混合 GP-GA 用于信息系统建模预测的研究

唐丽珏 李 淼 张 建
(中国科学院合肥智能机械研究所,合肥 230031)

摘 要 该文克服了传统建模方法在模型选取及参数估计方面的困难与不足，提出了利用改进的遗传程序设计和改进的遗传算法相结合的混合 GP-GA 算法。一方面,遗传程序设计中加入了简约压力项,控制了代码过度增长,实现了不加先验知识的简洁非线性模型的自动获取。另一方面,遗传算法采用 Gray 编码,随机整群抽样选择,以优化模型中的参数,这在一定程度上补偿了遗传程序设计在演化过程中具有较好结构的模型可能因为其中的参数未能达到最优而被淘汰的损失。仿真实例和实际应用均表明混合 GP-GA 算法优于普通的回归分析及单纯的遗传程序设计方法,提高了拟合和预测精度,并且更适合反映问题的实际情况。

关键词 混合 遗传程序设计 遗传算法 简约压力项

文章编号 1002-8331-(2004)25-0044-05 **文献标识码** A **中图分类号** TP301.6

Study on Modeling and Forecasting of Information System with Hybrid GP-GA

Tang Lijue Li Miao Zhang Jian
(Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031)

Abstract: This paper puts forward hybrid GP-GA algorithm by combining advanced Genetic Programming (GP) and advanced Genetic Algorithm (GA). It overcomes the difficulties of model selection and parameter optimization in traditional modeling methods. On the one hand, parsimony pressure is added to GP, which controls the code bloat. The compact non-linear model can be automatically achieved without any transcendent knowledge. On the other hand, Genetic Algorithm adopts Gray coding and stochastic universal sampling selection and optimizes parameters of the structure GP evolves. It makes up for the loss that results from the models that are washed out because of good structure and bad parameters. The simulation and the application show hybrid GP-GA is superior to simple GP and common regression analysis. The model it evolves is appropriate to reflect real world better.

Keywords: hybrid, Genetic Programming (GP), Genetic Algorithm (GA), parsimony pressure

1 引言

在实际应用领域中存在着由实验或计算机模拟所得的大量数据,存储数据和分析数据成为一项非常重要的工作。将数据归纳为一个抽象公式,用数学模型来定义数据,不仅可以大幅度减少存储量,而且数据模型具有普遍性和规律性,能够记录数据间隐含的系统特征,可以进行数据的预测和分析。很多问题可以归结为如图 1 所示的一个抽象系统。

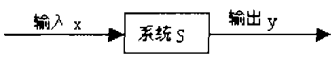

图 1 抽象系统

当系统 S 的结构已知，求最佳输入 x^* 使系统具有最佳输出 y^*,即是通常的一个优化问题;而若是已知有限个输入输出对 $\{(x_i,y_i);i=1,2,\ldots,n\}$,要求模拟系统的行为(即求系统 S 的结构),则是一个建模问题。

传统的优化算法，需要将所解决的问题用数学式子表示，常常要求解该数学函数的一阶或二阶导数，即要求其具有连续、可微、单峰等特性。这样方法的限制性太强,计算公式复杂,准备工作量大。而传统的解决建模问题的方法有数据拟合、回归分析及逼近论等方法。这时问题转化为确定某一给定函数(也称为模型)$f(x,w)$ 中的参数 $w=(w_1,w_2,\ldots,w_m)$ 去拟合、插值或逼近已知数据对。显然,模型 $f(x,w)$ 的选择对这类方法是至关重要的。因为,不适合的模型选择即是具有最优参数也不能很好地用来估计和外推系统的行为，从而使建模过程失去意义。然而,针对某个问题,即是对原问题的背景知识比较了解,如何选择一个合适的模型及进行参数估计也是很困难的。

针对这些问题,利用进化算法的自组织、自学习和自适应的特性,自动从大量的数据中获取描述数据的数学模型,实现信息系统的建模。自然选择消除了算法设计过程中的一个最大障碍:即需要预先定义描述问题的全部特点,并说明针对问题的不同特点算法应采取的措施。该文的主要思想是结合遗传程序设计和遗传算法各自的特性，利用前者演化模型的结构,利用后者优化结构参数,使得模型的生成趋于智能化、自动化。同时在遗传程序设计中加入了简约压力项,使得演化的结构更加简洁,便于优化。遗传算法中采用随机抽样遍历选择策略,减少了优化过程中出现的早熟收敛现象和停滞现象。

基金项目:国家 863 高技术研究发展计划项目:信息技术项目基金(编号:2001AA115180)资助

作者简介:唐丽珏,硕士研究生,研究方向为演化算法,人工智能及其应用。李淼,研究员,硕士生导师,研究方向为专家系统,人工智能及其应用。张建,副研究员,研究方向为专家系统。

2 遗传程序设计和遗传算法的比较

1975年，Holland教授受生物学中“生物进化”和“自然选择”学说的启发，提出了著名的遗传算法(Genetic Algorithm，简称GA)。90年代初，Koza教授在此基础上进一步提出了遗传程序设计(Genetic Programming，简称GP)。该算法试图研究计算科学的一个中心问题：计算机如何在没有明显编程的情况下解决问题。它为上述问题的解决提供了一个可能的工具。这一节将主要介绍它们的异同点。

2.1 相同点

首先，遗传程序设计和遗传算法同属于演化计算的范畴，借助生物演化的思想和原理解决实际问题。基本的流程如算法1：

算法1 演化计算的基本结构

```
{
  随机初始化种群 P(0)={x1,x2,...,xN},t:=0;
  计算 P(0)中个体的适应值;
  while(不满足终止规则)do
  {
  由 P(t)通过遗传操作形成新的种群 P(t+1);
  计算 P(t+1)中个体的适应值,t=t+1;
  }
}
```

由此可以看到，遗传程序设计和遗传算法都是利用演化过程中获得的信息自行组织搜索的，遵循适者生存，不适者淘汰的自然规律。这种自组织、自适应特征同时也赋予了它们具有能根据环境的变化自动发现环境的特性和规律的能力。

其次，在演化过程中可以看到适应度的评价是算法进化的自然驱动力，两者均利用适应性来控制群体中结构改变的过程。适应度高的个体具有较好的生存能力，它具有与环境更适应的基因结构。

再次，遗传程序设计和遗传算法均具有本质并行性[1]。一是内在并行性，即本身非常适合大规模并行演化，通过多个种群的演化和适当地控制种群的相互作用，可以提高求解的速度和质量。二是内含并行性，由于算法采用种群的方式组织搜索，从而它可以同时搜索解空间内的多个区域，并相互交流信息，这种搜索方式使得算法虽然每次只执行与种群规模成比例的计算，而实质上已进行了大约$O(N^3)$次有效搜索。

2.2 不同点

遗传程序设计和遗传算法作为演化计算的四大分支之二，在具体实现方面存在着较大差异。

2.2.1 编码

遗传算法常常采用二进制的0/1字符编码。过程如下：

假设种群中个体数目为n，x_t^i表示第t代的第i个个体，$i\in\{1,2,\dots,n\}$。每个个体有m个参数组成，每个参数用l位二进制表示。这样每个个体$x_t^i\in\{IB\}^{ml}$，$IB\in\{0,1\}$，每个个体基因位数目$L=ml$。个体x_t^i可以表示为ml维的行向量，即$x_t^i=[x_t^{i(1)}\dots x_t^{i(l)}\ x_t^{i(l+1)}\dots x_t^{i(2l)}\dots x_t^{i((m-1)l+1)}\dots x_t^{i(ml)}]$。第$t$代种群$X_t$可以表示为一个$n\times ml$的矩阵$X_t=[x_t^1,x_t^2,\dots,x_t^n]^T$。个体$x_t^i$的第$k$个长度为$l$的二进制码串转化为实数的解码函数$\Gamma$为：

$$\Gamma(x_t^i,k)=u_k+\frac{v_k-u_k}{2^l-1}\left(\sum_{j=1}^{l}x_t^{i(kl+j)}\times 2^{j-1}\right)\qquad(1)$$

式中v_k和u_k分别为第k个实数范围的上限和下限。

而遗传程序设计组成群体的个体是动态的树状结构。树的结点由终止符、原始函数与运算符组成。其中终止符结点也称为叶结点，它是将问题分级划分为子问题后最基本的解的成分。这种树状的层与结点是可变化的。终止符结点是问题的原始变量，根结点和中间结点统称为内部结点，它们则是组合这些原始变量的函数，类似于LISP语言中的S表达式。这样，每个分层结构对应问题的一个可能解，也可以理解为求解该问题的一个计算机程序。

两者不同的编码方式使得它们的应用范围有所不同。遗传算法不能描述层次化的问题，不能描述计算机程序，而且缺少动态可变性，因此它的主要领域是复杂的非线性优化问题。而遗传程序设计在解决优化控制、符号回归、规划、寻求博奕策略、解微分方程以及模式识别等领域的问题中效果尤为卓著。

2.2.2 初始种群的产生

遗传算法的初始种群由若干具有定长字符串的个体组成。产生初始种群的方法通常有两种。一种是完全随机的方法产生的，它适合于对问题的解无先验知识的情况。另一种可将某些先验知识转变为必须满足的一组要求，然后在满足这些要求的解中再随机地选取样本，这样选择初始种群可使遗传算法更快地达到最优解。

而遗传程序设计的初始群体由随机生成的个体树组成。个体树又由符合问题范围的函数和变量组成。初始群体的生成也可采用不同的方法，常用的有完全生成法、生长法、倾斜生成法和倾斜对半生成法[3]。

2.2.3 遗传操作

在演化过程中，基本的遗传操作有复制、交叉和变异三种。

复制：挑选现有的个体将其复制到新的进化代种群中，适应值越高的个体选中概率越大。遗传算法复制的是字符串，而遗传程序设计复制的是个体树。

交叉：遗传算法的交叉操作也称为基因重组，把两个父个体的部分字符串结构加以替换重组而生成新个体的操作；而遗传程序设计的交叉操作是从当前群体中，根据适应度值挑选两个父个体，父个体的不同部件(如子树、子程序或子公式等)重新组合产生两个子个体，如图2所示。

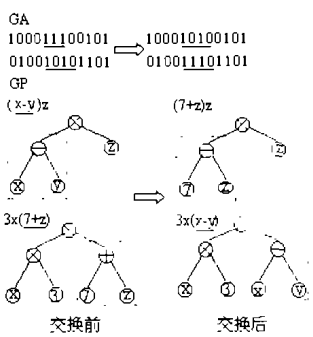

图2 交叉操作

变异：对于二进制编码的遗传算法个体而言，变异意味着基因位的翻转；在遗传程序设计中，由程序随机产生一棵新的子树，以代替被突变概率选中结点以下的原有子树部分，如图3所示。

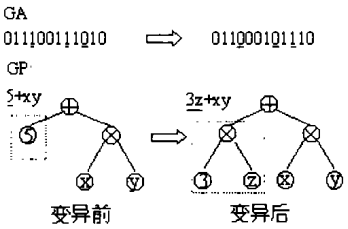

图 3　变异操作

3　混合 GP-GA

从第 2 节中不难发现 GA 主要用于搜索数值解，而 GP 着眼于寻求问题解的个体树。在以前的研究中，单纯地使用遗传程序设计实现信息系统的自动建模，虽然比起传统的方法具有较大的灵活性和智能性，但在演化过程中，具有较好结构的模型可能因为其中的参数未能达到最优而被淘汰，使最终搜索到的模型精度布告。GP 擅长于模型结构的自动搜索，而不是参数的寻优。而遗传算法主要应用于非线性参数的优化，两相互补，笔者尝试将遗传程序设计和遗传算法结合起来对信息系统建模。与文献[4,5,6]不同的是，对于遗传程序设计和遗传算法的使用采取串行方式，设计算法如下：

算法 2 混合 GP-GA

{

设置训练集和测试集；

/////自适应建模部分，其中 R1 是 GP 的运行次数

for(run=0;run< R1;run++)

{

随机产生初始种群 $P(0)$，t=0；

计算 $P(0)$中个体的适应值，加入简约压力项；

while(个体不满足终止规则)

{

由 $P(t)$根据遗传操作生成新的种群 $P(t+1)$；

按以上方法计算 $P(t+1)$的适应值，$t=t+1$；

}

记录该次运行的最佳个体；

}

从 R1 次运行所得的最优个体中选取一个结构简单的个体 p 作为参数待优化的对象；

/////参数优化部分，其中 R2 是 GA 的运行次数

分离个体 p 所包含的参数 $c_0,c_1,\ldots,c_N$；

参数Gray 编码；

for(run=0;run< R2;run++)

{

产生参数优化的初始种群 $Q(0)$，t=0；

将 $Q(0)$中的每个参数序列回代入结构，计算适应度；

while(个体不满足终止规则)

{

由 $Q(0)$根据随机遍历抽样法执行遗传操作，生成新的种群 $Q(t+1)$；

回代入结构，计算适应值，$t=t+1$；

}

记录该次运行的最佳个体，并计算其拟合误差；

}

从 R2 次运行中挑选拟合误差最小的参数序列，作为结构参数优化的结果；

/////生成最终问题解

将结构和参数合并，作为信息系统的最终模型；

}

在该算法中，有三处对标准 GP 和标准 GA 作了改进，作如下说明：

第一，GP 适应度衡量中简约压力项的加入。

在 GP 的演化过程中存在“规模爆炸问题”，即程序个体所表示的语法树，不管深度还是广度都会不断地膨胀，与此同时解的质量却没有任何改进。若对此情况不加以控制，则只会使进化过程变得越来越慢，影响效率，而且也使得对个体的解释变得很困难。

除了在算法中设定了个体规模，即限定了每个个体所含的结点数量，以及限定了个体中允许的树深，而且还加入了简约压力项。在适应值函数中加入简约因子，通过改进适应度衡量，防止过早收敛和停滞现象，更好地驱动自然选择。简约压力项使用基于大小的适应值惩罚量来阻止代码增长。对于个体 i，带有简约压力项的适应值函数表示如下：

$$f(i)=O(i)+\alpha S_i \tag{2}$$

其中 $O(i)$是个体 i 的原始适应度，S_i 给出了 i 的树状表示的复杂度，算法中指结点个数，α 是简约系数。适应值函数包含了适应精度和个体树的大小，既要求最小适应误差，也要求模型的最简单结构，以方便遗传算法进一步优化结构分离所得的参数。

第二，遗传算法中参数编码采用 Gray 码。

Gray 编码$(g_{l-1}g_{l-2}\ldots g_0)$与普通二进制编码$(b_{l-1}b_{l-2}\ldots b_0)$的关系，可以用下式表示：

$$g_k=\begin{cases} b_{l-1} & k=l-1 \\ b_{k+1}\oplus b_k & k\le l-2 \end{cases} \tag{3}$$

上式中⊕为模 2 加法。Gray 编码与普通二进制的逆关系可以表示为：

$$b_k=\sum_{i=k}^{l-1} g_i \pmod 2,\ k=0,1,\ldots,l-1 \tag{4}$$

相邻整数的二进制码可能具有较大的 Hamming 距离，如 7 和 8，二进制编码为 0111 和 1000，算法在这两个数之间转换需改变所有的位，这种缺陷也称为 Hamming 悬崖，将降低遗传算子的收敛效率，而 Gray 码的相邻整数之间只有一位不同，可以克服普通二进制的这个缺陷。

第三，遗传算法中采用随机遍历抽样选择[3]。

设定 n 为需要选择的个体数目，等距离选择个体，选择指针的距离为 $1/n$，第一个指针的为止由$[0,1/n]$区间的均匀随机参数而定。它提供了零偏差和最小个体扩展，有利于遗传算法的收敛。

4　计算实例

研究进行了两种类型的实验，一种是具有已知函数形式的仿真实验，另一种是将混合 GP-GA 应用于预测问题的求解。

4.1　仿真实例

研究的多项式函数含有两个变量，形式如下：

$$F(x,y)=1.3x^2+2xy-3y^2-1 \quad x,y\in[-2,2] \tag{5}$$

随机生成 200 对样本值，分成两数集 Group1 和 Group2，目的是要根据训练集拟合该函数，同时确定多项式的形式和系数，并且用测试集测试。实验分成两组，Group1 作训练集，Group2 作测试集；或反之。

GP运行的主要参数如表1,其中R是区间[-2,2]内的随机整数,适应值函数中α=0.01。

表1 GP的参数设置

运行参数	详细描述
函数集	+,-,*
终止符集	x,y,R∈[-2,2]
适应值函数	(2)
最大演化代数	101
种群规模	1000
初始种群产生方法	ramped half-and-half
最大初始树深	6
交叉后最大树深	17
内结点交叉概率	90%
任意点交叉概率	10%
变异概率	5%
选择方法	赌轮选择,规模为7
得分数	目标值与预测值之间的差值小于0.05,得1分。
终止规则	GP运行101代或有个体得分满100。

独立运行10次。值得说明的是,本次实验中函数集只包含+、-、*,而不包含sin、cos、exp等复杂运算符,这对于发现简单函数形式是比较适合的。

将10次运行的最好个体分别简化,选择其中一个结构简单的个体作为进一步参数优化的对象,它拟合了训练集Group1中50%的样本,以及测试集Group2中48%的个体,并不是符合要求的最理想个体,形式如下:

$$F(x,y)=1.33965x^2+2xy-3y^2-1.06768+0.02241x-0.00402y \quad (6)$$

将它与(5)式比较,多了两项0.02241x和-0.00402y,因为系数均接近于0,所以比起其他同类型的最好个体对整个值的影响较小。根据混合GP-GA,对该式进行参数分离,如下:

$$F(x,y)=c_0x^2+c_1xy+c_2y^2+c_3+c_4x+c_5y \quad (7)$$

共有6个参数:$c_0,c_1,\ldots c_5$。

实验的参数优化部分分为两组:种群规模为20和30,每组独立运行10次。遗传算法的控制参数为:交叉概率P_c=0.6;变异概率P_m=0.02;窗口大小为5;计算次数为4000。其中计算次数指算法在演化过程中从初始化到某演化代止时共计算目标函数值的次数,可用$(T+1)N$表示,T为进化代,N为种群规模。演化得到的一个最好个体为:

$$F(x,y)=1.30x^2+2.00xy-3.00y^2-0.99 \quad (8)$$

几乎与原函数(5)相同,在允许误差范围内能够正确拟合训练集和预测测试集中的所有样本。

表2,表3列出了单纯GP与混合GP-GA的最优个体拟合率和预测率的均值,括号中的值表示遗传算法的不同群体规模。表2中Group1为训练集,Group2为测试集;表3中反之。

表2 比较(Group1- Group2)

	拟合率(%)	预测率(%)
GP	65.8	64.2
GP-GA(20)	93	90.2
GP-GA(30)	83.9	76.5

表3 比较(Group2-Group1)

	拟合率(%)	预测率(%)
GP	58	55.2
GP-GA(20)	88.2	90.2
GP-GA(30)	57.2	58.1

从表中可以看出,混合GP-GA比单纯的遗传程序设计明显提高了算法的拟合率和预测率,而且混合GP-GA中遗传算法采用适当的小种群效果更好。

4.2 预测应用

数据来源于文献[7],见表3的实际数据一栏。目的是要根据甘肃省天水市1979-1991年的小麦条锈病发病因素建立合适的数学模型并预测1996年的流行等级。x表示感病品种种植面积占小麦播种面积的百分比,y表示去年秋苗全市普遍率,z表示冬季积雪天数,T表示实际流行等级。

参数设置基本同实例1。不同的是自适应建模过程中遗传程序设计的函数集为{+,-,*,/,sin,cos,exp},终止符集为{x,y,z,R},种群规模增大为5000,目标值与实际值的差值小于0.1得1分,α取0.05;参数优化过程中有23个参数参与优化,群体规模为20,计算次数为8000,根据种群大小及染色体长度越大,变异率选取越小,因此变异率降低为0.002。

传统方法是用线性回归来建模的,借助Matlab工具求解,比手工求解的精度要大些,线性回归模型为:

$$T=0.08x+0.252y+0.042z-3.0737 \quad (9)$$

单纯遗传程序设计演化的较好模型为:

$$T=\sin(\sin(0.04711x+2.83330)*\cos(\cos(0.0846Z)-\sin x-\sin(\sin x)))+0.04711x-\cos(0.0846z+0.38854y) \quad (10)$$

混合GP-GA演化的较好模型为:

$$T=0.08x+0.19\cos(-1.56z-0.52y-0.11)-0.74\sin(2.16\sin(0.87x-2.19)\cos(0.44\cos(-1.37z-1.54)+2.22\sin(-1.12x-1.59)-1.41\sin(1.37\sin(-1.60x+1.79)+1.51)+0.15)-0.34)-2.18 \quad (11)$$

表4列出了各项的比较结果,模型I,模型II,模型III分别表示线性回归,单纯GP和混合GP-GA的模型。

表4 实例2(线性回归,单纯GP,混合GP-GA的结果比较)

	实际数据					模型I	模型II	模型III
序号	年份	x	y	z	T	$f(x,y,z)$	$f(x,y,z)$	$f(x,y,z)$
1	1979	61.0	0.405	12	3	2.4124	2.8471	2.9342
2	1980	64.8	0.397	16	3	2.8823	2.9638	2.9342
3	1981	50.8	0.002	10	1	1.4108	0.9979	0.9552
4	1982	52.9	0.317	18	2	1.9942	2.0829	2.0061
5	1983	61.2	0.111	18	3	2.6063	3.0442	3.0099
6	1984	76.0	0.521	22	4	4.0616	4.0770	3.9258
7	1985	80.4	0.887	39	5	5.2198	4.8954	5.0203
8	1986	50.9	0.391	6	1	1.3488	0.8662	0.9679
9	1987	50.4	0.082	10	1	1.3990	1.0099	1.0696
10	1988	41.2	0.081	9	1	0.6207	1.0897	0.9869
11	1989	60.0	0.900	25	3	3.0031	3.0568	3.0714
12	1990	80.0	1.910	27	5	4.9416	4.8820	4.8946
13	1991	70.0	0.750	9	3	3.0933	2.6918	3.0253
14	1996	80.0	1.020	36	5	5.0953	4.9326	4.9152
	得分数					5	8	12
	拟合误差					1.0826	0.4343	0.2070
	最大绝对误差 $\max_{ae}$					0.5876	0.3082	0.1054
	最小绝对误差 $\min_{ae}$					0.0031	0.0021	0.0061
	最大相对误差 $\max_{re}$					0.4108	0.1338	0.0696
	最小相对误差 $\min_{re}$					0.0010	0.0021	0.0030
	预测误差					0.0954	0.0674	0.0848

其中,拟合误差=$\sqrt{\sum_{i=1}^{13}[f(x_i,y_i,z_i)-T_i]^2}$;

预测误差=$|f(x_{14},y_{14},z_{14})-T_{14}|$;

$\max_{ae}=\max_{1\le i\le 13}|f(x_i,y_i,z_i)-T_i|$;

$\min_{ae}=\min_{1\le i\le 13}|f(x_i,y_i,z_i)-T_i|$;

$$\max_{re}=\max_{1\le i\le 13}\left|\frac{f(x_i,y_i,z_i)-T_i}{T_i}\right|;$$

$$\min_{re}=\min_{1\le i\le 13}\left|\frac{f(x_i,y_i,z_i)-T_i}{T_i}\right|;$$

由表4可见,模型II、III的性能均优于模型I,而且都是自动获得的。由于实际预测的每个流行等级之间只相差1,则当绝对误差大于0.5的时候会引起错误估计的情况,如模型I的1979年,这会给生产带来一定损失,模型II、III相对来说出现错判的几率会小得多。比较模型II和模型III,后者具有更高的得分数,更小的拟合误差,预测误差也在允许范围之内,比前者稳定性更高,可靠性更好。模型III是该实例的最佳模型。

采用混合GP-GA演化的预测模型不仅具有良好的结构,而且具有更优的参数,使得拟合和预测效果更好。

5 结束语

针对采用传统的解决信息系统建模问题的方法中存在的困难与不足,该文利用遗传程序设计的建模过程可以无监督、自适应进行的特性,结合遗传算法在非线性参数优化方面的良好性能,提出了混合GP-GA算法,在一定程度上弥补了单纯遗传程序设计中具有较好结构的模型可能因为其中的参数未能达到最优而遭淘汰的损失。混合GP-GA在计算实例中与单纯GP、传统建模方法的比较均体现了其建模过程自动化与智能化,拟合精度更高,稳定性更好等优点,并且对于预测问题的应用能够最贴近地反映客观事实,非常适合用于信息系统的建模预测。这种策略应用于其他问题领域的研究有待于进一步探讨。(收稿日期:2004年2月)

参考文献

1.潘正君,康立山.演化计算[M].北京:清华大学出版社,1998

2.王小平,曹立明.遗传算法——理论、应用与软件实现[M].西安:西安交通大学出版社,2002

3.J R Koza.Genetic Programming I[M].MIT Press,Cambridge:MA,1992

4.Les M Howard,Donna J D'Angelo.The GA-P:a genetic algorithm and genetic programming hybrid[J].IEEE Expert,1995;10(3):11~15

5.曹宏庆,康立山,陈毓屏.动态系统的演化建模[J].计算机研究与发展,1999;36(2):923~931

6.傅传芳.基于GP/GA的数据建模方法[J].昌潍师专学报,2000;19(5):50~54

7.李淼.甘肃技术报告[R].合肥,2000

(上接43页)

软件工程的原则,也难以实现。依据提供原子计算的方式和消息格式,笔者设计了多种标准定向件,如SQL定向件、RPC定向件、CORBA定向件、EJB定向件、WEB SERVER定向件等。定向件以动态调用形式和计算源交互。定向控制件以统一的请求格式调用定向件,由定向件转换成对应的格式向具体的计算源请求。所有定向件由定向控制件管理、协调,以可插拔式、动态配置的架构实现,如果有非标准的计算源,可以设计新的定向件和其互动。定向控制件实施定向计算的调度,如前所述,其行为依据计算源配置件和原子计算描述件信息。如果一个原子计算有多个计算源,定向控制件可根据可靠性、准确度等按某种优化策略择一调度;也可根据网络情况、计算源的响应度进行调度,以实现网络负载平衡;还可并发调度再综合得最优结果。原子计算结果在向汇聚件返回的同时存储于历史件中。在有重复的原子计算调用的情况下,如果计算结果是无状态的,或者虽然是有状态的但历史件中的结果还未失效,定向控制件可直接从历史件中取出而不必向计算源请求,提高效率和节省网络传输。历史件的存储刷新可依某种策略进行,如LRU等。

汇聚件是整个结构的核心。汇聚件对复合计算的组合逻辑进行解释,通过定向控制件取得原子计算结果,实现原子计算对客户端的透明性。汇聚件还要具备条件判断、简单运算能力。汇聚件、复合计算表示件、注册部件一起构成一个仓库式体系结构:汇聚件是一个简单的解释机实现,复合计算表示件是知识仓库,知识就是组合逻辑、容错规则等,注册部件则是知识的输入。

系统采用XML描述复合计算、原子计算、计算源是基于以下几点考虑。

(1)XML是一种可自定义标记的语言,它的描述能力具有较大的丰富性和可扩展能力,这正是描述框架所注重的。

(2)一个XML模式可以映射成一个知识库。那么,一个有效的XML数据就可以看成是知识库的一个模型[7]。

(3)XML模式约束了XML树结构形状,描述逻辑能够很好地表示对象之间的结构关系。

4 总结

网络具有巨大的计算能力,这些计算是异构的、动态的但又是可组织的、可集成的。该文基于组合网格计算能力为新的应用的思想设计了一种计算网格的软件体系结构:CGSA,包括CGSA的部件构成及架构,原子计算、复合计算、计算源的XML描述,系统的实现考虑。提出了"复合计算"的概念并分析了其组合逻辑的形态。通过清晰地定义复合计算的组合逻辑并显式描述,CGSA解决了网格计算能力的动态集成,屏蔽了各计算源间的异构性,具有计算透明和完备可靠、动态可配置、可扩展、松耦合、计算源负载平衡等特点。CGSA易于实现软件的协同性、自适应性,以及需求的多目标性,实现了系统演变的静态性到系统演化的动态性的转变。基于CGSA结构,可以充分利用网格计算能力,快速构造新的应用,同时也为集成企业旧的业务系统提供了新的方法。(收稿日期:2004年4月)

参考文献

1.I Foster,C Kesselman.The Grid:Blueprint for a New Computing Infrastructure[M].San Francisco:CA,Morgan Kaufmann Publishers,1998

2.L Smarr,C Catlett.Metacomputing[J].Communications of the ACM,1992;35(6):44~52

3.I Foster,C Kesselman,S Tuecke.The anatomy of the grid:Enabling scalable virtual organizations[J].International Journal of Supercomputer Applications,2001;15(3):200~222

4.OGSA规范[S].http://www.gridforum.org/ogsi-wg/drafts/GS_Spec_draft03_2002-07-17.pdf

5.CAO Long-bing,DAI Ru-wei.Software Architecture of the Hall for Workshop of Metasynthetic Engineering[J].Journal of Software,2002;13(8)

6.Org Inc.Web Servers Description Language(WSDL)[M].Version 1.2,http://www.w3.org/tr/wsdl2

7.Dongwon Lee,Murali Mani,Makoto Murate.Reasoning About XML Schema Languages Using Formal Language Theory[R].Technical Report,Rj#10197,Log #95071,Ibm Almaden Research Center,2000-11-16