国家出版基金项目
NATIONAL PUBLICATION FOUNDATION

“十三五”国家重点图书出版规划项目

排序与调度丛书（二期）

排序问题的动态规划方法

柏孟卓 张新功 编著

清華大学出版社
北京

内 容 简 介

本书系统地介绍了排序理论和动态规划理论方面的研究成果，讨论动态规划方法在解决排序与调度问题中的应用。本书讨论了单机排序问题、分批排序问题、成组加工排序问题、可控排序问题、可拒绝排序问题、若干供应链排序问题以及双代理排序问题的动态规划解法，并介绍了利用动态规划算法设计完全多项式时间近似方案（FPTAS）的应用成果。读者通过本书可以对动态规划在排序问题中的应用有一个全面的了解和认识。

本书可以作为运筹与管理、计算机、自动化等相关学科的教师和学生的参考书，也适合对排序领域有兴趣的读者阅读。

图书在版编目（CIP）数据

排序问题的动态规划方法/柏孟卓，张新功编著. —北京：清华大学出版社，2023.8
（排序与调度丛书. 二期）
ISBN 978-7-302-64220-6

Ⅰ. ①排… Ⅱ. ①柏… ②张… Ⅲ. ①排序 Ⅳ. ①O223

中国国家版本馆 CIP 数据核字（2023）第 134195 号

责任编辑：汪 操
封面设计：常雪影
责任校对：欧 洋
责任印制：杨 艳

出版发行：清华大学出版社
网 址：http://www.tup.com.cn，http://www.wqbook.com
地 址：北京清华大学学研大厦 A 座 **邮 编**：100084
社 总 机：010-83470000 **邮 购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：三河市龙大印装有限公司
经 销：全国新华书店
开 本：170mm×240mm **印张**：9.75 **字 数**：183 千字
版 次：2023 年 8 月第 1 版 **印 次**：2023 年 8 月第1 次印刷
定 价：69.00 元

产品编号：098293-01

《排序与调度丛书》编辑委员会

丛书序言

我知道排序问题是从 20 世纪 50 年代出版的一本名为 *Operations Research*（《运筹学》，可能是 1957 年出版）的书开始的。书中讲到了 S. M. 约翰逊（S. M. Johnson）的同顺序两台机器的排序问题并给出了解法。约翰逊的这一结果给我留下了深刻的印象。第一，这个问题是从实际生活中来的。第二，这个问题有一定的难度，约翰逊给出了完整的解答。第三，这个问题显然包含着许多可能的推广，因此蕴含了广阔的前景。在 1960 年左右，我在《英国运筹学》（季刊）（当时这是一份带有科普性质的刊物）上看到一篇文章，内容谈到三台机器的排序问题，但只涉及四个工件如何排序。这篇文章虽然很简单，但我也从中受到一些启发。我写了一篇讲稿，在中国科学院数学研究所里做了一次通俗报告。之后我就到安徽参加“四清”工作，不意所里将这份报告打印出来并寄了几份给我，我寄了一份给华罗庚教授，他对这方面的研究给予很大的支持。这是 20 世纪 60 年代前期的事，接下来便开始了“文化大革命”，倏忽十年。20 世纪 70 年代初我从“五七”干校回京，发现国外学者在排序问题方面已做了不少工作，并曾在 1966 年开了一次国际排序问题会议，出版了一本论文集 *Theory of Scheduling*（《排序理论》）。我与韩继业教授做了一些工作，也算得上是排序问题在我国的一个开始。想不到在秦裕瑗、林诒勋、唐国春以及许多教授的努力下，跟随着国际的潮流，排序问题的理论和应用在我国得到了如此蓬勃的发展，真是可喜可贺！

众所周知，在计算机如此普及的今天，一门数学分支的发展必须与生产实际相结合，才称得上走上了健康的道路。一种复杂的工具从设计到生产，一项巨大复杂的工程从开始施工到完工后的处理，无不牵涉排序问题。因此，我认为排序理论的发展是没有止境的。我很少看小说，但近来我对一本名叫《约翰・克里斯托夫》的作品很感兴趣。这是罗曼・罗兰写的一本名著，实际上它是以贝多芬为背景的一本传记体小说。这里面提到贝多芬的祖父和父亲都是宫廷乐队指挥，当贝多芬的父亲发现他在音乐方面是个天才的时候，便想将他培养成一名优秀的钢琴师，让他到各地去表演，可以名利双收，所以强迫他勤学苦练。但贝多芬非常反感，他认为这样的作品显示不出人的气质。由于贝多芬有如此的感受，他才能谱出如《英雄交响曲》《第九交响曲》等深具人性的伟大

乐章。我想数学也是一样，只有在人类生产中体现它的威力的时候，才能显示出数学这门学科的光辉，也才能显示出作为一名数学家的骄傲。

任何一门学科，尤其是一门与生产实际有密切联系的学科，在其发展初期那些引发它成长的问题必然是相互分离的，甚至是互不相干的。但只要研究继续向前发展，一些问题便会综合趋于统一，处理问题的方法也会与日俱增、深入细致，可谓根深叶茂，蔚然成林。我们这套丛书已有数册正在撰写之中，主题纷呈，蔚为壮观。相信在不久以后会有不少新的著作出现，使我们的学科呈现一片欣欣向荣、繁花似锦的局面，则是鄙人所厚望于诸君者矣。

越民义

中国科学院数学与系统科学研究院

2019 年 4 月

前　言

排序问题是一类重要的组合最优化问题，是运筹学研究的一个非常活跃的分支，广泛地应用于管理科学、计算机科学、工程技术、制造业、运输业、分派销售和其他服务行业。它研究如何在有限的资源限制和约束下对于给定的一些“工件”或“活动”从时间上和顺序上进行合理的安排和分配，以使某目标（如生产效率、资源利用率和合格率等）达到或接近最优。

动态规划是运筹学的一个重要分支。动态规划方法是研究多阶段决策过程最优化的一种数学方法，通过把多阶段过程划分为一系列相互联系的单阶段过程，再逐个阶段求解，从而使整个过程达到目标最优。动态规划方法在工程技术、经济管理、工业生产和军事等方面都有着广泛的应用。动态规划方法没有统一的标准模型，没有统一的处理格式。它必须依据问题本身的特性，利用灵活的数学技巧来处理。在排序与调度领域中，存在大量 NP 难的多阶段决策问题，用动态规划方法求得精确最优解是非常有效的方法之一。

现有讨论排序问题的书籍中，绝大多数是从问题的模型角度进行分类研究，很少从解决问题的方法角度展开讨论。本书系统地介绍了排序理论和动态规划理论方面的研究成果，讨论动态规划方法在解决排序与调度问题中的应用。从众多用动态规划方法求解的排序问题中选取有代表性的部分问题，进行总结和分析。读者通过本书可以对动态规划在排序问题中的应用有全面的了解和认识。

本书共分 7 章。第 1 章介绍动态规划的基础知识；第 2 章介绍排序问题的基本理论；第 3 章讨论经典的单机排序问题的动态规划求解方法；第 4 章研究若干新型排序问题的动态规划解法，其中包括分批排序问题、成组加工排序问题、可控排序问题以及可拒绝排序问题；第 5 章讨论如何用动态规划方法解决供应链排序问题；第 6 章研究双代理排序问题的动态规划解法；第 7 章讨论利用动态规划算法设计完全多项式时间近似方案（FPTAS）的应用成果。其中第 1 章、第 5 章由沈阳师范大学柏孟卓撰写，第 6 章、第 7 章由重庆师范大学张新功撰写，第 2 章至第 4 章由柏孟卓、张新功共同撰写。

本书内容既包含了用动态规划方法求解排序问题的经典研究成果，也包含

了作者多年来研究工作积累的成果。本书从最初的构想到最终的成稿，一直得到唐国春教授的大力推动与悉心指导。此外，我们还得到了清华大学出版社编辑的细心帮助。在此向唐国春教授和清华大学出版社表示深深的感谢！

本书可以作为运筹与管理、计算机和自动化等相关学科的教师和学生的参考书。由于作者时间有限，书中或会有不妥之处，恳请读者批评指正！

作　者

2023 年 1 月

目　录

第 1 章　动态规划基础

动态规划(Dynamic Programming)是研究多阶段决策过程最优化的一种数学方法,是解决组合最优化问题的一个重要方法,在工程技术、经济管理、工业生产和军事等方面都有着广泛的应用。1951 年,美国数学家 R. Bellman 等人根据一类多阶段决策问题的特点,提出了解决这类问题的"最优化原理",并研究了许多实际问题和数学模型,从而建立了数学规划的一个新分支——动态规划。

1.1　多阶段决策过程

在现实生活中,常常会遇到这样一类问题,其整个活动的过程具有明显的阶段性和序列性,而对整个活动过程的控制也往往分阶段进行决策。因此,可将过程分成相互联系的若干阶段,在它的每一阶段都要做出决策,从而使整个过程达到最佳的活动效果。所以,各个阶段的决策不能任意选取,它既依赖于活动过程当前的状态,又影响过程的后续发展。例如最短路问题,在每个节点都要对下一步的路径做出决策,这种决策依赖于前序节点的选择,又影响着后继节点的选择。当各个阶段决策确定后,就组成了一个决策序列,也就确定了整个过程的一条活动路线。这种把一个问题看作一个前后关联、具有链状结构的多阶段过程就称为多阶段决策过程(图 1-1),这种问题称为多阶段决策问题。

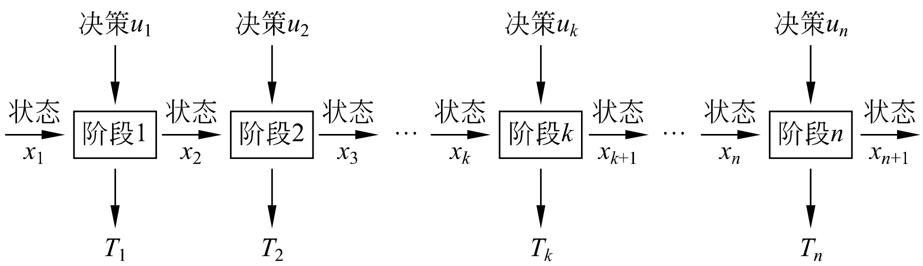

图 1-1

在多阶段决策问题中,每个阶段所做出的决策,一般来说与时间有关,决策依赖于系统当前的状态,做出决策后又引起状态的转变,因此需要在系统发展的不同时刻根据系统所处的状态不断进行决策,最后得出整个过程的最优决策。一个决策序列就是在变化的状态中产生出来的,因此具有"动态"的含义,

称这种解决多阶段决策最优化的过程为动态规划方法。另外,一些与时间无关的问题,也可以人为划分阶段,转化为多阶段决策过程。

下面我们给出几个多阶段决策问题的实例。

例 1.1　最短路问题

在图 1-2 所示的有向网络中,每条弧上的数字表示弧的两个顶点之间距离,求从起点 A 到终点 E 的一条最短路径。

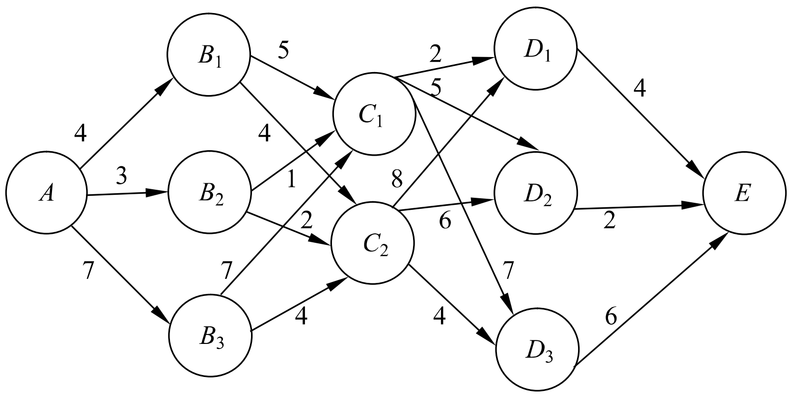

图 1-2

可以将从 A 到 E 的路划分为 4 个阶段:第 1 阶段需要选择从 A 到 B_1, B_2, B_3 的弧;第 2 阶段需要选择从 B_1, B_2, B_3 到 C_1, C_2 的弧;第 3 阶段需要选择从 C_1, C_2 到 D_1, D_2, D_3 的弧;第 4 阶段需要选择从 D_1, D_2, D_3 到 E 的弧。因此,这是一个 4 阶段决策过程。

如果不采用多阶段决策的方法,本题可以用穷举法求解,即把从起点 A 到终点 E 的全部路径列举出来,计算每条路径的长度,从中选出最短路径。显然,穷举法过于费时,并且一旦中间点增加几个,会使备选的路径数量大大增加。

例 1.2　设备更新问题

某工厂使用一台设备,在每年年初,领导部门就要决定是购置新设备还是继续使用旧设备。如果继续使用旧设备,就要支付一定的维修费;如果购置新设备,则要支付一定的购置费。已知该设备 5 年内在各年年初的价格为

第 1 年	第 2 年	第 3 年	第 4 年	第 5 年
11	11	12	12	13

并且已知使用不同时间(年)的设备所需要的维修费用为

使用年限	0～1 年	1～2 年	2～3 年	3～4 年	4～5 年
维修费用	5	6	8	11	18

现在的问题是如何制订一个 5 年之内的设备更新计划，使得总的支付费用最少？

易见，这是一个 5 阶段决策过程。

例 1.3　排序问题

设有 n 个工件 $J_1, J_2, \cdots, J_n$ 要在一台机器上加工，工件 J_i 的加工时间记为 p_i，工件 J_i 的工期记为 d_i，求使最大延误时间最小的加工顺序。

显然，任意一个加工顺序都可以看作一个 n 阶段的决策，每个阶段需要决策选择哪个工件进行加工。

例 1.4　多元函数极值问题

求函数 $f(x_1, x_2, \cdots, x_n)$ 的最大值，这是一个与时间无关的静态问题，但我们可以把确定 $x_1, x_2, \cdots, x_n$ 的值的过程当作动态模型来处理，例如，先确定 x_1，再确定 x_2，以此类推。

1.2　动态规划的基本思想

动态规划的基本思想：把多阶段问题转化为一系列相互联系的单阶段问题，利用各阶段之间的关系，按照一定的次序依次求解，最后得到原问题的最优解。

我们以例 1.1 的最短路问题为例来说明动态规划的基本思想。

先把整个过程分为 4 个阶段，如图 1-3 所示。

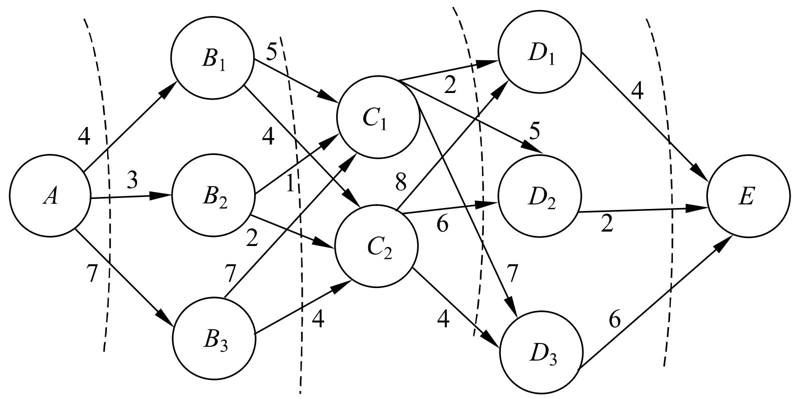

图 1-3

第 1 阶段以 A 为起点，B_1, B_2, B_3 为终点，有三种选择，如果我们选择 B_2 作为第 1 阶段的决策，则它既是第 1 阶段路径的终点，也是第 2 阶段路径的起点。第 2 阶段从 B_2 出发，有两个点 C_1, C_2 可供选择。若我们选择 C_2 作为第 2 阶段的决策，则 C_2 既是第 2 阶段的终点，也是第 3 阶段的起点。依次类推下去，各个阶段的决策不同，就会得到不同的路径。当一个阶段的起点确定了之

后，它就会直接影响后面各个阶段的路径选择，以及整条路径的长度。但是，后面阶段的决策不会受该阶段起点之前的各个阶段路径的影响。因此，我们可以从后向前考虑各个阶段的决策。

为描述方便，我们定义如下记号。

s_k：第 k 阶段路径的起点；

$w(x,y)$：弧(x,y)的权；

$f_k(s_k)$：顶点 s_k 到终点 E 的最短路的路径长度。

决策过程如下。

第 4 阶段：

无论第 3 阶段选择了哪一个点，在第 4 阶段都只有一种选择，且 D_1,D_2,D_3 到 E 的最短距离分别是 4，2，6，即：

当 $s_4=D_1$ 时，$f_4(D_1)=w(D_1,E)=4$；

当 $s_4=D_2$ 时，$f_4(D_2)=w(D_2,E)=2$；

当 $s_4=D_3$ 时，$f_4(D_3)=w(D_3,E)=6$。

第 3 阶段：

当 $s_3=C_1$ 时，$f_3(C_1)=\min\begin{Bmatrix}w(C_1,D_1)+f_4(D_1)\\w(C_1,D_2)+f_4(D_2)\\w(C_1,D_3)+f_4(D_3)\end{Bmatrix}=w(C_1,D_1)+f_4(D_1)=6$；

当 $s_3=C_2$ 时，$f_3(C_2)=\min\begin{Bmatrix}w(C_2,D_1)+f_4(D_1)\\w(C_2,D_2)+f_4(D_2)\\w(C_2,D_3)+f_4(D_3)\end{Bmatrix}=w(C_2,D_2)+f_4(D_2)=8$。

第 2 阶段：

当 $s_2=B_1$ 时，$f_2(B_1)=\min\begin{Bmatrix}w(B_1,C_1)+f_3(C_1)\\w(B_1,C_2)+f_3(C_2)\end{Bmatrix}=w(B_1,C_1)+f_3(C_1)=11$；

当 $s_2=B_2$ 时，$f_2(B_2)=\min\begin{Bmatrix}w(B_2,C_1)+f_3(C_1)\\w(B_2,C_2)+f_3(C_2)\end{Bmatrix}=w(B_2,C_1)+f_3(C_1)=7$；

当 $s_2=B_3$ 时，$f_2(B_3)=\min\begin{Bmatrix}w(B_3,C_1)+f_3(C_1)\\w(B_3,C_2)+f_3(C_2)\end{Bmatrix}=w(B_3,C_2)+f_3(C_2)=12$。

第 1 阶段：

$$f_1(A)=\min\begin{Bmatrix}w(A,B_1)+f_2(B_1)\\w(A,B_2)+f_2(B_2)\\w(A,B_3)+f_2(B_3)\end{Bmatrix}=w(A,B_2)+f_2(B_2)=10。$$

采用反向追踪的方法，可以得到点 A 到点 E 的最短路径应该是 A-B_2-C_1-D_1-E，最短距离是 10。

这个例子的推进过程是从最后一个阶段到第一个阶段的，这属于动态规划的逆向解法。有时分析问题的过程也可以从第一个阶段到最后一个阶段，称之为顺向解法。像这种逐层推进的方法，其关键在于建立相邻两个阶段之间的递推方程，也就是动态规划方法的基本方程。

最后，我们通过例 1.1 的分析，总结一下用动态规划方法进行多阶段决策过程的特点：

(1) 整个过程可以自然地或者人为地分为 n 个阶段；

(2) 逐阶段进行决策，可以采用逆向递推，也可以采用顺向递推；

(3) 在每一阶段，都不是仅仅考虑当前阶段的指标是否达到最优，而是要同时考虑本阶段之前的各个阶段的总体指标，根据全局的最优性来确定本阶段的决策；

(4) 在逆向递推进行到第一阶段，或者顺向递推进行到最后一个阶段，需要通过反向追踪回溯到最后一个阶段或第一个阶段，才能完成整体的最优决策。

1.3　动态规划基础知识

本节将详细介绍动态规划的基础知识。

1.3.1　基本概念和常用术语

1. 阶段

一个问题用动态规划方法来求解，首先应该适当地把问题的过程划分成若干个相互联系的**阶段**，以便按一定的次序求解。描述阶段的变量称为阶段变量，常用 k 表示。从过程开始到结束的阶段数目 N 称为历程。通常按照时间的自然特征来划分阶段，如例 1.1 就可以分成 4 个阶段来处理，即 $N=4$，$k=1,2,3,4$。对于不显露出时间特性的静态模型，也可以人为地划分一些阶段，作为一个过程来处理。一般来说，历程 N 可以是确定的，也可以是不确定的。根据历程，可以将多阶段决策过程分为：

(1) 定期多阶段决策过程：在决策之前就已知历程是确定的有限值；

(2) 不定期多阶段决策过程：预先知道历程是确定的有限值，但在得到最优解之前不知道它的具体值；

(3) 随机多阶段决策过程：历程是与外部条件有关的随机变量；

(4) 无限期多阶段决策过程：历程无限。

2. 状态

状态表示每个阶段开始时的客观条件，阶段的状态可以用阶段的某种特征来描述，决策的过程可以通过各个阶段状态的演变来说明。一个阶段通常有若干状态，阶段的一个状态也可以认为状态的某个“值”。描述状态的变量称为状态变量，第 k 阶段状态的状态变量记为 s_k。在例 1.1 中，某一阶段的状态就是该阶段支路的起点，它同时也是上一阶段某支路的终点。如在第 3 阶段，有两个状态，即点集 $\{C_1, C_2\}$，所以状态变量 s_3 可以取两个值，$s_3=C_1$ 或者 $s_3=C_2$。

3. 决策

所谓**决策**，就是在给定某阶段状态后，从该状态演变到下一个阶段状态所做的抉择。描述决策的变量称为决策变量。第 k 阶段的决策变量 x_k 的取值集与第 k 阶段的状态 s_k 有关，记为 $D_k(s_k)$。在例 1.1 中，第 3 阶段的状态变量 $s_3=C_1$ 时，决策变量 x_3 的取值集 $D_3(C_1)=\{D_1, D_2, D_3\}$，若选择 D_1，则 $x_3(C_1)=D_1$。

4. 策略

假定问题分为 N 个阶段，那么从第一阶段开始到第 N 阶段结束的过程中，决策变量序列 $(x_1, x_2, \cdots, x_N)$ 的一组值称为一个全过程策略，简称**策略**，记为 P。如在例 1.1 中，$x_4(E)=D_1$，$x_3(D_1)=C_1$，$x_2(C_1)=B_2$，$x_1(B_2)=A$ 构成一个策略 $A\text{-}B_2\text{-}C_1\text{-}D_1\text{-}E$。对逆向解法来说，决策变量序列 $(x_k, x_{k+1}, \cdots, x_N)$，$1 \leqslant k \leqslant N$ 的一组值称为一个子策略，记为 P_k。$C_1\text{-}D_1\text{-}E$ 就是例 1.1 的一个子策略；对顺向解法来说，决策变量序列 $(x_1, x_2, \cdots, x_k)$，$1 \leqslant k \leqslant N$ 的一组值称为一个子策略。子策略 $(x_k, x_{k+1}, \cdots, x_N)$ 或者 $(x_1, x_2, \cdots, x_k)$ 的取值集记为 $\widetilde{D}(s_k)$。

用动态规划方法解最优化问题，就是选择一种策略去控制一个多阶段过程的发展，以达到最佳的运行效果，此时称这个策略为最优策略。

5. 状态转移规律

对逆向解法来讲，第 k 阶段状态 s_k 是由后一阶段的状态 s_{k+1} 取决策变量

的某一个值 x_{k+1} 演变而来的。一般来讲，s_k 是 s_{k+1} 和 x_{k+1} 的函数 $s_k=T_k(s_{k+1},x_{k+1})$。类似地，对顺向解法来讲，第 k 阶段状态 s_k 是由前一阶段的状态 s_{k-1} 取决策变量的某一个值 x_{k-1} 演变而来的。**状态转移规律**就是指 $s_k=T_k(s_{k+1},x_{k+1})$或者 $s_k=T_k(s_{k-1},x_{k-1})$。

动态规划的决策过程中，状态具有无后效性，也称马尔可夫性。即状态应具有这样的性质：如果某阶段状态给定后，则在这阶段以后的过程发展不受这阶段以前各段状态的影响。也就是说，过程的过去历史只能通过当前的状态去影响它未来的发展，当前的状态是以往历史的一个总结。换而言之，一般的状态转移规律为 $s_k=T_k(s_{k+1},x_{k+1})$或者 $s_k=T_k(s_{k-1},x_{k-1})$的形式，不依赖于 $s_N,x_N,s_{N-1},x_{N-1},\cdots,s_{k+2},x_{k+2}$，或 $s_1,x_1,s_2,x_2,\cdots,s_{k-2},x_{k-2}$。在例 1.1 中，状态转移规律为 $s_k=x_{k+1}(s_{k+1})$。

6. 权函数

对第 k 阶段状态 s_k，当决策变量 x_k 取某个值(或方案)后，就有一个反映这个局部措施的效益指标 $w_k(s_k,x_k)$，称为**权函数**。在例 1.1 中，$w_k(s_k,x_k)=d(s_k,x_k)$，其中 $d(s_k,x_k)$表示 s_k 与 x_k 之间的距离，例如 $w_3(C_2,D_2)=6$。

7. 指标函数

指标函数是用来衡量决策过程效果优劣的一种数量指标，是关于状态和策略的数量函数。具体是指在某个阶段的某个状态出发，采取某个子策略时所产生的效益，反映从该阶段到决策过程最初阶段的各阶段效益总和(这里的总和是一个广义的概念，通常是实数的加法或乘法运算)。动态规划模型中的指标函数要具有可分离性，即过程的每个部分(子过程)都可以计算效益。对逆向解法来讲，在第 k 阶段状态 s_k 采取子策略$(x_k,x_{k+1},\cdots,x_N)$时的指标函数是指从阶段 k 到阶段 N 可获得的效益；对顺向解法来讲，在第 k 阶段状态 s_k 采取子策略$(x_1,x_2,\cdots,x_k)$时的指标函数是指从阶段 1 到阶段 k 可获得的效益。

指标函数的最优值称为最优指标函数，对于逆向解法来讲，它表示在第 k 阶段状态 s_k 采取最优子策略$(x_k,x_{k+1},\cdots,x_N)$，从阶段 k 到阶段 N 可获得的效益，记为 $F_k(s_k)$。通常 $F_k(s_k)$可写成下列形式

$$F_k(s_k)$$
$$=\underset{(x_k,x_{k+1},\cdots,x_N)\in\widetilde{D}(s_k)}{\mathrm{opt}}\{w_k(s_k,x_k)\cdot w_{k+1}(s_{k+1},x_{k+1})\cdot\cdots\cdot w_N(s_N,x_N)\};$$

对顺向解法来讲，它表示在第 k 阶段状态 s_k 采取最优子策略$(x_1,x_2,\cdots,x_k)$，从阶段 1 到阶段 k 可获得的效益，也记为 $F_k(s_k)$。通常 $F_k(s_k)$可写成下列形式：

$$F_k(s_k)=\underset{(x_1,x_2,\cdots,x_k)\in\widetilde{D}(s_k)}{\text{opt}}\{w_1(s_1,x_1)\cdot w_2(s_2,x_2)\cdot\cdots\cdot w_k(s_k,x_k)\}。$$

对于逆向解法来讲，$F_1(s_1)$代表全局最优值；对顺向解法来讲，$F_N(s_N)$代表全局最优值。如在例1.1中，$F_2(B_2)=7$ 表示从点 B_2 到点 E 的最短路径长度为7，$F_1(A)=10$ 表示从起点 A 到终点 E 的最短路径长度为10。

8. 递推方程

对逆向解法来讲，递推方程是

$$\begin{cases}F_{N+1}(s_{N+1})=0\text{ 或 }1,\\F_k(s_k)=\underset{x_k\in D_k(s_k)}{\text{opt}}\{w_k(s_k,x_k)\cdot F_{k+1}(s_{k+1})\},\quad k=N,N-1,\cdots,1。\end{cases}$$

在上述方程中，$F_{N+1}(s_{N+1})=0$ 或1为初始条件(也称边界条件)，当·为加法时，取 $F_{N+1}(s_{N+1})=0$；当·为乘法时，取 $F_{N+1}(s_{N+1})=1$。在例1.1中，递推方程为

$$\begin{cases}F_6(s_6)=0,\\F_k(s_k)=\min\limits_{x_k\in D_k(s_k)}\{w_k(s_k,x_k)+F_{k+1}(s_{k+1})\},\quad k=5,4,3,2,1。\end{cases}$$

对顺向解法来讲，递推方程是

$$\begin{cases}F_0(s_0)=0\text{ 或 }1,\\F_k(s_k)=\underset{x_k\in D_k(s_k)}{\text{opt}}\{w_k(s_k,x_k)\cdot F_{k-1}(s_{k-1})\},\quad k=1,2,\cdots,N。\end{cases}$$

其中 $F_0(s_0)=0$ 或1为初始条件，当·为加法时，取 $F_0(s_0)=0$；当·为乘法时，取 $F_0(s_0)=1$。

1.3.2 动态规划基本模型及基本原理

1. 动态规划基本模型

用动态规划方法解决多阶段决策过程的问题所需建立的模型称为动态规划模型。一般动态规划模型包括以下几个组成部分：

(1) 时间参量集 由于实际的决策过程是随时间而变化的，所以时间参量是模型的一个组成部分。如果决策是在离散的时间上采取的，则时间参量是离散的，相应的决策过程是离散过程；如果决策是在连续的时间上采取的，则时间参量是连续的，相应的决策过程是连续过程。

(2) 状态空间 在决策过程中，状态起着描述过程的作用，各个时刻的状态一旦确定，整个过程便随之确定。当决策的方式给定时，状态随时间的变化规

律可能是确定性的,也可能是随机性的,相应的决策过程称为确定性决策过程或随机性决策过程。

根据时间参量和状态空间的特性,动态规划模型可以分为离散确定型、离散随机型、连续确定型和连续随机型四类。

(3) 决策空间　在决策过程中,决策是影响或控制过程发展的外加因素,用决策变量来描述,决策变量的取值集合称为决策空间。

(4) 状态转移规律　状态转移规律描述了本阶段状态与上一阶段状态及上一阶段决策之间的关系,通常用状态转移方程刻画由一个阶段的状态到下一阶段的状态的演变规律。

(5) 权函数　权函数体现从一个阶段到下一阶段的阶段效应。

(6) 指标函数　指标函数体现从一个阶段到最后阶段的总效应。

下面,我们总结一下建立动态规划模型的基本步骤:

① 将过程进行恰当的分段,一般可以根据时间和空间划分;

② 正确选择状态变量 s_k,使它既能描述过程,又能满足无后效性;

③ 确定决策变量 x_k,及每个阶段的允许决策集合 $D_k(s_k)$;

④ 写出状态转移方程:$s_k=T_k(s_{k+1},x_{k+1})$或者 $s_k=T_k(s_{k-1},x_{k-1})$;

⑤ 根据题意写出最优指标函数:

$$F_k(s_k)=\underset{(x_k,x_{k+1},\cdots,x_N)\in\widetilde{D}(s_k)}{\mathrm{opt}}\{w_k(s_k,x_k)\cdot w_{k+1}(s_{k+1},x_{k+1})\cdot\cdots\cdot w_N(s_N,x_N)\},$$

或

$$F_k(s_k)=\underset{(x_1,x_2,\cdots,x_k)\in\widetilde{D}(s_k)}{\mathrm{opt}}\{w_1(s_1,x_1)\cdot w_2(s_2,x_2)\cdot\cdots\cdot w_k(s_k,x_k)\};$$

⑥ 根据问题的性质写出递推方程,也就是动态规划的基本方程:

$$\begin{cases}F_{N+1}(s_{N+1})=0\text{ 或 }1,\\ F_k(s_k)=\underset{x_k\in D_k(s_k)}{\mathrm{opt}}\{w_k(s_k,x_k)\cdot F_{k+1}(s_{k+1})\},\quad k=N,N-1,\cdots,1,\end{cases}$$

或

$$\begin{cases}F_0(s_0)=0\text{ 或 }1,\\ F_k(s_k)=\underset{x_k\in D_k(s_k)}{\mathrm{opt}}\{w_k(s_k,x_k)\cdot F_{k-1}(s_{k-1})\},\quad k=1,2,\cdots,N。\end{cases}$$

在计算时,根据边界条件从 $k=N$(或 $k=1$)开始由后向前(或由前向后)逐步递推,求得各阶段的最优决策和最优指标函数,最后算出 $F_1(s_1)$(或 $F_N(s_N)$)就是问题的最优值,最优决策需要按照求解步骤反向追踪,就可以得到全局最优策略。

用动态规划方法解决问题有两个特点。首先,只有当所有的阶段和状态都被考察过,从最后一个阶段出发,用追踪法往回考察所有的阶段后,才能得到最优解。此外,一般来讲,动态规划方法是指数算法。

2. 动态规划方法的基本原理

1951 年,美国数学家 R. Bellman 等人在建立动态规划这一数学分支时,根据一类多阶段决策问题的特点,提出了解决这类问题的“最优化原理”,它是动态规划的理论基础,能够解决许多类型决策过程的优化问题。

最优化原理的提出基于这样的原因:多阶段决策过程的特点是每个阶段都要进行决策,具有 N 个阶段的决策过程的策略是由 N 个相继进行的阶段决策构成的决策序列。由于前阶段的终止状态又是后一阶段的初始状态,因此确定阶段最优决策不能只从本阶段的效应出发,必须通盘考虑,整体规划。就是说,阶段 k 的最优决策不应只是本阶段的最优,而必须是本阶段及其所有后续阶段的总体最优,即关于整个后部子过程的最优决策。**最优化原理**的实质是在多阶段决策过程中,无论过去的过程如何,只从当前的状态和系统的最优化要求出发,找出下一步的最优决策。即最优策略的任何一部分子策略必然是最优子策略。

1.3.3 可用动态规划求解的问题的特征

动态规划是对于某一类问题的解决方法,重点在于如何鉴定“某一类问题”是动态规划可解的。

一个问题是否能够用动态规划算法去求解,取决于待求解问题本身是否具有以下两个重要性质:最优子结构性质和状态的无后效性。

1. 最优子结构性质

如果问题的最优解所包含的子问题的解也是最优的,我们就称该问题具有最优子结构性质(即满足最优化原理)。最优子结构性质为动态规划算法解决问题提供了重要线索。设计动态规划算法的核心在于什么是问题的最优子结构,如何找到这个最优子结构。如果某些递推方程不能保证最优化原理,就不能用动态规划方法求解。

2. 状态的无后效性

应用动态规划方法解决多阶段决策问题是通过拆分问题,定义问题状态和状态之间的关系,使得问题能够以递推的方式去解决。其中最关键的思想就是

将全局问题分解成子问题,一步步地找出每一步问题的最优解,最后得出全局问题的最优解。因此,动态规划的本质,是对问题状态的定义和状态转移方程的定义。这两点是动态规划模型中最关键的两个要素。其中状态是一种静态的量,在需要的时候去获取它的值;而状态转移方程则是对状态进行提取和操作,从而更新或得出我们需要的新的状态。

是否能够用动态规划来解决问题,除了要满足具有最优子结构性质,还有一个必备条件,就是各阶段的状态具有无后效性,即对于逆向解法,第 k 阶段的状态只与第 $k+1$ 阶段的状态有关,对于顺向解法,第 k 阶段的状态只与第 $k-1$ 阶段的状态有关,而与其他各阶段的状态无关。

上述两个必备条件可用来检查问题是否能用动态规划算法求解,而动态规划方法的有效性与子问题的重叠性有关。

3. 子问题的重叠性

子问题的重叠性是指在对问题进行求解时,每次产生的子问题并不总是新问题,有些子问题会被重复计算多次。动态规划算法正是利用了这种子问题的重叠性质,对每一个子问题只计算一次,然后将其计算结果保存起来,当需要再次计算已经计算过的子问题时,只需要简单地查看一下结果,从而获得较高的解题效率。

子问题的重叠性能省去很多重复的步骤,从而使动态规划算法与其他穷举思想的算法相比,具有较高的效率。

1.4 动态规划在组合优化问题中的应用

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。在第 1 章,我们以最短路问题为例介绍了动态规划的基本思想,本章我们继续介绍动态规划的其他重要应用。通过动态规划在各类问题中的实际应用,我们也可以更好地体会到动态规划是求解多阶段决策问题的一种途径,一种思想方法,而不是一个具体的算法。动态规划方法通常用于求解具有某种最优性质的问题。因为各种问题的性质不同,最优解满足的条件也各不相同,所以动态规划算法没有一个标准的表达式,它对不同问题有不同的数学表达式,因而没有统一的处理格式。它必须依据问题本身的特性,利用灵活的数学技巧来处理。

1.4.1 资源分配问题

资源分配问题是指将一定数量的一种或多种资源分配给使用者,以获取最

大的收益的一类问题。这类问题是动态规划方法的典型应用。

资源分配问题可以这样描述：某工厂用某种原料生产产品，现有原料总量为 w，用于生产 n 种产品。假设分配给生产第 i 种产品的原料为 x_i，获得的收益是 $g_i(x_i)$，$i=1,2,\cdots,n$。问如何分配资源可以使得该工厂收益最大？

如果收益函数 $g_i(x_i)$是线性的，该问题可以建立线性规划模型。如果 $g_i(x_i)$是非线性的，当 n 较大时，可以用动态规划的方法建立模型并求出最优解。这是一个与时间无关的问题，可以将分配资源的过程分成 n 个阶段，采用逆向解法。下面建立动态规划模型：

首先，将分配资源的过程分成 n 个阶段：第 i 阶段为第 i 种产品分配原料，$i=1,2,\cdots,n$；

然后，令状态变量 s_i 表示分配给用于生产第 i 种产品到第 n 种产品的原料数量；

决策变量 x_i 表示分配给生产第 i 种产品的原料数量；

状态转移方程：$s_{i+1}=s_i-x_i$；

最优值函数 $f_i(s_i)$表示在第 i 阶段当所选择物品的总量不超过 s_i 的情况下，能够达到的最大总价值；

边界条件：$f_{n+1}(s_{n+1})=0, 0\leqslant s_{n+1}\leqslant w$；

递推方程：$f_i(s_i)=\max\limits_{0\leqslant x_i\leqslant s_i}\{g_i(x_i)+f_{i+1}(s_i-x_i)\},\quad i=n-1,n-2,\cdots,1$；

最优值为 $f_1(w)$。

按上面的递推方程逐阶段计算，$f_1(w)$的值就是可获得的最大收益，再反向追踪，可得到最优决策，即用于生产各种产品的原料数量。

下面我们用一个具体的例子来运行这个动态规划算法。

例 1.5 某公司拟将五份资金分配给 A、B、C 三个项目，各项目可获得的利润如表 1-1 所示。问如何分配这五份资金，公司收益最大？

表 1.1 单位：万元

资金份数 / 收益 / 项目	0	1	2	3	4	5
A	0	4	6	7	7	7
B	0	2	4	6	8	9
C	0	4	5	6	6	6

解：

将问题按项目分成 3 个阶段，即 $N=3$。设分配给第 i 个到第 N 个项目的

资金份数共为 s_i，则 s_i 可能的取值有六种，即 $s_i=0,1,2,3,4,5(i=1,2,3)$；

边界条件：$f_4(s_4)=0, s_4=0,1,2,3,4,5$。

第 3 阶段：$f_3(0)=g_3(0)+f_4(0)=0$，

$f_3(1)=\max\{g_3(0)+f_4(1),\underline{g_3(1)+f_4(0)}\}=\max\{0+0,\underline{0+4}\}=4$，

类似地，计算出 $f_3(2)=5, f_3(3)=f_3(4)=f_3(5)=6$。

第 2 阶段：$f_2(0)=g_2(0)+f_3(0)=0$，

$$\begin{aligned}f_2(1)&=\max\{\underline{g_2(0)+f_3(1)},g_2(1)+f_3(0)\}\\&=\max\{\underline{0+4},2+0\}=4,\\f_2(2)&=\max\{g_2(0)+f_3(2),\underline{g_2(1)+f_3(1)},g_2(2)+f_3(0)\}\\&=\max\{0+5,\underline{2+4},4+0\}=6,\\f_2(3)&=\max\{g_2(0)+f_3(3),g_2(1)+f_3(2),\underline{g_2(2)+f_3(1)},\\&\qquad g_2(3)+f_3(0)\}\\&=\max\{0+6,2+5,\underline{4+4},6+0\}=8,\\f_2(4)&=\max\{g_2(0)+f_3(4),g_2(1)+f_3(3),g_2(2)+f_3(2),\\&\qquad \underline{g_2(3)+f_3(1)},g_2(4)+f_3(0)\}\\&=\max\{0+6,2+6,4+5,\underline{6+4},8+0\}=10。\end{aligned}$$

类似地，计算出 $f_2(5)=12$。

第 1 阶段：$f_1(0)=0, f_1(1)=4, f_1(2)=8, f_1(3)=10, f_1(4)=12$，

$$\begin{aligned}f_1(5)&=\max\{g_1(0)+f_2(5),\underline{g_1(1)+f_2(4)},\underline{g_1(2)+f_2(3)},g_1(3)+f_2(2),\\&\qquad g_1(4)+f_2(1),g_1(5)+f_2(0)\}\\&=\max\{0+12,\underline{4+10},\underline{6+8},7+6,7+4,7+0\}=14。\end{aligned}$$

因此，最大收益为 14。通过反向追踪可以得出最优资金分配方案：分配 1 份资金给项目 A，3 份资金给项目 B，1 份资金给项目 C；或者分配 2 份资金给项目 A，2 份资金给项目 B，1 份资金给项目 C。

1.4.2　背包问题

背包问题是经典的组合优化问题，也可以用动态规划的方法求解。

背包问题描述：设 n 是非负整数，给定一组物品共 n 个，每个物品有其自己的重量 $W_1,W_2,\cdots,W_n$ 和价值 $V_1,V_2,\cdots,V_n$，在限定的总重量 c 内，选择若干物品，使得所选物品的总价值最高。即找到一个子集 $S\subseteq\{1,2,\cdots,n\}$，使得在 $\sum\limits_{j\in S}W_j\leqslant c$ 的条件下，$\sum\limits_{j\in S}V_j$ 的值最大。

首先介绍一种逆向解法。

背包问题的动态规划算法 1：

按可选择物品的数量划分阶段，因此共有 n 个阶段；

状态变量 w 表示在第 k 阶段，考虑了第 k 个到第 n 个物品时，可能达到的总重量；

指标函数 $f_k(w)$ 表示当所选择物品的总重量不超过 w 的情况下，能够达到的最大总价值；

递推方程：

$$\begin{cases} f_{n+1}(w)=\begin{cases}0, & 0\leqslant w\leqslant c, \\ -\infty, & w<0,\end{cases} \\ f_k(w)=\max\{0+f_{k+1}(w), V_k+f_{k+1}(w-W_k)\}, \quad k=2,3,\cdots,n。\end{cases}$$

在上述递推方程中，$f_{n+1}(w)=\begin{cases}0, & 0\leqslant w\leqslant c, \\ -\infty, & w<0\end{cases}$ 是边界条件。$f_k(w)$的表达式中，第一个式子表示在第 k 阶段，考虑了第 k 个物品后，没有选择该物品，此时所选物品的总重量是 w，因此所选全部物品的价值和第$(k+1)$阶段考虑到第$(k+1)$个物品时的总价值一样。第二个式子表示在第 k 阶段，考虑了第 k 个物品后，选择了该物品，此时所选物品的总重量是 w，因此所选物品的总价值等于第$(k+1)$阶段总重量是$(w-W_k)$时的指标函数值$\max\limits_{w\leqslant c}\{f_1(w)\}$加上第 k 个物品的价值 V_k。

最优值函数为$\max\limits_{w\leqslant c}\{f_1(w)\}$。

下面我们通过具体实例来演示该算法的执行过程。

例 1.6　现有 3 个物品，重量分别为 5，4，2，价值分别为 25，18，8，背包能够承载总重量为 6，问：如何选择物品才能使装进背包的总价值最大？

解：首先，将问题分为 3 个阶段，边界条件 $f_4(w)=\begin{cases}0, & 0\leqslant w\leqslant 6, \\ -\infty, & w<0,\end{cases}$

然后按前述递推方程计算：

$$f_3(w)=\max\{f_4(w), 8+f_4(w-2)\}。$$

因此，$f_3(0)=f_3(1)=0$，$f_3(w)=8$(若 $2\leqslant w\leqslant 6$)。这意味着当 $2\leqslant w\leqslant 6$ 时，选择第 3 个物品最好。

接着由递推方程

$$f_2(w)=\max(f_3(w), 18+f_3(w-4))$$

可求得

$$f_2(0)=f_2(1)=0, f_2(2)=f_2(3)=8, f_2(4)=f_2(5)=18,$$

$$f_2(6)=\max(f_3(6), 18+f_3(2))=\max(8, 18+8)=26。$$

接着由递推方程

$$f_1(w)=\max(f_2(w), 25+f_2(w-5))$$

可求得

$$f_1(0)=f_1(1)=0, f_1(2)=f_1(3)=8, f_1(4)=18, f_1(5)=25,$$

$$f_1(6)=\max(f_2(6),25+f_2(1))=\max(26,25+0)=26。$$

因此，最优值为$\max\limits_{w\leqslant 6}\{f_1(w)\}=f_1(6)=26$，利用反向追踪得到最优解为选择重量为 4 和 2 的物品。

建立动态规划算法时不能墨守成规，可以运用各种技巧来简化计算。我们还可以按下面的方法得到背包问题的动态规划顺向解法。

背包问题的动态规划算法 2：

令 $V(i,j)$表示在前 $i(1\leqslant i\leqslant n)$个物品中能够装入容量为 $j(1\leqslant j\leqslant c)$的背包中的物品的最大值，则可以得到如下动态规划函数：

边界条件：$V(i,0)=V(0,j)=0, 0\leqslant i\leqslant n, 0\leqslant j\leqslant c$；

递推方程：

$$V(i,j)=\begin{cases}V(i-1,j), & j<V_i,\\ \max\{V(i-1,j),V(i-1,j-w_i)+V_i\}, & j\geqslant V_i。\end{cases}$$

边界条件表示把前面 i 个物品装入容量为 0 的背包和把 0 个物品装入容量为 j 的背包，得到的价值均为 0。

递推方程中，第一个式子表示如果第 i 个物品的重量大于背包的容量，则物品 i 不能装入背包，因此装入前 i 个物品得到的最大价值和装入前$(i-1)$个物品得到的最大价值是相同的。第二个式子表示如果第 i 个物品的重量不超过背包的容量，则会有以下两种情况：(1)如果第 i 个物品没有装入背包，则背包中物品的价值就等于把前$(i-1)$个物品装入容量为 j 的背包中所取得的价值；(2)如果把第 i 个物品装入背包，则背包中物品的价值等于把前$(i-1)$个物品装入容量为$(j-w_i)$的背包中的价值加上第 i 个物品的价值 V_i。显然，取二者中价值较大者作为把前 i 个物品装入容量为 j 的背包中的最优解。

最优值为 $V(n,c)$。

接下来我们利用背包问题的顺向动态规划算法来求例 1.7 的最优解。

例 1.7　有 5 个物品，其重量分别是{2,2,6,5,4}，价值分别为{6,3,5,4,6}，背包的容量为 10，求装入背包的物品和获得的最大价值。

解：置初始值 $V(i,0)=V(0,j)=0, i=0,1,\cdots,5; j=0,1,\cdots,10$。

第 1 阶段：$V(1,1)=V(0,1)=0$,

$V(1,j)=\max\{V(0,j),\underline{V(0,j-2)+6}\}=6, j=2,\cdots,10$。

第 2 阶段：$V(2,1)=V(1,1)=0$,

$V(2,j)=\max\{\underline{V(1,j)},V(1,j-2)+3\}=\max\{\underline{6},0+3\}=6$,

$j=2,3$,

$$V(2,j)=\max\{V(1,j),\underline{V(1,j-2)+3}\}=\max\{6,\underline{6+3}\}=9,\quad j=4,5,\cdots,10。$$

第 3 阶段：$V(3,1)=V(2,1)=0,$

$V(3,j)=V(2,j)=6, j=2,3,$

$V(3,j)=V(2,j)=9, j=4,5,$

$$V(3,j)=\max\{\underline{V(2,j)},V(2,j-6)+5\}=\max\{\underline{9},0+5\}=9,\quad j=6,7,$$

$$V(3,j)=\max\{V(2,j),\underline{V(2,j-6)+5}\}=\max\{9,\underline{6+5}\}=11,\quad j=8,9,$$

$$V(3,10)=\max\{V(2,10),\underline{V(2,4)+5}\}=\max\{9,\underline{9+5}\}=14。$$

第 4 阶段：$V(4,1)=V(3,1)=0,$

$V(4,j)=V(3,j)=6, j=2,3,$

$V(4,4)=V(3,4)=9,$

$$V(4,j)=\max\{\underline{V(3,j)},V(3,j-5)+4\}=\max\{\underline{9},0+4\}=9,\quad j=5,6,$$

$$V(4,7)=\max\{V(3,7),\underline{V(3,2)+4}\}=\max\{9,\underline{6+4}\}=10,$$

$$V(4,8)=\max\{V(3,8),\underline{V(3,3)+4}\}=\max\{11,\underline{6+4}\}=11,$$

$$V(4,9)=\max\{V(3,9),\underline{V(3,4)+4}\}=\max\{11,\underline{9+4}\}=13,$$

$$V(4,10)=\max\{\underline{V(3,10)},V(3,5)+4\}=\max\{\underline{14},9+4\}=14。$$

第 5 阶段：$V(5,1)=V(4,1)=0,$

$V(5,j)=V(4,j)=6, j=2,3,$

$$V(5,j)=\max\{\underline{V(4,j)},V(4,j-4)+6\}=\max\{\underline{9},0+6\}=9,\quad j=4,5,$$

$$V(5,6)=\max\{V(4,6),\underline{V(4,2)+6}\}=\max\{9,\underline{6+6}\}=12,$$

$$V(5,7)=\max\{V(4,7),\underline{V(4,3)+6}\}=\max\{10,\underline{6+6}\}=12,$$

$$V(5,8)=\max\{V(4,8),\underline{V(4,4)+6}\}=\max\{11,\underline{9+6}\}=15,$$

$$V(5,9)=\max\{V(4,9),\underline{V(4,5)+6}\}=\max\{13,\underline{9+6}\}=15,$$

$$V(5,10)=\max\{V(4,10),\underline{V(4,6)+6}\}=\max\{14,\underline{9+6}\}=15。$$

因此，装入背包内物品的最大价值是 15，用反向追踪法回溯，找到问题的最优解，即装入第 5 个、第 2 个和第 1 个物品。

1.4.3 设备更新问题

企业中经常会遇到一台设备应该使用多久之后再更新最划算的问题。一般来说，一台设备在比较新时，经济效益高，故障少，维修费用少，但随着使用年限的增加，经济效益减少，故障变多，维修费用随之增加。如果更新设备，可提

高年净收入，但是当年要支出一笔数额较大的更新费用。

设备更新问题的一般描述为：在已知一台设备的效益函数 $r(t)$，维修费用函数 $u(t)$，及更新费用函数 $c(t)$ 的条件下，要求在 n 年内的每年年初做出决策，是继续使用旧设备还是更换一台新的，使得 n 年总效益最大。

用逆推方法求解，建立动态规划模型：

按照机器的使用年限来划分阶段，$k=1,2,\cdots,n$；

决策变量 $x_k=\begin{cases}0, & \text{第 } k \text{ 年继续使用旧设备,}\\ 1, & \text{第 } k \text{ 年使用新设备;}\end{cases}$

状态变量 s_k：第 k 年年初，设备已经使用过的年限；

状态转移方程：$s_{k+1}=s_k(1-x_k)+1$，即表示如果第 k 年年初，使用新设备，则 $s_{k+1}=1$，否则 $s_{k+1}=s_k+1$；

最优指标函数 $f_k(s_k)$表示在第 k 年年初，使用一台已经用了 s_k 年的设备，到第 n 年年末的最大收益，则递推方程：

$$f_k(s_k)=\max\begin{cases}r(s_k)-u(s_k)+f_{k+1}(s_{k+1}), & x_k=0,\\ r(0)-u(0)-c(s_k)+f_{k+1}(1), & x_k=1,\end{cases}\quad k=n,n-1,\cdots,1;$$

递推方程中的第一个式子表示在第 k 年年初继续使用旧设备，没有购进新设备，第二个式子表示在第 k 年年初购进新设备。

边界条件：$f_{n+1}(s_{n+1})=0, s_{n+1}=1,2,\cdots,n$；

最优值：$f_1(0)$。

例 1.8　某企业使用一台设备，在每年年初，企业领导部门就要决定是购置新设备还是继续使用旧设备。如果购置新设备，就要支付一定的购置费用，如果继续使用旧设备，则需支付一定的维修保养费用。预测该设备在 5 年内每年年初的价格如表 1-2 所示，且已知使用不同年限的设备所需要的维修费用及能够带来的效益如表 1-3 所示，该企业领导该如何制定 5 年之内的设备更新计划，使得总的支付费用最少？

表 1-2　单位：万元

时间	第 1 年	第 2 年	第 3 年	第 4 年	第 5 年
价格	11	11	12	12	13

表 1-3　单位：万元

使用年限	0～1 年	1～2 年	2～3 年	3～4 年	4～5 年
维修费用	5	6	8	11	18
效益	30	26	20	14	10

解：用逆推方法求解。按照机器的使用年限来划分阶段，$k=1,2,3,4,5$。

$k=6, f_6(s_6)=0, s_6=1,2,3,4,5,$

$$k=5, f_5(1)=\max\left\{\begin{matrix}\underline{r(1)-u(1)+f_6(2)},\\ r(0)-u(0)-c(5)+f_6(1)\end{matrix}\right\}=\max\left\{\begin{matrix}\underline{26-6+0},\\ 30-5-13+0\end{matrix}\right\}=20,$$

$$f_5(2)=\max\left\{\begin{matrix}r(2)-u(2)+f_6(3),\\ \underline{r(0)-u(0)-c(5)+f_6(1)}\end{matrix}\right\}=\max\left\{\begin{matrix}20-8+0,\\ \underline{30-5-13+0}\end{matrix}\right\}=12,$$

$$f_5(3)=\max\left\{\begin{matrix}r(3)-u(3)+f_6(4),\\ \underline{r(0)-u(0)-c(5)+f_6(1)}\end{matrix}\right\}=\max\left\{\begin{matrix}14-11+0,\\ \underline{30-5-13+0}\end{matrix}\right\}=12,$$

$$f_5(4)=\max\left\{\begin{matrix}r(4)-u(4)+f_6(5),\\ \underline{r(0)-u(0)-c(5)+f_6(1)}\end{matrix}\right\}=\max\left\{\begin{matrix}10-18+0,\\ \underline{30-5-13+0}\end{matrix}\right\}=12,$$

$$k=4, f_4(1)=\max\left\{\begin{matrix}r(1)-u(1)+f_5(2),\\ \underline{r(0)-u(0)-c(4)+f_5(1)}\end{matrix}\right\}=\max\left\{\begin{matrix}26-6+12,\\ 30-5-12+20\end{matrix}\right\}=33,$$

$$f_4(2)=\max\left\{\begin{matrix}r(2)-u(2)+f_5(3),\\ \underline{r(0)-u(0)-c(4)+f_5(1)}\end{matrix}\right\}=\max\left\{\begin{matrix}20-8+12,\\ 30-5-12+20\end{matrix}\right\}=33,$$

$$f_4(3)=\max\left\{\begin{matrix}r(3)-u(3)+f_5(4),\\ \underline{r(0)-u(0)-c(4)+f_5(1)}\end{matrix}\right\}=\max\left\{\begin{matrix}14-11+12,\\ 30-5-12+20\end{matrix}\right\}=33,$$

$$k=3, f_3(1)=\max\left\{\begin{matrix}\underline{r(1)-u(1)+f_4(2)},\\ r(0)-u(0)-c(3)+f_4(1)\end{matrix}\right\}=\max\left\{\begin{matrix}\underline{26-6+33},\\ 30-5-12+33\end{matrix}\right\}=53,$$

$$f_3(2)=\max\left\{\begin{matrix}r(2)-u(2)+f_4(3),\\ \underline{r(0)-u(0)-c(3)+f_4(1)}\end{matrix}\right\}=\max\left\{\begin{matrix}20-8+33,\\ \underline{30-5-12+33}\end{matrix}\right\}=46,$$

$$k=2, f_2(1)=\max\left\{\begin{matrix}r(1)-u(1)+f_3(2),\\ \underline{r(0)-u(0)-c(2)+f_3(1)}\end{matrix}\right\}=\max\left\{\begin{matrix}26-6+46,\\ \underline{30-5-11+53}\end{matrix}\right\}=67,$$

$k=1, f_1(0)=30-5-11+f_2(1)=81$。

因此，5 年内该企业可以获得的最大收益是 81 万元，用反向追踪法回溯，找到问题的最优解，即在第 1 年、第 2 年和第 4 年买入新设备，第 3 年和第 5 年继续使用旧设备。

第 2 章　排序问题基本理论

近代排序论的研究中，排序问题(scheduling problem)是从 Johnson[1] 研究的有关流水作业环境开始的。随后中国科学院应用数学研究所越民义研究员就注意到排序问题的重要性和理论上的难度。1960 年他编写了国内第一本排序论讲义。20 世纪 70 年代初越民义和韩继业一起研究同顺序流水作业(同序作业)排序问题，开创了中国研究排序论的先河[2]。1985 年中国科学院自动化研究所疏松桂等把 scheduling 译为“调度”[3]。2003 年唐国春等[4]提出“排序”与“调度”作为 scheduling 的中文译名都只是描述 scheduling 的一个侧面。中国台湾的学术界把 scheduling 翻译成“排程”。2010 在唐国春提出把 scheduling 翻译成“排序与调度”[5]。本研究按照文献[4]的译法。

1974 年 Baker 给出定义：“排序是按时分配资源去执行一组任务”[6]，即 scheduling 是为完成若干项任务(jobs)而对资源(指包括机器在内的各种资源)按时间进行分配。接着 Baker 指出：“排序是一个决策函数”。2018 年 Pinedo 提出几乎相同的定义：“排序处理的是在一段时间内将稀缺资源分配给任务的问题，是以优化一个或者多个目标函数为目标的决策过程[7]。”因而按时间分配“任务和资源”就是 scheduling 最本质的特征。

由于排序领域内许多早期的研究工作是在制造业推动下发展起来的，所以在描述排序问题时很自然会使用制造业的术语。尽管现在排序问题在许多非制造业已取得了很多相当有意义的成果，但是制造业的术语仍然在使用。因而往往把资源(resources)称为机器(machines)，把任务(tasks)称为工件(jobs)。有时工件可能是由几个先后次序约束相互联系着的基本任务(elementary tasks)所组成。这种基本任务称为工序(operations)。排序中的“机器”和“工件”已经不是机器制造业中的“机床”和“机床加工的零件”，而是“机床”和“零件”关系的抽象概念。排序中的机器可以是数控机床、计算机 CPU、医院的病床或者医生、消防设备和机场跑道等，工件可以是零件、计算机终端、病人、森林起火点和降落的飞机等。因此排序问题中，工件是被加工的对象，是要完成的任务；机器是提供加工的对象，是完成任务所需要的资源。排序是指在一定的约束条件下对工件和机器按时间进行分配和安排次序，使某一个或某一些目标达到最优，是安排时间表的简称，这里工件和机器可以代表极其广泛的实际对象。