

Instance-Specific Algorithm Configuration

Yuri Malitsky

Instance-Specific Algorithm Configuration

 Springer

Yuri Malitsky
IBM Thomas J. Watson Research Center
Yorktown Heights
New York
USA

ISBN 978-3-319-11229-9 ISBN 978-3-319-11230-5 (eBook)

DOI 10.1007/978-3-319-11230-5

Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014956556

© Springer International Publishing Switzerland 2014

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. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

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.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

When developing a new heuristic or complete algorithm for a constraint satisfaction or constrained optimization problem, we frequently face the problem of choice. There may be multiple branching heuristics that we can employ, different types of inference mechanisms, various restart strategies, or a multitude of neighborhoods from which to choose. Furthermore, the way in which the choices we make affect one another is not readily perceptible. The task of making these choices is known as algorithm configuration.

Developers often make many of these algorithmic choices during the prototyping stage. Based on a few preliminary manual tests, certain algorithmic components are discarded even before all the remaining components have been implemented. However, by making the algorithmic choices beforehand developers may unknowingly discard components that are used in the optimal configuration. In addition, the developer of an algorithm has limited knowledge about the instances that a user will typically employ the solver for. That is the very reason why solvers have parameters: to enable users to fine-tune a solver for their specific needs.

Alternatively, manually tuning a parameterized solver can require significant resources, effort, and expert knowledge. Before even trying the numerous possible parameter settings, the user must learn about the inner workings of the solver to understand what each parameter does. Furthermore, it has been shown that manual tuning often leads to highly inferior performance.

This book shows how to automatically train a multi-scale, multi-task approach for enhanced performance based on machine learning techniques. In particular this work presents the methodology of Instance-Specific Algorithm Configuration (ISAC). ISAC is a general configurator that focuses on tuning different categories of parameterized solvers according to the instances they will be applied to. Specifically, this book shows that the instances of many problems can be decomposed into a representative vector of features. It further shows that instances with similar features often cause similar behavior in the applied algorithm. ISAC exploits this observation by automatically detecting the different subtypes of a problem and then training a solver for each variety. This technique is explored on a number of problem domains, including set covering, mixed integer, satisfiability,

and set partitioning. ISAC is then further expanded to demonstrate its application to traditional algorithm portfolios and adaptive search methodologies. In all cases, marked improvements are shown over the existing state-of-the-art solvers. These improvements were particularly evident during the 2011 SAT Competition, where a solver based on ISAC won seven medals, including a gold in the handcrafted instance category, and another gold in the randomly generated instance category. Later, in 2013, solvers based on this research won multiple gold medals in both the SAT Competition and the MaxSAT Evaluation.

The research behind ISAC is broken down into ten chapters. Chapters 1 and 2, respectively, introduce the problem domain and the relevant research that has been done on the topic. Chapter 3 then introduces the ISAC methodology, while Chap. 4 demonstrates its effectiveness in practice. Chapters 5, 6, and 7 demonstrate how the methodology can be applied to the problem of algorithm selection. Chapter 8 takes an alternate view and shows how ISAC can be used to create a framework that dynamically switches to the best heuristic to utilize as the problem is being solved. Chapter 9 introduces the concept that sometimes problems change over time, and that a portfolio needs a way to effectively retrain to accommodate the changes. Chapter 10 demonstrates how the ISAC methodology can be transparently expanded, while Chap. 11 wraps up the topics covered and offers ideas for future research.

The research presented in this book was carried out as the author's Ph.D. work at Brown University and his continuing work at the Cork Constraint Computation Centre. It would not have been possible without the collaboration with my advisor, Meinolf Sellmann, and my supervisor, Barry O'Sullivan. I would also like to thank all of my coauthors, who have helped to make this research possible. In alphabetical order they are: Tinus Abell, Carlos Ansotegui, Marco Collautti, Barry Hurley, Serdar Kadioglu, Lars Kotthoff, Christian Kroer, Kevin Leo, Giovanni Di Liberto, Deepak Mehta, Ashish Sabharwal, Horst Samulowitz, Helmut Simonis, and Kevin Tierney.

This work has also been partially supported by Science Foundation Ireland Grant No. 10/IN.1/I3032 and by the European Union FET grant (ICON) No. 284715.

Cork, Ireland
September 2013

Yuri Malitsky

Contents

1	Introduction	1
1.1	Outline	4
2	Related Work	7
2.1	Algorithm Construction	7
2.2	Instance-Oblivious Tuning	9
2.3	Instance-Specific Regression	11
2.4	Adaptive Methods	13
2.5	Chapter Summary	14
3	Instance-Specific Algorithm Configuration	15
3.1	Clustering the Instances	16
3.1.1	Motivation	16
3.1.2	Distance Metric	16
3.1.3	k-Means	18
3.1.4	g-Means	19
3.2	Training Solvers	20
3.2.1	Local Search	20
3.2.2	GGA	22
3.3	ISAC	23
3.4	Chapter Summary	24
4	Training Parameterized Solvers	25
4.1	Set Covering Problem	25
4.1.1	Solvers	26
4.1.2	Numerical Results	27
4.2	Mixed Integer Programming	30
4.2.1	Solver	32
4.2.2	Numerical Results	32
4.3	SAT	33
4.3.1	Solver	34
4.3.2	Numerical Results	35

4.4	Machine Reassignment	36
4.4.1	Solver	37
4.4.2	Numerical Results	38
4.5	Chapter Summary	40
5	ISAC for Algorithm Selection	41
5.1	Using ISAC as Portfolio Generator.....	42
5.2	Algorithm Configuration vs. Algorithm Selection of SAT Solvers	42
5.2.1	Pure Solver Portfolio vs. SATzilla	44
5.2.2	Meta-Solver Configuration vs. SATzilla.....	45
5.2.3	Improved Algorithm Selection	46
5.2.4	Latent-Class Model-Based Algorithm Selection	47
5.3	Comparison with Other Algorithm Configurators	48
5.3.1	ISAC vs. ArgoSmart.....	49
5.3.2	ISAC vs. Hydra	50
5.4	Chapter Summary	53
6	Dynamic Training	55
6.1	Instance-Specific Clustering	55
6.1.1	Nearest Neighbor-Based Solver Selection.....	56
6.1.2	Improving Nearest Neighbor-Based Solver Selection ...	58
6.2	Building Solver Schedules	61
6.2.1	Static Schedules	62
6.2.2	A Column Generation Approach	62
6.2.3	Dynamic Schedules.....	65
6.2.4	Semi-static Solver Schedules	66
6.2.5	Fixed-Split Selection Schedules	67
6.3	Chapter Summary	68
7	Training Parallel Solvers.....	71
7.1	Parallel Solver Portfolios	72
7.1.1	Parallel Solver Scheduling	72
7.1.2	Solving the Parallel Solver Scheduling IP.....	74
7.1.3	Minimizing Makespan and Post-processing the Schedule	74
7.2	Experimental Results.....	75
7.2.1	Impact of the IP Formulation and Neighborhood Size.....	76
7.2.2	Impact of Parallel Solvers and the Number of Processors	77
7.2.3	Parallel Solver Selection and Scheduling vs. the State of the Art.....	78
7.3	Chapter Summary	81

- 8 Dynamic Approach for Switching Heuristics** 83
 - 8.1 Learning Dynamic Search Heuristics 84
 - 8.2 Boosting Branching in Cplex for MIP 85
 - 8.2.1 MIP Features 86
 - 8.2.2 Branching Heuristics 86
 - 8.2.3 Dataset 88
 - 8.3 Numerical Results 88
 - 8.4 Chapter Summary 91
- 9 Evolving Instance-Specific Algorithm Configuration** 93
 - 9.1 Evolving ISAC 94
 - 9.1.1 Updating Clusters 96
 - 9.1.2 Updating Solver Selection for a Cluster 97
 - 9.2 Empirical Results 100
 - 9.2.1 SAT 100
 - 9.2.2 MaxSAT 103
 - 9.3 Chapter Summary 105
- 10 Improving Cluster-Based Algorithm Selection** 107
 - 10.1 Benchmark 108
 - 10.1.1 Dataset and Features 108
 - 10.2 Motivation for Clustering 110
 - 10.3 Alternate Clustering Techniques 112
 - 10.3.1 X-Means 112
 - 10.3.2 Hierarchical Clustering 112
 - 10.4 Feature Filtering 113
 - 10.4.1 Chi-Squared 113
 - 10.4.2 Information Theory-Based Methods 114
 - 10.5 Numerical Results 114
 - 10.6 Extending the Feature Space 117
 - 10.7 SNNAP 119
 - 10.7.1 Choosing the Distance Metric 120
 - 10.7.2 Numerical Results 121
 - 10.8 Chapter Summary 123
- 11 Conclusion** 125
- References** 129