

引用格式: 范苗, 许一川, 谢定君, 等. 基于离散蝙蝠算法的舰船作战系统测试性分析[J]. 中国舰船研究, 2018, 13(5): 139-146.  
FAN M, XU Y C, XIE D J, et al. Testability analysis of ship combat system based on discrete bat algorithm[J]. Chinese Journal of Ship Research, 2018, 13(5): 139-146.

# 基于离散蝙蝠算法的舰船作战系统测试性分析

范苗<sup>1</sup>, 许一川<sup>2</sup>, 谢定君<sup>1</sup>, 彭辉<sup>2</sup>

1 中国舰船研究设计中心, 湖北 武汉 430064

2 武汉大学 电气工程学院, 湖北 武汉 430072

**摘要:** [目的] 针对舰船作战系统中系统级测试性分析的测试集优化和测试序列优化问题, [方法] 以多信号流图模型为基础, 采用优化精度高、算法收敛速度快的离散蝙蝠算法(DBA)对最优完备测试集和各类故障测试序列进行求解, 并以舰船作战系统为研究对象, 将故障隔离率、平均测试代价和平均期望测试代价等主要指标与现有的优化方案进行仿真对比分析。[结果] 仿真结果表明: 与其他算法相比, 离散蝙蝠算法具有更好的优化效果, 其故障隔离率可提升5.5%、平均测试代价与期望测试代价可分别下降6.35%和17.01%。[结论] 研究成果验证了离散蝙蝠算法应用于系统级测试性分析的可行性和优势, 可为系统级测试性分析工程提供参考。

**关键词:** 舰船作战系统; 多信号流图模型; 离散蝙蝠算法(DBA); 测试集优化; 测试序列优化

中图分类号: U674.7\*03.5; TP301.6

文献标志码: A

DOI: 10.19693/j.issn.1673-3185.01101

## Testability analysis of ship combat system based on discrete bat algorithm

FAN Miao<sup>1</sup>, XU Yichuan<sup>2</sup>, XIE Dingjun<sup>1</sup>, PENG Hui<sup>2</sup>

1 China Ship Development and Design Center, Wuhan 430064, China

2 School of Electrical Engineering, Wuhan University, Wuhan 430072, China

**Abstract:** [Objectives] This paper concerns about test set optimization and test sequence optimization for system-level testability analysis. [Methods] The Discrete Bat Algorithm (DBA) is used to acquire the optimal complete test set and test sequence for each type of fault based on a multi-signal flow graph model. The convergence speed of the algorithm is accelerated while the optimization accuracy is guaranteed. Taking warship combat systems as the research object, such main indexes as fault isolation rate, average test cost and average expected test cost are compared with the existing optimization schemes. [Results] The simulation results show that the proposed method offers better optimization performance compared with other algorithms, among which the fault isolation rate is increased by 5.5% and the average test cost and expected test cost are decreased by 6.35% and 17.01% respectively. [Conclusions] The feasibility and advantages of the DBA applied to system-level testability analysis are verified, providing effective guidance for system testability engineering in the design phase.

**Key words:** ship combat system; multi-signal flow graph model; Discrete Bat Algorithm (DBA); test set optimization; test sequence optimization

## 0 引言

测试性作为装备的一种设计特性,是构成武

器装备质量特性的重要组成部分。优良的测试性设计可以提高装备的战备完好性、任务成功性和安全性,减少维修人力和其他保障资源,降低全寿

收稿日期: 2017-11-15

网络出版时间: 2018-7-17 11:32

基金项目: 国防基础科研计划资助项目(JCKY2016207B005); 国家自然科学基金资助项目(51477121)

作者简介: 范苗,女,1992年生,硕士。研究方向:舰船综合保障工程。E-mail: fanmiao\_zju@163.com

彭辉(通信作者),男,1975年生,博士,副教授。



期费用<sup>[1]</sup>。测试集优化以及测试序列求解是测试性分析的重要研究内容,其正确性和诊断能力的优劣将直接影响到装备维修保障的效率。

对舰船作战系统而言,系统级测试是指针对任务通道的测试,目的是验证作战系统的功能与性能是否满足技术规格书的定义,用以为故障诊断提供信息。目前,舰船作战系统所应用的测试性分析方案主要源自专家经验或启发式函数。随着舰船武器系统的复杂性和集成性的不断提高,舰船设备单元的层级和数量也在大量增加,因此难以保证诊断策略的正确性和优化精度。同时,人工手动测试方法耗时长、测试序列重复度高,其相应的测试性分析结果已无法满足现代舰船系统的维护效率要求。因此,研究适用于舰船作战系统的高效、准确的测试性分析方案以及实用化的工具与技术手段,已成为海军舰船装备能力发展亟待解决的问题。

测试集优化和测试序列求解是测试性分析的关键内容,在减少冗余测试、提高故障检测率和故障隔离率、降低期望测试代价等方面具有重要意义,其正确性和诊断能力的优劣将直接影响装备维修保障的效率。在测试集优化的相关研究中,有关学者提出了基于蚁群算法<sup>[2]</sup>、Rollout算法<sup>[3]</sup>、多目标离散粒子群算法<sup>[4]</sup>、模拟退火算法<sup>[5]</sup>的优化方案,其中前2种方法存在优化目标单一、优化精度不高等缺点,后2种方法存在初始搜索空间较为庞大、因适应度函数构造不唯一导致的搜索效率较低等问题。而测试序列优化是NP完全问题(Non-deterministic Polynomial complete problem),即在测试集优化的前提下,寻求测试代价最小的诊断策略<sup>[6]</sup>。为此,Shakeri等<sup>[7]</sup>提出了基于信息理论概念和启发式函数的AO\*算法,即通过启发式评估函数计算各候选测试的优先程度,自顶而下扩展诊断树节点并同时向上回溯修正。在此基础上,相关学者提出了改进信息熵算法以及一步前向回溯的Rollout算法<sup>[8]</sup>等改进方案,其通过缩小待扩展节点范围、减少回溯次数来降低原有AO\*算法的计算复杂度。然而,这些方法仍然存在高效启发式评价函数的选取问题以及不能保证获得全局最优解的问题。在智能算法方面,基于混合粒子群算法和蚁群算法<sup>[9]</sup>的优化方案已被成功用于求解最优诊断策略。由于适应度函数直接由测试序列所对应的测试代价构造,其优化精度较高、算法运行时间较长,故适用于对计算实时性要求不高的测试性分析。

鉴于此,本文将首先通过评估舰船作战系统初

始相关性矩阵的隔离能力,确定最小完备测试集所包含的测试数目,构造唯一的适应度函数,从而采用离散蝙蝠算法实现测试集的优化工作;然后,将对优化结果采用离散蝙蝠算法,得到比其他算法平均期望测试代价更小的测试序列,用以为测试性分析方案提供新的思路。

## 1 舰船作战系统的测试性建模

舰船作战系统是通过警戒、跟踪、目标识别、数据处理、威胁估计以及控制武器实现对敌作战功能的各要素综合体<sup>[10]</sup>。作为舰船的核心,作战系统集成了大量的先进电子武器装备<sup>[11]</sup>,包括雷达系统、作战指挥系统、武器系统和综合通信系统等分系统。本文将采用以多信号流图模型为基础的测试性建模理论,选取作战系统中较为典型的故障模式来表述各故障模式之间的关联并进行测试性的初步分析。1994年,美国康涅狄格大学的Pattipati教授和Somnath教授提出了多信号流图模型。该方法基于研究对象的结构和功能,以分层有向图的形式来表示信号流向与各功能单元之间的连接关系,即通过定义信号与组成单元所包含的故障模式、测试点所包含的测试信息,来表示系统组成、故障、信号以及测试点之间相关性的一种模型化表示方法<sup>[12]</sup>。

多信号流图模型的主要建模过程<sup>[12]</sup>如下:

1) 熟悉建模对象,明确系统的组成结构,识别和提取模型信息,主要包括系统各组成单元和功能单元的输入/输出信号、故障模式、故障率信息和测试信息等。

2) 构建多信号流图模型。首先,根据系统组成建立系统的结构模型(原理图或概念框图);然后,将功能信号添加至系统各组成单元,并根据信号流向确定组成单元的输入、输出以及各单元之间的连接关系,从而建立有向图;最后,添加测试点位置以及测试点所包含的测试信息,并设置模型中各节点的属性。

3) 对模型的测试性能进行初步评估,并结合工程实际对模型进行调整、校验和修改。

本文利用TesLab软件提供的测试性建模与分析工具,对系统的测试性水平进行分析和评估,并选取20种典型的故障模式来完成舰船作战系统的多信号流图建模。故障模式模型和相关性矩阵分别如图1和表1所示。表1中, $F_0$ 为无故障状态, $F_1 \sim F_{20}$ 分别为图1中的20种故障模式, $t_1 \sim t_{18}$ 为测试性建模过程中初步选定的测试方案。表1中的故障概率数据来源于各二级系统的

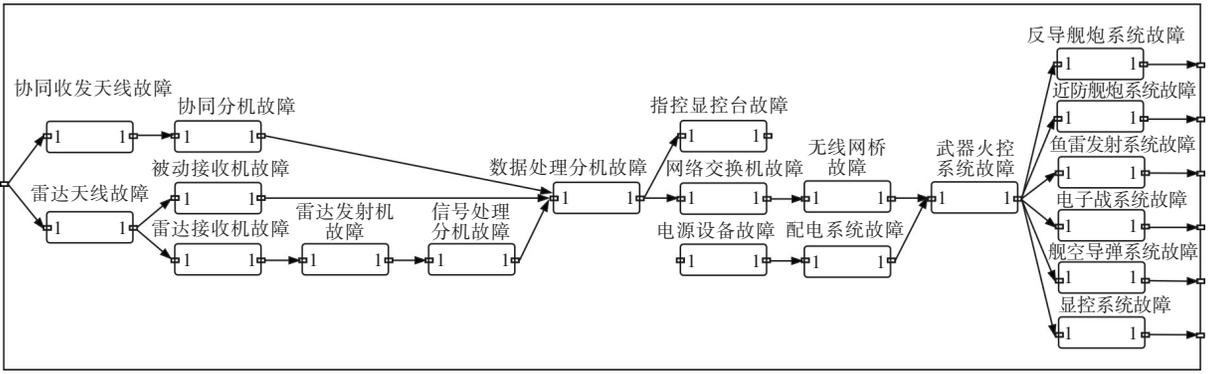


图1 舰船作战系统的故障模式模型图

Fig.1 Illustration model of ship combat system failure mode

表1 故障—测试的依赖关系矩阵  
Table 1 Dependency matrix of fault-test

故障模式	所选测试及其代价(上方为测试方案序号,下方为测试所对应的代价)																		故障概率
	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$	$t_{12}$	$t_{13}$	$t_{14}$	$t_{15}$	$t_{16}$	$t_{17}$	$t_{18}$	
$F_0$	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	0.25
$F_1$	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0.006 5
$F_2$	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.001 5
$F_3$	0	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	1	0.006 5
$F_4$	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0.006 5
$F_5$	0	1	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	1	0.001 5
$F_6$	0	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	1	0.001 5
$F_7$	0	1	0	0	0	0	1	1	1	0	0	0	1	1	0	0	0	1	0.006 5
$F_8$	0	1	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	1	0.006 5
$F_9$	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0.076
$F_{10}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.2
$F_{11}$	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	0.001 5
$F_{12}$	0	1	0	0	1	1	1	1	1	0	1	0	1	1	0	0	0	1	0.006 5
$F_{13}$	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0.006 5
$F_{14}$	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0.006 5
$F_{15}$	0	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	1	0.2
$F_{16}$	0	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	1	0.001 5
$F_{17}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0.006 5
$F_{18}$	0	1	0	0	0	0	1	1	1	0	1	0	0	1	0	0	0	1	0.2
$F_{19}$	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	1	0.006 5
$F_{20}$	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0.001 5

设计数据以及装备实际应用的统计数据(本文中数据仅为示例,并非实船数据)。

## 2 离散蝙蝠算法

2010年,提出了一种新的元启发式算法,即蝙蝠算法<sup>[13]</sup>(Bat Algorithm, BA),其优化机理是模拟自然界微型蝙蝠躲避障碍物并寻找目标的回声定位原理。针对这种已成功应用于连续域函数优化问题的随机搜索优化算法,为了对其进行改进以解决组合优化问题,文献[14-16]以旅行商(TSP)问题为例提出了离散蝙蝠算法(Discrete

Bat Algorithm, DBA),仿真结果表明离散蝙蝠算法在寻优能力、偏差率、收敛率、时间耗费等指标上的优势远超其他传统算法。由于测试性分析中的测试序列优化问题仅与旅行商问题的目标函数不同,因此可以采用DBA来完成测试执行顺序的优化求解,并可将该方法应用于测试集优化问题。

在离散蝙蝠算法中,每个蝙蝠的位置表示待优化问题的一个可行解。对于种群规模为  $m$  的蝙蝠而言,蝙蝠个体  $i(i=1, 2, \dots, m)$  发出声波的频率  $f(i)$  为

$$f(i) = f_{\min} + (f_{\max} - f_{\min})\beta \quad (1)$$

式中:  $f_{\max}$ ,  $f_{\min}$  分别为声波频率的最大值和最小值;  $\beta \in [0, 1]$ , 为随机数。

设蝙蝠个体  $i$  在  $t$  时刻的速度矩阵和位置向量分别为  $\mathbf{v}_t(i)$ ,  $\mathbf{x}_t(i)$ , 则蝙蝠个体  $i$  在  $t+1$  时刻的速度  $\mathbf{v}_{t+1}(i)$  和位置  $\mathbf{x}_{t+1}(i)$  为:

$$\mathbf{v}_{t+1}(i) = \mathbf{v}_t(i) + [\mathbf{x}_t(i) - \mathbf{x}_*]f(i) \quad (2)$$

$$\mathbf{x}_{t+1}(i) = \mathbf{x}_t(i) + \mathbf{v}_{t+1}(i) \quad (3)$$

式中:  $\mathbf{x}_*$  为  $t$  时刻搜索过程中蝙蝠的全局最优位置向量。

式(2)中, 向量  $\mathbf{x}_t(i)$  与向量  $\mathbf{x}_*$  的差值定义如下:

设  $\mathbf{x}_t(i) = (x_1, x_2, \dots, x_n)$ ,  $\mathbf{x}_* = (x_{*1}, x_{*2}, \dots, x_{*n})$ , 其中  $n$  为蝙蝠搜索空间的维度; 设  $\mathbf{v}_d$  为第  $i$  个蝙蝠的位置差, 初始化矩阵  $\mathbf{v}_d$  为空集。逐一检查  $\mathbf{x}_t(i)$ ,  $\mathbf{x}_*$  的每个分量, 若  $x_n - x_{*n} \neq 0$ , 则将向量  $(x_n, x_{*n})$  作为矩阵  $\mathbf{v}_d$  的第 1 列, 并以此方式逐列叠加。

式(2)中,  $\mathbf{v}_{t+1}(i)$  的定义如下:

设速度向量  $\mathbf{v}_t(i)$  为  $n \times 2$  的整数矩阵, 且矩阵元素为  $1 \sim n$  的自然数; 设  $\mathbf{v} = \mathbf{v}_t(i) \cup \mathbf{v}_d$ ,  $l = \text{round}[f(i) \cdot n] + 1$ , 其中  $\text{round}$  函数表示向下取整。从矩阵  $\mathbf{v}$  中随机抽取  $l$  行, 即可得到蝙蝠个体  $i$  在  $t+1$  时刻的速度向量  $\mathbf{v}_{t+1}(i)$ 。

式(3)中,  $\mathbf{x}_{t+1}(i)$  的定义如下:

根据式(2)可知  $\mathbf{v}_{t+1}(i)$  为  $l \times 2$  的矩阵, 设矩阵  $\mathbf{v}_{t+1}(i)$  中第 1 列元素为  $p$  和  $q$ , 交换向量  $\mathbf{x}_t(i)$  中第  $p$  个元素和第  $q$  个元素的位置, 如此逐列操作直至  $\mathbf{v}_{t+1}(i)$  的第  $l$  列, 即可得到  $t+1$  时刻蝙蝠个体  $i$  的位置向量  $\mathbf{x}_{t+1}(i)$ 。

在获取目标的过程中, 为了均衡局部搜索和全局搜索, 蝙蝠需要调节发射声波的响度和频率。设蝙蝠个体  $i$  在  $t$  时刻发出声波的响度为  $A_t$ , 初始声波响度为  $A_0$ , 脉冲发射速率为  $r_t$ , 则蝙蝠个体  $i$  在  $t+1$  时刻的声波响度  $A_{t+1}$  和脉冲发射速率  $r_{t+1}$  为:

$$A_{t+1} = \alpha A_t \quad (4)$$

$$r_{t+1} = r_0 [1 - \exp(-\gamma t)] \quad (5)$$

式中:  $\alpha \in [0, 1]$ , 为声波响度增强系数;  $r_0$  为初始脉冲发射速率;  $\gamma$  为脉冲发射速率衰减系数, 是一个常量。

局部搜索的主要步骤如下:

1) 在  $[0, 1]$  范围内生成一个随机数  $\text{rand}_1$ , 判断  $\text{rand}_1 > r_t$  是否成立。若成立, 则对当前位置施加随机扰动, 得到新解  $\mathbf{x}_{\text{new}}$ , 并计算新解的函数适

应值  $\text{fitness}(\mathbf{x}_{\text{new}})$ , 其中  $\text{fitness}$  为适应度函数。

2) 在  $[0, 1]$  范围内生成另一个随机数  $\text{rand}_2$ , 判断  $\text{rand}_2 > A_t$  和  $\text{fitness}(\mathbf{x}_{\text{new}}) < \text{fitness}[\mathbf{x}_t(i)]$  是否同时成立。如果成立, 则接受新解  $\mathbf{x}_{\text{new}}$ , 同时按照式(4)和式(5)更新  $A_t$  和  $r_t$ 。

3) 重复上述步骤不断更新蝙蝠位置向量, 直至达到最大迭代次数并输出最优解。

### 3 测试集优化

首先, 通过建立系统的多信号流图模型来得到相关性矩阵; 然后, 合并模糊组并删除冗余测试。以构建诊断用测试集为例, 测试集优化的目的是找到满足系统所要求故障隔离率的测试集, 并在满足最大故障隔离率的条件下令测试集中的测试数目最少、测试代价最低, 其中所优化的目标按重要性依次为最大故障隔离数、最小测试数目和最低平均测试代价。对于  $n$  维蝙蝠搜索空间, 初始测试集中的测试数目即为  $n$ 。本文采用二进制编码来表示相应测试的选取操作, 其中编码 1 表示选取相应序号的测试, 编码 0 表示不选取相应序号的测试。

#### 3.1 最小测试数目

在测试集寻优过程中, 为减少离散蝙蝠算法的迭代次数, 首先需要确定最优测试集中所包含的测试数量。通过分析系统初始相关性矩阵, 可以得到所设计的测试集合能隔离的最大故障数目  $I_{\max}$ 。理论上,  $n$  个测试的最大故障隔离数为  $2^n$ , 当  $2^n < I_{\max}$  时, 所选择的测试数目不可能使系统故障隔离率最大, 故需添加测试数目再次进行测试性分析。

本文采用链长为 1 的模拟退火方式, 按概率接受对测试进行扰动后的故障隔离数目。采用的扰动方式为三变换法, 即任选序号  $U, V, W$  ( $U < V < W \leq n$ ), 将序号  $U, V$  及其之间的二进制编码插入序号  $W$  之后。同时, 与传统优化问题中为避免陷入局部最优而进行的局部搜索步骤不同, 该算法的最大故障隔离数目已经确定, 因此当找到最优解后无需再添加随机扰动, 直接输出测试数目即可。求解最小测试数目的流程如图 2 所示。其中:  $N$  为采用信息熵算法求得的最优完备测试集中所包含的测试数目, 用以防止算法避开最优解;  $I_1$  为随机抽取产生的测试集的故障隔离数目;  $I_2$  为扰动后所选取测试集的故障隔离数目;  $e$  为经测试集优化后所得的最小完备测试集中的测试数目。

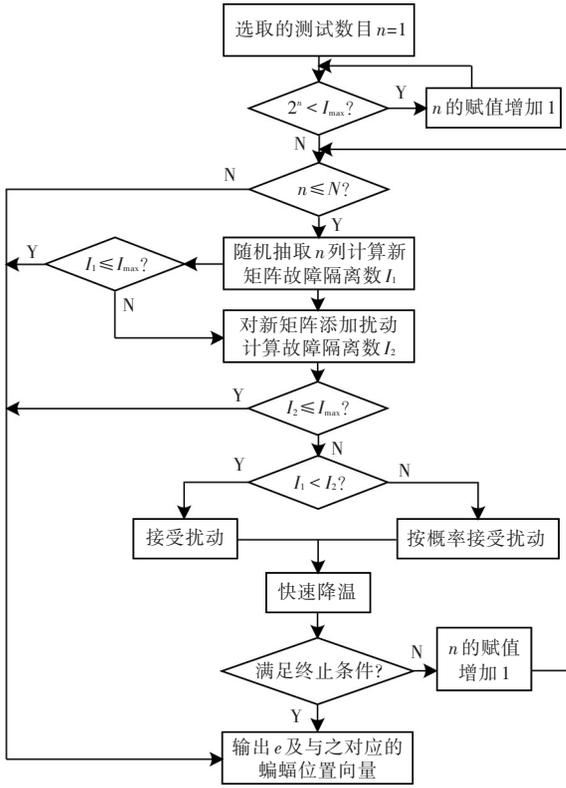


图2 确定最小测试数目的实现流程图

Fig.2 The realization flow chart for determining the minimum number of tests

### 3.2 基于离散蝙蝠算法的测试集优化

确定最小完备测试集的测试数目之后,需要构造合理的适应度函数并依据重要性来优化故障隔离数目和平均测试代价。本文采用统一的适应度函数以避免使用条件嵌套语句,从而减少程序的运行时间。

$$\frac{1}{I_{\max} - 1} - \frac{1}{I_{\max}} > \frac{1}{X}(c_{\max} - c_{\min}) \quad (6)$$

式中:  $c_{\max}$  为所选的测试代价列向量中  $n$  个最大测试代价之和;  $c_{\min}$  为所选的测试代价列向量中  $n$  个最小测试代价之和;  $X$  为使不等式成立的任一正实数。

为了优先考虑所选测试的故障隔离能力,式(6)实际上表示了2种最不理想的情况:一是故障隔离数目和平均测试代价同时达到最大;二是故障隔离数目第2大,平均测试代价最小。当找到合适的  $X$  值时,适应度函数可以简化为

$$fitness[x_i(i)] = \frac{1}{I} + \frac{1}{X} x_i(i)C \quad (7)$$

式中:  $fitness[x_i(i)]$  为适应度函数值;  $I$  为所选测试集的故障隔离数目;  $C$  为测试代价列向量。

采用离散蝙蝠算法时,由于二进制编码中蝙蝠位置向量的减法操作与TSP问题有所不同,故需要对第2节中关于蝙蝠位置向量差的操作进行

重新定义。设第  $i$  个蝙蝠的位置向量为  $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ , 当前全局最优的蝙蝠位置向量为  $x_* = (x_{*1}, x_{*2}, \dots, x_{*n})$ 。定义第  $i$  个蝙蝠位置向量的减法操作为:分别记录向量  $x_i$  和  $x_*$  每个分量中编码为0的序号集合矩阵  $x_{i0}$  和  $x_{*0}$ , 则第  $i$  个蝙蝠的位置向量差为  $v_d = [x_{i0}; x_{*0}]'$ 。同时,在初始解空间中加入确定最小测试数目过程中最终输出的蝙蝠位置向量,以进一步避免算法陷入局部最优,而其他操作定义、算子设计和算法步骤则与基本离散蝙蝠算法保持一致。

### 3.3 仿真对比分析

本文将以前舰船作战系统测试性模型为研究对象,对初始测试集进行优化。优化算法中的相关参数为:在确定最小测试数目的算法中,设初始温度  $T_0 = 1$ , 降温速率为0.99, 终止温度为  $6.6 \times 10^{-3}$ 。在离散蝙蝠算法中,设蝙蝠种群规模  $m = 10$ ,  $f_{\min} = 0$ ,  $f_{\max} = 1$ ,  $A_0 = 1.6$ ,  $r_0 = 0.0001$ 。由于测试集优化问题的复杂度较低,  $\alpha$  和  $\gamma$  的取值对优化结果的影响很小,所以根据经验取  $\alpha = 0.9$ ,  $\gamma = 0.9$ , 迭代次数  $Maxgen = 100$ 。将基于离散蝙蝠算法的测试集优化结果分别与混合粒子群—遗传算法(HPSOGA)、改进离散粒子群优化(DPSO)算法、信息熵算法、Rollout算法、蚁群优化(ACO)算法、最大/最小蚁群优化(MMACO)算法和模拟退火算法(SAA)进行比较,对各算法分别运行50次,仿真结果取平均值。其中各算法迭代次数与本文算法保持一致,均为150。

期望测试代价统一采用离散蝙蝠算法,迭代次数为  $Maxgen' = 300$ 。优化过程中的相关统计指标如表2所示。

从仿真结果可知,与其他主流算法相比,本文采用的优化算法在平均故障隔离数、平均测试个数、期望测试代价和最优比例方面均取得了更为理想的结果。其中故障隔离率最多可提高5.5%,平均测试个数、平均测试代价最多可分别下降1.4%和6.35%,达到最优的比例可增至86%。同时,在群智能优化算法中,本文算法的平均收敛时间也较短。仿真结果表明,本文算法可以成功应用于测试集优化问题。

## 4 测试序列优化

### 4.1 优化问题描述

在仅考虑发生单一故障的前提下,得到相关性矩阵并对测试集优化后,需要根据可靠性求解

表2 诊断测试集优化结果比较

Table 2 Comparison of test set optimization for fault diagnosis

算法类型	平均故障 隔离数	平均测试 个数	平均测试 代价	期望测试 代价	达到最优 的比例/%	最优测试集 (已删除冗余测试)	算法平均 收敛时间/s
HPSOGA	19.88	15.14	61.12	12.328	4	$\{t_1, t_4, t_5\}$	5.868
改进DPSO算法	21	15.22	60.96	12.216	32	$\{t_1, t_4, t_{17}\}$	2.964
信息熵算法	21	15	63	12.250 5	0	$\{t_{16}, t_{17}, t_{18}\}$	0.178
Rollout算法	21	15	62	12.253	0	$\{t_3, t_{17}, t_{18}\}$	1.012
ACO算法	20.94	15.2	60.98	12.216	40	$\{t_1, t_4, t_{17}\}$	3.641
MMACO算法	20.90	15.14	61.28	12.225 5	52	$\{t_1, t_{17}, t_{18}\}$	3.782
SAA	20.96	15.16	60.42	12.216	58	$\{t_1, t_4, t_{17}\}$	1.697
本文算法	21	15	59.57	12.216	86	$\{t_1, t_4, t_{17}\}$	1.536

测试代价最优的诊断策略。其中,各组成单元的可靠性数据(故障率或故障概率)可从可靠性设计分析资料中获取,而与测试相关的代价则应考虑测试时间、测试点设计费、研制费用和实施测试所需的费用等。因此,测试序列优化的目标即令隔离系统故障状态所需的总期望测试费用  $J$  最小,其适应度函数为

$$J = PIC = \sum_{b=0}^a \left\{ \sum_{j=1}^{|p_b|} c_{p_{b(j)}} \right\} p(F_b) \quad (8)$$

式中:  $\mathbf{P}$  为故障概率向量;  $\mathbf{I}$  为故障隔离矩阵<sup>[17]</sup>;  $a$  为故障模式数量;  $b$  为故障模式序号,  $b=0, 1, \dots, a$ ;  $|p_b|$  为隔离故障模式  $F_b$  所需的测试序列  $p_b$  的长度;  $j$  为测试序列  $p_b$  的测试序号,  $j=1, 2, \dots, |p_b|$ ;  $c_{p_{b(j)}}$  为  $p_b$  所对应测试序号  $j$  的代价值;  $p(F_b)$  为故障模式  $F_b$  发生的概率。

在测试性设计工程中,测试序列优化的目的是找到一组合适的测试执行顺序,使得期望测试费用最低,用以为装备维修保障提供支撑。

## 4.2 算法基本步骤

基于离散蝙蝠算法的测试序列优化的基本步骤如下:

1) 设蝙蝠种群的规模为  $m$ , 随机生成初始蝙蝠位置集合  $\mathbf{S}$ , 采用反向学习机制求出反向的蝙蝠位置集合  $\mathbf{S}^*$ 。根据适应值大小对集合  $\mathbf{S}$ ,  $\mathbf{S}^*$  中的位置进行优劣排序, 取适应度函数值前  $m$  位的蝙蝠位置作为初始解集。通过将初始解较为均匀地集中在优质解区域中, 可以在保证优化精度的同时减少算法运行时间。

2) 参数初始化。设置初始脉冲发射频率  $r_0$ 、初始声波响度  $A_0$ 、声波频率的最大值  $f_{\max}$  和最小值  $f_{\min}$ 、声波响度增强系数  $\alpha$ 、脉冲发射速率衰减系数  $\gamma$ 、算法迭代次数  $Maxgen$ 。设随机生成的第  $i$  个蝙蝠速度初始矩阵为  $\mathbf{v}_0(i)$ , 其中  $\mathbf{v}_0(i)$  是维度为

$n \times 2$  的矩阵, 其元素取值范围为自然数  $1 \sim n$ 。

3) 按照式(2)和式(3)分别更新每个蝙蝠的速度与位置。

4) 在  $[0, 1]$  范围内生成一个随机数  $rand_1$ , 判断  $rand_1 > r_t$  是否成立。若成立, 则对当前位置采用3变换法施加随机扰动, 产生新解  $\mathbf{x}_{\text{new}}$ , 并计算新解对应的函数适应值  $fitness(\mathbf{x}_{\text{new}})$ 。

5) 在  $[0, 1]$  范围内生成另一个随机数  $rand_2$ , 若  $rand_2 > A_t$  且适应度函数满足  $fitness(\mathbf{x}_{\text{new}}) < fitness[\mathbf{x}_i]$ , 则接受收步骤4)中产生的新解  $\mathbf{x}_{\text{new}}$ , 并按式(4)和式(5)更新  $A_t$  和  $r_t$ 。

6) 更新当前时刻最优解  $\mathbf{x}_*$ , 判断迭代终止条件是否满足。若不满足, 则转至步骤3); 若满足, 则停止迭代计算过程。

7) 输出最优解。

## 4.3 仿真对比分析

完成测试集优化后, 即可采用基于离散蝙蝠算法的优化方案来确定最优测试序列, 离散蝙蝠算法的相关参数为: 蝙蝠种群规模  $m=10$ ,  $f_{\min}=0$ ,  $f_{\max}=1$ ,  $A_0=1.5$ ,  $r_0=0.000 1$ ,  $\alpha=0.9$ ,  $\gamma=0.99$ ,  $Maxgen=300$ 。其中参数  $\alpha$  和  $\gamma$  的取值由仿真实验确定, 分别在  $0.9 \sim 1$  的范围内每隔  $0.01$  取值做  $100$  组实验。仿真结果表明, 当  $\alpha=0.9$ ,  $\gamma=0.99$  时, 算法运行结果最优。将基于离散蝙蝠算法的优化结果分别与基于信息熵、改进信息熵方案、Rollout算法、HPSO算法、ACO算法、MMA-CO算法进行对比, 对各智能算法分别运行  $50$  次, 其平均期望测试代价、最优期望测试代价、最优测试序列结果、算法平均收敛时间、平均偏差和收敛率指标的对比结果如表3所示。其中, 平均偏差为算法的最终运算结果与其最优结果之差的平均值, 收敛率为算法运行结果成功收敛至本算法最优期望测试代价的比例。



- [5] 姜伟, 王宏力, 何星. 基于模拟退火算法的实时故障诊断测试集优化[J]. 兵工自动化, 2012, 31(7): 78-81.  
JIANG W, WANG H L, HE X. Test set optimization of real-time fault diagnosis system based on simulated annealing algorithm [J]. Ordnance Industry Automation, 2012, 31(7): 78-81 (in Chinese).
- [6] HARTMANN C R P, VARSHNEY P, MEHROTRA K, et al. Application of information theory to the construction of efficient decision trees[J]. IEEE Transactions on Information Theory, 1982, 28(4): 565-577.
- [7] SHAKERI M, RAGHAVAN V, PATTIPATI K R, et al. Sequential testing algorithms for multiple fault diagnosis [J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2000, 30(1): 1-14.
- [8] TU F, PATTIPATI K. Rollout strategies for sequential fault diagnosis [C]//Proceedings of the IEEE AUTOTESTCON. Huntsville, AL, USA: IEEE, 2002: 269-295.
- [9] 叶晓慧, 王红霞, 程崇喜. 基于蚁群算法的系统级序贯测试优化研究[J]. 计算机测量与控制, 2010, 18(10): 2224-2227.  
YE X H, WANG H X, CHENG C X. Test sequencing optimization for system-level based on ant algorithm [J]. Computer Measurement and Control, 2010, 18(10): 2224-2227 (in Chinese).
- [10] 潘镜芙, 董晓明. 水面舰艇作战系统的回顾和展望[J]. 中国舰船研究, 2016, 11(1): 8-12.  
PAN J F, DONG X M. Review and prospect of the combat system for surface combatant ships [J]. Chinese Journal of Ship Research, 2016, 11(1): 8-12 (in Chinese).
- [11] 董晓明. 新一代水面舰艇作战系统发展理念及途径[J]. 中国舰船研究, 2015, 10(1): 1-6.  
DONG X M. Development concepts and approaches of the next generation combat system for surface combatant ships [J]. Chinese Journal of Ship Research, 2015, 10(1): 1-6 (in Chinese).
- [12] DEB S, PATTIPATI K R, RAGHAVAN V, et al. Multi-signal flow graphs: a novel approach for system testability analysis and fault diagnosis [J]. IEEE Aerospace and Electronic Systems Magazine, 1994, 10(5): 14-25.
- [13] YANG X S. A new metaheuristic bat-inspired algorithm [M]//GONZÁLEZ J R, PELTA D A, CRUZ C, et al. Nature inspired Cooperative Strategies for Optimization. Berlin, Heidelberg: Springer, 2010: 65-74.
- [14] 张酌远, 张志忠, 王炼红. 改进蝙蝠算法及其在商旅问题的应用[J]. 电脑与信息技术, 2015, 23(5): 9-11, 33.  
ZHANG Z Y, ZHANG Z Z, WANG L H. Improved bat algorithm and its application to traveling salesman problem [J]. Computer and Information Technology, 2015, 23(5): 9-11, 33 (in Chinese).
- [15] 李枝勇, 马良, 张惠珍. 求解最小比率旅行商问题的离散蝙蝠算法[J]. 计算机应用研究, 2015, 32(2): 356-359.  
LI Z Y, MA L, ZHANG H Z. Discrete bat algorithm for solving minimum ratio traveling salesman problem [J]. Application Research of Computers, 2015, 32(2): 356-359 (in Chinese).
- [16] 戚远航, 蔡延光, 蔡颖, 等. 旅行商问题的混沌混合离散蝙蝠算法[J]. 电子学报, 2016, 44(10): 2543-2547.  
QI Y H, CAI Y G, CAI H, et al. Chaotic hybrid discrete bat algorithm for traveling salesman problem [J]. Acta Electronica Sinica, 2016, 44(10): 2543-2547 (in Chinese).
- [17] 张延生, 乔中涛, 井建辉. 基于广度优先搜索算法的故障隔离矩阵生成方法[J]. 计算机测量与控制, 2016, 24(8): 16-20.  
ZHANG Y S, QIAO Z T, JING J H. Generating fault isolation matrix based on breadth first search algorithm [J]. Computer Measurement and Control, 2016, 24(8): 16-20 (in Chinese).