5.5 第五章作业

5.5.1 广告促销问题

Q: 某公司的营销管理员因经营需要，正在考虑如何安排两种不想管产品的促销活动，已知在两种产品的促销水平上的决策变量要受到资源的约束，假设用 x1, x2 表示两种产品促销活动的水平，则相应的约束为 $4x_1 + x_2 \leq 20$ 和 $x_1 + 4x_2 \leq 20$, 随着广告促销水平的增加，广告活动的回报会减少，从而想要取得同样程度销售增加量，就必须付出更多的广告成本。为此，营销管理员经过分析发现，对于产品 1，当广告促销水平为 x 时，对应的收入为 $3x_1 - (x_1 - 1)^2$(百万元); 而产品 2 相应的收入为 $3x_2 - (x_2 - 2)^2$(百万元)。试为该公司确定两种产品广告促销水平的最优组合。

A: 分析如下。

为最大化两种产品的总收入，定义以下变量：

- x_1：产品 1 的促销水平。
- x_2：产品 2 的促销水平。

目标函数

最大化总收入：

$$\max z = \left[3x_1 - (x_1 - 1)^2\right] + \left[3x_2 - (x_2 - 2)^2\right]$$

简化后：

$$z = -x_1^2 + 5x_1 - x_2^2 + 7x_2 - 5$$

约束条件

1. 资源约束：

$$\begin{cases} 4x_1 + x_2 \leq 20 \\ x_1 + 4x_2 \leq 20 \end{cases}$$

2. 非负约束：

$$x_1 \geq 0, \quad x_2 \geq 0$$

Code Snippet 5.5.1 ▶ Matlab 代码

```matlab
% 非线性规划问题求解
clear; clc;
```

```

% 定义目标函数
fun = @(x) -(-x(1)^2 + 5*x(1) - x(2)^2 + 7*x(2) - 5); % 负号转为最小化

% 初始点
x0 = [0; 0];

% 线性约束
A = [4, 1; 1, 4];
b = [20; 20];

% 上下界
lb = [0; 0];
ub = [Inf; Inf];

% 求解
options = optimoptions('fmincon', 'Display', 'iter');
[x, fval] = fmincon(fun, x0, A, b, [], [], lb, ub, [], options);

% 输出结果
fprintf('最优促销水平: x1 = %.2f, x2 = %.2f\n', x(1), x(2));
fprintf('最大总收入: %.2f 百万元\n', -fval);
```

求解方法

采用非线性规划方法求解。由于目标函数为严格凹函数（二次项系数负），全局最大值存在于可行域内临界点。求解步骤如下：

1. 计算偏导数，求临界点：

$$\frac{\partial z}{\partial x_1} = -2x_1 + 5, \quad \frac{\partial z}{\partial x_2} = -2x_2 + 7$$

   解得 $x_1 = 2.5$，$x_2 = 3.5$。

2. 验证临界点是否满足约束：

$$4(2.5) + 3.5 = 13.5 \leq 20, \quad 2.5 + 4(3.5) = 16.5 \leq 20$$

3. 由于目标函数凹性，确认该点为全局最大。
4. 使用 MATLAB 的 `fmincon` 函数验证，自动求解非线性规划问题。

MATLAB 代码思路

MATLAB 代码通过以下步骤实现：

- **目标函数**：定义 z 的相反数（因 `fmincon` 最小化）。
- **约束条件**：设置线性约束矩阵 A 和 b，以及非负下界。
- **求解与输出**：调用 `fmincon`，输出最优解和最大收入。

实验结果

运行 MATLAB 代码，得到以下结果：

- **最优促销水平**：$x_1 = 2.5$，$x_2 = 3.5$。
- **最大总收入**：13.50 百万元。

结果验证

- **约束验证**：

$$4(2.5) + 3.5 = 13.5 \leq 20, \quad 2.5 + 4(3.5) = 16.5 \leq 20$$

- **收入验证**：

$$z = \left[3(2.5) - (2.5-1)^2\right] + \left[3(3.5) - (3.5-2)^2\right] = 5.25 + 8.25 = 13.50$$

- **边界点比较**：边界点（如 (0,0)、(5,0)、(0,5)、(4,4)）收入分别为-5、-5、5、11，均低于 13.50，确认最优。

5.5.2 最优利润问题

Q：按如下题意建立优化命题。设有数量为 x_1 的某种原料可用于生产两种产品 A 和 B。若以数量 y_1 投入生产 A, 剩下的 $x_1 - y_1$ 投入生产 B, 则利润为 $g(y_1) + h(x_1 - y_1)$, 其中 g, h 为已知函数且 $g(0) = h(0) = 0$。再设 y_1 和 $x_1 - y_1$ 投入生产 A 和 B 后, 可回收再利用, 回收率分别为 $a, b \in [0, 1]$, 因此在第一阶段生产后回收总量为 $x_2 = ay_1 + b(x_1 - y_1)$, 将 x_2 再投入生产 A 和 B, 然后再回收……, 这样一共生产了 n 次。希望选择 $y_1, y_2, \cdots y_n$ 使总利润最大。
A：分析如下。

为最大化 n 阶段生产与回收的总利润，定义以下变量：

- y_k：第 k 阶段分配给产品 A 的原料数量，$k = 1, 2, \dots, n$。
- x_k：第 k 阶段开始时的可用原料数量，x_1 为已知。

目标函数

最大化总利润：

$$\max z = \sum_{k=1}^{n} [g(y_k) + h(x_k - y_k)]$$

其中 $g(y_k)$ 和 $h(x_k - y_k)$ 为产品 A 和 B 的利润函数，且 $g(0) = h(0) = 0$。

约束条件

1. 回收约束：

$$x_{k+1} = ay_k + b(x_k - y_k), \quad k = 1, 2, \dots, n-1$$

   其中 $a, b \in [0, 1]$ 为回收率。

2. 分配约束：

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

Code Snippet 5.5.2 ▶ Matlab 代码

```matlab
% 多阶段利润优化问题动态规划求解
clear; clc;

% 参数
n = 3; % 阶段数
x1 = 100; % 初始原料量
a = 0.5; % 产品 A 回收率
b = 0.3; % 产品 B 回收率
g = @(y) y^0.5; % 利润函数 g
h = @(y) 2*y^0.5; % 利润函数 h

% 状态空间离散化
x_max = x1;
x_grid = 0:0.1:x_max;
V = zeros(length(x_grid), n+1); % 价值函数

% 动态规划：从后向前
```

```matlab
for k = n:-1:1
    for i = 1:length(x_grid)
        x = x_grid(i);
        y_vals = 0:0.1:x; % 离散化 y
        profits = zeros(size(y_vals));
        for j = 1:length(y_vals)
            y = y_vals(j);
            if k == n
                profits(j) = g(y) + h(x - y); % 最后阶段
            else
                x_next = a*y + b*(x - y);
                [~, idx] = min(abs(x_grid - x_next));
                profits(j) = g(y) + h(x - y) + V(idx, k+1);
            end
        end
        [V(i, k), idx] = max(profits);
    end
end

% 输出结果
[max_profit, idx] = max(V(:, 1));
fprintf('最大总利润: %.2f\n', max_profit);
fprintf('初始原料量: %.2f\n', x_grid(idx));
```

求解方法

采用动态规划方法求解。问题涉及多阶段决策和回收过程，动态规划通过状态变量和递推关系分解问题。求解步骤如下：

1. 定义状态变量：$V_k(x)$ 为从第 k 阶段开始，原料量为 x 时的最大总利润。
2. 建立递推关系：
$$V_n(x) = \max_{0 \le y \le x} [g(y) + h(x-y)]$$
$$V_k(x) = \max_{0 \le y \le x} [g(y) + h(x-y) + V_{k+1}(ay + b(x-y))], \quad k = 1, \dots, n-1$$
3. 从 $k = n$ 向前递推至 $k = 1$，计算 $V_1(x_1)$。

4. 使用 MATLAB 实现数值动态规划，离散化状态空间。

MATLAB 代码思路

MATLAB 代码通过以下步骤实现：

- **参数初始化**：设置 $n = 3$，$x_1 = 100$，$a = 0.5$，$b = 0.3$，利润函数 $g(y) = y^{0.5}$，$h(y) = 2y^{0.5}$。
- **状态空间**：离散化原料量 x，定义价值函数 $V_k(x)$。
- **动态规划**：从后向前计算每阶段最大利润，考虑回收后的原料量。
- **求解与输出**：输出最大总利润和初始原料量。

实验结果

使用示例函数 $g(y) = y^{0.5}$，$h(y) = 2y^{0.5}$，参数 $a = 0.5$，$b = 0.3$，$n = 3$，$x_1 = 100$ 运行，得到：

- **最大总利润**：43.83。
- **最优分配**：通过递推追踪 y_k^* 确定。

结果验证

- **约束验证**：确保 $0 \le y_k \le x_k$，$x_{k+1} = 0.5y_k + 0.3(x_k - y_k)$ 满足回收约束。
- **合理性检查**：$g(y) = y^{0.5}$，$h(y) = 2y^{0.5}$ 为凹函数，动态规划保证全局最优，满足 $g(0) = h(0) = 0$。