Extended Abstract: Constraint Acquisition Based on Solution Counting

Christopher Coulombe
Université Laval, Québec, Canada

Claude-Guy Quimper
Université Laval, Québec, Canada

Abstract

We propose CABSC, a system that performs Constraint Acquisition Based on Solution Counting. In order to learn a Constraint Satisfaction Problem (CSP), the user provides positive examples and a Meta-CSP, i.e. a model of a combinatorial problem whose solution is a CSP. It allows listing the potential constraints that can be part of the CSP the user wants to learn. It also allows stating the parameters of the constraints and imposing constraints over these parameters. The CABSC reads the Meta-CSP using an augmented version of the language MiniZinc and returns the CSP that accepts the fewest solutions among the CSPs accepting all positive examples. This is done using a branch and bound where the bounding mechanism makes use of a model counter. Experiments show that CABSC is successful at learning constraints and their parameters from positive examples.

2012 ACM Subject Classification Computing methodologies → Modeling methodologies

Keywords and phrases Constraint acquisition, CSP, Model counting, Solution counting

Digital Object Identifier 10.4230/LIPIcs.CP.2022.27

1 Introduction

Constraint solvers are used to solve complex combinatorial problems. They require an expert to model the problem using the constraints available in the solver. The model creation is a crucial step, but is often time-consuming. One way to save time to the expert is to suggest a model based on sample solutions. For instance, a hospital that wants to automatize the creation of their work schedules for its staff might provide to the experts previous schedules. Assisted with software, the expert wants to discover what constraint generated the examples.

When two constraints are candidates for a model, the one that was the most likely used to generate the sample solutions is the most restrictive one [5]. To decide which constraint is the most restrictive, there are statistical approaches [4, 5] and approaches based on a ranking system [2]. Current methods analyze the constraint in isolation. However, a constraint that accepts many solutions could become a great option as the interactions between the constraints could make this choice more favorable. We propose the first approach that takes into account. It uses a model counter to make sure that the constraints suggested to the expert are those that are the most likely to explain the observed sample solutions.

In this paper, we propose CABSC, an algorithm for Constraint Acquisition Based on Solution Counting. CABSC uses examples of solutions to evaluate which constraints to keep from a chosen set of candidates. The selection process is based on solutions counting using model counters, an approach which differs from the current methods [5, 1, 2, 3]. We briefly describe our approach in Section 2, followed by an introduction to our results in Section 3.

2 The CABSC approach

CABSC models the process of learning constraints as a Meta-CSP. A Meta-CSP is a combinatorial problem whose solution is a CSP. In our case, the solution is the CSP we learn from the examples. When modeling the Meta-CSP, we list the mandatory constraints, i.e. the constraints that we know belong to the model, and also the possible constraints, those that could belong to the model. The variables of the Meta-CSP encode the possible activation of a constraint and also the parameters of the constraints, such as the coefficients of a linear constraint. Solving the Meta-CSP provides the learned model. To do so, we use a branch and bound to decide which constraint to keep and identify the values of the parameters. Our approach uses constraint programming to model a Meta-CSP and to define a family of CSPs from which we can learn. We therefore do not aim to learn any CSP but the optimal CSP among a set programmed through constraint programming. This approach is inspired from regression where one defines a family of functions (e.g. linear functions) and aims at finding the function from this family that best fits the data. Here, we aim at finding the CSP from a family of CSPs defined by the Meta-CSP that best explains the examples.

As there are multiple constraints that could belong to the learned model, we follow Beldiceanu and Simonis [2] and Picard-Cantin et al. [4] by selecting the constraints that minimize the number of solutions. However, instead of analyzing the constraints individually, our system reasons globally which allows us to consider multiple different constraints at once. The solution to our Meta-CSP is therefore a CSP whose constraints are satisfied by all observed examples and is as restrictive as possible, i.e. it minimizes the number of solutions.

Our approach has two main differences from most existing methods. The first difference is that a Meta-CSP is declared with constraint programming to define a family of CSPs from which we can learn. A second difference is that we use a criterion with a global view on the model to learn by considering the constraints as a whole instead of individually.

3 Results and Discussion

We tested CABSC on a nurse rostering problem with four benchmarks. Figure 1 presents the results obtained when running the solver on one of the benchmarks called **Overtime**. On the y-axis is the number of examples that are given to the solver. On the x-axis is the proportion of instances for which the solution is the best one returned by the solver, the second best, the third best, or whether the CSP that was used to produce the examples does not appear at all in the top-3 learned models.

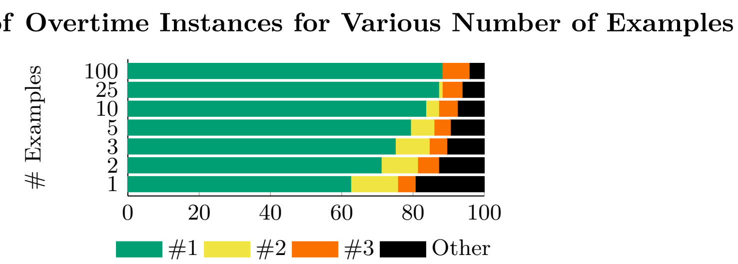

Figure 1 Classification of some instances in percentage for each number of examples.

4 Conclusion

In our article, we introduced CABSC, a technique for Constraint Acquisition Based on Solution Counting. Our approach learns the CSP that accepts all provided examples but that minimizes the size of its solution space. We prove that this criterion can return good solutions and that CABSC is successful at learning constraints and their parameters from positive examples.

References

1 Nicolas Beldiceanu and Helmut Simonis. A constraint seeker: Finding and ranking global constraints from examples. In *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming (CP 2011)*, pages 12–26, 2011.

2 Nicolas Beldiceanu and Helmut Simonis. A model seeker: Extracting global constraint models from positive examples. In *Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming (CP 2012)*, pages 141–157, 2012.

3 Christian Bessiere, Remi Coletta, Emmanuel Hebrard, George Katsirelos, Nadjib Lazaar, Nina Narodytska, Claude-Guy Quimper, and Toby Walsh. Constraint acquisition via partial queries. In Francesca Rossi, editor, *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI-19)*, pages 475–481, 2013.

4 Émilie Picard-Cantin, Mathieu Bouchard, Claude-Guy Quimper, and Jason Sweeney. Learning parameters for the sequence constraint from solutions. In *Proceedings of the 22nd International Conference on Principles and Practice of Constraint Programming (CP 2016)*, pages 405–420, 2016.

5 Émilie Picard-Cantin, Mathieu Bouchard, Claude-Guy Quimper, and Jason Sweeney. Learning the parameters of global constraints using branch-and-bound. In *Proceedings of the 23rd International Conference on Principles and Practice of Constraint Programming (CP 2017)*, pages 512–528, 2017.