本章作业（网络最优化）

- **作业 2.1:** 使用Dijkstra算法计算设备更新问题（图154）的最短路。
- **作业 2.2:** 通过构造说明最小成本流问题作为其特殊情况包含：最短路问题和最大流问题。
- **作业 2.3:** 考虑一个公司希望在员工与任务之间进行有效的分配。每个任务都需要特定的技能，并且每个员工都有一套技能。每个员工都有能处理的任务数的上限，每个任务需要一个员工去完成。
  给定数据：
  - 员工集合 $E = \{e_1, e_2, e_3\}$
  - 任务集合 $T = \{t_1, t_2, t_3, t_4\}$
  - 每个员工可以处理的任务数量为 $C = \{2, 1, 2\}$
  - 任务分配矩阵 A 定义为：

$$A = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

  其中 $A_{ij} = 1$ 表示员工 e_i 可以执行任务 t_j。
  1. 描述上述数据的流网络图。
  2. 使用Ford-Fulkerson算法或其他最大流算法，求出可以分配的最大任务数量。
  3. 指出哪个员工应该分配哪个任务以达到最大流量。
  4. 如果添加了一个新任务，它可以被 e_1 和 e_3 完成，应如何修改流网络？请指出修改后的最大流量分配。
- **Project 1.1[可选]:** 编程实现求解线性规划问题的单纯形算法，要求提交程序代码，用户指南及测试报告,测试报告需提供退化基解的相应求解结果。

1 绪论

2 线性规划

3 网络最优化

4 动态规划

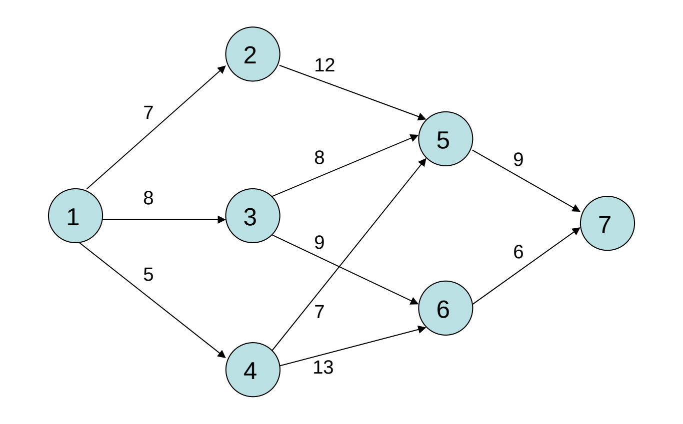
5 非线性规划基础理论

第四章 动态规划

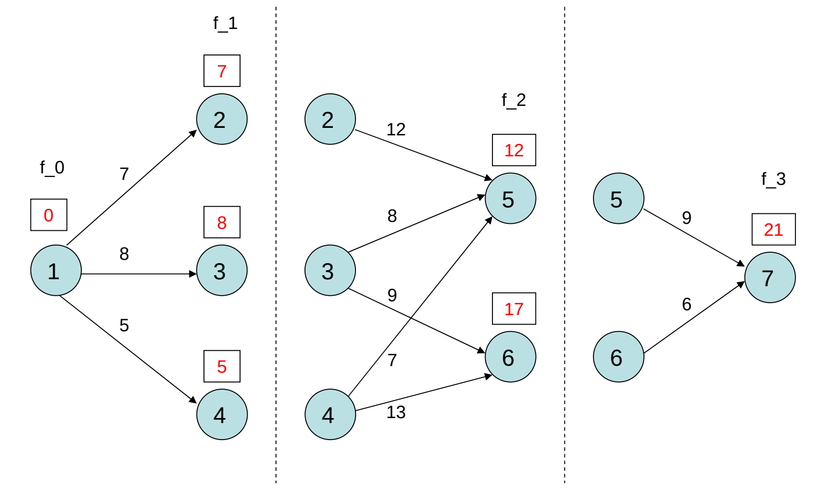
动态规划研究的是“决策过程的最优化”，具有广泛应用背景，并已建立了严密的理论基础。

动态规划主要思想：将问题分解为子问题，并重复使用已有的结论（即写出递归式）。

最短路径问题的一个例子



分阶段考虑



递归方程(Recursive Equation)

$$\begin{cases} f_i(x_i) = \min\limits_{(x_{i-1},x_i)\in E} \{d(x_{i-1},x_i) + f_{i-1}(x_{i-1})\}, \quad i = 1,2,3, \\ f_0(x_0) = 0. \end{cases}$$

动态规划的几个关键要素:

- Stages 阶段
- Alternatives 选择
- States 状态
- Recursive Equations 递归方程、状态转移方程

背包/货物装载(Knapsack/Cargo-Loading)
01背包问题： 最基本的背包问题就是01背包问题（01 knapsack problem）：一共有n件物品，第i（i从1开始）件物品的重量为w_i，价值为r_i。在总重量不超过背包承载上限W的情况下，能够装入背包的最大价值是多少？

我们可以使用枚举法，将所有的情况列举出来，最多有2^n种情况。
也可以写成如下的整数线性规划(Integer Linear Programming)：

$$
\begin{aligned}
\max \quad & z = \sum_{i=1}^{n} r_i m_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} w_i m_i \le W \\
& m_1, \cdots, m_n \in \{0, 1\}
\end{aligned}
$$

完全背包问题： 完全背包问题，就是每个物品可以有无穷个。

A general (n-Items, W-LB) knapsack problem can be represented by the following Integer Linear Programming:

$$
\begin{aligned}
\max \quad & z = \sum_{i=1}^{n} r_i m_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} w_i m_i \leq W \\
& m_1, \cdots, m_n \in \mathbb{Z}_+ \cup \{0\}
\end{aligned}
$$

在最差情况下，求解整数线性规划需要指数时间。

01背包问题

对于01背包问题，我们有如下动态规划解法：

- stage：在0/1背包问题的动态规划中，阶段表示问题求解过程中的决策点或迭代。每个阶段对应于选择将要放入背包或拒绝的一项物品。通常，阶段的数量等于要考虑的物品数量。例如，如果有5个物品，动态规划过程将有5个阶段。
- state：状态表示在每个阶段定义问题所需的信息。在0/1背包问题中，每个阶段的状态通常包括以下组件：
  1. 当前考虑的物品（例如，物品i）。
  2. 在该阶段背包的剩余容量（还可以添加多少重量到背包中）。

  因此，状态可以表示为一对（i, w），其中'i'是当前物品的索引，'w'是该阶段背包的剩余容量。状态有助于跟踪子问题及其解决方案，随着阶段的进行而不断更新。
- alternative：选择：选择指的是每个阶段可以做出的选择或决策。在0/1背包问题中，每个阶段有两种选择：
  1. 选择当前物品并将其添加到背包中，前提是其重量不超过剩余容量。
  2. 拒绝当前物品，继续下一个物品，不将其添加到背包中。

01背包问题

我们用$f(i,w)$ 表示前i个物品，放入最大承重 w的背包最大价值。我们还有$f(i,0)=0, i=0,1,...,n$. 那么，根据上述分析，对于$n \geq i > 1, W \geq w \geq 0$,我们有如下递归方程：

$$f(i,w)=\max\{f(i-1,w), f(i-1,w-w_i)+r_i,(w \geq w_i)\}$$

其中，$f(i-1,w)$ 表示不放入物品i的价值； $f(i-1,w-w_i)+r_i$表示放入i的价值。 使用动态规划可以将复杂度降至$\mathcal{O}(nW)$.

注：背包问题的判定形式（即是否能找到放入方式，使得不超过承重W的情况下达到价值V？）是NP-complete问题。对于一个数W，需要m=log W的位数来表示.因此，m才是输入规模的一部分。所以$O(n*W)=O(n2^m)$,所以是NP问题。

完全背包问题

我们仍用$f(i,w)$ 表示前i个物品，放入最大承重 w的背包最大价值。01背包只有两种情况即取0件和取1件，而这里是取0件、1件、2件...直到超过限重（$k > w/w_i$）

1. Stage i is represented by item i, $i = 1, 2, \cdots, n$.
2. The alternatives at stage i are represented by m_i, the number of units of item i included in the knapsack. It follows that $k = 0, 1, \cdots, \lfloor \frac{w}{w_i} \rfloor$.

我们有边界条件： $f(i,0)=0,\ i=0,1,...,n,$ 递归方程如下

$$f(i,w)=\max_{k=0,1,\cdots,\lfloor\frac{w}{w_i}\rfloor}\{r_ik+f(i-1,w-w_ik)\},\quad i=1,\cdots,n,1\leq w\leq$$

例子： 7-ton Vessel

Item i	1	2	3
w_i	2	3	1
r_i	31	47	15

设备更新模型(Equipment Replacement Model)

假设我们考虑的是一个跨度为n年的设备更新问题。每年初我们需要决定是否保留当前设备再使用一年或者更换一个新的设备。令$r(t)$, $c(t)$, $s(t)$分别表示一台 t-年龄设备的年营业收入，年运营成本及其残余价值。另外，在规划期内的任何一年购置一台新设备的成本是I.

动态规划

设备更新模型：设某公司现有一台3年龄的设备，需制定一个未来4年($n = 5$)的设备更新最优策略。该公司还规定6年龄的设备必须得以更换。一台新设备的成本是\$100,000. 下表给出的是设备更新问题的相关数据，其中t是机器年龄， $r(t), c(t), s(t)$ 分别表示t年龄机器的年营业收入，年运营成本及残余价值。

Table: 设备更新问题数据表

t	$r(t)$	$c(t)$	$s(t)$
0	20,000	200	–
1	19,000	600	80,000
2	18,500	1,200	60,000
3	17,200	1,500	50,000
4	15,500	1,700	30,000
5	14,000	1,800	10,000
6	12,200	2,200	5,000

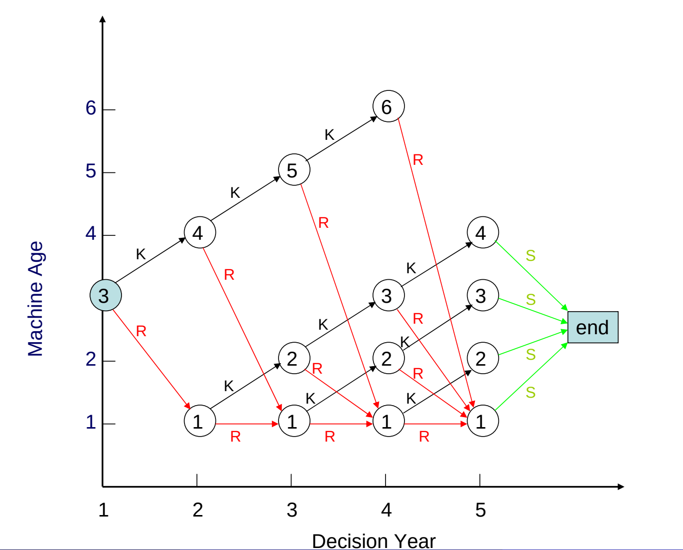
设备更新模型

- Stage i is represented by year i where $i = 1, \cdots, n$.
- The alternatives at stage i are either *keeping* or *replacing* the machine at the start of year i.
- The state at stage i is the age of the machine at the start of year i.
- Recursive equation:

$$f_i(t) = \max \begin{cases} r(t) - c(t) + f_{i+1}(t+1), t \le 6 & \text{if Keeping} \\ r(0) - c(0) + s(t) - I + f_{i+1}(1), & \text{if Replacing} \end{cases}$$
$$f_{n+1}(t) = s(t)$$

where $f_i(t)$ is defined as the maximum net income for years $i, i+1, \cdots, n$ by given a t-year-old machine at the start of year i.

设备更新模型



- **Exercise 3.1:** 设现有一台2年龄的设备，另规定5年龄的设备必须更换。在规划期购置新设备的成本分别是

$$(p_1, p_2, p_3, p_4, p_5) = (100, 105, 110, 115, 120).$$

试构建如下设备更新的动态规划模型并求其最优更新策略。

Table: 五年期设备更新

设备年龄t	残余价值v_t	运行费用c_t
0	-	30
1	50	40
2	25	50
3	10	75
4	5	90
5	2	-

本章作业（动态规划）

- **Exercise 3.2:**
  问题1：有一个移动机器人，可以在二维矩阵平面中移动，其移动的起点位于左上角，每次移动只能向右或者向下，其移动的终点是右下角，在这个矩阵形的平面中每个位置都有一个数值，代表机器人在经过这个区域时需要付出的代价。我们的目标就是，在这个平面中，找一条代价最小的路径，并且给出最小代价路径。
  问题2：若机器人从左上角出发，起点坐标设为（0，0），需要达到右下角，坐标设为（M，N）。请问有多少种不同的路径（无需考虑每格代价）?

2	3	1
1	9	1
6	4	2

本章作业（动态规划）

- **Project 3.3[可选]:** 编程实现最短路径算法，要求提交程序代码，用户指南及测试报告,测试报告需报告算法的求解时间随着问题规模变化情况。
  第二次作业包括第二章3道习题和本章2道习题，提交时间为10月22日24点之前。共有3个project题目，任选2个完成，学期结束前2周提交。

1 绪论

2 线性规划

3 网络最优化

4 动态规划

5 非线性规划基础理论

$$
\begin{aligned}
\min \quad & f(\mathbf{x}) \\
\text{s.t.} \quad & \mathbf{x} \in S \subset \mathbb{R}^n.
\end{aligned} \tag{51}
$$

在此，目标函数f是定义在$\mathbb{R}^n$上的实值函数，S是决策变量$\mathbf{x}$的可取值之集合，称为问题的可行域(feasible region).

最优化问题从属性上可以分为两大类：一类是具有连续变量的问题，另一类是离散变量的问题（即组合优化问题）。

非线性规划属于连续型最优化问题的范畴，通常可行域S可由一组方程来描述，即

$$S = \{\mathbf{x} \in \mathbb{R}^n \mid g_i(\mathbf{x}) \geq 0, i = 1, \cdots, m; h_j(\mathbf{x}) = 0, j = 1, \cdots, \ell\}.$$

非线性规划

$$
\begin{aligned}
\min \quad & f(\mathbf{x}) \\
\text{s.t.} \quad & g_i(\mathbf{x}) \geq 0, i = 1, \cdots, m \\
& h_j(\mathbf{x}) = 0, j = 1, \cdots, \ell
\end{aligned} \tag{52}
$$

这里，$f(\mathbf{x})$, $g_i(\mathbf{x})$, $h_j(\mathbf{x})$都是n变量、实值、确定的函数，且至少有一个是非线性的。

非线性规划

为求解一个非线性规划问题（即找出其最优解），与此相关的研究分两个方面：一是研究最优解的性质，二是设计有效算法来获得问题的解。

回顾最优解的定义

满足约束条件$\mathbf{x} \in S$的$\mathbf{x}$称为问题的可行解(feasible solution)，如果可行解$\mathbf{x}^* \in S$进一步满足

$$f(\mathbf{x}^*) \leqslant f(\mathbf{x}), \forall \mathbf{x} \in S. \quad (53)$$

则称$\mathbf{x}^*$为问题(51)的全局最优解(global optimal solution). 另外，在包含可行解$\mathbf{x}^* \in S$的适当邻域$U(\mathbf{x}^*)$里，成立

$$f(\mathbf{x}^*) \leqslant f(\mathbf{x}), \forall \mathbf{x} \in S \cap U(\mathbf{x}^*). \quad (54)$$

此时称$\mathbf{x}^*$为问题(51)的局部最优解(local optimal solution).

最优性条件

最优性条件：问题的最优解所满足的必要或者充分条件。

最优性条件将为各种求解算法的设计、分析提供必不可少的理论基础。

无约束问题的极值条件

一阶必要条件：设目标函数$f(\mathbf{x})$在点$\bar{\mathbf{x}}$处可微，若$\bar{\mathbf{x}}$ 是局部极小点，则$\nabla f(\bar{\mathbf{x}}) = 0$.

二阶必要条件：设目标函数$f(\mathbf{x})$在点$\bar{\mathbf{x}}$处二次可微，若$\bar{\mathbf{x}}$是局部极小点，则$\nabla f(\bar{\mathbf{x}}) = 0$, 并且Hesse矩阵$\nabla^2 f(\bar{\mathbf{x}}) \geq 0$.

二阶充分条件：设目标函数$f(\mathbf{x})$在点$\bar{\mathbf{x}}$处二次可微，若$\nabla f(\bar{\mathbf{x}}) = 0$且$\nabla^2 f(\bar{\mathbf{x}}) > 0$, 则$\bar{\mathbf{x}}$是局部极小点。

充要条件：设$f(\mathbf{x})$是定义在$\mathbb{R}^n$上的可微凸函数，则$\bar{\mathbf{x}}$是整体极小点（全局最优解）的充要条件是$\nabla f(\bar{\mathbf{x}}) = 0$.

可行方向

设$\bar{\mathbf{x}} \in S$，$\mathbf{d} \in \mathbb{R}^n$是非零向量。若存在$\delta > 0$使得：

$$\bar{\mathbf{x}} + \lambda \mathbf{d} \in S, \forall \lambda \in (0, \delta) \tag{55}$$

则称$\mathbf{d}$是S在$\bar{\mathbf{x}}$处的可行方向。

记S在$\bar{\mathbf{x}}$处的所有可行方向的集合为$F(\bar{\mathbf{x}}, S)$.

下降方向

设$f(\mathbf{x})$是$\mathbb{R}^n$上的实函数，$\bar{\mathbf{x}} \in \mathbb{R}^n$， $\mathbf{d}$是非零向量。若存在$\delta > 0$使得：

$$f(\bar{\mathbf{x}} + \lambda \mathbf{d}) < f(\bar{\mathbf{x}}), \forall \lambda \in (0, \delta) \tag{56}$$

则称$\mathbf{d}$为函数$f(\mathbf{x})$在$\bar{\mathbf{x}}$处的下降方向。

下降方向集的子集

如果$f(\mathbf{x})$是可微函数，且$\nabla f(\bar{\mathbf{x}})^T\mathbf{d} < 0$. 显然，此处的 $\mathbf{d}$ 为 $f(\mathbf{x})$ 在 $\bar{\mathbf{x}}$ 处的下降方向。记这样的方向集合为

$$D(\bar{\mathbf{x}}, f) = \{\mathbf{d} \mid \nabla f(\bar{\mathbf{x}})^T\mathbf{d} < 0\}.$$

结论

对于问题$\min\{f(\mathbf{x}) \mid \mathbf{x} \in S\}$，设$\bar{\mathbf{x}} \in S$，$f(\mathbf{x})$在$\bar{\mathbf{x}}$处可微。如果$\bar{\mathbf{x}}$是问题的局部最优解，则可行方向集中无下降方向，即

$$F(\bar{\mathbf{x}}, S) \cap D(\bar{\mathbf{x}}, f) = \emptyset. \tag{57}$$

记$\mathcal{I}(\bar{\mathbf{x}}) = \{i \in \{1, \cdots, m\} \mid g_i(\bar{\mathbf{x}}) = 0\}$,

$$D_f = D(\bar{\mathbf{x}}, f) = \{\mathbf{d} \mid \nabla f(\bar{\mathbf{x}})^T \mathbf{d} < 0\}, \tag{58}$$

$$F_g = F(\bar{\mathbf{x}}, g) = \{\mathbf{d} \mid \nabla g_i(\bar{\mathbf{x}})^T \mathbf{d} > 0, i \in \mathcal{I}(\bar{\mathbf{x}})\}, \tag{59}$$

$$F_h = F(\bar{\mathbf{x}}, h) = \{\mathbf{d} \mid \nabla h_j(\bar{\mathbf{x}})^T \mathbf{d} = 0, j = 1, \cdots, \ell\}. \tag{60}$$

设$\bar{\mathbf{x}}$为问题(52)的局部最优解， f和$g_i, i \in \mathcal{I}(\bar{\mathbf{x}})$在点$\bar{\mathbf{x}}$可微，$g_i, i \notin \mathcal{I}(\bar{\mathbf{x}})$在点$\bar{\mathbf{x}}$连续，$h_j$在点$\bar{\mathbf{x}}$连续可微，且$\{\nabla h_j(\bar{\mathbf{x}})\}_{j=1}^{\ell}$线性无关，则

$$D_f \cap F_g \cap F_h = \emptyset. \tag{61}$$

约束问题的最优性条件

Fritz-John条件：在问题(52)中，设$\bar{\mathbf{x}}$为可行点，f和$g_i, i \in \mathcal{I}(\bar{\mathbf{x}})$在点$\bar{\mathbf{x}}$可微， $g_i, i \notin \mathcal{I}(\bar{\mathbf{x}})$在点$\bar{\mathbf{x}}$连续，$h_j$在点$\bar{\mathbf{x}}$连续可微。如果$\bar{\mathbf{x}}$是局部最优解，则存在不全为零的数 $\lambda_0, \lambda_i, i \in \mathcal{I}(\bar{\mathbf{x}})$ 和 $\mu_j, j = 1, \cdots, \ell$ 使得

$$\lambda_0 \nabla f(\bar{\mathbf{x}}) - \sum_{i \in \mathcal{I}(\bar{\mathbf{x}})} \lambda_i \nabla g_i(\bar{\mathbf{x}}) - \sum_{j=1}^{\ell} \mu_j \nabla h_j(\bar{\mathbf{x}}) = 0. \quad (62)$$

其中$\lambda_0 \geq 0, \lambda_i \geq 0, i \in \mathcal{I}(\bar{\mathbf{x}})$.

证明:(1) 如果$\{\nabla h_j(\bar{\mathbf{x}})\}_{j=1}^{\ell}$线性相关，则存在不全为零的数$\mu_j, j = 1, \cdots, \ell$ 使得

$$\sum_{j=1}^{\ell} \mu_j \nabla h_j(\bar{\mathbf{x}}) = 0.$$

这时可令$\lambda_0 = 0, \lambda_i = 0, i \in \mathcal{I}(\bar{\mathbf{x}})$, 结论成立。

约束问题的最优性条件

证明（续）:(2)如果$\{\nabla h_j(\bar{\mathbf{x}})\}_{j=1}^{\ell}$线性无关。利用$D_f \cap F_g \cap F_h = \emptyset$.
即不等式组

$$\begin{cases} \nabla f(\bar{\mathbf{x}})^T \mathbf{d} < 0 \\ \nabla g_i(\bar{\mathbf{x}})^T \mathbf{d} > 0, i \in \mathcal{I}(\bar{\mathbf{x}}) \\ \nabla h_j(\bar{\mathbf{x}})^T \mathbf{d} = 0, j = 1, \cdots, \ell \end{cases} \tag{63}$$

无解。

证明(续)：令A是以 $\{\nabla f(\bar{\mathbf{x}}), -\nabla g_i(\bar{\mathbf{x}}), i \in \mathcal{I}(\bar{\mathbf{x}})\}$ 为列组成的矩阵，B是以$\{-\nabla h_j(\bar{\mathbf{x}}), j = 1, \cdots, \ell\}$为列组成的矩阵。
于是得

$$\begin{cases} A^T \mathbf{d} < 0 \\ B^T \mathbf{d} = 0 \end{cases} \tag{64}$$

无解。

下证

$$\begin{cases} A\mathbf{p}_1 + B\mathbf{p}_2 = 0 \\ \mathbf{p}_1 \geq 0 \end{cases} \tag{65}$$

有解。

定理 (凸集分离定理)

假设 C_1 和 C_2 是两个不相交的非空凸集，那么存在一个非零向量 w 和一个实数 b 使得对于所有 $x_1 \in C_1$ 和 $x_2 \in C_2$ 有：

$$w^T x_1 \geq b \quad \text{和} \quad w^T x_2 \leq b$$

这意味着超平面 $\{x : w^T x + b = 0\}$ 将 C_1 和 C_2 分开。

证明(续)：现定义

$$S_1 = \{\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \mid \mathbf{y}_1 = A^T\mathbf{d}, \mathbf{y}_2 = B^T\mathbf{d}, \mathbf{d} \in \mathbb{R}^n\},$$
$$S_2 = \{\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \mid \mathbf{y}_1 < \mathbf{0}, \mathbf{y}_2 = \mathbf{0}\}.$$

显然S_1和S_2为非空凸集，且$S_1 \cap S_2 = \emptyset$.

证明(续)：由凸集分离定理知，对$\forall \mathbf{d} \in \mathbb{R}^n, \forall \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \in S_2$，存在非零向量$\begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix}$使得$\mathbf{p}_1^T A^T \mathbf{d} + \mathbf{p}_2^T B^T \mathbf{d} \geq \mathbf{p}_1^T \mathbf{y}_1 + \mathbf{p}_2^T \mathbf{y}_2$.

首先令$\mathbf{y}_2 = \mathbf{0}$, 由$\mathbf{d}$的任意性（取$\mathbf{d} = \mathbf{0}$）及$\mathbf{y}_1 < \mathbf{0}$, $\Longrightarrow \mathbf{p}_1 \geq \mathbf{0}$.

再令$\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} \in S_2$, $\Longrightarrow \mathbf{p}_1^T A^T \mathbf{d} + \mathbf{p}_2^T B^T \mathbf{d} \geq 0$.

最后取$\mathbf{d} = -(A\mathbf{p}_1 + B\mathbf{p}_2)$, $\Longrightarrow A\mathbf{p}_1 + B\mathbf{p}_2 = \mathbf{0}$.

综上所述，即得(65)有解。

证明(续):

把$\mathbf{p}_1$的分量记作λ_0和$\lambda_i, i \in \mathcal{I}(\bar{\mathbf{x}})$, $\mathbf{p}_2$的分量记作$\mu_j, j = 1, \cdots, \ell$. 立即得到

$$\lambda_0 \nabla f(\bar{\mathbf{x}}) - \sum_{i \in I(\bar{\mathbf{x}})} \lambda_i \nabla g_i(\bar{\mathbf{x}}) - \sum_{j=1}^{\ell} \mu_j \nabla h_j(\bar{\mathbf{x}}) = 0. \tag{66}$$

KKT条件：设$\bar{\mathbf{x}}$为约束问题(52)的可行点，f和$g_i, i \in \mathcal{I}(\bar{\mathbf{x}})$ 在点$\bar{\mathbf{x}}$可微，$g_i, i \notin \mathcal{I}(\bar{\mathbf{x}})$ 在点$\bar{\mathbf{x}}$连续， h_j在点$\bar{\mathbf{x}}$连续可微，向量集 $\{\nabla g_i(\bar{\mathbf{x}}), i \in \mathcal{I}(\bar{\mathbf{x}}); \nabla h_j(\bar{\mathbf{x}}), j = 1, \cdots, \ell\}$ 线性无关。如果$\bar{\mathbf{x}}$是局部最优解，则存在数$\lambda_i \geq 0$ 和 μ_j 使得

$$\nabla f(\bar{\mathbf{x}}) - \sum_{i \in \mathcal{I}(\bar{\mathbf{x}})} \lambda_i \nabla g_i(\bar{\mathbf{x}}) - \sum_{j=1}^{\ell} \mu_j \nabla h_j(\bar{\mathbf{x}}) = 0. \tag{67}$$

证明：利用Fritz John条件，若LICQ成立，则$\lambda_0 \neq 0$.两边同除以 λ_0得证。

定义Lagrange函数 $L(\mathbf{x},\lambda,\mu)=f(\mathbf{x})-\sum_{i=1}^{m}\lambda_i g_i(\mathbf{x})-\sum_{j=1}^{\ell}\mu_j h_j(\mathbf{x})$.

若$\bar{\mathbf{x}}$为问题局部最优解，则存在乘子向量$\bar{\lambda}\geq 0,\bar{\mu}$使得

$$\nabla_{\mathbf{x}}L(\bar{\mathbf{x}},\bar{\lambda},\bar{\mu})=0.$$

此时，一阶必要条件可表达为

$$\text{(KKT)}\begin{cases}\nabla_{\mathbf{x}}L(\mathbf{x},\lambda,\mu)=0\\ g_i(\mathbf{x})\geq 0, i=1,\cdots,m\\ \lambda_i g_i(\mathbf{x})=0, i=1,\cdots,m\\ \lambda_i\geq 0, i=1,\cdots,m\\ h_j(\mathbf{x})=0, j=1,\cdots,\ell\end{cases}\tag{68}$$

下降算法

理论上可以用最优性条件求“非线性规划问题”的最优解，但在实践中并不切实可行。

在求解最优化问题时最常用的计算方法是“迭代下降算法”。

算法映射

算法映射：算法$\mathcal{A}$是定义在空间$\mathbf{X}$上的**点到集**的映射，即对每一点$\mathbf{x}^{(k)} \in \mathbf{X}$，经算法$\mathcal{A}$作用后产生一个点集$\mathcal{A}(\mathbf{x}^{(k)}) \subset \mathbf{X}$，任意选择一个点$\mathbf{x}^{(k+1)} \in \mathcal{A}(\mathbf{x}^{(k)})$作为$\mathbf{x}^{(k)}$的后续点。

闭映射

引入所谓算法的闭性，其实质是点到点映射的连续性的推广。

设$\mathbf{X}$和$\mathbf{Y}$分别是空间$\mathbb{E}^p$和$\mathbb{E}^q$中的非空闭集， $\mathcal{A}: \mathbf{X} \longrightarrow \mathbf{Y}$为点到集的映射。如果$\mathbf{x}^{(k)} \in \mathbf{X}, \mathbf{x}^{(k)} \to \mathbf{x}, \mathbf{y}^{(k)} \in \mathcal{A}(\mathbf{x}^{(k)}), \mathbf{y}^{(k)} \to \mathbf{y}$ 蕴涵着$\mathbf{y} \in \mathcal{A}(\mathbf{x})$, 则称映射$\mathcal{A}$在$\mathbf{x} \in \mathbf{X}$处是闭的。

解集合

为研究算法的收敛性，首先要明确解集合的概念。

在许多情况下，要使算法产生的点列收敛于全局最优解是极为困难的。因此，一般把满足某些条件的点集定义为解集合。当迭代点属于这个集合时，就停止迭代。

例如，在无约束最优化问题中，可以定义解集合为

$$\Omega = \{\bar{\mathbf{x}} \mid \|\nabla f(\bar{\mathbf{x}})\| = 0\},$$

在约束最优化问题中，解集合取为

$$\Omega = \{\bar{\mathbf{x}} \mid \bar{\mathbf{x}}\text{是KKT点}\}.$$

下降函数

一般地，下降算法总是与某个函数在迭代过程中函数值的减小联系在一起，因此需要给出下降函数的概念。

设$\Omega \subset \mathbf{X}$为解集合，$\mathcal{A}$为$\mathbf{X}$上的一个算法映射， $\psi(\mathbf{x})$是定义在$\mathbf{X}$上的连续实函数，若满足

当$\mathbf{x} \notin \Omega$且$\mathbf{y} \in \mathcal{A}(\mathbf{x})$时，$\psi(\mathbf{y}) < \psi(\mathbf{x})$
当$\mathbf{x} \in \Omega$且$\mathbf{y} \in \mathcal{A}(\mathbf{x})$时，$\psi(\mathbf{y}) \leq \psi(\mathbf{x})$

则称ψ是关于解集合Ω和算法$\mathcal{A}$的下降函数。

算法收敛性

设Ω为解集合，$\mathcal{A}$为$\mathbf{X}$上的算法映射。若以任意初始点$\mathbf{x}^{(0)} \in \mathbf{Y} \subset \mathbf{X}$ 出发，算法产生的序列的任一收敛子列的极限属于解集合，则称算法映射$\mathcal{A}$在$\mathbf{Y}$ 上收敛于解集合Ω.

定理： 设$\mathcal{A}$为$\mathbf{X}$上的一个算法，Ω为解集合，给定初始点 $\mathbf{x}^{(0)} \in \mathbf{X}$, 进行如下迭代：如果$\mathbf{x}^{(k)} \in \Omega$, 则停止迭代；否则取 $\mathbf{x}^{(k+1)} \in \mathcal{A}(\mathbf{x}^{(k)})$, $k := k+1$, 重复以上过程。这样产生迭代序列$\{\mathbf{x}^{(k)}\}$. 又设：

1. 序列$\{\mathbf{x}^{(k)}\}$含于$\mathbf{X}$的紧子集中；
2. 存在一个连续函数ψ, 它是关于Ω和$\mathcal{A}$的下降函数；
3. 映射$\mathcal{A}$在Ω的补集上是闭的。

则序列$\{\mathbf{x}^{(k)}\}$的任一收敛子列的极限属于Ω.

算法收敛性定理证明

证明： 先证序列$\{\mathbf{x}^{(k)}\}$对应的下降函数值数列$\{\psi(\mathbf{x}^{(k)})\}$有极限。

$\{\mathbf{x}^{(k)}\}$含于$\mathbf{X}$的紧子集，因此有收敛子列$\{\mathbf{x}^{(k)}\}_K$，设其极限为$\mathbf{x} \in \mathbf{X}$.
由ψ的连续性得，$\psi(\mathbf{x}^{(k)}) \longrightarrow \psi(\mathbf{x}), k \in K$. 即对$\forall\epsilon > 0, \exists N$使得
当$k \geq N$时，有$0 < \psi(\mathbf{x}^{(k)}) - \psi(\mathbf{x}) < \epsilon, k \in K$. 特别地，
$0 < \psi(\mathbf{x}^{(N)}) - \psi(\mathbf{x}) < \epsilon$.

又由ψ的下降性知，$\psi(\mathbf{x}^{(k)}) - \psi(\mathbf{x}^{(N)}) < 0, \forall k > N$. 于是有

$$0 < \psi(\mathbf{x}^{(k)}) - \psi(\mathbf{x}) = \psi(\mathbf{x}^{(k)}) - \psi(\mathbf{x}^{(N)}) + \psi(\mathbf{x}^{(N)}) - \psi(\mathbf{x}) < \epsilon, \forall k > N.$$

即得 $\lim\limits_{k\to\infty} \psi(\mathbf{x}^{(k)}) = \psi(\mathbf{x})$.

算法收敛性定理证明

证明（继）： 下证$\mathbf{x} \in \Omega$（反证法）。

假设$\mathbf{x} \notin \Omega$, 考虑序列$\{\mathbf{x}^{(k+1)}\}_K$. 由于它包含于紧集，所以也存在收敛子列$\{\mathbf{x}^{(k+1)}\}_{\bar{K}}$, $\bar{K} \subset K$, 且设其极限为$\bar{\mathbf{x}} \in \mathbf{X}$. 显然（同理前述证明）$\lim\limits_{k\to\infty} \psi(\mathbf{x}^{(k+1)}) = \psi(\bar{\mathbf{x}})$. 由$\psi(\mathbf{x}^{(k)})$极限的唯一性知，

$$\psi(\mathbf{x}) = \psi(\bar{\mathbf{x}}).$$

另外，对$k \in \bar{K} \subset K$有

$$\mathbf{x}^{(k)} \longrightarrow \mathbf{x}, \mathbf{x}^{(k+1)} \in \mathcal{A}(\mathbf{x}^{(k)}), \mathbf{x}^{(k+1)} \longrightarrow \bar{\mathbf{x}}.$$

算法$\mathcal{A}$在Ω的补集上是闭的，$\mathbf{x} \notin \Omega$, 因此$\mathcal{A}$在$\mathbf{x}$处是闭的，即有$\bar{\mathbf{x}} \in \mathcal{A}(\mathbf{x})$.

由于ψ是解集合Ω和算法$\mathcal{A}$的下降函数，$\mathbf{x} \notin \Omega$, 则有$\psi(\bar{\mathbf{x}}) < \psi(\mathbf{x})$. 这显然矛盾，所以$\mathbf{x} \in \Omega$.

实用收敛准则

在迭代下降算法里，当$\mathbf{x}^{(k)} \in \Omega$时才终止迭代。在实践中许多情况下，这是一个取极限的过程，需要无限次迭代。因此为解决实际问题，需要规定一些实用的终止迭代过程的准则，一般称为收敛准则或停机准则。

常用的收敛准则有以下几种：

1. $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \varepsilon$ or $\dfrac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|}{\|\mathbf{x}^{(k)}\|} < \varepsilon$
2. $f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k+1)}) < \varepsilon$ or $\dfrac{f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k+1)})}{|f(\mathbf{x}^{(k)})|} < \varepsilon$
3. $\|\nabla f(\mathbf{x}^{(k)})\| < \varepsilon$

在这里，ε为事先给定的充分小的正数。除此之外，还可以根据收敛定理，制定出其它的收敛准则。

收敛速率

评价算法优劣的标准之一是收敛的快慢，通常称为收敛速率。

一般定义如下：设序列$\{\mathbf{x}^{(k)}\}$收敛与$\mathbf{x}^*$，满足

$$0 \leq \overline{\lim_{k\to\infty}} \frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\|}{\|\mathbf{x}^{(k)} - \mathbf{x}^*\|^p} = \beta < \infty \tag{69}$$

的非负数p的上确界称为序列$\{\mathbf{x}^{(k)}\}$的收敛阶。

若在定义式(69)中，$p = 1$且$\beta < 1$，则称序列是（收敛比β）线性收敛的。

若在定义式(69)中，$p > 1$，或者$\{p = 1, \beta = 0\}$，则称序列是超线性收敛的。

计算复杂性

最优化可以追溯到十分古老的极值问题，然而它成为一门独立的学科是在二十世纪四十年代末，当时Dantzig提出了求解一般线性规划问题的单纯形法。此后各种最优化问题的理论及应用研究得到迅速发展，特别是线性规划由于其模型的普遍性和实用性，相关算法的进展引起广泛的重视。

随着实际问题的规模越来越大以及在计算机技术的推动下，人们开始从复杂性角度研究线性规划和非线性规划的算法。

迭代方法

最优化方法通常采用迭代方法求问题的最优解，其基本思想是：

给定一个初始点$\mathbf{x}^{(0)} \in \mathbb{R}^n$, 按照某一迭代规则产生一个点列$\{\mathbf{x}^{(k)}\}$, 使得当$\{\mathbf{x}^{(k)}\}$是有穷点列时，其最后一个点是最优化模型问题的最优解，当$\{\mathbf{x}^{(k)}\}$是无穷点列时，它有极限点且其极限点是最优化模型问题的最优解。

迭代方法

一个好的迭代算法应具备的典型特征是:

迭代点$\mathbf{x}^{(k)}$能稳定地接近局部极小点$\mathbf{x}^*$的小邻域，然后迅速收敛于$\mathbf{x}^*$. 一般地，对于某种算法我们需要证明其迭代点列$\mathbf{x}^{(k)}$的聚点（即子列的极限点）为一局部极小点。在实际计算中，当指定的收敛准则满足时，迭代即终止。

搜索方向与步长因子

设$\mathbf{x}^{(k)}$为第k次迭代点，$\mathbf{d}^{(k)}$为第k次搜索方向，α_k为第k次步长因子，则第$k+1$次迭代为：

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}. \tag{70}$$

从上述迭代格式可以看出，不同的搜索方向和不同的步长策略构成不同的方法。

搜索方向与步长因子

在最优化方法中，搜索方向$\mathbf{d}^{(k)}$一般选取的是某价值函数 (merit function) ψ 在$\mathbf{x}^{(k)}$处的下降方向，即$\mathbf{d}^{(k)}$满足

$$\nabla\psi(\mathbf{x}^{(k)})^T\mathbf{d}^{(k)} < 0. \tag{71}$$

步长因子的确定一般归结为解一维最优化问题

$$\min_{\alpha\geq 0}\psi(\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)}) \tag{72}$$

最优化迭代算法的基本结构之一

(a) 给定初始点$\mathbf{x}^{(0)}$

(b) 计算搜索方向$\mathbf{d}^{(k)}$, 即构造某价值函数ψ在$\mathbf{x}^{(k)}$点处的下降方向作为搜索方向;

(c) 确定步长因子α_k, 使该价值函数值有某种程度的下降;

(d) 迭代更新，令$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$.

(e) 判断停机准则，若$\mathbf{x}^{(k+1)}$满足某种终止条件，则停止迭代，得到近似最优解$\bar{\mathbf{x}} = \mathbf{x}^{(k+1)}$. 否则，返回(b)重复以上步骤。

最优化迭代算法的基本结构之二

(a) 给定初始点$\mathbf{x}^{(0)}$
(b) 构造某价值函数ψ在$\mathbf{x}^{(k)}$附近（如一定半径内）的二次近似模型；
(c) 求解该近似模型得到$\mathbf{s}^{(k)}$作为更新位移向量；
(d) 迭代更新，令$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$.
(e) 判断停机准则，若$\mathbf{x}^{(k+1)}$满足某种终止条件，则停止迭代，得到近似最优解$\bar{\mathbf{x}} = \mathbf{x}^{(k+1)}$. 否则，返回(b)重复以上步骤。