

基于关键边概率与路径层接近度的多路径覆盖测试

钱忠胜, 成轶伟, 俞情媛, 张 丁, 姚昌森, 秦朗悦

(江西财经大学信息管理学院, 江西南昌 330013)

摘要: 遗传算法解决多路径覆盖中难覆盖边的问题, 是当前软件测试数据自动生成领域的一个研究热点. 现有方法解决多路径覆盖问题的效果不够理想, 本文提出一种将关键边概率与路径层接近度相结合的多路径覆盖测试方法. 首先, 本文计算节点被穿越概率找到难覆盖节点, 通过难覆盖节点找到难覆盖边(即, 关键边), 生成目标路径. 然后, 本文根据关键边概率计算个体贡献度, 并通过程序的路径层图计算路径层接近度, 再由个体贡献度及路径层接近度设计适应度函数. 最后, 本文利用多种群遗传算法进化生成测试数据以覆盖目标路径, 在进化过程中子种群覆盖当前目标路径后, 继续尝试覆盖与其相似的其它路径. 实验结果表明, 该方法与同类经典方法相比, 在保证平均进化时间和平均进化代数占优的同时, 稳定性也有所提高, 生成时间增幅标准偏差较最优的降低 10.19%, 离散系数降低 10.79%. 进化代数增幅标准偏差较最优的降低 19.98%, 离散系数降低 28.02%.

关键词: 关键边概率; 路径层接近度; 多路径覆盖; 遗传算法; 适应度函数; 个体贡献度

基金项目: 国家自然科学基金(No.62262025); 江西省自然科学基金(No.20224ACB202012)

中图分类号: TP311

文献标识码: A

文章编号: 0372-2112(2023)05-1341-09

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20220983

An Approach to Multi-Path Coverage Testing Based on Key Edge Probability and Path Layer Proximity

QIAN Zhong-sheng, CHENG Yi-wei, YU Qing-yuan, ZHANG Ding, YAO Chang-sen, QIN Lang-yue

(School of Information Management, Jiangxi University of Finance & Economics, Nanchang, Jiangxi 330013, China)

Abstract: Using genetic algorithms to solve the problem of difficultly-covered edges in multi-path coverage is a hot research spot in the current field of automatic test data generation. An approach to multi-path coverage testing that combines the key edge probability and path layer proximity is proposed, for the existing methods are not efficient enough to solve the multi-path coverage problem. Firstly, it calculates the probabilities of the nodes being traversed to get difficultly-covered nodes, and then finds difficultly-covered edges (i.e., key edges), so as to generate the target paths. Secondly, the individual contribution is calculated according to the key edge probability, and the path layer proximity is computed through the path layer graph of the program, and then the fitness function is designed from the individual contribution and the path layer proximity. Finally, multi-population genetic algorithm is employed to generate test data in order to cover the target paths. After the subpopulation covers the current target path in the evolution process, it continues to try to cover other paths similar to the current target path. Experimental results show that compared with those similar classic methods, this approach guarantees an improved stability besides dominant average generation time and average evolution time. The standard deviation of the generation time increase is lower than the optimal one by 10.19%, and the variation coefficient is lessened by 10.79%. The standard deviation of evolutionary time increase is lower than the optimal one by 19.98%, and the variation coefficient is decreased by 28.02%.

Key words: key edge probability; path layer proximity; multi-path coverage; genetic algorithm; fitness function; individual contribution

Foundation Item(s): National Natural Science Foundation of China (No.62262025); Jiangxi Provincial Natural Science Foundation (No.20224ACB202012)

1 引言

目前对软件测试数据生成的研究较为广泛,研究者从不同角度提升测试效率. Sharifdipour 等^[1]提出一种模拟蚁群优化算法,提高蚂蚁的搜索能力. Jatana 等^[2]利用粒子群优化与变异测试来生成测试数据. 钱忠胜等^[3]通过支持向量机回归模型预测个体适应度,并在进化中重用优秀个体. 范书平等^[4]根据分支均衡度设计合理适应度函数,以提高进化生成效率. 钱忠胜等^[5]通过关键点概率计算贡献度,据此设计适应度函数,进化生成测试数据. Qian 等^[6]改进个体信息共享方法,充分利用种群资源. 杜莹等^[7]利用精英策略建立优化模型,得到最优目标路径集. Dang 等^[8]提出一种模糊聚类 and 进化优化增强变异测试的方法. Sun 等^[9]利用 ESM (Ensemble Surrogate Model) 模型生成测试数据. Kalaipriyan 等^[10]提出改进搜索策略的定向人工蜂群算法解决多目标优化问题.

由上可知,国内外学者已在测试生成领域取得一定成果,这些成果大多围绕多路径覆盖展开研究,但仍存在一定不足. 现有判断路径难易覆盖的方法均从节点的角度考虑,路径覆盖难易程度和其包含节点的难易覆盖程度有一定关系,但还有很多其它因素. 比如,包含几个难覆盖节点,并不代表该路径一定难覆盖. 此外,如何恰当地设计多路径覆盖中遗传算法的适应度函数,也值得进一步研究.

基于此,本文提出一种多路径覆盖测试,根据关键边概率和路径层接近度设计适应度函数,使优秀个体在进化过程中得到保护,同时结合个体信息共享策略的优势,既可保证个体信息在进化中被恰当地利用,又能避免进化时间被过多浪费,可进一步改善进化生成多路径覆盖测试数据的效率.

2 相关定义

为便于理解,先定义上面提到的相关概念,包括关键边概率、个体贡献度、路径层接近度等.

2.1 关键边概率

定义 1 关键边概率: 难覆盖边被测数据穿越的概率, 记为 $G(E_{i,j})$. 其中 $E_{i,j}$ 为难覆盖节点 N_i 与 N_j 构成的关键边. 其概率为这两个节点的概率乘积.

根据被测程序, 随机生成测试数据, 统计测试路径集中各节点的覆盖情况, 可用 $T_{i,j}$ 表示为式(1).

$$T_{i,j} = \begin{cases} 1, & \text{测试路径穿越节点 } N_j \\ 0, & \text{测试路径未穿越节点 } N_j \end{cases} \quad (1)$$

输入 m 组数据到被测程序中, 可得到 m 条穿越路径组成的集合. 这样, 可构成穿越矩阵, 记为 C , 可表示为式(2).

$$C = \begin{bmatrix} T_{1,1} & \cdots & T_{1,n} \\ \vdots & \ddots & \vdots \\ T_{m,1} & \cdots & T_{m,n} \end{bmatrix} \quad (2)$$

其中, 穿越矩阵的行代表执行被测程序得到的路径; 列代表路径中节点的被穿越情况. 根据穿越矩阵 C , 得到穿越节点 N_j 的路径数, 记为 S_j , 表示为

$$S_j = \sum_{i=1}^m T_{i,j} \quad (3)$$

被测程序的节点概率 $G(N_j)$, 可表示为

$$G(N_j) = S_j/m \quad (4)$$

被测程序关键边概率 $G(E_{i,j})$, 可表示为

$$G(E_{i,j}) = G(N_i) \times G(N_j) \quad (5)$$

2.2 个体贡献度

在进化算法中为判定个体优劣程度, 可通过计算个体贡献度以考虑其是否需要被保留.

定义 2 个体贡献度: 个体对进化生成目标路径所做贡献, 记为 $Con = (l_x, P_{tar})$. 其中 $l_x (1 \leq x \leq m)$ 表示个体数为 m 的种群中某个个体, P_{tar} 为目标路径集 Γ_{tar} 中某条目标路径.

个体贡献度可表示为式(6), 其中 $E_{i,j}$ 表示 l_x 对应的路径 P_{l_x} 与目标路径 P_{tar} 的相同关键边.

$$Con(l_x, P_{tar}) = \sum e^{-G(E_{i,j})} \quad (6)$$

2.3 路径层接近度

程序往往带来嵌套结构, 为在适应度函数中反映此结构带来的影响, 给出路径层图、路径层接近度等相关概念.

定义 3 路径层图 (Path Layer Graph, PLG): 路径中各节点层级的图形描述, 记为 $GL = (N, E, \rho)$. 其中 N 为节点集, 各节点均为基本语句块; E 为边集, 各边对应语句流向; ρ 为节点层级集合, 层级从外层到内层按顺序编号, 最外层为第一层, 层级越高, 其数值越大, 表示该节点越难被执行到.

从定义 3 不难看出, 可通过赋予路径上的各条边不同的权重, 从而反映嵌套结构带来的差异性.

定义 4 路径层接近度: 测试路径与目标路径所在层的接近程度, 记为 $\gamma(P_{l_x})$, 其中 P_{l_x} 为测试数据路径. 路径层接近度可表示为

$$\gamma(P_{l_x}) = \frac{\sum \rho(E_{i,j})}{\sum \rho(E_{x,y})} \quad (7)$$

$$E_{i,j} \in P_{l_x} \cap P_{tar}, E_{x,y} \in P_{tar}; 1 \leq i, j, x, y \leq n$$

其中, $E_{x,y}$ 是目标路径中的边, $E_{i,j}$ 是测试路径与目标路径的共同边, $\rho(E_{x,y})$ 与 $\rho(E_{i,j})$ 分别为边 $E_{x,y}$ 与 $E_{i,j}$ 的层级, 见式(8).

$$\rho(E_{h,k}) = \frac{\rho(N_h) + \rho(N_k)}{2}, 1 \leq h, k \leq n \quad (8)$$

其中, $\rho(N_h)$ 和 $\rho(N_k)$ 分别表示节点 h 和 k 的层级. 对于式(8), 因边由节点连接, 故计算边的层级应考虑到相关节点的层级.

3 关键边概率与路径层接近度结合的多路径覆盖

本文给出一种多路径覆盖测试, 主要包括个体适应度模块和遗传算法模块. 其总体框架如图 1 所示.

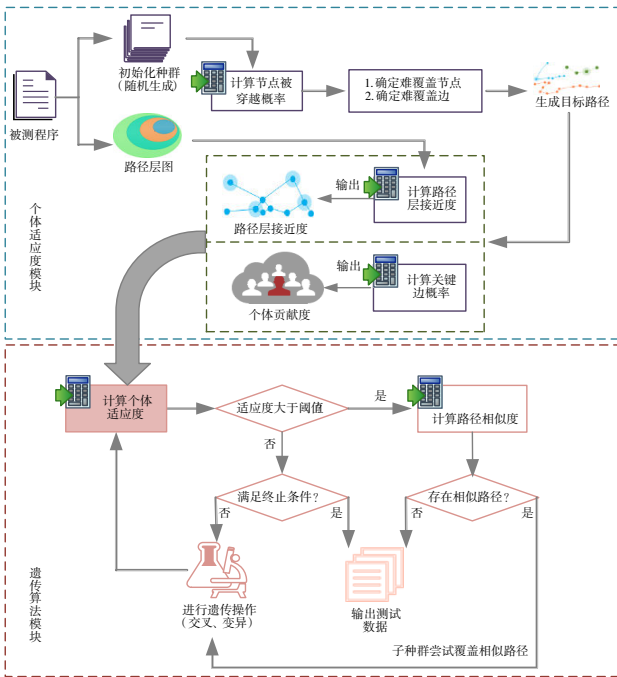


图 1 多路径覆盖测试总体框架

3.1 目标路径选取

为获得目标路径, 先计算被测程序的节点被穿越的概率, 概率为 0 标记为不可达节点, 为 1 标记为必达节点. 对于其它节点根据设定的阈值判断(阈值是所有节点被穿越概率的平均值, 该值会随着每组测试数据动态变化), 低于阈值的记为难覆盖节点. 若路径中某边的两节点均为难覆盖节点, 则该边为难覆盖边(即, 关键边), 路径则看成是目标路径. 按照各目标路径中难覆盖边概率之和对目标路径排序, 值越大, 排名越靠前, 根据实验需要, 择优挑选目标路径.

3.2 路径层接近度计算

处理结构复杂(如, 嵌套层级多)的被测程序时, 若能一定程度上反映各语句的执行情况, 会对提升测试数据生成效率有很大帮助. 因此提出路径层接近度概念, 其优势在于, 可通过赋予各节点及边不同权重来反映嵌套结构给语句执行带来的影响. 主要步骤有:(a) 设置节点层级;(b) 选取目标路径集;(c) 输入测试数据得到路径集;(d) 根据式(8)计算目标路径中所有边的

层级;(e) 寻找与目标路径中相同的边;(f) 利用式(7)计算路径层接近度.

3.3 关键边概率与路径层接近度结合的适应度函数设计

本文采用关键边概率和路径层接近度结合的方法设计适应度函数. 其原因是, 利用关键边概率可指引测试数据朝难覆盖路径演化, 同时通过路径层接近度能降低测试生成中因嵌套结构带来的影响, 两者结合会提高测试数据进化生成效率. 过程如下:

(1) 计算关键边概率: 各个边的穿越概率反映的是其在可达路径上的被覆盖难易程度, 概率越小则越难被覆盖, 由关键边(即, 难覆盖边)构成的路径即为难覆盖路径.

(2) 计算个体贡献度: 对个体优劣程度进行判定, 就其个体贡献度进行计算, 作为适应度函数设计时的参数.

(3) 计算路径层接近度: 在已有层接近度方法的基础上, 考虑嵌套结构对程序语句带来的影响, 提出路径层接近度方法. 该方法通过赋予各节点及边不同的权重, 以反映嵌套结构带来的差异性.

(4) 设计适应度函数: 前面介绍的关键边概率、个体贡献度和路径层接近度用于适应度函数设计. 关键边概率用来检测难覆盖路径, 并计算个体贡献度以评价种群个体优劣; 路径层接近度用来平衡被测程序中, 因存在嵌套结构而使各节点被执行难度不一所产生的影响. 它们互为补充, 均用于解决多目标路径覆盖问题, 以提高算法的效率. 据此, 将根据关键边概率计算得到的个体贡献度与路径层接近度结合, 设计适应度函数, 记为 $Fit(P)$, 可表示为

$$Fit(P) = \gamma(P) \times Con(l_x, P_{tar}) \quad (9)$$

其中, $\gamma(P)$ 表示路径层接近度(见式(7)), $Con = (l_x, P_{tar})$ 为个体贡献度(见式(6)). 多路径覆盖是要寻找最优测试数据集以覆盖所有目标路径, 即寻找多个最大适应度函数值问题.

3.4 多路径覆盖测试数据进化生成

当种群覆盖其对应目标路径后, 可尝试继续覆盖与目标路径相似的其它路径, 提高覆盖率. 为此, 引入路径相似度概念, 对两条路径进行相似性判断.

定义 5 路径相似度: 目标路径集中路径 P_x 与 P_y ($(1 \leq x, y \leq n) \& (x \neq y)$) 相同关键边个数, 与它们最大关键边个数之比, 记为 $Sim(P_x, P_y)$, 表示为

$$Sim(P_x, P_y) = \frac{\sum_{i=1}^{\min(\text{len}(P_x), \text{len}(P_y))} I(E_{x,i}, E_{y,i})}{\max(\text{len}(P_x), \text{len}(P_y))} \quad (10)$$

其中, $\text{len}(P_x)$ 与 $\text{len}(P_y)$ 分别表示路径 P_x 与 P_y 的关键边

个数, $E_{x,i}$ 为路径 P_x 上第 i 个位置的关键边, $E_{y,j}$ 为路径 P_y 上第 j 个位置的关键边, $l(E_{x,i}, E_{y,j})$ 统计目标路径 P_x 与 P_y 相同关键边个数, 表示为式(11).

$$l(E_{x,i}, E_{y,j}) = \begin{cases} 0, & E_{x,i} \text{ 与 } E_{y,j} \text{ 不相同} \\ 1, & E_{x,i} \text{ 与 } E_{y,j} \text{ 相同} \end{cases} \quad (11)$$

通过 3.1~3.3 节的过程, 最后利用遗传算法对目标路径进化生成覆盖它们的测试数据, 见算法 1.

算法 1 测试数据进化生成

输入: 目标路径集 Γ_{tar} 与个体数 m

输出: 测试数据集 testData

BEGIN

1. Set parameters; //设置参数相关
2. Pop[j] with m individuals generated for each $P_i \in \Gamma_{\text{tar}}$;
//生成个体数 m 的子种群
3. FOR $j=0 \rightarrow \text{Pop.length}$ DO //遍历所有子种群
4. currentPop \leftarrow Pop[j];
5. FOR each $P_k \in \Gamma_{\text{tar}}$ DO //遍历所有目标路径
6. IF $P_k \in \text{list}$ THEN // P_k 存在列表 list 中
7. IF P_k has similar paths THEN
//据式(10)寻找相似目标路径
8. Pop[j] continues to try to cover similar paths;
//继续尝试覆盖相似路径
9. testData \leftarrow the results covering the similar paths;
10. ELSE
11. testData \leftarrow the results covering target paths in list;
12. END IF
13. END IF
14. Get the max Fit of P_k ; //据式(9)获取最大适应度值
15. Get the best individual in currentPop;
//获取种群中最优个体
16. Calculate the fitness M of the best individual;
//据式(9)计算最优适应度值
17. IF $M \leq \text{Fit}$ THEN //适应度值小于目标适应度值
18. IF Pop[j].evolveTime \geq maxEvolveTime THEN
//进化代数超过设定的阈值
19. Add P_k to list; //将目标路径 P_k 加入列表 list
20. ELSE
21. Genetic operations are performed on currentPop;
//进行遗传进化
22. END IF
23. ELSE
24. testData \leftarrow the results covering P_k ;
25. END IF
26. END FOR
27. END FOR
28. OUTPUT testData;
- END

3.5 算法复杂度分析与对比

接下来分析本文方法时间复杂度, 主要来自适应度函数设计与个体信息共享两部分. 适应度函数设计的复杂度影响因素主要包括关键边概率 G , 路径层接近度 γ , 某条路径中关键边个数 E 以及目标路径数 P , 这一部分的时间复杂度可表示为 $O(G\gamma EP)$. 个体信息共享部分的主要影响因素有: 适应度 F , 路径相似度 S , 种群个体数 I 与目标路径数 P , 该部分的时间复杂度可表示为 $O(FSIP)$. 因此, 我们方法的总时间复杂度为 $O(G\gamma EP + FSIP)$. 由算法 1 可知, 适应度函数设计与个体信息共享复杂度均为 $O(n)$, 总时间复杂度为 $O(2n)$.

为更直观地展现各对比方法的执行时间差异, 选择近几年与本文方法最为相似且实验环境及参数均一致的文献[4~6]方法, 分别分析它们的时间复杂度, 并利用三角形分类程序, 执行 100 次这些方法得到其对应的平均进化时间, 结果如表 1 所示.

表 1 各对比方法时间复杂度分析

对比方法	时间复杂度	平均进化时间/s
文献[4]方法	$O(mn)$	1.846
文献[5]方法	$O(2n)$	0.514
文献[6]方法	$O(n^2)$	1.957
本文方法	$O(2n)$	0.681

从表 1 可看出, 本文方法的时间复杂度明显低于文献[6]方法. 因文献[4]方法为单目标路径覆盖策略, 故其时间复杂度与执行次数 m 有关, 当 $m > 2$ 时, 显然没有本文方法的复杂度低. 与文献[5]方法相比, 本文方法与其时间复杂度相同, 在平均进化时间上并不占优. 但本文方法综合考虑路径中节点与关键边信息, 更合理地选取目标路径, 可更有效地引导测试数据朝目标路径演化, 且利用路径层接近度能很好地降低测试生成时因嵌套结构带来的影响, 提升算法稳定性.

4 实验及其结果分析

前面对基于关键边概率与路径层接近度的多路径覆盖测试进行了详细说明, 本节通过实验对算法的有效性进行对比分析.

实验均在 Win10 系统, 内存 8 GB, 处理器 AMD Ryzen 7 4800U with Radeon Graphics (16 CPUs), 主频 1.8 GHz, Java 编程语言, JDK1.8 环境下进行. 多种群遗传参数设置为交叉概率 0.9, 变异概率 0.1, 最大进化代数 10 000.

4.1 实验设计

为检验本文方法的性能, 从两方面展开讨论(即, 基础程序和工业程序, 其程序信息如表 2 所示), 将本文方法与其它经典方法对比.

PG4~PG8, PG10~PG11(源自 Software-artifact Infrastructure Repository), PG9, PG13(源自文献[3]), PG12

表2 程序信息表

程序类型	程序编号	程序名称	代码行数	函数个数	目标路径数
基础程序	PG1	三角形分类	16	1	6
	PG2	冒泡排序	13	1	6
	PG3	最大公约数计算	20	1	6
工业程序	PG4	Vector	254	1	16
	PG5	Replace	564	21	30
	PG6	Sed	8 063	2	30
	PG7	Space	9 546	136	30
	PG8	Flex	10 459	162	30
	PG9	Go	28 547	2 982	100
	PG10	Jmeter	43 400	389	100
	PG11	Ant	80 500	672	100
	PG12	Mcknap	1 620	4	30
	PG13	Noark	26 010	285	100

(源自文献[1]).

4.1.1 基础程序上的实验

为论证本文多路径覆盖测试效率,在三角形分类程序PG1上,先选择最近3年较权威文献[4~6]中的方法进行对比实验,如图2所示.

此外,从实验结果可知,本文策略与文献[4~6]方法在不同种群个体数上均能100%覆盖目标路径,如图3所示.

下面对本文策略与文献[4~6]方法生成测试数据的平均进化代数指标展开对比分析,如图4所示.

从图4可看出,本文方法的平均进化代数均低于文献[4,6]方法,在种群数为20、50和100时,本文方法的平均进化代数低于文献[5]方法.

由上面对PG1程序的实验可发现,大部分情况下本文策略在更少的时间与进化代数内覆盖目标路径,说明本文方法是有效的.接下来,再选择冒泡排序PG2和最大公约数求解PG3两个基础程序进行对比,其实验结果分别如图5、图6所示.

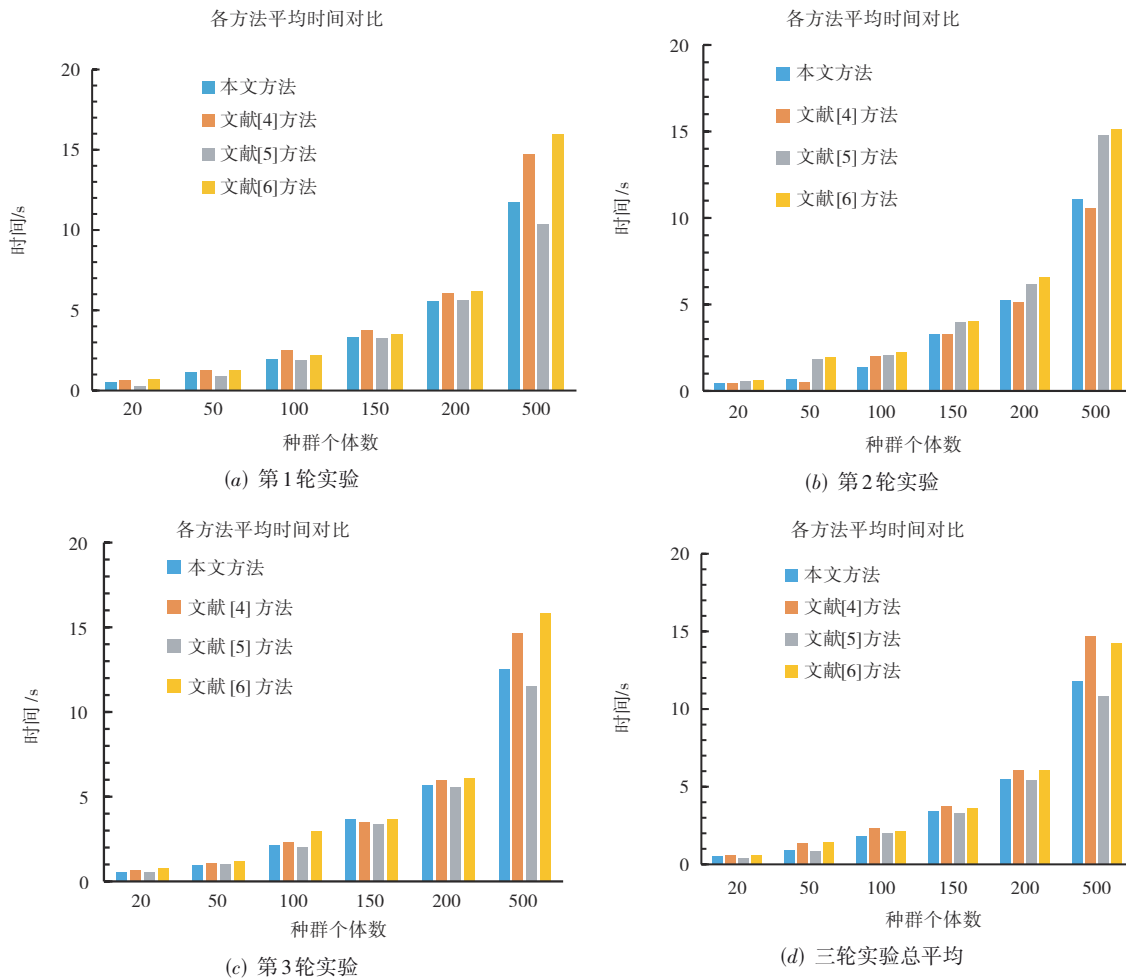


图2 PG1上测试数据生成平均时间对比

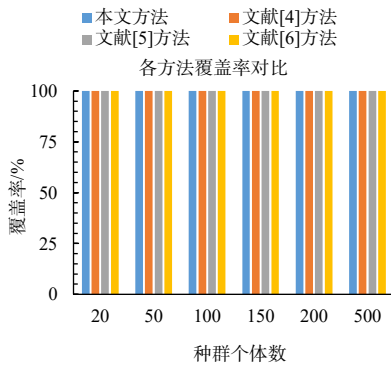


图3 PG1上目标路径覆盖率对比

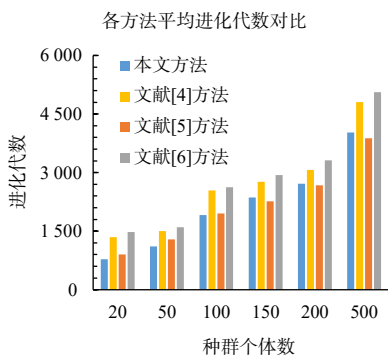


图4 PG1上测试数据生成平均进化代数对比

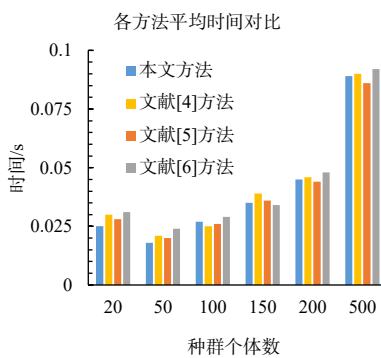


图5 PG2上测试数据生成平均时间

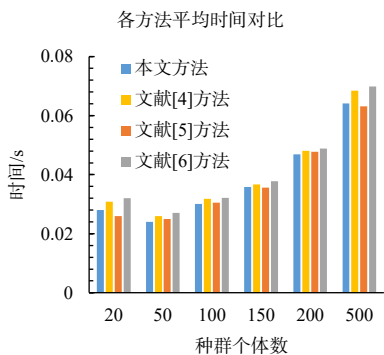


图6 PG3上测试数据生成平均时间

从以上实验结果可看出,本文方法准确性较高,能提高进化生成多路径覆盖测试数据的效率.本文方法与文献[4,6]中的方法相比,虽然覆盖率均为100%,但平均时间大部分情况下少于这两种方法,具有一定优势.相对于文献[5]方法,本文方法在时间方面优势不大明显,从理论上分析,本文方法将两节点间的逻辑关系加入到个体优劣的判断中,增加了路径覆盖难易程度的因素,会略有时间消耗,差距不大,但这样使本文方法的适应度函数设计更合理,稳定性也会有所提高.接下来在程序规模更大的工业程序上实验,对比以上方法的优劣,特别考虑了稳定性方面的因素.

4.1.2 工业程序上的实验

本节共选择10个工业程序进行实验,对比本文方法与其它方法的优劣.先对其中的5个工业程序,选取部分子函数,实验步骤同基础程序,在种群个体数均设置为100的条件下,对各方法分别进行50次实验,其结果如表3所示.

由表3可知,在进化代数阈值内,各方法均可100%覆盖目标路径;在平均生成时间及平均进化代数指标上,本文策略的平均生成时间,比文献[4,6]方法少得多;但相对于文献[5]来说,本文策略在多个指标上优势不明显,但根据4.1.1节的理论分析,我们适应度函数设计更合理,稳定性也会有所提高.

下面利用离散系数对各方法的稳定性进行检测.分别计算本文方法和文献[4~6]方法在5个工业程序上进化时间针对其在三角形分类程序上的增幅,计算它们的标准偏差和均值,得到离散系数.

标准偏差是一种衡量数据分散程度的指标,可表示为式(12),其中 \bar{X} 代表所采用的样本 X_1, X_2, \dots, X_n 的均值.

$$\text{Std} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2} \quad (12)$$

然而,标准偏差会受到测量尺度和量纲的影响,而离散系数表示标准偏差与均值之比,可将这种影响消除.离散系数值越高,其离散程度越大,稳定性越差,反之越好.离散系数 V 可表示为式(13),其中 Std 和 \bar{X} 分别来自式(12)中的标准偏差和均值.

$$V = \text{Std} / \bar{X} \quad (13)$$

本文方法与文献[4~6]方法的平均进化时间的离散系数对比如表4所示.

由表4可知,本文方法的离散系数低于文献[4]和文献[5]方法.与文献[6]方法相比,本文方法的离散系数略高,但通过在基础程序和工业程序上的实验结果表明,在平均进化时间上较文献[6]方法明显降低,本文方法依然占优.不同方法在进化代数上的离散系数对比如表5所示.

由表5可知,本文方法的离散系数低于文献[5]方法.鉴于本文方法在进化时间和进化代数上的离散系数

表3 工业程序上的实验结果

待测程序	指标	方法			
		文献[4]方法	文献[5]方法	文献[6]方法	本文方法
PG4	平均生成时间/s	28.91	23.60	36.49	26.81
	平均进化代数	4 725	3 468	5 005	4 155
	覆盖率/%	100	100	100	100
	平均生成时间增幅/(基于三角形分类程序)	77.97	76.50	83.26	78.76
	平均进化代数增幅/(基于三角形分类程序)	34.48	22.78	33.79	34.73
PG5	平均生成时间/s	427.01	370.95	456.80	386.31
	平均进化代数	6 403	5 491	6 975	5 848
	覆盖率/%	100	100	100	100
	平均生成时间增幅/(基于三角形分类程序)	98.51	98.50	98.66	98.53
	平均进化代数增幅/(基于三角形分类程序)	51.65	51.23	52.49	53.63
PG6	平均生成时间/s	174.80	126.75	193.66	145.63
	平均进化代数	8 432	6 292	8 635	6 936
	覆盖率/%	100	100	100	100
	平均生成时间增幅/(基于三角形分类程序)	96.36	95.62	96.85	96.09
	平均进化代数增幅/(基于三角形分类程序)	63.28	57.44	61.62	60.90
PG7	平均生成时间/s	72.16	52.57	84.67	57.41
	平均进化代数	5 192	4 184	5 312	4 463
	覆盖率/%	100	100	100	100
	平均生成时间增幅/(基于三角形分类程序)	91.17	89.45	92.79	90.08
	平均进化代数增幅/(基于三角形分类程序)	40.37	35.99	37.61	39.23
PG8	平均生成时间/s	134.77	115.29	152.20	120.70
	平均进化代数	8 432	5 947	7 869	6 561
	覆盖率/%	100	100	100	100
	平均生成时间增幅/(基于三角形分类程序)	95.27	95.19	95.99	95.28
	平均进化代数增幅/(基于三角形分类程序)	57.70	54.97	57.89	58.66

表4 关于进化时间的离散系数对比

方法	平均进化时间增幅/%					指标		
	PG4	PG5	PG6	PG7	PG8	标准偏差	平均值	离散系数
文献[4]方法	77.97	98.51	96.36	91.17	95.27	0.073 4	0.918 6	0.079 9
文献[5]方法	76.50	98.50	95.62	89.45	95.19	0.078 5	0.910 5	0.086 2
文献[6]方法	83.26	98.66	96.85	92.79	95.99	0.054 7	0.935 1	0.058 4
本文方法	78.76	98.53	96.09	90.08	95.28	0.070 5	0.917 5	0.076 9

表5 关于进化代数的离散系数对比

方法	平均进化代数增幅/%					指标		
	PG4	PG5	PG6	PG7	PG8	标准偏差	平均值	离散系数
文献[4]方法	34.48	51.65	63.28	40.37	57.70	0.106 9	0.495 0	0.215 9
文献[5]方法	22.78	51.23	57.44	35.99	54.97	0.131 6	0.444 8	0.295 9
文献[6]方法	33.79	52.49	61.62	37.61	57.89	0.110 5	0.486 8	0.227 1
本文方法	34.73	53.63	60.90	39.23	58.66	0.105 3	0.494 3	0.213 0

均低于文献[5]方法,因此本文方法的稳定性更好.与文献[4,6]方法相比,本文方法的进化代数离散系数更低,稳定性也更好.通过对进化时间和进化代数稳定性的实验说明4.1.1节的最后对稳定性的阐述是合理的.即,本文方法的适应度函数设计更合理,稳定性较好.

综上分析,本文方法较对比方法在中等规模的工业程序上略显优势,但在稳定性方面优势明显.接下来选择更大更复杂的工业程序,将本文方法与前面实验

中总体表现最好的文献[5]方法进行比较,以考察本文方法的优势.实验结果如表6所示.

本文方法在这三个大型复杂工业程序中相对文献[5]方法优势不大明显.在覆盖率方面,本文方法不如文献[5]方法能100%地覆盖目标路径,但差距不大.其原因是,文献[5]方法通过剔除易覆盖与不可达路径得到目标路径,而本文方法在目标路径选取时考虑路径中

表 6 两种方法生成测试数据的节约时间对比

待测程序	原始方法(仅采用插桩法)		文献[5]方法		本文方法		节约时间比	
	时间/s	覆盖率/%	时间/s	覆盖率/%	时间/s	覆盖率/%	文献[5]方法 较原始方法 节约时间/%	本文方法 较原始方法 节约时间/%
PG9	441.60	89	325.15	100	338.57	100	26.37	23.33
PG10	951.78	83	855.09	100	850.42	95	10.16	10.65
PG11	1 100.29	75	1 011.89	100	993.12	95	8.03	9.74

两节点间的逻辑关系,选出的目标路径相对更难被覆盖,但更合理且更有价值.在节约时间比方面,本文方法比文献[5]方法略占优,但仅考察这一指标并不能很好地反映本文方法的优势,因此接下来分析它们的稳定性,基于节约时间比计算两种方法的标准偏差、平均值以及离散系数,结果如表 7 所示.

由表 7 可知,本文方法的离散系数低于文献[5]方法,表明本文方法的稳定性更佳.

综上所述,本文方法较对比方法在大型复杂程序上优势不明显,差距不大,但在稳定性指标上本文方法

表 7 关于节约时间的离散系数对比

方法	节约时间比/%			指标		
	PG9	PG10	PG11	标准偏差	平均值	离散系数
文献[5]方法	26.37	10.16	8.03	0.081 9	0.148 5	0.551 4
本文方法	23.33	10.65	9.74	0.062 0	0.145 7	0.425 6

依旧占优.下面进一步将本文方法与国内外其它不同类型代表性方法进行对比.鉴于各方法的实验参数设置等均不一样,为体现对比实验的公平性,此处的实验参数均依据各对比方法来设定,结果如表 8 所示.

表 8 测试生成时间节约对比

待测程序	各种方法	覆盖率/%	平均进化时间/s	平均进化时间降低百分比/%
PG7	文献[7]方法	100	257.11	9.69
	本文方法	100	229.64	
PG8	文献[7]方法	100	283.47	10.72
	本文方法	100	253.09	
PG9	文献[3]方法	100	352.35	6.42
	本文方法	100	329.74	
PG12	文献[1]方法	100	535.14	50.71
	本文方法	100	263.78	
PG13	文献[3]方法	100	763.45	0.74
	本文方法	100	757.81	

由表 8 可知,这四种方法均能 100% 覆盖目标路径,但本文方法的平均进化时间更少.

总之,实验结果表明,本文方法优于大部分已有方法,虽然有时在时间、进化代数或覆盖率上略差于个别方法,但稳定性均占优.

4.2 有效性分析

我们利用方差分析对本文实验所得结果进行显著性分析,以检验其有效性,使用 SPSS22.0 软件对实验结果进行 F 检验.为便于分析,对 4.1.1 节的实验环境及参数均一致的各方法(即,文献[4~6]方法及本文方法),以程序 PG1 为例,对不同方法分别取种群个体数 100 条件下,50 组平均进化时间,对其进行方差分析,说明如下:

先假设各组平均进化时间无明显差异;接着计算统计量 $F=498.491$, F 界值为 2.666.对比 F 值与 F 界值可知,本文方法与文献[4~6]方法进化生成多路径覆盖测试数据时间构成的数据集得出的 F 值大于 F 界值,则假

设不成立.表明不同方法的平均进化时间差异显著,表明本文方法在生成覆盖目标路径测试数据的时间方面得到有效改善,排除实验中由于随机因素产生的影响.这说明本文方法可更加快速准确地进化生成测试数据.

5 总结与下一步工作

本文提出的一种基于关键边概率与路径层接近度的多路径覆盖策略,在进化代数及生成时间等方面较大部分方法更少,稳定性也有所提高.但当目标路径过多时,本文方法表现出的优势不太明显.接下来拟考虑结合机器学习方法,设计高效的模型来预测适应度值,并在进化过程中合理利用,有望进一步降低适应度计算时间,进而提升测试生成效率.

参考文献

- [1] SHARIFIPOUR H, SHAKERI M, HAGHIGHI H. Struc-

tural test data generation using a memetic ant colony optimization based on evolution strategies[J]. Swarm & Evolutionary Computation, 2018, 40(6): 76-91.

- [2] JATANA N, SURI B. Particle swarm and genetic algorithm applied to mutation testing for test data generation: A comparative evaluation[J]. Journal of King Saud University-Computer and Information Sciences, 2020, 2(4): 514-521.
- [3] 钱忠胜, 俞情媛, 宋涛, 等. 基于支持向量机回归模型的测试用例生成与重用[J]. 电子学报, 2021, 49(7): 1386-1391. QIAN Zhong-sheng, YU Qing-yuan, SONG Tao, et al. Test case generation and reuse based on support vector machine regression model[J]. Acta Electronica Sinica, 2021, 49(7): 1386-1391. (in Chinese)
- [4] 范书平, 张岩, 马宝英, 等. 基于均衡优化理论的路径覆盖测试数据进化生成[J]. 电子学报, 2020, 48(7): 1303-1310. FAN Shu-ping, ZHANG Yan, MA Bao-ying, et al. Evolutionary generation of test data for paths coverage based on balance optimization theory[J]. Acta Electronica Sinica, 2020, 48(7): 1303-1310. (in Chinese)
- [5] 钱忠胜, 祝洁, 朱懿敏, 等. 结合关键点概率与路径相似度的多路径覆盖策略[J]. 软件学报, 2022, 33(2): 434-454. QIAN Zhong-sheng, ZHU Jie, ZHU Yi-min, et al. Multi-path coverage strategy combining key point probability and path similarity[J]. Journal of Software, 2022, 33(2): 434-454. (in Chinese)
- [6] QIAN Zhongsheng, HONG Dafei, ZHAO Chang, et al. A strategy for multi-target paths coverage by improving individual information sharing[J]. KSII Transactions on Internet and Information Systems, 2019, 13(11): 5464-5488.
- [7] 杜莹, 孙百才, 巩敦卫, 等. 软件测试路径选择优化模型及其进化求解[J]. 软件学报, 2022, 33(9): 3297-3311. DU Ying, SUN Bai-cai, GONG Dun-wei, et al. Optimization model of path selection for software testing and its evolution-based solution[J]. Journal of Software, 2022, 33(9): 3297-3311. (in Chinese)
- [8] DANG Xiangying, GONG Dunwei, YAO Xiangjuan, et al. Enhancement of mutation testing via fuzzy clustering and multi-population genetic algorithm[J]. IEEE Transactions on Software Engineering, 2022, 48(6): 2141-2156.
- [9] SUN Baicai, GONG Dunwei, TIAN Tian, et al. Integrating an ensemble surrogate model's estimation into test data generation[J]. IEEE Transactions on Software Engineering, 2022, 48(4): 1336-1350.
- [10] KALAIPRIYAN T, RAJESWARI M, DEBNATH B, et al. Directed artificial bee colony algorithm with revamped

search strategy to solve global numerical optimization problems[J]. Automated Software Engineering, 2022, 29(1): 13.

作者简介



钱忠胜 男, 1977年1月出生, 江西鹰潭人. 2008年在上海大学获工学博士学位. 江西财经大学教授, 博士生导师. 主要研究方向为软件工程、机器学习、智能化软件等.
E-mail: changesme@163.com



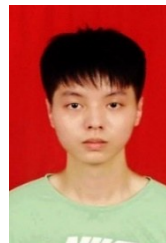
成轶伟 男, 1996年5月出生, 江苏泰州人. 江西财经大学信息管理学院硕士研究生. 主要研究方向为软件工程、智能推荐系统等.
E-mail: chengyiweiqq@163.com



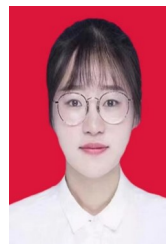
俞情媛 女, 1997年5月出生, 江苏淮安人. 江西财经大学信息管理学院博士研究生. 主要研究方向为软件工程、智能推荐系统等.
E-mail: 1045224864@qq.com



张丁 女, 1994年12月出生, 江西宜春人. 江西财经大学信息管理学院硕士研究生. 主要研究方向为软件工程、智能推荐系统等.
E-mail: 1006337533@qq.com



姚昌森 男, 1998年1月出生, 江西南昌人. 江西财经大学信息管理学院硕士研究生. 主要研究方向为软件工程、智能推荐系统等.
E-mail: 1124331766@qq.com



秦朗悦 女, 1998年4月出生, 江西南昌人. 江西财经大学信息管理学院硕士研究生. 主要研究方向为软件工程、智能推荐系统等.
E-mail: 502611360@qq.com