清华大学

决策方法论

2023 年秋季学期

作业 2

杨哲涵 2024 年 12 月 28 日

例题分析的深化 如果现在希望在五天的救灾活动之后, 仍然能保持不少于 350 辆的完好汽车. 那么可以修改代价函数, 通过一个因子 λ 严厉惩罚汽车数量少于 350 辆的情况.

更新后的贝尔曼方程如下

$$J_k(S_k) = \max_{x_k \in A_k(S_k)} (5S_k + 3x_k - \lambda \max(0, 350 - S_{k+1}) + J_{k+1}(0.9S_k - 0.2x_k))$$

我写了 Python 代码来模拟并计算最佳策略,λ 值被设置为 10000(只要够大就行). 经过计算, 最终结果如表所示:

表 1: 动态规划执行结果

Days	1	2	3	4	5	6
S_k	1000	900	810	714	500	350
x_k	0	0	75	713	500	

成功运送的物资为 23483.

此外, 如果取 $\lambda = 0$, 那么情形将退化到原始情况, 如果取 $\lambda \max(0, 350 - S_k)$ 中的 350 为其他数值, 则可计算其他希望的情况, 代码的通用性很好.

代码附后:

```
import numpy as np
def optimal_strategy(lambda_val):
    J = np.zeros((7, 1001))
    strategy = np.zeros((7, 1001))
    for k in range(6, 0, -1):
        for Sk in range(1000, -1, -1):
            max_value = -1
            best_xk = 0
            for xk in range(0, Sk + 1):
                if k == 6:
                    J[k][Sk] = 0
                    continue
                reward = 5 * Sk + 3 * xk - lambda_val * max(0, 350 - int(0.9 * Sk - 0.2 * xk))
                next_state = int(0.9 * Sk - 0.2 * xk)
                value = reward + J[k + 1][next_state]
                if value > max_value:
                    max_value = value
                    best_xk = xk
            J[k][Sk] = max_value
            strategy[k][Sk] = best_xk
    return J, strategy
lambda_val = 10000 # 0 表示允许少于 350
J_values, optimal_strategies = optimal_strategy(lambda_val)
print("Optimal total material transported:", J_values[1][1000])
s1=1000
s2=int(s1*0.9-optimal_strategies[1][s1]*0.2)
s3=int(s2*0.9-optimal_strategies[2][s2]*0.2)
s4=int(s3*0.9-optimal_strategies[3][s3]*0.2)
s5=int(s4*0.9-optimal_strategies[4][s4]*0.2)
s6=int(s5*0.9-optimal_strategies[5][s5]*0.2)
print(s1,s2,s3,s4,s5,s6)
print(optimal_strategies[1][s1],optimal_strategies[2][s2],optimal_strategies[3][s3],
optimal_strategies[4][s4],optimal_strategies[5][s5])
```