

国外计算机教材系列

# 人工智能

## ——理论与实践

Artificial Intelligence:  
Theory and Practice

英文版

Thomas Dean

[美] James Allen 著

Yiannis Aloimonos



电子工业出版社  
Publishing House of Electronics Industry  
<http://www.phei.com.cn>

国外计算机科学教材系列

# 人工智能——理论与实践

(英文版)

Artificial Intelligence:  
Theory and Practice

[美] Thomas Dean  
James Allen 著  
Yiannis Aloimonos

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

## 内 容 简 介

这是一本阐述人工智能基本理论及其实际应用的教材,由三位资深的人工智能专家精心编著而成。针对机器智能系统开发中涌现出的表达与计算问题,本书介绍了最新的研究成果,并讨论了系统实现中涉及到的实际问题。作者深入探讨了用于解决学习、规划和不确定性问题的传统符号推理技术,例如演绎推理、对策树等,并介绍了神经网络、概率推理等新技术。书中出现的重要算法在每章后面都附有其Lisp实现的源代码,以供读者在实验时进行参考。另外,本书还给出了丰富的人工智能应用系统的实例。

本书可作为高等院校计算机、控制、机电、数学等专业人工智能课程的教材,也可供从事人工智能研究及应用的科学工作者和工程技术人员学习参考。

English reprint Copyright © 2003 by PEARSON EDUCATION NORTH ASIA LIMITED and Publishing House of Electronics Industry.

Artificial Intelligence: Theory and Practice, ISBN: 0805325476 by Thomas Dean, James Allen, and Yiannis Aloimonos. Copyright © 1995.

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macau).

本书英文影印版由电子工业出版社和Pearson Education培生教育出版北亚洲有限公司合作出版。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有Pearson Education培生教育出版集团激光防伪标签,无标签者不得销售。

版权贸易合同登记号:图字:01-2003-0356

### 图书在版编目(CIP)数据

人工智能——理论与实践 = Artificial Intelligence: Theory and Practice/ (美)迪安(Dean, T.)等著.

-北京:电子工业出版社, 2003.6

(国外计算机科学教材系列)

ISBN 7-5053-8780-4

I. 人... II. 迪... III. 人工智能-教材-英文 IV. TP18

中国版本图书馆CIP数据核字(2003)第043200号

责任编辑:冯小贝

印刷者:北京东光印刷厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编:100036

经 销:各地新华书店

开 本:787 × 980 1/16 印张:37 字数:829千字

版 次:2003年6月第1版 2003年6月第1次印刷

定 价:49.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010) 68279077

## 教材出版委员会

主 任	杨芙清	北京大学教授 中国科学院院士 北京大学信息与工程学部主任 北京大学软件工程研究所所长
委 员	王 珊	中国人民大学信息学院院长、教授
	胡道元	清华大学计算机科学与技术系教授 国际信息处理联合会通信系统中国代表
	钟玉琢	清华大学计算机科学与技术系教授 中国计算机学会多媒体专业委员会主任
	谢希仁	中国人民解放军理工大学教授 全军网络技术研究中心主任、博士生导师
	尤晋元	上海交通大学计算机科学与工程系教授 上海分布计算技术中心主任
	施伯乐	上海国际数据库研究中心主任、复旦大学教授 中国计算机学会常务理事、上海市计算机学会理事长
	邹 鹏	国防科学技术大学计算机学院教授、博士生导师 教育部计算机基础课程教学指导委员会副主任委员
	张昆藏	青岛大学信息工程学院教授

## 出版说明

21世纪初的5至10年是我国国民经济和社会发展的重要时期,也是信息产业快速发展的关键时期。在我国加入WTO后的今天,培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡,是我国面对国际竞争时成败的关键因素。

当前,正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期,为使我国教育体制与国际化接轨,有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材,以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验,翻译出版了“国外计算机科学教材系列”丛书,这套教材覆盖学科范围广、领域宽、层次多,既有本科专业课程教材,也有研究生课程教材,以适应不同院系、不同专业、不同层次的师生对教材的需求,广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时,我们也适当引进了一些优秀英文原版教材,本着翻译版本和英文原版并重的原则,对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上,我们大都选择国外著名出版公司出版的高校教材,如Pearson Education培生教育出版集团、麦格劳-希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者,如道格拉斯·科默(Douglas E. Comer)、威廉·斯托林斯(William Stallings)、哈维·戴特尔(Harvey M. Deitel)、尤利斯·布莱克(Uyless Black)等。

为确保教材的选题质量和翻译质量,我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士,也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中,为提高教材质量,我们做了大量细致的工作,包括对所选教材进行全面论证;选择编辑时力求达到专业对口;对排版、印制质量进行严格把关。对于英文教材中出现的错误,我们通过与作者联络和网上下载勘误表等方式,逐一进行了修订。

此外,我们还将与国外著名出版公司合作,提供一些教材的教学支持资料,希望能为授课老师提供帮助。今后,我们将继续加强与各高校教师的密切联系,为广大师生引进更多的国外优秀教材和参考书,为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

## About the Authors

**Thomas Dean** is a Professor in the Computer Science Department at Brown University. His general research interests include temporal and spatial reasoning, planning, robotics, learning, and probabilistic inference. Professor Dean's recent work has led to the design and implementation of a temporal database system for applications involving mobile robots and factory automation. He is on the executive council and is a Fellow of the American Association for Artificial Intelligence (AAAI). He served as the program co-chair for the 1991 National Conference on Artificial Intelligence. Professor Dean was also a recipient of the NSF Presidential Young Investigator Award (1989–1994).

**James Allen** is the John H. Dessaurer Professor of Computer Science at the University of Rochester. He is a fellow of the AAAI and was a recipient of the NSF Presidential Young Investigator Award (1985–1989). In addition, Professor Allen was the Editor-in-Chief of *Computational Linguistics* from 1983 to 1993.

**Yiannis Aloimonos** is an Associate Professor at the Computer Science Department and the Institute for Advanced Computer Studies of the University of Maryland. He also heads the Computer Vision Laboratory of the Center for Automation Research. His research interests include computer vision and the integration of perception, reasoning, and action. He is a recipient of the NSF Presidential Young Investigator Award (1990–1995).

# Preface

**T**his book is designed to introduce students to a set of theoretical and computational techniques that serve as a foundation for the study of artificial intelligence (AI). The presentation is aimed at students with a background in computer science at about the sophomore or junior level in college. The emphasis is on algorithms and theoretical machinery for building and analyzing AI systems. Traditional symbolic AI techniques such as deductive inference, game-tree search, and natural language parsing are covered, as are hybrid approaches such as those employed in neural networks, probabilistic inference, and machine vision. The coverage is broad, with selected topics explored in greater depth but with no attempt to exhaustively survey the entire field.

## **Representation**

The book focuses on the importance of representation in the core chapters dealing with logic, search, and learning. It incorporates a more formal treatment of AI than is found in most introductory textbooks. This formal treatment is reflected in the attention given to syntax and semantics in logic and in the material concerning the computational complexity of AI algorithms.

The material on learning draws on recent unifying work in computational learning theory to explain a variety of techniques from decision trees to neural networks.

The book provides a consistent pedagogic example of AI in the real world through examples focusing on AI systems corresponding to robots and software automation "softbots." A wide range of other examples are also introduced to characterize both the potential and the variety of AI applications. The chapters on natural language processing, planning, uncertainty, and vision supply a state-of-the-art perspective unifying existing approaches and summarizing challenging areas for future research.

This book is not meant as an exhaustive survey of AI techniques. Subjects such as qualitative reasoning about physical systems and analogical reasoning are only briefly touched on in this text. Other subjects are given much more attention in this book than in traditional texts. Learning, planning, and probabilistic reasoning are treated in some depth, reflecting their increased importance in the field. The chapter on vision (Chapter 9, Image Understanding) is substantial in its coverage of topics to reflect the importance of perception in understanding intelligence and building artifacts that interact with the world in useful and interesting ways.

## **Theory and Practice**

Although the text emphasizes theoretical foundations, practical problems involved with the implementation of AI algorithms are addressed in every chapter. A self-contained introduction to symbolic programming in Common Lisp is provided to encourage students to perform computational experiments. Lisp code is given for many of the important algorithms described in the text; however, the text is designed so that the student can ignore Lisp and implementation issues altogether if he or she chooses. The code uses a carefully chosen subset of Common Lisp to teach algorithmic issues in AI. In contrast, other texts use AI algorithms to teach Lisp programming techniques.

All the algorithms in the text are described in English prose and pseudo code. In the case of algorithms that are also given in Lisp, most of the time the code appears in a Lisp Implementation appendix at the end of the chapter, but on some occasions it appears in the main body of the chapter. Code appears in the main body when it is considered particularly important that the student explore the underlying issues empirically. With most of the Lisp code relegated to appendices, instructors are free to choose the areas they want to emphasize empirically. The tight coupling between the descriptions of algorithms in the text and the accompanying Lisp code makes it easy for students to experiment, without the bother of using two texts with different perspectives and algorithmic approaches.



We use Lisp instead of Prolog because Lisp is closest in structure to languages such as Pascal and C that students are likely to be familiar with. We use Lisp instead of Pascal or C because the list processing and symbolic manipulation routines available in Lisp allow for elegant implementations of important algorithms that can be compactly listed. Note, however, that a library of C++ code is available (see the section on "Supplements") that mirrors the Common Lisp treatment in the text function for function and algorithm for algorithm.

## **To the Student**

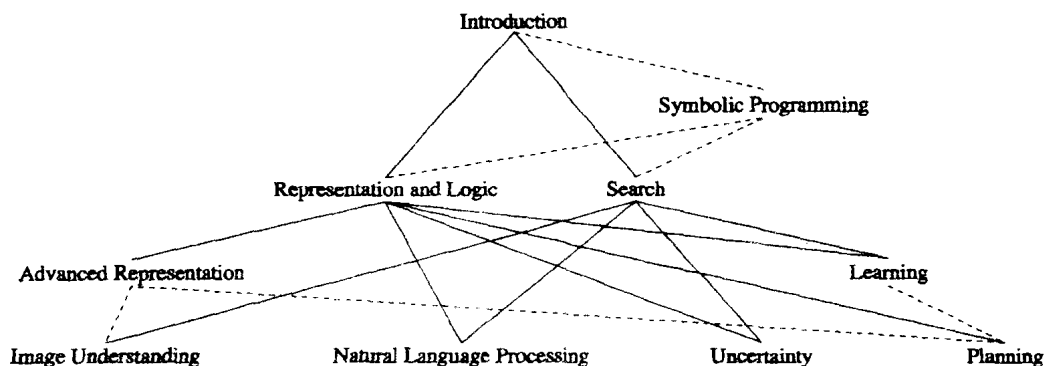
Preface material is usually aimed at instructors who are thinking of adopting a text for a course. Generally, students cut straight to the first chapter or the table of contents to get some idea of what the book is about. This book is designed to teach students about the theory and practice of building computer programs that perform interesting and useful tasks. With the exception of some diversions in the introductory chapter, we leave the philosophical conundrums to the philosophers and focus on techniques, algorithms, and analytical tools that we believe students will find useful in building sophisticated (even *intelligent*) computer programs.

The book describes precise problems, analyzes them from a computational perspective, and specifies efficient algorithms for their solution where possible. Along the way, we provide the necessary logic, computer science, and mathematics for you to understand the important issues and ultimately develop your own solutions and propose your own problems. Our hope is that you will find the techniques and ideas in this book useful, whether you pursue a career in engineering, computer science, business management, or any other area that requires you to think in terms of computational processes that have to interact with a complex and changing world.

## **To the Instructor**

The core material in the book is in the first five chapters covering basic introductory and motivational material, symbolic programming for courses interested in implementation, representation and logic, search, and learning. Within these core chapters, instructors have considerable flexibility regarding what to include and how much time to spend on particular topics.

The choice of topics and allocation of lecture time will depend on the background of the students taking the course. Often students have a reasonable background in boolean logic from a previous course in computer science, engineering, or mathematics, in which case the chapter on representation and logic can move along rather quickly. Search issues are generally familiar to computer science students, and the basic blind-search methods, including depth-first and breadth-first search, should take very little time.



**Figure 1** This graph illustrates some of the dependencies and connections among the chapters in this text. A solid line indicates a strong dependency between two chapters, and a dashed line indicates a connection or conditional dependency between two chapters that an instructor may wish to consider.

We recommend spending a significant amount of time on learning since the area is reasonably mature, the issues dramatically illustrate the role of representation and search, and students are generally fascinated with the prospect of building systems that learn.

Representation appears before search in the order of chapters because representation is the more fundamental idea as far as AI is concerned. We emphasize logic because it enables students to think precisely about representation. Pedagogically, there are situations that merit covering search before representation; if you are teaching Lisp and this is the students' first exposure to Lisp and symbolic programming, consider covering the search chapter first because the examples of search procedures provide a somewhat easier introduction to Lisp programming issues. With the exception of the section on discrimination networks at the end of the search chapter, the representation and search chapters can be covered in either order.

Figure 1 illustrates some of the dependencies and connections among the chapters in this text. A solid line indicates that one chapter should be covered before another chapter in order to fully understand all the material. A dashed line indicates a connection between two chapters that an instructor may wish to emphasize or a conditional dependency that an instructor may wish to account for. For example, the section on spatial representation and robot navigation in Chapter 6 (Advanced Representation) can be used to motivate and set the stage for topics covered in Chapter 9 (Image Understanding). All the chapters are conditionally dependent on Chapter 2 (Symbolic Programming) if implementation issues are to be covered and the students require instruction in symbolic programming methods. Additional information regarding chapter dependencies and synergies as well as suggestions for course syllabi are available in the Instructors Resource Guide (see the section on "Supplements").

## Supplements

Source supplemental materials for this book are available via anonymous FTP from `av.com` in the subdirectory `av/dean`. The supplemental materials for the book include the following items:

- Instructor's Guide and Solutions Manual—contain notes on each chapter, solutions to selected exercises, additional exercises for the Lisp (Chapter 2) and vision (Chapter 9) chapters, and sample exams with answers. This guide is available only on a 3½" disk from the publisher (ISBN 32548-4).
- Selected figures—selected figures in encapsulated PostScript format are available for overhead transparencies.
- Source code—the sample source code contained in the text is available in both Lisp and C++ implementations. Other implementations may be available (for example, the Scheme dialect of Lisp); check the `README` file in `bc/dean` for current status and recent developments.

To obtain the supplemental materials, FTP to `bc.av.com` as follows:

```
% ftp av.com
```

and log on as `anonymous`. Use your electronic mail address as your password and connect to the directory for this book by typing

```
% cd av/dean
```

Before retrieving supplements, it is a good idea to look at the `README` file to see if changes have been made since this book went to press. You can retrieve this file by typing

```
% get README
```

Type `quit` to exit FTP and read the `README` file. (Although you could read the file online, it is courteous not to load the FTP server while you are just reading.) Then log back on when you are ready to download the files that you want. Using FTP to retrieve archived files can get complicated. The `README` file will give you some additional advice, but you may find it helpful to consult your favorite UNIX guide or local artwork wizard.

## Thanks

We benefited from the efforts of many previous authors in organizing the material covered in this text and figuring out how to present the material to students. In particular, we would like to acknowledge the texts by Charniak and McDermott [1985], Nilsson [1980], and Winston [1979] that provided our first introductions to AI. We are also indebted to Davis [1990], Genesereth and Nilsson [1987], Ginsberg [1993], and Winston [1992], whose books we consulted while writing this text.

A lot of friends and colleagues contributed to this book by providing feedback to the authors during crucial times during its writing. In particular, we would like to thank the following people:

Mark Boddy	Robert Goldman	Bart Selman
Chris Brown	Steve Hanks	Jude Shavlik
Jack Breese	Eric Horvitz	Yoav Shoham
Eugene Charniak	Leslie Kaelbling	Mike Shwe
Ernie Davis	Paris Kanellakis	Austin Tate
Mark Drummond	Richard Korf	Prasad Tadepalli
Charles Dyer	Tom Mitchell	Dan Weld
Nort Fowler	Ann Nicholson	Mike Wellman

The production and editorial help from Addison-Wesley Publishing Company was exceptional. We would like to thank Carter Shanklin, the Acquisitions Editor for Computer Science at Addison-Wesley, for shepherding us through the entire process. Judith Hibbard, the Senior Production Editor, Melissa Standen, the Editorial Assistant, and Ari Davidow, the Technical Production Assistant, were amazingly helpful at all the right times. We would also like to thank Sara Larson from the University of Maryland at College Park, who helped with the vision chapter, and Mary Andrade and Dawn Nichols from Brown University, who provided invaluable assistance in keeping track of the many drafts and handling the extensive correspondence that was required.

A large number of students also provided feedback while we were writing the text, far too many to list, but the following deserve special mention:

Scott Baetz	Michael Littman
Ted Camus	Mike Perkowitz
Lloyd Greenwald	Kostadis Roussos
Shieu-Hong Lin	Smith Surasmith

A special thanks goes to Jon Monsarrat for his unflagging enthusiasm for the project and for the many hours he spent hacking UNIX, PostScript, and  $\text{\LaTeX}$ . Finally, each of us would like to thank our respective spouses and loved ones for being supportive when it was needed most.

*Thomas Dean*  
*James Allen*  
*Yiannis Aloimonos*

# 目录概览

第 1 章 绪论 .....	1
Introduction	
第 2 章 符号编程 .....	23
Symbolic Programming	
第 3 章 表达与逻辑 .....	71
Representation and Logic	
第 4 章 搜索 .....	131
Search	
第 5 章 学习 .....	179
Learning	
第 6 章 高级表达 .....	255
Advanced Representation	
第 7 章 规划 .....	297
Planning	
第 8 章 不确定性 .....	355
Uncertainty	
第 9 章 图像理解 .....	409
Image Understanding	
第 10 章 自然语言处理 .....	489
Natural Language Processing	
参考文献 .....	539
Bibliography	
词汇索引 .....	551
Vocabulary Index	
代码索引 .....	559
Code Index	

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	Robot Explorers, 2	
1.1	Artificial Intelligence in Practice 3	
	Examples of Artificial Intelligence Systems, 4	
1.2	Artificial Intelligence Theory 5	
	Examples of Artificial Intelligence Theory, 6	
1.3	Identifying and Measuring Intelligence 7	
1.4	Computational Theories of Behavior 9	
	Representation, 10	
	Syntax and Semantics, 11	
1.5	Automated Reasoning 12	
	Inference and Symbolic Manipulation, 13	
	Representing Common-Sense Knowledge, 14	
	Combinatorial Problems and Search, 14	
	Complexity and Expressivity, 15	
1.6	How This Book Is Organized 16	

Summary	18
Background	19
Exercises	20

## 2 SYMBOLIC PROGRAMMING

23

2.1	Rule-Based Reactive System Example	25
	Representing Sensors and Sensor Values as Symbols,	26
2.2	Introduction to Lisp	27
	Programming Language Requirements,	27
	Common Lisp,	27
	Lists and Lisp Syntax,	28
	Symbols,	28
	Programs and Documentation,	28
2.3	Interacting with Lisp	29
	The Lisp Interpreter,	29
2.4	Functions in Lisp	31
	Function Invocation,	31
	Procedural Abstraction,	32
	Conditional Statements,	33
	Recursive Functions,	35
	Evaluating Functions in Files,	35
2.5	Environments, Symbols, and Scope	36
	Assigning Values to Symbols,	36
	Eval and Apply Revisited,	37
	Structured Environments,	38
	Local Variables,	39
	Lexical Scoping,	40
2.6	More on Functions	42
	Functions with Local State,	42
	Lambda and Functions as Arguments,	43
2.7	List Processing	44
	Suspending Evaluation Using Quote,	44
	Building and Accessing Elements in Lists,	45
	Lists in Memory,	45
	Modifying List Structures in Memory,	46
	Alternative Parameter-Passing Conventions,	47
	Predicates on Lists,	48
	Built-In List Manipulation Functions,	48
	Optional Arguments,	49
	List-Processing Examples,	49
	Data Abstraction,	51
2.8	Iterative Constructs	53
	Mapping Functions to Arguments,	53

- General Iteration, 54
- Simple Iteration, 55
- 2.9 Monitoring and Debugging Programs 56
  - Tracing and Stepping Through Programs, 56
  - Formatted Output, 58
- 2.10 Rule-Based Reactive System Revisited 58
  - Summary 64
  - Background 65
  - Exercises 65

### 3 REPRESENTATION AND LOGIC

71

- 3.1 Propositional Logic 73
  - Syntax for  $\mathcal{P}$ , 74
  - Semantics for  $\mathcal{P}$ , 75
- 3.2 Formal System for  $\mathcal{P}$  76
  - Logical Axioms of  $\mathcal{P}$ , 77
  - Normal Forms, 78
  - Rules of Inference, 79
  - Proofs and Theorems, 79
  - Resolution Rule of Inference, 80
  - Completeness, Soundness, and Decidability, 81
  - Computational Complexity, 82
  - Solving Problems with Logic, 82
- 3.3 Automated Theorem Proving in  $\mathcal{P}$  84
  - Goal Reduction in  $\mathcal{P}$ , 85
  - Proof by Contradiction, 87
- 3.4 Predicate Calculus 88
  - Syntax for  $\mathcal{PC}$ , 89
  - Translating English Sentences into Logic, 90
  - More About Quantification, 91
  - Semantics for  $\mathcal{PC}$ , 91
- 3.5 Formal System for  $\mathcal{PC}$  93
  - Specifying Programs in Prolog, 94
  - Eliminating Quantifiers, 94
  - Learning and Deductive Inference, 96
  - Decidability, 98
- 3.6 Automated Theorem Proving in  $\mathcal{PC}$  99
  - Matching and Universal Instantiation, 99
  - Goal Reduction in  $\mathcal{PC}$ , 101
  - Unification, 103
  - Concept Description Languages, 107
  - Semantic Networks, 108



3.7	Nonmonotonic Logic	109
	Closed-World Assumption,	109
	Abductive and Default Reasoning,	111
	Minimal Models,	112
3.8	Deductive Retrieval Systems	113
	Forward and Backward Chaining,	114
	Reason Maintenance Systems,	116
	Nonmonotonic Data Dependencies,	118
	Summary	119
	Background	121
	Exercises	122
	Lisp Implementation: Data Dependencies	127

## 4 SEARCH

131

4.1	Basic Search Issues	133
	Search Spaces and Operators,	134
	Appliance Assembly Example,	135
	Exploiting Structure to Expedite Search,	136
4.2	Blind Search	137
	Depth-First Search,	138
	Depth-First Search Is Space Efficient,	139
	Breadth-First Search,	140
	Breadth-First Search Is Guaranteed,	141
	Iterative-Deepening Search,	141
	Iterative-Deepening Search Is Asymptotically Optimal,	143
	Searching in Graphs,	144
4.3	Heuristic Search	144
	Best-First Search,	145
	Admissible Evaluation Functions,	146
4.4	Optimization and Search	149
	Hill-Climbing Search,	149
	Local Minima and Maxima,	151
	Gradient Search,	153
	Simulated Annealing,	153
	Simulated Evolution and Genetic Algorithms,	154
	Application to Vehicle Routing,	158
4.5	Adversary Search	160
	Minimax Search,	160
	$\alpha$ - $\beta$ Search,	163
4.6	Indexing in Discrimination Trees	166
	Storing and Retrieving Predicate Calculus Formulas,	167
	Decision Trees,	168