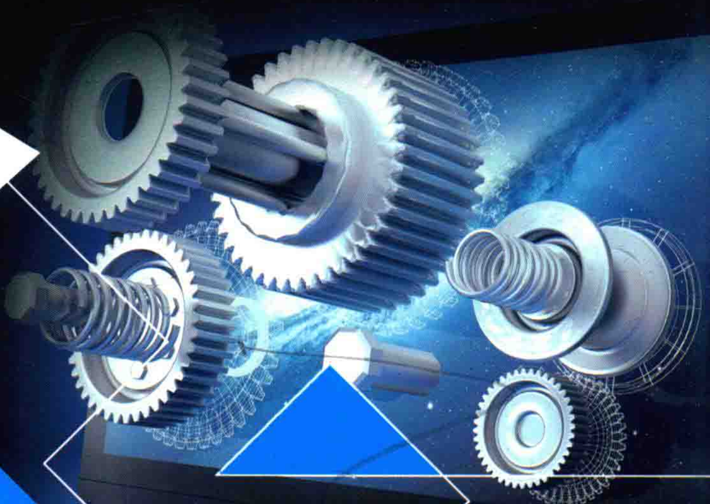
普通高等教育“十三五”规划教材

机械优化设计

Mechanical Optimization Design

李延斌　田　方　主编



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十三五”规划教材

机械优化设计

主编　李延斌　田　方
参编　乔景慧　乔赫廷　张明远
王慧明　张　禹　闫　明
吕晓仁　姜　彤　赵铁军

机 械 工 业 出 版 社

本书主要内容包括：优化设计基本理论、MATLAB 编程基础、MATLAB 优化工具箱及优化计算、典型机械零件的优化设计、渐开线行星齿轮传动、行星齿轮传动装置的优化设计。

本书可作为高等院校机械设计、制造及自动化专业的机械优化设计课程教材，也可作为机械设计专业课程设计的指导书。

图书在版编目（CIP）数据

机械优化设计/李延斌，田方主编．—北京：机械工业出版社，2017.12
普通高等教育“十三五”规划教材
ISBN 978-7-111-58479-7

Ⅰ.①机… Ⅱ.①李… ②田… Ⅲ.①机械设计-最优设计-高等学校-教材 Ⅳ.①TH122

中国版本图书馆 CIP 数据核字（2017）第 279120 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）
策划编辑：余　皞　责任编辑：余　皞　安桂芳
责任校对：郑　婕　封面设计：张　静
责任印制：孙　炜
北京玥实印刷有限公司印刷
2018 年 1 月第 1 版第 1 次印刷
184mm×260mm · 6.75 印张 · 161 千字
标准书号：ISBN 978-7-111-58479-7
定价：25.00 元

前 言

“机械优化设计”课程是高等院校机械类各专业的一门重要技术基础课。随着科学技术和本学科的发展，为了进一步满足教学的需要，编者结合近几年的教学实践，编写了本教材。

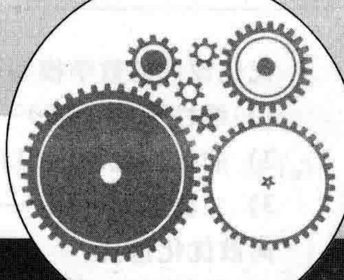
本教材着重介绍机械优化设计方法在典型机械零部件，特别是在行星齿轮减速器设计中的应用。全书以 MATLAB 为优化设计工具，注重工程实际应用。本教材主要内容分为六个部分：一是优化设计基本理论，包括优化数学模型的建立、一维探索最优化方法、无约束多维问题最优化方法、约束问题最优化方法；二是 MATLAB 编程基础，包括 MATLAB 用户界面、MATLAB 路径搜索、MATLAB 应用特点、编程基础、M 文件编辑和调试及文件管理等；三是 MATLAB 优化工具箱及优化计算，包括线性规划问题、二次规划问题、约束非线性规划问题的 MATLAB 优化工具箱使用方法；四是典型机械零件的优化设计，包括滚子链传动优化设计、蜗杆传动优化设计、机床主轴结构优化设计、螺栓组连接的优化设计；五是渐开线行星齿轮传动，包括渐开线行星齿轮传动的基本形式、行星齿轮减速器、行星齿轮传动的特点、行星齿轮传动的配齿计算等；六是行星齿轮传动装置的优化设计，包括 2K-H 型行星齿轮机构优化设计的数学模型，2K-H 型行星齿轮机构齿数选择、优化方法和计算举例，行星齿轮减速器的均载方法、主要部件设计、结构设计及装配。

参与本教材编写的有沈阳工业大学的李延斌、田方、乔景慧、乔赫廷、张明远、王慧明、张禹、闫明、吕晓仁、姜彤、赵铁军。本教材由李延斌、田方任主编。

由于编者的水平和时间有限，书中难免存在疏漏和不当之处，敬请广大读者批评指正。

编 者

目 录



第 1 章 优化设计基本理论

最优化设计是在现代计算机广泛应用的基础上发展起来的一项新技术，是根据最优化原理和方法，综合各方面因素，以人机配合方式或用自动探索方式，在计算机上进行半自动或全自动设计，选出在现有工程条件下最好设计方案的一种现代设计方法。

概括起来，最优化设计工作包括以下两部分内容：

1）将设计问题的物理模型转变为数学模型。建立数学模型时要选取设计变量，列出目标函数，给出约束条件。目标函数是设计问题所要求的最优指标与设计变量之间的函数关系式。

2）采用适当的最优化方法，求解数学模型。可归结为在给定条件（如约束条件）下求目标函数的极值或最优值问题。

1.1 优化设计的数学模型

任何一个最优化问题均可归结为如下的描述，即在满足给定的约束条件（可行域 D 内）下，选取适当的设计变量 $\boldsymbol{X}$，使其目标函数 $f(\boldsymbol{X})$ 达到最优值。其数学表达式（数学模型）为

设计变量：

$$\boldsymbol{X}=[x_1,x_2,\cdots,x_n]^{\mathrm{T}} \quad \boldsymbol{X}\in D\subset E^n \tag{1-1a}$$

在满足约束条件：

$$h_v(\boldsymbol{X})=0 \qquad (v=1,2,\cdots,p) \tag{1-1b}$$

$$g_u(\boldsymbol{X})\leqslant 0 \qquad (u=1,2,\cdots,m) \tag{1-1c}$$

的情况下，求目标函数 $f(\boldsymbol{X})=\sum\limits_{j=1}^{q}\omega_1\cdot f_j(\boldsymbol{X})$ 的最优值。

目标函数的最优值一般可用最小值（或最大值）的形式来体现，因此最优化设计的数学模型可简化表示为

$$\begin{aligned}&\min f(\boldsymbol{X}) && \boldsymbol{X}\in D\subset E^n\\ &\mathrm{s.t.}\,h_v(\boldsymbol{X})=0 && (v=1,2,\cdots,p)\\ &\quad\;\; g_u(\boldsymbol{X})\leqslant 0 && (u=1,2,\cdots,m)\end{aligned} \tag{1-2}$$

优化设计的数学模型可分为连续优化和离散优化。其中，连续优化包括：

1）线性规划（LP）——目标和约束均为线性函数。

2）非线性规划（NLP）——目标或约束中存在非线性函数。

3）二次规划（QP）——目标为二次函数，约束为线性函数。

离散优化包括：

1）整数规划（IP），决策变量（全部或部分）为整数。

2）整数线性规划（ILP）、整数非线性规划（INLP）。

3）纯整数规划（PIP）、混合整数规划（MIP）。

4）一般整数规划、0-1（整数）规划。

优化设计模型的分类如图 1-1 所示。

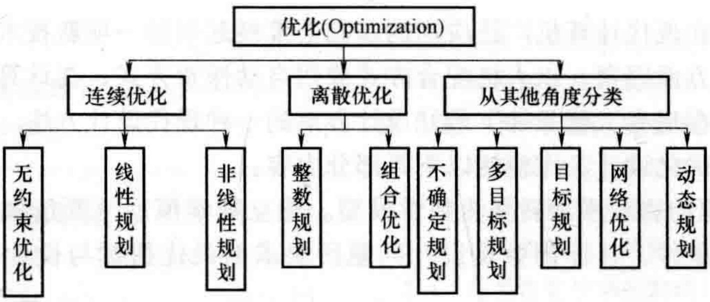


图 1-1　优化设计模型的分类

建模时需要注意的几个基本问题：

1）尽量使用实数优化，减少整数约束和整数变量。

2）尽量使用光滑优化，减少非光滑约束的个数，如尽量少使用绝对值、符号函数、多个变量求最大/最小值、四舍五入、取整函数等。

3）尽量使用线性模型，减少非线性约束和非线性变量的个数（如 $x/y<5$ 改为 $x<5y$）。

4）合理设定变量上下界，尽可能给出变量初始值。

5）模型中使用的参数数量级要适当（如小于 10^3）。

当式（1-1b）和式（1-1c）中的 $p=0$，$m=0$ 时，称为无约束最优化问题；反之称为约束最优化问题。机械最优化设计问题多属于约束非线性最优化问题。

1.1.1　设计变量

在设计过程中，进行选择并最终必须确定的各项独立参数称为设计变量。在选择过程中它们是变量，但这些变量一旦确定以后，设计对象也就完全被确定。最优化设计是研究如何合理优选这些设计变量值的一种设计方法。凡是可以根据设计要求事先给定的，则不是设计变量，而称为设计常量。只有那些需要在设计过程中优选的参数，才可看成是最优化设计过程中的设计变量。

设计变量的数目称为最优化设计的维数，如有 n 个设计变量，则称为 n 维设计问题。在一般情况下，若把具有 n 个设计变量的第 i 个变量记为 x_i，则全部设计变量可用 n 维矢量的形式表示，记为

$$X=\begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix}=(x_1,\cdots,x_i,\cdots,x_n)^{\mathrm{T}} \tag{1-3}$$

在最优化设计中，由各个设计变量的坐标轴所描述的空间称为设计空间。设计空间中的一个点就是一种设计方案。设计空间中的某点 k（一种设计方案）是由各设计变量所组成的矢量 $\boldsymbol{X}^{(k)}$ 决定的。点 k+1（另一种设计方案）则由另一组设计变量所组成的矢量 $\boldsymbol{X}^{(k+1)}$ 确定。最优化设计中通常采用直接探索法，就是在相邻的设计点间做一系列定向的设计移动。由 k 点到（k+1）点间的典型运动情况由下式给出：

$$\boldsymbol{X}^{(k+1)}=\boldsymbol{X}^{(k)}+a^{(k)}\boldsymbol{S}^{(k)} \tag{1-4}$$

矢量 $\boldsymbol{S}^{(k)}$ 决定移动的方向，标量 $a^{(k)}$ 决定移动的步长。

1.1.2 目标函数

在设计中，设计者总是希望所设计的产品具有最好的性能指标、最轻的重量或最大的经济效益等。在最优化设计中，可将所追求的设计目标（最优指标）用设计变量的函数形式表达出来，这一过程称为建立目标函数。即目标函数是优化设计中预期要达到的目标，表述为各设计变量的函数表达式：

$$f(\boldsymbol{X})=f(x_1,x_2,\cdots,x_n) \tag{1-5}$$

式（1-5）是优化设计的某项最重要的特征。如对于泵类液压元件，最常见的情况是以重量最轻或是体积最小作为目标函数。

目标函数是设计变量的标量函数，最优化设计的过程就是优选设计变量，使目标函数达到最优值或找出目标函数最小值（最大值）的过程。

只有一个目标函数的最优化设计称为单目标函数。若同一个优化设计中提出多个目标函数，则这种问题称为多目标函数的最优化问题。

目标函数越多，设计效果越好，但问题的求解也越复杂。对于多目标函数，可以独立地列出几个目标函数式：

$$f_1(\boldsymbol{X})=f_1(x_1,x_2,\cdots,x_n) \tag{1-6a}$$

$$f_2(\boldsymbol{X})=f_2(x_1,x_2,\cdots,x_n) \tag{1-6b}$$

$$\vdots$$

$$f_n(\boldsymbol{X})=f_n(x_1,x_2,\cdots,x_n) \tag{1-6c}$$

也可以把几个设计变量综合到一起，建立一个综合的目标函数表达式，即

$$f(\boldsymbol{X})=\sum_{j=1}^{q}f_j(\boldsymbol{X}) \tag{1-7}$$

式中，q 为最优化设计所追求的目标数目。

1.1.3 约束条件

目标函数取决于设计变量，但在很多实际问题中，设计变量取值范围是有限制的或必须

满足一定条件的。在最优化设计中，这种对设计变量取值时的限制条件，称为约束条件或设计约束。

约束条件可以用数学等式或不等式来表示。

等式约束对设计变量的约束严格，起着降低设计自由度的作用。其形式为

$$h_v(\boldsymbol{X})=0 \qquad (v=1,2,\cdots,p) \tag{1-8}$$

在机械最优化设计中，不等式约束更为普遍。不等式约束的形式为

$$g_u(\boldsymbol{X})\leqslant 0 \qquad (u=1,2,\cdots,m) \tag{1-9a}$$

或

$$g_u(\boldsymbol{X})\geqslant 0 \qquad (u=1,2,\cdots,m) \tag{1-9b}$$

式中，$h_v(\boldsymbol{X})=0$，$g_u(\boldsymbol{X})\leqslant 0$ 为设计变量的约束方程，即设计变量的允许变化范围。最优化设计，即是在设计变量允许范围内找出一组最优化参数 $\boldsymbol{X}^*=(x_1^*,\ x_2^*,\ \cdots,\ x_n^*)^{\mathrm{T}}$，使目标函数 $f(\boldsymbol{X})$ 达到最优值 $f(\boldsymbol{X}^*)$。

对等式约束来说，设计变量所代表的设计点必须在式（1-7）所表示的面（或线）上。对不等式约束来说，其极限情况 $g_u(\boldsymbol{X})=0$ 所表示的几何面（线）将设计空间分为两部分：一部分中的所有点均满足约束条件，即满足式（1-8）或式（1-9），这一部分空间称为设计点的可行域，并用 D 表示。设计变量在可行域中选取的设计点，称为可行设计点或简称可行点。另一部分中的所有点均不满足约束条件，即不满足式（1-8）或式（1-9）。若在这个区域选取设计点，则违背了约束条件，该域中的点称为非可行点。如果设计点落到某个约束的边界面（或边界线）上，则称为边界点，边界点是允许的极限设计方案。最优化设计的过程，即为寻找可行域内最优点或最优设计方案的过程。

1.2 优化方法

1.2.1 一维探索最优化方法

机械结构的最优化设计大都为多维问题，一维问题的情况很少。但是一维问题的最优化方法是优化方法中最基本的方法，在数值方法迭代计算过程中，都要进行一维探索。由于在最优化的大多数方法中，常常要进行一维探索，寻求最优步长或最优方向等，因此一维探索在最优化方法中有很重要的位置。一维探索进行得好坏，直接影响最优化问题的求解速度。

一维探索最优化的方法很多，下面仅介绍本文将采用的进退法。

由单峰函数的性质可知，在极小点左边函数值应单调下降，而在极小点右边函数值应单调上升。根据这一特点，可先给定初始点 a_0 及初始步长 h，求探索区间 $[a,\ b]$。

前进计算：将 a_0 及 a_0+h 代入目标函数进行运算，若 $f(a_0)>f(a_0+h)$，则将步长 h 增加 2 倍，并计算新点 a_0+3h。若 $f(a_0+h)\leqslant f(a_0+3h)$，则探索区间可取为

$$a=a_0;b=a_0+3h$$

否则将步长再加倍，并且重复上述计算。

后退计算：若 $f(a_0)<f(a_0+h)$，则将步长 h 缩为 $h/4$，并从 a_0 点出发，以 $h/4$ 为步长反方向探索，这时得到的后退点为 $a_0-h/4$。若 $f(a_0)<f(a_0-h/4)$，则探索区间可取为

$$a=a_0-\frac{h}{4};b=a_0+3h$$

否则将步长加倍，并继续后退。

1.2.2 无约束多维问题最优化方法

在求解目标函数的极小值的过程中，若对设计变量的取值范围不加任何限制，则称这种最优化问题为无约束最优化问题。无约束多维最优化问题的一般形式为

n 维设计变量 $$X=(x_1,x_2,\cdots,x_n)^{\mathrm{T}} \tag{1-10}$$

目标函数为 $$\min_{X\in E^n} f(x) \tag{1-11}$$

而对 x 没有任何限制。

在实际工程中，无约束条件的设计问题是非常少的，多数问题是有约束的。尽管如此，无约束最优化方法仍然是最优化设计的基本组成部分。因为约束最优化问题可以通过对约束条件的处理，转化为无约束最优化问题来求解。

无约束最优化方法中有直接求优法和间接求优法两种。直接求优法在迭代过程中仅用到函数值，而不要求计算函数导数等解析性质。一般说虽然其收敛速度较慢，但可以解决一些间接法不能解决的问题（如当函数不易求导情况）。无约束多维问题的最优化方法很多，这里仅简要介绍在多齿轮泵优化设计过程中要用到的 Powell 法。

对于一个 n 维问题，Powell 法的迭代计算过程如下：

第一轮探索均是从前一轮求得的最优点出发，并沿 n 个有顺序的线性独立方向（$\boldsymbol{S}_1^{(k)}$，$\boldsymbol{S}_2^{(k)}$，…，$\boldsymbol{S}_n^{(k)}$）进行一维探索。第一轮探索可由任意一点出发，即取 $\boldsymbol{X}_0^{(1)}=\boldsymbol{X}^{(0)}$，而方向可取为 n 个坐标轴的方向，即 $\boldsymbol{S}_i^{(k)}=\boldsymbol{e}_i$。当然，第一轮探索也可以任意取 n 个线性无关的方向组成方向组。现给出第 k 轮迭代的步骤：

1）初始点取前一轮迭代最后沿 $\boldsymbol{S}_{n+1}^{(k-1)}$ 方向求得的最优点 $\boldsymbol{X}^*$（即 $\boldsymbol{X}_{n+1}^{(k-1)}$，有时该点为 $\boldsymbol{X}_n^{(k-1)}$），然后由初始点 $\boldsymbol{X}_0^{(k)}$ 出发，沿 $\boldsymbol{S}_1^{(k)}$ 方向进行一维最优化探索，使函数 $f(\boldsymbol{X}_0^{(k)}+a_1\boldsymbol{S}_1^{(k)})$ 为最小方向，求得 $a_1^{(k)}$，并令 $\boldsymbol{X}_1^{(k)}=\boldsymbol{X}_0^{(k)}+a_1^{(k)}\boldsymbol{S}_1^{(k)}$；再由 $\boldsymbol{X}_1^{(k)}$ 出发，沿 $\boldsymbol{S}_2^{(k)}$ 方向使 $f(\boldsymbol{X}_1^{(k)}+a_2\boldsymbol{S}_2^{(k)})$ 最小，求得 $a_2^{(k)}$，并令 $\boldsymbol{X}_2^{(k)}=\boldsymbol{X}_1^{(k)}+a_2^{(k)}\boldsymbol{S}_2^{(k)}$。如此依次沿每个方向进行一维探索，直至求得全部的 $a_i^{(k)}$（$i=1$，2，…，n），每次令 $\boldsymbol{X}_i^{(k)}=\boldsymbol{X}_{i-1}^{(k)}+a_i^{(k)}\boldsymbol{S}_i^{(k)}$。

2）取共轭方向 $\boldsymbol{S}_{n+1}^{(k)}=\boldsymbol{X}_n^{(k)}-\boldsymbol{X}_0^{(k)}$，计算反映点 $\boldsymbol{X}_{n+1}^{(k)}=2\boldsymbol{X}_n^{(k)}-\boldsymbol{X}_0^{(k)}$。

3）令

$$f_1=f(\boldsymbol{X}_0^{(k)}) \tag{1-12}$$

$$f_2=f(\boldsymbol{X}_n^{(k)}) \tag{1-13}$$

$$f_3=f(\boldsymbol{X}_{n+1}^{(k)})=f(2\boldsymbol{X}_n^{(k)}-\boldsymbol{X}_0^{(k)}) \tag{1-14}$$

式中，

$$\boldsymbol{X}_0^{(k)}=\boldsymbol{X}_{n+1}^{(k-1)}$$

$$\boldsymbol{X}_n^{(k)}=\boldsymbol{X}_{n-1}^{(k)}+a_n^{(k)}\boldsymbol{S}_n^{(k)}=\sum_{i=1}^{n}a_i^{(k)}\boldsymbol{S}_i^{(k)}$$

4）计算第 k 轮迭代各方向上目标函数的下降值 $f(\boldsymbol{X}_{i-1}^{(k)})-f(\boldsymbol{X}_i^{(k)})$（$i=1$，2，…，$n$），并找出其中的最大值 $\Delta_m^{(k)}$，即

$$\Delta_m^{(k)}=\max_{i=1,2,\cdots,n}\left|f(\boldsymbol{X}_{i-1}^{(k)})-f(\boldsymbol{X}_i^{(k)})\right| \tag{1-15}$$

$\Delta_m^{(k)}$ 相应的方向 $S_m^{(k)}=X_i^{(k)}-X_{i-1}^{(k)}$。

5）若 $f_3<f_1$ 和 $(f_1+f_3-2f_2)(f_1-f_2-\Delta_m^{(k)})^2<0.5\Delta_m^{(k)}(f_1-f_3)^2$ 同时成立，则转入下一步；否则，在第 $k+1$ 轮迭代中仍用第 k 轮迭代用的同一方向组，即 $S_i^{(k+1)}=S_i^{(k)}(i=1,2,\cdots,n)$。关于迭代初始点，当 $f_2<f_3$，取第 $k+1$ 轮迭代的初始点 $X_0^{(k+1)}=X_n^{(k)}$，否则取 $X_0^{(k+1)}=X_{n+1}^{(k)}$ 为初始点，然后转回第 1 步。

6）如果上一步中两个不等式同时得到满足，则从 $X_n^{(k)}$ 出发，沿 $S_{n+1}^{(k)}$ 方向进行一维最优化探索，求得 $a^{(k)}$，得 $S_{n+1}^{(k)}$ 方向的最优点为

$$X=X_n^{(k)}+a^{(k)}S_{n+1}^{(k)} \tag{1-16}$$

7）取第 $k+1$ 轮迭代的方向组为

$$(S_1^{(k+1)},S_2^{(k+1)},\cdots,S_n^{(k+1)})=(S_1^{(k)},S_2^{(k)},\cdots,S_{m-1}^{(k)},S_{m+1}^{(k)},\cdots,S_n^{(k)},S_{n+1}^{(k)}) \tag{1-17}$$

也就是说，在新方向组中，去掉了原方向组中具有最大下降值的 $S_m^{(k)}$，并且将方向 $S_{n+1}^{(k)}$ 作为新方向组中的第 n 个方向，即取 $S_n^{(k+1)}=S_{n+1}^{(k)}$。初始点为 $X_0^{(k+1)}$，然后转回第 1 步继续运算。

8）每轮迭代结束时，都应按迭代终止条件进行检查，若满足迭代终止条件，则迭代运算可以结束。迭代终止条件为

$$\| X_i^{(k)}-X_i^{(k-1)} \| \leqslant \varepsilon_1 \quad (i=1,2,\cdots,n) \tag{1-18}$$

或

$$\left\| \frac{f(X^{(k)})-f(X^{(k-1)})}{f(X^{(k-1)})} \right\| \leqslant \varepsilon_2 \tag{1-19}$$

1.2.3 约束问题最优化方法

前面讨论的都是无约束条件下非线性函数的寻优方法，但在实际工程中，大部分问题的变量取值都有一定的限制，也就是属于有约束条件的寻优问题。因此，下面将介绍有约束问题的最优化方法，即设计变量的取值范围受到某种限制时的最优化方法。与无约束问题不同，约束问题目标函数的最小值是满足约束条件下的最小值，而不一定是目标函数的自然最小值。另外，只要由约束条件所决定的可行域 D 是一个凸集，目标函数是凸函数，其约束最优解就是全域最优解。否则，由于所选择的初始点不同，将探索到不同的局部最优解上，所以在这种情况下，探索结果经常与初始点选择有关。为了能得到全域最优解，在探索过程中最好能改变初始点，有时甚至要改换几次。

约束问题最优解的求解过程可归结为，寻求一组设计变量

$$X^*=(x_1^*,x_2^*,\cdots,x_3^*)^{\mathrm{T}},X\in D\subset E^n \tag{1-20}$$

在满足约束方程：

$$h_v(X)=0 \qquad (v=1,2,\cdots,p) \tag{1-21a}$$

$$g_u(X)\leqslant 0 \qquad (u=1,2,\cdots,m) \tag{1-21b}$$

的条件下，使目标函数值最小，即

$$f(X)\to\min f(X)=f(X^*) \tag{1-22}$$

这样所得的最优点 X^* 称为约束最优点。

由上式可见，约束条件可分为两类：等式约束和不等式约束。处理等式约束问题与不等式约束问题的方法有所不同，使约束问题最优化的方法也大致可以分为两大类：

1）约束最优化问题的直接解法。这种方法主要用于求解仅含不等式约束条件的最优化问题。当有等式约束条件时，仅当等式约束函数不是复杂的隐函数，且消元过程容易实现时，才可使用这种方法。其基本思想是在可行域内按照一定的原则直接探索出它们的最优点。而不需要将约束最优化问题转换成无约束问题去求优。属于直接解法的最优化方法有随机试验法、随机方向探索法、复合形法，可行方向法、可变容差法、简约梯度法以及广义简约梯度法、线性逼近法等。

2）约束最优化问题的间接解法。这种方法对于不等式约束问题和等式约束问题均有效。其基本思想是按照一定的原则构造一个包含原目标函数和约束条件的新目标函数，即将约束最优化问题的求解转换成无约束最优化问题的求解。显然，约束问题通过这种方法处理，就可以采用上述无约束最优化方法来求解。属于间接法的约束问题的最优化方法有消元法、拉格朗日乘子法和惩罚函数法等，在本节中仅介绍惩罚函数法。

惩罚函数法分为参数型和无参数型两大类。参数型惩罚函数法在构造惩罚函数时，需要引入一个或几个可调整的惩罚参数（因子）。SUMT（Sequential Unconstrained Minimization Technique）法，也称为序列无约束极小化方法，这是一种有代表性的参数型惩罚函数法。根据惩罚项的函数形式的不同，参数型惩罚函数法（SUMT 法）又分为内点法、外点法和混合点法三种。无参数型惩罚函数法又称为无参数 SUMT 法，是一种借助于修改惩罚函数来使参数自动选取的方法。由于此法较之前法无特别优越处，且收敛慢（多为线性收敛速率），故很少采用。

（1）内点法　内点法是求解不等式约束最优化问题的一种十分有效的方法，但不能处理等式约束。其特点是将构造的新的无约束目标函数——惩罚函数定义于可行域内，并在可行域内求惩罚函数的极值点，即求解无约束问题时的探索点总是保持在可行域内部。

对于目标函数 $f(\boldsymbol{X})$ 受约束于 $g_u(\boldsymbol{X})\leqslant 0(u=1,2,\cdots,m)$ 的最优化问题，利用内点法求解时，惩罚函数的一般形式为

$$\varphi(\boldsymbol{X},r^{(k)})=f(\boldsymbol{X})-r^{(k)}\sum_{u=1}^{m}\frac{1}{g_u(\boldsymbol{X})} \tag{1-23a}$$

或

$$\varphi(\boldsymbol{X},r^{(k)})=f(\boldsymbol{X})-r^{(k)}\left|\sum_{u=1}^{m}\ln g_u(\boldsymbol{X})\right| \tag{1-23b}$$

而对 $f(\boldsymbol{X})$ 受约束于 $g_u(\boldsymbol{X})\geqslant 0$（$u=1,2,\cdots,m$）的最优化问题，其惩罚函数的一般形式为

$$\varphi(\boldsymbol{X},r^{(k)})=f(\boldsymbol{X})+r^{(k)}\sum_{u=1}^{m}\frac{1}{g_u(\boldsymbol{X})} \tag{1-24a}$$

或

$$\varphi(\boldsymbol{X},r^{(k)})=f(\boldsymbol{X})+r^{(k)}\left|\sum_{u=1}^{m}\ln g_u(\boldsymbol{X})\right| \tag{1-24b}$$

式中，$r^{(k)}$ 为惩罚因子，是递减的正数序列。即

$$r^{(0)}>r^{(1)}>\cdots>r^{(k)}>r^{(k+1)}>\cdots>0$$

$$\lim_{k\to\infty}r^{(k)}=0$$

通常，取 $r^{(k)}=1.0,\ 0.1,\ 0.01,\ 0.001$。

上述惩罚函数表达式的右边的第二项 $r^{(k)}\sum_{u=1}^{m}\frac{1}{g_u(X)}$，称为惩罚项。

只要设计点 X 在探索过程中始终保持为可行点，则惩罚项必为正值，且当设计点由可行域内部远离约束边界处移向边界 $[g_u(X)=0]$ 时，惩罚项的值就要急剧增大直至无穷大，于是惩罚函数 $\varphi(X, r^{(k)})$ 也随之急剧增大直至无穷大。这时对设计变量起惩罚作用，使其在迭代过程中始终不会触及约束边界。因此，第二项使约束边界成为探索点是一个不能跳出可行域之外的障碍。

由惩罚函数的表达式可知，对惩罚函数 $\varphi(X, r^{(k)})$ 求无约束极值时，其结果将随给定的惩罚因子 $r^{(k)}$ 而异。为了取得约束面上的最优解，在迭代过程中就要逐渐减小惩罚因子的值，直至为零，这样才能迫使 $\varphi(X, r^{(k)})$ 的极值点 $X^*(r^{(k)})$ 收敛到原函数 $f(X)$ 的约束最优点 X^*。可以把每次迭代所得的 $\varphi(X, r^{(k)})$ 的无约束极值的最优解 $X^*(r^{(k)})$ 看作是以 $r^{(k)}$ 为参数的一条轨迹，当取 $r^{(0)}>r^{(1)}>\cdots>r^{(k)}>r^{(k+1)}>\cdots>0$ 且 $\lim\limits_{k\to\infty}r^{(k)}=0$ 时，点列 $\{X^*(r^{(k)})\}$ 就沿着这条轨迹趋于 $f(X)$ 的约束最优点。因此，惩罚因子 $r^{(k)}$ 又称为惩罚参数。内点法是随着惩罚参数 $r^{(k)}$ 递减序列，使惩罚函数的无约束极值点 $X^*(r^{(k)})$ 从可行域的内部向原目标函数的约束最优化点逼近，直到达到最优点。

（2）外点法　与内点法将惩罚函数定义于可行域内，且求解无约束问题的探索点总是保持在可行域内的特点不同，外点法的特点是将惩罚函数定义于约束可行域之外，且求解无约束问题的探索点是从可行域外部逼近原目标函数的约束最优解的。

对于目标函数 $f(X)$ 受约束于 $g_u(X)\leqslant 0(u=1, 2, \cdots, m)$ 的最优化设计问题，利用外点法求解时，作为无约束新目标函数的惩罚函数，其一般表达式为

$$\varphi(X,M^{(k)})=f(X)+M^{(k)}\sum_{u=1}^{m}\{\max[g_u(X),0]\}^{a} \tag{1-25}$$

$$\lim_{k\to\infty}M^{(k)}=+\infty$$

在惩罚项中，有

$$\max[g_u(X),0]=\frac{g_u(X)+|g_u(X)|}{2}=\begin{cases}g_u(X) & g_u(X)>0\\ 0 & g_u(X)\leqslant 0\end{cases} \tag{1-26}$$

由此可见，当探索点 $X^{(k)}$ 在可行域内时，惩罚项为零；若不在可行域内，则不为零，且 $M^{(k)}$ 越大，受到的惩罚也越大。因此，要使 $\varphi(X, M^{(k)})$ 极小，必须迫使式（1-25）中的惩罚项等于零，即迫使 $g_u(X)\leqslant 0(u=1, 2, \cdots, m)$。这就保证了在可行域内 $\varphi(X, M^{(k)})$ 与 $f(X)$ 是等价的。即

$$\varphi(X,M^{(k)})=\begin{cases}f(X)+M^{(k)}\sum\limits_{u=1}^{m}[g_u(X)]^2 & \text{（在可行域外）}\\ f(X) & \text{（在可行域内）}\end{cases}$$

当约束条件 $g_u(X)\geqslant 0(u=1, 2, \cdots, m)$ 时，惩罚函数的表达式为

$$\varphi(X,M^{(k)})=f(X)+M^{(k)}\sum_{u=1}^{m}\{\min[0,g_u(X)]\}^{a} \tag{1-27}$$

一般取 $a=2$。同样可得

$$0<M^{(0)}<M^{(1)}<\cdots<M^{(k)}<M^{(k+1)}<\cdots<\to+\infty$$

在惩罚项中，有

$$\min[0,g_u(X)]=\frac{g_u(X)-|g_u(X)|}{2}=\begin{cases}g_u(X) & g_u(X)<0\\ 0 & g_u(X)\geqslant 0\end{cases} \tag{1-28}$$

当约束条件中包括 $h_v(X)=0(v=0,1,\cdots,p)$ 的等式约束时，在式（1-25）和式（1-27）中的右边需加进第三项——惩罚项 $M^{(k)}\sum_{v=1}^{p}[h_v(X)]^2$。对惩罚函数 $\varphi(X,M^{(k)})$ 求无约束极值，其结果将随给定的惩罚因子 $M^{(k)}$ 的值而异。可以将惩罚函数无约束有值问题的最优解 X^*（$M^{(k)}$）看作以 $M^{(k)}$ 为参数的一条轨迹，当取 $0<M^{(0)}<M^{(1)}<\cdots<M^{(k)}<M^{(k+1)}<\cdots<\to+\infty$ 时，点列 $\{X^*(M^{(k)})\}$ 就沿着这条轨迹趋于原目标函数 $f(X)$ 的约束最优解。因此，外点法是随着惩罚因子（参数）$M^{(k)}$ 的递增序列，使惩罚函数的无约束 X^*（$M^{(k)}$）从可行域的外部向原目标函数的约束最优点逼近，直至达到最优点。随着惩罚因子的增加由求解一个惩罚函数 $\varphi(X,M^{(k)})$ 的极小值转入到求解另一个惩罚函数 $X^*(M^{(k+1)})$ 的极小值过程中，惩罚项 $M^{(k)}\sum_{u=1}^{m}[g_u(X)]^2$ 之值将逐渐减小，直至为零。

外点法的上述特点，使之很适用于等式的约束的最优化问题。因为在这种情况下，凡是不满足等式约束条件 $h_v(X)=0(v=0,1,\cdots,p)$ 的探索点均是外点。随着探索过程的进行，在求解惩罚函数 $\varphi(X,M^{(k)})=f(X)+M^{(k)}\sum_{v=1}^{p}[h_v(X)]^2$ 极小值的过程中，必须要求最后将惩罚项压缩为零，从而使惩罚函数的无约束极值点等于原目标函数的约束最优点 X^*。

（3）混合法　鉴于内点法和外点法各有优点和缺点，因此在惩罚函数法中又出现了所谓混合法。它是将内点法和外点法的惩罚函数形式结合在一起，用来求解既有不等式约束又有等式约束条件的最优化问题。

根据混合法的基本思想，作为新目标函数的惩罚函数，其处罚项由两部分组成，一部分反映不等式约束的影响并以内点法的构造形式列出；另一部分反映等式约束的影响并以外点法的构造形式列出。混合法惩罚函数的一般表达式为

$$\varphi(X,r^{(k)},M^{(k)})=f(X)-r^{(k)}\sum_{u=1}^{m}\frac{1}{g_u(X)}+M^{(k)}\sum_{v=1}^{p}[h_v(X)]^2 \tag{1-29}$$

即所构造的混合法的惩罚函数，同时含有障碍函数 $r^{(k)}\sum_{u=1}^{m}\frac{1}{g_u(X)}$ 和衰减函数 $M^{(k)}\sum_{v=1}^{p}[h_v(X)]^2$。根据 Fiacco 等建议的关系式 $M^{(k)}=\frac{1}{\sqrt{r^{(k)}}}$，可将惩罚因子统一用 $r^{(k)}$ 表示，则混合法的惩罚函数又可以表示为

$$\varphi(X,r^{(k)},M^{(k)})=f(X)-r^{(k)}\sum_{u=1}^{m}\frac{1}{g_u(X)}+\frac{1}{\sqrt{r^{(k)}}}\sum_{v=1}^{p}[h_v(X)]^2 \tag{1-30}$$

$$(r^{(0)}>r^{(1)}>\cdots>0,\lim_{k\to\infty}r^{(k)}\to 0)$$

混合法与内点法及外点法一样，属于序列无约束极小化方法（SUMT 法）中的一种。利用上式构造混合法的惩罚函数时，其求解具有内点法的特点。这时，其初始点 $X^{(0)}$ 应为内

点；而 $r^{(0)}$ 值可参照内点法选取。混合法的迭代步骤如下：

1）选取初始惩罚因子 $r^{(0)}$ 的值。在 SUMT 程序中，为了简化计算，常取 $r^{(0)}=1$。规定允许误差 $\varepsilon>0$。

2）在可行域 D 内选择一个严格满足所有不等式约束的初始点 $\boldsymbol{X}^{(0)}$。

3）求解 $\min\varphi$（$\boldsymbol{X}$，$r^{(k)}$），得 $\boldsymbol{X}^*$（$r^{(k)}$）。

4）检验迭代终止准则。如果满足式

$$\| \boldsymbol{X}^*(r^{(k)})-\boldsymbol{X}^*(r^{(k-1)}) \| \leqslant \varepsilon_1=10^{-5}-10^{-7}$$

和

$$\left\| \frac{\varphi(\boldsymbol{X}^*,r^{(k)})-\varphi(\boldsymbol{X}^*,r^{(k-1)})}{\varphi(\boldsymbol{X}^*,r^{(k-1)})} \right\| \leqslant \varepsilon_2=10^{-3}-10^{-4}$$

的要求，则停止迭代，并以 $\boldsymbol{X}^*$（$r^{(k)}$）为原目标函数 $f(\boldsymbol{X})$ 的结束最优解，否则转入下一步。

5）取 $r^{(k+1)}=Cr^{(k)}$，$\boldsymbol{X}^{(0)}=\boldsymbol{X}^*$（$r^{(k)}$），$k=k+1$，转向步骤 3），并取 $C=0.1$。

第 2 章

MATLAB编程基础

MATLAB 是 Math Works 公司开发的集算法开发、数据可视化、数据分析以及数值计算于一体的一种高级计算语言和交互式环境。它为满足工程计算的要求而出现，经过不断发展，目前已成为国际公认的优秀工程应用软件之一。MATLAB 不仅可以处理代数问题和数值分析问题，而且具有强大的图形处理及仿真模拟等功能，能很好地帮助工程师及科学家解决实际的技术问题，而且 MATLAB 自身也提供了相关专业领域的工具箱。MATLAB 的一个重要特点就是它有一套程序扩展系统和一组称之为工具箱（toolboxes）的特殊应用子程序。工具箱是 MATLAB 函数的子程序库，每一个工具箱都是为某一类学科专业和应用而定制的，主要包括信号处理、控制系统、神经网络、模糊逻辑、小波分析和系统仿真等方面的应用。

MATLAB 系统由以下 5 个主要部分组成，下面具体进行介绍。

（1）开发环境　由一系列工具组成。这些工具方便用户使用 MATLAB 的函数和文件，其中许多工具采用的是图形用户界面。包括 MATLAB 桌面和命令窗口、历史命令窗口、编辑器和调试器、路径搜索和用于浏览帮助、工作空间、文件的浏览器。

（2）MATLAB 数学函数库　这是一个包含大量计算算法的集合，这些函数包括从最简单最基本的函数（如加、正弦等）到诸如矩阵的特征向量、快速傅里叶变换等较复杂的函数。

（3）MATLAB 语言　这是一个高级的矩阵/阵列语言，它包含控制语句、函数、数据结构、输入输出和面向对象的编程特点。用户可以在命令窗口中将输入语句与执行命令同步，也可以先编写好一个较大的复杂的应用程序（M 文件）后再一起运行。

（4）图形处理　用 MATLAB 可以将矢量和矩阵用图形表现出来，并且可以对图形进行标注和打印。高层次的作图包括二维和三维数据可视化、图像处理、动画和表达式作图，低层次的作图包括定制图形的显示和为用户的 MATLAB 应用程序建立的图形用户界面。

（5）MATLAB 应用程序接口（API）　这是一个库，它允许用户编写可以和 MATLAB 进行交互的 C 或 Fortran 语言程序。

2.1 MATLAB 用户界面

在默认设置下，MATLAB 的用户界面通常包括 4 个窗口，如图 2-1 所示。它们分别是命令行窗口（Command Window）、命令历史窗口（Command History）、工作间管理窗口

(Workspace) 和当前路径窗口 (Current Dictionary)。对这些窗口的认识，是掌握 MATLAB 的基础，本节主要介绍这些窗口的基本知识。

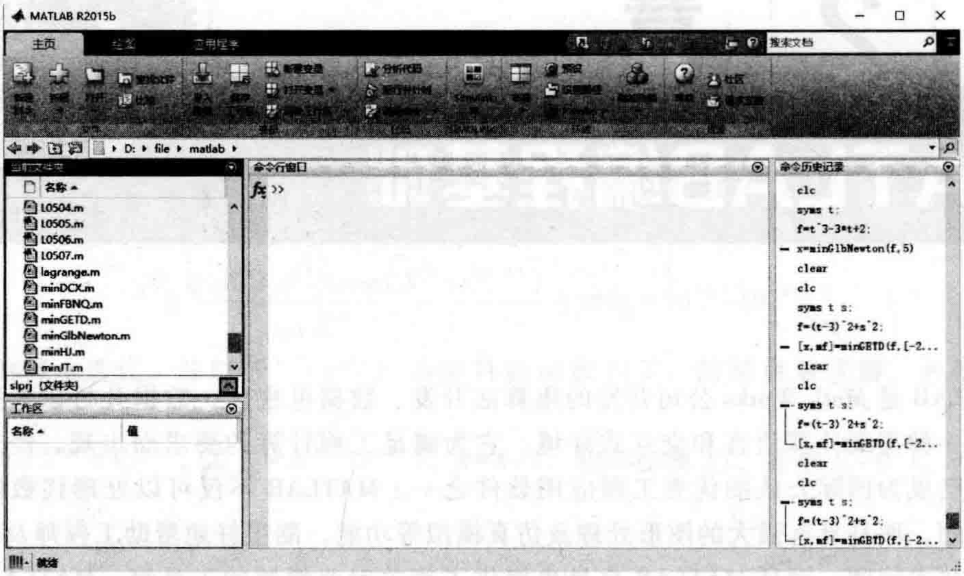


图 2-1　MATLAB 用户界面

在图 2-1 中，各项功能可用鼠标单击激活（也可用 “Desktop”→ “Desktop Layout” 来选择)。

2.1.1　命令行窗口 (Command Window)

在默认设置下，命令行窗口自动显示于 MATLAB 界面中，如果用户只想调出命令行窗口，也可执行 “Desktop”→“Desktop layout” → “Command Window Only” 命令。MATLAB 用户界面的右侧窗口就为命令行窗口，如图 2-2 所示。

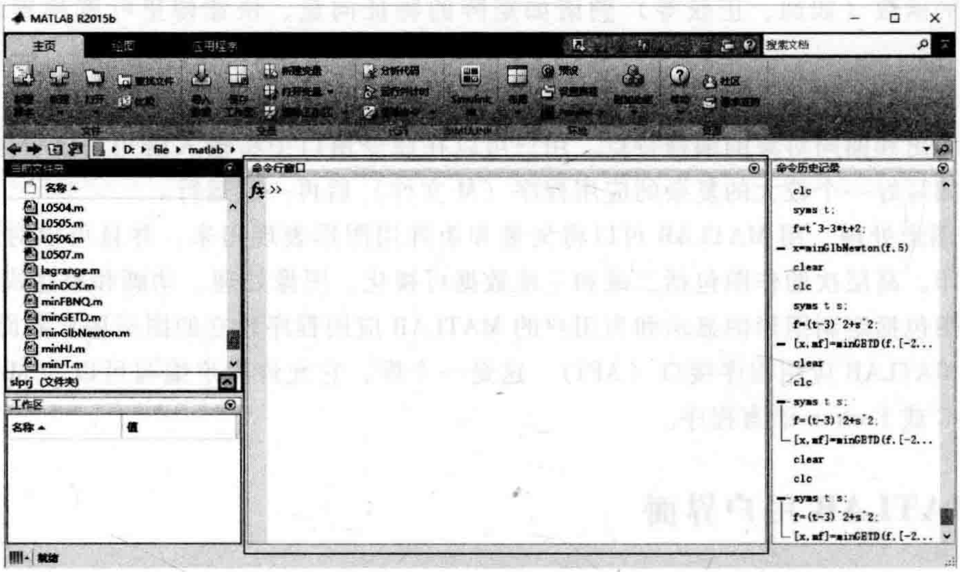


图 2-2　MATLAB 命令行窗口