



Springer

如何 求解问题

— 现代启发式方法 —

How to Solve It
Modern Heuristics

(美) Zbigniew Michalewicz David B. Fogel 著
曹宏庆 李艳 董红斌 吴志健 译 康立山 审校



中国水利水电出版社
www.waterpub.com.cn

913

11.10.16

011

如何求解问题

——现代启发式方法

[美] Zbigniew Michalewicz 著
David B. Fogel

曹宏庆 李艳 董红斌 吴志健 译
康立山 审校



A1018310

中国水利水电出版社

内 容 提 要

通过一系列贯穿于章节间的有趣难题,本书深入浅出地阐述了如何利用计算机来求解问题的一些现代启发式方法。

全书包括两部分,共分 15 章。第 1 章指出了造成问题求解困难的主要原因。第 2 章简要介绍了一些基本概念。第 3 章和第 4 章综述了传统的优化算法,包括穷举搜索法、局部搜索法、贪婪法、分而治之法、动态规划法和分枝定界法等。第 5 章阐明了两种现代搜索算法,即模拟退火法和禁忌搜索法。以上各章构成了本书的第一部分。书中第二部分主要阐述求解问题的演化方法。第 6 章和第 7 章介绍了设计一般演化算法的细节问题。第 8 章至第 10 章分别对于 TSP 问题、约束处理问题以及如何调整算法等问题详细综述了如何采用演化方法来求解这些问题所作的大量努力。第 11 章讨论了随时间变化的环境和噪声问题。第 12 章和第 13 章分别提供了神经网络和模糊系统的有关内容。第 14 章对混合系统和扩展演化算法作了简短的一般性讨论。最后第 15 章总结了全书的内容并给出了在实际求解问题时部分有价值的提示。

本书是一本学习如何通过现代启发式方法利用计算机来求解问题的教材,读者对象是高等学校理工科和经济管理专业的广大师生。同时本书丰富的文献综述对于从事计算机特定领域(如算法设计、演化计算、工程优化、神经网络、模糊系统等)研究的科技人员也具有很大的参考价值。

Translation from the English language edition: *How to Solve it: Modern Heuristics* by Zbigniew Michalewicz and David B. Fogel, Copyright © Springer-Verlag Berlin Heidelberg 2000. Springer-Verlag is a company in the BertelsmannSpringer publishing group. All Rights Reserved.

北京市版权局著作仅合同登记号: 图字 01-2002-5219

图书在版编目(CIP)数据

如何求解问题: 现代启发式方法/(美)麦克维克斯(Michalewicz, Z.)等著; 曹宏庆等译. —北京: 中国水利水电出版社, 2003

书名原文: *How to Solve It: Modern Heuristics*

ISBN 7-5084-1383-0

I. 如… II. ①麦…②曹… III. 电子计算机—计算方法 IV. TP301.6

中国版本图书馆 CIP 数据核字(2003)第 004231 号

书 名	如何求解问题——现代启发式方法
作 者	[美] Zbigniew Michalewicz David B. Fogel 著
译 者	曹宏庆 李艳 董红斌 吴志健
出版、发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sale@waterpub.com.cn 电话: (010) 63202266 (总机) 68331835 (发行部) 68359286 (万水)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787×1092 毫米 16 开本 23.5 印张 531 千字
版 次	2003 年 2 月第一版 2003 年 2 月北京第一次印刷
印 数	0001—3000 册
定 价	35.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换
版权所有·侵权必究

译者序

在用纸和笔进行手工求解问题的时代, Polya 的《How to Solve It》一书(1945 年出版)曾经是一本为人们提供了如何用数学方法来求解问题的百科全书, 它的出版获得了极大的成功, 被翻译成 17 种语言并多次出版。而在计算机技术已迅猛发展的今天, 许多实际问题的解答不再是用笔和纸所能计算的了。相反, 我们必须采用一些计算机算法进行数值逼近和扩展问题的范围, 这样才能求得这些问题的有用的答案。因此人们迫切需要一种改进的问题求解方法。本书正是顺应时代发展的要求, 通过一系列贯穿于章节间的有趣难题, 深入浅出地阐述了在 21 世纪如何利用计算机来求解问题的一些现代启发式方法。这些方法是 Polya 的专著中不可能涉及的, 但对于人们求解一些实际问题却极其有效。

我们一直与本书的第一作者美国北卡罗来纳大学的 Zbigniew. Michalewicz 教授保持着良好的合作关系, 这本书的原版是他在组织并参加 2000 年在武汉大学举办的演化计算国际研讨会时带来的。我们发现任何理工科和经济管理专业的大学师生阅读此书后都会有所收获, 不但可以从中学习到一些利用计算机求解问题的启发式方法, 同时对于培养自己求解问题的创造性思维和基本技能也大有裨益。现在我们将本书推荐给广大读者, 希望您能从中受益。并且, 我们极力鼓励您在了解本书内容的同时, 亲自动手编程去尝试和实践书中的难题以及附录 B 所提供的一些问题与项目, 这样做收获会更大。

本书的序言和第 5 章至第 8 章由曹宏庆翻译, 第 1 章至第 4 章由李艳翻译, 第 9 章至第 11 章由董红斌翻译, 第 12 章至第 15 章以及附录 A 和附录 B 由吴志健翻译。书中所有插图的翻译及处理工作由吴志健和喻敬贤完成。我们对此书的翻译一直抱着认真和严谨的态度, 对于在翻译过程中发现的原文中的一些错误和不解之处, 都与作者通过电子邮件进行了反复的交流与讨论, 校出了原文中 20 多处小错误, 并结合作者提供的勘误表对原著中的其他错误进行了改正。本书由曹宏庆负责统稿和校正工作, 最后由康立山教授审校定稿。此外, 覃俊、邹秀芬、康卓、田琳、蒋华、何峰、周爱民、杨辉、闫震宇、付朋辉等老师和同学在阅读初稿的过程中提出了许多宝贵的修改意见, 章毓文老师翻译了书中的谚语, 在此一并表示感谢。最后, 衷心感谢王国顺教授在为我们联系中国水利水电出版社的过程中所给予的热心帮助。

由于我们的水平有限, 译文中不确与谬误之处在所难免, 谨请读者批评指正。

译者

2002 年 9 月于武汉

译者通讯处: 武汉大学软件工程国家重点实验室 邮编 430072

Email: jxyu@whu.edu.cn (曹宏庆) llyyan2000@21cn.com (李艳)

hbdong@whu.edu.cn (董红斌) zjwu@public.wh.hb.cn (吴志健)

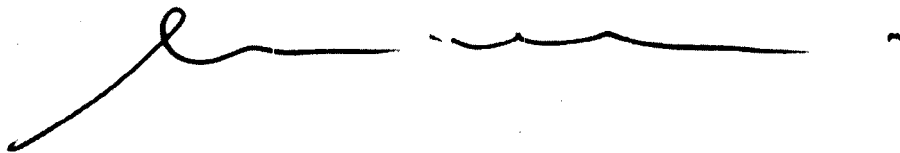
Preface to the Chinese Edition

It is a great pleasure for me to write a special preface for the Chinese edition of this book. First of all, the book was translated by a team of researchers of Evolutionary Computation Group from State Key Laboratory of Software Engineering at Wuhan University, the place I visited a few times in the past. During these trips I met wonderful people & I visited several great places around Wuhan. These trips resulted also with a few joint papers and our continuous cooperation. Second, the project was led by my close friends, Prof. Lishan Kang and Dr. Hongqing Cao. Third, I am very happy to see, that Chinese readers are exposed to continuation of my earlier work (my previous book, *Genetic Algorithms + Data Structures = Evolution Programs*, was translated into Chinese in 2000).

In this volume, I tried to put my favorite topic, evolutionary algorithms, in a broader perspective. On one hand, I tried to place these algorithms in a framework of other heuristic methods (e.g., ant systems or simulated annealing) and operation research techniques (e.g., linear programming). On the other hand, I also tried to describe them in a general context of problem solving activities: this is why the book is loaded with many interesting and unusual puzzles which I have been collecting for over 30 years. These puzzles illustrate many points raised in this text. The bottom line is that to solve a problem, an individual should know a whole range of available tools and algorithms as well as should possess problem solving skills (which is still more art than science).

I do hope that the book will be of interest to Chinese audience. If this is the case, I can promise another volume (in a couple of years from now): a book describing practical case studies of modern heuristic methods applied to a variety of complex problems of the largest corporations of the world. I am just in the process of designing & implementing several complex systems based on the principles outlined in *How to Solve It: Modern Heuristics*; I can not wait to report the results...

Best wishes to all of you,

A handwritten signature in black ink, consisting of a series of loops and a long horizontal stroke, representing the name Zbigniew Michalewicz.

Zbigniew Michalewicz

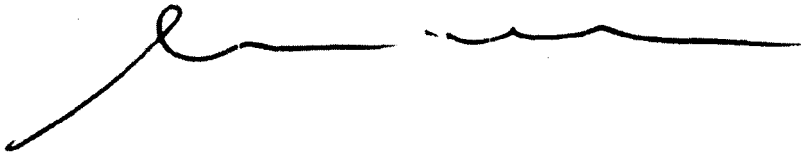
中译本序言

很荣幸我能为本书的中文版写一个特别的序言。第一，这本书是由武汉大学软件工程国家重点实验室演化计算研究室的成员翻译的。我曾多次到那里访问，访问中我结识了一些了不起的人物，参观了武汉的一些宏伟的名胜古迹，我与研究室之间建立了持续合作，并联合发表了一些论文。第二，这项工作是在我亲密的朋友康立山教授与曹宏庆博士的领导下完成。第三，我很高兴地看到中国读者能读到 my 早期研究成果的后续（我的前一本书：**Genetic Algorithms + Data Structures = Evolution Programs** 已译成中文《演化程序——遗传算法与数据编码的结合》由科学出版社于 2000 年出版了）。

在这本书里，我试图以更广阔的眼界来看待我所喜爱的主题——演化算法。一方面，我试图将这些算法置于其他的一些启发式方法（例如蚁族系统或模拟退火）和运筹学方法（例如线性规划）的框架之中；另一方面，我也试图以一种问题求解活动的一般的前因后果关系来描述它们。这就是为什么这本书包含我 30 多年来收集的许多有趣而又不同寻常的难题的原因。这些难题揭示了书中的一些观点。总之，本书最重要的一点是，一个人在求解问题时，既要广泛了解一些可用的工具和算法，同时也应掌握一些求解问题的技巧（这与其说是一门科学，不如说更是一门艺术）。

我很希望中国读者会对这本书感兴趣。如果是这样的话，我可以承诺另一本书的出版（从现在算起可能是两年后的事）——一本描述将这些现代启发式方法应用于解决世界上一些最大公司的各种复杂问题的实例研究的书。我正在应用《如何求解问题——现代启发式方法》这本书中的一些基本原理来设计和实现几个复杂的系统。我几乎迫不及待地想报道这些结果了……

祝各位读者好！



Zbigniew Michalewicz

序言

通世者对兰斯洛特说：
“我会告诉你处世的哲学。”
——佚名 《寻圣杯记》

Gyorgy Polya 的《如何求解问题》(英文名:《How to Solve It》)[287]一书被公认为是 20 世纪关于问题求解的最具贡献的重要文献之一。即使现在,当我们即将跨入新千年之际,这本书仍因其具有指导性的启发式思想而受广大师生所喜爱。此书的第一版面世于 1945 年,即第二次世界大战末和晶体管问世的前几年。这本书的出版很快获得了成功并在 1957 年又出了第二版。

《如何求解问题》是一本关于如何用数学方法来解决一些问题的一些方法的总纲要。也就是说,此书不仅提供了一些技术和过程的具体实例,并且还在如何进行类比、如何使用辅助设备以及如何从求解目标到已知条件逆向地思考问题等方面提出了许多指导性意见。实质上而言,这是一部关于如何手工求解问题的百科全书,并且更重要的是它是一本关于如何建立问题和解决问题的论著。

目前关于启发式求解问题的大多数文献对每种经典算法都提供了算法过程的详细描述。但遗憾的是,它们未能适当地指导人们何时可以使用这些算法,以及更重要地——何时不该使用这些算法。它们往往只提供了一本菜谱,而把决定某种特定方法是否适用于求解手边的问题的任务留给了读者。而读者通常对此毫无准备,既不知道其中可能涉及的有关问题,也不知道确实应该考虑的问题。

这一状况无疑是计算机革命的必然结果。如果说是某个事件促使 Polya 的书出现后近 50 年的今天,人们迫切需要一种改进的问题求解方法的话,那就是廉价的功能强大的台式计算机的出现。许多实际的挑战性问题的解答不再是用笔和纸所能计算的了。我们通过采用计算机算法进行数值逼近和扩展问题的范围,可以求得这些问题的有用的答案。由于计算机如此高效,解题者总是试图去“塑造”(hack)一个解,或者至少看起来像是一个解,而对该程序所实现的方法所作的一些假设并未给予充分的考虑。

结果,尽管人们已经在医药、国防、工业和金融等领域的性能优化方面取得了巨大进展,但却几乎没有发挥我们成果的潜能。例如,采用线性规划方法代替手工计算和主观推测理应每年可以节省成百亿美元,但是在实际条件下¹,人们对这种方法总是运用不当。一些公司和个人总在不断地近乎盲目地急于找到适合问题求解的现成的商品软件,其实它们并不存在。一些过时的其实并不存在的解。你可以想象一下,如果真正合适的方法用于

¹ 因为实际问题往往是非线性的。——译者注

求解问题，而这些问题远非简单的线性规划方法所能处理的话，那会节省甚至赚多少钱啊！

随着世界的发展正越来越面向自由开放的市场，竞争就日益成为寻求更有效的求解问题方法的一种驱动力。达尔文的变化-选择法则与自由市场动力学之间存在着一个贴切的类比。不能获取所需资源的实业将宣告破产，经济上就等同于自然界的“适者生存、不适者被淘汰”。他们所需的就是略微胜过其竞争对手而使其停业，只有那些努力采用现代启发式方法去解决他们的问题而受益的个人、企业和代理商将能得以幸存。

现在有必要对 Polya 的著作进行两方面的更新。首先，读者必须学习一些已有的特殊的技术，主要是一些计算机算法的应用；其次，读者必须懂得每种方法何时可以使用和何时不能使用以及如何构建自己的问题，以便能最好地应用一些启发式方法，而这些方法是 Polya 的专著——《如何求解问题》中所没有预料到的。

本书首次尝试为 21 世纪提供有关问题求解方法的一本综述。书中的主要观点都是通过直接的说明、类比、示例以及一系列贯穿于论述和章节间的问题和难题来表述的。本书旨在提供一本有关现代启发式方法的教材。我们相信这样的一门课程对于理学、商学或工学领域的学生都是必需的。在阅读此书之前要求读者具备离散数学的一些基础知识并熟悉计算机编程。不具备这些基本技能的读者应该花些时间去获取它们，这样做是值得的。而那些希望对算法的数学背景有更深入了解的读者会发现这本书里的材料是一块通向更高级课程的有用的踏脚石。

全书共分 15 章。首先从引言开始，我们对求解问题作了一般性的讨论。第 1 章指出了造成问题求解困难的主要根源。第 2 章简短地介绍了一些基本概念。第 3 章和第 4 章综述了一些经典的优化算法。第 5 章阐明了两种现代搜索算法。以上各章构成了本书的第一部分。继而我们转向求解问题的一种演化方法。第 6 章和第 7 章介绍了直觉和设计演化算法的一些细节。第 8 至第 10 章对于要求寻找项的特定排列、如何处理约束以及如何调整算法使适合于求解任务这些问题提出了一些挑战性观点。这些章节详细综述了人们在这些领域所作的诸多努力。第 11 章讨论了随时间变化的环境和噪声问题。接下来的两章（第 12 章和第 13 章）提供了关于神经网络和模糊系统的指南。第 14 章对混合系统和扩展演化算法作了简短的一般性讨论。第 15 章总结了全书的内容并给出了在实际求解问题时的一些有用提示。你会注意到在每一章（1, 2 等等）的前面都用一个相关难题的小节（I, II 等等）举例说明即将阐明的观点——因为问题的求解应该具有趣味性。我们希望这种方式能使本书更加引人入胜。

本书除了以上的主要内容外，还有两个补充信息的附录。附录 A 提供了概率论和统计学中一些基本概念，从概率的公理一直到统计的假设检验和线性回归。这一附录并不旨在提供随机过程的预备知识，而是为了阐明将概率论和统计学应用于问题求解时的一些重要观点。附录 B 列出了将本书用作大学课程的部分内容时建议采用的求解的问题和项目。即使你在阅读本书时缺乏一位老师的指导，我们也强烈地鼓励你在问题的求解中采取积极主动的态度，自己动手去实现、测试并应用你在这里所学的概念。附录 B 的内容可以充当这些应用的向导。

我们感谢所有在这本书上花费了时间并与我们分享他们的思想的人们——这些思想是

很有帮助的。特别地，我们要感谢 Dan Ashlock, Ole Caprani, Tom English, Larry Fogel, Larry Hall, Jim Keller, Kenneth Keutz-Delgado, Martin Schmidt 和 Thomas Stidsen。我们还要感谢 Kumar Chellapilla，他不仅审阅了本书的章节，并且还对书中一些插图的绘制给予了重要帮助。我们也诚挚地感谢在过去的两年里一些合作者们的帮助，其中许多合作的成果包含在本书中。这些合作者包括 Thomas Bäck, 曹宏庆, Ole Caprani, Dipankar Dasgupta, Kalyan Deb, Gusz Eiben, Susana Esquivel, Raul Gallerd, Özdemir GÖl, Robert Hinterding, Sławomir Koziel, 康立山, Moutaz Khouja, Witold Kosiński, Thiemo Krink, Guillermo Leguizamón, Brian Mayoh, Maciej Michalewicz, 郭涛, Krzysztof Trojanowski, Marc Schoenauer, Martin Schmidt, Thomas Stidsen, Roman Śmierzchalski, Martyna Weigl, Janek Wieczorek, Jing Xiao, Linxin Zhang。

我们感谢 Springer-Verlag 的执行编辑 Hans Wössner 在整个工作中提供的帮助，感谢 Leonard Bolc 教授给予的鼓励和 Antoni Mazurkiewicz 教授对本书中的许多难题所进行的有趣的讨论。特别感谢 Springer-Verlag 的英文版编辑 Joylene Vette 对本书的初稿所提出的宝贵意见。本书的第一作者还要感谢 Aarhus 大学所提供的优良的工作环境，在那里他度过了休假年（1998 年 8 月——1999 年 7 月）并受到国家科学基金（IRI-9322400、IRI-9725424）和 ESPRIT 信息技术的合作研究（CRIT-2）项目 20288——用于生态能量控制的演化实时优化系统的资助，这些工作构成了书中几章内容的基础。此外，他还要感谢 UNC-Charlotte 大学（美国）、国立 de San Luis 大学（阿根廷）和 Aarhus 大学（丹麦）的所有上过他在 1997 年到 1999 年所开设的课程并经历过艰苦的问题求解过程的研究生们。本书的第二作者感谢所有上过他于 1999 年冬季在 UC San Diego 所开设的关于机器学习和模式识别的课程的本科学们，他还要感谢自然选择公司的职员 Bill Porto, Peter Angeline 和 Gary Fogel 等所给予的支持和鼓励，并且特别感谢 Jacquelyn Moore 为这本书的完成放弃了许多周末和夜晚的休息时间。

如果你读完本书后发现它富有挑战性、趣味性和争议性，我们的目的也就达到了。我们希望你喜爱这本书，并从中受益。

Charlotte, NC
La Jolla, CA
1999 年 9 月

Zbigniew Michalewicz
David B. Fogel

目 录

引言	1
I 我的三个小孩的年龄有多大?	6
1 为何有些问题难以求解?	8
1.1 搜索空间的大小	8
1.2 给问题建模	12
1.3 随时间而变化	15
1.4 约束	16
1.5 证明问题	18
1.6 你辉煌成就的机会	19
1.7 小结	22
II 一个模型有多重要?	23
2 基本概念	26
2.1 表示方式	26
2.2 目标	27
2.3 评估函数	28
2.4 定义一个搜索问题	29
2.5 邻域和局部最优解	29
2.6 爬山法	31
2.7 你会落入这种圈套吗?	33
2.8 小结	35
III 7-11 连锁店里的价格是多少?	37
3 传统方法——第一部分	41
3.1 穷举搜索	43
3.1.1 枚举 SAT 问题	44
3.1.2 枚举 TSP 问题	45
3.1.3 枚举 NLP 问题	47
3.2 局部搜索	48
3.2.1 局部搜索和 SAT 问题	49
3.2.2 局部搜索和 TSP 问题	50
3.2.3 局部搜索和 NLP 问题	52
3.3 线性规划: 单纯形法	59
3.4 小结	62

IV 这些数是什么?	63
4 传统方法——第二部分	66
4.1 贪婪算法	66
4.1.1 贪婪算法和 SAT 问题	66
4.1.2 贪婪算法和 TSP 问题	67
4.1.3 贪婪算法和 NLP 问题	68
4.2 分而治之法	69
4.3 动态规划法	71
4.4 分枝定界法	78
4.5 A* 算法	81
4.6 小结	84
V 熊是什么颜色?	85
5 跳离局部最优	88
5.1 模拟退火	90
5.2 禁忌搜索	96
5.3 小结	103
VI 你的直觉如何?	104
6 演化方法	107
6.1 求解 SAT 的演化方法	109
6.2 求解 TSP 的演化方法	111
6.3 求解 NLP 的演化方法	114
6.4 小结	115
VII 这些东西中有一个与众不同	120
7 演化算法的设计	123
7.1 表示方式	126
7.1.1 固定长的符号向量	127
7.1.2 排列	127
7.1.3 有穷状态机	128
7.1.4 符号表达式	128
7.2 评估函数	129
7.3 变化算子	131
7.3.1 固定长的符号向量	131
7.3.2 排列	132
7.3.3 有穷状态机	133
7.3.4 符号表达式	134
7.4 选择	136
7.5 初始化	138

7.6 小结	139
VIII 最短路径是什么?	140
8 旅行商问题	143
8.1 寻找好的变化算子.....	145
8.2 结合局部搜索方法.....	161
8.3 其他可能性.....	163
8.3.1 边组装杂交.....	164
8.3.2 反序-杂交算子	166
8.4 小结	169
IX 斑马属谁?	171
9 约束处理技术.....	175
9.1 概述	176
9.1.1 $eval_f$ 的设计	177
9.1.2 $eval_u$ 的设计	179
9.1.3 $eval_f$ 和 $eval_u$ 之间的关系	179
9.1.4 拒绝不可行解.....	180
9.1.5 修补不可行个体.....	181
9.1.6 用修补后个体替换原个体.....	181
9.1.7 惩罚不可行个体.....	182
9.1.8 通过使用专门的表示方式和变化算子保持一个可行的种群	182
9.1.9 使用译码器.....	183
9.1.10 个体与约束的分离.....	184
9.1.11 探索搜索空间的可行部分与不可行部分的边界	184
9.1.12 寻找可行解.....	185
9.2 数值优化.....	186
9.2.1 基于保持解的可行性的方法.....	186
9.2.2 基于罚函数的方法.....	189
9.2.3 基于搜索可行解的方法.....	195
9.2.4 基于译码器的方法.....	201
9.2.5 混合方法	202
9.3 小结	204
X 你能调整问题吗?	206
10 针对问题调整算法.....	211
10.1 演化算法中的参数控制.....	211
10.2 用一个 NLP 说明问题.....	214
10.3 控制技术的分类.....	216
10.4 参数控制方法.....	219

10.4.1 表示方式	219
10.4.2 评估函数	220
10.4.3 变异算子和变异率	220
10.4.4 杂交算子和杂交率	222
10.4.5 父体的选择	224
10.4.6 种群	224
10.5 参数控制的组合形式	225
10.6 小结	226
XI 你能两步制胜吗?	229
11 随时间变化的环境和噪声	232
11.1 动态变化的世界	232
11.2 现实世界是有噪声的	239
11.3 小结	246
XII 元旦是星期几?	251
12 神经网络	254
12.1 阈神经元与线性划分函数	254
12.2 前馈多层感知器的反传	259
12.3 训练与测试	262
12.4 递归网络及其扩展结构	263
12.4.1 标准递归网络	264
12.4.2 Hopfield 网络	264
12.4.3 Boltzmann 机	265
12.4.4 多交互程序的网络	266
12.5 采用竞争网络进行聚类	267
12.6 应用神经网络求解 TSP	269
12.7 演化神经网络	270
12.8 小结	271
XIII 这根绳子有多长?	273
13 模糊系统	276
13.1 模糊集	276
13.2 模糊集和概率测度	277
13.3 模糊集的运算	278
13.4 模糊关系	280
13.5 设计模糊控制器	282
13.6 模糊聚类	286
13.7 模糊神经网络	289
13.8 模糊 TSP	291

13.9 演化模糊系统.....	292
13.10 小结	293
XIV 你喜欢简单的解决办法吗?	294
14 混合系统.....	299
15 总结	308
附录 A 概率与统计	317
A.1 概率的基本概念.....	317
A.2 随机变量.....	318
A.2.1 离散型随机变量.....	319
A.2.2 连续型随机变量.....	321
A.3 随机变量的描述性统计量.....	322
A.4 极限定理与极限不等式.....	324
A.5 随机变量的相加.....	325
A.6 在计算机中产生随机数.....	326
A.7 估计.....	327
A.8 统计的假设检验.....	329
A.9 线性回归.....	330
A.10 小结.....	331
附录 B 问题与项目	333
B.1 尝试一些实际问题.....	334
B.2 报道采用启发式方法的计算实验	338
参考文献.....	340

引言

理性的人努力使自己适应这个世界；

疯狂的人坚持使世界适应自己；

因此，一切进步取决于疯狂的人们。

—— G. B. 肖伯纳 《革命家箴言》

这不是一本算法专著。当然书中充满了算法，但那不是本书的主题。本书旨在讨论求解问题的可能性。它不仅要向你们提供现有求解问题方法的一些必要的预备知识，更重要的是帮助你们拓展自己的才能去构建新的问题和进行创造性的思维，以培养怎样求解问题的才能，而这种才能已成为一种失传的艺术，人们不再投入必要时间和进行严谨的思索去构建一个问题，为了调整对难题的支离破碎的表述，人们已变得自满并且只想抓到一个最简便的子程序，一种包医百病的灵丹妙药。实际上，这种魔术（套用子程序的办法）的麻烦在于它的成功率很低，而且常常需要借助一些诸如镜子、烟雾之类的外部装置（一些不切实际的假设），像玩魔术一样，采用这种方法求解实际问题的大部分所谓成功的应用其实都是不可靠的，只是得到一些幻觉罢了（一些自以为是的结果）。

有效地求解问题的重要性从来没有现在这么强烈。科技已经使人类具有了影响外部环境的巨大能力，以至于我们现在所做的一切都将对人类的明天产生不可磨灭的影响。而同样的科技还在继续扩大我们所接触的人数和影响我们生活的人数。结果，问题求解变得越来越难，因为有更多的因素需要考虑——但往往潜在的最坏情况是忽略这些因素，希望在不考虑它们时也能找到这些经过简化的问题的有效解来。不管你准备用什么样的方式对待自己的生活，是为你的国家着想善待资源、物尽其用，还是仅仅为今天早上上班找一条最佳路线，你都不能简单地将自己置身世外桃源，因为你和周边世界的联系已日益频繁和复杂。你可以通过合理地解决这些问题而收益大增，也可以因为这些问题求解不善而失之颇多。

我们都会面临问题，问题存在于现实和理想状态之间的差异。求解之道就是合理分配可利用的资源来减少现实与理想之间的差距。为了理解这句话，我们首先必须意识到所谓问题就是决策人的目的意识驱使的产物，没有追求的人也就不会提出任何问题。但问题在于我们不仅有追求的目标，而且很多时候这些想法各执一端，最起码不是相互补充的。为了寻找解决问题的最佳途径，你不能孤立地看待事物，而应考虑别人的行为如何影响你的解答。有可能一旦你与他们联系后就会改变自身资源的分配——这也许使你事半功倍，也可能阻碍你前进。不考虑“下一步将发生什么”而实施一个方案，就好像在玩只有一个玩家的扑克牌游戏：你和你的钱财很快就要分家。

将实际问题作为一个多人博弈来处理的先决条件是具有运用求解问题的技术的能力，也就是适用于各种情况的算法。不幸的是，实际问题总是与运用这些方法所要求的条

件或多或少有些出入。打个比方，有一种特殊的技术用来计算最优资源分配问题，它的代价函数和约束条件（等式约束和不等式约束）都是线性的。该方法快速又可靠。但它总是不适当地被应用于实际问题中，因为实际问题的代价函数和约束条件几乎都是非线性的。实际上，任何使用该方法的人都可能得到问题的根本不存在的正确解答。用这种解答所导致的损失是多方面的：不仅正确处理实际代价函数的约束条件的人会得到更好的解答，而且一个正确处理该情况的竞争对手也可以找到这个更好的解答！这只是利用一方的微弱优势去打败另一方（拉斯维加斯对这个原理的运用已有多年）。另一方面，用一个快速但近似的解法也许是正确的：如果你的竞争对手正用一个非常缓慢的过程去寻找正确解，那你也许能在此之前发现一些虽不那么精确但却有用的解答，而且有时候拥有一些解答比长时间等待一个更好的解更有用。究竟采取哪种合适的方式需要你自己去判断。

遗憾的是，判断力似乎正在衰减，或者至少已不流行了。也许是因为这要付出艰苦的劳动。为了在求解问题时作出正确的判断，你必须知道每种特定的求解方法的假设条件、计算开销和可靠性等等。为了获取这些信息通常需要付出很多时间和精力，而仅在互联网上查找或者使用看起来适合手边问题的现成的商业软件当然更容易一些。确实，一些商家和代理商委托购买着这种现成软件（COTS）。当然，这样做开始时花费很少，不过这样往往是捡了芝麻丢了西瓜。几年以后，当这个软件不能满足商业用途的实际需要，而当其他竞争对手又能在短时间内得出他们的解答时，买来的软件通常就被束之高阁、不被使用，工人们又得回过头来求解原来的问题，有时候甚至用手工作业！

与其面对这种情况哀叹，不如把它当作一个很好的机会。丰厚的奖励正等着那些能够有效地求解问题的人，而且你也能够发挥这方面的才能。我们必须从正确了解所要达到的目的入手。在大多数情况下，在试图使用一个熟悉的算法时，甚至改变我们的目标而去适应算法。这就好比是我们只有一个锤子却要把一个螺丝钉拧进木板里：我们把它硬锤了进去，尽管我们知道再也不能用螺丝刀把它起出来了。如果只用一把锤子，那就把所有的事物都当作钉子来对待。同样，如果你只有一组传统的求解问题的方法，那么所有的代价（目标）函数将被魔术般地转变成光滑、线性和凸的映射，而且每一个模型都按均方差进行优化。这将产生一些看上去“完美”但实际上却不中肯的答案。懂得合适决策者的目标函数应该或不应该线性化以及应该或不应该采用其他简化技术是非常重要的。用客观的、可度量的术语来形式化主观的目的并以此来识别近似和实际之间的差别是一项需要实践的挑战性工作，没有什么能够取代这一过程。

当我们对建立决策者的目标模型所需的精确度作出了判断之后，下一步就是要选择一个能找到对现有资源进行最优分配的策略。对于这个问题你们有多种选择。有很多书介绍了求解问题的算法，而且每一个都提出了一系列独立的技巧。仅仅在书中找算法就像去查找烹制蛋糕的食谱一样诱人。不过这种自然的趋势抛弃了创造力，它不能向你提供成为一个天才的机会。相反，它迫使你把问题简化以适应特定算法的限制。为了急于展示求解问题的技巧，这些算法通常被孤立地给出而忽略了不同过程间的重要的潜在的协同性。这就好像是不考虑各块之间的组合来完成一个复杂的拼图游戏。将不同的方法有效地混合起来使用往往可以得到实际问题的更好解答。有效地求解问题需要的不仅仅是对算法的了解；为了在允许

的时间内并考虑到参与竞争活动的人们的各种可预料的行为下完成要达到的目的，它还需要致力于确定各种方法的最优组合。认识到实际问题的复杂性是有效地求解它们的前提。

不幸的是，我们的麻烦在生命的早期就已出现，甚至早到上小学的时候。我们被教导去分解问题，去孤立地解决更简单、更小的子问题。这种方法适用时一切都很棒，但复杂的实际问题经常不能被简单地、有意义地分解。而且更糟糕的是，我们被填鸭式地灌输着问题的求解方法，一章一章地，却从未被强迫去思考我们所面临的问题是否应该仅仅用我们教科书里描述的方法去解决。当然，这些方法是必须的！否则为什么这些问题会在这一章出现呢？！这种情况不仅仅出现在中学和小学的数学课本中，它也出现在大学的大部分课本中，甚至还出现在许多阐明特殊技术并用实际例子进行演示的专著中。这些书中的问题和解答从来就没有分开过。

让我们来看看初中三年级的数学课本。也许那里有一章专门求解二次方程的。在这一章的后面有很多练习题。其中有一道是这样的：一个农夫有一个长方形的农场，它的周长是 110 米，面积是 700 平方米。问农场的边长各是多少？学生们知道为了解决这个问题，他应该用刚刚在这章讨论的方法。于是他会建立两个方程

并由它导出一个二次方程：

$$\begin{cases} 2x + 2y = 110 \\ xy = 700 \end{cases}$$

$$x(55 - x) = 700。$$

然后根据一些给定的公式，他很容易算出 x 并解决这个问题。

另一章是关于几何和讨论三角形的性质的。它解释了毕得哥拉斯定理，并且这一章末尾总结了另外一系列需要解决的问题。学生毫不怀疑他应该用这个定理来解决这些问题。

但是这看上去并不是正确的教育之道。问题和方法的关系应该通过对问题而不是对方法的讨论而得到。从长远来看，不这样做弊大于利，因为像所有的“分发品”，它使得学生不能够独立地思考问题。

让我们通过下面两个例子来说明这一点。第一个例子如下所述。有一个三角形 ABC ， D 是三角形内任意一点（如图 0.1），你所需要做的是证明

$$AD + DB < AC + CB，$$

其中 AD 表示点 A 到点 D 的距离，其他符号类似地表示所代表的两顶点间的距离。

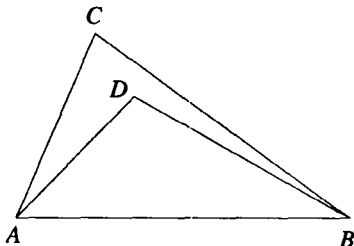


图 0.1 带内点 D 的三角形 ABC