基于交叉熵方法构件软件可靠性优化①

Cross-Entropy Method for Component-Based Software Reliability

孙 莉 (盐城工学院 信息工程学院 江苏 盐城 224051)

摘 要: 研究了构件软件可靠性的多目标优化问题中，如何实现最大化构件软件可靠性估计值的同时最小化构件软件可靠性估计方差。将可靠性优化过程转换成组合优化问题，设计出其重要抽样模型，运用交叉熵方法寻找最优重要抽样分布函数，解决构件软件多目标优化问题。最后给出交叉熵方法算法实现方案。

关键词: 组合优化 稀有事件 重要抽样 交叉熵方法

构件软件的可靠性估计值是获得整个软件系统可靠性的基础。许多研究文献中，研究重点都放在最大化构件软件可靠性，却很少考虑减少构件软件可靠性估计方差的问题。其实，降低构件软件可靠性估计方差本身就是一个非常重要的研究内容，否则会出现高可靠性和高估计方差的恶劣结果；而且，构件软件本身的特性又会使得估计方差在软件系统级传播和扩大，这样即使软件系统可靠性再高，但估计方差也会相当大。

因此，必须在最大化构件软件可靠性估计值的同时最小化构件软件可靠性估计方差。该问题是一个典型的多标准最优化问题。信息论中的交叉熵方法可以实现解决此类复杂的多目标优化问题。当某些优化问题没有办法直接获得解答和实现时，通常是寻找一种新的概率分布来拟合原事件概率分布。这样，难以解决的原来的优化问题(原事件概率分布)就转变成了比较容易解决的新的优化问题(新的概率分布)。为了保证新的概率分布充分的接近原事件概率分布，可以利用信息论中的交叉熵作为衡量标准，度量两种概率分布之间的信息量差异。故而，最小化构件软件可靠性估计方差的问题可以采用交叉熵方法解决。

1 组合优化

1.1 交叉熵

1997 年 Rubinstein 首次在估计复杂随机网络中稀有事件发生概率的解决方法时提出了一种通过使估计方差最小的适应性算法。2001 年 Rubinstein 又将上述算法简化，提出了交叉熵方法，该方法不仅可以估计稀有事件的概率也可以求解那些很难解答的组合优化问题。

交叉熵方法的主要思想是将“确定性”优化问题转变成一个相关的“随机”优化问题，即伴随随机优化问题(ASP)，然后用稀有事件模拟的随机模拟技术，即重要抽样方法(Importance Sampling，IS)来求解此“随机”优化问题[2]。

人们在解决某些优化问题时，常常会用一种概率分布来拟合原来的概率分布。假设通过一组试验估计得到的概率分布为 q，样本空间Ω，随机变量 X；而真实的概率分布为 p，具有相同的Ω和 X。则人们一般都会考虑的问题是：模拟的概率分布 q 和真实的概率分布 p 相比，相似程度如何？误差多大？或者如果相对于某个真实概率分布，现在存在着多个模拟概率分布，应该如何判断和比较哪个概率分布更好？通常，采用交叉熵表示两者之间的相似程。

交叉熵(Cross-Entropy)[3]：随机变量 X，两个分布 p 和 q，其中 p 为 X 的真实概率分布，q 为 X 的模拟概率分布。则定义下式为 q 对 p 的交叉熵：

$$H(p,q)= -\sum_{x\in\Omega} p(x)\log q(x) \qquad (1)$$

交叉熵主要是用来度量模拟的概率分布相对于原

① 收稿时间:2009-03-18

来的真实概率分布的逼真性和可靠性，是评价概率模型性能的较好工具[4]。交叉熵越小，表示 q 和 p 越相似。如果有两个被评估分布 q_1 和 q_2，若 $H(p,q1)<H(p,q_2)$，则表明 q_1 比 q_2 好，q_1 比 q_2 更接近真实分布 p。

1.2 组合优化

从 Rubinstein 首先把交叉熵方法运用到稀有事件模拟中，人们就不断尝试是否可以用交叉熵方法解决组合优化的问题。2004 年到 2005 年，交叉熵方法在研究组合优化问题中的 NP-hard 问题上很活跃，并且取得了令人瞩目的研究成果[5]。所谓组合优化是指在离散的有限的数学结构上，寻找一个满足给定约束条件并使其目标函数值达到最大或最小的解。

考虑下面的组合优化问题[3]：

$$z^{\bullet}=\max_{x\in\chi} S(x) \tag{2}$$

为了能够快速的解决该问题，可以发现上面等式的实质就是求解 $S(x)\geq z$ 的概率估计问题：

$$\ell=P(S(x)\geq z)$$

为此，定义一个在 χ 上概率分布簇 $f(x;\ u)$ 和一个 $z\in R$ 的示性函数集合 $\{I_{\{S(x)\geq z\}}, x\in\chi. z\in R\}$。$I_{\{S(X)\geq z\}}$ 具体表示如下：

$$I_{\{S(X)\geq z\}}=\begin{Bmatrix}1 \text{ if } s\geq z\\ 0 \text{ if } s<z\end{Bmatrix} \quad s\geq 0$$

这样就可以将该优化问题转化成一个伴随随机问题(等式 3)，并且可以使用交叉熵方法很容易的获得解答。

$$\begin{aligned}\ell(z)&=P_u(S(x)\geq z)\\&=\sum I_{\{S(X)\geq z\}} f(x;\ u)\\&=E_u I_{\{S(X)\geq z\}}\end{aligned} \tag{3}$$

P_u 和 E_u 分别是概率分布簇 $f(x;\ u)$ 的概率测度和期望值。

当估计 $z=z^*$ 时 $\ell=P(S(x)\geq z)$ 的值，如果可行解区域 χ 足够大，一般可以采用模拟的方法在所有的 $x\in\chi$ 中获得 x^*，使得 $z^*=S(x^*)\geq S(x)$。最常用的方法就是蒙特卡罗模拟，即从概率分布簇 $f(x;\ u)$ 中产生随机样本 $X_1,...,X_N$，用 $\frac{1}{N}\sum_{i=1}^{N} I_{\{S(X_i)\geq z\}}$ 作为 ℓ 的无偏估计。该方法的缺点是 N 要充分的大，即需要大量的模拟资源。

1.3 重要抽样

其实估计 $\ell=P(S(x)\geq z)$ 也可以通过重要抽样方法实现，即从另一个概率分布簇 $f(x;\ p)$ 中也产生随机样本 $X_1,...,X_N$。这样问题就表示成如下形式：

$$\hat{\ell}=\frac{1}{N}\sum_{i=1}^{N} I_{\{S(X_i)\geq z\}}\frac{f(x,u)}{f(x,p)} \tag{4}$$

$f(x,p)$ 是 $f(x,u)$ 的等价概率分布簇，选择的标准是 $f(x,p)$ 和 $f(x,u)$ 之间的交叉熵最小。

假设 $f(x,p)$ 是均匀分布，因为每个 Xi 只会取 0 或 1，所以下式显然成立：

$$f(X,p)=\prod_{i=1}^{N} p_i^{X_i}(1-p_i)^{1-X_i} \tag{5}$$

结合等式 4 和等式 5，可以得到如下等式：

$$p_i=\frac{\sum_{i=1}^{N} I_{\{S(X_i)\geq z\}} x_i}{\sum_{i=1}^{N} I_{\{S(X_i)\geq z\}}} \tag{6}$$

从上述分析可以得知：由于原始的最大化优化问题无法直接求得结果，故采用重要抽样方法变换，选择一个与初始的概率分布簇等价的概率分布簇代替，条件是满足两个概率簇的交叉熵最小。这种解决方案就是交叉熵方法。

不管应用交叉熵方法解决任何问题，它都可以描述成一个迭代算法，每次迭代一般有两个主要阶段：

(1) 根据指定某个指定的带动态参数集的随机机制产生一个随机样本

(2) 基于样本修正随机机制的参数，最典型的参数是概率分布，目的是为了在下一次迭代中产生“更好”的样本。

2 构件软件可靠性估计方差最小化

由于交叉熵方法为估计稀有事件的概率提供了一种快速确定最优参数的方法，故可以采用交叉

熵方法最小化构件软件可靠性估计方差。这样可以准确而又快速的获得解答，简化模型并提高算法效率。

2.1 模型建立

假设软件系统由 s 个构件软件组成(每个构件软件完成的功能互不相同)，约束条件为测试总代价的限制 C 以及构件软件可靠性估计值是无偏估计。故构件软件可靠性优化问题可用如下带约束的优化问题表示：

$$\begin{aligned} &\max \quad Rs=S(X); \\ &\min \quad Var(Rs) \end{aligned} \tag{7}$$

关于组合优化问题 $\max R_s = S(X)$ 在《A Cross-Entropy Based Algorithm for Combinatorial Optimization Problems》中已经给出了完整的交叉熵解决方案，本文主要探讨组合优化问题 $\min \mathrm{Var}(R_s)$ 的交叉熵方法的解决方案。

2.2 交叉熵方法解决方案

首先，确定最小化估计方差问题中需要解决的优化问题。设 γ^* 为估计方差的最小值，最小化问题可以表示成：

$$\begin{aligned} &Var(x)=T(x) \\ &\gamma^* = \mathrm{Var}(x^*) \\ &\quad = \min_{x\in\chi} T(x) \end{aligned} \tag{8}$$

定义一个在 χ 上的概率分布簇 $f(x;\ u)$ 和一个示性函数集合 $\{I_{\{T(x)\le\gamma\}}, x\in\chi.\gamma\in R\}$。$I_{\{T(X)\le\gamma\}}$ 为 χ 上不同 γ 值得示性函数集合，表示如下：

$$I_{\{T(X)\le\gamma\}} = \begin{Bmatrix} 1 \text{ if } t\le\gamma \\ 0 \text{ if } t>\gamma \end{Bmatrix} \quad t\ge 0$$

这样，等式 8 表示的优化问题就可以转换成一个伴随随机问题

$$\begin{aligned} P_u(T(x)\le\gamma) &= \sum I_{\{T(X)\le\gamma\}} f(x;\ u) \\ &= E_u I_{\{T(X)\le\gamma\}} \end{aligned} \tag{9}$$

其中，P_u 和 E_u 分别为相对于 $f(x;\ u)$ 的概率和期望值。

此类伴随随机问题的实质就是希望估计当 $\gamma=\gamma^*$ 时 $P_u(T(x)\le\gamma)$ 的值，需要使用重要抽样方法解决。即从另一个等价的概率分布簇 $f(x;\ v)$ 中也产生随机样本 $X_1,...,X_N$，并且 $f(x;\ u)$ 与 $f(x;\ v)$ 的交叉熵最小。在解决问题的实际过程中，一般可以通过抽样产生一个推断参数序列 $\{v_t, t\ge 0\}$ 和不同水平 $\{\gamma_t, t\ge 1\}$ 的序列。在每一次抽样中(N 个样本)，参数 v_t 的修正可以参照等式 6，γ_t 的选择可选择由 N 个样本计算出的 N 个总成本值的 ρ 分位数，这样可以保证事件 $\{T(X_i)\le\gamma\}$ 发生的概率至少为 ρ。

因此，可以初步给出用交叉熵方法解决最小化可靠性估计方差的解决思路，具体步骤如下：

第一步：初始化，设置 $\hat{v}_0 = u$，$X\sim p(x;u)$，ρ = 预定值，迭代计数器 t=0；

第二步：抽样。随机产生样本点，根据 $X\sim p(x;\hat{v}_t)$ 产生随机样本 $(X_1, X_2,..., X_N)$；然后计算出 $T(x_i)$ 并进行降序排序，即 $T(x_1)\ge T(x_2)\ge...\ge T(x_N)$；最后，令 $\hat{\gamma}_t = T(X_{\lceil\rho N\rceil})$，用样本功效值的 $\rho\cdot 100\%$ 来估计 γ_t；

第三步：修正参数。使用相同的随机样本 $(X_1, X_2,...X_N)$ 修正计算 $\hat{v}_t$，由等式 6 解出 $\hat{v}_t$

$$\hat{v}_t = \{p(X_1),...,p(X_N)\}$$

第四步：如果 $\hat{\gamma}_t = \gamma_t$，则迭代结束；否则 t=t+1，转向第二步

通过上述的迭代算法，可以获得最终解的集合 $M=\{x_i \mid x_i = 1\}$。通过这个解集合，我们可以确定软件系统最优化的构件软件配置(选择那些 $x_i = 1$ 的构件软件)，同时也可以获得可靠性估计方差的最小值 $\hat{\gamma}_t$。

2.3 交叉熵算法实现

根据上节对交叉熵方法解决方案的具体分析，可以给出对应的交叉熵算法。构件软件可靠性估计方差最小化的交叉熵算法如下：

算法 1. 软件可靠性估计方差最小化的交叉熵算法

① 定义初始概率分布 $p(x;u)$，ρ =预定值

② 当迭代终止条件未满足时，完成下列操作：

③ 根据 $p(x;u)$ 产生随机样本 $(X_1, X_2,..., X_N)$

/* 抽样，随机产生样本点 */

④ sort($T(x_i)$, H)

/* 对 $T(x_i)$ 降序排序，结果存入 H(X) 中 $H_{(1)}\ge H_{(2)}\ge\cdots\ge H_{(N)}$ */

⑤ q=$\lceil\rho\cdot N\rceil$；$\gamma_t = H_{\lceil\rho\cdot N\rceil}$

⑥ Q={$x_k : 1\le k\le q$}

⑦ $p_i = \dfrac{\sum_{x\in Q} x_i}{|Q|}$

/* 计算 p_i，修正参数 */

⑧ 跳回 2 进行判断循环是否继续

交叉熵算法的实现并不困难，每次迭代主要还是由两个关键步骤组成：

① 适应修正 γ_t。对于固定的 v_{t-1}，γ_t 为在参数 v_{t-1} 下的功效函数序列 T(X)的 $\rho\cdot100\%$。即 $\gamma_t = T_{\lceil \rho\cdot N \rceil}$。因此，每次迭代中获得的 γ_t 均不一样。

② 适应修正 v_t。对于固定的 v_{t-1} 和 γ_t，根据等式 6 可以很容易的求得 v_t。通过不断的修改 v_t 的值，可以使参数越来越满足规定要求。

3 实验数据分析

根据文献《NAT- PT 可扩展性和可靠性问题研究》[6]，得到在一定显著水平和安全性指标下所需要最少的测试样本的大小。将传统统计测试方法和基于交叉熵的重要抽样测试方法的所需要的最少抽样样本大小进行比较，如表 1 所示。

表 1　两种测试方法所需抽样样本量比较

传统统计测试方法样本量	重要抽样测试方法样本量
1.2×10^{12}	1.76×10^{9}
1.4×10^{12}	2.2×10^{9}
1.6×10^{12}	2.48×10^{9}
1.9×10^{12}	2.93×10^{9}
2.3×10^{12}	3.55×10^{9}

表 1 表明，基于交叉熵的重要抽样方法能大幅度地减少测试样本大小，从而有效地提高测试效率。

4 结论

近几年来，人们研究和讨论了一些多目标优化方法，如目标加权、目标规划和 Pareto 优化。到目前为止，解决多目标优化问题做的较好的是利用遗传算法和 Pareto 优化排序相结合的技术。这种技术的优点是可以在 Pareto 集的全部区域上探测，解的可行区域选择余地大。其不足之处在于所求的 Pareto 最优集分布不均匀，包含的真正有用的 Pareto 最优解少，算法效率低[7]。

交叉熵方法是一种结合信息学中熵的概念，用于衡量两种概率分布的相似程度以及判断哪个概率分布更接近于真实分布的比较简单的数学方法。交叉熵算法的收敛性肯定优于遗传算法和 Pareto 优化排序相结合所对应的算法，基本可以直接求出所有解。但交叉熵算法也存在着一定的缺点，算法的计算量比较大且计算也比较复杂。这还需要在以后的研究工作中改进。

参考文献

1 Jin TD. Complex system reliability analysis and optimization considering component reliability estimation uncertainty [Ph.D Thesis]. UMI-3043040, University of New Brunswick, New Jersey, 2001.

2 刘勋,马义德,钱志柏.一种基于交叉熵的改进型 PCNN 图像自动分割新方法.中国图象图形学报, 2005,10(5):579－584.

3 De Boer PT, Kroese DP. A Tutorial on the cross-entropy method. Annals of Operations Research, 2005,134:19－67.

4 周勇,刘三阳,杨曙光.基于交叉熵的通讯网的优化算法.系统工程与电子技术, 2004,26(10):1471－1533.

5 李卫华,周军,周连文,程英蕾.基于交叉熵及曲线进化的图像分割算法.计算机工程与应用, 2005,26(3):51－53.

6 叶润国,冯彦君,吴宇,等. NAT- PT 可扩展性和可靠性问题研究.微电子学与计算机, 2004,21(4):19－24.

7 马超,吕震宙,袁修开.基于重要抽样马尔可夫链模拟的可靠性灵敏度分析新方法.机械强度, 2008, 30(1):41－46.