. سپس با آموزش طبقه‌بندهای یادگیری ماشین بر روی این مجموعه به پیش‌بینی برش زمانی بهینه برای نمونه‌های جدید پرداخته شد. نتایج ارزیابی‌ها نشان می‌دهد که در مجموع روش پیشنهادی در مقایسه با سایر روش‌های تعیین برش زمانی عملکرد بهتری بر اساس معیارهای سنجش کارایی الگوریتم‌های زمانبندی دارد. برای مثال در مقایسه با الگوریتم ژنتیک که در میان روش‌های موجود بهترین عملکرد را دارد، میانگین زمان انتظار و میانگین تعداد تعویض‌متن روش پیشنهادی به ترتیب ۱۲ میلی‌ثانیه و ۱/۷۶ واحد بهبود و میانگین زمان برگشت حدود ۲ میلی‌ثانیه افت داشته است.

© ۱۴۰۱ ـ مجلهٔ محاسبات نرم، کلیه حقوق محفوظ است.

۱. مقدمه

زمانبندی یکی از مهم‌ترین خدمات سیستم‌عامل است که دسترسی فرآیندها به منابع سیستم را فراهم می‌کند. الزاماتی نظیر محاسبات سریع و چندوظیفه‌ای (اجرای بیش از یک فرآیند به صورت همزمان یا همروند) اهمیت الگوریتم‌های زمانبندی منابع را دوچندان کرده است [۱]. الگوریتم‌های متعددی برای مساله زمانبندی پردازنده مطرح شده است. یکی از الگوریتم‌های پایه زمانبندی، الگوریتم نوبت گردشی[۱] است که به اختصار RR نامیده می‌شود. در این الگوریتم، یک برش زمانی[۲] تعریف

✧ نوع مقاله: پژوهشی
* نویسنده مسئول
پست(های) الکترونیک: sahar72najafi@gmail.com (نجفی)
snoferesti@ece.usb.ac.ir (نوفرستی)

[1] Round Robin
[2] Time Quantum

نحوه ارجاع به مقاله: نجفی، سحر، نوفرستی، سمیرا، «تعیین برش زمانی پویا در الگوریتم زمانبندی نوبت گردشی با استفاده از یادگیری ماشین»، مجله محاسبات نرم، جلد ۱۰، شماره ۲، ص۳۲ـ۴۳، پاییز و زمستان ۱۴۰۰.

می‌شود که بیانگر حداکثر زمانی است که یک فرآیند می‌تواند پردازنده را در اختیار داشته باشد. در این الگوریتم پردازنده به صورت چرخشی به اندازه برش زمانی به فرآیندهای صف آماده تخصیص می‌یابد. هرگاه فرآیندی پردازنده را در اختیار بگیرد، دو حالت ممکن است رخ دهد: (۱) اگر اجرای فرآیند قبل از اتمام برش زمانی آن به پایان برسد، فرآیند پردازنده را آزاد می‌کند و سپس پردازنده به فرآیند بعدی در صف آماده اختصاص می‌یابد و (۲) اگر زمان انفجار فرآیند بیشتر از برش زمانی باشد، پس از اتمام برش زمانی، پردازنده از آن فرآیند گرفته شده و به فرآیند بعدی در صف آماده اختصاص می‌یابد. کارایی الگوریتم نوبت گردشی بسیار وابسته به اندازه برش زمانی است. به دلیل اهمیت اندازه برش زمانی، تحقیقات متعددی در این زمینه انجام گرفته است. به صورت کلی می‌توان تحقیقات پیشین را به دو دسته تقسیم کرد. دسته اول تحقیقاتی هستند که برش زمانی را ایستا در نظر می‌گیرند [۱-۴]. در این دسته از تحقیقات مقدار ثابت و مشخصی برای برش زمانی در نظر گرفته می‌شود و در طول اجرای زمانبندی این مقدار تغییر نمی‌کند. دسته دوم تحقیقاتی هستند که متکی بر برش زمانی پویا هستند [۵-۱۳]. در این دسته از تحقیقات مقدار برش زمانی برای هر دور از الگوریتم به صورت مجزا تعیین می‌شود. به دلیل اینکه بعد از اجرای بخشی از فرآیندها در هر دور از اجرای الگوریتم نوبت گردشی، زمان انفجار باقی‌مانده آنها تغییر می‌کند و در نتیجه مشخصات فرآیندهای صف آماده در دور بعدی متفاوت از دور قبلی است، روش‌های پویا برای تعیین برش زمانی به نتایج بهتری در مقایسه با روش‌های ایستا دست یافته‌اند.

در این مقاله، یک روش کارا برای تعیین برش زمانی به صورت پویا با استفاده از یادگیری ماشین ارائه خواهد شد. نتایج آزمایشات انجام گرفته، کارایی روش پیشنهادی را در مقایسه با سایر روش‌های تعیین برش زمانی بر اساس معیارهای میانگین تعداد تعویض متن، میانگین زمان انتظار و میانگین زمان برگشت فرآیندها نشان می‌دهد.

۲. مرور کارهای پیشین

همان‌طور که گفته شد، تحقیقات پیشین در زمینه تعیین برش زمانی به دو دسته ایستا و پویا تقسیم می‌شوند. در روش‌های ایستا یک عدد ثابت برای برش زمانی در نظر گرفته می‌شود که اغلب با سعی و خطا تعیین می‌گردد. برای نمونه، در تحقیقات [۱] و [۲] به ترتیب دو مقدار ثابت ۴ و ۱۰ به عنوان برش زمانی در نظر گرفته شده است.

یکی از الگوریتم‌هایی که به صورت ایستا برش زمانی را تعیین می‌کند، الگوریتم MMRRA است [۳]. این الگوریتم در ابتدا فرآیندها را به ترتیب صعودی زمان انفجارشان مرتب می‌کند و سپس با استفاده از رابطه (۱) برش زمانی را محاسبه می‌کند.

$$TQ = \sqrt{Median \times HB} \qquad (۱)$$

که در این رابطه، $Median$ و HB به ترتیب نشان‌دهنده میانه و بزرگترین زمان انفجار فرآیندهای صف آماده است.

در تحقیقی دیگر، یک مدل ریاضی غیرخطی برای تعیین برش زمانی در الگوریتم نوبت گردشی با هدف کاهش میانگین زمان انتظار فرآیندها پیشنهاد شده است [۴]. در این مدل، زمان انتظار یک فرآیند به صورت زمان برگشت منهای زمان اجرای آن محاسبه می‌شود. زمان برگشت فرآیند نیز برابر است با مجموع زمان‌هایی که فرآیند به اندازه یک برش زمانی کامل، پردازنده را در اختیار دارد به اضافه زمان‌های تعویض متن به اضافه مقدار آخرین زمان انفجار فرآیند به اضافه مجموع زمان‌های پردازش فرآیندهای قبلی و بعدی فرآیند در صف آماده که قبل از اتمام این فرآیند پردازنده را در اختیار داشته‌اند.

در مجموع، روش‌های ایستا به دلیل عدم توجه به تغییرات صف آماده در هر دور از الگوریتم نوبت گردشی از کارایی پایین‌تری در مقایسه با روش‌های پویا برخوردار هستند. روش‌های پویا، برش زمانی را برای هر دور از الگوریتم نوبت گردشی به صورت مجزا تعیین می‌کنند و به همین دلیل نسبت به روش‌های ایستا انعطاف‌پذیری بیشتری دارند. یکی از الگوریتم‌های معرفی شده برای تعیین برش زمانی به صورت پویا، الگوریتم IRRVQ است [۵]. در این الگوریتم ابتدا فرآیندها به ترتیب صعودی

زمان انفجارشان در صف آماده قرار می‌گیرند. سپس پردازنده با استفاده از الگوریتم زمانبندی نوبت گردشی با مقدار برش زمانی برابر با زمان انفجار فرآیند اول صف، به فرآیندها اختصاص می‌یابد.

یکی دیگر از الگوریتم‌های ارائه شده برای تعیین برش زمانی پویا، الگوریتم SARR است که ابتدا فرآیندهای موجود در صف آماده را به ترتیب صعودی زمان انفجارشان مرتب می‌کند. سپس میانه زمان انفجار فرآیندها در هر دور به عنوان برش زمانی آن دور در نظر می‌گیرد [۶].

در تحقیقی دیگر، از الگوریتم ژنتیک برای تعیین برش زمانی استفاده شده است [۷]. در این روش که GARR نامیده می‌شود، جمعیت اولیه شامل تعدادی کروموزوم است که هر یک از ژن‌های کروموزوم نشان‌دهنده یکی از فرآیندهای صف آماده است. جمعیت اولیه به صورت تصادفی تولید می‌شود، سپس برای هر یک از کروموزوم‌ها، زمان انفجار فرآیند میانی کروموزوم به عنوان برش زمانی در نظر گرفته شده و همه فرآیندها با این برش زمانی اجرا می‌شوند. بعد از اجرای کلیه فرآیندها، شایستگی یک کروموزوم بر اساس میانگین زمان انتظار فرآیندها محاسبه می‌شود. بعد از تولید جمعیت اولیه، با استفاده از عملگرهای ژنتیکی انتخاب، تولید مثل و جهش جمعیت بعدی تولید و بر اساس تابع شایستگی ارزش‌گذاری می‌شود. به طور مشابه ۱۰ نسل تولید شده و بهترین کروموزوم در این ۱۰ نسل یعنی کروموزومی که کمترین میانگین زمان انتظار را دارد به عنوان جواب مساله انتخاب می‌شود و زمان انفجار فرآیند میانی این کروموزوم به عنوان برش زمانی نهایی در نظر گرفته می‌شود.

در الگوریتم DABRR ابتدا فرآیندهای موجود در صف آماده به ترتیب صعودی زمان انفجارشان مرتب شده و سپس در ابتدای هر دور، میانگین زمان انفجار فرآیندهای موجود در صف آماده به عنوان برش زمانی در نظر گرفته می‌شود [۸].

در الگوریتم RDQ ابتدا فرآیندها بر حسب زمان انفجار به شکل صعودی مرتب شده و سپس برش زمانی به صورت تصادفی از بازه [min-max] که min میانه زمان انفجار فرآیندها و max بزرگترین زمان انفجار آنها است، انتخاب می‌شود [۹].

در الگوریتم EDRR، در دور اول اندازه برش زمانی ۰/۸ بزرگ‌ترین زمان انفجار فرآیندها در نظر گرفته می‌شود و در دور بعدی اندازه برش زمانی برابر بزرگترین زمان انفجار باقیمانده فرآیندها در صف آماده تعریف می‌شود، در نتیجه همه فرآیندها به طور کامل اجرا می‌شوند [۱۰].

در پژوهش [۱۱]، دو روش برای تعیین برش زمانی پویا پیشنهاد شده است. ابتدا صف آماده به دو نیمه تقسیم می‌شود. در روش اول، در هر دور، مجموع میانگین زمان انفجار فرآیندهای نیمه اول صف و میانگین زمان انفجار فرآیندهای نیمه دوم صف به عنوان برش زمانی در نظر گرفته می‌شود. در حالی که در روش دوم، در هر دور، تفاضل میانگین زمان انفجار فرآیندهای نیمه اول صف آماده و میانگین زمان انفجار فرایندهای نیمه دوم صف به عنوان برش زمانی در نظر گرفته می‌شود. مطابق نتایج این تحقیق، روش اول میانگین زمان برگشت و میانگین زمان انتظار کمتری دارد.

از دیگر روش‌های ارائه شده برای تعیین برش زمانی پویا، روشی مبتنی بر شبکه عصبی برای پیش‌بینی اندازه مناسب برش زمانی در الگوریتم نوبت گردشی با هدف به حداقل رساندن میانگین مربع خطا[1] بین مقدار برش زمانی تخمین زده شده و مقدار بهینه واقعی است [۱۲]. اشکال اصلی این روش این است که تعداد فرآیندها را ثابت و برابر ۱۰ در نظر گرفته است که یک فرض غیرواقعی است.

در تحقیقی دیگر، ابتدا فرآیندها توسط الگوریتم K-means و بر اساس ویژگی‌های زمان انفجار، وزن (زمان انفجار فرآیند تقسیم بر مجموع زمان انفجار فرآیندهای صف آماده) و تعداد دفعات تخصیص پردازنده خوشه‌بندی می‌شوند. سپس برش زمانی فرآیندهای یک خوشه با توجه به وزن آن خوشه (میانگین زمان انفجار فرآیندهای آن خوشه تقسیم بر مجموع میانگین زمان انفجار کلیه خوشه‌ها) تعیین می‌شود [۱۳].

به طور خلاصه، اغلب روش‌های پویا از یک ویژگی ساده مانند میانه یا میانگین زمان انفجار فرآیندها یا ترکیبی از آنها برای

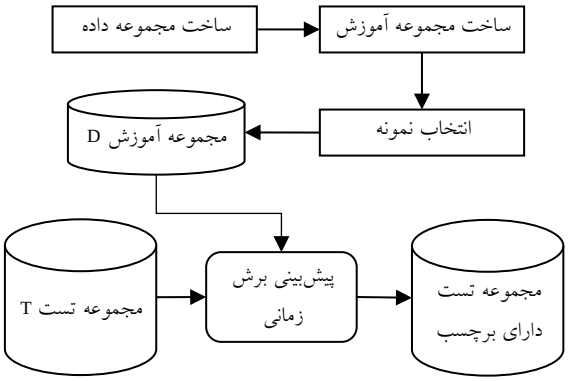
[1] Mean Square Error

تعیین برش زمانی در هر دور از الگوریتم نوبت گردشی استفاده می‌کنند. مزیت این روش‌ها سادگی و عیب اصلی آنها دقت کم است [۵، ۶، ۸-۱۱]. در مقابل، روش‌های هوشمند مانند الگوریتم ژنتیک، شبکه عصبی و خوشه‌بندی از دقت بالاتری در تعیین برش زمانی برخوردار هستند ولی پیچیدگی محاسباتی بالاتری دارند و زمان زیادی را صرف تعیین برش زمانی می‌کنند [۷، ۱۲، ۱۳]. لذا در این مقاله روشی با بکارگیری الگوریتم‌های یادگیری ماشین برای تعیین برش زمانی پویا پیشنهاد می‌شود. یادگیری ماشین در سال‌های اخیر بسیار مورد توجه محققان بوده و به صورت موفقیت‌آمیزی در کاربردهای مختلف استفاده شده است [۱۴-۱۷].

۳. روش پیشنهادی

مراحل روش پیشنهادی (که به اختصار MLDTQ[1] نامیده شده است)، در شکل (۱) نمایش داده شده است. همان طور که در این شکل مشاهده می‌شود، اولین مرحله ساخت مجموعه داده است. بر اساس تحقیقات نویسندگان مقاله حاضر، مجموعه داده استانداردی برای ارزیابی الگوریتم‌های تعیین برش زمانی وجود ندارد. در تحقیقات موجود مانند [۱۲] و [۱۳] مجموعه داده به صورت تصادفی ایجاد شده است. در سایر تحقیقات نیز برای ارزیابی روش پیشنهادی مثال‌هایی از فرآیندهای صف آماده ذکر شده است و کارایی روش پیشنهادی برای این مثال‌ها با سایر روش‌ها مقایسه شده است.

در تحقیق حاضر، ابتدا یک مجموعه داده ایجاد شده است که هر رکورد آن یک مثال از وضعیت صف آماده را نشان می‌دهد. هر رکورد در این مجموعه داده شامل تعداد فرآیندهای موجود در صف آماده و زمان انفجار هر یک از آنها است. تعداد فرآیندها به صورت تصادفی از بازه $[5,10]$ و زمان انفجار هر یک از فرآیندها، به تصادف از بازه $[8,100]$ انتخاب شده است. بازه در نظر گرفته شده برای تعداد فرآیندها و نیز محدوده مجاز برای زمان انفجار فرآیندها بر اساس تحقیقات پیشین انتخاب شده است. یکی از مزایای روش پیشنهادی این است که می‌توان مجموعه داده را برای کاربردی خاص و با توجه به ویژگی‌های فرآیندها و محدودیت‌های تعریف شده برای آن کاربرد تهیه کرد.



شکل (۱): مراحل روش پیشنهادی

الگوریتم‌های یادگیری ماشین باناظر نیاز به داده‌های آموزشی دارند که با آموزش بر روی آن یک مدل پیش‌بینی‌کننده ساخته می‌شود. از این مدل برای پیش‌بینی ویژگی هدف در نمونه‌های جدید استفاده می‌شود. مرحله دوم روش پیشنهادی، ساخت مجموعه آموزش است. به منظور ساخت مجموعه آموزش، برای هر رکورد در مجموعه داده اولیه، ویژگی‌های تعداد فرآیندها، بیشینه زمان انفجار فرآیندها، کمینه زمان انفجار فرآیندها، میانگین و میانه زمان انفجار فرآیندها که در تعیین برش زمانی بهینه مؤثر هستند، تعیین شده‌اند. بدین ترتیب یک مجموعه آموزش شامل ۲۸۰۰ نمونه و ۵ ویژگی ایجاد شده است. سپس برای هر رکورد از مجموعه آموزش ویژگی هدف، که همان برش زمانی بهینه است، محاسبه شده است.

برای تعیین برش زمانی بهینه (ویژگی هدف)، مقداری از بازه $[\alpha,\beta]$ که منجر به کمترین میانگین زمان برگشت فرآیندهای موجود در صف آماده می‌شود، انتخاب شده است. متغیرهای α و β به ترتیب با استفاده از روابط (۲) و (۳) بدست می‌آیند.

$$\alpha = max(minQ, minBT) \quad (۲)$$

$$\beta = max(median, \gamma \times maxBT) \quad (۳)$$

که $minQ$ حداقل مقدار تعیین شده برای برش زمانی (در این مقاله برای دور اول ۸ و برای سایر دورها ۵)، $minBT$ کمترین

[1] Machin Learning Dynamic Time Quantum

زمان انفجار فرآیندها، $median$ میانه زمان انفجار فرآیندها و $maxBT$ بیشینه زمان انفجار فرآیندهای صف آماده را نشان می‌دهد. در ضمن γ ضریب ثابتی است که مقدار آن بر اساس مرجع [۱۰] برابر ۰/۸ در نظر گرفته شده است. در واقع ایده به این صورت است که در هر دور، حداقل مقدار ممکن برای برش زمانی برابر بیشینه دو مقدار حداقل زمان انفجار فرآیندهای صف آماده و حداقل مقدار تعیین شده برای برش زمانی است. حداکثر مقدار ممکن برای برش زمانی نیز برابر بیشینه دو مقدار میانه زمان انفجار فرآیندها و ۰/۸ بزرگترین زمان انفجار در صف آماده است. هر مقداری در بازه $[\alpha, \beta]$ که منجر به کمترین میانگین زمان برگشت شود، به عنوان برش زمانی بهینه انتخاب می‌گردد. اگر چند برش زمانی منجر به میانگین زمان برگشت یکسان شوند، معیار کمترین تعداد تعویض‌متن برای تعیین مقدار بهینه برش زمانی در نظر گرفته می‌شود.

در مرحله سه، با استفاده از یک روش انتخاب نمونه، نمونه‌های زائد از مجموعه آموزش حذف شده‌اند. از آنجا که مجموعه آموزش به صورت تصادفی ایجاد شده است، احتمال وجود نمونه‌های مشابه که تاثیر چندانی در فرآیند یادگیری ندارند، وجود دارد. از طرف دیگر زمان اجرای الگوریتم پیشنهادی برای تعیین برش زمانی بسیار حائز اهمیت است. بنابراین سعی شده است با بکارگیری یک الگوریتم انتخاب نمونه، اندازه مجموعه آموزش تا حد ممکن کاهش یابد. بدین منظور با بکارگیری روش معرفی شده در مرجع [۱۸] برای انتخاب نمونه، حدود ۴۲ درصد نمونه‌ها حذف و مجموعه باقیمانده شامل ۱۶۲۴رکورد به عنوان مجموعه آموزش نهایی انتخاب شده است که در ادامه آن را D می‌نامیم.

در مرحله چهارم، یک طبقه‌بند یادگیری ماشین که بر روی مجموعه D آموزش داده شده است، برش زمانی بهینه را برای رکوردهای مجموعه تست پیش‌بینی می‌کند. برای پیش‌بینی برش زمانی به صورت پویا، برای هر دور از الگوریتم نوبت گردشی ویژگی‌های مستخرج از زمان انفجار فرآیندهای منتظر در صف آماده را به طبقه‌بند یادگیری ماشین داده و برش زمانی برای آن دور تعیین می‌شود. از آنجا که زمان انفجار فرآیندهای منتظر در صف در هر دور تغییر می‌کند، ویژگی‌های مستخرج از آنها نیز تغییر کرده و ممکن است برش زمانی متفاوتی برای هر دور پیش‌بینی شود.

در این مقاله برای زمانبندی فرآیندها بدین صورت عمل می‌شود که در ابتدا فرآیندهای موجود در صف آماده به ترتیب صعودی زمان انفجارشان مرتب می‌شوند. سپس فرآیندهای صف آماده با الگوریتم نوبت گردشی با برش زمانی پیش‌بینی شده توسط الگوریتم یادگیری ماشین زمانبندی می‌شوند. دلیل مرتب کردن فرآیندها این است که اغلب مطالعات موجود در زمینه تعیین برش زمانی ایستا و پویا از صف آماده مرتب استفاده کرده‌اند (به دلیل کارایی بالاتر در مقایسه با صف نامرتب) و برای مقایسه عادلانه در این تحقیق نیز بدین شکل عمل شده است. در بخش نتایج، نشان داده می‌شود که روش پیشنهادی برای حالتی که فرآیندهای صف آماده مرتب نباشند نیز عملکرد موفقی دارد.

۴. ارزیابی روش پیشنهادی

در این بخش کارایی روش پیشنهادی برای تعیین برش زمانی پویا بر اساس معیارهای میانگین زمان برگشت، میانگین زمان انتظار و میانگین تعداد تعویض متن ارزیابی شده و نتایج حاصل با نتایج بدست آمده از روش‌های موجود مقایسه می‌شود. برای محاسبه میانگین کل زمان برگشت، ابتدا میانگین زمان برگشت برای فرآیندهای هر رکورد در مجموعه تست محاسبه می‌شود. سپس میانگین این مقادیر به عنوان میانگین کل برای زمان برگشت گزارش می‌شود. میانگین کل زمان انتظار و میانگین تعداد تعویض متن هم به روشی مشابه به دست می‌آید. در ادامه این بخش، ابتدا مجموعه داده‌های مورد استفاده معرفی شده و سپس نتایج ارزیابی کارایی روش پیشنهادی ارائه می‌گردد.

۴,۱. مجموعه داده مورد استفاده

در این مقاله از ۳ مجموعه داده برای آموزش، ارزیابی و تست استفاده شده است. مجموعه آموزش شامل ۲۸۰۰ نمونه است که پس از اعمال الگوریتم انتخاب نمونه و حذف نمونه‌های زائد، تعداد نمونه‌های آن به ۱۶۲۴ کاهش یافته است. مجموعه‌های

تست و ارزیابی نیز هر یک شامل ۵۰۰ نمونه هستند. از مجموعه ارزیابی برای تنظیم متغیرهای طبقه‌بندهای یادگیری ماشین و نیز برای انتخاب نمونه استفاده شده است. هر کدام از ۳ مجموعه مذکور شامل ۵ ویژگی تعداد فرآیندهای موجود در صف آماده، بیشینه، کمینه، میانگین و میانه زمان انفجار فرآیندهای موجود در صف آماده به همراه یک ویژگی هدف به نام برش زمانی بهینه می‌باشد. زمان انفجار فرآیندها یک عدد تصادفی بین ۸ تا ۱۰۰ و تعداد فرآیندها یک عدد تصادفی بین ۵ تا ۱۰ است که بر اساس تحقیقات پیشین تعیین شده است.

۴،۲. نتایج ارزیابی روش پیشنهادی

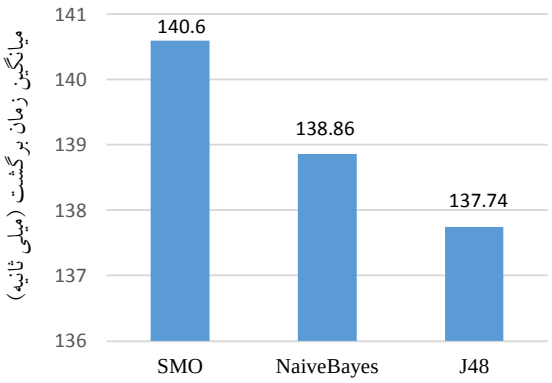
روش پیشنهادی با استفاده از زبان برنامه‌نویسی جاوا پیاده‌سازی شده است. برای پیاده‌سازی الگوریتم‌های یادگیری ماشین نیز از بسته داده‌کاوی وکا[1] استفاده شده است. وکا یک نرم‌افزار کد باز برای پیاده‌سازی الگوریتم‌های یادگیری ماشین و داده‌کاوی است که به زبان جاوا نوشته شده است و یک واسط برنامه‌نویسی کاربردی (API) دارد که امکان استفاده از آن را در برنامه‌های نوشته شده به زبان جاوا فراهم می‌کند.

در آزمایش اول، به بررسی کارایی طبقه‌بندهای یادگیری ماشین مختلف در پیش‌بینی برش زمانی پرداخته شده است. در این مقاله، از سه طبقه‌بند ماشین بردار پشتیبان، بیز ساده و درخت تصمیم استفاده شده است که در وکا به ترتیب با عنوان SMO، NaiveBayes و J48 شناخته می‌شوند. در جدول (۱)، کارایی سه طبقه‌بند فوق در پیش‌بینی برش زمانی بر اساس معیارهای جذر میانگین مربعات خطا[2] (RMSE) و میانگین خطای مطلق[3] (MAE) مقایسه شده است.

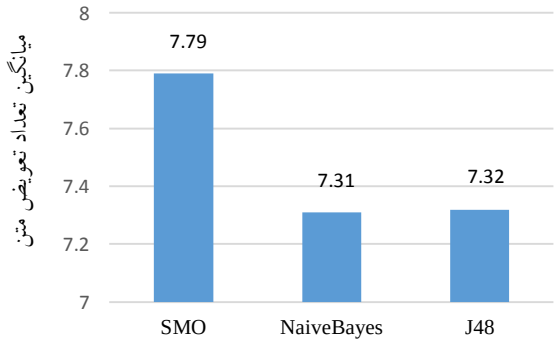
جدول (۱): کارایی طبقه‌بندهای یادگیری ماشین در پیش‌بینی برش زمانی

طبقه‌بند	RMSE	MAE
SMO	۰/۱۰۱	۰/۰۲۰
Naive Bayes	۰/۰۱۰	۰/۰۱۸
J48	۰/۱۰۳	۰/۰۱۵

در شکل (۲) میانگین زمان برگشت بر حسب میلی‌ثانیه و در شکل (۳) میانگین تعداد تعویض متن برای طبقه‌بندهای مختلف مورد مقایسه قرار گرفته است. بر اساس نتایج جدول (۱) و شکل‌های (۲) و (۳)، طبقه‌بند J48 کارایی بیشتری در مقایسه با دو طبقه‌بند دیگر به دست آورده است. بدین دلیل در ادامه از طبقه‌بند J48 استفاده شده است.



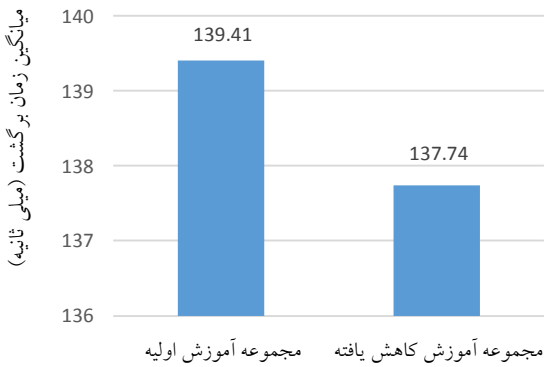
شکل (۲): میانگین زمان برگشت برای SMO، NaiveBayes و J48



شکل (۳): میانگین تعداد تعویض متن برای SMO، NaiveBayes و J48

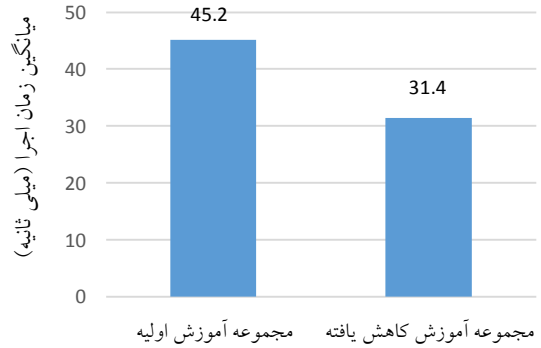
در آزمایش دوم، تاثیر انتخاب نمونه برای کاهش اندازه مجموعه آموزش بر کارایی روش پیشنهادی برای تعیین برش زمانی بررسی شده است. شکل (۴) میانگین زمان برگشت روش پیشنهادی در حالتی که آموزش طبقه‌بند J48 بر روی مجموعه آموزش اولیه با ۲۸۰۰ رکورد انجام شود را با حالتی که آموزش بر روی مجموعه کاهش‌یافته D با ۱۶۲۴ رکورد انجام شود، مقایسه می‌کند. همانطور که مشاهده می‌شود استفاده از مجموعه آموزش کاهش یافته منجر به بهبود معیار میانگین زمان برگشت در الگوریتم نوبت گردشی شده است.

[1] Weka
[2] Root Mean Square Error
[3] Mean absolute error



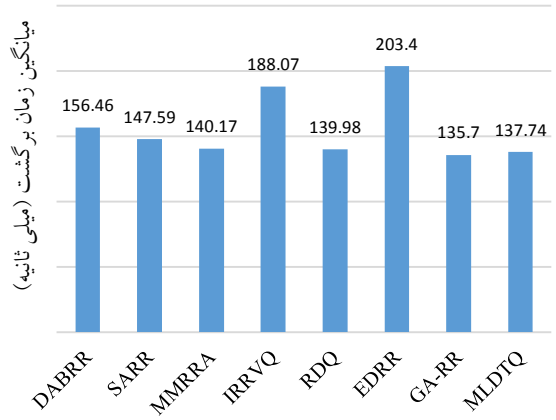
شکل (۴): مقایسه میانگین زمان برگشت برای مجموعه آموزش اولیه و مجموعه کاهش‌یافته

شکل (۵) میانگین زمان اجرای کل برنامه برای ۵۰۰ رکورد مجموعه تست را دو در حالت آموزش بر روی مجموعه ۲۸۰۰ رکوردی اولیه و آموزش بر روی مجموعه کاهش‌یافته نشان می‌دهد. زمان اجرای گزارش شده برابر با میانگین ۵ بار اجرای الگوریتم‌ها است. لازم به ذکر است که زمان اجرای نمایش داده شده در شکل (۵) شامل زمان ساخت مدل پیش‌بینی نیست بلکه تنها شامل زمان اجرای روش پیشنهادی جهت پیش‌بینی برش زمانی پویا برای همه نمونه‌های مجموعه تست و زمان اجرای الگوریتم نوبت گردشی برای زمانبندی فرآیندها بر اساس برش زمانی تعیین شده برای هر دور است. در واقع، مرحله ساخت مدل بر اساس نمونه‌های آموزشی به صورت غیربرخط (آفلاین) انجام می‌شود. مطابق شکل (۵)، زمان اجرای الگوریتم هنگامی که از مجموعه کاهش‌یافته استفاده می‌شود به طور متوسط ۱۳/۸ میلی‌ثانیه کمتر است.



شکل (۵): میانگین زمان اجرا برای مجموعه آموزش اولیه و مجموعه کاهش‌یافته

در آزمایش سوم، کارایی روش پیشنهادی با ۷ روش موجود تعیین برش زمانی شامل DABRR [۸]، SARR [۶]، MMRRA [۳]، IRRVQ [۵]، RDQ [۹]، EDRR [۱۰] و GARR [۷] مقایسه شده است. کلیه روش‌های مذکور با استفاده از زبان برنامه‌نویسی جاوا و پلتفرم Eclipse پیاده‌سازی و بر روی یک سیستم کامپیوتری با حافظه ۴ گیگابایت و پردازنده‌ای با سرعت ۲/۱ گیگاهرتز اجرا شده‌اند. لازم به ذکر است که در کلیه روش‌های مذکور، برش زمانی به صورت پویا و برای هر دور از الگوریتم تعیین می‌شود. به علاوه همه روش‌های موجود بجز EDRR و GARR در ابتدا فرآیندهای صف آماده را به ترتیب صعودی مرتب می‌کنند و الگوریتم نوبت گردشی بر روی صف مرتب شده اعمال می‌شود. شکل (۶) کارایی روش پیشنهادی را بر اساس معیار میانگین زمان برگشت با ۷ روش مذکور مقایسه می‌کند که مشاهده می‌شود روش پیشنهادی (یعنی MLDTQ) بعد از GARR بهترین میانگین زمان برگشت را دارد.

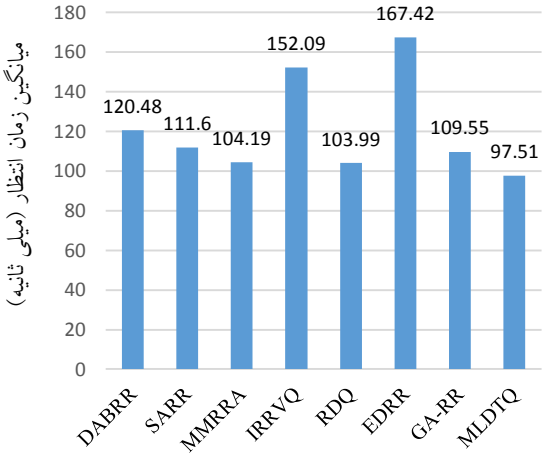


شکل (۶): میانگین زمان برگشت روش پیشنهادی و روش‌های موجود

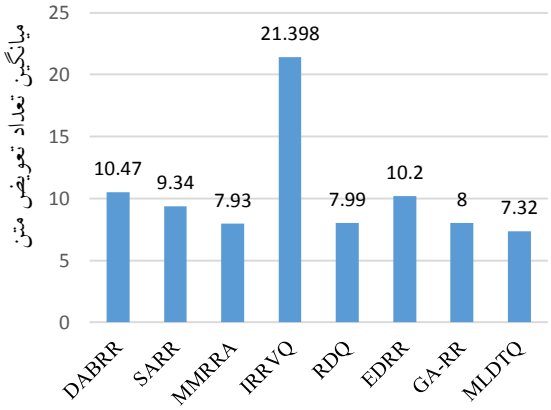
شکل (۷) میانگین زمان انتظار روش پیشنهادی را با سایر روش‌های تعیین برش زمانی پویا مقایسه می‌کند. همان طور که مشاهده می‌شود از بین ۷ روش موجود، روش EDRR، بیشترین میانگین زمان انتظار و روش پیشنهادی (MLDTQ) کمترین میانگین زمان انتظار را دارد. به عنوان نمونه، میانگین زمان انتظار روش پیشنهادی ۶/۴۸ میلی‌ثانیه کمتر از روش RDQ است که در میان روش‌های موجود بهترین کارایی را دارد.

شکل (۸) میانگین تعداد تعویض متن در روش پیشنهادی و

سایر روش‌ها را نشان می‌دهد. میانگین تعداد تعویض متن روش پیشنهادی به میزان ۰/۶۱ از روش MMRRA که در میان روش‌های موجود بهترین عملکرد را دارد، کمتر است.

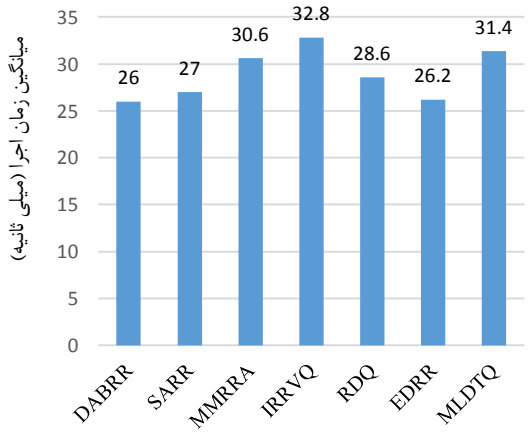


شکل (۷): میانگین زمان انتظار روش پیشنهادی و روش‌های موجود



شکل (۸): میانگین تعداد تعویض متن روش پیشنهادی با روش‌های دیگر

شکل (۹) میانگین زمان اجرای روش پیشنهادی و ۶ روش دیگر برای تعیین برش زمانی (و نه اجرای الگوریتم زمانبندی) را برای ۵ بار اجرا نشان می‌دهد. میانگین زمان ۵ مرتبه اجرای روش GARR، ۳۵۵ میلی‌ثانیه گزارش شده است که به دلیل تفاوت فاحش زمان اجرای این روش با سایر روش‌ها و برای وضوح بیشتر شکل از ارائه زمان اجرای این الگوریتم در شکل (۹) صرفنظر شده است. همانطور که در این شکل مشاهده می‌شود، روش DABRR دارای کمترین میانگین زمان اجرا و روش IRRVQ دارای بیشترین میانگین زمان اجرا می‌باشد. روش پیشنهادی برای تعیین برش زمانی ۵۰۰ نمونه داده تست، از بهترین روش موجود از نظر زمان اجرا یعنی روش DABRR، ۵/۴ میلی‌ثانیه (به طور تقریبی ۰/۰۱ میلی‌ثانیه برای هر نمونه تست) کندتر است. این در حالی است که مطابق شکل (۶) به طور متوسط میانگین زمان برگشت روش پیشنهادی ۱۸/۷۲ میلی‌ثانیه کمتر از روش DABRR است. به بیانی دیگر، اگرچه روش پیشنهادی زمان بیشتری در مقایسه با برخی از روش‌های موجود مانند DABRR برای تعیین برش زمانی پویا صرف می‌کند، اما به دلیل انتخاب مقدار بهتر برای برش زمانی، میانگین زمان اجرای الگوریتم نوبت گردشی را به مقدار قابل توجهی در مقایسه با دیگر روش‌ها کاهش می‌دهد به طوری که در مجموع زمان صرف شده برای تعیین برش زمانی و اجرای الگوریتم زمانبندی در روش پیشنهادی از سایر روش‌های موجود کمتر است.



شکل (۹): میانگین زمان اجرای روش پیشنهادی و روش‌های موجود برای تعیین برش زمانی

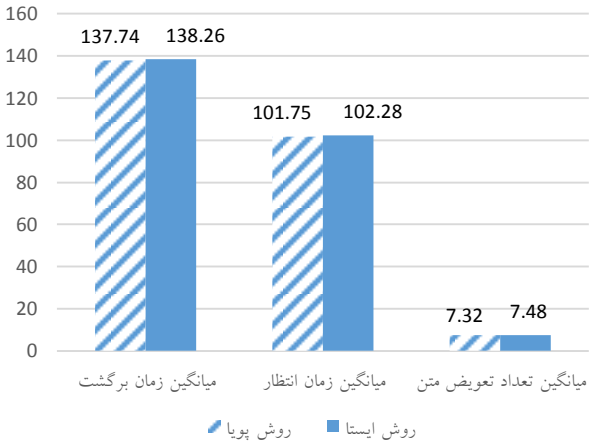
نتایج ارزیابی کارایی روش پیشنهادی در مقایسه با ۷ روش دیگر بر اساس معیارهای ارزیابی مختلف در جدول (۲) خلاصه شده است. در هر سطر بهترین نتیجه پررنگ شده است. همان‌طور که مشاهده می‌شود روش پیشنهادی در مجموع به نتایج بهتری در مقایسه با سایر روش‌ها دست یافته است.

در آزمایش چهارم، به مقایسه کارایی روش پیشنهادی در تعیین برش زمانی به صورت ایستا و پویا پرداخته شده است. در روش پویا با بکارگیری طبقه‌بند J48 در هر دور از الگوریتم زمانبندی

به پیش‌بینی برش زمانی برای فرآیندهای موجود در صف آماده پرداخته می‌شود. در حالی که در روش ایستا برای هر نمونه تست تنها یکبار طبقه‌بند J48 برای پیش‌بینی برش زمانی بر اساس وضعیت اولیه فرآیندهای صف آماده بکار گرفته می‌شود و در دورهای بعدی از همین برش زمانی استفاده می‌گردد. در شکل (۱۰) کارایی روش پیشنهادی در دو حالت ایستا و پویا بر اساس معیارهای میانگین زمان برگشت، میانگین زمان انتظار و میانگین تعداد تعویض‌متن نشان داده است. همانطور که مشاهده می‌شود، روش پیشنهادی در حالت پویا عملکرد بهتری نسبت به حالت ایستا دارد.

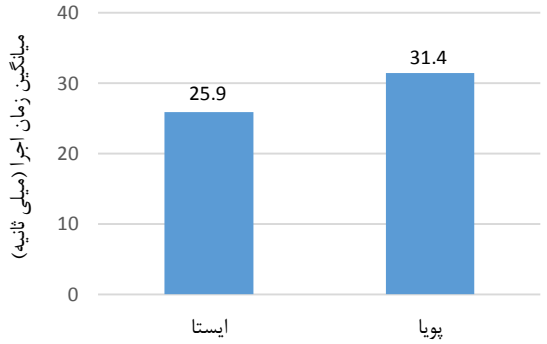
جدول (۲): خلاصه نتایج حاصل از ارزیابی کارایی روش پیشنهادی

روش	میانگین زمان برگشت	میانگین زمان انتظار	میانگین تعداد تعویض‌متن	میانگین زمان اجرای تعیین برش زمانی
DABRR	۱۵۶/۴۶	۱۲۰/۴۸	۱۰/۴۷	**۲۶**
SARR	۱۴۷/۵۹	۱۱۱/۶	۹/۳۴	۲۷
MMRRA	۱۴۰/۱۷	۱۰۴/۱۹	۷/۹۳	۳۰/۶
IRRVQ	۱۸۸/۰۷	۱۵۲/۰۹	۲۱/۴	۳۲/۸
RDQ	۱۳۹/۹۸	۱۰۳/۹۹	۷/۹۹	۲۸/۶
EDRR	۲۰۳/۴	۱۶۷/۴۲	۱۰/۲	۲۶/۲
GA-RR	**۱۳۵/۷**	۱۰۹/۵۵	۸	۳۵۵
MLDTQ	۱۳۷/۷۴	**۹۷/۵۱**	**۶/۲۴**	۳۱/۴



شکل (۱۰): میانگین زمان برگشت، میانگین زمان انتظار و میانگین تعداد تعویض‌متن حاصل از روش پیشنهادی برای دو حالت ایستا و پویا

شکل (۱۱) میانگین زمان اجرای روش پیشنهادی برای دو حالت ایستا و پویا را نشان می‌دهد. میانگین زمان اجرای روش پیشنهادی در حالت پویا ۵/۵ میلی‌ثانیه بیشتر از میانگین زمان اجرای آن نسبت به حالت ایستا می‌باشد. دلیل آن نیز اجرای الگوریتم تعیین برش زمانی پویا برای هر دور الگوریتم نوبت گردشی در مقابل تنها یکبار اجرای الگوریتم تعیین برش زمانی ایستا برای دور اول الگوریتم نوبت گردشی است.



شکل(۱۱): میانگین زمان اجرای روش پیشنهادی در حالت ایستا و پویا

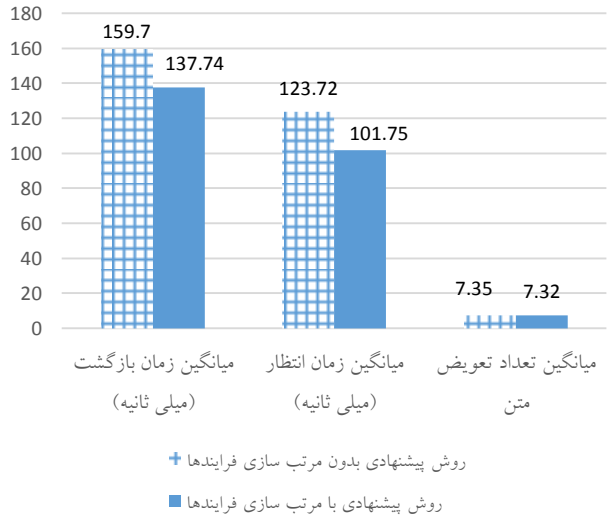
همانطور که پیش از این گفته شد، یک مجموعه تست استاندارد برای ارزیابی الگوریتم‌های تعیین برش زمانی وجود ندارد. اغلب تحقیقات موجود ارزیابی روش پیشنهادی را بر روی چند مثال نمونه و یا یک مجموعه تست ساخته شده به صورت تصادفی انجام داده‌اند. در جدول (۳)، میانگین زمان انتظار روش پیشنهادی برای مثال‌های ارائه شده در مقالات پیشین گزارش شده است. نتایج نشان می‌دهد که روش پیشنهادی در اغلب موارد کارایی بهتری در مقایسه با روش مرجع داشته است. الگوریتم GARR در برخی موارد عملکرد بهتری در مقایسه با روش پیشنهادی دارد اما همانطور که در ارزیابی زمان اجرای الگوریتم‌ها گفته شد، الگوریتم ژنتیک زمان اجرای بالایی دارد و برای کاربردهایی مانند تعیین برش زمانی مناسب نیست.

در آزمایش پایانی، کارایی روش پیشنهادی برای حالتی که فرآیندهای صف آماده در ابتدا به ترتیب صعودی بر حسب زمان انفجارشان مرتب می‌شوند و سپس الگوریتم نوبت گردشی اعمال می‌شود و حالتی که تنها الگوریتم نوبت گردشی اجرا شده و فرآیندهای صف آماده مرتب نمی‌شوند، ارزیابی شده است.

جدول (۳): کارایی روش پیشنهادی بر روی مثال‌های ارائه شده در تحقیقات موجود

نمونه تست	مرجع	میانگین زمان انتظار	
		روش پیشنهادی	روش مرجع
{48,57,73,72,45}	MMRRA [3]	۱۱۴/۲	۱۲۷/۵
{15,32,10,26,20}	IRRVQ [5]	۳۰/۲	۴۶/۲
{20,40,40,80}	SARR [6]	۴۵	۶۲/۵
{15,12,5,18,16}	GARR [7]	۲۰/۴	۲۰/۶
{12,14,10,12,11}	GARR [7]	۲۱/۸	۲۲
{80,70,20,15,75}	GARR [7]	۸۵	۷۱
{40,45,50,30,35}	GARR [7]	۷۰	۷۲
{19,40,7,25,18}	GARR [7]	۲۹	۳۵
{47,37,17,15,8}	GARR [7]	۳۳	۳۷/۶
{70,72,46,67,77}	GARR [7]	۱۱۹/۴	۱۱۹/۲
{32,26,17,45,38}	GARR [7]	۵۷	۵۳/۲
{78,57,18,28,33}	GARR [7]	۶۳/۸	۶۲/۶
{10,29,3,7,12}	RDQ [9]	۱۰/۵	۱۵/۲
{15,10,31,17}	SSR [13]	۱۹/۲۵	۲۶/۵

شکل (۱۲) نشان می‌دهد مرتب‌سازی فرآیندها بر حسب زمان انفجار منجر به بهبود معیارهای میانگین زمان برگشت و میانگین زمان انتظار فرآیندها می‌شود. همچنین مقایسه بین شکل(۱۲) و شکل‌های (۷) و (۸) نشان می‌دهد که روش پیشنهادی بدون مرتب‌سازی فرآیندها نسبت به روش EDRR که از مرتب‌سازی فرآیندها بهره نمی‌برد، میانگین زمان برگشت، میانگین زمان انتظار و میانگین تعداد تعویض‌متن کمتری دارد. در مقایسه با روش GARR که فرآیندهای صف آماده را مرتب نمی‌کند نیز روش پیشنهادی بدون مرتب‌سازی فرآیندها میانگین تعداد تعویض‌متن کمتر و میانگین زمان برگشت و انتظار بیشتری دارد. همانطور که پیشتر گفته شد، الگوریتم GARR در مقایسه با سایر روش‌ها زمان اجرای بسیار بیشتری دارد که بکارگیری آن در کاربردهای عملی را با مشکل مواجه می‌سازد.



شکل (۱۲): مقایسه روش پیشنهادی بدون مرتب‌سازی و با مرتب‌سازی فرآیندهای صف آماده

۵. نتیجه‌گیری و کارهای آتی

در این مقاله روشی کارا برای تعیین برش زمانی به صورت پویا با استفاده از یادگیری ماشین ارائه شد. در روش پیشنهادی ابتدا یک مجموعه داده که شامل تعداد فرآیندهای موجود در صف آماده و زمان انفجار هر فرآیند می‌باشد، به‌صورت تصادفی تولید شد و برای هر رکورد مجموعه داده، برش زمانی بهینه که منجر به تولید کمترین میانگین زمان برگشت و کمترین تعداد تعویض متن برای فرآیندهای موجود در صف آماده در الگوریتم نوبت گردشی شود، محاسبه گردید. سپس با کمک یک الگوریتم انتخاب نمونه، رکوردهای زائد مجموعه داده حذف شد و برای رکوردهای باقیمانده یک مجموعه آموزش شامل ویژگی‌های تعداد فرآیندها، بیشینه، کمینه، میانگین و میانه زمان انفجار فرآیندها ساخته شد. از این مجموعه آموزش برای پیش‌بینی مقدار برش زمانی برای نمونه‌های تست استفاده گردید. به این صورت که طبقه‌بند J48 با آموزش بر روی مجموعه مذکور به پیش‌بینی مقدار برش زمانی برای داده‌های مجموعه تست را ارائه داد. نتایج آزمایشات انجام گرفته نشان داد که روش پیشنهادی معیارهای میانگین زمان انتظار، میانگین زمان برگشت و میانگین تعداد تعویض متن را در مقایسه با روش‌های رایج

تعیین برش زمانی پویا بهبود می‌دهد. از جمله دیگر مزایای روش پیشنهادی امکان در نظر گرفتن محدودیت برای حداقل و حداکثر مقدار برش زمانی، بازه زمان انفجار فرآیندها و تعداد فرآیندها است. همچنین با اعمال روش پیشنهادی برای ایجاد مجموعه آموزش برای یک کاربرد خاص با ویژگی‌ها و محدودیت‌های تعریف شده می‌توان دقت تعیین برش زمانی برای آن کاربرد را بهبود بخشید. به عنوان کارهای آتی بکارگیری روش پیشنهادی برای تعیین برش زمانی در شکل‌های بهبود یافته الگوریتم نوبت گردشی و در کاربردهای جدید مانند زمانبندی وظایف در رایانش ابری و رایانش مه پیشنهاد می‌شود.

تعارض منافع: نویسندگان اعلام می‌کنند که هیچ تعارض منافعی ندارند.

مراجع

[1] Zouaoui S., Boussaid L., and Mtibaa A., "Priority based round robin (PBRR) CPU scheduling algorithm", International Journal of Electrical and Computer Engineering, 9(1): 190-202, 2019.

[2] Mishra M. K., "An improved round robin CPU scheduling algorithm", Journal of Global Research in Computer Science, 3(6): 64-69, 2012.

[3] Mora H., Abdullahi S. E., and Junaidu S. B., "Modified Median Round Robin Algorithm (MMRRA)", In 13th International Conference on Electronics, Computer and Computation (ICECCO), pp. 1-7, 2017.

[4] Saeidi S. and Baktash H. A., "Determining the optimum time quantum value in round robin process scheduling method", International Journal of Information Technology and Computer Science, 10: 67-73, 2012.

[5] Ishra M. K. and Rashid F., "An improved round robin CPU scheduling algorithm with varying time quantum", International Journal of Computer Science, Engineering and Applications, 4(4): 1, 2014.

[6] Matarneh R. J., "Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes", American Journal of Applied Sciences, 6(10): 1831-1837, 2009.

[7] Dhumall R. A., Maktum T. A., and Ragha L., "Dynamic Quantum based Genetic Round Robin Algorithm", International Journal of Advanced Research in Computer and Communication Engineering, 3(3): 5905-5908, 2014.

[8] Dash A. R. and Samantra S. K., "An optimized round robin cpu scheduling algorithm with dynamic time quantum", International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), 5(2): 7-26, 2016.

[9] Riaz R., Kazmi S. H., Kazmi Z. H., and Shah S. A., "Randomized Dynamic Quantum CPU Scheduling Algorithm", Journal of Information Communication Technologies and Robotic Applications, 9(2): 19-27, 2018.

[10] Farooq M. U., Shakoor A., and Siddique A. B., "An Efficient dynamic round robin algorithm for CPU scheduling", In 2017 International Conference on Communication, Computing and Digital Systems (C-CODE), pp. 244-248, 2017.

[11] Gull H., Iqbal S. Z., Saeed S., Alqatani M. A., and Bamarouf Y., "Design and Evaluation of CPU Scheduling Algorithms Based on Relative Time Quantum: Variations of Round Robin Algorithm", Journal of Computational and Theoretical Nanoscience, 15(8): 2483-2488, 2018.

[12] AlHeyasat O. and Herzallah R., "Estimation of Quantum Time Length for Round-robin Scheduling Algorithm using Neural Networks", In IJCCI (ICFC-ICNC), pp. 253-257, 2010.

[13] Mostafa M. and Amano H., "Dynamic round robin CPU scheduling algorithm based on K-means clustering technique", Applied Sciences, 10(15): 5134, 2020.

[۱۴] رزاق زاده ش.، نوروزی کیوی پ.، پناهی ب.، «الگوریتم ترکیبی مبتنی بر معماری گوسیپ با استفاده از SVM برای زمانبندی وظایف در رایانش ابری»، مجله محاسبات نرم، جلد ۹، شماره ۲، ص. ۸۴-۹۳، ۱۳۹۹.

[۱۵] بیرانوند ص.، زارع چاهوکی م. ع.، «مروری بر روش‌های تخمین هزینه نرم‌افزار مبتنی بر یادگیری ماشین»، مجله محاسبات نرم، جلد ۵، شماره ۱، ص. ۳۶-۶۵، ۱۳۹۵.

[۱۶] وثیقی ذاکر ا.، جلیلی س.، «پیش‌بینی ژن‌های بیماری با استفاده از دسته‌بند تک کلاسی ماشین بردار پشتیبان»، مجله محاسبات نرم، جلد ۴، شماره ۱، ص. ۷۴-۸۳، ۱۳۹۴.

[۱۷] ویسی ه.، قایدشرف ح.، ابراهیمی م.، «بهبود کارایی الگوریتم‌های یادگیری ماشین در تشخیص بیماری‌های قلبی با بهینه‌سازی داده‌ها و ویژگی‌ها»، مجله محاسبات نرم، جلد ۸، شماره ۱، ص. ۷۰-۸۵، ۱۳۹۸.

[۱۸] سالارپور ص.، انتخاب نمونه با استفاده از اتوماتاهای یادگیر، پایان‌نامه کارشناسی ارشد، دانشگاه سیستان و بلوچستان، زاهدان، ص. ۵۴-۶۰، پاییز ۱۳۹۹.