

# اولویت بندی مورد آزمایشی براساس الگوریتم ژنتیک و پوشش نقاط

## آزمایش

### چکیده

با بهینه سازی ترتیب اجرایی موارد آزمایشی، شیوه های اولویت بندی مورد آزمایشی می توانند بطور موثری بازدهی آزمایش نرم افزار را بالا برند. اولویت بندی مورد آزمایشی، در حال تبدیل شدن به یک موضوع قابل توجه در تحقیق آزمایش نرم افزار است. این مقاله با ترکیب الگوریتم ژنتیک با پوشش نقاط آزمایش، برخی نتایج تحقیقی معناداری را در اولویت بندی مورد آزمایشی به ویژه برای آزمایش کارکردی بدست می آورد. ابتدا، دو ارزیابی جدید اولویت بندی مورد آزمایشی APTC و اصلاحیه آن APRC-C را ارائه می دهد. این ارزیابی ها چون متمرکز بر پوشش نقاط آزمایش هستند، برای آزمایش جعبه سیاه مناسب تر می باشند. سپس، یک روش اولویت بندی مورد آزمایشی براساس الگوریتم ژنتیک ارائه می دهد که نمایش، انتخاب، تقابل و جهش آن برای آزمایش جعبه سیاه طراحی می شود. در نهایت، روش پیشنهادی را با داده های آزمایشات بازبینی می کند. نتایج آزمایشگاهی نشان می دهند که روش پیشنهادی می تواند اثر مطلوب را ایجاد کند.

**کلیدواژه ها:** آزمایش نرم افزار، اولویت بندی مورد آزمایشی، الگوریتم ژنتیک، تابع ارزیابی، آزمایش جعبه سیاه،

مهندسی نرم افزار

### 1. مقدمه

آزمایش نرم افزار یکی از مهم ترین ابزار برای تضمین کیفیت نرم افزار است. آمار نشان می دهد که آزمایش نرم افزار، بطور کلی بیش از 50٪ هزینه کلی توسعه نرم افزار را در بر می گیرد (1). با افزایش پیچیدگی نرم افزار، آزمایش نرم افزار دشوارتر و هزینه برتر می شود. چگونگی انتخاب چند مورد آزمایشی از مجموعه عظیم موجود برای آزمایش نرم افزار، بطور موثری تبدیل به یک مسئله مهم شده است.

فناوری اولویت بندی مورد آزمایشی<sup>1</sup> (TCP) یکی از مهم ترین دستورالعمل های تحقیق در مورد آزمایش نرم افزار است. براساس سطح اهمیت مبتنی بر برخی معیارهای خاص، TCP مورد آزمایشی را ترتیب داده و سپس آنها را بطور متوالی اجرا می کند. با بهینه سازی ترتیب اجرا برای موارد آزمایشی، TCP می تواند بشکلی موثر بازدهی آزمایش نرم افزار را افزایش دهد.

یک توضیح کلی برای TCP بدین شرح است: برای یک مجموعه مورد آزمایشی مشخص  $T$ ، جایگشت  $PT$  متعلق به  $T$  و یک تابع دسته بندی  $f: PT \rightarrow R$  برای یافتن  $T' \in PT$ ، فرض نمایید که  $f(T') \geq f(T'')$  برای  $\forall T'' \in PT (T'' \neq T')$  برقرار است. یک هدف معمول دسته بندی، به حداکثر رساندن سرعت تشخیص نقص اولیه است. سایر اهداف دسته بندی پوشش کد، سرعت پرریسک تشخیص نقص، سرعت تشخیص نقص کد اصلاحی مربوطه و قابلیت اعتماد سیستم و مواردی از این قبیل هستند.

تحقیقات کنونی روی شیوه های اولویت بندی مورد آزمایشی برپایه کد تمرکز دارند. معمولاً براساس توان پوششی برنامه (مانند عبارت، شاخه یا تابع) در هر مورد آزمایشی، وزن ها تعیین شده و روش گریدی<sup>2</sup> برای هدایت مرتب سازی آزمایش بکار می رود (2). برخی محققین سعی می کنند از روش های معمول فراگیری دستگاهی برای بهبود تأثیر اجرای فناوری TCP بهره گیرند که از این روش، جستجوی فرااکتشافی (3)، شبکه های بایسین (4) و تحلیل دسته (5) است.

با شبیه سازی فرآیند ارزیابی بیولوژیکی، الگوریتم ژنتیک (GA) راه حل بهینه را برای مسئله بهینه سازی جستجو می کند. GA یک مجموعه از راه حل های محتمل را در نظر می گیرد. این روش، بطور تصادفی در کل فضای جستجو نمونه برداری کرده و هر نمونه را براساس تابع انطباق آن ارزیابی می کند. در الگوریتم ژنتیک، برخی اپراتورها مانند انتخاب، تقابل و جهش بکار می روند که همواره تکرار می شوند (هرتکرار معادل یک چرخه از تکامل بیولوژیکی است) تا یک راه حل بهینه کلی یافت شود تا زمانی که شرط پایان چرخه برقرار شود.

در نسل داده های آزمایشی، فضای جستجوی GA دامنه ورودی نرم افزار و راه حل بهینه، برخی داده های آزمایشی برای تحقق اهداف آزمایشی مشخص می باشد. GA اغلب در آزمایش ساختاری با در نظر گیری مسیر حرکت به عنوان هدف بهینه بکار می رود (9-6). تحقیقات راجع به بهره گیری از GA در آزمایش کارکردی چندان زیاد نیستند، از جمله یک روش برپایه یک مشخصه زبانی Z و روش مبتنی بر پیش شرط ها یا پس شرط ها (11). بدلیل هزینه بالای رسمی سازی، این روش ها برای سیستم های بزرگ روش های دشواری هستند.

این مقاله با هدف آزمایش کارکردی، یک ارزیابی جدید اولویت بندی مورد آزمایشی بنام درصد میانگین پوشش نقاط آزمایش<sup>3</sup>(APTC) و اصلاحیه آن APRC-C را پیشنهاد می دهد. سپس، با استفاده از APTC یا APRC-C به عنوان تابع انطباق، ما یک روش اولویت بندی مورد آزمایشی بر پایه الگوریتم ژنتیک را پیشنهاد دادیم. در پایان، روش خود را با بررسی های شبیه سازی بازبینی نمودیم.

## 2. ارزیابی اولویت بندی مورد آزمایشی

### 1-2 برخی توابع ارزیابی موجود

هدف از آزمایش نرم افزار این است که نقایص نرم افزاری تا بیشترین حد ممکن و با بیشترین سرعت ممکن ردیابی شوند. در مقایسه با آزمایش نرم افزار با توالی تصادفی، یکی از بزرگترین فواید TCP، توانایی بررسی سریعتر خطاهاست. روترمال به این اشاره می کند که TCP را می توان با ارزیابی رابطه بین تعداد موارد آزمایشی اجرا شده و خطاهای تشخیص داده شده ارزیابی کرد.

روترمال یک ارزیابی تحت عنوان APFD (درصد میانگین تشخیص عیب) را پیشنهاد نمود. البایوم (12) رابطه APFD را ارائه داد. فرض نمایید که مجموعه مورد آزمایشی T که شامل n مورد آزمایشی و m نقص است قابل تشخیص باشد. برای توالی موارد آزمایشی داده شده،  $TF_i$  نشان دهنده تقدم اولین مورد آزمایشی برای تشخیص i امین نقص در توالی می باشد، در نتیجه یک  $APFD = 1 - \frac{TF_1 + TF_2 + \dots + TF_m}{nm} + \frac{1}{2n}$  وجود دارد. APFD بین صفر تا 100 درصد تغییر می کند که هرچه مقدار آن بیشتر گردد، تشخیص نقص سریعتر انجام می شود. چون نمی توان

اطلاعات تشخیص نقص موارد آزمایشی را پیش بینی کرد، لی ژنگ و همکاران (13)  $APBC^4$  (درصد میانگین پوشش بلوکی)،  $APDC^5$  (درصد میانگین پوشش تصمیم) و  $APSC^6$  (درصد پوشش عبارت) را پیشنهاد کردند که می توانند توان پوششی برای برنامه ریزی (به ترتیب بلوک، تصمیم، عبارت) توالی موارد آزمایشی را ارزیابی کنند. روابط این ارزیابی ها مشابه  $APFD$  می باشد بطور مثال

$$APBC = 1 - \frac{TB_1 + TB_2 + \dots + TB_m}{nm} + \frac{1}{2n}$$

است که در آن  $TB_i$  بیانگر تقدم اولین مورد آزمایشی برای پوشش  $i$ امین بلوک است. زمانی که اطلاعات پوششی را بتوان از طریق روش های تحلیل پوشش بدون اجرای موارد آزمایشی بدست آورد، در این صورت می توان از  $APBC$ ،  $APDC$  و  $APSC$  پیش از اینکه آزمایش نرم افزار پایان یابد استفاده کرد. با این حال، این ارزیابی ها به وضوح برای آزمایش ساختاری یا آزمایش جعبه سفید مناسب تر هستند. بنابراین، ما یک ارزیابی جدید اولویت بندی مورد آزمایشی برپایه پوشش نقاط آزمایش پیشنهاد می کنیم.

## 2-2 درصد میانگین پوشش نقاط آزمایش ( $APTC$ )

در آزمایش جعبه سیاه، آزمایش کنندگان موارد آزمایشی را براساس مشخصات نرم افزار طرح می کنند. در ابتدا، الزامات نرم افزاری را به الزامات آزمایش نرم افزار با استفاده از تحلیل الزامات تبدیل می کنند. در مرحله دوم، الزامات آزمایش را به بسیاری از نقاط آزمایش تجزیه می کنند و سپس موارد آزمایشی مورد هدف را به ترتیب در نقاط آزمایش طراحی کرده و سپس یک مجموعه موارد آزمایشی را تشکیل می دهند. هر مورد آزمایشی می تواند متناسب با یک یا تعداد بیشتری نقاط آزمایش باشد. برای یک مجموعه موارد آزمایشی  $\Phi = \{T_1, T_2, \dots, T_m\}$ ،  $APTC$  را بصورت زیر تعریف کنید:

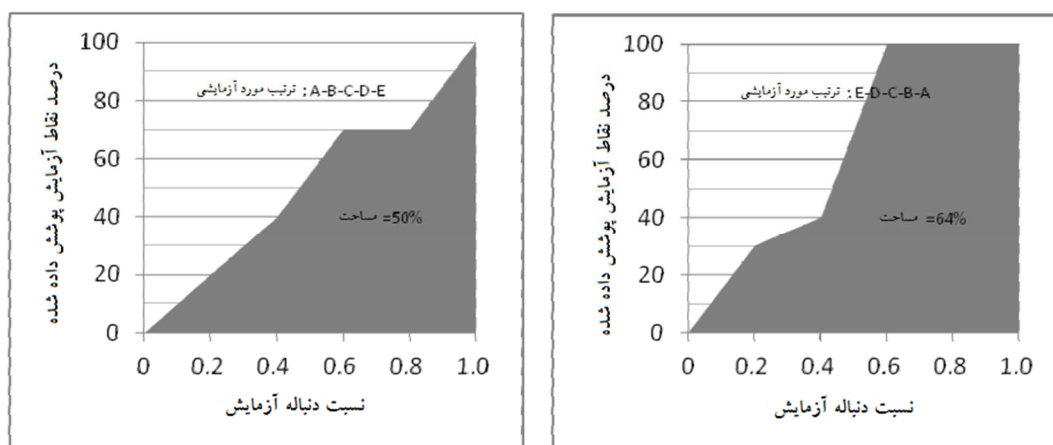
$$APTC = 1 - \frac{TT_1 + TT_2 + \dots + TT_m}{nm} + \frac{1}{2n} \quad (1)$$

<sup>4</sup> average percentage of block coverage

<sup>5</sup> average percentage of decision coverage

<sup>6</sup> average percentage of statement coverage

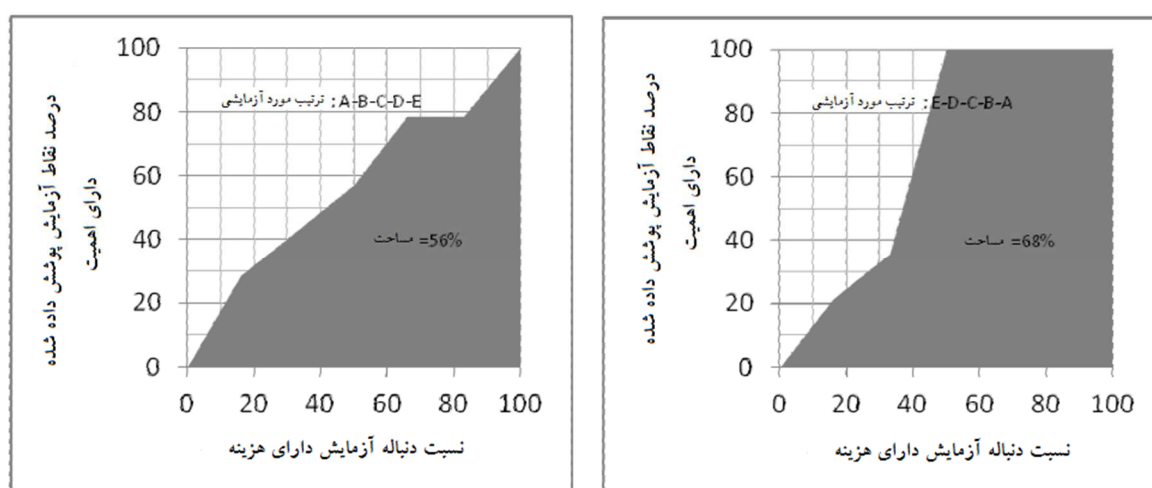




شکل 1. APTC در توالی های مختلف مورد آزمایشی

فرض کنید که هزینه مورد آزمایشی B دو برابر سایر موارد آزمایشی باشد و میزان اهمیت نقاط آزمایش پوشش داده شده با مورد آزمایشی B (یعنی نقطه آزمایش 1، 5، 6، 7) دو برابر نقاط آزمایش دیگر است. براساس رابطه، محاسبه APTC-C مربوط به A-B-C-D-E و E-D-C-B-A که برابر 56٪ و 68٪ است، ساده می باشد.

پس از جایگزینی مختصات افقی و قائم با " کسر دنباله آزمایشی با هزینه " و " درصد نقاط آزمایش پوشش داده شده با اهمیت "، می توانیم نتیجه گیریم که دیگرام APTC-C مشابه APTC است که در شکل 2 مشاهده می شود.



شکل 2. APTC-C برای توالی های مختلف مورد آزمایشی

در مورد تفاوت های بزرگ در اهمیت نقاط آزمایش و یا هزینه های موارد آزمایشی، APTC-C مناسب تر از APTC است.

### 3. اولویت بندی مورد آزمایشی براساس الگوریتم ژنتیک

#### 3-1 فرآیند الگوریتم ژنتیک

در الگوریتم ژنتیک (GA)، هر راه حل موثر برای مسئله یک " کروموزوم " مربوط به هر فرد از جمعیت نامیده می شود. یک کروموزوم یک رشته کدگذاری شده با استفاده از روش کدگذاری ویژه است و هرواحد از رشته کدگذاری شده یک " ژن " نامیده می شود. با مقایسه مقادیر انطباقی، GA مزایا و معایب کروموزوم ها را از یکدیگر تفکیک می کند. کروموزوم دارای مقدار انطباقی بزرگتر، برجسته تر است.

در GA، تابع انطباق برای محاسبه مقدار انطباق کروموزوم متناظر استفاده می شود. انتخاب برای گزینش یک فرد براساس قوانین مشخص و تشکیل جمعیت والدین استفاده می شود. تقابل برای بخش تبادلی ژن های دو فرد برای تولید کروموزوم های فرزند آنها استفاده می شود؛ جهش برای تغییر چند ژن از کروموزوم های منتخب بکار می رود تا یک ژن جدید بدست آید.

گام های اصلی GA شامل گام های زیر است:

گام 1: برای مقداردهی اولیه یک جمعیت دارای N کروموزوم، ژن های هر کروموزوم را بصورت تصادفی گرفته و آنها را در محدوده تعریف مسئله حفظ کنید. تعداد نسل را با *Generation* نشان داده و برابر صفر در نظر بگیرید.

گام 2: برای ارزیابی هر کروموزوم با استفاده از تابع انطباق، مقدار انطباق هر کروموزوم را محاسبه کنید، بهترین کروموزومی که انطباق آن بزرگترین مقدار است ذخیره کرده و آن را بهترین کروموزوم بنامید.

گام 3: برای انتخاب با استفاده از روشی مانند چرخ رولت، جمعیت دارای N کروموزوم انتخابی را تولید کنید.

گام 4: انجام تقابل براساس احتمال  $P_c$ . هر زوج کروموزوم انتخابی، چند ژن را برای تولید دو فرزند آنها تولید کرده و آنها را تعویض کنید. سایر کروموزوم ها در جمعیت حفظ می شوند.

گام 5. جهش براساس احتمال  $p_m$ . برخی کروموزوم های جدید بطور جداگانه از طریق تغییر چند ژن از کروموزوم انتخابی متناظر تولید می شود. سایر کروموزوم های انتخاب نشده در جمعیت حفظ می شود.

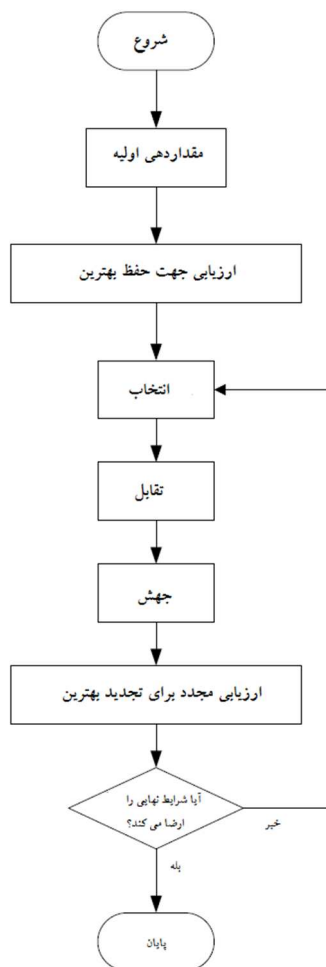
گام 6. هر کروموزوم را با استفاده از تابع انطباق مجدداً محاسبه نمایید. اگر بزرگترین مقدار انطباق در جمعیت جدید بهتر از بهترین آن باشد، جایگزین بهترین مقدار کنید.

گام 7. مقدار مذکور را  $++ Generation$  در نظر بگیرید. اگر  $Generation$  از حداکثر تولید مشخص شده تجاوز کند یا بهترین مقدار الزام خطای مشخص شده را نتیجه دهد، الگوریتم را پایان دهید. در غیر این صورت به گام 3 بروید.

### 3-2 طرح نمایش

در TCP، هر کروموزوم یک توالی مورد آزمایشی است. آنچه مهم است، ترتیب موارد آزمایشی بجای ترکیب خاص آنهاست. بنابراین، ما یک عدد طبیعی منحصر بفرد را برای هر مورد آزمایشی اختصاص می دهیم و سپس هر فرد می تواند به صورت یک توالی منظم از اعداد کدگذاری شود. مقیاس جمعیت اولیه، بر ظرفیت جستجو و بازدهی عملیاتی GA تأثیر می گذارد در نتیجه معمولاً بین 20 تا 150 متغیر است. بعنوان یک مثال، نمایش یک فرد  $T_1-T_2-T_3-...-T_m$  مربوط به دنباله موارد آزمایشی  $\Phi = \{T_1, T_2, \dots, T_m\}$  بصورت شکل 4 نشان داده می شود.





شکل 3. فلوچارت GA

T1	T2	T3	...	Tm
----	----	----	-----	----

شکل 4. طرح نمایش

### 3-3 طراحی تابع انطباق

تابع انطباق برای محاسبه مقدار انطباق هر کروموزوم و هدایت مسیر جستجوی GA بکار می رود. بنابراین، بخش کلیدی از اجرای الگوریتم ژنتیک است. بطور کلی، مقدار انطباق بین صفر و یک است. افراد دارای مقدار بزرگتر بهتر بوده و احتمال بیشتری برای تکامل آنها به نسل بعد وجود دارد. بطور کلی، برای یک مورد آزمایشی، هرچه تعداد نقاط آزمایش برای پوشش بیشتر شود، احتمال یافتن نقایص بیشتر می شود. بنابراین، با در نظرگیری افزایش پوشش موارد آزمایشی برای نقاط آزمایش به عنوان هدف تکاملی، APTC برای استفاده به عنوان یک

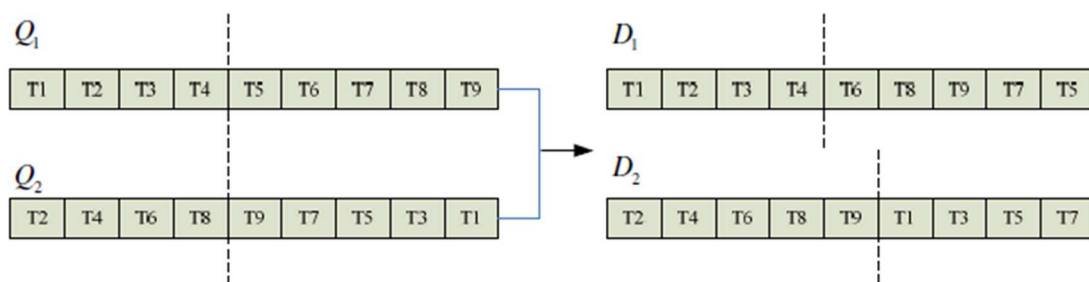
تابع انطباق گزینه خوبی است. علاوه بر این، اگر هدف، افزایش سطح اهمیت نقاط آزمایش پوشش داده شده در هزینه آزمایش واحد باشد، در این صورت APTC-C برای ارزیابی انطباق مناسب تر است.

### 3-4 طرح انتخاب

از انتخاب چرخ رولت استفاده کنید. بطور مثال، برای یک جمعیت دارای  $K$  نفر،  $fitness_i$  برای نشان دادن انطباق  $i$ امین نفر بکار می رود، انتخاب چرخ رولت شامل 5 گام است. ابتدا، درصد انطباق  $fitness_i / \sum fitness_i$  را محاسبه نمایید که قابلیت هر فرد را برای تولید زایش فرزند نشان می دهد. دوم، افراد را به ترتیب نزولی درصد انطباق آنها مرتب کنید. سوم، برای هر فرد، کلیه درصدهای انطباق افرادی را که در جلوی آن هستند، جمع کنید. سپس، اولین فردی که انطباق کل آن بیشتر از عدد تصادفی  $r_s \in [0,1]$  است، انتخاب کنید. در نهایت، گام های فوق را بصورت چرخه ای درآورید تا افراد کافی به دنیا آیند. همانطور که می توان مشاهده کرد، افراد دارای تناسب بیشتر، به احتمال بیشتری برای تولید نسل بعدی انتخاب خواهند شد که سازگار با اصول تکامل است.

### 3-5 طرح تقابل

فرض کنید که  $Q_1$  و  $Q_2$  افراد انتخاب شده برای انجام تقابل باشند،  $D_1$  و  $D_2$  افراد پس از تقابل باشند. احتمال تقابل را  $p_c$  در نظر گرفته و یک عدد تصادفی  $r_c \in [0,1]$  را تعیین کنید، زمانی که  $r_c$  کمتر از  $p_c$  بصورتی که در زیر شرح داده می شود باشد،  $D_1$  و  $D_2$  را بدست آورید. برای یک نقطه تقابل تصادفی  $D_1, k (1 \leq k \leq m)$  متشکل از دو بخش است: اولین بخش،  $K$  مورد آزمایشی اولیه  $Q_1$  است و دومین بخش حاصل از  $Q_2$  به جز  $K$  مورد آزمایشی اولیه  $Q_1$  می باشد. بطور مشابه،  $D_2$  متشکل از دو بخش است: اولین بخش،  $(m-k)$  مورد آزمایشی اولیه  $Q_2$  است و دومین بخش حاصل از  $Q_1$  به جز  $(m-k)$  مورد آزمایشی اولیه  $Q_2$  می باشد. شکل 5 یک مثال را نشان می دهد. مقدار  $p_c$  بطور کلی بین 0.4 و 0.99 است.



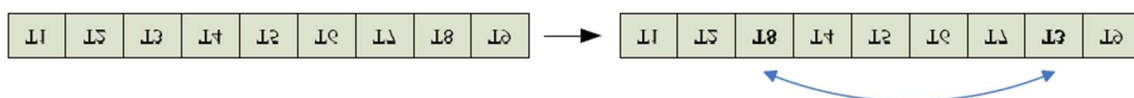
شکل 5. طرح تقابل

### 3-6 طرح جهش

جهش، برخی ژن های کروموزوم انتخابی را برای تولید یک فرد جدید تغییر می دهد. احتمال جهش را  $p_m$  در نظر گرفته و یک عدد تصادفی  $r_m \in [0,1]$  تعیین کنید و زمانی که کمتر از  $p_m$  بصورتی که در زیر شرح داده می شود باشد، فرد جدید را بدست آورید: براساس اهمیت نقاط آزمایش پوشش داده شده با موارد آزمایشی، دو مورد آزمایشی از فرد والد را با استفاده از روشی همانند انتخاب چرخ رولت انتخاب نمایید. سپس، جایگاه های دو مورد آزمایشی انتخابی را برای شکل گیری یک فرد جدید تعویض کنید. شکل 6 یک مثال را نشان می دهد. مقدار  $p_m$  بطور کلی بین 0.001 و 0.1 است.

موارد آزمایشی را برای انجام جهش با استفاده از روش انتخاب چرخ رولت براساس اهمیت نقاط

آزمایش تحت پوشش آنها انتخاب کنید



شکل 6. طرح جهش

### 4. آزمایش شبیه سازی

ما از برنامه طبقه بندی مثلث برای نمایش اثر روش پیشنهادی بهره می گیریم. برنامه طبقه بندی مثلث، مقادیر و رابطه سه متغیر اولیه  $(X,Y,Z)$  را تحلیل می کند تا نوع مثلثی که بوسیله آنها ترسیم می شود، تعیین کند. بطور کلی 7 نقطه آزمایش برنامه طبقه بندی مثلث همانند آنچه در جدول 2 نشان داده شده وجود دارد. ما کل

6 برنامه آزمایش جعبه سیاه را طراحی می کنیم. رابطه بین موارد آزمایشی و نقاط آزمایش در جدول 3 نشان داده شده است.

جدول 2. نقاط آزمایش برنامه طبقه بندی مثلث

شماره	عنوان	توضیح	اهمیت
1	IsNumError	قضاوت کنید آیا $(x,y,z)$ خارج از محدوده قابل قبول است یا خیر یا بررسی کنید آیا هر متغیر کمتر از صفر است یا بیشتر از مقدار حداکثر.	1
2	IsTriangleError	قضاوت کنید آیا $(x,y,z)$ می تواند یک مثلث را تشکیل دهد یا خیر یا بررسی کنید آیا جمع هر دو ضلع بیشتر از ضلع دیگر است یا خیر.	1
3	IsTriangle	قضاوت کنید آیا $(x,y,z)$ برای اضلاع مثلث مناسب است یا خیر یا بررسی کنید آیا هر متغیر در محدوده قابل قبول بوده و جمع هر دو ضلع بیشتر از هر ضلع دیگر است یا خیر.	1
4	IsScalTriangle	قضاوت کنید آیا مثلث مختلف الاضلاع است یا خیر مشروط بر اینکه $(x,y,z)$ بتواند مثلث را تشکیل دهد یا بررسی کنید آیا $x \neq y \neq z$ برقرار است یا خیر	2
5	IsRightTriangle	قضاوت کنید آیا مثلث متعلق به مثلث قائم است یا خیر مشروط بر اینکه $(x,y,z)$ بتواند مثلث را تشکیل دهد یا بررسی کنید آیا $x^2+y^2=z^2$ برقرار است یا خیر	2
6	IsIsosTriangle	قضاوت کنید آیا مثلث متعلق به مثلث متساوی الساقین است یا خیر مشروط بر اینکه $(x,y,z)$ بتواند مثلث را تشکیل دهد یا بررسی کنید آیا $x=y$ ، $y=z$ or $x=z$ برقرار است یا خیر	2
7	IsEquiTriangle	قضاوت کنید آیا مثلث متعلق به مثلث متساوی الاضلاع است یا خیر مشروط بر اینکه $(x,y,z)$ بتواند مثلث را تشکیل دهد یا بررسی کنید آیا $x=y=z$ برقرار است یا خیر	3

جدول 3. رابطه بین موارد آزمایشی و نقاط آزمایش برنامه طبقه بندی مثلث

No.	نام مورد آزمایشی	هزینه مورد آزمایشی	1	2	3	4	5	6	7
A	NumErrorTest	1	X						
B	TriangleErrorTest	1		X					
C	ScalTriangleTest	2			X	X			
D	RightTriangleTest	2			X		X		
E	IsosTriangleTest	2			X			X	
F	EquiTriangleTest	3						X	X

مقادیر پارامترهای GA را همانطور که در جدول 4 نشان داده در نظر بگیرید. بهترین سه نتیجه با روش پیشنهادی با استفاده از APTC و APTC-C بطور جداگانه بدست می آید که در جدول 5 و 6 نشان داده شده است.

جدول 4. مقادیر پارامترهای GA در برنامه طبقه بندی مثلث

No.	نام	مقدار
$P_c$	احتمال تقابل	0.85
$P_m$	احتمال جهش	0.05
$G_{max}$	حداکثر تعداد تکرارها	50
$N$	مقیاس جمعیت	150

جدول 5. نتایج جزئی آلتگرام APTC

No.	توالی موارد آزمایشی	APTC
1	C-F-D-E-A-B/D-C-E-F-A-B/F-E-D-C-A-B	0.4881
2	A-C-D-F-E-B/C-A-D-E-F-B/F-D-C-A-E-B	0.4643
3	A-C-D-B-F-E/C-A-D-B-F-E/D-B-F-A-C-E	0.4405

جدول 6. نتایج جزئی آلتگرام APTC-C

No.	توالی موارد آزمایشی	APTC_C
1	D-C-E-A-B-F/D-C-E-B-A-F	0.8030
2	C-D-E-A-B-F/C-D-E-B-A-F	0.7955
3	D-C-A-E-B-F/D-C-B-E-A-F	0.7879

## 5. نتیجه گیری

کاربرد الگوریتم های ژنتیک در اولویت بندی مورد آزمایشی، می تواند بطور موثری اغماض در ترتیب موارد آزمایشی اجرا شده را کاهش داده و در نتیجه بازدهی آزمایش نرم افزار را افزایش دهد.

این مقاله یک ارزیابی جدید اولویت بندی مورد آزمایشی APTC و اصلاحیه آن APTC-C را برای آزمایش کارکردی پیشنهاد می کند. از آنجا که این ارزیابی ها روی پوشش نقاط آزمایش تمرکز دارند، برای آزمایش جعبه سیاه مناسب تر هستند. علاوه بر این، این مقاله یک روش اولویت بندی مورد آزمایشی با استفاده از الگوریتم های ژنتیک را ارائه می کند که APTC یا APTC-C را به عنوان تابع انطباق انتخاب کرده است. طرح های نمایش، انتخاب، تقابل و جهش در GA به آزمایش جعبه سیاه کمک می کند. ما گام های مشخص این روش را ارائه داده و آن را با داده های آزمایشگاهی صحت آزمایی کردیم.

نتایج آزمایشگاهی نشان می دهند که روش پیشنهادی می تواند نتایج مورد انتظار را بدست دهد. این روش یک رویکرد فنی موثر نسبت به مسئله اولویت بندی مورد آزمایشی ارائه می دهد. در آینده، ما تحقیق بیشتری در مورد بحث نقاط آزمایش و کاربردهای GA در تولید خودکار موارد آزمایشی جعبه سیاه انجام خواهیم داد.