Studies in Computational Intelligence

Volume 805

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

The books of this series are submitted to indexing to Web of Science, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink.

More information about this series at http://www.springer.com/series/7092

Bartłomiej Jacek Kubica

Interval Methods for Solving Nonlinear Constraint Satisfaction, Optimization and Similar Problems

From Inequalities Systems to Game Solutions

Springer

Bartłomiej Jacek Kubica
Department of Applied Informatics
Warsaw University of Life Sciences
Warsaw, Poland

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-3-030-13794-6 ISBN 978-3-030-13795-3 (eBook)
https://doi.org/10.1007/978-3-030-13795-3

Library of Congress Control Number: 2019931826

© Springer Nature Switzerland AG 2019
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. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

To my beloved son, Stefanek, the joy of my life.

Preface

This book contains recent research on interval methods for solving nonlinear constraint satisfaction, optimization and similar problems. It presents a comprehensive survey of applications: it includes various branches of robotics, artificial intelligence systems, economy, control theory, dynamical systems theory and others. The book is completed with three Appendices, describing the notation, representation of numbers, used as intervals' endpoints and example implementations of the interval data type in a few programming languages.

Warsaw, Poland Bartłomiej Jacek Kubica

Acknowledgements

The author is grateful to everyone, who helped him in the process of preparation of this monograph.

Many thanks to Prof. Roman Wyrzykowski for all his help and for providing the `MICLab` project, crucial in performing several of the practical experiments.

I would like to acknowledge my friend and 'closest approximation to mentor I have ever had', Adam Woźniak.

Thanks to Prof. Sergey P. Shary, my great friend, and his wife, Irene A. Sharaya, who, unfortunately, is no longer between us.

Particular thanks to my Mother, my Wife and all my family.

And—last, but not least—thanks to my Lord, Jesus Christ, because without Him we can do nothing (John 15, 5).

Contents

1 Introduction 1
References 3

2 Interval Calculus 5
2.1 Introduction 5
2.2 Basics of Interval Computations 5
2.3 Operations on Intervals 6
2.3.1 Interval Arithmetic 6
2.3.2 Interval Enclosures of Other Operations and Functions 7
2.3.3 Auxiliary Operations 7
2.4 Properties and Features of the Interval Calculus 8
2.5 Interval Extension of a Function 8
2.5.1 Most Common Forms of Interval Extensions 9
2.5.2 How to Construct Formulae for Interval-Valued Functions? 9
2.6 Comparison of Intervals 10
2.7 A Metric on the Space of Intervals 13
2.8 Open or Closed Intervals? 13
2.9 Purposes of the Interval Calculus 14
References 15

3 Bounding Derivatives by Algorithmic Differentiation 17
3.1 Interval Algorithms and Derivatives Computation 17
3.1.1 Basic Approaches 17
3.2 Algorithmic Differentiation 18
3.3 Implementation of AD 19
3.3.1 Forward AD with Operator Overloading 19
3.3.2 Forward AD with Dual Numbers 19
3.3.3 Reverse Mode AD 20

3.4 State-of-the-Art Libraries . . . 21
3.4.1 ADHC Library . . . 22
3.4.2 Computing Arbitrary Many Derivatives . . . 24
3.5 Summary . . . 25
References . . . 25

4 Branch-and-Bound-Type Methods . . . 27
4.1 Preliminary Remarks . . . 27
4.2 Introduction . . . 27
4.3 The Solution Set . . . 29
4.4 Generic Algorithm . . . 31
4.5 Analysis of the B&BT Algorithm . . . 33
4.6 The Second Phase—Quantifier Elimination . . . 34
4.6.1 Herbrand Expansion . . . 34
4.6.2 Shared Quantities . . . 36
4.6.3 Existentially Quantified Formulae . . . 37
4.6.4 When is the Second Phase Not Necessary? . . . 38
4.7 Necessary Conditions . . . 38
4.8 Seeking Local Optima of a Function . . . 39
4.9 Example Heuristics . . . 41
4.10 Conclusions . . . 42
References . . . 43

5 Solving Equations and Inequalities Systems Using Interval B&Bt Methods . . . 47
5.1 Constraint Satisfaction Problems . . . 47
5.2 Solving Systems of Nonlinear Equations . . . 48
5.3 Interval Newton Operators . . . 50
5.4 Other Verification Tests . . . 53
5.4.1 Miranda Test . . . 53
5.4.2 Using Quadratic Approximation . . . 53
5.4.3 Borsuk Test . . . 54
5.4.4 Computing Topological Degree . . . 54
5.4.5 Obstruction Theory Test . . . 55
5.5 Consistency Enforcing . . . 56
5.5.1 Hull-Consistency . . . 56
5.5.2 Box-Consistency . . . 58
5.5.3 Higher-Order Consistencies . . . 59
5.6 Heuristics for Choosing and Parameterizing the Tools . . . 61
References . . . 62

6 Solving Quantified Problems Using Interval Methods 65
6.1 Interval Global Optimization 65
6.1.1 Branch-and-Bound Algorithm 66
6.1.2 Processing a Box in Interval Global Optimization 69
6.2 Pareto Sets of Multicritria Problems 70
6.2.1 Tools 72
6.3 Game Solutions 73
6.3.1 Algorithm 74
6.3.2 Tools 75
6.4 Summary 75
References 75

7 Parallelization of B&BT Algorithms 79
7.1 Introduction 79
7.2 Generic Algorithm 79
7.3 Basic Implementation Details 81
7.3.1 Data Structures 81
7.3.2 Memory Management 81
7.4 Parallelization of the B&BT Algorithm 82
7.5 Shared Memory Implementations 82
7.5.1 Storage of L 83
7.5.2 Storage of L_{ver} and L_{pos} 83
7.5.3 Shared Quantities 84
7.6 Distributed Memory Implementations 85
7.6.1 Load Balancing 85
7.6.2 Termination Detection 85
7.6.3 Advanced Issues 86
7.7 Parallelization of Rejection/Reduction Tests 87
7.7.1 Parallelization of Existence Tests 87
7.7.2 Modern Architectures 88
7.8 Summary 88
References 88

8 Interval Software, Libraries and Standards 91
8.1 Main Issues in Implementing Interval Libraries 91
8.1.1 IEEE 754 Standard 92
8.2 C-XSC 92
8.2.1 Basic Types 93
8.2.2 The Use of BLAS 93
8.2.3 The Toolbox and Additional Software 93
8.2.4 Author's Solvers and Libraries 94
8.3 Other Libraries 94
8.3.1 PROFIL/BIAS 94
8.3.2 Boost::Interval 94

8.3.3 Other Packages . . . 95
8.3.4 GPU Libraries . . . 96
8.3.5 IEEE Standard 1788–2015: Standard for Interval Arithmetic . . . 96
References . . . 98

9 Applications of Interval B&BT Methods . . . 101
9.1 Introduction . . . 101
9.2 Robotics . . . 101
9.2.1 Manipulator Kinematics . . . 102
9.2.2 Mobile Robots . . . 103
9.2.3 Path Planning . . . 106
9.3 Measurements and Estimation . . . 106
9.3.1 Parameter Estimation . . . 106
9.3.2 State Estimation . . . 107
9.3.3 Outliers . . . 107
9.3.4 Processing Statistical Samples Under Interval Uncertainty . . . 108
9.4 Artificial Intelligence Systems . . . 109
9.4.1 Neural Networks . . . 109
9.4.2 Support Vector Machines . . . 111
9.5 Control Theory . . . 112
9.5.1 Stability Checking . . . 113
9.5.2 Designing a Controller . . . 114
9.5.3 H_∞-Control . . . 115
9.5.4 Model-Predictive-Control . . . 115
9.6 Nonlinear Dynamics, Chaos and Differential Equations . . . 116
9.7 Economical Modeling and Multiagent Systems . . . 117
9.7.1 Economy Modeling . . . 117
9.7.2 Queueing Systems . . . 121
9.7.3 Decision Making . . . 123
9.8 Summary . . . 125
References . . . 126

Appendix A: Notation . . . 133

Appendix B: Standards for Numerical Computation . . . 135

Appendix C: Implementations of the Interval Class in Various Languages . . . 143

Solutions . . . 155

Acronyms

AD	Algorithmic (Automatic) Differentiation
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
B&B	Branch-and-Bound method
B&BT	Branch-and-Bound-type method
B&P	Branch-and-Prune method
BC	Box-Consistency
CLR	Common Language Runtime (the .NET virtual machine)
CPU	Central Processing Unit
CSP	Constraint Satisfaction Problem
DBN	Deep Belief Network
DP	Dynamic Programming
FPU	Floating-Point Unit
GPU	Graphics Processing Unit
GS	Gauss–Seidel
HC	Hull-Consistency
HIBA_USNE	Heuristical Interval Branch-and-prune Algorithm for Underdetermined and well-determined Systems of Nonlinear Equations
IRR	Internal Rate of Return
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
MCDM	Multi-Criteria Decision-Making
MPC	Model Predictive Control
MVA	Mean-Value Analysis
NaN	Not-a-Number or Not-any-Number
NE	Nash Equilibrium/Equilibria
NPV	Net Present Value
ODE	Ordinary Differential Equations
PDE	Partial Differential Equations

PDF	Probability Density Function
PID	Proportional–Integral–Derivative controller
PPS	Partitioning Parameter Space
RPC	Remote Procedure Call
SIVIA	Set Inversion Via Interval Analysis
SLAM	Simultaneous Localization and Mapping
SNE	Strong Nash Equilibrium/Equilibria
SVM	Support Vector Machine
TBB	Threading Building Blocks
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
VaR	Value at Risk
XML	eXtensible Markup Language

List of Figures

Fig. 2.1 Comparison of intervals $x = [0, 4]$ and $y = [1, 5]$ Example (2.3) 11
Fig. 2.2 Comparison of intervals $x = [0, 5]$ and $y = [1, 3]$ 11
Fig. 2.3 Comparison of intervals $x = [0, 3]$ and $y = [3, 5]$ 13
Fig. 4.1 Solution set consisting of separate points 30
Fig. 4.2 Solution set of measure zero—a manifold 30
Fig. 4.3 Solution set with a non-empty interior 30
Fig. 5.1 Illustration of the Newton operators: pointwise (left) and interval (right) ones 50
Fig. 5.2 Expression tree of constraint (5.8) 58
Fig. 9.1 Left: both feasible 3R manipulator configurations, right: three examples of uncountably many feasible 5R manipulator configurations 102
Fig. 9.2 The robot in an environment with regular obstacles 105
Fig. 9.3 The robot in an environment with irregular obstacles 105
Fig. 9.4 Left: the original map—the robot can determine, in which room it is, thanks to the presence of a pillar, center: an additional object may confuse the robot and make it think, it is in the other room, right: now, as the additional object has different shape than the pillar, the robot will hopefully realize that something is wrong—it should nowhere see what it does. 108
Fig. 9.5 A single neuron model 109
Fig. 9.6 An artificial neural network 110
Fig. 9.7 The "margin" between linearly separate sets—maximized by SVMs 111
Fig. 9.8 A system with negative feedback—closed loop control 114
Fig. 9.9 Control system 115
Fig. 9.10 Fuzzy set as a collection of nested intervals 119

Fig. 9.11 Open queueing network example 122
Fig. 9.12 Closed queueing network example 122
Fig. B.1 The IEEE 754 format 135
Fig. B.2 The unum 1.0 format 139
Fig. B.3 The unum 3.0 format 139