经典原版书库

网络流

理论、算法与应用

（英文版）

本书只供在
中国大陆销售

拉文德拉 K. 阿胡亚
托马斯 L. 马南提 著
詹姆斯 B. 沃林

机械工业出版社
China Machine Press

经典原版书库

网络流

理论、算法与应用

(英文版)

拉文德拉 K. 阿胡亚
托马斯 L. 马南提 著
詹姆斯 B. 沃林

机械工业出版社
China Machine Press

专家指导委员会

（按姓氏笔画顺序）

前　言

如果你不想在死后被人迅速遗忘，要么著书立说，
要么做一番值得大书特书的功业。
——本杰明·富兰克林

网络流是一个令人激动的研究领域，它集众多学生、专业人士和研究人员所最为钟爱的数学和计算科学知识于一身。它还融合了深奥的学术内容与广泛的适用性，涉及宽泛的应用领域，比如化学和物理、计算机网络、工程学的大多数分支、制造、公共政策、调度和路由、电信、运输等等。网络流是一个传统的学科，19世纪古斯塔夫·基尔霍夫及其他杰出的物理学家就开始了相关研究，但同时它又充满活力，许多新的研究成果和新方法不断涌现。其传统植根于机械制造学、工程学和应用数学等传统学科，以及计算机科学和运筹学等现代领域。

在本书中，我们试图介绍以上的各个方面，从而满足作为一本网络流综合教材兼顾传统知识和现代技术的需要，并提供有关理论、算法及应用的综合见解。经过我们的悉心安排，本书可以作为高年级本科生和研究生的导论性教材或进阶教材，也可作为研究人员和专业人士的参考书。此外，我们也力争让本书尽可能地可读、容易理解并具有一定深度，以方便那些在计算机科学和优化方面背景知识有限的读者。

本书具有如下特色：

- 深入而全面地论述最短路径、最大流及最小代价流问题，包括对用于这些核心模型的独特的多项式-时间算法的描述。
- 强调功能强大的算法策略和分析工具，比如数据缩放、几何改进变元以及势函数自变量。
- 通俗易懂地介绍若干重要数据结构，包括*d*-堆、斐波那契堆以及动态树。
- 阐述其他重要的网络优化主题以及切实可行的求解技术，如拉格朗日松弛法。
- 通过一系列的应用来介绍各种新主题，并且有完整的一章来介绍应用。
- 用专门的一章来介绍算法的经验性测试。
- 包括150多个网络流应用，涉及工程学、管理学及科学领域的各个方面。
- 包括800多道难度不一的习题，许多习题扩展了书中的内容。
- 包括大约400幅插图，用以说明书中的材料。
- 提供详尽的参考注释以及大量的历史背景和文献指南。

如上所述，本书只对一些核心主题进行了详尽介绍，而不打算细述所有的方面。此外，尽管本书主要关注于高效算法的设计和分析问题，但也着重强调了应用方面。

为了将本书的材料组织成一个整体并使之更具可读性，我们花费了大量的时间，而且还使用若干研究案例来增强本书的表现形式。因此，本书对一些主题的介绍有别于当前的文献中的讨论。我们希望这种方式不会引起混淆，并且事实上这有助于促进读者对相关材料的理解，使得在文献中看起来不太相关的主题之间的关联性更为明显。

致教师和学生

本书适用于具有各种背景知识的学生。不过，学生应当具备一定的数学知识（例如，对基本的证明方法的掌握），并有基本的计算机程序设计经验，但不必是数学、计算机科学或优化方面的专家。当然，若具有这些领域的基础知识，本书学起来会更加轻松。在第3章及附录A、B和C中，我们提供了一些相关的背景材料。

本书涉及的内容较多，一学期的教学时间恐难以完成。第1~3章、第19章、第5~12章、第16章、第17章的特定内容以及第13~15章、第18章的部分内容，可以作为网络流和网络优化课程的教学内容。因为每一章的内容都是模块化的，教师可以使用每一章中前几节的基础性材料，并根据需要选用相关的补充材料。对于讲授算法的高级课程，应关注于第4~12章，并且应该学习这些章节中的全部内容。

在讲授书中的算法时，我们认为，重要的是要理解算法设计和分析的基础方法及特定的结果。因此建议老师和学生在使用本书的过程中，尽可能多地回顾第3章中对算法设计和分析的讨论。

本书中讨论的许多主题都涉及到特定结构的线性程序。因此，我们本应该在介绍大多数内容时采用线性编程方法。然而，除了第17章及第15、16章的部分内容外，本书几乎自始至终都在借用网络或图形的方式来陈述观点。我们相信使用这种方法（虽然有时候使用线性编程方法会显得更有条理）会带来几个好处：首先，相关的材料容易被更多的读者所接受。其次，这种方法允许那些相关知识有限的学生使用简单的几何和代数描述，并应用具体的设置来学习线性编程方法的各种思想；同时，熟悉线性编程的学生也可以从另外一个角度来强化他们对书中材料的理解。事实上，若读者具备线性编程的背景知识，我们鼓励老师和学生将书中的内容和线性编程的一般结果明确地对应起来。

虽然本书提供了一些数值计算习题，用于测试大家对书中内容的理解，但许多习题还是专门围绕着某些应用和理论的。老师可以使用这些材料（并适当引申）作为讲课素材，也可以自行修改书中的习题以满足教学所需。

致普通读者

对于应用数学、计算机科学、工程学、管理科学/运筹学领域的专业人士以及各个应用领域的从业人员而言，可以按需从书中汲取特定的信息，而不必通读全书。本书主要按模型类型（如最短路径或生成树问题）来组织内容，读者可以沿着这些主线来进行学习。至于相关的应用，可以参考本书19.10节中的一系列表格，其中总结了书中介绍的各种网络流应用。每章末的参考注释也提到了许多书中没有讨论或是只在习题中出现的应用。此外，书中对文献的引用是有所取舍的（与应用相关的文献例外）。第1章末提到了更详细的文献引用信息。

本书使用伪代码来描述大多数算法，这样可以使那些对计算机程序设计不甚了解的读者更易理解这些复杂的算法。借用伪代码，我们能够以更加通用的形式来描述算法，但若要使用算法，则必须将这些伪代码转换成特定的编程语言，并添加诸如输入/输出和错误处理过程之类的相关实现代码。

信息反馈

对这样一本篇幅巨大且内容有深度的图书而言，错误是在所难免的。在写作此书之时，我们采纳了许多人的意见。我们非常乐于接受大家对本书的任何评论意见，包括指正错误。请将反馈信息发往如下地址：

Professor James B. Orlin
Sloan School of Management，MIT
Cambridge，MA 02139，USA
e-mail: jorlin@eagle.mit.edu
fax: 617-258-7579

致谢

许多人对本书作出了贡献，他们之中有的帮助我们加深了对网络流的理解，有的对本书的内容安排给出了有益的建议，有的对本书的初稿提出了建设性的反馈意见。

我们由衷地感谢如下人士。首先感谢E. Dinic、Jack Edmonds、Hal Gabow、Fred Glover、Matsao Iri、Bruce Golden、Richard Karp、A. Karzanov、Darwin Klingman、Eugene Lawler、Robert Tarjan、Eva Tardos、Richard Wong等人，感谢他们的直接协作或论著，由此我们学到了许多关于网络和网络优化的知识；还有George Dantzig、Lester Ford和Delbert Fulkerson等先驱者所做的努力，他们定义了我们现在所知道的网络流领域。我们希望本书能够真正反映出上述学者的研究理念。为了更好地设定本书的写作框架，我们分发了调查表以征求意见，许多同事（由于人太多，在此就不一一列出了）对此作出了积极响应。在提炼本书的整体架构方面，他们的意见起到了非常大的作用。Anant Balakrishnan和Janny Leung阅读并注释了部分手稿；S. K. Gupta、Leslie Hall、Prakash Mirchandani以及Steffano Pallottino均详细注释了大段的手稿。在此要特别感谢Steffano Pallottino，感谢他对手稿的细心审查并作了大量的纠正。Bill Cunningham对本书最初的样章给出了详细的建议。上述这些人的每一项建议都极大地提升了本书的质量和价值。数年来，我们的许多博士生也帮助测试和改进了我们的想法。其中一些学生（包括Murali Kodialam、Yusin Lee、Tim Magee、S. Raghavan、Rina Schneur及Jim Walton）帮助设计了许多习题。这些学生和许多其他的学生一同提出了手稿中的错误并提出建设性意见。从这一点上说，我们要尤其感谢MIT 1991年网络优化专业春季班的学生们。我们还要感谢Prentice Hall出版公司介绍的来自普林斯顿大学的审校者Leslie Hall。Charu Aggarwal和Ajay Mishra在调整本书版式、校对全书内容及抽取索引方面做了大量的工作，在此向他们表示诚挚的谢意。

Ghanshyam Hoshing、Karen Martel以及Laura Terrell在输入本书的手稿方面均做了大量出色的工作，尤其是Ghanshyam Hoshing输入、绘制并编排了书中的大部分文本和图片。

在此一并感谢印度理工学院坎普尔分校工业与管理工程系以及麻省理工学院斯隆管理学院和运筹学中心，他们为我们提供了一个良好的研究网络流的环境，并促成了本书的面世。还要感谢美国国家科学基金会、美国海军研究局、美国交通部和GTE实验室，没有他们对我们的研究工作的支持，就没有本书。

我们还要感谢我们的父母，他们是Kailash和Ganesh Das Ahuja、Florence和Lee Magnanti、Roslyn和Albert Orlin，感谢他们的关爱和鼓励，是他们培养了我们的求知欲望。最后，我们要向我们的妻子（Smita Ahuja、Beverly Magnanti、Donna Orlin）和孩子（Saumya和Shaman Ahuja；Randy Magnanti；Jenna、Ben和Caroline Orlin）表示内心的谢意，感谢他们在我们写作本书时对我们表现出的爱和理解。在写作本书之初，我们低估了这项浩大的工程，我们的家庭成员也不曾料到我们会为此夺走了他们如此多的时间和精力。因此，这部作品不仅属于我们，也属于他们！

拉文德拉 K. 阿胡亚
托马斯 L. 马南提
詹姆斯 B. 沃林

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国

家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzedu@hzbook.com
联系电话：(010) 68995264
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

CONTENTS

1

INTRODUCTION

Begin at the beginning . . . and go on till you come to the end:
then stop.
—Lewis Carroll

Chapter Outline

1.1 INTRODUCTION

Everywhere we look in our daily lives, networks are apparent. Electrical and power networks bring lighting and entertainment into our homes. Telephone networks permit us to communicate with each other almost effortlessly within our local communities and across regional and international borders. National highway systems, rail networks, and airline service networks provide us with the means to cross great geographical distances to accomplish our work, to see our loved ones, and to visit new places and enjoy new experiences. Manufacturing and distribution networks give us access to life's essential foodstock and to consumer products. And computer networks, such as airline reservation systems, have changed the way we share information and conduct our business and personal lives.

In all of these problem domains, and in many more, we wish to move some entity (electricity, a consumer product, a person or a vehicle, a message) from one point to another in an underlying network, and to do so as efficiently as possible, both to provide good service to the users of the network and to use the underlying (and typically expensive) transmission facilities effectively. In the most general sense, this objective is what this book is all about. We want to learn how to model application settings as mathematical objects known as network flow problems and to study various ways (algorithms) to solve the resulting models.

Network flows is a problem domain that lies at the cusp between several fields of inquiry, including applied mathematics, computer science, engineering, management, and operations research. The field has a rich and long tradition, tracing its roots back to the work of Gustav Kirchhof and other early pioneers of electrical engineering and mechanics who first systematically analyzed electrical circuits. This early work set the foundations of many of the key ideas of network flow theory and established networks (graphs) as useful mathematical objects for representing many physical systems. Much of this early work was descriptive in nature, answering such questions as: If we apply a set of voltages to a given network, what will be the resulting current flow? The set of questions that we address in this book are a bit different: If we have alternative ways to use a network (i.e., send flow), which alternative will be most cost-effective? Our intellectual heritage for answering such questions is much more recent and can be traced to the late 1940s and early 1950s when the research and practitioner communities simultaneously developed optimization as an independent field of inquiry and launched the computer revolution, leading to the powerful instruments we know today for performing scientific and managerial computations.

For the most part, in this book we wish to address the following basic questions:

1. *Shortest path problem.* What is the best way to traverse a network to get from one point to another as cheaply as possible?
2. *Maximum flow problem.* If a network has capacities on arc flows, how can we send as much flow as possible between two points in the network while honoring the arc flow capacities?
3. *Minimum cost flow problem.* If we incur a cost per unit flow on a network with arc capacities and we need to send units of a good that reside at one or more points in the network to one or more other points, how can we send the material at minimum possible cost?

In the sense of traditional applied and pure mathematics, each of these problems is trivial to solve. It is not very difficult (but not at all obvious for the later two problems) to see that we need only consider a finite number of alternatives for each problem. So a traditional mathematician might say that the problems are well solved: Simply enumerate the set of possible solutions and choose the one that is best. Unfortunately, this approach is far from pragmatic, since the number of possible alternatives can be very large—more than the number of atoms in the universe for many practical problems! So instead, we would like to devise algorithms that are in a sense "good," that is, whose computation time is small, or at least reasonable, for problems met in practice. One way to ensure this objective is to devise algorithms whose running time is guaranteed not to grow very fast as the underlying network becomes larger (the computer science, operations research, and applied mathematics communities refer to the development of algorithms with such performance guarantees as *worst-case analysis*). Developing algorithms that are good in this sense is another major theme throughout this book, and our development builds heavily on the theory of computational complexity that began to develop within computer science, applied mathematics, and operations research circles in the 1970s, and has flourished ever since.

The field of computational complexity theory combines both craftsmanship and theory; it builds on a confluence of mathematical insight, creative algorithm design, and the careful, and often very clever use of data structures to devise solution methods that are provably good in the sense that we have just mentioned. In the field of network flows, researchers devised the first, seminal contributions of this nature in the 1950s before the field of computational complexity theory even existed as a separate discipline as we know it today. And throughout the last three decades,