文章编号:1671-4598(2022)10-0063-07 DOI:10.16526/j.cnki.11—4762/tp.2022.10.011 中图分类号:TP273+.5 文献标识码:A

面向飞行器发射控制系统可靠性提升的测试分析

张子剑，姜 利，龙中权，林 海，刘 虎
（北京宇航系统工程研究所，北京 100076）

摘要：系统的可靠性与测试性密切相关，良好的测试性设计，可以提升系统的可靠性，一个可靠的系统不是通过最后的测试结果测试出来的，而是在设计中将测试性考虑其中，系统测试性对系统的基本可靠性有着很大的影响；为了找到飞行器发射控制系统最优测试项，对可靠性与测试性二者的关联进行深入研究，通过系统设计初期的结构组成进行测试性分析，建立系统多信号流图模型，研究系统的相关性矩阵，并进行系统的故障检测率、故障隔离率等参数分析；提出了基于测试覆盖度的遗传算法进行测试项的优化选择，通过此方法进行实验仿真找到系统的最优测试项，成功满足系统的故障检测率、故障隔离率要求，将改进的遗传算法和传统的遗传算法进行对比，改进后的算法有着更好的收敛性，对系统分析更好。

关键词：飞行器发射控制；系统可靠性；故障树分析；遗传算法

Test and Analysis of Reliability Improvement in Aircraft Launch Control System

ZHANG Zijian, JIANG Li, LONG Zhongquan, LIN Hai, LIU Hu
(Beijing Institute of Aerospace Systems Engineering, Beijing 100076, China)

Abstract: The reliability of system is closely related to testability. A good testability design can improve the reliability of system. A reliable system is not tested by the final test results, but testability is considered in the design, which has a great influence on the basic reliability of the system. In order to find the optimal vehicle emission control system test, the association of the reliability and testability is deeply studied, through the system design of the early structure testability analysis, the system signal is established by the flow graph model, the system of the correlation matrix, and the system parameters such as fault detection rate and fault isolation rate analysis. The optimal selection of test items based on the test coverage of the genetic algorithm is proposed. This method is used to find out the optimal test item by the experimental simulation, the system fault detection rate and fault isolation rate requirement are successfully met, the improved genetic algorithm is compared with the traditional genetic algorithm, the improved algorithm has better convergence and analysis for the system.

Keywords: aircraft launch control; system reliability; fault tree analysis; genetic algorithm

0 引言

对于系统的可靠性进行理论研究时候，需要通过系统测试性分析完成系统的整体测试和检测来提高系统的可靠性，保证系统在发射前有正常状态，从而保证系统的发射成果，同时如果遇到问题，可以快速地发现问题并解决[1]。国内外对于导弹、火箭等武器系统的研究，是需要进行大量弹射实验和演示验证，完成重点成果的输出[2-3]。飞行器进行试验发射前，需要在地面对飞行器各个设备的工作状态进行逐一检查，完成检查后按照规定流程进行发射，地面测试系统的测试结果决定飞行器是否具备发射条件，所以飞行器测试也十分重要[4]。

1 故障模型分析

对于本文研究的飞行器发射控制系统进行故障分析时候，由于系统结构组成复杂，故采取逐层分析的故障分析方法。分析3个子部分之间联系的故障形式，从而分析出系统的可能故障形式，分析其原因和产生的后果，最后针对可能潜在出现的故障采用相应的措施和检测手段，从而保证整个系统的可靠性和安全性。

1.1 故障分析

1.1.1 故障模式

针对飞行器发射控制系统进行研究时候，由于其组成部件数量巨多，故可能会产生很多的故障形式，而且系统完成很多功能，每个功能实现过程中可能会出现很多故障模式，但是，分析故障时候要把握完备性和唯一性两个基本要求，即要将系统故障识别全面并且识别出的故障模式是明确的，彼此之间互相排斥，不存在交叉部分。并且对于系统的故障分析要全面，尽可能将系统的所有可能发生的故障进行分析和整理，进而可以分析出系统的薄弱环节，从而在进行系统的设计的时候，可以对这部分薄弱的环节进行重点设计，提高其可靠度。故障模式获取方法有很多，包括分析、预测、实验、统计等。

当我们进行系统的故障分析的时候，可以根据如下原则进行故障模式分析[5]：

1）对于具有相似的系统组成部分，可以根据已有的相

收稿日期:2022-06-16； 修回日期:2022-07-02。

作者简介:张子剑(1987-),男,江西九江人,硕士,高级工程师,主要从事飞行器电气系统设计方向的研究。

引用格式:张子剑 ,姜 利,龙中权,等.面向飞行器发射控制系统可靠性提升的测试分析[J].计算机测量与控制,2022,30(10):63-69.

似功能单元或者具有相似结构的单元出现的故障，从而进行系统的故障分析。

2）根据系统的统计数据和测试数据不断丰富系统的故障模式。

3）对于常见的元器件、零部件等，可以根据国内外标准手册确定失效模式。

1.1.2 故障原因分析

通过对系统进行故障识别后，为了提高系统可靠度，还需要通过故障原因分析找出系统的每个故障模式产生的原因，进而可以做出针对性的改进，从系统设计阶段就可以消除故障产生的可能性。对于故障原因进行分析的时候可以从两个角度进行分析，即成因机理和成因阶段。成因阶段就是包括设计阶段、制造阶段和使用阶段，从而可以根据阶段设计相应措施进而消减故障，降低系统故障的可能性，从而提高系统的可靠性[6]。

1.1.3 严酷度分析

结合已有的针对武器系统的严酷度定义，本文对飞行器发射控制系统的严酷度进行定义如表 1 所示[7]。

表 1 严酷度定义

严酷度类别	严重程度定义
Ⅰ类(灾难的)	飞行器控制系统误动作造成地面设备、人员严重毁伤
Ⅱ类(致命的)	飞行器可以成功发射，但是其他系统无法配合进行通讯，飞行器失控
Ⅲ类(临界的)	飞行器发射控制系统和其他系统进行联调实验中无法正常进行。
Ⅳ类(轻度的)	飞行器发射控制系统在发射前系统自检出现错误信息

1.2 系统结构功能分析

对于系统来讲，由于要保证人员安全和发射过程中各个系统接收开始动作信号的同步，采用分布式设计，主要分为 3 个子部分，远端发控设备、前端测控设备、飞行器控制设备。由于飞行器发射控制系统中，直流稳压电源、光电转换模块、飞行器控制供电模块等可靠性非常高，并且由于设计重量、费用的要求，故拟在远端发控装置、前端测控装置、时序控制器这三部分进行冗余余度设计。经过理论计算和推导，飞行器发射控制单元采用三冗余结构，前端测控单元采用主备份二冗余设计，远端不采用冗余设计。并且实际飞行器发射控制系统要完成飞行器发射需要严格按照时序完成动作，实际运行过程飞行器控制系统需要实时监测火工品的电压，以及完成地面和飞行器的电压切换。地面工作人员通过远端发控计算机完成对飞行器的实时监测，并根据监测结果，产生相应得动作指令，各个子系统需要完成对应的功能，飞行器发射系统的硬件架构如图 1 所示。

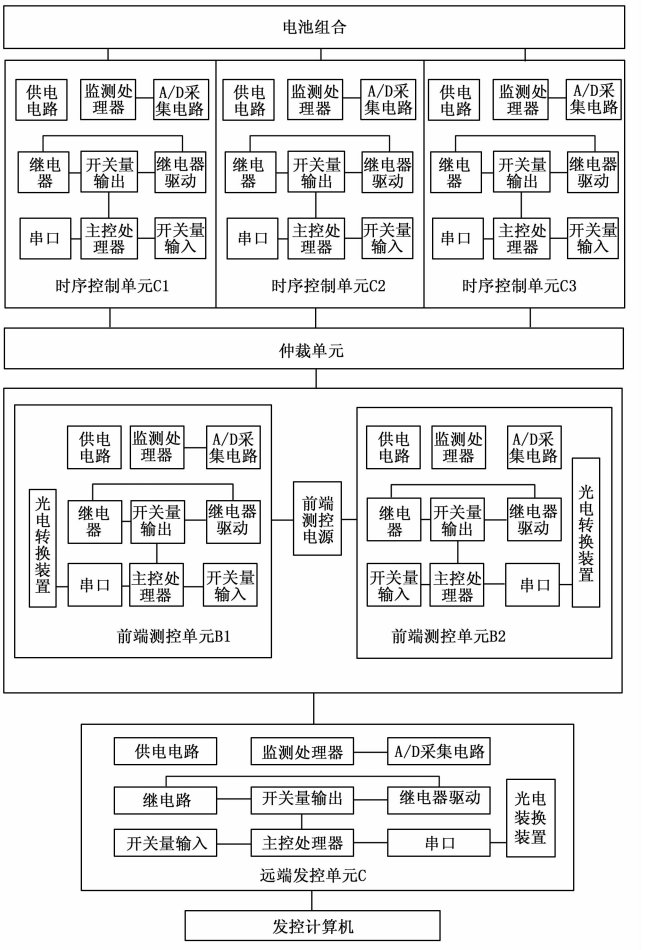

图 1 硬件架构图

由于在设计系统阶段需要考虑如何最大可能地提高系统的可靠性，在设计过程中就将系统的测试考虑其中。考虑系统的测试点、组成单元之间的信号关系，通过合理布置单元之间的关系，布置相关的测试点，分析关键的测试，以便当系统出现故障的时候，可以及时准确地完成故障的检测并隔离，采用相应的解决措施，完成故障的清除，可以更好地完成飞行器发射任务，提高系统的可靠性[8]。

针对初步设计的飞行器发射控制系统硬件架构，进行系统时序测试、火工品的通路测试、火工品电压测试，针对测试需要，本文先以火工品通路测试为例，进行故障树分析，基本事件表如表 2 所示。

表 2 基本事件表

事件代码	事件	事件代码	事件
A	火工品电路继电器卡死	D3	时序控制单元 C3 异常
B	火工品失效	E1	前端测控主单元 B1 异常
C	仲裁单元失效	E2	前端测控从单元 B2 异常
D1	时序控制单元 C1 异常	F	远端发控单元异常
D2	时序控制单元 C2 异常		

火工品电路失效，可能是由于火工品电路继电器不动作，继电器卡死，有可能火工品没有问题。进而分析控制火工品电路的时序控制单元，火工品控制单元的控制是由三冗余的时序控制单元通过仲裁单元输出信号从而完成控

制[9]。因此。需要分析仲裁单元是否故障，当仲裁单元没有故障的时候，进而分析时序控制单元中的火工品继电器驱动电路。本文将模型进行简化，分析到单元级。当判断时序控制单元无故障的时候，进而分析前端测控装置，远端发控装置是否故障[10]。通过此方式递推寻找基本事件。由于对于本文研究的飞行器发射控制系统，火工品电路相对关键，进而对火工品通路测试进行分析的时候。故障树分析结果如图 2 所示。

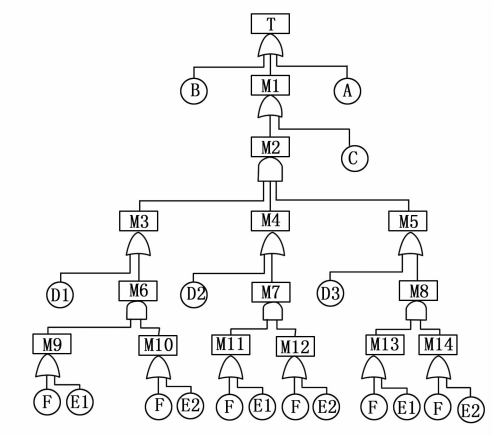

图 2　故障树结果图

2　测试建模分析

2.1　多信号流图模型

本文在研究了飞行器发射控制系统结构和功能的基础上，分析信号和单元之间关系，对于测试关键问题进行 FTA 的研究，从而可以分析出在关键测试中系统各个组成部分的主要故障部位。要对系统进行测试性分析，需要建立系统的测试性分析模型，通过分析系统的各个功能模块的信号和测试之间的关系，从而可以分析整个系统的测试性。系统的多信号流程图模型是目前可以简化的分析系统故障和测试之间关系的方法，该方法在测试性建模中应用广泛，适合用于工程中的系统分析。

2.2　测试性故障建模分析

在上一节对系统火工品通路的故障树模型分析后，结合时序测试、火工品电压等测试，简化对模型的测试性分析，从底事件入手对系统进行系统的多信号流图建模分析，确定多信号流图中的模型的各个组成。由于系统组成繁杂，本文从系统级进行分析，对应部件级分析类似，不再赘述，系统的模型要素如下。

2.2.1　故障源

系统中主要可以分析的故障源如下：c_1：发控计算机；c_2：远端发控单元；c_3：前端测控单元 B_1；c_4：前端测控单元 B_2；c_5：前端切换装置：c_6 时序控制单元 C_1；c_7：时序控制单元 C_2；c_8：时序控制单元 C_3；c_9：仲裁单元。

2.2.2　故障测点和相关信号

根据火工品测试、时序测试等关键测试和其相关信号，可以进行相关测点设置和测试信号分析。故障测点和信号如表 3 所示。

表 3　飞行器控制系统故障源和测试点总结

故障源	故障模式	测试信号	测试点
c_1	发控计算机失效	输出控制指令	TP_1
c_2	开关量输入、输出模块故障	牵动、时统信号	TP_2
c_3	开关量输入、输出模故障	时统、牵动、弹射信号	TP_3
c_4	开关量输入、输出模块故障	时统、牵动信号	TP_4
	电压回采电路故障	火工品电压	
c_5	切换装置故障	开关量信号	TP_5
c_6	开关量输入、输出模块故障	时统、牵动信号	TP_6
	电压回采电路故障	火工品电压	
c_7	开关量输入模故障	时统、牵动信号	TP_7
	电压回采电路故障	火工品电压	
c_8	开关量输入、输出模块故障	时统、牵动、弹射等信号	TP_8
	电压回采电路故障	火工品电压	
c_9	仲裁单元故障	开关量信号	TP_9

2.2.3　故障源、测试点、测试、和信号之间的关系

根据上述得到飞行器发射控制系统的故障、测试、测试点、测试之间的关系，并且得到多信号流程图模型如图 3 所示，至此完成系统测试性建模。

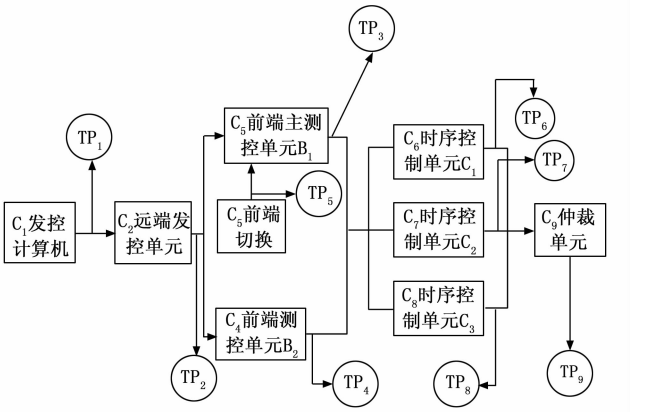

图 3　多信号流程图

2.3　测试性故障建模分析

上节完成系统的测试性建模，对于多信号流图模型进行分析的时候，可以引入相关性矩阵，相关性矩阵又称依存矩阵（***D*** 矩阵），表示系统中故障源和测试相关的关系矩阵。***D*** 矩阵是用数字定性地去表示测试和故障、故障和故障、测试和测试的依存关系。在进行系统测试性分析的时候，矩阵可以将测试性模型和故障分析联系起来，并且可以包括故障信息和测试信息。本文研究的相关性矩阵[11]如式（1）所示：

$$\boldsymbol{D}=\begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{bmatrix} \tag{1}$$

根据各模块的属性，结合建立的可达矩阵建立相关性矩阵，数据如表 4 所示。

表 4 相关性矩阵

C/T	TP_1	TP_2	TP_3	TP_4	TP_5	TP_6	TP_7	TP_8	TP_9
C_1	1	1	1	1	0	1	1	1	1
C_2	0	1	1	1	0	1	1	1	1
C_3	0	0	1	0	0	1	1	1	1
C_4	0	0	0	1	0	1	1	1	1
C_5	0	0	1	0	1	1	1	1	1
C_6	0	0	0	0	0	1	0	0	1
C_7	0	0	0	0	0	0	1	0	1
C_8	0	0	0	0	0	0	0	1	1
C_9	0	0	0	0	0	0	0	0	1

3 故障测试分析

3.1 故障特征建立

对于系统进行故障分析主要分为单故障分析和多故障分析。对于上述建立的系统相关矩阵进行分析，从而可以验证上述建立的故障测试矩阵。对于测试系统来说，在同一个时刻，最多出现一种故障，此时就是单故障。并且对于多数的测试系统来说，为了简化模型，多进行单故障的假设和分析。单故障分析主要进行三方面的思考，分析系统中模型建立的准确性、故障检测的完整性、测点设置的冗余性。根据系统的故障测试矩阵，可以从单故障和多故障方向进行分析，具体分析如下。

3.1.1 单故障分析

对于测试系统来说，在同一个时刻，最多出现一种故障，此时就是单故障。并且对于多数的测试系统来说，为了简化模型，多进行单故障的假设和分析。单故障分析主要进行三方面的思考，分析系统中模型建立的准确性、故障检测的完整性、测点设置的冗余性。根据系统的故障测试矩阵，可以从上述 3 个方面进行分析，具体分析如下。

1）模型建立的准确性：在进行系统模型的分析时，分析系统是否存在模糊项，即故障测试矩阵中是存在相同的行向量。如果存在不同行的元素值是相同的，就可以认为相同行对应的故障是没有办法进行区分的，即出现无法准确定位的故障，进而说明模型建立的准确性不够。此时需要对建立的模型的进行优化，保证故障测试矩阵没有模糊故障的情况存在。

2）故障检测的完整性：分析比较故障测试矩阵，观察矩阵是否出现行向量中每一个元素值都是 0。如果出现某行元素值全为 0，则表示该行代表的故障是通过测试无法检测到的。说明对于系统进行测试点布置的时候，无法完成系统所有故障的检测。因此，当故障测试矩阵中任意一行的行向量中所有的元素值不全为 0，说明系统测试可以检测所有故障，故障检测具有完整性。

3）测试性的冗余性：测点设置的冗余性即分析系统是否出现测试相同的情况，如果分析比较出故障检测矩阵中有些列向量对应的元素值都是相同的，说明出现了相同的测试项，测试冗余选择一个测试进行分析。如果故障测试矩阵中没有出现相同的列向量，则反映出测试项具有合理性。

根据相关性矩阵分析可知，飞行器发射控制系统，不存在某行向量的元素全为 0，不存在系统的列向量，不存在相同的行向量，因此，对于单故障分析而言，该测试性模型测点布置合理，并且模型可以准确反应系统各单元组成之间的关系，以及之间信号、故障关系。

3.1.2 多故障分析

对应大多数复杂系统来说，系统在相同时刻会同时发生多个故障。此时，不能对系统进行单故障分析，要采用多故障分析的方式，分析判断出系统中是否出现了由于多故障情况产生的隐藏故障和掩盖故障，从而准确分析整个系统，对系统进行故障的分析。

1）系统的隐藏故障：系统在某一时间内产生多个故障模式，系统中所有测试产生的故障结果和系统中某一个测点测试故障结果相同，我们就可以确定是这个故障使得系统无法进行故障定位，并且隐藏了其它测试点的故障特征信息。通过对系统的故障测试矩阵进行分析，如果出现这种情况，则认为该系统存在隐藏故障。

2）系统的掩盖故障：系统在某一时间内产生多个故障模式，系统中所有测试产生的故障结果和故障外某一个测点测试故障结果相同，我们就可以确定是这个故障使得系统无法进行故障定位，并且掩盖了其它测试点的故障特征信息。通过对系统的故障测试矩阵进行分析，如果出现这种情况，则认为该系统存在隐藏故障。

结合计算出实际得到的相关性矩阵，通过分析系统的测试结果，判断矩阵中行向量的之间的关系，发现系统中不存在掩盖故障，但是存在隐藏故障的情况，通过比较分析得到系统得隐藏故障如表 5 所示。

表 5 隐藏故障集

故障源	隐藏故障集	故障源	隐藏故障集
c_1	$c_2 \sim c_4, c_6 \sim c_9$	c_6	空集
c_2	$c_3 \sim c_4, c_6 \sim c_9$	c_7	空集
c_3	$c_6 \sim c_9$	c_8	空集
c_4	$c_6 \sim c_9$	c_9	空集
c_5	$c_6 \sim c_9$		

3.2 测试性参数

在对复杂系统进行测试性分析的时候，需要对故障建模的模型进行分析，分析测试点和测试性的选择，从而保证模型的正确性和测试的合理性。在大多数系统进行测试性分析的时候，主要需要考虑故障检测率和故障隔离率两个评价参数。

3.2.1 故障检测率

故障检测率的定义为在系统工作测试的一段时间内，按照系统的测试方法和流程完成系统的检测，正确检测出的系统故障的数量和被检测系统发生的故障总数的百分比，数学模型可以表示为：

$$FDR = \frac{N_D}{N_T} \times 100\% \tag{1}$$

式中，N_D 为系统检测项正确检测出的故障源数；N_T 为系统在一段工作时间实际产生的故障总数。

对于一些系统和设备来讲，假如各个故障发生的概率 λ 是一个常数，那么上式可以改写为：

$$FDR = \frac{\lambda_D}{\lambda} \times 100\% = \frac{\sum \lambda_{Di}}{\sum \lambda_i} \times 100\% \tag{2}$$

式中，λ_D 为系统能够被检测出的故障模式的概率总和；λ 为系统所有故障模式的发生总概率之和；λ_i 为第个故障发生的故障率；λ_{Di} 为第个系统被检测出的故障模式发生的故障率。

3.2.2　故障隔离率

故障隔离率定义为：在系统工作测试的一段时间内，按照系统的测试方法和流程完成系统的检测，正确检测出的系统故障的被正确隔离的故障数和被检测系统发生的故障总数的百分比，数学模型可以表示为：

$$FDR = \frac{N_L}{N_D} \times 100\% \tag{3}$$

式中，N_L 为在规定时间内测试项可以正确检测到的故障隔离数；N_D 为系统检测项正确检测出的故障源数。

对于一些系统和设备来分析及预计的数学模型为：

$$FDR = \frac{\lambda_L}{\lambda_D} \times 100\% = \frac{\sum \lambda_{Li}}{\lambda_D} \times 100\% \tag{4}$$

式中，λ_D 为系统能够被检测出的故障模式的概率总和；λ_L 为系统可隔离不大于个可更换单元数的故障模式的故障率之和；λ_{Li} 为系统可隔离不大于个可更换单元数中第个故障模式故障率；L 表示系统所有的可以被隔离更换的单元数。

3.3　软件设计思路和编程方法

对复杂系统进行测试性建模后，需要对系统测试性进行分析和验证。由于系统测试性参数有很多，为满足这些测实性要求，可以进行不同测试点布置和测试项的设计，如何计算得出最优的测试选择方法，成为复杂系统进行测试性分析过程中重要研究问题。

当面对测试项选择最优设计问题时候，智能算法在该类问题的解决上提供很好的方法，并且有很多学者提出多种算法，如粒子群算法[12-13]、遗传算法[14]、模拟退火算法[15-16]、贪婪算法[17-18]等。本文在遗传算法的基础上对系统进行测试性分析和验证。本文在遗传算法的基础上对系统进行测试性分析和验证。遗传算法是一个解最优化问题的概率论方法，它通过模仿自然界的进化过程来解决问题，而自然界的种群通过杂交、变异和自然选择经历着持续的变化[19-20]。对于各种各样的问题遗传算法都可以容易在计算机上实施。对于直接采用数学方法进行处理很麻烦的问题，遗传算法在解决这些复杂问题特别有用处，并且遗传算法在处理大规模、实际离散或者连续问题上非常有效。只要保证以模拟自然进化为基础，遗传算法可以被适当的设计和修正来探索解决问题的特有特征。因此，为解决各种优化问题，遗传算法形成一种方法和方法论。

当研究根据上述遗传算法流程，研究测试覆盖性问题，通过改进遗传算法，提出了一种基于测试覆盖度的算法，算法设计流程如下所示。

1）初始化：本文采用实数编码的方式，用浮点数表示每一个基因，每一个浮点数即表示一个故障源的解，通过对多个故障源进行适应度函数的计算可得到测试测试评价函数值，集合的数目即表征了染色体的长度，初始化种群数目为 50。

2）适应度计算：在适应度函数设计过程中，考虑系统的测试性指标，即故障检测率、故障隔离率。适应度计算应该遵循着，非负、连续的。并且应该考虑，如何在测试代价较低的情况下，系统的故障检测率和故障隔离率较高。本文提出的算法和传统算法在初始化，终止条件上设置相同，最大的区别和不同在于适应度函数的设计。传统适应度可取为：

$$f(T_s) = \left(1 - \frac{\sum_{t_i \in T_s} c_j}{\sum_{t_i \in T} c_i}\right) + \left(\frac{\gamma_{FD}}{\gamma'_{FD}}\right)^{\alpha} + \left(\frac{\gamma_{FI}}{\gamma'_{FI}}\right)^{\beta} \tag{5}$$

本文提出一种基于测试覆盖率的适应度函数，函数公式如下：

$$f(T_s) = \left(1 - \frac{\sum_{t_i \in T_s} c_i k_i}{\sum_{t_j \in T} c_j k_j}\right) + \gamma(T_s) \tag{6}$$

式中，c_i 测试费用；T_S 测试项；k_i 可以对测试进行评价的函数。

具体表达式为：

$$k(t_i) = \frac{h(t_i)}{d(t_i)} \tag{7}$$

式中，$h(t_i)$ 为测试项的覆盖度，即某个故障发生时，能够被测试项检测到的最小的数量；$d(t_i)$ 为测试项能够检测出系统中的故障数量。

对于测试覆盖度进行分析，结合系统建立的多信号流图模型，对系统进行分析，系统总共故障源个数为 10，根据测试和故障源的关系，可以得出评价函数如表 6 所示。

表 6　评价函数值

故障源	$h(ti)$	$d(ti)$	$k(ti)$
TP1	8	1	8
TP2	3	1	3
TP3	4	2	2
TP4	3	2	1.5
TP5	1	1	1
TP6	1	4	0.25
TP7	1	4	0.25
TP8	1	4	0.25
TP9	1	3	0.33

由于测试评价函数 $k(t_i)$ 和 $h(t_i)$ 成正比，和 $d(t_i)$ 成反比，所以，当测试评价函数值低的时候，其测试越为重要，越要优先进行，越可以将其优秀的个体进行遗传。所以，测点 6，7，8 最为重要，测试优先集会被最先选中。将各个测试函数值用折线如图 4 所示。

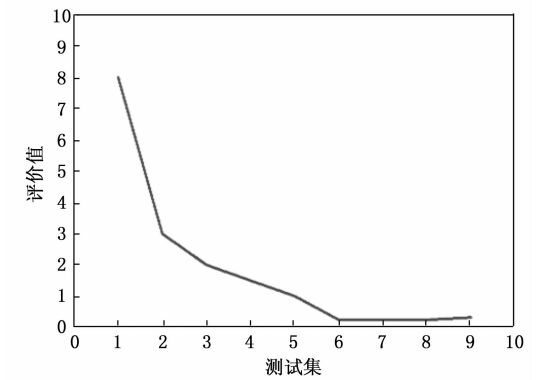

图 4 评价函数值

3）选择操作：本文的选择函数为式（8）：

$$\gamma(T_s)=\begin{cases}2 & \gamma_{FD}\geqslant\gamma'_{FD},\gamma_{FI}\geqslant\gamma'_{FI}\\ \left(\dfrac{\gamma_{FD}}{\gamma'_{FD}}+1\right) & \gamma_{FD}\leqslant\gamma'_{FD},\gamma_{FI}\geqslant\gamma'_{FI}\\ \left(\dfrac{\gamma_{FI}}{\gamma'_{FI}}+1\right) & \gamma_{FD}\geqslant\gamma'_{FD},\gamma_{FI}\leqslant\gamma'_{FI}\\ \left(\dfrac{\gamma_{FD}}{\gamma'_{FD}}\right)+\left(\dfrac{\gamma_{FI}}{\gamma'_{FI}}\right) & \gamma_{FD}\geqslant\gamma'_{FD},\gamma_{FI}\leqslant\gamma'_{FI}\end{cases}\tag{8}$$

式中，γ_{FD}为故障检测率；γ_{FI}故障隔离率；γ'_{FD}系统要求达到的故障检测率；γ'_{FI}系统要求达到的故障隔离率。

4）交叉变异：对于遗传算法，交叉概率多数在 0.3～1 之间，变异概率多数在 0～0.08 之间，本文选择交叉概率 0.6，变异概率为 0.008，根据此条件进行算法的计算，完成适应度函数的计算。

5）设置停止条件：本文规定程序停止运行的准则，计算出系统的适应度值，并将运行结果进行统计。

3.4 测试计算结果分析

根据上述算法设计，在 VS 中使用 MFC 完成，将遗传算法进行改进，提出基于测试覆盖率的算法进行编程，算法软件主要是 3 个组成部分：一个部分是将测试参数进行输入；一个部分是显示运行结果，显示算法优选测试集和故障率，检测率；最后一个部分显示算法的测试集和对应的适应度函数值，如图 5 所示。

图 5 算法结果

3.4.1 算法测试结果

本文中选用算法的参数选择，交叉概率选择为 0.6，变异概率为 0.008，并且给定测试项的个数，设置不同的停止规定条件，通过算法的运行可以计算出不同测试集的故障检测率，故障隔离率。通过输入不同的测试集合，可以得出相应的结果，计算结果如图 5 所示。将改进算法的输出结果列于表 7。

表 7 适应度值

测试集合	故障检测率/%	故障隔离率/%	适应度函数值
1,2,3,4,5,6,7,8,9	100	100	1.08
2,3,4,5,6,7,8,9	100	100	1.22
2,3,4,5,6,7,8	100	100	1.35
3,4,5,6,7,8	100	88.89	1.31
3,4,5,6,7	100	66.67	1.27
4,5,6	88.89	44.44	1.15
……	……	……	……

根据计算出的结果，可以分析计算出相应测试集对应的故障检测率和隔离率的折线图 6 所示。

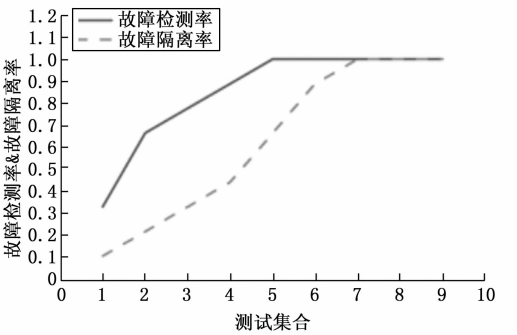

图 6 故障检测率和故障隔离率

从图 6 中可以清晰地知道，在算法计算中 7 个测试项就可以完成使得系统得故障检测率达到 100%，故障隔离率达到 100%，并且此时测试项最少，也就是说算法可以选择出最优个体组合为：｛2，3，4，5，6，7，8｝。测试项个数为 6 的时候，系统的故障隔离率没有达到 100%，即系统出现模糊项组合。

算法将测试项 1 和 9 进行了剔除，可以得出最新的相关性矩阵，如表 8 所示。根据此表进行分析，表格中任意的行向量都是不一样的，即没有模糊组，故障隔离率达到 100%。相关性矩阵中，每一个列向量都不一样，即没有冗余测试。并且系统的故障检测率和故障隔离率都为 100%。

表 8 算法改进相关性矩阵

C/T	TP_2	TP_3	TP_4	TP_5	TP_6	TP_7	TP_8
C_1	1	1	1	0	1	1	1
C_2	1	1	1	0	1	1	1
C_3	0	1	0	0	1	1	1
C_4	0	0	1	0	1	1	1
C_5	0	1	0	1	1	1	1
C_6	0	0	0	0	1	0	0
C_7	0	0	0	0	0	1	0
C_8	0	0	0	0	0	0	1
C_9	0	0	0	0	0	0	0

3.4.2 适应度函数值

当选择的测试项不同是计算出的适应度函数值如上述

表中所示，将其变化趋势用曲线图表示如图 7 所示。算法在 7 的时候适应度最高，该测试最优测试，会优先遗传。

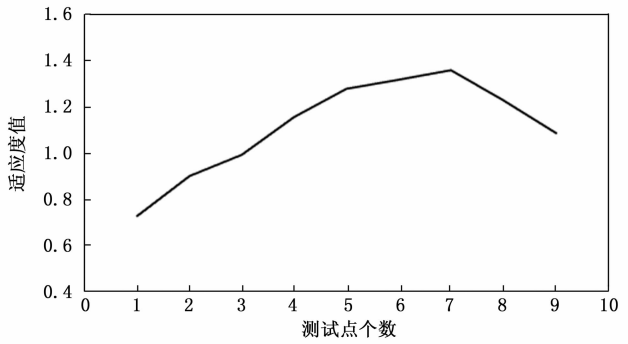

图 7　适应度函数值

3.4.3　算法的收敛性对比

根据本文提出算和传统遗传算法对比，其算法的收敛性如图 8 所示。本文设计算法在 15 代的就达到最优解，而传统遗传算在 21 代左右才收敛，本文算法收敛性更好。

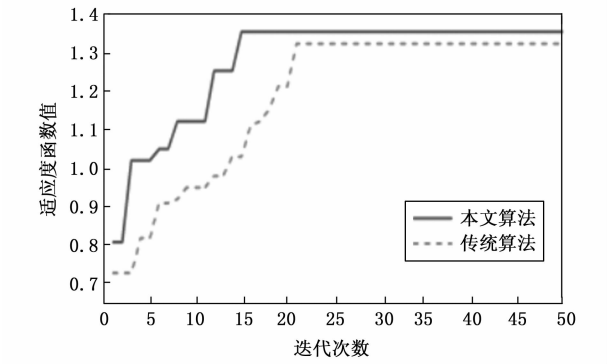

图 8　算法结果

根据本文提出算和传统遗传算法对比，其算法的收敛性如图 8 所示。由于本文的遗传算法自动对适应度函数进行优化设计，相比较传统的遗传算法适应度函数中的选择函数进行了 4 个范围的分类处理，对不同情况下的故障检测率和故障隔离率增添了 3 种可能情况，提高了算法的准确性，本文设计算法在 15 代的就达到最优解，而传统遗传算在 21 代左右才收敛，本文算法收敛性更好。

本文采用了小规模案例进行了分析，对于大规模案例的问题上，本文算法收敛性同样好于传统遗传算法，因为对选择函数的优化处理是具有通用性，同样适合大规模案例。

4　结束语

本文主要是分析了系统的故障模式和结构功能，根据实际系统的要求的功能，进行主要测试的分析，进行完成故障树模型，通过分析，进行相应的测试和测试点问题的讨论，建立多信号流图模型。结合系统的故障检测率和故障隔离率进行的系统级的故障、信号、测试点分析。为了更方便的进行测试分析和计算，进行遗传算法的研究，提出一种基于测试覆盖的改进遗传算法，并且和传统的遗传算法进行对比，改进后的算法有这更好的收敛性，对系统分析更好。由于篇幅问题，只是对系统级进行分析，同样的方法可以对具体的组成单元分析，更好地完成相关的测试分析和测点的布置，有利于指导系统进行设计。

参考文献：

[1] 张志华. 可靠性理论及工程应用 [M]. 北京：科学出版社，2012.

[2] 杨孟飞. 航天器控制计算机容错技术 [M]. 北京：国防工业出版社，2014.

[3] 许　忠，高玉水，徐振辉，等. 导弹测试发控系统可靠性研究 [J]. 火力与指挥控制，2006 (S1)：11－13.

[4] KITAMURA S，MORI K，SHINDO S. Modified multiobjective particle swarm optimization method and its application to energy management system for factories [J]. Electrical Engineering in Japan，2010，156 (4)：33－42.

[5] 徐　昕，高　飞，韩秀利，等. 快速机动发射运载火箭测发控系统的设计分析与展望 [J]. 计算机测量与控制，2020，28 (10)：125－129.

[6] DRYDEN M H. Design for reliability [J]. Technometrics，1976，15 (5)：399－436.

[7] 张学英，易　航，汪　洋，等. 运载火箭测发控系统通用化设计 [J]. 导弹与航天运载技术，2012 (4)：15－19.

[8] 李行善，左　毅，孙　杰. 自动测试系统集成技术 [M]. 北京：电子工业出版社，2004.

[9] 韩兆福，葛银茂，王锡仁. 基于 VXI 总线的机载火控系统自动测试仪的设计 [J]. 测试技术学报，2002，16 (1)：28－31.

[10] ELSAYED E A. Reliability engineering [M]. Data McGraw，2012：3－5.

[11] 曾声奎. 可靠性设计分析基础 [M]. 北京：北京航空航天大学出版社，2015.

[12] ARUN G，SESHI-REDDY D. Advanced fuzzy controller of fault analysis in a grid tied hybrid system [J]. Indian Journal of Science and Technology，2016，9 (35)：1－6.

[13] HABTEMARIAM G M，MOHAPATRA S K. A genetic algorithm-based approach for test case prioritization [M]. Springer Cham，2019：24－37.

[14] 刘振鹏，王雪峰，薛　雷，等. 实数编码遗传算法的改进及并行化实现 [J]. 河北大学学报：自然科学版，2019，39 (1)：86－92.

[15] FAISAL A，BILAL H. A hybrid cuckoo search and simulated annealing algorithm [J]. Journal of Intelligent Systems，2018，28 (4)：683－698.

[16] LING H，YU Z，QINGYU L，et al. Task-scheduling scheme based on greedy algorithm in integrated radar and communication systems [J]. The Journal of Engineering，2019 (19)：5864－5867.

[17] 崔　鹏，刘红静. 测试集问题的集合覆盖贪心算法的深入近似 [J]. 软件学报，2006 (7)：14－20.

[18] PRAHLAD A. Quality and reliability of missile system [J]. Defence Science Journal，2012，52 (1)：5－10.

[19] 菅鲁京，李运泽，张加迅，等. 考虑模糊推理及蒙特卡洛方法的毁伤评估研究 [J]. 航空兵器，2018，308 (6)：78－83.

[20] 苏　春，黄　茁，许映秋. 基于遗传算法和蒙特卡洛仿真的设备维修策略优化 [J]. 东南大学学报 (自然科学版)，2006，36 (6)：941－945.