# 一种新的约束优化遗传算法及其工程应用*

吴华伟[1a]，陈特放[1a]，黄伟明[1b]，许　炳[2]，胡春凯[1b]

(1. 中南大学 a. 信息科学与工程学院；b. 粉末冶金研究院，长沙 410083；2. 空军驻长沙地区军事代表室，长沙 410205)

**摘　要**：提出一种新的用于求解约束优化问题的遗传算法，该算法利用佳点集方法初始化个体以维持种群的多样性。在进化过程中，通过可行解与不可行解算术交叉对问题的决策空间进行搜索；对可行种群与不可行种群分别采用高斯变异和柯西变异，从而协调算法的勘探和开采能力。几个标准测试问题的实验结果表明该算法的有效性；应用新算法求解两个工程优化设计问题，结果表明该算法的可行性。

**关键词**：约束优化问题；遗传算法；算术交叉；变异

**中图分类号**：TP301　　**文献标志码**：A　　**文章编号**：1001-3695(2013)02-0367-04

doi:10.3969/j.issn.1001-3695.2013.02.012

# Novel constrained optimization genetic algorithm and its engineering applications

WU Hua-wei[1a], CHEN Te-fang[1a], HUANG Wei-ming[1b], XU Bing[2], HU Chun-kai[1b]

(1. *a. School of Information Science & Engineering, b. Research Institute of Powder Metallurgy, Central South University, Changsha* 410083, *China*; 2. *Military Representative Office of Air Force in Changsha, Changsha* 410205, *China*)

**Abstract**: This paper proposed a novel genetic algorithm to solve constrained optimization problems. It introduced the individual generation based on good-point-set method into the genetic algorithm initial step, which maintained the population diversity of the genetic algorithm. In the evolution process, it searched the decision space of a problem through the arithmetic crossover operator of feasible and infeasible solutions. In order to coordinate the exploitation and the exploration ability of the algorithm, it used Gaussian and Cauchy mutation operators to the feasible and infeasible subpopulation respectively. It tested several benchmark problems and two engineering design problems. The results show that the proposed method is an effective way for constrained optimization problems.

**Key words**: constrained optimization problem; genetic algorithm; arithmetic crossover; mutation

## 0　引言

约束优化问题是广泛存在于工程应用和科学研究中的一类优化问题。不失一般性，一个非线性约束优化问题可描述为

$$\begin{aligned} &\min f(\boldsymbol{x}) \quad \boldsymbol{x}=(x_1,\cdots,x_n)\\ &\text{s.t. } g_j(\boldsymbol{x})\leqslant 0 \quad j=1,2,\cdots,p\\ &h_j(\boldsymbol{x})=0 \quad j=p+1,\cdots,m\\ &l_i\leqslant x_i\leqslant u_i \quad i=1,2,\cdots,n \end{aligned} \tag{1}$$

其中：$\boldsymbol{x}\in\Omega\subseteq S$ 为决策变量，$\Omega$ 为可行域，$S$ 为决策空间。一般地，$S$ 为 $\mathrm{R}^n$ 中的 $n$ 维长方体。变量 $x_i$ 的取值为 $[l_i,u_i]$。$f(\boldsymbol{x})$ 为目标函数，$g_j(\boldsymbol{x})\leqslant 0$ 为第 $j$ 个不等式约束条件，$h_j(\boldsymbol{x})=0$ 为第 $j$ 个等式约束条件。通常将等式约束转换为两个不等式约束来处理：

$$\begin{cases} h_j(\boldsymbol{x})-\sigma\leqslant 0\\ -h_j(\boldsymbol{x})-\sigma\leqslant 0 \end{cases} \tag{2}$$

其中：$\sigma$ 为容忍值，一般取一个很小的正数。

遗传算法是一种基于种群迭代的随机全局优化方法，其求解过程中不依赖于目标函数的解析性质，无须借助问题的梯度信息，能以较大概率收敛到全局最优解，更适合于求解约束优化问题，但它需要结合一种合适的约束处理技术。利用遗传算法求解约束优化问题已成为进化计算领域的研究热点之一，并涌现出大量的约束优化遗传算法[1,2]。

Michalewicz 等人[3]将现有的基于进化算法的约束处理技术分成惩罚函数法、多目标法、可行解优于不可行解法和其他混合方法四类。惩罚函数法是最常用的约束处理方法之一，但很难确定合适的惩罚因子；多目标法将约束条件作为一个或多个目标来处理，但会导致算法的计算量增大；可行解优于不可行解法则没有充分利用不可行解进行搜索[4]。针对上述方法的不足，本文提出一种新的求解约束优化问题的遗传算法。

## 1　改进的约束优化遗传算法

遗传算法的基本操作包括交叉、变异和选择操作，它是由 $NP$(种群大小)个 $n$ 维(变量个数)的参数矢量 $\boldsymbol{x}_i^t(i=1,2,\cdots,NP)$ 构成的种群在搜索空间进行寻优，其中 $t$ 表示第 $t$ 代。

### 1.1　种群初始化

由遗传算法的机理可知，交叉、变异和选择操作在一定程度上对种群具有依赖性[5]。在求解约束优化问题前无法预知全局最优解所在区域的情况下，初始种群个体必须充分代表解空间的个体，在有限数量内最大限度地表征所有个体的信息，这样才能使算法以较快的方式快速逼近最优解。因此，初始种

**收稿日期**：2012-06-10；**修回日期**：2012-07-24　　**基金项目**：国家“863”计划资助项目(2009AA034302)

**作者简介**：吴华伟(1979-)，男，湖北襄阳人，博士，主要研究方向为智能控制、飞机着陆系统(csuwhw2008@163.com)；陈特放(1957-)，湖南涟源人，教授，博导，主要研究方向为机车故障诊断、智能交通系统.

群的确定问题实质上是一个如何利用有限的个体来全面科学地表征问题解空间特征的优化设计问题[5]。佳点集方法是一种有效的、可以减少实验次数的实验方法,在相同取点个数的条件下,佳点序列要比其他方法选取的点序列更均匀。另外,佳点集方法的精度与维数无关,能克服均匀设计方法的不足,且产生的初始种群具有较好的多样性[6]。因此,本文采用佳点集方法来产生初始种群个体。图 1 是采用佳点集方法产生的规模为 80 的初始种群个体分布,其中,变量的维数为 2,变量的取值为[ -20,20]。

从图 1 可以看出,与随机方法相比,佳点集方法产生的初始种群个体分布均匀,具有较好的多样性。

### 1.2 适应度函数

本质上,约束优化遗传算法具有两个明确的目标[7]:a)群体快速地靠近或进入可行域;b)收敛于全局最优解。在无约束优化问题中,通常利用目标函数来评价个体,而在求解约束优化问题时,必须同时考虑目标函数和约束条件。遗传算法通常是根据适应度值的大小来判断个体优劣,每个个体均有其对应的适应度值,该值由适应度函数来度量。对于约束优化问题式(1),本文采用式(3)作为个体的适应度函数[4]:

$$f(x) = \mathrm{fit}(k) + \mathrm{voi}(k) \tag{3}$$

其中:

$$\mathrm{fit}(k) = f(x),\ \mathrm{voi}(k) = \sum_{j=1}^{2m-p} \max(0, g_j(x))$$
$$k = 1, 2, \cdots, n$$

$\mathrm{fit}(k)$对应于个体的目标函数值;$\mathrm{voi}(k)$对应于所求问题的约束违反程度,反映了个体与约束边界的接触程度。由式(3)可知,当 $\mathrm{voi}(k) = 0$ 时,说明该个体是可行个体,否则是不可行个体。

### 1.3 交叉操作

在遗传算法中,交叉操作的作用是增加种群个体的多样性,扩大寻优范围,使得遗传算法具有较强的搜索能力。交叉操作是遗传算法搜索整个决策空间的重要操作,它通过组合父代个体的信息形成新的子代个体。传统的交叉方法是基于二进制表达,常用的有单点交叉、两点交叉、多点交叉等,即随机选择一个、两个或多个位置进行交叉,这些交叉操作产生的个体都是随机的,具有一定的盲目性,在一定程度上降低了算法的搜索效率。

由文献[4]可知,若两个个体进行算术交叉操作,其产生的后代个体一定位于两者之间的连线上。鉴于此,本文将种群分为可行子种群和不可行子种群,分别从两个子种群中随机选择个体进行算术交叉操作,如图 2 所示。

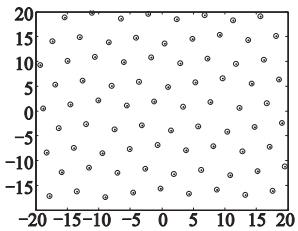


图1 佳点集方法产生的80个初始种群分布

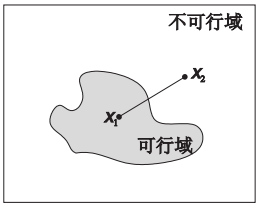


图2 可行解与不可行解算术交叉示意图

假设 $\boldsymbol{x}_1^t$ 和 $\boldsymbol{x}_2^t$ 是任意给定的两个个体,其进行算术交叉操作产生新个体为

$$\begin{cases} \boldsymbol{x}_1^{t+1} = \lambda \boldsymbol{x}_2^t + (1-\lambda)\boldsymbol{x}_1^t \\ \boldsymbol{x}_2^{t+1} = \lambda \boldsymbol{x}_1^t + (1-\lambda)\boldsymbol{x}_2^t \end{cases} \tag{4}$$

其中:$t$ 为当前进化代数,$\lambda$ 为[0,1]之间的随机数。若 $\boldsymbol{x}_1^t$ 为当前最优个体,$\boldsymbol{x}_2^t$ 为随机选择的个体,$\boldsymbol{x}_1^t$ 和 $\boldsymbol{x}_2^t$ 进行算术交叉操作,产生的后代个体更靠近当前最优解。

由图 2 可知,若 $x_1$ 为可行解,$x_2$ 为不可行解,$x_1$ 和 $x_2$ 进行算术交叉操作后产生的后代个体一定在 $x_1$ 与 $x_2$ 之间的连线上,不断向可行域边界靠近。这对于那些全局最优解位于可行域边界上或附近、或可行域占搜索空间比例很小的问题尤其有效。

### 1.4 变异操作

变异操作可以避免算法陷入局部最优,同时也能维持种群个体的有效性。为了平衡算法的全局搜索和局部搜索能力,本文对可行种群与不可行种群分别采用高斯变异和柯西变异操作。图 3 是高斯分布和柯西分布概率密度函数比较曲线,其中 $\sigma = 1, \mu = 0, a = 1$。

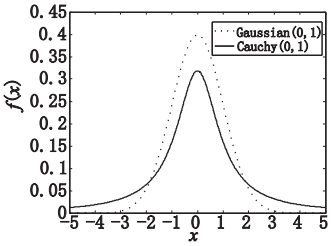


图3 柯西分布与高斯分布比较

从图 3 可以看出,柯西分布在原点处的峰值比高斯分布小,而两端长扁形状趋近于零的速度比高斯分布慢。因此,基于柯西分布的邻域产生小扰动的能力相对高斯分布有所下降,而产生大扰动的能力增强,这样经过柯西变异的个体就更有可能跳出局部最优,从而增强了算法的全局搜索能力。

高斯变异算子的表达式为

$$x_i' = x_i + \eta_1 \sigma_1 \tag{5}$$

其中:$\eta_1$ 为扰动幅值参数,$\sigma_1$ 为满足高斯分布的随机变量。

柯西变异算子的表达式为

$$x_i' = x_i + \eta_2 \sigma_2 \tag{6}$$

其中:$\eta_2$ 为扰动幅值参数,$\sigma_2$ 为满足柯西分布的随机变量。

对不可行种群进行柯西变异操作,增强算法的全局搜索能力,从不同的搜索方向快速地靠近可行域边界或进入可行域里面,从而实现了约束优化遗传算法的第一个目标。对可行种群进行高斯变异操作以加强算法的局部搜索能力,向全局最优解逼近,达到了算法的第二个目标。因此,该混合变异操作可以平衡算法的全局和局部搜索能力。

### 1.5 选择操作

由于在交叉和变异操作中对可行个体和不可行个体分别采用不同的策略,因此在种群中需要保持不可行个体的存在,本文对可行和不可行个体分别进行选择。针对可行种群,按其目标函数值进行排序,选择目标函数值小的个体进入下一代种群;对于不可行种群,分别按其目标函数值和约束违反度进行排序,随机产生一个[0,1]之间均匀分布的数 rand,若 $\mathrm{rand} > b$,则按其目标函数值选择个体;否则,按其约束违反度选择个体进入下一代种群,$b$ 为一设定的常数。这种机制选择出来的不可行个体要么拥有最小目标函数值,要么拥有最小约束违反程度,从而提高了算法对解空间的探索能力。

### 1.6 算法步骤

综上所述,本文提出的约束优化遗传算法步骤如下:

a)设置算法参数,初始化种群。种群规模 $N$,交叉概率 $p_c$,

变异概率 $p_m$,利用佳点集方法产生初始种群。检验种群中是否有可行解,若没有,则重新初始化,直至种群中至少含有一个可行解。令 $k=0$。

b)计算种群中各个体的适应度值,将种群分为可行子种群和不可行子种群。

c)按给定的交叉概率选择可行个体与不可行个体进行算术交叉操作。

d)按给定的变异概率对可行个体与不可行个体分别进行高斯变异和柯西变异。

e)按照 1.5 节所给出的方法进行选择操作,选择个体进入到下一代种群。

f)判断算法是否满足结束条件,若满足,则算法结束,输出最优解;否则,执行 g)。

g)令 $k=k+1$,返回 b)。

## 2　数值实验及分析

为了测试本文算法的性能,从文献[1]选取两个标准测试问题 $P_1$ 和 $P_2$ 进行测试。两个测试问题的具体表达式为

$$P_1:\min f(x)=(x_1-10)^3+(x_2-20)^3$$

$$\text{s.t. } g_1=-(x_1-5)^2-(x_2-5)^2+100\leqslant 0$$

$$g_2=-(x_1-6)^2+(x_2-5)^2-82.81\leqslant 0$$

$$13\leqslant x_1\leqslant 100,0\leqslant x_2\leqslant 100$$

$$P_2:\max f(x)=\left|\frac{\sum_{i=1}^{n}\cos^4(x_i)-2\prod_{i=1}^{n}\cos^2(x_i)}{\sqrt{\sum_{i=1}^{n}ix_i^2}}\right|$$

$$\text{s.t.}\quad g_1=0.75-\prod_{i=1}^{n}x_i\leqslant 0$$

$$g_2=\sum_{i=1}^{n}x_i-7.5n\leqslant 0$$

$$n=20,0\leqslant x_i\leqslant 10,i=1,2,\cdots,20$$

将本文算法记为 ICOGA,并与文献[1]中的 SR、文献[8]中的 SMES 和文献[9]中的 PA 算法得到的结果进行比较。在比较实验中,ICOGA 的参数设置为:种群规模 $N=100$,交叉概率 $p_c=0.7$,变异概率 $p_m=0.1$。对问题 $P_1$,迭代次数设置为 1 000,对于 $P_2$,迭代次数设置为 3 000。其他算法的参数设置分别见其各自的文献。每个测试问题在相同条件下独立运行 30 次实验,记录其最好值(best)、平均值(mean)和最差值(worst)。表 1 给出了在上述参数设置下,四种算法对两个标准测试问题的寻优结果比较。

表 1　四种算法对两个问题的寻优结果比较

| 测试函数 | 算 法 | 已知最优值 | 最好值(best) | 平均值(mean) | 最差值(worst) |
|---|---|---|---|---|---|
| $P_1$ | SR | −6 961.814 | −6 961.814 | −6 875.940 | −6 350.262 |
| | SMES | | −6 961.814 | −6 961.284 | −6 952.482 |
| | PA | | −6 961.814 | −6 961.811 | −6 961.803 |
| | ICOGA | | −6 961.814 | −6 961.814 | −6 961.814 |
| $P_2$ | SR | −0.803 619 | −0.803 515 | −0.781 975 | −0.726 288 |
| | SMES | | −0.803 601 | −0.785 238 | −0.751 322 |
| | PA | | −0.803 617 | −0.801 299 | −0.792 583 |
| | ICOGA | | −0.803 619 | −0.801 537 | −0.793 621 |

从表 1 可知,本文所提出的 ICOGA 对两个问题 30 次实验中一致地找到全局最优解。对于问题 $P_1$,四种算法都找到了全局最优解,ICOGA 得到的平均值和最差值比其他三种算法都要优。对于问题 $P_2$,其他三种算法都没找到其全局最优解。图 4 给出了 ICOGA 对两个测试问题的寻优曲线,从图 4 可以看出,ICOGA 能快速地收敛到问题的全局最优解。基于以上比较和分析,ICOGA 要优于其他三种算法。

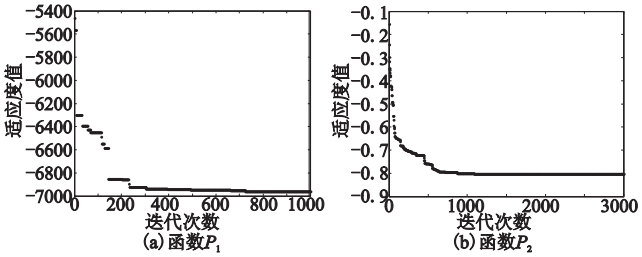


图4　函数$P_1$和$P_2$的寻优曲线

## 3　ICOGA 在工程设计问题中的应用

为了进一步验证本文算法的有效性,将其应用到压力容器优化设计问题和拉压弹簧优化设计问题中。

### 3.1　压力容器优化设计问题

压力容器优化设计问题如图 5 所示。在图 5 中,设计变量分别为:两段帽的厚度为 $T_a$(记为 $x_2$),壳的厚度为 $T_b$(记为 $x_1$),内径为 $R$(记为 $x_3$),容器中间圆柱部分的长度为 $L$(记为 $x_4$)。$R$ 和 $L$ 是连续的变量;$T_a$ 和 $T_b$ 是 0.062 5 inch 的倍数,因为它是标准冷轧钢板的有效厚度。压力容器优化设计问题的目标是总的费用最小,其中包括材料、成型加工和焊接的费用。其目标函数和约束条件为

$$\min f(\boldsymbol{x})=0.6224x_1x_3x_4+1.7781x_2x_3^2+3.1661x_1^2x_4+19.84x_1^2x_3$$

$$\text{s.t.}\quad g_1(\boldsymbol{x})=x_1+0.0193x_3\leqslant 0$$

$$g_2(\boldsymbol{x})=x_2+0.00954x_3\leqslant 0$$

$$g_3(\boldsymbol{x})=-\pi x_3^2x_4+\frac{4}{3}\pi x_3^3+1296000\leqslant 0$$

$$g_4(\boldsymbol{x})=x_4-240\leqslant 0$$

其中:$0.1\leqslant x_1,x_2\leqslant 99,10\leqslant x_3,x_4\leqslant 200$。

利用 ICOGA 对压力容器优化设计问题进行求解,并与文献[5]中的方法和文献[10]的 PSO-ABC 算法进行比较,结果如表 2 所示。从表 2 中的结果可知,ICOGA 优于文献[5,10]中的方法。

表 2　几种算法对压力容器优化设计问题的结果比较

| 算法 | $x_1(T_b)$ | $x_2(T_a)$ | $x_3(R)$ | $x_4(L)$ | $f(x)$ |
|---|---|---|---|---|---|
| Kannan | 1.125 0 | 0.625 0 | 58.291 0 | 43.690 0 | 7 198.042 8 |
| Deb | 0.937 5 | 0.500 0 | 48.329 0 | 112.679 0 | 6 410.381 1 |
| Coello | 0.812 5 | 0.437 5 | 40.323 9 | 200.000 0 | 6 288.744 5 |
| CPSO | 0.812 5 | 0.437 5 | 42.091 2 | 176.746 0 | 6 061.077 7 |
| PSO-ABC | 0.812 5 | 0.437 5 | 42.098 4 | 176.636 6 | 6 059.714 3 |
| ICOGA | 0.812 5 | 0.437 5 | 42.092 4 | 176.602 8 | 6 057.909 2 |

### 3.2　拉压弹簧优化设计问题

拉压弹簧优化设计问题的目标是使其重力最小,包括三个连续变量和四个非线性不等式约束,其中包括侥曲度、剪应压力、冲击频率和外直径。拉压弹簧优化设计问题的结构如图 6 所示。

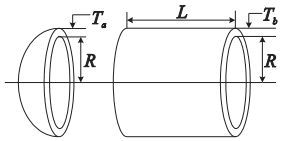


图5　压力容器优化设计问题

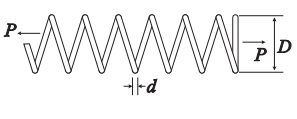


图6　拉压弹簧优化设计问题

在图 6 中,设计变量分别为评价线盘直径 $D$(记为 $x_2$)、焊丝直径 $d$(记为 $x_1$)和绕线圈数 $P$(记为 $x_3$)。其目标函数和约束条件为

$$\min f(\boldsymbol{x})=(x_3+2)x_2x_1^2$$

$$\text{s.t. } g_1(\boldsymbol{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leqslant 0$$

$$g_2(\boldsymbol{x}) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leqslant 0$$

$$g_3(\boldsymbol{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leqslant 0$$

$$g_4(\boldsymbol{x}) = \frac{x_1 + x_2}{1.5} - 1 \leqslant 0$$

其中:$0.05 \leqslant x_1 \leqslant 2, 0.25 \leqslant x_2 \leqslant 1.3, 2 \leqslant x_3 \leqslant 15$。

利用 ICOGA 对拉压弹簧优化设计问题进行求解,并与文献[5]中的方法和文献[10]的 PSO-ABC 算法进行比较,结果如表 3 所示。从表 3 中的结果可知,ICOGA 优于文献[5]中的方法,比 PSO-ABC 算法要稍劣。

表 3 几种算法对拉压弹簧优化设计问题的结果比较

| 算法 | $x_1(d)$ | $x_2(D)$ | $x_3(P)$ | $f(x)$ |
|---|---|---|---|---|
| Belegundu | 0.050 000 | 0.315 900 | 14.250 000 | 0.012 833 4 |
| Arora | 0.053 396 | 0.399 180 | 9.185 400 | 0.012 730 3 |
| Coello | 0.051 480 | 0.351 661 | 11.632 201 | 0.012 704 8 |
| CPSO | 0.051 728 | 0.357 644 | 11.244 543 | 0.012 674 7 |
| PSO-ABC | 0.050 000 | 0.374 430 | 3.200 110 | 4.867 7e-3 |
| ICOGA | 0.050 935 | 0.351 541 | 5.009 216 | 0.012 551 4 |

## 4 结束语

本文提出一种改进的约束优化遗传算法。为了维持种群个体的多样性,利用佳点集方法产生初始种群。将种群分为可行子种群和不可行子种群,分别从两个子种群中随机选择个体进行算术交叉操作,使算法从不同搜索方向逼近全局最优解。对可行个体与不可行个体分别采用高斯变异与柯西变异,从而达到了约束优化进化算法的两个目标。两个标准测试问题和两个工程设计优化问题的实验结果表明,该算法具有较强的通用性和稳定性。

**参考文献:**


[1] RUNARSSON T P, YAO Xin. Stochastic ranking for constrained evolu-tionary optimization[J]. IEEE Trans on Evolutionary Computation,2000,4(3):284-294.

[2] 王翔, 董晓马, 阎瑞霞, 等. 改进 DE/EDA 算法在求解难约束优化问题中的应用研究[J]. 计算机应用研究,2010,27(11):4114-4117.

[3] MICHALEWICZ Z, SCHOENAUER M. Evolutionary algorithm for constrained parameter optimization problems [J]. Evolutionary Computation,1996,4(1): 1-32.

[4] 梁昔明, 龙文, 秦浩宇, 等. 基于种群个体可行性的约束优化进化算法[J]. 控制与决策,2010,25(8):1129-1132.

[5] 龙文. 求解两类优化问题的混合进化算法及其应用[D]. 长沙:中南大学,2011.

[6] 张铃, 张钹. 佳点集遗传算法[J]. 计算机学报,2001,24(9):917-924.

[7] CAI Zi-xing, WANG Yong. A multiobjective optimization based on evolutionary algorithm for constrained optimization[J]. IEEE Trans on Evolutionary Computation, 2006,10(3): 658-675.

[8] MEZURA-MONTES E, COELLO C A C. A simple multimembered evolution strategy to solve constrained optimization problems [J]. IEEE Trans on Evolutionary Computation,2005,9(1): 1-17.

[9] 梁昔明,秦浩宇,龙文. 一种求解约束优化问题的遗传算法[J]. 计算机工程,2010,36(14): 147-149.

[10] 王珂珂,吕强,赵汗青,等. 求解工程约束优化问题的 PSO-ABC 混合算法[J]. 计算机应用研究,2012,29(4):1230-1233.


(上接第 366 页)间。第二组测试同样选取文档大小为 44.3 MB,查询的阈值设定为 0.5,分别执行表 1 所示的五个查询用例。第三组测试查询的阈值设定为 0.5,查询用例为 $Q_1$,然后选取大小不同的五个文档进行测试。从图 5~7 明显看出在三种情况下 CUTwigList 性能都要优于 TwigList。其原因是 CUTwigList 根据阈值设定了过滤策略,匹配之前减少了所处理节点的个数,大大减少了出入栈的操作;而 TwigList 算法需要遍历整个文档树的节点,匹配过程产生大量无关的中间结果。

## 5 结束语

本文提出了一种非归并不确定 XML 小枝模式查询算法,即 CUTwigList 算法,扩展了 P-文档使之适用于包含连续节点,采用特定的标志规则对分布节点进行区别,使用小素数对文档树进行编码;查询匹配过程中利用阈值对不符合要求的节点进行过滤,构建结果匹配链表,枚举输出时根据节点互斥属性和阈值再过滤,得到最终的匹配结果。在理论分析和实验测试方面均表明 CUTwigList 具有高效性。进一步的工作是对多维连续分布的编码和查询技术作更深入的研究。

**参考文献:**


[1] 王建卫,郝忠孝. 概率关系模式与概率 XML 模式转换算法研究[J]. 计算机应用研究,2011,28(2):609-612.

[2] ZHANG C, NAUGHTON J, DeWITT D, *et al*. On supporting containment queries in relational database management systems[C]// Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM Press,2001:425-426.

[3] AI-KHALIFA S, JAGADISH H V, KOUDAS N, *et al*. Structural joins: a primitive for efficient XML query pattern matching[C]//Proc of the 18th International Conference on Data Engineering. Washington DC: IEEE Computer Society,2002:141-152.

[4] NICOLAS B, NICK K, DIVESH S. Holistic twig joins: optimal XML pattern matching[C]//Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2002: 310-321.

[5] LI Ya-wen, WANG Guo-ren, XIN Jun-chang, *et al*. Holistically twig matching in probabilistic XML[C]//Proc of the 25th International Conference on Data Engineering. Washington DC: IEEE Computer Society,2009:1649-1656.

[6] CHEN Song-ting, LI Hua-gang, TATEMURA J C, *et al*. Twig$^2$Stack: bottom-up processing of generalized-tree-pattern queries over XML documents[C]//Proc of the 32nd International Conference on Very Large Data Bases. 2006:283-294.

[7] QIN Lu, YU J X, DING Bo-lin. TwigList: make twig pattern matching fast[C]//Proc of the 32nd International Conference on Very Large Data Bases. 2006:313-324.

[8] 王建卫,郝忠孝. 一种概率 XML 树化简算法[J]. 计算机应用研究,2010,27(12):4541-4547.

[9] ABITEBOUL S, CHAN T H, KHARLAMOV E. Aggregate queries for discrete and continuous probabilistic XML[C]//Proc of the 13th International Conference on Database Theory. New York: ACM Press,2010:50-61.