

基于改进蚁群的测试序列优化算法

李丹阳, 蔡金燕, 杜敏杰, 朱 赛

(军械工程学院光学与电子工程系, 河北 石家庄 050003)

摘要:针对故障诊断中的测试序列优化问题,提出一种改进蚁群算法的解决方法。该方法根据二值属性系统的特点,定义状态集向量及测试向量,将故障测试隔离过程转化为向量的位运算过程,将序列优化问题转化为一种最小代价的动态树构造问题,设计灵活的状态转移规则,并根据动态树的分层结构特点,提出一种分层加权和遗传变异相结合的信息素更新策略,解决这种动态树结构的寻优问题。仿真结果表明:该算法以较高的效率收敛到已知最优解,高效实用,为大规模复杂系统的测试优化问题提供了一条新的解决途径,具有一定的工程应用价值。

关键词:测试序列优化;蚁群算法;二值属性系统;动态树

中图分类号:TP277;TP391.9;TM571.62;O212.6

文献标志码:A

文章编号:1674-5124(2013)04-0105-04

Test sequencing optimization based on improved Ant Algorithm

LI Dan-yang, CAI Jin-yan, DU Min-Jie, Zhu Sai

(Department of Electronic and Optical Engineering, Ordnance Engineering College, Shijiazhuang 050003, China)

Abstract: For solving the problem of test sequencing optimization in fault diagnosis, an improved ant algorithm was presented in this paper. According to the feature of the binary attribute system, state-set vector and test-set vector were defined, the fault testing segregation process was transformed to the process of vector operation and the problem of test sequencing optimization was transformed to construct a dynamic tree with minimum cost. Transfer rule of the ant state was designed and a kind of layered weighted pheromone update mechanism was presented, which combine with the variation in GA and solved the optimization problem of tree-construction. Simulation results show that the algorithm convergences to the optimal solution with high efficiency and provides a new way to solve the test sequencing optimization for large-scale complicated system.

Key words: test sequencing optimization; ant algorithm; binary attribute system; dynamic tree

0 引言

测试序列优化问题是在考虑系统中各种故障发生的先验概率和各种测试代价的情况下,设计一组最优的测试序列,使其满足故障隔离精度的要求,并且整体的测试代价最小。一般将测试序列优化问题看作一种最小代价动态树的构造问题,该问题已被证明是 NP-complete 问题,传统的全局寻优算法在解决这一问题时因为计算量过大而难以在实际中应

用。在实际的测试诊断过程中,考虑到时间和测试代价,刻意追求一种全局最优的测试方案并无太大意义;因此,采用近似算法来求解最优测试序列成为解决测试序列优化问题的研究热点。

目前主要的解决方法是以信息熵概念为基础的各种启发式方法^[1-6]。Shakeri 等人基于信息熵概念,采用霍夫曼编码方法构建启发函数,提出 AO 算法及其改进的 AO* 算法;Tu Fang 提出了一种基于信息熵的 Rollout 启发式搜索算法,提高了算法的效率和优化质量;黄以锋将 Rollout 算法扩展到多值属性系统,对算法性能进行了分析;杨成林提出了一种将测试节点优选和测试序列优化相结合的解决方案。

收稿日期:2012-05-02;收到修改稿日期:2012-07-11

基金项目:河北省重点基础研究项目(10963529D)

作者简介:李丹阳(1987-),男,河南濮阳市人,硕士研究生,专业方向为电子系统性能检测与故障诊断。

以上提出的各种方法都是基于信息熵概念的启发式搜索算法,由于太过于依赖启发函数,对于问题内在信息的挖掘不够,难以保证找到全局最优解。针对以上问题,根据二值属性系统的特点,定义了状态集向量及测试向量,将测试隔离过程表示为向量的位运算过程,在此基础上提出一种改进的蚁群算法,利用蚁群搜索过程中的反馈信息,充分挖掘问题本身的信息特征,制定了相应的状态转移规则,并提出了分层加权的信素更新策略,引入遗传算法中的变异方法,提高了搜索效率和优化质量。

1 问题的描述

测试序列优化问题由以下要素组成:

(1)系统的故障状态由一个有限集 F 表示。 $F = \{f_0, f_1, \dots, f_m\}$, 其中 f_0 通常可表示为系统无故障状态, f_i ($0 \leq i \leq m$) 表示 m 种故障状态中的第 i 种故障状态。

(2)系统中各故障状态发生的先验概率为 P 。 $P = [p(f_0), \dots, p(f_m)]^T$, 其中 $p(f_i)$ ($0 \leq i \leq m$) 表示故障状态 f_i 发生的先验概率。

(3)系统中 n 个可测点组成可测集 $T = \{t_1, t_2, \dots, t_n\}$, 与之对应的测试费用集为 $C = [c_1, c_2, \dots, c_n]$ 。

(4)故障-测试相关矩阵 $D_{(m+1) \times n} = [d_{ij}]$, 其中当测试点 t_j 察觉到故障状态 f_i 时, $d_{ij} = 1$, 否则 $d_{ij} = 0$ 。如表 1 所示。显然当 f_0 表示系统无故障状态时, $d_{0j} = 0$ ($1 \leq j \leq n$)。

表 1 故障-测试相关矩阵

故障状态	t_1	t_2	\dots	t_n	故障率
f_0	c_1	c_2	\dots	c_n	$p(f_0)$
f_1	d_{01}	d_{02}	\dots	d_{0n}	$p(f_1)$
\vdots	d_{11}	d_{12}	\dots	d_{1n}	\vdots
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
f_m	d_{m1}	d_{m2}	\dots	d_{mn}	$p(f_m)$

(5)定义测试代价 J

$$J = P^T A C = \sum_{i=0}^m \sum_{j=1}^n a_{ij} p(f_i) c_j \quad (1)$$

其中 $A = [a_{ij}]$ 是一个 $(m+1) \times n$ 阶矩阵, 当选择测试 t_j 用于隔离故障状态 f_i 时, $a_{ij} = 1$, 否则 $a_{ij} = 0$ 。

测试序列优化问题可描述为在可测集 T 中选择一组测试, 并给出一种测试序列方案, 使能够隔离状态集 F 中的全部故障状态, 并使代价 J 最小。

根据二值属性特点, 给出如下定义:

定义 1: 在二值属性系统中, 定义二值状态集向量 $s = [s_i]^T$ ($0 \leq i \leq m, s_i = 1$ or 0)。用 s 表示系统目前的状态集。其中当 f_i 在当前的故障状态集中时, $s_i = 1$, 否则 $s_i = 0$ 。

由以上定义, 系统故障状态全集表示为 $s = [1, 1, \dots, 1]^T$, 当隔离出故障 f_k ($0 \leq k \leq m$) 时, $s = [0, \dots, 0, 1, 0, \dots, 0]^T$, 即 $s_k = 1$ 。

定义 2: 定义测试向量 $\text{test}_j = D(:, j)$ ($1 \leq j \leq n$)。其中 j 为与测试向量相应的测试, $D(:, j)$ 为 D 矩阵的第 j 列。

二值系统中选择测试 t_j 对当前状态集 s 进行测试隔离的过程, 可表示为测试向量 test_j 与当前状态集 s 的位与运算以及 s 与以上结果的位减运算。即隔离出的两个子集用 s_1 和 s_0 表示, 则 $s_1 = s \wedge \text{test}_j$, $s_0 = s - s_1$ 。

定义 3: 当 $s \wedge \text{test}_j \neq s$ 且 $s \wedge \text{test}_j \neq 0$ 时, 则 test_j 对于状态 s 具有隔离能力, 称 t_j 为状态 s 的有效测试。

由定义 3 可得以下推论:

推论 1: 对于一个拥有 $m+1$ 个故障的状态集, 将其中的故障全部隔离至多需要 m 次有效测试。

测试序列优化问题可描述为选择一组测试向量 test , 将故障状态全集 $s = [11 \dots 1]^T$ 隔离为单故障状态, 且使式(1)所示的测试代价最小。

2 用改进蚁群算法优化测试序列

2.1 蚁群算法简介

蚁群算法由 M.Dorigo 于 1991 年在其博士论文中首次被提出, 其设计的人工蚁模型中引入了真实蚂蚁觅食行为中的正反馈机制、信息素挥发机制以及基于概率的状态转移策略, 形成一种有效的群智能寻优算法。之后经过多次改进, 成功解决了许多复杂的组合优化问题, 这些初步研究已显示出蚁群优化算法在求解复杂优化问题(特别是离散优化问题)方面的优越性, 是一种很有发展前景的方法^[7]。

2.2 改进蚁群算法

测试序列优化问题是一种特殊的动态规划问题, 用传统的蚁群算法解决具有一定难度。为解决此类问题, 首先定义系统故障状态全集 $s = [11 \dots 1]^T$ 为初始蚁群, 将每一种故障状态看作一只相应位置上的蚂蚁, 每只蚂蚁的目标就是找到相应的故障状态。搜索过程中, 当每只蚂蚁都找到相应的故障状态时, 此代搜索结束。算法的整体流程如图 1 所示。

每个蚁群的具体搜索过程如图 2。每次搜索开始令 $s = [11 \dots 1]^T$ 为系统故障状态全集, 故障状态全部隔离时, 搜索结束。

传统的蚁群算法在解决动态树结构寻优问题时往往效果不佳, 对于复杂的问题甚至难以得到最优解或较理想的近优解, 因此必须在传统蚁群算法的基础上进行改进。

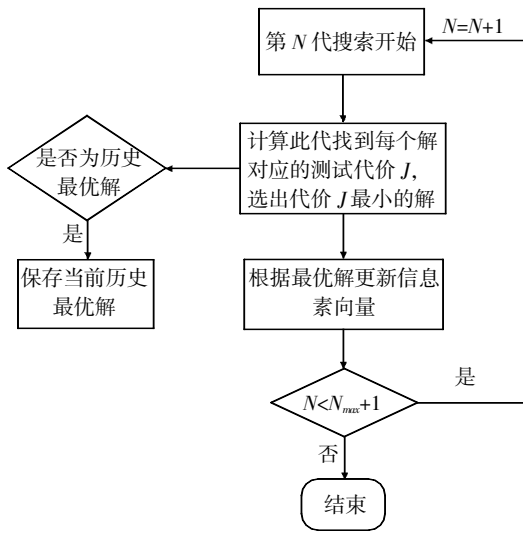


图 1 算法搜索整体流程

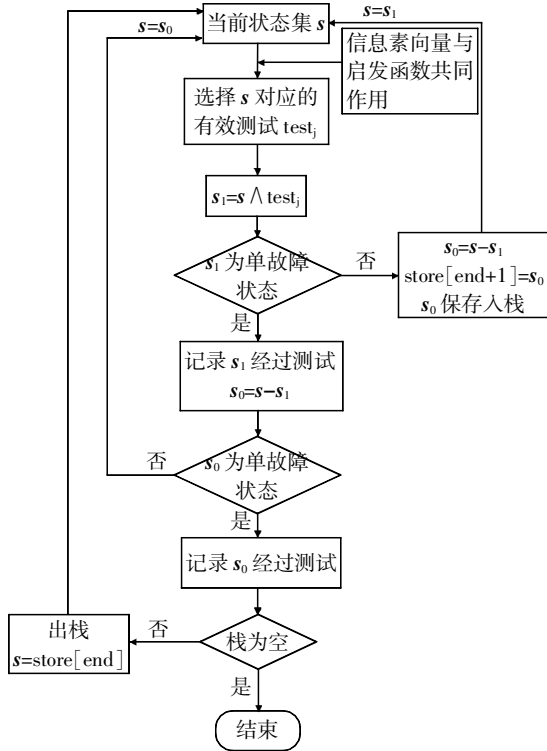


图 2 蚁群的搜索过程

2.2.1 初始解空间

由定义 3 可知,在隔离同一故障时相同的测试不可能出现两次,且由推理 1 可知,在所给测试能够保证隔离出故障集中的每一个故障时,算法一定能够找到一组解,从而保证了算法初始解空间的存在。

2.2.2 状态转移规则

初始时刻定义信息素向量 $\mathbf{Tau}=[11\cdots 1]^T$,为 1 行 n 列的行向量,其中 n 为全部测试数。当前状态 s 选择下一个测试 $test$ 的转移概率为

$$TP_k = \frac{\mathbf{Tau}(\text{allow}(s)_k)^\alpha \mathbf{Eta}(\text{allow}(s)_k)^\beta}{\sum_{k=1}^{\text{length}(\text{allow}(s))} \mathbf{Tau}(\text{allow}(s)_k)^\alpha \mathbf{Eta}(\text{allow}(s)_k)^\beta} \quad (2)$$

其中 α 为信息素浓度对于转移概率的重要程度, β 为启发函数对于转移概率的重要程度,可根据具体情况确定 α, β 值。蚁群将根据式(2)给出的转移概率公式,采用轮盘赌方法确定下一次选择的测试^[8]。

式(2)中 allow 函数的功能是在每次测试完成后,根据当前状态 s ,给出下一次所有可用的测试集,即定义 3 中定义的有效测试集。 k 可理解为可选测试的指针。传统的蚁群算法往往忽略启发函数项,经试验启发函数对算法的最优解质量具有重要的影响,因此本文在综合考虑了测试代价、故障概率的基础上,给出一个启发函数 \mathbf{Eta} ,其设计思想是尽量选择使测试后产生的两个子集中总故障概率平衡的测试,表达式如下:

$$\mathbf{Eta} = \frac{p_{s_1}^2 + p_{s_0}^2}{p_{s_1} p_{s_0} \text{cost}_{\text{test}}} \quad (3)$$

其中 $p_{s_1} = \sum_{f_i \in s_1} p(f_i)$,即 s_1 状态中所含故障状态的概率总和; $p_{s_0} = \sum_{f_i \in s_0} p(f_i)$, $\text{cost}_{\text{test}}$ 为测试 test 的代价。

2.2.3 分层加权的的信息素更新规则

在动态树结构的寻优问题中,树中节点的层次位置是重要的信息,对于搜寻最优解有很大的影响。而传统的蚁群算法中,信息素的更新往往是均匀的,没有利用节点的层次位置信息,因而很难在动态树结构的寻优问题中找到最优解,针对以上问题本文提出了一种分层加权的的信息素更新机制,信息素更新公式为

$$\mathbf{Tau}(\text{num}+1) = \rho \mathbf{Tau}(\text{num}) + \Delta \mathbf{Tau}(\text{num}) \quad (4)$$

$$\Delta \mathbf{Tau}_j(\text{num}) = [\Delta \mathbf{Tau}_j(\text{num})] \quad (1 \leq j \leq n) \quad (5)$$

$$\Delta \mathbf{Tau}_j(\text{num}) = \frac{Q}{\text{cost}} \sum_{l \in \text{testlist}} (\lambda^{l-1} \sum_{i \in \text{testlist}} p_{ji}(i)) \quad (6)$$

初始时刻,信息素向量中各元素的值相等。在以后的搜索中,将对每代搜索出的最优的测试序列进行信息素更新。式(4)中 ρ 为每代信息素保留的权值, $1-\rho$ 为信息素衰减系数。式(6)中 Q 为常数, cost 为每代最优的测试代价; λ 为本文定义的层间衰减系数; l 为隔离故障状态 f_i 时经过的测试数,即经过第一个测试点时 $l=1$, 经过第 n 个测试点时 $l=n$;

$\sum_{i \in \text{testlist}} p_{ji}(i)$ 表示在最优测试序列的第 l 层中隔离故障

f_i 用到的测试 $test_j$ 的总概率。经过分层加权处理,算法更易于区分测试的层次和重要度,加快算法的收敛速度。

为提高算法的全局搜索能力,避免算法因信息素差距过大而过早进入停滞状态,本文将采用最小蚂蚁系统策略限定信息素浓度值。

2.2.4 变异策略

为了避免蚁群算法陷入局部最优值,引入遗传算法的变异思想,在每代搜索结束后对信息素向量中浓度值低于阈值的项进行变异,提高其浓度值至最大值。经试验证明,变异操作可有效提高算法搜索到最优值的概率。

3 试验验证及算法分析

3.1 待优化问题

为验证算法的有效性,本文将对文献[1]中的超外差接收机系统进行测试序列优化。该系统有 22 个先验概率和 36 个预备测试,测试代价均为 1,其相关矩阵 D 如表 2 所示,其中矩阵元素值为 1 表示该测试检测到相应故障,否则值为 0。各故障的先验概率为

$$P=[0.00185, 0.00923, 0.185, 0.00185, 0.00185, 0.00923, 0.00185, 0.00923, 0.185, 0.185, 0.185, 0.00185, 0.00923, 0.185, 0.00923, 0.00185, 0.00923, 0.00185, 0.00185, 0.00185, 0.00185, 0.00185, 0.00185]^T$$

3.2 结果验证

表 3 所示算法采用的参数值,最终给出最优解中的一组如图 3,最优测试代价为 3.3535,和文献[1]中的最优测试代价 3.35 比较,本文提出的算法给出了一个比较满意的结果^[9]。

测试代价值随进化代数的变化情况如图 4 所示,可以看出算法以较高的效率收敛于最优值。

表 2 超外差接收机系统故障-测试相关矩阵

故障	测试
1	000000101110100000010111100000000001
2	111111111111111101111111110111111111
3	000000000110100000010111110000000001
4	000000000000000000010100000000000001
5	00010010111011100001011110000001011
6	111111111111111101101111110111111111
7	100100100100000000010010001001001001
8	101101101110100000010111101011010011
9	101101101110100000010111100101010011
10	000101101110100000010111100000100011
11	000000101110100000010111100000000101
12	000000000000000000000000000000000001
13	000111101110100000010111100000000011
14	101101101111111110001011110010111111
15	000000000000000000011100000000000001
16	000000000111100000010100000000000001
17	000000111110100000010111100000000101
18	111101101110100000010111100001010011
19	111111111111111111111111111111111111
20	000101101110100000010111100001110011
21	100100100110010010010010101001001001
22	111111111111111110111101111111111111

表 3 算法主要参数说明

参数	取值	参数说明
num	100	每代蚁群个数
itermax	20	最大搜索代数
α	1	信息素重要程度
β	3	启发函数重要程度
ρ	0.9	信息素保留比例
Q	5	信息素常数
λ	0.8	层间衰减系数

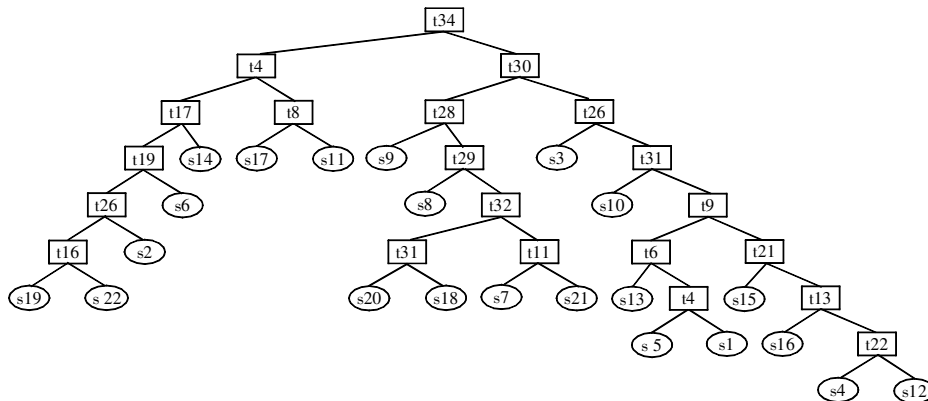


图 3 测试序列优化结果