Artificial Intelligence: Foundations, Theory, and Algorithms

Series editors

Barry O’Sullivan, Cork, Ireland
Michael Wooldridge, Oxford, UK

More information about this series at http://www.springer.com/series/13900

David Bergman · Andre A. Cire
Willem-Jan van Hoeve · John Hooker

Decision Diagrams for Optimization

Springer

David Bergman
Department of Operations and Information Management, School of Business
University of Connecticut
Storrs, CT
USA

Andre A. Cire
Department of Management, UTSC
University of Toronto
Toronto, ON
Canada

Willem-Jan van Hoeve
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA
USA

John Hooker
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA
USA

ISSN 2365-3051 ISSN 2365-306X (electronic)
Artificial Intelligence: Foundations, Theory, and Algorithms
ISBN 978-3-319-42847-5 ISBN 978-3-319-42849-9 (eBook)
DOI 10.1007/978-3-319-42849-9

Library of Congress Control Number: 2016953636

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

Foreword

This book provides an excellent demonstration of how the concepts and tools of one research community can cross into another, yielding powerful insights and ideas.

Early work on decision diagrams focused on modeling and verifying properties of digital systems, including digital circuits and abstract protocols. Decision diagrams (DDs) provided a compact representation of these systems and a useful data structure for algorithms to construct these representations and to answer queries about them. Fundamentally, though, they were used to solve problems having yes/no answers, such as: "Is it possible for the system to reach a deadlocked state?", or "Do these two circuits compute the same function?"

Using DDs for optimization introduces an entirely new set of possibilities and challenges. Rather than just finding some satisfying solution, the program must find a "best" solution, based on some objective function. Researchers in the digital systems and verification communities recognized that, given a DD representation of a solution space, it is easy to count the number of solutions and to find an optimal solution based on very general classes of objective functions. But, it took the skill of leading experts in optimization, including the authors of this book, to fully expand DDs into a general-purpose framework for solving optimization problems.

The authors show how the main strategies used in discrete optimization, including problem relaxation, branching search, constraint propagation, primal solving, and problem-specific modeling, can be adapted and cast into a DD framework. DDs become a data structure for managing the entire optimization process: finding upper bounds and feasible solutions, storing solutions to subproblems, applying global constraints, and guiding further search. They are especially effective for solving problems that fare poorly with traditional optimization techniques, including linear programming relaxation, such as ones having combinatorial constraints and non-convex cost functions.

Over the 10 years in which the authors have been developing these ideas, they have transformed DDs well beyond what has been used by the verification community. For example, whereas most DD-based verification techniques build the representations from the bottom up, based on a symbolic execution of the system description, the authors build their DDs from the top down, based on a direct encoding of the solution space. Some of the ideas presented here have analogs in the verification world, for example the idea of restricted and relaxed DDs are similar to the abstraction-refinement approaches used in verification. Others, however, are strikingly new. Perhaps some of these ideas could be transferred back to the verification community to increase the complexity and classes of systems they can verify.

Pittsburgh, USA, August 2016 *Randal E. Bryant*