

# 基于混合离散二进制粒子群—遗传算法的 测试配置方法研究

秦玉峰, 史贤俊, 郭家豪

(海军航空大学 岸防兵学院, 山东 烟台 264001)

**摘要:** 针对目前测试性建模工作中尚无具体方法指导测试配置这一问题, 通过对系统内故障传播关系进行分析, 提出了一种混合离散二进制粒子群—遗传算法用于求解测试配置的最优方案, 使系统的测试性模型在满足规定测试性指标下使用的测试数量最少; 将系统测试配置方案进行二进制粒子编码, 并在粒子群算法中引入遗传算子, 使混合算法具有较快的搜索速度的同时避免陷入局部最优; 最后通过实例计算与仿真, 证明所提出算法计算结果正确且对于指导复杂系统测试性建模工作具有实际应用价值。

**关键词:** 测试性模型; 测试配置; 二进制粒子群算法; 遗传算法

## Research on Test Configuration Method Based on Hybrid Discrete Binary Particle Swarm Optimization — Genetic Algorithm

Qin Yufeng, Shi Xianjun, Guo Jiahao

(Shore Defense School, Naval Aviation University, Yantai 264001, China)

**Abstract:** In view of the present testability modeling work, there is no specific methods to guide the test configuration, through the analysis of system failure propagation relation, this paper proposes a hybrid discrete binary particle swarm — genetic algorithm for solving the optimal scheme of test configuration, make the system testability model under the prescriptive testability index using the least number of tests. Binary particle coding is carried out in the system test configuration scheme, and genetic operators are introduced in particle swarm optimization algorithm, so that the hybrid algorithm has fast searching speed and avoids getting trapped in local optimum. In the end, it is proved that the proposed algorithm is correct and has practical application value to guide the test modeling of complex system.

**Keywords:** testability model; test configuration; BPSO; GA

### 0 引言

测试性建模是对系统中故障—测试信息与功能信息流向的一种图示或数学描述, 选择一种好的测试性模型是进行测试性分析与设计的前提<sup>[1]</sup>。一个好的测试性模型应既能体现系统的物理结构组成和功能信息流向, 又可以描述系统故障与测试之间的逻辑关系和对测试资源的占用关系等。为了提高系统的测试性指标, 建模人员在建立系统测试性模型时通常会设置大量的测试。目前系统的测试配置工作基本上都由建模人员凭借经验完成, 由于大型复杂系统的故障模式数量极多, 若为每个故障模式都设置测试, 对此测试性模型进行分析必然会使系统的测试性指标如: 故障检测率 (fault detection rate, FDR) 和故障隔离率 (fault isolation rate, FIR) 等达到最高<sup>[2]</sup>。但复杂的测试性模型增加了整个测试性分析工作的难度, 过多的冗余测试大幅降低了整个系统的诊断效率, 根据此模型对系统进行

设计和开发也会产生一系列问题, 如: 增加系统的设计难度、增加全寿命周期费用、提高研发成本等。因此有必要解决测试性建模过程中测试的优化配置问题, 简化测试性模型, 以最简约有效的方案最大程度地检测并隔离系统中的故障。

基于以上问题, 本文提出了一种混合离散二进制粒子群—遗传算法 (binary particle swarm optimization—genetic algorithm, BPSO—GA) 对复杂系统测试性建模过程中测试优化配置问题进行求解。

### 1 测试优化配置的数学描述与建模

#### 1.1 故障可达性矩阵

测试性建模工作主要是面对系统的故障空间, 然后为故障配置相应的测试。目的是建立系统内部各个功能单元之间的故障传播关系和故障与测试资源的逻辑关系<sup>[3]</sup>。根据系统的组成结构和故障模式、影响及危害性分析 (failure mode, effects and criticality analysis, FMECA) 可知, 系统内部的故障模式存在着一定的传播关系, 故障模式之间的这种传播关系可以用故障可达性矩阵表示。假设系统故障流图  $G = \langle FM, E \rangle$  中包含  $n$  个故障模式节点:  $FM = \{fm_1, fm_2, \dots, fm_n\}$ ;  $m$  条边:  $E = \{ \langle fm_i, fm_j \rangle \mid fm_i \in$

收稿日期: 2018-05-18; 修回日期: 2018-06-23。

**作者简介:** 秦玉峰(1995-), 男, 黑龙江大庆人, 硕士, 主要从事系统测试性建模、分析技术等方向的研究。

史贤俊(1968-), 男, 山东莱州人, 教授, 博士生导师, 主要从事自动化技术、飞行器测试性设计、故障诊断技术等方向的研究。

$FM, fm_j \in FM$ 。有序对  $\langle fm_i, fm_j \rangle$  表示故障模式节点  $fm_i$  到  $fm_j$  有一条有向边, 则定义  $n$  阶方阵  $f(G) = [f_{ij}]$  为系统的故障邻接矩阵, 其中:

$$f_{ij} = \begin{cases} 1 & \langle f_i, f_j \rangle \in E \\ 0 & \langle f_i, f_j \rangle \notin E \end{cases} \quad (1)$$

定义故障可达性矩阵  $F(G)$  如下: 若故障模式  $fm_i$  可以引起故障模式  $fm_j$  发生, 即从  $fm_i$  到  $fm_j$  存在一条通路, 则  $n$  阶方阵  $F(G) = [F_{ij}]$  称为系统的故障可达性矩阵, 其中:

$$F_{ij} = \begin{cases} 1 & \text{从 } fm_i \text{ 到 } fm_j \text{ 之间存在一条通路} \\ 0 & \text{从 } fm_i \text{ 到 } fm_j \text{ 之间不存在通路} \end{cases} \quad (2)$$

下面给出由故障邻接矩阵求取故障可达性矩阵的一般方法<sup>[4]</sup>:

1) 对故障邻接矩阵  $f(G)$  进行如下变换:

$$\mathbf{R} = [r_{ij}] = f(G) + (f(G))^2 + \dots + (f(G))^n = \sum_{k=1}^n (f(G))^k \quad (3)$$

式中,  $\mathbf{R}$  是表示任意两故障模式之间通路数目的矩阵。

2) 将矩阵  $\mathbf{R}$  中的全部非零元素置换为 1, 为零的元素保持不变, 就得到故障可达性矩阵, 其中:

$$F(G) = \begin{cases} 1 & r_{ij} \neq 0 \\ 0 & r_{ij} = 0 \end{cases} \quad (4)$$

## 1.2 测试配置方案

系统测试配置方案可用  $n$  维向量表示:

$$V_T = [v_1, v_2, \dots, v_n] \quad (5)$$

其中:  $v_i = 1$  表示为故障模式  $fm_i$  配置测试;  $v_i = 0$  表示不为故障模式  $fm_i$  配置测试。将  $V_T$  中  $v_i = 1$  的个数记为  $m$ , 表示为系统配置测试的总数量。其中: 系统最优测试配置方案应为最小完备测试集合, 即去除该集合中任意一个测试就不再满足测试性参数指标要求。

## 1.3 故障一测试相关性矩阵

测试性模型具有图示形式和矩阵形式两种表示方法, 故障一测试相关性矩阵以测试性图示模型为基础进行相关性分析得到, 故障一测试相关性矩阵便是对系统功能单元故障与测试资源相关性的数学描述。根据测试配置方案  $V_T$  为系统配置测试, 构建故障与测试资源的逻辑关系, 故障一测试相关性矩阵  $D_{n \times m}$  可由下式计算得到:

$$D_{n \times m} = F(G) \times \begin{bmatrix} v_1 & 0 & \dots & 0 \\ \vdots & v_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & v_n \end{bmatrix}_{n \times n} \quad (6)$$

去除矩阵中全零列  $\rightarrow$   $\begin{bmatrix} d_{11} & d_{12} & \dots & d_{1m} \\ d_{21} & d_{22} & \dots & d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nm} \end{bmatrix}_{n \times m}$

在建模过程中, 通常为系统作下列假设: (1) 被测对象仅有两种状态: 正常状态和故障状态。被测对象在正常状态下无故障可以正常工作; 反之被测对象不能正常工作。

(2) 当被测对象处于故障状态时, 则假设仅有一个组成元件 (或部件) 发生了故障。(3) 某一组成单元发生了故障, 在信息流动方向可达的各个测试点上, 测量的有效性是一样的。(4) 每个测试的测试结果都是二值且可靠的, 各测试之间相互独立。则故障一测试相关性矩阵内的元素: 若故障模式  $fm_i$  与测试  $t_j$  相关, 则  $d_{ij} = 1$ , 否则  $d_{ij} = 0$ 。

## 1.4 测试性参数指标

### 1.4.1 故障检测率

故障检测率描述了系统对故障模式的检测能力<sup>[5]</sup>, 可根据故障一测试相关性矩阵计算得到:

$$\gamma_{FD} = \frac{L_D}{n} \times 100\% \quad (7)$$

式中,  $L_D$  为矩阵  $D_{n \times m}$  中非全零行向量个数。若考虑各故障模式的故障率  $\lambda$ , 此时故障检测率的计算公式为:

$$\gamma_{FD} = \frac{\sum_{i=1}^{L_D} \lambda_i}{\sum_{i=1}^n \lambda_i} \times 100\% \quad (8)$$

### 1.4.2 故障隔离率

故障隔离率描述了系统对故障模式的隔离和诊断能力, 可根据故障一测试相关性矩阵计算得到:

$$\gamma_{FI} = \frac{L_I}{L_D} \times 100\% \quad (9)$$

式中,  $L_I$  为矩阵  $D_{n \times m}$  中非全零且唯一的行向量的个数。若考虑各故障模式的故障率  $\lambda$ , 此时故障隔离率的计算公式为:

$$\gamma_{FI} = \frac{\sum_{i=1}^{L_I} \lambda_i}{\sum_{i=1}^{L_D} \lambda_i} \times 100\% \quad (10)$$

## 1.5 优化模型

### 1.5.1 优化对象和目标

优化对象即对系统配置测试的方案, 因此, 优化对象模型可以用式 (5) 形式表示。测试最优配置方案需满足以下目标: 在使系统达到规定的测试性指标参数的情况下, 模型中测试数量最少。因此, 优化目标函数可表示为:

$$\min m \quad (11)$$

### 1.5.2 优化约束条件

在本文中, 优化目标规定系统需要达到规定的测试性指标这一前提。因此, 优化约束条件可表示为:

$$\begin{cases} \gamma_{FD} \geq FDR \\ \gamma_{FI} \geq FIR \end{cases} \quad (12)$$

式中, 系统要求达到的测试性指标分别记作 FDR、FIR, 在建模时若有充足的先验信息, 在计算  $\gamma_{FD}$ 、 $\gamma_{FI}$  时可以考虑故障率等因素, 使模型更接近实际。

## 2 基于混合离散二进制粒子群一遗传算法的测试优化配置方法

### 2.1 遗传算法基本原理

20 世纪 60 年代, Holland 受生物进化机制启发, 提出

了一种模拟自然界生物遗传进化的自适应概率搜索算法，称为遗传算法（Genetic Algorithm, GA）<sup>[6-8]</sup>。遗传算法将进化理论中“适者生存”这一基本思想引入串结构当中，并且在串之间进行有规则的随机信息交换。随着算法的不断运行，好的品质被保留下来并加以结合，以此来繁衍出更佳的个体。新一代个体中不但包含着上一代个体的大量信息，而且由于好的特征被不断地继承下来，在总体特征上也不断胜过旧的一代，从而使整个种群向前进化发展。遗传算法的基本概念包括：

**种群 (Population)：**是指遗传算法求解过程中，在解空间中的一个子集，即初始情况下多个解的集合。遗传算法从该初始种群开始计算求解。

**个体 (Individual)：**是指某个解集中的单个解，一个个体由某种数据结构描述其基本特征信息。本文中的个体即用二进制编码来表示一个测试配置方案。

**适应度函数 (Fitness)：**是遗传算法实现优胜劣汰的主要依据，其作用是计算种群中每个个体的环境适应度并进行评估，它决定了染色体的优劣程度。

**染色体 (Chromosome)：**是将个体进行编码后得到的编码串，也可称为基因型个体，编码串中的单个元素称为一个基因。

**遗传操作 (Genetic Operator)：**遗传操作包括选择、交叉、变异 3 种基本形式，是一种从原始种群产生新种群的操作。

GA 包含以下要素：

$$GA = (P(0), N, l, s, Q_r, Q_c, Q_m, P_r, P_c, P_m, f) \quad (13)$$

式中， $P(0)$  为初始种群； $N$  为种群规模； $l$  为染色体长度； $s$  为对个体的选择策略； $Q_r$  为选择算子； $Q_c$  为交叉算子； $Q_m$  为变异算子； $P_r$  为选择概率； $P_c$  为交叉概率； $P_m$  为变异概率； $f$  为适应度函数。

### 2.2 离散二进制粒子群算法基本原理

Eberhart 博士和 Kennedy 博士于 1995 年提出了粒子群优化算法 (Particle Swarm Optimization, PSO)<sup>[9-11]</sup>。为解决实际问题，二人于 1997 年又提出了离散二进制粒子群优化算法 (Binary Particle Swarm Optimization, BPSO)<sup>[12-13]</sup>。粒子群算法仅根据粒子的速度进行搜索，没有遗传算法复杂的选择、交叉和变异操作，结构简单，运行速度较快并且拥有记忆功能，但粒子仅仅将当前搜索到的最优位置作为共享信息，容易陷入局部最小，从而出现所谓的“早熟收敛”现象。

粒子群中任意一个粒子的信息都可以用如下信息来描述：①粒子的当前位置  $X_i = (x_{i1}, x_{i2}, \dots, x_{im})$ ；②粒子搜索到的当前最优解  $P_i = (p_{i1}, p_{i2}, \dots, p_{im})$ ，记作 Pbest，称为个体极值；③粒子在搜索空间中的飞行速度  $V_i = (v_{i1}, v_{i2}, \dots, v_{im})$ ，这里  $i = 1, 2, \dots, n$ 。若粒子  $i$  的当前位置优于其历史最优位置，则将粒子  $i$  的当前位置更新作为历史最优位置。 $v_{ij}$  和  $x_{ij}$  的更新公式为：

$$v_{ij} = w \cdot v_{ij} + c_1 \cdot rand_1() \cdot (p_{ij} - x_{ij}) +$$

$$c_2 \cdot rand_1() \cdot (p_{gj} - x_{ij}) \quad (14)$$

$$x_{ij} = x_{ij} + v_{ij} \quad (15)$$

式中， $w$  为惯性权重； $c_1, c_2$  为粒子群学习因子； $rand_1, rand_2$  是两个随机分布在  $(0, 1)$  之间的正实数。整个粒子群也存在一个历史最优位置  $P_g = (P_{g1}, P_{g2}, \dots, P_{gn})$ ，记作 Gbest，即粒子群搜索到的最优解，称为全局极值。

对于 BPSO 中的任一粒子，其每一维的  $x_{ij}$  和  $Pbest_{ij}$  都用 0 或者 1 表示，而  $v_{ij}$  则变为粒子位置取 1 的概率，速度越快则粒子位置取 1 的概率越高。由于神经网络中的 Sigmoid 函数具有相似的特点，所以一般用该函数将粒子速度映射到区间  $[0, 1]$  内：

$$Sig(v_{ij}) = \frac{1}{1 + \exp(-v_{ij})} \quad (16)$$

则粒子位置更新公式可表示为：

$$x_{ij} = \begin{cases} 1 & rand() < sigmoid(v_{ij}) \\ 0 & rand() \geq sigmoid(v_{ij}) \end{cases} \quad (17)$$

### 2.3 混合离散二进制粒子群—遗传算法

由于 GA 存在搜索速度慢，没有种群的移动，不能参考历史信息等缺点<sup>[14]</sup>。为解决系统的测试优化配置问题，本文将二进制粒子群算法与遗传算法相结合，提出一种 BP-SO-GA 算法。该算法将遗传算法和离散二进制粒子群算法相结合，通过综合两种算法的优点，使新算法既能保证较快的搜索速度和成功率，又可以拥有良好的信息共享机制，避免陷入局部最优。

BPSO-GA 的流程如图 1 所示。

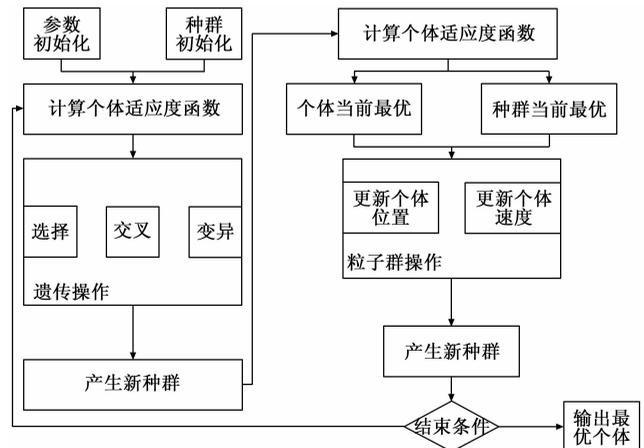


图 1 BPSO-GA 流程图

BPSO-GA 的总体思路是：首先对每一代种群分别计算其中个体对应的适应度函数，并进行遗传操作。在经过规定代数的迭代后再进行粒子群操作产生新种群，将种群中达到指标要求的个体在数据库中进行存储。最后对数据库中个体进行比较，将数据库中测试数量最少的个体输出为最优个体。下面介绍基于 BPSO-GA 的测试配置问题求解的几个关键步骤：

1) 粒子编码：将第  $k$  个粒子的二进制编码设置为  $V_{T_k} = [v_{1k}, v_{2k}, \dots, v_{mk}]$ ，此时种群中每一个粒子都对应一种测试

的配置方案;

2) 为防止 Sigmoid 函数饱和使算法早熟, 本文将粒子最大速度设置为 2, 此时的 Sigmoid 函数为:

$$Sig(v_{ij}) = \begin{cases} 0.88 & v_{ij} > 2 \\ \frac{1}{1 + \exp(-v_{ij})} & -2 \leq v_{ij} \leq 2 \\ 0.12 & v_{ij} < -2 \end{cases} \quad (18)$$

3) 构造适应度函数: 首先计算种群中个体的测试性指标  $\gamma_{FD}$  和  $\gamma_{FI}$ 。由于个体的适应度主要由 3 个参数决定: 测试配置数量  $m$ 、系统故障检测率  $\gamma_{FD}$ 、系统故障检测率  $\gamma_{FI}$ 。由于测试数量  $m$  越少, 测试性指标  $\gamma_{FD}$ 、 $\gamma_{FI}$  越大, 个体的适应度越大。因此, 本文将适应度函数设置为:

$$f(s) = \varphi \cdot \frac{\gamma_{FD} \cdot \gamma_{FI}}{m} \quad (19)$$

$\varphi$  为权值系数。

### 3 实例验证

对某装备涡扇发动机的传动供电系统建立测试性模型, 系统功能框图如图 2 所示。该系统有 25 个故障模式, 根据系统功能框图和先验故障传播关系, 建立系统的故障邻接矩阵  $f(G)$ , 如表 1 所示。

表 1 传动供电系统故障邻接矩阵

	$fm_1 - fm_{25}$
$fm_1$	101100000000000000000000000000
$fm_2$	011100000000000000000000000000
$fm_3$	001011000000000000000000000000
$fm_4$	000111000000000000000000000000
$fm_5$	000010111010010000000000000000
$fm_6$	000001111010010000000000000000
$fm_7$	000000100000000000000000000000
$fm_8$	000000010000000000000000000000
$fm_9$	000000001000001100000000000000
$fm_{10}$	000000000100001100000000000000
$fm_{11}$	000000000010001100000000000000
$fm_{12}$	000000000001001100000000000000
$fm_{13}$	000000000000101100000000000000
$fm_{14}$	000000000000011100000000000000
$fm_{15}$	000000000000001011000000000000
$fm_{16}$	000000000000000111000000000000
$fm_{17}$	000000000000000010000000000000
$fm_{18}$	000000000000000001000000000000
$fm_{19}$	00000000000000000001000011
$fm_{20}$	000000000000000000000100011
$fm_{21}$	0000000000000000000000010011
$fm_{22}$	00000000000000000000000001011
$fm_{23}$	000000000000000000000000000111
$fm_{24}$	00000000010110000000000010
$fm_{25}$	00000000010110000000000001

根据公式 (3), (4) 可计算得到系统故障可达性矩阵  $F(G)$ , 限于文章篇幅, 此处不再赘述。得到故障可达性矩

阵后为混合算法设置初始参数:  $N_{max} = 100, Popsiz = 20, P_c = 0.8, P_m = 0.1, c_1 = c_2 = 2, \omega = 0.9, v_{max} = 1, \varphi = 1$ 。

仿真算例 (一): 考虑为所有故障模式都设置测试的情况, 仿真结果如表 2 所示。

表 2 测试配置仿真结果(一)

$V_T$	$f(s)$	$\gamma_{FD}$	$\gamma_{FI}$	$m$	不能检测到的故障	不能隔离的故障
11111111 11111111 11111111 1111	0.04	1.0	1.0	25	无	无

由表 2 可知, 在所有故障模式均可测的情况下系统的故障检测率  $\gamma_{FD}$  和故障隔离率  $\gamma_{FI}$  都为 100%。但在实际中由于接口设计、物理结构、成本等问题无法为所有的故障模式都设置测试点, 此时的测试性模型的测试性指标虽然都为 100%, 却无法落实到系统研发设计当中。

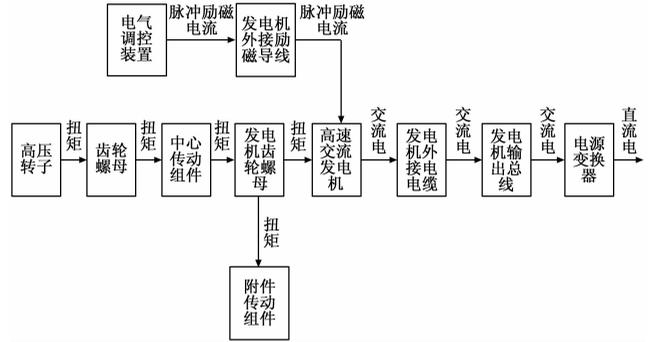


图 2 传动供电系统功能框图

仿真算例 (二): 计算  $FDR = 0.8, FIR = 0.6$  时的优化仿真结果, 如表 3 所示。

表 3 测试优化配置仿真结果(二)

$V_T$	$f(s)$	$\gamma_{FD}$	$\gamma_{FI}$	$m$	不能检测到的故障	不能隔离的故障
01111 01000 01101 01101 10001	0.046	0.96	0.63	13	$fm_8$	$fm_9$ $fm_{10}$ $fm_{11}$ $fm_{14}$ $fm_{15}$ $fm_{19}$ $fm_{22}$ $fm_{24}$ $fm_{25}$

上述结果表明, 当系统对故障检测率和隔离率的要求为  $FDR \geq 0.8, FIR \geq 0.6$  时。表 3 中列出了 BPSO-GA 得出的最优解, 以及它们不能够检测到的故障模式和不能够隔离的故障模式。由该测试配置方案可得最少要设置  $m = 13$  个测试才能满足该指标。

(下转第 149 页)