

اولویت‌دهی پیشینه‌محور موارد آزمون بر اساس کارایی کشف خطا و سالمندی در آزمون رگرسیون نرم‌افزاری

یلدا فضل‌علیزاده	علیرضا خلیلیان	محمد عبداللهی ازگمی	سعید پارسا
دانشکده مهندسی کامپیوتر	دانشکده مهندسی کامپیوتر	دانشکده مهندسی کامپیوتر	دانشکده مهندسی کامپیوتر
دانشگاه علم و صنعت ایران	دانشگاه علم و صنعت ایران	دانشگاه علم و صنعت ایران	دانشگاه علم و صنعت ایران
ya_alizadeh@comp.iust.ac.ir	khalilian@comp.iust.ac.ir	azgomi@iust.ac.ir	parsa@iust.ac.ir

به دلیل محدودیت زمان و منابع در دسترس برای گروه آزمون در دنیای واقعی، آزمون کامل اغلب غیرعملی است [۱،۲]. فنون گوناگونی برای حل مشکل هزینه زیاد آزمون رگرسیون ارائه شده است که یکی از آنها "اولویت‌دهی موارد آزمون" است [۲]. اولویت‌دهی موارد آزمون به این معناست که بر اساس یک معیار، برای دستیابی سریع‌تر به هدف آزمون، ترتیب‌دهی خاصی بر اجرای دنباله آزمون‌ها اعمال شود. هدف اصلی از اولویت‌دهی موارد آزمون، یافتن بهترین ترتیب اجرای دنباله آزمون‌ها برای "افزایش نرخ کشف خطا تا حد امکان" است [۳]. از طرفی، در محیط‌های محدودیت‌دار نیاز شدیدتری به اعمال فنونی جهت "انجام پیش‌تر مهمترین آزمون‌ها" وجود دارد. مسئله یافتن بهترین ترتیب اجرای ممکن، معادل با مسئله کوله‌پشتی 0/1 است که NP-hard بوده و راه حل قطعی برای آن وجود ندارد [۲].

متأسفانه در اغلب فنون اولویت‌دهی موجود، دید محدود شده‌ای وجود دارد و محققان در مطالعات تجربیشان عموماً آزمون رگرسیون را به‌عنوان یک آزمون "یک دفعه‌ای" در نظر گرفته‌اند، درحالی‌که آزمون رگرسیون ذاتاً آزمونی "دامه‌دار" و "دراز مدت" است و بهترین مدل برای آن، اجراهای سلسله‌وار آزمون‌ها بعد از هر تغییر نرم‌افزار است؛ به‌گونه‌ای که کارایی هر کدام از آن‌ها بر آزمون‌های بعدی مؤثر است.

در همین راستا در [۴] نخستین بار بحث "حافظه‌دار کردن" آزمون رگرسیون و استفاده از اطلاعات پیشینه کارایی موارد آزمون در اولویت‌دهی مجموعه آزمون، مطرح گردید تا بتوان بر مشکل غیرقابل پیش‌بینی بودن کارایی فن اولویت‌دهی موارد آزمون غلبه کرد. فن مکاشفه‌ای ارائه شده در [۴] در واقع ترکیبی است از "فن انتخاب مبتنی بر سابقه در آزمون رگرسیون" و سپس اولویت‌دهی موارد آزمون. به این ترتیب که در هر گام از پیشینه و اطلاعات سابقه اجرای موارد آزمون، برای انتخاب‌های بعدی موارد آزمون استفاده می‌شود و در هر مرحله زیرمجموعه‌ای از مجموعه آزمون اولیه را برای اولویت‌دهی و اجرا بر روی نسخه جدید نرم‌افزار انتخاب می‌کند. نشان داده شده که چنین مکاشفه‌ای می‌تواند با گذشت اجراهای طولانی، هزینه را کاهش داده و کارایی آزمون رگرسیون را در محیط‌های محدودیت‌دار کاهش دهد.

چکیده: در مرحله نگهداری نرم‌افزار، پس از هر تغییر جزئی کد، آزمون‌های جدیدی طراحی می‌شوند، که اجرای آنها و تمام موارد آزمون قبلی، برای اعتبارسنجی نسخه جدید نرم‌افزار ضروری است. اجرای مکرر حجم زیاد آزمون‌ها، هزینه هنگفتی را بر توسعه‌دهندگان نرم‌افزار تحمیل می‌کند و به دلیل محدودیتهای منابع، آزمون کامل، اغلب غیرعملی است. برای حل این مشکل، از اولویت‌دهی موارد آزمون استفاده می‌شود. مشکل اغلب فنون اولویت‌دهی موجود این است که اجرای مکرر آزمون‌ها، تأثیرگذاری کارایی آزمون‌ها در هر گام بر کارایی مراحل بعدی آزمون و نیز محدودیت زمان و منابع آزمون محیط‌های واقعی اجرای آزمون را نادیده می‌گیرند. به دلیل همین محدودیت‌ها، از "پیشینه اجراهای قبلی موارد آزمون" در اولویت‌دهی آنها در اجراهای بعدی استفاده می‌شود. در روش ارائه شده، اولویت‌دهی بر پایه دو معیار سابقه کشف خطا و سالمندی موارد آزمون انجام می‌شود. نتایج مطالعات تجربی، حاکی از بهبود عملکرد در کشف خطا و پایداری بیشتر نتایج روش فوق است.

واژه‌های کلیدی: آزمون رگرسیون نرم‌افزار، اولویت‌دهی موارد آزمون، اولویت‌دهی پیشینه‌محور، کارایی تاریخی کشف خطا، سالمندی موارد آزمون.

۱- مقدمه

آزمون فرایندی است که برای آشکارسازی عیوب نرم‌افزار به کار می‌رود و نیز مشخص می‌کند که آیا نرم‌افزار با توجه به یکسری خصوصیات منتخب، به سطحی مشخصی از کیفیت رسیده است یا خیر [۱]. بسیاری از هزینه‌های توسعه نرم‌افزار، مربوط به آزمون‌های مکرر در "مرحله نگهداری نرم‌افزار در حال تکامل و اصلاح" است. زیرا ایجاد هر تغییر، حذف یا اضافه در کد نرم‌افزار، نسخه جدیدی از آن را بدست می‌دهد، که نیازمند طراحی آزمون‌های جدید در هر گام و انجام مجدد تمام آزمون‌های قبل است. این آزمون‌های مکرر که برای اطمینان از عدم تأثیرپذیری نامطلوب بخشهای اصلاح نشده کد از تغییرات صورت می‌گیرد "آزمون رگرسیون نرم‌افزاری" نامیده می‌شود.

1 می‌گیرد. اثر این تعریف برای H_{tc} این است که اجرای موارد آزمون را که بندرت و یا اصلاً خطایی آشکار نکرده‌اند محدود می‌کند.

ملاحظه می‌شود که فن پیشنهاد شده در (۱)، در حقیقت یک "انتخاب پیشینه‌محور" در آزمون رگرسیون است و لازمه اولویت‌دهی موارد آزمون این است که از یکی از فنون موجود اولویت‌دهی بر روی نتایج این انتخاب استفاده شود. نکته دیگر در مورد این فن این است که در رابطه (۱) تنها استفاده از یکی از تعاریف ذکر شده در دو بخش بالا برای H_{tc} برای پیشنهاد اجرای موارد آزمون، به‌عنوان استراتژی تعیین احتمالات انتخاب موارد آزمون امکان‌پذیر است.

انتخاب پارامتر h_k (مجموعه مشاهدات تاریخی اجراهای هر مورد آزمون از اولین تا k امین اجرا) که تنها با دو مقدار 0 یا 1، آن‌هم فقط بر اساس اجرای مرحله قبلی هر مورد آزمون در تعیین احتمال انتخاب جاری آن، یعنی این‌که مورد آزمون اخیراً اجرا نشده باشد و یا این‌که در اجرای قبلی خطا آشکار کرده است یا نه، ملاک مناسبی برای بالا رفتن احتمال انتخاب آن در اجرای بعدی آن نیست.

۳- روش پیشنهادی

چند عامل در افزایش اولویت یک مورد آزمون در انتخاب‌های بعدی مؤثر است. یکی از این عوامل، کارایی کشف خطای آن در تعداد دفعاتی است که مورد آزمون اجرا شده است. باید توجه داشت که تعداد اجرای موارد آزمون و تعداد دفعاتی که سبب آشکار شدن وجود خطا در نرم‌افزار شده‌اند را نمی‌توان جدا از هم در نظر گرفت و این دو توأم با هم، به‌صورت یک عامل تأثیرگذار در پیشنهاد کارایی موارد آزمون در اولویت‌دهی نقش دارند. بنابراین اگر در k امین اجرای آزمون رگرسیون، تعداد دفعاتی که اجرای مورد آزمون tc بر روی نسخه جدید نرم‌افزار با شکست مواجه شده باشد را fc_k و کل تعداد دفعات اجرای آن تا این مرحله را با ec_k نشان دهیم، برای هر مورد آزمون tc می‌توان رابطه اولویت مورد آزمون با کارایی آن را در k امین اجرا به‌صورت زیر نوشت:

$$PR_k \approx \frac{fc_k}{ec_k} \quad (2)$$

$$ec_k = \sum_{i=1}^{k-1} e_i \quad (4) \quad fc_k = \sum_{i=1}^{k-1} f_i \quad (3)$$

$$f_i = \begin{cases} 1 & \text{اگر مورد آزمون در جلسه آزمون } i \text{ ام خطا آشکار کند} \\ 0 & \text{در غیر این صورت} \end{cases}$$

$$e_i = \begin{cases} 1 & \text{اگر مورد آزمون در جلسه آزمون } i \text{ ام اجرا شده باشد} \\ 0 & \text{در غیر این صورت} \end{cases}$$

به‌عبارت دیگر، اولویت پیشنهادی‌محور هر مورد آزمون در هر گام، با نسبت آشکارسازی خطا بر کل دفعات اجرای آن متناسب است.

علاوه بر پیشنهاد کارایی موارد آزمون، عامل مؤثر دیگر بر اولویت‌دهی این است که موارد آزمون برای مدت طولانی اجرا نشده باقی

مشکل فن پیشینه‌محور جاری این است که در به‌کارگیری اطلاعات پیشینه اجرای موارد آزمون، تنها اثر اخیراً اجرا شدن یا نشدن، و یا آشکارسازی خطا یا عدم آن در اجرای قبلی، آن‌هم به صورت صفر و یک در نظر گرفته شده است. در حالی‌که برای اولویت‌دهی کارایی پیشنهادی‌محور، هم کارایی کشف خطا در طول اجراها باید در نظر گرفته شود و هم باید از متروک شدن دائمی یکسری از آزمون‌ها جلوگیری شود. در روش پیشنهادی، اولویت‌دهی موارد آزمون، بر اساس سابقه کارایی موارد آزمون در کشف خطا و نیز سالمندی موارد آزمون انجام شده و سپس به ارزیابی این روش بر روی برنامه محک پرداخته می‌شود.

۲- کارهای پیشین

نخستین بار در سال ۱۹۹۷ مسئله اولویت‌دهی موارد آزمون به این نام، توسط چهار تن از محققان مطرح گردید [۵] که این فن، موارد آزمون را بر اساس پوشش کد اصلاح شده انتخاب و سپس اولویت‌دهی می‌نمود. پس از آن، در سال‌های بعد اولویت‌دهی موارد آزمون توسط چند تن دیگر از محققان این حوزه ادامه یافت [۶-۱۰] و انواع فنون کد محور برای اولویت‌دهی موارد آزمون توسط آنها ارائه شد. بیشتر فنون ارائه شده، کدمحور و بر پایه روش‌های حریصانه‌اند. سایر فنون موجود، اولویت‌دهی مدل محور [۱۱] و پیشنهادی‌محور [۱۲، ۴] هستند.

۱-۲ اولویت‌دهی پیشنهادی‌محور موارد آزمون

در [۴] اولویت‌دهی پیشنهادی‌محور موارد آزمون به این صورت تعریف شده است: فرض می‌کنیم احتمال انتخاب هر مورد آزمون tc در زمان t ، $P_{tc,t}(H_{tc}, \alpha)$ در نظر گرفته شود، به‌طوری‌که H_{tc} یک مجموعه از مشاهدات مرتب شده بر اساس زمان، به‌صورت $\{h_1, h_2, \dots, h_t\}$ باشد، که از اجراهای قبلی برای هر مورد آزمون بدست آمده‌اند. برای هر کدام از موارد آزمون موجود در دنباله آزمون، مجموعه مشاهدات مرتب یعنی h ها نشانگر عملکرد مورد آزمون tc در طول پیشنهادی‌محور اجرای آن است. بر این اساس، احتمال انتخاب هر مورد آزمون در مجموعه T' بر اساس پیشنهادی‌محور اجرای آن به‌صورت روابط زیر قابل تعریف است:

$$P_k = \alpha h_k + (1 - \alpha) P_{k-1} \quad (1)$$

$$k \geq 1, 0 \leq \alpha < 1$$

ثابت هموارکننده در رابطه (۱) برای وزن‌دهی به تاریخچه مشاهدات به‌کار رفته است. بر اساس رابطه (۱) اولویت‌دهی پیشنهادی‌محور بر اساس تعاریف متفاوت از $H_{tc,t}$ می‌تواند انجام شود.

تاریخچه اجرای آزمون. برای هر جلسه آزمون i که مورد آزمون tc در آن اجرا شود، h_i در احتمال انتخاب بعدی مقدار 0 و در غیر این صورت 1 می‌گیرد. این انتخاب سبب گردش بین موارد آزمون در جریان اجراهای طولانی می‌شود.

۲) کارایی کشف خطای نشان داده شده. برای هر جلسه آزمون i که مورد آزمون tc در آن خطا آشکار کند، h_i مقدار 0 و در غیر این صورت

بنابر (۲)، (۶) و (۷)، رابطه اولویت‌دهی در k امین گام اجرا برای هر مورد آزمون، به صورت زیر نوشته می‌شود:

$$PR_k = \alpha \frac{fc_k}{ec_k} + (1 - \alpha) PR_{k-1} + (1 - \alpha)^2 h_k \quad (۸)$$

$$0 \leq \alpha < 1, \quad k \geq 1$$

ضریب α یک ثابت نرم‌کننده برای کنترل میزان تأثیر عوامل فوق در اولویت‌دهی است. ضریب $(1 - \alpha)^2$ به عنوان ضریب h_k ، که نسبت به ضرایب α و $1 - \alpha$ کوچک‌کننده‌تر است، باعث می‌شود میزان تأثیر سالمندی که بر حسب شرایط اجرای مورد آزمون هر مرتبه یک واحد افزایش خواهد داشت، در مقابل مقادیر ec_k/fc_k و PR_{k-1} که هر دو مقادیری کسری هستند، به اشتباه تأثیر بالایی در اولویت‌دهی نداشته باشد و تأثیر عوامل فوق را نپوشاند.

در رابطه (۸)، PR_k اولویت موارد آزمون در k امین گام اجرا است. پس از اولویت‌دهی، با توجه به زمان و منابع محدود آزمون، به تعداد کافی از موارد آزمون با شروع از اولویت‌های بالا اجرا می‌گردد.

۴- ارزیابی روش پیشنهادی

برای ارزیابی فنون اولویت‌دهی آزمون، متریک APFD که متوسط وزن‌دار درصد خطاهای کشف شده در اجرای مجموعه آزمون است، ارائه شده است [۲] که فرمول محاسبه آن به این صورت است:

$$APFD = 1 - \frac{TF_1 + TF_2 + \dots + TF_m}{nm} + \frac{1}{2n} \quad (۹)$$

در رابطه (۹)، n تعداد موارد آزمون و m تعداد خطاهای موجود را نشان می‌دهد. هر TF_i در رابطه فوق محل مورد آزمون را در دنباله مرتب موارد آزمون در رشته اولویت‌دهی شده نشان می‌دهد که برای اولین بار خطای i ام را آشکار نموده است.

به منظور تحلیل و ارزیابی روش پیشنهادی، از مجموعه زیرممنس [۱۳] استفاده شده است. استفاده از این محک به سبب متداول بودن استفاده از آنها در ارزیابی فنون اولویت‌دهی موارد آزمون است. برای هر برنامه تعدادی نسخه‌های خطادار با خطاهای کاشته شده در نظر گرفته شده است.

در مطالعه تجربی انجام شده روی برنامه tcas از مجموعه زیرممنس، از مجموعه‌های حاوی چند خطا، تعداد ۳۰ نسخه چندخطایی به تصادفی انتخاب، و به عنوان ۳۰ مرحله از تکرار آزمون رگرسیون از آنها استفاده شده است. به علاوه روش اولویت‌دهی پیشنهادی، روی ۱۰۰۰ مجموعه با پوشش انشعاب اجرا شده است. همچنین در پایان هر گام اجرا و پس از کشف خطاها، متریک APFD برای رشته اولویت‌دهی شده موارد آزمون محاسبه شده است.

در ارزیابی فوق، فرض می‌شود که به دلیل محدودیت در منابع آزمون، بتوان تنها ۷۰٪ از کل مجموعه اولویت‌دهی شده را اجرا کرد. به این ترتیب، برخلاف مطالعات موجود اولویت‌دهی که به اجرای کل

نمانند و اطمینان داشته باشیم که همه موارد آزمون بالاخره اجرا شده و خطاهای مربوطه آشکار خواهند شد. در همین راستا عامل h_k در k امین اجرا برای هر مورد آزمون به صورت زیر تعریف می‌شود:

$$h_k = \begin{cases} 0 & \text{اگر مورد آزمون در جلسه آزمون } k-1 \text{ ام اجرا شده باشد} \\ h_{k-1} + 1 & \text{در غیر این صورت} \end{cases} \quad (۵)$$

یعنی در حقیقت عامل h_k همانند یک "شمارنده" عمل می‌کند. برای جلوگیری از سالمندی موارد آزمون، شمارنده‌ای برای هر کدام تخصیص می‌یابد که وظیفه آن شمردن سن (تعداد دفعات عدم اجرا) است. لذا می‌توان نوشت:

$$PR_k \approx h_k \quad (۶)$$

به عبارتی شمارنده (سن) هر مورد آزمون و در نتیجه اولویت اجرای آن در هر بار اجرا نشدن، افزایش می‌یابد تا جز موارد آزمون اجرایی قرار بگیرد و اجرا شود. در صورت اجرای مورد آزمون، شمارنده (سن این مورد آزمون) برابر 0 می‌شود و همین عملیات تکرار می‌شود. به این ترتیب هیچ مورد آزمون برای مدت طولانی، اجرا نشده باقی نمی‌ماند و خطای مربوطه را آشکار می‌کند.

در نهایت سومین عاملی که بر اساس مطالعات تجربی [۴]، باید در اولویت‌دهی موارد آزمون در هر گام اجرا دخالت داشته باشد، اولویت و رتبه پیشین هر مورد آزمون در اجرای قبلی آزمون رگرسیون است:

$$PR_k \approx PR_{k-1} \quad (۷)$$

دلیل استفاده از اولویت اجرای قبلی مورد آزمون و تعریف رابطه بصورت بازگشتی این است که اولاً باعث "ترمتر شدن انتخابهای موارد آزمون" در گام‌های متوالی اجرا و عدم بروز تغییرات شدید در رشته آزمون اجرایی در هر گام اجرا نسبت به گام قبلی می‌شود. ثانیاً در مواردی که سابقه اجرای موارد آزمون یکسان باشد، یعنی نسبت تعداد دفعات آشکارسازی خطا بر اجرای آنها و نیز وضعیت سالمندی موارد آزمون یکسان باشد، باید عامل دیگری در تعیین اولویت صحیح بین آنها قائل شد. این مهم با استفاده از اولویت اجرای قبلی موارد آزمون تأمین می‌شود. در رابطه پیشنهادی مقدار PR_0 را می‌توان بر حسب معیارهای گوناگون، مثلاً درصد پوشش کد موارد آزمون محاسبه کرد که به این ترتیب، اثر پوشش موارد آزمون به دلیل بازگشتی بودن رابطه، در تمامی اولویت‌دهی‌های بعدی هم شرکت می‌کند، که شاخص بسیار مناسبی برای قدرت موارد آزمون، در تمام فنون اولویت‌دهی کدمحور محسوب می‌شود. زیرا منطقی است که فرض کنیم موارد آزمون که درصد بیشتری از کد برنامه را می‌پوشانند، احتمال پوشش قسمت‌های خطادار و آشکارسازی خطاها در آنها بیشتر است، لذا اولویت بالاتری به آنها داده می‌شود. لذا رابطه اولویت‌دهی پیشنهادی در حقیقت یک فن اولویت‌دهی است که اساس آن، پیشینه کارایی موارد آزمون در کشف خطا بوده و پوشش محور نیز محسوب می‌شود.

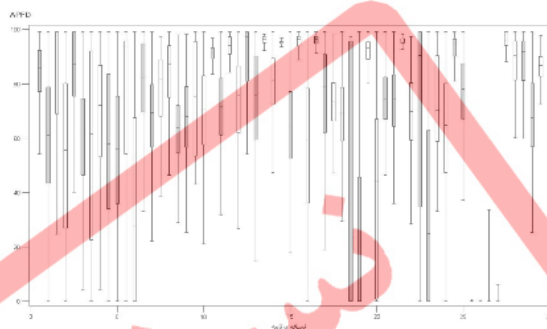
مختلف ضریب کنترلی α و نیز تأثیر کسر مناسب اجرای آزمون برای بیشینه نمودن متوسط کشف خطای آزمون در محیط‌های محدودیت دار از نظر منابع مورد نظر است.

مراجع

- [1] Burnstein, I. Practical software testing: a process-oriented approach. ISBN 0-387-95131-8, Springer-Verlag, New York, Inc, 2003.
- [2] Rothermel, G., Untch, R. H., Chu, C., Harrold, M. J., "Test Case Prioritization: An Empirical Study", *IEEE Proceedings International Conference on Software Maintenance*, August 1999, pages 179-188.
- [3] Rothermel, G. R., Untch, H., Chu, C. Harrold, M. J., "Prioritizing Test Cases for Regression Testing", *IEEE Transactions on Software Engineering*, pages 102-112, Aug. 2001, Vol. 27, No. 10, October 2001, pages 929-948.
- [4] Kim, J. M., Porter, A. "A History-Based Test Prioritization Technique for Regression Testing in Resource Constrained Environments", *Proc. 24th Int'l Conf. Software Eng.*, pp. 119-129, 2002.
- [5] Wong, W. E., Horgan, J. R., London, S., Agrawal, H., "A study of effective regression testing in practice", *In Proceedings of the Eighth Int. Symp.m on Sof. Rel Eng.* Pages 230-238, November 1997.
- [6] Elbaum S., Malishevsky, A. G., Rothermel, G., "Test Case Prioritization: A Family of Empirical Studies", *IEEE Transactions on Software Engineering*, Vol. 28, No. 2, pages 159-182, February 2002.
- [7] Elbaum S., Kallakuri, P., Malishevsky, A. G., Rothermel, G., Kanduri, S., "Understanding the Effects of Changes on the Cost-Effectiveness of Regression Testing Techniques", *Journal of Software Testing, Verification, and Reliability*. V. 13, no. 2, June, 2003, pages 65-83.
- [8] Li, Z., Harman, M., Hierons, R., "Search algorithms for regression test case prioritization", *IEEE Transactions on Software Engineering*, 33 (4) (2007) 225-237.
- [9] Malishevsky, A. G., Elbaum, S., Rothermel, G., Kanduri, S., "Selecting a Cost-Effective Test Case Prioritization Technique", *Software Quality Journal*, 2004 Kluwer Academic Publishers. Manufactured in The Netherlands, 12, 185-210, 2004.
- [10] Malishevsky, A. G., Ruthruff, J. R., Rothermel, G., Elbaum S., "Cost-cognizant test case prioritization", *Technical report*, Department of Computer Science and Engineering, University of Nebraska-Lincoln, March 2006.
- [11] Korel, B., Koutsogiannakis, G., Tahat, L. H., "Model-based test prioritization heuristic methods and their evaluation", *Proceedings of the 3rd international workshop on Advances in model-based testing*, 2007, Pages: 34-43, Publisher ACM New York, NY, USA.
- [12] Park, H., Ryu, H., Baik, J., "Historical Value-Based Approach for Cost-Cognizant Test Case Prioritization to Improve the Effectiveness of Regression Testing", *paper appears in: Secure System Integration and Reliability Improvement*, 2008. SSIRI '08. Second International Conference on 14-17 July 2008, page(s): 39-46, Japan.
- [13] Rothermel, G., Elbaum, S., Kinneer, A., Do, H., *Software-artifact infrastructure repository*, <http://www.cse.unl.edu/~ga-leilo/sir>.

زیرمجموعه اولویت‌دهی شده می‌پردازند روش پیشنهادی، با اجرای تعداد کمتری از موارد آزمون به "صرفه‌جویی در منابع آزمون" نیز کمک کرده است.

برای مقایسه و نمایش کارایی روش اولویت‌دهی پیشنهادی از نمودار جعبه‌ای استفاده شده است که نتایج "متوسط کشف خطا برای ۱۰۰۰ دنباله آزمون با پوشش انشعاب با روش پیشنهادی" را با "متوسط کشف خطای ۱۰۰۰ دنباله آزمون با اولویت تصادفی" مقایسه می‌کند. برای این منظور، ۳۰ جفت جعبه مجزا رسم شده است که همگی در یک نمودار در شکل ۱ نشان داده شده‌اند.



شکل ۱: نمودار جعبه‌ای مقایسه روش پیشنهادی با روش اولویت تصادفی موارد آزمون برای ۳۰ نسخه و ۱۰۰۰ دنباله آزمون برای هر نسخه.

در شکل ۱، هر کدام از زوج جعبه‌ها با رنگ یکسان، نتایج متوسط کشف خطا (APFD) را برای یک نسخه خاص، توسط روش اولویت‌دهی پیشنهادی (جعبه سمت چپ) و روش اولویت تصادفی (جعبه سمت راست)، نشان می‌دهند. گستره جعبه‌ها، در واقع گستردگی مقادیر متوسط کشف خطا (ملاک کارایی روش اولویت‌دهی) را نشان می‌دهد. همان‌گونه که در شکل ۱ ملاحظه می‌شود، روش پیشنهادی تا حد قابل ملاحظه‌ای چه از نظر کشف سریعتر خطاها (بالتر بودن جعبه) و چه از نظر پایداری عملکرد در کشف خطا در اولویت‌دهی دنباله‌های گوناگون (گستره کمتر جعبه)، بهتر عمل کرده است.

۵- نتیجه‌گیری

در این مقاله روش جدیدی برای اولویت‌دهی مبتنی بر پیشینه موارد آزمون، با در نظر گرفتن محدودیت‌های واقعی در محیط اجرا ارائه شده است. این روش در راستای مشکل هزینه زیاد اجرای تکراری کل موارد آزمون اتخاذ شده است. در روش پیشنهادی، آن‌چه که سبب بالا رفتن اولویت مورد آزمون رشته آزمون اجرایی می‌شود، بالا بودن احتمال اجرای اخیر آن، سابقه کارایی بالای آن در جریان اجرای طولانی مدت در کشف خطا و مدت طولانی اجرا نشدن مورد آزمون و مسئله سالمندی مورد آزمون است.

در راستای این کار می‌توان از گسترش مطالعه تجربی بر روی برنامه‌های محک دیگر، برنامه‌های بزرگتر و نیز خطاهای واقعی به‌عنوان کار آتی نام برد. همچنین انجام مطالعات دقیق‌تر در مورد تأثیر مقادیر