

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

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

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

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

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

© Springer International Publishing Switzerland 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

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

1	Introduction	1
1.1	Motivation for the Book	1
1.2	A New Solution Technology	2
1.3	An Example	4
1.4	Plan of the Book	8
2	Historical Overview	11
2.1	Introduction	11
2.2	Origins of Decision Diagrams	12
2.3	Decision Diagrams in Optimization	15
2.3.1	Early Applications	15
2.3.2	A Discrete Optimization Method	16
2.3.3	Decision Diagrams in Constraint Programming	17
2.3.4	Relaxed Decision Diagrams	18
2.3.5	A General-Purpose Solver	19
2.3.6	Markov Decision Processes	21
3	Exact Decision Diagrams	23
3.1	Introduction	23
3.2	Basic Definitions	24
3.3	Basic Concepts of Decision Diagrams	25
3.4	Compiling Exact Decision Diagrams	27
3.4.1	Dynamic Programming	28
3.4.2	Top-Down Compilation	30
3.5	Maximum Independent Set Problem	32

3.6	Set Covering Problem	34
3.7	Set Packing Problem	37
3.8	Single-Machine Makespan Minimization	39
3.9	Maximum Cut Problem	42
3.10	Maximum 2-Satisfiability Problem	44
3.11	Compiling Decision Diagrams by Separation	46
3.12	Correctness of the DP Formulations	50
4	Relaxed Decision Diagrams	55
4.1	Introduction	55
4.2	Top-Down Compilation of Relaxed DDs	57
4.3	Maximum Independent Set	59
4.4	Maximum Cut Problem	60
4.5	Maximum 2-Satisfiability Problem	63
4.6	Computational Study	64
4.6.1	Merging Heuristics	64
4.6.2	Variable Ordering Heuristic	65
4.6.3	Bounds vs. Maximum BDD Width	66
4.6.4	Comparison with LP Relaxation	67
4.7	Compiling Relaxed Diagrams by Separation	74
4.7.1	Single-Machine Makespan Minimization	76
5	Restricted Decision Diagrams	83
5.1	Introduction	83
5.2	Top-Down Compilation of Restricted DDs	85
5.3	Computational Study	86
5.3.1	Problem Generation	87
5.3.2	Solution Quality and Maximum BDD Width	89
5.3.3	Set Covering	90
5.3.4	Set Packing	92
6	Branch-and-Bound Based on Decision Diagrams	95
6.1	Introduction	95
6.2	Sequential Branch-and-Bound	96
6.3	Exact Cutsets	97
6.4	Enumeration of Subproblems	98
6.4.1	Exact Cutset Selection	100

6.5	Computational Study	100
6.5.1	Results for the MISP	101
6.5.2	Results for the MCP	104
6.5.3	Results for MAX-2SAT	108
6.6	Parallel Branch-and-Bound	109
6.6.1	A Centralized Parallelization Scheme	112
6.6.2	The Challenge of Effective Parallelization	113
6.6.3	Global and Local Pools	113
6.6.4	Load Balancing	114
6.6.5	DDX10: Implementing Parallelization Using X10	116
6.6.6	Computational Study	116
7	Variable Ordering	123
7.1	Introduction	123
7.2	Exact BDD Orderings	125
7.3	Relaxed BDD Orderings	130
7.4	Experimental Results	131
7.4.1	Exact BDDs for Trees	132
7.4.2	Exact BDD Width Versus Relaxation BDD Bound	132
7.4.3	Relaxation Bounds	134
8	Recursive Modeling	137
8.1	Introduction	137
8.2	General Form of a Recursive Model	139
8.3	Examples	141
8.3.1	Single-Machine Scheduling	141
8.3.2	Sequence-Dependent Setup Times	142
8.3.3	Minimum Bandwidth	144
8.4	State-Dependent Costs	146
8.4.1	Canonical Arc Costs	147
8.4.2	Example: Inventory Management	151
8.5	Nonserial Recursive Modeling	153
9	MDD-Based Constraint Programming	157
9.1	Introduction	157
9.2	Constraint Programming Preliminaries	158
9.3	MDD-Based Constraint Programming	164

9.3.1	MDD Propagation	166
9.3.2	MDD Consistency	167
9.3.3	MDD Propagation by Intersection	169
9.4	Specialized Propagators	173
9.4.1	Equality and Not-Equal Constraints	173
9.4.2	Linear Inequalities	174
9.4.3	Two-Sided Inequality Constraints	174
9.4.4	ALLDIFFERENT Constraint	175
9.4.5	AMONG Constraint	176
9.4.6	ELEMENT Constraint	177
9.4.7	Using Conventional Domain Propagators	178
9.5	Experimental Results	178
10	MDD Propagation for SEQUENCE Constraints	183
10.1	Introduction	183
10.2	MDD Consistency for SEQUENCE Is NP-Hard	186
10.3	MDD Consistency for SEQUENCE Is Fixed Parameter Tractable	189
10.4	Partial MDD Filtering for SEQUENCE	190
10.4.1	Cumulative Sums Encoding	191
10.4.2	Processing the Constraints	192
10.4.3	Formal Analysis	194
10.5	Computational Results	196
10.5.1	Systems of SEQUENCE Constraints	198
10.5.2	Nurse Rostering Instances	201
10.5.3	Comparing MDD Filtering for SEQUENCE and AMONG	203
11	Sequencing and Single-Machine Scheduling	205
11.1	Introduction	205
11.2	Problem Definition	207
11.3	MDD Representation	208
11.4	Relaxed MDDs	211
11.5	Filtering	214
11.5.1	Filtering Invalid Permutations	214
11.5.2	Filtering Precedence Constraints	215
11.5.3	Filtering Time Window Constraints	215
11.5.4	Filtering Objective Function Bounds	216
11.6	Inferring Precedence Relations from Relaxed MDDs	218

11.7 Refinement	219
11.8 Encoding Size for Structured Precedence Relations	221
11.9 Application to Constraint-Based Scheduling	222
11.9.1 Experimental Setup	223
11.9.2 Impact of the MDD Parameters	224
11.9.3 Traveling Salesman Problem with Time Windows	227
11.9.4 Asymmetric Traveling Salesman Problem with Precedence Constraints	228
11.9.5 Makespan Problems	230
11.9.6 Total Tardiness	233
References	235
Index	249

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