应用数学和力学编委会编
重庆出版社出版

文章编号: 1000_0887(2005) 10_1168_07

多变量、多约束连续或离散的非线性规划的一个通用算法*

孙焕纯[1], 王跃方[1], 柴 山[2]

(1. 大连理工大学 工程力学系, 大连 116024;
2. 山东理工大学 机械学院, 山东 淄博 255012)

(我刊原编委孙焕纯来稿)

摘要: 利用目标函数对约束函数关于设计变量的一阶微分或差分之比, 给出了一个求解非线性规划的通用算法· 不论变量和约束有多少, 也不论变量是连续的还是离散的, 这一算法都比较有效, 尤其对离散非线性规划更有效· 该方法是一种搜索法, 勿需解任何数学方程, 只需要计算函数值以及函数对变量的偏微分或差分值· 许多数值例题和运筹学中一些经典问题, 如 1) 一、二维的背包问题; 2) 一、二维资源分配问题; 3) 复合系统工作可靠性问题; 4) 机器负荷问题等, 经用此法求解验证均较传统方法更有效和可靠· 该方法的主要优点是: 1) 不受问题的规模限制; 2) 只要在可行域(集) 内存在目标函数和约束函数及其一阶导数或差分的值, 肯定可以搜索到最优的解, 没有不收敛和不稳定的问题·

关 键 词: 连续或离散非线性规划; 搜索算法; 相对微分/ 差分法
中图分类号: O221.2; 242.23 文献标识码: A

引 言

对非线性规划, 已有大量的研究成果问世, 但多变量、多约束的非线性规划, 尤其是离散非线性规划的有效算法极少, 通用的算法更难得一见· 对于连续非线性规划, 已有的多种算法虽然各有所长[1], 但在解决多约束、多变量工程优化问题时, 计算工作量都不小· 特别是花费在加快收敛速度上的工作量常常很大· 从实用上看, 它们往往不如一些看起来笨拙, 但计算过程简便的算法的效果好· 原因是后者省掉了花在提高收敛速度上的额外的工作量, 收敛速度可能反而更快·

本文提出了这样一个算法, 我们称之为相对微分(差分) 法· 它是一种搜索法, 适合于求解多变量、多约束的非线性规划问题· 此法勿需解任何数学方程, 只需要计算函数值以及函数对变量的一阶偏微分或差分值, 即可开展搜索过程· 对多个实例的计算表明, 本法可以有效地求出非线性规划问题的最优解, 且收敛速度比较理想·

* 收稿日期: 2003_08_25; 修订日期: 2005_05_28
基金项目: 国家自然科学基金资助项目(10002005; 10421002)
作者简介: 孙焕纯(1927—), 男, 辽宁大连人, 教授, 博导(联系人. Tel: + 86_411_84709671; Fax: + 86_411_84708390; E_mail: yfwang@dlut.edu.cn)·

1　非线性规划的数学模型

熟知的非线性规划的模型表示为下面的形式:

求　$\boldsymbol{X} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$,

使

$$f(\boldsymbol{X}) \rightarrow \min, \quad \text{s.t. } h_j(\boldsymbol{X}) = 0, \quad g_{j'}(\boldsymbol{X}) \leqslant 0$$
$$(j = 1, \cdots, e(< n), j' = 1, \cdots, m), \tag{1}$$

其中目标函数$f(\boldsymbol{X})$、约束函数 $h_j(\boldsymbol{X})$ 和$g_{j'}(\boldsymbol{X})$ 是设计变量$\boldsymbol{X}$的任意连续或离散函数; n、e和m 分别是设计变量、等式约束和不等式约束函数的个数; $h_j(\boldsymbol{X})$ 和 $g_{j'}(\boldsymbol{X})$ 可以是线性的, 也可以是非线性的. 为消除各约束的量纲、量级的影响, 便于选择最严约束, 对各约束函数均做无量纲归一化处理, 即取$h_j(\boldsymbol{X})$ 和 $g_{j'}(\boldsymbol{X})$ 表达式中常数项的绝对值为 1. 如前所述, 当变量数和约束数较多时, 规划问题(1) 的计算工作量往往很大, 很难有效地解决实际优化问题.

2　解非线性规划的相对微分(差分) 法[2]

当存在多个等式约束时, 在搜索寻优前先用消去法消去多余的一个等式约束, 同时消去相同数目的变量. 记该约束为 $h(\boldsymbol{X})$, 仍用 $\boldsymbol{X}$ 表示消去后的设计变量, 对规划(1), 若设计变量连续, 定义如下的相对微分

$$\beta_i = \frac{\partial f(\boldsymbol{X})}{\partial x_i}\mathrm{d}x_i \Bigg\backslash \frac{\partial G(\boldsymbol{X})}{\partial x_i}\mathrm{d}x_i = \frac{\partial f(\boldsymbol{X})}{\partial x_i}\Bigg\backslash \frac{\partial G(\boldsymbol{X})}{\partial x_i} \qquad (i = 1, \cdots, n'). \tag{2}$$

若 $\boldsymbol{X}$ 为离散变量, 则定义相对差分:

$$\beta_i = \frac{\Delta f(\boldsymbol{X})}{\Delta x_i}\Delta x_i \Bigg\backslash \frac{\Delta G(\boldsymbol{X})}{\Delta x_i}\Delta x_i = \frac{\Delta f(\boldsymbol{X})}{\Delta x_i}\Bigg\backslash \frac{\Delta G(\boldsymbol{X})}{\Delta x_i} \qquad (i = 1, \cdots, n'), \tag{3}$$

其中 $n' = n + 1 - e$ 为消去后的变量个数. 式(2)、(3) 中的 $G(\boldsymbol{X})$ 为当前设计点 $\boldsymbol{X}$ 处的最严约束. $G(\boldsymbol{X}) = \max\limits_j(g_{j'}(\boldsymbol{X}) \leqslant 0, h(\boldsymbol{X}))$. 对含隐式约束的连续变量优化问题, 约束函数对 $\boldsymbol{X}$ 的导数往往不易求得. 此时可用差分方法, 按式(3) 计算各 β_i, 再挑选它们的最小、最大值, 分别记为:

$$\beta_l = \min_i \beta_i, \quad \beta_b = \max_i \beta_i. \tag{4}$$

应当说明, 在离散规划中亦可将离散函数视为连续函数, 采用式(2) 计算 β_i, 可能更省时. 因为计算时, 只是利用 β_i 的正负号判断变量沿标架的搜索方向, 利用其量级作为选取优化变量和每轮步数的参考数据, 并不需要求它的精确数值.

2.1　相对微分(差分)算法的优化过程

下面给出相对微分(差分)算法的主要思想. 记 K 为优化的搜索次数, 算法的流程是:

Ⅰ. 令 $K = 0$, 在 $h(\boldsymbol{X}) = 0$ 上选取可行初始设计点 $\boldsymbol{X}^K$ (上角标字母代表搜索次数, 下同).

Ⅱ. 计算$f(\boldsymbol{X}^K)$ 和 $g_{j'}(\boldsymbol{X}^K)$, $j' = 1, \cdots, m$, $h(\boldsymbol{X}^K) = 0$, 并得到最严约束 $G(\boldsymbol{X}^K)$. 会有两种情况: 1) $G(\boldsymbol{X})$ 为不等式约束, 此时必有 $G(\boldsymbol{X}) > 0$; 2) $G(\boldsymbol{X}) = h(\boldsymbol{X})$, 此时必有$g_{j'}(\boldsymbol{X}) \leqslant 0$($j' = 1, \cdots, m$).

Ⅲ 根据 $\boldsymbol{X}$ 的连续或离散性质, 按式(2) 或(3) 计算 β_i, $i = 1, \cdots, n'$, 按式(4) 计算相对微分(差分) β_l、β_b, 得到它们的变量序号 l、b. x_l、x_b 即为下一次搜索时需改变的设计变量.

当 $G(\boldsymbol{X}^K)$ 为不等式约束时, 只能是 $G(\boldsymbol{X}^K) > 0$. 设以 β_i^g 表示β_i, 此时只有下述一种情况, 即所有的 $\beta_i^g < 0$, 其他任一 $\beta_i^g > 0$ 的情况是不存在的, 否则 $G(\boldsymbol{X}^K)$ 就是无效约束了. 为了使

$G(\boldsymbol{X}^K)$ 减少, 将设计点引入可行域(集) 内, 并且使目标函数增加得尽可能少• 选择:

$$\beta_b = \max_i(\beta_i),\quad \beta_{b'} = \max_{i\neq b}(\beta_i)$$

当 $(\partial h/\partial x_b)|_{X=X^K}\cdot(\partial h/\partial x_{b'})|_{X=X^K} > 0$ 则取 $dx_b\cdot dx_{b'} < 0$ 以确保 $h(\boldsymbol{X}^{K+1}) = 0$; 当 $(\partial h/\partial x_b)|_{X=X^K}\cdot(\partial h/\partial x_{b'})|_{X=X^K} < 0$ 则取 $dx_b\cdot dx_{b'} > 0$ 以确保 $h(\boldsymbol{X}^{K+1}) = 0$•

首先根据 $(\partial G/\partial x_b)|_{X=X^K}\cdot dx_b^K < 0$ 使 $G(\boldsymbol{X}^K)$ 减少, 适当定出 dx_b^K, 得到 $x_b^{K+1} = x_b^K + dx_b^K$, 代入 $h(\boldsymbol{X}^{K+1}) = 0$ 求出 $x_{b'}^{K+1} = x_{b'}^K + dx_{b'}^K$• 当 $G(\boldsymbol{X}^K) = h(\boldsymbol{X}^K) = 0$ 时, 必有 $g_{j'}(\boldsymbol{X}^K) \leqslant 0$ ($j' = 1, \cdots, m$), 这时存在3种可能情况(以 β_i^h 表示 β_i), 这就是在等式约束边界上搜索, 详见2.2节• 在搜索过程中, 可能出现不等式约束变成最严约束的情况, 那么就按 $G(\boldsymbol{X}^K) > 0$ 继续搜索• 直到满足所有约束条件, 即 $|G(\boldsymbol{X}^K)| < \varepsilon$, ε 为给定的误差限值, 且继续搜索不到更优值为止• 转至Ⅳ•

Ⅳ. 迭代终止, 优化结束•

在步骤Ⅲ中, 可能出现下面2种情况, 需要特别地加以说明•

1) 若在 $\boldsymbol{X}^K$ 处有两个约束的值相同并且都是最大值, 可取其中的任意一个为最严约束, 搜索1次后, 再重新比较, 确定新的最严约束•

2) 若 $G(\boldsymbol{X})$ 为不等式约束, 当出现某一 $\beta_i > 0$ 时, 因为优化 x_i 目标函数减少, 约束函数也减少, 不起约束作用• 此时回到 $G(\boldsymbol{X}^K) = h(\boldsymbol{X}^K) = 0$ 的边界上继续搜索, 如在搜索过程中出现某一不等式约束大于零, 则令其为新的最严约束 $G(\boldsymbol{X}^K) > 0$, 继续搜索•

2.2 等式约束边界上的搜索法

考虑下面的非线性规划问题:

$$\min f(\boldsymbol{X}),\quad \text{s.t. } h(\boldsymbol{X}) = 0,\quad g_j(\boldsymbol{X}) \leqslant 0 \qquad (j = 1, \cdots, m) \tag{5}$$

设当前设计点为 $\boldsymbol{X}^K$, 记等式约束的相对微分(差分) 为 β_i^h, $i = 1, 2, \cdots, n$, 根据 β_i^h 的正负号, 分以下3种情况讨论•

1) 全部的 $\beta_i^h < 0$

先在 $h(\boldsymbol{X}^K) = 0$ 上搜索, 选择 $\beta_l^h = \min\limits_i \beta_i^h$, $\beta_b^h = \max\limits_i \beta_i^h$, 对应的变量分别为 x_l^K 和 x_b^K• 如果 $(\partial h/\partial x_l)|_{x_l=x_l^K} > 0$, 取 $dx_l > 0$; 反之, 若 $(\partial h/\partial x_l)|_{x_l=x_l^K} < 0$, 则取 $dx_l < 0$, 使 $f(\boldsymbol{X}^K)$ 减少得多些• 不难看出, 此时必有 $(\partial h/\partial x_b)|_{x_b=x_b^K}\cdot dx_b < 0$, 使 $h(\boldsymbol{X}^{K+1}) = 0$ 而 $f(\boldsymbol{X}^K)$ 增加得最少• 由于 $h(\boldsymbol{X}) \approx h(\boldsymbol{X}^K) + (\partial h/\partial x_l)|_{x_l=x_l^K}dx_l + (\partial h/\partial x_b)|_{x_b=x_b^K}dx_b$, 为保持 $h(\boldsymbol{X}^{K+1}) = 0$, 应使下式成立:

$$dx_b = -[h(\boldsymbol{X}^K) + (\partial h/\partial x_l)|_{x_l=x_l^K}\cdot dx_l]/(\partial h/\partial x_b)|_{x_b=x_b^K} \tag{6}$$

给定 dx_l 后, 即可确定 dx_b, 得到后继点 $\boldsymbol{X}^{K+1}$• 如果步长 dx_l 取得较大, 按上式计算的 dx_b 一般不能保证 $h(\boldsymbol{X}^{K+1}) = 0$, 此时应令 $x_l^{K+1} = x_l^K + dx_l$, 代入 $h(\boldsymbol{X}^{K+1}) = 0$ 后求出 x_b^{K+1}•

以上讨论的是连续规划中步长的选择方法• 对于离散规划问题, 只需将上一段及式(6)中的偏导数 $\partial h/\partial x_l$ 和 $\partial h/\partial x_b$ 换成差分 $\Delta h/\Delta x_l$ 和 $\Delta h/\Delta x_b$ 即可•

2) 全部的 $\beta_i^h > 0$

和情况1) 类似, 不同的是: 为使 $h(\boldsymbol{X}^K) = 0$, 调整 x_b 使 $f(\boldsymbol{X}^K)$ 减少得多些, 调整 x_l 使 $f(\boldsymbol{X}^K)$ 增加得少些• 方法是先给定 dx_b, 代入 $h(\boldsymbol{X}^K) = 0$ 求出 dx_l• 首先适当地定出 dx_b^K, 求出 $x_b^{K+1} = x_b^K + dx_b^K$, 再代入 $h(\boldsymbol{X}^{K+1}) = 0$, 求得 $x_l^{K+1} = x_l^K + dx_l^K$•

3) β_i^h 的符号有正也有负

记 β_l^h 和 β_b^h 为所有 β_i^h 中的最小(负)值和最大(正)值•　此时, x_l 和 x_b 的变化可同时使 $f(\boldsymbol{X}^K)$ 减少•　关于 x_l 和 x_b 的变化应按下述条件选取 $\mathrm{d}x_l$ 和 $\mathrm{d}x_b$:

$$\text{当}\left.\frac{\partial h}{\partial x_l}\right|_{\boldsymbol{X}=\boldsymbol{X}^K}\cdot\left.\frac{\partial h}{\partial x_b}\right|_{\boldsymbol{X}=\boldsymbol{X}^K}<0,\ \text{必须有}\ \mathrm{d}x_l\cdot\mathrm{d}x_b>0\ \text{且}$$

$$\left.\frac{\partial h}{\partial x_b}\right|_{\boldsymbol{X}=\boldsymbol{X}^K}\cdot\mathrm{d}x_b<0,\ \text{或}\left.\frac{\partial h}{\partial x_l}\right|_{\boldsymbol{X}=\boldsymbol{X}^K}\cdot\mathrm{d}x_l>0;$$

$$\text{当}\left.\frac{\partial h}{\partial x_l}\right|_{\boldsymbol{X}=\boldsymbol{X}^K}\cdot\left.\frac{\partial h}{\partial x_b}\right|_{\boldsymbol{X}=\boldsymbol{X}^K}>0,\ \text{必须有}\ \mathrm{d}x_l\cdot\mathrm{d}x_b<0\ \text{且}$$

$$\left.\frac{\partial h}{\partial x_b}\right|_{\boldsymbol{X}=\boldsymbol{X}^K}\cdot\mathrm{d}x_b<0,\ \text{或}\left.\frac{\partial h}{\partial x_l}\right|_{\boldsymbol{X}=\boldsymbol{X}^K}\cdot\mathrm{d}x_l>0\text{•}$$

这样可确保 $h(\boldsymbol{X}^{K+1})=0$, 且 $f(\boldsymbol{X}^{K+1})$ 减少得最多、最快•　首先若 $|\beta_l|<\beta_b$, 适当地定出 $\mathrm{d}x_b^K$, 求得 $x_b^{K+1}=x_b^K+\mathrm{d}x_b^K$, 再代入 $h(\boldsymbol{X}^{K+1})=0$, 求得 $x_l^{K+1}=x_l^K+\mathrm{d}x_l^K$; 若 $|\beta_l|>\beta_b$, 则先适当地定出 $\mathrm{d}x_l^K$, 求得 $x_l^{K+1}=x_l^K+\mathrm{d}x_l^K$, 再代入 $h(\boldsymbol{X}^{K+1})=0$, 求得 $x_b^{K+1}=x_b^K+\mathrm{d}x_b^K$•

在上述3种情况下, 求得 x^{K+1} 之后, 继续下一次搜索•

搜索类算法的一个关键问题是如何选择恰当的步长•　本法虽然在选取步长时任意性较大且与最严约束值有关, 但可以给出一个规定性方法•　对连续规划采用黄金分割法, 在搜索的初始阶段, 步长可大一些, 随着搜索次数的增加, 最严约束绝对值将逐步减小, 所以步长将逐步减小; 对离散规划, 采用 Fibonacci 数列法, 只能按给定的许用离散集进行搜索, 步长不能随意改变•　限于篇幅, 细节从略•

3　数值算例

例 1[1]

$$\begin{cases}\min f(\boldsymbol{X})=4x_1-x_2^2-12,\\ \text{s.t. } h(\boldsymbol{X})=1-x_1^2/25-x_2^2/25=0,\\ \qquad g(\boldsymbol{X})=1+x_1^2/34-10x_1/34+x_2^2/34-10x_2/34\leqslant 0,\qquad x_1,x_2\geqslant 0\text{•}\end{cases}\tag{7}$$

解　这是文[1]359—362页例题8.4_1•　本法的解题过程如表1•

表1　例1的搜索过程

$K-1$	$\boldsymbol{X}^{K-1}$	h,g,f	最严约束	β_1,β_2	$\mathrm{d}x_1,\mathrm{d}x_2$	注释
0	$(3,\ 4)^{\mathrm{T}}$	$0,-0.325,-16$	$h(\boldsymbol{X})$	$\frac{16.7}{-1},\ \frac{-25}{-1}$	$-3_-,\ 1_-$	使 $h(\boldsymbol{X})=0$, $f(\boldsymbol{X})$ 下降最多
1	$(0_+,\ 5_-)^{\mathrm{T}}$	$0,\ 0.265,-37$	$g(\boldsymbol{X})$	$\frac{13.6}{-1},\ \frac{170}{-1}$	$0.9801,-0.1$	使 $h(\boldsymbol{X})=0$, $g(\boldsymbol{X})$ 下降, $\boldsymbol{X}$ 不可行
2	$(0.9801,\ 4.9)^{\mathrm{T}}$	$0,\ 0.005,-32$	$g(\boldsymbol{X})$	$\frac{16.92}{-1},\ \frac{1666}{-1}$	$0.0629,-0.01$	使 $h(\boldsymbol{X})=0$, $g(\boldsymbol{X})$ 下降, $\boldsymbol{X}$ 不可行
3	$(1.043,\ 4.89)^{\mathrm{T}}$	$0,-0.0097,-31.74$	$h(\boldsymbol{X})$	$\frac{47.93}{-1},\ \frac{-25}{-1}$	$0.0431,\ 0.009$	使 $h(\boldsymbol{X})=0$, $f(\boldsymbol{X})$ 下降最多
4	$(0.9999,\ 4.899)^{\mathrm{T}}$	$4\times10^{-7},\ 3\times10^{-4},$ -32.001	$g(\boldsymbol{X})$	$\frac{17}{-1},\ \frac{1649}{-1}$	$0.0015,-0.0003$	使 $h(\boldsymbol{X})=0$, $g(\boldsymbol{X})$ 下降, $\boldsymbol{X}$ 不可行
5	$(1.0014,\ 4.8987)^{\mathrm{T}}$	$-2.5\times10^{-6},$ $-2.8\times10^{-5},-31.992$				已达全部约束边界

因此 $\boldsymbol{X}^* = (1.001\,4,\ 4.898\,7)^{\mathrm{T}}$, $f(\boldsymbol{X}^*) = -31.992$, $h(\boldsymbol{X}^*) = -2.5\times10^{-6}$, $g(\boldsymbol{X}^*) = -2.8\times10^{-5}$. $\boldsymbol{X}^*$ 为全局最优解. 与之相比, 文[1] 的解为 $\boldsymbol{X}^* = (1.001,\ 4.898)^{\mathrm{T}}$, $h(\boldsymbol{X}^*) = 5.53\times10^{-6}$, $f(\boldsymbol{X}^*) = -31.992$①.

例 2

$$\begin{cases} \min f(\boldsymbol{X}) = x_1^2 + x_2^2 + x_3^2 - 2x_1x_2 - 3x_2x_3 - 4x_3x_4 - 5x_4x_5, \\ \qquad h(\boldsymbol{X}) = (x_1 + x_2 + x_3 + x_4 + x_5)/50 - 1 = 0, \\ \text{s.t.}\, g(\boldsymbol{X}) = (3x_1 + 2x_2 + 5x_3 + 2x_4 + 5x_5)/170 - 1 \leqslant 0, \\ \qquad\qquad x_1, x_2, x_3 \in \{0, 1, 2, 3, \dots\}. \end{cases} \tag{8}$$

这是受等式约束的整数规划, 尚不知有什么有效的解法. 本法的求解过程列于表 2.

表 2　　　　例 2 的搜索过程

$K-1$	$\boldsymbol{X}^{K-1}$	h, g, f	最严约束	$\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$	Δx_i	注释
0	$(10, 10, 10, 10, 10)^{\mathrm{T}}$	0, 0, −1 100	$h(\boldsymbol{X})$	$\frac{50}{1}, \frac{-1\,450}{1}, \frac{-2\,450}{1}, \frac{-4\,500}{1}, \frac{-2\,500}{1}$	$\Delta x_1 = -1$, $\Delta x_4 = 1$	Δx_4 使 $f(\boldsymbol{X})$ 降最多, Δx_1 使 $f(\boldsymbol{X})$ 增加最少, 加大步长, 直至 $x_1 = 0$
1	$(0, 10, 10, 20, 10)^{\mathrm{T}}$	0, −0.058 8, −1 900	$h(\boldsymbol{X})$	—, $\frac{-450}{1}, \frac{-4\,450}{1}, \frac{-4\,500}{1}, \frac{-5\,000}{1}$	$\Delta x_2 = -1$, $\Delta x_5 = 1$	Δx_5 使 $f(\boldsymbol{X})$ 降最多, Δx_2 使 $f(\boldsymbol{X})$ 增加最少, 加大步长, 直至 $x_2 = 7$
2	$(0, 7, 10, 20, 13)^{\mathrm{T}}$	0, −0.005 9, −2 161	$h(\boldsymbol{X})$	—, $\frac{-750}{1}, \frac{-4\,000}{1}, \frac{-5\,250}{1}, \frac{-5\,000}{1}$	$\Delta x_2 = -1$, $\Delta x_4 = 1$	Δx_4 使 $f(\boldsymbol{X})$ 降最多, Δx_2 使 $f(\boldsymbol{X})$ 增加最少, 加大步长, 直至 $x_2 = 0$
3	$(0, 0, 10, 27, 13)^{\mathrm{T}}$	0, −0.005 9, −2 735	$h(\boldsymbol{X})$	—, —, $\frac{4\,350}{1}, \frac{-5\,250}{1}, \frac{-6\,750}{1}$	$\Delta x_3 = -1$, $\Delta x_5 = 1$	Δx_5 使 $f(\boldsymbol{X})$ 降最多, Δx_3 使 $f(\boldsymbol{X})$ 增加最少, 加大步长, 直至 $x_3 = 0$
4	$(0, 0, 0, 27, 23)^{\mathrm{T}}$	0, −0.005 9, −3 105	$h(\boldsymbol{X})$	—, —, —, $\frac{-5\,750}{1}, \frac{-6\,750}{1}$	$\Delta x_4 = -1$, $\Delta x_5 = 1$	$h(\boldsymbol{X}) = 0$, $g(\boldsymbol{X}) = -0.005\,9$
5	$(0, 0, 0, 26, 24)^{\mathrm{T}}$	0, 0.011 8, —				不可行

$\boldsymbol{X}^* = \{0, 0, 0, 27, 23\}^{\mathrm{T}}$, $f(\boldsymbol{X}^*) = -3\,105$. . 这是一个全局最优解. 如果将 $g(\boldsymbol{X}^*)$ 中的 170 改为 180, 则 $g(\boldsymbol{X}^*)$ 成为无效约束, $\boldsymbol{X}^* = \{0, 0, 0, 25, 25\}^{\mathrm{T}}$, $f(\boldsymbol{X}^*) = -3\,125$.

例 3[3]

$$\begin{cases} \min f(\boldsymbol{X}) = (x_1 + 6)^2 + (x_2 - 2)^2 + 2x_1x_2 - 40, \\ \text{s.t.}\, h(\boldsymbol{X}) = x_1^2 - x_2 = 0,\ g_1(\boldsymbol{X}) = 1 - x_1 \leqslant 0,\ g_2(\boldsymbol{X}) = x_2^2/3 - 1 \leqslant 0. \end{cases} \tag{9}$$

解　本题取自文[3] 第 429 页, 原书未给出答案. 一法, 在等式约束 $h(\boldsymbol{X})$ 上搜索, 为使 $h(\boldsymbol{X}) = 0$ 成立, 必须同时优化两个变量. 步骤列于表 3.

---

①　按文[1] 的 $\boldsymbol{X}^*$ 校核, 约束函数值应为 $f(\boldsymbol{X}^*) = -32.886$, $h(\boldsymbol{X}^*) = -0.035\,7$, $g(\boldsymbol{X}^*) = -0.000\,232$.

表 3　例 3 的搜索过程

$K-1$	X^{K-1}	h, g_1, g_2	$f(X^{K-1})$	最严约束	β_1, β_2	dx_1, dx_2	注释
0	$(1.2, 1.44)^T$	0, −0.2, −0.308 8	15.609 6	$h(X)$	$\frac{7.2}{1}, \frac{1.28}{1}$	−0.2, 0.44	使 $h(X)=0, f(X)$ 下降并保证 $g_1(X)\leqslant 0$
1	$(1, 1)^T$	0, 0, −0.666 7	12	$h(X)$	$\frac{8}{1}, \frac{0}{1}$	0, 0	x_1 已达下界, 不能再减少, 为保证 $h(X)=0, dx_2=0$

$X^* = (1, 1)^T, f(X^*) = 12$, 为一全局最优解. 另法, 由 $h(X)=0$ 得 $x_2 = x_1^2$, 原问题化作:

$$\begin{cases} \min f(X) = (x_1+6)^2 + (x_1^2-2)^2 + 2x_1^3 - 40, \\ \text{s.t.}\ g_1(X) = 1 - x_1 \leqslant 0,\quad g_2(X) = x_1^4/3 - 1 \leqslant 0. \end{cases} \tag{10}$$

经过两轮搜索后, 得最优解 $x_1^* = 1, f^* = 12$.

一般地说, 对于非线性规划问题(1), 由于等式约束的存在, 独立的设计变量只有 $n-e$ 个. 如有可能的话, 应当用等式约束消去易于消去的变量, 使变量数目减少, 且规划中只含不等式约束. 当然, 消元后目标函数和约束函数的非线性程度会大大提高, 但是并不影响本法搜索时的有效性. 如前所述, 使用本法时勿需解任何方程, 是一种搜索类算法, 因此函数是否是高度非线性不会改变算法的效率, 这也体现出本法的优越性.

例 4

$$\min f(X) = (x_1-1)^2 + (x_2-1)^2,\quad \text{s.t.}\ g(X) = 1 - x_1 - x_2 \leqslant 0. \tag{11}$$

解　此题中的约束实际上是无效的, 如果用一般的方法很难事先判断出来. 本法则很容易识别. 解题过程如下:

1) $X^0 = (2, 2)^T$, $g(X) = -3$, $\beta_1 = \beta_2 = 2/(-1)$. 令 $dx_1 = dx_2 = -1.5$.

2) $X^1 = (0.5, 0.5)^T, g(X) = 0$, $\beta_1 = \beta_2 = -1/(-1) > 0$. 这意味着增加 x_1 和 x_2 会使 $f(X)$ 和 $g(X)$ 同时减少, 即目标函数下降, 但约束又不违反, 最优点在可行域内. 所以实际上这是个无约束问题. 不论约束是多么的复杂, 只要发现某一 $\beta_i > 0$, 即可断定原不等式约束规划为无约束规划问题. 这是本法的又一优点.

例 5[4]

$$\min f(X) = x_1^4 + x_1x_2 + (1+x_2)^2. \tag{12}$$

这是一个无约束优化问题, 用本法经 15 步迭代后 $x_1^* = 0.695 87$, $x_2^* = -1.347 8$, $f^* = -0.582 4$. 此解的精度可以由解析法验证: 令 $\partial f/\partial x_1 = \partial f/\partial x_2 = 0$, 可得到关于 x_1 的方程 $8x_1^3 - x_1 - 2 = 0$, 解之得唯一的实根 $x_1 = 0.695 88$, $x_2 = -1.347 8$, 与本法求得的解几乎一致.

此题取自文[4] 349 页例 11_7, 原文的解是 $x_1^* = 0, x_2^* = -1, f^* = 0$, 显然不是真解. 这可能是因为忽略了目标函数的交叉项 x_1x_2 造成的, 也说明书中方法没有判别最优解的准则. 另外, 与 $f(X) = 0$ 对应的另一解 $X = (1, -1)^T$ 也被漏掉了.

4　结　　论

实例表明, 本文提出的相对微分及差分法, 既适用于连续规划问题, 也适用于离散规划问题. 对于无约束规划问题, 本法可看成是最速下降法的发展; 对约束规划问题, 它是文[2] 的相对差商法的变形和普遍适用性的推广. 本法的特点是: 1) 既能解不等式约束问题, 也能解等

式约束优化问题; 2) 只要在可行域(集)内存在目标函数和约束函数及其一阶导数(差分)的值, 肯定可搜索到最优解, 没有不收敛和不稳定的问题; 3) 以最严约束为统一的单约束, 节省了计算工作量; 4) 给出了在等式最严约束边界上及不等式最严约束边界近旁搜索的策略; 5) 给出了判断无效约束的方法; 6) 不受问题规模的限制•

作者用本法还计算了一些非线性规划问题, 效果都很理想, 限于篇幅, 不再一一列出•

致谢 本课题承大连理工大学211工程建设项目资助, 在此表示衷心感谢•

[参 考 文 献]

[1] Himmelblau D M. Applied Nonlinear Programming[M]. New York: McGraw_Hill, 1972, 221—391.

[2] 孙焕纯、柴山、王跃方, 等. 离散变量结构优化设计(增订版)[M]. 大连: 大连理工大学出版社, 2002, 95—131.

[3] Bazara M S, Shetty C M. Nonlinear Programming Theory and Algorithms[M]. New York: John Wiley & Sons Inc, 1979, 253—496.

[4] 刘夏石. 工程结构优化设计原理、方法和应用[M]. 北京: 科学出版社, 1984, 419—558.

A Universal Approach for Continuous or Discrete Non_Linear Programmings With Multiple Variables and Constraints

SUN Huan_chun[1], WANG Yue_fang[1], CHAI Shan[2]

(1. Department of Engineering Mechanics, Dalian University of Technology, Dalian 116024, P.R.China;

2. School of Mechanical Engineering, Shandong University of Technology, Zibo, Shandong 255012, P.R.China)

Abstract: A universal numerical approach for nonlinear mathematic programming problems is presented with an application of ratios of first_order differentials/differences of objective functions to constraint functions with respect to design variables. This approach can be efficiently used to solve continuous and, in particular, discrete programmings with arbitrary design variables and constraints. As a search method, this approach requires only computations of the functions and their partial derivatives or differences with respect to design variables, rather than any solution of mathematic equations. The present approach has been applied on many numerical examples as well as on some classical operational problems such as one_dimensional and two_dimensional knap_sack problems, one_dimensional and two_dimensional resource_distribution problems, problems of working reliability of composite systems and loading problems of machine, and more efficient and reliable solutions are obtained than traditional methods. The present approach can be used without limitation of modeling scales of the problem. Optimum solutions can be guaranteed as long as the objective function, constraint functions and their first_order derivatives/differences exist in the feasible domain or feasible set. There are no failures of convergence and instability when this approach is adopted.

Key words: continuous or discrete nonlinear programming; search algorithm; relative differential/difference method