# 第四章 动态规划

\*动态规划是解决多阶段决策过程的最优化问题的一种方法。所谓多阶段决策过程是指这样一类决策过程，由于它的特殊性，我们可以把它按时间分为若干阶段(有时称为“时段”)，而在每一个阶段都需要作出决策，以便使整个过程取得最佳的经济效益。在多阶段决策问题中，各个阶段采取的决策一般说来是与时间有关系的。而且，前一阶段采取的策略如何，不但与该阶段的经济效益有关系，更直接影响以后各阶段的经济效益。可见，它是一个动态的问题，因此，处理的方法称为动态规划方法。

\*动态规划也可以用来处理一些本来与时间没有关系的静态模型，这只要在静态模型中人为地引进“时间”因素，分成时段，就可以把它看作多阶段的动态模型，用动态规划方法去处理。

## §4.1 最短路径问题与最优化原则

### 一、问题：

所谓最短路径问题是指给定始点和终点，并知道由始点到终点的各种可能的路径，问题是要找一条由始点到终点的最短路，即长度最短的路。

### 二．例子：

图4.1是表示由始点A到终点E的路线图。由A至B()是第一阶段；由某至C是第二阶段；由某至D是第三阶段；由某至E是第四阶段。

引进几个符号和概念：

(1) n表示由某点(例如点之间的阶段数。例如：A到E的阶段数

(2) S表示在任一阶段所处的位置，称S为状态变量(S可取

2,3)。

(3) 称为决策变量，它表示当处于状态S还有n个阶段要走时，下一步所选取的点(例如：当时，它表示现在处于点，还有三个阶段要走，下一步选取点，即要走的路线)。

(4)表示现在处于状态S，还有n个阶段要走，由S至终点E的最短距离(例如：表示现在处于点A，还有4个阶段要走，由A到终点E的最短距离)。

(5)表示点S到下一步所选的点的距离(例如：

表示由A到的距离，即)。

分析：

由A到B的第一阶段有三种走法：

(1)，若选此走法，最短距离为

(2)，若选此走法，最短距离为

(3)，若选此走法，最短距离为

应当在上面三种走法中选取距离最小者，也即

要求，必须先算出。

现在算。有两种走法：

(1)，若选此走法，最短距离为

(2)，若选此走法，最短距离为

应当在上面两种走法中选取距离最小者。

同理，的算法如下：

由于

从后往前倒推：

时，

，路径：

，路径：

时，

，路径：

，路径：

，路径：

时，

，路径：

，路径：

，路径：

时，

，路径：

在例1中，我们主要利用了n个阶段与个阶段的关系式(称为递推关系式或函数方程)：

而时，有

上述递推关系式可以用“最优化原则”来描述：

一个过程的最优策略具有这样的性质，即无论其初始状态和初始决策如何，其今后诸决策对以第一个决策所形成的状态作为初始状态的过程而言，必须构成最优策略。

当处于状态S，还有n个阶段要走，由S至终点E的最短距离应该是：当第一步已采取决策(即走法)后，以后按最优路线走个阶段的总距离中的最小者。

## §4.2 动态规划的算法本质

1.自上而下分治策略的复杂性

2.自下而上的动态规划填表法

例1：以上例为例解释

例2：世界系列差别赛

1.问题：有两个队A和B进行一系列比赛，看哪个队先赢得n场比赛。

设是A队需要再赢i场比赛(则A队赢), B队需要再赢j场比赛(则B队赢)，这时A队赢的概率。

我们假设A和B队有相同的竞争力，各有50%赢某场的概率。

例如：，A队需要再赢场，B队需要再赢场，则

。

2.的递归式：

，

(4.1)

，

解以上递归方程，令，是调用的最大次数，由(4.1)式

解得 。即。

但递归计算有重复计算。例如：计算需要计算和

，而计算都需要计算。

3.列表计算(见图4.2):

\*解释计算规则

4.算法：

FUNCTION Odds (i, j : integer) : real;

VAR s, k : integer;

BEGIN

FOR s := 1 TO i+j DO

BEGIN

P[0, s] := 1.0;

P[s, 0] := 0.0;

FOR k := 1 TO s-1 DO

P[k, s-k] := (P[k-1, s-k] + P[k, s-k-1])/2.0;

END;

RETURN (P[i, j]);

END;

5.算法分析：

总时间：。

## §4.3 动态规划举例—Halin图中的TSP问题

### 一．Halin图的定义

1.Halin图的定义：

在平面上嵌入一棵树T，T的每个内部顶点的度数至少为3，并且至少有一个内部顶点。作一个圈C连接T的所有叶顶点。T的所有叶顶点组成C上的所有顶点。这样得到的平面图称为Halin图。树T称为Halin图的特征树，圈C称为Halin图的伴随圈。(见图4.3)

2.轮

设是一个Halin图，其中树T是一颗星，即单个结点v连接到其它互不相邻的n个顶点，那么H就称为轮。一个轮是最简单的Halin图，记为。(见图4.4)

3. 扇

假设T至少有两个非叶子顶点，w是树T中任意一个非叶子顶点，而且w仅与T中一个非叶子顶点相邻。不妨令w相邻的叶子顶点集合为C(w)，注意到这些顶点在圈C上是一段连续的顶点。我们将导出子图称为一个扇，w为扇的中心。(见图4.3)

4.扇收缩

假设F是Halin图的一个扇，则把图H由收缩扇F成为一个伪点，所得到的图记为表示把H中的扇收缩成一点后的图。

，被定义如下：

(1)如果一条边的两个端点都在F中，那么删除该边；

(2)如果一条边的两个端点都在中，那么这条边保持不变；

(3)如果一条边的一个端点在中，另一个端点在F中，那么现在这条边在中的端点不变，另一端点为。

\*注意：

(1)任一Halin图H如果包含一个扇F，那么H收缩扇F后得到的图仍是一个Halin图。(见图4.3)

(2)设是一个Halin图，反复收缩H的扇，最后必然得到一个轮。

(3)存在求Halin图的特征树和伴随圈的O(n)时间的算法，该算法也是判定一个图是否Halin图的算法。

### 二．求赋权Halin图中的旅行售货员问题(TSP)的算法

1.TSP问题：设G是一个图，对任意边上有一个正的权

，求G中一个哈密顿圈C，使得C上的权在G的所有哈密顿圈中最小。

\*TSP问题对一般图而言是NP-难的，但在Halin图上，却有O(n)时间的算法求解。

2.在Halin图上求解TSP问题的算法：

设H是一个赋权Halin图。

(2.1)若H是一个轮(如图4.5)

令

则最短哈密顿圈：

(2.2)若H不是轮，则H包含一个扇F(见图4.6)

令

收缩扇F，得。给H’的各边一个新的赋权如下，其中表示原来H的边j上的权值，为H’的边h上的新权值。

那么，H’中最短哈密顿圈的权与H中最短哈密顿圈的权相等。

\*举例说明：

算法：

1.输入赋权Halin图；

2.以T的某个内部顶点v为根，后序遍历T;

3.每当遍历到一个扇的中心w时，与w相邻的叶子已遍历完。令为扇，构造，按(2.2)方法给H’的边赋权。若H’是轮则执行步骤4；否则继续后序遍历T, 执行步骤3；

4.若H’是轮，按照(2.1)的方法，求H’的最短哈密顿圈C’；

5.按上述过程的逆序过程，逐层将H’中的最短哈密顿圈C’恢复成H中的最短哈密顿圈C”，最后求得原图H中的最短哈密顿圈。