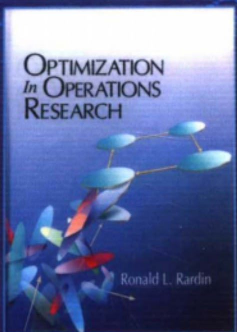
英文版

PEARSON
Prentice Hall

运筹学

——优化模型与算法



Optimization in Operations Research

[美] Ronald L. Rardin 著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

英文版

运筹学

——优化模型与算法

本书改变了运筹学引论教学的基本方式，展示了运筹学的宽广范畴，强化了对基本原理的掌握，激发了学生们对此领域的学习热情，提供了与实践密切联系的建模工具，开发了运用工具所需的必要技巧。

本书是一本适应当今运筹学发展趋势的优秀的综合性入门教材，主要特点是重视建模和算法的结合，引入了相关的建模工具以及用其进行模型开发的基本技巧。全书共分14章，前3章介绍数学模型的问题求解和改进搜索的基本概念与原理，其余内容则覆盖了确定型优化领域的几乎全部内容，除了传统的线性规划的模型、算法、对偶理论和灵敏度分析等内容以外，还包括了网络流、整数／组合优化、非线性规划和目标规划等领域的基本模型和主要算法。此外，本书还包含了遗传算法、模拟退火、禁忌搜索和分支切割算法等前沿内容。全书采用统一的理论框架，以简单的“改进搜索”思路贯穿始终，全面且循序渐进地演绎了各种优化算法和方法，包括传统的单纯形法、牛顿法、网络流算法以及各种启发式算法，使读者感受到每次引入的新算法都建立在以往算法的基础上，直观且逻辑性强，易于理解。本书收录了丰富的实际案例，并有大量上机习题，便于理论结合实践。

Ronald L. Rardin

美国数学规划和优化理论及其应用运筹学方面的著名学者。于1974年从佐治亚理工学院获得博士学位，长期任普度大学工业工程系教授、普度大学能源建模研究组（PEMRG）主任和Regenstrief医疗保健工程研究中心（RCHE）主任，还曾担任美国国家自然科学基金会运筹学和服务企业项目主任。Rardin教授的教学和研究重点是大规模优化的建模与算法，包括在医疗保健系统、交通与物流系统以及能源规划方面的应用。他曾四次荣获普度大学在工业工程方面的Pritsker杰出教学奖，是美国工业工程学会、运筹学与管理科学学会以及数学规划学会的会员。Rardin教授现已加入阿肯色大学。

PEARSON Education

PEARSON Prentice Hall

责任编辑：马　岚
封面设计：喻　晓

ISBN 978-7-121-04925-5

定价：89.00元

运 筹 学

——优化模型与算法

（英文版）

Optimization in Operations Research

［美］ Ronald L. Rardin 著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书是一本适应当今运筹学发展趋势的优秀的综合性入门教材，主要特点是重视建模和算法的结合，引入了相关的建模工具以及用其进行模型开发的基本技巧。全书共分14章，前3章介绍数学模型的问题求解和改进搜索的基本概念与原理，其余内容则覆盖了确定型优化领域的几乎全部内容，除了传统的线性规划的模型、算法、对偶理论和灵敏度分析等内容以外，还包括了网络流、整数/组合优化、非线性规划和目标规划等领域的基本模型和主要算法。此外，本书还包含了遗传算法、模拟退火、禁忌搜索和分支切割算法等前沿内容。全书采用统一的理论框架，以简单的“改进搜索”思路贯穿始终，全面且循序渐进地演绎了各种优化算法和方法，包括传统的单纯形法、牛顿法、网络流算法以及各种启发式算法，使读者感受到每次引入的新算法都建立在以往算法的基础上，直观且逻辑性强，易于理解。本书收录了丰富的实际案例，并有大量上机习题，便于理论结合实践。

本书适用于工程和数学专业本科生或研究生的运筹学引论课程，也可以作为管理专业本科生或研究生的选修课教材，以及诸如线性规划、整数规划和组合优化、网络流和非线性规划等子领域的入门课程的教材。

版权贸易合同登记号　图字：01-2007-3855

图书在版编目（CIP）数据

运筹学：优化模型与算法：英文/（美）拉丁（Rardin, R. L.）著. –北京：电子工业出版社，2007.9

ISBN 978-7-121-04925-5

Ⅰ. 运…　Ⅱ. 拉…　Ⅲ. 运筹学–英文　Ⅳ. 022

中国版本图书馆CIP数据核字（2007）第133089号

责任编辑：马　岚

印　　刷：
装　　订：北京牛山世兴印刷厂

出版发行：电子工业出版社

北京市海淀区万寿路173信箱　邮编：100036

开　　本：787 × 980　1/16　　印张：58.5　　字数：1310千字

印　　次：2007年9月第1次印刷

定　　价：89.00元

凡所购买电子工业出版社的图书有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

About the Author

Ronald L. Rardin is professor of Industrial Engineering at Purdue University in West Lafayette, Indiana. Born and raised in Kansas, he received his B.A. and M.P.A. degrees from the University of Kansas. After working in city government, consulting and distribution for five years, he continued his education, earning a Ph.D. in Industrial Engineering from the Georgia Institute of Technology. He remained at Georgia Tech as a faculty member for nine years. Professor Rardin joined the faculty of Purdue University in 1982. His teaching and research interests center on optimization modeling and algorithms, particularly for large-scale integer and combinatorial problems. He is co-author of numerous research papers in that field and, with R. Gary Parker, author of a comprehensive graduate text *Discrete Optimization.* Professor Rardin is a recipient of Purdue University's Pritsker award for outstanding undergraduate teaching in industrial engineering. He is a member of IIE, INFORMS, and the Mathematical Programming Society.

推荐者序

在当今世界一流大学的教学实践中，对于运筹学入门课程的一个普遍的共识是：这门课程不仅是介绍如何对各类已知结构的优化问题进行求解的应用数学课程，而且是传授如何对实际问题进行数学建模的工程课程；随着时间的发展，建模和算法相结合的重要性显得越来越突出。Rardin教授的*Optimization in Operations Research*就是一本综合性的、兼顾建模和求解算法两方面的优秀的运筹学入门教材。该书内容丰富，语言生动平实，广受赞誉，曾获1998年美国工业工程师学会年度最佳著作奖（IIE Book of the Year Award），并迅速成为美国多所著名大学的工业工程系、运筹学系、计算机系以及应用数学系所开设的优化方法或数学规划类本科生课程的标准教材。

下面具体列出本书的一些特色。

第一，内容丰富全面。全书共分14章，前3章介绍了数学模型的问题求解和改进搜索的基本概念和原理，第4章至第14章则完整涵盖了运筹学中确定型优化领域的几乎全部内容，除了传统的线性规划的模型、算法、对偶理论和灵敏度分析等内容外，还包括了网络流、整数/组合优化、非线性规划和目标规划等领域的基本模型和主要算法。此外，本书还包含了介绍遗传算法、模拟退火、禁忌搜索和分支切割算法等前沿内容的章节。

第二，全书采用了统一的理论框架。作者用简单的“改进搜索”（Improving Search）的思路贯穿全书，全面且循序渐进地演绎了各种优化算法和方法，包括传统的单纯形法、牛顿法、网络流算法以及各种启发式算法。这种方式使读者能够感受到，每个新算法的引入都建立在以往算法的基础上，直观且逻辑性强，易于理解，非常受学生的欢迎。

第三，语言生动，平实易懂。作者在写作上独具匠心，尽可能采用易于理解的直观的描述性文字，结合大量图表，讲解由浅入深，不牵涉复杂的数学理论和技巧，很容易帮助读者建立优化的基本认识，非常适合初级或中级读者。作者在书中许多地方用一两页篇幅的“知识精要”（prime）的形式，精炼地介绍了与当前知识点相关但又超出本书主题的其他领域知识，例如矩阵理论和实分析基础等，免去了读者频繁参考其他教材的不便。

第四，本书收录了丰富的实际案例。作者以多年教学研究的经验，在书中提供了60多个实际案例，这些实际案例一方面提供了详实的应用背景，一方面展示了优化建模的各种方法和技巧，有助于激发学生的学习兴趣和热情。

第五，本书安排了大量的上机习题。作者非常重视建模和算法相结合的教学，引入了相关的建模工具和使用该工具进行模型开发的基本技巧。每一章后面都附带了很多上机练习，以帮助读者迅速理解和应用各种建模技术和优化算法，使理论和实践很好地结合起来。全书共有300多个源自实践的练习和其他500多道习题，并在书后提供了近半数的计算题和小型描述题的答

案；特别需要指出的是，这些习题都可以通过使用可免费下载的主流建模工具软件GAMS试用版来完成。此外，专门为教师提供的教师手册中提供了全部习题的答案[①]，因此无论对学生还是对教师，这本书的教学价值都十分珍贵。

总之，本书是一本内容全面且别具特色的优秀运筹学入门教材，同时也可作为运筹学研究人员进行建模和算法应用的参考书。

本人于2004年至2005年期间在普渡大学工业工程系做访问学者，我接触的该系教授和研究生不约而同对Rardin教授的教学以及教材的高度评价给我留下了深刻的印象，由此产生了将此优秀教材及其蕴含的先进教学思想引入国内运筹学教学实践的想法。在此特别感谢普渡大学工业工程系Ronald Rardin教授本人以及Mark Lawley教授、Jean-Philippe Richard教授、系主任Dennis Engi教授、Gavriel Salvendy教授、Bo Zeng博士、Jianhong Qiao博士和Lingfeng Ma博士的帮助。

在本书英文影印版即将出版之际，本人获悉，Rardin教授已于2007年1月正式加入了阿肯色大学（http://www.ineg.uark.edu/1520.php）。

程　朋
清华大学自动化系

① 详见书末所附“教学支持说明”——编者注。

前　　言

随着20世纪60年代一系列开创性的入门教材的出版，运筹学真正成为了一门学科。此后的30多年里，这个领域发生了大量事情，但一般的入门课程设置和新出版的教材在很大程度上仍然沿着那些开创性教材所遵循的思路。

传统的教材编写思路有许多优点，但我之所以选定在运筹学的确定型部分（数学规划）着手撰写这本教材，是因为我觉得学生们还需要学习更多的东西。教材的重点应该放在对技能和直觉的传授上，学生们因此能把它们带到实际工作与后续课程中去。学生们理应对当今运筹学工具的多才多艺和普遍适用能力而感到兴奋不已。

本书的主要创新之处是努力实现上述想法。书中的大量算法（包括单纯形法）都是围绕一个共同的“改进搜索”的范例来开发的，这个范例是为学生们无论在课内还是课外都能顺利地不断吸收新的材料而准备的。为强调潜在应用的丰富性和多样性，本书从一开始就涉及了全部数学规划领域内的内容（包括线性规划、整数规划、非线性规划和多目标规划）。前3章和其他章节的很大部分内容，都专门用来表达和描述实例，这些实例大多来源于实际应用的报告。全书在内容组织上采用这样的风格：使查找和翻阅材料尽可能容易，从而方便教师和学生以多种配置方式来使用本书。

本书是为工程和数学专业的本科生和研究生的第一门运筹学课程而设计的，但也可以作为管理专业本科生/研究生的选修课教材，以及诸如线性规划、整数规划和组合优化、网络流、非线性规划等子领域的入门课程的教材。此外，本书全面且循序渐进的内容覆盖，使得它对工程师和研究人员来说也是一本很有价值的参考书。

我对如何讲授数学规划的重新思考起始于这样的信念：如果仅限于讲授线性规划和单纯形法，那么对这个课程来说是不公平的。首先，在工程和管理实践中遇到的绝大多数优化问题无法直接建模为线性规划，它们往往引入了不可忽略的离散、非线性和多目标等因素。也许更重要的是，运筹学的实践已经发展到这样一个阶段，即每天都要处理各种整数、非线性和多目标优化模型。在为撰写本书而研究运筹学应用文献的过程中，我很惊讶地发现离散优化应用的数量明显超过了仅使用线性规划应用的数量，而且到处可见目标规划的实例。如果学生们除了经典的线性规划之外没有见过任何其他优化问题，他们必然会错误地定论：运筹学领域太狭窄了，无法处理实际规划和设计任务的复杂性。

过多关注单纯形法也会带来一个更微妙的学习上的障碍。对于单纯形法，无论是从为找到合适的基集而进行组合搜索的角度讲授，还是从处理线性方程组的角度讲授，几十年来对单纯形法的不断改进几乎完全模糊了它与其他数学规划工具之间的关系。如今一个仅学过单纯形法的学生很难产生对优化原理的深入见解，更难以据此建立对更宽广领域的理解。对他来说，稍

新一些的线性规划的内点算法好像是完全不相干的。

我并非建议所有引论课程都要深入探究整数和非线性规划技术，也不打算剔除线性规划和单纯形法的内容，而是认为需要的是一种发展的认识——把线性规划当成目前解决得最好的但远不是唯一的数学规划的形式；教给学生们直觉和范例，使他们在今后的课程和实践中遇到更复杂的问题时，能够很好地理解。当需要花费时间复习先修知识以深入到更广的问题中时，相关的材料就应该已准备好，这些材料应尽可能围绕与基本线性规划相同的概念而设计。

本书从一开始就通过讨论不同类型的案例，来努力避免把线性规划和优化视为等同的概念。正如一般在概论中所提到的，首要的工作是把优化问题描述为标准的数学规划形式。然而，在引入线性规划的同时，引入整数、非线性和多目标的情形，将激发学生的兴奋点，因为他们会看到如此大量的应用至少可以建模为这种标准形式。他们也将很早就学会识别不同的模型类型，并敏感于处理这些模型的相对难易程度。我的经验是，初学者往往会感到下标、决策变量、约束和目标函数等难以处理（正如当我们仅讨论线性规划时他们所做的那样），但是更广阔范围的数学描述形式实际上就存在于他们先前的训练之中。二进制变量往往是唯一的新的数学概念，学生们发现它们是相当直观的。

对改进搜索（爬山法）范例的讨论开始于第3章。我们首先定义了一些基本概念：改进和可行方向、步长、局部和全局解以及人工启动等。通过自由使用非线性例子，可以用一种直观和几何的方式来处理改进搜索范例。然后，在第5章中把单纯形法看成是一种优美而特殊的线性规划方法进行了讨论。对单纯形法用改进搜索范例加以处理，甚至许多教师也不熟悉这种方式，但实际上并没有多大变化。在牢固建立了爬山法范例的基础上，在第6章中引入内点法，在第10章中引入网络算法，在第13章和第14章中引入非线性规划算法，甚至于在第12章中引入整数规划程序，都变得容易多了。任何一门单独的课程都没有足够的时间涵盖所有这些内容，但是这个范例为学生将来的学习做好了准备，无论这些内容出现在当前课程或后续课程中，或者出现在职业实践中。

如果学生们除了那些预先设计的小到足以手算的例子外，从未见过优化方法的实际应用，那么很难指望他们能对运用优化方法产生兴奋点。同样，对模型进行数学描述的技能需要长期的练习，这些技能总是需要最长的时间来培养。这就是我写这样一本既展示运筹学应用的多样性，又反复强调优化建模的重要性的教材的原因。每个算法及其原理分析都基于一个短故事，而许多计算习题都是从模型描述开始的。专门讨论模型描述的几章列举了大量来自运筹学实践的真实例子和习题。课时数有限的教师将会感到这些精心挑选的例子很适合教学使用，因为它们既足够长且保留了实际应用的真实感，又足以自成一体，可以用15分钟到30分钟讲完。

任何严肃的数学规划计算都需要使用计算机，因而我也尝试在运筹学引论课程教学改革过程中体现这一点。判断学生的模型描述是否正确的一种方法是将它输入优化软件，然后检查输出结果。这就是在每个带有数值常量的模型描述习题中也包含计算机求解步骤的原因。为加深学生对原理和算法的理解，大量的小规模习题仍要求手工计算；当然条件允许时同样可以使用计算机。我去掉了一些传统引论课程中采用的纯手算例子，因为它们既不能提供求解实际规模问题的工具，也无法提供深入到有效计算机算法中的见解。

考虑到很少有教师能在一门课中覆盖本书的每个论题，并且可能更少有学生能读完本书的每一页，我尽可能使本书在内容上便于灵活裁剪和组合。我在最大程度上减小了章节之间的相关性，而且这些相关性根据参考文献可加以明确区分。全书在结构上可以分解为许多简短的小节，每个小节精心命名，专门讨论一个问题。重要的定义、原理和算法都放置在醒目的文本框内，以便于快速查找。在凡是需要复习或补充背景知识的地方，都安排了一到两页的“知识精要”。原本需要几页纸来严格推导的原理，也被封装在简短的“样本例子”中，并专门加以标注，以便于查找。在书中的习题部分，还设计了几种专门的小图标来分别表明哪些习题需要用计算机或图形计算器，哪些习题在书后提供了答案。

有意使用本教材的教师可能需要花些时间整理他们的教案，以适应本教材的创新之处。实际上他们所花的时间要比最初估计的少得多，因为本书的变化主要影响到教学内容的次序以及概念框架，而对讲授内容的影响相对较小。通过网站 http://vig.prenhall.com/catalog/academic/product/0,1144,0023984155,00.html 可得到教学大纲的范例、课件、软件和其他辅助材料，希望这些材料能对教师有所帮助。

我希望为本书付出的努力能得到他人的公允评价。实际上，在普渡大学我自己的课堂上，学生们的学习效果的改进和学习动力的加强无疑已经让我得到了巨大的回报。在此，我对本书从初稿到出版的整个过程中帮助过我的人们表示衷心的感谢。

同时我要感谢我的妻子Blanca和儿子Rob在我撰写本书的漫长岁月里的鼓励和耐心；感谢两位给予我极大支持的系主任 Fred Leimkuhler 和 Martin Thomas，以及几位始终给予我帮助的 Macmillan 和 Prentice Hall 的编辑。我特别怀念责任编辑 David Johnstone，他启动了撰写本书的征程，然而在本书即将成型之际意外去世。

Ronald L. Rardin 教授
普渡大学工业工程系[①]
西拉斐特，印第安纳 47907-1287
rardin@ecn.purdue.edu
http://www.ecn.purdue.edu/~rardin/

① Rardin 教授已于 2007 年 1 月正式加入阿肯色大学，详细信息及最新联络方式见 http://www.ineg.uark.edu/1520.php ——编者注。

Preface

Operations Research really emerged as an academic discipline with the publication in the late 1960s of a series of pioneering introductory textbooks. A great deal has happened to the field in the three decades since, but most current introductory courses, and most new texts that have emerged along the way, still largely follow the treatment pioneered in those seminal books.

There is much good in that traditional development, but I undertook this new text on the deterministic (mathematical programming) half of OR because I believe students need something more. Emphasis should be on skills and intuitions they can carry away and apply in real settings or later courses. Modeling (or at least formulation) should be central. Students should be excited about the versatility and applicability of OR tools.

Most of the innovations of this book try to deliver on that vision. Numerous algorithms (including simplex) are developed in the context of a common improving search paradigm designed to prepare students for absorbing new material as they encounter it within the course and later. The full breadth of mathematical programming (including linear, integer, nonlinear, and multiobjective programming) is treated from the beginning to highlight the rich diversity of potential applications. Three entire chapters, and large portions of others, are devoted to describing and formulating examples, most of which are based on reports of real applications. The entire development is structured to facilitate its use in a variety of settings by making it as easy as possible to locate material and move around.

The book is intended for first undergraduate and graduate OR courses in engineering and mathematics, and undergraduate/graduate electives in management, as well as introductory courses in subtopics such as linear programming, integer and combinatorial optimization, network flows, and nonlinear programming. Its comprehensive and reentrant coverage should also make it a valuable reference for practitioners and researchers.

My rethinking of how to present mathematical programming begins with the conviction that you cannot do justice to the topic by teaching only linear programming and the simplex method. First of all, the overwhelming majority of optimization problems actually encountered in engineering and management practice cannot be modeled straightforwardly as linear programs. They involve discrete, nonlinear, and multiobjective elements that have to be confronted. Perhaps more importantly, OR practice has evolved to the point where integer, nonlinear, and multiobjective optimization models are dealt with every day. I was surprised myself, in researching the OR applications literature for this book, that accounts of discrete optimization significantly outnumber those using linear programming alone. Goal programs are everywhere. If students have never seen anything but classic LP, they will inevitably conclude OR is too narrow to deal with the complexities of real planning and design tasks.

Focusing heavily on the simplex algorithm also introduces a more subtle roadblock to learning. Regardless of whether simplex is taught as a combinatorial search for the right basic set or as a sequence of linear equation manipulations, decades of refinement on simplex development have almost totally obscured its relation to all the other tools of mathematical programming. A student who has learned only the simplex now carries away little insight around which to build any understanding of the wider field. Even the newer interior point algorithms for linear programming seem completely foreign.

I am not sugggesting that every introductory course must delve deeply into integer and nonlinear programming techniques, or that LP and the simplex should be discarded. What I believe we need is a development that treats LP as the best solved but far from the only form of mathematical program, and that equips students with the intuitions and paradigms to understand more advanced topics when they encounter them in later courses or practice. Where there is the time or the prerequisites to go more deeply into wider topics, that material should be there, developed as much as possible around the same concepts as those of basic LP.

This book tries to avoid equating linear programming with optimization by treating other cases from the very beginning. As in any introduction, the early priority is on formulating optimization problems in standard mathematical programming format. But by including integer, nonlinear, and multiobjective cases along with LPs, students are encouraged to get excited about how vast a range of applications can at least be modeled in this standard way. They can also begin early to recognize different model classes and to be alert to their relative tractability. My experience is that beginners will struggle with intracacies of subscripts, decision variables, constraints, and objectives—just as they do if we treat only LPs—but that the broader range of mathematical forms is well within their prior training. Binary variables are often the only new mathematical concept, and students find them quite intuitive.

Treatment of the improving search (hillclimbing) paradigm begins in Chapter 3 with basic notions of improving and feasible directions, step sizes, local and global solutions, and artificial starts. With the freedom to use nonlinear examples, this can be done in an intuitive and geometic way. Then the simplex algorithm is presented in Chapter 5 as an elegant specialization to linear programming. That improving search treatment of the simplex will seem unfamiliar even to many instructors, but very little is actually changed. Still, with the hillclimbing paradigm firmly established, it becomes much easier to develop interior point methods in Chapter 6, network algorithms in Chapter 10, nonlinear programming methods in Chapters 13-14, and even integer programming procedures in Chapter 12. Time probably will not permit covering all those topics in any single course, but my hope is that the paradigm prepares students for future learning, whether it comes in the current course, in later ones, or in professional practice.

Students can hardly be expected to get excited about using optimization methods if they have never seen them applied except on contrived examples small enough for hand computation. Also formulation skills, which I always find take the longest to develop, demand constant practice. These are why I have tried to make the book both an exposition of the rich diversity of OR applications and a continuing discourse on optimization modeling. Every algorithm and analytic principle is developed in the context of a brief story, and many computational exercises begin with a formulation step. The several whole or partial chapters devoted to formulation are full of realistic examples drawn from operations reseach practice, and exercises add many others. Time-short instructors will find both long enough to preserve a sense of the real application, yet sufficiently self-contained to be presented in 15 to 30 minutes of class time.

All serious mathematical programming computation requires computers, and I have tried to reflect that as well in my updated conceptualization of introductory OR. One way for students to judge whether their formulations are correct is to input them to optimization software and examine the results. That is why every formulation exercise with numerical constants also includes a computer solution step. Hand computation is still employed in numerous small exercises to develop student understanding of principles and algorithms, but here too the computer is exploited when possible. A few purely hand topics of traditional introductory courses have been dropped altogether because they provide neither tools for solving problems of practical size nor insights into the working of effective computer algorithms.

Realizing that few instructors will cover every topic in the book, and that even fewer students will read every page, I have tried the make the book as easy to move around in as possible. Dependencies between sections are minimized and clearly identified with explicit references. The entire development is broken into short and carefully labeled subsections addressing one topic each. Every major definition, principle and algorithm is set out for quick reference in an easy-to-spot box. One- or two-page "Primers" concisely review prerequisite material as it arises in the development. Computations and theoretical principles, which may be derived over several pages, are recapped in brief "Sample Exercises" also marked for easy identification. Icons in the exercises indicate which require computers ( ) or graphic calculators ( ), and which have answers provided at the back of the book ( ).

Instructors who consider adopting the book may have to spend some time aligning their lectures and other materials with my innovations. Still, the cost is much less than it might seem at first glance because the changes have less to do with what is taught than the sequence and conceptual framework within which it is developed. I have tried to help with sample syllabi, notes, software and other aids posted through Internet site http://www.ecn.purdue.edu/~rardin/oorbook/.

I hope the effort to convert will prove justified for others. It has certainly been rewarded with improved learning and greater student motivation in my own classes at Purdue. I owe a great deal of thanks to the hundreds who have aided in its development as they studied from emerging drafts.

I also want to thank my wife Blanca and son Rob for their encouragement and patience through the long years this book has been underway, my two supportive department heads Ferd Leimkuhler and Marlin Thomas, and a sequence of helpful editors with Macmillan and Prentice-Hall. Special recognition is due editor David Johnstone, who launched me on this quest, and was senselessly murdered just as it began to take shape.

RONALD L. RARDIN, PROFESSOR
School of Industrial Engineering
Purdue University
West Lafayette, IN 47907-1287
rardin@ecn.purdue.edu
http://www.ecn.purdue.edu/~rardin/

Contents

Chapter 1 Problem Solving with Mathematical Models 1

1.1 OR Application Stories 1
1.2 Optimization and the Operations Research Process 3
1.3 System Boundaries, Sensitivity Analysis, Tractability and Validity 8
1.4 Descriptive Models and Simulation 11
1.5 Numerical Search and Exact versus Heuristic Solutions 14
1.6 Deterministic versus Stochastic Models 16
1.7 Perspectives 19
Exercises 20

Chapter 2 Deterministic Optimization Models in Operations Research 23

2.1 Decision Variables, Constraints, and Objective Functions 23
2.2 Graphic Solution and Optimization Outcomes 27
2.3 Large-Scale Optimization Models and Indexing 39
2.4 Linear and Nonlinear Programs 46
2.5 Discrete or Integer Programs 52
2.6 Multiobjective Optimization Models 59
2.7 Classification Summary 63
Exercises 64

Chapter 3 Improving Search 77

3.1 Improving Search, Local and Global Optima 77
3.2 Search with Improving and Feasible Directions 88
3.3 Algebraic Conditions for Improving and Feasible Directions 99
3.4 Unimodel and Convex Model Forms Tractable for Improving Search 109
3.5 Searching and Starting Feasible Solutions 118
Exercises 127

Chapter 4 Linear Programming Models 131

4.1 Allocation Models 131
4.2 Blending Models 135
4.3 Operations Planning Models 139
4.4 Shift Scheduling and Staff Planning Models 148
4.5 Time–Phased Models 153

4.6 Models with Linearizable Nonlinear Objectives 157
Exercises 164

CHAPTER 5 SIMPLEX SEARCH FOR LINEAR PROGRAMMING 175
5.1 LP Optimal Solutions and Standard Form 175
5.2 Extreme-Point Search and Basic Solutions 186
5.3 The Simplex Algorithm 197
5.4 Dictionary and Tableau Representations of Simplex 207
5.5 Two Phase Simplex 211
5.6 Degeneracy and Zero-Length Simplex Steps 220
5.7 Convergence and Cycling with Simplex 224
5.8 Doing It Efficiently: Revised Simplex 225
5.9 Simplex with Simple Upper and Lower Bounds 238
Exercises 245

CHAPTER 6 INTERIOR POINT METHODS FOR LINEAR PROGRAMMING 251
6.1 Searching through the Interior 251
6.2 Scaling with the Current Solution 262
6.3 Affine Scaling Search 269
6.4 Log Barrier Methods for Interior Point Search 275
6.5 Dual and Primal-Dual Extensions 287
Exercises 293

CHAPTER 7 DUALITY AND SENSITIVITY IN LINEAR PROGRAMMING 299
7.1 Generic Activities versus Resources Perspective 299
7.2 Qualitative Sensitivity to Changes in Model Coefficients 304
7.3 Quantifying Sensitivity to Changes in LP Model Coefficients: A Dual Model 315
7.4 Formulating Linear Programming Duals 324
7.5 Primal-to-Dual Relationships 329
7.6 Computer Outputs and What If Changes of Single Parameters 338
7.7 Bigger Model Changes, Reoptimization, and Parametric Programming 354
Exercises 363

CHAPTER 8 MULTIOBJECTIVE OPTIMIZATION AND GOAL PROGRAMMING 373
8.1 Multiobjective Optimization Models 373
8.2 Efficient Points and the Efficient Frontier 378
8.3 Preemptive Optimization and Weighted Sums of Objectives 384
8.4 Goal Programming 389
Exercises 400

CHAPTER 9 ***SHORTEST PATHS AND DISCRETE DYNAMIC PROGRAMMING 409***

9.1 Shortest Path Models 409
9.2 Dynamic Programming Approach to Shortest Paths 417
9.3 Shortest Paths From One Node to All Others: Bellman-Ford 426
9.4 Shortest Paths From All Nodes to All Others: Floyd–Warshall 433
9.5 Shortest Path From One Node to All Others With Costs Nonnegative: Dijkstra 440
9.6 Shortest Paths From One Node to All Others in Acyclic Digraphs 446
9.7 CPM Project Scheduling and Longest Paths 450
9.8 Discrete Dynamic Programming Models 458
Exercises 468

CHAPTER 10 ***NETWORK FLOWS 475***

10.1 Graphs, Networks, and Flows 475
10.2 Cycle Directions for Network Flow Search 483
10.3 Rudimentary Cycle Direction Search Algorithms for Network Flows 494
10.4 Integrality of Optimal Network Flows 500
10.5 Transportation and Assignment Models 502
10.6 Other Single-Commodity Network Flow Models 511
10.7 Network Simplex Algorithm for Optimal Flows 519
10.8 Cycle Canceling Algorithms for Optimal Flows 529
10.9 Multicommodity and Gain/Loss Flows 536
Exercises 543

CHAPTER 11 ***DISCRETE OPTIMIZATION MODELS 555***

11.1 Lumpy Linear Programs and Fixed Charges 555
11.2 Knapsack and Capital Budgeting Models 560
11.3 Set Packing, Covering, and Partitioning Models 566
11.4 Assignment and Matching Models 574
11.5 Traveling Salesman and Routing Models 584
11.6 Facility Location and Network Design Models 594
11.7 Processor Scheduling and Sequencing Models 602
Exercises 613

CHAPTER 12 ***DISCRETE OPTIMIZATION METHODS 627***

12.1 Solving by Total Enumeration 627
12.2 Relaxations of Discrete Optimization Models and Their Uses 630