

## 基于Web应用前端行为模型的测试用例生成

刘子文, 虞莉娟, 苏义鑫, 赵耀, 石柱

### 引用本文

刘子文, 虞莉娟, 苏义鑫, 赵耀, 石柱 [基于Web应用前端行为模型的测试用例生成](#)[J]. 计算机科学, 2023, 50(7): 18-26.

LIU Ziwen, YU Lijuan, SU Yixing, ZHAO Yao, SHI Zhu. [Test Case Generation Based on Web Application Front-end Behavior Model](#) [J]. Computer Science, 2023, 50(7): 18-26.

---

### 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

#### Similar articles recommended (Please use Firefox or IE to view the article)

#### [基于机器视觉的Web应用页面元素识别及可视化脚本生成](#)

Web Application Page Element Recognition and Visual Script Generation Based on Machine Vision  
计算机科学, 2022, 49(11): 65-75. <https://doi.org/10.11896/jsjcx.220200122>

#### [基于人体关节点的低头异常行人检测](#)

Detection of Head-bowing Abnormal Pedestrians Based on Human Joint Points  
计算机科学, 2021, 48(5): 163-169. <https://doi.org/10.11896/jsjcx.200800214>

#### [Web应用测试用例生成研究](#)

Study of Test Case Generation for Web Applications  
计算机科学, 2010, 37(6): 159-163.

#### [Web应用测试的通用编解码器研究](#)

Research of Generic Codec for Web Application Testing  
计算机科学, 2013, 40(8): 157-160.

#### [一种基于极大团的关键时间段挖掘方法](#)

Mining Method of Key Time Interval Based on Maximum Clique  
计算机科学, 2012, 39(6): 166-169.

# 基于 Web 应用前端行为模型的测试用例生成

刘子文 虞莉娟 苏义鑫 赵耀 石柱

武汉理工大学自动化学院 武汉 430070

(373629126@qq.com)

**摘要** 基于 Web 应用前端模型的测试用例生成是 Web 应用测试的重要过程,但现有的大多数针对 Web 应用的模型都只关注 Web 页面及其事件,忽略了事件触发条件和后续动作。为了更准确地描述现代 Web 应用程序的动态行为,文中定义了一种新的 Web 应用前端行为模型(Front-end Behavior Model,FBM)。由于模型中迁移的触发条件可能存在内部变量,即迁移之间存在依赖关系,使得生成的测试用例不能按照输入序列执行,从而影响测试效果。为此,提出了一种优化分组遗传算法来实现可执行迁移路径 FTP 的自动生成,该算法考虑 FTP 生成问题的特点,对染色体初始化和适应度函数做出合理的设计,加入修补算子调整个体长度生成了满足迁移覆盖的 FTP,并且通过引入自适应的遗传算子和模拟退火接收机制来减少迭代次数,从而加快求解速度。实验结果表明,该算法能够在更高的求解效率的基础上有效保证迁移路径的可行性以及覆盖率。

**关键词:** Web 应用测试;前端行为模型;可执行测试用例生成;分组遗传算法

**中图分类号** TP311.53

## Test Case Generation Based on Web Application Front-end Behavior Model

LIU Ziwen, YU Lijuan, SU Yixing, ZHAO Yao and SHI Zhu

School of Automation, Wuhan University of Technology, Wuhan 430070, China

**Abstract** Test case generation based on Web application front-end model is an important process of Web application testing, but most existing models for Web applications only focus on Web pages and their events, ignoring event triggering conditions and subsequent actions. Therefore, in order to describe the dynamic behavior of modern Web applications more accurately, this paper defines a new Web application front-end behavior model(FBM). Because there may be internal variables in the triggering conditions of transition in the model, that is, there are dependencies between transitions, which will make the generated test cases cannot be executed according to the input sequence, thus affecting the test results. Therefore, an optimized grouping genetic algorithm is proposed to automatically generate the feasible transition path(FTP). Considering the characteristics of FTP generation problem, the algorithm makes a reasonable design of chromosome initialization and fitness function, and adds a repair operator to adjust the individual length to generate FTP which satisfies the migration coverage. This paper also introduces an adaptive genetic operator and simulated annealing receiving mechanism to reduce the number of iterations, thus improving the solution speed. Experimental results show that the algorithm can effectively guarantee the feasibility and coverage of transition path on the basis of higher solution efficiency.

**Keywords** Web application testing, Front-end behavior model, Feasible test case generation, Grouping genetic algorithm

## 1 引言

随着 Web 应用的迅猛增长,其安全漏洞的数量和复杂程度也在不断增加。Web 应用程序的安全性测试已成为当前软件工程领域的一个重要问题,其中测试用例生成是检测及验证 Web 应用是否安全可靠的关键一环。基于模型的测试方法(Model-based Testing, MBT)是从模型中自动生成测试用例的技术,是目前检验 Web 应用质量和可靠性最有效的方法之一<sup>[1-2]</sup>。

目前,研究人员已经提出了多种模型来表示现代 Web 应用,但大多只考虑了 Web 应用事件驱动和异步的特性<sup>[3]</sup>,忽略了事件触发可能引起的参数或 DOM 元素的变化。因此,这些模型并不能精确表示 Web 应用行为,而且从这些模型中生成的测试用例,也因为事件触发的条件和动作以及后续操作条件的未知,导致测试用例不可行。

由于 Web 应用动态行为主要由页面以及事件构成,因此统一建模语言 UML 中的状态图经常被用来表示 Web 应用。从状态图模型中生成测试用例的方法有很多,其中最常用的

到稿日期:2022-09-16 返修日期:2023-03-24

基金项目:国家自然科学基金(62176193)

This work was supported by the National Natural Science Foundation of China(62176193).

通信作者:虞莉娟(ylj@whut.edu.cn).

是遗传算法(Genetic Algorithm, GA)<sup>[4-5]</sup>。在 MBT 领域,为了符合迁移覆盖准则<sup>[6]</sup>,许多研究使用 GA 来生成基于迁移覆盖的迁移路径(TPs)。一条迁移路径(Transition Path, TP)是从状态图的初始状态开始的连续迁移<sup>[7]</sup>。若 TP 中的每一个迁移都可以按其顺序执行,那么 TP 就被视为一条可行的迁移路径(Feasible Transition Path, FTP)。由于状态图中的迁移可能会包含应用程序中的变量,变量会影响到后续迁移是否可行,这样的一组迁移被称为依赖迁移组(Dependent Transition Group, DTG)。因为 GA 在进行交叉操作时交叉点位置的确认是随机的,如果交叉点被确定在 DTG 之间,在交叉操作完成后会导致迁移路径的不可执行。例如迁移  $t_1$  中的动作是  $i++$ ,迁移  $t_2$  中的保护条件是  $i \geq 4$ ,如果迁移  $t_1$  的执行次数不足 4 次,就无法触发迁移  $t_2$  的执行。这就使得迁移路径不能按照输入测试用例的序列来执行,导致迁移无法被完全覆盖。

因此,本文基于 UML 状态图为现代 Web 应用定义一种新的前端行为模型(FBM),来解决 Web 应用的模型表示问题。并且针对模型中存在依赖迁移可能导致生成的测试用例不可执行的问题,提出了一种优化分组遗传算法(Optimized Grouping Genetic Algorithm, OGGA),从 FBM 模型中生成满足迁移覆盖的可执行测试用例。本文的主要贡献如下:

(1)提供了一个 FBM 来表示现代 Web 应用的动态行为,其中不仅考虑了 Web 页面和事件,还考虑了事件触发条件和后续动作。

(2)提出了一个 OGGA,它利用分组遗传算法 GGA 通过分组进行交叉操作的特性,应用优化技术得到最优解,从 FBM 中生成可执行且满足迁移覆盖的 FTP。

(3)通过 5 个系统模型对 OGGA 进行迁移覆盖率和算法性能的对比实验,结果表明了所提算法的实用性和有效性。

## 2 背景

### 2.1 现代 Web 应用的基本概念

对于现代 Web 应用,其动态行为主要由页面和事件两个主要成分表征。其中页面可以用 DOM 树来表示,DOM 树中的所有节点均可以通过执行前端 JS 代码/后端代码动态修改,从而实现 Web 页面的动态更新。因为 Xpath 可以通过路径表达式来定位 DOM 树中的元素、文本和属性<sup>[8]</sup>,具有很强的搜索和导航能力,所以一般采用 Xpath 来获取页面元素。

对于 Web 应用事件,Web 应用属于事件驱动型软件,事件的触发导致前端或后端代码的执行,从而使 Web 应用从当前页面迁移到新页面。在现代 Web 应用中,前端会先利用一组 JS 函数(即事件处理函数)对用户操作进行预处理,完成前端的业务逻辑功能<sup>[9]</sup>,而函数中对应的保护条件决定了之后的跳转页面以及参数或 DOM 元素的变化。因此,Web 页面的动态更改不仅与事件有关,而且还与事件的触发条件以及参数的变化有关。

### 2.2 基于模型的测试

基于模型的测试的思想是通过执行从模型中产生的测试用例来确定软件基于模型规范是否正常工作<sup>[10]</sup>。在基于

模型的测试中,测试模型一般使用图表来构建,例如 UML 状态图。由于图是基于形式符号的,它可以清楚地表达测试目标的行为<sup>[11]</sup>,因此,在基于模型的测试中,可以通过使用正式符号构建测试模型来生成测试用例。

迁移覆盖测试是一种用于检查状态图中所有迁移是否全部执行的测试方法<sup>[12]</sup>。然而,如第 1 节中所介绍,DTG 的存在导致有些 TP 不是 FTP,这可能会造成测试失败,因此必须要创建一组考虑迁移之间依赖关系的 FTP。

### 2.3 分组遗传算法

分组遗传算法(Grouping Genetic Algorithm, GGA)是由 Falkenauer<sup>[13-14]</sup>提出的一种改进的遗传算法。与传统 GA 不同,GGA 的染色体和遗传算子不是面向个体而是面向群体,染色体由基因分组的形式表示,如下例所示。

染色体( $\{2\ 4\}\{1\ 3\}\{6\ 7\}\{5\ 8\}$ )

与 GA 的交叉和变异操作不同的是,GGA 中的交叉操作是在组和组之间随机放置交叉点,变异操作则是创建新组或删除现有组。基于这种特性,GGA 已经被广泛应用于解决有分组元素的问题。对于有“计数器问题”的状态图模型,当使用传统 GA 来生成 FTP 时,由于交叉点可以设置在执行序列之间,因此迁移的执行序列很难在进化过程中保持。但 GGA 的交叉点设置在组间,因此 GGA 可以通过分组来保护执行序列。

## 3 现代 Web 前端行为模型定义及表示

页面和事件是表征 Web 应用动态行为的两个重要组成部分,事件触发不同的执行条件会导致出现不同的 Web 页面,并且引用户事件回调或服务消息对参数的不同操作。为准确、全面地反映 Web 应用的动态行为,本节定义了一种基于 UML 状态图的 Web 应用前端行为模型 FBM,将 Web 应用动态行为与模型的状态和迁移对应起来。

### 3.1 前端行为模型

现代 Web 应用中,事件执行条件与 Web 页面的关系以及事件回调引发的后续操作在表征其动态行为时必不可少。为此,本节针对现代 Web 应用,定义了一种新的前端行为模型 FBM,其定义如下。

**定义 1(前端行为模型, FBM)** FBM 为一个四元组  $\langle S, I, O, T \rangle$ ,其中  $S$  为非空的有限状态集合,表示页面的 URL 和对应的 DOM;  $T$  为非空的有限状态迁移集合,表示状态之间的迁移;  $I$  为非空的有限输入集合  $t \in T$ ;  $O$  为非空的有限输出集合。具体来说,迁移可以表示为一个五元组  $\langle src, event, cond, act, targ \rangle$ ,其中  $src(t)$  和  $targ(t)$  分别表示迁移  $t$  的源状态和目标状态;  $event(t)$  表示在当前源状态下用户触发的事件;  $cond(t)$  表示在相应事件处理函数中的执行条件;  $act(t)$  代表用户事件回调或服务消息对参数或 DOM 元素的后续操作。

### 3.2 FBM 状态及迁移表示

对于现代 Web 应用程序,FBM 模型中的状态可以用页面 URL 和 DOM 来表示。然而,由于 DOM 的细微更改均可能导致状态的无限扩展,若将 DOM 直接映射为 FBM 状态,

则不仅可能导致状态空间爆炸,而且会使状态之间的关系变得极其复杂。因此,本节考虑在不影响 FBM 模型语义的前提下,对页面 DOM 进行适当的抽象。

目前,针对页面 DOM 的抽象大多集中在 DOM 的结构或内容上,即利用提取到的 DOM 的元素节点或内容节点来表示 DOM。然而,对于具有相同结构的两个 DOM 而言,它们很可能表示不同的功能,不能直接视为一个。另一方面,对 DOM 内容的抽象虽然可以区分两个具有不同页面内容的 DOM,但它们可能将同类页面标识为不同页面。因此,现有的 DOM 抽象方法均不适用。

为此,本文提出了一种基于 DOM 结构和属性的 Web 页面抽象方法。已知在 Web 页面的 DOM 树中,属性节点为页面元素定义属性或定制元素的样式,而元素可通过属性绑定事件及处理函数,实现前端功能。因此,本节基于 DOM 结构和属性定义了抽象 DOM,以表示 FBM 状态,定义如下。

**定义 2(抽象 DOM)** 抽象 DOM 由对应于页面 DOM 叶子节点的二元组集合  $\{\langle ele_0, (attr_{00}, \dots, attr_{0j}, \dots, attr_{0n}) \rangle, \dots, \langle ele_i, (attr_{i0}, \dots, attr_{ij}, \dots, attr_{in}) \rangle, \dots, \langle ele_n, (attr_{n0}, \dots, attr_{nj}, \dots, attr_{nm}) \rangle\}$  构成,其中  $ele_i$  表示页面 DOM 的第  $i$  个叶子节点元素,  $(attr_{i0}, \dots, attr_{ij}, \dots, attr_{in})$  为该元素对应的属性集合。

在 Web 应用程序中,某些特定事件只能通过一系列用户操作来触发。例如在教育综合管理系统 (EduFei) 的请假申请模块中,若学生需要请假,则需要选择请假类别,然后填写请假的时间和原因,最后单击提交按钮完成请假申请。在此过程中,单个操作不能触发事件处理函数的执行以及页面状态的更改。因此,这些导致前端或后端代码执行的用户操作序列应组合为 FBM 迁移上的一个事件,而将事件触发的关联条件表示为迁移条件,将对参数的后续操作表示为迁移的响应操作。

### 3.3 示例

本节以 EduFei 管理系统为例,对其 FBM 模型进行说明。EduFei 包含 17 个功能模块,允许学生执行与学校相关的任务,如课程公选、课程表、成绩查询等,其中课程公选模块的部分 FBM 如图 1 所示。图中  $S_0$  表示教育综合管理系统登录页面,学生可在此页面登入系统,若其输入的用户名或密码为空或错误 ( $T_0$  &  $T_1$ ),则学生仍停留在登录页 ( $S_0$ ),否则 ( $T_2$ ) 进入主界面 ( $S_1$ ); 此时学生可以执行多种操作 (如查看课程表、提交科研成果等)。以点击课程公选按钮 ( $T_3$ ) 为例,学生进入课程公选页面 ( $S_2$ ),在该页面有 5 门课程,学生需要至少选择 3 门课程,已选课程数  $count$  起始默认为 0。每门课程之后都绑定了一个选择课程按钮,学生若点击按钮 ( $T_4$ ),则已选课程数为  $count+1$ 。当点击完成按钮时,需要检查  $count$  是否大于或等于 3,如果  $count$  小于 3,则 ( $T_5$ ) 进入选课未完成状态 ( $S_3$ ),点击继续选课按钮 ( $T_6$ ) 会将  $count$  重新置为零,并返回选课页面 ( $S_2$ ) 继续添加; 如果  $count$  大于或等于 3,则 ( $T_7$ ) 进入完成状态 ( $S_4$ ),选课成功,点击返回按钮 ( $T_8$ ) 返回选课页面 ( $S_2$ )。为了便于理解,将状态用相应的网页标记,页面之间迁移的详细信息如表 1 所列。

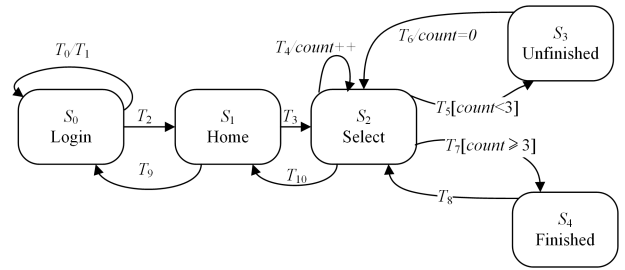


图 1 EduFei 课程公选模块的 FBM

Fig. 1 FBM of EduFei course elective module

表 1 EduFei 课程公选模块的行为模型 FBM 迁移信息

Table 1 Behavior model FBM transition information of EduFei course elective module

Trans	Src	event	cond	act	Targ
$T_0$	$S_0$	click, Xpath://input [ @ value = 'Login' ] (username, password)	username. value = = t'    password. value = = t'	alert ('invalid params')	$S_0$
$T_1$	$S_0$	click, Xpath://input [ @ value = 'Login' ] (username, password)	username. value    password. value = = false	alert ('username or password error')	$S_0$
$T_2$	$S_0$	click, Xpath://input [ @ value = 'Login' ] (username, password)	username. value & & password. value = = true	document. login. value = 1	$S_1$
$T_3$	$S_1$	click, link = Check	—	—	$S_2$
$T_4$	$S_2$	click, Xpath://input [ @ value = 'Select' ]	—	count = count + 1	$S_2$
$T_5$	$S_2$	click, Xpath://input [ @ value = 'Submit' ]	count < 3	alert ('Insufficient elective courses')	$S_3$
$T_6$	$S_3$	click, Xpath://input [ @ value = 'Continue' ]	—	count = 0	$S_2$
$T_7$	$S_2$	click, Xpath://input [ @ value = 'Submit' ]	count > = 3	isFinished = 1, document. SelectCourse. submit ()	$S_4$
$T_8$	$S_4$	click, Xpath://input [ @ value = 'Accomplish' ]	—	—	$S_2$
$T_9$	$S_1$	click, Xpath://input [ @ value = 'LogOff' ]	—	—	$S_0$
$T_{10}$	$S_2$	click, Xpath://input [ @ value = 'BackHome' ]	—	—	$S_1$

## 4 用于生成 FTP 的优化分组遗传算法

本节描述了一种使用 OGGA 生成具有依赖迁移对的 Web 前端行为模型 FBM 中满足迁移覆盖的 FTP 的技术。使用传统的 GA 很难在具有 DTG 的模型中生成满足迁移覆盖的 FTP,图 2 给出了传统 GA 解决具有 DTG 的问题时,其交叉操作破坏基因组的例子,其中父染色体 A 和 B 可用于从图 1 模型中生成 FTP。由于存在内部变量  $i$ ,且迁移 7 的执行条件为  $[count \geq 3]$ ,所以迁移 4 必须执行 3 次才能执行迁移 7。在本例中,父染色体 A 中拥有可行的迁移序列,可以执行迁移 7,但当其与父染色体 B 发生交叉操作后,可行的迁移序列被破坏,其被破坏的概率为  $3/8$ 。为了避免可行的迁移序列被破坏,交叉点必须要在迁移序列之外。由于 GGA 具有按组来执行交叉操作的特性,因此在需要这样特定的迁移序列时,可以将其作为一组来进行保护。接下来将详细介绍对 GGA 的改进方法,使其适用于 FTP 生成问题。

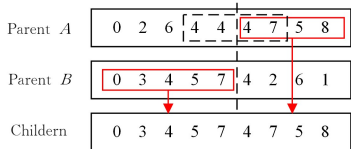


图 2 交叉操作对可行的迁移序列的破坏过程

Fig. 2 Destruction process of feasible transition sequence by crossover operation

### 4.1 染色体表示

本文将迁移路径进行分组编码,染色体表示为若干迁移组成的集合,图 3 给出了本文中使用的染色体的例子,染色体中有若干个组,每个组中有若干个基因。在发生交叉时将交叉点放置在组与组之间,以此来保护可行迁移序列。然而,分组不能保护可行迁移序列不受其他遗传操作的影响,例如,变异操作会随机选择一个基因并改变它,如果变异算子选择了可行迁移序列中的基因,那么可行迁移序列就可能被破坏。因此,可以选择将某些基因设置为“固定值”,变异算子不能改变被设置为固定值的基因,固定基因将存在于群体中以引领群体的进化方向,直到群体灭绝。当染色体初始化时,将第一个创建的固定基因的值指定为“组 ID”。

如图 3 所示,第一组和第二组分别将 3 和 6 指定为固定基因,组 ID 分别为 3 和 6。当第三组初始化时,迁移 7 被指定为固定基因,组 ID 的值被赋为 7,由于迁移 7 和迁移 4 是依赖迁移对,所以迁移 4 被添加并指定为固定基因。

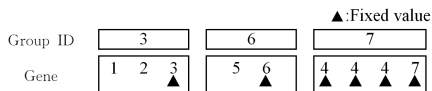


图 3 染色体表示

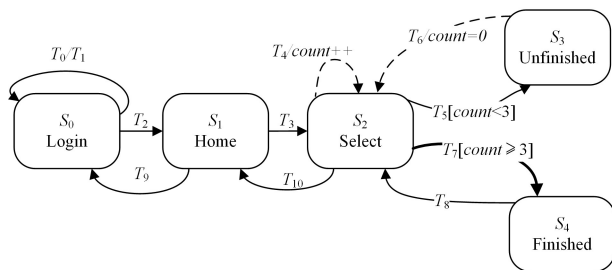
Fig. 3 Chromosome representation

### 4.2 染色体初始化

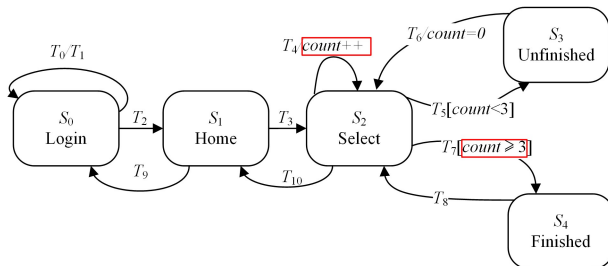
群体初始化从随机基因的插入开始,最先插入的基因被指定为固定基因,并且基于该基因(迁移)形成一个组。如果固定基因的转换条件中有一个内部变量,则它被认为是一个受影响的迁移,记为  $t'$ 。在 FBM 模型中搜寻另一个迁移,其

动作能够符合  $t'$  保护条件并对  $t'$  产生影响,记为  $t$ 。搜索到  $t$  之后,分析  $t$  中包含的动作和为了通过  $t'$  所需要执行  $t$  的次数,然后在染色体中添加所需次数的  $t$ ,并将其指定为固定基因。如果一个固定基因没有依赖迁移(没有转换条件,或条件中没有内部变量),则相应的过程中就不再添加固定基因。

搜寻  $t$  的过程如图 4 所示,迁移  $t_7$  为受影响的迁移,迁移  $t_4, t_6, t_8$  属于第一级迁移集合  $t_{n1}$ ,然后在  $t_{n1}$  中找到迁移  $t_4$ ,其动作满足迁移  $t_7$  的保护条件。如果在  $t_{n1}$  中没有搜索到  $t$ ,则深入到连接下一级的迁移集合  $t_{n2}$  中继续寻找,搜寻过程不断深入直到找到  $t$  为止。



(a) 从第一级迁移集合  $t_{n1}$  中搜索  $t$



(b) 在  $t_{n1}$  中寻找找到满足守卫条件的  $t$

图 4 寻找依赖迁移  $t$  的过程

Fig. 4 Process of finding dependent transition  $t$

通过此过程生成的组是由  $t'$  和与其匹配的  $t$  组成,具有相同  $t'$  的组也大概率具有相同的  $t$ 。随着遗传操作的重复,组将通过连接到  $t'$  和  $t$  的迁移来形成。另外,如果固定基因的条件中没有内部变量或者条件不存在时,即可认为不存在  $t$ 。在这种情况下,可以基于固定基因利用连接到固定基因的迁移来形成一个组。固定基因充当确定组的身份的元素,因此每个组被赋予一个与固定基因具有相同值的组 ID。如果在交叉操作时,染色体中的一个组与另一个组具有相同的组 ID,为了增加基因的多样性,则将该组删除。

迁移  $t$  添加完成后则组的初始化过程完成,并且组被插入到染色体中。之后将随机基因添加到每个组中,直到达到染色体的初始长度。染色体的初始长度设置为与模型中的迁移总数相同,因为在模型中实现迁移覆盖的最小 FTP 的长度与迁移总数相同。算法 1 给出了染色体初始化过程的概要伪代码。

#### 算法 1 种群初始化

输入:FBM(S,T)

输出:初始化后的染色体

- function InitChromosome()
- for  $i=1$  to  $\text{RandNum}(n_{tr}/4, n_{tr}/3)$  do \*  $n_{tr}$ : 迁移数量 \* /
- $G \leftarrow \text{CreatGroup}()$

```

4.   t' ← RandNum(0, ntr - 1)
5.   if 基因 t' 没有依赖迁移 t then
6.     G ← InsertGene(t')
7.   else
8.     n ← Count(t)
9.     G ← InsertGene(t', n * t)
10.  end if
11.  NewGroup ← G
12.  InsertGroup(NewGroup)
13. end for
14. while ngene < Lc do /* ngene: 染色体中的基因数目 */
15.   InsertGeneRandomm(RandNum(0, ntr - 1))
16. end while
17. end function

```

### 4.3 适应度算法

适应度是用来描述个体对环境的适应程度,它取决于遗传特性,适应度函数的设计包括建立一个问题的目标以及定义必须满足的元素来判断染色体是否接近最优解。本文使用迁移覆盖准则来衡量测试的充分性,适应度将随着覆盖率的增加而减小,其中覆盖率指染色体中所覆盖的 FBM 模型的迁移数量的比例。另外,覆盖率的计算还要考虑连通性和可行性,所有迁移必须连接起来才能成为一条 FTP。两个迁移是否连通可通过检查后一个迁移的源状态和前一个迁移的目标状态是否一致来判断,若一致,则将其视为连通。通过了连通性的检测后再对路径的可行性进行检查,表 1 记录了 FBM 模型迁移的动作中所分配变量的值,参照表来检查变量是否满足所执行迁移的守卫条件,满足则记录执行结果。检查过程从第一个基因开始,如果遇到不能通过的守卫条件,则执行中断。同时将所有通过的迁移记录为已覆盖的迁移,适应度将根据已覆盖的迁移数量来确定,如果模型中的所有迁移都被覆盖,则适应度为 0。

具体计算适应度的过程如算法 2 所示,在种群进化过程中,个体包含的迁移越多,迁移的关联性越好,则个体的适应度值越小。

#### 算法 2 适应度计算

输入: FBM(S, T), 初始化后的染色体

输出: 适应度值

```

1. function SetFitness()
2.   fitness ← 0
3.   for i = 1 to ntr - 1 do
4.     if genei.targetstate ≠ genei+1.sourcestate then
5.       break
6.     end if
7.     if CheckFeasibility(genei) = true then /* 检查路径可行性 */
8.       IncreaseCoveredTransition(genelist, genei)
9.     else
10.      break
11.    end if
12.  end for
13. for i = 1 to ntr do
14.  if CheckCoveredTransition(genelist, i) = false then /* 检查迁移覆盖范围 */

```

```

15.   FitnessIncrease()
16.   end if
17. end for
18. end function

```

### 4.4 遗传操作算子

种群中的染色体由遗传算子实现相互作用,遗传算子主要有选择、交叉、变异和复制 4 种,在解决 FTP 生成的问题中,选择和复制操作可以使用现有的遗传算法中的方法。但是由于交叉和变异算子的设计必须考虑染色体是以群体的形式形成,并且其中包括必须保持特定迁移序列的群体,因此需要对交叉和变异算子进行重新设计。

#### 4.4.1 交叉算子

交叉算子应主要考虑对特定迁移序列的组的保护,与现有的 GGA 不同的是,考虑到 TP 的特性,某些迁移的执行条件其他迁移执行达到一定次数,因此染色体中必须允许出现相同的基因。为了使基因尽可能多样化,以满足迁移覆盖,同时允许相同基因出现,交叉算子可以通过个体的交叉组合在染色体中组织具有不同组 ID 的组。交叉过程如图 5 所示。首先从两个父染色体中随机确定交叉点,然后将父染色体  $P_1$  的交叉区域的组注入父染色体  $P_2$  的交叉点中,如果  $P_2$  中原有的组与注入的组具有相同的组 ID,则将其删除。

为了提高算法的收敛性和种群多样性以及避免形成局部最优,本文引入了改进的动态自适应交叉算子<sup>[15]</sup>,个体之间发生交叉的概率如式(1)所示:

$$p_{cro} = \begin{cases} p_{cro1} - \frac{(p_{cro1} - p_{cro2})(f - f_{avg})}{e^{f_{max} - f_{avg}}}, & f \geq f_{avg} \\ p_{cro1}, & f < f_{avg} \end{cases} \quad (1)$$

其中,  $p_{cro1} > p_{cro2}$ ,  $f$  是交叉的两个个体中较高的适应度值,  $f_{max}$  是剩余个体中适应度的最大值,  $f_{avg}$  是种群的适应度值的平均值。交叉操作结束后,个体中的分组都将拥有不同的组号,同时允许相同基因的重现。

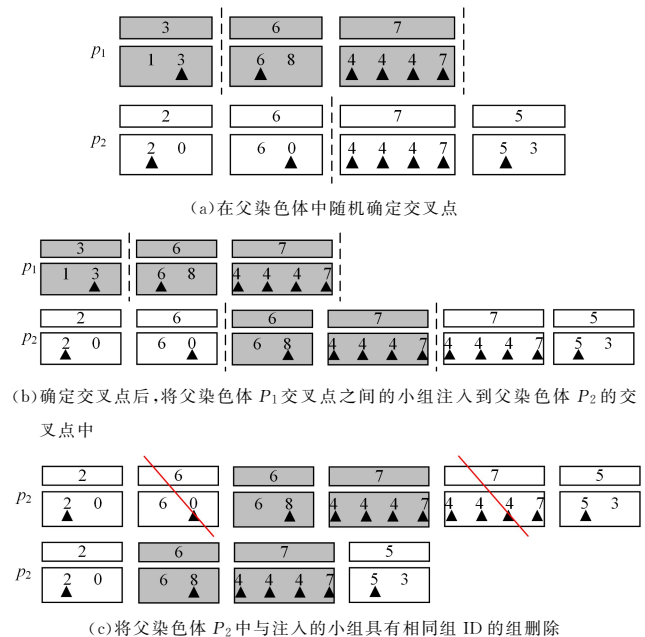
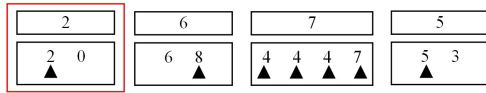


图 5 FTP 生成中的 GGA 交叉操作

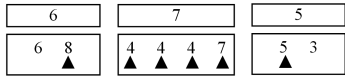
Fig. 5 GGA crossover operation in FTP generation

#### 4.4.2 变异算子

交叉操作可以形成新的子代个体,但如果父染色体在交叉区域内的组 ID 均相同,则不能产生新的个体。变异算子可以通过将个体的基因进行变异来产生新的个体,本文所设计的变异方法的操作过程如图 6 所示。



(a) 在染色体中随机选择一个组作为变异的目标



(b) 删除所选的分组,但是删除后染色体中的分组若少于 3 个,则不删除该组

图 6 FTP 生成中的 GGA 变异操作

Fig. 6 GGA mutation operation in FTP generation

变异操作对作为变异目标的染色体随机选择并删除其中的一个组,但是当染色体中的组的数量少于 3 个时,变异后的个体的交叉点将无法形成交叉区域且会降低个体的多样性,因此当染色体中的组数少于 3 组时应终止其变异操作。本文同样也引入了改进的动态自适应变异算子,个体发生变异的概率如式(2)所示:

$$p_{mut} = \begin{cases} p_{mut1} - \frac{(p_{mut1} - p_{mut2})(f - f_{avg})}{e^{f_{max} - f_{avg}}}, & f \geq f_{avg} \\ p_{mut1}, & f < f_{avg} \end{cases} \quad (2)$$

对于经过交叉和变异操作之后的个体,本文应用模拟退火算法来接收。当产生新个体  $P_{new}$  时,接收它的概率由 Metropolis 准则<sup>[16]</sup>确定,即只在  $P_{new}$  的适应度值更低时才将其接收,反之则以一定概率接收原个体,其概率计算式如式(3)所示:

$$p_{n_{k+1}} = \begin{cases} e^{-\frac{f_{k+1} - f_k}{t_{k+1}}}, & f_{k+1} < f_k \\ 1, & f_{k+1} \geq f_k \end{cases} \quad (3)$$

其中,  $f_k$  是原个体的适应度值;  $f_{k+1}$  是新个体的适应度值;  $t_{k+1}$  是算法中的控制系数。

#### 4.4.3 修补算子

在对遗传算子的设计和改进中可能会出现染色体长度过度增长的情况,为了避免这种情况的发生,可以采用限制解的长度的方法。但是由于满足模型迁移覆盖的 FTP 长度不确定,并且不同的模型的 FTP 的长度也不同,所以解的长度不能被限制为确定的数字。本文选取长度值  $L_r$  对染色体长度进行限制,并在搜寻最优解时适当调整解的长度。若交叉和变异后的染色体长度不同于  $L_r$ ,则需要添加或删除基因来进行修补。

具体步骤如算法 3 所示。如果染色体长度长于  $L_r$ ,则需要删除染色体中的基因,首先检查基因之间的连接,如果不关联则将其删除。若要删除的基因为固定值时,则不删除,继续搜索其他删除对象,直到当前染色体的长度等于  $L_r$ 。然而,由于组号不同的组可能覆盖了相同的区域,删除了除固定值以外的所有断开的基因之后,当前染色体的长度仍然大于  $L_r$ 。为了解决这个问题,当基因删除完成时,染色体的长度如果大于  $L_r$ ,则删除随机组来达到目标长度。当染色体长度短于  $L_r$  时,则需将基因添加到染色体中。首先寻找不关联的

基因,在其后添加与其关联的新基因。如果基因均关联,则在末尾添加关联基因,过程持续到染色体长度等于  $L_r$ 。

#### 算法 3 修补操作

输入: FBM(S, T), 染色体限制长度  $L_r$

输出: 修补后的染色体

```

1. function Repair()
2.   i, j, k ← 0
3.   while ngene > Lr do
4.     if genei.targetstate ≠ genei+1.sourcestate then
5.       DeleteGene(genei) /* 删除不关联基因 */
6.       if ngene ≤ Lr then
7.         break
8.       end if
9.     end if
10.    if i < ngene - 1 then
11.      i ← i + 1
12.    else
13.      j ← random(ngroup)
14.      DeleteGene(groupj) /* 如基因全部关联,则删除随机组 */
15.    end if
16.  end while
17.  while ngene < Lr do
18.    if genek.targetstate ≠ genek+1.sourcestate then
19.      InsertAfterLocation(genek, NewRelevantGene(genek)) /*
      在基因后添加新的关联基因 */
20.    else if (genek.targetstate = genek+1.sourcestate) and (k = ngene -
      1) then
21.      InsertEndLocation(NewRelevantGene(genek)) /* 尾端添
      加新的关联基因 */
22.    end if
23.    if k = ngene - 1 then
24.      k ← k + 1
25.    end if
26.  end while
27. end function

```

在添加了染色体修补后,染色体会根据确定的  $L_r$  进行搜索,如果在充分搜索后没有找到特定长度的最优解,则该长度被认为没有最优解,然后增加  $L_r$  继续搜索。为了计算增加  $L_r$  所需的遗传代数  $G_r$ ,需要分析模型的复杂度和染色体长度增加所需的遗传代数  $G_r$  之间的关系。本文用状态数(NSS)、迁移数(NT)和圈复杂度(CC)来表示模型的结构复杂度。CC 是基于 McCabe 的圈复杂度<sup>[17]</sup>,用来分析控制流图的结构复杂度。Cruz-Lemus<sup>[18]</sup>将其定义为  $|NSS - NT + 2|$ ,以数字形式表示状态图的复杂性,对于某一模型,  $G_r$  可近似表示为式(4):

$$G_r = E \times \frac{-3 \times NSS + 8 \times NT + 5 \times CC}{fitness} \quad (4)$$

其中,  $E$  是误差率,可以看出,  $E$  与  $L_r$  的增加速度有关,它决定了最优解的长度和得到最优解所需的时间。总体而言,这个增加染色体长度的过程为,若长度  $L_r$  的个体搜索次数达到  $G_r$  次后没有找到最优解,则个体长度  $L_r + 1$ ,并进入下一轮迭代周期。

以上是本文为适应 FTP 生成问题对 GGA 做出的改进,通过分组和设置固定值来保护特定迁移组,使其不受遗传

操作的影响。然后对染色体初始化和适应度函数做出合理的设计,使得生成的迁移路径能够完全覆盖依赖迁移组。引入自适应的遗传算子和模拟退火的接收机制来动态计算个体发生交叉和变异的概率并接收。最后针对可行迁移路径的长度不可预测的问题引入相应的修补算子。

#### 4.5 可行迁移路径生成

本文提出的 OGGA 算法以生成符合条件的染色体为最优解,得到最优解需要的迭代次数代表了算法的收敛效率,同时以是否达到算法预设的最大迭代次数为终止条件。

算法具体步骤如下:

Step1 设置初始化参数,包括初始种群规模  $n_{pop}$ 、最大遗传代数  $G$ 、交叉概率  $p_c$ 、变异概率  $p_m$  等参数。

Step2 由 4.1 节所述的染色体表示方式和 4.2 节所述的染色体初始化方法,产生初始种群  $P$ 。

Step3 使用 4.3 节中的算法 2 计算适应度值。

Step4 检查是否满足操作结束的条件,即个体的适应度值到达期望值或是到达预设的最大迭代次数,若满足,则执行步骤 8。

Step5 对种群中的个体进行选择,然后使用交叉和变异算子进行自适应的交叉和变异操作。

Step6 将新产生的个体根据 4.3 节中的算法 3 修补至

长度  $L_r$ , 产生新的种群集合  $P'$ 。

Step7 重复步骤 3—步骤 6,若达到增加  $L_r$  所需的  $G_r$ ,则个体长度为  $L_r + 1$ ,然后进入下一轮迭代。

Step8 停止遗传操作,输出满足条件的个体及其覆盖率。

### 5 实验与分析

为分析验证本文介绍的算法的有效性及其性能,将实验分为两部分。5.1 节以 inres initiator 和 automatic teller machine(ATM)两个模型为例,对比本文设计的算法与其他方法在生成 FTP 上的成功率。5.2 节以 3 个模型为例,对比生成相同条数的 FTP 所需的遗传迭代次数,根据结果分析本文所提出的改进算法的性能以及在加快求解速度上的效果。

#### 5.1 算法有效性实验分析

为了验证本文算法生成 FTP 的有效性,本节将以 inres initiator 和 ATM 两个模型为例,对比本文算法与文献[19]、文献[20]、文献[21]以及文献[22]中的方法生成 FTP 的成功率,两个系统的模型如图 7 和图 8 所示。为了简化条件和动作,模型中只表示了涉及 DTG 的守卫条件和动作,删除了与内部变量无关的条件和动作,迁移用  $t_n$  表示,实验中设置的遗传参数如表 2 所列。

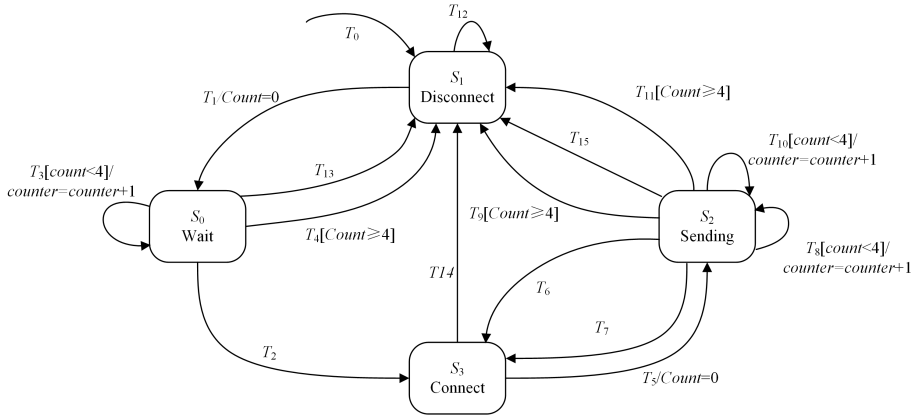


图 7 inres initiator 系统 FBM 模型

Fig. 7 FBM model of inres initiator system

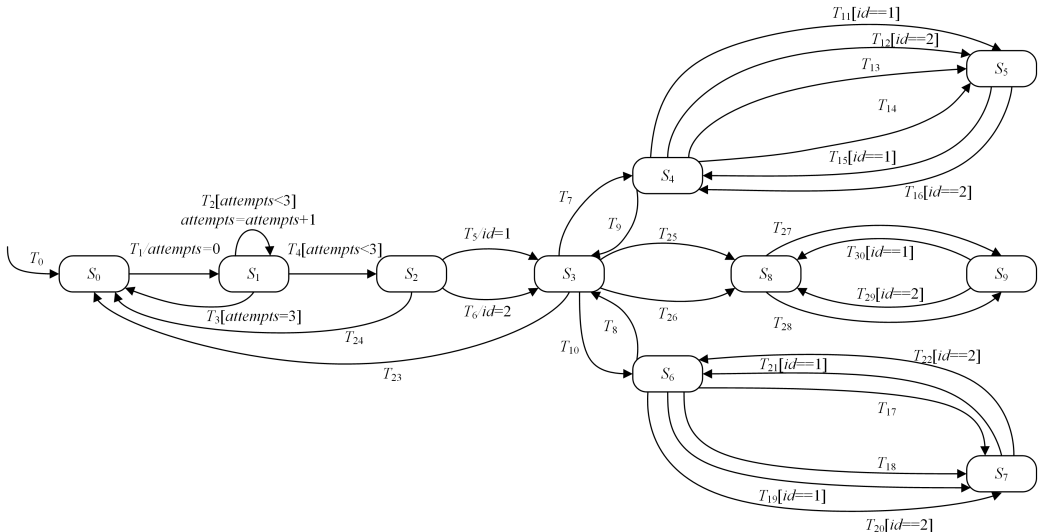


图 8 ATM 系统 FBM 模型

Fig. 8 FBM model of ATM system

表 2 两个模型所取的遗传参数

Table 2 Genetic parameters taken from two models

Model	$n_{pop}$	$p_c$	$p_m$	$E$
ATM	70	100	5	2
inres initiator	60	100	5	2

在两个系统 FBM 模型的基础上,使用给定的遗传参数生成 FTP,5 种方法分别为:文献[19]使用的传统遗传算法(GA),文献[20]设计的一种从状态图中自动生成测试用例(AGeTeSC)的方法,文献[21]中介绍的粒子群优化算法(PSO),文献[22]中的分组遗传算法(GGA)以及本文采用的优化分组遗传算法(OGGA)。5 种方法生成 FTP 的成功率的对比结果如表 3 所列。另外,实验中的 GGA 和 OGGA 生成的 FTP 满足迁移全覆盖。

表 3 生成可行迁移路径的成功率结果比较

Table 3 Comparison of success rate results of transition

算法	path generation				成功率/%	
	可执行路径	总路径数	成功率/%	可执行路径	总路径数	成功率/%
GA	87	36	90	45	96.7	80.0
AGeTeSC	75	48	89	67	84.3	71.6
PSO	59	37	65	50	90.8	74.0
GGA	2000	4000	2000	4000	100.0	100.0
OGGA	600	500	600	500	100.0	100.0

从表 3 可以看出,对于 ATM 和 inres initiator 两个系统模型,采用本文方法生成的迁移路径为 FTP 的成功率均为 100%,与采用 GGA 的成功率持平;相比使用传统 GA 分别提高了 3.3% 和 20%;相比应用 AGeTeSC 分别提高了 15.7% 和 28.4%;相比使用 PSO 分别提高了 9.2% 和 26%。本文提出的 OGGA 将依赖迁移保护在染色体的分组中,使得其依赖关系在经历交叉变异时不被破坏,因此能够提高所生成迁移路径可行的成功率。在利用 GGA 生成迁移路径的实验中,两个系统的初始种群规模  $n_{pop}$  均为 80,相比本文提出的 OGGA,初始种群规模分别减小了 12.5% 和 25%。说明 OGGA 能够在初始种群规模更小的情况下达到相同的迁移覆盖效果。

## 5.2 算法性能实验分析

本节对 3 个具有依赖迁移的模型使用所提出的 OGGA 来生成 FTP,并比较其与 GGA 在生成迁移路径时所需的遗传代数等。本实验的目的有两个:1)验证本文提出的 OGGA 的性能;2)证明 OGGA 可以解决具有依赖迁移的模型难以生成 FTP 的问题。警报系统(AS)、比赛判定系统(GWC)、错误等级检测系统(FLC)这 3 个模型均来自文献[22],模型图如图 9—图 11 所示。可以看出,3 个系统模型中均包含 DTG,且其中变量的值在不断增大,这也使得生成 FTP 的难度逐渐增大。

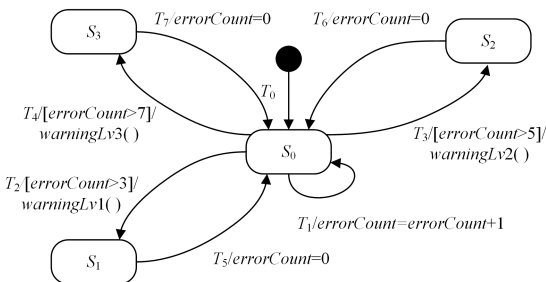


图 9 AS 系统 FBM 模型

Fig. 9 FBM model of AS system

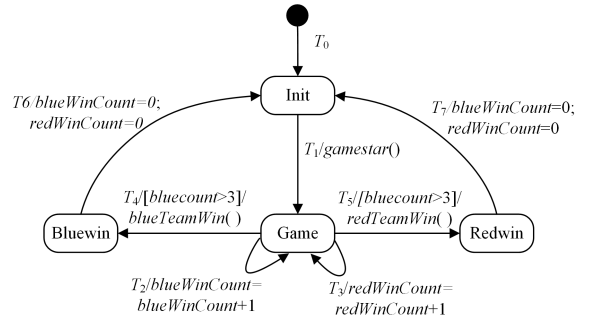


图 10 GWC 系统 FBM 模型

Fig. 10 FBM model of GWC system

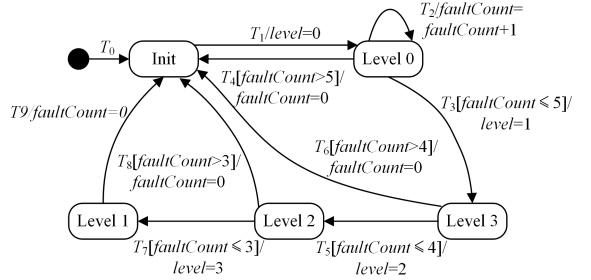


图 11 FLC 系统 FBM 模型

Fig. 11 FBM model of FLC system

为了得到准确的对比结果,遗传参数等参照文献[22]中的数值,即对每个模型同样生成 500 条迁移路径,遗传参数设置为: $n_{pop}=80$ , $p_c=100$ , $p_m=5$ ,和  $E=2$ 。实验生成的迁移路径均为可行迁移路径,实现了迁移覆盖。实验结果如表 4 所列,可以看出,由于本文提出的 OGGA 引入了自适应的遗传算子和基于模拟退火的接收机制,收敛速度相比 GGA 有了显著提升,生成相同数量的 FTP 所需的平均迭代次数分别减少了 15.24%,13.91% 和 16.29%,生成的迁移路径的平均长度大致相等。

表 4 生成迁移路径的结果比较

Table 4 Results comparison of generative ransition paths

模型	$G_{avg}$		$L_{avg}$		$n_{ftp}$	
	GGA	OGGA	GGA	OGGA	GGA	OGGA
AS	125.50	106.37	25.22	25.87	500	500
GWC	68.07	58.60	15.60	15.32	500	500
FLC	411.20	342.22	30.16	29.71	500	500

通过本文提出的 OGGA,随着迭代的进行,限制长度  $L_r$  逐渐增加,染色体随之覆盖所有依赖迁移对,最后创建一个覆盖所有迁移的 FTP。综上所述,本文使用 OGGA 生成测试用例的方法可以同时满足可执行和对迁移的完全覆盖,另外引入的自适应遗传算子和接收机制也有效加快了迭代的收敛速度,使其具有更优的性能。

**结束语** 本文首先提出了一种新的 FBM 来描述现代 Web 应用的动态行为,它不仅可以描述网页和事件,还表示出了事件触发条件和后续动作。然后针对存在依赖迁移的模型使用遗传算法生成的测试用例不可行的问题,提出了一种优化的分组遗传算法 OGGA。将依赖迁移分成一组并设置固定值,以此来保护特定迁移组不受遗传操作的影响。通过对染色体初始化和适应度函数的设计,使得生成的迁移路径

能够完全覆盖依赖迁移组。使用自适应的遗传算子动态计算个体交叉和变异的概率,并用基于模拟退火的接收机制来接收。由于满足模型迁移覆盖的可行迁移路径是不可预测的,所以引入修补算子对个体进行修补,同时在迭代的过程中逐渐增加长度限制。实验验证了本文算法生成可行迁移路径的有效性(即测试用例的可行性),并在缩小种群规模及减少迭代次数方面有了明显的提升。

在未来的工作中,我们将尝试提出一种能够自动构建前端行为模型的方法,来进一步提高 Web 应用测试用例生成的自动化。由于 Web 应用的安全威胁有可能来自于后端代码漏洞,因此,如何将前端行为模型与后端相结合来进一步提高 Web 应用测试用例的可靠性也是需要继续研究的内容。

## 参考文献

- [1] LEBEAU F, LEGEARD B, PEUREUX F, et al. Model-based vulnerability testing for web applications[C]//2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops. IEEE, 2013:445-452.
- [2] ANDREWS A, ALHADDAD A, BOUKHRIS S. Black-box model-based regression testing of fail-safe behavior in web applications[J]. Journal of Systems and Software, 2019, 149: 318-339.
- [3] GAO P, XU Y, SONG F, et al. Model-based automated testing of JavaScript Web applications via longer test sequences[J]. Frontiers of Computer Science, 2022, 16(3): 1-14.
- [4] JAIN N, PORWAL R. Automated test data generation applying heuristic approaches-a survey[M]//Software Engineering. Singapore: Springer, 2019: 699-708.
- [5] SHARMA A, PATANI R, AGGARWAL A. Software testing using genetic algorithms[J]. International Journal of Computer Science & Engineering Survey, 2016, 7(2): 21-33.
- [6] PANIGRAHI S S, JENA A K. Spider Monkey Particle Swarm Optimization(SMP SO) With Coverage Criteria for Optimal Test Case Generation in Object-Oriented Systems[J]. International Journal of Open Source Software and Processes(IJOSSP), 2022, 13(1): 1-20.
- [7] SHIROLE M, KUMAR R. UML behavioral model based test case generation: a survey[J]. ACM SIGSOFT Software Engineering Notes, 2013, 38(4): 1-13.
- [8] JOSHI S, AGRAWAL N, KRISHNAPURAM R, et al. A bag of paths model for measuring structural similarity in web documents[C]//Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2003:577-582.
- [9] MIRZAAGHAEI M, MESBAH A. DOM-based test adequacy criteria for web applications[C]//Proceedings of the 2014 International Symposium on Software Testing and Analysis. 2014: 71-81.
- [10] PRETSCHNER A, PRENNINGER W, WAGNER S, et al. One evaluation of model-based testing and its automation[C]//Proceedings of the 27th International Conference on Software Engineering. 2005:392-401.
- [11] MATTIELLO G R, ENDO A T. Model-based testing leveraged for automated web tests[J]. Software Quality Journal, 2021: 1-29.
- [12] PRADHAN S, RAY M, SWAIN S K. Transition coverage based test case generation from state chart diagram[J]. Journal of King Saud University-Computer and Information Sciences, 2019, 34(3): 993-1002.
- [13] FALKENAUER E. A hybrid grouping genetic algorithm for bin packing[J]. Journal of Heuristics, 1996, 2(1): 5-30.
- [14] FALKENAUER E. A new representation and operators for genetic algorithms applied to grouping problems[J]. Evolutionary Computation, 1994, 2(2): 123-144.
- [15] GAO X D, ZHOU L J, ZHANG S D, et al. Research on test data automatic generation based on improved genetic algorithm[J]. Computer Science, 2017, 44(3): 209-214.
- [16] LIN M, LIU B X, LIN X Y. Hybrid discrete cuckoo search algorithm with metropolis criterion for traveling salesman problem [J]. Journal of Nanjing University (Natural Sciences), 2017, 53(5): 972-983.
- [17] MCCABE T J. A complexity measure[J]. IEEE Transactions on Software Engineering, 1976(4): 308-320.
- [18] CRUZ-LEMUS J A, MAES A, GENERO M, et al. The impact of structural complexity on the understandability of UML state-chart diagrams[J]. Information Sciences, 2010, 180(11): 2209-2220.
- [19] KALAJI A S, HIERONS R M, SWIFT S. An integrated search-based approach for automatic testing from extended finite state machine(EFSM) models[J]. Information and Software Technology, 2011, 53(12): 1297-1318.
- [20] SWAIN R, PANTHI V, BEHERA P K, et al. Automatic test case generation from UML state chart diagram[J]. International Journal of Computer Applications, 2012, 42(7): 26-36.
- [21] KALAEI A, RAFAE V. Model-based test suite generation for graph transformation system using model simulation and search-based techniques [J]. Information and Software Technology, 2019, 108: 1-29.
- [22] CHOI Y M, LIM D J. Automatic feasible transition path generation from UML state chart diagrams using grouping genetic algorithms[J]. Information and Software Technology, 2018, 94: 38-58.



**LIU Ziwen**, born in 1998, postgraduate. His main research interests include software test, computer control, computer software and machine learning.



**YU Lijuan**, born in 1975, associate professor, graduate supervisor. Her main research interests include computer control, new energy system control and artificial intelligence.