第 26卷第 2期 计 算 机 应 用 研 究 Vol 26 No. 2
2009年 2月 Application Research of Computers Feb. 2009

基于改进遗传规划算法的数据拟合 *

邵桂芳[1]，周绮凤[1]，陈桂强[2]
(1.厦门大学 模式识别与智能系统研究所，福建 厦门 361005；2.重庆通信学院 自动化学院，重庆 400035)

摘 要：针对传统数据拟合方法需预先估计基函数、依赖于应用领域等问题，基于遗传规划的动态可变特性，提出将遗传规划与最小二乘法结合，设计具有一定通用性和自适应能力的数据拟合算法。在分析传统遗传规划算法的基础上，详细介绍了算法改进方法，并针对各种类型的拟合数据进行了对比实验。实验结果表明，该算法不仅可以应用到多种场合，而且可以提高拟合效率与精度。

关键词：遗传规划；改进；数据拟合；最小二乘法
中图分类号：TP301 **文献标志码**：A **文章编号**：1001-3695(2009)02-0481-04

Data fitting based on improved genetic programming

SHAO Gui-fang[1], ZHOU Qi-feng[1], CHEN Gui-qiang[2]
(1. *Institute of Pattern Recognition & Intelligent System, Xiamen University, Xiamen Fujian* 361005, *China*; 2. *Dept of Automation, Chongqing Communication Institute Quarterly, Chongqing* 400035, *China*)

Abstract: There are many problems in current data fitting methods, such as it needs to estimate the radical function in advance and depends on the application field, and so on. Based on the dynamic alterable property of genetic programming (GP), combined GP with least square method, and designed a new data fitting method which had universal and self-adaptive capacity. Firstly, analyzed the traditional GP. Secondly, introduced the improved method in details. Finally, finished some contrastive experiments based on various fitting data. Experiment results show that this method can be applied in many fields, and it can improve the fitting efficiency and precision.

Key words: genetic programming; improve; data fitting; least square method

在科学实验中，经常涉及大量数据，通常采用数据拟合方法来探索这些数据隐含的内在规律。数据拟合在很多领域都有应用，如工程数据分析、图像数据分析等，拟合方法也有很多，如最小二乘法、支持向量机、样条函数方法等[1]。传统的曲线拟合，如最小二乘法，先要根据专业知识，从理论上推导，或者根据以往的经验，确定变量之间的函数关系，从而预先确定方程的结构形式(线性形式、对数形式或多项式等)，然后再进行参数估计。但实际中很难准确判定方程的结构形式，尤其是数据无明显规律或数据量比较大的情况，很难估计数据间的关系，因此传统的曲线拟合均存在一定的局限性。

近年来，进化计算、自然计算等很多模仿自然生物进化过程的算法应运而生，并在很多领域取得了较好的应用效果。遗传规划早在 1992年就由 Koza教授提出，并应用于很多领域[2]，但国内对遗传规划的研究非常少，其应用也仅限于机器人的路径规划方面。

遗传规划遵从生物进化的优胜劣汰，适者生存原则，具有动态可变性、能够描述层次化问题等优点。目前，也有人利用遗传规划进行数据拟合[3~5]，但都直接利用传统算法进行计算，很难保证一次搜索到最优结果，而且针对不同的应用，需要依据经验重新设置算法参数。

针对传统数据拟合算法的不足，如需要预先设计基函数，不同应用需设计不同的基函数等，基于遗传规划在其他领域的成功应用[6]，将遗传规划引入到数据拟合领域，并进行算法改进，结合最小二乘法，提出了一种具有一定自适应能力的数据拟合算法。

传统遗传规划算法

遗传规划是从一组随机生成的初始可行解开始，通过选择、交叉和突变等遗传操作，逐步迭代而逼近问题的最优解。传统遗传规划算法操作步骤如下：

a)个体表示。传统遗传规划算法个体表示有三种方法，即树状结构、线状结构和网状结构。其中，树状结构操作简单方便，使用最为广泛。

b)种群生成。遗传规划初始个体的生成方法主要有三种，即完全法、生长法和混合法。只用完全法形成的初始群体，个体间在结构上很相似，而只用生长法形成的初始群体，个体间在结构上的差异可能会很大。为了增加群体的多样性，同时又保证结构的合理性，可将两种方法结合起来。

c)交叉。遗传规划包括三种交叉算子，即子树交叉、模块交叉和自身交叉。通常采用子树交叉方式来产生新个体，即随机从父代中选择出一些个体，选择其中最好的个体进行交叉操作，如果选择出的两个个体适应度相同，则选择含有节点少的个体。

d)变异。遗传规划主要有子树突变和点突变两种变异算子。点突变只针对节点进行替换操作，如选择的节点是函数，

收稿日期：2008-05-20；**修回日期**：2008-08-03 **基金项目**：国家自然科学基金资助项目(60443004)；重庆市自然科学基金资助项目(2007BB2415)

作者简介：邵桂芳(1978-)，女，黑龙江阿城人，讲师，博士，主要研究方向为模式识别与图像处理、人工智能与机器人等(gfshao@xmu.edu.cn)；周绮凤(1976-)，女，讲师，博士，主要研究方向为模式识别与智能系统；陈桂强(1980-)，男，讲师，硕士，主要研究方向为智能控制与模式识别.

则随机从函数集中选择一个代替;子树突变是对所选择节点以下的整个子数用随机生成的新子树代替。

e)新一代个体。传统遗传规划算法通常采用误差绝对值作为适应度函数来评价个体的优劣,并分别从父代和通过交叉变异产生的子代中选择出一部分个体构成新种群,再重复 c)和 d)的遗传操作,直到达到最大进化代数。

由于遗传规划是一种随机性很强的全局搜索优化算法,是否能收敛到全局最优解与初始种群的质量、参数选择、遗传操作及适应度值的计算方法等有很大关系。因此,有必要对其进行改进,以提高其收敛性能,使其更适合实际应用。

算法改进

. 参数设置

1)个体表示 树状结构在程序中可以采用数组和类等方式灵活存储,简单方便,因此采用树状结构表示个体。遗传规划用终止符集抽象表示问题的输入,用函数集模拟对输入的处理,针对数据拟合问题,可以设计终止符集为 $T = \{x, \alpha\}$,函数集 $F = \{+, -, \times, /, \sin, \cos, \log, \cdots\}$。函数集越详尽,产生的个体越复杂,因此,函数集到底应该包含多少运算符要依据实际应用情况进行调整,以提高算法效率。

2)种群产生 为了保证个体结构合理以及种群具有多样性,采用混合法产生初始种群。同时,为了防止个体在进化过程中无限制地增大,限制了个体的节点数目最大为 25,并按式(1)选择节点:

$$\text{node} = \begin{cases} f & \text{random\_num} > (\text{flag} * \text{flag} + 3) / (\text{size} - i + 1) \\ t & \text{else} \end{cases} \tag{1}$$

其中:size为限制个体的最大节点数;flag为条件标志,初始为 1,每从函数集中选择一次,自动加 1,从终止符集中选择一次,则自动减 1;i为产生节点数;random_num为产生的随机数。

3)适应度函数 数据拟合最终的目的就是使拟合出来的函数能最大限度地包含原始数据,能充分展现出原始数据的内在规律。在未知数据内在规律的情况下,利用遗传规划可以产生不同结构的基函数。产生基函数后,如何评价这些基函数。遗传规划主要通过适应度函数来评价每个个体,鉴于最小二乘法在确定拟合函数结构后,具有较好的效果。因此,本文利用最小二乘法来设计适应度函数:

$$f_i = \sum_{j=0}^{M} |p(x_j) - y_j|^2 \quad (i = 1, 2, \cdots, N) \tag{2}$$

其中:p为遗传规划产生的个体,即拟合函数。

4)终止条件 为了提高算法效率,保证每次运行能找出较优的结果,设置了如式(3)所示的终止条件,三个条件只要有一个满足就结束程序。

$$p \begin{cases} f_i < 0.001 \\ \text{hits} \geq 90\% \\ \text{gen} = \text{max\_gen} \end{cases} \tag{3}$$

其中:hits表示拟合匹配程度;max_gen为进化代数。

. 轮盘赌选择进行交叉

遗传规划的交叉操作主要有子树交叉、自身交叉和模块交叉等三种方法。本文选择了子树交叉法来实现交叉操作,为了保证种群的多样性,引入了轮盘赌来选择进行交叉的个体,具体操作流程如下:

假设,种群中有 N 个染色体 $c_1, c_2, \cdots, c_N$,其适应度函数值分别为 $f_1, f_2, \cdots, f_N$,优化问题为适应度函数值越小越好。

a)归一化处理整个种群中染色体的适应度值

$$P(c_0) = 0;\ P(c_i) = P(c_{i-1}) + (1/f_i)/\xi;\ i = 1, 2, \cdots, N \tag{4}$$

其中 $\xi = \sum_{i=1}^{N} 1/f_i$, $f_i \neq 0$。如果 $f_i = 0$,可以令 $f'_i = f_i + \zeta$,重新计算式(3)。ζ为常数,$f'_i \neq 0$。

对于适应度函数值越大越好的情况,可以将式(4)改为

$$P(c_1) = 0;\ P(c_i) = P(c_{i-1}) + f_i/\xi\ \ i = 1, 2, \cdots, N \tag{5}$$

其中 $\xi = \sum_{i=1}^{N} f_i$。

b)随机产生一个数 β, $\beta \in [0, 1]$。如果 $P(c_{i-1}) < \beta < P(c_i)$,那么选择第 i个染色体进行交叉操作。

c)重复步骤 b)选择另一个染色体 c_j, $j \neq i$。

d)如果所选择的两个个体相同或适应度值相等,则重复 b),直到选择出两个不同的个体。

e)在选择出的两个染色体中用均匀分布的随机方法分别选择交叉点,包括交叉点在内的交叉点以下的子树称为交叉段。

f)交叉两个父代个体的交叉段,完成交叉操作。

. 混合变异

为提高算法搜索能力,使结果更接近于全局最优解,避免早熟收敛,本文选择了两种方式来实现突变,即大规模突变和子树突变。

. . 大规模突变

大规模突变是根据整个种群的适应度情况来确定是否进行突变,其目的主要是为了提高较差个体的性能,提高种群多样性,避免陷入早熟,大规模突变的操作过程如下:

a)利用式 $f_i = 1/(f_{\text{population}-i} + 0.0001)$,将最小适应度变为最大适应度问题。

b)计算整个种群适应度:

$$\text{fit}_{\text{population}} = (N \times f_{\text{optimal}})/(\sum_{i=1}^{N} f_i) \tag{6}$$

其中:N 为种群大小;f_{optimal}为最优个体适应度。本文对种群个体进行了按适应度排序,所以第 0代个体就是最优个体,即 $f_{\text{optimal}} = f_0$。

c)如果 $\text{fit}_{\text{population}} > \alpha$,则转 d)进行大规模突变,否则不进行。$\alpha$可根据情况确定,本文中 $\alpha = 0.9$。

d)对种群中第 $N/4 \sim N$ 的个体利用随机产生新个体方式来替代。

. . 子树突变

子树突变主要发生在利用轮盘赌选择了两个父代个体进行交叉后,依据如下步骤进行:

a)随机产生一个数 β,如果 $\beta < P_{\text{mute}}$则转 b)进行子树突变,否则不进行。

b)随机产生变异点,随机生成一棵树结构个体,用该树代替变异个体变异点以下的子树。

c)如果变异后的个体节点数超过最大限制节点,则转 b)重新进行子树变异。

. 新一代个体的选择

由于遗传规划过程存在很大的随机性,适应度高的个体不一定是最差个体。但传统方法中,适应度较大的个体很少有机会在下一代中起作用。为了解决该问题,给适应度较大的个体一定的表现机会,本文引入了按适应度排序,具体操作如下:

a)复制父辈个体。

b)复制通过选择交叉、变异等产生的子代个体。

c)将父辈与子代个体组合在一起,按适应度从小到大排序,从中选择出较优的 N 个个体形成新一代种群。

d)继续进行其他操作。

通过上述改进操作,提高了种群的多样性,并大大降低了遗传规划早熟收敛的概率。

实验分析

为了使遗传规划算法更好地应用在数据拟合领域,前文讨论了在参数设置、交叉个体选择、变异及新一代个体选择等方面的改进。为了验证算法的有效性,在 VC6下实现了该算法,并针对不同类型的数据进行了大量实验,分别与最小二乘法和传统遗传规划算法进行比较。下面给出了 3组不同性质的实验分析,这些实验都是在同一组进化参数下进行的,即 $P_{\text{cross}} = 6, P_{\text{mute}} = 0.01, N = 20, \max\_gen = 50$。区别是函数集和终止符集有所不同。

. 实验

针对明显有规律性的数据,在同样的初始条件下,分别给出 3组测试数据,每组数据包含 10个数据项,如表 1所示。函数集 $F = \{+, -, \times, /\}$,终止符集 $T = \{x\}$,将本文算法分别与最小二乘法和传统遗传规划算法进行比较。

表 1　测试数据 1

编号 i	x_i	y_{1i}	y_{2i}	y_{3i}
1	-0.4	0.76	1.55	-0.24
2	-0.3	0.79	1.39	-0.21
3	-0.2	0.85	1.25	-0.15
4	-0.1	0.91	1.11	-0.09
5	0	0.99	1	0
6	0.1	1.11	0.91	0.11
7	0.2	1.24	0.84	0.24
8	0.3	1.39	0.79	0.39
9	0.4	1.56	0.76	0.57
10	0.5	1.75	0.75	0.74

. . 与最小二乘法比较

针对表 1的 3组测试数据,分别假设基函数是 2阶多项式和 3阶多项式形式,用最小二乘法进行拟合,与本文算法拟合结果进行对比,得到如表 2所示的计算结果。

表 2　实验结果对比 1

组	算法名称	拟合方程	误差
1	2阶拟合	$1.0114x^2 + 0.9964x + 0.9992$	0.004 33
	3阶拟合	$0.0447x^3 + 1.0047x^2 + 0.9902x + 0.9995$	0.004 3
2	本文算法	$x^2 + x + 1$	0.004 5
	2阶拟合	$0.9735x^2 - 0.9949x + 1.002$	0.004 0
	3阶拟合	$0.1496x^3 + 0.9511x^2 - 1.0157x + 1.0031$	0.003 0
	本文算法	$x^2 - x + 1$	0.004 5
3	2阶拟合	$0.9811x^2 + 0.9977x + 0.0027$	0.005 1
	3阶拟合	$-0.0408x^3 + 0.9872x^2 + 1.0033x + 0.0024$	0.005 0
	本文算法	$x^2 + x$	0.005 0

表 2中的误差为平均误差,可以利用式 (7)进行计算,主要用来评价拟合结果的优劣。

$$e = \left(\left(\sum_{k=1}^{n} |f(x_k) - y_k|^2\right)/n\right)^{1/2} \tag{7}$$

. . 与传统遗传规划算法比较

利用传统遗传规划算法和本文算法,针对表 1的每组数据分别进行 51次实验,记录每次找到拟合结果时的进化代数,并进行统计,得到如表 3所示的进化代数对比数据。图 1为本文算法针对三组测试数据得到的拟合曲线与实际数据的拟合情况对比图。以表 1的第 1组数据 (即 x_i, y_{1i})为例,图 2为传统遗传规划算法和本文算法在 51次实验中,每次实验成功时的进化代数对比图;图 3为最优个体适应度变化曲线;图 4为计算过程中种群多样性的变化曲线。

表 3　进化代数

进化代数	算法	最小值	最大值	平均值
1	传统 GP	9	50	29
	本文算法	0	49	21
2	传统 GP	8	50	28
	本文算法	0	47	18
3	传统 GP	8	50	26
	本文算法	0	41	9

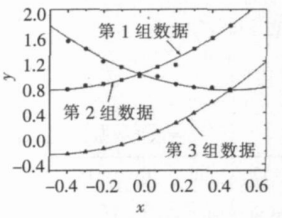

图 1　拟合效果对比图 1

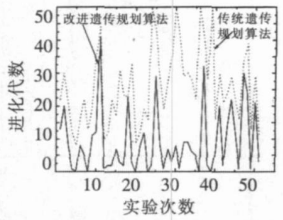

图 2　进化代数对比图

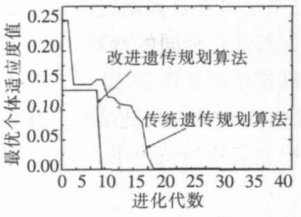

图 3　适应度变化曲线

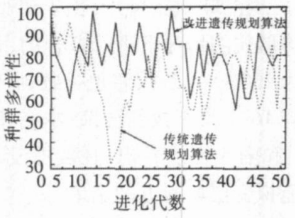

图 4　种群多样性变化曲线

. . 分析

从表 2可以看出,在数据量比较小、且有一定规律性的情况下,利用最小二乘法和本文算法均能得到较好的拟合结果,最小二乘法在有些情况下反而能得出比本文算法更逼近的拟合结果。

从图 1可以看出,在不改变程序的情况下,本文算法可以适应不同数据的拟合,并且基本能找到最佳的拟合函数。

从表 3和图 2可以看出,针对表 1给出的 3组测试数据所进行的 51次实验,本文算法总体上要比传统遗传规划算法运行效率高,即能在较短的时间内找出最佳拟合结果,某些情况下,甚至在第 0代就能找到结果。而传统遗传规划算法需要运行较长时间才能找到拟合结果,有些情况甚至进化完 50代都不一定能找到最佳拟合结果。

从图 3和 4可以看出,本文算法与传统遗传规划算法相比有两个优势,即适应度能快速收敛和种群多样性比较好,这也证明了本文算法的改进是有效的。

. 实验

实验 1主要针对一些规范数据进行实验,这些数据在数量比较小时,通过观察甚至就可以看出其中的规律,难以说明本文算法的优势。在同等条件下,针对一些表面上难以看出其关系的数据进行测试。其中 1组数据如表 4所示。

表 4　测试数据 2

编号 i	1	2	3	4	5
x_i	2	3	4	5	7
y_i	106.42	108.20	109.58	109.50	110.00
编号 i	6	7	8	9	10
x_i	8	10	11	14	15
y_i	109.93	110.49	110.59	110.60	110.90
编号 i	11	12	13		
x_i	16	18	19		
y_i	110.76	111.00	111.20		

针对表 4数据,分别采用最小二乘法、传统遗传规划算法和本文算法进行拟合,通过 50次实验,得到拟合结果如表 5所示。其中误差利用式 (7)计算得到。

表 5 实验结果对比 2

名称	方程	误差	成功率
传统 GP	112 + 111. 18327 / - 0. 311128x^2 - 27. 55132	0. 628 5	69%
最小二乘法	- 0. 02026x^2 + 0. 61161x + 106. 45	0. 501 1	100%
本文算法	111. 4 - 9/x - 1/x^2	0. 205 9	90%

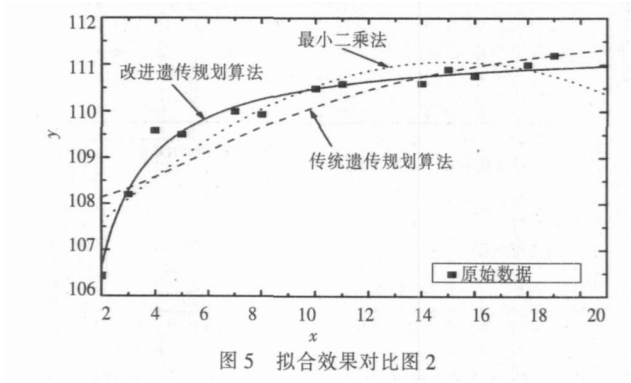

图 5 拟合效果对比图 2

在 50次实验中,由于最小二乘法是在已经确定基函数结构形式的情况下进行拟合的,每次都逼近相同的函数,成功率为 100%,平均误差也一致;而传统遗传规划算法,50次实验中有 16次没有找到结果;本文算法有 5次没有找到结果。从表 5和图 5可以看出,在数据没有明显规律的情况下,本文算法能拟合出最逼近的结果。

. 实验

前面 2组实验,主要针对小批量数据,用来验证算法的有效性和可移植性。本次实验主要通过大量数据,来验证算法的执行效率。利用 MATLAB 6. 5随机生成各种数据点,作为测试数据,如 $y = x + x^2 + \mathrm{normrnd}(-1, 1, 1, 201)$, $x \in [-10, 10]$, x 按 0. 1变化,表示产生 201个数据点,并给 y添加了满足正态高斯分布的随机噪声,在与前面实验同样的条件下,分别用最小二乘法、传统 GP和本文算法进行拟合,得到如表 6所示结果。

修改函数集为 $F = \{+, -, \times, /, \sin, \cos\}$,其他条件不变,利用 $y = \sin(2x) + \mathrm{normrnd}(-0.1, 0.1, 1, 1, 1001)$, $x \in [1, 101]$产生 1 001个测试数据,分别用三种方法进行拟合,得到如表 6所示结果。其他实验情况这里就不再列出。

表 6 实验结果对比 3

	名称	方程	误差	耗时 /ms
	最小二乘法方程	0. 99x^2 + 0. 97x - 0. 7	1. 501 9	2
1	GP	$x^2 + x$	1. 311 6	40
	本文算法	$x^2 + x$ - 1	1. 032 7	15
	最小二乘法方程	8. 07e(-2. 04x^2)	1. 501 9	3
2	GP	$\sin(2x)$	0. 144 8	76
	本文算法	$\sin(\cos x + 2x - \sin(\cos x))$	1. 032 7	25

从表 6可以看出,随着数据点个数的增加,传统 GP和本文算法运算时间也加长,而最小二乘法运算时间变化很小,但最小二乘法的拟合结果却不如传统 GP和本文算法精确。

结束语

传统数据拟合方法需要依据经验知识预先确定拟合函数的结构形式,由此设计的程序也只能解决某一领域的某一确定问题,可移植性较差。而遗传规划具有动态可变特性,不需要先验知识,因此,本文将遗传规划引入到数据拟合领域。同时,结合实际应用经验,为了提高运算效率、避免传统遗传规划算法的早熟收敛、种群多样性差等问题,从适应度函数设计、交叉个体选择、变异操作等方面对其进行了改进。针对不同类型的数据,将本文算法与最小二乘法和传统遗传规划算法等进行了比较,验证了本文算法的有效性,证明其具有一定通用性,可直接移植到其他领域进行应用,并且具有较高的计算精度和执行效率。但本文算法也存在一定的问题,即针对比较复杂的应用场合,需要适当调整函数集和终止符集。

参考文献:

[1] 吴宗敏. 散乱数据的拟合 ——模型、方法与理论 [M]. 北京:科学出版社, 2006: 6-20.

[2] KOZA J R, KEANE M A, STREETER M J, *et al* Genetic programming Ⅳ: routine human-competitive machine intelligence [M]. Norway: Kluwer Academic Publishers, 2003: 15-25.

[3] 侯进军,熊令纯. 遗传程序设计在数据拟合中的应用 [J]. 长沙电力学院学报, 1999, 14 (2): 141-144.

[4] 黄丽剑,李郝林. 遗传规划在测量数据拟合中的应用 [J]. 自动化仪表, 2001, 22 (10): 15-16.

[5] XIA Y, TIAN S P, WEI H Y, *et al* Application of genetic programming and least square method on data fitting [J]. Chinese Journal of Electron Devices, 2007, 30 (4): 1387-1390.

[6] 邵桂芳,李祖枢,陈桂强. 基于进化计算的控制结构设计方法 [J]. 中南大学学报, 2007, 38 (增刊 1): 207-212.

(上接第 474页)

[3] LOHN J D, COLOMBANO S P. A circuit representation technique for automated circuit design [J]. IEEE Trans on Evolutionary Computation, 1999, 3 (3): 205-219.

[4] 李少波,胡建军. 基于遗传编程 (GP)与键合图的机电系统自动设计 [J]. 系统仿真学报, 2002 (11): 1513-1516.

[5] LI Shao-bo, CHEN Xi, HU Jian-jun. Sustainable HFC genetic algorithms based with adaptive migration structure [C]//Proc of IEEE International Conference on Wireless Communications, Networking and Mobile Computing 2007: 653-657.

[6] LI Shao-bo, HU Jian-jun. Evolving vibration absorbers based on genetic programming and bond graphs [C]//Proc of International Conference on Computational Intelligence and Security. [S. l]: IEEE Press, 2006: 202-207.

[7] KOZA J R, BENNETT F H, ANDRE D, *et al* Automated synthesis of analog electrical circuits by means of genetic programming [J]. IEEE Trans on Evolutionary Computation, 1997, 1 (2): 109-128.

[8] CHEN Yue-hui, YANG Bo, ABRAHAM A. Flexible neural trees ensemble for stock index modeling [J]. Neurocomputing, 2007, 70 (4-6): 697-703.

[9] TAY E, FLOWERS W, BARRUS J. Automated generation and analysis of dynamic system designs [J]. Research in Engineering Design, 1998 (10): 15-29.

[10] ABRAHAM A, JAIN R, THOMAS J, *et al* D-SCIDS: distributed soft computing intrusion detection system [J]. Journal of Network and Computer Applications, 2007, 30 (1): 81-98.

[11] 赵曙光. 可编程逻辑器件原理、开发及应用 [M]. 西安:西安电子科技大学出版社, 2000.

[12] YAO X, HIGUICHI T. Promises and challenges of evolvable hardware [J]. IEEE Trans on Systems Man and Cybernetics—Part C: Applications and Reviews, 1999, 29 (1): 87-97.