

基于差分进化基因表达式编程的全局函数优化

李太勇^{1,2,3} 唐常杰² 吴江^{1,3} 邱江涛¹

(西南财经大学经济信息工程学院 成都 610074)¹ (四川大学计算机学院 成都 610065)²

(西南财经大学中国支付体系研究中心 成都 610074)³

摘要 为了提高基因表达式编程(Gene Expression Programming, GEP)在函数优化时的效率,将差分进化(Differential Evolution, DE)引入到 GEP 中,提出了基于差分进化的基因表达式编程的全局优化算法 DEGEPO。主要工作包括:(1)针对全局函数优化问题,根据 GEP 和 DE 的特点设计了新的基因编码;(2)设计了新的变异和交叉算子;(3)提出了 DEGEPO 算法并进行了算法分析;(4)实验验证了算法的有效性。相对于传统 GEP, DEGEPO, 优化结果精度平均提高了 2~4 个数量级。

关键词 遗传算法, 基因表达式编程, 差分进化, 函数优化

中图分类号 TP311.13 **文献标识码** A

Global Function Optimization Based on Gene Expression Programming with Differential Evolution

LI Tai-yong^{1,2,3} TANG Chang-jie² WU Jiang^{1,3} QIU Jiang-tao¹

(School of Economic Information Engineering, Southwest University of Finance and Economics, Chengdu 610074, China)¹

(School of Computer Science, Sichuan University, Chengdu 610065, China)²

(The Research Center for China Payment System, Southwestern University of Finance and Economics, Chengdu 610074, China)³

Abstract To improve the efficiency in function optimization via Gene Expression Programming(GEP), Differential Evolution(DE) was introduced into GEP. A novel algorithm called DEGEPO was proposed. The main work of this paper included (1) the gene in GEP was redesigned to adapt global function optimization; (2) novel mutation and crossover operations were applied; (3) a parameter optimization algorithm based on GEP with DE called DEGEPO was proposed and it was also analyzed; (4) experiments demonstrated the efficiency and effectiveness of DEGEPO. Compared with basic GEP, the precision of DEGEPO increased 2~4 orders of magnitude averagely.

Keywords Genetic algorithm(GA), Gene expression programming(GEP), Differential evolution, Function optimization

1 引言

许多科学研究和工程实践中的问题均可归结为最优化问题,用进化算法求解最优化问题是当前的研究热点^[1-3]。函数优化问题是最优化问题中的重要内容。

定义 1(全局函数优化) 设 S 为 R^n 上的有界子集, $f: S \rightarrow R$ 为 n 维实值函数。函数 f 在 S 域上的全局优化就是寻求点 X_{\max} (或 X_{\min}) $\in S$ 使得 $f(X_{\max})$ (或 $f(X_{\min})$) 在 S 上全局最大(或最小),即 $\forall X \in S: f(X_{\max}) \geq f(X)$ (或 $f(X_{\min}) \leq f(X)$)。

全局函数优化分成全局最大化和全局最小化两种方式。在实际应用中,这两种方式可以相互转化。

基因表达式编程(Gene Expression Programming, GEP)是遗传进化算法家族的新成员,广泛应用于函数发现、分类、时间序列分析、组合优化、函数优化等领域。但现有的 GEP

进行函数优化时存在优化精度低、难于收敛等特点^[4]。

本文将差分进化(Differential Evolution, DE)引入到 GEP 中,对全局函数优化问题进行研究,扩充了 GEP 的应用领域,提高了全局函数优化的效率。主要做了下列工作:(1)针对函数优化问题,设计了新的基因编码;(2)设计了新的变异和交叉算子;(3)提出了 DEGEPO 算法并进行了算法分析;(4)实验验证了算法的有效性。相对于传统 GEP, DEGEPO, 优化结果的精度平均提高了 2~4 个数量级。

2 基本概念和术语

2.1 基因表达式编程

2001 年 12 月, Candida Ferreira 在遗传算法的基础上提出了基因表达式编程的概念^[4]。GEP 同传统的遗传算法和遗传编程在一些主要步骤上很相似,但在个体的编码方法及结果的表现形式等方面又有明显的区别。在 GEP 中,个体采

到稿日期:2008-12-24 返修日期:2009-03-07 本文受国家自然科学基金(60773169),国家科技支撑计划重大项目(2006BAI05A01),西南财经大学“211 工程”三期青年教师成长项目(211QN09071)资助。

李太勇(1979—),男,博士研究生,讲师,CCF 学生会员,主要研究方向为数据库与知识工程, E-mail: lityscu@163.com; 唐常杰(1946—),男,教授,博士生导师,主要研究方向为数据库与知识工程; 吴江(1980—),男,博士,讲师,主要研究方向为数据库与知识工程; 邱江涛(1972—),男,博士,讲师,主要研究方向为数据库与知识工程。

用固定长度的线性编码来表示。个体是一条染色体，它是由一个或多个基因线性组成的固定长度的字符串；基因是一个线性的、固定长度的、由函数符集（运算符或初等函数符）和终结符集（变量或常量）组成的字符集合，分为头部和尾部。头部由函数符或者终结符组成，尾部由终结符组成。

Ferreira 提出的个体编码方法可以避免在遗传操作中产生大量的无效结构，大大提高了算法的效率。其编码方法为：将表达式根据其运算法则表示为表达式树(Expression Tree, ET)，再广度优先遍历该表达式树，得到的符号序列即为个体编码的有效部分。关于 GEP 的研究和应用参见文献[4-7]。

例 1 数学表达式 $(x^2 + y^2)/(x - y)$ 对应的 ET 如图 1 所示。对图 1 采取从左到右的广度优先遍历，得到该表达式对应的个体编码序列为 $/+ - * * xyxyxy$ ，该序列被称为表达式 $(x^2 + y^2)/(x - y)$ 的 K 表达式(K-expression)^[1]。同时，将 K 表达式按以上过程的逆过程进行解码，可得对应的数学表达式。

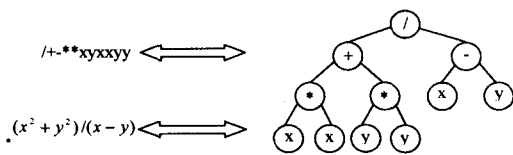


图 1 ET 结构示例

为了保证 K 表达式的合法性，染色体编码被分为头部和尾部两部分。头部可以出现函数符或者终结符，而尾部只能出现终结符。当头部和尾部满足下式，就一定能得到一个有效的染色体

$$t = h(n - 1) + 1 \quad (1)$$

其中， h 为头部长度， t 为尾部长度， n 为函数符集中函数符的参数最大个数，例如函数符“+”带两个参数。

2.2 差分进化

差分进化(Differential Evolution, DE)最初由 Rainer Storn 和 Kenneth Price 于 1995 年提出，它是一种实数编码的进化算法，具有较强的全局搜索能力和收敛速率。在解决复杂的全局优化问题方面，DE 被实践证明是一种有效的全局最优解的搜索算法^[8,9]。DE 的主要特点是算法简单、收敛速度快、所需领域知识少，比较适合解决复杂的优化问题。DE 具有记忆个体最优解和种群内信息共享的特点，即通过种群内个体间的合作与竞争来实现对优化问题的求解。

3 基于差分进化基因表达式编程的全局函数优化

3.1 基因编码

把一个问题的可行解从其解空间转换到算法所能处理的搜索空间的方法称为编码。为了适应函数优化问题，在设计编码时考虑两个因素：编码的结构和参数的可微调性。

定义 2(函数优化基因) 函数优化基因是一个三元组，记为 $g = \{H, T, X\}$ ，其中 H 为基因头部，由数学运算符、初等函数符和自然数组成； T 为基因尾部，由自然数组成； X 为一个常数序列，包含多个常数，采用实数编码。头部和尾部的自然数表示一个常数在常数域中的位置序号。其中，头部元素来自函数符集合 F_s ，尾部元素来自终结符集 T_s 。若无特别说明，下面的“基因”都指“函数优化基因”。一个函数优化基因代表问题空间的一个维度。

定义 3(个体) 假设函数优化问题是一个 N 维优化问题，则第 i 个个体定义成

$$D_i = \{g_{i1}, g_{i2}, \dots, g_{iN}\}$$

个体中基因的多少由问题空间的维度确定。

定义 4(种群) 种群是一个包含 S 个个体构成的集合，定义为

$$G = \{D_1, D_2, \dots, D_S\}$$

例 2 设函数符集为 $F_s = \{+, -, *, /\}$ ，终结符集为 $T_s = \{0, 1, 2, 3, 4\}$ ，常数集合为 $C = \{0.050, 0.985, 0.628, 0.138, 0.196\}$ ，基因头部长度 $h = 3$ ，尾部长度 $t = 4$ 。某函数优化基因编码为

$*/124021$

在该基因中，出现了自然数，构建其基因表达式树时，自然数表示一个常数在常数集合中的位置，如图 2 所示。

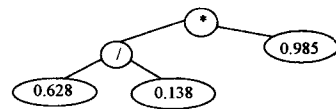


图 2 函数优化基因示例

该表达式树表示的实数为 $(0.628/0.138) * 0.985 = 4.8536231884$ 。可见，通过这种编码方式，常数集合中的几个简单数据可以表达更丰富的实数。在实际应用中，如果产生的实数超过预定的范围，则在计算目标函数值时，丢弃该个体，但是仍然参与下一代种群的进化。

从定义 1 可以看出，函数优化基因与传统基因的基因头部和尾部是一致的，只是在传统基因的基础上添加了常数域。

3.2 遗传算子

在进化过程中，遗传算子主要包括变异算子、交叉算子和选择算子。

3.2.1 变异算子

对于基因头部和尾部，采用传统的变异方法进行变异；根据变异概率决定是否进行变异，变异时基因头部元素能变异成函数符或终结符，而基因尾部只能变异成终结符。

对于基因中的常数域部分，采用差分进化方法进行变异，变异策略为

$$V_i = X_{best} + F(X_{r1} - X_{r2}) + F(X_{r3} - X_{r4}) \quad (2)$$

其中， V_i 是当前种群中的第 i 个个体的常数序列， X_{best} 是当前种群的最优个体的常数序列， $X_{r1}, X_{r2}, X_{r3}, X_{r4}$ 是当前种群中互不相同的 4 个随机个体的常数序列， F 是一个比例缩放因子，一般取值范围为 $(0, 2)$ 。

3.2.2 交叉算子

对于基因头部和尾部，采用传统的交叉算子进行交叉；根据交叉概率决定是否进行交叉操作，交叉时随机产生交叉位置，对参与交叉的两个个体交换交叉位置后的内容。

对于基因中的常数域部分，采用差分进化方法进行交叉，交叉生成实验个体的常数域。交叉策略为

$$D_{Uij} = \begin{cases} V_{ij}, & rand(j) \leq P_c \\ X_{ij}, & otherwise \end{cases} \quad (3)$$

其中， D_{Uij} 为实验个体 D_U 中第 j 个常数分量， $rand(j)$ 为 $[0, 1]$ 之间的随机数， P_c 为交叉概率。

3.2.3 选择算子

对于函数优化问题，选用目标函数作为适应度函数。新

一代个体的选择策略为

$$D_i^{t+1} = \begin{cases} D_{U_i}, & f(D_{U_i}) \geq f(D_i) \\ D_i, & \text{otherwise} \end{cases} \quad (4)$$

其中,函数 f 为适应度函数,即目标函数。式(4)确保了每一代的最优个体保留到下一代。

3.3 差分进化基因表达式编程的全局函数优化算法

综上所述,基于差分进化基因表达式编程的全局函数优化算法 DEGEPO 描述如下。

算法 1 DEGEPO(基于差分进化 GEP 的全局函数优化)

输入:种群大小 M ,最大进化代数 N ,基因头部长度 h ,函数符集 F_s ,终结符集 T_s ,各种算子的概率,寻优空间的上下界 X_{\min}, X_{\max}

输出:最优个体 D_{best}

步骤:

- 1 $i=0$; //进化代数
- 2 生成初始种群 G_0 ;
- 3 计算当前种群中的最优个体 D_{best} ;
- 4 while($i < N$) {
- 5 Mutate(G_i); //对 G_i 进行变异
- 6 CrossOver(G_i); //对 G_i 进行交叉
- 7 Select(G_i); //对 G_i 进行选择
- 8 计算当前种群中的最优个体 D_{best} ;
- 9 $i++$;
- 10 return D_{best} ;

算法在第 2 步生成初始种群。生成初始种群时,头部和尾部按照传统 GEP 方式生成。生成常数域时,按下式生成

$$G_i = \{X_j = X_{\min} + \text{rand}() \cdot (X_{\max} - X_{\min}) \mid 1 \leq j \leq M\} \quad (5)$$

其中, $\text{rand}()$ 为 $[0, 1]$ 之间的随机数,确保生成的实数在 $[X_{\min}, X_{\max}]$ 内; M 为常数域中常数的个数。

在标准 DE 算法中,经过交叉和选择操作的个体的值可能不在 $[X_{\min}, X_{\max}]$ 区间,一般处理方法是将该个体映射到 $[X_{\min}, X_{\max}]$ 的上下边界。而在 DEGEPO 中,丢弃这部分个体。

DEGEPO 从两方面对参数进行优化。一是采用传统 GEP 进化方法,通过 GEP 的良好性能对参数的结构(表达式树)进行优化;二是采用传统 DE,通过 DE 的择优思想对实数值本身进行进化。GEP 和 DE 结合后,可有效实现参数的微调功能。

命题 1 传统 DE 算法是 DEGEPO 的特例。

证明:根据 DEGEPO 的编码规则,当基因的头部第一个元素为终结符时,则该基因只由常数域中的一个常数构成,其余常数不参与运算,此时 DEGEPO 退化传统 DE。所以,传统 DE 算法是 DEGEPO 的特例。

4 实验和性能分析

4.1 实验环境

采用 4 个函数作为算法实验对象。其中 $F_1 - F_2$ 来自文献[4], $F_3 - F_4$ 是函数优化的基准测试函数[10]。

$$F_1(X) = -x_1 \sin(4x_1) - 1.1x_2 \sin(2x_2), 0 \leq x_i \leq 10 \quad (6)$$

$$F_2(X) = 0.5 + \frac{\sin \sqrt{x_1^2 + x_2^2} - 1.25 \cos(x_3 x_4 x_5)}{\sqrt{1 + 0.001(x_3^2 + x_4^2)}}, |x_i| \leq 10 \quad (7)$$

$$F_3(X) = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1 x_2 - 4x_2^2 + 4x_2^4, |x_i| \leq 5 \quad (8)$$

$$F_4(X) = (x_2 - \frac{5.1}{4\pi} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10 - 5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15 \quad (9)$$

F_1 和 F_2 用来评估 DEGEPO 算法进行全局最大化时的性能, F_3 和 F_4 用来评估 DEGEPO 算法进行全局最小化时的性能。

为了评估 DEGEPO 的性能,与传统的基于 GEP 的参数优化方法 GEPO[4] 进行比较。主要参数设置如表 1 所列。

表 1 参数设置

参数名	参数值	
	GEPO	DEGEPO
最大进化代数	10000	10000
种群大小	50	50
函数符集	+ * /	+ * /
终结符集	?	0-9
随机常数范围	[-10, 10]	[-10, 10]
头部长度	6	6
基因长度	20	13
交叉概率	0.6	0.6
变异概率	0.025	0.025
比例缩放因子 F	-	0.8

因为 DEGEPO 和 GEPO 的编码结构不同,所以头部长度相同的基因其基因长度不同。为了对两种方法进行比较,选取了相同的交叉和变异概率。

采用 Matlab 7 进行仿真,系统软硬件环境为 P4 2.0G, 512M 内存, Windows XP。

4.2 实验结果

对 $F_1 - F_4$ 分别采用 GEPO 和 DEGEPO 算法执行 10 次。假设 10 次进化得到的最优目标值的算术平均值为 T' , 理想目标值为 T 。对全局最大化问题,采用 $\lg(T - T')$ 来评价算法的性能;对全局最小化问题,采用 $\lg(T' - T)$ 来评价算法的性能; $\lg(T - T')$ 和 $\lg(T' - T)$ 反映了 T' 和 T 的逼近程度。该值越小,表明 T' 越逼近 T 。各函数的进化曲线如图 3-图 6 所示。

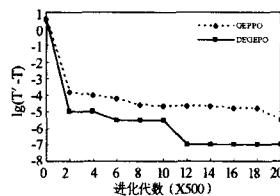


图 3 F_1 函数进化曲线

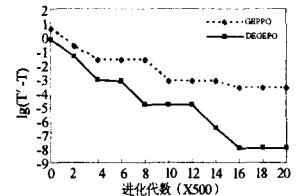


图 4 F_2 函数进化曲线

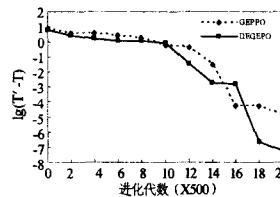


图 5 F_3 函数进化曲线

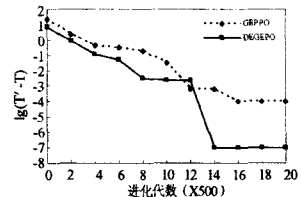


图 6 F_4 函数进化曲线

图 3-图 6 表明, DEGEPO 和 GEPO 都能在一定精度内找到全局最优解。在进化初期, DEGEPO 和 GEPO 的性能大致相当。在精度相同的情况下, DEGEPO 比 GEPO 的平均进化代数要低。在经过指定代数的进化后, DEGEPO 比 GEPO 求解的全局最优解的精度平均要高出 2~4 个数量

(下转第 172 页)

之后得到的过程定义也可以作为子过程,组合后形成新的过程定义。

(3)开发 workflow 模板库中不存在的过程定义。对于一些新系统比较特殊的业务功能,领域 workflow 模板库中如果不存在相应的工作流模板,需要进行开发。

(4)对可复用资产进行维护。可复用资产的积累是一个逐渐演化的过程,具体系统的过程定义对可复用资产提供反馈,根据使用者的反馈可以对领域业务本体及 workflow 模板进行修正,以不断促进可复用资产的完善和充实。

这 4 个阶段是一个迭代的、逐渐精化的过程,如图 2 所示。

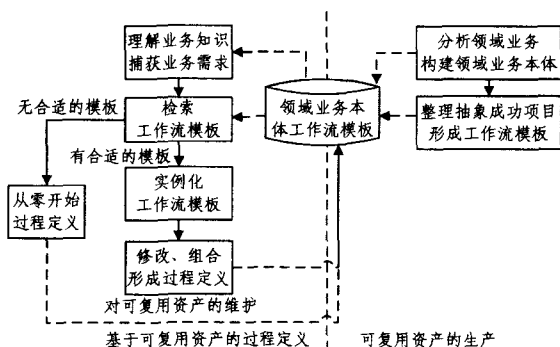


图 2 基于可复用资产的工作流过程定义方法

4 基于复用的过程定义生成工具

为了验证上述方法的可行性和有效性,在扩充 SynchroFlow 过程定义工具的基础上开发了一个基于复用的工作流过程定义工具。SynchroFlow 过程定义工具主要包括:组织机构建模、过程定义、界面设计等功能模块。基于复用的过程定义工具增加了管理和使用特定领域的可复用资产的机制,主要包括:①查看 workflow 模板;②通过浏览或检索选择 workflow 模板;③细化、组合、修改 workflow 模板;④删除 workflow 模板;⑤

添加 workflow 模板;⑥实例化 workflow 模板;⑦使用情况反馈;⑧领域业务本体编辑;⑨用户管理。

结束语 本文在“面向流程管理的软件生产线”课题提出的架构下,研究了特定领域 workflow 过程定义的复用问题,提出了基于领域业务本体和 workflow 模板的工作流过程定义方法。该方法以已有工作为基础,充分利用过去 workflow 系统开发中积累的知识和资产,降低了过程定义的复杂度。workflow 模板等可复用资产是领域中经受过实践检验的解决方案,对它们进行复用能有效地提高过程定义的质量。基于该方法实现的过程定义工具在实际开发中的使用效果也证明了其可行性与有效性。当然,基于复用的过程定义方法依赖于可复用资产的积累与完善,整个生产线的集成环境、工具集以及对可复用资产的可信性度量也是下一步的重要研究内容。

参考文献

- [1] 罗海滨,范玉顺,吴澄. 工作流技术综述[J]. 软件学报,2000,11(7):899-907
- [2] 岳晓丽,杨斌,郝克刚. 信牌驱动式工作流计算模型[J]. 计算机研究与发展,2000,37(12):1513-1519
- [3] 龚晓庆,葛玮,郝克刚. 信牌驱动模型中的工作流模式[J]. 西北大学学报,2004,34(1):13-18
- [4] 杨芙清,王千祥,梅宏,等. 基于复用的软件生产技术[J]. 中国科学(E辑),2001,31(4):363-371
- [5] van der Aalst W M P, Basetn T. Inheritance of workflow: an approach to tackling problems related to change[J]. Theoretical Computer Science, 2002, 270: 125-203
- [6] Studer R, Benjamins V R, Fensel D. Knowledge Engineering, Principles and Method[J]. Data and Knowledge Engineering, 1998, 25(1/2): 161-197
- [7] 朱礼军. 万维网环境下基于领域知识的信息资源管理模式研究[D]. 北京: 中国农业大学, 2004

(上接第 142 页)

级。这主要是因为 DEGEPO 算法不仅对基因头部和尾部进行传统进化,而且对常数域部分通过 DE 算法进行进化,提高了算法的微调性能。

结束语 将 DE 算法引入到 GEP 中,设计了新的基因编码和新的遗传算子。提出了一种新颖的全局函数优化算法,扩充了 GEP 的应用领域。实验验证了算法的有效性。在未来的工作中,将研究此方法在多模态函数优化中的应用。

参考文献

- [1] Francois O. An evolutionay strategy for global minimization and its Markov chain analysis[J]. IEEE Transaction on Evolutionary Computation, 1998, 2(3): 77-90
- [2] 潘长城,徐晨,李国. 解全局优化问题的差分进化策略[J]. 深圳大学学报:理工版,2008,25(2):211-215
- [3] 孟红云,张小华,刘三阳. 用于约束多目标优化问题的双群体差分进化算法[J]. 计算机学报,2008,31(2):228-235

- [4] Ferreira C. Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence [M]. Second, revised and extended edition. Netherland: Springer-Verlag Berlin Heidelberg, 2006
- [5] 黄晓冬,唐常杰,李智,等. 基于基因表达式编程的函数关系发现方法[J]. 计算机科学,2003,30(增刊 A)(10):282-285
- [6] 唐常杰,彭京,张欢,等. 基于基因表达式编程发现知识的三项新技术[J]. 计算机应用,2005,25(9):1978-1981
- [7] Zhou Chi, Xiao Weimin, Tirpak T M, et al. Evolution Accurate and Compact Classification Rules with Gene Expression Programming [J]. IEEE Trans. on Evolutionary Computation, 2003, 7(6): 519-531
- [8] Rainer S, Kenneth P. Differential Evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces[J]. Global Optimization, 1997(11): 341-359
- [9] Price K V. An Introduction to Differential Evolution [J]. New Ideas in Optimization, 1999: 79-108
- [10] 王凌. 智能优化算法及其应用[M]. 北京:清华大学出版社,2001