Solve Optimization Problems with Unknown Constraint Networks

Mohamed-Bachir Belaid[1], Arnaud Gotlieb[1], and Nadjib Lazaar[2]

[1]Simula Research Laboratory, Oslo, Norway
[2]LIRMM, University of Montpellier, CNRS, Montpellier, France
{*bachir@simula.no, arnaud@simula.no, lazaar@lirmm.fr*}

Abstract

In most optimization problems, users have a clear understanding of the function to optimize (e.g., minimize the makespan for scheduling problems). However, the constraints may be difficult to state and their modelling often requires expertise in Constraint Programming. Active constraint acquisition has been successfully used to support non-experienced users in learning constraint networks through the generation of a sequence of queries. In this paper, we propose Learn&Optimize, a method to solve optimization problems with known objective function and unknown constraint network. It uses an active constraint acquisition algorithm which learns the unknown constraints and computes boundaries for the optimal solution during the learning process. As a result, our method allows users to solve optimization problems without learning the overall constraint network.

1 Introduction

Constraint Programming (CP) is a framework to model and solve satisfaction and optimization problems. Modeling problems into constraints is a crucial step in CP that usually requires expertise rarely available. In order to ease the creation of CP models, Bessiere et.al introduced constraint acquisition in [7] with the aim to learn CP models based on a set of assignment instances (passive learning) or specific queries that help to classify assignments (active learning).

More precisely, sevral techniques and tools are available for constraint acquisition. Conacq [5] learns constraints that correctly classify a given set of positive (solutions) and negative (non-solutions) assignments in a passive way. Quacq [4] is an active learning algorithm that accepts or rejects constraints through interacting with the user with queries formed of partial assignments. Quacq was further extended to learn all possible constraints from a negative query in [2], to speed-up queries generation in [1] and to exploit the structure of learned problems in [13]. Other approaches for learning constraint models have also been developed, in particular to acquire global constraints such as in ModelSeeker [3].

Learning of optimization models is also a topic that received recently a lot of attention in the literature. Elmachtoub et.al propose a general framework called *Smart Predict then Optimize* (SPO) where parameters to optimization problems are predicted using advanced machine learning methods [8]. Similarly, Mandi et al. proposed PREDICT&OPTIMIZE, a method that solves optimization problems where weights for the objective function are unknown in [10].

Since problems are usually modeled up to the point where they are solved, Bessiere et.al introduced in [6] a method where the acquisition process stops once a complete positive assignment is found. In this paper we extend the method of [6] to solve optimization problems where the function to optimize is known and the constraints have to be learned. Our idea is inspired from [9] where a precedence graph for the assembly line balancing problem is learned from past feasible (i.e., only positive) production sequences. During the learning process the method attempts to find the optimal solution using the upper-bound (solution to the maximum graph) and the lower-bound (solution to the minimum graph).[1] Results in [9] (and latter in [12]) confirm that, in many cases, the bounds are equal or very close to each other. This method is limited to solving a specific problem (i.e., the assembly line balancing problem) and the learning part is limited to using feasible production sequences, to remove inconsistent precedence relations, and to direct interviews to confirm mandatory relations.

In this paper, we propose, LEARN&OPTIMIZE, a general solution that uses an active constraint acquisition method (e.g., QUACQ) for the constraint learning part. During the acquisition process, LEARN&OPTIMIZE computes upper and lower bounds for the optimal value. If these two bounds are equal, then optimality is proven. If these boundaries are close, we can say that a good-enough quality solution is found. This work falls under the interactive optimization framework, where users are allowed to interact with the system to improve solutions and find optimal ones quickly (please refer to [11] for a general overview).

2 Learn&Optimize

In this section, we introduce LEARN&OPTIMIZE, a method to solve optimization problems with unknown constraints.

[1]See Figure 3 in [9] for more details.

2.1 Notations

The learner and the user share a common knowledge to communicate altogether, which is materialized by the vocabulary (X, D, f) where X is a set of n variables, D a specified finite domain, and f an objective function on X.[2] In addition to the vocabulary, the learner owns the language $\mathcal{L}$, from which it can build the basis, $\mathcal{B}$, a set of constraints on specified sets of variables from X. An assignment $e \in D^X$, is *positive* (or feasible) if it satisfies the target network T, and it is *negative* otherwise. We denote by e^* an optimal solution to "f w.r.t. T". Here, we assume that the user is always right and that $T \subseteq \mathcal{B}$.

2.2 Learn&Optimize Description

Learn&Optimize is presented in Algorithm 1. It takes as input the vocabulary (X, D, f), the language $\mathcal{L}$, a time bound `cutoff`, a feasible solution e_p,[3] and a precision value ϵ. Learn&Optimize returns the bounds (lb, ub) and two assignments characterizing the bounds (e_l, e_u). First, the basis is created in line 4 on $(X, \mathcal{L})$ w.r.t. the given feasible e_p. Here, we keep in $\mathcal{B}$ only constraints accepting e_p. That is, having a feasible solution e_p is sufficient to guarantee that constraints in $\mathcal{B}$ accept at least one solution, while $T \subseteq \mathcal{B}$ holds. Then, the learned network L is initialized to the empty set and the flag *optimal* to $false$ (lines 5-6). Learn&Optimize loops until an optimal solution w.r.t. a given precision ϵ is found $(up - lb \leq \epsilon)$, the optimal solution to (L) given f is classified positive, or the learning and optimization time reaches the `cutoff` time bound (line 7). At each iteration, we generate an example to submit to the user (line 9). If the submitted example e is feasible (i.e., $e \models T$), we reduce $\mathcal{B}$ by removing all constraints not accepting e (line 10). If the user says *no*, a learning process based on QuAcq is called, where a constraint is learned from a negative example (line 11). Doing so, Learn&Optimize ensures to change the state of the basis $\mathcal{B}$ and/or the learned network L and to have new bounds (lb, ub) at each iteration. Note that the lower and the upper bounds correspond to objective values of optimal solutions e_l and e_u of, respectively, (L) and $(L \cup \mathcal{B})$ given f (lines 12-13). If the process reaches a precision of ϵ or e_l is classified as positive (line 14), Learn&Optimize returns the optimal solution ($optimal = true$). Otherwise, Learn&Optimize reaches the time bound value of `cutoff` and returns a near-optimal solution.

[2]Throughout the paper, we assume that it is a minimization problem.

[3]Handcrafted, simple to generate, but not necessary a solution of good quality.

Algorithm 1: LEARN&OPTIMIZE: Solve an optimization problem with known objective function, f, and unknown constraint network, T.

```
1  In: $(X, D, f)$; $\mathcal{L}$; cutoff; $e_p$; $\epsilon$;
2  Out: $((e_l, lb), (e_u, ub))$;
3  begin
4      $\mathcal{B} \leftarrow$ CreateBasis$(X, \mathcal{L}, e_p)$;
5      $L \leftarrow \emptyset$;
6      $optimal \leftarrow false$;
7      do in $t <$ cutoff
8          while $\neg optimal$ do
9              $e \leftarrow$ queryGenerator$(X, D, L, B)$;
10             if $ask(e) = yes$ then reduce$(\mathcal{B}, e)$;
11             else learn$(L, B, e)$;
12             $e_l \leftarrow OptSol(L, f)$; $e_u \leftarrow OptSol(L \cup \mathcal{B}, f)$;
13             $lb \leftarrow f(e_l)$; $ub \leftarrow f(e_u)$;
14             if $((ub - lb) \leq \epsilon)$ or $(ask(e_l) = yes)$ then
15                 $optimal \leftarrow true$;
16     return $((e_l, lb), (e_u, ub))$;
```

3 Conclusion

In this paper, we have presented, LEARN&OPTIMIZE, a method to solve optimization problems in the context where the optimization function is known and the constraint network is not. Some evidence from the literature show that optimal solutions can be found without necessary fully modelling the problem [9]. In our future work, we will further investigate the application of LEARN&OPTIMIZE to some well-chosen real world problems.

References

[1] H. A. Addi, C. Bessiere, R. Ezzahir, and N. Lazaar. Time-bounded query generator for constraint acquisition. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 1–17. Springer, 2018.

[2] R. Arcangioli, C. Bessiere, and N. Lazaar. Multiple constraint aquisition. In *IJCAI: International Joint Conference on Artificial Intelligence*, pages 698–704, 2016.

[3] N. Beldiceanu and H. Simonis. A model seeker: Extracting global

constraint models from positive examples. In *International Conference on Principles and Practice of Constraint Programming*, pages 141–157. Springer, 2012.

[4] C. Bessiere, R. Coletta, E. Hebrard, G. Katsirelos, N. Lazaar, N. Narodytska, C.-G. Quimper, and T. Walsh. Constraint acquisition via partial queries. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[5] C. Bessiere, R. Coletta, F. Koriche, and B. O'Sullivan. A sat-based version space algorithm for acquiring constraint satisfaction problems. In *European Conference on Machine Learning*, pages 23–34. Springer, 2005.

[6] C. Bessiere, R. Coletta, and N. Lazaar. Solve a constraint problem without modeling it. In *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, pages 1–7. IEEE, 2014.

[7] C. Bessiere, F. Koriche, N. Lazaar, and B. O'Sullivan. Constraint acquisition. *Artificial Intelligence*, 244:315–342, 2017.

[8] A. N. Elmachtoub and P. Grigas. Smart "predict, then optimize". *CoRR*, abs/1710.08005, 2017.

[9] H. Klindworth, C. Otto, and A. Scholl. On a learning precedence graph concept for the automotive industry. *European journal of operational research*, 217(2):259–269, 2012.

[10] J. Mandi, P. J. Stuckey, T. Guns, et al. Smart predict-and-optimize for hard combinatorial optimization problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1603–1610, 2020.

[11] D. Meignan, S. Knust, J.-M. Frayret, G. Pesant, and N. Gaud. A review and taxonomy of interactive optimization methods in operations research. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(3):1–43, 2015.

[12] C. Otto and A. Otto. Multiple-source learning precedence graph concept for the automotive industry. *European Journal of Operational Research*, 234(1):253–265, 2014.

[13] D. C. Tsouros, K. Stergiou, and C. Bessiere. Structure-driven multiple constraint acquisition. In *International Conference on Principles and Practice of Constraint Programming*, pages 709–725. Springer, 2019.