用改进的遗传算法实现架构恢复*

李青山+, 陈 平

(西安电子科技大学 软件工程研究所,陕西 西安 710071)

Implementing Architecture Recovery by Using Improved Genetic Algorithm

LI Qing-Shan+, CHEN Ping

(Software Engineering Institute, Xidian University, Xi'an 710071, China)

+ Corresponding author: Phn: 86-29-8202457, Fax: 86-29-8202458, E-mail: liqingshan@sei.xidian.edu.cn

http://www.xidian.edu.cn

Received 2002-11-05; Accepted 2003-03-04

Li QS, Chen P. Implementing architecture recovery by using improved genetic algorithm. *Journal of Software*, 2003,14(7):1221~1228.

http://www.jos.org.cn/1000-9825/14/1221.htm

Abstract: Architecture recovery is crucial to supporting software maintenance and evolution. The clustering problem that could implement architecture recovery is considered as optimizing problem in this paper. Through improving important parameters and core steps of general genetic algorithm, such as initial population, select operator, self-adapting ability of crossover probability and mutation probability, a hybrid genetic clustering algorithm (HGCA) is designed and implemented. An experiment is given to analyze the availability, effectiveness and synthetical performance of the algorithm. The results show that compared to general GA, the HGCA can produce good initial population, better convergence efficiency and convergence precision. Moreover, the value of the MoJo similarity metrics presents the correctness and effectiveness of HGCA recovering software architecture.

Key words: architecture recovery; clustering algorithm; genetic algorithm; object oriented reverse engineering

摘 要: 高层架构恢复对软件维护和软件进化至关重要.把实现架构恢复的聚类问题看作优化问题,通过对常规遗传算法中初始群体产生策略、选择操作方法、交叉概率和变异概率的自适应性等重要参数和关键环节的改进,设计并实现了混合遗传聚类算法(hybrid genetic clustering algorithm,简称 HGCA).同时也对该算法的有效性和综合性能进行了实验分析,结果表明,该算法对初始群体的产生有较好的约束作用.与传统遗传算法相比,它的群体性能和收敛性能都较优,且收敛精度高.同时,基于 MoJo 度量模型的相似性度量值充分说明了 HGCA 算法对架构恢复的正确性和有效性.

关键词: 架构恢复;聚类算法;遗传算法;面向对象逆向工程

中图法分类号: TP311 **文献标识码**: A

逆向工程作为软件工程中一个新的研究领域,目前受到广泛关注[1].逆向工程通过分析目标系统,发现系统

* Supported by the Defence Pre-Research Project of the 'Tenth Five-Year-Plan' of China No.413060601 ("十五"国防预研基金)

第一作者简介: 李青山(1973－),男,甘肃西峰人,博士生,讲师,主要研究领域为逆向工程,程序理解,面向对象,软件体系结构.

元素以及它们之间的关系,产生系统不同形式和不同层次的抽象表示,完成从程序空间到设计空间,再到问题空间的映射.逆向分析涉及到多个层次,高层架构的恢复可以帮助用户理解目标系统框架结构,了解系统组件及其关系,这对系统维护和进化至关重要.

早期逆向工程中架构恢复的研究主要面向过程范型,常见的架构信息获取与分析手段主要有:基于关系查询[2]、数据挖掘以及体系结构样式[3]等方法,聚类也是一类有效的逆向分析方法.自动聚类算法最早用于人工智能领域中,用于聚合相似的实体,在逆向工程领域,聚类用于发现程序的结构.许多工具都采用聚类算法实现高层架构恢复,比如 Rigi,Arch[4]等,都以实体间关联关系的多少反映其耦合度,以"块间松耦合、块内紧耦合"为度量准则,主要的聚类算法有图理论算法、构造算法和层次化算法[5].图理论算法利用结点聚合实现聚类,主要包括最小横跨树聚类算法和聚合算法两类;构造算法包括密度搜索技术、模型分析以及基于模糊集合的聚类技术等;而层次聚类算法主要有自下而上逐层合并与自上而下逐层分裂两类.对于面向对象系统,架构恢复的模型和算法的构造应该以这类系统结构特征和交互特征为依据,而且,应该结合应用领域和问题域的特点.

本文结合面向对象范型的特征,设计了源代码模型和相似性度量,用于抽象 OO 系统,度量类之间的结构关系和交互关系.以此为基础,抽取目标系统高层架构的聚类问题被看作优化问题.结合常规遗传算法的全局搜索能力好、局部搜索速度慢、易在搜索中后期陷入随机搜索的特点,我们通过改进初始群体产生策略、迭代结束策略、选择方法、交叉概率和变异概率的自适应性等遗传算法关键要素,设计并实现了混合遗传聚类算法.

我们的工作以国家"十五"军事电子预研重点课题"系统应用软件逆向工程开发工具研究"为背景,研制一组逆向工程工具,以提供符合 UML 标准的动态模型和高层模型的逆向生成.这组工具支持从源程序逆向地产生相应的动态模型和高层抽象模型,并集成在 Rational Rose®建模环境中,从而为用户理解面向对象目标系统动态模型和高层静态模型提供计算机辅助支持.混合遗传聚类算法是本工具中实现高层抽象模型的核心算法.

1 聚类模型

1.1 聚类环境

聚类的目的是为了得到系统的高层架构划分,从而理解系统高层子系统的组成及其相互间的依赖关系.每个子系统完成特定功能,比如监控、解析、代码生成等,或提供一套特定服务,比如内存管理、文件管理等.子系统从结构组织和功能实现上一般具有紧耦合的特点,而子系统之间的依赖反映了一定的高层交互特征,具有松耦合特征.传统聚类算法一般面向变量访问、过程调用等关系的多少往往用于度量实体间的关联度.而对于面向对象范型,由于其以类为主要关注点和结构单位,类对数据和操作进行了封装,高层架构元素范围的划分也以类为结构单位实现特定功能,所以对面向对象目标系统聚类,聚类单位也应该为类,元素间关联度的度量也应该以类之间的结构关系和交互关系为依据.考虑到类间关系的多样性,不能仅仅以类间关联关系的多少来度量相互间的耦合度,反映类间关系的相似性度量(similarity metrics)应能反映类间的静态结构关系和动态交互关系.基于此,结合多种经典的度量模型[6],我们用聚类质量 ClQua 度量类之间以及类簇之间的关联度.

聚类实现的 3 个要素是源代码模型、相似性度量和聚类算法.在我们的逆向工程工具中,分别用带权类依赖图 WCDG、聚类质量 ClQua、混合遗传聚类算法 HGCA(hybrid genetic clustering algorithm)实现这 3 个要素.具体而言,反射解析器解析系统源代码生成带权类依赖图.反射解析器是本工具基于反射原理和开放编译的元对象协议实现的面向对象程序分析工具[7,8].以带权类依赖图为输入混合遗传聚类算法通过优化搜索找到类集合的一个以聚类质量为适应度函数的最优或次优划分对应目标系统高层架构的一种结果.这个结果被逆向分析表示工具以特定格式呈现,例如本工具中以集成在 Rational Rose 中的 package 的 sterotype 表示.聚类环境如图 1 所示.

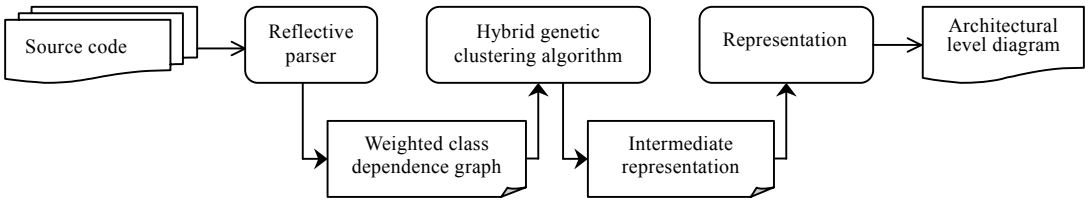

Fig.1 Automated clustering environment of the reverse engineering tool
图 1 逆向工程工具自动聚类环境

1.2 源代码模型与相似性度量模型

定义 1.1. 类 C_A 和 C_B 的操作相关性(operation coupling)度量 $OpCpl_{A,B}$ 定义为类 A 访问类 B 的操作的个数与类 B 访问类 A 的操作的个数之和.

定义 1.2. 类 C_A 和 C_B 的继承相关性(inheritance coupling)度量 $InCpl_{A,B}$ 定义为类 A 到类 B 或类 B 到类 A 在它们的继承树上的路径长度的倒数.

定义 1.3. 类 C_A 和 C_B 的类相关性(class coupling)度量 $ClCpl_{A,B}$ 定义为 $ClCpl_{A,B}=OpCpl_{A,B}+InCpl_{A,B}$.

定义 1.4. 带权类依赖图 WCDG(weighted class dependency graph)定义为 $WCDG=(C,A,W)$,其中:

- $C=(c_i|c_i$ 为面向对象目标系统的一个类);
- $A=(\langle u,v\rangle|u,v\in C$,且 $P(u,v)$成立,谓词 $P(u,v)$说明弧$\langle u,v\rangle$对应权值 $w_j,w_j\in W)$;
- $W=(w_j|\ w_j=ClCpl_{u,v},w_j\in R,R$ 为实数集合),W 为权值集合.

定义 1.5. 设类簇 B_i 内包含 N_i 个类,E_i 个类间的依赖关系,每个依赖关系对应弧的权值为 $W_k(k=1,2,...,E_i)$,则类簇 B_i 内聚度(cohension)Coh_i 定义为

$$Coh_i=2\sum_{k=1}^{E_i}W_k\Big/N_i(N_i-1).$$

Coh_i 值越大,类簇 B_i 内类间相关度越高.

定义 1.6. 设类簇 B_i 和 B_j 分别包含 N_i 和 N_j 个类,$E_{i,j}$ 个类簇间的依赖关系,每个类簇间依赖关系对应弧的权值为 $W_k(k=1,2,...,E_{i,j})$,则类簇 B_i 和 B_j 耦合性(coupling)$Cpl_{i,j}$ 定义为

$$Cpl_{i,j}=\sum_{k=1}^{E_{i,j}}W_k\Big/2N_iN_j\ .$$

$Cpl_{i,j}$ 值越大,类簇 B_i 和 B_j 间相关度越高.

定义 1.7. 设 $WCDG$ 被划分成 m 个类簇,Coh_i 为第 i 个类簇的内聚度,$Cpl_{i,j}$ 为第 i 个类簇和第 j 个类簇之间的耦合度,则该 $WCDG$ 的聚类质量(clustering quality)$ClQua$ 定义为

$$ClQua=\begin{cases}(m-1)\sum_{i=1}^{m}Coh_i\Big/\sum_{i,j=1}^{m}Cpl_{i,j}, & \text{if } m>1\\ Coh_1, & \text{if } m=1\end{cases}.$$

利用 Coh_i 均值与 $Cpl_{i,j}$ 均值之比定义 $ClQua$.$ClQua$ 值越大,该 m 个类簇簇间耦合度相对较低,簇内内聚度相对较高.

1.3 理想算法

定义 1.8. 设 $S=(c_1,c_2,...,c_n),c_i\in C,C\in$WCDG,$P=(B_1,B_2,...,B_m)$是 S 的一个非空子集的集合.B_i 为 S 的一个类簇当且仅当

- $\bigcup_{i=1}^{m}B_i=S$;
- $B_i\cap B_j=\varnothing,\forall 1\le i,j\le m,i\ne j$,

且称 P 为 S 的一个 m 阶划分.

在本工具中,聚类问题本质上是以特定度量为依据的优化问题.具体而言,问题空间的点为 WCDG 中类集合的划分,求解反映系统高层架构信息的聚类运算对应在这个空间中寻找聚类质量为最大的划分.理想状况下,通过求得 S 的每个划分,比较它们之间的聚类质量,可以求得最优解.但由于 n 个元素的 k 阶不同的划分个数为

$$S_{n,k}=\begin{cases}1, & k=1\text{或者}k=n\\ S_{n-1,k-1}+kS_{n-1,k}, & \text{其他}\end{cases}.$$

这个值随集合 S 的大小呈几何级数增长.所以,理论上最优的聚类算法实际上无法有效聚类.

定义 1.9. 设 $P=(B_1,B_2,...,B_m)$是类集合 S 的一个 m 阶划分,则划分 NP 是划分 P 的邻接划分当且仅当:除划分 P 中的一个类簇中的一个类在 NP 中的另一个类簇中之外,其他部分 NP 与 P 完全相同.并定义操作 $NP(P)$为邻接操作.

以邻接操作为基础,利用改进的遗传算法解决类集合划分组合优化问题是我们的思路.

2 混合遗传聚类算法 HGCA

2.1 遗传算法特点

遗传算法[9]是一类以 Darwin 自然进化论与 Mendel 遗传变异理论为基础的求解全局优化问题的仿生型算法,其本质是一种求解问题的高效并行全局搜索算法.它能在搜索过程中自动获取和积累有关搜索空间的知识,并自适应地控制搜索过程,从而得到最优解或准最优解.遗传算法是一个以适应度函数为依据,通过对群体个体施加遗传操作实现群体内个体结构重组的迭代处理过程.在这一过程中,群体个体一代一代地得以优化并逐渐逼近最优解.

许多传统搜索算法都是单点搜索算法,即通过一些变动规则,问题的解从搜索空间中的当前解(点)移到另一解(点).这种点对点的搜索算法,对于多峰分布的搜索空间常常会陷于局部的某个单峰的优解.而遗传算法是采用同时处理群体中多个个体的方法,同时搜索空间中的多个解进行评估,使遗传算法具有良好的全局搜索性能,减少了陷于局部优解的风险.

尽管遗传算法比其他传统搜索方法有更强的鲁棒性,但它更擅长于全局搜索,局部搜索能力不足.研究发现,遗传算法可以用极快的速度达到最优解的 90%左右,但要达到真正的最优解则要花费很长的时间.一些对比实验还表明,如果兼顾收敛速度和解的品质两个指标,单纯的遗传算法未必比其他方法更优越[9].采用混合模型可以利用传统方法的搜索到局部极小点速度快、精度高的特点.有效地提高遗传算法的局部搜索能力.从而改善其解的品质.

另一方面,遗传算法对问题特定的知识利用较少.在求解组合优化问题时,若不能利用问题的固有知识来缩小搜索空间,则会产生搜索的组合爆炸.所以,设计遗传算法时往往需要在其通用性与有效性之间进行折衷.HGCA 算法通过对遗传算法中控制参数和遗传操作的改进,改善了其全局搜索性能和收敛性能.具体而言,在初始群体产生、变异概率自适应控制方面充分利用领域知识;在选择算子设计和交叉操作实现方面结合爬山搜索算法核心思想.

2.2 初始群体产生策略

在常规的和许多改进的遗传算法中,初始种群的产生主要采用完全的随机方式,而没有解决初始种群中各个体在解空间中的分布情况.这有可能让许多个体都集中在某一局部区域内,不利于扩大搜索空间和收敛到全局最优解.HGCA 算法首先在初始种群的产生上要求各个体之间保持一定的距离,使各个体尽可能均匀地分布在整个解空间上,从而避免进化初期的未成熟收敛.

定义 2.1. 已知 $NP(P)$为邻接操作.设运算 $Ex(S)$为随机求集合 S 中一个元素的运算,则个体 $A_i(t)$的 p 级邻接划分定义为

$$A_j(t)=\overbrace{Ex(NP(...Ex(NP(Ex(NP(}^{2p}A_i(t)))))...)),$$

p 定义为个体 $A_i(t)$和 $A_j(t)$之间的邻接距离.

产生策略:

1. 用户根据领域特征,通过交互输入划分$\alpha^*(0)$作为初始群体$\alpha(0)$的启发元;
2. 把$\alpha^*(0)$加入到初始群体$\alpha(0)$中;

3. 产生$\alpha^*(0)$的若干 p 级邻接划分,加入到初始群体$\alpha(0)$中;

4. 如果不足群体规模,选择当前适应度函数值最大的个体,再求其若干 p 级邻接划分,加入到初始群体中;否则,转到步骤 5;

5. 直至达到群体规模,初始群体$\alpha(0)$产生.

实验验证,p 应该取 0.05N~0.1N,其中 N 为群体规模.

初始种群采取这种方式产生就能保证初始群体中的各个体间有较明显的差别,使它们能够比较均匀地分布在解空间上,保证初始种群含有较丰富的模式,从而增加搜索收敛于全局最优解的可能.

2.3 交叉与变异的自适应性

交叉和变异是遗传算法中两个起重要作用的算子.交叉操作的作用是组合交叉个体中有价值的信息产生新的后代,它在群体进化期间大大加快搜索速度,是主要算子;变异操作的作用是保持群体中基因的多样性,避免搜索的局部化,是偶然的、次要的,是辅助算子.交叉算子需要保证前一代中优秀个体的性状能在后一代的新个体中尽可能得到遗传和继承,是保证遗传算法快速收敛的关键.但它往往会陷于局部,变异搜索可随机扩大搜索范围,并且可迁,正好弥补了交叉搜索的局限性.但是,在一般遗传算法中,变异方向不明确,很少反映问题域信息,变异效果受到影响.

HGCA 算法充分利用传统搜索算法的强局部搜索能力,将爬山思想应用到交叉操作中,使得优化搜索的收敛速度加快.同时,根据聚类环境中个体的领域特征,HGCA 算法可以自适应地改变交叉概率和变异概率的大小,从而提高算法的聚类准确性和计算效率.具体而言,根据领域特征,建立一个个体模式库 IPL.该模式库存储着特定领域(比如本项目是军事指挥控制领域)高层架构模式.在算法迭代过程中,随着不断产生新群体的代数增加,交叉操作产生新个体的可能性降低,这时,交叉概率自适应减小.当减小到一定阈值时,进行变异操作.按照一定概率从个体模式库抽取一个或多个个体代替当前群体中适应度值小的若干个体.在本算法中,交叉概率和变异概率不是固定的,而是随着优化搜索进度自适应地变化,从而克服了一般遗传算法进化中后期由于个体竞争减弱而引起的随机搜索趋势的缺陷.

2.4 其他控制参数

本算法在迭代停止条件设定方面采用两种条件的结合形式,即算法连续两次迭代所获得的改变量小于要求的精度值条件与达到算法规定的最大迭代次数的结合.在具体实现过程中,用$|\Sigma\,\alpha(t+1)-\Sigma\,\alpha(t)|/N$<某个阈值 ξ 表达准则“相邻两代群体平均适应度值差别”;用$|\alpha^*(t+1)-\alpha^*(t)|$<某个阈值$\eta$表达准则“最优个体适应度值差别”,其中$\alpha^*(t)$为第 t 代最优个体.这两个适应度差别的综合作用用于衡量进化平缓度,最大迭代次数 Max$G$ 用于控制绝对终止条件.

在选择算子设定方面,本算法采用期望值法,它比常用的适应度比例法性能更优[9].适应度比例法是基于概率的选择,存在统计误差.期望值方法先按照期望值 $M = Nf_i \Big/ \sum_{i=1}^{N} f_i$ 的整数部分安排个体被选中的次数,而对其期望值的小数部分再应用适应度比例法.

2.5 HGCA算法框架

混合遗传聚类算法.

输入:目标系统的带权类依赖图 WCDG、初始群体$\alpha(0)$的启发元$\alpha^*(0)$、个体模式库 IPL;

输出:解空间的一个个体,对应 WCDG 中类集合 C 的一个划分 P.

步骤:

1. N←群体规模值 //* 用户输入群体规模,范围为 0.08|C|~0.30|C|
2. 以个体$\alpha^*(0)$为启发元产生初始群体$\alpha(0)$
3. Max$G$←最大代值;ξ←阈值 1;η←阈值 2;μ←阈值 3 //* Max$G$ 控制绝对终止条件;ξ为度量相邻两代个体 //* 适应度差别的阈值;ξ为度量相邻两代最优个体适应度差别的阈值;μ为度量变异界限的阈值.
4. DO WHILE (t<Max$G$) //* t 为群体进化的代值,初值为 0

4.1. 计算当前代群体的每个个体的适应度值 $ClQua(\alpha_i(t)),i=1,2,\ldots,N$;

4.2. 以期望值 $f_i/\sum f_i/N$ 和概率 $f_i/\sum f_i$ 进行选择操作,共有 $m(m<N)$个个体被选中.设个体$\alpha_i(t),i=1,2,\ldots m$.被选择 k_i 次(其中$\sum k_i=N$);

4.3. DO WHILE ($k<=m$) //*用于产生新一代群体

4.3.1. 对被选中的每个个体求其前 k_i 个适应度值高的邻接划分 NP 爬山

4.3.2. 把 k_i 个 NP 划分对应的个体加入到新一代群体$\alpha(t+1)$中

4.4. END DO

4.5. $p_m \leftarrow 1/p_{\text{new}}(\alpha(t+1),\alpha(t))$ //* $p_{\text{new}}(\alpha(t+1),\alpha(t))$为从群体$\alpha(t)$到群体$\alpha(t+1)$产生新个体的可能性,与变异概率 p_m 是反比关系

4.6. IF ($p_m<\mu$) THEN 从个体模式库中抽取若干个体代替当前群体中适应度值小的若干个体 //*变异操作

4.7. IF ($|\sum\alpha_i(t+1)-\sum\alpha_i(t)|/N<\xi \&\& |\alpha^*(t+1)-\alpha^*(t)|<\eta$) THEN 转向步骤 6

4.8. $t \leftarrow t+1$

5. END DO

6. 计算群体$\alpha(t)$中每个个体的适应度值 $ClQua(\alpha_i(t)),i=1,2,\ldots,N$,求得最优个体$\alpha^*(t)$

7. $\alpha^*(t)$即为最优解点,对应类集合 C 中的一个划分 P //*根据用户要求,可以得到多个次优解

3 实验研究与性能分析

基于本算法,我们已经完成了本项目要求的逆向工具生成 2.0 版本,可以实现高层架构的恢复.我们用自主开发的客户服务系统平台产品 iCALL®的一个关键模块进行了实验,该模块有 67 个类.对实验结果从架构恢复准确性和算法综合性能两个方面进行了分析.

3.1 有效性分析

我们用 MoJo 度量模型[10]来度量两个划分的差异性.由于测试系统是我们自主开发的实际可运行系统.其高层架构模型与代码有着良好的一致性且有对应文档.我们以该高层架构模型对应的类集合的划分作为标准的参考划分,以本算法中重要参数作为测试点进行多次聚类,同时说明 HGCA 算法的有效性和参数变化对聚类结果的影响.

定义 3.1[10]. 设 A,B 为类集合的两个划分,操作 Move 定义为把划分中一个类簇中的一个类移动到另一个类簇中;操作 Join 定义为把两个类簇合并为一个类簇;$Mno(A,B)$定义为把划分 A 转换为划分 B 的最小的 Move 和 Join 操作次数,则 MoJo 度量模型定义为 $MoJo(A;B)=\text{Min}(Mno(A;B),Mno(B;A))$.

MoJo 距离为 0 表示两个划分完全相等;MoJo 距离越大,两个划分差异性越大.

定义 3.2. 设划分 B 为权威标准的参考划分,则划分 A 的相似质量(similarity quality)定义为 $SimQua(A)=(1-MoJo(A,B)/n)\times 100\%$,$n$ 为类集合中元素个数.

在本实验中,权威标准的参考划分为测试系统的高层架构模型对应的划分,混合聚类算法的 3 个主要参数为初始群体、群体规模、变异概率.

Table 1 The influence of the parameters to clustering effect

表 1 参数变化对聚类效果影响结果比较

Initial population	Population size	Mutation probability	Similarity quality (%)
HGCA strategy	10	0	62.08
Stochastic strategy	10	0	59.55
HGCA strategy	20	0	63.33
Stochastic strategy	20	0	60.02
HGCA strategy	10	(0,0.01]	79.34
Stochastic strategy	10	(0,0.01]	72.97
HGCA strategy	20	(0,0.01]	81.34
Stochastic strategy	20	(0,0.01]	74.12

实验结果说明,在群体规模和变异概率两个参数相同的情况下,采用 HGCA 的各个体之间保持一定距离的初始群体产生策略进行聚类后,产生的划分的相似质量比随机产生的初始群体后聚类得到的划分的相似质量要大.这表明,HGCA 策略产生的初始群体比随机产生的初始群体效果要好;反映特定域信息的变异操作可以提高高层架构恢复的准确性;相对较大的群体规模设置对算法有效性有益.

3.2 综合性能分析

我们用在线指标、离线指标、最优指标和运行指标[11]这 4 种性能指标度量本算法在应用中性能的好坏.为了比较说明 HGCA 的性能优点,我们以常规 GA 作为参照.

定义 3.3(在线指标). $X_{\text{online}}(T)=\frac{1}{T}\sum_{t=1}^{T}f_e(t)$.其中,$T$ 是进化代数,$f_e(t)$是第 t 代的平均适应度函数.$X_{\text{online}}(T)$表示到 T 代为止所有适应度函数值的平均性能.它考虑的是群体的性能,在线指标用于说明算法在线性能.

定义 3.4(离线指标). $X_{\text{offline}}(t)=\frac{1}{T}\sum_{t=1}^{T}f_e^*(t)$.其中,$f^*_e(t)$是第 t 代最好的个体的适应度函数值,$X_{\text{offline}}(T)$表示至第 T 代每次最好的适应度函数值的平均.它只考虑最优点进化趋势,群体的性能不考虑.离线指标用于说明算法的收敛性.

定义 3.5(最优指标). $f_{op}(t)=f^*(t)$,其中,$f^*(t)$为到第 t 代为止,最好的适应度函数值.

定义 3.6(运行指标). 有两个值,$\overline{f}_{run}$ 为连续运行最优平均指标,ρ_{run} 为连续运行最优方差指标.$\overline{f}_{run}=\frac{1}{r}\sum_{j=1}^{r}f_j^*(T)$,$\rho_{run}=\frac{1}{r}\sqrt{\sum_{j=1}^{r}(f_j^*(T)-\bar{f}_{run})^2}$,其中 $f_j^*(T)$ 是第 j 次经过 T 次迭代后遗传计算所求得的最优目标,$\overline{f}_{run}$ 表示 r 次独立运行所得最优目标值的平均,ρ_{run} 是 r 次运行所得最优目标值的标准偏差.由于遗传算法中包含许多随机性操作,依靠一次运行不能完全说明问题,而且易引起数据污染,因此连续运行指标是重要的,$\overline{f}_{run}$ 用于说明算法的总体性能,ρ_{run} 用于说明算法的稳定程度.

从图 2 和图 3 可以看到,HGCA 比常规 GA 的在线性能和离线性能都好,说明 HGCA 的群体性能和收敛性都较优,能够更快地达到最优并且精度更高.

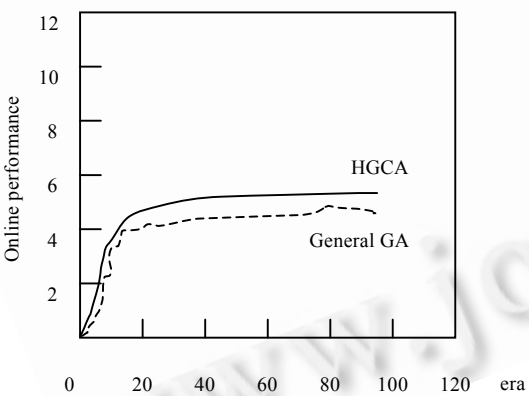

Fig.2 Comparison of online performance between HGCA and general GA

图 2 HGCA 与常规 GA 在线性能比较

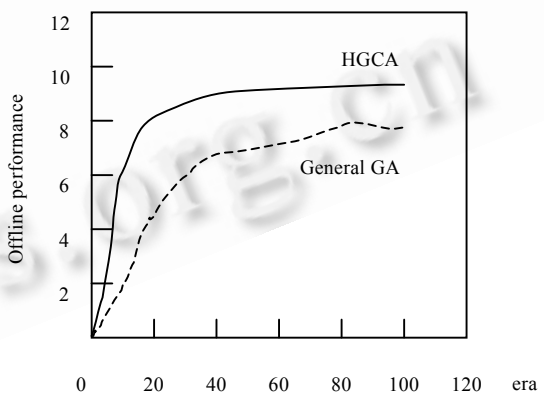

Fig.3 Comparison of offline performance between HGCA and general GA

图 3 HGCA 与常规 GA 离线性能比较

Table 2 Comparison of computation result between HGCA and general GA

表 2 HGCA 与常规 GA 计算结果比较

	$\overline{f}_{\text{run}}$	ρ_{run}	$f_{op}(T)$
HGCA	6.527	0.784	5.046
General GA	5.298	0.429	4.921

从表 2 的计算结果可知,HGCA 的 $\overline{f}_{\text{run}}$,$\rho_{\text{run}}$ 比常规 GA 要大,说明它运行的总体性能较好,每次运行结果接近最优的概率高,算法的稳定程度高.$f_{op}(T)$是 20 次运行中精度最高一次的结果.HGCA 比常规 GA 的收敛精度要高.

4 结束语

本文把逆向工程中高层架构恢复的聚类问题看作是优化问题,通过改进初始群体产生策略、迭代结束策略、选择方法、交叉概率、变异概率的自适应性等常规遗传算法关键参数,实现了混合遗传聚类算法.通过实验研究,从聚类的有效性和综合性能两个方面分析了算法效果.实验结果表明,HGCA 的群体性能和稳定性能较优,能够更快地达到最优并且精度高.同时,基于遗传算法实现高层架构恢复,可以同时得到多个次优解,为用户提供参考.通过用户交互,领域知识和专家知识的介入会进一步提高架构恢复的准确性和合理性.

References:

[1] Breuer PT, Lano KC. Creating specifications from code: Reverse engineering techniques. Journal of Software Maintenance: Research and Practice, 1991,(3):145~162.

[2] Murphy GC, Notkin D, Sullivan K. Software reflexion model: Bridging the gap between source and higher-level model. In: Proceedings of the 3rd ACM SIGSOFT Symposium on the Foundations of Software Engineering. New York: ACM Press, 1995. 18~28.

[3] Harris DR, Reubenstein HB, Yeh AS. Reverse engineering to the architectural level. In: Proceedings of the 17th International Conference on Software Engineering ICSE. New York: ACM Press, 1995. 186~195.

[4] Bellay B, Gall H. A comparison of four reverse engineering tools. In: Proceedings of the 4th Working Conference on Reverse Engineering WCRE'97. Amsterdam, 1997. 2~12. http://www.science.uva.nl/research/WCRE97/.

[5] Wiggerts T. Using clustering algorithms in legacy systems remodularization. In: Proceedings of the 4th Working Conference on Reverse Engineering WCRE'97. Amsterdam, 1997. 33~44. http://www.science.uva.nl/research/WCRE97/.

[6] Chidamber SR, Kemerer CF. A metrics suite for object oriented design. IEEE Transactions on Software Engineering, 1994,20(6):476~493.

[7] Chen P. A study on reflective architecture and object identity [Ph.D. Thesis]. Xi'an: Xidian University, 1991 (in Chinese with English abstract).

[8] Wang W. A mechanism based on the reflection and open compilers to realize the instrumentations of C++ programs [MS. Thesis]. Xi'an: Xidian University, 2003 (in Chinese with English abstract).

[9] Chen GL, Wang XF, Zhuang ZQ, Wang DS. Genetic Algorithms and Its Applications. Beijing: People's Post and Telecommunications Publishing House, 1996 (in Chinese).

[10] Tzerpos V, Holt RC. MoJo: A distance metric for software clusterings. In: Proceedings of the 6th Working Conference on Reverse Engineering WCRE'99. Atlanta, 1999. 187~193.

[11] Starkweather T, McDaniel S. A comparison of genetic sequencing operators. In: Belew R, Booker L, eds. Proceedings of the 4th International Conference on Genetic Algorithms. Los Altos: Morgan Kaufmann Publishers, 1991. 69~76.

附中文参考文献:

[7] 陈平.反射结构和对象标识的研究[博士学位论文].西安:西安电子科技大学,1991.

[8] 王伟.一种基于反射和开放编译技术的 C++植入机制[硕士学位论文].西安:西安电子科技大学,2003.

[9] 陈国良,王煦法,庄镇泉,王东生.遗传算法及其应用.北京:人民邮电出版社,1996.