

# 基于复杂系统遗传算法的多路径覆盖测试用例生成方法

于 博 姜淑娟 张艳梅

(中国矿业大学计算机科学与技术学院 徐州 221116)

**摘 要** 针对目前复杂系统多路径覆盖测试用例生成方法较少的问题,提出一种新的基于复杂系统的多路径覆盖测试用例生成方法。首先改进遗传算法,在种群进化中对父代选择、个体进化的学习能力和种群的自适应更新方法进行改进,以有效避免算法收敛过慢或者“早熟”现象。然后根据多路径覆盖测试的特点与要求,设计基于路径匹配的适应度函数,使得运行一次算法便可生成覆盖多条目标路径的多个测试用例。最后将该方法用于几个基准程序。实验结果表明,与已有方法比较,此方法的测试用例生成效率显著提高。

**关键词** 复杂系统,多路径覆盖,测试用例生成,遗传算法,适应度函数

## Multiple Paths Test Case Generation Based on Complex System Genetic Algorithm

YU Bo JIANG Shu-juan ZHANG Yan-mei

(School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China)

**Abstract** In the light of the lack of effective methods to generating test case for multiple paths coverage based on complex system, we proposed a novel evolutionary generation approach of test case for multiple paths coverage. First, generic algorithm was improved; the ability of parent evolutionary selection, individual evolution and adaptive update method of populations were improved in the evolution of population, which could solve the algorithm for early slow convergence or premature effectively. And then, according to the features and requirements of multiple paths coverage, fitness function based on path-match was designed so that in one run, it was able to generate multiple test data to cover multiple target paths. Finally, the proposed approach was applied into several benchmark programs. The experimental results show that the proposed approach can improve the efficiency of test data generation effectively.

**Keywords** Complex system, Multiple paths coverage, Test case generating, Genetic algorithms, Fitness function

## 1 引言

测试用例的自动生成一直是被广泛研究的问题,该问题已转化成测试用例是否满足测试原则的一个搜索问题,这就使得全局搜索能力强、鲁棒性好的遗传算法(Genetic Algorithm,简称 GA)应用于测试用例生成技术。

约翰·霍兰德等人<sup>[1]</sup>于 20 世纪 60 年代在对细胞自动机进行研究时首次提出,遗传算法是用于解决最优化问题的搜索算法。遗传算法作为自适应随机搜索算法,还存在着早熟收敛(过早陷入局部最优解)和收敛速度慢这两个难题。目前,许多学者对这两个问题进行了相关研究,如 Rahila 等人<sup>[2]</sup>以带有种群和性别选择的 K 均值聚类算法来改进遗传算法;Dr. Velur Rajappa 等人<sup>[3]</sup>将基于图论的遗传算法应用到测试用例生成中;Tan 等人<sup>[4]</sup>将模拟退火算法与遗传算法相结合应用于测试用例生成,以提高算法效率。Ahmed 等人<sup>[5]</sup>把遗传算法生成路径覆盖的测试用例转化为多个优化问题,使得一次运行 GA 可以生成穿越多个目标路径的测试用例。本文改进 Ahmed 等人<sup>[5]</sup>提出的适应度函数的设计方案,

将程序控制流图的所有可达路径作为所需覆盖的目标路径,并借鉴庄健等人<sup>[6]</sup>提出的基于复杂系统的算法对其规则进行修正与完善,将其应用于测试用例自动生成。

## 2 算法设计

### 2.1 测试用例生成问题建模

基于遗传算法的路径覆盖测试用例生成问题可以定义为式(1):

$$\left. \begin{aligned} & \text{Fittest } \bar{X} \\ & \text{Accessto All } P \\ & \text{Minimize}^* G \\ & \text{Subjectto } \bar{X} \subset \bar{\Omega} \end{aligned} \right\} \quad (1)$$

式中, $X = \{x_1, x_2, \dots, x_m\}^T$ 为一组测试用例, $\bar{X}$ 是有效测试用例组合, $P = \{p_1, p_2, \dots, p_n\}$ 是可达路径集,Minimize\* G 表示尽可能减少进化代数 G, $\bar{\Omega} = \{X_1, X_2, \dots, X_N\}$ 为所有测试用例集合,N 为种群规模。

到稿日期:2011-11-08 返修日期:2011-12-14 本文受国家自然科学基金(60970032),江苏省自然科学基金(BK2008124),江苏省“青蓝工程”,江苏省研究生培养创新工程项目(CX10B\_157Z)资助。

于 博(1988-),男,硕士生,主要研究方向为软件分析与测试、进化计算等,E-mail:elias\_yu@163.com;姜淑娟(1966-),女,博士,教授,博士生导师,主要研究方向为程序设计语言、编译技术、软件工程;张艳梅 女,博士生,主要研究方向为软件分析与测试等。

根据遗传算法特性,  $X = \{x_1, x_2, \dots, x_m\}^T$  中的每个变量  $x_i = c_{i1}c_{i2}\dots c_{il}$  为一条染色体, 由  $l$  个遗传基因组成, 所以集合  $X$  可以表示为式(2);

$$X = \begin{bmatrix} c_{11} c_{12} \dots c_{1l} \\ c_{21} c_{22} \dots c_{2l} \\ \dots \\ c_{m1} c_{m2} \dots c_{ml} \end{bmatrix} \quad (2)$$

式中, 位于同列的  $c$  称为等位基因。一条染色体的基因个数  $l$  由问题规模决定; 问题的一组测试用例包含变量个数越多, 变量越长, 则  $l$  越大; 反之, 变量个数少, 变量越短, 则  $l$  越小。

## 2.2 遗传算法改进

### 2.2.1 基于“幂定律”的子代选择策略

生物进化的过程中, 少数个体占有绝大部分资源是一个非常普遍的现象, 在测试用例生成过程中, 由少数适应度高的个体作为父代个体进行繁殖, 能够使算法更好地进行收敛。本文算法对父代的选择引入“幂定律”(Power Law)<sup>[7]</sup>代替传统的“赌轮法”, 对母代的选择方式采用顺序选择的策略。因此在  $t$  代时,  $X_i$  被选为父代个体的概率如式(3)所示:

$$P(T_t(\bar{Q}) = X_i) = \frac{a^{-\text{rank}(f(X_i^t))}}{\sum_{j=1}^N a^{-\text{rank}(f(X_j^t))}} \quad (3)$$

$$\text{rank}(f(X_{\text{best}}^t)) = 0, \text{rank}(f(X_{\text{worst}}^t)) = N-1$$

式中,  $T_t$  为选择算子,  $\bar{Q}$  为  $t$  代种群,  $\text{rank}$  为排序函数,  $a$  为选择概率基数, 种群规模为  $N$  的情况下, 一代中适应度最好的个体排序函数值为 0, 适应度最差的个体排序函数值为  $N-1$ 。

使用“幂定律”策略选择父代, 使得种群一代中适值最高的个体被选为父代的概率远远大于其他个体。

例如, 式中以  $a=5$  为基数, 在具有一定种群规模的情况下, 种群一代中适值最高的个体被选为父代的概率接近 80%, 适值前两名的个体被选为父代的概率接近 96%。这样规划的优点是可以大幅加速遗传算法的收敛速度, 同时又不是完全选择最优个体进行繁殖, 是带有一定竞争进化的。缺点是易“早熟”, 因此, “幂定律”选择算子必须配合其他算法, 避免过早陷入局部最优解。

### 2.2.2 环境-基因双演化交叉算子

一般普通遗传算法的交叉算子只考虑父代的基因编码形式, 而往往忽略个体的学习能力以及环境对个体的影响<sup>[6]</sup>。自然界中存在的趋同进化现象说明了环境对生物的进化有着很大的影响, 生物从环境中进行后天学习对生物进化历程有着相当大的影响力, 生物可以通过改变周围环境等方法达到加速进化的目的, 把环境影响加入到种群进化中有助于扩大进化时基因的探索范围。本文针对积累历史经验的环境变量的获取将文献[6]中的环境-基因双演化交叉算子进行修正, 修正后的算法提高了交叉算子效率并使遗传算法具有一定的历史学习能力。新的环境-基因双演化交叉算子具体描述如式(4)所示:

$$T_t(X_i^{t+1} | X_i^t, X_j^t) = \begin{cases} X_i^t + r_1 E, & p_e \leq 0.5 \\ X_j^t + r_1 E, & p_e > 0.5 \end{cases} \quad (4)$$

$$E = \begin{cases} r_2 E^{-1} + r_3 (X_i^t - X_i^{t-1}), & p_e \leq 0.5 \\ r_2 E^{-1} + r_3 (X_i^t - X_j^t), & p_e > 0.5 \end{cases}$$

式中,  $t$  为进化代数,  $X_{i,t}$  为父亲,  $X_{j,t}$  为母亲,  $E$  为环境变量

集,  $r_1$  为学习系数,  $r_2$  为遗忘系数,  $r_3$  为修改系数,  $p_e = \text{random}(0, 1)$  为规则选择概率。

环境-基因双演化交叉算子可以使群体进化不仅局限于较优个体, 还可以对较优个体周围的空间进行探索, 提高交叉算子效率; 环境变量  $E$  可以获取较优个体的历史进化经验, 或者可以积累历史进化中环境对不同个体的影响, 使得群体进化时具有学习能力。

### 2.2.3 自适应更新策略

在 GA 种群规模  $N$  有限的情况下, 为了防止种群的“早熟”并且保证种群的多样性, 在 GA 中引入自适应更新策略, 如式(5)所示:

$$T_r(X_i^{t+1} | X_i^t, \hat{X}_i^t) = \begin{cases} \hat{X}_i^t, & f(\hat{X}_i^t) \geq f(X_i^t) \\ \hat{X}_i^t, & f(\hat{X}_i^t) < f(X_i^t), Np_r < i \leq N \\ X_i^t, & \text{otherwise} \end{cases} \quad (5)$$

式中,  $T_r$  为自适应更新策略算子,  $Np_r$  为更新规模变量,  $N$  为种群规模,  $\hat{X}_i^t$  为交叉、变异等操作后产生的新个体。更新规模变量  $Np_r$  也采用反馈的修改方式, 如式(6)所示:

$$Np_r^{t+1} = \begin{cases} Np_r + \alpha, & f(X_{\text{best}}^{t+1}) \leq f(X_{\text{best}}^t) \\ Np_r - \beta, & f(X_{\text{best}}^{t+1}) > f(X_{\text{best}}^t) \end{cases} \quad (6)$$

式中,  $Np_r^{t+1} \in [1, \text{PopulationSize}]$ ,  $\alpha$  为递增常量,  $\beta$  为递减常量。种群进化过程中, 当进化顺利时,  $Np_r$  增大, 优胜劣汰的机会增加, 有助于帮助算法收敛; 当种群陷入局部最优解而进化不明显时,  $Np_r$  减小, 子代直接替换父代的机会上升, 有助于解决种群“早熟”的问题。

## 2.3 适应度函数设计

在基于路径覆盖原则的测试用例生成中, 已知可达路径集合  $P$  中记录节点序号的多维数组的形式保存了每条可达路径。Ahmed 等人<sup>[5]</sup>首次提出组合目标路径与实际路径的判定条件差异(distance)和不匹配节点数(violation)的适应度函数设计思路, 本文中适应度函数评估要进行的工作是将每组测试用例得到的路径与每条已知可达路径比较, 得到两个适应值, 第一个适应值说明了该组测试用例得到的路径的匹配程度, 第二个适应值说明了该组测试用例得到的路径与不匹配的可达路径的前提条件的差距, 两个值构成适应值。具体 GA 适应度函数设计如式(7)所示:

$$f_1(X_i) = \sum_{j=1}^n c_j(p(X_i)) \prod_{k=1}^{\min(|p_j|, |p(X_i)|)} w_{jk} M_{jk}(p(X_i)) \quad (7)$$

$$f_2(X_i) = \sum_{j=1}^n D_{c_j(p(X_i))+1}(X_i)$$

$$F_i = \{f_1(X_i), f_2(X_i)\}$$

$$\vec{F} = \{F_1, F_2, \dots, F_N\}$$

式中,  $f_1(X_i)$  表示测试用例  $X_i$  得到路径的匹配程度,  $n$  表示可达路径数,  $p(X_i)$  是  $X_i$  得到的路径数据,  $c_j(p(X_i))$  表示  $X_i$  得到的路径与编号为  $j$  的目标路径中节点连续匹配的个数,  $|p_j|$  表示编号为  $j$  的目标路径中节点数目,  $|p(X_i)|$  表示  $X_i$  得到的路径的节点数目,  $w_{jk}$  是当前节点权重(节点的深度越大, 匹配时权值越大, 本文是线性设定权重),  $M_{jk}(p(X_i))$  表示  $X_i$  得到的路径与目标路径  $j$  当前编号为  $k$  的节点是否匹配(匹配值为 1, 不匹配值为 0)。  $f_2(X_i)$  表示求得路径与所有目标路径前提条件的差距,  $c_j(p(X_i))+1$  表示求得路径与  $j$

目标路径的第一个不匹配节点,  $D(X_i)$  为求得前谓词条件的差值。  $f_1(X_i)$  与  $f_2(X_i)$  为共同构成编号为  $i$  的个体的适应值,  $\bar{F}$  为种群适应值集合。

## 2.4 算法步骤

基于以上提出的复杂系统遗传算法的多路径测试数据生成步骤如下:

1. 插装被测程序。
2. 生成可达路径集, 判断  $P$  中可达路径的数目, 若为 0 则终止算法。
3. 设置 GA 参数(种群规模  $N$ , 进化代数上限  $G$ , 交叉概率  $p_c$ , 变异概率  $p_s$ , 个体染色体数目  $m$ , 基因位数  $l$ ; “幂定律”选择概率基数  $a$ ; 环境变量学习系数  $r_1$ , 遗忘系数  $r_2$ , 修改系数  $r_3$ ; 自适应更新策略更新规模变量  $Np_r$ , 递增常量  $\alpha$ , 递减常量  $\beta$ )。
4. 初始化种群。
5. 执行被测程序, 得到路径数据。
6. 种群进行适应值计算, 参照  $f_1(X_i)$  进行排序, 若有并列项则参考  $f_2(X_i)$ , 若仍并列则顺序随机排序, 保证无并列项。
7. 判断是否有完全匹配  $P$  中可达路径的个体  $X_i$ , 若有则记录该组测试用例, 并删除  $P$  中该路径编码。
8. 判断是否满足终止条件或判断  $P$  中可达路径的数目, 若为 0 则终止算法输出结果, 否则执行步骤 9。
9. 否则进行进化, 对种群中所有个体依次执行以下步骤:
  - 9.1 根据文中 2.2.1 节描述“幂定律”选择算子选择个体作为父代, 顺序选择其他个体作为母代。
  - 9.2 采用贝努利实验以概率为  $p_c$  判断交叉事件为真, 执行步骤 9.3; 否则, 以子代个体直接代替父代个体, 执行步骤 9.4。
  - 9.3 根据文中 2.2.2 节描述环境-基因双演化算子, 将父母杂交生子代个体, 按照式(4)修改环境变量。
  - 9.4 对子代个体的每个基因采用贝努利实验以概率为  $p_s$  判断变异事件是否为真, 如果是, 则该基因位变异。
  - 9.5 根据文中 2.2.3 节描述的种群自适应更新策略, 获得新一代种群, 并按照式(6)更新规模变量  $Np_r$ , 跳转至步骤 6。

2.5 算法设计分析

(1) 合理的适值函数是 GA 效率的核心

本文提出的适值函数清晰完善地表达了一组测试用例对路径覆盖的贡献大小。在实际计算中, 适值函数表达该用例与全部路径的前提条件的差距的部分可以不用计算, 如此便极大地减小了计算量, 只有在适值函数的第一部分, 两组或以上测试用例表达与全部可达路径匹配程度相等时, 再进入后一部分的计算。文中的适值函数不仅考虑了测试用例生成路径与可达路径集中匹配的节点的数目, 而且加入了两组权值, 一组为连续匹配节点长度, 另一组为匹配节点深度, 来衡量测试用例对路径覆盖的贡献, 较好的个体会更容易获得较高的适应值, 更有利于指导种群的进化。

(2) 复杂系统遗传算法的融入改善搜索效率

本文将复杂系统遗传算法融入测试用例生成过程, 该算法将生物界中既有的进化因素, 通过建模的方式, 重新考虑遗传算法的参数平衡关系与算法结构, 通过前人经验, 改进后的算法在解决高维搜索问题时有很高的效率在软件测试应用

中, 当面对复杂程度较高的程序时, 该算法的优势会更加突出, 可以帮助解决原有传统遗传算法收敛速度慢与“早熟”的问题, 并且增强搜索效率。

## 3 实验

### 3.1 实验环境及参数

为了验证本文方法的有效性, 我们选择 3 个基准程序: 三角形分类、冒泡排序、求最大最小值作为待测程序。程序结构如表 1 所列。实验代码用 Java 语言编写, 实验硬件环境: CPU: Intel Core i3-2100; 内存: 2G 1333Mhz。GA 参数设置如表 2 所列。

表 1 待测程序结构表

名称	结构
三角形分类	三个选择语句嵌套。
冒泡排序	两个循环嵌套, 内循环嵌套一个选择语句。
求最大最小值	一个循环嵌套两个顺序选择语句。

表 2 GA 参数设置

序号	参数名称	值
1	最大进化代数(G)	5000
2	交叉概率( $P_c$ )	0.7~0.9
3	变异概率( $P_s$ )	0.05~0.3
4	染色体条数(m)	1
5	幂定律基数(a)	5
6	环境变量学习系数( $r_1$ )	0.1
7	环境变量遗忘系数( $r_2$ )	0.2
8	环境变量修正系数( $r_3$ )	0.3
9	更新规模( $Np_r$ )	5
10	递增系数( $\alpha$ )	1
11	递减系数( $\beta$ )	2

### 3.2 实验结果

首先, 以三角形分类程序进行测试, 目标路径选择全部 4 条可达路径, 设定找到全路径或者进化代数超过 5000 为终止条件。与采用 Ahmed 方法的遗传算法进行对比, 进行 500 次实验, 对每次实验的进化代数与进化时间取平均值, 实验结果如表 3 所列。

表 3 三角形分类程序实验结果

种群范围	种群规模	平均进化代数			总进化时间(ms)		
		本文进化方法	Ahmed 方法	代数比 (%)	本文进化方法	Ahmed 方法	时间比 (%)
$[0, 128]^3$	30	17	63	27.0	4	3	125
$[0, 256]^3$	50	11	74	14.9	6	5	120
$[0, 512]^3$	80	11	352	3.1	11	31	35.5
$[0, 1024]^3$	100	10	512	1.9	14	67	20.9
$[0, 2048]^3$	200	9	629	1.4	30	206	14.6

由表 3 可以发现, 本文进化方法无论在时间还是在进化代数上均比 Ahmed 方法都有显著的提高, 并且随着种群范围与种群规模的同时扩大, 进化代数优化效果更好, 证明了改进算法对种群进化方向引导的正确性, 并且随着种群规模的提高, 进化效率明显提高。

第二组实验, 以求最大最小值程序进行测试, 目标路径选择全部可达路径, 设定找到全路径或者进化代数超过 5000 为终止条件。与采用传统进化方法的遗传算法进行对比, 进行 500 次实验, 对每次实验进化代数与进化时间取平均值, 实验结果如表 4 所列。遗传算法的传统进化方法指在种群进化过程中, 不使用本文方法, 仅使用“轮盘赌”方式的父代选择策

(下转第 153 页)

- [4] 邓水光,尹建伟,李莹,等. 基于二分图匹配的语义 Web 服务发现方法[J]. 软件学报,2008,31(8):1364-1374
- [5] Wang H C, Lee C S, Ho T S. 结合主观和客观 QoS 要素研究个性化 Web 服务选取问题[J]. 专家系统与应用,2007,32(2):571-584
- [6] Wang P. 用户模糊感知下基于 QoS 的 web 服务选取研究[J]. 专家系统与应用,2009,36(3):4460-4466
- [7] Tiwari S, Kaushik S. A Non-functional Properties-based Web Service Recommender System[C]// International Conference on Computational Intelligence and Software Engineering(CISE), IEEE Computer Society, 2009
- [8] Lou Yuan-sheng, Tao Zhen-hong, Yue Lu-lu. A QoS-based Web Service Dynamic Composition Framework[C]// Ninth International Symposium on Distributed Computing and Applications to

Business, Engineering and Science. IEEE Computer Society, 2010

- [9] Sui Xin, Wang Suo-zhu, Li Zhao-wei. Research on the Model of Integration with Semantic Web and Agent Personalized Recommendation System[C]// Proceedings of 13th International Conference on Computer Supported Cooperative Work in Design. IEEE Computer Society, 2009
- [10] Laliwala Z, Sorathia V, Chaudhary S. Semantic and Rule-based Event-driven Services-oriented Agricultural Recommendation System[C]// Proceedings of the 26th IEEE International Conference on Distributed Computing Systems Workshops (ICDC-SW'06). 2006
- [11] 李国勇,陈蜀宇,高峥. Web 服务中的跨应用单点登[J]. 重庆工学院学报:自然科学版,2011,25(2):68-73

(上接第 141 页)

略,根据交叉、变异概率随机地进行交叉、变异操作,并且直接选取子代替换父代的进化方式。

表 4 最大最小值程序实验结果

参数个数	种群规模	目标路径数	平均进化代数			总进化时间(ms)		
			本文进化方法	传统进化方法	代数比 (%)	本文进化方法	传统进化方法	时间比 (%)
3	100	8	3	49	6.1	2	15	13.3
4	100	16	4	615	0.7	4	249	1.6
5	100	32	20	>5000	—	23	—	—
6	100	64	178	—	—	333	—	—
7	100	128	4150	—	—	7741	—	—

由表 4 可以发现,本文进化方法在效率上有显著的提高。在路径数不断增大的情况下,并没失去可计算性,但实验过程中,对该待测程序的最后 1 条路径的覆盖消耗了巨大的资源(约占总计划代数的 30%),在这种情况下需要另一种进化机制来完善进化,下一步准备把本文 2.3 节提到的前件差距  $f_2(X_i)$  引入到进化中,以进一步解决疑难路径的覆盖问题。

第三组实验,以冒泡排序程序进行测试,所有可达路径作为目标路径,设定找到所有目标路径测试数据或者进化代数超过 5000 为终止条件。与采用传统进化方法的遗传算法进行了对比,进行 500 次实验,对每次实验的进化代数与进化时间取平均值,实验结果如表 5 所列。

表 5 冒泡排序程序实验结果

参数个数	种群规模	目标路径数	平均进化代数			总进化时间(ms)		
			本文进化方法	传统进化方法	代数比 (%)	本文进化方法	传统进化方法	时间比 (%)
3	30	6	2	2	100	1	1	100
4	50	24	2	2	100	1	8	13
5	100	120	6	8	75	24	18	133
6	200	720	24	30	80	831	556	149
7	500	5040	80	96	83	51950	29280	177

由表 5 可以发现,本文方法在进化代数上略有进步,但时间效率并不理想,原因是,冒泡排序的待测程序在进化时需要的变异概率较高,随机算法这方面有一定的优势,并且由于随机算法设计简单,运行效率极快,因此在时间上本文算法不占优势。对比表 3 中的数据可以发现,本文方法在指导定向进

化的效率上十分优秀,具有很强的进化指导性,但处理需要更多随机因素的进化时,往往有所改进,但进步不大,同样需要引进新的机制来处理此种情况,比如动态变异概率的参数设计。

**结束语** 本文将复杂系统遗传算法的思想融入基于路径覆盖的测试用例生成中,并通过实验验证,该方法切实可行,且有助于提高解决该问题的效率。

应用本文方法已开发出一个完全实现的实验工具,但工具还有如下几个问题:自动插装一定情况下提供信息不完整,需手工处理;程序包含循环结构的处理策略会耗费大量实验资源;GA 中种群个体的编码方式单一,处理非数值型变量有困难;由于算法实现复杂,在处理时间上需进一步优化等。以上都是下一步需要解决的问题。

## 参考文献

- [1] Genetic algorithm [G/OL]. Wikipedia, 6 June 2011. [http://en.wikipedia.org/wiki/Genetic\\_algorithm](http://en.wikipedia.org/wiki/Genetic_algorithm), 2011-06-06
- [2] Patel R, Raghuvanshi M M, Jaiswal A N. Modifying Genetic Algorithm with Species and Sexual Selection by using K-means Algorithm[C]// IEEE International Advance Computing Conference 2009. Patiala, India, March 2009
- [3] Rajappa V, Biradar A, Panda S. Efficient Software Test Case Generation Using Genetic Algorithm Based Graph Theory[C]// First International Conference on Emerging Trends in Engineering and Technology. 2008
- [4] Tan X B, Cheng Long-xin, Xu Xiu-mei. Test Data Generation Using Annealing Immune Genetic Algorithm[C]// 2009 Fifth International Joint Conference on INC, IMS and IDC
- [5] Ahmed M A, Hermadi I. GA-based multiple paths test data generator[J]. Computer & Operations Research, 2008, 35(10): 3107-3124
- [6] 庄健,杨清宇,杜海峰,等. 一种高效的复杂系统遗传算法[J]. 软件学报,2010,21(11):2790-2801
- [7] Albert R, Jeoung H, Barabasi A L. Diameter of the world wide Web[J]. Nature, 1999, 401(9): 130-131