

# 基于改进粒子群算法的雷达装备测试性设计优化技术

刘丽亚<sup>1</sup>, 杜舒明<sup>2</sup>, 闫俊锋<sup>1</sup>, 商学敏<sup>1</sup>

(1. 北京航天测控技术有限公司, 北京 100041; 2. 中国电子科技集团公司 第十四研究所, 南京 210000)

**摘要:** 针对雷达装备测试性优化设计的实际技术需求, 对雷达测试性优化设计过程进行了分析; 以雷达装备各阶段数据为基础, 综合考虑测试效能、测试代价、可靠性约束等要素, 研究了基于测试代价和测试效能的测试性优化方法, 并给出了基于最小测试代价的雷达系统测试性优化模型; 针对模型多目标优化求解问题, 给出了一种基于改进的粒子群优化算法; 该算法引入混沌理论, 使初始种群呈现多样性, 避免了传统粒子群算法的早熟现象, 同时提高了搜索的精度和速度; 通过对案例的仿真与验证表明, 利用这种改进的粒子群算法对基于最小测试代价的测试性优化模型进行求解时, 能够在满足模型目标函数的约束条件下, 获得全局最优解。

**关键词:** 测试性; 优化; 雷达装备

## Optimization Technology of Radar Equipment Testability Design Based on Improved Particle Swarm Optimization

Liu Liya<sup>1</sup>, Du Shuming<sup>2</sup>, Yan Junfeng<sup>1</sup>, Shang Xuemin<sup>1</sup>

(1. Beijing Aerospace Measurement and Control Technology Co., Ltd, Beijing 100041, China;  
2. 14<sup>th</sup> Research Institute of China Electronic Technology Group, Nanjing 210000, China)

**Abstract:** According to the actual technical requirements of radar equipment testability optimization design, the process of radar testability optimization design is analyzed. Based on the data of each stage of radar equipment and considering the factors of test efficiency, test cost and reliability constraint, the test optimization method based on test cost and test efficiency is studied, and the test optimization model of radar system based on minimum test cost is given. An improved particle swarm optimization algorithm is proposed to solve the multi-objective optimization problem. The chaos theory is introduced into the algorithm, which makes the initial population diversified, avoids the premature phenomenon of traditional particle swarm optimization, and improves the search accuracy and speed. Through the simulation and verification of the case, it is shown that this improved particle swarm optimization algorithm can obtain the global optimization model based on the minimum test cost under the constraints of the objective function of the model.

**Keywords:** testability; optimization; radar equipment

## 0 引言

随着武器装备的复杂性不断地提高, 雷达系统的测试性设计的应用也越来越广泛, 测试性设计已经渗入到雷达系统的设计、研制、验证及应用的各个阶段。如果雷达系统的测试性水平低的话, 系统的故障监测、故障检测及诊断等性能都会存在致命隐患, 严重影响装备的战备完好性, 为了从根本上提高雷达系统的测试性设计能力, 必须研究测试性设计优化技术<sup>[1]</sup>。当前, 测试性设计的优化技术是测试性领域研究热度较高的一门技术学科, 我国正在研发的一些雷达装备都对测试性设计都有迫切需求, 因此攻克雷达装备测试性优化设计的各种关键技术, 注重测试性优化设计的理论和算法研究, 解决该技术的难点问题, 为雷达装备的测试性设计技术发展提供技术支撑, 具有重要的

理论意义和工程价值。

雷达的测试性设计优化技术本质是一个多目标优化问题。20世纪中期出现的粒子群算法是通过仿生学的角度, 根据进化理论, 提出的一种新型优化算法。该算法对解决复杂优化问题, 有着很强的适应性, 在解决优化问题中应用广泛。但是, 标准粒子群算法存在容易陷入局部极值的缺点, 因此, 研究者们为了避免其过早收敛, 提出了各种各样的改进算法。本文在离散粒子群算法的基础上, 引入混沌理论, 通过控制种群的多样性来提高算法的总体性能, 并在实际应用和试验中进行了验证。

## 1 雷达装备测试性优化设计过程分析

测试性是装备的一种固有设计特性, 好的测试性设计能够提高装备的故障检测和故障隔离能力。当前, 测试性设计的优化技术是测试性领域研究热度较高的一门技术学科, 我国正在研发的一些雷达装备都对测试性设计都有迫切需求, 因此攻克雷达装备测试性优化设计的各种关键技术, 注重测试性优化设计的理论和算法研究, 解决该技术

收稿日期: 2020-05-14; 修回日期: 2020-05-28。

作者简介: 刘丽亚(1979-), 女, 河北保定人, 研究生, 工程师, 主要从事测试性与诊断技术相关方向的研究。

的难点问题, 为雷达装备的测试性设计技术发展提供技术支持, 具有重要的理论意义和工程价值。

雷达装备的测试性设计要从装备的研制之初开始, 开展全寿命周期各阶段的测试任务分解和分析, 此过程贯穿于整个寿命周期的各阶段。在优化设计分析过程中, 要综合来自各方面的信息, 包括装备自身的功能、性能、维修保养信息等, 还要考虑测试资源以及测试费用等多种约束条件。可以从宏观和微观角度两个方面来分析:

1) 从宏观角度来分析如图 1 所示。

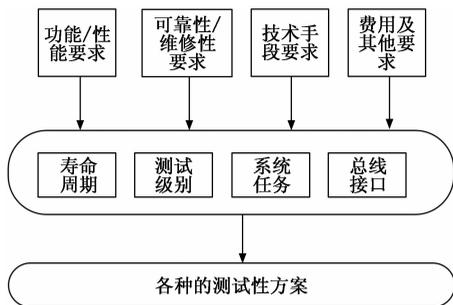


图 1 测试性方案设计内容

从宏观角度对雷达装备的各方面信息和数据进行综合评估和考虑, 针对寿命周期、测试级别、系统任务、总线接口等方面的不同, 综合权衡, 制定出多种情况下适合雷达装备的全寿命周期测试优化方案。

2) 从微观角度来分析如图 2 所示。

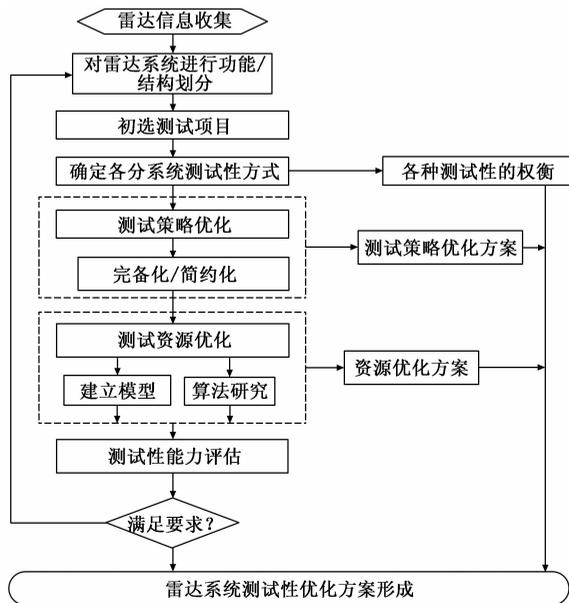


图 2 测试性方案优化设计技术路线

要依据雷达装备的不同情况, 明确不同状态、不同单元测试性方案的设计要求, 然后确定装备测试与诊断具体实施过程, 最后给出测试性方案优化设计技术路线。

一般情况下, 针对雷达装备测试性设计的原则, 是在保证测试指标的情况下, 尽可能的测试代价最小, 或者在保证一定费用上限的情况下, 尽可能多地实现重要的测试

参数。测试覆盖型问题是一个 NP 完全问题, 目前有许多相应的算法可以应用, 也取得了一定的效果, 但是由于这些算法都对模型进行了一定程度的简化, 因此准确性和效率都不太尽人意, 还需要结合具体的问题建立更加完整和准确的模型, 以提高算法的有效性来进行优化搜索。

## 2 基于最小测试代价的雷达系统测试性优化模型

本文以雷达装备各阶段数据为基础, 综合考虑测试效能、测试代价、可靠性约束等要素, 研究基于测试代价和测试效能的测试性优化方法, 根据测试性验证后暴露出来的测试性问题, 在进行测试性设计改进时, 综合考虑, 实现雷达装备的测试性优化。

所谓的测试性优化设计是一个综合考虑功能、性能和可靠性约束的过程, 要从装备的测试性设计一开始就对各组成要素进行权衡, 以得到最优的测试性优化方案。针对雷达系统进行基于测试性效能、测试代价等元素的测试性优化技术, 其基本思路如图 3 所示。

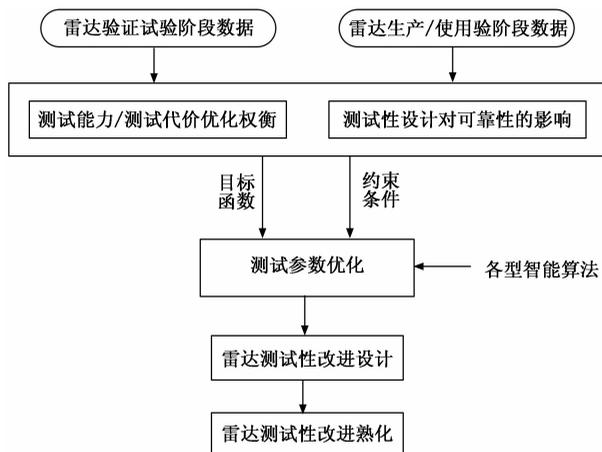


图 3 基于测试效能、测试代价的 BIT 设计优化技术

如图 3 所示, 首先, 根据雷达系统在生产、使用和验证阶段所产生的数据, 综合权衡各测试性设计的测试效能和测试代价, 然后, 利用各种智能型搜索算法, 进行测试性参数集的局部优化, 进而在此基础上进行全局优化, 最后, 以基本可靠性代价以及测试性指标为约束条件, 以雷达系统的综合效能为优化目标, 进行多目标优化权衡, 给出测试性设计的改进方案。

### 2.1 基于测试性/BIT 设计的测试代价与效能量化分析

测试性/BIT 设计本身会引起系统重量增加, 功耗增加, 并且造成基本可靠性降低, 但同时, 测试性/BIT 又可以提高武器装备的测试效能, 因此如何平衡测试性/BIT 设计时测试代价, 基本可靠性和测试性性能提高之间的关系, 使三者达到均衡最优, 会对未来的测试性/BIT 设计工作具有重要的指导意义。

#### 2.1.1 测试代价

设  $C$  表示代价, 是对测试代价、基本可靠性代价、测试效能等要素的统一度量。对于某一测试性/BIT 设计, 假设其测试代价为  $C_i(T_i)$ ,  $T_i$  表示第  $i$  个测试性/BIT, 尺寸

增加带来电路空间、散热等代价  $C_{1a}(T_i)$ ；重量增加带来的设备重量压力，尤其是在机载设备中，需要严格限定重量，根据专家经验设定其重量代价为  $C_{1b}(T_i)$ ；功耗增加带来的功耗代价为  $C_{1c}(T_i)$ 。对于软件部分，代码存储会带来内存的代价  $C_{1d}(T_i)$ 、占用 CPU 会带来系统计算资源代价  $C_{1e}(T_i)$ 。

表 1 测试代价

| 测试点影响 |        | 定量代价                          |
|-------|--------|-------------------------------|
| 硬件    | 尺寸增加   | 电路空间、散热等代价 $C_{1a}(T_i)$      |
|       | 重量增加   | 装备负载负担变大,代价 $C_{1b}(T_i)$     |
|       | 需要激励   | 对激励的需求增加导致功耗增加 $C_{1c}(T_i)$  |
| 软件    | 内存增加   | 代码本身存储代价 $C_{1d}(T_i)$        |
|       | 计算资源增加 | 占用 cpu,拖累系统运行效率 $C_{1e}(T_i)$ |

因此，综合测试  $T_i$  的软硬件部分，可得其测试代价：

$$C_1(T_i) = C_{1a}(T_i) + C_{1b}(T_i) + C_{1c}(T_i) + C_{1d}(T_i) + C_{1e}(T_i) \quad (1)$$

### 2.1.2 基本可靠性代价量化分析

对于测试性/BIT 来说，设其基本可靠性代价为  $C_2(T_i)$ ，由  $T_i$  带来的软硬件结构增加导致平均故障间隔时间变短，造成的维修任务增多带来的代价为  $C_{2a}(T_i)$ ，由于软硬件结构增加带来的周期性维护保障费用代价  $C_{2b}(T_i)$ 。

表 2 基本可靠性代价

| 测试点影响       | 定量代价                           |
|-------------|--------------------------------|
| 组件复杂,故障增多   | 平均无故障时间变短,带来维修代价 $C_{2a}(T_i)$ |
| 结构复杂,保障任务增加 | 常规保障任务增加,带来保障代价 $C_{2b}(T_i)$  |

因此，基本可靠性代价：

$$C_2(T_i) = C_{2a}(T_i) + C_{2b}(T_i) \quad (2)$$

### 2.1.3 测试性性能效益

增加测试性/BIT，将会提高系统故障的检测、诊断与隔离能力，极大缩短故障维护时间，在关键任务执行过程中，当出现重大故障时，及时检测出来能避免重大损失。因此，增加测试性/BIT，带来测试性性能的提高，将产生重大效益。假定其 BIT 测试性能效益为  $C_3(T_i)$ ，其故障检测、诊断能力的提高，能及时发现潜在故障并报警，根据专家经验，带来的潜在效益为  $C_{3a}(T_i)$ ；其故障定位能力的提高，可以极大地减小维修保障代价，带来的经济效益为  $C_{3b}(T_i)$ 。

表 3 测试性性能效益

| 测试点影响        | 效益提高                               |
|--------------|------------------------------------|
| 故障检测率(FDR)提高 | 监测系统状态;及时检测出故障,带来的效益 $C_{3a}(T_i)$ |
| 故障隔离率(FIR)提高 | 快速定位故障,缩短维修诊断时间带来的效益 $C_{3b}(T_i)$ |

则：

$$C_3(T_i) = C_{3a}(T_i) + C_{3b}(T_i) \quad (3)$$

## 2.2 系统优化模型

1) 以测试代价和基本可靠性代价为约束条件，测试性性能效益最优为目的函数：

测试性性能指标用测试性能效益来表示  $C_3(T_i)$ ，因此系统的效能模型为：以测试性性能效益为目标函数，以测试代价  $C_1(T_i)$  和基本可靠性代价  $C_2(T_i)$  为约束，求解目标为：求出一组测试使得在满足  $\sum C_1(T_i)$ 、 $\sum C_2(T_i)$  不大于某个限定值的情况下， $\sum C_3(T_i)$  达到最大。假设测试代价阈值  $D_1$ 、基本可靠性代价阈值  $D_2$ 。因此，优化模型为：

$$\begin{cases} \phi(x) = \sum C_3(T_i) \\ \sum C_1(T_i) \leq D_1 \\ \sum C_2(T_i) \leq D_2 \\ FDR \geq m, FIR \geq n \end{cases} \quad (4)$$

2) 以测试代价、基本可靠性代价和测试性指标为约束，以系统综合效能为目标，建立综合效能模型：

设 BIT 总效益为  $L_s$ 。假设测试代价上限  $D_1$ 、基本可靠性代价上限  $D_2$ ，检测率最小值为  $m$ ，隔离率最小值为  $n$ 。因此，优化模型为：

$$\begin{cases} L_s = \sum C_3(T_i) - \sum C_1(T_i) - \sum C_2(T_i) \\ \sum C_1(T_i) \leq D_1 \\ \sum C_2(T_i) \leq D_2 \\ FDR \geq m, FIR \geq n \end{cases} \quad (5)$$

利用以上两种优化模型，我们可以分别得到基本测试性指标效益最优的测试性设计方案和综合效能效益最高的测试性方案。

## 3 基于改进离散粒子群算法的优化模型求解

### 3.1 基本粒子群算法

在粒子群算法中，粒子的位置对应于原问题的解。粒子的适应值就是将粒子的位置带入到目标函数中所得到的目标函数值。粒子的速度决定粒子下一步向哪里飞以及飞多远。以下是两个粒子群算法中的核心公式：

$$v_{i,j}^{(k+1)} = v_{i,j}^{(k)} + c_1 \cdot r_1 \cdot (pbest_{i,j}^{(k)} - x_{i,j}^{(k)}) + c_2 \cdot r_2 \cdot (gbest_{i,j}^{(k)} - x_{i,j}^{(k)}) \quad (6)$$

$$x_{i,j}^{(k+1)} = x_{i,j}^{(k)} + v_{i,j}^{(k+1)} \quad (7)$$

其中： $x_i$  为粒子当前的位置； $v_i$  粒子当前的速度，为了限制某个粒子的飞翔范围过大，需要为粒子的每一个维度设置飞翔区间  $[-v_{i,max}, v_{i,max}]$ ； $pbest_i$  为粒子本身搜索过的最好的位置； $gbest$  为群体中所有粒子经过的最好位置（群体经验）； $c_1$ 、 $c_2$  为正常数，称为学习因子； $r_1$ 、 $r_2$  为  $[0, 1]$  区间的随机数。

基本粒子群算法具有搜索速度快的优点，但同时也存在容易陷入局部最优的缺点。造成这种情况的原因是  $v_i$  具有随机性且没有记忆能力。因此，在  $v_i$  前引入惯性权重  $w$  参数，通过调整此参数来平衡算法全局搜索和局部搜索之间的矛盾。引入惯性权重后，式 (6) 和式 (7) 调整为如下公式：

$$v_{i,j}^{(k+1)} = \omega \cdot v_{i,j}^{(k)} + c_1 \cdot r_1 (pBest_{i,j}^{(k)} - x_{i,j}^{(k)}) + c_2 \cdot r_2 (gBest_{i,j}^{(k)} - x_{i,j}^{(k)}) \quad (8)$$

$$x_{i,j}^{(k+1)} = x_{i,j}^{(k)} + v_{i,j}^{(k+1)} \quad (9)$$

### 3.2 基于改进离散粒子群算法的优化模型求解

改进离散粒子群算法的思想, 是引入混沌思想, 利用该思想的遍历特性, 初始化粒子的速度和位置。通过提高初始种群的分布多样性, 以及引进惯性权重参数, 使离子能够进行自适应调整, 避免种群的早熟现象, 进而提高算法的收敛速度和精度。

#### 3.2.1 编码过程及混沌初始化

混沌算法具有遍历性、随机性和规律性的特点。利用混沌搜索进行种群初始位置的产生, 可以使种群分布更加均匀, 有利于全局寻优。IDPSO 的编码过程及算法混沌初始化的方式如下:

1) 编码顺序如下:

(1) 通过相关性建模方法建立故障-测试相关性矩阵。其中, 备选测试集数目为  $n$ , 种群中每个粒子对应一个备选测试集, 且每个粒子是由长度为  $n$  的二进制码组成。

(2) 初始化种群  $X = \{x_1, x_2, \dots, x_M\}$ , 其中  $M$  为种群规模。

(3) 最后, 粒子  $x_i$  中的各分量通过下式转换而得到二进制编码向量。

$$x_{ij} = \begin{cases} 1, \epsilon < si \text{ gmoid}(v_{ij}) \\ 0, \epsilon \geq si \text{ gmoid}(v_{ij}) \end{cases} \quad (10)$$

其中:  $\epsilon$  是判断阈值, 这里取  $\epsilon = 0.5$ , 粒子速度  $v_{ij}$  是从区间  $[v_{\min}, v_{\max}]$  随机产生的实数。

2) 混沌初始化:

混沌初始化是利用混沌搜索的方法在速度区间  $[v_{\min}, v_{\max}]$  内产生混沌变量作为种群的初始速度:

- (1) 随机产生一个 (0, 1) 区间的粒子, 作为混沌初值;
- (2) 种群中其他粒子元素进行映射。

$$z_{k+1} = T(z_k) = \begin{cases} 2z_k, & 0 \leq z_k \leq 0.5 \\ 2(1 - z_k), & 0.5 < z_k \leq 1 \end{cases} \quad (11)$$

3) 为防止 Tent 映射迭代到不动点 (例如: 0, 0.25, 0.5, 0.75), 对序列加入小扰动, 使 Tent 映射重新进入混沌状态。如下所示:

$$\text{If } z_k = 0, 0.25, 0.5, 0.75 \text{ or } z_k = z_{k-m} \\ z_{k+1} = T(z_k) + 0.1 \cdot \text{rand}(0, 1); \text{ else then} \\ z_{k+1} = T(z_k).$$

其中:  $z_k$  为混沌变量,  $z_k \in [0, 1]$ ;  $m = \{1, 2, 3, 4, 5\}$ 。

4) 将混沌序列中的每个变量转换为粒子速度中的每个变量。

$$v_{ij} = v_{\min} + z_{ij} \cdot (v_{\max} - v_{\min}) \quad (12)$$

通过上述粒子编码过程和混沌初始化过程, 完成了粒子种群的初始化。

#### 3.2.2 适应度计算方法

计算适应度值的目的是为了评价种群中每个粒子的优劣, 利用启发式规则以及罚函数的方法修正越界粒子, 以

解决测试优化选择问题。

适应度计算流程如图 4 所示。

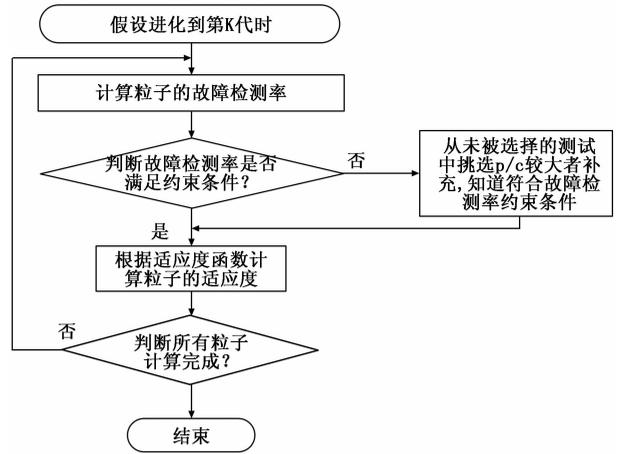


图 4 适应度计算流程

当进化到第  $K$  代时, 计算粒子所选测试集的故障检测率, 当计算的故障检测率大于等于要求的故障检测率时, 根据式 (13) 计算粒子的适应度, 如不满足此约束条件, 则从未被选择的测试集中挑选  $p/c$  较大者进行补充, 直到符合约束条件。依此类推, 直到所有的粒子完成计算。

其中,  $p$  为  $F_i$  中能被相应测试检测的故障概率之和,  $c$  为相应测试的测试代价。选择比值最大的测试加入测试集  $T_i$ , 并将其检测的故障加入故障集  $F_i$ 。

计算  $T_i$  的适应度:

$$Fitness = \frac{\alpha \cdot \sum_{t_i \in T} c_i}{\sum_{t_i \in T} c_i + \sum_{t_i \in T} c_i} - \beta \cdot \max(0, \gamma_{FI}^* - \gamma_{FI}) \quad (13)$$

式中,  $c_i$  表示  $t_i$  的测试代价,  $\alpha$  和  $\beta$  为常数, 且  $\alpha, \beta \in (0, 1)$ 。当测试集  $T_i$  满足故障隔离率指标时, 该个体的适应度由第一部分决定, 当测试集  $T_i$  不满足故障隔离率指标时, 该个体的适应度由第二部分决定, 确保每次利用适应度函数值所选择的个体为当前种群的较优个体。

#### 3.2.3 混沌粒子群算法求解最优解流程

综上所述, 用于测试优化选择的 IDPSO 算法实现流程如下:

- 1) 对  $Popsize, \omega_{\max}, \omega_{\min}, \alpha$  和  $\beta, c_1$  和  $c_2$  以及最大迭代次数  $N_{\max}$  进行参数初始化;
- 2) 种群初始化, 产生初始种群。
- 3) 初始化个体极值以及群体极值, 并计算每个粒子的适应度, 并进而更新个体极值和全局极值。
- 4) 根据自适应策略调整权重, 迭代寻优, 并更新计数器;
- 5) 判断是否遍历了所有粒子, 若全部遍历, 则算法结束, 输出最优测试集; 否则转到 3)。

#### 3.2.4 仿真与验证

为了验证改进的粒子群算法对测试性优化设计的应用的有效性, 我们对雷达装备某单元进行了算法验证。该系统有 15 个先验故障, 20 个备用测试。表 4 为单元故障与可

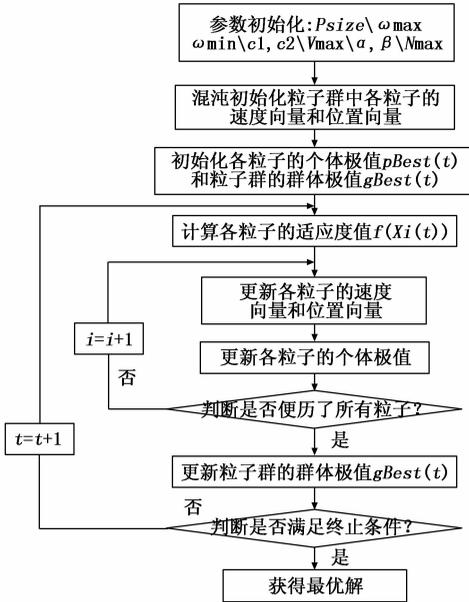


图 6 混沌粒子群算法求解最优解流程图

选测试之间的对应关系, 表 5 给出了故障的先验概率, 表 6 是各测试对应的测试费用。系统测试性设计的目标为: 故障检测率不低于 90%, 故障隔离率不低于 80%, 关键故障检测率为 1, 且费用最小。

表 4 故障与测试对应关系

|          | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ | $t_6$ | $t_7$ | $t_8$ | $t_9$ | $t_{10}$ | $t_{11}$ | $t_{12}$ | $t_{13}$ | $t_{14}$ | $t_{15}$ | $t_{16}$ | $t_{17}$ | $t_{18}$ | $t_{19}$ | $t_{20}$ |   |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|---|
| $f_1$    | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        |   |
| $f_2$    | 1     | 0     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 1 |
| $f_3$    | 1     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0        | 0        | 0        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 0 |
| $f_4$    | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 0 |
| $f_5$    | 1     | 0     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 0 |
| $f_6$    | 0     | 0     | 0     | 1     | 1     | 1     | 1     | 1     | 1     | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 0 |
| $f_7$    | 1     | 0     | 0     | 1     | 1     | 1     | 1     | 1     | 1     | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 0 |
| $f_8$    | 0     | 0     | 0     | 0     | 0     | 1     | 1     | 1     | 1     | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 0 |
| $f_9$    | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 0        | 0 |
| $f_{10}$ | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 0        | 0 |
| $f_{11}$ | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        | 0        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 0        | 0 |
| $f_{12}$ | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        | 0        | 0        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 0        | 0 |
| $f_{13}$ | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        | 0        | 0        | 0        | 1        | 1        | 1        | 1        | 1        | 0        | 0        | 0 |
| $f_{14}$ | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        | 0        | 0        | 0        | 0        | 1        | 1        | 0        | 0        | 0        | 0        | 0 |
| $f_{15}$ | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0        | 0        | 0        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 0        | 0 |

按照式 (5) 对上述系统建立其测试点优化配置模型, 并对算法中各个参数进行设置: 种群规模  $Psize=30$ ; 惯性

表 5 故障先验概率

| $f_1$    | $f_2$    | $f_3$    | $f_4$    | $f_5$    |
|----------|----------|----------|----------|----------|
| 0.0010   | 0.0010   | 0.0010   | 0.0100   | 0.0100   |
| $f_6$    | $f_7$    | $f_8$    | $f_9$    | $f_{10}$ |
| 0.0100   | 0.0100   | 0.0020   | 0.0010   | 0.0100   |
| $f_{11}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ |
| 0.0100   | 0.0025   | 0.0015   | 0.0100   | 0.0100   |

表 6 测试费用

| $t_1$    | $t_2$    | $t_3$    | $t_4$    | $t_5$    | $t_6$    | $t_7$    | $t_8$    | $t_9$    | $t_{10}$ |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 6.0      | 6.6      | 12.0     | 6.0      | 5.2      | 9.0      | 5.0      | 6.0      | 2.0      | 3.6      |
| $t_{11}$ | $t_{12}$ | $t_{13}$ | $t_{14}$ | $t_{15}$ | $t_{16}$ | $t_{17}$ | $t_{18}$ | $t_{19}$ | $t_{20}$ |
| 0.7      | 1.8      | 3.6      | 8.0      | 3.0      | 6.0      | 4.5      | 0.9      | 2.0      | 3.0      |

权重  $\omega_{max}=1.3$ ,  $\omega_{min}=0.4$ ; 学习因子  $c_1=c_2=2$ ; 最大速度  $v_{max}=4$ ; 常数  $\alpha=\beta=0.5$ ; 最大迭代次数  $N_{max}=200$ 。根据模型中的目标函数, 利用本文的改进离散粒子群算法进行求解。所得最优解为:  $[1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1]$ , 即最优完备测试集  $\{t_1, t_2, t_3, t_9, t_{10}, t_{11}, t_{12}, t_{15}, t_{17}, t_{19}, t_{20}\}$ 。此时系统故障检测率 100%, 故障隔离率为 82.2%, 关键故障检测率为 100%, 测试代价为 45.2, 基本上能够在 50 代左右收敛得到最优解, 收敛速度明显加快。

测试优化选择仿真结果表明, 利用该改进算法进行测试性设计优化, 能够满足基于最小测试代价的测试性优化设计的要求, 在综合评价搜索速度、搜索精度、以及成功率的情况下, 该算法能够在满足测试性优化模型中目标函数的要求下, 获得全局最优解。

#### 4 结束语

雷达系统的测试性及诊断能力对其全寿命周期的费用将产生非常可观的影响, 以降低装备全寿命过程总成本为优化目标, 以提高装备的测试性诊断能力为目的, 研究雷达系统的测试性优化技术, 是目前必须要解决的重要课题。

本文给出的基于改进粒子群算法的雷达装备测试性设计优化方法, 在传统粒子群算法的基础上, 增加了混沌搜索的思想, 克服了种群分布不均匀的缺点, 避免了过早陷入局部最优的情况, 并且有效地提高了算法的搜索速度。仿真试验表明, 该方法综合性能好, 对提高雷达装备测试性优化设计具有重要的应用价值。

#### 参考文献:

- [1] 张勇. 装备测试性虚拟验证试验关键技术研究 [D]. 长沙: 国防科学技术大学, 2012.
- [2] 刘晓芹, 黄考利. 故障诊断中传感器配置优化的复杂性分析 [J]. 计算机工程与设计, 2011, 32 (3): 1062-1065
- [3] 汤文超, 李文海, 罗恬颖. 基于费用函数的测试性优化分配方法研究. [J]. 测试技术学报, 2015, 29 (4): 355-358.
- [4] 陈希祥, 邱静, 刘冠军. 装备系统测试性方案优化设计技术研究 [J]. 中国机械工程, 2010, 21 (2): 141-145.
- [5] 严俊豪. 系统级测试性设计优化方法与实现 [D]. 成都: 电子科技大学, 2014.
- [6] 李保来, 吴登勇, 孙超. 基于机载计算机的 BIT 测试优化设计 [J]. 工业控制计算机, 2016, 29 (4): 33-39.
- [7] 王刚. 装备测试性参数优化选择技术研究 [D]. 长沙: 国防科学技术大学, 2010.
- [8] 胡政, 温照森, 钱彦岭. 可测试性设计中的优化问题及求解算法 [J]. 计算机工程与应用, 2000, 11: 42-44.