

基于组织进化粒子群优化的测试用例自动生成*

潘晓英, 陈皓

(西安邮电大学 计算机学院, 西安 710121)

摘要: 针对组合测试用例生成问题的具体特点, 结合组织进化思想及粒子群优化算法, 设计了适合问题求解的编码方式及操作算子等, 提出了一种基于组织进化粒子群优化的测试用例自动生成算法。该方法用于选择当前局部优化覆盖的测试用例, 在此基础上构建满足两两覆盖的测试用例集。仿真实验表明, 该方法能有效地降低测试用例数目。

关键词: 组织进化; 粒子群; 测试用例; 两两覆盖

中图分类号: TP18 **文献标志码:** A **文章编号:** 1001-3695(2012)06-2065-03

doi:10.3969/j.issn.1001-3695.2012.06.015

Using organizational evolutionary particle swarm techniques to generate test cases for combinatorial testing

PAN Xiao-ying, CHEN Hao

(School of Computer Science & Technology, Xi'an University of Posts & Telecommunications, Xi'an 710121, China)

Abstract: Based on the analysis of the characteristics of combinatorial testing, this paper proposed an organizational evolutionary particle swarm algorithm (OEPST) to generate test cases for combinatorial testing. This algorithm was used to select the test cases of local optimal coverage in current environment based on these test cases, and then built a test suite satisfying the pair-wise coverage criterion. The empirical result shows that this approach can reduce effectively the number of test case.

Key words: organizational evolutionary; particle swarm; test cases; pair-wise coverage

随着软件产品化以及软件危机的不断出现, 软件测试作为发现软件缺陷的主要手段和唯一有效的方法而日益得到重视, 软件测试中处于重中之重的测试用例设计要求也随之上升到了更高的层次。在诸多的软件测试方法中, 组合测试充分考虑了系统中各种因素以及因素间相互作用可能产生的影响, 并可以根据实际需要, 用尽可能少的测试数据尽可能多地覆盖影响系统的因素。在实际的软件测试中, 测试数据通常只需达到对各个参数两两组合的覆盖。因此, 在软件测试中, 两两组合覆盖方法是一种实用而有效的方法, 但两两组合覆盖的测试数据生成一直是人们研究的重要课题^[1]。

近年来, 组合覆盖方法已经在软件测试中得到了比较成功的应用, 其中测试用例的生成技术主要包括代数方法、启发式算法和元启发式算法三类。代数方法能产生符合要求的最小测试用例集, 如基于正交表的构造方法^[2]、二水平二维组合覆盖表的构造方法^[3]等。但这类方法大多都要求因素集合中的所有因素可选取值的数量相等, 而事实上这不太可能。由于启发式算法解决 NP 问题的优越性, 它在组合测试用例生成问题中也得到了广泛的应用, 大多数采用了逐条生成测试用例的方式。每次按照一定的规则产生一条测试用例, 直到所有的测试因素全部被覆盖为止, 如 CATS^[4]、AETG^[5]、TCG^[6]、DDA^[7] 以及解空间树法 PSST^[8] 等。元启发式搜索算法^[9-11] 近年来也被广泛地应用于组合测试用例生成问题, 如爬山、模拟退火、遗传、蚁群等算法。该类方法解决测试用例生成问题时主要有两

种策略: a) 直接以组合测试用例集为目标进行搜索, 直接以二维矩阵作为可行解进行迭代, 直到得到的矩阵满足覆盖需求的条件; b) 在 one test a time 的框架下, 使用元启发式搜索算法生成单条测试用例。这类策略以单条测试用例作为可行解, 其解空间相对较小, 运行效率上优势较高。

本文考虑到已有方法的一些不足之处, 结合 one test a time 算法的主要框架, 在两两组合测试中引入了组织进化及粒子群优化算法的思想, 充分结合两者的优点, 并针对组合测试用例自动生成问题提出了一种新的算法 OEPST(using organizational evolutionary particle swarm techniques to generate test cases for combinatorial testing)。

1 两两覆盖组合测试模型

假设影响待测系统软件的参数一共有 N 个, 形成集合 $P = \{P_1, P_2, \dots, P_N\}$, 对于其中的任何一个参数 $P_i \in P, 1 \leq i \leq N$, 经过等价类划分等前期处理后包含 L_i 个可能的取值, 用 V_i 表示, 即 $V_i = \{V_{i,1}, V_{i,2}, \dots, V_{i,L_i}\}$ 。称一个 N 元组 (v_1, v_2, \dots, v_N) ($v_1 \in V_1, v_2 \in V_2, \dots, v_N \in V_N$) 为待测系统的一条测试用例; 相应地, 称由多个这样的 N 元组所构成的集合为该待测系统的一个测试用例集。如果一个测试用例集 TS 满足两两测试, 那么任意两个参数的取值组合 $(V_{i,l}, V_{j,k})$ 存在一个测试用例 TC , 使得 $(V_{i,l}, V_{j,k})$ 被 TC 覆盖。

收稿日期: 2011-11-07; **修回日期:** 2011-12-08 **基金项目:** 国家自然科学基金资助项目(61050003, 61105064); 陕西省自然科学基金资助项目(2011JM8007); 陕西省教育厅科研项目(2010JK837)

作者简介: 潘晓英(1981-), 女, 浙江丽水人, 副教授, 博士, 主要研究方向为智能计算、软件测试等(xiaoying_pan@163.com); 陈皓(1978-), 男, 讲师, 博士, 主要研究方向为进化计算、智能信息处理等。

2 组织进化粒子群优化算法

组织进化粒子群优化算法^[12]是在粒子群算法的^[13]基础上,结合组织学习模型^[14]而提出了一类新型优化算法。这类算法将粒子群中的粒子看做是具有记忆能力、通信能力、响应能力、协作能力和自学习能力的智能粒子。为了更好地保留粒子的这些特性,组织进化粒子群算法从组织的角度出发,将粒子以个体的形式组成不同的组织,充分利用了组织间个体的协作特性和自学习特性。这类算法由于充分考虑了一般粒子的记忆特性和组织之间相互竞争、相互协作的特性,使得粒子能够更好地适应环境,从而引导粒子群不断进化,最终达到全局优化的目的。同时文献[12]从理论上证明了算法具有全局收敛性,研究结果表明组织进化粒子群算法具有很强的寻优能力,可以在较小的计算代价下获得令人满意的解,并且具有很强的鲁棒性。

3 成对测试用例生成的组织进化粒子群优化

3.1 算法框架

受 one test a time 算法启发,首先构建所有参数覆盖组合 UC,然后从一个空的测试用例集开始,每次利用组织进化粒子群优化算法增加一个测试用例到测试用例集中,该测试用例为当前覆盖 UC 中参数组合最多的测试用例,同时删除该测试用例在 UC 覆盖的参数组合,一直到 UC 为空为止。具体描述如下:

- a) 初始化参数组合 UC 和测试用例集 TS;
- b) 利用组织进化粒子群优化算法计算当前覆盖 UC 最多的测试用例 TC;
- c) 将 TC 添加到集合 TS 中,同时删除 TC 覆盖的 UC 中参数组合;
- d) 重复步骤 b) 和 c),直到 UC 为空。

3.2 编码方式

编码就是将问题的潜在解使用适合算法的基因编码表示。在组合测试中,由于不同的参数取值个数不同,一个参数在基因编码中所占的位数也不同。对于每一个参数 x_i ,其可能取值为 k_i ,则该参数所占基因位长度 l_i 必须满足条件: $2^{l_i-1} \leq k_i \leq 2^{l_i}$ 。如果某一编码没有对应的取值,则可取所有取值中的任意一个,以加大覆盖力度。另一方面,为了避免参数类型的转换,本文采用参数取值的序号来表示参数的实际取值。

例如,假设一个系统 S 输入由四个参数组成 (P_1, P_2, P_3, P_4),参数取值个数依次为 4, 2, 2, 3, 那么测试用例 (3, 0, 1, 2) 就表示 P_1 取第 3 个值, P_2 取第 0 个值, P_3 取第 1 个值, P_4 取第 2 个值。整个系统可以采用长度为 6 的二进制编码表示: P_1 占用位 b_0 和 b_1 ; P_2 占用位 b_2 ; P_3 占用位 b_3 ; P_4 则占用位 b_4 和 b_5 。对于 P_4 而言,前三个编码 00、01、10 分别为 $v_{4,0}, v_{4,1}, v_{4,2}$, 编码 11 可以随机选取这三个中的一个。

3.3 适应度函数

在生成测试用例集时,要求在满足两两覆盖测试准则的前提下,测试用例数尽可能地少,因此在每一代进化时选择尽可能多地覆盖 UC 中参数组合的测试用例。设群体数量为 N, 个体 TC_i 覆盖的 UC 中参数组合数为 $C_i, 1 \leq i \leq N$, 则个体 TC_i 的适应度函数可以表示为 $f_i = C_i / \sum_{i=1}^N C_i$, 其中 N 为候选的个体个

数,且有 $\sum_{i=1}^N f_i = 1$ 。

3.4 分裂算子

若组织 org 满足 $|org| > \max_{os}$, 则进行分裂操作, 其中: $|org|$ 表示组织 org 中的成员 (粒子) 个数; $\max_{os} (< N_0)$ 表示允许的最大组织成员数; N_0 是初始化中所有组织成员个数总和; \max_{os} 和 N_0 均为预先设定的参数。分裂的具体操作为: 从 org 中选择三分之一到三分之二个成员组成子代组织 org_{c1} , 其他成员组成子代组织 org_{c2} ; 然后从当前种群中删除组织 org, 并把两个新的组织加入到下一代进化种群中。

3.5 协作合并算子

由两个组织中的领导产生两个新的个体 (类似离散交叉算子作用), 加入到原有组织当中, 同时原来两个组织进行合并。

假设两个父代组织分别为 $org_1 = \{x_1, x_2, \dots, x_{m_1}\}$ 和 $org_2 = \{y_1, y_2, \dots, y_{m_2}\}$, m_1 和 m_2 分别为组织 org_1 和 org_2 的成员 (粒子) 个数, x_p 和 y_q 分别为组织 org_1 和 org_2 的领导, 则由 org_1 和 org_2 合并产生的一个子代组织为 $org_c = \{x_1, \dots, x_{p-1}, x_p', x_{p+1}, \dots, x_{m_1}, y_1, \dots, y_{q-1}, y_q', y_{q+1}, \dots, y_{m_2}\}$ 。令 $U = (u_1, u_2, \dots, u_n)$ 和 $L = (l_1, l_2, \dots, l_n)$ 是分别由 x_p 和 y_q 相互协作后得到的新个体, 产生方式如下:

$$\begin{cases} U = (x_{p,1}, x_{p,2}, \dots, x_{p,i_1-1}, y_{q,i_1}, y_{q,i_1+1}, \dots, y_{q,i_2}, x_{p,i_2+1}, x_{p,i_2+2}, \dots, x_{p,n}) \\ L = (y_{q,1}, y_{q,2}, \dots, y_{q,i_1-1}, x_{p,i_1}, x_{p,i_1+1}, \dots, x_{p,i_2}, y_{q,i_2+1}, y_{q,i_2+2}, \dots, y_{q,n}) \end{cases}$$

其中: $1 < i_1 < n, 1 < i_2 < n$, 且 $i_1 < i_2$ 。

U 和 L 产生后, x_p' 和 y_q' 分别由下式确定:

$$x_p' = \begin{cases} U & \text{fitness}(U) \geq \text{fitness}(x_p) \\ x_p & \text{else} \end{cases}$$

$$y_q' = \begin{cases} L & \text{fitness}(L) \geq \text{fitness}(y_q) \\ y_q & \text{else} \end{cases}$$

最后从当前种群中删除 org_1 和 org_2 , 并把 org_c 加入到下一代种群。

3.6 自学习算子

自学习算子通过利用组织领导的游泳信息来指导组织个体不断进化。设两个父代组织为 $org_{i1} = \{x_1, x_2, \dots, x_{m_1}\}$ 和 $org_{i2} = \{y_1, y_2, \dots, y_{m_2}\}$, m_1 和 m_2 分别为组织 org_{i1} 和 org_{i2} 的成员个数, 且 $\text{fitness}(org_{i1}) \geq \text{fitness}(org_{i2})$ 。将自学习算子作用在由协作合并算子得到的组织 org_c 上, 得到新的组织。具体的方式为以领导的值作为参考来对其他的个体进行变异。

3.7 速度更新算子

令 $h^g(t) = (h_1^g, h_2^g, \dots, h_n^g)^T$ 表示所有组织中的最优领导经历的位置, $h^i(t) = (h_1^i, h_2^i, \dots, h_n^i)^T$ 表示第 i 个组织的领导经历过的最好位置, 则第 i 个组织中的粒子在第 k 代的速度以及位置的更新方式为

$$\begin{cases} v_j^i(k+1) = w(k)v_j^i(k) + c_1r_1(h_j^i(k) - x_j^i(k)) + c_2r_2(h_j^g(k) - x_j^i(k)) \\ x_j^i(k+1) = x_j^i(k) + v_j^i(k+1) \end{cases}$$

其中: w 是惯性权因子; c_1 和 c_2 为加速常数; r_1 和 r_2 是两个在 $[0, 1]$ 范围内服从均匀分布的随机向量。

3.8 算法描述

假设测试用例中涉及到的参数个数为 N, 分别为 (x_1, x_2, \dots, x_N) , 每个参数的可能取值分别为 k_i 个, 算法设计如下:

- a) 初始化参数组合 UC 和测试用例集 TS (TS 初始为空)。
- b) 以组织进化粒子群优化算法计算当前覆盖 UC 最多的

测试用例 TC 。

(a) 初始化。假设包含 N_0 个组织,每个组织中包含 M 个粒子,这些组织形成种群 P_k ,此时 $k \leftarrow 0$ (每个粒子就是一个测试用例)。

(b) 判断是否满足终止条件,若是则终止,跳转到 c); 否则转(c)。

(c) 更新组织中每个粒子的速度与位置信息。

(d) 从 P_k 中选择两个父代组织,完成协作合并算子,得到 org_c 。

(e) 对 org_c 执行自学习操作,得到 org_d 。

(f) 若 org_d 满足分裂条件,则执行分裂算子。

(g) 若 P_k 中组织的个数大于 1,则转(d);否则转(h)。

(h) 若 P_k 中只有一个组织,对其进行分裂条件判断,进行分裂算子操作;否则转(i)。

(i) 令 $k \leftarrow k + 1$,转(b)。

c) 将 TC 添加到集合 TS 中,并删除 TC 覆盖的 UC 中的参数组合。

d) 重复步骤 b) 和 c),直到 UC 为空。

4 实验结果与分析

为了测试所提算法 OEPST 的性能,笔者采取了五组数据进行测试,如表 1 所示。其中, R_1 表示 4 个 3 值参数组合; R_2 为 13 个 3 值参数组合; R_3 为 20 个 10 值参数组合; R_4 为 100 个 2 值参数组合; R_5 为 61 个参数,其中 15 个 4 值参数,17 个 3 值参数,29 个 2 值参数。

表 1 测试数据属性

数据集	参数个数	每个参数的取值个数
R_1	4	3
R_2	13	3
R_3	20	10
R_4	100	2
R_5	61	15(4), 17(3), 29(2)

以上文所讨论的 OEPST 实现了一个组合测试工具,其中的参数设置为 $N_0 = 10, M = 10, \max_{gen} = 20, c_1 = c_2 = 2, w = 0.4 \sim 0.9$,最大迭代次数设置为 200 代。对一种输入参数的组合,运行 5 次,并取其平均值,其运行结果如表 2 所示。同时表中列出了 AETG、IPO 和 GACT 对这五组测试数据的生成结果。

表 2 OEPST 生成结果与其他算法的比较

method	R_1	R_2	R_3	R_4	R_5
AETG	9	15	180	10	41
IPO	9	19	218	15	36
GACT	9	18	197	15	37
OEPST	9	16	185	14	34

从表中的结果来看,该算法能找到符合要求的测试用例集,且集合中包含的测试用例较少。在组合数目比较小时,算法并不占优势;但是在组合数据较大时,算法所得结果已经比较接近 AETG 的结果。

为了进一步测试算法的性能,本文对 4 值输入的参数进行了测试,其参数个数从 10 以 10 的步长增加到 50。表 3 给出了此种情况下三种方法的生成结果。

表 3 n 个 4 值输入参数的测试用例数

n	10	20	30	40	50
IPO	31	40	46	49	52
GACT	29	36	40	43	47
OEPST	29	34	39	40	43

相应地,固定输入参数的个数为 10 个,但每个参数的取值个数从 5 以步长 5 增加到 20。表 4 给出了此种情况下三种方法的生成结果。

表 4 10 个输入参数不同取值个数的测试用例数

n	5	10	15	20
IPO	45	171	363	621
GACT	44	150	324	562
OEPST	42	144	307	541

从以上的测试结果可以看出,尽管 OEPST 在参数个数较少、取值个数较少的组合情况下没有太多的优势,但是在组合数目较多的情况下优势比较明显,能比其他方法显著减少生成的组合测试用例数目。

5 结束语

本文结合组合测试用例生成特点,引入了独特的组织进化思想及性能优越的粒子群优化算法,并结合 AETG 生成框架,提出一种基于组织进化粒子群优化的测试用例自动生成算法 OEPST。仿真实验结果表明,该方法在组合数据较大的情况下具备了比较明显的优势,能显著减少测试用例生成的个数。

参考文献:

- [1] KOBAYASHI N, TSUCHIYA T, KIKUNO T. A new method for constructing pair-wise covering designs for software testing[J]. *Information Processing Letters*, 2002, 81(2): 85-91.
- [2] LIE Yu, TAI K C. Ip-parameter-order: a test generation strategy for pair-wise testing[J]. *IEEE Trans on Software Engineering*, 2002, 28(1): 109-111.
- [3] 聂长海,徐宝文,史亮. 一种新的二水平多因素系统两两组合覆盖测试数据生成算法[J]. *计算机学报*, 2006, 29(6): 841-848.
- [4] GRINDAL M, OFFUTT J, ANDLER S F. Combination testing strategies: a survey, GMU Technical Report ISE-TR-04-05[R]. [S.l.]: George Mason University, 2004.
- [5] COHEN D M, DALAL S R, FREDMAN M L, et al. The AETG system: an approach to testing based on combinatorial design[J]. *IEEE Trans on Software Engineering*, 1997, 23(7): 437-444.
- [6] TUNG Tu-wen, ALDIWAN W S. Automating test case generation for the new generation mission software system[C]//Proc of IEEE Aerospace Conference. 2000: 434-437.
- [7] COLBOURN C J, COHEN M B, TURBAN R C. A deterministic density algorithm for pairwise interaction coverage[C]//Proc of IASTED International Conference on Software Engineering. 2004: 345-352.
- [8] 史亮,聂长海,徐宝文. 基于解空间树的组合测试数据生成[J]. *计算机学报*, 2006, 29(6): 849-857.
- [9] COHEN M B, COLBOURN C J, LING A C H. Constructing strength three covering arrays with augmented annealing[J]. *Discrete Mathematics*, 2007, 308(13): 2709-2722.
- [10] COHEN M B, GIBBONS P B, MUGRIDGE W B. Constructing test suites for interaction testing[C]//Proc of the 25th International Conference on Software Engineering. Washington DC: IEEE Computer Society, 2003: 38-48.
- [11] NUMELA K H. Upper bounds for covering arrays by tabu search[J]. *Discrete Applied Mathematics*, 2004, 138(9): 143-152.
- [12] 丛琳,沙宇恒,焦李成. 组织进化粒子群算法用于求解函数优化[J]. *模式识别与人工智能*, 2007, 20(2): 145-153.
- [13] Van den BERGH F, ENGELBRECHT A P. A cooperative approach to particle swarm optimization[J]. *IEEE Trans on Evolutionary Computation*, 2004, 8(3): 225-239.
- [14] 刘静,钟伟才,刘芳,等. 组织协同进化分类算法[J]. *计算机学报*, 2003, 26(4): 446-453.