第 22卷 第 4期
2006年 8月

哈尔滨商业大学学报(自然科学版)
Journal of Harbin University of Commerce (Natural Sciences Edition)

Vol 22 No. 4
Aug 2006

MATLAB中用遗传算法求解约束非线性规划问题

王 勇

(哈尔滨商业大学 基础科学学院,黑龙江 哈尔滨 150028)

摘 要:约束非线性规划问题的求解往往是运筹学中的 NP问题,利用 MATLAB中的遗传算法工具箱中的函数方便、快捷的求得了两个实例的最优解,进一步指出了遗传算法与传统的最优化算法的区别.

关键词:遗传算法;约束非线性规划;MATLAB

中图分类号:O221 **文献标识码**:A **文章编号**:1672 - 0946(2006)04 - 0116 - 02

Solution of optimization with nonliear constraints programming by genetic alogorithm in MATLAB

WANG Yong

(School of Basic Science, Harbin University of Commerce, Harbin 150028, China)

Abstract: The solution process to optimization nonlinear constraints programming often concerns NP problems in operations research. This paper employs the functions of genetic algorithm in MATLAB tool kit, and gets swiftly and conveniently two optional solutions in two cases concerned. And points out the differences between the genetic algorithm and the traditional optimal algorithm.

Key words: genetic algorithm; optimization nonliear constraints programming; MATLAB

约束非线性规划问题是运筹学中的一个重要分支,在经济、管理、计划,以及军事、生产自动化方面有着重要应用,但它的求解往往比较复杂.而遗传算法是一个新兴的方法,1975年 Hollland在他的著作《Adaptation in Natural and Artifical Systems》中首次提出遗传算法,其基本思想是从一个代表最优化问题解的一组初值开始进行搜索,这组解称为一个种群,种群有一定数量、通过基因编码的个体组成,其中每一个个体称为染色体,不同个体通过染色体的复制、交叉、变异又生成新的个体,依照适者生存的规则,个体也在一代一代进化,通过若干代的进化最终得出条件最优的个体.很快就将其应用于求解非线性最优化问题,在著名的数学软件 MATLAB中有一个有效地工具箱 - -遗传算法工具箱,本文即使用遗传算法工具箱为主要工具,求解约束非线性规划问题.

1 遗传算法的一般步骤

1)选择 N 个个体构成初始种群 P_0[1],并求出种群内各个个体的函数值.染色体用实数数组来表示,种群可由随机数生成函数建立.在 MATLAB中使用遗传算法求解函数 gaopt(),则会自动生成所需的初始种群 P_0.

2)设值代数为 $i=1$,即设置为第一代.

3)计算选择函数的值,所谓选择即通过概率的形式从种群中选择若干个个体的方式.遗传算法工具箱提供了 3个选择函数:roulette()实现了轮盘选择算法,normGeomSelect()函数实现了归一化几何选择方法,tournSelect()实现了锦标赛形式的选择方式,本文使用 normGeomSelect()函数确定选

收稿日期:2006 - 04 - 29.

作者简介:王 勇(1972 -),男,硕士,教师,研究方向:运筹学与控制论.

择函数值.

4)通过染色体个体基因的复制、交叉、变异等创造新的个体,构成新的种群 P_{i+1},其中复制、交叉、变异都有相应的 MATLAB函数,可使手工计算量大大减少.

5) $i = i + 1$,若终止条件不满足,则转到步骤 3)继续进化处理.

2　约束非线性规划问题

约束非线性规划问题的一般描述是

$$\min_{x\ s.t\ G(x)\le 0} f(x)$$

其中: $x = [x_1, x_2, \cdots, x_n]^T$. 为求解方便,约束条件还可以进一步细化为线性等式约束、线性不等式约束、x变量的上下界向量,还允许一般非线性函数的等式和不等式约束[2],这时原规划问题可以改写成

$$\min f(x)\quad x\ s.t\begin{cases} Ax \le B \\ A_{eq}x = B_{eq} \\ x_m \le x \le x_M \\ C(x) \le 0 \\ C_{eq}(x) = 0 \end{cases}$$

MATLAB最优化工具箱中提供了一 fmincon()函数,专门用于求解各种约束下的最优化问题. 该函数的调用格式是

$[x, f_{opt}, flag, c] = fmincon(F, x_0, A, B, A_{eq}, B_{eq}, x_m, x_M, CF, OPT, p_1, p_2, \cdots)$

其中: F为给目标函数写的 M 函数或 inline()函数, x_0 为初始搜索点, CF为给非线性约束函数编写的 M 函数, OPT为控制选项.

该函数可以处理很多非线性规划问题的求解. 但是,注意到搜索函数需给出初值,这样就导致对不同的初值可能得出不同的搜索结果,很难得出全局最优解,即这种传统的寻优方式不一定能得出满意的结果.

3　两个实例

考虑遗传算法与传统最优化算法比较主要有以下几点不同:

a) 遗传算法从一个种群开始对问题的最优解进行并行搜索,更利于全局最优化解的搜索,需注意的是它需要指出各个变量的范围.

b) 遗传算法不依赖导数信息或其他辅助信息来进行最优解搜索,而只由目标函数和对应于目标函数的适应度水平来确定搜索方向.

c) 遗传算法采用的是概率性规则而不是确定性规则,所以每次得出的结果不一定完全相同,有时甚至会有较大差异,这就需要我们增加搜索的代数.

遗传算法最优化工具箱中有一个 gaopt()函数,它的调用极其简单. 即使对遗传算法理解不多,只须利用 MATLAB语言描述出目标函数,就可以得出最优解,需要注意的是, gaopt()函数能求解的是问题的最大化问题,所以在编写目标函数时应加以留意. 下面用两个实例来说明该函数的应用.

例 1.
$$\min F = x_1^3 + 2x_2^2x_3 + 2x_3$$
$$s.t\begin{cases} x_1^2 + x_2 + x_3^2 = 4 \\ x_1^2 - x_2 + 2x_3 \le 2 \\ x_1, x_2, x_3 \ge 0 \end{cases}$$

按照遗传算法工具箱编写如下函数,描述最优化问题的目标函数,为使其转化为最大化问题,目标函数两端同乘.

```
function [sol, f] = c10mga3(sol, options)
x = sol(1:5);
f = -x(1)^3 - 2x(2)^2*x(3) - 2x(3)
```

使用 函数 gaopt(),并设定自变量的求解范围为 $0 \le x_i \le 5$, $i = 1, 2, 3$,则可由下面的函数求解最优化问题,

```
≫ [a, b, c] = gaopt([0,5; 0,5; 0,5],
'c10mga3'); a, c
a = 0.000 0, 4.000 0, 0.000 0, 0.000 0,
```

即 $x_1 = 0.000\ 0$, $x_2 = 4.000\ 0$, $x_3 = 0.000\ 0$, $F = 0.000\ 0$

在该问题中设定搜索代数为 100,就得出了最优解.

例 2.
$$\max f(x) = -2x_1^2 + 2x_1x_2 - 2x_2^2 + 4x_1 + 6x_2$$
$$s.t\begin{cases} 2x_1 - x_2 \le 0 \\ x_1 + 5x_2 \le 5 \\ x_1, x_2 \ge 0 \end{cases}$$

编写 M 函数如下:

```
function[sol, f] = c10mga3(sol, options)
x = sol(0:1);
f = -2x(1)^2 + 2x(1)x(2) - 2x(2)^2 + 4x(1) + 6x(2)
```

```
≫ [a, b] = gaopt([0,1; 0,1], 'c10mga3'); a, c
```

求得 $x_1 = 0.658$, $x_2 = 0.868$, $f(x_1, x_2) = 6.613$

这两个例子,一个求最小化,一个求最大化,利用

(下转 122页)

$$\begin{pmatrix} b_{11} & & & & & \\ & \ddots & & & & \\ & & b_{1n_1} & & & \\ & & & \ddots & & \\ & & & & b_{s1} & \\ & & & & & \ddots \\ & & & & & & b_{sn_s} \end{pmatrix} \cdot \begin{pmatrix} c_{11} & & & & & & \\ & \ddots & & & & & \\ & & c_{1(n_1-1)} & & * & & \\ & & & c^{(1)} & & & \\ & & & & \ddots & & \\ & & & & & c_{s1} & \\ & & & & & & \ddots \\ & & & & & & & c_{s(n_s-1)} \\ & & & & & & & & c^{(s)} \end{pmatrix} \cdot \begin{pmatrix} P_1^{-1} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & P_s^{-1} \end{pmatrix}$$

令 $P = \begin{pmatrix} P_1 & & \\ & \ddots & \\ & & P_s \end{pmatrix}$,则

$$P^{-1} = \begin{pmatrix} P_1^{-1} & & \\ & \ddots & \\ & & P_s^{-1} \end{pmatrix}.$$

$$L = \begin{pmatrix} b_{11} & & & & & \\ & \ddots & & & & \\ & & b_{1n_1} & & & \\ & & & \ddots & & \\ & & & & b_{s1} & \\ & & * & & & \ddots \\ & & & & & & b_{sn_s} \end{pmatrix}$$

$$U = \begin{pmatrix} c_{11} & & & & & & \\ & \ddots & & & & & \\ & & c_{1(n_1-1)} & & * & & \\ & & & c^{(1)} & & & \\ & & & & \ddots & & \\ & & & & & c_{s1} & \\ & & & & & & \ddots \\ & & & & & & & c_{s(n_s-1)} \\ & & & & & & & & c^{(s)} \end{pmatrix}$$

所以 $A = (PLP^{-1})(PUP^{-1})$成立.

参考文献:

[1] SOUROUR A. Factorization Theorem for matrices[J]. Linear Maltilinear Algebra, 1996, 19: 141 - 147.

[2] JOHN R. S. Introduction to Algebraic K - Theory[M]. New York: Chapman and Hall, 1981.

(上接 117页)

MATLAB都得到了最优解,其中省略了计算机生成的 c参数的单独显式(因其可读性不强),就这两个问题而言,从第 40代起就可得出较精确的结果,通过 100代的搜索,所得出的结果是具有高精度的的解. 可见,在条件允许的情况下,可将搜索过程的代数设置的大一些.

4 结 语

由此可见,从最优化问题求解的方法看,最优化工具箱中的函数一次只能搜索到一个解,对非凸性问题来说往往可能找到一个局部最优值,而用遗传算法则可以同时从一组初值点出发,有可能找到更好的局部优值甚至是全局最优值. 在实际求解问题中,为改善求解的精度和速度,可以考虑这样的策略,先用遗传算法初步定出较好最优值所在的大概位置,然后以该位置为初值,调用最优化工具箱中函数快速、准确的求出该最优值.

参考文献:

[1] 薛定宇,陈阳泉. 高等应用数学问题的 MATLAB求解[M]. 北京:清华大学出版社, 2004.

[2] 康加福,汪定伟. 一种求解非线性规划问题的改进遗传算法[J]. 东北大学学报, 1997, 18(5): 124 - 128.