

改进的局部 ART 测试算法研究*)

孙林 周兴社 董云卫 张凡 陈俊文

(西北工业大学计算机学院 西安 710072)

摘要 局部 ART 算法虽然克服了测试样本点与样本生成区域已测试顶点有可能距离很近的问题,但它对生成区域周围的已测试样本点对样本选择的影响考虑不足。本文针对局部 ART 算法的缺陷,提出了改进的局部 ART 算法,将生成区域周围一定范围内的已测试顶点都加入到临近已测试样本集中,在保持时间复杂度的同时,使测试样本的分布更加均匀。实验结果表明,改进的局部 ART 算法的错误检测能力得到了提升。

关键词 改进的局部 ART 算法,局部 ART 算法,D-ART,RRT

Research on Advanced Adaptive Random Testing by Localization

SUN Lin ZHOU Xing-she DONG Yun-wei ZHANG Fan CHEN Jun-wen

(School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract Although Adaptive Random Testing has solved the problem that the generated test case may be too close to the nearby executed vertexes of the test generating zone, it does not consider the influences of the test cases near the test generating zone. This paper proposes the Advanced Adaptive Random Testing by Localization algorithm, which adds all the executed test cases near the zone to the nearby executed test set and thus spreads the test cases more evenly while maintaining the complexity. The experiment results show that the failure detecting ability of the Advanced ART by Localization is promoted.

Keywords Advanced ART by localization, ART by localization, D-ART, RRT

1 引言

错误模式指导致错误产生的程序输入的分布模式。文献[1]中指出,不同待测程序的错误模式不尽相同。二维输入空间的错误模式主要可以分为三种类型:块形模式、条形模式和点模式,如图 1 所示,阴影区域表示引起错误的输入。对于块形模式和条形模式,如果随机测试样本点能均匀地分布于输入空间内,那么检测到第一个错误所需要的测试样本数 $F\text{-measure}$ 将变少。

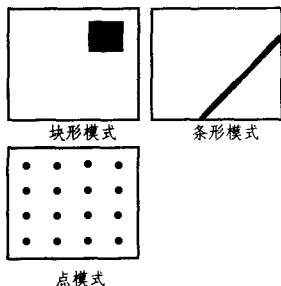


图 1 三种错误模式

随机测试方法通过在输入域内随机选取测试样本点进行测试,简单、易于实现,但样本有可能分布不均匀。为了达到测试样本均匀分布的目的,文献[2]提出了适应性随机测试方法 ART,它在随机测试过程中利用了已测试样本点的位置信息,选取那些离已测试样本点尽量远的测试样本点为测试样

本,从而达到测试样本点的均匀分布。基于这种思想,文献[3]提出了动态划分 ART 算法,它维持一个输入子域的集合 C ,初始化为整个输入域。算法从集合 C 中选取面积最大的一个输入子域,作为生成测试样本点的目标区域,并将其从 C 中删除掉,从这个子域中随机生成一个测试样本点进行测试。如果测试样本点检测出程序错误,那么测试终止;否则,样本点将候选区域划分为四个子域,将这四个子域加入到集合 C 中。重复上面的步骤,直至检测到程序出错。

动态划分 ART 算法生成的测试样本点在糟糕的情况下有可能离已测试样本很近。如图 2 所示,样本点 T_1, T_2, T_3 是根据上述算法生成出来的,但是它们的距离很近,并没有达到预想的均匀分布的目的。

为了解决这个问题,文献[4,5]分别提出了局部 ART 算法和随机划分 RRT 算法。本文在分析了这两个算法的不足的基础上,提出了改进的局部 ART 算法,在距离计算增加不多的情况下,使测试样本的分布更加均匀。

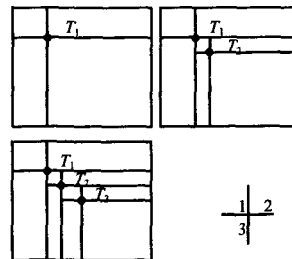


图 2 动态划分 ART 算法的糟糕情况

*)国家 863 项目(2006AA012162)。孙林 硕士研究生,主要研究方向为嵌入式系统与嵌入式系统测试;周兴社 教授,博士生导师;张凡 博士研究生,主要研究方向为嵌入式计算;陈俊文 硕士研究生。

2 相关算法原理

局部 ART 算法分为三步。首先,按照和动态划分 ART 算法相同的步骤,从输入子域的集合 C 中选取面积最大区域为目标区域,其四个顶点中已测试过的点组成一个临近已测试样本集 E 。第二步,如果在目标区域中使用 D-ART 算法,那么在该区域中随机生成一个测试样本候选集, $T = \{T_1, T_2, \dots, T_k\}$, 计算 T 中每一个顶点与集合 E 中元素之间距离的最小值,然后最小值最大的那个点即为选中的测试样本点 T_q ; 如果在目标区域使用 RRT 算法,那么在 E 中每个顶点附近设置一个排除域,在目标区域内排除域外生成测试样本点 T_q 。第三步,如果 T_q 是一个导致待测程序错误的测试样本,则算法终止,否则 T_q 将目标区域分成了四个子区域,并将其加入集合 C 中。

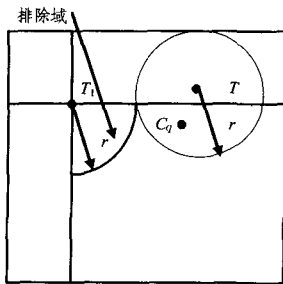


图3 局部 ART 算法的糟糕情况

局部 ART 算法有效地解决了生成测试点与已测试顶点可能距离很近的问题,但对于边界附近的已测试样本点,它并不能消除其对测试样本点生成的影响。如图3所示,按照局部 ART 的算法生成测试样本点 T_q ,虽然有效地保持了与已测试样本点 T_1 之间的距离,但无法保证与已测试样本点 T 之间的距离。

随机划分 RRT 算法与局部 ART 算法的不同点是它在目标区域的内部又划定了一个小区域,并将这个小区域作为生成测试样本的目标区域。如图4所示,在这个区域内生成测试样本 C_q 。虽然这种算法有效地避免了 C_q 与已测试样本点之间的距离,但是对测试空间的限制太大,排除掉了阴影区域所示的目标区域范围,而这个范围内的测试样本点并非都是不可取的。

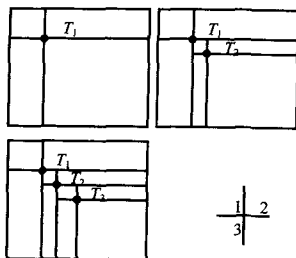


图4 随机划分的 RRT 算法

3 改进的局部 ART 算法

本文为解决图3所示的问题,提出了改进的局部 ART 算法。本算法通过扩大临近已测试样本集 E 的选择范围,将图3中如 T 一样的对测试样本有影响的已测试样本也加入 E 中,以消除临近的非顶点已测试样本点对测试样本选择的影响。

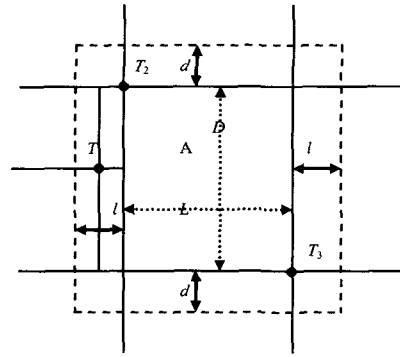


图5 临近已测试样本点的范围

对于选定目标区域 A ,在其周围圈定虚线所示的区域,其大小如图5所示。 d 和 l 可有多种选择方式,对于应用 D-ART 的改进局部 ART 算法,本文选定 $\frac{d}{D} = \frac{l}{L} = f$ 。对于应用 RRT 的改进局部 ART 算法,本文取 $d=l=r$,其中 r 为排除域的半径。距离目标区域 r 以外的已测试样本点,其排除域与目标区域没有重合,对测试样本的选择没有影响。将虚线区域内的已测试样本点加入集合 E 中,从而考虑到了所有的临近已测试样本点对测试样本选择的影响。具体算法如下所示。

改进的局部 ART 算法描述

(1)初始化子域链表 L 为整个输入域 $\{(i_{\min}, j_{\min}), (i_{\min}, j_{\max}), (i_{\max}, j_{\max}), (i_{\max}, j_{\min})\}$ 。已测试样本集 TE 为空。

(2)去除 L 中面积最大的元素,并将其作为目标区域。设置临近已测试样本集 E 为空。

(3)如果使用 D-ART 算法来生成测试用例,那么按照图5所示,以 $l=f * L, d=f * D$ 为参数来生成临近测试样本点范围,其中 f 为控制范围大小的比例参数。将 TE 中所有在此虚线范围内的元素加入集合 E 。在目标区域内随机生成一个测试样本候选集 $T = \{T_1, T_2, \dots, T_N\}$ 。计算样本点和临近已测试样本点之间的距离,并标记为 $dist(T_j, E_i)$ 。按照如下的标准选择测试样本点 T_q 。

$$\forall j \in \{1, 2, \dots, k\} (\min_{i=1}^n dist(C_q, E_i) \geq \min_{i=1}^n dist(C_j, E_i))$$

如果是用 RRT 算法来生成测试样本,那么 $l=d=r$, r 为排除域半径。将 TE 中所有在此虚线范围内的元素加入集合 E 。在 E 中每一个元素周围设置一个以 r 为半径的圆形排除域。在目标区域内持续生成测试样本点,直至样本点在排除域外,此样本点即为测试样本点 T_q 。

(4)如果 T_q 是一个导致错误输出的样本,那么检测到错误,算法终止;否则,将目标区域以 T_q 为中心分成四个输入子域,将这些子域加入链表 L 中。将 T_q 加入集合 TE 中。转到步骤 2。

在使用 RRT 改进局部 ART 算法时,排除域半径 r 的大小需要小心地设计。为方便起见,我们使用排除比例 f 来表示 r 与目标区域的最短边边长的比值。因为邻近已测试样本集中的元素有可能分布于目标区域周围的任何地方,所以 f 的上限为 50%。否则,目标区域有可能被排除域全部覆盖,从而导致不能生成测试样本。

本算法通过扩大临近已测试样本点的范围,使生成的测试样本点与临近测试样本点都保持了一定的距离,有效地考虑了目标区域周边的已测试样本点对测试样本生成的影响。

4 改进的局部 ART 算法性能仿真

我们通过在二维的方形输入域内进行模拟试验来评估本文所提算法在块形模式下的错误检测能力。为此我们选择 Mersenne Twister^[6]作为随机数生成器。每次仿真过程中,首先选定错误输入域的大小,然后在输入域内随机生成一个点,在该点的周围扩展成一个方形的错误区域。错误区域的面积为错误输入域的大小,其值为错误率与输入域面积的乘积。我们选取错误率为 0.001。

4.1 使用 D-ART 改进局部 ART 算法性能的分析

我们试验的第一部分是检测使用 D-ART 时改进的局部 ART 算法与原算法之间的性能比较及 f 的范围在 $[0.1 \sim 0.6]$ 时算法的性能变化。我们选定测试样本候选集的大小 N 的区间为 $[5, 25]$ 。对于 f 与 N 的每一个值,我们进行了 50,000 次仿真试验。取其平均值作为平均 F_{measure} 。对于使用 D-ART 的局部 ART 算法,我们选定 $N=10$ 。

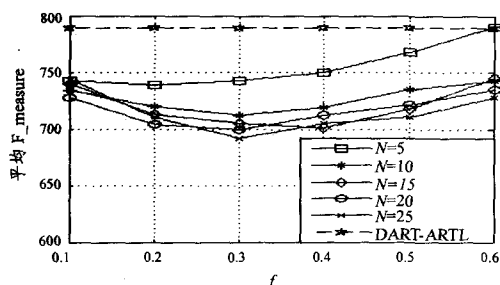


图6 使用 DART 的局部 ART(虚线部分)与改进局部 ART(实线部分)性能比较

试验结果如图 6 所示。对于候选节点集的大小为 10 的情况下,改进的局部 ART 算法比原算法的平均 F_{measure} 值减少了,即检测到第一个错误需要的测试次数减少。随着 N 的增加,平均 F_{measure} 值也变小了。当节点数从 5 增加到 10 时,效果尤其明显。但当 N 大于 15 时, F_{measure} 的取值趋于稳定。这可以解释为,当 N 的取值大于 15 时,样本候选集中的元素已经能很好地散布于目标区域,从这个集合中选出的测试样本与临近已测试集的距离已经足够淘汰掉后续增加的新样本。对于不同大小的候选集, f 取值范围为 $[0.3, 0.4]$ 时, F_{measure} 的取值趋于最小。当 f 大于 0.4 时,随着 f 取值的增加, F_{measure} 值变大。

当 f 取不同的值时,集合 E 中包含样本数的变化如图 7 所示。当 $f=0$ 时,邻近已测试样本的候选区域的面积为 0,此时相当于局部 ART 算法,集合 E 最小。随着 f 的增加,集合 E 逐渐变大。当 $f=0.4$ 时, E 的大小增加了一倍。

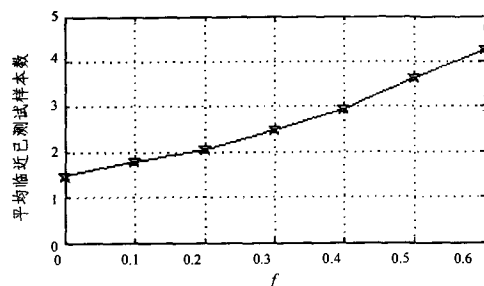


图7 集合 E 的平均大小

4.2 使用 RRT 改进局部 ART 算法性能的分析

本文试验的第二部分是使用了 RRT 时改进局部 ART 算法与原算法之间的性能比较。选择错误率为 0.001。选择 f 的范围为 $[0\%, 40\%]$,以 10% 为间隔。对 f 的每种情况,我们做了 50,000 次仿真实验,取其平均值作为 F_{measure} 。仿真结果如图 8 所示。

当 f 为 0 时,局部 ART 算法与改进的局部 ART 算法都相当于动态划分的 ART 算法,其 F_{measure} 值最大,为 800。随着 f 值的增加,两种算法的 F_{measure} 的值都开始变小。但对于任意特定的 f 值,改进的局部 ART 算法比局部 ART 算法的 F_{measure} 值都要小,表明改进过后算法的性能得到了提升。

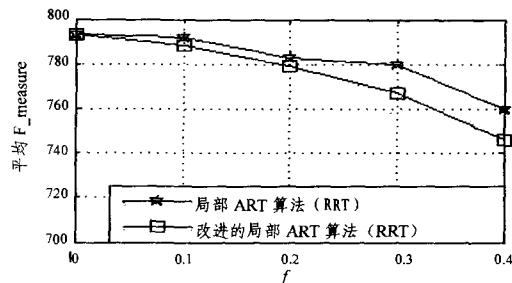


图8 使用 RRT 的局部 ART 算法与改进的局部 ART 算法性能比较

结束语 本文针对局部 ART 和动态划分的 ART 的不足,提出了改进的局部 ART 算法。D-ART 和 RRT 算法需要测试样本同所有的已测试样本进行距离计算,而改进的局部 ART 算法的距离计算仅限于临近的已测试样本点,大大降低了距离计算量。本算法通过扩大原算法临近已测试样本点的候选范围,既确保了生成的测试样本与所有已测试样本的最小距离,又克服了局部 ART 的缺陷。同时,还保证了测试样本生成的区域的合理,克服了随机划分 RRT 算法的缺陷,使测试样本的分布更加均匀。

参考文献

- [1] Chan F T, Chen T Y, Mak I K, et al. Proportional sampling strategy: guidelines for software testing practitioners. Information and Software Technology, 1996, 38: 775-782
- [2] Chen T Y, Leung H, Mak I K. Adaptive Random Testing // Proceedings of the 9th Asian Computing Science Conference (ASIAN 2004). 2004: 320-329
- [3] Chen T Y, Eddy G, Merkel R, et al. Adaptive Random Testing Through Dynamic Partitioning // Proceedings of the 4th International Conference on Quality Software (QSIC 2004). IEEE Computer Society, September 2004: 79-86
- [4] Chen T Y, Huang D H. Adaptive Random Testing by Localization // Proceedings of the 11th Asia-Pacific Software Engineering Conference
- [5] Mayer J. Restricted Adaptive Random Testing by Random Partitioning // Proceedings of the International Conference on Software Engineering Research and Practice (SER). 2006
- [6] Matsumoto M, Nishimura T, Twister M. A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. ACM Transactions on Modeling and Computer Simulation (TOMACS), 8(1): 3-30