博士论丛

湖南省自然科学基金资助项目
湖南省教育厅资助项目
湖南商学院优秀博士学位论文著作资助项目

遗传程序设计技术及应用研究

⊙王四春　著

中南大学出版社

·博士论丛·

湖南省自然科学基金资助项目　湖南省教育厅资助项目
湖南商学院优秀博士学位论文著作资助项目

遗传程序设计技术及应用研究

王四春　著

中南大学出版社

图书在版编目(CIP)数据

遗传程序设计技术及应用研究/王四春著. —长沙:中南大学出版社,2006.5

ISBN 7-81105-324-1

Ⅰ.遗… Ⅱ.王… Ⅲ.程序设计-研究 Ⅳ.TP311.1

中国版本图书馆 CIP 数据核字(2006)第 070459 号

遗传程序设计技术及应用研究

王四春 著

□责任编辑 邓立荣
□责任印制 文桂武
□出版发行 中南大学出版社
社址:长沙市麓山南路 邮编:410083
发行科电话:0731-8876770 传真:0731-8710482
□印 装 中南大学印刷厂

□开 本 850×1168 1/32 □印张 7.25 □字数 175 千字
□版 次 2006 年 5 月第 1 版 □2006 年 5 月第 1 次印刷
□书 号 ISBN 7-81105-324-1/TP·012
□定 价 25.00 元

图书出现印装问题,请与经销商调换

前 言

本专著研究的课题是在湖南省自然科学基金项目“基于GP理论的多准则决策函数稳定性分析研究”(05JJ40103)和湖南省教育厅项目“网络共同进化算法及应用研究”(04C313)的资助下完成的。

目前，GP(Genetic Programming，遗传程序设计，又称遗传编程)是十分活跃的研究领域，被视为解决多目标决策问题、软件复用工程、CASE等复杂问题分析设计和软件危机的强有力工具。然而，当前GP在整体上还处在初步的研究阶段。应用数学工具建立较为完善的GP理论，对其问题约束、个体表示、自定义函数、适应度函数、选择策略、遗传算子以及与此相关的算法设计、结构描述、数学建模开展深入的探索，有助于实际问题的解决。

专著中研究了GP的模式定理、算法性能优化技术、个体程序树表示、自定义函数共同进化模型、快速求解适应度函数权值方法、多目标决策函数模型和稳定性分析方法以及软件复用技术，给出了GP的模式定理及其在微观和宏观概念下的定义，深入解释了模式创建的内在机制和算法的进化行为。运用Markov链分析法，验证了GP的收敛性问题，得出了在最优值保留条件下，算法可收敛到全局最优解。

采用排序生成法、最优生成法和平均生成法三种改进方法，提高了GP算法的收敛性能，加快寻优过程，避免因出现过大的群体规模而产生的负面作用。一致性交叉可以避免传统交叉算子在代码片段选择上的盲目性，能有针对性地创建环境相似的程序树。编辑算子在“基因内区”概率保留的前提下，对冗余代码进行简化，可以在合理的计算时间内获得易于理解的结果。

利用对线性表示的个体进行位置信息编码的思想，提出了一种新的基于树的线性后缀形式的 GP 个体程序树表示方法，实现了多种形式的遗传操作，并给出形式化定义，设计并实现了一个基于栈的 GP 算法。

专著中提出了一种 GP 自定义函数共同进化的模型和方法，该模型和方法在解决大规模复杂问题时，如学习分类任务，性能比传统的不带 ADFs 的 GP 和带有 ADFs 的 GP 方法更好。

专著中提出了一种快速、准确的求解 GP 适应度函数权值的新方法，通过对 GP 适应度函数调整参数的选择，快速精确地计算树的权值，克服了传统的方法，如 CV 和 L－曲线方法，不能快速被应用的不足。

专著中提出了一种新的基于 GP 的多目标决策函数模型和稳定性分析方法。在已知可行性方案效用值和权重的情况下，可以设计一种有效的算法来计算稳定性。通过 GP 系统产生的决策函数比通过传统的分层处理(AHP)方法产生的决策函数更稳定。

将 GP 思想和方法应用到基于组件的软件开发中，提出了基于 GP 的组件概念，设计了组件的进化算法。建立了一种分布式环境下基于 GP 组件的软件开发和复用过程的设计思想和方法(简称 GP－CBD 方法)，并系统地分析了组件体系结构，给出了一种规范的组件形式化及语义模型，探讨了组件复用过程应该遵循的自然法则，从而为基于组件的软件复用和编程过程的自动化提供了一种可行的新途径。

本书可供计算机、控制理论与控制工程等专业高年级本科生、研究生使用，也可供其他相关专业科研人员参考。

目 录

第1章 绪 论

1.1 GP概述

自计算机出现以来，计算机科学的一个重要目标就是让计算机自动进行程序设计，即只要明确地告诉计算机要解决的问题，而不需要告诉它如何去做，这是Arthur Samuel早在20世纪50年代作为计算机科学的核心问题而提出的。遗传程序设计(Genetic Programming，简称GP，又称遗传编程)便是在该领域的一种尝试，它借鉴生物界的自然选择和遗传机制，采用遗传算法(Genetic Algorithm，GA)的基本思想，但使用一种更为灵活的表示方式——分层树结构来表示解空间。这些分层树结构的叶结点是问题的原始变量，中间结点则是组合这些原始变量的函数。它们很类似于LISP语言中的S-表达式。这样，每一个分层树原始结构对应问题的一个解，也可以理解为求解该问题的一个计算机程序。遗传程序设计就是采用遗传操作，动态地改变这些结构，以获得解决该问题的可行的计算机程序。

遗传程序设计的思想是Stanford大学的Koza. J. R[1]在20世纪90年代初提出的。他提出了两个论点：

(1)各领域中许多看起来不同的问题都可以看成为寻找一定的计算机程序的问题：给定程序的某个输入则产生所需的输出，即程序归纳问题。

(2)遗传程序设计提供了实现归纳的方法，亦即遗传程序设计可搜索程序空间中特别适合求解(或近似求解)所给问题的程

序。由遗传程序设计产生的程序(结构)是“适应”的结果，正是适应性导致了所需的程序结构。

1.1.1 GP 的基本思想

GP 的基本思想是：随机产生一个适用于所给问题环境的初始群体，即问题的搜索空间；构成群体(population)的个体(individual)都有一个适应度；按照达尔文适者生存的原则，用遗传算子处理高适应度的个体，产生下一代群体；如此循环下去，所给问题的解或近似解将会在某一代出现。

与遗传算法一样，遗传程序设计的遗传算子包括复制、交叉和变异等。

总之，遗传程序设计是通过下面步骤来解决问题的：

(1)确定输入及控制参量

主要包括：①确定函数集和终止符集，根据问题领域特点，合理确定解决问题所需要的函数集合及终止符集合(包括变量和常数)；②确定评价方法，群体中的每个个体是否适应环境，影响着对其进行遗传操作，因此，必须选择某种方式对其适应性进行评价；③确定控制参量，如群体大小、迭代次数、交叉概率等；④程序运行终止准则；⑤最优结果确定及其他参数。

(2)随机产生初始解

依照指定的形成规则随机生成原始群体(初始解)。

(3)重复执行以下步骤，直到满足要求为止。

①评价群体中每一个个体，根据其解决问题的优劣程度，给每一个个体赋予一个适应度。

②生成新一代群体，即首先根据概率准则，选择适应度高的个体，复制到新一代群体中，然后随机选择适应度比较高的两个个体，进行随机交叉，产生新一代个体。另外，根据情况还可以加入一些其他的辅助操作，如变异、封装等。

(4)在所有的迭代结果中，选择最好的结果作为遗传程序设

计的解。

根据遗传程序设计的基本思想，遗传程序设计的一次运行并不能保证能够得到问题的解，如何确定控制参数才使遗传程序设计达到最佳效果，这与所给问题环境有关，通常需经多次反复实验而定，所以说遗传程序设计是一种与实验相关的算法，由于使用计算机，从而使这种实验成为可能。

图 1 -1 是 GP 的流程图，图中 M 表示群体中的个体数，变量 i 表示一个个体，变量 Gen 是当前代的代号。

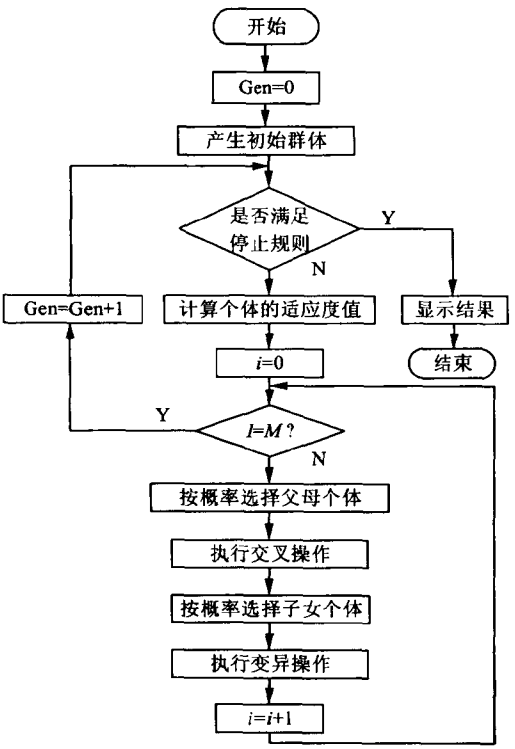

图 1 -1 GP 计算基本流程

1.1.2 GP的主要特点

与传统的处理方法(如机器学习、人工智能)不同，GP注重解的适应性，而不太强调传统求解过程所遵循的原则，这反映了GP的主要特点。

(1)正确性(correctness)：GP处理的对象是不确切的解，它以误差作为进化的驱动，它所得到的解可能按传统的观点是不精确的。

(2)一致性(consistency)：传统方法在求解过程中是保持一致性的，即所得到的解是无矛盾、不冲突的。而GP则同时处理很多不一致、有冲突的解。并且这种不一致性能增加群体的多样性，从而能帮助遗传程序设计算法较快地解决问题。

(3)不推理性(justifiability)：传统的方法根据假设和已知条件以及逻辑规则，通过推理来获取结论。而遗传程序设计则不以逻辑推理为依据，它通过不断尝试和反复实验来求解问题。

(4)确定性(certainty)：传统的方法以确定性的转移规则来求解具有严格数学描述的问题，而GP则以概率性的转移规则为基础。因此，其结果具有不确定性。

(5)秩序性(orderliness)：大多数传统的求解方法与算法不仅是确定的，而且其处理过程与步骤都是紧凑有序和同步的。而GP则是采用一种无序、非对等、独立、分布的异步并行处理方式，而且不受某中心处理过程的控制。

(6)简洁性(parsimony)：简洁性是科学的指导原则。虽然，在遗传程序设计中也可以将减少其结构的复杂性作为其目标之一，但它主要以误差为适应度，作为其进化的驱动力。

(7)决断性(decisiveness)：传统算法大都采用确定的终止准则，此时可在一定程度上判断解所处的位置。而进化是一个连续的过程，因此GP无法定义明确的终止点，它通常需要通过人为

干预终止其进化过程。

GP是在遗传算法的基础上发展起来的全局搜索算法，但它克服了一些遗传算法的缺陷，与遗传算法有一定的区别，主要表现在以下几个方面：

(1)由于遗传算法直接对定长字符串进行操作，所以不能描述层次化的问题。而遗传程序设计个体的树形表达方式则弥补了这一点。

(2)遗传算法定长的字符串描述方法不具备动态可变性，每一种结构仅适用于某类问题的求解。而遗传程序设计的程序结构不再考虑等位基因的位置，这带来了极大的灵活性，具有动态改变大小、形状的能力。

(3)遗传程序设计涉及到的交叉算子更能体现语法结构的合理性，因子树的交换受上下文语义环境的限制。

1.2 GP研究现状及存在问题

1.2.1 GP理论和技术研究概况

1989年，美国斯坦福大学的Koza教授基于自然选择原则，创造性地提出了用层次化的计算机程序来表达问题的遗传程序设计方法，成功地解决了许多问题。1992年，Koza教授发表了他的专著《遗传程序设计：基于自然选择法则的计算机程序设计》[1]。1994年，他又出版了《遗传程序设计》第二册《可重用程序的自动发现》[2]，深化了遗传程序设计的研究。同时，一些相关的有代表性的技术和研究成果也相继出现，使程序设计自动化展现了新的局面。

近年来，随着遗传程序设计研究的不断深入和发展，遗传程序设计运用遗传算法的思想自动生成计算机程序，解决了许多问

题，如预测、分类、符号回归和图像处理。作为一种新技术，它已经与遗传算法并驾齐驱。1996 年举行了第一届遗传程序设计国际会议。该领域已引起越来越多的国内外控制理论与工程、人工智能和计算机等相关学者们的兴趣。目前，国内外对 GP 的研究主要集中在理论、技术及应用等方面。

GP 理论和技术研究主要包括模式定理、算法性能、程序结构、GP 操作、适应度评价、GP 个体表示法和 GP 应用等方面。遗传程序设计的基本算法又涉及了程序结构的形成、GP 操作、适应度评价和 GP 个体表示法等 4 方面技术。

1. GP 理论

虽然 GP 已成功地解决了许多问题，但是它的理论基础还很薄弱，下面简要介绍 GP 理论研究的现状及不足。

(1)可进化性(evolvability)：Altenberg、Michael[3~4]探讨了可进化性的概念，即一个群体具有生成好于任何已存在群体的变体的能力。他利用 Prince 的协方差和选择定理，阐明了适应度测量、表示方法和遗传操作的关系，并分析了进化性与代码在进化程序内出现的扩散现象的联系。这种理论分析指明了寻找改进 GP 系统性能的方法和方向。

(2)适应度图(fitness landscapes)：适应图的概念是由生物学家 Sewell Wright 提出来的，它是反映群体中的个体与其适应度间的映射关系的一种可视图。Kinnear[5]提出几种度量适应度图的方式，探求哪种方式与他所研究的一些问题的难度更具相关性。

(3)GP 模式定理：自从 20 世纪 70 年代 Holland 的模式定理问世以来，人们常用它来解释 GA 的工作原理，认为 GA 解决问题只是层次化地组合相对合适和短的模式来形成问题的解答(构造块假设)。同 GA 一样，GP 也有相应的模式定理。研究 GP 的模式定理，首先要对语法树定义一个模式概念，然后再扩充 Holland 的模式定理。目前有一些关于 GP 的模式定义，它们都是

把模式定义为一个或多个树或树块(Fragment)。根据模式的元素是否与它所在程序的位置有关，把模式分为两类：第1类模式的元素与它所在程序的位置无关，可在同一程序中出现多次；而第2类模式的元素与所在的位置有关，在程序中至多出现一次。

首先介绍第1类模式。Koza[1]第1个解释GP的工作原理，指出Holland的模式定理同样可用于GP。他把GP模式定义为一个用Lisp的S－表达式集表示的模式，进而得到类似GA的模式定理。O'Reilly[6~8]细化了Koza的工作，把模式H定义为一个序偶集，每个序偶指出了一个S－表达式树或树块至少应被个体例化(匹配)的个数。但是他们也承认此模式定理是基于不堪一击的假设下获得的。Whigham[9~11]在基于上下文无关文法的GP系统中，产生了与上下文无关文法的模式概念和有关的模式定理。它与O'Reilly的模式定理的不同之处在于，两者的模式概念不同，O'Reilly的模式可由多个子树表示，而Whigham的模式由单个子树表示。

第2类模式引进了位置信息。Rosca和Ballard[12]提出的模式是一个连续树块，只有叶结点有通配符，模式只在程序的根位置上，因此简称为基于根的树模式。Poli和Langdon[13~14]提出了与GA的模式定义较近似的模式定义，模式是一个树或树块，通配符可在树块中的任何位置，只代表一个函数或终止符，而不是其他定义中所指的子树，这样，模式的采样空间为大小和程序相同的程序子空间，而上述其他模式定义的采样空间是大小和形状不同的程序组成的子空间。Poli和Langdon使用变异和交叉的简单形式，即点变异和点交叉来获得相对复杂的GP模式定理。

总之，从传统意义上来讲，模式定理可很好地用来解释GA算法是怎样进化的。模式定理可被看成是GA算法的宏观模型，这就意味着它可以根据当前代测得的宏观量(如模式适应度、种群适应度、模式中个体数量等)来确定下一代种群的属性。这些

与微观模型形成鲜明对比的宏观量，隐含了大量遗传算法的自由度信息，由它们可以推导出易于理解和研究的等式。而对传统GP模式定理的争论焦点是，它们仅提供了在下一代模式实例数量期望值的下界$E[m(H,t+1)]$，而不是一个精确值，这使得很困难用GP模式定理去预测其未来行为。因此，GP的模式理论值得进一步深入地研究，特别是开展获得精确公式的模式理论方面的研究，以进一步拓展遗传程序设计理论；阐明模式创建的内在机制，突破传统遗传程序设计模式定理只集中在模式的生存与遭破坏的表述上的局限性，并从微观和宏观角度给出具体的定义。目前，这些方面的研究，文献报道较少。

2. GP算法

GP算法的本质是运用GA的思想，通过进化可执行程序来解决问题。GP的主要特点在于它是可变长的、层次化的、常常是树结构的遗传模块，而且大多数情况下，程序个体是可执行的，也就是说常常通过某类解释器解释程序，以获得个体的适应度评价。对于传统GA来说，个体通常是定长的位串或字符串，而且个体的适应度评价由其结构和所要解决的问题来确定。

在GP系统执行基本算法之前，要进行5个准备工作：确定终止符集、函数集、适应度函数、控制参数、终止准则。选择函数和终止符集要满足充分性，即能充分地表达问题。合理选择它们是机器学习中的一个共同问题。适应度评价随问题的不同而异，对一些问题要结合正确性、简洁性、时间等综合因素确定适应度评价。终止符集、函数集以及适应度评价的确定不仅会影响程序的搜索空间、搜索的难易程度，而且还会影响问题的最终解决。近几年，国内外以快速提高优化效率为目标，改善GP算法性能的研究报道有文献[15~20]；但通过GP个体树快速生成、初始群体性能的提高、一致性交叉、编辑算子和权值逐步适应法等简单有效的方法未见有文献报道；特别是如何优化控制参数，