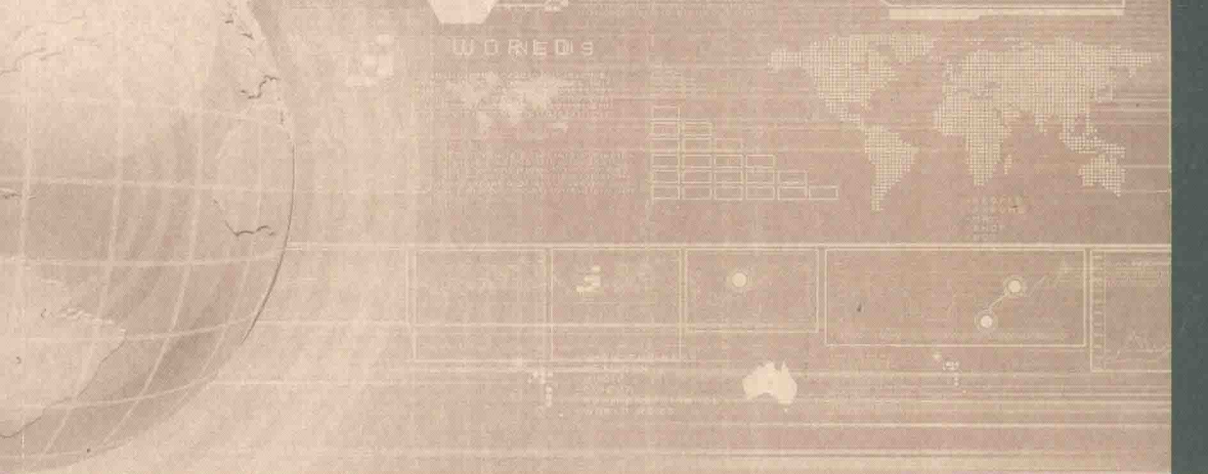




网络流优化高效智能 算法及其应用

谢凡荣 贾仁安 著



科学出版社

网络流优化高效智能算法及其应用

谢凡荣 贾仁安 著

科学出版社

北 京

内 容 简 介

本书从便于计算机求解角度深入探讨网络流优化的理论应用前沿问题,与实际紧密结合,具有实用价值。本书首先介绍网络流优化的 8 个主要问题及其高效智能算法,接着介绍网络流优化高效智能算法在解决 16 个相关优化问题中的应用创新,最后介绍 2 个实际应用案例。书中所有算法都已在计算机上编程实现。两个附录分别列出实现最小费用最大流算法 MCMF-NA 的 C++ 核心源代码(C++类)和实现多级供应链优化基于生成树改进遗传算法 rst-GA 的 C 语言关键源代码。

本书可作为数学专业、计算机科学与技术专业、管理科学与工程专业等理工类和经管类高年级本科生与研究生学习网络流优化理论及其应用的教材或参考书,也可作为有关研究人员的参考书。

图书在版编目(CIP)数据

网络流优化智能算法及其应用/谢凡荣,贾仁安著. —北京:科学出版社, 2015

ISBN 978-7-03-043838-6

I. ①网… II. ①谢… ②贾… III. ①网络流—算法理论 IV. ①O157.5

中国版本图书馆 CIP 数据核字(2015)第 055215 号

责任编辑:魏如萍 / 责任校对:张怡君
责任印制:霍 兵 / 封面设计:无极书装

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

文林印务有限公司 印刷

科学出版社发行 各地新华书店经销

*

2015 年 4 月第 一 版 开本:720×1000 1/16

2015 年 4 月第一次印刷 印张:19 1/2

字数:420 000

定价:86.00 元

(如有印装质量问题,我社负责调换)

前 言

优化的根本目的就是在原有基础上改善,并力求在考虑范围内找到最佳结果。优化技术是一种以数学为基础,用于寻求各种实际问题优化的应用技术。网络优化是图论的一个分支,在计算机管理、工程设计等学科,网络优化有着十分广泛的应用。网络优化就是研究如何有效地计划、管理和控制网络系统,使之发挥最大的社会和经济效益;就是研究与(赋权)图有关的最优化问题。国内外不少学者在从事网络优化的研究。

网络优化中基本问题包括最短路问题、最大容量路问题、最大流问题、最小费用最大流问题、最小费用流问题、最大利润流问题、最小饱和流问题等。实际中的许多问题,如运输问题、指派问题、项目赶工问题等,都可以转化为这些基本问题来解决。国内外有不少学者对这些基本问题及其应用进行了系统深入的研究,并且取得了很好的研究成果。例如,求解最短路问题的常用算法有 Dijkstra 算法和 Floyd 算法,求解最大流问题的常用算法有 Ford-Fulkerson 算法和 Dinic 算法等,它们可广泛用于求解其他相关优化问题。所谓算法,就是一种搜索过程或规则,它是基于某种思想和机制,通过一定途径或规则得到满足用户要求的问题的解的方法。

现实中网络优化问题往往模型十分复杂(如大规模、非线性、不连续、多目标等),有不少学者采用遗传算法(Genetic Algorithm)等软计算方法来研究网络优化问题的模型求解。软计算方法是对生物系统及其行为特征进行模拟而开发出的适合大规模并行计算且具有智能特征的通用性较强的计算、优化模式和方法,它包括通过对人类模糊思维方式的模拟而产生的模糊计算方法、通过自然界中生物进化机制的模拟而产生的进化计算方法和通过对动物脑神经的模拟而产生的神经计算方法等。作为进化计算方法,遗传算法是一类借鉴生物界的进化规律(适者生存,优胜劣汰遗传机制)演化而来的随机化搜索方法;它是由美国的 J.Holland 教授于 1975 年首次提出,其主要特点是直接对结构对象进行操作,不存在求导和函数连续性的限定;具有内在的隐并行性和更好的全局寻优能力;采用概率化的寻优方法,能自动获取和指导优化的搜索空间,自适应地调整搜索方向,不需要确定的规则。采用遗传算法等软计算方法来研究网络优化问题是网络优化研究的一个热点。

由于人们一般同时追求多个目标,因此研究网络优化问题的多目标规划模型与算法一直是国内外学者关注的一个主题。由于问题的复杂性与人的有限理性,人们对问题最优解的追求往往让位于追求满意解或近似最优解或有效解(非劣

解)。而所谓启发式算法,就是通过仔细设计启发函数来选择当前结点的下一步结点的搜索方法,往往在很短的时间内就可搜索到问题的最优解,也往往可在多项式时间内搜索到 NP 问题的较优解,但无法证明它不会得到较坏的解。有时人们发现,在某些特殊情况下启发式算法会得到很坏的答案或效率极差,然而造成那些特殊情况的数据结构在现实中也许永远不会出现。启发式算法在解决实际问题中十分常用。因此,网络优化问题的启发式算法研究也是国内外学者十分关注的主题。

随着计算机的发展和普及,大规模的复杂问题可以借助于计算机技术来解决。这就要求人们建立决策支持系统。为了便于建立有关的决策支持系统,笔者致力于探究网络优化中最大容量路问题、最大流问题、最小费用最大流问题、最小费用流问题、最大利润流问题、最小饱和流问题等问题的便于计算机编程实现的高效数值求解算法及其应用,并取得了系列研究成果。在此基础上,本书对网络流优化高效智能算法及其应用,进行了深入探讨,系统介绍我们在这方面取得的研究成果。

网络流理论是网络优化的重要组成部分。对网络流理论及其应用进行研究一直是国内外学者关注的主题。网络常指赋权有向连通图,权是与弧(或顶点)有关的数量指标;根据实际问题的需要,可以赋予权不同的含义,如表示距离、时间、费用、承载能力、可靠性等;可以给顶点赋权,也可以给弧赋权;可以赋一个权,也可以赋多个权;权可以是非负数,也可以是负数。赋权图在图论及其应用中有着重重要的地位,它不仅指出各顶点间的邻接关系,而且同时也表示各顶点间的数量关系。赋权图被广泛应用于解决工程技术及科学生产管理等领域的优化问题。在经典网络流理论中,对每一条弧给定一个表示弧的最大输送量的权(非负数,称为弧的容量),给定另一个表示通过弧的物质或信息流的数量的权(非负数,称为弧的流);网络可行流是指定义在所有弧上的满足容量约束和平衡约束的一组非负数。有上下界网络拓广了经典网络的概念,定义了弧的容量上界表示弧的最大输送量(非负数),还定义了弧的容量下界表示弧的最小输送量(非负数);可以把经典网络看做是所有弧容量下界为零弧容量上界非负的有上下界网络。经典网络中总存在可行流,但有上下界网络中不一定存在可行流。对有上下界网络,可类似定义最大流、最小流、最小费用流、最小费用最大流等概念;研究有上下界网络的最大流、最小流、最小费用流、最小费用最大流等的存在性及其构造性理论与算法,是具有理论应用价值的研究课题。有上下界网络流理论在管理中有着广泛的应用前景,例如,最短工期项目计划问题可化为有上下界网络最大流与最小截问题来求解,城市公交车的线路安排问题、水资源的合理调配问题、人员安排问题、货物运输问题等都可以归结为有上下界网络最小费用流与最小费用最大流问题来解决。最短路问题可看作有上下界网络最小费用流问题的一种特例,且是网络流理论的重要基础,最大容量路问题在高效求解网络最大流问题中能发挥十分重要的作用。因此本书首先介绍最短路问题和最大容量路问题;第 1~8 章是网络流优化理论方法的原始创新,第 9~25 章是网络流优化理论方法的应用创新。

本书共 25 章, 前 8 章介绍网络流优化的 8 个主要问题(即网络最短路问题、网络最大容量路问题、网络最大流问题、网络最小费用最大流问题、有上下界网络最大流与最小流问题、有上下界网络最小费用流与最小费用最大流问题、网络最大利润流问题、网络最小饱和流问题)及其高效数值求解算法, 第 9~24 章介绍网络流优化高效智能(数值求解)算法在解决相关优化问题中的应用创新, 第 25 章介绍两个实际应用案例。第 1~24 章的每一章介绍一个问题及其高效智能(数值求解)算法。本书介绍的所有算法都已用 Visual Basic 5.0 或 Visual C++ 6.0 在计算机上编程实现, 并举例说明了算法的应用; 为了方便有兴趣的读者能尽快进入这一研究领域, 书末给出了两个附录(即附录 A 和附录 B), 附录 A 列出了实现最小费用最大流数值算法 MCMF-NA 的 C++ 程序核心源代码(C++ 类), 附录 B 列出了实现多级供应链优化基于生成树改进遗传算法 rst-GA 的 C 语言程序关键源代码, 可供读者调用或参考。

本书能为建立相关决策支持系统提供帮助, 可使某些大规模复杂问题的解决程序化。出版本书, 是笔者多年从事网络流优化高效智能算法及其应用研究的一个总结, 是在这一研究领域进一步深入研究的新起点。本书介绍的研究成果, 是多项国家级、省部级等研究项目(第一作者谢凡荣作为主要完成人(排名第二)完成了国家自然科学基金项目(实代数几何中构造性理论与算法(10761006), 已结题), 主持(排名第一)完成了江西省自然科学基金(网络优化高效能算法及其在管理中的应用(2010GZS0151), 已结题), 主持(排名第一)完成了江西省高等学校教学研究省级一般课题(运筹学课程教学改革研究(JXJG-04-1-69), 已结题), 作为主要完成人(排名第七)完成了国家自然科学基金项目(江西农村规模养殖能源生态系统管理对策反馈仿真研究(70761004), 已结题), 作为主要完成人(排名第二)完成了江西省自然科学基金(拟变分不等式理论及其应用(2007GZS2120), 已结题), 作为主要完成人(排名第二)完成了江西省教育厅项目(交通配流理论、实证与控制策略设计研究(赣教技字[2007]10 号), 已结题), 作为主要成员(排名第三)参加了国家自然科学基金项目(反馈分析方法系与低碳生态能源系统工程互促的深入研究(71261018), 在研))支持的结晶。本书的出版, 受到“江西省‘十二五’重点学科‘管理科学与工程’出版项目”资助, 还得到“南昌大学博士科研启动项目”资助。

谢凡荣 贾仁安

2015 年 1 月

目 录

前言	i
第 1 章 网络最短路问题	1
1.1 图的基本概念	1
1.2 最短路问题	3
1.3 结论	5
第 2 章 网络最大容量路问题	6
2.1 基本概念与理论	6
2.2 数值算法	8
2.3 应用举例	9
2.4 结论	10
第 3 章 网络最大流问题	11
3.1 概念和依据	11
3.2 数值算法	13
3.3 应用举例	14
3.4 结论	15
第 4 章 网络最小费用最大流问题	16
4.1 概念与依据	16
4.2 数值算法	24
4.3 应用举例	27
4.4 结论	30
第 5 章 有上下界网络最大流与最小流问题	31
5.1 概念和依据	31
5.2 数值算法	36
5.3 应用举例	38
5.4 结论	43
第 6 章 有上下界网络最小费用流与最小费用最大流问题	44
6.1 理论与算法	44
6.2 应用举例	49
6.3 结论	51

第 7 章	网络最大利润流问题	52
7.1	概念和依据	52
7.2	数值算法	56
7.3	应用举例	57
7.4	结论	58
第 8 章	网络最小饱和流问题	60
8.1	遗传算法	60
8.2	数值实验	64
8.3	结论	72
第 9 章	管理安排问题	73
9.1	概念和依据	73
9.2	启发式数值算法	76
9.3	案例	77
9.4	结论	80
第 10 章	车间最优逐月生产计划问题	82
10.1	问题描述及其数学模型	82
10.2	数学模型的求解	83
10.3	数值算法	84
10.4	算例	86
10.5	结论	86
第 11 章	多阶段存储问题	87
11.1	问题描述及其数学模型	87
11.2	数学模型的求解	88
11.3	数值算法	89
11.4	应用举例	91
11.5	结论	91
第 12 章	最短工期项目计划问题	92
12.1	基于运输网络最大流最小截启发式数值算法	92
12.2	基于有上下界网络最大流最小截数值算法	102
第 13 章	有重要应用价值的一类线性规划问题	115
13.1	理论与算法	115
13.2	应用举例	119
13.3	结论	120
第 14 章	运输问题	122
14.1	问题描述及其数学模型	122
14.2	数学模型的求解	124

14.3	数值算法	125
14.4	应用举例	126
14.5	结论	127
第 15 章	指派问题	128
15.1	问题描述及其数学模型	128
15.2	数学模型的分析与求解	129
15.3	数值算法	130
15.4	应用举例	131
15.5	结论	132
第 16 章	缺省指派问题	133
16.1	问题描述及其数学模型	133
16.2	数学模型的分析求解	134
16.3	数值算法	136
16.4	应用举例	140
16.5	结论	142
第 17 章	运输问题的多反而少悖论	143
17.1	模型与算法	143
17.2	应用举例	145
17.3	结论	147
第 18 章	有容量限制与边界条件约束运输问题	148
18.1	模型与算法	148
18.2	应用举例	150
18.3	结论	151
第 19 章	供给总量限定需求区间约束运输问题	152
19.1	问题描述及其数学模型	152
19.2	数学模型的求解	153
19.3	数值算法	157
19.4	应用举例	161
19.5	结论	163
第 20 章	运输时限费用优化问题	164
20.1	问题描述及其数学模型	164
20.2	数学模型的求解	165
20.3	应用举例	169
20.4	结论	170
第 21 章	固定费用运输问题	171
21.1	模型与算法	171

21.2	应用举例	175
21.3	结论	176
第 22 章	固定费用运输问题的多反而少悖论	177
22.1	模型与算法	177
22.2	应用举例	182
22.3	结论	183
第 23 章	非线性固定费用运输问题	184
23.1	问题描述及其数学模型	184
23.2	遗传算法	185
23.3	数值实验	190
23.4	结论	195
第 24 章	多级供应链优化问题	196
24.1	问题描述及其数学模型	196
24.2	基于生成树改进遗传算法	198
24.3	基于生成树改进遗传算法的 C 语言实现方法	206
24.4	数值例子	207
24.5	结论	210
第 25 章	应用案例	211
25.1	“协会+农户”生猪产业供应链网络饲料运送 和生猪农产品销售运输 最优方案计算及应用	211
25.2	银河杜仲沼液种植三层目标轮换六级储存网络系统设计	218
参考文献		230
附录 A	实现最小费用最大流数值算法 MCMF-NA 的 C++核心源代码(C++类)	238
附录 B	实现多级供应链优化基于生成树改进遗传算法 rst-GA 的 C 语言关键源代码	254

第1章 网络最短路问题

1.1 图的基本概念

生活中的示意图(由点、线组成)用来反映对象间的关系,点表示对象,线表示关系。这样的例子有很多,比如,铁路交通图、电话线分布图、煤气管道图、航空线图、比赛情况图。在示意图中,点的位置,线的长短曲直,对反映对象间的关系不重要。从本质上讲,示意图与工程图、几何图不同。关系有对称与非对称之分;对称关系用无向线(不带箭头的线)表示,非对称关系用有向线(带箭头的线)表示。

图是由点(顶点)、线组成,不带箭头的线(无向线)又称为边,带箭头的线(有向线)又称为弧。

无向图:由点(顶点)、边组成,记为 $G=(V,E)$, V 是 G 的点(顶点)集, E 是 G 的边集。联结点(顶点) $v_i, v_j \in V$ 的边,记为 $[v_i, v_j]$ 或 $[v_j, v_i]$ 。

有向图:由点(顶点)(node, vertex)、弧(arc)组成,记为 $D=(V,A)$, V 是 D 的点(顶点)集, A 是 D 的弧集。一条方向从 v_i 指向 v_j 的弧记为 (v_i, v_j) 。

常用记号(notation): $p(G)$ 表示 G 的点(顶点)数, $p(D)$ 表示 D 的点(顶点)数, $q(G)$ 表示 G 的边数, $q(D)$ 表示 D 的弧数。

专门名字(nomenclature):考虑无向图 $G=(V,E)$ 。若边 $e=[u,v] \in E$,则称 u, v 是 e 的端点,也称 u, v 是相邻的。称 e 是点 u (与点 v)的关联边。若图 G 中,某个边 e 的两个端点相同,则称 e 是环(loop)。若两个点(顶点)之间有多于一条的边,称这些边为多重边。无环、无多重边的图称为简单图,无环、有多重边的图称为多重图。以点(顶点) v 为端点的边的个数称为 v 的次(度)(degree),环在计算次(度)时算作两次。称次(度)为1的点为悬挂点,悬挂点的关联边称为悬挂边,次(度)为零的点称为孤立点。

定理 1.1.1 图 $G=(V,E)$ 中,所有点(顶点)的次(度)之和是边数的两倍,即

$$\sum_{v \in V} d(v) = 2q$$

次(度)为奇数的点,称为奇点,否则称为偶点。

定理 1.1.2 任一个图中,奇点的个数为偶数。

给定一个图 $G=(V,E)$,一个点、边交错序列 $(v_{i_1}, e_{i_1}, v_{i_2}, e_{i_2}, \dots, v_{i_{k-1}}, e_{i_{k-1}}, v_{i_k})$,如

果满足 $e_i = [v_i, v_{i+1}]$ ($i=1, 2, \dots, k-1$)，则称为一条联结 v_i 和 v_k 的链，记为 $(v_i, v_{i_2}, \dots, v_{i_{k-1}}, v_k)$ ，有时称点(顶点) $v_{i_2}, \dots, v_{i_{k-1}}$ 为链的中间点。

链 $(v_i, v_{i_2}, \dots, v_{i_{k-1}}, v_k)$ 中，若 $v_i = v_k$ ，则称之为一个闭链，记为 $(v_i, v_{i_2}, \dots, v_{i_{k-1}}, v_i)$ 。若链 $(v_i, v_{i_2}, \dots, v_{i_{k-1}}, v_k)$ 中，点(顶点) $v_i, v_{i_2}, \dots, v_{i_{k-1}}, v_k$ 都不相同，则称之为初等链；若闭链 $(v_i, v_{i_2}, \dots, v_{i_{k-1}}, v_i)$ 中， $v_i, v_{i_2}, \dots, v_{i_{k-1}}$ 都不相同，则称之为初等闭链，初等闭链又称之为圈(cycle)。若链(闭链)中含的边都不相同，则称之为简单链(闭链)。

图 G 中，若任何两个顶点之间，至少有一条链，则称 G 是连通图，否则称为不连通图。若 G 是不连通图，它的每一个连通的部分称为 G 的一个连通分图(分图、分支)。

给定一个图 $G=(V, E)$ ，如果图 $G'=(V', E')$ ，使 $V=V'$ 及 $E' \subseteq E$ ，则称 G' 是 G 的一个支撑子图。

设 $v \in V(G)$ ，用 $G-v$ 表示从图 G 中去掉点(顶点) v 及 v 的关联边后得到的一个图。

现讨论有向图。给定一个有向图 $D=(V, A)$ ，从 D 中去掉所有弧上的箭头，就得到一个无向图，称之为 D 的基础图，记为 $G(D)$ 。

给 D 中一条弧 $a=(u, v)$ ，称 u 为 a 的始点， v 为 a 的终点，称弧 a 是从 u 指向 v 的。

设 $(v_i, a_{i_1}, v_{i_2}, a_{i_2}, \dots, v_{i_{k-1}}, a_{i_{k-1}}, v_k)$ 是 D 中的一个点(顶点)弧交错序列，如果这个序列在基础图 $G(D)$ 中所对应的点边序列是一条链，则称这个点弧交错序列是 D 的一条链。类似定义闭链和初等链(闭链)。

如果 $(v_i, a_{i_1}, v_{i_2}, a_{i_2}, \dots, v_{i_{k-1}}, a_{i_{k-1}}, v_k)$ 是 D 中的一条链，并且对 $t=1, 2, \dots, k-1$ ，均有 $a_{i_t}=(v_{i_t}, v_{i_{t+1}})$ ，称之为从 v_i 到 v_k 的一条途径。若途径的第一个点和最后一个点相同，称之为闭途径。类似定义初等途径(闭途径)。初等途径又称之为路，初等闭途径又称之回路。

对无向图，链与途径(圈与回路)这两个概念一致。类似无向图，可定义简单有向图，多重有向图。

权是指与顶点或边(弧)有关的数量指标；根据实际问题需要，可赋予权不同含义，如距离、时间、费用等；可赋一个权，也可赋多个权；权可以是非负数，也可以是负数；赋权越多的图，反映的问题越贴近实际，考虑起来也越复杂。

赋权图(网络)不仅指出各点(顶点)间的邻接关系，同时也表示各点之间的数量关系。赋权图被广泛用于解决工程技术与科学生产管理等领域的最优化问题。

本书中有关图的在此节未提到的其他一些基本概念参见 *Graph Theory with Applications*^[1]；在本书中，我们不加区分地采用 $+\infty$ 或 ∞ 表示正无穷大。

1.2 最短路问题

1.2.1 概念与定义

给定简单赋权连通有向图(网络) $N = (s, t, V; A; W)$, 其中 $V = \{1, 2, \dots, n\}$ 是 N 的顶点集, $A = \{a_{ij} \mid \text{弧 } a_{ij} = (i, j) \text{ 存在}\}$ 是 N 的弧集, $W = (W_{ij})_{n \times n}$ 是 N 的权关联矩阵, W_{ij} 表示弧 $a_{ij} = (i, j)$ 的权(从顶点 i 到顶点 j 的弧 $a_{ij} = (i, j)$ 的长(费用等))。当 $i = j$ 时, 规定 $W_{ij} = 0$; 当 $i \neq j$ 时, 如果弧 (i, j) 不存在, 规定 $W_{ij} = \infty$ 。给定两顶点 s (始点)、 t (终点)。设 P 是从顶点 s 到顶点 t 的一条路, 定义路 P 的权是 P 上所有弧的权之和, 记为 $w(P)$ 。最短路问题就是在所有从顶点 s 到顶点 t 路中, 求一条权最小的路, 即求一条从顶点 s 到顶点 t 的路 P_0 , 使 $w(P_0) = \min_P w(P)$, 式中对 N 中所有从顶点 s 到顶点 t 的路 P 取最小, 称 P_0 是从顶点 s 到顶点 t 的最短路。路 P_0 的权称为从顶点 s 到顶点 t 的距离, 记为 $d(s, t)$ 。

最短路问题, 是重要的最优化问题之一, 可直接用于解决生产实际中的许多问题(如管道铺设、线路安排、厂区布局、设备更新等), 常被作为解决其他问题的基本工具^[2]。

1.2.2 常用数值算法

以下算法 Dijkstra-NA 是基于 Dijkstra 算法给出的求网络 $N = (s, t, V; A; W)$ 中从始点 s 到所有其他顶点最短路的数值算法。

Dijkstra-NA:

Step1 令 $S := \{s\}$, $P(s) := 0$, $\text{prev}(s) := 0$; 对任意顶点 $i \in V \setminus \{s\}$, 令 $T(i) := +\infty$, $\text{prev}(i) := -1$, $k := s$ 。其中 S 表示永久标号顶点的集合, $P(i)$ 表示给顶点 i 赋予的永久标号(代表从始点 s 到顶点 i 的最短距离), $T(i)$ 表示给顶点 i 赋予的临时标号, $\text{prev}(i)$ 表示在从始点 s 到顶点 i 的最短路中顶点 i 的紧前顶点(即前点), k 表示此时进入永久标号顶点集合的顶点。

Step2 如果 $S = V$, 则对任意顶点 $i \in S$ 而言, 从始点 s 到顶点 i 的最短距离 $d(s, i) = P(i)$, 转 Step5; 否则, 转 Step3。

Step3 对任意顶点 $j \in V \setminus S$ 及 $k \in S$, 如果满足 $(k, j) \in A$ 且 $T(j) > P(k) + W_{kj}$, 则令 $T(j) := P(k) + W_{kj}$, $\text{prev}(j) := k$ 。

Step4 令 $T(i) = \min_{j \in V \setminus S} \{T(j)\}$ 。如果 $T(i) < +\infty$, 则令 $P(i) := T(i)$, $S := S \cup \{i\}$, $k := i$, 转 Step2; 否则, 对任意顶点 $i \in S$ 而言从始点 s 到顶点 i 的最短距离

$d(s, i) = P(i)$ ，而对任意顶点 $i \in V \setminus S$ 而言从始点 s 到顶点 i 的最短距离 $d(s, i) = T(i)$ ，转 Step5。

Step5 如果 $t \notin S$ ，则不存在从顶点 s 到顶点 t 的最短路，停止。否则令 $\text{Path}(s, t) := \emptyset$ ， $l := \text{prev}(t)$ ， $r := t$ ，转 Step6。在此， $\text{Path}(s, t)$ 表示由弧构成的从顶点 s 到顶点 t 的最短路， $\emptyset$ 代表空集。

Step6 令 $\text{Path}(s, t) := \text{Path}(s, t) \cup \{(l, r)\}$ 。

Step7 如果 $l = s$ ，则 $\text{Path}(s, t)$ 是从顶点 s 到顶点 t 的最短路，停止；否则，令 $r := l$ ， $l := \text{prev}(r)$ ，转 Step6。

Dijkstra-NA 算法只适用于权都为非负数的情形，而对有些权为负数的情形不适用。Dijkstra-NA 算法是多项式时间算法，在最坏情况下它的时间复杂性为 $O(n^2)$ 。

以下算法 Floyd-NA 是基于 Floyd 算法给出的求网络 $N = (s, t, V; A; W)$ 中任意两顶点间最短路的数值算法。

Floyd-NA:

Step1 令 $U_{ij} := W_{ij}$ ， $S_{ij} := j$ ($i, j = 1, 2, \dots, n$)， $k := 1$ 。其中， U_{ij} 表示从顶点 i 到顶点 j 的最短距离， S_{ij} 表示在从顶点 i 到顶点 j 的最短路中顶点 i 的紧后顶点 (即后点)。

Step2 对任意 $i, j \in \{1, 2, \dots, n\}$ ，如果 $U_{ij} > U_{ik} + U_{kj}$ ，则令 $U_{ij} := U_{ik} + U_{kj}$ ， $S_{ij} := S_{ik}$ ；否则， U_{ij} 和 S_{ij} 保持不变。

Step3 如果 $k = n$ ，则转 Step4；否则，令 $k := k + 1$ ，转 Step2。

Step4 如果 $U_{st} = +\infty$ ，则不存在从顶点 s 到顶点 t 的最短路，停止。否则，令 $\text{Path}(s, t) := \emptyset$ ， $l := s$ ， $r := S_{st}$ ，转 Step5。在此， $\text{Path}(s, t)$ 表示由弧构成的从顶点 s 到顶点 t 的最短路， $\emptyset$ 代表空集。

Step5 令 $\text{Path}(s, t) := \text{Path}(s, t) \cup \{(l, r)\}$ 。

Step6 如果 $r = t$ ，则 $\text{Path}(s, t)$ 是从顶点 s 到顶点 t 的最短路，停止；否则，令 $l := r$ ， $r := S_{rt}$ ，转 Step5。

Floyd-NA 算法仅适用于网络 $N = (s, t, V; A; W)$ 中不存在负费用回路的情形。而负费用回路是指回路上所有弧的权之和为负数的回路。显然，它不仅适用于权都为非负数的情形，对虽然有些权为负数但不存在负费用回路的情形也适用。Floyd-NA 算法也是多项式时间算法，在最坏情况下它的时间复杂性为 $O(n^3)$ 。

1.2.3 应用举例

我们考虑以下设备更新问题。

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的，还

继续使用旧的。若购置新设备,就要支付一定的购置费用;若继续使用旧设备,则需支付一定的维修费用。已知该设备在各年年初的价格为:第 i 年年初购进一台设备的价格为 p_i , $i=1,2,\dots,n$;显然, $p_i \geq 0$ ($i=1,2,\dots,n$),如表 1-1 所示。

表 1-1 购置设备价格

年初	第 1 年	第 2 年	...	第 n 年
价格	p_1	p_2	...	p_n

已知使用不同时间(年)的设备所需的维修费用为:一台设备在被使用的第 $i-1$ 年至第 i 年期间的维修费用为 c_i , $i=1,2,\dots,n$;显然, $c_i \geq 0$ ($i=1,2,\dots,n$),如表 1-2 所示。如何制订一个 n 年之内的设备更新计划,使总支付费最小。

表 1-2 设备维修费用

使用年数	0~1	1~2	...	$(n-1) \sim n$
维修费用	c_1	c_2	...	c_n

我们可以按以下方法将上述设备更新问题转化为网络最短路问题来求解。

构造网络 $N=(s,t,V;A;W)$;始点 $s=1$,终点 $t=n+1$, $V=\{1,2,\dots,i,\dots,n,n+1\}$ 是顶点集,顶点 i 表示第 i 年年初购进一台新设备($i \in \{1,2,\dots,n\}$),顶点 $n+1$ 表示第 n 年年底; $A=\{(i,j)|1 \leq i < j \leq n+1\}$ 是弧集,弧 (i,j) 表示在第 i 年年初购进的设备一直使用到第 j 年年初(即第 $j-1$ 年底); $W=(W_{ij})_{(n+1) \times (n+1)}$ 是权关联矩阵;当 $i=j$ 时,规定 $W_{ij}=0$;当 $i \neq j$ 时,如果弧 (i,j) 不存在,规定 $W_{ij}=\infty$,否则,有 $i < j$,规定 $W_{ij}=p_i + \sum_{k=1}^{j-i} c_k$ 。于是,求(制订)最优设备更新计划等价于在网络 $N=(s,t,V;A;W)$ 中求从顶点 s 到顶点 t 的最短路,可用 Dijkstra-NA 算法或 Floyd-NA 算法来求解。

1.3 结 论

本章给出的两个求网络最短路数值算法都是多项式时间算法,可作为有效工具用于解决其他复杂的相关优化问题(比如,用于解决第4章论述的网络最小费用最大流问题),因此具有重要的理论与应用价值。

第2章 网络最大容量路问题

网络最大容量路问题在《网络最优化》^[3]一书中已经提出,笔者通过改进建立^[4,5]了在运输网络中求最大容量路的两个高效数值算法(它们是强多项式时间算法,且易于在计算机上编程实现)。因为这两个算法在高效寻求网络最大流中发挥十分重要的作用^[6,7],因此本章主要介绍我们取得的这一研究结果。

2.1 基本概念与理论

定义 2.1.1 把简单赋权连通有向图 $N = (V, s, t; A; C)$ 称为运输网络, 其中 $V = \{1, 2, \dots, n\}$ 为 N 中顶点的集合, 顶点 s 是 N 的源(发点), 顶点 t 是 N 的沟(收点), $A = \{a_{ij} \mid \text{弧 } a_{ij} = (i, j) \text{ 存在}\}$ 为 N 中弧的集合, $C = (C_{ij})_{n \times n}$ 为 N 的容量矩阵(C_{ij} 表示从顶点 i 到顶点 j 的直接运输通过能力)。当 $i = j$ 时, 规定 $C_{ij} = +\infty$ 。当 $i \neq j$ 时, 如果不存在弧 $a_{ij} = (i, j)$, 规定 $C_{ij} = 0$; 否则规定 $C_{ij} = C(a_{ij}) > 0$, 其中 $C(a_{ij})$ 表示弧 a_{ij} 的容量。

定义 2.1.2 给定运输网络 $N = (V, s, t; A; C)$ 。 N 中的一条路 P 的容量 $C(P)$ 就是 P 的所有弧的最小容量(即 $C(P) = \min_{(i,j) \in P} \{C_{ij}\}$)。 N 中的一条途径 Q 的容量 $C(Q)$ 就是 Q 的所有弧的最小容量(即 $C(Q) = \min_{(i,j) \in Q} \{C_{ij}\}$)。在 N 中从顶点 i 到顶点 j ($i \neq j$) 的所有路中, 容量最大的路称为 N 的从顶点 i 到顶点 j 的最大容量路。

引理 2.1.1 设 Q 为运输网络 $N = (V, s, t; A; C)$ 的一条途径, P 为 Q 所包含的路, 则 $C(Q) \leq C(P)$ 。

引理 2.1.2 在运输网络 $N = (V, s, t; A; C)$ 中, 从发点 s 到所有其他顶点的最大容量路的容量按从大到小顺序排列依次为 $u_1 \geq u_2 \geq u_3 \geq \dots \geq u_{n-1} > 0$ 。并假定: ① $P_1, P_2, \dots, P_{n-1}$ 分别代表对应于 $u_1, u_2, \dots, u_{n-1}$ 的从发点 s 到其他顶点的最大容量路, 即 $u_i = C(P_i)$, 且所有的 P_i ($i = 1, 2, \dots, n-1$) 以 s 为始点, 它们的终点互不相同; ② P_i ($i = 1, 2, \dots, n-1$) 的长(即 P_i 包含弧的条数) $L(P_i)$ 尽可能少; ③ 若 $C(P_i) = C(P_j)$, $P_i \neq P_j$, $L(P_i) \leq L(P_j)$, 则 $i < j$ 。在上述假设条件下, 则 P_i 只有一条弧。

定理 2.1.1 给定运输网络 $N = (V, s, t; A; C)$, 在引理 2.1.2 的假设条件下, 记 $S_0 = \{s\}$, $T_0 = V - S_0$, 则有以下两个结论成立: ① $u_1 = \max_{j \in T_0} \{C_{sj}\}$, 取达到上述最大值的顶点为 j_1 , 则 $P_1 = P(s, j_1)$ (其中, $P(s, j_1)$ 表示以 s 为始点 j_1 为终点的路, 由于

它只有一条弧, 因此它只包含 s 为始点 j_1 为终点的弧); ②假定 $u_1, u_2, \dots, u_k$ 已求出, 对应的最大容量路依次为 $P_1 = P(s, j_1), P_2 = P(s, j_2), \dots, P_k = P(s, j_k)$, 其中 $P(s, j_i)$ 表示以 s 为始点 j_i 为终点的路 ($i=1, 2, \dots, k$), 并记 $S_k = \{s, j_1, j_2, \dots, j_k\}$, $T_k = V - S_k$, 则 $u_{k+1} = \max_{j_i \in S_k, j' \in T_k} \{\min\{u_i, C_{jj'}\}\}$, 达到上述最大值的 j' 可取为 j_{k+1} , $P_{k+1} = P(s, j_{k+1})$ 。

证明 (1)利用引理 2.1.2, 结论显然成立。

(2)设 P_{k+1} 的终点为 j' , 显然 $j' \in T_k$ 。因为 P_{k+1} 至少由一条弧, 所以可设 P_{k+1} 的最后一条弧为 (j_i, j') 。于是可将 P_{k+1} 看作从顶点 s 到顶点 j_i 的路 $P'(s, j_i)$ 加上弧 (j_i, j') 组成。由引理 2.1.1 知, $C(P'(s, j_i)) \geq C(P_{k+1})$ 。由定义 2.1.2 知, $u_i \geq C(P'(s, j_i))$ 。所以 $u_i \geq u_{k+1}$, $i < k+1$ 。从而 $u_{k+1} \leq \max_{j_i \in S_k, j' \in T_k} \{\min\{u_i, C_{jj'}\}\}$ 。具有上式右端形式的值都代表从顶点 s 到 T_k 中的某顶点 j' 的一条途径 Q 的容量 $C(Q)$ 。设途径 Q 包含的从顶点 s 到 j' 的路为 P , 由引理 2.1.1 知 $C(Q) \leq C(P)$ 。根据 u_{k+1} 的排列顺序以及定义 2.1.2 知 $u_{k+1} = C(P_{k+1}) \geq C(P)$ 。所以, $u_{k+1} \geq \max_{j_i \in S_k, j' \in T_k} \{\min\{u_i, C_{jj'}\}\}$ 。于是, 我们有 $u_{k+1} = \max_{j_i \in S_k, j' \in T_k} \{\min\{u_i, C_{jj'}\}\}$ 。故结论成立。

引理 2.1.3 用 $N(i, j, m)$ 表示由顶点集合 $\{i\} \cup \{j\} \cup \{1, 2, \dots, m\}$ 生成的运输网络 $N = (V, s, t; A; C)$ 的子网络 ($1 \leq i, j \leq n$; $m \leq n$); 特别地, 用 $N(i, j, 0)$ 表示由顶点集合 $\{i\} \cup \{j\}$ 生成的运输网络 $N = (V, s, t; A; C)$ 的子网络 ($1 \leq i, j \leq n$)。用 $U_{ij}(m)$ 表示 $N(i, j, m)$ 中从顶点 i 到顶点 j 的最大容量路的容量 ($1 \leq i, j \leq n$; $m \leq n$); 特别地, 用 $U_{ij}(0)$ 表示 $N(i, j, 0)$ 中从顶点 i 到顶点 j 的最大容量路的容量 ($1 \leq i, j \leq n$)。在上述条件下, 于是有以下结论成立: ① $U_{ij}(0) = C_{ij}$; ② $N(i, j, n) = N$, $U_{ij}(n) = U_{ij}$, 其中 U_{ij} 表示运输网络 N 中从顶点 i 到顶点 j 的最大容量路的容量; ③ $N(i, j, m-1)$ 是 $N(i, j, m)$ 的子网络, $U_{ij}(m-1) \leq U_{ij}(m)$; ④ $N(i, m, m)$ 和 $N(i, m, m-1)$ 是相同网络, $U_{im}(m) = U_{im}(m-1)$ 。

定理 2.1.2 给定运输网络 $N = (V, s, t; A; C)$, 在引理 2.1.3 的前提条件下, 则有以下两个结论成立: ① $U_{ij}(0) = C_{ij}$ ($1 \leq i, j \leq n$); ② $U_{ij}(m) = \max\{U_{ij}(m-1), \min\{U_{im}(m-1), U_{mj}(m-1)\}\}$ ($1 \leq i, j \leq n$; $1 \leq m \leq n$)。

证明 (1)由引理 2.1.3 知, $U_{ij}(0) = C_{ij}$ ($1 \leq i, j \leq n$)。

(2)将 $N(i, j, m)$ 中所有从顶点 i 到顶点 j 的路分成两类: 第一类不通过顶点 m , 第二类通过顶点 m 。下面对这两类分别加以讨论。

(a)网络 $N(i, j, m)$ 中从顶点 i 到顶点 j 不通过顶点 m 的路全体, 恰好是 $N(i, j, m-1)$ 中从顶点 i 到顶点 j 的路的全体。因此, 第一类路中的最大容量路容量应该是 $U_{ij}(m-1)$ 。

(b)如果在 $N(i, j, m)$ 中存在以顶点 i 为始点以顶点 j 为终点而且通过顶点 m 的