文章编号：1001－2486(2002)04－0100－05

基于 One－port 模式的测试点选取问题※

戴　丽，郁殿龙，谢　政

（国防科技大学理学院，湖南 长沙　410073）

摘　要：在故障诊断过程中，每个测试点检测故障所需的时间可能不同。对于每个测试点一次检测所有可检测故障点的问题已经获得解决。对于每个测试点一次只能检测一个故障点，分两种情况加以讨论。若要求检测时间之和最小，给出了最优算法；若要求最大检测时间最小，证明了其是 NP 完全问题，并给出近似算法。最后给出一个实例对算法加以说明。

关键词：故障；检测；NP 完全；近似算法；多项式算法

中图分类号：O224.1　　**文献标识码**：A

The Problem of Choosing the Testing Points Based on One Port Mode

DAI Li，YU Dian-long，XIE Zheng

（College of Science，National Univ. of Defense Technology，Changsha 410073，China）

Abstract：During the course of diagnosing the faults，the time for each testing point to detect the fault may be different. The problem that each testing point can detect once all the fault points that it can detect has been solved. If each testing point can detect only one fault point once，two questions will be discussed. If the sum of all the detection time is required to be minimal，an optimization algorithm is provided；If the maximum detection time is required to be minimal，the problem is proved to be NP-completeness and an approximate algorithm is given. Finally an example is given to explain the algorithms.

Key words：fault；detection；NP-completeness；approximate algorithm；polynomial algorithm

1　模型

测试点选取在故障诊断中是一个不容忽视的问题，选择良好的测试集可以使测试代价和所占用的测试资源消耗达到最小。测试点选取问题已经在工程上得到了良好的应用[1]。在检测故障过程中，设备选测试集为 $T=\{x_1, x_2, \cdots, x_n\}$，$x_i$（$i=1, 2, \cdots, n$）为测试点，所需检测的故障集为 $F=\{y_1, y_2, \cdots, y_m\}$，$y_j$（$j=1, 2, \cdots, m$）为故障点。$x_i$ 检测 y_j 所需的时间为 t_{ij}。各个测试相互独立，且每个测试只有“正常”、“异常”两种情况，分别用“1”和“0”表示。布尔矩阵 $R_{TF}=(r_{ij})_{n\times m}$为稀疏矩阵，表示集合 T 和 F 的关系矩阵，其中 $r_{ij}=1 \Leftrightarrow x_i$ 可以检测 y_j。$\forall x_i \in T$，称 $N(x_i)=\{y_j \mid r_{ij}=1, 1\leqslant j\leqslant m\}$ 为 x_i 可检测的故障集。若 $\forall i\in\{1, 2, \cdots, n\}$，对于 x_i 可以检测的所有故障点 y_j，令 $t_{ij}=c_i$，如果 x_i 在 c_i 时间内可以一次性将 $N(x_i)$ 中的故障点都检测完，我们称之为 all－port 模式。文献[2]解决了基于 all－port 模式，并且所有检测时间之和最小的测试点选取问题。如果 x_i 一次只能检测 $N(x_i)$ 中的一个故障点，称之为 one－port 模式。对于 one－port 模式，要研究如何选取最佳测试点集 A，使得目标函数值最小。显然目标函数不同，算法也不同。我们给出两个目标函数加以讨论，一个是所有检测时间之和，另一个是最大检测时间。为了便于讨论，构造赋权二部图 $G=(X, Y; E, w)$，其中 $X=\{x_1, x_2, \cdots, x_n\}$，$Y=\{y_1, y_2, \cdots, y_m\}$，如果 x_i 可以检测 y_j，则在 x_i 和 y_j 之间连一条边，并且对边 x_iy_j 赋权为 t_{ij}。假设 Y 中没有孤立点。

※ 收稿日期：2002－01－11
作者简介：戴丽（1974－），女，助教，博士。

2 目标函数为 A 中测试点所有检测时间之和

设最佳测试点集为 A，对于 $x_i \in A$，假设 x_i 检测 y_{i_1}，y_{i_2}，…，y_{i_l}，则 $f=\sum_{x_i \in A}\sum_{j=1}^{l} t_{ii_j}$。令 $N(y_j)=\{x_i \mid x_i \in X,\ x_iy_j \in E\}$ 表示 y_j 的邻点集。

定理 1 设二部图 $G=(X, Y; E)$，$\forall y_j \in Y$，若 $t_{j_0j}=\min\limits_{x_i \in N(y_j)} t_{ij}$，则令测试点 x_{j_0} 检测 y_j，将集合 $\{x_{1_0}, x_{2_0}, \cdots, x_{m_0}\}$ 中重复的元素只保留一个得到集合 A 即为最佳测试点集，即此时目标函数 $f=\sum_{x_i \in A}\sum_{j=1}^{l} t_{ij}$ 最小，而 l 恰好是 x_i 检测的次数。

证 由于测试点检测时间之和与 Y 中顶点被检测的时间之和相等。因此只要证明以 A 为测试点集，Y 中顶点被检测的时间之和最小即可。假设任意一组测试点（x_1'，x_2'，…，x_m'）中 x_j' 检测顶点 y_j（$j=1$，…，m），由于 $t_{j_0j}=\min\limits_{x_i \in N(y_j)} t_{ij}$，故 $t_{j_0j} \leqslant t_{j'j}$

所以
$$\sum_{j=1}^{m} t_{j_0j} \leqslant \sum_{j=1}^{m} t_{j'j}$$

而
$$\sum_{x_i \in A}\sum_{j=1}^{l} t_{ii_j} = \sum_{j=1}^{m} t_{j_0j}$$

故
$$f=\sum_{x_i \in A}\sum_{j=1}^{l} t_{ii_j} \leqslant \sum_{j=1}^{m} t_{j'j}$$

由定理 1 可得到如下最优算法。

Step0（初始化置）$A=\varnothing$，$j=1$。

Step1（寻找 y_j 对应的测试点）

若 $j>m$，则算法结束。否则，在 X 中取一个顶点 x_{j_0} 满足

$$t_{j_0j}=t(x_{j_0}y_j)=\min_{x_i \in N(y_j)} t_{ij}$$

Step2 置 $A=A \cup \{x_{j_0}\}$，令 $j=j+1$，转 Step1。

此时目标函数 $f=\sum_{x_i \in A}\sum_{j=1}^{l} t_{ij}$。可以估计该算法的复杂性：$\forall y_j \in Y$，至多需要 n 次比较，而 $|Y|=m$，故算法复杂性为 $O(mn)$。

3 目标函数为 A 中测试点最大检测时间

设最佳测试点集为 A，设 $x_i \in A$，x_i 检测 y_{i_1}，y_{i_2}，…，y_{i_l}，则 $f=\max\limits_{x_i \in A}\{\sum_{j=1}^{l} t(x_iy_{i_j}) \mid x_iy_{i_j} \in E\}$。要使 f 最小即在图 G 中找一个子图 H 满足

$$\max_{i \in \{1,\cdots,n\}}\{\sum t(x_iy_j) \mid x_iy_j \in E(H)\} = \min_{L \subseteq G}\max_{i \in \{1,\cdots,n\}}\{\sum t(x_iy_j) \mid x_iy_j \in E(L)\}。$$

称此问题为 DOPM（diagnosed in one port mode），首先证明 DOPM 是 NP 完全的。

DOPM 的判定问题是：

给定二部图 $G=(X, Y; E)$，$t_{ij} \equiv 1$ 为定义在边集 E 上的权（此处我们限制 $t \equiv 1$，即为原问题的一个特殊情形），$K \in N$。

问：是否存在子图 H 使

$$d_H(y_j) \geqslant 1\ (j=1, \cdots, m) \text{ 且 } \max_{i \in \{1,\cdots,n\}}\{\sum t(x_iy_j) \mid x_iy_j \in E(H)\} \leqslant K$$

我们将证明 DOPM 与一种任务安排问题（记作 WA，work arrangement）等价。而该任务安排问题是 NP 完全问题[3]，WA 问题的具体叙述如下：

给定任务集 A，完成任务 $a\in A$ 所需的工作时间 $l(a)\in Z^+$，机器数 s，截止时间 $D\in Z^+$。

问：是否存在一种安排，使用权 A 的所有任务都有在截止时刻前完成，即是否存在 A 的一个划分 $A=A_1\cup A_2\cup\cdots\cup A_s$，满足

$$\max\left\{\sum_{a\in A_i} l(a) \mid 1\leqslant i\leqslant s\right\}\leqslant D$$

定理 2　DOPM 是 NP 难题。

证　显然 DOPM∈NP，对于任一个给定的 G 子图 L，总是可以在多项式时间内查明是否有 $\max\limits_{i\in\{1,\cdots,n\}}\left\{\sum t(x_iy_j) \mid x_iy_j\in E(L)\right\}\leqslant K$ 成立。

证 DOPM∝WA。

给定 DOPM 的一个实例：$X=\{x_1, x_2, \cdots, x_n\}$，$Y=\{y_1, y_2, \cdots, y_m\}$，二部图 $G=(X, Y; E)$，$t_{ij}\equiv 1$ 为定义在边集 E 上的权（此处我们限制 $t\equiv 1$，即为原问题的一个特殊情形），$K\in N$。

构造 WA 的一个实例：任务集 $A=\{a_1, a_2, \cdots, a_m\}$，机器数为 n，对任意的 $a\in A$ 所需的工作时间 $l(a)=1$，取 $D=K$。

一方面，若 DOPM 的实例有解，即存在二部图 G 的一个子图 H，使得

$$\mathrm{d}_H(y_j)\geqslant 1\ (j=1, \cdots, m)\ \text{且} \max_{i\in\{1,\cdots,n\}}\left\{\sum t(x_iy_j) \mid x_iy_j\in E(H)\right\}\leqslant K$$

由于 $\omega\equiv 1$，故上式即为

$$\max_{i\in\{1,\cdots,n\}}\{\mathrm{d}_H(x_i)\}\leqslant K$$

令 $A_i=\{a_j \mid x_iy_j\in E(H)\}\ (i=1, \cdots, n)$，则 $\sum\limits_{a\in A_i} l(a)=\mathrm{d}_H(x_i)$。

由 $\mathrm{d}_H(y_j)\geqslant 1\ (j=1, \cdots, m)$ 知 $A=A_1\cup A_2\cup\cdots\cup A_n$，由 $\max\limits_{i\in\{1,\cdots,n\}}\{\mathrm{d}_H(x_i)\}\leqslant K$ 知

$$\max\left\{\sum_{a\in A_i} l(a) \mid 1\leqslant i\leqslant n\right\}\leqslant D=K$$

于是，WA 的实例也有解。

另一方面，若 WA 的实例有解，即任务集 A 有分划 $A=A_1\cup A_2\cup\cdots\cup A_n$，使得

$$\max\left\{\sum_{a\in A_i} l(a) \mid 1\leqslant i\leqslant n\right\}\leqslant D=K$$

则构造二部图 G 的子图 H 如下：

$$V(H)=V(G)$$

$$E(H)=\{x_iy_j \mid a_j\in A_i,\ i=1, \cdots, n;\ j=1, \cdots, m\}$$

由 $\max\left\{\sum\limits_{a\in A_i} l(a) \mid 1\leqslant i\leqslant n\right\}\leqslant D=K$ 知 $\max\limits_{i\in\{1,\cdots,n\}}\{d_H(x_i)\}\leqslant K$，即

$$\max_{i\in\{1,\cdots,n\}}\left\{\sum \omega(x_iy_j) \mid x_iy_j\in E(H)\right\}\leqslant K$$

由 $A=A_1\cup A_2\cup\cdots\cup A_n$ 知 $d_H(y_j)\geqslant 1\ (j=1, \cdots, m)$，于是，DOPM 的实例也有解。

综上两方面，我们证明了 DOPM 可以多项式变换到 WA（任务安排问题）。从而证明了，DOPM 是 NP 难题。

既然 DOPM 是 NP 难题，一般说来，对于这类问题是没有有效算法的（即多项式算法）。因此我们给出一个近似算法，并估计由该算法得到的近似解与精确解的接近程度。

算法思想是在二部图 $G=(X, Y; E)$ 中，对 Y 中顶点 $y_1, y_2, \cdots, y_m$ 依次确定它们对应的测试点。当 $y_1, y_2, \cdots, y_{j-1}$ 都选好各自的测试点后，对于 y_j，我们总是用检测时间增加最小的测试点来检测 y_j。

FF（first fit）近似算法：

Step0（初始化置）$u(x_1)=u(x_2)=\cdots u(x_n)=0$，$N'(x_1)=\cdots N'(x_n)=\varnothing$，$j=1$。

Step1（寻找 y_j 对应的测试点）

若 $j>m$，则算法结束。否则，在 X 中取一个顶点 x_k 满足，

$$t(x_k y_j)=\min\{t(x_i y_j)+u(x_i)\mid x_i y_j\in E\}$$

Step2 置 $N'(x_k)=N'(x_k)\cup\{y_j\}$，$u(x_k)=u(x_k)+t(x_k y_j)$

令 $j=j+1$，转 Step1。

此时目标函数 $f=\max\{u(x_1),\cdots,u(x_n)\}$。

FF 算法复杂性估计：显然该算法最多循环 m 次，每次循环中，Step1 中要比较 n 次，因此 FF 算法的复杂性为 $O(mm)$，其中 $n=|X|$，$m=|Y|$。

定理 3 设 $t_{\max}=\max\{t(e)\mid e\in E\}$，$t_{\min}=\min\{t(e)\mid e\in E\}$，对任意实例有

$$\frac{FE(I)}{OPT(I)}\leqslant m\frac{t_{\max}}{t_{\min}}\text{，其中 } m=|Y|$$

证 设 u_i 表示由 FF 算法所确定的 y_1，y_2，…，y_i 被检测的当前时间，由 Step1 可知

$$u_{i+1}\leqslant u_i+t_{\max}$$

因此对于任意实例有

$$FF(I)\leqslant mt_{\max}$$

显然

$$OPT(I)\geqslant t_{\min}$$

故有

$$FF(I)\leqslant mt_{\max}\leqslant m\frac{t_{\max}}{t_{\min}}OPT(I)\text{。}$$

4 算例

例 在二部图 $G=(X,Y;E,w)$ 中（如图 1），用前面讨论过的两种目标函数的算法选取相对应的最佳测试点集。

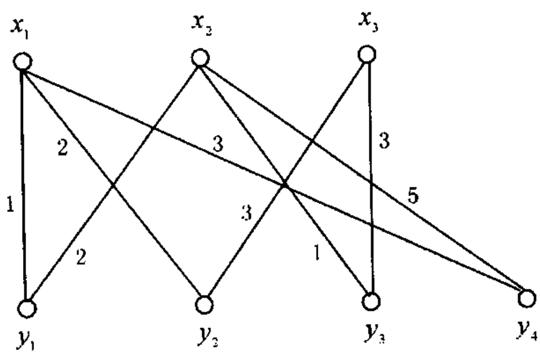

图 1 二部图

Fig. 1 Diagram of example

解 (1) 对于目标函数为测试点检测时间之和，我们选取最佳测试点集如下。

$t_{11}=\min\{t_{11},t_{21}\}=1$，选取测试点 x_1 检测 y_1；

$t_{12}=\min\{t_{12},t_{32}\}=2$，选取测试点 x_1 检测 y_2；

$t_{13}=\min\{t_{23},t_{33}\}=1$，选取测试点 x_2 检测 y_3；

$t_{14}=\min\{t_{14},t_{24}\}=3$，选取测试点 x_1 检测 y_4；

最佳测试点集 $A=\{x_1,x_2\}$，其中 x_1 检测 y_1，y_2，y_4；x_2 检测 y_3。

目标函数 $f=t_{11}+t_{12}+t_{23}+t_{14}=7$。

(2) 对于目标函数为测试点最大工作时间，我们选取最佳测试点集如下。

初始化：$u(x_1)=u(x_2)=u(x_3)=0$，$N'(x_1)=N'(x_2)=N'(x_3)=\varnothing$

寻找 y_1 对应的测试点：

$$\min\{t(x_iy_1)+u(x_i)\mid x_iy_1\in E\}=\min\{t_{11},t_{21}\}=t_{11}=1$$

$$N'(x_1)=N'(x_1)\cup\{y_1\}=\{y_1\},\ u(x_1)=u(x_1)+t(x_1y_1)=1$$

寻找 y_2 对应的测试点：

$$\min\{t(x_iy_2)+u(x_i)\mid x_iy_2\in E\}=\min\{t_{12}+1,t_{32}\}=t_{32}=3$$

$$N'(x_3)=N'(x_3)\cup\{y_2\}=\{y_2\},\ u(x_3)=u(x_3)+t(x_3y_2)=3$$

寻找 y_3 对应的测试点：

$$\min\{t(x_iy_3)+u(x_i)\mid x_iy_3\in E\}=\min\{t_{23},t_{33}+3\}=t_{23}=1$$

$$N'(x_2)=N'(x_2)\cup\{y_3\}=\{y_3\},\ u(x_2)=u(x_2)+t(x_2y_3)=1$$

寻找 y_4 对应的测试点：

$$\min\{t(x_iy_4)+u(x_i)\mid x_iy_4\in E\}=\min\{t_{14}+u(x_1),t_{24}+u(x_2)\}=t_{14}+u(x_1)=4$$

$$N'(x_1)=N'(x_1)\cup\{y_4\}=\{y_1,y_4\},\ u(x_1)=u(x_1)+t(x_1y_4)=4$$

最佳测试点集为 $A=\{x_1,x_2,x_3\}$，其中 x_1 先检测 y_1，然后检测 y_4；x_2 检测 y_3；x_3 检测 y_2；目标函数 $f=\max\{u(x_1),u(x_2),u(x_3)\}=4$。

参考文献：

[1] 温熙森，徐永成，易小山等. 智能机内测试理论与应用［M］. 北京：国防工业出版社，2002.

[2] 谢政，郁殿龙. 测试点的选取问题［J］. 国防科技大学学报，2001，23（1）：93－96.

[3] 谢政，李建平. 网络算法与复杂性理论［M］. 长沙：国防科技大学出版社，1995.

[4] Garey M R，Johnson D S. Computers and Intractability：A Guide to the Theory of NP－Completeness［M］. Freeman，1979.