

测试不可靠条件下基于量子进化算法的测试优化选择

雷华军¹, 秦开宇²

(1. 西南科技大学信息工程学院, 四川绵阳 621010; 2. 电子科技大学航空航天学院, 四川成都 611731)

摘 要: 测试优化选择是复杂电子系统测试性设计中的一个重要问题. 首先从测试容差的角度分析了测试发生漏检和虚警的原因, 在此基础上建立了测试不可靠条件下一种新的测试选择模型, 模型以测试代价、漏检代价和虚警代价之和最小为优化目标, 以故障检测率和故障隔离率为约束条件; 然后提出一种改进的量子进化算法对模型求解, 该算法通过改进一种已有可靠测试选择算法而成, 包括种群初始化、适应度计算和种群的进化策略. 最后通过两个仿真实例验证了求解算法及模型的有效性和优越性.

关键词: 测试性设计; 测试优化选择; 不可靠测试; 量子进化算法

中图分类号: TP206 **文献标识码:** A **文章编号:** 0372-2112 (2017)10-2464-09

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2017.10.022

Optimal Selection of Imperfect Tests Based on Improved Quantum-Inspired Evolutionary Algorithm

LEI Hua-jun¹, QIN Kai-yu²

(1. School of Information Engineering, Southwest University of Science and Technology, Mianyang, Sichuan 621010, China;
2. School of Aeronautics & Astronautics, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China)

Abstract: Optimal selection of tests is an important problem that arises in design for testability for complex electronic systems. Firstly, from the perspective of test tolerance, the reason of how tests produce miss detection and false alarm is analyzed. Then, a new mathematic model for the problem of test selection in the presence of imperfect tests is developed. It consists of minimizing the sum of test cost, miss detection cost and false alarm cost, subject to lower bound constraints on fault detection rate and fault isolation rate. To optimize the model, an improved quantum-inspired evolutionary algorithm is proposed. It is formed by making some improvements to an extant algorithm that has been used for optimal selection of perfect tests, including population initialization, fitness calculation and the strategy of population evolution. Finally, two simulation examples are used to validate the effectiveness and superiority of the solution method and the model.

Key words: design for testability; test selection; imperfect tests; quantum-inspired evolutionary algorithm

1 引言

对于大型复杂电子系统, 传统的测试和诊断方法存在故障信息获取困难、诊断准确性差、虚警率高等问题. 大量的研究和实践表明: 要从根本上解决上述问题, 必须在系统设计之初就开展测试性设计^[1]. 测试选择是测试性设计中的一项重要内容, 其目的是选择满足测试性指标要求且在一定的优化准则下最佳的测试组合. 它对测试位置、参数和手段等进行配置和优化, 以便

采取经济可靠的方式获取系统的状态信息, 从而满足系统的故障诊断需求. 本文将观测某一系统参数是否正常的过程定义为一个测试, 而不区分所采用的观测手段.

近年来, 国内外学者在传感器布局优化^[2-4]、模拟电路测点选择^[5-8]以及基于相关性矩阵的测试集优化^[9-11]等方面开展了广泛的研究, 提出了大量的测试优选模型. 但这些模型大多假定故障与测试之间具有确定的因果关系, 认为所有测试都是可靠的, 不存在漏

检和虚警.然而,实际中的测试受到多种因素的影响,包括测试设备自身故障、环境影响、不合理的测试容差等,使得故障与测试之间的逻辑关系往往表现出不确定性.针对该问题,文献[12~15]研究了考虑传感器自身故障率的布局优化,数学模型为带约束的多目标优化.文献[16]建立了测试不可靠条件下的测试优选模型,并利用遗传算法求解,但该研究缺乏对测试不可靠原因的理论分析,忽略了测试不可靠对故障检测可靠性的影响.另一方面,即便测试可靠,测试选择都是一个NP-完全问题,当问题规模较大时,要得到最优的测试集通常较困难,多目标优化模型无疑使得问题的求解更加困难.

鉴于上述问题,本文从以下两方面开展工作:

(1)建立了测试不可靠条件下一种新的单目标测试选择模型,模型以系统的故障检测率、故障隔离率为约束条件,以测试代价、漏检代价和虚警代价之和最小为优化目标.

(2)提出一种基于量子进化算法(Quantum-inspired Evolutionary Algorithm, QEA)的模型求解算法,该算法是对文献[10]中的测试选择算法的进一步改进.

2 问题描述及数学模型

2.1 故障-测试相关性矩阵

测试选择前,在故障空间对系统进行测试性建模是首要步骤,目的是获取如下所示的故障-测试相关性矩阵 $D = (d_{ij})_{m \times n}$,它描述了故障和测试之间的因果关系^[17].

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

矩阵的行对应故障集 $F = \{f_1, f_2, \dots, f_m\}$,各故障的故障率为 $P = (p_1, p_2, \dots, p_m)$.矩阵的列对应测试集 $T = \{t_1, t_2, \dots, t_n\}$,各测试的代价为 $C = (c_1, c_2, \dots, c_n)$.通常假设所有测试均对应一个二值输出(正常/异常),若测试 $t_j (j=1, 2, \dots, n)$ 输出正常, $t_j = 0$, 否则 $t_j = 1$.在二值输出的假设下,矩阵元素 d_{ij} 为布尔变量.若故障 f_i 能被测试 t_j 检测, $d_{ij} = 1$, 否则 $d_{ij} = 0$.矩阵的第 i 行向量 $F_i = (d_{i1}, d_{i2}, \dots, d_{in})$ 描述了故障 f_i 发生时全部测试的输出,可视为故障 f_i 的特征;第 j 列向量 $T_j = (d_{1j}, d_{2j}, \dots, d_{mj})^T$ 描述了测试 t_j 可检测的所有故障,反映了 t_j 的故障检测能力.有关 D 矩阵详细的分析与讨论可参考文献[18].

2.2 漏检与虚警

假设故障 f_i 与测试 t_j 相关,即 $d_{ij} = 1$.理想情况下,

t_j 对 f_i 的检测特性如图 1 所示.图中曲线①表示 f_i 故障时 t_j 观测值的概率分布,曲线②表示 f_i 正常时 t_j 观测值的概率分布.由图 1 可知, f_i 与 t_j 之间存在确定的因果关系:(1)若故障 f_i 发生,测试 t_j 的观测值总是落在预设的容差外,因此 $t_j = 1$,检测到 f_i 的概率为 1;(2)若故障 f_i 正常,测试 t_j 的观测值总是落在容差内,因此 $t_j = 0$,指示 f_i 发生故障的概率为 0.

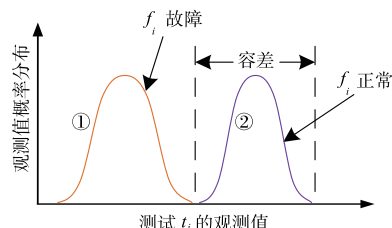


图1 测试的理想故障检测特性

实际中的测试受多种因素的影响往往表现出不可靠性,其故障检测特性如图 2 所示.由此得到 f_i 与 t_j 之间的不确定因果关系:(1)若故障 f_i 发生,测试 t_j 不能确保 100% 检测到 f_i ,可能发生漏检,因此由 t_j 输出正常不能认为 f_i 一定无故障;(2)若故障 f_i 正常,测试 t_j 可能错误的指示 f_i 发生故障而导致虚警,因此由 t_j 输出异常也不能断定 f_i 就一定故障.

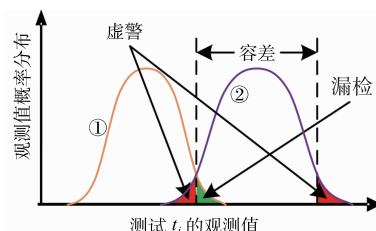


图2 测试的实际故障检测特性

为了描述 f_i 与 t_j 之间的不确定因果关系,引入参数 pd_{ij} 和 pf_{ij} . pd_{ij} 表示 t_j 对 f_i 的检测概率, pf_{ij} 表示 t_j 对 f_i 的虚警概率,分别定义如下:

$$pd_{ij} = p(t_j = 1 | d_{ij} = 1, f_i = 1) \quad (1)$$

$$pf_{ij} = p(t_j = 1 | d_{ij} = 1, f_i = 0) \quad (2)$$

其中, $f_i = 1$ 表示 f_i 发生, $f_i = 0$ 表示 f_i 正常.为了定义的完整性,当 f_i 与 t_j 不相关(即 $d_{ij} = 0$),对应的 $pd_{ij} = pf_{ij} = 0$,由此得到检测概率矩阵 $Pd = (pd_{ij})_{m \times n}$ 和虚警概率矩阵 $Pf = (pf_{ij})_{m \times n}$.

2.3 测试性指标计算

由于系统同时发生多个故障的概率很小,故假设系统在任意时刻最多只有一个故障发生.设 $x = (x_1, x_2, \dots, x_n)$ 为待求的测试集,若测试 $t_j (j=1, 2, \dots, n)$ 被选中, $x_j = 1$, 否则, $x_j = 0$.记 $F_{ix} = F_i \cdot * x = (d_{ij}x_j)_{1 \times n}$, 其中符号 $*$ 表示两个向量对应元素相乘,结果为同维向量.

(1)故障检测率.一般定义为:在规定的时间内由

测试集 \mathbf{x} 正确检测到的故障模式的总故障率同系统所有故障模式的总故障率之比^[1]. 理想情况下, 故障 f_i 被

\mathbf{x} 检测的条件为: \mathbf{F}_{ix} 为非零向量, 即 $\sum_{j=1}^n d_{ij}x_j \geq 1$. 设 \mathbf{F}_D 为 \mathbf{x} 可检测的故障构成的集合, 则有:

$$\mathbf{F}_D = \{f_i \mid f_i \in \mathbf{F}, \sum_{j=1}^n d_{ij}x_j \geq 1\} \quad (3)$$

考虑测试的不可靠性, 故障检测率可表示为:

$$\gamma_{FD}(\mathbf{x}) = \frac{\sum_{f_i \in \mathbf{F}_D} p_i \cdot (1 - \prod_{j=1}^n (1 - pd_{ij})^{d_{ij}})}{\sum_{i=1}^m p_i} \quad (4)$$

其中, $\prod_{j=1}^n (1 - pd_{ij})^{d_{ij}}$ 为 $f_i (\in \mathbf{F}_D)$ 被 \mathbf{x} 漏检的概率.

(2) 故障隔离率. 一般定义为: 在规定时间内由测试集 \mathbf{x} 正确隔离的故障模式的总故障率与同一时间内检测到故障的总故障率之比^[1]. 理想情况下, 故障 f_i 与 $f_k (i \neq k)$ 被 \mathbf{x} 隔离的条件为: f_i 与 f_k 均可被 \mathbf{x} 检测, 且向量 \mathbf{F}_{ix} 与 \mathbf{F}_{kx} 不同, 即 $\mathbf{F}_{ix} \oplus \mathbf{F}_{kx} = 1, \forall f_i, f_k \in \mathbf{F}_D, i \neq k$. 其中符号 \oplus 表示向量的异或运算, 如果两个向量对应元素都相同, 结果为 0, 否则为 1. 设 \mathbf{F}_I 为 \mathbf{x} 可隔离的故障构成的集合, 则有:

$$\mathbf{F}_I = \{f_i \mid f_i \in \mathbf{F}_D, \sum_{\substack{f_k \in \mathbf{F}_D, \\ i \neq k}} \mathbf{F}_{ix} \oplus \mathbf{F}_{kx} = m - 1\} \quad (5)$$

考虑测试的不可靠性, 故障隔离率可表示为:

$$\gamma_{FI}(\mathbf{x}) = \frac{\sum_{f_i \in \mathbf{F}_I} p_i \cdot \prod_{j=1}^n (pd_{ij})^{d_{ij}}}{\sum_{f_i \in \mathbf{F}_D} p_i \cdot (1 - \prod_{j=1}^n (1 - pd_{ij})^{d_{ij}})} \quad (6)$$

其中, $\prod_{j=1}^n (pd_{ij})^{d_{ij}}$ 为 $f_i (\in \mathbf{F}_I)$ 被 \mathbf{x} 正确隔离的概率.

2.4 优化目标

由式(4)可知, 即使为故障 f_i 配置了相应的测试, f_i 仍可能被漏检. 此外, f_i 也可能被虚警. 这两类错误影响测试集 \mathbf{x} 的故障检测可靠性. 在系统运行过程中, f_i 被漏检, 可能导致故障效应的进一步升级, 使系统无法正常运行甚至崩溃; 相对的, f_i 被虚警, 可能导致系统不必要的停机和维修操作. 因此在测试不可靠的条件下, 仅考虑测试代价是不全面的, 还应考虑这两类错误潜在的影响.

基于此, 提出如下单目标函数 $E(\mathbf{x})$:

$$E(\mathbf{x}) = TC(\mathbf{x}) + LC(\mathbf{x}) + AC(\mathbf{x}) \quad (7)$$

其中, $TC(\mathbf{x})$ 、 $LC(\mathbf{x})$ 和 $AC(\mathbf{x})$ 分别为测试集 \mathbf{x} 的测试代价、漏检代价和虚警代价, 具体定义如下:

(1) 测试代价 $TC(\mathbf{x})$

$$TC(\mathbf{x}) = \sum_{j=1}^n c_j \cdot x_j \quad (8)$$

(2) 漏检代价 $LC(\mathbf{x})$

$$LC(\mathbf{x}) = \sum_{f_i \in \mathbf{F}_D} p_i \cdot \prod_{j=1}^n (1 - pd_{ij})^{d_{ij}} \cdot cl_i \quad (9)$$

其中, $p_i \cdot \prod_{j=1}^n (1 - pd_{ij})^{d_{ij}}$ 为故障 f_i 发生且被漏检的概率, cl_i 表示 f_i 漏检导致的费用.

(3) 虚警代价 $AC(\mathbf{x})$

$$AC(\mathbf{x}) = \sum_{f_i \in \mathbf{F}_D} (1 - p_i) \cdot (1 - \prod_{j=1}^n (1 - pf_{ij})^{d_{ij}}) \cdot ca_i \quad (10)$$

其中 $(1 - p_i) \cdot (1 - \prod_{j=1}^n (1 - pf_{ij})^{d_{ij}})$ 为故障 f_i 正常且被虚警的概率, ca_i 表示 f_i 虚警导致的费用.

式(10)中, 参数 cl_i 和 ca_i 通常根据故障危害度和维修费用等因素确定, 或由专家依据经验给出. 目标函数 $E(\mathbf{x})$ 以漏检代价和虚警代价的形式量化了两类错误对故障检测可靠性的影响, 并结合测试代价, 使其更能全面的衡量测试集 \mathbf{x} 的优劣.

2.5 优化模型

综合以上分析, 得到如下测试选择模型:

$$\begin{cases} \min & E(\mathbf{x}) \\ \text{s. t.} & \begin{cases} \gamma_{FD}(\mathbf{x}) \geq \gamma_{FD}^* \\ \gamma_{FI}(\mathbf{x}) \geq \gamma_{FI}^* \end{cases} \end{cases} \quad (11)$$

其中, γ_{FD}^* 、 γ_{FI}^* 分别为系统要求达到的最低故障检测率和故障隔离率. 特别的, 当测试均可靠时, 即 $pd_{ij} = 1, pf_{ij} = 0$, 将这两个条件带入式(11)即得到测试可靠条件下的优化模型, 详见文献[10]中的式(3).

3 模型求解算法

3.1 QEA 概述^[19]

QEA 是 Han 和 Kim 于 2002 年提出的一种基于量子计算原理的进化算法, 它采用量子比特编码染色体, 一个量子比特可表示如下:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (12)$$

其中, $|0\rangle$ 和 $|1\rangle$ 表示两个不同的量子态, α 和 β 均为复数, 分别为状态 $|0\rangle$ 和 $|1\rangle$ 的概率幅. 一个量子比特可能处于 $|0\rangle$ 或 $|1\rangle$ 两个基态, 或者两基态的任意叠加态, $|\alpha|^2$ 和 $|\beta|^2$ 分别指定了该量子比特处于状态 $|0\rangle$ 和 $|1\rangle$ 的概率. 用量子比特编码的染色体, 如下所示:

$$\mathbf{q} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_l \\ \beta_1 & \beta_2 & \cdots & \beta_l \end{bmatrix} \quad (13)$$

式中: l 为染色体长度, 每个量子比特的概率幅满足 $|\alpha_k|^2 + |\beta_k|^2 = 1, (k = 1, 2, \dots, l)$. 采用量子比特编码, 一个长度为 l 的染色体表达 2^l 个解的叠加; 而传统的二进制编码, 一个染色体只能表示一个确定的解. 因此相比采用二进制编码的进化算法, QEA 更容易保持种群的多样性.

通过对量子染色体 \mathbf{q} 的随机观测,得到一个确定的二进制解 $\mathbf{a} = (a_1, a_2, \dots, a_l)$, 即 $a_k \in \{0, 1\}$ ($k = 1, 2, \dots, l$). 具体的观测过程为:对量子染色体 \mathbf{q} 的每个量子比特 $[\alpha_k, \beta_k]^T$ ($k = 1, 2, \dots, l$), 产生一个随机数 Rnd , $\text{Rnd} \in [0, 1]$. 若 $\text{Rnd} < |\beta_k|^2$, 则 $a_k = 1$, 否则, $a_k = 0$.

在 QEA 中,量子染色体的进化更新通过量子旋转门实现,量子染色体 \mathbf{q} 的第 k ($k = 1, 2, \dots, l$) 位量子比特更新方式如下:

$$\begin{bmatrix} \alpha'_k \\ \beta'_k \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta_k) & -\sin(\Delta\theta_k) \\ \sin(\Delta\theta_k) & \cos(\Delta\theta_k) \end{bmatrix} \begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix} \quad (14)$$

其中, $[\alpha'_k, \beta'_k]^T$ 为更新后的第 k 位量子比特, $\Delta\theta_k$ 为旋转角,其取值由下式确定^[10]:

$$\Delta\theta_k = \begin{cases} \theta \times \{\text{sign}(\alpha_k \beta_k) \times (b_k - a_k)\}, & f(\mathbf{a}) < f(\mathbf{b}) \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

式中, θ 为旋转角的幅值, $\text{sign}(\cdot)$ 为符号函数, $f(\cdot)$ 为适应度函数, b_k 为进化目标 \mathbf{b} 的第 k 位. 按照式(14)和(15)更新所有量子比特位,即实现对 \mathbf{q} 的更新. 可看出更新量子 \mathbf{q} 时,需知道 \mathbf{q} 的二进制解 \mathbf{a} 、进化目标 \mathbf{b} 以及它们的适应度.

文献[10]将 QEA 用于可靠测试的优化选择,提出 IQEA 求解算法,并通过大量实例验证了算法的优良性能. 通过前期的仿真实验,发现 IQEA 在求解某些测试选择问题时仍存在早熟现象,求解成功率不高,且不能直接用求解模型(11). 为此,对 IQEA 进行改进,以示区别,将原算法命名为 IQEA_I,改进算法命名为 IQEA_II. 关于 QEA 及 IQEA_I 的详细介绍参考文献[10,19].

3.2 种群初始化

算法开始时,初始化种群 $\mathbf{Q}(0) = \{\mathbf{q}_1^0, \mathbf{q}_2^0, \dots, \mathbf{q}_M^0\}$, 其中, M 为种群规模, \mathbf{q}_j^0 ($j = 1, 2, \dots, M$) 为第 j 个量子染色体,定义如下:

$$\mathbf{q}_j^0 = \begin{bmatrix} \alpha_{j1}^0 & \alpha_{j2}^0 & \dots & \alpha_{jn}^0 \\ \beta_{j1}^0 & \beta_{j2}^0 & \dots & \beta_{jn}^0 \end{bmatrix} \quad (16)$$

\mathbf{q}_j^0 表达了解空间中所有解的取值概率分布.

在 IQEA_I 中,所有 α_{jk}^0 和 β_{jk}^0 ($j = 1, 2, \dots, M; k = 1, 2, \dots, n$) 均初始化为 $1/\sqrt{2}$, 这意味着在初始化阶段每种测试组合以相同的概率叠加. 当随机观测 \mathbf{q}_j^0 时,每种测试组合出现的概率亦相同,本文称这种初始化方法为等概率方式. 从搜索空间的角度,量子染色体的初始值决定了算法的起始搜索空间,等概率方式得到的起始搜索空间的分布几乎为随机噪声,即算法随机地从某个空间出发开始搜索. 文献[20]指出若已知问题相关的先验信息,利用这些信息初始化量子染色体可改善算法的性能. 受此启发,提出一种利用贪婪解初始量子

染色体的方法.

假设利用贪婪算法已求得了模型(11)的一个解为 $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)$. 一般来说, \mathbf{x}^0 为局部最优解或近似最优解,但获得 \mathbf{x}^0 的计算开销通常较小,且仅需一次. 若 \mathbf{x}^0 在搜索空间的位置处于全局最优解的邻近区域,那么利用 \mathbf{x}^0 初始化量子染色体,可使得起始搜索空间亦邻近最优解,从而引导算法可能在较短时间里找到全局最优解. 假设共有 L ($1 \leq L \leq M$) 个量子染色体利用 \mathbf{x}^0 初始化,其中第 l ($l = 1, 2, \dots, L$) 个量子染色体 \mathbf{q}_l^0 通过替换 \mathbf{x}^0 得到,替换方式为:用式(17)替换 \mathbf{x}^0 中为“1”的位置,用式(18)替换 \mathbf{x}^0 中为“0”的位置. 照此方式初始化所有 L 个量子染色体.

$$\begin{bmatrix} \alpha_{lk}^0 \\ \beta_{lk}^0 \end{bmatrix} = \begin{cases} \begin{bmatrix} \sqrt{\frac{0.5 - 2\delta}{L-1}(l-1) + \delta} \\ \sqrt{1 - \frac{0.5 - 2\delta}{L-1}(l-1) - \delta} \end{bmatrix} & \text{if } x_k^0 = 1 \\ \begin{bmatrix} \sqrt{1 - \frac{0.5 - 2\delta}{L-1}(l-1) - \delta} \\ \sqrt{\frac{0.5 - 2\delta}{L-1}(l-1) + \delta} \end{bmatrix} & \text{if } x_k^0 = 0 \end{cases} \quad (17)$$

$$\begin{bmatrix} \alpha_{lk}^0 \\ \beta_{lk}^0 \end{bmatrix} = \begin{cases} \begin{bmatrix} \sqrt{1 - \frac{0.5 - 2\delta}{L-1}(l-1) - \delta} \\ \sqrt{\frac{0.5 - 2\delta}{L-1}(l-1) + \delta} \end{bmatrix} & \text{if } x_k^0 = 1 \\ \begin{bmatrix} \sqrt{\frac{0.5 - 2\delta}{L-1}(l-1) + \delta} \\ \sqrt{1 - \frac{0.5 - 2\delta}{L-1}(l-1) - \delta} \end{bmatrix} & \text{if } x_k^0 = 0 \end{cases} \quad (18)$$

式(17)(18)中, δ ($0 \leq \delta \ll 1$) 为观测到“1”或“0”的最小概率. 容易验证,当随机观测式(17)的量子比特时,观测到“1”的概率大于观测到“0”的概率;而观测式(18)的量子比特时,结果正好相反. 因此,当观测 \mathbf{q}_l^0 时, \mathbf{x}^0 出现的概率最大.

为使初始种群具有足够的多样性,其余 $M - L$ 个量子个体采用等概率方式初始化. 参数 L 和 δ 对算法性能有直接影响,但理想的值较难确定,为简单起见, L 和 δ 分别设置为: $L = M/2, \delta = 0.01$.

3.3 贪婪算法

贪婪算法的关键在于贪婪准则的选择和贪婪解的构造策略. 针对前者,提出如下贪婪准则:

$$k' = \arg \max_{\substack{k \in \{1, 2, \dots, n\} \\ x_k^0 = 0}} \left\{ \frac{1}{\Delta E} [\Delta \gamma_{FD} \cdot U(\gamma_{FD}^* - \gamma_{FD}(\mathbf{x}^0)) + \Delta \gamma_{FI} \cdot U(\gamma_{FI}^* - \gamma_{FI}(\mathbf{x}^0))] \right\} \quad (19)$$

其中, $\Delta \gamma_{FD}$ 、 $\Delta \gamma_{FI}$ 和 ΔE 分别为新选择一个测试后故障检测率、故障隔离率及目标函数的增量; $U(\cdot)$ 为阶跃函数.

针对后者,类比模拟电路测点选择采用的故障字典分割方法^[5,6],提出如下构造策略:

Step 1 初始化贪婪解 $\mathbf{x}^0 = (0, 0, \dots, 0)_{1 \times n}$, $F_D = \emptyset, F_I = \emptyset$.

Step 2 初始化模糊组集 $\mathbf{AS} = \{\mathbf{AS}_1, \mathbf{AS}_2, \dots, \mathbf{AS}_r\}$ 和子矩阵集 $\mathbf{DS} = \{\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_r\}$. 一开始 \mathbf{x}^0 为零向量,故有 $r = 1, \mathbf{AS} = \{\mathbf{F}\}, \mathbf{DS} = \{\mathbf{D}\}$.

Step 3 选择使得式(19)取最大值的测试 t_k . 设 t_k 为某一未选中的测试(即 $x_k^0 = 0$), 初始化集合 $FD_k \leftarrow F_D, FI_k \leftarrow F_I$.

Step 3.1 对子矩阵 $D_q (q = 1, 2, \dots, r)$, 根据测试 t_k 对应列元素的值将 AS_q 分成两个模糊组, 对应元素为“1”的故障添加到 AS_{q1} , 对应元素为“0”的故障添加到 AS_{q0} .

Step 3.2 更新 $FD_k, FD_k \leftarrow FD_k \cup AS_{q1}$.

Step 3.3 更新 FI_k . 若 $|AS_{qi}| = 1$ 且 $AS_{qi} \in FD_k (i = 0, 1), FI_k \leftarrow FI_k \cup AS_{qi}$.

Step 3.4 对其余子矩阵重复 Step 3.1 到 Step 3.3, 直到所有子矩阵均被处理.

Step 3.5 根据 FD_k 及 FI_k 计算故障检测率 γ_{FD} 、故障隔离率 γ_{FI} 及目标函数 E_k , 进而计算式(19)的比值, 假定为 Δ_k .

Step 3.6 对当前未选中的测试重复 Step 3.1 到 Step 3.5.

Step 3.7 选取最大 Δ_k 对应的测试 t_k , 令 x^0 相应位置元素等于 1, 即 $x_k^0 = 1$. 同时, 更新 $F_D \leftarrow FD_k, F_I \leftarrow FI_k, \gamma_{FD}(x^0) \leftarrow \gamma_{FD}, \gamma_{FI}(x^0) \leftarrow \gamma_{FI}, E(x^0) \leftarrow E_k$.

Step 4 若 $\gamma_{FD}(x^0) \geq \gamma_{FD}^*, \gamma_{FI}(x^0) \geq \gamma_{FI}^*$ 均成立, 算法停止, x^0 即为所求的贪婪解; 否则转步骤 5.

Step 5 更新 AS 和 DS .

Step 5.1 对测试 t_k 执行 Step 3.1, 将 AS 中的 AS_q 替换为 AS_{q1} 和 AS_{q0} , 相应的 D_q 被按行分割成 D_{q1} 和 D_{q0} , 同时替换 D_q 为 D_{q1} 和 D_{q0} .

Step 5.2 若 $|AS_{qi}| = 1$ 且 $AS_{qi} \subset F_D (i = 0, 1)$, 将 AS_{qi} 从 AS 中去掉, 同时从 DS 去掉 D_{qi} . 重复 Step 5.1 到 Step 5.2 直到所有子矩阵均被处理, 转 Step 3.

上述步骤每选择一个测试的最坏时间复杂度为 $O(pm)$, 其中 p 为当前未选测试的个数. 显然, 刚开始 $p = n$. 假设在求解 x^0 的过程中一共选择了 q 个测试, 则贪婪算法的最坏时间复杂度为: $O(nm + (n-1)m + \dots + (n-q+1)m) = O\left(\frac{2n-q+1}{2}qm\right)$. 考虑到 Step 5.2 总是去掉单行的子矩阵, 因此贪婪算法实际的时间复杂度低于 $O\left(\frac{2n-q+1}{2}qm\right)$.

3.4 适应度计算

第 g 代, 通过对 $Q(g)$ 的随机观测, 得到一组二进制解 $P(g) = \{x_1^g, x_2^g, \dots, x_M^g\}$. 为了评估 $x_j^g (j = 1, 2, \dots, M)$ 的适应度, 定义如下适应度函数:

$$f(x_j^g) = \frac{\alpha}{E(x_j^g)} - \beta \cdot \max(0, \gamma_{FD}^* - \gamma_{FD}(x_j^g)) - \gamma \cdot \max(0, \gamma_{FI}^* - \gamma_{FI}(x_j^g)) \quad (20)$$

式中, α, β, γ 均为正常数. 显然, 若 x_j^g 为可行解, $f(x_j^g)$

> 0 , 目标函数值 $E(x_j^g)$ 越小, 其适应度越大. 当 x_j^g 为不可行解时, 使 $f(x_j^g) < 0$, 确保可行解的适应度一定大于不可行解的适应度. 据此, 设置这些参数的取值为: $\alpha = 1, \beta = \gamma = 20$.

由文献[10]的时间复杂度分析结果, 不难得出计算 x_j^g 适应度的时间复杂度为 $O(m \log m)$.

3.5 种群的进化策略

设 $b_j^g = (b_{j1}^g, b_{j2}^g, \dots, b_{jn}^g)$ 为量子染色体 $q_j^g (j = 1, 2, \dots, M)$ 搜索到的历史最优解, $b_{\text{global}}^g = (b_1, b_2, \dots, b_n)$ 为整个种群搜索到的历史最优解. 在 IQEA_1 中, 选择 q_j^g 进化目标的策略为: 若 $f(x_j^g) < f(b_j^g)$, 进化目标为 b_j^g ; 否则, 进化目标为 b_{global}^g . 该方法在更新种群时综合利用了全局极值和个体极值, 但仍存在早熟现象. 究其原因, 一个可能的解释是: 全局极值 b_{global}^g 较频繁的作为进化目标虽使种群快速收敛, 但同时导致种群多样性快速丧失, 以至于到算法中后期, 大多数量子染色体已趋于收敛, 不再具有搜索能力, 此时若仍未找到最优解将使得算法最终陷入局部极值. 针对该问题, 从增加种群多样性的角度提出两种更新种群的方法, 分别用于算法的不同阶段.

(1) 基于多次随机观测的种群更新

Step 1 对量子染色体 $q_j^g (j = 1, 2, \dots, M)$ 进行 $K (K \geq 1)$ 次随机观测, 得到 K 个解 $x_{j(1)}^g, x_{j(2)}^g, \dots, x_{j(K)}^g$.

Step 2 根据式(20)分别计算这些解的适应度, 从中选择适应度最大的解, 记为 $x_{j(\text{best})}^g$, 同时更新个体极值 b_j^g 及全局极值 b_{global}^g .

Step 3 选择 $x_{j(\text{best})}^g$ 和 b_j^g 分别作为 q_j^g 的解和进化目标, 并按照式(14)和(15)更新 q_j^g .

Step 4 重复 Step 1 ~ 3, 更新 $Q(g)$ 中的所有量子染色体.

如 3.1 节所述, 量子染色体本质上是一个描述各个解取值概率的概率模型. 当一个量子染色体未收敛, 多次随机观测可得到多个不同的解, 此举有助于扩大搜索空间和保持种群多样性. 但当 K 值较大时, 多次随机观测会显著的增加计算量, 并且随着进化代数增加, 量子染色体趋于收敛, 多次随机观测一个量子染色体将会得到几乎相同的多个解. 因而, 该种群更新方法适用于算法的初始阶段.

利用上述步骤更新一个量子染色体的时间复杂度为 $O(Km \log m)$, 因而更新整个种群的时间复杂度为 $O(MKm \log m)$.

(2) 基于多源极值的种群更新

Step 1 对 $Q(g) = \{q_1^g, q_2^g, \dots, q_M^g\}$ 进行一次随机观测, 得到一组解 $P(g) = \{x_1^g, x_2^g, \dots, x_M^g\}$.

Step 2 根据式(20)计算 $x_j^g (j = 1, 2, \dots, M)$ 的适

应度,同时更新个体极值 b_j^g 及全局极值 b_{global}^g .

Step 3 若 $\text{mod}(g, R)$ 为 0, 选择 b_{global}^g 作为 $q_j^g (j=1, 2, \dots, M)$ 的进化目标, 并按照式 (14) 和 (15) 更新 q_j^g , 其中 $\text{mod}(g, R)$ 表示 g 除以 R 的余数. 否则, 转 Step4.

Step 4 按照事先设定的概率 $pc (pc \geq 0)$, 从 $Q(g)$ 中随机选择 W 对量子染色体 $q_{A_i}^g$ 和 $q_{B_i}^g (i=1, 2, \dots, W)$.

Step 4.1 以 $q_{A_i}^g$ 的个体极值作为 $q_{B_i}^g$ 的进化目标, $q_{B_i}^g$ 的个体极值作为 $q_{A_i}^g$ 的进化目标, 并按照式 (14) 和 (15) 分别更新 $q_{A_i}^g$ 和 $q_{B_i}^g$. 照此方式, 更新其余 $W-1$ 对量子染色体.

Step 4.2 其余 $M-2W$ 个量子染色体以自身的个体极值作为进化目标, 并按照式 (14) 和 (15) 进行更新.

其中 Step 3 利用全局极值更新种群, 实现整个种群的信息共享. Step 4 随机配对染色体并互换进化目标, 实现个体间的信息共享. 对一个量子染色体来说, 引导其进化的目标可能是全局极值、自身的个体极值以及其它染色体的个体极值, 这就是“多源极值”名字的由来.

上述步骤的时间复杂度为 $O(Mm \log m)$ ($m \log m \geq n$) 或 $O(Mn)$ ($m \log m \leq n$), 可以看出交换进化目标的操作不增加时间复杂度.

3.6 IQEA_II 流程

综合 IQEA_I 及上述改进措施, 得到 IQEA_II 的流程如下:

Step 1 设置进化代数计数器 $g=0$ 以及最大进化代数 G .

Step 2 调用 3.3 节的贪婪算法生成 x^0 , 然后按 3.2 节的方法初始化种群 $Q(g) = \{q_1^g, q_2^g, \dots, q_M^g\} |_{g=0}$.

Step 3 若 $g < G_1$, 调用基于多次随机观测的种群更新方法, 参数 G_1 控制多次随机观测执行的代数, 且满足 $G_1 < R$; 否则, 转 Step 4.

Step 4 调用基于多源极值的种群更新方法.

Step 5 增加计数器 $g \leftarrow g+1$, 若 $g > G$, 转 Step6; 否则, 转 Step3.

Step 6 输出 b_{global}^g 作为最优测试集.

需要指出的是, Step3 和 Step4 中涉及的旋转角度值由 IQEA_I 中的方法确定, 即 $\theta_{\text{max}} = 0.05\pi$, $\theta_{\text{min}} = 0.01\pi$. 综合前述时间复杂度的分析结果, 可得 IQEA_II 的时间复杂度为 $O((G + (K-1)G_1)Mm \log m)$ ($m \log m \geq n$) 或 $(G + (K-1)G_1)Mn$ ($m \log m \leq n$). 可以看出, 3.2 节的种群初始化方法不增加整个算法的时间复杂度, 但多次随机观测次数 K 及其执行代数 G_1 影响算法的时间复杂度.

4 实例仿真

本节通过两个实例验证 IQEA_II 及模型 (11) 的有

效性, 限于篇幅, 所有实例系统的故障-测试相关性矩阵 D 见相应文献. 对于每个实例, 假定故障率 p_i 为 $(0.001, 0.1)$ 之间的随机数, 测试代价 c_j 为 $(1, 10)$ 之间的随机数, 检测概率 pd_{ij} 为 $(0.98, 1)$ 之间的随机数, 虚警率 pf_{ij} 为 $(0, 0.02)$ 之间的随机数, 漏检费用 cl_i 为 $(100, 200)$ 之间的随机数, 虚警费用 ca_i 为 $(50, 100)$ 之间的随机数. 系统要求的故障检测率不低于 0.9, 故障隔离率不低于 0.85. 为了结果的客观性, 表或者图中所示结果都是 30 次独立试验后取的平均值.

实例 1 来自文献 [21] 的雷达发射机, 它共有 18 个故障, 21 个测试 (去掉一个冗余测试); 实例 2 来自文献 [22] 的超外差接收器系统, 它共有 22 个故障, 36 个测试.

4.1 测试可靠

为了直观比较 IQEA_I 与 IQEA_II 的求解性能, 首先考虑测试可靠的情况. IQEA_II 的性能受参数 G_1 、 K 、 R 和 pc 的影响, 通过对两个实例的求解, 最终确定两组合适的参数. 假定 G_1 的可能取值为 $\{10, 20, 30\}$, K 的可能取值为 $\{1, 2, 3, 4\}$, R 的可能取值为 $\{40, 60, 80\}$, pc 的可能取值为 $\{0, 0.35, 0.5, 0.7\}$, 因此共有 144 种组合. 为简单起见, 分 2 步确定这些参数:

(1) 确定 R 和 pc 的最佳取值组合. 设置 IQEA_II 的参数为: $M=8$, $G=500$, $G_1=0$, 其中 $G_1=0$ 意味着 IQEA_II 不包含 Step 3. 表 1 为 R 和 pc 的不同组合对应的统计结果, 据此确定利用 IQEA_II 求解两个实例时, R 和 pc 的取值均为 60 和 0.35. 从表 1 也不难看出, 大多数情况下随机配对交换演化目标的操作 ($pc > 0$) 有助于改善算法的求解精度和成功率.

表 1 参数 R 和 pc 的不同组合对应的统计结果

实例	例 1		例 2	
	平均目标值	成功率 (%)	平均目标值	成功率 (%)
(40, 0)	19.7155	56.67	46.9102	3.33
(40, 0.35)	19.6352	60.00	43.1750	33.33
(40, 0.50)	19.8795	50.00	44.0627	20.00
(40, 0.70)	20.3571	20.00	46.8437	20.00
(60, 0)	19.5435	63.33	49.9983	0.00
(60, 0.35)	19.4749	66.67	42.8215	33.33
(60, 0.50)	20.2769	33.33	45.6292	20.00
(60, 0.70)	20.2769	33.33	46.8446	16.16
(80, 0)	20.1165	40.00	50.5191	0.00
(80, 0.35)	19.7155	56.67	43.2187	26.67
(80, 0.50)	19.8759	50.00	45.6709	20.00
(80, 0.70)	19.8759	50.00	46.5440	16.67

(2) 确定 G_1 和 K 的最佳取值组合. 设置 IQEA_II

的参数为: $M = 8, G = 500, R = 60, pc = 0.35$. 表 2 为 G_1 和 K 的不同组合对应的统计结果, 据此确定利用 IQEA_II 求解实例 1 和 2 时, G_1 和 K 的取值组合分别为 (20, 2) 和 (20, 3). 由表 2 可看出, 相比单次随机观测 ($K = 1$), 多次随机观测 ($K > 1$) 的求解精度和成功率更高, 但计算时间也随之增加.

表 2 参数 K 和 G_1 的不同组合对应的统计结果

实例	例 1			例 2		
	平均目标值	成功率 /%	平均运行时间 /s	平均目标值	成功率 /%	平均运行时间 /s
(10, 1)	19.6353	60.00	4.8458	42.8535	36.67	6.0646
(10, 2)	19.3947	70.00	4.9698	41.8043	46.67	6.1706
(10, 3)	19.1540	80.00	5.0451	41.9066	53.33	6.1857
(10, 4)	19.3947	70.00	5.1068	42.1370	50.00	6.2850
(20, 1)	19.7957	53.33	4.8574	42.5597	36.67	5.9923
(20, 2)	18.9936	86.67	5.0347	41.7890	46.67	6.2183
(20, 3)	19.1540	80.00	5.2036	40.7818	70.00	6.4333
(20, 4)	19.6353	60.00	5.3380	41.9516	50.00	6.6459
(30, 1)	19.3947	70.00	4.9414	42.9352	36.67	5.9776
(30, 2)	19.1540	80.00	5.1496	41.8528	53.33	6.3135
(30, 3)	19.2342	76.67	5.4118	41.1409	53.33	6.6534
(30, 4)	19.0738	83.33	5.6650	41.1888	50.00	6.9629

由上述讨论, 可以看出 IQEA_II 对参数敏感. 通过对比不同参数组合的统计结果, 表明前述针对量子种群的进化更新而提出的改进措施是有效的.

分别应用 IQEA_I 和 IQEA_II 来求解实例 1 和 2, 其参数设置如表 3 所示. 两算法对实例 1 和 2 的优化结果列于表 4, 优化过程中性能指标与进化代数的关系曲线如图 3 和 4 所示.

表 3 IQEA_I 和 IQEA_II 的参数设置

实例	算法	参数
例 1	IQEA_I	$M = 30, G = 300$
	IQEA_II	$M = 30, G = 300, R = 60, pc = 0.35, G_1 = 20, K = 2$
例 2	IQEA_I	$M = 30, G = 300$
	IQEA_II	$M = 30, G = 300, R = 60, pc = 0.35, G_1 = 20, K = 3$

根据表 4, 对于实例 1, IQEA_I 和 IQEA_II 在 30 次独立实验中均找到了最优解 (成功率 100%), 平均收敛时间相当, 但 IQEA_I 的平均收敛代数小于 IQEA_II. 对于实例 2, IQEA_II 的成功率依然为 100%, 而 IQEA_I 的成功率只有 80%, 说明 IQEA_II 有助于防止早熟现象, 较 IQEA_I 更容易求得全局最优解. 若只考虑收敛到最优解的独立实验, 两算法的平均收敛时间仍然相

当, 但 IQEA_I 的平均收敛代数仍小于 IQEA_II.

表 4 IQEA_I 和 IQEA_II 对实例 1 和 2 的优化统计结果

实例	算法	最优目标值	成功率 /%	最大收敛代数	平均收敛代数	平均收敛时间/s
例 1	IQEA_I	18.6728	100	89	47.9667	2.7213
	IQEA_II	18.6728	100	101	62.2333	2.5935
例 2	IQEA_I	39.6172	80	241	148.4167	9.2822
	IQEA_II	39.6172	100	253	186.9333	9.7471

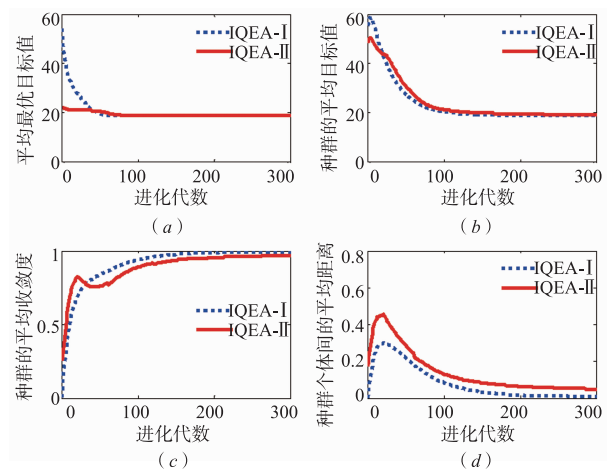


图 3 实例 1 的性能指标随进化代数的变化曲线

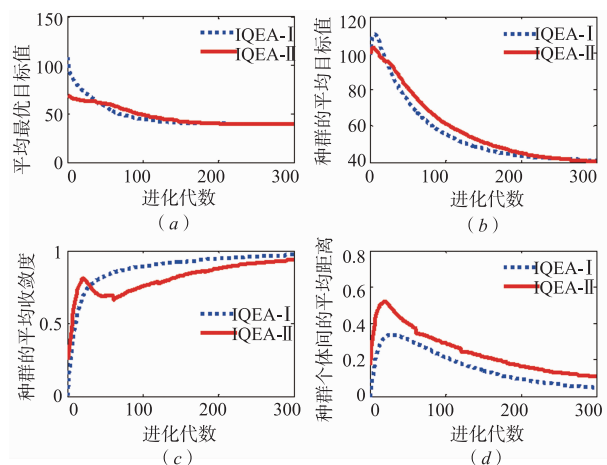


图 4 实例 2 的性能指标随进化代数的变化曲线

由图 3(a) 和 4(a) 可见, 初始时 IQEA_II 的平均最优目标值远低于 IQEA_I, 缘于 IQEA_II 初始化种群时融入了贪婪解的信息. 由图 3(a)(b)、图 4(a)(b) 可见, 随后的一段时间内 IQEA_I 的平均最优目标值和种群的平均目标值优于 IQEA_II, 这是因为 IQEA_I 更新种群时频繁使用全局极值, 致使其收敛速度快于 IQEA_II. 图 3(c)(d)、图 4(c)(d) 显示 IQEA_I 快速收敛的同时, 其种群多样性也较快丧失, 以至于到算法后期, 种群趋于收敛, 持续搜索能力较弱. 相比而言,

IQEA_II 采用多种保持种群多样性的措施,收敛速度虽慢于 IQEA_I,但能有效的防止算法陷入局部极值.关于图中所示性能指标的定义可参考文献[23].

4.2 测试不可靠

为验证模型(11),对第 $i(i=1,2)$ 个实例分两种情况讨论,然后利用 IQEA_II 求解:(1)目标函数仅为测试代价 $TC(\mathbf{x})$,求得实例 1 和 2 的最优测试集分别为 \mathbf{x}_1^2 和 \mathbf{x}_2^2 ;(2)目标函数为总代价 $E(\mathbf{x})$,求得实例 1 和 2 的最优测试集分别为 \mathbf{x}_1^3 和 \mathbf{x}_2^3 .同时,记测试可靠时实例 1 和 2 的最优测试集分别为 \mathbf{x}_1^1 和 \mathbf{x}_2^1 .

根据模型(11)相关指标的计算方法对比最优解 \mathbf{x}_i^j ($i=1,2;j=1,2,3$),结果如表 5 所示.

表 5 两实例的 3 个最优解对比结果

实例	最优解	故障检测率	故障隔离率	测试代价	总代价
例 1	\mathbf{x}_1^1	0.9006	0.8294	18.6728	57.9475
	\mathbf{x}_1^2	0.9006	0.8825	22.7939	63.0990
	\mathbf{x}_1^3	0.9980	0.8512	28.1304	59.1904
例 2	\mathbf{x}_2^1	0.9120	0.8508	39.6172	85.4355
	\mathbf{x}_2^2	0.9120	0.8508	39.6172	85.4355
	\mathbf{x}_2^3	0.9109	0.8528	42.9024	83.1159

从表 5 可以看出,最优测试集及其测试性指标随着优化模型的不同而不同:

(1)对于实例 1, \mathbf{x}_1^1 的测试代价和总代价均最小,但不满足测试性指标要求.若按文献[16]仅以测试代价为优化目标, \mathbf{x}_1^2 为最优测试集.综合考虑两者, \mathbf{x}_1^3 比 \mathbf{x}_1^2 更优.虽然 \mathbf{x}_1^3 的测试代价大于 \mathbf{x}_1^2 ,但漏检代价和虚警代价之和小于 \mathbf{x}_1^2 ,即测试集 \mathbf{x}_1^3 权衡了测试代价和故障检测可靠性.

(2)对于实例 2,若仅考虑测试代价,该系统在测试可靠和不可靠条件下具有相同的最优测试集 \mathbf{x}_2^1 和 \mathbf{x}_2^2 ,但它们的漏检代价和虚警代价之和较大,综合测试代价和故障检测可靠性, \mathbf{x}_2^3 更优.

通过上述对比,验证了模型(11)的有效性.

5 结论

本文在建立测试选择模型时分析了实际测试的不可靠检测特性,通过量化测试不可靠对故障检测可靠性的影响,构造了一种兼顾测试代价和故障检测可靠性的单目标优化模型.针对模型求解,通过改进 IQEA_I,提出一种更通用的测试选择算法 IQEA_II.仿真结果表明,IQEA_II 克服了 IQEA_I 易陷入局部极值的缺点,具有较高的求解成功率;单目标测试选择模型实现了测试代价和故障检测可靠性两者的权衡.

需要指出,本文在建立模型时将测试代价、漏检代

价和虚警代价三者等权重相加作为优化目标,这样处理是否合理,怎样确定它们的权重.从算法的角度,验证 IQEA_II 性能的实例规模还不够大,该算法是否适用于求解大规模测试选择问题或其它组合优化问题.这些问题都有待未来进一步深入研究.

参考文献

- [1] 田仲,石君友.系统测试性设计与验证[M].北京:北京航空航天大学出版社,2003.
TIAN Zhong, SHI Jun-you. System Testability Design, Analysis and Verification [M]. Beijing: Beijing University of Aeronautic and Astronautics Press, 2003. (in Chinese)
- [2] RAGHURAJ R, BHUSHAN M, RENGASWAMY R. Locating sensors in complex chemical plants based on fault diagnostic observability criteria [J]. AICHE Journal, 1999, 45(2): 310-322.
- [3] BHUSHAN M, RENGASWAMY R. Design of sensor network based on the signed directed graph of the process for efficient fault diagnosis [J]. Industrial and Engineering Chemistry Research, 2000, 39(4): 999-1019.
- [4] BAGAJEWICZ M, FUXMAN A, URIBE A. Instrumentation network design and upgrade for process monitoring and fault detection [J]. AICHE Journal, 2004, 50(8): 1870-1880.
- [5] STARZYK J A, LIU D, LIU Z H, et al. Entropy-based optimum test points selection for analog fault dictionary techniques [J]. IEEE Transactions on Instrumentation and Measurement, 2004, 53(3): 754-761.
- [6] YANG C L, TIAN S L, LONG B. Application of heuristic graph search to test points selection for analog fault dictionary techniques [J]. IEEE Transactions on Instrumentation and Measurement, 2009, 58(7): 2145-2158.
- [7] LEI H J, QIN K Y. Quantum-inspired evolutionary algorithm for analog test point selection [J]. Analog Integrated Circuits and Signal Processing, 2013, 75(3): 491-498.
- [8] LEI H J, QIN K Y. Greedy randomized adaptive search procedure for analog test point selection [J]. Analog Integrated Circuits and Signal Processing, 2014, 79(2): 371-383.
- [9] 陈希祥,邱静,刘冠军.基于混合二进制粒子群-遗传算法的测试优化选择研究[J].仪器仪表学报,2009,30(8): 1674-1680.
CHEN Xi-xiang, QIU Jing, LIU Guan-jun. Optimal test selection based on hybrid BPSO and GA [J]. Chinese Journal of Scientific Instrument, 2009, 30(8): 1674-1680. (in Chinese)
- [10] 雷华军,秦开宇.基于改进量子进化算法的测试优化选择[J].仪器仪表学报,2013,34(4): 838-844.
LEI Hua-jun, QIN Kai-yu. Optimal test selection based on improved quantum-inspired evolutionary algorithm [J].

- Chinese Journal of Scientific Instrument, 2013, 34(8): 838 – 844. (in Chinese)
- [11] 马羚, 李海军, 王成刚, 等. 考虑换件维修代价的测试优化选择[J]. 仪器仪表学报, 2015, 36(2): 280 – 286.
MA Ling, LI Hai-jun, WANG Cheng-gang, et al. Optimized test selection method considering the cost of alternative maintenance [J]. Chinese Journal of Scientific Instrument, 2015, 36(2): 280 – 286. (in Chinese)
- [12] BHUSHAN M, RENGASWAMY R. Design of sensor location based on various fault diagnostic observability and reliability criteria [J]. Computers and Chemical Engineering, 2000, 24(2–7): 735 – 741.
- [13] 杨光, 刘冠军, 李金国, 等. 基于故障检测和可靠性约束的传感器布局优化[J]. 电子学报, 2006, 34(2): 348 – 351.
YANG Guang, LIU Guan-jun, LI Jin-guo, et al. Optimal sensor placement based on various fault detectability and reliability criteria [J]. Acta Electronica Sinica, 2006, 34(2): 348 – 351. (in Chinese)
- [14] 朱喜华, 李颖晖, 李宁, 等. 基于改进离散粒子群算法的传感器布局优化设计[J]. 电子学报, 2013, 41(10): 2104 – 2108.
ZHU Xi-hua, LI Ying-hui, LI Ning, et al. Optimal sensor placement design based on improved discrete PSO algorithm [J]. Acta Electronica Sinica, 2013, 41(10): 2104 – 2108. (in Chinese)
- [15] 马羚, 李海军, 王成刚, 等. 基于改进离散粒子群算法的传感器优化配置[J]. 电子学报, 2015, 43(12): 2408 – 2413.
MA Ling, LI Hai-jun, WANG Cheng-gang, et al. Optimal sensor placement based on improved discrete PSO algorithm [J]. Acta Electronica Sinica, 2015, 43(12): 2408 – 2413. (in Chinese)
- [16] ZHANG S G, PATTIPATI K R, HU Z, et al. Optimal selection of imperfect tests for fault detection and isolation [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2013, 43(6): 1370 – 1384.
- [17] LUO J H, TU H Y, PATTIPATI K, et al. Diagnosis knowledge representation and inference [J]. IEEE Instrumentation and Measurement Magazine, 2006, 9(4): 45 – 52.
- [18] SHEPPARD J W, BUTCHER S G W. A formal analysis of fault diagnosis with D-matrices [J]. Journal of Electronic Testing: Theory and Applications, 2007, 23(4): 309 – 322.
- [19] HAN K H, KIM J H. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(6): 580 – 593.
- [20] HAN K H, KIM J H. Quantum-inspired evolutionary algorithms with a new termination criterion, H_c gate, and two-phase scheme [J]. IEEE Transactions on Evolutionary Computation, 2004, 8(2): 156 – 169.
- [21] 蒋荣华, 王厚军, 刘震, 等. 复杂电子系统可测性建模方法研究[J]. 计算机应用研究, 2009, 26(9): 3392 – 3310.
JIANG Rong-hua, WANG Hou-jun, LIU Zhen, et al. Researching on method of modeling complex electronic system for testability [J]. Application Research of Computers, 2009, 26(9): 3392 – 3310. (in Chinese)
- [22] PATTIPATI K R, ALEXANDRIDIS M. Application of heuristic search and information theory to sequential fault diagnosis [J]. IEEE Transactions on System, Man, and Cybernetics (Part A), 1990, 20(4): 872 – 887.
- [23] PLATEL M D, SCHLIEBS S, KASABOV N. Quantum-Inspired evolutionary algorithm: A multimodel EDA [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(6): 1218 – 1232.

作者简介



雷华军 男, 1981 年生于四川通江, 博士, 现为西南科技大学信息工程学院讲师. 研究方向为复杂电子系统的测试性设计及故障诊断.
E-mail: leihuajun2010@163.com



秦开宇 男, 1967 年生于四川遂宁, 博士, 现为电子科技大学航空航天学院教授、博士生导师. 研究方向为复杂电子系统的仿真、测试、验证与评估技术.
E-mail: kyqin@uestc.edu.cn