Optimize Sequential Testing For System-level Based On Ant Algorithm

PAN Jia-liang，YE Xiao-hui，WANG Hong-xia

Department of Electronic Engineering, Naval University of Engineering, 44033, China

Panjialiang@sina.com whyxh007@yahoo.com.cn mimiconan@sina.com

Abstract：Directed towards the test sequencing problem in the sequential fault diagnosis for systems, the test sequencing problem is converted to searching the least complete test sequencing based on ant algorithm. The optimization of the test sequencing is searched by setting up transfer rule of the ant state and feedback of the pheromone. The better performance and feasibility of the proposed method is demonstrated by experimental results. The method is a effective solution for the test sequencing problem.

Key words：testability analysis;sequential test; ant algorithm;least complete test sequencing

基于蚁群算法的系统级序贯测试优化

潘佳梁，叶晓慧，王红霞

海军工程大学电子工程学院，湖北 武汉 44033

Panjialiang@sina.com whyxh007@yahoo.com.cn mimiconan@sina.com

【摘要】针对系统故障诊断中的序贯测试优化问题，本文基于蚁群算法，把测试序贯优化问题转换为蚁群建立最小完备测试序列的问题。通过构建蚁群的状态转移规则和信息素反馈机制，实现对测试最优路径的搜索。实例证明此算法具有良好的性能和可行性，有效的解决了系统级的序贯测试优化问题。

【关键词】测试性分析；序贯测试；蚁群算法；最小完备测试序列

1、引言

系统级测试和诊断策略设计又称为序贯测试优化生成，它是测试性设计和分析的重要研究内容之一。其目的是设计一组测试序列，使其在满足测试性指标要求的前提下消耗尽可能小的期望测试代价。该问题引起了国内外学者的广泛关注[1-9]。Pattipati[1]提出了运用信息熵和Haffman编码构建启发式函数，通过AO*算法解决了序贯测试问题。杨鹏博士[2-3]提出了不确定AO*算法和启发式搜索算法分别对测试不可靠条件下和多模式情况下的序贯测试问题进行了研究，取得了很好的效果。俞龙江博士[4]基于蚁群算法很好的解决了测试集优化问题，但算法对于测试顺序没有进一步的进行研究。黎琼炜博士[5]针对以往序贯测试优化算法复杂度太大的情况，提出间接熵法来解决系统级的序贯测试优化。

已有的序贯测试优化算法大多通过构建复杂的启发式函数来进行启发式搜索，这有利于解决复杂系统中的序贯测试优化，但搜索过程对于启发式函数的依赖程度较大，搜索过程对于先验知识要求也较多。本文基于蚁群算法，把测试序贯优化问题转换为搜索最小完备测试序列问题，利用蚁群记忆性和信息素反馈机制提出了一种解决序贯测试优化问题的蚁群优化算法。

2、序贯测试优化问题的提出

（ i ）用一个有限集合表示系统的状态 $F=(f_0,f_1,\cdots,f_m)$，其中 f_0 表示系统状态没有故障。$f_i(1\le i\le m)$ 表示 m 种可能的故障中的任意一种；

（ ii ）系统状态的先验概率表示为 $P_f=[p(f_0),p(f_1),\cdots,p(f_m)]$，$p(f_0)$ 表示系统没有故障存在的概率，$p(f_i)(1\le i\le m)$ 表示第 f_i 种故障发生的先验概率。我们假设系统最多同时存在一个故障（单故障模式）；

（iii）n 个可测点组成可测集 $T=(t_1,t_2,\cdots,t_n)$，测试费用为 $c=[c_1,c_2,\cdots,c_n]$，其中 t_i 的测试费用为 c_i；

（iv）故障—测试关联矩阵 $D_{(m+1)\times n}=[ft_{ij}]$，若已知系统状态为 f_i 时，t_j 测试为Fail，则 $ft_{ij}=1$，若 t_j 测试为Pass，则 $ft_{ij}=0$。显然无故障状态下 $ft_{0j}=0(\forall j)$。

定义1：如果一个测试序列

$T_s=\{t_{j1},t_{j2}\cdots t_{jn}\}, j_n\in 1,2,\cdots,n$ 能够满足系统要求的故障检测率和故障隔离率，则该测试序列是完备的。

定义 2：由所有完备测试序列构成的集合 $T=\{T_{s1},T_{s2},\cdots T_{sn}\}$ 为完备测试序列集。

定义 3：对于完备测试序列集 T 的子集 $T'\left(|T'|\le|T|\right)$，如果 T' 能够满足系统要求的故障检测率和故障隔离率，则该测试序列集是完备测试序列集的完备测试序列子集。对于完备测试序列子集 $T_k'\left(1\le k\le 2^n-1\right)$，如果 $|T^*|=\min_s|T_s'|\left(1\le s\le k\right)$，则 T^* 为最小完备测试序列。

最优测试序列问题可以描述为：在现有的信息下寻找一个最小完备测试序列，它采用完备测试序列集 T 中的测试序列以最小的成本来判断系统处于哪一种故障状态。成本函数由下式表示：

$$J=p^TAc=\sum_{i=0}^{m}\sum_{j=1}^{n}a_{ij}p(f_i)c_j$$

式中 $A=(a_{ij})$ 是一个 $(m+1)\times n$ 的二值矩阵，如果 t_j 在测试路径中能够隔离故障状态 f_i，则 $a_{ij}=1$，否则 $a_{ij}=0$。

3、蚁群算法

蚁群算法(ant algorithm)是一种源于大自然中生物世界的新的仿生类算法[6]，作为与遗传算法同属一类的通用型随机优化方法，它通过模拟自然界的蚂蚁的寻径行为，群体智能模型进行优化搜索，蚁群算法不需要先验知识，最初随机的选择搜索路径，随着解空间搜索过程的深入，选择逐渐变得有规律，并最终逼近甚至达到全局最优解。蚁群算法对搜索空间的深入是从观察蚁群觅食活动从中启发而建立的机制，主要包括三个方面：

（1）蚂蚁的记忆。一只蚂蚁搜索过的路径在下次搜索时就不会被选择，由此在蚁群算法中建立 tabu(禁忌)列表来进行模拟；

（2）蚂蚁利用信息素（pheromone）进行相互通信。蚂蚁在所选择的路径上会释放一种叫做信息素的物质，当同伴进行路径选择时，会根据路径上的信息素进行选择。

（3）蚂蚁的集群活动。通过一只蚂蚁的运动很难到达食物源，但整个蚁群进行搜索就完全不同。当某些路径上通过的蚂蚁越来越多时，在路径上留下的信息素数量也越来越多，导致信息素强度增大，蚂蚁选择该路径的概率随之增加，从而进一步增加该路径的信息素强度。通过这种正反馈，蚂蚁最终可以找到一条食物源和巢穴之间的最短路径。因此，模拟这种现象从而利用群体智能建立的路径选择机制，使蚁群算法的搜索向最优解推进。

可以将蚁群算法模型理解成增强型学习系统(reinforcement learning system)[7]。这种选择不满足马尔可夫性：某时刻采取的行动只与上一时刻的行动相关，与前面所有时刻采取的行动无关。这显而易见，因为蚂蚁每次选择路径后，就将该路径存到 tabu 列表中，选择下一条路径时，只能在 tabu 列表中不包括的路径中进行选择，而 tabu 表正是蚂蚁前面所有时刻采取的行动形成的。因此，蚁群算法模型不能理解为动态规划和蒙特卡罗这样的增强学习算法，而应理解为一种 Q 学习[8]，信息素可理解为 Q 学习中的回报。

4、用于序贯测试优化的蚁群算法

蚁群算法应用到序贯测试优化问题时，很容易想到可以让蚂蚁随机分布到各个测试上，每个蚂蚁从当前所在的测试出发，选择还未走过的测试，直到走过的所有测试组成的测试序列集是完备测试序列为止。每次迭代从每个蚂蚁完成的完备测试序列集中选择最优测试序列，经过多次迭代便求得序贯测试优化问题的最优解。

蚁群算法的流程如图 1 所示，可以表示为：

（1）初始化 $A(t)$ {初始化蚁群}

（2）评价 $A(t)$ {根据目标函数对每只蚂蚁的测试序列路径做评价}

（3)蚂蚁移动 {蚂蚁依据前面蚂蚁所留下的信息素和自己的判断选择路径}

（4）释放信息素 {根据评价，对蚂蚁所经过的测试序列路径按一定的比例释放信息素。}

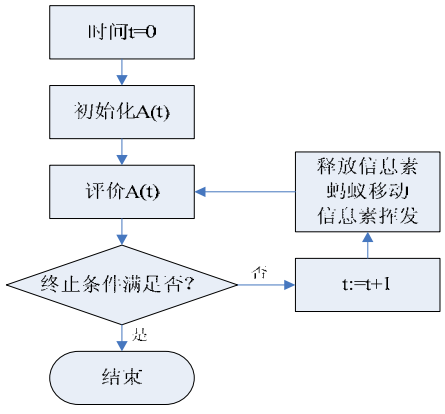

图1 蚁群算法流程图

（5）信息素挥发 {信息素随着时间不断消散}

978-1-935068-01-3 © 2009 SciRes.

4.1 状态转移规则

初始时刻，各条路径上的信息素量相等，设 $\tau_{ij}(0)=C$（C 为常数）。蚂蚁 $k(k=1,2,\cdots,m)$ 在运动过程中根据各条路径上的信息素量决定转移方向。蚂蚁系统所使用的状态转移规则被称为随机比例规则，它给出了位于测试 i 的蚂蚁 k 选择移动到测试 j 的概率。在 t 时刻，蚂蚁 k 在测试 i 选择测试 j 的转移概率 $P_{ij}^{k}(t)$ 为：

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)}{\sum\limits_{s\in allowed_k}\tau_{is}^{\alpha}}, & j\in allowed_k\\ 0, & otherwise\end{cases} \quad (1)$$

其中，$allowed_k=\{0,1,\cdots,n-1\}$ 表示蚂蚁 k 下一步允许选择的测试。τ_{ij} 表示由测试 i 转移到测试 j 时，边 (i,j) 上的信息素轨迹强度。由上式可知，转移概率 $P_{ij}^{k}(t)$ 与 τ_{ij}^{α} 成正比。参数 α 反映了蚂蚁在运动过程中所积累的信息在蚂蚁选择路径中的重要性。与真实蚁群不同，人工蚁群系统具有记忆功能。为了满足蚂蚁不允许访问已经访问过的测试这个约束条件，为每个蚂蚁都设计了一个数据结构，称为禁忌表（tabu list）。禁忌表记录了在 t 时刻蚂蚁已经走过的测试，不允许在本次循环中再经过这些测试。当本次循环结束后，禁忌表被用来计算该蚂蚁当前所建立的解决方案（即蚂蚁所经过的测试序列集）。之后，禁忌表被清空，该蚂蚁又可以自由地进行选择。

4.2 信息素更新

M.Dorigo 曾给出三种不同的模型[9]，分别称为蚁周系统（antcycle system）、蚁量系统（ant-quantity system）和蚁密系统（ant-density system）[10]。在此，采用蚁周模型来进行信息素的更新，蚁周系统是在蚂蚁已经建立了完整信息后再释放信息素，利用的是整体信息。各路径上信息素量根据下式调整：

$$\tau_{ij}(t+n)=\rho\cdot\tau_{ij}(t)+\Delta\tau_{ij}(t,t+n) \quad (2)$$

$$\Delta\tau_{ij}(t,t+n)=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t,t+n) \quad (3)$$

$$\Delta\tau_{ij}^{k}(t,t+n)=\begin{cases}\dfrac{Q}{L_k}, & \text{若蚂蚁}k\text{本循环中经过路径}(i,j)\\ 0, & \text{否则}\end{cases} \quad (4)$$

$$Q=\frac{\sum_{i=1}^{n}c_i}{J} \quad (5)$$

其中，$\Delta\tau_{ij}^{k}(t,t+n)$ 表示第 k 只蚂蚁在时刻 $(t,t+n)$ 留在路径 (i,j) 上的信息素量，其值视蚂蚁表现的优劣程度而定。$\Delta\tau_{ij}(t,t+n)$ 表示本次循环中路径 (i,j) 的信息素量的增量；ρ 为信息素轨迹的衰减系数，通常系数 $\rho<1$ 来避免路径上轨迹量的无限累加；Q 为单个蚂蚁释放的信息素总量，L_k 为第 k 只蚂蚁在本次循环中所走的测试序列长度。单个蚂蚁释放的信息素总量 Q 的大小是由测试序列的优劣程度决定，$\sum c_i$ 是所有测试点的测试费用总和，是个常数，J 是测试序列成本函数，Q 与 $\sum c_i$ 成正比，与 J 成反比。成本越小，则单个蚂蚁释放信息总量 Q 越大，则测试路径越好，信息素释放的就越多；反之，信息素释放越少。

4.3 序贯测试优化算法实现

（1）初始化过程

设 $t=0$；　{t 为时间计数器}

$N_c=0$；　{N_c 循环次数计数器}

$\tau_{ij}(t)=0$；　{为每条路径 (i,j) 设一个信息素轨迹强度的初始值}

$\Delta\tau_{ij}=\dfrac{\sum_{i=1}^{n}C_i}{n}$；　{轨迹强度的增量的初值设为所有测试点费用的平均值}

$tabu_k=\varnothing$　{在初始阶段，禁忌表为空}

将 m 只蚂蚁随机置于 n 个节点上：

设置 $s=1$；　{s 为禁忌表索引，将各蚂蚁的初始所在测试位置置于当前禁忌表中}

for k=1 to n do

for k=1 to $b_i(t)$ do　{$b_i(t)$ 为 t 时刻位于测试 i 的蚂蚁个数，$m=\sum bi(t)$}

$tabu_k(s)=i$

（2）重复直到满足测试性指标或者禁忌表满为止

设置 $s=s+1$

for i=1 to n do

for k=1 to $b_i(t)$ do

以概率 $P_{ij}^{k}(t)$ 选择测试 j；

将蚂蚁 k 移动到 j；

将刚刚选择的测试 j 加到 $tabu_k$ 中；

（3）for k=1 to n do

根据禁忌表的记录计算 L_k；

for s=1 to $n-1$ do　{搜索蚂蚁 k 的禁表}

设 $(h,l)=(tabu_k(s),tabu_k(s+1))$ {(h,l) 是在蚂蚁 k 的禁忌表中连接测试 $(s,s+1)$ 的路径}

$$\Delta\tau_{hl}(t+n)=\Delta\tau_{hl}(t+n)+\frac{Q}{L_k}$$

（4）对于每一条路径 (i,j)，根据公式（2）计算 $\tau_{ij}(t+n)$

设 $t=t+n$

对每条路径 (i,j) 设 $\Delta\tau_{ij}(t,t+n)=0$

（5）记录到目前为止的最短路径

If $N_c<N_{c\max}$

Then

清空所有的禁忌表

设置 $s=1$

for i=1 to n do

for k=1 to $b_i(t)$ do

$tabu_k(s)=i$ {一次循环后蚂蚁又回到初始位置}

设 $t=t+1$

对于每一条路径 (i,j)，设置 $\Delta\tau_{ij}(t,t+1)=0$

返回步骤（2）

else

输出最短路径；

end

5、案例验证

5.1 仿真计算

应用文献[1]中的案例验证本文的算法，已知故障—测试关联矩阵及故障概率分布如表 1 所示，并假设所有备选测试的费用均为 1。

文献 [1] 采用 AO^* 算法构建的诊断策略为 $\{t_2\rightarrow t_4\rightarrow t_1\}$，即最小完备测试序列 $T_s=\{t_2,t_4,t_1\}$，平均测试成本为 2.18，故障检测率和故障隔离率均为 100%。

利用本文的蚁群算法来构建诊断策略，参数设置如下：循环次数 $N_{c\max}=50$；蚂蚁数量 $m=100$；信息素衰减系数 0.8。算法共找到 4 组最小完备测试序列 $\{t_2,t_4,t_1\}$、$\{t_4,t_2,t_1\}$、$\{t_2,t_4,t_5\}$、$\{t_4,t_2,t_5\}$，平均测试成本都为 2.18，故障检测率和隔离率均为 100%。

表 1 故障—测试相关矩阵

状态	t_1	t_2	t_3	t_4	t_5	概率
	测试费用均为1					
f_0	0	0	0	0	0	0.70
f_1	0	1	0	0	1	0.01
f_2	0	0	1	1	0	0.02
f_3	1	0	0	1	1	0.10
f_4	1	1	0	0	0	0.05
f_5	1	1	1	1	0	0.12

AO^* 算法是采用启发式搜索算法，所以每次搜索是建立在前一次搜索结果得到的基础上，AO^* 算法搜索的优劣程度依赖于构建的启发式函数的好坏。本文的算法，能够避免构建复杂的启发式函数，而是从解的质量上去衡量测试集的优劣，自适应的更新诊断策略，因此蚁群算法能有效的搜索到最小完备测试序列。

5.2 仿真分析

序贯测试的蚁群算法参数对算法的性能有很大的影响。循环次数 N_c 的大小和蚂蚁数量 m 的大小能决定寻找解的速度和解的质量。蚂蚁数量 m 对算法性能影响如表 2 所示，循环次数 N_c 对算法性能的影响如表 3 所示。如果蚂蚁的数量 m 与测试数 n 相近或者太少，搜索解的过程会很快，但是由于蚂蚁数量太少，容易导致蚂蚁陷入局部最优解或者根本无法找到适合的解；循环次数 N_c 太少，会导致信息素的更新不够，路径之间的差异不明现，不易找到最优解。若蚂蚁数量 m 和循环次数 N_c 过多，则信息素的更新时间变长，导致算法寻优时间过长。

信息素的衰减系数同样影响解的质量和寻优的效率。信息素的衰减系数决定了蚁群算法收敛的快慢，若信息素衰减系数较小，则较差的路径信息素的挥发大于信息素的增加，此路径在早期就会被舍弃，这样加快了算法的收敛，但也可能过早的收敛到局部最优路径，从而无法找到全局最优路径。若信息素衰减系数较大，信息素的挥发对路径选择的贡献会较低，收敛过程较慢，不易过早收敛于局部最优路径，但这样也可能使算法循

表 2 蚂蚁数量对算法性能的影响

蚂蚁数量 m	循环次数 N_c	总执行次数 N	搜索到最优解的次数（占总执行次数的比例）	运行时间 t（秒）
20	20	30	12（40%）	0.9531
50	20	30	23（76.67%）	1.6094
100	20	30	30（100%）	3.1094
200	20	30	30（100%）	6.2969

表 3 循环次数对算法性能的影响

循环次数 N_c	蚂蚁数量 m	总执行次数 N	搜索到最优解的次数（占总执行次数的比例）	运行时间 t（秒）
10	100	30	22（73.33%）	1.7813
20	100	30	28（93.33%）	3.3594
40	100	30	30（100%）	6.1875
80	100	30	30（100%）	14.1250

978-1-935068-01-3 © 2009 SciRes.

环结束时还未建立起最优路径。信息素的挥发程度应该大于最差路径的信息素增量，小于最优路径的信息素增量。在对蚂蚁数量 $m=100$，循环次数 $N_c=80$ 的情况下，不同的信息素衰减系数对算法性能的影响如图 2 所示。可以看到，信息素衰减系数不会影响蚁群逐步寻优的趋势，蚁群找到最优解的数量随着循环次数逐渐增多，最后所有蚂蚁形成唯一的最优路径。在信息素衰减系数小的情况下（$\rho=0.01,\rho=0.2$），算法分别在第26代和第35代就形成唯一的最优路径。在信息素衰减系数大的情况下（$\rho=0.8,\rho=0.95$），算法虽然没有形成唯一的最优路径，但是蚁群始终是向最优解方向进行移动。

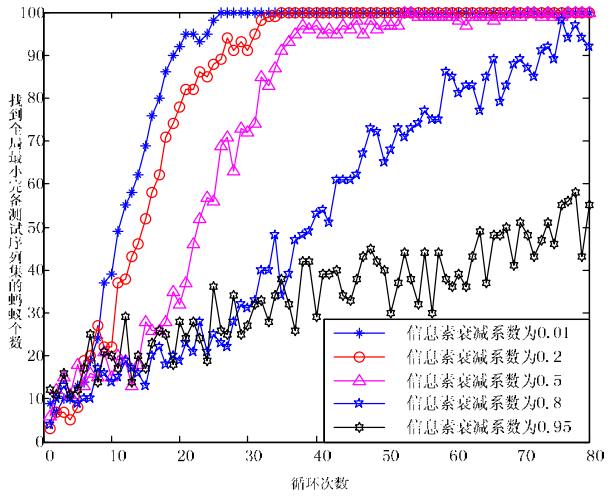

图2 信息素衰减系数对算法性能的影响

5.3 实例验证

为了进一步验证本文算法，应用文献中的实例进行了仿真计算。表 4 中列出了各系统主要统计数据，包括被测系统的状态数、候选的测试数以及模式数目。应用本文的蚁群算法得到诊断策略的平均测试费用如表 4 所示。这些实例验证了本算法对于序贯测试优化问题的有效性。

表 4 测试实例的主要参数及计算结果

系统名称	状态数	测试数	平均测试费用	运行时间(秒)
汽车刹车系统[11]	8	9	2.98	0.8756
阿波罗检测系统[1]	10	15	3.40	1.5628
超外差接收机[1]	22	36	3.35	5.2635
反坦克导弹发射系统[12]	25	22	4.44	4.8265

6、结论

序贯测试优化问题其实就是在所有测试中选择最小完备测试序列，针对本文建立的数学模型，应用蚁群优化算法中的蚁周系统模型，把求最小完备测试序列转换为蚂蚁搜索最优路径问题，本文的算法避免了构造复杂的启发式算法，并且具有通用性。应用本算法通过对实例的求解，得到了与已知最优解相一致的结论，应用本文算法能够对复杂系统进行测试性分析并且生成有效的诊断策略。由于该方法中测试、故障隔离结论的定义是广义的，故它不仅适用于系统级，也适用于模块级和元件级的测试诊断，在工程应用中有一定的前景。

致谢

感谢我的导师叶晓慧教授以及王红霞博士对我研究的支持和帮助，感谢亲人的鼓励和关心，感谢各位评审的专家与教授。

References（参考文献）

[1] Pattipati K R, Alexandridis M. Application of heuristic search and information theory to sequential fault diagnosis[J]. IEEE Trans. on System, Man, and Cy-bernetics, 1990, 20(4): 872-887.
[2] Yang Pen, Qiu Jing, Liu Guan-Jun. Efficient diagnostic strategy generation with non-independent tests[J]. Journal of national university of defense technology. 2008, 30(1):99-103.
杨鹏,邱静,刘冠军. 基于非独立测试的诊断策略优化生成[J]. 国防科技大学学报, 2008, 30(1): 99-103.
[3] Yang Pen, Qiu Jing, Liu Guan-Jun. Optimization method for diagnostic strategy with unreliable test[J]. Chinese Journal ofScientific Instrument. 2008, 29(4): 850-854.
杨鹏,邱静,刘冠军. 测试不可靠条件下的诊断策略优化研究[J]. 仪器仪表学报, 2008, 29(4): 850-854.
[4] Yu Long-jiang, Peng Xi-yuan, Peng Yu. A Test Set Optimization Method Based on Ant Algorithm[J]. Acta electronica sinica. 2003, (08): 1178-1181.俞龙江, 彭喜源, 彭宇. 基于蚁群算法的测试集优化[J]. 电子学报, 2003, (08): 1178-1181.
[5] Li Qiongwei, Yi Xiao-shan, Liu Guan-jun. An Indirect-Entropy Algorithm for System-Level Fault Isolation[J]. Systems Engineering and Electronics. 2001, 23(2), 51-54.
黎琼炜, 易晓山, 刘冠军. 系统级故障隔离的间接熵法[J]. 系统工程与电子技术, 2001, 23(2), 51-54.
[6] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies[A]. Proc.1st European Conf. On artificial life [C].Paris, France:Elsevier, 1991:134-142.
[7] http://reinforcementlearning.ai-depot.com/Intro.html〔Z/OL〕.
[8] Gambardella M, Dorigo M. Ant-Q:A reinforcement learning approach to the traveling salesman problem. Proc. of ML-95, Twelfth Intern Conf. on Machine Learning[C]. Morgan Kaufmann, 1995.252-260.
[9] M Dorigo, V Maniezzo, A Colorni. Positive Feedback as a Search Strategy. Technical Report 91-016.
[10] B Bullnheimer, R F Hartl, C Strauss. Applying the ant System to the Vehicle Routing Problem. Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. Kluwer, Boston, 1998. 109~120.
[11] Zuzec A, Biasizzo A, Novak F. Sequential Diagnosis tool[J]. Microprocessors and Microsystems, 2000, 24:191-197.
[12] Simpson W R, Sheppard J W. SystemTest and Diagnosis[M]. Boston: Kluwer Academic Publishers, 1994: 91-138