快速收敛的可行方向算法

唐燮黎　　陆晓敏　　赵　引

（河海大学土木工程学院　南京　210098）

摘　要　用算例研究了尝试法计算中初始步长选取和收敛速度的关系，进而提出了一种新的步长确定法——约束近似法，避免了人为给定初始步长的缺陷.对比计算表明，本文方法还可大幅度地提高可行方向法的收敛速度.

关键词　优化；可行方向法；步长计算

中图号　O221.2

考虑一个有 n 个设计变量 $\boldsymbol{x}=[x_1\ x_2\cdots\ x_n]^{\mathrm{T}}$，$p$ 个不等式约束的数学规划问题：

极小化　$F(\boldsymbol{x})$　　(1)

满足　$g_j(\boldsymbol{x})\leqslant 0\qquad j=1,2,\cdots,p$

式中的目标函数 F 和约束 g_j 为 $\boldsymbol{x}$ 的线性或非线性函数，并假设它们对设计变量的偏导数

$$\nabla F=\left[\frac{\partial F}{\partial x_1}\ \frac{\partial F}{\partial x_2}\cdots\frac{\partial F}{\partial x_n}\right]^{\mathrm{T}} \tag{2}$$

和

$$\nabla g_j=\left[\frac{\partial g_j}{\partial x_1}\ \frac{\partial g_j}{\partial x_2}\cdots\frac{\partial g_j}{\partial x_n}\right]^{\mathrm{T}}\qquad j=1,2,\cdots,p$$

可求得.从初始可行设计 $\boldsymbol{x}_0$ 出发，可行方向法的迭代公式如下：

$$\boldsymbol{x}_{i+1}=\boldsymbol{x}_i+\alpha\boldsymbol{S}_i\qquad i=0,1,2,\cdots \tag{3}$$

式中：$\boldsymbol{S}_i$——迭代方向；α——步长.

记

$$J=\{j\mid g_j(\boldsymbol{x}_i)=0,j=1,2,\cdots,p\} \tag{4}$$

为 $\boldsymbol{x}_i$ 点的有效约束指标集，方向矢量 $\boldsymbol{S}_i$ 须满足

$$\boldsymbol{S}_i^{\mathrm{T}}\nabla F<0 \tag{5}$$

$$\boldsymbol{S}_i^{\mathrm{T}}\nabla g_j<0\qquad j\in J，当约束曲面凸向可行域外 \tag{6}$$

$$\boldsymbol{S}_i^{\mathrm{T}}\nabla g_j\leqslant 0\qquad j\in J，当约束曲面凸向可行域内或\ g_j\ 为线性约束 \tag{7}$$

条件式(5)～(7)保证了沿迭代方向 $\boldsymbol{S}_i$ 至少可走一小步得到一个改善的可行解.

依据确定方向 $\boldsymbol{S}_i$ 方法的不同，可行方向法还可进一步分类.其中，Zoutendijk[1] 曾最早提出用如下线性规划来确定 $\boldsymbol{S}_i$：

$$\left.\begin{array}{ll}\text{极小化} & -\beta\\ \text{满足} & \boldsymbol{S}_i^{\mathrm{T}}\nabla g_j+\theta_j\beta\leqslant 0\qquad j\in J\\ & \boldsymbol{S}_i^{\mathrm{T}}\nabla F+\beta\leqslant 0\\ & |S_{ik}|\leqslant 1\qquad k=1,2,\cdots,n\end{array}\right\} \tag{8}$$

解线性规划问题有许多成熟的标准算法，而且指标集 J 所包含的有效约束数目一般不多，因此关于方向矢量 $\boldsymbol{S}_i$ 的子优化问题易于求解，计算工作量不大，该法目前仍广为使用.

迭代方向确定后步长的计算可表达成一个单变量的约束极小化问题：

极小化　$F(\boldsymbol{x}_i+\alpha\boldsymbol{S}_i)$

满足　$g_j(\boldsymbol{x}_i+\alpha\boldsymbol{S}_i)\leqslant 0\qquad j=1,2,\cdots,p$　　(9)

收稿日期：1997-08-26

第一作者简介：唐燮黎，男，教授，工程力学专业，主要从事结构计算方法和优化设计研究.

由于可行方向法在每一迭代步将优化分解为求方向 S_i 的线性规划问题和求步长 α 的单变量约束极小化问题,所以适用于求解有众多设计变量的优化问题.对大规模的优化设计问题采用可行方向法的优点还在于:

a. 每一迭代过程结束时都能获得一个可行设计,因此即使在接近最优解时停止迭代,一般也能得到满足实际需要的优化设计.

b. 迭代过程中目标函数单调下降,即每走一迭代步都可得到一个更好的可行设计.与之相比,一些基于约束和目标函数近似表达式的优化方法在收敛过程中有时会产生目标值突然增大的跳跃现象,甚至得不到收敛的优化解.

然而,常见的可行方向算法多数采用尝试法计算步长,也有其不足之处:

a. 为检验迭代点是否位于可行域内和是否到达约束边界,需反复计算约束函数值.对工程中提出的大量优化问题,主要约束常不能用设计变量显式表达,约束值计算较费机时.因此,须要众多的约束计算次数是影响现有可行方向类算法计算效率的一个主要问题.

b. 用尝试法计算步长时必须先给出一初始迭代步长,但在实际使用中有时无法通过预测选取一个适宜的初始步长,此外,在接近最优解的迭代步中,初始步长也应逐步减小,对其变化方式难以作出普遍适用的规定.

本文以 Zoutendijk 方法决定方向矢量 S_i 为例,研究可行方向类算法中步长的最佳确定法.首先,通过算例对常用尝试法的步长选取和收敛速度作一研究,其后提出一个基于约束函数一次和二次近似式的步长确定法,该法可大幅度提高可行方向类算法的收敛速度,并消除了必须人为给定初始步长的缺陷.

1 尝试法确定步长

实际计算中,为提高可行方向法的效率,在计算步长时一般不要求作精确的一维搜索;在确定有效约束时,采用约束容差法,即预先规定一个小的正数 ε_c,凡满足

$$-\varepsilon_c \leqslant g_j \leqslant \varepsilon_c \tag{10}$$

的约束为有效约束.在算法实现时,计算步长 α 的一维问题(式(9))还可分成两步:先求出 α_b 使迭代沿 S_i 方向前进到可行域边界,再通过一维搜索计算 α_i,即要求

$$F(\boldsymbol{x}_i + \alpha_i \boldsymbol{S}_i) = \min_{0 \leqslant \alpha \leqslant \alpha_b} f(\boldsymbol{x}_i + \alpha \boldsymbol{S}_i) \tag{11}$$

Kirsch[2] 曾提出在求出 α_b 后计算下例导数

$$\left.\frac{\mathrm{d}F}{\mathrm{d}\alpha}\right|_{\alpha=\alpha_b} = \boldsymbol{S}_i^{\mathrm{T}} \nabla F(\boldsymbol{x}_i + \alpha_b \boldsymbol{S}_i)$$

若结果小于零,可直接取 $\alpha_i = \alpha_b$,$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \boldsymbol{S}_i$,转入下一轮迭代求方向 S_{i+1};否则,再由式(11)作一维搜索确定 α_i.

这里取定一个初始步长作为每一次迭代开始的试算步长,用尝试法求出 α_b,使设计点位于可行域边界.通过 $\boldsymbol{x}_i$、$\boldsymbol{x}_i + \alpha_b \boldsymbol{S}_i$ 两点处的目标值和 $\boldsymbol{x}_i$ 点目标函数关于 α 的导数值构成一个二次函数 $H(\alpha) = a\alpha^2 + b\alpha + c$ 代替目标函数作近似一维搜索,求出 α_i.具体计算步骤如下:

a. 输入初始步长 $\alpha^{(1)}$,置 $k = 1$.

b. 计算 $\boldsymbol{x}_i^{(k)} = \boldsymbol{x}_i + \boldsymbol{\alpha}^{(k)} \boldsymbol{S}_i$,$g_j(\boldsymbol{x}_i^{(k)})$,确定指标集

$$J^* = \{j \mid g_j > \varepsilon_c, j = 1,2,\cdots,p\} \qquad J = \{j \mid -\varepsilon_c \leqslant g_j \leqslant \varepsilon_c, j = 1,2,\cdots,p\}$$

若 J^*,J 均是空集,转 c;若 J^* 为空集,J 为非空集,令 $\alpha_b = \alpha^{(k)}$,转 e;若 J^* 为非空集,转 d.

c. 令 $\alpha^{(k+1)} = \alpha^{(k)} + (k+1)\alpha^{(1)}$,置 $k = k+1$,转 b.

d. 用线性插入法(图 1)求出 α_b,置迭代点于可行域边界.

e. 计算沿 S_i 方向目标函数的二次近似式 $H(\alpha) = a\alpha^2 + b\alpha + c$,$\alpha_H = -b/(2a)$,为使 $H(\alpha)$ 取极小值的步长,取 $\alpha_i = \min(\alpha_b, \alpha_H)$.

例 1 极小化 $F = x_1^2 + x_2^2$

满足 $g_1 = x_1^2/20 - x_2 + 1 \leqslant 0$ $\qquad$ $g_2 = x_2^2/20 - x_1 + 1 \leqslant 0$

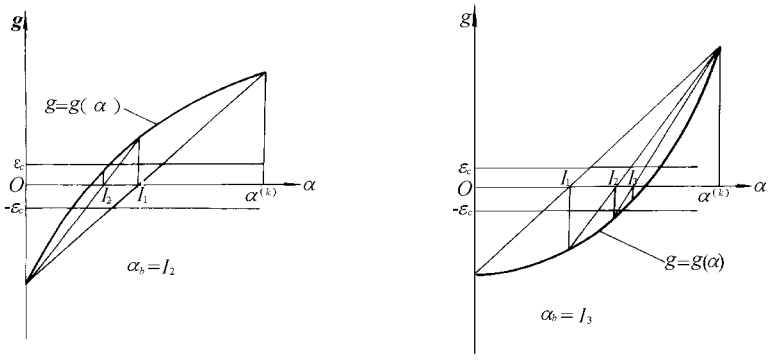

图 1　用线性插入法把迭代点拉回约束边界示意图

Fig.1　Scheme of putting iterative points back to the boundary using linear interpolation

这是 Kirsch[2] 在说明可行方向法的迭代过程时所举的一个例子. 选用同样的设计变量初值 $x_{10}=6$, $x_{20}=3$. 初始目标值 $F_0=45$. 用不同的初始步长进行迭代,收敛情形列于表 1. 由表可见,总迭代次数都为 3 次,但收敛于最优解所需约束计算次数不同. 当 $\alpha^{(1)}=0.7$ 时,所需次数最少,为 18 次.

表 1　例 1 用尝试法计算步长的优化结果

Table 1　Results of optimization by the try-and-error method(example 1)

初始步长	迭代	约束计算	最优解			初始步长	迭代	约束计算	最优解		
$\alpha^{(1)}$	次数	次数	F_{opt}	x_{1opt}	x_{2opt}	$\alpha^{(1)}$	次数	次数	F_{opt}	x_{1opt}	x_{2opt}
0.3	3	23	2.229 45	1.055 81	1.055 81	0.7	3	18	2.229 16	1.055 74	1.055 74
0.4	3	22	2.229 16	1.055 74	1.055 74	0.8	3	20	2.229 22	1.055 75	1.055 75
0.5	3	21	2.229 14	1.055 73	1.055 73	0.9	3	21	2.229 30	1.055 77	1.055 77
0.6	3	21	2.229 13	1.055 73	1.055 73	1.0	3	22	2.229 39	1.055 79	1.055 79

例 2　此例选自文献[3]. 这是一个有吊索加强的悬臂梁受强度约束的优化问题,取结构材料费用为目标函数,形成数学优化问题后为:

极小化　$F=(20/\sqrt{3})x+20y$

满足　$g_1=10T/x\leqslant 80$　　$g_2=750/y-\dfrac{150-\sqrt{3}}{4y}T\leqslant 10$　　$g_3=\dfrac{187.5}{y}-\dfrac{450-3\sqrt{3}}{64y}T\leqslant 0$

$g_4=\dfrac{27T^2}{409.6y}+\dfrac{3\sqrt{3}}{64y}T\leqslant 10$　　$g_5=x\geqslant 0$　　$g_6=y\geqslant 0$

以上式子中　$T=50\Big/\left(\dfrac{0.16y}{\sqrt{3}x}+2.503\right)$

从初始设计 $x_0=3$, $y_0=9$, $F_0=214.641$ 出发,选用不同的初始步长均通过 5 次迭代后收敛于同一最优解 $x_{opt}=3.061\,81$, $y_{opt}=5.769\,23$, $F_{opt}=150.739$. 最优解位于 g_2 和 g_3 约束曲面的交汇处. 表 2 列出不同的初始步长和收敛需要的约束计算次数的关系.

表 2　初始迭代步长与优化过程约束计算次数的关系(例 2)

Table 2　Relationship between tle initial iterative step length and the number of calculation in optimization(example 2)

初始步长 $\alpha^{(1)}$	0.05	0.1	0.2	0.3	0.4	0.8
约束计算次数	33	29	25	25	28	30

例 3　**极小化**　$F=x_1(x_2+2x_3)$

满足　$g_1=3-x_1x_2^2\leqslant 0$　$g_2=2-x_1x_3^2\leqslant 0$

$g_3=1-x_3\leqslant 0$　$g_4=x_3-2x_1\leqslant 0$

迭代初值取为 $x_{10}=x_{20}=x_{30}=2$, $F_0=12$. 最优解位于 g_1, g_2, g_3 约束曲面的交点,其值为 $x_{1opt}=0.793\,70$, $x_{2opt}=1.944\,16$, $x_{3opt}=1.587\,40$, $F_{opt}=4.062\,94$. 初始步长选取和迭代收敛关系列于表 3.

表 3　初始迭代步长与优化收敛速度的关系(例 3)

Table 3　Relationship between the initial iterative step and the convergence rate of optimization(example 3)

初始步长 $\alpha^{(1)}$	0.005	0.01	0.04	0.05	0.06	0.10
迭代次数	18	16	7	17	17	17
约束计算次数	81	66	61	58	61	70

2 约束近似法确定步长

尝试法的算例研究表明,即使选用‘最佳’初始步长,获得优化设计需要的约束计算次数仍然较多.原因来自两个方面:(a)选用较小的初始步长会引起沿迭代方向要走较多的试探步才能跨出可行域,确定有效约束.(b)选用较大的初始步长会引起有效约束确定后需要较多的线性插入才能达到约束边界.本文提出的解决方法是:在每一迭代步,用‘动态’方式确定初始步长,而不采用一个固定的初始步长;在寻求约束边界时,用比较精确的约束近似式而不是用线性近似式.具体来说,在每一迭代步中确定迭代方向 S_i 后,约束作为步长 α 的函数

$$G_j(\alpha) = g_j(\boldsymbol{x}_i + \alpha \boldsymbol{S}_i) \tag{12}$$

将先用线性近似式

$$L_j(\alpha) = G_j(0) + \alpha\left(\frac{\mathrm{d}G_j}{\mathrm{d}\alpha}\right)_0 = g_j(\boldsymbol{x}_i) + \alpha \boldsymbol{S}_i^{\mathrm{T}} \nabla g_j(\boldsymbol{x}_i) \tag{13}$$

代替,求出达到约束面的近似步长(图 2):

$$\alpha_j = -\,G_j(0) \Big/ \left(\frac{\mathrm{d}G_j}{\mathrm{d}\alpha}\right)_0 \tag{14}$$

取

$$\alpha_l = \min_{\substack{1 \leqslant j \leqslant p \\ \alpha_j > 0}} \alpha_j \tag{15}$$

为该迭代步的初始试探步长.

若点 $\boldsymbol{x}_i + \alpha_l \boldsymbol{S}_i$ 不在可行域边界上,我们将用约束的二次近似函数法迭代计算 α_b.设对应于 α_1, α_2(第一次迭代时 $\alpha_1 = 0, \alpha_2 = \alpha_l$)的约束值 $G_j(\alpha_1), G_j(\alpha_2)$ 和 α_1 的约束导数值 $\left(\frac{\mathrm{d}G_j}{\mathrm{d}\alpha}\right)_{\alpha_1}$ 已知,则 $G_j(\alpha)$ 可用 α 的二次函数近似表达(图 3):

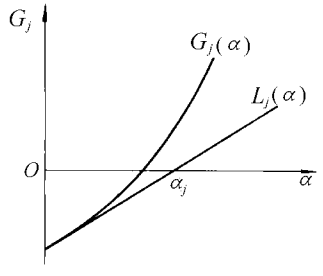

图 2 约束的线性近似

Fig.2 Linear approximation of constraints

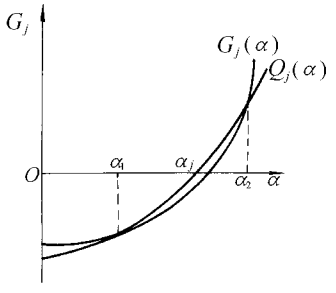

图 3 约束的二次近似

Fig.3 Quadratic approximation of constraints

$$G_j(\alpha) \approx Q(\alpha) = a(\alpha - \alpha_1)^2 + b(\alpha - \alpha_1) + c \tag{16}$$

由条件 $\quad Q(\alpha_1) = G_j(\alpha_1) \qquad Q(\alpha_2) = G_j(\alpha_2) \qquad \left(\frac{\mathrm{d}Q}{\mathrm{d}\alpha}\right)_{\alpha_1} = \left(\frac{\mathrm{d}G_j}{\mathrm{d}\alpha}\right)_{\alpha_1}$

可解出 $\quad C = G_j(\alpha_1) \qquad b = \left(\frac{\mathrm{d}G_j}{\mathrm{d}\alpha}\right)_{\alpha_1} \qquad a = \dfrac{G_j(\alpha_2) - b(\alpha_2 - \alpha_1) - c}{\alpha_2 - \alpha_1}$ (17)

令 $Q(\alpha) = 0$ 给出

$$\alpha_j = \alpha_1 + (-\,b \pm \sqrt{b^2 - 4ac})/(2a) \tag{18}$$

使 $b^2 - 4ac < 0$ 的约束,对步长没有限制,不在考虑之列.式(18)中正、负号选取应满足 $\alpha_j > \alpha_1$,当两个解都大于 α_1 时选取较小者.

使用上述公式,用约束近似法确定步长的计算步骤为:

a. 计算 $G_j(0)$ 和 $\left(\frac{\mathrm{d}G_j}{\mathrm{d}\alpha}\right)_0$,并用式(14)和(15)计算 α_1 及 $G_j(\alpha_1)$,令 $\alpha_1 = 0, \alpha_2 = \alpha_1, k = 1$.

b. 由式(17)和(18)求出 α_j,令 $\alpha^{(k)} = \min\alpha_j$,计算

$$x_i^{(k)} = x_i + \alpha^{(k)} S_i \qquad G_j(x_i^{(k)})$$

确定指标集

$$J^* = \{j \mid g_j > \varepsilon_c, j = 1,2,\cdots,p\} \qquad J = \{j \mid -\varepsilon_c \leqslant g_j \leqslant \varepsilon_c, j = 1,2,\cdots,p\}$$

若 J^* 为空集，J 为非空集，转 d；否则转 c.

c. 若 $\alpha^{(k)} < \alpha_2$，置 $\alpha_1 = \alpha^{(k)}$；若 $\alpha^{(k)} > \alpha_2$，置 $\alpha_1 = \alpha_2, \alpha_2 = \alpha^{(k)}$，并算 $(\frac{dG}{d\alpha})\alpha_1$，置 $k = k + 1$，返回 b.

d. 令 $\alpha_b = \alpha^{(k)}$，计算沿 S_i 方向目标函数二次近似式 $H(\alpha) = a_1\alpha^2 + b_1\alpha + c_1$，使 $H(\alpha)$ 取极小值的步长 $\alpha_H = -b_1/(2a_1)$，取 $\alpha_i = \min(\alpha_b, \alpha_H)$.

用本文提出的约束近似法计算步长，对上述几个算例作对比计算，结果表明，收敛于最优解的迭代次数基本不变，但每次迭代中步长计算的效率大大提高，从而节省了大量机时. 表 4 是尝试法和约束近似法计算结果的比较. 对每个算例，除计算步长的方法不同外，其他方面，如迭代方向计算，初始设计变量，约束容差的取值等均完全相同.

表 4　尝试法和约束近似法计算结果比较

Table 4　Comparison of results calculated by the try-and-error method and that of constraint approximation

方　法	算例 1			算例 2			算例 3		
	迭代次数	约束计算次数	最优目标值	迭代次数	约束计算次数	最优目标值	迭代次数	约束计算次数	最优目标值
尝试法	3	18	2.22916	5	25	150.739	17	58	4.06293
约束近似法	3	8	2.22912	5	9	150.739	18	28	4.06294

注：对表中的 3 个算例，尝试法的初始步长分别取为 0.7，0.2 和 0.05.

3　结语和讨论

a. 通过算例，分析了可行方向类算法用尝试法计算步长的缺陷，提出用约束的线性近似计算每次迭代的初始步长，用约束的二次近似使设计逼近可行域边界的算法. 对比计算结果充分表明这一方法的有效性，即达到优化解需要的约束计算次数不到尝试法的一半.

b. 所建议的方法采用迭代点处的约束及其导数值决定约束的直线近似(图 2)，采用与步长 α_1 对应点的约束及其导数值和步长 α_2 对应点的约束值决定约束的二次近似(图 3)，目的是最大限度地减少约束值计算次数. 在许多工程问题中，约束值的计算需要求解线性代数方程组，计算量较大. 而约束导数的计算可以利用已有的计算分析中间结果，增加的计算量相对较小.

c. 为确定起见，本文采用 Zoutendijk 方法计算可行方向. 实际上，所提出的约束近似法计算步长对可行方向类的其他算法也适用.

参　考　文　献

1　Haftka R T, Gürdal Z, Kamat M P. Elements of structural optimization. Kluwer Academic Publishers, 1990. 155 ~ 158

2　Kirsch U. Optimum Structural Design. McGraw-Hill Inc, 1981. 170 ~ 173

3　唐燮黎. 齐次约束的切平面近似及在优化设计中的应用. 河海大学学报, 1992, 20(6): 80 ~ 86

Feasible Direction Method of Rapid Convergence

Tang Xieli　　Lu Xiaomin　　Zhao Yin

(*College of Civil Engineering, Hohai Univ., Nanjing* 210098)

Abstract　In this paper, the relationship between the initial step length and the convergence rate for the try-and-error method is studied with examples, then a new method for determining the step length, which is called the method of constraint approximation, is presented. The new method avoids the drawback that the initial step length should be set subjectively. Furthermore, the results of comparative calculation show that the new method can considerably improve the rate of convergence for the feasible direction method.

Key words　optimization; feasible direction method; step length calculation