

## 第四章 动态规划

### §1 引言

#### 1.1 动态规划的发展及研究内容

动态规划(dynamic programming)是运筹学的一个分支,是求解决策过程(decision process)最优化的数学方法。20 世纪 50 年代初 R. E. Bellman 等人在研究多阶段决策过程(multistep decision process)的优化问题时,提出了著名的最优性原理(principle of optimality),把多阶段过程转化为一组单阶段问题,逐个求解,创立了解决这类过程优化问题的新方法—动态规划。1957 年出版了他的名著《Dynamic Programming》,这是该领域的第一本著作。

动态规划问世以来,在经济管理、生产调度、工程技术和最优控制等方面得到了广泛的应用。例如最短路线、库存管理、资源分配、设备更新、排序、装载等问题,用动态规划方法比用其它方法求解更为方便。

虽然动态规划主要用于求解以时间划分阶段的动态过程的优化问题,但是一些与时间无关的静态规划(如线性规划、非线性规划),只要人为地引进时间因素,把它视为多阶段决策过程,也可以用动态规划方法方便地求解。

应指出,动态规划是求解某类问题的一种方法,是考察问题的一种途径,而不是一种特殊算法(如线性规划是一种算法)。因而,它不象线性规划那样有一个标准的数学表达式和明确定义的一组规则,而必须对具体问题进行分析处理。因此,在学习时,除了要对基本概念和方法正确理解外,应以丰富的想象力去建立模型,用创造性的技巧去求解。

#### 例 1 最短路线问题

图 1 是一个线路网,连线上的数字表示两点之间的距离(或费用)。试寻求一条由 A 到 G 距离最短(或费用最省)的路线。

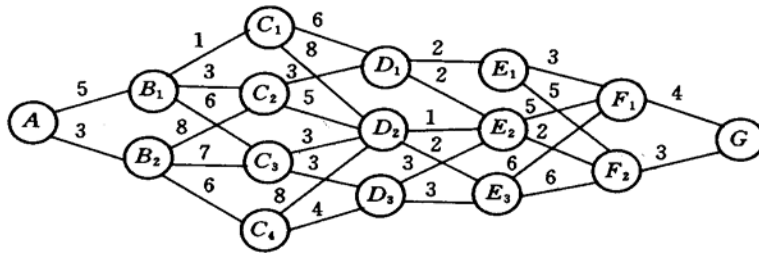


图 1 最短路线问题

#### 例 2 生产计划问题

工厂生产某种产品,每单位(千件)的成本为 1(千元),每次开工的固定成本为 3(千元),工厂每季度的最大生产能力为 6(千件)。经调查,市场对该产品的需求量第一、二、三、四季度分别为 2, 3, 2, 4(千件)。如果工厂在第一、二季度将全年的需求都生产出来,自然可以降低成本(少付固定成本费),但是对于第三、四季度才能上市的产品需付存储费,每季每千件的存储费为 0.5(千元)。还规定年初和年末这种产品均无库存。试制定一个生产计划,即安排每个季度的产量,使一年的总费用(生产成本和存储费)最少。

#### 1.2 决策过程的分类

根据过程的时间变量是离散的还是连续的,分为离散时间决策过程(discrete-time

decision process) 和连续时间决策过程 (continuous-time decision process); 根据过程的演变是确定的还是随机的, 分为确定性决策过程 (deterministic decision process) 和随机性决策过程 (stochastic decision process), 其中应用最广的是确定性多阶段决策过程。

## §2 基本概念、基本方程和计算方法

### 2.1 动态规划的基本概念和基本方程

一个多阶段决策过程最优化问题的动态规划模型通常包含以下要素。

#### 2.1.1 阶段

阶段(step)是对整个过程的自然划分。通常根据时间顺序或空间顺序特征来划分阶段, 以便按阶段的次序解优化问题。阶段变量一般用  $k = 1, 2, \dots, n$  表示。在例 1 中由  $A$  出发为  $k = 1$ , 由  $B_i (i = 1, 2)$  出发为  $k = 2$ , 依此下去从  $F_i (i = 1, 2)$  出发为  $k = 6$ , 共  $n = 6$  个阶段。在例 2 中按照第一、二、三、四季度分为  $k = 1, 2, 3, 4$ , 共四个阶段。

#### 2.1.2 状态

状态(state)表示每个阶段开始时过程所处的自然状况。它应能描述过程的特征并且无后效性, 即当某阶段的状态变量给定时, 这个阶段以后过程的演变与该阶段以前各阶段的状态无关。通常还要求状态是直接或间接可以观测的。

描述状态的变量称状态变量(state variable)。变量允许取值的范围称允许状态集合(set of admissible states)。用  $x_k$  表示第  $k$  阶段的状态变量, 它可以是一个数或一个向量。用  $X_k$  表示第  $k$  阶段的允许状态集合。在例 1 中  $x_2$  可取  $B_1, B_2$ , 或将  $B_i$  定义为  $i (i = 1, 2)$ , 则  $x_2 = 1$  或  $2$ , 而  $X_2 = \{1, 2\}$ 。

$n$  个阶段的决策过程有  $n + 1$  个状态变量,  $x_{n+1}$  表示  $x_n$  演变的结果。在例 1 中  $x_7$  取  $G$ , 或定义为  $1$ , 即  $x_7 = 1$ 。

根据过程演变的具体情况, 状态变量可以是离散的或连续的。为了计算的方便有时将连续变量离散化; 为了分析的方便有时又将离散变量视为连续的。

状态变量简称为状态。

#### 2.1.3 决策

当一个阶段的状态确定后, 可以作出各种选择从而演变到下一阶段的某个状态, 这种选择手段称为决策(decision), 在最优控制问题中也称为控制(control)。

描述决策的变量称决策变量(decision variable), 变量允许取值的范围称允许决策集合(set of admissible decisions)。用  $u_k(x_k)$  表示第  $k$  阶段处于状态  $x_k$  时的决策变量, 它是  $x_k$  的函数, 用  $U_k(x_k)$  表示  $x_k$  的允许决策集合。在例 1 中  $u_2(B_1)$  可取  $C_1, C_2$  或  $C_3$ , 可记作  $u_2(1) = 1, 2, 3$ , 而  $U_2(1) = \{1, 2, 3\}$ 。

决策变量简称决策。

#### 2.1.4 策略

决策组成的序列称为策略(policy)。由初始状态  $x_1$  开始的全过程策略记作  $p_{1n}(x_1)$ , 即

$$p_{1n}(x_1) = \{u_1(x_1), u_2(x_2), \dots, u_n(x_n)\}.$$

由第  $k$  阶段的状态  $x_k$  开始到终止状态的后部子过程的策略记作  $p_{kn}(x_k)$ , 即

$$p_{kn}(x_k) = \{u_k(x_k), \dots, u_n(x_n)\}, \quad k = 1, 2, \dots, n-1.$$

类似地, 由第  $k$  到第  $j$  阶段的子过程的策略记作

$$p_{kj}(x_k) = \{u_k(x_k), \dots, u_j(x_j)\}.$$

可供选择的策略有一定的范围，称为允许策略集合(set of admissible policies)，用  $P_{1n}(x_1), P_{kn}(x_k), P_{kj}(x_k)$  表示。

### 2.1.5. 状态转移方程

在确定性过程中，一旦某阶段的状态和决策为已知，下阶段的状态便完全确定。用状态转移方程(equation of state transition)表示这种演变规律，写作

$$x_{k+1} = T_k(x_k, u_k), k = 1, 2, \dots, n. \quad (1)$$

在例 1 中状态转移方程为  $x_{k+1} = u_k(x_k)$ 。

### 2.1.6. 指标函数和最优值函数

指标函数(objective function)是衡量过程优劣的数量指标，它是定义在全过程和所有后部子过程上的数量函数，用  $V_{k,n}(x_k, u_k, x_{k+1}, \dots, x_{n+1})$  表示， $k = 1, 2, \dots, n$ 。指标函数应具有可分离性，即  $V_{k,n}$  可表为  $x_k, u_k, V_{k+1,n}$  的函数，记为

$$V_{k,n}(x_k, u_k, x_{k+1}, \dots, x_{n+1}) = \varphi_k(x_k, u_k, V_{k+1,n}(x_{k+1}, u_{k+1}, \dots, x_{n+1}))$$

并且函数  $\varphi_k$  对于变量  $V_{k+1,n}$  是严格单调的。

过程在第  $j$  阶段的阶段指标取决于状态  $x_j$  和决策  $u_j$ ，用  $v_j(x_j, u_j)$  表示。指标函数由  $v_j(j = 1, 2, \dots, n)$  组成，常见的形式有：

阶段指标之和，即

$$V_{k,n}(x_k, u_k, x_{k+1}, \dots, x_{n+1}) = \sum_{j=k}^n v_j(x_j, u_j),$$

阶段指标之积，即

$$V_{k,n}(x_k, u_k, x_{k+1}, \dots, x_{n+1}) = \prod_{j=k}^n v_j(x_j, u_j),$$

阶段指标之极大（或极小），即

$$V_{k,n}(x_k, u_k, x_{k+1}, \dots, x_{n+1}) = \max_{k \leq j \leq n} (\min) v_j(x_j, u_j).$$

这些形式下第  $k$  到第  $j$  阶段子过程的指标函数为  $V_{k,j}(x_k, u_k, \dots, x_{j+1})$ 。

根据状态转移方程指标函数  $V_{k,n}$  还可以表示为状态  $x_k$  和策略  $p_{kn}$  的函数，即  $V_{k,n}(x_k, p_{kn})$ 。在  $x_k$  给定时指标函数  $V_{k,n}$  对  $p_{kn}$  的最优值称为最优值函数(optimal value function)，记为  $f_k(x_k)$ ，即

$$f_k(x_k) = \underset{p_{kn} \in P_{kn}(x_k)}{\text{opt}} V_{k,n}(x_k, p_{kn}),$$

其中  $\text{opt}$  可根据具体情况取  $\max$  或  $\min$ 。

### 2.1.7 最优策略和最优轨线

使指标函数  $V_{k,n}$  达到最优值的策略是从  $k$  开始的后部子过程的最优策略，记作  $p_{kn}^* = \{u_k^*, \dots, u_n^*\}$ 。 $p_{1n}^*$  是全过程的最优策略，简称最优策略(optimal policy)。从初始状态  $x_1 (= x_1^*)$  出发，过程按照  $p_{1n}^*$  和状态转移方程演变所经历的状态序列  $\{x_1^*, x_2^*, \dots, x_{n+1}^*\}$  称最优轨线(optimal trajectory)。

### 2.1.8 递归方程

如下方程称为递归方程

$$\begin{cases} f_{n+1}(x_{n+1}) = 0 \text{ 或 } 1 \\ f_k(x_k) = \underset{u_k \in U_k(x_k)}{\text{opt}} \{v_k(x_k, u_k) \otimes f_{k+1}(x_{k+1})\}, k = n, \dots, 1 \end{cases} \quad (2)$$

在上述方程中，当 $\otimes$ 为加法时取 $f_{n+1}(x_{n+1})=0$ ；当 $\otimes$ 为乘法时，取 $f_{n+1}(x_{n+1})=1$ 。动态规划递归方程是动态规划的最优性原理的基础，即：最优策略的子策略，构成最优子策略。用状态转移方程（1）和递归方程（2）求解动态规划的过程，是由 $k=n+1$ 逆推至 $k=1$ ，故这种解法称为逆序解法。当然，对某些动态规划问题，也可采用顺序解法。这时，状态转移方程和递归方程分别为：

$$\begin{cases} x_k = T_k^r(x_{k+1}, u_k), k = 1, \dots, n \\ f_0(x_1) = 0 \text{ 或 } 1 \\ f_k(x_{k+1}) = \underset{u_k \in U_{k+1}^r(x_{k+1})}{\text{opt}} \{v_k(x_{k+1}, u_k) \otimes f_{k-1}(x_k)\}, k = 1, \dots, n \end{cases}$$

例 3 用 lingo 求解例 1 最短路线问题。

```
model:
Title Dynamic Programming;
sets:
vertex/A,B1,B2,C1,C2,C3,C4,D1,D2,D3,E1,E2,E3,F1,F2,G/:L;
road(vertex,vertex)/A B1,A B2,B1 C1,B1 C2,B1 C3,B2 C2,B2 C3,B2 C4,
C1 D1,C1 D2,C2 D1,C2 D2,C3 D2,C3 D3,C4 D2,C4 D3,
D1 E1,D1 E2,D2 E2,D2 E3,D3 E2,D3 E3,
E1 F1,E1 F2,E2 F1,E2 F2,E3 F1,E3 F2,F1 G,F2 G/:D;
endsets
data:
D=5 3 1 3 6 8 7 6
6 8 3 5 3 3 8 4
2 2 1 2 3 3
3 5 5 2 6 6 4 3;
L=0,,,,,,,,,,,,,;
enddata
@for(vertex(i) | i#GT#1:L(i)=@min(road(j,i):L(j)+D(j,i)));
end
```

纵上所述，如果一个问题能用动态规划方法求解，那么，我们可以按下列步骤，首先建立起动态规划的数学模型：

- (i) 将过程划分成恰当的阶段。
- (ii) 正确选择状态变量 $x_k$ ，使它既能描述过程的状态，又满足无后效性，同时确定允许状态集合 $X_k$ 。
- (iii) 选择决策变量 $u_k$ ，确定允许决策集合 $U_k(x_k)$ 。
- (iv) 写出状态转移方程。
- (v) 确定阶段指标 $v_k(x_k, u_k)$ 及指标函数 $V_{kn}$ 的形式（阶段指标之和，阶段指标之积，阶段指标之极大或极小等）。
- (vi) 写出基本方程即最优值函数满足的递归方程，以及端点条件。

### §3 逆序解法的计算框图

以自由终端、固定始端、指标函数取和的形式的逆序解法为例给出计算框图，其它情况容易在这个基础上修改得到。

一般化的自由终端条件为

$$f_{n+1}(x_{n+1,i}) = \varphi(x_{n+1,i}), \quad i = 1, 2, \dots, n_{n+1} \quad (3)$$

其中  $\varphi$  为已知。固定始端条件可表示为  $X_1 = \{x_1\} = \{x_1^*\}$ 。

如果状态  $x_k$  和决策  $u_k$  是连续变量，用数值方法求解时需按照精度要求进行离散化。设状态  $x_k$  的允许集合为

$$X_k = \{x_{ki} \mid i = 1, 2, \dots, n_k\}, \quad i = 1, 2, \dots, n_k, k = 1, 2, \dots, n.$$

决策  $u_{ki}(x_{ki})$  的允许集合为

$$U_{ki} = \{u_{ki}^{(j)} \mid j = 1, 2, \dots, n_{ki}\}, \quad i = 1, 2, \dots, n_k, k = 1, 2, \dots, n.$$

状态转移方程和阶段指标应对  $x_k$  的每个取值  $x_{ki}$  和  $u_{ki}$  的每个取值  $u_{ki}^{(j)}$  计算，即  $T_k = T_k(x_{ki}, u_{ki}^{(j)})$ ,  $v_k = v(x_{ki}, u_{ki}^{(j)})$ 。最优值函数应对  $x_k$  的每个取值  $x_{ki}$  计算。基本方程可以表为

$$\begin{aligned} f_k^{(j)}(x_{ki}) &= v_k(x_{ki}, u_{ki}^{(j)}) + f_{k+1}(T_k(x_{ki}, u_{ki}^{(j)})), \\ f_k(x_{ki}) &= \operatorname{opt}_j f_k^{(j)}(x_{ki}), \\ j &= 1, 2, \dots, n_{ki}, i = 1, 2, \dots, n_k, k = n, \dots, 2, 1. \end{aligned} \quad (4)$$

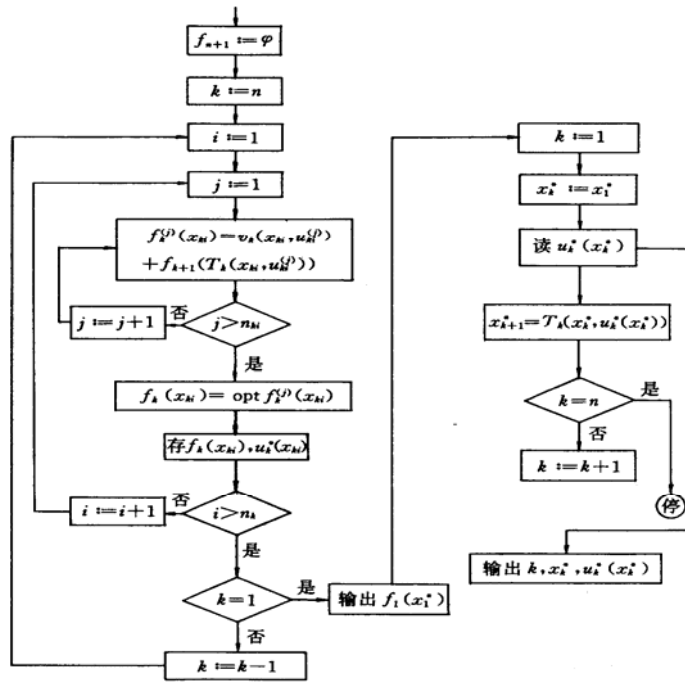


图2 解法框图

按照 (3), (4) 逆向计算出  $f_1(x_1^*)$ , 为全过程的最优值。记状态  $x_{ki}$  的最优决策为  $u_{ki}^*(x_{ki})$ , 由  $x_1^*$  和  $u_{ki}^*(x_{ki})$  按照状态转移方程计算出最优状态, 记作  $x_k^*$ 。并得到相应的最优决策, 记作  $u_k^*(x_k^*)$ 。于是最优策略为  $\{u_1^*(x_1^*), u_2^*(x_2^*), \dots, u_n^*(x_n^*)\}$ 。

算法程序的框图如图 2 所示。

图的左边部分是函数序列的递推计算, 可输出全过程最优值  $f_1(x_1^*)$ , 如果需要还可以输出后部子过程最优值函数序列  $f_k(x_{ki})$  和最优决策序列  $u_k^*(x_{ki})$ 。计算过程中存  $f_k(x_{ki})$  是备计算  $f_{k-1}$  之用, 在  $f_{k-1}$  算完后可用  $f_{k-1}$  将  $f_k$  替换掉; 存  $u_k^*(x_{ki})$  是备右边部分读  $u_k^*(x_k^*)$  之用。

图的右边部分是最优状态和最优决策序列的正向计算, 可输出最优策略  $\{u_1^*(x_1^*), u_2^*(x_2^*), \dots, u_n^*(x_n^*)\}$  和最优轨线  $\{x_1^*, x_2^*, \dots, x_n^*\}$ 。

#### §4 动态规划与静态规划的关系

动态规划与静态规划 (线性和非线性规划等) 研究的对象本质上都是在若干约束条件下的函数极值问题。两种规划在很多情况下原则上可以相互转换。

动态规划可以看作求决策  $u_1, u_2, \dots, u_n$  使指标函数  $V_{1n}(x_1, u_1, u_2, \dots, u_n)$  达到最优 (最大或最小) 的极值问题, 状态转移方程、端点条件以及允许状态集、允许决策集等是约束条件, 原则上可以用非线性规划方法求解。

一些静态规划只要适当引入阶段变量、状态、决策等就可以用动态规划方法求解。下面用例子说明。

例 4 用动态规划解下列非线性规划

$$\begin{aligned} \max \quad & \sum_{k=1}^n g_k(u_k); \\ \text{s.t.} \quad & \sum_{k=1}^n u_k = a, u_k \geq 0. \end{aligned}$$

其中  $g_k(u_k)$  为任意的已知函数。

解 按变量  $u_k$  的序号划分阶段, 看作  $n$  段决策过程。设状态为  $x_1, x_2, \dots, x_{n+1}$ , 取问题中的变量  $u_1, u_2, \dots, u_n$  为决策。状态转移方程为

$$x_1 = a, x_{k+1} = x_k - u_k, k = 1, 2, \dots, n.$$

取  $g_k(u_k)$  为阶段指标, 最优值函数的基本方程为 (注意到  $x_{n+1} = 0$ )

$$\begin{aligned} f_k(x_k) &= \max_{0 \leq u_k \leq x_k} [g_k(x_k) + f_{k+1}(x_{k+1})]; \\ 0 &\leq x_k \leq a, k = n, n-1, \dots, 2, 1; \\ f_{n+1}(0) &= 0. \end{aligned}$$

按照逆序解法求出对应于  $x_k$  每个取值的最优决策  $u_k^*(x_k)$ , 计算至  $f_1(a)$  后即可利用状态转移方程得到最优状态序列  $\{x_k^*\}$  和最优决策序列  $\{u_k^*(x_k^*)\}$ 。

与静态规划相比, 动态规划的优越性在于:

(i) 能够得到全局最优解。由于约束条件确定的约束集合往往很复杂, 即使指标函数较简单, 用非线性规划方法也很难求出全局最优解。而动态规划方法把全过程化为

一系列结构相似的子问题，每个子问题的变量个数大大减少，约束集合也简单得多，易于得到全局最优解。特别是对于约束集合、状态转移和指标函数不能用分析形式给出的优化问题，可以对每个子过程用枚举法求解，而约束条件越多，决策的搜索范围越小，求解也越容易。对于这类问题，动态规划通常是求全局最优解的唯一方法。

(ii) 可以得到一族最优解。与非线性规划只能得到全过程的一个最优解不同，动态规划得到的是全过程及所有后部子过程的各个状态的一族最优解。有些实际问题需要这样的解族，即使不需要，它们在分析最优策略和最优值对于状态的稳定性时也是很有用的。当最优策略由于某些原因不能实现时，这样的解族可以用来寻找次优策略。

(iii) 能够利用经验提高求解效率。如果实际问题本身就是动态的，由于动态规划方法反映了过程逐段演变的前后联系和动态特征，在计算中可以利用实际知识和经验提高求解效率。如在策略迭代法中，实际经验能够帮助选择较好的初始策略，提高收敛速度。

动态规划的主要缺点是：

(i) 没有统一的标准模型，也没有构造模型的通用方法，甚至还没有判断一个问题能否构造动态规划模型的准则。这样就只能对每类问题进行具体分析，构造具体的模型。对于较复杂的问题在选择状态、决策、确定状态转移规律等方面需要丰富的想象力和灵活的技巧性，这就带来了应用上的局限性。

(ii) 用数值方法求解时存在维数灾(curse of dimensionality)。若一维状态变量有  $m$  个取值，那么对于  $n$  维问题，状态  $x_k$  就有  $m^n$  个值，对于每个状态值都要计算、存储函数  $f_k(x_k)$ ，对于  $n$  稍大的实际问题的计算往往是不现实的。目前还没有克服维数灾的有效的一般方法。

## §5 若干典型问题的动态规划模型

### 5.1 最短路线问题

对于例 1 一类最短路线问题 (shortest Path Problem)，阶段按过程的演变划分，状态由各段的初始位置确定，决策为从各个状态出发的走向，即有  $x_{k+1} = u_k(x_k)$ ，阶段指标为相邻两段状态间的距离  $d_k(x_k, u_k(x_k))$ ，指标函数为阶段指标之和，最优值函数  $f_k(x_k)$  是由  $x_k$  出发到终点的最短距离 (或最小费用)，基本方程为

$$f_k(x_k) = \min_{u_k(x_k)} [d_k(x_k, u_k(x_k)) + f_{k+1}(x_{k+1})], k = n, \dots, 1;$$

$$f_{n+1}(x_{n+1}) = 0.$$

利用这个模型可以算出例 1 的最短路线为  $AB_1C_2D_1E_2F_2G$ ，相应的最短距离为 18。

### 5.2 生产计划问题

对于例 2 一类生产计划问题 (Production planning problem)，阶段按计划时间自然划分，状态定义为每阶段开始时的储存量  $x_k$ ，决策为每个阶段的产量  $u_k$ ，记每个阶段的需求量 (已知量) 为  $d_k$ ，则状态转移方程为

$$x_{k+1} = x_k + u_k - d_k, x_k \geq 0, k = 1, 2, \dots, n. \quad (5)$$

设每阶段开工的固定成本费为  $a$ ，生产单位数量产品的成本费为  $b$ ，每阶段单位数量产品的储存费为  $c$ ，阶段指标为阶段的生产成本和储存费之和，即

$$v_k(x_k, u_k) = cx_k + \begin{cases} a + bu_k, & u_k > 0 \\ 0 & u_k = 0 \end{cases} \quad (6)$$

指标函数  $V_{kn}$  为  $v_k$  之和。最优值函数  $f_k(x_k)$  为从第  $k$  段的状态  $x_k$  出发到过程终结的最小费用, 满足

$$f_k(x_k) = \min_{u_k \in U_k} [v_k(x_k, u_k) + f_{k+1}(x_{k+1})], k = n, \dots, 1.$$

其中允许决策集合  $U_k$  由每阶段的最大生产能力决定。若设过程终结时允许存储量为  $x_{n+1}^0$ , 则终端条件是

$$f_{n+1}(x_{n+1}^0) = 0. \quad (7)$$

(5) ~ (7) 构成该问题的动态规划模型。

### 5.3 资源分配问题

一种或几种资源(包括资金)分配给若干用户, 或投资于几家企业, 以获得最大的效益。资源分配问题(resource allocating Problem)可以是多阶段决策过程, 也可以是静态规划问题, 都能构造动态规划模型求解。下面举例说明。

例 5 机器可以在高、低两种负荷下生产。 $u$  台机器在高负荷下的年产量是  $g(u)$ , 在低负荷下的年产量是  $h(u)$ , 高、低负荷下机器的年损耗率分别是  $a_1$  和  $b_1$  ( $0 < b_1 < a_1 < 1$ )。现有  $m$  台机器, 要安排一个  $n$  年的负荷分配计划, 即每年初决定多少台机器投入高、低负荷运行, 使  $n$  年的总产量最大。如果进一步假设  $g(u) = \alpha u$ ,  $h(u) = \beta u$  ( $\alpha > \beta > 0$ ), 即高、低负荷下每台机器的年产量分别为  $\alpha$  和  $\beta$ , 结果将有什么特点。

解 年度为阶段变量  $k = 1, 2, \dots, n$ 。状态  $x_k$  为第  $k$  年初完好的机器数, 决策  $u_k$  为第  $k$  年投入高负荷运行的台数。当  $x_k$  或  $u_k$  不是整数时, 将小数部分理解为一年中正常工作时间或投入高负荷运行时间的比例。

机器在高、低负荷下的年完好率分别记为  $a$  和  $b$ , 则  $a = 1 - a_1$ ,  $b = 1 - b_1$ , 有  $a < b$ 。因为第  $k$  年投入低负荷运行的机器台数为  $x_k - u_k$ , 所以状态转移方程是

$$x_{k+1} = au_k + b(x_k - u_k) \quad (8)$$

阶段指标  $v_k$  是第  $k$  年的产量, 有

$$v_k(x_k, u_k) = g(u_k) + h(x_k - u_k) \quad (9)$$

指标函数是阶段指标之和, 最优值函数  $f_k(x_k)$  满足

$$f_k(x_k) = \max_{0 \leq u_k \leq x_k} [v_k(x_k, u_k) + f_{k+1}(x_{k+1})], \quad (10)$$

$$0 \leq x_k \leq m, \quad k = n, \dots, 2, 1.$$

及自由终端条件

$$f_{n+1}(x_{n+1}) = 0, \quad 0 \leq x_{n+1} \leq m. \quad (11)$$

当  $v_k$  中的  $g, h$  用较简单的函数表达式给出时, 对于每个  $k$  可以用解析方法求解极值问题。特别, 若  $g(u) = \alpha u$ ,  $h(u) = \beta u$ , (10) 中的  $[v_k(x_k, u_k) + f_{k+1}(x_k)]$  将是  $u_k$  的线性函数, 最大值点必在区间  $0 \leq u_k \leq x_k$  的左端点  $u_k = 0$  或右端点  $u_k = x_k$  取得, 即每年初将完好的机器全部投入低负荷或高负荷运行。

### §6 具体的应用实例

例 6 设某工厂有 1000 台机器, 生产两种产品  $A$ 、 $B$ , 若投入  $x$  台机器生产  $A$  产



品, 则纯收入为  $5x$ , 若投入  $y$  台机器生产  $B$  种产品, 则纯收入为  $4y$ , 又知: 生产  $A$  种产品机器的年折损率为  $20\%$ , 生产  $B$  种产品机器的年折损率为  $10\%$ , 问在  $5$  年内如何安排各年度的生产计划, 才能使总收入最高?

解 年度为阶段变量  $k=1,2,3,4,5$ 。

令  $x_k$  表示第  $k$  年初完好机器数,  $u_k$  表示第  $k$  年安排生产  $A$  种产品的机器数, 则  $x_k - u_k$  为第  $k$  年安排生产  $B$  种产品的机器数, 且  $0 \leq u_k \leq x_k$ 。

则第  $k+1$  年初完好的机器数

$$x_{k+1} = (1-0.2)u_k + (1-0.1)(x_k - u_k) = 0.9x_k - 0.1u_k \quad (12)$$

令  $v_k(x_k, u_k)$  表示第  $k$  年的纯收入,  $f_k(x_k)$  表示第  $k$  年初往后各年的最大利润之和。

显然

$$f_6(x_6) = 0 \quad (13)$$

则

$$\begin{aligned} f_k(x_k) &= \max_{0 \leq u_k \leq x_k} \{v_k(x_k, u_k) + f_{k+1}(x_{k+1})\} \\ &= \max_{0 \leq u_k \leq x_k} \{5u_k + 4(x_k - u_k) + f_{k+1}(x_{k+1})\} = \max_{0 \leq u_k \leq x_k} \{u_k + 4x_k + f_{k+1}(x_{k+1})\} \end{aligned} \quad (14)$$

(1)  $k=5$  时, 由 (13)、(14) 式得

$$f_5(x_5) = \max_{0 \leq u_5 \leq x_5} \{u_5 + 4x_5\}$$

$u_5 + 4x_5$  关于  $u_5$  求导, 知其导数大于零, 所以  $u_5 + 4x_5$  在  $u_5$  等于  $x_5$  处取得最大值, 即  $u_5 = x_5$  时,  $f_5(x_5) = 5x_5$ 。

(2)  $k=4$  时, 由 (12)、(14) 式得

$$\begin{aligned} f_4(x_4) &= \max_{0 \leq u_4 \leq x_4} \{u_4 + 4x_4 + 5x_5\} \\ &= \max_{0 \leq u_4 \leq x_4} \{u_4 + 4x_4 + 5(0.9x_4 - 0.1u_4)\} = \max_{0 \leq u_4 \leq x_4} \{0.5u_4 + 8.5x_4\} \end{aligned}$$

当  $u_4 = x_4$  时,  $f_4(x_4) = 9x_4$

(3)  $k=3$  时,

$$\begin{aligned} f_3(x_3) &= \max_{0 \leq u_3 \leq x_3} \{u_3 + 4x_3 + 9x_4\} \\ &= \max_{0 \leq u_3 \leq x_3} \{u_3 + 4x_3 + 9(0.9x_3 - 0.1u_3)\} = \max_{0 \leq u_3 \leq x_3} \{0.1u_3 + 12.1x_3\} \end{aligned}$$

当  $u_3 = x_3$  时,  $f_3(x_3) = 12.2x_3$

(4)  $k=2$  时,

$$f_2(x_2) = \max_{0 \leq u_2 \leq x_2} \{u_2 + 4x_2 + 12.2x_3\} = \max_{0 \leq u_2 \leq x_2} \{-0.22u_2 + 14.98x_2\}$$

当  $u_2 = 0$  时,  $f_2(x_2) = 14.98x_2$ 。

(5)  $k=1$  时,

$$f_1(x_1) = \max_{0 \leq u_1 \leq x_1} \{u_1 + 4x_1 + 14.98x_2\} = \max_{0 \leq u_1 \leq x_1} \{-0.498u_1 + 17.482x_1\}$$

当  $u_1 = 0$  时,  $f_1(x_1) = 17.482x_1$ 。因为

$$x_1 = 1000 \text{ (台)}$$

所以由 (12) 式, 进行回代得

$$x_2 = 0.9x_1 - 0.1u_1 = 900 \text{ (台)}$$

$$x_3 = 0.9x_2 - 0.1u_2 = 810 \text{ (台)}$$

$$x_4 = 0.9x_3 - 0.1u_3 = 648 \text{ (台)}$$

$$x_5 = 0.9x_4 - 0.1u_4 = 518.4 \text{ (台)}$$

注:  $x_5 = 518.4$  台中的 0.4 台应理解为有一台机器只能使用 0.4 年将报废。

例 7 求解下面问题

$$\begin{aligned} \max z &= u_1 u_2^2 u_3 \\ \begin{cases} u_1 + u_2 + u_3 = c & (c > 0) \\ u_i \geq 0 & i = 1, 2, 3 \end{cases} \end{aligned}$$

解: 按问题的变量个数划分阶段, 把它看作为一个三阶段决策问题。设状态变量为  $x_1, x_2, x_3, x_4$ , 并记  $x_1 = c$ ; 取问题中的变量  $u_1, u_2, u_3$  为决策变量; 各阶段指标函数按乘积方式结合。令最优值函数  $f_k(x_k)$  表示第  $k$  阶段的初始状态为  $x_k$ , 从  $k$  阶段到 3 阶段所得到的最大值。

$$\text{设 } x_3 = u_3, \quad x_3 + u_2 = x_2, \quad x_2 + u_1 = x_1 = c$$

则有

$$u_3 = x_3, \quad 0 \leq u_2 \leq x_2, \quad 0 \leq u_1 \leq x_1$$

用逆推解法, 从后向前依次有

$$f_3(x_3) = \max_{u_3=x_3} \{u_3\} = x_3 \text{ 及最优解 } u_3^* = x_3$$

$$f_2(x_2) = \max_{0 \leq u_2 \leq x_2} \{u_2^2 f_3(x_3)\} = \max_{0 \leq u_2 \leq x_2} \{u_2^2 (x_2 - u_2)\} = \max_{0 \leq u_2 \leq x_2} h_2(u_2, x_2)$$

$$\text{由 } \frac{dh_2}{du_2} = 2u_2 x_2 - 3u_2^2 = 0, \text{ 得 } u_2 = \frac{2}{3}x_2 \text{ 和 } u_2 = 0 \text{ (舍去)}$$

$$\text{又 } \frac{d^2 h_2}{du_2^2} = 2x_2 - 6u_2, \text{ 而 } \left. \frac{d^2 h_2}{du_2^2} \right|_{u_2=\frac{2}{3}x_2} = -2x_2 < 0, \text{ 故 } u_2 = \frac{2}{3}x_2 \text{ 为极大值点。}$$

$$\text{所以 } f_2(x_2) = \frac{4}{27}x_2^3 \text{ 及最优解 } u_2^* = \frac{2}{3}x_2。$$

$$f_1(x_1) = \max_{0 \leq u_1 \leq x_1} \{u_1 f_2(x_2)\} = \max_{0 \leq u_1 \leq x_1} \left\{ u_1 \frac{4}{27} (x_1 - u_1)^3 \right\}$$

$$\text{同样利用微分法易知 } f_1(x_1) = \frac{1}{64}x_1^4, \text{ 最优解 } u_1^* = \frac{1}{4}x_1。$$

由于  $x_1$  已知, 因而按计算的顺序反推算, 可得各阶段的最优决策和最优值。即

$$u_1^* = \frac{1}{4}c, \quad f_1(x_1) = \frac{1}{64}c^4$$

由

$$x_2 = x_1 - u_1^* = c - \frac{1}{4}c = \frac{3}{4}c$$

所以

$$u_2^* = \frac{2}{3}x_2 = \frac{1}{2}c, \quad f_2(x_2) = \frac{1}{16}c^3$$

由

$$x_3 = x_2 - u_2^* = \frac{3}{4}c - \frac{1}{2}c = \frac{1}{4}c$$

所以

$$u_3^* = \frac{1}{4}c, \quad f_3(x_3) = \frac{1}{4}c$$

因此得到最优解为:  $u_1^* = \frac{1}{4}c, \quad u_2^* = \frac{1}{2}c, \quad u_3^* = \frac{1}{4}c$ ;

$$\text{最大值为: } \max z = f_1(c) = \frac{1}{64}c^4.$$

#### 习 题 四

1. 用 Matlab 编程求例 6 的解。
2. 有四个工人, 要指派他们分别完成 4 项工作, 每人做各项工作所消耗的时间如表 1 所示。

表 1

工人 \ 工作	A	B	C	D
甲	15	18	21	24
乙	19	23	22	18
丙	26	17	16	19
丁	19	21	23	17

问指派哪个人去完成哪项工作, 可使总的消耗时间为最小? 试对此问题用动态规划方法求解。

3. 为保证某一设备的正常运转, 需备有三种不同的零件  $E_1, E_2, E_3$ 。若增加备用零件的数量, 可提高设备正常运转的可靠性, 但增加了费用, 而投资额仅为 8000 元。已知备用零件数与它的可靠性和费用的关系如表 2 所示。

表 2

备件数	增加的可靠性			设备的费用 (千元)		
	$E_1$	$E_2$	$E_3$	$E_1$	$E_2$	$E_3$
1	0.3	0.2	0.1	1	3	2
2	0.4	0.5	0.2	2	5	3
3	0.5	0.9	0.7	3	6	4

现要求在既不超出投资额的限制, 又能尽量提高设备运转的可靠性的条件下, 问各种零件的备件数量应是多少为好?

4. 某工厂购进 100 台机器, 准备生产 I、II 两种产品, 若生产产品 I, 每台机器每年可收入 45 万元, 损坏率为 65%; 若生产产品 II, 每台机器每年收入为 35 万元, 损坏率为 35%, 估计三年后将会有新型机器出现, 旧的机器将全部淘汰。试问每年应如何

安排生产，使在三年内收入最多？

5. 3 名商人各带 1 名随从乘船渡河，一只小船只能容纳 2 人，由他们自己划行。随从们密约，在河的任一岸，一旦随从人数比商人多，就杀商人。此密约被商人知道，如何乘船渡河的大权掌握在商人们手中，商人们怎样安排每次乘船方案，才能安全渡河呢？

6. 某一印刷厂有六项加工任务，对印刷车间和装订车间所需时间（单位：天）如表 3 所示，试求最优的加工顺序和总加工天数。

表 3

任务 车间	1	2	3	4	5	6
印刷车间	3	10	5	2	9	11
装订车间	8	12	9	6	5	2