安全关键系统的软件可靠性评估方法

张德平[1,2] 徐宝文[2,3]

(南京航空航天大学计算机科学与与技术学院 南京 210016)[1]

(南京大学软件新技术国家重点实验室 南京 210093)[2] (南京大学计算机科学与技术系 南京 210093)[3]

摘 要 基于统计测试的Markov使用链模型对安全关键系统的可靠性估计提出了一种有效的方法。该方法利用重要抽样技术在保证估计的无偏性条件下,以可靠性估计的方差最小为目的,通过Ali-Silvey距离度量两个分布之间的差异,调整各个状态之间的转移概率分布,修正测试剖面,增加关键操作的遍历概率。最后给出了软件可靠性估计的最优测试剖面生成迭代算法。仿真结果表明,该方法能明显降低估计方差,在提高估计精度的同时能有效地加速统计测试。

关键词 软件可靠性,统计测试,Markov使用模型,重要抽样,Ali-Silvey距离

中图法分类号 TP311 **文献标识码** A

Estimation Method of Software Reliability for Safety-critical System

ZHANG De-ping[1,2] XU Bao-wen[2,3]

(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)[1]

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)[2]

(Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China)[3]

Abstract Importance sampling is a change-of-measure technique for speeding up the simulation of rare events in stochastic systems. In this paper we established a technique for computing optimal state transition probabilities for software reliability estimation based on a Markov usage model. By suitable changes of the probabilities of state transitions during test, an iterative method based on the Ali-Silvey distance was proposed for this choice. A learning algorithm for the computation of optimal transition probabilities of the Markov chain usage model was also presented and experimental results of this algorithm were reported.

Keywords Software reliability, Statistical testing, Markov usage model, Importance sampling, Ali-silvey distance

1 引言

在软件可靠性测试过程中,由于软件可靠性关注软件提供连续服务的能力,因此需要开发仿真软件使用方式的操作剖面,依据操作剖面产生测试用例并执行,以暴露对软件可靠性影响较大的软件缺陷,从而达到可靠性增长目的。在一个大的可信软件系统中,与安全关键功能相关的操作只是很小一部分,其使用频率一般都非常低,如核电紧急停堆系统软件可能一年甚至几年才运行一次,其原因是触发关键软件投入运行的紧急事件(如反应核的温度超过规定值)非常稀少。因而,安全关键软件所规定的失效率也非常低(一般在10^{-7}至10^{-9}失效/小时之间)。如采用传统的软件统计测试方法,即使花费很长的测试时间,也难有效地对该安全关键软件实施充分的测试而获得精确的可靠性估计。

基于此,许多统计测试方法被提出并研究[1-14],其中基于重要抽样技术的可靠性评估加速测试方法是目前研究的热点[4-12]。它主要基于统计测试,利用Markov链使用模型精确描述软件的使用,通过改变软件测试剖面,加大稀有事件发生的概率,达到减少测试成本,降低估计方差,提高评估精确度的目的。

2 可靠性度量模型

Markov链使用模型[4-7,10-12]是一个具有唯一初态和终态的Markov链,可用强连通有向图$G=(V,A)$和函数$p:V\times V\rightarrow[0,1]$表示,它具有如下性质:

• $V=\{1,2,\cdots,n\}$是节点集,表示软件系统的使用状态(如:软件的初始化、终止、屏幕上的输入/输出等)。

• 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。

• 如果(i,j)为一条边,转移概率$p(i,j)$表示从状态i一

到稿日期:2011-06-20 返修日期:2011-07-20 本文受国家自然科学基金(90818027, 91018005),国家高技术研究专题项目与发展计划(863)(2009AA01Z147),国家重点基础研究发展规划(973)(2009CB320703)资助。

张德平(1973-),男,博士,讲师,主要研究方向为软件测试技术、软件可靠性、数理统计等,E-mail:depingzhang@nuaa.edu.cn;徐宝文(1961-),男,博士,教授,博士生导师,主要研究方向为程序设计语言、软件工程、并行与网络软件等。

步转移到状态 j 的概率，否则 $p(i,j)=0$。转移概率 $p(i,j)$ 满足 $0\leqslant p(i,j)\leqslant 1$ 并且对任意的状态 i 均有：$\sum_{j=1}^{n}p(i,j)=1$。

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的概率，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的概率，其元素 $q(i,j)$ 满足 $\sum_{j=1}^{n}q(i,j)=1$。假定状态 1 为初态，状态 n 为终态，并且状态 n 为吸收态，即一旦进入状态 n 就不再离去，亦即 $p(n,n)=1$，$p(n,j)=0,\forall j\neq n$。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 Markov 链使用模型从初态 1 到终态 n 的一条遍历路径，记为 $\boldsymbol{x}=(x_1,x_2,\cdots,x_L)$，$L$ 为一次使用过程中执行操作的总数，即路径长度。x_i 是一次执行过程中的第 i 个操作。操作 x_i 执行后执行的下一操作 x_j 根据转移概率 $p(x_i,x_j)$ 随机选择。因此，软件一次使用过程中选择路径 $\boldsymbol{x}=(x_1,x_2,\cdots,x_L)$ 的执行概率为：

$$f(\boldsymbol{x},P)=P\{X=\boldsymbol{x}\}=\prod_{i=1}^{L}p(x_i,x_j)，\text{满足}\sum_{j}p(x_i,x_j)=1 \tag{1}$$

式中，$\boldsymbol{x}$ 是按操作剖面 P 随机生成的测试用例，x_j 与状态 x_i 相邻的可一步转移到的状态，即 $p(x_i,x_j)>0$，$X=(X_1,X_2,\cdots)$ 表示执行路径随机向量。基于 Markov 链使用模型的软件统计测试的每个测试例都对应了上述一条遍历路径 $\boldsymbol{x}$。记示性函数 $I_f(\boldsymbol{x})$ 为：

$$I_f(\boldsymbol{x})=\begin{cases}1, & \text{路径 }\boldsymbol{x}\text{ 执行过程中任一操作发生失效}\\ 0, & \text{否则}\end{cases}$$

显然，系统是否正确运行依赖于软件系统的规格说明，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 $I_f(X)$ 定义知，系统 π 的可靠性 $R(\pi)$ 为：

$$R(\pi)=1-E_p[I_f(X)]=1-\ell \tag{2}$$

式中，E_p 是相对于概率分布 $f(\boldsymbol{x},P)$ 求期望，ℓ 为系统的失效概率，即 $\ell=E_p[I_f(X)]$。

3 基于重要抽样的可靠性估计

由式(2)知软件可靠性无偏估计最直接的方法是：根据 Markov 链使用模型选出 N 条遍历路径 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N$，由矩估计方法可直接可得失效概率的无偏估计：$\hat{\ell}=\frac{1}{N}\sum_{i=1}^{N}I_f(\boldsymbol{x}_i)$。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，软件失效不容易观察到，这样估计出的可靠性会偏高，如果要得到较为精确的可靠性估计，要求测试的测试路径数 N 要足够大，这必然会导致测试开销过大。

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

$$\ell=E_p[I_f(X)]=\int I_f(\boldsymbol{x})W(\boldsymbol{x},P,Q)f(\boldsymbol{x},Q)\mathrm{d}\boldsymbol{x} \tag{3}$$

式中，$W(\boldsymbol{x},P,Q)=f(\boldsymbol{x},P)/f(\boldsymbol{x},Q)$ 为似然比率。所以失效概率 ℓ 的无偏估计 $\hat{\ell}$ 可表示为：

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

式中，$\boldsymbol{x}_i$ 为按测试剖面 Q 随机生成测试路径。

重要抽样方法[8]的关键就是如何确定一个最优的测试剖面 Q^*，使得按最优测试剖面生成测试用例运行后估计出的方差达到最小，即

$$Q^*=\arg\min_{Q}\{\frac{1}{N}(E_Q[I_f(X)W(X,P,Q)]^2-(E_Q^2[I_f(X)W(X,P,Q)]))\} \tag{5}$$

式中，arg 表示使方差期望达到最小时 Q 的取值，上式第二项为 ℓ 无偏估计的平方，其大小等于 ℓ^2，与 Q 无关，故有：

$$\begin{aligned}Q^*&=\arg\min_{Q}\{\int I_f^2(\boldsymbol{x})\,W^2(\boldsymbol{x},P,Q)f(\boldsymbol{x},Q)\mathrm{d}\boldsymbol{x}\}\\&=\arg\min_{Q}\{\int I_f(\boldsymbol{x})\,W(\boldsymbol{x},P,Q)f(\boldsymbol{x},P)\mathrm{d}\boldsymbol{x}\}\\&=\arg\min_{Q}\{E_P[I_f(X)W(X,P,Q)]\}\end{aligned} \tag{6}$$

式(6)求解一般比较复杂，很难求出解析解，用启发式算法求解时常常会因为在迭代生成渐近最优测试剖面的过程中需要大量的内存空间(来存储仿真信息)和大量的计算而影响算法的效率。

注意到式(6)中期望表达形式 $E_p[I_f(X)W(X,P,Q)]=\int I_f(\boldsymbol{x})\frac{f(\boldsymbol{x},P)}{f(\boldsymbol{x},Q)}f(\boldsymbol{x},P)\mathrm{d}\boldsymbol{x}$，可以看成两个概率分布 $h_1(\boldsymbol{x})$ 和 $h_2(\boldsymbol{x})$ 之间的 Ali-Silvey 距离[16]，即

$$D_{AS}=\int h_1(\boldsymbol{x})\frac{h_1(\boldsymbol{x})}{h_2(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x} \tag{7}$$

其中，$h_1(\boldsymbol{x})=\dfrac{I_f(\boldsymbol{x})f(\boldsymbol{x},P)}{\int I_f(\boldsymbol{x})f(\boldsymbol{x},P)\mathrm{d}\boldsymbol{x}}$，$h_2(\boldsymbol{x})=f(\boldsymbol{x},Q)$。

由此可得，方差最小等价于两个分布 $h_1(\boldsymbol{x})$ 和 $h_2(\boldsymbol{x})$ 之间的 Ali-Silvey 距离(或差异)最小。因此，确定最优测试剖面使得估计方差最小可转化为确定最优测试剖面 Q^* 使得概率分布 $h_1(\boldsymbol{x})$ 和 $h_2(\boldsymbol{x})$ 之间的距离最小。

为了便于计算，这里引入交叉熵 $D(h_1(\boldsymbol{x}),h_2(\boldsymbol{x}))$ 来度量 $h_1(\boldsymbol{x})$ 和 $h_1(\boldsymbol{x})$ 之间的距离，因此，最优测试剖面的确定问题就转化为确定 Q^* 使 $D(h_1(\boldsymbol{x}),h_2(\boldsymbol{x}))$ 达到最小，即

$$\begin{aligned}Q^*&=\arg\min_{Q}D_{AS}\approx\arg\min_{Q}\{D(h_1(\boldsymbol{x}),h_2(\boldsymbol{x}))\}\\&=\arg\max_{Q}\{E_P[I_f(X)\ln f(X,Q)]\}\end{aligned} \tag{8}$$

由式(8)可得最优测试剖面，由式(4)得到精确估计软件可靠性估计的重要抽样方法。

4 最优测试剖面生成

为确定最优测试剖面 Q^*，可以采用两种方法，一种是直接由式(8)求出解析解；另一种是当解析解求解不可行时，可通过一种递归算法生成一个测试剖面序列 $\{Q_0^*,Q_1^*,\cdots\}$，利用上一次迭代中得到的近似测试剖面生成测试路径来获得新的测试剖面，直到测试剖面序列收敛。

实际上，在迭代的第一步将 Q_0^* 初始化为操作剖面 P，即 $Q_0^*=P$，通过随机仿真，得式(8)的第一个近似最优解 Q_1^*：

$$Q_1^*\approx\arg\max_{Q}\frac{1}{N}\sum_{i=1}^{N}I_f(\boldsymbol{x}_i)\ln f(\boldsymbol{x}_i,Q) \tag{9}$$

在第 j 次迭代中渐近最优测试剖面 Q_j^* 的计算式：

$$\begin{aligned}Q_j^*&=\arg\max_{Q}\{\int E_{Q_{j-1}^*}[I_f(X)W(X,P,Q_{j-1}^*)\ln f(X,Q)]\}\\&\approx\arg\max_{Q}\frac{1}{N}\sum_{i=1}^{N}I_f(\boldsymbol{x}_i)W(\boldsymbol{x}_i,P,Q_{j-1}^*)\ln f(\boldsymbol{x}_i,Q)\end{aligned} \tag{10}$$

注意到 $f(\boldsymbol{x},Q)=\prod_i q(x_i,x_j)$，满足 $\sum_j q(x_i,x_j)=1$，则由拉格朗日乘子法可得式(10)的解可转化为如下最大值问题求解：

$$\max_Q\{E_{Q_{j-1}}[I_f(X)W(X,P,Q_{j-1})\sum_i \ln q(x_i,x_j)]+\sum_i u_i(\sum_j q(x_i,x_j)-1)\} \tag{11}$$

式(11)关于 $q(x_l,x_m)$ 求偏导，并令其等于零，由此可得 Q_j^* 的元素 $q(x_l,x_m)$ 为：

$$q(x_l,x_m)=\frac{E_{Q_{j-1}}[I_f(X)W(X,P,Q_{j-1})\sum_i I_{\{X_i=x_l,X_{i+1}=x_m\}}]}{E_{Q_{j-1}}[I_f(X)W(X,P,Q_{j-1})\sum_i I_{\{X_i=x_l\}}]} \tag{12}$$

因此可得 $q(x_l,x_m)$ 的估计式：

$$\hat{q}(x_l,x_m)=\frac{\sum_{k=1}^{N} I_f(\boldsymbol{x}_k)W(\boldsymbol{x}_k,P,Q_{j-1})\sum_i I_{\{X_i=x_l,X_{i+1}=x_m\}}}{\sum_{k=1}^{N} I_f(\boldsymbol{x}_k)W(\boldsymbol{x}_k,P,Q_{j-1})\sum_i I_{\{X_i=x_l\}}} \tag{13}$$

其中，测试路径 $\boldsymbol{x}_k(k=1,2,\cdots,N)$ 按测试剖面 Q_{j-1} 随机生成，示性函数 $I_{\{X_i=x_l\}}$ 取 1 表示测试路径 $\boldsymbol{x}_k$ 执行了操作 x_l，否则取 0；而 $I_{\{X_i=x_l,X_{i+1}=x_m\}}$ 取 1 则表示测试路径 x_k 在执行完成操作 x_l 后接着执行操作 x_m，否则取 0。

由式(12)得到的可靠性估计具有与文献[12]相同的性质。

性质 1 如果 $f(X;P)>0$，测试剖面 Q 的元素满足式(12)，则由式(4)及式(1)估计出的可靠性估计是无偏估计。

性质 2 如果 $f(X;P)>0$，测试剖面 Q 的元素满足式(12)，则由式(4)及式(1)估计出的可靠性是方差为零的无偏估计。

根据测试剖面的生成过程，可以得到一种求解最优或渐近最优测试剖面的迭代算法，然后利用求解出的最优测试剖面仿真，估计出安全关键软件系统的可靠性：初始化测试剖面 Q_0 为操作剖面 P 随机生成测试路径，利用式(13)修正测试剖面 Q_1，然后再利用修正后的测试剖面 Q_1 进行抽样，如此循环，直到测试剖面不再变化或变化不大时停止。具体算法由算法 1 给出。

算法 1 估计软件可靠性的迭代算法

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 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N$，观察其是否发生软件失效。
3. 由生成的 N 条测试路径 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N$，由式(13)求解出 $\tilde{Q}_j$，应用平滑技术修正得到

   $\hat{Q}_j=a\tilde{Q}_j+(1-a)\hat{Q}_{j-1}$，$a\in(0.4,0.9)$
4. 令 $j=j+1$，重复步骤 2－3，直到对于某个给定的 d(如 $d=4$)有 $\hat{Q}_j=\hat{Q}_{j-1}=\cdots=\hat{Q}_{j-d}$ 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 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_M$，应用下式估计出软件可靠性：

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

在步骤 3 中利用式(13)修正测试剖面时，若分母为零，则从当前状态转移到相邻其它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的有效性，这里采用文献[12]给出的软件实例加以验证。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，关键操作为边(4,6)，(8,9) 和(8,10)。假定各个操作失效的先验概率用向量 f 表示：$f=(0,0.2,0.001,0.001,0.001,0.2,0.001,0.001,0.001,0.001,0.001,0.001)$。状态之间的转移概率(即操作剖面)如图 1 所示。

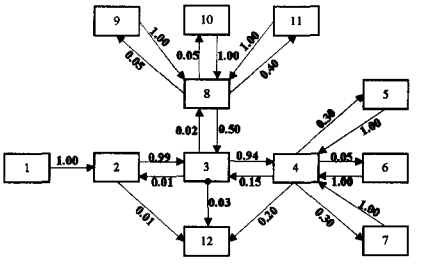

图 1 Gutjahr 给出的马尔可夫链使用模型

在算法 1 中设定参数分别为：$\alpha=0.4$，生成的测试路径数为 $N=30000$，当相邻两次测试剖面最大变化量小于 $\varepsilon=0.0001$ 时停止迭代，则由算法 1 得到最优(或近优)测试剖面 Q^*。分别采用标准方法(Standard Method)、模拟退火算法(Simulated Annealing)和交叉熵方法(Cross Entropy Method)，抽样 $N=5000$ 条测试路径仿真模拟测试 200 次，各得到 200 个可靠性估计，其可靠性估计值散点图和方差散点图如图 2 和图 3 所示。

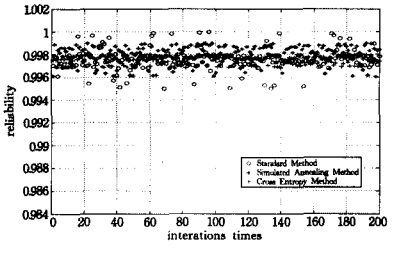

图 2 3 种不同剖面下的可靠性估计

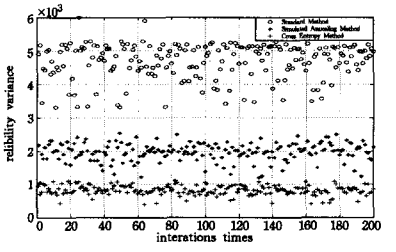

图 3 3 种不同测试剖面下的可靠性估计的方差

由 200 次仿真估计出系统的 200 个可靠性数据，其盒状图如图 4 所示。

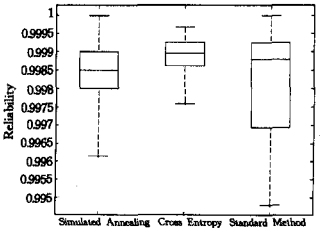

图 4 3 种不同剖面下可靠性估计的盒形图

200 次估计中，采用原始操作剖面 P 时可靠性估计的平均值为 0.9977，估计方差的平均值为 4.7213×10^{-3}；采用由模拟退火算法得出的测试剖面 T 时的可靠性估计的平均值为 0.9981，估计方差的平均值为 1.8412×10^{-3}；采用由交叉熵方法得出的最优测试剖面 Q^* 时的可靠性估计的平均值为 0.9975，估计方差的平均值为 7.9444×10^{-4}。采用最优测试剖面可以显著降低估计的方差，且交叉熵方法要略优于模拟退火算法。由盒状图可知，采用最优测试剖面时数据最集中，而采用标准方法时，数据波动最大，模拟退火算法次之。

在 200 次仿真模拟过程中，遍历路径数为 1000 条时，由操作剖面、模拟退火算法得出的测试剖面、最优测试剖面分别生成的测试路径中关键操作累计遍历次数被记录，各关键操作的平均遍历次数如表 1 所列。

表 1　200 次仿真模拟中各关键操作平均遍历次数

	操作 6	操作 9	操作 10
标准方法	86.5	28.66	105.3
模拟退火方法	158.3	120.3	230.6
交叉熵方法	235	95	350

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 9 的遍历机会要低于模拟退火算法。

结束语　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的测试机会，加速软件测试，在提高软件系统质量的同时降低软件总费用。该方法是为实现测试目标而采用的一种导向性测试数据集模拟生成方法，它利用“功效”值最好的测试数据集包含的失效信息，把修正测试剖面归结为一个迭代优化问题迭代求解，估计安全关键软件可靠性，加速软件测试。

参 考 文 献

[1] 赵亮，王建民，孙家广. 统计测试的软件可靠性保障能力研究[J]. 软件学报，2008，19(6)：1379-1385

[2] 杨善林，丁帅，褚伟. 一种基于效用和证据理论的可信软件评估方法[J]. 计算机研究与发展，2009，46(7)：1152-1159

[3] 覃志东，雷航，桑楠，等. 安全关键软件可靠性验证测试方法研究[J]. 航空学报，2005，26(3)：334-339

[4] Gutjahr W J. Failure risk estimation via Markov software usage models[C]//Schoitsch E. ed. SAFECOMP 96，Proc. of the 15th International conference on computer safety，reliability and security. Springer，1997：183-192

[5] Gutjahr W J. Importance sampling of test cases in Markovian software usage models[J]. Probability in the Engineering and Informational Sciences，1997，11：19-36

[6] Gutjahr W J. Software dependability evaluation based on Markov usage models[J]. Performance Evaluation，2000，40(4)：199-222

[7] Doerner K，Laure E. High performance computing in the optimization of software test plans[J]. Optimization and Engineering，2002，3：67-87

[8] Doerner K，Gutjahr W J. Extracting test sequences from a Markov software usage model by ACO[C]//LNCS. Springer Verlag，2003，2724：2465-2476

[9] 颜炯，王戟，陈火旺. 基于重要抽样的软件统计测试加速[J]. 计算机工程与科学，2005，27(3)：64-66

[10] Yan J，Zhou K P，Deng C H，et al. Importance Sampling Based Safety-Critical Software Statistical Testing Acceleration[C]// International Conference on Computational Intelligence and Software Engineering (CiSE). 2010：1-4

[11] 徐云青，徐义峰，李舟军. 基于使用模型的软件可靠性加速测试[J]. 计算机应用与软件，2009，26(3)：147-148

[12] 张德平，聂长海，徐宝文. 软件可靠性评估的重要抽样方法研究[J]. 软件学报，2009，20(10)：2859-2866

[13] 张德平，聂长海，徐宝文. 测试资源受约束的安全关键软件加速测试方法[J]. 计算机科学，2009，36(5)：138-141

[14] 张德平，查日军. 基于 Markov 链使用模型的加速统计测试方法[J]. 东南大学学报，已录用

[15] Chari K，Hevner A. System test planning of software：an optimization approach[J]. IEEE transactions on software engineering，2006，32(7)：503-509

[16] Orsak G C，Aazhang B. Constrained solutions in importance sampling via robust statistics[J]. IEEE Trans. Inform. Theory，1991，37：307-316

(上接第 109 页)

额外的断连检测和处理所引起的。

参 考 文 献

[1] Wolfson O，Chamberlain S，Dao S，et al. Cost and Imprecision in Modeling the Position of Moving Objects[C]//Proceedings of the Fourteenth International Conference on Data Engineering (ICDE). 1998

[2] Lam G H K，Leong HV，Chan SC. GBL：Group-based location updating in mobile environment[C]//Proceedings of the 9th International Conference on Database Systems for Advanced Applications. Jeju Island，Korea，2004：762-774

[3] Wolfson O，Sistla A P，Chamberlain S，et al. Updating and Querying Databases that Track Mobile Units[J]. Distributed and Parallel Databases，1999，7：257-287

[4] Wolfson O，Xu B，Chamberlain S，et al. Moving Objects Databases：Issues and Solutions[C]//Proceedings of the Tenth International Conference on Scientific Database Management (SSDBM). 1998

[5] Civilis A，Jensen C S，Nenortaite J，et al. Effient Tracking of Moving objects with Precision Guarantees[C]//Proceedings of the First Ann. International Conference on Mobile and Ubiquitous Systems：Networking and Services (Mobiquitous). 2004

[6] Lam K Y，Ulusoy O，Lee T S H，et al. An efficient method for geneating location updates for proceeding of location-dependent continuous queries[C]//Proceedings of the 6th International Conference on Database Systems fro Advanced Applications. Hong Kong，China，2001：218-225

[7] Wolfson O，Yin H. Accuracy and resource consumption in tracking and location prediction[C]//Proceedings of the 7th International Symposium on Spatial and Temporal Databases. Santorini Island，Greede，2003：325-343

[8] 胡志智，孟小峰，郭妍妍，等. 基于模拟预测的移动对象位置主动更新策略[J]. 计算机研究与发展，2004，41(增刊)：43-49

[9] Chen J D，Meng X F，Li B，et al. Tracking network-constrained moving objects with group updates[C]//Proceedings of the 7th International Conference on Web-Age Information Management (WAIM 2006). Hong Kong，China，2006：158-169