Lecture Notes in Computer Science 12735

More information about this subseries at http://www.palgrave.com/gp/series/7407

Peter J. Stuckey (Ed.)

Integration of Constraint Programming, Artificial Intelligence, and Operations Research

18th International Conference, CPAIOR 2021
Vienna, Austria, July 5–8, 2021
Proceedings

Springer

Editor
Peter J. Stuckey
Monash University
Melbourne, VIC, Australia

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-030-78229-0 ISBN 978-3-030-78230-6 (eBook)
https://doi.org/10.1007/978-3-030-78230-6

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

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

Preface

This volume contains the papers that were presented at the 18th International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR 2021), held in Vienna, Austria as a hybrid physical/virtual conference in response to the COVID-19 pandemic.

The conference received a total of 87 submissions, including 75 regular paper and 12 extended abstract submissions. The regular papers reflect original unpublished work, whereas the extended abstracts contain either original unpublished work or a summary of work that was published elsewhere. Each regular paper was reviewed by at least three Program Committee members. The reviewing phase was followed by an author response period and a general discussion by the Program Committee. The extended abstracts were reviewed for appropriateness for the conference. At the end of the review period, 30 regular papers were accepted for presentation during the conference and publication in this volume, and 6 abstracts were accepted for short presentation at the conference. Among the 30 regular papers, two were published directly in the journal Constraints via a fast-track review process. The abstracts of these papers can be found in this volume.

In addition to the regular papers and extended abstracts, three invited talks, whose abstracts and/or articles can be found in this volume, were given by Maya Gupta (Didero, USA), Adam Elmachtoub (Columbia University, USA), and Nikolaj Bjørner (Microsoft Research, USA).

The conference program included a Master Class on the topic "Explanation and Verification of Machine Learning Models" organized by Alexey Ignatiev and Nina Narodytska with invited talks by Alessio Lomuscio (Imperial College London, UK), Gagandeep Singh (University of Illinois Urbana-Champaign, USA), Guy Katz (Hebrew University of Jerusalem, Israel), Guy Van den Broeck (University of California, Los Angeles, USA), João Marques-Silva (CRNS, France), and Sameer Singh (University of California, Irvine, USA).

Of the regular papers accepted to the conference a committee comprising of myself, Helmut Simonis, and Louis-Martin Rousseau selected for the Best Paper Award the paper "Between Steps: Intermediate Relaxations between big-M and Convex Hull Formulations" by Jan Kronqvist, Ruth Misener, and Calvin Tsay and selected for the Best Student Paper Award the paper "Improving the filtering of Branch-and-Bound MDD Solver" by Xavier Gillard, Vianney Coppé, Pierre Schaus, and André Augusto Cire.

We acknowledge the generous support of our sponsors including, at the time of writing, the Vienna Center for Logic and Algorithms (VCLA), Artificial Intelligence Journal (AIJ), Springer, and TU Wien.

July 2021 Peter J. Stuckey

Organization

Program Chair

Peter J. Stuckey	Monash University, Australia

Conference Chair

Nysret Musliu	TU Wien, Austria

Master Class Chairs

Alexey Ignatiev	Monash University, Australia
Nina Narodytska	VMWare Research, USA

Program Committee

Fahiem Bacchus	University of Toronto, Canada
Chris Beck	University of Toronto, Canada
Nicolas Beldiceanu	LS2N, IMT Atlantique, France
Jeremias Berg	University of Helsinki, Finland
Armin Biere	Johannes Kepler University Linz, Austria
Mats Carlsson	RISE Research Institutes of Sweden, Sweden
Andre Augusto Cire	University of Toronto, Canada
Carleton Coffrin	Los Alamos National Laboratory, USA
Emir Demirović	Delft University of Technology, the Netherlands
Bistra Dilkina	University of Southern California, USA
Ambros Gleixner	HTW Berlin and Zuse Institute Berlin, Germany
Tias Guns	KU Leuven, Belgium
Emmanuel Hebrard	LAAS, CNRS, France
Philip Kilby	Data61 and the Australian National University, Australia
Joris Kinable	Amazon, USA
Zeynep Kiziltan	University of Bologna, Italy
Lars Kotthoff	University of Wyoming, USA
T. K. Satish Kumar	University of Southern California. USA
Jimmy Lee	The Chinese University of Hong Kong, Hong Kong
Michele Lombardi	University of Bologna, Italy
João Marques-Silva	IRIT, CNRS, France
Ian Miguel	University of St Andrews, Scotland
Nysret Musliu	TU Wien, Austria
Nina Narodytska	VMware Research, USA

Jakob Nordström	University of Copenhagen, Denmark, and Lund University, Sweden
Barry O'Sullivan	University College Cork, Ireland
Justin Pearson	Uppsala University, Sweden
Laurent Perron	Google, France
Claude-Guy Quimper	Université Laval, Canada
Louis-Martin Rousseau	Polytechnique Montréal, Canada
Domenico Salvagnin	University of Padova, Italy
Scott Sanner	University of Toronto, Canada
Pierre Schaus	UCLouvain, Belgium
Thomas Schiex	INRAE, France
Laurent Simon	LaBRI, Bordeaux Institute of Technology, France
Helmut Simonis	University College Cork, Ireland
Guido Tack	Monash University, Australia
Charlotte Truchet	LS2N, Université de Nantes, France
Pascal Van Hentenryck	Georgia Institute of Technology, USA
Petr Vilím	IBM, Czech Republic
Mark Wallace	Monash University, Australia
Roland Yap	National University of Singapore, Singapore

Additional Reviewers

Aghaei, Sina
Akgün, Özgür
Antuori, Valentin
Artigues, Christian
Belov, Gleb
Besançon, Mathieu
Blais, Nicolas
Boudreault, Raphaël
Cappart, Quentin
Eifler, Leon
Espasa Arxer, Joan
Geibinger, Tobias
Gent, Ian
Hendel, Gregor
Hoffmann, Ruth
Hojny, Christopher
Hu, Xinyi
Huang, Isaac
Huguet, Marie-José
Jefferson, Christopher
Jeong, Jihwan
Karahalios, Anthony
Katsirelos, George
Kletzander, Lucas
Le Bodic, Pierre
Leo, Kevin
Maher, Stephen
Meel, Kuldeep S.
Mischek, Florian
Nagarajan, Harsha
Portoleau, Tom
Prestwich, Steve
Pulatov, Damir
Siu, Charles
Spiegel, Christoph
Van Hoeve, Willem-Jan
Vavrille, Mathieu
Wang, Ruiwei
Winter, Felix
Yang, Hojin
Zhang, Han
Zhuowei, Zhong

Extended Abstracts

The following extended abstracts were accepted for presentation at the conference:

- Marleen Balvert. IRELAND: an MILP-based algorithm for learning interpretable input-output relationships from large binary classification data.
- Nick Doudchenko, Miles Lubin, Aditya Paliwal, Pawel Lichocki, and Ross Anderson. MipConfigBench: A dataset for learning in the space of Mixed-Integer Programming algorithms.
- Eleftherios Manousakis, Grigoris Kasapidis, Chris Kiranoudis, and Emmanouil Zachariadis. A matheuristic for the Production Routing Problem: Infeasibility Space Search and Mixed Integer Programming.
- Thibault Prunet, Nabil Absi, Valeria Borodin, and Diego Cattaruzza. Storage Location Assignment Problem in Fast Pick Areas: A novel formulation and decomposition method.
- Jana Koehler, Josef Bürgler, Urs Fontana, Etienne Fux, Florian Herzog, Marc Pouly, Sophia Saller, Anastasia Salyaeva, Peter Scheiblechner, and Kai Waelti. Cable Tree Wiring - Benchmarking Solvers on a Real-World Scheduling Problem with a Variety of Precedence Constraints.
- Mathijs de Weerdt, Robert Baart, and Lei He. Single-Machine Scheduling with Release Times, Deadlines, Setup Times, and Rejection.

Abstracts

Why You Should Constrain Your Machine Learned Models

Maya Gupta

Didero, USA
founders@didero.com

Abstract. Common use of machine learning is to gather what training examples one can, train a flexible model with some smoothness regularizers, test it on a held-out set of random examples, and *hope* it works well in practice. But we will show that by adding constraints, we can prepare our models better for their futures, and be more certain of their performance. Based on 8 years of experience at Google researching, designing, training, and launching hundreds of machine-learned models, I will discuss dozens of ways that we found one can constrain ML models to produce more robust, fairer, safer, more accurate models that are easier to debug and that when they fail, do so more predictably and reasonably. This talk will focus on two classes of model constraints: shape constraints, and rate constraints. The most common shape constraint is monotonicity, and it has long been known how to learn monotonic functions over one input using isotonic regression. We will discuss new R&D about 6 different practically useful shape constraints, and how to impose them on flexible, mulit-layer models. The second class of constraints, rate constraints, refers to constraints on a classifiers' output statistics, and is commonly used to make classifiers act responsibly for different groups. For example, we may constrain a classifier used globally to be at least 80% accurate on training examples from India or China, as well as minimizing classification errors on average. We will point listeners to Google's open-source Tensor Flow libraries to impose these constraints, and papers with more technical detail.

Contextual Optimization: Bridging Machine Learning and Operations

Adam Elmachtoub

Columbia University, New York, USA
adam@ieor.columbia.edu

Abstract. Many operations problems are associated with some form of a prediction problem. For instance, one cannot solve a supply chain problem without predicting demand. One cannot solve a shortest path problem without predicting travel times. One cannot solve a personalized pricing problem without predicting consumer valuations. In each of these problems, each instance is characterized by a context (or features). For instance, demand depends on prices and trends, travel times depend on weather and holidays, and consumer valuations depend on user demographics and click history. In this talk, we review recent results on how to solve such contextual optimization problems, with a particular emphasis on techniques that blend the prediction and decision tasks together.

Complete Symmetry Breaking Constraints for the Class of Uniquely Hamiltonian Graphs

Avraham Itzhakov and Michael Codish

Department of Computer Science,
Ben-Gurion University of the Negev,
Beer-Sheva, Israel
{itzhakoa,mcodish}@cs.bgu.ac.il

Abstract. Graph search problems are fundamental in graph theory. Such problems include: existence problems, where the goal is to determine whether a simple graph with certain graph properties exists, enumeration problems, which are about finding all solutions modulo graph isomorphism, and extremal problems, where we seek the smallest/largest solution with respect to some target such as the number of edges or vertices in a solution. Solving graph search problems is typically hard due to the enormous search space and the large number of symmetries.

One common approach to break symmetries in constraint programming is to add symmetry breaking constraints which are satisfied by at least one member of each isomorphism class. A symmetry breaking constraint is called *complete* if it is satisfied by exactly one member of each isomorphism class and *partial* otherwise. A universal measure for the size of a symmetry breaking constraint is the size of its representation in propositional logic. All known techniques to define complete symmetry breaking constraints for graph search problems are based on predicates which are exponential in size. There is no known polynomial size complete symmetry breaking constraint for graph search problems.

This paper introduces, for the first time, a complete symmetry breaking constraint of polynomial size for a significant class of graphs: the class of uniquely Hamiltonian graphs. This is the class of graphs that contain exactly one Hamiltonian cycle. We introduce a canonical form for uniquely Hamiltonian graphs and prove that testing whether a given uniquely Hamiltonian graph is canonical can be performed efficiently. Based on this canonicity test, we construct a complete symmetry breaking constraint of polynomial size which is satisfied only by uniquely Hamiltonian graphs which are canonical. We apply the proposed symmetry breaking constraint to determine the, previously unknown, smallest orders for which uniquely Hamiltonian graphs of minimum degree 3 and girths 3 and 4 exist.

Given that it is unknown if there exist polynomial sized complete symmetry breaking constraints for graphs, this paper makes a first step in the direction of identifying specific classes of graphs for which such constraints do exist.

Supported by the Israel Science Foundation, grant 625/17.

Variable Ordering for Decision Diagrams: A Portfolio Approacho

Anthony Karahalios and Willem-Jan van Hoeve

Carnegie Mellon University, Pittsburgh PA 15213, USA
{akarahal,vanhoeve}@andrew.cmu.edu

Abstract. Relaxed decision diagrams have recently been successfully applied within a range of solution methodologies for discrete optimization, including constraint programming, integer linear programming, integer nonlinear programming, and combinatorial optimization. The variable ordering is often of crucial importance for their effectiveness. For example, Bergman et al. [1, 2] demonstrate that a variable ordering that yields a small exact diagram typically also provides stronger dual bounds from the relaxed diagram. When decision diagrams are built from a single top-to-bottom compilation, dynamic variable orderings can be very effective. For example, a recent work by Cappart et al. [3] deploys deep reinforcement learning to dynamically select the next variable during compilation. Dynamic variable orderings are less applicable, however, to compilation via iterative refinement, in which case the ordering must be specified in advance. In this work, we consider variable ordering strategies for the latter case.

Oftentimes there is no single variable ordering strategy that dominates all others for a given set of problem instances. Selecting the best ordering, or more generally the best algorithm, from a set of alternatives is a well-studied problem in artificial intelligence, in the context of *algorithm portfolios*. There are several ways to construct an algorithm portfolio: using static or dynamic features, formulating predictive models at the algorithm or portfolio level, predicting one algorithm to run per instance or creating a schedule of algorithms to run, using a fixed portfolio or updating it online [4]. We consider several different portfolio mechanisms: an offline predictive model of the single best algorithm using classifiers, an online low-knowledge algorithm selection, a static uniform time-sharing portfolio, and a dynamic online time allocator.

As a case study, we consider the graph coloring problem, for which a decision diagram approach was recently introduced [5, 6]. It uses an iterative refinement procedure much like Benders decomposition or lazy-clause generation, by repeatedly refining conflicts in the diagram until the solution is conflict free. Our experimental results show that predictive methods using classification models or exploration phases can lead to more instances solved optimally. However, these methods may lead to delayed optimality results on problem instances that are easy to solve. Another insight is that a mixed portfolio can outperform a clairvoyant selection of the best individual ordering for each

Partially supported by Office of Naval Research Grant No. N00014-18-1-2129 and National Science Foundation Award #1918102.

instance, by yielding a solution with a unique best upper bound from one ordering and a unique best lower bound from a different ordering.

References

1. Bergman, D., Cire, A.A., van Hoeve, W.-J., Hooker, J.N.: Variable ordering for the application of BDDs to the maximum independent set problem. In: Beldiceanu, N., Jussien, N., Pinson, É. (eds.) CPAIOR 2012. LNCS, vol. 7298, pp. 34–49. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29828-8_3
2. Bergman, D., Cire, A.A., van Hoeve, W.-J., Hooker, J. N.: Optimization bounds from binary decision diagrams. Inform. J. Comput. **26**(2), 253–268 (2014)
3. Cappart, Q., Goutierre, E., Bergman, D., Rousseau, L.-M.: Improving optimization bounds using machine learning: decision diagrams meet deep reinforcement learning. In: Proceedings of AAAI, pp. 1443–1451. AAAI Press (2019)
4. Kotthoff, L.: Algorithm selection for combinatorial search problems: a survey. In: Bessiere, C., De Raedt, L., Kotthoff, L., Nijssen, S., O'Sullivan, B., Pedreschi, D. (eds). Data Mining and Constraint Programming. LNCS, vol. 10101, pp. 149–190. Springer, Cham (2016).https://doi.org/10.1007/978-3-319-50137-6_7
5. van Hoeve, W.-J.: Graph Coloring with Decision Diagrams. Under review.http://www.optimization-online.org/DB_HTML/2021/01/8215.html
6. van Hoeve, W.-J.: Graph coloring lower bounds from decision diagrams. In: Bienstock, D., Zambelli, G. (eds.) IPCO 2020. LNCS, vol. 12125, pp. 405–419. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45771-6_31

Contents