

# 基于智能算法的软件测试过程质量控制管理研究

陈志坚, 陈星

(天翼视联科技股份有限公司)

【摘要】软件测试过程质量控制管理是提高软件质量和可靠性的关键。文章综述了利用智能算法辅助软件测试过程质量控制管理的研究, 分析了智能算法的应用场景和优势, 介绍了几种智能算法及其在测试用例生成、优化、评估和缺陷预测等方面的应用案例, 并对未来的趋势和挑战进行了展望。

【关键词】软件测试; 过程质量控制管理; 智能算法

中图分类号: TP311.53

DOI: 10.13655/j.cnki.ibci.2025.S2.041

## Research on Quality Control Management of Software Testing Process Based on Intelligent Algorithms

CHEN Zhi-jian, CHEN Xing

(Tianyi Shilian Technology Co., Ltd.)

【Abstract】Quality control management of software testing process is the key to improving software quality and reliability. This article reviews the research on the quality control management in the software testing process supported by using intelligent algorithms, analyzes the application scenarios and advantages of intelligent algorithms, introduces several intelligent algorithms and their application cases in test case generation, optimization, evaluation and defect prediction, and looks forward to future trends and challenges.

【Keywords】software testing; process quality control management; intelligent algorithm

### 1 引言

软件测试是软件开发的重要环节, 软件测试过程的质量控制管理是保证软件测试质量和效果的关键<sup>[1]</sup>。传统方法已无法满足软件系统及软件测试过程的复杂性和多样性需求, 因此使用智能算法来辅助软件测试过程质量控制管理是有意义和价值的<sup>[2]</sup>。智能算法能模拟人类智能, 处理复杂、不确定、动态的问题, 提供优化、近似、推理、分类等功能<sup>[3]</sup>。

### 2 常用的智能算法及其应用案例

#### 2.1 遗传算法

遗传算法是一种模拟生物进化的智能算法, 通过对候选解进行选择、交叉、变异等操作, 不断产生新解, 并根据适应度函数评估解优劣, 逐渐逼近最优解或近似最优解。遗传算法基本流程如图 1 所示。

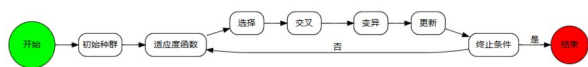


图1 遗传算法的基本流程

遗传算法在软件测试过程质量控制管理中的一项应用是测试用例生成。测试用例生成指根据软件需求和规格, 设计和构造一组可检测软件缺陷的输入输出数据及相应执行和验证步骤。其目标是找到一组能够最大化测试覆盖率和有效性的测试用例, 即覆盖并发现软件尽可能多的层面和缺陷。测试用例生成可以看作是一个优化问题, 即在给定条件与约束下, 寻找最优测试用例集合<sup>[4]</sup>。遗传算法可利用其自适应、自学习、自组织的特性, 对测试用例进行搜索和优化, 从而提高测试用例生成的效率和质量。

遗传算法在测试用例生成中的应用主要包括编码、初始种群构建、适应度函数设计、选择、交叉、变异、种群更新和终止条件判断等步骤。其中一个应用案例是基于遗传算法的测试用例生成系统。该系统可依据软件需求和规格, 自动地生成高质量、高效率的测试用例, 其特点和优势如下。

GATCGS 采用基于图的测试用例编码方式, 将测试用例表示为一个由节点和边组成的有向图, 其中节点表示软件的状态, 边表示状态的转移, 图表示软件的执行路径。这种编码方式可以有效地反映软

件的结构特征和测试覆盖率的度量。在适应度函数设计上,系统采用了一种基于覆盖率和有效性的测试用例,将测试用例的覆盖率和有效性作为优劣的评价标准。这种适应度函数可以有效地反映测试用例的质量和效果,以及测试资源和约束的影响。

在遗传操作环节,GATCGS采用了一种基于图的测试用例交叉和变异操作,将测试用例的图结构作为交叉和变异的对象,对测试用例的节点和边进行交换和修改,从而产生新的测试用例。这种交叉和变异操作可以有效地保证测试用例的搜索空间的扩展和探索,以及测试用例的多样性和创新性。在种群更新方面,系统采用了一种基于精英策略的测试用例更新方式,将测试用例的适应度作为更新的依据,从当前种群和新产生的个体中选择优秀的测试用例,作为下一代种群,同时保留最优的测试用例,作为精英个体。这种更新方式可以有效地保证测试用例的种群的更新和进化,以及测试用例的优良个体的保留和积累。GATCGS在实际的测试项目中进行了应用和验证,结果表明,GATCGS能够自动地生成高质量和高效果的测试用例,提高测试的覆盖率和有效性,节省测试的时间和成本,提高测试的效率和质量。GATCGS的系统结构和运行流程如图2所示。



图2 GATCGS的系统结构和运行流程

GATCGS已在实际测试项目中开展应用和验证,例如在某电子商务系统测试中,其生成的测试用例覆盖率达到95%,而传统方法只能达到80%;同时,GATCGS生成的测试用例可发现系统中的90%的缺陷,而传统的方法只能发现60%。这些结果表明,GATCGS能够自动地生成高质量、高效果的测试用例,提高测试的覆盖率和有效性,节省测试的时间和成本,提高测试的效率和质量。

## 2.2 粒子群优化算法

粒子群优化算法是一种模拟鸟群或鱼群的协同行为的智能算法,通过对一组候选解进行速度和位置的更新,使得解能够在搜索空间中“飞行”或“游动”,不断追随自身最优解与全局最优解,从而逐渐逼近最优解或近似最优解。粒子群优化算法的基本流程如图3所示。



图3 粒子群优化算法的基本流程

粒子群优化算法在软件测试过程质量控制管理中的一面应用是软件测试用例优化。软件测试用例优化是指对已有测试用例进行分析处理,以减少冗余和无效,提高执行效率和效果,节省测试的时间和成本。软件测试用例优化的目标是找到一组“最小化测试用例数量、最大化覆盖率和有效性”的测试用例例子集,即能够用最少的测试用例覆盖和发现软件的尽可能多的层面和缺陷。软件测试用例优化可以看作是一个多目标优化问题,即在给定的条件和约束下,寻找最优的测试用例例子集合<sup>[5]</sup>。粒子群优化算法可以利用其协同、自适应、自学习的特性,对测试用例进行搜索和优化,从而提高测试用例优化的效率和质量。

粒子群优化算法在软件测试用例优化中的应用主要包括编码、初始种群构建、速度和位置更新、终止条件判断等步骤。其中一个典型安全是是基于粒子群优化算法的软件测试用例优化系统(PSOTCOS),该系统可根据已有测试用例集合,自动地生成高质量、高效果的测试用例例子集合,PSOTCOS的特点和优势如下。

PSOTCOS采用了一种基于二进制串的测试用例编码方式,将测试用例例子集合表示为由0和1组成的二进制串,其中0表示该测试用例不被选中,1表示该测试用例被选中。这种编码方式可以有效地反映测试用例和测试目标的特征和约束。在目标函数设计上,系统采用了一种基于多目标优化模型,将测试用例的数量、覆盖率、有效性作为测试用例的优化目标,分别定义为数量目标函数、覆盖率目标函数、有效性目标函数,然后将它们综合为一个综合目标函数。这种目标函数可以有效地反映测试用例的多目标优化问题,以及测试资源和约束的影响。在算法实现方面,PSOTCOS采用基于惯性权重的粒子群优化策略,将惯性权重设为变量,根据迭代过程动态调整惯性权重的值,从而平衡搜索空间的探索和利用,以及收敛性和稳定性;同时引入多种群协同机制,将种群划分为多个子种群,每个子种群采用不同的参数和策略进行更新,同时进行子种群之间的信息交换和合作,从而提高搜索空间的扩展和搜索效

率。PSOTCOS 已在实际测试项目中进行了应用和验证,例如在某医疗信息系统的测试中,PSOTCOS 能够优化原有的测试用例集合:测试用例的数量减少 50%,而测试用例的覆盖率和有效性分别提高了 10% 和 15%;同时,PSOTCOS 优化的测试用例能够发现系统中的 95% 的缺陷,而原有测试用例只能发现 80% 的缺陷<sup>[6]</sup>。这些结果表明,PSOTCOS 能够自动地生成高质量、高效果的测试用例集合,减少测试用例的数量的同时增加覆盖率和有效性,节省测试的时间和成本,提高测试的效率和质量。PSOTCOS 的系统结构和运行流程如图 4 所示。



图4 PSOTCOS的系统结构和运行流程

### 3 基于智能算法的软件测试过程质量控制管理的未来发展趋势和挑战

#### 3.1 未来发展趋势

智能算法的多样化和融合方面,需借助人工智能技术的创新,提供更多的智能算法,如模拟退火算法、蚁群算法、人工免疫算法、差分进化算法、深度学习算法等,针对不同的软件测试问题,提供不同的解决方案<sup>[6]</sup>。同时,组合与融合不同的智能算法,形成更强大和更灵活的混合智能算法,提高软件测试的效果和性能。软件测试过程质量控制管理需向全面化和智能化发展,涵盖测试计划、测试设计、测试执行、测试评估、测试报告等软件测试过程的各个方面和环节,实现软件测试过程的自动化,提高效率和质量。此外,还需推进个性化与定制化建设,根据不同的软件系统、软件测试项目、软件测试人员及软件测试环境等,提供个性化和定制化的解决方案,实现软件测试过程的自适应的调整和优化,提高软件测试过程的适应性和灵活性。

#### 3.2 挑战

一是智能算法理论和方法的完善和创新,需提高智能算法的可靠性和有效性,重点解决收敛性、稳定性、参数设置、策略优化、效果评估等问题。二是软件测试过程质量控制管理对复杂性和多样性的处理能力提升,需提高软件测试过程质量控制管理的可行性和可解性,实现对高维、非线性、不确定、动态

等特征问题的有效地建模和分析。三是软件测试过程质量控制管理的实际应用和推广,需提高技术的普及性和实用性,有效地推动和促进软件测试人员的认知和接受,软件测试环境的兼容和支持,促进软件测试标准和规范的制定和遵守。

## 4 结语

本文综述了基于智能算法的软件测试过程质量控制管理的相关研究,分析了智能算法在软件测试过程质量控制管理中的应用场景和优势,介绍了遗传算法、粒子群优化算法等常用的智能算法,及其在软件测试用例生成、软件测试用例优化、软件测试覆盖率评估、软件测试缺陷预测等方面的具体应用案例,并对基于智能算法的软件测试过程质量控制管理的未来发展趋势和挑战进行了展望。基于智能算法的软件测试过程质量控制管理是一个新兴的研究领域,具有广阔的应用前景和巨大的发展潜力,值得进一步研究和探索。

#### 参考文献

- [1] 杨玉, 王博, 徐震阳, 等. 软件质量智能分析平台的构建研究[J]. 现代信息科技, 2023, 7 (6): 120-123.
- [2] 王敏, 马敏, 张志. 基本路径测试用例设计的改进方法[J]. 计算机应用与软件, 2019 (10): 17-22.
- [3] 杨春静, 杨达, 葛长赞. 基于智能算法的软件测试方法研究[J]. 软件, 2023, 44 (5): 32-34.
- [4] 周学花. 基于智能优化算法的软件测试数据生成模型及其应用研究[J]. 北华航天工业学院学报, 2023, 33 (3): 10-12.
- [5] 党向盈, 李金凤. 融合智能优化的软件测试用例生成方法[J]. 软件工程, 2022, 25 (12): 59-62.
- [6] 曲延羽, 邓琬云, 熊铎程, 等. 基于VS的智能驾驶软件测试系统的设计与实现[J]. 汽车与驾驶维修(汽车版), 2021 (8): 36-37+40.