1000-9825/2003/14(07)1243 ©2003 Journal of Software 软 件 学 报 Vol.14, No.7

Pareto 强度值演化算法求解约束优化问题*

周育人[1,2+], 李元香[2], 王 勇[2], 康立山[2]

[1](华南理工大学 计算机科学与工程学院,广东 广州 510640)
[2](武汉大学 软件工程国家重点实验室,湖北 武汉 430072)

A Pareto Strength Evolutionary Algorithm for Constrained Optimization

ZHOU Yu-Ren[1,2+], LI Yuan-Xiang[2], WANG Yong[2], KANG Li-Shan[2]

[1](College of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China)
[2](State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)
+ Corresponding author: E-mail: zhouyuren@hotmail.com
Received 2002-08-12; Accepted 2002-10-17

Zhou YR, Li YX, Wang Y, Kang LS. A Pareto strength evolutionary algorithm for constrained optimization. *Journal of Software*, 2003,14(7):1243~1249.
http://www.jos.org.cn/1000-9825/14/1243.htm

Abstract: A new approach is presented to handle constraints optimization using evolutionary algorithms. It neither uses any penalty function nor makes distinction between feasible solutions and infeasible solutions. The new technique treats constrained optimization as a two-objective optimization. One objective is original objective function, the other is the degree violating the constraints. Individual's ranking procedure is based on its Pareto strength which appears first in multi-objective optimization. Pareto strength evaluates an individual's fitness dependent on the number of dominated points in the population. By combining Pareto strength ranking procedure with minimal generation gap modal, a new real-coded genetic algorithm is proposed. The new approach is compared against other evolutionary optimization techniques in several benchmark functions. The results obtained show that the new approach is a general, effective and robust method. Its performance outperforms some other techniques. Especially for some complicated optimization problems with inequality and equality constraints, the proposed method provides better numerical accuracy.
Key words: evolutionary algorithm; constrained optimization; multi-objective; Pareto strength

摘 要: 提出了一种求解约束函数优化问题的方法.它不使用传统的惩罚函数,也不区分可行解和不可行解.新的演化算法将约束优化问题转换成两个目标优化问题,其中一个为原问题的目标函数,另一个为违反约束条件的程度函数.利用多目标优化问题中的 Pareto 优于关系,定义个体 Pareto 强度值指标以便对个体进行排序选优,根据 Pareto 强度值排序和最小代数代沟模型设计出新的实数编码遗传算法.对常见测试函数的数值实验证实了新方法的有效性、通用性和稳健性,其性能优于现有的一些演化算法.特别是对于一些既有等式约束又有不等式约束的复杂非线性规划问题,该算法获得了更高精度的解.

* Supported by the National Natural Science Foundation of China under Grant No.69703011 (国家自然科学基金)
第一作者简介: 周育人(1965－),男,湖南岳阳人,博士生,副教授,主要研究领域为演化计算,并行计算.

关键词: 演化算法;约束优化问题;多目标;Pareto 强度值
中图法分类号: TP18 **文献标识码**: A

演化算法(evolutionary algorithm)在许多优化问题上取得了成功,与传统的非线性规划方法相比,演化算法不需要梯度(导数)等信息,同时它又是一个全局性搜索方法,相对而言,它陷入局部最优的机会较小.研究人员现已提出多种演化算法求解约束优化问题,Michalewicz 在文献[1]中将现有的约束优化演化算法分成 4 类:保存可行解方法、惩罚函数法、可行解优于不可行解方法以及其他混合方法.这些方法各有特点,主要的不足在于,它们往往对某一类问题有效而对另一类问题则效果不佳.与确定性非线性规划一样,演化算法求解约束优化问题面对的主要问题是如何处理约束条件.本文提出将约束优化问题(最小值问题)转换成两个目标函数的最小值问题,一个为原问题的目标函数,另一个为违反约束条件的程度函数;利用多目标优化问题中的 Pareto 优于关系,我们定义个体 Pareto 强度指标,根据 Pareto 强度指标,对上述两个目标函数组成的向量进行排序,使用演化算法求出原问题的解.数值实验证实了新方法的可行性和有效性,对既有等式约束又有不等式约束的测试函数,实验显示新方法求解的精度优于已有的一些演化算法.

1 相关的工作

一个非线性规划问题(nonlinear programming,简称 NLP)一般可表示为求目标函数 f 的最小化:

$$\text{Minimize } f(\boldsymbol{x}), \boldsymbol{x}=(x_1,x_2,\ldots,x_n)\in R^n.$$

这里,$\boldsymbol{x}\in F\subseteq S$,$S$ 为目标函数的搜索空间,F 为可行区域,一般地,S 为 R^n 中的 n 维长方体:$l(i)\le x_i\le u(i)$,$l(i),u(i)$为常数,$i=1,\ldots,n$.可行区域 F 满足 m 个附加的不等式和等式约束条件:$g_j(\boldsymbol{x})\le 0$, $j=1,\ldots,q$ 和 $h_j(\boldsymbol{x})=0$, $j=q+1,\ldots,m$.

演化算法处理约束条件常见的方法为罚函数法,惩罚函数一般取为个体 $\boldsymbol{x}$ 到可行区域的距离的函数,通常用下面定义的 $f_j(\boldsymbol{x})$构造罚函数:

$$f_j(x)=\begin{cases}\max\{0,g_j(x)\}, & 1\le j\le q\\ |h_j(x)|, & q+1\le j\le m\end{cases}.$$

$\sum_{j=1}^{m} f_j(x)$ 表示个体 $\boldsymbol{x}$ 违反约束的程度,也表示个体 $\boldsymbol{x}$ 到可行区域的距离.

罚函数法有静态罚函数和动态罚函数之分,研究表明,动态罚函数的性能优于静态罚函数.Joines 构造的动态罚函数是较为成功的一个,其个体的适应值(第 t 代)定义为

$$fitness(\boldsymbol{x})=f(\boldsymbol{x})+(C\times t)^{\alpha}\sum_{j=1}^{m} f_j^{\beta}(x), \tag{1}$$

这里,C,α,β是需要调整的参数.当演化代数增加时,惩罚项也增加.

带参数的罚函数法的主要缺点是算法的性能强烈地依赖于参数的选取,若惩罚参数过小,则算法找到的解远离真正的最优解;若参数过大,则会引发计算的困难.另外,文献[2]指出式(1)定义的算法容易产生早熟收敛.

演化算法处理约束优化问题的另一个常见方法是在搜索群体中区分可行解和不可行解.Powell 用下式定义个体的适应值:

$$fitness(\boldsymbol{x})=f(\boldsymbol{x})+r\sum_{j=1}^{m} f_j^{\beta}(x)+\theta(t,\boldsymbol{x}), \tag{2}$$

其中 r 为参数.式(2)只不过在普通惩罚函数基础上增加了一项$\theta(t,\boldsymbol{x})$,$\theta(t,\boldsymbol{x})$为与进化代数 t 有关的针对不可行解的惩罚项.

文献[3]在式(2)的基础上提出了基于下式适应值的锦标赛选择算子:

$$fitness(\boldsymbol{x})=\begin{cases}f(x), & x\text{是可行解}\\ f_{\max}+\sum_{j=1}^{m} f_j(x), & x\text{不是可行解}\end{cases}, \tag{3}$$

其中 $f_{\max}$ 为当前群体中可行解的最大函数值.

式(3)定义的选择算子体现了以下原则:(1) 两个个体都为可行解时比较它们的函数值;(2) 当两个个体都

为不可行解时比较它们违反约束的程度;(3) 可行解总是优于不可行解.文献[4]提出了进一步的改进方法,取得了良好的效果.

区分可行解与不可行解方法的缺点是,求解过程依赖于群体中存在可行解,对于可行解比例为 0 的等式约束问题要转化为不等式约束问题,这必然会影响算法的性能和精度.

2 基于 Pareto 强度值的个体排序

回顾普通的惩罚函数 $f(\boldsymbol{x})+r\sum_{j=1}^{m}f_j(x)$,它可以看成是目标函数 $f(\boldsymbol{x})$和违反约束条件的程度函数 $\sum_{j=1}^{m}f_j(x)$ 的权组合.实现罚函数法的困难在于其参数难以选择和控制,我们能否同时直接对这两个函数求最小值?如果 $\boldsymbol{x}^*$既是 $f(\boldsymbol{x})$的最小值又是 $\sum_{j=1}^{m}f_j(x)$ 的最小值($\sum_{j=1}^{m}f_j(x)$ 的最小值为 0),那么,$\boldsymbol{x}^*$就是原约束问题的解,这样问题就转化为求由 $f(\boldsymbol{x})$和 $\sum_{j=1}^{m}f_j(x)$ 组成的二维向量的最小值问题.但是,向量之间比较大小与普通实数的比较大小有本质的不同,一般任意两个向量之间不能比较大小,向量之间的大小关系是偏序关系,演化算法的排序选优似乎无法进行下去.下面我们利用向量之间的 Pareto 优于关系,在群体中对个体引入 Pareto 强度值概念,由 Pareto 强度值对个体进行排序,从而使演化选优得以进行.

以下记 $s_1(\boldsymbol{x})=f(\boldsymbol{x}),s_2(\boldsymbol{x})=\sum_{j=1}^{m}f_j(x)$.

考虑向量最小化问题:

$$\text{Minimize}\quad \boldsymbol{y}=\boldsymbol{s}(\boldsymbol{x})=(s_1(\boldsymbol{x}),s_2(\boldsymbol{x})),$$

$\boldsymbol{x}=(x_1,x_2,\dots,x_n)\in F\subseteq S\subseteq R^n$,$F,S$ 的定义如前所述.

定义 1. 设 $\boldsymbol{a}\in S,\boldsymbol{b}\in S$,称 $\boldsymbol{a}$ Pareto 优于 $\boldsymbol{b}$(Pareto dominate 记 $\boldsymbol{a}\prec\boldsymbol{b}$)或 $\boldsymbol{b}$ Pareto 劣于 $\boldsymbol{a}$ 当且仅当 $\forall i\in\{1,2\}:s_i(\boldsymbol{a})\le s_i(\boldsymbol{b})$且 $\exists j\in\{1,2\}:s_j(\boldsymbol{a})<s_j(\boldsymbol{b})$;称 $\boldsymbol{a}$ 覆盖 $\boldsymbol{b}$(cover 记 $\boldsymbol{a}\prec=\boldsymbol{b}$)当且仅当 $\boldsymbol{a}\prec\boldsymbol{b}$ 或 $f(\boldsymbol{a})=f(\boldsymbol{b})$.

显然,Pareto 优于关系是偏序关系,对于 $\boldsymbol{a}\in S,\boldsymbol{b}\in S$,可能会出现既没有 $\boldsymbol{a}\prec\boldsymbol{b}$,也没有 $\boldsymbol{b}\prec\boldsymbol{a}$.

定义 2. 设 $\boldsymbol{x}_i$ 为群体 P_t 中的一个个体,用 $S(\boldsymbol{x}_i)$表示群体中 Pareto 劣于 $\boldsymbol{x}_i$ 的个体个数,称为 $\boldsymbol{x}_i$ 的强度值,即

$$S(\boldsymbol{x}_i)=\#\{\boldsymbol{x}_j|\boldsymbol{x}_j\in P_t \text{ 且 } \boldsymbol{x}_i\prec\boldsymbol{x}_j\},\tag{4}$$

其中#表示集合的基数.

强度指标 $S(\boldsymbol{x}_i)$反映了个体 $\boldsymbol{x}_i$ 在群体 P_t 中 Pareto 优于关系的强弱程度[5],若 $\boldsymbol{x}_i$ 的强度值大,则群体中劣于的个体多,若 $\boldsymbol{x}_i$ 的强度值小,则群体中劣于的个体少,从而我们可以根据强度值来对群体中的个体进行排序.对于强度值相等的情况,则比较它们违反约束的程度.下面是个体 Pareto 强度排序步骤:

(1) 由强度值公式(4)计算群体中每个个体的强度值,强度值大的个体为优;

(2) 对于强度值相同的个体,比较它们违法约束条件的程度 $\sum_{j=1}^{m}f_j(x)$, $\sum_{j=1}^{m}f_j(x)$ 值小的个体为优.

数值实验表明,仅仅使用上述 Pareto 强度排序的算法效果不稳定.究其原因,Pareto 强度是求向量$(s_1(\boldsymbol{x}),s_2(\boldsymbol{x}))$的最小值,若算法找到的最优点 $\boldsymbol{x}^*$不满足 $s_2(\boldsymbol{x}^*)=0$,则 $\boldsymbol{x}^*$就不是原问题的解.所以,在使用 Pareto 强度排序时,不仅要求$(s_1(\boldsymbol{x}),s_2(\boldsymbol{x}))$的最小值,还应使 $s_2(\boldsymbol{x})$趋于 0.下一节我们给出一个实现上述两个目标的演化算法.

3 Pareto 强度值遗传算法

3.1 实数编码遗传算法

演化算法,例如,演化策略(evolution strategy,简称 ES)、演化规划(evolutionary programming,简称 EP)和遗传算法(genetic algorithm,简称 GAs),都成功地应用于函数优化问题.演化策略和演化规划针对连续变量使用浮点编码,变异算子作为产生后代的首要算子;遗传算法则主要使用二进制编码,其产生后代的首要算子为杂交算子.然而,由于认识到二进制编码求解函数优化问题效果不理想,遗传算法也采用实数编码[6],称其为实数编码遗传算法(real-coded genetic algorithms,简称 RCGA),现在已有多种实数编码方法,如模拟二进制杂交(simulated binary crossover,简称 SBX)、混和杂交(blend crossover,简称 BLX)、单峰正态分布杂交(unimodal normal distribution crossover,简称 UNDX)[7]、单形杂交(simplex crossover,简称 SPX)[8]等.文献[9]对采用自适应变异的

演化策略和采用 UNDX 杂交算子的实数编码遗传算法进行了比较,结果显示,实数编码遗传算法在求解高维和多极值函数优化问题方面性能占优势.

对于实数编码遗传算法,重组算子的设计是一个重要方面.另外,群体替换模型的设计也在算法的性能上发挥着重要的作用.最小代数代沟(minimal generation gap,简称 MGG)模型是一个首先由 Satoh 提出,现在已被广泛使用的模型[10].它较好地均衡了算法的探索(exploration)和开发(exploitation)能力,将重组算子和选择算子按照以下列方式交叉进行:

(1) 从第 t 代群体 $P(t)$中随机地选取μ个父体;

(2) 对μ个父体使用重组算子产生λ个后代;

(3) 从群体中随机选取两个父体;

(4) 在所选的两个父体中,一个由λ个后代中最好的个体替换掉,另一个由剩下的$\lambda-1$ 个后代使用赌轮选择算子替换掉.

3.2 Pareto强度值遗传算法

利用 Pareto 强度值和改进的最小代数代沟(MGG)模型,我们下面给出求解约束条件的函数优化问题的 Pareto 强度值遗传算法:

(1) 初始化,群体规模为 N,置 t=0;

(2) 从第 t 代群体 $P(t)$中随机地选取μ个父体;

(3) 对μ个父体使用单形杂交算子(见第 3.3 节)产生λ个后代;

(4) 从λ个后代中选取两个后代,其中一个为λ个后代中 Pareto 强度值最大的个体,若强度值最大的个体不惟一,则比较它们违反约束条件的程度,程度小的当选;另一个父体为剩下的$\lambda-1$ 个后代中违反约束条件最小的个体;

(5) 重复步骤(2)~(4),直到选取 N 个后代为止,由这 N 个后代将群体 $P(t)$中的个体整体替换掉;

(6) 若满足停机条件则停机,否则 $t=t+1$,转步骤(2).

3.3 多父体单形杂交算子

多父体单形杂交算子使用 $n+1$ 个父体向量($\boldsymbol{x}_i,i=0,1,\ldots,n$)重组产生后代,这($n+1$)个向量形成 R^n 中的一个单形(simplex).将单形沿各个方向($\boldsymbol{x}_i-\boldsymbol{o}$)以一定的比例扩张(其中 $\boldsymbol{o}$ 是 $n+1$ 个向量的中心),从扩张后的单形中随机地取一点即为一个后代.例如,如图 1 所示,我们考虑二维空间中的 3 个向量,$\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\boldsymbol{x}^{(3)}$为 3 个向量,这 3 个向量形成一个单形,把这个单形以比例($1+\varepsilon$)向外扩张(ε称为扩张比率),令 $\boldsymbol{o}=1/3(\boldsymbol{x}^{(1)}+\boldsymbol{x}^{(2)}+\boldsymbol{x}^{(3)})$,$\boldsymbol{y}^{(j)}=(1+\varepsilon)(\boldsymbol{x}^{(j)}-\boldsymbol{o})(j=1,2,3)$,由 $\boldsymbol{y}^{(1)},\boldsymbol{y}^{(2)},\boldsymbol{y}^{(3)}$产生一个新单形,在新单形中随机地取一点 $\boldsymbol{z}$,$\boldsymbol{z}=k_1\cdot\boldsymbol{y}^{(1)}+k_2\cdot\boldsymbol{y}^{(2)}+k_3\cdot\boldsymbol{y}^{(3)}+\boldsymbol{o}$,其中 k_1,k_2,k_3 为[0,1]中的 3 个随机数,满足 $k_1+k_2+k_3=1$,$\boldsymbol{z}$ 即为一个三父体单形杂交算子产生的后代.

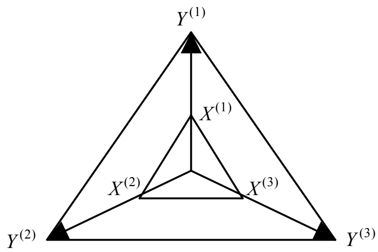

Fig.1 SPX with three parents in two-dimensional space
图 1 二维三父体单形杂交

4 数值实验

我们选取 5 个常见测试函数 $G1$~$G5$(见文献[4,11])来检验新算法.它们都是高维的带约束条件的函数优化问题,约束条件有等式约束、不等式约束以及两者的混合.

Minimize $G1(\boldsymbol{x})=3x_1+0.000001x_1^3+2x_2+(0.000002/3)x_2^3$,

Subject to $x_4-x_3+0.55\ge 0$, $-x_4+x_3+0.55\ge 0$,

$1000\sin(-x_3-0.25)+1000\sin(-x_4-0.25)+894.8-x_1=0$,

$1000\sin(x_3-0.25)+1000\sin(x_3-x_4-0.25)+894.8-x_2=0$,

$1000\sin(x_4-0.25)+1000\sin(x_4-x_3-0.25)+1294.8=0$,

$0\le x_i\le 1200$ $(i=1,2)$, $-0.55\le x_i\le 0.55$ $(i=3,4)$.

已知最优解为 $\boldsymbol{x}^*=(679.9453, 1026.067, 0.1188764, -0.3962336)$,

$G1(\boldsymbol{x}^*)=5126.4981$.

Minimize $G2(\boldsymbol{x}^*)=\mathrm{e}^{x_1x_2x_3x_4x_5}$,

Subject to $x_1^2+x_2^2+x_2^2+x_3^2+x_4^2+x_5^2-10=0$,

$x_2x_3-5x_4x_5=0$, $x_1^3+x_2^3+1=0$,

$-2.3\le x_i\le 2.3$ $(i=1,2)$, $-3.2\le x_i\le 3.2$ $(i=3,4,5)$,

已知最优解为 $\boldsymbol{x}^*=(-1.717143, 1.595709, 1.827247, -0.7636413, -0.763645)$,

$G2(\boldsymbol{x}^*)=0.0539498$.

Minimize $G3(\boldsymbol{x}^*)=(x_1-10)^2+5(x_2-12)^2+x_3^4+3(x_4-11)^2+10x_5^6+7x_6^2+x_7^4-4x_6x_7-10x_6-8x_7$,

Subject to $127-2x_1^2-3x_2^4-x_3-4x_4^2-5x_5\ge 0$, $282-7x_1-3x_2-10x_3^2-x_4+x_5\ge 0$,

$196-23x_1-x_2^2-6x_6^2+8x_7\ge 0$, $-4x_1^2-x_2^2+3x_1x_2-2x_3^2-5x_6+11x_7\ge 0$,

$-10.0\le x_i\le 10.0$ $(i=1,\dots,7)$,

已知最优解为 $\boldsymbol{x}^*=(2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, 1.594227)$,

$G3(\boldsymbol{x}^*)=680.6300573$.

Minimize $G4(\boldsymbol{x})=x_1+x_2+x_3$,

Subject to $-1+0.0025(x_4+x_6)\le 0$, $-1+0.0025(x_5+x_7-x_4)\le 0$,

$-1+0.01(x_8-x_5)\le 0$, $-x_1x_6+833.33252x_4+100x_1-83333.333\le 0$,

$-x_2x_7+1250x_5+x_2x_4-1250x_4\le 0$, $-x_3x_8+1250000+x_3x_5-2500x_5\le 0$,

$100\le x_1\le 10000$, $1000\le x_i\le 10000$ $(i=2,3)$, $10\le x_i\le 1000$ $(i=4,\dots,8)$,

已知最优解为 $\boldsymbol{x}^*=(579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, 395.5979)$,

$G4(\boldsymbol{x}^*)=7049.3307$.

Minimize $G5(\boldsymbol{x})=x_1^2+x_2^2+x_1x_2-14x_1-16x_2+(x_3-10)^2+4(x_4-5)^2+(x_5-3)^2+2(x_6-1)^2+5x_7^2+7(x_8-11)^2+2(x_9-10)^2+(x_{10}-7)^2+45$,

Subject to $105-4x_1-5x_2+3x_7-9x_8\ge 0$, $-3(x_1-2)^2-4(x_2-3)^2-2x_3^2+7x_4+120\ge 0$,

$-10x_1+8x_2+17x_7-2x_8\ge 0$, $-x_1^2-2(x_2-2)^2+2x_1x_2-14x_5+6x_6\ge 0$,

$8x_1-2x_2-5x_9+2x_{10}+12\ge 0$, $-5x_{12}-8x_2-(x_3-6)2+2x_4+40\ge 0$,

$3x_1-6x_2-12(x_9-8)^2+7x_{10}\ge 0$, $-0.5(x_1-8)^2-2(x_2-4)^2-3x_5^2+x_6+30\ge 0$,

$-10.0\le x_i\le 10.0$ $(i=1,\dots,10)$,

已知最优解为 $\boldsymbol{x}^*=(2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$,

$G5(\boldsymbol{x}^*)=24.3062091$.

数值实验在 MATLAB 中完成(若需 MATLAB 代码,可与我们联系).在我们的计算中,群体规模取为 50,多父体单形杂交的父体个数μ取为 $n+1$(n 为自变量的个数),每次产生后代个数λ取为 30,多父体单形杂交中的ε称为扩张比率,ε取为 3~6.对于问题 $G1$~$G3$(自变量的维数 4~7 维),我们运行 500 代;对于问题 $G4$~$G5$(自变量的维数 8~10 维),我们运行 1 000 代.对每个问题在相同条件下独立运行 20 次,记录其最好结果、最差结果和平均结果.为了对比,我们将新算法运算结果(记为 ZW)与另外两个较新的演化算法进行比较,它们分别为文献[4]中的随机排序法(记为 RY)和文献[11]中的同态映射法(记为 KM).比较的结果见表 1.

对于问题 $G1$,约束条件既有等式约束又有不等式约束.文献[11]中的算法“没有得到合格的解”;文献[4]中的算法把约束条件 $h_j(\boldsymbol{x})=0$ 由不等式约束 $|h_j(\boldsymbol{x})| \le \delta$ 来代替,δ 取为 0.000 1.我们用新算法独立运行 20 次,有 18 次找到最优值 5 126.498 109 59,与边界的距离都小于 10^{-11},比如,我们找到的一个典型的最优点为 $\boldsymbol{x}^*=(679.94527895320,1026.06717631112,0.11887639366,-0.39623353936)$, 最优点离边界的距离为 $s_2(\boldsymbol{x}^*)=2.273736754432\times10^{-13}$.

对于问题 $G2$,约束条件为等式约束,与 $G1$ 的情况类似,我们找到的一个典型的最优点为(−1.7171435596, 1.59570967771,−1.8272457729,0.76364307213,−0.7636430866),最优值为 0.053 949 847 770 27,最优点离边界的距离为 $2.220446049250\times10^{-15}$.

Tabel 1 Comparison among ZW (new method), RY (in Ref.[4]) and KM (in Ref.[11]) (20 independent runs)

表 1 新算法与文献[4](RY)和文献[11](KM)中算法的比较(独立重复 20 次)

fcn		$G1$	$G2$	$G3$	$G4$	$G5$
Optimal		5 126.498 1	0.053 949 8	680.630 057 3	7 049.330 7	24.306 209 1
Best	ZW	5 126.498 11	0.053 949 831	680.630 057 3	7 049.248 020 5	24.306 209 068
	RY	5 126.497	0.053 957	680.630	7 054.316	24.307
	KM	–	0.054	680.91	7147.9	24.620
Mean	ZW	5 126.526 54	0.053 950 257	680.630 057 3	7 051.287 429 2	24.325 487 652
	RY	5 128.881	0.057 006	680.656	7 559.192	24.374
	KM	–	0.064	681.16	8 163.6	24.826
Worst	ZW	5 127.156 41	0.053 972 292	680.630 057 3	7 058.235 358 5	24.362 999 860
	RY	5 142.472	0.216 915	680.763	8 835.665	24.642
	KM	–	0.557	683.18	9 659.3	25.069

对于问题 $G3$,新算法独立重复运行 20 次全部找到最优解 680.630 057 374 402.

对于问题 $G4$,文献[4,11]引述的已知最优解为 7 049.330 7,我们找到了新的最优解 7 049.248 020 528 77,最优点为(579.30643727677, 1359.97061959552, 5109.97096365681, 182.01767893950, 295.60116145374, 217.98232106050, 286.41651748576, 395.60116145374).

从表 1 可以看出,新算法实验结果出乎意料地好.对于测试函数 $G1$~$G5$,无论是等式约束还是不等式约束,新算法的运算结果在最好结果、平均结果和最差结果诸方面都比文献[4]和文献[11]中算法的结果要好,这充分地显示了新算法的通用性、有效性和稳健性.

需要说明的是,新算法的实现过程对于单形杂交算子中的扩张比率 ε 较为敏感,在我们测试的几个问题中,ε 取 3~6 效果较好.

5 小 结

本文提出了处理约束优化问题的新方法——基于 Pareto 强度值的实数编码遗传算法.该方法将约束优化问题转换成两目标优化问题,其中一个为原问题的目标函数,另一个为违反约束条件的程度函数.利用 Pareto 强度值和最小代数代沟模型设计出新的实数编码遗传算法.数值实验显示新算法是一种便于实现、通用性强、高效稳健的方法.下一步的工作我们将用大量的实际问题进一步检验新方法,改进算法的编码,以提高算法的性能,并将算法推广到约束组合优化问题.

References:

[1] Michalewicz Z, Schoenauer M. Evolutionary algorithms for constrained parameter optimization problems. Evolutionary Computation, 1996,4(1):1~32.

[2] Michalewicz Z. Genetic algorithms, Numerical optimization and constraints. In: Eshelman LJ, ed. Proceedings of the 6th International Conference on Genetic Algorithms. San Mateo: Morgan Kaufmann Publishers, 1995. 151~158.

[3] Deb K. An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering, 2000,186(2-4):311~338.

[4] Runarsson TP, Yao X. Stochastic ranking for constrained evolutionary optimization. IEEE Transactions on Evolutionary Computation, 2000,4(3):284~294.

[5] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation, 1999,3(4):257~271.

[6] Beyer H-G, Deb K. On self-adaptive features in real-parameter evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 2001,5(3):250~270.

[7] Ono I, Kita H, Kobayashi S. A robust real-coded genetic algorithm using unimodal normal distribution crossover augmented by uniform crossover: effects of self adaptation of crossover probabilities. In: Banzhaf W, Daida J, Eiben E, eds. GECCO'99: Proceedings of the Genetic and Evolutionary Computation Conference. San Mateo: Morgan Kaufmann Publishers, 1999. 496~503.

[8] Tsutsui S, Yamamura M, Higuchi T. Multi-Parent recombination with simplex crossover in real coded genetic algorithms. In: Banzhaf W, Daida J, Eiben E, eds. GECCO'99: Proceedings of the Genetic and Evolutionary Computation Conference. San Mateo: Morgan Kaufmann Publishers, 1999. 657~664.

[9] Kita H. A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms. Evolutionary Computation, 2001,9(2):223~241.

[10] Deb K, Joshi D, Anand A. Real-Coded evolutionary algorithms with parent-centric recombination. Technical Report, KanGAL Report No.2001003, Kanpur: Indian Institute of Technology, 2001.

[11] Koziel S, Michalewicz Z. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Evolutionary Computation, 1999,7(1):19~44.

敬 告 作 者

《软件学报》创刊以来,蒙国内外学术界厚爱,收到许多高质量的稿件,其中不少在发表后读者反映良好,认为本刊保持了较高的学术水平.但也有一些稿件因不符合本刊的要求而未能通过审稿.为了帮助广大作者尽快地把他们的优秀研究成果发表在我刊上,特此列举一些审稿过程中经常遇到的问题,请作者投稿时尽量予以避免,以利大作的发表.

1. 读书偶有所得,即匆忙成文,未曾注意该领域或该研究课题国内外近年来的发展情况,不引用和不比较最近文献中的同类结果,有的甚至完全不列参考文献.

2. 做了一个软件系统,详尽描述该系统的各个方面,如像工作报告,但采用的基本上是成熟技术,未与国内外同类系统比较,没有指出该系统在技术上哪几点比别人先进,为什么先进.一般来说,技术上没有创新的软件系统是没有发表价值的.

3. 提出一个新的算法,认为该算法优越,但既未从数学上证明比现有的其他算法好(例如降低复杂性),也没有用实验数据来进行对比,难以令人信服.

4. 提出一个大型软件系统的总体设想,但很粗糙,而且还没有(哪怕是部分的)实现,很难证明该设想是现实的、可行的、先进的.

5. 介绍一个现有的软件开发方法,或一个现有软件产品的结构(非作者本人开发,往往是引进的,或公司产品),甚至某一软件的使用方法.本刊不登载高级科普文章,不支持在论文中引进广告色彩.

6. 提出对软件开发或软件产业的某种观点,泛泛而论,技术含量少.本刊目前暂不开办软件论坛,只发表学术文章,但也欢迎材料丰富,反映现代软件理论或技术发展,并含有作者精辟见解的某一领域的综述文章.

7. 介绍作者做的把软件技术应用于某个领域的工作,但其中软件技术含量太少,甚至微不足道,大部分内容是其他专业领域的技术细节,这类文章宜改投其他专业刊物.

8. 其主要内容已经在其他正式学术刊物上或在正式出版物中发表过的文章,一稿多投的文章,经退稿后未作本质修改换名重投的文章.

本刊热情欢迎国内外科技界对《软件学报》踊跃投稿.为了和大家一起办好本刊,特提出以上各点敬告作者.并且欢迎广大作者和读者对本刊的各个方面,尤其是对论文的质量多多提出批评建议.