第8章　最优化问题及Matlab求解

8.1 最优化问题概述

8.2 线性规划问题

8.3 整数规划问题

8.4 其他常见优化问题简介

8.5 现代智能优化算法简介

最优化问题的提出

- 实例
  - 生产计划中，在各种资源有限的前提下，如何安排生产，使生产成本达到最低？
  - 工程施工中，要铺设一条从A地到B地输油管道，中间要经过n个中间站，而对于每个中间站又有m_i个可选方案，如果各个方案在不同两点间的所需经费已知，如何选择一条最佳路线，使得总费用最低？
  - 金融投资中，如何选择和设计证券组合或者投资项目组合，以便在可以接受的风险限度内获得尽可能大的投资回报？
  - 机械设计中，如何在满足工作条件、裁荷和工艺要求，并在强度、刚度、寿命、尺寸范围及其他一些技术要求的限制条件下，寻找一组参数，以获得设计指标达到最优的设计方案？
  - 针对化学过程如何设计控制方案，才能既优化其性能，又能保证其鲁棒性？
  - 在电力分配中，由N个火力发电厂组成一个供电网，要求输出总负荷为S，该如何分配每个发电厂的发电量，在满足各电厂发电量约束的条件下使得总的生产消耗为最小

最优化问题的提出

- 数学描述
  - 各类问题资源的最优利用问题，所有类似的这种课题统称为最优化问题，研究解决这些问题的科学一般就总称之为最优化理论和方法，用数学语言描述的话：

    最优化方法就是在给定的约束条件下，如何在某种范围内选取一些决策变量的取值，使得一个或者多个既定目标达到最优状态（极大、极小或者某种妥协状态）的一门学科。

最优化理论和方法的产生和发展

- 一些古老的方法
  - 黄金分割法
  - 阿基米德证明：如果给定平面几何图形的周长，则在各种图形中，圆所包围的面积为最大。
- 古典最优化方法——精确的分析方法
  - 理论基础的建立——牛顿和莱布尼茨在他们所创建的微积分理论
  - 有约束的最优化问题
  - 变分法
- 最优化理论和方法
  - 由于军事上的需要产生了运筹学
  - 线性规划，非线性规划，动态规划
- 现代优化方法
  - 遗传算法，神经网络，模拟退火

最优化问题的典型实例1

- 资源利用问题— 问题描述

  某工厂生产*A*、*B*两种产品，

  制造1吨*A*产品需要耗煤8吨，耗电3千瓦，耗时2个工作日；

  制造1吨*B*产品需要耗煤4吨，用电4千瓦，耗时9个工作日。

  已知制造1吨产品*A*和*B*分别可以获利6000元和8000元。

  现在该厂原料有煤300吨，电100千瓦，如果需要在200工作日内生产这两种产品并达到利润最大，应当如何安排*A*和*B*的生产数量

资源 \ 活动	生产产品A（吨）	生产产品B（吨）	资源的供应量
煤（吨）	8	4	300
电力（千瓦）	3	4	100
工作日	2	9	200
利润（千元）	6	8	—

最优化问题的典型实例1

- 资源利用问题—问题分析
  - 设x_1和x_2分别代表产品*A*、*B*计划数（单位：吨），*f*表示利润（单位：千元），则问题就是确定*A*、*B*的生产数量x_1和x_2，既可以充分利用资源，又可以使利润最大化
  - 生产*A*可获利8x_1（千元），生产*B*可获利8x_2（千元），故目标函数为 $f = 6x_1 + 8x_2$
  - 优化设计的目标就是使得函数*f*最大化
  - 煤的消耗总量应该小于300吨 $\Rightarrow 8x_1 + 4x_2 \leq 300$
  - 电力资源的消耗不得高于其供应量100千瓦 $\Rightarrow 3x_1 + 4x_2 \leq 100$
  - 劳动力时间的消耗不得高于200工作日 $\Rightarrow 2x_1 + 9x_2 \leq 200$
  - 每种产品数量满足非负限制 $\Rightarrow x_1 \geq 0,\ x_2 \geq 0$

最优化问题的典型实例1

- 资源利用问题—数学模型
  - 模型表达

$$
\begin{cases}
\max & f = 6x_1 + 8x_2 \\
\text{s.t.} & 8x_1 + 4x_2 \le 300 \\
& 3x_1 + 4x_2 \le 100 \\
& 2x_1 + 9x_2 \le 200 \\
& x_1 \ge 0,\ \ x_2 \ge 0
\end{cases}
$$

最优化问题的典型实例2

- 分派问题— 问题描述

  假设某个项目有4项连续的任务构成，即完成了任务1才能开始任务2，完成了任务2之后才能开始任务3，以此类推。并且规定由项目组中的甲乙丙丁四名成员每人完成且仅完成其中的一项任务，四个项目组成员分别完成四项任务的时间如表所示，应该如何分配这些任务，即让哪个成员去完成哪个任务，可以使得花费的总时间最短

时间 成员	任务1（天）	任务2（天）	任务3（天）	任务4（天）
甲	20	12	33	26
乙	22	15	29	23
丙	21	13	31	24
丁	22	16	32	23

最优化问题的典型实例2

- **分派问题— 问题分析**
  - 设计变量

$$x_{ij}=\begin{cases}1 & \text{当指派第}i\text{个成员去完成第}j\text{个任务时}\\ 0 & \text{当不指派第}i\text{个成员去完成第}j\text{个任务时}\end{cases}$$

  - 其中x_{ij}为0-1变量，i代表项目组成员的序号，j代表任务的序号，则：
  - x_{11}、x_{12}、x_{13}、x_{14}分别代表指派甲完成任务1、任务2、任务3、任务4
  - x_{21}、x_{22}、x_{23}、x_{24}分别代表指派乙完成任务1、任务2、任务3、任务4
  - x_{31}、x_{32}、x_{33}、x_{34}分别代表指派丙完成任务1、任务2、任务3、任务4
  - x_{41}、x_{42}、x_{43}、x_{44}分别代表指派丁完成任务1、任务2、任务3、任务4
  - 该问题的目标就是选择一种合适的一对一的组合，使得最后所花费的时间总和最小。根据上述假设，我们可以得到目标函数，即所花费的总时间为：

$$\begin{aligned}f&=20x_{11}+12x_{12}+33x_{13}+26x_{14}+22x_{21}+15x_{22}+29x_{23}+23x_{24}\\&\quad+21x_{31}+13x_{32}+31x_{33}+24x_{34}+22x_{41}+16x_{42}+32x_{43}+23x_{44}\\&=\sum_{i=1}^{4}\sum_{j=1}^{4}E_{ij}x_{ij}\end{aligned}$$

最优化问题的典型实例2

- 分派问题— 问题分析
  - 每个人只能完成一项任务　每项工作有且仅有一人去完成

$$\begin{cases} x_{11} + x_{12} + x_{13} + x_{14} = 1 \\ x_{21} + x_{22} + x_{23} + x_{24} = 1 \\ x_{31} + x_{32} + x_{33} + x_{34} = 1 \\ x_{41} + x_{42} + x_{43} + x_{44} = 1 \end{cases} \qquad \begin{cases} x_{11} + x_{21} + x_{31} + x_{41} = 1 \\ x_{12} + x_{22} + x_{32} + x_{42} = 1 \\ x_{13} + x_{23} + x_{33} + x_{43} = 1 \\ x_{14} + x_{24} + x_{34} + x_{44} = 1 \end{cases}$$

- 数学模型

$$\begin{cases} \min \quad f = \sum_{i=1}^{4}\sum_{j=1}^{4} E_{ij}x_{ij} \\ \text{s.t.} \quad \sum_{j=1}^{4} x_{ij} = 1 \ \ (i = 1,2,3,4) \\ \qquad \sum_{i=1}^{4} x_{ij} = 1 \ \ (j = 1,2,3,4) \\ \qquad x_{ij} = 0 \text{ 或 } 1 \end{cases}$$

最优化问题的典型实例3

- **投资决策问题— 问题描述**

  某企业有n个项目可供选择投资，并且至少要对其中一个项目投资。已知该企业拥有总资金A元，投资于第i(i=1,2,…n)个项目需花资金a_i元，并预计可收益b_i元。试选择最佳投资方案。

最优化问题的典型实例3

- **投资决策问题— 问题分析**
  - 我们可以设定该问题的目标是要在选择相应投资项目之后使得投资收益率最大，对于某项目我们是投资还是不投资，于是我们令设计变量为$x_i(i=1,2,\ldots n)$，其中当我们决定投资第i个项目时，$x_i=1$，当我们决定不投资第i个项目时，$x_i=0$
  - 目标为投资收益率最大，故目标函数应为总收益和总投资的比值

$$f=\sum_{i=1}^{n} b_i x_i \,/ \sum_{i=1}^{n} a_i x_i$$

  - 至少要对一个项目投资，并且总的投资金额不能超过总资金A

$$\Rightarrow 0<\sum_{i=1}^{n} a_i x_i \leq A$$

  - 由于$x_i(i=1,2,\ldots n)$只取值0或1

$$\Rightarrow x_i(1-x_i)=0,\, i=1,\cdots,n.$$

最优化问题的典型实例3

- 数学模型

$$
\begin{cases}
\max \quad f = \dfrac{\displaystyle\sum_{i=1}^{n} b_i x_i}{\displaystyle\sum_{i=1}^{n} a_i x_i} \\
\text{s.t.} \quad 0 < \displaystyle\sum_{i=1}^{n} a_i x_i \le A \\
\qquad\quad x_i(1-x_i) = 0,\ i = 1,\cdots,n
\end{cases}
$$

最优化问题的数学描述

- 最优化问题的数学模型

  由各例子可以看出，最优化问题涉及的领域非常广泛，形式也千变万化，各自有不同的机理和解决方法，但是它们却可以用统一的数学形式表达，简单的说，均可以转化为最小（或最大）化一个n维变量x的实函数f(x)，其中对变量含有多种约束。一般情况下，最优化问题的数学模型可以表达如下：

$$
\begin{cases}
\min \quad f(\mathbf{x}) \\
\text{s.t.} \qquad h_i(\mathbf{x}) = 0 \qquad i = 1, 2, \cdots p, \quad p < n \\
\qquad\qquad g_j(\mathbf{x}) \le 0 \qquad j = 1, 2, \cdots m
\end{cases}
$$

- 最优化问题的三要素

  设计变量、目标函数、约束条件

最优化问题的数学描述

❖ 最优化问题分类

- 根据设计变量的特征分类
  - 按照设计变量的维数进行分类：

    一维优化问题、n维优化问题
  - 根据设计变量的取值进行分类

    离散最优化、连续最优化
- 根据目标函数的类型分类
  - 只有一个目标的优化问题称为单目标优化问题
  - 存在两个或者两个以上目标函数的优化问题，称为多目标优化问题

最优化问题的数学描述

- 最优化问题分类
  - 根据约束条件的类型分类
    - 有无约束——无约束优化问题和约束优化问题
    - 约束类型——线性规划、二次规划、整数规划、0-1规划、非线性规划等
  - 根据最优化问题的解分类
    - 如果最优化问题的解不随时间变化，则称其为静态最优化问题或参数最优化问题
    - 如果最优化问题的解随时间而变，则称其为动态最优化问题

最优化问题的解决方案

❖ 解决方案的步骤

① 提出需要进行最优化的问题，确定研究问题的范围，并为问题的解决准备一些先决条件，例如与问题相关的数据和资料

② 建立能够反映上述实际情况的最优化问题的数学模型，确定设计变量、目标函数和有关约束条件

③ 对建立的数学模型进行分析和修正，选择合适的优化算法

④ 运用合适的软件编写优化算法程序，对最优化模型进行求解

⑤ 如果是理论问题，对优化结果进行分析和比较，总结规律；如果是实际问题，将所获得的最优解应用到实际问题中进行验证和实施，再进行理论分析

❖ 解决方案的关键问题

- 数学模型的建立
- 优化算法的编制

最优化问题的解决方案

❖ 优化问题的求解方法

① 解析法

- 对于最优化数学模型中的目标函数和约束条件，如果其具有明确的数学解析表达式，则一般可以按照函数求极值的必要条件，用导数方法或者变分法等数学分析的手段求出其解析解，然后按照问题的实际物理意义确定问题的最优解。

② 数值解法

- 如果目标函数或者约束条件较为复杂，或者并没有明确的数学解析表达式，抑或是以现有的解析方法和手段无法求取解析解的优化问题，我们可以用数值解法来解决。基本的思想就是用搜索的方法经过一系列的迭代，使得产生的这些序列能够逐步逼近问题的最优解，数值解法常常需要经验或者试验，同时结果也需要通过实际问题的验证才是有效的。

最优化问题的解决方案

❖ 优化问题的求解方法

③ 混合解法

- 混合解法即是结合了上述两种方法，例如以梯度法为代表的一类解法，这类解法往往是解析法和数值算法相结合的一种方法

④ 其他优化方法

- 诸如以网络图为基础的图论方法和近代发展起来的各种智能优化算法如遗传算法、神经网络方法、蚁群算法、禁忌搜索、粒子群算法等等。

8.1 最优化问题概述

8.2 线性规划问题

8.3 整数规划问题

8.4 其他常见优化问题简介

8.5 现代智能优化算法简介

❖ 什么是线性规划

- 如果最优化问题三要素中的目标函数和约束条件都是线性的，则该最优化问题称为线性规划问题

❖ 线性规划的应用和发展

- 线性规划（Linear Programming，简写成LP）是最优化理论和方法中的重要领域之一。其理论上的完整性、方法上的有效性以及应用上的广泛性，较其他分支都成熟的多，同时很多实际问题都可以转化为线性规划来解决
- 线性规划的要点就是在满足线性约束条件的前提下，使预定的线性目标函数达到最优。现在线性规划已不仅仅是一种数学方法，在理论上，它启发了诸如对偶、凸性等最优化理论的核心概念，在实际中更是大量被运用于经济学和管理学领域，成为科学决策的一个有效手段

线性规划的典型实例

❖ 运输问题

- 设有两个建材厂C1和C2，每年沙石的产量分别为35万吨和55万吨，这些沙石需要供应到W1、W2和W3三个建筑工地，每个建筑工地对沙石的需求量分别为26万吨、38万吨和26万吨，各建材厂到建筑工地之间的运费（万元/万吨）如表所示，问题是应当怎么调运才能使得总运费最少？

运费 工地 / 建材厂	W_1	W_2	W_3
C_1	10	12	9
C_2	8	11	13

❖ 运输问题

▪ 问题分析

- 假设x_{ij}代表建材厂C_i运往建筑工地W_j的数量（万吨），则各建材厂和工地之间的运量可以用下表来表示

分配量（万吨） 工地 / 建材厂	W_1	W_2	W_3	输出总量
C_1	x_{11}	x_{12}	x_{13}	35
C_2	x_{21}	x_{22}	x_{23}	55
接收总量	26	38	26	90

- 目标函数即为总运费

$$f = 10x_{11} + 12x_{12} + 9x_{13} + 8x_{21} + 11x_{22} + 13x_{23}$$

- 建材厂C1、C2的输出量应分别为建材厂C1、C2的产量

$$\Rightarrow \begin{matrix} x_{11} + x_{12} + x_{13} = 35 \\ x_{21} + x_{22} + x_{23} = 55 \end{matrix}$$

- 各工地的沙石需求量应当为从各建材厂接收到沙石的总量

$$\Rightarrow \begin{matrix} x_{11} + x_{21} = 26 \\ x_{12} + x_{22} = 38 \\ x_{13} + x_{23} = 26 \end{matrix}$$

- 运输量x_{ij}为一非负值

$$\Rightarrow x_{ij} \geq 0$$

运输问题

- 数学模型

$$
\begin{cases}
\min & f = 10x_{11} + 12x_{12} + 9x_{13} + 8x_{21} + 11x_{22} + 13x_{23} \\
\text{s.t.} & x_{11} + x_{12} + x_{13} = 35 \\
& x_{21} + x_{22} + x_{23} = 55 \\
& x_{11} + x_{21} = 26 \\
& x_{12} + x_{22} = 38 \\
& x_{13} + x_{23} = 26 \\
& x_{ij} \geq 0 \quad (i = 1,2;\ j = 1,2,3)
\end{cases}
$$

线性规划问题的特征

- 均可以用一组设计变量来表示一种实施方案
- 每个问题都有一定的约束条件，这些条件可以用一组线性等式或者线性不等式表达
- 在上述前提下，一般都有一个目标函数，该函数用于衡量方案的优劣，可以表达为设计变量的一个线性函数，我们的目的一般为使得目标函数达到最大值或者最小值

线性规划问题的标准型

❖ 线性规划问题的一般标准型

- 根据线性规划的定义，线性规划问题即求取设计变量**x=[x$_1$, x$_2$, … x$_n$]**的值，在线性约束条件下使得线性目标函数达到最大，线性规划问题的一般标准型为：

$$
\begin{cases}
\max & f = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{s.t.} & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\
& \cdots\cdots\cdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \\
& x_j \geq 0 \quad j = 1, 2, \cdots n
\end{cases}
$$

其中，c_i、a_{ij}、b_i 为给定的常数

线性规划问题的标准型

- 线性规划问题的标准型的特点
  - 目标函数为设计变量的线性函数，且需要极大化；
  - 约束条件为设计变量的一组线性等式，也称为约束方程组；
  - 设计变量$x_1, x_2, \dots x_n$都有非负限制。

线性规划问题的标准型

❖ 线性规划问题的矩阵标准型

- 利用向量或矩阵符号，线性规划问题的标准型还可以用矩阵形式表达：

$$\begin{cases} \max & f = \mathbf{cx} \\ \text{s.t.} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{cases}$$

- 其中 $\mathbf{c} = [c_1 \quad c_2 \quad \cdots \quad c_n]$ 为n维行向量

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$ 为$m \times n$维矩阵

$\mathbf{b} = [b_1 \quad b_2 \quad \cdots \quad b_n]^T$ 为m维列向量

$\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^T$ 为n维列向量

$\mathbf{x} \geq \mathbf{0}$ 是指其各分量 $x_1, x_2, \cdots x_n \geq 0$

线性规划问题的标准型

❖ 不同类型的非标准型化为标准型的方法

- **问题为极小化目标函数**

设原有线性规划问题为极小化目标函数 $\min \quad f = c_1x_1 + c_2x_2 + \cdots + c_nx_n$ 则可设 $f' = -f$

将极小化目标函数问题转化为极大化目标函数问题 $\max \quad f' = -c_1x_1 - c_2x_2 - \cdots - c_nx_n$

- **约束条件为不等式**

如果原有线性规划问题的约束条件为不等式，则可增加一个或减去一个非负变量，使约束条件变为等式，增加或减去的这个非负变量称为松弛变量。例如，假如约束为

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

则可以在不等式的左边增加一个非负变量 x_{n+1}，使不等式变为等式

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + x_{n+1} = b_i$$

如果约束为 $a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geq b_i$ 则可在不等式的左边减去一个非负变量 x_{n+1}，使不等式变为等式 $a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - x_{n+1} = b_i$

- **模型中的某些变量没有非负限制**

若对某个变量 x_j 并无限制，取值可正可负，这时可设两个非负变量 x_j' 和 x_j''，令 $x_j = x_j' - x_j''$

注意到，因为对原设计变量进行了代换，还需要将代换式代入目标函数和其他约束条件做相应的代换，这样就可以满足线性规划标准型对变量非负的要求

线性规划问题的标准型

❖ 例子

- 将线性规划模型标准化 $\begin{cases} \min & f = x_1 - 2x_2 + x_3 \\ \text{s.t.} & x_1 + x_2 + x_3 \le 3 \\ & x_1 + x_2 - 2x_3 \ge 1 \\ & -x_1 + 2x_2 + 3x_3 = 4 \\ & x_1 \ge 0 \\ & x_2 \ge 0 \end{cases}$

- 将目标函数两边乘上-1转化为求极大值 $\max \quad f = -x_1 + 2x_2 - x_3$
- 原问题的约束条件中的前两个条件均为不等式，在第一个不等式的左边加上一个松弛变量x_4，在第二个不等式的左边减去一个松弛变量x_5，将两者转化为等式约束

$$x_1 + x_2 + x_3 + x_4 = 3$$

$$x_1 + x_2 - 2x_3 - x_5 = 1$$

- 原问题对设计变量x_3没有非负限制，故在此引入非负变量x_3^1和 x_3^2，令$x_3 = x_3^1 - x_3^2$
- 经过上述步骤整理后的标准型为 $\begin{cases} \max & f = -x_1 + 2x_2 - x_3^1 + x_3^2 \\ \text{s.t.} & x_1 + x_2 + x_3^1 - x_3^2 + x_4 = 3 \\ & x_1 + x_2 - 2x_3^1 + 2x_3^2 - x_5 = 1 \\ & -x_1 + 2x_2 + 3x_3^1 - 3x_3^2 = 4 \\ & x_1, x_2, x_3^1, x_3^2, x_4, x_5 \ge 0 \end{cases}$

线性规划问题中解的概念

❖ 概述

- 为了帮助分析线性规划求解过程，先介绍线性规划解的概念。仍然考虑式中的线性规划的矩阵标准型：

$$\begin{cases} \max & f = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{cases}$$

- 求解上述线性规划问题实际上就是要求出向量$\mathbf{x}=[x_1,x_2,\ldots x_n]^T$使其满足$\mathbf{A}\mathbf{x}=\mathbf{b}$和$\mathbf{x}\geqslant 0$，且目标函数$f$达到最大值，这个向量称为**线性规划问题的解**。
- 当求解$\mathbf{A}\mathbf{x}=\mathbf{b}$时，假设独立方程的个数为$m$个，设计变量的维数为$n$，根据线性代数的知识，如果$m=n$，则方程有唯一解，无优化的自由度；如果$m>n$，方程个数大于未知数的个数，则有些约束可能不能被满足，上述两类问题不在我们探讨的范围之列，也就是我们**仅讨论$m<n$**的情况，在这个前提下，方程将有无穷多组解，如果需要直接从这无穷多组解中找出一个非负解使得目标函数取得最大值是很难的

线性规划问题中解的概念

❖ 基本解

- 如果线性规划问题的解存在，则它必定是满足**Ax=b**的有限多个“基本解”中选出的，那么我们的第一个任务就是找出满足方程**Ax=b**的基本解
- 假设独立方程的个数为m个，故**Ax=b**的系数矩阵**A**的秩为m，于是**A**中必有m个列向量是线性无关的，不妨假设A中的前m个列向量线性无关，则这m个列向量可以构成矩阵**A**的m阶非奇异子矩阵，用矩阵**B**表示：

$$\mathbf{B} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & & \ddots & \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 & \cdots & \mathbf{P}_m \end{bmatrix}$$

- **B**是一个m阶的满秩方阵，称**B**为线性规划问题的基，其每一个列向量$\mathbf{P}_j$称为基向量，基向量所对应的设计变量称为基变量，记为 $\mathbf{x}_B = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix}^T$
- **A**中其余n-m个列向量则构成非基矩阵，用矩阵**N**表示：

$$\mathbf{N} = \begin{bmatrix} \mathbf{P}_{m+1} & \mathbf{P}_{m+2} & \cdots & \mathbf{P}_n \end{bmatrix}^T$$

- 非基矩阵**N**的每一个列向量称为非基向量，非基向量所对应的设计变量称为非基变量，记为$\mathbf{x}_N = \begin{bmatrix} x_{m+1} & x_{m+2} & \cdots & x_N \end{bmatrix}^T$

线性规划问题中解的概念

- 基本解
  - 根据上述分析，可以将方程组$\mathbf{Ax}$=$\mathbf{b}$转化为$\begin{bmatrix}\mathbf{B} & \mathbf{N}\end{bmatrix}\begin{bmatrix}\mathbf{x}_B \\ \mathbf{x}_N\end{bmatrix} = \mathbf{b}$
  - 于是有 $\mathbf{x} = \begin{bmatrix}\mathbf{x}_B \\ \mathbf{x}_N\end{bmatrix} = \begin{bmatrix}\mathbf{B}^{-1}\mathbf{b} \\ 0\end{bmatrix}$
  - 上式称为约束方程组$\mathbf{Ax}$=$\mathbf{b}$的一个**基本解**，
  - 一般来说，如果线性规划问题中有n个设计变量，在$\mathbf{Ax}$=$\mathbf{b}$中有m个约束方程(n>m)，则基本解的数量小于或等于C_n^m
  - 基本解**不是线性规划问题的解**，而是仅满足约束方程组的解

线性规划问题中解的概念

❖ 可行解、可行域

- 上面的分析仅考虑了约束方程组**Ax=b**，下面进一步考虑线性规划问题的非负约束。我们称既满足约束方程组**Ax=b**，又满足非负约束**x**≥0的解为线性规划问题的可行解，即可行解满足线性规划问题的所有约束。可行解的集合称为可行域，记作：

$$D = \quad \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0$$

❖ 基本可行解

- 特别的，若线性规划问题的基本解能够满足线性规划问题中的非负约束，即：

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} \geq 0$$

则称该解$\mathbf{x}_B$为基本可行解，简称基可行解，称**B**为可行基。基可行解的数量不会超过 C_n^m 个。显然，基本可行解一定是可行解，基可行解是可行域中一种特殊的解

❖ 最优解

- 能使得线性规划问题的目标函数值达到最大的可行解称为最优解。线性规划问题中的最优解，一定可以在基可行解中找到，而基可行解的数量是有限的，因而这就在理论上保证了可以在有限的步骤之内求出线性规划问题的最优解。

线性规划问题中解的概念

❖ 实例

- 线性规划标准型为$\begin{cases} \max & f = -x_1 + 2x_2 - x_3 \\ \text{s.t.} & x_1 + x_2 + x_3 = 3 \\ & x_1 - x_2 - 2x_3 = 1 \\ & x_1, x_2, x_3 \geq 0 \end{cases}$ 于是 $\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & -2 \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{P}_3 \end{bmatrix}$

- 取矩阵$\mathbf{A}$的线性无关列$\mathbf{P}_1$和$\mathbf{P}_2$构成2阶非奇异子矩阵 $\mathbf{B}_1 = \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$，$\mathbf{B}_1$是线性规划问题的一个基矩阵，与其对应的基变量为$x_1$和$x_2$，即$\mathbf{x}_{B_1} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$，相应的非基矩阵和非基变量分别为

  $$\mathbf{N}_1 = \mathbf{P}_3 = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \qquad \mathbf{x}_{N_1} = x_3$$

  令非基变量x_3=0，可以求出对应基矩阵$\mathbf{B}_1$的基本解为$\mathbf{x}_{\mathbf{B}_1} = \begin{bmatrix} 2 & 1 & 0 \end{bmatrix}$，基矩阵$\mathbf{B}_1$解向量的各分量均为非负，故是线性规划问题的基本可行解。如果这个解可以使得目标函数取得最大值则该解为最优解。是否最优解的判断方法将在后续的章节中探讨。

- 若取矩阵$\mathbf{A}$的后两个线性无关列$\mathbf{P}_2$和$\mathbf{P}_3$，构成线性规划的另一个基矩阵$\mathbf{B}_2 = \begin{bmatrix} \mathbf{P}_2 & \mathbf{P}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix}$

  用相同的方法进行分析，可知，此时的基变量为x_2和x_3，非基变量为x_1，于是令x_1=0，可以得到对应基矩阵$\mathbf{B}_2$的一组基本解为：$\mathbf{x}_{\mathbf{B}_2} = \begin{bmatrix} 0 & 7 & -4 \end{bmatrix}$，由于对应基矩阵$\mathbf{B}_2$解向量的第二个分量即$x_3$为负，故该解不是线性规划问题的基本可行解

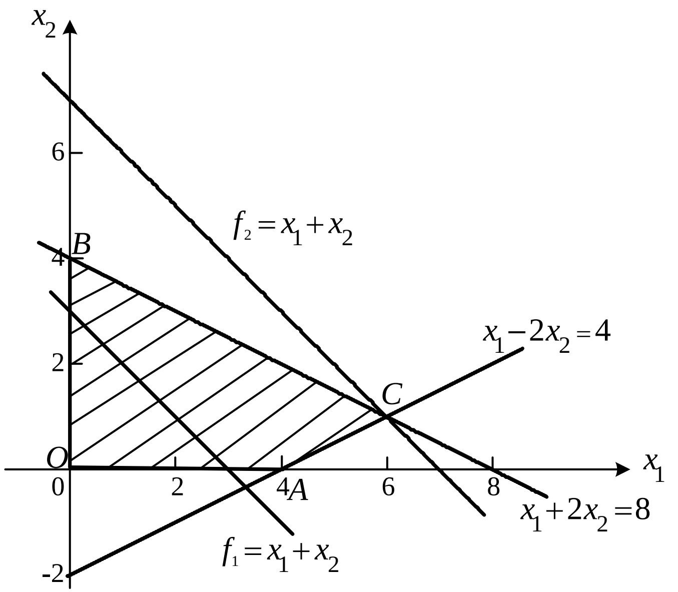
- 由理论分析可知，该线性规划问题基本解的个数为3个，也就是还可以选取$\mathbf{P}_1$和$\mathbf{P}_3$构成基矩阵$\mathbf{B}_3$，求取该问题的第三个基本解，只要有一个基矩阵，就可以求出一个对应的基本解，至于该基本解是否基本可行解和最优解则需要进一步判断。

线性规划问题的解法：图解法

❖ 图解法

$$\begin{cases} \max & f = x_1 + x_2 \\ \text{s.t.} & x_1 - 2x_2 \le 4 \\ & x_1 + 2x_2 \le 8 \\ & x_1, x_2 \ge 0 \end{cases}$$

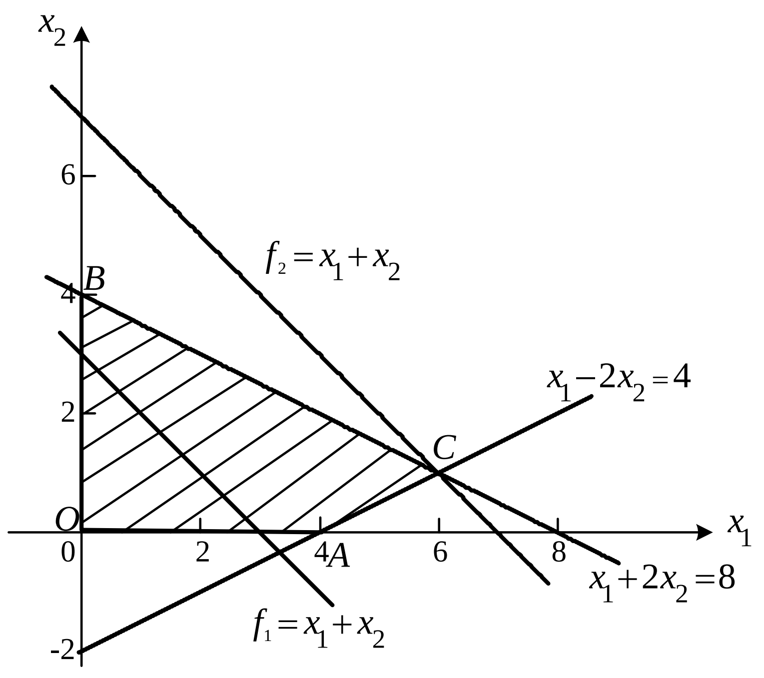
- 分析可行域
  - 引入平面直角坐标系，以x_1作为横轴，以x_2作为纵轴，由于线性规划问题满足非负条件$x_1, x_2 \ge 0$，故问题的探讨局限在平面直角坐标系的第一象限
  - 分析$x_1-2x_2\le 4$，取直线$x_1-2x_2=4$，则直线上的点和直线以上的区域满足该不等式
  - 分析$x_1+2x_2\le 8$，取直线$x_1+2x_2=8$，则直线上的点和直线以下的区域满足不等式
  - 于是可行域为四边形$ACBO$内的区域（包括边界上的点），在图中用阴影表示



线性规划问题的解法

❖ 图解法

- 分析最优解
  - 分析目标函数$f=x_1+x_2$，可以将其改写成为$x_2=-x_1+f$可以发现改写后的方程是以f为参量，以-1为斜率的一族平行的直线，这些平行线越向右上方移动，离原点越远，对应的目标函数值就越大。当直线运动到经过点C时，即不能再继续向上移动，否则将脱离线性规划问题的可行域，故线性规划问题在点C达到最大值



线性规划问题的MATLAB求解

❖ **MATLAB标准型**

$$\begin{cases} \min & f = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} \\ & \mathbf{lb} \le \mathbf{x} \le \mathbf{ub} \end{cases}$$

- **MATLAB标准型和前面理论知识讲解中有所不同，**
  - 在上述模型中，有一个需要**极小化**的目标函数f，以及需要满足的**约束条件**
  - 假设x为n维设计变量，且线性规划问题具有不等式约束m_1个，等式约束m_2个，那么：c、x、lb 和ub 均为n维列向量，b为m_1维列向量，$\mathbf{b}_{eq}$为m_2维列向量，A为$m_1 \times n$维矩阵，$\mathbf{A}_{eq}$为$m_2 \times n$维矩阵

线性规划问题的MATLAB求解

- MATLAB标准型
  - 注意事项
    - MATLAB标准型是对目标函数求极小，如果遇到是对目标函数求极大的问题，在使用MATLAB求解时，需要在函数前面加一个负号转化为对目标函数求极小的问题；
    - MATLAB标准型中的不等式约束形式为“≤”，如果在线性规划问题中出现“≥”形式的不等式约束，则我们需要在两边乘以(-1)使其转化为MATLAB中的“≤”形式。如果在线性规划问题中出现了“<”或者“>”的约束形式，则我们需要通过添加松弛变量使得不等式约束变为等式约束
    - 之后，我们只需要将所有的约束（包括不等式约束和等式约束）转化为矩阵形式的即可

线性规划问题的MATLAB求解

$$\begin{cases} \max & f = 4x_1 - 2x_2 + x_3 \\ \text{s.t.} & 2x_1 - x_2 + x_3 \le 12 \\ & -8x_1 + 2x_2 - 2x_3 \ge 8 \\ & -2x_1 + x_3 = 3 \\ & x_1 + x_2 = 7 \\ & x_1, x_2, x_3 \ge 0 \end{cases}$$

❖ 将问题转化为MATLAB标准型

- 原问题是对目标函数求极大，故添加负号使目标变为：**min f=-4x_1+2x_2-x_3**
- 原问题中存在“≥”的约束条件，故添加负号使其变为**8x_1-2x_2+2x_3≤-8**
- 将约束整理为矩阵形式 $\begin{bmatrix} 2 & -1 & 1 \\ 8 & -2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \le \begin{bmatrix} 12 \\ -8 \end{bmatrix}$ $\begin{bmatrix} -2 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \end{bmatrix}$
- 用MATLAB表达则为

```
c=[-4; 2; -1];                     %将目标函数转化为求极小
A=[2 -1 1; 8 -2 2]; b=[12; -8];    %不等式约束系数矩阵
Aeq=[-2 0 1; 1 1 0];beq=[3; 7];    %等式约束系数矩阵
lb=[0; 0; 0];ub=[Inf; Inf; Inf]    %对设计变量的边界约束
```

线性规划问题的MATLAB求解

- **函数调用格式**
  - MATLAB优化工具箱中求解线性规划问题的命令为**linprog**，其函数调用方法有多种形式如下所示

```
x = linprog(c, A, b)
x = linprog(c, A, b, Aeq, beq)
x = linprog(c, A, b, Aeq, beq, lb, ub)
x = linprog(c, A, b, Aeq, beq, lb, ub, x0)
x = linprog(c, A, b, Aeq, beq, lb, ub, x0, options)
x = linprog(problem)
[x, fval] = linprog(...)
[x, fval, exitflag] = linprog(...)
[x, fval, exitflag, output] = linprog(...)
[x, fval, exitflag, output, lambda] = linprog(...)
```

线性规划问题的MATLAB求解

❖ 输入参数

- MATLAB工具箱中的linprog函数在求解线性规划问题时，提供的参数为：模型参数、初始解参数和算法控制参数。
  - 模型参数x、c、lb、ub、b、b*eq*、A和A*eq*在MATLAB标准型中已经介绍了其具体物理意义和获得方法
  - x0为线性规划问题的初始解，该设置仅在中型规模算法中有效，而在默认的大型规模算法和单纯形算法中，MATLAB将忽略一切初始解。
  - options为包含算法控制参数的结构变量，我们可以通过optimset命令对这些具体的控制参数进行设置，例如下述格式

    options = optimset('param1', value1, 'param2', value2, ...)

    该命令格式创建一组控制参数结构变量options，将参数的具体值赋给单引号之间的参数，任何未被指定的参数将被赋值为[ ]，参数值为[ ]的具体的含义是将该组控制参数传递给优化函数时将使用MATLAB提供的默认值

线性规划问题的MATLAB求解

参数名称	参数设置
Diagnostics	设置是否显示函数优化中的诊断信息，可以选择on或者off（默认值），该功能主要显示一些退出信息，即linprog函数运算终止的原因
Display	设置显示信息的级别，当该参数值为off时，不显示任何输出信息；当参数值为iter时，将显示每一步迭代的输出信息，iter参数值仅对大型规模算法和中型规模的单纯形算法有效；当参数值为final时，仅显示最终的输出信息
Simplex	当该参数值为on时，函数采用单纯形算法
LargeScale	设置是否采用大型规模算法，当参数值为on（默认值）时，使用大型规模算法；当参数值为off时，使用中型规模算法
MaxIter	算法运行中的最大迭代次数，对于大型规模算法，默认值为85，对于单纯形算法，其默认值为10×设计变量的个数，对于中型有效集算法为10×max(设计变量的个数，不等式约束的个数+边界约束的个数)
TolFun	函数计算终止的误差限，对于大型规模算法其默认值为1e-8，该控制参数对于中型规模的有效集算法无效。

线性规划问题的MATLAB求解

❖ **输出参数**

linprog函数返回的输出参数有x、fval、exitflag、lambda和output。

- 输出参数x为线性规划问题的最优解
- 输出参数fval为线性规划问题在最优解x处的函数值
- 输出参数exitflag返回的是优化函数计算终止时的状态指示，说明算法终止的原因，其取值和其代表的具体原因如表所示

值	物理意义
1	已经收敛到解x
0	已经达到最大迭代次数限制options.MaxIter
-2	没有找到问题的可行点
-3	问题无有限最优解
-4	在算法执行过程中遇到了NaN值
-5	原线性规划问题和其对偶问题均不可行
-7	搜索方向变化太小，无法进一步获得更优解，说明原线性规划问题或者约束条件是病态的

线性规划问题的MATLAB求解

❖ 输出参数

- 输出参数output是一个返回优化过程中相关信息的结构变量，其属性如表所示

属性名称	属性含义
output.iterations	优化过程的实际迭代次数
output.algorithm	优化过程中所采用的具体算法
output.cgiterations	0（仅用于大型规模算法，为了后向兼容性而设置的参数）
output.message	退出信息

- 输出参数lambda是返回线性规划问题最优解x处的拉格朗日乘子的一个结构变量,其总维数等于约束条件的个数，其非零分量对应于起作用的约束条件，其属性如表所示。

属性名称	属性含义
ineqlin	线性不等式约束的拉格朗日乘子
eqlin	线性等式约束的拉格朗日乘子
upper	上界约束的拉格朗日乘子
lower	下界约束的拉格朗日乘子

线性规划问题的MATLAB求解

❖ 命令详解

- **x = linprog(c, A, b)**

  该函数调用格式求解线性规划问题

$$\begin{cases} \min & f = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \end{cases}$$

- **x = linprog(c, A, b, Aeq, beq)**

  该函数调用格式求解线性规划问题

$$\begin{cases} \min & f = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} \end{cases}$$

即该函数调用格式解决的是既含有线性等式约束，又含有线性不等式约束的线性规划问题，如果在线性规划问题中无线性不等式约束，则可以设A=[]以及b=[]

线性规划问题的MATLAB求解

❖ 命令详解

- **x = linprog(c, A, b, Aeq, beq, lb, ub)**

$$\begin{cases} \min & f = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} \\ & \mathbf{lb} \le \mathbf{x} \le \mathbf{ub} \end{cases}$$

该函数调用格式求解线性规划问题

即在线性规划问题的求解过程中进一步考虑了对设计变量的约束，其中lb和ub均是和设计变量维数相同的列向量，如果对设计变量没有上界约束，可以设置ub(i) = Inf，如果没有下界约束则可以设置lb(i) = -Inf，和(2)类似，如果问题中没有等式约束，则可以设Aeq=[]以及beq=[]

线性规划问题的MATLAB求解

❖ 命令详解

- x = linprog(c, A, b, Aeq, beq, lb, ub, x0)

  在前面调用方法的基础上设置线性规划问题求解的初始解为x0，该参数仅在使用有效集算法时生效，否则当使用默认的内点算法时，将忽略任何初始点，即参数无效。

- x = linprog(c, A, b, Aeq, beq, lb, ub, x0, options)

  用options指定的优化参数进行最小化。可使用optimset来设置这些参数

- [x, fval] = linprog(...)

  在优化计算结束之时返回线性规划问题在解x处的目标函数值fval。

- [x, fval, exitflag] = linprog(...)

  在优化计算结束之时返回exitflag值，描述函数计算的退出条件。

- [x, fval, exitflag, output] = linprog(...)

  在优化计算结束之时返回返回结构变量output

- [x, fval, exitflag, output, lambda] = linprog(...)

  在优化计算结束之时返回线性规划问题最优解x处的拉格朗日乘子lambda

线性规划问题的MATLAB求解

❖ 用MATLAB求解线性规划问题

$$\begin{cases} \max & f = x_1 + x_2 \\ \text{s.t.} & x_1 - 2x_2 \leq 4 \\ & x_1 + 2x_2 \leq 8 \\ & x_1, x_2 \geq 0 \end{cases}$$

```
c=[-1;-1];
        %目标函数，为转化为极小，取目标函数中设计变量的相反数
A=[1 -2;1 2];      %线性不等式约束
b=[4;8];
lb=[0;0]; %设计变量的边界约束，由于无上界，故设置ub=[Inf;Inf];
ub=[Inf;Inf];

[x,fval]=linprog(c, A, b, [], [], lb, ub)

Optimization terminated.
x =
   6.0000
   1.0000
fval =
  -7.0000
```

线性规划问题的MATLAB求解

$$\begin{cases} \max & f = 4x_1 + 3x_2 \\ & 3x_1 + 4x_2 \le 12 \\ & 3x_1 + 3x_2 \le 10 \\ & 4x_1 + 2x_2 \le 8 \\ & x_1, x_2 \ge 0 \end{cases}$$

```
c=[-4;-3];          %目标函数，为转化为极小，取目标函数中设计变量的相反数
A=[3 4;3 3;4 2];    %线性不等式约束
b=[12;10;8];
lb=[0;0]; %设计变量的边界约束，由于无上界，故设置ub=[Inf; Inf]
ub=[Inf;Inf];

[x,fval,exitflag]=linprog(c,A,b,[],[],lb,ub)

x =
   0.8000
   2.4000
fval =
  -10.4000
exitflag =
    1
```

$$
\begin{cases}
\max & f = x_1 + 3x_2 - x_3 \\
\text{s.t.} & x_1 + x_2 + 2x_3 = 4 \\
 & -x_1 + 2x_2 + x_3 = 4 \\
 & x_j \geq 0 \, (j = 1, 2, 3)
\end{cases}
$$

```
c=[-1;-3;1];            %目标函数，为转化为极小，取目标函数中设计变量的相反数
Aeq=[1 1 2;-1 2 1];              %线性等式约束
beq=[4;4];
lb=[0;0;0];                      %设计变量的边界约束，由于无上界，故设置
ub=[Inf;Inf;Inf]
ub=[Inf;Inf;Inf];

[x, fval, exitflag]=linprog(c, [], [], Aeq, beq, lb, ub)

Optimization terminated.
x =
    1.3333
    2.6667
    0.0000
fval =
   -9.3333
exitflag =
     1
```

$$
\begin{cases}
\max & f = 3x_1 - x_2 - x_3 \\
\text{s.t.} & x_1 - 2x_2 + x_3 \le 11 \\
 & -4x_1 + x_2 + 2x_3 \ge 3 \\
 & -2x_1 + x_3 = 1 \\
 & x_j \ge 0 \, (j = 1,2,3)
\end{cases}
$$

```
c=[-3;1;1];              %目标函数，为转化为极小，故取目标函数中设计变量的相反数
A=[1 -2 1;4 -1 -2];      %线性不等式约束
b=[11;-3];
Aeq=[-2 0 1];            %线性等式约束
beq=[1];
lb=[0;0;0];              %设计变量的边界约束，由于无上界，故设置ub=[Inf; Inf; Inf]
ub=[Inf;Inf;Inf];

[x, fval, exitflag, output, lambda]=linprog(c, A, b, Aeq, beq, lb, ub)
```

```
Optimization terminated.
x =
   4.0000
   1.0000
   9.0000
fval =
  -2.0000
exitflag =
     1
```

线性规划问题的MATLAB求解

❖ 生产计划问题—问题的提出

某工厂需要生产*A*、*B*两种产品以满足市场的需求。这两种产品的生产均需要经过两道工艺流程：

- 每生产1kg的*A*产品在第一道工艺流程耗时4小时，在第二道工艺流程耗时6小时；
- 每生产1kg的*B*产品在第一道工艺流程耗时6小时，在第二道工艺流程耗时8小时；
- 由于生产计划的要求，可供用的第一道工艺流程工时为240小时，第二道工艺流程工时为480小时。

在化学品生产的过程中一般会伴随着副产品的生产：

- 在生产B产品的同时，会产出副产品C，每生产1kg的B产品会产生2kg的副产品C，而不需外加任何费用，由于副产品C的利用率问题，使得产品C中的一部分可盈利，其他部分只能报废。

根据核算：

- 出售1kg的A产品可盈利600元，
- 出售1kg的B产品可以盈利1000元，
- 出售1kg的C产品可以盈利300元，而报废1kg的C产品需要亏损200元。
- 经市场预测，在计划期内，产品C最大销售量为50kg，

此时，应当如何安排A、B两种产品的产量，使该工厂的预计总盈利可以达到最大。

线性规划问题的MATLAB求解

❖ 生产计划问题—问题分析

在本例中，重点是设计变量的选取方法。因为副产品C的出现和限制销售量使得问题显得稍复杂。如果我们用x_1和x_2分别代表产品A和产品B的产量，作为该问题的设计变量，由于产品C的产量为$2x_2$，且如果$2x_2$大于50的话，其小于50的部分会产生盈利，但其超出的部分会产生亏损，即产品C的单位利润会在300和-200之间变化，在这个前提下，总利润和产量之间就产生了非线性关系，我们会发现在确定目标函数和约束条件时比较困难。于是，我们需要另辟蹊径，寻求设计变量的选择方法，解决这个问题。

在两个设计变量难以解决的前提下，我们可以设置多个设计变量，使得目标函数和约束条件均为线性。

线性规划问题的MATLAB求解

❖ **生产计划问题— 问题解答**

从产品*C*的约束出发，既然*C*产品可能产生盈利，也可能产生亏损，则设置相应的设计变量来表示其产生盈利的部分和产生亏损的部分，即产品*C*的销售量和报废量，故在这个原则下，我们得到问题的设计变量为：

产品*A*、*B*、*C*的产量：x_1、x_2、x_3 、

产品*C*的报废量：x_4

于是产品*C*的产量即为其销售量和报废量之和，即x_3+x_4将预计总盈利作为该问题的目标函数

$$\Rightarrow f=600x_1+1000x_2+300x_3-200x_4$$

*C*是伴随产品*B*出现的，其数量之间满足的约束关系

$$\Rightarrow x_3+x_4=2x_2$$

*C*的最大销量为50kg $\Rightarrow x_3 \le 50$

每道工艺流程的总工时是确定的，例如第一道工艺，生产$x_1$kg的*A*耗时$4x_1$小时，生产$x_2$kg的*B*要$6x_2$小时，其总和必须小于240小时；对第二道工艺流程的分析也一样

$$\Rightarrow \begin{aligned} 4x_1+6x_2 &\le 240 \\ 6x_1+8x_2 &\le 480 \end{aligned}$$

线性规划问题的MATLAB求解

❖ 生产计划问题——数学模型

$$\begin{cases} \max & f = 600x_1 + 1000x_2 + 300x_3 - 200x_4 \\ \text{s.t.} & -2x_2 + x_3 + x_4 = 0 \\ & x_3 \le 50 \\ & 2x_1 + 3x_2 \le 120 \\ & 3x_1 + 4x_2 \le 240 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{cases}$$

```
c=[-600;-1000;-300;200];
A=[2 3 0 0;3 4 0 0;0 0 1 0]; b=[120;240;50];
Aeq=[0 -2 1 1]; beq=[0];
lb=[0;0;0;0]; ub=[Inf;Inf;Inf;Inf];

[x,fval]=linprog(c,A,b,Aeq,beq,lb,ub)

Optimization terminated.
x =
   22.5000
   25.0000
   50.0000
    0.0000
fval =
 -5.3500e+004
```

由上述结果可知，当*A*产品的产量为22.5kg，*B*产品的产量为25kg时，该工厂预计盈利的最大值为5.35万元，其中伴随*B*产品产生的*C*产品恰好为50kg，为可销售的最大值。

线性规划问题的MATLAB求解

❖ 连续投资问题—问题的提出

某机构现在拥有资本200万元，为了获取更大的收益，该机构决定将这200万元进行投资，以期最大回报，现在共有四个方案可供选择，投资的方式为每年初将机构持有的所有资本都用于投资。

方案1：从第1年到第4年的每年年初都需要投资，次年末回收本利1.15

方案2：第3年初投资，到第5年末收回本利1.25，最大投资额为80万元

方案3：第2年初投资，到第5年末收回本利1.40，最大投资额为60万元

方案4：每年初投资，每年末收回本利1.06

那么应该采用何种投资组合策略，使得该机构5年末的总资本最大

线性规划问题的MATLAB求解

❖ 连续投资问题—问题分析

由于方案有4种可选，且每种开始投资的期限一般也不相同，故选择设计变量时需要考虑这两个因素，最直观的选法是令x_{ij}为第i年初投资方案j的资金数，此时，设计变量可以用下表来表示：

投资金额 \ 年份 / 方案	第1年	第2年	第3年	第4年	第5年
1	x_{11}	x_{21}	x_{31}	x_{41}	
2			x_{32}		
3		x_{23}			
4	x_{14}	x_{24}	x_{34}	x_{44}	x_{54}

线性规划问题的MATLAB求解

❖ 连续投资问题—问题解答

在第1年时，将所有的100万元用于投资，可选方案1和方案2

$$\Rightarrow x_{11} + x_{14} = 200$$

由于在第1年年末投资方案4的资金在第1年年末即收回本利1.06，故该部分资金即1.06x_{14}可用于第2年的投资，即

$$x_{21} + x_{23} + x_{24} = 1.06x_{14}$$

方案3的最大投资额为60万元 $\Rightarrow x_{23} \le 60$

可用于第3年投资的资本数来源于第1年投资方案1收回的本利1.15x_{11}和第2年投资方案4收回的本利1.11x_{24}，故

$$x_{31} + x_{32} + x_{34} = 1.15x_{11} + 1.06x_{24}$$

方案2的最大投资额为80万元 $\Rightarrow x_{32} \le 80$

可用于第4年投资的资本数来源于第2年投资方案1收回的本利1.15x_{21}和第3年投资方案4收回的本利1.11x_{34}，故

$$x_{41} + x_{44} = 1.15x_{21} + 1.06x_{34}$$

可用于第5年投资的资本数来源于第3年投资方案1收回的本利1.15x_{31}和第4年投资方案4收回的本利1.11x_{44}，故

$$x_{54} = 1.15x_{31} + 1.06x_{44}$$

线性规划问题的MATLAB求解

❖ 连续投资问题— 问题解答

通过上述连续投资方式，在第5年末可以获得的本利资本总和为：

$$f = 1.15x_{41} + 1.25x_{32} + 1.40x_{23} + 1.06x_{54}$$

我们的目的就是选择最佳的投资策略，极大化目标函数f的值

线性规划数学模型

$$\begin{cases}
\max & f = 1.40x_{23} + 1.25x_{32} + 1.15x_{41} + 1.06x_{54} \\
\text{s.t.} & x_{11} + x_{14} = 200 \\
 & 1.06x_{14} - x_{21} - x_{23} - x_{24} = 0 \\
 & 1.15x_{11} + 1.06x_{24} - x_{31} - x_{32} - x_{34} = 0 \\
 & 1.15x_{21} + 1.06x_{34} - x_{41} - x_{44} = 0 \\
 & 1.15x_{31} + 1.06x_{44} - x_{54} = 0 \\
 & x_{23} \le 60 \\
 & x_{32} \le 80 \\
 & x_{ij} \ge 0\ (i = 1,2,\ldots,5;\ j = 1,2,3,4)
\end{cases}$$

线性规划问题的MATLAB求解

❖ 连续投资问题—Matlab求解

```
c=[0;0;0;-1.40;0;0;-1.25;0;-1.15;0;-1.06];
Aeq=[1 1 0 0 0 0 0 0 0 0 0; 0 1.06 -1 -1 -1 0 0 0 0 0 0; 1.15 0 0 0 1.06 -1 -1 -1 0 0 0;
   0 0 1.15 0 0 0 0 1.06 -1 -1 0; 0 0 0 0 0 1.15 0 0 0 1.06 -1];
beq=[200;0;0;0;0];
lb=[0;0;0;0;0;0;0;0;0;0;0];
ub=[Inf;Inf;Inf;60;Inf;Inf;80;Inf;Inf;Inf;Inf];
[x,fval]=linprog(c,[],[],Aeq,beq,lb,ub)
```

```
Optimization terminated.
x =
  123.5459
   76.4541
   21.0414
   60.0000
    0.0000
   34.5138
   80.0000
   27.5639
   53.4154
    0.0000
   39.6909
fval =
  -287.5000
```

由运行结果可知，采取上述最佳投资方案之后，在第5年末所得到的总资本数为287.5万元。

线性规划问题的MATLAB求解

❖ 饲养场的配料问题

某饲养场有5种饲料．已知各种饲料的单位价格和每百公斤饲料的蛋白质、矿物质、维生素含量如表所示，又知该场每日至少需蛋白质70单位、矿物质3单位、维生素10毫单位．间如何混合调配这5种饲料．才能使总成本最低?

饲料的成分和单价

饲料种类	成分			饲料单价
	蛋白质/单位	矿物质/单位	维生素/毫单位	
1	0.30	0.10	0.05	2
2	2.20	0.05	0.10	7
3	1.00	0.02	0.02	4
4	0.60	0.20	0.20	3
5	1.80	0.05	0.08	5

线性规划问题的MATLAB求解

❖ 饲养场的配料问题

设第 i 种饲料的用量为 x_i（百公斤），i=1,2, … ,5，则对应的总成本（元）为：

$$f = 2x_1 + 7x_2 + 4x_3 + 3x_4 + 5x_5$$

要求蛋白质的总含量不少于 70 单位，则：$0.3x_1 + 2.2x_2 + x_3 + 0.6x_4 + 1.8x_5 \geq 70$

要求矿物质的总含量不少于 3 单位，则：$0.1x_1 + 0.05x_2 + 0.02x_3 + 0.2x_4 + 0.05x_5 \geq 3$

要求维生素的含量不少于 10 毫单位，则：$0.05x_1 + 0.1x_2 + 0.02x_3 + 0.2x_4 + 0.08x_5 \geq 10$

而且各饲料的用量应当满足非负约束，即：$x_j \geq 0\ (j = 1, 2, ..., 5)$

因而该配料问题的数学模型如下：

$$\begin{cases} \min & f = 2x_1 + 7x_2 + 4x_3 + 3x_4 + 5x_5 \\ \text{s.t.} & 0.3x_1 + 2.2x_2 + x_3 + 0.6x_4 + 1.8x_5 \geq 70 \\ & 0.1x_1 + 0.05x_2 + 0.02x_3 + 0.2x_4 + 0.05x_5 \geq 3 \\ & 0.05x_1 + 0.1x_2 + 0.02x_3 + 0.2x_4 + 0.08x_5 \geq 10 \\ & x_j \geq 0\ (j = 1, 2, ..., 5) \end{cases}$$

线性规划问题的MATLAB求解

❖ 饲养场的配料问题

```
c=[2;7;4;3;5];
A=[-0.3 -2.2 -1 -0.6 -1.8; -0.1 -0.05 -0.02 -0.2 -0.05; -0.05 -0.1 -0.02 -0.2 -0.08];
b=[-70;-3;-10];
lb=[0;0;0;0;0];
ub=[Inf;Inf;Inf;Inf;Inf];
[x,fval]=linprog(c,A,b,[],[],lb,ub)
Optimization terminated.
x =
    0.0000
    0.0000
    0.0000
   39.7436
   25.6410
fval =
  247.4359
```

由运行结果可知，只需要后两种饲料即可，其中饲料4需要3974.36公斤，饲料52564.1公斤，且此时的最低成本为247.4。

线性规划问题的MATLAB求解

回顾问题前面的例题，运输问题就是一个相当典型的线性规划问题

❖ 运输问题

- 设有两个建材厂C1和C2，每年沙石的产量分别为35万吨和55万吨，这些沙石需要供应到W1、W2和W3三个建筑工地，每个建筑工地对沙石的需求量分别为26万吨、38万吨和26万吨，各建材厂到建筑工地之间的运费（万元/万吨）如表所示，问题是应当怎么调运才能使得总运费最少？

运费 工地 / 建材厂	W_1	W_2	W_3
C_1	10	12	9
C_2	8	11	13

线性规划问题的MATLAB求解

❖ 运输问题

回顾问题前面的例题，这是一个相当典型的线性规划问题，

$$
\begin{cases}
\min & f = 10x_{11} + 12x_{12} + 9x_{13} + 8x_{21} + 11x_{22} + 13x_{23} \\
\text{s.t.} & x_{11} + x_{12} + x_{13} = 35 \\
 & x_{21} + x_{22} + x_{23} = 55 \\
 & x_{11} + x_{21} = 26 \\
 & x_{12} + x_{22} = 38 \\
 & x_{13} + x_{23} = 26 \\
 & x_{ij} \geq 0 \ \ (i = 1,2;\ j = 1,2,3)
\end{cases}
$$

线性规划问题的MATLAB求解

❖ 运输问题

```
c=[10;12;9;8;11;13];
  Aeq=[1 1 1 0 0 0;
       0 0 0 1 1 1
       1 0 0 1 0 0
       0 1 0 0 1 0
       0 0 1 0 0 1];
  beq=[35;55;26;38;26];
  lb=[0;0;0;0;0;0];
  ub=[Inf;Inf;Inf;Inf;Inf;Inf];
  [x,fval]=linprog(c,[],[],Aeq,beq,lb,ub)
```

```
Optimization terminated.
  x =
     0.0000
     9.0000
    26.0000
    26.0000
    29.0000
     0.0000
  fval =
   869.0000
```

线性规划问题的MATLAB求解

❖ 绝对值问题

- 问题的提出

求解如下优化问题
$$\begin{cases} \min & f = |x| + |y| + |z| \\ \text{s.t.} & x + y \le 1 \\ & 2x + z = 3 \end{cases}$$

- 问题分析

乍一看，该问题并非线性规划问题，因为目标函数中含有变量的绝对值，并非线性的问题，于是我们需要将上述问题转化成我们可以求解的线性规划问题。

事实上，如果我们设计两个与x相关的非负变量m和n，使其满足
$$m = \frac{x + |x|}{2}, \quad n = \frac{|x| - x}{2}$$

根据以上两个式子，我们可以得到： $x = m - n, \quad |x| = m + n$

于是，我们同上如上的代换方式，就可以将该问题顺利转化为线性规划来求解。

线性规划问题的MATLAB求解

❖ 绝对值问题

- 问题解答

根据以上的分析，我们作如下代换

$$|x| = x_1 + x_2, \quad x = x_1 - x_2$$

$$|y| = x_3 + x_4, \quad y = x_3 - x_4$$

$$|z| = x_5 + x_6, \quad y = x_5 - x_6$$

于是问题变为：

$$\begin{cases} \min & f = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\ \text{s.t.} & x_1 - x_2 + x_3 - x_4 \le 1 \\ & 2x_1 - 2x_2 + x_5 - x_6 = 3 \\ & x_j \ge 0 \; (j = 0,1,\ldots 6) \end{cases}$$

线性规划问题的MATLAB求解

❖ 绝对值问题

```
c=[1;1;1;1;1;1];
A=[1 -1 1 -1 0 0];
b=[1];
Aeq=[2 -2 0 0 1 -1];
beq=[3];
lb=[0;0;0;0;0;0];
ub=[Inf;Inf;Inf;Inf;Inf;Inf];
[x,fval]=linprog(c,A,b,Aeq,beq,lb,ub)
```

```
Optimization terminated.
x =
    1.0936
    0.0000
    0.0000
    0.0936
    0.8129
    0.0000
fval =
    2.0000
```

8.1 最优化问题概述

8.2 线性规划问题

8.3 整数规划问题

8.4 其他常见优化问题简介

8.5 现代智能优化算法简介

- 什么是整数规划

　　线性规划问题，其设计变量都是连续变量，其最优解可以出现分数或小数。但当设计变量表示零件数、劳动力人数、设备的台数时，其取值则必须为非负整数，此时一般的线性规划求解方法就可能失效。我们称要求设计变量的部分分量或者全部分量取整数值的最优化问题为整数规划问题

- 整数规划的分类
  - 纯整数规划：全部设计变量都取整数
  - 混合整数规划：部分设计变量取整数
  - 0-1规划：设计变量仅取0或1两个值

典型的整数规划问题

❖ 装载问题

有一列用于运货的火车，其最大承载能力为b。现有n种不同的货物p_1, p_2, …, p_n可供装载，设每件p_i的重量为a_i，装载收费为c_i $(i=1,2,\ldots,n)$，则应采用何种装载方案能够使得该列火车载货的收入最大？

设x_j为列车上装载p_j的数量，则x_j必为非负整数，根据该货船最大可承载b吨货物可知所有集装箱的重量之和必须b，故有约束条件：

$$\sum_{j=1}^{n} a_j x_j \leq b$$

由对每个j种货物收费为c_j，可知载货的总收入为：

$$f = \sum_{j=1}^{n} c_j x_j$$

该例的目标即使得目标函数f最大化。综合上述分析可得如下整数规划问题：

$$\begin{cases} \max \quad f = \displaystyle\sum_{i=1}^{n} c_j x_j \\ \text{s.t.} \quad \displaystyle\sum_{j=1}^{n} a_j x_j \leq b \\ \qquad\quad x_j \geq 0, \text{且取整数值} \ (j=1,2,\ldots,n) \end{cases}$$

典型的整数规划问题

❖ 工厂选址问题

某地区有m座铁矿$A_1,A_2,\ldots,A_m$，A_i每年的产量为a_i ($i=1,2,\ldots,m$)，该地区已有一个钢铁厂B_0，每年铁的用量为p_0，每年固定运营费用为r_0。由于当地经济的发展，政府拟建立一个新的钢铁厂，于是今后该地区的m座铁矿将全部用于支持这两个钢铁厂的生产运营。现在有n个备选的厂址，分别为$B_1, B_2, \ldots, B_n$，若在B_j ($j=1,2,\ldots,n$)处建厂，则每年固定的运营费用为r_j。由A_i向B_j每运送1t钢铁的运输费用为c_{ij} ($i=1,2,\ldots,m; j=0,1,\ldots,n$)。那么应当如何选择新厂厂址，铁矿所开采出来的铁矿石又当如何分配给两个钢铁厂，才能使每年的总费用（固定运营费用和煤的运费）最低？

钢铁厂B_0每年需要用铁p_0，而且今后该地区m座铁矿将全部用于支持这两个钢铁厂的生产，故新的钢铁厂每年用铁量p为该m座铁矿的总产量减去B_0的用铁量：

$$p=\sum_{i=1}^{m} a_i - p_0$$

令设计变量为v_i，若$v_i=1$则表示选择B_i作为新厂厂址，否则$v_i=0$：

$$v_i=\begin{cases}1 & B_i\text{作为新厂厂址}\\ 0 & B_i\text{不作为新厂厂址}\end{cases}\quad (i=1,2,\ldots,n)$$

典型的整数规划问题

❖ 工厂选址问题

设x_{ij}为每年从Ai运往Bj的钢铁数量，于是每年的总费用为：

$$f = \sum_{i=1}^{m}\sum_{j=0}^{n} c_{ij}x_{ij} + \sum_{j=1}^{n} r_j v_j + v_0$$

由铁矿Ai运出的所有钢铁将等于铁矿A_i的产量a_i $\Rightarrow \sum_{j=0}^{n} x_{ij} = a_i \ (i = 1,2,\ldots,m)$

原钢铁厂B_0钢铁的用量p_0为m座铁矿为其供应，故其收量应当等于m座铁矿对分别对其供应量的总和，即：

$$\sum_{i=1}^{m} x_{i0} = p_0$$

同样的，对于备选厂B_j，可知其钢铁的用量为p，且由m座铁矿供应，由于备选厂址只有一座，故在p前面需要乘以系数v_j，即代表如果选择B_j为备选厂址，则用铁矿，否则，则该厂不存在，不需要使用铁矿，此时，对应的x_{ij}将全部取零值，故：

$$\sum_{i=1}^{m} x_{ij} = pv_j \ (j = 1,2,\ldots,n)$$

由A_i向B_j钢铁的运输量均为非负实数 $\Rightarrow x_{ij} \geq 0 \ (i = 1,2,\ldots,m;\ j = 0,1,\ldots,n)$

备选的钢铁厂只有一处

$$\Rightarrow \sum_{j=1}^{n} v_j = 1$$

典型的整数规划问题

❖ 工厂选址问题

根据如上分析，根据设计变量的取值规则，要么建厂取0，要么不建厂取1，同时该问题还要确定如果选择了厂址，应当如何分配m座铁矿对两个钢铁厂的钢铁供应量x_{ij}，而该变量的取值为非负实数即可，故该问题为一混合整数规划问题，且为混合0-1规划，可以归纳为如下形式：

$$
\begin{cases}
\min \quad f = \sum\limits_{i=1}^{m}\sum\limits_{j=0}^{n} c_{ij}x_{ij} + \sum\limits_{j=1}^{n} r_j v_j + v_0 \\
\text{s.t.} \quad \sum\limits_{j=0}^{n} x_{ij} = a_i \ (i = 1,2,\ldots,m) \\
\qquad \sum\limits_{i=1}^{m} x_{i0} = p_0 \\
\qquad \sum\limits_{i=1}^{m} x_{ij} = p v_j \ (j = 1,2,\ldots,n) \\
\qquad \sum\limits_{j=1}^{n} v_j = 1 \ (v_j\text{取0或1}) \\
\qquad x_{ij} \geq 0 \ (i = 1,2,\ldots,m;\ j = 0,1,\ldots,n)
\end{cases}
$$

典型的整数规划问题

❖ 背包问题

夫妇两人要赴*A*地进行长途旅行，需要整理行李，现有3个旅行包，其容积大小分别为10升、15升和20升，两人在列出物品清单后根据需要已经整理出了10个包装袋，其中一些包装袋中装的是必带物品，共有7件，其体积大小分别为4升、3升、1.5升、2.5升、4.5升、7.6升和1.9升。尚有8个包装袋可带可不带，不带则可在*A*地购买，这些可选包装袋的容积和其对应物品在*A*地的价格如表所示。

物品	1	2	3	4	5	6	7	8
容积	2.5	4	5.5	4.8	3.7	1.6	7.5	4.5
价格	20	50	105	75	55	80	200	100

试根据上述信息给出一个合理的打包方案。

在这个问题中，我们需要确定的是，选带哪几个可选的包装袋，且将必带物品和选带物品放到哪个旅行包中。为此我们设第*i*个包装袋是否放在第*j*个旅行包中，并以此作为设计变量。同时，设第*i*个包装袋的容积可以用w_i (i=1,2,3,…,15)来表示，可选包装袋对应的价格用p_i (i=8,9,10,…,15)来表示。由于第*i*个包装袋要么在第*j*个旅行包中，要么在要么不在，故设只取0和1，且表述如下：

$$x_{ij}=\begin{cases}1 & \text{包装袋}i\text{在旅行包}j\text{中}\\ 0 & \text{包装袋}i\text{不在旅行包}j\text{中}\end{cases}\qquad (i=1,2,\ldots,15;\ j=1,2,3)$$

典型的整数规划问题

❖ 背包问题

由于每个旅行包的容积确定，故装入第j个旅行包中的所有包装袋的容积的总和必须小于第j个旅行包的容积，即需要满足约束条件：

$$\sum_{i=1}^{15} w_i x_{ij} \leq r_j \ (j = 1,2,3)$$

由于旅行袋中有7件为必带，故这7个包装袋必然在3个旅行包中的其一，设包装袋的编号为i，则在设计变量x_{i1}、x_{i2}和x_{i3}中必有一个取值为1，另外两个取值为0，其和为1。根据上述分析，对于7件必带的包装袋必须满足如下约束：

$$\sum_{j=1}^{3} x_{ij} = 1 \, (i = 1,2,\ldots,7)$$

对于可选的包装袋，则其要么在某个旅行包中，要么不在旅行包中，设包装袋的编号为i，如果它在某个旅行包中，则设计变量x_{i1}、x_{i2}和x_{i3}取值之和为1，如果它不在旅行包之中，则设计变量x_{i1}、x_{i2}和x_{i3}取值之和为0，故对于可选的包装袋必须满足如下约束：

$$\sum_{j=1}^{3} x_{ij} \leq 1 \, (i = 8,2\ldots,15)$$

典型的整数规划问题

❖ 背包问题

我们的目标就是使得在到达A地之后，所买物品价格最低，即不在旅行包中的包装袋的总价格最低，如果某包装袋i不在旅行包中，则有：$\sum_{j=1}^{3} x_{ij} = 0$ 故其价格可以用 $p_i\left(1-\sum_{j=1}^{3} x_{ij}\right)$来表示，故所有不在旅行包中的包装袋的价值$f$可表达如下

$$f = \sum_{i=8}^{15} p_i\left(1-\sum_{j=1}^{3} x_{ij}\right)$$

我们确定打包方案的原则就是使得f取得最小值，故综合以上分析，该背包问题的数学模型为：

$$\begin{cases}
\min \quad f = \sum_{i=8}^{15} p_i\left(1-\sum_{j=1}^{3} x_{ij}\right) \\
\text{s.t.} \quad \sum_{i=1}^{15} w_i x_{ij} \le r_j \ (j = 1,2,3) \\
\qquad \sum_{j=1}^{3} x_{ij} = 1 \ (i = 1,2,\ldots,7) \\
\qquad \sum_{j=1}^{3} x_{ij} \le 1 \ (i = 8,2\ldots,15) \\
\qquad x_{ij} = 1,0 \ (i = 1,2\ldots,15, j = 1,2,3)
\end{cases}$$

整数规划问题的数学模型

❖ 一般形式

在整数规划中还有许多其他典型的问题，例如在第3章中提到的分派问题，还有旅行商问题、下料问题等，其问题均可以归结为如下的一般形式

$$
\begin{cases}
\max \quad f = \mathbf{cx} & (1) \\
\text{s.t.} \quad \mathbf{Ax} = \mathbf{b} & (2) \\
\qquad \mathbf{x} \geq 0, \text{且全部或者部分取整数值} & (3)
\end{cases}
$$

上述形式是仿照线性规划中的标准型给出的，其中设计变量x为n维的列向量，c为n维的行向量，A为$m\times n$的矩阵，且A行满秩，b为m维列向量。

在模型中，(1)可以是最大化也可以是最小化，对于约束(2)，可以是等式的形式也可以是不等式的形式。对于对设计变量的约束(3)，如果要求x全部分量为整数，则为纯整数规划；如果要求x的部分分量为整数，则为混合整数规划，如果要求x分量的取值只能为0和1，则为0-1规划。

0-1规划的MATLAB求解

❖ 0-1规划问题的MATLAB标准型

$$\begin{cases} \min & f = \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} \\ & \mathbf{x} = 0,1 \end{cases}$$

- 在上述模型中，目标函数f需要极小化，不等式约束形式为"≤"。假设x为n维设计变量，且线性规划问题具有不等式约束m_1个，等式约束m_2个，那么：c、x均为n维列向量，b为m_1维列向量，b_{eq}为m_2维列向量，A为$m_1 \times n$维矩阵，A_{eq}为$m_2 \times n$维矩阵。
- 如果不满足标准型的要求，则需要对原问题进行转化，化为标准型之后才能使用相关函数，标准化的方法和线性规划中的类似

❖ 0-1规划问题的MATLAB求解函数

- MATLAB优化工具箱中求解0-1规划问题的命令为bintprog

0-1规划的MATLAB求解

❖ bintprog的调用格式

```
x = bintprog(f)
x = bintprog(f, A, b)
x = bintprog(f, A, b, Aeq, beq)
x = bintprog(f, A, b, Aeq, beq, x0)
x = bintprog(f, A, b, Aeq, Beq, x0, options)
[x, fval] = bintprog(...)
[x, fval, exitflag] = bintprog(...)
[x, fval, exitflag, output] = bintprog(...)
```

❖ 输入参数

- MATLAB工具箱中的bintprog函数在求0-1规划问题时，提供的参数有如下几种
- 模型参数： x、c、b、b*eq*、A和A*eq*
- 初始解参数：x0
- 算法控制参数： options，我们可以通过optimset命令对这些具体的控制参数进行设置，其中主要参数的设置方法如下一页的表格所示

0-1规划的MATLAB求解

bintprog中的控制参数设置

参数名称	参数设置
BranchStrategy	设置算法中分枝变量的选择策略，当该参数值为mininfeas时，选择最可能为整数的变量进行分枝，即分枝变量最接近0或1，但不等于0或1；当该参数值为maxinfeas(默认)时，选择最不可能为整数的变量进行分枝，即分枝变量最接近0.5
MaxIter	设置算法运行中的最大迭代次数，默认值为100000*设计变量的个数
MaxNodes	设置算法搜索的最大节点数，默认值为1000*设计变量的个数
MaxRLPIter	设置算法在求解各个节点的松弛线性规划问题时的最大迭代次数，默认值为100*设计变量的个数
MaxTime	设置算法运行的最大CPU时间，以秒为单位，默认值为7200s
NodeDisplayInterval	设置节点显示区间。即在每次显示迭代报告之前搜索节点的数目。默认值为20
NodeSearchStrategy	设置算法中分枝节点的选择策略，当该参数值为df时，为深度优先搜索，即选择最下层的孩子节点进行分枝；当该参数值为bn（默认）时，为广度优先搜索，即选择目标函数值最优的节点进行分枝
TolFun	函数计算终止的误差限，其默认值为1e-3
TolXInteger	设置判断一个数值是否为正整数的误差限，默认值为1.0e-8，即如果一个数和与其最邻近的正整数之差小于1.0e-8，则被认为是该正整数
TolRLPFun	设置求解松弛线性规划问题的目标函数计算终止误差限，默认值为 1.0e-6
Diagnostics	设置是否显示函数优化中的诊断信息，可以选择on或者off（默认值），该功能主要显示一些退出信息，即bintprog函数运算终止的原因
Display	设置显示信息的级别，当该参数值为off时，不显示任何输出信息；当参数值为iter时，将显示每一步迭代的输出信息，iter参数值仅对大型规模算法和中型规模的单纯形算法有效；当参数值为final时，仅显示最终的输出信息

0-1规划的MATLAB求解

❖ 输出参数

- x为0-1规划问题的最优解
- fval为0-1规划问题在最优解x处的函数值
- exitflag返回的是bintprog计算终止时的状态指示，说明算法终止的原因，其取值和其代表的具体原因如表所示
- 输出参数output是一个返回优化过程中相关信息的结构变量，它所包含的属性及属性代表的意义如表所示。

参数exitflag的物理意义

值	物理意义
1	已经收敛到解x
0	已经达到最大迭代次数限制options.MaxIter
-2	优化问题无可行解
-4	搜索节点数超过设置的最大节点数
-5	搜索时间超过设置的最大CPU时间options.MaxTime.
-6	在求解某节点的线性松弛问题时进行迭代的次数超过算法设置的在求解各个节点的松弛线性规划问题时的最大迭代次数options.MaxRLP

参数output所包含的信息

属性名称	属性含义
output.iterations	优化过程的实际迭代次数
output.algorithm	优化过程中所采用的具体算法
output.nodes	优化过程中搜索过的节点数目
output.time	优化过程中执行算法消耗的CPU时间
output.branchStrategy	优化过程中选择分枝变量的策略
output.nodeSearchStrategy	优化过程中选择分枝节点的策略
output.message	退出信息

0-1规划的MATLAB求解

❖ 命令详解

- **x = bintprog(f)**

  该函数调用格式求解如下形式的0-1规划问题

$$\begin{cases} \min & f = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{x} = 0,1 \end{cases}$$

- **x = bintprog(c, A, b)**

  该函数调用格式求解如下形式的0-1规划问题

$$\begin{cases} \min & f = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \mathbf{x} = 0,1 \end{cases}$$

- **x = bintprog (c, A, b, Aeq, beq)**

  该函数调用格式求解如下形式的0-1规划问题

$$\begin{cases} \min & f = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} \\ & \mathbf{x} = 0,1 \end{cases}$$

- **x = bintprog (c, A, b, Aeq, beq, x0)**

  在前一个调用格式的基础上同时设置求解算法的初始解为x0，如果初始解x0不在0-1规划问题的可行域中，算法将采用默认的初始解

- **x = bintprog (c, A, b, Aeq, beq, x0, options)**

  用options指定的优化参数进行最小化。可以使用optimset来设置这些参数

0-1规划的MATLAB求解

❖ 命令详解（续）

- [x, fval] = bintprog(...)

  在优化计算结束之时返回整数规划问题在解x处的目标函数值fval
- [x, fval, exitflag] = bintprog(...)

  在优化计算结束之时返回exitflag值，描述函数计算的退出条件。
- [x, fval, exitflag, output] = bintprog(...)

  在优化计算结束之时返回结构变量output

应用实例

❖ **求解0-1规划问题**

$$
\begin{cases}
\max & f = 20x_1 + 6x_2 + 8x_3 + 9x_4 \\
\text{s.t.} & 10x_1 + 6x_2 + 5x_3 + 2x_4 \le 19 \\
 & 7x_1 + 2x_2 + 2x_3 + 4x_4 \le 11 \\
 & 2x_1 + x_2 + x_3 + 10x_4 \le 12 \\
 & x_4 \le x_2 + x_3 \\
 & x_i = 0\text{或}1\ (i = 1,2,3,4)
\end{cases}
$$

```
c=[20; 6; 8; 9];
A=[-10 -6 -5  -2; -7 -2 -2  -4; -2 -1 -1 -10; 0  1  1  -1];b=[-4; -4; -2; 1];
lb=[0; 0; 0; 0];ub=[1; 1; 1; 1];
M=1:4;                                  %均要求为整数变量
Tol=1e-8;                               %判断是否整数的误差限
[x1, fval]=bintprog(c, A, b)            %调用bintprog函数
```

```
Optimization terminated.
x1 =
     0
     1
     0
     1
fval =
    15
```

用bintprog函数求解0-1规划的结果

应用实例

❖ 旅行者的背包问题

- 问题的提出

　　一个旅行者为了出行需要整理其行李，但是他的背包最多只能承受25kg的物品。该旅行者共有6件物品可供携带，物品的对应编号、重量及其价值如表所示，由于背包的限制，该旅行者决定携带尽可能高价值的物品，那么他应该如何选择可使携带物品的总价值最大？

物品编号	1	2	3	4	5	6
重量（千克）	4.8	6.2	5.7	3.6	4.4	8.5
价值（元）	120	180	150	100	90	230

应用实例

❖ 旅行者的背包问题

- 问题分析

设w_j为物品j的重量，p_j为物品j的价值。选取设计变量为x_j (j=1,2,…,10)，x_j=1表示携带物品j，x_j=0表示不携带物品j，则所带物品的总重量为：

$$w = \sum_{i=1}^{6} w_j x_j = 4.8x_1 + 6.2x_2 + 5.7x_3 + 3.6x_4 + 4.4x_5 + 8.5x_6$$

由于背包最多能承受25kg的物品，故必须满足$W \leq 25$：

$$4.8x_1 + 6.2x_2 + 5.7x_3 + 3.6x_4 + 4.4x_5 + 8.5x_6 \leq 25$$

同时可根据每件物品的价值来计算带物品的总价值为

$$p = \sum_{i=1}^{6} p_j x_j = 120x_1 + 180x_2 + 150x_3 + 100x_4 + 90x_5 + 230x_6$$

应用实例

❖ 旅行者的背包问题

- 数学模型

该问题是要使得旅行者携带物品的总价值最大，即将*p*的值极大化，于是我们得到该问题的数学模型为：

$$\begin{cases} \max \quad f = 120x_1 + 180x_2 + 150x_3 + 100x_4 + 90x_5 + 230x_6 \\ \text{s.t.} \quad 4.8x_1 + 6.2x_2 + 5.7x_3 + 3.6x_4 + 4.4x_5 + 8.5x_6 \le 25 \\ \qquad x_i = 0,1 \ \ (i = 1,2,\ldots,6) \end{cases}$$

这是一个0-1规划模型。可以用MATLAB中bintprog函数求解，

应用实例

❖ 旅行者的背包问题

- 代码和结果

```
c=[-120; -180; -150; -100; -90; -230];
A=[4.8 6.2 5.7 3.6 4.4 8.5];
b=[25];
[x, fval, exitflag, output]=bintprog(c, A, b)
```

```
Optimization terminated.
x =
    0
    1
    1
    1
    0
    1
fval =
  -660
exitflag =
    1
```

根据上述结果，当所选择的物品的序号为2、3、4、6时，相应物品的价值之和为660元。

8.1 最优化问题概述

8.2 线性规划问题

8.3 整数规划问题

8.4 其他常见优化问题简介

- 非线性规划
- 多目标规划

8.5 现代智能优化算法简介

非线性规划

❖ 什么是非线性规划

- 线性规划问题和整数规划问题，其共同的特征是最优化问题中的目标函数和约束条件均为设计变量的线性函数。但在实际建模过程中还有大量的问题，其目标函数或约束条件很难用线性函数来表达，当目标函数或约束条件中有一个以上是非线性函数时，就不能用线性的方法来处理，而要采用非线性的方法，那么我们称这种问题为非线性规划问题

非线性规划

❖ 非线性规划的产生和发展

- 自从1951年H. W. Kuhn及A. W. Tucker探讨了非线性规划解的最优性条件，为非线性规划奠定了理论基础之后，非线性规划逐渐形成了一门十分重要且比较活跃的新兴学科，出现了许多解非线性规划问题的有效的算法。由70年代开始，该分支得到迅速发展：理论方面，非线性规划借鉴了数学理论中其他分支的成果，逐步形成自身的学科特色；在应用方面，非线性规划为系统的优化和管理提供了有力的工具。
- 随着电子计算机的应用，非线性规划在最优设计、管理科学、质量控制等许多领域得到越来越深入的应用。非线性规划发展到今天，虽然已经提出许多求解方法和策略，但是对于非线性规划的最优化问题目前还没有适于各种不同情况的一般算法，各个方法都有自己特定的适用范围。因而，这是需要人们更深入的进行研究的一个领域。

非线性规划—典型问题

❖ 选址问题

- 问题的提出

一家大型连锁超市在某地有家分店 $A_i\ (i=1,2,\ldots,n)$ ，数学语言描述可在平面直角坐标系给出其位置表述：A_1的坐标为(x_1,y_1)，A_2的坐标为(x_2,y_2)，以此类推，A_n的坐标为(x_n,y_n)。

现在超市拟在当地选择一个理想的位置建立一个供货点，由于该超市各分店在经营规模上的不同，出货量也不同，导致供货点对各分店的送货频率不同，假设供货点每周给A_i送货的次数为$c_i(i=1,2,\ldots,n)$，同时假设每公里的运输费保持定值m元/公里。那么超市应当把供货点设在什么地方可以使得运输成本最低？

非线性规划—典型问题

❖ 选址问题

- 问题分析

假设供货点坐标为(x,y)，那么由供货点到某分店A_i的距离和运输费分别为：

$$s_i = \sqrt{(x-x_i)^2 + (y-y_i)^2}$$

$$f_i = c_i m \sqrt{(x-x_i)^2 + (y-y_i)^2}$$

故运输的总成本为对各个分店运输成本的总和，整个问题的数学模型可以表达为：

$$\min \ f = \sum_{i=1}^{n} f_i = \sum_{i=1}^{n} c_i m \sqrt{(x-x_i)^2 + (y-y_i)^2}$$

上述问题的目标函数为设计变量x和y的非线性函数，故为非线性规划问题，由于设计变量x和y不受任何条件的约束，故为无约束非线性规划。

非线性规划—典型问题

❖ 营业计划制定问题

- 问题的提出

某公司销售两种建材A和B以满足市场的需要，生产建材A和B均要消耗资原材料M和N，其中每吨A建材需要消耗10吨M和18吨N，每吨B需要消耗20吨M和12吨N，已知产品的利润是销售量的函数，现有原材料200吨M和100吨N，产品的售价、和资源的对应关系如表所示，该公司应当如何制定生产销售计划使得销售额最大。

	消耗资源*M*(吨)	消耗资源*N*(吨)	单位售价(万元/吨)
A	1	1.8	p_1=5-0.01x_1
B	2	1.2	p_2=6-0.03x_2
资源限制	200	100	

非线性规划—典型问题

❖ 营业计划制定问题

- 问题分析

设x_1和x_2为产品A和产品B的销售量，由A的单价为p_1，B的单价为p_2，则可知公司的销售收入为 $f = p_1x_1 + p_2x_2$ ，在该例中，单位售价为销量的函数，故由表中的公式可得 $f = (5 - 0.01x_1)x_1 + (6 - 0.03x_2)x_2$ 最优化的目标为使得销售额最大，即取得最大值。由原材料的限制，可得约束条件为：

$$x_1 + 2x_2 \le 200$$

$$1.8x_1 + 1.2x_2 \le 100$$

且考虑到和的产量应当为非负实数，故该问题的数学模型为：

$$\begin{cases} \max & f = (5 - 0.01x_1)x_1 + (6 - 0.03x_2)x_2 \\ \text{s.t.} & x_1 + 2x_2 \le 200 \\ & 1.8x_1 + 1.2x_2 \le 100 \\ & x_1, x_2 \ge 0 \end{cases}$$

上述问题的目标函数为设计变量x_1和x_2的非线性函数，约束条件为设计变量的x_1和x_2的线性函数，为有约束的非线性规划问题。

非线性规划—Matlab函数

一维搜索问题：

fminbnd

多维无约束优化问题：

fminunc

fminsearch

多维约束优化：

fmincon

多目标规划

❖ 什么是多目标规划问题

- 在前面所述的最优化问题，无论是线性规划、整数规划还是非线性规划，其目标函数都只有一个。但在实际问题中，衡量一个设计方案的好坏往往不止一个标准，常常要考虑多个目标。例如研究生产过程时，人们既要提高生产效率，同时还要考虑产品质量，又要考虑成本以降低生产费用，可能还希望生产过程中的环保问题，即废渣、废水、废气造成的污染小。在设计导弹的过程中，既要射程远，又要燃料省，还要重量轻且打击精度高。在进行投资决策时，既希望回报高的同时又希望降低投资风险，如此等等。这就向我们提出了一个多指标最优化问题。我们把在这样的背景下建立起来的最优化称之为多目标规划问题。

❖ 多目标规划问题的发展

- 多目标规划法（Goal Programming，简称GP）也是最优化理论和方法中的一个重要分支，它是在线性规划的基础上，为解决多目标决策问题而发展起来的一种数学方法。其概念和数学模型是由A.Charnes和W.W.Cooper在1961年提出的，经过Ijiri,Sang.M.Lee等人的改进，并逐步发展和成熟，它在经济管理与规划、人力资源管理、政府管理、大型工程的最优化等重要问题上都有广泛的应用。

多目标规划— 典型问题

❖ 木梁设计问题

用直径为 1（单位长）的圆木制成截面为矩形的梁。为使重量最轻面强度最大，问截面的宽和高应取何尺寸？

假设矩形截面的宽和高分别为 x_1 和 x_2，那么根据几何知识可得：

$$x_1^2 + x_2^2 = 1$$

且此时木梁的截面面积为 x_1x_2。同时根据材料力学的知识，木梁的强度取决于截面矩量 $\frac{1}{6}x_1x_2^2$，故若要使得重量最轻，实际上目标即为横截面积最小，又要强度最大，故目标为截面矩量最大，于是容易列出如下数学模型：

$$\begin{cases} \min & f_1(\mathbf{x}) = x_1x_2 \\ \max & f_2(\mathbf{x}) = \dfrac{1}{6}x_1x_2^2 \\ & x_1^2 + x_2^2 = 1 \\ & x_1, x_2 \geq 0 \end{cases}$$

多目标规划— 典型实例

❖ 工厂采购问题

某工厂需要采购某种生产原料，该原料市场上有 A 和 B 两种，单价分别为 2 元/kg 和 1.5 元/kg。现要求所花的总费用不超过 300 元，购得的原料总重量不少于 120kg，其中 A 原料不得少于 60kg。问如何确定最佳采购方案，花最少的钱，采购最多数量的原料。

设 A、B 两种原料分别采购 x_1 、x_2 kg，那么总的花费为： $f_1(\mathbf{x}) = 2x_1 + 1.5x_2$

购得的原料总量为： $f_2(\mathbf{x}) = x_1 + x_2$

那么我们求解的目标即是使得花最少的钱买最多的原料，即最小化 $f_1(\mathbf{x})$ 的同时极大化 $f_2(\mathbf{x})$ 。

多目标规划— 典型实例

同时要满足所花的总费用不得超过 300 元，原料的总重量不得少于 120kg，A 原料不得少于 60kg，于是得到约束条件如下：

$$x_1 + x_2 \geq 120$$

$$2x_1 + 1.5x_2 \leq 300$$

$$x_1 \geq 60$$

又考虑到购买的数量必须要满足非负的条件，由于对 x_1 已经有相应的约束条件，故只需添加对 x_2 的非负约束即可。

综合以上分析，得到最优化数学模型如下：

$$\begin{cases} \min \quad f_1(\mathbf{x}) = 2x_1 + 1.5x_2 \\ \max \quad f_2(\mathbf{x}) = x_1 + x_2 \\ \qquad x_1 + x_2 \geq 120 \\ \qquad 2x_1 + 1.5x_2 \leq 300 \\ \qquad x_1 \geq 60 \\ \qquad x_2 \geq 0 \end{cases}$$

fgoalattain函数

8.1 最优化问题概述

8.2 线性规划问题

8.3 整数规划问题

8.4 其他常见优化问题简介

8.5 现代智能优化算法简介

- 遗传算法
- 模拟退火算法

遗传算法

遗传算法（Genetic Algorithm）是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。

由美国的J.Holland教授1975年首先提出，其主要特点是直接对结构对象进行操作，不存在求导和函数连续性的限定；具有内在的隐并行性和更好的全局寻优能力；采用概率化的寻优方法，能自动获取和指导优化的搜索空间，自适应地调整搜索方向，不需要确定的规则。遗传算法的这些性质，已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域。它是现代有关智能计算中的关键技术。

遗传算法— 标准过程

- **建初始状态**

　　初始种群是从解中随机选择出来的，将这些解比喻为染色体或基因，该种群被称为第一代，这和符号人工智能系统的情况不一样，在那里问题的初始状态已经给定了。

- **评估适应度**

　　对每一个解(染色体)指定一个适应度的值，根据问题求解的实际接近程度来指定(以便逼近求解问题的答案)。不要把这些“解”与问题的“答案”混为一谈，可以把它理解成为要得到答案，系统可能需要利用的那些特性。

- **繁殖(包括子代突变)**

　　带有较高适应度值的那些染色体更可能产生后代(后代产生后也将发生突变)。后代是父母的产物，他们由来自父母的基因结合而成，这个过程被称为“杂交”。

- **下一代**

　　如果新的一代包含一个解，能产生一个充分接近或等于期望答案的输出，那么问题就已经解决了。如果情况并非如此，新的一代将重复他们父母所进行的繁衍过程，一代一代演化下去，直到达到期望的解为止。

遗传算法— 遗传算子

遗传操作包括以下三个基本(genetic operator):选择(selection)；交叉(crossover)；变异(mutation)。这三个遗传算子有如下特点：

选择：从群体中选择优胜的个体，淘汰劣质个体的操作叫选择。

交叉：把两个父代个体的部分结构加以替换重组而生成新个体的操作。通过交叉，遗传算法的搜索能力得以飞跃提高。

变异：群体中的个体串的某些基因座上的基因值作变动

遗传算法引入变异的目的有两个：

一是使遗传算法具有局部的随机搜索能力。当遗传算法通过交叉算子已接近最优解邻域时，利用变异算子的这种局部随机搜索能力可以加速向最优解收敛。显然，此种情况下的变异概率应取较小值，否则接近最优解的积木块会因变异而遭到破坏。

二是使遗传算法可维持群体多样性，以防止出现未成熟收敛现象。此时收敛概率应取较大值。

遗传算法— MATLAB实现

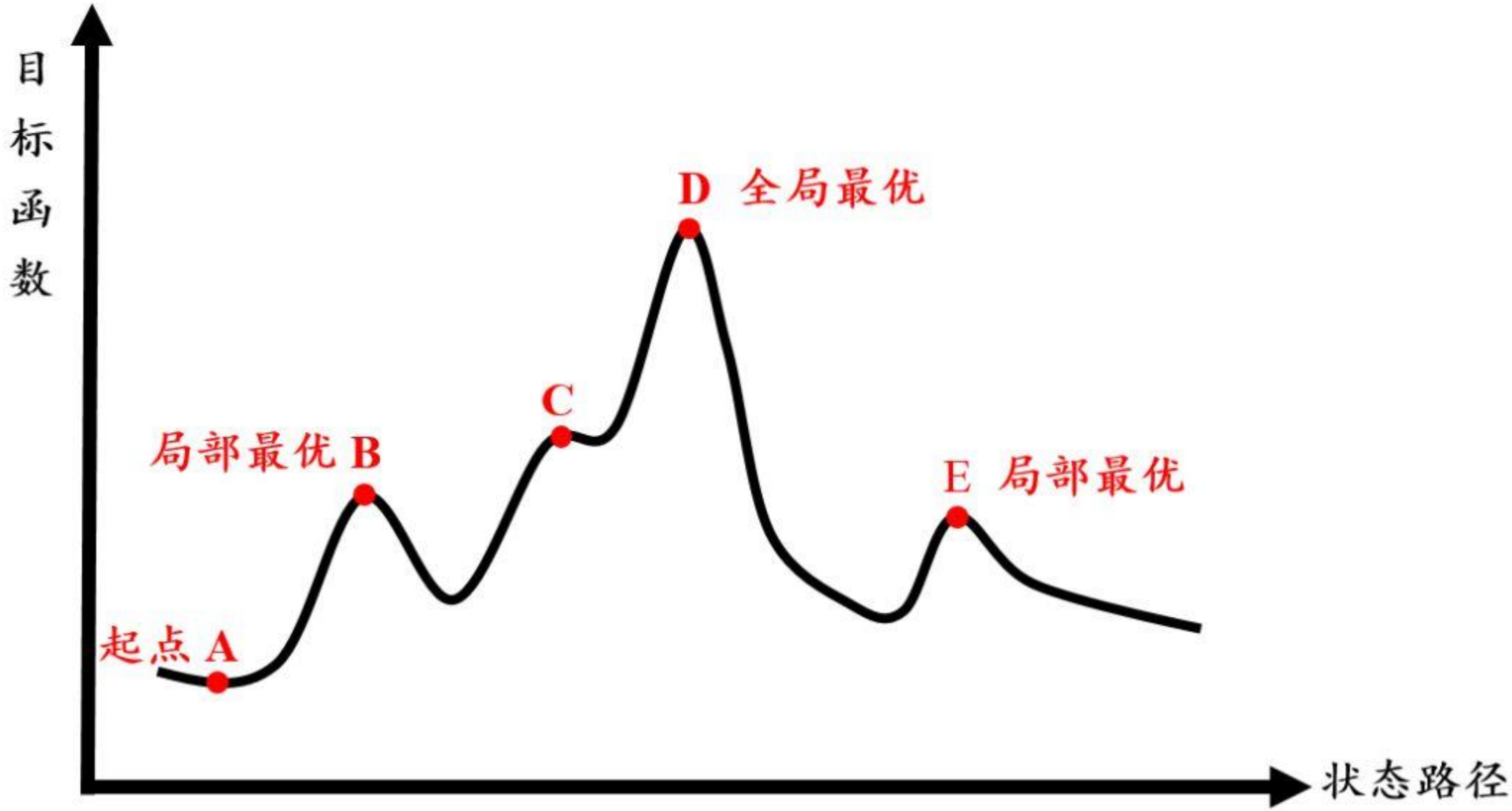
❖ 调用方法

```
x = ga(fitnessfcn, nvars)

x = ga(fitnessfcn, nvars, A, b)

x = ga(fitnessfcn, nvars, A, b, Aeq, beq)

x = ga(fitnessfcn, nvars, A, b, Aeq, beq, LB, UB)

x = ga(fitnessfcn, nvars, A, b, Aeq, beq, LB, UB, nonlcon)

x = ga(fitnessfcn, nvars, A, b, Aeq, beq, LB, UB, nonlcon, options)

[x, fval] = ga(...)

[x, fval, exitflag] = ga(...)
```

模拟退火算法

模拟退火算法来源于固体退火原理，将固体加温至充分高，再让其徐徐冷却，加温时，固体内部粒子随温升变为无序状，内能增大，而徐徐冷却时粒子渐趋有序，在每个温度都达到平衡态，最后在常温时达到基态，内能减为最小。

模拟退火算法在搜索策略上引入了适当的随机因素和物理系统退火过程的自然机理，使得在迭代过程中出现可以接受使目标函数值变“好”的试探点，也可以以一定的概率接受使目标函数值变“差”的试探点，接受概率随着温度的下降逐渐减小。这样避免了搜索过程陷入局部最优解，有利于提高求得全局最优解的可靠性。因此，模拟退火算法具有实用范围广，求得全局最优解的可靠性高，算法简单，便于实现等优点。



模拟退火算法以A点作为初始值，在搜索到局部最优解B点后，会以一定的概率接受往C点方向的移动，可能通过反复地移动搜索就能找到最终的全局最优解D点。

模拟退火算法描述：

1、在指定区间随机产生一定数量的初始解，计算初始目标函数值；
2、结合温度系数更新初始解，计算更新后的目标函数值并计算其与初始目标函数值的差值；
3、根据需要来做判断，这里假设取最大值：若差值大于等于0，则总是接受该移动；若差值小于0，根据Metropolis接受准则来确定取值，根据Metropolis准则，粒子在温度T时趋于平衡的概率为exp(-ΔE/(kT))，其中E为温度T时的内能，ΔE为其改变数，k为玻尔兹曼常数。Metropolis准则可表示为：

$$P=\begin{cases} 1 & if\ E(x_{new})\geq E(x_{old}) \\ e^{\frac{E(x_{new})-E(x_{old})}{kT}} & if\ E(x_{new})<E(x_{old}) \end{cases}$$

由于能量差dE小于0，温度越高，降温的概率就越大；温度越低，则出现降温的概率就越小。

4、重复2,3直至降温结束或达到规定精度要求。

模拟退火算法— MATLAB求解

在MATLAB中提供了使用模拟退火算法解决无约束或者边界约束最优化问题的求解函数**simulannealbnd**，调用格式为：

x = simulannealbnd(fun, x0)

x = simulannealbnd(fun, x0, lb, ub)

x = simulannealbnd(fun, x0, lb, ub, options)

[x, fval] = simulannealbnd(...)

[x, fval, exitflag] = simulannealbnd(...)

[x, fval, exitflag, output] = simulannealbnd(fun,...)