

許驥民 宋成秋 編著

机械结构优化设计

哈尔滨工业大学出版社

OPTIMAL DESIGN OF MECHANICAL STRUCTURES

DESIGN OPTIMAL DE STRUCTURE MECHANIQUE

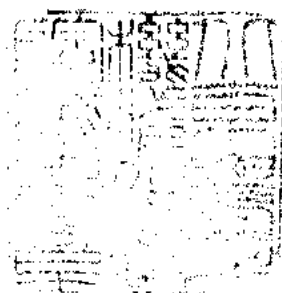
機械構造の最適化設計

OPTIMIERUNGSVERFAHREN ÜBER DER KONSTRUKTION FÜR MASCHINENSTRUKTUR

ОПТИМАЛЬНЫЙ ПРОЕКТ МЕХАНИЧЕСКИХ КОНСТРУКЦИЙ

机械结构优化设计

孙靖民 米成秋 编著



哈尔滨工业大学出版社

内 容 简 介

本书结合机械结构设计问题，阐述优化设计的基本概念和基本理论，内容包括：建立优化设计问题的数学模型，并总结介绍优化设计的概念和方法；必要的数学基础知识和一维搜索方法；无约束优化方法和线性、非线性规划，其中包括近些年发展起来的，使用效果较好的广义简约梯度法、二次规划迭代法和优化准则法；优化方法应用举例。本书并附有数学规划法、优化准则法和有限元—优化设计的 FORTRAN 语言源程序。

本书可供从事系统分析、管理科学，特别是机械结构设计方面的工程技术人员和科研人员阅读，也可作为高等院校有关专业高年级学生和研究生教学参考书。

15341/14

机械结构优化设计

孙培民 米成秋 编著

*

哈尔滨工业大学出版社出版发行

哈尔滨工业大学印刷厂印刷

*

开本 787×1092 1/16 印张 17.75 字数 410,000

1985年5月第1版 1985年5月第1次印刷

印数 1—15,000

书号 15341·14 定价 3.65元

目 录

引 言	(1)
第一章 总论	(3)
§1. 优化设计问题举例	(3)
§2. 人字架的优化设计	(15)
§3. 优化设计的基本概念	(19)
§4. 优化设计的基本方法	(22)
第二章 基础知识	(25)
§1. 梯度的概念	(25)
§2. 多元函数的泰勒 (Taylor) 展开	(29)
§3. 多元函数的极值条件	(31)
§4. 函数的凸性	(33)
§5. 坐标变换及二次型函数的主轴线	(36)
§6. 一维搜索区间的确定和缩短	(39)
§7. 一维搜索的试探方法	(41)
§8. 一维搜索的插值方法	(45)
第三章 无约束优化问题的解法	(49)
§1. 最速下降法	(50)
§2. 牛顿 (Newton) 型方法	(54)
§3. 轭方向及共轭方向法	(57)
§4. 共轭梯度	(61)
§5. 变尺度法	(65)
§6. 鲍威尔 (Powell) 方法	(72)
§7. 单形替换法	(79)
第四章 线性规划和几何规划方法简介	(84)
§1. 线性规划问题的标准形式	(84)
§2. 基本可行解的转换	(86)
§3. 单纯形方法	(90)
§4. 单纯形法应用举例	(94)
§5. 最小二乘法	(99)
§6. 非线性规划问题的线性化解法——线性逼近法	(103)
§7. 对偶问题的概念	(107)
§8. 几何规划简介	(112)

第五章 非线性规划解法之一 —— 间接方法	(117)
§1. 约束变量法和拉格朗日 (Lagrange) 乘子法.....	(117)
§2. 库恩——塔克 (Kuhn—Tucker) 条件.....	(127)
§3. 制约函数法.....	(133)
§4. 乘子法简述.....	(141)
§5. 二次规划迭代法.....	(144)
§6. 简约梯度法和广义简约梯度法.....	(150)
第六章 非线性规划解法之二 —— 直接方法	(163)
§1. 可行方向法.....	(163)
§2. 梯度投影法.....	(165)
§3. 复形法和伸缩保差法简述.....	(176)
§4. 关于多目标函数优化问题的解法.....	(180)
小结 —— 方法的选用.....	(183)
第七章 优化准则法	(184)
§1. 满应力设计和能量准则.....	(184)
§2. 单个位移约束下的优化准则设计.....	(187)
§3. 多位移约束下的优化准则设计.....	(189)
§4. 优化准则法和数学规划法的一致性.....	(196)
第八章 机械结构优化设计举例	(199)
§1. 机床主轴的优化问题.....	(199)
§2. 主轴的优化准则法设计.....	(201)
§3. 主轴的数学规划法优化设计.....	(204)
§4. 主轴的有限元——优化设计.....	(208)
附 录	(217)
一、优化准则法源程序.....	(217)
二、数学规划法源程序.....	(225)
三、有限元——优化设计源程序.....	(243)
参考文献	(279)

引 言

机械产品的设计一般需要经过调查分析、方案拟定、技术设计、零件工作图绘制等环节。在传统设计中，这些环节几乎全都是由设计人员靠手工工具完成的。随着科学技术的发展，市场竞争的需要，要求机械产品更新换代的周期日益缩短，设计质量要求日益提高。因而，这种传统设计方法越来越不能适应发展的需要。

另外，机械产品设计总是希望能获得性能好、使用可靠、成本低等技术经济效益。这就要求设计者能从一系列可能的设计方案中选择一个最好的方案。但在传统设计中，由于分析和计算手段以及时间和费用的限制，可提供选择的方案是有限的，而且也不一定都能从中选出最好的。

近二十年来，随着电子计算机的应用，在机械设计领域内，已经可以用现代化的设计方法和手段进行结构设计，来满足上面对机械产品提出的要求。

现代化的设计工作已不再是过去那种凭借经验或直观判断来确定结构方案，也不是像过去“安全寿命可行设计”方法那样：在满足所提出的要求的前提下，先确定结构方案，再根据安全寿命等准则，对该方案进行强度、刚度等的分析、校核，然后进行修改，以确定结构尺寸。而是借助电子计算机，应用一些精确度较高的力学的数值分析方法（如有限元法等）进行分析计算，并从大量的可行设计方案中寻找出一种最优的设计方案，从而实现用理论设计代替经验设计，用精确计算代替近似计算，用优化设计代替一般的安全寿命可行设计。

机械结构的优化设计方法是六十年代迅速发展起来的一种设计方法，是机械设计现代化的重要发展方向之一。

优化方法在结构设计中的应用，既可以使方案在规定的设计要求下达到某些优化的结果，又不必耗费过多的计算工作量。因此，产品结构、生产工艺等的优化已经成为市场竞争的一种手段。例如，据资料介绍，利用一个化工优化系统（CHEOPS）的计算机手段，对一个化工厂进行设计。根据所给数据，在16小时内，进行一万六千个可行设计的选择，从中选出一个成本最低、产量最大的方案，并给出必需的精确数据。而在这之前，求解这个问题，曾用一组工程师工作了一年，但仅作了三个设计方案，而它们的效率却没有一个可以和上述优化方案相比。又例如，美国贝尔（Bell）飞机公司采用优化方法解决450个设计变量的大型结构优化问题。在对一个机翼进行重设计中，减轻重量达35%。波音（Boeing）公司也有类似的情况，在747机身的设计中，收到了减轻重量、缩短生产周期、降低成本的效果。武汉钢铁公司所引进的1700薄板轧机是西德DMAG公司提供的。该公司在对此产品进行优化修改后，就多盈利几百万马克。

优化方法不仅用于产品结构的设计、工艺方案的选择，也用于运输路线的确定、商品流通量的调配、产品配方的配比等等。目前，优化方法在机械、冶金、石油、化工、

电机、建筑、宇航、造船、轻工等部门都已得到广泛的应用。

在机械结构优化设计方面，国内已经用来解决过起重机塔架、雷达接收天线结构、机械传动方案、机床多轴箱方案、建筑结构等的设计，取得减轻重量达 30~40%，提高承载能力15%以上等等效果。此外，优化方法还是计算机辅助设计的重要组成环节。

机械结构的优化设计有两个主要的环节或内容：一是建立方程，即优化问题的提法或数学模型，它可以是解析式、试验数据或经验公式；另一个是优化问题的求解方法，即优化方法。

第一章 总论

在优化方法中，通常是根据分析对象的设计要求，应用有关专业的基础理论和具体技术知识进行推导来建立相应的方程或方程组。对机械结构类的分析对象来说，主要是根据力学、机械设计基础知识和各专业机械设备的知识来推导方程或方程组（动力学问题中多为偏微分或常微分方程组的形式），这些方程反映结构诸参数之间的内在联系，通过它可以研究各参数对设计对象工作性能的影响。

下面通过几个具体例子，说明机械结构优化设计中建立方程组的方法和步骤（公式的推导尽量简略，以减少篇幅）。

§ 1 优化设计问题举例

例 1. 设计一个如图 (1-1) 所示的钢板弯曲弹簧，用于传递随时间变化的力 P 。为了调节和补偿装配中积累间隙，要求弹簧的变形 Δ 应尽可能的大。因此，对弯曲弹簧优化设计的目标是弹簧变形 Δ 最大。

按材料力学有关弯曲弹簧的应变能公式，得

$$U = \int_0^\pi \frac{M_b^2(Rd\theta)}{2EJ}$$

$$= \int_0^\pi \frac{(PR\sin\theta)^2(Rd\theta)}{2E\left(\frac{bt^3}{12}\right)}$$

或

$$U = \frac{6P^2R^3}{Ebt^3} \int_0^\pi \sin^2\theta d\theta = \frac{3\pi P^2R^3}{Ebt^3}$$

式中的 E 是弹性模量， J 是抗弯惯性矩， M_b 是弯矩，其它符号如图中所示。

变形 Δ 为

$$\Delta = \frac{dU}{dP} = \frac{3\pi R^3(2P)}{Ebt^3} = \frac{6\pi PR^3}{Ebt^3} \quad (1-1)$$

这是一个用设计变量 b , R , t 表示的可计算的函数，称为目标函数。

根据钢板弹簧不同厚度 t 的使用数据，为避免疲劳断裂，要求其弯曲应力 σ_b 应满足下面的关系

$$\sigma_b \leq \frac{47000}{t^{0.25}k}$$

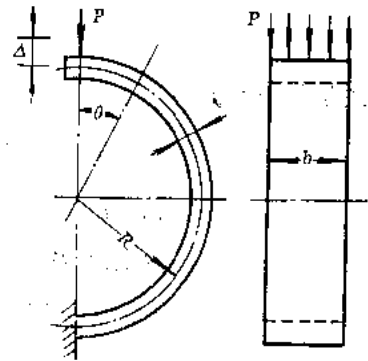


图 1-1

式中 k 是安全系数, 上式称为约束条件。若令 $y = \sigma_b t^{0.28}$, 则上述约束条件可写成下面的形式

$$y \leq y_{max} = \frac{47000}{k}$$

当 $t \ll R$ 时, 可以导出

$$\sigma_b \approx \frac{MC}{J} = \frac{PR \frac{t}{2}}{\frac{bt^3}{12}} = \frac{6PR}{bt^2}$$

这样, 则有

$$y = \sigma_b t^{0.28} = \frac{6PR}{bt^2} t^{0.28} = \frac{6PR}{bt^{1.72}} \quad (1-2)$$

它是一个由载荷和弹簧的几何尺寸组成的因子, 代表载荷环境。

因为制作弹簧时只能在一定范围内选取标准钢板的厚度 t_i , 所以我们对所用钢板厚度 t 应是间断的约束, 即

$$t = t_i \quad (i = 1, 2, \dots, n) \quad (1-3)$$

此外, 还应对尺寸 R 和 b 加上约束

$$R_{min} \leq R \leq R_{max} \quad (1-4)$$

$$b \leq b_{max} \quad (1-5)$$

这些都是设计时应满足的条件, 可概括为约束条件。

这样, 则问题可归结为: 求 b , R 和 $t = t_i$ 的值, 使弹簧的弯曲变形

$$\Delta = \frac{6\pi P}{Et^3} \cdot \frac{R^3}{b} \quad \text{为极大}$$

且满足约束条件

$$y \leq y_{max} \quad \text{或} \quad \frac{6P}{t_i^{1.72}} \cdot \frac{R}{b} \leq \frac{47000}{k}$$

$$b \leq b_{max}$$

$$R_{min} \leq R \leq R_{max}$$

$$t = t_i \quad (i = 1, 2, \dots, n)$$

例 2. 对如图 (1-2) 所示的分度机构进行优化设计。

根据经验和计算, 槽轮机构上的驱动销和槽轮因接触而产生的最大力为 F_{max} , 如图所示。因而可据此确定最大的切应力 τ_{max} , 使机构既在最大工作速度 ω_{max} 下工作, 又能有较长的工作寿命。设计时可以根据经验取

$$\tau_{max} = k_1 \sqrt{\frac{F_{max}}{Td}}$$

式中的系数

$$k_1 = \frac{0.2426}{\sqrt{\frac{(1-\nu_1^2)}{E_1} + \frac{(1-\nu_2^2)}{E_2}}}$$

T 和 d 是槽轮厚度和槽宽，如图 (1—2) 所示； F_{max} 是最大动态接触力，它和 ω 及结构尺寸有关。根据转动刚体动力学和图示尺寸，可以导出

$$F_{max}kD \approx (I_G + I_L)\omega^2(\ddot{\beta})_{max}$$

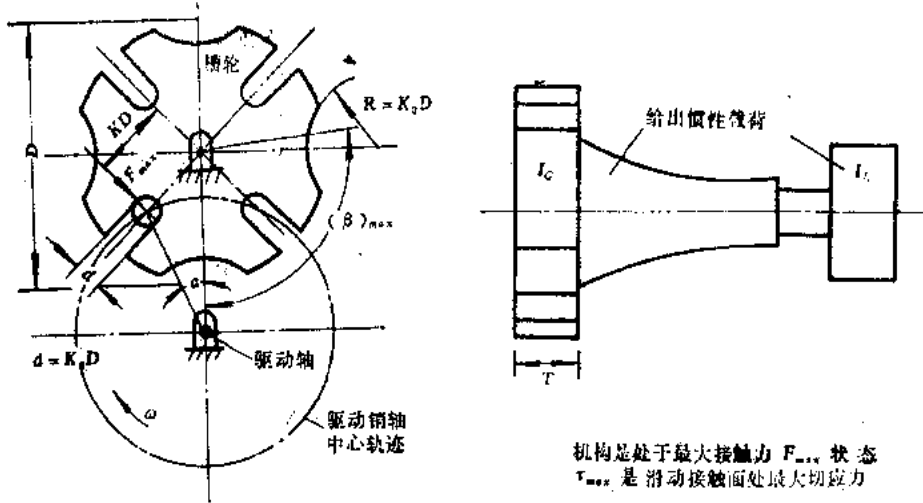


图 1—2

设计时可按槽轮厚度 T 取 $I_G = k_3 TD^4$ ， $I_L = \left(1 \sim \frac{2}{3}\right)I_G$ 。于是得出

$$F_{max} \approx \frac{(I_G + I_L)\omega^2}{kD} \left(\frac{\partial^2 \beta}{\partial \alpha^2}\right)_{max}$$

式中， I_G 和 I_L 是槽轮的质量惯性矩和载荷惯性矩； $\ddot{\beta}_{max} = \left(\frac{\partial^2 \beta}{\partial \alpha^2}\right)_{max}$ 是在 F_{max} 位置时的值； D 是槽轮直径； ω 是驱动轮角速度； k 是常数。

这样，则有

$$\tau_{max} = k_1 \sqrt{\frac{F_{max}}{Td}} \approx k_1 \sqrt{\frac{(I_G + I_L)\omega^2 \ddot{\beta}_{max}}{kDTd}}$$

槽轮机构的设计目标是要求比值

$$\left(\frac{\tau_{max}}{\omega}\right)^2 \approx \frac{k_1^2 (I_G + I_L) \ddot{\beta}_{max}}{kDTd} \quad (1-6)$$

最小。它是设计变量 D 和 T 的可计算函数，亦即设计问题的目标函数。

有两种设计方法，一是给定 τ_{max} ，得 ω 的最大值；另一种是给定 ω ，确定最小允许的 τ_{max} 值。这可使接触磨损引起损坏的机会最小。

上述诸式中的系数 k_1 至 k_3 和 $\ddot{\beta}_{max}$ 等可以从有关设计资料中选取。

这样，则所设计机构的优化问题可以归结为，求 D 和 T 的值，使槽轮机构的切应力

和速度之比 $\left(\frac{\tau_{max}}{\omega}\right)^2 \approx \frac{k_1^2 (I_G + I_L) \ddot{\beta}_{max}}{kDTd}$ 为最小，且满足约束条件

$$d = k_2 D$$

$$I_G = k_3 T D^4$$

$$k_3 \leq \frac{T}{D} \leq k_4$$

$$D \leq D_{max}$$

$$T \leq T_{max}$$

例 3. 在图 (1-3) 所示的接触系统中, θ 是接触块与鼓轮间的接触角, N 是接触

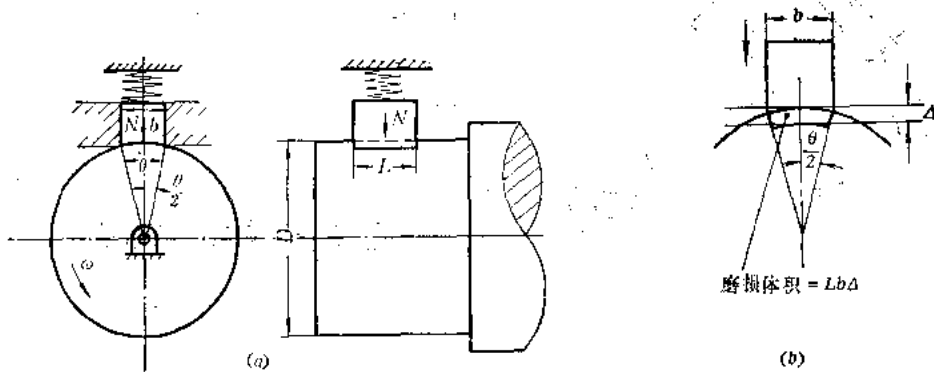


图 1-3

压力, ω 是鼓轮的迴转角速度。它们都是定值。设计时, 要求因磨损而更换接触块的次数最少。

根据接触磨损规律, 接触的运动件间材料的磨损量 (体积) 可写成下式

$$V = \frac{\mu N x}{3p}$$

式中 μ 是磨损系数; N 是作用的正压力; $x = \frac{D}{2} \omega t$ 是接触点间在 t 时间内的相对滑动距离; p 是材料的压强, 可近似地取为其屈服限 σ_s 的 3 倍, 即 $p \approx 3\sigma_s$; D 是鼓轮直径。

上式可改写成

$$V \approx \frac{\mu N \frac{D}{2} \omega t}{3 \cdot 3\sigma_s} = \frac{\mu N D \omega t}{18\sigma_s}$$

从图 (1-3b) 可看出, 磨损体积是

$$V = Lb\Delta$$

其中的 Δ 是磨损深度, b 和 L 是接触块尺寸。

优化设计的目标是使接触件的磨损率 $\frac{\Delta}{t}$ 最小。由上两式得 (以 $b = D \sin \frac{\theta}{2}$ 代入)

$$\frac{\Delta}{t} \approx \frac{\mu N D \omega}{18\sigma_s L b} = \frac{\mu N \omega}{18\sigma_s L \sin \frac{\theta}{2}} \quad (1-7)$$

它是设计变量 D 、 L 、 b 的可计算函数，是目标函数。

由于正压力 N 的作用，鼓轮回转需要克服一定的摩擦力矩 M_f ，其值为

$$M_f = \frac{fND}{2}$$

式中 f 是摩擦系数。

设计时，不允许产生过大的摩擦力矩，即 M_f 应有一个限制值 $(M_f)_{max}$ 。此外，还要考虑对接触块应有空间尺寸的限制。这些都是设计时应满足的约束条件。

这样，就可以把所讨论的问题综述为：求 D 、 b 、 L 之值，使接触件间的磨损率

$$\frac{\Delta}{t} \approx \frac{\mu ND\omega}{18\sigma_s Lb} = \frac{\mu N\omega}{18\sigma_s L \sin \frac{\theta}{2}}$$

为最小，且满足约束条件

$$M_f = \frac{fND}{2} \leq (M_f)_{max}$$

$$b = D \sin \frac{\theta}{2} \leq b_{max}$$

$$L \leq L_{max}$$

$$D_{min} \leq D \leq D_{max}$$

例 4. 轴承和轴承系统的优化设计。

对于动压式滑动轴承，当取无量纲形式的表达式时，通过计算可以得出它的：

$$\text{承载能力系数} = \frac{F\psi^2}{\mu v L}$$

$$\text{润滑油流量系数} = \frac{Q}{\psi v DL}$$

$$\text{轴承的功耗} = \frac{fPv}{10^2}$$

$$\text{轴承的温升} = \frac{fPv}{427c_p \rho Q}$$

$$\text{摩擦阻力系数} = \frac{f}{\psi}$$

$$\text{圆柱轴承的最小油膜厚度} = \frac{D}{2} \psi \left(1 - \frac{e}{c}\right);$$

$$\text{轴颈的失稳转速 (指开始半速涡动时的轴颈转速)} \quad n_{st} = n_{K1} \sqrt{\frac{\bar{m}}{p^2 k_{ev}}} \text{ 等等。}$$

上述各式的： P 是轴承载荷； D 是轴承直径； L 是轴承长度； v 是轴颈圆周速度； μ 是润滑油粘度； c 是半径间隙； e 是轴颈和轴承中心间的偏心距； Q 是润滑油流量； f 是摩擦系数； $\psi = \frac{2c}{D}$ 是间隙比； c_p 是油的比热系数； ρ 是油的比重； $\bar{m} = \frac{\Omega \psi^3}{\mu L} m$ 是转轴分配到轴承上的无量纲质量； m 是转轴分配到轴承上的质量； Ω 是转轴的工作角

速度； k_{ep} 是当量刚度； γ 是刚度和阻尼的比例系数； n_{K1} 是转轴的第一临界转速。

优化设计时，可以取滑动轴承的最大承载能力、最小功耗、最小流量、最小温升或振动过程中的油膜稳定性等等之中的一个或几个的组合做为目标函数。其约束条件可以是最小油膜厚度、轴承温升、轴承功耗、轴承转速、轴承的长径比等等。

对一般的轴承系统，可以从动力学角度考虑它的优化设计。

若把轴承系统看做是由支承和轴承处的轴所组成，则在工作时，由轴和支承的质量、轴承系统刚度和阻尼组成一个振动系统。在外力作用下，它会产生沿垂直和水平两个方向的强迫振动。如果忽略垂直和水平方向上的刚度和阻尼的相互影响，则可以对它的两个方向的振动分别进行研究。若只考虑系统在垂直方向上的振动，则它可以简化成图(1-4)所示的力学模型。

图中 m_1 、 k_1 和 c_1 是轴的当量质量、轴承刚度系数和阻尼系数； m_2 、 k_2 和 c_2 是支承的质量、支承座的刚度系数和阻尼系数。这是一个两自由度的振动系统。

设计时，可以选择、调整 m_1 、 k_1 、 c_1 、 m_2 、 k_2 、 c_2 使系统强迫振动引起的振幅 X_1 和激振力 P 之比 $\frac{X_1}{P}$ ，即动柔度最小（或动刚度最大）。但是必须避免共振，同时 m_1 、 k_1 、 c_1 、 m_2 、 k_2 、 c_2 等应有一个设计对象所能允许的变化范围。

当忽略阻尼影响时，可以通过系统的两个自由度振动的运动方程

$$M\ddot{x} + Kx = p \quad (1-8)$$

解出其动柔度 $\frac{X_1}{P}$ 。式中的 M 和 K 是系统的质量矩阵和刚度矩阵。

避免共振就是要避免激振力频率 ω （例如轴的工作频率 ω ），与系统的固有振动频率 ω_i 重合。工程上按系统固有频率值给出一个频率禁区，使激振力频率不落在频率禁区内。一般要求激振力频率 ω 避开 $(1 \pm 20\%)\omega_i$ 禁区范围。

这样，问题可归结为：确定设计变量 $x = (m_1, m_2, k_1, k_2, c_1, c_2)$ 使目标函数

$$F = \frac{X_1}{P} = f(x) \text{ 最小, 约束条件为:}$$

$$\text{若 } \omega_i > \omega, \text{ 则 } \omega_i > 1.2\omega;$$

$$\text{若 } \omega_i < \omega, \text{ 则 } \omega_i < 0.8\omega;$$

$$x_i^F \leq x_i \leq x_i^E.$$

其中 x_i 分别是 m_1 、 m_2 、 k_1 、 k_2 、 c_1 、 c_2 等设计变量。在实际设计中，轴的当量质量一般是给定的。这时，设计变量中不应再包括 m_1 。

例 5. 平面四连杆机构的优化设计。

平面四连杆机构的设计主要是根据运动学的要求，确定其几何尺寸，以实现给定的

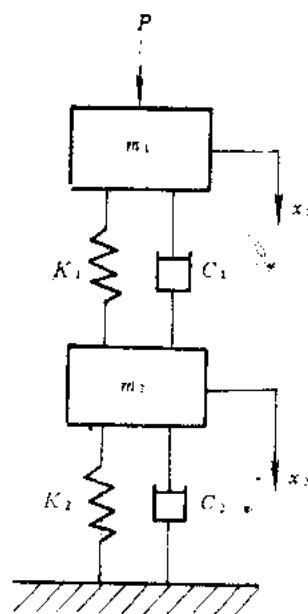


图 1-4

运动规律。

图(1—5)所示是一个曲柄摇杆机构。图中 x_1 、 x_2 、 x_3 、 x_4 分别是曲柄 AB、连杆 BC、摇杆 CD 和机架 AD 的长度。 φ_0 是曲柄输入角， ψ_0 是摇杆输出的起始位置角。这里，规定 φ_0 为摇杆在右极限位置角 ψ_0 时的曲柄起始位置角，它们可以由 x_1 、 x_2 、 x_3 和 x_4 确定。通常规定曲柄长度 $x_1 = 1.0$ ，而在这里 x_4 是给定的，所以只有 x_2 和 x_3 是设计变量。

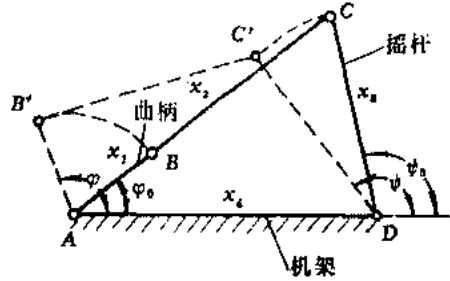


图 1—5

设计时，可在给定最大和最小传动角的前提下，当曲柄从 φ_0 位置转到 $\varphi_0 + 90^\circ$ 时，要求摇杆的输出角最优地实现一个给定的运动规律 $f_0(\varphi)$ 。例如，要求

$$\psi = f_0(\varphi) = \psi_0 + \frac{2}{3\pi} (\varphi - \varphi_0)^2$$

对于这样的设计问题，可以取机构的期望输出角 $\psi = f_0(\varphi)$ 和实际输出角 $\psi_i = f_i(\varphi)$ 的平方误差积分准则做为目标函数，使 $f(x) = \int_{\varphi_0}^{\varphi_0 + 90^\circ} [\psi - \psi_i]^2 d\varphi$ 最小。

当把输入角 φ 取 s 个点进行数值计算时，它可以化约为 $f(x) = f(x_2, x_3) = \sum_{i=1}^s [\psi_i - \psi_{i1}]^2$ 最小。

相应的约束条件有：

1. 曲柄与机架共线位置时的传动角

$$\text{最大传动角 } \gamma_{max} \leq 135^\circ$$

$$\text{最小传动角 } \gamma_{min} \geq 45^\circ$$

对本问题可以计算出

$$\gamma_{max} = \arccos \left[\frac{x_2^2 + x_3^2 - 36}{2x_2x_3} \right]$$

$$\gamma_{min} = \arccos \left[\frac{x_2^2 + x_3^2 - 16}{2x_2x_3} \right]$$

所以有

$$x_2^2 + x_3^2 - 2x_2x_3 \cos 135^\circ - 36 \geq 0$$

和

$$x_2^2 + x_3^2 - 2x_2x_3 \cos 45^\circ + 16 \geq 0$$

2. 曲柄存在条件

$$x_2 \geq x_1$$

$$x_3 \geq x_1$$

$$x_4 \geq x_1$$

$$x_2 + x_3 \geq x_1 + x_4$$

$$x_4 - x_1 \geq x_2 - x_3$$

3. 边界约束

当 $x_1=1.0$ 时, 若给定 x_4 , 则可求出 x_2 和 x_3 的边界值。例如, 当 $x_4=5.0$ 时, 则有曲柄存在条件和边界值限制条件如下:

$$x_2 + x_3 - 6 \geq 0$$

$$4 - x_2 + x_3 \geq 0$$

和

$$1 \leq x_2 \leq 7$$

$$1 \leq x_3 \leq 7$$

例 6. 机床传动系统的优化设计。

这里用一个机床主传动系统为例, 说明在优化设计时建立数学模型的方法。

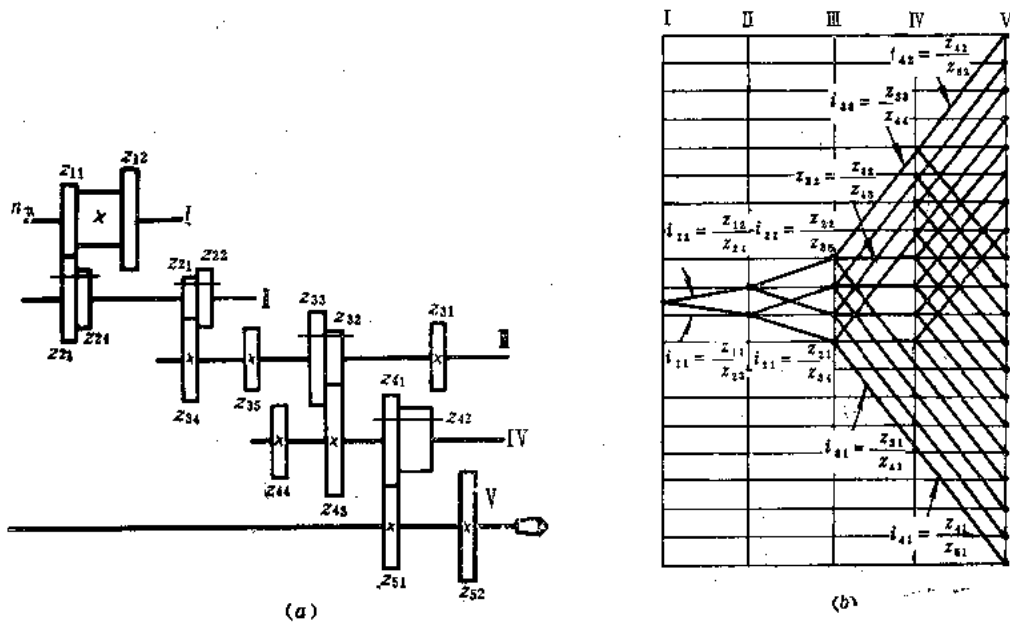


图 1-6

图(1-6a, b)所示分别是某车床主传动的传动系统图和相应的传动结构图。图中没有画出摩擦离合器, 因为它的结构尺寸参数是按现有结构选取的, 不需进行优选。由图可知: 它共有四个传动组, 即在 I—II 轴间的由 i_{11} 和 i_{12} 组成的第一传动组, II—III 轴间由 i_{21} 和 i_{22} 组成的第二传动组, III—IV 轴间由 i_{31} 、 i_{32} 和 i_{33} 组成的第三传动组, IV—V 轴间由 i_{41} 和 i_{42} 组成的第四传动组。IV 轴上的齿轮 z_{41} 是公用齿轮。

各传动组的模数依次为 m_1, m_2, m_3, m_4 。

这个传动系统的设计变量有以下三类, 即:

- 1) 各传动组的最低传动比, 分别是 $i_{11}, i_{21}, i_{31}, i_{41}$;
- 2) 各传动组的最小主动轮齿数, 分别是 $z_{11}, z_{21}, z_{31}, z_{41}$ (由于 z_{41} 是公用齿轮, 所以 z_{42} 就不是独立的变量);
- 3) 各传动组的模数, 分别是 m_1, m_2, m_3, m_4 。

所以共有 12 个设计变量。

说明一点，当采用变位齿轮时，还要考虑变位齿轮传动付和标准齿传轮动付中心距的差值 ΔA_i 。

目标函数取传动路线中各对啮合齿轮中心距之和最小，它可写成

$$f(\mathbf{x}) = \sum_{j=1}^4 \frac{m_j z_{j1}}{2} \left(1 + \frac{1}{i_{j1}} \right) \quad (1-9)$$

约束条件包括：

1) 由于结构尺寸引起的齿轮齿数、传动比值、中心距的限制（可以是上限、下限或上下限）；

2) 齿轮线速度的限制；

3) 齿轮弯曲和接触强度的限制等。

这台车床共有 67 个约束条件，虽然形式各异，但都可统一写成不等式约束的形式

$$g_j(\mathbf{x}) \leq 0 \quad (j = 1, 2, \dots)$$

或

$$x_i^T \leq x_i \leq x_i^U \quad (i = 1, 2, \dots)$$

这样，问题可归结为：求 $\mathbf{x} = (i_{11}, i_{21}, i_{31}, i_{41}, z_{11}, z_{21}, z_{31}, z_{41}, m_1, m_2, m_3, m_4)^T$ 的值，使

$$f(\mathbf{x}) = \sum_{j=1}^4 \frac{m_j z_{j1}}{2} \left(1 + \frac{1}{i_{j1}} \right) \rightarrow \min$$

s. t. (即受约束于)

$$g_j(\mathbf{x}) \leq 0 \quad (j = 1, 2, \dots)$$

$$x_i^T \leq x_i \leq x_i^U \quad (i = 1, 2, \dots)$$

例 7. 直升飞机尾仓部分桁架结构的优化设计。

图 (1-7a) 是某直升飞机尾仓部分的外观图。图 (1-7b) 是它的内部桁架结构的两个视图。这种桁架结构，设计时要求其总重量最小，但各杆在受载时，各杆单元的应力 σ_i 和节点位移 y_i 以及振动的固有频率 ω 都应有所限制。

图示的桁架结构，共有 108 个杆单元，28 个节点。若每个节点考虑 3 个自由度，共有 72 个自由度 (28 个节点中有 4 个节点被固定)。对于这样的结构，目前都是用有限元法来计算它的 y_i ， ω^2 和 σ_i 。当取各杆横截面积 A_i 为设计变量 (可以把 A_i 进行分组处理，以减少设计变量数目) 时，相应的计算公式是

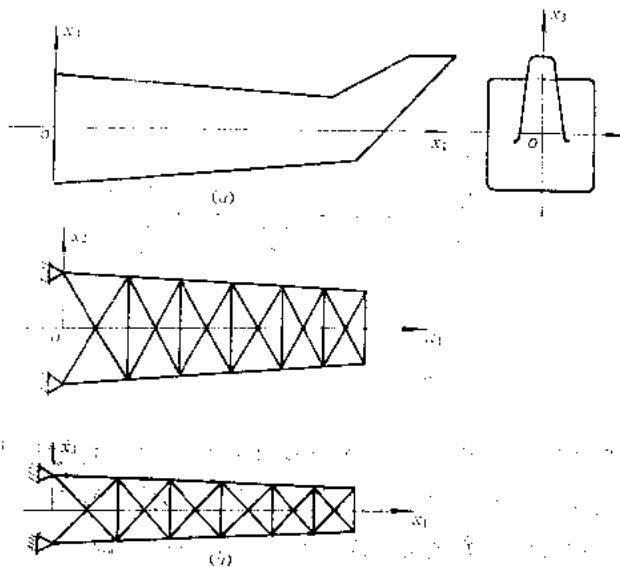


图 1-7

$$K(A)y = F \quad (1-10)$$

$$K(A)\bar{y} = \omega^2 M(A)\bar{y} \quad (1-11)$$

式中， $K(A)$ 和 $M(A)$ 是结构刚阵和质阵； F 是载荷阵； y 是节点位移列阵； ω 和 \bar{y} 是固有频率和对应的振型矩阵。

式 (1-10) 和 (1-11) 是由许多个平衡方程所组成的方程组 (又称状态方程)，通过求解，可以得出与某一设计方案 (某一组设计变量所确定的方案) 相应的 y_j 、 ω^2 和 σ_i 值。

优化设计的目标是结构总重量最轻。若 L_i 是各杆单元的长度，则总重量 (ρ 是材料密度)

$$f(A) = W = \rho g \sum_{i=1}^{108} L_i A_i \quad (1-12)$$

设计时，要求各 σ_i 都不能大于允许的 $\sigma_{i,允}$ ； y_j 必须小于允许的 $y_{j,允}$ ；计算的最低固有频率 ω^2 应大于频率的下限值 ω_0^2 ，各设计变量 A_i 应限制在一定界限之内。

这样，本优化问题归结为：求一组变量 A ，使目标函数

$$f(A) \rightarrow \min$$

s. t.

$$|\sigma_i| \leq \sigma_{i,允} \quad (i = 1, 2, \dots, 108)$$

$$|y_j| \leq y_{j,允} \quad (j = 1, 2, \dots, 72)$$

$$\omega^2 \geq \omega_0^2$$

$$A_i^{\text{下}} \leq A_i \leq A_i^{\text{上}}$$

当然，这些约束条件可统一写成不等式 $g(A) \leq 0$ 的形式。

例8. 机床主轴结构的优化设计。

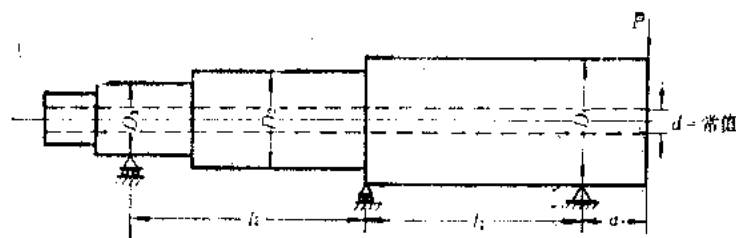


图 1-8

图 (1-8) 所示是一个机床主轴的典型结构原理图。对于这类问题，目前也是采用有限元法，利用式 (1-10) 和 (1-11) 形式的状态方程来计算轴端变形 y 和固有频率 ω 。

优化设计的任务是确定 D_1 、 l_1 和 a ，保证 y 和 ω 在允许限内，使结构的重量最轻。

这时，问题归结为：求 D_1 、 l_1 、 a 的值，使重量 $f(D_1, l_1) = \gamma \pi [\Sigma (D_i^2 - d^2) l_i + (D_n^2 - d^2) a]$ 为最小，并满足条件

$$\begin{aligned}
 \dot{y} &\leq y_{\text{允}} \\
 \omega^2 &\geq \omega_0^2 \\
 D_i^{\text{下}} &\leq D_i \leq D_i^{\text{上}} \\
 l_i^{\text{下}} &\leq l_i \leq l_i^{\text{上}} \\
 a^{\text{下}} &\leq a \leq a^{\text{上}} \\
 N^{\text{下}} &\leq \frac{l_1}{a} \leq N^{\text{上}}
 \end{aligned}
 \quad (i=1, 2, \dots, n)$$

式中： γ 是材料的比重； D_i 和 l_i 是阶梯形主轴的外径和对应的长度； D_a 是和 a 对应的外径。

在主轴结构动力优化设计时，也可取由振型和质量确定的能耗为目标函数。约束条件可以取激振力频率避开 $(1 \pm 20\%) \omega$ 的禁区范围。

例 9. 汽车悬挂系统的优化设计。

在汽车结构系统设计中，希望汽车能在不同速度和道路条件下，驾驶员座位的最大加速度为最小，同时还须满足一系列的动态响应和设计参数的约束。设计参数是选择系统的弹簧常数 k_i 和阻尼系数 c_i 。

图 (1—9) 是 5 个自由度的汽车悬挂系统。图中的 m_1 是驾驶员及其座位的质量，它由弹簧 k_1 和阻尼器 c_1 支持。其它部分，如车体、车轮、车轴等的质量、弹簧和阻尼分别用 m_2 、 m_3 、 m_4 、 m_5 及 k_2 、 k_3 、 c_2 、 c_3 表示，如图 (1—9) 所示。 k_4 、 k_5 和 c_4 、 c_5 表示轮胎的刚度和阻尼系数。车体对其质量中心的惯性矩用 I 表示； L 表示轴距长度。 $f_1(t)$ 和 $f_2(t)$ 表示由于道路表面起伏不平引起的前后轮的位移函数。 z_i 是坐标。

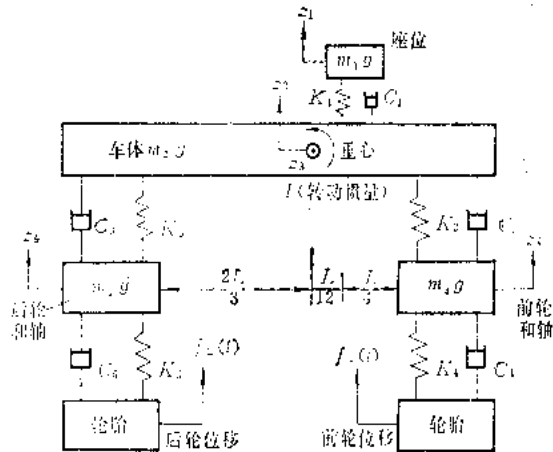


图 1—9

根据拉格朗日运动方程

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{z}_i} \right) - \frac{\partial T}{\partial z_i} + \frac{\partial U}{\partial z_i} - Q_i = 0 \quad (i=1, 2, \dots, 5)$$

可以导出汽车的运动方程。式中的 T 是系统的动能， U 是与弹性力对应的系统位能， Q_i 是非保守广义力。

设 d 是最大加速度的上限。

设计变量取 k_1 、 k_2 、 k_3 、 c_1 、 c_2 、 c_3 和 d ，即 $x = (k_1, k_2, k_3, c_1, c_2, c_3, d)^T$ 。

根据运动方程可以写出状态方程

$$\dot{z}(t) = M(x)z(t) + F(t) \quad (1-13)$$

其中的 $z(t) = [z_1, z_2, z_3, z_4, z_5, z_1, z_2, z_3, z_4, z_5]^T$, $M(x)$ 是由各质量、刚度系数、阻尼系数及 L 和 I 组成的矩阵; $F(t)$ 是由 $m_4, m_5, k_4, k_5, f_1(t), f_2(t)$ 组成的矩阵。

前后轮垂直位移和路面有关, 设分别为按正弦规律变化的位移函数 $f_1(t)$ 和 $f_2(t)$, 其值可表述成

$$f_1(t) = \begin{cases} v(t) & 0 \leq t \leq T_1 \\ 0 & \text{非上述情况} \end{cases} \quad (1-14)$$

$$f_2(t) = f_1(t - t_0) \quad (\text{即比前轮滞后 } t_0)$$

式中 T_1 是路面不平的停止时间。

根据运动方程和位移函数可建立优化问题的数学模型。设计的目标是驾驶员座位的最大绝对加速度最小, 即 $f = \max |\ddot{z}_1^i(t)| \rightarrow \min (i = 1, 2, \dots, p)$ 。其中的 $\ddot{z}_1^i(t)$ 是对第 i 种道路条件 $f_1^i(t)$ 和 $f_2^i(t)$ 下的座位加速度。当规定最大加速度的上限值为 d 时, 则 $|\ddot{z}_1^i(t)| \leq d$ 。

各组成部件之间的允许相对位移约束可写成

$$\psi_s = \int_0^r L_s[t, z(t), x] dt = 0 \quad (s = 1, 2, \dots, 6p)$$

$$x_j^F \leq x_j \leq x_j^U$$

设计变量的变化范围也可写成不等式约束的形式, $g(x) \leq 0$ 。

这样, 优化问题变成:

$$\text{目标函数 } f = \max |\ddot{z}_1^i(t)| \rightarrow \min$$

或写成 $\bar{f} = d$ (d 是 $|\ddot{z}_1^i(t)| \leq d$ 中的最小者)

约束条件是

$$\dot{z}(t) = M(x)z(t) + F(t)$$

$$\psi_s = \int_0^r L_s[t, z(t), x] dt = 0$$

$$g_s(x) \leq 0$$

例 10. 某建筑单位拟盖一批 2 人、3 人和 4 人的宿舍单元, 要确定每一种宿舍单元的数目, 以获得最大利润。其限制条件如下:

- 1) 预算不能超过 9000 (千元);
- 2) 宿舍单元总数不得少于 350 套;
- 3) 每类宿舍单元的百分比为: 2 人的是总数的 20%, 3 人的是总数的 60%, 4 人的是总数的 40% (百分比总和超过 100, 这是上限);
- 4) 建造价格为: 2 人的宿舍单元是 20 (千元), 3 人的宿舍单元是 25 (千元), 4 人的宿舍单元是 30 (千元);
- 5) 净利润为: 2 人的宿舍单元是 2 (千元) 3 人的宿舍单元 3 (千元), 4 人的宿舍单元是 4 (千元)。

根据上述条件, 利润总数就是目标函数。若令 x_1, x_2, x_3 分别是 2 人、3 人和 4

人的宿舍单元数日, 则利润总数为

$$f(x) = 2x_1 + 3x_2 + 4x_3 \quad (1-15)$$

其值应是最大值。

约束条件有以下几个:

1. 预算不超过9000 (千元), 即有

$$20x_1 + 25x_2 + 30x_3 \leq 9000$$

2. 宿舍单元总数最少是350套, 即有

$$x_1 + x_2 + x_3 \geq 350$$

3. 每类宿舍单元数的约束不等式是 (设宿舍总套数为 $350 + x_5$)

$$x_1 \leq 0.2(350 + x_5)$$

$$x_2 \leq 0.6(350 + x_5)$$

$$x_3 \leq 0.4(350 + x_5)$$

因此, 问题可归结为: 求 x_1, x_2, x_3 等值, 使目标函数 $f(x) = 2x_1 + 2x_2 + 4x_3$ 为极大, 且满足约束条件

$$20x_1 + 25x_2 + 30x_3 \leq 9000$$

$$x_1 + x_2 + x_3 \geq 350$$

$$x_1 - 0.2x_5 \leq 70$$

$$x_2 - 0.6x_5 \leq 210$$

$$x_3 - 0.4x_5 \leq 140$$

当然还可以举出一些其它行业的例子。但不管是哪个专业范围内的问题, 都可以按照如下的方法和步骤来建立相应的优化设计问题的数学模型。

1. 根据设计要求, 应用专业范围内的现行理论和经验等, 对优化对象进行分析。必要时, 需要对传统设计中的公式进行改进, 并尽可能反映该专业范围内的现代技术进步的成果。

2. 就结构诸参数进行分析, 以确定设计的原始参数、设计常数和设计变量 (说明见 §3)。

3. 根据设计要求, 确定并构造目标函数和相应的约束条件, 有时要构造多目标函数。

4. 必要时对数学模型进行规范化, 以消除诸组成项间由于量纲不同等原因导致的数量悬殊的影响。

有时不了解结构 (或系统) 的内部特性, 则可建立黑箱 (Black box) 模型。

§ 2 人字架的优化设计

为了对结构的优化设计有一具体的认识, 现以人字架的优化设计为例子以说明。虽然此设计采用简单的解析法和作图法, 但从中可以了解优化的具体过程, 以及优化问题的一些基本概念。

一、问题

图(1-10)所示的人字架由两个钢管构成,其顶点受 $2P = 30000\text{kg}$ 重力。已知人字架跨度 $2B = 152\text{cm}$, 钢管壁厚 $T = 0.25\text{cm}$, 钢管材料的弹性模量 $E = 2.1 \times 10^8 \text{kg/cm}^2$, 材料比重 $\gamma = 7.8 \times 10^{-5} \text{kg/cm}^3$, 许用压应力 $\sigma_y = 4200 \text{kg/cm}^2$ 。求在钢管压应力 σ 不超过许用压应力 σ_y 和失稳临界应力 σ_c 的条件下, 人字架的高 H 和钢管平均直径 D , 使钢管总重量 W 为最小。

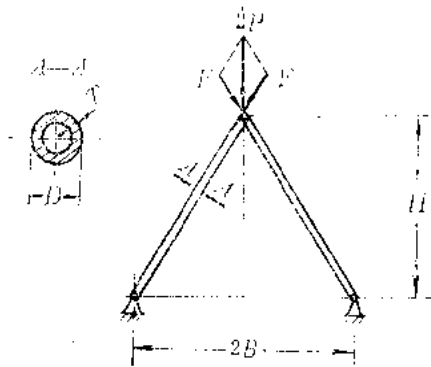


图 1-10

根据以上所述, 可以把人字架的优化设计问题归结为:

求 $x = (D, H)^T$, 使结构重量

$$W(x) \rightarrow \min$$

但应满足强度约束条件

$$\sigma(x) \leq \sigma_y$$

和稳定约束条件

$$\sigma(x) \leq \sigma_c$$

二、强度、稳定条件

钢管所受的压力

$$F = \frac{PL}{H} = \frac{P(B^2 + H^2)^{\frac{1}{2}}}{H}$$

压杆失稳的临界力, 见图(1-11)

$$F_c = \frac{\pi^2 EI}{L^2}$$

式中 I 是钢管截面惯性矩

$$I = \frac{\pi}{4} (R^4 - r^4) = \frac{A}{8} (T^2 + D^2)$$

A 是钢管截面面积 (r, R 为截面内、外半径, $D = R + r$)

$$A = \pi (R^2 - r^2) = \pi TD$$

钢管所受的压应力

$$\sigma = \frac{F}{A} = \frac{P(B^2 + H^2)^{\frac{1}{2}}}{\pi TDH}$$

钢管的临界应力

$$\sigma_c = \frac{F_c}{A} = \frac{\pi^2 E (T^2 + D^2)}{8(B^2 + H^2)}$$

因此, 强度约束条件 $\sigma \leq \sigma_y$ 可以写成



图 1-11

$$\frac{P(B^2 + H^2)^{\frac{1}{2}}}{\pi TDH} \leq \sigma_y$$

稳定约束条件 $\sigma \leq \sigma_c$ 可以写成

$$\frac{P(B^2 + H^2)^{\frac{1}{2}}}{\pi TDH} \leq \frac{\pi^2 E(T^2 + D^2)}{8(B^2 + H^2)}$$

三、解析法

上述优化问题是以 D 和 H 为设计变量的二维问题，而且只有两个约束条件，可以用解析法进行求解。

假定使人字架总重量

$$W(D, H) = 2\gamma AL = 2\pi\gamma TD(B^2 + H^2)^{\frac{1}{2}}$$

为最小的最优解，刚好满足强度条件，即有

$$\sigma(D, H) = \sigma_y$$

从而可以将设计变量 D 用设计变量 H 表示

$$D = \frac{P(B^2 + H^2)^{\frac{1}{2}}}{\pi T\sigma_y H}$$

将 D 代入目标函数 $W(D, H)$ 中，得

$$W(H) = \frac{2\gamma P}{\sigma_y} \cdot \frac{B^2 + H^2}{H}$$

根据极值必要条件

$$\frac{dW}{dH} = \frac{2\gamma P}{\sigma_y} \cdot \frac{d}{dH} \left(\frac{B^2 + H^2}{H} \right) = \frac{2\gamma P}{\sigma_y} \left(1 - \frac{B^2}{H^2} \right) = 0$$

得

$$H^* = B = \frac{152}{2} = 76\text{cm}$$

$$D^* = \frac{\sqrt{2}P}{\pi T\sigma_y} = 6.43\text{cm}$$

$$W^* = \frac{4\gamma PB}{\sigma_y} = 8.47\text{kg}$$

把所得参数代入稳定条件，可以证明

$$\sigma(D^*, H^*) \leq \sigma_c(D^*, H^*)$$

即稳定约束条件得到满足。所以， H^* 、 D^* 这两个参数是满足强度约束和稳定约束，且使结构最轻的最佳参数。

四、作图法

在设计平面 $D-H$ 上画出代表

$$\sigma(D, H) = \sigma_y$$

和

$$\sigma(D, H) = \sigma_c(D, H)$$

两条曲线，如图（1—12）所示。两曲线将设计平面分成两部分，其中不带阴影线的区域是同时满足

$$\sigma \leq \sigma_y$$

和

$$\sigma \leq \sigma_e$$

两个约束条件的区域，称为可行域。然后再画出一族重量等值线

$$W(D, H) = C$$

（ C 为一系列常数）。从图中可以看出，等值线在可行域内部无中心，故此约束优化问题的极值点处于可行域边界与等值线的切点处，从而找到极值点 X^* 的坐标

$$D^* = 6.43\text{cm}$$

$$H^* = 76\text{cm}$$

通过 X^* 的等值线就是最轻结构重量，其值为

$$W^* = 8.47\text{kg}$$

最优点 x^* 处于强度曲线上，说明此时强度条件刚好满足，而稳定条件不但满足且有一定裕量。这表明强度约束条件为起作用约束，它影响极值点的位置，稳定约束条件为不起作用约束，它不影响极值点的位置。

五、讨论

若将许用压应力 σ_y 由 4200kg/cm^2 提高到 7030kg/cm^2 ，这时强度约束条件发生变化，因而可行域也发生变化，如（图 1—12）所示。若仍按上述解析法进行求解，还假定最优点刚好满足强度条件，得

$$H = B = 76\text{cm}$$

$$D = \frac{\sqrt{2}P}{\pi T\sigma_y} = 3.84\text{cm}$$

$$W = \frac{4\gamma PH}{\sigma_y} = 5.06\text{kg}$$

当在 $D-H$ 设计平面上标出此点时，可以看出它位于等值线

$$W(D, H) = 5.06$$

与强度曲线

$$\sigma(D, H) = 7030$$

的切点 x_1 处。但是， x_1 点位于可行域之外，它不满足稳定条件。这也可以通过将 x_1 点处的 D 和 H 的上述数值代入稳定条件而得到证实。因此，这表明 x_1 不是最优点。

用作图法可找出最优点位于强度曲线和稳定曲线的交点 x_1^* 处。它的坐标值就是最

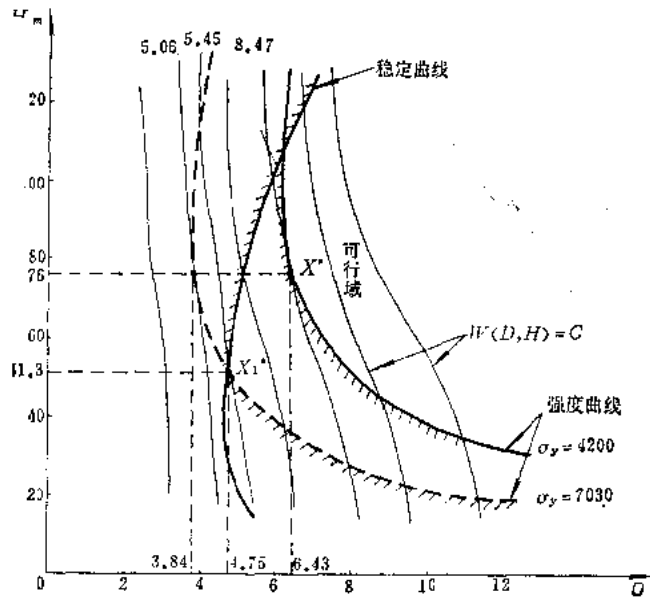


图 1—12

优参数，其值为

$$H_1^* = 51.3\text{cm}$$

$$D_1^* = 4.75\text{cm}$$

通过 x_1^* 的等值线值即为最轻结构重量，其值为

$$W_1^* = 5.45\text{kg}$$

因为 x_1^* 点的位置是由强度曲线和稳定曲线的交点所决定的，所以强度约束条件和稳定约束条件都得到满足，且二者都是起作用的约束条件。最优点仍处于可行域边界与等值线的切点位置。

从上面的讨论可知，对于具有不等式约束条件的优化问题，判断哪些约束是起作用的，哪些约束是不起作用的，这对求解优化问题是很关键的。

§ 3 优化设计的基本概念

在人字架优化设计的基础上，本节对一般优化问题的基本概念作概括的说明，以便突出其数学的实质，为后继各章中优化方法的讨论作必要的准备。

一、设计变量

一个设计方案可以用一组基本参数的数值来表示。这些基本参数可以是构件长度、截面尺寸、某些点的坐标值等几何量，也可以是重量、惯性矩、力或力矩等物理量，还可以是应力、变形、自振频率、效率等代表工作性能的导出量。但是，对某个具体的优化设计问题，并不是要求对所有的的基本参数都用优化方法进行修改调整。例如，对某个机械结构进行优化设计，一些工艺、结构布置等方面的参数，或者某些工作性能的参数，可以根据已有的经验预先取为定值。这样，对这个设计方案来说，它们就成为**设计常数**。而除此之外的基本参数，则需要在优化设计过程中不断进行修改、调整，一直处于变化的状态，这些基本参数称作**设计变量**，又叫做优化参数。

设计变量的全体实际上是一组变量，可用一个列向量表示：

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^T$$

称作**设计变量向量**。向量中分量的次序完全是任意的，可以根据使用的方便任意选取。例如，在例1中的 b, R, t 相当于 x_1, x_2, x_3 三个变量；例3中的 D, b, L 相当于 x_1, x_2, x_3 三个变量；例4中的 m_1, m_2, \dots, c_2 相当于 x_1, x_2, \dots, x_6 六个变量等等。这些设计变量可以是一些结构尺寸参数，也可以是一些化学成分的含量或电路参数等。一旦规定了这样一种向量的组成，则其中任意一个特定的向量都可以说是一个“设计”。由 n 个设计变量为坐标所组成的实空间称作**设计空间**。一个“设计”，可用设计空间中的一点表示，此点可看成是设计变量向量的端点（始点取在坐标原点），称作**设计点**。

二、约束条件

设计空间是所有设计方案的集合，但这些设计方案有些是工程上所不能接受的（例如面积取负值等）。如果一个设计满足所有对它提出的要求，就称为可行（或可接受）设计，反之则称为不可行（或不可接受）设计。

一个可行设计必须满足某些设计限制条件，这些限制条件称作**约束条件**，简称**约束**。在工程问题中，根据约束的性质可以把约束区分成**性能约束**和**侧面约束**两大类。针对性能要求而提出的限制条件称作性能约束。例如，选择某些结构必须满足受力的强度、刚度或稳定性等要求，比方要求桁架某点变形不超过给定值。不是针对性能要求，只是对设计变量的取值范围加以限制的约束称作侧面约束，例如允许选择的尺寸范围等，比方要求桁架的高在其上下限范围之间。

约束又可按其数学表达形式分成等式约束和不等式约束两种类型。等式约束

$$h(x) = 0$$

要求设计点在 n 维设计空间的约束曲面上。不等式约束

$$g(x) \leq 0$$

要求设计点在设计空间中约束曲面 $g(x) = 0$ 的一侧（包括曲面本身）。所以，约束对设计点在设计空间中的活动范围有所限制。凡满足所有约束条件的设计点，它在设计空间中的活动范围称作**可行域**。如满足不等式约束

$$g_j(x) \leq 0 \quad (j = 1, 2, \dots, m)$$

的设计点活动范围，它是由 m 个约束曲面

$$g_j(x) = 0 \quad (j = 1, 2, \dots, m)$$

所形成的 n 维子空间（包括边界）。若满足两个或更多个 $g_j(x) = 0$ 点的集合称作交集。在三维空间中两个约束的交集是一条空间曲线，三个约束的交集是一个点。在 n 维空间中 r 个不同约束的交集的维数是 $n - r$ 的子空间。等式约束 $h(x) = 0$ 可看成是同时满足 $h(x) \leq 0$ 和 $h(x) \geq 0$ 两个不等式约束，代表 $h(x) = 0$ 曲面。

约束函数有的可以表示成显式形式，即反映设计变量之间明显的函数关系，如例 1 和例 3 中的约束条件，这类约束称作显式约束。有的只能表示成隐式形式，如例 8 和例 9 中的复杂结构的性能约束函数（变形、应力、频率等），需要通过有限元法计算求得，机构的运动误差要用数值积分来计算，这类约束称作隐式约束。

三、目标函数

在所有的可行设计中，有些设计比另一些要“好些”，如果确实是这样，则“较好”的设计比“较差”的设计必定具备某些更好的性质。倘若这种性质可以表示成设计变量的一个可计算函数，则我们就可以考虑优化这个函数，以得到“更好”的设计。这个用来使设计得以优化的函数称作**目标函数**，用它可以评价设计方案的好坏，所以它又被称作**评价函数**，记作 $f(x)$ 或 $F(x)$ ，用以强调它对设计变量的依赖性。目标函数可以是结构重量、体积、功耗、产量、成本或其它性能指标（如变形，应力等）和经济指标等。

建立目标函数是整个优化设计过程中比较重要的问题。当对某一设计性能有特定的要求，而这个要求又很难满足时，则若针对这一性能进行优化将会取得满意的效果。但在某些设计问题中，可能存在两个或两个以上需要优化的指标，即多目标函数的问题。例如，设计一台机器，期望得到最低的造价和最少的维修费用。

目标函数是 n 维变量的函数，它的函数图象只能在 $n + 1$ 维空间中描述出来。为了在 n 维设计空间中反映目标函数的变化情况，常采用目标函数等值面的方法。目标函数的等

值面，其数学表达式为

$$F(x) = c \quad (1-16)$$

(c 为一系列常数)，代表一族 n 维超曲面。如在二维设计空间中 $F(x_1, x_2) = c$ ，代表 x_1-x_2 设计平面上的一族曲线。

四、优化问题的数学模型

优化问题的数学模型是实际优化设计问题的数学抽象。在明确设计变量、约束条件、目标函数之后，优化设计问题就可以表示成一般数学形式：

求设计变量向量 $x = (x_1, x_2, \dots, x_n)^T$ 使

$$F(x) \rightarrow \min$$

且满足约束条件

$$h_k(x) = 0 \quad (k=1, 2, \dots, l) \quad (1-17)$$

$$g_j(x) \leq 0 \quad (j=1, 2, \dots, m)$$

利用可行域概念，可将数学模型的表达进一步简练。设同时满足 $g_j(x) \leq 0$ ($j=1 \sim m$) 和 $h_k(x) = 0$ ($k=1 \sim l$) 的设计点集合为 R ，即 R 为优化问题的可行域，则优化问题的数学模型可简练地写成：

求 x 使

$$\min_{x \in R} F(x) \quad (1-18)$$

符号“ \in ”表示“从属于”。

在实际优化问题中，对目标函数一般有两种要求形式：目标函数极小化 $F(x) \rightarrow \min$ 或目标函数极大化 $F(x) \rightarrow \max$ 。由于求 $F(x)$ 的极大化与求 $-F(x)$ 的极小化等价，所以今后优化问题的数学表达一律采用目标函数极小化形式。

优化问题可以从不同的角度进行分类。例如，按其有无约束条件分成无约束优化问题和约束优化问题。也可以按约束函数和目标函数是否同时为线性函数，分成线性规划问题和非线性规划问题。如例10的目标函数和约束条件都是线性的，属于线性规划问题。例1和例2的目标函数和约束条件都是非线性的，而例3的目标函数是非线性的，约束条件则是线性的，这属于非线性规划问题。还可以按问题规模的大小进行分类，例如设计变量和约束条件的个数都在50以上属大型，10个以下属小型，10~50属中型。随着电子计算机容量的增大和运算速度的提高，划分界限将会有所变动。

五、优化问题的几何解释

无约束优化问题就是在没有限制的条件下，对设计变量求目标函数的极小点。在设计空间内，目标函数是以等值面的形式反映出来，则无约束优化问题的极小点即为等值面的中心。

约束优化问题是在可行域内对设计变量求目标函数的极小点，此极小点在可行域内或在可行域边界上。用(图 1-13)可以说明有约束的二维优化问题，极值点所处位置的不同情况。图中的(a)是约束函数和目标函数均为线性函数的情况，等值线为直线，可行域为 n 条直线围成的多角形，则极值点处于多角形的某一顶点上。图中的(b)是约束函数和目标函数均为非线性函数的情况，极值点位于可行域内等值线的中心处，约束对极值点的选取无影响，这时的约束为不起作用约束，约束极值点和无约束极值点相同。

图中的(c)和(d)均为约束优化问题极值点处于可行域边界的情况，约束对极值点的位置影响很大。图(c)中的约束 $g_1(x) = 0$ 在极值点处是起作用约束，图(d)中的约束 $g_2(x) = 0$ 在极值点处是起作用约束，而图(e)中的约束 $g_1(x) = 0$ 和 $g_2(x) = 0$ 同时在极值点处为起作用约束。多维问题最优解的几何解释可借助于二维问题进行想像。

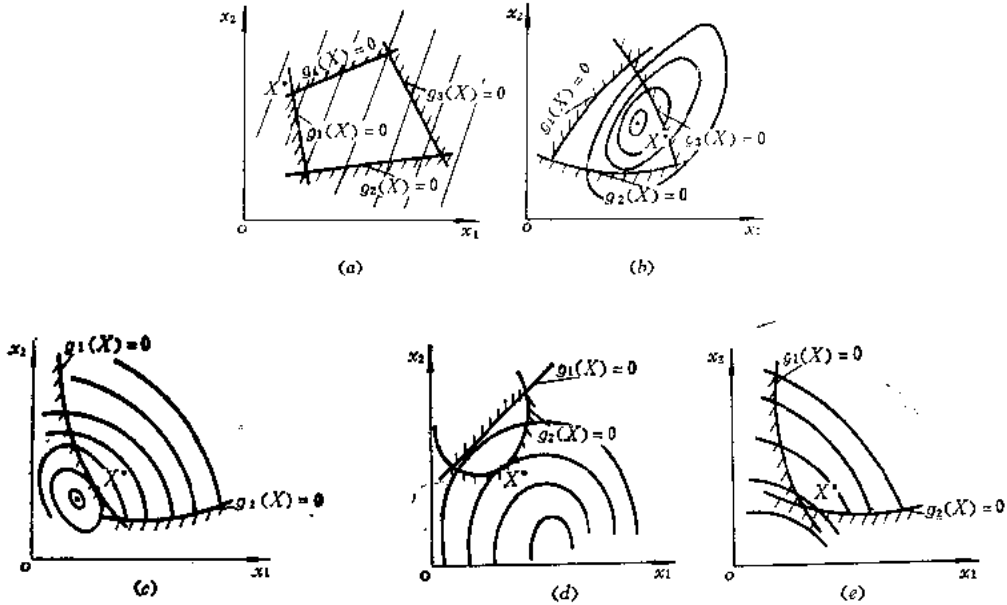


图 1-13

§ 4 优化设计的基本方法

求解优化问题可以用解析解法，也可以用数值的近似解法。解析解法就是把所研究的对象用数学方程（数学模型）描述出来，然后再用数学解析方法（如微分、变分方法等）求出优化解。但是，在很多情况下，优化设计的数学描述比较复杂，因而不便于甚至不可能用解析方法求解；另外，有时对象本身的机理无法用数学方程描述，而只能通过大量试验数据用插值或拟合方法构造一个近似函数式，再来求其优化解，并通过试验来验证；或直接以数学原理为指导，从任取一点出发通过少量试验（探索性的计算），并根据试验计算结果的比较，逐步改进而求得优化解。这种方法是属于近似的、迭代性质的数值解法。数值解法不仅可用于求复杂函数的优化解，也可以用于处理没有数学解析表达式的优化设计问题。因此，它是实际问题中常用的方法，很受重视。其中具体方法较多，并且目前还在发展。但是，应当指出，对于复杂问题，由于不能把所有参数都完全考虑并表示出来，只能是一个近似的最优化的数学描述。由于它本来就是一种近似，那么，采用近似性质的数值方法对它们进行解算，也就谈不到对问题的精确性有什么影响了。

不管是解析解法，还是数值解法，都分别具有针对无约束条件和有约束条件的具体

方法。

可以按照对函数导数计算的要求，把数值方法分为需要计算函数的二阶导数、一阶导数和零阶导数（即只要计算函数值而不须计算其导数）的方法。

在结构优化设计中，大致可分为两类设计方法。一类是优化准则法，它是从一个初始设计 x^i 出发（ i 不是指数，而是上角标， x^i 是 $x^{(i)}$ 的简写），着眼于在每次迭代中应满足的优化条件，接着迭代公式（其中 C_i 为一对角矩阵）

$$x^{i+1} = C_i x^i \quad (1-19)$$

来得到一个改进的设计 x^{i+1} ，而无需再考虑目标函数和约束条件的信息状态。

另一类设计方法是数学规划法，它虽然也是从一个初始设计 x^i 出发，对结构进行分析，但是按照如下迭代公式

$$x^{i+1} = x^i + \Delta x_i \quad (1-20)$$

得到一个改进的设计 x^{i+1} 。

在这类方法中，许多算法是沿着某个搜索方向 d^i 以适当步长 a_i 的方式实现对 x^i 的修改，以获得 Δx_i 值的。此时式（1-20）可写成

$$x^{i+1} = x^i + a_i d^i \quad (1-21)$$

而它的搜索方向 d^i 是根据几何概念和数学原理，由目标函数和约束条件的局部信息状态形成的。也有一些算法是采用直接逼近的迭代方式获得 x^i 的修改量 Δx_i 的。

在数学规划法中，采用式（1-21），即 $x^{K+1} = x^K + a_K d^K$ 进行迭代运算时，求 n 维函数 $f(x) = f(x_1, x_2, \dots, x_n)$ 的极值点的具体算法可以简述如下：

首先，选定初始设计点 x^0 ，从 x^0 出发沿某一规定方向 d^0 求函数 $f(x)$ 的极值点，设此点为 x^1 ；然后，再从 x^1 出发沿某一规定方向 d^1 求函数 $f(x)$ 的极值点，设此点为 x^2 。如此继续，如图（1-14）所示。一般地说，从点 x 出发，沿某一规定方向 d^K 求函数 $f(x^K)$ 的极值点 x^K ($K=1, 2, \dots, n$)。这样的搜索过程就组成求 n 维函数 $f(x)$ 极值（优

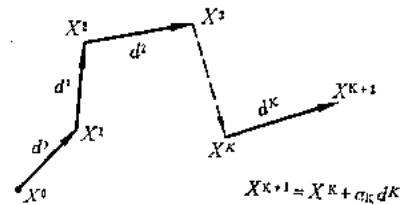


图 1-14

化值)的基本过程。它实际上是通过一系列 (n 个) 的一维搜索过程来完成的。其中的每一次一维搜索过程都可以统一叙述为：在过点 x^K 的 d^K 方向上，求一元函数 $f(x^{K+1}) = f(x^K + a_K d^K)$ 的极值点的问题。既然是在过点 x^K 沿 d^K 方向上求 $f(x^K + a_K d^K)$ 的极值点，那么这里只有 a_K 是唯一的变量。因为无论 a_K 取什么值，点 $x^{K+1} = x^K + a_K d^K$ 总是位于过 x^K 点的 d^K 方向上。所以这个问题就是以 a_K 为变量的一元函数 $\varphi(a_K)$ 求极值的问题。这种一元函数求极值的过程可简称为一维搜索过程，它是确定 a_K 的值使 $f(x^K + a_K d^K)$ 取极值的过程。所以，数学规划法的核心，一是建立搜索方向 d^K ，二是计算最佳步长 a_K 。

由于数值迭代是逐步逼近最优点而获得近似解的，所以要考虑优化问题解的收敛性及迭代过程的终止条件。

收敛性是指某种迭代程序产生的序列 $\{x^K (K=0, 1, \dots)\}$ 收敛于

$$\lim_{K \rightarrow \infty} x^{K+1} = x^K$$

点列 $\{x^K\}$ 收敛的必要和充分条件是：对于任意指定的实数 $\varepsilon > 0$ ，都存在一个只与 ε 有关而与 x 无关的自然数 N ，使得当两自然数 $m, p > N$ 时，满足

$$\|x^m - x^p\| \leq \varepsilon$$

或

$$\sqrt{\sum_{i=1}^n (x_i^m - x_i^p)^2} \leq \varepsilon$$

或

$$|x_i^m - x_i^p| \leq \varepsilon_i = \frac{\varepsilon}{\sqrt{n}}$$

根据这个收敛条件，可以确定迭代终止准则，一般采用以下几种迭代终止准则：

1. 当相邻两设计点的移动距离已达到充分小时。若用向量模计算它的长度，则

$$\|x^{K+1} - x^K\| \leq \varepsilon_1$$

或用 x^{K+1} 和 x^K 的坐标轴分量之差表示为

$$|x_i^{K+1} - x_i^K| \leq \varepsilon_2 \quad (i=1, 2, \dots, n)$$

2. 当函数值的下降量已达到充分小时。即

$$|f(x^{K+1}) - f(x^K)| \leq \varepsilon_3$$

或用其相对值

$$\left| \frac{f(x^{K+1}) - f(x^K)}{f(x^K)} \right| \leq \varepsilon_4$$

3. 当某次迭代点的目标函数梯度已达到充分小时，即

$$\|\nabla f(x^K)\| \leq \varepsilon_5$$

采用哪种收敛准则，可视具体问题而定。

一般地说，采用优化准则法进行结构设计时，由于对其设计的修改较大，所以迭代的收敛速度较快，迭代次数平均为十多次，且与结构的大小无关。因此可用于大型、复杂结构的优化设计，特别是需要利用有限元法进行性能约束计算时较为合适。但是，数学规划法在数学方面有一定的理论基础，它已经发展成为应用数学的一个重要分支。其计算结果的可信程度较高，精确程度也好些。它是优化方法的基础，而且目前优化准则法和数学规划法的解题思路 and 手段实质上也很相似。所以，必须对数学规划法有系统的了解。当然，也没有必要对其中类型繁多的具体方法都进行叙述。这里只着重介绍某些典型的和目前看来比较有效的方法，以期了解一些重要优化方法的思路和实质，达到启发思路、举一反三的目的。

第二章 基础知识

机械结构设计中的优化问题，其设计变量的个数一般都在两个以上，这样它的目标函数就是一个多元函数。高等数学中的极值问题，主要是针对一元或二元函数进行讨论的，因此在讲述机械结构设计优化方法之前，有必要对多元函数的性质及极值条件进行一些补充。此外，高等数学中的极值问题一般采用解析方法求解，而机械结构优化设计多用数值解法。一维搜索是多元函数极值问题数值解法的基础，也在这里进行讨论。所以，本章所讲述的多元函数的性质及一维搜索，既是高等数学极值问题的补充，也是机械结构设计优化方法的基础。

§ 1 梯度的概念

一、方向导数

我们知道，一个二元函数 $f(x_1, x_2)$ 在 $x_0(x_{10}, x_{20})$ 点处的偏导数，其定义是

$$\left. \frac{\partial f}{\partial x_1} \right|_{x_0} = \lim_{\Delta x_1 \rightarrow 0} \frac{f(x_{10} + \Delta x_1, x_{20}) - f(x_{10}, x_{20})}{\Delta x_1}$$

$$\left. \frac{\partial f}{\partial x_2} \right|_{x_0} = \lim_{\Delta x_2 \rightarrow 0} \frac{f(x_{10}, x_{20} + \Delta x_2) - f(x_{10}, x_{20})}{\Delta x_2}$$

而 $\left. \frac{\partial f}{\partial x_1} \right|_{x_0}$ 和 $\left. \frac{\partial f}{\partial x_2} \right|_{x_0}$ 分别是函数 $f(x_1, x_2)$ 在 x_0 点处沿坐标轴 x_1 和 x_2 方向的变化率。因此，函数 $f(x_1, x_2)$ 在 $x_0(x_{10}, x_{20})$ 点处沿某一方向 d (本书采用字母 p 或 d 表示方向) 的变化率如图 (2-1) 所示，其定义应为

$$\left. \frac{\partial f}{\partial d} \right|_{x_0} = \lim_{\Delta d \rightarrow 0} \frac{f(x_{10} + \Delta x_1, x_{20} + \Delta x_2) - f(x_{10}, x_{20})}{\Delta d}$$

称它为该函数沿此方向的方向导数。据此，偏导

数 $\left. \frac{\partial f}{\partial x_1} \right|_{x_0}$ ， $\left. \frac{\partial f}{\partial x_2} \right|_{x_0}$ 也可看成是函数 $f(x_1, x_2)$ 分别沿 x_1 ， x_2 坐标轴方向的方向导数。

所以方向导数是偏导数概念的推广，偏导数是方向导数的特例。

方向导数与偏导数之间的数量关系，可从下述推导中求得

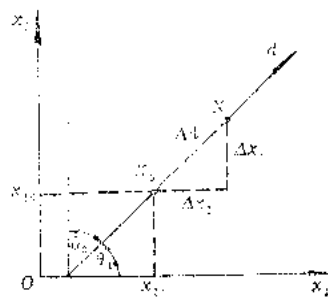


图 2-1

$$\begin{aligned}
\left. \frac{\partial f}{\partial d} \right|_{x_0} &= \lim_{\Delta d \rightarrow 0} \frac{f(x_{10} + \Delta x_1, x_{20} + \Delta x_2) - f(x_{10}, x_{20})}{\Delta d} \\
&= \lim_{\Delta d \rightarrow 0} \frac{f(x_{10} + \Delta x_1, x_{20}) - f(x_{10}, x_{20})}{\Delta x_1} \cdot \frac{\Delta x_1}{\Delta d} \\
&\quad + \lim_{\Delta d \rightarrow 0} \frac{f(x_{10} + \Delta x_1, x_{20} + \Delta x_2) - f(x_{10} + \Delta x_1, x_{20})}{\Delta x_2} \cdot \frac{\Delta x_2}{\Delta d} \\
&= \left. \frac{\partial f}{\partial x_1} \right|_{x_0} \cos \theta_1 + \left. \frac{\partial f}{\partial x_2} \right|_{x_0} \cos \theta_2
\end{aligned}$$

同样，一个三元函数 $f(x_1, x_2, x_3)$ 在 x_0 (x_{10}, x_{20}, x_{30}) 点处沿 d 方向的方向导数 $\left. \frac{\partial f}{\partial d} \right|_{x_0}$

如图 (2-2) 所示，可类似地表示成下面的形式

$$\begin{aligned}
\left. \frac{\partial f}{\partial d} \right|_{x_0} &= \left. \frac{\partial f}{\partial x_1} \right|_{x_0} \cos \theta_1 \\
&\quad + \left. \frac{\partial f}{\partial x_2} \right|_{x_0} \cos \theta_2 + \left. \frac{\partial f}{\partial x_3} \right|_{x_0} \cos \theta_3
\end{aligned}$$

依此类推，即可得到 n 元函数 $f(x_1, x_2, \dots, x_n)$

在 x_0 点处沿 d 方向的方向导数

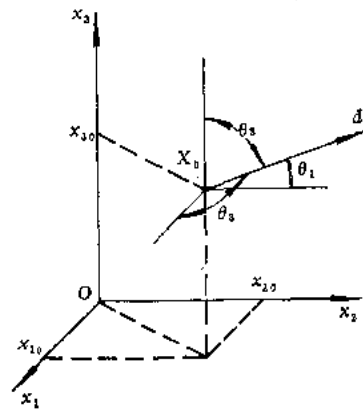


图 2-2

$$\begin{aligned}
\left. \frac{\partial f}{\partial d} \right|_{x_0} &= \left. \frac{\partial f}{\partial x_1} \right|_{x_0} \cos \theta_1 + \left. \frac{\partial f}{\partial x_2} \right|_{x_0} \cos \theta_2 + \dots + \left. \frac{\partial f}{\partial x_n} \right|_{x_0} \cos \theta_n \\
&= \sum_{i=1}^n \left. \frac{\partial f}{\partial x_i} \right|_{x_0} \cos \theta_i
\end{aligned} \tag{2-1}$$

其中的 $\cos \theta_i$ 为 d 方向和坐标轴 x_i 方向之间夹角的余弦。

二、二元函数的梯度

考虑到二元函数具有鲜明的几何解释，并且可以象征性地把这种解释推广到多元函数中去，所以梯度概念的引入也先从二元函数入手。二元函数 $f(x_1, x_2)$ 在 x_0 点处的方向导数 $\left. \frac{\partial f}{\partial d} \right|_{x_0}$ 的表达式可改写成下面的形式

$$\left. \frac{\partial f}{\partial d} \right|_{x_0} = \left. \frac{\partial f}{\partial x_1} \right|_{x_0} \cos \theta_1 + \left. \frac{\partial f}{\partial x_2} \right|_{x_0} \cos \theta_2 = \left[\left. \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right]_{x_0} \begin{Bmatrix} \cos \theta_1 \\ \cos \theta_2 \end{Bmatrix}$$

令

$$\nabla f(x_0) \equiv \begin{Bmatrix} \left. \frac{\partial f}{\partial x_1} \right|_{x_0} \\ \left. \frac{\partial f}{\partial x_2} \right|_{x_0} \end{Bmatrix} = \left[\left. \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right]_{x_0}^T$$

并称它为函数 $f(x_1, x_2)$ 在 x_0 点处的梯度。

设

$$\mathbf{d} = \begin{Bmatrix} \cos\theta_1 \\ \cos\theta_2 \end{Bmatrix}$$

为 \mathbf{d} 方向单位向量, 则有

$$\frac{\partial f}{\partial \mathbf{d}} \Big|_{\mathbf{x}_0} = \nabla f(\mathbf{x}_0)^T \mathbf{d}$$

即函数 $f(x_1, x_2)$ 在 \mathbf{x}_0 点处沿某一方向 \mathbf{d} 的方向导数 $\frac{\partial f}{\partial \mathbf{d}} \Big|_{\mathbf{x}_0}$ 等于函数在该点处的梯度 $\nabla f(\mathbf{x}_0)$ 与 \mathbf{d} 方向单位向量的内积。在这里, 我们可以把向量之间的内积写成向量之间的投影形式, 即

$$\frac{\partial f}{\partial \mathbf{d}} \Big|_{\mathbf{x}_0} = \nabla f(\mathbf{x}_0)^T \mathbf{d} = \|\nabla f(\mathbf{x}_0)\| \cos(\nabla f, \mathbf{d})$$

其中 $\|\nabla f(\mathbf{x}_0)\|$ 代表梯度向量 $\nabla f(\mathbf{x}_0)$ 的模, $\cos(\nabla f, \mathbf{d})$ 表示梯度向量与 \mathbf{d} 方向夹角的余弦。在 \mathbf{x}_0 点处函数沿各方向的方向导数是不同的, 它随 $\cos(\nabla f, \mathbf{d})$ 变化, 即随所取方向的不同而变化。其最大值发生在 $\cos(\nabla f, \mathbf{d})$ 取值为 1 时, 也就是当梯度方向和 \mathbf{d} 方向重合时其值最大。可见梯度方向是函数值变化最快的方向, 而梯度的模就是函数变化率的最大值。

当在 x_1-x_2 平面内画出 $f(x_1, x_2)$ 的等值线

$$f(x_1, x_2) = c$$

(c 为一系列常数) 时, 从图 (2-3) 可以看出, 在 \mathbf{x}_0 处等值线的切线方向 \mathbf{d} 是函数变化率为零的方向, 即有

$$\frac{\partial f}{\partial \mathbf{d}} \Big|_{\mathbf{x}_0} = \|\nabla f(\mathbf{x}_0)\| \cos(\nabla f, \mathbf{d}) = 0$$

所以

$$\cos(\nabla f, \mathbf{d}) = 0$$

可知梯度 $\nabla f(\mathbf{x}_0)$ 和切线方向 \mathbf{d} 垂直, 从而推得梯度方向为等值面的法线方向。梯度

$\nabla f(\mathbf{x}_0)$ 方向为函数变化率最大方向, 也就是最速上升方向。负梯度 $-\nabla f(\mathbf{x}_0)$ 方向为函数变化率取最小值方向, 即最速下降方向。与梯度成锐角的方向为函数上升方向, 与负梯度成锐角的方向为函数下降方向。

例, 求二元函数

$$f(x_1, x_2) = x_1^2 + x_2^2 - 4x_1 - 2x_2 + 5$$

在 $\mathbf{x}_0 = (0, 0)^T$ 处函数变化率最大的方向和数值。由于函数变化率最大的方向是梯度方向, 这里用单位向量 \mathbf{p} 表示, 函数变化率最大的数值是梯度的模 $\|\nabla f(\mathbf{x}_0)\|$ 。求 $f(x_1, x_2)$ 在 \mathbf{x}_0 点处的梯度方向和数值, 计算如下

$$\nabla f(\mathbf{x}_0) = \begin{Bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{Bmatrix} \Big|_{\mathbf{x}_0} = \begin{Bmatrix} 2x_1 - 4 \\ 2x_2 - 2 \end{Bmatrix} \Big|_{\mathbf{x}_0} = \begin{Bmatrix} -4 \\ -2 \end{Bmatrix}$$

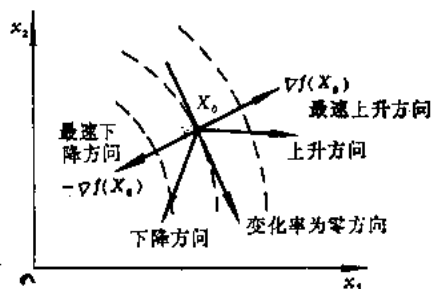


图 2-3

$$\|\nabla f(x_0)\| = \sqrt{\left(\frac{\partial f}{\partial x_1}\right)^2 + \left(\frac{\partial f}{\partial x_2}\right)^2} = \sqrt{(-4)^2 + (-2)^2} = 2\sqrt{5}$$

$$p = \frac{\nabla f(x_0)}{\|\nabla f(x_0)\|} = \frac{\begin{Bmatrix} -4 \\ -2 \end{Bmatrix}}{2\sqrt{5}} = \begin{Bmatrix} -\frac{2}{\sqrt{5}} \\ 1 \\ -\frac{2}{\sqrt{5}} \end{Bmatrix}$$

在 x_1-x_2 平面上画出函数等值线和 $x_0(0,0)$ 点处的梯度方向 p , 如图 (2-4) 所示。从图中可以看出, 在 x_0 点函数变化率最大的方向 p 即为等值线的法线方向, 也就是同心圆的半径方向。

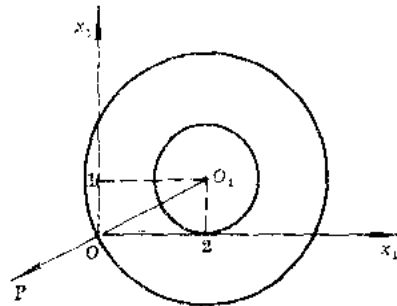


图 2-4

三、多元函数的梯度

将二元函数推广到多元函数, 对于函数 $f(x_1, x_2, \dots, x_n)$ 在 $x_0(x_{10}, x_{20}, \dots, x_{n0})$ 处的梯度 $\nabla f(x_0)$, 可定义为

$$\nabla f(x_0) \equiv \begin{Bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{Bmatrix}_{x_0} = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]_{x_0}^T \quad (2-2)$$

对于 $f(x_1, x_2, \dots, x_n)$ 在 x_0 处沿 d 的方向导数可表示为

$$\frac{\partial f}{\partial d} \Big|_{x_0} = \sum_{i=1}^n \frac{\partial f}{\partial x_i} \Big|_{x_0} \cos\theta_i = \nabla f(x_0)^T d = \|\nabla f(x_0)\| \cos(\nabla f, d) \quad (2-3)$$

其中

$$d \equiv \begin{Bmatrix} \cos\theta_1 \\ \cos\theta_2 \\ \vdots \\ \cos\theta_n \end{Bmatrix}$$

为 d 方向上的单位向量。

$$\|\nabla f(x_0)\| = \left[\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} \right)_{x_0}^2 \right]^{\frac{1}{2}}$$

为梯度 $\nabla f(x_0)$ 的模。

$$p = \frac{\nabla f(x_0)}{\|\nabla f(x_0)\|}$$

为梯度方向单位向量, 它与函数等值面 $f(x) = c$ 相垂直, 也就是和等值面上过 x_0 的一切曲线相垂直, 如图 (2-5) 所示。



图 2-5

§ 2 多元函数的泰勒 (Taylor) 展开

多元函数的泰勒展开在优化方法中十分重要,许多方法及其收敛性证明都是从它出发的。

一元函数 $f(x)$ 在 $x=x_0$ 点处的泰勒展开式为

$$f(x) = f(x_0) + f'(x_0)\Delta x + \frac{1}{2}f''(x_0)\Delta x^2 + \dots$$

其中 $\Delta x \equiv x - x_0$, $\Delta x^2 \equiv (x - x_0)^2$

二元函数 $f(x_1, x_2)$ 在 $x_0(x_{10}, x_{20})$ 点处的泰勒展开式为

$$\begin{aligned} f(x_1, x_2) = & f(x_{10}, x_{20}) + \left. \frac{\partial f}{\partial x_1} \right|_{x_0} \Delta x_1 + \left. \frac{\partial f}{\partial x_2} \right|_{x_0} \Delta x_2 \\ & + \frac{1}{2} \left(\left. \frac{\partial^2 f}{\partial x_1^2} \right|_{x_0} \Delta x_1^2 + 2 \left. \frac{\partial^2 f}{\partial x_1 \partial x_2} \right|_{x_0} \Delta x_1 \Delta x_2 + \left. \frac{\partial^2 f}{\partial x_2^2} \right|_{x_0} \Delta x_2^2 \right) + \dots \end{aligned}$$

其中 $\Delta x_1 \equiv x_1 - x_{10}$, $\Delta x_2 \equiv x_2 - x_{20}$

将上述展开式写成矩阵形式,有

$$\begin{aligned} f(x) = & f(x_0) + \left[\left. \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right]_{x_0} \begin{Bmatrix} \Delta x_1 \\ \Delta x_2 \end{Bmatrix} \\ & + \frac{1}{2} [\Delta x_1, \Delta x_2] \begin{bmatrix} \left. \frac{\partial^2 f}{\partial x_1^2} \right|_{x_0} & \left. \frac{\partial^2 f}{\partial x_1 \partial x_2} \right|_{x_0} \\ \left. \frac{\partial^2 f}{\partial x_2 \partial x_1} \right|_{x_0} & \left. \frac{\partial^2 f}{\partial x_2^2} \right|_{x_0} \end{bmatrix} \begin{Bmatrix} \Delta x_1 \\ \Delta x_2 \end{Bmatrix} + \dots \\ = & f(x_0) + \nabla f(x_0)^T \Delta x + \frac{1}{2} \Delta x^T \mathbf{G}(x_0) \Delta x + \dots \end{aligned} \quad (2-4)$$

其中

$$\mathbf{G}(x_0) \equiv \begin{bmatrix} \left. \frac{\partial^2 f}{\partial x_1^2} \right|_{x_0} & \left. \frac{\partial^2 f}{\partial x_1 \partial x_2} \right|_{x_0} \\ \left. \frac{\partial^2 f}{\partial x_2 \partial x_1} \right|_{x_0} & \left. \frac{\partial^2 f}{\partial x_2^2} \right|_{x_0} \end{bmatrix}$$

称作函数 $f(x_1, x_2)$ 在 x_0 点处的海赛 (Hessian) 矩阵,它是由函数 $f(x_1, x_2)$ 在 x_0 处的二阶偏导数所组成的方阵。由于函数的二次连续性,有

$$\left. \frac{\partial^2 f}{\partial x_1 \partial x_2} \right|_{x_0} = \left. \frac{\partial^2 f}{\partial x_2 \partial x_1} \right|_{x_0}$$

所以 $\mathbf{G}(x_0)$ 矩阵为对称方阵。

例: 求二元函数

$$f(x_1, x_2) = x_1^2 + x_2^2 - 4x_1 - 2x_2 + 5$$

在 $x_0 = \begin{Bmatrix} x_{10} \\ x_{20} \end{Bmatrix} = \begin{Bmatrix} 2 \\ 1 \end{Bmatrix}$ 点处的二阶泰勒展开式。

二阶泰勒展开式为

$$f(x_1, x_2) = f(x_{10}, x_{20}) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2} (x - x_0)^T \mathbf{G}(x_0) (x - x_0)$$

将 x_0 的具体数值代入, 有

$$f(x_{10}, x_{20}) = 0$$

$$\nabla f(x_0) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}_{x_0} = \begin{bmatrix} 2x_1 - 4 \\ 2x_2 - 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0$$

(上式黑体字“0”代表零向量)

$$G(x_0) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}_{x_0} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

所以得

$$f(x_1, x_2) = \frac{1}{2} [x_1 - x_{10}, x_2 - x_{20}] G(x_0) \begin{Bmatrix} x_1 - x_{10} \\ x_2 - x_{20} \end{Bmatrix}$$

$$= \frac{1}{2} [x_1 - 2, x_2 - 1] \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{Bmatrix} x_1 - 2 \\ x_2 - 1 \end{Bmatrix}$$

$$= (x_1 - 2)^2 + (x_2 - 1)^2$$

此函数的图象是以 x_0 点为顶点的旋转抛物物面, 如图 (2-6) 所示。

将二元函数的泰勒展开式推广到多元函数时, 则 $f(x_1, x_2, \dots, x_n)$ 在 x_0 点处泰勒展开式的矩阵形式为

$$f(x) = f(x_0) + \nabla f(x_0)^T \Delta x + \frac{1}{2} \Delta x^T G(x_0) \Delta x + \dots$$

其中

$$\nabla f(x_0) = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \dots \quad \frac{\partial f}{\partial x_n} \right]_{x_0}^T$$

为函数 $f(x)$ 在 x_0 点处的梯度,

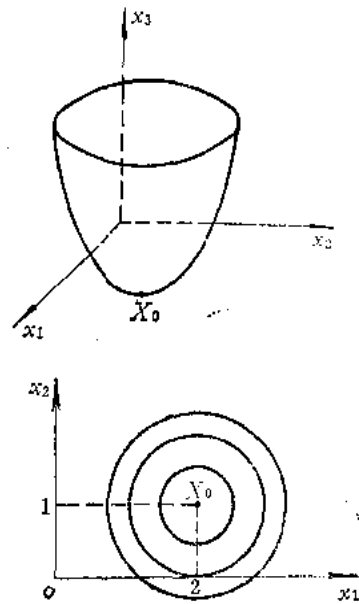


图 2-6

$$G(x_0) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}_{x_0} \quad (2-5)$$

成

为函数 $f(x)$ 在 x_0 点处的海赛矩阵。

若将函数的泰勒展开式只取到线性项，即取

$$z(x) = f(x_0) + \nabla f(x_0)^T (x - x_0)$$

则 $z(x)$ 是过 x_0 点和函数 $f(x)$ 所代表的超曲面相切的切平面。

§ 3 多元函数的极值条件

优化设计方法是使目标函数取得极值，所谓极值是指函数取得极大值或极小值。

我们知道，对于可微的一元函数 $f(x)$ ，在给定区间内某点 $x = x_0$ 处取得极值，其必要条件是

$$f'(x_0) = 0$$

即函数的极值必须在驻点处取得。此条件是必要的，但不充分，也就是说驻点不一定是极值点。检验驻点是否为极值点，一般用二阶导数的符号来判断。若 $f''(x_0) > 0$ ，则 x_0 为极小点；若 $f''(x_0) < 0$ 则 x_0 为极大点。若 $f''(x_0) = 0$ ， x_0 是否为极值点，还需逐次检验其更高阶导数的符号。开始不为零的导数阶数若为偶次，则为极值点，若为奇次，则为拐点，而不是极值点。

对于二元函数 $f(x_1, x_2)$ ，若在 $x_0(x_{10}, x_{20})$ 点处取得极值，其必要条件是

$$\left. \frac{\partial f}{\partial x_1} \right|_{x_0} = \left. \frac{\partial f}{\partial x_2} \right|_{x_0} = 0$$

即
$$\nabla f(x_0) = 0$$

为了判断从上述必要条件求得的 x_0 是否是极值点，需要建立极值的充分条件。根据二元函数 $f(x_1, x_2)$ 在 x_0 点处的泰勒展开式，考虑上述极值必要条件，有

$$\begin{aligned} f(x_1, x_2) = & f(x_{10}, x_{20}) + \frac{1}{2} \left(\left. \frac{\partial^2 f}{\partial x_1^2} \right|_{x_0} \Delta x_1^2 \right. \\ & \left. + 2 \left. \frac{\partial^2 f}{\partial x_1 \partial x_2} \right|_{x_0} \Delta x_1 \Delta x_2 + \left. \frac{\partial^2 f}{\partial x_2^2} \right|_{x_0} \Delta x_2^2 \right) + \dots \end{aligned}$$

设 $A = \left. \frac{\partial^2 f}{\partial x_1^2} \right|_{x_0}$ ， $B = \left. \frac{\partial^2 f}{\partial x_1 \partial x_2} \right|_{x_0}$ ， $C = \left. \frac{\partial^2 f}{\partial x_2^2} \right|_{x_0}$

则
$$\begin{aligned} f(x_1, x_2) = & f(x_{10}, x_{20}) + \frac{1}{2} (A \Delta x_1^2 + 2B \Delta x_1 \Delta x_2 + C \Delta x_2^2) + \dots \\ = & f(x_{10}, x_{20}) + \frac{1}{2A} [(A \Delta x_1 + B \Delta x_2)^2 + (AC - B^2) \Delta x_2^2] + \dots \end{aligned}$$

若 $f(x_1, x_2)$ 在 x_0 点处取得极小值，则要求在 x_0 点附近的一切点 x 均须满足

$$f(x_1, x_2) - f(x_{10}, x_{20}) > 0$$

即要求
$$\frac{1}{2A} [(A \Delta x_1 + B \Delta x_2)^2 + (AC - B^2) \Delta x_2^2] > 0$$

或者要求 $A > 0$ ， $AC - B^2 > 0$

也就是

$$\frac{\partial^2 f}{\partial x_1^2} \Big|_{x_0} > 0$$

$$\left[\frac{\partial^2 f}{\partial x_1^2} \cdot \frac{\partial^2 f}{\partial x_2^2} - \left(\frac{\partial^2 f}{\partial x_1 \partial x_2} \right)^2 \right] \Big|_{x_0} > 0$$

此条件反映了 $f(x_1, x_2)$ 在 x_0 点处的海赛矩阵 $G(x_0)$ 的各阶主子式均大于零, 即对于

$$G(x_0) = \begin{vmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{vmatrix} \Big|_{x_0}$$

要求

$$\frac{\partial^2 f}{\partial x_1^2} \Big|_{x_0} > 0$$

$$|G(x_0)| = \begin{vmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{vmatrix} \Big|_{x_0} > 0$$

所以, 二元函数在某点处取得极值的充分条件是要求在该点处的海赛矩阵为正定。

例: 求函数

$$f(x_1, x_2) = x_1^2 + x_2^2 - 4x_1 - 2x_2 + 5$$

的极值。

首先, 根据极值的必要条件求驻点。

$$\nabla f(x) = \begin{Bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{Bmatrix} = \begin{Bmatrix} 2x_1 - 4 \\ 2x_2 - 2 \end{Bmatrix} = 0$$

得驻点为

$$x_0 = \begin{Bmatrix} x_{10} \\ x_{20} \end{Bmatrix} = \begin{Bmatrix} 2 \\ 1 \end{Bmatrix}$$

再根据极值的充分条件, 判断此驻点是否为极值点。由于

$$G(x_0) = \begin{vmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{vmatrix} \Big|_{x_0} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

则 $G(x_0)$ 的一阶主子式

$$\frac{\partial^2 f}{\partial x_1^2} \Big|_{x_0} = 2 > 0$$

和二阶主子式

$$|G(x_0)| = \begin{vmatrix} 2 & 0 \\ 0 & 2 \end{vmatrix} = 4 > 0$$

均大于零, 故 $G(x_0)$ 为正定矩阵。

$$x_0 = \left\{ \begin{array}{c} 2 \\ 1 \end{array} \right\}$$

为极小点，相应的极值为

$$f(x_0) = 0$$

对于多元函数 $f(x_1, x_2, \dots, x_n)$ ，若在 x^* 点处取得极值，则极值的必要条件为

$$\nabla f(x^*) = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \dots \quad \frac{\partial f}{\partial x_n} \right]_{x^*}^T = 0 \quad (2-6)$$

极值的充分条件为

$$G(x^*) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}_{x^*} \quad \text{正定} \quad (2-7)$$

即要求 $G(x^*)$ 的下列各阶主子式均大于零

$$\left. \frac{\partial^2 f}{\partial x_1^2} \right|_{x^*} > 0$$

$$\begin{vmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{vmatrix}_{x^*} > 0$$

$$\begin{vmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} & \frac{\partial^2 f}{\partial x_3 \partial x_2} & \frac{\partial^2 f}{\partial x_3^2} \end{vmatrix}_{x^*} > 0$$

$$\dots \dots \dots$$

$$|G(x^*)| > 0$$

一般说来，多元函数的极值条件在优化方法中仅具有理论意义。因为对于复杂的目标函数，海赛矩阵不易求得，它的正定性就更难判定了。

§ 4 函数的凸性

根据函数极值条件所确定的极小点 x^* ，是指函数 $f(x)$ 在 x^* 附近的一切 x 均满足不等式

$$f(x) > f(x^*)$$

所以称函数 $f(x)$ 在 x^* 处取得局部极小值，称 x^* 为局部极小点。（有时在局部极小

值和局部极小点前还加上“严格”二字，以区别于满足不等式 $f(x) \geq f(x^*)$ 的情况。) 因此，根据函数极值条件所确定的极小点只是反映函数在 x^* 附近的局部性质。

优化问题一般是要求目标函数在某一区域内的最小点，也就是要求全局极小点。函数的局部极小点并不一定就是全局极小点，只有函数具备某种性质时，二者才等同。因此对局部极小和全局极小之间的关系应该作进一步的说明。

关于这个问题，从一元函数的情况可以得到启发。设 $f(x)$ 为定义在区间 $[a, b]$ 上的一元函数，如果它的图形是下凸的，从图(2-7)中容易看出它的极小点 x^* ，同时也是函数 $f(x)$ 在区间 $[a, b]$ 上的最小点。我们称这样的函数具有凸性。如果 $f(x)$ 具有二阶导数，且 $f''(x) \geq 0$ ，则函数 $f(x)$ 向下凸，这说明函数的凸性可由二阶导数的符号来判断。为了研究多元函数 $f(x)$ 的凸性，需要首先阐明函数定义域所应具有的性质，所以先介绍凸集的概念。

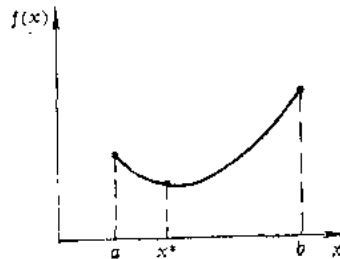


图 2-7

一、凸集

一个点集(或区域)，如果连结其中任意两点 x_1 和 x_2 的线段都全部包含在该集合内，就称该点集为凸集，否则称非凸集，如图(2-8)所示。凸集的概念可以用数学的语言简练地表示为：如果对一切 $x_1 \in R, x_2 \in R$ 及一切满足 $0 \leq a \leq 1$ 的实数 a ，点 $ax_1 + (1-a)x_2 \equiv y \in R$ ，则称集合 R 为凸集。凸集既可以是有限界的，也可以是无界的。 n 维空间中的 r 维子空间也是凸集(例如三维空间中的平面)。

凸集具有以下性质：

若 A 是一个凸集， β 是一个实数， a 是凸集 A 中的动点，即 $a \in A$ ，则集合

$$\beta A = \{x: x = \beta a, a \in A\}$$

还是凸集。

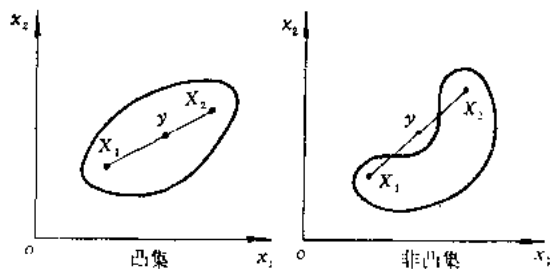


图 2-8

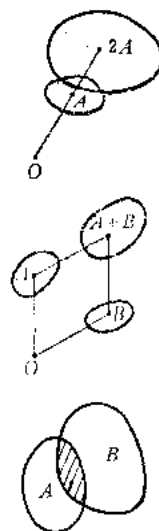


图 2-9

若 A 和 B 是凸集， a, b 分别是凸集 A, B 中的动点，即 $a \in A, b \in B$ ，则集合 $A+B = \{x: x = a+b, a \in A, b \in B\}$ 还是凸集。

任何一组凸集的交集还是凸集。这三个性质如图(2-9)所示。

二、凸函数

函数 $f(x)$, 如果在连结其凸集定义域内任意两点 x_1, x_2 的线段上, 函数值总小于或等于用 $f(x_1)$ 及 $f(x_2)$ 作线性内插所得的值, 那么称 $f(x)$ 为凸函数。用数学语言表达为

$$f[ax_1 + (1-a)x_2] \leq af(x_1) + (1-a)f(x_2) \quad (2-8)$$

其中 $0 \leq a \leq 1$

若上二式均去掉等号, 则 $f(x)$ 称作严格凸函数。一元函数 $f(x)$ 若在 $[a, b]$ 内为凸函数, 其函数图象表现为在曲线上任意两点所连的直线不会落在曲线弧以下, 如图(2-10)所示。

下面给出凸函数的一些简单性质:

设 $f(x)$ 为定义在凸集 R 上的一个凸函数, 对任意实数 $a > 0$, 则函数 $af(x)$ 也是定义在 R 上的凸函数。

设 $f_1(x)$ 和 $f_2(x)$ 为定义在凸集 R 上的两个凸函数, 则其和 $f_1(x) + f_2(x)$ 也是 R 上的凸函数。

对任意两个正数 α 和 β , 函数 $\alpha f_1(x) + \beta f_2(x)$ 也是在 R 上的凸函数。

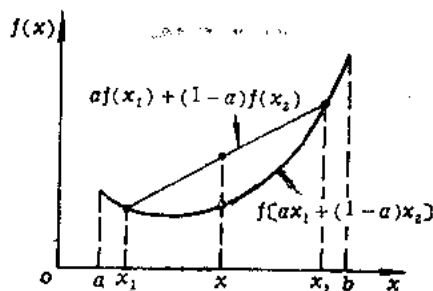


图 2-10

三、凸性条件

现在我们来研究怎样判断一个函数是否具有凸性的问题。

设 $f(x)$ 为定义在凸集 R 上, 且具有连续一阶导数的函数, 则 $f(x)$ 在 R 上为凸函数的充分必要条件是对凸集 R 内任意不同两点 x_1, x_2 , 不等式

$$f(x_2) \geq f(x_1) + (x_2 - x_1)^T \nabla f(x_1) \quad (2-9)$$

恒成立。

这是根据函数的一阶导数信息——函数的梯度 $\nabla f(x)$ 来判断函数的凸性。也可以用二阶导数信息——函数的海赛矩阵 $G(x)$ 来判断函数的凸性。

设 $f(x)$ 为定义在凸集 R 上且具有连续二阶导数的函数, 则 $f(x)$ 在 R 上为凸函数的充分必要条件是海赛矩阵 $G(x)$ 在 R 上处处半正定。(证明可查阅参考文献[22])

四、局部极小和全局极小的关系

设 $f(x)$ 为定义在凸集 R 上的一个凸函数, 则 $f(x)$ 的任何局部极小点同时也必为 $f(x)$ 在 R 上的全局极小点。

证明: 设 x_1 为 $f(x)$ 在 R 上的一个极小点, 因为 $f(x)$ 为 R 上的凸函数, 则对于 R 上的任意点 x_2 有

$$f(x_2) \geq f(x_1) + (x_2 - x_1)^T \nabla f(x_1)$$

由于 x_1 为极小点, $x_2 - x_1$ 为一允许上升方向, 因此有

$$(x_2 - x_1)^T \nabla f(x_1) \geq 0$$

从而可得

$$f(x_2) \geq f(x_1)$$

知 x_1 为全局极小点。

§ 5 坐标变换及二次型函数的主轴线

一、坐标变换

在优化问题中，为了简化优化过程，可对二次型函数形式的原目标函数进行坐标变换。这种变换的目的是：

1. 尽可能使目标函数的等值线或等值面变成椭圆形的或超椭圆形的，甚至变成圆形或超球形的，这就是改变函数的偏心率。

2. 尽量使目标函数的各个独立变量之间没有交互相乘的作用，如把目标函数变换成为一系列独立项的平方和的形式等。

一般地说，一个二次型函数，可以通过原来的坐标中心平移（改变或消去一次项和常数项）、坐标轴旋转（消去交互相乘项）和坐标尺度的伸缩等的变换来简化原函数，从而简化优化的过程。图（2—11）所示是一个二维图形通过平移、旋转、伸缩等坐标和尺度变换后的变化情形。

下面再对第二项目的，即尽量使目标函数的各个独立变量之间没有交互相乘的作用，做一简单说明。

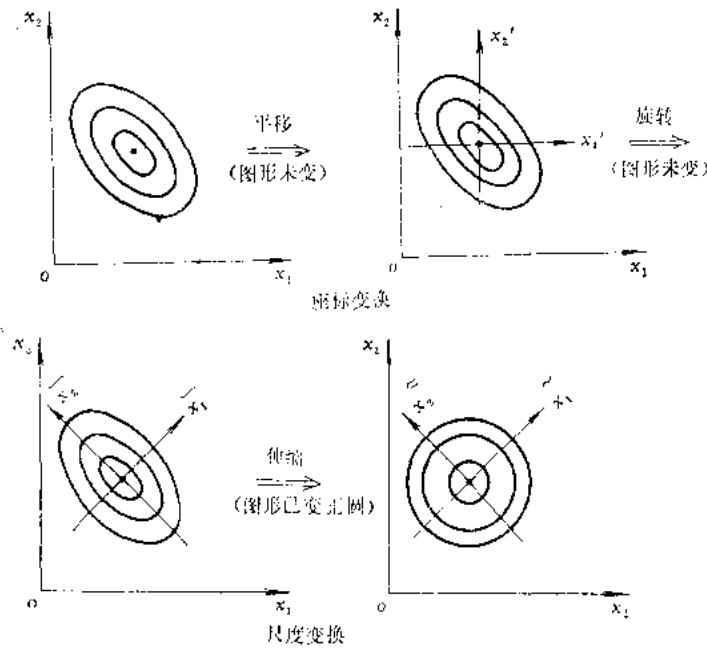


图 2—11

由《线性代数》可知，若二次型函数 $x^T G x$ 的系数矩阵 G 的特征值是 $\lambda_1, \lambda_2, \dots, \lambda_n$ ，则存在正交矩阵 Q ，使得 G 化为对角矩阵，即有

$$Q^T G Q = \begin{vmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{vmatrix} = \Lambda \quad (2-10)$$

而对于实二次型 $f(\mathbf{x}) = \mathbf{x}^T \mathbf{G} \mathbf{x} = \sum_{i=1}^n \sum_{j=1}^n g_{ij} x_i x_j$ (其中的 $\mathbf{G} = \mathbf{G}^T$ 是对称矩阵), 一定可以找到一个正交矩阵 \mathbf{Q} , 通过 $\mathbf{x} = \mathbf{Q} \mathbf{y}$ 的尺度变换使所给二次型变为标准形式, 即有

$$\mathbf{x}^T \mathbf{G} \mathbf{x} = \mathbf{y}^T \mathbf{Q}^T \mathbf{G} \mathbf{Q} \mathbf{y} = \mathbf{y}^T \mathbf{\Lambda} \mathbf{y} = \sum_{i=1}^n \lambda_i y_i^2 \quad (2-11)$$

式中的 $\lambda_1, \lambda_2, \dots, \lambda_n$ 是对称矩阵 \mathbf{G} 的特征值, 且为实数。若变换后所得的 \mathbf{G} 是单位矩阵, 则称此标准形式为规范形式。

例如, 对于二次型函数 $f(\mathbf{x}) = 5x_1^2 + 5x_2^2 - 8x_1x_2$ 可以写成

$$f(\mathbf{x}) = [x_1 \ x_2] \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix}$$

当采用

$$\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \frac{\sqrt{2}}{6} \begin{bmatrix} 3 & -1 \\ 3 & 1 \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix}$$

的尺度变换, 则有

$$\begin{aligned} f(\mathbf{x}) &= [y_1 \ y_2] \frac{\sqrt{2}}{6} \begin{bmatrix} 3 & 3 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix} \frac{\sqrt{2}}{6} \begin{bmatrix} 3 & -1 \\ 3 & 1 \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix} \\ &= [y_1 \ y_2] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix} = y_1^2 + y_2^2 = \varphi(\mathbf{y}) \end{aligned}$$

这说明, $f(\mathbf{x}) = 5x_1^2 + 5x_2^2 - 8x_1x_2$ 可以变换为 $\varphi(\mathbf{y}) = y_1^2 + y_2^2$ 。而这时 $\varphi(\mathbf{y})$ 的等值线已经是一个以 $(0, 0)$ 为中心的圆了。即通过尺度变换改变了原来二次型函数的偏心率, 从而使优化过程更简捷了。

对于一般的二次型函数 $f(\mathbf{x}) = a + \mathbf{b}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x}$ (其中的 a 是常数, \mathbf{b} 是常数列阵), 同样, 可以经过尺度变换使之变成一系列独立项的平方和的形式。这时, 经过 $\mathbf{x} = \mathbf{Q} \mathbf{y}$ 的尺度变换, 有

$$\varphi(\mathbf{y}) = a + \mathbf{b}^T \mathbf{Q} \mathbf{y} + \frac{1}{2} \mathbf{y}^T \mathbf{Q}^T \mathbf{G} \mathbf{Q} \mathbf{y} \quad (2-12)$$

其中的 $\mathbf{b}^T \mathbf{Q}$ 可以写成

$$[b_1, b_2, \dots, b_n] \begin{pmatrix} q_{11} & q_{12} & \dots & q_{1n} \\ q_{21} & q_{22} & \dots & q_{2n} \\ \dots & \dots & \dots & \dots \\ q_{n1} & q_{n2} & \dots & q_{nn} \end{pmatrix} = [d_1, d_2, \dots, d_n] = \mathbf{d}^T \quad (2-13)$$

利用此式, 则可写出

$$\mathbf{b}^T \mathbf{Q} \mathbf{y} = \mathbf{d}^T \mathbf{y} = [d_1, d_2, \dots, d_n] \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{Bmatrix} = \sum_{i=1}^n d_i y_i \quad (2-14)$$

把式 (2-14) 代入 (2-12), 则得

$$\varphi(\mathbf{y}) = a + \sum_{i=1}^n d_i y_i + \frac{1}{2} \sum_{i=1}^n \lambda_i y_i^2 \quad (2-15)$$

可以把式(2-15)中的常数值 a 进行分解,使之满足下式的配方要求

$$\frac{1}{2} \sum_{i=1}^n \lambda_i \left[y_i^2 + 2 \frac{d_i}{\lambda_i} y_i + \left(\frac{d_i}{\lambda_i} \right)^2 \right]$$

即把 a 分解为 $a = c + \frac{1}{2} \sum_{i=1}^n \frac{d_i^2}{\lambda_i}$, 则得

$$\varphi(\mathbf{y}) = c + \frac{1}{2} \sum_{i=1}^n \lambda_i \left(y_i + \frac{d_i}{\lambda_i} \right)^2 \quad (2-16)$$

可见,通过求 \mathbf{G} 的特征值的办法,可以使二次型函数变为包括有 y_i^2 或 $\left(y_i + \frac{d_i}{\lambda_i} \right)^2$ 的一些独立项的平方和的形式,从而消去各独立变量之间的交互相乘作用,也改变了原函数的偏心率,使原函数的等值线(面)变成圆形或球形。

这样,可以就每一项分别进行极值(优化值)的计算,即把问题变换成可以分别在 n 个单方向上进行极值搜索的问题,最后达到求二次型函数极值的目的。而且,这也把二次型函数的优化问题和矩阵的特征值问题联系起来。

例如,在式(2-16)中,当逐一地使 $y_i + \frac{d_i}{\lambda_i} = 0$, 或逐一地改变 y_i 的值,直到 $y_i = -\frac{d_i}{\lambda_i}$ 时, $\varphi(\mathbf{y})$ 就达到极小值。

不断改变 y_i 的值使之达到 $y_i = -\frac{d_i}{\lambda_i}$ 的过程,实际上是沿着 y_i 对应的方向(可看成是搜索方向)进行单方向的一维搜索,求出在该方向上函数的极值。

除了可以通过上述一些变换方法使目标函数简化外,还可采用泰勒级数展开式,并根据需要仅取至一阶项或二阶项,达到使原目标函数降阶的目的。

二、二次型函数的主轴线

所有二次型函数都具有一组相互正交的主轴线(正交轴系),轴的数目和函数的维数相等。例如椭圆就有长轴和短轴两个主轴。

对于以原点为中心的椭球面,或超椭球面,它的方程可以写成

$$\frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} = c \quad (c > 0) \quad (2-17)$$

其中 \mathbf{G} 是 $n \times n$ 阶对称正定矩阵。

该(超)椭球面上的 \mathbf{x} 点处的法线(梯度)方向为 $\mathbf{G}\mathbf{x}$ 。如果这个法线方向 $\mathbf{G}\mathbf{x}$ 和向量 \mathbf{x} 共线或平行,即有关系式

$$\mathbf{G}\mathbf{x} = \lambda \mathbf{x} \quad (2-18)$$

时,则此法线方向 $\mathbf{G}\mathbf{x}$ 称为(超)球面 $\frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} = c$ 的主轴方向。

当把 \mathbf{G} 经过变换使之对角化后,就有 $\mathbf{G}\mathbf{x} = \lambda \mathbf{x}$ 的关系(这里的 λ 是对角矩阵的元

素)。因此, $Gx = \lambda x$ 说明, (超)球面 $\frac{1}{2}x^T Gx = c$ 的主轴方向实际上就是矩阵 G 的特征向量系。而比例因子 λ 是矩阵 G 的与特征向量 x 相应的特征值。

将会看到, 可以把原来是平行于坐标轴线的一组基向量 $e_1 = (1, 0, 0, \dots, 0)$, $e_2 = (0, 1, 0, \dots, 0)$, $\dots, e_n = (0, 0, \dots, 1)$ 通过变换获得一组平行于二次型函数的主轴线(相当于采用旋转方法求矩阵的特征值和特征向量), 从而形成一个新的相互共轭(正交)的方向系, 且可以在不超过 n 次沿这些共轭方向进行极小化后, 使极值点收敛于曲线的中心。这里又把二次型函数的优化问题与求矩阵的特征值和特征向量(或共轭方向)系的问题联系起来。

顺便谈一下极小化方法的收敛性问题。对一般二次型函数应用某一种极小化方法经过一定次数(其具体次数和函数变量的个数有关)的运算后就可以达到极小点, 那么这个方法或计算过程是二次收敛的。因为, 对函数的系数矩阵 G 共轭的方向组是它的主轴系, 所以, 以共轭方向为基础的一些优化方法都具有二次型函数的收敛性。

§ 6 一维搜索区间的确定和缩短

在第一章中曾经指出, 当采用数学规划法求多元函数 $f(x)$ 的极值点 x^* 时, 一般要进行一系列的迭代计算

$$x^{K+1} = x^K + \alpha_K d^K \quad (K = 0, 1, 2, \dots)$$

其中 d^K 为第 $K+1$ 次迭代的搜寻方向, α_K 为沿 d^K 搜索的最佳步长。当方向 d^K 给定, 求最佳步长 α_K 时, 就是求一元函数

$$f(x^{K+1}) = f(x^K + \alpha_K d^K) = \varphi(\alpha_K)$$

的极值问题, 它称作一维搜索。而求多元函数极值点, 需要进行一系列的一维搜索。可见一维搜索是优化搜寻方法的基础。

欲求一元函数 $f(a)$ 的极小点 a^* (为书写简便, 这里仍用同一符号 f 表示相应的一元函数), 首先必须确定 a^* 所在的区间, 然后再不断缩小此区间, 从而求得 a^* 的数值近似解。

一、确定搜索区间的外推法

在一维搜索时, 我们假设函数 $f(a)$ 具有如图(2-12)所示的单谷性, 即在所考虑的区间内部, 函数 $f(a)$ 有唯一的极小点 a^* 。为了确定极小点 a^* 所在的区间 $[a, b]$, 应使函数 $f(a)$ 在 $[a, b]$ 区间里形成“高一低一高”趋势。

为此, 从 $a=0$ 开始, 以初始步长 h_0 向前试探。如果函数值上升, 则步长变号, 即改变试探方向。如果函数值下降, 则维持原来的试探方向, 并将步长加倍。区间的始点、中间点依次沿试探方向移动一步。此过程一直进行到函数值再次上升时为止, 即可找到搜索区间的终点。最后得到的三点即为搜索区间的始点、中间点和终点, 形成函数值的“高一低一高”趋势。

图(2-13)表示沿 a 的正向试探。每走一步都将区间的始点, 中间点沿试探方向移动一步(进行换名)。经过三步最后确定搜索区间 $[a_1, a_3]$, 并且得到区间始点、中

间点和终点 $a_1 < a_2 < a_3$ 所对应的函数值 $y_1 > y_2 < y_3$ 。

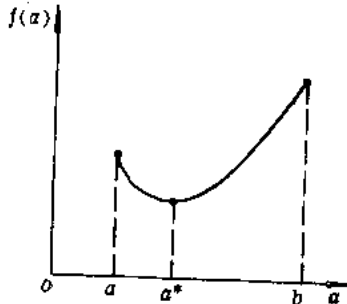


图 2-12

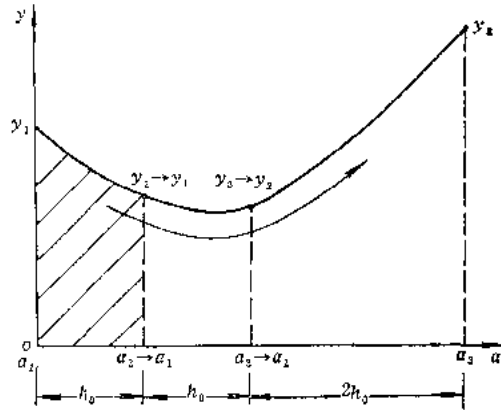


图 2-13

图 (2-14) 所表示的情况是, 开始是沿 a 的正方向试探, 但由于函数值上升而改变了试探方向, 最后得到始点, 中间点和终点 $a_1 > a_2 > a_3$ 及它们的对应函数

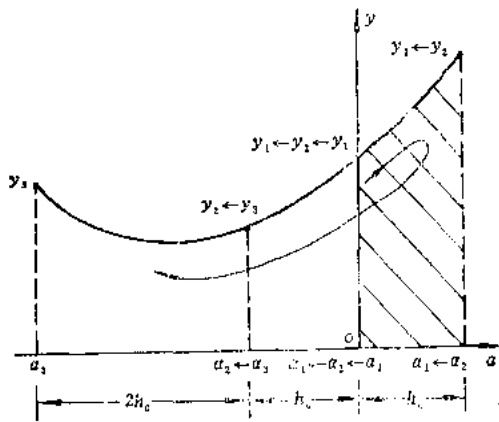


图 2-14

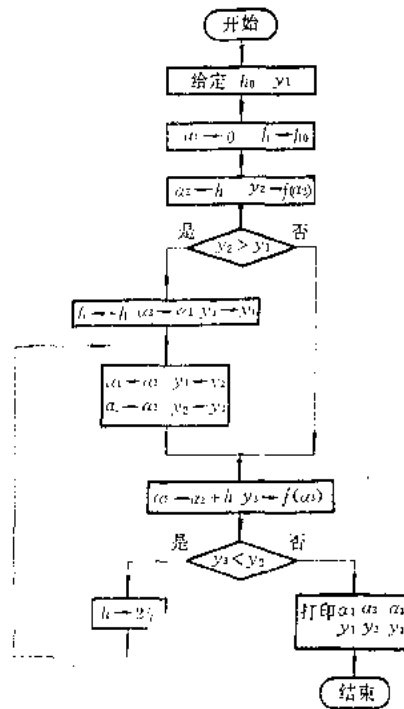


图 2-15

值 $y_1 > y_2 < y_3$, 从而形成单谷区间 $[a_3, a_1]$ 为一维搜索区间。

上述确定搜索区间的外推法, 其程序框图如图 (2-15) 所示。

二、区间消去法原理

搜索区间 $[a, b]$ 确定之后, 采用区间消去法逐步缩短搜索区间, 从而找到极小点的数值近似解。假定在搜索区间 $[a, b]$ 内任取两点 $a_1, b_1, a_1 < b_1$, 并计算函数值 $f(a_1), f(b_1)$ 。于是将有下列三种可能情形:

1. $f(a_1) < f(b_1)$, 如图 (2-16) 所示。由于函数为单谷, 所以极小点必在区间 $[a, b_1]$ 内。

2. $f(a_1) > f(b_1)$, 如图 (2—17) 所示。同理, 极小点应在区间 $[a_1, b_1]$ 内。

3. $f(a_1) = f(b_1)$, 如图 (2—18) 所示, 这时极小点应在 $[a_1, b_1]$ 内。

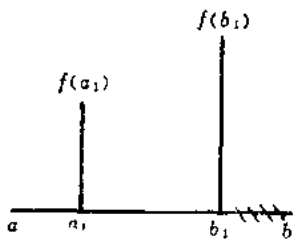


图 2—16

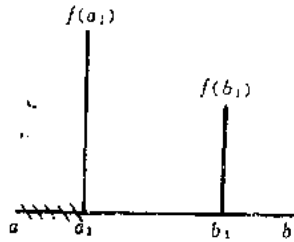


图 2—17

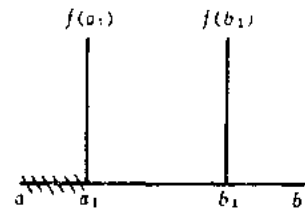


图 2—18

根据以上所述, 只要在区间 $[a, b]$ 内取两个点, 算出它们的函数值并加以比较, 就可以把搜索区间 $[a, b]$ 缩短成 $[a, b_1]$, $[a_1, b]$ 或 $[a_1, b_1]$ 。应当指出, 对于第一种情况, 我们已算出区间 $[a, b_1]$ 内 a_1 点的函数值, 如果要把搜索区间 $[a, b_1]$ 进一步缩短, 只需在其内再取一点算出函数值并与 $f(a_1)$ 加以比较, 即可达到目的。对于第二种情况, 同样只需再计算一点函数值就可以把搜索区间继续缩短。第三种情形与前面两种情形不同, 因为在区间 $[a_1, b_1]$ 内缺少已算出的函数值。要想把区间 $[a_1, b_1]$ 进一步缩短, 需在其内部取两个点 (而不是一个点) 计算出相应的函数值再加以比较才行。如果经常发生这种情形, 为了缩短搜索区间, 需要多计算一倍数量的函数值, 这就增加了计算工作量。因此, 为了避免多计算函数值, 我们把第三种情形合并到前面两种情形中去。例如, 可以把前面三种情形改为下列两种情形:

1. 若 $f(a_1) < f(b_1)$, 则取 $[a, b_1]$ 为缩短后的搜索区间。
2. 若 $f(a_1) \geq f(b_1)$, 取 $[a_1, b]$ 为缩短后的搜索区间。

因此, 为了每次缩短区间, 只需要在区间内再插入一点并算出函数值。对于插入点的位置, 可以用不同的方法来确定, 从而形成不同的一维搜索方法。概括起来, 可将一维搜索方法分成两大类。一类称作试探法, 这类方法是按某种给定的规律来确定区间内插入点的位置。此点位置的确定仅仅按照区间缩短如何加快, 而不顾及函数值的分布关系。属于试探法一维搜索的有黄金分割法, 斐波纳契 (Fibonacci) 法等。另一类一维搜索方法称作插值法或函数逼近法, 这类方法是根据某些点处的某些信息, 如函数值, 一阶导数, 二阶导数等, 构造一个插值函数来逼近原来函数, 用插值函数的极小点作为区间的插入点。属于插值法一维搜索的有二次插值法, 三次插值法等。下面我们分别讨论这两类一维搜索方法。

§ 7 一维搜索的试探方法

一、斐波纳契 (Fibonacci) 法

根据区间消去法原理, 要缩短区间就得计算函数值, 计算函数值的次数越多, 区间也就缩短得越小。可见, 区间的缩短倍数 (即原区间长度与缩短后的区间长度之比) 与计算函数值的次数有关。在计算函数值次数一定的情况下, 最初搜索区间与最终搜索区间长度之比, 可以作为衡量取点方式优劣的一个标准——这个比值越大, 意味着相应的取点方式越好。或者换一个说法: 设 L_n 是这样一个区间的长度——按某种取点方式求

n 次函数值后,把搜索区间的长度缩短为 $\frac{1}{2}$,则最优的取点方式是在可能遇到的各种情况下,应该保证使 L_n 最大。

为导出最优的取点方式,先估计一下 L_n 的上界。设 L_i 的上界为 F_i ($i=1, 2, \dots, n$),显然 F_i 就是计算 i 次函数值后把搜索区间缩短到 $\frac{1}{2}$ 的最大区间长度。由于至少要计算两次函数值才能缩短区间,所以

$$F_0 = F_1 = 1 \quad (2-19)$$

现在估计对应于计算 n 次函数值的上界 F_n 。设最初的两个试探点为 a_1 和 a_2 ($a_1 < a_2$),那么剩下还可以计算 $n-2$ 次函数值,见图(2-19)。极小点可能位于 $[a, a_1]$ 内,也可能位于区间 $[a_1, b]$ 内。当极小点位于 $[a, a_1]$ 内时,

我们必须能够借助于在其中计算 $n-2$ 次函数值而把这个区间缩短到 $\frac{1}{2}$,故应有

$$a_1 - a \leq F_{n-2}$$



图 2-19

当极小点位于 $[a_1, b]$ 内时,除了可再计算 $n-2$ 次函数值外,还可以利用其中已计算的一点 a_2 处的函数值,所以总共可以计算 $(n-2)+1=n-1$ 个函数值,故应有

$$b - a_1 \leq F_{n-1}$$

于是得知

$$L_n = b - a \leq F_{n-2} + F_{n-1}$$

因为 F_n 为 L_n 的上界,故有

$$F_n = F_{n-1} + F_{n-2} \quad (2-20)$$

因此可以按递推关系

$$\begin{aligned} F_0 &= F_1 = 1 \\ F_n &= F_{n-1} + F_{n-2} \quad (n=2, 3, \dots) \end{aligned}$$

构造下面的数列,此数列称作斐波纳契数,如

表 2-1

n	0	1	2	3	4	5	6	7	8	9	10
F_n	1	1	2	3	5	8	13	21	34	55	89
n	11	12	13	14	15	16	17	18	19	20	
F_n	144	233	377	610	987	1597	2584	4781	8765	10946	

表(2-1)所示。由于 F_n 为 L_n 的上界 $L_n \leq F_n$,因此倘若某种取点方式能保证在求 n 次函数值后,可把搜索区间缩短为最初区间长度的 $\frac{1}{F_n}$,那么就有理由认为这种取点方式是最优的。

例如取 $n=20$,则 $F_{20}=10946$,最优的取点方式可以通过计算20次函数值就能把原来长度 L 缩短为

$$\frac{L}{F_{20}} = \frac{L}{10946} \approx 0.00009L$$

所以计算函数值的次数 n 可以根据寻求极小点的精度要求来确定。如欲使最终区间长度

不超过原长度的 $\varepsilon = \frac{1}{100}$ ，只需取满足 $F_n \geq \frac{1}{\varepsilon} = 100$ 所对应的 $n=11$ ， $F_{11}=144$ 即可。也就是说，需要计算11次函数值就可以将区间缩短成原来的 $\frac{1}{100}$ 之内。若取 $\varepsilon = \frac{1}{10000}$ ，则 $F_n \geq \frac{1}{\varepsilon} = 10000$ 。当取 $n=20$ 时， $F_{20}=10946$ ，所以计算20次函数值即可将区间缩短成原来的 $\frac{1}{10000}$ 之内。

从上面的例子可以看出，当我们计算 n 次函数值时，原区间将缩短为 $\frac{L}{F_n}$ ，而若计算 $n-1$ 次时，则原区间就缩短为 $\frac{L}{F_{n-1}}$ 。因此，从计算 $n-1$ 个点到计算 n 个点之间，区间缩短率就是 $\frac{F_{n-1}}{F_n}$ ，这就是相邻两次计算之间的缩短率。

当确定要对函数值计算 n 次时，则原区间长度 L 将被分成 F_n 个单位长度的窄小搜索区间。这样在原区间 L 长度（试验范围）内，所有可能的试验点总数将是 $F_n - 1$ ，如图（2-20）所示。根据相邻两次计算之间缩短率是 $\frac{F_{n-1}}{F_n}$ 的定义，第一个试验点将放在 $L' =$

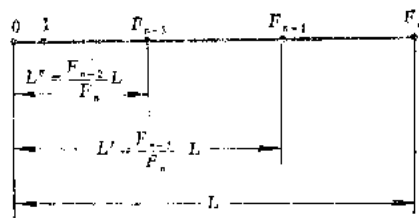


图 2-20

$\frac{F_{n-1}}{F_n}L$ 处或试验范围的 $\frac{F_{n-1}}{F_n}$ 位置上。接着在 L' 长度基础上，再进一步缩短搜索区间，那么下一个试验点将放在

$$L'' = \frac{F_{n-2}}{F_{n-1}}L' = \frac{F_{n-2}}{F_{n-1}} \cdot \frac{F_{n-1}}{F_n} = \frac{F_{n-2}}{F_n}L$$

长度处或试验范围的 $\frac{F_{n-2}}{F_n}$ 位置上。由 $F_n = F_{n-1} + F_{n-2}$ 可以推得 $L' = L - L''$ ，即两个试验点处在本次搜索区间的对称位置上。例如若 $n=11$ ，有 $F_{11}=144$ ， $F_{10}=89$ ， $F_9=55$ ，因此第一个试验点应放在试验范围的 $\frac{89}{144}$ 位置上，而第二个试验点应放在试验范围的 $\frac{55}{144}$ 位置上。

从某种意义上讲，斐波纳契法是缩短搜索区间最优的方法，但是在实际计算时，要直接用到斐波纳契数，这就增加了计算上的困难。因此，在实际计算中常用到另外一种一维搜索的试探方法——黄金分割法。

二、黄金分割法

黄金分割法适用于 $[a, b]$ 区间上的任何单谷函数求极小值问题。对函数除要求“单谷”外不作其它要求，甚至可以不连续。因此，这种方法的适应面相当广。黄金分割法也是建立在区间消去法原理基础上的试探方法，即在搜索区间 $[a, b]$ 内适当插入两点

a_1, a_2 , 并计算其函数值。 a_1, a_2 将区间分成三段, 应用函数的单谷性质, 通过函数值大小的比较, 删去其中一段, 使搜索区间得以缩短。然后再在保留下来的区间上作同样的处置, 如此迭代下去, 使搜索区间无限缩小, 从而得到极小点的数值近似解。

黄金分割法对插入点 a_1, a_2 的位置同样要求它们相对区间 $[a, b]$ 两端点具有对称性, 即

$$\begin{aligned} a_1 &= b - \lambda(b - a) \\ a_2 &= a + \lambda(b - a) \end{aligned} \quad (2-21)$$

其中 λ 为待定常数。

除对称要求外, 黄金分割法还要求在保留下来的区间内再插入一点所形成的区间新三段, 与原来区间的三段具有相同的比例分布。设原区间 $[a, b]$ 长度为 1 如图(2-21)所示, 保留下来的区间 $[a, a_2]$ 长度为 λ , 区间缩短率为 λ 。为了保持相同的比例分布, 新插入点 a_3 应在 $\lambda(1-\lambda)$ 位置上, a_1 在原区间的 $1-\lambda$ 位置应相当于在保留区间的 λ^2 位置。

故有

$$\begin{aligned} 1 - \lambda &= \lambda^2 \\ \lambda^2 + \lambda - 1 &= 0 \end{aligned}$$

取方程正数解, 得

$$\lambda = \frac{\sqrt{5} - 1}{2} \approx 0.618$$

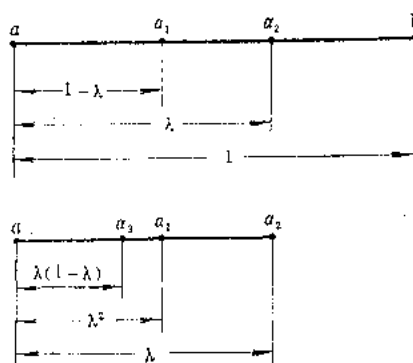


图 2-21

若保留下来的区间为 $[a_1, b]$, 根据插入点的对称性, 也能推得同样的 λ 值。所谓“黄金分割”是指将一线段分成两段的方法, 使整段长与较长段的长度比值等于较长段与较短段长度的比值, 即

$$1 : \lambda = \lambda : (1 - \lambda)$$

同样算得 $\lambda \approx 0.618$ 。可见黄金分割法能使相邻两次搜索区间都具有相同的缩短率 0.618, 所以黄金分割法又被称作 0.618 法。

在斐波纳契法中, 相邻两次搜索区间的缩短率是不相同的, 如用 n 个试验点来缩短给定的区间, 区间长度第一次的缩短率为 $\frac{F_{n-1}}{F_n}$, 其后各次分别为

$$\frac{F_{n-2}}{F_{n-1}}, \quad \frac{F_{n-3}}{F_{n-2}}, \quad \dots, \quad \frac{F_2}{F_1}$$

可以证明, 当 $n \rightarrow \infty$ 时, $\frac{F_{n-1}}{F_n} \rightarrow 0.618$

所以黄金分割法可以说是以不变的区间缩短率 0.618 代替斐波纳契法每次不同的缩短率的试探方法。

计算 n 个试验点, 用斐波纳契法可将原区间 L 缩短到 $\frac{L}{F_n}$, 用黄金分割法可将原区间连续缩短 $n-1$ 次, 最后缩短到 $\lambda^{n-1}L$ 。当 $n \rightarrow \infty$ 时二者之比为

$$\frac{\lambda^{n-1}}{1} \approx 1.17$$

这就是说，黄金分割法缩短后的区间长度为斐波纳契法的1.17倍，由此可见其效果是好的。当 n 取为2至11时，两种方法缩短后区间长度的比较列入表(2—2)中。

表 2—2

n	2	3	4	5	6	7	8	9	10	11
$\frac{1}{F_n}$	0.5	0.333	0.20	0.125	0.078	0.048	0.029	0.018	0.01	0.0069
λ^{n-1}	0.618	0.382	0.23	0.146	0.083	0.054	0.034	0.021	0.013	0.0081

黄金分割法的搜索过程是：

1. 给出初始搜索区间 $[a, b]$ 及收敛精度 ε ，将 λ 赋以 0.618。
 2. 按坐标点计算公式(2—21)计算 a_1 和 a_2 ，并计算其对应的函数值 $f(a_1)$ ， $f(a_2)$ 。
 3. 根据区间消去法原理缩短搜索区间。为了能用原来的坐标点计算公式，需进行区间名称的代换，并在保留区间中计算一个新的试验点及其函数值。
 4. 检查区间是否缩短到足够小和函数值收敛到足够近，如果条件不满足则返回到步骤 2。
 5. 如果条件满足，则取最后二试验点的平均值作为极小点的数值近似解。
- 黄金分割法的程序框图如图(2—22)所示。

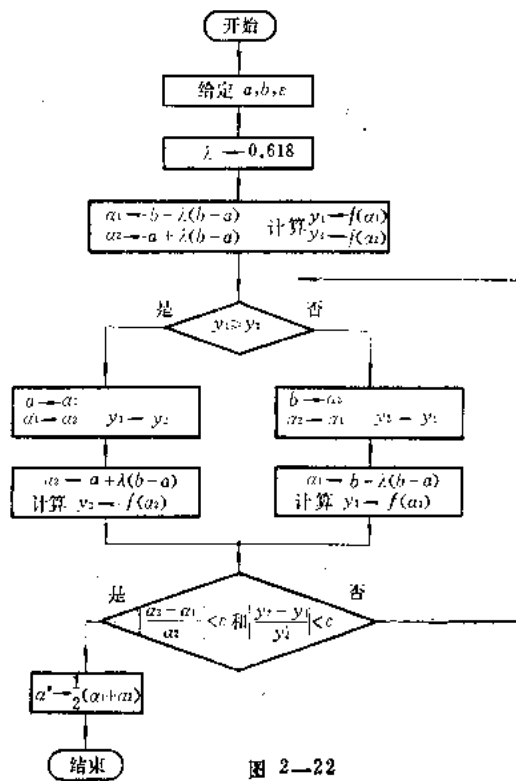


图 2—22

§ 8 一维搜索的插值方法

假定我们的问题是在某一确定区间内寻求函数的极小点位置，虽然没有函数表达式，但能够给出若干试验点处的函数值。我们可以根据这些点处的函数值，利用插值方法建立函数的某种近似表达式，进而求出函数的极小点，并用它作为原来函数极小点的近似值。这种方法称作插值方法，又称作函数逼近法。

插值方法和试探方法都是利用区间消去法原理将初始搜索区间不断缩短，从而求得极小点的数值近似解。二者不同之处在于试验点位置的确定方法不同。在试探法中试验点位置是由某种给定的规律确定的，它不考虑函数值的分布。例如，斐波纳契法试验点

位置是按斐波纳契数确定的。黄金分割法是按等比例 0.618 缩短率确定的。而在插值法中，试验点位置是按函数值近似分布的极小点确定的。所以，试探法仅仅利用了试验点函数值大小的比较，而插值法还要利用函数值本身或者其导数信息。由于试探法仅对试验点函数值的大小进行比较，而函数值本身没有得到充分利用，这样即使对一些简单的函数，例如二次函数，也不得不象一般函数那样进行同样多的函数值计算。插值法则是利用函数在已知试验点的值（或导数值）来确定新试验点的位置。当函数具有比较好的解析性质时（例如连续可微性），插值方法比试探方法效果更好。

多项式是函数逼近的一种常用工具。在搜索区间内我们可以利用若干试验点处的函数值来构造低次多项式，用它作为函数的近似表达式，并用这个多项式的极小点作为原函数极小点的近似。常用的插值多项式为二次或三次多项式。一般说来，三次插值公式的收敛性好一些，但在导数不便计算时，三点二次插值是常用的一维搜索方法。这里只介绍二次插值方法。

二次插值方法又称作抛物线法。它是利用 $y=f(a)$ 在单谷区间中的三点 $a_1 < a_2 < a_3$ 的相应函数值 $f(a_1) > f(a_2) < f(a_3)$ ，作出如下的二次插值多项式

$$P(a) = a_0 + a_1 a + a_2 a^2 \quad (2-22)$$

它应满足条件

$$P(a_1) = a_0 + a_1 a_1 + a_2 a_1^2 = y_1 = f(a_1)$$

$$P(a_2) = a_0 + a_1 a_2 + a_2 a_2^2 = y_2 = f(a_2)$$

$$P(a_3) = a_0 + a_1 a_3 + a_2 a_3^2 = y_3 = f(a_3)$$

多项式 $P(a)$ 的极值点可从极值的必要条件求得

$$P'(a_p) = a_1 + 2a_2 a_p = 0$$

$$a_p = -a_1 / 2a_2$$

为了确定这个极值点，只需计算出系数 a_1 和 a_2 。其算法是利用 a_0, a_1, a_2 的联立方程组中相邻两个方程消去 a_0 ，从而得到对于 a_1, a_2 的方程组：

$$a_1(a_1 - a_2) + a_2(a_1^2 - a_2^2) = y_1 - y_2$$

$$a_1(a_2 - a_3) + a_2(a_2^2 - a_3^2) = y_2 - y_3$$

解得

$$a_1 = \frac{(a_2^2 - a_3^2)y_1 + (a_3^2 - a_1^2)y_2 + (a_1^2 - a_2^2)y_3}{(a_1 - a_2)(a_2 - a_3)(a_3 - a_1)}$$

$$a_2 = -\frac{(a_2 - a_3)y_1 + (a_3 - a_1)y_2 + (a_1 - a_2)y_3}{(a_1 - a_2)(a_2 - a_3)(a_3 - a_1)}$$

所以

$$a_p = -\frac{a_1}{2a_2} = \frac{1}{2} \frac{(a_2^2 - a_3^2)y_1 + (a_3^2 - a_1^2)y_2 + (a_1^2 - a_2^2)y_3}{(a_2 - a_3)y_1 + (a_3 - a_1)y_2 + (a_1 - a_2)y_3}$$

如果令

$$c_1 = \frac{y_3 - y_1}{a_3 - a_1}$$

成

$$c_2 = \frac{\frac{y_2 - y_1}{a_2 - a_1} - c_1}{a_2 - a_3}$$

$$\text{则 } a_p = \frac{1}{2} \left(a_1 + a_3 - \frac{c_1}{c_2} \right) \quad (2-23)$$

这样就得到了 $f(a)$ 极小点 a^* 的近似解 a_p , 如图 (2-23) 所示。如果区间长度 $|a_3 - a_1|$ 足够小, 则由 $|a_p - a^*| < |a_3 - a_1|$ 便得出我们所要求的近似极小点 $a^* \doteq a_p$ 。如果不满足上述要求, 则必须缩小区间 $[a_1, a_3]$ 。根据区间消去法原理, 需要区间内已知两点的函数值。其中点 a_2 的函数值 $y_2 = f(a_2)$ 已知, 另外一点可取 a_p 点并计算其函数值 $y_p = f(a_p)$ 。当 $y_2 < y_p$ 时取 $[a_1, a_p]$ 为缩短后的搜索区间见图 (2-23a) (若 $y_2 \geq y_p$ 时则取 $[a_2, a_3]$ 为缩短后的搜索区间), 在新的区间内再用二次插值法插入新的极小点近似值 \tilde{a}_p , 见图 (2-23b)。如此不断进行下去, 直到满足精度要求为止。为了在每次计算插入点的坐标时能应用

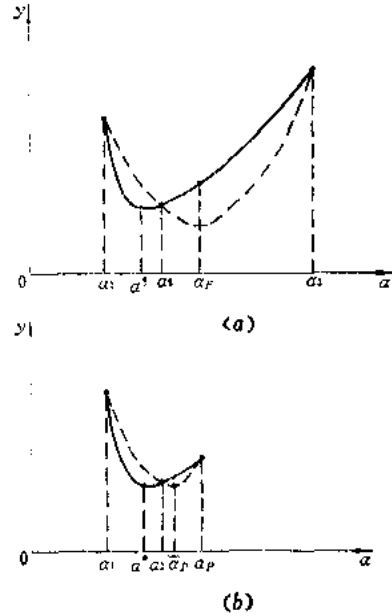


图 2-23

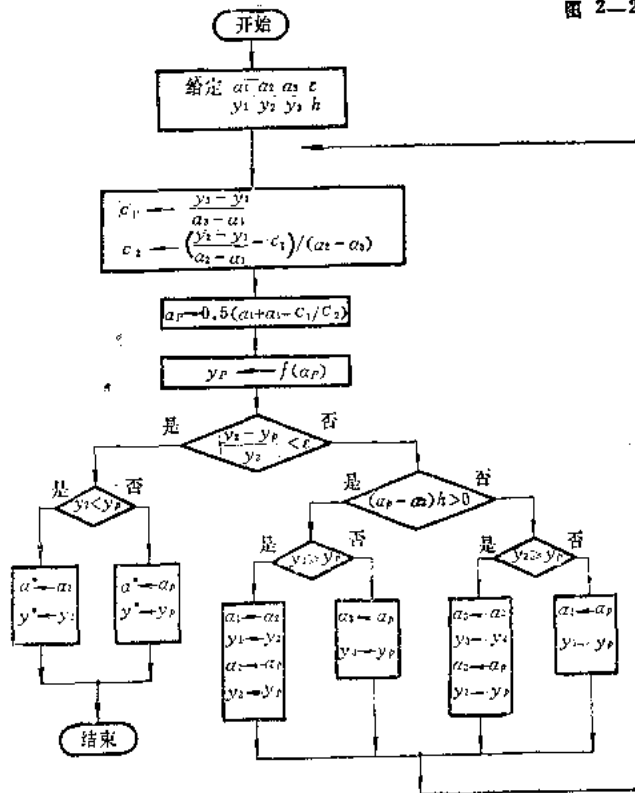


图 2-24

同一计算公式，新区间端点的坐标及函数值名称需换成原区间端点的坐标及函数值名称，即在每个新区间上仍取 $a_1 < a_2 < a_3$ 三点及其函数值 $y_1 > y_2 < y_3$ 。这样当计算插入点（即抛物线极小点）位置时可以仍用原来的计算公式。

二次插值法的计算程序框图如图（2—24）所示。

在应用上述二次插值法进行一维搜索之前，需使用一维搜索的外推法确定初始搜索区间。在此区间上函数值形成“高一低一高”的单谷形态。图（2—24）程序框图中的“ h ”就是在进行外推法求初始搜索区间过程中所形成的最后步长， h 可正可负，分别表示沿 a 正向或反向进行一维搜索。

第三章 无约束优化问题的解法

第一章第一节所列举的机械结构优化设计问题，都是在一定的限制条件下追求某一指标为最小，所以它们都属于约束优化问题。但是，也有些实际问题，其数学模型本身就是一个无约束优化问题，或者除了在非常接近最终极小点的情况下，都可以按无约束问题来处理。研究无约束优化问题的另一个原因是，通过熟悉它的解法可以为研究约束优化问题打下良好的基础。第三个原因是，约束优化问题的求解可以通过一系列无约束优化方法来达到。所以无约束优化问题的解法是优化设计方法的基本组成部分，也是优化方法的基础。

无约束优化问题是：求 n 维设计变量

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^T$$

使目标函数 $f(\mathbf{x}) \rightarrow \min$ ，而对 \mathbf{x} 没有任何限制条件。

对于无约束优化问题的求解，可以直接应用第二章讲述的极值条件来确定极值点位置。这就是把求函数极值的问题变成求解方程

$$\nabla f = 0$$

的问题。即求 \mathbf{x} ，使其满足

$$\begin{aligned} \frac{\partial f}{\partial x_1} &= 0 \\ \frac{\partial f}{\partial x_2} &= 0 \\ &\vdots \\ \frac{\partial f}{\partial x_n} &= 0 \end{aligned} \quad (3-1)$$

这是一个含有 n 个未知量， n 个方程的方程组，并且一般是非线性的。除了一些特殊情况外，一般说来非线性方程组的求解与求无约束极值一样也是一个困难问题，甚至前者比后者更困难。对于非线性方程组，一般是很难用解析方法求解的，需要采用数值计算方法逐步求出非线性联立方程组的解。但是，与其用数值计算方法求解非线性方程组，倒不如用数值计算方法直接求解无约束极值问题。因此，本章将介绍求解无约束优化问题常用的数值解法。

数值计算方法最常用的是搜寻方法，其基本思想是从给定的初始点 \mathbf{x}^0 出发，沿某一搜寻方向 \mathbf{d}^0 进行搜索，确定最佳步长 α ，使函数值沿方向 \mathbf{d}^0 下降最大，依此方式按下述公式不断进行，形成迭代的下降算法：

$$\mathbf{x}^{K+1} = \mathbf{x}^K + \alpha_K \mathbf{d}^K \quad (K = 0, 1, 2, \dots) \quad (3-2)$$

各种无约束优化方法的区别就在于确定其搜索方向 \mathbf{d}^K 的方法不同。所以，搜寻方向的构成问题乃是无约束优化方法的关键。

在 $x^{K+1} = x^K + \alpha_K d^K$ 中, d^K 是第 $K+1$ 次搜索或迭代方向, 称为搜索 (搜寻) 或迭代方向, 它是根据数学原理由目标函数和约束条件的局部信息状态形成的, α_K 为步长因子, $\alpha_K d^K$ 为步距, 若 $\|d^K\|=1$ 时, 则 α_K 就是步长。确定 d^K 的方法很多, 相应地, 确定使 $f(x^K + \alpha_K d^K)$ 取极值的 $\alpha_K = \alpha^*$ 的方法也是不同的。例如, 根据一元函数求极值的分析方法, 当把 $f(x^K + \alpha_K d^K)$ 简写成 $f(x + \alpha d)$ 并取其泰勒级数展开式至二阶项时, 有

$$\begin{aligned} f(x + \alpha d) &\cong f(x) + \alpha d^T \nabla f(x) + \frac{1}{2} (\alpha d)^T G (\alpha d) \\ &= f(x) + \alpha d^T \nabla f(x) + \frac{1}{2} \alpha^2 d^T G d \end{aligned}$$

对 α 进行微分并令其等于零, 则求得 $f(x + \alpha d)$ 的极值, 即 $\alpha = \alpha^*$ 时的公式为

$$d^T \nabla f(x) + \alpha^* d^T G d = 0$$

或

$$\alpha^* = - \frac{d^T \nabla f(x)}{d^T G d}$$

d^K 和 α_K 形成和确定方法不同就派生出不同的 n 维无约束优化问题的数值解法。因此, 可对无约束优化的算法进行分类。其分类原则就是依式

$$x^{K+1} = x + \alpha_K d^K$$

中的 d^K 和相应的 α_K 的形成或确定方法而定的。

图 (3-1) 是按迭代式 (3-2) 对无约束优化问题进行极小值计算的算法粗框图。

其中一个框是形成 d 的, 另一框是确定 α 的。显然, 对不同的形成 d 和确定 α 的算法, 只要改变这两框中的内容即可。

根据构成搜寻方向所使用的信息性质的不同, 无约束优化方法可以分为两类。一类是利用目标函数的一阶或二阶导数的无约束优化方法, 如最速下降法, 共轭梯度法, 牛顿法及变尺度法等。另一类是只利用目标函数值的无约束优化方法, 如坐标轮换法, 单形替换法及鲍威尔 (Powell) 法等。本章将分别讨论上述两类无约束优化方法。

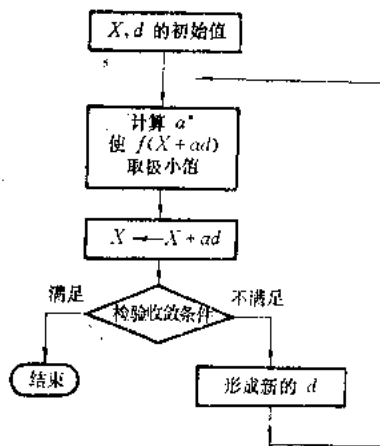


图 3-1

§ 1 最速下降法

优化设计是追求目标函数值 $f(x)$ 最小, 因此, 一个很自然的想法是从某点 x 出发, 其搜索方向 d 取该点的负梯度方向 $-\nabla f(x)$ (最速下降方向), 使函数值在该点附近的范围内下降最快。按此规律不断走步, 形成以下迭代的算法

$$\mathbf{x}^{K+1} = \mathbf{x}^K - a_K \nabla f(\mathbf{x}^K) \quad (K=0,1,2,\dots) \quad (3-3)$$

由于最速下降法是以负梯度方向作为搜索方向，所以最速下降法又称为梯度法。

为了使目标函数值沿搜索方向 $-\nabla f(\mathbf{x}^K)$ 能获得最大的下降值，其步长因子 a_K 应取一维搜索的最佳步长。即有

$$f(\mathbf{x}^{K+1}) = f[\mathbf{x}^K - a_K \nabla f(\mathbf{x}^K)] = \min_a f(\mathbf{x}^K - a \nabla f(\mathbf{x}^K)) = \min_a \varphi(a)$$

根据一元函数极值必要条件和多元复合函数求导公式，得

$$\varphi'(a) = -\{\nabla f[\mathbf{x}^K - a_K \nabla f(\mathbf{x}^K)]\}^T \nabla f(\mathbf{x}^K) = 0$$

即

$$[\nabla f(\mathbf{x}^{K+1})]^T \nabla f(\mathbf{x}^K) = 0$$

或写成

$$(\mathbf{d}^{K+1})^T \mathbf{d}^K = 0$$

由此可知，在最速下降法中，相邻两个迭代点上的函数梯度相互垂直。而搜寻方向就是负梯度方向，因此相邻两个搜寻方向互相垂直。这就是说在最速下降法中，迭代点向函数极小点靠近的过程，走的是曲折的路线。这一次的搜寻方向与前一次的搜寻方向互相垂直，形成“之”字形的锯齿现象，见图

(3-2)。从直观上可以看到，在远离极小点的位置，每次迭代可使函数值有较多的下降。可是在接近极小点的位置，由于锯齿现象使每次迭代行进的距离缩短，因而收敛速度减慢。这种情况似乎与“最速下降”的名称相矛盾，其实不然，这是因为梯度是函数的局部性质。从局部上看，在一点附近函数的下降是快的，但从整体上看则走了许多弯路，因此函数的下降并不算快。

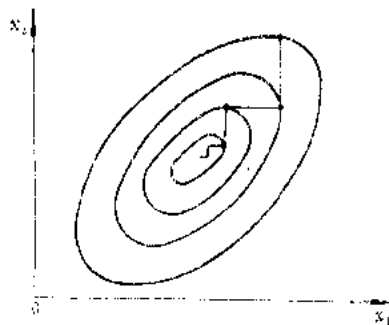


图 3-2

例：求目标函数 $f(\mathbf{x}) = x_1^2 + 25x_2^2$ 的极小点。

取初始点 $\mathbf{x}^0 = [2, 2]^T$

则初始点处函数值及梯度分别为

$$f(\mathbf{x}^0) = 104$$

$$\nabla f(\mathbf{x}^0) = \begin{Bmatrix} 2x_1 \\ 50x_2 \end{Bmatrix}_{\mathbf{x}^0} = \begin{Bmatrix} 4 \\ 100 \end{Bmatrix}$$

沿负梯度方向进行一维搜索，有

$$\mathbf{x}^1 = \mathbf{x}^0 - a_0 \nabla f(\mathbf{x}^0)$$

$$= \begin{Bmatrix} 2 \\ 2 \end{Bmatrix} - a_0 \begin{Bmatrix} 4 \\ 100 \end{Bmatrix} = \begin{Bmatrix} 2 - 4a_0 \\ 2 - 100a_0 \end{Bmatrix}$$

a_0 为一维搜索最佳步长，应满足极值必要条件

$$f(\mathbf{x}^1) = \min_a f[\mathbf{x}^0 - a \nabla f(\mathbf{x}^0)]$$

$$= \min\{(2-4a)^2 + 25(2-100a)^2\}$$

$$= \min\varphi(a)$$

$$\varphi'(a_0) = -8(2-4a_0) - 5000(2-100a_0) = 0$$

从而算出一维搜索最佳步长

$$a_0 = \frac{626}{31252} = 0.02003072$$

及第一次迭代设计点位置和函数值

$$x^1 = \begin{Bmatrix} 2-4a_0 \\ 2-100a_0 \end{Bmatrix} = \begin{Bmatrix} 1.919877 \\ -0.3071785 \times 10^{-2} \end{Bmatrix}$$

$$f(x^1) = 3.686164$$

从而完成了最速下降法的第一次迭代。继续作下去，经 10 次迭代后，得到最优解

$$x^* = [0, 0]^T$$

$$f(x^*) = 0$$

这个问题的目标函数 $f(x)$ 的等值线为一族椭圆，迭代点从 x^0 走的是一段锯齿形路线，见图 (3-3)。

若将上例的目标函数 $f(x) = x_1^2 + 25x_2^2$ 引入变换

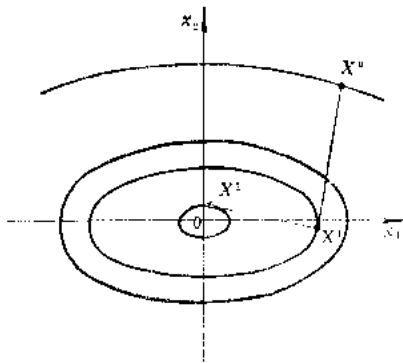


图 3-3

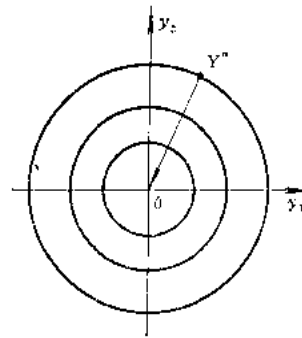


图 3-4

$$y_1 = x_1$$

$$y_2 = 5x_2$$

则函数 $f(x_1, x_2)$ 变为

$$\psi(y_1, y_2) = y_1^2 + y_2^2$$

其等值线就由一族椭圆变成一族同心圆，见图 (3-4)。仍从 $x^0 = [2, 2]^T$ 即 $y^0 = [2, 10]^T$ 出发进行最速下降法寻优。此时有

$$\psi(y^0) = 104$$

$$\nabla\psi(y^0) = \begin{Bmatrix} 2y_1 \\ 2y_2 \end{Bmatrix} \Big|_{y^0} = \begin{Bmatrix} 4 \\ 20 \end{Bmatrix}$$

沿负梯度 $-\nabla\psi(y^0)$ 方向进行一维搜索, 有

$$\begin{aligned} y^1 &= y^0 - \alpha'_0 \nabla\psi(y^0) \\ &= \left\{ \begin{array}{c} 2 \\ 10 \end{array} \right\} - \alpha'_0 \left\{ \begin{array}{c} 4 \\ 20 \end{array} \right\} = \left\{ \begin{array}{c} 2 - 4\alpha'_0 \\ 10 - 20\alpha'_0 \end{array} \right\} \end{aligned}$$

α'_0 为一维搜索最佳步长, 可由极值条件算出:

$$\begin{aligned} \psi(y^1) &= \min_{\alpha'} \psi(y^0 - \alpha' \nabla\psi(y^0)) \\ &= \min_{\alpha'} \Phi(\alpha') \\ \Phi(\alpha') &= (2 - 4\alpha')^2 + (10 - 20\alpha')^2 \\ \Phi'(\alpha'_0) &= -8(2 - 4\alpha'_0) - 40(10 - 20\alpha'_0) = 0 \\ \alpha'_0 &= \frac{26}{52} = 0.5 \end{aligned}$$

从而算得第一次走步后设计点的位置及其相应的目标函数值为

$$\begin{aligned} y^1 &= \left\{ \begin{array}{c} 2 - 4\alpha'_0 \\ 10 - 20\alpha'_0 \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ 0 \end{array} \right\} \\ \psi(y^1) &= 0 \end{aligned}$$

可见经过坐标变换后, 只需经过一次迭代, 就可找到最优解 $x^* = [0, 0]^T$, $f(x^*) = 0$ 。

比较以上两种函数形式

$$\begin{aligned} f(x_1, x_2) &= x_1^2 + 25x_2^2 = \frac{1}{2} [x_1 \ x_2] \begin{bmatrix} 2 & 0 \\ 0 & 50 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} \\ \psi(y_1, y_2) &= y_1^2 + y_2^2 = \frac{1}{2} [y_1 \ y_2] \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix} \end{aligned}$$

可以看出它们中间的对角形矩阵不同, 同时 $f(x_1, x_2)$ 的等值线为一族椭圆而 $\psi(y_1, y_2)$ 的等值线为一族同心圆。这是由于经过尺度变换

$$\begin{aligned} y_1 &= x_1 \\ y_2 &= 5x_2 \end{aligned}$$

即 x_1 轴的度量不变, 而把 x_2 轴的度量放大 5 倍, 从而把等值线由椭圆变成圆了。这说明上面两个二次型函数的对角形矩阵刻画了椭圆的长短轴, 它们是表示度量的矩阵或者是表示尺度的矩阵。最速下降法的收敛速度和变量的尺度关系很大, 这一点可从最速下降法收敛速度的估计式上看出来。在适当条件下, 有

$$\|x^{K+1} - x^*\| \leq \left(1 - \frac{m^2}{M^2}\right) \|x^K - x^*\| \quad (3-4)$$

其中 M 为 $f(x)$ 的海赛矩阵最大特征值上界, m 为其最小特征值下界。对于等值线为椭圆的二次型函数 $f(x) = x_1^2 + 25x_2^2$, 其海赛矩阵

$$G = \begin{bmatrix} 2 & 0 \\ 0 & 50 \end{bmatrix}$$

两个特征值分别为 $\lambda_1 = 2, \lambda_2 = 50$ 。因此 $m = 2, M = 50$

$$\|x^{K+1} - x^*\| \leq \left(1 - \frac{2^2}{50^2}\right) \|x^K - x^*\| = \frac{624}{625} \|x^K - x^*\|$$

可见等值线为椭圆的长短轴相差越大，收敛就越慢。而对等值线为圆的二次函数 $\psi(y_1, y_2) = y_1^2 + y_2^2$ ，其海赛矩阵

$$G = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

两个特征值相等 $\lambda_1 = \lambda_2 = 2$ ，因此 $m = M = 2$ ，代入 (3-4) 式，有

$$\|y^{K+1} - y^*\| \leq 0$$

得

$$y^{K+1} = y^*$$

即经过一次迭代便可到达极值点。

当相邻两个迭代点之间满足关系式 (3-4) 时 (右边的系数为小于等于 1 的正的常数)，我们称相应的迭代方法是具有线性收敛速度的迭代法。因此，最速下降法是具有线性收敛速度的迭代法。

最速下降法算法的程序框图如图 (3-5) 所示。

最速下降法是一个求解极值问题的古老算法，早在 1847 年就已由柯西 (Cauchy) 提出。此法直观、简单。由于它采用了函数的负梯度方向作为下一步搜索方向，所以收敛速度较慢，越是接近极值点收敛越慢，这是它的主要缺点。应用最速下降法可以使目标函数在开头几步下降很快，所以它可与其它无约束优化方法配合使用。特别是一些更有效的方法都是在对它改进后，或在它的启发下获得的，因此最速下降法仍是许多有约束和无约束优化方法的基础。

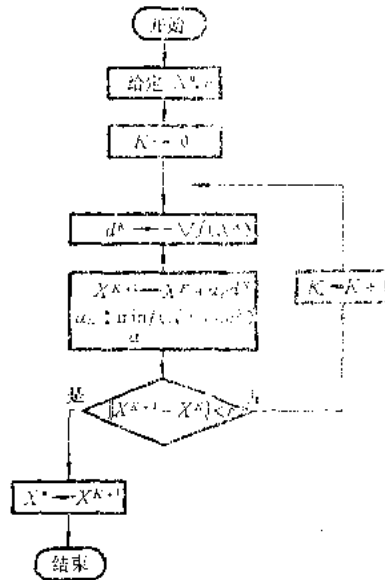


图 3-5

§ 2 牛顿 (Newton) 型方法

牛顿方法和最速下降法一样，也是求解极值问题古老的算法之一。

我们首先研究一维情况。对于一元函数 $f(x)$ ，假定已给出极小点 x^* 的一个较好的近似点 x_0 ，因为一个连续可微的函数在极小点附近与一个二次函数很接近，所以我们在 x_0 点附近用二次函数 $\varphi(x)$ 来逼近 $f(x)$ 。即在 x_0 点将 $f(x)$ 用泰勒公式展开，保留到二次项，有

$$f(x) \approx \varphi(x)$$

$$\varphi(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2}f''(x_0)(x-x_0)^2$$

然后,以二次函数 $\varphi(x)$ 的极小点 x_1 作为 $f(x)$ 极小点 x^* 的一个新的近似点。由极值的必要条件

$$\varphi'(x_1) = 0$$

即

$$\varphi'(x_1) = f'(x_0) + f''(x_0)(x_1 - x_0) = 0$$

得

$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)}$$

若在 x_1 点附近再进一步找 x^* 的近似点 x_2 , 可用 x_1 代替上式的 x_0 , 如此迭代下去, 得到一维情况下的牛顿法迭代公式:

$$x_{K+1} \approx x_K - \frac{f'(x_K)}{f''(x_K)} \quad (K=0, 1, 2, \dots) \quad (3-5)$$

从式 (3-5) 可以看出, 新的近似点是通过原近似点 x_0 处的切线

$$\varphi'(x) = f'(x_0) + f''(x_0)(x-x_0)$$

代替曲线 $f'(x)$, 用求解方程 $\varphi'(x) = 0$ 代替求解 $f'(x) = 0$ 而得到的。或者说是用切线与 x 轴的交点 x_1 来近似曲线与 x 轴的交点 x^* , 如图 (3-6) 所示。

对于多元函数 $f(x)$, 设 x^K 为 $f(x)$ 极小点 x^* 的一个近似点, 在 x^K 处将 $f(x)$ 进行泰勒展开, 保留到二次项, 得

$$f(x) \approx \varphi(x) = f(x^K) + \nabla f(x^K)^T(x-x^K) + \frac{1}{2}(x-x^K)^T \nabla^2 f(x^K)(x-x^K)$$

其中 $\nabla^2 f(x^K)$ 为 $f(x)$ 在 x^K 处的海赛矩阵。设 x^{K+1} 为 $\varphi(x)$ 的极小点, 它作为 $f(x)$ 极小点 x^* 的下一个近似点, 根据极值必要条件

$$\nabla \varphi(x^{K+1}) = 0$$

即

$$\nabla f(x^K) + \nabla^2 f(x^K)(x^{K+1} - x^K) = 0$$

得

$$x^{K+1} = x^K - [\nabla^2 f(x^K)]^{-1} \nabla f(x^K) \quad (K=0, 1, 2, \dots) \quad (3-6)$$

这就是多元函数求极值的牛顿法迭代公式。

对于二次函数, $f(x)$ 的上述泰勒展开式不是近似的, 而是精确的。海赛矩阵 $\nabla^2 f(x^K)$ 是一个常矩阵, 其中各元素均为常数。因此, 无论从任何点出发, 只需一步就可找到极小点。在第一章中已经指出, 若某一迭代方法能使二次型函数在有限次迭代内达到极小点, 则称此迭代方法是二次收敛的, 因此牛顿方法是二次收敛的。

例: 用牛顿法求 $f(x_1, x_2) = x_1^2 + 25x_2^2$ 的极小值。

取初始点 $x^0 = [2, 2]^T$

则初始点处的函数梯度、海赛矩阵及其逆阵分别是

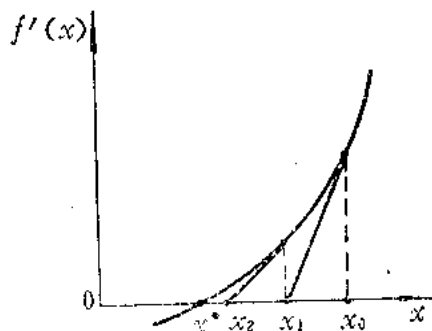


图 3-6

$$\nabla f(x^0) = \begin{Bmatrix} 2x_1 \\ 50x_2 \end{Bmatrix}, x^0 = \begin{Bmatrix} 4 \\ 100 \end{Bmatrix}$$

$$\nabla^2 f(x^0) = \begin{bmatrix} 2 & 0 \\ 0 & 50 \end{bmatrix}$$

$$[\nabla^2 f(x^0)]^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{50} \end{bmatrix}$$

代入牛顿法迭代公式，得

$$\begin{aligned} x^1 &= x^0 - [\nabla^2 f(x^0)]^{-1} \nabla f(x^0) \\ &= \begin{Bmatrix} 2 \\ 2 \end{Bmatrix} - \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{50} \end{bmatrix} \begin{Bmatrix} 4 \\ 100 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \end{aligned}$$

从而经过一次迭代即求得极小点 $x^* = [0, 0]$ 及函数极小值 $f(x^*) = 0$ 。

对于非二次型函数，如果采用上述牛顿法迭代公式，有时会使函数值上升，即出现 $f(x^{K+1}) > f(x^K)$ 。为此，需对上述牛顿法进行改进，提出所谓“阻尼牛顿法”。

如果我们把

$$d^K = -[\nabla^2 f(x^K)]^{-1} \nabla f(x^K)$$

看作是一个搜索方向，称其为牛顿方向。则阻尼牛顿法采取如下的迭代公式

$$\begin{aligned} x^{K+1} &= x^K + \alpha_K d^K \\ &= x^K - \alpha_K [\nabla^2 f(x^K)]^{-1} \nabla f(x^K) \end{aligned} \quad (3-7)$$

($K = 0, 1, 2, \dots$)

其中 α_K 为沿牛顿方向进行一维搜索的最佳步长，可称为阻尼因子。 α_K 可通过如下极小化程过求得

$$f(x^{K+1}) = f(x^K + \alpha_K d^K) = \min_{\alpha} f(x^K + \alpha d^K)$$

这样，原来的牛顿法就相当于阻尼牛顿法的步长因子 α_K 取成固定值 1 的情况。由于阻尼牛顿法每次迭代都在牛顿方向上进行一维搜索，这就避免了迭代后函数值上升的现象，从而保持了牛顿法二次收敛的特性，而对初始点的选取并没有苛刻的要求。

阻尼牛顿法的计算步骤如下：

1. 给定初始点 x^0 ，收敛精度 ε ，置 $K \leftarrow 0$
2. 计算 $\nabla f(x^K)$ ， $\nabla^2 f(x^K)$ ， $[\nabla^2 f(x^K)]^{-1}$ 和 $d^K = -[\nabla^2 f(x^K)]^{-1} \nabla f(x^K)$
3. 求 $x^{K+1} = x^K + \alpha_K d^K$ ，其中 α_K 为沿 d^K 进行一维搜索的最佳步长。
4. 检查收敛精度。若 $\|x^{K+1} - x^K\| < \varepsilon$ 则 $x^* = x^{K+1}$ ，停机；否则，置 $K \leftarrow K+1$ ，返回到 2 继续进行搜索。

阻尼牛顿法程序框图如图 (3-7) 所示。

牛顿法和阻尼牛顿法统称为牛顿型方法。这类方法的主要缺点是每次迭代都要计算函数的二阶导数矩阵，并对该矩阵求逆。这样工作量很大。特别是矩阵求逆，当维数高时工作量更大。另外，从计算机存储方面考虑，牛顿型方法所需的存储量也是很大的。最速下降法的收敛速度比牛顿法慢，而牛顿法又存在上述缺点。针对这些缺点，近年来人们研究了很多改进的算法，如针对最速下降法（梯度法）提出只用梯度信息，但比最速下降法收敛速度快的共轭梯度法；针对牛顿法提出变尺度法等。这将在下几节中予以讨论。

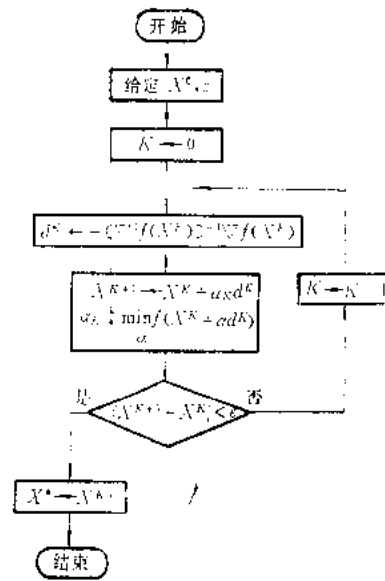


图 3-7

§ 3 共轭方向及共轭方向法

为了克服最速下降法的锯齿现象以提高其收敛速度，有一类共轭方向法是人们广为使用的。由于这类方法的搜索方向取的是共轭方向，因此先介绍共轭方向的概念和性质。

一、共轭方向的概念

共轭方向的概念是在研究二次函数

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c \quad (3-8)$$

(\mathbf{G} 为对称正定矩阵) 时引出的。本节和以后几节所介绍的方法有一个共同的特点，就是首先以 (3-8) 式的二次函数为目标函数给出有关算法，然后再把算法推广到一般的目标函数中去。

为了直观起见，首先考虑二维情况。二元二次函数的等值线为一族椭圆，任选初始点 \mathbf{x}^0 沿某个下降方向 \mathbf{p}_0 作一维搜索，得 \mathbf{x}^1

$$\mathbf{x}^1 = \mathbf{x}^0 + \alpha_0 \mathbf{p}_0$$

因为 α_0 是沿 \mathbf{p}_0 方向搜索的最佳步长，即在 \mathbf{x}^1 点处函数 $f(\mathbf{x})$ 沿 \mathbf{p}_0 方向的方向导数为零。考虑到 \mathbf{x}^1 点处方向导数与梯度之间的关系，故有

$$\left. \frac{\partial f}{\partial \mathbf{p}_0} \right|_{\mathbf{x}^1} = [\nabla f(\mathbf{x}^1)]^T \mathbf{p}_0 = 0 \quad (3-9)$$

\mathbf{p}_0 与某一等值线相切于 \mathbf{x}^1 点。下一次迭代，如果按最速下降法，选择负梯度 $-\nabla f(\mathbf{x}^1)$ 方向为搜索方向，那将发生锯齿现象。为避免锯齿的发生，我们可取下一次的迭代搜索

方向 p_1 直指极小点 x^* ，如图 (3—8) 所示。如果能够选定这样的搜索方向，那么对于二元二次函数只需顺次进行 p_0, p_1 两次直线搜索就可以求到极小点 x^* ，即有

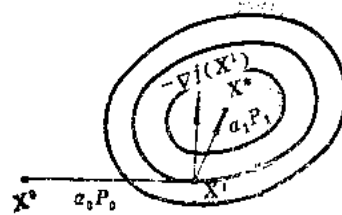


图 3—8

$$x^* = x^1 + \alpha_1 p_1$$

其中 α_1 为 p_1 方向上的最佳步长。那么这样的 p_1 方向应该满足什么条件呢？对于由 (3—8) 式所表示的二次函数 $f(x)$ ，有

$$\nabla f(x^1) = Gx^1 + b$$

当 $x^1 \neq x^*$ 时 $\alpha_1 \neq 0$ ，由于 x^* 是函数 $f(x)$ 的极小点，应满足极值必要条件，故有

$$\nabla f(x^*) = Gx^* + b = 0$$

即

$$\nabla f(x^*) = G(x^1 + \alpha_1 p_1) + b = \nabla f(x^1) + \alpha_1 Gp_1 = 0$$

将等式两边同时左乘 p_0^T ，并注意到式 (3—9) 和 $\alpha_1 \neq 0$ 的条件，得

$$p_0^T Gp_1 = 0 \quad (3-10)$$

这就是为使 p_1 直指极小点 x^* ， p_1 所必须满足的条件。满足式 (3—10) 的两个向量 p_0 和 p_1 称为 G 的共轭向量，或称 p_0 和 p_1 对 G 是共轭方向。

二、共轭方向的性质

定义：设 G 为 $n \times n$ 对称正定矩阵，若 n 维空间中有 m 个非零向量 p_0, p_1, \dots, p_{m-1} 满足

$$p_i^T Gp_j = 0 \quad (i, j = 0, 1, 2, \dots, m-1) \quad (i \neq j) \quad (3-11)$$

则称 p_0, p_1, \dots, p_{m-1} 对 G 共轭，或称它们是 G 的共轭方向。

当 $G = I$ (单位矩阵) 时，式 (3—11) 变成

$$p_i^T p_j = 0 \quad (i \neq j)$$

即向量 p_0, p_1, \dots, p_{m-1} 互相正交。由此可见，共轭概念是正交概念的推广，正交是共轭的特例。

性质 1：若非零向量系 p_0, p_1, \dots, p_{m-1} 是对 G 共轭的，则这 m 个向量是线性无关的。

性质 2：在 n 维空间中互相共轭的非零向量的向量个数不超过 n 。

性质 3：从任意初始点 x^0 出发，顺次沿 n 个 G 的共轭方向 p_0, p_1, \dots, p_{n-1} 进行一维搜索，最多经过 n 次迭代就可以找到二次函数 $f(x)$ (由式 (3—8) 所表示) 的极小点 x^* 。此性质表明这种迭代方法具有二次收敛性。

三、共轭方向法

共轭方向法是建立在共轭方向性质 3 的基础上的，它提供了求二次函数极小点的原则方法。其步骤是：

1. 选定初始点 x^0 ，下降方向 p_0 和收敛精度 ε ，置 $K \leftarrow 0$ 。
2. 沿 p_K 方向进行一维搜索，得 $x^{K+1} = x^K + \alpha_K p_K$ 。
3. 判断 $\|\nabla f(x^{K+1})\| < \varepsilon$ 是否满足，若满足则打印 x^{K+1} ，停机，否则转 4。
4. 提供新的共轭方向 p_{K+1} ，使 $p_j^T G p_{K+1} = 0$ ， $j = 0, 1, 2, \dots, K$ 。
5. 置 $K \leftarrow K+1$ ，转 2。

共轭方向法程序框图如图 (3—9) 所示。提供共轭向量系的方法有许多种，从而形成各种具体的共轭方向法，如共轭梯度法，鲍威尔 (Powell) 法等。这些方法将在下面几节中予以讨论。这里首先介绍格拉姆—斯密特 (Gram—Schmidt) 向量系共轭化方法，它是格拉姆—斯密特向量系正交化方法的推广。

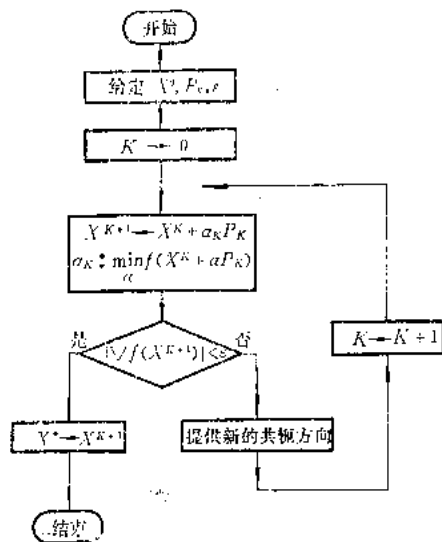


图 3—9

设已选定线性无关向量系 v_0, v_1, \dots, v_{n-1} (例如，它们是 n 个坐标轴上的单位向量)，首先

取 $p_0 = v_0$

令 $p_1 = v_1 + \beta_{10} p_0$

其中 β_{10} 是待定系数，它根据 p_1 与 p_0

共轭条件来确定，即

$$p_0^T G p_1 = p_0^T G (v_1 + \beta_{10} p_0) = 0$$

$$\beta_{10} = - \frac{p_0^T G v_1}{p_0^T G p_0}$$

从而求得与 p_0 共轭的

$$p_1 = v_1 - \frac{p_0^T G v_1}{p_0^T G p_0} p_0$$

设已求得共轭向量 p_0, p_1, \dots, p_K ，现求 p_{K+1} 。令

$$p_{K+1} = v_{K+1} + \sum_{r=0}^K \beta_{K+1,r} p_r$$

为使 p_{K+1} 与 p_j ($j = 0, 1, 2, \dots, k$) 共轭，应有

$$p_j^T G p_{K+1} = p_j^T G (v_{K+1} + \sum_{r=0}^K \beta_{K+1,r} p_r) = 0$$

由此解得

$$\beta_{K+1,j} = - \frac{p_j^T G v_{K+1}}{p_j^T G p_j} \quad (3-12)$$

于是

$$p_{K+1} = v_{K+1} - \sum_{j=0}^K \frac{p_j^T G v_{K+1}}{p_j^T G p_j} p_j \quad (3-13)$$

例：求

$$\mathbf{G} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

的一组共轭向量系 $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$

选三个坐标轴上的单位向量 $\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2$ 作为一组线性无关向量系:

$$\mathbf{e}_0 = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \quad \mathbf{e}_1 = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} \quad \mathbf{e}_2 = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$$

取 $\mathbf{p}_0 = \mathbf{e}_0 = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$

设 $\mathbf{p}_1 = \mathbf{e}_1 + \beta_{10}\mathbf{p}_0$

$$\beta_{10} = -\frac{\mathbf{p}_0^T \mathbf{G} \mathbf{e}_1}{\mathbf{p}_0^T \mathbf{G} \mathbf{p}_0} = -\frac{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}}{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}} = \frac{1}{2}$$

$$\mathbf{p}_1 = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} + \frac{1}{2} \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} \frac{1}{2} \\ 1 \\ 0 \end{Bmatrix}$$

设 $\mathbf{p}_2 = \mathbf{e}_2 + \beta_{21}\mathbf{p}_1 + \beta_{20}\mathbf{p}_0$

$$\beta_{21} = -\frac{\mathbf{p}_1^T \mathbf{G} \mathbf{e}_2}{\mathbf{p}_1^T \mathbf{G} \mathbf{p}_1} = -\frac{\begin{bmatrix} \frac{1}{2} & 1 & 0 \end{bmatrix} \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}}{\begin{bmatrix} \frac{1}{2} & 1 & 0 \end{bmatrix} \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{Bmatrix} \frac{1}{2} \\ 1 \\ 0 \end{Bmatrix}} = \frac{2}{3}$$

$$\beta_{20} = -\frac{\mathbf{p}_0^T \mathbf{G} \mathbf{e}_2}{\mathbf{p}_0^T \mathbf{G} \mathbf{p}_0} = -\frac{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}}{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}} = 0$$

得

$$p_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \frac{2}{3} \begin{pmatrix} \frac{1}{2} \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ \frac{2}{3} \\ 1 \end{pmatrix}$$

计算表明

$$p_i^T G p_j = \begin{cases} \neq 0 & (i=j) \\ = 0 & (i \neq j) \end{cases} \quad (i, j = 0, 1, 2)$$

说明 p_0, p_1, p_2 对 G 共轭。

上述算法是针对二次函数的，但也可以用于一般非二次函数。非二次函数在极小点附近可用二次函数来近似

$$f(x) \approx f(x^*) + \frac{1}{2}(x - x^*)^T G(x^*)(x - x^*)$$

上式中的海赛矩阵 $G(x^*)$ 相当于二次函数中的矩阵 G ，但 x^* 未知。当迭代点 x^0 充分靠近 x^* 时，可用 $G(x^0)$ 构造共轭向量系。更有效的共轭方法是构造共轭向量系时避开海赛矩阵，这将在下节中予以讨论。

§ 4 共轭梯度法

共轭梯度法是共轭方向法中的一种，因为在该方法中每一个共轭向量都是依赖于迭代点处的负梯度而构造出来的，所以称作共轭梯度法。为了利用梯度求共轭方向，我们首先来研究共轭方向与梯度之间的关系。

考虑二次函数

$$f(x) = \frac{1}{2}x^T G x + b^T x + c$$

从 x^K 点出发，沿 G 的某一共轭方向 p_K 作一维搜索，到达 x^{K+1} 点，即

$$x^{K+1} = x^K + \alpha_K p_K$$

或

$$x^{K+1} - x^K = \alpha_K p_K$$

而在 x^K, x^{K+1} 点处的梯度 g_K, g_{K+1} 分别为

$$g_K = G x^K + b$$

$$g_{K+1} = G x^{K+1} + b$$

所以有

$$g_{K+1} - g_K = G(x^{K+1} - x^K) = \alpha_K G p_K \quad (3-14)$$

若 p_i 和 p_K 对 G 是共轭的，则有

$$p_i^T G p_K = 0$$

利用式 (3-14) 对两端前乘 p_i^T ，即得

$$p_i^T (g_{K+1} - g_K) = 0 \quad (3-15)$$

这就是共轭方向与梯度之间的关系。此式表明沿方向 p_K 进行一维搜索，其终点 x^{K+1} 与始点 x^K 的梯度之差 $g_{K+1} - g_K$ 与 p_K 的共轭方向 p_i 正交。共轭梯度法就是利用这个性质做到不必计算矩阵 G 就能求得共轭方向的。

共轭梯度法的计算过程如下：

1. 设初始点 x^0 ，第一个搜索方向取 x^0 点的负梯度 $-g_0$ ，即

$$p_0 = -g_0 \quad (3-16)$$

沿 p_0 进行一维搜索，得 $x^1 = x^0 + \alpha_0 p_0$ ，并算出 x^1 点处的梯度 g_1 。 x^1 是以 p_0 为切线和某等值曲线的切点。根据梯度和该点等值面的切面相垂直的性质，因此 g_1 和 p_0 正交，有 $p_0^T g_1 = 0$ ，从而 g_1 和 g_0 正交，即 $g_1^T g_0 = 0$ ， p_0 和 g_1 组成平面正交系。

2. 在 p_0, g_1 所构成的平面正交系中求 p_0 的共轭方向 p_1 ，作为下一步的搜索方向。把 p_1 取成 $-g_1$ 与 p_0 两个方向的线性组合，即

$$p_1 = -g_1 + \beta_0 p_0 \quad (3-17)$$

β_0 为待定常数，它可以根据共轭方向与梯度的关系求得。

由
$$p_1^T (g_1 - g_0) = 0$$

有
$$(-g_1 + \beta_0 p_0)^T (g_1 - g_0) = 0$$

将此式展开，考虑到 $g_1^T p_0 = 0$ ， $g_1^T g_0 = 0$ ，可求得

$$\beta_0 = \frac{g_1^T g_1}{g_0^T g_0} = \frac{\|g_1\|^2}{\|g_0\|^2} \quad (3-18)$$

$$p_1 = -g_1 + \frac{\|g_1\|^2}{\|g_0\|^2} p_0$$

沿 p_1 方向进行一维搜索，得 $x^2 = x^1 + \alpha_1 p_1$ ，并算出该点梯度 g_2 ，有 $p_1^T g_2 = 0$ ，即

$$(-g_1 + \beta_0 p_0)^T g_2 = 0 \quad (3-19)$$

因为 p_0 和 p_1 共轭，根据共轭方向与梯度的关系式 (3-15) 有

$$p_0^T (g_2 - g_1) = 0$$

考虑到 $p_0^T g_1 = 0$ ，因此 $p_0^T g_2 = 0$ ，即 g_2 和 g_0 正交。又根据 (3-19) 式得 $g_1^T g_2 = 0$ ，即 g_2 又和 g_1 正交。由此可知 g_0, g_1, g_2 构成一个正交系。

3. 在 g_0, g_1, g_2 所构成的正交系中，求与 p_0 及 p_1 均共轭的方向 p_2 。

设
$$p_2 = -g_2 + \gamma_1 g_1 + \gamma_0 g_0$$

其中 γ_1, γ_0 为待定系数。因为要求 p_2 与 p_0 和 p_1 均共轭，根据 (3-15) 式共轭方向与梯度的关系，有

$$(-g_2 + \gamma_1 g_1 + \gamma_0 g_0)^T (g_1 - g_0) = 0$$

$$(-g_2 + \gamma_1 g_1 + \gamma_0 g_0)^T (g_2 - g_1) = 0$$

考虑到 g_0, g_1, g_2 相互正交，从而有

$$\gamma_1 g_1^T g_1 - \gamma_0 g_0^T g_0 = 0$$

$$-g_2^T g_2 - \gamma_1 g_1^T g_1 = 0$$

设 $\beta_1 = -\gamma_1$ ，得

$$\beta_1 = -\gamma_1 = \frac{g_2^T g_2}{g_1^T g_1} = \frac{\|g_2\|^2}{\|g_1\|^2}$$

$$\gamma_0 = \gamma_1 \frac{g_1^T g_1}{g_0^T g_0} = -\beta_1 \beta_0$$

因此

$$\begin{aligned} \hat{p}_2 &= -g_2 + \beta_1 g_1 + \beta_0 g_0 \\ &= -g_2 - \beta_1 g_1 - \beta_0 g_0 \\ &= -g_2 + \beta_1 (-g_1 + \beta_0 p_0) \\ &= -g_2 + \beta_1 p_1 \end{aligned}$$

从而得出

$$p_2 = -g_2 + \frac{\|g_2\|^2}{\|g_1\|^2} p_1$$

再沿 p_2 方向继续进行一维搜索，如此继续下去可求得共轭方向的递推公式为

$$p_{K+1} = -g_{K+1} + \frac{\|g_{K+1}\|^2}{\|g_K\|^2} p_K \quad (3-20)$$

$$(K = 0, 1, 2, \dots, n-1)$$

沿着这些共轭方向一直搜索下去，直到最后迭代点处梯度的模小于给定允许值为止。若目标函数为非二次函数，经 n 次搜索还未达到最优点时，则以最后得到的点作为初始点，重新计算共轭方向，一直到满足精度要求为止。

共轭梯度法的程序框图如图 (3-10) 所示。

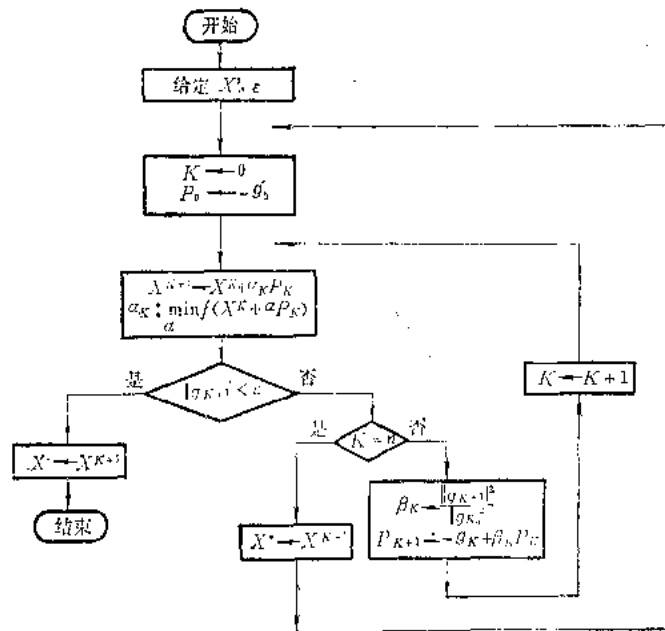


图 3-10

例：用共轭梯度法求二次函数

$$f(x_1, x_2) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2$$

的极小点及极小值。

取初始点

$$x^0 = [1, 1]^T$$

则

$$\bar{\mathbf{g}}_0 = \nabla f(\mathbf{x}^0) = \begin{Bmatrix} 2x_1 - 2x_2 - 4 \\ 4x_2 - 2x_1 \end{Bmatrix}_{\mathbf{x}^0} = \begin{Bmatrix} -4 \\ 2 \end{Bmatrix}$$

取 $\mathbf{p}_0 = -\mathbf{g}_0 = \begin{Bmatrix} 4 \\ -2 \end{Bmatrix}$

沿 \mathbf{p}_0 方向进行一维搜索, 得

$$\mathbf{x}^1 = \mathbf{x}^0 + \alpha_0 \mathbf{p}_0 = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} + \alpha_0 \begin{Bmatrix} 4 \\ -2 \end{Bmatrix} = \begin{Bmatrix} 1 + 4\alpha_0 \\ 1 - 2\alpha_0 \end{Bmatrix}$$

α_0 为最佳步长, 可通过 $f(\mathbf{x}^1) = \min_{\alpha} \varphi_1(\alpha)$, $\varphi'_1(\alpha_0) = 0$

求得 $\alpha_0 = \frac{1}{4}$

则 $\mathbf{x}^1 = \begin{Bmatrix} 1 - 4\alpha_0 \\ 1 - 2\alpha_0 \end{Bmatrix} = \begin{Bmatrix} 2 \\ \frac{1}{2} \end{Bmatrix}$

为建立第二个共轭方向 \mathbf{p}_1 , 需计算 \mathbf{x}^1 点处的梯度及系数 β_0 值, 得

$$\mathbf{g}_1 = \nabla f(\mathbf{x}^1) = \begin{Bmatrix} 2x_1 - 2x_2 - 4 \\ 4x_2 - 2x_1 \end{Bmatrix}_{\mathbf{x}^1} = \begin{Bmatrix} -1 \\ -2 \end{Bmatrix}$$

$$\beta_0 = \frac{\|\mathbf{g}_1\|^2}{\|\mathbf{g}_0\|^2} = \frac{5}{20} = \frac{1}{4}$$

从而求得第二个共轭方向

$$\mathbf{p}_1 = -\mathbf{g}_1 + \beta_0 \mathbf{p}_0 = \begin{Bmatrix} 1 \\ 2 \end{Bmatrix} + \frac{1}{4} \begin{Bmatrix} 4 \\ -2 \end{Bmatrix} = \begin{Bmatrix} 2 \\ \frac{3}{2} \end{Bmatrix}$$

再沿 \mathbf{p}_1 进行一维搜索, 得

$$\mathbf{x}^2 = \mathbf{x}^1 + \alpha_1 \mathbf{p}_1 = \begin{Bmatrix} 2 \\ \frac{1}{2} \end{Bmatrix} + \alpha_1 \begin{Bmatrix} 2 \\ \frac{3}{2} \end{Bmatrix} = \begin{Bmatrix} 2 + 2\alpha_1 \\ \frac{1}{2} + \frac{3}{2}\alpha_1 \end{Bmatrix}$$

α_1 为最佳步长, 通过 $f(\mathbf{x}^2) = \min_{\alpha} \varphi_2(\alpha)$, $\varphi'_2(\alpha_1) = 0$

求得 $\alpha_1 = 1$

则 $\mathbf{x}^2 = \begin{Bmatrix} 2 + 2\alpha_1 \\ \frac{1}{2} + \frac{3}{2}\alpha_1 \end{Bmatrix} = \begin{Bmatrix} 4 \\ 2 \end{Bmatrix}$

计算 \mathbf{x}^2 点处的梯度

$$\mathbf{g}_2 = \nabla f(\mathbf{x}^2) = \begin{Bmatrix} 2x_1 - 2x_2 - 4 \\ 4x_2 - 2x_1 \end{Bmatrix}_{\mathbf{x}^2} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

说明 \mathbf{x}^2 点满足极值必要条件, 再根据 \mathbf{x}^2 点的海赛矩阵

$$\mathbf{G}(\mathbf{x}^2) = \begin{bmatrix} 2 & -2 \\ -2 & 4 \end{bmatrix}$$

是正定的, 可知 \mathbf{x}^2 满足极值充分必要条件. 故 \mathbf{x}^2 为极小点, 即

$$\mathbf{x}^* = \mathbf{x}^2 = \begin{Bmatrix} 4 \\ 2 \end{Bmatrix}$$

而函数极小值为 $f(\mathbf{x}^*) = -8$

从共轭梯度法的计算过程可以看出，第一个搜索方向取作负梯度方向，这就是最速下降法。其余各步的搜索方向是将负梯度偏转一个角度，也就是对负梯度进行修正。所以共轭梯度法实质上是对最速下降法进行的一种改进，故它又被称作旋转梯度法。

上述共轭梯度法是 1964 年由弗来彻 (Fletcher) 和里维斯 (Reeves) 两人提出的。此法的优点是程序简单，存储量少，具有最速下降法的优点，而在收敛速度上比最速下降法快，具有二次收敛性。

§ 5 变尺度法

一、尺度矩阵的概念

变量的尺度变换是放大或缩小各个坐标，从而可以把函数的偏心程度降低到最低限度。尺度变换技巧能显著地改进几乎所有极小化方法的收敛性质。如在第一节用最速下降法求 $f(x_1, x_2) = x_1^2 + 25x_2^2$ 的极小值时，需要进行 10 次迭代才能达到极小点 $\mathbf{x}^* = (0 \ 0)^T$ 。但是，若作变换，

$$\begin{aligned} y_1 &= x_1 \\ y_2 &= 5x_2 \end{aligned}$$

即把 x_2 的尺度放大 5 倍，就可以将等值线为椭圆的函数 $f(x_1, x_2)$ 变换成等值线为圆的函数 $\psi(y_1, y_2) = y_1^2 + y_2^2$ ，从而消除了函数的偏心，用最速下降法只需一次迭代即可求得极小点。

对于一般二次函数

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$$

如果进行尺度变换

$$\mathbf{x} \leftarrow \mathbf{Q} \mathbf{x}$$

则在新的坐标系中，函数 $f(\mathbf{x})$ 的二次项变为

$$\frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} \rightarrow \frac{1}{2} \mathbf{x}^T \mathbf{Q}^T \mathbf{G} \mathbf{Q} \mathbf{x}$$

选择这样变换的目的，仍然是为了减低二次项的偏心程度。若矩阵 \mathbf{G} 是正定的，则总存在矩阵 \mathbf{Q} 使

$$\mathbf{Q}^T \mathbf{G} \mathbf{Q} = \mathbf{I} \text{ (单位矩阵)}$$

将函数偏心度变为零。

用 \mathbf{Q}^{-1} 右乘等式两边，得

$$\mathbf{Q}^T \mathbf{G} = \mathbf{Q}^{-1}$$

再用 \mathbf{Q} 左乘等式两边，得

$$\mathbf{Q} \mathbf{Q}^T \mathbf{G} = \mathbf{I}$$

所以 $QQ^T = G^{-1}$

这说明二次函数矩阵 G 的逆阵, 可以通过尺度变换矩阵 Q 来求得。这样, 牛顿法迭代过程中的牛顿方向便可写成

$$d^K = -G^{-1}\nabla f(x^K) = -QQ^T\nabla f(x^K)$$

牛顿迭代公式变为

$$x^{K+1} = x^K + \alpha_K d^K = x^K - \alpha_K QQ^T \nabla f(x^K)$$

例如在上例中, 二次函数

$$\begin{aligned} f(x_1, x_2) &= x_1^2 + 25x_2^2 \\ &= \frac{1}{2} [x_1 \ x_2] \begin{bmatrix} 2 & 0 \\ 0 & 50 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} \\ &= \frac{1}{2} x^T G x \end{aligned}$$

其中

$$G = \begin{bmatrix} 2 & 0 \\ 0 & 50 \end{bmatrix}$$

若取

$$Q = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{5\sqrt{2}} \end{pmatrix}$$

的变换 $x \leftarrow Qx$, 则在变换后的坐标系中, 矩阵 G 变为

$$Q^T G Q = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{5\sqrt{2}} \end{pmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 50 \end{bmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{5\sqrt{2}} \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

从而求得

$$G^{-1} = QQ^T = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{5\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{5\sqrt{2}} \end{pmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{50} \end{bmatrix}$$

这与在第二节牛顿法例题中所得结果一致, 而且只需通过一次迭代即可求得极小点 $x^* = [0, 0]^T$ 和极小值 $f(x^*) = 0$ 。

试比较牛顿法迭代公式

$$x^{K+1} = x^K - \alpha_K QQ^T \nabla f(x^K)$$

和梯度法迭代公式

$$x^{K+1} = x^K - \alpha_K \nabla f(x^K)$$

可以看出, 差别在于牛顿法中多了 QQ^T 部分。 QQ^T 实际上是在 X 空间内测量距离大小的一种度量, 称作尺度矩阵

$$H = QQ^T$$

如在未进行尺度变换前, 向量 X 长度的概念是

$$\|X\| = (X^T X)^{\frac{1}{2}}$$

变换后向量 X 对于 H 尺度下的长度

$$\begin{aligned} \|X\|_H &= [(QX)^T(QX)]^{\frac{1}{2}} \\ &= [X^T(QQ^T)X]^{\frac{1}{2}} \\ &= (X^T H X)^{\frac{1}{2}} \end{aligned}$$

这样的长度定义，在确定“长度”这个纯量大小时，使得某些方向起的作用比较大，另一些方向起的作用比较小。为使这种尺度有用，必须对一切非零向量的 X 均有 $X^T H X > 0$ ，即要求尺度矩阵 H 正定。既然牛顿法迭代公式可用尺度变换矩阵 $H = Q Q^T$ 表示出来，即

$$x^{K+1} = x^K - a_K H \nabla j(x^K)$$

它和梯度法迭代公式只差一个尺度矩阵 H ，那么牛顿法就可看成是经过尺度变换后的梯度法。经过尺度变换，使函数偏心率减小到零，函数的等值面变为球面（或超球面），使设计空间中任意点处函数的梯度都通过极小点，用最速下降法只需一次迭代就可达到极小点。这就是对变换前的二次函数，在使用牛顿方法时，由于其牛顿方向直接指向极小点，因此只需一次迭代就能找到极小点的原因所在。

二、变尺度矩阵的建立

对于一般函数 $f(x)$ ，当用牛顿法寻求极小点时，其牛顿迭代公式为

$$x^{K+1} = x^K - a_K G_K^{-1} g_K \quad (K=0, 1, 2, \dots)$$

其中

$$g_K \equiv \nabla f(x^K)$$

$$G_K \equiv \nabla^2 f(x^K)$$

为了避免在迭代公式中计算海赛矩阵的逆阵 G_K^{-1} ，可用在迭代中逐步建立的变尺度矩阵

$$H_K \equiv H(x^K)$$

来替换 G_K^{-1} ，即构造一个矩阵序列 $\{H_K\}$ 来逼近海赛逆矩阵序列 $\{G_K^{-1}\}$ 。每迭代一次，尺度就改变一次，这正是“变尺度”的含义。这样，上式变为

$$x^{K+1} = x^K - a_K H_K g_K \quad (K=0, 1, 2, \dots) \quad (3-21)$$

其中 a_K 是从 x^K 出发，沿方向

$$d^K = -H_K g_K$$

作一维搜索而得到的最佳步长。这个迭代公式代表面很广，例如当 $H_K = I$ （单位矩阵）时，它就变成最速下降法。以上就是变尺度法的基本思想。

为了使变尺度矩阵 H_K 确实与 G_K^{-1} 近似，并具有容易计算的特点，必须对 H_K 附加某些条件。

1. 为保证迭代公式具有下降性质，要求 $\{H_K\}$ 中的每一个矩阵都是对称正定的。因为若要求搜索方向 $p_K = -H_K g_K$ 为下降方向，即要求 $g_K^T p_K < 0$ 也就是 $-g_K^T H_K g_K < 0$ ，这样 $g_K^T H_K g_K > 0$ ，即 H_K 应为对称正定。

2. 要求 H_K 之间的迭代具有简单的形式。显然

$$H_{K+1} = H_K + E_K$$

为最简单的形式，其中 E_K 为校正矩阵，上式称作校正公式。

3. 要求 $\{H_K\}$ 必须满足拟牛顿条件。

所谓拟牛顿条件，可由下面的推导给出。设迭代过程已进行到 $K+1$ 步， x^{K+1} ， g_{K+1} 均已求出，现在推导 H_{K+1} 所必须满足的条件。当 $f(x)$ 为具有正定矩阵 G 的二次函数时，根据泰勒展开可得

$$g_{K+1} = g_K + G(x^{K+1} - x^K)$$

则

$$G^{-1}(g_{K+1} - g_K) = x^{K+1} - x^K$$

因为具有正定海赛矩阵 G_{K+1} 的一般函数，在极小点附近可用二次函数很好地近似，所以我们就联想到如果迫使 H_{K+1} 满足类似于上式的关系：

$$H_{K+1}(g_{K+1} - g_K) = x^{K+1} - x^K$$

那么 H_K 就可以很好地近似于 G_K^{-1} 。因此，把上面的关系式称作拟牛顿条件（或拟牛顿方程）。为简便起见，记

$$y_K = g_{K+1} - g_K$$

$$s_K = x^{K+1} - x^K$$

则拟牛顿条件可写成

$$H_{K+1}y_K = s_K \quad (3-22)$$

根据上述拟牛顿条件，不通过海赛矩阵求逆就可以构造一个矩阵 H_{K+1} 来逼近海赛矩阵的逆阵 G_{K+1}^{-1} ，这类方法统称作拟牛顿法。由于变尺度矩阵的建立应用了拟牛顿条件，所以变尺度法也是属于一种拟牛顿法。还可以证明，变尺度法对于具有正定矩阵 G 的二次函数，能产生对 G 共轭的搜索方向，因此变尺度法又可以看成是一种共轭方向法。

三、变尺度法的一般步骤

对一般多元函数 $f(x)$ ，用变尺度法求极小点 x^* ，其一般步骤是：

1. 选定初始点 x^0 和收敛精度 ε 。
2. 计算 $g_0 = \nabla f(x^0)$ ，选取初始对称正定矩阵 H_0 （例如 $H_0 = I$ ），置 $K \leftarrow 0$ 。
3. 计算搜索方向 $p_K = -H_K g_K$ 。
4. 沿 p_K 方向进行一维搜索 $x^{K+1} = x^K + \alpha_K p_K$ ，计算 $g_{K+1} = \nabla f(x^{K+1})$ ， $s_K = x^{K+1} - x^K$ ， $y_K = g_{K+1} - g_K$ 。
5. 判断是否满足迭代终止准则，若满足则 $x^* = x^{K+1}$ ，停机，否则转 6。
6. 当迭代 n 次后还没找到极小点时，重置 H_K 为单位矩阵 I ，并以当前设计点为初始点 $x^0 \leftarrow x^{K+1}$ ，返回到 2 进行下一轮迭代，否则转到 7。
7. 计算矩阵 $H_{K+1} = H_K + E_K$ ，置 $K \leftarrow K+1$ 返回到 3。

对于校正矩阵 E_K ，可由具体的公式来计算，不同的公式对应不同的变尺度法，将在下面进行讨论。但不论哪种变尺度法， E_K 必须满足拟牛顿条件

$$H_{K+1}y_K = s_K$$

即 $(H_K + E_K)y_K = s_K$
 或 $E_K y_K = s_K - H_K y_K$
 满足上式的 E_K 有无穷多个，因此上述变尺度法（属于拟牛顿法）构成一族算法。
 变尺度法计算程序框图如图（3—11）所示。

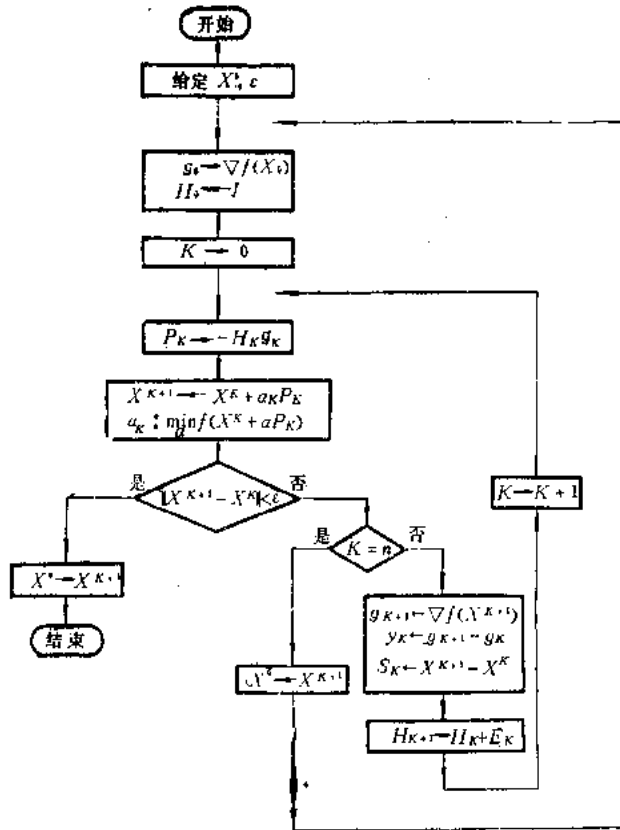


图 3—11

四、DFP 算法

在变尺度法中，校正矩阵 E_K 取不同的形式，就形成不同的变尺度法。DFP 算法中的校正矩阵 E_K 取下列形式：

$$E_K = \alpha_K u_K u_K^T + \beta_K v_K v_K^T \quad (3-23)$$

其中 u_K, v_K 是 n 维待定向量， α_K, β_K 是待定常数， $u_K u_K^T, v_K v_K^T$ 都是对称秩一的矩阵，它们可以说是一种最简单的矩阵。

根据校正矩阵 E_K 要满足拟牛顿条件

$$E_K y_K = s_K - H_K y_K$$

则有

$$(\alpha_K u_K u_K^T + \beta_K v_K v_K^T) y_K = s_K - H_K y_K$$

即

$$\alpha_K u_K u_K^T y_K + \beta_K v_K v_K^T y_K = s_K - H_K y_K$$

满足上面方程的待定向量 u_K 和 v_K 有多种取法，我们取

$$\alpha_K u_K u_K^T y_K = s_K$$

$$\beta_K \mathbf{v}_K \mathbf{v}_K^T \mathbf{y}_K = -\mathbf{H}_K \mathbf{y}_K$$

注意到 $\mathbf{u}_K^T \mathbf{y}_K$ 和 $\mathbf{v}_K^T \mathbf{y}_K$ 都是数量，不妨取

$$\mathbf{u}_K = \mathbf{s}_K$$

$$\mathbf{v}_K = \mathbf{H}_K \mathbf{y}_K$$

这样就可以定出

$$\alpha_K = \frac{1}{\mathbf{s}_K^T \mathbf{y}_K}$$

$$\beta_K = -\frac{1}{\mathbf{y}_K^T \mathbf{H}_K \mathbf{y}_K}$$

从而可得 DFP 算法的校正公式

$$\mathbf{H}_{K+1} = \mathbf{H}_K + \frac{\mathbf{s}_K \mathbf{s}_K^T}{\mathbf{s}_K^T \mathbf{y}_K} - \frac{\mathbf{H}_K \mathbf{y}_K \mathbf{y}_K^T \mathbf{H}_K}{\mathbf{y}_K^T \mathbf{H}_K \mathbf{y}_K} \quad (3-24)$$

$$(K=0, 1, 2, \dots)$$

DFP 算法的计算步骤和变尺度法的一般步骤相同，只是具体计算校正矩阵时应按上面公式进行。

例：用 DFP 算法求

$$f(x_1, x_2) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2$$

的极值解。

1. 取初始点 $\mathbf{x}^0 = [1, 1]^T$ ，为了按 DFP 法构造第一次搜寻方向 \mathbf{p}_0 ，需计算初始点处的梯度

$$\mathbf{g}_0 = \nabla f(\mathbf{x}^0) = \begin{Bmatrix} 2x_1 - 2x_2 - 4 \\ 4x_2 - 2x_1 \end{Bmatrix}_{\mathbf{x}^0} = \begin{Bmatrix} -4 \\ 2 \end{Bmatrix}$$

并取初始变尺度矩阵为单位矩阵 $\mathbf{H}_0 = \mathbf{I}$ ，则第一次搜寻方向为

$$\mathbf{p}_0 = -\mathbf{H}_0 \mathbf{g}_0 = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} -4 \\ 2 \end{Bmatrix} = \begin{Bmatrix} 4 \\ -2 \end{Bmatrix}$$

沿 \mathbf{p}_0 方向进行一维搜索，得

$$\begin{aligned} \mathbf{x}^1 &= \mathbf{x}^0 + \alpha_0 \mathbf{p}_0 \\ &= \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} + \alpha_0 \begin{Bmatrix} 4 \\ -2 \end{Bmatrix} = \begin{Bmatrix} 1 + 4\alpha_0 \\ 1 - 2\alpha_0 \end{Bmatrix} \end{aligned}$$

其中 α_0 为一维搜索最佳步长，应满足

$$f(\mathbf{x}^1) = \min_{\alpha} f(\mathbf{x}^0 + \alpha \mathbf{p}_0) = \min_{\alpha} (40\alpha^2 - 20\alpha - 3)$$

得

$$\alpha_0 = 0.25$$

$$\mathbf{x}^1 = \begin{Bmatrix} 2 \\ 0.5 \end{Bmatrix}$$

2. 再按 DFP 法构造 \mathbf{x}^1 点处的搜寻方向 \mathbf{p}_1 ，需计算

$$\mathbf{g}_1 = \begin{cases} 2x_1 - 2x_2 - 4 \\ 4x_2 - 2x_1 \end{cases}_{\mathbf{x}^1} = \begin{cases} -1 \\ -2 \end{cases}$$

$$\mathbf{y}_0 = \mathbf{g}_1 - \mathbf{g}_0 = \begin{cases} -1 \\ -2 \end{cases} - \begin{cases} -4 \\ 2 \end{cases} = \begin{cases} 3 \\ -4 \end{cases}$$

$$\mathbf{s}_0 = \mathbf{x}^1 - \mathbf{x}^0 = \begin{cases} 2 \\ 0.5 \end{cases} - \begin{cases} 1 \\ 1 \end{cases} = \begin{cases} 1 \\ -0.5 \end{cases}$$

代入校正公式 (3-24)

$$\begin{aligned} \mathbf{H}_1 &= \mathbf{H}_0 + \frac{\mathbf{s}_0 \mathbf{s}_0^T}{\mathbf{s}_0^T \mathbf{y}_0} - \frac{\mathbf{H}_0 \mathbf{y}_0 \mathbf{y}_0^T \mathbf{H}_0}{\mathbf{y}_0^T \mathbf{H}_0 \mathbf{y}_0} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{\begin{cases} 1 \\ -0.5 \end{cases} [1 \quad -0.5] \begin{cases} 3 \\ -4 \end{cases} [3 \quad -4]}{[1 \quad -0.5] \begin{cases} 3 \\ -4 \end{cases} [3 \quad -4] \begin{cases} 3 \\ -4 \end{cases}} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{1}{5} \begin{bmatrix} 1 & -0.5 \\ -0.5 & 0.25 \end{bmatrix} - \frac{1}{25} \begin{bmatrix} 9 & -12 \\ -12 & 16 \end{bmatrix} \\ &= \begin{bmatrix} 21 & 19 \\ 25 & 50 \\ 19 & 41 \\ 50 & 100 \end{bmatrix} \end{aligned}$$

则第二次搜寻方向为

$$\begin{aligned} \mathbf{p}_1 &= -\mathbf{H}_1 \mathbf{g}_1 \\ &= - \begin{bmatrix} 21 & 19 \\ 25 & 50 \\ 19 & 41 \\ 50 & 100 \end{bmatrix} \begin{cases} -1 \\ -2 \end{cases} = \begin{cases} 8 \\ 5 \\ 6 \\ 5 \end{cases} \end{aligned}$$

再沿 \mathbf{p}_1 进行一维搜索, 得

$$\begin{aligned} \mathbf{x}^2 &= \mathbf{x}^1 + \alpha_1 \mathbf{p}_1 \\ &= \begin{cases} 2 \\ 0.5 \end{cases} + \alpha_1 \begin{cases} 8 \\ 5 \\ 6 \\ 5 \end{cases} = \begin{cases} 2 + \frac{8}{5} \alpha_1 \\ 0.5 + \frac{6}{5} \alpha_1 \end{cases} \end{aligned}$$

其中 α_1 为一维搜索最佳步长, 应满足

$$f(\mathbf{x}^2) = \min_{\alpha} f(\mathbf{x}^1 + \alpha \mathbf{p}_1) = \min_{\alpha} \left(\frac{8}{5} \alpha^2 - 4\alpha + \frac{11}{2} \right)$$

得 $\alpha_1 = \frac{5}{4}$

$$\mathbf{x}^2 = \begin{Bmatrix} 4 \\ 2 \end{Bmatrix}$$

3. 为了判断 \mathbf{x}^2 点是否为极值点, 需计算 \mathbf{x}^2 点处的梯度及其海赛矩阵

$$\mathbf{g}_2 = \begin{Bmatrix} 2x_1 - 2x_2 - 4 \\ 4x_2 - 2x_1 \end{Bmatrix} \Big|_{\mathbf{x}^2} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

$$\nabla^2 f(\mathbf{x}^2) = \begin{bmatrix} 2 & -2 \\ -2 & 4 \end{bmatrix}$$

梯度为零向量, 海赛矩阵正定, 可见 \mathbf{x}^2 点满足极值充要条件, 因此 \mathbf{x}^2 为极小点。此函数的极值解为

$$\mathbf{x}^* = \mathbf{x}^2 = [4 \ 2]^T$$

$$f(\mathbf{x}^*) = -8$$

当初始矩阵 \mathbf{H}_0 选为对称正定矩阵时, DFP 算法将保证以后的迭代矩阵 \mathbf{H}_K 都是对称正定的, 即使将 DFP 算法施用于非二次函数也是如此, 从而保证算法总是下降的。这种算法用于高维问题 (如 20 个变量以上), 收敛速度快, 效果好。DFP 算法是无约束优化方法中最有效的方法之一, 因为它不单纯是利用向量传递信息, 还采用了矩阵来传递信息。DFP 算法是戴维登 (Davidon) 于 1959 年提出的, 后来由弗来彻 (Fletcher) 和鲍威尔 (Powell) 于 1963 年作了改进, 故用三人名字的字头命名。

DFP 算法由于舍入误差和一维搜索不精确, 有可能导致 \mathbf{H}_K 奇异, 而使数值稳定性方面不够理想。所以 1970 年提出更稳定的算法公式, 称作 EFGS (Broyden—Fletcher—Goldfarb—Shanno) 算法, 其校正公式为

$$\mathbf{H}_{K+1} = \mathbf{H}_K + \left[\left(1 + \frac{\mathbf{y}_K^T \mathbf{H}_K \mathbf{y}_K}{\mathbf{s}_K^T \mathbf{y}_K} \right) \mathbf{s}_K \mathbf{s}_K^T - \mathbf{H}_K \mathbf{y}_K \mathbf{s}_K^T - \mathbf{s}_K \mathbf{y}_K^T \mathbf{H}_K \right] / \mathbf{s}_K^T \mathbf{y}_K \quad (3-25)$$

因为变尺度法的有效性促使其不断发展, 所以出现过许多变尺度的算法。1970 年黄 (Huang) 从共轭条件出发对变尺度法做了统一处理, 写出了统一公式

$$\mathbf{u}_K = a_{11}^K \mathbf{s}_K + a_{12}^K \mathbf{H}_K \mathbf{y}_K$$

$$\mathbf{v}_K = a_{21}^K \mathbf{s}_K + a_{22}^K \mathbf{H}_K \mathbf{y}_K$$

并取

$$\mathbf{E}_K = \mathbf{s}_K (\mathbf{u}_K)^T + \mathbf{H}_K \mathbf{y}_K (\mathbf{v}_K)^T \quad (3-26)$$

可以看出, 当取 $a_{12}^K = a_{21}^K = 0$, $a_{11}^K = a_{22}^K$, $a_{22}^K = \beta_K$ 时, 就是 DFP 法的公式。

取 $a_{12}^K = a_{21}^K$, $a_{22}^K = 0$ 时, 就得到 EFGS 法的公式。

还可以取 $\mathbf{u}_K = -\mathbf{v}_K$ 及 $a_{11}^K = 0$ 或 $a_{12}^K = 0$, $a_{11}^K = a_{22}^K = -a_{12}^K = -a_{21}^K$ 等等。

就是说, 对 a_{ij}^K 的不同赋值即可得不同变尺度算法。如取 $a_{11}^K = a_{22}^K = 0$ 时得麦考密克 (McCormick) 算法; 取 $a_{11}^K = a_{22}^K = 0$ 时, 即得皮尔逊 (Pearson) 算法等等。

§ 6 鲍威尔 (Powell) 法

鲍威尔法是直接利用函数值来构造共轭方向的一种共轭方向法。这种方法是在研究

具有正定矩阵 G 的二次函数

$$f(x) = \frac{1}{2} x^T G x + b^T x + c$$

的极小化问题时形成的，其基本思想是在不用导数的前提下，在迭代中逐次构造 G 的共轭方向。

一、共轭方向的生成

设 x^K, x^{K+1} 为从不同点出发，沿同一方向 p_j 进行一维搜索而得到的两个极小点，如图 (3-12) 所示。根据梯度和等值面相垂直的性质， p_j 和 x^K, x^{K+1} 两点处的梯度 g_K, g_{K+1} 之间存在关系：

$$\begin{aligned} p_j^T g_K &= 0 \\ p_j^T g_{K+1} &= 0 \end{aligned}$$

另一方面，对于上述二次函数，其 x^K, x^{K+1} 两点处的梯度可表示为

$$\begin{aligned} g_K &= G x^K + b \\ g_{K+1} &= G x^{K+1} + b \end{aligned}$$

两式相减，得

$$g_{K+1} - g_K = G(x^{K+1} - x^K)$$

因而有

$$p_j^T (g_{K+1} - g_K) = p_j^T G(x^{K+1} - x^K) = 0 \quad (3-27)$$

若取方向 $p_K = x^{K+1} - x^K$ ，如图 (3-12) 所示，则 p_K 和 p_j 对 G 共轭。这说明只要沿 p_j 方向分别对函数作两次一维搜索，得到两个极小点 x^K 和 x^{K+1} ，那么这两点的连线所给出的方向就是与 p_j 一起对 G 共轭的方向。

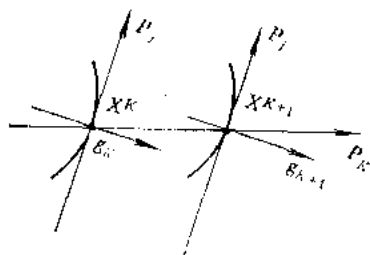


图 3-12

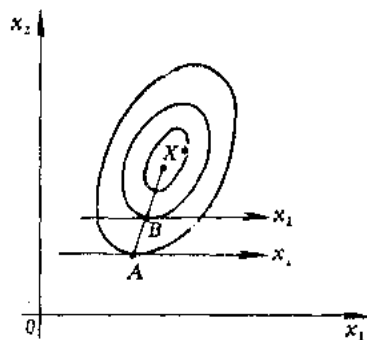


图 3-13

对于二维问题， $f(x)$ 的等值线为一族椭圆，A, B 为沿 x_1 方向上的两个极小点，分别处于等值线与 x_1 方向的切点上，如图 (3-13) 所示。根据上述分析，则 A, B 两点的连线 \overline{AB} 就是与 x_1 轴一起对 G 共轭的方向。沿此共轭方向进行一维搜索就可找到函数 $f(x)$ 的极小点 x^* 。

二、基本算法

现在针对二维情况来描述鲍威尔的基本算法，如图 (3-14) 所示。

1. 任选一初始点 x^0 ，再选两个线性无关的向量，如坐标轴单位向量 $e_1 = [1 \ 0]^T$ 和

$e_2 = [0, 1]^T$ 作为初始搜索方向。

2. 从 x^0 出发, 顺次沿 e_1, e_2 作一维搜索得点 x_1^0, x_2^0 , 二点连线得一新方向

$$p_1 = x_2^0 - x_1^0$$

用 p_1 代替 e_1 形成两个线性无关向量 e_2, p_1 , 作为下一轮迭代的搜索方向。再从 x_1^0 出发, 沿 p_1 作一维搜索得点 x_0^1 , 作为下一轮迭代的初始点。

3. 从 x_1^1 出发, 顺次沿 e_2, p_1 作一维搜索, 得到点 x_1^1, x_2^1 , 二点连线得一新方向

$$p_2 = x_2^1 - x_1^1$$

x_1^1, x_2^1 两点是从不同点 x_0^1, x_1^1 出发, 分别沿 p_1 方向进行一维搜索而得的极小点, 因此 x_0^1, x_2^1 两点连线的方向 p_2 同 p_1 一起对 G 共轭。再从 x_0^1 出发, 沿 p_2 作一维搜索得点 x^2 , 因为 x^2 相当于从 x^0 出发分别沿 G 的两个共轭方向 p_1, p_2 进行两次一维搜索而得到的点, 所以 x^2 点即是二维问题的极小点 x^* 。

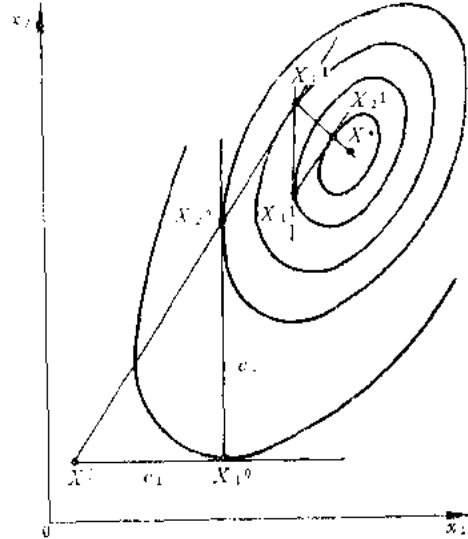


图 3-14

把二维情况的基本算法扩展到 n 维, 则鲍威尔基本算法的要点是: 在每一轮迭代中总有一个始点 (第一轮的始点是任选的初始点) 和 n 个线性独立的搜索方向。从始点出发顺次沿 n 个方向作一维搜索得一终点, 由始点和终点决定了一个新的搜索方向。用这个方向替换原来 n 个方向中的一个, 于是形成新的搜索方向组。替换的原则是去掉原方向组的第一个方向而将新方向排在原方向的最后。此外规定, 从这一轮的搜索终点出发沿新的搜索方向作一维搜索而得到的极小点, 作为下一轮迭代的始点。这样就形成算法的循环。因为这种方法在迭代中逐次生成共轭方向, 而共轭方向是较好的搜索方向, 所以鲍威尔法又称作方向加速法。

上述基本算法仅具有理论意义, 不要说对于一般函数, 就是对于二次函数, 这个算法也可能失效, 因为在迭代中的 n 个搜索方向有时变成线性相关而不能形成共轭方向。这时张不成 n 维空间, 可能求不到极小点, 所以上述基本算法有待改进。

三、改进的算法

在鲍威尔基本算法中, 每一轮迭代都用连结始点和终点所产生出的搜索方向去替换原向量组中的第一个向量, 而不管它的“好坏”, 这是产生向量组线性相关的原因所在。因此在改进的算法中首先判断原向量组是否需要替换, 如果需要替换, 还要进一步判断原向量组中哪个向量最坏, 然后再用新产生的向量替换这个最坏的向量, 以保证逐次生成共轭方向。

改进算法的具体步骤如下:

1. 给定初始点 x^0 (记作 x_0^0), 选取初始方向组, 它由 n 个线性无关的向量 $p_1^1,$

p_1^k, \dots, p_n^k (如 n 个坐标轴单位向量 e_1, e_2, \dots, e_n) 所组成, 置 $K \leftarrow 0$ 。

2. 从 x_0^K 出发, 顺次沿 $p_1^K, p_2^K, \dots, p_n^K$ 作一维搜索得 $x_1^K, x_2^K, \dots, x_n^K$ 。接着以 x_n^K 为起点, 沿方向

$$p_{n+1}^k = x_n^K - x_0^K$$

移动一个 $x_n^K - x_0^K$ 的距离, 得到

$$x_{n+1}^k = x_n^K + (x_n^K - x_0^K) = 2x_n^K - x_0^K$$

x_0^K, x_n^K, x_{n+1}^k 分别称为一轮迭代的始点, 终点和反射点。始点、终点和反射点所对应的函数值分别表示为

$$F_0 = f(x_0^K)$$

$$F_n = f(x_n^K)$$

$$F_{n+1} = f(x_{n+1}^k)$$

同时计算各中间点处的函数值, 并记为

$$f_i = f(x_i^K) \quad (i = 1, 2, \dots, n)$$

计算 n 个函数值之差 $f_0 - f_1, f_1 - f_2, \dots, f_{n-1} - f_n$

记作 $\Delta_i = f_{i-1} - f_i \quad (i = 1, 2, \dots, n)$

其中最大者记作

$$\Delta_m = \max_{1 \leq i \leq n} \Delta_i = f_{m-1} - f_m$$

3. 根据下述判别条件确定是否要对原方向组进行替换。

若 $2\Delta_m \leq F_0 - 2F_n + F_{n+1}$, 下轮迭代仍用原方向组, 并以 x_n^K, x_{n+1}^k 中函数值小者作为下轮迭代的始点。

若 $2\Delta_m > F_0 - 2F_n + F_{n+1}$, 下轮迭代应对原方向组进行替换, 将 p_m^K 补充到原方向组的最后位置, 而除掉 p_n^K 。即新方向组为 $p_1^K, p_2^K, \dots, p_{m-1}^k, p_{m+1}^k, \dots, p_n^K, p_m^K$ 作为下轮迭代的搜索方向。下轮迭代的始点取为沿 p_{n+1}^k 方向进行一维搜索的极小点 x_0^{K+1} 。

4. 判断是否满足收敛准则, 若满足则取 x_0^{K+1} 为极小点, 否则置 $K \leftarrow K+1$, 返回 2, 继续进行下一轮迭代。

这样重复迭代的结果, 后面加进去的向量都彼此对 G 共轭, 经 n 轮迭代即可得到一个由 n 个共轭方向所组成的方向组。对于二次函数, 最多不超过 n 次就可找到极小点, 而对一般函数, 往往要超过 n 次才能找到极小点 (这里的“ n ”表示设计空间的维数)。

改进后的鲍威尔法程序框图如图 (3-15) 所示。

例: 用鲍威尔法求函数

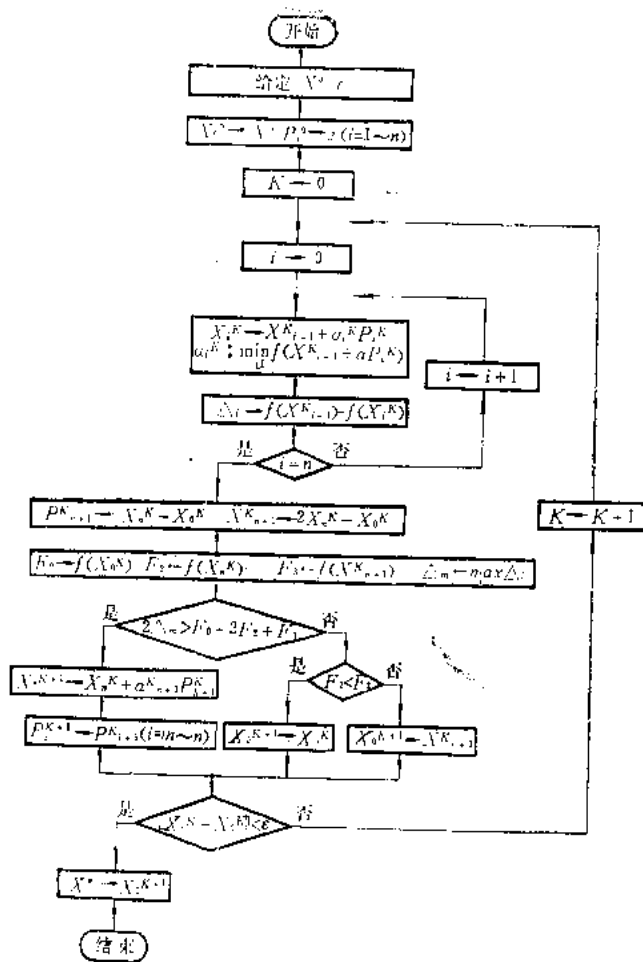
$$f(x_1, x_2) = 10(x_1 + x_2 - 3)^2 + (x_1 - x_2)^2$$

的极小值。

选初始点 $x_0^0 = [0, 0]^T$, 初始搜索方向 $p_1^0 = e_1 = [1, 0]^T, p_2^0 = e_2 = [0, 1]^T$ 。初始点处的函数值 $F_0 = f_0 = f(x_0^0) = 25$ 。

第一轮迭代:

1. 沿 p_1^0 方向进行一维搜索, 得



(注: 此图第五框中的 $i \leftarrow 0$ 应为 $i \leftarrow 1$)

图 3-15

$$x_1^0 = x_0^0 + \alpha_1 p_1^0 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} + \alpha_1 \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} = \begin{Bmatrix} \alpha_1 \\ 0 \end{Bmatrix}$$

$$f_1 = f(x_1^0) = 10(\alpha_1 - 5)^2 + \alpha_1^2$$

最佳步长 α_1 可通过

$$\frac{\partial f_1}{\partial \alpha_1} = 20(\alpha_1 - 5) + 2\alpha_1 = 0$$

得

$$\alpha_1 = \frac{100}{22} = 4.5455$$

$$x_1^0 = \begin{Bmatrix} 4.5455 \\ 0 \end{Bmatrix}$$

从而算出 x_1^0 点处的函数值及沿 p_1^0 走步后函数值的增量

$$f_1 = f(x_1^0) = 22.727$$

$$\Delta_1 = f_0 - f_1 = 250 - 22.727 = 227.273$$

2. 再沿 p_2^0 方向进行一维搜索, 得

$$x_2^0 = x_1^0 + \alpha_2 p_2^0 = \begin{Bmatrix} 4.5455 \\ 0 \end{Bmatrix} + \alpha_2 \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 4.5455 \\ \alpha_2 \end{Bmatrix}$$

$$f_2 = f(x_2^0) = 10(4.5455 + \alpha_2 - 5)^2 + (4.5455 - \alpha_2)^2$$

最佳步长 α_2 的计算可根据

$$\frac{\partial f_2}{\partial \alpha_2} = 20(\alpha_2 - 0.4545) - 2(4.5455 - \alpha_2) = 0$$

进行, 得

$$\alpha_2 = \frac{18.181}{22} = 0.8264$$

$$x_2^0 = \begin{Bmatrix} 4.5455 \\ 0.8264 \end{Bmatrix}$$

从而算出第一轮终点 x_2^0 处的函数值及沿 p_2^0 走步后的函数值增量

$$F_2 = f_2 = f(x_2^0) = 15.214$$

$$\Delta_2 = f_1 - f_2 = 22.727 - 15.214 = 7.513$$

取沿 p_1^0, p_2^0 走步后函数值增量中的最大者

$$\Delta_m = \Delta_1 = 227.273$$

终点 x_2^0 的反射点及其函数值为

$$x_3^0 = 2x_2^0 - x_0^0 = 2 \begin{Bmatrix} 4.5455 \\ 0.8264 \end{Bmatrix} - \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 9.091 \\ 1.6528 \end{Bmatrix}$$

$$F_3 = f(x_3^0) = 385.24$$

3. 为确定下一轮迭代的搜索方向和起始点, 需检查判别条件 $2\Delta_m > F_0 - 2F_2 + F_3$ 是否满足。

因为 $F_0 - 2F_2 + F_3 = 250 - 2 \times 15.214 + 385.24 = 604.812$, 所以不满足判别条件, 因而下轮迭代应继续使用原来的搜索方向 e_1, e_2 。

因为 $F_2 < F_3$, 所以取 x_2^0 为下轮迭代起始点。

第二轮迭代:

第二轮初始点及其函数值为

$$x_0^1 = x_2^0 = \begin{Bmatrix} 4.5455 \\ 0.8264 \end{Bmatrix}$$

$$F_0 = f_0 = f(x_0^1) = 15.214$$

1. 沿 e_1 方向 (即 x_1 轴方向) 进行一维搜索, 相当于固定 $x_2 = 0.826$, 改变 x_1 使函数 $f(x_1, x_2)$ 的值极小。设计点 x_1^1 位置可通过函数对 x_1 的偏导数等于零求得, 即

$$f(x) = 10(x_1 + 0.8264 - 5)^2 + (x_1 - 0.8264)^2$$

$$\frac{\partial f}{\partial x_1} = 20(x_1 - 4.1736) + 2(x_1 - 0.8264) = 0$$

$$x_1 = \frac{85.1248}{22} = 3.8693$$

得

$$x_1^1 = \begin{Bmatrix} 3.8693 \\ 0.8264 \end{Bmatrix}$$

x_1^1 点处的函数值及函数增量值分别为

$$f_1 = f(x_1^1) = 10.185$$

$$\Delta_1 = f_0 - f_1 = 15.214 - 10.185 = 5.029$$

2. 再沿 e_2 方向 (即 x_2 轴方向) 进行一维搜索, 相当于固定 $x_1 = 3.869$, 改变 x_2 使函数 $f(x_1, x_2)$ 的值极小。设计点 x_2^1 位置可通过函数对 x_2 的偏导数等于零求得, 即

$$f(x) = 10(3.8693 + x_2 - 5)^2 + (3.8693 - x_2)^2$$

$$\frac{\partial f}{\partial x_2} = 20(x_2 - 1.1307) - 2(3.8693 - x_2) = 0$$

$$x_2 = \frac{30.3526}{22} = 1.3797$$

得

$$x_2^1 = \begin{Bmatrix} 3.8693 \\ 1.3797 \end{Bmatrix}$$

第二轮终点 x_2^1 处的函数值及沿 x_2 方向函数增量值分别为

$$F_2 = f_2 = f(x_2^1) = 6.818$$

$$\Delta_2 = f_1 - f_2 = 10.185 - 6.818 = 3.367$$

取沿 x_1, x_2 方向函数增量值最大者

$$\Delta_m = \Delta_1 = 5.029$$

终点 x_2^1 的反射点及其函数值分别为

$$x_3^1 = 2x_2^1 - x_0^1 = 2 \begin{Bmatrix} 3.8693 \\ 1.3797 \end{Bmatrix} - \begin{Bmatrix} 4.5455 \\ 0.8264 \end{Bmatrix} = \begin{Bmatrix} 3.1931 \\ 1.9330 \end{Bmatrix}$$

$$F_3 = f(x_3^1) = 1.747$$

3. 为确定下轮迭代的搜索方向和起始点, 需检查判别条件 $2\Delta_m > F_0 - 2F_2 + F_3$

$$F_0 - 2F_2 + F_3 = 15.214 - 2 \times 6.818 + 1.747 = 3.325$$

故判别条件满足, 应进行方向替换。用新方向 p_3^1 替换 e_1 , 下轮迭代搜索方向为 e_2, p_3^1 。

$$p_3^1 = x_2^1 - x_0^1 = \begin{Bmatrix} 3.8693 \\ 1.3797 \end{Bmatrix} - \begin{Bmatrix} 4.5455 \\ 0.8264 \end{Bmatrix} = \begin{Bmatrix} -0.6762 \\ 0.5533 \end{Bmatrix}$$

下轮迭代起始点 x_0^2 为从 x_2^1 出发, 沿 p_3^1 方向进行一维搜索的极小点, 可通过下面计算求得。

$$x_0^2 = x_2^1 + \alpha_3 p_3^1 = \begin{Bmatrix} 3.8693 \\ 1.3797 \end{Bmatrix} + \alpha_3 \begin{Bmatrix} -0.6762 \\ 0.5533 \end{Bmatrix}$$

$$= \begin{Bmatrix} 3.8693 - 0.6762\alpha_3 \\ 1.3797 + 0.5533\alpha_3 \end{Bmatrix}$$

$$f(x_3^2) = 10(3.8693 - 0.6762\alpha_3 + 1.3797 + 0.5533\alpha_3 - 5)^2 + (3.8693 - 0.6762\alpha_3 - 1.3797 - 0.5533\alpha_3)^2$$

通过 $\frac{df}{d\alpha_3} = 0$

求得
$$\alpha_3 = \frac{24896}{1229} = 2.0257$$

因此，下轮迭代初始点及其函数值为

$$x_0^2 = \begin{cases} 2.4995 \\ 2.5091 \end{cases}$$

$$F_0 = f_0 = (x_0^2) = 0.0008$$

可见已足够接近极值点 $x^* = (2.5 \ 2.5)^T$ 及极小值 $f(x^*) = 0$ 。

鲍威尔方法是鲍威尔于1964年提出的，以后又经过他本人的改进。该法是一种有效的共轭方向法，它可以在有限步内找到二次函数的极小点。对于非二次函数只要具有连续二阶导数，用这种方法也是有效的。

§ 7 单形替换法

一、基本原理

函数的导数是函数性态的反映，它对选择搜寻方向提供了有用的信息，如最速下降法、共轭梯度法、变尺度法和牛顿法等，都是利用函数一阶或二阶导数信息来建立搜寻方向。在不计算导数的情况下，先算出若干点处的函数值，从它们之间的大小关系中也可以看出函数变化的大概趋势，为寻求函数的下降方向提供依据。这里所说的若干点，一般取在单纯形的顶点上。所谓单纯形是指在 n 维空间中具有 $n+1$ 个顶点的多面体。利用单纯形的顶点，计算其函数值并加以比较，从中确定有利的搜索方向和步长，找到一个较好的点取代单纯形中较差的点，组成新的单纯形来代替原来的单纯形。使新单纯形不断地向目标函数的极小点靠近，直到搜索到极小点为止。这就是单形替换法的基本思想。在线性规划中，我们将提到单纯形法，那是因为线性规划问题是在凸多面体顶点集上进行迭代求解。这里是无约束极小化中的单形替换法，利用不断替换单纯形来寻找无约束极小点。虽然二者都用到单纯形，但决不可以把这两种方法混淆起来。为此我们将通常在无约束极小化中所说的单纯形法，称作单形替换法，以避免和线性规划中的单纯形法相混淆。

现以二元函数 $f(x_1, x_2)$ 为例，说明单形替换法的基本原理。

如图 (3-16) 所示，在 x_1-x_2 平面上取不在同一直线上的三点 x_1, x_2, x_3 ，以它们为顶点组成一单纯形（即三角形）。计算各顶点函数值，设

$$f(x_1) > f(x_2) > f(x_3)$$

这说明 x_3 点最好， x_1 点最差。为了寻找极小点，一般说来，应向最差点的反对称方向进行搜索，即通过 x_1 并穿过 x_2x_3 的中点 x_4 的方向进行搜索。在此方向上取点 x_5 ，使

$$\bar{x}_5 = x_4 + (x_4 - x_1) = 2x_4 - x_1$$

x_5 点称作 x_1 点相对于 x_4 点的反射点，计算反射点的函数值 $f(x_5)$ ，可能出现以下几种情形。

1. $f(x_5) < f(x_3)$

即反射点比最好点还好，说明搜索方向正确，还可以往前迈进一步，也就是可以扩张。这时取扩张点

$$x_6 = x_4 + \alpha(x_4 - x_1)$$

其中 α 为扩张因子，一般取 $\alpha = 1.2 \sim 2.0$

如果 $f(x_6) < f(x_5)$ ，说明扩张有利，就以 x_6 代替 x_1 构成新单纯形 $x_2x_3x_6$ 。否则说明扩张不利，舍弃 x_6 ，仍以 x_5 代替 x_1 构成新单纯形 $x_2x_3x_5$ 。

2. $f(x_3) \leq f(x_5) < f(x_2)$

即反射点比最好点差，但比次差点好，说明反射可行，则以反射点代替最差点，仍构成新单纯形 $x_2x_3x_5$ 。

3. $f(x_2) \leq f(x_5) < f(x_1)$

即反射点比次差点差，但比最差点好，说明 x_5 走得太远，应缩回一些，即收缩。这时取收缩点

$$x_7 = x_4 + \beta(x_5 - x_4)$$

其中 β 为收缩因子，常取成 0.5。

如果 $f(x_7) < f(x_1)$ ，则用 x_7 代替 x_1 构成新单纯形 $x_2x_3x_7$ ，否则 x_7 不用。

4. $f(x_5) \geq f(x_1)$

即反射点比最差点还差，这时应收缩得更多一些，即将新点收缩在 x_1x_4 之间，取收缩点

$$x_8 = x_4 - \beta(x_4 - x_1) = x_4 + \beta(x_1 - x_4)$$

如果 $f(x_8) < f(x_1)$ ，则用 x_8 代替 x_1 构成新单纯形 $x_2x_3x_8$ ，否则 x_8 不用。

5. 若 x_1x_4 方向上的所有点都比最差点差，即 $f(x) > f(x_1)$ ，说明不能沿此方向搜索。这时应以 x_3 为中心缩边，使顶点 x_1 ， x_2 向 x_3 移近一半距离，得新单纯形 $x_3x_9x_{10}$ ，如图 (3-17) 所示，在此基础上进行寻优。

以上说明，可以通过反射，扩张，收缩和缩边等方式得到一个新单纯形，其中至少有一个顶点的函数值比原单纯形要小。

二、计算步骤

将上述对二元函数的处置方法扩展应用到多元函数 $f(x)$ 中，其计算步骤如下：

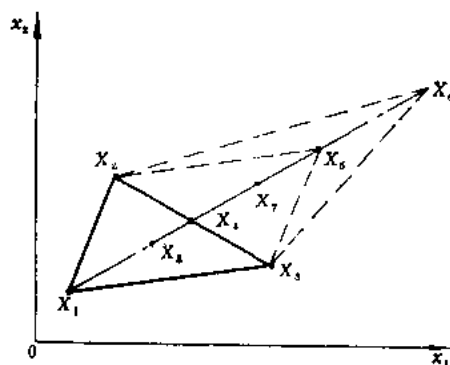


图 3-16

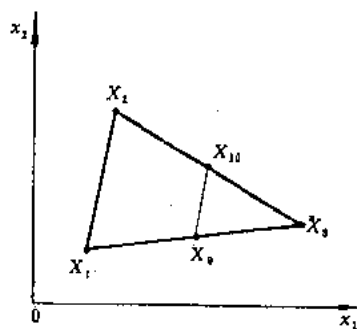


图 3-17

1. 构造初始单纯形。选初始点 x_0 。从 x_0 出发沿各坐标轴方向走步长 h , 得 n 个顶点 $x_i (i=1, 2, \dots, n)$ 与 x_0 构成初始单纯形。这样可以保证此单纯形各棱是 n 个线性无关的向量, 否则就会使搜索范围局限在某个较低维的空间内, 有可能找不到极小点。

2. 计算各顶点函数值

$$f_i = f(x_i) \quad (i=0, 1, 2, \dots, n)$$

3. 比较函数值的大小, 确定最好点 x_L , 最差点 x_H 和次差点 x_G , 即有

$$f_L = f(x_L) = \min_i f_i \quad (i=0, 1, 2, \dots, n)$$

$$f_H = f(x_H) = \max_i f_i \quad (i=0, 1, 2, \dots, n)$$

$$f_G = f(x_G) = \max_i f_i \quad (i=0, 1, 2, \dots, h-1, h+1, \dots, n)$$

4. 检验是否满足收敛准则

$$\left| \frac{f_H - f_L}{f_L} \right| < \epsilon$$

如满足则 $x^* = x_L$, 停机, 否则转 5。

5. 计算除 x_H 点之外各点的“重心” x_{n+1}

$$x_{n+1} = \frac{1}{n} \left(\sum_{i=0}^n x_i - x_H \right) \quad (3-28)$$

反射点

$$x_{n+2} = 2x_{n+1} - x_H \quad (3-29)$$

$$f_{n+2} = f(x_{n+2})$$

当 $f_L \leq f_{n+2} < f_G$ 时, 以 x_{n+2} 代替 x_H , f_{n+2} 代替 f_H , 构成一新单纯形, 然后返回到 3。

6. 扩张: 当 $f_{n+2} < f_L$ 时, 取扩张点

$$x_{n+3} = x_{n+1} + \alpha(x_{n+2} - x_{n+1}) \quad (3-30)$$

并计算其函数值 $f_{n+3} = f(x_{n+3})$ 。若 $f_{n+3} < f_{n+2}$ 则以 x_{n+3} 代替 x_H , f_{n+3} 代替 f_H 形成一新单纯形; 否则以 x_{n+2} 代替 x_H , f_{n+2} 代替 f_H 形成新单纯形, 然后返回到 3。

7. 收缩: 当 $f_{n+2} \geq f_G$ 时则需收缩。如果 $f_{n+2} < f_H$, 则取收缩点

$$x_{n+4} = x_{n+1} + \beta(x_{n+2} - x_{n+1}) \quad (3-31)$$

并计算其函数值 $f_{n+4} = f(x_{n+4})$, 否则在上式中以 x_H 代替 x_{n+2} , 计算收缩点 x_{n+4} 及其函数值 f_{n+4} 。如果 $f_{n+4} < f_H$, 则以 x_{n+4} 代替 x_H , f_{n+4} 代替 f_H , 得新单纯形, 返回到 3, 否则转 8。

8. 缩边: 将单纯形缩边, 可将各向量

$$x_i - x_L \quad (i=0, 1, 2, \dots, n)$$

的长度都缩小一半, 即

$$x_i = x_L - \frac{1}{2}(x_i - x_L) = \frac{1}{2}(x_i + x_L) \quad (i=0, 1, 2, \dots, n)$$

并返回到 2。

单形替换法计算框图如图 (3-18) 所示。

例: 试用单形替换法求

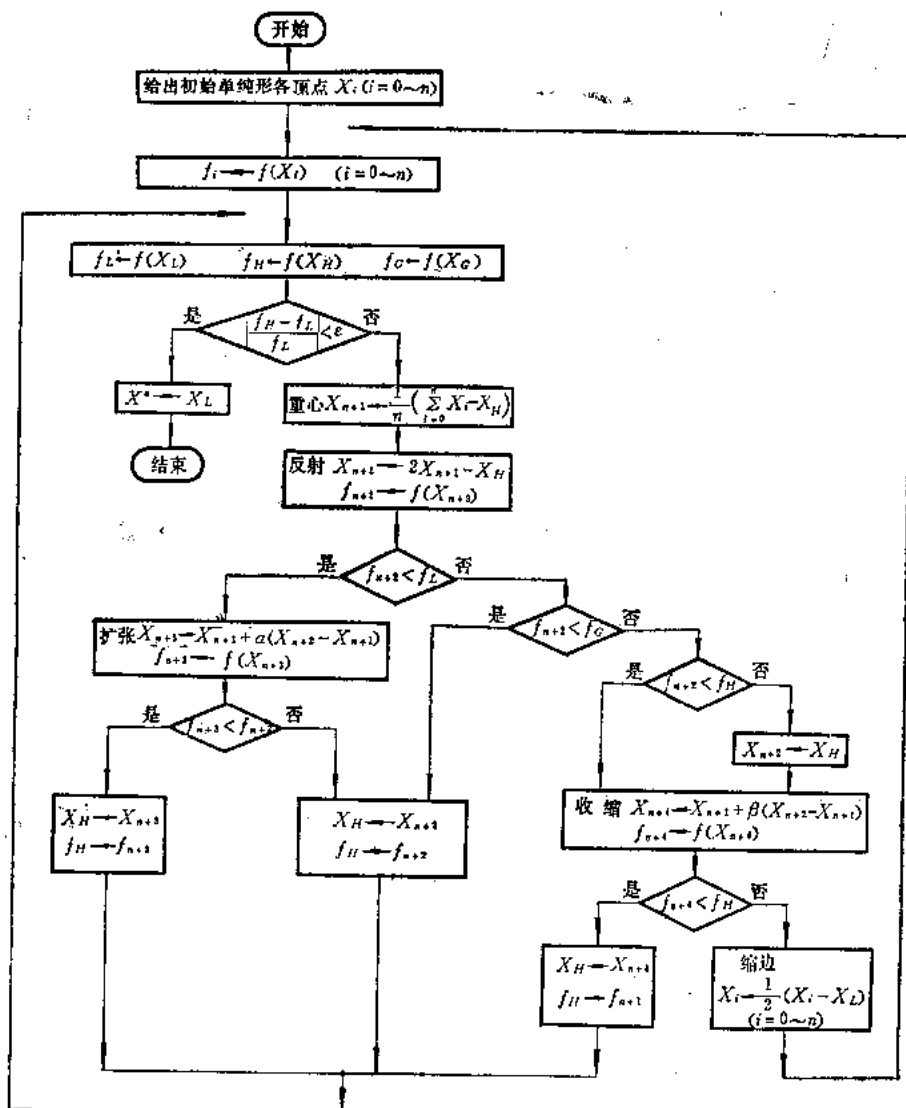


图 3-18

$$f(x_1, x_2) = 4(x_1 - 5)^2 + (x_2 - 6)^2$$

的极小值。

选 $x_0 = [8 \ 9]^T$, $x_1 = [10 \ 11]^T$, $x_2 = [8 \ 11]^T$ 为顶点作初始单纯形, 如图 (3-19) 所示。计算各顶点函数值 $f_0 = f(x_0) = 45$, $f_1 = f(x_1) = 125$, $f_2 = f(x_2) = 61$ 。可见最好点 $x_L = x_0$, 最差点 $x_H = x_1$, 次差点 $x_G = x_2$ 。

求 x_0, x_2 的重心 x_3

$$x_3 = \frac{1}{n} \left(\sum_{i=0}^n x_i - x_H \right) = \frac{1}{2} (x_0 + x_2) = \begin{Bmatrix} 8 \\ 10 \end{Bmatrix}$$

求反射点 x_4 及其函数值 f_4

$$x_4 = 2x_3 - x_1 = 2 \begin{Bmatrix} 8 \\ 10 \end{Bmatrix} - \begin{Bmatrix} 10 \\ 11 \end{Bmatrix} = \begin{Bmatrix} 6 \\ 9 \end{Bmatrix}$$

$$f_4 = f(x_4) = 13$$

由于 $f_4 < f_0$, 故需扩张, 取 $\alpha = 2$ 得扩张点 x_5

$$\begin{aligned} x_5 &= x_3 + 2(x_4 - x_3) \\ &= \begin{Bmatrix} 8 \\ 10 \end{Bmatrix} + 2 \left[\begin{Bmatrix} 6 \\ 9 \end{Bmatrix} - \begin{Bmatrix} 8 \\ 10 \end{Bmatrix} \right] \\ &= \begin{Bmatrix} 4 \\ 8 \end{Bmatrix} \end{aligned}$$

$$f_5 = f(x_5) = 8$$

由于 $f_5 < f_4$, 故以 x_5 代替 x_1 , 由 $x_0 x_2 x_5$ 构成新单纯形, 进行下一循环。

经 32 次循环, 可将目标函数值降到 1×10^{-3} , 接近极小值 $f^* = f(x^*) = f(5, 6) = 0$ 。

单形替换法当问题维数 n 较高时, 需要经过很多次迭代, 因此一般用于 $n < 10$ 的情况。

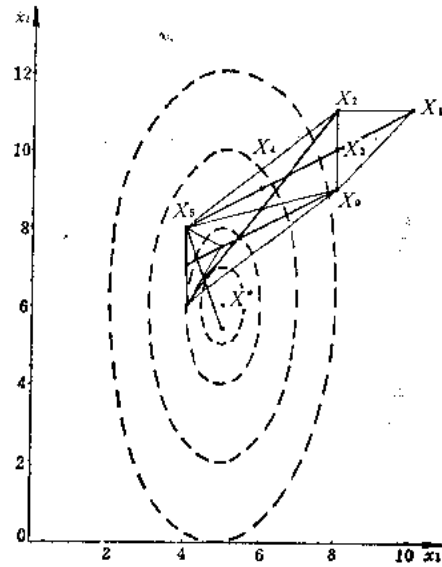


图 3-19

第四章 线性规划和几何规划方法简介

在第一章 §1 例 10 中，它的目标函数和约束条件都是线性的，像这类约束函数和目标函数都为线性函数的优化问题称作线性规划问题。它的解法在理论上和方法上都很成熟，实际应用也很广泛。虽然大多数工程设计是非线性的，但是也有采用线性逼近方法求解非线性问题的。此外，线性规划方法还常被用作解决优化问题的子问题的工具，如在可行方向法中可行方向的寻求就是采用线性规划方法。当然，对于真正的线性优化问题，线性规划方法就更有用了。

在求解线性规划问题时，如果原问题的约束数多于设计变量数，把它转化为对偶问题求解将是有益的。几何规划的传统解法是借助于算术——几何平均值不等式定理，把原问题化成它的对偶问题。在化成对偶问题这一点上，线性规划与几何规划有相似之处，所以本章结合线性规划的对偶问题，顺便对几何规划作一简单介绍。

§ 1 线性规划问题的标准形式

线性规划问题的标准形式是：

求 $x = (x_1, x_2, \dots, x_i, \dots, x_n)^T$

使目标函数

$$f(x) = \sum_{i=1}^n c_i x_i \rightarrow \min$$

要求满足约束条件

$$\begin{aligned} \sum_{i=1}^n a_{ji} x_i &= b_j & (j = 1, 2, \dots, m) \\ x_i &\geq 0 & (i = 1, 2, \dots, n) \end{aligned} \quad (4-1)$$

约束条件包括两部分：一是等式约束条件，二是变量的非负要求，它是标准形式中出现的唯一不等式形式。如果线性规划问题，除变量的非负要求外，还有其它不等式约束条件，可通过引入松弛变量将不等式约束化成上述等式约束形式。

例如，约束条件为

$$\begin{aligned} 2x_1 + x_2 &\leq 2 \\ x_1 &\geq 0, \quad x_2 \geq 0 \end{aligned}$$

通过引入松弛变量 $x_3 \geq 0$ 将第一个不等式约束条件化成等式形式：

$$\begin{aligned} 2x_1 + x_2 + x_3 &= 2 \\ x_1 &\geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0 \end{aligned}$$

这样，可行域就由二维空间 ABC 变成三维空间 $A'B'C'$ ，如图 (4-1) 所示。

如果在原来的问题中有一些变量并不要求是非负的，那么我们可以把它们写成两个

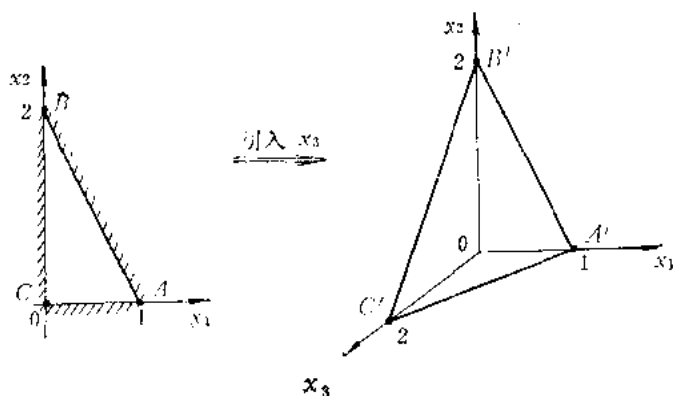


图 4-1

非负变量的差。例如

$$x_k = x_k' - x_k''$$

$$x_k' \geq 0, x_k'' \geq 0$$

当把松弛变量以及新的非负变量引入后，再将所有的变量重新编序，原来的线性规划问题就变成 (4-1) 式标准形式。注意，引入的松弛变量通常在目标函数中并不出现，而新的非负变量一般将出现在目标函数中。

线性规划问题的标准形式可写成如下的矩阵形式：

$$\begin{aligned} \text{求 } x \text{ 使} \quad & f(x) = c^T x \rightarrow \min \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{4-1a}$$

其中 $A \equiv [a_{ij}]_{m \times n}$, $b \equiv \{b_j\}$, $c \equiv \{c_j\}$, 而 0 代表零向量。

在所有有实际意义的线性规划问题中，总有 $m < n$ 。因为如果 $m = n$ ，从 (4-1) 方程组中唯一决定 x ，即方程组 $Ax = b$ 只有唯一解，没有可供选择的 x ，这样也就不存在所谓最优化问题。如果 $m > n$ ，方程组 $Ax = b$ 变成矛盾方程组，不存在严格满足方程组的解。所以只有当 $m < n$ 时，方程组 $Ax = b$ 的解才是不定的，一般将有无穷多个解，我们就可以从中找出使目标函数 $f(x)$ 取最小值的解来。

为了今后讨论问题方便起见，我们先给出几个定义。

若满足 $Ax = b$ 的任意 n 维的解向量 $x = (x_1, x_2, \dots, x_n)$ 中有 $n - m$ 个分量 $x_j (j = m + 1, \dots, n)$ 为零，而余下的是不超过 m 个非零的解向量 $x_j (j = 1, 2, \dots, m)$ ，则这组解称为 $Ax = b$ 的基本解。即在基本解中最多只能有 m 个非零分量。当使某一组 $n - m$ 个分量取零值时，就可从方程组 $Ax = b$ 中求得一个解向量。所以基本解的个数就等于从 n 中取 $n - m$ 的组合数 C_n^{n-m}

$$C_n^{n-m} = C_n^m = \frac{n!}{m!(n-m)!}$$

在基本解中若不超过 m 个的非零分量都为非负，称这样的基本解为基本可行解。基本可行解同时满足方程组 $Ax = b$ 和不等式 $x \geq 0$ 。基本可行解都处于线性规划问题可行域的凸多面体的各顶点上。以图 (4-2) 所示的线性规划问题为例，可行域由四条直

线围成, 基本可行解处在四边形的四个顶点 A, B, C, D 上。该问题有四个不等式约束条件, 即 $m=4$, 化成标准形式需引入四个松弛变量, 因此原来的二维问题变成六维问题, 即 $n=6$ 。基本解的个数应为 $C_6^4=15$, 此 15 个基本解可在图中表示出来。因为基本解应至少有 $n-m=6-4=2$ 个分量为零, x 的前两个分量 (原来的分量 x_1, x_2) 分别为零表示图中的两个坐标轴, 后四个分量 (松弛变量) 分别为零表示四条约束边界线。基本解要求其中有两个分量为零, 即基本解处在上述两类直线的交点上。所以基本解的个数也就是图 (4-2) 所示的每两条直线交点的个数, 共 15 个。(四条约束边界线分别和坐标轴 x_1, x_2 相交有 8 个交点, 四条约束边界线两两相交有 $C_4^2=6$ 个交点, 两个坐标轴相交有一个交点, 共 15 个交点)。但只有可行域 (图中阴影部分) 的四个顶点 A, B, C, D 才是基本可行解。因此基本可行解比基本解要少得多。

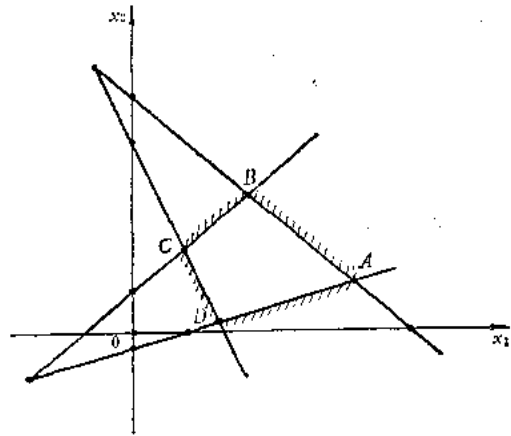


图 4-2

基本解的组数有 $C_n^m = \frac{n!}{(n-m)!m!}$

个。当然, 可行解的组数要比基本解的组数少得多。但是, 当 m 和 n 很大时, 可行解的组数仍然是很多的。因此, 需要有一个合适的计算方法, 从多组可行解中挑出一组来, 使目标函数 $f(x)$ 值为极小值。这组使目标函数为极小值的可行解称为优化解。

若可行域这个凸多面体有解, 一般说来目标函数将在该多面体的某一顶点取得最小值, 即最优点处于多面体的某一顶点上。(特殊情况可能最优点处于多面体的某一棱上或某一平面上。) 为了寻求最优点, 需要从凸多面体上的某一顶点找到另一个顶点, 即从某一个基本可行解转换到另一个基本可行解。

§ 2 基本可行解的转换

一、从一个基本解转到另一个基本解

单纯形方法是一种获得可行解, 并能从中确定优化解的很有效的方法。

为了理解单纯形方法, 先说明如何从 $Ax=b$ 中算出基本解, 又如何从一组基本解转到另一组基本解。

把约束条件的线性方程组 $Ax=b$ 写成展开的形式

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots & \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned} \tag{4-2}$$

从方程组 (4-2) 中并不能明显地看出它的一组解来, 但是如果对这个方程组进行一系列的初等变换, 就可以从中找到一组基本解。如选定某个系数 a_{1K} 作为主元, 采用高

$$\begin{aligned} 5x_1 - 4x_2 + 13x_3 - 2x_4 + x_5 &= 20 \\ x_1 - x_2 + 5x_3 - x_4 + x_5 &= 8 \end{aligned}$$

当顺次用 a_{11} 和 a_{22} 为轴元素时, 则得

$$\begin{aligned} x_1 + 0 - 7x_3 + 2x_4 - 3x_5 &= -12 \\ 0 + x_2 - 12x_3 + 3x_4 - 4x_5 &= -20 \end{aligned}$$

从而得一组基本解

$$x_1 = -12, x_2 = -20, x_3 = x_4 = x_5 = 0$$

因为 x_1 和 x_2 皆为负值, 所以它不是可行解。

如果在上述基础上, 再用 $a_{25} = -4$ 为轴元素, 则得正则方程组

$$\begin{aligned} x_1 - \frac{3}{4}x_2 + 2x_3 - \frac{1}{4}x_4 + 0 &= 3 \\ 0 - \frac{1}{4}x_2 + 3x_3 - \frac{3}{4}x_4 + x_5 &= 5 \end{aligned}$$

得又一组基本解

$$x_1 = 3, x_5 = 5, x_2 = x_3 = x_4 = 0$$

因为此时的 x_1 和 x_5 都是正值, 所以这组基本解是可行解。和前一组基本解相比, 这里是把 x_2 从基本解中排除出去而把 x_5 引进基本解中来了。

如果把 $x_i = b_i'$ 这 m 个变量称为基本变量, 那么上面的变换就是把 x_2 从基本变量中去掉, 把 x_5 引进基本变量中来, 从而完成从一组基本解到另一组基本解的变换。

二、从一个基本可行解转到另一个基本可行解

要使变换后所得的基本解变成可行解, 还要研究这样的方法, 即如何使某个选定的变量 x_K ($K = m+1, \dots, n$) 进入基本变量, 来替换另一个现在还在基本变量中的 x_s ($s = 1, 2, \dots, m$), 原来的基本解会变成可行解。

应看出, 当已经得到一组可行解, 即现在所有的 b_i' 都是非负时, 若要求把 x_K 选进基本变量的下一组基本解是可行解的话, 则在第 K 列所有系数中不能取任何负值的 a'_{iK} 作为轴元素。因为这样将使 b_i' 为负值, 结果对应的 x_K 必将是负的, 它就不是可行解的一个元素。

因此, 第一个要求是, 若 b_i' 都是非负的, 则必须 $a'_{iK} > 0$ 才可选做轴元素进行转轴运算, 以便用 x_K 去代替 x_i 。这个过程是: 反复进行转轴运算, 直到 x_i 从某个正值变成 0, 而 x_K 则从 0 变成某个正值时为止。

假定经过转轴运算后, 得到新的正则形式的方程组是

$$\left(\begin{array}{cccc|cccc} 1 & 0 & \dots & 0 & \dots & a'_{1K} & \dots & \dots \\ 0 & 1 & \dots & 0 & \dots & a'_{2K} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & \dots & a'_{iK} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & \dots & \dots & 1 & \dots \end{array} \right) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_K \end{pmatrix} = \begin{pmatrix} b_1' \\ b_2' \\ \vdots \\ b_i' \\ \vdots \\ \theta \end{pmatrix} \quad (4-4)$$

其中 θ 是正值, $K > m$ 。

式 (4-4) 给出一组解

$$\begin{aligned} x_K &= \theta \\ x_1 &= b'_1 - a'_{1K}x_K = b'_1 - a'_{1K}\theta \\ x_2 &= b'_2 - a'_{2K}x_K = b'_2 - a'_{2K}\theta \\ &\dots\dots\dots \\ x_i &= b'_i - a'_{iK}x_K = b'_i - a'_{iK}\theta \end{aligned} \quad (4-5)$$

如果式 (4-5) 是可行解, 且 $x_K = \theta > 0$ 又是其中的一个基本变量, 则在 x_1, x_2, \dots, x_m 中必然有一个 (假定它是 $x_s (s \leq m)$) 是零, 其余皆为正。当然这个变量 x_s 就应从基本变量中排除出去。这就是说, 只有取式 (4-5) 中各差值的最小者为零时, 才能保证使其余各差值皆为正。所以, 由条件

$$\min(b'_i - a'_{iK}\theta) = 0$$

可知, 只有保证

$$\min \left[\frac{b'_i}{a'_{iK}} \right] = \frac{b'_s}{a'_{sK}} = \theta = x_K \quad (4-6)$$

才能使 x_K 进入可行解的基本变量中去, 并把 x_s 从可行解的基本变量中排除出去。

同时对非负的 b'_i 又有 $a'_{iK} > 0$ 的要求。

式 (4-6) 中的 a'_{sK} 就是进行转轴运算时应取的轴元素。

这是一个规则, 称为 θ 规则。它说明: 若想用 x_K 取代 x_s 成为可行解中的基本变量, 就应选 $b'_s - a'_{sK}\theta = 0$ (其余的仍为非负) 所对应的第 s 行为转轴行。即所选的行要满足条件

$$\begin{aligned} a'_{iK} &> 0 \\ \theta &= \min \left[\frac{b'_i}{a'_{iK}} \right] = \frac{b'_s}{a'_{sK}} = x_K \end{aligned}$$

例如, 上例中

$$\begin{aligned} x_1 - \frac{3}{4}x_2 + 2x_3 - \frac{1}{4}x_4 + 0 &= 3 \\ 0 - \frac{1}{4}x_2 + 3x_3 - \frac{3}{4}x_4 + x_5 &= 5 \end{aligned}$$

已得可行解: $x_1 = 3, x_5 = 5, x_2 = x_3 = x_4 = 0$ 。此时的基本变量是 x_1 和 x_5 , 非基本变量是 x_2, x_3 和 x_4 。

由于 $b_1 = 3, b_2 = 5$, 是非负的。而 x_2 和 x_4 的系数又全是负的, 所以不能用 x_2 或 x_4 来取代 x_1 或 x_5 。但是, 如果 x_3 的系数是正值, 则可取 x_3 所在的第 3 列为转轴列。

考虑到 $\frac{3}{2}$ 比 $\frac{5}{3}$ 小, 则取第一行为转轴行。于是取 $a'_{13} = 2$ 为轴元素, 使 x_3 取代 x_1 成为基本变量。经过转轴运算, 得

$$\begin{aligned} \frac{1}{2}x_1 - \frac{3}{8}x_2 + x_3 - \frac{1}{8}x_4 + 0 &= \frac{3}{2} \\ -\frac{3}{2}x_1 - \frac{7}{8}x_2 + 0 - \frac{3}{8}x_4 + x_5 &= \frac{1}{2} \end{aligned}$$

得可行解, $x_3 = \frac{3}{2}, x_5 = \frac{1}{2}, x_1 = x_2 = x_4 = 0$

如果取第二行为转轴行, $a'_{23} = 3$ 为轴元素, 则解为: $x_1 = -\frac{1}{3}, x_3 = \frac{5}{3}, x_2 = x_4 = x_5 = 0$ 。它不是可行解。

对于这个具体例子, 我们先后算出了四组基本解:

$$x_1 = -12, x_2 = -20, x_3 = x_4 = x_5 = 0$$

$$x_1 = 3, x_5 = 5, x_2 = x_3 = x_4 = 0$$

$$x_3 = \frac{3}{2}, x_5 = \frac{1}{2}, x_1 = x_2 = x_4 = 0$$

$$x_1 = -\frac{1}{3}, x_3 = \frac{5}{3}, x_2 = x_4 = x_5 = 0$$

它们虽然都是方程组的解, 但只有两组是可行解。

三、初始基本可行解的求法

上面已讨论了如何从一个基本可行解转换到另一个基本可行解的算法, 那么最初的基本可行解如何求得呢?

当用添加松弛变量的方法把不等式约束转换成等式约束时, 我们往往会发现这些松弛变量就可以作为初始基本可行解中的一部分基本变量。例如, 假若约束条件为

$$x_1 - x_2 + x_3 \leq 5$$

$$x_1 + 2x_2 - x_3 \leq 10$$

$$x_i \geq 0, i = 1, 2, 3$$

引入松弛变量 x_4, x_5 可将前两个不等式约束条件转换成等式形式:

$$x_1 - x_2 + x_3 + x_4 + 0 = 5$$

$$x_1 + 2x_2 - x_3 + 0 + x_5 = 10$$

$$x_i \geq 0, i = 1, 2, 3, 4, 5$$

于是, 我们立即得到一组基本可行解

$$x_4 = 5, x_5 = 10, x_1 = x_2 = x_3 = 0$$

但是, 如果不等式约束条件右端项 b_i 是负的, 它所对应的松弛变量就不能作为基本可行解的基本变量, 所以上述方法并不是总能成功的。这时需引入人工变量, 经过变换再将它从基本变量中替换出去, 其具体作法将在 §4 单纯形法举例中予以介绍。

§ 3 单纯形方法

前面阐述了应用 θ 规则所规定的条件, 可以做到从一组基本可行解转换到另一组可行解。但哪一组可行解是优化解呢? 当然可以用各组可行解分别代入目标函数 $f(x)$, 取使 $f(x) \rightarrow \min$ (或 \max) 者为优化解。但是, 通过下面的分析, 可以找出确定优化解的规则。

对于可行解 (若由前 m 个变量组成可行解的基本变量时), 目标函数可以写成

$$f(x) = \sum_{i=1}^m c_i b'_i = c_1 b'_1 + c_2 b'_2 + \dots + c_m b'_m + 0 + \dots + 0$$

如果还有另一组可行解，它的基本变量中包含有 $x_K = \theta (K > m)$ ，即

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_1 \\ \vdots \\ x_K \end{pmatrix} = \begin{pmatrix} b_1' - a'_{1K}\theta \\ b_2' - a'_{2K}\theta \\ \dots \\ b_1' - a'_{1K}\theta \\ \dots \\ \theta \end{pmatrix}$$

其中的 $x_s = b_s' - a'_{sK}\theta = 0 \quad (s \leq m)$ 。

它所对应的目标函数值是

$$\begin{aligned} \bar{f}(x) &= c_1(b_1' - a'_{1K}\theta) + c_2(b_2' - a'_{2K}\theta) + \dots + c_m(b_m' - a'_{mK}\theta) \\ &\quad + 0 + \dots + c_K\theta + 0 + \dots + 0 = \sum_{i=1}^m c_i b_i' - \sum_{i=1}^m c_i a'_{iK}\theta + c_K\theta \end{aligned}$$

令 $f(a_K) = \sum_{i=1}^m c_i a'_{iK}$,

则 $\bar{f}(x) = f(x) + [c_K - f(a_K)]\theta = f(x) + r\theta$

式中 $r = c_K - f(a_K)$ 称为相对价值系数。

显然，对极小化问题，应要求 $\bar{f}(x) < f(x)$ ，即 $r\theta$ 应是负值。由于 θ 是正值，则就应要求 $r = c_K - f(a_K)$ 为负值。只要它仍是负值，则目标函数 $f(x)$ 还没有达到极小值，还有下降的趋势，就还可以进行转轴运算，选取另一组可行解。因此，一俟 $r = c_K - f(a_K)$ 为正，即可停止转轴运算。对应的可行解就是优化解。

可能有几组 $c_K - f(a_K)$ 都为负值。对极小化问题应取

$$\min [c_j - f(a_j)] = c_K - f(a_K) \quad (4-7)$$

这又是一个规则，称为最速变化规则（即目标函数值变化最大规则）。

上面的方法是利用约束条件方程组解出可行解，再用目标函数检验优化解的方法。

计算时，也可以直接把目标函数和约束条件同时列为转轴运算方程组。采用边计算可行解，边校验目标函数值的变化情况的办法来求优化解。这时，对于极小化问题，只要

$$f(x) = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$$

式中的系数 c_K 有一个或几个是负值时，就说明 $f(x)$ 值还可以减小，就应把对应于 $c_K = \min_j [c_j]$ 的变量 x_K 选进可行解的基本变量中去。

一个 θ 规则，即

$$x_K = \theta = \min \left[\frac{b_i'}{a'_{iK}} \right] = \frac{b_s'}{a'_{sK}}$$

$$a'_{iK} > 0$$

一个最速变化规则，即

$$\min_j [c_j - f(a_j)] = c_K - f(a_K)$$

或

$$\min_j [c_j] = \theta_K$$

就构成单纯形方法的基础。

对于极大值问题，则最速变化规则应取 \max 号。

上述单纯形方法的整个运算过程可以用框图表示，如图 (4-3) 所示。

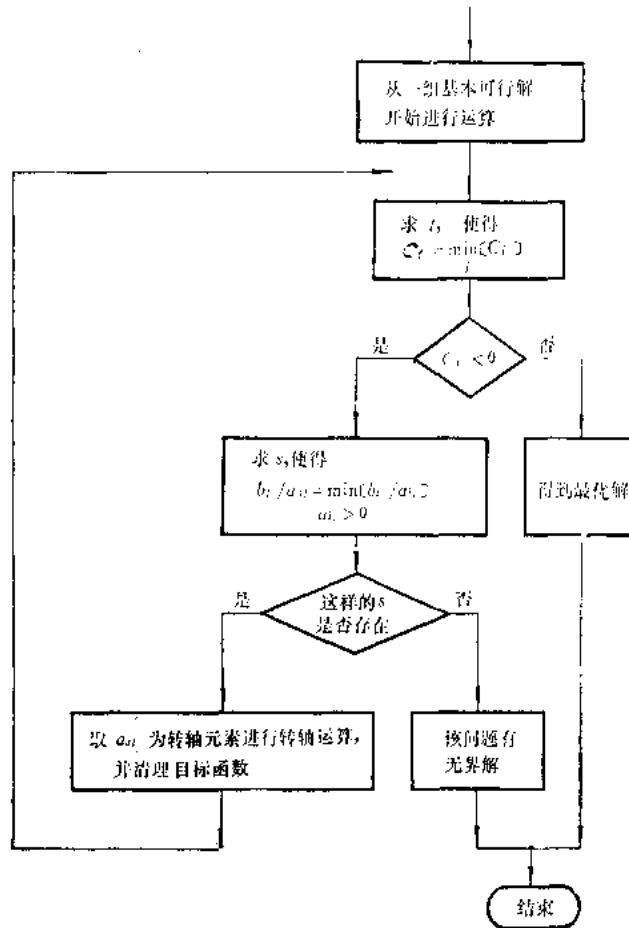


图 4-3

为了便于用列表方式进行单纯形法的计算，现采用矩阵形式写出单纯形方法的运算过程。

把约束条件方程组的系数矩阵 A 分块

$$A = [E : F]$$

相应地把 x 也分块写成

$$x = [x_E : x_F]^T$$

则目标函数值为

$$f(x) = c^T x = c_E x_E + c_F x_F$$

因而标准的线性规划问题就化为

$$\min \quad c_E x_E + c_F x_F$$

$$\text{s.t.} \quad E x_E + F x_F = b$$

(4-8)

$$x \geq 0$$

对应于 E 的解是 x_E , 若此时有 $x_F = 0$, 则得一组可行解是

$$x = [x_E; 0] \quad \text{而} \quad x_E = E^{-1}b$$

如果有任何一组含有 $x_F \neq 0$ 的可行解, 则由

$$Ex_E + Fx_F = b$$

有

$$x_E = E^{-1}b - E^{-1}Fx_F$$

把它代入目标函数

$$c_E x_E + c_F x_F$$

中去, 得

$$c_E(E^{-1}b - E^{-1}Fx_F) + c_F x_F = c_E E^{-1}b + (c_F - c_E E^{-1}F)x_F$$

此式是用 x_F 值来表达任一组可行解时目标函数 $c_E x_E + c_F x_F$ 的值。式中的

$$c_F - c_E E^{-1}F = r$$

是相对价值系数。

若讨论的是极小化问题, 则当 r 含负值分量时, 目标函数还有变化的可能, 还要进行转轴运算。因此, 对应的那个变量应进入基 E 中来。即通过 r 向量可以确定哪一个变量应进入基 E 中来。

导出价值系数向量 r 的表达式后, 就能够用矩阵形式写出单纯形法计算的表 (简称单纯形表)。它的初始形式 (把目标函数和约束方程组同时列出) 是

$$\left(\begin{array}{c|c} A & b \\ \hline c & 0 \end{array} \right) \xrightarrow{\text{分块}} \left(\begin{array}{c|c|c} E & F & b \\ \hline c_E & c_F & 0 \end{array} \right) \quad (4-9)$$

进行初等变换, 得相应的单纯形表 (正则形矩阵)

$$T = \left(\begin{array}{c|c|c} I & E^{-1}F & E^{-1}b \\ \hline 0 & c_F - c_E E^{-1}F & -c_E E^{-1}b \end{array} \right) \quad (4-10)$$

这就使原矩阵式 (4-9) 中的 E 转变为单位阵, 给出一组可行解; 同时计算出目标函数中的相对价值系数 $r = c_F - c_E E^{-1}F$ 的值。

上述运算可以利用 $E = L^T D L$ 或 $E = LU$ 的分解法进行。其步骤如下。

1. 取 $E y_0 = b$, 计算满足 $E y_0 = b$ 的现行解 $x_E = y_0$ 。
2. 解 $\lambda E = c_E$, 得 $\lambda = c_E E^{-1}$ 。并令 $r = c_F - c_E E^{-1}F = c_F - \lambda F$ 。若 $r \geq 0$ 就停止计算, 现行解就是优化解。
3. 按最负的 r , 选其对应的向量 $\{a_j\}$ 进基 ($\{a_j\}$ 是 A 中的第 j 列元素列阵)。并求解 $E y_j = \{a_j\}$ 。
4. 计算比值 $\frac{y_{i0}}{y_{ij}}$, 并选最小非负的一个确定离基向量。
5. 取新的 E , 返回步骤 1。

可以看出, 在这个运算过程中, 主要是反复计算如下的线性方程组:

$$E y_0 = b; \quad \lambda E = c_E; \quad E y_j = \{a_j\}$$

这种方法是利用对 E^{-1} 的求解技术进行计算的。可以在有关资料中找到现成的程序。

§ 4 单纯形法应用举例

下面用第一章 §1 中例 10 为例，说明单纯形方法的具体算法。

对于此例，可以通过引入“松弛变量” x_4, x_5, x_6, x_7, x_8 的方法把不等式约束写成等式约束的形式，则问题变成

$$f(x) = 2x_1 + 3x_2 + 4x_3 \rightarrow \max$$

s.t.

$$20x_1 + 25x_2 + 30x_3 + x_4 = 9000$$

$$x_1 + x_2 + x_3 - x_5 = 350$$

$$x_1 - 0.2x_6 + x_8 = 70$$

$$x_2 - 0.6x_6 + x_7 = 210$$

$$x_3 - 0.4x_6 + x_8 = 140$$

在引入松弛变量后，这个线性规划问题变成，求 $x = (x_1, x_2, x_3)$ 的一组值，使目标函数

$$f(x) = 2x_1 + 3x_2 + 4x_3$$

为最大值，其约束条件是

$$20x_1 + 25x_2 + 30x_3 + x_4 = 9000$$

$$x_1 + x_2 + x_3 - x_5 = 350$$

$$x_1 - 0.2x_6 + x_8 = 70$$

$$x_2 - 0.6x_6 + x_7 = 210$$

$$x_3 - 0.4x_6 + x_8 = 140$$

这里共有 5 个约束方程，却有 x_1, x_2, \dots, x_8 等 8 个未知数，其中的 x_4, x_5, x_6, x_7, x_8 是松弛变量。这就相当于在可能的 8 个变量中每次同时取出 5 个来进行组合，计有 $\frac{8!}{3!5!} = 56$ 种可能的不同组合。从中要选出能获得最大利润（目标函数值最大）的

一种组合，就是所求的优化解。

可取系数为 1 的 x_4, x_5, x_6, x_7, x_8 这 5 个松弛变量做为初始基本解的基本变量。但是由于 x_5 的系数是 -1，不是正值，而对应的 b 是正值，所以 x_5 是不能进入可行解的基本变量的。因此，再引入一个 x_9 ，即约束方程组变为

$$20x_1 + 25x_2 + 30x_3 + x_4 = 9000$$

$$x_1 + x_2 + x_3 - x_5 + x_9 = 350$$

$$x_1 - 0.2x_6 + x_8 = 70$$

$$x_2 - 0.6x_6 + x_7 = 210$$

$$x_3 - 0.4x_6 + x_8 = 140$$

由此引起一个问题，就是要保证最后能把 x_9 从优化解中排除出去。为了做到这一

点, 可以给 x_0 一个很大的系数 c_0 , 对于极大值问题它取负值 (对于极小值问题它应取正值)。而只要 $f(x)$ 还没有达到极值, 运算过程还可以继续进行下去。在给 x_0 一个大值的系数 c_0 后, 目标函数中将增加 c_0x_0 一项。因此, 只要 x_0 还不是零, 目标函数就没有达到极大值。

这样, 本线性规划问题是 (取 $c_3 = -1000$) :

$$\max f(x) = 2x_1 + 3x_2 + 4x_3 - 1000x_0$$

s. t.

$$20x_1 + 25x_2 + 30x_3 + x_4 = 9000$$

$$x_1 + x_2 + x_3 - x_5 + x_6 = 350$$

$$x_1 - 0.2x_6 + x_7 = 70$$

$$x_2 - 0.6x_6 + x_8 = 210$$

$$x_3 - 0.4x_6 + x_9 = 140$$

令 $x_1 = x_2 = x_3 = x_5 = 0$, 则得 $x_4 = 9000$, $x_6 = 70$, $x_7 = 210$, $x_8 = 140$, $x_9 = 350$ 。它是一组可行解。

以这组可行解为出发点用单纯形表进行运算。表 (4—1) 是它的初始形式。下面先说明作表方法。

表 4—1

$c_i \rightarrow$	解	2	3	4	0	0	0	0	0	-1000	0	-991	θ_i
$c_i \downarrow$		p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	B	校核	
0	x_4	20	25	30	1	0	0	0	0	0	9000	9076	300
-1000	x_7	1	1	1	0	-1	0	0	0	1	350	353	350
0	x_6	1	0	0	0	-0.2	1	0	0	0	70	71.8	
0	x_7	0	1	0	0	-0.6	0	1	0	0	210	211.4	
0	x_8	0	0	①	0	-0.4	0	0	1	0	140	141.6	140
	$f(a_j)$	-1000	-1000	-1000	0	1000	0	0	0	-1000	-350000	-353000	
	$c_j - f(a_j)$	1002	1003	1004	0	-1000	0	0	0	0	350000	352009	

↑

1. 第一列的数值是可行解中各基本变量在目标函数 $f(x)$ 中的对应系数 c_i 的值。
2. 第二列是“解”列。其中列出进入本次可行解中的基本变量。哪一个变量应进入, 是根据在形成正则方程组的单位矩阵中元素“1”所在行的位置确定的。
3. p_i 的各列代表对应于约束方程中各变量 x_j 的系数 a_{ij} 的列向量。这里 p_1, p_2 和 p_3 是变量 x_1, x_2 和 x_3 的系数列; 从 p_4 到 p_9 是松弛变量 x_4 到 x_9 的系数列。
4. B 列是规定值列向量。
5. “校核”列用于核实其它列中的计算。这一列的值是其它列中对应元素值的

和。

6. 最后一列是当本次运算完成时, 记录 θ_i 值用的。

7. 最后两行是记录 $f(\mathbf{a}_j)$ 和 $c_j - f(\mathbf{a}_j)$ 值用的。

8. 表上和 c_j 所对应的一行, 标出在目标函数 $f(\mathbf{x})$ 中和每个变量 x_j 相对应的系数 c_j 的值。

这里是取各松弛变量 (其数量和约束不等式个数相同。对于等式约束, 采用加一个松弛变量再减这个松弛变量的办法。但须对两者加以区分) 为初始可行解的基本变量。这在表 (4-1) 中是容易实现的。因为表中的一行就是一个对应的约束方程, 所以只要把 5 个约束方程的相应系数 a_{ij} 填入表中对应位置即可。

例如, 方程 $20x_1 + 25x_2 + 30x_3 + x_4 = 9000$, 就是表中第一行。所以只要在第一行的 p_1 列处填系数 20, p_2 列处填 25, p_3 列处填 30, p_4 列处填 1, 从 p_5 到 p_9 各列处都填 0; 最后在 B 列处填 9000 就可以了。

同样, 对其它 4 个约束方程, 分别在第二至第五行的 p_j 列处填上对应于 x_j 的系数即可。

然后, 再把校核列填上。例如, 第一行的校核值是 $20 + 25 + 30 + 1 + 0 + 0 + 0 + 0 + 0 + 9000 = 9076$ 。

由于单位矩阵是由 p_4, p_6, p_7, p_8 和 p_9 这 5 个列向量组成的, 所以初始可行解的基本变量是 x_4, x_6, x_7, x_8, x_9 。因此, 在各个 p_j 列中出现“1”元素的行上, 就在“解”列内该行处记下变量 x_j 。例如, 因为 p_4 列中的元素“1”出现在第一行, 所以在“解”列中的第一行记 x_4 ; p_6 列中的元素“1”出现在第二行, 所以“解”列中第二行记 x_6 ; 等等。

和“解”列中的各个变量相对应的目标函数 $f(\mathbf{x})$ 的系数应按次序列入 c_i 列中 (c_i 和 x_i 的序号应是一致的)。

因为 $f(\mathbf{a}_j) = c_1 a_{1j} + c_2 a_{2j} + \dots + c_n a_{nj}$, 则 $f(\mathbf{a}_j)$ 行就是在 c_i 列中取每个元素乘以 p_j 列中的对应元素 a_{ij} , 再相加的结果。例如 $f(\mathbf{a}_1) = 0 \times 20 + (-1000) \times 1 + 0 \times 1 + 0 \times 0 + 0 \times 0 = -1000$ 。

对应于 B 列中的 $f(\mathbf{a}_j)$ 实际上就是这次运算的目标函数值, 因为 B 列中的元素就是本次运算的可行解中的基本变量 x_j 的值。

$c_j - f(\mathbf{a}_j)$ 行的值就是表上方的 c_j 行中的值减去相应的 $f(\mathbf{a}_j)$ 的值。如 $c_1 - f(\mathbf{a}_1) = 2 - (-1000) = 1002$ 等。

根据最速变化规则, 在下一组可行解的基本变量中, 应进入的变量应是所有 $c_j - f(\mathbf{a}_j)$ 中最大者那一列的变量。现在表 (4-1) 中是 $c_3 - f(\mathbf{a}_3) = 1004$ 最大, 所以 x_3 应进入下一组可行解的基本变量中去。这在表 (4-1) 的下边用箭头标出了, 同时又把 p_3 列框起来了。

既然 x_3 要进入可行解的基本变量中, 那么就应从上次可行解基本变量中排除出一个变量。由 θ 规则 $\theta = \min \left[\frac{b_i'}{a_{iK}'} \right]$ 和 $a_{iK}' > 0$ (这里 $K=3$ 已确定了) 可知, 需要分别计算各行的 θ_i 值并写在 θ_i 列中。

取 \mathbf{B} 列中的各元素 b_i' ，分别用 p_3 列中对应元素 a'_{i3} 除之，得： $\theta_1 = \frac{b_1'}{a'_{13}} = \frac{9000}{30} = 300$ ， $\theta_2 = \frac{b_2'}{a'_{23}} = 350$ 。因为 $a'_{33} = a'_{43} = 0$ ，所以 θ_3 和 θ_4 处空着。最后的 $\theta_5 = \frac{b_5'}{a'_{53}} = 140$

这时 θ_5 最小，所以它所对应的 x_3 就是应排除出基本变量的变量。这在表 (4-1) 的右边也用箭头标出了。因此，下一组可行解中的基本变量将是 x_4, x_5, x_6, x_7, x_8 。在表 (4-1) 中把位于 x_3 所在的行和 p_3 列交点处的元素用圆圈上，称它是轴元素。它就是 a'_{iK} ，这里是 a'_{53} 。

为了保证用 x_3 取代 x_5 进入下一组可行解基本变量中，就要对 p_3 进行初等变换，使除轴元素 $a'_{iK} = a'_{53} = 1$ 外，其余各元素 $a'_{iK} (i \neq 5)$ 全为零。这样，就计算出从 p_1 列直到 \mathbf{B} 列中各元素的值，如表 (4-2) 所示。

表 4-2

$c_j \rightarrow$	解	2	3	4	0	0	0	0	0	-1000	0	-991	θ_i
$c_i \downarrow$		p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	\mathbf{B}	校核	
0	x_4	20	25	0	1	12	0	0	-30	0	4800	4828	192
-1000	x_5	1	1	0	0	-0.6	0	0	-1	1	210	211.4	210
0	x_6	1	0	0	0	-0.2	1	0	0	0	79	71.8	
0	x_7	0	1	0	0	-0.6	0	1	0	0	210	211.4	210
4	x_8	0	0	1	0	-0.4	0	0	1	0	140	141.6	
	$f(a_j)$	-1000	-1000	4	0	598.4	0	0	1004	-1000	-209440	-210833.6	
$c_j - f(a_j)$		1002	1003	0	0	-598.4	0	0	-1004	0	209440	209842.6	

表 (4-2) 是从表 (4-1) 变换来的。由于表 (4-2) 中已把 x_3 列为可行解的基本变量，所以在 c_i 列和 x_3 对应的系数换成 x_3 的系数 $c_3=4$ 。这样， p_3 列中的 $f(a_3)$ 值变成 4。

p_5 列中的 $f(a_5)$ 值变成 $-1000 \times (-0.6) + 4 \times (-0.4) = 598.4$ 。 p_9 列中的 $f(a_9)$ 值变成 $-1000 \times (-1) + 4 = 1004$ 。

\mathbf{B} 列中的 $f(a_i) = -1000 \times 210 + 4 \times 140 = -209440$ 。

由表 (4-2) 得轴元素为 $a'_{iK} = a'_{53} = 25$ 。所以得

$$\frac{a'_{11}}{a'_{12}} = \frac{20}{25} = 0.8 \quad \frac{a'_{12}}{a'_{12}} = \frac{25}{25} = 1 \quad \frac{a'_{13}}{a'_{12}} = \frac{0}{25} = 0 \quad \frac{a'_{14}}{a'_{12}} = \frac{1}{25} = 0.04$$

$$\frac{a'_{15}}{a'_{12}} = \frac{12}{25} = 0.48 \quad \frac{a'_{16}}{a'_{12}} = \frac{a'_{17}}{a'_{12}} = 0 \quad \frac{a'_{18}}{a'_{12}} = \frac{-30}{25} = -1.2 \quad \frac{a'_{19}}{a'_{12}} = 0$$

其结果见表 (4-3)。

p_1 列中除基轴行处元素 $\frac{a'_{11}}{a'_{12}} = \frac{20}{25} = 0.8$ 外, 其余各元素分别是 0.2, 1, -0.8,

0。其余各列中各元素的值如表 (4-3) 所示。

表 4-3

$c_j \rightarrow$	解	2	3	4	0	0	0	0	0	-1000	0	-991	θ_j
$c_i \downarrow$		p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	B	校核	
3	x_2	0.8	1	0	0.04	0.48	0	0	-1.2	0	192	193.12	240
-1000	x_9	0.2	0	0	-0.04	-1.08	0	0	0.2	1	18	18.26	90
0	x_6	①	0	0	0	-0.2	1	0	0	0	70	71.80	70
0	x_7	-0.8	0	0	-0.04	-1.08	0	1	1.2	0	18	18.28	
4	x_3	0	0	1	0	-0.4	0	0	1	0	140	141.60	
	$f(a_j)$	-197.6	3	4	40.12	1079.84	0	0	-199.6	-1000	-16864	-17134.34	
c_j	$-f(a_j)$	199.6	0	0	-40.12	-1079.84	0	0	199.6	0	16864	16143.24	

↑

用同样方法可以作出表 (4-4) 和表 (4-5)。

表 4-4

$c_j \rightarrow$	解	2	3	4	0	0	0	0	0	-1000	0	-991	θ_j
$c_i \downarrow$		p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	B	校核	
3	x_2	0	1	0	0.04	0.64	-0.8	0	-1.2	0	136	135.68	
-1000	x_9	0	0	0	-0.04	-1.04	-0.2	0	(0.2)	1	4	3.92	20
2	x_1	1	0	0	0	-0.2	1	0	0	0	70	71.80	
0	x_7	0	0	0	-0.04	-1.24	0.8	1	1.2	0	74	75.72	61.66
4	x_3	0	0	1	0	-0.4	0	0	1	0	140	141.60	140
	$f(a_j)$	2	3	4	40.12	1039.92	199.6	0	-199.6	-1000	-2892	-2802.96	
c_j	$-f(a_j)$	0	0	0	-40.12	-1039.92	-199.6	0	199.6	0	2892	1811.96	

↑

表 4—5

$c_j \rightarrow$	解	2	3	4	0	0	0	0	0	-1000	0	-991	θ_j
$c_i \downarrow$		p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	B	校核	
3	x_2	0	1	0	-0.2	-5.6	-2.0	0	0	6	160	159.2	
0	x_3	0	0	0	-0.2	-5.2	-1	0	1	5	20	19.6	
2	x_1	1	0	0	0	-0.2	1	0	0	0	70	71.8	
0	x_7	0	0	0	0.2	5.0	2.0	1	0	-6	50	52.2	
4	x_3	0	0	1	0.2	4.8	1	0	0	-5	120	122.0	
	$f(a_j)$	2	3	4	0.2	2.0	0	0	0	-2	1100	1109.2	
$c_j - f(a_j)$		0	0	0	-0.2	-2.0	0	0	0	-998	-1100	-2100.2	

当计算反复进行直到所有 $c_j - f(a_j) \leq 0$ 时，目标函数就达到最大值。现在表 (4—5) 中的 $c_j - f(a_j) \leq 0$ ，所以表 (4—5) 的结果给出优化解。

从表 (4—5) 直接读得优化解的基本变量为： $x_1 = 70$ ， $x_2 = 160$ ， $x_3 = 120$ ($x_7 = 50$ ， $x_8 = 20$ 不包含在目标函数 $f(x)$ 中)。

和优化解对应的目标函数值是 $f(x) = 1100$ 。

结果表明，2人宿舍应是70个单元，3人宿舍应是160个单元，4人宿舍应是120个单元。这时的总利润是1100000元。

下面进行约束条件的校核。

1. 总投资为 $1400000 + 4000000 + 3600000 = 9000000$ 是对的。
2. 宿舍单元数是 $70 + 160 + 120 = 350$ 也是对的。
3. 每类宿舍所占百分比的限制

$$x_1 = \frac{70}{350} \times 100 = 20\%$$

$$x_2 = \frac{160}{350} \times 100 = 45.7\%$$

$$x_3 = \frac{120}{350} \times 100 = 34.4\%$$

也是对的。

§ 5 最小二乘法

在实验数据处理时，多项式的拟合是经常遇到的问题。例如，由实验测得函数 $b = f(a)$ 在 p 个点上的值分别是 $b_i = f(a_i)$ ($i = 1, 2, \dots, p$)。

现在要求出该函数的一个近似表达式。最简单的是设所求的表达式为一个次数 n 低

从而 $\frac{\partial F}{\partial x_1} = 0$ 给出

$$\sum_{K=1}^p \sum_{j=1}^n a_{K1} a_{Kj} x_j = \sum_{K=1}^p a_{K1} b_K$$

对 x_2 求导, 有

$$\frac{\partial F}{\partial x_2} = 2 \sum_{K=1}^p (a_{K1} a_{K2} x_1 + a_{K2}^2 x_2 + a_{K2} a_{K3} x_3 + \dots - a_{K2} b_K)$$

或写成
$$\frac{\partial F}{\partial x_2} = 2 \left[\sum_{K=1}^p \sum_{j=1}^n a_{K2} a_{Kj} x_j - \sum_{K=1}^p a_{K2} b_K \right]$$

从而极值条件 $\frac{\partial F}{\partial x_2} = 0$ 给出

$$\sum_{K=1}^p \sum_{j=1}^n a_{K2} a_{Kj} x_j = \sum_{K=1}^p a_{K2} b_K$$

因此, 所得的 n 个方程组将是线性方程组

$$\sum_{K=1}^p \sum_{j=1}^n a_{Ki} a_{Kj} x_j = \sum_{K=1}^p a_{Ki} b_K \quad (i = 1, 2, \dots, n) \quad (4-14)$$

或简写成矩阵形式

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b} \quad (4-14a)$$

若记 $\bar{\mathbf{A}} = \mathbf{A}^T \mathbf{A}$, $\bar{\mathbf{b}} = \mathbf{A}^T \mathbf{b}$, 则有

$$\bar{\mathbf{A}} \mathbf{x} = \bar{\mathbf{b}} \quad (4-14b)$$

其中的 $\mathbf{A}^T \mathbf{A} = \bar{\mathbf{A}}$ 是对称正定矩阵。

从式 (4-14) 的 n 个方程式可以解出 n 个变量 x_j 。解的矩阵形式可自式 (4-14a) 得

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (4-15)$$

下面具体看一下 $\bar{\mathbf{A}} = \mathbf{A}^T \mathbf{A}$, $\bar{\mathbf{b}} = \mathbf{A}^T \mathbf{b}$ 矩阵的系数。

如果把式 (4-14) 的 n 个方程式中的系数 a_{Ki} , a_{Kj} 分别写成原始数据 a_K ($K = 1, 2, \dots, p$) 的幂的形式, 当 $i=1$ 时, 有

$$\begin{aligned} \sum_{K=1}^p b_K a_{K1} &= \sum_{K=1}^p (a_{K1} x_1 + a_{K2} x_2 + \dots + a_{Kn} x_n) a_{K1} \\ &= \sum_{K=1}^p (a_K x_1 + a_K^2 x_2 + \dots + a_K^n x_n) a_K \end{aligned}$$

或

$$\sum_{K=1}^p b_K a_K = \sum_{K=1}^p (a_K^2 x_1 + a_K^3 x_2 + \dots + a_K^{n+1} x_n)$$

当 $i=2$ 时, 有

$$\sum_{K=1}^p b_K a_{K2} = \sum_{K=1}^p (a_K x_1 + a_K^2 x_2 + \dots + a_K^n x_n) a_K^2$$

或

$$\sum_{k=1}^p b_k a_k^2 = \sum_{k=1}^p (a_k^3 x_1 + a_k^4 x_2 + \dots + a_k^{n+2} x_n)$$

当 $i=n$ 时, 有

$$\sum_{k=1}^p b_k a_k^n = \sum_{k=1}^p (a_k^{n+1} x_1 + a_k^{n+2} x_2 + \dots + a_k^{2n} x_n)$$

可见, 式 (4—14a) 或 (4—14b) 的系数矩阵 $\bar{A} = A^T A$ 和右端列阵 $\bar{b} = A^T b$ 中的第 i 行各元素将分别是

$$\begin{aligned} a_{i1} &= \sum_{k=1}^p a_k^{i+1}, & a_{i2} &= \sum_{k=1}^p a_k^{i+2} \dots \dots \\ a_{in} &= \sum_{k=1}^p a_k^{i+n}, & b_i &= \sum_{k=1}^p b_k a_k^i \end{aligned}$$

可以对矩阵 $\bar{A} = A^T A$ 利用线性代数中的方法进行分解来求解式 (4—15)。得 x 的值。

有些问题, 式 (4—15) 解得的 x 值不一定是精确解 (优化解)。这时可进行迭代计算。

和牛顿型法相似, 迭代时取 $-(A^T A)^{-1} A^T b$ 为搜索方向, 则有

$$x^{K+1} = x^K - (A_K^T A_K)^{-1} A_K^T b_K \quad (4-16)$$

为了简明, 把 A_K 写成 A , b_K 写成 b , 则有

$$x^{K+1} = x^K - (A^T A)^{-1} A^T b \quad (4-16a)$$

反复上述迭代运算, 直到获得满意的结果为止。

通常, 当 n 较大时, 用上述方法求出的点列并不一定收敛。可以有几种方法来克服这一困难。其一是像阻尼牛顿法那样, 取

$$d^K = -(A^T A)^{-1} A^T b$$

用迭代式

$$x^{K+1} = x^K + \alpha_K d^K = x^K - \alpha_K (A^T A)^{-1} A^T b \quad (4-17)$$

计算 x^{K+1} 。方法是从 x^K 出发沿 d^K 方向对 $F(x)$ 作一维搜索, 求出满足

$$F(x^K + \alpha_K d^K) = \min_{\alpha} F(x^K + \alpha_K d^K)$$

的 α_K 值, 从而按式 (4—17) 算出 x^{K+1} 来。和阻尼牛顿法一样, 这种方法称为阻尼最小二乘法。

§ 6 非线性规划问题的线性化解法——线性逼近法

线性规划问题是数学规划中提出较早的一类问题, 它的求解方法在理论和算法上也较成熟, 在实际工作中有比较广泛的应用。因此, 自然就会想到, 对一些非线性规划问题, 可否把非线性函数线性化, 再用线性规划方法求解? 回答当然是肯定的。这类方法如“割平面法”、“小步梯度法”等。下面仅做简单说明。

假设原非线性问题是:

$$\begin{aligned} & \min_x f(x) \\ \text{s.t.} & \quad h_i(x) = 0 \quad (i = 1, 2, \dots, l) \\ & \quad g_j(x) \leq 0 \quad (j = 1, 2, \dots, p) \quad (l + p = m) \end{aligned}$$

当给出某个较“合理”的初始点 x^0 (或 x^K , 不管它是否是可行点) 时, 可以在该点把函数作泰勒展开至一阶项, 得近似线性方程

$$f(x) = f(x^0) + \nabla f(x^0)^T(x - x^0) = f^0(x) \quad (4-18)$$

或 $f^K(x) = f(x^K) + \nabla f(x^K)^T(x - x^K) \quad (4-18a)$

和

$$h_i(x) = h_i(x^0) + \nabla h_i(x^0)^T(x - x^0) = h_i^0(x) = 0 \quad (4-19)$$

$$g_j(x) = g_j(x^0) + \nabla g_j(x^0)^T(x - x^0) = g_j^0(x) \leq 0 \quad (4-20)$$

或

$$h_i^K(x) = h_i(x^K) + \nabla h_i(x^K)^T(x - x^K) = 0 \quad (4-19a)$$

$$g_j^K(x) = g_j(x^K) + \nabla g_j(x^K)^T(x - x^K) \leq 0 \quad (4-20a)$$

由于 $f(x^K)$, $\nabla f(x^K)$, $h_i(x^K)$, $\nabla h_i(x^K)$, $g_j(x^K)$ 和 $\nabla g_j(x^K)$ 均为常向量, 则上述问题是一个线性规划问题。可采用线性规划方法求解, 但是属于线性逼近性质的方法。

例如, 非线性问题:

$$\begin{aligned} & \min f(x) = 4x_1 - x_2^2 - 12 \\ \text{s.t.} & \quad h_1(x) = 25 - x_1^2 - x_2^2 = 0 \\ & \quad g_1(x) = 10x_1 - x_1^2 + 10x_2 - x_2^2 - 34 \geq 0 \\ & \quad g_2(x) = x_1 \geq 0 \\ & \quad g_3(x) = x_2 \geq 0 \end{aligned}$$

当取初始点 $x^0 = (2, 4)$ 时, 则在点 x^0 处的线性近似函数可分别计算如下。

对于目标函数 $f(x) = 4x_1 - x_2^2 - 12$, 有 $f(x^0) = -20$ 。而 $\frac{\partial f}{\partial x_1} = 4$, $\frac{\partial f}{\partial x_2} = -2x_2$ 。

所以 $f^0(x) = f(x^0) + 4(x_1 - x_1^0) - 2x_2^0(x_2 - x_2^0) = -20 + 4(x_1 - 2) - 8(x_2 - 4)$, 即 $f^0(x) = 4x_1 - 8x_2 + 4$ 。

对于等式约束 $h_1(x) = 25 - x_1^2 - x_2^2$, 有 $h_1(x^0) = 5$ 。而 $\frac{\partial h_1}{\partial x_1} = -2x_1$, $\frac{\partial h_1}{\partial x_2} = -2x_2$ 。

所以 $h_1^0(x) = h_1(x^0) - 2x_1^0(x_1 - x_1^0) - 2x_2^0(x_2 - x_2^0) = 5 - 4(x_1 - 2) - 8(x_2 - 4)$, 即 $h_1^0(x) = 45 - 4x_1 - 8x_2 = 0$ 。

对于不等式约束 $g_1(x) = 10x_1 - x_1^2 + 10x_2 - x_2^2 - 34 \geq 0$, 有 $g_1(x^0) = 6$ 。而 $\frac{\partial g_1}{\partial x_1} = 10 - x_1$, $\frac{\partial g_1}{\partial x_2} = 10 - 2x_2$ 。所以 $g_1^0(x) = g_1(x^0) + (10 - 2x_1^0)(x_1 - x_1^0) + (10 - 2x_2^0)(x_2 - x_2^0) =$

$$6 + 6(x_1 - 2) + 2(x_2 - 4), \text{ 即 } g_1^0(x) = 6x_1 + 2x_2 - 14 \geq 0$$

对于不等式约束 $g_2(x) = x_1 \geq 0$, 有 $g_2(x^0) = 2$ 。而 $\frac{\partial g_2}{\partial x_1} = 1, \frac{\partial g_2}{\partial x_2} = 0$ 。所以

$$g_2^0(x) = g_2(x^0) + (x_1 - x_1^0) = 2 + x_1 - 2, \text{ 即 } g_2^0(x) = x_1 \geq 0。$$

同样, 得 $g_3^0(x) = x_2 \geq 0$ 。

综合上面的结果, 得线性化后的线性规划问题:

$$\begin{aligned} \min f^0(x) &= 4x_1 - 8x_2 + 4 \\ \text{s.t. } h_1^0(x) &= 45 - 4x_1 - 8x_2 = 0 \\ g_1^0(x) &= -14 + 6x_1 + 2x_2 \geq 0 \\ g_2^0(x) &= x_1 \geq 0 \\ g_3^0(x) &= x_2 \geq 0 \end{aligned}$$

图(4-4)示出了非线性函数及其在点 $x^0 = (2, 4)$ 处线性化后的图形。

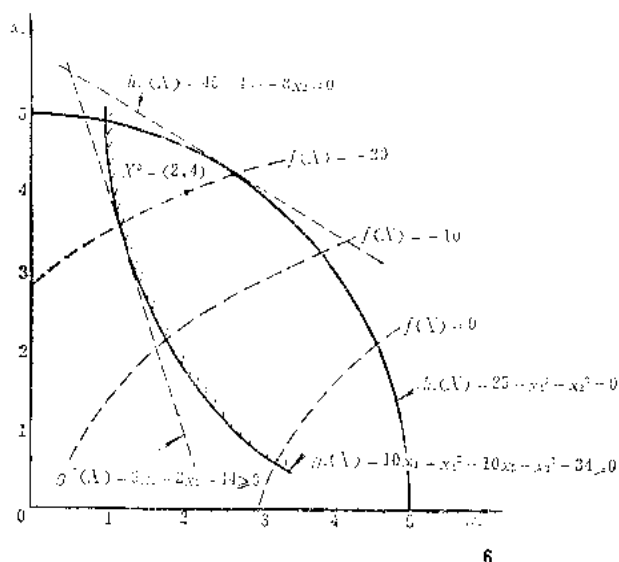


图 4-4

非线性函数经过在点 x^0 处线性化, 得到式(4-18)至(4-20)以后, 原来的非线性规划问题就变成如下的线性规划问题了。

$$\begin{aligned} \min f^0(x) \\ \text{s.t. } h_i^0(x) &= 0 & (i = 1, 2, \dots, l) & \quad (l + p = m) \\ g_j^0(x) &\leq 0 & (j = 1, 2, \dots, p) \end{aligned}$$

一、割平面法

割平面法主要用于不等式约束的非线性规划问题。设这个非线性规划问题经过在点 x^0 处线性化后得到逼近的线性规划问题为:

$$\min f^0(x)$$

$$\text{s.t. } g_j^0(\mathbf{x}) \leq 0 \quad (j = 1, 2, \dots, m)$$

我们期望它的解是比较接近于原问题的解。把这个近似问题的解记为 \mathbf{x}^1 ，再在 \mathbf{x}^1 点把原函数展开，得

$$f^1(\mathbf{x}) = f(\mathbf{x}^1) + \nabla f(\mathbf{x}^1)^T(\mathbf{x} - \mathbf{x}^1) \quad (4-21)$$

$$g_j^1(\mathbf{x}) = g_j(\mathbf{x}^1) + \nabla g_j(\mathbf{x}^1)^T(\mathbf{x} - \mathbf{x}^1)$$

然后再解下述线性规划问题

$$\begin{aligned} & \min f^1(\mathbf{x}) \\ \text{s.t. } & g_j^0(\mathbf{x}) \leq 0 \\ & g_j^1(\mathbf{x}) \leq 0 \quad (j = 1, 2, \dots, m) \end{aligned}$$

如此不断把新点处的近似约束引入约束条件中来，直到求解过程收敛为止。新的近似约束和原近似约束同时考虑可较快收敛于精确解。

在计算过程进行一段时间后，可把某些老的近似约束丢掉。也可以在开始时就选若干点，并求出这些点上的约束函数线性式，组成约束方程组。

若原问题是凸规划问题，这种方法将收敛于问题的最优解。对非凸规划问题，某些约束的线性近似可能把原问题可行域切掉一些，可能最优点恰好就在这些被切去的区域里。因为这种方法实际上是用线性近似约束把原问题可行域切掉一部分，所以称为“割平面法”。

二、小步梯度法

线性逼近法求解是按下面的迭代公式对设计点 \mathbf{x}^K 进行修改，而获得新的设计点 \mathbf{x}^{K+1} 。

$$\mathbf{x}^{K+1} = \mathbf{x}^K + \Delta \mathbf{x}_K$$

当把上式写成

$$x_i^{K+1} - x_i^K \leq \delta_i^K \quad (i = 1, 2, \dots, n)$$

而 $\delta_i^K > 0$ 是一个小数值时，可把此式作为一个约束列入原问题中去求解。当用梯度法求解时，这种方法就是用小数值 δ_i 限制各寻优方向步长的方法。可称为小步梯度法。

只有当 \mathbf{x}^K 是可行解时，此法收敛较快，否则过程收敛较慢。

三、非线性规划法

对于等式和不等式约束的非线性规划问题，有一种解法是把最速下降法（梯度法）和线性规划法结合起来求解。它的解法步骤如下：

第一步，当 \mathbf{x}^K 是不可行点时，用最速下降法把它拉到满足约束集内。此时的函数形式取为

$$T(\mathbf{x}) = \sum_{i=1}^l h_i^2(\mathbf{x}) + \sum_{j=1}^p \mu_j g_j^2(\mathbf{x})$$

第二步，再用线性规划法。每次线性规划阶段移步后，要进行一次判别，看是否满足

$$|x_i^{K+1} - x_i^K| \leq \varepsilon$$

此法使用效果比前两种方法要好些。

对于非线性规划问题，我们也可以通过泰勒级数展开的办法把约束取成线性的，目标函数取成二次函数。这种约束为线性而目标函数是二次函数的优化问题，通常称为二次规划问题。这类问题也是非线性规划中研究得较早也是比较成熟的一类问题。有多种求解二次规划问题的方法，其中一种实际上可以看成是线性规划问题中单纯形法的推广。因此用这样的处理办法来解非线性规划问题可以称为二次规划问题的线性规划解法。因为它涉及到第五章中将要讨论的基础，所以留到第五章的“二次规划迭代法”中作扼要说明。

§ 7 对偶问题的概念

下面通过一个线性规划问题，来说明对偶问题的概念。

设有一个 n 个变量， m 个约束的线性规划问题：求一组变量 $x = (x_1, x_2, \dots, x_n)$ 的值，使目标函数

$$f(x) = c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \min$$

或

$$f(x) = c^T x \rightarrow \min$$

约束条件为：

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq b_2$$

.....

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \geq b_m$$

和

$$x_i \geq 0 \quad (i = 1, 2, \dots, n)$$

或

$$f(x) = c^T x \rightarrow \min$$

$$Ax \geq b$$

$$x \geq 0$$

式中 c_i 和 b_i 都是非负的数。

假设用一组非负的变量 y_1, y_2, \dots, y_m 分别依次与上述约束不等式的每一个方程式相乘，并将其相加。由于 $y \geq 0$ ，则得

$$x_1(a_{11}y_1 + a_{21}y_2 + \dots + a_{m1}y_m) + x_2(a_{12}y_1 + a_{22}y_2 + \dots + a_{m2}y_m) + \dots + x_n(a_{1n}y_1 + a_{2n}y_2 + \dots + a_{mn}y_m) \geq b_1y_1 + b_2y_2 + \dots + b_my_m$$

如果有条件

$$a_{11}y_1 + a_{21}y_2 + \dots + a_{m1}y_m \leq c_1$$

$$a_{12}y_1 + a_{22}y_2 + \dots + a_{m2}y_m \leq c_2$$

.....

$$a_{1n}y_1 + a_{2n}y_2 + \dots + a_{mn}y_m \leq c_n$$

那么上式就变成

$$x_1c_1 + x_2c_2 + \dots + x_nc_n \geq b_1y_1 + b_2y_2 + \dots + b_my_m$$

或

$$f(x) \geq \phi(y)$$

这说明, 当 $f(x)$ 是最小值, $\phi(y)$ 是最大值时, 有 $f(x) = \phi(y)$ 。也就是说, 对应于:

$$\text{求 } x = (x_1, x_2, \dots, x_n) \quad (n \text{ 个变量})$$

使

$$\begin{aligned} f(x) = c^T x \rightarrow \min \\ \text{s.t. } Ax \geq b \\ x \geq 0 \end{aligned}$$

有对偶问题

$$\text{求 } y = (y_1, y_2, \dots, y_m) \quad (m \text{ 个变量})$$

使

$$\phi(y) = b_1y_1 + b_2y_2 + \dots + b_my_m \rightarrow \max$$

或

$$\phi(y) = b^T y \rightarrow \max$$

s.t.

$$a_{11}y_1 + a_{21}y_2 + \dots + a_{m1}y_m \leq c_1$$

$$a_{12}y_1 + a_{22}y_2 + \dots + a_{m2}y_m \leq c_2$$

$$\dots \dots \dots$$

$$a_{1n}y_1 + a_{2n}y_2 + \dots + a_{mn}y_m \leq c_n$$

和

$$y_j \geq 0 \quad (j = 1, 2, \dots, m)$$

或

$$A^T y \leq c$$

$$y \geq 0$$

即有

$$\min_x f(x) = \max_y \phi(y)$$

上面的结果归纳为:

原问题	对偶问题
$\min_x f(x) = c^T x$	$\max_y \phi(y) = b^T y$
s.t.	s.t.
$Ax \geq b$	$A^T y \leq c$
$x \geq 0$	$y \geq 0$

从这里可以看出, 对偶的主要特征是 (对线性规划问题而言):

- 1) 原问题中的 n 个变量被 m 个对偶变量所代替。
- 2) 原问题中系数 a_{ij} 的矩阵 A 转置成为对偶问题中系数 a_{ji} 的矩阵 A^T 。
- 3) 原问题中目标函数 $f(x)$ 的系数 c_i 当作对偶问题中约束方程式的右端项; 同样, 对偶问题中目标函数的系数 b_j 是原问题中约束方程式的右端项。
- 4) 将原问题中约束方程式的不等号倒置, 就是对偶问题中约束方程式的不等号。

5) 原问题中寻求目标函数 $f(x)$ 的最小值变为对偶问题中寻求目标函数 $\phi(y)$ 的最大值。

显然，利用上述对偶关系，可以达到简化计算的目的。因为对线性规划原问题可以在原问题和对偶问题中任选一个来计算。若 $m < n$ ，则选原问题进行计算；若 $m > n$ ，则选对偶问题进行计算。一般地说，约束方程式数愈少，计算工作量也愈少。

下面说明用拉格朗日函数获得上述对偶问题定义的方法。

原问题是：

$$\begin{aligned} & f(x) = c^T x \rightarrow \min \\ \text{s.t.} \quad & Ax \geq b \quad \text{或} \quad b - Ax \leq 0 \\ & x \geq 0 \end{aligned} \quad (4-22)$$

它的拉格朗日函数为

$$F(x, \lambda) = c^T x + \lambda^T (b - Ax)$$

相应的 K-T 条件是

$$\begin{aligned} \nabla_x F(x, \lambda) &= 0 \\ \lambda^T (b - Ax) &= 0 \\ \lambda &\geq 0 \end{aligned}$$

若取 $\lambda^T = y^T$ ，则 $\lambda^T (b - Ax) = y^T (b - Ax)$ 。于是拉格朗日函数变成

$$F(x, y) = c^T x + y^T (b - Ax) \quad (4-23)$$

就 $F(x, y)$ 对 y 求导，得

$$\nabla_y F(x, y) = b - Ax$$

这样，则有

$$c^T x + y^T (b - Ax) - y^T \nabla_y F(x, y) = c^T x$$

所以，原问题

$$\begin{aligned} & f(x) = c^T x \rightarrow \min \\ \text{s.t.} \quad & b - Ax \leq 0 \\ & x \geq 0 \end{aligned} \quad (4-24)$$

相当于下述问题

$$\begin{aligned} & \min_{x \geq 0} [F(x, y) - y^T \nabla_y F(x, y)] \\ \text{s.t.} \quad & \nabla_y F(x, y) \leq 0 \\ & y \geq 0 \end{aligned} \quad (4-25)$$

同样，对式 (4-22) 的对偶问题

$$\begin{aligned} & \phi(y) = b^T y \rightarrow \max \\ \text{s.t.} \quad & A^T y \leq c \quad \text{或} \quad A^T y - c \leq 0 \\ & y \geq 0 \end{aligned} \quad (4-26)$$

它的拉格朗日函数为

$$\phi(y, \mu) = -b^T y + \mu^T (A^T y - c)$$

相应的 K-T 条件是

$$\begin{aligned} \nabla_y \phi(y, \mu) &= 0 \\ \mu^T (A^T y - c) &= 0 \\ \mu &\geq 0 \end{aligned}$$

若取 $\mu^T = x^T$, 则 $\mu^T (A^T y - c) = x^T (A^T y - c)$ 。于是拉格朗日函数变成

$$\phi(x, y) = -b^T y + x^T (A^T y - c) \quad (4-27)$$

就 $\phi(x, y)$ 对 x 求导, 得

$$\nabla_x \phi(x, y) = A^T y - c$$

这样, 则有

$$-b^T y + x^T (A^T y - c) - x^T \nabla_x \phi(x, y) = -b^T y$$

所以

$$\begin{aligned} -\phi(x) &= -b^T y \rightarrow \min & (4-28) \\ \text{s.t.} \quad A^T y &\leq c \\ y &\geq 0 \end{aligned}$$

相当于下述问题

$$\begin{aligned} \min_{x \geq 0} [\phi(x, y) - x^T \nabla_x \phi(x, y)] \\ \text{或} \quad \min_{x \geq 0} [-b^T y + x^T (A^T y - c) - x^T \nabla_x \phi(x, y)] & (4-29) \\ \text{s.t.} \quad \nabla_x \phi(x, y) &\leq 0 \\ x &\geq 0 \end{aligned}$$

因为由式 (4-15) 有

$$\nabla_x F(x, y) = -(A^T y - c)$$

和

$$\begin{aligned} F(x, y) &= c^T x + y^T (b - Ax) = c^T x + y^T b - y^T Ax \\ &= b^T y - x^T (A^T y - c) \end{aligned}$$

所以, 有

$$-b^T y + x^T (A^T y - c) - x^T \nabla_x \phi(x, y) = -F(x, y) + x^T \nabla_x F(x, y)$$

这样

$$\min_{x \geq 0} [-b^T y + x^T (A^T y - c) - x^T \nabla_x \phi(x, y)] = \min_{x \geq 0} [-F(x, y) + x^T \nabla_x F(x, y)]$$

又可写成

$$-\min_{x \geq 0} [F(x, y) - x^T \nabla_x F(x, y)] = \max_{x \geq 0} [F(x, y) - x^T \nabla_x F(x, y)]$$

于是两个对偶问题可以写成下面的形式

$$\begin{aligned} \min_{x \geq 0} [F(x, y) - x^T \nabla_x F(x, y)] & (4-30) \\ \text{s.t.} \end{aligned}$$

$$\begin{aligned} \nabla_x \bar{F}(x, y) &\leq 0 \\ y &\geq 0 \end{aligned}$$

和

$$\max_{y > 0} [F(x, y) - x^T \nabla_x F(x, y)] \quad (4-31)$$

s. t.

$$\begin{aligned} \nabla_x F(x, y) &\geq 0 \\ x &\geq 0 \end{aligned}$$

若目标函数是二次型，而约束条件是线性函数时，称为二次规划问题。把上面的概念应用于二次规划问题时，当原问题是

$$c^T x + x^T D x \rightarrow \max \quad (4-32)$$

$$\text{s. t.} \quad A x \leq b \quad \text{或} \quad A x - b \leq 0$$

把它转换成极小问题，它的拉格朗日函数为

$$F(x, \lambda) = -(c^T x + x^T D x) + \lambda^T (A x - b)$$

仿照式 (4-22)，原问题可以写成

$$\min_{x > 0} [F(x, \lambda) - \lambda^T \nabla_x F(x, \lambda)] \quad (4-33)$$

$$\text{s. t.} \quad \begin{aligned} \nabla_x F(x, \lambda) &\leq 0 \\ \lambda &\geq 0 \end{aligned}$$

再按式 (4-31)，则有式 (4-32) 的对偶问题

$$\begin{aligned} \max_{\lambda > 0} [F(x, \lambda) - x^T \nabla_x F(x, \lambda)] \\ \text{s. t.} \quad \nabla_x F(x, \lambda) &\geq 0 \\ x &\geq 0 \end{aligned} \quad (4-34)$$

由于
则
所以

$$\begin{aligned} \nabla_x F(x, \lambda) &= -c - 2Dx + \lambda^T A \\ x^T \nabla_x F(x, \lambda) &= -x^T (c - \lambda^T A) - 2x^T D x \end{aligned}$$

$$\begin{aligned} F(x, \lambda) - x^T \nabla_x F(x, \lambda) &= -c^T x - x^T D x + 2x^T D x + \\ x^T c - x^T \lambda^T A + \lambda^T A x - \lambda^T b &= -b^T \lambda + x^T D x \end{aligned}$$

因此，式 (4-26) 又可写成

$$\begin{aligned} \max_{\lambda > 0} [- (b^T \lambda - x^T D x)] \\ \text{s. t.} \quad c &\leq -2Dx + A^T \lambda \\ x &\geq 0 \end{aligned}$$

或等价于

$$\begin{aligned} \min_{\lambda > 0} (b^T \lambda - y^T D y) \\ \text{s. t.} \quad c &\leq -2Dx + A^T \lambda \\ x &\geq 0 \end{aligned} \quad (4-35)$$

于是式 (4-32) 的对偶问题是式 (4-35)，即

$$\begin{aligned} \min_{\lambda > 0} (b^T \lambda - x^T D x) &= \max_{x > 0} (C^T x + x^T D x) \\ C &\leq -2Dx + A^T \lambda \quad Ax \leq b \end{aligned}$$

§ 8 几何规划简介

具有下列形式的目标函数的优化问题是几何规划问题。

$$F(x) = \sum_{j=1}^m c_j y_j(x) \quad (4-36)$$

式中系数 c_j 是正常数, 即 $c_j > 0$; $x = (x_1, x_2, \dots, x_n)$;

$$y_j(x) = \prod_{i=1}^n x_i^{a_{ij}} = x_1^{a_{1j}} x_2^{a_{2j}} \dots x_n^{a_{nj}} \quad (4-37)$$

即 $y_j(x)$ 是指数为实常数 a_{ij} 的乘积函数。

解这类几何规划问题可以借助于“权”和对偶函数, 使问题简化。

可以把“权”定义为: 当目标函数在点 x^* 取得最优值 $F(x^*)$ 时, 目标函数值与其各组成项 $c_j y_j(x^*)$ 的比值。若记 ω_j 为第 j 项的权, 则有

$$\omega_j = \frac{c_j y_j(x^*)}{F(x^*)} \quad (4-38)$$

因为 $F(x^*) = \sum_{j=1}^m c_j y_j(x^*)$, 所以显然有

$$\sum_{j=1}^m \omega_j = 1 \quad (4-39)$$

我们称它为权的归一性或正规性条件。

又由 $F(x)$ 为极小值的必要条件 $\frac{\partial F}{\partial x_i} = 0$, 有

$$\frac{\partial F}{\partial x_i} = \sum_{j=1}^m c_j a_{ij} \frac{y_j(x)}{x_i} = 0 \quad (i = 1, 2, \dots, n)$$

由式 (4-38) 得

$$c_j y_j(x^*) = \omega_j F(x^*)$$

代入, 则 (此时 x_i 应取 x_i^*)

$$\frac{\partial F}{\partial x_i} = \sum_{j=1}^m \left(\frac{a_{ij}}{x_i^*} \right) \omega_j F(x^*) = \sum_{j=1}^m a_{ij} \omega_j \frac{F(x^*)}{x_i^*} = 0$$

考虑到 $\frac{F(x^*)}{x_i^*}$ 是已经确定的值, 所以有

$$\sum_{j=1}^m a_{ij} \omega_j = 0 \quad (i = 1, 2, \dots, n) \quad (4-40)$$

我们称它为 a_{ij} 和 ω_j 的正交性条件。

几何规划问题的对偶函数有多种形成方法。这里, 我们把它理解为: 几何规划是在目标函数 $F(x)$ 的各部分 $c_j y_j(x)$ 之间寻求分配总目标值的最优方案, 而不直接去求变

量的最优值。因此，就要对各组成项对应的权 ω_j 进行优选，即求解一组 m 个最优的权来。

现在，从 a_{ij} 和 ω_j 的正交性条件可以提供 n 个以 ω_j 为变量的方程，再加上 ω_j 的正规性条件给出的一个方程式，则共有 $n+1$ 个方程式可以求解 m 个权 ω_j 。

定义 $m - (n+1) = m - n - 1$ 为问题的自由度或问题的难度。当 $m = n+1$ 时，自由度为零，问题有唯一解。

对于几何规划问题，目前对自由度为零和 1 的问题具有特殊的优越性，问题的求解比较方便。自由度等于 1 时，可以用一维搜索法求解最优权。当自由度大于 1 时，几何规划并没有优越性。

自式 (4-39) 的 $\sum_{j=1}^m \omega_j = 1$ ，则目标函数在点 x^* 处的值 $F(x^*)$ 可以写成

$$F(x^*) = F(x^*)^{\sum_{j=1}^m \omega_j} = \prod_{j=1}^m [F(x^*)]^{\omega_j}$$

取 $F(x^*) = \prod_{j=1}^m F(x^*)^{\omega_j}$ 的形式，则有

$$F(x^*) = \prod_{j=1}^m \left[\frac{c_j y_j(x^*)}{\omega_j} \right]^{\omega_j} = \prod_{j=1}^m \left[\frac{c_j}{\omega_j} \right]^{\omega_j} \prod_{j=1}^m [y_j(x^*)]^{\omega_j}$$

自式 (4-37) 可得

$$\begin{aligned} \prod_{j=1}^m [y_j(x^*)]^{\omega_j} &= \prod_{j=1}^m \left[\prod_{i=1}^n (x_i^*)^{a_{ij}} \right]^{\omega_j} = \prod_{j=1}^m \prod_{i=1}^n (x_i^*)^{a_{ij} \omega_j} \\ &= \prod_{i=1}^n (x_i^*)^{\sum_{j=1}^m a_{ij} \omega_j} \end{aligned}$$

但自式 (4-32) 的正交性条件 $\sum_{j=1}^m a_{ij} \omega_j = 0$ ，则

$$\prod_{j=1}^m [y_j(x^*)]^{\omega_j} = \prod_{i=1}^n (x_i^*)^0 = 1$$

因此，可得

$$F(x^*) = \prod_{j=1}^m \left[\frac{c_j}{\omega_j} \right]^{\omega_j} \quad (4-41)$$

可见，目标函数 $F(x)$ 的极小值 $F(x^*)$ 并不依赖于原来的变量 x 。在确定 x^* 之前就已经确定了 $F(x^*)$ 值。

自柯西不等式（几何不等式）

$$\omega_1 v_1 + \omega_2 v_2 + \dots + \omega_n v_n \geq v_1^{\omega_1} \cdot v_2^{\omega_2} \dots v_n^{\omega_n}$$

若令 $u_j = \omega_j v_j$ ，则有

$$u_1 + u_2 + \dots + u_n \geq \left[\frac{u_1}{\omega_1} \right]^{\omega_1} \left[\frac{u_2}{\omega_2} \right]^{\omega_2} \dots \left[\frac{u_n}{\omega_n} \right]^{\omega_n} \quad (4-42)$$

当 $\frac{u_1}{\omega_1} = \frac{u_2}{\omega_2} = \dots = \frac{u_n}{\omega_n}$ 时，式 (4-42) 的等号成立。

当 ω_j 满足式 (4-39) 的正规性条件时，式 (4-42) 就成为几何规划的基础。

取 $u_i = c_i y_i$ ，则式 (4-34) 给出

$$c_1 y_1 + c_2 y_2 + \dots + c_m y_m \geq \left[\frac{c_1 y_1}{\omega_1} \right]^{\omega_1} \cdot \left[\frac{c_2 y_2}{\omega_2} \right]^{\omega_2} \dots \left[\frac{c_m y_m}{\omega_m} \right]^{\omega_m} \quad (4-43)$$

它的左端各项是原来的 $F(x) = \sum_{j=1}^m c_j y_j(x)$ ，即原函数；右边各项的乘积

$\prod_{j=1}^m \left[\frac{c_j y_j(x)}{\omega_j} \right]^{\omega_j}$ 则称为对偶函数。

把式 (4-37) 代入上式，则其右端变为

$$\left[\frac{c_1}{\omega_1} \right]^{\omega_1} \left(x_1^{a_{11}} \cdot x_2^{a_{21}} \dots x_n^{a_{n1}} \right)^{\omega_1} \cdot \left[\frac{c_2}{\omega_2} \right]^{\omega_2} \left(x_1^{a_{12}} \cdot x_2^{a_{22}} \dots x_n^{a_{n2}} \right)^{\omega_2} \dots \left[\frac{c_m}{\omega_m} \right]^{\omega_m} \left(x_1^{a_{1m}} \cdot x_2^{a_{2m}} \dots x_n^{a_{nm}} \right)^{\omega_m}$$

或写成

$$\left[\frac{c_1}{\omega_1} \right]^{\omega_1} \cdot \left[\frac{c_2}{\omega_2} \right]^{\omega_2} \dots \left[\frac{c_m}{\omega_m} \right]^{\omega_m} x_1^{\sum_{j=1}^m a_{1j} \omega_j} \cdot x_2^{\sum_{j=1}^m a_{2j} \omega_j} \dots x_n^{\sum_{j=1}^m a_{nj} \omega_j}$$

如果权 ω_j 满足正交性条件，即 $\sum_{j=1}^m a_{ij} \omega_j = 0$ ，则上式变成

$$\left[\frac{c_1}{\omega_1} \right]^{\omega_1} \cdot \left[\frac{c_2}{\omega_2} \right]^{\omega_2} \dots \left[\frac{c_m}{\omega_m} \right]^{\omega_m}$$

从而式 (4-35) 可以写成

$$c_1 y_1 + c_2 y_2 + \dots + c_m y_m \geq \left[\frac{c_1}{\omega_1} \right]^{\omega_1} \cdot \left[\frac{c_2}{\omega_2} \right]^{\omega_2} \dots \left[\frac{c_m}{\omega_m} \right]^{\omega_m}$$

或

$$\sum_{j=1}^m c_j y_j(x) \geq \prod_{j=1}^m \left[\frac{c_j}{\omega_j} \right]^{\omega_j} \quad (4-44)$$

这样，求原函数 $F(x) = \sum_{j=1}^m c_j y_j(x)$ 的极小值问题就变成求对偶函数

$\prod_{j=1}^m \left[\frac{c_j}{\omega_j} \right]^{\omega_j} = \phi(\omega)$ 的极大值问题，即有

$$\min_x F(x) = \max_{\omega} \phi(\omega)$$

例：设目标函数的形式是

$$f(x) = x_1^2 + 2x_2 + \frac{3}{x_1 x_2} + 2x_1 x_2$$

求 x_i 的正值，使 $f(x)$ 为极小。

这个问题，系数 a_{ij} 和 c_j 分别是（这里的变量数 $i=1,2$ ，项数 $j=1,2,3,4$ ）：

$$\begin{array}{lll} a_{11} = 2, & a_{21} = 0, & c_1 = 1, \\ a_{12} = 0, & a_{22} = 1, & c_2 = 2, \end{array}$$

$$\begin{aligned} a_{13} &= -1, & a_{23} &= -1, & c_3 &= 3, \\ a_{14} &= 1, & a_{24} &= 1, & c_4 &= 2. \end{aligned}$$

则对偶函数的目标函数是

$$\phi(\omega) = \left(\frac{1}{\omega_1}\right)^{\omega_1} \left(\frac{2}{\omega_2}\right)^{\omega_2} \left(\frac{3}{\omega_3}\right)^{\omega_3} \left(\frac{2}{\omega_4}\right)^{\omega_4}$$

相应的约束条件是

$$\begin{aligned} \omega_1 + \omega_2 + \omega_3 + \omega_4 &= 1 \\ a_{11}\omega_1 + a_{12}\omega_2 + a_{13}\omega_3 + a_{14}\omega_4 &= 0 \\ a_{21}\omega_1 + a_{22}\omega_2 + a_{23}\omega_3 + a_{24}\omega_4 &= 0 \end{aligned}$$

或

$$\begin{aligned} \omega_1 + \omega_2 + \omega_3 + \omega_4 &= 1 \\ 2\omega_1 - \omega_3 + \omega_4 &= 0 \\ \omega_2 - \omega_3 + \omega_4 &= 0 \end{aligned}$$

满足上面三个约束条件的任何 ω_j 值, 将给出原问题 $f(x)$ 的极小值解。

由上面三个约束条件可得

$$\omega^1 = \left\{ \omega_1 = \frac{1}{15}, \quad \omega_2 = \frac{2}{15}, \quad \omega_3 = \frac{7}{15}, \quad \omega_4 = \frac{1}{3} \right\}$$

和

$$\omega^2 = \left\{ \omega_1 = \frac{1}{10}, \quad \omega_2 = \frac{1}{5}, \quad \omega_3 = \frac{9}{20}, \quad \omega_4 = \frac{1}{4} \right\}$$

分别代入 $\phi(\omega)$ 中得

$$\phi(\omega^1) = 7.44, \quad \phi(\omega^2) = 7.89.$$

取两者中的大者, 它给出 $\phi(\omega)$ 的极大值。

则有

$$\min f(x) = \max \phi(\omega) = 7.89$$

根据权 ω_j 的定义式 (4-38)

$$\omega_j = \frac{c_j y_j(x^*)}{F(x^*)}$$

可以求出

$$y_j(x^*) = \frac{\omega_j F(x^*)}{c_j}$$

或

$$c_j y_j(x^*) = \omega_j F(x^*)$$

从而给出 $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ 的值。

如原问题有约束条件时, 需要把约束条件按处理目标函数那样进行处理, 以便把约束条件的影晌考虑进来。这时, 不仅对偶函数 $\phi(\omega)$ 中要考虑约束条件的影晌, 对偶约束条件 (正交性条件) $\sum_{j=1}^m a_{ij} \omega_j = 0$ 也要考虑约束条件的影晌。

若原问题的约束条件是

$$g(x) = \sum_{K=1}^p g_K(x) \quad (4-45)$$

其中的

$$g_K(x) = c_{rK}(x) \prod_{j=1}^n x_j^{a_{rKj}} \leq 1 \quad (4-46)$$

则对偶目标函数应扩展为

$$\phi(\omega) = \prod_{j=1}^n \left[\frac{c_j}{\omega_j} \right]^{\omega_j} \cdot \prod_{K=1}^p \left[\frac{c_{rK} \omega_{r0}}{\omega_{rK}} \right]^{\omega_{rK}} \quad (4-47)$$

其中 $\omega_{r0} = \sum_{K=1}^p \omega_{rK} \geq 0$

而对偶约束条件应扩展为

$$\sum_{j=1}^n a_{1j} \omega_j + \sum_{K=1}^p a_{rK1} \omega_{rK} = 0 \quad (4-48)$$

例：求目标函数 $f(x) = 3x_1 + 5x_2^2$ 的极小值。约束条件是 $\frac{20}{x_1 x_2} \leq 10$ 。

把约束条件改写为 $2x_1^{-1} x_2^{-1} \leq 1$

这里只有一个约束条件， $m=1$ ，且只有一项，即 $K=1$ ，有

$$\begin{aligned} c_1 &= 3, & c_2 &= 5, & c_{11} &= 2, \\ a_{11} &= 1, & a_{12} &= 0, & a_{111} &= -1, & a_{112} &= 1, \\ a_{21} &= 0, & a_{22} &= 2. \end{aligned}$$

所以，有

$$\phi(\omega) = \left[\frac{c_1}{\omega_1} \right]^{\omega_1} \left[\frac{c_2}{\omega_2} \right]^{\omega_2} \left[\frac{c_{11} \omega_{10}}{\omega_{11}} \right]^{\omega_{11}} = \left[\frac{3}{\omega_1} \right]^{\omega_1} \left[\frac{5}{\omega_2} \right]^{\omega_2} \left[\frac{2\omega_{10}}{\omega_{11}} \right]^{\omega_{11}}$$

$$\omega_1 + \omega_2 = 1$$

$$a_{11}\omega_1 + a_{12}\omega_2 + a_{111}\omega_{11} = 0 \quad \text{或} \quad \omega_1 - \omega_{11} = 0$$

$$a_{21}\omega_1 + a_{22}\omega_2 + a_{112}\omega_{11} = 0 \quad \text{或} \quad 2\omega_2 - \omega_{11} = 0$$

解此三式的联立方程式，得

$$\omega_1 = \frac{2}{3}, \quad \omega_2 = \frac{1}{3}, \quad \omega_{11} = \frac{2}{3}。$$

又由 $\omega_{r0} = \sum_{K=1}^p \omega_{rK} \geq 0$ ，得 $\omega_{10} = \omega_{11} = \frac{2}{3} \geq 0$

所以得

$$\min f(x) = \max \phi(\omega) = \left[\frac{3}{\frac{2}{3}} \right]^{\frac{2}{3}} \left[\frac{5}{\frac{1}{3}} \right]^{\frac{1}{3}} \left[\frac{2 \cdot \frac{2}{3}}{\frac{2}{3}} \right]^{\frac{2}{3}} = 10.66$$

根据 $f_1(x) = \min f(x) \cdot \omega_1$ 和 $f_2(x) = \min f(x) \cdot \omega_2$ 可得：

$$3x_1 = 10.66 \times \frac{2}{3}, \quad \text{即} \quad x_1 = 2.37,$$

$$5x_2^2 = 10.66 \times \frac{1}{3}, \quad \text{即} \quad x_2 = 0.843。$$

第五章 非线性规划解法之一

——间接方法

第四章已介绍了一类约束非线性优化问题，即非线性规划问题的线性逼近求解方法，这一章和下一章将介绍另一大类求解非线性规划问题的方法。

这类方法的基础是无约束优化问题的求解技术，即把无约束优化问题的求解方法推广应用于约束优化问题中去。

有两种可能的途径。一是利用某些技巧把约束条件（函数）引进原目标函数中去，形成一个新的改造过的目标函数，或一个新的增广目标函数，从而使原来的约束优化问题转化为无约束优化问题，然后采用无约束优化方法求解。由于这是把约束结合进目标函数中，作间接处理的方法，所以称为间接方法。二是把约束条件看成是一组限制曲面，并引进考虑约束的额外步骤进行求解。由于这种方法是在约束条件的限制范围内直接应用无约束优化方法求解的，所以称为直接方法。

在间接方法中，有一组算法是通过消去变量和约束的途径，把约束函数和目标函数结合起来，形成一个隐含约束函数的新目标函数。另一组算法则是把约束函数进行特殊加权处理，使之成为一个新的增广目标函数的组成项。

下面我们把约束非线性优化问题——非线性规划问题一般地表述为：

求一组变量 $\mathbf{x} = (x_1, x_2, \dots, x_n)$ 使目标函数

$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$$

极小（极大问题可以写成 $-f(\mathbf{x})$ 的极小问题），且满足约束条件

$$h_i(\mathbf{x}) = h_i(x_1, x_2, \dots, x_n) = 0$$

$$g_j(\mathbf{x}) = g_j(x_1, x_2, \dots, x_n) \leq 0$$

$$(i = 1, 2, \dots, l; j = l + 1, \dots, p) \quad (l + p = m)$$

对于等式约束优化问题，在多元微积分中一般介绍的是采用解析方法（如约束变量法、拉格朗日乘子法等）进行求解。显然，对实际的结构优化问题主要是采用数值解法，但是解析方法的思路却是数值解法的基础。因此，下面还要对约束变量法和拉格朗日乘子法加以说明。

§ 1 约束变量法和拉格朗日(Lagrange)乘子法

在第一章§2中，曾对有约束的二元函数求极值的方法做过说明。现在把它推广到有 m 个等式约束 $h_k(\mathbf{x}) = 0 (k = 1, 2, \dots, m < n)$ 和 n 个变量 $\mathbf{x} = (x_1, x_2, \dots, x_n)$ 的一般情形。

一、约束变量法

对上面的问题，它的优化解应当是既满足 m 个等式约束条件，又要使目标函数 $f(\mathbf{x})$ 取极值（局部极值）的一组变量 \mathbf{x} 。但是，若要对 m 个等式约束所提供的方程组求解 n 个

变量, 由于 $m < n$, 它是一个非线性不定的方程组。不过可以通过 m 个等式约束条件, 解出 n 个变量中的 m 个, 即把这 m 个变量用余下的 $n - m$ 个变量表示。就是用 $n - m$ 个变量代入目标函数 $f(x)$ 中去。这样, 就把原来的约束问题转变成一个新的无约束问题。为此, 可以把变量分组。例如把前 m 个变量 $x_k (k = 1, 2, \dots, m)$ 做为未知变量(状态变量)写在方程的左边, 而余下的 $n - m$ 个变量做为已知变量(决策变量)移到方程的右边, 并利用在极值点 x^* 处的

$$df(x^*) = \sum_{i=1}^n \frac{\partial f}{\partial x_i} dx_i = \nabla f(x^*)^T dx = 0$$

和

$$dh_k(x^*) = \sum_{i=1}^n \frac{\partial h_k}{\partial x_i} dx_i = \nabla h_k(x^*)^T dx = 0 \quad (k = 1, 2, \dots, m)$$

的条件来求解。

经过这样的处理, 则 m 个约束方程就可写成

$$\sum_{i=1}^m \frac{\partial h_k}{\partial x_i} dx_i = - \sum_{i=m+1}^n \frac{\partial h_k}{\partial x_i} dx_i \quad (k = 1, 2, \dots, m) \quad (5-1)$$

这就是把 $dx_i (i = 1, 2, \dots, m)$ 用余下的 $n - m$ 个微小向量 $dx_j (j = m + 1, \dots, n)$ 表示, 从而各 x_j 可以任意选取, 求解出 m 个 x_i 。

把式 (5-1) 展开, 则有

$$\frac{\partial h_1}{\partial x_1} dx_1 + \frac{\partial h_1}{\partial x_2} dx_2 + \dots + \frac{\partial h_1}{\partial x_m} dx_m = - \frac{\partial h_1}{\partial x_{m+1}} dx_{m+1} - \dots - \frac{\partial h_1}{\partial x_n} dx_n$$

$$\frac{\partial h_2}{\partial x_1} dx_1 + \frac{\partial h_2}{\partial x_2} dx_2 + \dots + \frac{\partial h_2}{\partial x_m} dx_m = - \frac{\partial h_2}{\partial x_{m+1}} dx_{m+1} - \dots - \frac{\partial h_2}{\partial x_n} dx_n$$

.....

$$\frac{\partial h_m}{\partial x_1} dx_1 + \frac{\partial h_m}{\partial x_2} dx_2 + \dots + \frac{\partial h_m}{\partial x_m} dx_m = - \frac{\partial h_m}{\partial x_{m+1}} dx_{m+1} - \dots - \frac{\partial h_m}{\partial x_n} dx_n$$

为了简明, 用矩阵来表述上式, 并进行 dx 的求解运算。

令 $dx_A = [dx_1, dx_2, \dots, dx_m]^T$ $dx_B = [dx_{m+1}, \dots, dx_n]^T$

$$A = A \begin{pmatrix} h_1, h_2, \dots, h_m \\ x_1, x_2, \dots, x_m \end{pmatrix} = \begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \dots & \frac{\partial h_1}{\partial x_m} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \dots & \frac{\partial h_2}{\partial x_m} \\ \dots & \dots & \dots & \dots \\ \frac{\partial h_m}{\partial x_1} & \frac{\partial h_m}{\partial x_2} & \dots & \frac{\partial h_m}{\partial x_m} \end{pmatrix}$$

$$B = \begin{pmatrix} \frac{\partial h_1}{\partial x_{m+1}} & \frac{\partial h_1}{\partial x_{m+2}} & \dots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_{m+1}} & \frac{\partial h_2}{\partial x_{m+2}} & \dots & \frac{\partial h_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial h_m}{\partial x_{m+1}} & \frac{\partial h_m}{\partial x_{m+2}} & \dots & \frac{\partial h_m}{\partial x_n} \end{pmatrix}$$

$$\nabla h = [\nabla h_1, \nabla h_2, \dots, \nabla h_m] \text{ 而 } \nabla h_i = \left[\frac{\partial h_i}{\partial x_1}, \frac{\partial h_i}{\partial x_2}, \dots, \frac{\partial h_i}{\partial x_m} \right]^T$$

则式 (5-1) 可以写成 $A dx_A = -B dx_B$ (5-1a)

如果 A 是非奇异的, 则可解得

$$dx_A = -A^{-1}B dx_B \quad (5-2)$$

这就是用 $n-m$ 个变量表示 x_i 的 m 个解。

如果不能保证 A 的非奇异条件, 则可调整 x_i 的组合, 即可另选一组 m 个变量为状态变量, 使组成的矩阵 A 是非奇异的。有时称矩阵 A 为雅可比 (Jacobian) 矩阵 (它是一阶偏导数矩阵)。

$$\text{再令 } C = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_m} \right], \quad D = \left[\frac{\partial f}{\partial x_{m+1}}, \dots, \frac{\partial f}{\partial x_n} \right]$$

则 $df = \nabla f(x)^T dx = [C; D] \begin{Bmatrix} dx_A \\ dx_B \end{Bmatrix} = C dx_A + D dx_B$

在极值点处, 有

$$df = C dx_A + D dx_B = 0$$

把 $dx_A = -A^{-1}B dx_B$ 代入上式, 得

$$df = (-CA^{-1}B + D) dx_B$$

$$\text{若令 } \frac{df}{dx_B} \equiv \left[\frac{\partial f}{\partial x_{m+1}}, \dots, \frac{\partial f}{\partial x_n} \right], \text{ 则有 } \frac{df}{dx_B} = D - CA^{-1}B$$

用 df 表达式可以检验利用 m 个约束方程组 $\frac{\partial h_k}{\partial x_i}$ 解得的 x_i , 是否能使目标函数值有改进。

在极值点处的 $\frac{df}{dx_B} = D - CA^{-1}B = 0$ 的条件给出联系 x_B 和 x_A 之间关系的 $n-m$ 个方程的方程组。用这些再和 m 个约束方程联立共得 n 个方程式, 从而可以解得 n 个变量 $x = (x_1, x_2, \dots, x_n)$ 。

以上就是约束变量法的解题思路和方法。

为了写出 $D - CA^{-1}B$ 的具体展开形式, 用一个 $n=5, m=3$ 的简单情形做例子进行说明。

这时

$$A = \begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \frac{\partial h_1}{\partial x_3} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \frac{\partial h_2}{\partial x_3} \\ \frac{\partial h_3}{\partial x_1} & \frac{\partial h_3}{\partial x_2} & \frac{\partial h_3}{\partial x_3} \end{pmatrix} \quad B = \begin{pmatrix} \frac{\partial h_1}{\partial x_4} & \frac{\partial h_1}{\partial x_5} \\ \frac{\partial h_2}{\partial x_4} & \frac{\partial h_2}{\partial x_5} \\ \frac{\partial h_3}{\partial x_4} & \frac{\partial h_3}{\partial x_5} \end{pmatrix} \quad dx_A = \begin{Bmatrix} dx_1 \\ dx_2 \\ dx_3 \end{Bmatrix} \quad dx_B = \begin{Bmatrix} dx_4 \\ dx_5 \end{Bmatrix}$$

$$C = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \frac{\partial f}{\partial x_3} \right] \quad D = \left[\frac{\partial f}{\partial x_4} \quad \frac{\partial f}{\partial x_5} \right]$$

记 A 的逆阵为

$$A^{-1} = \frac{1}{|A|} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{21} & \bar{A}_{31} \\ \bar{A}_{12} & \bar{A}_{22} & \bar{A}_{32} \\ \bar{A}_{13} & \bar{A}_{23} & \bar{A}_{33} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}$$

一般地说, \bar{A}_{ij} 中不含 $\partial h_i / \partial x_j$ 项。

这样, 则 $CA^{-1}B$ 可以写成

$$\begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \frac{\partial f}{\partial x_3} \end{bmatrix} \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} \frac{\partial h_1}{\partial x_4} & \frac{\partial h_1}{\partial x_5} \\ \frac{\partial h_2}{\partial x_4} & \frac{\partial h_2}{\partial x_5} \\ \frac{\partial h_3}{\partial x_4} & \frac{\partial h_3}{\partial x_5} \end{pmatrix}$$

为了简化, 我们只取相乘所得 1×2 阶矩阵的第一个元素来分析。其形式是

$$\begin{aligned} & \frac{\partial f}{\partial x_1} \left(A_{11} \frac{\partial h_1}{\partial x_4} + A_{12} \frac{\partial h_2}{\partial x_4} + A_{13} \frac{\partial h_3}{\partial x_4} \right) + \frac{\partial f}{\partial x_2} \left(A_{21} \frac{\partial h_1}{\partial x_4} + A_{22} \frac{\partial h_2}{\partial x_4} + A_{23} \frac{\partial h_3}{\partial x_4} \right) \\ & + \frac{\partial f}{\partial x_3} \left(A_{31} \frac{\partial h_1}{\partial x_4} + A_{32} \frac{\partial h_2}{\partial x_4} + A_{33} \frac{\partial h_3}{\partial x_4} \right) \end{aligned}$$

当把 A_{ij} 计算出来并代入, 则得下面的形式

$$\frac{1}{|A|} \left\{ \begin{array}{c} \frac{\partial f}{\partial x_1} \begin{vmatrix} \frac{\partial h_1}{\partial x_4} & \frac{\partial h_1}{\partial x_2} & \frac{\partial h_1}{\partial x_3} \\ \frac{\partial h_2}{\partial x_4} & \frac{\partial h_2}{\partial x_2} & \frac{\partial h_2}{\partial x_3} \\ \frac{\partial h_3}{\partial x_4} & \frac{\partial h_3}{\partial x_2} & \frac{\partial h_3}{\partial x_3} \end{vmatrix} - \frac{\partial f}{\partial x_2} \begin{vmatrix} \frac{\partial h_1}{\partial x_4} & \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_3} \\ \frac{\partial h_2}{\partial x_4} & \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_3} \\ \frac{\partial h_3}{\partial x_4} & \frac{\partial h_3}{\partial x_1} & \frac{\partial h_3}{\partial x_3} \end{vmatrix} + \frac{\partial f}{\partial x_3} \begin{vmatrix} \frac{\partial h_1}{\partial x_4} & \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} \\ \frac{\partial h_2}{\partial x_4} & \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} \\ \frac{\partial h_3}{\partial x_4} & \frac{\partial h_3}{\partial x_1} & \frac{\partial h_3}{\partial x_2} \end{vmatrix} \end{array} \right\}$$

其中

$$\begin{aligned} |A| = & \frac{\partial h_1}{\partial x_1} \frac{\partial h_2}{\partial x_2} \frac{\partial h_3}{\partial x_3} + \frac{\partial h_2}{\partial x_1} \frac{\partial h_3}{\partial x_2} \frac{\partial h_1}{\partial x_3} + \frac{\partial h_3}{\partial x_1} \frac{\partial h_1}{\partial x_2} \frac{\partial h_2}{\partial x_3} - \frac{\partial h_1}{\partial x_3} \frac{\partial h_2}{\partial x_1} \frac{\partial h_3}{\partial x_2} \\ & - \frac{\partial h_2}{\partial x_3} \frac{\partial h_3}{\partial x_2} \frac{\partial h_1}{\partial x_1} - \frac{\partial h_3}{\partial x_3} \frac{\partial h_1}{\partial x_2} \frac{\partial h_2}{\partial x_1} \end{aligned}$$

也可把上式中括号 $\{\dots\}$ 内的三个 3×3 阶行列式分别用 $A\left(\frac{h_1, h_2, h_3}{x_4, x_2, x_3}\right)$, $A\left(\frac{h_1, h_2, h_3}{x_4, x_1, x_3}\right)$

和 $A\left(\frac{h_1, h_2, h_3}{x_4, x_1, x_2}\right)$ 代表, 则它可简写成

$$\frac{1}{|A|} \left\{ \frac{\partial f}{\partial x_1} A\left(\frac{h_1, h_2, h_3}{x_4, x_2, x_3}\right) - \frac{\partial f}{\partial x_2} A\left(\frac{h_1, h_2, h_3}{x_4, x_1, x_3}\right) + \frac{\partial f}{\partial x_3} A\left(\frac{h_1, h_2, h_3}{x_4, x_1, x_2}\right) \right\}$$

同样方法, 可以写出该 1×2 阶矩阵的第二个元素

$$\frac{1}{|A|} \left\{ \frac{\partial f}{\partial x_1} A\left(\frac{h_1, h_2, h_3}{x_5, x_2, x_3}\right) - \frac{\partial f}{\partial x_2} A\left(\frac{h_1, h_2, h_3}{x_5, x_1, x_3}\right) + \frac{\partial f}{\partial x_3} A\left(\frac{h_1, h_2, h_3}{x_5, x_1, x_2}\right) \right\}$$

结果, $D - CA^{-1}B$ 行矩阵可以写成

$$\left[\frac{\partial f}{\partial x_4}, \frac{\partial f}{\partial x_5} \right] - \frac{1}{|A|} \left[\left\{ \frac{\partial f}{\partial x_1} A \left(\frac{h_1 h_2 h_3}{x_4 x_2 x_3} \right) - \frac{\partial f}{\partial x_2} A \left(\frac{h_1 h_2 h_3}{x_4 x_1 x_3} \right) + \frac{\partial f}{\partial x_3} A \left(\frac{h_1 h_2 h_3}{x_4 x_1 x_2} \right) \right\}, \right. \\ \left. \left\{ \frac{\partial f}{\partial x_1} A \left(\frac{h_1 h_2 h_3}{x_5 x_2 x_3} \right) - \frac{\partial f}{\partial x_2} A \left(\frac{h_1 h_2 h_3}{x_5 x_1 x_3} \right) + \frac{\partial f}{\partial x_3} A \left(\frac{h_1 h_2 h_3}{x_5 x_1 x_2} \right) \right\} \right]$$

从而可得行矩阵的两个元素分别为

$$\frac{\partial f}{\partial x_4} |A| - \frac{\partial f}{\partial x_1} A \left(\frac{h_1 h_2 h_3}{x_4 x_2 x_3} \right) + \frac{\partial f}{\partial x_2} A \left(\frac{h_1 h_2 h_3}{x_4 x_1 x_3} \right) - \frac{\partial f}{\partial x_3} A \left(\frac{h_1 h_2 h_3}{x_4 x_1 x_2} \right)$$

和

$$\frac{\partial f}{\partial x_5} |A| - \frac{\partial f}{\partial x_1} A \left(\frac{h_1 h_2 h_3}{x_5 x_2 x_3} \right) + \frac{\partial f}{\partial x_2} A \left(\frac{h_1 h_2 h_3}{x_5 x_1 x_3} \right) - \frac{\partial f}{\partial x_3} A \left(\frac{h_1 h_2 h_3}{x_5 x_1 x_2} \right)$$

它们又分别是下面的行列式的展开形式

$$A_4 \left(\frac{f, h_1, h_2, h_3}{x_4, x_1, x_2, x_3} \right) = \begin{vmatrix} \frac{\partial f}{\partial x_4} & \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \frac{\partial f}{\partial x_3} \\ \frac{\partial h_1}{\partial x_4} & \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \frac{\partial h_1}{\partial x_3} \\ \frac{\partial h_2}{\partial x_4} & \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \frac{\partial h_2}{\partial x_3} \\ \frac{\partial h_3}{\partial x_4} & \frac{\partial h_3}{\partial x_1} & \frac{\partial h_3}{\partial x_2} & \frac{\partial h_3}{\partial x_3} \end{vmatrix}$$

$$A_5 \left(\frac{f, h_1, h_2, h_3}{x_5, x_1, x_2, x_3} \right) = \begin{vmatrix} \frac{\partial f}{\partial x_5} & \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \frac{\partial f}{\partial x_3} \\ \frac{\partial h_1}{\partial x_5} & \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \frac{\partial h_1}{\partial x_3} \\ \frac{\partial h_2}{\partial x_5} & \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \frac{\partial h_2}{\partial x_3} \\ \frac{\partial h_3}{\partial x_5} & \frac{\partial h_3}{\partial x_1} & \frac{\partial h_3}{\partial x_2} & \frac{\partial h_3}{\partial x_3} \end{vmatrix}$$

因此， $D - CA^{-1}B$ 行矩阵可写成

$$\left[A_4 \left(\frac{f, h_1, h_2, h_3}{x_4, x_1, x_2, x_3} \right) \quad A_5 \left(\frac{f, h_1, h_2, h_3}{x_5, x_1, x_2, x_3} \right) \right]$$

而 $[D - CA^{-1}B]dx_B = 0$ 给出

$$A_4 \left(\frac{f, h_1, h_2, h_3}{x_4, x_1, x_2, x_3} \right) dx_4 + A_5 \left(\frac{f, h_1, h_2, h_3}{x_5, x_1, x_2, x_3} \right) dx_5 = 0$$

这样，对于 n 维问题的一般情况， $df = (-CA^{-1}B + D)dx_B = 0$ 可以写成

$$\sum_{K=m+1}^n A_K \left(\frac{f, h_1, h_2, \dots, h_m}{x_K, x_1, x_2, \dots, x_m} \right) dx_K = 0 \quad (5-3)$$

考虑到 $dx_K (K = m+1, \dots, n)$ 是任意的微小向量，而 df 对所有的允许变量必须为零，所以由式 (5-3) 可得约束极值 (优化值) 存在的必要条件

$$A_K \left(\frac{f, h_1, h_2, \dots, h_m}{x_K, x_1, x_2, \dots, x_m} \right) = 0 \quad (K = m+1, \dots, n) \quad (5-4)$$

它共有 $n-m$ 个方程,当然不能解出 n 个变量 $x=(x_1, x_2, \dots, x_n)$ 。但是,还有 m 个约束方程 $h_K=0$ 可以利用。这样,总共有 n 个方程,从而可解出 n 个变量 $x_i (i=1, 2, \dots, n)$ 。由式(5-4)和 $h_K=0$ 解出的变量 x 就是既满足约束条件,又使目标函数成为极值的优化解 $x^*=(x_1^*, x_2^*, \dots, x_n^*)$ 。

为了明确,再把式(5-4)具体地复写如下

$$A_K \left(\begin{array}{c} f, h_1, h_2, \dots, h_m \\ x_K, x_1, x_2, \dots, x_m \end{array} \right) = \begin{array}{c} \left. \begin{array}{cccc} \frac{\partial f}{\partial x_K} & \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \dots & \frac{\partial f}{\partial x_m} \\ \frac{\partial h_1}{\partial x_K} & \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \dots & \frac{\partial h_1}{\partial x_m} \\ \frac{\partial h_2}{\partial x_K} & \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \dots & \frac{\partial h_2}{\partial x_m} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial h_m}{\partial x_K} & \frac{\partial h_m}{\partial x_1} & \frac{\partial h_m}{\partial x_2} & \dots & \frac{\partial h_m}{\partial x_m} \end{array} \right| = 0 \quad (5-4a) \\ (K=m+1, \dots, n) \end{array}$$

举例:

目标函数是 $f(x) = f(x_1, x_2, x_3) = 7x_1 - 6x_2 + 4x_3$, 约束条件是 $h_1(x) = x_1^2 + 2x_2^2 + 3x_3^2 - 1 = 0$; $h_2(x) = 5x_1 + 5x_2 - 3x_3 - 6 = 0$ 。求极小值。

这里 $n=3$, $m=2$ 。所以 $k=m+1=3$ 。因为 $n-m=1$, 则只有一个 $A_3 \left(\begin{array}{c} f, h_1, h_2 \\ x_3, x_1, x_2 \end{array} \right) = 0$

的条件。

由 $f(x)$ 和 $h_1(x)$, $h_2(x)$ 得

$$\begin{aligned} \frac{\partial f}{\partial x_3} = 4, \quad \frac{\partial f}{\partial x_1} = 7, \quad \frac{\partial f}{\partial x_2} = -6; \quad \frac{\partial h_1}{\partial x_3} = 6x_3, \quad \frac{\partial h_1}{\partial x_1} = 2x_1, \\ \frac{\partial h_2}{\partial x_3} = 4x_2, \quad \frac{\partial h_2}{\partial x_1} = -3, \quad \frac{\partial h_2}{\partial x_2} = 5, \quad \frac{\partial h_2}{\partial x_2} = 5. \end{aligned}$$

所以

$$A_3 \left(\begin{array}{c} f, h_1, h_2 \\ x_3, x_1, x_2 \end{array} \right) = \begin{array}{c} \left. \begin{array}{ccc} \frac{\partial f}{\partial x_3} & \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \\ \frac{\partial h_1}{\partial x_3} & \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} \\ \frac{\partial h_2}{\partial x_3} & \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} \end{array} \right| = \begin{array}{ccc} 4 & 7 & -6 \\ 6x_3 & 2x_1 & 4x_2 \\ -3 & 5 & 5 \end{array} = 4x_1 - 16x_2 - 390x_3 = 0 \end{array}$$

用它和 $h_1(x)=0$ 及 $h_2(x)=0$ 一起联立求解,得两组可能解,如表(5-1)所列

表 5-1

	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
解 1	0.947	0.207	-0.0772	5.08
解 2	0.534	0.535	-0.219	-0.346

下面说明在极值点 x^* 处

$$df = \sum_{i=1}^n \frac{\partial f}{\partial x_i} dx_i = \nabla f(x^*)^T dx = 0$$

和

$$dh_K = \sum_{i=1}^n \frac{\partial h_K}{\partial x_i} dx_i = \nabla h_K(x^*)^T dx = 0 \quad (K=1, 2, \dots, m)$$

两式的几何意义。

若等式约束 $h_K(x) = 0 (K=1, 2, \dots, m)$ 彼此间是线性无关的, 则这些等式约束条件在 n 维空间中就定义了一个 $n-m$ 维的超曲面。若它们在点 x^* 处是可微的话, 则它们的导数集就定义了一个 $n-m$ 维的切平面。例如, 当 $m=1$ 时, dx 只能在图 (5-1a) 所示的切平面内移动, 所以减少了一个自由度。 $m=2$ 时, 其切平面形成图 (5-1b) 所示的交

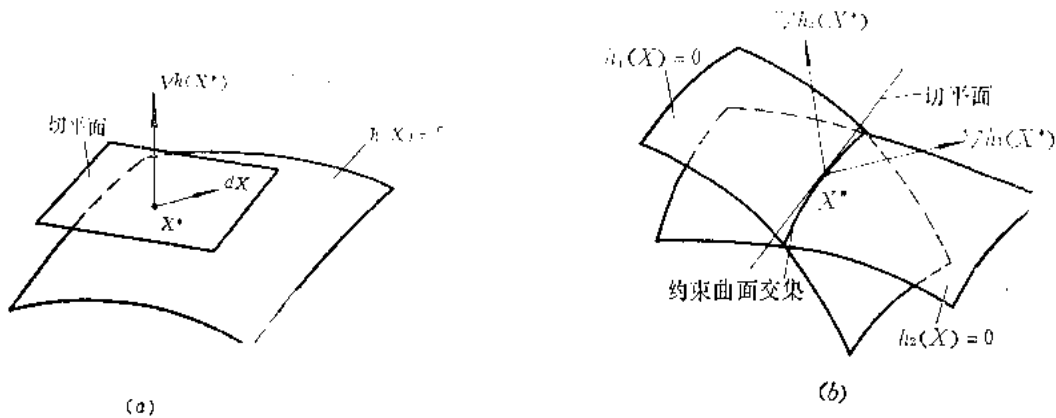


图 5-1

集(交线)。此时 dx 既要和 $\nabla h_1(x^*)$ 垂直, 又要和 $\nabla h_2(x^*)$ 垂直, 因而减少了两个自由度。

$dh_K = \nabla h_K(x^*)^T dx = 0$ 说明, 在极值点 x^* 处, dx 是和约束函数梯度 $\nabla h(x^*)$ 正交的。而 dx 应是目标函数 $f(x)$ 和约束条件 $h(x) = 0$ 交截的约束极值点 x^* 处的切平面中的向量。如果把 dx 看成是移动方向的话, 则 $df = \nabla f(x^*)^T dx = 0$ 说明, 在约束极值点 x^* 处, 沿着 dx 方向作微小移动时, 目标函数 $f(x)$ 的增量 $df(x)$ 应为零。

图 (5-2) 所示为具有 m 个约束条件的一般情况, x^* 是 $f(x)$ 和 $h_K(x) = 0$ 交截面上的一个极值点。 $\nabla h_K(x^*)$ 是 $h_K(x) = 0$ 在点 x^* 处的梯度向量。 dx 是在点 x^* 处切于约束曲面 $h_K(x) = 0$ 的诸切平面上的切向量。当 dx 是微量时, 则可认为此切向量是处在约束曲面 $h_K(x) = 0$ 之上的。

这样, 由 $df = \nabla f(x^*)^T dx = 0$ 和 $dh_K = \nabla h_K(x^*)^T dx = 0 (K=1, 2, \dots, m)$ 可知, 切向量 dx 和法向量 $\nabla h_K(x^*) (K=1, 2, \dots, m)$ 是正交的。而在约束极值点 x^* 处, 沿着所有切向量方向作微小移动时, $f(x^*)$ 值必须保持不变, 即增量 $df = 0$ 。

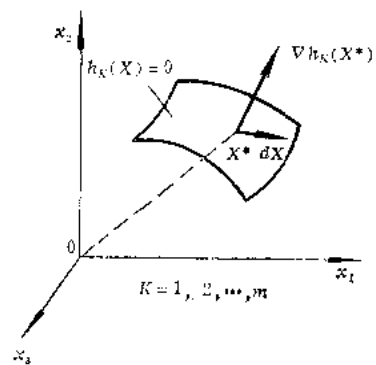


图 5-2

这说明, 对于受有 m 个约束 $h_K(x) = 0 (K=1, 2, \dots, m)$ 的目标函数 $f(x)$ 求极值的问题, 就是要求出这样的一点 x^* (极值点), 它满足 $h_K(x^*) = 0 (K=1, 2, \dots, m)$, 并对所有满足 $\nabla h_K(x^*)^T dx = 0 (K=1, 2, \dots, m)$ 的向量 dx 都有条件 $\nabla f(x^*)^T dx = 0$ 。显然, 这一约束极值点 x^* 是在 m 个约束的交集上。

二、拉格朗日乘子法

采用约束变量法求解约束优化问题，要考虑选取哪个或哪些变量（或它们的增量）进行消去运算，以及变量处理的均等性或对称性要求。当变量较多时，计算麻烦，甚至不可能求解。为了避免这些问题，可以采用拉格朗日乘子法。

拉格朗日乘子法又称条件极值法。下面简单说明这种方法的思路。

前面给出，在极值点 x^* 处有

$$\sum_{i=1}^n \frac{\partial f}{\partial x_i} dx_i = 0$$
$$\sum_{i=1}^n \frac{\partial h_K}{\partial x_i} dx_i = 0 \quad (K=1, 2, \dots, m)$$

把 m 个等式约束给出的 m 个 $\sum_{i=1}^n \frac{\partial h_K}{\partial x_i} dx_i = 0$ 分别乘以待定系数 $\lambda_K (K=1, 2, \dots, m)$ 再和 $\sum_{i=1}^n \frac{\partial f}{\partial x_i} dx_i = 0$ 相加，得

$$\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} + \lambda_1 \frac{\partial h_1}{\partial x_i} + \lambda_2 \frac{\partial h_2}{\partial x_i} + \dots + \lambda_m \frac{\partial h_m}{\partial x_i} \right) dx_i = 0 \quad (5-5)$$

可以通过其中的 m 个方程

$$\frac{\partial f}{\partial x_i} + \lambda_1 \frac{\partial h_1}{\partial x_i} + \lambda_2 \frac{\partial h_2}{\partial x_i} + \dots + \lambda_m \frac{\partial h_m}{\partial x_i} = 0 \quad (5-6)$$

来求解 m 个 $\lambda_1, \lambda_2, \dots, \lambda_m$ ，使得 m 个变量的微分 dx_1, dx_2, \dots, dx_m 的系数全为零。这样，则式(5-5)的等号左边就只剩下 $n-m$ 个变量的微分 $dx_{m+1}, dx_{m+2}, \dots, dx_n$ 的项，即它变成

$$\sum_{j=m+1}^n \left(\frac{\partial f}{\partial x_j} + \lambda_1 \frac{\partial h_1}{\partial x_j} + \lambda_2 \frac{\partial h_2}{\partial x_j} + \dots + \lambda_m \frac{\partial h_m}{\partial x_j} \right) dx_j = 0 \quad (5-7)$$

但 $dx_{m+1}, dx_{m+2}, \dots, dx_n$ 应是任意的量，则应有

$$\frac{\partial f}{\partial x_j} + \lambda_1 \frac{\partial h_1}{\partial x_j} + \lambda_2 \frac{\partial h_2}{\partial x_j} + \dots + \lambda_m \frac{\partial h_m}{\partial x_j} = 0 \quad (5-8)$$

$$(j=m+1, m+2, \dots, n)$$

式(5-6)和(5-8)及等式约束 $h_K(x) = 0 (K=1, 2, \dots, m)$ 就是点 x 达到约束极值的必要条件。

式(5-6)和(5-8)可以合并写成

$$\frac{\partial f}{\partial x_i} + \lambda_1 \frac{\partial h_1}{\partial x_i} + \lambda_2 \frac{\partial h_2}{\partial x_i} + \dots + \lambda_m \frac{\partial h_m}{\partial x_i} = 0 \quad (5-9)$$

$$(i=1, 2, \dots, n)$$

根据目标函数 $F(x)$ 存在极值的条件是 $\frac{\partial F}{\partial x_i} = 0 (i=1, 2, \dots, n)$ ，则上述函数的

约束极值条件可以转换成无约束的函数极值条件。办法是，把原来的目标函数 $f(x)$ 改造成如下形式的新的目标函数。

$$F(x, \lambda) = f(x) + \sum_{K=1}^m \lambda_K h_K(x) \quad (5-10)$$

式中 $h_K(x)$ 就是原目标函数 $f(x)$ 的等式约束条件, 而待定系数 λ_K 称为拉格朗日乘子, $F(x, \lambda)$ 称为拉格朗日函数。这种方法称为拉格朗日乘子法。

在式 (5-10) 中显然多出了 m 个待定系数 λ_K (可看成是新的变量), 而 $x = (x_1, x_2, \dots, x_n)$ 有 n 个变量。结果共有 $n+m$ 个变量。但是 $\frac{\partial F}{\partial x_i} = 0$ ($i=1, 2, \dots, n$) 可提供 n 个方程, 再加上 m 个等式约束条件 $h_K(x) = 0$, 共有 $n+m$ 个方程, 完全足以解出这 $n+m$ 个变量。

由于 $\frac{\partial F}{\partial \lambda_K} = 0$ 给出 $h_K(x) = 0$, 所以这 $n+m$ 个方程可以看成是通过下述条件给出的

$$\begin{aligned} \frac{\partial F}{\partial x_i} &= 0 \quad (i=1, 2, \dots, n) \\ \frac{\partial F}{\partial \lambda_K} &= 0 \quad (K=1, 2, \dots, m) \end{aligned}$$

这样, 拉格朗日乘子法可以叙述如下。

设 $x = (x_1, x_2, \dots, x_n)^T$, 目标函数是 $f(x)$, 约束条件是 $h_K(x) = 0$ ($K=1, 2, \dots, m$) 的 m 个等式约束方程。为了求出 $f(x)$ 的可能的极值点 $x^* = (x_1^*, x_2^*, \dots, x_n^*)^T$, 引入拉格朗日乘子 λ_K ($K=1, 2, \dots, m$), 并构成一个新的目标函数

$$F(x, \lambda) = f(x) + \sum_{K=1}^m \lambda_K h_K(x)$$

把 $F(x, \lambda)$ 做为一个新的无约束条件的目标函数来求解它的极值点, 所得结果就是在满足约束条件 $h_K(x) = 0$ ($K=1, 2, \dots, m$) 的原目标函数 $f(x)$ 的极值点。自 $F(x, \lambda)$ 具有极值的必要条件

$$\begin{aligned} \frac{\partial F}{\partial x_i} &= 0 \quad (i=1, 2, \dots, n) \\ \frac{\partial F}{\partial \lambda_K} &= 0 \quad (K=1, 2, \dots, m) \end{aligned}$$

可得 $m+n$ 个方程, 从而解得 $x = (x_1, x_2, \dots, x_n)^T$ 和 λ_K ($K=1, 2, \dots, m$) 共 $m+n$ 个未知变量的值。由上述方程组求得的 $x^* = (x_1^*, x_2^*, \dots, x_n^*)^T$ 是函数 $f(x)$ 极值点的坐标值。

按照式 (5-9) 给出的条件, 拉格朗日乘子法也可以用另一种方式叙述如下:

设 x^* 是目标函数 $f(x)$ 在等式约束 $h_K(x) = 0$ 条件下的一个局部极值点, 而且在该点处各约束函数的梯度 $\nabla h_K(x^*)$ ($K=1, 2, \dots, m$) 是线性无关的 (符合此条件的点称为正则点), 则存在一个向量 λ (在 m 个约束函数规定的集内), 使得下式成立

$$\nabla F = \nabla f(x^*) + \lambda^T \nabla h(x^*) = 0 \quad (5-11)$$

式中 $\lambda^T = [\lambda_1, \lambda_2, \dots, \lambda_m]$, $\nabla h(x^*)^T = [\nabla h_1(x^*), \nabla h_2(x^*), \dots, \nabla h_m(x^*)]$ 。

为了说明拉格朗日乘子的物理意义, 我们看函数 $f(x) = f(x_1, x_2)$ 的一个二维问题,

且只有一个约束条件 $h(x) = h(x_1, x_2) = 0$ 时的简单情况。此时式(5-10)的形式是

$$F(x, \lambda) = f(x) + \lambda h(x)$$

由式(5-9)得

$$\begin{aligned} \frac{\partial f}{\partial x_1} + \lambda \frac{\partial h}{\partial x_1} = 0 \quad \text{或} \quad \lambda = - \frac{\partial f}{\partial x_1} / \frac{\partial h}{\partial x_1} \\ \frac{\partial f}{\partial x_2} + \lambda \frac{\partial h}{\partial x_2} = 0 \quad \text{或} \quad \lambda = - \frac{\partial f}{\partial x_2} / \frac{\partial h}{\partial x_2} \end{aligned}$$

所以, 可以写出

$$\lambda = - \frac{\frac{\partial f}{\partial x_i}}{\frac{\partial h}{\partial x_i}} \quad (i = 1, 2)$$

式中 $\frac{\partial f}{\partial x_i}$ 是单位变量的目标值变化率, 而 $-\frac{\partial h}{\partial x_i}$ 则是单位变量的约束值变化率。可以称

$\frac{\partial f}{\partial x_i} / \frac{\partial h}{\partial x_i}$ 为优化效率或敏感度系数。而且从 $\lambda = - \frac{\partial f}{\partial x_1} / \frac{\partial h}{\partial x_1} = - \frac{\partial f}{\partial x_2} / \frac{\partial h}{\partial x_2}$ 可知, 各变量改变导致的优化效率是相等的, 且等于一个常数 λ 。

对于结构优化设计问题, 若目标函数 $f(x)$ 是结构重量, 约束条件是结构刚度或某点的变形, 则 $\frac{\partial f}{\partial x_i}$ 可以理解为结构重量的收益, 而 $-\frac{\partial h}{\partial x_i}$ 可以理解为结构刚度的支出。则

$\lambda = - \frac{\partial f}{\partial x_i} / \frac{\partial h}{\partial x_i}$ 就意味着付出单位的结构刚度支出所能获得的结构重量收益。这时的 λ 就反映结构刚度对其重量的优化效率。

举例: 用拉格朗日乘子法计算在约束条件 $h(x_1, x_2) = 2x_1 + 3x_2 - 6 = 0$ 的情况下, 目标函数 $f(x_1, x_2) = 4x_1^2 + 5x_2^2$ 的极值点坐标。

改造的目标函数是 $F(x, \lambda) = 4x_1^2 + 5x_2^2 + \lambda(2x_1 + 3x_2 - 6)$, 则 $\frac{\partial F}{\partial x_1} = 8x_1 + 2\lambda = 0$,

$$\frac{\partial F}{\partial x_2} = 10x_2 + 3\lambda = 0, \quad \frac{\partial F}{\partial \lambda} = 2x_1 + 3x_2 - 6 = 0。$$

自 $\frac{\partial F}{\partial x_1}$ 和 $\frac{\partial F}{\partial x_2}$ 等于零两式解得极值点坐标是

$$x_1 = -\frac{1}{4}\lambda \quad x_2 = -\frac{3}{10}\lambda$$

把它们代入 $\frac{\partial F}{\partial \lambda} = 0$ (即约束条件 $2x_1 + 3x_2 - 6 = 0$) 中去, 得 $2\left(-\frac{1}{4}\lambda\right) + 3\left(-\frac{3}{10}\lambda\right) - 6 = 0$, 它给出 $\lambda = -\frac{30}{7}$ 。所以得

$$x_1 = \frac{1}{4} \cdot \frac{30}{7} = 1.071 \quad x_2 = \frac{3}{10} \cdot \frac{30}{7} = 1.286$$

即极值点 x^* 坐标是 $x_1^* = 1.071$, $x_2^* = 1.286$ 。

§ 2 库恩—塔克(Kuhn—Tucker)条件

可以把拉格朗日乘子法推广应用于具有不等式约束的优化问题中去。为此,用引入人为变量 x_{n+K} 的办法把不等式约束 $g_K(x) \leq 0$ 转换成等式约束 $x_{n+K}^2 + g_K(x) = 0$ 。这样,拉格朗日函数可写成

$$F(x, x_{n+K}, \mu) = f(x) + \sum_{K=1}^m (x_{n+K}^2 + g_K(x)) \mu_K \quad (5-12)$$

式中 μ_K 是对应于不等式约束 $g_K(x) \leq 0$ 的拉格朗日乘子。

由式(5-12)可得,在极值点处有

$$\frac{\partial F}{\partial x_i} = \frac{\partial f}{\partial x_i} + \sum_{K=1}^m \mu_K \frac{\partial g_K}{\partial x_i} = 0 \quad (i = 1, 2, \dots, n) \quad (5-13)$$

$$\frac{\partial F}{\partial x_{n+K}} = 2\mu_K x_{n+K} = 0 \quad (K = 1, 2, \dots, m) \quad (5-14)$$

$$\frac{\partial F}{\partial \mu_K} = (x_{n+K}^2 + g_K(x)) = 0 \quad (K = 1, 2, \dots, m) \quad (5-15)$$

式(5-13)具有和等式约束时的式(5-11)相同含意。式(5-15)是保证不等式 $g_K(x) \leq 0$ 成立的。式(5-14)说明,可能有两种情况: $\mu_K \neq 0$, $x_{n+K} = 0$ 或 $\mu_K = 0$, $x_{n+K} \neq 0$ 。当 $x_{n+K} = 0$ 时,优化解必须在约束边界上求出,因而这个约束条件是起作用的。而当 $x_{n+K} \neq 0$ 时,优化解则位于约束边界的内部,因而这个约束条件是不起作用的。或者换一种说法,就是在一个可行点 x 处,对于一个不等式约束 $g_K(x) \leq 0$,如果它的作用只相当于一个等式约束,即 $g_K(x) = 0$,则说它是在该可行点处起作用的约束;而如果它的作用是 $g_K(x) < 0$,就说它是在点 x 处不起作用的约束。因此,只有 $\mu_K \neq 0$ 所对应的约束条件是起作用的。显然,任何一个等式约束 $h_K(x) = 0$ 都是在该可行点处起作用的约束。

在可行点 x 处,起作用的约束在 x 的邻域内限制了可行性范围,而其余的(不起作用的)约束在 x 的邻域内就没有影响。

这样,式(5-13)中就不应包括不起作用的约束条件了。因此,我们定义一个以整数为元素的集合 J ,它由该试验点 x 处那些起作用约束的下标所组成,即 $g_j(x) = 0$, $j \in J$ 。那么,就可以把具有不等式约束时的拉格朗日乘子法叙述为:

欲求极小值的目标函数是 $f(x)$,它具有约束 $g_j(x) \leq 0$, ($j = 1, 2, \dots, m$)。若 x^* 是优化点,并存在 $\mu_j \geq 0$,使得

$$\frac{\partial f}{\partial x_i} + \sum_{j \in J} \mu_j \frac{\partial g_j}{\partial x_i} = 0 \quad (i = 1, 2, \dots, n) \quad (5-16)$$

则 x^* 可能就是一个约束极小点。

上式通常称为Kuhn—Tucker条件(有时简写成K—T条件)。可见Kuhn—Tucker条件是拉格朗日乘子法应用于不等式约束优化问题的结果。因而Kuhn—Tucker条件可叙述为:

设 x^* 是问题

$$\begin{aligned} & \min f(x) \\ \text{s.t. } & h_K(x) = 0 \quad (K=1, 2, \dots, l) \\ & g_j(x) \leq 0 \quad (j=1, 2, \dots, m-l) \end{aligned}$$

的相对极小值点，且假定 x^* 是该约束集的正则点，则存在一个向量 $\lambda (\in R^l)$ 和另一个向量 $\mu (\in R^{m-l}, \mu \geq 0)$ 使得下式成立

$$\nabla F = \nabla f(x^*) + \lambda^T \nabla h(x^*) + \mu^T \nabla g(x^*) = 0 \quad (5-17)$$

$$\mu^T g(x^*) = 0 \quad (5-18)$$

式中 $\mu^T = [\mu_1, \mu_2, \dots, \mu_{m-l}]$ ， $\nabla g(x^*)^T = [\nabla g_1, \nabla g_2, \dots, \nabla g_{m-l}]$ 。

注意， $\nabla g(x^*)$ 是只考虑起作用的不等式约束。 $\mu \geq 0$ 的非负要求，是从 x^* 是正则点，因而要求 $\nabla h_K(x^*)$ 和 $\nabla g_j(x^*)$ 应是线性无关的条件考虑的。同时也是为了保证 $-\nabla f(x^*)$ 是诸向量 $\nabla h_K(x^*)$ 和 $\nabla g_j(x^*)$ 的线性组合。而从几何上解释，则是从 $\nabla f(x^*)$ 应在 $\nabla h(x^*)$ 和 $\nabla g(x^*)$ 所形成的角锥之内的条件考虑的。

从上面的讨论可知，Kuhn-Tucker条件除了能提供求优化解的线索之外，还能用来检验求得解是否是相对极小值点。

式 (5-16) 的几何含义，在上面已提到两次了。把式 (5-16) 改写成

$$-\frac{\partial f}{\partial x_i} = \sum_{j \in I} \mu_j \frac{\partial g_j}{\partial x_i} \quad (i=1, 2, \dots, n) \quad (5-19)$$

它表明，在约束极小值点 x^* 处，函数 $f(x)$ 的负梯度一定能表示成所有起作用约束在该点梯度（法向量）的非负线性组合。

下面以二维问题为例，说明其几何意义。

图 (5-3) 是考虑 $g_1(x)$ 和 $g_2(x)$ 两个约束都起作用的情况，以图 (5-4b) 所示的情况为基础，考虑在点 x^K 处目标函数的负梯度 $-\nabla f(x^K)$ 时的图形。如果点 x^K 是极值点，则式 (5-16) 的条件要求点 x^K 一定要落在 $g_1(x)$ 和 $g_2(x)$ 的交线上，而且 $-\nabla f(x^K)$ 和 $\nabla g_1(x^K)$ 及 $\nabla g_2(x^K)$ 应该

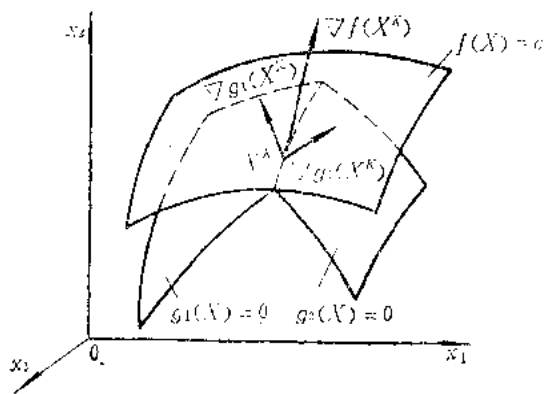


图 5-3

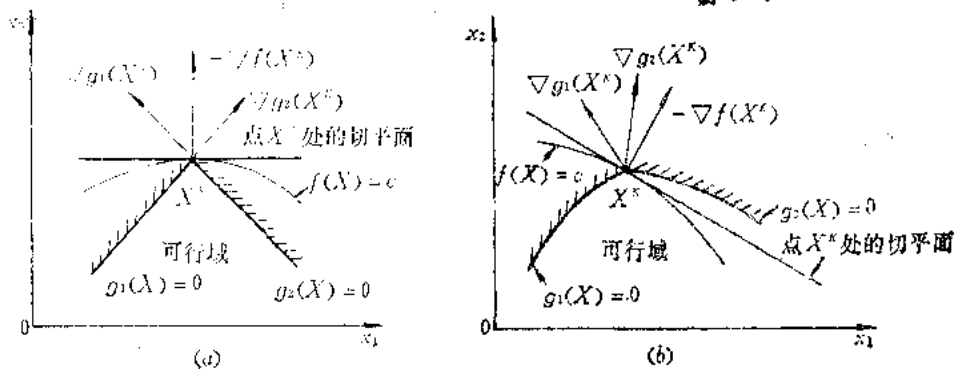


图 5-4

线性相关，即三者应该共面。

图(5—4a)是在点 x^K 处的截面图形。过点 x^K 作目标函数的负梯度 $-\nabla f(x^K)$ ，它垂直于目标函数 $f(x)$ 的等值线，且指向目标函数值的最速减小方向。再作约束函数的梯度 $\nabla g_1(x^K)$ 和 $\nabla g_2(x^K)$ ，它们分别垂直于 $g_1(x)=0$ 和 $g_2(x)=0$ 二曲线并形成一锥形夹角区域。此时可能出现两种情况。

第一， $-\nabla f(x^K)$ 落在 $\nabla g_1(x^K)$ 和 $\nabla g_2(x^K)$ 所张成的锥角区外的一侧，如图(5—4b)所示。这时，当过点 x^K 作出与 $-\nabla f(x^K)$ 垂直的切平面，并从 x^K 出发向此切平面的 $-\nabla f(x^K)$ 所在一侧移动时，目标函数值可以减小。但是，这一侧有一部分区域是可行域(在图中，这样的区域是由 $f(x)$ 和 $g_2(x)$ 形成的)。结果是既可减小目标函数值，又不破坏约束条件，这说明 x^K 仍可沿约束曲面移动而不致破坏约束条件，且目标函数值还能够得到改变(减小)。所以点 x^K 不是稳定的最优点，即它不是约束最优点或局部极值点。

第二， $-\nabla f$ 落在 ∇g_1 和 ∇g_2 张成的锥角之内，如图(5—4a)所示。此时，作出和 $-\nabla f$ 垂直的过 x^K 的目标函数等值面的切平面，把空间分成两个区域。当从 x^K 出发向包含 $-\nabla f$ 的一侧移动时，将可使目标函数值减小。但这一侧的任何一点都不落在可行区域内。显然，此时的点 x^K 就是约束最优点或局部极值点 x^* 。沿此点再作任何移动都将破坏约束条件，故它是稳定点。

由于 $-\nabla f(x^*)$ 和 $\nabla g_1(x^*)$ ， $\nabla g_2(x^*)$ 在一个平面内，则前者可看成是后两者的线性组合。又因 $-\nabla f(x^*)$ 处于 $\nabla g_1(x^*)$ 和 $\nabla g_2(x^*)$ 的夹角之间，所以线性组合的系数为正，即有

$$-\nabla f(x^*) = \mu_1 \nabla g_1(x^*) + \mu_2 \nabla g_2(x^*)$$

其中 $\mu_1 > 0$ ， $\mu_2 > 0$ 。

这就是目标函数在两个起作用的约束条件下，使 x^* 成为条件极值点的必要条件。

当约束条件有三个时，则要求 $-\nabla f(x^*)$ 处于 $\nabla g_1(x^*)$ 、 $\nabla g_2(x^*)$ 和 $\nabla g_3(x^*)$ 形成的角锥之内。

推广到 n 维优化问题，当具有 m 个起作用约束条件时，就得到式(5—19)，即

$$\frac{\partial f}{\partial x_i} + \sum_{j \in J} \mu_j \frac{\partial g_j}{\partial x_i} = 0 \quad (i = 1, 2, \dots, n)$$

的几何意义的说明。

下面再简单说明Kuhn—Tucker乘子 μ 的计算。

如果记

$$-\frac{\partial f}{\partial x_i} = b_i \quad (i = 1, 2, \dots, n) \quad (5-20)$$

或

$$-\nabla f = \mathbf{b} \quad (5-20a)$$

而

$$\frac{\partial g_j}{\partial x_i} = \nabla g_{ij} = a_{ij} \quad (i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m) \quad (5-21)$$

和

$$\nabla g_i = \mathbf{A}_i = [a_{1i}, a_{2i}, \dots, a_{ni}]^T$$

并记

$$\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_m] = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix}$$

则

$$\frac{\partial f}{\partial x_i} + \sum_{j \in J} \mu_j \frac{\partial g_j}{\partial x_i} = 0 \quad (i = 1, 2, \dots, n)$$

可以写成

$$\sum_{j=1}^m a_{ij} \mu_j = b_i \quad (i = 1, 2, \dots, n) \quad (5-22)$$

或

$$\mathbf{A}\boldsymbol{\mu} = \mathbf{b} \quad (5-22a)$$

一般地说, 在工程问题的最优解处, 起作用约束的个数小于 n , 即 $m < n$ 。所以方程组是超定的。根据最小二乘法的原则, 只要 $\mathbf{A}^T \mathbf{A}$ 是非奇异的, 则解可能是

$$\boldsymbol{\mu} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (5-23)$$

把它代入 (5-22a), 得

$$\mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} = \mathbf{b}$$

或

$$[\mathbf{I} - \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T] \mathbf{b} = \mathbf{0} \quad (5-24)$$

如果求得的 μ 都是非负的, 且满足式 (5-24) 的条件, 就说明解是满足 K-T 条件的极小值解。

库恩-塔克 (K-T) 条件应用举例

若给定优化问题的数学模型为

$$\begin{aligned} f(\mathbf{x}) &= (x_1 - 2)^2 + x_2^2 \rightarrow \min \\ \text{s.t.} \quad g_1(\mathbf{x}) &= x_1^2 + x_2 - 1 \leq 0 \\ g_2(\mathbf{x}) &= -x_2 \leq 0 \\ g_3(\mathbf{x}) &= -x_1 \leq 0 \end{aligned}$$

利用 K-T 条件确定极值点 \mathbf{x}^* 。

此问题在设计空间 x_1-x_2 平面上的图形如图 (5-5) 所示。它的 K-T 条件可表示为

$$\begin{aligned} \frac{\partial f(\mathbf{x}^*)}{\partial x_i} + \sum_{j \in J} \lambda_j \frac{\partial g_j(\mathbf{x}^*)}{\partial x_i} &= 0 \quad (i = 1, 2) \\ g_j(\mathbf{x}^*) &= 0 \quad (j \in J) \\ \lambda_j &\geq 0 \quad (j \in J) \end{aligned}$$

其中 J 为在 \mathbf{x}^* 处起作用约束下标的集合, 因 \mathbf{x}^* 待求, 所以 J 未知, 只能根据各种可能情况进行试验。现按八种情况分析如下:

1. 若 g_1, g_2, g_3 三个约束都在 \mathbf{x}^* 处起作用, 则 K-T 条件中的第一个方程可写成

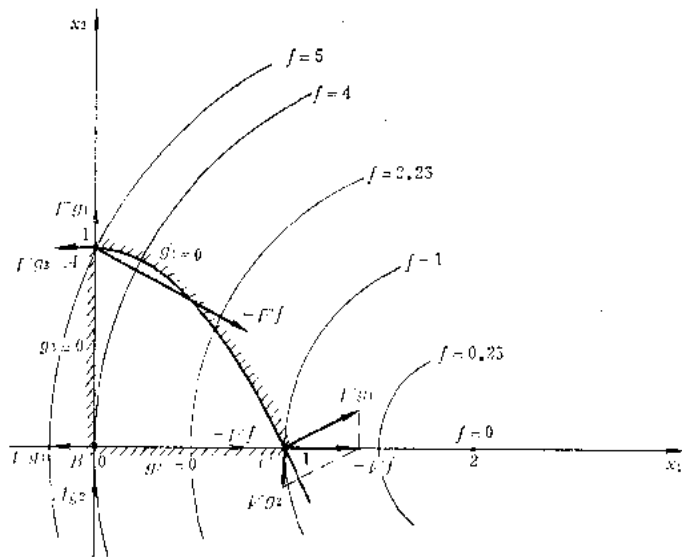


图 5-5

$$\begin{aligned} \frac{\partial f(\mathbf{x}^*)}{\partial x_1} + \lambda_1 \frac{\partial g_1(\mathbf{x}^*)}{\partial x_1} + \lambda_2 \frac{\partial g_2(\mathbf{x}^*)}{\partial x_1} + \lambda_3 \frac{\partial g_3(\mathbf{x}^*)}{\partial x_1} &= 0 \\ \frac{\partial f(\mathbf{x}^*)}{\partial x_2} + \lambda_1 \frac{\partial g_1(\mathbf{x}^*)}{\partial x_2} + \lambda_2 \frac{\partial g_2(\mathbf{x}^*)}{\partial x_2} + \lambda_3 \frac{\partial g_3(\mathbf{x}^*)}{\partial x_2} &= 0 \end{aligned}$$

将 f, g_1, g_2, g_3 的具体表达式代入, 得

$$\begin{aligned} 2(x_1^* - 2) + 2\lambda_1 x_1^* - \lambda_3 &= 0 \\ 2x_2^* + \lambda_1 - \lambda_2 &= 0 \end{aligned}$$

三个起作用约束在 \mathbf{x}^* 处取等式形式, 有

$$\begin{aligned} g_1(\mathbf{x}^*) &= (x_1^*)^2 + x_2^* - 1 = 0 \\ g_2(\mathbf{x}^*) &= -x_2^* = 0 \\ g_3(\mathbf{x}^*) &= -x_1^* = 0 \end{aligned}$$

这里是三个方程, 两个未知数, 属矛盾方程组, 无解。所以不存在三个起作用约束的极值点。

2. 若 g_1, g_3 在 \mathbf{x}^* 处为起作用约束, K-T 条件为

$$\begin{aligned} 2(x_1^* - 2) + 2\lambda_1 x_1^* - \lambda_3 &= 0 \\ 2x_2^* + \lambda_1 &= 0 \\ g_1(\mathbf{x}^*) &= (x_1^*)^2 + x_2^* - 1 = 0 \\ g_3(\mathbf{x}^*) &= -x_1^* = 0 \end{aligned}$$

从约束方程中解得 $x_1^* = 0, x_2^* = 1$, 相当于图 (5-5) 中的 A 点。代回第一方程组, 得

$$\begin{aligned} \lambda_1 &= -2 < 0 \\ \lambda_3 &= -4 < 0 \end{aligned}$$

不满足非负要求, 所以 A 点不是极值点。

3. 若 g_2, g_3 在 \mathbf{x}^* 处为起作用约束, K-T 条件为

$$2(x_1^* - 2) - \lambda_3 = 0$$

$$\begin{aligned} 2x_2^* - \lambda_2 &= 0 \\ g_2(x^*) &= -x_2^* = 0 \\ g_3(x^*) &= -x_1^* = 0 \end{aligned}$$

从约束方程中解得 $x_1^* = 0$, $x_2^* = 0$, 相当于图 (5-5) 中的 B 点。代回第一方程组, 得

$$\begin{aligned} \lambda_2 &= 0 \\ \lambda_3 &= -4 < 0 \end{aligned}$$

同样不满足非负要求, 所以 B 点也不是极值点。

4. 若 g_1 , g_2 在 x^* 处为起作用约束, K-T 条件为

$$\begin{aligned} 2(x_1^* - 2) + 2\lambda_1 x_1^* &= 0 \\ 2x_2^* + \lambda_1 - \lambda_2 &= 0 \\ g_1(x^*) &= (x_1^*)^2 + x_2^* - 1 = 0 \\ g_2(x^*) &= -x_2^* = 0 \end{aligned}$$

从约束方程中解得 $x_1^* = \pm 1$, $x_2^* = 0$ 。但 $x_1^* = -1$ 不满足第三个不等式约束条件 $g_3(x^*) = -x_1^* < 0$, 故取 $x_1^* = 1$, $x_2^* = 0$, 相当于图 (5-5) 中的 C 点。代回第一方程组, 得

$$\begin{aligned} \lambda_1 &= 1 \geq 0 \\ \lambda_2 &= 1 \geq 0 \end{aligned}$$

满足非负要求, 这样 C 点满足全部 K-T 条件, 所以 C 点为极值点。

5. 若只有 g_1 一个约束在 x^* 处起作用, K-T 条件可写成

$$\begin{aligned} 2(x_1^* - 2) + 2\lambda_1 x_1^* &= 0 \\ 2x_2^* + \lambda_1 &= 0 \\ g_1(x^*) &= (x_1^*)^2 + x_2^* - 1 = 0 \end{aligned}$$

从第一方程组解得 $x_1^* = \frac{2}{1 + \lambda_1}$, $x_2^* = -\frac{\lambda_1}{2}$ 。因为假定只有一个约束 g_1 在 x^* 处起作用,

那么第二个约束在 x^* 处不起作用, 有 $g_2(x^*) = -x_2^* < 0$ 即 $x_2^* > 0$ 。根据 $x_2^* = -\frac{\lambda_1}{2} > 0$ 有

$\lambda_1 < 0$, 不满足非负要求, 故此点不是极值点。

6. 若只有 g_2 一个约束在 x^* 处起作用, K-T 条件为

$$\begin{aligned} 2(x_1^* - 2) &= 0 \\ 2(x_2^* - \lambda_2) &= 0 \\ g_2(x^*) &= -x_2^* = 0 \end{aligned}$$

解上方程组, 得 $x_1^* = 2$, $x_2^* = 0$, $\lambda_2 = 0$ 。此解不满足 $g_1(x^*) < 0$ 的要求, 故此点不是极值点。

7. 若只有 g_3 一个约束在 x^* 处起作用, K-T 条件为

$$\begin{aligned} 2(x_1^* - 2) - \lambda_3 &= 0 \\ 2x_2^* &= 0 \\ g_3(x^*) &= -x_1^* = 0 \end{aligned}$$

解上方程组, 得 $x_1^* = 0$, $x_2^* = 0$, $\lambda_3 = -4 < 0$, 不满足非负要求, 故此点不是极值点。

8. 若 g_1 , g_2 , g_3 在 x^* 处都不起作用, K-T 条件为

$$2(x_1^* - 2) = 0$$

$$2x_2^* = 0$$

解得 $x_1^* = 2, x_2^* = 0$, 此点不满足 $g_1(x^*) < 0$ 要求, 故不是极值点。

从上述八种情况的分析可以看出, 利用 K—T 条件求极值点往往是很繁琐的, 需要确定哪些约束在极值点处起作用。

库恩—塔克 (K—T) 条件也可以叙述为在极值点处目标函数的负梯度为起作用的各约束函数梯度的非负线性组合, 即

$$-\nabla f(x^*) = \sum_{j \in J} \lambda_j \nabla g_j(x) \quad \text{其中 } \lambda_j \geq 0$$

在极值点 C 处, 起作用约束为 g_1, g_2 , 则应有

$$-\nabla f(x^*) = \lambda_1 \nabla g_1(x^*) + \lambda_2 \nabla g_2(x^*)$$

而

$$\nabla f(x^*) = \begin{Bmatrix} 2(x_1^* - 2) \\ 2x_2^* \end{Bmatrix} \Big|_{\substack{x_1^* = 2 \\ x_2^* = 0}} = \begin{Bmatrix} -2 \\ 0 \end{Bmatrix}$$

$$\nabla g_1(x^*) = \begin{Bmatrix} 2x_1^* \\ 1 \end{Bmatrix} \Big|_{\substack{x_1^* = 2 \\ x_2^* = 0}} = \begin{Bmatrix} 4 \\ 1 \end{Bmatrix}$$

$$\nabla g_2(x^*) = \begin{Bmatrix} 0 \\ -1 \end{Bmatrix}$$

代入上式得

$$-\begin{Bmatrix} -2 \\ 0 \end{Bmatrix} = \lambda_1 \begin{Bmatrix} 4 \\ 1 \end{Bmatrix} + \lambda_2 \begin{Bmatrix} 0 \\ -1 \end{Bmatrix}$$

要求 $\lambda_1 = \lambda_2 = 1$

所以 C 点处目标函数的负梯度 $-\nabla f$ 为起作用约束函数梯度 $\nabla g_1, \nabla g_2$ 的非负线性组合, 如图 (5—5) 所示, 而在 A, B 两点均不满足上述条件。

§ 3 制约函数法

制约函数法在形式上和拉格朗日乘子法相似, 因此可以看成是它的推广。它的思路很简单, 即并不直接去解原有约束的优化问题

$$\begin{aligned} & f(x) \rightarrow \min \\ \text{s.t.} \quad & h_i(x) = 0 \quad (i = 1, 2, \dots, l) \\ & g_j(x) \leq 0 \quad (j = 1, 2, \dots, p) \quad (l + p = m) \end{aligned}$$

而是把它变成如下形式的无约束优化问题

$$F(x, r_K, M_K) = f(x) + r_K \sum_{j=1}^p G[g_j(x)] + M_K \sum_{i=1}^l h_i(x)^2 \quad (5-25)$$

其中 $G[g_j(x)]$ 和 $h_i(x)^2$ 诸项要求具有这样的特性, 当约束条件不满足时, 函数 $F(x, r_K, M_K)$ 将受到制约; 当约束条件满足时, 它就不受制约。因此, 可称函数 $F(x, r_K, M_K)$ 为制约函数, 而 $G[g_j(x)]$ 和 $h_i(x)^2$ 为制约项。制约函数中的因子 r_K 和 M_K 起着在目标函数

$f(x)$ 和约束项之间进行加权的作用。有人称它们为响应系数。所以也就相应地称函数 $F(x, r_K, M_K)$ 所代表的曲面为响应曲面。 r_K 和 M_K 是一个可以由小到大或由大到小改变的序列, 因此式(5—25)是一个无约束优化问题的无穷序列。对它可以采用某种无约束极小化方法求出无约束的极小解, 它也就是约束极小化问题的解。因此, 这种方法又称为序列无约束极小化方法, 简称为 SUMT (是 Sequential Unconstrained Minimization Technique 的缩写) 方法。

在式(5—25)形式的约束函数中, 和不等式约束对应的约束项 $G[g_j(x)]$ 有多种形式, 目前常用的有两类, 一类是“惩罚项”, 另一类是“障碍项”。这样, 只包含有惩罚项的约束函数 $F(x, r_K)$ 又可称为惩罚函数; 而只包含有障碍项的约束函数 $F(x, r_K)$ 就称为障碍函数。

下面先介绍只具有不等式约束的优化问题的两类方法, 即求解惩罚函数 $F(x, r_K)$ 极小值的外点法和求解障碍函数 $F(x, r_K)$ 极小值的内点法, 它们又可统称为罚函数法。

一、外点法

外点法的基本思想是: 采用把不等式约束写成惩罚项的形式, 加权后再加到目标函数中去的方法形成约束函数。

对惩罚项有以下要求:

- 1) 它在所论空间上必须是连续函数;
- 2) 当点 x 不在可行域内时, 其值大于零; 而在可行域上时, 它的值等于零。

符合上述要求的惩罚项可以取为 $\{\max[0, g_j(x)]\}^2$ 或其它形式。因此, 外点法的约束函数形式可以写成

$$F(x, r_K) = f(x) + r_K \sum_{j=1}^p \{\max[0, g_j(x)]\}^2 \quad (5-26)$$

其中加权因子 r_K 是一个可以由小到大改变的序列正的常数, 称为罚因子。

从对惩罚项的要求可以看出, 它的作用是, 当约束条件不满足时, 惩罚项 $r_K \sum_{j=1}^p g_j(x)^2 > 0$, 从而 $F(x, r_K) > f(x)$ 。这就反映了对不满足约束条件的一种惩罚。而增大罚因子 r_K 的值也就是增大惩罚项的作用。当 r_K 增大到某一程度时, 函数 $F(x, r_K)$ 中惩罚项 $r_K \sum_{j=1}^p g_j(x)^2$ 与目标函数 $f(x)$ 相比起着决定性的作用。当约束条件得到满足时, 则总有 $r_K \sum_{j=1}^p g_j(x)^2 = 0$, 从而有 $F(x, r_K) = f(x)$, 即不受到惩罚。这就是惩罚项和惩罚函数的含义。

例如, 对于 $f(x) = \frac{x}{2}$ (它是一条直线), 约束条件是 $1 - x \leq 0$ 的简单问题, 它的惩罚函数是

$$F(x, r_K) = \frac{x}{2} + r_K \{\max[0, (1-x)]\}^2$$

它表示一条抛物线。

r_K 为 10、20、30、40 等不同数值的情况如图(5—6)所示。随着 r_K 值的增大,

越来越接近真实的极小点。一旦进入约束边界， $F(x, r_K)$ 又变成原目标函数 $f(x)$ 。

看来好像 r_K 值越大越好，最好一开始就把 r_K 选得足够大，使 $F(x, r_K)$ 的极小点接近由约束曲面规定的可行域。但随着 r_K 的增大， $F(x, r_K)$ 的极小化越困难。为了说明这一点，看一个两杆桁架的约束优化问题（这里略去它的数学模型）的情况。图（5—7 a, b, c, d）是当顺次取 $r_K = 10^{-10}, 10^{-9}, 10^{-8}$ 和 10^{-7} 时的情形。图中的实线是函数 $F(x, r_K)$ 的等值线，虚线是约束边界。

可以看出，当 $r_K = 10^{-7}$ 时，可行极小值点 x^* 很接近约束极小值点，但和 $r_K = 10^{-10}, 10^{-9}, 10^{-8}$ 的图形相比，函数 $F(x, r_K)$ 的等值线图形显得严重的畸变或偏心。

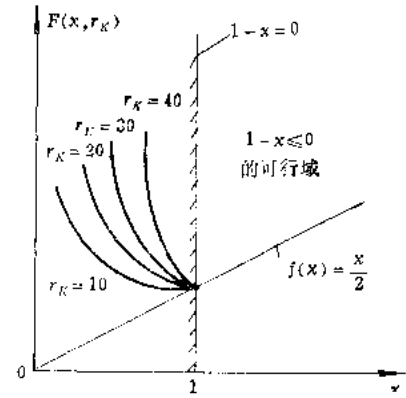


图 5—6

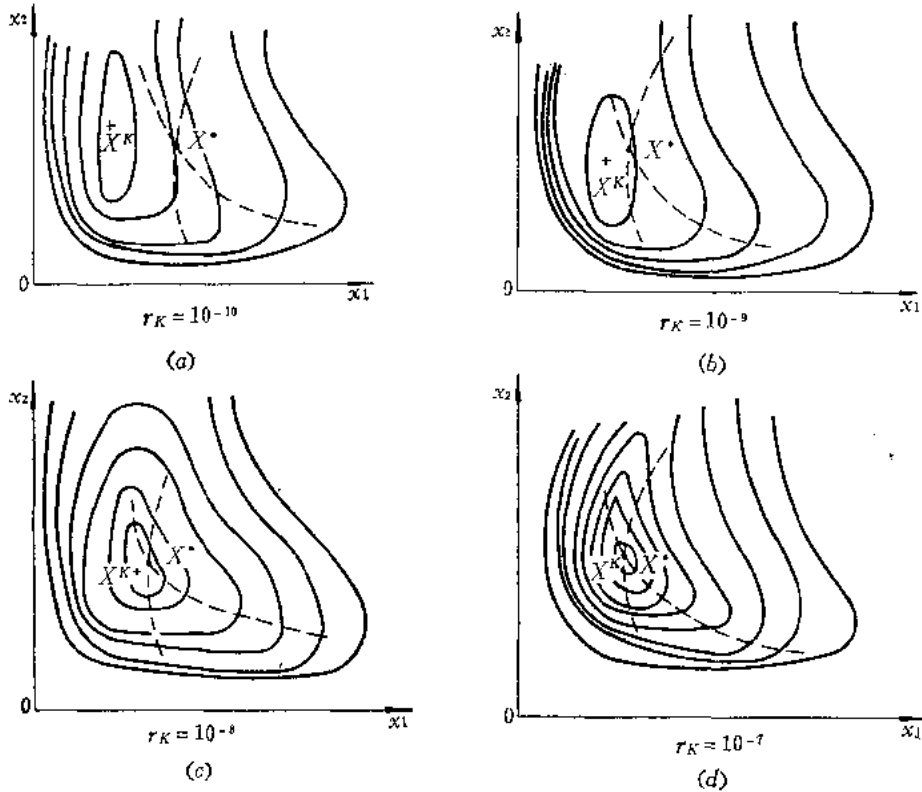


图 5—7

数将变得更加难于处理。直到最后不论从哪一个初始点出发，实际上都不可能求得极值点。因此，虽然从表面上看，为了迫使可行极值点趋向约束极值点，最后的 r_K 要取“大值”，但为了使 $F(x, r_K)$ 的极小化过程不至于太困难， r_K 值又要取“小值”。所以通常都是采用序列极小化方法。

从图中还可以看出，随着 r_K 值由小增大，可行极值点 x^K 是从约束条件所确定的可行

区域外部逐渐移向约束极值点 x^* 的。

二、内点法

内点法的基本思想是把不等式约束写成障碍项的形式，其它和外点法相同。但对障碍项要求：

- 1) 它在所论空间上必须是连续函数；
- 2) 当迭代点 x 在可行域内部距约束边界较远的地方，它的取值不大；而当 x 接近约束边界时，它的取值陡然增大，甚至为无穷大。

符合这些要求的障碍项可取为 $-\frac{1}{g_j(x)}$ 或 $-\ln[-g_j(x)]$ 等。因此，内点法的约束函数的形式可以是

$$F(x, r_K) = f(x) - r_K \sum_{j=1}^p \frac{1}{g_j(x)} \quad (5-27)$$

或

$$F(x, r_K) = f(x) - r_K \sum_{j=1}^p \ln[-g_j(x)] \quad (5-27a)$$

其中 r_K 是可以由大到小改变的正常数。只要点 x 是可行点，约束函数 $g_j(x)$ 对约束函数 $F(x, r_K)$ 就起障碍性的制约作用。当点 x 在可行域内远离约束曲面时，障碍项的作用很小。而当点 x 在可行域内趋向于某一约束曲面边界时，相应的约束函数 $g_j(x)$ 将趋近于零，于是障碍项的值就急速增加，以至趋于无穷大。这就保证了所求得的点不会越出可行域的边界，即在搜索过程中能保证点 x 与边界始终保持一段距离，成为一种障碍，使搜索点 x 不致跳出边界。因此，要想求得目标函数 $f(x)$ 的约束极小值点，就要设法使 r_K 起作用，让它不断地由大变小以减小障碍作用。

例如，对于 $f(x) = \frac{x}{2}$ ，约束条件为 $1-x \leq 0$ 的问题，当采用内点法时，它的障碍函数是

$$F(x, r_K) = \frac{x}{2} - r_K \frac{1}{1-x}$$

r_K 为0.25, 0.04和0.01等不同数值的情况如图(5-8)所示。随着 r_K 值的减小，障碍函数 $F(x, r_K)$ 从可行域内部向着约束边界弯过去，从而无约束极小点从可行域内部越来越接近真实的极小值点。

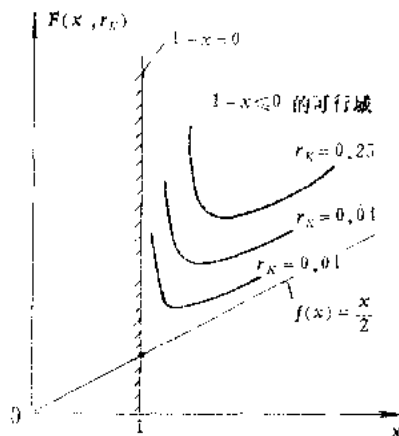


图 5-8

和外点法相反，在内点法中 r_K 则需要由大逐渐变小地来选取。仍用上面那个两杆桁架的约束优化问题来说明这一点。图(5-9 a, b, c)是当顺次取 $r_K = 10^7, 10^6, 10^5$ 时的情形。

从图中可以看出，当 r_K 逐渐减小时，可行极小值点 x^k 是从可行域内部逐渐向约束极小值点 x^* 趋近的。同样， $F(x, r_K)$ 的极小值点越是逼近真实极小值点，它的等值线就

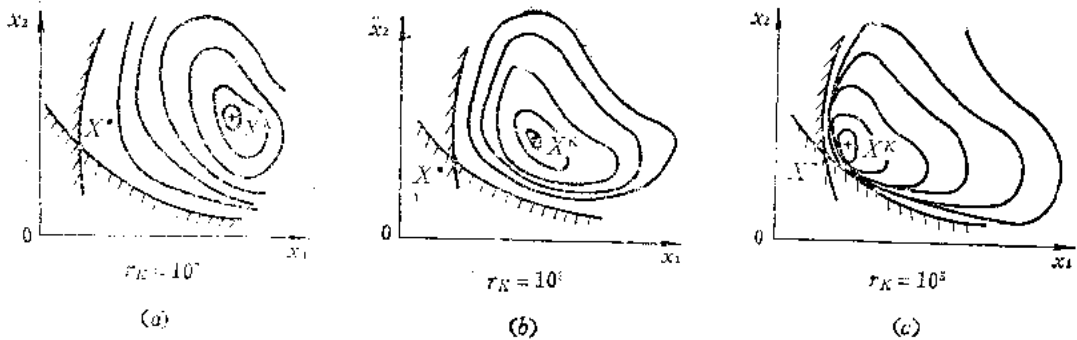


图 5-9

变得越加偏心。因此，虽然从表面上看，为了迫使可行极小值点趋向约束极小值点，最后的 r_K 要取“小值”；但是，为了使 $F(x, r_K)$ 的极小化过程不致太困难， r_K 值又要取“大值”。所以通常也都是采用序列极小化方法。

归纳上面的说明，可以把两种方法简述如下。

用外点法时，求 $x = (x_1, x_2, \dots, x_n)^T$ 使 $f(x)$ 极小，其约束条件是 $g_j(x) \leq 0$ ($j=1, 2, \dots, p$)的约束优化问题，被改造成为，求 $x = (x_1, x_2, \dots, x_n)^T$ 使 $F(x, r_K) = f(x) + r_K \sum_{j=1}^p \{\max[0, g_j(x)]\}^2$ 极小的无约束优化问题。其中的 r_K 是一个参数，它可以取如下一系列数值：

$$0 < r_1 < r_2 < \dots < r_{K-1} < r_K < \dots$$

$$\lim_{K \rightarrow \infty} r_K = \infty$$

即求解一系列 r_K 值时的 $F(x, r_K)$ 的优化解。当 r_K 逐步由小变大，直到趋于无穷大时，以 r_K 为参数的函数 $F(x, r_K)$ 的一系列优化解 $x^* = x(r_K)$ 逐渐趋向于原问题的优化解 x^* 。这就可以认为，对应于 r_K 的一系列优化解 x^* 是沿着以 r_K 为参数的一条迹线移动的，并且随着 r_K 由小变大而沿着这条迹线逐步趋向于原问题的优化解。由于在这种方法中，解 x^* 的轨迹是从由约束条件所定义的可行区域外部向着可行区域逐步趋向约束边界的，所以可称这种方法为“外点（罚函数）法”。

相反，内点法时，是把原问题变换成求 $x = (x_1, x_2, \dots, x_n)^T$ ，使 $F(x, r_K) = f(x) - r_K \sum_{j=1}^p \frac{1}{g_j(x)}$ 或 $F(x, r_K) = f(x) - r_K \sum_{j=1}^p \ln[-g_j(x)]$ 极小的无约束优化问题。其中 r_K 也是一个参数，它可以取如下一系列数值：

$$r_0 > r_1 > r_2 > \dots > r_{K-1} > r_K > \dots > 0$$

$$\lim_{K \rightarrow \infty} r_K = 0$$

即求解一系列 r_K 值时的 $F(x, r_K)$ 的优化解。当 r_K 不断减小时，以 r_K 为参数的函数 $F(x, r_K)$ 的一系列优化解 $x^* = x(r_K)$ ，也就逐步趋向原问题 $f(x)$ 的优化解 x^* 。由于在这种方法中，解 x^* 的轨迹是从约束条件所定义的可行区域内部向约束边界趋近的，所以又称它为“内点（罚函数）法”。

另外，表面上看，最后的 r_K 要取“大值（或小值）”；但是，为了使 $F(x, r_K)$ 的极小化过程不至于太困难，参数 r_K 却要取“小值（或大值）”。所以，不管是外点法还

是内点法，它们的优化求解过程都是以 r_K 为参数的一系列优化过程。因此，这类方法又称为“序列无约束极小化方法 (SUMT)”。

对于只含等式约束的极小化问题，可以采用类似外点法的约束函数的形式，即

$$\begin{aligned} f(x) \rightarrow \min \\ \text{s.t. } h_i(x) = 0 \quad (i = 1, 2, \dots, l) \end{aligned}$$

它的约束函数可写成

$$F(x, M_K) = f(x) + M_K \sum_{i=1}^l h_i(x)^2 \quad (5-28)$$

一般地说，目标函数 $f(x)$ 的约束条件是既有等式约束 $h_i(x) = 0$ ，又有不等式约束 $g_j(x) \leq 0$ 。这时约束函数将是用内点法和外点法的混合形式，如式(5-25)那样的形式。相应地称这种方法为“混合(罚函数)法”。在混合法中，问题常表述为：

求 $x = (x_1, x_2, \dots, x_n)^T$ ，使 $f(x)$ 极小；约束条件是

$$\begin{aligned} h_i(x) = 0 \quad (i = 1, 2, \dots, l) \\ g_j(x) \leq 0 \quad (j = 1, 2, \dots, p) \quad (l + p = m) \end{aligned}$$

相应的无约束优化问题的目标函数形式是

$$F(x, r_K, M_K) = f(x) - r_K \sum_{j=1}^p \frac{1}{g_j(x)} + M_K \sum_{i=1}^l h_i(x)^2 \quad (5-29)$$

或

$$F(x, r_K, M_K) = f(x) - r_K \sum_{j=1}^p \ln[-g_j(x)] + M_K \sum_{i=1}^l h_i(x)^2 \quad (5-29a)$$

也有的写成

$$F(x, r_K) = f(x) - r_K \sum_{j=1}^p \frac{1}{g_j(x)} + \frac{1}{\sqrt{r_K}} \sum_{i=1}^l h_i(x)^2 \quad (5-30)$$

或

$$F(x, r_K) = f(x) - r_K \sum_{j=1}^p \ln|g_j(x)| + \frac{1}{r_K} \sum_{i=1}^l h_i(x)^2 \quad (5-30a)$$

下面举个具体例子。

求函数 $f(x) = x_1^2 + x_2^2$ 的极小值点，使满足约束条件 $x_1 \geq 1$ 。

此约束条件可以改写为 $g_1(x) = 1 - x_1 \leq 0$ 。

显然，这个问题的优化解是 $x_1^* = 1$ ， $x_2^* = 0$ ， $f(x^*) = 1$ 。现在分别用内点法和外点法求解。

用内点法，此时有 $F(x, r_K) = x_1^2 + x_2^2 - r \ln(x_1 - 1)$ 。求 $F(x, r_K)$ 的极小值。由 $\frac{\partial F}{\partial x_1} = 2x_1 - r_K \frac{1}{x_1 - 1} = 0$ 和 $\frac{\partial F}{\partial x_2} = 2x_2 = 0$ 。得 $x_2 = 0$ ， $x_1^2 - x_1 - \frac{r_K}{2} = 0$ 或 $x_1 = \frac{1}{2} \pm \sqrt{\frac{1}{4} + \frac{r_K}{2}}$ 。其中负号使 $x_1 < 0$ 与原题要求不符，所以只能取正号，即取 $x_1 = \frac{1 + \sqrt{1 + 2r_K}}{2}$ 。

则得优化解为

$$x^K = \left(\frac{1 + \sqrt{1 + 2r_K}}{2}, 0 \right)$$

依次取 $r_K = 1, \frac{1}{10}, \frac{1}{100}, \dots, 0$ 时, 可得

$$x^1 = \left(\frac{1 + \sqrt{3}}{2}, 0 \right), x^2 = \left(\frac{1 + \sqrt{\frac{13}{10}}}{2}, 0 \right), x^3 = \left(\frac{1 + \sqrt{\frac{102}{100}}}{2}, 0 \right), \dots, x^K = (1, 0) = x^*$$

可见, 当 $r_K \rightarrow 0$ 时, $x^K \rightarrow x^* = (1, 0)$ 。图 (5-10) 示出了 r_K 从 $1, \frac{1}{10}, \frac{1}{100}, \dots, 0$ 的过程, 相应的 x^K 值的轨迹。它们都在可行域内部, 最后趋于边界。

用外点法, 此时有 $F(x, r_K) = x_1^2 + x_2^2 + r_K(1 - x_1)^2$ 。

求 $F(x, r_K)$ 的极小值。由 $\frac{\partial F}{\partial x_1} = 2x_1 - 2r_K(1 - x_1) = 0$ 和 $\frac{\partial F}{\partial x_2} = 2x_2 = 0$ 。得 $x_2 = 0$,

$x_1 = \frac{r_K}{r_K + 1}$ 。则得优化解为

$$x^K = \left(\frac{r_K}{r_K + 1}, 0 \right) \text{ 或 } x^K = \left(\frac{1}{1 + \frac{1}{r_K}}, 0 \right)$$

依次取 $r_K = 1, 10, 100, \dots, \infty$ 时, 可得

$$x^1 = \left(\frac{1}{2}, 0 \right), x^2 = \left(\frac{10}{11}, 0 \right), x^3 = \left(\frac{100}{101}, 0 \right) \dots$$

$$x^K = (1, 0) = x^*$$

即当 $r_K \rightarrow \infty$ 时, $x^K \rightarrow x^* = (1, 0)$ 。图 (5-11) 示出了 r_K 从 $1, 10, 100, \dots, \infty$ 的过

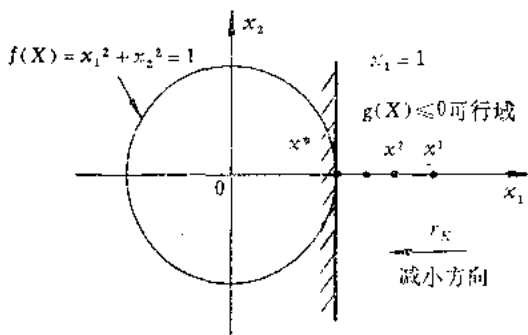


图 5-10

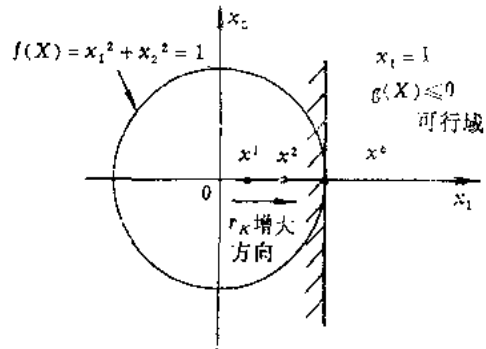


图 5-11

程, 相应的 x^K 值的轨迹, 它们是从可行域的外部最后趋于边界的。

比较两种方法可以看出, 外点法对于初始点及每一步迭代得到的迭代点都不要求是可行点, 而且在迭代过程中并不能保证得到的点是可行点。但在迭代过程中, 一般地说它可以使迭代点一步步地进入可行域。内点法要求给出的初始点必须是可行点, 而且每一步迭代得到的迭代点还必须保持是可行点。这虽然可使设计者有较多的可行方案供选择, 但要考虑初始点确定的问题。外点法对于处理等式约束和不等式约束都可使用, 而内点法却只能处理不等式约束。在外点法中, 由于约束函数 $g_j(x)$ 值的大小过于悬殊 (即尺度问题) 而带来很多问题。量纲不同就是经常出现的问题之一。为了避免这一问

题, 可使约束条件无量纲化或规一化, 或者采用整调罚因子的加权方法。外点法当 r_K 增大时, 有机器溢出的问题, 这就要考虑选择比例因子; 而对内点法, 当 r_K 趋于零时, 就要考虑误差的选取问题。此外, 两种方法的计算量都较大, 化费机时多。因此, 虽然两种方法能取得较好的效果, 但目前已有被其它方法取代的趋势。

采用内点法时, 可以用下面的办法选取初始点。

首先, 令 y^0 是第一次选取的内点。显然它不一定能使所有 m 个 (假定共有 m 个用内点法处理的不等式约束, 当然其中不包括按外点法处理的等式和不等式约束) 约束的值都等于或小于零, 即不能满足约束条件。

$$g_j(y^0) \leq 0 \quad (j=1, 2, \dots, m)$$

而只能使其中的 l 个约束条件得到满足, 即使 l 个约束的值小于零 (简称 L 组), 其余的 $m-l$ 个约束条件得不到满足, 它们的值将等于或大于零 (简称 M 组), 即

$$L\text{组: } g_j(y^0) < 0 \quad (j=1, 2, \dots, l)$$

$$M\text{组: } g_j(y^0) \geq 0 \quad (j=l+1, \dots, m)$$

现在的目标是: 从点 y^0 开始进行迭代计算, 以便能把从 y^0 点经过 K 次的移动, 到达点 y^K 的过程中出现满足条件 $g_j(y^K) < 0$ 的约束条件, 都及时地转入 L 组中去。如此继续, 直到 M 组中的约束条件都逐步进入 L 组中去为止。

仿照内点法, 可以认为这个迭代计算过程是对一个改造过的目标函数

$$\phi(y, r_K) = \sum_{j \in M} g_j(y) - r_K \sum_{j \in L} \frac{1}{g_j(y)}$$

进行优化来完成的。

其中的 r_K 是一个逐次减小的数列。如果 $r_K \rightarrow 0$ 时还不能算出满足所有约束条件的点 y^K , 就说明在这样的约束组中没有可行的内点。这种方法可以称为“逐次进入约束”的初始点选取法。

下面给出外点法和内点法的算法步骤和相应的框图。

外点法的算法步骤和框图:

- 1) 选取一个适度的 r_K 值, 并从任意一个初始点 x^0 出发, 求 $F(x, r_K) = f(x) + r_K \sum_{j=1}^p \{\max[0, g_j(x)]\}^2$ 的一个最小点 x^K 。
- 2) 了解约束违反情况, 以确定 x^K 是否在可行域内。
- 3) 如果 x^K 是可接受的点, 则计算结束; 否则再取一个比原来 r_K 大的新的 r_{K+1} , 可用 $r_{K+1} = cr_K$ ($c > 1$) 实现。从 x^K 点出发, 又回到第一步, 求 $F(x, r_K)$ 的极小点。

相应的框图如图 (5-12a) 所示。

内点法的算法步骤和框图:

- 1) 给出一个满足所有不等式 $g_j(x) \leq 0$ 的初始点 x^0 和 r_K 的初值, 求 $F(x, r_K) = f(x) - r_K \sum_{j=1}^p \frac{1}{g_j(x)}$ 的极小值点 x^K 。
- 2) 检验 x^K 是否已收敛到问题的最优解。
- 3) 如果收敛性条件未满足, 再选取一个 $r_{K+1} = cr_K$ ($c < 1$), 算出一个新的初始点,

回到第一步，重新开始极小化过程。

相应的框图如图(5-12b)所示。

上面所讨论的约束函数方法，最优解都是通过求解一系列无约束最优化问题获得的。近来在一些书刊中介绍一种“精确罚函数法”，它的目的是针对约束函数法的上述缺点，即不需要通过一系列无约束优化过程才能获得优化解，而是期望只进行一次无约束优化过程就能得到优化解。不过这种方法要计算二阶偏导数，所以下节要介绍的乘子法应用广泛。但由于篇幅所限，这里就不对精确罚函数法做介绍了。

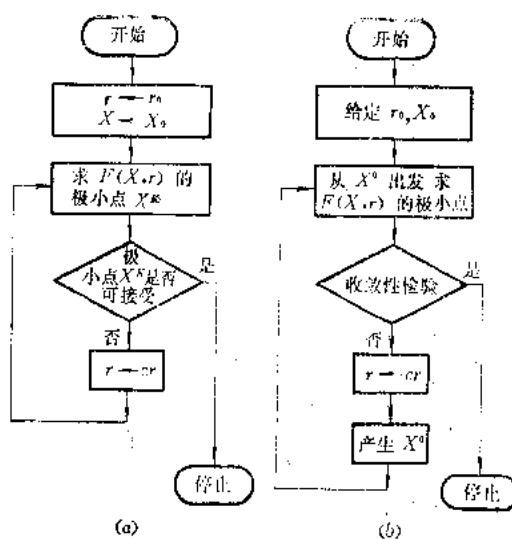


图 5-12

§ 4 乘子法简述

乘子法又称PHR (Powell, Hestens和Rochafellar) 法，它是在约束函数法和拉格朗日乘子法的基础上发展起来的，是近年来比较受重视并已日益得到较多应用的一种方法。它比普通的约束函数法的收敛速度快。

我们先看等式约束优化问题的拉格朗日乘子法。若目标函数是 $f(x)$ ，等式约束条件是 $h_i(x) = 0$ ($i = 1, 2, \dots, l$)，相应的拉格朗日函数是

$$L(x, \lambda) = f(x) + \sum_{i=1}^l \lambda_i h_i(x)$$

一般地说，它的二阶偏导数矩阵并不是正定的。然而，若把上面的拉格朗日函数按外点法引入等式约束的制约项 $\sum_{i=1}^l h_i(x)^2$ ，而形成增广的拉格朗日函数

$$F(x, \lambda, r) = f(x) + \sum_{i=1}^l \lambda_i h_i(x) + r \sum_{i=1}^l h_i(x)^2 \quad (5-31)$$

则可证明，在一定条件下，必存在一个 r' ，对一切满足 $r \geq r'$ 的制约函数 $F(x, \lambda, r)$ 的二阶偏导数矩阵总是正定的。

因为函数的二阶偏导数矩阵为正定是函数极值的充分条件，所以当罚因子 r 取足够大的定值 $r \geq r'$ ，但不必趋于无穷，而且恰好选取了一个 $\lambda = \lambda^*$ 时，就可以根据函数极值的必要条件

$$\nabla F(x^*, \lambda^*, r) = 0$$

求得函数的极小点 x^* (这里的 r 是数量， $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_l]^T$ 是向量)。

这就是说，如果已经确定了 $r \geq r'$ 和 $\lambda = \lambda^*$ 的值，则只要对式(5-31)进行一次无约束极小化就可以求得原问题的优化解 x^* 。

当然，要想做到这一点，需要首先求取 $r \geq r'$ 的值，并在此基础上采用下面的迭代

公式求解拉格朗日乘子向量 λ

$$\tilde{\lambda} = \lambda + rH(x^K) \quad (5-32)$$

其中的 $H(x^K)$ 是第 K 次迭代计算时得到优化解 x^K 时约束条件 $h_i(x^K)$ 的列向量。

因此,乘子法的基本思想和计算过程可以简述如下。

在构造增广的拉格朗日函数 $F(x, \lambda, r)$ 以后,假若能够确定 $\lambda = \lambda^*$, $r \geq r'$,则只要对它求一次无约束极小化就能得到原问题的优化解 x^* 。但是 λ^* 也是未知的。可以采用迭代式(5-32)进行计算。为此,须先考虑对任一 λ ,当 r 值固定时,函数 $F(x, \lambda, r)$ 的极小值计算。即从任一初始乘子向量 λ 开始,对取定的 λ 和 r 值求函数 $F(x, \lambda, r) = F(x)$ 的极小点 $x = x(\lambda)$,并计算 $H(x^K)$,再利用求得的 x 及式(5-32)调整计算乘子向量 λ 。经过这样的迭代后得新的 λ ,一般说它应该更接近于 λ^* 。于是再用它来构造 $F(x, \lambda^*, r)$ 后,求得的极小点 x 也将比上次求得的结果更接近 x^* 。这样,求 $F(x, \lambda, r)$ 的无约束极小点和调整乘子的过程相辅相成,交替进行,促使 $x \rightarrow x^*$,同时 $\lambda \rightarrow \lambda^*$ 。按式(5-32)调整 λ ,以期逐步求出 λ^* 的过程又称乘子迭代。这是原来SUMT法中所没有的。因此,称这种方法为乘子法。

在具体计算时,要解决如何选取 r 值的问题。因为从理论上说,乘子法中无需把罚因子 r 趋于无穷,只要取 r 大于等于 r' 值即可。但是 r 如果一开始就取得很大,将会使制约函数的极小化过程遇到困难,甚至做不下去。因此也是不合适的。为此,鲍威尔(Powell)建议采用如下的具体算法。

1) 选取初始乘子向量 λ_0 和适当选取的罚因子 $r_0 > 0$,并事先规定 r 的增长系数 $\beta (> 1)$,精度控制参数 ϵ 和某个 $\gamma (< 1)$ 。置 $K = 0$ 。

2) 求 $\min F(x, \lambda, r_K)$,设所得的无约束最优解为 x^K ,再算出

$$\|H(x^K)\| = \left(\sum_{i=1}^l h_i(x^K)^2 \right)^{\frac{1}{2}}$$

3) 按式(5-32)由向量 λ , x^K 及罚因子 r_K 求出新的乘子向量 $\tilde{\lambda}$ 。

4) 如果 $\|H(x^K)\| < \epsilon$,则已经得到问题的解 $x^* = x^K$, $\lambda^* = \tilde{\lambda}$,计算结束;否则转到下一步。

5) 取罚因子

$$r_{K+1} = \begin{cases} \beta r_K & \text{若 } \|H(x^K)\| / \|H(x^{K+1})\| > \gamma \\ r_K & \text{反之} \end{cases}$$

再置 $K \leftarrow K + 1$ 后回到第2步。

Powell 建议取 $\beta = 10$, $\gamma = 0.25$ 。

初始乘子向量可取 $\lambda^0 = 0$,即第一次极小化可用普通外点法进行。下一次极小化的初始点可以用前一次所得的最优解。也可以不采用 r 的限制条件,而总是取 $r_{K+1} = \beta r_K$ 。此时可取 $\beta = 2 \sim 4$ 。

对于不等式约束,可以采用引入人为变量 z (这里为了简明,不用 x_{n+k} 而改用 z 代表人为变量)的办法把它转化成如下形式的等式约束

$$h_j(x, z) = g_j(x) + z_j^2 \quad (j = 1, 2, \dots, p)$$

并组成增广的拉格朗日函数

$$F(x, z, \lambda, r) = f(x) + \sum_{j=1}^p [\lambda_j h_j(x, z) + r h_j(x, z)^2] \quad (5-33)$$

然后, 仿照前面对等式约束的乘子法的办法对式(5-33)进行无约束极小化计算, 求得极小点 x^* 。但是, 应当注意, 此时的乘子迭代公式(5-33)的形式应是

$$\tilde{\lambda}_j = \lambda_j + r h_j(x, z) = \lambda_j + r(g_j(x) + z_j^2)$$

这时由于增加了人为变量 z , 在 λ 和 r 值已定的条件下, 极小化函数由 $\min_x F(x)$ 扩大成为 $\min_{x, z} F(x, z)$ 。这就使得原来的 n 维极值问题扩大成 $n+p$ 维极值问题了。因而势必要增加计算量, 而且还要给数值极小化过程带来困难。所以, 有必要把计算予以简化。

可以把函数 $F(x, z) = f(x) + \sum_j [\lambda_j (g_j(x) + z_j^2) + r (g_j(x) + z_j^2)^2]$ 的极小化过程分两步进行。即 $\min_{x, z} F(x, z)$ 变成 $\min_x \{ \min_z F(x, z) \}$ 。就是先把 x 固定对变量 z 求极小值, 再对 x 求极小值。

根据分析, 若 $\lambda_j + r g_j(x) \geq 0$ 有 $z_j(z) = 0$, 则

$$\tilde{\lambda}_j = \lambda_j + r g_j(x) = \max\{0, \lambda_j + r g_j(x)\}$$

若 $\lambda_j + r g_j(x) < 0$, 有 $\lambda_j + r(g_j(x) + z_j^2) = 0$, 则

$$\tilde{\lambda}_j = 0 = \max\{0, \lambda_j + r g_j(x)\}$$

所以, 函数 $F(x, z)$ 对 z_j 求极小值的结果以及乘子 λ_j 的迭代公式可以采用统一的形式

$$F(x) = f(x) + \frac{1}{r} \sum_{j=1}^p \{ [\max\{0, \lambda_j + r g_j(x)\}]^2 - \lambda_j^2 \} \quad (5-34)$$

此式就是不等式约束优化问题的增广拉格朗日函数。它和式(5-33)给出的 $F(x, z)$ 不同的地方是, 其中已经不再包含人为变量 $z_j (j = 1, 2, \dots, p)$ 了。因此, 在实际计算的极小化循环中, 仍然可以只要对给定的 λ 和 r 的函数 $F(x, \lambda, r)$ 求 n 维的无约束优化问题。并且此时的乘子迭代公式是

$$\tilde{\lambda}_j = \max\{0, \lambda_j + r g_j(x(\lambda))\} \quad (5-35)$$

其中也不再出现人为变量 z_j 了。

就式(5-34)和(5-35)按照等式约束时的办法, 反复地交替进行序列化迭代计算, 就可求得极小值点 x^* 。而且这时不管乘子向量的初值 λ^0 如何选取, 迭代所得的 λ 总是非负的, 从而就满足 Kuhn-Tucker 条件的要求。

可以根据前后两次迭代的乘子向量值 $\tilde{\lambda}$ 和 λ 是否已经满足精度要求来确定计算终止的条件。它的计算框图如图(5-13)所示。

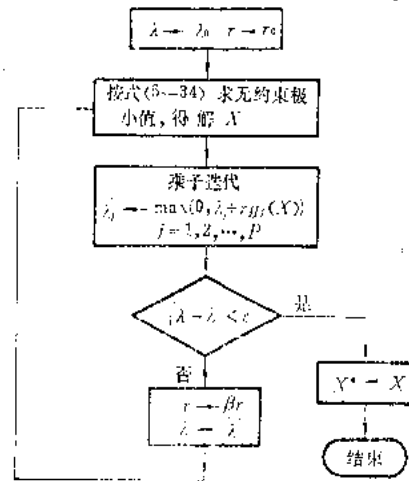


图 5-13

§ 5 二次规划迭代法

对于约束非线性的优化问题，当把它转化为无约束问题以后，就可以选用任何一种合适的无约束优化方法来求解。牛顿方法是具有二次收敛特性的一种无约束优化方法，对于二次函数它只需一次迭代就可到达极值点。因此，一个很自然的想法是，当把非线性规划问题转化成无约束优化问题之后，如何采用牛顿法来求解。下面就来讨论这个问题。

假定原问题是

$$\begin{aligned} \min f(x) \\ \text{s.t. } h(x) = 0 \end{aligned} \quad (5-36)$$

这里的 $h(x)$ 是向量函数，即

$$h(x) = [h_1(x), h_2(x), \dots, h_m(x)]^T$$

以上等式约束问题相应的拉格朗日函数是

$$L(x, \lambda) = f(x) + \lambda^T h(x)$$

其中 λ 为拉格朗日乘子向量，有

$$\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)^T$$

通过引入拉格朗日函数，将上面的约束优化问题转化为无约束问题。此问题的极值(Kuhn—Tucker)条件是

$$\begin{aligned} \nabla L(x, \lambda) = \nabla f(x) + \nabla h(x)\lambda = 0 \\ h(x) = 0 \end{aligned} \quad (5-37)$$

其中 $\nabla h(x)$ 为 $n \times m$ 矩阵，即

$$\nabla h(x) = [\nabla h_1(x), \nabla h_2(x), \dots, \nabla h_m(x)]$$

而
$$\nabla h_j(x) = \left[\frac{\partial h_j(x)}{\partial x_1}, \frac{\partial h_j(x)}{\partial x_2}, \dots, \frac{\partial h_j(x)}{\partial x_n} \right]^T \quad (j=1, 2, \dots, m)$$

上面方程的求解可以有两种途径，一是直接采用牛顿方法，另一种是把它线性化联立求解。通过以下的叙述可以看出，这两种途径都导致相同的结果。

若对式(5-37)采用牛顿法求解，则相应的迭代公式分别是

$$x^{K+1} = x^K - G_K^{-1} \nabla_x L$$

或

$$x^{K+1} = x^K - G_K^{-1} [\nabla f(x^K) + \nabla h(x^K)\lambda^K]$$

和

$$x^{K+1} = x^K - [\nabla h(x^K)]^{-1} h(x^K)$$

其中 ∇_x 表示对 x 求导，由上两式可得

$$G_K(x^{K+1} - x^K) + \nabla h(x^K)\lambda^K = -\nabla f(x^K)$$

和

$$\nabla h(x^K)(x^{K+1} - x^K) = -h(x^K)$$

写成矩阵形式是

$$\begin{pmatrix} G_K & \nabla h(x^K) \\ \nabla h(x^K) & 0 \end{pmatrix} \begin{Bmatrix} x^{K+1} - x^K \\ \lambda^K \end{Bmatrix} = - \begin{Bmatrix} \nabla f(x^K) \\ h(x^K) \end{Bmatrix} \quad (5-38)$$

如果取

$$\begin{aligned} \mathbf{d} &= \mathbf{x}^{K+1} - \mathbf{x}^K \\ \mathbf{x} &= \mathbf{x}^K \\ \lambda &= \lambda^K \end{aligned}$$

并且用

$$\mathbf{B} \approx \mathbf{G}$$

(\mathbf{B} 可以用式(3—24)计算,但须把其中的 \mathbf{s} 和 \mathbf{y} 互换位置),则式(5—38)可以写成

$$\begin{bmatrix} \mathbf{B} & \nabla \mathbf{h}(\mathbf{x}) \\ \nabla \mathbf{h}(\mathbf{x}) & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{d} \\ \lambda \end{Bmatrix} = - \begin{Bmatrix} \nabla f(\mathbf{x}) \\ \mathbf{h}(\mathbf{x}) \end{Bmatrix} \quad (5-39)$$

关于求解由库恩—塔克条件所建立的联立方程组(5—37)的另一种途径是,对式(5—37)在点 \mathbf{x} 和 λ 处进行泰勒展开,取其一阶项分别使之线性化,则得

$$\nabla^2 f(\mathbf{x}^K)(\mathbf{x}^{K+1} - \mathbf{x}^K) + (\lambda^K)^T \nabla^2 \mathbf{h}(\mathbf{x}^K)(\mathbf{x}^{K+1} - \mathbf{x}^K) + \nabla \mathbf{h}(\mathbf{x}^K) \lambda^K + \nabla f(\mathbf{x}^K) = 0$$

和 $\mathbf{h}(\mathbf{x}^K) + \nabla \mathbf{h}(\mathbf{x}^K)(\mathbf{x}^{K+1} - \mathbf{x}^K) = 0$

或当令 $\nabla_x^2 L(\mathbf{x}, \lambda) = \nabla^2 f(\mathbf{x}^K) + (\lambda^K)^T \nabla^2 \mathbf{h}(\mathbf{x}^K)$ 时,可写成

$$\nabla_x^2 L(\mathbf{x}, \lambda)(\mathbf{x}^{K+1} - \mathbf{x}^K) + \nabla f(\mathbf{x}^K) + \nabla \mathbf{h}(\mathbf{x}^K)(\lambda^K)^T = 0$$

和 $\mathbf{h}(\mathbf{x}^K) + \nabla \mathbf{h}(\mathbf{x}^K)(\mathbf{x}^{K+1} - \mathbf{x}^K) = 0$

和前面一样,如果取 $\mathbf{x}^{K+1} - \mathbf{x}^K = \mathbf{d}$, $\mathbf{x}^K = \mathbf{x}$, $\lambda^K = \lambda$,并且考虑对 $\nabla_x^2 L(\mathbf{x}, \lambda)$ 进行 $\mathbf{B} \approx \nabla_x^2 L(\mathbf{x}, \lambda)$ 的迭代运算,则上面两个线性化的方程式同样可以写成式(5—39)的形式。

式(5—39)的原问题将是下面的二次规划问题

$$\min_{\mathbf{d}} \frac{1}{2} \mathbf{d}^T \mathbf{B} \mathbf{d} + \nabla f(\mathbf{x})^T \mathbf{d} \quad (5-40)$$

$$\text{s.t.} \quad \nabla \mathbf{h}(\mathbf{x})^T \mathbf{d} + \mathbf{h}(\mathbf{x}) = 0$$

这是因为式(5—40)的拉格朗日函数是

$$L(\mathbf{d}, \lambda) = \frac{1}{2} \mathbf{d}^T \mathbf{B} \mathbf{d} + \nabla_x f(\mathbf{x})^T \mathbf{d} + \lambda^T [\nabla \mathbf{h}(\mathbf{x})^T \mathbf{d} + \mathbf{h}(\mathbf{x})]$$

它的局部极小点的 Kuhn—Tucker 条件是

$$\nabla_x L = \mathbf{B} \mathbf{d} + \nabla f(\mathbf{x}) + \nabla \mathbf{h}(\mathbf{x}) \lambda = 0$$

和 $\nabla_\lambda L = \nabla \mathbf{h}(\mathbf{x})^T \mathbf{d} + \mathbf{h}(\mathbf{x}) = 0$

它们就是原问题的 K—T 条件式(5—37)用牛顿法求解的迭代公式(5—39)。所以,用式(5—39)表达的问题和式(5—40)的二次规划问题是等价的。因此,通过反复求解由式(5—40)表达的二次规划问题就能获得原问题式(5—37)的解。

通常,把 \mathbf{B} 的初始值取为单位矩阵,即取 $\mathbf{B}_0 = \mathbf{I}$,而在以后各次循环迭代时,运用拟牛顿法(如DFP或BFGS法)逐步修改 \mathbf{B}_k ,并由式(5—40)解出一维搜索方向 \mathbf{d}_k ,然后按迭代公式

$$\mathbf{x}^{K+1} = \mathbf{x}^K + \alpha_k \mathbf{d}^K$$

计算 \mathbf{x}^{K+1} 。

因此，这里我们同样是要解决寻求 d^k 和计算 a_k 的问题。

一、寻求搜索方向 d

利用式 (5-40) 可以解出搜索方向 d ，对于这样的二次规划问题可以有多种解法。下面对其中的一种方法作扼要的说明。它实际上可以看成是线性规划问题中单纯形法的推广。

二次规划问题（设有 m 个等式约束）

$$\min f(x) = c^T x + \frac{1}{2} x^T G x$$

s. t.

$$A x = b$$

$$x \geq 0$$

其中 m 维向量 $b \geq 0$ ， G 是 $n \times n$ 阶的对称半正定矩阵（即 $f(x)$ 是凸函数，问题是凸规划问题）。

它的拉格朗日函数是

$$F(x, u, v) = c^T x + \frac{1}{2} x^T G x + u^T (A x - b) - v^T x$$

由于变量 x 是 n 维， u 是 m 维， v 是 n 维，所以这里有 $2n + m$ 个变量。

相应的 Kuhn—Tucker 条件是

$$c + G x + A^T u - v = 0$$

$$A x = b$$

$$v^T x = 0 \text{ (或 } v_i x_i = 0, i = 1, 2, \dots, n)$$

$$v \geq 0$$

$$x \geq 0$$

因为 $v^T x = 0$ 是非线性方程组，所以这里只有 $n + m$ 个线性方程组。

可以采用线性规划中的单纯形法，通过 $n + m$ 个线性方程组，求得 $2n + m$ 个变量中的 $n + m$ 个可行解来。

考虑到条件 $x \geq 0$ 和 $v \geq 0$ ，则由 $v^T x = 0$ （或 $v_i x_i = 0$ ）可知， x_i 和 v_i ($i = 1, 2, \dots, n$) 不能同时为正。所以在 x_i 和 v_i 这 $2n$ 个变量中最多也只能有 n 个为正。根据 $x \geq 0$ 和 $v \geq 0$ 的条件，不为正的 x_i 或 v_i 是不能进入可行解的。即在 $2n + m$ 个变量中，至多有 $m + n$ 个不为零的变量进入可行解。这也正好符合单纯形法的选取可行解的要求，即每组可行解中有 $m + n$ 个变量不为零，而其余的 n 个变量都为零。

这样，我们就可以采用单纯形法利用所提供的 $n + m$ 个线性方程组，求出满足 $v^T x = 0$ 条件的 $n + m$ 个非零变量来。它就是这个二次规划问题的满足 Kuhn—Tucker 条件的优化解。

具体作法是，对

$$c + G x + A^T u - v = 0$$

$$A x = b$$

引入松弛变量，然后用单纯形法不断地对可行解进行迭代、修正，直到松弛变量都变成

零后,即得优化解。不过在每次迭代时,都要保证条件

$$v^T x = 0$$

得到满足,即 x_i 和 v_i 不能同时进入可行解内。

这种方法的具体算法可以参考有关的书刊资料。

对于具有不等式约束的非线性规划问题是

$$\begin{aligned} & \min f(x) \\ \text{s.t.} \quad & h(x) = 0 \\ & g(x) \leq 0 \end{aligned} \quad (5-41)$$

此时的二次规划问题可规格化为

$$\begin{aligned} & \min_d \frac{1}{2} d^T B d + \nabla f(x)^T d \\ \text{s.t.} \quad & \nabla h(x)^T d + h(x) = 0 \\ & \nabla g(x)^T d + g(x) \leq 0 \end{aligned} \quad (5-42)$$

二、计算步长 α

在确定搜索方向 d 以后,接着就是计算步长 α 。为了避免约束条件(它们形成可行域的约束曲面)对步长的影响,可以取前两节中介绍的罚函数法、精确罚函数法或乘子法中的任何一种函数 $\Phi(x, \lambda)$ 作为搜索函数。例如,可以取下面三种函数形式之一(或其它形式):

1. 罚函数法

$$\Phi_1(x, r) = f(x) + r \sum_{i=1}^l |h_i(x)| + r \sum_{j=1}^p \max[g_j(x), 0]$$

式中 r 取充分大的值。

这时的优化步长因子 α 由以下一维搜索问题给出

$$\begin{aligned} & \min_{\alpha} \varphi(\alpha) = \Phi_1(x + \alpha d, r) \\ \text{s.t.} \quad & 0 < \alpha \leq \delta \end{aligned}$$

其中 δ 是任意正值。

2. 拉格朗日函数

$$\Phi_2(x, \lambda, \mu) = f(x) + \lambda^T h(x) + \mu^T g(x)$$

步长因子 α 由下式求解

$$\begin{aligned} & \min_{\alpha} \varphi(\alpha) = \Phi_2(x + \alpha d, \lambda, \mu) \\ \text{s.t.} \quad & 0 < \alpha \leq 1 \end{aligned}$$

3. 乘子法

$$\Phi_3(x, \mu, \nu) = f(x) + \sum_{i=1}^l \lambda_i |h_i(x)| + \sum_{j=1}^p \mu_j \max[g_j(x), 0]$$

其中

$$\begin{aligned} \mu_j^K &= \max\left\{\mu_j^K, \frac{1}{2}(\mu_j^K + \mu_j^{K-1})\right\} \\ \lambda_i^K &= \max\left\{|\lambda_i^K|, \frac{1}{2}(|\lambda_i^K| + \lambda_i^{K-1})\right\} \end{aligned}$$

步长因子 α 由下式求解

$$\begin{aligned} \min_{\alpha} \varphi(\alpha) &= \Phi_3(x + \alpha d, \mu, \nu) \\ \text{s.t.} \quad & 0 < \alpha \leq 1 \end{aligned}$$

从上述作法中可以看出,首先是解一个二次规划问题,以确定搜索方向 d ;接着通过原问题组成的无约束函数进行搜索,以计算步长 α ;然后按 $x^{K+1} = x^K + \alpha_K d^K$ 迭代式确定 x^{K+1} 。如此反复进行迭代运算,直到获得优化点。我们把这种作法称为二次规划迭代法。

三、算法步骤和框图

二次规划迭代法的一般算法步骤如下:

1. 给定初值 $x^0, B_0 = I$;
2. 解式(5-40)形式的二次规划问题,求出下一步搜索方向 d^K ;
3. 在方向 d^K 上按某种形式的函数进行一维搜索,得极小点 x^{K+1} ;
4. 若满足收敛准则就结束,否则修改 B_K ,回到2步重新进行迭代计算。

算法的简单框图如图(5-14)所示。

二次规划迭代法例题

$$\begin{aligned} \min f(x) &= (x_1 - 2)^2 + (x_2 - 1)^2 \\ \text{s.t.} \quad h_1(x) &= x_1 - 2x_2 + 1 = 0 \\ g_1(x) &= -\frac{x_1^2}{4} + x_2^2 - 1 \leq 0 \end{aligned}$$

不可行初始点取为 $x^0 = [2, 2]^T$ 。

首先,求在 x^0 点的函数值和梯度值

$$\begin{aligned} h_1(x^0) &= -1 & g_1(x^0) &= 4 \\ \nabla f(x^0) &= \begin{Bmatrix} 0 \\ 2 \end{Bmatrix} & \nabla h_1(x^0) &= \begin{Bmatrix} 1 \\ -2 \end{Bmatrix} \\ \nabla g_1(x^0) &= \begin{Bmatrix} 1 \\ 4 \end{Bmatrix} \end{aligned}$$

取 $B_0 = I$ 为初始迭代矩阵。为了求出搜索方向而形成的二次规划子问题是

$$\begin{aligned} \min \quad & \frac{1}{2}d_1^2 + \frac{1}{2}d_2^2 + 2d_2 \\ \text{s.t.} \quad & d_1 - 2d_2 = 1 \\ & d_1 + 4d_2 \leq -4 \end{aligned}$$

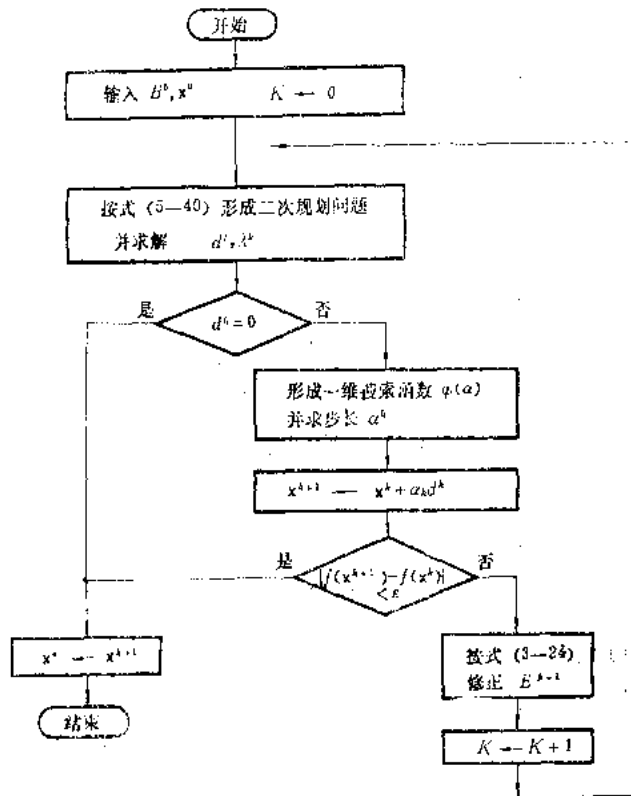


图 5-14

这里用手算办法求解此二次规划问题，设不等式约束达到等式，则问题变成

$$\min \frac{1}{2}d_1^2 + \frac{1}{2}d_2^2 + 2d_2$$

$$\text{s.t.} \quad \begin{aligned} d_1 - 2d_2 &= 1 \\ d_1 + 4d_2 &= -4 \end{aligned}$$

显然，它的解是联立线性方程组的解，即

$$\mathbf{d}^0 = \left[-\frac{2}{3}, -\frac{5}{6} \right]^T$$

用最优化条件，即K—T条件来检验一下，看 \mathbf{d}^0 是否是不等式约束二次规划问题的最优解。自拉格朗日函数

$$L(\mathbf{d}, \lambda) = \frac{1}{2}d_1^2 + \frac{1}{2}d_2^2 + 2d_2 + \lambda_1(d_1 - 2d_2 - 1) + \lambda_2(d_1 + 4d_2 + 4)$$

有K—T条件要求在最优点处应满足：

- 1) $\nabla L(\mathbf{d}, \lambda) = 0$;
- 2) 拉格朗日乘子 λ_2 为非负;
- 3) 满足约束条件。

首先，看 $\nabla L(\mathbf{d}, \lambda) = 0$ ，即

$$\begin{aligned} \frac{\partial L}{\partial d_1} &= d_1 + \lambda_1 + \lambda_2 = 0 \\ \frac{\partial L}{\partial d_2} &= d_2 + 2 - 2d_1 + 4\lambda_2 = 0 \end{aligned}$$

把 $\mathbf{d}^0 = \left[-\frac{2}{3}, -\frac{5}{6} \right]^T$ 代入，有

$$\begin{aligned} \lambda_1 + \lambda_2 &= \frac{2}{3} \\ -2\lambda_1 + 4\lambda_2 &= -\frac{7}{6} \end{aligned}$$

解得： $\lambda_1 = \frac{23}{36}$ ， $\lambda_2 = \frac{1}{36}$ 。

可见，条件1) 和2) 已满足。而 \mathbf{d}^0 是从约束解出的，自然满足条件3)。因此

$$\mathbf{d}^0 = \left[-\frac{2}{3}, -\frac{5}{6} \right]^T, \lambda^0 = \left[\frac{23}{36}, \frac{1}{36} \right]^T$$

是所求不等式约束二次规划问题的最优解， \mathbf{d}^0 就是搜索方向。

取下面的函数做为线性搜索函数

$$\Phi(a) = f(\mathbf{x}^0 + a\mathbf{d}^0) + \mu_1 |h_1(\mathbf{x}^0 + a\mathbf{d}^0)| + \mu_2 \max\{0, g_1(\mathbf{x}^0 + a\mathbf{d}^0)\}$$

此处，取 $\mu_1 = |\lambda_1| = \frac{23}{36}$ ； $\mu_2 = |\lambda_2| = \frac{1}{36}$ 。把 \mathbf{d}^0 代入可得

$$h_1(\mathbf{x}^0 + a\mathbf{d}^0) = 2 - \frac{2}{3}a - 2\left(2 - \frac{5}{6}a\right) + 1 = a - 1$$

$$g_1(\mathbf{x}^0 + a\mathbf{d}^0) = \frac{1}{6}\left(2 - \frac{2}{3}a\right)^2 + \left(2 - \frac{5}{6}a\right)^2 - 1 = 4 - 4a + \frac{29}{36}a^2$$

由于要求搜索时有 $0 < \alpha \leq 1$, 于是

$$|h_1(x^0 + \alpha d^0)| = 1 - \alpha$$

$$\max\{0, g_1(x^0 + \alpha d^0)\} = 4 - 4\alpha + \frac{29}{36}\alpha^2$$

则

$$\Phi(\alpha) = \frac{4}{9}\alpha^2 + (1 - \frac{5}{6}\alpha)^2 + \frac{23}{36}(1 - \alpha) + \frac{1}{36}(4 - 4\alpha + \frac{29}{36}\alpha^2)$$

$$\Phi'(\alpha) = \frac{8}{9}\alpha - \frac{5}{3}(1 - \frac{5}{6}\alpha) - \frac{23}{36} - \frac{1}{9} + \frac{29}{648}\alpha$$

计算可得 $\Phi'(0) = -2.416$, $\Phi(0) = 1.71$ 。

我们采用不精确的线性搜索。为了保证函数的下降特性, 要求 α 满足下面的条件

$$\Phi(\alpha) \leq \Phi(0) + 0.1\Phi'(0)$$

首先, 取 $\alpha = 1$ 作试探, 有

$$\Phi(\alpha) = \Phi(1) = 0.4946$$

而 $\Phi(0) + 0.1\Phi'(0) = 1.468$

所以, 上面的条件满足。于是第一次线性搜索的结果是

$$x^1 = x^0 + \alpha d^0 = [1.333 \quad 1.167]^T$$

进行下一次求解搜索方向之前, 要更新矩阵 B 。这里采用 BFGS 公式对 B 进行修正,

得
$$B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.3902 & 0.4878 \\ 0.4878 & 0.6098 \end{bmatrix} + \begin{bmatrix} 0.7803 & 0.9751 \\ 0.9751 & 1.2185 \end{bmatrix}$$

或

$$B_1 = \begin{bmatrix} 1.3902 & 0.4873 \\ 0.4873 & 1.6087 \end{bmatrix}$$

从而完成了第一次迭代。如此反复进行第二、三、四次迭代以后, 可得最优点和相应的极值分别为

$$x^* = \begin{Bmatrix} 0.8228769 \\ 0.9114384 \end{Bmatrix}$$

$$f(x^*) = 1.3934$$

§ 6 简约梯度法和广义简约梯度法

一、简约梯度法

在第四章解线性规划问题的单纯形法中, 我们是采取把 n 个变量划分为 m 个基本变量和 $n - m$ 个非基本变量, 即把 x 写成如下形式

$$x = (x_1, x_2, \dots, x_m; x_{m+1}, \dots, x_n) = (x_E, x_F)$$

并且相应地把约束条件组成的方程组的系数矩阵也进行分组, 再通过从一个基本可行解到另一个基本可行解进行逐步求解。而在本章 §1 约束变量法中, 在讨论具有 m 个等式约束的非线性规划问题解法时, 也是把 n 个变量分成两部分来处理, 且相应地把约束条件组成的系数矩阵进行分组, 然后用消去法进行分析求解。

现在把这种解题思路推广应用于约束是线性的，但目标函数是非线性的优化问题。由于目标函数是非线性的，所以不能像线性规划那样，最优点一定处于可行域的顶点上。我们采用搜寻的方法来寻找最优点，即采用公式 $x^{K+1} = x^K + \alpha_K d^K$ 进行迭代求解。这时可以把问题描述为：求

$$x = (x_1, x_2, \dots, x_n)^T$$

使非线性的目标函数

$$f(x) \rightarrow \min, x \in R^n$$

满足线性约束条件

$$Ax = b$$

$$x \geq 0$$

其中 A 是 $m \times n$ 阶实矩阵， b 是 m 维实向量 ($m \leq n$)。

当把 x 分为 $x = (x_E, x_F)^T$ 两部分后，可以在 $x_F \geq 0$ 的条件下任取其值（即 x_F 的 $n-m$ 个分量可为任意正值或零），并把 x_F 看成是自变量（或非基变量），它是一个 $n-m$ 维向量。这样， m 维向量 x_E 可以作为因变量（或基变量）处理。

若系数矩阵相应地分块为

$$A = [E : F]$$

其中 E 是 $m \times m$ 阶矩阵， F 是 $m \times (n-m)$ 阶矩阵，则约束条件可以写成

$$Ax = [E : F] \begin{Bmatrix} x_E \\ x_F \end{Bmatrix} = b$$

或

$$Ex_E + Fx_F = b$$

于是可以解出

$$x_E = E^{-1}(b - Fx_F) = E^{-1}b - E^{-1}Fx_F \quad (5-43)$$

只要 E 是非奇异的，就可以解出 x_E ，当然要求 $x_E \geq 0$ 。

把式 (5-43) 的 x_E 代入目标函数 $f(x)$ ，得

$$f(x) = f(x_E, x_F) = f[x_E(x_F), x_F] = F(x_F)$$

原来的问题变成了一个以 x_F 为变量的 $n-m$ 维无约束优化问题，从而可以在 $n-m$ 维空间里采用梯度法进行迭代求解。这就是简约梯度法的基本思想。它利用某些约束条件消去非独立的变量，以使 n 维约束优化问题转变成 $n-m$ 维无约束优化问题，从而去掉这些约束，所以简约梯度法也是属于把约束问题转化为无约束问题的间接解法。

现在对目标函数 $f(x) = F(x_F)$ 在 $n-m$ 维空间里取梯度。因为此时只有变量 $x_F = (x_{m+1}, \dots, x_n)$ ，所以目标函数 $F(x_F)$ 的梯度是对 x_F 取的。为了区别，令这时的梯度

$$\nabla_F F(x_F) = \left(\frac{\partial F}{\partial x_{F(1)}}, \frac{\partial F}{\partial x_{F(2)}}, \dots, \frac{\partial F}{\partial x_{F(n-m)}} \right)^T$$

称为简约梯度或既约梯度。因为 x_E 是 x_F 的函数，根据复合函数求导公式，有

$$\frac{\partial F}{\partial x_{F(i)}} = \frac{\partial f}{\partial x_{E(i)}} + \sum_{j=1}^m \frac{\partial f}{\partial x_{E(j)}} \frac{\partial x_{E(j)}}{\partial x_{F(i)}} \quad (i=1, 2, \dots, n-m)$$

写成向量形式为

$$\nabla_F F(\mathbf{x}_F) = \nabla_F f(\mathbf{x}) - (\mathbf{E}^{-1}\mathbf{F})^T \nabla_E f(\mathbf{x}) \quad (5-44)$$

其中

$$\nabla_F f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_{F(1)}}, \frac{\partial f}{\partial x_{F(2)}}, \dots, \frac{\partial f}{\partial x_{F(n-m)}} \right)^T$$

$$\nabla_E f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_{E(1)}}, \frac{\partial f}{\partial x_{E(2)}}, \dots, \frac{\partial f}{\partial x_{E(m)}} \right)^T$$

根据梯度法的迭代公式, 当取负梯度方向 $-\nabla_F F(\mathbf{x}_F)$ 为 $n-m$ 维空间中的搜索方向 \mathbf{d}^k 时, 迭代公式可写成

$$\mathbf{x}_F^{K+1} = \mathbf{x}_F^K - \alpha_K \nabla_F F(\mathbf{x}_F) \quad (5-45)$$

为了简化书写, 令 $\mathbf{z}_F = -\nabla_F F(\mathbf{x}_F)$, 且有

$$\mathbf{z}_F^K = (z_{F(m+1)}^K, z_{F(m+2)}^K, \dots, z_{F(n)}^K)^T$$

则 $n-m$ 维空间中的迭代公式写成

$$\mathbf{x}_F^{K+1} = \mathbf{x}_F^K + \alpha_K \mathbf{z}_F^K \quad (5-46)$$

考虑到搜索是沿负梯度方向, 并要求 $\mathbf{x} \geq 0$, 由式 (5-45) 可知, 当 $x_{F(j)}^K = 0$ 和 $\nabla_F F_{(j)}(\mathbf{x}_F^{K+1}) > 0$ 时, 将出现 $x_{F(j)}^{K+1} < 0$ 的可能, 所以要求式 (5-46) 中的

$$\mathbf{z}_{F(j)}^K = \begin{cases} 0 & (\text{当 } x_{F(j)}^K = 0 \text{ 和 } \nabla_F F_{(j)}(\mathbf{x}_F^K) > 0 \text{ 时}) \\ -\nabla_F F_{(j)}(\mathbf{x}_F^K) & (\text{其余情况}) \end{cases} \quad (5-47)$$

解出 \mathbf{x}_F^{K+1} 后可以计算 \mathbf{x}_E^{K+1} 。由式 (5-43) 和 (5-46) 有

$$\mathbf{x}_E^{K+1} = \mathbf{E}^{-1}\mathbf{b} - \mathbf{E}^{-1}\mathbf{F}(\mathbf{x}_F^K + \alpha_K \mathbf{z}_F^K) = \mathbf{E}^{-1}\mathbf{b} - \mathbf{E}^{-1}\mathbf{F}\mathbf{x}_F^K - \alpha_K \mathbf{E}^{-1}\mathbf{F}\mathbf{z}_F^K$$

得到

$$\mathbf{x}_E^{K+1} = \mathbf{x}_E^K - \alpha_K \mathbf{E}^{-1}\mathbf{F}\mathbf{z}_F^K$$

为了对应起见, 记

$$\mathbf{z}_E^K = -\mathbf{E}^{-1}\mathbf{F}\mathbf{z}_F^K \quad (5-48)$$

看作是 m 维空间中的搜索方向, 则有 m 维空间中的迭代公式

$$\mathbf{x}_E^{K+1} = \mathbf{x}_E^K + \alpha_K \mathbf{z}_E^K \quad (5-49)$$

将 $n-m$ 维空间搜索方向 \mathbf{z}_F^K 和 m 维空间搜索方向 \mathbf{z}_E^K 组合起来, 形成 n 维空间搜索方向 $\mathbf{z}^K = (\mathbf{z}_E^K, \mathbf{z}_F^K)^T$, 则在 n 维空间中的迭代公式将是

$$\mathbf{x}^{K+1} = \mathbf{x}^K + \alpha_K \mathbf{z}^K \quad (5-50)$$

式中 α_K 为步长因子, 对它有以下两个要求。

首先, 要求新的 $\mathbf{x}^{K+1} \geq 0$, 即要求满足约束条件

$$\mathbf{x}^{K+1} = \mathbf{x}^K + \alpha_K \mathbf{z}^K \geq 0$$

或

$$x_i^{K+1} = x_i^K + \alpha_K z_i^K \geq 0 \quad (i = 1, 2, \dots, n)$$

这就需要确定 α_K 的取值范围。当 $z_i^K \geq 0$ 时, 上式恒成立, 而当 $z_i^K < 0$ 时应有

$$\alpha_K \leq -\frac{x_i^K}{z_i^K}$$

所以

$$a_{k+1} = \min_{x_i^k < 0} \left(-\frac{x_i^k}{z_i^k} \right) \quad (5-51)$$

其次, 要求 a_k 保证使 $f(x^{k+1})$ 是下降的, 即要求

$$f(x^{k+1}) = f(x^k + a_k z^k) = \min_{1 \leq i \leq n} f(x^k + a_k z^k) \quad (5-52)$$

如果求出的 a_k 满足 $a_k < a_{k+1}$, 说明 x_i^k 的分量仍具有 $x_i^k > 0$ 的性质, 即可将 a_k 代入到 $x^{k+1} = x^k + a_k z^k$ 中求出 x^{k+1} 。如果 $a_k = a_{k+1}$, 这时 x_i^k 的分量有一些是零。我们把满足

$$x_i^k + a_k z_i^k = 0$$

的那些分量从基变量 x_k 中除掉 (即出基), 而从 $x_{i^k}^k$ 中将其数值最大的依次补进基变量 x_k 中来 (即进基), 再重新进行迭代运算。

计算结束的判断是 $\|z^k\| < \epsilon_1$, ϵ_1 是预先给定的大于零的控制误差;

总结上述, 简约梯度法的算法可叙述如下。

给定 x^1 , ϵ_1 , ϵ_2 , 满足 $Ax^1 = b$, $x_1^1 \geq 0$ 。

第一步: 将 x 分为 $x = (x_E, x_F)^T$, 并相应地把 A 分为 $A = (E; F)$, 记 x_E 所对应的下标集合为 I_E , x_F 对应的下标集合为 I_F 。

第二步: 根据式 (5-44) 求简约梯度 $\nabla_{x_F} F(x_F)$ 。

第三步: 对于 $j \in I_F$, 根据式 (5-47) 取 $n-m$ 维空间搜索方向 $z^k_{(j)}$ 。

第四步: 按式 (5-48) 求 m 维空间搜索方向 z^k_E , 并结合 n 维空间搜索方向 $z^k = (z^k_E, z^k_F)^T$ 。若 $\|z^k\| < \epsilon_1$, 则停止, 否则做第五步。

第五步: 按式 (5-51) 计算 a_{k+1} , 按式 (5-52) 计算 a_k 和按式 (5-50) 计算 x^{k+1} 。若 $\|x^{k+1} - x^k\| < \epsilon_2$ 则停止, 否则做第六步。

第六步: 若 $a_k < a_{k+1}$, 则转第二步。若 $a_k = a_{k+1}$, 则进行换基处理, 即找出基向量中 $x_{k(i)}^k = 0$ 的 $i \in I_E$ 者, 将此 i 从 I_E 中除掉, 换入 $\max_{i \in I_F} (x_i^k / z_i^k)$ 的 j 。于是得到新的 I_E 和 I_F , 转第一步。

上述计算步骤可用框图反映出来, 如图 (5-15) 所示。

举例: 用简约梯度法求解下面的约束优化问题。

$$\min f(x) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2$$

s.t.

$$x_1 + x_2 \leq 2$$

$$x_1 + 5x_2 \leq 5$$

$$-x_1 \leq 0$$

$$-x_2 \leq 0$$

引入松弛变量 x_3, x_4 , 把上述约束条件化为规范形式 (即等式约束形式), 则原问题表示为

$$\min f(x) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2$$

s.t.

$$x_1 + x_2 + x_3 = 2$$

$$x_1 + 5x_2 + x_4 = 5$$

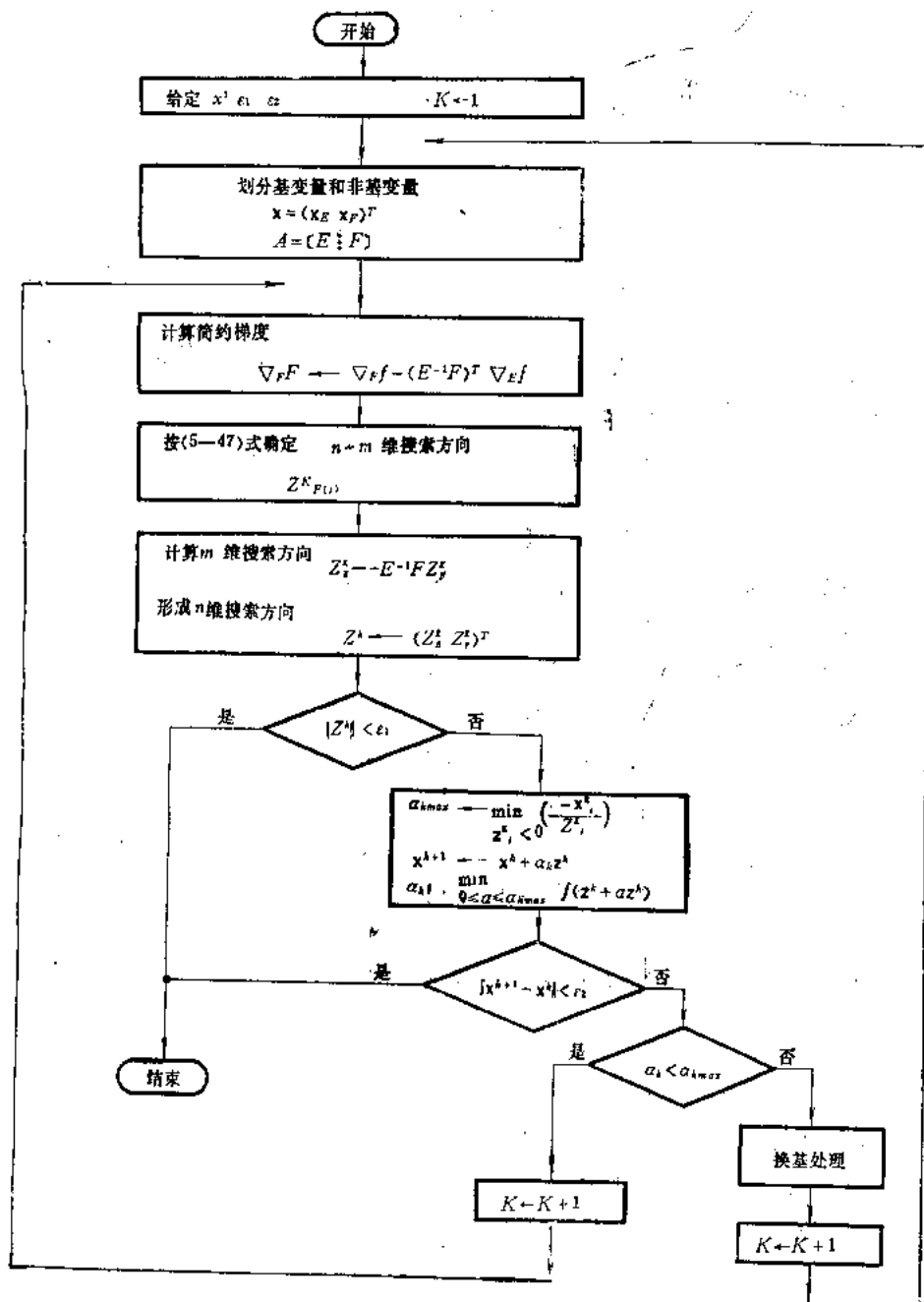


图 5-15

$$x_1, x_2, x_3, x_4 \geq 0$$

系数矩阵

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 5 & 0 & 1 \end{bmatrix} \quad b = \begin{Bmatrix} 2 \\ 5 \end{Bmatrix}$$

取 $\epsilon_1 = \epsilon_2 = 10^{-3}$, 并选初始点 $x^1 = (0, 0, 2, 5)^T$ 取 x^1 中的大者为 x_E , 把 x 分为 $x_E^1 = (x_3, x_4)^T$, $x_F^1 = (x_1, x_2)^T$, 则对第一循环有

$$E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad F = \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix} \quad I_E = (3, 4) \quad I_F = (1, 2)$$

$$\nabla_F f(x^1) = (-4, -6)^T, \quad \nabla_E f(x^1) = (0, 0)^T$$

$$\nabla_F F(x_F^1) = \nabla_F f - (E^{-1}F)^T \nabla_E f = \begin{Bmatrix} -4 \\ -6 \end{Bmatrix} - \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix} \right)^T \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} -4 \\ -6 \end{Bmatrix}$$

$$z_F^1 = -\nabla_F F(x_F^1) = \begin{Bmatrix} 4 \\ 6 \end{Bmatrix}$$

$$z_E^1 = -E^{-1}Fz_F^1 = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix} \begin{Bmatrix} 4 \\ 6 \end{Bmatrix} = \begin{Bmatrix} -10 \\ -34 \end{Bmatrix}$$

所以

$$z^1 = (4, 6, -10, -34)^T$$

此时的 $\|z^1\| > 10^{-3}$, 且 z^1 有负值分量, 所以求

$$a_{1\max} = \min\left\{\frac{2}{10}, -\frac{5}{-34}\right\} = \frac{5}{34}$$

$$f(x^1 + a_1 z^1) = 56a_1^2 - 52a_1$$

求解一维搜索问题

$$\min(56a_1^2 - 52a_1)$$

$$\text{s.t.} \quad 0 \leq a_1 \leq \frac{5}{34}$$

得

$$a_1 = \frac{5}{34}$$

则

$$x^2 = x^1 + \frac{5}{34} z^1 = \left(\frac{10}{17}, \frac{15}{17}, \frac{9}{17}, 0\right)^T$$

$$\|x^2 - x^1\| = \frac{5}{34} \|z^1\| > 10^{-3}$$

第二循环取 x^2 中大者为 x_E , 即 $x_E = (x_1, x_2)^T$, $x_F = (x_3, x_4)^T$, $I_E = (1, 2)$, $I_F = (3, 4)$

则有

$$E = \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix} \quad E^{-1} = \begin{bmatrix} \frac{5}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} \end{bmatrix} \quad F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\nabla_E f(x^2) = \left(-\frac{58}{17}, -\frac{62}{17}\right)^T \quad \nabla_F f(x^2) = (0, 0)^T$$

$$\nabla_F F(x_F^2) = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} - \begin{bmatrix} \frac{5}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} -\frac{58}{17} \\ -\frac{62}{17} \end{Bmatrix} = \begin{Bmatrix} \frac{57}{17} \\ \frac{1}{17} \end{Bmatrix}$$

$$z_F^2 = -\nabla_F F(x_F^2) = -\begin{Bmatrix} \frac{57}{17} \\ \frac{1}{17} \end{Bmatrix}$$

$$z_E^2 = -E^{-1}Fz_F^2 = \begin{pmatrix} \frac{5}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -\frac{57}{17} \\ \frac{1}{17} \end{pmatrix} = \begin{pmatrix} \frac{71}{17} \\ -\frac{14}{17} \end{pmatrix}$$

所以

$$z^2 = \left(\frac{71}{17}, -\frac{14}{17}, -\frac{57}{17}, -\frac{1}{17} \right)^T$$

此时仍有 $\|z^2\| > 10^{-3}$, 且 z^2 有负值分量, 所以计算

$$a_{2max} = \min\left(\frac{15/17}{14/17}, \frac{9/17}{57/17}\right) = \frac{9}{57} = 0.128$$

$$f(x^2 + a_2 z^2) = 43.556a_2^2 - 11.23a_2 - 6.432$$

求解一维搜索问题

$$\begin{aligned} & \min f(x^2 + a_2 z^2) \\ \text{s.t.} \quad & 0 \leq a_2 \leq 0.128 \end{aligned}$$

得

$$a_2 = 0.128$$

于是有

$$x^3 = x^2 + a_2 z^2 = (1.125, 0.775, 0, 0)^T$$

如此, 继续进行计算, 就可得最优解

$$x^4 = (1.125, 0.775, 0, 0)^T$$

其迭代过程如图 (5-16) 所示。

三、广义简约梯度法

广义简约梯度法又称 GRG (Generalized Reduced Gradient) 法。这种方法是将简约梯度法推广应用于具有非线性等式约束和上、下界约束的优化问题上。即它适用于解如下形式的优化问题。求 $x = (x_1, x_2, \dots, x_n)^T$ 使目标函数 $f(x) \rightarrow \min$, 满足约束条件

$$h(x) = 0$$

$$a \leq x \leq c$$

其中 $h(x) = [h_1(x), h_2(x), \dots, h_m(x)]^T$ 为 m 维向量函数。

同线性约束的情况类似, 对于任何可行解 x 可分成基变量 x_E 和非基变量 x_F 两部分, 则上、下界约束也相应分组为

$$a_E \leq x_E \leq c_E$$

$$a_F \leq x_F \leq c_F$$

我们还假定 m 维向量

$$\nabla_E h_i(x) = \left[\frac{\partial h_i(x)}{\partial x_{E1}}, \frac{\partial h_i(x)}{\partial x_{E2}}, \dots, \frac{\partial h_i(x)}{\partial x_{Em}} \right]^T \quad (i = 1, 2, \dots, m)$$

是线性无关的, 因此 $m \times m$ 矩阵

$$\nabla_E h(x) = [\nabla_E h_1(x), \nabla_E h_2(x), \dots, \nabla_E h_m(x)]$$

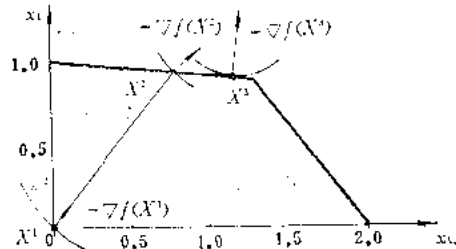


图 5-16

是非奇异的。并记 $n-m$ 维向量

$$\nabla_F h_i(\mathbf{x}) = \left[\frac{\partial h_i(\mathbf{x})}{\partial x_{F_1}}, \frac{\partial h_i(\mathbf{x})}{\partial x_{F_2}}, \dots, \frac{\partial h_i(\mathbf{x})}{\partial x_{F_{(n-m)}}} \right]^T \quad (i=1, 2, \dots, m)$$

和 $(n-m) \times m$ 矩阵

$$\nabla_F \mathbf{h}(\mathbf{x}) = [\nabla_F h_1(\mathbf{x}), \nabla_F h_2(\mathbf{x}), \dots, \nabla_F h_m(\mathbf{x})]$$

同简约梯度法一样，我们可以通过

$$\mathbf{h}(\mathbf{x}) = \mathbf{h}(\mathbf{x}_E, \mathbf{x}_F) = \mathbf{0}$$

求出用 \mathbf{x}_F 表达的 \mathbf{x}_E ，并且将目标函数 $f(\mathbf{x})$ 表示成 \mathbf{x}_F 的函数 $F(\mathbf{x}_F)$ 。应用拉格朗日乘子法，在分别对 \mathbf{x}_E 和 \mathbf{x}_F 求导后得

$$\frac{\partial F(\mathbf{x}_F)}{\partial x_{F_j}} = \frac{\partial f(\mathbf{x})}{\partial x_{F_j}} + \lambda_1 \frac{\partial h_1(\mathbf{x})}{\partial x_{F_j}} + \lambda_2 \frac{\partial h_2(\mathbf{x})}{\partial x_{F_j}} + \dots + \lambda_m \frac{\partial h_m(\mathbf{x})}{\partial x_{F_j}} \quad (j=1, 2, \dots, n-m)$$

$$\frac{\partial F(\mathbf{x}_F)}{\partial x_{E_i}} = \frac{\partial f(\mathbf{x})}{\partial x_{E_i}} + \lambda_1 \frac{\partial h_1(\mathbf{x})}{\partial x_{E_i}} + \lambda_2 \frac{\partial h_2(\mathbf{x})}{\partial x_{E_i}} + \dots + \lambda_m \frac{\partial h_m(\mathbf{x})}{\partial x_{E_i}} \quad (i=1, 2, \dots, m)$$

当引入 m 维向量

$$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)^T$$

则上式可写成如下的向量方程

$$\nabla_F F(\mathbf{x}_F) = \nabla_F f(\mathbf{x}) + \nabla_F \mathbf{h}(\mathbf{x}) \boldsymbol{\lambda}$$

$$\nabla_E F(\mathbf{x}_F) = \nabla_E f(\mathbf{x}) + \nabla_E \mathbf{h}(\mathbf{x}) \boldsymbol{\lambda}$$

由于 $f(\mathbf{x}) = F(\mathbf{x}_F)$ ，自然有 $\nabla_E F(\mathbf{x}_F) = \mathbf{0}$ ，所以从第二式解得

$$\boldsymbol{\lambda} = -[\nabla_E \mathbf{h}(\mathbf{x})]^{-1} \nabla_E f(\mathbf{x})$$

代入到第一式得简约梯度

$$\nabla_F F(\mathbf{x}_F) = \nabla_F f(\mathbf{x}) - \nabla_F \mathbf{h}(\mathbf{x}) [\nabla_E \mathbf{h}(\mathbf{x})]^{-1} \nabla_E f(\mathbf{x}) \quad (5-53)$$

求出简约梯度后，可以类似于线性约束的情形，令搜索方向

$$\mathbf{z}_{F^{(j)}}^K = \begin{cases} 0 & \text{若 } \mathbf{x}_{F^{(j)}}^K = \mathbf{a}_{F^{(j)}} \text{ 且 } \nabla_F F^{(j)}(\mathbf{x}_F^K) > 0 \\ & \text{或 } \mathbf{x}_{F^{(j)}}^K = \mathbf{c}_{F^{(j)}} \text{ 且 } \nabla_F F^{(j)}(\mathbf{x}_F^K) < 0 \\ -\nabla_F F^{(j)}(\mathbf{x}_F^K) & \text{(其余情况)} \end{cases} \quad (5-54)$$

根据 \mathbf{x}_F 为问题最优点的库恩-塔克条件，有（其推导将在第七章中给出）

$$\nabla_F F^{(j)}(\mathbf{x}_F) \begin{cases} = 0 & \text{当 } \mathbf{a}_{F^{(j)}} < \mathbf{x}_{F^{(j)}} < \mathbf{c}_{F^{(j)}} \\ \geq 0 & \text{当 } \mathbf{x}_{F^{(j)}} = \mathbf{a}_{F^{(j)}} \\ \leq 0 & \text{当 } \mathbf{x}_{F^{(j)}} = \mathbf{c}_{F^{(j)}} \end{cases} \quad (5-55)$$

如果 $\mathbf{z}_{F^{(j)}}^K \neq \mathbf{0}$ ，且式 (5-55) 的条件不满足，可按式 (5-50) 进行迭代，即

$$\mathbf{x}_F^{K+1} = \mathbf{x}_F^K + \alpha_K \mathbf{z}_F^K$$

对 \mathbf{x}_F 进行修正。然后再采用牛顿法解以 \mathbf{x}_E 为未知变量的非线性方程组

$$h_i(\mathbf{x}_E^K, \mathbf{x}_F^{K+1}) = 0 \quad (i=1, 2, \dots, m)$$

此时有 $\mathbf{x}_E^{K+1} = \mathbf{x}_E^K - [\nabla_E \mathbf{h}(\mathbf{x}_E^K, \mathbf{x}_F^{K+1})]^{-1} \mathbf{h}(\mathbf{x}_E^K, \mathbf{x}_F^{K+1})$ (5-56)

除上面用牛顿法求解 \mathbf{x}_E^{K+1} 外，还可以采用泰勒展开的方法。将 \mathbf{x}_E^{K+1} 表示成

$$\mathbf{x}_E^{K+1} = \mathbf{x}_E^K + \Delta \mathbf{x}_E$$

要求迭代前后均满足等式约束条件，因此有

$$\begin{aligned} \mathbf{h}(\mathbf{x}_E^K, \mathbf{x}_F^K) &= 0 \\ \mathbf{h}(\mathbf{x}_E^K + \Delta\mathbf{x}_E, \mathbf{x}_F^K + \Delta\mathbf{x}_F) &= 0 \end{aligned}$$

将第二式进行泰勒展开, 有

$$h_i(\mathbf{x}_E^K, \mathbf{x}_F^K) + (\nabla_E h_i)^T \Delta\mathbf{x}_E + (\nabla_F h_i)^T \Delta\mathbf{x}_F = 0 \quad (i=1, 2, \dots, m)$$

写成向量形式为

$$\begin{aligned} (\nabla_E \mathbf{h})^T \Delta\mathbf{x}_E + (\nabla_F \mathbf{h})^T \Delta\mathbf{x}_F &= 0 \\ \Delta\mathbf{x}_E &= -[(\nabla_E \mathbf{h})^T]^{-1} (\nabla_F \mathbf{h})^T \Delta\mathbf{x}_F \end{aligned} \quad (5-57)$$

对变量 \mathbf{x}_F 前面已按简约梯度方向进行一维搜索, 所得结果为

$$\Delta\mathbf{x}_F = \alpha_K \mathbf{z}_F^K$$

那么可求得

$$\begin{aligned} \mathbf{x}_E^{K+1} &= \mathbf{x}_E^K + \Delta\mathbf{x}_E = \mathbf{x}_E^K - [(\nabla_E \mathbf{h})^T]^{-1} (\nabla_F \mathbf{h})^T \Delta\mathbf{x}_F \\ &= \mathbf{x}_E^K - \alpha_K [(\nabla_E \mathbf{h})^T]^{-1} (\nabla_F \mathbf{h})^T \mathbf{z}_F^K \end{aligned}$$

若令

$$\mathbf{z}_E^K = -[(\nabla_E \mathbf{h})^T]^{-1} (\nabla_F \mathbf{h})^T \mathbf{z}_F^K \quad (5-58)$$

则对于基变量 \mathbf{x}_E^{K+1} 也可写成和非基变量 \mathbf{x}_F^{K+1} 相似的迭代公式

$$\mathbf{x}_E^{K+1} = \mathbf{x}_E^K + \alpha_K \mathbf{z}_E^K \quad (5-59)$$

通常, 当 \mathbf{x}_F 取值为 $\mathbf{a}_F \leq \mathbf{x}_F \leq \mathbf{c}_F$ 时, 满足约束条件的 \mathbf{x}_F 应取值为 $\mathbf{a}_E \leq \mathbf{x}_E \leq \mathbf{c}_E$ 。若 $\mathbf{a}_E < \mathbf{x}_E^{K+1} < \mathbf{c}_E$, 则转入下一步迭代。若对于 \mathbf{x}_E 的某一个分量有 $x_E^{K+1}(j) = \mathbf{a}_{E(j)}$ 或 $x_E^{K+1}(j) = \mathbf{c}_{E(j)}$, 就要进行换基处理, 即把 $x_{E(j)}$ 归入非基变量 \mathbf{x}_F 中去, 同时用 \mathbf{x}_F 中的大者 $x_{F(i)}$ 代替 $x_{E(j)}$ 归入基变量 \mathbf{x}_E 中去。

下面列出广义简约梯度法计算的大致步骤。

首先, 选取一个基本容许解 (初始可行解) $\mathbf{x}^0 = (\mathbf{x}_E^0, \mathbf{x}_F^0)^T$ 。

第一步: 根据式 (5-53) 计算简约梯度 $\nabla_F F(\mathbf{x}_F^K)$ 。

第二步: 由式 (5-54) 计算出 $\mathbf{z}_{F(j)}^K$, 再计算相应的 \mathbf{x}_F^{K+1} 。

第三步: 用牛顿法解非线性方程组 $\mathbf{h}(\mathbf{x}_E^K, \mathbf{x}_F^{K+1}) = 0$, 即按(5-56)式计算 \mathbf{x}_E^{K+1} 。或者采用泰勒展开方法按式 (5-59) 计算 \mathbf{x}_E^{K+1} 值。

第四步: 调整变量, 进行换基, 再返回到第一步。

需要说明两点: 第一, 在解非线性方程组 $\mathbf{h}(\mathbf{x}_E, \mathbf{x}_F^{K+1}) = 0$ 时, 若用牛顿法则每一步都要计算 $[(\nabla_E \mathbf{h}(\mathbf{x}_E^K, \mathbf{x}_F^{K+1}))^T]^{-1}$, 工作量很大。此时可以用 $[(\nabla_E \mathbf{h}(\mathbf{x}^K))^T]^{-1}$ 来代替它。因为在用式 (5-53) 计算 $\nabla_F F(\mathbf{x}_F^K)$ 时已经计算过 $[(\nabla_E \mathbf{h}(\mathbf{x}^K))^T]^{-1}$ 了, 所以并不增加额外工作量。第二, 如果迭代一次后基不改变, 则搜索方向可以不用式 (5-54) 计算, 而像在无约束问题中那样, 用变尺度法中的方向来代替这里的梯度法中的方向, 以加速收敛, 加速对变量 \mathbf{x}^K 的修改。

广义简约梯度法的计算程序框图和简约梯度法的相似, 这里就不再作介绍。

在广义简约梯度法中, 根据确定 \mathbf{z}_F^K 的方法不同, 又具体分为 GRG 法, GRGS 法和 GRGC 法等。

广义简约梯度法目前已成为解一般非线性规划问题最有效的方法之一。某些数值计算表明, 此法在计算精度和收敛速度方面都较好。特别是对大型问题更具有优越性。

举例：用GRG法计算下面的约束优化（非线性规划）问题。

$$\begin{aligned} \min f(x) &= 4x_1 - x_2^2 - 12 \\ \text{s.t.} \quad h_1(x) &= 25 - x_1^2 - x_2^2 = 0 \\ g_1(x) &= x_1^2 + x_2^2 - 10x_1 - 10x_2 + 34 \leq 0 \\ g_2(x) &= -x_1 \leq 0 \\ g_3(x) &= -x_2 \leq 0 \end{aligned}$$

此问题的可行域及目标函数的等值线如图(5-17)所示,从图中可看出最优点应处于曲线 $h_1(x) = 0$ 和 $g_1(x) = 0$ 的交点处,即 $x^* = (1.00128, 4.89872)^T$ 现用GRG法对最优点进行搜索。

取初始点 $x^0 = (2, 4)^T$, 如图所示它不是可行点。为了加速目标函数的极小化进程, 在目标函数中加入惩罚项 $-10^5 x_3$, 其中 x_3 为取负值的人为变量, 即有 $-10^{10} \leq x_3 \leq 0$ 。并将此人为变量 x_3 同时加入到约束函数 $h_1(x)$ 中, 形成约束条件 $h_1(x) = 25 - x_1^2 - x_2^2 + x_3 = 0$ (类似情况可参考第四章§5单纯形法应用举例)。当 $x_3 \rightarrow 0$ 时目标函数和约束函数都恢复到原问题形式。

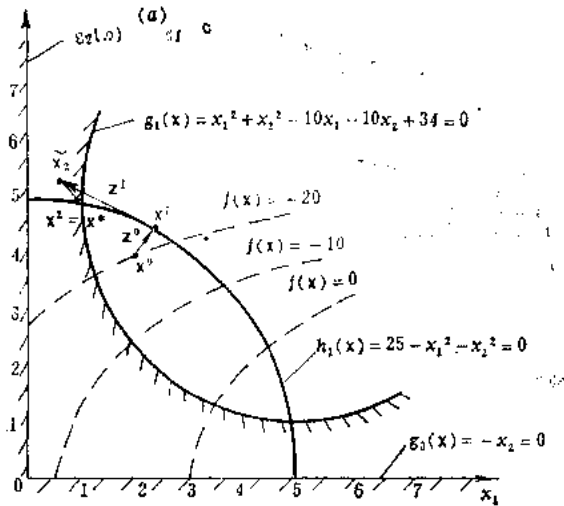


图 5-1.

为了将不等式约束函数 $g_1(x)$ 转化为等式约束函数 $h_2(x)$ 形式, 还需引入非负的松弛变量 x_4 , 这样原来的约束优化问题变为GRG的标准形式:

$$\begin{aligned} \min f(x) &= 4x_1 - x_2^2 - 10^5 x_3 - 12 \\ \text{s.t.} \quad h_1(x) &= 25 - x_1^2 - x_2^2 + x_3 = 0 \\ h_2(x) &= x_1^2 + x_2^2 - 10x_1 - 10x_2 + x_4 + 34 = 0 \\ 0 &\leq x_1 \leq 10^{10} & 0 &\leq x_2 \leq 10^{10} \\ -10^{10} &\leq x_3 \leq 0 & 0 &\leq x_4 \leq 10^{10} \end{aligned}$$

现在原问题变为有 x_1, x_2, x_3, x_4 四个变量和 $h_1(x) = 0, h_2(x) = 0$ 两个等式约束的优化问题, 因而就应有两个自变量和两个因变量。取 x_1, x_2 为自变量(非基变量), x_3, x_4 为因变量(基变量), 即有

$$\begin{aligned} x_F &= (x_1, x_2)^T \\ x_B &= (x_3, x_4)^T \end{aligned}$$

取初始点 $x^0 = (2, 4, -5, 6)^T$, 代入上述GRG标准形式中知为可行点。设 $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 10^{-4}$, 为了计算初始点 x^0 的简约梯度 $\nabla_F F(x^0)$, 需计算下列各量:

$$\begin{aligned} \nabla_F f(x^0) &= \left\{ \begin{array}{c} 4 \\ -2x_2 \end{array} \right\}_{x^0} = \left\{ \begin{array}{c} 4 \\ -8 \end{array} \right\} \\ \nabla_E f(x^0) &= \left\{ \begin{array}{c} -10^5 \\ 0 \end{array} \right\} \end{aligned}$$

$$\nabla_F h(x^0) = \begin{bmatrix} -2x_1 & 2x_1 - 10 \\ -2x_2 & 2x_2 - 10 \end{bmatrix}_{x^0} = \begin{bmatrix} -4 & -6 \\ -8 & -2 \end{bmatrix}$$

$$\nabla_E h(x^0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

根据式 (5-53) 计算 x^0 处的简约梯度

$$\begin{aligned} \nabla_F F(x^0) &= \nabla_F f(x^0) - \nabla_F h(x^0) [\nabla_E h(x^0)]^{-1} \nabla_E f(x^0) \\ &= \begin{Bmatrix} 4 \\ -8 \end{Bmatrix} - \begin{bmatrix} -4 & -6 \\ -8 & -2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{Bmatrix} -10^5 \\ 0 \end{Bmatrix} \\ &= \begin{Bmatrix} 4 \\ -8 \end{Bmatrix} - \begin{Bmatrix} 4 \times 10^5 \\ 8 \times 10^5 \end{Bmatrix} = \begin{Bmatrix} -399996 \\ -800008 \end{Bmatrix} \end{aligned}$$

显然 $\|\nabla_F F(x^0)\| > \varepsilon_1$, 需进行搜索。

搜索方向的计算: 自变量的搜索方向为

$$z_F^0 = -\nabla_F F(x^0) = (399996 \quad 800008)^T$$

而因变量的搜索方向需根据式 (5-58) 计算:

$$\begin{aligned} z_E^0 &= -[\nabla_E h(x^0)]^{-1} [\nabla_F h(x^0)]^T z_F^0 \\ &= -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} -4 & -6 \\ -8 & -2 \end{bmatrix}^T \begin{Bmatrix} 399996 \\ 800008 \end{Bmatrix} \\ &= \begin{bmatrix} 4 & 8 \\ 6 & 2 \end{bmatrix} \begin{Bmatrix} 399996 \\ 800008 \end{Bmatrix} = \begin{Bmatrix} 8000048 \\ 3999992 \end{Bmatrix} \end{aligned}$$

所以在 x_1, x_2, x_3, x_4 设计空间中的搜索方向为

$$z^0 = (z_F^0, z_E^0)^T = (399996 \quad 800008 \quad 8000048 \quad 3999992)^T$$

利用设计变量的上、下界限判断步长 α , 得

$$\alpha_0 = \frac{5}{8000048} = 6.2499625 \times 10^{-7}$$

则第一次走步后到达设计点 x^1 为

$$x^1 = x^0 + \alpha_0 z^0 = \begin{Bmatrix} 2 \\ 4 \\ -5 \\ 6 \end{Bmatrix} + 6.2499625 \times 10^{-7} \begin{Bmatrix} 399996 \\ 800008 \\ 8000048 \\ 3999992 \end{Bmatrix} = \begin{Bmatrix} 2.25 \\ 4.5 \\ 0 \\ 8.5 \end{Bmatrix}$$

x^1 不是可行点, 需应用式 (5-56)

$$\tilde{x}_E^1 = x_E^1 - [\nabla_E h(x_E^1, x_F^1)]^{-1} h(x_E^1, x_F^1)$$

调整因变量, 以便使 x^1 成为可行点 (在这里, 可行点的获得实际上分成两步走, 先沿约束切平面移动而超出可行域, 然后再用牛顿法将设计点拉回到可行域内)。由于 α 值是根据 x_3 取上界限值而定, 所以在按 (5-56) 式调整因变量时取 x_2, x_4 作为因变量, 即

$$x_E^1 = \begin{Bmatrix} x_2^1 \\ x_4^1 \end{Bmatrix} = \begin{Bmatrix} 4.5 \\ 8.5 \end{Bmatrix}$$

$$x_F^1 = \begin{Bmatrix} x_1^1 \\ x_3^1 \end{Bmatrix} = \begin{Bmatrix} 2.25 \\ 0 \end{Bmatrix}$$

$$h(x_E^1, x_F^1) = \begin{Bmatrix} 25 - 2 \cdot 2.25^2 - 4 \cdot 5^2 \\ 2 \cdot 2.25^2 + 4 \cdot 5^2 - 10 \times 2.25 - 10 \times 4.5 + 8.5 + 34 \end{Bmatrix} = \begin{Bmatrix} -0.3125 \\ 0.3125 \end{Bmatrix}$$

$$\nabla_E h(x_E^1, x_F^1) = \begin{bmatrix} -2x_2 & 2x_2 \\ 0 & 1 \end{bmatrix}_{x_2=4.5} = \begin{bmatrix} -9 & 9 \\ 0 & 1 \end{bmatrix}$$

根据 (5-56) 式得

$$\begin{aligned} \begin{Bmatrix} \tilde{x}_2^1 \\ \tilde{x}_4^1 \end{Bmatrix} &= \begin{Bmatrix} x_2^1 \\ x_4^1 \end{Bmatrix} - \begin{bmatrix} -9 & 9 \\ 0 & 1 \end{bmatrix}^{-1} \begin{Bmatrix} -0.3125 \\ 0.3125 \end{Bmatrix} = \begin{Bmatrix} 4.5 \\ 8.5 \end{Bmatrix} - \begin{bmatrix} -\frac{1}{9} & 1 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} -0.3125 \\ 0.3125 \end{Bmatrix} \\ &= \begin{Bmatrix} 4.5 \\ 8.5 \end{Bmatrix} - \begin{Bmatrix} 0.3472 \\ 0.3125 \end{Bmatrix} = \begin{Bmatrix} 4.1528 \\ 8.1875 \end{Bmatrix} \end{aligned}$$

经过第一次调整因变量后, $x^1 = (2.25, 4.1528, 0, 8.1875)^T$ 仍为不可行点, 按牛顿法公式

$$\tilde{x}_E^1 = \tilde{x}_E^1 - [\nabla_E h(\tilde{x}_E^1, x_F^1)]^{-1} h(\tilde{x}_E^1, x_F^1)$$

继续调整因变量 $x_E^1 = (x_2, x_4)^T$, 直到 x^1 为可行点为止。最后得到可行点。

$$x^1 = (2.25, 4.465, 0, 8.151)^T$$

对自变量 x_1, x_3 在 x^1 处的简约梯度为

$$\begin{aligned} \nabla_F F(x^1) &= \nabla_F f(x^1) - \nabla_F h(x^1) [\nabla_E h(x^1)]^{-1} \nabla_E f(x^1) \\ &= \begin{Bmatrix} 4 \\ -10^5 \end{Bmatrix} - \begin{bmatrix} -4.5 & -5.5 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -8.93 & -1.07 \\ 0 & 1 \end{bmatrix}^{-1} \begin{Bmatrix} -8.93 \\ 0 \end{Bmatrix} \\ &= \begin{Bmatrix} 4 \\ -10^5 \end{Bmatrix} - \begin{Bmatrix} -4.5 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 8.5 \\ -10^5 \end{Bmatrix} \end{aligned}$$

因为 x_3^1 位于其上界处, 且 $\nabla_F F_3(x^1) < 0$, 根据式 (5-54) 取 $z_3^1 = 0, z_1^1 = -\nabla_F F_1(x^1)$, 即

$$\begin{aligned} z_F^1 &= (z_1^1, z_3^1)^T = (-8.5, 0)^T \\ z_N^1 &= (z_2^1, z_4^1)^T = -[\nabla_E h(x^1)]^{-1} [\nabla_F h(x^1)]^T z_F^1 \\ &= -\begin{bmatrix} -8.03 & 0 \\ -1.07 & 1 \end{bmatrix}^{-1} \begin{bmatrix} -4.5 & -5.5 \end{bmatrix}^T \begin{Bmatrix} -8.5 \\ 0 \end{Bmatrix} \\ &= \begin{Bmatrix} 4.283 \\ -42.165 \end{Bmatrix} \end{aligned}$$

从而得到 x^1 处的搜寻方向

$$z^1 = (z_1^1, z_2^1, z_3^1, z_4^1)^T = (-8.5, 4.283, 0, -42.165)^T$$

利用上、下界约束求步长, 得

$$\alpha_1 = \min \left\{ \frac{2.25}{-8.5}, \frac{8.151}{-42.165} \right\} = \min(0.2647, 0.1933) = 0.1933$$

$$\tilde{x}^2 = x^1 + \alpha_1 z^1 = \begin{Bmatrix} 2.25 \\ 4.465 \\ 0 \\ 8.151 \end{Bmatrix} + 0.1933 \begin{Bmatrix} -8.5 \\ 4.283 \\ 0 \\ -42.165 \end{Bmatrix} = \begin{Bmatrix} 0.607 \\ 5.293 \\ 0 \\ 0 \end{Bmatrix}$$

x^2 为不可行点, 取因变量 x_1, x_2 按 (5-56) 式进行因变量调整, 最后得可行点

$$x^2 = (1.00128, 4.89872, 0, 0)^T$$

对自变量 x_3, x_4 在 x^2 处的简约梯度为

$$\begin{aligned} \nabla_F F(x^2) &= \nabla_F f(x^2) - \nabla_F h(x^2) [\nabla_E h(x^2)]^{-1} \nabla_E f(x^2) \\ &= \begin{Bmatrix} -10^5 \\ 0 \end{Bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -2.00256 & -7.99744 \\ -9.7974 & -0.2026 \end{bmatrix}^{-1} \begin{Bmatrix} 4 \\ -9.7974 \end{Bmatrix} \end{aligned}$$

$$= \begin{Bmatrix} -10^5 \\ 0 \end{Bmatrix} - \begin{Bmatrix} 1.0156 \\ -0.754 \end{Bmatrix} = \begin{Bmatrix} -10^5 \\ 0.754 \end{Bmatrix}$$

由于自变量 x_3 位于其上界处,且 $\nabla_{F_3}(x^2) < 0$,而自变量 x_4 位于其下界处,且 $\nabla_{F_4}(x^2) > 0$,根据式(5-54)取 $z_{F^2} = (0, 0)^T$ 。根据式(5-58)得 $z_{E^2} = (0, 0)^T$,从而得在 x^2 处的搜索方向

$$z^2 = (z_{E^2}, z_{F^2})^T = (0, 0, 0, 0)^T$$

所以 x^2 为极小点,即

$$x^* = x^2 = (1.00128, 4.89872, 0, 0)^T$$

相应的极小值为 $f(x^*) = -31.9923$

在全部计算过程中,计算目标函数和约束函数值共22次,计算目标函数梯度和约束导数共11次,中间经过了两次换基。

第六章 非线性规划解法之二

——直接方法

在前一章中，我们介绍了非线性规划的间接解法，它是把约束结合到目标函数中形成一个广义目标函数，再用无约束优化方法求解的。这种方法实际上是对约束作了间接的处理。与此相反，本章将讨论的方法，是把约束作为限制设计点的曲面或子空间，在此曲面或子空间范围内直接用无约束优化方法来求解约束优化问题的。显然这时需要考虑约束的影响。其具体做法是，首先寻找一个指向约束极值点的搜寻方向，然后沿此方向进行一维搜索，并对步长加以限制以使设计点不超出可行域。当搜索到达边界时再重复上述步骤。可见，非线性规划问题的直接解法，可以看成是无约束优化方法对于约束问题的推广，从而可以建立直接方法与无约束优化方法的对应关系。例如，直接方法中的可行方向法对应于无约束优化方法中的最速下降法；复形法对应于无约束优化方法中的单形替换法；梯度投影法则是将最速下降法应用到具有线性约束的非线性规划问题上。

§ 1 可行方向法

在非线性规划问题的直接方法中，可行方向法是最大的一类。可行方向法采用的迭代公式和无约束优化问题时的形式相同，都是

$$x^{K+1} = x^K + \alpha_K d^K$$

但是，在无约束优化问题中，由于没有约束条件的限制，搜索方向 d^K 都是可行的。这里则不同。可行方向法在确定可行方向 d^K 和步长因子 α_K 时要考虑约束条件，即这里的搜索方向 d^K 和步长因子 α_K 总应该选得使 x^{K+1} 在可行域内。可见可行方向法就是在一系列可行方向上渐次移动，从而得到一系列逐步改进的可行点 x^{K+1} 的迭代方法。它也是由两部分组成：首先，在每次迭代时须选一个可行方向 d^K ；其次，还须确定步长。一般是从点 x^K 出发，先沿目标函数梯度 $\nabla f(x^K)$ 方向前进，当遇到约束曲面的边界时就改变方向，否则将穿越约束边界进入不可行区域。

对于极小化问题，可行方向 d^K 的选取最少要满足如下的两点要求：

1. 可行点 x^K 沿着它移动时，目标函数值应下降，或至少不变。

当把目标函数在点 x^K 处泰勒展开时，有

$$f(x) = f(x^K) + (x - x^K)^T \nabla f(x^K) + o(\|x - x^K\|)$$

只要 $x - x^K$ 很小，并记 $x - x^K = ad(a > 0)$ 时，若

$$d^T \nabla f(x^K) < 0$$

就一定有 $f(x) < f(x^K)$ 。所以，只要

$$d^T \nabla f \leq 0 \quad (\text{或} \quad \nabla f^T d \leq 0) \quad (6-1)$$

d 就是函数 $f(x)$ 在点 x^K 处的下降方向（这个条件也可直接从方向导数的概念获

得)。当然，这里的 \mathbf{d} 和无约束问题时的 \mathbf{d} 是不同的，它是处在由所有起作用约束曲面的梯度所张开的空间内。

2. 可行点 \mathbf{x}^K 沿着它移动时，不能破坏约束条件，即不能离开可行域。为此要求

$$\mathbf{d}^T \nabla g \leq 0 \quad (\nabla g \text{ 是 } J \text{ 维向量}) \quad (6-2)$$

它保证 \mathbf{d} 是可行方向。

这个条件可以说明如下。

假定约束条件是不等式的。当 $0 \leq \alpha \leq \bar{\alpha}$ (而 $\bar{\alpha} > 0$) 时，应有

$$g_j(\mathbf{x} + \alpha \mathbf{d}) \leq 0 \quad (j = 1, 2, \dots, m)$$

在 \mathbf{x} 点处把此函数进行泰勒展开，有

$$g_j(\mathbf{x} + \alpha \mathbf{d}) = g_j(\mathbf{x}) + \alpha \mathbf{d}^T \nabla g_j(\mathbf{x}) + o(\alpha)$$

其中 $o(\alpha)$ 表示比 α 高阶的无穷小。从上式可知，当 $g_j(\mathbf{x}) < 0$ (α 充分小时)，总有

$$g_j(\mathbf{x} + \alpha \mathbf{d}) \leq 0$$

若对某些 j ， $g_j(\mathbf{x}) = 0$ ，则只要

$$\mathbf{d}^T \nabla g_j(\mathbf{x}) < 0$$

\mathbf{d} 就是可行方向。

若函数 $g_j(\mathbf{x})$ 是线性的，则

$$\mathbf{d}^T \nabla g_j(\mathbf{x}) \leq 0$$

是 \mathbf{d} 为可行方向的充要条件。

当 \mathbf{d} 同时满足式(6-1)和(6-2)时，它就是可行下降方向。它与目标函数的梯度 Δf 及约束函数的梯度 ∇g 之间的夹角都大于或等于 90° (或与负梯度 $-\nabla f$ ， $-\nabla g$ 方向的夹角小于 90°)。即有

$$\theta(\nabla f, \mathbf{d}) \geq 90^\circ \quad \text{和}$$

$$\theta'(\nabla g, \mathbf{d}) \geq 90^\circ$$

所以，可行方向 \mathbf{d}^K 将处于图(6-1)中阴影区所示的锥内。

根据对极小化问题的上述要求，可以有不同的形成可行方向 \mathbf{d}^K 的方法。

例如，若在 \mathbf{x}^K 处没有起作用的约束(即 J 是空集)，则 \mathbf{d}^K 和 $-\nabla f$ 就可以是同一个方向。即 \mathbf{d}^K 可以选取最速下降方向。

也可以采用满足式(6-1)和(6-2)的随机向量的办法形成可行方向 \mathbf{d}^K 。

考虑到一般结构优化设计方案要使结构性能得到最充分的利用，就应使尽可能多的约束达到临界状态。这就意味着要使设计的优化点位于约束曲面上或几个起作用的约束曲面的交集上。因此，也可以把 $-\nabla f$ 投影到所有起作用约束曲面的交集上去的方法，即用梯度投影方法来形成可行方向 \mathbf{d}^K 。

在确定可行方向以后，还要解决步长选取的问题。对于极小化问题，要求在保证 \mathbf{x}^{K+1} 位于可行域内的前提下，函数值尽可能地减小，而且希望不必计算很多个约束 g_j 的值。可能有以下两种情况。

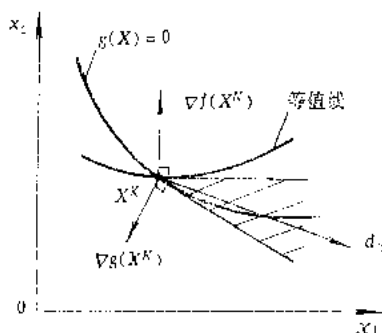


图 6-1

1. 在点 x^{k+1} 处, 有一个或多个约束函数 g_i 是起作用的, 如图 (6-2a) 所示。

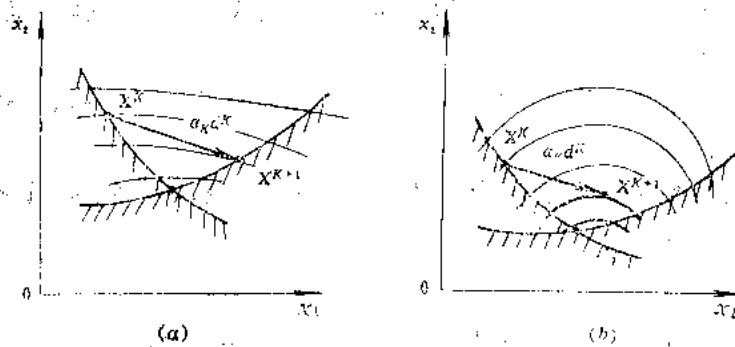


图 6-2

这时, 通常采用试凑方法选取步长。而为了简化计算, 可以给约束曲面取成具有某个“厚度”, 即只要满足

$$-\epsilon \leq g_i \leq 0 \quad (6-3)$$

时, 就认为该约束是起作用的约束了。

2. 点 x^{k+1} 可以是关于 a 的一维无约束极小点, 如图 (6-2b) 所示。这时可以像无约束优化问题那样, 采用一种无约束极小化算法, 直到又碰到约束曲面为止。

在可行方向中, 每迭代一次需要解一个线性规划问题, 其目的只是为了求一个可行方向。对于线性约束问题, 可以采用梯度投影法以简化问题的求解。

§ 2 梯度投影法

一、基本思想

梯度投影法是在这样一个想法的支配下提出的, 那就是既希望实现可行方向法的原理, 而又不要每步都去解一个线性规划问题。这种方法和可行方向法一样, 首先要求迭代点必须在可行域内, 其次在每一个迭代点处都试图利用目标函数的负梯度来产生一个下降的可行方向。它们的主要区别在于方向如何计算确定。当迭代点处的目标函数负梯度方向同时是可行方向时, 自然就取这个方向为下降可行方向。当目标函数负梯度方向不是可行方向, 即 $-\nabla f(x)$ 指向可行域之外时, 也可以采用将负梯度 $-\nabla f(x)$ 向边界投影的办法, 来产生一个下降可行方向 d 。这种梯度投影法在迭代过程中只考虑迭代点 x 处起作用的不等式约束 (也包括等式约束), 每一个起作用约束在几何上都对应一个超曲面。如果把 $-\nabla f(x)$ 向所有这些超曲面的交集上投影, 那么通过投影所得到的向量 d 就属于所有超曲面, 从而它的方向就是一个可行方向。当 d 是非零向量, 同时 d 与 $-\nabla f(x)$ 的点积大于零时, 方向 d 就是一个下降方向。那么, 只要保持搜索点在可行域内的前提下, 沿着这个投影方向 d 作一维搜索, 就可达到目标函数值在这一方向上的极小点, 或者遇到可行域的一个新的边界 (即某个或某几个新的不等式约束变成起作用约束)。这样就可以求到新的迭代点, 以此方式不断进行迭代, 直到求得问题的优化解为止。所以, 梯度投影法就是利用目标函数负梯度 $-\nabla f(x)$ 在迭代点 x 起作用约

束曲面交集上的投影，来求解约束优化问题的直接方法。虽然这个方法是对一般的非线性规划问题叙述的，但它的效验主要限于约束函数是线性函数的问题，因此本节假定全部约束都是线性的。

在第五章 §2 中曾计算过 n 维优化问题极值点处具有 m 个起作用约束，当 $n > m$ 时的拉格朗日乘子向量 μ 。

设 $g_j(x)$ 为在 x 点处等式和起作用不等式的约束函数，共 m 个。当 $g_j(x)$ 为线性函数时， $\frac{\partial g_j}{\partial x_i}$ 为常数，令

$$a_{ij} = \frac{\partial g_j}{\partial x_i} \quad (i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m)$$

$$g_j(x) = \sum_{i=1}^n a_{ij} x_i - c_j \quad (j = 1, 2, \dots, m)$$

若 x 为极值点，根据库恩—塔克条件有

$$-\frac{\partial f}{\partial x_i} = \sum_{j=1}^m \mu_j \frac{\partial g_j}{\partial x_i} \quad (i = 1, 2, \dots, n)$$

记
$$\nabla g_j = \left[\frac{\partial g_j}{\partial x_1}, \frac{\partial g_j}{\partial x_2}, \dots, \frac{\partial g_j}{\partial x_n} \right]^T$$

$$A_j = [a_{1j}, a_{2j}, \dots, a_{nj}]^T = \nabla g_j$$

$$A = [A_1, A_2, \dots, A_m] = [a_{ij}]_{n \times m}$$

$$b_i = -\frac{\partial f}{\partial x_i} \quad (i = 1, 2, \dots, n)$$

$$B = \left[-\frac{\partial f}{\partial x_1}, -\frac{\partial f}{\partial x_2}, \dots, -\frac{\partial f}{\partial x_n} \right]^T = -\nabla f(x)$$

则上式可写成

$$\sum_{j=1}^m a_{ij} \mu_j = b_i \quad (i = 1, 2, \dots, n)$$

或

$$A\mu = B \quad (6-4)$$

其中乘子向量 $\mu = [\mu_1, \mu_2, \dots, \mu_m]^T$

式 (6-4) 是 m 个变量 μ_j 的 n 个方程的方程组，当 $n > m$ 时，可采用最小二乘法求解，其解为

$$\mu = (A^T A)^{-1} A^T B$$

代入式 (6-4) 得

$$[I - A(A^T A)^{-1} A^T] B = 0 \quad (6-5)$$

或

$$PB = 0$$

其中的

$$P = I - A(A^T A)^{-1} A^T$$

在迭代点 x 处，各个起作用约束曲面 $g_j(x)$ 是和相应的向量 A_j 相垂直的一个超平面。

而诸约束曲面在点 x 处有一交集。若称目标函数负梯度向量 $B = -\nabla f$ 向这个交集上的投影为梯度投影, 并记作 B_p 。由于 B_p 处于交集上, 它应和各 A_j 正交, 即有 $A_j^T B_p = 0$ ($j=1, 2, \dots, m$)。因此, B_p 应当等于 B 减去所有和各 A_j 平行的分量, 即

$$B_p = B - \sum_{j=1}^m \mu_j A_j = B - A\mu \quad (6-6)$$

利用 $A_j^T B_p = 0$, 则式 (6-6) 可写成

$$A_j^T \left(B - \sum_{j=1}^m \mu_j A_j \right) = A_j^T (B - A\mu) = 0 \quad (j=1, 2, \dots, m)$$

或写成矩阵形式

$$A^T B - A^T A\mu = 0$$

则得

$$\mu = (A^T A)^{-1} A^T B$$

这和前面用最小二乘法求解所得的结果式 (6-5) 一致。把此值代入式 (6-6), 得

$$\begin{aligned} B_p &= B - A(A^T A)^{-1} A^T B \\ &= [I - A(A^T A)^{-1} A^T] B \\ &= PB \end{aligned}$$

可见, 目标函数梯度投影 B_p 可以通过它的梯度 B 和矩阵 P 求得。因此, 矩阵 $P = I - A(A^T A)^{-1} A^T$ 称为投影矩阵。

库恩——塔克条件实际上是要求在极值点处梯度投影 B_p 为零向量, 并且还要求与起作用约束相对应的各算子 μ_i 为非负。如果不满足上述极值条件, 则仍应沿梯度投影向量

$$d^K = B_p = PB = -P\nabla f(x^K)$$

的方向进行走步, 即在交集子空间 ($n-m$ 维) 中按下式进行搜寻,

$$x^{K+1} = x^K + \alpha_K d^K = x^K - \alpha_K P\nabla f(x^K) \quad (6-7)$$

当各起作用的约束函数表示为

$$g_j(x) = \sum_{i=1}^n \frac{\partial g_j}{\partial x_i} x_i - c_j = \sum_{i=1}^n a_{ij} x_i - c_j$$

或写成向量表达式形式为

$$G(x) = A^T x - C$$

则若迭代点 x^K 沿梯度投影方向进行走步到达 $x^{K+1} = x^K - \alpha_K P\nabla f(x^K)$ 点, 那么在 x^K 点处起作用的约束在 x^{K+1} 点处仍起作用。因为

若 $A^T x^K - C = 0$ (0 代表零向量)

则
$$\begin{aligned} A^T x^{K+1} - C &= A^T (x^K - \alpha_K P\nabla f) - C \\ &= A^T \{x^K - \alpha_K [I - A(A^T A)^{-1} A^T] \nabla f\} - C \\ &= A^T x^K - C - \alpha_K [A^T - A^T A (A^T A)^{-1} A^T] \nabla f \\ &= 0 \end{aligned}$$

也就是说, 如果 x^K 位于由 m 个线性约束所确定的超平面交集内, 那么按式 (6-7) 移动所得到新的迭代点 x^{K+1} 仍旧位于这个交集内。

二、步长的计算

如果目标函数 $f(x)$ 在从 x^K 沿 $d^K = B_p$ 走步过程中, 在可行域内没有极小点, 那么在使用梯度投影法时我们就把迭代点尽可能地移远, 直到碰到一个或多个新的约束曲面, 也就是说从 x^K 移到 x^{K+1} 是在保证 x^{K+1} 可行的前提下选择最大的 α 。然后在新的迭代点处目标函数负梯度 $-\nabla f$ 再对起作用约束曲面的交集作投影。

由于约束函数 $g_j(x)$ 是关于 x 的线性函数, 所以迭代点在沿梯度投影 B_p 方向移动过程中, $g_j(x)$ 也是 α 的线性函数, 即

$$g_j(x) = A_j^T x - C_j$$

变成

$$\begin{aligned} g_j(\alpha) &= A_j^T (x^K + \alpha B_p) - C_j \\ &= g_j(x^K) + \alpha A_j^T B_p \end{aligned}$$

对于所有在 x^K 处不起作用的约束, 我们可以算出使 $g_j(\alpha) = 0$ 的 α_j 值, 即

$$\alpha_j = -\frac{g_j(x^K)}{A_j^T B_p} = -\frac{g_j(x^K)}{A_j^T d^K} \quad (6-8)$$

因为这种计算只是对那些不起作用的约束进行的, 而且 x^K 又是可行点, 因此 $g_j(x^K) < 0$, 从而 α_j 的符号只依赖于数量积 $A_j^T B_p$ 的符号。注意到 $A_j = \nabla g_j$, 所以

$$\frac{dg_j}{d\alpha} = A_j^T B_p$$

由此可见 α_j 的符号取决于函数 g_j 对于 α 的变化率。如果这个变化率是负的, 则当 α 逐渐增加时, g_j 值减小, 迭代点就越来越远离该约束曲面而深入其可行域内, 所以在确定 α 时我们就用不到考虑 $\alpha_j < 0$ 的情况。而对于所有 $\alpha_j > 0$ 的 α_j 值, 又要取其中的最小值, 以保证沿梯度投影方向移动不超出可行域, 即对应于第一个不起作用约束曲面的步长。根据这些想法, 我们按下式来确定步长

$$\alpha_M = \min_{\substack{\alpha_j > 0 \\ j \in J}} [\alpha_j] \quad (6-9)$$

使用上式, 只对那些实际上是不起作用的约束 (即 $j \in J$), 求出相应的 α_j 。所有那些在构成 x^K 点的投影矩阵时用到过的约束, 在由步长 α_M 所确定的点处也将是起作用的, 而且至少还得加上一个新的起作用约束, 即 (6-9) 式中 α_M 所对应的约束。

正象一般的可行方向法那样, 如果沿着梯度投影 $d^K = -P \nabla f(x)$ 方向, 目标函数 $f(\alpha)$ 在 $\alpha = 0$ 与 $\alpha = \alpha_M$ 之间有一个最小值点, 则 α_M 就不是“最优”的步长。因此如果目标函数是非线性的, 我们就要计算出在 α_M 点处的数量积 $(d^K)^T \nabla f(\alpha_M)$, 如果它是正的, 那就需要用内插法或其它一维搜索方法求出那个介于 0 与 α_M 间的 α^* 来。然后按此 α^* 进行走步, 这样的移动既保证不超出可行域, 又能使目标函数值最小, 即

$$\begin{aligned} x^{K+1} &= x^K + \alpha^* d^K \\ f(x^{K+1}) &= \min_{0 \leq \alpha \leq \alpha_M} f(x^K + \alpha d^K) \end{aligned}$$

这时每一次移动不一定要碰到一个新的约束曲面, 但是仍旧不可能离开任何一个已经碰到的约束曲面。

三、约束的删除

矩阵 A 是由起作用的约束法向量所组成, 根据库恩——塔克条件, 当梯度投影为零向量时, 乘子 μ 为非负。所以当梯度投影 $B_p = -P\nabla f$ 为零向量形成投影矩阵时, 就要删除掉对应于乘子 μ_j 为负值的约束。这可以按式

$$\mu = (A^T A)^{-1} A^T B$$

计算 μ 值, 从中挑出取负值的 μ_j 。如果在某点处有几个 μ_j 都是负的, 我们每次还是只能删除掉一个约束。一个合理的删除规则是首先将负得最多的 μ_j 所对应的约束删除掉, 即求 t 使

$$\mu_t = \min[\mu_j]$$

在矩阵 A 中除掉 A_t 列向量, 此时起作用约束下标集合将是 $J(x^K) - \{t\}$ 。按此集合再计算新的投影矩阵 P 及梯度投影 $B_p = -P\nabla f(x^K)$ 。沿 B_p 进行走步, 当 B_p 变为零向量时再计算和检验 μ 值, 并删除其负值所对应的约束。由上面的分析可看到, 在每次迭代中都要重新计算投影矩阵 P 以及系数向量 μ , 其中都包括有矩阵求逆, 这要花费较多的计算时间。为避免其中的矩阵求逆计算, 已有现成的递推公式, 详见参考文献[1]。

四、计算步骤

现在我们把梯度投影法的迭代步骤总结如下:

1. 给定初始可行点 x^1 及收敛精度 $\epsilon > 0$, 令 $K=1$ 。
 2. 确定等式约束和起作用不等式约束集合 $J(x^K)$ 。
 3. 若 $J(x^K) = \phi$ (ϕ 表示空集), 则令 $d^K = -\nabla f(x^K)$, 否则令 A 为 m 个起作用约束法向量 A_j 所组成的 $n \times m$ 矩阵, 由 $P = I - A(A^T A)^{-1} A^T$ 建立投影矩阵 P , 取梯度投影方向为搜寻方向, 即 $d^K = -P\nabla f(x^K)$ 。
 4. 若 $d^K = 0$ (0 表示零向量), 则沿 d^K 方向进行一维搜索 $\bar{x} = x^K + a_M d^K$, 步长 a_M 由所有不起作用约束 $a_j = -\frac{f_j(x^K)}{A_j^T d^K}$ 的最小正值来确定, 即 $a_M = \min_{a_j > 0} [a_j]$ 。
 5. 若 $(d^K)^T \nabla f(\bar{x}) > 0$, 则在 $0 \sim a_M$ 进行一维搜索求 a^* , 使 $f(x^K + a^* d^K) \rightarrow \min$, 并依此步长进行走步 $x^{K+1} \leftarrow x^K + a^* d^K$ 。否则 $x^{K+1} \leftarrow \bar{x}$, 转 7。
 6. 当 $d^K = 0$, 若 $J(x^K) = \phi$, 则 x^K 为最优点, 否则计算 $\mu = (A^T A)^{-1} A^T B$, 若对应所有起作用不等式约束的 $\mu \geq 0$ 则 x^K 为最优点。若有某些 μ_t 取负值, 则求 t 使 $\mu_t = \min[\mu_j]$, 在 $J(x^K)$ 集合中去掉相应下标 t , 即 $J(x^K) \leftarrow J(x^K) - \{t\}$, 在矩阵 A 中删除 A_t 列向量, 返回到 3。
 7. 若 $\|x^{K+1} - x^K\| < \epsilon$, 则 x^{K+1} 为最优点, 否则令 $K \leftarrow K+1$ 转 2。
- 程序框图如图 (6-3) 所示。

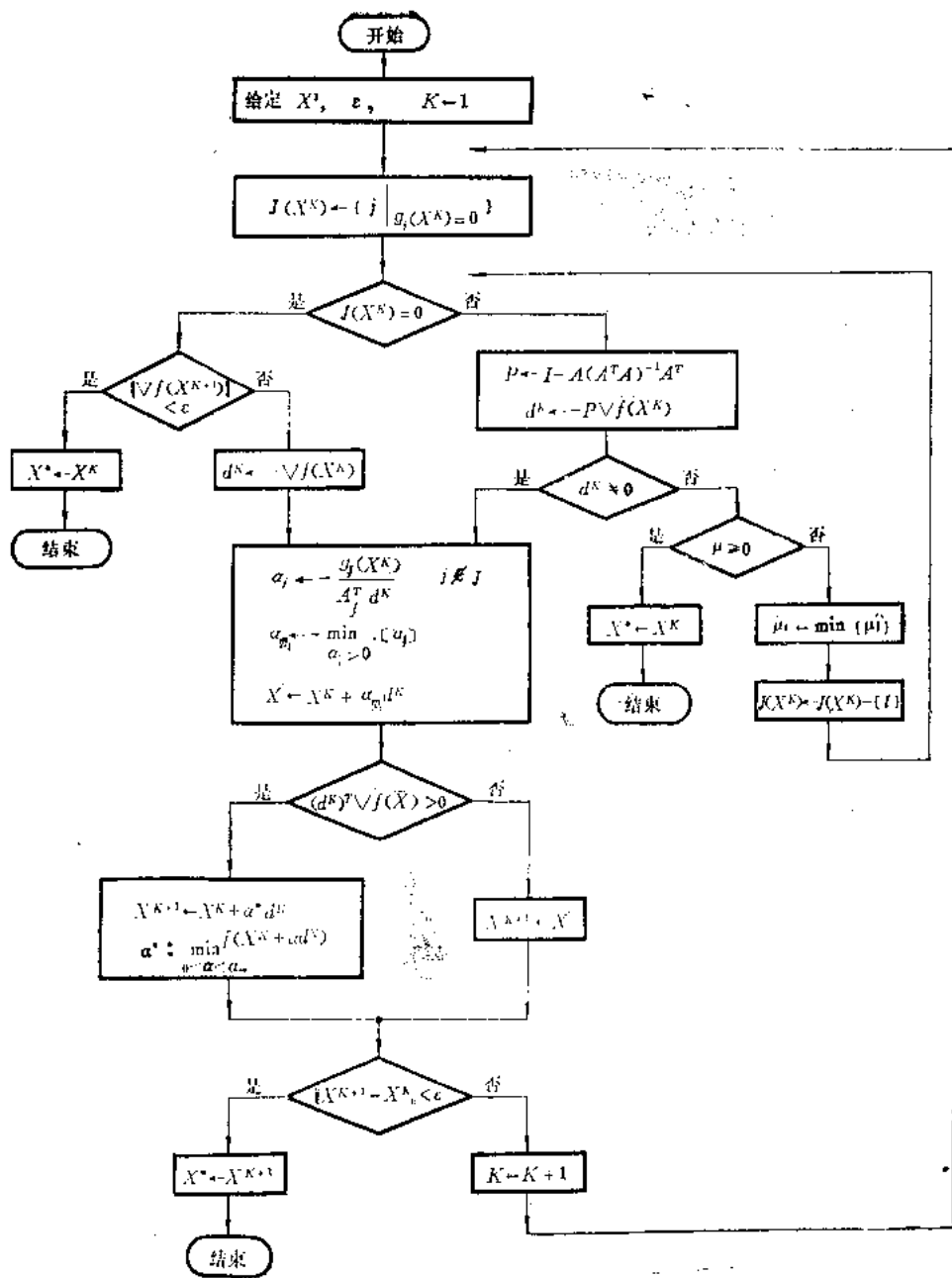


图 6-3

五、举例

用梯度投影法求解下面的优化问题。

$$f(x) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2$$

$$\text{s.t.} \quad x_1 + x_2 \leq 2$$

$$x_1 + 5x_2 \leq 5$$

$$-x_1 \leq 0$$

$$-x_2 \leq 0$$

设

$$\begin{aligned}g_1(x) &= x_1 + x_2 - 2 \leq 0 \\g_2(x) &= x_1 + 5x_2 - 5 \leq 0 \\g_3(x) &= -x_1 \leq 0 \\g_4(x) &= -x_2 \leq 0\end{aligned}$$

因为本题中不含等式约束，所以在矩阵 \mathbf{A} 中不包含和乘子 λ 相对应的项，只含和乘子 μ 相对应的项。

1. 取初始可行点 $x^1 = (0 \ 0)^T$, $\varepsilon = 10^{-3}$

$$\begin{aligned}\text{则 } \nabla f(x^1) &= (4x_1 - 2x_2 - 4 \quad 4x_2 - 2x_1 - 6)^T \Big|_{\substack{x_1=0 \\ x_2=0}} \\ &= (-4 \quad -6)^T\end{aligned}$$

在 $x^1(0 \ 0)^T$ 点处满足 $-x_1=0$ 和 $-x_2=0$ 的等式条件， g_3, g_4 是起作用约束，所以

$$\mathbf{A} = [\mathbf{A}_3 \quad \mathbf{A}_4] = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

因为
故有

$$\mathbf{A} = \mathbf{A}^T, \quad \mathbf{A}^T \mathbf{A} = \mathbf{I}$$

$$\mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T = \mathbf{I}$$

$$\mathbf{P} = \mathbf{I} - \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$d^1 = -\mathbf{P} \nabla f(x^1) = (0 \ 0)^T = 0$$

根据库恩——塔克条件判断 x^1 是否是极值点，需要计算乘子 μ 是否非负。

$$\begin{aligned}\mu &= -(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \nabla f(x^1) \\ &= -\mathbf{A} \nabla f(x^1) = -\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} -4 \\ -6 \end{pmatrix} = \begin{pmatrix} -4 \\ -6 \end{pmatrix}\end{aligned}$$

不满足乘子非负要求，故 x^1 不是极值点。 $\mu_3 = -4$, $\mu_4 = -6$ ，取其中最小者 $\mu_4 = -6$ ，删除 μ_4 所对应的约束在矩阵 \mathbf{A} 中的列向量 \mathbf{A}_4 ，得

$$\mathbf{A} = [\mathbf{A}_3] = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

$$\mathbf{A}^T = [-1 \ 0]$$

$$\mathbf{A}^T \mathbf{A} = [-1 \ 0] \begin{pmatrix} -1 \\ 0 \end{pmatrix} = 1$$

$$\mathbf{P} = \mathbf{I} - \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$$

$$= \mathbf{I} - \mathbf{A} \mathbf{A}^T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} -1 \\ 0 \end{pmatrix} [-1 \ 0] = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$d^1 = -P \nabla f(x^1) = - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -4 \\ -6 \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \end{pmatrix} \neq 0$$

沿 d^1 方向进行走步, 分别与约束曲面 g_1, g_2 相遇, 根据 (6-8) 式

$$a_j = - \frac{g_j(x^1)}{A_j^T d^1}$$

计算相应步长 a_1, a_2 , 因为

$$A_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, g_1(x^1) = -2, \quad A_2 = \begin{pmatrix} 1 \\ 5 \end{pmatrix}, g_2(x^1) = -5$$

故有

$$a_1 = - \frac{g_1(x^1)}{A_1^T d^1} = - \frac{-2}{[1 \ 1] \begin{pmatrix} 0 \\ 6 \end{pmatrix}} = \frac{2}{6} = \frac{1}{3}$$

$$a_2 = - \frac{g_2(x^1)}{A_2^T d^1} = - \frac{-5}{[1 \ 5] \begin{pmatrix} 0 \\ 6 \end{pmatrix}} = \frac{5}{30} = \frac{1}{6}$$

而

$$A_3 d^1 = [-1 \ 0] \begin{pmatrix} 0 \\ 6 \end{pmatrix} = 0$$

说明 d^1 处于 $g_3=0$ 约束面上。

$$a_M = \min_{a_j > 0} [a_j] = \min [a_1, a_2] = a_2 = \frac{1}{6}$$

说明从 x^1 沿 d^1 方向走步, 首先碰到 $g_2=0$ 约束面, 到达 \bar{x} 点

$$\bar{x} = x^1 + a_M d^1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \frac{1}{6} \begin{pmatrix} 0 \\ 6 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

为了判断从 x^1 到 \bar{x} 是否存在目标函数的极小点, 需计算 $(d^1)^T \nabla f(\bar{x})$ 值, 如果此值为正说明 $0 \sim a_M$ 之间有使目标函数值为极小的步长 a^* 。

$$\nabla f(\bar{x}) = (4x_1 - 2x_2 - 4 \quad 4x_2 - 2x_1 - 6)^T \Big|_{\substack{x_1=0 \\ x_2=1}} = -(-6 \ -2)^T$$

$$(d^1)^T \nabla f(\bar{x}) = [0 \ 6] \begin{pmatrix} -6 \\ -2 \end{pmatrix} = -12 < 0$$

说明 $0 \sim a_M$ 之间不存在使 $f(x)$ 极小的 a^* , a_M 为所求步长, \bar{x} 为走步后终点, 有

$$x^2 = \bar{x} = (0 \ 1)^T$$

进行迭代点收敛判断

$$\|x^2 - x^1\| = \frac{1}{6} \|d^1\| = 1 > 10^{-3}$$

需继续进行迭代。

2. 在 $x^2 = (0 \ 1)^T$ 处 $g_2(x) = x_1 + 5x_2 - 5 = 0$ $g_3(x) = -x_1 = 0$ 为起作用约束, 所以矩阵

$$A = [A_2 \ A_3] = \begin{bmatrix} 1 & -1 \\ 5 & 0 \end{bmatrix}$$

$$A^T = \begin{bmatrix} 1 & 5 \\ -1 & 0 \end{bmatrix}$$

$$A^T A = \begin{bmatrix} 1 & 5 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 5 & 0 \end{bmatrix} = \begin{bmatrix} 26 & -1 \\ -1 & 1 \end{bmatrix}$$

$$(A^T A)^{-1} = \frac{1}{25} \begin{bmatrix} 1 & 1 \\ 1 & 26 \end{bmatrix}$$

$$P = I - A(A^T A)^{-1}A^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{1}{25} \begin{bmatrix} 1 & -1 \\ 5 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 26 \end{bmatrix} \begin{bmatrix} 1 & 5 \\ -1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\nabla f(x^2) = (-6 \ -2)^T$$

$$d^2 = -P\nabla f(x^2) = (0 \ 0)^T = 0$$

欲判断 x^2 是否为极值点, 根据库恩——塔克条件需计算相应的乘子 μ

$$\mu = -(A^T A)^{-1}A^T \nabla f(x^2)$$

$$= -\frac{1}{25} \begin{bmatrix} 1 & 1 \\ 1 & 26 \end{bmatrix} \begin{bmatrix} 1 & 5 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -6 \\ -2 \end{bmatrix} = \frac{1}{5} \begin{bmatrix} 2 \\ -28 \end{bmatrix}$$

$\mu_2 = \frac{2}{5}$, $\mu_3 = -\frac{28}{5} < 0$ 不满足乘子非负要求, 故 x^2 不是极值点。从矩阵 A 中删除 μ_3 所对应的列向量 A_3 , 则得

$$A = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

$$A^T = [1 \ 5]$$

$$A^T A = [1 \ 5] \begin{bmatrix} 1 \\ 5 \end{bmatrix} = 26$$

$$(A^T A)^{-1} = \frac{1}{26}$$

$$P = I - A(A^T A)^{-1}A^T$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{1}{26} \begin{bmatrix} 1 \\ 5 \end{bmatrix} [1 \ 5] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{1}{26} \begin{bmatrix} 1 & 5 \\ 5 & 25 \end{bmatrix}$$

$$= \frac{1}{26} \begin{bmatrix} 25 & -5 \\ -5 & 1 \end{bmatrix}$$

$$\begin{aligned} \mathbf{d}^2 &= -\mathbf{P}\nabla f(\mathbf{x}^2) \\ &= -\frac{1}{26} \begin{bmatrix} 25 & -5 \\ -5 & 1 \end{bmatrix} \begin{bmatrix} -6 \\ -2 \end{bmatrix} = \begin{bmatrix} \frac{70}{13} \\ -\frac{14}{13} \end{bmatrix} \end{aligned}$$

因为 \mathbf{d}^2 的模无关紧要，故取 $\mathbf{d}^2 = (5 \ -1)^T$

从 $\mathbf{x}^2 = (0 \ 1)^T$ 沿 $\mathbf{d}^2 = (5 \ -1)^T$ 方向进行走步，分别与不起作用约束 g_1, g_4 相遇，相应的步长为

$$\alpha_1 = -\frac{g_1(\mathbf{x}^2)}{\mathbf{A}_1^T \mathbf{d}^2} = -\frac{-1}{[1 \ 1] \begin{bmatrix} 5 \\ -1 \end{bmatrix}} = \frac{1}{4}$$

$$\alpha_4 = -\frac{g_4(\mathbf{x}^2)}{\mathbf{A}_4^T \mathbf{d}^2} = -\frac{-1}{[0 \ -1] \begin{bmatrix} 5 \\ -1 \end{bmatrix}} = 1$$

$$\alpha_M = \min(\alpha_1, \alpha_4) = \alpha_1 = \frac{1}{4}$$

可见走步首先与约束 g_1 相遇，到达 $\bar{\mathbf{x}}$ 点，

$$\bar{\mathbf{x}} = \mathbf{x}^2 + \alpha_M \mathbf{d}^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \frac{1}{4} \begin{bmatrix} 5 \\ -1 \end{bmatrix} = \begin{bmatrix} \frac{5}{4} \\ -\frac{3}{4} \end{bmatrix}$$

根据 $(\mathbf{d}^2)^T \nabla f(\bar{\mathbf{x}})$ 的值判断 $0 \sim \alpha_M$ 之间是否有极值点 α^* 。

$$(\mathbf{d}^2)^T \nabla f(\bar{\mathbf{x}}) = [5 \ -1] \begin{bmatrix} \frac{5}{2} \\ -\frac{23}{2} \end{bmatrix} = 24 > 0$$

说明 $0 \sim \alpha_M$ 之间存在 α^*

从 $\mathbf{x}^2 = (0 \ 1)^T$ 出发沿 $\mathbf{d}^2 = (5 \ -1)^T$ 方向进行一维搜索

$$\mathbf{x}^3 = \mathbf{x}^2 + \alpha \mathbf{d}^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \alpha \begin{bmatrix} 5 \\ -1 \end{bmatrix} = \begin{bmatrix} 5\alpha \\ 1-\alpha \end{bmatrix}$$

得一维优化问题

$$f(\alpha) = 62\alpha^2 - 28\alpha - 4 \rightarrow \min$$

$$\text{s.t.} \quad 0 \leq \alpha \leq \frac{1}{4}$$

解得

$$\alpha^* = \frac{7}{31}$$

$$x^3 = x^2 + \alpha^* d^2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \frac{7}{31} \begin{pmatrix} 5 \\ -1 \end{pmatrix} = \begin{pmatrix} \frac{35}{31} \\ \frac{24}{31} \end{pmatrix}$$

检查收敛要求

$$\|x^3 - x^2\| = \alpha^* \|d^2\| = \frac{7}{31} \sqrt{26} > 10^{-3}$$

不满足收敛条件，需继续迭代。

3. 在 $x^3 = \left[\frac{35}{31} \quad \frac{24}{31} \right]^T$ 点起作用约束只有 ε_2 ，所以矩阵

$$A = \begin{pmatrix} 1 \\ 5 \end{pmatrix}$$

这和前面第二次迭代时，删去 μ_3 所对应的列向量 A_3 后，所得的 A 矩阵相同。故有

$$P = \frac{1}{26} \begin{pmatrix} 25 & -5 \\ -5 & 1 \end{pmatrix}$$

$$\nabla f(x^3) = \left[-\frac{32}{31} \quad -\frac{160}{31} \right]^T$$

则
$$d^3 = -P \nabla f(x^3) = -\frac{1}{26} \begin{pmatrix} 25 & -5 \\ -5 & 1 \end{pmatrix} \begin{pmatrix} -\frac{32}{31} \\ -\frac{160}{31} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = 0$$

梯度投影为零向量，根据库恩——塔克条件，若乘子 μ 为非负，则 x^3 为极值点。

$$\mu = -(A^T A)^{-1} A^T \nabla f(x^3) = -\frac{1}{26} [1 \quad 5] \begin{pmatrix} -\frac{32}{31} \\ -\frac{160}{31} \end{pmatrix} = \frac{32}{31} > 0$$

满足库恩——塔克条件的全部要求，故 $x^3 = \left(\frac{35}{31} \quad \frac{24}{31} \right)^T$ 为最优解。

此优化问题的可行域及各迭代点如图 (6-4) 所示。

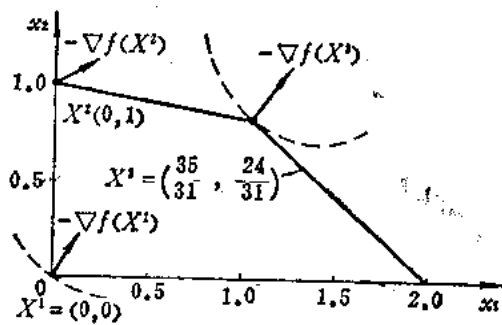


图 6-4

六、放宽约束方法

下面再简单介绍采用放宽约束的办法来处理不等式约束时的梯度投影法的具体做法。这种方法也称具有约束补偿的梯度法。

这时，若取 $\varepsilon > 0$ 为一个小参数，则在

$$-\varepsilon \leq g_j(x) \leq \varepsilon$$

范围内变化的设计点 x 仍认为是可行点，即认为它并没有破坏约束。这样的约束函数称为 ε -约束函数或 ε -有效约束。这种处理办法有两种做法。一是取 $-\varepsilon \leq g_j(x) \leq \varepsilon$ 为允许约束，当然此时的 $g_j(x) \leq \varepsilon$ 是稍有些违反的约束；二是要求 x 必须是可行点，即要求 $-\varepsilon \leq g_j(x) \leq 0$ 。

针对上述放宽约束的情况，若设计点为 x^0 ，则 ε -约束函数就可能有三种情况：

- 1) 临界约束，此时 $g_j(x^0) = 0$ ；
- 2) 接近临界的约束，此时 $-\varepsilon \leq g_j(x^0) < 0$ ；
- 3) 稍被破坏的约束，此时 $0 < g_j(x^0) \leq \varepsilon$ 。

这样一来，对可行方向 d 的要求将从式 (6-1) 的 $\nabla f^T d < 0$ 和式 (6-2) 的 $\nabla g^T d \leq 0$ 这两点扩大成三点，即：

- 1) 可行点 x 沿着它移动时，目标函数值应下降或最少不变， $\nabla f^T d < 0$ 或 $\nabla f^T d \rightarrow \min$ ；
- 2) 可行点 x 沿着它移动时，不能破坏约束条件， $\nabla g^T d \leq 0$ （在这里 $\nabla g(x)$ 中可能含有等式约束）；
- 3) 在进行迭代运算过程中，步长的选取应考虑小参数 ε 的约束，其说明如下。

若迭代公式是

$$x^{K+1} = x^K + \alpha_K d^K$$

或为了下面讨论时书写方便，用 $dx^K = \alpha_K d^K$ ，并且只用 $dx = \alpha d$ 的书写形式。则这个要求可以写成

$$dx^T W dx \leq \xi^2$$

其中的

$$W = \begin{bmatrix} \omega_1 & & & 0 \\ & \omega_2 & & \\ & & \ddots & \\ 0 & & & \omega_n \end{bmatrix}$$

是正定权方阵。

§ 3 复形法和伸缩保差法简述

复形法是对应于单形替换法的一种约束优化方法，也就是单形替换法用于无约束优化问题，而复形法则是用于约束优化问题的求解。复形法的顶点个数通常是 $k \geq n + 1$ 。它适用于具有如下类型的约束条件

$$\begin{aligned} g_j(x) &\leq 0 \\ x_{i, \max} &\leq x_i \leq x_{i, \min} \quad (i = 1, 2, \dots, n) \end{aligned}$$

第一个复形的顶点坐标可取

$$x_{ij} = x_{i \min} + r_{ij}(x_{i \max} - x_{i \min}) \quad (i = 1, 2, \dots, n) \quad (6-10)$$

其中 $0 < r_{ij} < 1$ 。

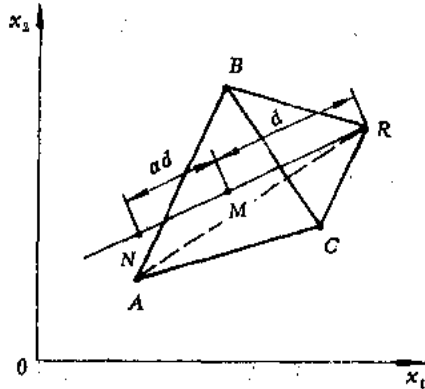


图 6-5

对于二维问题，其复形可以如图 (6-5) 所示。

图中的 M 是三角形 ACB 的重心， R 是抛弃点 (目标函数值中的最坏的点)， d 是 RM 的距离。 N 是 R 点的反射点，距离 $NM = ad$ ，即不像在单形替换法中那样，要求反射点和其它各点之间的距离相等。

求反射点 N 的坐标可用公式

$$\begin{aligned} x_{iN} &= \alpha(x_{iM} - x_{iR}) + x_{iM} \\ &= (1 + \alpha)x_{iM} - \alpha x_{iR} \end{aligned} \quad (6-11)$$

三角形重心点 M 的坐标用下式计算

$$x_{iM} = \frac{1}{k-1} \left(\sum_{j=1}^k x_{ij} - x_{iR} \right) \quad (6-12)$$

式中 k 是复形的顶点数目，应取 $k \geq n+1$ ； n 是变量个数。可取 $k = 2n$ ， $\alpha = 1.3$ 。若 n 大于 5，则 k 可以取得小些。

和单形替换法一样，复形法中也有些规则要考虑 (下面说明规则时是用图 (6-5) 作参考的)。

首先，复形法中的每一个点都应满足所有约束条件的要求。

规则 1：若出现违反约束条件，则向着已经确定顶点重心 M 移动的试验点的移动距离要减半。

规则 2：算出各顶点处的目标函数值，抛弃其中最坏的点 R 。从 R 指向 M 的方向形成反射点的移动方向。沿此方向移动距离 ad 。若新生成的反射点 N 是可行点，就计算此点处的目标函数值。

规则 3：若新生成的反射点 N 处的目标函数值在新形成的复形中是最坏的，则可取前次移动距离之半。这样得到的点记为 N' ，则它的坐标是

$$x_{iN'} = \frac{1}{2}(x_{iN} - x_{iM}) \quad (6-13)$$

规则 4：若在可行解区域内约束 $g_k \leq 0$ 总是违反，则移动距离再减半，并这样继续进行下去，一直到满足约束为止。

规则 5：若生成的反射点不满足对 x_i 为显式的约束条件，则在适合的边界以内重新取一个适当小的距离以获得可行点。

下面举一个例子说明方法。

求目标函数 $f(x_1, x_2) = (x_1 - 0.5)^2 + (x_2 - 1.0)^2$ 的极小值。受有约束条件 $g(x) = x_1^2 + 2x_2^2 - 4 \leq 0$ 。

对这个问题的约束还有

$$-2 < x_1 < 2$$

$$-\sqrt{2} \leq x_2 \leq \sqrt{2}$$

这样的约束相当于 $g(x) \leq 0$ 所围的方形区域。

取 $k=2n=4$ ，即取 4 个顶点的复形为第一个复形，并从原点开始搜索。

根据式 (6-10)，并取随机数 r_{1j} 分别为 0.1 和 0.9，可求得第二个顶点坐标为 (-1.600, 1.131)。这是因为由式 (6-10) 有

$$x_{12} = -2 + 0.1(2+2) = -2 + 0.4 = -1.600,$$

$$x_{22} = -\sqrt{2} + 0.9(\sqrt{2} + \sqrt{2}) = -\sqrt{2} + 1.8\sqrt{2} = 0.8\sqrt{2} = 1.131.$$

代入约束条件 $g(x) = x_1^2 + 2x_2^2 - 4 = 2.56 + 2.558 - 4 = 1.118$ ，不满足 $g(x) \leq 0$ 的要求。应用规则 1，点 8 取在 x_1 和 x_2 的中点。这是一个可行点，取为第二个顶点。同样，生成点 x_4 和点 x_6 ，从而形成第一个复形。

在每一个顶点计算目标函数值，最大值的点是摒弃点，即摒弃点 x_6 。用式 (6-11) 和 (6-12) 确定点 x_6 。应用合适的规则进行计算。其中几个循环的结果如表 (6-1) 所列。

伸缩保差法是由单形替换法发展而来的，它可用于求解约束优化问题，所以又称约束单形法。它的基本思想是把

$$\begin{aligned} & \min f(x) \\ & \text{s.t. } h_i(x) = 0 \quad (i=1, 2, \dots, l) \\ & \quad g_j(x) \leq 0 \quad (j=1, 2, \dots, p) \quad (l+p=m) \end{aligned}$$

的多约束优化问题化为一个如下形式的单约束优化问题

$$\begin{aligned} & \min f(x) \\ & \text{s.t. } T(x) - \phi^K \leq 0 \end{aligned} \quad (6-14)$$

式中 ϕ^K 称为第 K 步计算时的容许公差判据（或准则），它随着 K 值的增加逐渐减小，即有 $\phi^0 \geq \phi^1 \geq \dots \geq 0$ ； $T(x)$ 称为约束破坏估计量。由式 (6-14) 可知，当 K 值逐渐增加时， $T(x)$ 也应逐渐减小，并趋近于 0，即越来越趋向于符合原问题中的多约束条件。

可以取

$$\phi^K = \min \left\{ \phi^{K-1}, \frac{l+1}{r+1} \sum_{i=1}^{r+1} \|x_i^K - x^{K_{r+2}}\| \right\} \quad (6-15)$$

$$\phi^0 = 2(m+1)t \quad (6-16)$$

式中： t 是单形替换法中的步长； l 是等式约束数目； x_i^K 是单形的第 i 个顶点； $r = n - m$ ； $x^{K_{r+2}}$ 是单形的中心（当 $n = r$ 时）。

约束破坏估计量可取为

$$T(x) = \left[\sum_{i=1}^l h_i^2(x) + \sum_{j=1}^p \mu_j g_j^2(x) \right]^{\frac{1}{2}} \quad (6-17)$$

当 $T(x^K) > 0$ 时， x^K 不在可行域内；当 $T(x^K) = 0$ 时， x^K 在可行域内。而当 $0 \leq T(x^K) \leq \phi^K$ 时，则称 x^K 在近似可行域内。因此，此方法实际上是在可行域和近似可行域内搜索最优点。

表 6-1

序号	r_1	r_2	重心 M			顶点 K				抛弃点	规则	结果的图形
			x_{1M}	x_{2M}	$f_m(x)$	x_{1K}	x_{2K}	$g(x)$	$f_K(x)$			
1						0	0	-4	1.2500	1		1
2	0.1, 0.9					-1.600	1.131	1.118		2	1	1, 2
3		0	0	1.2500	-0.800	-0.566	-2.719	1.8784	1, 3			
4	0.6, 0.7					0.400	0.566	-3.199	0.1984	3		1, 3, 4
5	0.5, 0.2					0	-0.848	-2.562	3.6651	4		1, 3, 4, 5
6						-0.306	1.970			5	2	1, 3, 4, 6
7			-0.133	0.377	0.7888	-0.306	1.414	0.094		6	5	1, 3, 4, 7
8						-0.220	0.896	-2.346	0.5292	7	4	1, 3, 4, 8
9			0.060	0.487	0.4568	1.178	0.384	-2.317	0.8391	8	2	1, 9, 4, 8
10						1.042	1.415			1	2	10, 9, 4, 8
11			0.453	0.615	0.1504	1.042	1.414	1.086		10	5	11, 9, 4, 8
12						0.748	1.015	-1.380	0.0617	11	4	12, 9, 4, 8
13						-0.821	1.401	0.600		9	2	12, 13, 4, 8
14			0.309	0.826	0.0668	-0.256	1.114	-1.452	0.5845	13	4	12, 14, 4, 8
15						0.027	0.970	-2.117	0.2246	14	3	12, 15, 4, 8
16			0.392	0.850	0.0342	1.188	0.790	-1.340	0.5174	8	2	12, 15, 4, 16
17						0.790	0.820	-2.031	0.1165	16	3	12, 15, 4, 17
18						1.451	0.579	-1.224	1.0816	15	2	12, 18, 4, 17
19			0.646	0.800	0.0613	1.049	0.690	-1.947	0.3975	18	3	12, 19, 4, 17
20						0.848	0.745	-2.171	0.1861	19	3	12, 20, 4, 17
21						1.309	1.242	0.799		4	2	12, 20, 21, 17
22			0.795	0.860	0.1066	1.052	1.051	-0.684	0.3073	21	4	12, 20, 22, 17
23						0.924	0.956	-1.318	0.1817	22	3	12, 20, 23, 17
24			0.821	0.930	0.1079	0.786	1.171	-0.640	0.1110	20	2	12, 24, 23, 17
25			0.775	1.002	0.0756	0.581	1.062	-1.407	0.0104	23	2	12, 24, 25, 17
26			0.705	1.083	0.0489	0.595	1.425	0.415		17	2	12, 24, 25, 26
27						0.650	1.254	-0.432	0.0870	26	4	12, 24, 25, 27

在实际问题中，目标函数往往用间断的数值给出，或者是用连续的和间断的数值混合起来的形式给出。例如，需要的某种容器，它的直径 D 是以 6 厘米为单位提供的。若最小容器 ($D < 36$ 厘米) 的隔层间距只有 12, 14, 15 和 16 厘米的，而最大容器 ($D > 30$ 厘米) 的隔层间距只有 12, 14, 16 和 18 厘米的。如果用容器的制造价格

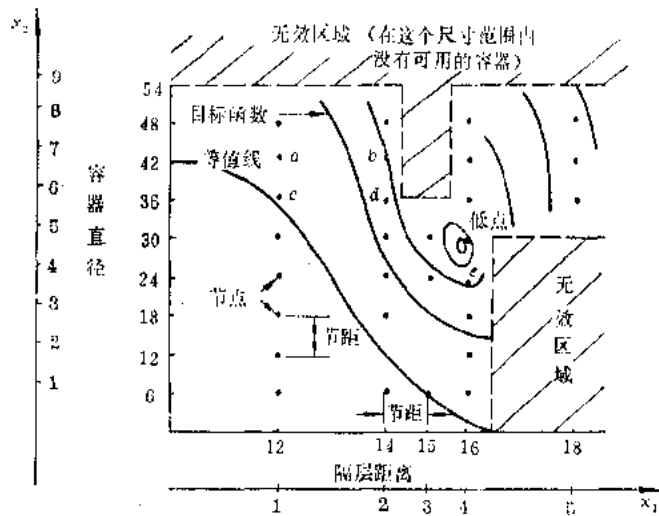


图 5-6

为目标函数，并要求价格最低，则这样的优化问题的可行解区域如图(6-6)所示。

这样的优化问题可以采用单变量搜索方法，但须以节距为步长。例如，以点 f 为搜索起点，先沿 x_2 轴搜索到达点 a ，再沿 x_1 轴搜索到达点 b 。虽然点 b 比点 a 好，但只是局部优化解。也可以先沿 x_1 轴再沿 x_2 轴搜索，结果得点 e 为优化点。它是全局的最优解。

由于步长须等于节距，所以不能进一步搜索到真正的极值点 o 。

这类问题可以采用改进的复形法来处理。其方法和规则与前面介绍的复形法的相同，只是要求复形的 $k=2n$ 个可行顶点都要选在可行区域内的节点上。

若函数是以间断值出现的线性函数，或者可以在局部区域内使函数线性化，则可以采用整型数的线性规划方法。它是以线性规划为基础的，不过要求最后的解中的变量值是整数，因此这种方法有时称为整规法。

§ 4 关于多目标函数优化问题的解法

机械结构优化设计，在许多情况下是多目标的问题。例如，滑动轴承、齿轮传动和连杆机构等就有按多目标进行设计的。

近年来，多目标函数最优化问题的解法发展很快，也出现了一些新的求解方法。但是还缺乏系统的研究，也没有形成一套完整的方法。下面仅就某些方法予以简要说明。

多目标函数优化问题可以表示为：

$$\text{求 } \mathbf{x} = (x_1, x_2, \dots, x_n)^T$$

使 $\mathbf{F}(\mathbf{x}) \rightarrow \min$

而 $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})]$

定义两个概念。

如果第 K 个目标函数的最优解是 $x^K (k=1, 2, \dots, m)$, 相应的极小值是 $\min f_K(x)$ 。

当 m 个多目标函数的最优解都相同, 且都是 x^K 时, 就称此解为最优解。

但 m 个多目标函数的最优解一般都是不相同的, 即一般地说 $x^{K-1} \neq x^K$ 。这时称它们都是非劣解。

根据这个概念, 则多目标函数优化问题的解法可大致分为:

1. 把多目标问题转化成一个或一系列单目标问题, 而把这些单目标问题求出的解, 做为多目标函数优化问题的一个解。显然, 这个解一般只能是原问题的一个或一部分的非劣解。至于其它的非劣解不得而知。

2. 直接求非劣解, 然后从中选择较好的解。

现在对几个求解方法简要地予以说明。

一、主要目标法

这种方法是把优化设计问题的多个目标函数予以分析, 然后从中选择一个目标作为主要目标, 其余的目标转化成为约束条件。

设 m 个目标是 $f_1(x), f_2(x), \dots, f_m(x)$ 。

若选其中的 $f_1(x)$ 为主要目标, 其它目标 $f_j(x)$ 具有的上、下限值为 $f_j(x)_{max}$ 和 $f_j(x)_{min}$ 。则问题可以提为:

$$\begin{aligned} & \min f_1(x) \\ & \text{s.t. } f_j(x)_{min} \leq f_j(x) \leq f_j(x)_{max} \quad (j=2, 3, \dots, m) \end{aligned}$$

二、线性加权方法

把 m 个目标 $f_1(x), f_2(x), \dots, f_m(x)$ 写成

$$F(x) = \sum_{j=1}^m \omega_j f_j(x)$$

ω_j 是权因子, 它标志着各目标的重要程度。不过其值不易选定。

另外, 这时要把各目标函数先化成无量纲的项。无量纲化的简单办法是假定各目标函数的一个标准值 $f_{jH}(x)$, 然后取

$$\bar{f}_j(x) = \frac{f_j(x)}{f_{jH}(x)}$$

则

$$F(x) = \sum_{j=1}^m \omega_j \bar{f}_j(x)$$

权因子 ω_j 可以是常数, 即在整个计算过程中取成一个定值; 也可以取为设计变量 x 的函数, 即取为 $\omega_j(x)$ 。

三、理想点法

先对 m 个目标函数求出它们各自的最优解 x^* 和最优值 f^*

$$f_j^* = \min f_j(x) \quad (j=1, 2, \dots, m)$$

然后取它们的相对误差

$$\frac{f_j^* - f_j(x)}{f_j^*}$$

组成相对误差平方和形式的目标函数

$$\varphi(x) = \sum_{i=1}^m \left(\frac{f_i^* - f_i(x)}{f_i} \right)^2$$

于是

$$\min_x \varphi(x)$$

求得的这组 x 就是最优解。

四、乘除法

若 m 个目标 $f_1(x), f_2(x), \dots, f_m(x)$ 中有 $f_1(x), f_2(x), \dots, f_K(x)$ 等 K 个要求最小, 其余 $m-K$ 个 $f_{K+1}(x), \dots, f_m(x)$ 个要求最大。则可形成下面的新的目标函数

$$\phi(x) = \frac{f_1(x) \cdot f_2(x) \cdots f_K(x)}{f_{K+1}(x) \cdots f_m(x)}$$

$$\min_x \phi(x)$$

所得的解是最优解。

五、功效系数法

设 m 个目标 $f_1(x), f_2(x), \dots, f_m(x)$, 它们分别具有功效系数 $d_i (0 \leq d_i \leq 1)$ 。形成一个新的目标函数

$$D = \sqrt[m]{d_1 \cdot d_2 \cdots d_m} \quad (0 \leq D \leq 1)$$

功效系数为 0 是坏方案, 为 1 是最优方案。因此 D 值越大越好。所以问题是

$$\max_{d_i} D$$

功效系数 d_i 的取值可以有如图 (6-7) 所示的几种类型。

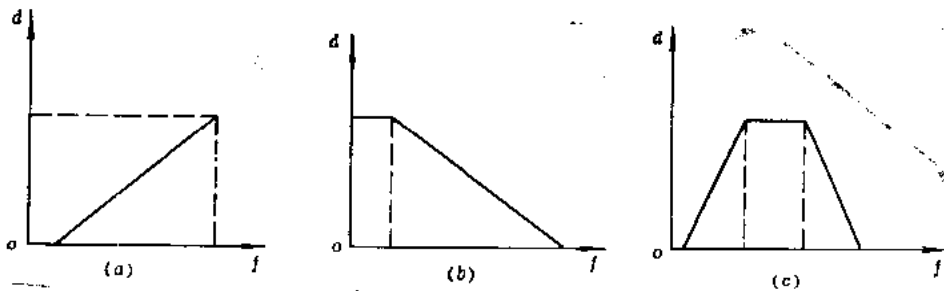


图 6-7

根据不同类型的目标函数选取不同类型的功效系数。

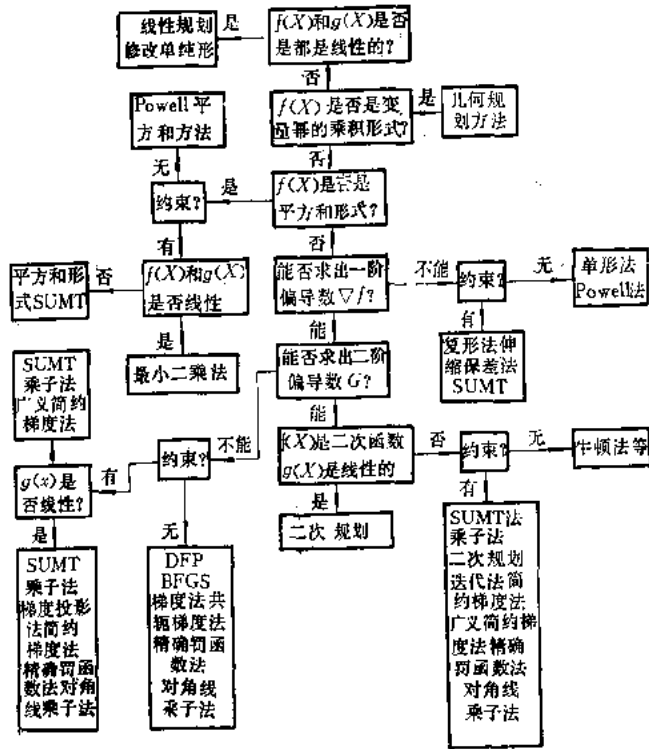
从 $D = \sqrt[m]{d_1 \cdot d_2 \cdots d_m}$ 可以看出, 只要有一个目标的功效系数为 0, 即该目标的指标不能接受时, 整个目标也就为 0, 成为最坏方案。

另外, 这种方法中, 各目标函数的量纲不同并不影响对问题的求解。

小结 —— 方法的选择

选择优化方法要考虑：函数是线性还是非线性、是否有一阶或二阶偏导数、有无约束条件以及变量数目等因素。在第一章 §4 中已经指出，在实际问题中，主要是采用数值解法，很少采用解析解法。

综合各章所讲的内容，可以把方法的选择以框图形式表示如下：



第七章 优化准则法

在第一章里曾经指出,随着结构优化设计的发展,形成了一类被称作“优化准则法”的优化设计方法。虽然,最初它只是用在以应力为约束的桁架重量最轻设计中,如“同步失效准则”和“满应力设计”中,而且所获得的优化设计方案并不一定是重量最轻的设计方案。但是,由于它对设计变量的修改较大,因而具有收敛速度快,迭代次数少,且与结构大小及复杂程度无关等特点。所以逐步被推广应用于解决具有位移(变形)、固有频率(或频率禁区)约束及多工况、多约束的结构优化设计问题。特别是,由于有限元法在结构分析中的广泛应用,如可以用有限元法(以板、梁等作离散单元)计算结构的性能(如位移或变形、固有频率、振型等响应量)及其导数,使得优化准则法的应用范围扩大到可以对弹性连续体结构进行优化设计。虽然如此,但优化准则法目前仍处于发展阶段,还有许多工作要做,包括其理论基础等。下面我们先从“满应力设计”开始,来说明优化准则法及其与最优设计之间的联系。

§ 1 满应力设计和能量准则

满应力设计是一种准则设计方法,它主要是针对桁架结构的最轻设计发展起来的。它不同于用数学原理求极值的数学规划法,而是直接从结构力学的原理出发,以满应力为准则。即在既定的桁架结构形式下,使其各组成杆件的材料在各种载荷条件下,尽可能充分发挥其潜力,达到使桁架结构重量最轻的目的。

所谓满应力就是组成桁架的各杆件中的应力达到了材料的许用应力,因而材料能得到充分利用。如果结构承受有多个独立载荷的作用(即多工况),那么满应力设计的任务就是要使桁架结构中每一个杆件至少在一种载荷条件下达到满应力。

为了说明方法,用一个由 n 根杆件组成的桁架为例。若各组成杆件的截面积是 A_i ,长度是 l_i ($i=1, 2, \dots, n$)。当然要求 $A_i \geq 0$, $l_i \geq 0$,材料的比重是 ρ_i ,那么桁架的重量是

$$f(\mathbf{A}) = \sum_{i=1}^n A_i l_i \rho_i$$

按满应力的要求,第 i 个杆件中的应力至少在一种载荷条件下应达到许用应力极限 $[\sigma_i]$,则有

$$\frac{s_i}{[\sigma_i]} = A_i$$

式中 s_i 是第 i 个杆在各种载荷下的最大轴力。

如果把上式改写成等式约束形式 $h_i(\mathbf{A}) = -A_i + \frac{s_i}{[\sigma_i]}$, 则问题的提法是: 求

$$\mathbf{A} = \{A_i\} (i=1, 2, \dots, n)$$

使
$$f(\mathbf{A}) = \sum_{i=1}^n A_i l_i \rho_i \longrightarrow \min$$

相应的满应力约束条件是

$$h_i(\mathbf{A}) = -A_i + \frac{s_i}{[\sigma_i]} = 0 \quad (i=1, 2, \dots, n)$$

$$A_i \geq 0$$

考虑到 n 个约束条件都是等式, 因而可以唯一确定 n 个设计变量 \mathbf{A} 。这 n 个设计变量 \mathbf{A} 和重量 $f(\mathbf{A})$ 的最小实际上并无直接联系, 因而所得结果并不一定是最轻重量的优化解。除非把所有约束改成不等式约束 $g_i(\mathbf{A}) \leq 0$ 的形式, 并且使 $f(\mathbf{A})$ 的最小和约束条件联系起来求解 \mathbf{A} , 才能保证结构重量为最轻。虽然如此, 但在不少情况下, 用满应力设计所得结果和优化解是接近的。

如果把每一根杆件的应力达到其许用值看作整个桁架的一种可能的破坏形式, 那么满应力设计就是同步失效设计, 这些方法就是用一个准则来代替原来的优化问题。这种寻求一个满足某种准则的设计, 暂且不管目标函数的做法是满应力设计的特点。

上述满应力设计的过程是: 先试选一组截面 \mathbf{A}^0 作为初始设计, 通过各工况的结构分析求出各杆件的最大应力 σ_i^0 (如果这个应力比该杆件允许应力高, 说明该杆件截面小了, 否则就是大了), 从而求出新的一组截面 \mathbf{A}^1 。这就完成了第一个循环。如果这个新的设计还没有达到满应力, 接着再以 \mathbf{A}^1 作为第二个循环的起点, 求出 \mathbf{A}^2 。如此迭代循环下去, 直到某一循环的起点 \mathbf{A}^K 和终点 \mathbf{A}^{K+1} 足够接近时, 便认为已收敛到满应力设计了。如果用 σ_i^K 表示第 K 次循环中得到的某杆的应力, 则上述循环可以写成下面的形式

$$A_i^{K+1} = A_i^K \frac{\sigma_i^K}{[\sigma_i]} = c_i^K A_i^K \quad (7-1)$$

式中

$$c_i^K = \frac{\sigma_i^K}{[\sigma_i]}$$

称作应力比。满应力设计程序框图如图 (7-1) 所示。

当采用应力比时, 满应力设计方法又称比例满应力设计法。这种方法实质上是假定各杆的内力是不随截面积的改变而改变的, 这对静不定桁架并不存在这种性质。

应力比这个比例因子代表各杆件截面积的裕量。当 $c_i < 1$ 时说明杆件面积大了, 还有裕量; $c_i > 1$ 时说明杆件面积小了, 需要加大。只有在 $c_i \rightarrow 1$ 的情况下, 才是满应力的设计方案。考虑到这一点, 可以采用“超松弛”的办法加速收敛。这时可把式 (7-1) 改写成

$$A_i^{K+1} = (c_i^K)^\beta A_i^K \quad (7-1a)$$

式中 $\beta > 1$ 是超松弛系数。

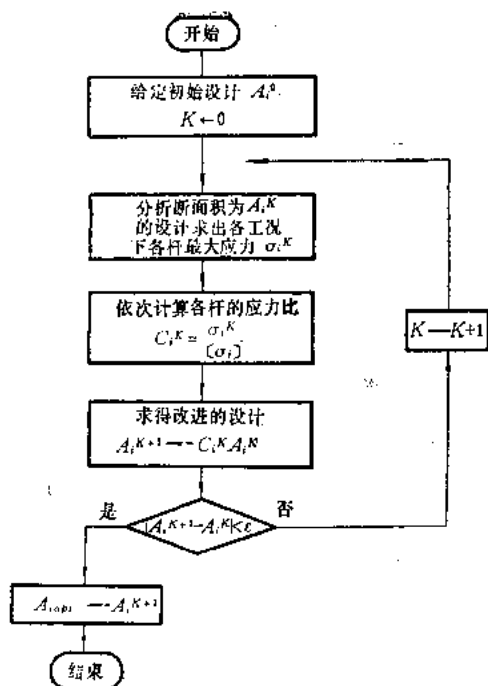


图 7-1

严格地说，齿行法是利用可行性调整步(射线步)和比较目标函数值的作法，已经不再是满应力设计法，而具有某种数学规划法的内容了。

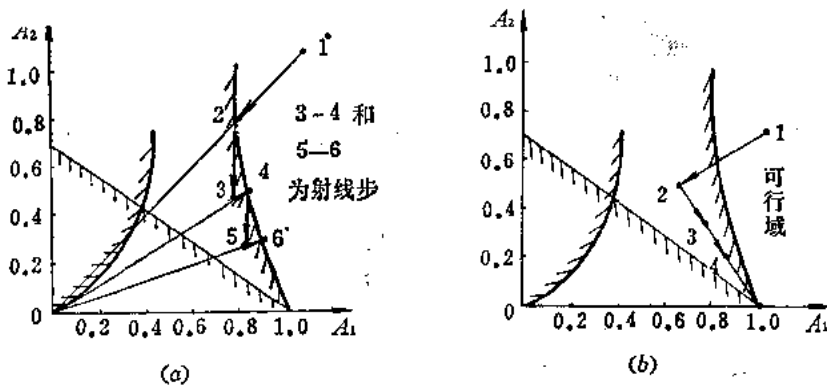


图 7-2

与满应力设计十分类似的“满应变能设计”也是一种准则设计法。它是根据能量准则进行设计的。当结构在载荷作用下发生变形，于是各部分将储存一定的应变能。结构某一部分储存应变能的多少是衡量它能分担整个结构所承受的载荷多少的标志。因此，为了最大限度地发挥材料的效用，应尽可能使材料在结构中的分布和各处应变能成比例，且具有相同的承载能力指标。

以桁架为例，设 ϵ_i 是第 i 号杆件在载荷作用下的应变。为了使结构的重量最小，其中各杆件的应变能最大值 $U_{i,max} = \frac{1}{2} E(\epsilon_i)^2 A_i l_i$ 都应等于材料的最大允许值。这就是最轻结构的应变能准则。根据这个准则进行结构的设计，选择各个杆件截面的方法，称为

前已指出，这种方法所得结果并不一定是最优结果。例如，一个由三杆组成的超静定结构经过计算，解得满应力设计时的结果是重量 $W = 2.828$ ，而其最优解是 $W^* = 2.6389$ ，即满应力设计的解不是最优解。

为了避免满应力设计收敛到非最优解，可以在满应力设计法中增加射线步。即从原点出发连接上一步比例满应力的设计点，再沿此连线方向加一步射线步，使之回到约束曲面。为了判断收敛性，在每个射线步后所得到的设计点上，计算结构的重量，并与前一次重量相比较。即走一步满应力步，接着换一步射线步。如图(7-2a)所示。图(7-2b)是不采用射线步时的情况。由于这时的整个走法类似齿形，所以称为齿行法。

“满应变能设计”或“能量准则”方法。

考虑到很难使应变能和最大允许值相等，所以一般地最轻结构的满应变能设计方法是要求各杆应变能与其最大允许值的比趋近于结构总应变能 U 和各杆件最大允许值之和 U_{max} 的比例，且等于某一个常数，即

$$\frac{U_i}{U_{imax}} \rightarrow \frac{U}{U_{max}} = \frac{1}{\alpha} \quad (\alpha \geq 1)$$

两边同乘以 αA_i ，可得

$$\alpha A_i \frac{U_i}{U_{imax}} = A_i$$

从而可得一个设计用的迭代公式

$$A_i^{K+1} = A_i^K \alpha \frac{U_i}{U_{imax}} = C_i^K A_i^K$$

其中 $C_i^K = \alpha \frac{U_i}{U_{imax}}$

满应力设计或满应变能设计的算法不是唯一的。最简单的办法是按式(7-1)的迭代公式进行循环迭代分析，逐次按比例对杆件截面尺寸 A_i 进行修改，直到达到满应力或满应变能状态为止。

对动态问题，则和某阶固有频率及振型相对应的能量有系统位能和动能。同样，也可以用上述概念进行处理。这时的设计方法也是能量准则方法。

§ 2 单个位移约束下的优化准则设计

任何一个设计方案是否是最优设计的基本检验方法，就是看它是否满足 Kuhn-Tucker 条件。对前面的桁架问题，若仅以结构的位移 z 为约束条件，考虑结构最小重量设计问题，则它可以表示为：求最佳的一组设计变量 A_i ($i=1, 2, \dots, n$)，使得桁架结构重量 $f(\mathbf{A}) = \sum_{i=1}^n \rho_i A_i l_i \rightarrow \min$ ，而且桁架的某节点指定方向位移 $z_i \leq z_i^a$ ，同时截面积应满足 $A_i \geq A_i^L$ ($i=1, 2, \dots, n$)。

对于只有一个位移约束的简单情况，可以假定它是临界约束，即有 $z = z^a$ 。则拉格朗日函数是

$$L(\mathbf{A}) = f(\mathbf{A}) + \lambda(z - z^a)$$

它的导数是

$$\frac{\partial L}{\partial A_i} = \frac{\partial f}{\partial A_i} + \lambda \frac{\partial z}{\partial A_i}$$

根据 Kuhn-Tucker 条件， A_i 为最轻设计的必要条件是（留待 §3 说明）

$$\frac{\partial f}{\partial A_i} + \lambda \frac{\partial z}{\partial A_i} = \begin{cases} \geq 0 & \text{当 } A_i = A_i^L \\ = 0 & \text{当 } A_i > A_i^L \end{cases}$$

$$\lambda \geq 0$$

(7-2)

$$\lambda(A_i l - A_i) = 0$$

由
得

$$\frac{\partial f}{\partial A_i} + \lambda \frac{\partial z}{\partial A_i} = 0$$

$$-\frac{\partial z}{\partial A_i} / \frac{\partial f}{\partial A_i} = \frac{1}{\lambda} = \text{常数} \quad (7-3)$$

式中 $-\frac{\partial z}{\partial A_i}$ 表示 A_i 有单位增值时，位移 z 的减小率，即结构的刚度收益； $\frac{\partial f}{\partial A_i}$ 表示 A_i 有单位增值时，结构重量支出。式 (7-3) 可以称为优化效率，它表明在结构的最优设计中，每个设计变量 A_i 作单位改变时，由此引起的结构整体的优化效率（即刚度收益和重量支出之比）应彼此相等，都等于一个统一的常数。显然，优化效率可以看成是一个优化准则。因此，式 (7-2) 就是所讨论的优化问题的准则方程。这样，优化准则可以概括为：在最轻（最优）结构中，设计变量都被调整到具有相等的优化效率上。

单个位移 z ，可以应用材料力学的莫尔（Mohr）公式写出其表达式。假定在外荷载作用下，桁架各杆的内力为 s_K ，又设在与位移 z 相应的单位虚荷载作用下的各杆内力为 s_K^V ，则根据莫尔公式，位移 z 为

$$z = \sum_K \frac{s_K s_K^V l_K}{E_K A_K}$$

对于静定结构，内力 s_K 以及单位虚荷载引起的内力 s_K^V 都是不随断面积 A_K 变化的，所以我们将不随 A_K 变化的量用 t_K 表示，有

$$t_K = \frac{s_K s_K^V l_K}{E_K}$$

这样位移 z 可表示成

$$z = \sum_K \frac{t_K}{A_K}$$

当计算位移 z 对设计变量 A_i 的导数时，由于 t_K 为常数，故有

$$\frac{\partial z}{\partial A_i} = \frac{\partial}{\partial A_i} \left(\sum_K \frac{t_K}{A_K} \right) = -\frac{t_i}{A_i^2}$$

式 (7-3) 中的 $\frac{\partial f}{\partial A_i}$ 可表示为

$$\frac{\partial f}{\partial A_i} = \rho_i l_i$$

则式 (7-3) 可以写成

$$-\frac{t_i}{A_i} / (\rho_i A_i l_i) = \frac{1}{\lambda} = \text{常数} \quad (7-3a)$$

而 $\frac{t_i}{A_i}$ 相当于 i 号杆件的虚应变能， $\rho_i A_i l_i$ 是 i 号杆件的重量。这样，优化准则又可以解释为：在最轻结构中，相应于每个设计变量的杆件中的比虚应变能（即单位重量中

的虚应变能)都相等,等于一个常数。由此可以理解为优化准则的物理意义。

在求解式(7-2)时,将会遇到的一个问题是,到底应取其中的

$$\frac{\partial f}{\partial A_i} + \lambda \frac{\partial z}{\partial A_i} = 0$$

还是

$$\frac{\partial f}{\partial A_i} + \lambda \frac{\partial z}{\partial A_i} \geq 0$$

因为在计算之前,我们并不知道 $A_i = A_K^L$ 还是 $A_i > A_K^L$ 。用优化准则法的术语来说,就是 A_i 是主动变量(参数)还是被动变量(参数)尚待确定。所谓主动变量是指 $A_i > A_K^L$ 的设计变量。被动变量则是指 $A_i = A_K^L$ 的设计变量。也就是需要根据杆件截面积大小的极限(又称边界约束条件)来确定主、被动变量(或称主、被动参数)。

§ 3 多位移约束下的优化准则设计

一、准则方程

对于受有多个变形位移约束和截面尺寸上、下限约束的桁架最小重量设计问题,可以表示为:求最优截面尺寸 $\mathbf{A} = \{A_i\} (i=1, 2, \dots, n)$

使
$$f(\mathbf{A}) = \sum_{i=1}^n \rho_i A_i l_i \rightarrow \min$$

但受有变形位移约束(又可称为性能约束)

$$z_j - \bar{z}_j \leq 0 \quad (j=1, 2, \dots, m) \quad (7-4)$$

和尺寸约束(又可称为侧面约束)

$$A_i^L \leq A_i \leq A_i^u \quad (i=1, 2, \dots, n) \quad (7-5)$$

不同工况下同一个杆件变形位移约束也可以理解为不同的变形位移约束。

当把不等式(7-4)和(7-5)用引入松弛变量 a_j , b_i 和 d_i 的办法变成等式约束

$$\begin{aligned} z_j - \bar{z}_j + a_j^2 &= 0 \\ A_i^L - A_i + b_i^2 &= 0 \\ A_i - A_i^u + d_i^2 &= 0 \end{aligned}$$

以后,就可以写出如下的拉格朗日函数

$$\begin{aligned} L = f(\mathbf{A}) + \sum_{j \in J} \lambda_j (z_j - \bar{z}_j + a_j^2) + \sum_{i=1}^n \mu_i (A_i^L - A_i + b_i^2) \\ + \sum_{i=1}^n \nu_i (A_i - A_i^u + d_i^2) \end{aligned} \quad (7-6)$$

其中 λ_j , μ_i 和 ν_i 是非负的常数,称作拉格朗日乘子。

使结构重量 $f(\mathbf{A}) = \sum_{i=1}^n \rho_i A_i l_i$ 为极小的必要条件由下列方程确定,由

$$\frac{\partial L}{\partial A_i} = 0 \quad (i=1, 2, \dots, n)$$

$$\frac{\partial L}{\partial a_j} = 0 \quad (j=1, 2, \dots, m)$$

$$\frac{\partial L}{\partial b_i} = 0$$

$$\frac{\partial L}{\partial d_i} = 0$$

得

$$\frac{\partial L}{\partial A_i} = \frac{\partial f}{\partial A_i} + \sum_{j \in J} \lambda_j \frac{\partial z_j}{\partial A_i} - \mu_i + \nu_i = 0 \quad (i=1, 2, \dots, n)$$

$$\frac{\partial L}{\partial a_j} = \lambda_j a_j = 0 \quad (j=1, 2, \dots, m)$$

$$\frac{\partial L}{\partial b_i} = \mu_i b_i = 0$$

$$\frac{\partial L}{\partial d_i} = \nu_i d_i = 0$$

分析 $\lambda_j a_j = 0$ 有两种可能情况:

1) 若 $a_j = 0$ 时, 则有 $z_j = \bar{z}_j$, $\lambda_j \geq 0$, 表示不等式约束 $z_j \leq \bar{z}_j$ 是起作用的约束。

2) 若 $a_j \neq 0$ 时, 则有 $z_j < \bar{z}_j$, $\lambda_j = 0$ 。表示不等式约束 $z_j \leq \bar{z}_j$ 是不起作用的约束。

这说明

$$\lambda_j \begin{cases} \geq 0 & \text{若 } z_j = \bar{z}_j \text{ (相应的不等式约束是起作用约束)} \\ = 0 & \text{若 } z_j < \bar{z}_j \text{ (相应的不等式约束是不起作用约束)} \end{cases}$$

同样, 可得

$$\mu_i \begin{cases} \geq 0 & \text{若 } A_i = A_i^L \text{ 取下限边界值 (它是起作用约束)} \\ = 0 & \text{若 } A_i > A_i^L \text{ (它是不起作用约束)} \end{cases}$$

$$\nu_i \begin{cases} \geq 0 & \text{若 } A_i = A_i^u \text{ 取上限边界值 (它是起作用约束)} \\ = 0 & \text{若 } A_i < A_i^u \text{ (它是不起作用约束)} \end{cases}$$

需要说明, 当 $A_i^L < A_i < A_i^u$ 时, $\mu_i = \nu_i = 0$, 即截面尺寸上、下限边界侧面约束都不起作用, 可以在此区域内对 A_i 进行优化。而当 $A_i = A_i^L$ 时, 即下限边界侧面约束起作用。由于此时自然满足 $A_i < A_i^u$, 所以有 $\mu_i \geq 0, \nu_i = 0$ 。当 $A_i = A_i^u$ (自然包含 $A_i > A_i^L$), 即上限边界侧面约束起作用, 有 $\mu_i = 0, \nu_i \geq 0$ 。

综合上述结果, 则最优点的必要条件按最优点所处的位置, 可表示为:

$$\frac{\partial L}{\partial A_i} = \frac{\partial f}{\partial A_i} + \sum_{j \in J} \lambda_j \frac{\partial z_j}{\partial A_i} \begin{cases} = 0, & \text{若 } A_i^L < A_i < A_i^u \\ \geq 0, & \text{若 } A_i = A_i^L \\ \leq 0, & \text{若 } A_i = A_i^u \end{cases} \quad (7-7)$$

注意这里已不包含和侧面约束相应的乘子向量 μ, ν 。

在式 (7-7) 的基础上, 可以写出以式 (7-6) 表述的具有不等式约束极值问题的 Kuhn-Tucker 条件

$$\frac{\partial f}{\partial A_i} + \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial A_i} \begin{cases} = 0 & (\text{若 } A_i^L < A_i < A_i^u) \\ \geq 0 & (\text{若 } A_i = A_i^L) \\ \leq 0 & (\text{若 } A_i = A_i^u) \end{cases} \quad (7-8)$$

$$\lambda_j (z_j - \bar{z}_j) = 0 \quad (j = 1, 2, \dots, m)$$

$$\lambda_j \geq 0$$

仿照前节的概念, 式 (7-8) 是本优化问题的准则方程。同样, 在这里也存在一个如何区分主、被动参数的问题。此外, 这里还要确定哪些不等式约束在迭代中是起作用的约束, 即哪些不等式约束包含在 J 集合之中。为此, 需要计算乘子 λ_j 的值。因此, 为了求解准则表达式

$$\frac{\partial f}{\partial A_i} + \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial A_i}$$

以确定最优解 A , 需要解决如下一些问题:

1) 求解一组包含 λ_j 的方程组, 以便确定约束是有效 (起作用) 的还是无效 (不起作用) 的;

2) 根据 $A_i \geq A_i^L$ 和 $A_i \leq A_i^u$ 区分主动和被动参数 (设计变量)。以便下次迭代时, 只把主动参数取做优化时的设计变量 (因为被动参数的值已由侧面约束边界值限定, 而不能再改变了), 利用 $\frac{\partial f}{\partial A_i} + \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial A_i} = 0$ 求解。

3) 需要计算性能约束的导数 $\frac{\partial z_j}{\partial A_i}$ 。

原则上, 可以采用试算修正方法来解决前两个问题。即事先假设好哪些约束是有效和无效, 哪些设计变量是主动和被动, 进行计算。这种方法称作试凑法 (trial-and-error method), 将在第八章结合具体问题作进一步说明。这种试算的迭代公式仍采用式 (7-1) 的形式, 即 $\mathbf{A}^{K+1} = \mathbf{C}^K \mathbf{A}^K$ 进行。当然, 这里的迭代乘子 \mathbf{C}^K 应有自己的表达形式。

从上面的推导过程中可以看出, 优化问题的准则方程实际是由所讨论的优化问题的最优解应满足 Kuhn-Tucker 条件推导出来的。这时的迭代公式是用来寻求满足 Kuhn Tucker (以后简称为 K-T) 条件的极小值点 (设计点)。

虽然上面是就变形位移约束进行推导的, 但这种优化准则法目前已推广应用于具有应力、变形位移、固有频率、振型等不同的性能约束条件下结构的优化设计问题, 并且采用有限元法进行性能约束值的计算。从而设计变量也将不限于杆件的截面尺寸 A , 还可能是所取单元的长、宽、厚等等。对于这类问题, 按上述思路同样可以导出式 (7-7) 及 (7-8) 形式的准则方程和 K-T 条件。

若用 z 泛指结构的某个性能, $f(x)$ 为目标函数, x 为设计变量, 相应的上、下限取为 x_{max} 和 x_{min} , 则这类优化问题的 K-T 条件可以写成。

$$\frac{\partial f}{\partial x_i} + \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i} \begin{cases} = 0 & (\text{若 } x_{i, \min} < x_i < x_{i, \max}) \\ \geq 0 & (\text{若 } x_i = x_{i, \min}) \\ \leq 0 & (\text{若 } x_i = x_{i, \max}) \end{cases} \quad (7-9)$$

$$\lambda_j (z_j - \bar{z}_j) = 0$$

$$\lambda_j \geq 0 \quad (j = 1, 2, \dots, m)$$

二、性能值求导

由式(7-9)可知, 利用 K-T 条件建立优化问题的准则方程, 需要计算各性能值的导数(或约束函数的梯度)。在结构优化中有时将性能导数称作敏感度。从前几章的介绍中, 我们可以看出, 在数学规划法中有许多优化方法是利用函数的梯度求解的。使用效果说明, 借助函数梯度的方法往往是很有效的。

函数梯度可以用微分法计算, 也可以用差分方法计算。下面说明当用有限元法计算结构性能值时, 它们的导数计算公式。

1. 弹性结构节点位移 y 的求导

在有限元法中, 弹性结构节点位移 y 可按下式求得

$$Ky = F \quad (7-10)$$

式中 y 是结构节点位移(静态变形)列阵, K 是结构总刚阵, F 是结构各节点的载荷列阵。

式(7-10)两边对设计变量 x_i 求导, 得

$$\frac{\partial K}{\partial x_i} y + K \frac{\partial y}{\partial x_i} = \frac{\partial F}{\partial x_i}$$

或

$$K \frac{\partial y}{\partial x_i} = \frac{\partial F}{\partial x_i} - \frac{\partial K}{\partial x_i} y$$

由于具体结构刚阵 K 是已知的, 可以计算出 $\frac{\partial K}{\partial x_i}$ 的值, 所以 $\frac{\partial K}{\partial x_i} y$ 是已知的。同样,

$\frac{\partial F}{\partial x_i}$ 也是已知的。

这样, 在用有限元法计算出结构的节点位移列阵 y 以后, 就可算出 $\frac{\partial F}{\partial x_i} - \frac{\partial K}{\partial x_i} y$ 的值,

令之为 R 。于是有

$$K \frac{\partial y}{\partial x_i} = R \quad (7-11)$$

应用线性方程组的解法, 可得性能 y 的导数值 $\frac{\partial y}{\partial x_i}$ 。

2. 结构的自由振动频率(固有频率) f 求导。

在有限元法中, 由

$$K\tilde{y} = \omega^2 M\tilde{y} \quad (7-12)$$

可以求出结构的固有频率（角频率 ω ）和相应的振型 \tilde{y} 。式中的 \mathbf{M} 是结构的总质阵。

式 (7-12) 两边对 x_i 求导，得

$$\frac{\partial \mathbf{K}}{\partial x_i} \tilde{y} + \mathbf{K} \frac{\partial \tilde{y}}{\partial x_i} = \frac{\partial(\omega^2)}{\partial x_i} \mathbf{M} \tilde{y} + \omega^2 \frac{\partial \mathbf{M}}{\partial x_i} \tilde{y} + \omega^2 \mathbf{M} \frac{\partial \tilde{y}}{\partial x_i}$$

方程两边左乘 \tilde{y}^T 并移项，得

$$\frac{\partial(\omega^2)}{\partial x_i} \tilde{y}^T \mathbf{M} \tilde{y} = \tilde{y}^T \frac{\partial \mathbf{K}}{\partial x_i} \tilde{y} - \omega^2 \tilde{y}^T \frac{\partial \mathbf{M}}{\partial x_i} \tilde{y} + \tilde{y}^T \mathbf{K} \frac{\partial \tilde{y}}{\partial x_i} - \omega^2 \tilde{y}^T \mathbf{M} \frac{\partial \tilde{y}}{\partial x_i}$$

考虑到 \mathbf{K} 和 \mathbf{M} 是对称阵，有 $\mathbf{K}^T = \mathbf{K}$ ， $\mathbf{M}^T = \mathbf{M}$ ，并且由于

$$\begin{aligned} \tilde{y}^T \mathbf{K} \frac{\partial \tilde{y}}{\partial x_i} - \omega^2 \tilde{y}^T \mathbf{M} \frac{\partial \tilde{y}}{\partial x_i} &= (\tilde{y}^T \mathbf{K} - \omega^2 \tilde{y}^T \mathbf{M}) \frac{\partial \tilde{y}}{\partial x_i} \\ &= (\mathbf{K} \tilde{y} - \omega^2 \mathbf{M} \tilde{y})^T \frac{\partial \tilde{y}}{\partial x_i} \end{aligned}$$

考虑式 (7-12)，则上式给出

$$\tilde{y}^T \mathbf{K} \frac{\partial \tilde{y}}{\partial x_i} - \omega^2 \tilde{y}^T \mathbf{M} \frac{\partial \tilde{y}}{\partial x_i} = 0$$

所以得

$$\frac{\partial(\omega^2)}{\partial x_i} \tilde{y}^T \mathbf{M} \tilde{y} = \tilde{y}^T \frac{\partial \mathbf{K}}{\partial x_i} \tilde{y} - \omega^2 \tilde{y}^T \frac{\partial \mathbf{M}}{\partial x_i} \tilde{y}$$

若按式 (7-12) 解出的振型是法式化的，即 $\tilde{y}^T \mathbf{M} \tilde{y} = 1$ ，且 $\tilde{y} = y_1$ 。则上式化为

$$\frac{\partial(\omega^2)}{\partial x_i} = y_1^T \frac{\partial \mathbf{K}}{\partial x_i} y_1 - \omega^2 y_1^T \frac{\partial \mathbf{M}}{\partial x_i} y_1$$

因为 $\omega = 2\pi f$ ，则 $\frac{\partial(\omega^2)}{\partial x_i} = 8\pi^2 f \frac{\partial f}{\partial x_i}$ ，所以上式又可写成

$$\frac{\partial f}{\partial x_i} = \frac{1}{8\pi^2 f} y_1^T \frac{\partial \mathbf{K}}{\partial x_i} y_1 - \frac{f}{2} y_1^T \frac{\partial \mathbf{M}}{\partial x_i} y_1 \quad (7-13)$$

对于具体结构，可以根据它的 \mathbf{K} ， \mathbf{M} 计算出相应的 $\frac{\partial \mathbf{K}}{\partial x_i}$ 和 $\frac{\partial \mathbf{M}}{\partial x_i}$ 。因而，由式 (7-

13) 可以算出性能 ω^2 或 f 的导数值 $\frac{\partial(\omega^2)}{\partial x_i}$ 或 $\frac{\partial f}{\partial x_i}$ 。

三、迭代乘子 C

考虑到结构性能约束函数常是隐含设计变量 x_i 的非线性方程，对式 (7-9) 的准则方程的求解可采用线性迭代的方法。这种求解是从某个初始设计变量开始，按迭代公式

$$x_i^{K+1} = c_i x_i^K \quad (7-14)$$

反复进行线性迭代，直到求出满足准则方程的设计变量。

为了确定迭代乘子 c_i ，可以引入松弛因子 α ，并且对准则方程两边乘以 $(1-\alpha)x_i$ ，得

$$(1-\alpha)x_i \left(\frac{\partial f}{\partial x_i} + \sum_{j \in J} \lambda_j \frac{\partial z_j}{\partial x_i} \right) = 0$$

展开并移项, 得

$$(1-\alpha)x_i \frac{\partial f}{\partial x_i} = -(1-\alpha) \sum_{j \in J} \lambda_j \frac{\partial z_j}{\partial x_i} x_i$$

或

$$x_i - \alpha x_i = - \frac{(1-\alpha)}{\frac{\partial f}{\partial x_i}} \sum_{j \in J} \lambda_j \frac{\partial z_j}{\partial x_i} x_i$$

则有

$$x_i = \alpha x_i - \frac{(1-\alpha)}{\frac{\partial f}{\partial x_i}} \sum_{j \in J} \lambda_j \frac{\partial z_j}{\partial x_i} x_i = \left[\alpha - \frac{(1-\alpha)}{\frac{\partial f}{\partial x_i}} \sum_{j \in J} \lambda_j \frac{\partial z_j}{\partial x_i} \right] x_i \quad (7-15)$$

令

$$c_i = \alpha - \frac{(1-\alpha)}{\frac{\partial f}{\partial x_i}} \sum_{j \in J} \lambda_j \frac{\partial z_j}{\partial x_i} \quad (7-16)$$

式(7-15)就可写成式(7-14)的形式。

式(7-16)中的 α 是松弛因子, 要求 $|\alpha| < 1$ (可以取 $\alpha = 0.8 \sim 0.9$)。

显然, 应用式(7-14)进行设计变量 x_i 的迭代计算结果应该受到侧面(即 x_i 的边界值)约束。如果计算得到的 $x_i^{K+1} = c_i^K x_i^K \geq x_{i,max}$, 应取 $x_i^{K+1} = x_{i,max}$; 而当 $x_i^{K+1} = c_i^K x_i^K \leq x_{i,min}$ 时, 则应取 $x_i^{K+1} = x_{i,min}$ 。

因此, 可以建立设计变量 x_i 的迭代方程

$$x_i^{K+1} = \begin{cases} c_i^K x_i^K & (\text{若 } x_{i,min} < c_i^K x_i^K < x_{i,max}) \\ x_{i,min} & (\text{若 } c_i^K x_i^K \leq x_{i,min}) \\ x_{i,max} & (\text{若 } c_i^K x_i^K \geq x_{i,max}) \end{cases} \quad (7-17)$$

但是, 在用式(7-16)确定迭代乘子 c_i 时, 必须先确定 $\lambda_j (j \in J)$ 的值。可以直接根据 λ_j 值应满足准则方程的条件求解 λ_j , 也可以从另一角度来求解 λ_j 。

λ 的选取, 要求使下一步设计点处的结构性能 z_j 满足其给定值 \bar{z}_j , 即设计点由第 K 步 x^K 走到第 $K+1$ 步达到 x^{K+1} 时, 结构性能由 $z_j^K \rightarrow z_j^{K+1}$, 且有 $z_j^{K+1} = \bar{z}_j$ 。

把结构性能 z_j 在点 z_j^K 的邻域内作泰勒展开, 略去二阶以上各项, 得

$$z_j^{K+1} - z_j^K = \sum_{i=1}^n \frac{\partial z_j}{\partial x_i} \Delta x_i = \sum_{i=1}^n \frac{\partial z_j}{\partial x_i} (c_i^K x_i^K - x_i^K)$$

或

$$\bar{z}_j - z_j^K = \sum_{i=1}^n \frac{\partial z_j}{\partial x_i} \Delta x_i = \sum_{i=1}^n \frac{\partial z_j}{\partial x_i} (c_i^K x_i^K - x_i^K) \quad (7-18)$$

考虑式(7-17)的三种可能情况, 并记 $x_{i,min} < c_i^K x_i^K < x_{i,max}$ 为集合 R_0 , 称为主动参数集合; $c_i^K x_i^K \leq x_{i,min}$ 为集合 R_{p1} 和 $c_i^K x_i^K \geq x_{i,max}$ 为集合 R_{p2} , 统称为被动参数集

合, 则式 (7-18) 可以写成

$$\begin{aligned} \bar{z}_j - z_j^K &= \sum_{i \in R_{p1}} \frac{\partial z_j}{\partial x_i} (x_{i \min} - x_i^K) + \sum_{i \in R_{p2}} \frac{\partial z_j}{\partial x_i} (x_{i \max} - x_i^K) \\ &+ \sum_{i \in R_a} \frac{\partial z_j}{\partial x_i} (c_i^K x_i^K - x_i^K) \end{aligned} \quad (7-19)$$

由式 (7-16) 可以写出

$$(c_i^K - 1)x_i^K = \left[\begin{array}{c} a \frac{\partial f}{\partial x_i} - (1-a) \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i} \\ \frac{\partial f}{\partial x_i} \end{array} - 1 \right] x_i^K$$

或

$$(c_i^K - 1)x_i^K = \left[\begin{array}{c} (a-1) - \frac{(1-a) \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i}}{\frac{\partial f}{\partial x_i}} \end{array} \right] x_i^K \quad (7-20)$$

所以

$$\begin{aligned} \sum_{i \in R_a} \frac{\partial z_j}{\partial x_i} (c_i^K x_i^K - x_i^K) &= \sum_{i \in R_a} \frac{\partial z_j}{\partial x_i} \left[\begin{array}{c} (a-1) - \frac{(1-a) \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i}}{\frac{\partial f}{\partial x_i}} \end{array} \right] x_i^K \\ &= \sum_{i \in R_a} \frac{\partial z_j}{\partial x_i} (a-1)x_i^K - (1-a) \sum_{i \in R_a} \frac{\frac{\partial z_j}{\partial x_i} x_i^K \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i}}{\frac{\partial f}{\partial x_i}} \end{aligned}$$

于是式 (7-19) 就成为

$$\begin{aligned} \bar{z}_j - z_j^K &- \sum_{i \in R_{p1}} \frac{\partial z_j}{\partial x_i} (x_{i \min} - x_i^K) - \sum_{i \in R_{p2}} \frac{\partial z_j}{\partial x_i} (x_{i \max} - x_i^K) \\ &+ (1-a) \sum_{i \in R_a} \frac{\partial z_j}{\partial x_i} x_i^K = -(1-a) \sum_{i \in R_a} \frac{\frac{\partial z_j}{\partial x_i} x_i^K}{\frac{\partial f}{\partial x_i}} \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i} \end{aligned} \quad (7-21)$$

令

$$\begin{aligned} b_j &= \bar{z}_j - z_j^K - \sum_{i \in R_{p1}} \frac{\partial z_j}{\partial x_i} (x_{i \min} - x_i^K) - \sum_{i \in R_{p2}} \frac{\partial z_j}{\partial x_i} (x_{i \max} - x_i^K) \\ &+ (1-a) \sum_{i \in R_a} \frac{\partial z_j}{\partial x_i} x_i^K \\ a_{jl} &= -(1-a) \sum_{i \in R_a} \frac{\frac{\partial z_j}{\partial x_i} x_i^K}{\frac{\partial f}{\partial x_i}} \frac{\partial z_l}{\partial x_i} \quad (l = 1, 2, \dots, m) \end{aligned}$$

则式 (7—21) 可给出下述的 m 个方程式

$$a_{j1}\lambda_1 + a_{j2}\lambda_2 + \dots + a_{jm}\lambda_m = b_j \quad (7-22)$$

$$(j = 1, 2, \dots, m)$$

从式 (7—22) 中可以唯一地解出 λ_j 的值。

由于方程各系数 a_{ji} 和 b_j 都随 x_i^K 变化, 因此方程组的解 λ_j 也将随 x_i^K 变化。即在迭代过程中, 设计点在设计空间不断移动, 设计点所处位置不同, 相对应的 λ_j 值也不同。即伴随着优化参数 (设计变量) 的迭代, 也存在着乘子 λ 的迭代。

观察式 (7—16) 的 c_i 表达式, 可以发现, 此时的迭代乘子不仅和约束函数 z 的导数 $\frac{\partial z}{\partial x_i}$ 有关, 而且也和目标函数 $f(x)$ 的导数 $\frac{\partial f}{\partial x_i}$ 有关。所以, 这种准则方法已经与满应力设计法不同, 它不再是和目标函数无关了。因此, 这种优化准则法就具有数学规划法的性质, 是准则思想和数学规划的结合, 故称之为优化准则法。

§ 4 优化准则法和数学规划法的一致性

虽然, 在满应力设计的一类准则设计中, 不考虑目标函数值, 因而其解不是最优解。这反映了它和数学规划法的不同, 这是它的特点。但是, 在优化准则法中, 由于准则方程是目标函数梯度和诸约束梯度的线性组合, 所以已经失去了原来的满应力类设计与目标函数无关的特点, 而具有数学规划法的性质。它实际上已经把准则法和数学规划法结合起来了。优化准则法的这个性质, 可以从对式 (7—9) 的分析中看出来。

式 (7—9) 是非线性规划问题

$$\begin{aligned} \min & f(x) \\ \text{s.t.} & z_j - \bar{z}_j \leq 0 \\ & x_{m+n} - x \leq 0 \\ & x - x_{m+n} \leq 0 \end{aligned}$$

的 $K-T$ 条件。

按照数学规划法的解法, 我们把它转换成某种无约束函数的形式, 然后采用迭代公式

$$x^{K+1} = x^K + \alpha_K d^K$$

从一点 x^K 开始, 沿方向 d^K 进行一维搜索, 确定步长因子 α_K , 得到下一步的设计点 x^{K+1} , 直到获得最优解。

在优化准则法的解法中, 从一点 x^K 开始, 采用迭代公式

$$x_i^{K+1} = c_i^K x_i^K \quad (i = 1, 2, \dots, n)$$

对它进行修改, 获得 x^{K+1} 。这种修改是靠选择、调整迭代乘子 c_i^K , 使之满足式 (7—9) 实现的。而 c_i^K 值又是由拉格朗日乘子 λ 以及约束函数偏导数和目标函数偏导数确定的。

求解同一问题的两种不同方法的等价性质, 说明两个迭代公式, 即 $x^{K+1} = x^K + \alpha_K d^K$

和 $x_i^{K+1} = c_i^K x_i^K$ 之间存在有某种相互联系的关系和相似性质。

我们可以把 $x^{K+1} = x^K + a_K d^K$ 写成增量投影形式

$$x_i^{K+1} - x_i^K = a_K d_i^K \quad (7-23)$$

另一方面, 从迭代公式 $x_i^{K+1} = c_i^K x_i^K$ 有

$$x_i^{K+1} - x_i^K = c_i^K x_i^K - x_i^K = (c_i^K - 1)x_i^K$$

如果按数学规划法, 取负梯度方向为搜索方向, 对问题的拉格朗日函数进行极小化。可以把 d 在 x_i 轴的投影 d_i 写成

$$d_i = - \left(\frac{\partial f}{\partial x_i} + \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i} \right)$$

因而有 (为了不致混淆, 把 $x_i^{K+1} - x_i^K = a_K d_i^K$ 中的步长 a_K 写成 \bar{a})

$$\bar{a} d_i = -\bar{a} \left(\frac{\partial f}{\partial x_i} + \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i} \right) = -\bar{a} \left[1 + \frac{\sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i}}{\frac{\partial f}{\partial x_i}} \right] \frac{\partial f}{\partial x_i}$$

或

$$\bar{a} d_i = - \left[\bar{a} + \frac{\bar{a} \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i}}{\frac{\partial f}{\partial x_i}} \right] \frac{\partial f}{\partial x_i}$$

如果要求 $\bar{a} d_i = (c_i - 1)x_i$, 则自式 (7-16) 和 (7-20) 有

$$\left[(a-1) - \frac{(1-a) \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i}}{\frac{\partial f}{\partial x_i}} \right] x_i = \left[- \left(\bar{a} + \frac{\bar{a} \sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i}}{\frac{\partial f}{\partial x_i}} \right) \right] \frac{\partial f}{\partial x_i}$$

或

$$\left(a-1 - \frac{\sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i}}{\frac{\partial f}{\partial x_i}} + a \frac{\sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i}}{\frac{\partial f}{\partial x_i}} \right) x_i = \left[-\bar{a} \left(1 + \frac{\sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i}}{\frac{\partial f}{\partial x_i}} \right) \right] \frac{\partial f}{\partial x_i}$$

或

$$\left[a \left(1 + \frac{\sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i}}{\frac{\partial f}{\partial x_i}} \right) - \left(1 + \frac{\sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i}}{\frac{\partial f}{\partial x_i}} \right) \right] x_i = -\bar{a} \left(1 + \frac{\sum_{j \in I} \lambda_j \frac{\partial z_j}{\partial x_i}}{\frac{\partial f}{\partial x_i}} \right) \frac{\partial f}{\partial x_i}$$

上式给出

$$(a-1)x_i = -\bar{a} \frac{\partial f}{\partial x_i}$$

或

$$\bar{\alpha} = -\frac{(a-1)x_i}{\frac{\partial f}{\partial x_i}} \quad (7-24)$$

这就是说，如果在数学规划法中按式（7—24）取步长 $\bar{\alpha}$ 值，而在优化准则法中是按式（7—16）计算 c_i 的值，则优化准则法和数学规划法是一致的。

第八章 机械结构优化设计举例

为了将前面所讨论的优化方法应用于机械结构的优化设计中，我们通过机床主轴结构优化的实例进行说明。在实例中将常用的优化方法贯穿起来，并在附录中配有相应的源程序全文。通过用不同优化方法所得到的优化设计方案的对比，进一步掌握各种优化方法的特点，以便在工程实践中灵活运用，取得较好的优化效果。

§ 1 机床主轴的优化问题

一、数学模型的建立

机床主轴一般多为阶梯轴形式，其支承也常为多支承系统。为了便于用材料力学公式进行主轴的强度和刚度计算，常将阶梯轴简化成以当量直径表示的等截面轴，并只分析两支承系统。如果要分析阶梯轴多支承系统，需要通过离散化，采用有限元法进行结构分析计算，这将在本章的后半部分进行讨论。

机床主轴内孔常常用于通过待加工的棒料，其直径往往不是按主轴的强度、刚度要求选取，所以主轴内径 d (图8—1) 可以不作为优化参数。主轴外径 D ，两支承距离 l 及伸出端长度 a 对主轴刚度、强度性能有很大影响，因此取这三个量作为优化参数 (设计变量) 进行优化。故设计变量向量 x 可表示为

$$x = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} l \\ D \\ a \end{Bmatrix}$$

机床主轴的变形对加工质量影响很大，因此对主轴性能的要求，主要表现为刚度要求。即要求主轴伸出端的挠度 y 不超过给定的静变形 y_0 ，即 $y \leq y_0$ 。根据这一要求可建立主轴静态性能约束条件

$$g(x) = y - y_0 \leq 0$$

在给定外力 P 的情况下，主轴伸出端 C 处的挠度 y 为设计变量 x 的函数，即 $y = y(x)$ 。根据材料力学中的莫尔 (Mohr) 积分公式

$$y = \int_L \frac{MM^0 dx}{EI}$$

(其中 M, M^0 分别为作用在 C 点处的外力 P 和单位力所引起的弯矩， L 为主轴全长。 E, I 分别为主轴材料的弹性模量和截面惯性矩。) 可算出 C 点挠度

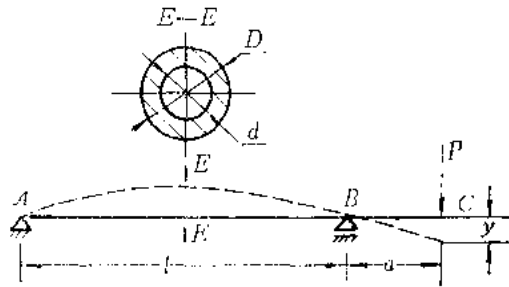


图 8—1

$$y = \frac{Pa^2(l+a)}{3EI}$$

其中 $I = \frac{\pi}{4}(R^4 - r^4) = \frac{\pi}{64}(D^4 - d^4)$

这样静态性能约束条件可表示为

$$g(x) = \frac{64P}{3\pi E} \cdot \frac{x_3^2(x_1 + x_3)}{x_2^4 - d^4} - y_0 \leq 0 \quad (8-1)$$

除上述反映刚度要求的性能约束条件外，还有反映优化参数变化范围的侧面约束条件

$$\begin{aligned} l_{min} &\leq l \leq l_{max} \\ D_{min} &\leq D \leq D_{max} \\ a_{min} &\leq a \leq a_{max} \end{aligned}$$

写成设计变量形式为

$$\begin{aligned} x_{i_{min}} - x_i &\leq 0 & (i = 1 \sim 3) \\ x_i - x_{i_{max}} &\leq 0 & (i = 1 \sim 3) \end{aligned} \quad (8-2)$$

主轴的优化问题是在满足上述约束条件下，使主轴结构重量为最小，即最轻设计。这样优化的目标函数为主轴的结构重量。

$$\begin{aligned} W(x) &= \gamma V \\ &= \frac{\pi\gamma}{4}(l+a)(D^2 - d^2) \\ &= \frac{\pi\gamma}{4}(x_1 + x_3)(x_2^2 - d^2) \end{aligned} \quad (8-3)$$

其中 γ 为主轴材料的比重， V 为主轴体积。

因此，主轴优化问题的数学模型可表示为：

$$\begin{aligned} \text{求} \quad & x = [x_1 \quad x_2 \quad x_3]^T \\ \text{使} \quad & W(x) = \frac{\pi\gamma}{4}(x_1 + x_3)(x_2^2 - d^2) \rightarrow \min \end{aligned} \quad (8-4)$$

要求满足

$$\begin{aligned} \frac{64P}{3\pi E} \cdot \frac{x_3^2(x_1 + x_3)}{x_2^4 - d^4} - y_0 &\leq 0 \\ x_{i_{min}} - x_i &\leq 0 & (i = 1 \sim 3) \\ x_i - x_{i_{max}} &\leq 0 & (i = 1 \sim 3) \end{aligned}$$

二、优化方法的选择

在第一章我们已经指出结构的优化方法，可以分成数学规划法和优化准则法两大类。数学规划法通常采用搜寻方式进行寻优，即按一定的搜寻方向进行走步，使目标函数值不断下降，最终找到最优点。数学规划法变化比较平稳，但运算次数较多，不适合复杂的结构分析计算（如有限元法的结构分析）。优化准则法是按照一定的优化准则来寻找最优点的，在实际使用时迭代次数较少，所以适合于复杂的结构分析计算。但是，正如第七章所指出的那样，该方法的主要困难，在于如何正确区分起作用约束和不起作

用约束，所以优化准则法多用于约束条件个数不多的优化问题。

对于机床主轴优化问题，由于采用材料力学的结构分析方法，同时约束条件仅有七个，所以既可以采用数学规划方法，也可以采用优化准则方法。下面我们分别采用这两种优化方法，对机床主轴进行结构优化设计，并将其优化结果作出分析比较。

§ 2 主轴的优化准则法设计

一、优化准则法公式

优化准则法是按照库恩——塔克条件所建立的优化准则进行优化，将优化问题中的性能约束和侧面约束分开进行考虑。对于性能约束又只考虑其中起作用的，对于不起作用的性能约束，将其相应的拉格朗日乘子取成零值，则自动消除其影响。对于上述机床主轴的优化问题，仅有一个静态变形的性能约束条件 $y \leq y_0$ ，根据第七章所给出的具有位移约束的优化准则，(7-9)式可写成

$$\frac{\partial W}{\partial x_i} + \lambda \frac{\partial y}{\partial x_i} \begin{cases} = 0 & \text{当 } x_{imin} < x_i^* < x_{imax} \\ \geq 0 & \text{当 } x_i^* = x_{imin} \\ \leq 0 & \text{当 } x_i^* = x_{imax} \end{cases} \quad (8-5)$$

$$\lambda(y - y_0) = 0$$

$$\lambda \geq 0$$

根据约束函数 y 的一阶泰勒展开和迭代后设计点应满足起作用约束条件的要求，可将上述优化准则写成如式(7-17)所表示的再设计迭代方程形式：

$$x_i^{K+1} = \begin{cases} c_i^K x_i^K & \text{当 } x_{imin} < c_i^K x_i^K < x_{imax} \\ x_{imin} & \text{当 } c_i^K x_i^K \leq x_{imin} \\ x_{imax} & \text{当 } c_i^K x_i^K \geq x_{imax} \end{cases} \quad (8-6)$$

其中乘子 c_i^K 可由式(7-16)得

$$c_i^K = a - \frac{1-a}{\frac{\partial W}{\partial x_i}} \lambda \frac{\partial y}{\partial x_i}$$

a 为迭代松弛因子，取 $a = 0.6 \sim 0.9$ 。按式(7-18)所反映的关系，可得

$$\lambda = \frac{y_0 - y(x^K) + (1-a) \sum_{i \in R_2} \frac{\partial y}{\partial x_i} x_i^K - \sum_{i \in R_{P1}} \frac{\partial y}{\partial x_i} (x_{imin} - x_i^K) - \sum_{i \in R_{P2}} \frac{\partial y}{\partial x_i} (x_{imax} - x_i^K)}{-(1-a) \sum_{i \in R_2} \left(\frac{\partial y}{\partial x_i} \right)^2 x_i^K / \frac{\partial W}{\partial x_i}} \quad (8-7)$$

在再设计迭代过程中，若设计点超出可行域时，则采取沿性能函数 $y(x)$ 的梯度方向进行走步，使其进入可行域。这种走步称作性能修改的梯度走步方式，其性能修改公式为

$$x_i^{K+1} = \begin{cases} x_i^K + \beta \frac{\partial y}{\partial x_i} & \text{当 } x_{i \min} < x_i^K + \beta \frac{\partial y}{\partial x_i} < x_{i \max} \\ x_{i \min} & \text{当 } x_i^K + \beta \frac{\partial y}{\partial x_i} \leq x_{i \min} \\ x_{i \max} & \text{当 } x_i^K + \beta \frac{\partial y}{\partial x_i} \geq x_{i \max} \end{cases} \quad (8-8)$$

其中

$$\beta = \frac{y_0 - y(x^K) - \sum_{i \in R_{P1}} \frac{\partial y}{\partial x_i} (x_{i \min} - x_i^K) - \sum_{i \in R_{P2}} \frac{\partial y}{\partial x_i} (x_{i \max} - x_i^K)}{\sum_{i \in R_2} \left(\frac{\partial y}{\partial x_i} \right)^2} \quad (8-9)$$

根据机床主轴优化问题数学模型中 $y(x)$, $W(x)$ 的具体形式, 代入到上述再设计迭代方程和性能修改公式中, 所用到的各偏导数如下

$$y(x) = \frac{64P}{3\pi E} \cdot \frac{x_3^2(x_1 + x_3)}{x_2^4 - d^4}$$

$$W(x) = \frac{\pi\gamma}{4} (x_1 + x_3) (x_2^2 - d^2)$$

$$\frac{\partial W}{\partial x_1} = \frac{\partial W}{\partial x_3} = \frac{\pi\gamma}{4} (x_2^2 - d^2) = \frac{W}{x_1 + x_3}$$

$$\frac{\partial W}{\partial x_2} = \frac{\pi\gamma}{4} (x_1 + x_3) \cdot 2x_2 = \frac{2Wx_2}{x_2^2 - d^2}$$

$$\frac{\partial y}{\partial x_1} = \frac{64P}{3\pi E} \cdot \frac{x_3^2}{x_2^4 - d^4} = \frac{y}{x_1 + x_3}$$

$$\frac{\partial y}{\partial x_2} = -\frac{64P}{3\pi E} \cdot \frac{x_3^2(x_1 + x_3)}{(x_2^4 - d^4)^2} \cdot 4x_2^3 = -\frac{4yx_2^3}{x_2^4 - d^4}$$

$$\frac{\partial y}{\partial x_3} = \frac{64P}{3\pi E} \cdot \frac{x_3(2x_1 + 3x_3)}{x_2^4 - d^4} = \frac{\partial y}{\partial x_1} \cdot \frac{2x_1 + 3x_3}{x_3}$$

二、主程序框图

计算机进行的是数值计算, 各量多取其近似值, 故一般不能判断两个量是否绝对相等, 只能用两个量足够接近来判断。所以, 对于约束条件 $y \leq y_0$, 采用

$$\left| \frac{y - y_0}{y_0} \right| < YEP \quad \text{或 } y < y_0$$

来代替, 其中 YEP 为性能容许误差范围带, 在计算时要予先给定。

当不满足上述约束条件时, 计算机应按 (8-8) 式进行性能修改。要从当前值 x_1 算出性能修改走步后的值 x_2 , 就需按 (8-9) 式计算 β 值, 而 β 值的计算又依赖于走步后 x_2 主、被动变量的划分, 从而使计算无从下手。常用的解决办法是采用试凑方法 (trial-and-error method)。开始计算时, 假定走步后全部参数都是主动参数, 从而可计算 β 和走步后的设计变量向量 x_2 。然后检验 x_2 的主、被动变量分类是否与原假设相同。如不

同，再按当前的 x_2 分类计算 β 值，一直进行到实际分类与假定分类一致为止。试凑法计算过程如图(8-2)所示。性能修改计算过程在程序中用子程序RDTZ(2)表示。

当满足性能约束条件，但其重量未收敛到给定精度WEP范围内时，则计算机应进行参数优化，使重量进一步减轻。这种形式的走步称作重量减轻步。重量减轻步主要按(8-6)(8-7)式计算 λ, C_i^k 值，从而由设计变量的当前值 x_1 算得走重量减轻步后的值 x_2 。这一计算过程编制成子程序RDTZ(1)，因为性能修改步计算 β 和重量减轻步计算 λ 公式相差不大，可用同一过程RDTZ采用不同的值参来调节。

当性能约束条件和重量收敛二者均已达到要求时，所得的设计变量当前值 x_1 ，即为优化参数的最佳值，并由计算机打印出“TERMINATE”(终止)字样，表示优化过程结束。

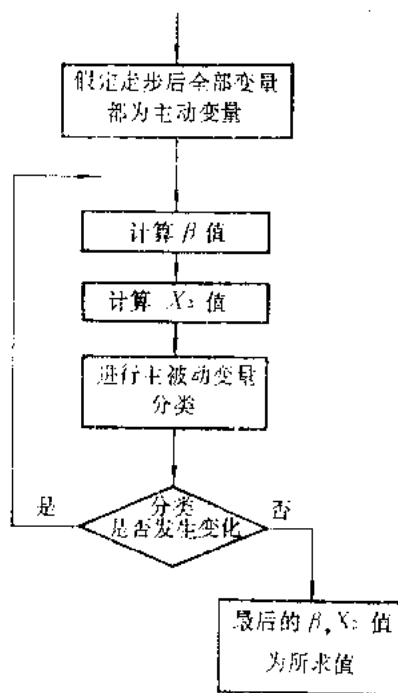


图 8-2

在优化过程中所用到的各量如 $y(x), W(x), \frac{\partial y}{\partial x_i}, \frac{\partial W}{\partial x_i}$ 等均由数学模型子程序 WYD 提供。从框图(8-3)中可以看到，优化过程中的 W_1, W_2 均属于满足性能约束条件的主轴结构重量，因为只有对这些重量进行比较才有意义。

优化准则法主程序框图如图(8-3)所示，用 FORTRAN-4 语言编制的源程序原文在附录一中列出。

三、计算结果分析

初始数据由表(8-1)给出。

表 8-1

N	$P(\text{kg})$	$DD(\text{cm})$	$y_0(\text{cm})$	YEP	WEP
3	1500	8	0.005	0.01	0.01
设计变量	$x_1(\text{cm})$	$x_2(\text{cm})$	$x_3(\text{cm})$		
初始值	48	10	12		
下限值	30	6	9		
上限值	65	14	15		

取迭代松弛因子 $\alpha = 0.7$ ，采用附录一程序，在 PDP11/23 型计算机上进行计算，其计算结果由表(8-2)给出。

h.

表 8-2

YIT	x_1 (cm)	x_2 (cm)	x_3 (cm)	W (kg)
10	30,000000	7,483065	9,000000	11,228251

从上述结果可看出，跨长 x_1 及伸出端 x_3 均到达下限值，而外径 x_2 取某一中间值。这是因为对于跨长和伸出端长，它们对重量的影响和对挠度的影响是一致的，即它们的值越小，重量越轻，挠度也越小，这从挠度公式

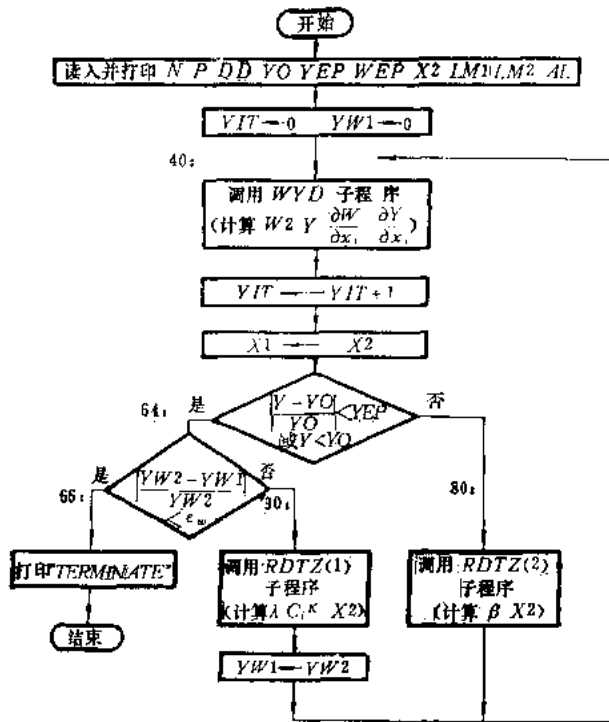


图 8-3

$y = \frac{Pa^2(l+a)}{3EI}$ 中即可看出。而对外径来说，外径越大，挠度越小，但重量也越重，所以若在限制挠度情况下使重量最轻，外径只能取某一中间值才是最优的。

§ 3 主轴的数学规划法优化设计

一、数学模型的简化

数学规划方法针对约束的处置不同，可分成约束的直接处理方法和约束的间接处理方法两类。这里采用的序列无约束极小化方法 (SUMT 法) 属于约束的间接处理方法，即将约束条件以制约项的形式加入到目标函数中形成广义目标函数，然后对广义目标函数进行序列无约束极小化处理，从而得到原来约束优化问题的最优解。

由于约束条件的种类不同，约束函数值可能在数值上差别很大。为了照顾到所有约

束函数在广义目标函数中的比重,需将约束条件进行归一化处理,即将约束函数中各项取值接近于1。另外,约束条件应表示成在可行域内取正值,这样由约束函数所构成的制约函数项在可行域内也会取正值,以便进行无约束极小化处理。因此,原机床主轴优化问题的数学模型简化成如下形式:

$$\begin{aligned} \text{求} \quad & \mathbf{x} = [x_1 \quad x_2 \quad x_3]^T \\ \text{使} \quad & f(\mathbf{x}) = \frac{\pi Y}{4} (x_1 + x_3)(x_2^2 - d^2) \rightarrow \min \end{aligned}$$

$$\text{要求满足} \quad g_{x_1}(\mathbf{x}) = 1 - \frac{Px_3^2(x_1 + x_3)}{3EIy_0} \geq 0 \quad (8-10)$$

$$g_{x_2}(\mathbf{x}) = \frac{x_1}{x_{1\min}} - 1 \geq 0$$

$$g_{x_3}(\mathbf{x}) = \frac{x_3}{x_{2\min}} - 1 \geq 0$$

$$g_{x_4}(\mathbf{x}) = 1 - \frac{x_2}{x_{2\max}} \geq 0$$

$$g_{x_5}(\mathbf{x}) = \frac{x_2}{x_{3\min}} - 1 \geq 0$$

在约束条件中未将 $x_1 \leq x_{1\max}$ 和 $x_3 \leq x_{3\max}$ 考虑进去,这是因为在优化过程中,以满足伸出端挠度不超过给定值为条件,尽量减少主轴结构重量。无论从减小伸出端挠度要求上看,还是从降低主轴重量上看,都要求主轴跨距长 x_1 和伸出端长 x_3 往小处变化,所以对其上限值可以不作为约束条件处理。这样可以消去一些不必要的约束条件,更有利于优化问题的求解。

二、优化的具体方法

从优化问题的数学模型中可以看出,本问题只含不等式约束而不含等式约束,所以可以采用SUMT法中的障碍函数法(通常称作SUMT内点法),将约束优化问题转化为无约束优化问题。采用内点法的好处是优化过程中的中间结果都满足约束条件,即属于可行解,都可以作为改进设计的参考。和(5-27)式相似,广义目标函数 $y(\mathbf{x}, r)$ 可表示成

$$\begin{aligned} y(\mathbf{x}, r) &= f(\mathbf{x}) + r \cdot SG(\mathbf{x}) \\ &= f(\mathbf{x}) + r \sum_{k=1}^{s+k} \frac{1}{Gx_k(\mathbf{x})} \end{aligned} \quad (8-11)$$

$$\text{其中} \quad SG(\mathbf{x}) = \sum_{k=1}^{s+k} \frac{1}{Gx_k(\mathbf{x})}$$

为障碍函数项, r 为障碍因子。 r 值在无约束优化过程中逐渐减小而趋于零,结果使最终的广义目标函数 $y(\mathbf{x}, r)$ 趋于原约束优化问题的目标函数 $f(\mathbf{x})$ 。

对于SUMT内点方法,要求初始点 \mathbf{x}^0 在可行域内。一旦初始点选在可行域内,则通过障碍函数的作用,使在以后的无约束极小化过程中,设计点均可自动保证在可行域内。

对于广义目标函数 $y(x, r)$ 的无约束极小化, 本问题采用鲍威尔方法。第一轮搜寻方向采用 n 个坐标方向, 即 $p_i = e_i (i = 1 \sim n)$ 。经过 n 次一维搜索, 设计点由 x^0 移动到 x_2 , 则得一模式方向 $p_{n+1} = x_2 - x^0$ 。根据一轮起始点 x^0 , 终点 x_2 , 反射点 $x_3 = 2x_2 - x^0$ 三点处的广义目标函数值 F_0, F_2, F_3 以及两个相邻搜索方向上广义目标函数值之差的极大值 DFM 是否满足条件

$$2DFM > F_0 - 2F_2 + F_3$$

来决定是否吸收 p_{n+1} 到下一轮搜寻方向组中来。如果满足上述不等式关系, 则淘汰 DFM 所对应的 p_{DFM} 方向, 而吸收 p_{n+1} 并排在方向组的最后, 然后沿 p_{n+1} 进行一维搜索得到下轮的初始点。如果不满足上述关系式, 则维持原来搜索方向不变, 初始点就取终点 x_2 和反射点 x_3 中广义目标函数值的较小者。鲍威尔方法优化过程在程序中用子程序 MINM 表示。

MINM 子程序中所用到的一维搜索采用二次插值法, 用“高一低一高”三点 (a_1, y_1) 、 (a_2, y_2) 和 (a_3, y_3) 决定一抛物线, 取其最低点 a_4 作为缩小搜索区间的内插点。 a_4 的大小由下列公式求得

$$a_4 = 0.5(a_1 + a_2 - c_1/c_2)$$

其中

$$c_1 = (y_3 - y_1)/(a_3 - a_1)$$

$$c_2 = \left(\frac{y_2 - y_1}{a_2 - a_1} - c_1 \right) / (a_2 - a_3)$$

利用区间消去法原理, 不断缩小搜索区间, 当区间缩小到足够小或函数值足够接近时, 停止搜索而选得一维搜索最佳点 a^* 。这一过程在程序中用子程序 LINE 描述, 其中 h_0 表示初始试验步长增量, t 代表步长 a_t 。

三、程序结构与框图

整个程序分为数学模型过程 FGX(x), 序列无约束极小化过程 SUMT 和主程序三部分。在 SUMT 过程中又包含有求函数值子程序 FUNCP, PENA, 一维搜索子程序 LINE 和鲍威尔方法子程序 MINM。我们将主程序和 SUMT 子程序框图表示在图 (8—4) 中。

框图中首先输入原始数据, 包括主轴内径 DD , 伸出端受力 P , 允许挠度 DER (即 y_0), 优化问题维数 n , 约束个数 gK , 收敛精度 $EP1$ 和 $EP2$, 初始设计点 x 及设计变量的上下限数组 BV 和 BL 。障碍因子 r 以控制台变量形式输入。另外还输入障碍因子缩小倍数 c 和一维搜索试验步长增量 h_0 。

然后在某一障碍因子下进行无约束极小化, 是保证初始设计点在可行域内的情况下进行无约束极小化过程的。根据 SUMT 内点法的要求, 设计点应当始终保持在可行域内, 所以对无约束极小化后所得的设计点, 仍然需要检验各约束条件是否全被满足。在满足全部约束条件的前提下, 降低障碍因子 r 值, 进行下一轮的无约束极小化。这样逐次减少障碍因子, 直至广义目标函数值不再进一步下降为止, 最后打印出 20 个 E , 20 个 G 和优化结果, 并算出目标函数值的下降相对量。

如果在进行某次无约束极小化之后, 设计点越出可行域, 这时只打印 20 个 G , 而不打印 20 个 E , 这表示未取得优化结果, 但此方案可作参考给予输出。

框图中有关符号的意义统一在附录二中进行说明。

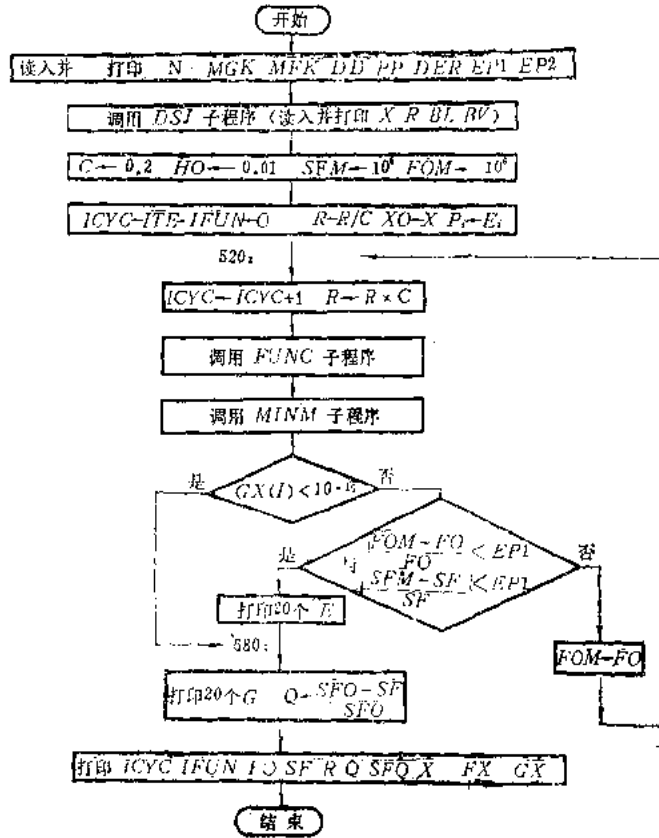


图 8-4

四、计算结果分析

初始数据由表 (8-3) 给出。

表 8-3

$DD(\text{cm})$	$P(\text{kg})$	$DER(\text{cm})$	N	gK	$EP1$	$EP2$
9	1.5×10^3	5×10^{-3}	3	5	10^{-7}	10^{-5}
设计变量		$x_1(\text{cm})$		$x_2(\text{cm})$		$x_3(\text{cm})$
初始值		48		10		12
下限值		30		6		9
上限值		65		14		15

采用附录二源程序, 在 Z80 型计算机上进行计算, 其计算结果如表 (8-4) 所示。

表 8—4

ICYC	ITE	IFUN	FO	FX	R
10	23	553	11,256605	11,256605	$0,512 \times 10^{-12}$
$x_1(\text{cm})$		$x_2(\text{cm})$		$x_3(\text{cm})$	
30,000231		7,490966		9,000052	

其中表明整个优化共进行10个序列的无约束极小化，含有23轮搜索，调用函数共达 553次。

从以上计算结果可以看出，由于障碍因子变得非常小，因此广义目标函数中的障碍项实际上已消失，广义目标函数值已非常接近原优化问题的目标函数值。其优化结果和前节优化准则法所得结果已非常接近，二者比较见表（8—5）

表 8—5

类别	$x_1(\text{cm})$	$x_2(\text{cm})$	$x_3(\text{cm})$	$W(\text{kg})$
优化准则法	30,000000	7,483065	9,000000	11,228251
数学规划法	30,000231	7,490966	9,000052	11,256605
初始设计	48,000000	10,000000	12,000000	33,448509

§ 4 主轴的有限元——优化设计

一、具有静、动态约束的优化问题

将阶梯轴简化成当量直径的等截面轴只是一种近似计算方法，实际上主轴多为阶梯轴形状，为了计算阶梯轴的静变形，振动频率或振型等，常采用有限元法进行结构分析。

图（8—5）所示的机床主轴为三支承系统，受力和力矩作用。本优化问题除要求伸出端点挠度 y 不超过给定值 $y_0 = 0,005\text{cm}$ 外，还要求主轴结构的第一阶自振频率 f 不低于给定值 $f_0 = 1200\text{Hz}$ 。对结构尺寸的比例要求是两支点跨长与伸出端长之比为 3。在

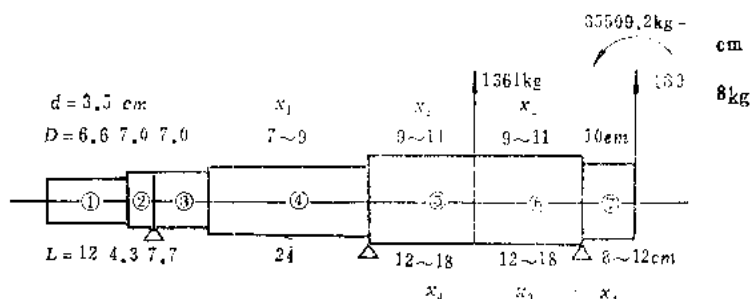


图 8—5

上述条件下进行最轻设计。

根据有限元法可将主轴结构分离成7个单元，每个单元都有一些结构尺寸参数。一般都选取对结构静、动态性能以及结构重量影响较大的结构尺寸参数作为优化参数。对上述主轴结构，我们选④⑤⑥单元的外径及⑤⑥⑦单元的长度作为优化参数。考虑具体结构，有些几何尺寸是相同的，这些优化参数可以合并，以便降低优化问题的维数。如⑤⑥单元的外径和长度均一致，可将以四个结构参数作为两个优化参数 x_2, x_3 处理。为了便于编制程序，一般将优化参数和结构参数的对应关系用一信息矩阵 W 表示。矩阵的行对应同一优化参数，矩阵的列对应同一单元数。矩阵元素为1表示优化参数和结构参数有对应关系，矩阵元素为零表示二者无对应关系。

$$W(1:4, 1:7) = \begin{matrix} & \textcircled{1} & \textcircled{2} & \textcircled{3} & \textcircled{4} & \textcircled{5} & \textcircled{6} & \textcircled{7} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

反映优化参数变化范围的侧面约束条件为

$$x_{i \min} \leq x_i \leq x_{i \max} \quad (i = 1 \sim 4)$$

除侧面约束外，还有反映静、动态性能及结构尺寸比例的限制，这些称作性能约束条件。如

$$\begin{aligned} y &\leq y_0 \\ f &\geq f_0 \\ u &= L[6]/L[7] = u_0 \end{aligned}$$

其中 u_0 为给定的⑥⑦单元跨长比，对于不同的结构形式， u_0 也可以变化。

这样，具有静、动态约束条件的结构优化问题，其数学模型可表示为：

$$\begin{aligned} \text{求} & \quad \mathbf{x} = [x_1, x_2, \dots, x_n]^T \\ \text{使} & \quad W(\mathbf{x}) \rightarrow \min \end{aligned} \quad (8-12)$$

$$\text{要求满足} \quad H(\mathbf{x}) = \frac{u}{u_0} - 1 = 0$$

$$G_1(\mathbf{x}) = \frac{y}{y_0} - 1 \leq 0$$

$$G_2(\mathbf{x}) = 1 - \frac{f}{f_0} \leq 0$$

$$G_i(\mathbf{x}) = x_{i \min} - x_i \leq 0 \quad (i = 1 \sim n, j = i + 2)$$

$$G_j(\mathbf{x}) = x_i - x_{i \max} \leq 0 \quad (i = 1 \sim n, j = n + 2 + i)$$

上述约束条件既包括等式约束和不等式约束，也包括性能约束及侧面约束，因此本优化问题具有代表性。

二、优化方法的选择

具有静、动态性能约束和结构尺寸侧面约束的优化问题，当结构形状较复杂需要用有限元法进行结构分析计算时，应考虑合适的优化方法。我们在前两节把机床主轴简化成当量直径的等截面轴，采用材料力学分析方法进行结构计算，分别用优化准则法和数学规划法进行优化设计。优化结果表明优化准则法收敛较快，迭代次数仅需10次，即总共只进行10次的结构重分析计算，而数学规划方法需要进行10次序列无约束极小化处理，总共需进行23轮搜索，553次结构的重分析计算，这说明此法迭代次数较多，收敛很慢。当采用材料力学方法进行结构的重分析计算时，由于计算公式简单，每次重分析计算工作量并不大，迭代次数多的问题还不太突出。但若采用有限元法进行结构的重分析计算时，每次重分析都要重新组集刚、质阵，重新解一组多元联立方程式，这将会占用一定的计算机机时。如果采用上述的数学规划方法，进行500多次的结构重分析计算，所用计算机机时数将会是非常大的。因此，结构的有限元——优化设计不宜采用数学规划方法。

采用优化准则方法进行结构的有限元——优化设计，由于迭代次数少，收敛快，是比较现实可行的。在优化准则法中，需要约束函数的导数信息，因此要求在用有限元法进行结构分析的同时，还要求计算结构的静、动态性能对优化参数的导数（即静、动态性能响应量）。虽然计算性能响应量要花费一定的计算机机时，但由于总的迭代次数只不过10次，因此进行有限元——优化设计计算时间还是有限的。

三、优化准则法的具体公式

根据具有静、动态性能约束和结构尺寸侧面约束的主轴有限元——优化问题，在最优点处应满足如式(5—17) (5—18)所表示的库恩——塔克条件，具体表示为

$$\begin{aligned} \frac{\partial W(x^*)}{\partial x_i} + \lambda_0 \frac{\partial H(x^*)}{\partial x_i} + \sum_{j=1}^{2n+2} \lambda_j \frac{\partial G_j(x^*)}{\partial x_i} &= 0 \quad (i=1 \sim n) \\ \lambda_j G_j(x^*) &= 0 \quad (j=1 \sim 2n+2) \\ \lambda_j &\geq 0 \quad (j=1 \sim 2n+2) \end{aligned} \quad (8-13)$$

现将侧面约束

$$\begin{aligned} G_j(x) &= x_{i_{min}} - x_i \leq 0 \quad (i=1 \sim n, j=i+2) \\ G_j(x) &= x_i - x_{i_{max}} \leq 0 \quad (i=1 \sim n, j=n+i+2) \end{aligned}$$

共2n个约束条件中拿出，单独予以考虑，并注意到三个性能约束函数

$$\begin{aligned} H(x) &= \frac{u}{u_0} - 1 \\ G_1(x) &= \frac{y}{y_0} - 1 \\ G_2(x) &= 1 - \frac{f}{f_0} \end{aligned}$$

对优化参数的导数与结构性能响应量有以下关系：

$$\frac{\partial H(\mathbf{x})}{\partial x_i} = \frac{1}{u_0} \frac{\partial u(\mathbf{x})}{\partial x_i}$$

$$\frac{\partial G_1(\mathbf{x})}{\partial x_i} = \frac{1}{y_0} \frac{\partial y(\mathbf{x})}{\partial x_i} \quad \frac{\partial G_2(\mathbf{x})}{\partial x_i} = -\frac{1}{f_0} \frac{\partial f(\mathbf{x})}{\partial x_i}$$

从上述库恩——塔克条件中可推得最优解 \mathbf{x}^* 所应满足的优化准则是

$$\frac{\partial W(\mathbf{x}^*)}{\partial x_i} + \frac{\lambda_0}{u_0} \frac{\partial u(\mathbf{x}^*)}{\partial x_i} + \frac{\lambda_1}{y_0} \frac{\partial y(\mathbf{x}^*)}{\partial x_i} - \frac{\lambda_2}{f_0} \frac{\partial f(\mathbf{x}^*)}{\partial x_i}$$

$$\begin{cases} = 0 & \text{当 } x_{i \min} < x_i^* < x_{i \max} \\ \geq 0 & \text{当 } x_i^* = x_{i \min} \\ \leq 0 & \text{当 } x_i^* = x_{i \max} \end{cases} \quad (8-14)$$

$$\lambda_1 \geq 0$$

$$\lambda_2 \geq 0$$

迭代形式的再设计方程如(7-17)式,即可写为

$$x_i^{K+1} = \begin{cases} C_i^K x_i^K & \text{当 } x_{i \min} < C_i^K x_i^K < x_{i \max} \\ x_{i \min} & \text{当 } C_i^K x_i^K \leq x_{i \min} \\ x_{i \max} & \text{当 } C_i^K x_i^K \geq x_{i \max} \end{cases}$$

$$\text{其中 } C_i^K = \alpha - \frac{1-\alpha}{\frac{\partial W(\mathbf{x}^K)}{\partial x_i}} \left(\frac{\lambda_0}{u_0} \frac{\partial u(\mathbf{x}^K)}{\partial x_i} + \frac{\lambda_1}{y_0} \frac{\partial y(\mathbf{x}^K)}{\partial x_i} - \frac{\lambda_2}{f_0} \frac{\partial f(\mathbf{x}^K)}{\partial x_i} \right) \quad (8-15)$$

再设计后的设计点应处于性能约束曲面上,即有

$$z(\mathbf{x}^{K+1}) = z_0$$

(z, z_0 分别表示性能函数值及其给定值),通过泰勒展开,有如(7-19)式所表示的

$$z(\mathbf{x}^{K+1}) = z(\mathbf{x}^K) + \sum_{i \in R_a} \frac{\partial z(\mathbf{x}^K)}{\partial x_i} [C_i^K x_i^K - x_i^K]$$

$$+ \sum_{i \in R_{p_1}} \frac{\partial z(\mathbf{x}^K)}{\partial x_i} [x_{i \min} - x_i^K] + \sum_{i \in R_{p_2}} \frac{\partial z(\mathbf{x}^K)}{\partial x_i} [x_{i \max} - x_i^K] = z_0$$

将 C_i^K 的关系式(8-15)代入,即可得到关于 $\lambda_0, \lambda_1, \lambda_2$ 的三元一次联立方程,如(7-21) (7-22)式所给出的

$$a_{r0} \lambda_0 + a_{r1} \lambda_1 + a_{r2} \lambda_2 = b_r \quad (r=0,1,2)$$

其中

$$a_{r0} = -\frac{1-\alpha}{u_0 z_0} \sum_{i \in R_a} \frac{\frac{\partial z}{\partial x_i} \cdot x_i^K}{\frac{\partial W}{\partial x_i}} \cdot \frac{\partial u}{\partial x_i}$$

$$a_{r1} = -\frac{1-\alpha}{y_0 z_0} \sum_{i \in R_{p_1}} \frac{\frac{\partial z}{\partial x_i} \cdot x_i^K}{\frac{\partial W}{\partial x_i}} \cdot \frac{\partial y}{\partial x_i} \quad (8-16)$$

$$a_{r2} = \frac{1-a}{f_0 z_0} \sum_{i \in R_a} \frac{\partial z}{\partial x_i} \cdot x_i^K \cdot \frac{\partial f}{\partial x_i}$$

$$b_r = 1 - \frac{z}{z_0} - \frac{1}{z_0} \sum_{i \in R_{P1}} \frac{\partial z}{\partial x_i} [x_{i \min} - x_i^K] - \frac{1}{z_0} \sum_{i \in R_{P2}} \frac{\partial z}{\partial x_i} [x_{i \max} - x_i^K] \\ + \frac{1-a}{z_0} \sum_{i \in R_a} \frac{\partial z}{\partial x_i} x_i^K$$

当 $r=0$ 取 $z=u$, $z_0=u_0$

当 $r=1$ 取 $z=y$, $z_0=y_0$

当 $r=2$ 取 $z=f$, $z_0=f_0$

关于主、被动变量集合 R_a, R_{P1}, R_{P2} 的划分, 仍利用试凑法。最后解出 $\lambda_0, \lambda_1, \lambda_2$, 算出 c_i^K 从而求得走步后的设计点 x^{K+1} 。

四、性能求导公式

性能求导包括静变形求导和固有频率求导两部分。通过有限元法求解静态变形和固有频率时, 采用的是平面梁单元, 其单元刚阵 $[k]$ 有如下形式。

$$[k] = EI \begin{pmatrix} \frac{12}{l^3} & & & & & \\ \frac{6}{l^2} & \frac{4}{l} & \text{对称} & & & \\ -k_{11} & -k_{21} & k_{11} & & & \\ k_{21} & \frac{2}{l} & -k_{21} & k_{22} & & \end{pmatrix}$$

由单元刚阵组集成总刚阵 $[K]$ 。若各结点处的静变形 (挠度和转角) 和外载荷分别用 $\{y\}$ $\{F\}$ 表示, 则有如 (7-10) 式所表示的

$$[K] \{y\} = \{F\} \quad (8-17)$$

将方程两边分别对优化参数 x_i 求导, 得

$$[K] \frac{\partial \{y\}}{\partial x_i} = \frac{\partial \{F\}}{\partial x_i} - \frac{\partial [K]}{\partial x_i} \{y\} \quad (8-18)$$

由于优化参数是取外径和跨长, 可以很容易求得 $\frac{\partial [K]}{\partial x_i}$ 的具体表达式。从上述两方程中即可解出 $\{y\}, \frac{\partial \{y\}}{\partial x_i}$ 。对于本优化问题, 静态变形约束条件是指伸出端挠度, 所以只取

$\{y\}, \frac{\partial \{y\}}{\partial x_i}$ 列向量中的倒数第二个元素。

主轴振动的有限元方程为 (7-12) 式所表示的

$$[K] \{\tilde{y}\} = \omega^2 [M] \{\tilde{y}\} \quad (8-19)$$

其中 $\{\tilde{y}\}, \omega$ 为振型及其相应自振角频率

$$\omega = 2\pi f$$

[M] 为结构总质阵，由单元质阵[m]组集而成。对于平面梁单元，单元质阵为

$$[m] = \frac{\rho A}{980} \begin{bmatrix} \frac{13l}{35} & & & & & \\ & \frac{11l^2}{210} & l^3 & & & \\ & & 105 & & & \\ & \frac{9l}{70} & \frac{13l^2}{420} & & m_{11} & \\ & & & & & \\ -m_{32} & & -\frac{l^3}{140} & & -m_{21} & m_{22} \end{bmatrix} \quad \text{对称}$$

其中 ρ 代表材料密度， A 为单元横截面积。当用反迭代法求解上述振动方程时，需将它变成相对于规格化振型 $\{\tilde{y}\}_{\text{规}}$ 的迭代形式：

$$XL[K]\{\tilde{y}\}_{\text{规}}^{K+1} = [M]\{\tilde{y}\}_{\text{规}}^K \quad (8-20)$$

其中 XL 为 $K+1$ 次迭代振型的最大绝对值，迭代一直进行到 $\{\tilde{y}\}_{\text{规}}^{K+1}$ 足够接近 $\{\tilde{y}\}_{\text{规}}^K$ 时为止。最后得到规格化振型 $\{\tilde{y}\}_{\text{规}}$ ，一阶自振频率

$$f = \frac{1}{2\pi} \sqrt{\frac{1}{XL}}$$

和法式化振型

$$\{y_1\} = \frac{\{\tilde{y}\}_{\text{规}}}{u}$$

其中

$$u = \sqrt{\{\tilde{y}\}_{\text{规}}^T [M] \{\tilde{y}\}_{\text{规}}}$$

对于法式化振型，惯性力在它上所作的功为 1，即有

$$\{y_1\}^T [M] \{y_1\} = 1$$

利用此性质可得自振频率对优化参数的导数，如 (7-13) 式所表示的

$$\frac{\partial f}{\partial x_i} = \frac{1}{8\pi^2 f} \{y_1\}^T \frac{\partial [K]}{\partial x_i} \{y_1\} - \frac{f}{2} \{y_1\}^T \frac{\partial [M]}{\partial x_i} \{y_1\} \quad (8-21)$$

优化参数取外径和跨长， $\frac{\partial [M]}{\partial x_i}$ 可从 [M] 的各元素的解析表达式进行求导而得。

五、综合性能修改

在进行优化再设计时，一般要求设计点处在由等式和不等式约束条件

$$H(x) = \frac{u}{u_0} - 1 = 0$$

$$G_i(x) = \frac{y}{y_0} - 1 \leq 0$$

$$G_2(x) = 1 - \frac{f}{f_0} \leq 0$$

所决定的可行域内。在优化过程中如果设计点跑出可行域，也还需要将设计点重新拉回到可行域内。对于多个性能约束条件，一般选取设计点沿各性能约束函数梯度的线性组合方向进行走步。即取主动变量的变化为

$$\Delta x_i^K = x_i^{K+1} - x_i^K = \beta_0 \frac{\partial H(x^K)}{\partial x_i} + \beta_1 \frac{\partial G_1(x^K)}{\partial x_i} + \beta_2 \frac{\partial G_2(x^K)}{\partial x_i}$$

将具体约束条件代入，得

$$\Delta x_i^K = \frac{\beta_0}{u_0} \frac{\partial u(x^K)}{\partial x_i} + \frac{\beta_1}{y_0} \frac{\partial y(x^K)}{\partial x_i} - \frac{\beta_2}{f_0} \frac{\partial f(x^K)}{\partial x_i} \quad (8-22)$$

根据走步后 $z(x^{K+1}) = z_0$ 来确定系数 $\beta_0, \beta_1, \beta_2$ 的值。同再设计迭代求解 $\lambda_0, \lambda_1, \lambda_2$ ，相似，经整理可得关于 $\beta_0, \beta_1, \beta_2$ 的三元一次联立方程

$$A_{r0}\beta_0 + A_{r1}\beta_1 + A_{r2}\beta_2 = B_r \quad (r=0, 1, 2) \quad (8-23)$$

其中

$$A_{r0} = -\frac{1}{u_0 z_0} \sum_{i \in R_a} \frac{\partial z}{\partial x_i} \cdot \frac{\partial u}{\partial x_i}$$

$$A_{r1} = \frac{1}{y_0 z_0} \sum_{i \in R_a} \frac{\partial z}{\partial x_i} \cdot \frac{\partial y}{\partial x_i}$$

$$A_{r2} = -\frac{1}{f_0 z_0} \sum_{i \in R_a} \frac{\partial z}{\partial x_i} \cdot \frac{\partial f}{\partial x_i}$$

$$B_r = 1 - \frac{z}{z_0} - \frac{1}{z_0} \sum_{i \in R_{p1}} \frac{\partial z}{\partial x_i} [x_{i_{min}} - x_i^K] - \frac{1}{z_0} \sum_{i \in R_{p2}} \frac{\partial z}{\partial x_i} [x_{i_{max}} - x_i^K]$$

$$\text{当 } r=0 \text{ 时, } z=u \quad z_0=u_0$$

$$\text{当 } r=1 \text{ 时, } z=y \quad z_0=y_0$$

$$\text{当 } r=2 \text{ 时, } z=f \quad z_0=f_0$$

比较综合性能修改和再设计迭代，可发现二者之间有很多相似之处。为了区分设计变量中的主、被动参数，也都需要按照试凑方法进行多次试凑，才能最后定出方程各系数，解出 β 或 λ ，从而求得 x^{K+1} 。这样在程序上可用一个过程 RDTZ(Q) 来描述，引入值参 Q 来区分二者的不同，从而可简化程序，便于计算。

六、程序结构及框图

主轴有限元——优化设计的源程序包括三部分内容，一是结构分析及响应量计算子程序 FHG，二是优化再设计迭代及性能综合修改子程序 RDTZ，三是优化准则法设计子程序 CRITI。

结构分析及响应量计算过程主要是根据结构的参数按有限元方法，对结构静、动态特性进行分析计算，并求出它们对优化参数的导数。此过程在程序中以 FHG 子程序表示。

优化再设计迭代及性能综合修改过程在程序中以 RDTZ(Q) 表示。当 $Q=1$ 时只进行优化再设计迭代，即对优化参数，按照优化准则进行优化再设计。由于主、被动变量的划分是采用试凑方法，所以每走一步重量减轻步都有一个试凑过程。当设计点未进入可

行域时，应进行性能修改，属于 $Q=2$ 情况，步长计算仍采用试凑方法。性能综合修改一直进行到设计点进入可行域为止，然后再转入到 $Q=1$ 进行优化再设计。

程序里所反映的迭代前后结构重量的更替与比较，是在结构满足全部约束条件的情况下，也就是在可行解的范围内进行。当重量收敛到一定程度时，计算机打印出结束信息“FINAL SOLUTION”（最后结果），表示计算结束，最后所得的迭代点即为最优点。为了加速优化过程，计算表明无论设计点是否在可行域内，第一步都采用重量减轻步，效果较好。因此在程序设计中，应首先检查迭代次数，当迭代次数 $YIT=1$ 时，不作约束条件的检查就直接转到执行优化再设计子程序。

上述计算步骤均由主程序加以控制，整个计算程序粗框图如图（8—6）所示。由 FORTRAN 语言编制的源程序由附录三给出，

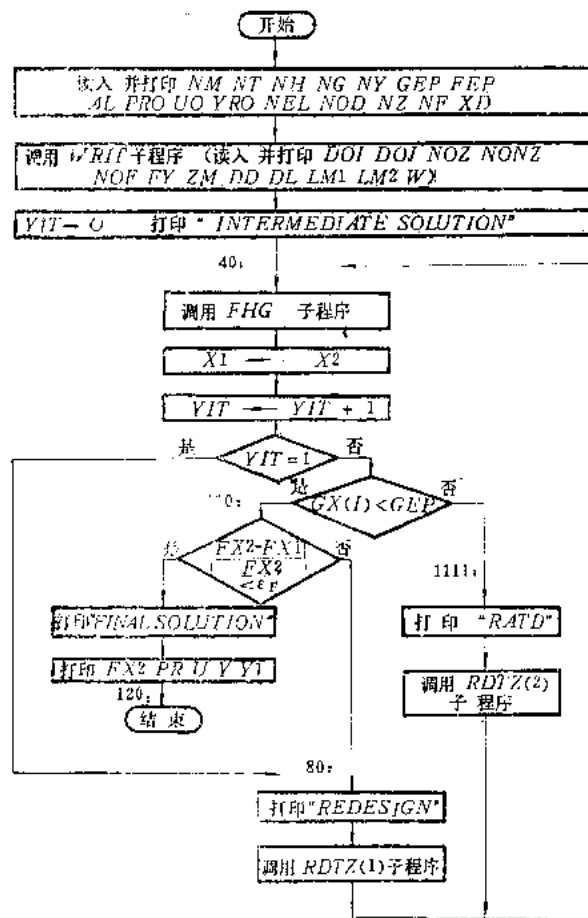


图 8—6

七、计算结果分析

采用附录三源程序，在 UNIVAC—1100 型计算机上进行计算。初始设计方案与最佳设计方案在结构尺寸参数，结构静、动态性能以及结构重量方面的比较如表（8—6）所示。

在最佳参数中，外径比初始方案有所降低，取为上下限之间的中间值，这是由于结构性能与结构重量对外径有相反趋势的要求。即从减轻结构重量角度，要求外径向其下限方向变化，而从满足静、动特性要求出发，又要求外径向其上限方向变化，所以最后取上下限之间的某一最佳值。但对跨长来说，不论从减轻结构重量还是从满足静、动态

表 8—6

项 目	初 始 设 计	最 佳 设 计
x_1 (cm)	8	7,15175
x_2 (cm)	10	9,05796
x_3 (cm)	15	12,00000
x_4 (cm)	10	8,00000
u	1,500	1,500
y (cm)	0,006099	0,005019
f (Hz)	1252,7894	1199,4610
W (kg)	34,114566	25,284759

性能要求出发，都是向下限方向变化，因此最佳值取到跨长的下限。优化结果表明，它既满足所提出的静、动态性能要求，又使结构重量有明显的减轻。

优化的具体过程如表 (8—7) 所示。

表 8—7

YIT	u	y (cm)	f (Hz)	W (kg)	说 明
0	1,5000	0,00609966	1252,7894	34,1146	初 始 值
1	1,5000	0,00509100	1202,0485	25,2327	重 量 减 轻
2	1,5000	0,00500133	1199,9397	25,2978	性 能 修 改
3	1,5000	0,00510795	1186,0912	24,8151	重 量 减 轻
4	1,5000	0,00500187	1199,4610	25,2848	性 能 修 改

从优化过程中可看出，总迭代次数只有 4 次 (一般均小于 10 次)，便于采用有限元方法进行结构的重分析计算。重量减轻和性能修改两种走步交替进行，使设计点始终保持在可行域内进行优化，从而保证设计点的移动稳定地趋向最优点。

符号说明

1. 主轴的优化准则法设计主程序 ZZF 中全局量:
 - N 设计变量个数
 - P 主轴端部所受的外力 (kg)
 - DD 主轴内径 (cm)
 - Y 主轴端部挠度 (cm)
 - Y0 主轴端部挠度的最大允许值
 - YEP 挠度的最大允许相对偏差 ϵ_y
 - YW1 再设计迭代前的主轴重量
 - YW2 再设计迭代后的主轴重量
 - WEP 主轴重量的允许收敛精度 ϵ_w
 - YIT 迭代的次序号
 - AL 再设计迭代松弛因子 α
 - TI(N) 设计变量次序号
 - X0(N) 标志设计变量为主、被动性质的信息数组
 - X1(N) 迭代前的设计变量数组
 - X2(N) 迭代后的设计变量数组
 - LM1(N) 设计变量的下界值数组
 - LM2(N) 设计变量的上界值数组
 - DW(N) 重量对设计变量的导数值数组
 - DY(N) 挠度对设计变量的导数值数组
2. 计算重量、变形及其对设计变量偏导数子程序 WYD 中局部量:
 - PI 圆周率 π
 - E 材料弹性模量 (kg/cm²)
 - GAM 材料比重 γ (kg/cm³)
3. 再设计迭代和性能修改子程序 RDTZ 中局部量:
 - Q 参量, 当 $Q=1$ 时作再设计迭代, 当 $Q=2$ 时作性能修改
 - CHA 确定设计变量的主、被动性质所作的迭代序号
 - LU 拉格朗日乘子 λ
 - MU 性能修改系数 β
 - J 主动变量个数
 - JJ 主动变量转化为被动变量的个数
 - A,B 计算 λ, β 时所用的工作单元
 - CCI(N) 系数 C_i 值数组

主 轴 的 优 化 准 则 法 源 程 序

(用FORTRAN-4语言编写, 在PDP 11/23型计算机上执行)

```

PROGRAM ZZP
DIMENSION X0(10),X1(10),X2(10),LM1(10),LM2(10),DW(10),DY(10)
1 TI(10),CCI(10)
INTEGER YII,X0,TI
REAL X1,X2,LM1,LM2,DW,DY
READ (5,*)N,P,DD,Y0,YEP,WEP
READ (5,*)(X2(I),I=1,N)
READ (5,*)(LM1(I),I=1,N)
READ (5,*)(LM2(I),I=1,N)
READ (5,*)AL
WRITE(6,5)
5 FORMAT(//10X,14HOPTIMAL-DESIGN,2X,3H(1)//)
WRITE(6,10)N,P,DD,Y0,YEP,WEP
10 FORMAT(2X,2HN = ,I3/2X,2HP = ,F15.8,5X,3HDD = ,F15.8,/2X,3HY0 = ,F
15X,4HYEP = ,F14.8,5X,4HWEP = ,F14.8)
DO 20 I=1,N
20 TI(I) = I
WRITE(6,105)(TI(I),I=1,N)
WRITE(6,110)(X2(I),I=1,N)
WRITE(6,120)(LM1(I),I=1,N)
WRITE(6,130)(LM2(I),I=1,N)
105 FORMAT(2X,3HI = ,3(2X,I15))

```

主程序 ZZP

变量说明, 读入并打印原始数据:
N, P, DD, Y0, YEP, WEP,
X2, LM1, LM2。

```

110 FORMAT(2X,3HX2=,3(2X,F15.8))
120 FORMAT(2X,4HLM1=,3(2X,F15.8))
130 FORMAT(2X,4HLM2=,3(2X,F15.8))
    YW1 = 0
    YIT = 0
    WRITE(6,30)AL
30  FORMAT(2X,3HAL=,F8.3)
40  CALL WYD(N,DD,P,Y0,Y,YW2,X2,DW,DY)
    YIT = YIT + 1
    WRITE(6,50)YIT
50  FORMAT(/2X,4HYIT=,I3)
    DO 60 I = 1,N
60  X1(I) = X2(I)
    IF((ABS(Y-Y0)/Y0)-YEP) 64,62,62
62  IF(Y-Y0)64,80,80
64  IF(ABS((YW2-YW1)/YW2)-WEP) 66,90,90
66  WRITE(6,70)
70  FORMAT(2X,9HTERMINATE)
    GOTO 100
80  CALL RDTZ(2,N,X0,X1,X2,LM1,LM2,DW,DY,TI,CCI,Y0,Y,AL)
    GOTO 40
90  CALL RDTZ(1,N,X0,X1,X2,LM1,LM2,DW,DY,TI,CCI,Y0,Y,AL)
    YW1 = YW2
    GOTO 40
100 STOP

```

调用子程序 WYD, 计算重量, 变形及其对设计变量的偏导数。

进行收敛准则的判别, 若不满足则调用子程序 RDTZ, 进行再设计迭代或性能修改, 否则结束

子程序 WYD

计算重量- 形状及其对设计变量的偏导数。

```

END
SUBROUTINE WYD(N,DD,P,Y0,Y,YW2,X,DW,DY)
REAL DW(N),X(N),DY(N)
PI = 3.14159
E = 2.1E6
GAM = 0.0078
YW2 = PI * GAM * (X(1) + X(3)) * (X(2) ** 2 - DD * ** 2) / 4
DW(1) = YW2 / (X(1) + X(3))
DW(2) = 2 * YW2 * X(2) / (X(2) ** 2 - DD * ** 2)
DW(3) = DW(1)
Y = 64 * P * X(3) * ** 2 * (X(1) + X(3)) / 3 / PI / (X(2) ** 4 - DD * ** 4) / E
DY(1) = Y / (X(1) + X(3))
DY(2) = - 4 * Y * X(2) * ** 3 / (X(2) ** 4 - DD * ** 4)
DY(3) = DY(1) * (2 * X(1) + 3 * X(3)) / X(3)
WRITE(6,140)Y,YW2
140  FORMAT(2X,2HY = ,F14.8,5X,4HYW2 = ,F14.8)
WRITE(6,150) (DW(I),I = 1,N)
150  FORMAT(2X,3HDDW = ,5(2X,F18.8))
WRITE(6,160) (DY(I),I = 1,N)
160  FORMAT(2X,3HDY = ,5(2X,F18.8))
RETURN
END
SUBROUTINE RDTZ(Q,N,X0,X1,X2,LM1,LM2,DW,DY,TI,CCI,Y0,Y,AL)
INTEGER CHA,Q,X0(N),TI(N)

```

子程序 RDTZ

```

REAL MU, LUX, I(N), X2(N), LM1(N), LM2(N), DW(N), DY(N), CCI(N)
WRITE(6, 165) Q
165  FORMAT(2X, 2HQ = , I3)
DO 170 I = 1, N
170  X0(I) = I
    CHA = 0
180  WRITE(6, 190) CHA
190  FORMAT(2X, 4HCHA = , I3)
    J = 0
DO 200 I = 1, N
    IF(X0(I).EQ.1) J = J + 1
200  CONTINUE
    WRITE(6, 210) J
210  FORMAT(2X, 2HJ = , I3)
    IF(J.EQ.0) GOTO 460
    B = Y0 - Y
    A = 0
DO 290 I = 1, N
    IF(X0(I) - 1) 250, 220, 250
220  IF(Q - 1) 240, 240, 230
230  A = A + DY(I) * * 2
    GOTO 250
    240  B = B + (1 - AL) * DY(I) * X1(I)
    A = A - (1 - AL) / DW(I) * X1(I) * DY(I) * * 2
250  IF(X0(I) + 1) 270, 260, 270

```

进行再设计迭代和性能修改。根据参数 Q 的
值决定迭代方式。

当 Q = 1 时进行再设计迭代。

当 Q = 2 时进行性能修改。

```

260 B = B - DY(I) * (LM1(I) - X1(I))
270 IF(X0(I) + 2) 290,280,290
280 B = B - DY(I) * (LM2(I) - X1(I))
290 CONTINUE
WRITE(6,300)A,B
300 FORMAT(2X,2HA = ,F20.10,5X,2HB = ,F20.10)
IF(Q.EQ.2) MU = B/A
IF(Q.EQ.1) LU = B/A
IF((Q.EQ.1),AND.(LU.LT.0)) LU = 0
IF(Q - 1) 340,340,310
310 WRITE(6,320) MU
320 FORMAT(2X,3HMU = ,E20.10)
DO 330 I = 1,N
IF(X0(I).EQ.1) X2(I) = X1(I) + MU * DY(I)
330 CONTINUE
GOTO 380
340 WRITE(6,350) LU
350 FORMAT(2X,3HLU = ,E20.10)
DO 360 I = 1,N
IF(X0(I).NE.1) GOTO 360
CI = AL - (1 - AL) * LU * DY(I) / DW(I)
X2(I) = CI * X1(I)
CCI(I) = CI
360 CONTINUE
WRITE(6,370) (CCI(I), I = 1, N)

```

```

370  FORMAT(2X,3HCI = ,5(2X,F14.6))
380  JJ = 0
      DO 410 I = 1,N
      IF(X0(I),NE.1) GOTO 410
      IF((X2(I),GT,LMI(I)).AND.(X2(I),LT,LM2(I))) GOTO 410
      IF(X2(I),LE,LMI(I)) GOTO 390
      IF(X2(I),GE,LM2(I)) GOTO 400
390  JJ = JJ + 1
      X2(I) = LMI(I)
      X0(I) = - 1
      GOTO 410
400  JJ = JJ + 1
      X2(I) = LM2(I)
      X0(I) = - 2
410  CONTINUE
      WRITE(6,415) JJ
415  FORMAT(2X,3HJJ = ,I3)
      WRITE(6,420) (TI(I),I = 1,N)
      WRITE(6,430) (X1(I),I = 1,N)
      WRITE(6,440) (X2(I),I = 1,N)
      WRITE(6,450) (X0(J),J = 1,N)
420  FORMAT(2X,3HTI = ,8(6X,I10))
430  FORMAT(2X,3HX1 = ,8(2X,F14.6))
440  FORMAT(2X,3HX2 = ,8(2X,F14.6))
450  FORMAT(2X,3HX0 = ,8(6X,I10))

```

检查主、被动变量分类有无变化，如有变化则在子程序内部进行迭代，否则返回主程序。

```
IF(JJ,EQ,0) GOTO 460  
CHA = CHA + 1  
GOTO 180  
460 RETURN  
END
```

〔注〕 该程序系哈工大机械教研室研究生朱爱武所编。

二、数学规划法源程序

符号说明

1. 主轴的数学规划法设计主程序 SUMTMP 中全局量:
 - N 设计变量向量 x 的维数
 - MGK 不等式约束条件的个数
 - MFK 目标函数的个数
 - DD 主轴内径 $d(\text{cm})$
 - PP 主轴伸出端所受外力 (kg)
 - DER 伸出端允许挠度
 - EP1 允许收敛精度 ε_1
 - EP2 允许收敛精度 ε_2
 - X(N) 设计变量向量数组
 - X0(N), X3(N), XX(N) 中间工作单元
 - GX(MGK) 不等式约束函数数组
 - FX(MFK) 目标函数数组
 - BL(N) 设计变量的下限值数组
 - BV(N) 设计变量的上限值数组
 - R(MGK) 障碍因子
2. 计算目标函数和约束函数子程序 FGIH 中局部量:
 - E 主轴材料弹性模量 (kg/cm^2)
 - PI 圆周率 π
 - GAM 主轴材料比重 γ (kg/cm^3)
 - CJ 主轴截面惯性矩 (cm^4)
3. 计算广义目标函数子程序 FVNC 中局部量:
 - IFVN 调用广义目标函数的次数
 - SF, SG 中间工作单元
 - Y 广义目标函数值
4. 一维搜索中计算广义目标函数子程序 PENA 中局部量:
 - T0 前次搜索的步长值
 - S(N, N) 搜索方向数组
5. 二次插值法一维搜索子程序 LINE 中局部量:
 - H0 一维搜索的初始步长值
 - HT 一维搜索的步长增量值
 - T1, T2, T3, T4 一维搜索中的步长坐标值
 - Y1, Y2, Y3, Y4 相应步长下的广义目标函数值
 - C1, C2 求一维搜索最佳步长所用的系数

- A 工作单元
- T 最佳搜索步长
- 6. 无约束极小化 Powell 法子程序 MINM 中局部量:
 - SDX 一轮初始点与终点在设计空间中的距离
 - ITE 无约束极小化迭代的轮数
 - DF 相邻两次一维搜索广义目标函数值之差
 - DFM 在一轮无约束极小化过程中相邻两次一维搜索函数之差的最大值
 - F1 前次一维搜索最佳步长所对应的函数值
 - F2 本次一维搜索最佳步长所对应的函数值
 - F3 一轮无约束极小化过程终点的反射点 X^3 所对应的函数值
 - JDFI 与函数值之差最大值相对应的方向下标
- 7. 序列无约束极小化子程序 SUMT 中局部量:
 - C 障碍因子的缩小系数
 - SFM 前一轮无约束极小化初始点的目标函数值
 - SF 一轮无约束极小化初始点的目标函数值
 - ICYC 无约束极小化的序列数
 - F0M 前一轮无约束极小化初始点的广义目标函数值
 - F0 一轮无约束极小化初始点的广义目标函数值
 - Q 经序列无约束极小化处理后目标函数缩小的比例数

主 轴 的 数 学 规 划 法 源 程 序
(用FORTRAN-4语言编写, 在Z80型计算机上执行)

```

PROGRAM SUMTMP
DIMENSION X(10),X0(10),X3(10),XX(10),GX(10),
1 FX(10),R(10),BL(10),BV(10)
COMMON/CHI/DD,PP,DER/CH2/IFVN,EP1,EP2/CH3/S(10,10)
READ(5,300)N,MGK,MFK
300 FORMAT(3I5)
WRITE(5,5)N,MGK,MFK
5 FORMAT(10X,2HN = ,I5,10X,4HMGK = ,I5,10X,4HMFK = ,I5,/)
READ(5,400)DD,PP,DER
400 FORMAT(3F10.5)
WRITE(5,10)DD,PP,DER
10 FORMAT(10X,3HDD = ,F15.6,10X,3HPP = ,F15.6,10X,4HDER = ,F15.6/),
READ(5,500)EP1,EP2
500 FORMAT(2F15.10)
WRITE(5,15)EP1,EP2
15 FORMAT(10X,4HEP1 = ,F15.10,10X,4HEP2 = ,F15.10,/)
CALL DSJ(N,MGK,X,BL,BV,R)
CALL SUMT(N,MGK,MFK,X,X0,X3,XX,GX,FX,BL,BV,R)
STOP
END
SUBROUTINE DSJ(N,MGK,X,BL,BV,R)
DIMENSION X(N),R(MGK),BL(N),BV(N)

```

主 程 序 SUMTMP

变量说明, 读入并打印原始数据:
N,MGK,MFK,DD,PP,DER,EP1,EP2

调用子程序 DSJ
调用子程序 SUMT

子 程 序 DSJ

读入并打印原始数据 X, R, BL, BV

```

READ(5,17) (X(I),I=1,N)
READ(5,17) (R(I),I=1, MGK)
READ(5,17) (BL(I),I=1,N)
READ(5,17) (BV(I),I=1,N)
17  FORMAT(8E15.5)
WRITE(5,20) X
20  FORMAT(/,10X,2HX = ,/ ,6(2X,F15.6),/)
WRITE(5,25) R
25  FORMAT(/,10X,2HR = ,/ ,6(2X,F15.6),/)
WRITE(5,30) BL
30  FORMAT(/,10X,3HBL = ,/ ,6(2X,F15.6),/)
WRITE(5,40) BV
40  FORMAT(/,10X,3HBV = ,/ ,6(2X,F15.6),/)
RETURN
END

SUBROUTINE FGH(N, MGK, MFK, X, GX, FX, BL, BV)
DIMENSION X(N), GX(MGK), FX(MFK), BL(N), BV(N)
COMMON /CH1/DD, PP, DER
E = 2.1E+6
PI = 3.1415926
GAM = 0.0078
CJ = PI*(X(2)**4 - DD**4)/64.
FX(1) = PI*GAM*(X(1) + X(3))*(X(2)*X(2) - DD*DD)/4.
GG(1) = 1. - PP*X(3)*X(3)*(X(1) + X(3))/3. /E/CJ/DER

```

子 程 序 FGH

形成目标函数和约束函数。

```

GX(2) = X(1)/BL(1) - 1.
GX(3) = X(2)/BL(2) - 1.
GX(4) = 1. - X(2)/BV(2)
GX(5) = X(3)/BL(3) - 1.
RETURN
END

```

```

SUBROUTINE FVNC(N, MGK, MFK, X, GX, FX, BL, BV, Y, R, SF, SF0)
DIMENSION X(N), GX(MGK), FX(MFK), BL(N), BV(N), R(N), R(MGK)
COMMON /CHI/DD, PP, DER/CH2/IFVN, EP1, EP2
CALL FGH(N, MGK, MFK, X, GX, FX, BL, BV)
IFVN = IFVN + 1
SF = 0.
SG = 0.
DO 100 I = 1, MFK
SF = SF + FX(I)
DO 110 J = 1, MGK
IF(GX(J).LE.1.E-7) GO TO 110
SG = SG + R(J)/GX(J)
100 CONTINUE
Y = SF + SG
IF(IFVN.EQ.1) SF0 = SF
IF(IFVN - 1) 170, 120, 170
120 WRITE(5, 130) Y, SF, SG
130 FORMAT(/10X, 2HY = , F15.6, 10X, 3HSG = , F15.10, 10X, 3HSG = , F15.10)

```

子 程 序 FVNC

调用子程序 FGH形成广义目标函数。

```

130  WRITE(5,135)R
135  FORMAT(/,10X,2HR = ,/,6(2X,F15.10))
      WRITE(5,140) X
140  FORMAT(/,10X,2HX = ,/,6(2X,F15.10))
      WRITE(5,150) FX
150  FORMAT(/,10X,3HFX = ,F15.10)
      WRITE(5,160) GX
160  FORMAT(/,10X,3HGX = ,F15.10)
170  RETURN
      END

SUBROUTINE PENA(N,MGK,MFK,XX,GX,
1  FX,BL,BV,R,Y,T,T0,IJ,SF,SF0)
  DIMENSION XX(N),GX(MGK),FX(MFK),BL(N),BV(N),R(MGK)
COMMON /CH1/DD,PP,DER/CH2/IFVN,EPI,EP2/CH3/S(10,10)
  T0 = T - T0
  DO 800 K = 1,N
800  XX(K) = XX(K) + T0*S(IJ,K)
  T0 = T
  CALL FVNC(N,MGK,MFK,XX,GX,FX,BL,BV,Y,R,SF,SF0)
  RETURN
  END

SUBROUTINE LINE(N,MGK,MFK,X,XX,GX,
1  FX,BL,BV,R,Y,T,H0,IJ,SF,SF0,Y0)
  DIMENSION X(N),XX(N),GX(MGK),FX(MFK),

```

子 程 序 PENA

——维搜索过程中调用广义目标函数

子 程 序 LINE

```

1  BL(N),BV(N),R(MGK)
COMMON /CH1/DD,PP,DER/CH2/IFVN,EP1,EP2/CH3/S(10,10)
DO 200 I=1,N
XX(I)=X(I)
HT=H0
T2=H0
T0=0
T1=0
Y1=Y0
210 CALL PENA(N,MGK,MFK,XX,GX,FX,BL,BV,R,Y2,T2,T0,IJ,SF,SF0)
DO 220 I=1,MGK
IF(GX(I).GE. 1.E-15)GO TO 220
T2=0.5*T2
GO TO 210
CONTINUE
220 IF(Y2.LT. Y1)GO TO 240
HT=-HT
T3=T1
Y3=Y1
T1=T2
Y1=Y2
T2=T3
Y2=Y3
T3=T2+HT
240 CALL PENA(N,MGK,MFK,XX,GX,FX,BL,BV,R,Y3,T3,T0,IJ,SF,SF0)

```

用二次插值法进行一维搜索，确定最佳
步长因子T

```

DO 250 I = 1, MGK
IF(GX(I).GE.1.E-15)GO TO 250
HT = 0.5*HT
GO TO 240

250 CONTINUE
IF(Y2.LE.Y3) GO TO 260
HT = HT + HT
GO TO 230

260 CONTINUE
IF(ABS(T2 - T1). LT. 1.E-6.OR.ABS(T2 - T3).LT.1.E-6) GO TO 350
C1 = (Y3 - Y1)/(T3 - T1)
C2 = ((Y2 - Y1)/(T2 - T1) - C1)/(T2 - T3)

IF(ABS(C2).LT.1.E-10) GO TO 350
T4 = 0.5*(T1 + T3 - C1/C2)

IF((T4 - T1)*(T3 - T4).LE.1.E-4)GO TO 350
CALL PENA(N, MGK, MFK, XX, GX, FX, BL, BV, R, Y4, T4, T0, IJ, SF, SF0)
IF(ABS(Y2) - 1.) 280, 290, 290

280 A = 1
GO TO 300

290 A = Y2
300 CONTINUE
IF(ABS((Y2 - Y4)/A).GE.EP1)GO TO 310

```

```
IF(Y2.GE.Y4)GO TO 360
GO TO 350
CONTINUE
310 IF((T4 - T2)*HT.LE.0.)GO TO 330
IF(Y2.LT.Y4)GO TO 320
T1 = T2
Y1 = Y2
T2 = T4
Y2 = Y4
GO TO 270
320 T3 = T4
Y3 = Y4
GO TO 270
CONTINUE
330 IF(Y2.LT.Y4) GO TO 340
T3 = T2
Y3 = Y2
T2 = T4
Y2 = Y4
GO TO 270
340 T1 = T4
Y1 = Y4
GO TO 270
350 T = T2
Y = Y2
```

```

. 234 .
      GO TO 370
360  T = T4
      Y = Y4
370  WRITE(5,380) T,Y
380  FORMAT(10X,10H*****10X,2HT = ,E15.8,10X,2HY = ,E15.8)
      RETURN
      END

```

输出相应的最佳步长 T 和对应的广义目标函数值。

```

SUBROUTINE MINM(N,MGK,MFK,X,X0,X3,XX,
1  GX,FX,BL,BV,R,F0,T,H0,SF,SF0,ITE)
DIMENSION X(N),X0(N),X3(N),XX(N),GX(MGK),
1  FX(MFK),BL(N),BV(N),R(MGK)
COMMON /CH1/DD,PP,DER/CH2/IFVN,EP1,EP2/CH3/S(10,10)
SDX = 1.E+6
900  CONTINUE
      IF(SDX.LT.EP1) GOTO 1030
      ITE = ITE + 1
      Y0 = F0
      F1 = F0
      DFM = 0.
      JDFI = 1
      DO 920 I = 1,N
      IJ = I
      CALL LINE(N,MGK,MFK,X,XX,GX,FX,BL,BV,R,F2,T,H0,IJ,SF,SF0,Y0)
      DO 910 J = 1,N

```

子程序 MINM

用 Powell 法进行无约束极小化。

```

910 X(J) = X(J) + T*S(I,J)
    DF = F1 - F2
    Y0 = F2
    F1 = F2
    IF(DF.LE.DFM) GOTO 920
    DFM = DF
    JDFI = 1
920 CONTINUE
    DO 930 I = 1, N
    X3(I) = 2.*X(I) - X0(I)
    S(N+1,I) = X(I) - X0(I)
930 CONTINUE
    CALL FVNC(N,MGK,MFK,X3,GX,FX,BL,BV,F3,R,SF,SF0)
    SDX = 0
    DO 940 I = 1, N
    SDX = SDX + (X(I) - X0(I))**2
    SDX = SQRT(SDX)
    WRITE(5,945) SDX
945 FORMAT(10X,12H***SDX*** ; ,F15.10,/)
    IF(F0 - 2.*F2 + F3.GT.2.*DFM) GO TO 980
    DO 955 I = JDFI, N
    DO 950 J = 1, N
    S(I,J) = S(I+1,J)
950 CONTINUE
955 Y0 = F2

```



```

CALL LINE(N, MGK, MFK, X, XX, GX, FX, BL, BV, R, F0, T, H0, IJ, SF, SF0, Y0)
WRITE(5, 960)
960  FORMAT(/, 10X, 10HAAAAAAAAAAAA)
DO 970 K = 1, N
X0(K) = X(K) + T*S(N, K)
970  CONTINUE
GO TO 900
980  CONTINUE
IF(F3, GE, F2) GO TO 1000
985  F0 = F3
DO 990 I = 1, N
X0(I) = X3(I)
X(I) = X3(I)
990  CONTINUE
GO TO 1020
1000 DO 1010 K = 1, N
1010 X0(K) = X(K)
F0 = F2
1020 CONTINUE
GO TO 900
1030 RETURN
END

```

SUBROUTINE SUMT(N, MGK, MFK, X, X0, X3, XX, GX, FX, BL, BV, R)
DIMENSION X(N), X0(N), X3(N), XX(N), GX(MGK),

```

1 FX(MFK),BL(N),BV(N),R(MGK)
COMMON/CH1/DD,PP,DER/CH2/IFVN,EP1,EP2/CH3/S(10,10)
C=0.2
H0=0.01
SFM=1.E+6
F0M=1.E+6
IFVN=0
ICYC=0
ITE=0
DO 5 I=1,10
DO 6 J=1,10
6 S(I,J)=0
5 CONTINUE
DO 500 K=1,MGK
500 R(K)=R(K)/C
DO 505 I=1,N
505 X0(I)=X(I)
DO 515 I=1,N
DO 510 J=1,N
IF(I.EQ.J) S(I,J)=1.
IF(I.NE.J) S(I,J)=0
510 CONTINUE
515 CONTINUE
520 ICYC=ICYC+1
DO 525 I=1,MGK

```

用制约函数内点法求解约束优化问题的
过程

```

525 R(I) = C*R(I)
      WRITE(5,530) ICYC
530  FORMAT(/,10X,5HICYC = ,3X,I5)
      WRITE(5,540) R
540  FORMAT(/,10X,2HR = ,/6(2X,E15.8))
550  CALL FVNC(N,MGK,MFK,X,GX,FX,BL,BV,F0,R,SF,SF0)
      CALL MINM(N,MGK,MFK,X,X0,X3,XX,GS,FX,
1 BL,BV,R,F0,T,H0,SF,SF0,ITE)
      WRITE(5,560) ITE,F0,SF
560  FORMAT(/,10X,4HITE = ,I10,10X,3HF0 = ,F20.10,10X,3HSF = ,F20.10)
      DO 580 I=1,N
          XK = X(I)
          KI = I
          WRITE(5,570) KI,XK
570  FORMAT(/,20X,2HX(,I2,2H) = ,F20. 10)
580  CONTINUE
      DO 590 I = 1, MGK
          IF(GX(I),LT.1.E-15) GO TO 680
590  CONTINUE
          IF(ABS((SFM - SF)/SF),LT.EP2.AND.
1 ABS((F0M - F0)/F0),LE.EP1) GO TO 650
          SFM = SF
          F0M = F0
          GO TO 520
650  WRITE(5,600)

```

收 敛 准 则

```

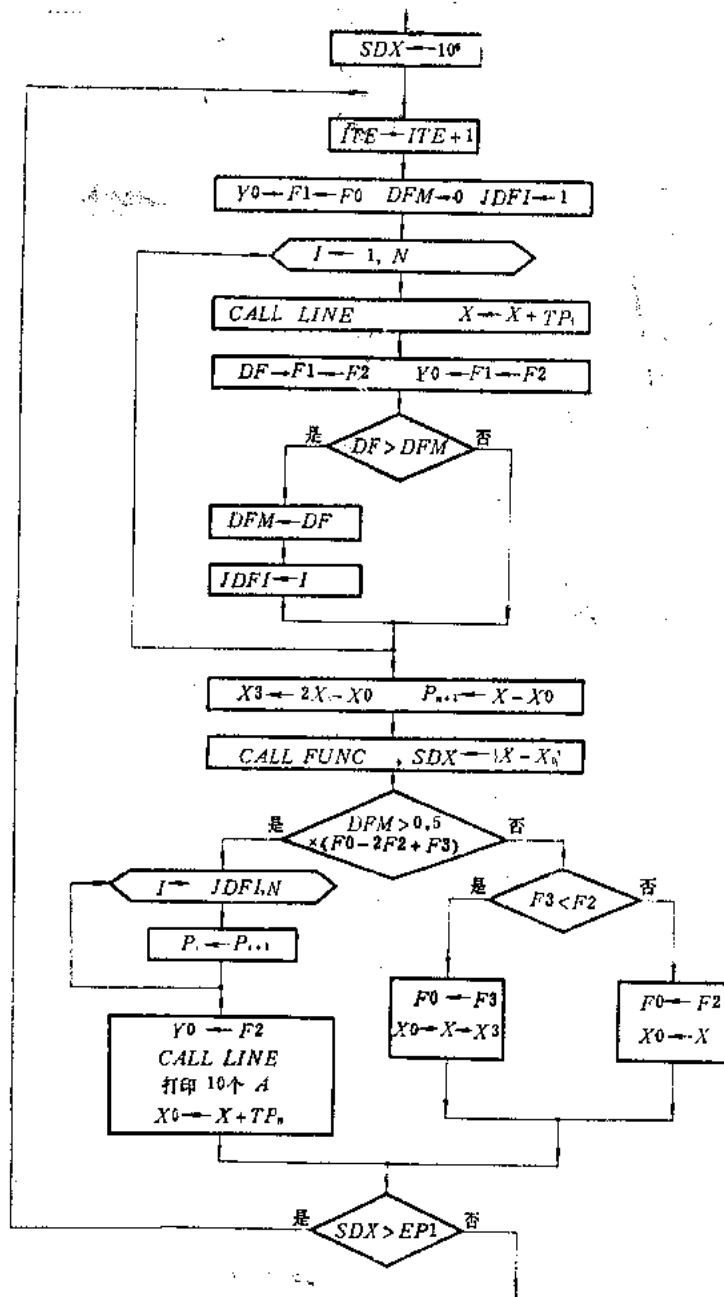
600  FORMAT(/,20HEEEEEEEEEEEEEEEEEEE)
680  WRITE(5,690)
690  FORMAT(/,20HGGGGGGGGGGGGGGGGGGGG)
    Q = (SF0 - SF)/SF0
    WRITE(5,700) ICYC
700  FORMAT(/,10X,6HCYCLE = ,I5)
    WRITE(5,705) IFVN,F0,SF
705  FORMAT(/,12X,4HFVN = ,15,/,13X,3HF0 = ,E20.10,/,13X,3HSF = ,E20.10)
    WRITE(5,710) R
710  FORMAT(/,10X,2HR = ,/,6(2X,E15.8))
    WRITE(5,740) Q,SF0,SF
    WRITE(5,715)
715  FOR MAT (11X,14HOPT1--VARIABLE;)
    DO 730 I = 1,N
    XK = X(I)
    KI = I
    WRITE(5,720) KI,XK
720  FORMAT(/,31X,2HX(,12,2H) = ,E20.10)
730  CONTINUE
740  FORMAT(/,12X,2HQ = ,E20.10,/,10X,4HSF0 = ,E20.10,/,
1  11X,3HSF = ,E20.10,/)
    WRITE(5,750) FX
750  FORMAT(/,10X,16HOBJECT FUNCTION: ,/,34X,3HFX = ,E20.10)
    WRITE(5,760)
760  FORMAT(/,15X,11HSUBJECT TO: ,/)

```

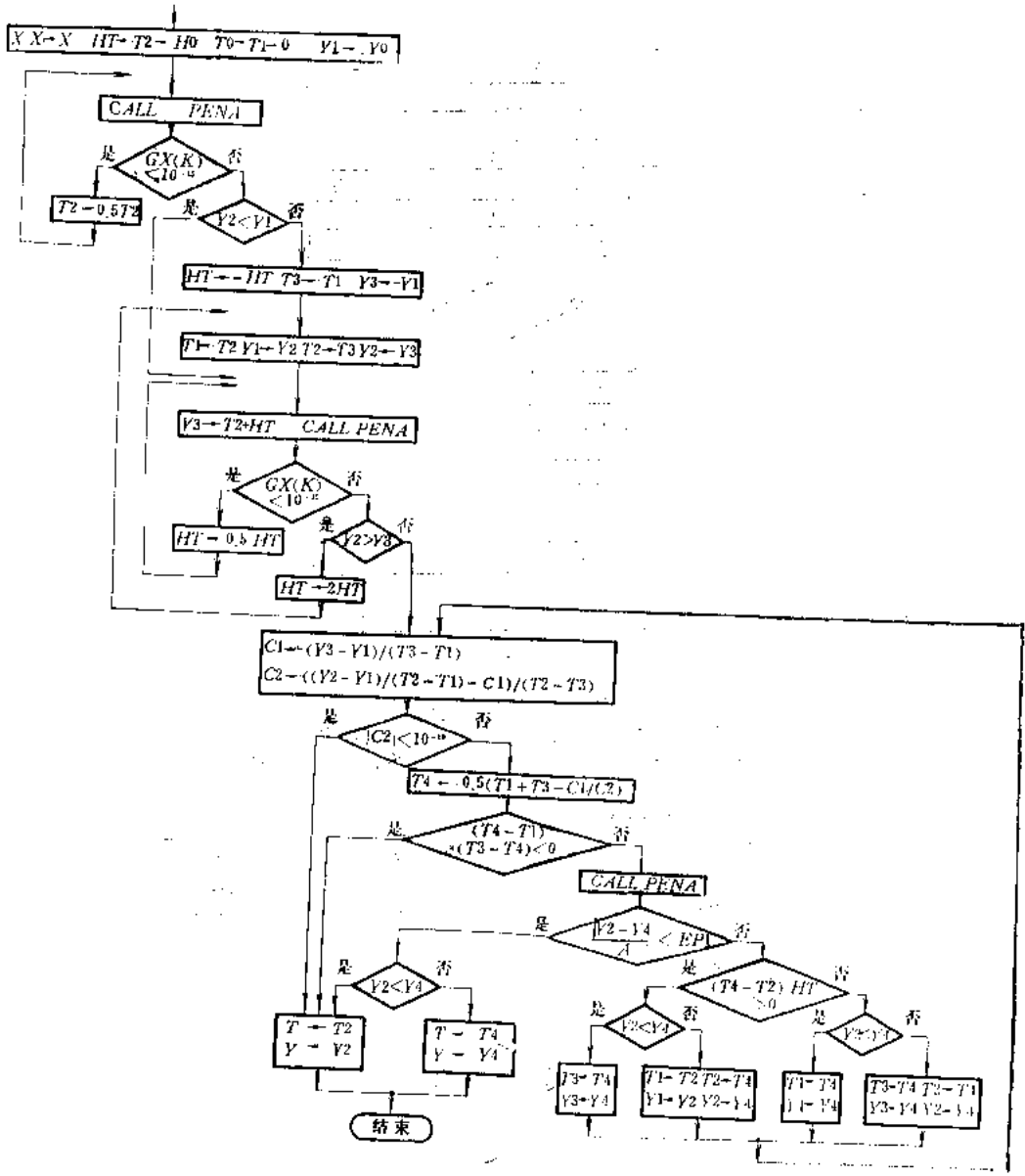
最后结果的输出。

```
DO 780 I = 1, MGK
  KX = 1
  XG = GX(I)
  WRITE(5, 770) KX, XG
770  FORMAT(30X, 3HGX(, I2, 2H) = , E20.10, />
780  CONTINUE
      RETURN
      END
```

[注] 该程序系哈工大机床教研室研究生曹宏毅所编。



附图 1



附图 2

三、有限元——优化设计源程序

符号说明

1. 主轴的有限元——优化准则法设计主程序 UU 中全局量:

- NEL 结构单元总数
- NE 单元序号
- NOD 结点总数
- ND 结点自由度总数
- NZ 具有约束的自由度总数
- NF 具有载荷作用的结点总数
- NM 设计变量(优化参数)总数
- NT 外径类优化参数的个数
- YIT 迭代次序号
- PI 圆周率 π
- E 材料弹性模量 (kg/cm^2)
- IZ 截面惯性矩
- RO 材料比重 (kg/cm^3)
- ROI 材料密度 ($\text{kg} \cdot \text{sec}^2/\text{cm}^4$)
- XD 主轴内径
- FX1 再设计迭代前的结构重量
- FX2 再设计迭代后的结构重量
- NY 静态变形约束个数
- NH 等式约束的个数
- NG 等式和不等式约束的总数
- U 结构尺寸比例实际值
- U0 尺寸比例给定值
- YR 主轴伸出端挠度
- YR0 伸出端挠度最大允许值
- PR 主轴一阶自振频率
- PR0 一阶自振频率最小允许值
- FEP 主轴重量再设计迭代的收敛精度 ε_r
- GEP 约束函数值的允许相对偏差 ε_g
- AL 再设计迭代松弛因子 α
- DD(NEL) 单元外径数组
- DL(NEL) 单元跨长
- DOI(NEL) 单元左结点序号
- DOJ(NEL) 单元右结点序号

NOZ(NZ) 具有约束结点的自由度序号
 NONZ(NZ) 对应各自由度的约束值
 NOF(NF) 具有载荷作用的结点序号
 NY(NF) 对应各结点的外力值
 ZM(NF) 对应各结点的外力矩值
 X0(NM) 标志优化参数为主、被动参数的信息数组
 X1(NM) 迭代前的优化参数值
 X2(NM) 迭代后的优化参数值
 LM1(NM) 优化参数的下限值
 LM2(NM) 优化参数的上限值
 W(NM,NEL) 优化参数与结构参数对应关系的信息矩阵
 TS(ND,ND) 结构总刚阵
 TM(ND,ND) 存放总刚阵的工作单元
 TSD(ND,ND) 总刚阵对外径类设计变量的偏导数
 TSL(ND,ND) 总刚阵对跨长类设计变量的偏导数
 TSA(ND,ND) 结构总质阵
 TAD(ND,ND) 总质阵对外径类设计变量的偏导数
 TAL(ND,ND) 总质阵对跨长类设计变量的偏导数
 Y(ND) 各自由度的静态变形值
 Y1(ND) 结构的一阶法式化振型
 DU(NM) 尺寸比例对设计变量的偏导数
 DP(NM) 一阶固有频率对设计变量的偏导数
 DY(NM) 主轴端部挠度对设计变量的偏导数
 DYY(ND,NM) 各自由度的静态变形对设计变量的偏导数矩阵
 DFX(NM) 结构重量对设计变量的偏导数
 GX(NG) 约束函数数组

2. 优化准则法设计子程序 CRITI
3. 原始数据的输入与打印子程序 WRIT
4. 计算目标函数、约束函数及其偏导数子程序 FHG
5. 计算目标函数、等式约束函数及目标函数对设计变量的偏导数子程序 WD
6. 计算结构静、动态性能子程序 STDY 中局部量:
 - F(ND) 各自由度上的外载荷(力或力矩)值
 - FF(ND) 自重转移到单元结点各自由度上的等效载荷值
 - FFD(ND) 载荷对外径类优化参数的偏导数
 - FFL(ND) 载荷对跨长类优化参数的偏导数
 - DIS(ND) 规一化振型
 - SM(4,4) 单元刚阵
 - SD(4,4) 单元刚阵对外径类优化参数的偏导数

- SL(4,4) 单元刚阵对跨长类优化参数的偏导数
 SMA(4,4) 单元质阵
 SAD(4,4) 单元质阵对外径类优化参数的偏导数
 SAL(4,4) 单元质阵对跨长类优化参数的偏导数
 BOOL 逻辑变量 BOOL=TRUE 表示振动方程有解
 PA 反迭代后的一阶固有圆频率
 PB 反迭代前的一阶固有圆频率
 XL 一阶振型的最大幅值
7. 建立单元刚、质阵子程序 SMNE 中局部量:
- L 单元跨长
 D 单元外径
 D1 单元内径
 D11 单元内径的平方
 D2 单元外径的平方
 A 单元截面积
 DA 单元截面积对外径的导数
 DI 截面惯性矩对外径的导数
 Q 单元单位长度的自重
 M 单元单位长度的质量
 DQ 单位长度自重对外径的导数
 DM 单位长度质量对外径的导数
8. 组集结构总体刚、质阵子程序 TSME 中局部量:
- S 单元矩阵中所论结点的行号
 T 单元矩阵中所论结点的列号
 K 总体矩阵中所论结点的行号
 P 总体矩阵中所论结点的列号
 SI 单元矩阵中所论结点相邻上一行的行号
 SJ 单元矩阵中所论结点相邻前一列的列号
 KI 总体矩阵中所论结点相邻上一行的行号
 KJ 总体矩阵中所论结点相邻前一列的列号
 SIM 单元矩阵行号
 SJN 单元矩阵列号
 KIM 总体矩阵行号
 KJN 总体矩阵列号
 A(KIM,KJN) 总体矩阵
 B(SIM,SJN) 单元矩阵
9. 边界条件处理子程序 BCTR 中局部量:
- S(ND) 存放边界约束值的工作单元

- T(ND,ND) 存放被处理矩阵的工作单元
10. 一阶自振频率求导子程序 DERP, DRP 中局部量:
T1(ND,ND) 存放质阵求导的矩阵工作单元
 11. 静变形求导子程序 DERY, DRY
 12. 求解线性方程组的高斯——约当消去法子程序 GJ
 13. 行、列矩阵乘积、方阵与列阵乘积子程序 MATMU1, MATMU2
 14. 求解拉格朗日乘子 λ 和性能修改系数 β 子程序 LUBA 中局部量:
A(0:2, 0:3) 线性方程组增广矩阵
B(0:2) 方程组右端项, 存放 λ 或 β
T(0:3) 工作单元
 15. 再设计迭代和性能修改子程序 RDTZ 中局部量:
CHA 确定优化参数的主、被动性质所作迭代的序号
Q 参量, 当 $Q=1$ 时作再设计迭代; 当 $Q=2$ 时作性能修改

主 轴 的 有 限 元 —— 优 化 准 则 法 源 程 序
 (用FORTRAN—77语言编写,在UNIVAC 1100型计算机上执行)

```

PROGRAM UU
COMMON/C1/PI,E,I,Z,RO,RO1,C2/NEL,NOD,ND,NZ,NF/C3/NM,NT,YIT/C4/
1 XD/C5/FX1,FX2/C6/NY,NH,NG/C7/FEP,GEP
INTEGER DOI(40),DOJ(40),NOZ(20),NOF(20),W(40,40),X0(40)
REAL X1(40),X2(40),LM1(40),LM2(40),DD(40),DL(40),NONZ(40),FY(20),
1 ZM(20),TS(80,80),TM(80,80),TSD(80,80),TSL(80,80),TSA(80,80),TAL
2 (80,80),Y(80),Y1(80),DFX(40),DYY(70,40),DP(40),DU(40)
3 DY(40),TAD(80,80),GEP,GX(10)
READ*,NM,NT,NH,NG,NY,GEP,FEP,AL,PR0,U0,YR0
READ*,NEL,NOD,NZ,NF,XD
WRITE(6,0010) NM,NT,NH,NG,NY,GEP,FEP,AL,PR0,U0,YR0
WRITE(6,0020) NEL,NOD,NZ,NF,XD
0010 FORMAT(5X,'NM=',I3/5X,'NT=',I3/5X,'NH=',I3/5X,'NG=',I3/5X,'NY=',
1 ,I3/5X,'GEP=',F9.4/5X,'FEP=',F9.4/5X,'AL=',F9.4/5X,'PR0=',F9.4
2 ,/5X,'U0=',F9.4/5X,'YR0=',F9.4)
0020 FORMAT(5X,'NEL=',I3/5X,'NOD=',I3/5X,'NZ=',I4/5X,'NF=',I4/5X,
1 'XD=',F9.4)
PI=3.1415926
RO=0.0078
E=2.1E6
RO1=RO/980
ND=2*NOD
    
```

主 程 序 UU

进行有限元——优化准则法设计
 变量说明
 读入并打印原始数据

```

CALL WRITE(DOI,DOJ,DD,DL,NOZ,NOZ,NOF,FY,ZM,LM1,LM2,W)
CALL CRITI(DOI,DOJ,NOZ,NOZ,NOF,FY,ZM,DD,DL,LM,TS,TSD,
1 TSL,TSA,TAD,TAL,Y,Y1,PR,U,GX,YR0,U0,PR0,DYY,DP,DU,DY,DXF,
2 W,LM1,LM2,AL,X0,X1,X2)
STOP
END
SUBROUTINE CRITI(DOI,DOJ,NOZ,NOZ,NOF,FY,ZM,DD,DL,TM,TS,TSD,
1 TSL,TSA,TAD,TAL,Y,Y1,PR,U,GX,YR0,U0,PR0,DYY,DP,DU,DY,DFX,
2 W,LM1,LM2,AL,X0,X1,X2)
COMMON/C1/PI,E,IZ,RO,RO1/C2/NEL,NOD,ND,NZ,NF/C3/NM,NT,
1 YIT/C4/XD/C5/FX1,FX2/C6/NY,NH,NG/C7/FEP,GEF
INTEGER YIT,DOI(NEL),DOJ(NEL),NOZ(NZ),NOF(NF),W(NM,NEL),
X0(NM)
REAL NONZ(NZ),FY(NF),ZM(NF),DD(NEL),DL(NEL),TM(ND)
1 ND),TS(ND,ND),TSD(ND,ND),TSL(ND,ND),TSA(ND,ND),TAD(ND,ND),
2 TAL(ND,ND),Y(ND),Y1(ND),GX(NG),YR0,U0,DYY(ND,NM),DP(NM),
3 DY(NM),DU(NM),DFX(NM),X2(NM),LM1(NM),LM2(NM),X1(NM),
YIT = 0
WRITE(6,0030) YIT
0030 FORMAT(//,5X,'INTERMEDIATE SOLUTION'//5X,'YIT = ',I5)
0040 CALL FHG(DOI,DOJ,NOZ,NOZ,NOF,FY,ZM,DD,DL,TM,TS,TSD,TSL,TSA,
1 TAD,TAL,Y,Y1,PR,U,GX,YR0,U0,PR0,DYY,DP,DU,DY,DFX,X2,W)
YR = Y(ND - 1)
WRITE(6,0045)FX1,FX2

```

调用子程序WRITE及CRITI
结束

子 程 序CRITI

进行优化准则设计

变量说明

调用子程序FHG,计算目标函数,约束函数及其对设计变量的偏导数。

```

0045 FORMAT(5X,'FX1 = ',F20.10,5X,'FX2 = ',F20.10)
WRITE(6,0060) (GX(I), I = 1,NG)
0060 FORMAT(2X,'GX = ',/8(2X,E14.6))
DO 0082 I = 1,NM
0082 X1(I) = X2(I)
YIT = YIT + 1
WRITE(6,0084) YIT
0084 FORMAT(///5X,'YIT = ',I5)
IF(YIT.EQ.1)GOTO 0080
DO 0095 I = 1,NH
IF(ABS(GX(I)),LE.GEP) THEN
GOTO 0095
ELSE
GOTO 1111
END IF
0095 CONTINUE
DO 0110 I = NH + 1,NG
IF(GX(I).LE.GEP) THEN
GOTO 0110
ELSE
GOTO 1111
END IF
0110 CONTINUE
IF((ABS(FX2 - FX1)/FX2),LT.FEP) THEN
WRITE(6,0065) (X2(I), I = 1,NM)

```

收敛精度检验，若满足则返回到主程序，
否则调用子程序RDTZ，进行再设计迭代
或性能修改。

```

0065 FORMAT(//,5X,'FINAL SOLUTION'//,5X,'XMIN = ',/,10(1X,E12.6))
WRITE(6,0068)FX2
WRITE(6,0072)PR
WRITE(6,0074)U
WRITE(6,0075)(Y(I),I=1,ND)
WRITE(6,0076)(Y1(I),I=1,ND)
0068 FORMAT(5X,'FMIN = ',F14.6)
0074 FORMAT(5X,'U(XMIN) = ',F14.6)
0072 FORMAT(5X,'PR(XMIN) = '//,F14.6)
0075 FORMAT(5X,'Y(XMIN) = ',/,10(1X,E12.6))
0076 FORMAT(5X,'Y1(XIN) = ',/,10(1X,E12.6))
GOTO 0120
END IF
0080 PRINT *,'REDESIGN'
CALL RDTZ(1,X0,X1,X2,LM1,LM2,DFX,DP,DU,DY,U,YR,PR,U0,YR0,
1 PR0,AL)
FX1 = FX2
GOTO 0040
1111 PRINT *,'RATZ'
CALL RDTZ(2,X0,X1,X2,LM1,LM2,DFX,DP,DU,DY,U,YR,PR,U0,YR0,
1 PR0,AL)
GOTO 0040
0120 RETURN
END

```

```

SUBROUTINE WRIT(DOI,DOJ,DD,DL,NOZ,NOZ,NONZ,NOF,FY,ZM,LM1,LM2,W)
COMMON /C2/NEL,NOD,ND,NZ,NF/C3/NM
INTEGER DOI(NEL),DOJ(NEL),NOZ(NZ),NOF(NF),W(NM,NEL)
REAL NONZ(NZ),FY(NF),ZM(NF),DD(NEL),DL(NEL),LM1(NM),LM2(NM)
READ*,DOI,DOJ,NOZ,NOZ,NONZ,NOF,FY,ZM,DD,DL
READ*,LM1,LM2,W
WRITE(6,0140) DOI
WRITE(6,0150) DOJ
WRITE(6,0160) NOZ
WRITE(6,0170) NONZ
WRITE(6,0180) NOF
WRITE(6,0190) FY
WRITE(6,0200) ZM
WRITE(6,0210) DD
WRITE(6,0220) DL
WRITE(6,0230) LM1
WRITE(6,0240) LM2
WRITE(6,0250)W

0140 FORMAT(5X,'DOI = ',/,10(18,5X))
0150 FORMAT(5X,'DOJ = ',/,10(18,5X))
0160 FORMAT(5X,'NOZ = ',/,10(18,5X))
0170 FORMAT(5X,'NONZ = ',/,10(1X,E12.6))
0180 FORMAT(5X,'NOF = ',/,10(18,5X))
0190 FORMAT(5X,'FY = ',/,10(1X,E12.6))
0200 FORMAT(5X,'ZM = ',/,10(1X,E12.6))

```

子程序 WRIT

读入并打印原始数据


```

0210 FORMAT(5X,'DD = ',/,10(1X,E12.6))
0220 FORMAT(5X,'DL = ',/,10(1X,E12.6))
0230 FORMAT(5X,'LMI = ',/,10(1X,E12.6))
0240 FORMAT(5X,'LM2 = ',/,10(1X,E12.6))
0250 FORMAT(5X,'W = ',/,10(18,5X))

```

```

RETURN
END

```

```

SUBROUTINE FHG(DOI,DOJ,NOZ,NOZ,NOZ,NOF,FY,ZM,DD,DL,DM,TS,TSD,TSL,
1 TSA,TAD,TAL,Y,Y1,PR,U,GX,YR0,U0,PR0,DYY,DP,DU,DY,DFX,X2,W)
COMMON/C1/PI,E,IZ,RO,RO1/C2/NEL,NOD,ND,NZ,NF/C3/NM,NI,
1 YIT/C4/XD/C5/FX1,FX2/C6/NY,NH,NG
INTEGER DOI(NEL),DOJ(NEL),NOZ(NZ),NOF(NF)W(NM,NEL)
REAL NONZ(NZ),FY(NF),ZM(NF),DD(NEL),DL(NEL),TM(ND,ND),
1 TS(ND,ND),TSD(ND,ND),TSL(ND,ND),TSA(ND,ND),TAD(ND,ND),
1 TAL(ND,ND),Y(ND),Y1(ND),DP(NM),DU(NM),DYY(ND,NM),DY(NM),
2 DFX(NM),X2(NM),F(80),FF(80),FFD(80),DIS(80),GX(NG)
CALL WD(DD,DL,U,DFX,X2,W)
CALL STDY(DOI,DOJ,NOZ,NOZ,NOZ,NOF,FY,ZM,DD,DL,DM,TS,TSD,TSL,TSA,
1 TAD,TAL,F,FF,FFD,FEL,Y,Y1,PR,DIS)
CALL DERY(2,DOI,DOJ,NOZ,NOZ,DD,DL,DM,TS,TSD,TSL,FF,FFD,FFL,
1 Y,DYY,DY,W)
CALL DERP(2,DOI,DOJ,NOZ,NOZ,DD,DL,FF,FFD,FFL,TSD,TSL,TAD,
1 TAL,Y1,PR,DP,W)
DO 0010 I=1,NM

```

程 子 序 FHG

计算目标函数, 约束函数及其对设计变
量的偏导数

```

0010 DU(1) = 0.
      DU(NM-1) = 1./DL(NEL)
      DU(NM) = DL(NEL-1)/DL(NEL)/DL(NEL)
      GX(1) = U/U0-1.
      GX(2) = 1-PR/PR0
      GX(3) = Y(ND-1)/YR0-1,
      RETURN
      END

SUBROUTINE WD(DD,DL,U,DEX,X2,W)
COMMON/C1/PI,E,I,Z,R0/C2/NEL,NOD/C3/NM,NT,YIT/C4/XD/C5/FX1,
1 FX2/C6/NY,NH
INTEGER YIT,W(NM,NEL)
REAL DD(NEL),DL(NEL),DFX(NM),X2(NM)
IF(YIT.EQ.0) THEN
DO 0260 I=1,NM
DO 0260 NE=1,NEL
IF((W(I,NE).EQ.1).AND.(I.LE.NT)) X2(I) = DD(NE)
IF((W(I,NE).EQ.1).AND.(I.GT.NT)) X2(I) = DL(NE)
0260 CONTINUE
WRITE(6,0270) X2
0270 FORMAT(5X,'2X = ',/,10(1X,E12.6))
FX1 = 0.
ELSE
DO 0280 I=1,NM
DO 0280 NE=1,NEL
IF((W(I,NE).EQ.1).AND.(I.LE.NT)) DD(NE) = X2(I)

```

子程序 WD

计算目标函数及其对设计变量的偏导数。
计算等式约束函数值。

```

      IF((W(I,NE).EQ.1).AND.(I.GT.NT))DL(NE) = X2(I)
0280  CONTINUE
      END IF
      FX2 = 0.
      DO 0290 NE = 1,NEL
0290  FX2 = FX2 + PI * RO * (DD(NE) * DD(NE) - XD * XD) * DL(NE) / 4.
      DO 0300 I = 1,NM
      DFX(I) = 0.
      DO 0300 NE = 1,NEL
      IF((W(I,NE).EQ.1).AND.(I.LE.NT))DFX(I) = DFX(I) + PI * RO * DD(NE) *
1 DL(NE) / 2.
      IF((W(I,NE).EQ.1).AND.(I.GT.NT)) DFX(I) = DFX(I) + PI * RO * (DD (NE)
1 * DD(NE) - XD * XD) / 4.
0300  CONTINUE
      U = DL(NEL - 1) / DL(NEL)
      WRITE(6,0310) U
0310  FORMAT(5X,'U = ',5(2X,F15.7))
      RETURN
      END

      SUBROUTINE STDY(DOI,DOJ,NOZ,NOZ,NOZ,NOF,FY,ZM,DD,DL,TM,TS,TSD,
1 TSL,TSA,TAD,TAL,F,FF,FFD,FFL,Y,Y1,PR,DIS)
      COMMON /C1/PI/C2/NEL,NOD,ND,NZ,NF
      INTEGER DOI(NEL),DOJ(NEL),NOZ(NZ),NOF(NF),IK(80),JK(80),
1 INDEX(80)

```

子程序 STDY

用有限元法计算结构的静态变形

```

REAL NONZ(NZ),FY(NF),ZM(NF),DD(NEL),DL(NEL),TM(ND,ND)),
1 TS(ND,ND),TSD(ND,ND),TSL(ND,ND),TSA(ND,ND),TAD(ND,ND),
2 TAL(ND,ND),F(ND),FF(ND),FFD(ND),FFL(ND),Y(ND),YI(ND),
8 DIS(ND),SM(4,4),SD(4,4),SL(4,4),SMA(4,4),SAD(4,4),SAL(4,4),
LOGICAL BOOL
DO 0320 I=1,ND
F(I)=0.
FF(I)=0.
FFD(I)=0.
FFL(I)=0.
DO 0320 J=1,ND
TS(I,J)=0.
TSD(I,J)=0.
TSL(I,J)=0.
TSA(I,J)=0.
TAD(I,J)=0.
0320 TAL(I,J)=0.
DO 0330 NE=1,NEL
CALL SMNE(1,NE,DOI,DOJ,DD,DL,SM,SD,SL,SMA,SAD,SAL,FF,FFD,FFL)
CALL TSME(NE,DOI,DOJ,SM,TS)
CALL TSME(NE,DOI,DOJ,SMA,TSA)
0330 CONTINUE
DO 0340 I=1,NF
K=NOF(I)
F(2*K-1)=FY(I)

```

```

0340 F(2*K) = ZM(I)
      DO 0350 K = 1, NOD
        F(2*K-1) = F(2*K-1) + FF(2*K-1)
0350 F(2*K) = F(2*K) + FF(2*K)
      CALL BCTR(0, NOZ, NONZ, INDEX, TS, F)
      DO 0360 I = 1, ND
        IK(I) = 0
        JK(I) = 0
      DO 0360 J = 1, ND
0360 TM(I, J) = TS(I, J)
        BOOL = .TRUE.
        CALL GJ(ND, TS, F, I, E-7, IK, JK, BOOL)
        IF(.NOT. BOOL) STOP 'FAIL.'
      DO 0370 I = 1, ND
0370 Y(I) = F(I)
        WRITE(6, 0380) Y
0380 FORMAT(5X, 'Y = ', /, 10(1X, E12.6))
        N = 0
        PA = 0.
0390 PB = PA
        XL = 0.
      DO 0400 I = 1, ND
        IF(ABS(F(I)).GE.ABS(XL)) XL = F(I)
0400 CONTINUE
      DO 0410 J = 1, ND

```

用反迭代法计算结构的一阶泛式化振型
及一阶固有频率

```

0410 DIS(I) = F(I)/XL
      CALL MATMU2(ND, TSA, DIS, F)
      DO 0420 I = 1, ND
      IF(INDEX(I).EQ.0) F(I) = 0.
0420 FF(I) = F(I)
      DO 0430 I = 1, ND
      IK(I) = 0
      JK(I) = 0
      DO 0430 J = 1, ND
      TS(I, J) = TM(I, J)
0430 CALL GJ(ND, TS, F, 1, E - 7, IK, JK, BOOL)
      IF(.NOT.BOOL) STOP'FAIL2'
      DO 0440 I = 1, ND
0440 DIS(I) = F(I)
      CALL MATMU2(ND, TSA, DIS, Y1)
      CALL MATMU1(ND, DIS, Y1, AF)
      CALL MATMU1(ND, DIS, FF, AZ)
      PA = AZ/AF
      AF = SQRT(AF)
      DO 0450 I = 1, ND
      Y1(I) = DIS(I)/AF
0450 F(I) = Y1(I)
      IF(ABS((PA - PB)/PA).GE.1.E - 4) THEN
      N = N + 1
      GOTO 0390

```

```
END IF
PR = 0.159155 * SQRT(PA)
WRITE(6,0460)PR
0460 FORMAT(5X,'PR= ',F15.7)
RETURN
END
```

```
SUBROUTINE SMNE(NA,NE,DOI,DOJ,DD,DL,SM,SD,SL,SMA,SAD,SAL,FF,
1 FFD,FFL)
COMMON/C1/PI,E,IZ,RO,ROI/C2/NEL,NOD,ND/C4/XD
INTEGER DOI(NEL),DOJ(NEL)
REAL DD(NEL),DL(NEL),SM(4,4),SD(4,4),SL(4,4),SMA(4,4),SAD(4,4),
1 SAL(4,4),FF(ND),FFD(ND),FFL(ND),L,M,IZ
L = DL(NE)
D = DD(NE)
D1 = XD
D11 = D1 * D1
D2 = D * D
A = PI * (D2 - D11) / 4.
DA = PI * D / 2.
IZ = PI * (D2 * D2 - D11 * D11) / 64.
DI = PI * D * D2 / 16.
EI = E * IZ
Q = RO * A
DQ = RO * DA
```

子 程 序 SMNE

计算单元附、质阵各元素值
计算自重转移到单元结点各自由度上
的等效荷载值

```

M = RO1 * A
DM = RO1 * DA
SM(1,1) = 12./L/L/L
SM(2,1) = 6./L/L
SM(2,2) = 4./L
SM(3,1) = -SM(1,1)
SM(3,2) = -SM(2,1)
SM(3,3) = SM(1,1)
SM(4,1) = SM(2,1)
SM(4,2) = 2./L
SM(4,3) = SM(3,2)
SM(4,4) = SM(2,2)
SMA(1,1) = 13. * L/35.
SMA(2,1) = 11. * L * L/210.
SMA(2,2) = L * L * L/105.
SMA(3,1) = 9. * L/70.
SMA(3,2) = 13. * L * L/420.
SMA(3,3) = SMA(1,1)
SMA(4,1) = -SMA(3,2)
SMA(4,2) = -L * L * L/140.
SMA(4,3) = -SMA(2,1)
SMA(4,4) = SMA(2,2)
DO 0470 I = 1,4
DO 0470 J = 1,I
SM(I,J) = EI * SM(I,J)

```



```

• 260 •
SMA(I,J) = M * SMA(I,J)
SM(J,I) = SM(I,J)
0470 SMA(J,I) = SMA(I,J)
K = DOI(NE)
FF(2 * K - 1) = FF(2 * K - 1) - Q * L / 2.
FF(2 * K) = FF(2 * K) - Q * L * L / 12.
K = DOJ(NE)
FF(2 * K - 1) = FF(2 * K - 1) - Q * L / 2.
FF(2 * K) = FF(2 * K) + Q * L * L / 12.
IF(NA.EQ.2) THEN
SL(1,1) = -36./L * * 4
SL(2,1) = -12./L * * 3
SL(2,2) = -4./L/L
SL(3,1) = -SL(1,1)
SL(3,2) = -SL(2,1)
SL(3,3) = SL(1,1)
SL(4,1) = SL(2,1)
SL(4,2) = -2./L/L
SL(4,3) = SL(3,2)
SL(4,4) = SL(2,2)
DII = DI/IZ
DO 0480 I = 1,4
DO 0480 J = 1,I
SD(I,J) = DII * SM(I,J)
SL(I,J) = EI * SL(I,J)

```

计算单元刚度、质阵各元素及等效
荷载对设计变量的偏导数

```

SD(J,I) = SD(I,J)
SL(J,I) = SL(I,J)
K = DOI(NE)
FFD(2*K-1) = FFD(2*K-1) - DQ*L/2.
FFD(2*K) = FFD(2*K) - DQ*L*L/12.
FFL(2*K-1) = FFL(2*K-1) - Q/2.
FFL(2*K) = FFL(2*K) - Q*L/6.
K = DOJ(NE)
FFD(2*K-1) = FFD(2*K-1) - DQ*L/2.
FFD(2*K) = FFD(2*K) + DQ*L*L/12.
FFL(2*K-1) = FFL(2*K-1) - Q/2.
FFL(2*K) = FFL(2*K) + Q*L/6.
SAL(1,1) = 13./35
SAL(2,1) = 11.*L/105.
SAL(2,2) = L*L/35.
SAL(3,1) = 9./70.
SAL(3,2) = 13.*L/210.
SAL(3,3) = SAL(1,1)
SAL(4,1) = -SAL(3,2)
SAL(4,2) = -3.*L*L/140.
SAL(4,3) = -SAL(2,1)
SAL(4,4) = SAL(2,2)
BM = DM/M
DO 0490 I = 1,4
DO 0490 J = 1,I

```

```
SAL(I,J) = M * SAL(I,J)
SAD(I,J) = BM * SMA(I,J)
SAL(J,I) = SAL(I,J)
SAD(J,I) = SAD(I,J)
0490
END IF
RETURN
END
```

```
SUBROUTINE TSME(NE,DOI,DOJ,B,A)
COMMON/C2/NEL,NOD,ND
INTEGER P,S,T,SI,SJ,SIM,SJN,DOI(NEL),DOJ(NEL),Z(2)
REAL A(ND,ND),B(4,4)
Z(1) = DOI(NE)
Z(2) = DOJ(NE)
DO 0500 I = 1,2
DO 0500 J = 1,2
K = Z(I)
P = Z(J)
KI = (K - 1) * 2
KJ = (P - 2) * 2
IF(K.EQ.DOI(NE)) THEN
S = 1
ELSE
S = 2
END IF
```

子 程 序 TSME
组集结构总体附、质阵

```

IF(P.EQ.DOI(NE)) THEN
T = 1
ELSE
T = 2
END IF
SI = (S - 1) * 2
SJ = (T - 1) * 2
DO 0500 M = 1, 2
DO 0500 N = 1, 2
KIM = KI + M
KJN = KJ + N
SIM = SI + M
SJN = SJ + N
0500 A(KIM, KJN) = A(KIM, KJN) + B(SIM, SJN)
RETURN
END

```

```

SUBROUTINE BCTR(H, NOZ, NONZ, INDEX, T, S)
COMMON /C2/NEL, NOD, ND, ZN
INTEGER H, INDEX(ND), NOZ(NZ)
REAL NONZ(NZ), T(ND, ND), S(ND)
DO 0510 I = 1, ND
0510 INDEX(I) = I + 30
DO 0520 I = 1, NZ
K = NOZ(I)

```

子 程 序 BCRT

边界条件处理

```

0520 INDEX(K) = 0.
DO 0550 I = 1,ND
IF(INDEX(I).EQ.0) THEN
S(I) = 0.
DO 0530 J = 1,NZ
K = NOZ(J)
IF(NONZ(J).NE.0.) S(K) = NONZ(J)
0530 CONTINUE
DO 0540 K = 1,ND
T(I,K) = 0.
0540 T(K,I) = 0.
IF(H.EQ.0) T(I,I) = 1.
END IF
0550 CONTINUE
RETURN
END

```

```

SUBROUTINE DERP(G,DOI,DOJ,NOZ,NONZ,DD,DL,FF,FFD,FFL,TSD,TSL,
1 TAD,TAL,Y1,PR,DP,W)
COMMON /C1/PI/C2/NEL,NOD,ND,NZ/C3/NM,NT
INTEGER G,DOI(NEL),DOJ(NEL),NOZ(NZ),W(NM,NEL)
REAL NONZ(NZ),DD(NEL),DL(NEL),FF(ND),FFD(ND),FFL(ND),DP(NM),
1 SD(4,4),SL(4,4),SAD(4,4),SAL(4,4),TSD(ND,ND),TSL(ND,ND),
2 TAD(ND,ND),TAL(ND,ND),Y1(ND)
CALL DERP(G,1,NT,FFD,SD,SAD,TSD,TAD,DOI,DOJ,NOZ,NONZ,DD,DL,FF,

```

子 程 序 DERP,DRP

计算结构一阶固有频率对于设计变量的
偏导数

```

1 FFD,FFL,Y1,PR,DP,W)
CALL DRP(G,NT+1,NM,FFL,SL,SAL,TSL,TAL,DOI,DOJ,NOZ,NOZ,DD,DL,DL,
1 FF,FFD,FFL,Y1,PR,DP,W)
RETURN
END

SUBROUTINE DRP(G,N,N1,A,S,S1,T,T1,DOI,DOJ,NOZ,NOZ,DD,DL,FF,
1 FFD,FFL,Y1,PR,DP,W)
COMMON/C1/PI/C2/NEL,NOD,ND,NZ/C3/NM
INTEGER G,DOI(NEL),DOJ(NEL),NOZ(NZ),W(NM,NEL),INDEX(80)
REAL A(ND),S(4,4),S1(4,4),T(ND,ND),T1(ND,ND),NONZ(NZ),DD(NEL),
1 DL(NEL),FF(ND),FFD(ND),DP(NM),B(80),C(80),Y1(ND),
2 SM(4,4),SD(4,4),SL(4,4),SMA(4,4),SAD(4,4),SAL(4,4)
NA = G
DO 0640 I = N,N1
DO 0620 K = 1,ND
A(K) = 0.
DO 0620 J = 1,ND
T(K,J) = 0.
0620 T1(K,J) = 0.
DO 0630 NE = 1,NEL
IF(W(L,NE).EQ.1) THEN
IF(N.EQ.1) THEN
CALL SMNE(NA,NE,DOI,DOJ,DD,DL,SM,S,SL,SMA,S1,SAL,FF,A,FFL)
ELSE

```

```

CALL SMNE(NA,NE,DOI,DOJ,DD,DL,SM,SD,S,SMA,SAD,S1,FF,FFD,A)
END IF
CALL TSME(NE,DOI,DOJ,S,T)
CALL TSME(NE,DOI,DOJ,S1,T1)
END IF
0630 CONTINUE
CALL BCTR(1,NOZ,NOZ,INDEX,T,A)
CALL MATMU2(ND,T,Y1,B)
CALL MATMU2(ND,T1,Y1,C)
CALL MATMU1(ND,Y1,B,AA)
CALL MATMU1(ND,Y1,C,BB)
0640 DP(I) = 0.012665148 * AA/PR - 0.5 * PR * BB
RETURN
END

```

```

SUBROUTINE DERY(G,DOI,DOJ,NOZ,NOZ,DD,DL,DM,TS,TSD,TSL,FF,
1 FFD,FFL,Y,DYY,DY,W)
COMMON /C2/NEL,NOD,ND,NZ/C3/NM,NT
INTEGER G,DOI(NEL),DOJ(NEL),NOZ(NZ),W(NM,NEL)
REAL NOZ(NZ),DD(NEL),DL(NEL),FF(ND),FFD(ND),FFL(ND),TM(ND,
1 ND),TS(ND,ND),TSD(ND,ND),TSL(ND,ND),Y(ND),DYY(ND,NM),SD(4,4)
1 SL(4,4),DY(NM)
CALL DRY(G,1,NT,FFD,SD,TSD,DOI,DOJ,NOZ,NOZ,DD,DL,DM,TS,FF,
1 FFD,FFL,Y,DYY,W)
CALL DRY(G,NT+1,NM,FFL,SL,TSL,DOI,DOJ,NOZ,NOZ,DD,DL,DM,TS,

```

子 程 序 DERY,DRY

计算结构扭曲变形对于设计变量的偏导数

*

```
1 FF,FFD,FEL,Y,DYY,W)
DO 0010 I=1,NM
0010 DY(I)=DYY(ND-1,I)
RETURN
END

SUBROUTINE DRY(G,N,N1,A,S,T,DOI,DOJ,NOZ,NOZ,NONZ,DD,DL,DM,TS,
1 FF,FFD,FEL,Y,DYY,W)
COMMON/C2/NEL,NOD,ND,NZ/C3/NM
INTEGER G,DOI(NEL),DOJ(NEL),NOZ(NZ),IK(40),JK(40),W(NM,NEL),
INDEX(80)REAL A(ND),S(4,4),T(ND,ND),NONZ(NZ),DD(NEL),DL(NEL),
1 TM(ND,ND),TS(ND,ND),FF(ND),FFD(ND),FEL(ND),Y(ND),
2 DYY(ND,NM),SM(4,4),SD(4,4),SL(4,4),SMA(4,4),SAL(4,4),SAD(4,4),
8 B(40),C(80)
LOGICAL BOOL
BOOL=.TRUE.
NA=G
DO 0610 I=N,N1
DO 0570 K=1,ND
A(K)=0.
DO 0570 J=1,ND
0570 T(K,J)=0.
DO 0580 NE=1,NEL
IF(W(I,NE).EQ.1) THEN
IF(N,EN.1)THEN
```



```

CALL SMNE(NA,NE,DOI,DOJ,DD,DL,SM,S,SL,SMA,SAD,SAL,FF,A,FFL)
ELSE
CALL SMNE(NA,NE,DOI,DOJ,DD,DL,SM,SD,S,SMA,SAD,SAL,FF,FFD,A)
END IF
CALL TSME(NE,DOI,DOJ,S,T)
END IF
0580 CONTINUE
CALL BCTR(J,NOZ,NOZ,INDEX,T,A)
CALL MATMU2(ND,T,Y,B)
DO 0590 K = 1,ND
C(K) = A(K) - B(K)
IK(K) = 0
JK(K) = 0
DO 0590 J = 1,ND
TS(K,J) = TM(K,J)
CALL GJ(ND,TS,C,1,E - 7,IK,JK,BOOL)
IF(.NOT.BOOL) STOP/FAIL3
DO 0600 J = 1,ND
0600 DYY(J,I) = C(J)
0610 CONTINUE
RETURN
END

```

```

SUBROUTINE GJ(N,A,B,EPS,IK,JK,BOOL)
INTEGER IK(N),JK(N)
REAL A(N,N),B(N)
LOGICAL BOOL
DO 40 K=1,N
C=0.
DO 10 I=1,N
DO 5 L=1,K-1
IF(I.EQ,IK(L)) GO TO 10
CONTINUE
DO 10 J=1,N
IF(ABS(A(I,J)).LE.ABS(C)) GO TO 10
C=A(I,J)
IO=I
JO=J
CONTINUE
10 IF(ABS(C).LE.EPS) GO TO 15
GO TO 20
15 BOOL=.FALSE.
GO TO 50
20 IK(K)=IO
JK(K)=JO
DO 30 J=1,N
DO 25 L=1,K
IF(J.EQ,JK(L)) GO TO 30

```

子 程 序 GJ

求解线性方程组 $AX=B$ 的高斯—约当
过程

```

25      CONTINUE
      IF(A(IO,J).EQ.0) GO TO 30
      D,A(IO,J) = A(IO,J)/C
      DO 30 I = 1,N
      IF(I.EQ.IO) GO TO 30
      A(I,J) = A(I,J) - A(I,JO)*D
30      CONTINUE
      D,B(IO) = B(IO)/C
      DO 35 I = 1,N
      IF(I.EQ.IO) GO TO 35
      B(I) = B(I) - A(I,JO)*D
35      CONTINUE
      DO 40 I = 1,N
      A(I,JO) = 0.
      DO 45 K = 1,N
      A(1,JK(K)) = B(JK(K))
      DO 50 K = 1,N
      B(K) = A(1,K)
50      CONTINUE
      RETURN
      END

```

```

SUBROUTINE MATMUL(N,A,B,C)
DIMENSION A(N),B(N)
C = 0.

```

子程序 MATMUL, MATMUL2
 计算行、列矩阵的乘积, 即
 $C = A_{1 \times N} \cdot B_{N \times 1}$

```

DO 5 I=1,N
  C=C+A(I)*B(I)
RETURN
END

```

5

```

SUBROUTINE MATMU2(N,A,B,C)
DIMENSION A(N,N),B(N),C(N)
DO 5 I=1,N
  C(I)=0.
DO 5 J=1,N
  C(I)=C(I)+A(I,J)*B(J)
RETURN
END

```

5

```

SUBROUTINE LUBA(A,B)
REAL A(0:2,0:3),B,(0:2),T(0:3)
B(2)=A(2,3)/A(2,2)
B(1)=0
IF(A(0,0).NE.0) GOTO 30
IF(A(1,0).NE.0) THEN
DO 10 J=0,3
  T(J)=A(0,J)
  A(0,J)=A(1,J)
  A(1,J)=T(J)
GO TO 30

```

10

• 271 •

计算方阵与列矩阵的乘积, 即

$$C_{N \times 1} = A_{N \times N} \cdot B_{N \times 1}$$

子程序 LUBA

求解拉格朗日乘子 λ 或性能修改因子 β

```
END IF
IF(A(2,0).NE.0) THEN
DO 20 J=0,3
T(J) = A(0,J)
A(0,J) = A(2,J)
A(2,J) = T(J)
20 GO TO 30
END IF
30 DO 40 J=0,3
T(0) = A(1,0)
A(1,J) = A(1,J) - A(0,J) * T(0) / A(0,0)
T(1) = A(2,0)
40 A(2,J) = A(2,J) - A(0,J) * T(1) / A(0,0)
50 CONTINUE
IF(A(1,1).NE.0) THEN
T(2) = A(2,1)
DO 60 J=1,3
60 A(2,J) = A(2,J) - A(1,J) * T(2) / A(1,1)
ELSE
DO 70 J=1,3
T(J) = A(1,J)
A(1,J) = A(2,J)
70 A(2,J) = T(J)
END IF
PRINT *, 'A = ', A
```

```

IF(A(2,2).GT.1.E-8) THEN
B(2) = A(2,3)/A(2,2)
B(1) = (A(1,3) - B(2) * A(1,2))/A(1,1)
END IF
IF(A(0,0).NE.0) THEN
B(0) = (A(0,3) - B(1) * A(0,1) - B(2) * A(0,2))/A(0,0)
ELSE
B(0) = 0
END IF
PRINT *, 'B = ', B
RETURN
END

```

```

SUBROUTINE RDTZ(Q,X0,X1,X2,LM1,LM2,DFX,DF,DU,DY,U,YR,PR,U0,
1 YR0,PR0,AL)
COMMON/C3/NM
INTEGER X0(NM),CHA,Q
REAL X1(NM),X2(NM),LM1(NM),LM2(NM),A(0:2,0:3),B(0:2),
1 DFX(NM),DF(NM),DU(NM),DY(NM),AA(0:2,0:3)
DO 0010 I=1,NM
0010 X0(I) = 1
CHA = 0
0020 PRINT *, 'CHA = ', CHA
J = 0
DO 0030 I=1,NM

```

子 程 序 RDTZ

进行再设计迭代性能修改
当Q = 1时进行再设计迭代
当Q = 2时进行性能修改

```

* 274 *
0030 IF(X0(I),EQ,1) J=J+1
      CONTINUE
      PRINT *, 'J = ', J
      IF(J,EQ,0)GOTO 0120
      A(0,3) = 1 - U/U0
      A(1,3) = 1 - YR/YR0
      A(2,3) = 1 - PR/PR0
      DO 0040 J=0,2
      DA 0040 K=0,2
      0040 A(J,K) = 0.
      DO 0050 I=1,NM
      IF(X0(I),EQ,1) THEN
      IF(Q,EQ,1) THEN
      A0 = - (1 - AL)/DEX(I) * X1(I)
      ELSE
      A0 = 1
      END IF
      A(0,0) = A(0,0) + A0 * DU(I) * * 2/U0 * * 2
      A(0,1) = A(0,1) + A0 * DU(I) * DY(I)/U0/YR0
      A(0,2) = A(0,2) - A0 * DU(I) * DP(I)/U0/PR0
      A(1,1) = A(1,1) + A0 * DY(I) * * 2/YR0 * * 2
      A(1,2) = A(1,2) - A0 * DY(I) * DP(I)/YR0/PR0
      A(2,2) = A(2,2) - A0 * DP(I) * * 2/PR0 * * 2
      IF(Q,EQ,1) THEN
      B0 = (1 - AL) * * X1(I)

```

```

ELSE
B0 = 0
ENDLF
A(0,3) = A(0,3) + B0*DU(I)/U0
A(1,3) = A(1,3) + B0*LD(I)/IR0
A(2,3) = A(2,3) + B0*DB(I)/FR0
END IF
IF(X0(I).EQ.-1)THEN
A(0,3) = A(0,3) - DU(I)*(LM1(I) - X1(I))/U0
A(1,3) = A(1,3) - DY(I)*(LM1(I) - X1(I))/YR0
A(2,3) = A(2,3) - DP(I)*(LM1(I) - X1(I))/PR0
END IF
IF(X0(I).EQ.-2)THEN
A(0,3) = A(0,3) - DU(I)*(LM2(I) - X1(I))/U0
A(1,3) = A(1,3) - DY(I)*(LM2(I) - X1(I))/YR0
A(2,3) = A(2,3) - DP(I)*(LM2(I) - X1(I))/PR0
END IF
0050 CONTINUE
A(0,1) = A(1,0)
A(2,0) = -A(0,2)
A(2,1) = -A(1,2)
PRINT*, 'A = ', A
** CALL LUAB(A,B)
IF(Q.EQ.1) THEN
DO 0060 I=1,2

```



```

IF(B(I).LT.0.) B(I) = 0
0060 CONTINUE
0062 PRINT *, 'LU = ', B
END IF
IF(Q.EQ.2) PRINT *, 'BA = ', B
DO 0070 I = 1, NM
IF(X0(I).EQ.1) THEN
A0 = B(0)/U0 * DU(I) + B(1)/YR0 * DY(I) - B(2)/FR0 * DF(I)
IF(Q.EQ.1) THEN
CI = AL - (1 - AL)/DFX(I) * A0
PRINT *, 'CI = ', CI
X2(I) = CI * X1(I)
END IF
IF(Q.EQ.2) X2(I) = X1(I) + A0
END IF
0070 CONTINUE
JJ = 0
DO 0080 I = 1, NM
IF((X0(I).EQ.1).AND.(X2(I).LE.LM1(I))) THEN
JJ = JJ + 1
X2(I) = LM1(I)
X0(I) = - 1
END IF
IF((X0(I).EQ.1).AND.(X2(I).GE.LM2(I))) THEN
JJ = JJ + 1

```

```

X2(I) = LM2(I)
X0(I) = - 2
END IF
0080 CONTINUE
WRITE(6,0090) X1
WRITE(6,0100) X2
WRITE(6,0110) X0
0090 FORMAT(2X,'X1 = ',/,8(2X,F14.6))
0100 FORMAT(2X,'X2 = ',/,8(2X,F14.6))
0110 FORMAT(2X,'X0 = ',/,8(110,6X))
IF(JJ.GT.0) THEN
CHA = CHA + 1
GOTO 0020
END IF
0120 RETURN
END

```

[**的说明]若进一步考虑 $\lambda < 0$ 对它的影响，需重新计算 λ 值，应将自注**起至0060止一段程序改为

```

DO 0052 I = 0,2
DO 0052 I = 0,3
0052 AA(I,J) = A(I,J)
0055 CALL LUBA(A,B)
IT = 1
IF(Q.EQ.1) THEN
DO 0053 I = 1,2

```

```

IF(B(*).LT.0.) THEN
IT=1
GO TO 0057
END IF
0060 CONTINUE
IF(IT.EQ.0) GOTO 0062
0057 DO 0058 I=1,2
IF(B(I).LT.0) THEN
DO 0053 J=0,2
AA(I,J),AA(J,I)=0
AA(J,I)=1
0053 A(I,J)=AA(I,J)
AA(I,3)=0
END IF
A(J,3)=AA(I,3)
0058 CONTINUE
GOTO 0055

```

[注]该程序系哈工大机床教研室研究生宋爱武所调。

参 考 文 献

- [1] R.L. 福克斯著 张建中等译 工程设计的优化方法 科学出版社 1981年
- [2] D.M. 希梅尔布劳著 张义葵等译 实用非线性规划 科学出版社 1981年
- [3] M. 阿佛里耳著 李克熹等译 非线性规划(上、下册) 上海科技出版社 1980年
- [4] 南京大学数学系计算数学专业 最优化方法 科学出版社 1978年
- [5] 中国科学院数学研究所运筹室优选法小组 优选法 科学出版社 1975年
- [6] D.J. 华尔德 C.S. 皮特勒著 龙云程译 优选法基础 科学出版社 1978年
- [7] 席少霖 赴风治 最优化计算方法 上海科技出版社 1983年
- [8] 李炳威 结构的优化设计 科学出版社 1979年
- [9] 孙靖民主编 机床结构计算的有限元法 机械工业出版社 1983年
- [10] 薛嘉庆 最优化原理与方法 冶金工业出版社 1983年
- [11] 蔡宣三 最优化与最优控制 清华大学出版社 1983年
- [12] 马国瑜 化工最优化基础 化学工业出版社 1982年
- [13] R.H. 加拉格尔 O.C. 齐恩基威克茨著 陈孝安等译 最佳结构设计 国防工业出版社 1978年
- [14] 陈立周 张含英 吴清一 吴继庾 机械优化设计 上海科技出版社 1982年
- [15] 李维铮等 运筹学 清华大学出版社 1982年
- [16] 钱令希 工程结构优化设计 水利电力出版社 1983年
- [17] 邓乃扬等 无约束最优化计算方法 科学出版社 1983年
- [18] 王永乐 机械工程师优化设计基础 黑龙江科学技术出版社 1983年
- [19] 米成秋 孙靖民 机床部件的有限元——优化设计 哈尔滨工业大学学报 1982年 第二期
- [20] 米成秋 孙靖民 机床结构优化方法初探 哈尔滨工业大学学报 1983年 第三期
- [21] 曹宏毅 孙靖民 优化方法剖析 哈尔滨工业大学科技情报 1984年 第一期
- [22] D.G. 鲁恩伯杰著 夏尊铨等译 线性与非线性规划引论 科学出版社 1980年
- [23] 冯康 数值计算方法 国防工业出版社 1978年
- [24] 中国科学院沈阳计算技术研究所等 电子计算机常用算法 科学出版社 1976年
- [25] K.I. 马吉德著 蓝倜恩译 结构最优设计 中国建筑出版社 1980年
- [26] 程极泰 最优设计的数学方法 国防工业出版社 1981年
- [27] 上海计算技术研究所 复旦大学数学系 怎样用算法语言编程序 上海科学技术出版社 1980年

- [28] E.J. Haug J.S. Arora, "Applied Optimal Design" Awiley—Interscience Publication 1979
- [29] R.J. Aguilar, "Systems Analysis and Design" Prentice—Hall Inc, Englewood, Cliffs N.J. 1973
- [30] P.E. Gill W. Murray M.H. Wright, "Practical Optimization" Academic Press Inc (London) LTD. 1981
- [31] R.C. Johnson, "Mechanical Design Synthesis with Optimization Applications" Van Nostrand Reinhold Company 1971
- [32] Gordon S.G, Beveridge and Robert S. Schechter, " Optimization: Theory and Practics" McGraw—Hill Book Company 1970