贪心算法在系统故障诊断策略生成中的应用①

孙　煜，刘松风，马　力

(91872 部队，北京 102442)

摘　要: 诊断策略是具有一定顺序的测试序列。诊断策略生成的目标是隔离故障，并使测试开销达到最小。本文介绍测试序列生成的各种信息启发式算法并通过对比各种算法诊断策略生成的期望开销，着重讨论了 Rollout 算法的优越性。

关键词: 诊断策略；信息启发式；贪心算法；Rollout 算法

Application of Greedy Algorithm to Sequential Fault Diagnosis

SUN Yu, LIU Song-Feng, MA Li

(91872th Units, Beijing 102442, China)

Abstract: Diagnostic strategy is to have a certain order of test sequence. Diagnosis strategy aims to generate fault isolation, and to minimize testing costs. This paper introduces the test sequence generated by comparing the various algorithms and an algorithm on diagnostic strategies discussed Rollout algorithm.

Keywords: diagnostic strategy; information heuristic; greedy algorithm; Rollout algorithm

1　引言

诊断策略是具有一定顺序的测试序列。在装备的生产制造和维修中，测试序列可以用来评估系统的测试性，以改进系统测试性的设计，比如确定最优测试点的放置位置；可以作为交互式的维修辅助手段，通过可视化的诊断树快速定位和排除系统故障，大大缩短维修时间。

测试序列生成的各种算法的关系可用图 1 表示。

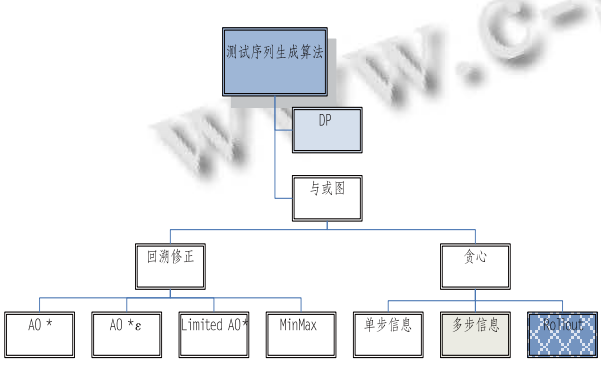

图 1　测试序列生成算法关系

测试序列生成是一个 NP 完全问题[1]。测试序列生成算法可分为两类，一类是基于动态规划(DP)的算法。一类是基于与或图(AO 图)的启发式搜索算法。

DP 算法虽然可以得到最优解，但具有 $o(mn3^n)$ (m 为系统状态数，n 为测试数目)的时间复杂度，且对存储空间也有要求，因此当 n>20 时已不再适用。

基于与或图搜索的算法可分为两类，第一类算法通过不断回溯修正诊断策略步骤得到最优解，时间复杂度高；第二类算法通过设计优秀的评估函数，通过贪心策略就可达到相同的目的，时间复杂度低。本文将介绍各种信息启发式贪心搜索算法并详细介绍 Rollout 算法生成诊断策略。

2　测试序列问题的描述

2.1 测试序列生成问题的组成

用于系统故障诊断的测试序列生成问题，有以下几部分组成[1]：

(1) 故障失效集合 $S=[s_0,s_1,...,s_m]$， $s_l(1\le l\le m)$ 表

① 收稿时间:2010-05-13;收到修改稿时间:2010-06-21

示 m 个潜在失效故障中的一个失效故障，s_0 表示无故障状态；

(2) 失效故障的概率 $P=[p(s_0),p(s_1),...,p(s_m)]^T$；

(3) n 个可用测试的集合 $T=[t_1,t_2,...,t_n]$，和测试对应的测试费用集合 $C=[c_1,c_2,...,c_n]^T$；

(4) 失效故障的维修费用集 $F=[f_1,f_2,...,f_m]$；

(5) 故障测试相关矩阵 $D=[d_{ij}]$，如果测试 t_j 能够检测到失效故障 s_l，d_{ij}=1，否则为 0。

测试序列的平均费用可用下面公式描述[2]：

$$J=\sum_{l=0}^{m}\{\sum_{j=1}^{|p_l|}c_{p_l[j]}\}p(s_l)$$

p_l 代表为了隔离故障 $s_l(1\le l\le m)$ 所用的测试集合。最优的测试序列即使平均测试费用 J 最小的序列。

2.2 标题测试序列生成问题的组成

寻求最优测试序列的过程就是寻找二进制决策树的过程。测试序列生成的目标是隔离系统故障，并使相应的测试费用最小。一个测试序列可用与或图描述。

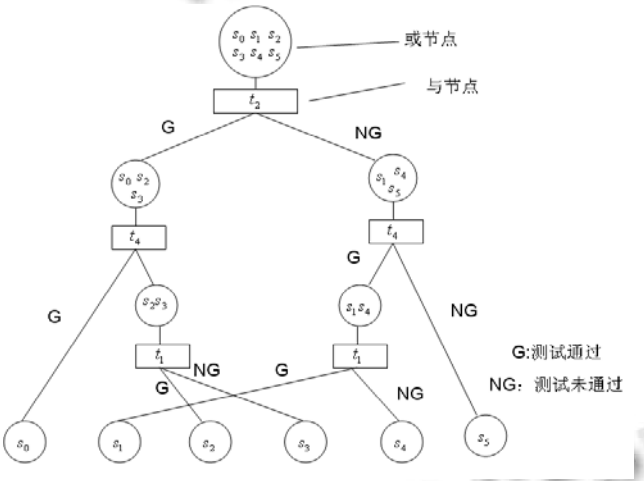

图 2 测试序列与或图

如图 2，测试序列与或图是一个由与节点(方框节点，代表一个测试)和或节点(圆圈节点，代表一个系统状态的集合)组成的有向图，其中包含一个根节点 S，为或节点，它包含所有的系统状态，代表待解决的问题。

一个问题，若将其视为根节点，那么该问题求解的过程，就是不断寻找后继可解节点，从而使根节点可解的与或图生成过程。这样生成的所有节点可解的与或图，就是一个解图，若此解图满足生成过程中的代价最小，就称其为最优解图。

把最优测试序列生成问题和与或图理论相联系，得到以下结论[3]：

(1) 节点 S 表示待求解的测试序列生成的初始问题，它包含系统的所有故障状态与无故障状态；

(2) 中间或节点对应为求解初始问题需求解的子问题，包含多个故障状态，为模糊子集；

(3) 与节点代表最优的测试；

(4) 或节点代表了系统状态的模糊子集。

由此可见，最优测试序列生成的问题，就是与或图寻找最优解图的过程。以下的各种算法，都是基于这个思想实现的。

3 信息启发式贪心算法及发展

AO*相关算法通过不断回溯修正，搜索到问题的最优解。回溯操作是造成算法时间复杂度高的直接原因。因此基于信息论的贪心算法被提了出来。目前主要有三种信息启发式搜索，分别为单步信息启发式搜索、多步信息启发式搜索与 Rollout 信息启发式搜索。

3.1 单步信息启发式搜索(SIG)

单步信息启发式搜索技术以最大化单位测试费用的信息增益为选择测试的标准。算法中，对于模糊集 x，测试 t_k 被选择，如果这个测试最大化了单位测试费用的信息增益：

$$b(x,t_j)=\frac{IG(x,t_j)}{c_j}$$

$$IG(x,t_j)=-\{p(x_{jp})\log_2 p(x_{jp})+p(x_{jf})\log_2 p(x_{jf})\}$$

因此 $k=\arg\max_j\{\frac{IG(x,t_j)}{c_j}\}$

其中 x_{jp},x_{jf} 是模糊集 x 通过测试 t_k，展开的左右子集（通过测试子集与不通过测试子集），$p(x_{jp})$和$p(x_{jf})$代表左右子集的条件概率。此算法是贪心算法，因此算法的时间复杂度为 $o(mn)$。

另一个相关的启发式搜索算法是分离搜索法。它对测试的评估标准为：

$$d_c(x,t_j)=p(x_{jp})*p(x_{jf})$$

类似，一个测试 t_k 将被选择，如果这个测试最大化了当前模糊集节点的评估标准。

当每个测试费用相同的时候，上面的单步信息启发式搜索和分离搜索法得到一样的诊断树。在系统故障状态的先验概率相等的情况下，分离搜索法得到的测试费用 J_d 有上限：

$$\frac{J_d}{J^*} \le 2q + \frac{2q\log_2(m+1)}{[1+\log_2 q+\log_2\log_2(m+1)]}$$

其中，J^*代表最优测试费用。

在系统故障状态的先验概率不相等的情况下，分离搜索方法可能产生病态问题，使得$\frac{J_d}{J^*}$为一个很大的比例值。然而，单步信息启发式搜索方法，根据经验判断，得到的测试费用一般不超过最优测试费用的 2 倍。

3.2 多步信息启发式搜索(MIG)

单步信息启发式搜索只通过一层的隔离结果判定测试的优劣。如果对模糊集 x 下的每一个测试，首先通过单步信息启发式搜索生成一个 MSTEP 层的部分诊断树，然后用此部分诊断树的隔离结果判定测试的优劣，并选择此意义下最优的测试应用于模糊集 x。则得到多步信息启发式搜索。对于模糊集 x，选择测试的步骤如下：

(1) 从现有可用的测试集合中选择一个测试t_j，把模糊集 x 根据输出结果的通过或者失败拆分成两个子集。

(2) 对于每一个子集，基于单步的信息启发式搜索选择可用的最优测试，并根据输出结果的通过和失败拆分子集。

(3) 对于每个子集递归的重复步骤(2)，直到部分诊断树的深度达到 MSTEP。计算部分诊断树的单位测试费用的信息增益。

(4) 对于(1)中每一个被选择的测试重复步骤(2)和(3)。

(5) 对于当前的模糊集 x，选择一个最优测试，基于这个最优测试生成的部分诊断树对模糊集 x 具有最大的单位测试费用的信息增益。

部分诊断树的测试费用的信息增益用下列公式计算：

$$B(x,G_j)=\frac{IG(x,G_j)}{COST(x,G_j)}$$

$$IG(x,G_j)=-\sum_{q=1}^{l_j}\frac{p(x_{jq})}{p(x)}\log_2(\frac{p(x_{jq})}{p(x)})$$

$$COST(x,G_j)=\sum_{q=1}^{l_j}\left(\frac{p(x_{jq})}{p(x)}\sum_{k=1}^{|P_q|}c_{P_q[k]}\right)$$

G_j 是模糊集 x 基于测试 t_j 生成的 MSTEP 层的部分诊断树。$IG(x,G_j)$表示部分诊断树G_j的信息增益。$COST(x,G_j)$表示部分诊断树的平均测试开销。$x_{j1},x_{j2},\cdots x_{jl_j}$表示部分诊断树$G_j$的叶子节点（或中间叶子节点，为子模糊集）。$l_j$表示叶子节点和中间叶子节点的个数。$P_q$是在部分诊断树中，从节点 x 到叶子节点或中间叶子节点x_{jq}，用到的测试序列。$|P_q|$是测试序列P_q的元素个数。$P_q[k]$是P_q的第个测试。$c_{P_q[k]}$是P_q的k第个测试的测试费用。图 3 是各种测试序列选择算法对于各种不同规模的实际问题的时间开销。

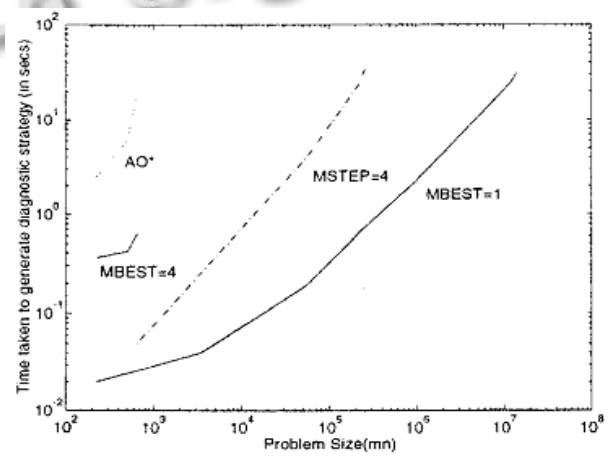

图 3　问题规模和算法的时间开销

3.3 ROLLOUT 信息启发式搜索(RIG)

3.3.1 算法描述

多步信息启发式搜索在选择当前模糊集的测试时，预先生成 MSTEP 层的部分解图(部分诊断树)，通过评估该部分解图隔离故障的能力来选择测试。若将生成 MSTEP 层的部分解图改为生成基于单步信息启发式搜索的完全解图，并通过计算该解图的平均测试开销来评估测试，则得到 Rollout 信息启发式算法。具体的步骤如下[2]：

Step1.

初始化一个模糊集节点集合 Z，只包含根节点 S。初始化一个图 G。如果 S 是终端节点，则将 S 加入到 G，退出。G 就是问题的解。

Step2.

重复下面步骤，直到集合 Z 为空，然后退出。G

就是问题的解。

Step2.1

从 Z 中移除一个模糊集节点 x_i，加入到图 G 中。如果 x_i 是一个已解节点，则重新选另一个节点，并加入到图 G 中。对节点 x_i 的可用测试集 T_i 中的所有测试 t_j，重复下面步骤。

Step2.1.1

初始化一个模糊集节点集合 Y' 与一个图 G'，将 t_j 加入到 G' 中。用 t_j 将节 x_i 分成两个子模糊集 x_{ijp} 与 x_{ijf} (分别表示测试通过集合与测试失败集合)。将 x_{ijp}、x_{ijf} 加入到 Y' 中。重复下面步骤直到 Y' 为空。

Step2.1.1.1.

从中移除一个模糊集节点 y_k'，加入到图 G' 中。如果 y_k' 是一个已解的节点，则重新选另一个节点，并加入到图 G' 中。否则计算其可用测试集 T_k' 中的所有测试的单位测试费用的信息增益 $b(y_k', t_l'), t_l' \in T_k'$。

Step2.1.1.2

选择 $t'^* \in T_k', b(y_k', t'^*) = \max_{t_l' \in T_k'}[b(y_k', t_l')]$，将 t'^* 加入到 G' 中。用 t'^* 将 y_k' 分成两个子模糊集，加入到 Y' 中。

Step2.1.2

计算图的平均测试开销，作为测试 t_j 的评估。

$$h(x_i, t_j) = \sum_{q=1}^{l_j} \left(\frac{p(x_{jq})}{p(x)} \sum_{k=1}^{|P_q|} c_{P_q[k]} \right)$$

$x_{j1}, x_{j2}, \cdots x_{jl_j}$ 表示 G' 的叶子节点。l_j 表示叶子节点的个数。P_q 是从 t_j 到节点叶子节点 x_{jq} 的测试序列(包括 t_j)。$|P_q|$ 是测试序列 P_q 的元素个数。$P_q[k]$ 是 P_q 的第 k 个测试。$c_{P_q[k]}$ 是 P_q 的第 k 个测试的测试费用。

Step2.2.

选择 $t^* \in T_i$，$h(x_i, t^*) = \underset{j \in T_i}{Min}[h(x_i, t_j)]$，将节点 x_i 展开。将 t^* 加入到图 G 中。将用 t^* 拆分的子模糊集 x_{ip}, x_{if} 加入到 Z 中。

Rollout 信息启发式算法通过预先生成某个测试下的单步信息启发式解图来评估测试的性能。若将预先生成解图的方法改为 HEF 引导的贪心算法，就得到各种 RHEF 算法[4]。

4 举例

针对表 1 所示例子，从初始节点 S 出发，按如下步骤选择第一步最佳测试：

表 1 相关矩阵、故障概率和测试费用

故障 S_i	测试 t t_1	t_2	t_3	t_4	t_5	故障概率 P（S_i）
	测试费用 C 1	1	1	1	1	
S_0	0	0	0	0	0	0.70
S_1	0	1	0	0	1	0.01
S_2	0	0	1	1	0	0.02
S_3	1	0	0	1	1	0.10
S_4	1	1	0	0	0	0.05
S_5	1	1	1	1	0	0.12

使用测试 t_1，将 S 分为“通过”和“不通过”两个模糊集：$\{s_0, s_1, s_2\}$、$\{s_3, s_4, s_5\}$。首先，对模糊集节点 $\{s_0, s_1, s_2\}$ 运用单步信息启发式搜索，建立诊断树。然后由底至顶计算 $\{s_0, s_1, s_2\}$ 的测试开销。得出 $h(\{s_0, s_1, s_2\}) = 1.97$。对 $\{s_3, s_4, s_5\}$ 进行同样操作。得出 $h(\{s_3, s_4, s_5\}) = 1.56$。

则 t_1 在节点 S 下的测试开销为：$h(S, t_1) = c_1 + p(\{s_0, s_1, s_2\})h(\{s_0, s_1, s_2\}) + p(\{s_3, s_4, s_5\})h(\{s_3, s_4, s_5\}) = 2.86$

对 t_2 采用同样的处理方法，得到 t_2 的测试开销为 2.18。依次，计算其他测试的开销。计算结果如图 4 所示。

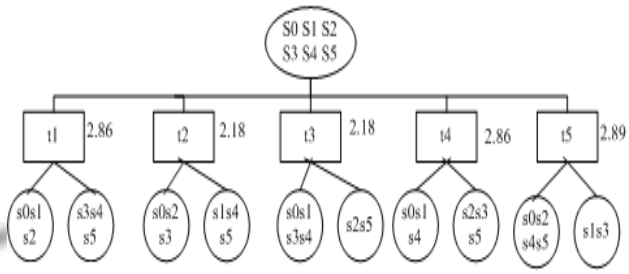

图 4 S 所有可用测试的测试开销

对 S 选择测试开销最小的测试。如图 4 所示，t_2, t_3 的测试开销都为 2.18，则选择下标较小者 t_2。使用 t_2 将 S 分为两个模糊子集，至此第一步测试已确定，然后对 t_2 的两个直接后继节点采用同样的方法进一步展开，直到得到完整的解图。

与单步信息启发式搜索比较。如果只应用单步信息启发式搜索，那么第一个测试将选择 t_1，则这棵诊断树的期望测试开销将是 2.86。而采用 ROLLOUT 算法的第一步测试将选择 t_2，期望的测试开销为 2.18。结果如图 5 所示。

$R_n(u, j, t)$

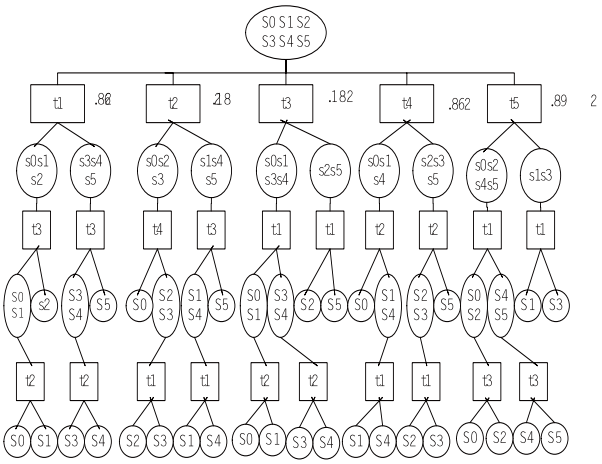

图 5 比较 ROLLOUT 算法与 SIG 算法的解图

5 结论

ROLLOUT 信息启发式搜索算法的优点：

(1) 适用于大规模的系统，可以解决 50000 故障源和 45000 测试点的测试性序列生成问题。

(2) 有广泛的适用性和可执行性。

(3) 保证诊断策略的质量，接近最优解。

Rollout 信息启发式算法是基于单步信息启发式搜索算法和 Rollout 战略提出的一种新型算法。其执行效率高，充分体现了信息启发式贪心搜索的优越特征。其解的质量优于其他信息启发式搜索。在测试诊断邻域有广泛应用的美国 QSI 公司的 Teams 软件版本 10.0 采用的也是此算法。

参考文献

1 Pattipati KR, Alexandrids MG. “Application of Heuristic Search and Information Theory to Sequential Fault Diagnosis”. IEEE Trans. Syst., Man, Cybern., 1990 vol.20:872－887.

2 Tu F, Pattipati KR, “Rollout Strategies for Sequential Fault Diagnosis”, IEEE Trans., 2002:32－47.

3 Raghavan V, Shakeri M and Pattipati KR, “Optimal and Near-Optimal Test Sequencing Algorithms with Realistic Test Models”, IEEE Trans. Syst., Man, Cybern., 1999, vol. 29(1):79－88.

4 田仲等. 系统测试性设计分析与验证. 北京:航空航天大学出版社, 2004.