

好的，我们来到了规划类问题的最后一章，动态规划。其实说是最后一章，倒不如说这是对前面三章规划的总结融合，采用最适合的解法去解决规划问题。

1 动态规划-简介

动态规划(Dynamic Programming)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）。动态规划是一种通过将复杂问题分解为一组更简单的子问题来解决复杂问题的方法。

八股文：动态规划是运筹学的一个分支，通常用来解决多阶段决策过程最优化问题。动态规划的基本想法就是将原来的原问题转换为一系列互相联系的子问题，然后通过逐层地推来求得最后的解。目前，动态规划常常出现在算法竞赛或者笔试面试中，在数学建模中应用较少。

1.1 已知的结果下，简单的解法

某厂为扩大生产能力，拟订购某成套4-6套，以分配给其所管辖的1, 2, 3个分厂使用。预计各分厂分得不同套数的设备后，每年创造的利润（万元）如下表所示。该厂应订购几套设备并如何分配，才能使每年预计创造利润总额最大？

表 1. 不同分厂套数产生利润

利润 \ 套数 分厂	0套	1套	2套	3套	4套	5套	6套
1	0	3	5	6	7	6	5
2	0	4	6	7	8	9	10
3	0	2	5	9	8	8	7

解：

$$\text{目标函数 } \max f(x) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

$$\text{其中 } x_{ij} = \begin{cases} 0 & (x_i \neq j \text{套}) \\ 1 & (x_i = j \text{套}) \end{cases}$$

同时给出约束条件

$$\text{s.t.} \begin{cases} \sum_{i=0}^m \sum_{j=0}^n x_{ij} = 3 \\ \text{总套数不超过4-6套} \end{cases}$$

2 动态规划-适用题目

2.1 动态规划适用的赛题：

最常见的题目大概就是贪心算法一类问题，主要包括最短路径，背包问题，硬币问题，

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。

2.2 模型假设

1. 工厂产品生产示例1：

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 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：

代码 1. Example1

```
1      %This is functionk.m
2  function [max, uk]=functionk(k_Add_1_xk,k)
3  % k_Add_1_xk=518.4;
4  % k=5;
5      if k == 6
6          par_u = 1;
7          par_x = 4;
8      else
9          par_u = 1 - k_Add_1_xk * 0.1;
10         par_x = 4 + k_Add_1_xk * 0.9;
11     end
12     [max, uk]= maxvalue(par_u, par_x)
13 end
14      %This is maxvalue.m
15 function [par,key]=maxvalue(par_u, par_x)
16     if par_u>=0
17         par=par_u+par_x;
18         key=1;
19     else
20         par=par_x;
```

```

21     key=0;
22     end
23 end
24     %This is Example1.m
25 clc
26 clear
27 k.Add_1_xk = 0;
28 max = 0;
29 uk = [];
30 for i = [6:-1:1];
31     [k.Add_1_xk, u_k] = functionk(k.Add_1_xk, i);
32     uk=[uk u_k];
33 end
34 uk=flip(uk);
35 maxprofit = k.Add_1_xk*1000;
36 x = 1000;
37 x_last = 0;
38 x_last = x;
39 number=[x_last];
40 for i = 2:5
41     x = 0.9 * x - 0.1 * uk(i) * x_last;
42     x_last = x;
43     number=[number x_last];
44 end
45 disp('The number of intact machines from the first year is:');
46 number
47 disp(strcat('maxvalue is ',num2str(maxprofit)))
48 %The number of intact machines from the first year is:
49 %number = 1.0e+03 *
50 %    1.0000    0.9000    0.8100    0.6480    0.5184
51 %The maxvalue is 19733.8

```

2. 最短路径寻找示例2:

下图是一个线路网，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。

这种题的解法是较为类似的，我们可以直接套用模板。该模板可以适用于无约束的最短路径这一类问题。求解步骤:

- (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_k 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。

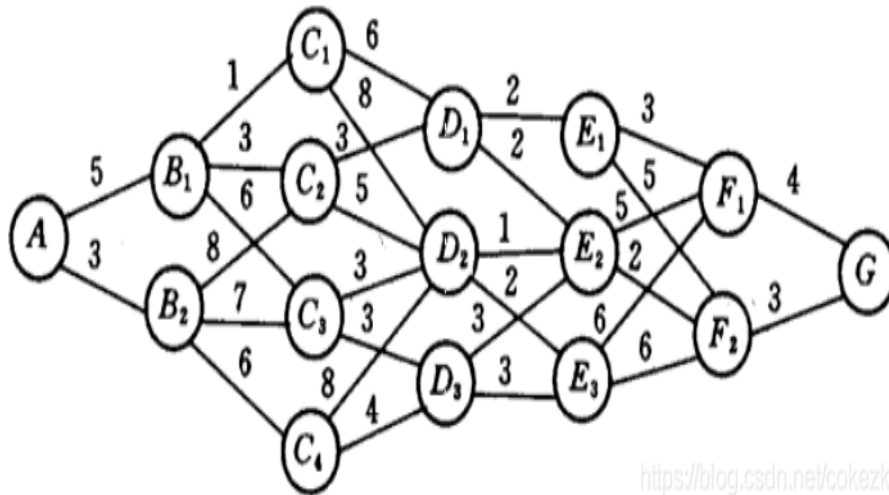


图 1. 示例2

解:

代码 2. Example2

```

1      % This is Example2.m
2      clc,clear
3      now=[5,3,0,0 ,0,0,0,0 ,0,0,0,0 ,0,0,0,0
4          1,3,6,0 ,0,8,7,6 ,0,0,0,0 ,0,0,0,0
5          6,8,0,0 ,3,5,0,0 ,0,0,0,0 ,0,0,0,0
6          2,2,0,0 ,1,2,0,0 ,3,3,0,0 ,0,0,0,0
7          3,5,0,0 ,5,2,0,0 ,6,6,0,0 ,0,0,0,0
8          4,0,0,0 ,3,0,0,0 ,0,0,0,0 ,0,0,0,0
9          0,0,0,0 ,0,0,0,0 ,0,0,0,0 ,0,0,0,0];
10     % Distance matrix, the i-th row represents the i-th stage
11     % (the last stage is all zeros);
12     % n*m columns, n starting points,
13     % m different choices, next starting point starts after m numbers
14     h=7 ;
15     % Number of stages
16     a=4 ;

```

```
17 % Number of starting points
18 b=4;
19 % Number of decisions ( usually equal to the number of starting points )
20 % Only need to modify the code above %
21 % to nest it into other shortest path calculations %
22
23
24 now(now==0)=inf;
25 % First, convert it into a full matrix, use 00 to represent no route,
26 % then convert 0 to infinity
27 road_sum=zeros(h,a*b);
28 % Save the shortest distance of each stage
29
30 for i=h-1:-1:1
31 % Different stages
32     for n=1:a
33         % Different starting points in each stage
34         for k=1:b
35             % Different decisions in each starting point
36 road_sum(i,(n-1)*a+k)=now(i,(n-1)*a+k)+min(road_sum(i+1,(k-1)*a+1:k*a));
37 % Current path plus the shortest path in the previous stage
38         end
39     end
40 end
41
42 road=[];
43 % Used to save the selected starting points
44 for i=1:h-1
45     if i==1
46         start=find(road_sum(1,:)==min(road_sum(1,:)));
47 % Find the first point after the starting point
48     else
49 start=find(road_sum(i,(start-1)*a+1:start*a)==min(road_sum(i,(start-1)*a+1:start*a)));
50 % Find the first point after the starting point
51     end
52     if length(start)>0
53         disp(' Note: There are multiple optimal solutions ...
54             for this problem. ...
55             An automatic path has been planned for you.')
56         start=start(1);
57     end
58     road=[road,start] ; % Save
```

```
59 end
60
61 disp(['The shortest path sum is: ' num2str(min(road_sum(1,:)))])
62 disp('One of the corresponding path choices is: ...
63 starting from the starting point,')
64 for i=road
65 disp(['choose the next starting point ' num2str(i) ','])
66 end
67 %One of the shortest is 18,A->B1->C2->D1->E2->F2->G
```

3 动态规划-代码实现

代码 3. Example

```
1 %This is example.m
2 num=input('please input the number of vehicles:');
3 a1=0;b2=0;c3=0;
4 max=0;
5 for i=0:4
6     for j=0:4
7         for k=0:4
8             x1=i;x2=j;x3=k;
9             if (x1+x2+x3==num)
10                 switch x1
11                     case 0
12                         a1=0;
13                         ...
14                     case 6
15                         a1=5;
16                     otherwise
17                         end
18                 switch x2
19                     case 0
20                         b2=0;
21                         ...
22                     case 6
23                         b2=10;
24                     otherwise
25                         end
26                 switch x3
27                     case 0
```

```
28         c3=0;
29         ...
30     case 6
31         c3=7;
32     otherwise
33     end
34     if max<=a1+b2+c3
35         max=a1+b2+c3;
36         disp(x1);
37         disp(x2);
38         disp(x3);
39         disp(max);
40         disp('Over');
41     end
42 end
43 end
44 end
45 end
```

4 动态规划-实战演练

1998年数学建模B题《建模案例：最佳灾情巡视路线》最短路径规划。

References

- [1] 谢中华. MATLAB与数学建模[A].北京航空航天大学出版社[M]:科学技术协会,2021-02-14.
- [2] 狗头狗不狗. 数学建模（7）动态规划以及matlab实现[M].https://blog.csdn.net/qq_43649786/article/details/98843408:2022-02-28
- [3] tutou20082008. 数学建模98的题,找高手指点[M].<https://blog.csdn.net/tutou20082008/article/details/5816135>,2010-08-16.
- [4] 风中的余韵.matlab 动态规划的一道关于具体的应用实例的小题[M].<https://blog.csdn.net/Zhangxy555/article/details/126024099>,2022-07-27.
- [5] 请缨提旅 .matlab数学建模工具箱，国赛必备[L].https://www.ilovematlab.cn/thread-259600-1-1.html?_dsign=77f9daf3,2013-8-19.
- [6] cokezkx.数学建模（一）规划问题[B].<https://blog.csdn.net/cokezkx/article/details/97778450>, 2019-07-30.

5 数学建模算法大全第四章习题答案

1. 直接给出解: x_i, u_i 分别代表第*i*年投入的A,B两种产品的生产:

$$x_1 = 1000, x_2 = 900, x_3 = 0, x_4 = 0, x_5 = 0$$

$$u_1 = 0, u_2 = 0, u_3 = 810, u_4 = 648, u_5 = 518.4$$

- 2.