

软件可靠性评估的重要抽样方法^{*}

张德平^{1,2+}, 聂长海^{3,4}, 徐宝文^{1,3,4}

¹(东南大学 计算机科学与工程学院,江苏 南京 210096)

²(南京航空航天大学 理学院,江苏 南京 210016)

³(南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093)

⁴(南京大学 计算机科学与技术系,江苏 南京 210093)

Importance Sampling Method of Software Reliability Estimation

ZHANG De-Ping^{1,2+}, NIE Chang-Hai^{3,4}, XU Bao-Wen^{1,3,4}

¹(Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

²(College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

³(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

⁴(Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China)

+ Corresponding author: E-mail: depingzhang@nuaa.edu.cn

Zhang DP, Nie CH, Xu BW. Importance sampling method of software reliability estimation. Journal of Software, 2009,20(10):2859–2866. <http://www.jos.org.cn/1000-9825/3553.htm>

Abstract: This paper proposes an effective method for computing optimal state transition probabilities for software reliability estimation based on a Markov usage model. This method uses Cross-Entropy to measure the differences between the operational profile and the sampling distribution with zero variance. By adjusting the probabilities of state transitions during test, an iterative method based on the Cross-Entropy is proposed for this choice, and an unbiased reliability estimator with zero variance is obtained. Simulation results show that the testing profile with Cross-Entropy method performs significantly better than the simulated annealing algorithm. Moreover, this strategy can more effectively accelerate software statistical testing.

Key words: software reliability; statistical testing; Markov usage model; importance sampling; cross-entropy method

摘要: 基于统计测试的马尔可夫使用模型对软件可靠性评估提出了一种有效的估计方法。该方法利用重要抽样技术在保证可靠性估计无偏性的条件下,利用交叉熵度量操作剖面与零方差抽样分布之间的差异,通过启发式迭代过程调整各个状态之间的转移概率来修正测试剖面。从理论上证明了利用修正测试剖面测试估计的可靠性是方差为0的无偏估计。最后给出了软件可靠性估计的最优测试剖面生成的启发式迭代算法。仿真结果表明,该方法与模拟退火算法相比,能够明显降低估计的方差,在提高估计精度的同时加快统计测试速度。

* Supported by the National Natural Science Foundation of China under Grant Nos.60773104, 60721002, 90818027 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2008AA01Z143, 2009AA01Z147 (国家高技术研究发展计划(863))

Received 2008-11-28; Accepted 2008-12-30

关键词: 软件可靠性;统计测试;马尔可夫使用模型;重要抽样;交叉熵方法

中图法分类号: TP311 文献标识码: A

统计测试是一种随机测试方法,它基于软件使用模型生成测试用例进行测试,根据测试结果进行统计分析、评估软件质量,是高可靠性软件开发的重要组成部分.统计测试假定每个使用的数据输入都具有相同的缺陷检测率,根据转移概率在各个状态下选择下一操作,对使用频繁的操作进行更多的测试,能够有效地检测出那些对软件可靠性影响较大的软件缺陷.因此,统计测试充分性的判定准则一般都基于测试使用与实际使用的差异程度,以确保根据测试结果估计出的可靠性能够代表软件实际使用的可靠性^[1,4,5].这对于特定操作环境是正确的,但却无法贯穿整个软件系统的全部操作集合.典型的例子是高可信性软件,特别是安全关键软件,其关键操作,如核电系统的紧急停堆处理,由于使用概率非常小,在统计测试中往往得不到充分测试,而这类操作的可靠性指标往往很高,其失效经常会产生严重后果.为了充分测试这些关键操作,获得软件可靠性的无偏估计,需要执行大量的统计测试用例,从而导致统计测试代价过高.基于此,许多统计测试方法被提了出来并进行理论研究^[4-10],其中大多数为“控制”统计测试方法^[6-9],主要利用重要抽样技术,在保证软件可靠性或失效风险估计为无偏估计的前提下,通过修正操作剖面来控制估计的方差,提高了估计的准确性.利用重要抽样方法精确估计软件可靠性的关键和难点在于抽样分布的选取,即如何确定一个最优的测试剖面,使得按最优测试剖面生成测试用例运行后所估计的方差达到最小.

1 马尔可夫链使用模型的可靠性度量

马尔可夫链使用模型是一个具有唯一初态和终态的马尔可夫链,可用强连通有向图 $G=(V,A)$ 和函数 $p:V \times V \rightarrow [0,1]$ 来表示,它具有如下性质:

- $V=\{1,2,\dots,n\}$ 是节点集,表示软件系统的使用状态;
- A 为边集,其元素表示在某个状态下选定某个操作时软件状态间的转移.从状态 i 到状态 j 的边 e 定义为一个有序对 (i,j) ,任意两个状态 i 和 j 之间的一个方向最多只有一条有向边相连;
- 转移概率 $p(i,j)$ 满足 $0 \leq p(i,j) \leq 1$,表示在程序投入运行阶段,从当前状态 i 转移到状态 j 的概率.

定义操作剖面 P 为软件系统在实际运行中的各个操作被执行概率,即 $P=(p(i,j))_{n \times n}$. 测试剖面 Q 为软件系统在测试过程中各个操作被执行概率.假定状态 1 为初态,状态 n 为终态,并且状态 n 为吸收态,它表示一旦进入状态 n 就不再离去,即 $p(n,n)=1, p(n,j)=0, \forall j \neq n$. 进一步假定每一个状态 $i \in V$ 都是从初态可达的,即在 G 中总存在一条从状态 1 到状态 i 的有向路径.软件的一次使用对应于马尔可夫使用模型从初态 1 到终止状态(包括终态 n 和失效状态)的一条遍历路径,记为 $\mathbf{x}=(x_1, x_2, \dots, x_L)$, L 为一次使用过程中执行操作的总数,即路径长度. x_i 是一次使用过程中遍历的第 i 个操作, x_L 可以是终态 n ,也可以是失效状态.操作 x_i 执行后再执行的下一操作 x_j 根据转移概率 $p(x_i, x_j)$ 随机选择.因此,在软件一次使用过程中,路径 $\mathbf{x}=(x_1, x_2, \dots, x_L)$ 的执行概率为

$$f(\mathbf{x}; P) = P\{X = \mathbf{x}\} = \prod_{i=1}^L p(x_i, x_j) \text{ 满足 } \sum_j p(x_i, x_j) = 1,$$

其中, \mathbf{x} 是按操作剖面 P 随机生成的执行路径, x_j 为与状态 x_i 相邻的可一步转移到的状态,即 $p(x_i, x_j) > 0$. $\mathbf{X}=(X_1, X_2, \dots)$ 是表示执行路径的随机向量.基于马尔可夫链使用模型的软件统计测试的每个测试用例都对应于上述一条遍历路径 \mathbf{x} . 记示性函数 $I_f(\mathbf{x})$ 为

$$I_f(\mathbf{x}) = \begin{cases} 1, & \text{如果路径 } \mathbf{x} \text{ 执行过程中失效} \\ 0, & \text{否则} \end{cases}$$

显然,系统是否正确运行依赖于软件系统的规格说明书,对于给定的软件系统 π ,其规格说明书是给定的.定义软件可靠性 R 为软件一次使用过程中无失效发生的概率,则由 $I_f(\mathbf{X})$ 定义可知,系统 π 的可靠性 $R(\pi)$ 为

$$R(\pi) = 1 - E_p[I_f(\mathbf{X})] = 1 - \ell \quad (1)$$

其中, E_P 是相对于概率分布 $f(\mathbf{x}; P)$ 求期望, ℓ 为系统的失效概率, 即 $\ell = E_P[I_f(\mathbf{X})]$.

注意到, 当沿某条含有圈的路径进行测试时, 至少有一个操作(记为 C)在测试中多次出现. 这样, 随着操作 C 执行次数的增加, 这条测试路径失效概率便趋近于 1. 如果执行的绝大多数路径中都出现圈, 则当操作重复执行的次数足够大时, 就会导致整个软件系统可靠性估计过低. 为了避免可靠性估计过于悲观, 这里假定, 当操作重复出现时只认为出现 1 次^[11,12].

2 基于交叉熵的重要抽样方法

由公式(1)可知, 获得系统可靠性无偏估计最直接的方法是: 根据马尔可夫链使用模型选出 N 条遍历路径 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$, 由矩估计直接可得失效概率的无偏估计:

$$\hat{\ell} = \frac{1}{N} \sum_{i=1}^N I_f(\mathbf{x}_i) \quad (2)$$

由此可得系统可靠性的无偏估计 $\hat{R}(\pi)$.

然而对于安全关键软件系统, 软件失效不易观察到. 如果要得到较为精确的可靠性估计, 需要测试的路径数(抽样样本数) N 要足够大, 这样必然会导致测试的开销过大. 为了克服这种方法的不足, 一般采用重要抽样方法: 通过选择合适的抽样分布即测试剖面 Q 进行测试, 增加稀有状态遍历的概率, 提高软件失效被观察的机会. 为保证按测试剖面进行的估计为无偏估计, 常采用似然比率进行加权调节, 因为

$$\ell = E_P[I_f(\mathbf{X})] = \int I_f(\mathbf{x}) \frac{f(\mathbf{x}; P)}{f(\mathbf{x}; Q)} f(\mathbf{x}; Q) d\mathbf{x} = \int I_f(\mathbf{x}) W(\mathbf{x}, P, Q) f(\mathbf{x}; Q) d\mathbf{x} \quad (3)$$

其中, $W(\mathbf{x}, P, Q) = f(\mathbf{x}; P) / f(\mathbf{x}; Q)$ 为似然比率. 所以, 失效概率 ℓ 的无偏估计 $\hat{\ell}$ 可表示为

$$\hat{\ell} = \frac{1}{N} \sum_{i=1}^N I_f(\mathbf{x}_i) W(\mathbf{x}_i, P, Q) \quad (4)$$

其中, \mathbf{x}_i 为按测试剖面 Q 随机生成测试用例.

注意到, 在公式(4)中, 当最优测试剖面 Q^* 满足如下公式(5)的形式时, 只需按最优测试剖面 Q^* 随机生成一条测试路径即可获得失效概率 ℓ 方差为 0 的无偏估计, 即满足公式(5)的测试剖面为零方差的抽样分布.

$$f(\mathbf{x}; Q^*) = \frac{I_f(\mathbf{x}) \cdot f(\mathbf{x}; P)}{\ell} \quad (5)$$

而满足公式(5)的最优测试剖面 Q^* 是真值 ℓ 的函数, 实际应用中并不知道. 因此, 使用 Q^* 进行抽样并不现实. 由 $f(\mathbf{x}; Q^*)$ 的形式可知, 在 $f(\cdot; P)$ 的概率分布簇 $\{f(\cdot; Q)\}$ 找到一个最接近 $f(\mathbf{x}; Q^*)$ 的概率分布 $f(\cdot; Q)$, 由此确定参数 Q , 即最优(或渐近最优)测试剖面, 这样便可得到方差为 0 或接近于 0 的无偏估计. 常用来描述两个概率分布差异值的是 K-L 距离或交叉熵^[13,14].

这样, 最优测试剖面的确定问题就转化为确定推断参数 Q , 使得 $f(\mathbf{x}; Q)$ 与概率分布 $f(\mathbf{x}; Q^*)$ 的交叉熵最小, 即

$$Q^* = \arg \min_Q \left\{ - \int f(\mathbf{x}; Q^*) \ln f(\mathbf{x}; Q) d\mathbf{x} \right\} \quad (6)$$

将公式(5)代入公式(6)可得如下等价推断参数确定形式:

$$Q^* = \arg \max_Q \left\{ \int I_f(\mathbf{x}) f(\mathbf{x}; P) \ln f(\mathbf{x}; Q) d\mathbf{x} \right\} = \arg \max_Q \{ E_P [I_f(\mathbf{X}) \ln f(\mathbf{X}; Q)] \} \quad (7)$$

注意到 $f(\mathbf{x}; Q) = \prod_{i=1}^L q(x_i, x_j)$ 满足 $\sum_j q(x_i, x_j) = 1$, 则由拉格朗日乘法可知, 公式(7)可以转化为如下优化问题来求解:

$$\max_Q \left\{ E_P \left[I_f(\mathbf{X}) \sum_i \ln q(X_i, X_{i+1}) \ln f(\mathbf{X}; Q) \right] + \sum_i u_i \left(\sum_j q(X_i, X_j) - 1 \right) \right\} \quad (8)$$

其中, u_i 为拉格朗日因子. 上式关于 $q(x_i, x_m)$ 求偏导, 令其等于 0, 可得 Q^* 的元素 $q(x_i, x_m)$ 为

$$q(x_i, x_m) = E_p \left[\mathbf{I}_f(\mathbf{X}) \sum_i \mathbf{I}_{\{X_i=x_i, X_{i+1}=x_m\}} \right] / E_p \left[\mathbf{I}_f(\mathbf{X}) \sum_i \mathbf{I}_{\{X_i=x_i\}} \right] \quad (9)$$

由此可得基于重要抽样方法估计软件可靠性的最优测试剖面。

3 软件可靠性估计的性质

性质 3.1. 如果 $f(\mathbf{x}; P) > 0$, 测试剖面 Q 的元素满足公式(9), 则由公式(4)及公式(1)估计出的可靠性估计是无偏估计。

证明: 因为 $\int \mathbf{I}_f(\mathbf{x}) W(\mathbf{x}, P, Q) f(\mathbf{x}; Q) d\mathbf{x} = \int \mathbf{I}_f(\mathbf{x}) \frac{f(\mathbf{x}, P)}{f(\mathbf{x}, Q)} f(\mathbf{x}; Q) d\mathbf{x} = E_p[\mathbf{I}_f(\mathbf{X})] = \ell$, 由此可知, 公式(4)为 ℓ 的无偏估计. 由 $R(\pi) = 1 - E_p[\mathbf{I}_f(\mathbf{X})] = 1 - \ell$ 可得可靠性 $R(\pi)$ 的无偏估计, 得证. \square

为了得到按元素满足公式(9)的测试剖面抽样估计方差性质, 需要利用重要抽样方法如下的一个性质:

引理 3.2. 设随机变(向)量 \mathbf{X} 具有概率密度函数 $f(x) > 0, x \in R^1$, 积分 $I = \int h(x) dx$ 重要抽样估计为

$$\hat{g}(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^N \frac{h(X_i)}{f(X_i)}, X_i \sim f(x) \quad (10)$$

如果对于 $\forall x \in \{x: f(x) > 0\}$ 存在常数 c 满足 $h(x) = cf(x)$, 则重要抽样估计 $\hat{g}(\mathbf{X})$ 是积分 I 的方差为 0 的无偏估计.

证明: 显然, $\hat{g}(\mathbf{X})$ 为积分 I 的无偏估计. 记 $s(x) = h(x)/f(x)$, 则 $\hat{g}(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^N s(X_i)$, 由此可得 $\hat{g}(\mathbf{X})$ 的方差为

$$\text{Var}(\hat{g}(\mathbf{X})) = \text{Var}\left(\frac{1}{N} \sum_{i=1}^N s(X_i)\right) = \frac{\text{Var}(s(X_i))}{N} = \frac{1}{N} \int [s(x) - E(s(X))]^2 f(x) dx.$$

因为对于 $\forall x \in \{x: f(x) > 0\}$ 存在常数 c 满足 $h(x) = cf(x)$, 显然, $h(x)/f(x) = c$, 则有 $E[s(X)] = E[h(X)/f(X)] = c$. 因此可得 $\hat{g}(\mathbf{X})$ 的方差为 0, 得证. \square

利用引理 3.2, 类似于文献[15]可以得到:

性质 3.3. 如果 $f(\mathbf{x}; P) > 0$, 测试剖面 Q 的元素满足公式(9), 则由公式(4)及公式(1)估计出的可靠性是方差为 0 的无偏估计。

证明: 估计的无偏性由性质 3.1 保证, 下证估计的方差为 0. 易知, 要证可靠性估计的方差为 0, 只需证明失效概率的估计具有零方差. 由引理 3.2 可知, 如果失效概率估计的似然比率函数 $W(\mathbf{x}, P, Q)$ 为常数, 即可证得估计的方差为 0. 假定 $\mathbf{x} = (x_1, x_2, \dots, x_L, x_{L+1})$ 为一次测试执行过程中最终观察到失效的测试路径, 则按操作剖面 P 选择路径 \mathbf{x} 执行的概率为 $f(\mathbf{x}; P) = \prod_{i=1}^L p(x_i, x_{i+1})$, 按测试剖面 Q 选择路径 \mathbf{x} 执行的概率为 $f(\mathbf{x}; Q) = \prod_{i=1}^L q(x_i, x_{i+1})$. 由示性函数 $\mathbf{I}_f(\mathbf{X})$ 的性质及状态转移的马尔可夫性, 公式(9)的分子可表示为

$$\begin{aligned} E_p \left[\mathbf{I}_f(\mathbf{X}) \sum_{i=1}^L \mathbf{I}_{\{X_i=x_i, X_{i+1}=x_m\}} \right] &= \sum_{i=1}^L E_p \left[\mathbf{I}_f(\mathbf{X}) \mathbf{I}_{\{X_i=x_i, X_{i+1}=x_m\}} \right] = \sum_{i=1}^L E \{ \mathbf{I}_f(\mathbf{X}) = 1, X_i = x_i, X_{i+1} = x_m \} \\ &= \sum_{i=1}^L P \{ \mathbf{I}_f(\mathbf{X}) = 1 | X_{i+1} = x_m \} P \{ X_{i+1} = x_m | X_i = x_i \} P \{ X_i = x_i \} \\ &= \sum_{i=1}^L r_m P(x_i, x_m) P \{ X_i = x_i \} \end{aligned} \quad (11)$$

其中, r_m 表示系统从状态 m 出发, 在进入终态之前观察到失效发生的概率. 类似地, 公式(9)的分母可以表示为

$$E_p \left[\mathbf{I}_f(\mathbf{X}) \sum_{i=1}^L \mathbf{I}_{\{X_i=x_i\}} \right] = \sum_{i=1}^L r_i \cdot P \{ X_i = x_i \} \quad (12)$$

将公式(11)和公式(12)分别代入公式(9), 可得转移概率:

$$q(x_i, x_m) = \frac{\sum_{i=1}^L r_m P(x_i, x_m) P\{X_i = x_i\}}{\sum_{i=1}^L r_i \cdot P\{X_i = x_i\}} = \frac{r_m P(x_i, x_m)}{r_i} \quad (13)$$

由此可得:

$$f(\mathbf{x}; Q) = \prod_{i=1}^L q(x_i, x_{i+1}) = \frac{r_2 P(x_1, x_2)}{r_{x_1}} \cdot \frac{r_3 P(x_2, x_3)}{r_{x_2}} \cdot \dots \cdot \frac{r_{x_{L+1}} P(x_L, x_{L+1})}{r_L} \quad (14)$$

因此,似然比率 $W(\mathbf{x}, P, Q)$ 可以表示为

$$W(\mathbf{x}, P, Q) = \frac{f(\mathbf{x}; P)}{f(\mathbf{x}; Q)} = \frac{r_{x_1}}{r_{L+1}} \quad (15)$$

注意到 x_1 为初态, x_{L+1} 为失效状态. 由于 \mathbf{x} 是测试执行过程中最终观察到失效的测试路径, 则必有 $r_{L+1}=1$. 故测试路径 \mathbf{x} 的似然比率为 r_1 . 由 r_1 的定义可知, r_1 是独立于测试路径的常数. 又因为公式(9)中所有的样本均是最终观察到失效的测试路径 $I_f(\mathbf{x})=1$, 所以每条测试路径都有相同的常数似然比率 r_1 . 由引理 3.2 可知, $\hat{\pi}$ 估计的方差为 0, 进一步可得可靠性 $R(\pi)$ 估计的方差也为 0, 得证. \square

4 最优测试剖面生成

注意到由公式(9)生成最优测试剖面 Q 的抽样分布是操作剖面 P , 对于安全关键软件的关键操作, 由于在实际使用的概率较小, 按 P 抽样时很难观察到失效. 这里再次采用重要抽样方法, 通过一种递归算法生成一个测试剖面序列 $\{Q_0^*, Q_1^*, \dots\}$, 利用上一迭代得到的近似测试剖面生成下一代测试剖面来得新的测试剖面, 直到测试剖面序列收敛为止.

实际上, 在迭代的第 1 步将 Q_0^* 初始化为操作剖面 P , 即 $Q_0^* = P$, 通过随机仿真, 获得公式(9)的第 1 个近似最优解 Q_1^* :

$$Q_1^* = \arg \max_Q \{E_P[I_f(\mathbf{X}) \ln f(\mathbf{X}; Q)]\} \approx \arg \max_Q \frac{1}{N} \sum_{i=1}^N I_f(\mathbf{x}_i) \ln f(\mathbf{x}_i; Q) \quad (16)$$

在第 2 次迭代过程中, N 条测试路径则按照 Q_1^* 抽样, 生成第 2 个近似最优解 Q_2^* :

$$\begin{aligned} Q_2^* &= \arg \max_Q \{E_P[I_f(\mathbf{X}) \ln f(\mathbf{X}; Q)]\} = \arg \max_Q \left\{ \int I_f(\mathbf{x}) f(\mathbf{x}; P) \frac{f(\mathbf{x}; Q_1^*)}{f(\mathbf{x}; Q_1^*)} \ln f(\mathbf{x}; Q) d\mathbf{x} \right\} \\ &= \arg \max_Q E_{Q_1^*} [I_f(\mathbf{X}) W(\mathbf{X}, P, Q_1^*) \ln f(\mathbf{X}; Q)] \approx \arg \max_Q \frac{1}{N} \sum_{i=1}^N I_f(\mathbf{x}_i) W(\mathbf{x}_i, P, Q_1^*) \ln f(\mathbf{x}_i; Q). \end{aligned}$$

由此可得第 j 次迭代中渐近最优测试剖面 Q_j^* 的计算式为

$$Q_j^* = \arg \max_Q \{E_P[I_f(\mathbf{X}) \ln f(\mathbf{X}; Q)]\} \approx \arg \max_Q \frac{1}{N} \sum_{i=1}^N I_f(\mathbf{x}_i) W(\mathbf{x}_i, P, Q_{j-1}^*) \ln f(\mathbf{x}_i; Q).$$

注意到 $f(\mathbf{x}; Q) = \prod_{i=1}^L q(x_i, x_j)$ 满足 $\sum_j q(x_i, x_j) = 1$, 则由拉格朗日乘法可将上式的解转化为如下最大值问题

$$\text{来求解: } \max_Q \left\{ E_{Q_{j-1}^*} \left[I_f(\mathbf{X}) W(\mathbf{X}, P, Q_{j-1}^*) \sum_i \ln q(X_i, X_j) \right] + \sum_i u_i \sum_j q(X_i, X_j) \right\}.$$

上式关于 $q(x_i, x_m)$ 求偏导, 并令其等于 0, 由此可得 Q_j^* 的元素 $q(x_i, x_m)$ 为

$$q(x_i, x_m) = \frac{E_{Q_{j-1}^*} \left[I_f(\mathbf{X}) W(\mathbf{X}, P, Q_{j-1}^*) \sum_i I_{\{X_i = x_i, X_{i+1} = x_m\}} \right]}{E_{Q_{j-1}^*} \left[I_f(\mathbf{X}) W(\mathbf{X}, P, Q_{j-1}^*) \sum_i I_{\{X_i = x_i\}} \right]} \quad (17)$$

因此可得 $q(x_i, x_m)$ 的估计式为

$$\hat{q}(x_i, x_m) = \frac{\sum_{k=1}^N I_f(x_k) W(x_k, P, Q_{j-1}) \sum_i I_{\{x_i=x_k, x_{i+1}=x_m\}}}{\sum_{k=1}^N I_f(x_k) W(x_k, P, Q_{j-1}) \sum_i I_{\{x_i=x_k\}}} \quad (18)$$

其中,测试路径 $x_k, k=1, 2, \dots, N$ 按测试剖面 Q_{j-1} 随机生成,示性函数 $I_{\{x_i=x_k\}}$ 取 1 时表示测试路径 x_k 执行了操作 x_i , 否则取 0; $I_{\{x_i=x_k, x_{i+1}=x_m\}}$ 取 1 时表示测试路径 x_k 在执行完成操作 x_i 后接着执行操作 x_m , 否则取 0.

由此可得到一种基于交叉熵的迭代算法求解最优或渐近最优测试剖面,然后利用此最优测试剖面采用重要抽样方法,估计出软件系统的可靠性:初始化测试剖面 Q_0 为操作剖面 P 随机生成测试路径,利用公式(18)修正测试剖面 Q_1 ,然后利用修正后的测试剖面 Q_1 再进行抽样,如此循环, ..., 直到测试剖面不再发生变化或变化不大时停止.具体算法如下:

算法 4.1. 估计软件可靠性的交叉熵算法.

1. 初始化测试剖面 \hat{Q}_0 为操作剖面 P , 迭代次数 $j=1$.
2. 根据测试剖面 \hat{Q}_{j-1} 生成测试路径 x_1, x_2, \dots, x_N , 观察其是否发生软件失效.
3. 由生成的 N 条测试路径 x_1, x_2, \dots, x_N , 由公式(18)求解出 \tilde{Q}_j , 应用平滑技术修正得到:

$$\hat{Q}_j = \alpha \tilde{Q}_j + (1 - \alpha) \hat{Q}_{j-1}, \alpha \in (0.4, 0.9).$$

4. 令 $j=j+1$, 重复步骤 2、步骤 3, 直到对于某个给定的 d (如 $d=4$) 有 $\hat{Q}_j = \hat{Q}_{j-1} = \dots = \hat{Q}_{j-d}$ 或相邻两次测试剖面最大变化量小于给定的 ϵ 为止.
5. 根据生成测试剖面 \hat{Q}_j 随机生成 M 条测试路径 x_1, x_2, \dots, x_M , 应用下式估计出软件可靠性:

$$\hat{R}(\pi) = 1 - \frac{1}{M} \sum_{i=1}^M \frac{f(x_i; P)}{f(x_i; \hat{Q}_j)} \cdot I_f(x_i).$$

在步骤 3 中,当利用公式(18)修正测试剖面时,若分母为 0, 则从当前状态转移到相邻其他状态的概率为上一次迭代中的相应转移概率不变.另外,步骤 3 中采用平滑技术主要是减少 \hat{Q}_j 的某些元素为 0, 避免算法在最初阶段就开始一直在局部最优解上搜索.

5 实例分析

为说明基于统计测试的软件可靠性评估方法,采用 Gutjahr^[7]给出的列车调度软件实例加以验证,如图 1 所示.软件包括 12 个操作,每次执行从操作 1 开始,操作 12 是软件终止运行时执行的操作,边(4,6),(8,9)和(8,10)导致关键操作 6,9,10 执行.各个操作失效的先验概率用向量 f 表示: $f=(0, 0.001, 0.001, 0.001, 0.001, 0.0001, 0.001, 0.001, 0.0001, 0.0001, 0.001, 0.001)$.

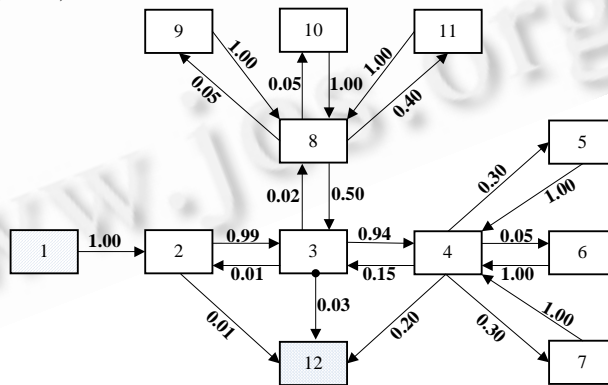


Fig.1 A Markov usage model from Gutjahr

图 1 Gutjahr 给出的马尔可夫链使用模型

采用模拟退火(SA)算法^[9]和交叉熵(CE)方法以及标准统计测试方法(Standard)(利用操作剖面 P)分别抽样 $N=2000$ 条测试路径仿真模拟测试 100 次,得到软件系统可靠性的 100 个估计值.3 种情形下的可靠性估计和方差分别如图 2 和图 3 所示.

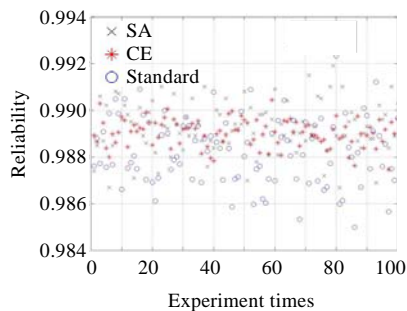


Fig.2 Reliability estimation of three different profiles

图 2 3 种不同剖面下可靠性估计

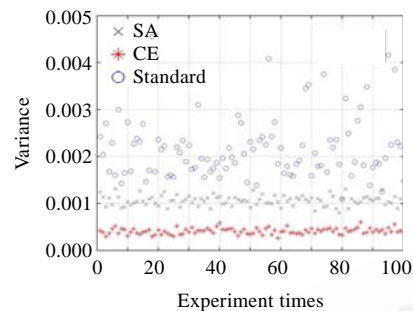


Fig.3 Variance of three different profiles

图 3 3 种不同剖面下可靠性估计的方差

在 100 次仿真中,CE 方法估计的可靠性平均值为 0.989 1,方差平均值为 4.36×10^{-4} ;SA 算法估计出的可靠性平均值为 0.989 3,方差平均值为 1.14×10^{-3} ;标准统计测试方法可靠性估计的平均值为 0.988 9,方差的平均值为 2.46×10^{-3} .由此可见,采用交叉熵方法较模拟退火算法和标准方法都能显著降低估计的方差.3 种情形下的盒形图如图 4 所示.

从盒形图可以看出,利用交叉熵方法估计出的可靠性比较集中于其中位数附近,模拟退火算法得到的估计值的分散程度相对要大一些,而采用标准方法得到的估计值的分散程度最大.

在 100 次仿真中,每次遍历路径为 2 000 条.采用标准方法仿真时,操作 6 被遍历的累积次数为 20 576 次,关键操作 9 和操作 10 被遍历的累积次数分别为 476 次和 514 次.采用交叉熵方法时,操作 6 被遍历的次数为 79 394 次,关键操作 9 和操作 10 被遍历的累积次数分别为 4 550 和 4 468 次.采用模拟退火算法时,操作 6 被遍历的累积次数为 55 090 次,关键操作 9 和操作 10 被遍历的累积次数分别为 10 898 和 8 162 次.由此可见,采用交叉熵方法和模拟退火算法都可明显提高稀有操作的遍历机会.交叉熵方法对关键操作 6 的遍历机会要大于模拟退火算法,而对关键操作 8 和操作 9 的遍历机会要低于模拟退火算法.

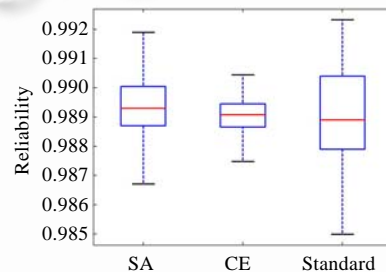


Fig.4 Box plot of three different profiles

图 4 3 种不同剖面下可靠性估计的盒形图

6 结论

本文基于重要抽样技术,利用交叉熵通过一种修正机制调节操作剖面,在保证可靠性估计是无偏估计的前提下降低估计的方差,充分发挥重要抽样技术的优点,增加使用概率小的关键操作测试机会,加速软件测试.该方法是把修正测试剖面归结为一个优化问题迭代求解,估计软件可靠性,加速软件测试.利用交叉熵方法求解最优测试剖面的不足之处在于:需要的仿真开销要比模拟退火算法及标准方法都要大,并且无法得到最优测试剖面的精确解,只能得到一个满足可靠性估计精度的近似最优解.

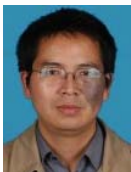
References:

- [1] Whittaker JA, Poore JH. Markov analysis of software specifications. ACM Trans. on Software Engineering and Method, 1994,2(1): 93-106.
- [2] Whittaker JA, Thomason MG. A Markov chain model for statistical software testing. IEEE Trans. on Software Engineering, 1994, 20(10):812-824.

- [3] Walton GH, Poore JH, Trammell J. Statistical testing of software based on a usage model. *Software-Practice and Experience*, 1995, 25(1):97-108.
- [4] Walton GH, Poore JH. Measuring complexity and coverage of software specifications. *Information and Software Technology*, 2000, 42(12):859-872.
- [5] Feng H, Xu XS, Wang J. The determination of resemblance between the usage chain and the test chain in statistical testing. *Computer Engineering & Science*, 2003,25(1):17-19 (in Chinese with English abstract).
- [6] Gutjahr WJ. Failure risk estimation via Markov software usage models. In: Schoitsch E, ed. *Proc. of the 15th Int'l Conf. on Computer Safety, Reliability and Security (SAFECOMP'96)*. Springer-Verlag, 1997. 183-192. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.27.8529&rep=rep1&type=pdf>
- [7] Gutjahr WJ. Importance sampling of test cases in Markovian software usage models. *Probability in the Engineering and Informational Sciences*, 1997,11:19-36. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.310&rep=rep1&type=pdf>
- [8] Doerner K, Laure E. High performance computing in the optimization of software test plans. *Optimization and Engineering*, 2002, 3(1):67-87.
- [9] Yan J, Wang J, Chen HW. Software statistical test acceleration based on importance sampling. *Computer Engineering & Science*, 2005,27(3):64-66 (in Chinese with English abstract).
- [10] Doerner K, Gutjahr WJ. Extracting test sequences from a Markov software usage model by ACO. LNCS 2724, Springer-Verlag, 2003. 2465-2476. <http://www.springerlink.com/content/q3teb3ueknhqueh1/fulltext.pdf>
- [11] Krishnamurthy S, Mathur AP. On the estimation of reliability of a software system using reliabilities of its components. In: *Proc of the 8th Int'l Symp. Software Reliability Engineering*. 1997. 146-155. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=630860&isnumber=13710>
- [12] Chari K, Hevner A. System test planning of software: An optimization approach. *IEEE Trans. on Software Engineering*, 2006,32(7): 503-509.
- [13] Zhang DP, Nie CH, Xu BW. Cross-Entropy method based on Markov decision process for optimal software testing. *Journal of Software*, 2008,19(10):2770-2779 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/2770.htm>
- [14] Boer DPT, Kroese DP, Mannor S, Rubinstein RY. A tutorial on the cross-entropy method. *Annals of Operations Research*, 2005, 134:19-67.
- [15] Boer DPT, Nicola VF. Adaptive state-dependent importance sampling simulation of Markovian queuing networks. *European Trans. on Telecommunication (Special Issue on Rare Event Simulation)*, 2002,13(4):303-315.

附中文参考文献:

- [5] 冯华,徐锡山,王戟. 统计测试中测试链与使用链的相似性判别. *计算机工程与科学*, 2003,25(1):17-19.
- [9] 颜炯,王戟,陈火旺. 基于重要抽样的软件统计测试加速. *计算机工程与科学*, 2005,27(3):64-66.
- [13] 张德平,聂长海,徐宝文. 基于马尔可夫决策过程用交叉熵方法优化软件测试的研究. *软件学报*, 2008,19(10):2770-2779. <http://www.jos.org.cn/1000-9825/19/2770.htm>



张德平(1973—),男,湖南澧县人,博士,讲师,CCF 学生会员,主要研究领域为软件测试技术,软件可靠性,数理统计与随机过程.



徐宝文(1961—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为程序设计语言,软件工程,并行与网络软件.



聂长海(1971—),男,博士,副教授,主要研究领域为软件测试技术,模糊信息处理,神经网络.