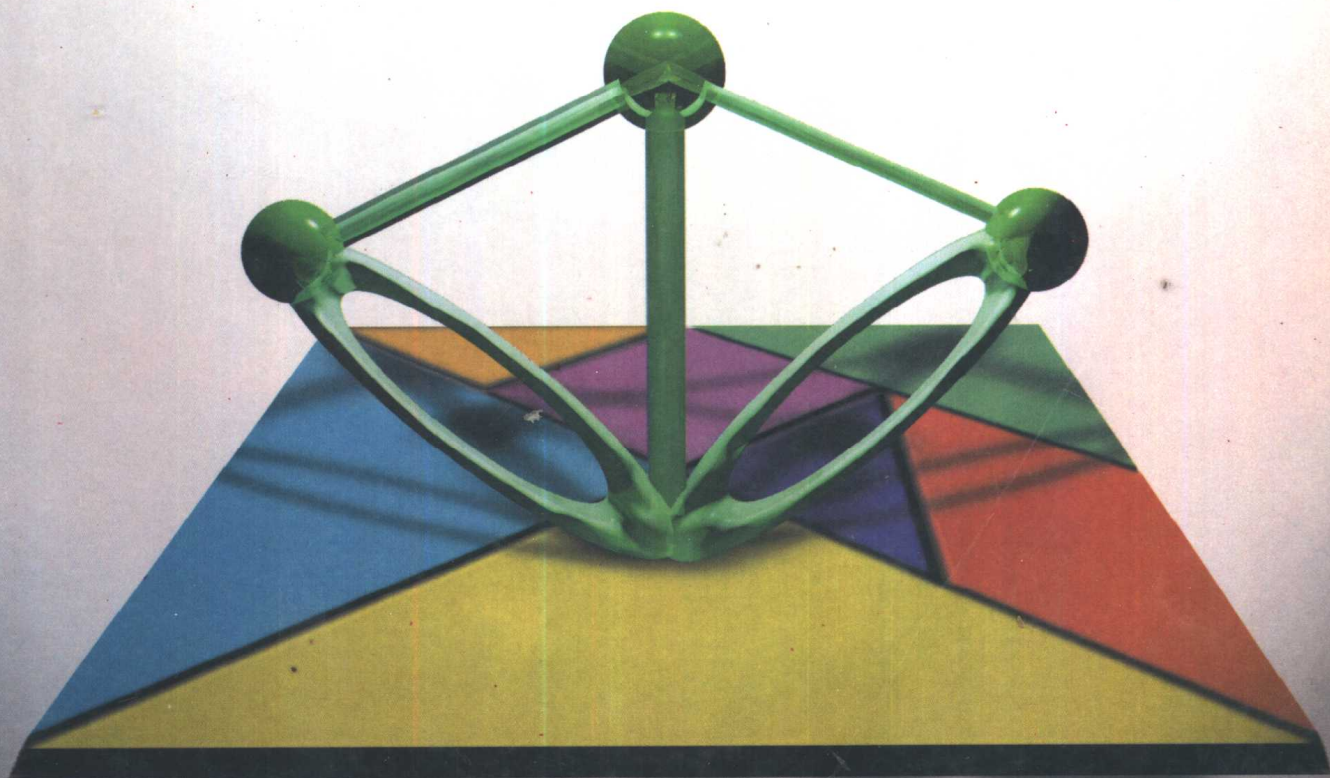


计算机科学组合学丛书

计算机算法导引

设计与分析

卢开澄 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

计算机科学组合学丛书(三)

计算机算法导引

——设计与分析

卢开澄 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书为《组合数学(算法与分析)》下册的第二版,书中内容、结构都作了极大的改变。本书讨论了动态规划、优先策略、分治策略、线性规划的分解原理、最佳二分树、密码学等 29 个问题,对算法和它的复杂性作了分析。本书可作为计算机系本科学生及研究生教材,对数学系师生和科研工作者可作为参考书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

计算机算法导引:设计与分析/卢开澄编著. —2 版. —北京:清华大学出版社,1996.10
ISBN 7-302-02277-1

I. 计… II. 卢… III. ① 电子计算机-计算方法-设计 ② 电子计算机-计算方法-分析(数学) IV. TP301.6

中国版本图书馆 CIP 数据核字(96)第 16193 号

出版者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者:北京市密云胶印厂

发行者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:21 字数:493 千字

版 次:1996 年 11 月第 1 版 2000 年 12 月第 5 次印刷

书 号:ISBN 7-302-02277-1/TP·1118

印 数:11001~14000

定 价:21.00 元

计算机科学组合学丛书

前 言

电子计算机的出现是 20 世纪的大事,它改变了我们这个世界的面貌。可以毫不夸张地说,它的影响遍及所有的角落,几乎无处不感觉到它的存在。数学更不例外。严格地说,电子计算机本身就是近代数学的辉煌成就。将计算机与数学割裂开来,既不合理也不可能。组合学也就是在计算机科学蓬勃发展的刺激下而崛起的,从而成为近若干年来最活跃的数学分支。它研究的问题有的可追溯到欧拉和哈密尔顿等 18 世纪的数学家,但它成为一新的分支还是近若干年的事。它从与计算机科学相结合中获得了广阔的发展空间,从而也为计算机科学奠定了理论基础。

什么是计算机科学呢?有的学者将它定义为研究算法的一门学科。研究算法无疑是计算机科学的重要领域,也是本丛书的核心内容,贯穿始终。组合学家在 20 世纪 70 年代初建立的算法复杂性“NP 理论”,至今仍然令无数计算机科学工作者与数学工作者为之折腰。

计算机科学里的组合学内容十分广泛。本丛书涉及组合分析、图论、组合算法、近代密码学、编码理论及算法复杂性等七部分。

组合分析是算法的理论基础。组合分析之与组合算法犹如数学分析之与计算数学,众所周知,前者是后者的理论根基。

图论原本是组合数学这个“家族”的主要成员,只因它已成长壮大,故自立门户独立出去。

算法复杂性的 NP 理论是近 30 年的一大成就。研究表明对于一类叫做 NPC 类的困难问题,至今都不存在有效算法,但它们难度相当,只要其中任何一个找到多项式解法,则全体都获得解决;或证明它们根本不存在有效办法。不论是前者还是后者都还看不见露到海平面上的桅杆塔,它吸引了众多的有志之士。密码学是其中十分引人入胜的分支。如若设计好的密码,对它的破译等价于某一 NPC 类困难问题,无疑这样的密码将是牢不可破的。

在计算机网络深入普及的信息时代,信息本身就是时间,就是财富。信息的传输通过的是脆弱的公共信道,信息储存于“不设防”的计算机系统中,如何保护信息的安全使之不被窃取及不至于被篡改或破坏,已成为当今被普遍关注的重大问题。密码是有效而且可行的办法。在计算机网络的刺激下,近代密码学便在算法复杂性理论的基础上建立起来了。密码作为一种技术,自从人类有了战争,不久便有了它。但作为一门学科则是近 20 多年的事。甚至于它已成为其它学科的基础。密码也从此走出“军营”,进入百姓家。

实际中的“优化”问题是大量的,半个多世纪以来它曾经几度辉煌。近来在计算机科学的影响下,又出现了若干闪光点,十分耀眼,引人注目。

实际上密码也是一种编码。如果说密码学研究的编码是保证通信的保密与安全,则编码理论研究的是通信中如何纠错与检错。计算机纠错码是既实用、理论上又饶有趣味的分支。

本丛书是作者在清华大学计算机科学与技术系长期工作的总结。它不是一部“长篇”记述,而是互相关联又彼此相对独立,因此难免有少量交叉。它们涉及的面如此之广,囿于作者的水平,缺点和错误在所难免,敬请读者不吝指正。谢谢。

序 言

有人说“计算机科学是一门研究算法的科学”。不论这个说法是否全面,算法无疑是计算机科学的重要组成部分。它近来发展极其迅速,说是异彩纷呈并不为过。“算法与算法复杂性分析”已是计算机专业本科生,特别是研究生的一门必需掌握的内容。

与算法有关的还有一个大家熟悉的公式:

程序=算法+数据结构

这说明算法的研究不单是数学问题,还和数据结构密切相关。这个观点在这里必须突出地强调,必须强调的还有一点,那就是“实践”,只有通过动手实践才能掌握算法的实质。正因为这个原因,实例是本书的重要组成部分。

本书是在“组合数学(算法与分析)”下册的基础上改写而成。第1章至第5章及第18章、第19章由卢华明执笔,第6章至第12章及第16章由黄连生完成。没有他们的合作,本书的出版可能还得拖相当一段时间。作者深知书中存在不少缺点与错误,还望读者不吝指教。

绪 论

快速电子计算机的出现是本世纪的一件大事,因为它改变了我们这个世界的面貌。可以毫不夸张地这么说,今天没有一个角落不感觉到它的存在,无处不感受到它的影响。

快速电子计算机贵在神速,也就是它的运算速度快,同时它的发展之迅速也是惊人的。从第一台计算机问世到现在才 50 个年头,50 岁对于一个人来说才过中年,可是计算机已从每秒 5000 次运算,发展到现在的运算速度每秒数以亿次计。元器件从继电器、真空管、晶体管、集成电路、大规模集成电路,发展到超大规模集成电路。结构上从每台机器作为运算工具的“单兵作战”模式,发展到今天将范围遍及各大洲的计算机网络,算法也从单机的串行计算发展到网络上并行的分布计算。

一台作 10^9 次/秒运算的计算机的效率超过 10 亿人同时工作。它可以作中期的天气预报,可以控制庞大的化工厂生产过程,以至于还可以驾驭现代化战争,从这个意义上来说它确实是近乎神奇的。不过必须强调的是这些都是在人的安排下进行工作的。以人们利用计算机作天气预报为例,必须依据天气预报的偏微分方程及编好程序,指导计算机依照人的意志一步步地工作。

在强调计算机虽然神通广大,还是在人的控制下工作的同时,还应说明计算机并非什么都能做,有的事情理论上它根本做不了。讨论哪些事计算机能做,哪些事计算机做不了,这属于可计算性理论研究的范畴。本书后面还将看到有些问题理论上计算机虽是能做的,但实际上又是不可能完成的,比如说开动计算机必须运行几个、几十个世纪以至更长时间,它实际上是不能完成的。比如拿最简单例子,26 个英文字母全排列,它的排列数为:

$$26! \approx 4 \times 10^{26}$$

以每年 365 天计算,共有

$$365 \times 24 \times 3600 = 3.1536 \times 10^7 \text{ 秒}$$

以每秒能完成 10^7 个排列的超高速电子计算机来做这项工作,需要

$$4 \times 10^{26} / (3.1536 \times 10^4) \approx 1.2 \times 10^{12} \text{ 年}$$

即使计算机运行速度随着技术的提高,恐怕也还是不可能实现的。因计算机的速度再提高也有它的极限。这里是在假定计算机从不休息的前提下作估计的。

“算法与算法复杂性”是研究算法的一门学科。它还很年轻远未定型,还处在发展中。本书是采取一个问题一个问题地讨论的方式进行,这里的每个问题也可能是一种算法,也可能是某一个解题的对象即问题。首先还要回答:什么是算法。如若某一种方法不仅适用于解决具体的一个问题,而是能用之解决若干个问题,则这个方法便可称之为一种算法了。

评价算法的效率是要对它所需要的时间和存贮单元进行估计,前者为时间复杂性分析,后者为空间复杂性分析。

如何描述算法步骤呢?本书采用自然语言叙述的办法,避免用类似于某种高级语言

的形式化办法。有些算法的实质是极为直观,非要形式化不可,则可能适得其反将问题异乎寻常地复杂化了。实例更是帮助正确理解算法的重要环节。总之,通过算法的描述,辅以例子,力求将它理解得正确。作者有这样体会,有的方法只有通过自始至终实践一遍才能了解透彻。

本书用到一些符号和术语,也在这里介绍一下。

$N = \{0, 1, 2, \dots\}$; $R =$ 实数集合;

$N^+ = \{1, 2, \dots\}$; $R^+ =$ 正实数集合;

$R^* = R^+ \cup \{0\}$ 。

令 f 和 g 是由 N 到 R^* 的两个函数。

定义 令 $f: N \rightarrow R^*$,

$O(f) \triangleq \{g: N \rightarrow R^* \mid \exists c \in R^*, \exists n_0 \in N, \forall n \geq n_0, \text{使 } g(n) \leq cf(n)\}$

也就是 $O(f)$ 是函数 $g: N \rightarrow R^*$ 的集合,它以 $cf(n)$ 为上界。亦即若存在 $c \in R^*$, 且 $\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = c$, 则 $g \in O(f)$ 。基于上述概念,若 $f(n) = n^3, g(n) = n^2$, 则 $g \in O(f)$, 但 $f \notin O(g)$ 。

定义 令 $f: N \rightarrow R^*$,

$\Omega(f) \triangleq \{g: N \rightarrow R^* \mid \exists n_0 \in N, \exists c \in R^+, \forall n \geq n_0, \text{使 } g(n) \geq cf(n),$

若 $\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = \infty$ 或 $\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = c > 0$, 则 $g \in \Omega(f)$ 。

定义 令 $f: N \rightarrow R^*$, $\Theta(f) = O(f) \cap \Omega(f)$,

若 $\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = c, c \in R^+$, 则 $g \in \Theta(f)$ 。

其中 $c \in R^+$ 表示 $c \neq 0, c \neq \infty$, $g \in \Theta(f)$ 意味着 g 和 f 同阶。

O, Ω, Θ 有以下的性质(假定 $f, g, h: N \rightarrow R^*$):

(1) 若 $f \in O(g), g \in O(h)$, 则 $f \in O(h)$;

(2) $f \in O(g) \Leftrightarrow g \in \Omega(f)$;

(3) 若 $f \in \Theta(g)$, 则 $g \in \Theta(f)$;

(4) $O(f+g) = O(\max\{f, g\})$ 。

这 4 个性质的证明留作习题。

目 录

绪论	IX
第 1 章 动态规划	1
1.1 最短路径问题	1
1.2 最佳原理	3
1.3 流动推销员(或旅行商)问题.....	11
1.4 矩阵链乘问题.....	14
1.5 最长公共子序列.....	16
1.6 图的任意两点间的最短距离.....	18
1.7 整数规划问题.....	20
1.8 同顺序流水作业的任务安排问题.....	25
1.9 可靠性问题.....	27
1.10 设备更新问题	29
习题	33
第 2 章 优先策略	36
2.1 最短树的库鲁斯卡尔(Kruskal)算法	36
2.2 求最短树的普林(Prim)算法	37
2.3 求最短路径的戴克斯德斯(Dijkstra)算法	38
2.4 文件存储问题.....	39
2.5 有期限的任务安排问题.....	41
习题	42
第 3 章 分治策略	45
3.1 二分查找.....	45
3.2 整数乘法.....	46
3.3 矩阵乘积的斯德拉逊(Strassen)算法	47
3.4 矩阵乘积的维诺格拉德 Winograd 算法	50
3.5 布尔矩阵的乘法问题.....	51
习题	53
第 4 章 哈佛曼(Huffman)编码、FFT 算法和数据压缩	55
4.1 哈佛曼(Huffman)编码	55

4.2	快速傅里叶变换(FFT)	58
4.3	卷积及其应用	70
4.4	数论变换	72
	习题	74
第5章	线性规划的分解原理	76
5.1	线性规划和单纯形法简介	76
5.2	丹捷-卧佛(Dantzig-Wolfe)分解算法	81
	习题	89
第6章	最佳二分树	91
6.1	二分树	91
6.2	最佳二分树	94
	习题	100
第7章	内存分类法之一:插入分类法、塞尔(Shell)分类法	101
7.1	分类	101
7.2	分类的下界估计	101
7.3	二分插入分类法	104
7.4	塞尔(Shell)分类法	106
	习题	108
第8章	内存分类法之二:递选分类法、堆集分类	111
8.1	递选分类法	111
8.2	二分树递选分类法	112
8.3	堆集分类法	113
	习题	117
第9章	内存分类法之三:下溢分类法、快速分类法	118
9.1	下溢分类法	118
9.2	快速分类法	121
	习题	125
第10章	内存分类法之四:归并分类法和基数分类法	127
10.1	归并分类法	127
10.2	福德-庄生(Ford-Johnson)归并插入分类法	129
10.3	基数分类法	133
	习题	134

第 11 章 求第 k 个元素	135
11.1 求最小及第二小元素	135
11.2 求第 k 个元素	136
习题	138
第 12 章 外存分类法	139
12.1 外存归并分类法	139
12.2 置换选择段的构造	141
12.3 三条带的外存归并分类法	143
12.4 阶式归并法	147
习题	148
第 13 章 分类网络	149
13.1 分类网络举例	149
13.2 0-1 原理	150
13.3 归并网络	153
13.4 巴特塞尔(Batcher)奇偶归并网络	154
习题	156
第 14 章 查找及均衡树	157
14.1 AVL 树——关于高度均衡的二分树	157
14.2 关于高度均衡的二分树的插入和删除	161
习题	164
第 15 章 2-3 树和 2-3-4 树	165
15.1 2-3 树	165
15.2 2-3-4 树	167
15.3 红黑树	169
习题	170
第 16 章 B-树	171
16.1 B-树概念	171
16.2 插入和删除	172
习题	175
第 17 章 哈希表	176
17.1 什么是哈希表	176

17.2	哈希函数的构造方法	176
17.3	解决冲突的方法	177
17.4	哈希算法的分析(线性探测法分析)	180
17.5	二重哈希法	181
	习题	182
第 18 章	DFS 算法和 BFS 算法	184
18.1	概述	184
18.2	DFS 算法	185
18.3	无向图的 DFS 算法	187
18.4	有向图的 DFS 算法	189
18.5	互连通块问题	192
18.6	强连通块问题	193
18.7	BFS 算法	197
	习题	198
第 19 章	α-β 剪枝技术和分支定界法	200
19.1	α - β 剪枝技术	200
19.2	分支定界法和流动推销员问题	200
19.3	同顺序加工任务安排问题	204
	习题	207
第 20 章	整数规划	208
20.1	概述	208
20.2	0-1 规划和它的 DFS 搜索(隐枚举)解法	210
20.3	分支定界法在解整数规划中的应用	218
	习题	220
第 21 章	串匹配	221
21.1	概述	221
21.2	KMP 克鲁斯-摩尼斯-普拉特(Knuth-Morris-Pratt)算法	222
21.3	BM 坡艺尔-摩尔(Boyer-Moore)算法	224
21.4	RK 拉宾-卡普(Rabin-Karp)算法	225
	习题	226
第 22 章	概率算法	228
22.1	概率算法举例	228
22.2	随机数产生法	231

22.3	素数的概率判定算法	232
	习题	233
第 23 章	并行算法	234
23.1	并行计算机和并行算法的基本概念	234
23.2	递推关系的并行计算	237
23.3	图的并行算法举例	238
23.4	矩阵乘积的并行计算	242
23.5	分布计算	244
	习题	245
第 24 章	脉动阵列的并行处理	246
24.1	矩阵和向量乘法的并行处理	246
24.2	矩阵乘法的并行处理	247
24.3	带状矩阵的并行乘法	249
	习题	252
第 25 章	计算几何	253
25.1	关于线段问题	253
25.2	求凸包问题	257
	习题	259
第 26 章	NP 完备理论	260
26.1	确定型图灵机	260
26.2	可满足性问题	263
26.3	非确定型图灵机与库克(Cook)定理	265
26.4	几个 NP 完备的例子	269
26.5	复杂度类	277
	习题	279
第 27 章	近似算法	281
27.1	任务安排的近似算法	281
27.2	装箱问题的近似算法	285
27.3	流动推销员问题的近似算法	287
27.4	顶点覆盖问题的近似算法	294
	习题	295
第 28 章	密码学简介	297

28.1	什么是密码?	297
28.2	背包公钥密码.....	300
28.3	RSA 公钥密码	301
28.4	数字签名.....	303
28.5	Hash 算法	303
	习题.....	304
第 29 章	LP 问题的多项式算法	305
29.1	Klee 和 Minty 举例	305
29.2	Хачиян(哈奇扬)算法	308
29.3	Karmarkar 算法.....	311
	习题.....	321

第1章 动态规划

1.1 最短路径问题

数学规划是研究最优化的一类数学问题,包含有线性规划,非线性规划等内容。和动态规划有什么关系呢?动态规划实际上是研究一类最优化问题的算法,应用范围十分广。下面通过若干实例,介绍如何将一个最优化问题通过动态规划来求解的基本原理。

如图 1.1.1,从 A_0 点要铺设一条管道到 A_6 点,中间必须经过 5 个中间站,第一站可以在 A_1, B_1 两地中任选一个,类似地,第二、三、四、五站可供选择的地点分别是: $\{A_2, B_2, C_2, D_2\}, \{A_3, B_3, C_3\}, \{A_4, B_4, C_4\}, \{A_5, B_5\}$ 。连接两地间管道的距离(或造价)用连线上的数字表示,要求选一条从 A_0 到 A_6 的铺管线路,使总距离最短(或总造价最小)。

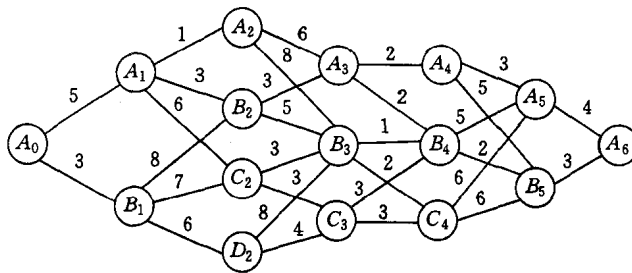


图 1.1.1

我们首先想到使用穷举法。在第一段,有两种路径选择: A_0A_1, A_0B_1 。在第二段,若选 A_0A_1 ,第二段路径有三种选择: A_1A_2, A_1B_2, A_1C_2 ;若选 A_0B_1 ,也有三种选择: B_1B_2, B_1C_2, B_1D_2 。所以两段共有 6 种选择。依次类推,从 A_0 到 A_6 共有 $2 \times 3 \times 2 \times 2 \times 2 \times 1 = 48$ 种不同路径。可通过 $48 \times 5 = 240$ 次加法,47 次比较,即通过各种可能方案的穷举,最后可求出从 A_0 到 A_6 的最短路径是:

$$A_0 \rightarrow A_1 \rightarrow B_2 \rightarrow A_3 \rightarrow B_4 \rightarrow B_5 \rightarrow A_6$$

相应的最短距离是 18。

但我们注意到最短路径有这样一个特性,即如果最短路径的第 k 站通过 P_k ,则这一最短路径在由 P_k 出发到达终点的那一部分路径,对于始点为 P_k 到终点的所有可能的路径来说,必定也是距离最短的。这一特性很容易证明。读者可自己来完成它。

根据最短路径这一特性,启发我们计算时从最后一段开始,从后向前逐步递推的方法,求出各点到 A_6 的最短路径,最后求得从 A_0 到 A_6 的最短路径。步骤如下:

$k=6$ 时

设 $f(A_5)$ 表示由 A_5 到 A_6 的最短距离, $f(B_5)$ 表示由 B_5 到 A_6 的最短距离,显然有:

$$f(A_5) = 4, \quad f(B_5) = 3.$$

$k=5$ 时

$$\begin{aligned} f(A_4) &= \min\{d(A_4, A_5) + f(A_5), d(A_4, B_5) + f(B_5)\} \\ &= \min\{\underline{3 + 4}, 5 + 3\} = 7 \end{aligned}$$

这里括号里的底线表示最小值所取的项。即 $f(A_4)$ 取的是 $A_4 \rightarrow A_5 \rightarrow A_6$ ，而不是 $A_4 \rightarrow B_5 \rightarrow A_6$ 。以后同此，不再说明。

$$\begin{aligned} f(B_4) &= \min\{d(B_4, A_5) + f(A_5), d(B_4, B_5) + f(B_5)\} \\ &= \min\{5 + 4, \underline{2 + 3}\} = 5 \end{aligned}$$

$$\begin{aligned} f(C_4) &= \min\{d(C_4, A_5) + f(A_5), d(C_4, B_5) + f(B_5)\} \\ &= \min\{6 + 4, \underline{6 + 3}\} = 9 \end{aligned}$$

$k=4$ 时

$$\begin{aligned} f(A_3) &= \min\{d(A_3, A_4) + f(A_4), d(A_3, B_4) + f(B_4)\} \\ &= \min\{2 + 7, \underline{2 + 5}\} = 7 \end{aligned}$$

$$\begin{aligned} f(B_3) &= \min\{d(B_3, B_4) + f(B_4), d(B_3, C_4) + f(C_4)\} \\ &= \min\{\underline{1 + 5}, 2 + 9\} = 6 \end{aligned}$$

$$\begin{aligned} f(C_3) &= \min\{d(C_3, B_4) + f(B_4), d(C_3, C_4) + f(C_4)\} \\ &= \min\{\underline{3 + 5}, 3 + 9\} = 8 \end{aligned}$$

$k=3$ 时

$$\begin{aligned} f(A_2) &= \min\{d(A_2, A_3) + f(A_3), d(A_2, B_3) + f(B_3)\} \\ &= \min\{\underline{6 + 7}, 8 + 6\} = 13 \end{aligned}$$

$$\begin{aligned} f(B_2) &= \min\{d(B_2, A_3) + f(A_3), d(B_2, B_3) + f(B_3)\} \\ &= \min\{\underline{3 + 7}, 5 + 6\} = 10 \end{aligned}$$

$$\begin{aligned} f(C_2) &= \min\{d(C_2, B_3) + f(B_3), d(C_2, C_3) + f(C_3)\} \\ &= \min\{\underline{3 + 6}, 3 + 8\} = 9 \end{aligned}$$

$$\begin{aligned} f(D_2) &= \min\{d(D_2, B_3) + f(B_3), d(D_2, C_3) + f(C_3)\} \\ &= \min\{8 + 6, \underline{4 + 8}\} = 12 \end{aligned}$$

$k=2$ 时

$$\begin{aligned} f(A_1) &= \min\{d(A_1, A_2) + f(A_2), d(A_1, B_2) + f(B_2), d(A_1, C_2) + f(C_2)\} \\ &= \min\{1 + 13, \underline{3 + 10}, 6 + 9\} = 13 \end{aligned}$$

$$\begin{aligned} f(B_1) &= \min\{d(B_1, B_2) + f(B_2), d(B_1, C_2) + f(C_2), d(B_1, D_2) + f(D_2)\} \\ &= \min\{8 + 10, \underline{7 + 9}, 6 + 12\} = 16 \end{aligned}$$

$k=1$ 时

$$\begin{aligned} f(A_0) &= \min\{d(A_0, A_1) + f(A_1), d(A_0, B_1) + f(B_1)\} \\ &= \min\{\underline{5 + 13}, 3 + 16\} = 18 \end{aligned}$$

上述计算结果可表示如图 1.1.2, 其中 $\textcircled{\frac{A_5}{4}}$ 表示从 A_5 出发到终点的最短路径长度为

4, 即 $A_5 \xrightarrow{4} A_6$, 余此类推。

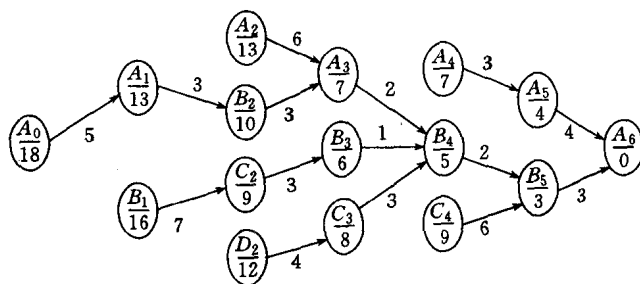


图 1.1.2

一共要用 15 次比较运算和 28 次加法运算就可得到从 A_0 到 A_6 的最短距离,而且在这过程中,还得到其它各点到 A_6 的最短路径和最短距离。

一般地,若我们考虑如图 1.1.3 所示从始点 $O(0,0)$ 到终点 $E(m,n)$ 的最短路径(也称格路)问题。若用穷举法,则需

$$(m+n-1)C(n+m,n) = \frac{(n+m-1)(n+m)!}{n!m!}$$

次加法及 $\frac{(n+m)!}{n!m!} - 1$ 次比较运算。组合数 $C(n+m,n)$

为从 O 点到 E 点的路径数,这是考虑到图 1.1.3 的每一格路和 m 个 x , n 个 y 的任一排列一一对应。比如排列 $\underbrace{x \cdots x}_m \underbrace{y \cdots y}_n$, 对应于从 O 点先沿 x 方向走 m 块,后沿 y 方向走 n 块到 E 点。相当于由 m 个 y , n 个 x 构成的长度为 $m+n$ 的符号串数目,即从 $m+n$ 个格子中任选 m 个格子作为 x 、其余为 y 的方案数。

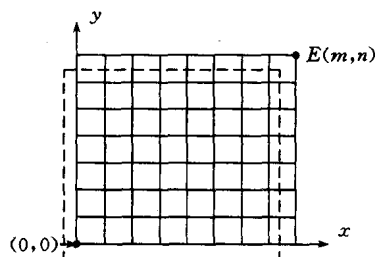


图 1.1.3

但用后一种方法只需 $2mn+m+n$ 次加法及 mn 次比较就够了。不难知道图 1.1.3 由虚线包围的矩形域内的点要作 2 次比较,2 次加法,在这以外的点只需 1 次加法,无需作比较。

当 $m=n$ 时,穷举法需进行 $(2n-1) \cdot (2n)! / (n!)^2$ 次加法, $(2n)! / (n!)^2 - 1$ 次比较。后一种方法只要作 $2(n^2+n)$ 次加法, n^2 次比较。由 Stirling 公式

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$$

可知穷举法的运算量是 n 的指数函数,后一种算法则只是 n^2 量级。

1.2 最佳原理

从前节例子知道:一个最短路径问题可变成多段判决问题,利用了最短路径的一个性质:从起点到终点的最短路径也是该路径上各点到终点的最短路径。与此类似的问题

很多,故可抽象成组合优化问题中的一个重要的最佳原理:假设为了解决某一优化问题,需要依次作出 n 个决策 D_1, D_2, \dots, D_n , 如若这个决策序列是最优的,对于任何一个整数 $k, 1 < k < n$, 不论前面 k 个决策是怎样的,以后的最优决策只取决于由前面决策所确定的当前状态,即以后的决策 $D_{k+1}, D_{k+2}, \dots, D_n$ 也是最优的。

本章的这节和以后各节主要举例说明如何灵活运用最佳原理。动态规划与线性、非线性规划不尽相同,实际它是算法的一种。最佳原理说起来很简单,如何灵活应用它则又是另一回事。它用途很广,利用它来解决一个新问题的本身也就是一种创造。我们希望下面的例子能帮助读者掌握它的技巧,并达到举一反三。

例 1 某工厂购进 1000 台机器,准备生产 P_1, P_2 两种产品,若生产产品 P_1 , 每台机器每年可收入 50 千元,损坏率达 65%;若生产产品 P_2 , 每台机器年收入为 40 千元,但损坏率仅有 40%;估计三年后将会有新的机器出现,旧的机器将全部淘汰,试问应如何安排生产,使三年内收入最多? 计划以一年为周期。

以上的问题原可以化为整数规划问题求解。设 x_1, x_2, x_3 分别是第 1、2、3 年中用以生产产品 P_1 的机器数, y_1, y_2, y_3 分别是第 1、2、3 年用以生产产品 P_2 的机器数。则得到求

$$\max z = 50000(x_1 + x_2 + x_3) + 40000(y_1 + y_2 + y_3)$$

约束条件

$$\begin{cases} x_1 + y_1 = 1000 \\ x_2 + y_2 = 0.35x_1 + 0.60y_1 \\ x_3 + y_3 = 0.35x_2 + 0.60y_2 \end{cases}$$

$x_i, y_i \geq 0$ 的整数, $i=1, 2, 3$

现应用最佳原理变成多段判决问题如下:

设 $P_i(n)$ 为 n 台机器在以后 i 年内的最大收益。若只考虑安排一年的生产, x_3 为生产 P_1 的机器数目。从 $i=1$ 开始,即一年后的最大收益为:

$$\begin{aligned} P_1(n) &= \max_{0 \leq x_3 \leq n} \{50000x_3 + 40000(n - x_3)\} \\ &= 50000n \end{aligned}$$

即

$$x_3 = n$$

进而考虑 $i=2$, 即若考虑两年的生产, x_2 为两年中第一年生产 P_1 的机器数目,则一年后机器剩余数为 $0.35x_2 + 0.60(n - x_2)$, 故有

$$P_2(n) = \max_{0 \leq x_2 \leq n} \{50000x_2 + 40000(n - x_2) + P_1(0.35x_2 + 0.60(n - x_2))\}$$

由于 $P_1(k) = 50000k$, 故

$$\begin{aligned} P_2(n) &= \max_{0 \leq x_2 \leq n} \{50000x_2 + 40000n - 40000x_2 + 50000(0.60n - 0.25x_2)\} \\ &= \max_{0 \leq x_2 \leq n} \{-2500x_2 + 70000n\} \\ &= 70000n \end{aligned}$$

即

$$x_2 = 0$$

最后考虑 $i=3$, 若考虑三年的生产, 第一年生产 P_1 的机器数目设为 x_1 , 则有

$$P_3(1000) = \max_{0 \leq x_1 \leq 1000} \{50000x_1 + 40000(1000 - x_1) + P_2(0.35x_1 + 0.60(1000 - x_1))\}$$