HEURISTICS

Intelligent Search Strategies for Computer Problem Solving

Judea Pearl

HEURISTICS

Intelligent Search Strategies for Computer Problem Solving

Judea Pearl

Department of Computer Science
University of California
Los Angeles, California

ADDISON-WESLEY PUBLISHING COMPANY
Reading, Massachusetts • Menlo Park, California
London • Amsterdam • Don Mills, Ontario • Sydney

This book is in the Addison-Wesley Series in Artificial Intelligence.

Library of Congress Cataloging in Publication Data

Pearl, Judea.
Heuristics : intelligent search strategies for computer problem solving.

(Artificial intelligence series)
Bibliography: p.
Includes index.
1. Artificial intelligence. 2. Operations research 3. Problem-solving—Data processing. 4. Heuristic programming. I. Title. II. Series: Artificial intelligence series (Addison-Wesley Publishing Company)
Q335.P38 1984 001.53'5 83-12217
ISBN 0-201-05594-5

ABCDEFGHIJ-MA-8987654

Preface

This book is about *heuristics*, popularly known as rules of thumb, educated guesses, intuitive judgments or simply *common sense*. In more precise terms, heuristics stand for strategies using readily accessible though loosely applicable information to control problem-solving processes in human beings and machine. This book presents an analysis of the nature and the power of typical heuristic methods, primarily those used in artificial intelligence (AI) and operations research (OR) to solve problems of search, reasoning, planning and optimization on digital machines.

The discussions in this book follow a three-phase pattern: Presentation, characterization, and evaluation. We first present a set of general-purpose problem-solving strategies guided by heuristic information (Chapters 1 and 2), then highlight the general principles and properties that characterize this set (Chapters 3 and 8) and, finally, we present mathematical analyses of the performances of these strategies in several well-structured domains (Chapters 5, 6, 7, 9, and 10). Some psychological aspects of how people discover and use heuristics are discussed briefly in Chapters 1 and 4.

The original intention in writing this book was to provide a cohesive and hospitable package for the theoretical results obtained at the UCLA Cognitive Systems Laboratory in the past three years. Some of these results were scattered in various reports, proceedings, and archival journals, and others buried in notebooks awaiting a sufficient incentive for refinement and publication. The compilation of this body of research into a single volume under a consistent notation and a unified logical thread now offers readers easier access to the main results and a better opportunity to assess their range of applicability. These original works are covered in Chapters 5, 6, 7, 9 and 10 and the last sections of Chapters 3 and 8.

Under the persuasive influence of colleagues and students, I later broadened the scope of the book to include some introductory material: an overview of heuristics in typical problem-solving situations (Chapter 1), a description and taxonomy of the basic heuristic search strategies used in AI (Chapters 2 and 8), a formal exposition of their main properties (Chapter 3), and a general discussion on the nature of heuristics (Chapter 4).

As it now stands, the book can fulfill three roles—it can serve as a monograph, as a reference, or as a textbook. The researcher or practitioner familiar

with the standard literature on heuristic search can skip the introductory sections and turn directly to the theoretical topics, which are relatively self-contained. Casual readers, curious enough to gain an understanding of the type of problems and techniques emerging from the AI brewery, are advised to follow the introductory material at their own pace and select the advanced reading which best matches their tastes, backgrounds, and ultimate objectives. Special care was taken to adhere to traditional mathematical notation (avoiding programming-based jargons), thereby ensuring that readers from engineering, operations research, mathematics, or computer science will find the presentation familiar and comfortable. Finally, as a textbook, the book can be used as a reference for an introductory course in AI, and as a full text in a more advanced, graduate-level class on AI control strategies or the analysis of algorithms. At UCLA, for example, we have used Chapters 1, 2, 3 (Section 3.1), and 8 to cover roughly half of the first graduate course in artificial intelligence. The other chapters are used as a text in a graduate course on heuristic algorithms that serves the curriculum of two major fields: machine intelligence and computer science theory. These chapters should also be ideal for an operations research class that focuses on the taxonomy and analysis of combinatorial search techniques.

A large part of the material is presented in the language of mathematics, mainly that of elementary probability theory. It is assumed, therefore, that the reader is familiar with the basic concepts of events, random variables, distribution functions, expectations, and generating functions, which correspond to a first course in probability theory. Chapters 5 and 6 make use of some results from the theory of branching processes. These are described and summarized in Appendix 5-A, which readers are advised to read before proceeding with the text of Sections 5.4 and 6.3. Familiarity with graph theory can be helpful but is not essential because the concepts needed for our purposes are summarized in the introductory sections to Chapters 2 and 3.

The mathematical nature of our analyses should not by any means prevent the less mathematically trained reader from following the basic concepts and techniques, or from appreciating the impact of the main results. A conscious effort was made to introduce each topic in an intuitive—even metaphorical—style so that the presentation as a whole would be both technically correct and easy to follow. The "theorem-proof" structure of some chapters should not automatically suggest dryness. Most theorems either summarize the discussions that precede them, or facilitate valuable shortcuts in the discussions that follow. In all cases, however, I kept in mind that a theorem's primary mission is to resolve some curious dilemma, and it is in the description of these dilemmas that my main effort was invested.

The choice of algorithm description language deserves an explanatory remark. The commitment to make this book accessible to readers from all areas of the applied sciences has resulted in the following guideline: whenever possible algorithms should be as readable by nonprogrammers as the rest of the text. For that reason I have avoided using any of the existing formal

languages and, moreover, have purposely attempted to give algorithms an *informal*, proselike flavor. Along this line I have avoided an excessive use of recursions, have made repetitive references to the underlying data structures, have emphasized the purpose of steps and the meaning of the variables, and have consistently used *redundant* connecting phrases such as: "else continue," "as soon as," and others. I hope readers will agree that the improved legibility of these descriptions more than makes up for their "impure" appearance; besides, in the not very distant future computers too will be able to accept these natural descriptions and convert them into executable programs.

I take great pleasure in closing this preface by acknowledging those who assisted me with this book. First, I would like to thank the members of the Cognitive Systems Laboratory at UCLA upon whose works and ideas many of the sections are based; Rina Dechter, Nam Huyn, Jin Kim, Scott Kurman, Gerard Michon, Igor Roizen, Michael Tarsi, Chuck Siska, and Rick Verstraete. Acknowledgment is due the National Science Foundation for sponsoring the research that led to many of these results. My academic and professional colleagues have all been very supportive of this endeavor: Nils Nilsson has provided continuous encouragement and Richard Karp is responsible for everything I know about branching processes, especially the ideas of Section 5.4. The comments of John McCarthy, Donald Michie, Allen Newell, and Herbert Simon helped improve the presentation of many sections, especially those relating the historical developments of the key ideas. The largest portion of this book was typed and corrected on a UNIX system by Lorna Freeman, assisted by Tovah Hollander, Terry Peters and Anna Gibbons. Last, I owe a great debt of thanks to my family for their support and understanding: to Danny for insightful suggestions on the first draft, to Michelle for her keen proofreading, and especially to my wife Ruth for tolerating, supporting, and encouraging my longest project since our wedding day.

Tel Aviv, Israel
January 1984

J.P.

Prologue

The study of heuristics draws its inspiration from the ever-amazing observation of how much people can accomplish with that simplistic, unreliable information source known as *intuition*. We drive our cars with hardly any thought of how they function and only a vague mental picture of the road conditions ahead. We write complex computer programs while attending to only a fraction of the possibilities and interactions that may take place in the actual execution of these programs. Even more surprisingly, we maneuver our way successfully in intricate social situations having only a guesswork expectation of the behavior of other persons around and even less certainty of their expectations of us. Yet, when these expectations fail we are able to master the great power of humor and recover gracefully. No computer program has yet been designed that exhibits such capabilities. Evidently, the little information we possess is so effectively organized that its poor state of reliability hardly hinders our normal, everyday activities.

Early attempts to equip machines with humanlike intelligence have quickly revealed that machines, too, cannot possibly be expected to operate with only precise and detailed knowledge of their task environment. Rather, this knowledge must, in some way, be summarized or abstracted so as to provide, like human intuition, a steady stream of tentative, unpolished, yet swift and informative *advice* for managing the primitive computational steps that make up a problem-solving process. The information content of this advice came to be known as *heuristic knowledge*. The science of heuristics, then, comprises both empirical and theoretical studies aimed toward understanding the workings of heuristic knowledge; how it is acquired, stored, and used by people, how it can be represented and utilized by machines, and what makes one heuristic succeed where others fail. This book focuses on the latter two topics.

Contents

PART III
Game-Playing Programs

PART I

Problem-Solving Strategies and the Nature of Heuristic Information

Heuristics! Patient rules of thumb,
So often scorned: Sloppy! Dumb!
Yet, slowly, common sense become.

(ODE TO AI)

By the pricking of my thumbs,
Something wicked this way comes.

(MACBETH)

Chapter 1

Heuristics and Problem Representations

1.1 TYPICAL USES OF HEURISTICS IN PROBLEM SOLVING

Heuristics are criteria, methods, or principles for deciding which among several alternative courses of action promises to be the most effective in order to achieve some goal. They represent compromises between two requirements: the need to make such criteria simple and, at the same time, the desire to see them discriminate correctly between good and bad choices.

A heuristic may be a **rule of thumb** that is used to guide one's actions. For example, a popular method for choosing ripe cantaloupe involves pressing the spot on the candidate cantaloupe where it was attached to the plant, and then smelling the spot. If the spot smells like the inside of a cantaloupe, it is most probably ripe. This rule of thumb does not guarantee choosing only ripe cantaloupe, nor does it guarantee recognizing each ripe cantaloupe judged, but it is effective most of the time.

As an example of a less clear-cut use of heuristics, consider a chess grand master who is faced with a choice of several possible moves. The grand master may decide that a particular move is most effective because that move results in a board position that "appears" stronger than the positions resulting from the other moves. The criterion of "appearing" stronger is much simpler for the grand master to apply than, say, rigorously determining which move or moves forces a checkmate. The fact that grand masters do not always win indicates that their heuristics do not guarantee selecting the most effective move. Finally, when asked to describe their heuristics, grand masters can only give partial and rudimentary descriptions of what they themselves seem to apply so effortlessly.

Even the decision to begin reading this book reflects a tacit use of heuristics—heuristics that have led the reader to expect greater benefit from engaging in this activity rather than doing something else at this point in time.

It is the nature of good heuristics both that they provide a simple means of indicating which among several courses of action is to be preferred, and that they are not necessarily guaranteed to identify the most effective course of action, but do so sufficiently often.

Most complex problems require the evaluation of an immense number of possibilities to determine an exact solution. The time required to find an exact solution is often more than a lifetime. Heuristics play an effective role in such problems by indicating a way to reduce the number of evaluations and to obtain solutions within reasonable time constraints.

To illustrate the role of heuristics in problem solving, we make use of five puzzlelike problems which have served as expository tools in both psychology and artificial intelligence (AI). The expository power of puzzles and games stems from their combined *richness* and *simplicity.* If we were to use examples taken from complex real-life problems, it would take us more than a few pages just to lay the background and specify the situation in sufficient detail so that the reader could appreciate the solution method discussed. Moreover, even after making such an effort we cannot be sure that different readers, having accumulated different experiences, would not obtain diversely different perceptions of the problem situation described. Games and puzzles, on the other hand, can be stated with sufficient precision so as to lay a common ground for discussion using but a few sentences and, yet, their behavior is sufficiently rich and unpredictable to simulate the complexities encountered in real-life situations. This, in fact, is what makes games and puzzles so addictive.

1.1.1 The 8-Queens Problem

To solve this problem, one must place eight queens on a chess board such that no queen can attack another. This is equivalent to placing the queens so that no row, column, or diagonal contains more than one queen.

We can attempt to solve this problem in many ways, from consulting a mystic to looking up the solution in a puzzle book. However, a more constructive method of attack would be to forgo hopes of obtaining the final solution in one step and, instead, to attempt reaching a solution in an **incremental,** step-by-step manner, in much the same way that a treasure hunter "closes in" on a target following a series of local decisions, each based on new information gathered along the way. We can start, for example, with an arbitrary arrangement of eight queens on the board and transform the initial arrangement iteratively, going from one board configuration to another, until the eight queens are adequately dispersed. As with the treasure-hunt, however, we must also make sure that the sequence of transformations is not random but **systematic** so that we do not generate the same configuration over and over and so that we do not miss the opportunity of generating the desired configuration. The former is a requirement of efficiency, whereas the latter is one of utility.

One way to systematize the search is to attempt to **construct** (rather than

transform) board configurations in a step-by-step manner. Starting with the empty board we may attempt to place the queens on the board one at a time, ruling out violations of the problem constraints, until a satisfactory configuration with eight queens is finally achieved. Moreover, because we can easily deduce from the problem specification that there must be exactly one queen in every row, we can assign the first queen to the first row, the second queen to the second row, and so on, thus reducing the number of alternative positions considered for each new queen to less than 8. The feature that allows us to systematize the search in this manner is the fact that we cannot recover from constraint violations by future operations. Once a partial configuration violates any of the problem constraints (i.e., it contains two or more attacking queens), it cannot be rectified by adding more queens to the board, and so many useless configurations can be eliminated at an early stage of the search process.

Assume now that at some stage of the search, the first three queens were positioned in the cells marked by Q's in Figure 1.1 and that we must decide where to position the fourth queen, in cell A, B, or C. The role of heuristics in this context would be to provide rule-of-thumb criteria for deciding, at least tentatively, which of the three positions appears to have the highest chance of leading to a satisfactory solution.

In devising such heuristics, we may reason that to be able to place all the eight queens we need to leave as many options as possible for future additions. That means we need to pay attention to the number of cells in the unfilled rows that remain unattacked by the previously placed queens. A candidate placement is to be preferred if it leaves the highest **number of unattacked cells** in the

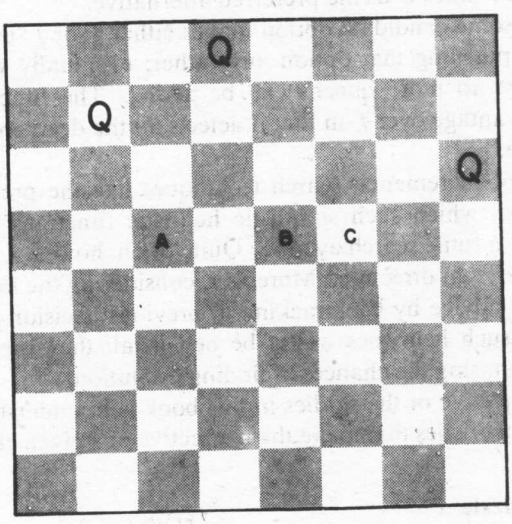

Figure 1.1
A typical decision step in constructing a solution to the 8-Queens problem.