

基于运行剖面的列控系统测试用例生成研究

古 俐,徐中伟,梅 萌

(同济大学 电子与信息工程学院,上海 200000)

摘 要:为了提高列控系统(列车运行控制系统,简称列控,是保证列车安全、快速运行的系统)测试的可靠性,生成合理的测试用例,需要设计相应的运行剖面来仿真实际使用的列控系统的情况,并且基于得到的运行剖面来生成相应的测试用例。因此,通过对列控中心系统的输入输出特点以及功能需求的研究与总结,提出了通过构建任务剖面、系统模式剖面、环境剖面来最终完成列控系统运行剖面构建的方法,并在最终得到的运行剖面基础上,应用优先级度量机制和程序插桩方法对粒子群算法进行合理改进,最终实现了测试用例的自动生成。实验结果证明,通过合理地构建运行剖面,可以保证测试功能的完整性以及测试数据的合理性,改进后的粒子群算法能够进一步提高测试的时间效率。

关键词:列控系统;运行剖面;粒子群算法;测试用例生成;适应度函数

中图分类号:TP399

文献标识码:A

文章编号:1673-629X(2020)05-0001-05

doi:10.3969/j.issn.1673-629X.2020.05.001

Research on Test Case Generation of Train Control System Based on Operation Profile

GU Li, XU Zhong-wei, MEI Meng

(School of Electronics and Information Engineering, Tongji University, Shanghai 200000, China)

Abstract: In order to improve the reliability of the train control system (train operation control system, which is to ensure the safe and fast operation of the train) and generate reasonable test cases, it is necessary to design the corresponding operational profile to simulate the actual use of the train control system, and the corresponding test case is generated based on the obtained running profile. Through the research and summary of the input and output characteristics and functional requirements of the column control center system, the method of constructing the running profile of the train control system by constructing the task profile, system mode profile and environmental profile is proposed, and the final operational profile is obtained. On the basis of the operational profile, the priority measurement mechanism and the program instrumentation method are used to improve the particle swarm optimization algorithm, and finally the automatic generation of test cases is realized. The experiment shows that the integrity of the test function and the rationality of the test data can be ensured by constructing the running profile reasonably. The improved particle swarm optimization algorithm can further improve the time efficiency of the test.

Key words: train control system; running profile; particle swarm optimization; test case generation; fitness function

0 引言

近年来,中国铁路建设进入了蓬勃发展的新时期,目前已经全面进入高铁时代,这对于国家构建综合交通体系,提高铁路运输效率起到了巨大的作用。但高铁系统庞大复杂,一旦防控环节处理不当,可能对高铁安全运行产生巨大的影响,因此,保证铁路网的安全运行成为迫在眉睫的问题。而列控中心作为地面车站的控制中心,其性能决定着整个铁路线的安全,对其功能

进行有效测试,是保证其安全运行的方法^[1]。

列控中心测试场景较为复杂,测试案例涵盖的内容也相当广泛,现有的测试用例的编写运行主要靠人力手工作业,耗费了大量的时间且易出现错误,因此根据列控中心平台测试需求,高效合理地自动构造和生成测试用例是改善测试效果,降低测试成本的有效方法。

文中重点关注列控系统运行剖面的构建,并考虑

收稿日期:2019-05-30

修回日期:2019-09-30

网络出版时间:2019-12-18

基金项目:国家自然科学基金委高铁联合基金(U1734211)

作者简介:古 俐(1996-),女,硕士,研究方向为基于通信的列车控制系统;徐中伟,博士,教授,研究方向为基于通信的列车控制系统、安全苛求系统,可信计算、软件构件化开发及应用等。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20191218.1115.064.html>

将粒子群算法与构建的运行剖面相结合。通过合理构建列控系统的可靠性测试数据生成模型,能够将生成测试数据所需信息通过软件的任务、系统模式、环境剖面来建模实现,在所得到的运行剖面的基础上,应用粒子群算法来提高自动生成测试数据的效率与质量。

1 运行剖面 and 测试用例生成技术简述

1.1 运行剖面技术

运行剖面的构建是软件可靠性测试的重点,良好的运行剖面能够正确指导软件的测试,帮助生成合理可靠的测试用例。运行剖面指的是对系统输入值在其可能输入范围内出现概率分布的定义,它描述了软件实际使用的情况,构建时需要充分分析被测软件的需求、功能以及输入输出,从而生成可靠的运行剖面,得到可信的测试数据^[2]。

运行剖面的基本概念最早出现在文献[3]中,John Musa 提出了构造运行剖面的基本方法以及基于运行剖面的测试数据生成方法。基本的运行剖面生成方法由五步构成:找出客户剖面、建立用户剖面、定义系统模式剖面、确定功能剖面、完成运行剖面。

Musa 对软件分析的原则适用于多种软件,但需要结合实际系统情况进行改进。在国内,很多领域都结合并改进了运行剖面来提高软件测试的可靠性。例如,文献[4]提出了将复杂软件的剖面构建过程拆分为软件运行剖面与若干模块运行剖面的剖面构建方法,减少了局部的复杂度;文献[5]提出了适合于航天实时软件使用的剖面构造方法,并在建模的基础上结合遍历算法实现了测试数据的生成;文献[6]采用基于 UML 图和定性分析的混合方法,建立了军用软件运行剖面元模型,为军用软件可靠性实例化提供了支撑。

1.2 测试用例生成技术

以往的测试用例生成主要靠人力手工完成,这对测试人员的专业水平提出了一定的要求,容易导致实际工程测试中测试用例生成的盲目性。并且根据以往统计,设计测试用例阶段的开销占软件测试全部开销近一半的比重,因此,研究测试用例生成算法,使得程序能够高效合理地自动构建测试用例有十分重要的意义^[7]。

近年来,很多学者对测试用例自动生成这一领域做了大量研究,研究表明,在多种测试用例生成算法中,启发性算法具有良好的方向性、能够动态反馈信息、鲁棒性强等优点。启发性自动用例生成算法主要有四类:随机测试、遗传算法、蚁群算法、粒子群算法,其算法框架相似,都是先对被测程序进行结构性分析得到生成测试目标,对被测程序进行插桩进而来构造

适应度函数,最后根据适应度函数选择修正得到的测试用例,大部分研究都是在基本算法框架上进行改进,来提高生成测试用例的效率^[8]。文献[9]将迷宫算法与蚁群算法结合来优化 RBC 平台测试用例的序列集,实现了优化自动生成的测试用例,缩短测试时间的目的;文献[10]设计了遗传算法与粒子群算法的混合优化算法,将改进的粒子群算法作为遗传算法的重要算子来实现最优解的快速搜索;文献[11]提出了基于正交搜索的粒子群优化测试用例生成方法,通过利用奇异值分解,改进粒子速度项来提高覆盖率,减少运行时间。

2 整体技术方案

文中通过分析列控系统的输入特点,对操作剖面生成方法进行了改进,使得输入数据反映实际输入的情况,得到合理列控系统的运行剖面。在所得运行剖面的基础上,结合启发性算法来改进测试数据生成技术,提高测试用例生成效率。

2.1 运行剖面技术

运行剖面的建立是进行实时软件可靠性测试的重点,建立运行剖面时需符合系统的实际使用情况。列控系统具有用户单一、输入输出复杂、时间空间相关性的特点,基于以上的特点提出了改进的运行剖面构造流程,并根据列控系统分析测试需求来构建任务剖面、系统模式剖面以及环境剖面,最终完成运行剖面的构造^[12]。

2.1.1 任务剖面构造

任务剖面是由根据列控系统功能总结出的具有代表性的功能特征点组成的,在获得功能点的基础上进行进一步的模式划分可以得到系统模式剖面。获取功能特征点前需要明确所属功能实体,对于列控系统,它的功能实体分为七大类:轨道电路、区间信号机点灯、车站联锁、车地通信、临时限速、邻站列控中心、异物侵限,这七类为列控系统最高层功能特征,在此基础上进一步的划分直至功能特征点不可再分即得到最后具有代表性的功能特征点列表。列控系统关键功能特征点划分如图 1 所示,设 $M = \{m_i | i = 1, 2, \dots, 7\}$ 为系统功能状态合集。

2.1.2 系统模式剖面构造

系统模式剖面的构造基于所得的系统功能状态合集 M ,需要选取功能特征点进行模式划分,划分时需确定模式参数 p_i 。模式参数决定了系统的功能状态模式及其出现的概率,如临时限速这一功能实体中包含临时限速命令下达这一功能点^[13],其系统模式分为执行失败、执行成功、紧急制动三种功能模式,根据功能测试要求确定系统模式概率取值,进而得到系统模

式剖面 $M_p = \{ (m_i, p_j) \mid i = 1, 2, \dots, 7, j = 1, 2, \dots, n \}$, 对于单一系统模式的功能点取 $p_i = 1$ 。

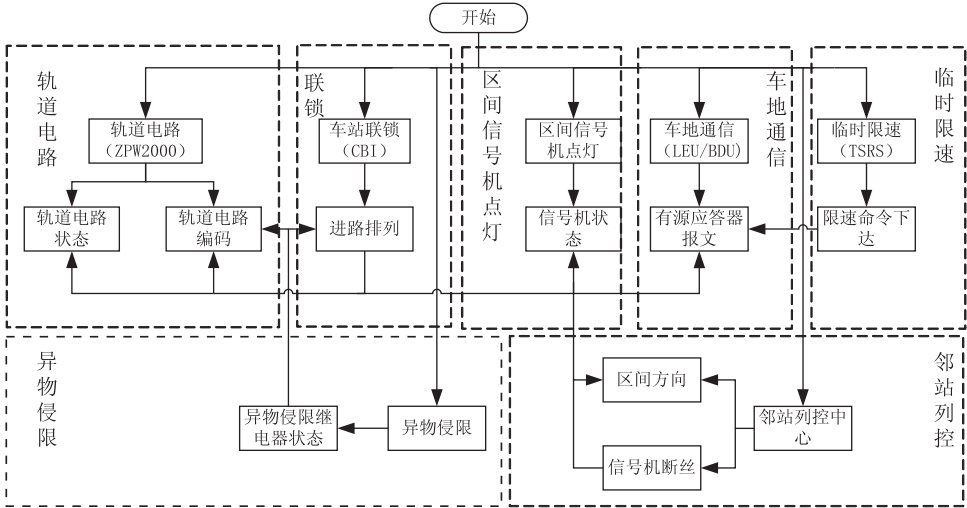


图1 任务剖面构造过程流程

2.1.3 环境剖面构造

环境剖面基于系统模式剖面 M_p ,由若干操作运行值 $E = \{ e_i \mid i = 1, 2, \dots, n \}$ 构成,这些运行值会影响系统输出,但不影响系统模式的变化。在列控系统中,某些输入参数间存在约束关系,如一个操作常基于另一个操作进行,所以在运行值间需添加参数间的约束条件。并且列控系统属于实时软件,输入具有序列性,需对每个子剖面的功能运行值进行时序描述形成功能运行值集 $CombSet = \{ X[e_i(t)] \mid i = 1, 2, \dots, n, t \in [0, T] \}$, $X[e_i(t)]$ 为在某种功能模式下,操作运行值的变化过程^[14]。

2.2 改进的粒子群算法

在运行剖面构建完成后,需要选取合适的搜索算法来生成可靠性测试数据。文中选用的测试用例搜索算法是粒子群算法,相比其他搜索算法,该算法具有较强的鲁棒性和全局搜索能力,便于有效地生成测试用例。在传统的粒子群算法的基础上,根据之前所构造的列控中心运行剖面来改进算法步骤,选取合适的参数,最终达到优化测试序列,缩短测试时间的目的。设计的粒子群测试用例生成算法流程如图2所示,具体执行流程如下:

2.2.1 输入空间建模

根据已构建好的列控中心运行剖面可得到符合覆盖强度和需求规范的功能运行值集 $CombSet = \{ X[e_i(t)] \mid i = 1, 2, \dots, n, t \in [0, T] \}$,即根据任务剖面获得总结分解后的功能特征点,在获得的功能特征点的基础上根据系统模式剖面确定每个功能特征点所具有的模式参数,进而根据环境剖面获得在每种模式下的功能运行值及相关约束条件,完成功能运行值集。最后获得的功能运行值集在构造过程中严格按照上文所提出的列控中心运行剖面构造步骤,覆盖了测试大

纲要求的必需的功能点,满足了覆盖强度,同时预先剔除了无效的功能特征点,有效缩减了功能运行值集的大小,为之后的测试用例生成减少了计算量,并且避免因因此引起的适应度值误差。

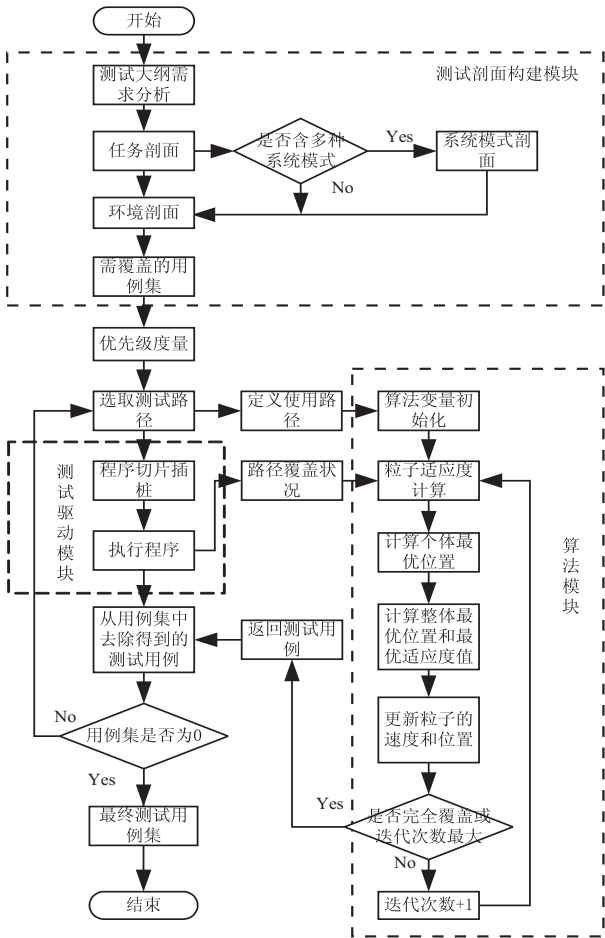


图2 粒子群测试用例生成算法流程

2.2.2 优先级度量

将得到的功能运行值集作为测试用例生成的输入条件,之后引入了优先级度量机制,即先选择高优先权

的测试任务来执行。主要思想是每次划分割面时,根据测试大纲各部分所占条例数分配概率,最后得到每个功能运行值对应的操作序列概率,概率越大,优先权越高^[15]。优先权的引入可以减少有价值任务的等待时间,尽量避免因测试时间不足导致重要功能无法测试到的可能,提高了生成数据的可靠性。

2.2.3 测试用例生成

步骤 1:种群粒子初始化,根据优先级度量选定要执行的测试路径序列,结合得到的功能运行值集确定种群规模 m ,即粒子群向量 $X = \{X_i \mid i = 1, 2, \cdots, m\}$,粒子维数 n ,迭代次数 t ,惯性权重 w ,学习因子 c_1, c_2 以及粒子群的位置 $X_i = \{X_{il} \mid l = 1, 2, \cdots, n\}$ 和速度 $V_i = \{V_{il} \mid l = 1, 2, \cdots, n\}$ 。其中粒子群的位置通过均匀随机分布从功能运行值集中选取,粒子自身最优位置 $P_i = \{P_{il} \mid l = 1, 2, \cdots, n\}$ 为对应的 X_i 。

步骤 2:适应度函数设计,设计目标是使粒子所走路径尽可能贴近目标测试路径,通过在程序中插入适应度函数,来判断路径是否被覆盖,否则进入新一轮的迭代^[10]。考虑到所构造的列控系统的特点,在构造适应度函数时采用了分支距离法,即当选取的测试路径序列中存在 m 个分支点, n 个参数时,设计分支函数 $f_k = \{f_k(x_1, x_2, \cdots, x_n) \mid k = 1, 2, \cdots, m\}$ 插入到路径的每个分支前, f_k 的值取决于分支条件是否满足,若满足条件取值为 0,否则取值定为偏离值大小。

最后得到的适应度函数如下:

$$F = \sum_{k=1}^n f_k$$

通过适应度函数可得到每个粒子的适应值,进而来衡量测试用例的优劣。

步骤 3:将得到的适应度值与粒子自身最优位置 P_i 、种群最优位置 G_i 相比较,若适应度函数值更小代表所走路径更贴合最优路径,则更新 P_i 、 G_i 。

步骤 4:更新粒子的速度和位置。

速度的更新公式如下:

$$v_{id}^{t+1} = v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (g_{id}^t - x_{id}^t)$$

位置的更新公式如下:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}$$

其中, $d = 1, 2, \cdots, n$ 表示粒子所处空间维数, $i = 1, 2, \cdots, m$ 表示种群中某个粒子。

速度函数表明在选择用例时,是在上一次用例选择的基础上结合了粒子本身经验用例选择和全局最优用例,能够很好地平衡全局和局部的搜索能力,进而得到合适的用例集。

步骤 5:判断粒子是否达到步骤 1 中所设置的最大迭代次数 t 或是找到最优解,若满足条件即输出,不满足的话跳转到步骤 2 继续迭代,直至达到结束条件,

返回最优的单个测试用例。根据得到的测试用例 t 更新 CombSet,将其中的 t 剔除,跳转到步骤 1 重复上述步骤,直至 CombSet 中所有的测试用例被覆盖,输出最终的测试用例集。

3 性能评估

3.1 测试数据合理性

现有的列控系统测试用例一般是由人力编写,经过多次审核校验,确定各部分功能点测试比例以及测试内容,进而完成最后的测试大纲。但人力编写具有耗费时间,易出错的缺点,通过自动化生成测试用例,能够在满足测试功能完整、测试数据合理的情况下,提高测试的时间效率。

文中提出的运行剖面在构造过程中的各部分取值概率参考了往年列控测试大纲中实际运行的功能条例数,使得根据剖面生成的数据从统计意义上更加接近于真实的使用情况。假设最终要生成的测试用例总数 $N = 1\ 000$,将生成的测试用例按照功能归类,最终根据上述算法所得到的测试用例抽取情况如表 1 所示。

表 1 操作序列的抽取结果

	ZPW 2000	区间 点灯	CBI	LEU	TSRS	邻站 TCC
理论概率	0.2	0.05	0.2	0.05	0.3	0.2
抽中组数	179	48	216	57	294	206
占总数比例	0.179	0.048	0.216	0.057	0.294	0.206

表 1 数据显示,所抽取到的功能条例与预期抽取功能概率相差不大,抽样比较稳定,数据符合预期结果,各部分需要测试的功能比例分配与列控测试大纲中的比例相差不大,进而保证了生成测试数据的可靠性与合理性。

3.2 算法改进的优越性

选取了测试大纲中轨道电路状态更改的用例,针对上述提出的基于粒子群的运行剖面用例生成方法用 Python 语言在 PyCharm 平台进行仿真,并将其与传统的深度优先搜索遍历方式进行比较。选取用例的路径总节点数为 8,变量个数为 3,假设输入数据范围为 0 到 30,所取间隔为 2,粒子群算法中的搜索空间维度 $n = 3$,学习因子 $c_1 = c_2 = 2$,惯性权重 $w = 1.1$,种群规模 $m = 10$,迭代次数分别取 $t = 1、5、10$ 。

仿真结果如图 3 所示,其中虚线为在不同输入数据下粒子群算法所花费时间,实线为在不同输入数据下深度优先搜索遍历算法所花费时间。且在不同输入数据下粒子群算法的分支覆盖率均为 100%。从仿真结果中可以看出,在输入数据范围较小时,深度优先搜索遍历方法的优势明显,而当 r 值变大,深度优先搜索

遍历方法的优势则逐渐减小;同时,可以看出,在满足分支覆盖率的前提下,尽量减少粒子群算法的迭代数可以减少用例生成时间。因此对于输入数据范围较大的列控系统而言,文中提出的基于粒子群的运行剖面用例生成方法能在满足分支覆盖率的前提下,有效地减少用例生成时间并进行用例约减。

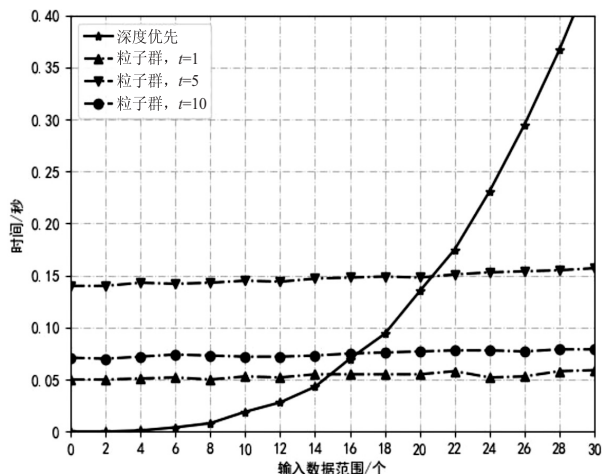


图3 运行剖面用例生成方法方针对比

4 结束语

为了提高列控软件测试的效率,改善人力测试的不足,分析并设计了列控系统运行剖面的构建过程、基于粒子群算法的运行剖面用例生成方法。提出的列控系统运行剖面结合了列控系统功能特征,通过构造合理的运行剖面来生成合适比例的测试用例,满足了测试的功能需求。在生成测试用例时,采用了粒子群算法,与构造好的剖面相结合来生成测试用例。针对上述提出的基于粒子群的运行剖面用例生成方法,选取了轨道电路状态更改的用例来进行仿真,并将其与传统的深度优先搜索遍历方式进行比较,证明该方法能够根据需求生成合适功能比例的用例,减少冗余的测试用例生成,并且在满足用例覆盖率的前提下,减少用例的生成时间。

参考文献:

[1] 刘 强. CTCS-3 级列控系统功能测试方法分析[J]. 中国安全科学学报, 2018, 28(S2): 79-83.

[2] 王令赛,姜淑娟,张艳梅,等. 基于正交搜索的粒子群优化测试用例生成方法[J]. 电子学报, 2014, 42(12): 2345-2351.

[3] MUSA J D. Operational profiles in software reliability engineering[J]. IEEE Software, 1993, 10(2): 14-32.

[4] 李 奥. 软件可靠性测试操作剖面构建方法研究及其应用[D]. 北京: 北京邮电大学, 2017.

[5] 张 旭. 基于使用剖面的航天软件可靠性测试数据生成方法的研究[D]. 北京: 中国航天第二研究院, 2009.

[6] 孙 媛,唐金国,徐吉辉. 基于 UML 的军用软件运行剖面元模型研究[J]. 计算机与数字工程, 2018, 46(2): 337-342.

[7] 杨 杰,王兴起. 改进的可靠性测试用例自动生成方法[J]. 杭州电子科技大学学报: 自然科学版, 2018, 38(4): 41-46.

[8] 聂 鹏,耿 技,秦志光. 软件测试用例自动生成算法综述[J]. 计算机应用研究, 2012, 29(2): 401-405.

[9] HU Naiwen, ZHENG Wei. The improved ant colony algorithm test sequence optimization based on the RBC test platform[C]//2014 IEEE 17th international conference on intelligent transportation systems (ITSC). Qingdao: IEEE, 2014: 8-11.

[10] 史娇娇. 基于遗传粒子群优化算法的测试数据自动生成技术研究[D]. 徐州: 中国矿业大学, 2014.

[11] 陈云飞,李 征,赵瑞莲. 基于 PSO 的多目标测试用例预优化[J]. 计算机科学, 2014, 41(5): 72-77.

[12] ZHAO Y. The optimization method of Markov chain usage model based on three-parameter interval number[C]//IEEE international conference on grey system & intelligent services. Leicester, UK: IEEE, 2015.

[13] WANG Y, YE F, ZHU X, et al. A method for software reliability test case design based on Markov chain usage model [C]//International conference on quality, reliability, risk, maintenance, and safety engineering. Chengdu: IEEE, 2013: 1207-1210.

[14] GITTENS M, LUTFIYYA H L, BAUER M A. An extended operational profile model [C]//International symposium on software reliability engineering. Saint-Malo: IEEE, 2005.

[15] 赵显琼,李开成,唐 涛,等. 基于 UML 的 CTCS-3 级列控系统运营场景分析方法研究及应用[J]. 铁道通信信号, 2010, 46(8): 4-8.